DOCKER SECURITY FOR DEV

12.03.2019

Agenda

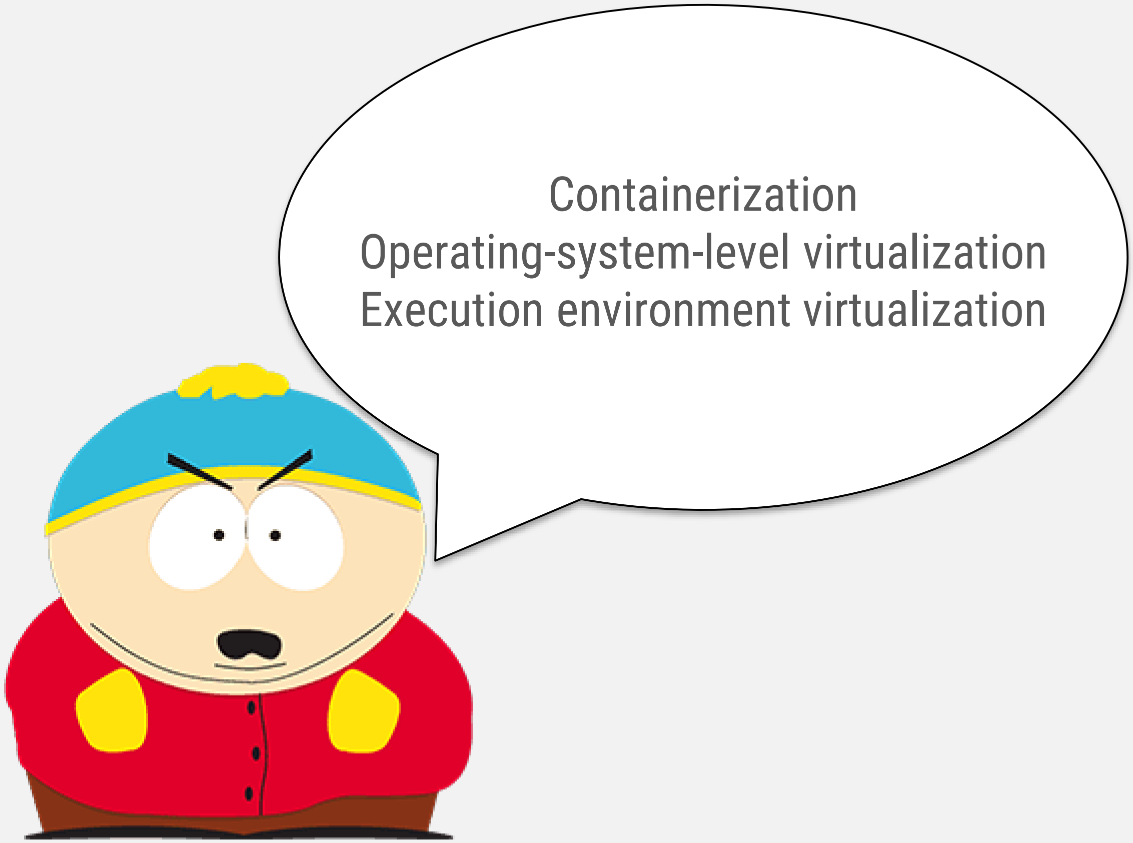
- Docker world
- Containers VS Virtual machine
- Security concerns
- Conclusion

Whoami

- ➔ M.Sc Computer Security
- ➔ M.Sc Software Development
- ➔ Worked previously as an embedded software developer
- ➔ Actually working at ImmunIT
  - ➔ Pentesting
  - ➔ Secure coding training
  - ➔ Security awareness training
  - ➔ Social Engineering
  - ➔ Project Management
  - ➔ R&D developer

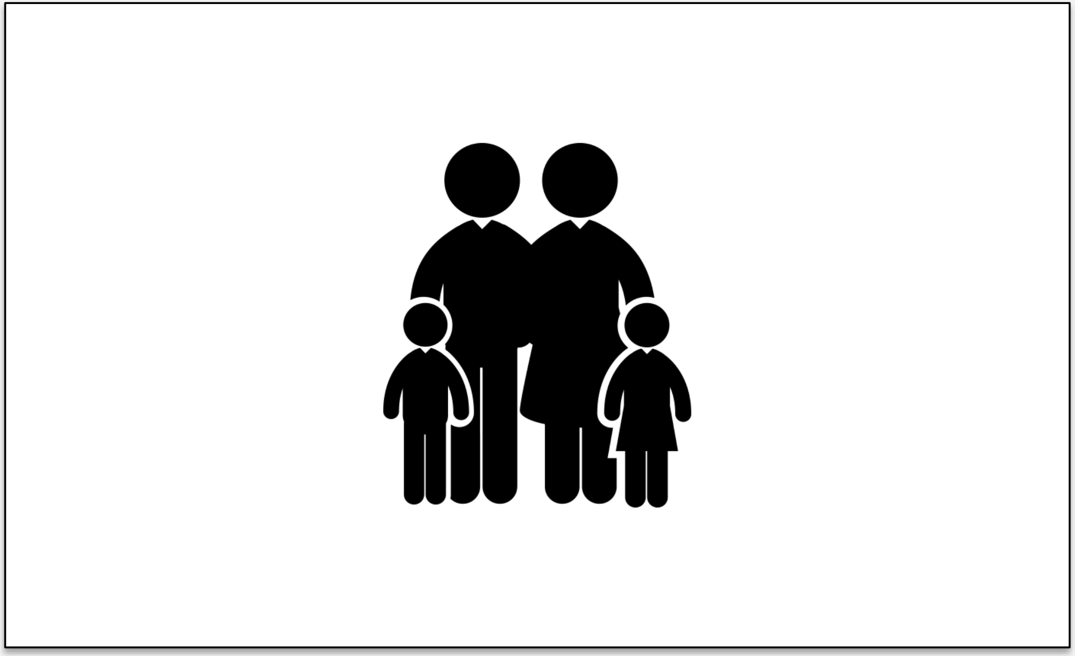
DOCKER WORLD

What is docker?

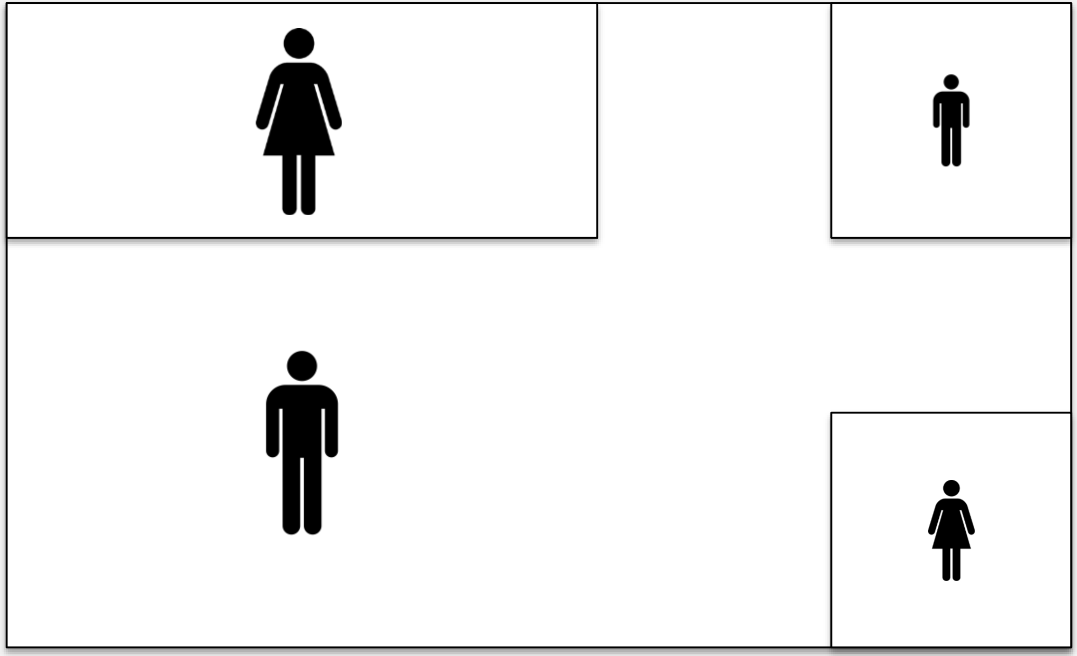


Containerization
Operating-system-level virtualization
Execution environment virtualization

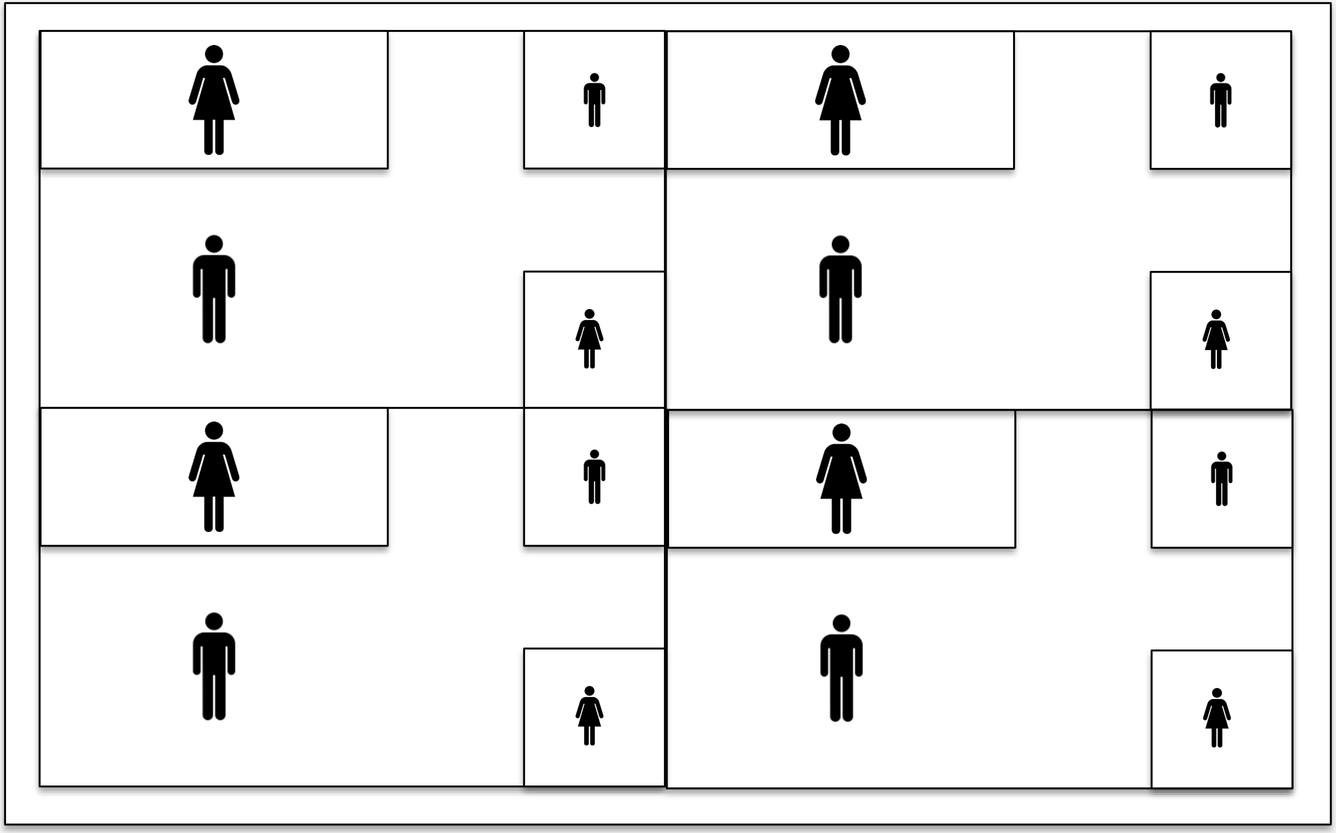
What is docker?



What is docker?



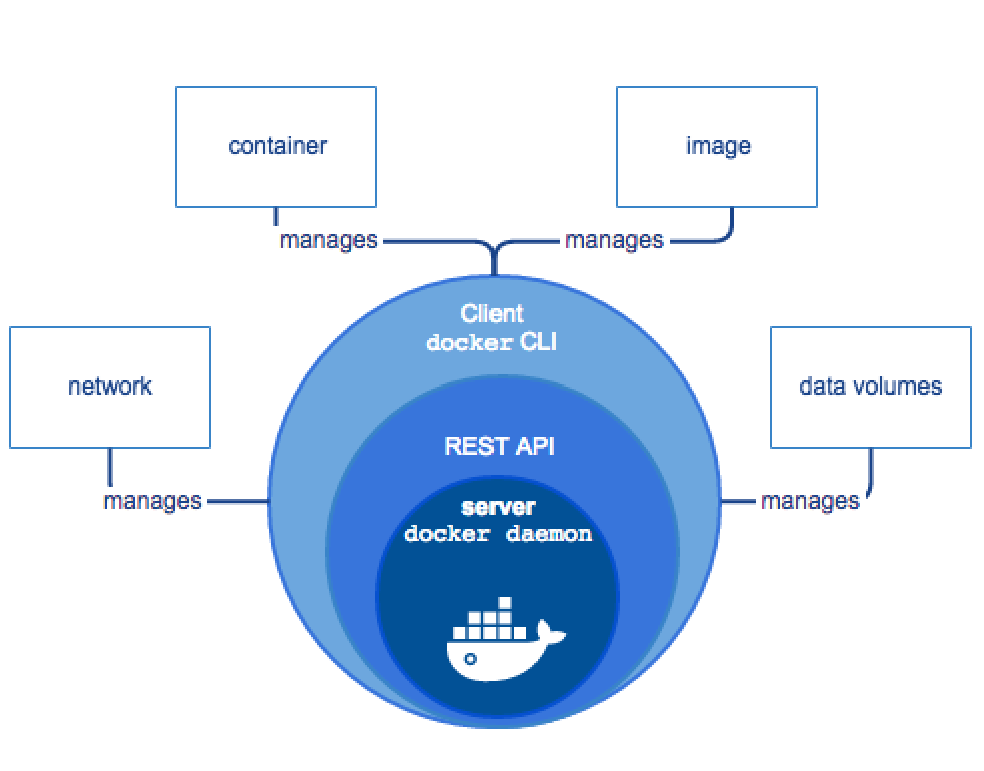
What is docker?



Why using docker?

- ➔ Isolate services
- ➔ Simplify micro-services enhancement and maintenance
- ➔ Avoid dependency issues
- ➔ Allow to execute untrusted code safely
- ➔ Reduce risks involved by a compromise
- ➔ etc

How it works?



container

image

manages

manages

Client
docker CLI

network

REST API

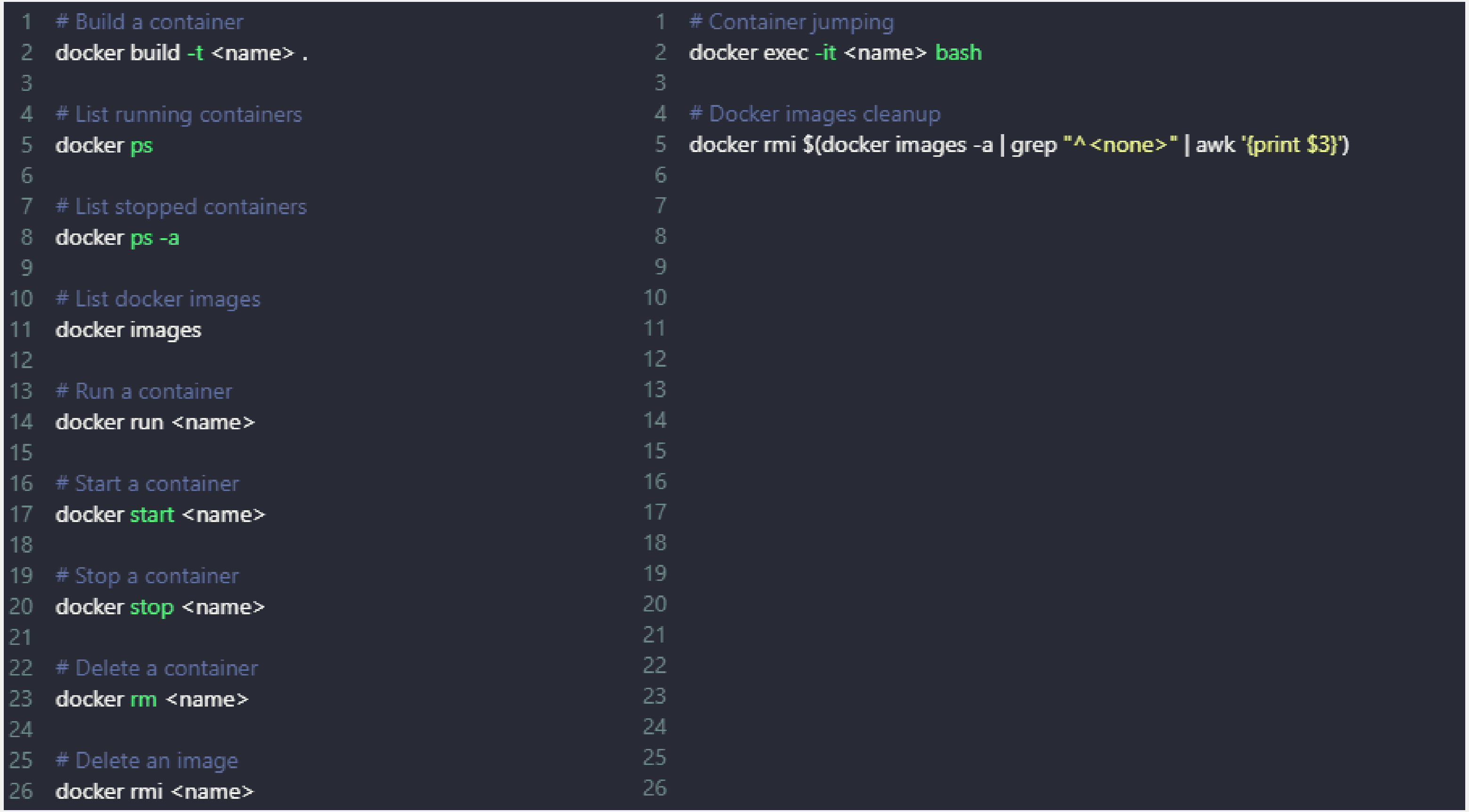
data volumes

manages

server
docker daemon

manages

Docker basics



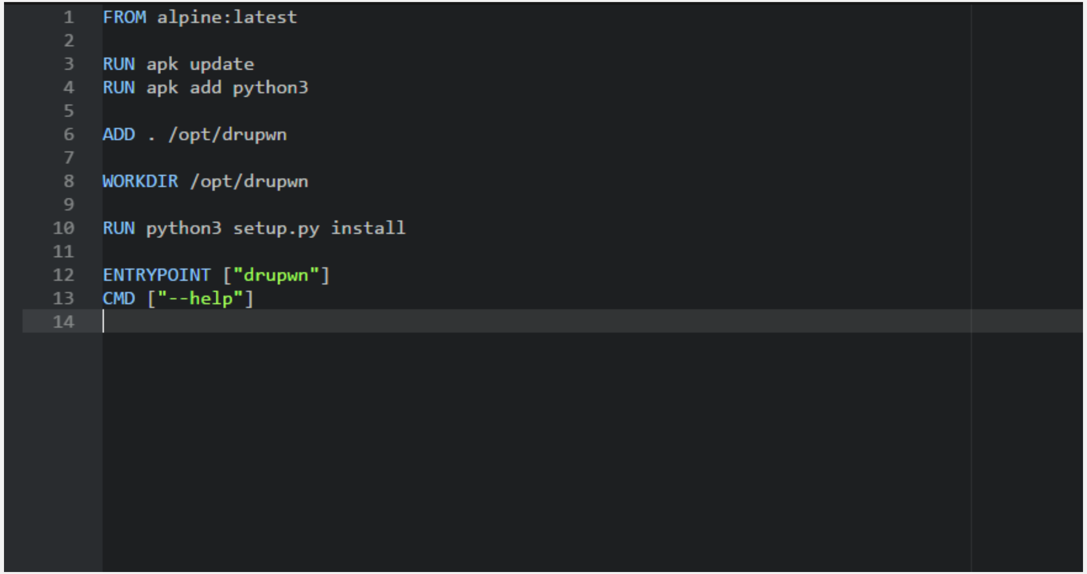
```
# Build a container
docker build -t <name> .

# List running containers
docker ps

# List stopped containers
docker ps -a

# List docker images
docker images

# Run a container
docker run <name>

# Start a container
docker start <name>

# Stop a container
docker stop <name>

# Delete a container
docker rm <name>

# Delete an image
docker rmi <name>
```

```
# Container jumping
docker exec -it <name> bash

# Docker images cleanup
docker rmi $(docker images -a | grep "^<none>" | awk '{print $3}')
```

Dockerfile & docker-compose

Dockerfile

➔ Defines a docker image

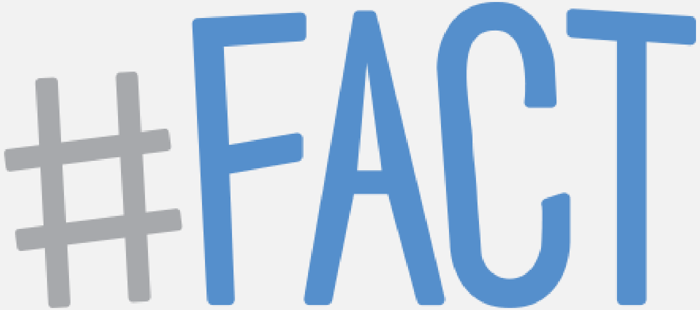


```
FROM alpine:latest

RUN apk update
RUN apk add python3

ADD . /opt/drupwn

WORKDIR /opt/drupwn

RUN python3 setup.py install

ENTRYPOINT ["drupwn"]
CMD ["--help"]

```

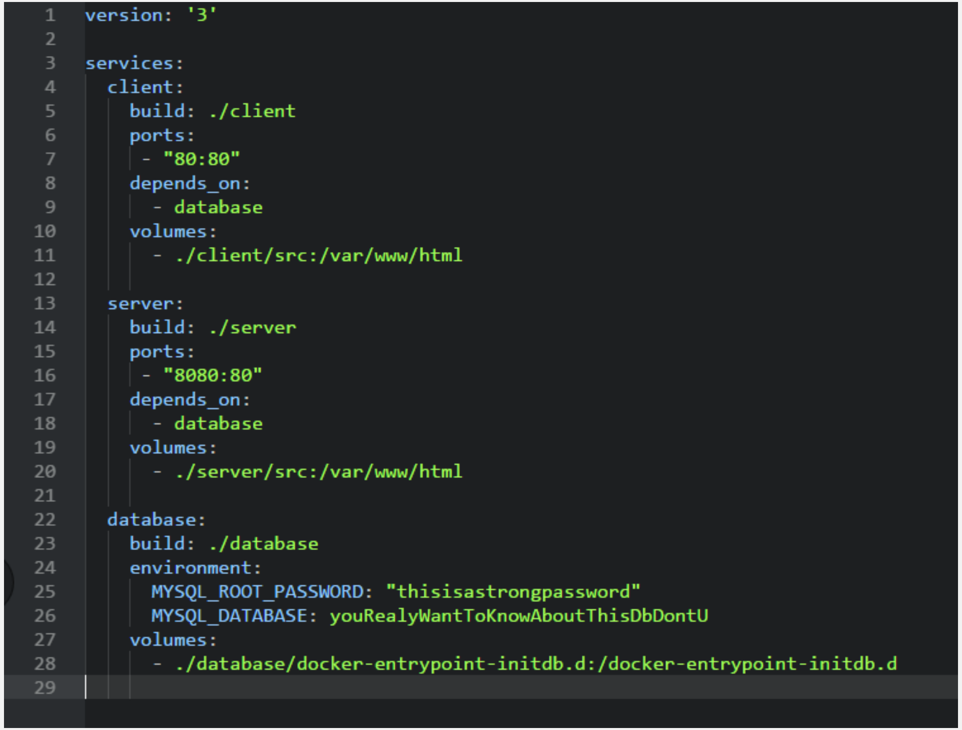
Dockerfile & docker-compose

Docker-compose

➔ Defines a containers stack



➔ Overwrite Dockerfile behaviors



```yaml
version: '3'

services:
  client:
    build: ./client
    ports:
     - "80:80"
    depends_on:
      - database
    volumes:
      - ./client/src:/var/www/html

  server:
    build: ./server
    ports:
     - "8080:80"
    depends_on:
      - database
    volumes:
      - ./server/src:/var/www/html

  database:
    build: ./database
    environment:
      MYSQL_ROOT_PASSWORD: "thisisastrongpassword"
      MYSQL_DATABASE: youRealyWantToKnowAboutThisDbDontU
    volumes:
      - ./database/docker-entrypoint-initdb.d:/docker-entrypoint-initdb.d

```

Orchestration

→ Automates image buildings

→ Automates deployment

→ Resilient

→ Macro management

→ Live metrics

Orchestration

Orchestrators

kubernetes

MESOSPHERE

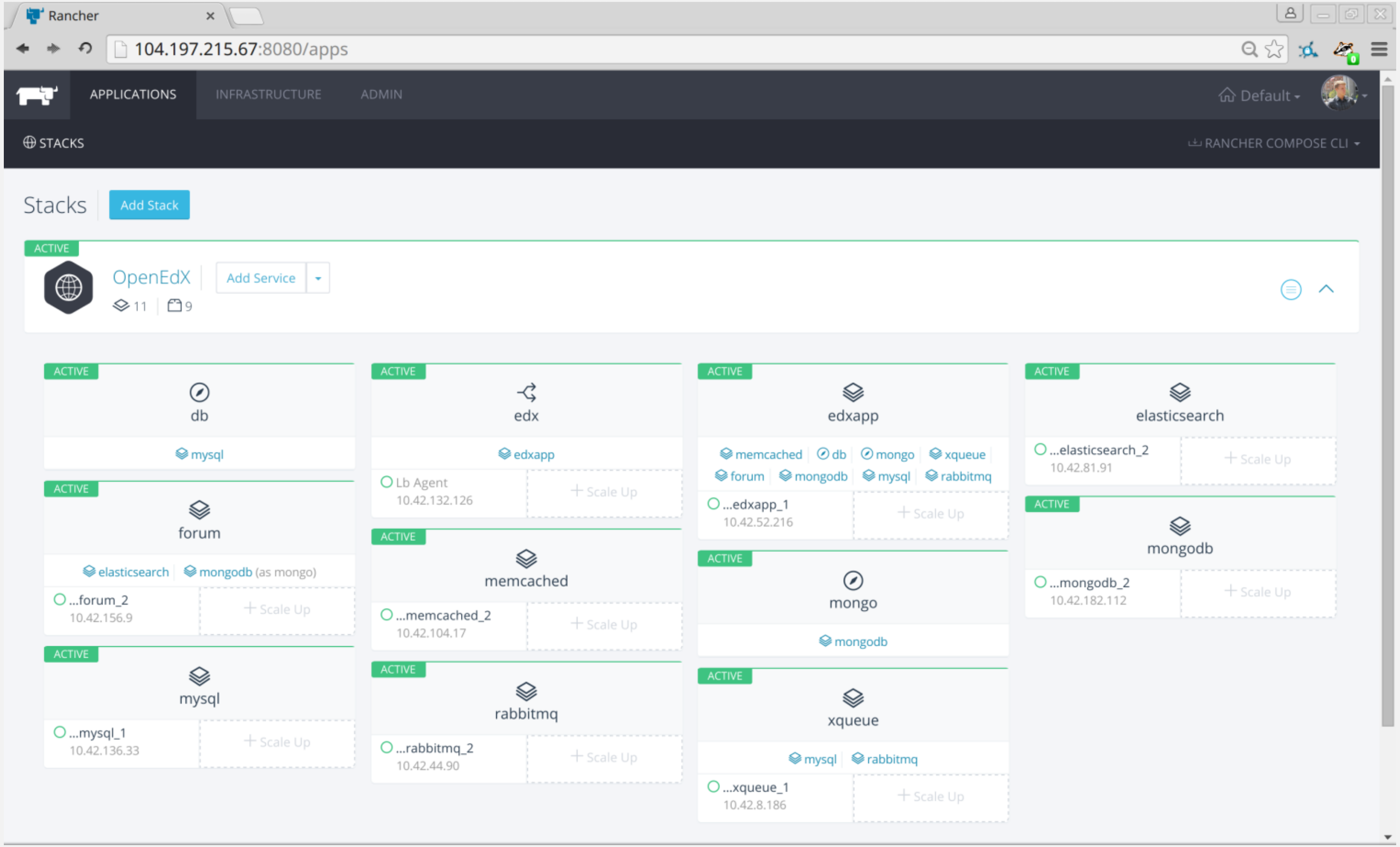
docker
SWARM

Orchestration

Registry & rancher

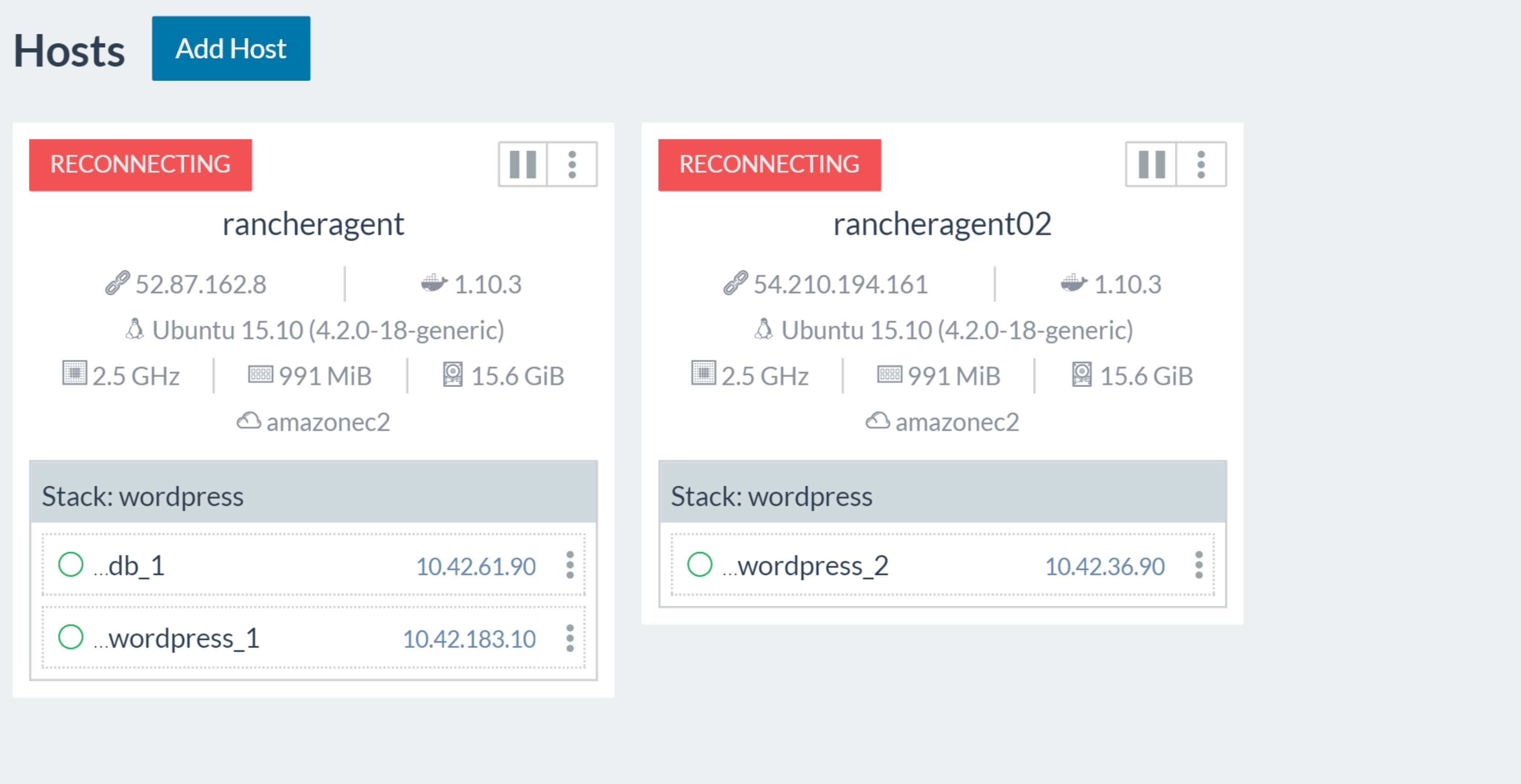
Orchestration

Rancher overview



Orchestration

Rancher overview



RECONNECTING

rancheragent

52.87.162.8 | 1.10.3

Ubuntu 15.10 (4.2.0-18-generic)

2.5 GHz | 991 MiB | 15.6 GiB

amazonec2

Stack: wordpress

○ ...db_1 10.42.61.90

○ ...wordpress_1 10.42.183.10

RECONNECTING

rancheragent02

54.210.194.161 | 1.10.3

Ubuntu 15.10 (4.2.0-18-generic)

2.5 GHz | 991 MiB | 15.6 GiB

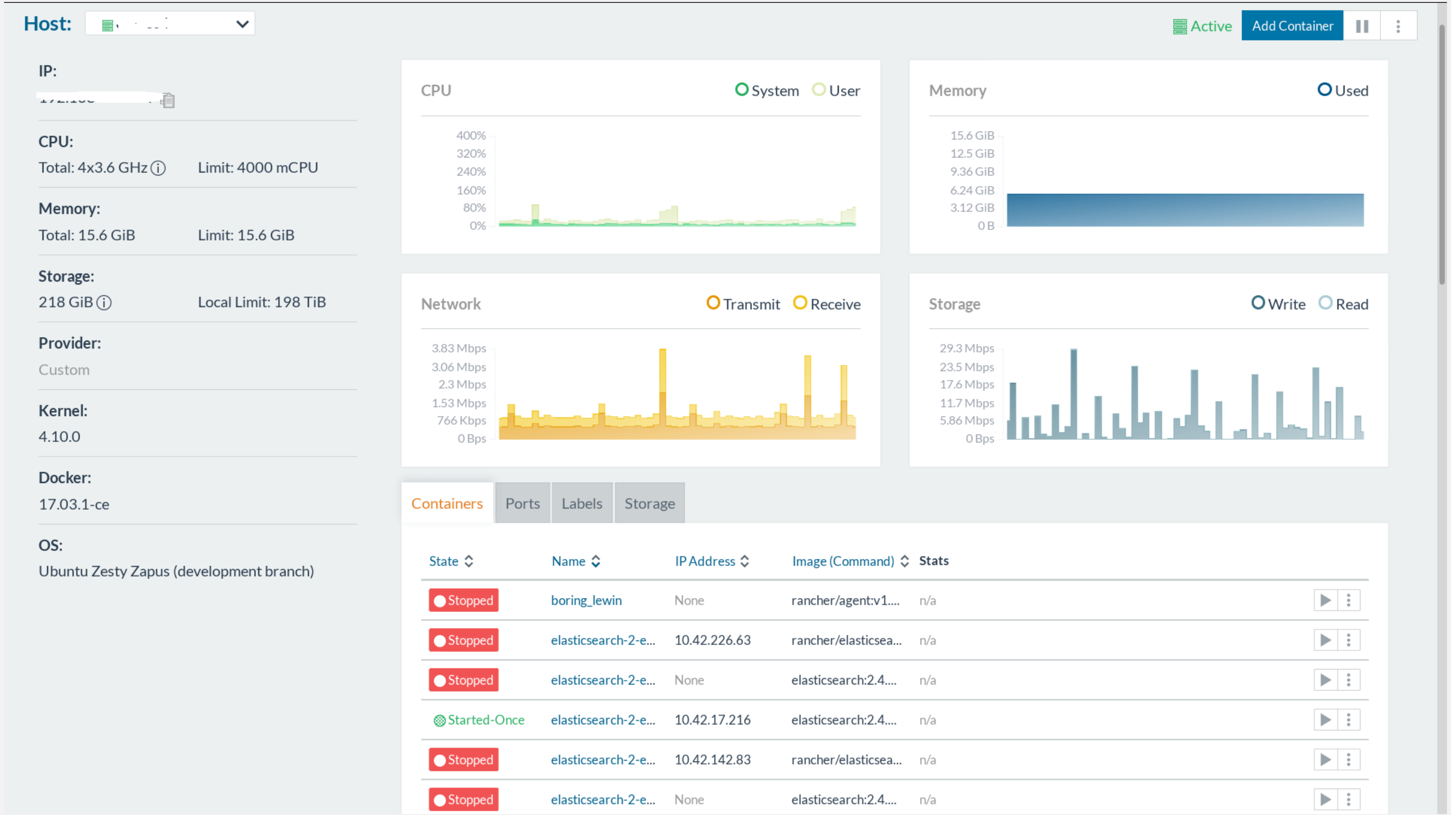
amazonec2

Stack: wordpress

○ ...wordpress_2 10.42.36.90

Orchestration

Rancher overview



IP:
[redacted]

CPU:
Total: 4x3.6 GHz ⓘ Limit: 4000 mCPU

Memory:
Total: 15.6 GiB Limit: 15.6 GiB

Storage:
218 GiB ⓘ Local Limit: 198 TiB

Provider:
Custom

Kernel:
4.10.0

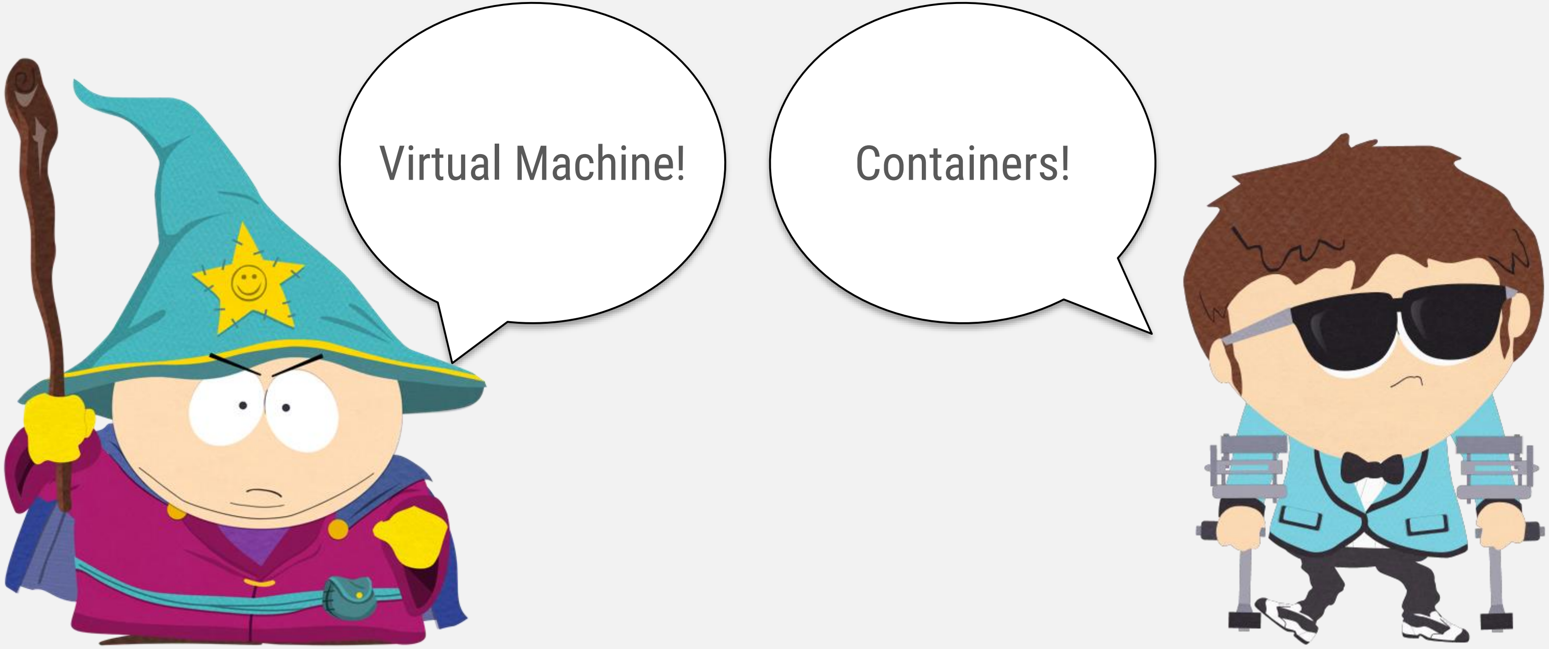
Docker:
17.03.1-ce

OS:
Ubuntu Zesty Zapus (development branch)

State	Name	IP Address	Image (Command)	Stats
Stopped	boring_lewin	None	rancher/agent:v1....	n/a
Stopped	elasticsearch-2-e...	10.42.226.63	rancher/elasticsea...	n/a
Stopped	elasticsearch-2-e...	None	elasticsearch:2.4....	n/a
Started-Once	elasticsearch-2-e...	10.42.17.216	elasticsearch:2.4....	n/a
Stopped	elasticsearch-2-e...	10.42.142.83	rancher/elasticsea...	n/a
Stopped	elasticsearch-2-e...	None	elasticsearch:2.4....	n/a

Containers VS Virtual Machine The millennial war

Containers VS Virtual Machine



Containers VS Virtual Machine

Containers VS Virtual Machine

Security concerns

Kernel namespace

- Containers process are running in their own kernel namespace
  - Provides segregation
  - Decreases risk exposure
- Containers get their own network stack

UPDATE

User namespace

➔ Best way to prevent privilege escalation attack

➔ Configured on the host level

➔ Prevent root usage

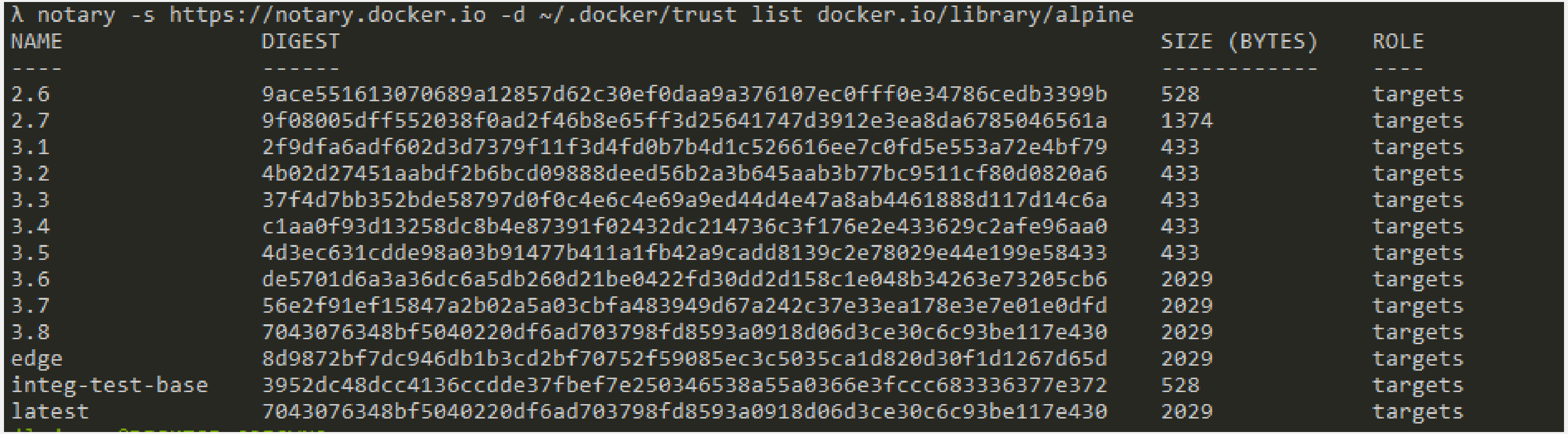
Tools

Docker notary

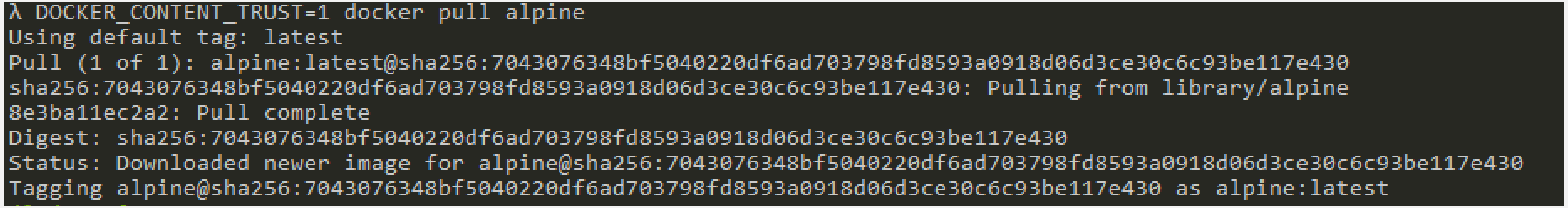
- ➔ Verify image signature
- ➔ Ensure integrity
- ➔ Avoid backdoors
- ➔ Cross platform

Tools

Docker notary



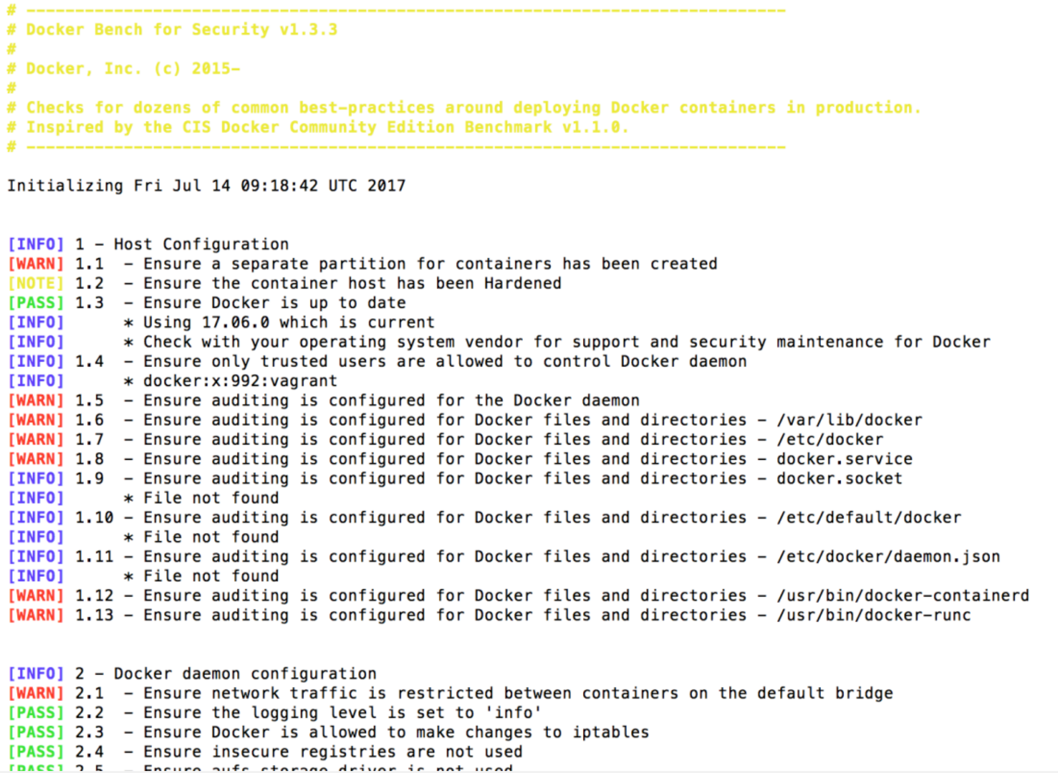
```
λ notary -s https://notary.docker.io -d ~/.docker/trust list docker.io/library/alpine
NAME               DIGEST                                                              SIZE (BYTES)    ROLE
----               ------                                                              ------------    ----
2.6                9ace551613070689a12857d62c30ef0daa9a376107ec0fff0e34786cedb3399b    528             targets
2.7                9f08005dff552038f0ad2f46b8e65ff3d25641747d3912e3ea8da6785046561a    1374            targets
3.1                2f9dfa6adf602d3d7379f11f3d4fd0b7b4d1c526616ee7c0fd5e553a72e4bf79    433             targets
3.2                4b02d27451aabdf2b6bcd09888deed56b2a3b645aab3b77bc9511cf80d0820a6    433             targets
3.3                37f4d7bb352bde58797d0f0c4e6c4e69a9ed44d4e47a8ab4461888d117d14c6a    433             targets
3.4                c1aa0f93d13258dc8b4e87391f02432dc214736c3f176e2e433629c2afe96aa0    433             targets
3.5                4d3ec631cdde98a03b91477b411a1fb42a9cadd8139c2e78029e44e199e58433    433             targets
3.6                de5701d6a3a36dc6a5db260d21be0422fd30dd2d158c1e048b34263e73205cb6    2029            targets
3.7                56e2f91ef15847a2b02a5a03cbfa483949d67a242c37e33ea178e3e7e01e0dfd    2029            targets
3.8                7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430    2029            targets
edge               8d9872bf7dc946db1b3cd2bf70752f59085ec3c5035ca1d820d30f1d1267d65d    2029            targets
integ-test-base    3952dc48dcc4136ccdde37fbef7e250346538a55a0366e3fccc683336377e372    528             targets
latest             7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430    2029            targets
```



```
λ DOCKER_CONTENT_TRUST=1 docker pull alpine
Using default tag: latest
Pull (1 of 1): alpine:latest@sha256:7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430
sha256:7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430: Pulling from library/alpine
8e3ba11ec2a2: Pull complete
Digest: sha256:7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430
Status: Downloaded newer image for alpine@sha256:7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430
Tagging alpine@sha256:7043076348bf5040220df6ad703798fd8593a0918d06d3ce30c6c93be117e430 as alpine:latest
```

Tools

Docker bench security



```
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.3
#
# 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 14 09:18:42 UTC 2017


[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
[PASS] 1.3  - Ensure Docker is up to date
[INFO]      * Using 17.06.0 which is current
[INFO]      * Check with your operating system vendor for support and security maintenance for Docker
[INFO] 1.4  - Ensure only trusted users are allowed to control Docker daemon
[INFO]      * docker:x:992:vagrant
[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
[INFO] 1.9  - Ensure auditing is configured for Docker files and directories - docker.socket
[INFO]      * File not found
[INFO] 1.10 - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO]      * File not found
[INFO] 1.11 - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[INFO]      * File not found
[WARN] 1.12 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-containerd
[WARN] 1.13 - Ensure auditing is configured for Docker files and directories - /usr/bin/docker-runc


[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
```

Tools

Traefik

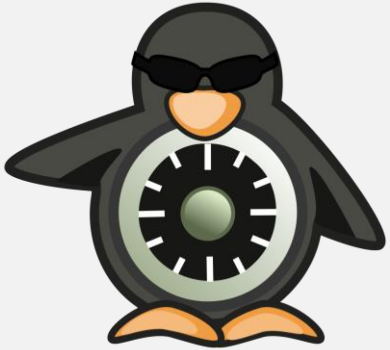
Tools

CoreOS

Tools

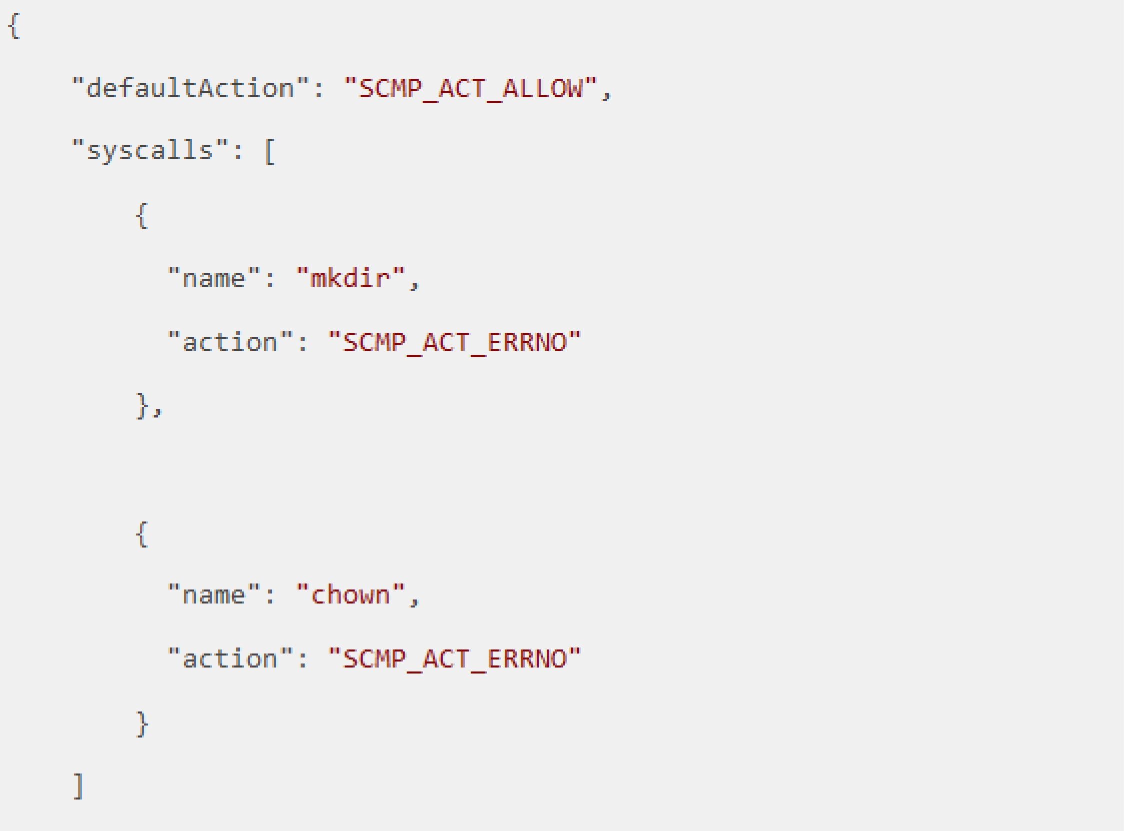
Dockscan

Container hardening & Access Control Management



Container hardening & Access Control Management

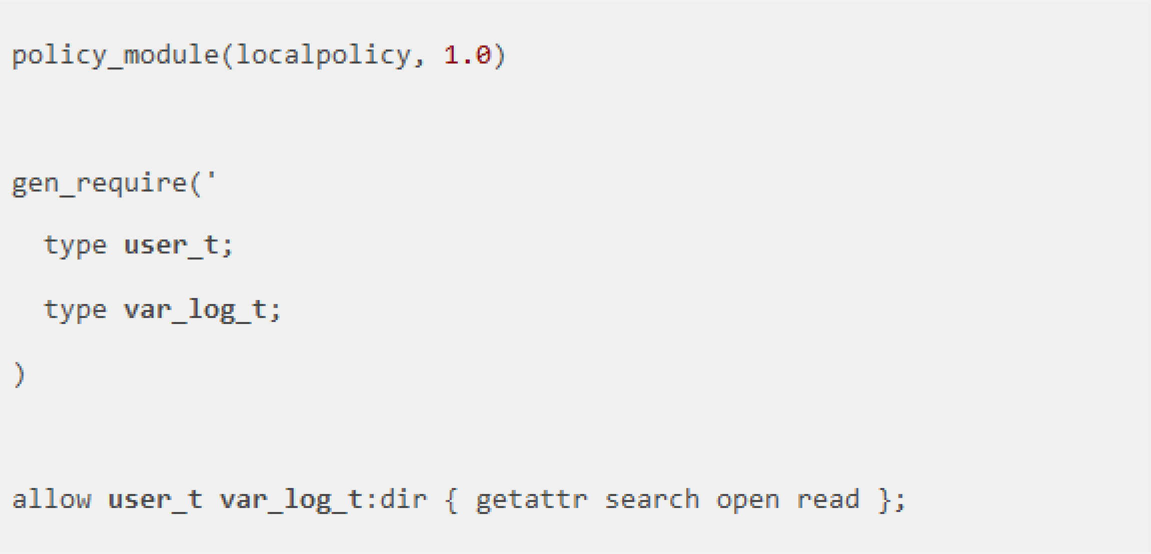
Seccomp



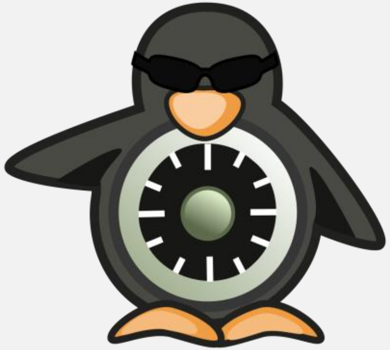
```
{
    "defaultAction": "SCMP_ACT_ALLOW",
    "syscalls": [
        {
          "name": "mkdir",
          "action": "SCMP_ACT_ERRNO"
        },

        {
          "name": "chown",
          "action": "SCMP_ACT_ERRNO"
        }
    ]
```

Container hardening & Access Control Management

SE Linux

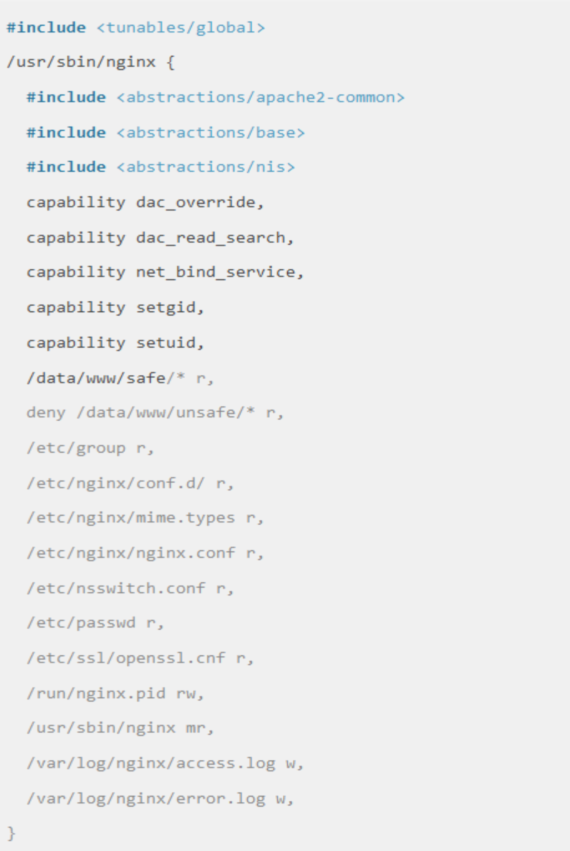


```
policy_module(localpolicy, 1.0)


gen_require('
  type user_t;
  type var_log_t;
')


allow user_t var_log_t:dir { getattr search open read };
```



Container hardening & Access Control Management

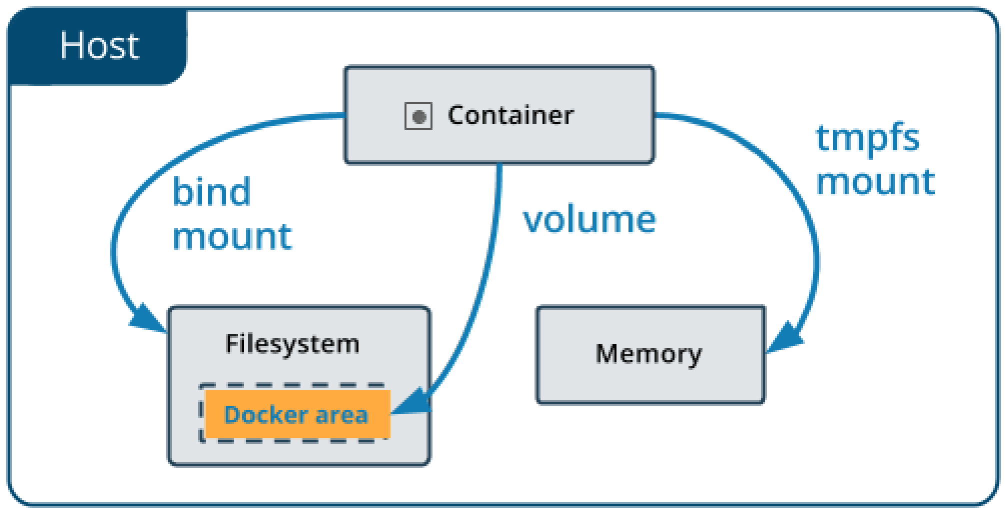
App Armor



```
#include <tunables/global>
/usr/sbin/nginx {
  #include <abstractions/apache2-common>
  #include <abstractions/base>
  #include <abstractions/nis>
  capability dac_override,
  capability dac_read_search,
  capability net_bind_service,
  capability setgid,
  capability setuid,
  /data/www/safe/* r,
  deny /data/www/unsafe/* r,
  /etc/group r,
  /etc/nginx/conf.d/ r,
  /etc/nginx/mime.types r,
  /etc/nginx/nginx.conf r,
  /etc/nsswitch.conf r,
  /etc/passwd r,
  /etc/ssl/openssl.cnf r,
  /run/nginx.pid rw,
  /usr/sbin/nginx mr,
  /var/log/nginx/access.log w,
  /var/log/nginx/error.log w,
}
```

Flags

Volume vs mount vs tmpfs



Flags

Winner is volume

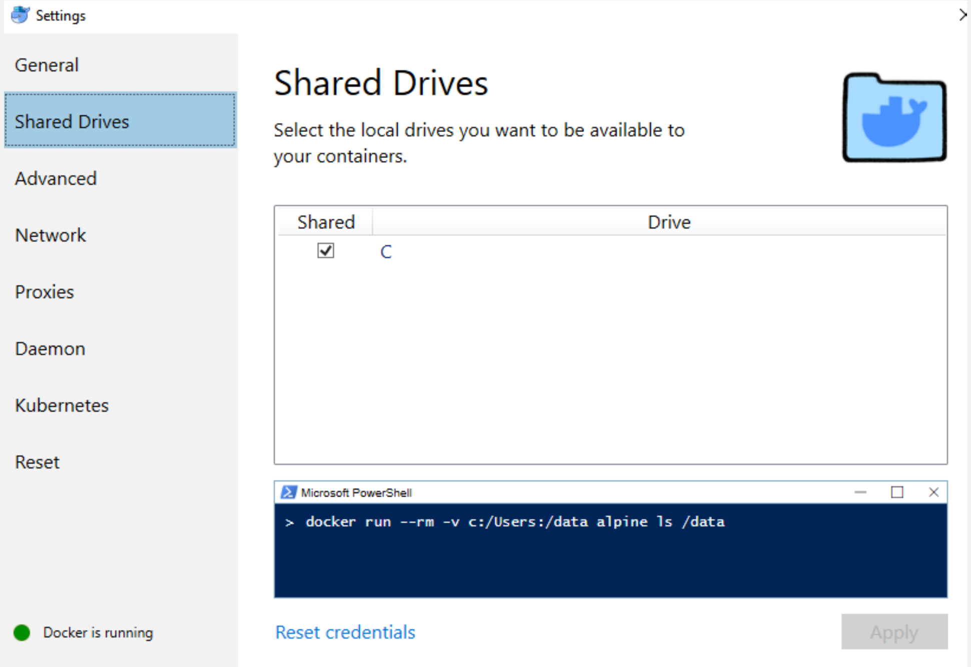
- ➔ Easier to back up
- ➔ Can be managed through the docker CLI
- ➔ Cross-platform
- ➔ Safe sharing
- ➔ Remote volume
- ➔ Data encryption (LVM, LUKS)

Flags

Winner is volume

→ Avoid mounting sensitive folder

→ Use ro flag when needed



Shared Drives

Select the local drives you want to be available to your containers.

Shared	Drive
☑	C

```
> docker run --rm -v c:/Users:/data alpine ls /data
```

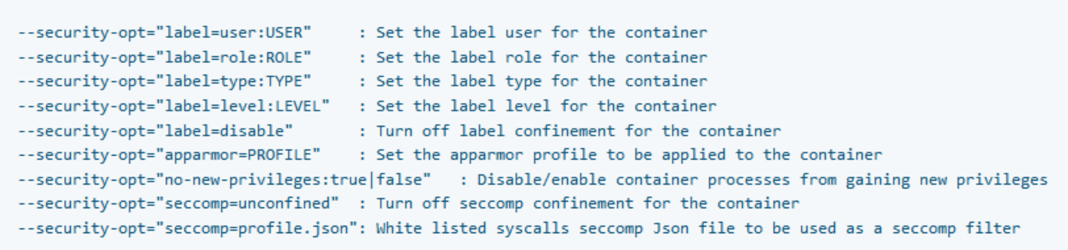
Flags

Privileged container

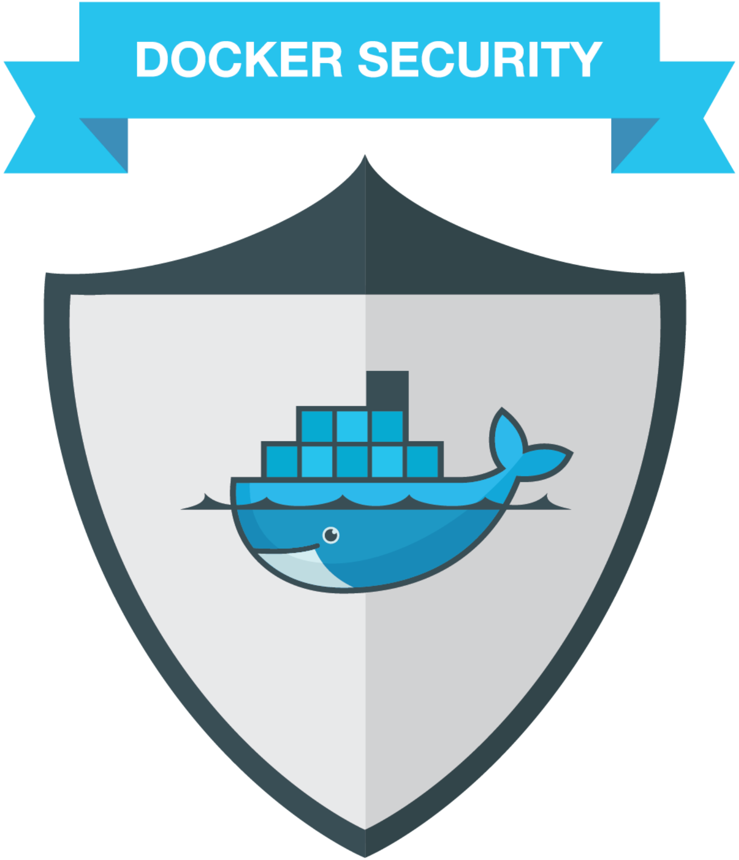
- ➔ Privileged container run as a proper OS
- ➔ Can modify interfaces / iptables
- ➔ Access host devices

Flags

Security opt

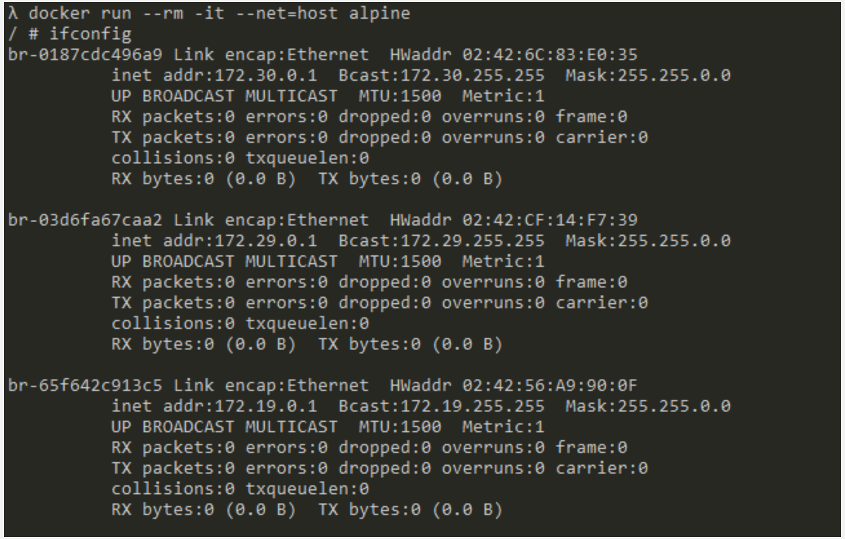


```
--security-opt="label=user:USER"     : Set the label user for the container
--security-opt="label=role:ROLE"     : Set the label role for the container
--security-opt="label=type:TYPE"     : Set the label type for the container
--security-opt="label=level:LEVEL"   : Set the label level for the container
--security-opt="label=disable"       : Turn off label confinement for the container
--security-opt="apparmor=PROFILE"    : Set the apparmor profile to be applied to the container
--security-opt="no-new-privileges:true|false"   : Disable/enable container processes from gaining new privileges
--security-opt="seccomp=unconfined"  : Turn off seccomp confinement for the container
--security-opt="seccomp=profile.json": White listed syscalls seccomp Json file to be used as a seccomp filter
```

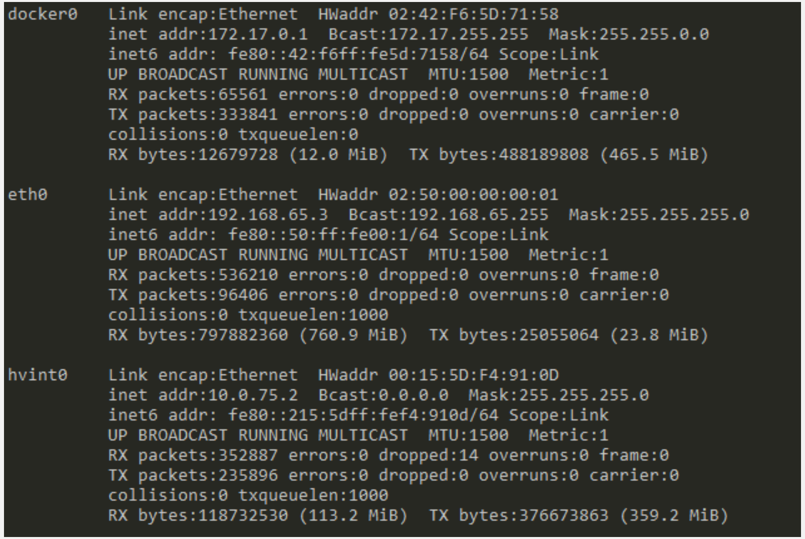


Flags

Network namespace



```
λ docker run --rm -it --net=host alpine
/ # ifconfig
br-0187cdc496a9 Link encap:Ethernet  HWaddr 02:42:6C:83:E0:35
          inet addr:172.30.0.1  Bcast:172.30.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

br-03d6fa67caa2 Link encap:Ethernet  HWaddr 02:42:CF:14:F7:39
          inet addr:172.29.0.1  Bcast:172.29.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

br-65f642c913c5 Link encap:Ethernet  HWaddr 02:42:56:A9:90:0F
          inet addr:172.19.0.1  Bcast:172.19.255.255  Mask:255.255.0.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```



```
docker0   Link encap:Ethernet  HWaddr 02:42:F6:5D:71:58
          inet addr:172.17.0.1  Bcast:172.17.255.255  Mask:255.255.0.0
          inet6 addr: fe80::42:f6ff:fe5d:7158/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:65561 errors:0 dropped:0 overruns:0 frame:0
          TX packets:333841 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:12679728 (12.0 MiB)  TX bytes:488189808 (465.5 MiB)

eth0      Link encap:Ethernet  HWaddr 02:50:00:00:00:01
          inet addr:192.168.65.3  Bcast:192.168.65.255  Mask:255.255.255.0
          inet6 addr: fe80::50:ff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:536210 errors:0 dropped:0 overruns:0 frame:0
          TX packets:96406 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:797882360 (760.9 MiB)  TX bytes:25055064 (23.8 MiB)

hvint0    Link encap:Ethernet  HWaddr 00:15:5D:F4:91:0D
          inet addr:10.0.75.2  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::215:5dff:fef4:910d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:352887 errors:0 dropped:14 overruns:0 frame:0
          TX packets:235896 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:118732530 (113.2 MiB)  TX bytes:376673863 (359.2 MiB)
```

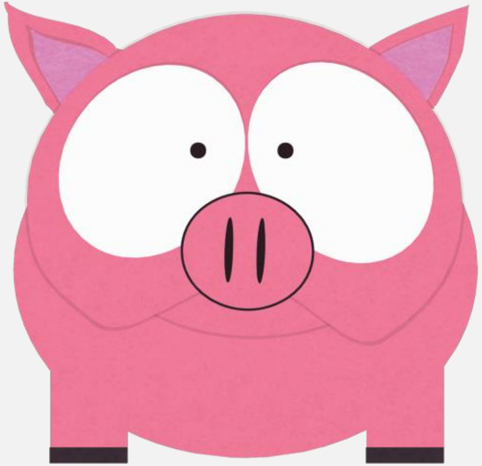
Flags

Network namespace

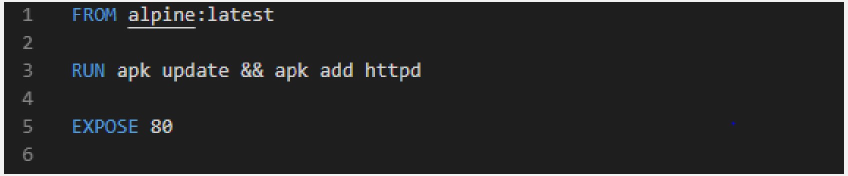
- ➔ Use dedicated networks
- ➔ Isolate containers on separated networks
- ➔ Create networks for exposed containers
- ➔ Segregate and segment networks as your own internal network

Ports exposure

- ➔ Control services exposure
- ➔ Do not expose unnecessary ports



Ports exposure

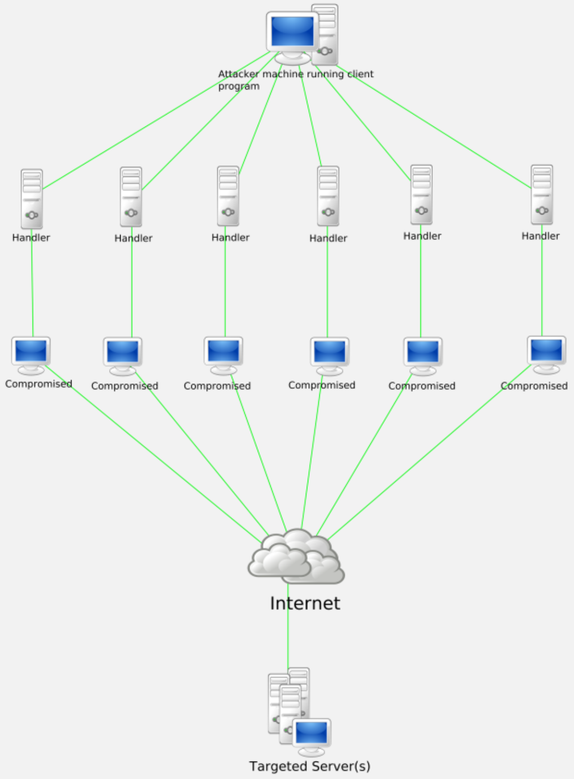


```
FROM alpine:latest

RUN apk update && apk add httpd

EXPOSE 80

```

EXPOSE keyword is overwritten by **–p** flag at runtime

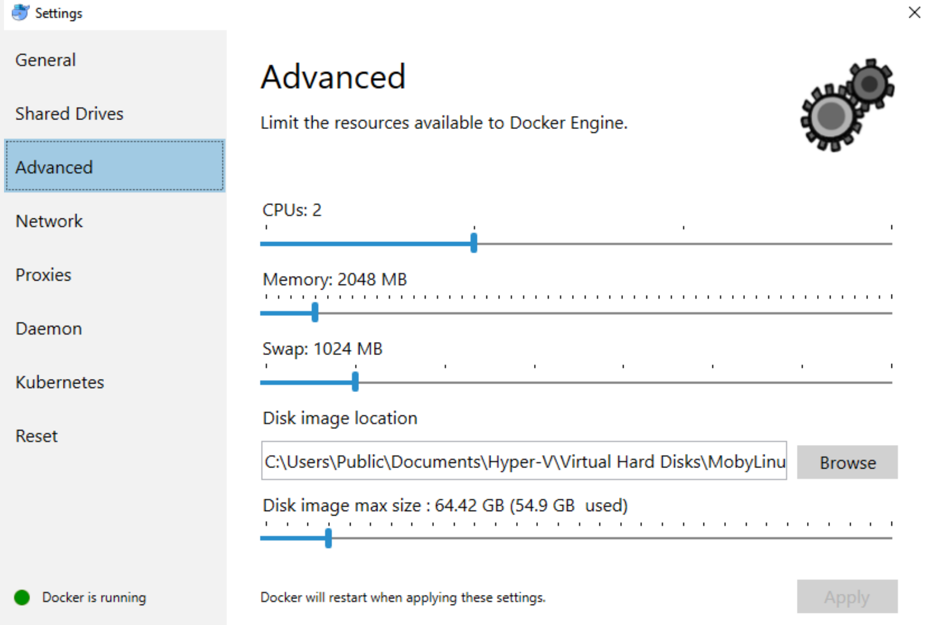
```
Docker run –rm –it –p 0.0.0.0:1337:80 alpine
Docker run –rm –it –p 127.0.0.1:1337:80 alpine
```

How to avoid Denial of Service attack



How to avoid Denial of Service attacks

By default, a container has no resource constraints and can use as much of a given resource as the hosts's kernel scheduler will allow



Advanced

Limit the resources available to Docker Engine.

CPUs: 2

Memory: 2048 MB

Swap: 1024 MB

Disk image location

C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks\MobyLinu

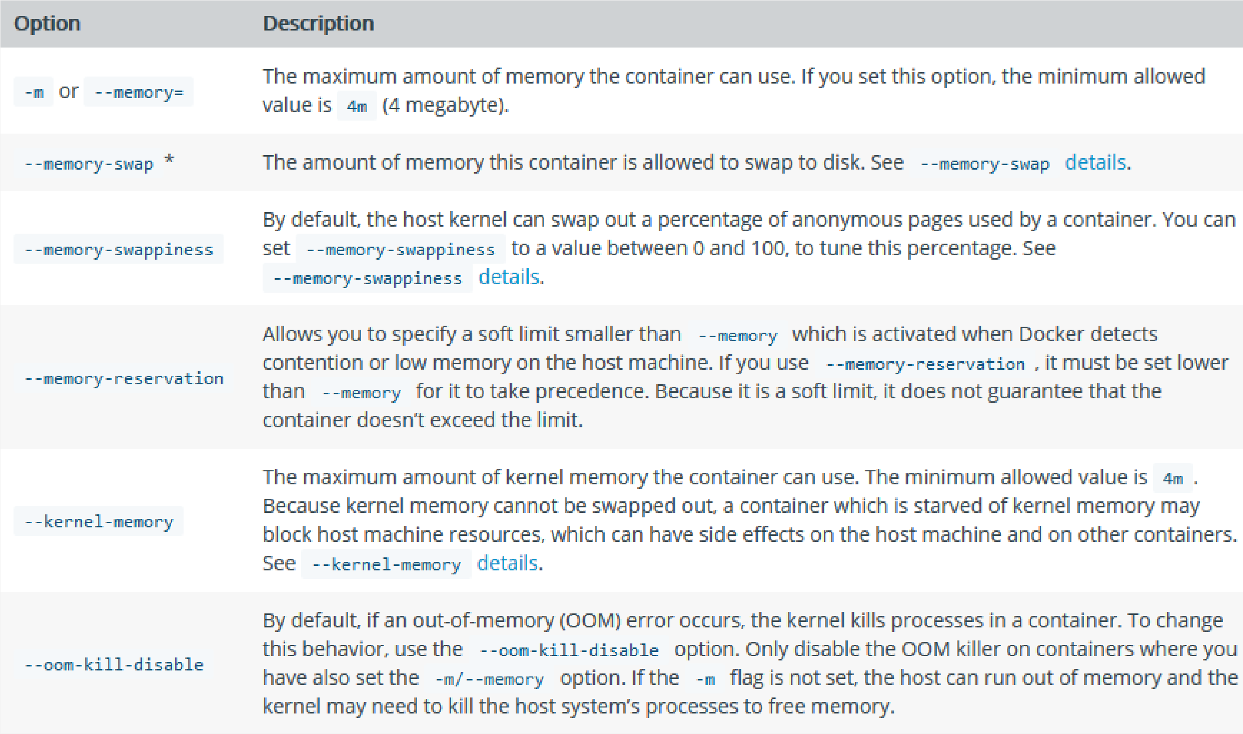
Disk image max size : 64.42 GB (54.9 GB used)

Docker is running

Docker will restart when applying these settings.

How to avoid Denial of Service attacks

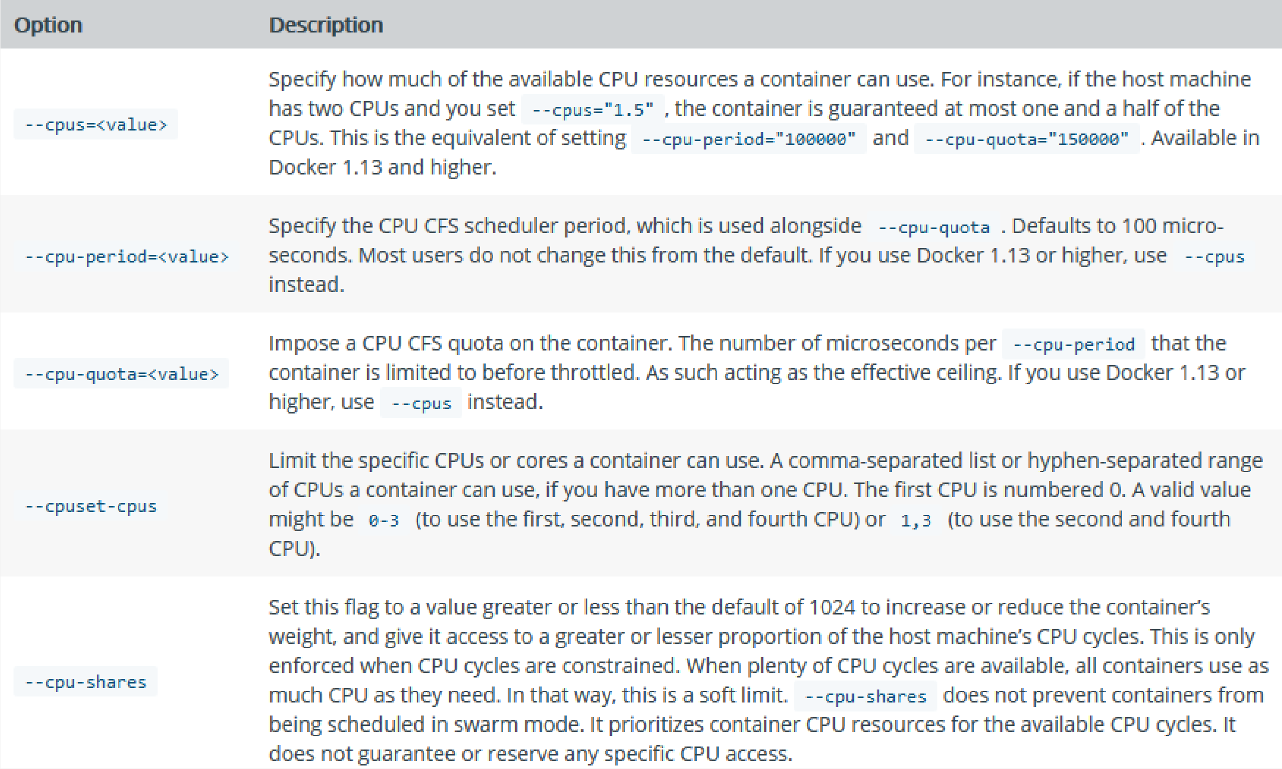
Memory usage



Option	Description
`-m` or `--memory=`	The maximum amount of memory the container can use. If you set this option, the minimum allowed value is `4m` (4 megabyte).
`--memory-swap` *	The amount of memory this container is allowed to swap to disk. See `--memory-swap` details.
`--memory-swappiness`	By default, the host kernel can swap out a percentage of anonymous pages used by a container. You can set `--memory-swappiness` to a value between 0 and 100, to tune this percentage. See `--memory-swappiness` details.
`--memory-reservation`	Allows you to specify a soft limit smaller than `--memory` which is activated when Docker detects contention or low memory on the host machine. If you use `--memory-reservation` , it must be set lower than `--memory` for it to take precedence. Because it is a soft limit, it does not guarantee that the container doesn't exceed the limit.
`--kernel-memory`	The maximum amount of kernel memory the container can use. The minimum allowed value is `4m` . Because kernel memory cannot be swapped out, a container which is starved of kernel memory may block host machine resources, which can have side effects on the host machine and on other containers. See `--kernel-memory` details.
`--oom-kill-disable`	By default, if an out-of-memory (OOM) error occurs, the kernel kills processes in a container. To change this behavior, use the `--oom-kill-disable` option. Only disable the OOM killer on containers where you have also set the `-m/--memory` option. If the `-m` flag is not set, the host can run out of memory and the kernel may need to kill the host system's processes to free memory.

How to avoid Denial of Service attacks

CPU usage



Option	Description
`--cpus=<value>`	Specify how much of the available CPU resources a container can use. For instance, if the host machine has two CPUs and you set `--cpus="1.5"` , the container is guaranteed at most one and a half of the CPUs. This is the equivalent of setting `--cpu-period="100000"` and `--cpu-quota="150000"` . Available in Docker 1.13 and higher.
`--cpu-period=<value>`	Specify the CPU CFS scheduler period, which is used alongside `--cpu-quota` . Defaults to 100 micro-seconds. Most users do not change this from the default. If you use Docker 1.13 or higher, use `--cpus` instead.
`--cpu-quota=<value>`	Impose a CPU CFS quota on the container. The number of microseconds per `--cpu-period` that the container is limited to before throttled. As such acting as the effective ceiling. If you use Docker 1.13 or higher, use `--cpus` instead.
`--cpuset-cpus`	Limit the specific CPUs or cores a container can use. A comma-separated list or hyphen-separated range of CPUs a container can use, if you have more than one CPU. The first CPU is numbered 0. A valid value might be `0-3` (to use the first, second, third, and fourth CPU) or `1,3` (to use the second and fourth CPU).
`--cpu-shares`	Set this flag to a value greater or less than the default of 1024 to increase or reduce the container's weight, and give it access to a greater or lesser proportion of the host machine's CPU cycles. This is only enforced when CPU cycles are constrained. When plenty of CPU cycles are available, all containers use as much CPU as they need. In that way, this is a soft limit. `--cpu-shares` does not prevent containers from being scheduled in swarm mode. It prioritizes container CPU resources for the available CPU cycles. It does not guarantee or reserve any specific CPU access.

Conclusion

Best practices

- ➔ Harden your containers
- ➔ Isolate your containers
- ➔ Keep up to date the underlying operating system
- ➔ Use security tool to monitor your containers

Conclusion

Consider your containers as any physical machine and ensure their compliance towards your company security policies

THANKS FOR YOUR ATTENTION

QUESTIONS ?