Sign, Attest, and Verify! A practical guide for software supply chain security

Anushka Mittal

Vishal Choudhary

KubeDay India
December 8, 2023

About us

- Intern at Nirmata
- Kyverno Contributor

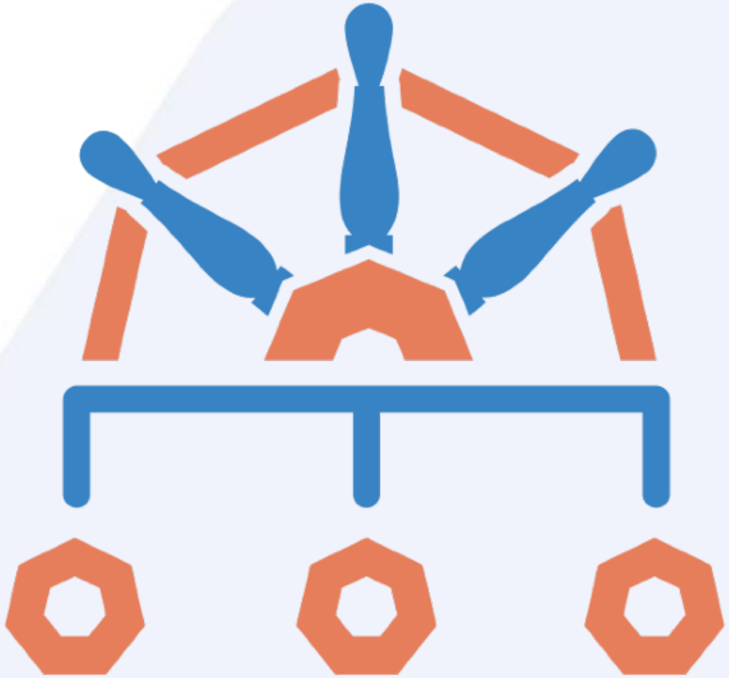
Vishal Choudhary

- Software Developer at Nirmata
- Kyverno Contributor

Anushka Mittal

Agenda

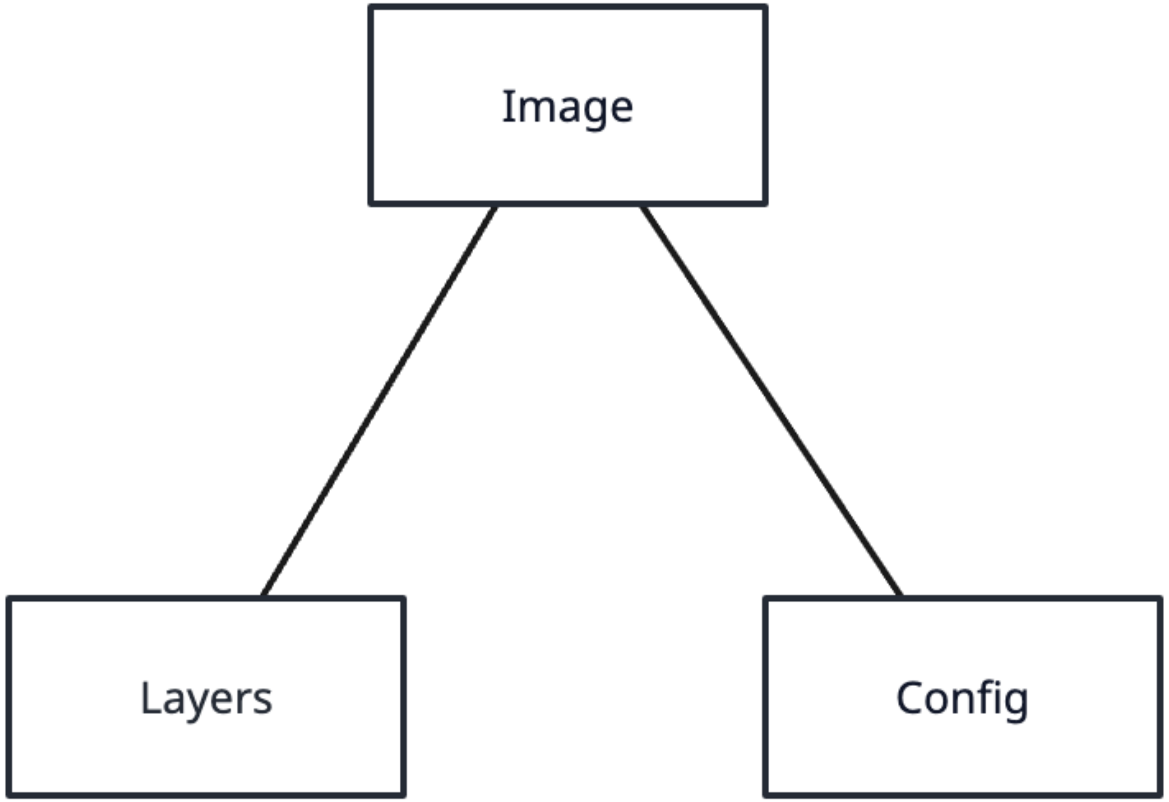
- Understanding Container images
- Image metadata
- Why sign and verify images?
- Notary/Cosign
- Kyverno and Image verification
- Demo



Understanding Container Images

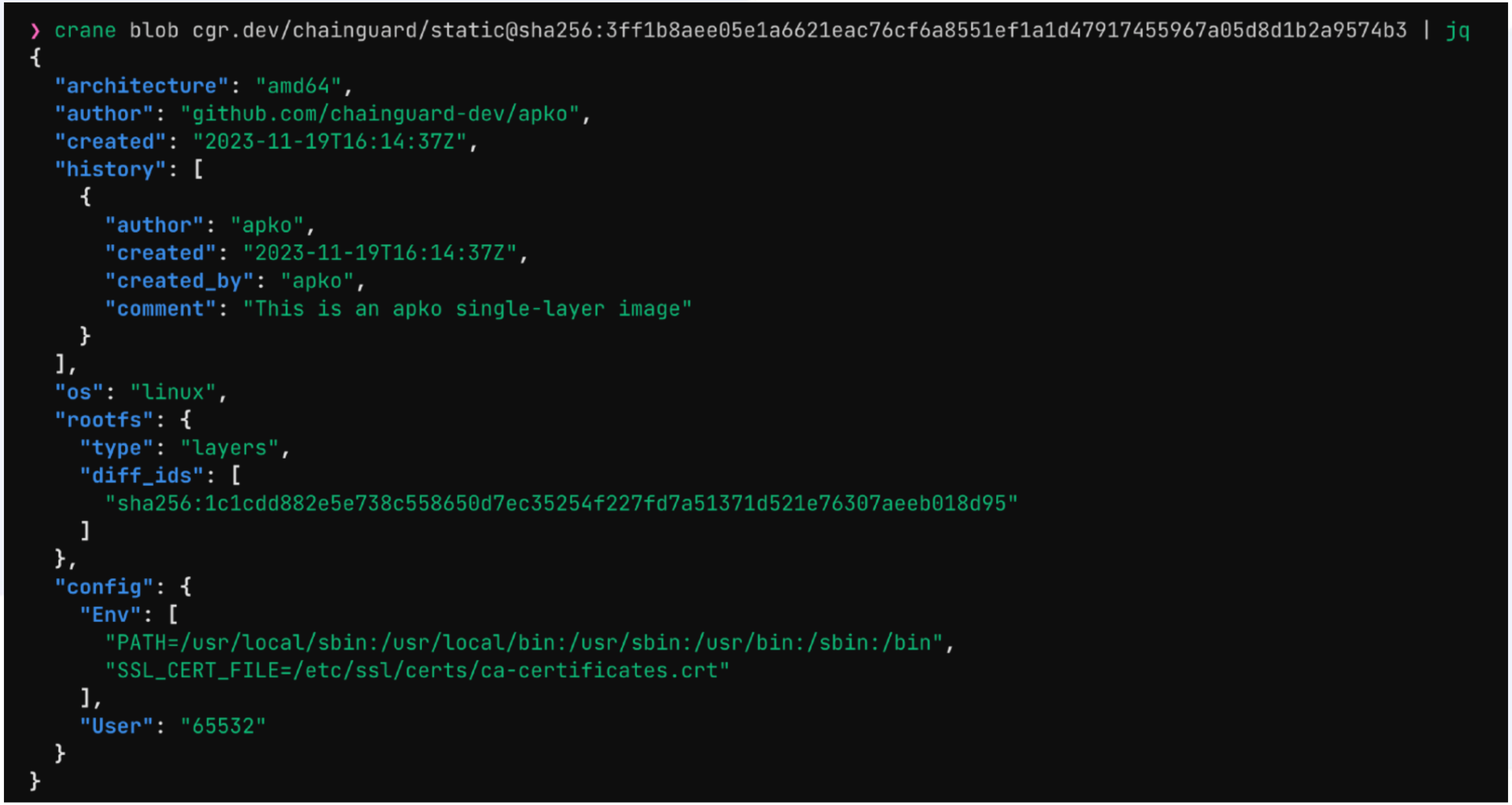
- Static file with executable code
- Used to create containers
- Stored in content addressable registries

Components of container Images



Container Image: Config

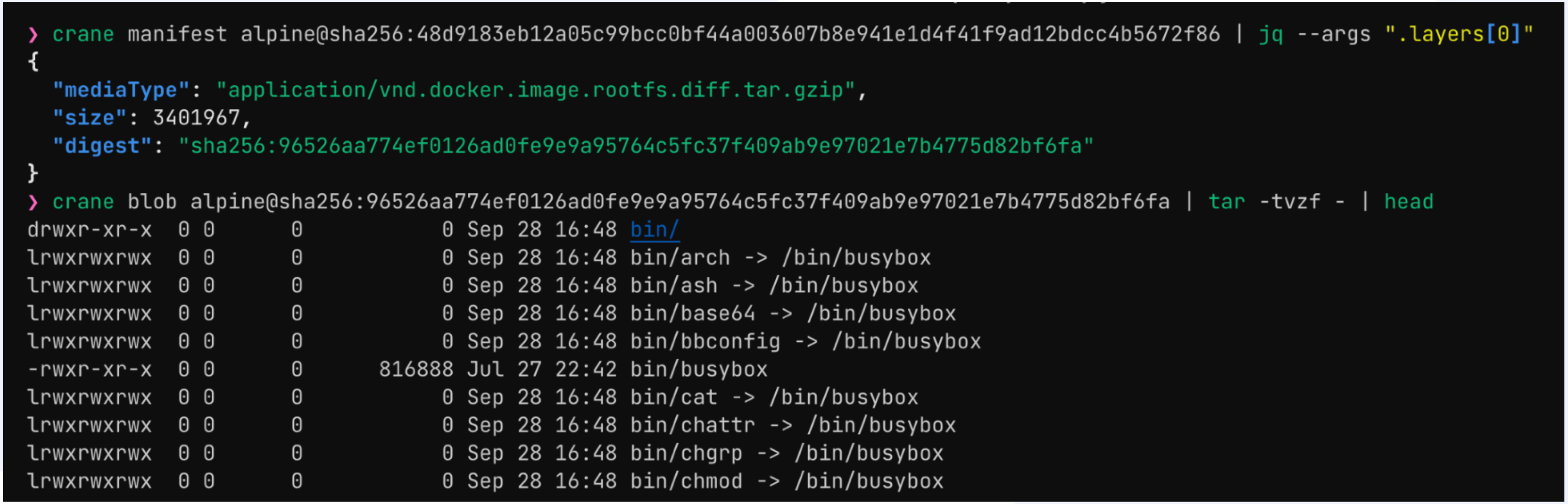
Config is a JSON data containing information about the image



```
❯ crane blob cgr.dev/chainguard/static@sha256:3ff1b8aee05e1a6621eac76cf6a8551ef1a1d47917455967a05d8d1b2a9574b3 | jq
{
  "architecture": "amd64",
  "author": "github.com/chainguard-dev/apko",
  "created": "2023-11-19T16:14:37Z",
  "history": [
    {
      "author": "apko",
      "created": "2023-11-19T16:14:37Z",
      "created_by": "apko",
      "comment": "This is an apko single-layer image"
    }
  ],
  "os": "linux",
  "rootfs": {
    "type": "layers",
    "diff_ids": [
      "sha256:1c1cdd882e5e738c558650d7ec35254f227fd7a51371d521e76307aeeb018d95"
    ]
  },
  "config": {
    "Env": [
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
      "SSL_CERT_FILE=/etc/ssl/certs/ca-certificates.crt"
    ],
    "User": "65532"
  }
}
```

Container Image: Layers

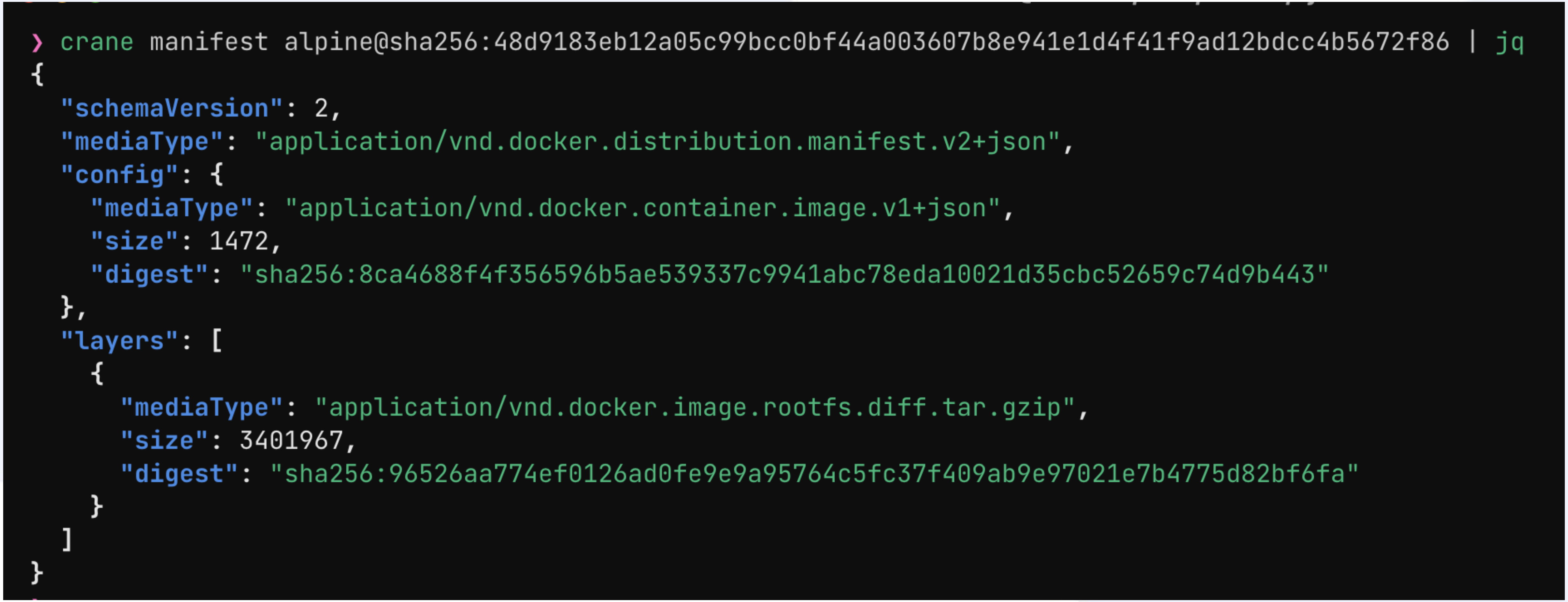
Layers are chunks of the container image. It can be of any media type.



```
❯ crane manifest alpine@sha256:48d9183eb12a05c99bcc0bf44a003607b8e941e1d4f41f9ad12bdcc4b5672f86 | jq --args ".layers[0]"
{
  "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
  "size": 3401967,
  "digest": "sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa"
}
❯ crane blob alpine@sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa | tar -tvzf - | head
drwxr-xr-x  0 0      0           0 Sep 28 16:48 bin/
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/arch -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/ash -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/base64 -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/bbconfig -> /bin/busybox
-rwxr-xr-x  0 0      0      816888 Jul 27 22:42 bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/cat -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/chattr -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/chgrp -> /bin/busybox
lrwxrwxrwx  0 0      0           0 Sep 28 16:48 bin/chmod -> /bin/busybox
```

Container Image: Manifest

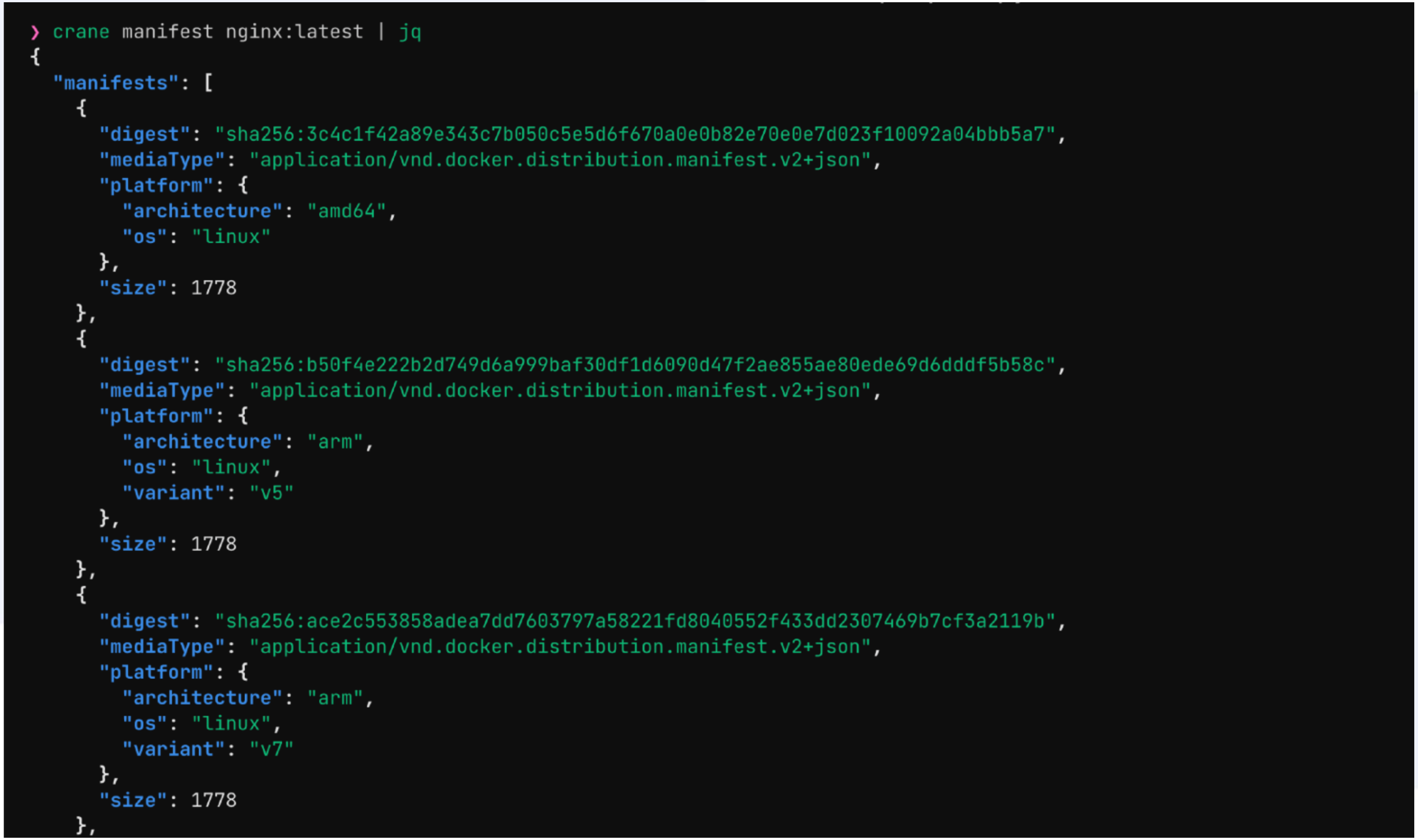
Manifest contains the information of all the components of an image.



```
❯ crane manifest alpine@sha256:48d9183eb12a05c99bcc0bf44a003607b8e941e1d4f41f9ad12bdcc4b5672f86 | jq
{
  "schemaVersion": 2,
  "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
  "config": {
    "mediaType": "application/vnd.docker.container.image.v1+json",
    "size": 1472,
    "digest": "sha256:8ca4688f4f356596b5ae539337c9941abc78eda10021d35cbc52659c74d9b443"
  },
  "layers": [
    {
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
      "size": 3401967,
      "digest": "sha256:96526aa774ef0126ad0fe9e9a95764c5fc37f409ab9e97021e7b4775d82bf6fa"
    }
  ]
}
```

Container Image: Manifest List

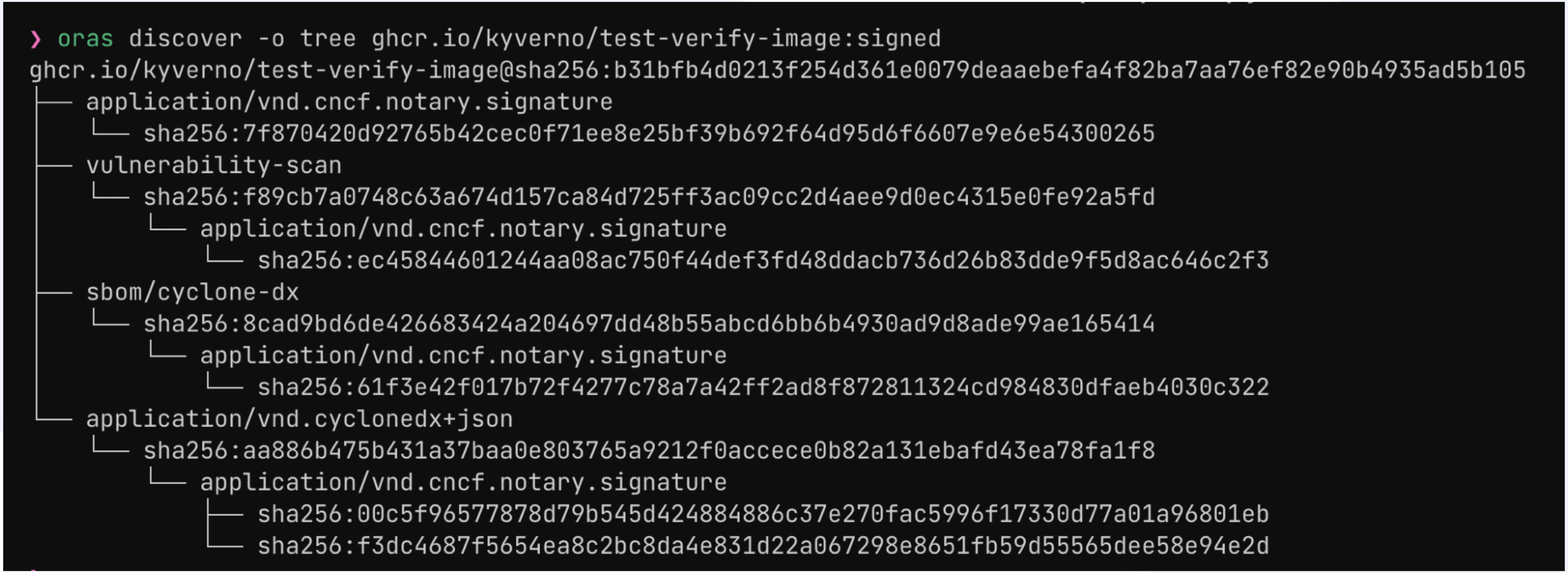
Manifest List is the manifest of all manifest for different platforms.



```
❯ crane manifest nginx:latest | jq
{
  "manifests": [
    {
      "digest": "sha256:3c4c1f42a89e343c7b050c5e5d6f670a0e0b82e70e0e7d023f10092a04bbb5a7",
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "platform": {
        "architecture": "amd64",
        "os": "linux"
      },
      "size": 1778
    },
    {
      "digest": "sha256:b50f4e222b2d749d6a999baf30df1d6090d47f2ae855ae80ede69d6dddf5b58c",
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "platform": {
        "architecture": "arm",
        "os": "linux",
        "variant": "v5"
      },
      "size": 1778
    },
    {
      "digest": "sha256:ace2c553858adea7dd7603797a58221fd8040552f433dd2307469b7cf3a2119b",
      "mediaType": "application/vnd.docker.distribution.manifest.v2+json",
      "platform": {
        "architecture": "arm",
        "os": "linux",
        "variant": "v7"
      },
      "size": 1778
    },
```

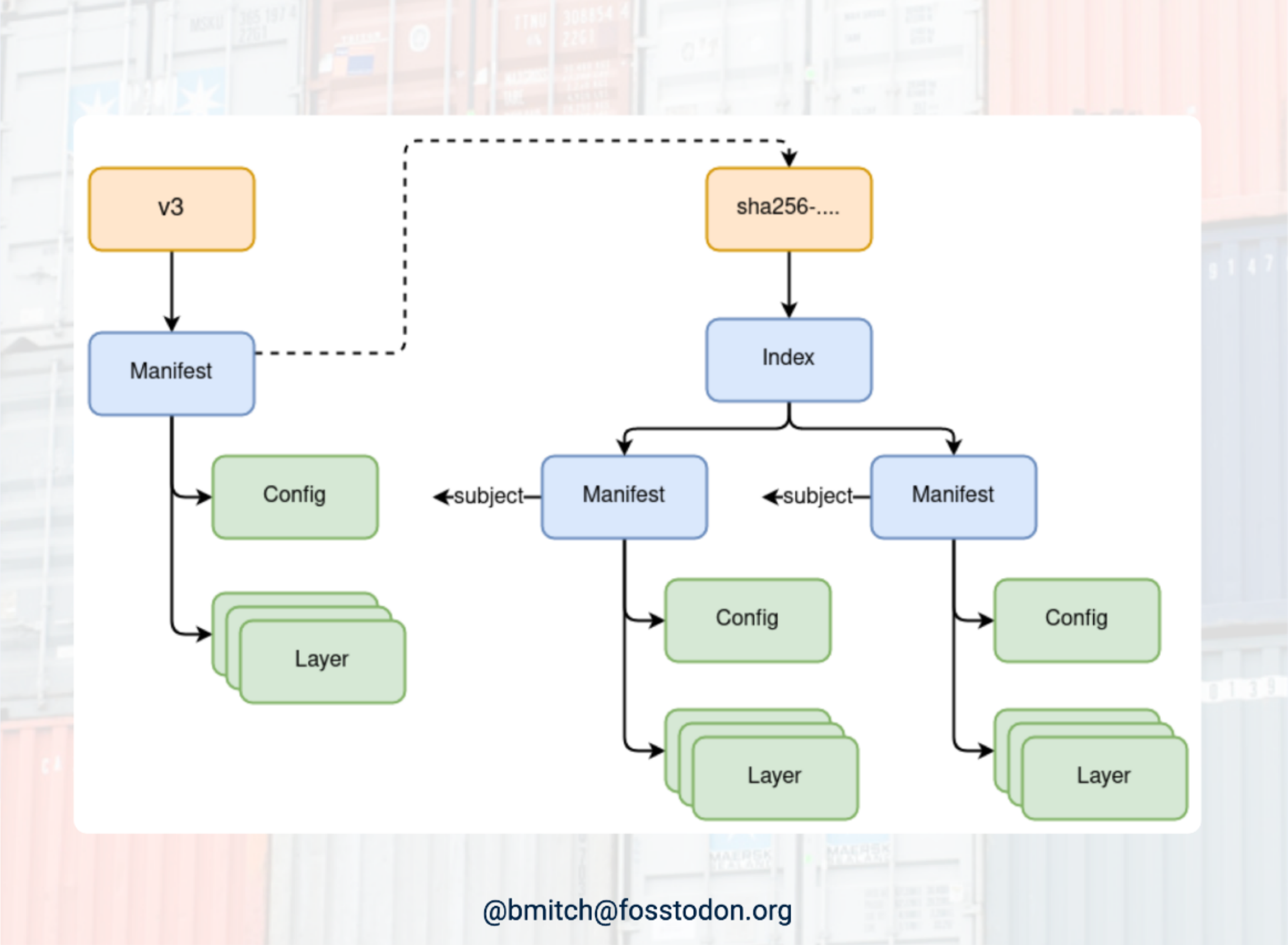
Container Image: Referrers

Referrers are a list of manifests with a subject relationship to a specified digest. This can be accessed using referrers API



```
❯ oras discover -o tree ghcr.io/kyverno/test-verify-image:signed
ghcr.io/kyverno/test-verify-image@sha256:b31bfb4d0213f254d361e0079deaaebefa4f82ba7aa76ef82e90b4935ad5b105
├── application/vnd.cncf.notary.signature
│   └── sha256:7f870420d92765b42cec0f71ee8e25bf39b692f64d95d6f6607e9e6e54300265
├── vulnerability-scan
│   └── sha256:f89cb7a0748c63a674d157ca84d725ff3ac09cc2d4aee9d0ec4315e0fe92a5fd
│       └── application/vnd.cncf.notary.signature
│           └── sha256:ec45844601244aa08ac750f44def3fd48ddacb736d26b83dde9f5d8ac646c2f3
├── sbom/cyclone-dx
│   └── sha256:8cad9bd6de426683424a204697dd48b55abcd6bb6b4930ad9d8ade99ae165414
│       └── application/vnd.cncf.notary.signature
│           └── sha256:61f3e42f017b72f4277c78a7a42ff2ad8f872811324cd984830dfaeb4030c322
└── application/vnd.cyclonedx+json
    └── sha256:aa886b475b431a37baa0e803765a9212f0accece0b82a131ebafd43ea78fa1f8
        └── application/vnd.cncf.notary.signature
            ├── sha256:00c5f96577878d79b545d424884886c37e270fac5996f17330d77a01a96801eb
            └── sha256:f3dc4687f5654ea8c2bc8da4e831d22a067298e8651fb59d55565dee58e94e2d
```

Summary



Credit: Modifying the Immutable: Attaching Artifacts to OCI Images - Brandon Mitchell: link

Image metadata

- Vulnerabilities and Vulnerability scanning
  - What are vulnerabilities?
  - Why do we care about vulnerabilities?
  - What is vulnerability scanning?

Image metadata

- Vulnerabilities and Vulnerability scanning.
  - What are vulnerabilities?
  - Why do we care about vulnerabilities?
  - What is vulnerability scanning?
- SBOMs and their lifecycle

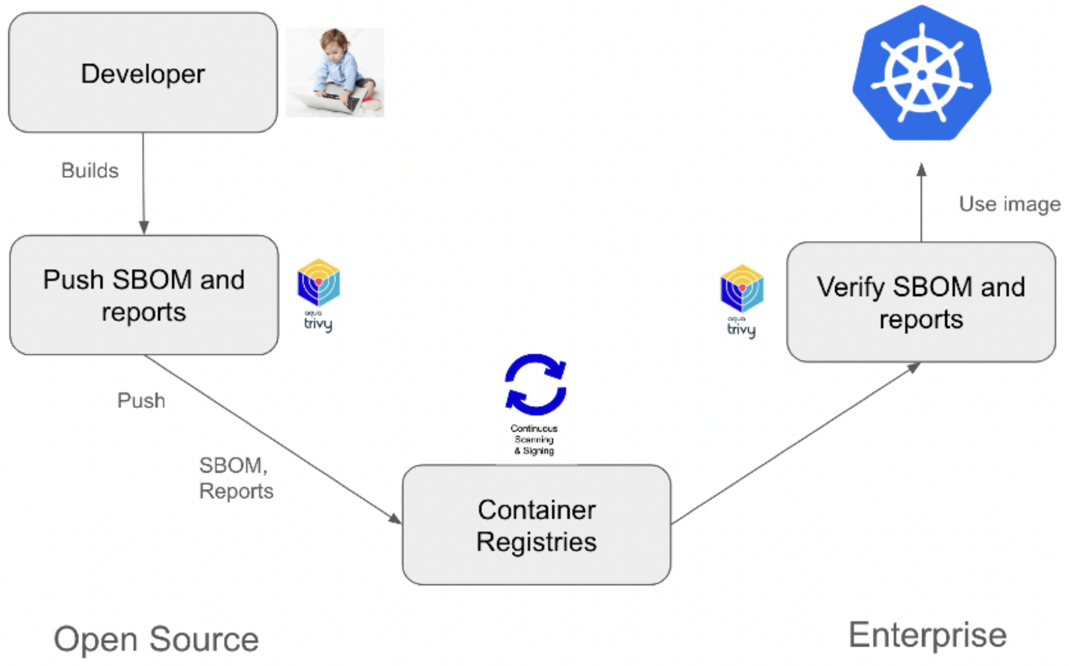


Image metadata

- Vulnerabilities and Vulnerability scanning
  - What are vulnerabilities?
  - Why do we care about vulnerabilities?
  - What is vulnerability scanning?
- SBOMs and their lifecycle
- Problems
  - Storing metadata in image
  - Trusting metadata

Sign and Verify Images

- Image security is the most important component of container security
- Ensures that images are from the actual developer and hasn't been tampered with.
- Builds integrity and creates a trusted environment.
- Ensures that no bad actor can access the cluster.
- **How do we sign our images?**

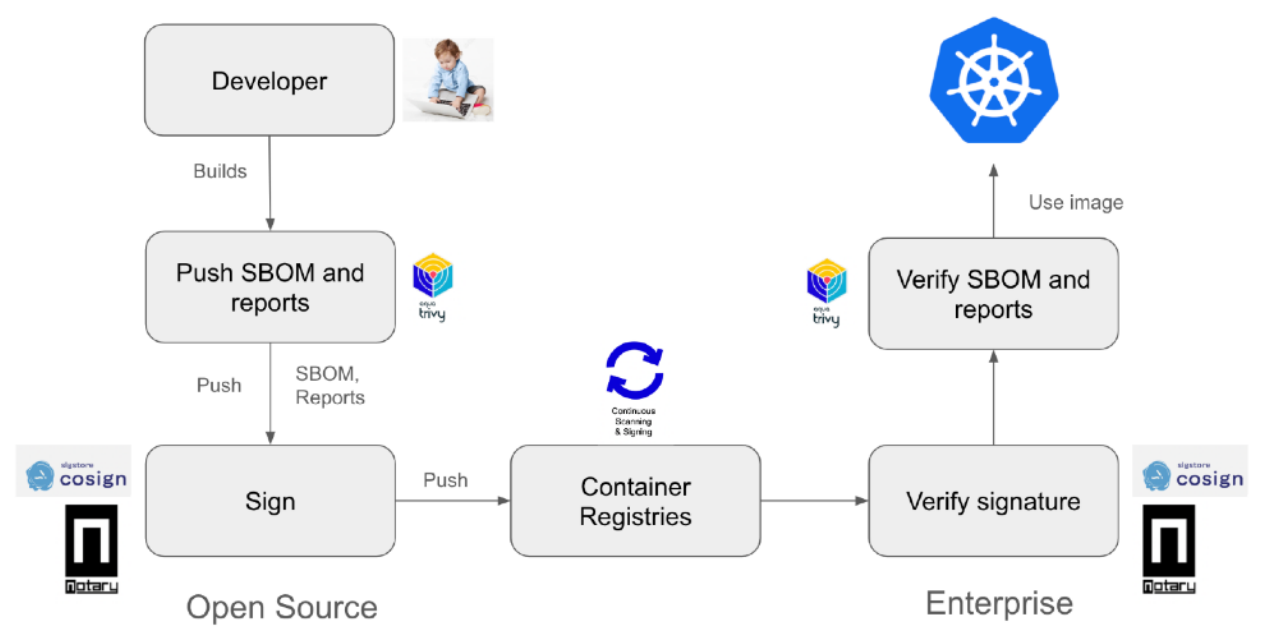


Image signing and verification options

Notary

- Uses x509 key-cert pair for signing and verification
- Supports only key based signing and verification for images and attestations
- Supports OCI 1.1 attestations
- Notary Project 1.0.0 released!

Cosign

- Supports Keyed and Keyless based signing and verification
- Supports in-toto attestations
- Verifies signature timestamps receipts
- Stores and monitors a transparency log of signatures to detect malicious activity
- Supports private infrastructure using TUF
- Cosign 2.0 released!

What is Kyverno

- Kyverno is a CNCF incubating project
- “Govern” in Greek
- Policy engine designed for Kubernetes
- Admission controller and more!
- Native to Kubernetes (no new programming language required)
- 2.5B+ downloads, 3000+ community members and adopters

Kyverno for Image verification

- Verify image signatures (Cosign and Notary)
- Verify attestations
- Any OCI registry
- OCI 1.1 referrers API support
- Supports Key (and KMS), keyless and certs
- Decision caching
- Multiway checks (any, all, atleast)

Demo

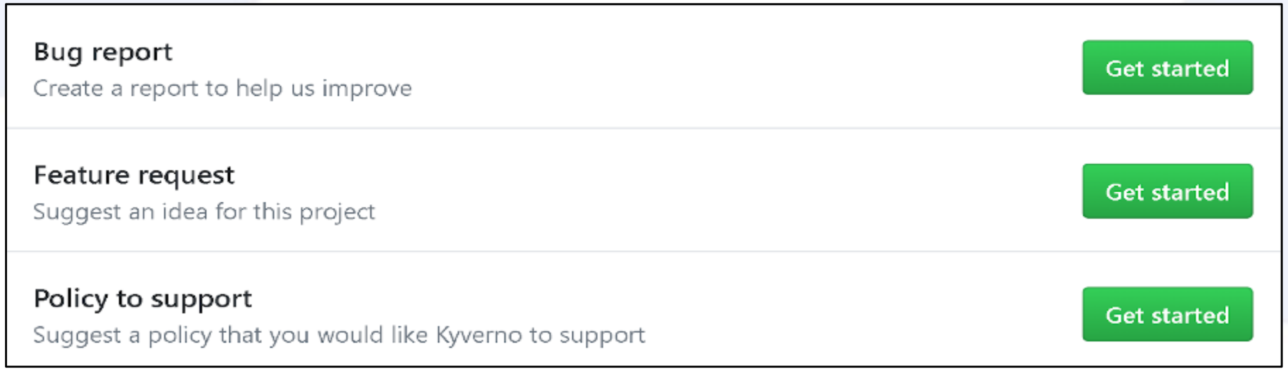
For Cosign and Notary:

- Image Verification
- Attestation Verification

Join the Kyverno Community

- Kyverno docs & samples: **https://kyverno.io**
- Slack Channel: **https://slack.k8s.io/#kyverno**
- Monthly community meetings
- Weekly contributor meetings

Join **https://groups.google.com/g/kyverno**



Bug report
Create a report to help us improve

Feature request
Suggest an idea for this project

Policy to support
Suggest a policy that you would like Kyverno to support

Reach out to us!

- Intern at Nirmata
- Kyverno Contributor

Vishal Choudhary
X: @vishalwastaken
GitHub: vishal-chdhry
Slack: vishal-chdhry

- Software Developer at Nirmata
- Kyverno Contributor

Anushka Mittal
X: @_Anushkamittal
GitHub: anushkamittal2001
Slack: Anushka Mittal

Thank you!

Resources

- Modifying the Immutable: Attaching Artifacts to OCI Images
- Kyverno overview
- Improve Vulnerability Management with OCI Artifacts -It Is That Easy!