FROM ze:ro to kind: Hero

@rusdacent

DISCLAIMER

- Это не доклад
- Лимитов нет
- Опыт передаём, вопросы задаём
- Давайте жить дружно ©™®

whoami

- DCG#7812

whoami

- DCG#7812
- B4CKSP4CE

whoami

- DCG#7812
- B4CKSP4CE
- SPb Reliability Meetup

whoami

- DCG#7812
- B4CKSP4CE
- SPb Reliability Meetup
- Luntry

whoami

- DCG#7812
- B4CKSP4CE
- SPb Reliability Meetup
- Luntry
- Технологический Болт Генона

С чего начинается образ?

С чего начинается образ?

Кто писал свой Dockerfile?

С чего начинается образ?

Без какой инструкции невозможен Dockerfile?

С чего начинается образ?

FROM ze:ro

Base image

Откуда берём?

Base image

Откуда берём?

- DockerHub, ghcr.io (GitHub), quay.io (RedHat),...

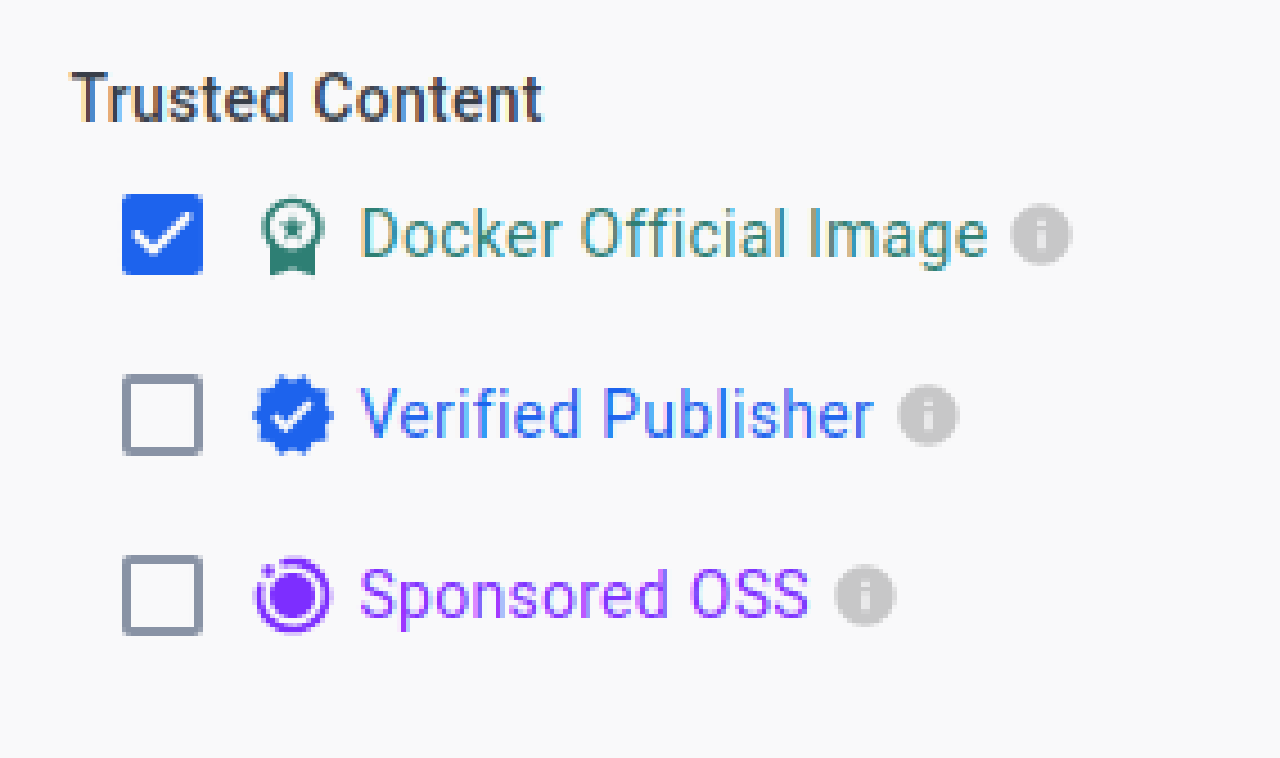
Base image

Откуда берём?

- DockerHub, ghcr.io (GitHub), quay.io (RedHat),...
- Собираем сами

DockerHub

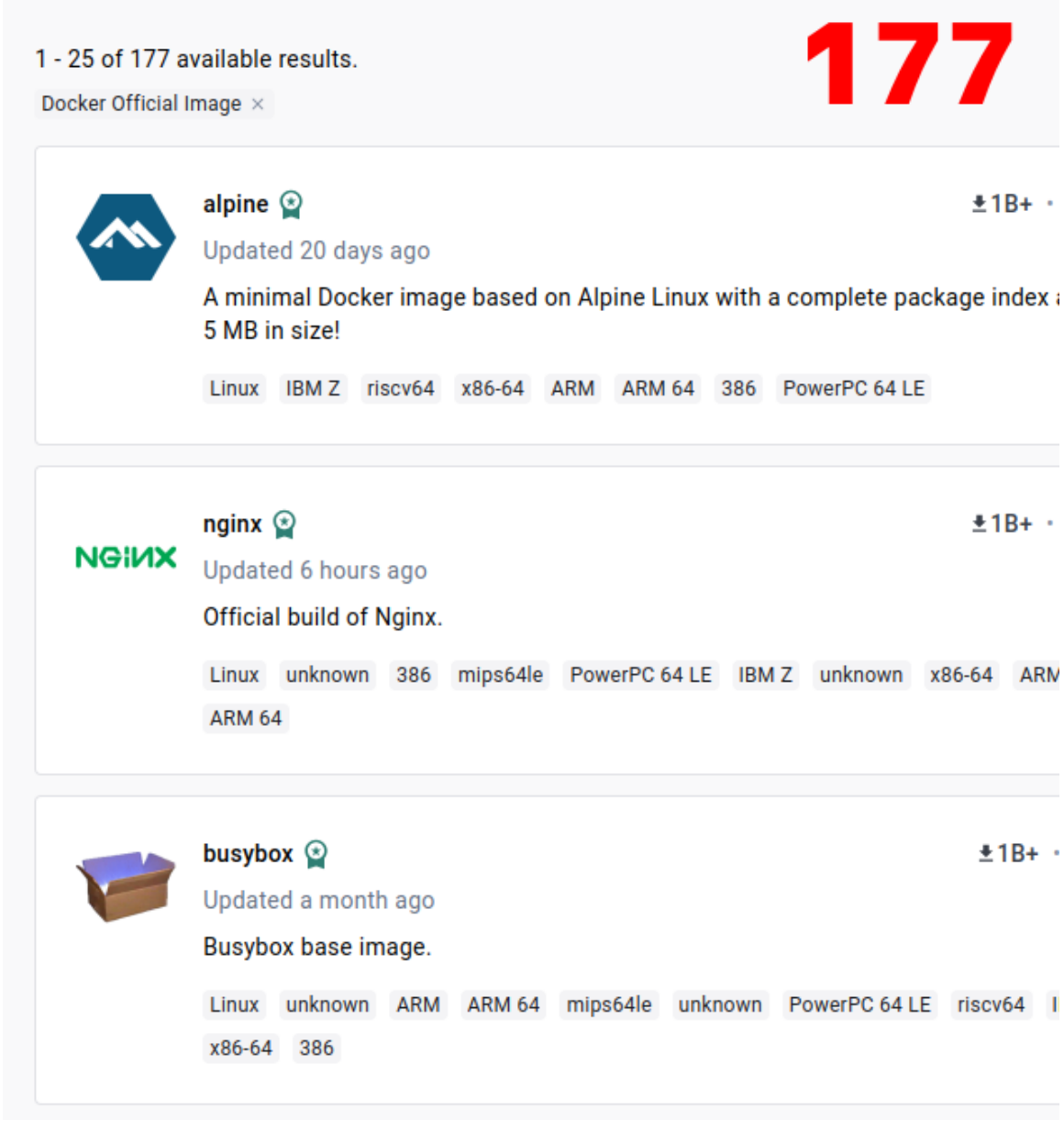
Каким доверяем?



DockerHub

Official

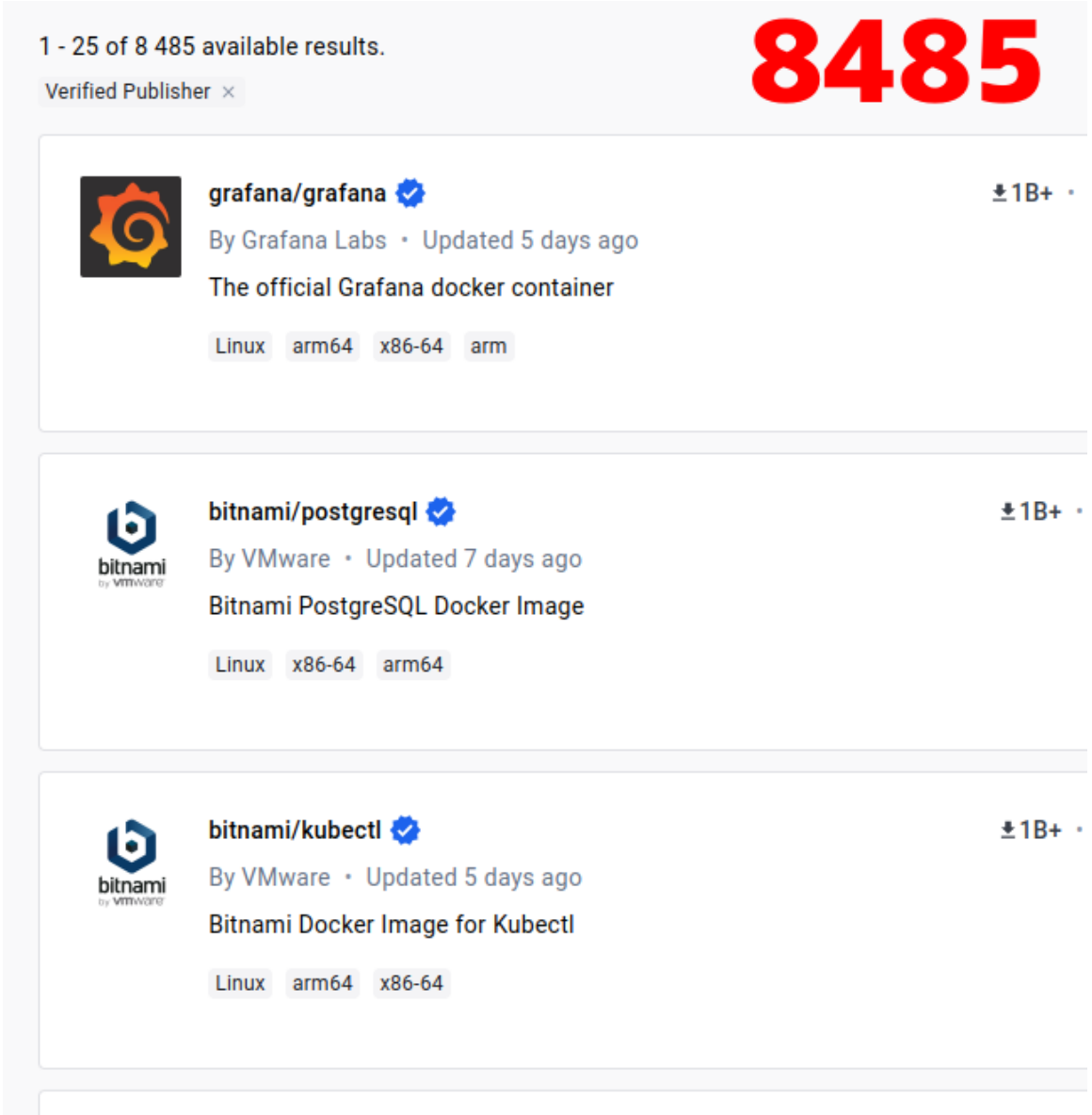
> Docker Official Images are a curated set of Docker open source and "drop-in" solution repositories.



DockerHub

Verified Publisher

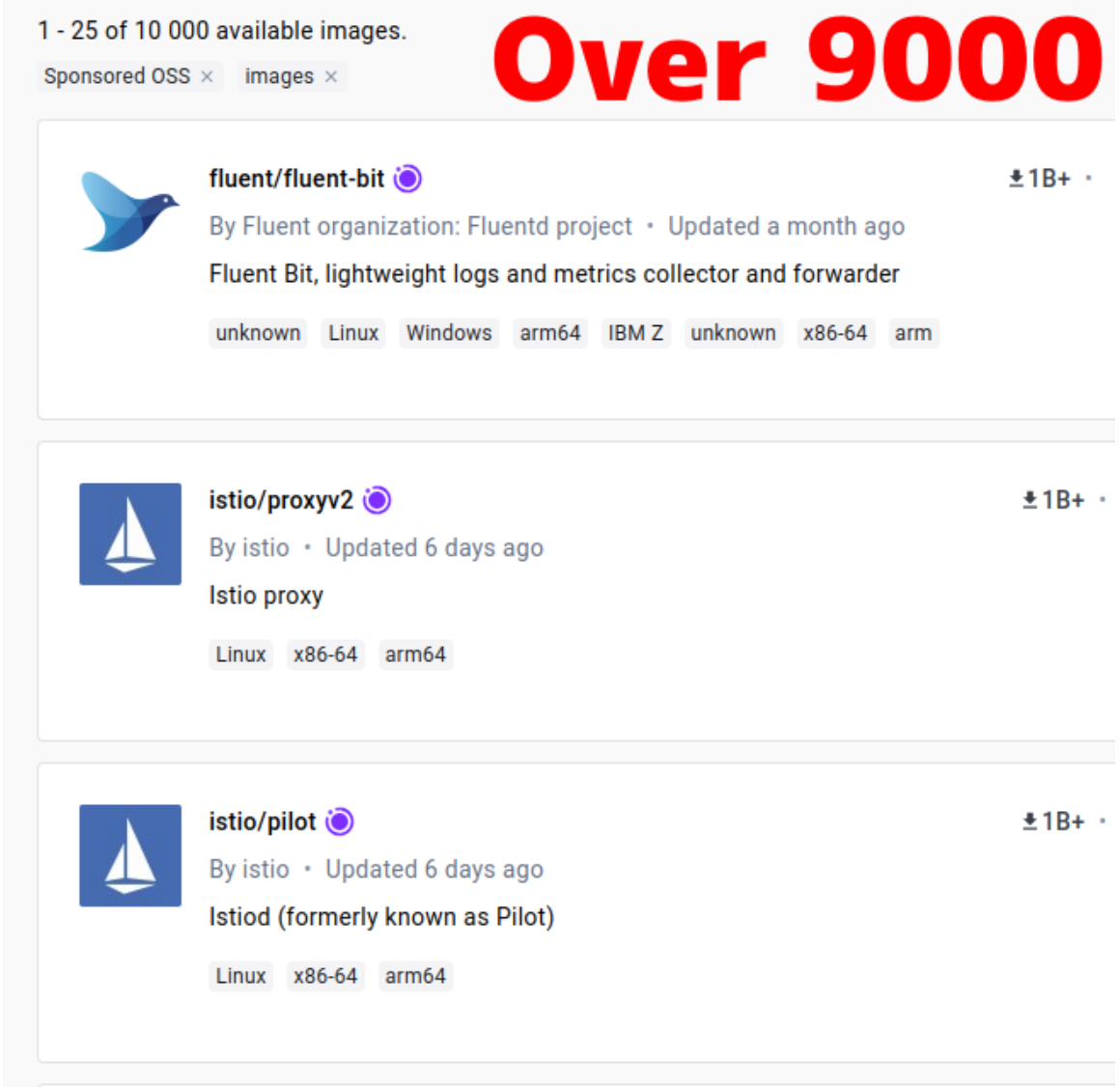
> High-quality images from publishers verified by Docker. These products are published and maintained directly by a commercial entity. These images are not subject to rate limiting.



DockerHub

Sponsored OSS

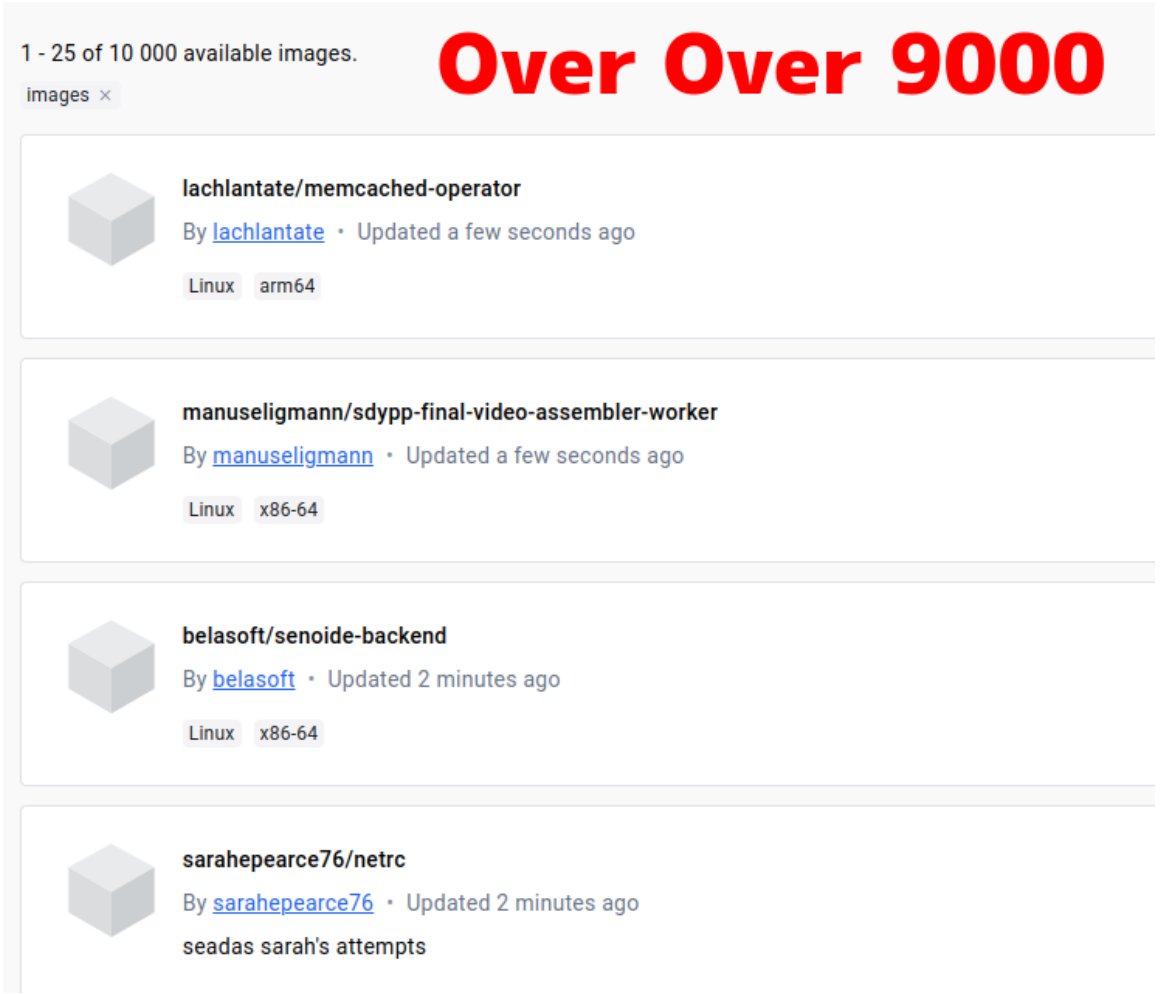
> Docker-Sponsored Open Source Software. These are images published and maintained by open-source projects that are sponsored by Docker through our open source program.



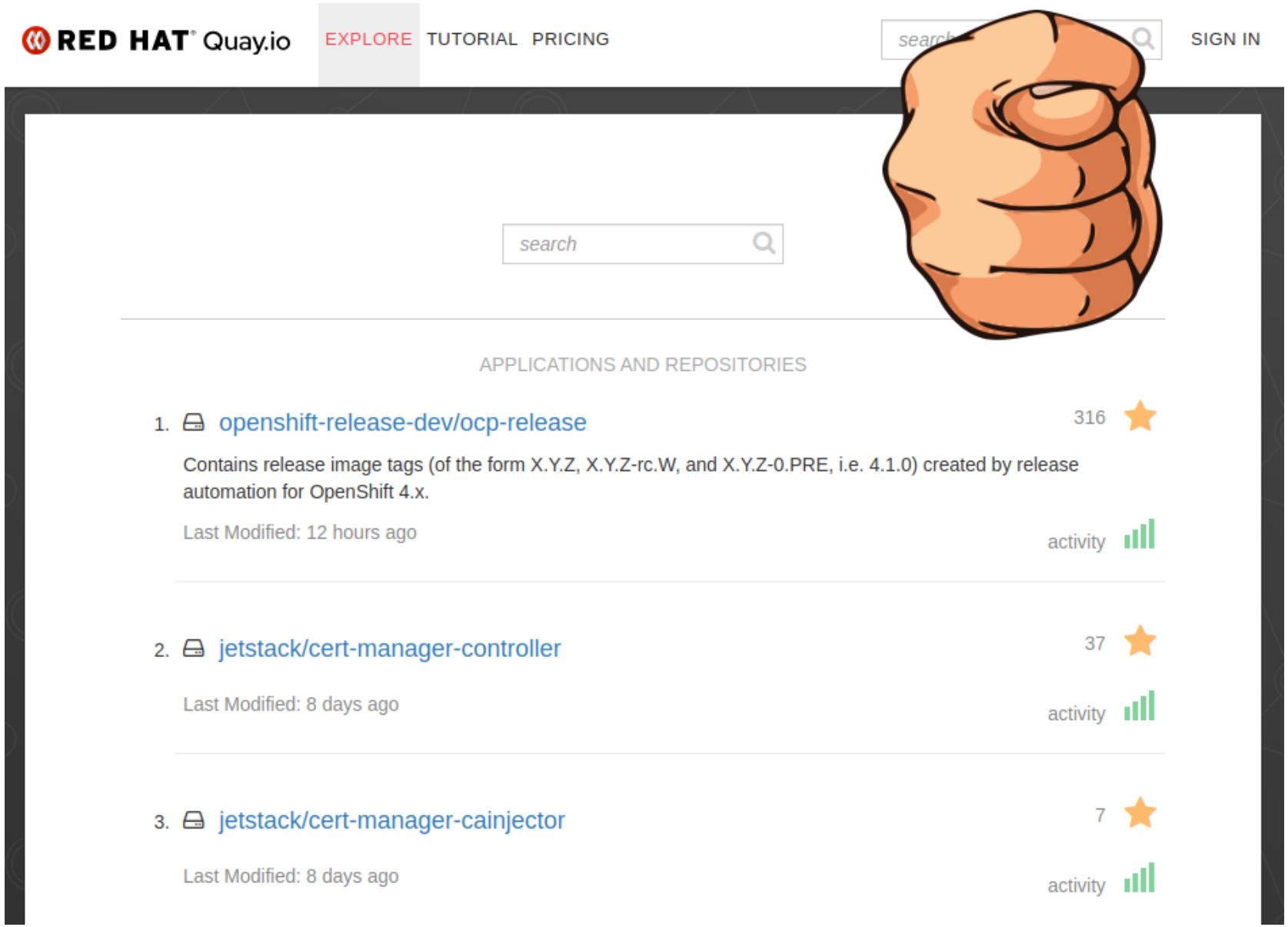
DockerHub

Other

> to the 150 million images created by the Docker community (2020)



Quay

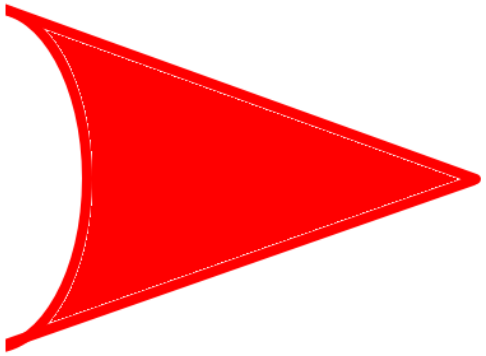

RED HAT Quay.io
EXPLORE
TUTORIAL
PRICING
SIGN IN
search
APPLICATIONS AND REPOSITORIES
1. openshift-release-dev/ocp-release
316
Contains release image tags (of the form X.Y.Z, X.Y.Z-rc.W, and X.Y.Z-0.PRE, i.e. 4.1.0) created by release automation for OpenShift 4.x.
Last Modified: 12 hours ago
activity
2. jetstack/cert-manager-controller
37
Last Modified: 8 days ago
activity
3. jetstack/cert-manager-cainjector
7
Last Modified: 8 days ago
activity

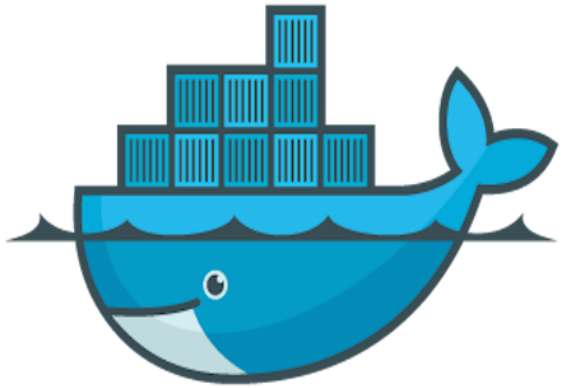
GitHub, gitlab.com

Base image

Кто знает что у него в базовом образе?

Base image (Кто собирал базовый образ сам?)





Base image

Кто собирал базовый образ сам?

Какие?

Base image

Кто собирал базовый образ сам?

Зачем?

```
$ docker run -it --rm redos-7.3:latest cat /etc/redos-release


RED OS release MUROM (7.3.2)
```

С чего начинается образ?

С мук выбора

SAST для Dockerfile

SAST для Dockerfile

Кто сканит Dockerfile?

SAST для Dockerfile

Что используется?

SAST для Dockerfile (Hadolint)

- Опирается на рекомендации Docker

SAST для Dockerfile (Hadolint)

- Опирается на рекомендации Docker
- Использует ShellCheck для скриптов в `RUN`

SAST для Dockerfile (Hadolint)

- Опирается на рекомендации Docker
- Использует ShellCheck для скриптов в `RUN`
- OpenSource

SAST для Dockerfile (Hadolint)

- Опирается на рекомендации Docker
- Использует ShellCheck для скриптов в `RUN`
- OpenSource
- Расширямость

SAST для Dockerfile (Hadolint)

- Haskell

```haskell
module Hadolint.Rule.DL3003 (rule) where

import Hadolint.Rule
import Hadolint.Shell (ParsedShell, usingProgram)
import Language.Docker.Syntax (Instruction (..), RunArgs (..))

rule :: Rule ParsedShell
rule = simpleRule code severity message check
  where
    code = "DL3003"
    severity = DLWarningC
    message = "Use WORKDIR to switch to a directory"
    check (Run (RunArgs args _)) = foldArguments (not . usingProgram "cd") args
    check _ = True
{-# INLINEABLE rule #-}
```

https://github.com/hadolint/hadolint/blob/master/src/Hadolint/Rule/DL3003.hs

```haskell
markGood :: Acc -> Acc
markGood Empty = Empty
markGood (Acc stageid good silent bad) =
  Acc stageid (good |> Set.insert stageid) (silent |> Set.delete stageid) (bad |> Set.delete stageid)

markSilentByAlias :: Text.Text -> Acc -> Acc
markSilentByAlias _ Empty = Empty
markSilentByAlias silentname (Acc stageid good silent bad) =
  Acc stageid good (silent |> Set.union stages) (bad |> remove stages)
  where
    stages = Set.filter byName bad
    byName (StageID BaseImage {alias = Nothing} _) = False
    byName (StageID BaseImage {alias = Just als} _) = unImageAlias als == silentname
    remove set fromThis = Set.difference fromThis set

getCurrentStageName :: State Acc -> Text.Text
getCurrentStageName (State _ (Acc (StageID BaseImage {image, alias = Nothing} _) _ _ _)) = imageName image
getCurrentStageName (State _ (Acc (StageID BaseImage {alias = Just als} _) _ _ _)) = unImageAlias als
getCurrentStageName _ = ""
```

https://github.com/hadolint/hadolint/blob/master/src/Hadolint/Rule/DL3049.hs

DL3013

Vlastimil Zeman edited this page on Nov 1, 2017 · 3 revisions

Pin versions in pip.

Problematic code:

```
FROM python:3.4
RUN pip install django
RUN pip install https://github.com/Banno/carbon/tarball/0.9.x-fix-events-callback
```

Correct code:

```
FROM python:3.4
RUN pip install django==1.9
RUN pip install git+https://github.com/Banno/carbon@0.9.15
```

https://habr.com/ru/companies/flant/articles/787494/

SAST для Dockerfile (Hadolint)

```
1 FROM scratch as base
  <string>:2:18:
  |
  2 | COPY --chmod=777 hadolint /bin/hadolint
  | ^
  invalid flag: --chmod
2 COPY --chmod=777 hadolint /bin/hadolint
3 LABEL org.opencontainers.image.source="https://github.com/hado
4 CMD ["/bin/hadolint", "-"]
5
6 FROM debian:bullseye-slim as debian
7 COPY --chmod=777 hadolint /bin/hadolint
8 LABEL org.opencontainers.image.source="https://github.com/hado
9
```

https://hadolint.github.io/hadolint/

SAST для Dockerfile (Hadolint)

COPY

COPY has two forms:

```
COPY [--chown=<user>:<group>] [--chmod=<perms>] <src>... <dest>
COPY [--chown=<user>:<group>] [--chmod=<perms>] ["<src>",... "<dest>"]
```

This latter form is required for paths containing whitespace

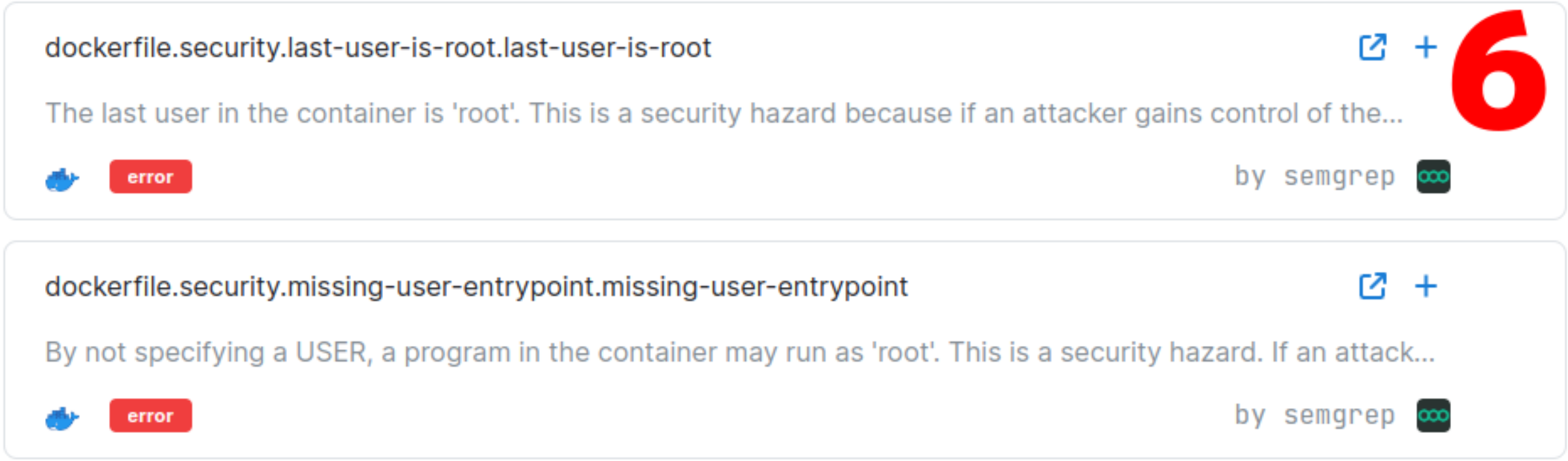
https://docs.docker.com/reference/dockerfile/#copy

SAST для Dockerfile (Semgrep)

SAST для Dockerfile (Semgrep)

Кто использует Semgrep?

SAST для Dockerfile (Semgrep)



https://semgrep.dev/p/dockerfile

SAST для Dockerfile (Semgrep)

missing-user-entrypoint

simple | advanced

```
rules:
  - id: missing-user-entrypoint
    patterns:
      - pattern: |
          ENTRYPOINT $...VARS
      - pattern-not-inside: |
          USER $USER
          ...
    fix: |
      USER non-root
      ENTRYPOINT $...VARS
    message: By not specifying a USER, a program in the
    container may run as 'root'.
      This is a security hazard. If an attacker can
      control a process running as
      root, they may have control over the container.
      Ensure that the last USER
      in a Dockerfile is a USER other than 'root'.
    severity: ERROR
```

test code | metadata | docs

```
FROM busybox

# uncomment for ok
#USER notroot

RUN git clone https://github.com/returntocorp/semgrep
RUN pip3 install semgrep

# ruleid: missing-user-entrypoint
ENTRYPOINT semgrep -f p/xss

# TODO: metavar bug
# ok: missing-user-entrypoint
ENTRYPOINT ["semgrep", "--config", "localfile", "targets"]

```

https://semgrep.dev/playground/r/dockerfile.security.missing-user-entrypoint.missing-user-entrypoint?editorMode=advanced

SAST для Dockerfile (Checkov)

```
FROM node:alpine
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000 22
HEALTHCHECK CMD curl --fail http://localhost:3000 || exit 1
CMD ["node","app.js"]
```

```
$ checkov -d . --framework dockerfile
```

https://www.checkov.io/7.Scan Examples/Dockerfile.html

SAST для Dockerfile (Checkov)

```
Check: CKV_DOCKER_1: "Ensure port 22 is not exposed"
        FAILED for resource: /Dockerfile.EXPOSE
        File: /Dockerfile:6-6
        Guide: https://docs.prismacloud.io/. . .
                6 | EXPOSE 3000 22

Check: CKV_DOCKER_3: "Ensure that a user for the container has been created"
        FAILED for resource: /Dockerfile.
        File: /Dockerfile:1-8
        Guide: https://docs.prismacloud.io/. . .
                1 | FROM node:alpine
                2 | WORKDIR /usr/src/app
                3 | COPY package*.json ./
                4 | RUN npm install
                5 | COPY . .
                6 | EXPOSE 3000 22
                7 | HEALTHCHECK CMD curl --fail http://localhost:3000 || exit 1
                8 | CMD ["node","app.js"]
```

https://www.checkov.io/5.Policy Index/dockerfile.html

SAST для Dockerfile (Checkov)

```python
class RootUser(BaseDockerfileCheck):
    def __init__(self) -> None:
        name = "Ensure the last USER is not root"
        id = "CKV_DOCKER_8"
        supported_instructions = ("USER",)
        categories = (CheckCategories.APPLICATION_SECURITY,)
        super().__init__(name=name, id=id, categories=categories, supported_instructions=supported_instructions)

    def scan_resource_conf(self, conf: list[_Instruction]) -> tuple[CheckResult, list[_Instruction] | None]:
        last_user = conf[-1]
        if last_user["value"] == "root":
            return CheckResult.FAILED, [last_user]

        return CheckResult.PASSED, [last_user]

```

https://github.com/bridgecrewio/checkov/blob/main/checkov/dockerfile/checks/RootUser.py

SAST для Dockerfile (Checkov)

```
metadata:
  id: "CKV2_DOCKER_2"
  name: "Ensure that certificate validation isn't disabled with curl"
  category: "APPLICATION_SECURITY"
definition:
    cond_type: attribute
    resource_types:
      - RUN
    attribute: value
    operator: not_regex_match
    value: ".*(curl[^\\|&;]*\\s+((--insecure)|(-[^-\\s]*k))).*"
```

https://github.com/bridgecrewio/checkov/blob/main/checkov/dockerfile/checks/graph_checks/RunUnsafeCurl.yaml

SAST для Dockerfile (Checkov)

```
metadata:
  id: "CKV2_DOCKER_12"
  name: "Ensure that certificate validation isn't disabled for npm via the 'NPM_CONFIG_STRICT_SSL' environmnet variable"
  category: "APPLICATION_SECURITY"
definition:
  or:
    - cond_type: attribute
      resource_types:
        - ARG
        - ENV
      attribute: value
      operator: not_regex_match
      value: "(.*\\s+)?(((NPM_CONFIG_STRICT_SSL)|(npm_config_strict_ssl))(=|\\s+)((false)|('false')|(\"false\"))).*"
    - cond_type: attribute
      resource_types:
        - RUN
      attribute: value
      operator: not_regex_match
      value: "(.*[\\s;&|]+)?(((NPM_CONFIG_STRICT_SSL)|(npm_config_strict_ssl))=((false)|('false')|(\"false\"))) .*"
```

https://github.com/bridgecrewio/checkov/blob/main/checkov/dockerfile/checks/graph_checks/EnvNpmConfigStrictSsl.yaml

SAST для Dockerfile

Что ещё может быть?

SAST для Dockerfile

Всё что угодно

SAST для Dockerfile (Leaky Vessels)

> CVE-2024-21626: уязвимость типа order-of-operations в команде WORKDIR в runc. Позволяет получить несанкционированный доступ к операционной системе хоста и потенциально скомпрометировать всю систему.

> CVE-2024-23651: состояние гонки в Buildkit, приводящее к непредсказуемому поведению

> CVE-2024-23652: проблема, позволяющая удалять произвольные файлы или каталоги на этапе удаления контейнера в Buildkit

> CVE-2024-23653: уязвимость возникает из-за недостаточной проверки привилегий в интерфейсе GRPC Buildkit

https://xakep.ru/2024/02/05/leaky-vessels/

SAST для Dockerfile (Leaky Vessels)

> $ runc --version
> runc version 1.1.7-0ubuntu1~22.04.2
> . . .

> cat /proc/11/cwd/../../../../etc/passwd
> root: x:0:0:root:/root:/bin/bash
> messagebus: x:498:498:User for D-Bus:/run/dbus:/usr/sbin/nologin
> lp: x:497:497:Printing daemon:/var/spool/lpd:/usr/sbin/nologin
> systemd-timesync: x:484:484:systemd Time
> . . .

https://nitroc.org/en/posts/cve-2024-21626-illustrated/

SAST для Dockerfile (Leaky Vessels)

```
common.Rule{
  ID: 1,
  CVE: "CVE-2024-21626",
  Name: "runc process.cwd & Leaked fds Container Breakout",
  Blogpost: "https://snyk.io/blog/cve-2024-21626-runc-process-cwd-container-breakout",
  Inst:        "WORKDIR",
  ArgRegex:    regexp2.MustCompile("/proc/self/fd/", 0),
},
common.Rule{
  ID: 2,
  CVE: "CVE-2024-23651",
  Name: "Buildkit Mount Cache Race: Build-time Race Condition Container Breakout",
  Blogpost: "https://snyk.io/blog/cve-2024-23651-docker-buildkit-mount-cache-race",
  Inst:        "RUN",
  ArgRegex:    regexp2.MustCompile("--mount=type=cache", 0),
},
common.Rule{
. . .
```

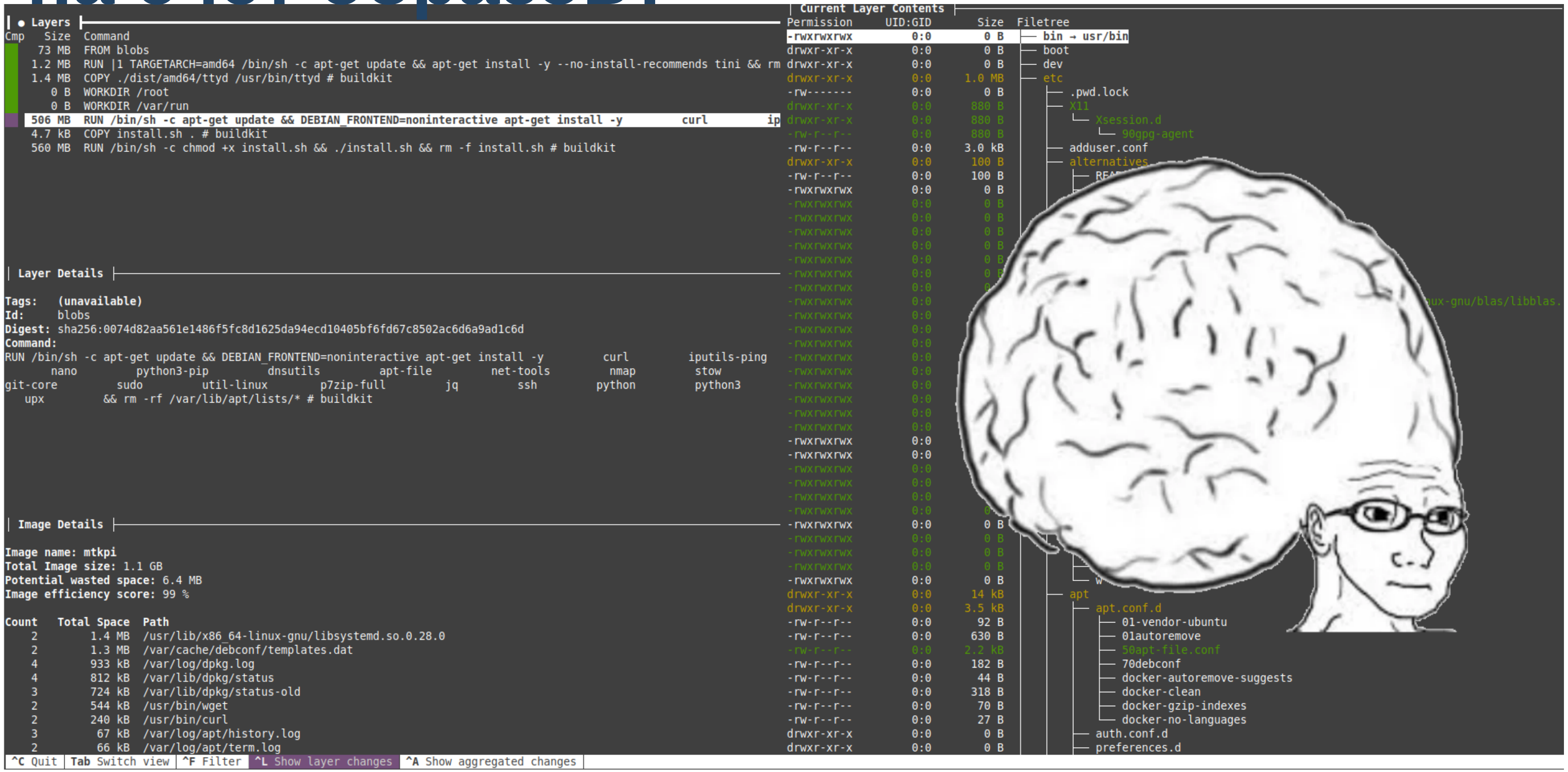
https://github.com/snyk/leaky-vessels-static-detector/

SAST для Dockerfile (Leaky Vessels)

Умеет не только Dockerfile, но и образы

Что на счёт образов?

Что на счёт образов?



Что на счёт образов?

Есть что!

Dockle

Проверяет образ на

- CIS Docker Benchmark <- Центр интернет-безопасности (CIS)
- best-practice for Dockerfile
- Example

```
docker run --rm goodwithtech/dockle:v0.4.14 goodwithtech/dockle-test:v2
```

https://github.com/goodwithtech/dockle

Dockle

```
FATAL   - CIS-DI-0009: Use COPY instead of ADD in Dockerfile
        * Use COPY : /bin/sh -c #(nop) ADD file:81c0a803075715d1a6b4f75a29f8a01b21cc170cfc1bff6702317d1be2fe71a3 in /app/credentials.json
FATAL   - CIS-DI-0010: Do not store credential in environment variables/files
        * Suspicious filename found : app/credentials.json (You can suppress it with "-af credentials.json")
        * Suspicious ENV key found : MYSQL_PASSWD on /bin/sh -c #(nop)  ENV MYSQL_PASSWD=password (You can suppress it with --accept-key)
FATAL   - DKL-DI-0005: Clear apt-get caches
        * Use 'rm -rf /var/lib/apt/lists' after 'apt-get install|update' : /bin/sh -c apt-get update && apt-get install -y git
FATAL   - DKL-LI-0001: Avoid empty password
        * No password user found! username : nopasswd
WARN    - CIS-DI-0001: Create a user for the container
        * Last user should not be root
INFO    - CIS-DI-0005: Enable Content trust for Docker
```

Dockle

```
FROM debian:latest

# DKL-LI-0005
RUN apt-get update && apt-get install -y git
# DKL-LI-0001
RUN useradd nopasswd -p ""
RUN chmod u+s /etc/shadow
RUN chmod g+s /etc/passwd
# CIS-DI-0009
# CIS-DI-0010
ADD credentials.json /app/credentials.json
COPY sample.txt /app/sample.txt
RUN chmod u+s /app/sample.txt
RUN chmod g+s /app/sample.txt
# CIS-DI-0010
ENV MYSQL_PASSWD password
RUN chmod g-s /app/sample.txt
VOLUME /usr
```

https://github.com/goodwithtech/dockle-ci-test/blob/master/Dockerfile

Dockle (CIS-DI-0009)

> Use COPY instead of ADD in Dockerfile

Что он хочет от нас и зачем?

Dockle (ADD vs COPY)

```
ADD credentials.json /app/credentials.json
```

> ADD instruction introduces risks such as adding malicious files from URLs without scanning and unpacking procedure vulnerabilities.

CIS-DI-0009

Bonus

```
ADD --checksum=sha256:24454f830cdb571e2c4ad15481119c43b3cafd48dd869a9b2945d1036d1dc68d \
     https://mirrors.edge.kernel.org/pub/linux/kernel/Historic/linux-0.01.tar.gz /
```

https://docs.docker.com/develop/develop-images/instructions/#add-or-copy

Dockle (CIS-DI-0010 - Files)

```
func (a CredentialAssessor) RequiredFiles() []string {
        return []string{
                "credentials.json",
                "credential.json",
                // TODO: Only check .docker/config.json
                // "config.json",
                "credentials",
                "credential",
                "password.txt",
                "id_rsa",
                "id_dsa",
                "id_ecdsa",
                "id_ed25519",
                "secret_token.rb",
                "carrierwave.rb",
                "omniauth.rb",
                "settings.py",
                "database.yml",
                "credentials.xml",
        }
}
```

Dockle (CIS-DI-0010 - Extensions)

```
func (a CredentialAssessor) RequiredExtensions() []string {
        return []string{
                // reference: https://github.com/eth0izzle/shhgit/blob/master/config.yaml
                // TODO: potential sensitive data but they have many false-positives.
                //       Dockle need to analyze each file.
                //".pem",
                //".key",
                //".p12",
                //".pkcs12",
                //".pfx",
                //".asc",

                ".secret",
                ".ovpn",
                ".private_key",
                ".cscfg",
                ".rdp",
                ".mdf",
                ".sdf",
                ".bek",
                ".tpm",
                ".fve",
                ".jks",
                ".psafe3",
                ".agilekeychain",
                ".keychain",
                ".pcap",
                ".gnucache",
        }
}
```

https://github.com/goodwithtech/dockle/blob/master/pkg/assessor/credential/credential.go

Dockle

MTKPI – Multi Tool Kubernetes Pentest Image

> $ dockle mtkpi

> FATAL - DKL-DI-0001: Avoid sudo command
> * Avoid sudo in container : RUN /bin/sh -c apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y curl iputils-ping nano python3-pip dnsutils apt-file net-tools nmap stow git-core sudo util-linux p7zip-full jq ssh python python3 upx && rm -rf /var/lib/apt/lists/* # buildkit
> ...

https://github.com/r0binak/MTKPI

Dockle

```
RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y \
        curl \
        iputils-ping \
        nano \
        python3-pip \
. . .
        nmap \
        stow \
        git-core \
        sudo \
        util-linux\
        p7zip-full \
. . .
        python3 \
        upx \
        && rm -rf /var/lib/apt/lists/*
```

https://github.com/r0binak/MTKPI/blob/main/Dockerfile

Clair

> Vulnerability Static Analysis for Containers

- RedHat
- Интеграция с Quay
- Использует отдельный пакет - Claircore

- https://secdb.alpinelinux.org/
- http://repo.us-west-2.amazonaws.com/2018.03/updates/x86_64/mirror.list
- https://cdn.amazonlinux.com/2/core/latest/x86_64/mirror.list
- https://cdn.amazonlinux.com/al2023/core/mirrors/latest/x86_64/mirror.list
- https://deb.debian.org/
- https://security-tracker.debian.org/tracker/data/json
- https://nvd.nist.gov/feeds/json/cve/1.1/
- https://linux.oracle.com/security/oval/com.oracle.elsa-*.xml.bz2
- https://packages.vmware.com/photon/photon_oval_definitions/
- https://access.redhat.com/security/data/metrics/cvemap.xml
- https://access.redhat.com/security/cve/
- https://access.redhat.com/security/data/oval/v2/PULP_MANIFEST
- https://support.novell.com/security/oval/
- https://api.launchpad.net/1.0/
- https://security-metadata.canonical.com/oval/com.ubuntu.*.cve.oval.xml
- https://osv-vulnerabilities.storage.googleapis.com/

https://github.com/quay/clair

Snyk (Demo)

Test › Docker nginx:1.25.4-bookworm-perl

Docker nginx:1.25.4-bookworm-perl

Vulnerabilities	82 via 134 paths	**Dependencies**	155	**Source**	Docker
Target OS	debian:12				

Issues | Dependencies

Severity

- Critical 1
- High 1
- Medium 1
- Low 79

CRITICAL SEVERITY

Integer Overflow or Wraparound

Vulnerable module: zlib/zlib1g

Introduced through: zlib/zlib1g@1:1.2.13.dfsg-1

Detailed paths

- **Introduced through**: nginx@1.25.4-bookworm-perl › zlib/zlib1g@1:1.2.13.dfsg-1

https://snyk.io/docker-images/

Trivy

Targets

- Container Image
- Filesystem
- Git Repository (remote)
- Virtual Machine Image
- Kubernetes
- AWS

Scanners

- OS packages and software dependencies in use (SBOM)
- Known vulnerabilities (CVEs)
- IaC issues and misconfigurations
- Sensitive information and secrets
- Software licenses

https://aquasecurity.github.io/trivy/

Trivy (Report)

usr/local/bin/kdigger (gobinary)

Total: 7 (UNKNOWN: 0, LOW: 0, MEDIUM: 3, HIGH: 4, CRITICAL: 0)

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
golang.org/x/crypto	CVE-2023-48795	MEDIUM	fixed	v0.0.0-20220926161630-eccd6366d1be	0.17.0	ssh: Prefix truncation attack on Binary Packet Protocol (BPP) https://avd.aquasec.com/nvd/cve-2023-48795
golang.org/x/net	CVE-2022-41721	HIGH		v0.0.0-20220927171203-f486391704dc	0.1.1-0.20221104162952-702349b0e862	x/net/http2/h2c: request smuggling https://avd.aquasec.com/nvd/cve-2022-41721
	CVE-2022-41723				0.7.0	net/http, golang.org/x/net/http2: avoid quadratic complexity in HPACK decoding https://avd.aquasec.com/nvd/cve-2022-41723
	CVE-2023-39325				0.17.0	golang: net/http, x/net/http2: rapid stream resets can cause excessive work (CVE-2023-44487) https://avd.aquasec.com/nvd/cve-2023-39325
	CVE-2023-3978	MEDIUM			0.13.0	golang.org/x/net/html: Cross site scripting https://avd.aquasec.com/nvd/cve-2023-3978
	CVE-2023-44487				0.17.0	HTTP/2: Multiple HTTP/2 enabled web servers are vulnerable to a DDoS attack... https://avd.aquasec.com/nvd/cve-2023-44487
golang.org/x/text	CVE-2022-32149	HIGH		v0.3.7	0.3.8	golang: golang.org/x/text/language: ParseAcceptLanguage takes a long time to parse complex tags https://avd.aquasec.com/nvd/cve-2022-32149

https://github.com/quarkslab/kdigger

Trivy (Report)

> $ trivy image --severity CRITICAL mtkpi

usr/local/bin/kubescape (gobinary)

Total: 2 (CRITICAL: 2)

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
github.com/moby/buildkit	CVE-2024-23652	CRITICAL	fixed	v0.12.1	0.12.5	moby/buildkit: possible host system access from mount stub cleaner https://avd.aquasec.com/nvd/cve-2024-23652
	CVE-2024-23653					moby/buildkit: Buildkit's interactive containers API does not validate entitlements check https://avd.aquasec.com/nvd/cve-2024-23653

Trivy (Misconfigurations)

```
$ ls iac/
Dockerfile  deployment.yaml  main.tf mysql-8.8.26.tar
$ trivy conf --severity HIGH,CRITICAL ./iac
```

```
Dockerfile (dockerfile)
=======================
Tests: 23 (SUCCESSES: 22, FAILURES: 1, EXCEPTIONS: 0)
Failures: 1 (HIGH: 1, CRITICAL: 0)

...

deployment.yaml (kubernetes)
============================
Tests: 28 (SUCCESSES: 15, FAILURES: 13, EXCEPTIONS: 0)
Failures: 13 (MEDIUM: 4, HIGH: 1, CRITICAL: 0)
```

https://aquasecurity.github.io/trivy/v0.49/docs/scanner/misconfiguration/

Trivy (SBOM)

- Может делать анализ на основе уже сгенерированного

```
#Scan CycloneDX and show the result in tables
$ trivy sbom /path/to/report.cdx
```

- Может генерировать сам

```
$ trivy image --format cyclonedx --output result.cdx mtkpi
--format string
format (table,json,template,sarif,cyclonedx,spdx,spdx-json,github,cosign-vuln)
(default "table")
```

https://www.techtarget.com/searchsecurity/tip/SBOM-formats-compared-CycloneDX-vs-SPDX-vs-SWID-Tags

⚠ SBOM ⚠

NodeJS Npm

	Private	Public	Comment
Trivy	1219	150	Missed packages
cdxgen	1483	766	
CycloneDX Tool	-	594	Not worked with some packages without version
gemnasium	1329	584	No Dependency Graph, Missed packages, False Positive Dependencies
Custom solution	-	-	

https://www.youtube.com/watch?v=j81idELbnDQ

Trivy (Compliance)

> ⚠️ EXPERIMENTAL
>
> This feature might change without preserving backwards compatibilit

```
$ trivy image --compliance docker-cis mtkpi
Summary Report for compliance: CIS Docker Community Edition Benchmark v1.1.0
```

ID	Severity	Control Name	Status	Issues
4.1	HIGH	Ensure a user for the container has been created	FAIL	1
4.2	HIGH	Ensure that containers use trusted base images (Manual)	-	-
4.3	HIGH	Ensure unnecessary packages are not installed in the container (Manual)	-	-
4.4	CRITICAL	Ensure images are scanned and rebuilt to include security patches	FAIL	1
4.5	LOW	Ensure Content trust for Docker is Enabled (Manual)	-	-
4.6	LOW	Ensure HEALTHCHECK instructions have been added to the container image	FAIL	1
4.7	HIGH	Ensure update instructions are not use alone in the Dockerfile	PASS	0
4.8	HIGH	Ensure setuid and setgid permissions are removed in the images (Manual)	-	-
4.9	LOW	Ensure COPY is used instead of ADD in Dockerfile	PASS	0
4.10	CRITICAL	Ensure secrets are not stored in Dockerfiles	PASS	0
4.11	MEDIUM	Ensure verified packages are only Installed (Manual)	-	-

https://aquasecurity.github.io/trivy/v0.49/docs/compliance/compliance/

Grype

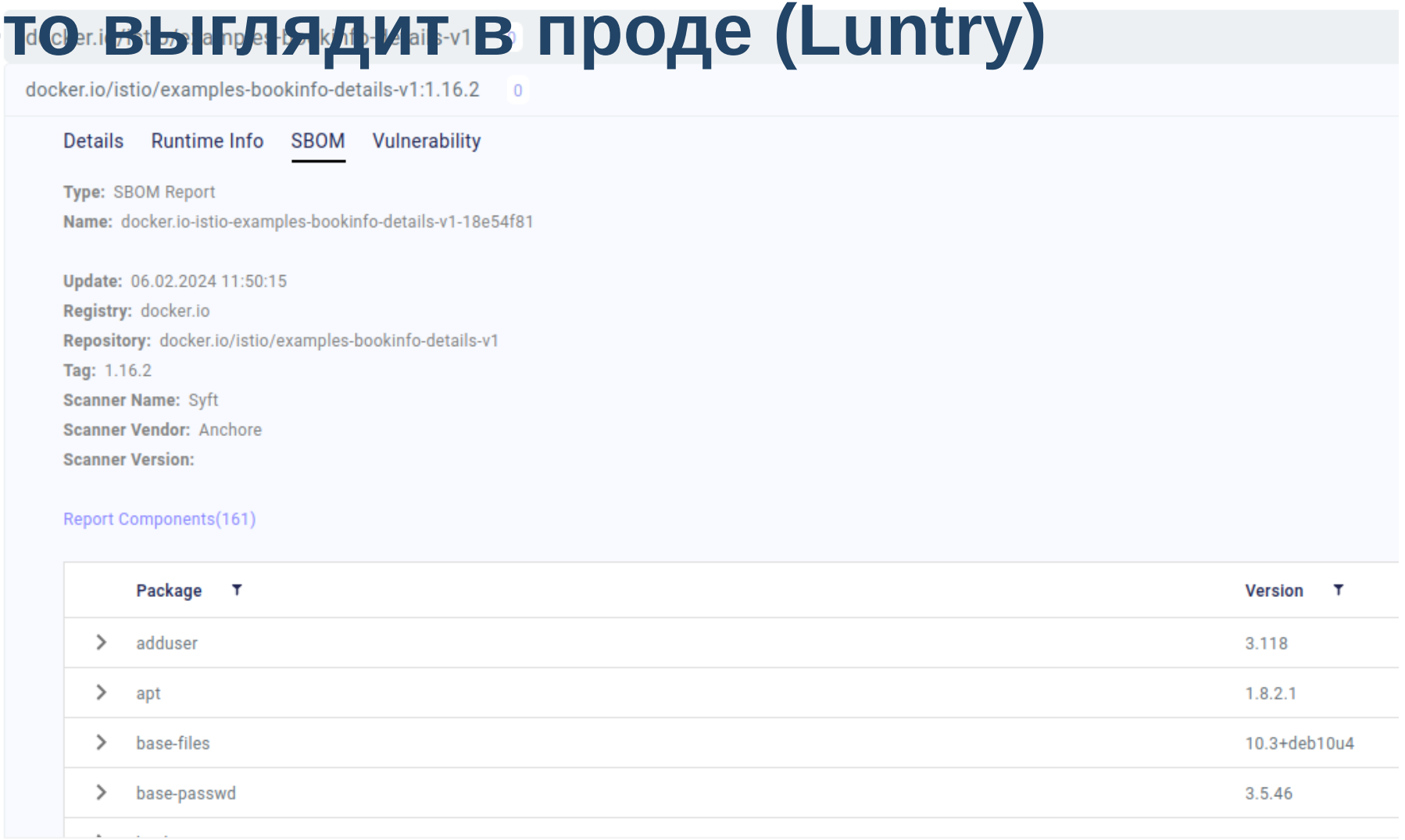
Grype + Syft ≈ Trivy

> Syft - CLI tool and library for generating a Software Bill of Materials from container images and filesystems

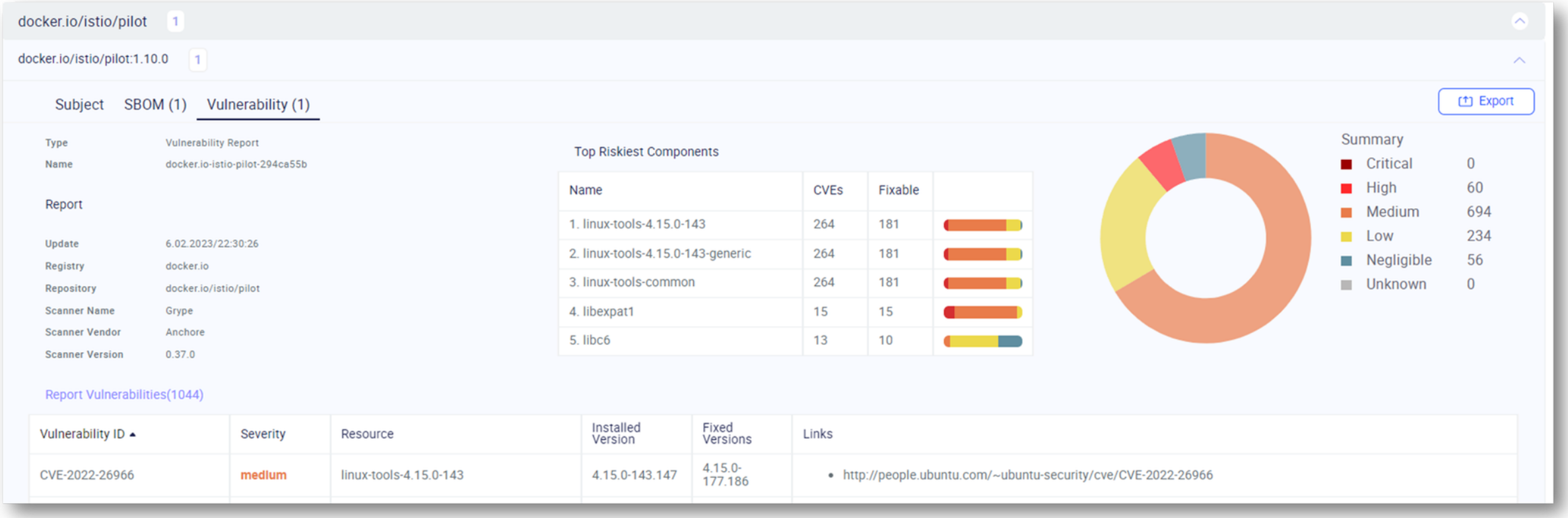
- Alpine Linux SecDB: https://secdb.alpinelinux.org/
- Amazon Linux ALAS: https://alas.aws.amazon.com/AL2/alas.rss
- RedHat RHSAs: https://www.redhat.com/security/data/oval/
- Debian Linux CVE Tracker: https://security-tracker.debian.org/tracker/data/json
- Github GHSAs: https://github.com/advisories
- National Vulnerability Database (NVD): https://nvd.nist.gov/vuln/data-feeds
- Oracle Linux OVAL: https://linux.oracle.com/security/oval/
- RedHat Linux Security Data: https://access.redhat.com/hydra/rest/securitydata/
- Suse Linux OVAL: https://ftp.suse.com/pub/projects/security/oval/
- Ubuntu Linux Security: https://people.canonical.com/~ubuntu-security/

https://github.com/anchore/grype

Как это выглядит в проде (Luntry)



Как это выглядит в проде (Luntry)



Что дальше?

НАСКАНИРОВАЛСЯ И СПИТ

Status	Pipeline
failed	#319049365 error
failed	#317510274 error
failed	#317485818 error

Registry

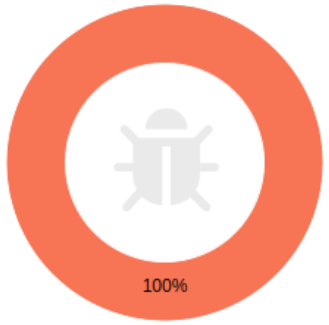
"Чужие"

- Docker Hub
- GitHub
- GitLab
- Quay
- Chainguard
- . . .

"Свои"

- registry
- GitLab
- Quay
- Harbor
- Artifactory / JFrog Container Registry
- Nexus
- . . .

Quay



Quay Security Scanner has detected **1** vulnerabilities.

1 High-level vulnerabilities.

Vulnerabilities

CVE	SEVERITY	PACKAGE	CURRENT VERSION	FIXED IN VERSION
GO-2024-2454	High	github.com/lestrrat-go/jwx	v1.2.28	(None)

SEVERITY NOTE

Note that this vulnerability was originally given a CVSSv3 score of **7.5** by NVD but was subsequently reclassified as High by osv/go

VECTORS

Attack Vector	Attack Complexity	Privileges Required	User Interaction	Scope	Confidentiality Impact	Integrity Impact	Availability Impact
Network	Low	None	None	Unchanged	High	High	High
Adjacent Network	High	Low	Required	Changed	Low	Low	Low
Local		High			None	None	None
Physical							

https://quay.io/repository/minio/minio/manifest/sha256:6d289c0a13e261dc63ddd642ce4151290f07352e47f3b95695c161d97689c0c7

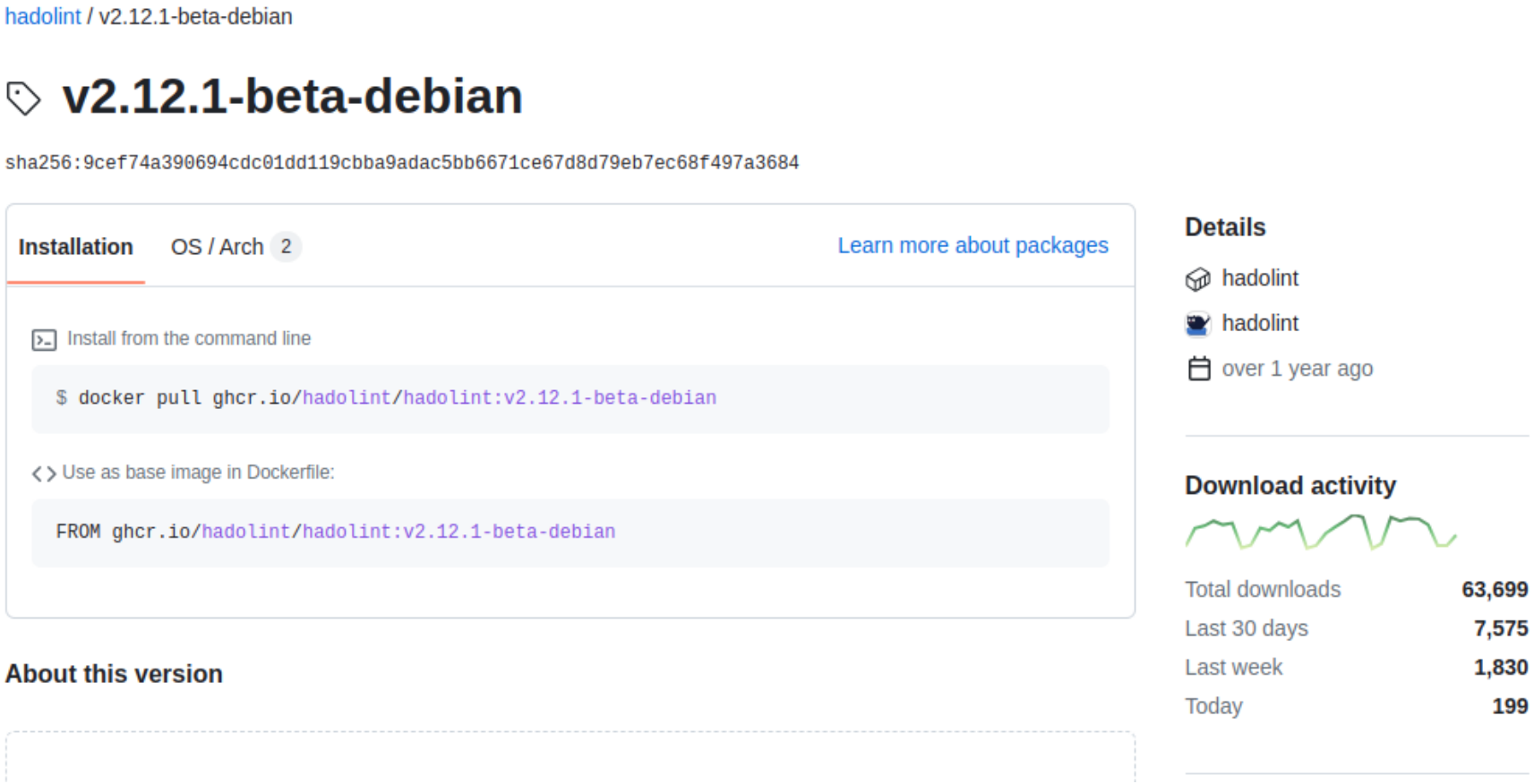
Quay

Manifest Layers

```
>_ CMD ["minio"]
>_ ENTRYPOINT ["/usr/bin/docker-entrypoint.sh"]
>_ VOLUME [/data]
>_ EXPOSE map[9000/tcp:{}]
>_ COPY dockerscripts/docker-entrypoint.sh /usr/bin/docker-entrypoint.sh # buildkit
>_ COPY LICENSE /licenses/LICENSE # buildkit
>_ COPY CREDITS /licenses/CREDITS # buildkit
>_ COPY /go/bin/mc /usr/bin/mc # buildkit
>_ COPY /go/bin/minio /usr/bin/minio # buildkit
>_ COPY /etc/ssl/certs/ca-certificates.crt /etc/ssl/certs/ # buildkit
>_ ENV MINIO_ACCESS_KEY_FILE=access_key MINIO_SECRET_KEY_FILE=secret_key MINIO_ROOT_USER_FILE=access_key MINIO_
MINIO_UPDATE_MINISIGN_PUBKEY=RWTx5Zr1tiHQLwG9keckT0c45M3AGeHD6IvimQHpyRywVWGbP1aVSGav MINIO_CONFIG_ENV_FILE=cor
>_ LABEL name=MinIO vendor=MinIO Inc <dev@min.io> maintainer=MinIO Inc <dev@min.io> version=RELEASE.2024-02-17T
Storage, API compatible with Amazon S3 cloud storage service. description=MinIO object storage is fundamentally
ideal for large, private cloud environments with stringent security requirements and delivers mission-critical
```

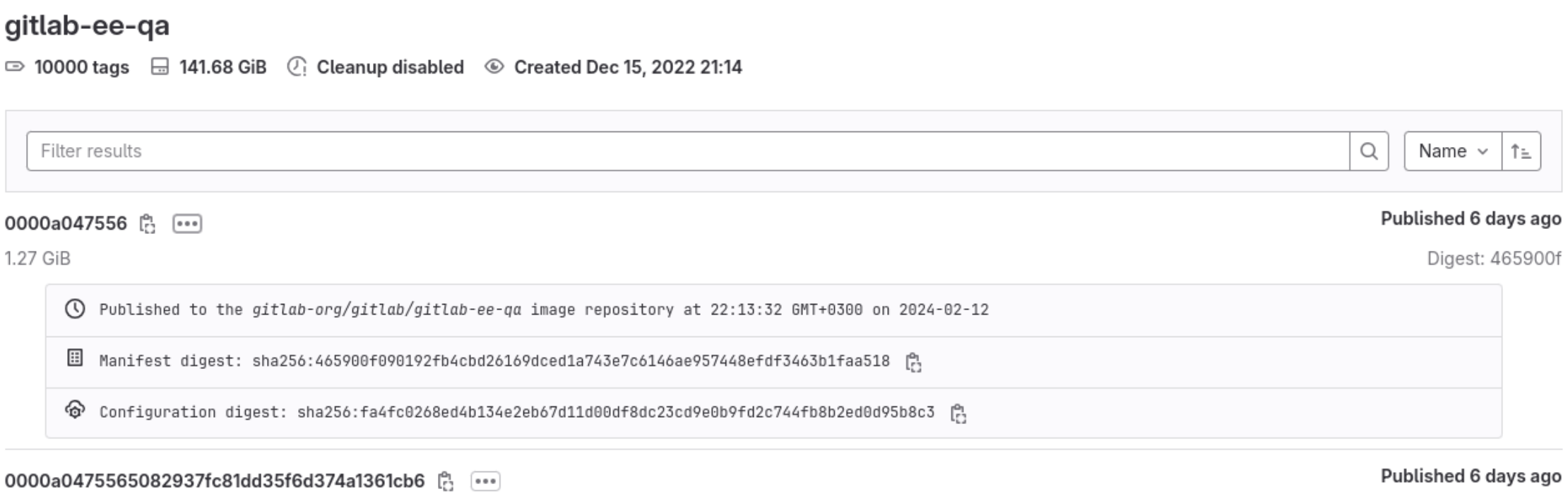
https://quay.io/repository/minio/minio/manifest/sha256:6d289c0a13e261dc63ddd642ce4151290f07352e47f3b95695c161d97689c0c7

GitHub



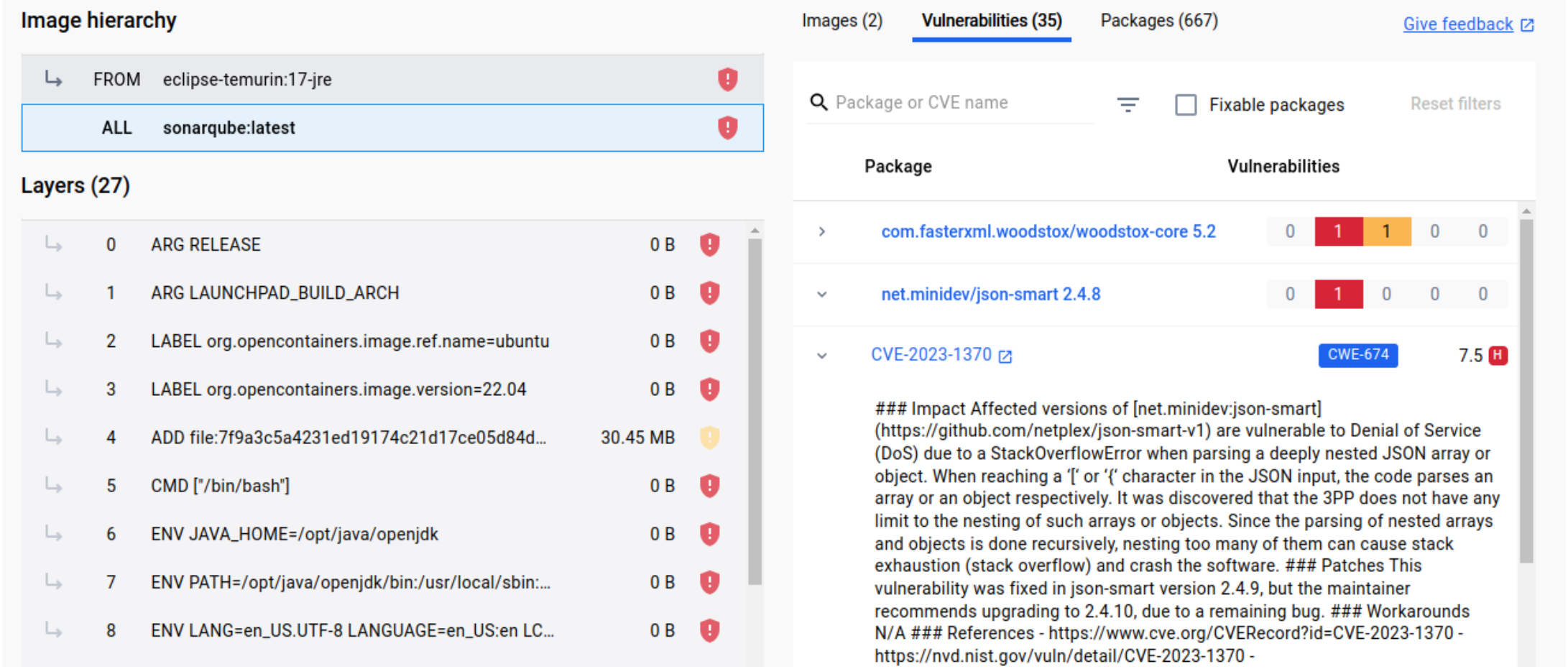
https://github.com/orgs/hadolint/packages/container/hadolint/50290021?tag=v2.12.1-beta-debian

GitLab



https://gitlab.com/gitlab-org/gitlab/container_registry/3728789

Docker Hub



https://hub.docker.com/layers/library/sonarqube/latest/images/sha256-03bed176ce9691799b787ef920b5cabe2daefef7ecf64a77dbf7384f43a3ae42?context=explore&tab=vulnerabilities

Docker Hub (Log4Shell)

Docker Hub (Docker Scout)

- Умеет в интеграцию с
  - Artifactory
  - Amazon Elastic Container Registry
  - Azure Container Registry
- Требуется платный аккаунт
- Утверждается, что доступен для sponsored и verified аккаунтов, но я не нашёл ни одного живого примера

https://docs.docker.com/scout/integrations/

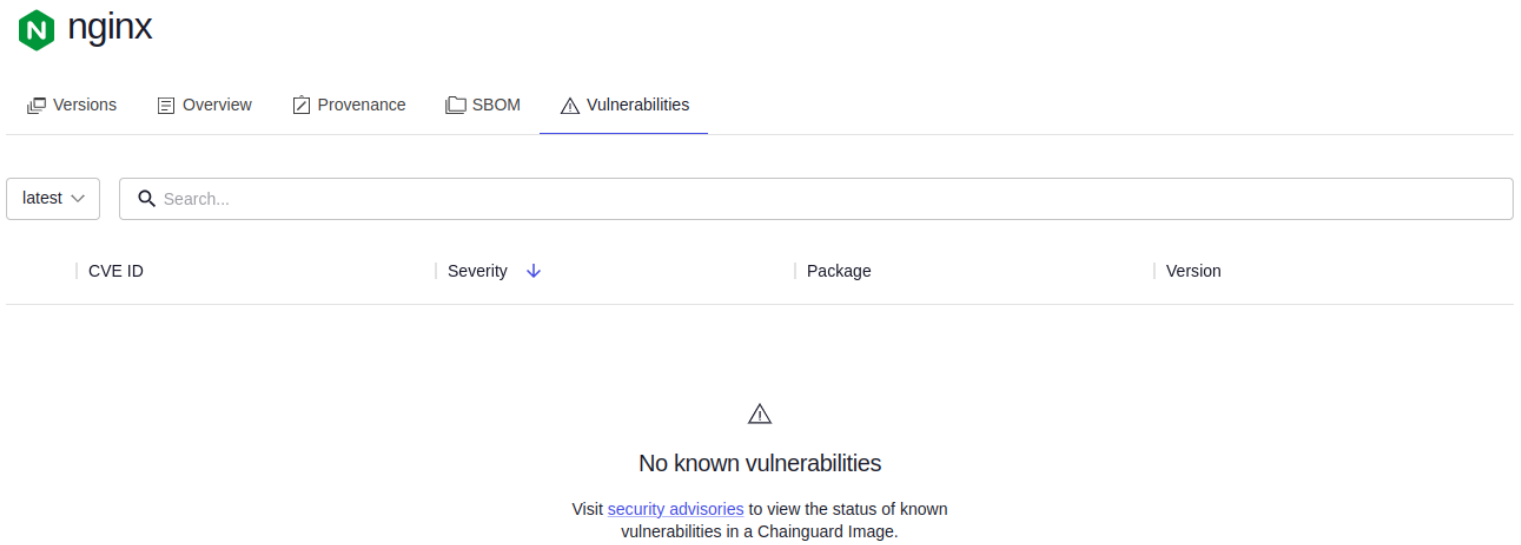
Harbor (Trivy, ~~Clair~~)

https://goharbor.io/docs/2.10.0/administration/vulnerability-scanning/

Chainguard

Цель проекта минимзация поверхности атаки.

> Minimal, hardened images with SBOMs and signatures



https://www.chainguard.dev/chainguard-images

Chainguard (Wolfi)

Собственный дистрибутив с собственной экосистемой сборки

> The key features of Wolfi are:
>
> - Provides a high-quality, build-time SBOM as standard for all packages
> - Packages are designed to be granular and independent, to support minimal images
> - Uses the proven and reliable APK package format
> - Fully declarative and reproducible build system
> - Designed to support glibc and musl

https://www.chainguard.dev/unchained/introducing-wolfi-the-first-linux-un-distro

Chainguard (Grype)

name	date	low	med	high	crit	unk	tot
cgr.dev/chainguard/nginx:latest	2024-02-17 14:34:01	0	0	0	0	0	0
nginx:latest	2024-02-17 14:34:22	4	30	12	2	2	121
cgr.dev/chainguard/nginx:latest	2024-01-21 14:33:57	0	0	0	0	0	0
nginx:latest	2024-01-21 14:34:18	4	25	11	2	9	123

https://github.com/chainguard-dev/image-comparison/blob/main/data.csv

Chainguard

Минусы

- latest
- Своя экосистема

Плюсы

- Открытая экосистема
  - Build image - apko
  - Build APK - melange
  - Package repo
- Есть описание сборки через Dockerfile
- SLSA - Supply chain Levels for Software Artifacts
- cosign

Chainguard (cosign)

```
$ COSIGN_EXPERIMENTAL=1 cosign verify cgr.dev/chainguard/nginx:latest | jq
Verification for cgr.dev/chainguard/nginx:latest --
The following checks were performed on each of these signatures:
- The cosign claims were validated
- Existence of the claims in the transparency log was verified offline
- Any certificates were verified against the Fulcio roots.
```


```
[
{
"critical": {
"identity": {
"docker-reference": "ghcr.io/distroless/nginx"
},
"image": {
"docker-manifest-digest":
"sha256:634ee2ce22a62ed1a22e11d11a09b6aa9134322d85f0467878fbaae0a28eba1e"
},
"type": "cosign container image signature"
},
```

Cosign

```
- "Keyless signing" with the Sigstore public good
    Fulcio certificate authority and Rekor transparency log (default)

- Hardware and KMS signing

- Signing with a cosign generated encrypted private/public keypair

- Container Signing, Verification and Storage in an OCI registry.

- Bring-your-own PKI
```

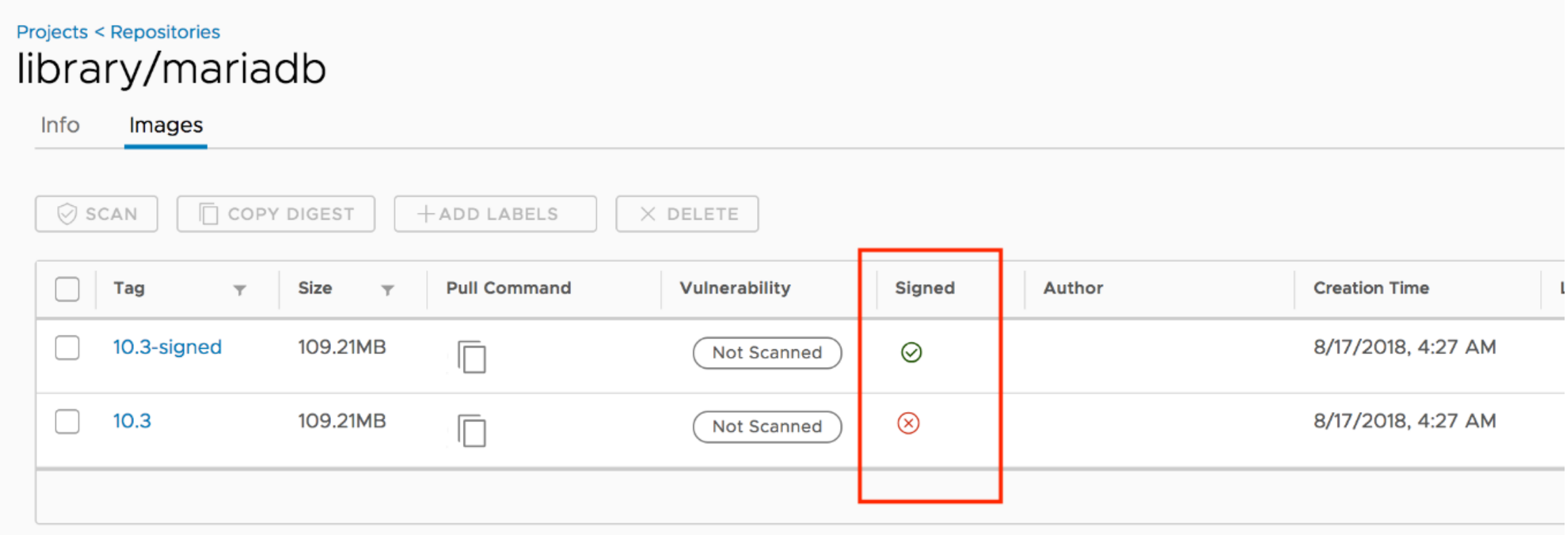
https://github.com/sigstore/cosign

Cosign

```
$ cosign sign @sha256:...

Generating ephemeral keys...
Retrieving signed certificate...
. . .
```

```
$ cosign verify --key cosign.pub @sha256:...
```

Cosign



https://goharbor.io/docs/2.10.0/working-with-projects/working-with-images/sign-images/

Cosign + Trivy = attestation

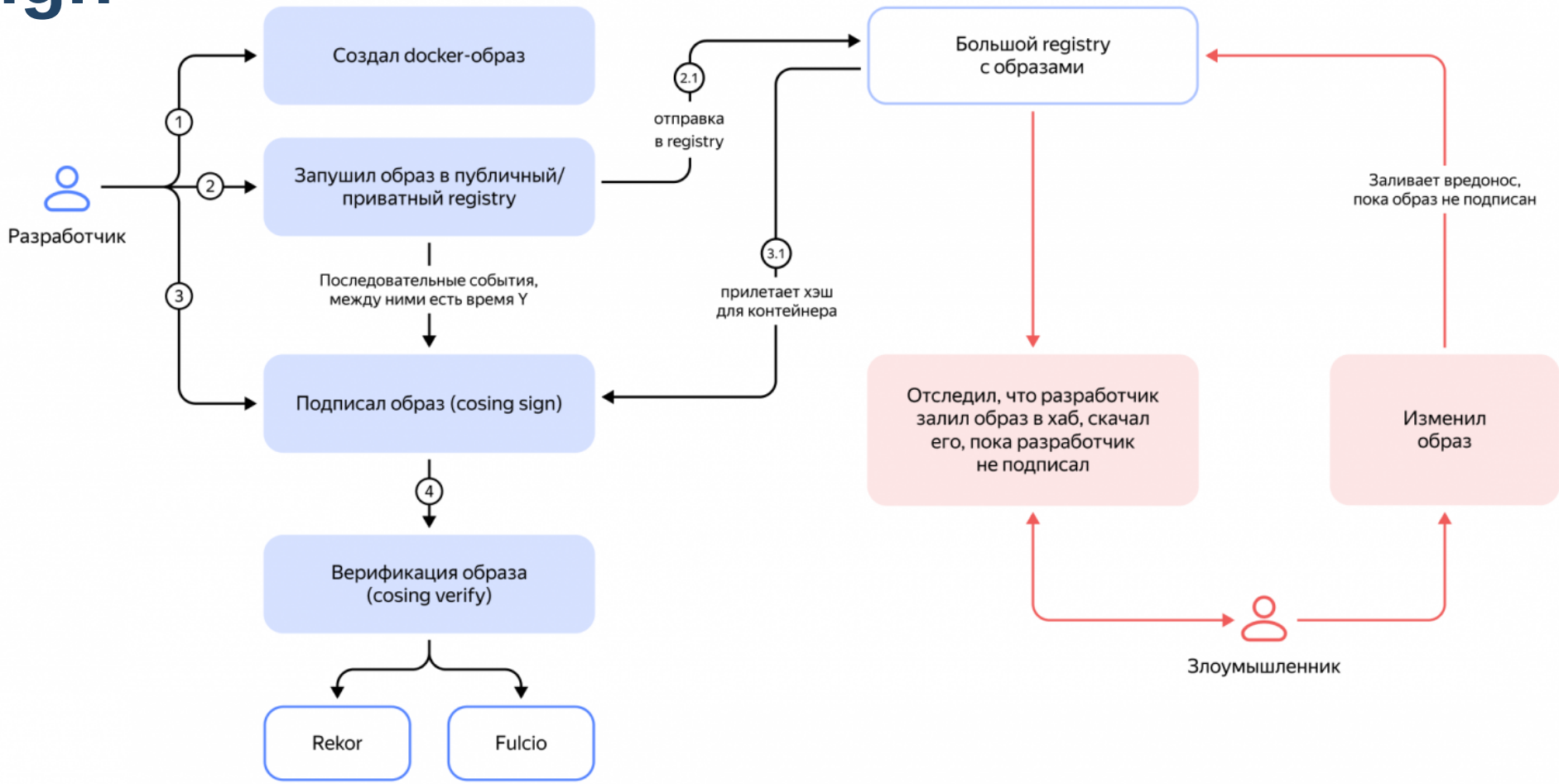
```
$ trivy image --format cosign-vuln --output vuln.json $IMG
```

```
$ cosign attest --key /path/to/cosign.key --type vuln --predicate vuln.json $IMG
```

```
$ cosign verify-attestation --key /path/to/cosign.pub --type vuln $IMG
```

https://aquasecurity.github.io/trivy/v0.49/docs/supply-chain/attestation/vuln/

Cosign



https://habr.com/ru/companies/yandex_cloud_and_infra/articles/752924/

Practice (Kyverno)

```
spec:
  validationFailureAction: Enforce
. . .
  rules:
    - name: checking-vulnerability-scan-not-older-than-one-hour
      match:
        any:
        - resources:
            kinds:
              - Pod
      verifyImages:
      - imageReferences:
        - "*"
        attestations:
        - type: https://cosign.sigstore.dev/attestation/vuln/v1
          conditions:
          - all:
            - key: "{{ time_since('','{{ metadata.scanFinishedOn }}', '') }}"
              operator: LessThanOrEquals
              value: "123h"
```

Practice (Kyverno)

```
error: statefulsets.apps "meilisearch" could not be patched: admission
        webhook "mutate.kyverno.svc-fail" denied the request:

resource StatefulSet/meilisearch/meilisearch was blocked
                                    due to the following policies

verify-image:
  autogen-verify-image: 'image signature verification failed for ghcr.io
  patrickdung/meilisearch-crossbuild:v0.24.0-unsigned:
  failed to verify image: fetching signatures: remote image:
  GET https://ghcr.io/v2/pat<...>manifests/sha256-fe<...>cbd.sig:
    MANIFEST_UNKNOWN: manifest unknown'
```

https://www.cloudnative.quest/posts/security/2022-01-01/improve-supply-chain-security-with-github-actions-and-open-source-tools/

Practice (Policy Controller)

> The policy-controller admission controller can be used to enforce policy on a Kubernetes cluster based on verifiable supply-chain metadata from cosign.

```
apiVersion: policy.sigstore.dev/v1alpha1
kind: ClusterImagePolicy
metadata:
  name: keyless-attestation-sbom-spdxjson
spec:
  images:
  - glob: "**"
  authorities:
  - name: keyless
    keyless:
      url: "https://fulcio.sigstore.dev"
      identities:
        - issuer: https://token.actions.githubusercontent.com
          subject: "https://github.com/sigstore/policy-controller/.github/workflows/policy-tester-examples.yml@refs/heads/main"
    ctlog:
      url: https://rekor.sigstore.dev
    attestations:
    - name: must-have-spdxjson
      predicateType: spdxjson
      policy:
        type: cue
        data: |
          predicateType: "https://spdx.dev/Document"
```

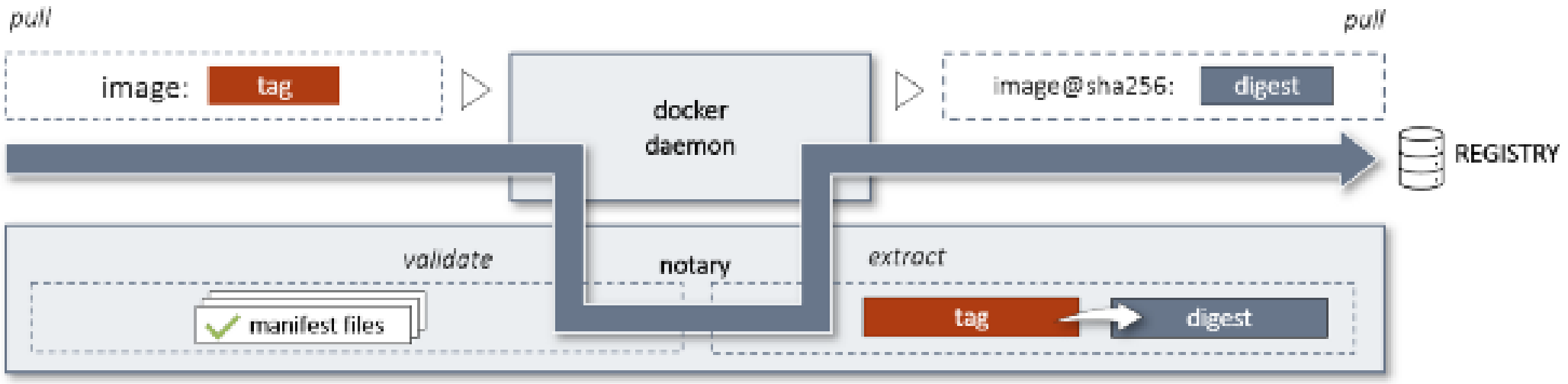
https://github.com/sigstore/policy-controller

Practice (DCT + Notary)

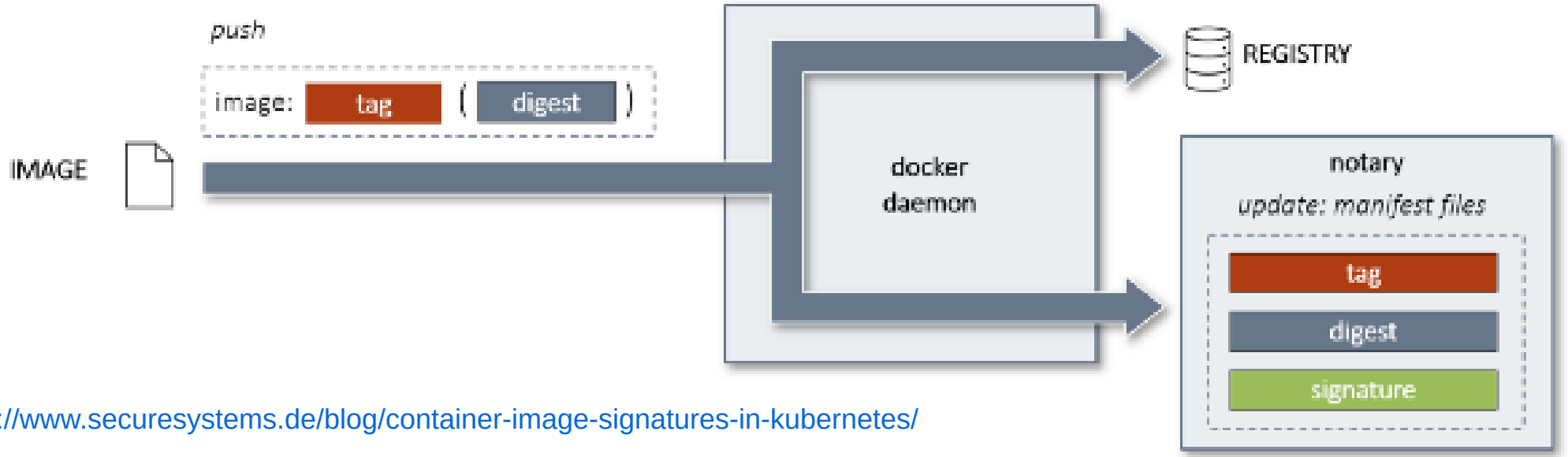
Docker Content Trust (DCT) - инструментарий, позволяющий подписывать части собираемых образов и публиковать в Notary

Notary - сервис, который хранит подписи для файлов

https://docs.docker.com/engine/security/trust/



Practice (DCT + Notary)



https://www.securesystems.de/blog/container-image-signatures-in-kubernetes/

Practice (DCT + Notary)

```
$ export DOCKER_CONTENT_TRUST=1
```

```
$ docker image pull nigelpoulton/tu-demo
Using default tag: latest

Error: remote trust data does not exist for docker.io/nigelpoulton/tu-demo:
- notary.docker.io does not have trust data for docker.io/nigelpoulton/tu-demo
```

```
$ docker image pull alpine:latest
Pull (1 of 1): alpine:latest@sha256:c5b. . .d4f27761f8e1ad6b
docker.io/library/alpine@sha256:c5b. . .d6b: Pulling from library/alpine
4abcf2066143: Pull complete
Digest: sha256:c5b. . .d6b
Status: Downloaded newer image for alpine@sha256:c5b. . .7d6b
Tagging alpine@sha256:c5b. . .d6b as alpine:latest
docker.io/library/alpine:latest
```

Practice (DCT + Notary)

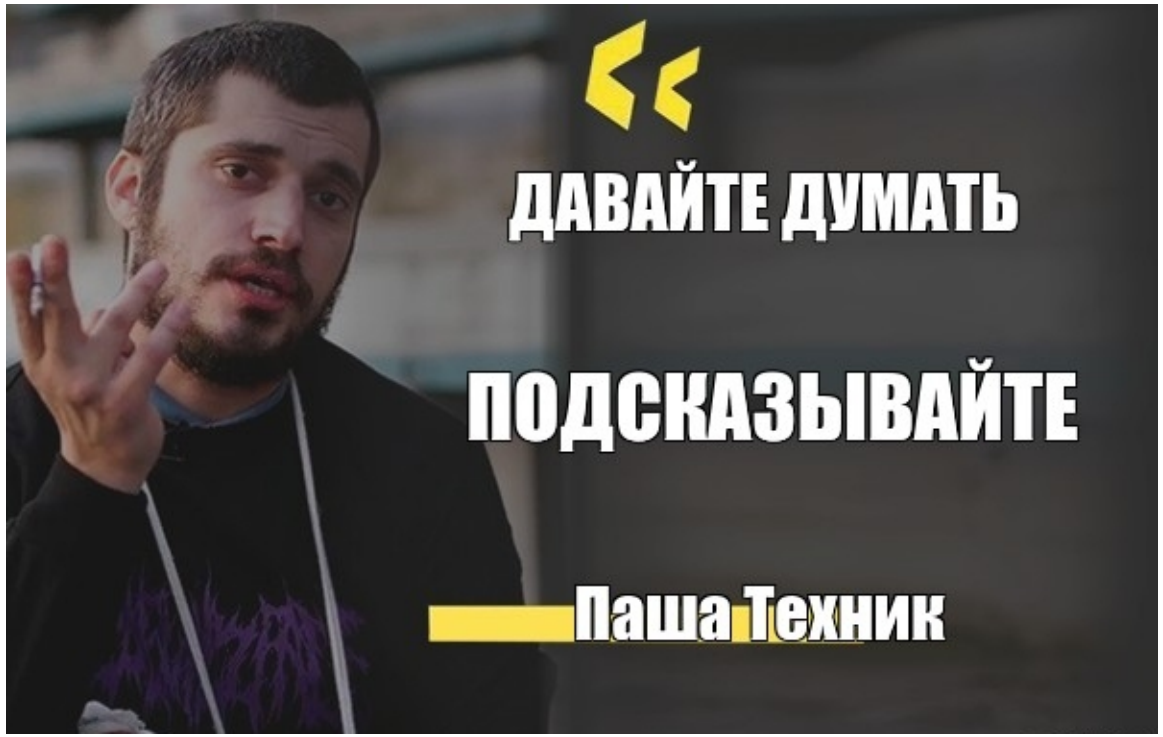
```
$ docker trust key generate circlecikey

$ docker trust signer add --key circlecikey.pub circlecikey \
                                  examplenamespace/examplenamespace

$ docker build . -t examplenamespace/exampleimage:0.1.0

$ docker trust sign examplenamespace/exampleimage:0.1.0
```

https://discuss.circleci.com/t/sign-docker-images-with-docker-content-trust/40319

Practice (Public -> Private)

Какие у кого правила для попадания образа из внешнего реджистри в доверенные для использования?

Давайте подумаем



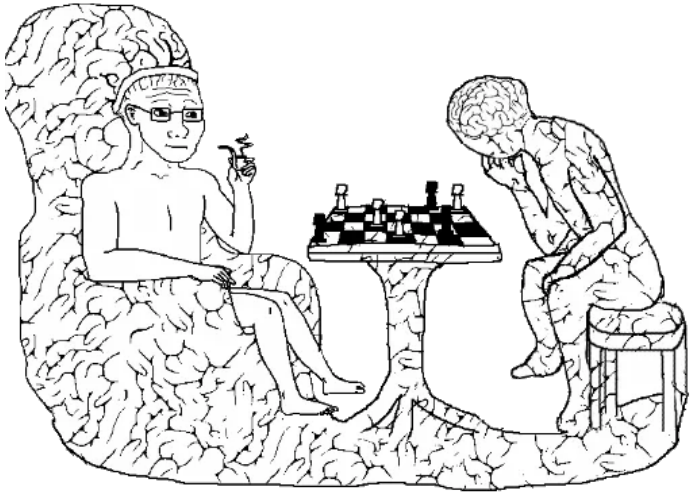
Practice (Public -> Private)

УЯЗВИМОСТИ

https://youtu.be/NFwQ-bsIxbo?t=3522

Какой концептуальный вывод?

Делая "SAST" не забывайте о "DAST"!



Всем спасибо!

Вопросы?