Container Hardening

Overview of standards and RETEX

Arcelone 2025

Container Hardening

Containers, DockerD, compose, CI/CD pipeline

Arcelone 2025

Container Hardening

Containers, DockerD, compose, ~~CI/CD pipeline~~

Arcelone 2025

What are we going to harden ?

- Images
- Containers
- Architecture
- Docker Daemon
- ~~CI/CD build stage~~

Standards

Standards

- ANSSI
  - 16 rules
  - container, architecture, (dockerD--, host--)
- CIS
  - 118 rules (300 pages)
  - Image, container, architecture, dockerD++, swarm mode, host
  - 2 Levels of security (Does or does not inhibit the utility of the technology)

NIST SP 800-190 Application Container Security Guide

Standards

ANSSI

- R1 Isoler les systèmes sensibles de fichiers de l'hôte
- R2 Restreindre l'accès aux périphériques de l'hôte
- R3 Interdire la connexion du conteneur au réseau bridge docker0
- R4 Isoler l'interface réseau de l'hôte
- R5 Créer un réseau dédié pour chaque connexion
- R6 Dédier des namespaces PID, IPC et UTS pour chaque conteneur
- R7 Dédier un namespace USER ID pour chaque conteneur
- R7+ Restreindre la création des Namespace USER ID à l'utilisateur root
- R7- Restreindre le partage du Namespace USER ID de l'hôte
- R8 Interdire l'utilisation des capabilities
- R8- Limiter l'utilisation des capabilities
- R9 Dédier les Control groups pour chaque conteneur
- R10 Limiter l'utilisation de la mémoire de l'hôte pour chaque conteneur
- R11 Limiter l'utilisation du CPU de l'hôte pour chaque conteneur
- R12 Restreindre en lecture le système de fichiers racine de chaque conteneur
- R12- Limiter l'écriture de l'espace de stockage de chaque conteneur
- R12- - Limiter l'écriture de l'espace de stockage de l'ensemble des conteneurs
- R13 Créer un système de stockage pour les données non persistantes
- R14 Créer un système de stockage pour les données persistantes ou partagées
- R15 Restreindre l'accès aux répertoires et aux fichiers sensibles
- R16 Exporter les journaux avec Docker
- R16- Exporter les journaux depuis l'intérieur du conteneur

Standards

ANSSI - EASY

- R1 Isoler les systèmes sensibles de fichiers de l’hôte
- R2 Restreindre l’accès aux périphériques de l’hôte
- R3 Interdire la connexion du conteneur au réseau bridge docker0
- R4 Isoler l’interface réseau de l’hôte
- R6 Dédier des namespaces PID, IPC et UTS pour chaque conteneur
- R9 Dédier les Control groups pour chaque conteneur
- R15 Restreindre l’accès aux répertoires et aux fichiers sensibles
- R16 Exporter les journaux avec Docker
- R16- Exporter les journaux depuis l’intérieur du conteneur

Standards

ANSSI - EASY

• R1 Isoler les systèmes sensibles de fichiers de l'hôte	✓Ne pas monter /, /dev, /proc, /sys, et ne pas utiliser --privileged
• R2 Restreindre l'accès aux périphériques de l'hôte	✓Rien à faire, --device si besoin
• R4 Isoler l'interface réseau de l'hôte	✓Ne pas utiliser --network host
• R6 Dédier des namespaces PID, IPC et UTS pour chaque conteneur	✓Ne pas utiliser --pid, --ipc et --uts
• R9 Dédier les Control groups pour chaque conteneur	✓ Ne pas utiliser --cgroup-parent

Standards

ANSSI - EASY

- R3 Interdire la connexion du conteneur au réseau bridge docker0
- R15 Restreindre l'accès aux répertoires et aux fichiers sensibles
- R16 Exporter les journaux avec Docker
- R16- Exporter les journaux depuis l'intérieur du conteneur

✓ Ajouter "bridge": "none", ou "icc": false dans daemon.json

✓ Restreindre au strict minimum (et lecture seule)

✓ logging drivers -> RTFM -> splunk, gelf, AWS, label, regex, TLS ....

- Global (daemon) or override per container
- From inside the container in case the log are not STDERR or STDOUT (symlink /var/log)

Standards

ANSSI - EASY

- EX of compose

```yaml
services:
  proxy:
    image: nginx:latest
    logging:
      driver: syslog
      options:
        syslog-address: "tcp://192.168.0.42:123"
        tag: "nginx"
    ports:
      - "80:80"
```

Standards

ANSSI - MEDIUM

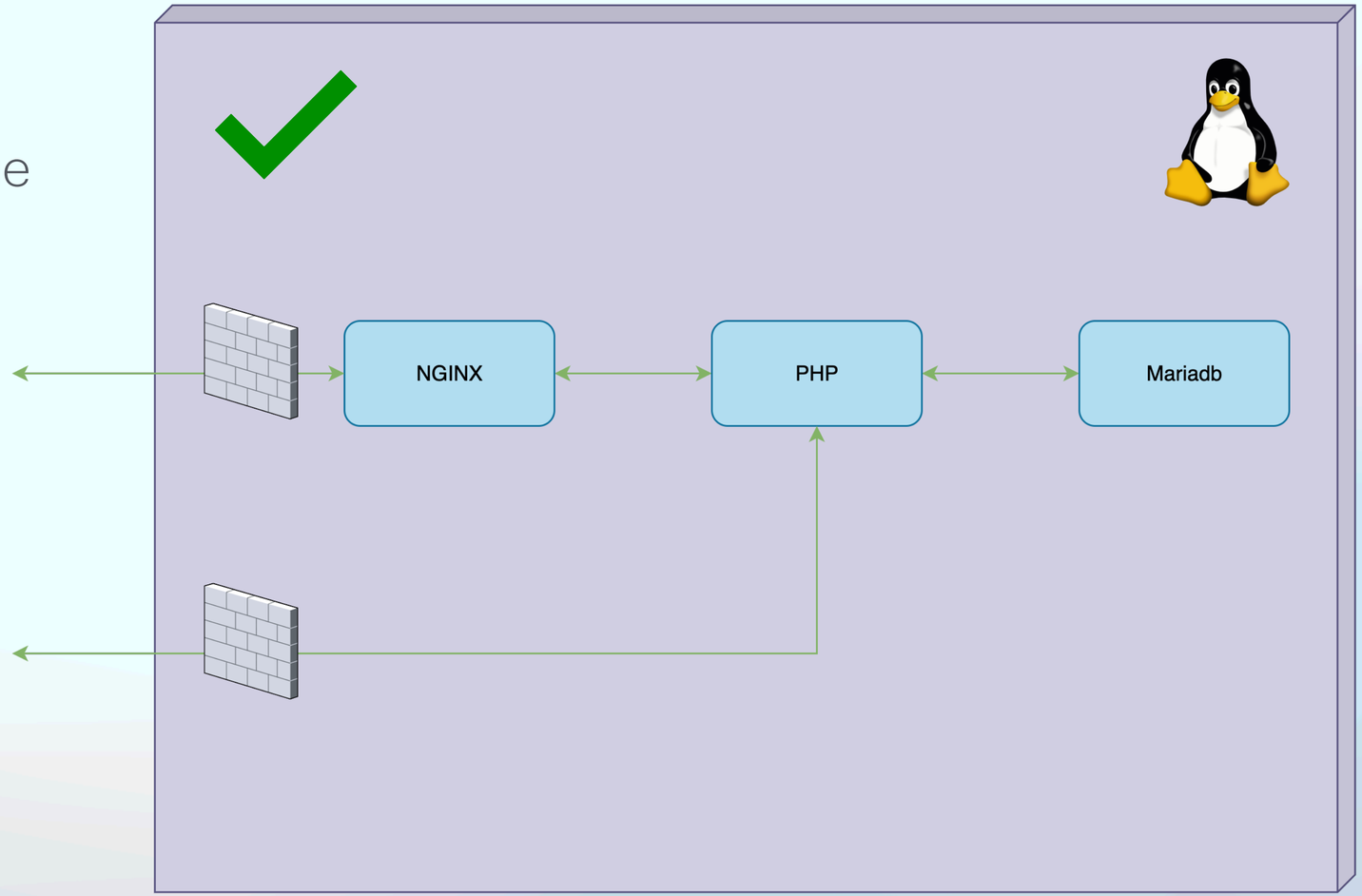
- R5 Créer un réseau dédié pour chaque connexion
- R7 Dédier un namespace USER ID pour chaque conteneur
- R7+ Restreindre la création des Namespace USER ID à l’utilisateur root
- R7- Restreindre le partage du Namespace USER ID de l’hôte
- R10 Limiter l’utilisation de la mémoire de l’hôte pour chaque conteneur
- R11 Limiter l’utilisation du CPU de l’hôte pour chaque conteneur
- R14 Créer un système de stockage pour les données persistantes ou partagées

Standards

ANSSI - MEDIUM

- R5 Créer un réseau dédié pour chaque connexion

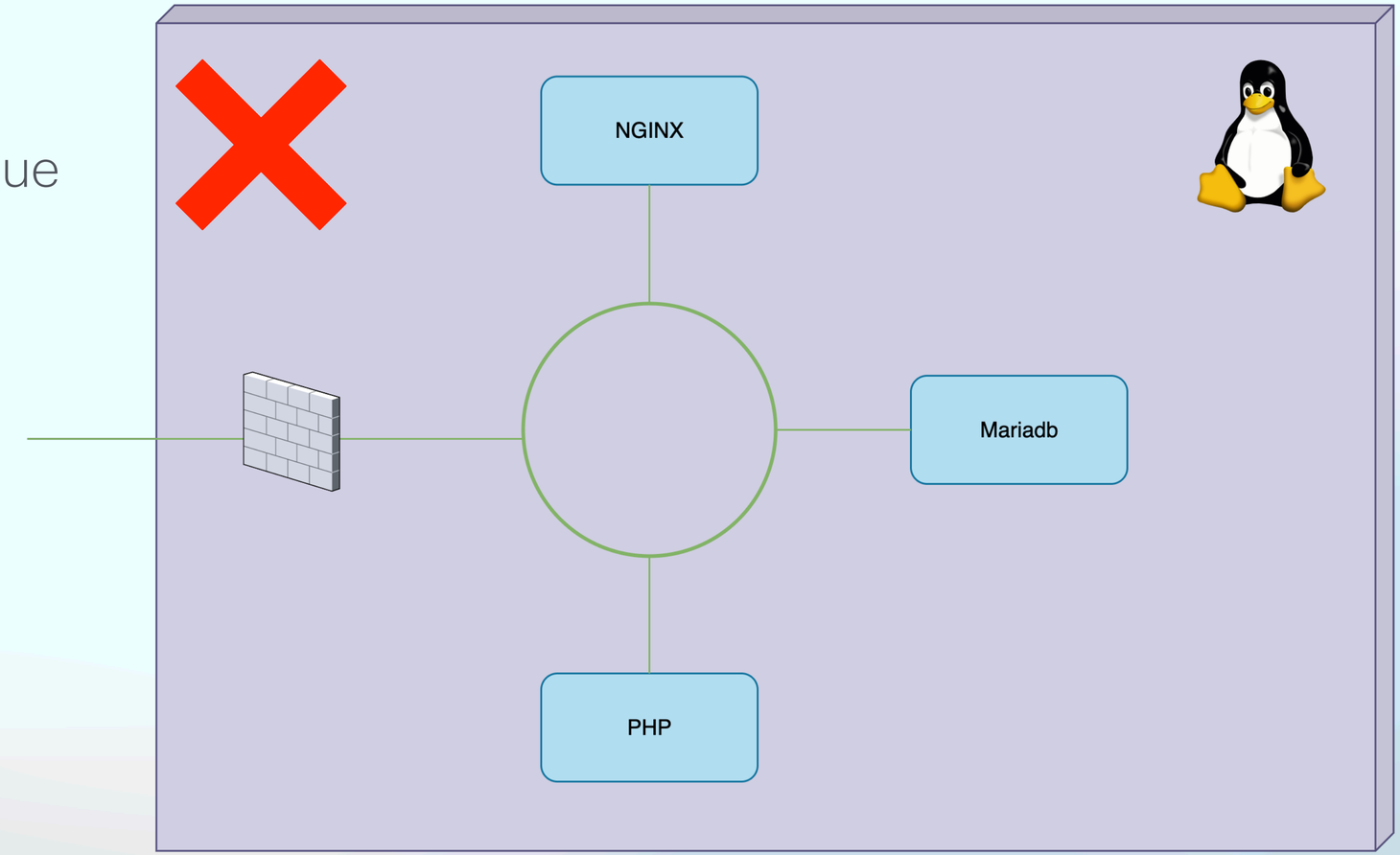
Relation 1:1



Standards

ANSSI - MEDIUM

- R5 Créer un réseau dédié pour chaque connexion



Standards

ANSSI - MEDIUM

- R7 Dédier un namespace USER ID pour chaque conteneur
- R7+ Restreindre la création des Namespace USER ID à l'utilisateur root
- R7- Restreindre le partage du Namespace USER ID de l'hôte

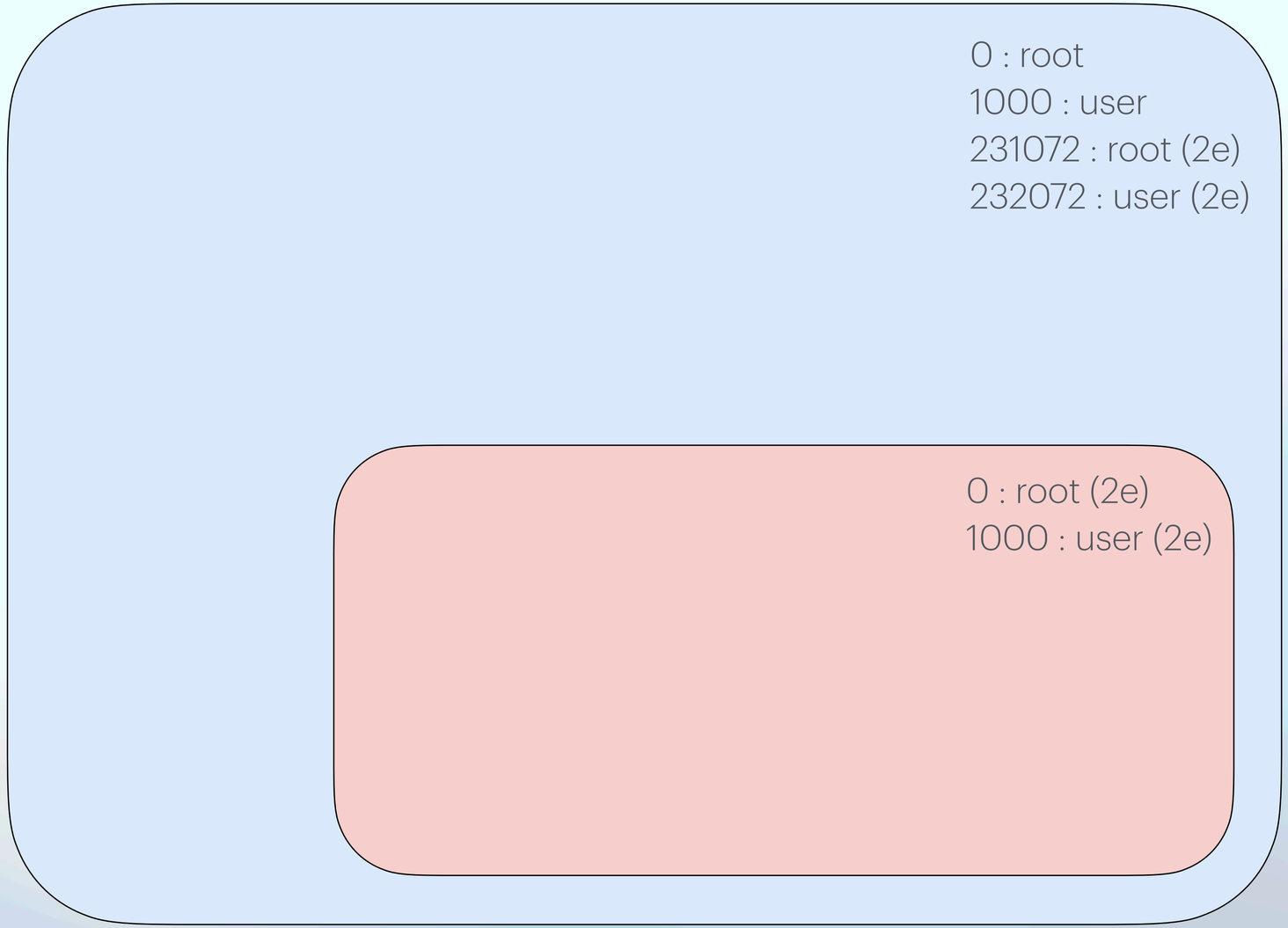
```
#/etc/subuid and /etc/subgid
dockremap:231072:65536
```

```
#daemon.json
{
  "userns-remap": "default"
}
```

- Cf: https://docs.docker.com/engine/security/userns-remap/

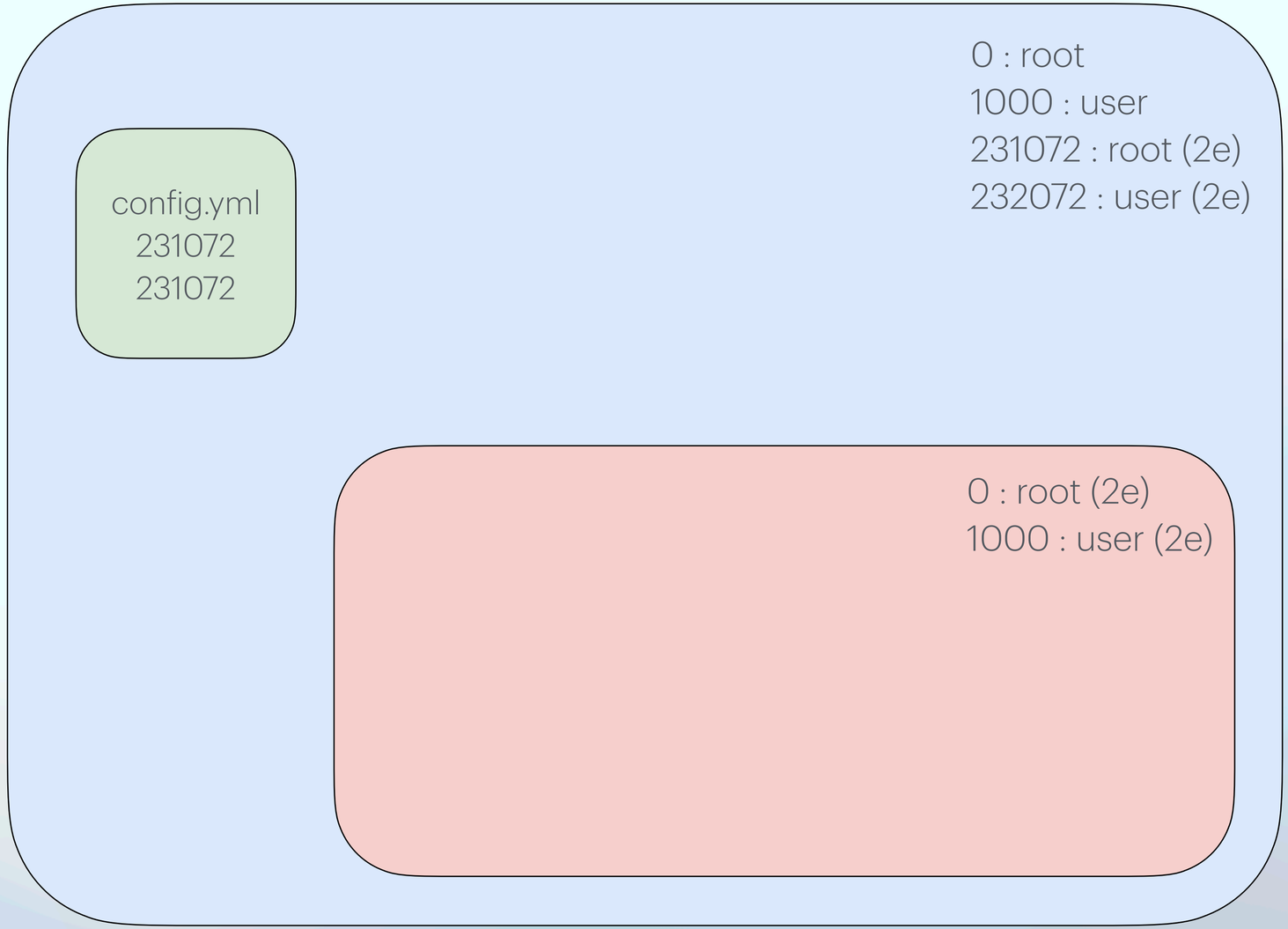
Standards

ANSSI - MEDIUM



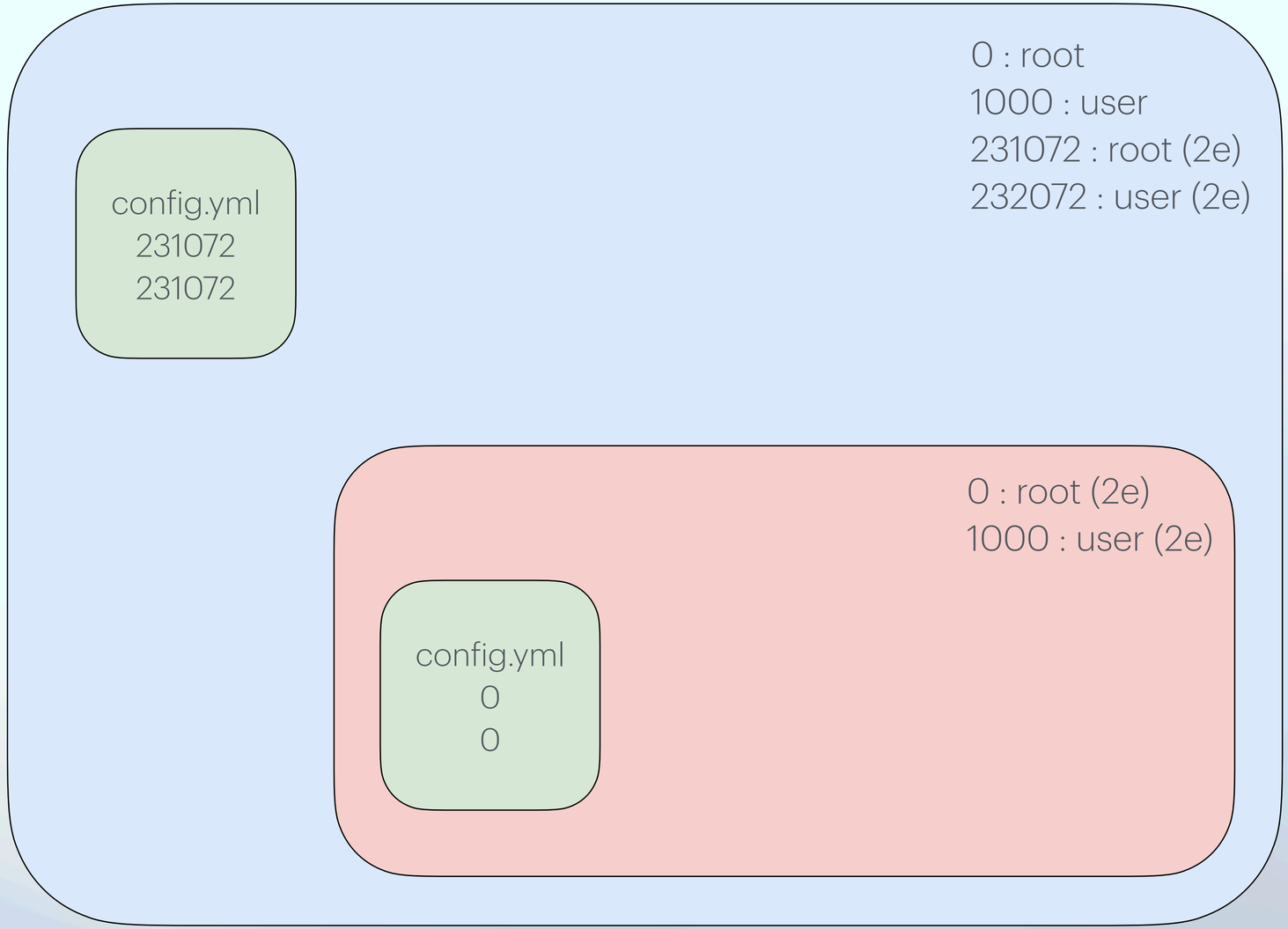
Standards

ANSSI - MEDIUM



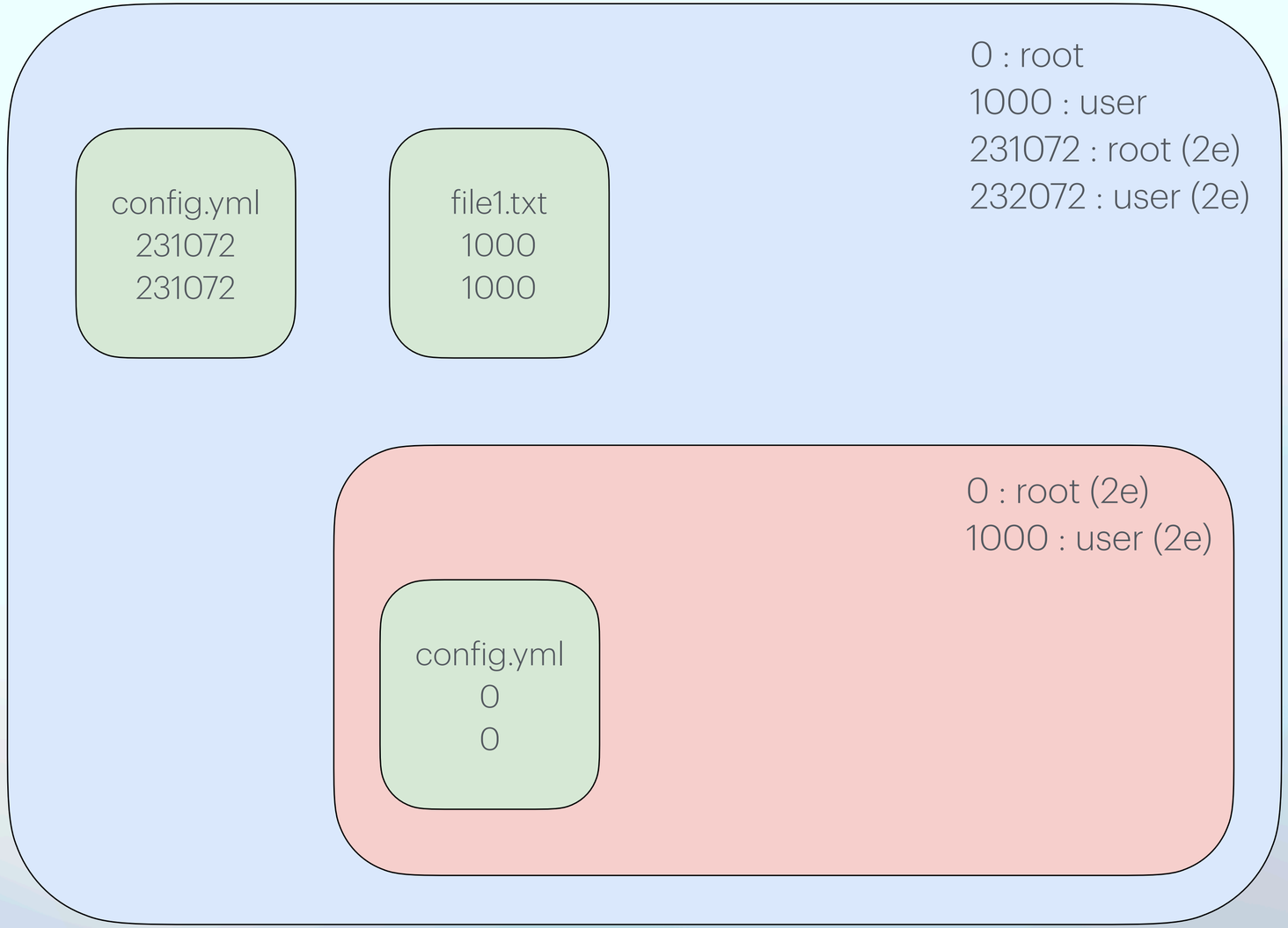
Standards

ANSSI - MEDIUM



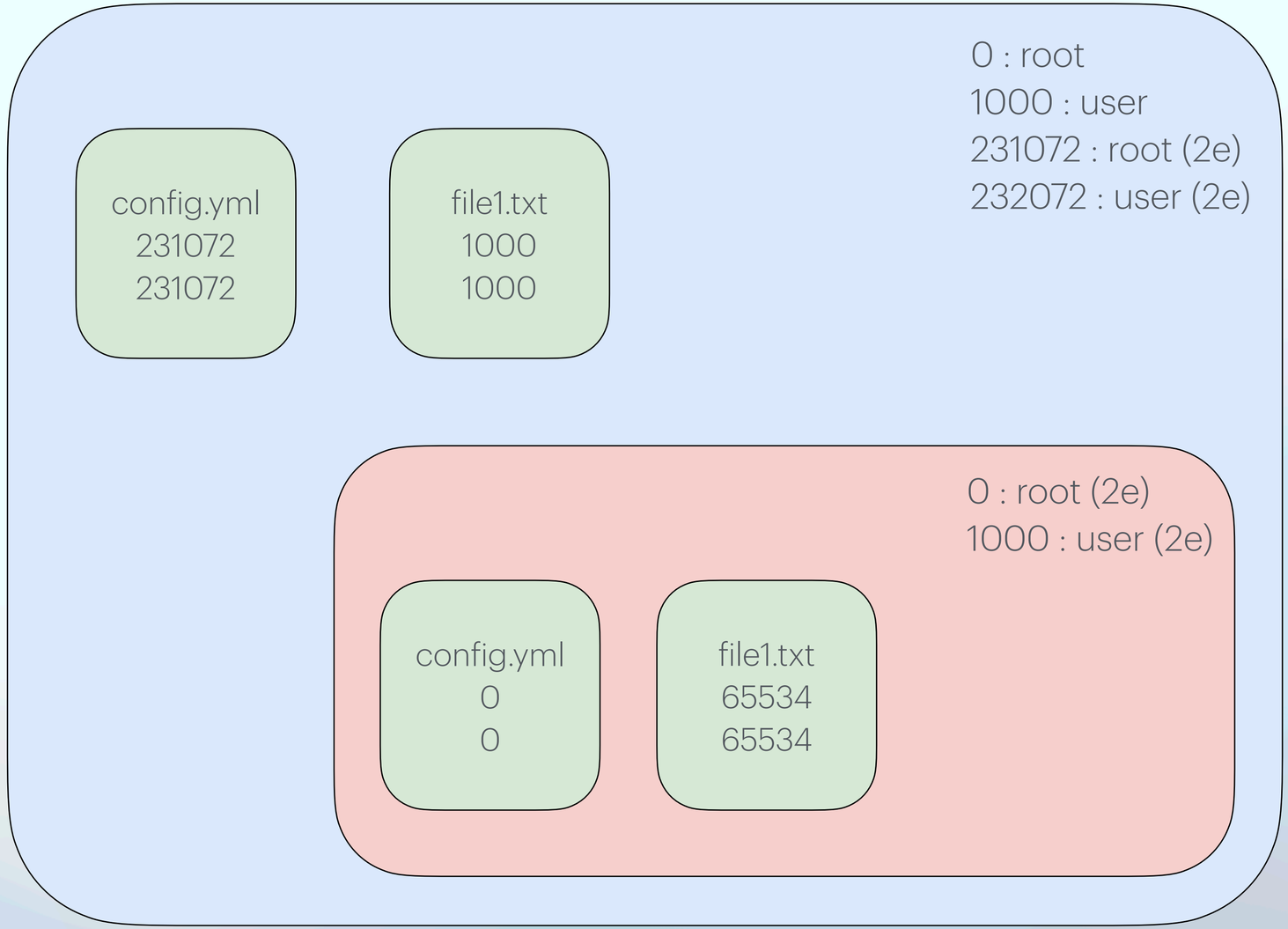
Standards

ANSSI - MEDIUM



Standards

ANSSI - MEDIUM



Standards

ANSSI - MEDIUM

- R7 Dédier un namespace USER ID pour chaque conteneur
- R7+ Restreindre la création des Namespace USER ID à l'utilisateur root
- R7- Restreindre le partage du Namespace USER ID de l'hôte

```
sysctl kernel.unprivileged_userns_clone

echo "kernel.unprivileged_userns_clone = 0" \
    >> /etc/sysctl.d/99-unpriv_userns.conf
```

```
#docker-compose.yaml
userns_mode: "host"
```

- Cf: https://docs.docker.com/engine/security/userns-remap/

Standards

ANSSI - MEDIUM

- R10 Limiter l’utilisation de la mémoire de l’hôte pour chaque conteneur
- R11 Limiter l’utilisation du CPU de l’hôte pour chaque conteneur

```
mem_limit: 300m
memswap_limit: 300m

Bonus:
    mem_reservation: 200m
    cpus: '0.030'
    cpu_period: '0.030'
    cpu_quota: '0.030'

Bonus:
    cpu_percent: #(Win only?)
    cpu_shares:
    cpu_count: #(Win only?)
    cpu_rt_runtime:
    cpu_rt_period:
```

if memory="300m" and memswap_limit="1g", the container can use 300m of memory and 700m (1g - 300m) swap.

Standards

ANSSI - MEDIUM

- R14 Créer un système de stockage pour les données persistantes ou partagées

```
volumes:
  - text:/var/run/txt/:roZ
  - type: volume
    source: db-data
    target: /data
    volume:
      nocopy: true
      subpath: sub
  - type: bind
    source: /opt/config.yml
    target: /etc/config.yml
    bind:
      selinux: Z
```

Penser à secret et config pour compose

Standards

ANSSI - MEDIUM

- EX de compose

```yaml
services:
  proxy:
    image: nginx:stable-alpine
    ports:
      - "192.128.1.15:80:80"
    mem_limit: 200m
    memswap_limit: 400m
    mem_reservation: 20m
    cpus: '0.020'
    volumes:
      - ./proxy/nginx.conf:/etc/nginx/conf.d/default.conf:ro
    networks:
      - front
      - back
  php:
    image: php:8.3.0-fpm
    mem_limit: 200m
    memswap_limit: 400m
    mem_reservation: 20m
    cpus: '0.020'
    volumes:
      - ./proxy/code.php:/var/www/html/code.php:ro
    networks:
      - front
      - back-db
  db:
    image: mysql:8.4
    mem_limit: 200m
    memswap_limit: 400m
    mem_reservation: 20m
    cpus: '0.020'
    volumes:
      - db-data:/var/lib/mysql
    networks:
      back-db

volumes:
  db-data:

networks:
  front:
  back:
    internal: true
  back-db:
    internal: true
```

Standards

ANSSI - HARD

- R8 Interdire l’utilisation des capabilities
- R8- Limiter l’utilisation des capabilities
- R12 Restreindre en lecture le système de fichiers racine de chaque conteneur
- R12- Limiter l’écriture de l’espace de stockage de chaque conteneur
- R12- - Limiter l’écriture de l’espace de stockage de l’ensemble des conteneurs
- R13 Créer un système de stockage pour les données NON persistantes

Standards

ANSSI - HARD

- R12 Restreindre en lecture le système de fichiers racine de chaque conteneur
- R12- Limiter l’écriture de l’espace de stockage de chaque conteneur
- R12- - Limiter l’écriture de l’espace de stockage de l’ensemble des conteneurs

- Dedicated partition for /var/lib/docker

Standards

ANSSI - HARD

- R13 Créer un système de stockage pour les données non persistantes

```
tmpfs:
    - <path>:<options>

tmpfs:
    - /tmp:noexec,nodev,nosuid,mode=1700,size=64m,uid=<YourApp>,gid=<YourApp>
```

- https://docs.docker.com/engine/storage/tmpfs/#options-for---tmpfs

Default to mode=1777

Standards

ANSSI - HARD

- R8 Interdire l’utilisation des capabilities
- R8- Limiter l’utilisation des capabilities

```
cap_drop:
    - ALL
# If needed
cap_add:
    - CAP_CHOWN # Or other
```

- https://docs.docker.com/engine/security/#linux-kernel-capabilities
- https://github.com/moby/moby/blob/master/daemon/pkg/oci/caps/defaults.go#L6-L19
- https://docs.docker.com/engine/security/seccomp/
- https://github.com/moby/profiles/blob/main/seccomp/default.json
- https://man7.org/linux/man-pages/man7/capabilities.7.html

Capability Key	Capability Description
AUDIT_WRITE	Write records to kernel auditing log.
CHOWN	Make arbitrary changes to file UIDs and GIDs (see chown(2)).
DAC_OVERRIDE	Bypass file read, write, and execute permission checks.
FOWNER	Bypass permission checks on operations that normally require the file system UID of the process to match the UID of the file.
FSETID	Don't clear set-user-ID and set-group-ID permission bits when a file is modified.
KILL	Bypass permission checks for sending signals.
MKNOD	Create special files using mknod(2).
NET_BIND_SERVICE	Bind a socket to internet domain privileged ports (port numbers less than 1024).
NET_RAW	Use RAW and PACKET sockets.
SETFCAP	Set file capabilities.
SETGID	Make arbitrary manipulations of process GIDs and supplementary GID list.
SETPCAP	Modify process capabilities.
SETUID	Make arbitrary manipulations of process UIDs.
SYS_CHROOT	Use chroot(2), change root directory.

https://docs.docker.com/engine/containers/run/#runtime-privilege-and-linux-capabilities

Capabilities

How to check (even in minimal container)

- CapPrm (Permitted)
- CapEff (Effective)
- CapBnd (Bounding)
- CapInh (Inherited)
- CapAmb (Ambient)

```
$ capsh --decode=3
0x0000000000000003=cap_chown,cap_dac_override
```

```
cat /proc/1/status

Name:   sh
Umask:  0022
State:  S (sleeping)
Tgid:   1
Ngid:   0
Pid:    1
PPid:   0
TracerPid:  0
Uid:    0   0   0   0
Gid:    0   0   0   0
CapInh: 0000000000000000
CapPrm: 00000000a80425fb
CapEff: 00000000a80425fb
CapBnd: 00000000a80425fb
CapAmb: 0000000000000000
<......>
```

Standards

ANSSI - HARD

- EX de compose

```yaml
db:
  image: mysql:8.4
  read-only: true
  mem_limit: 200m
  memswap_limit: 400m
  mem_reservation: 20m
  cpus: '0.020'
  cap_drop:
    - ALL
  cap_add:
    - CAP_SETGID
    - CAP_SETUID
    - CAP_CHOWN
    - CAP_DAC_OVERRIDE
  tmpfs:
    - /tmp:rw,noexec,nosuid,nodev,mode=1700,uid=999,gid=999,size=16m
    - /var/run/mysqld/:rw,noexec,nosuid,nodev,mode=1700,uid=999,gid=999,size=16m
  environment:
    MYSQL_ROOT_PASSWORD_FILE: /run/secrets/db-root-password
    MYSQL_DATABASE_FILE: /run/secrets/db-database
    MYSQL_USER_FILE: /run/secrets/db-user
    MYSQL_PASSWORD_FILE: /run/secrets/db-password
  secrets:
    - db-root-password
    - db-database
    - db-user
    - db-password
  volumes:
    - db-data:/var/lib/mysql
  networks:
    back
```

Standards

CIS - Container

- 5.9 Ensure that only needed ports are open on the container
- 5.14 Ensure that incoming container traffic is bound to a specific host interface
- 5.15 Ensure that the 'on-failure' container restart policy is set to (max) '5'
- 5.19 Ensure that the default ulimit is overwritten at runtime if needed
- 5.27 Ensure that container health is checked at runtime
- 5.29 Ensure that the PIDs cgroup limit is used
- 5.32 Ensure that the Docker socket is not mounted inside any containers

Standards

CIS - Container

- 5.14 Ensure that incoming container traffic is bound to a specific host interface
- 5.15 Ensure that the 'on-failure' container restart policy is set to (max) ‘5'
- 5.29 Ensure that the PIDs cgroup limit is used

```
# [IP:][port | range]:(port | range)[/PROTOCOL]
ports:
    - "127.0.0.1:5000-5010:5000-5010/udp"

restart: on-failure:5

pids_limit: 10
```

Standards

CIS - Container

- 5.9 Ensure that only needed ports are open on the container
- 5.19 Ensure that the default ulimit is overwritten at runtime if needed
- 5.27 Ensure that container health is checked at runtime
- 5.32 Ensure that the Docker socket is not mounted inside any containers

- Privilégier une configuration coté daemon
- Ajouter un health check de façon externe si absent de l’image

Standards

CIS - Container

- EX de compose

```yaml
services:
  db:
    image: mysql:8.4
    read-only: true
    restart: on-failure:5
    mem_limit: 200m
    memswap_limit: 400m
    mem_reservation: 20m
    pids_limit: 50
    cpus: '0.020'
    port:
      - "192.168.1.15:3126:3126/tcp"
    cap_drop:
      - ALL
    cap_add:
      - CAP_SETGID
      - CAP_SETUID
      - CAP_CHOWN
      - CAP_DAC_OVERRIDE
    tmpfs:
      - /tmp:rw,noexec,nosuid,nodev,mode=1700,uid=999,gid=999,size=16m
      - /var/run/mysqld/:rw,noexec,nosuid,nodev,mode=1700,uid=999,gid=999,size=16m
    volumes:
      - db-data:/var/lib/mysql
```

DockerD

DockerD

- CIS Chapter 2 and 3
  - 43 rules (19 for general config, 24 for config file)
  - Installation in rootless mode

DockerD

- Installation in rootless mode
  - br_netfilter may be required

```
{
  "icc": false,
  "log-level": "info",
  "userns-remap": "default",
  "log-driver": "syslog",
  "log-opts": {
    "syslog-address": "udp://1.2.3.4:1111"
  },
  "default-ulimits": {
    "nofile": {
      "Name": "nofile",
      "Hard": 64000,
      "Soft": 48000
    }
  },
  "no-new-privileges": true,
  "live-restore": true,
  "userland-proxy": false,
  "selinux-enabled": true # If applicable
}
```

CIS - Images

Standards

CIS

- 4.1 Ensure that a user for the container has been created
- 4.2 Ensure that containers use only trusted base images
- 4.3 Ensure that unnecessary packages are not installed in the container
- 4.4 Ensure images are scanned and rebuilt to include security patches
- 4.5 Ensure Content trust for Docker is Enabled
- 4.6 Ensure that HEALTHCHECK instructions have been added to container images
- 4.7 Ensure update instructions are not used alone in Dockerfiles
- 4.8 Ensure setuid and setgid permissions are removed
- 4.9 Ensure that COPY is used instead of ADD in Dockerfiles
- 4.10 Ensure secrets are not stored in Dockerfiles
- 4.11 Ensure only verified packages are installed
- 4.12 Ensure all signed artifacts are validated

Standards

CIS - EASY

- 4.1 Ensure that a user for the container has been created
- 4.3 Ensure that unnecessary packages are not installed in the container
- 4.6 Ensure that HEALTHCHECK instructions have been added to container images
- 4.7 Ensure update instructions are not used alone in Dockerfiles
- 4.8 Ensure setuid and setgid permissions are removed
- 4.9 Ensure that COPY is used instead of ADD in Dockerfiles
- 4.10 Ensure secrets are not stored in Dockerfiles

Standards

CIS - EASY

- 4.1 Ensure that a user for the container has been created

```
# Dockerfile, containerfile
RUN groupadd -r -g 999 myapp && \
    useradd -r -u 999 -g 999 -d /var/www/html -s /sbin/nologin myapp
```

```
cat /etc/passwd
cat /etc/group
ls /proc
cat /proc/<x>/status
```

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginx

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginx:stable

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginx:stable

debian:bookworm-slim

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginx:stable-alpine

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginx:stable-alpine-slim

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginxinc/nginx-unprivileged:stable-alpine-slim

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

nginxinc/nginx-unprivileged:stable-alpine-slim

```
FROM nginxinc/nginx-unprivileged:stable-alpine-slim@sha256:8a998104a9836b2fdb10e858d5195a0f6e6f368ce5e1748746d4865d4c1cbd3a
```

Standards

CIS - EASY

- 4.3 Ensure that unnecessary packages are not installed in the container

Standards

CIS - EASY

- 4.6 Ensure that HEALTHCHECK instructions have been added to container images

```
# Dockerfile, containerfile
HEALTHCHECK --interval=5m --timeout=3s \
  CMD curl -f http://localhost/status || exit 1
```

```
HEALTHCHECK --interval=10s --timeout=5s \
    CMD pg_isready -U ${POSTGRES_USER:-postgres} || exit 1
```

Standards

CIS - EASY

- 4.7 Ensure update instructions are not used alone in Dockerfiles

```
# https://repo1.dso.mil/dsop/opensource/debian/debian13.x/debian-13-slim/
RUN apt-get update -y && \
    apt-get upgrade -y && \
    apt install -y --no-install-recommends ca-certificates adduser && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/* \
    /var/log/dpkg.log \
    /var/log/apt/term.log \
    /var/log/apt/history.log \
    /var/cache/ldconfig/aux-cache
```

Standards

CIS - EASY

- 4.8 Ensure setuid and setgid permissions are removed

```
# CIS
RUN find / -perm /6000 -type f -exec chmod a-s {} \; || true
# https://repo1.dso.mil
RUN find / -path /proc -prune -o -type f \( -perm -4000 -o -perm -2000 \) \
    -exec echo "Found: {}" \; -exec chmod u-s,g-s,o-s {} \;
```

Standards

CIS - EASY

- 4.9 Ensure that COPY is used instead of ADD in Dockerfiles
- 4.10 Ensure secrets are not stored in Dockerfiles

- URL and remote file threat
- https://docs.docker.com/build/building/secrets/

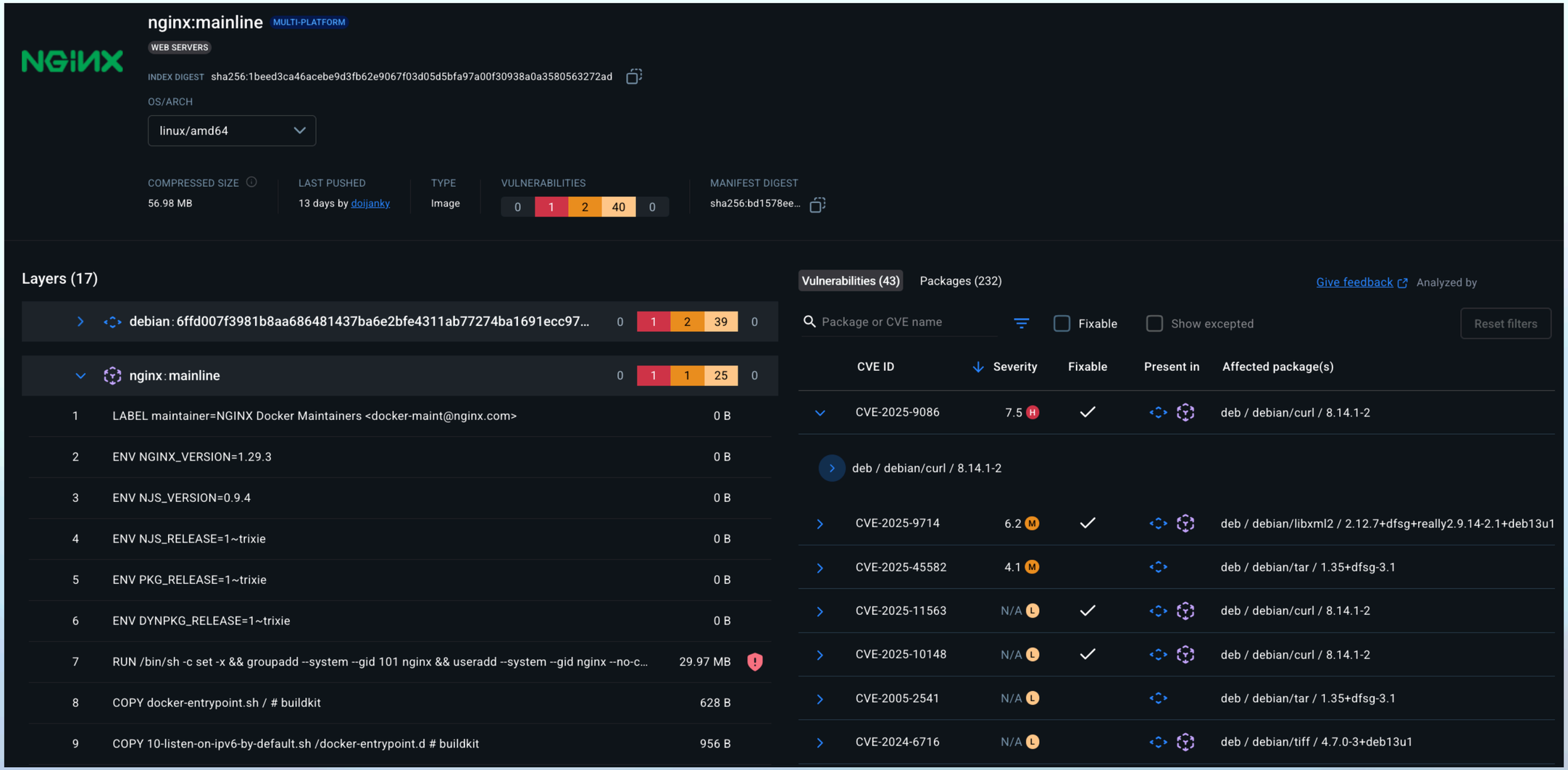
Standards

CIS - MEDIUM

- 4.4 Ensure images are scanned and rebuilt to include security patches

Standards

CIS - MEDIUM

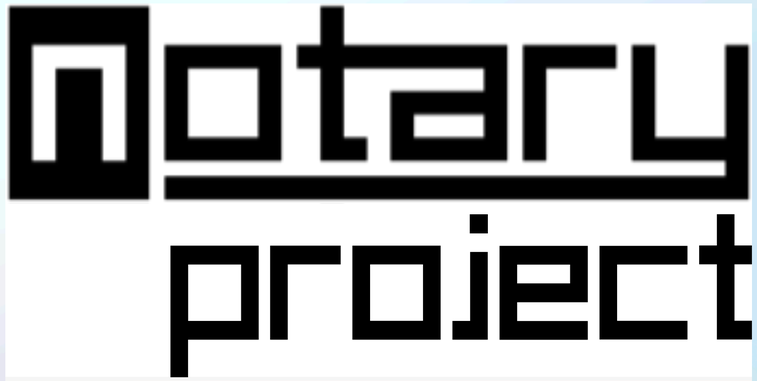


Standards

CIS - ???

- 4.2 Ensure that containers use only trusted base images
- 4.5 Ensure Content trust for Docker is Enabled
- 4.11 Ensure only verified packages are installed
- 4.12 Ensure all signed artifacts are validated

- https://github.com/notaryproject/notation
- https://notaryproject.dev/



Sources, Ressources & Tools

Sources & Ressources

- ANSSI (2020) : https://cyber.gouv.fr/publications/recommandations-de-securite-relatives-au-deploiement-de-conteneurs-docker
- CIS : https://downloads.cisecurity.org/#/ (section Docker)
- Docker sec : https://docs.docker.com/engine/security/
- Gitlab of the DoD : https://repo1.dso.mil/dsop
- Felix Dreißig (aka. F30) :
  - https://www.codecentric.de/en/knowledge-hub/blog/7-ways-to-replace-kaniko-in-your-container-image-builds
  - https://speakerdeck.com/f30/beyond-kaniko-navigating-unprivileged-container-image-creation
  - Unprivileged Image Builds: What are the Challenges and Where are we Today? - Felix Dreissig
- Yann Schepens : Et l'ANSSI dit Voici comment sécuriser un container

Tools

- CIS 1.8 : https://github.com/Arcelone/docker-bench-security
- CIS bench 1.6 : https://github.com/docker/docker-bench-security

- https://gitlab.com/gitlab-org/gitlab-runner/-/issues/38957#note_2672881352
- https://gitlab.com/gitlab-org/gitlab/-/issues/503827#note_2611675005
- https://gitlab.com/gitlab-org/gitlab-runner/-/issues/27235#note_2614870872
- https://gitlab.com/gitlab-org/gitlab-runner/-/issues/36810#note_2609007382
- https://github.com/containers/common/blob/main/pkg/seccomp/seccomp.json
- https://podman.io/docs/installation?utm_source=chatgpt.com#seccompjson