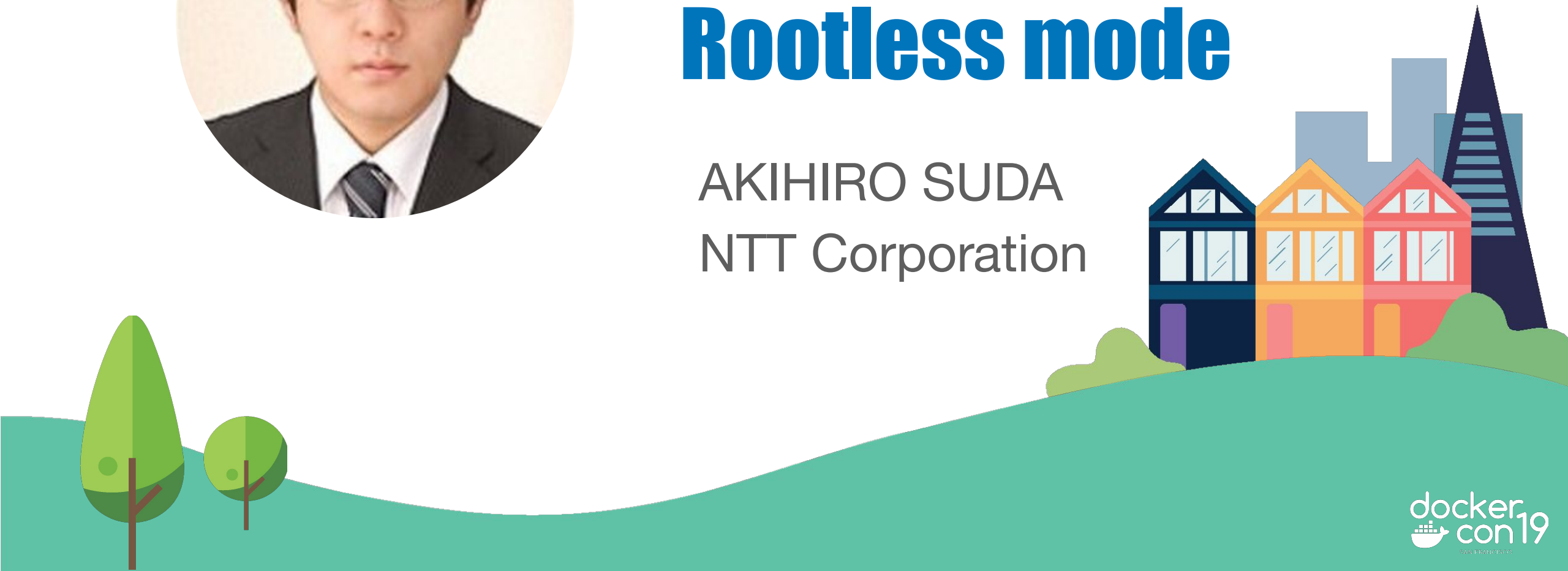
NTT

Hardening Docker daemon with Rootless mode

AKIHIRO SUDA
NTT Corporation



dockercon19

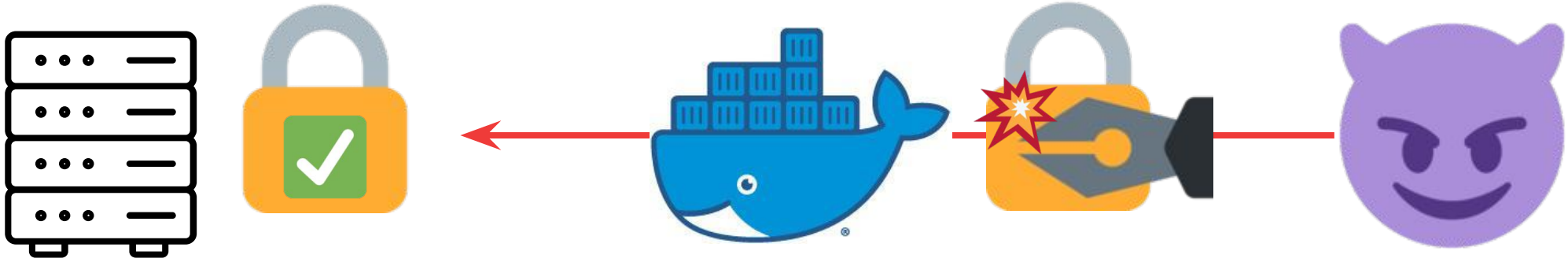
NTT

About me

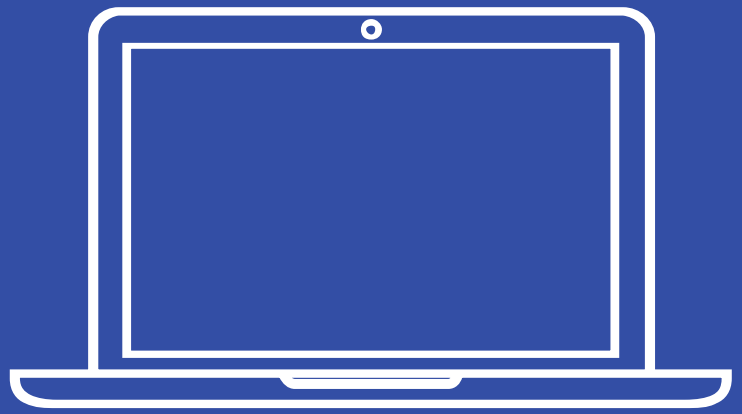
- Software Engineer at NTT
- Maintainer of Moby, containerd, and BuildKit
- Docker Tokyo Community Leader

Rootless Docker

- Run Docker as a non-root user on the host
- Protect the host from potential Docker vulns and misconfiguration



Demo



Don’t confuse with..

```
$ sudo docker
```

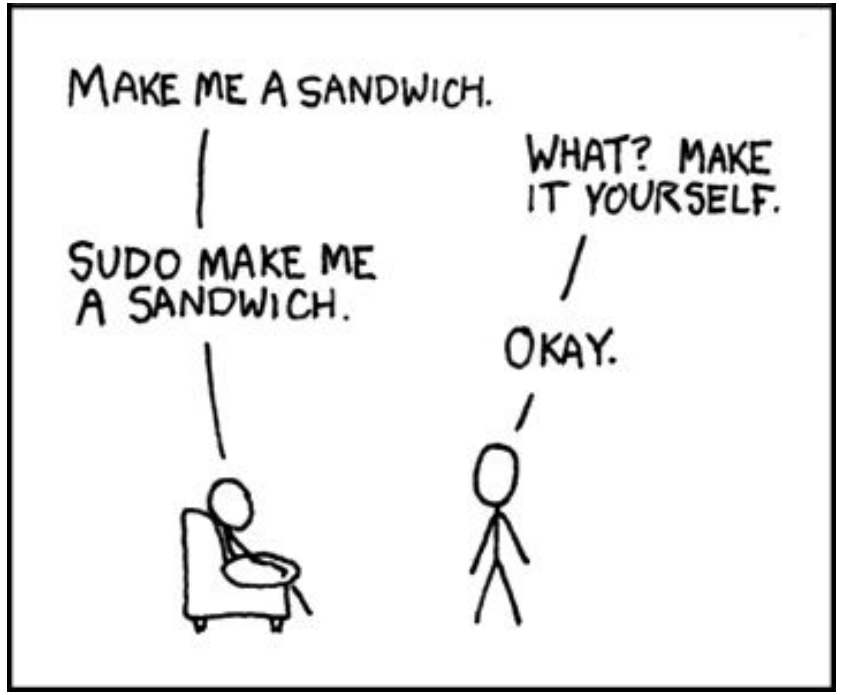


Image: https://xkcd.com/149/

Don’t confuse with..

```
$ sudo docker
```

```
$ usermod -aG docker penguin
```

~~Rootless~~ Docker

Non-root username: “penguin”

```
$ ls -l /var/run/docker.sock

srw-rw---- 1 root docker 0 May  1 12:00 /var/run/docker.sock

$ sudo usermod -aG docker penguin
```

Rootless Docker

Non-root

```
$ ls -l /va
srw-rw---                                  cker.sock
$ sudo us
```

Image: https://twitter.com/llegaspacheco/status/1111783777372639232

Rootless Docker

Non-root

$ ls -l /va

$ sudo us

docker run -v /:/host

Image: https://twitter.com/llegaspacheco/status/1111783777372639232

Don’t confuse with..

```
$ sudo docker
```

```
$ usermod -aG docker penguin
```

```
$ docker run --user 42
```

NTT

Don’t confuse with..

```
$ sudo docker
```

```
$ usermod -aG docker penguin
```

```
$ docker run --user 42
```

```
$ dockerd --userns-remap
```

All of them run the daemon as the root!

NTT

Rootless Docker

- Rootless Docker refers to running the Docker daemon (and containers of course) as a non-root user
- Even if it got compromised, the attacker wouldn't be able to gain the root on the host (unless you have `sudo` configured with `NOPASSWD`)

NTT

Some caveats apply..

- No OverlayFS (except on Ubuntu)
- Limited network performance by default
- TCP/UDP port numbers below 1024 can't be listened on
- No cgroup
  - `docker run: --memory` and `--cpu-*` flags are ignored
  - `docker top:` does not work

NTT

You can install it under your $HOME right now!

```
curl -fsSL https://get.docker.com/rootless | sh
```

- `sudo` is not required
- But `/etc/subuid` and `/etc/subgid` need to be configured to contain your username
  - configured by default on recent distros

NTT

You can install it under your $HOME right now!

```
curl -fsSL https://get.docker.com/rootless | sh
```

- The installer shows helpful error if `/etc/sub[ug]id` is unconfigured
  - Thanks to Tõnis Tiigi and Tibor Vass!
- Feel free to ask me after this session if it doesn't work

Katacoda scenario available!

https://www.katacoda.com/courses/docker/rootless

Katacoda

Rootless Docker

Step 2 of 4

Step 2 - Install Rootless Docker

Docker have made available a script which will deploy the required components for the new Rootless version.

Run the following command as *lowprivuser* to execute the script and install the components.

```
curl -sSL https://get.docker.com/rootless | sh ✓
```

After this has finished, proceed to the next step to setup the user environment and start launching containers.

CONTINUE

Terminal

```
lowprivuser@host01:~$ curl -sSL https://get.docker.com/rootless | sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 59.8M  100 59.8M    0     0  1700k      0  0:00:36  0:00:36 --:--:-- 1575k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 39 14.0M   39 5675k    0     0   787k      0  0:00:18  0:00:07  0:00:11  787k
```

Motivation

Harden containers

- Docker has a lot of features for hardening containers, so root-in-container is still contained by default
  - namespaces, capabilities
  - seccomp, AppArmor, SELinux...
- But there is no such thing as vulnerability-free software; root-in-container could break out with an exploit
  - CVE-2019-5736 runc breakout (Feb 11, 2019)

NTT

Harden containers

- And people often make misconfiguration!
- *"We found 3,822 Docker hosts with the remote API exposed publicly."*

  -- Vitaly Simonovich and Ori Nakar (March 4, 2019)

  https://www.imperva.com/blog/hundreds-of-vulnerable-docker-hosts-exploited-by-cryptocurrency-miners/

Harden containers

- Rootless mode per se doesn't fix vulns and misconfigurations - but it can mitigate attacks

- Attacker won't be able to:
  - access files owned by other users
  - modify firmware and kernel (→ undetectable malware)
  - ARP spoofing

Caution: not panacea!

- If Docker had a vuln, attackers still might be able to:
  - Mine cryptocurrencies
  - Springboard-attack to other hosts

- Not effective for potential vulns on kernel / VM / HW side

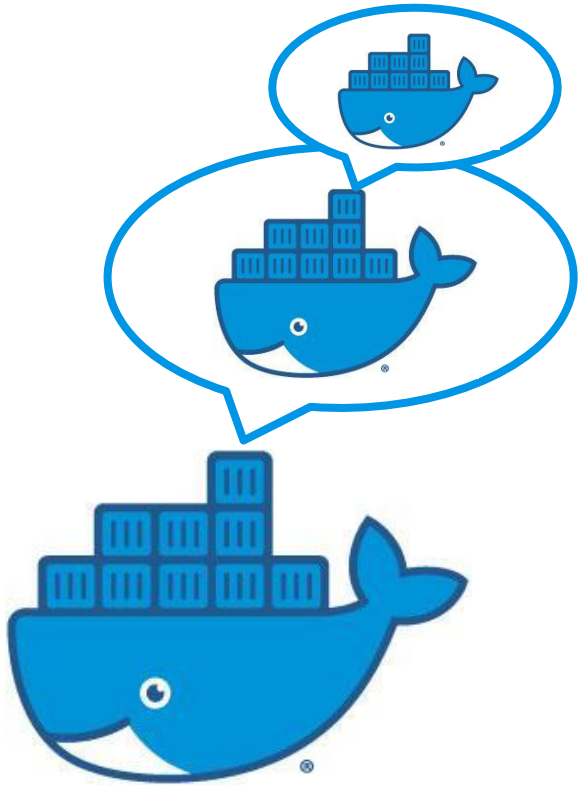
NTT

High-performance Computing (HPC)

- HPC users are typically disallowed to gain the root on the host
- Good news: GPU (and perhaps FPGA devices) are known to work with Rootless mode

Docker-in-Docker

- There are a lot of valid use cases to allow a Docker container to call Docker API
  - FaaS
  - CI
  - Build images
  - ...



NTT

Docker-in-Docker

- Two types of Docker-in-Docker, both had been unsafe without Rootless

```
$ docker run -v /var/run/docker.sock:/var/run/docker.sock
```

```
$ docker run --privileged docker:dind
```

How it works

Pretend to be the root

- User namespaces allow non-root users to pretend to be the root
- Root-in-UserNS can have fake UID 0 and also create other namespaces (MountNS, NetNS..)

Pretend to be the root

- But Root-in-UserNS cannot gain the real root
  - Inaccessible files still remain inaccessible
  - Kernel modules cannot be loaded
  - System cannot be rebooted

Pretend to be the root

```
$ id -u

1001

$ ls -ln

-rw-rw---- 1 1001 1001 42 May  1 12:00 foo
```

Pretend to be the root

```
$ docker run -v $(pwd):/mnt -it alpine

/ # id -u

0

/ # ls -ln /mnt

-rw-rw---- 1 0    0    42 May  1 12:00 foo
```

Still running as 1001 on the host

Still owned by 1001 on the host

NTT

Pretend to be the root

```
$ docker run -v /:/host -it alpine

/ # ls -ln /host/dev/sda

brw-rw---- 1 65534    65534   8,    0 May  1 12:00 /host/dev/sda

/ # cat /host/dev/sda

cat: can't open '/host/dev/sda': Permission denied
```

Still owned by root(0) on the host

Sub-users (and sub-groups)

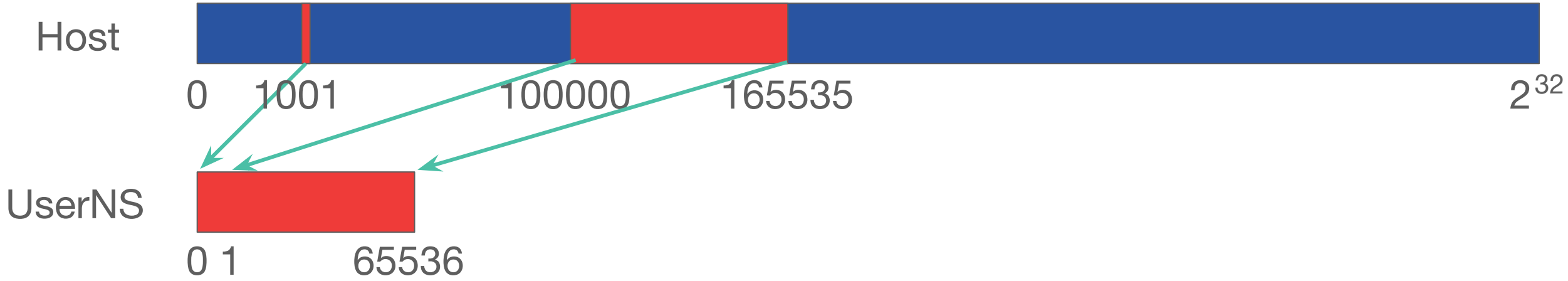
- Put users in your user account so you can be a user while you are a user

- Sub-users are used as non-root users in a container
  - `USER` in Dockerfile
  - `docker run --user`

Sub-users (and sub-groups)

primary user | sub-users start | sub-users len

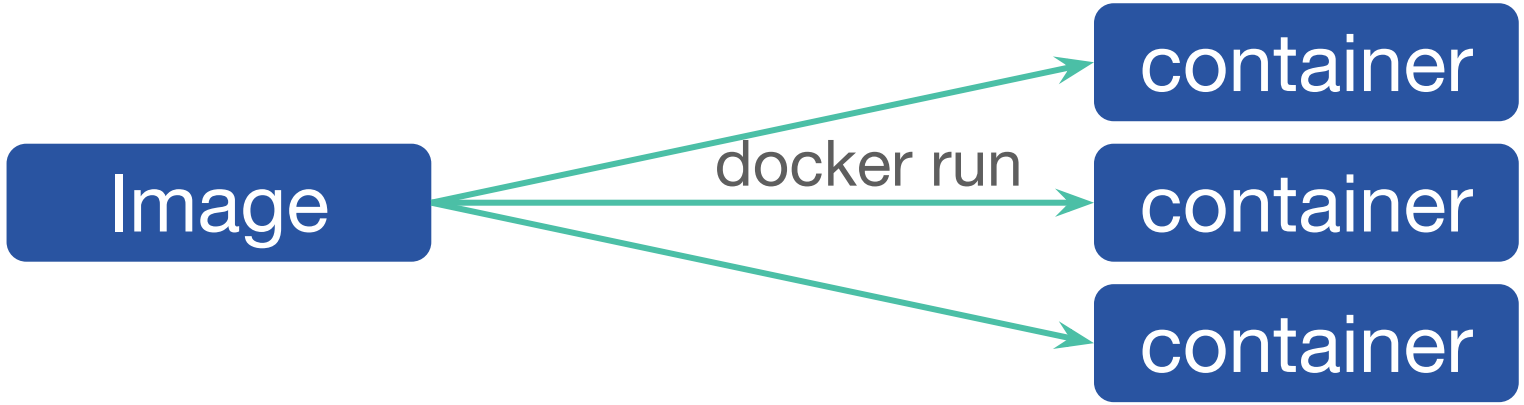
- If `/etc/subuid` contains "`1001:100000:65536`"



- Having 65536 sub-users should be enough for most containers

Snapshotting

- A container has a mutable copy of the image
- Copying file takes time and wastes disk space
- Rootful Docker uses OverlayFS to reduce extra copy

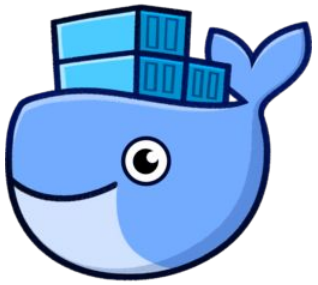


Snapshotting

- OverlayFS is currently unavailable for Rootless mode (unless you have Ubuntu's kernel patch)
- On ext4, files are just copied instead; Slow and wasteful
- But on XFS "reflink" is used to deduplicate files
  - `copy_file_range(2)`
  - Slow but not wasteful

Networking

- Non-root user can create NetNS but cannot create a vEth pair across the host and a NetNS
- VPNKit is used instead of vEth pair
  - User-mode network stack based on MirageOS TCP/IP
  - Also used by Docker for Mac/Win



Practical Tips

systemd service

```
$ systemctl --user start docker

$ systemctl --user stop docker
```

- The unit file is in your home:

  `~/.config/systemd/user/docker.service`

- To enable user services on system startup:

```
$ sudo loginctl enable-linger penguin
```

NTT

Enable OverlayFS

- The vanilla kernel disallows mounting OverlayFS in user namespaces
- But if you install Ubuntu kernel, you can get support for OverlayFS

  https://lists.ubuntu.com/archives/kernel-team/2014-February/038091.html

Enable XFS reflink

- If OverlayFS is not available, use XFS to deduplicate files
  - efficient for dedupe but slow
  - otherwise (i.e. ext4) all files are duplicated per layer
- `~/.config/docker/daemon.json`:

```
{“storage-driver”: “vfs”,
 “data-root”:”/mnt/xfs/foo”}
```

- Make sure to format with `` `mkfs.xfs -m reflink=1` ``,

Change network stack: slirp4netns

- The default network stack (VPNKit) is slow
- Install slirp4netns (v0.3.0+) to get better throughput
  - iperf3 benchmark (container to host):

    514Mbps → 9.21 Gbps
  - still slow compared to native vEth 52.1 Gbps

Benchmark: https://fosdem.org/2019/schedule/event/containers_k8s_rootless/

Change network stack: slirp4netns

- https://github.com/rootless-containers/slirp4netns
- `./configure && make && make install`
- RPM/DEB is also available for most distros (but sometimes outdated)
- If slirp4netns is installed on `$PATH`, Docker automatically picks up

Change network stack: lxc-user-nic

- Or install lxc-user-nic to get native performance
  - SETUID binary (executed as the root)
    - potentially result in root privilege escalation

      if lxc-user-nic had vuln

```
$ sudo apt-get install liblxc-common
```

Change network stack: lxc-user-nic

- `/etc/lxc/lxc-usernet` needs to be configured:

```
# USERNAME TYPE   BRIDGE   COUNT
penguin          veth     lxcbr0     1
```

Count of dockerd and LXC containers
(Not count of Docker containers)

- `$DOCKERD_ROOTLESS_ROOTLESSKIT_NET` needs to be set to `lxc-user-nic`

Exposing TCP/UDP ports below 1024

- Exposing port numbers below 1024 requires CAP_NET_BIND_SERVICE

```
$ sudo setcap cap_net_bind_service=ep \
    ~/bin/rootlesskit
```

```
$ docker run -p 80:80 ...
```

NTT

Future work

Docker 19.09? 20.03?

dockercon19 SAN FRANCISCO

NTT

FUSE-OverlayFS

- FUSE-OverlayFS can emulate OverlayFS without root privileges on any distro (requires Kernel 4.18)
- Faster than XFS dedupe but slightly slower than real OverlayFS
- containerd will be able to support FUSE-OverlayFS
- Docker will be able to use containerd snapshotter

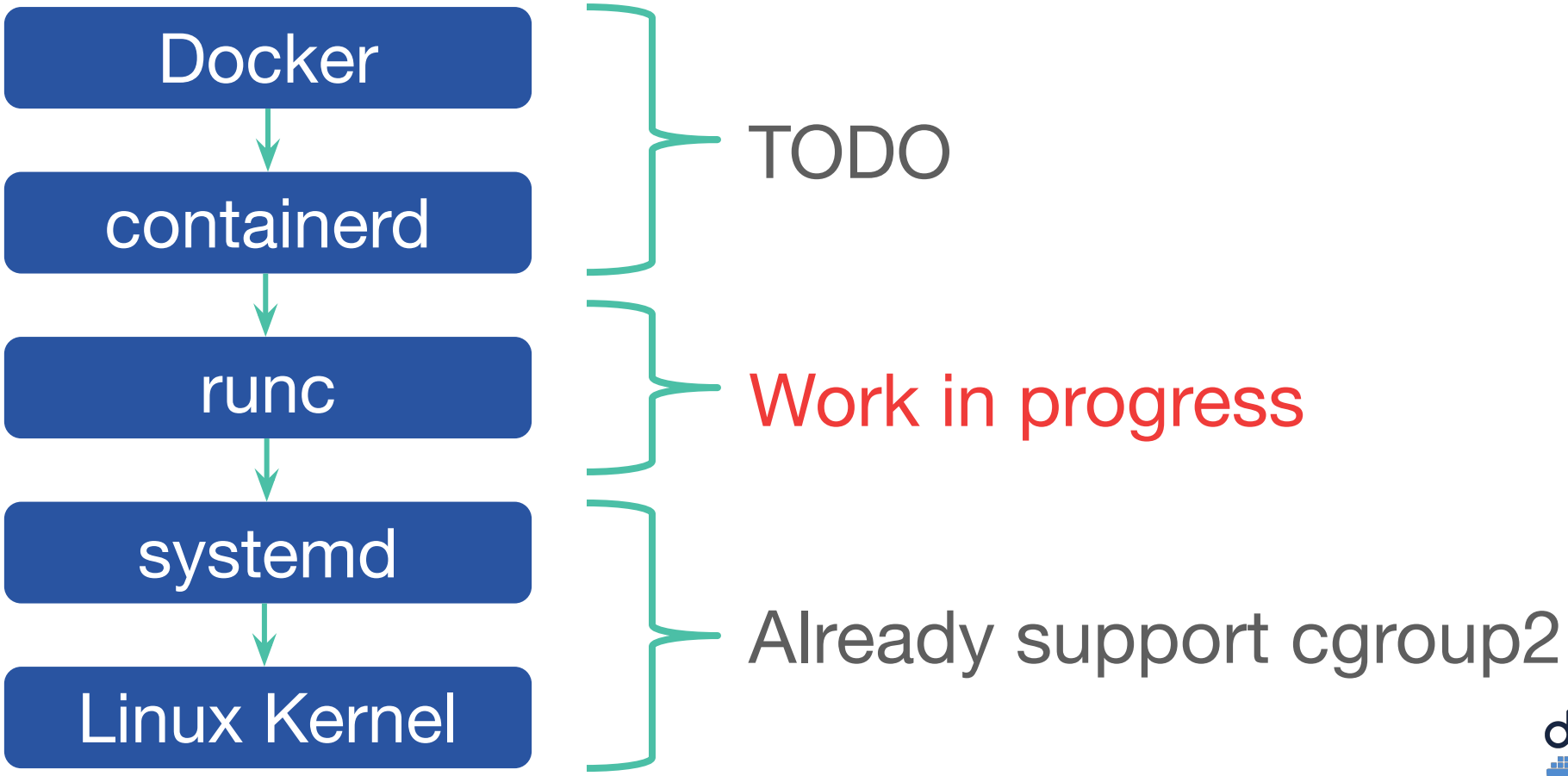
https://github.com/moby/moby/pull/38738

OverlayFS

- There has been also discussion to push Ubuntu's patch to the real OverlayFS upstream
- Likely to take more time?

cgroup2

cgroup2 is needed for safely supporting rootless cgroup



NTT

cgroup2

- runc doesn't support cgroup2 yet, but "crun" already supports cgroup2 https://github.com/giuseppe/crun
- OCI (Open Containers Initiative) is working on bringing proper cgroup2 support to OCI Runtime Spec and runc https://github.com/opencontainers/runtime-spec/issues/1002

NTT

LDAP

- Configuring `/etc/subuid` and `/etc/subgid` might be painful on LDAP environments
- NSS module is under discussion for LDAP environments

  https://github.com/shadow-maint/shadow/issues/154

  - No need to configure /etc/subuid and /etc/subgid

LDAP

- Another way: emulate sub-users using a single user
- runROOTLESS: An OCI Runtime Implementation with sub-users emulation https://github.com/rootless-containers/runrootless
  - Uses Ptrace and Xattr for emulating syscalls
  - 2-15 times performance overhead 😢

    https://github.com/rootless-containers/runrootless/issues/14

NTT

LDAP

- seccomp could be used for accelerating ptrace, but we are still facing implementation issues
- We are also looking into possibility of using "Seccomp Trap To Userspace" (introduced in Kernel 5.0)
  - Modern replacement for ptrace

Join us at Open Source Summit !

- Thursday, May 2, 12:30 PM - 02:30 PM
- Room 2020
- Three BuildKit talks

  including this →

NTT

get.docker.com/rootless

Questions?

docker con 19 SAN FRANCISCO