Docker Bench for Security

Docker Bench for Security is a script that runs a series of automated tests checking the container for common best-practices. They are based off the CIS Docker Benchmark. Docker bench requires Docker 1.13.0 or later in order to run.

Running Docker Bench can be accomplished in 3 Easy steps.

In this example I am going to run the script on a mysql container that I had recently pulled. You have the option to pull the Docker Security image first but you can simply run the script and it'll pull it automatically before running the checks.

For Ubuntu

1. `docker run -itd mysql`

2.
```
docker run -it --net host --pid host --userns host --cap-add audit_control \
    -e DOCKER_CONTENT_TRUST=$DOCKER_CONTENT_TRUST \
    -v /etc:/etc:ro \
    -v /usr/bin/containerd:/usr/bin/containerd:ro \
    -v /usr/bin/runc:/usr/bin/runc:ro \
    -v /usr/lib/systemd:/usr/lib/systemd:ro \
    -v /var/lib:/var/lib:ro \
    -v /var/run/docker.sock:/var/run/docker.sock:ro \
    --label docker_bench_security \
    docker/docker-bench-security
```

3. Remediate

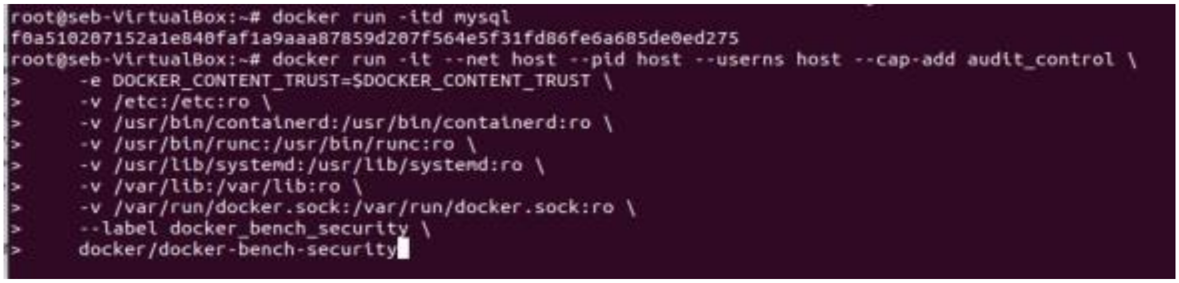
For MAC

```
docker run -it --net host --pid host --userns host --cap-add audit_control \
    -e DOCKER_CONTENT_TRUST=$DOCKER_CONTENT_TRUST \
    -v /etc:/etc \
    -v /var/lib:/var/lib:ro \
    -v /var/run/docker.sock:/var/run/docker.sock:ro \
    --label docker_bench_security \
    docker/docker-bench-security
```

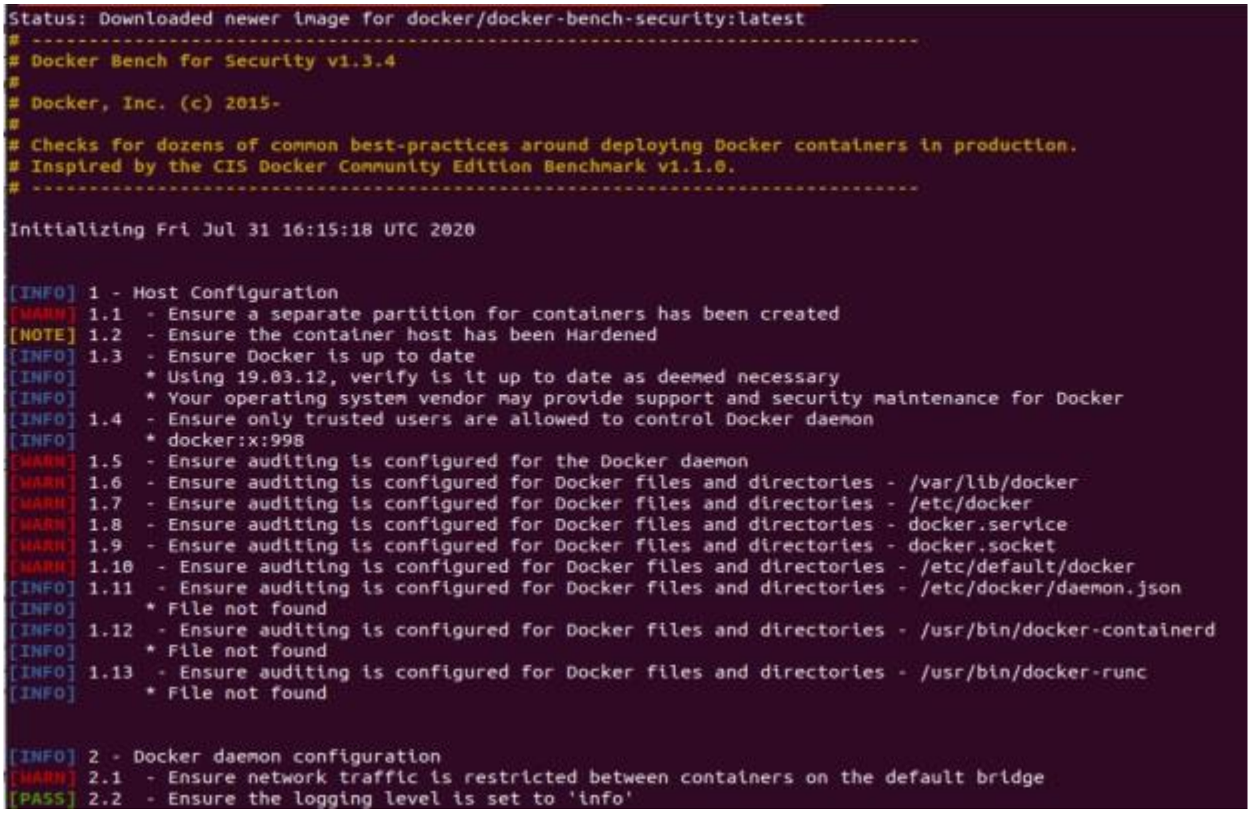
Link to the Docker Bench Github

- https://github.com/docker/docker-bench-security

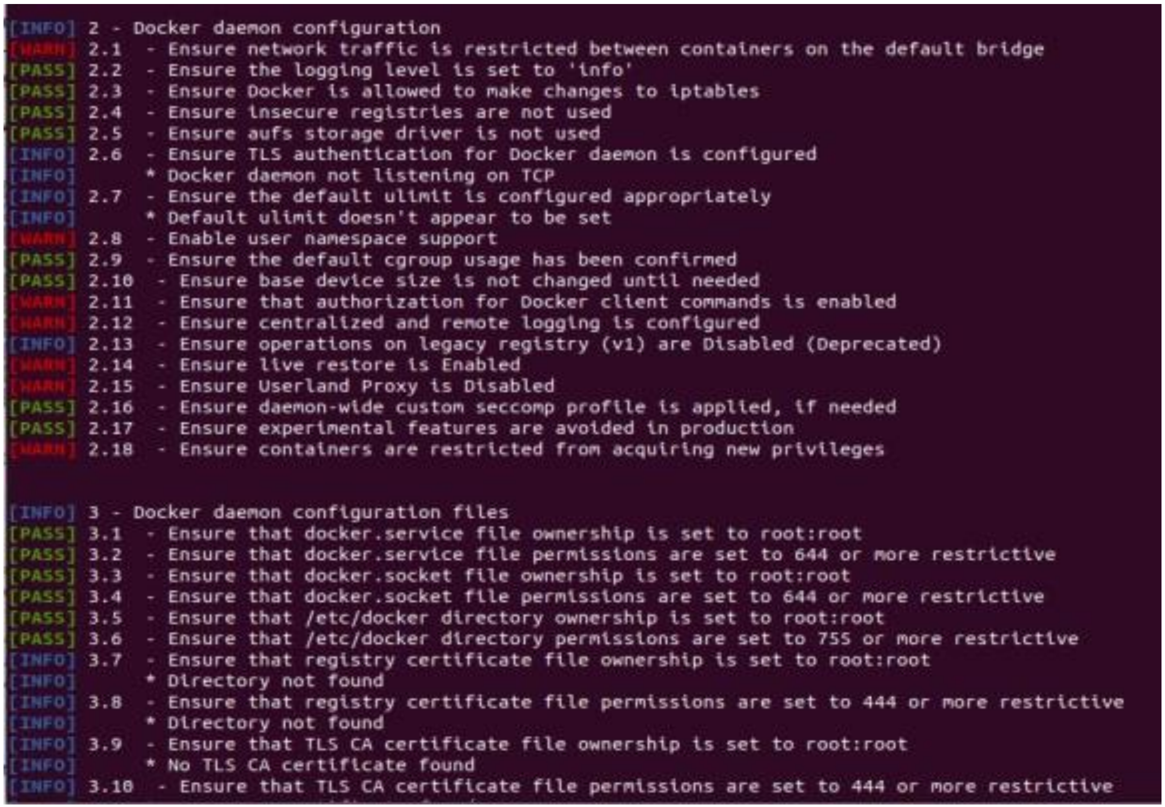
Screenshots of the output



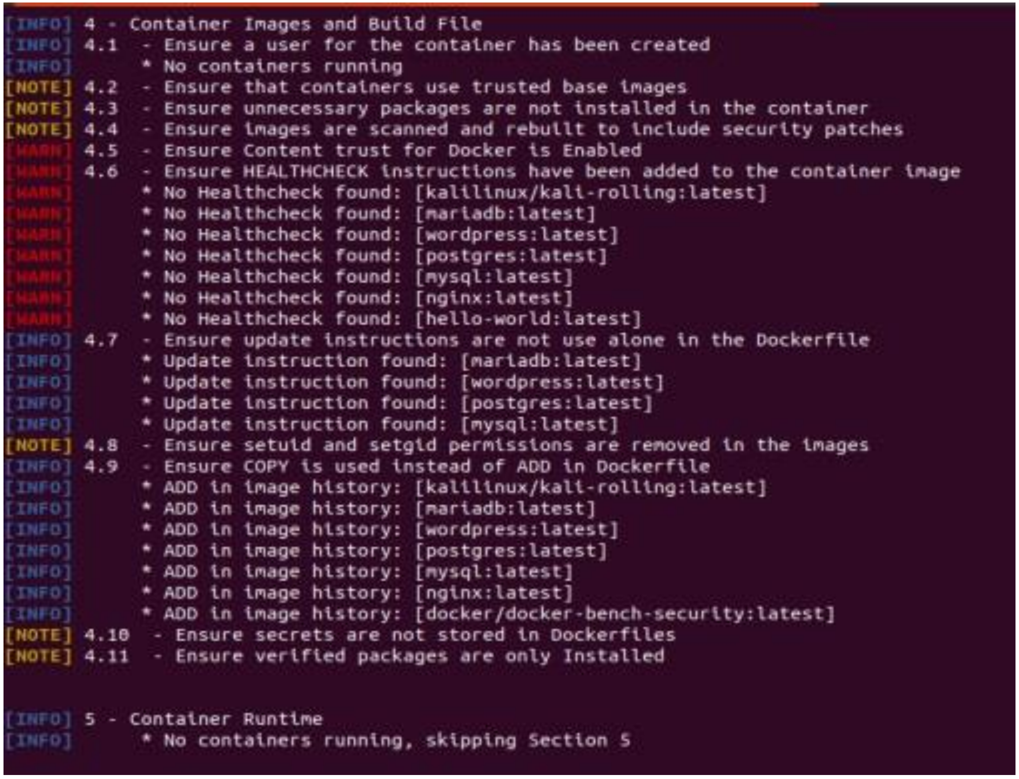
```
root@seb-VirtualBox:~# docker run -itd mysql
f0a510207152a1e840faf1a9aaa87859d207f564e5f31fd86fe6a685de0ed275
root@seb-VirtualBox:~# docker run -it --net host --pid host --userns host --cap-add audit_control \
>     -e DOCKER_CONTENT_TRUST=$DOCKER_CONTENT_TRUST \
>     -v /etc:/etc:ro \
>     -v /usr/bin/containerd:/usr/bin/containerd:ro \
>     -v /usr/bin/runc:/usr/bin/runc:ro \
>     -v /usr/lib/systemd:/usr/lib/systemd:ro \
>     -v /var/lib:/var/lib:ro \
>     -v /var/run/docker.sock:/var/run/docker.sock:ro \
>     --label docker_bench_security \
>     docker/docker-bench-security
```



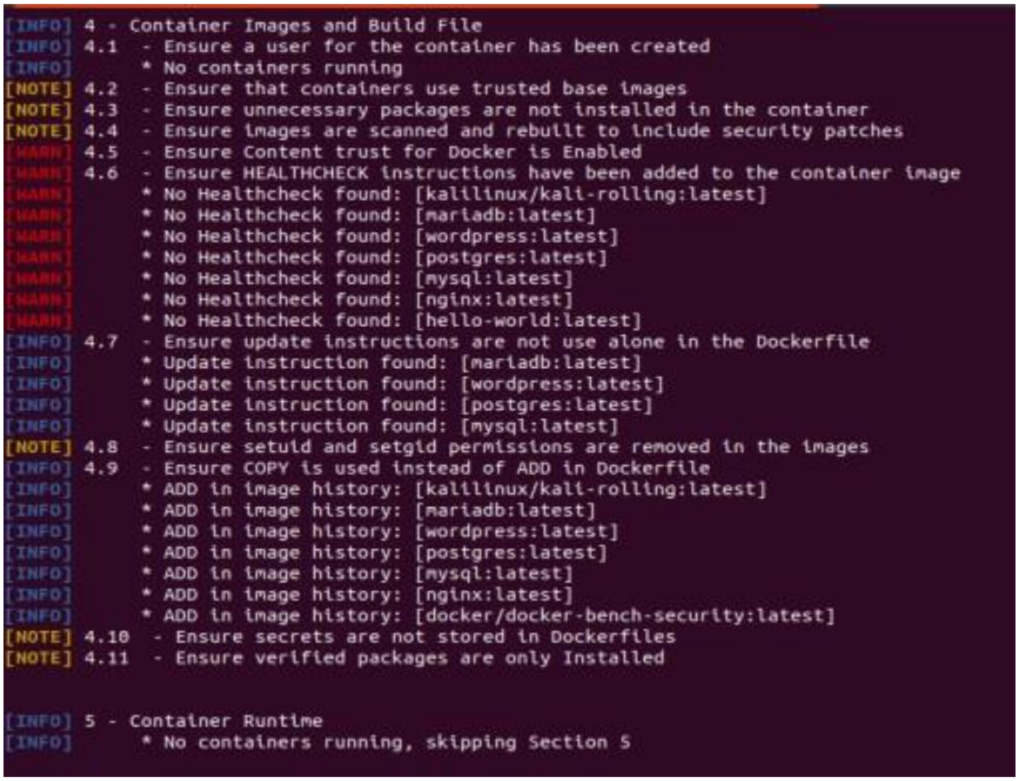
```
Status: Downloaded newer image for docker/docker-bench-security:latest
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.4
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Community Edition Benchmark v1.1.0.
# ------------------------------------------------------------------------------

Initializing Fri Jul 31 16:15:18 UTC 2020


[INFO] 1 - Host Configuration
[WARN] 1.1  - Ensure a separate partition for containers has been created
[NOTE] 1.2  - Ensure the container host has been Hardened
[INFO] 1.3  - Ensure Docker is up to date
[INFO]      * Using 19.03.12, verify is it up to date as deemed necessary
[INFO]      * Your operating system vendor may provide support and security maintenance for Docker
[INFO] 1.4  - Ensure only trusted users are allowed to control Docker daemon
[INFO]      * docker:x:998
[WARN] 1.5  - Ensure auditing is configured for the Docker daemon
[WARN] 1.6  - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.7  - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.8  - Ensure auditing is configured for Docker files and directories - docker.service
[WARN] 1.9  - Ensure auditing is configured for Docker files and directories - docker.socket
[WARN] 1.10  - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO] 1.11  - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[INFO]      * File not found
[INFO] 1.12  - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
[INFO]      * File not found
[INFO] 1.13  - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-runc
[INFO]      * File not found


[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge
[PASS] 2.2  - Ensure the logging level is set to 'info'
```



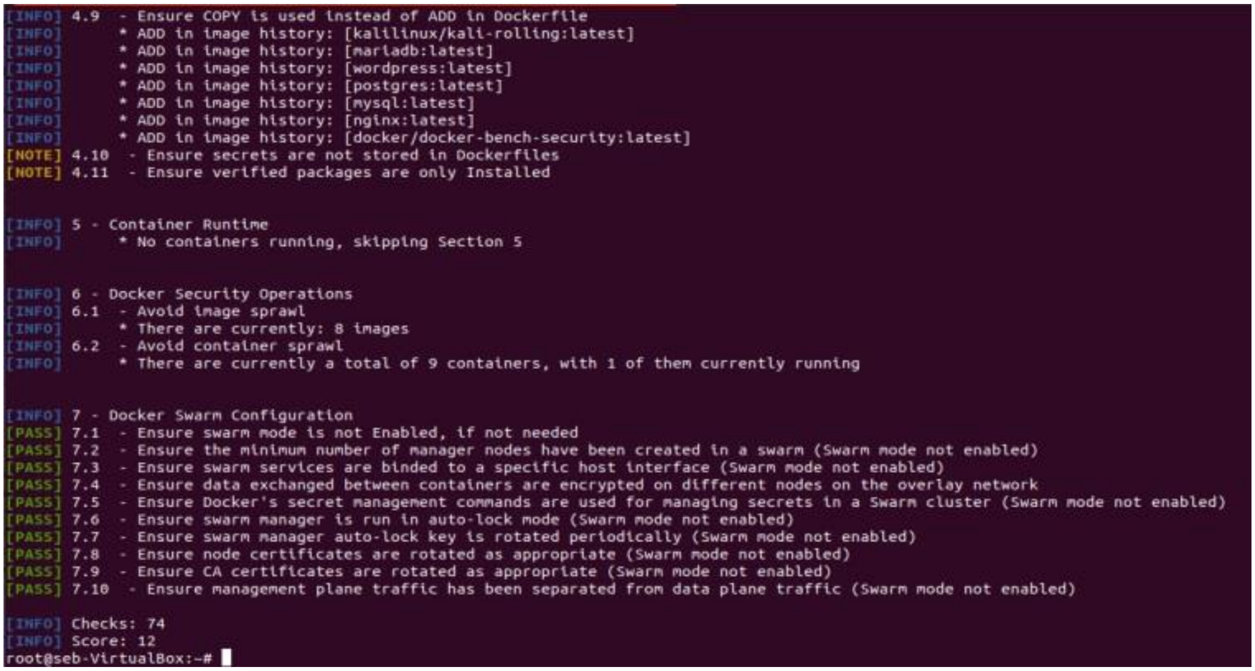
```
[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge
[PASS] 2.2  - Ensure the logging level is set to 'info'
[PASS] 2.3  - Ensure Docker is allowed to make changes to iptables
[PASS] 2.4  - Ensure insecure registries are not used
[PASS] 2.5  - Ensure aufs storage driver is not used
[INFO] 2.6  - Ensure TLS authentication for Docker daemon is configured
[INFO]      * Docker daemon not listening on TCP
[INFO] 2.7  - Ensure the default ulimit is configured appropriately
[INFO]      * Default ulimit doesn't appear to be set
[WARN] 2.8  - Enable user namespace support
[PASS] 2.9  - Ensure the default cgroup usage has been confirmed
[PASS] 2.10  - Ensure base device size is not changed until needed
[WARN] 2.11  - Ensure that authorization for Docker client commands is enabled
[WARN] 2.12  - Ensure centralized and remote logging is configured
[INFO] 2.13  - Ensure operations on legacy registry (v1) are Disabled (Deprecated)
[WARN] 2.14  - Ensure live restore is Enabled
[WARN] 2.15  - Ensure Userland Proxy is Disabled
[PASS] 2.16  - Ensure daemon-wide custom seccomp profile is applied, if needed
[PASS] 2.17  - Ensure experimental features are avoided in production
[WARN] 2.18  - Ensure containers are restricted from acquiring new privileges


[INFO] 3 - Docker daemon configuration files
[PASS] 3.1  - Ensure that docker.service file ownership is set to root:root
[PASS] 3.2  - Ensure that docker.service file permissions are set to 644 or more restrictive
[PASS] 3.3  - Ensure that docker.socket file ownership is set to root:root
[PASS] 3.4  - Ensure that docker.socket file permissions are set to 644 or more restrictive
[PASS] 3.5  - Ensure that /etc/docker directory ownership is set to root:root
[PASS] 3.6  - Ensure that /etc/docker directory permissions are set to 755 or more restrictive
[INFO] 3.7  - Ensure that registry certificate file ownership is set to root:root
[INFO]      * Directory not found
[INFO] 3.8  - Ensure that registry certificate file permissions are set to 444 or more restrictive
[INFO]      * Directory not found
[INFO] 3.9  - Ensure that TLS CA certificate file ownership is set to root:root
[INFO]      * No TLS CA certificate found
[INFO] 3.10  - Ensure that TLS CA certificate file permissions are set to 444 or more restrictive
```



```
[INFO] 4 - Container Images and Build File
[INFO] 4.1  - Ensure a user for the container has been created
[INFO]      * No containers running
[NOTE] 4.2  - Ensure that containers use trusted base images
[NOTE] 4.3  - Ensure unnecessary packages are not installed in the container
[NOTE] 4.4  - Ensure images are scanned and rebuilt to include security patches
[WARN] 4.5  - Ensure Content trust for Docker is Enabled
[WARN] 4.6  - Ensure HEALTHCHECK instructions have been added to the container image
[WARN]      * No Healthcheck found: [kalilinux/kali-rolling:latest]
[WARN]      * No Healthcheck found: [mariadb:latest]
[WARN]      * No Healthcheck found: [wordpress:latest]
[WARN]      * No Healthcheck found: [postgres:latest]
[WARN]      * No Healthcheck found: [mysql:latest]
[WARN]      * No Healthcheck found: [nginx:latest]
[WARN]      * No Healthcheck found: [hello-world:latest]
[INFO] 4.7  - Ensure update instructions are not use alone in the Dockerfile
[INFO]      * Update instruction found: [mariadb:latest]
[INFO]      * Update instruction found: [wordpress:latest]
[INFO]      * Update instruction found: [postgres:latest]
[INFO]      * Update instruction found: [mysql:latest]
[NOTE] 4.8  - Ensure setuid and setgid permissions are removed in the images
[INFO] 4.9  - Ensure COPY is used instead of ADD in Dockerfile
[INFO]      * ADD in image history: [kalilinux/kali-rolling:latest]
[INFO]      * ADD in image history: [mariadb:latest]
[INFO]      * ADD in image history: [wordpress:latest]
[INFO]      * ADD in image history: [postgres:latest]
[INFO]      * ADD in image history: [mysql:latest]
[INFO]      * ADD in image history: [nginx:latest]
[INFO]      * ADD in image history: [docker/docker-bench-security:latest]
[NOTE] 4.10  - Ensure secrets are not stored in Dockerfiles
[NOTE] 4.11  - Ensure verified packages are only Installed


[INFO] 5 - Container Runtime
[INFO]      * No containers running, skipping Section 5
```



```
[INFO] 4 - Container Images and Build File
[INFO] 4.1  - Ensure a user for the container has been created
[INFO]      * No containers running
[NOTE] 4.2  - Ensure that containers use trusted base images
[NOTE] 4.3  - Ensure unnecessary packages are not installed in the container
[NOTE] 4.4  - Ensure images are scanned and rebuilt to include security patches
[WARN] 4.5  - Ensure Content trust for Docker is Enabled
[WARN] 4.6  - Ensure HEALTHCHECK instructions have been added to the container image
[WARN]      * No Healthcheck found: [kalilinux/kali-rolling:latest]
[WARN]      * No Healthcheck found: [mariadb:latest]
[WARN]      * No Healthcheck found: [wordpress:latest]
[WARN]      * No Healthcheck found: [postgres:latest]
[WARN]      * No Healthcheck found: [mysql:latest]
[WARN]      * No Healthcheck found: [nginx:latest]
[WARN]      * No Healthcheck found: [hello-world:latest]
[INFO] 4.7  - Ensure update instructions are not use alone in the Dockerfile
[INFO]      * Update instruction found: [mariadb:latest]
[INFO]      * Update instruction found: [wordpress:latest]
[INFO]      * Update instruction found: [postgres:latest]
[INFO]      * Update instruction found: [mysql:latest]
[NOTE] 4.8  - Ensure setuid and setgid permissions are removed in the images
[INFO] 4.9  - Ensure COPY is used instead of ADD in Dockerfile
[INFO]      * ADD in image history: [kalilinux/kali-rolling:latest]
[INFO]      * ADD in image history: [mariadb:latest]
[INFO]      * ADD in image history: [wordpress:latest]
[INFO]      * ADD in image history: [postgres:latest]
[INFO]      * ADD in image history: [mysql:latest]
[INFO]      * ADD in image history: [nginx:latest]
[INFO]      * ADD in image history: [docker/docker-bench-security:latest]
[NOTE] 4.10  - Ensure secrets are not stored in Dockerfiles
[NOTE] 4.11  - Ensure verified packages are only Installed


[INFO] 5 - Container Runtime
[INFO]      * No containers running, skipping Section 5
```



```
[INFO] 4.9  - Ensure COPY is used instead of ADD in Dockerfile
[INFO]      * ADD in image history: [kalilinux/kali-rolling:latest]
[INFO]      * ADD in image history: [mariadb:latest]
[INFO]      * ADD in image history: [wordpress:latest]
[INFO]      * ADD in image history: [postgres:latest]
[INFO]      * ADD in image history: [mysql:latest]
[INFO]      * ADD in image history: [nginx:latest]
[INFO]      * ADD in image history: [docker/docker-bench-security:latest]
[NOTE] 4.10  - Ensure secrets are not stored in Dockerfiles
[NOTE] 4.11  - Ensure verified packages are only Installed


[INFO] 5 - Container Runtime
[INFO]      * No containers running, skipping Section 5


[INFO] 6 - Docker Security Operations
[INFO] 6.1  - Avoid image sprawl
[INFO]      * There are currently: 8 images
[INFO] 6.2  - Avoid container sprawl
[INFO]      * There are currently a total of 9 containers, with 1 of them currently running


[INFO] 7 - Docker Swarm Configuration
[PASS] 7.1  - Ensure swarm mode is not Enabled, if not needed
[PASS] 7.2  - Ensure the minimum number of manager nodes have been created in a swarm (Swarm mode not enabled)
[PASS] 7.3  - Ensure swarm services are binded to a specific host interface (Swarm mode not enabled)
[PASS] 7.4  - Ensure data exchanged between containers are encrypted on different nodes on the overlay network
[PASS] 7.5  - Ensure Docker's secret management commands are used for managing secrets in a Swarm cluster (Swarm mode not enabled)
[PASS] 7.6  - Ensure swarm manager is run in auto-lock mode (Swarm mode not enabled)
[PASS] 7.7  - Ensure swarm manager auto-lock key is rotated periodically (Swarm mode not enabled)
[PASS] 7.8  - Ensure node certificates are rotated as appropriate (Swarm mode not enabled)
[PASS] 7.9  - Ensure CA certificates are rotated as appropriate (Swarm mode not enabled)
[PASS] 7.10  - Ensure management plane traffic has been separated from data plane traffic (Swarm mode not enabled)

[INFO] Checks: 74
[INFO] Score: 12
root@seb-VirtualBox:~# 
```