Container Images & Security 101: 5 Need To Know Facts

Phil Estes, Principal Engineer, AWS
CNCF Ambassador, containerd project maintainer



CLOUDNATIVE
SECURITYCON
NORTH AMERICA 2024

JUNE 26-27 | SEATTLE, WA #CNSCon





What is a **container image**?



Docker Image Specification, v2.2

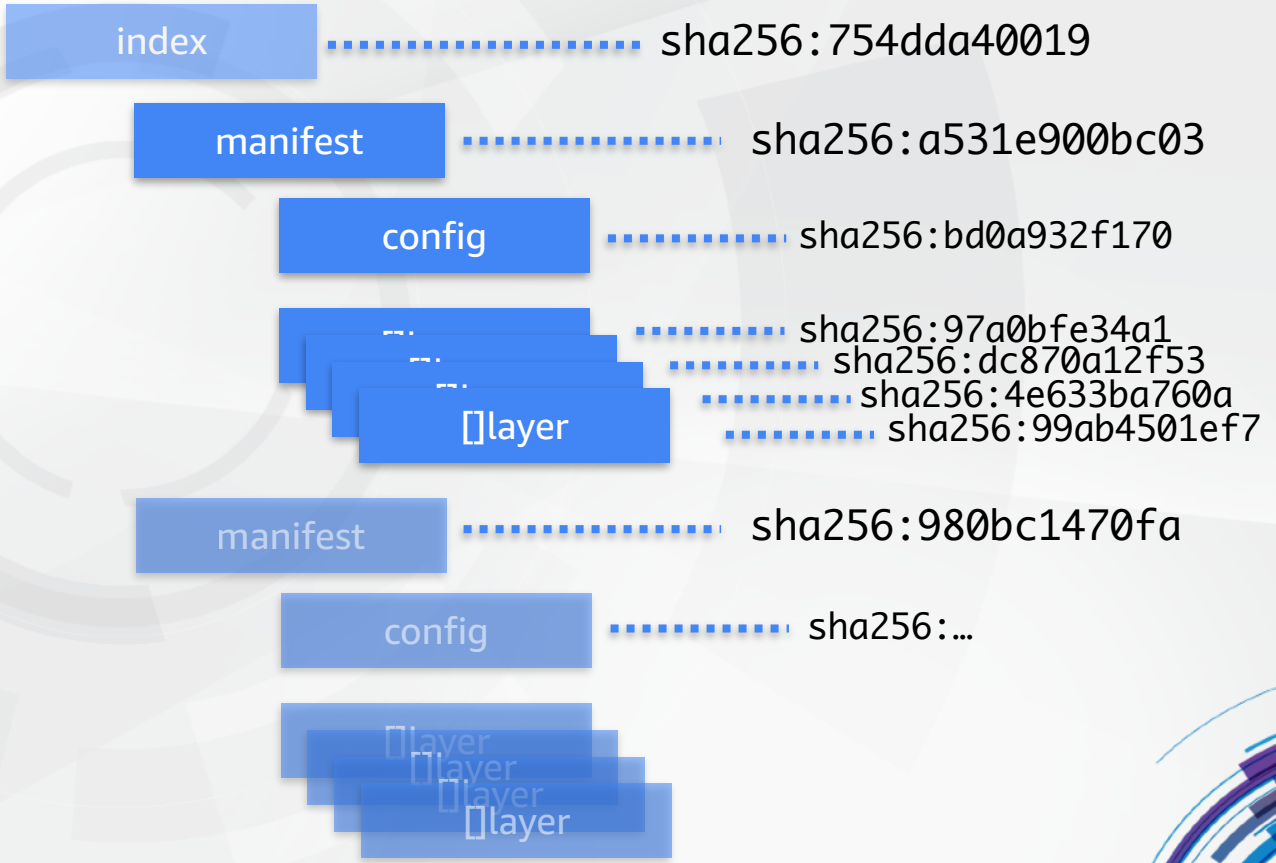


OCI Image Specification v1.1

Bottom line: a **container image** conforms to the OCI specification
***Adjacent terms**: Docker image*

1 Verifiable

A container image is a collection of objects in a ***DAG***, each having a **descriptor** with a **digest** (hash) for verification and a **mediaType** to distinguish how to process them



```
$ Demo time
```

/var/lib/containerd/io.containerd.content.v1.content/blobs/sha256/

```
# ls
-r--r--r-- 1 root root    1483 Jun 24 16:11 011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f
-r--r--r-- 1 root root     528 Jun 24 16:11 02c6e08a008a0a71bf5c6b13d87876f494213eb2880c23acb45c15859245baa6
-r--r--r-- 1 root root    1472 Jun 24 16:11 04f98b6119b460b9d3a59120c8f84e3e6dbd337cd37c7288185fde2bd06b2a43

# file 011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f
011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f: JSON data

# sha256sum 011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f
011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f  <repeated>

# json_pp < 011cd18df9954a4143ac1256824ad34026d382d26714a2b222d0a8f06286224f
{
   "architecture" : “arm”,
   “os” : “linux”,
   “variant” : “v6”,
   "config" : {
  …
}
```

2 Signable

Signing container images







- Formalizing the artifact definition (artifactType)
- Adding the subject field in image spec
- /v2/../referrers API in distribution spec

```
$ Demo time
```

3 Encryptable

https://github.com/containers/ocicrypt

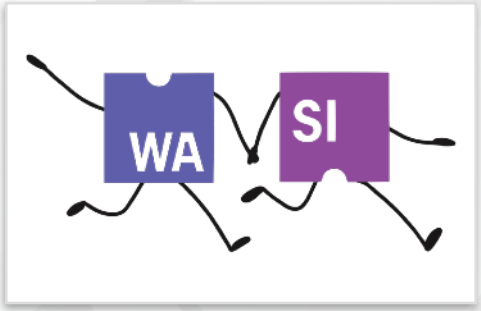
```
$ manifest-tool inspect localhost:5000/alpine:encrypted

Name:   localhost:5000/alpine:encrypted (Type: application/vnd.oci.image.index.v1+json)
Digest: sha256:24991a9edaf9c073ac8943f63d9ee146d0ab4ecd0e35442c575df1f325cb8cc3
 * Contains 1 manifest references (1 image, 0 attestation):
[1]     Type: application/vnd.oci.image.manifest.v1+json
[1]   Digest: sha256:914e95f61493f5593c9c4bf7ffb3193ae680547ac3c6fa04fa72cc20647ba13b
[1]   Length: 1762
[1] Platform:
[1]    -      OS: linux
[1]    -    Arch: arm64
[1]    - Variant: v8
[1] # Layers: 1
     layer 01: digest = sha256:d389647d19fe03f97825439ee24a5350e933436ba46a8eca63c0efbb1f0c8374
                 type = application/vnd.oci.image.layer.v1.tar+gzip+encrypted
```

```
$ Demo time
```

4 Extensible

Containerd “runwasi” shim



Using standard OCI image manifest

Uses a custom media type:
application/vnd.bytecodealliance.wasm.component.layer.v0+wasm

Homebrew package manager



Using “near” standard OCI image manifest

Significant use of annotations, read by the brew tooling

```
$ Demo time
```

https://github.com/opencontainers/image-spec/blob/main/manifest.md#guidelines-for-artifact-usage

Guidelines for Artifact Usage

Content other than OCI container images MAY be packaged using the image manifest. When this is done, the `config.mediaType` value MUST be set to a value specific to the artifact type or the empty value. If the `config.mediaType` is set to the empty value, the `artifactType` MUST be defined. If the artifact does not need layers, a single layer SHOULD be included with a non-zero size. The suggested content for an unused `layers` array is the empty descriptor.

The design of the artifact depends on what content is being packaged with the artifact. The decision tree below and the associated examples MAY be used to design new artifacts:

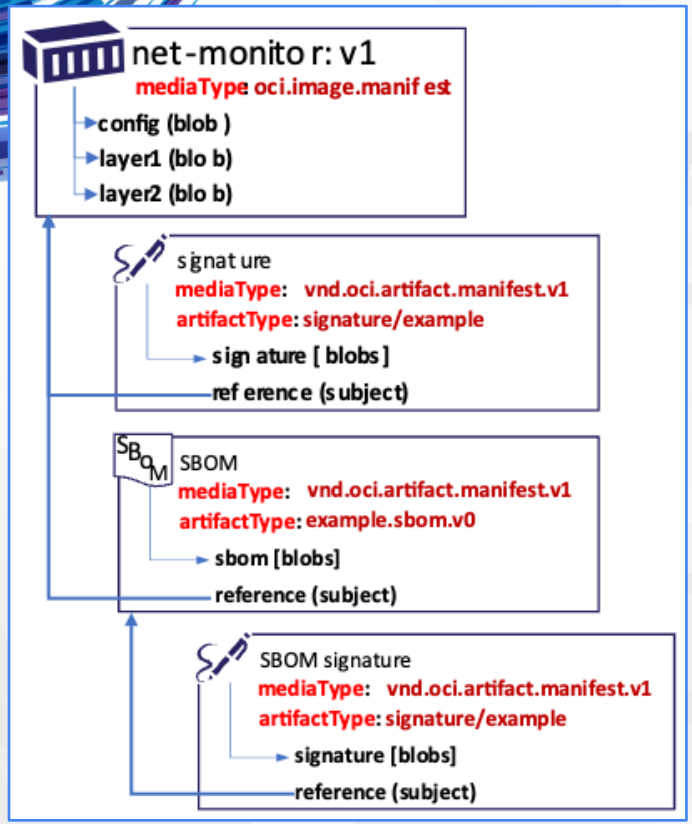
1. Does the artifact consist of at least one file or blob? If yes, continue to 2. If no, specify the `artifactType`, and set the `config` and a single `layers` element to the empty descriptor value. Here is an example of this with annotations included:

```
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.oci.image.manifest.v1+json",
  "artifactType": "application/vnd.example+type",
  "config": {
    "mediaType": "application/vnd.oci.empty.v1+json",
    "digest": "sha256:44136fa355b3678a1146ad16f7e8649e94fb4fc21fe77e8310c060f61caaff8a",
    "size": 2
  },
  "layers": [
    {
      "mediaType": "application/vnd.oci.empty.v1+json",
      "digest": "sha256:44136fa355b3678a1146ad16f7e8649e94fb4fc21fe77e8310c060f61caaff8a",
      "size": 2
    }
  ],
  "annotations": {
    "oci.opencontainers.image.created": "2023-01-02T03:04:05Z",
    "com.example.data": "payload"
```

OPEN CONTAINER INITIATIVE

Specifying how content *other than* standard images are stored in OCI registries

5 Integral to the SSP



SLSA SPDX® docker scout syft

Recent Advancements:

> BuildKit SBOM+attestation support
> Kubernetes working on SLSA levels
> Signed K8s releases (SBOMs available)
> SPDX standards progress
> OSS and commercial SBOM tooling

https://github.com/kubernetes/kube-state-metrics/issues/2282
https://docs.docker.com/build/attestations/
https://slsa.dev
https://spdx.dev

```
$ Demo time
```

Thank you!