


# **BEST PRACTICES IN IMPLEMENTING CONTAINER IMAGE PROMOTION PIPELINES**

A man in a grey jumpsuit is shown from the chest up, carrying a large, black, cylindrical container on his back. He is making peace signs with both hands. The word "CONTAINERS" is overlaid in white, bold, sans-serif font across his chest. The background is a dark, industrial-looking interior with various mechanical components and a red object on the left.

**CONTAINERS**

A large orange circle with a thin black outline, containing white text.

**SOFTWARE  
I LIKE**

A large blue circle with a thin black outline, containing white text.

**SOFTWARE  
I KNOW  
REALLY WELL**

**I LIKE YOU**



**BUT I DON'T TRUST YOU**



# BARUCH SADOGURSKY

CHIEF STICKER OFFICER

(ALSO 🎩 OF DEVELOPER ADVOCACY)



JBARUCH@JFROG.COM

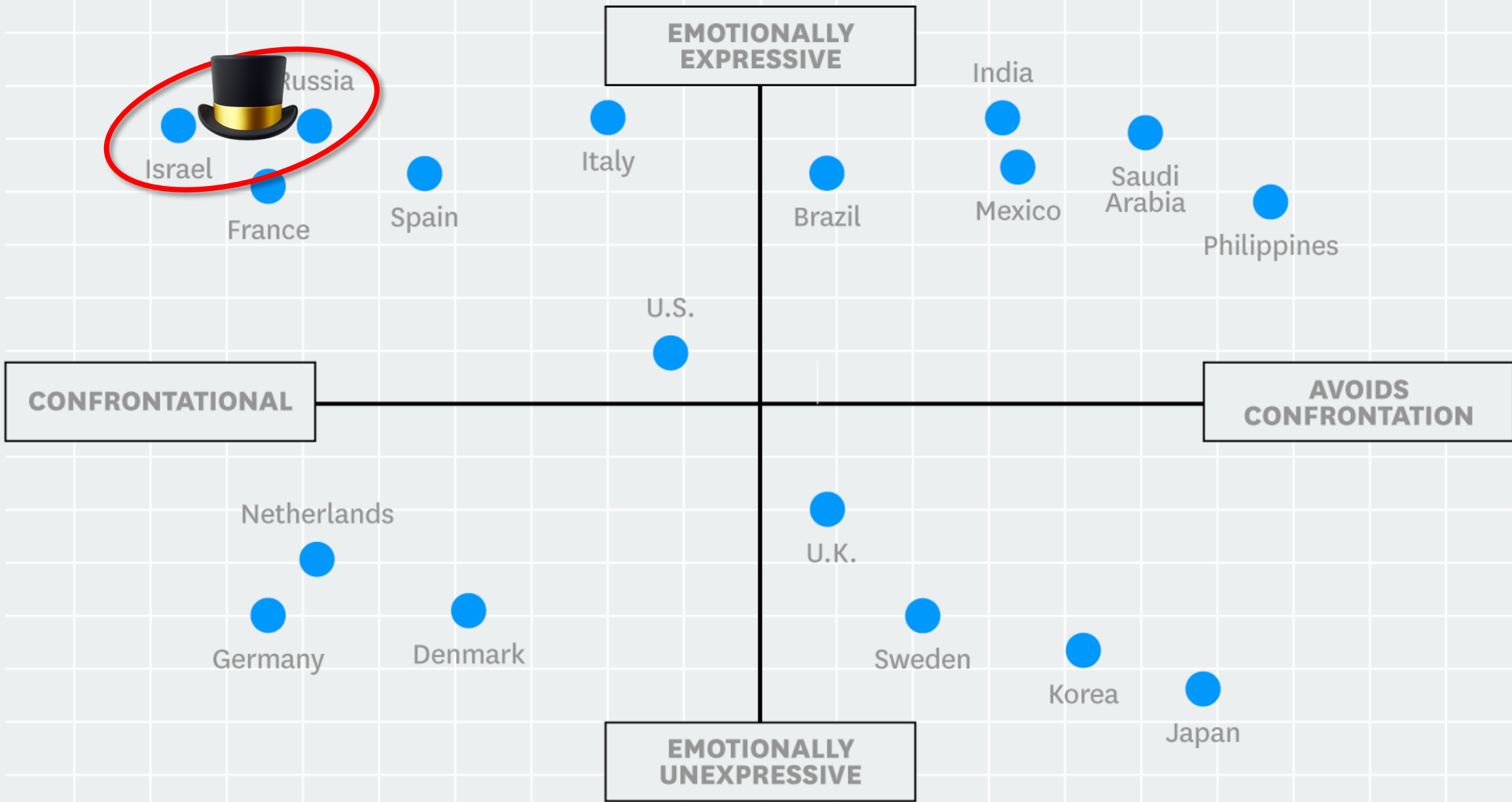


@JBARUCH



+1(408)890-9281





@ErinMeyerINSEAD's "Culture Map"

# SHOWNOTES

➤ <http://jfrog.com/shownotes>

➤ Slides

➤ Video

➤ Links

➤ Comments, Ratings

➤ Raffle

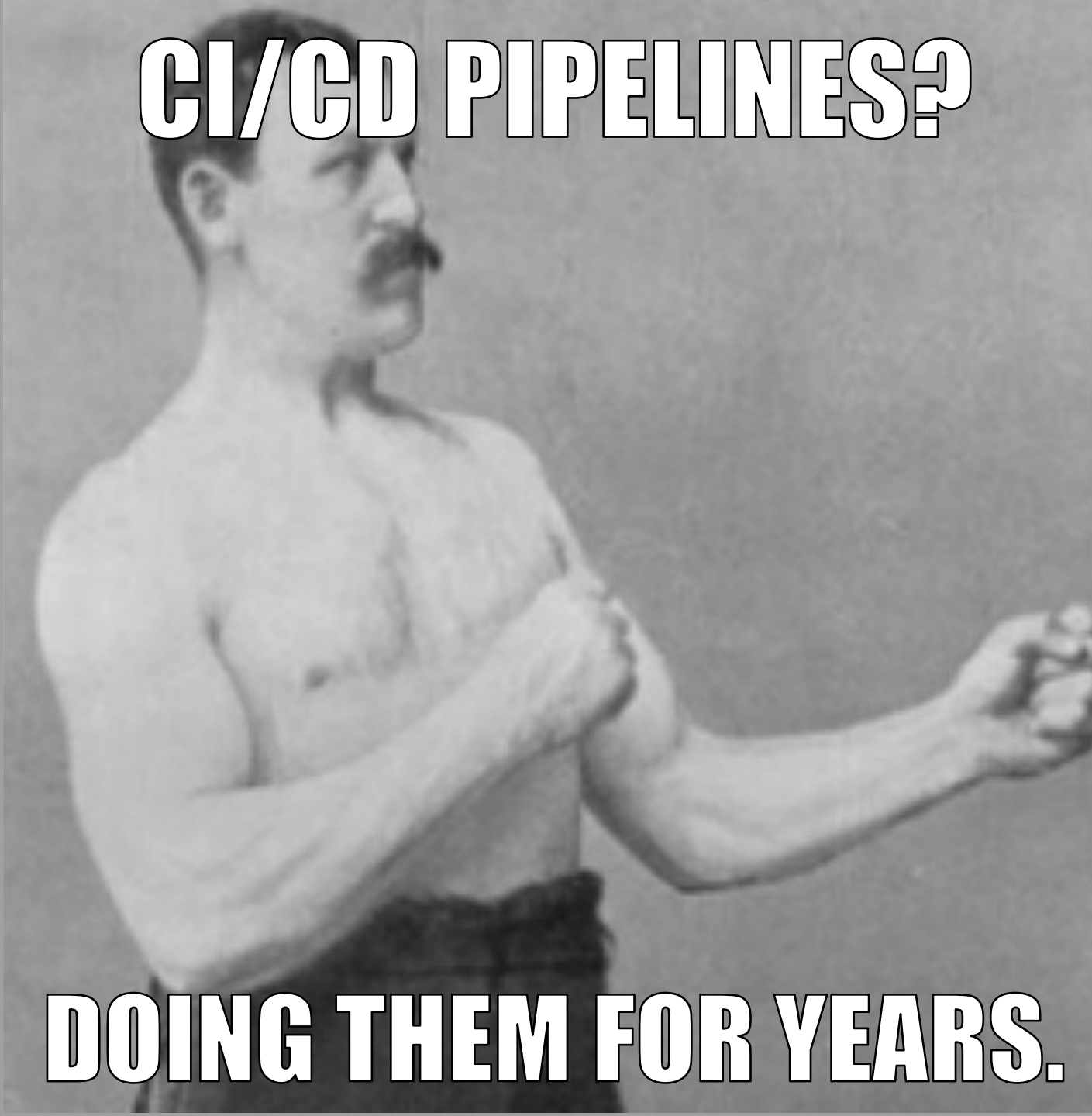
**DO WE HAVE AN EXISTING PATTERN?**

**DO WE NEED TO ADAPT IT?**

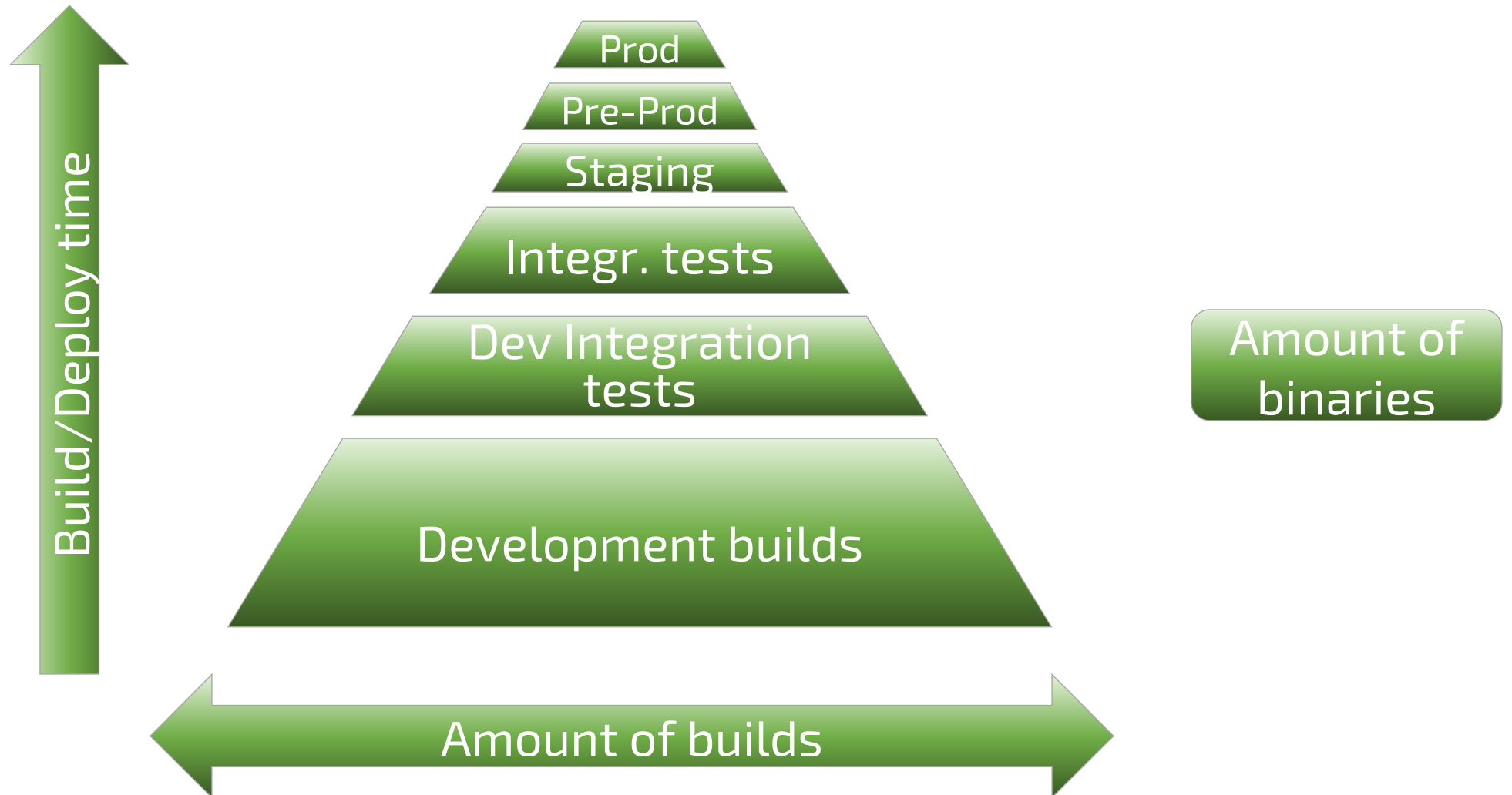


**CI/CD PIPELINES?**

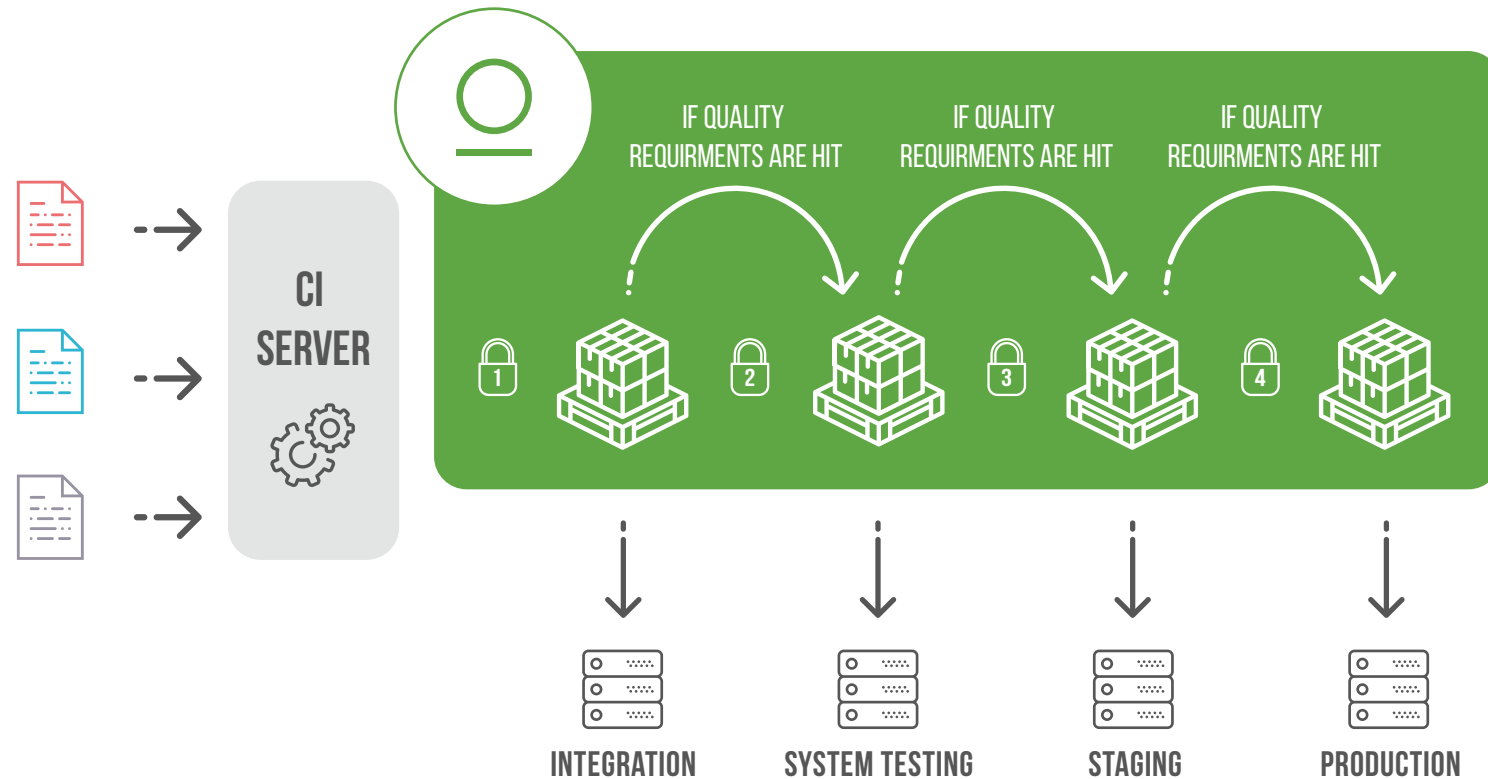
**DOING THEM FOR YEARS.**



# THE PROMOTION PYRAMID



# PIPELINE: QUALITY GATES AND VISIBILITY



@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



- QUALITY GATES -

# \$docker build



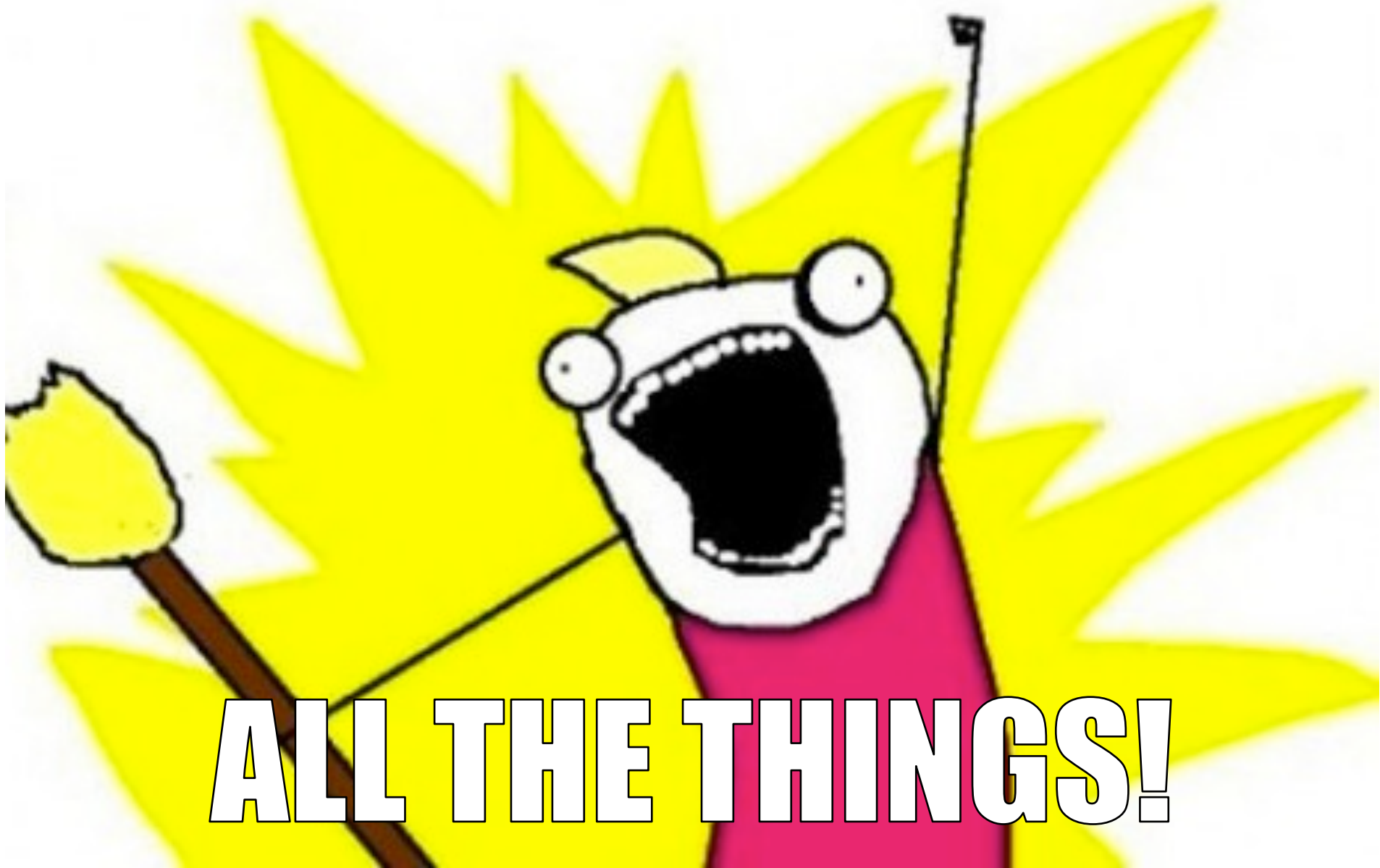
@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>

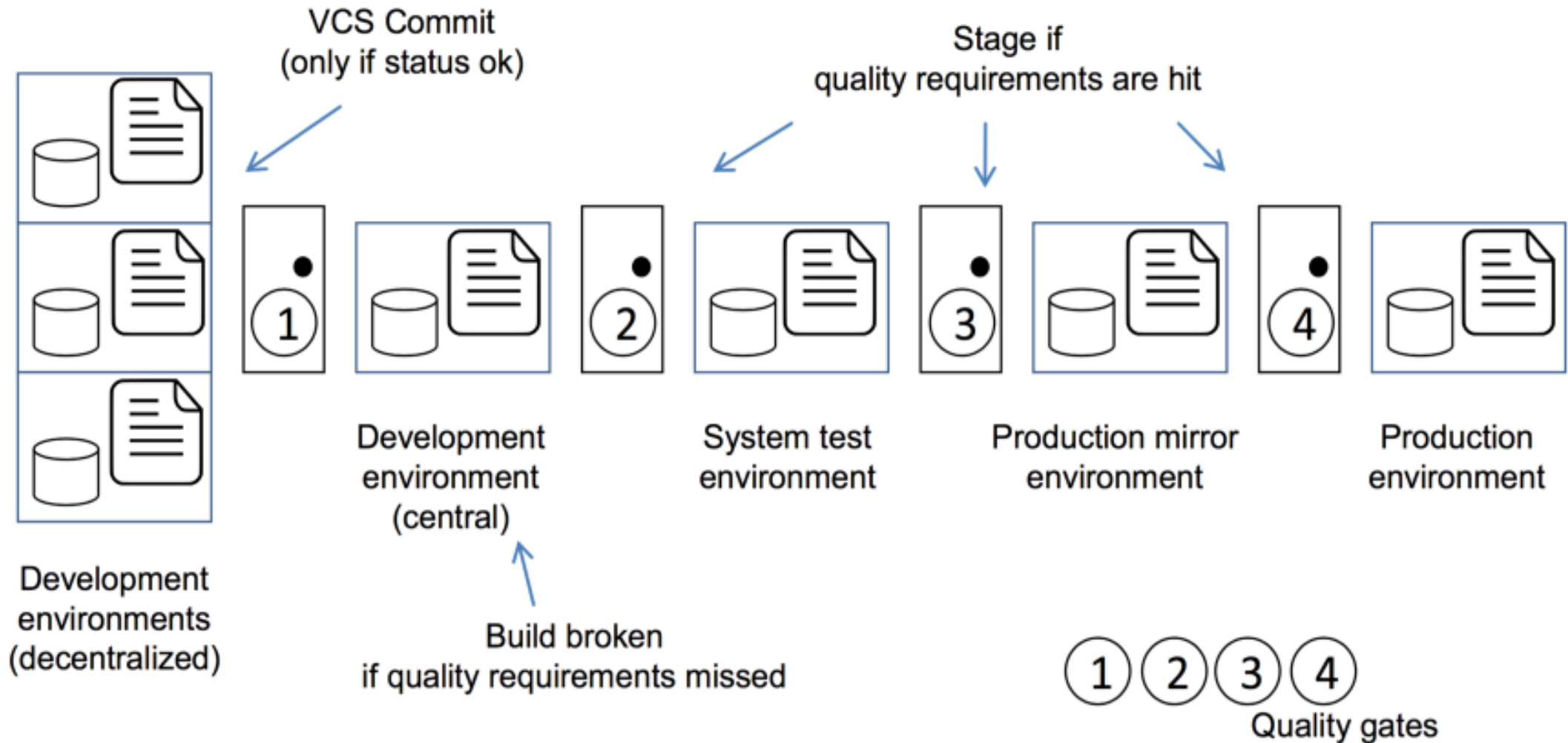


# DOCKER BUILD



# ALL THE THINGS!

# LET'S docker build **IN EVERY ENV!**





**FAST AND CHEAP BUILDS**

**NOT ALWAYS THE WAY TO GO**

# THAT'S WHY.

FROM ubuntu

Latest version

RUN apt-get install -y software-properties-common python

RUN apt-get install -y nodejs

RUN mkdir /var/www

Latest version

ADD app.js /var/www/app.js

Latest version

Latest version

CMD ["/usr/bin/node", "/var/www/app.js"]



# THAT'S WHY.

FROM ubuntu:19.04

Better now?

RUN apt-get install -y software-properties-common python

RUN apt-get install -y nodejs

RUN mkdir /var/www

ADD app.js /var/www/app.js

CMD ["/usr/bin/node", "/var/www/app.js"]

# THAT'S WHY.

FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98

And now?

RUN apt-get install -y software-properties-common python

RUN apt-get install -y nodejs

RUN mkdir /var/www

ADD app.js /var/www/app.js

CMD ["/usr/bin/node", "/var/www/app.js"]

# THAT'S WHY.

```
FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98
```

```
RUN apt-get install -y software-properties-common python
```

```
RUN apt-get install -y nodejs
```

```
RUN mkdir /var/www
```



What about those?

```
ADD app.js /var/www/app.js
```

```
CMD ["/usr/bin/node", "/var/www/app.js"]
```

# THAT'S WHY.

FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98

RUN mvn clean install



What about this?

CMD "java -jar Main.class"



# THAT'S WHY.

FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98

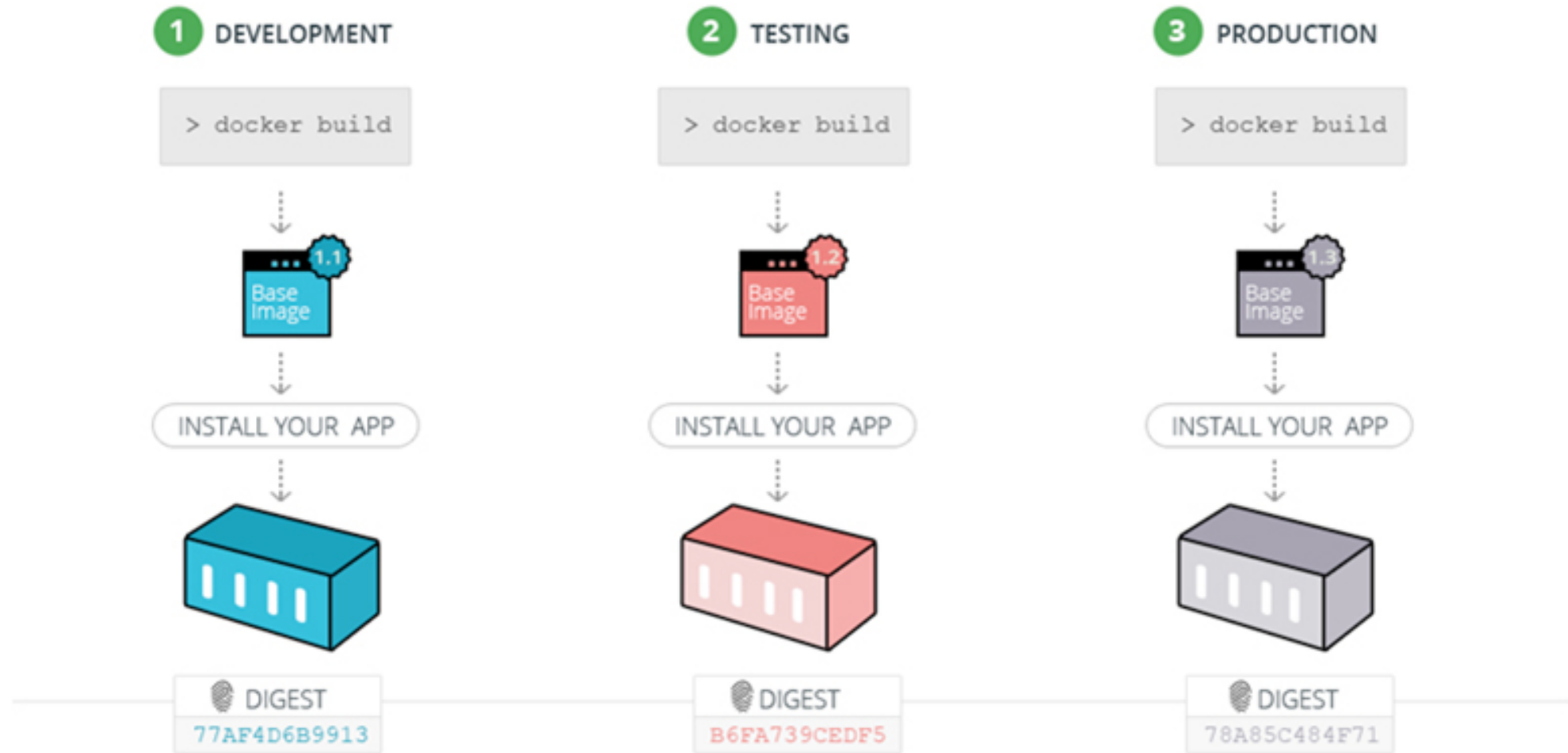
RUN download\_random\_sh\*t\_from\_the\_internet.sh



And how about this?

CMD ["/usr/bin/node", "/var/www/app.js"]

# THAT'S WHY YOU DON'T TRUST DOCKER



@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>

**I DON'T ALWAYS BUILD PROMOTION PIPELINES**



**BUT WHEN I DO, IT'S WITH  
IMMUTABLE AND STABLE BINARIES**

## 1 DEVELOPMENT

```
> docker build
```



INSTALL YOUR APP



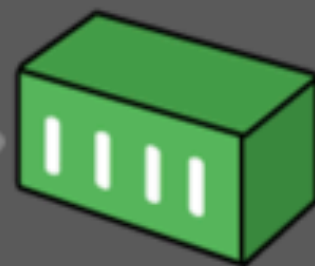
DIGEST

77AF4D6B9913

## 2 TESTING

```
> build once
```

```
> automatically promote the same image  
through the pipeline to production
```



DIGEST

77AF4D6B9913

## 3 PRODUCTION



DIGEST

77AF4D6B9913

PROMOTE

PROMOTE



# WHAT'S UP WITH THE GATES?!

- QA SHOULDN'T TEST DEV IMAGES
- NON-TESTED IMAGES SHOULDN'T BE STAGED
- NON-STAGED, NON-TESTED OR DEV IMAGES SHOULDN'T END UP IN PRODUCTION!!!

**LET'S BUILD ROCK-SOLID PIPELINE!**





# HOW DO I SEPARATE DEV FROM PROD?!

@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



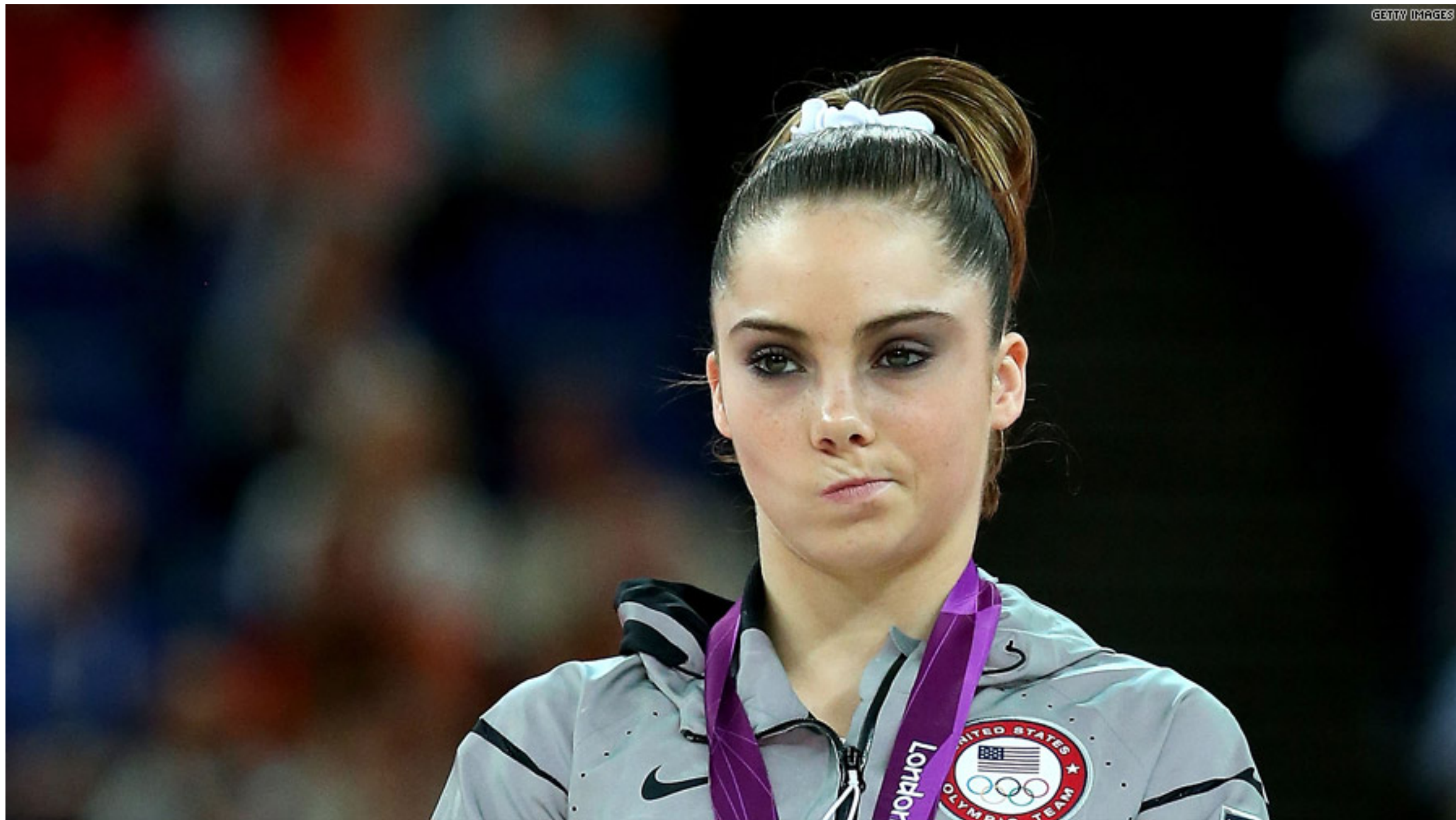
# OPTION 1: METADATA TAGS

## LABEL

```
LABEL <key>=<value> <key>=<value> <key>=<value> ...
```

The `LABEL` instruction adds metadata to an image. A `LABEL` is a key-value pair. To include spaces within a `LABEL` value, use quotes and backslashes as you would in command-line parsing. A few usage examples:

```
LABEL "com.example.vendor"="ACME Incorporated"  
LABEL com.example.label-with-value="foo"  
LABEL version="1.0"  
LABEL description="This text illustrates \  
that label-values can span multiple lines."
```



# OPTION 2: DOCKER REPOSITORIES

## Repositories

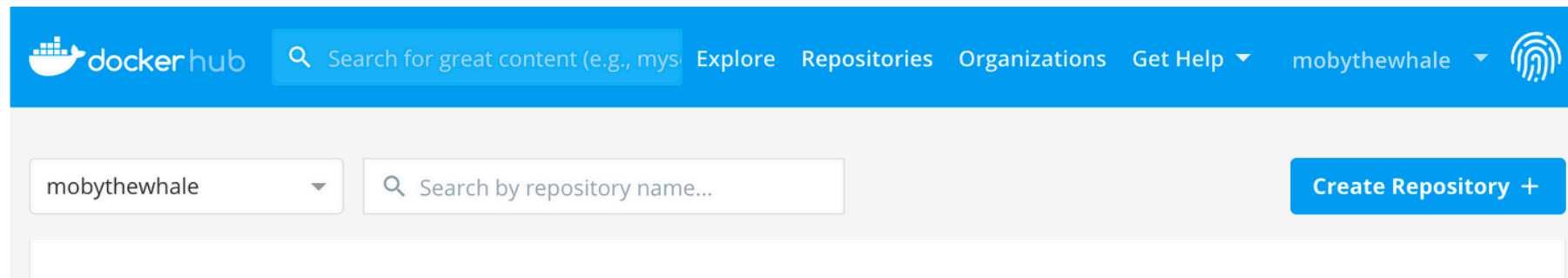
*Estimated reading time: 5 minutes*

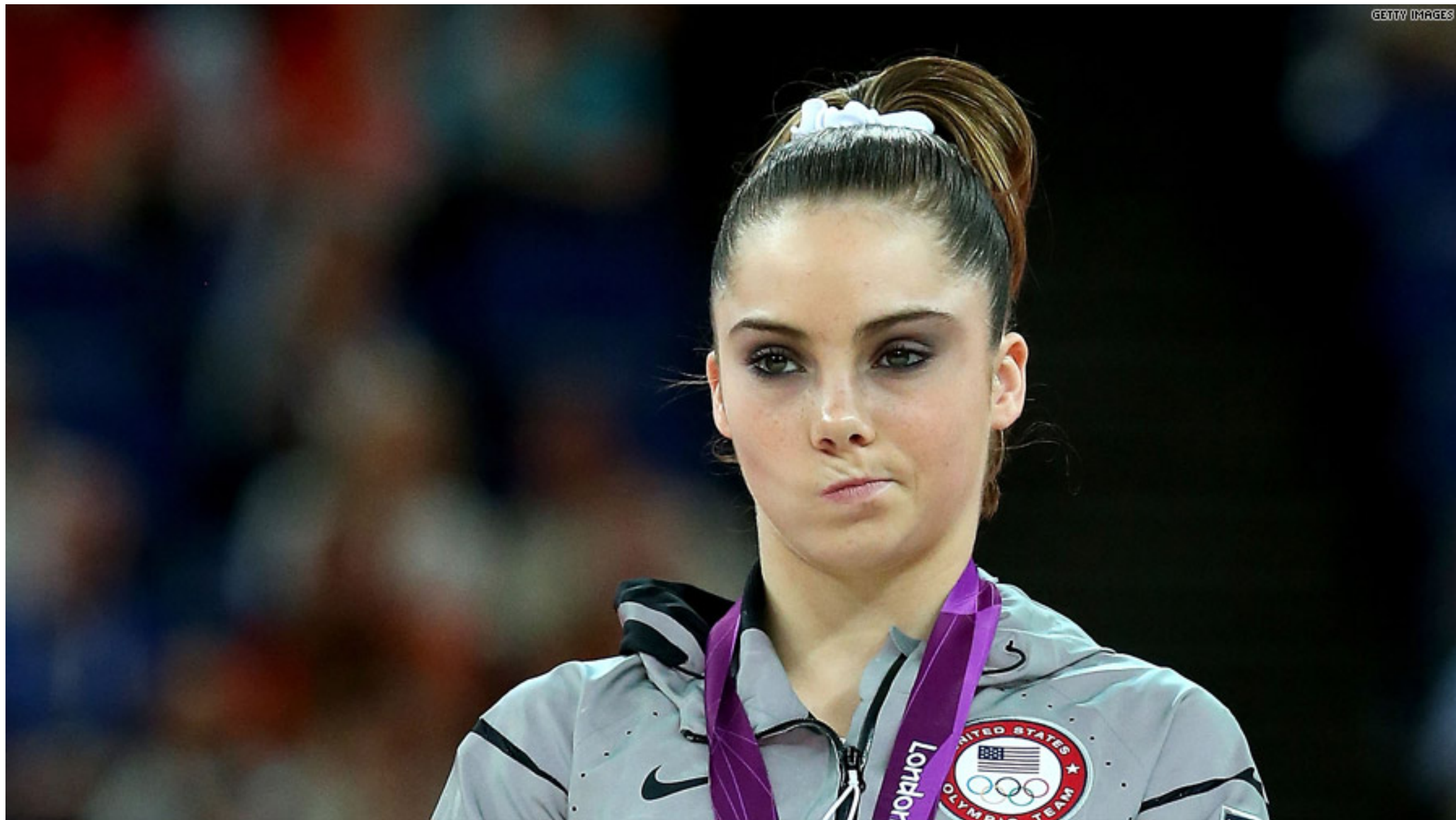
Docker Hub repositories allow you share container images with your team, customers, or the Docker community at large.

Docker images are pushed to Docker Hub through the `docker push` command. A single Docker Hub repository can hold many Docker images (stored as **tags**).

## Creating repositories

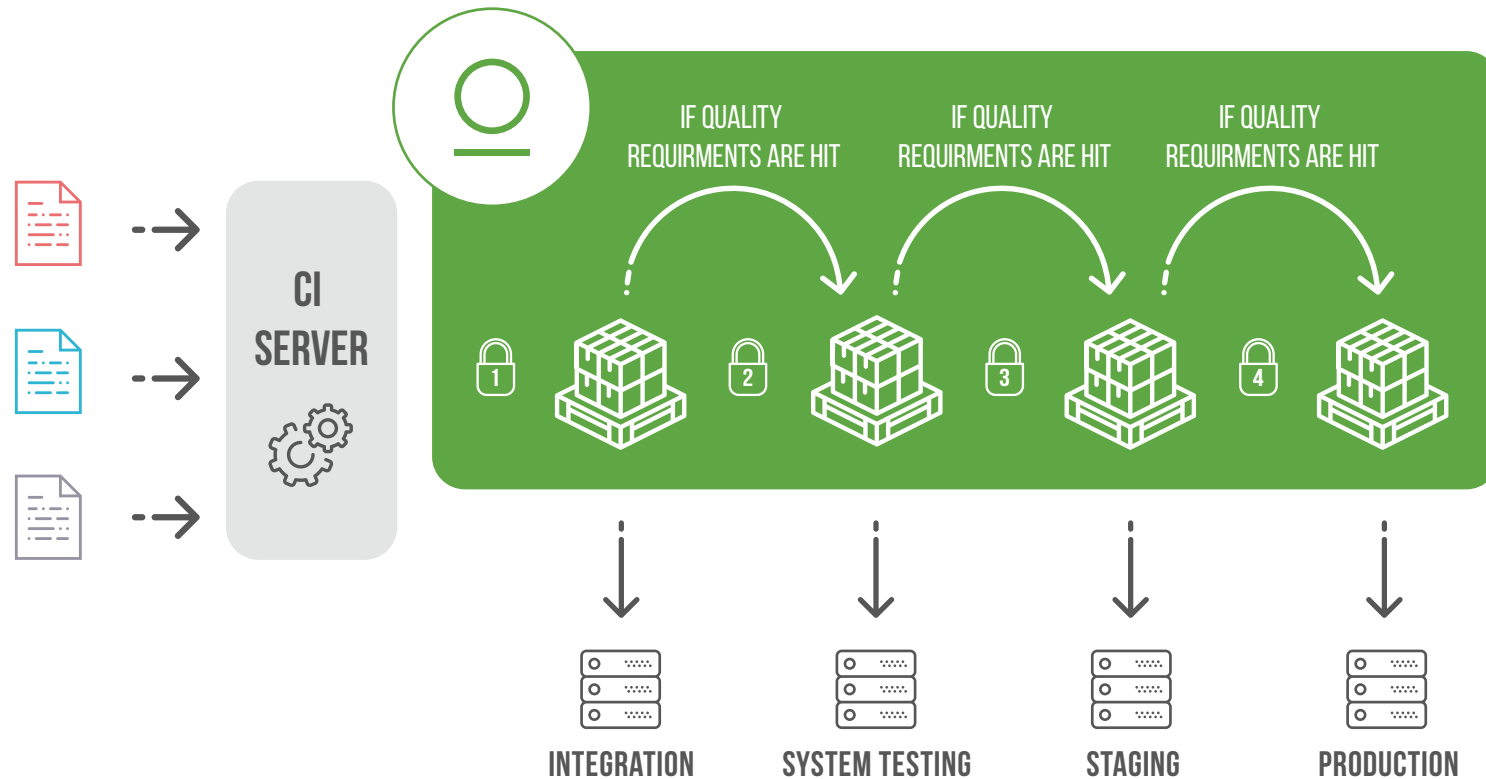
To create a repository, sign into Docker Hub, click on **Repositories** then **Create Repository**:







# SEPARATE REGISTRIES PER ENVIRONMENT



@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



- QUALITY GATES -



# TRUMPED-UP LIMITATIONS



“640K OUGHT TO BE ENOUGH FOR ANYBODY.”

BILL GATES

© Lifhack Quotes

@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



# THE ANATOMY OF DOCKER TAG

## Tag an image for a private repository

To push an image to a private registry and not the central Docker registry you must tag it with the registry hostname and port (if needed).

```
$ docker tag 0e5574283393 myregistryhost:5000/fedora/httpd:version1.0
```

Wait a second, how can I  
have more than one  
registry per host now?!



@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>

# HOW CAN WE SUPPORT THIS?

`https://host:8081/registry/docker-dev/busybox`

`https://host:8081/registry/docker-qa/busybox`

`https://host:8081/registry/docker-staging/busybox`

`https://host:8081/registry/docker-prod/busybox`

**“ONE REGISTRY PER HOST OUGHT TO BE ENOUGH FOR  
ANYBODY.”**



# PANIC!



@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>

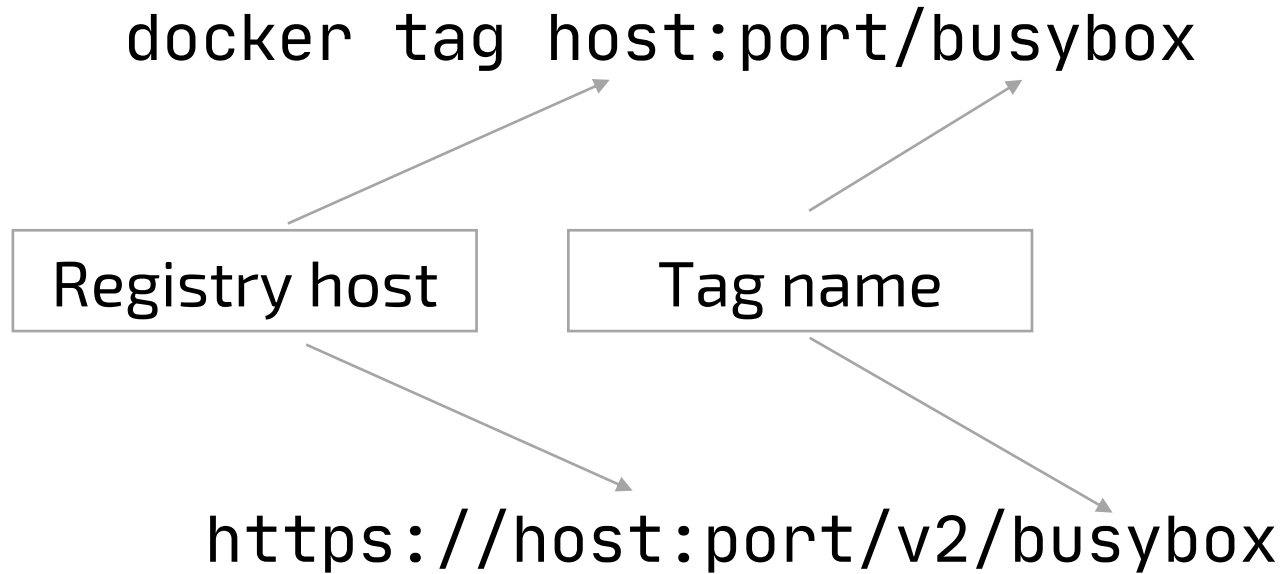
# VIRTUAL HOSTS/PORTS TO THE RESCUE

```
docker tag host:port/busybox
```

Registry host

Tag name

# VIRTUAL HOSTS/PORTS TO THE RESCUE





# VIRTUAL HOSTS/PORTS TO THE RESCUE

```
docker tag host:port/busybox
```

Registry host

Tag name

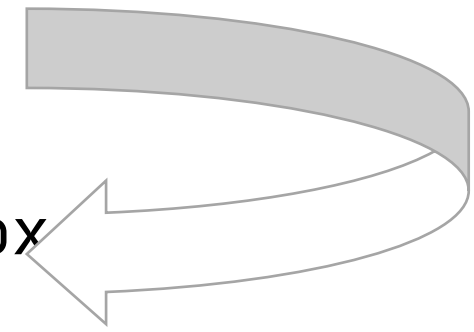
```
https://host:port/v2/busybox
```

```
https://host:8081/registry/docker-dev/busybox
```

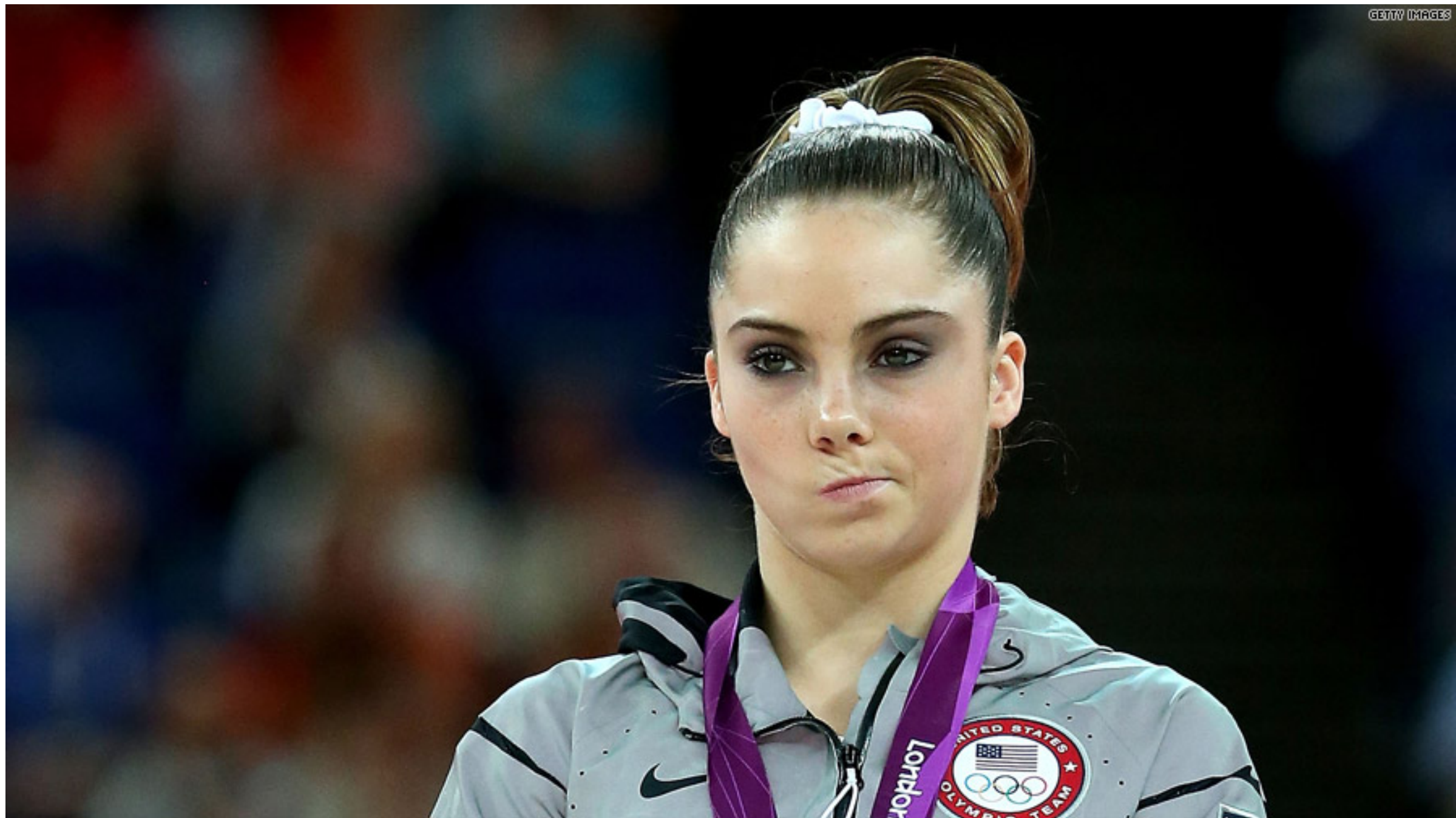
Context name

Registry name

Tag name



```
server {  
    listen 5001;  
  
    server_name 192.168.99.100;  
    if ($http_x_forwarded_proto = '') {  
        set $http_x_forwarded_proto $scheme;  
    }  
    rewrite ^/(v1|v2)/(.*) /artifactory/api/docker/docker-dev/$1/$2;  
    ...  
}
```




# LET'S ABUSE THINGS!

## Tag an image for a private repository

To push an image to a private registry and not the central Docker registry you must tag it with the registry hostname and port (if needed).

**no-one uses this anyway**

```
$ docker tag 0e5574283393 myregistryhost:5000/fedora/httpd:version1.0
```



# LET'S ABUSE THINGS!

## Tag an image for a private repository

To push an image to a private registry and not the central Docker registry you must tag it with the registry hostname and port (if needed).

```
$ docker tag 0e5574283393 myregistryhost:5000/staging/httpd:version1.0
```

# BUT THEN YOU REALIZE...

Wait a second, now I need to pull, retag and push for every step?!

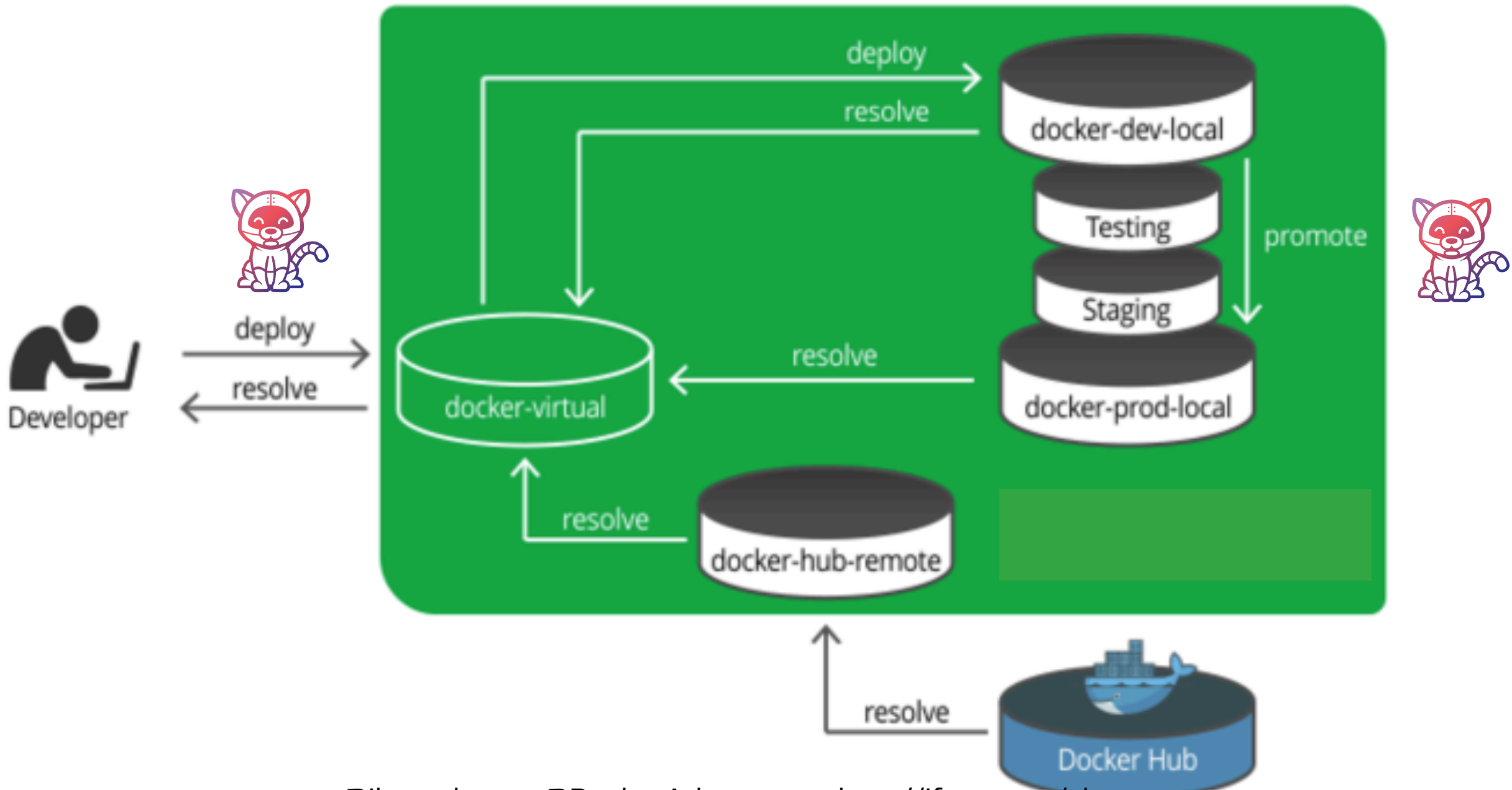


@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



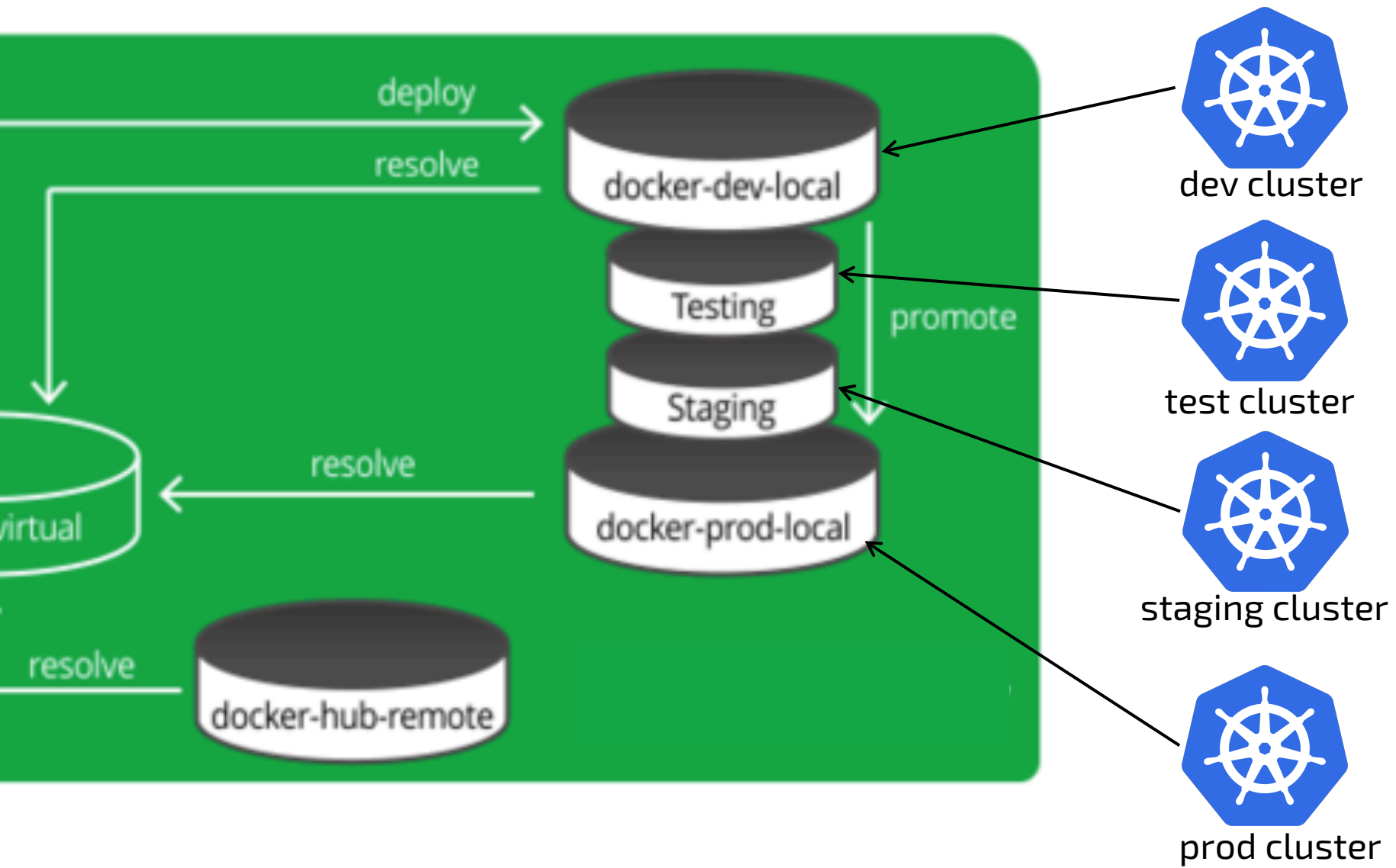


@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>





@jbaruch

@DockerAtlanta

<http://jfrog.com/shownotes>



**REPOSITORY (DOCKER):** Top level directory in a registry

**REPOSITORY (THE REST OF THE WORLD):** A registry



# WIN-WIN-WIN

- SINGLE POINT OF ACCESS TO MULTIPLE REGISTRIES WHEN NEEDED
- COMPLETELY ISOLATED ENVIRONMENTS
- IMMEDIATE AND FREE PROMOTIONS





**BUT I LOVE MY LATEST!**

Application

Dashboard

JFrog Container Registry

Packages

Builds

Artifacts

Distribution

Pipelines

Security & Compliance

JFrog Container Registry license 7.3.2 rev 70302900 undefined © Copyright 2020 JFrog Ltd

### Artifact Repository Browser

Set Me Up Deploy

Tree Simple

- > docker
- > helmhub
- > artifactory-build-info
- > docker-dev-local
- > docker-local
- > docker-prod-local
  - > jfrog-docker-app
    - > 11
    - > 14
    - > 15
    - > 16
    - > 17
    - > 18
    - > 19
    - > 20
    - > 21
    - > 23
    - > 25
    - > 26
    - latest
      - manifest.json
      - sha256\_0ce121b6a2ffae7f0fc0b7c20d3a0a46f...
      - sha256\_2fbb74461eb3d3e7d2e0c730359e88c...
      - sha256\_5c0d6befd531e53b8f0d96dce9e3418...
- > Trash Can

### manifest.json

Download View Actions

General Effective Permissions Properties Followers Builds

Add: Property | Property Set

Name \* Value Add

Recursive ?

#### 9 Properties

Filter by Property Delete

Property	Value(s)
build.name	demo_kubecon
build.number	26
build.timestamp	1575339511425
docker.manifest	latest
docker.manifest.digest	sha256:d002e5eafe3c4e2e1d69830342f7cf1d2150e3735bd17...
docker.manifest.type	application/vnd.docker.distribution.manifest.v2+json
docker.refersTo	26
docker.repoName	jfrog-docker-app

@jbaruch

@DockerAtlanta


http://jfrog.com/shownotes



# WIN-WIN

- SIMPLICITY OF LATEST
- ALWAYS KNOW WHAT IT REALLY MEANS
- AS LONG AS YOU PROMOTED IMMUTABLE ARTIFACT





But what about the rest  
of the dependencies?

- Application
- Dashboard
- JFrog Container Registry
- Packages
- Builds
- Artifacts
- Distribution
- Pipelines
- Security & Compliance

Packages > library/ubuntu



library/ubuntu

Latest version: sha256\_\_2fefff9eeca4e736f9f8e57813a97fe930554f474f7795ffa5a9261adeaaf44 07-11-19 08:50:20 -0800

Scanned Xray

2 Versions

5 Downloads

Versions

View By: List Graph

Filter

Version	Repositories	Digest	Modified	Downloads	Xray Status
sha256__134c7fe821b9d3594...	1   docker-remote-cache	134c7fe821b9d359490...	07-11-19 07:38:29 -0800	2	Requires an Xray license
sha256__2fefff9eeca4e736f9f...	1   docker-remote-cache	2fefff9eeca4e736f9f8e...	07-11-19 08:50:20 -0800	3	Requires an Xray license

@jbaruch

@DockerAtlanta

http://jfrog.com/shownotes



Application

Dashboard

JFrog Container Registry

Packages

Builds

Artifacts

Distribution

Pipelines

Security & Compliance

Artifact Repository Browser

Tree Simple



- > docker
- > helmhub
- > artifactory-build-info
- > docker-dev-local
- > docker-local
- > docker-prod-local
- ▼ generic-local
  - ▼ java
    - > jdk-8u91-linux-x64.tar.gz
  - ▼ org/jfrog/example/gradle/webservice/1.1.2
    - > webservice-1.1.2.war
  - ▼ tomcat
    - > apache-tomcat-8.tar.gz
- > helm-local



# OWN YOUR DEPENDENCIES

- YOUR BASE IMAGE
- YOUR INFRA
- YOUR APPLICATION FILES



# CONCLUSIONS

- BUILD ONLY ONCE
- SEPARATE ENVIRONMENTS
- PROMOTE WHAT YOU'VE BUILT
- OWN YOUR DEPENDENCIES



# Q&A AND LINKS

- @jbaruch
- @DockAtlanta
- <http://jfrog.com/shownotes>

