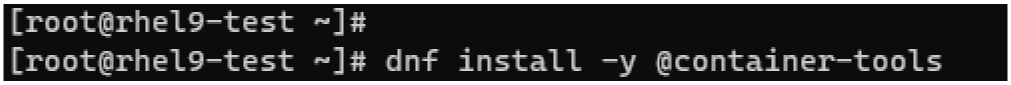
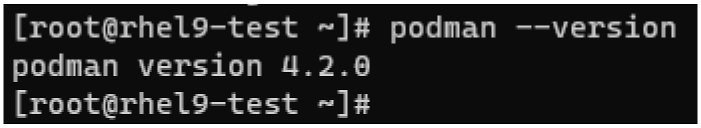
Container: Lecture 2

1. Install container tools-



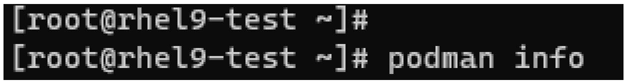
```
[root@rhel9-test ~]#
[root@rhel9-test ~]# dnf install -y @container-tools
```

2. Check podman version-



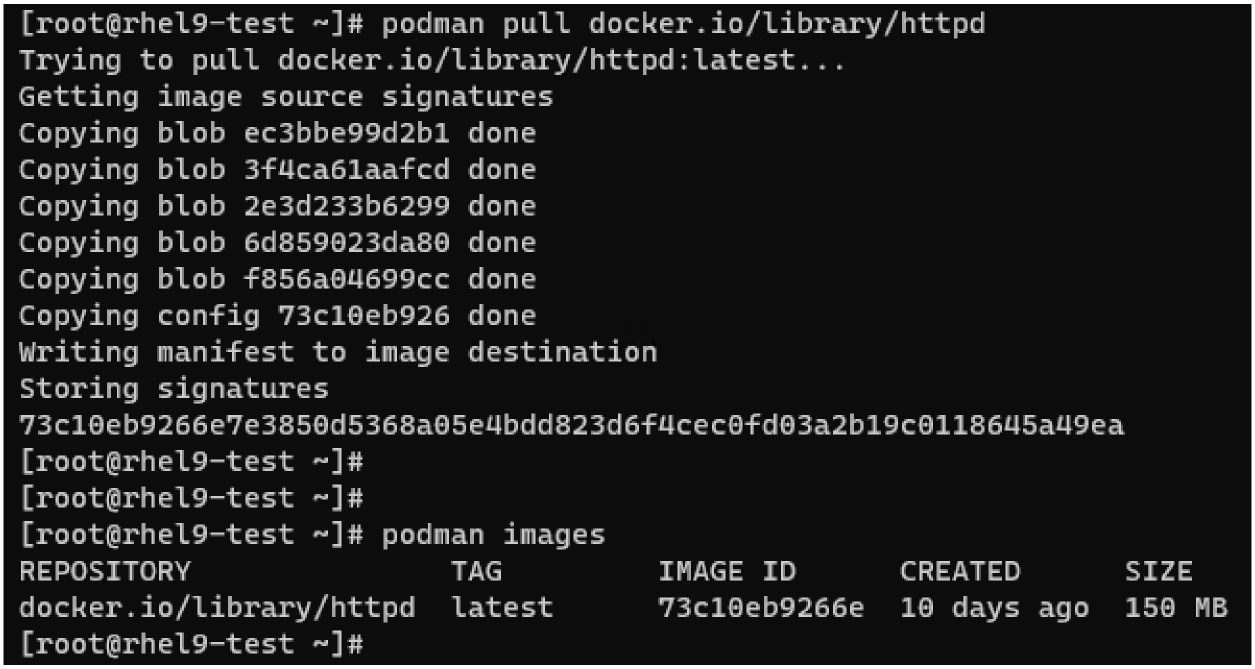
```
[root@rhel9-test ~]# podman --version
podman version 4.2.0
[root@rhel9-test ~]#
```

3. Check podman info for more detail-



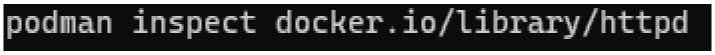
```
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman info
```

4. Now, pull docker http container image & verify it-



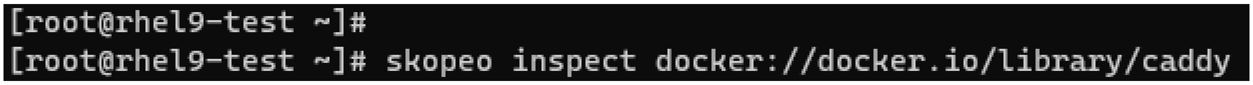
```
[root@rhel9-test ~]# podman pull docker.io/library/httpd
Trying to pull docker.io/library/httpd:latest...
Getting image source signatures
Copying blob ec3bbe99d2b1 done
Copying blob 3f4ca61aafcd done
Copying blob 2e3d233b6299 done
Copying blob 6d859023da80 done
Copying blob f856a04699cc done
Copying config 73c10eb926 done
Writing manifest to image destination
Storing signatures
73c10eb9266e7e3850d5368a05e4bdd823d6f4cec0fd03a2b19c0118645a49ea
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY               TAG         IMAGE ID      CREATED       SIZE
docker.io/library/httpd  latest      73c10eb9266e  10 days ago   150 MB
[root@rhel9-test ~]#
```

5. We can inspect this downloaded container image-



```
podman inspect docker.io/library/httpd
```

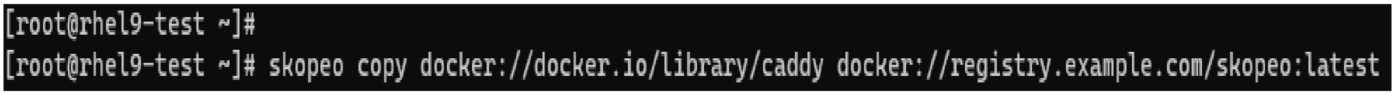
6. If we want to inspect a container image without downloading or pull, use below skopeo command-



```
[root@rhel9-test ~]#
[root@rhel9-test ~]# skopeo inspect docker://docker.io/library/caddy
```

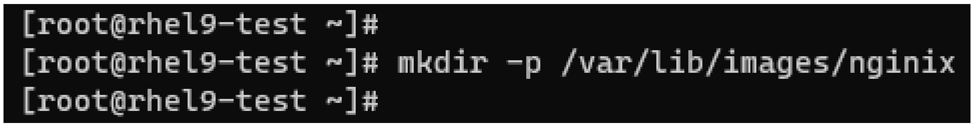
Note: We must have skopeo package installed in our machine. If not, install it using 1st command.

7. If we want to copy one container image to another, we use skopeo for this-



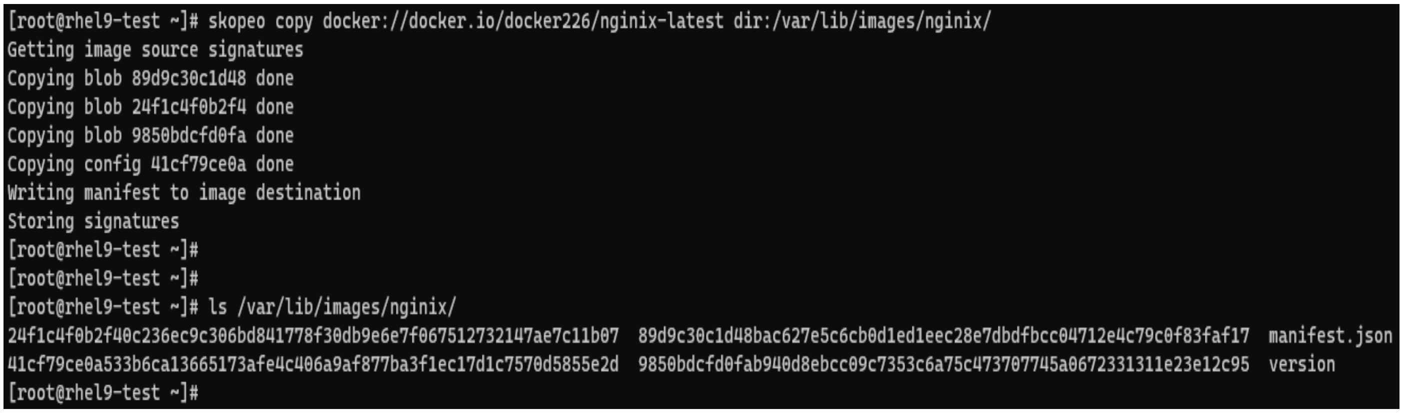
```
[root@rhel9-test ~]#
[root@rhel9-test ~]# skopeo copy docker://docker.io/library/caddy docker://registry.example.com/skopeo:latest
```

8. We want to pull container image at the location other than the default one. First make one directory as shown-



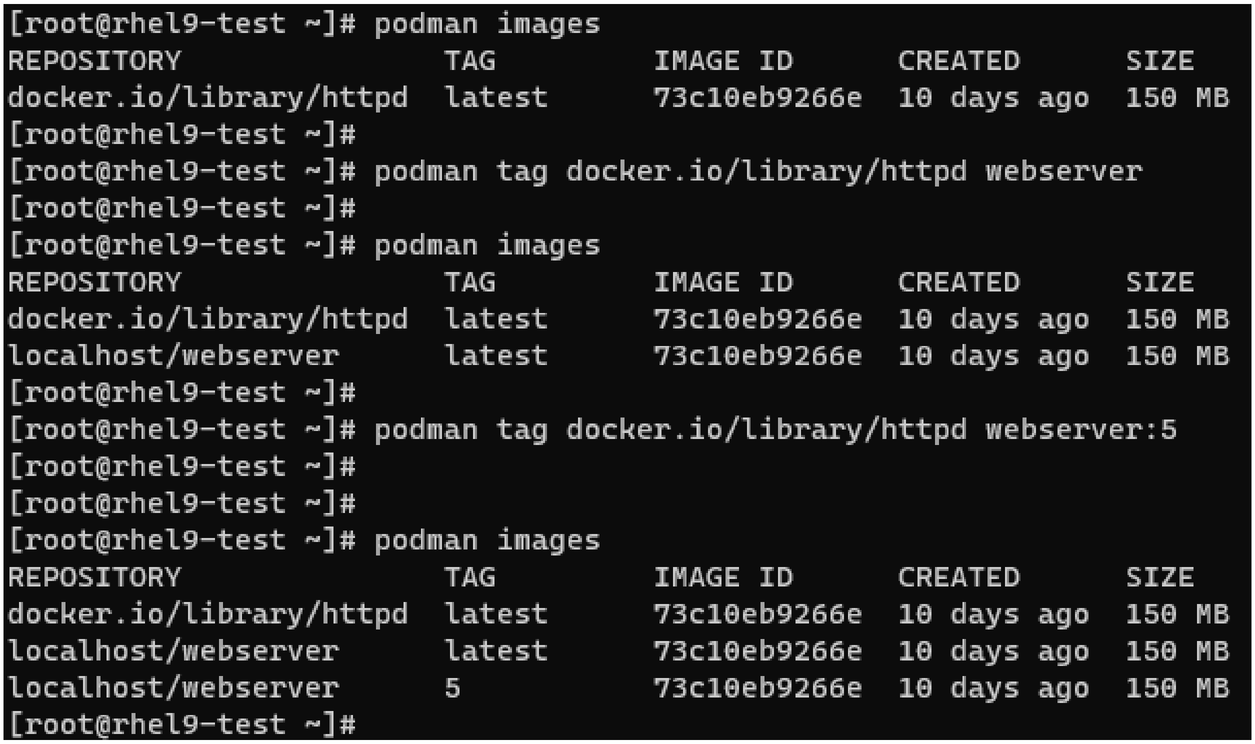
```
[root@rhel9-test ~]#
[root@rhel9-test ~]# mkdir -p /var/lib/images/nginix
[root@rhel9-test ~]#
```

9. Pull container image using this newly created directory & verify the same-



```
[root@rhel9-test ~]# skopeo copy docker://docker.io/docker226/nginix-latest dir:/var/lib/images/nginix/
Getting image source signatures
Copying blob 89d9c30c1d48 done
Copying blob 24f1c4f0b2f4 done
Copying blob 9850bdcfd0fa done
Copying config 41cf79ce0a done
Writing manifest to image destination
Storing signatures
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# ls /var/lib/images/nginix/
24f1c4f0b2f40c236ec9c306bd841778f30db9e6e7f067512732147ae7c11b07  89d9c30c1d48bac627e5c6cb0d1ed1eec28e7dbdfbcc04712e4c79c0f83faf17  manifest.json
41cf79ce0a533b6ca13665173afe4c406a9af877ba3f1ec17d1c7570d5855e2d  9850bdcfd0fab940d8ebcc09c7353c6a75c473707745a0672331311e23e12c95  version
[root@rhel9-test ~]#
```

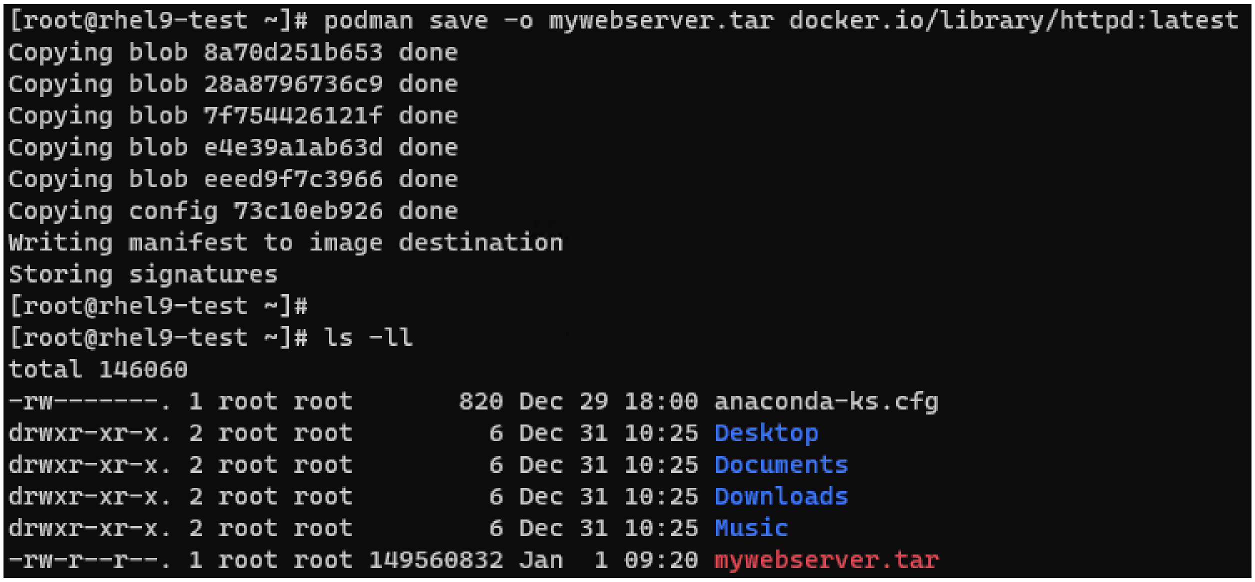
10. If we want to give custom tag to pulled container image, use below snap-



```
[root@rhel9-test ~]# podman images
REPOSITORY                TAG         IMAGE ID      CREATED       SIZE
docker.io/library/httpd  latest      73c10eb9266e  10 days ago   150 MB
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman tag docker.io/library/httpd webserver
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY                TAG         IMAGE ID      CREATED       SIZE
docker.io/library/httpd  latest      73c10eb9266e  10 days ago   150 MB
localhost/webserver      latest      73c10eb9266e  10 days ago   150 MB
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman tag docker.io/library/httpd webserver:5
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY                TAG         IMAGE ID      CREATED       SIZE
docker.io/library/httpd  latest      73c10eb9266e  10 days ago   150 MB
localhost/webserver      latest      73c10eb9266e  10 days ago   150 MB
localhost/webserver      5           73c10eb9266e  10 days ago   150 MB
[root@rhel9-test ~]#
```

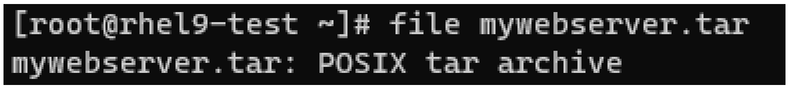
Note: Custom tag just crate a copy of original container image with custom name.

11. If we want to create an archive of container image, use command shown below-



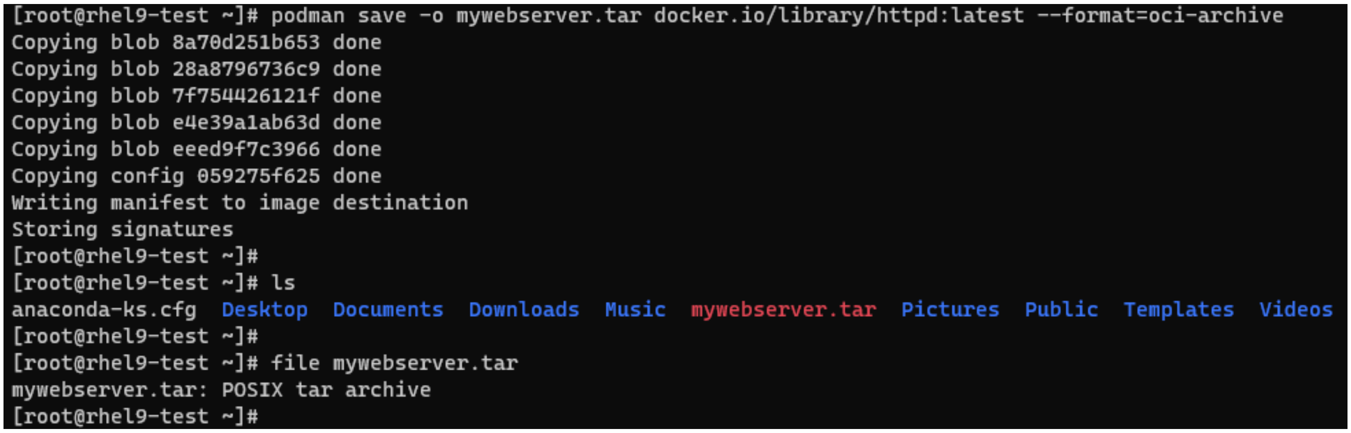
```
[root@rhel9-test ~]# podman save -o mywebserver.tar docker.io/library/httpd:latest
Copying blob 8a70d251b653 done
Copying blob 28a8796736c9 done
Copying blob 7f754426121f done
Copying blob e4e39a1ab63d done
Copying blob eeed9f7c3966 done
Copying config 73c10eb926 done
Writing manifest to image destination
Storing signatures
[root@rhel9-test ~]#
[root@rhel9-test ~]# ls -ll
total 146060
-rw-------. 1 root root       820 Dec 29 18:00 anaconda-ks.cfg
drwxr-xr-x. 2 root root         6 Dec 31 10:25 Desktop
drwxr-xr-x. 2 root root         6 Dec 31 10:25 Documents
drwxr-xr-x. 2 root root         6 Dec 31 10:25 Downloads
drwxr-xr-x. 2 root root         6 Dec 31 10:25 Music
-rw-r--r--. 1 root root 149560832 Jan  1 09:20 mywebserver.tar
```

12. Verify its file type-



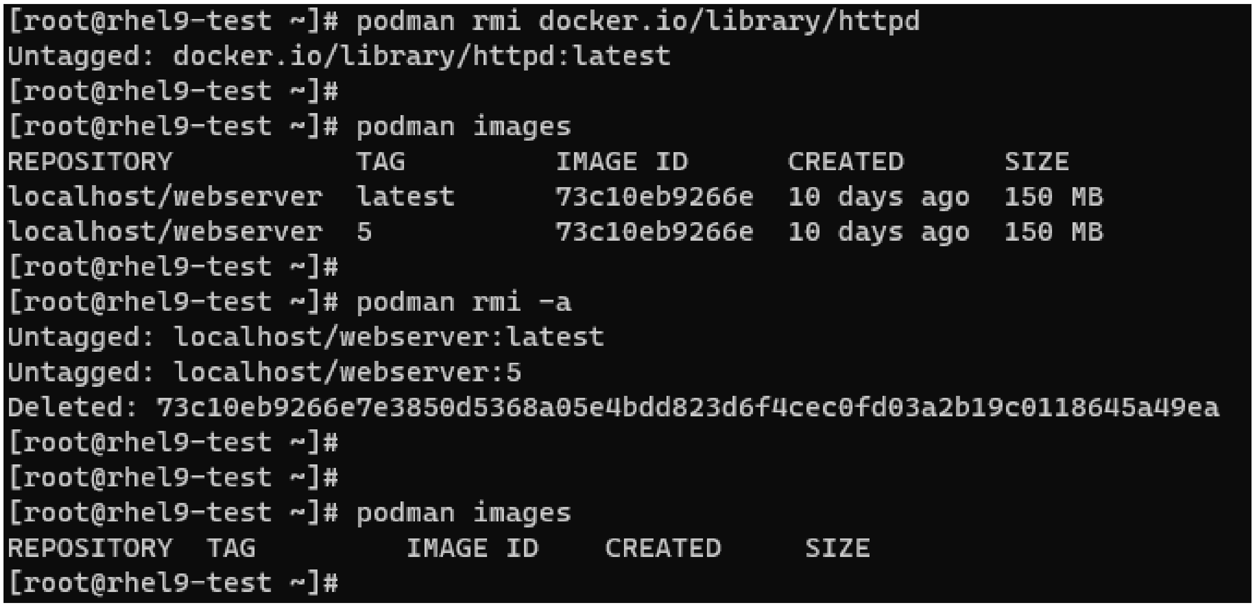
```
[root@rhel9-test ~]# file mywebserver.tar
mywebserver.tar: POSIX tar archive
```

13. If we want to archive a container image in OCI archive format, use below command-



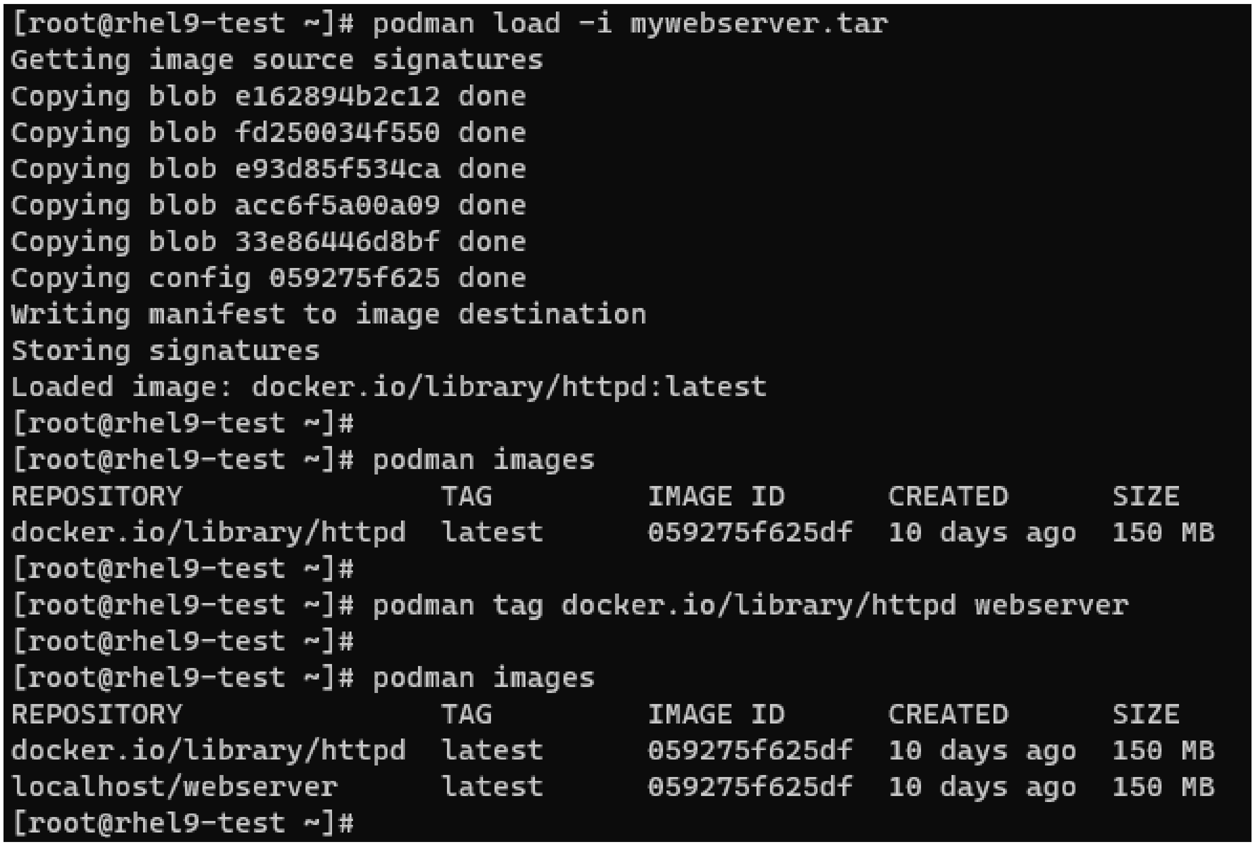
```
[root@rhel9-test ~]# podman save -o mywebserver.tar docker.io/library/httpd:latest --format=oci-archive
Copying blob 8a70d251b653 done
Copying blob 28a8796736c9 done
Copying blob 7f754426121f done
Copying blob e4e39a1ab63d done
Copying blob eeed9f7c3966 done
Copying config 059275f625 done
Writing manifest to image destination
Storing signatures
[root@rhel9-test ~]#
[root@rhel9-test ~]# ls
anaconda-ks.cfg  Desktop  Documents  Downloads  Music  mywebserver.tar  Pictures  Public  Templates  Videos
[root@rhel9-test ~]#
[root@rhel9-test ~]# file mywebserver.tar
mywebserver.tar: POSIX tar archive
[root@rhel9-test ~]#
```

14. We can remove a single container image or all container image at once. Removing all to create a container image using archive-



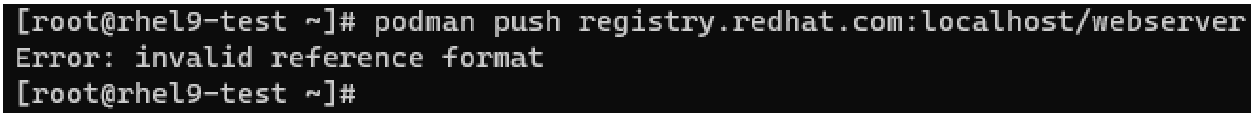
```
[root@rhel9-test ~]# podman rmi docker.io/library/httpd
Untagged: docker.io/library/httpd:latest
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY           TAG         IMAGE ID      CREATED      SIZE
localhost/webserver  latest      73c10eb9266e  10 days ago  150 MB
localhost/webserver  5           73c10eb9266e  10 days ago  150 MB
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman rmi -a
Untagged: localhost/webserver:latest
Untagged: localhost/webserver:5
Deleted: 73c10eb9266e7e3850d5368a05e4bdd823d6f4cec0fd03a2b19c0118645a49ea
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
[root@rhel9-test ~]#
```

15. Now load the container image from its archive created earlier & give it a tag-



```
[root@rhel9-test ~]# podman load -i mywebserver.tar
Getting image source signatures
Copying blob e162894b2c12 done
Copying blob fd250034f550 done
Copying blob e93d85f534ca done
Copying blob acc6f5a00a09 done
Copying blob 33e86446d8bf done
Copying config 059275f625 done
Writing manifest to image destination
Storing signatures
Loaded image: docker.io/library/httpd:latest
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY               TAG         IMAGE ID      CREATED      SIZE
docker.io/library/httpd  latest      059275f625df  10 days ago  150 MB
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman tag docker.io/library/httpd webserver
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY               TAG         IMAGE ID      CREATED      SIZE
docker.io/library/httpd  latest      059275f625df  10 days ago  150 MB
localhost/webserver      latest      059275f625df  10 days ago  150 MB
[root@rhel9-test ~]#
```

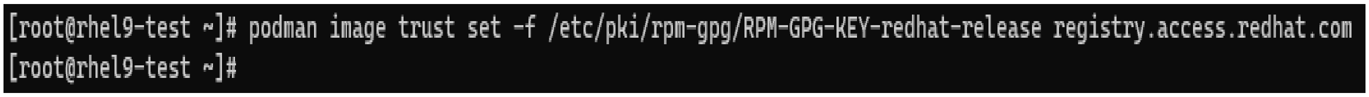
16. Now, we can push our custom container image to Redhat registry if we have proper subscription-



```
[root@rhel9-test ~]# podman push registry.redhat.com:localhost/webserver
Error: invalid reference format
[root@rhel9-test ~]#
```

Note: Here we are getting error. We need to login first for this & should have required subscription to push container image. It can't be done just using developer subscription. Login process was already explained in lecture 1.

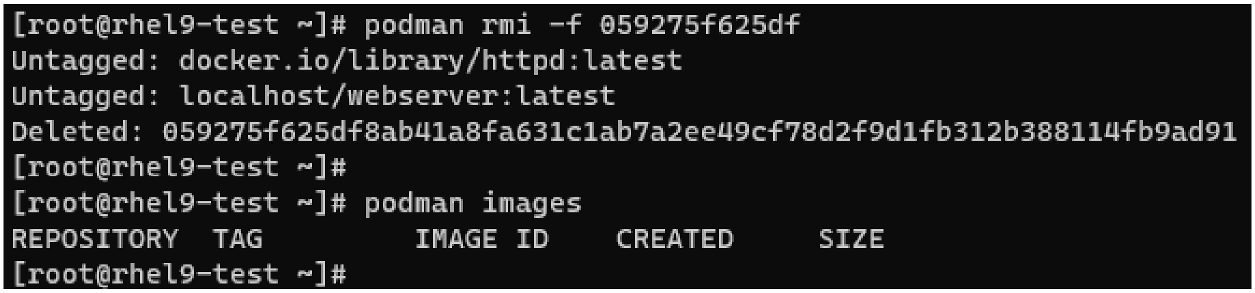
17. We can set a trust for the image pulled from different registry using GPG key as shown-



```
[root@rhel9-test ~]# podman image trust set -f /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release registry.access.redhat.com
[root@rhel9-test ~]#
```

Note: Caution before using this. It may not allow you to download container image as it will check the signature.

18. If we are done with Lab, remove the images as shown-



```
[root@rhel9-test ~]# podman rmi -f 059275f625df
Untagged: docker.io/library/httpd:latest
Untagged: localhost/webserver:latest
Deleted: 059275f625df8ab41a8fa631c1ab7a2ee49cf78d2f9d1fb312b388114fb9ad91
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
[root@rhel9-test ~]#
```

This is it for Lecture 2!!!