


docker exec - without Docker



Oliver Seitz @ Web&Wine June 2023



What is docker exec?

→ The `docker exec` command runs a new command in a running container.

(<https://docs.docker.com/engine/reference/commandline/exec/>)

```
$ docker exec [-it] <container_id> <command>
```

What is docker exec?

→ The `docker exec` command runs a new command in a running container.

(<https://docs.docker.com/engine/reference/commandline/exec/>)

```
$ docker exec [-it] <container_id> <command>
```

Show me the content of the `/usr/share/nginx/html` folder inside my container:

```
$ docker exec ec074b7e737d ls -al /usr/share/nginx/html
```

Open an interactive bash in my container

```
$ docker exec -it ec074b7e737d bash
```

How does this magic work?

Docker

containerd

(<https://github.com/containerd/containerd>)

runc

(<https://github.com/opencontainers/runc>)

container

Looking deeper!

Docker

containerd

(<https://github.com/containerd/containerd>)

runc

(<https://github.com/opencontainers/runc>)

container

cgroups

(Linux Kernel)

namespaces

(Linux Kernel)

...

Secure by design?

Cgroups

Limiting Processes

- cpu
- memory
- pid
- io
- devices
- freezer
- rdma, ...

Quelle: <https://man7.org/linux/man-pages/man7/cgroups.7.html>

Quelle: <https://man7.org/linux/man-pages/man7/namespaces.7.html>

Bild Quelle: <https://unsplash.com/photos/3wPJxh-piRw>

Namespaces

Isolating Processes

- network
- mount
- pid
- user
- time
- uts
- cgroup
- ipc



Cgroups

Cgroups - cpu

QUOTA	PERIOD
max X us execution time per Y us period

Example: 20000 100000 → 20ms each 100ms → 20% of the time

Bild Quelle: <https://unsplash.com/photos/iYkqHp5cGQ4>

Cgroups - memory

Memory usage

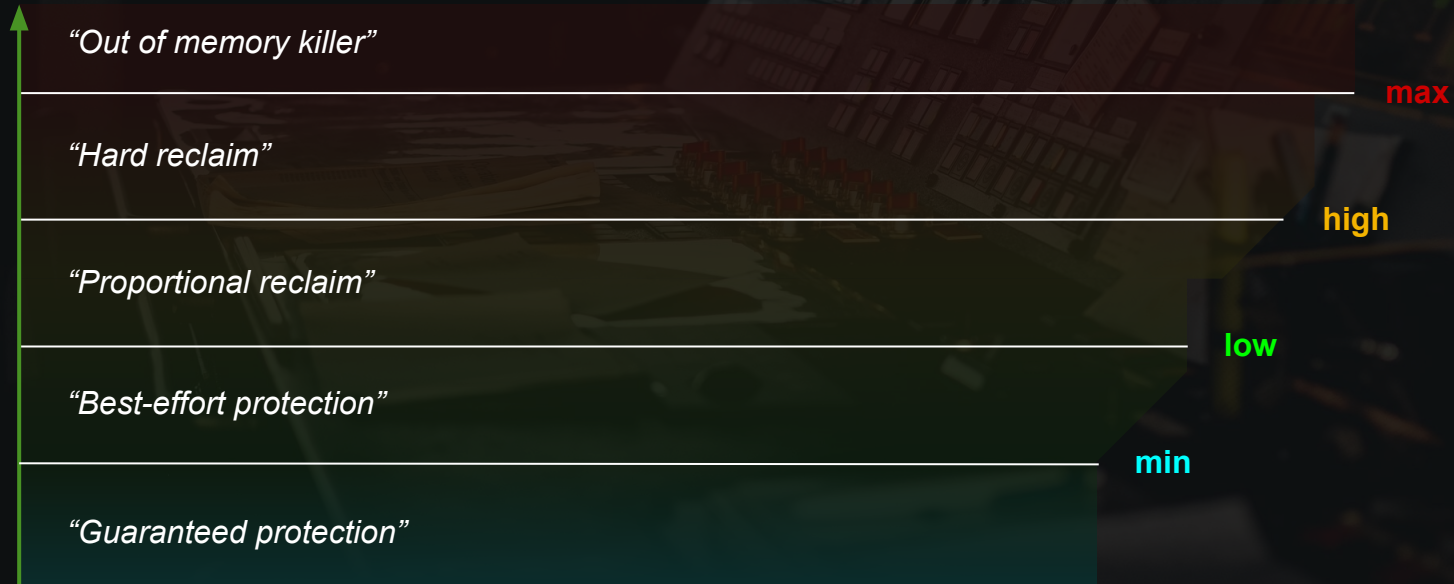


Bild Quelle: <https://unsplash.com/photos/iYkqHp5cGQ4>

Cgroups - pid

→ Limit the number of processes allowed in a cgroup

Bild Quelle: <https://unsplash.com/photos/iYkqHp5cGQ4>

Cgroups - io

rbps	Max bytes/s reading
wbps	Max bytes/s writing
riops	Max read IOPs
wiops	Max write IOPs

Bild Quelle: <https://unsplash.com/photos/iYkqHp5cGQ4>



Namespaces

Namespaces

There are eight types of namespaces in Linux:

network

mount

pid

user

time

uts

cgroup

ipc

Bild Quelle: <https://unsplash.com/photos/fAwcMpTjiRk>

Namespaces

There are eight types of namespaces in Linux:

network

mount

pid

user

time

uts

cgroup

ipc

 Provide isolation of network interfaces

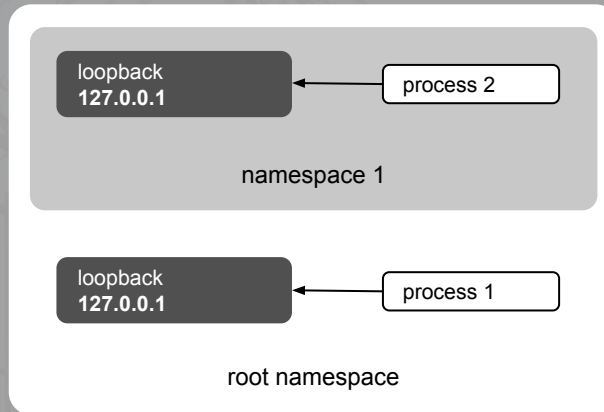


Bild Quelle: <https://unsplash.com/photos/fAwcMpTjiRk>

Namespaces

There are eight types of namespaces in Linux:

network

mount

pid


user

time

uts

cgroup

ipc

 Provide isolation for mounts

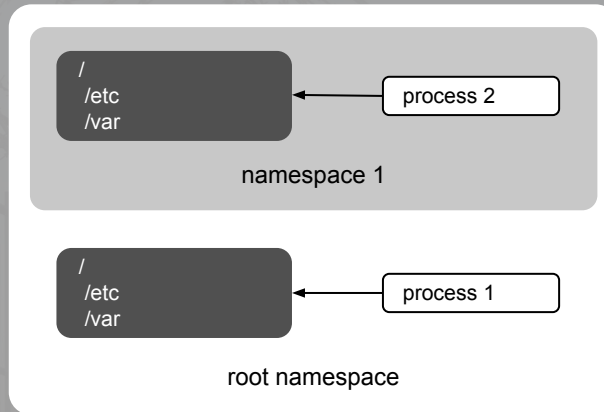


Bild Quelle: <https://unsplash.com/photos/fAwcMpTjiRk>

Namespaces

There are eight types of namespaces in Linux:

network

mount

pid

user

time

uts

cgroup

ipc

 Provide isolation for process ids

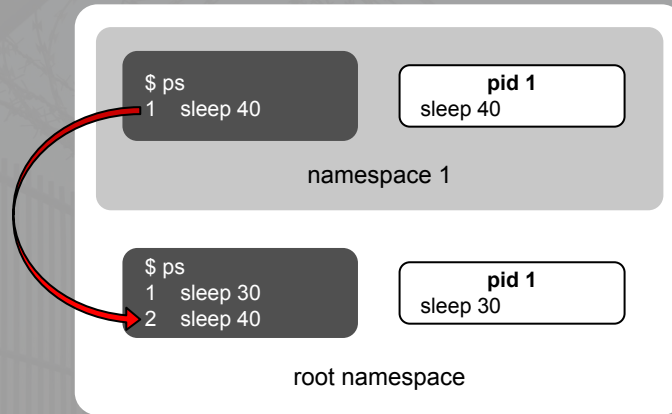


Bild Quelle: <https://unsplash.com/photos/fAwcMpTjiRk>

Namespaces

There are eight types of namespaces in Linux:

network

mount

pid

user

time

uts

cgroup

ipc

Provide isolation for users and groups

```
$ ls -n
-rw-r--r-- 1 65534 65534 0 Jun 22 01:25 test
-rw-r--r-- 1 0 0 0 Jun 22 01:34 test2
```

namespace 1

```
$ ls -n
-rw-r--r-- 1 0 0 0 Jun 22 01:25 test
-rw-r--r-- 1 1000 1000 0 Jun 22 01:34 test2
```

root namespace



Note

test was created in root ns

test2 was created in ns 1

Bild Quelle: <https://unsplash.com/photos/fAwcMpTjiRk>

CMD Dump of the demos (#1, #2 mean "use two different shells")

```
# =====  
# CGROUPS DEMO  
# =====
```

```
# apt install -y stress-ng
```

```
# stress-ng --cpu 1
```

```
# htop
```

```
# mkdir /sys/fs/cgroup/cpu/demo
```

```
# ls /sys/fs/cgroup/cpu/demo
```

```
# echo 10000 > /sys/fs/cgroup/cpu/demo/cpu.cfs_quota_us
```

```
# cat /sys/fs/cgroup/cpu/demo/cpu.cfs_quota_us
```

```
# cat /sys/fs/cgroup/cpu/demo/cpu.cfs_period_us
```

```
# ps -u # find pid
```

```
# echo [pid] > /sys/fs/cgroup/cpu/demo/cgroup.procs
```

```
# htop
```

```
# =====  
# NETWORK NAMESPACE DEMO  
# =====
```

```
#1 unshare -n --kill-child /bin/bash
```

```
#1 ip link
```

```
#1 ip link set dev lo up
```

```
#1 nc -l 8080
```

```
#2 ps -u # find pid
```

```
#2 nsenter -t [pid] -n /bin/bash
```

```
#2 nc 127.0.0.1 8080
```

```
# =====  
# EXEC DEMO  
# =====
```

```
#1 docker run -d -p 8080:80 -v ./html:/usr/share/nginx/html nginx
```

```
#1 docker exec -it [container-id] /bin/bash
```

```
#1 ls -alh /usr/share/nginx/html
```

```
#1 curl 127.0.0.1
```

```
#2 ps -aux # get pid of nginx
```

```
#2 nsenter -t [] -a /bin/bash
```