



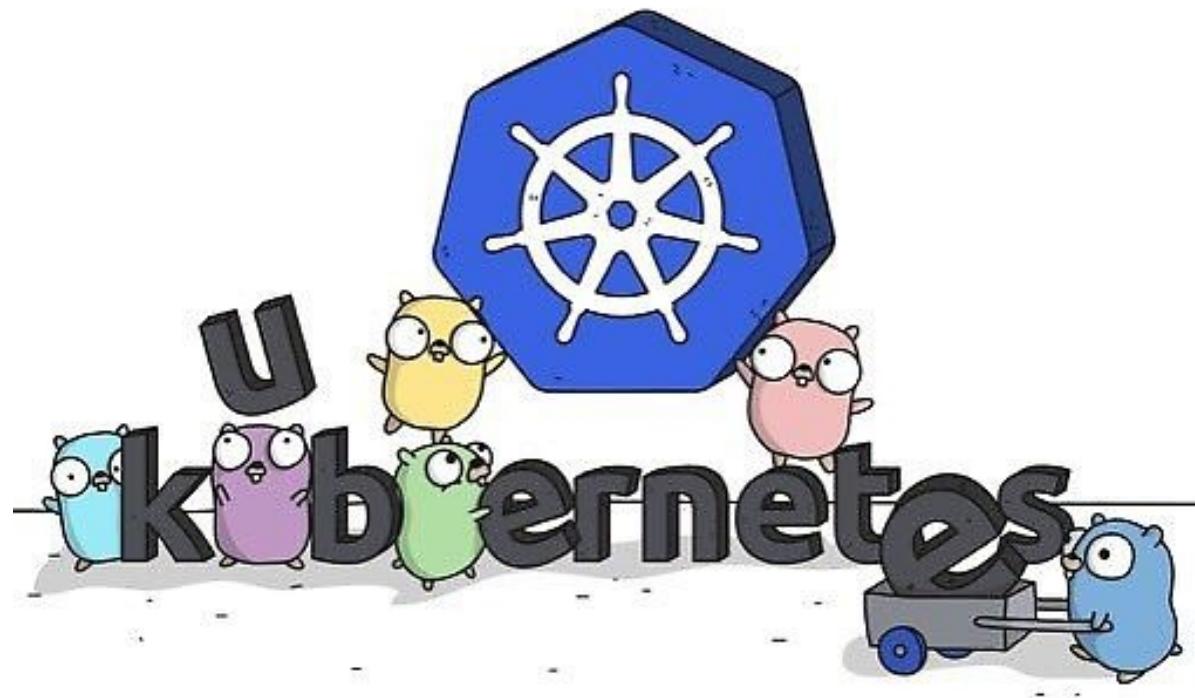
Best Practices In Implementing Container Image Promotion Pipelines



Who likes docker ?

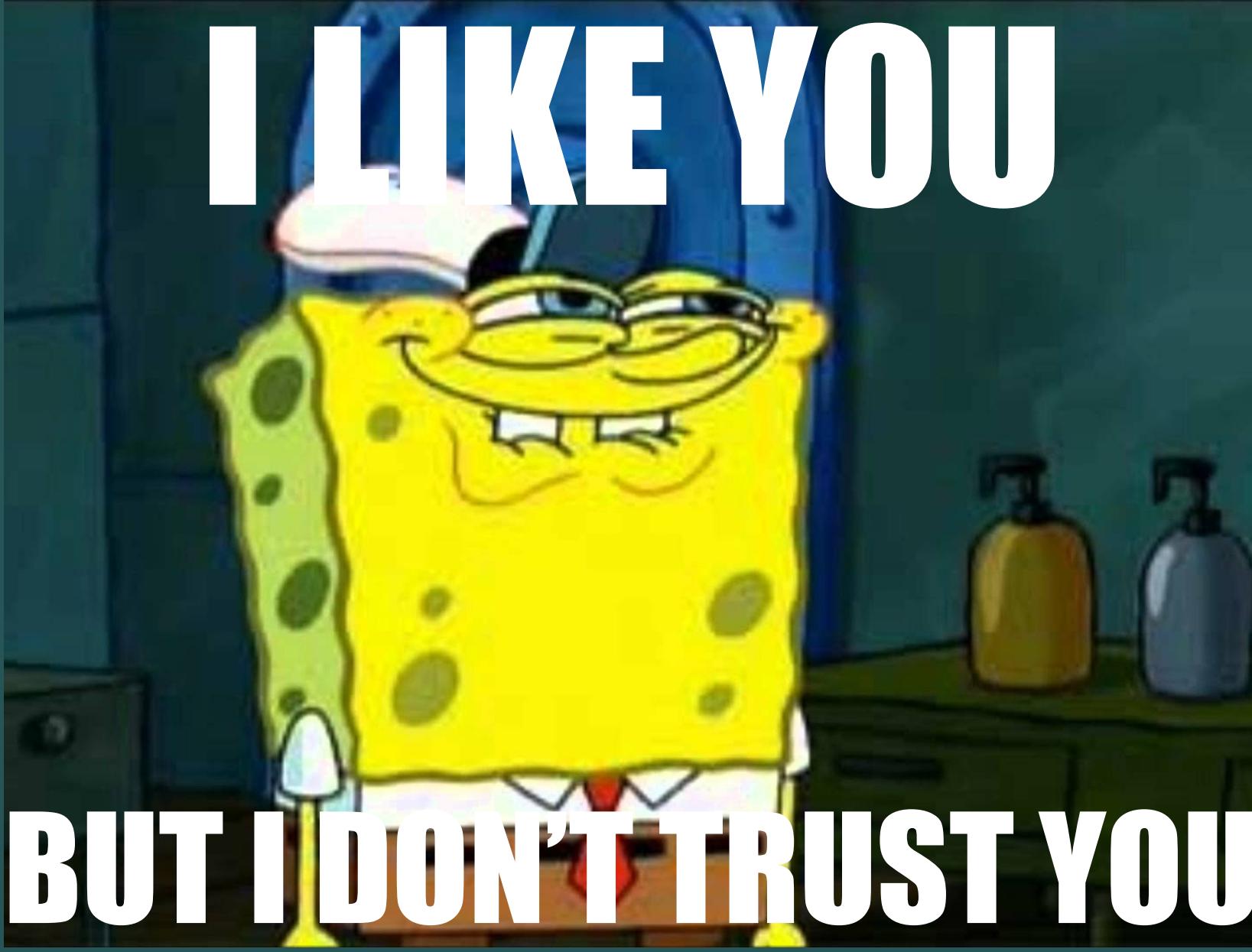


Now what about K8s?



I LIKE YOU

BUT I DON'T TRUST YOU





Batel Zohar | High Touch
Solutions Team
@BatelZohar



WIN
A JFROG SHIRT!



<https://bit.ly/JakartaK8JFrog>

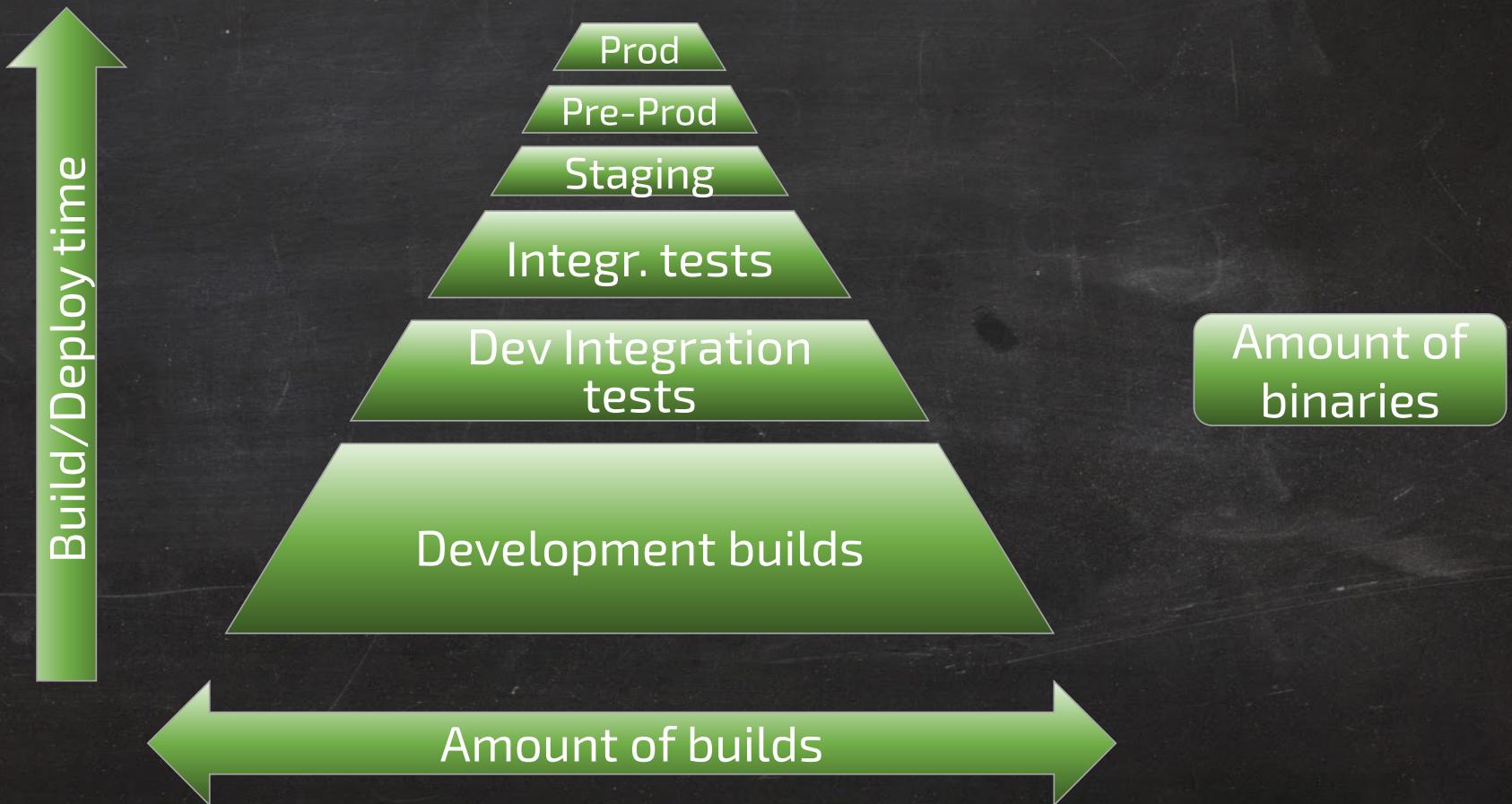
CI/CD PIPELINES?



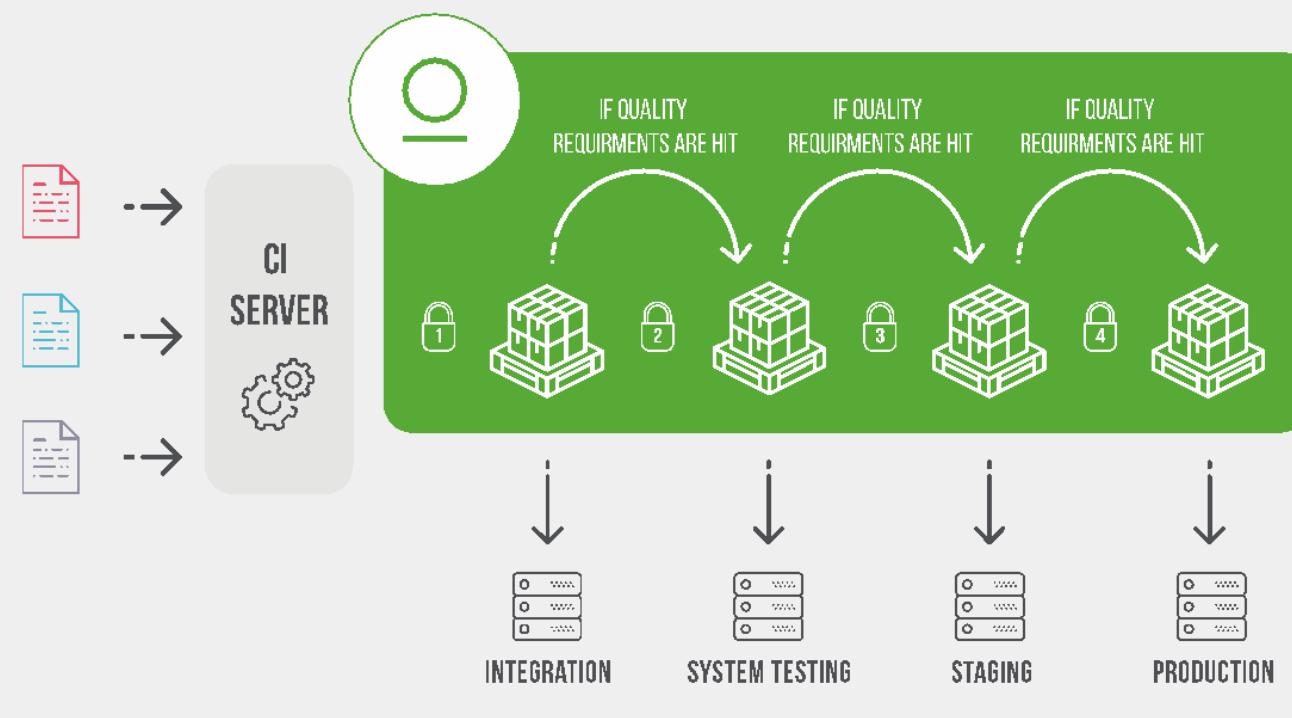
We're doing fine

DOING THEM FOR YEARS.

THE PROMOTION PYRAMID



PIPELINE: QUALITY GATES AND VISIBILITY

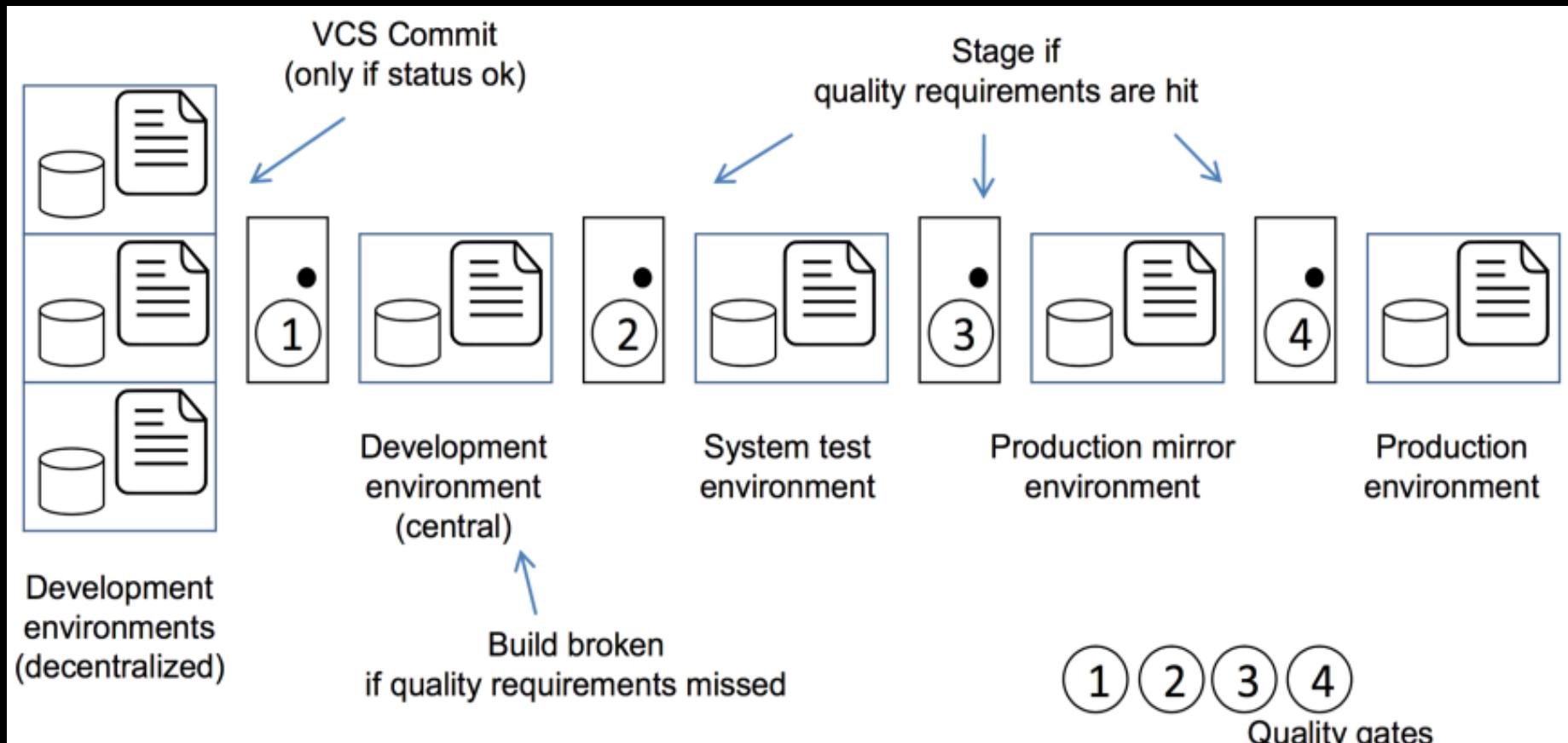


- QUALITY GATES -

WE COULD HAVE HAD IT ALL



LET'S docker build IN EVERY ENV!



EVERYTHING IS GONNA BE ALRIGHT

\$docker build



FAST AND CHEAP BUILDS

NOT ALWAYS THE WAY TO GO



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

```
FROM ubuntu
```

```
RUN apt-get install -y python-software-properties 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"]
```



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

```
FROM ubuntu
```

```
RUN apt-get install -y python-software-properties 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"]
```



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

FROM ubuntu



RUN apt-get install -y python-software-properties 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"]



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

```
FROM ubuntu
latest version ←
RUN apt-get install -y python-software-properties python
latest version →
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"]
```



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

```
FROM ubuntu
latest version
latest version
RUN apt-get install -y python-software-properties python
RUN apt-get install -y nodejs
RUN mkdir /var/www
Same here
ADD app.js /var/www/app.js
CMD ["/usr/bin/node", "/var/www/app.js"]
```



The unbearable lightness of \$ docker build

<https://github.com/kstaken/dockerfile-examples/blob/master/nodejs/Dockerfile>

```
FROM ubuntu
latest version
latest version
RUN apt-get install -y python-software-properties python
RUN apt-get install -y nodejs
RUN mkdir /var/www
Same here
ADD app.js /var/www/app.js
Same here
CMD ["/usr/bin/node", "/var/www/app.js"]
```



Let's change it a bit

```
FROM ubuntu:19.4
```



Better now?

```
RUN apt-get install -y python-software-properties 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"]
```



Let's change it a bit

Maybe now?

```
FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98
```

```
RUN apt-get install -y python-software-properties 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"]
```



Let's change it a bit

```
FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98
```

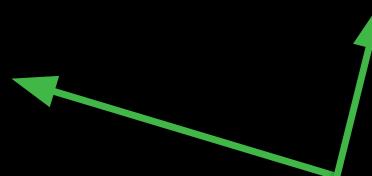
```
RUN apt-get install -y python-software-properties python
```

```
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"]
```



What about
those?

What about Maven?

```
FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98
```

```
RUN mvn clean install
```

```
CMD ["java -jar Main.class"]
```

What about
this?



What about a Unicorns container?

```
FROM ubuntu:4033353383af19ec179c01dda7f355a246c6adcafaf93c8f98
```

```
RUN download_random_unicorn_farm_from_the_internet.sh
```

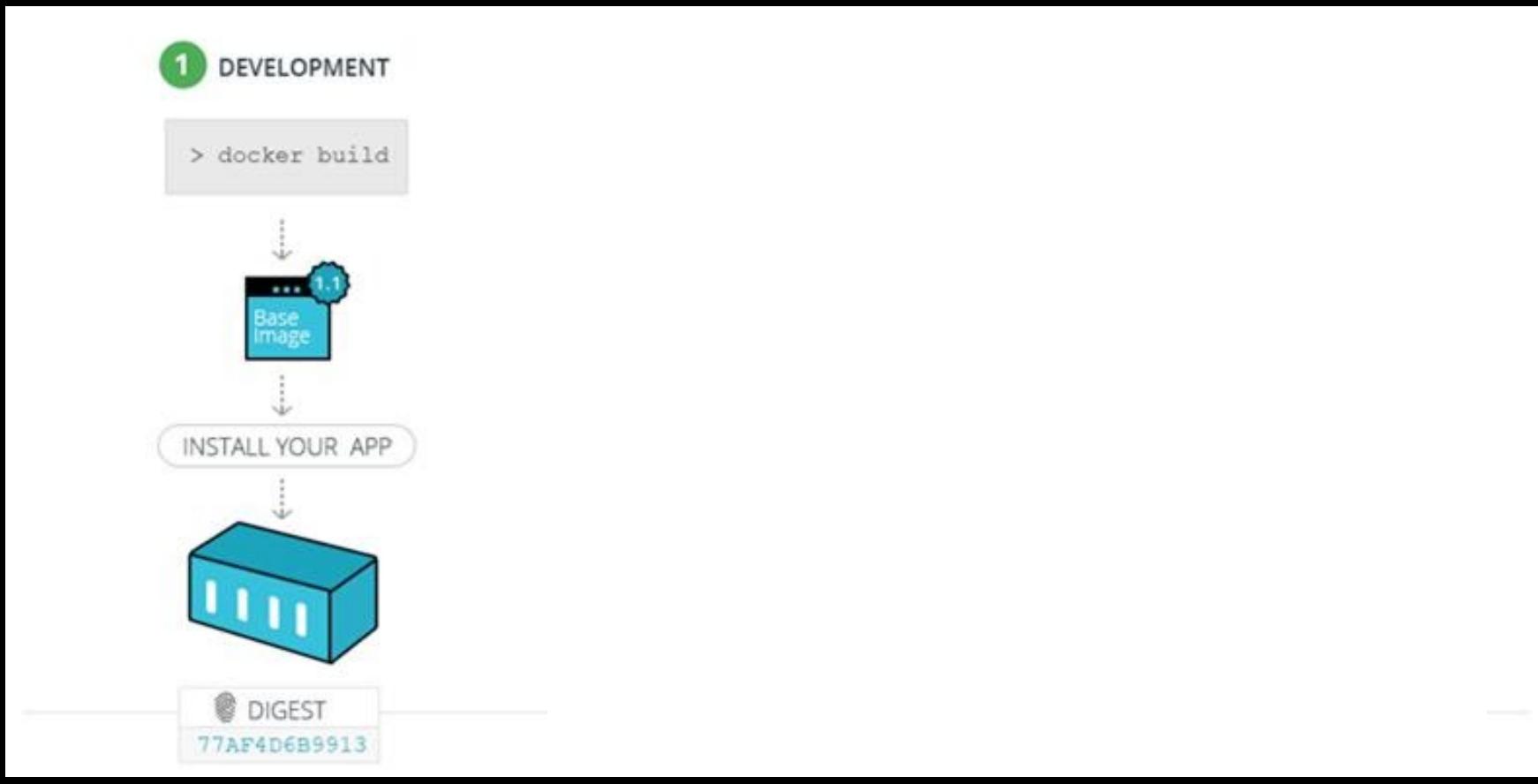


What about
this?

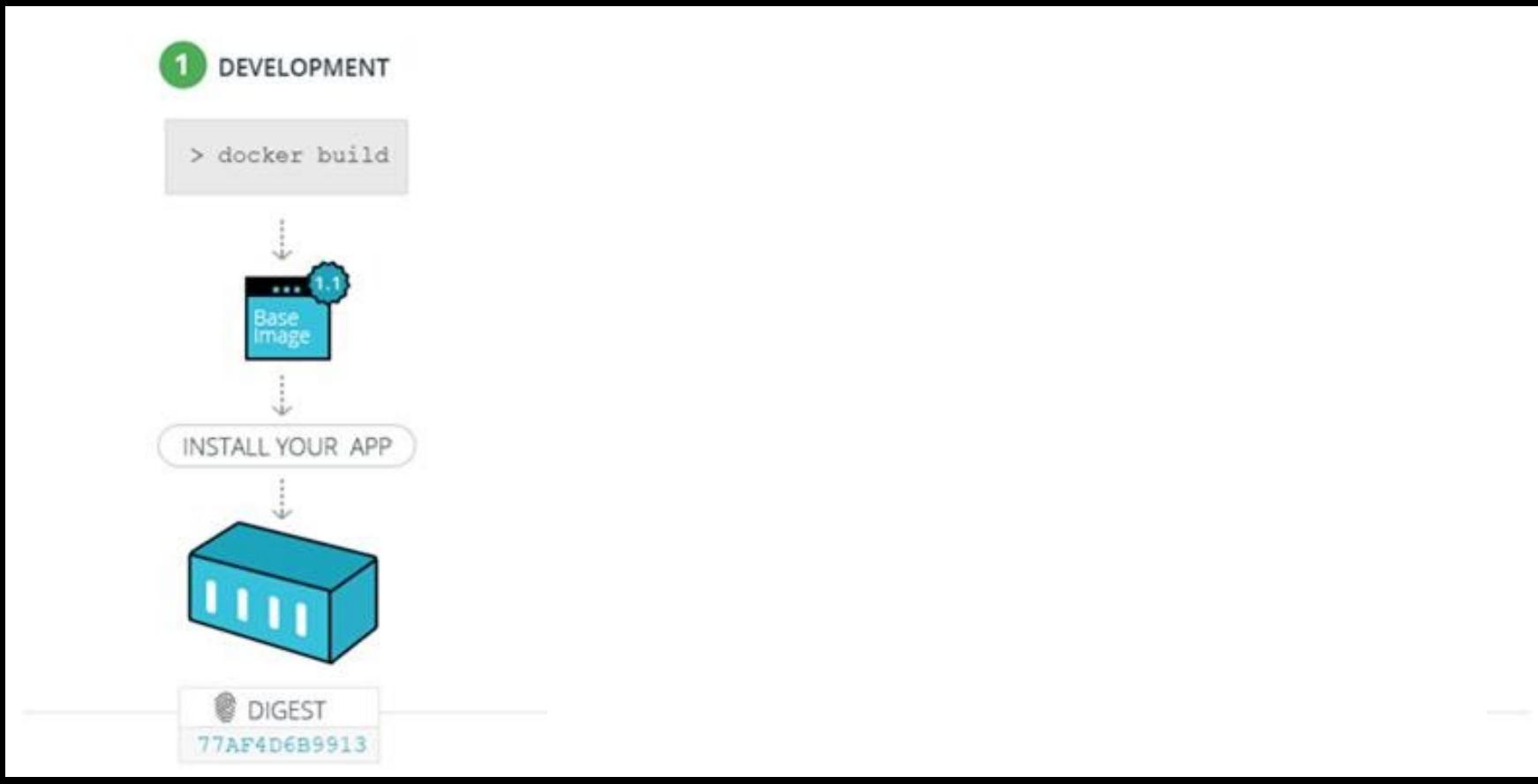
```
CMD ["/usr/bin/node", "/var/www/app.js"]
```



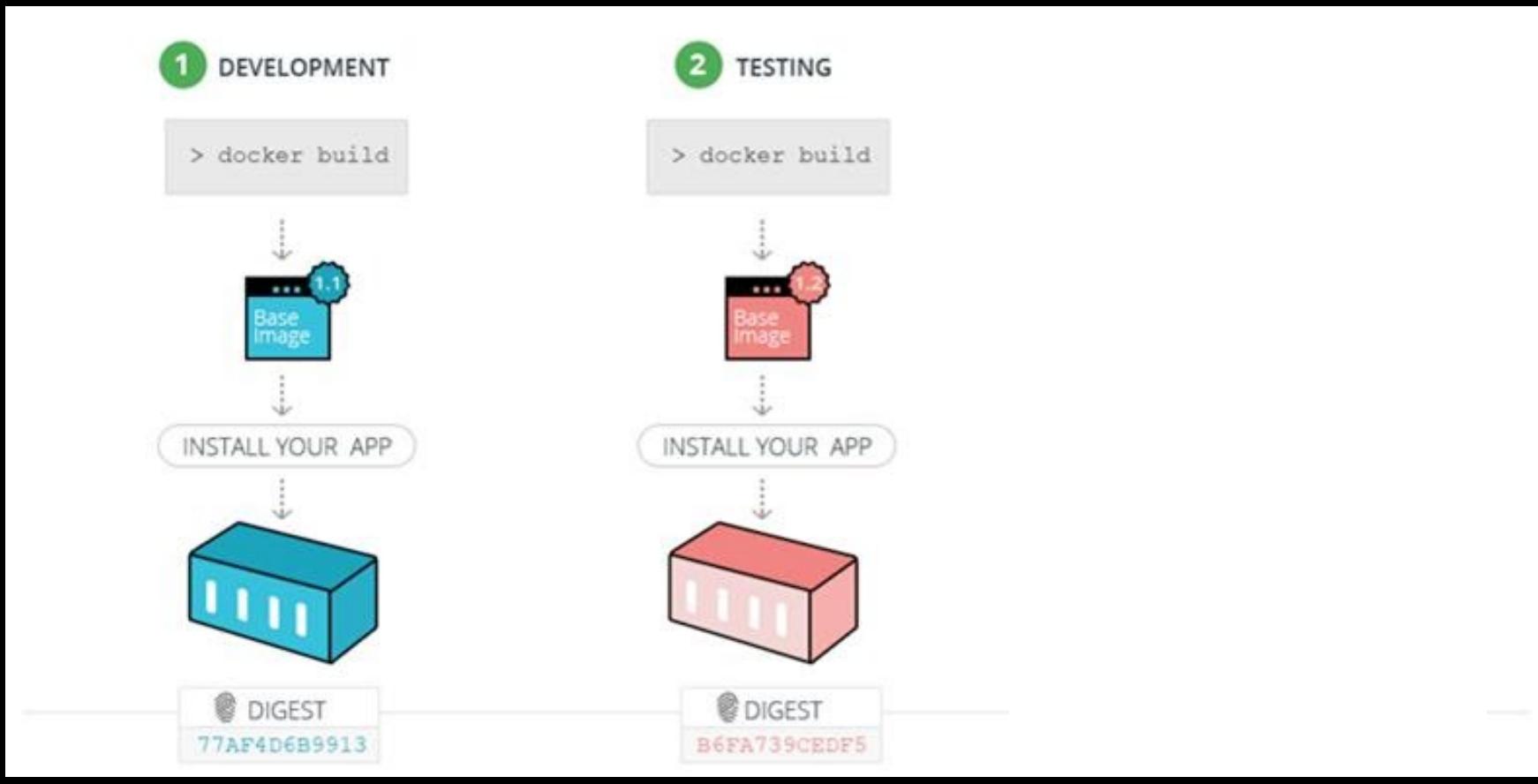
THAT'S WHY YOU DON'T TRUST DOCKER



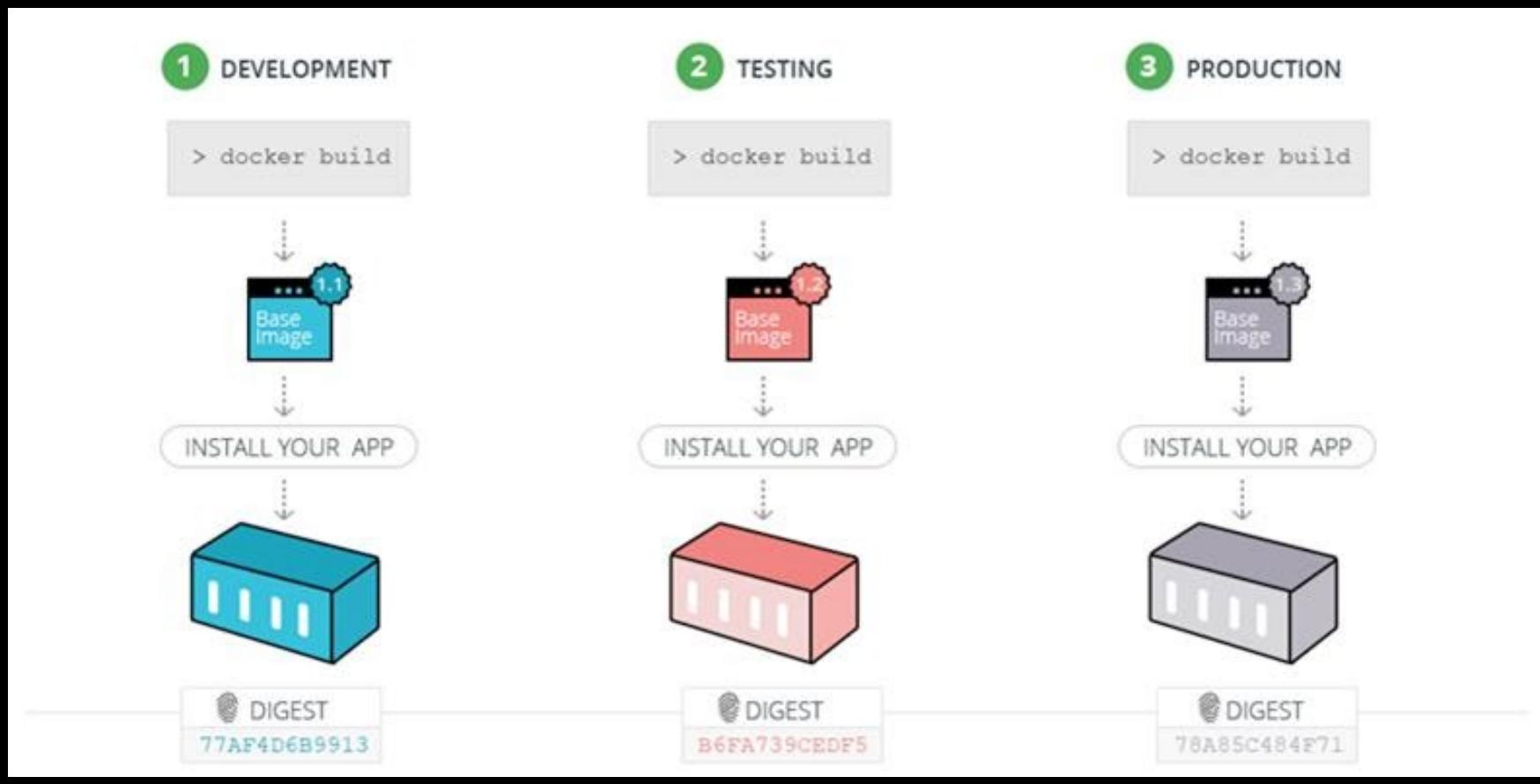
THAT'S WHY YOU DON'T TRUST DOCKER



THAT'S WHY YOU DON'T TRUST DOCKER



THAT'S WHY YOU DON'T TRUST DOCKER



I DON'T ALWAYS BUILD PROMOTION PIPELINES

A black and white photograph of an older man with a full, grey beard and mustache. He is wearing a dark suit jacket over a light-colored shirt. He is seated, holding a cigar in his right hand, which is resting on a dark surface. His gaze is directed towards the camera with a slight smile.

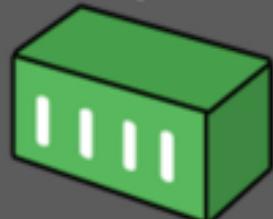
BUT WHEN I DO, IT'S WITH
IMMUTABLE AND STABLE BINARIES

1 DEVELOPMENT

```
> docker build
```



INSTALL YOUR APP

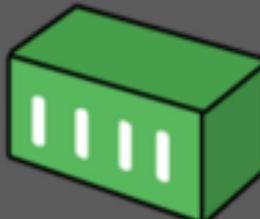


DIGEST
77AF4D6B9913

2 TESTING

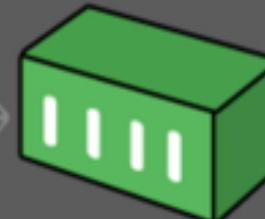
> build once

> automatically promote the same image through the pipeline to production



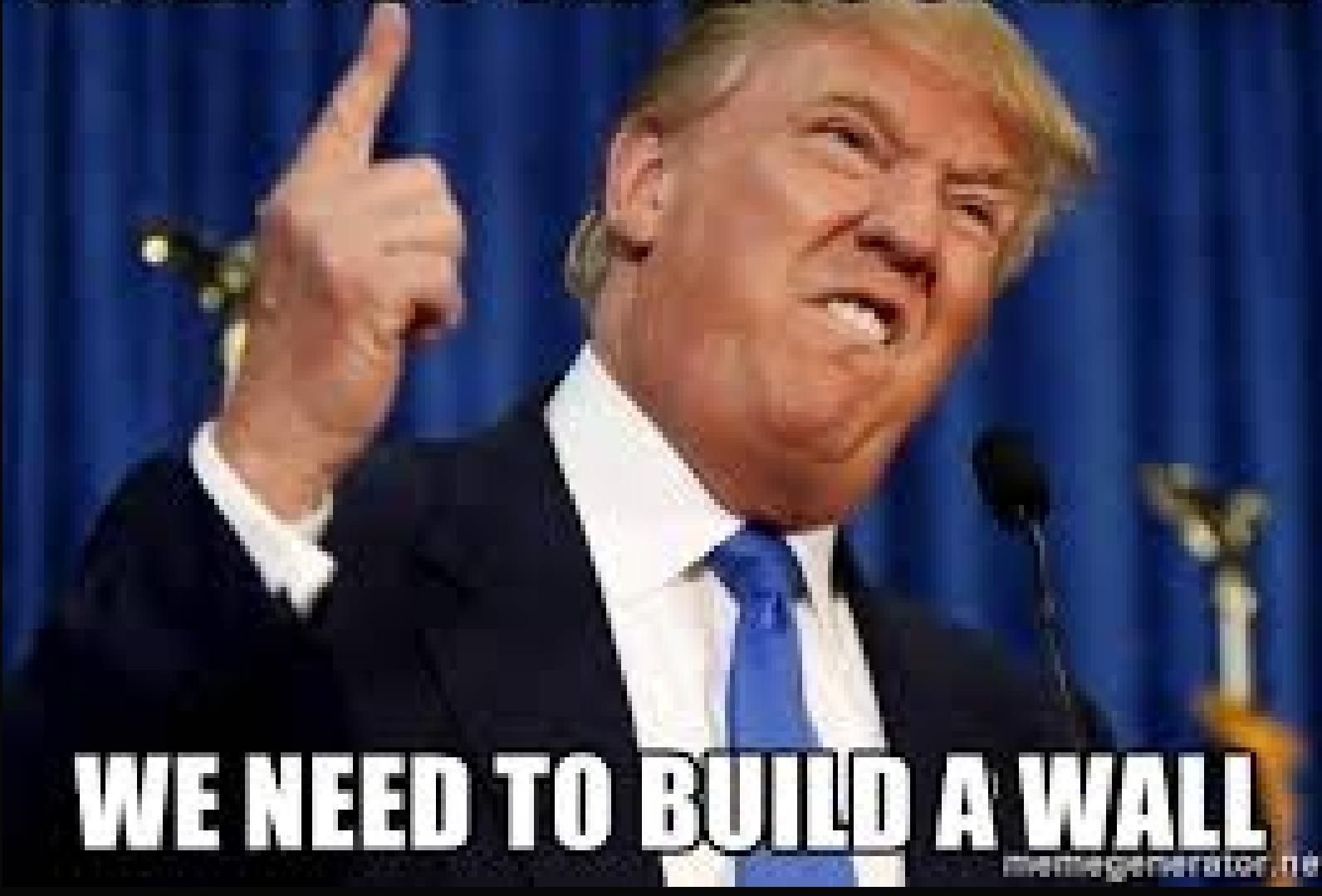
DIGEST
77AF4D6B9913

3 PRODUCTION



DIGEST
77AF4D6B9913

WHEN YOU SEE LOTS OF ENEMY



WE NEED TO BUILD A WALL

memegenerator.net

What's up with the gates?!

What's up with the gates?!

What's up with the gates?!

- QA shouldn't test dev images

What's up with the gates?!

- QA shouldn't test dev images
- non-tested images shouldn't be staged

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!





**SO HOW THE HELL
DO I SEPARATE
DEV FROM PROD?!**

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

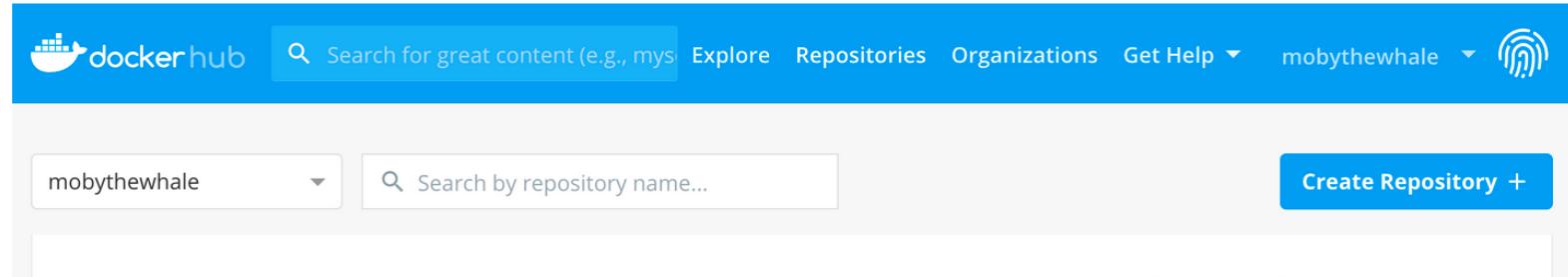
Estimated reading time: 5 minutes

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**).

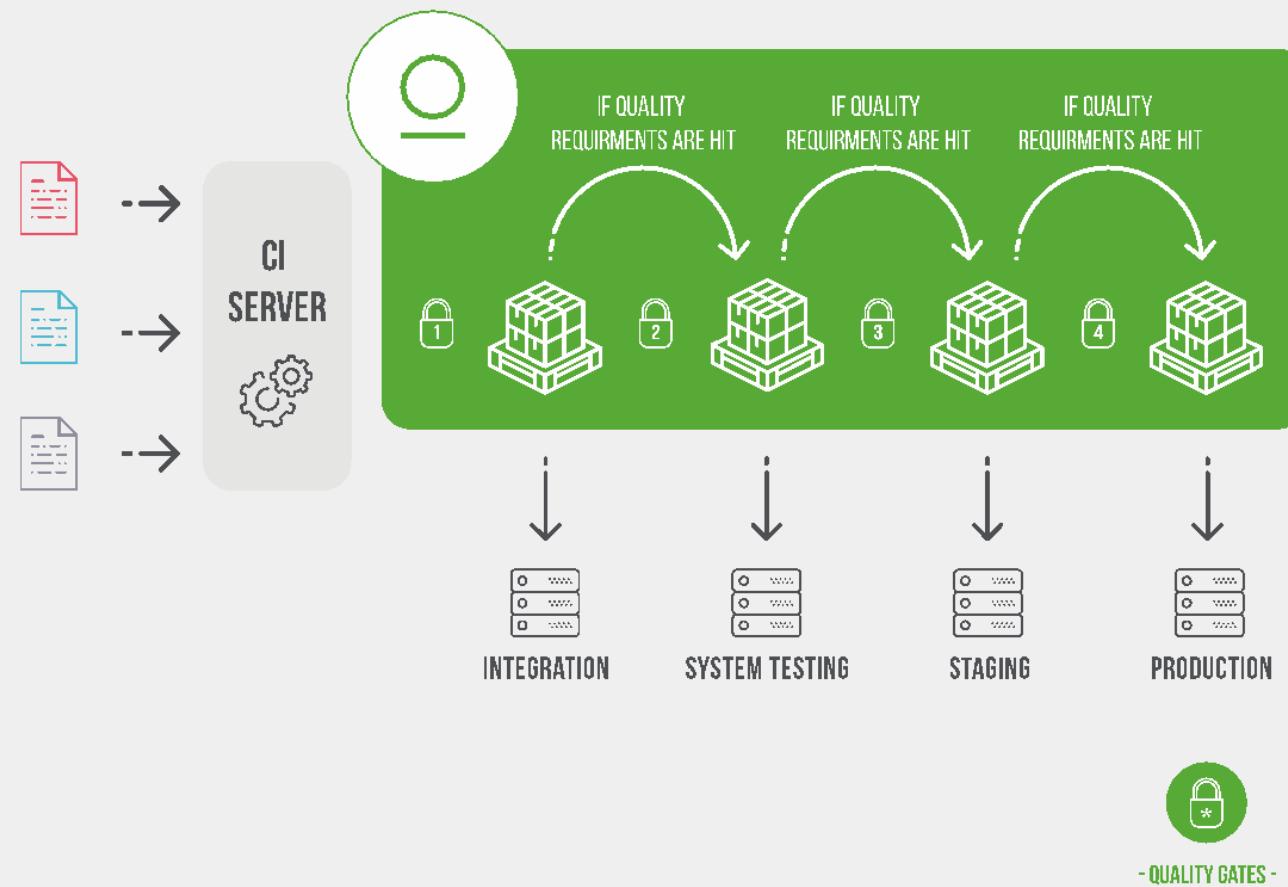
Creating repositories

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





Separate registries per environment





Trumped-up limitations



imgflip.com

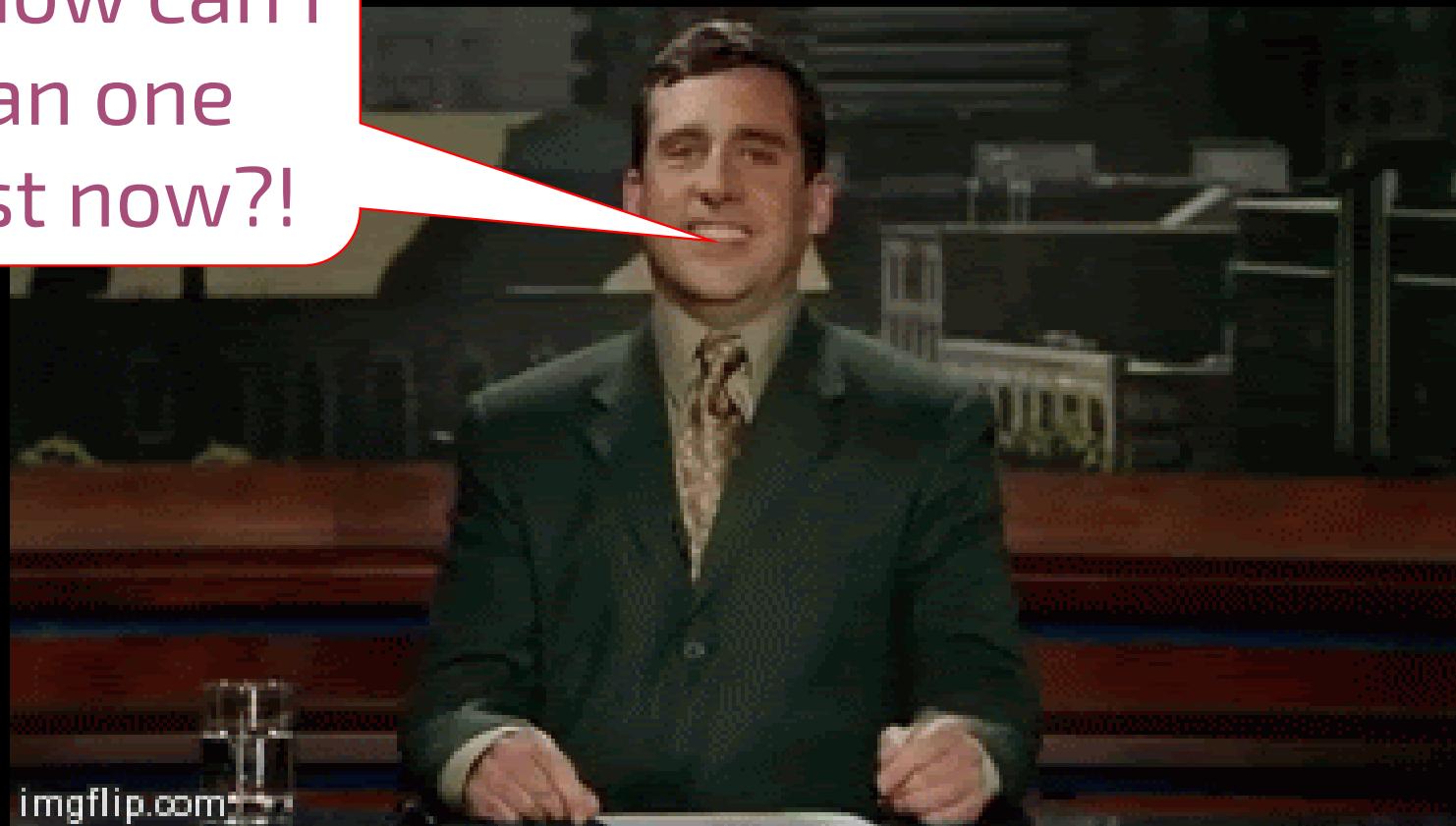
The Anatomy of Docker Tag

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?!



imgflip.com

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!



Virtual hosts/ports to the rescue

docker tag host:port/busybox

Registry host

Tag name

<https://host:port/v2/busybox>

Virtual hosts/ports to the rescue

docker tag host:port/busybox

Registry host

Tag name

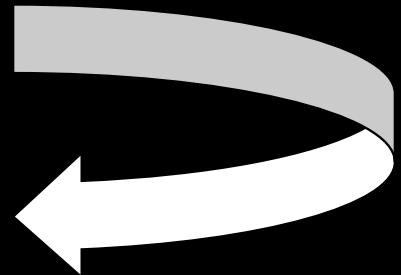
<https://host:port/v2/busybox>

<https://host:8081/registry/docker-dev/busybox>

Context name

Registry name

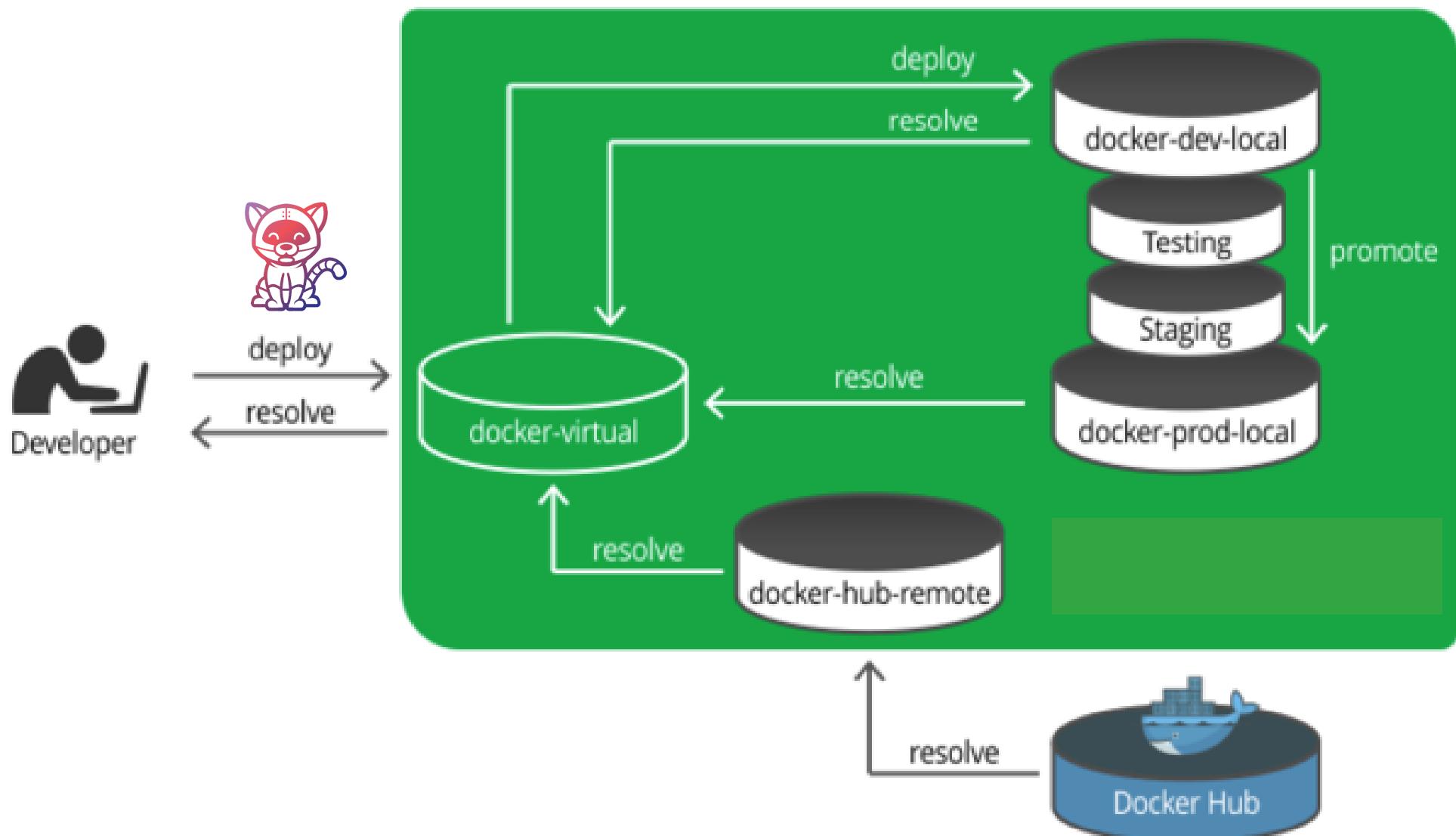
Tag name

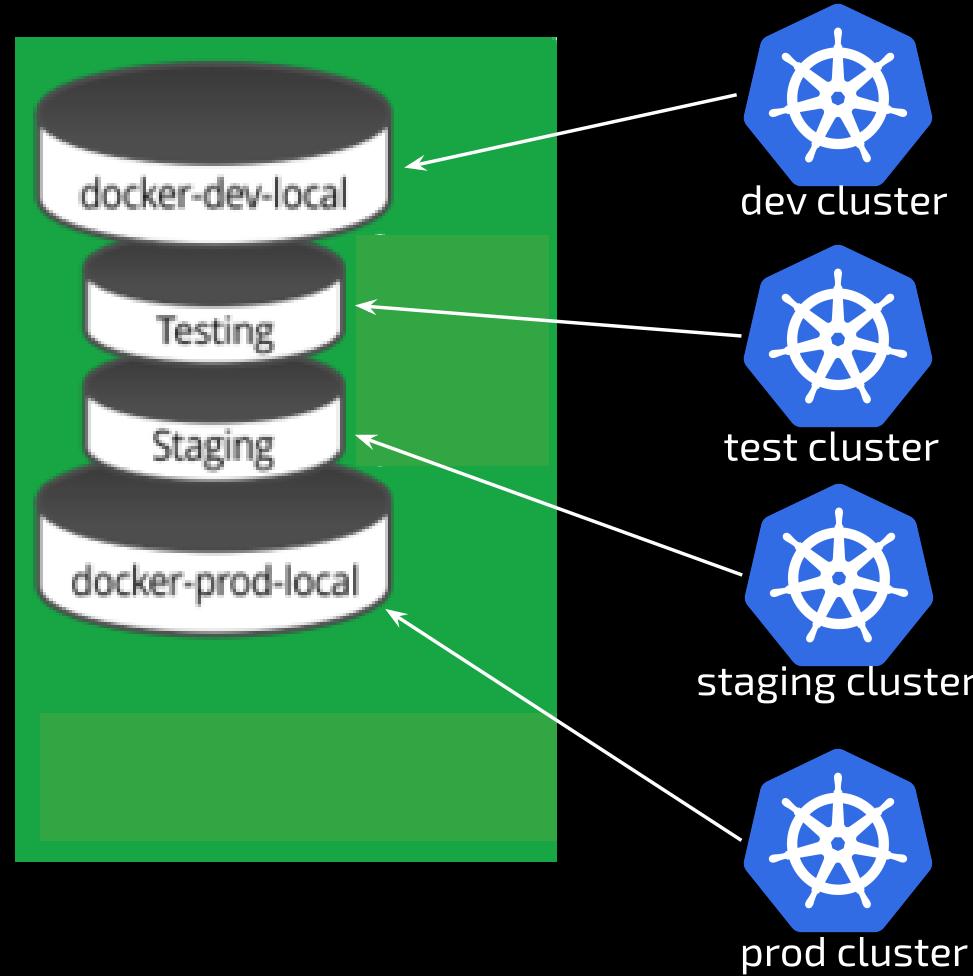


```
server {  
    listen 5001  
  
    server_name 192.168.99.100  
    if ($http_x_forwarded_proto = "")  
        set $http_x_forwarded_proto $scheme;  
  
    rewrite ^/(v1|v2)/(.* ) /artifactory/api/docker/docker-dev/  
    ...  
}  
}
```

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









Win-win-win

Win-win-win

Win-win-win

- Single point of access to multiple registries when needed

Win-win-win

- Single point of access to multiple registries when needed
- Completely isolated environments

Win-win-win

- Single point of access to multiple registries when needed
- Completely isolated environments
- Immediate and free promotions

BUT I LOVE MY LATEST!



Happily serving 74 artifacts

Tree Simple ⚡

- > docker
- > artifactfactory-build-info
- > docker-local
 - > nginx
 - > 1.16-perl
 - > 1.16.1
 - > 1.17-perl
 - > 1.17.10
 - > 1.17.10-alpine-perl
 - > latest
 - manifest.json
 - sha256_123275d6e508d282237a22fefa5aef822b71
 - sha256_6cd6a943ce277e76a108b1c860c5991d3c9a
 - sha256_a50b5ac4a7fb2ab0d0cc3f2d3c1063d9f7751
 - sha256_e791337790a6181d5ce870b3bb16de1a4d5
 - > ubuntu
 - > docker-production-local
 - > docker-staging
 - > generic-local
 - > bintray-docker-remote
 - > docker-remote
 - > bintray-docker-remote-cache
 - > docker-remote-cache

> Trash Can

latest

General

Docker Info

Effective Permissions

Properties

>

Tag Info

Title:	nginx:latest
Digest:	sha256:6b3b6c113f98e901a8b1473dee4c268cf37e93d72bc0a01e57c65b4ab99e58ee
Total Size:	51.0 MB
Ports:	80/tcp
Label Count:	1

Docker Layers Visualization

> CMD

latest

> STOPSIGNAL

> EXPOSE

> RUN

> RUN

> ENV

> ENV

> FNV



Win-win

Win-win

Win-win

- Simplicity of latest

Win-win

- Simplicity of latest
- Always know what it really means

Win-win

- Simplicity of latest
- Always know what it really means
- As long as you promoted immutable artifact

A photograph of Seth Meyers, host of "Late Night with Seth Meyers". He is seated at his desk, wearing a dark suit, white shirt, and patterned tie. He is gesturing with his right hand while speaking. A yellow speech bubble with a red outline is positioned in the lower-left foreground, containing the text "But what about the rest of the dependencies?".

But what about the rest
of the dependencies?

- [Dashboard](#)
- [JFrog Container Registry](#)
- [Packages](#)
- [Builds](#)
- [Artifacts](#)
- [Distribution](#)
- [Pipelines](#)
- [Security & Compliance](#)

Happily serving 74 artifacts

Tree Simple



- > docker
- > artifact-build-info
- > docker-local
- > docker-production-local
- > docker-staging
- ▼ generic-local
 - └ apache-tomcat-8.5.54.zip
 - └ apache-tomcat-8.5.54
 - > bin
 - > conf
 - > lib
 - logs
 - > temp
 - > webapps
 - work
 - BUILDING.txt
 - CONTRIBUTING.md
 - LICENSE
 - NOTICE
 - README.md
 - RELEASE-NOTES
 - RUNNING.txt
 - > bintray-docker-remote
 - > docker-remote
 - > generic-local-remote

apache-tomcat-8.5.54.zip

General

Effective Permissions

Properties

Followers



Info

Name:	apache-tomcat-8.5.54.zip
Repository Path:	generic-local/apache-tomcat-8.5.54.zip
URL to file:	http://34.77.212.178:8082/artifactory/generic-local/apache-tomcat-8.5.54.zip
Module ID:	N/A
Deployed By:	admin
Size:	10.34 MB
Created:	22-04-20 00:30:53 +00:00
Last Modified:	22-04-20 00:40:54 +00:00
Downloads:	0
Remote Downloads:	0

Checksums

SHA-256:	8449e3c0ae37d1f8a7733aa921f89a865d3beec64f0ba69db8a28ade9d556e8c (Uploaded: Identical)
SHA-1:	b7ec124bb1175a1b768d43799d70426ba4e89b88 (Uploaded: Identical)
MDS:	7f533f393df64030e45aded0a438e053 (Uploaded: Identical)

Own your dependencies

Own your dependencies

Own your dependencies

- Your base image

Own your dependencies

- Your base image
- Your infra

Own your dependencies

- Your base image
- Your infra
- Your application files

CONCLUSIONS

CONCLUSIONS

CONCLUSIONS

- Build only once

CONCLUSIONS

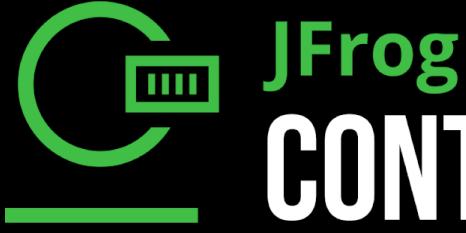
- Build only once
- Separate environments

CONCLUSIONS

- Build only once
- Separate environments
- Promote what you've built

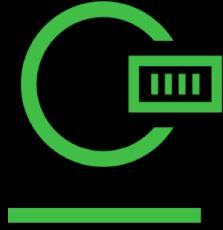
CONCLUSIONS

- Build only once
- Separate environments
- Promote what you've built
- Own your dependencies



The only **hybrid**,
enterprise-ready container
registry





JFrog

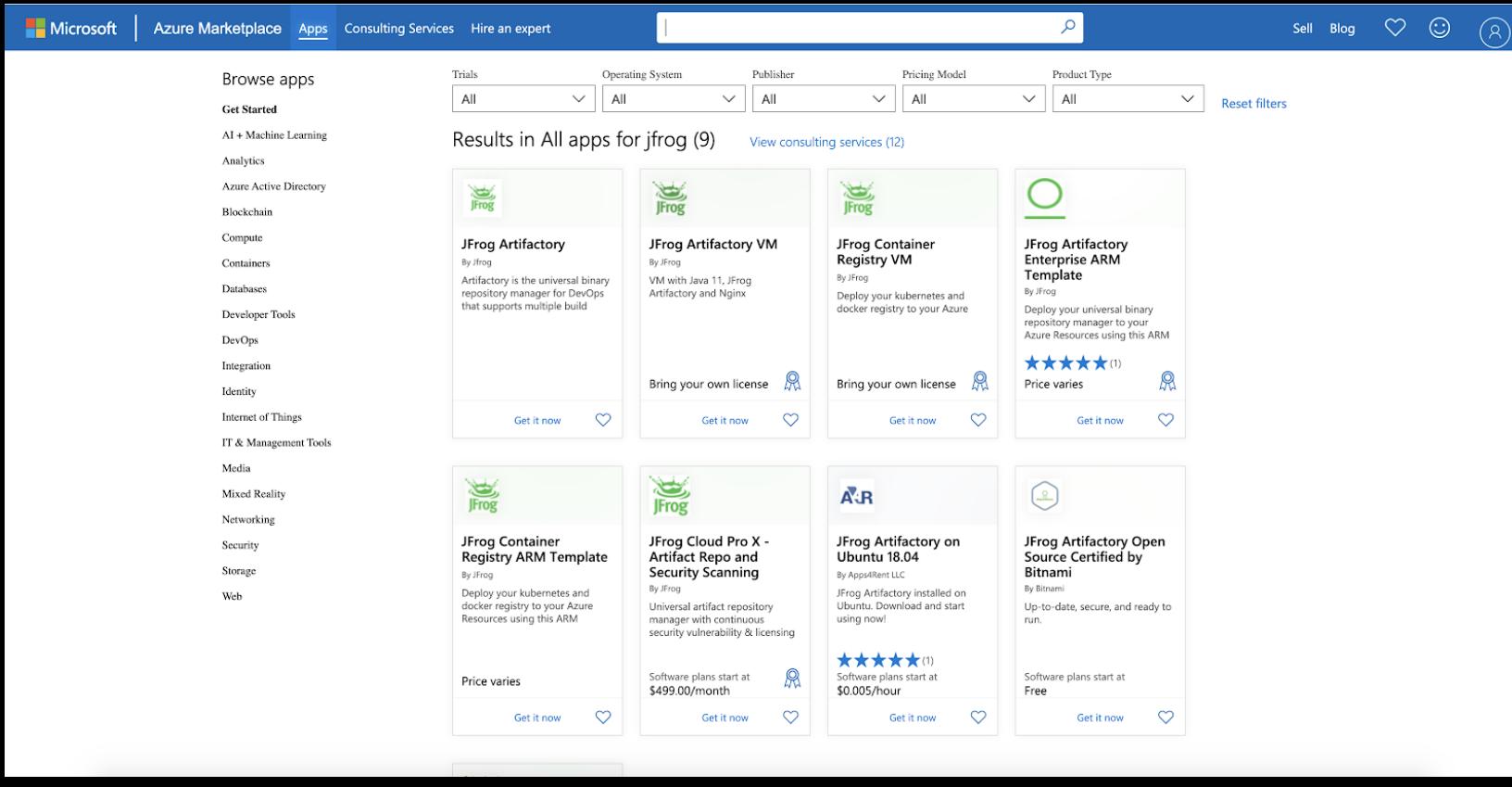
CONTAINER REGISTRY

Demo time



Creates everything on one click with Azure marketplace

<https://azuremarketplace.microsoft.com/en-us/marketplace/apps?search=jfrog>



The screenshot shows the Azure Marketplace search results for "jfrog". The search bar at the top contains the query "jfrog". Below the search bar, there are several filter options: Trials (All), Operating System (All), Publisher (All), Pricing Model (All), and Product Type (All). A "Reset filters" button is also present. The main area displays nine results for "jfrog" apps:

- JFrog Artifactory** by JFrog: Artifactory is the universal binary repository manager for DevOps that supports multiple build types.
- JFrog Artifactory VM** by JFrog: VM with Java 11, JFrog Artifactory and Nginx.
- JFrog Container Registry VM** by JFrog: Deploy your Kubernetes and Docker registry to your Azure.
- JFrog Artifactory Enterprise ARM Template** by JFrog: Deploy your universal binary repository manager to your Azure Resources using this ARM.
- JFrog Container Registry ARM Template** by JFrog: Deploy your Kubernetes and Docker registry to your Azure Resources using this ARM.
- JFrog Cloud Pro X - Artifact Repo and Security Scanning** by JFrog: Universal artifact repository manager with continuous security vulnerability & licensing.
- JFrog Artifactory on Ubuntu 18.04** by Appsecent LLC: JFrog Artifactory installed on Ubuntu. Download and start using now!
- JFrog Artifactory Open Source Certified by Bitnami** by Bitnami: Up-to-date, secure, and ready to run.

Each app listing includes a "Get it now" button and a "Love" icon. The "JFrog Artifactory Enterprise ARM Template" has a 5-star rating. The "JFrog Artifactory Open Source Certified by Bitnami" listing indicates "Software plans start at Free".

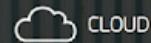


JFrog Container Registry Available NOW!

[PRODUCTS](#)[PLATFORM](#)[RESOURCES](#)[SERVICES](#)[PRICING](#)[COMPANY](#)[COMMUNITY](#)

JFROG CONTAINER REGISTRY

The world's most advanced, powerful, hybrid Docker and Helm registry.

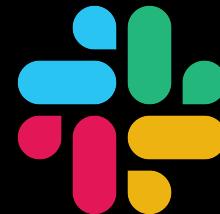
[DOWNLOAD FREE NOW >](#)[TRY FREE NOW >](#)

zip download. Other options including Docker, RPM, Debian click [here](#).

<http://jfrog.com/container-registry>



For the Community By the Community



QA Time

- Presentation Slides
- Video later
- JFrog T-Shirts Raffle
 - *Win 1 of 3*
- *batelt@jfrog.com*

