Syft Generate software SBOMs

START HERE

`syft <image>`
Package table for an image

`syft <src> -o json=<file>`
Full Syft-native SBOM

`syft dir:<path>`
Scan a project directory

`syft <image> --scope all-layers`
Include packages from every layer

`syft <src> -o spdx-json=<file>`
Write an SPDX SBOM

`syft convert <sbom> -o <fmt>`
Change SBOM format

`syft <src> -o cyclonedx-json=<file>`
Write a CycloneDX SBOM

`syft attest -o <fmt> <image>`
Signed SBOM attestation

Source image, dir, archive, file → `catalogers` → **Syft** finds packages → `-o format` → **SBOM** SPDX, CycloneDX, JSON → `grype / attest` → **Scan or sign** Grype, Trivy, Cosign

From source to SBOM

CONCEPTS

Syft only lists what it can find in the source: packages and files, not vulnerabilities. Feed the SBOM to Grype or Trivy to scan it.

With no scheme, Syft tries the Docker daemon, then Podman, then containerd, then pulls from the registry. Pin it with --from or a scheme prefix.

Table is the default output but loses detail. The Syft JSON format (-o json) keeps everything and can be converted later.

SOURCES

`syft <image>`
Scan an image (daemon, else registry)

`syft dir:<path>`
Scan a directory

`syft <path>`
Directory, archive or SIF by path

`syft registry:<image>`
Pull from registry, skip the daemon

`syft docker:<image>`
Use the Docker daemon only

`syft podman:<image>`
Use the Podman daemon only

`syft --from <source> <target>`
Choose the source type explicitly

Same list as the prefixes, e.g. --from registry or --from docker-archive.

`syft docker-archive:<tar>`
Tarball from docker save

`syft oci-archive:<tar>`
OCI archive tarball

`syft oci-dir:<dir>`
OCI layout directory

`syft file:<file>`
Single file (e.g. a binary)

`syft --from containers-storage <image>`
Podman or CRI-O local storage

`syft sbom:<file>`
Read an existing SBOM as the source

SOURCES (cont.)

`syft <image> --platform <os/arch>`
Pick a platform of a multi-arch image

OUTPUT

`syft <src> -o json`
Syft JSON, all details

`syft <src> -o spdx-json`
SPDX as JSON

`syft <src> -o cyclonedx-json`
CycloneDX as JSON

`syft <src> -o <fmt>=<file>`
Write that format to a file

`syft <src> -o <fmt1>=<f1> -o <fmt2>=<f2>`
Several formats in one scan

`syft <src> -o spdx-tag-value`
SPDX tag-value text

`syft <src> -o cyclonedx-xml`
CycloneDX as XML

`syft <src> -o spdx-json@2.2`
Pin the format version

`syft <src> -o table`
Human-readable table (default)

`syft <src> -o purls`
One package URL per line

`syft <src> -o template -t <file.tmpl>`
Custom Go template output

`SYFT_FORMAT_PRETTY=true`
Indent JSON output

Without =<file> the SBOM goes to stdout; with it, nothing but logs. Handy: syft <src> -o json | jq ...

SCOPE AND FILTERING

`syft <image> --scope squashed`
Final filesystem only (default)

`syft <image> --scope all-layers`
Packages from every layer

`syft <src> --exclude '<glob>'`
Skip paths matching a glob

SCOPE AND FILTERING (cont.)

`syft <src> --exclude '**/node_modules/**'`
Ignore dependency folders

`syft <src> --select-catalogers <expr>`
Add or drop catalogers by name or tag

For example --select-catalogers="+sbom-cataloger,-rpm"; check names with syft cataloger list.

`syft cataloger list`
List available catalogers

`syft <src> --enrich all`
Fetch extra data from online sources

`syft <src> --source-name <n> --source-version <v>`
Name the described subject

CONVERT AND ATTEST

`syft convert <sbom> -o <fmt>`
Convert an SBOM to another format

`syft convert <sbom> -o <fmt>=<file>`
Convert into a file

`syft convert - -o spdx-json`
Convert an SBOM read from stdin

`syft attest -o <fmt> <image>`
SBOM as in-toto attestation, keyless

Needs cosign on PATH and push access; the image must be in a registry.

`syft attest --key cosign.key -o <fmt> <image>`
Attest with a local key

Conversion is marked experimental and can lose data, because the formats do not hold the same fields. Keep the Syft JSON as the source of truth.

Attest accepts exactly one output format (syft-json, cyclonedx-json, spdx-json or spdx-tag-value) and only registry images.

CONFIG AND CLI

`syft --version`
Print the version

`syft -c <file> <src>`
Use a config file

`syft <src> -v / -vv`
Verbose / debug logging

`syft <src> -q`
Quiet: suppress logging

`SYFT_<OPTION>=<value>`
Set any option by environment variable

Nested config keys are joined with underscores, e.g. SYFT_SCOPE=all-layers.

`.syft.yaml`
Config file in the working directory

Private registries use your Docker credentials (docker login) when Syft pulls with the registry source.

Install: brew install syft, or curl -sSfL https://get.anchore.io/syft | sudo sh -s -- -b /usr/local/bin

RECIPES

SBOM of an image, then scan it

```
syft <image> -o
json=sbom.syft.json
grype sbom:sbom.syft.json
```

Several formats from one scan

```
syft dir:. \
  -o spdx-json=sbom.spdx.json \
  -o cyclonedx-
json=sbom.cdx.json
```

Everything in every layer, no test files

```
syft <image> --scope all-layers
\
  --exclude '**/test/**' -o
table
```

Sign and verify an SBOM attestation

```
syft attest --key cosign.key -o
spdx-json <image> > att.json
cosign attach attestation --
attestation att.json <image>
cosign verify-attestation --key
cosign.pub --type spdxjson
<image>
```