Diff

A : A file or Directory was Added
C : A file or Directory was Changed
D : A file or Directory was Deleted

Commit

Straightforward Approach, but not recommended

Create an image from a running Container

podman commit <name_of_container> <name_of_image>

Sharing

1. Push to a Registry Server

- Firstly tag the image according to which registry server you are willing to push it to
podman tag <old_image_name> <new_image_tag>
- podman push <image_name>:<tag>

2. We create a Compressed version (.tar) of the Image and then we can share it

Send it

podman save <image_name> -o <name_of_file>.tar

To Use it

podman load -i <tar_file_path>

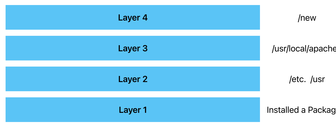
Containerfile

Containerfile ==== Dockerfile

A Containerfile lists a set of instructions that the container runtime can use to build an image

Each instructions causes a change, that results in a new image layer

Image Layers are stacked together to form a container image.



We create a Container Image using a Previously existing Container Image (Base Image)

Base Image will determine your Linux Distribution and its Components

- Package Manager
- Init System
- File System
- Preinstalled dependencies

INSTRUCTIONS

FROM
Set which base image to use, Takes base image name as argument.
This instruction is mandatory and the very first Instruction of your Containerfile.

WORKDIR
Sets the current working directory within the Container.
Instructions that follow after the WORKDIR run at the defined location.

MAINTAINER
The Person or Organisation who owns or who as developed or who maintains this image.

COPY
Copy file from host machine to inside the container.

ADD
Copy a file from a given URL inside your Container
Unpack tar archives in the destination mentioned inside the container.

RUN
To Run/Execute any command.

USER
Changes the active User.

LABEL
Adds a key-value pair to the metadata.

EXPOSE
Add a port to the image metadata that the process within the container is actually bound to.

ENV
Define Environment Variable

ARG
Define Build time Variable

VOLUME
Defines where to store data outside the Container.
Mounting a Volume inside the Container.

ENTRYPOINT
Set the executable command to run when the container starts.

CMD
Provides the Default Arguments for the ENTRYPOINT.

Building a Containerfile into a Container Image

podman build -t image-httpd:v1 /root/my-image/

EXAMPLE:

```
FROM    docker.io/library/ubuntu
MAINTAINER      Deep Shah (deep.shah.7677@gmail.com)
LABEL   info This is Testing Image
RUN     apt-get update
RUN     apt-get install apache2 -y
ENV     TRAINING DO188
EXPOSE  80
COPY    websitedata /var/www/html/index.html
USER    root
WORKDIR /var/www/html
RUN     apt-get install vim -y
CMD     ["apache2ctl", "-D", "FOREGROUND"]
~
~
```

Troubleshooting

podman logs <container_name>

Common Container Issues:

Port Mapping

- podman port <container_name>
- Check with the Config file
- podman inspect <container_name>
nsenter -t <PID_of_Container> -n ss -pant

Network Connectivity

- podman inspect <container_name>
networksettings
- podman network connect <network_name> <container_name>

Name Resolution Issues

- podman network inspect <network_name>
"dns_enabled" : true
- cd /etc/containers/networks/<network_name>.json

Not Associating Proper Environment Variables

- podman logs <container_name>