Preparation

- In a web browser, go to: https://ibm.biz/kubecon-secure-deployment
- This shortened link goes to a Katacoda instance, where you'll find the lab instructions and be given access to a cluster that you can use to complete the steps.
- This link will continue to work after the session, but you'll need to start again once your session expires!

DevSecOps Kubernetes Pipeline Workshop

From @ibm and @controlplaneio

IBM Cloud

Michael Hough

Developer, IBM Cloud Container Registry

Maintainer, Portieris

@molepigeon

controlplane

Sam Irvine

Infrastructure Engineer

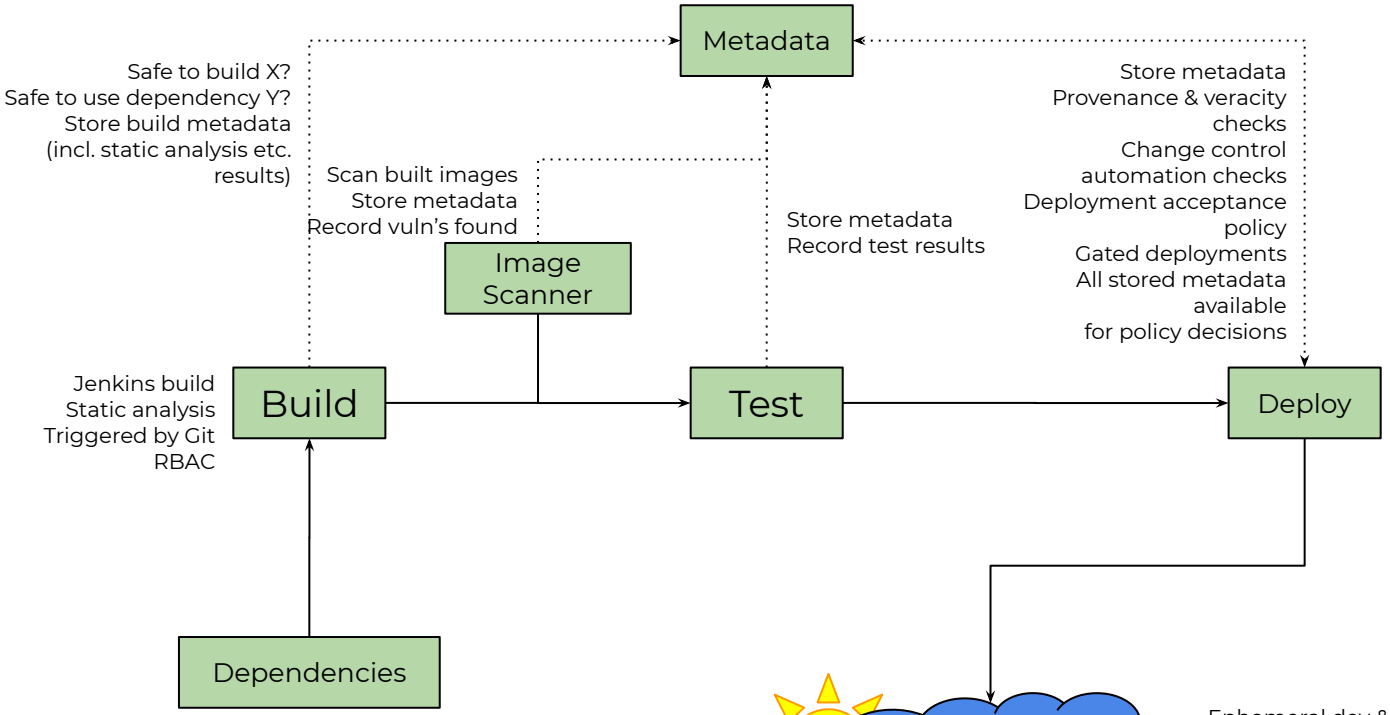
Preparation

- In a web browser, go to: https://ibm.biz/kubecon-secure-deployment
- This shortened link goes to a Katacoda instance, where you'll find the lab instructions and be given access to a cluster that you can use to complete the steps.
- This link will continue to work after the session, but you'll need to start again once your session expires!

Secure Pipelines


/ROOT:#
STOP
#HaCk3r
RIGHT
WRONG
FRAGILE
#YOU HAVE BEEN#
HACKED!!
TO BE FREE
I read your email

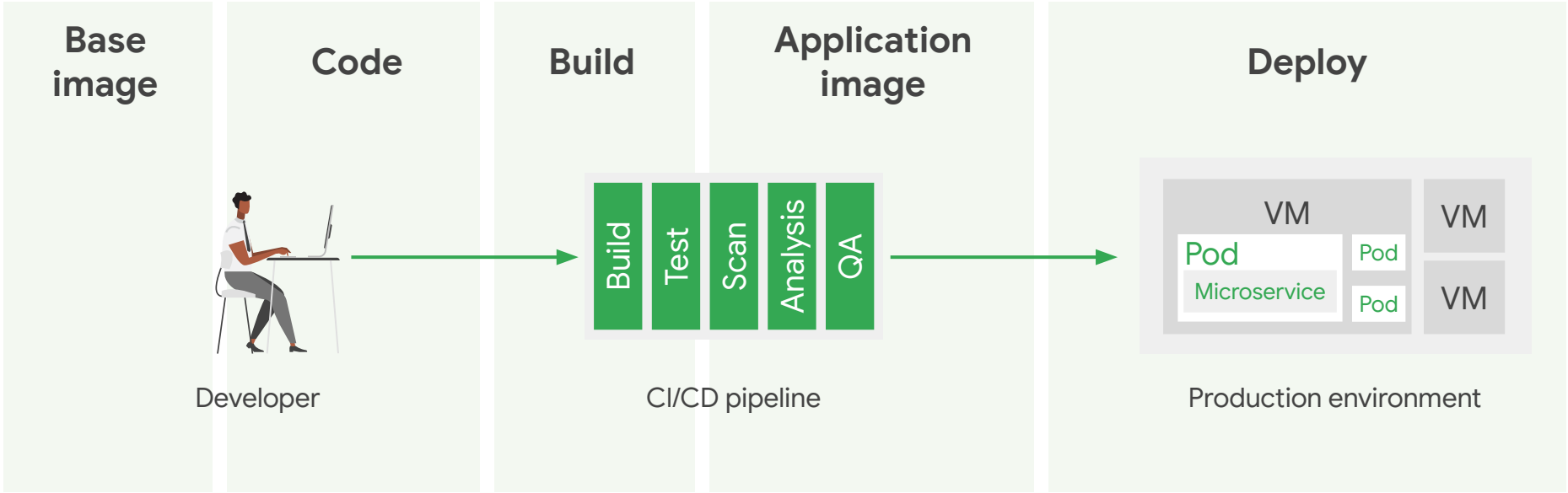
Secure Cloud-Native Delivery



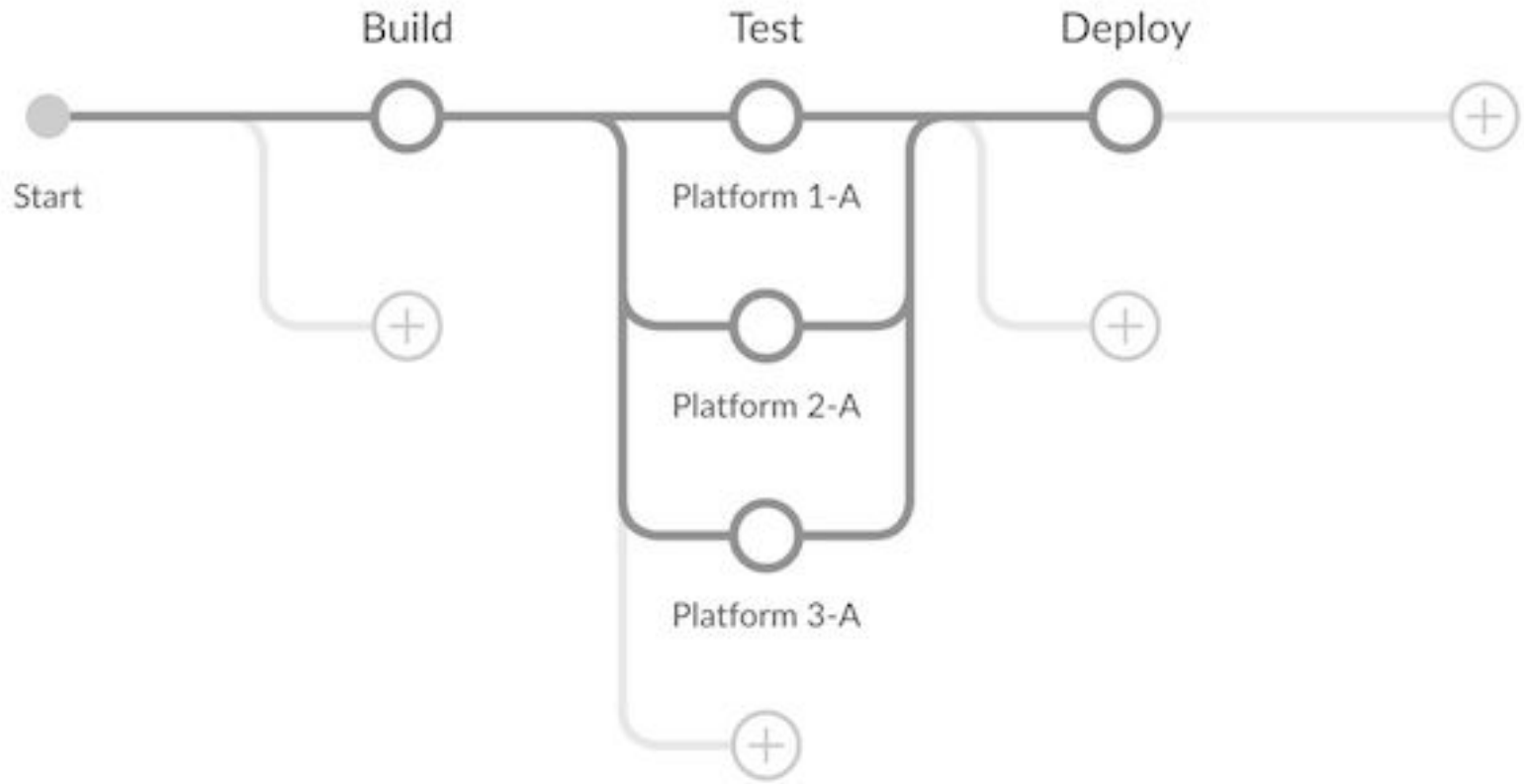
@sublimino

controlplane

Stages of the CDLC (Container Delivery Lifecycle)

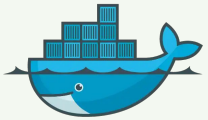


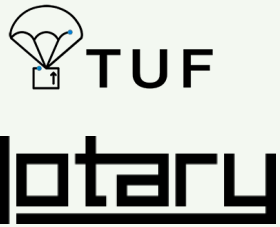
https://www.slideshare.net/FranklinMosley/devsecops-a-new-hope-for-security-in-cicd


Build
Test
Deploy
Start
Platform 1-A
Platform 2-A
Platform 3-A

Open-source supply chain today

Base image	Code	Build	Application image	Deploy
Images: Docker Distribution (Hub)	**Updates:** TUF, Notary	**Pipeline metadata:** Grafeas, in-toto	**Vulnerability scanning:** Clair, Micro Scanner, Anchore Open Source Engine	**Admission control:** K8s admission controllers, Kritis, Portieris

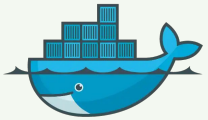




controlplane

Open-source supply chain today

Base image	Code	Build	Application image	Deploy
Images: Docker Distribution (Hub)	**Updates:** TUF, Notary	**Pipeline metadata:** Grafeas, in-toto	**Vulnerability scanning:** Clair, Micro Scanner, Anchore Open Source Engine	**Admission control:** K8s admission controllers, Kritis, Portieris



TUF

Notary

Grafeas

clair

aqua MicroScanner

PORTIERIS

KUBESEC.IO

controlplane

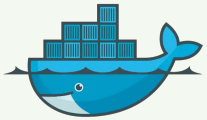
Build Flow

- Build image (base image from Docker Hub)
- Assert absence of vulnerabilities in image (Harbor)
- Cryptographically sign image for later verification
- Push image to container registry
- Attempt to deploy image to cluster
- Verify image has been signed with an admission controller
- Reject images that have not followed due process and organisational policy

Harbor

notary

Notary
Cryptographic
image signing



Docker
Distribution
Container registry

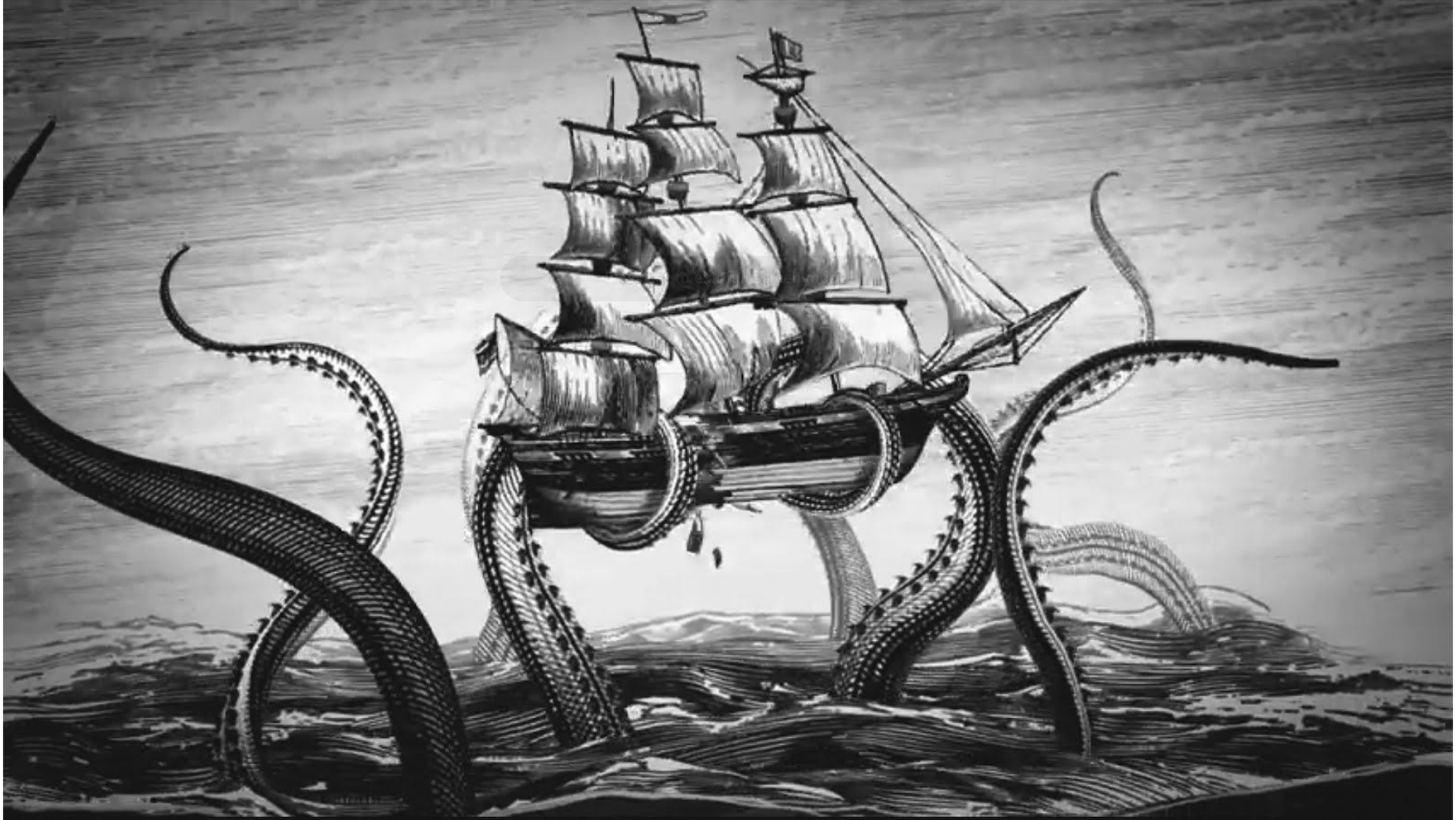
clair

Clair
Image vulnerability
scanning

Harbor

- Container image registry (a “self-hosted Docker Hub”)
- Joined CNCF in July 2018
- Capable of running inside a cluster for inception-esque self-referential image pulls

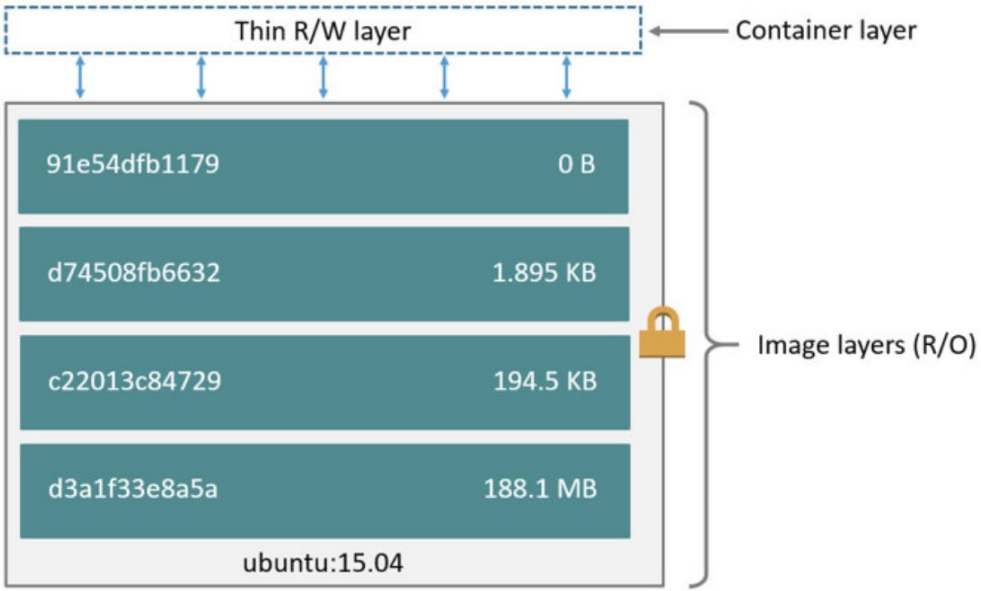
Vulnerable Images



What Can Image Scanning Detect?

- This depends upon the depth of the tool
- Some will just scan installed operating system package manager versions
- Others will check filesystem permissions for all entities, extra binaries, secrets, policies etc.

Image vulnerability scanning approaches



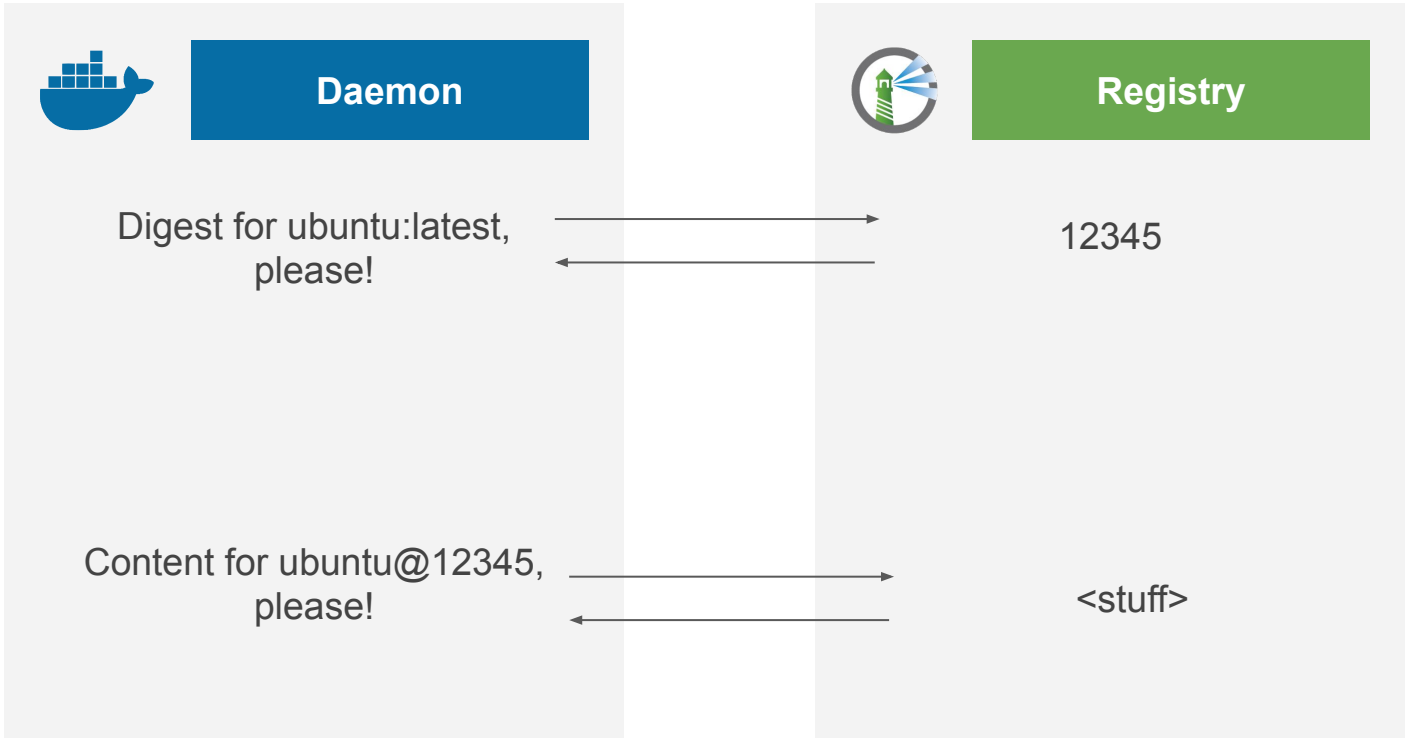
https://sysdig.com/blog/container-security-docker-image-scanning/

- Components to scan: package-level vs. code-level
  - OS packages
  - App library packages
  - JARs, WARs, TARs, etc.
  - Malware
  - Misconfigurations, e.g., secrets
- Scan type
  - Layer-by-layer
  - UnionFS top layer only

Clair vs. MicroScanner vs. Anchore

	Scanning depth	OS covered	Maintainer
clair	Packages		CoreOS
aqua MicroScanner	Packages	Alpine, CentOS, Debian, Oracle Linux, RHEL, Ubuntu	Aqua Security
anchore	Packages, files, software artifacts		Anchore


notary


Daemon
Registry
Digest for ubuntu:latest, please!
12345
Content for ubuntu@12345, please!
<stuff>

Daemon

Notary

Digest for ubuntu:latest, please!

12345, and it's signed by Alice, Bob and Charlie

I trust Bob…

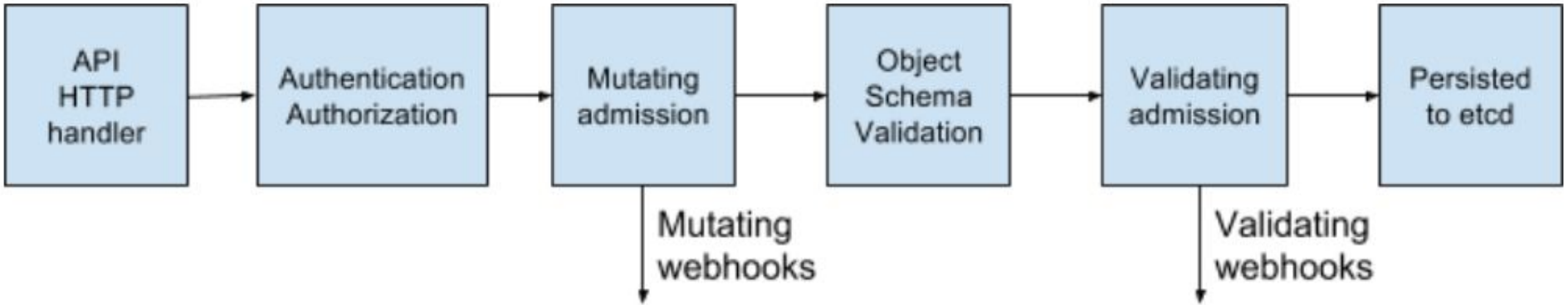
And that's his digital signature!

Registry

Content for ubuntu@12345, please!

<stuff>

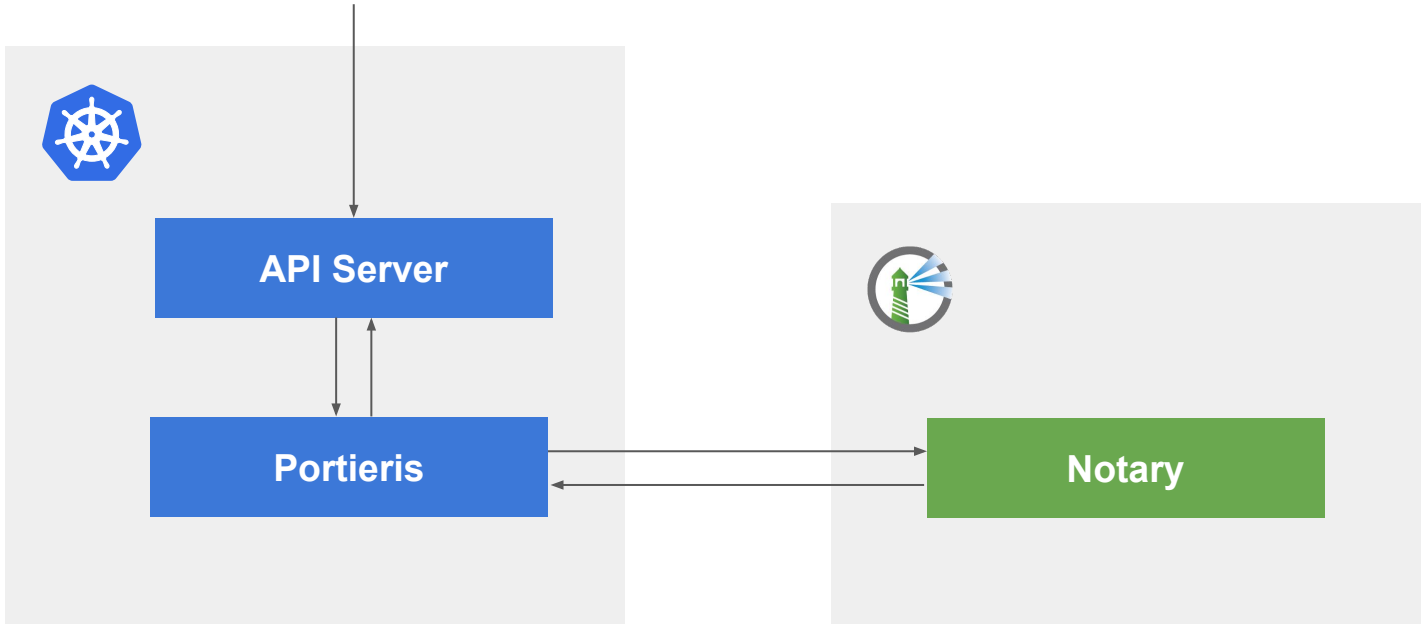
Extensible Admission Controllers



http://blog.kubernetes.io/2018/01/extensible-admission-is-beta.html

controlplane

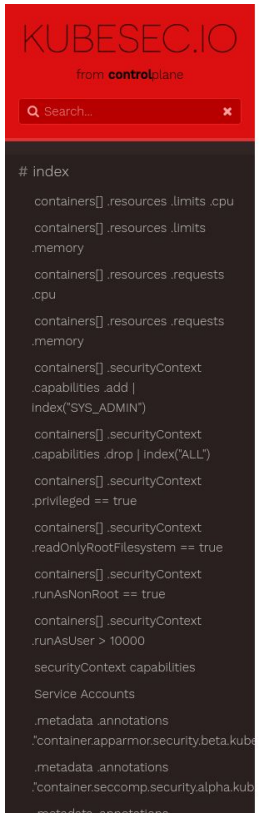

PORTIERIS

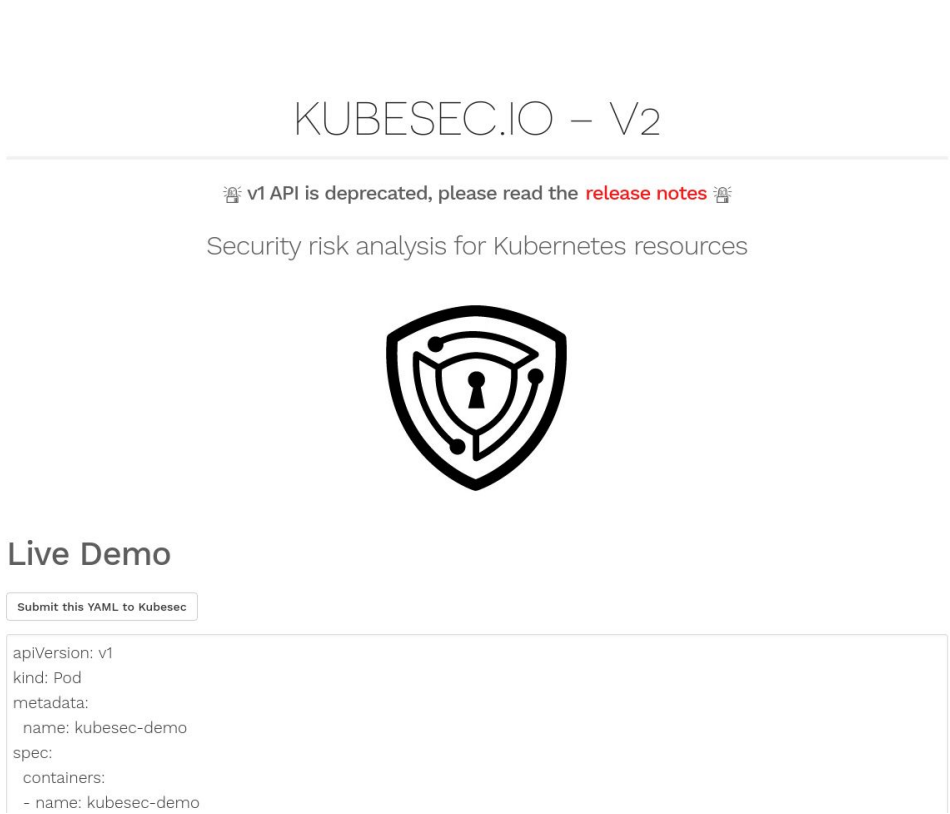

API Server
Portieris
Notary

```
image: ibmcom/portieris:0.5.1
```

```
image: ibmcom/portieris@sha256:19b6e9df327...
```

kubesec.io - risk score for K8S YAML





controlplane

kubesec.io - example insecure pod

```
{
  "score": -30,
  "scoring": {
    "critical": [{
      "selector": "containers[] .securityContext .privileged == true",
      "reason": "Privileged containers can allow almost completely unrestricted host access"
    }],
    "advise": [{
      "selector": "containers[] .securityContext .runAsNonRoot == true",
      "reason": "Force the running image to run as a non-root user to ensure least privilege"
    }, {
      "selector": "containers[] .securityContext .capabilities .drop",
      "reason": "Reducing kernel capabilities available to a container limits its attack surface",
      "href": "https://kubernetes.io/docs/tasks/configure-pod-container/security-context/"
    },
   ...
```

controlplane

More Admission Control

Minimum viable security

- We have
  - Verified the contents of an image are not insecure
  - Signed the image to confirm we have tested it
  - Prevented unsigned images from being deployed to production
- These are the building blocks of a secure pipeline
  - But only focus on the contents of the image and not its runtime configuration
- PodSecurityPolicy and NetworkPolicy should be use to limit the behaviour of the application at runtime
- Further admission controllers can be added to enhance security

Threat Model

- Attacks wholly or partially mitigated:
  - Container image and application supply chain with known CVEs
  - Theft of users' container registry credentials
  - Some build server compromises
- Extant risk:
  - Compromised user or insider threat
  - Zero day vulnerabilities
  - ...the rest of the Kubernetes attack surface!

Try for yourself!

https://ibm.biz/kubecon-secure-deployment

Summary

Vulnerable images

- Tooling can automatically identify vulnerabilities in your apps
- … and prevent you from shipping to production if they’re vulnerable
- CVEs are a likely way for an attacker to being their assault on your systems
- Never ship CVEs to production
- Stuff that’s in production today can be affected by a CVE tomorrow - make sure to stay on top of patching.

Know what you’re deploying

- Scanning for vulnerabilities is important - but only makes any sense if that same image is deployed to production
- Asserting that the image that runs in production contains what you think it does is another basic security precaution that is too-often overlooked
- This security measure can prevent the compromise of access to your container registry from compromising production