Establishing Image Provenance and Security in Kubernetes

Adrian Mouat

Container Solutions

info@container-solutions.com
www.container-solutions.com

Photo by Eddie Howell

©Stuart Caie
flickr.com/photos/kyz/3252484607

KWTFIGOIYC

@adrianmouat
Container Solutions

It should be immediately clear for all containers

- What it is
- Where it came from
- How it was created
- If it’s up-to-date

Need to **understand** Kubernetes

…it doesn’t always work with you

```
        </body>
        </html>
amouat@marvin Pizza$ curl localhost:8000

        <!DOCTYPE html>
        <html lang=en>
        <head>
            <meta charset=utf-8>
            <title>Quattro Formaggi</title>
        </head>
        <body>
            <h1>Today's Pizza is Quattro Formaggi</h1>
            <img src="./img/quattro.jpg" width="400"/>
            <p>Cheese, cheese, cheese and more cheese</p>
        </body>
        </html>
amouat@marvin Pizza$ docker stop $(docker ps -lq)
c93a2e9cf00b
amouat@marvin Pizza$ docker push amouat/pizza:today
The push refers to repository [docker.io/amouat/pizza]
07bf02eafabe: Layer already exists
93867f94cfb2: Layer already exists
f04dfc518748: Layer already exists
today: digest: sha256:3f4d3a196c48f03947ef9c783791e7c3e48857d4d9bb32c64d4026490368a502 size: 951
amouat@marvin Pizza$ kubectl version
Client Version: version.Info{Major:"1", Minor:"8", GitVersion:"v1.8.6", GitCommit:"6260bb08c46c31eea6cb538b34a9ceb3e406689c", GitTreeState:"clean
", BuildDate:"2017-12-21T06:34:11Z", GoVersion:"go1.8.3", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"9+", GitVersion:"v1.9.6-gke.1", GitCommit:"cb151369f60073317da686a6ce7de36abe2bda8d", GitTreeState
:"clean", BuildDate:"2018-04-07T22:06:59Z", GoVersion:"go1.9.3b4", Compiler:"gc", Platform:"linux/amd64"}
amouat@marvin Pizza$ kubectl get nodes
NAME                                       STATUS    ROLES     AGE       VERSION
gke-cluster-1-default-pool-fae89b63-q7l2   Ready     <none>    1h        v1.9.6-gke.1
gke-cluster-1-default-pool-fae89b63-wz4v   Ready     <none>    1h        v1.9.6-gke.1
gke-cluster-1-default-pool-fae89b63-xsrt   Ready     <none>    1h        v1.9.6-gke.1
amouat@marvin Pizza$ kubectl get pods
No resources found.
amouat@marvin Pizza$
```

*The **default** semantics for pulling images in Kubernetes are **confusing** and **dangerous***

@adrianmouat

(Partial) Solution

```
imagePullPolicy: Always
```

Make Default with

`AlwaysPullImages`

Admission Controller

Docker Swarm Mode Resolving

Images are **resolved to digest**
Before being sent to nodes

Kubernetes views tags as **immutable**

Docker views tags as **mutable**

Both are useful.

@adrianmouat

Suggestion for new tooling

Tags are ***immutable*** by default

"Flows" for named ***sequences***

WHAT IS IT?

- Name images appropriately
  - SemVer, githash
- Use labels and metadata
  - OCI annotations

HOW WAS IT CREATED?

*“**Reproducibility** is a **virtue**”*

Dinah McNutt
Google Release Engineer
@dinahSBR

Reproducible Docker Builds

- Use tagged base images
  - or digests
- Version package installed software
  - run a mirror for total control
- Take care with downloaded s/w
  - Checksums and GPG FTW

Not Binary Reproducible

- File timestamps
- Other metadata
  - Build container IDs
  - Created timestamp

Bazel

@adrianmouat
Container Solutions

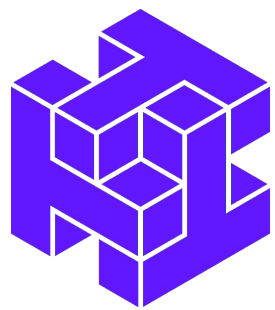
IS IT UP-TO-DATE?

Photo by Aqua Mechanical

clair

aqua

NeuVector



Twistlock


weaveworks

PROVE IT!

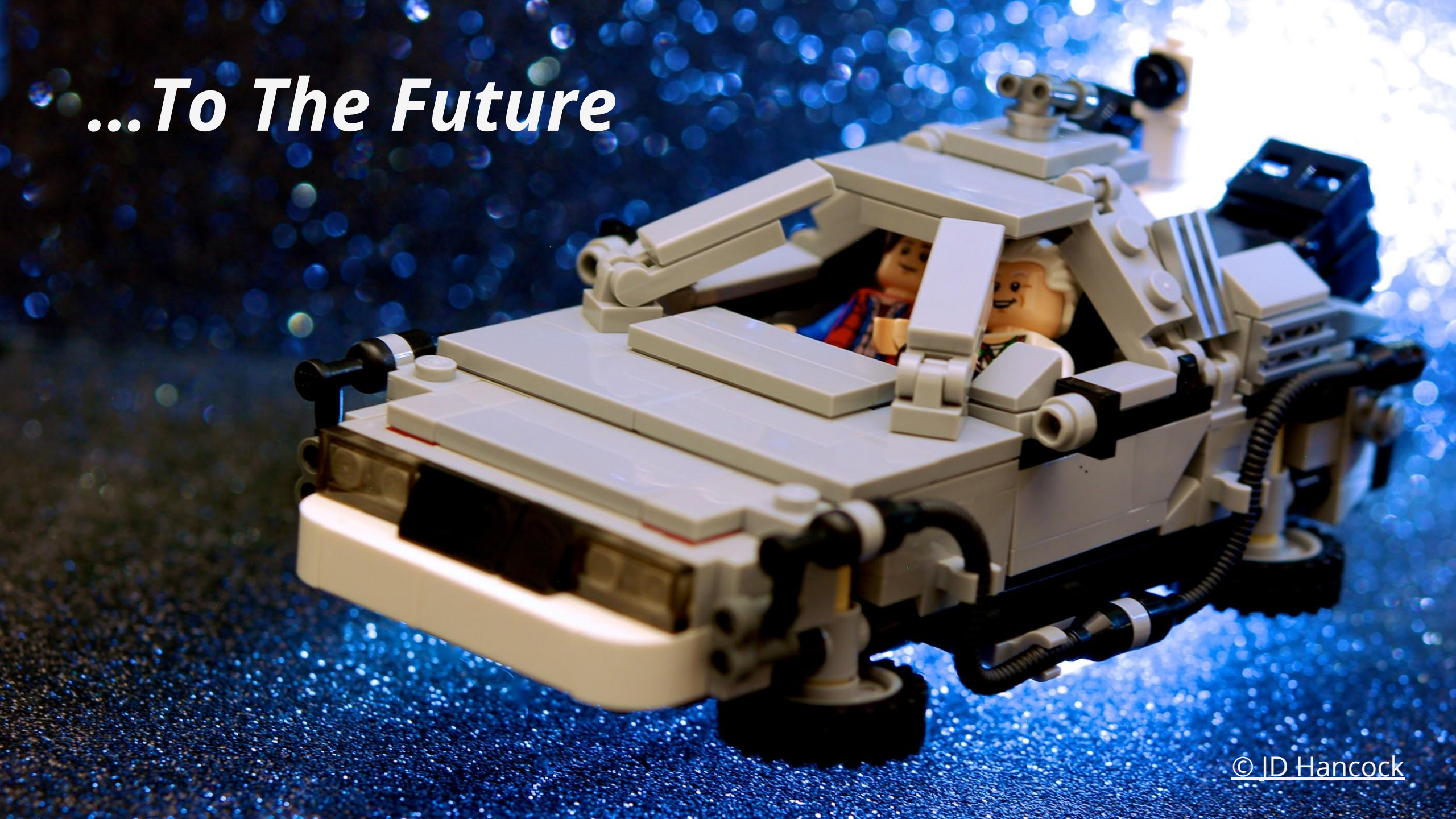
- Digests are great
  - Content hashes
  - Unwieldy
- GPG signing useful


notary

Only run images from a ***controlled*** registry

- Not easily possible
- Should be

...To The Future

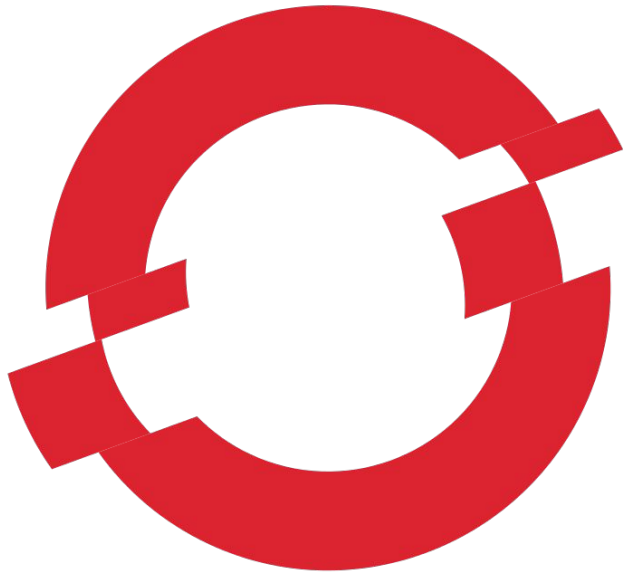


Kubernetes' ***strength*** is its ***distributed*** architecture

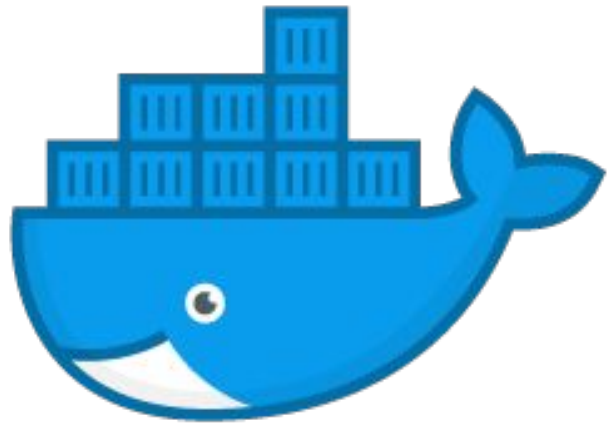
Kubernetes' ***weakness*** is its ***distributed*** architecture

@adrianmouat

More ***holistic*** solutions



OPENSHIFT



docker

More ***tooling***

Grafeas

Trow.io

Image Management for Clusters

- KWTFIGOIYC
- Understand Kubernetes
- Think Holistic
- Provenance
- Reproducibility
- Tooling

References

- Trow https://trow.io
- Grafeas https://grafeas.io/
- OCI Annotations https://github.com/opencontainers/image-spec/blob/master/annotations.md
- Release Engineering (from Google SRE Book) https://landing.google.com/sre/book/chapters/release-engineering.html
- AlwaysPullImages Admission Controller https://kubernetes.io/docs/admin/admission-controllers/#alwayspullimages
- ImageStreams in OpenShift https://blog.openshift.com/image-streams-faq/
- Docker EE https://www.docker.com/enterprise-edition
- Notary https://github.com/theupdateframework/notary
- Weave Flux https://www.weave.works/oss/flux/
- Clair https://github.com/coreos/clair
- Aqua https://www.aquasec.com/
- NeuVector https://neuvector.com/
- Twistlock https://www.twistlock.com/
- Bazel https://bazel.build/
- Kaniko https://github.com/GoogleContainerTools/kaniko