



Podman vs Docker

A Fedora fanboys perspective

FSCK 2023 // philipp // 03.06.2023

Agenda

- Why Podman?
- Architectural differences
- How to use (with compose)?
- Rootless mode and `sub{u,g}id` mapping
- Common problems
- Networking in usermode - or how to deploy an open mail relay
- ...and other funny stories

Why Podman?

- Docker is rootfull by default
- Podman is rootless
- Docker also provides an rootless mode - never tried
- Around 3 years ago Fedora dropped cgroup v1
 - cgroup v2 is way cleaner and more secure
 - Docker did not support cgroup v2
- Fedora promoted Podman as a new container tool
 - Toolbox technology also depends on Podman



vs



Architectural differences

Docker

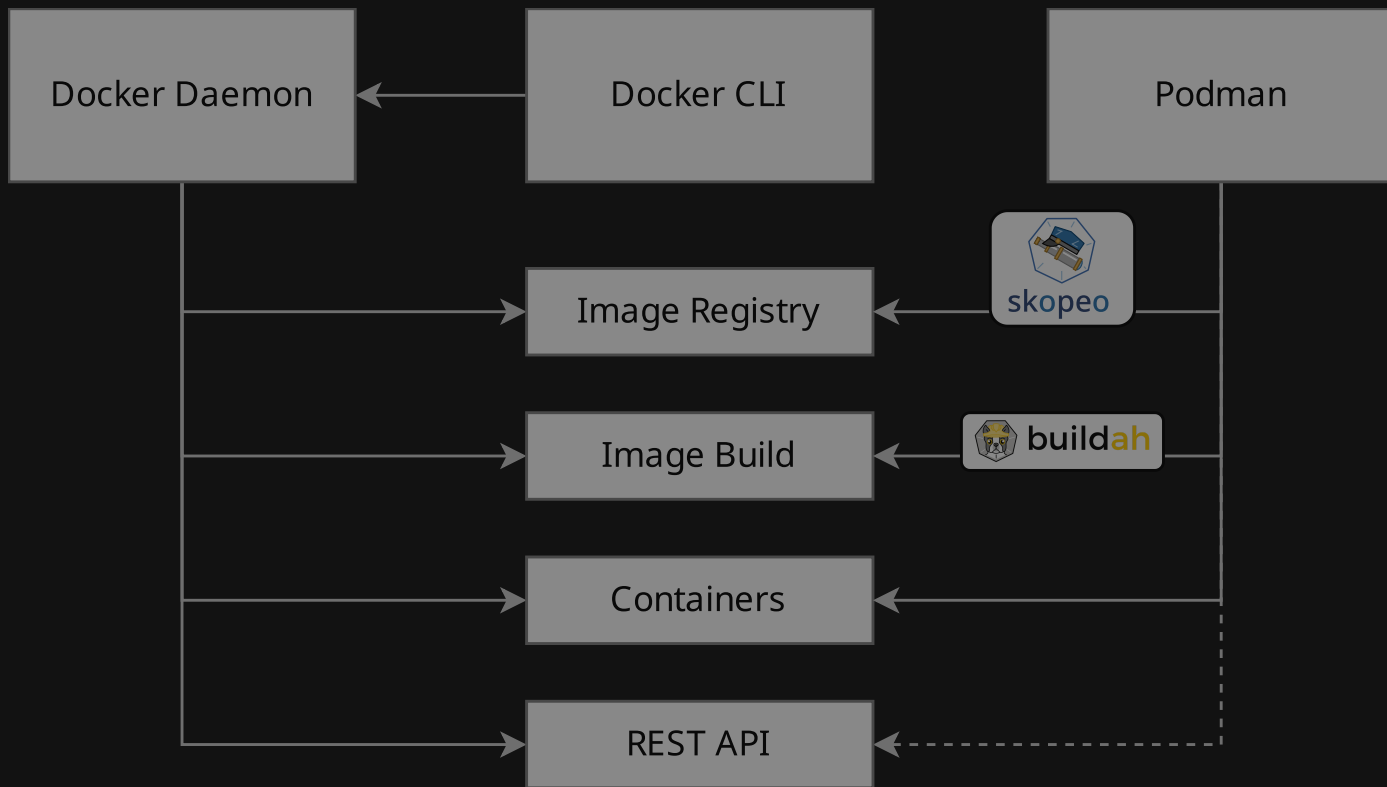
- Client server architecture
 - Requires daemon running in the background
- Docker daemon runs rootfull by default
- Docker group is identically powerful than sudo
- Nested containers require socket passthrough
- Monolithic application (run, build and pull containers)

Podman

- Podman has an optional socket
 - Docker REST API compliant
- Rootless by default
 - Sub{u,g}ids might need to be configured
- Nesting Podman inside containers works ootb
- Reuse of Buildah and skopeo ("UNIX philosophy")
- Podman/buildah/skopeo can easily be used in unprivileged CI/CD pipelines where docker fails

Podman Desktop is a free (as in free beer) and open source alternative to Docker Desktop

Architectural differences



How to use (with compose)?

- Podman Compose
 - Converts compose file into Podman CLI commands
 - More verbose (ugly) output
 - Sequential task processing
- Docker Compose
 - Cleaner CLI output
 - Parallel task processing
 - Podman 4.X strongly recommended

```
systemctl --user start podman.socket  
export DOCKER_HOST="unix://${XDG_RUNTIME_DIR}/podman/podman.sock"
```

How to use (with compose)?

- Podman Compose
 - Converts compose file into Podman CLI commands
 - More verbose (ugly) output
 - Sequential task processing
- Docker Compose
 - Cleaner CLI output
 - Parallel task processing
 - Podman 4.X strongly recommended

```
systemctl --user start podman.socket  
export DOCKER_HOST="unix://${XDG_RUNTIME_DIR}/podman/podman.sock"
```

Source: <https://twitter.com/ialanmoran/status/1001671953571303425>



← Thread



WoodETH
@ialanmoran



I completely forgot that ~2 months ago I set up "alias docker='podman'" and it has been a dream. [#nobigfatdaemons](#) [@projectatomic](#)

5:49 AM · May 30, 2018

10 Retweets 20 Likes



WoodETH  @ialanmoran · May 30, 2018



Replying to [@ialanmoran](#)

Only downside is no Mac OS support (Main dev machine)



Joe Thompson @caffeinepresent · May 30, 2018




Replying to [@ialanmoran](#) and [@projectatomic](#)

So, what reminded you?



1



WoodETH  @ialanmoran · May 31, 2018



Replying to [@caffeinepresent](#) and [@projectatomic](#)

docker help 🙄



1



Security implications of rootfull containers

```
[philipp@spectre ~]$ id
uid=1000(philipp) gid=1000(philipp) groups=1000(philipp),10(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[philipp@spectre ~]$ docker run --rm -v /:/host-root -it alpine sh
/ # id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy),20(dialout),26(tape),27(video)
/ # █
```

- Root folder of host can easily be mounted to rootfull container
 - ``docker`` group basically equals ``sudo``
 - ``sudo`` can be properly logged using e.g. ``auditd``
 - ``docker`` makes logging hard by using client-server-architecture
 - Ability to create files you cannot remove/modify with your user

Rootless mode and `sub{u,g}id` mapping

- Rootfull containers use 1:1 mapping
- `docker` group equals `sudo` permissions
 - To try out run `docker run --rm -v /:/host:Z -it alpine`
- Subid ranges for users and groups in rootless containers
 - root => host user
 - user => subid
- Subid mappings result in permissions issues
 - `podman unshare <command>`
 - e.g. `podman unshare rm -r ./pgdata/`

Rootless networking: Privileged ports

- Rootless containers can only expose ports >1024 on the host

Allow rootlesskit to bind privileged ports

```
sudo setcap cap_net_bind_service=ep $(pwd rootlesskit)
```

Better: Redirect specific ports using e.g. `firewall-cmd` or `nftables`

```
firewall-cmd --permanent --add-forward-port=port=80:proto=tcp:toport=8080
```

Rootless networking: Privileged ports

- Rootless containers can only expose ports >1024 on the host

Allow rootlesskit to bind privileged ports

```
sudo setcap cap_net_bind_service=ep $(pwd rootlesskit)
```

Better: Redirect specific ports using e.g. `firewall-cmd` or `nftables`

```
firewall-cmd --permanent --add-forward-port=port=80:proto=tcp:toport=8080
```

How not to do it:

```
echo 0 | sudo tee /proc/sys/net/ipv{4,6}/ip_unprivileged_port_start
```

Problems using Podman

- Compose does not work
 - Enable socket for compatibility
 - Use podman 4.X (do not use Debian 10)
- Volume mount permissions (SELinux)
 - Append `:z` to volume mount
 - Common first-time-fedora-user problem
- Pod2Pod DNS resolution does not work
 - Install `aardvark-dns` (typical arch issue)
- No sub{u,g}ids are available
 - Add entries to `/etc/sub{u,g}id` as shown previously

Remember

- `podman system migrate`
- `podman unshare <command>`
- (or just use fedora)

```
/etc/containers/registries.conf
```

```
unqualified-search-registries = ["quay.io", "docker.io"]
```

Problems using NSS and LDAP

- Subids are generated automatically by `useradd` command
 - LDAP accounts are fetched using NSS
- Writing own NSS plugin for fetching the subids...
- ...or doing some shell shenanigans

Problems using NSS and LDAP

- Subids are generated automatically by ``useradd`` command
 - LDAP accounts are fetched using NSS
- Writing own NSS plugin for fetching the subids...
- ...or doing some shell shenanigans

```
#!/usr/bin/env bash

range=65536

getent passwd | grep '/home-selfnet' | tr ':' ' ' | sort -nk 3 |
while read user _ uid _; do
    echo "${user}:$(( ${uid} * ${range} - 1 )):${range}"
done | uniq
```

Subid ranges are generated to start from ``$uid * 216`` until ``$uid * 216 + 216 - 1``. This way, each user has 2¹⁶ available subids and no overlap between individual users happens. Such subid ranges are usually generated by the ``useradd`` command, but this is currently not automatically done for LDAP users.

Story Time □

Rootless networking: Network mode

- Host network interfaces cannot be created in rootless mode
- Default Podman 3.X network driver resolves addresses to localhost
 - Security impact on IP whitelists
- Possible solution: Switch to slirp4netns (default since Podman 4.X)
 - uses hardcoded `tap0` interface

```
services:
  mailserver:
    network_mode: "slirp4netns:port_handler=slirp4netns"
    environment:
      - NETWORK_INTERFACE=tap0
      ...
```

Rootless networking: Postfix config

```
case "${PERMIT_DOCKER}" in
    "host" )
        _notify 'inf' "Adding ${CONTAINER_NETWORK}/16 to my networks"
        postconf -e "$(postconf | grep '^mynetworks =') ${CONTAINER_NETWORK}/16"
        ;;
    "network" )
        _notify 'inf' "Adding docker network in my networks"
        postconf -e "$(postconf | grep '^mynetworks =') 172.16.0.0/12"
        ;;
    * )
        _notify 'inf' 'Adding container ip in my networks'
        postconf -e "$(postconf | grep '^mynetworks =') ${CONTAINER_IP}/32"
        ;;
```

Rootless networking: Postfix config

```
case "${PERMIT_DOCKER}" in
    "host" )
        _notify 'inf' "Adding ${CONTAINER_NETWORK}/16 to my networks"
        postconf -e "$(postconf | grep '^mynetworks =') ${CONTAINER_NETWORK}/16"
        ;;
    "network" )
        _notify 'inf' "Adding docker network in my networks"
        postconf -e "$(postconf | grep '^mynetworks =') 172.16.0.0/12"
        ;;
    "none" )
        _notify 'inf' "Clearing Postfix's 'mynetworks'"
        postconf -e "mynetworks ="
        ;;
    * )
        _notify 'inf' 'Adding container ip in my networks'
        postconf -e "$(postconf | grep '^mynetworks =') ${CONTAINER_IP}/32"
        ;;
```

...and even more stories

The screenshot shows the GitLab interface for the 'Clickable' project. The left sidebar contains navigation links: Clickable, Project overview, Pinned, Issues (35), Merge requests (6), Manage, Plan, Code, Build, Deploy, Operate, and Monitor. The main content area is titled 'Clickable > Clickable > Issues'. It features filters for 'Open 1', 'Closed 4', and 'All 5'. A search bar contains the text 'podman'. Below the filters, four issues are listed:

- Podman ~/.local/share/containers takes A LOT (45GB!) of disk space building simple app**
#390 · created 5 months ago by Jami Kettunen
Labels: container, question
- Start using fully qualified container image names for Podman**
#389 · created 5 months ago by Jami Kettunen
Labels: bug, container
- Clickable with podman doesn't install symlinks in .click's**
#382 · created 7 months ago by Semphriss
Labels: bug, container
- Support Podman / Fedora**
#367 · created 10 months ago by Philipp Fruck
Label: feature request

Thanks! - Questions?

FCK 2023 // philipp // 03.06.2023