

BEST PRACTICES IN IMPLEMENTING CONTAINER IMAGE PROMOTION PIPELINES

コンテナイメージのプロモーションパイプライン
ベストプラクティス

A still from the movie "Dr. Strangelove or: How to Succeed in Business Without Really Trying". Dr. Strangelove, played by Peter Sellers, is shown from the chest up, wearing a grey military uniform. He has a shocked or manic expression on his face. Behind his head, a large black nuclear missile is positioned horizontally. He is making peace signs with both hands. The background is a dimly lit control room with various instruments and a red emergency button visible on the left.

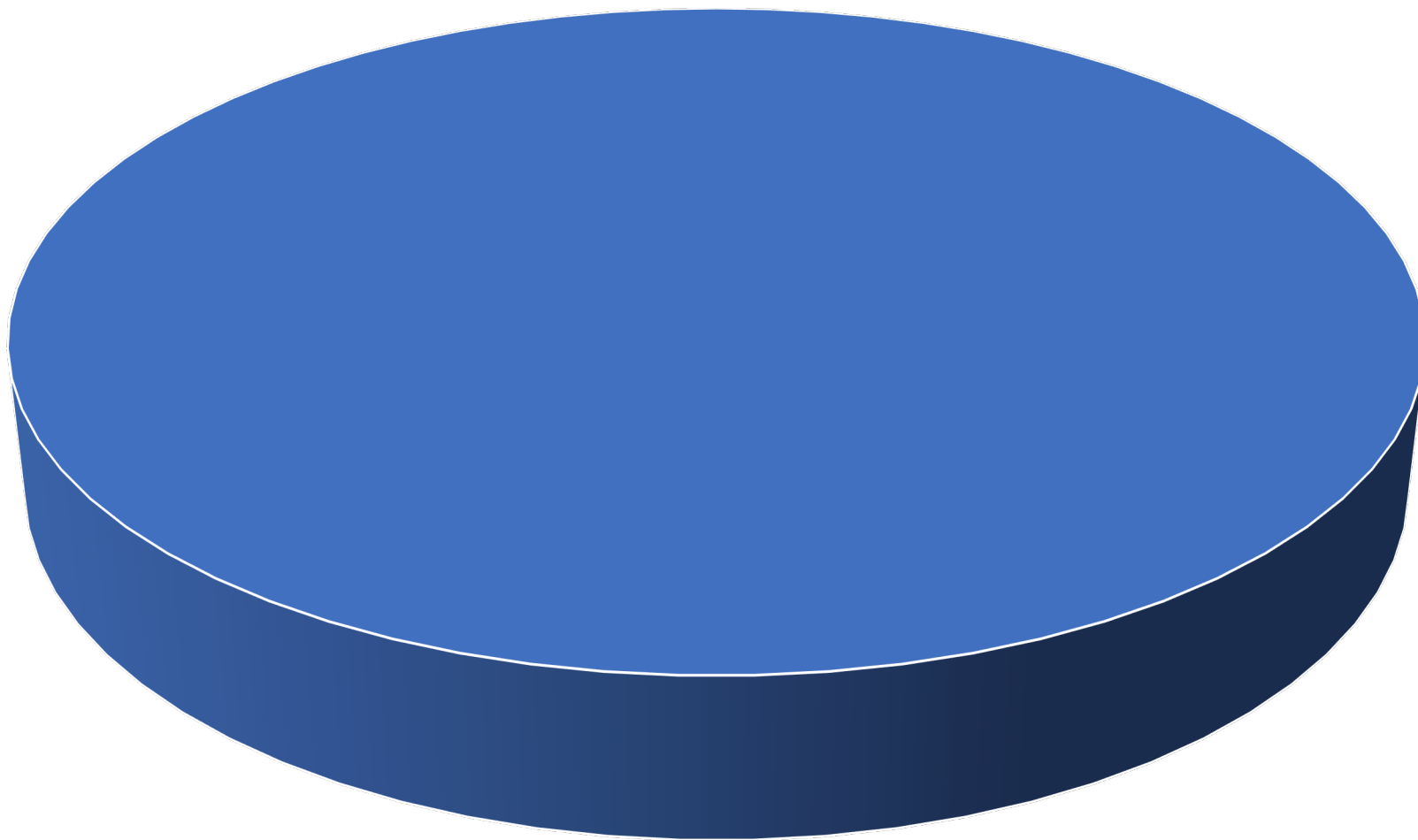
CONTAINERS

POLL TIME

会場のお客様に聞く

Dockerをご存知ですか？

Familiar with Docker*



*Among Docker meetup attendees

*Docker Meetupの参加者の結果

気に入ってるソフトウェア

よく知ってるソフトウェア

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



気に入ってるけど...

信用してはいない！

BUT I DON'T TRUST YOU

バウ・サドガスキー

BARUCH SADOGURSKY

CHIEF STICKER OFFICER

(ALSO 🎩 OF DEVELOPER ADVOCACY)



JBARUCH@JFROG.COM

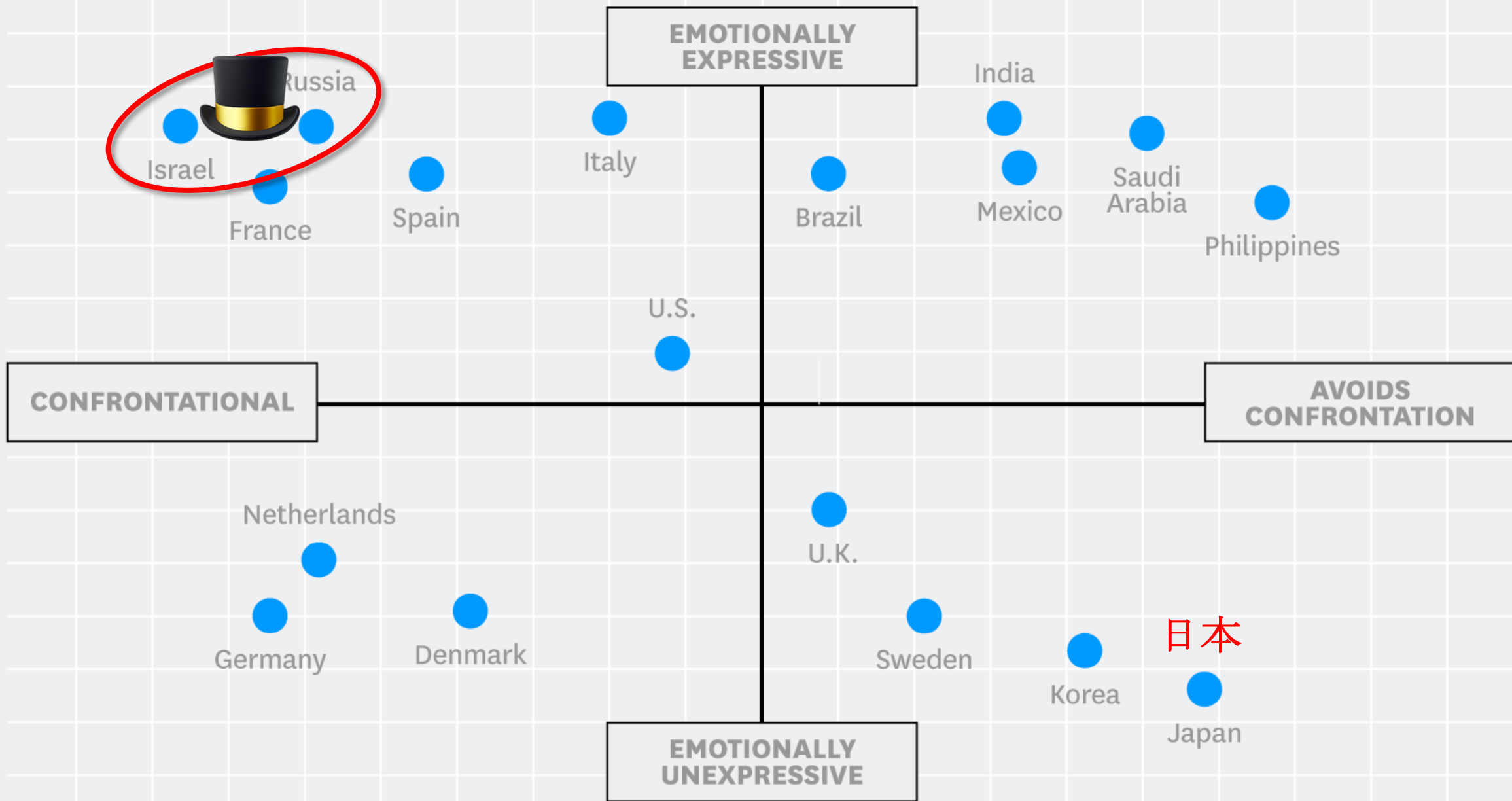


@JBARUCH



+1(408)890-9281





SHOWNOTES

➤ <http://jfrog.com/shownotes>

➤ Slides

➤ Video

➤ Links

➤ Comments, Ratings

➤ Raffle

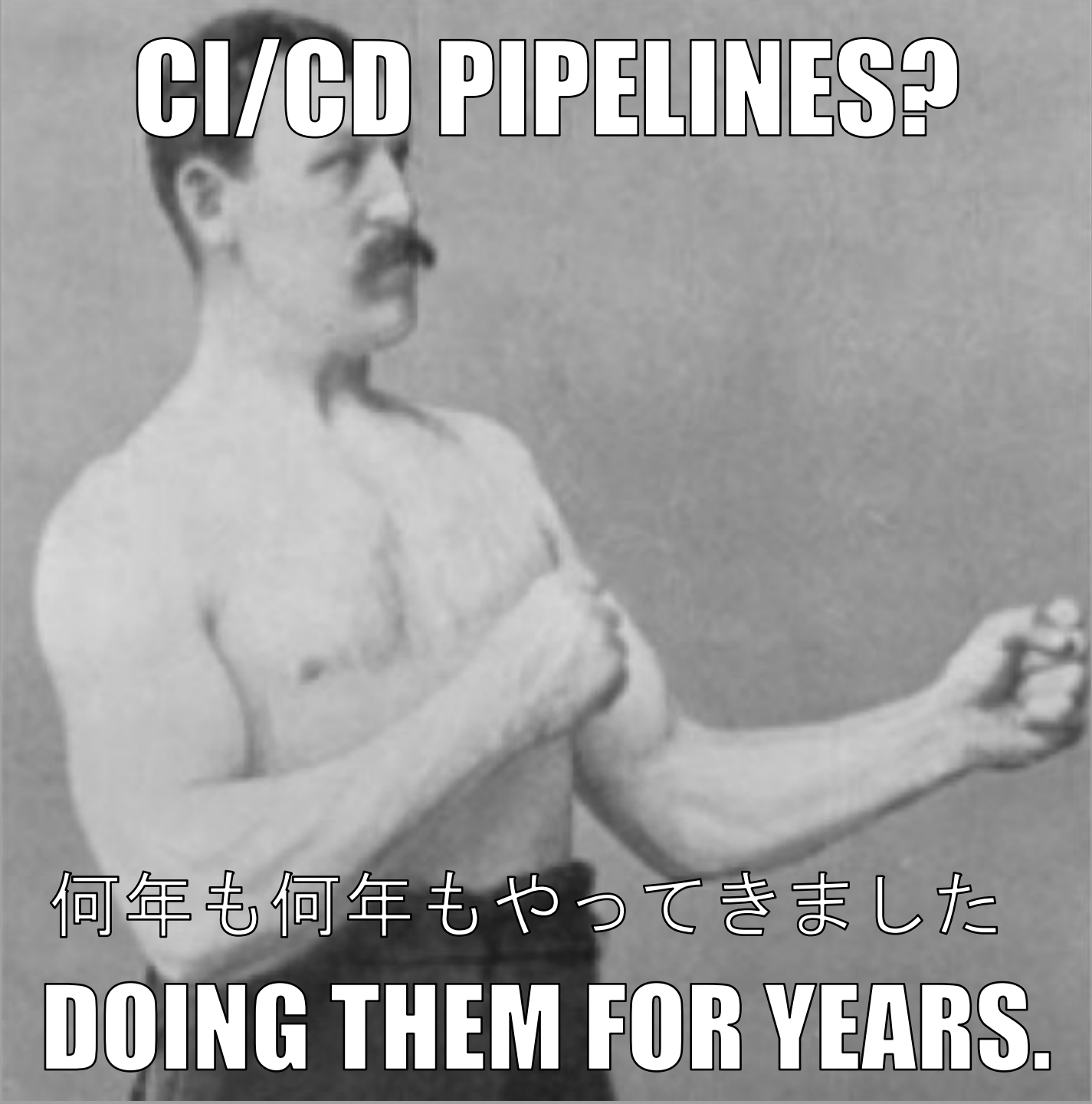
A woman with long dark hair, wearing a white lab coat, is smiling and pointing her right index finger towards the viewer. The background features a world map with North America highlighted in yellow and green, and a chalkboard on the left with the words 'ASIA' and 'AMERICA' written on it.

DO WE HAVE AN EXISTING PATTERN?

お約束のパターンってある？

その通りにやらなきゃ
ダメ？

DO WE NEED TO ADAPT IT?



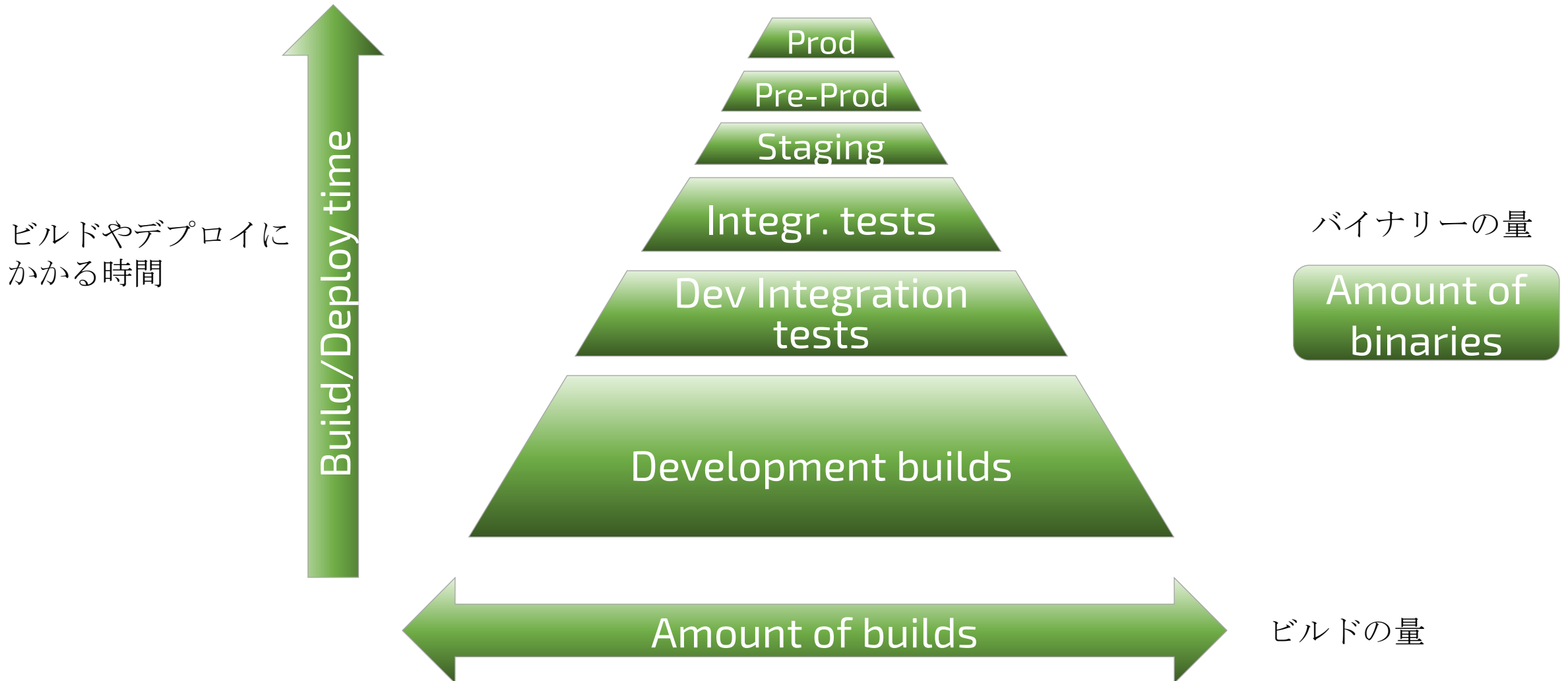
CI/CD PIPELINES?

何年も何年もやってきました

DOING THEM FOR YEARS.

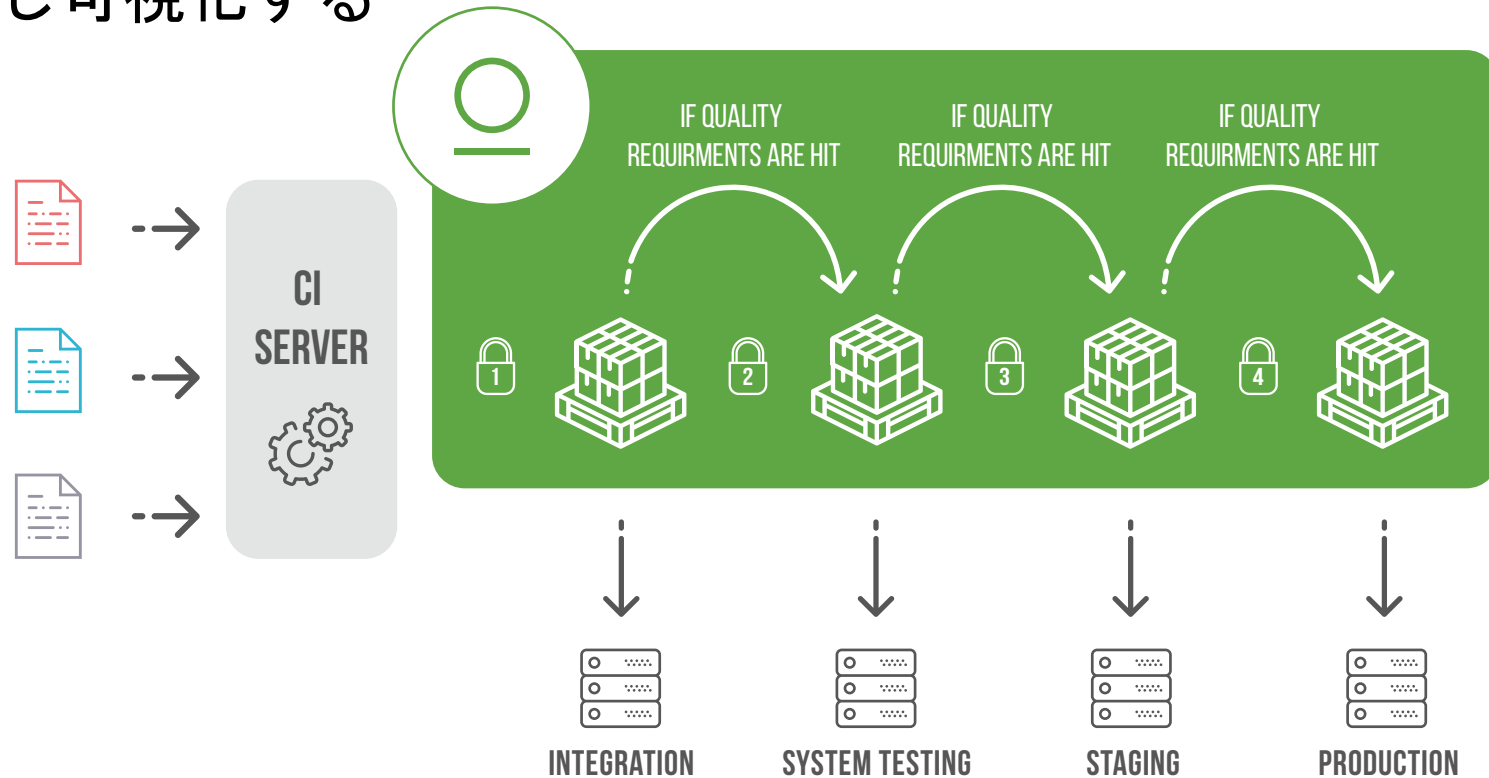
THE PROMOTION PYRAMID

プロモーションのピラミッド



PIPELINE: QUALITY GATES AND VISIBILITY

品質を管理し可視化する



\$docker build

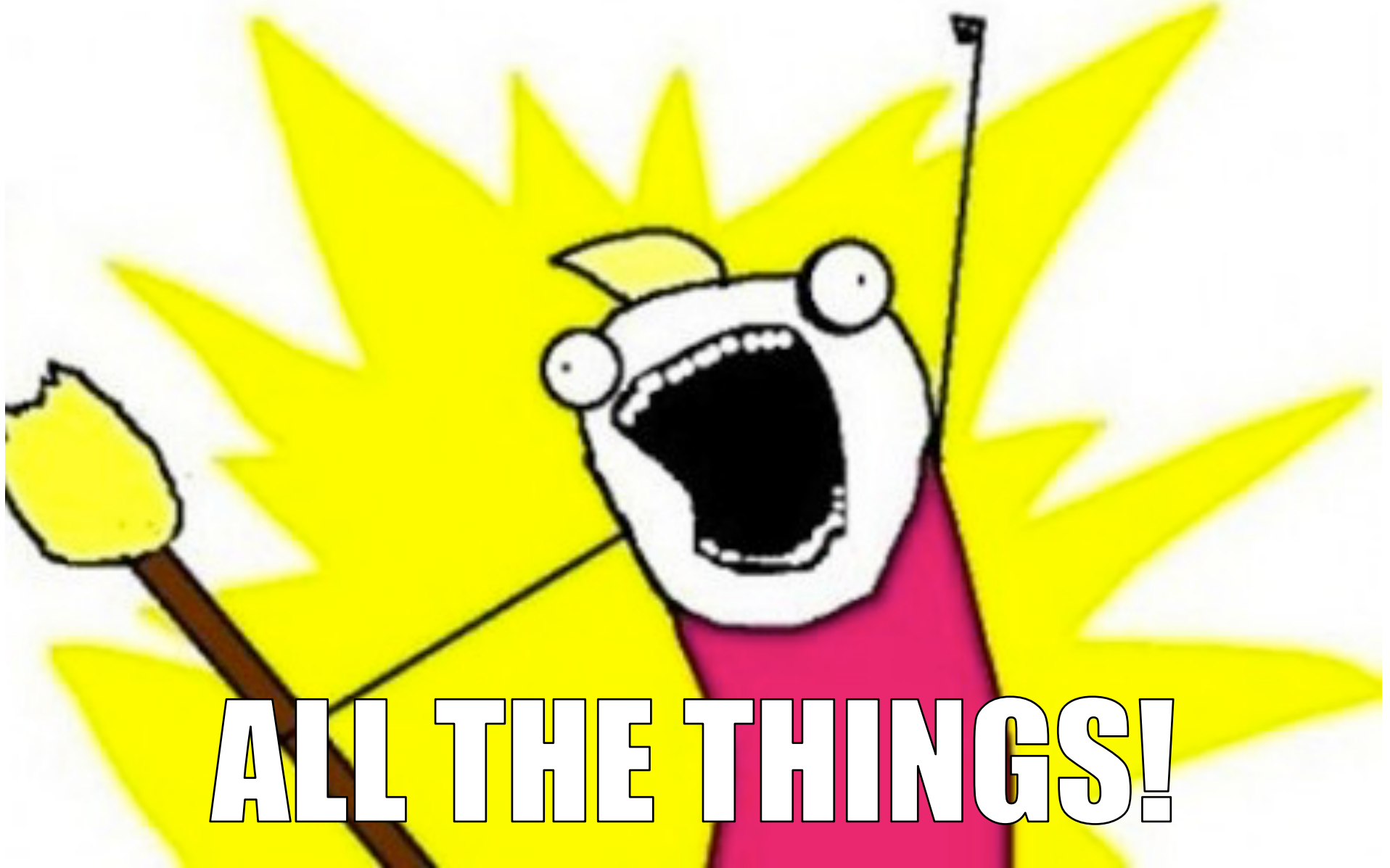


@jbaruch

#DevelopersSummit

<http://jfrog.com/shownotes>

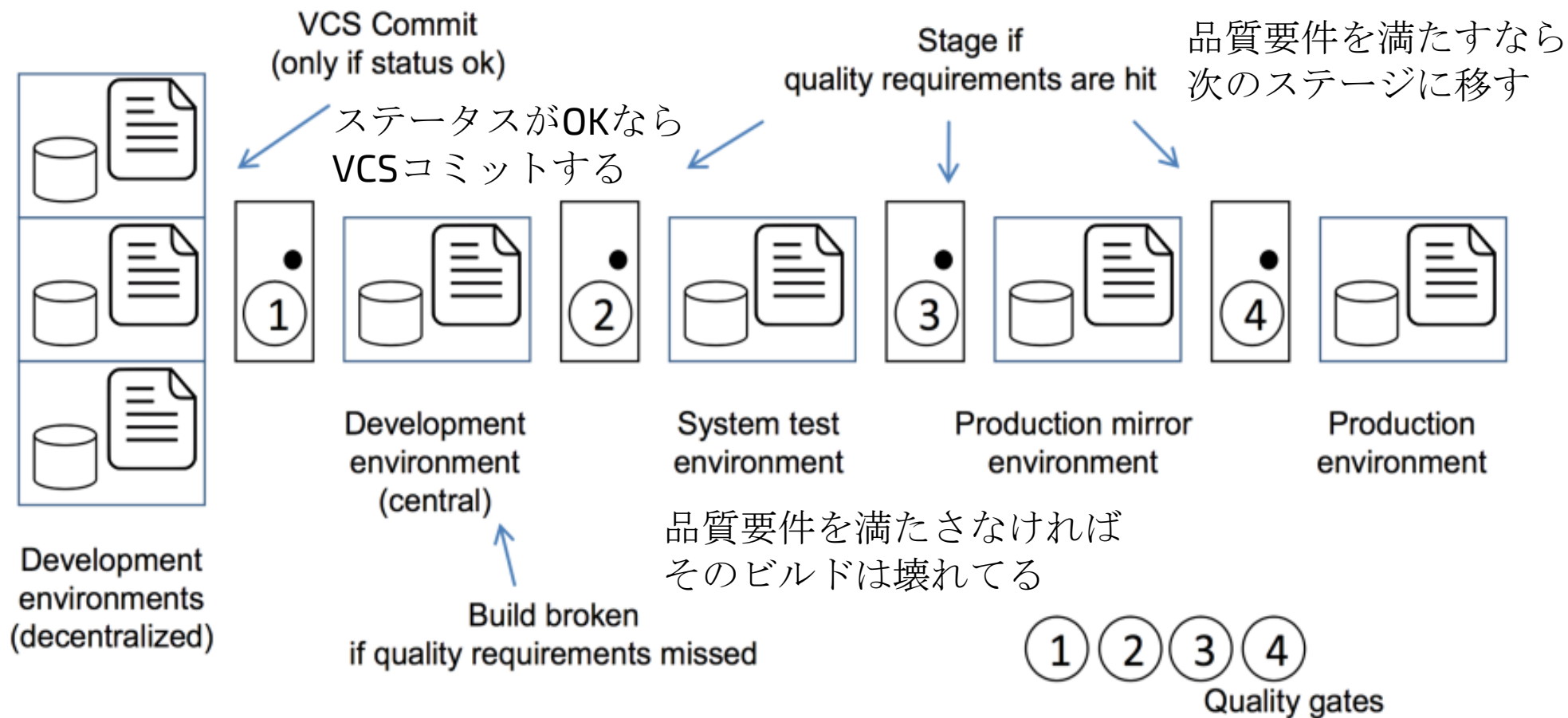
DOCKER BUILD



ALL THE THINGS!

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

全部の環境でdocker buildしてしまえ！



FAST AND CHEAP BUILDS

手早くても手を抜いて作ると...

結構めんどくさい事になる

NOT ALWAYS THE WAY TO GO

THAT'S WHY. なぜなら...

FROM ubuntu  Latest version

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

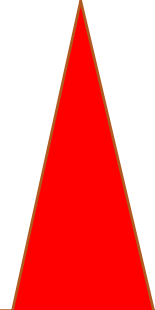
RUN apt-get install -y nodejs

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. なぜなら...

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. なぜなら...

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. なぜなら...

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?

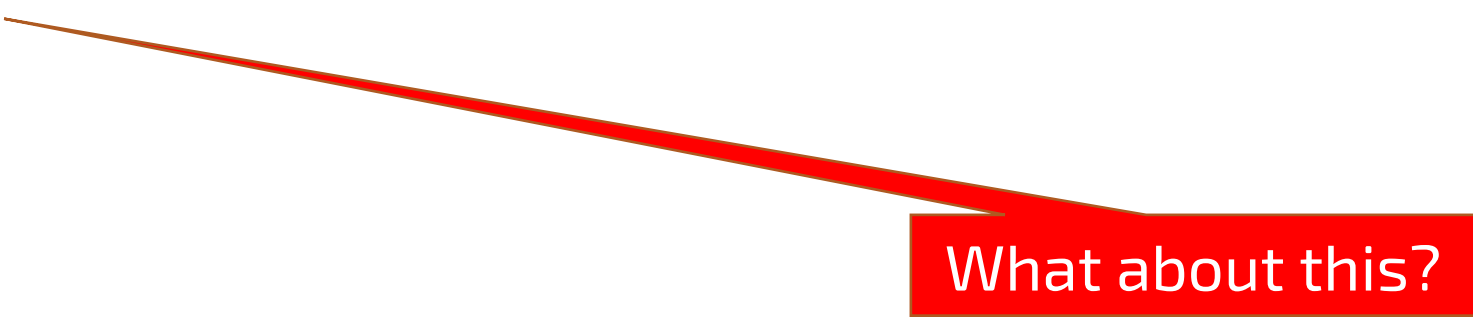
ADD app.js /var/www/app.js

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

THAT'S WHY. なぜなら...

FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98

RUN mvn clean install



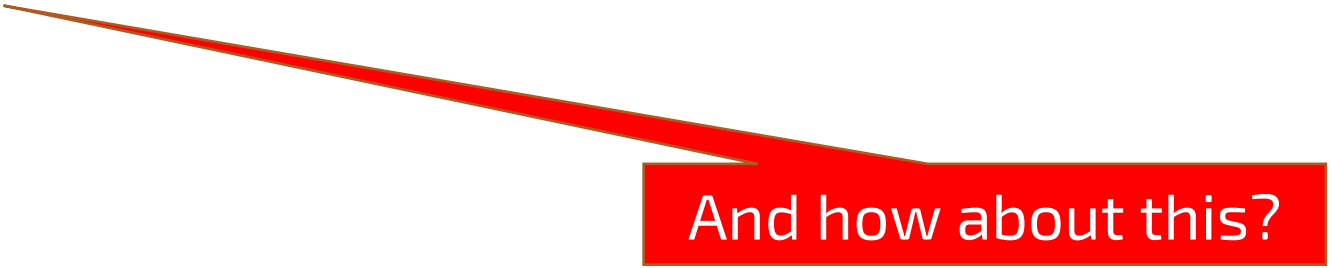
What about this?

CMD "java -jar Main.class"

THAT'S WHY. なぜなら...

FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98

RUN download_random_sh*t_from_the_internet.sh

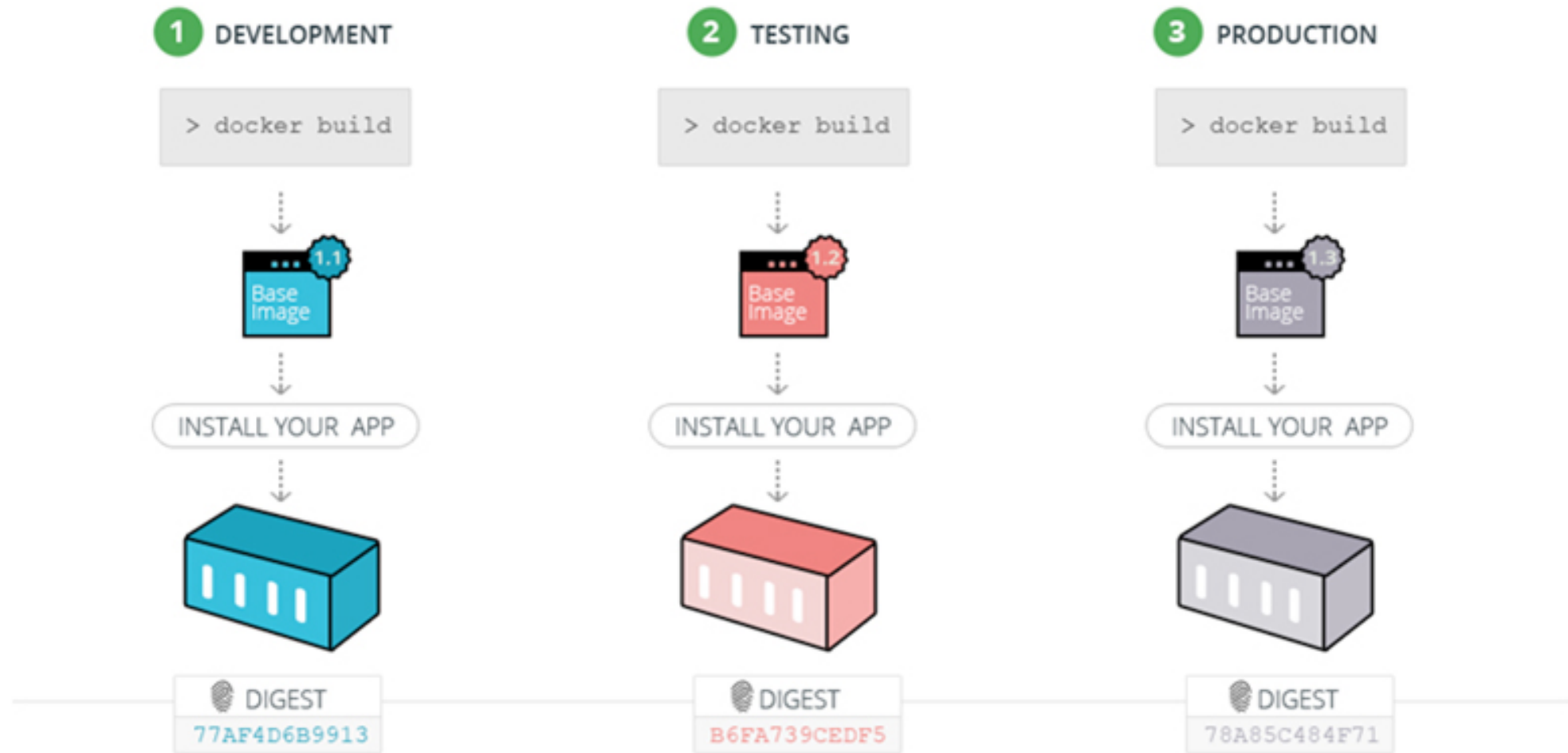


And how about this?

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

THAT'S WHY YOU DON'T TRUST DOCKER

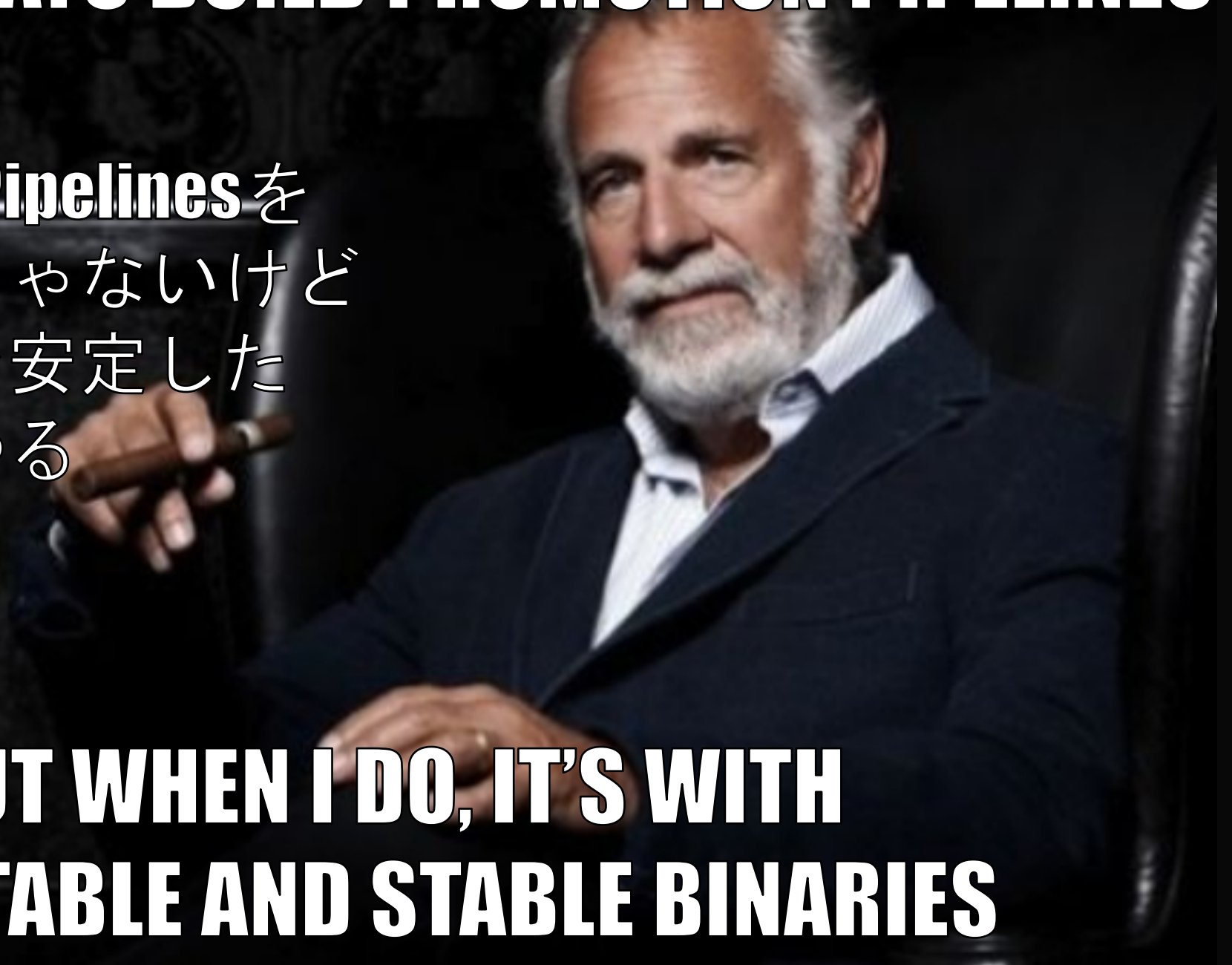
だからDockerはあてにならない



I DON'T ALWAYS BUILD PROMOTION PIPELINES

常に**Promotion Pipelines**を
ビルドする訳じゃないけど
やる時は不変で安定した
バイナリーでやる

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



1 DEVELOPMENT

```
> docker build
```



INSTALL YOUR APP



DIGEST

77AF4D6B9913

2 TESTING

```
> build once
```

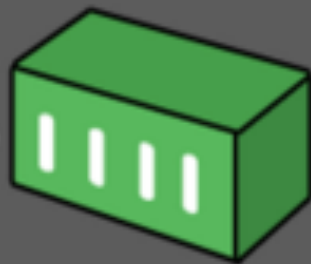
```
> automatically promote the same image  
through the pipeline to production
```



DIGEST

77AF4D6B9913

3 PRODUCTION



DIGEST

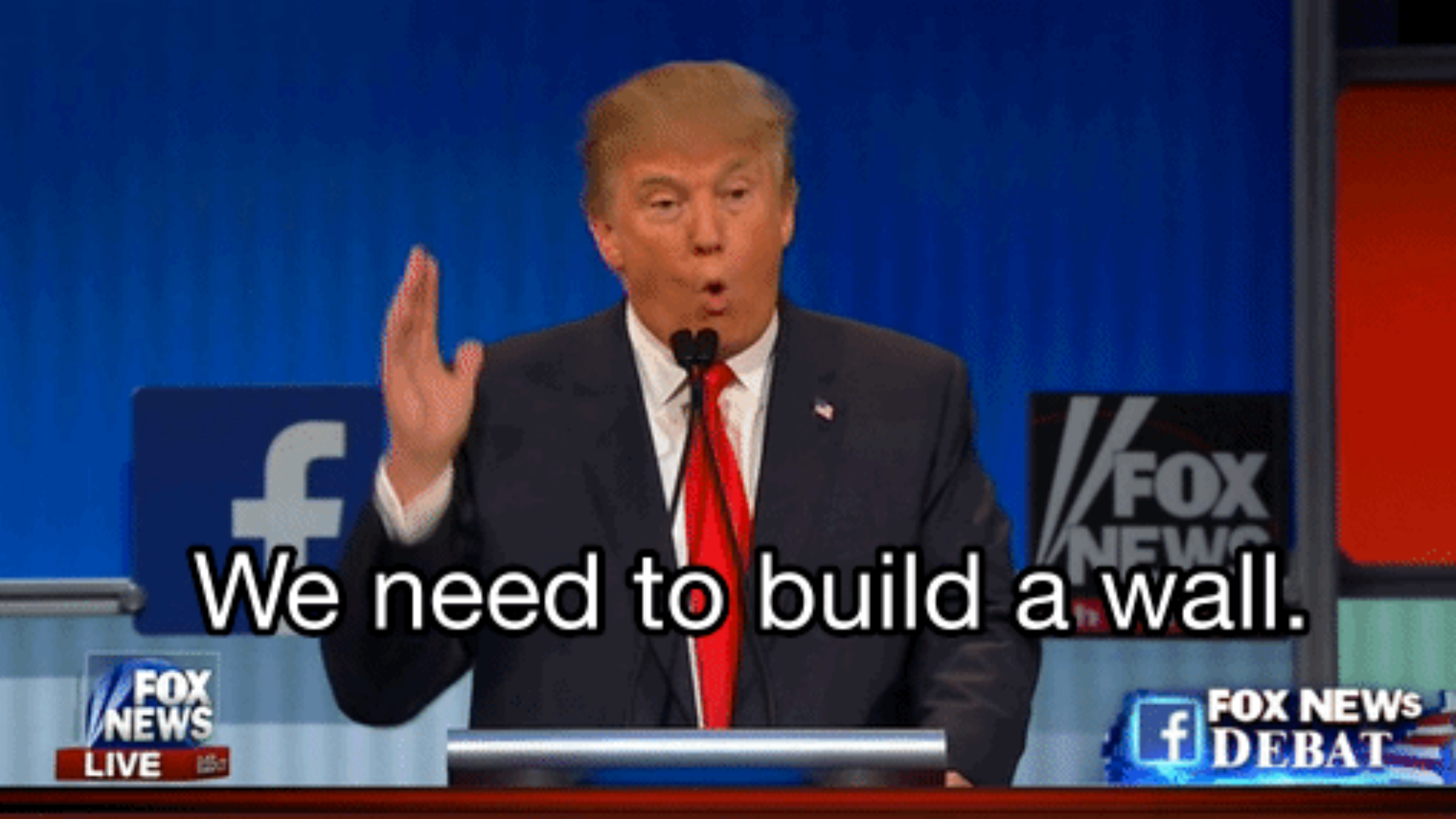
77AF4D6B9913

PROMOTE

PROMOTE

WHAT'S UP WITH THE GATES?!

それじゃGateはどうなるの？

A medium shot of Donald Trump speaking at a podium. He is wearing a dark suit, a white shirt, and a red tie. His right hand is raised with the palm facing forward. The background is a blue wall with a large white Facebook 'f' logo on the left and a 'FOX NEWS' logo on the right. A large white text overlay with a black outline reads 'We need to build a wall.'

We need to build a wall.

**FOX
NEWS**

LIVE

11:57a



**FOX NEWS
DEBATE**

WHAT'S UP WITH THE GATES?!

いやいやコイツじゃなくて

- 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!!!

プロモートされず、テストもされていないdev imageは本番環境に移行させない!!

LET'S BUILD ROCK-SOLID PIPELINE!

鉄板のPipelineを作ってみようぜ



HOW DO I SEPARATE DEV FROM PROD?!

開発環境と本番環境を
どう切り離すのか？

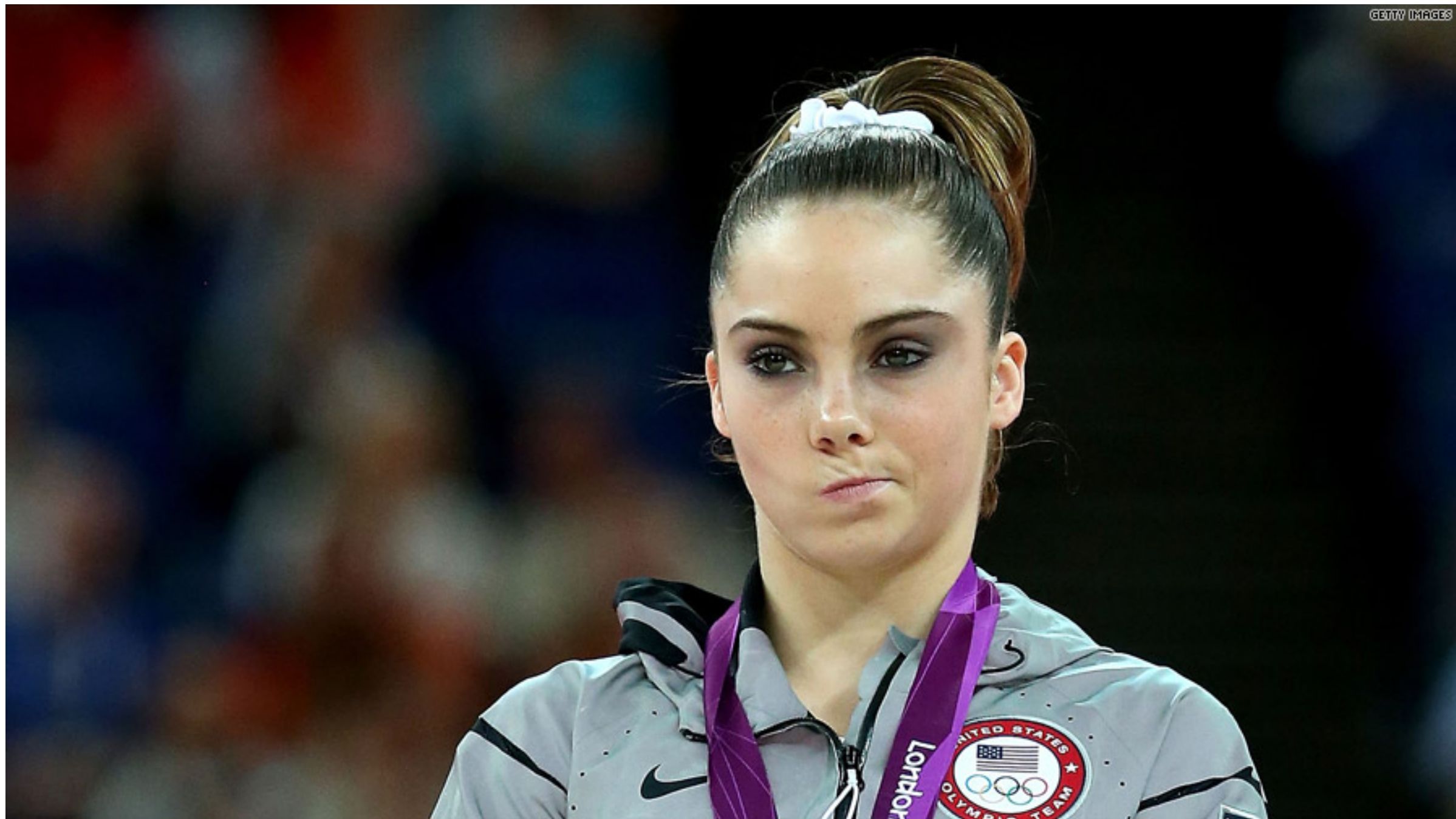
OPTION 1: METADATA TAGS

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

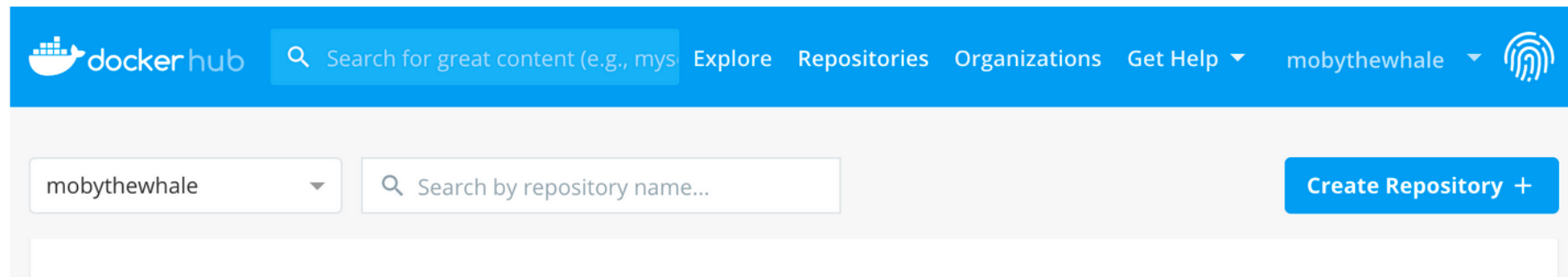
Estimated reading time: 5 minutes

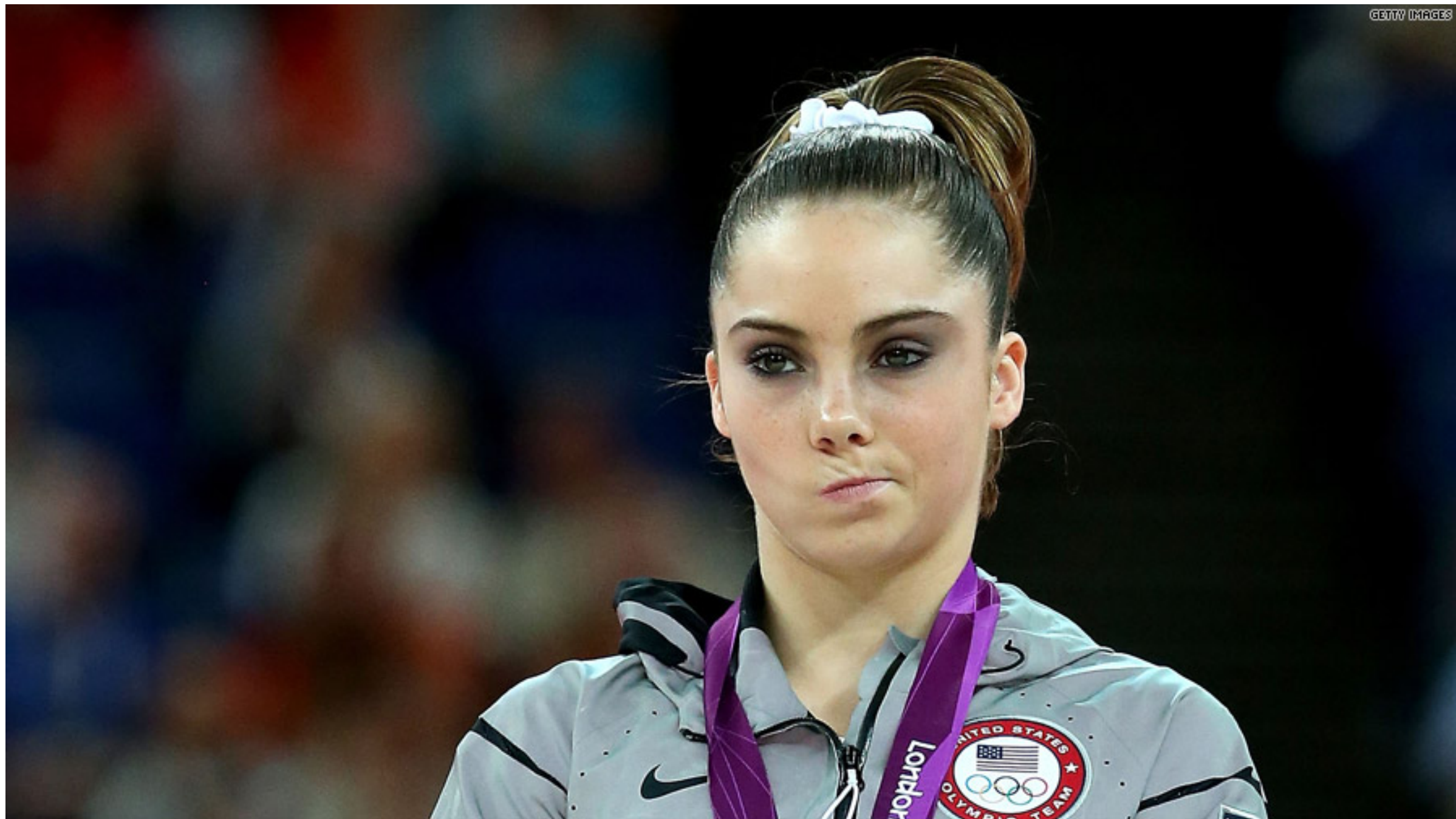
Docker Hub repositories allow you share container images with your team, customers, or the Docker community at large.

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

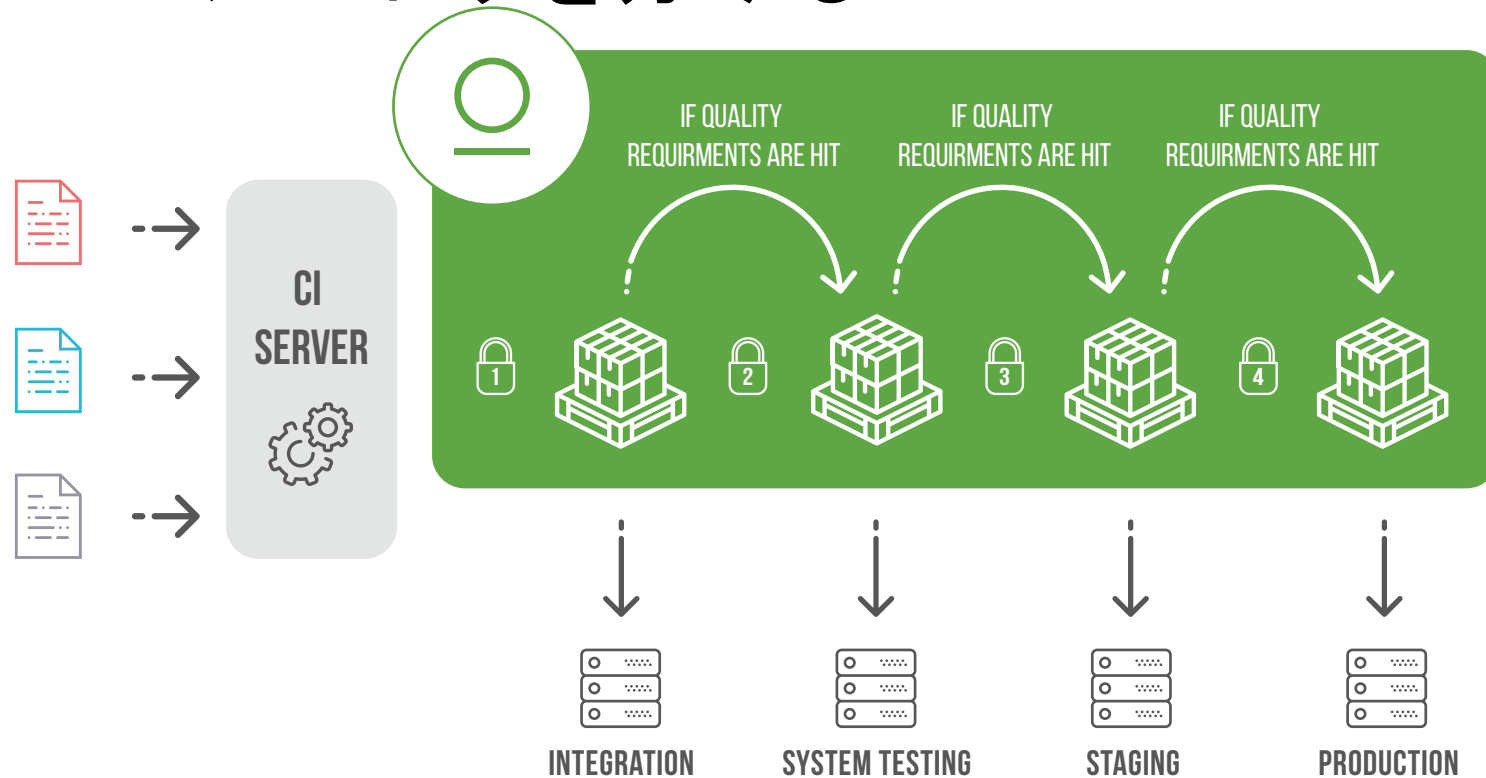
To create a repository, sign into Docker Hub, click on **Repositories** then **Create Repository**:





SEPARATE REGISTRIES PER ENVIRONMENT

環境ごとにレジストリを分ける



REQUIREMENTS 大切なこと...

- REGISTRY PER ENVIRONMENT
- EASY ACCESS TO ALL THE IMAGES
- FAST PROMOTION
- KEEP WORKING WITH LATEST AND KNOWING WHAT IT IS

常に最新のモノを十分理解して利用する

THE ANATOMY OF DOCKER TAG

Docker Tagの構造

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?!

ちょっと待ってよ！
ホストごとに1つ以上の
レジストリって持てる？



HOW CAN WE SUPPORT THIS?

こうゆうのをどうサポートする？

`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`

PANIC! パニック



@jbaruch

#DevelopersSummit

<http://jfrog.com/shownotes>

VIRTUAL HOSTS/PORTS TO THE RESCUE

`docker tag host:port/busybox`

Registry host

Tag name

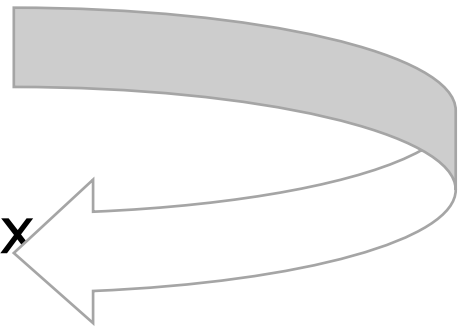
`https://host:port/v2/busybox`

`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;  
    ...  
}
```

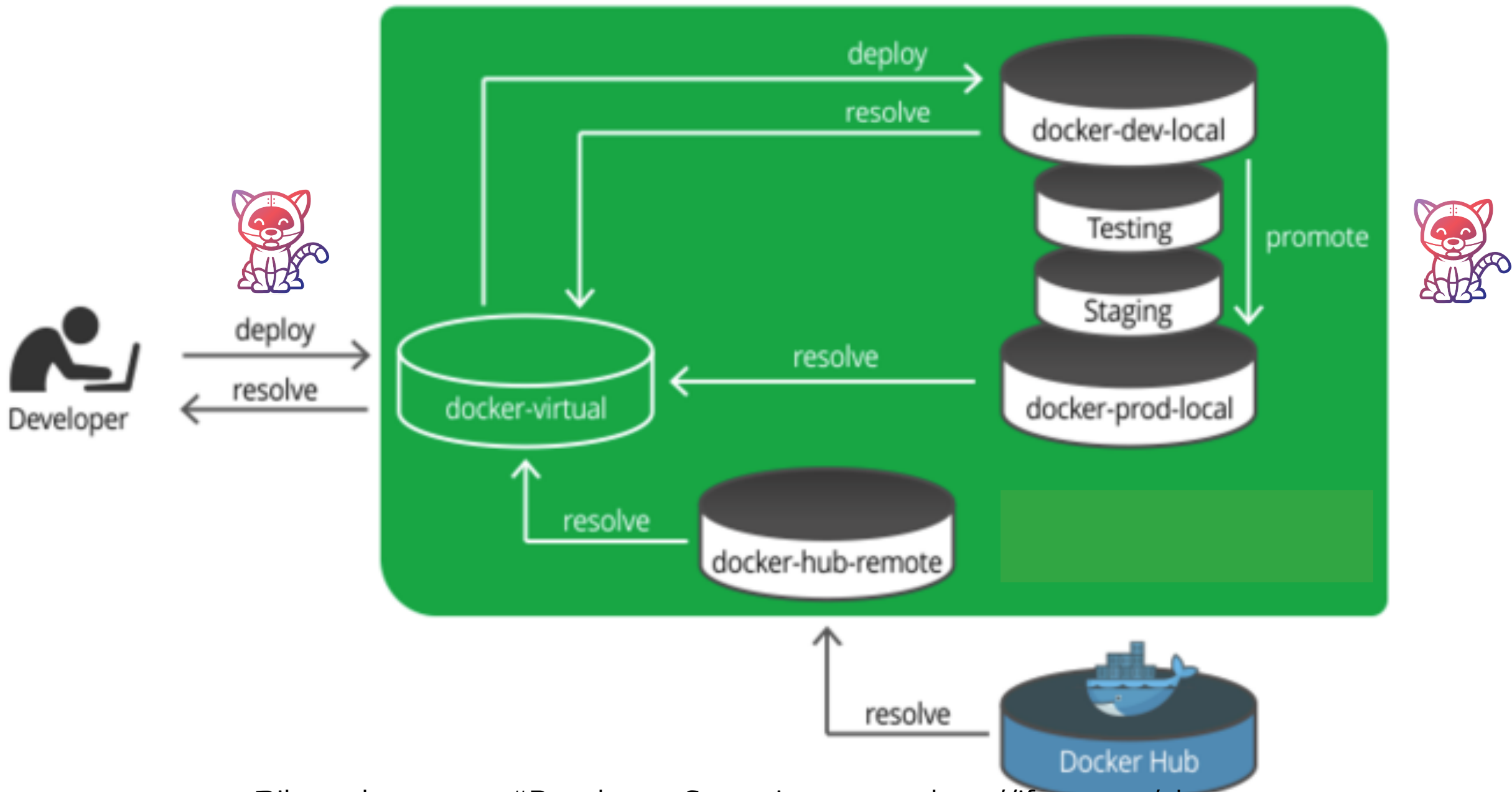
BUT THEN YOU REALIZE...

ここでハタと気づく..

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

ちょっと待ってよ！
つまりステップ毎にプルして、
タグ付け直してプッシュするっ
てこと？ ありえないでちゅ...

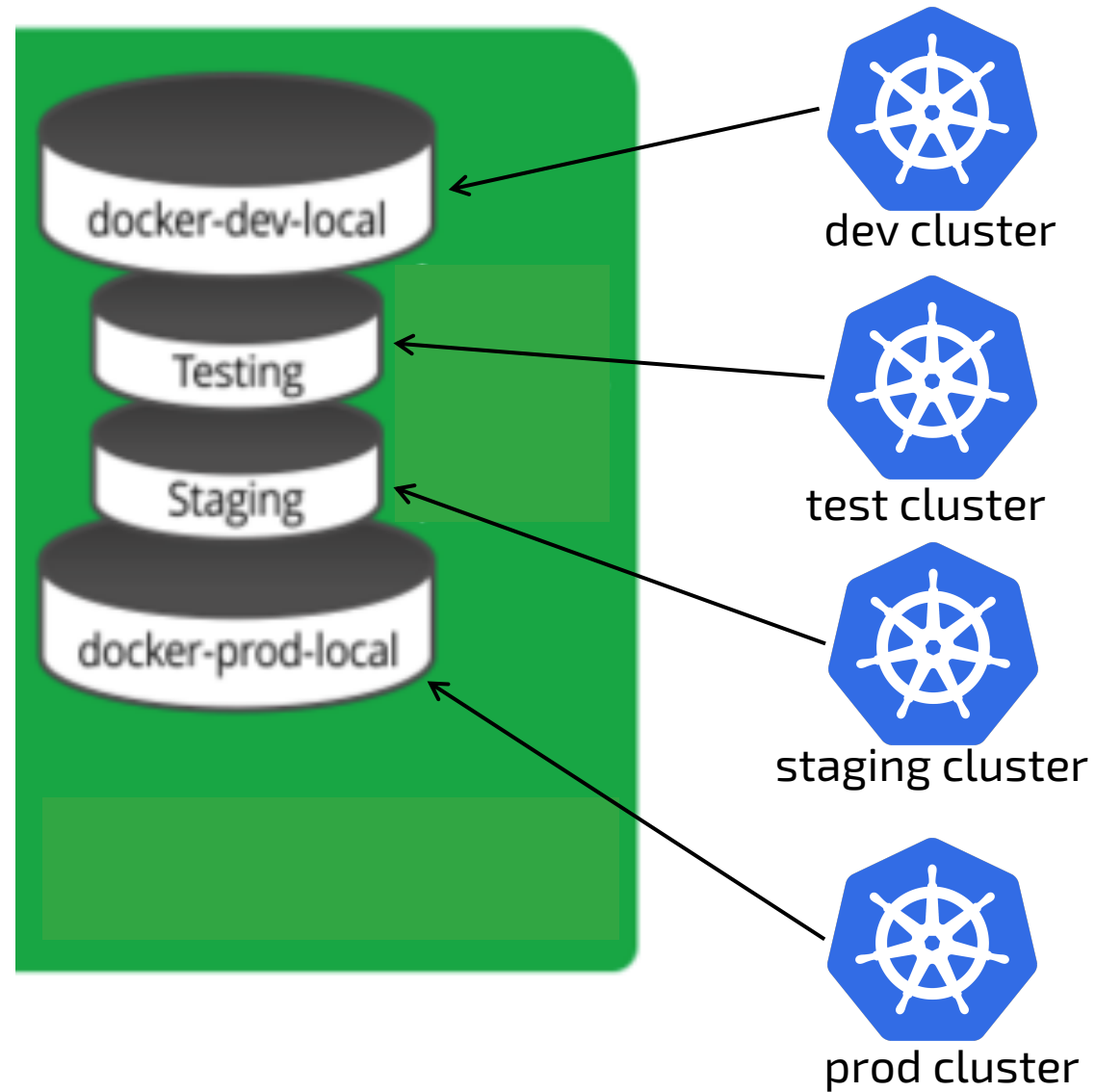




@jbaruch

#DevelopersSummit

<http://jfrog.com/shownotes>



REPOSITORY (DOCKER): Top level directory in a registry

REPOSITORY (THE REST OF THE WORLD): A registry



でも私、最新のヤツが好きなの
BUT I LOVE MY LATEST!



Artifact Repository Browser

Set Me Up Deploy

Tree Simple

- docker-prod-local
 - jfrog-docker-app
 - 11
 - 14
 - 15
 - 16
 - 17
 - 18
 - 19
 - 20
 - 21
 - 23
 - 25
 - 26
 - latest
 - manifest.json
 - sha256_0ce121b6a2ffae7f0fc0b7c20d3a0a46f62762dd7c1f14e
 - sha256_2fbb74461eb3d3e7d2e0c730359e88c1fd89e6b2d69d
 - sha256_5c0d6befd531e53b8f0d96dce9e3418f9b9531f903a42
 - sha256_6cb6319b1c7f389d3b74928b1a346b7573d4351a69da
 - sha256_6ef14aff1139e1065ec0928ae1c07f2cff8c2b35e760f4b
 - sha256_85a73a180fcd5ded2e134a9e9dca93c09068d1b4d529
 - sha256_90e9442097e7396b610beada64696692f066b6e66d2e
 - sha256_962785d3b7f9bd4442aa6df1862d8ad64e0ee1d57ec0
 - sha256_976199b2ac17369bb93597d858f5d7c4e52d6dfd3f358
 - sha256_9a0b0ce99936ce4861d44ce1f193e881e5b40b5bf1847

Trash Can

latest

manifest.json

Download View Actions

General Effective Permissions Properties Watchers Builds Xray

Add: Property | Property Set

Name * Value Add

☐ Recursive ?

9 Properties

Filter by Property

Delete

Property	Value(s)
sha256	d002e5eafe3c4e2e1d69830342f7cf1d2150e3735bd17ee13e2258e0ad99f3dc
docker.repoName	jfrog-docker-app
build.timestamp	1575339511425
docker.manifest.digest	sha256:d002e5eafe3c4e2e1d69830342f7cf1d2150e3735bd17ee13e2258e0ad9...
build.name	demo_kubecon
docker.manifest	latest
build.number	26
docker.manifest.type	application/vnd.docker.distribution.manifest.v2+json
docker.refersTo	26

@jbaruch

#DevelopersSummit

http://jfrog.com/shownotes

WIN-WIN

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

不変の配布物をプロモートする限りは

REQUIREMENTS 大切なこと...

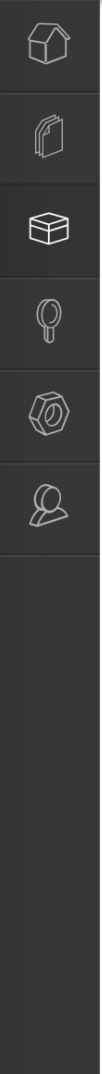
- REGISTRY PER ENVIRONMENT
- EASY ACCESS TO ALL THE IMAGES
- FAST PROMOTION
- KEEP WORKING WITH LATEST AND KNOWING WHAT IT IS

But what about the rest
of the dependencies?

でもそれ以外の依存関
係はどーするの？

依存関係は？





Package Viewer Docker

< Back to list of images

	Image Name	Downloads	Last Modified
	library/ubuntu	5	Nov 7, 2019 10:50:20 AM

2 Items

Details ☒ Xray

Tag Name	Repository	Digest	Last Modified ▾	Downloads	Xray Status
sha256:2feffff9eeca4e7...	docker-remote-cache	 2feffff9eeca4e736f9f8e57	Nov 7, 2019 10:50:20 AM	 3 Downloads	N/A
sha256:134c7fe821b9d...	docker-remote-cache	 134c7fe821b9d359490cd	(Nov 7, 2019 9:38:29 AM	 2 Downloads	N/A

@jbaruch

#DevelopersSummit

http://jfrog.com/shownotes



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
 - └─ ✓ 📁 tomcat
 - > 📄 apache-tomcat-8.tar.gz
- > 📦 helm-local
- > 📦 bintray-docker-remote

OWN YOUR DEPENDENCIES

自分でしっかり管理する

- YOUR BASE IMAGE
- YOUR INFRA
- YOUR APPLICATION FILES

自分のアプリケーションファイル

CONCLUSIONS まとめ...

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

自分の依存関係は自分で管理する

Q&A AND LINKS

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

Ayana Yokota / 横田紋奈(よこな)

デベロッパーアドボケイト at JFrog

Twitter: @ihcomega

マイブームはサウナ





を当てよう!



当選者発表は
2月14日(金) 12:50

JFrogブースで行います

