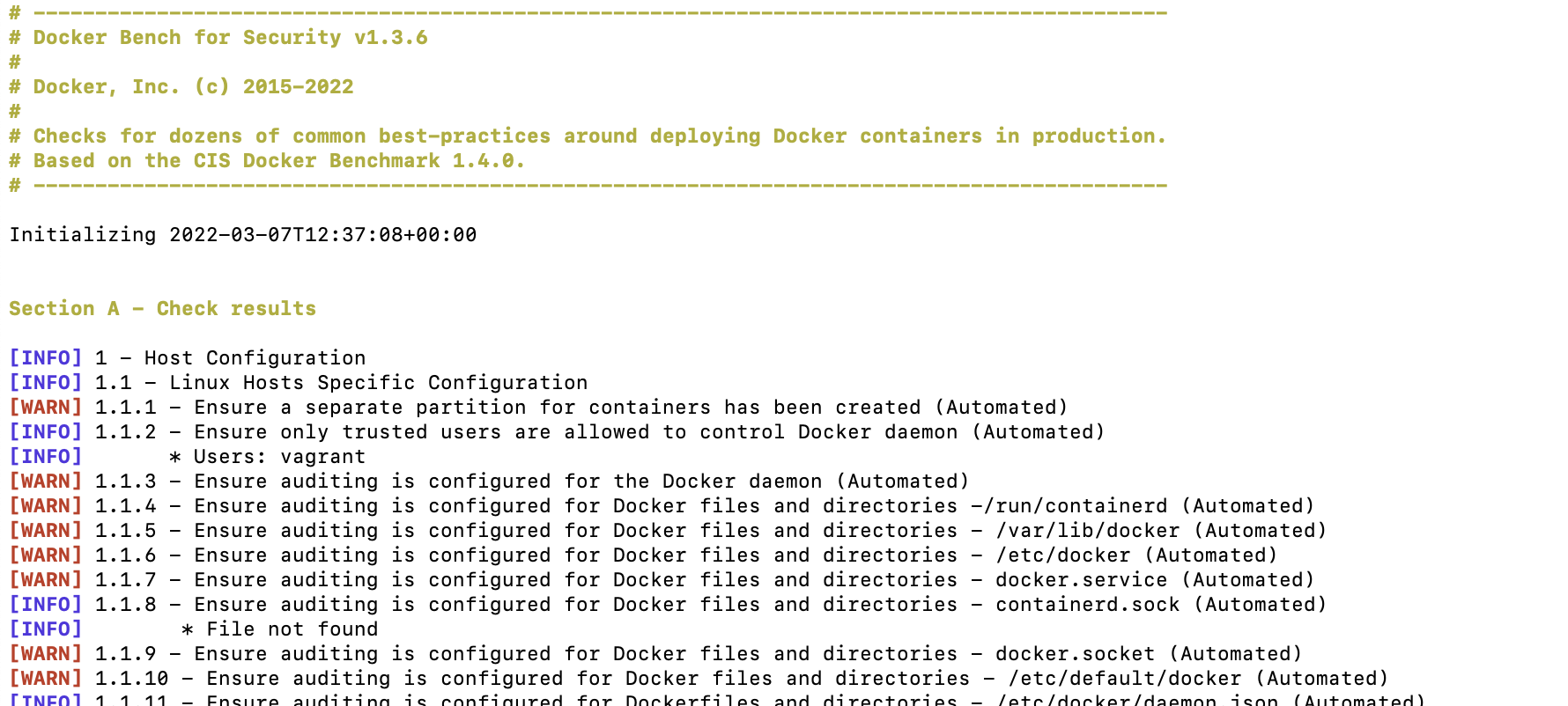
The Docker Bench for Security is a script that checks for dozens of common best-practices around deploying Docker containers in production.

Link: <https://github.com/docker/docker-bench-security>

# Docker Bench for Security

[](https://github.com/docker/docker-bench-security/blob/master/img/benchmark_log.png)

The Docker Bench for Security is a script that checks for dozens of common best-practices around deploying Docker containers in production. The tests are all automated, and are based on the [CIS Docker Benchmark v1.4.0](https://www.cisecurity.org/benchmark/docker/).

We are making this available as an open-source utility so the Docker community can have an easy way to self-assess their hosts and docker containers against this benchmark.

## Running Docker Bench for Security > Run from your base host

You can simply run this script from your base host by running:

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

cd docker-bench-security

sudo sh docker-bench-security.sh

### Run with Docker

Please note that the *docker/docker-bench-security* image is out-of-date and and a manual build is required. See [*#405*](https://github.com/docker/docker-bench-security/issues/405) for more information.

We packaged docker bench as a small container for your convenience. Note that this container is being run with a lot of privilege -- sharing the host's filesystem, pid and network namespaces, due to portions of the benchmark applying to the running host.

The easiest way to run your hosts against the Docker Bench for Security is by running our pre-built container:

docker run --rm --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

Don't forget to adjust the shared volumes according to your operating system. Some examples are:

1. On Ubuntu the docker.service and docker.secret files are located in /lib/systemd/system folder by default.

docker run --rm --net host --pid host --userns host --cap-add audit\_control \

-e DOCKER\_CONTENT\_TRUST=$DOCKER\_CONTENT\_TRUST \

-v /etc:/etc:ro \

-v /lib/systemd/system:/lib/systemd/system: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

1. The /etc/hostname file is missing on macOS, so it will need to be created first. Also, Docker Desktop on macOS doesn't have /usr/lib/systemd or the above Docker binaries.

sudo touch /etc/hostname

docker run --rm --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

### Note

Docker bench requires Docker 1.13.0 or later in order to run.

Note that when distributions don't contain auditctl, the audit tests will check /etc/audit/audit.rules to see if a rule is present instead.

### Docker Bench for Security options

-b optional Do not print colors

-h optional Print this help message

-l FILE optional Log output in FILE, inside container if run using docker

-u USERS optional Comma delimited list of trusted docker user(s)

-c CHECK optional Comma delimited list of specific check(s) id

-e CHECK optional Comma delimited list of specific check(s) id to exclude

-i INCLUDE optional Comma delimited list of patterns within a container or image name to check

-x EXCLUDE optional Comma delimited list of patterns within a container or image name to exclude from check

-n LIMIT optional In JSON output, when reporting lists of items (containers, images, etc.), limit the number of reported items to LIMIT. Default 0 (no limit).

-p PRINT optional Disable the printing of remediation measures. Default: print remediation measures.

By default the Docker Bench for Security script will run all available CIS tests and produce logs in the log folder from current directory, named docker-bench-security.log.json and docker-bench-security.log.

If the docker container is used then the log files will be created inside the container in location /usr/local/bin/log/. If you wish to access them from the host after the container has been run you will need to mount a volume for storing them in.

The CIS based checks are named check\_<section>\_<number>, e.g. check\_2\_6 and community contributed checks are named check\_c\_<number>.

sh docker-bench-security.sh -c check\_2\_2 will only run check 2.2 Ensure the logging level is set to 'info'.

sh docker-bench-security.sh -e check\_2\_2 will run all available checks except 2.2 Ensure the logging level is set to 'info'.

sh docker-bench-security.sh -e docker\_enterprise\_configuration will run all available checks except the docker\_enterprise\_configuration group

sh docker-bench-security.sh -e docker\_enterprise\_configuration,check\_2\_2 will run allavailable checks except the docker\_enterprise\_configuration group and 2.2 Ensure the logging level is set to 'info'

sh docker-bench-security.sh -c container\_images,container\_runtime will run just the container\_images and container\_runtime checks

sh docker-bench-security.sh -c container\_images -e check\_4\_5 will run just the container\_images checks except 4.5 Ensure Content trust for Docker is Enabled

Note that when submitting checks, provide information why it is a reasonable test to add and please include some kind of official documentation verifying that information.

## Building Docker image

You have two options if you wish to build and run this container yourself:

1. Use Docker Build:

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

cd docker-bench-security

docker build --no-cache -t docker-bench-security .

Followed by an appropriate docker run command as stated above.

1. Use Docker Compose:

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

cd docker-bench-security

docker-compose run --rm docker-bench-security