Podman, Buildah and Skopeo

“coreutils” for containers
Aleksandra Fedorova

- DevOps/CI/Build Engineer
- CI Engineer at Red Hat
- Fedora Linux Ambassador
- https://medium.com/quantum-integration
- Fedora User Group NRW @ Meetup.com
coreutils

- cd, cp, ls, cat, sort,..
- small, simple, consistent
- easy to use
- easy to embed
- available for any user
- available on any system(*)

https://opensource.com/article/18/4/gnu-core-utilities
Docker
docker pull
docker build
docker run

Docker service

container
container
root permissions

user permissions
open container initiative

- Organization established by Docker and other companies in 2015
- Defines open specifications for containers and container images
- OCI Image Specification
- OCI Runtime Specification
- Sets a framework for container ecosystem

https://www.opencontainers.org/
Toolbox
$ cat ./message
Hello world!
$ cat ./Dockerfile
FROM fedora
RUN touch somefile
ADD message .
CMD cat message

$ buildah bud -t myImage .
...

$ podman run myImage
Hello world!
$ buildah bud . test -t 1.0.0
$ cat ~/Dockerfile | buildah bud -f - .
$ buildah bud -f Dockerfile.first -f Dockerfile.second .
$ buildah bud github.com/somewhere/something
$ buildah bud -f dev/Dockerfile https://10.0.0.10/context.tar.gz
#!/bin/bash

containerId=$(buildah from fedora)
bUILDah run $containerId touch somefile
buildah add $containerId message
buildah config --cmd "cat message" $containerId
buildah commit $containerId myImage
# containerId=$(buildah from scratch)
# mountpath=$(buildah mount $containerId)
# dnf install --installroot $mountpath --release=29 buildah \
    --setopt install_weak_deps=false -y
# dnf --installroot $mountpath clean all
# buildah umount $containerId
# buildah commit $containerId myimage
# podman run myimage buildah --help
- Does not require access to system-wide service
- Stores temporary data and images in the user directory
- Supports multiple Dockerfiles, from local fs and network
- Allows interactive debugging of the image building process
- Easily embeddable in scripts

https://buildah.io
https://github.com/containers/libpod
alias docker=podman

$ podman run -it fedora /bin/bash
$ podman ps -a
$ podman rm deda2991f9fd
$ podman image rm fedora
$ podman pull debian:latest
$ podman inspect 7ac0a23445fe

...running-containers-with-podman
embedding podman in a systemd unit

```
[root@localhost ~]# cat /etc/systemd/system/mariadb-podman.service
[Unit]
Description=Custom MariaDB Podman Container
After=network.target

[Service]
Type=simple
TimeoutStartSec=5m
ExecStartPre=-/usr/bin/podman rm "mariadbpodman"
ExecStart=/usr/bin/podman run --name mariadbpodman -v /root/mysql-data:/var/lib/mysql/data:Z -e MYSQL_USER=user -e MYSQL_PASSWORD=pass -e MYSQL_DATABASE=db -p 3306:3306 registry.access.redhat.com/rhscl/mariadb-102-rhel7
ExecReload=-/usr/bin/podman stop "mariadbpodman"
ExecReload=-/usr/bin/podman rm "mariadbpodman"
ExecStop=/usr/bin/podman stop "mariadbpodman"
Restart=always
RestartSec=30

[Install]
```
- Does not require access to system-wide service
- Stores temporary data and images in the user directory
- Drop-in replacement for docker
- Runs container as a child process
- Provides resource isolation
- Easily embeddable in systemd unit files with cgroups, resource management, socket activation and so on

https://github.com/containers/libpod
https://github.com/containers/skopeo
$ skopeo inspect docker://docker.io/fedora
{
  "Name": "docker.io/library/fedora",
  "Digest": "sha256:4a861283a7f0a8ce3d19b42f4c0a10d7012a4d12f785149d82a0800cdbc498b0",
  "RepoTags": ["29", "30", "latest", "modular"],
  "Created": "2018-11-07T00:23:59.230741167Z",
  "DockerVersion": "17.06.2-ce",
  "Labels": null,
  "Architecture": "amd64",
  "Os": "linux",
  "Layers": ["sha256:d0483bd5a55488f5ba6383a5cc8553d5101864f03acd07eabc5df7563c3692cf"
  ]
}
$ mkdir output
$ skopeo copy docker://fedora:latest dir:output
$ ls output/
8c568f1043264e34f0a8774587266565c7e5e54e9ea6b97ab459086d18ac5175d0483bd5a55488f5ba6383a5cc8553d5101864f03acd07eabc5df7563c3692cf
manifest.json
version

https://github.com/containers/skopeo
- Does not require access to system-wide service
- Stores temporary data and images in the user directory
- Inspect images without downloading them
- Copy images without retagging them
- Supports multiple storage formats – registries, filesystems

https://github.com/containers/libpod
storages and registries

$ ls /etc/containers/
policy.json
registries.conf
registries.d
storage.conf
$ man storage.conf
Aleksandra Fedorova

- Fedora User Group NRW @ Meetup.com
- bookwar @ IRC
- https://telegram.me/bookwar
- bookwar@fedoraproject.org