Tutorial

PREPARED FOR YOU

YOURNAME

Version 0.2

Table of Contents

1. History and Revisions . . . 1
2. Preface . . . 2
   2.1. Confidentiality, Copyright, and Disclaimer . . . 2
   2.2. About This Document . . . 2
   2.3. Audience . . . 2
   2.4. Additional Background and Related Documents . . . 2
   2.5. Terminology . . . 2
3. OCI . . . 3
4. Buildah and Podman overview . . . 4
   4.1. Buildah . . . 4
   4.2. Podman . . . 4
   4.3. Comparison . . . 4

1. History and Revisions

Version	Date	Authors	Changes
0.2		YOURNAME <yourname@youremail.com>	Initial version of the document

2. Preface

2.1. Confidentiality, Copyright, and Disclaimer

This is public document and you can do what you like with it

2.2. About This Document

This document contains details of how podman and buildah works.

2.3. Audience

The audience of the document is YOU’s Linux Administrators

2.4. Additional Background and Related Documents

This document does not contain anything extra

2.5. Terminology

Provide a glossary for terminology that may not be common knowledge with the intended audience. Define terms and expand then define acronyms. If the terminology table exceeds a full page in length, it should probably be moved to an Appendix with a reference to the appendix in this section in place of the table.

Table 1. Terminology Table

Term	Definition
OCI	Open Container Initiative
Image	container image
Runtime	container runtime is software that executes containers and manages images

3. OCI

Long before Open Container Initiative (OCI) was created there were many different container runtimes, each with it’s own specification. The OCI was created as open governance body formed by Docker and other container leaders under the auspices of the Linux Foundation, for the purpose of creating unified open industry standard. Currently this standard contains the Runtime Specification and the Image Specification (image-spec).

4. Buildah and Podman overview

4.1. Buildah

The Buildah is CLI tool to create container. You can either start from scratch or use existing image from image registries like docker.com, quay.io or redhat.com. In our case we’re using Fedora image which is served by docker.com

```
ctr1=$(buildah from "${1:-fedora}")
```

- images can be built in either the OCI image format or the traditional upstream docker * image format
- mount and umount working container’s root filesystem for manipulation
- use the updated contents of a container’s root filesystem as a filesystem layer to create * a new image

4.2. Podman

Podman is a daemonless container engine for developing, managing, and running OCI Containers on your Linux System.

4.3. Comparison

Buildah specializes in building OCI images and can replicate all of the commands that are found in a Dockerfile. This allows building images with and without Dockerfiles while not requiring any root privileges.

Podman specializes in all of the commands and functions that help you to maintain and modify OCI images, such as pulling and tagging. It also allows you to create, run, and maintain those containers created from those images. For building container images via Dockerfiles, Podman uses Buildah’s golang API and can be installed independently from Buildah.

A major difference between Podman and Buildah is their concept of a container. Podman allows users to create "traditional containers" where the intent of these containers is to be long lived. While Buildah containers are really just created to allow content to be added back to the container image. An easy way to think of it is the buildah run command emulates the RUN command in a Dockerfile while the podman run command emulates the docker run command in functionality. Because of this and their underlying storage differences, you can not see Podman containers from within Buildah or vice versa.

In short, Buildah is an efficient way to create OCI images while Podman allows you to manage and maintain those images and containers in a production environment using familiar container cli commands. For more details, see the Container Tools Guide.

Let’s take a look at our script called journal.sh

```
#!/bin/bash

ctr1=$(buildah from "${1:-fedora}")

## Get all updates and install ruby
buildah run "$ctr1" -- dnf update -y
buildah run "$ctr1" -- dnf install -y rubygems liberation-fonts-common
buildah run "$ctr1" -- gem install asciidoctor
buildah run "$ctr1" -- gem install --pre asciidoctor-pdf
buildah run "$ctr1" -- gem install coderay pygments.rb rouge
buildah run "$ctr1" -- mkdir -p /workdir

## Include some buildtime annotations
buildah config --annotation "com.example.build.host=$(uname -n)" "$ctr1"

## Commit this container to an image name
buildah commit "$ctr1" "${2:-$USER/journal}"
```