# Container Tooling by Red Hat

Franta Lachman

Tomas Tomecek

# Docker done right

## Docker done right

(You make your own opinion :)

#### Who's Franta and Tomas?

## The tools

- podman
- buildah
- skopeo
- udica
- toolbox
- CRI O

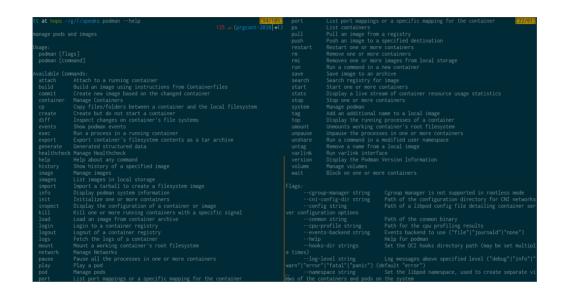
#### The tools

- podman
- buildah
- skopeo
- udica
- toolbox
- CRI O
- Demos: <a href="https://github.com/containers/Demos.git">https://github.com/containers/Demos.git</a>

# The background tools

- crun
- bubblewrap
- fuse-overlayfs
- conmon
- containers/{storage,image}

# podman



# podman (T)

- github.com/containers/libpod
- "The manager of pods"
- The same CLI as docker command

- Credits:
  - https://www.slideshare.net/AkihiroSuda/rootless-containers

- Credits:
  - https://www.slideshare.net/AkihiroSuda/rootless-containers
- Running a container as an unprivileged user

- Credits:
  - https://www.slideshare.net/AkihiroSuda/rootless-containers
- Running a container as an unprivileged user
- Very tricky to be done on the system level
  - User namespaces (UID mapping), Filesystem, Networking

- Credits:
  - https://www.slideshare.net/AkihiroSuda/rootless-containers
- Running a container as an unprivileged user
- Very tricky to be done on the system level
  - o User namespaces (UID mapping), Filesystem, Networking
- Demo github.com/containers/Demos/security

- Credits:
  - $\begin{tabular}{ll} $\circ$ $ $https://devconfcz2020a.sched.com/event/YOo2/the-state-of-container-security \\ \end{tabular}$

- Credits:
  - $\circ \ \underline{https://devconfcz2020a.sched.com/event/YOo2/the-state-of-container-security}$
- Security is not very popular.
- Security hates user experience (setenforce 0).

- Credits:
  - $\begin{tabular}{ll} $\circ$ $ $https://devconfcz2020a.sched.com/event/YOo2/the-state-of-container-security \\ \end{tabular}$
- Security is not very popular.
- Security hates user experience (setenforce 0).
- podman run --privileged VS. podman run --cap-drop

- Credits:
  - <a href="https://devconfcz2020a.sched.com/event/YOo2/the-state-of-container-security">https://devconfcz2020a.sched.com/event/YOo2/the-state-of-container-security</a>
- Security is not very popular.
- Security hates user experience (setenforce 0).
- podman run --privileged VS. podman run --cap-drop
- SELinux blocked most of docker's CVEs
  - breakouts of containers (filesystem based)

# Oci-seccomp-bpf-hook

- <a href="https://github.com/containers/oci-seccomp-bpf-hook">https://github.com/containers/oci-seccomp-bpf-hook</a>
- Generate a seccomp profile based on live container behaviour.

# Oci-seccomp-bpf-hook

- <a href="https://github.com/containers/oci-seccomp-bpf-hook">https://github.com/containers/oci-seccomp-bpf-hook</a>
- Generate a seccomp profile based on live container behaviour.
- Also, what is seccomp?

## Oci-seccomp-bpf-hook

- <a href="https://github.com/containers/oci-seccomp-bpf-hook">https://github.com/containers/oci-seccomp-bpf-hook</a>
- Generate a seccomp profile based on live container behaviour.
- Also, what is seccomp?
- No demo, the tool is not packaged, yet.

### udica

- Credits
- Examines container configuration
- Generate SELinux policy
- Demo: Demos/security/SELinuxUdica



# Features in podman and not in docker (F)

- <u>upstream table</u>
- UX enhancements
- not-merged proposals for docker
- pod manipulation

#### Features in podman and not in docker I/VII

- podman container checkpoint
- podman container restore

Checkpoints one or more running containers. The container name or ID can be used.

```
Usage:
   podman container checkpoint [flags] CONTAINER [CONTAINER...]

Examples:
   podman container checkpoint --keep ctrID
   podman container checkpoint --all
   podman container checkpoint --leave-running --latest
```

#### Features in podman and not in docker II/VII

- podman container cleanup
- podman container exists
- podman container runlabel
- podman healthcheck run

```
Examples:

podman container cleanup --latest

podman container cleanup ctrID1 ctrID2 ctrID3

podman container cleanup --all

podman container exists containerID

podman container exists myctr || podman run --name myctr [etc...]
```

#### Features in podman and not in docker III/VII

- podman image exists
- podman image sign | podman image trust
- podman image tree

# Features in podman and not in docker IV/VII

- podman mount
- podman umount

(Un)mount a working container's root filesystem.

#### Features in podman and not in docker V/VII

• podman varlink

```
Run varlink interface

Description:
Run varlink interface. Podman varlink listens on the specified unix domain socket for
Tools speaking varlink protocol can remotely manage pods, containers and images.

Usage:
podman varlink [flags] [URI]

Examples:
podman varlink unix:/run/podman/io.podman
podman varlink --timeout 5000 unix:/run/podman/io.podman
```

## Features in podman and not in docker VI/VII

• podman system service

Run API service

Description: Run an API service

Enable a listening service for API access to Podman commands.

# Features in podman and not in docker VII/VII

- podman play
- podman play kube
- podman generate
- podman generate kube
- podman pod \*

# podman $\rightarrow$ k8s (T)

- podman can create definitions based on live containers
- podman generate kube
- podman generate systemd

# podman $\rightarrow$ k8s (T)

- podman can create definitions based on live containers
- podman generate kube
- podman generate systemd
- Demo!

## skopeo (T)

- move container images around
- registry, dockerd, containers/storage, a file
- transports

```
skopeo inspect docker://docker.io/usercont/packit
{
    "Name": "docker.io/usercont/packit",
    "Digest": "sha256:4918b3a8511b7ca91...
    "RepoTags": [
            "latest",
            "prod"
    ],
    "Created": "2020-02-19T15:24:13.408449229Z",
```



# buildah (F)

- shell-based container builder tool
- no container runtime deamon needed
- simple, efficient



#### buildah: benefits

- no need for build/install requirements in the final image
- read/write volumes during the build
- rootless usage
  - o buildah images available at <u>quay.io/repository/buildah/stable</u>
- buildah build-using-dockerfile

## toolbox (F)

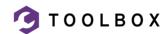
Toolbox is a tool that offers a familiar package based environment for developing and debugging software that runs fully unprivileged using Podman.

[user@hostname ~]\$ toolbox create
Created container: fedora-toolbox-30

Enter with: toolbox enter

[user@hostname ~]\$ toolbox enter

●[user@toolbox ~]\$



#### You've seen these demos

- intro (F)
- rootless containers (T)
- building images (F)
  - basic
  - speeding up builds
- security (T)
  - basic
  - o udica
- from podman to k8s

#### The end

- github.com/TomasTomecek/speaks
- github.com/lachmanfrantisek
- 😝 gitlab.com/lachmanfrantisek
- o github.com/containers
- <u>© @TomasTomec</u>
- S blog.tomecek.net