KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2023

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Securing Container Supply Chain in CICD with Notary Project, ORAS and Harbor

Yi Zha, Notary Project maintainer, Senior PM at Microsoft
Yan Wang, Harbor maintainer, Staff Engineer at VMware

Supply Chain risk: Time to Act!

742%

average annual increase in Software Supply Chain attacks

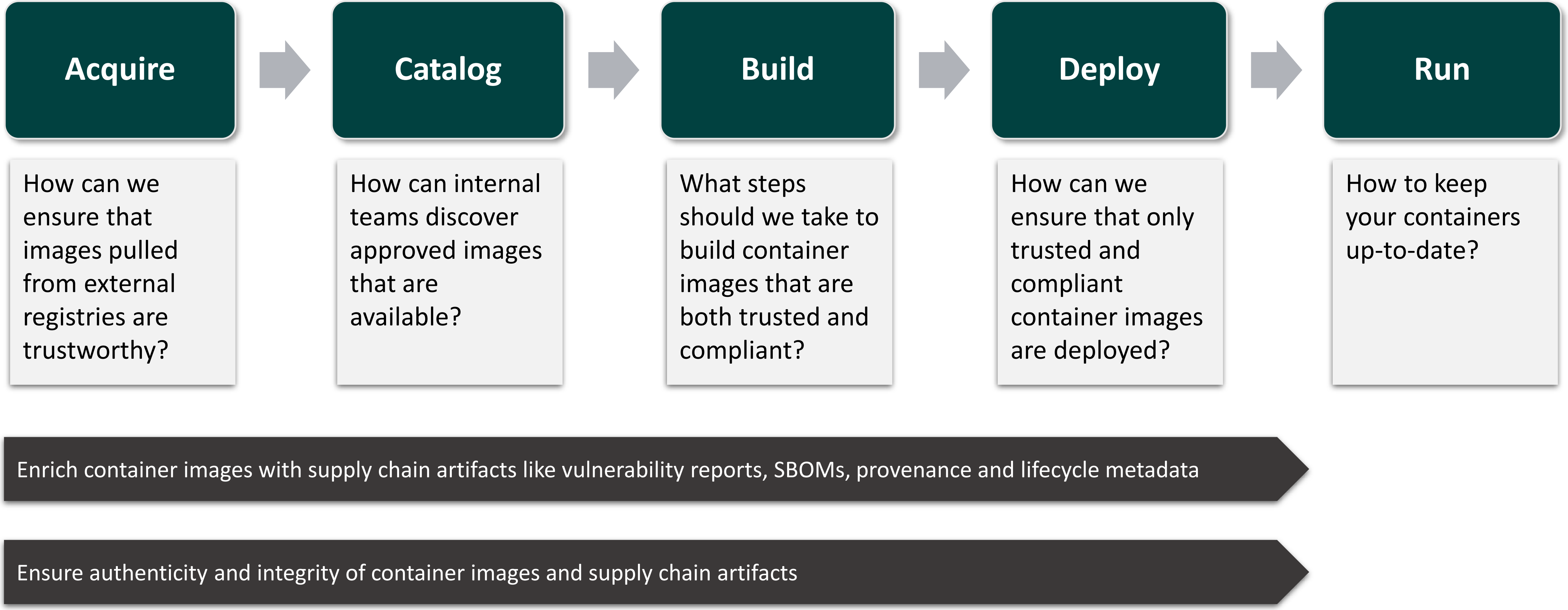
Source: Sonatype 2023 software supply chain report

87%

of container images have high or critical vulnerabilities

Source: Sysdig 2023 Cloud-Native Security and Usage Report

Secure your container supply chain



Acquire

How can we ensure that images pulled from external registries are trustworthy?

Catalog

How can internal teams discover approved images that are available?

Build

What steps should we take to build container images that are both trusted and compliant?

Deploy

How can we ensure that only trusted and compliant container images are deployed?

Run

How to keep your containers up-to-date?

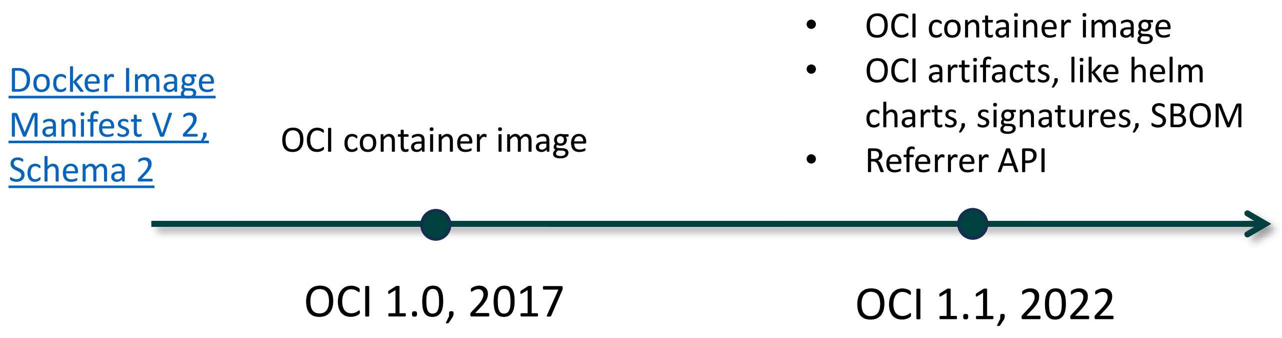
Enrich container images with supply chain artifacts like vulnerability reports, SBOMs, provenance and lifecycle metadata

Ensure authenticity and integrity of container images and supply chain artifacts

Open Container Initiative (OCI) – Content other than OCI images

The OCI is a Linux Foundation project that aims to establish open standards for container technology around image formats, runtimes, and distribution.

- OCI image specification
- OCI runtime specification
- OCI distribution specification



Docker Image Manifest V 2, Schema 2

OCI container image

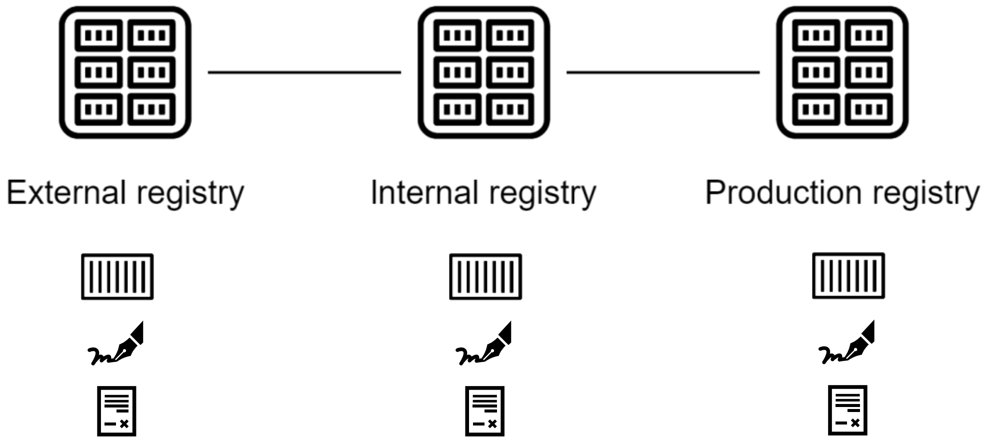
OCI 1.0, 2017

- OCI container image
- OCI artifacts, like helm charts, signatures, SBOM
- Referrer API

OCI 1.1, 2022

```
localhost:5000/net-monitor@sha256:e5e13b0c77cbb769548077
└── application/vnd.cncf.notary.signature
|     └── sha256:b5f3c7d27160b760ef07aac82a0d11e34fdb560f8
└── application/spdx+json
      └── sha256:6cbf7cc5ffa82b030b57ff820d49a86c143d8c6ac4
            └── application/vnd.cncf.notary.signature
                  └── sha256:7d27b760ef07aac82a0d15f3c160b1e34
```

OCI – Portability and interoperability


External registry

Internal registry

Production registry

Ensuring authenticity and integrity of artifacts across multi-cloud environments

ORAS – Manage and distribute OCI artifacts in OCI compliant registries

Attaching supply chain artifacts as referrers to the container image

```
$ oras attach localhost:5000/net-monitor:v1 spdx.json --artifact-type=application/spdx+json
```

Discover graph of artifacts

```
$ oras discover localhost:5000/net-monitor:v1 -o tree
```

```
localhost:5000/net-monitor@sha256:e5e13b0c77cbb769548077
└── application/vnd.cncf.notary.signature
|     └── sha256:b5f3c7d27160b760ef07aac82a0d11e34fdb560f8
└── application/spdx+json
      └── sha256:6cbf7cc5ffa82b030b57ff820d49a86c143d8c6ac4
            └── application/vnd.cncf.notary.signature
                  └── sha256:7d27b760ef07aac82a0d15f3c160b1e34
```

Move container images and referrers

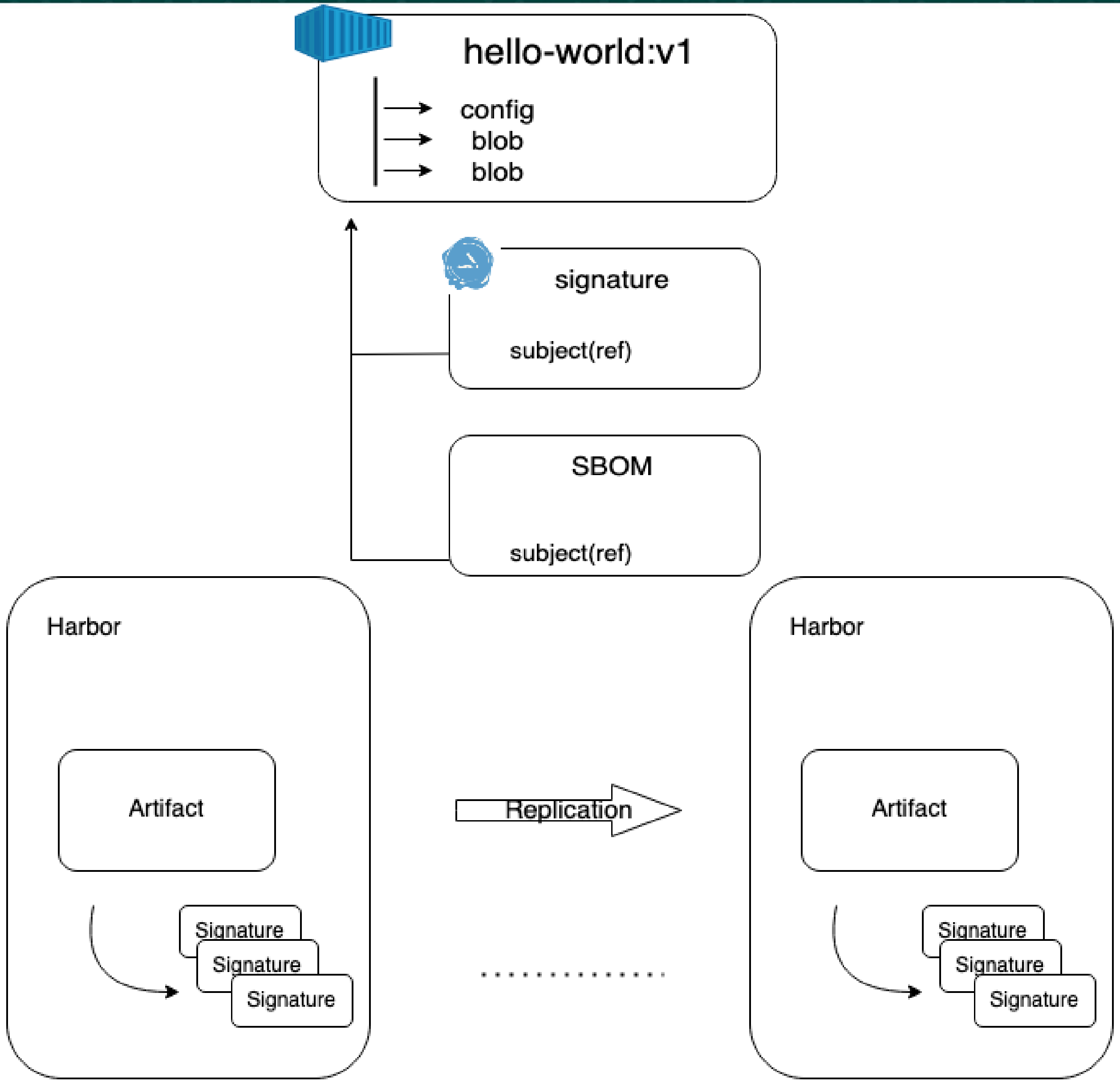
```
$ oras copy -r localhost:5000/net-monitor:v1 $REGISTRY/$PATH/net-monitor:v1
```

ORAS CLI and Go library are production ready! ORAS CLI v1.1.0 and ORAS go library v2.3.0 are compliant with OCI image-spec v1.1.0-rc4 and distribution-spec v1.1.0-rc3. Learn more from OCI Registry As Storage | OCI Registry As Storage (oras.land)

Harbor

Supporting OCI Distribution Spec v1.1.0

- Manage the linkage between subject artifacts and their accessories.
- Support the ability for any v1.1.0 compatible client to push an OCI artifact that references a subject artifact.
- Enable the replication of artifact accessories to another Harbor instance.



Notary Project - Ensuring authenticity and integrity of artifacts

The Notary Project is a set of specifications and tools intended to provide a cross-industry standard for securing software supply chains by using authentic container images and other OCI artifacts.

Main sub-projects of the Notary Project:

notary	This repository contains the source code for the server and the client of the initial TUF-based implementation circa 2016.
specifications	This repository contains the latest Notary Project requirements, scenarios, specifications, and security audits to overcome the challenges from the initial implementation of 2016.
notation	This repository contains the source code for the convenient CLI implementation of the new Notary Project specifications.
notation-go	This repository contains the source code for the convenient Golang library implementation of the new Notary Project signing and verification flow.
notation-core-go	This repository contains the source code for the Golang library implementation of the Notary Project signature (hereafter "Notary Project signature") specification and wrapping (COSE and JWS).

Learn more about Notary Project terms

Notary Project tooling Notation is ready for production use!

- ✓ **Seamless integration to existing PKI infrastructures**
- ✓ **Privacy and data compliance**
- ✓ **Signing keys are securely stored in KMS:** Azure, AWS, Hashicorp (WIP)
- ✓ **Signature formats:** JWS (JSON format) and COSE (binary format)
- ✓ **Signature portability:** Compatible with OCI v1.1
- ✓ **Extensibility:**
  - Bring your own plugins for signing and verification
  - More signing schema: CA, Signing authority
- ✓ **Fine tuned trust policies:** Don’t trust anyone by default. You specify who you trust and verification level
- ✓ **Revocation:** revoke an identity or a signature

Sign an image

```
$ notation sign $IMAGE --id <key_id> --plugin <name>
```

Verify an image

```
$ notation cert add --type ca –store mystore root_cert.pem
$ notation policy import mypolicy.json
$ notation verify $IMAGE
```

List signatures

```
$ notation list
```

Learn more at Notary Project website.

Demo

Workflow:	Build and push a container image in Harbor
	Sign the container image using Notary Project’s Notation CLI
	Generate SBOM for the container image
	Attach SBOM to the container image using ORAS CLI
	Sign the SBOM using Notary Project’s Notation CLI
	Generate the vulnerability report for the container image
	Attach vulnerability report to the container image with ORAS CLI
	Sign the vulnerability report using Notary Project’s Notation CLI
	View the graph of container images and associated artifacts from Harbor GUI
	Enable the signature checking policy in Harbor before pulling the image
	Use Docker CLI to pull the image

What’s next?

Standard

- OCI spec 1.1 GA

Integration

- Integrate with various client tooling with Harbor
- Notary Project tooling integration with popular CICD pipelines

More signing stories

- Sign and verify arbitrary data
- Sign container images before they are pushed to the registry

More Verification stories -- OSS [Ratify](), production ready on 9/26/2023

- Verification with OPA Policies in k8s
- Verifying images at container runtime (experimental)

Welcome to the community

- [ORAS]()
- [Harbor]()
- [Notary Project]()

Q&A