# Container: Lecture 6

1. Install container packages-

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

2. Check whether required packages is installed or not-

```
[root@rhel9-test ~]# yum module list | grep container
```

## Root Full Container:

3. Pull 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  11 days ago  150 MB
[root@rhel9-test ~]#
```

4. Run it in background & use port 8080 for this web server & verify the same using this port-

```
[root@rhel9-test ~]# podman run -dit --name=mywebpage -p 8080:80 73c10eb9266e
25906e061c33f0ac46795ae9b7fba965fe45963d55e853b04d7244b706bc80ba
[root@rhel9-test ~]#
[root@rhel9-test ~]# curl 192.168.111.128:8080
<html><body><h1>It works!</h1></body></html>
[root@rhel9-test ~]#
```

5. Check current status of container (Whether running or not). If we want to check logs, use shown command-

```
[root@rhel9-test ~]# podman ps
CONTAINER ID  IMAGE                           COMMAND          CREATED        STATUS            PORTS                   NAMES
25906e061c33  docker.io/library/httpd:latest  httpd-foreground  3 minutes ago  Up 3 minutes ago  0.0.0.0:8080->80/tcp  mywebpage
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman logs 25906e061c33
```

Note: It will show all the logs generated for this container.

6. To check what are the process running for this container image, use command shown-

```
[root@rhel9-test ~]# podman top -l
USER        PID         PPID        %CPU        ELAPSED          TTY         TIME        COMMAND
root        1           0           0.000       4m54.203143698s  pts/0       0s          httpd -DFOREGROUND
www-data    3           1           0.000       4m54.203259096s  pts/0       0s          httpd -DFOREGROUND
www-data    4           1           0.000       4m54.203299308s  pts/0       0s          httpd -DFOREGROUND
www-data    5           1           0.000       4m54.203336447s  pts/0       0s          httpd -DFOREGROUND
[root@rhel9-test ~]#
```

7. Now, go inside this container image using its bash shell & will verify the index file content & then exit out from here-

```
[root@rhel9-test ~]# podman exec -it mywebpage /bin/bash
root@25906e061c33:/usr/local/apache2#
root@25906e061c33:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules
root@25906e061c33:/usr/local/apache2#
root@25906e061c33:/usr/local/apache2# cd htdocs/
root@25906e061c33:/usr/local/apache2/htdocs# ls
index.html
root@25906e061c33:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>It works!</h1></body></html>
root@25906e061c33:/usr/local/apache2/htdocs#
```

```
root@25906e061c33:/usr/local/apache2# exit
exit
[root@rhel9-test ~]#
```

8. We want to use custom webpage (Available on host machine) using persistent storage. For this, follow below steps-

```
[root@rhel9-test ~]# mkdir /html
[root@rhel9-test ~]# echo "Welcome to Cricbuzz World!" > /html/index.html
[root@rhel9-test ~]#
[root@rhel9-test ~]# cat /html/index.html
Welcome to Cricbuzz World!
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# chmod -R 77 /html
[root@rhel9-test ~]#
[root@rhel9-test ~]# ls -lZ /html
total 4
----rwxrwx. 1 root root unconfined_u:object_r:default_t:s0 27 Jan  2 10:04 index.html
[root@rhel9-test ~]#
[root@rhel9-test ~]# semanage fcontext -a -t httpd_sys_content_t '/html(/.*)?'
[root@rhel9-test ~]# restorecon -vRF /html
Relabeled /html from unconfined_u:object_r:default_t:s0 to system_u:object_r:httpd_sys_content_t:s0
Relabeled /html/index.html from unconfined_u:object_r:default_t:s0 to system_u:object_r:httpd_sys_content_t:s0
[root@rhel9-test ~]#
[root@rhel9-test ~]# ls -lZ /html
total 4
----rwxrwx. 1 root root system_u:object_r:httpd_sys_content_t:s0 27 Jan  2 10:04 index.html
[root@rhel9-test ~]#
```

Note: Here we have to set 777 permission to parent directory & subdirectories. Change selinux context.

9. Stop all the running container & remove it-

```
[root@rhel9-test ~]# podman stop mywebpage
mywebpage
[root@rhel9-test ~]#
```

```
[root@rhel9-test ~]# podman rm -a
25906e061c33f0ac46795ae9b7fba965fe45963d55e853b04d7244b706bc80ba
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9-test ~]# podman ps -a
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9-test ~]#
```

10. Check the available container image-

```
[root@rhel9-test ~]# podman images
REPOSITORY                TAG         IMAGE ID      CREATED      SIZE
docker.io/library/httpd   latest      73c10eb9266e  11 days ago  150 MB
[root@rhel9-test ~]#
```

11. We will run container in background & use the path of index.html of host machine to mount it to container. Verify the same-

```
[root@rhel9-test ~]# podman run -dit --name=mywebpage -p 8080:80 -v /html/:/usr/local/apache2/htdocs 73c10eb9266e
ee9c73e7e80abc489b368758621f23247ad27b590c50709c5f4cd4e221622a45
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps
CONTAINER ID  IMAGE                           COMMAND           CREATED         STATUS             PORTS                 NAMES
ee9c73e7e80a  docker.io/library/httpd:latest  httpd-foreground  18 seconds ago  Up 12 seconds ago  0.0.0.0:8080->80/tcp  mywebpage
[root@rhel9-test ~]#
```

12. Verify it using default & new http port-

```
[root@rhel9-test ~]# curl 192.168.111.128
curl: (7) Failed to connect to 192.168.111.128 port 80: Connection refused
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# curl 192.168.111.128:8080
Welcome to Cricbuzz World!
[root@rhel9-test ~]#
```

13. Now we will stop the container & verify it-

```
[root@rhel9-test ~]# podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps -a
CONTAINER ID  IMAGE                           COMMAND           CREATED         STATUS                     PORTS                 NAMES
ee9c73e7e80a  docker.io/library/httpd:latest  httpd-foreground  17 minutes ago  Exited (0) 7 minutes ago  0.0.0.0:8080->80/tcp  mywebpage
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman rm -a
ee9c73e7e80abc489b368758621f23247ad27b590c50709c5f4cd4e221622a45
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps -a
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9-test ~]#
```

This testing is done.

14. Next, pull MySQL container image & verify the same-

```
[root@rhel9-test ~]# podman pull docker.io/library/mysql
Trying to pull docker.io/library/mysql:latest...
Getting image source signatures
Copying blob 054e8fde88d0 done
Copying blob 36bb5e56d458 done
Copying blob 3d2f9b664bd3 done
Copying blob df6519f81c26 done
Copying blob 0ed027b72ddc done
Copying blob 029615974 7f1 done
Copying blob f2b494c50c7f done
Copying blob 132bc0d471b8 done
Copying blob 135ec7033a05 done
Copying blob 5961f0272472 done
Copying blob 75b5f7a3d3a4 done
Copying config 7484689f29 done
Writing manifest to image destination
Storing signatures
7484689f290f1defe06b65befc54cb6ad91a667cf0af59a265ffe76c46bd0478
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY                TAG         IMAGE ID      CREATED      SIZE
docker.io/library/httpd   latest      73c10eb9266e  11 days ago  150 MB
docker.io/library/mysql   latest      7484689f290f  3 weeks ago  550 MB
[root@rhel9-test ~]#
```

15. Run it in background using default port & set root password. Verify it & then login into container using shell-

```
[root@rhel9-test ~]# podman run -d --name=test-mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD='Jerry@4321' 7484689f290f
5af5c6a489c8ddd45051a212d2cc0de2e12ee87453785f3f5b7290e5724ccc27
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps
CONTAINER ID  IMAGE                           COMMAND     CREATED         STATUS             PORTS                   NAMES
5af5c6a489c8  docker.io/library/mysql:latest  mysqld      10 seconds ago  Up 10 seconds ago  0.0.0.0:3306->3306/tcp  test-mysql
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman exec -it 5af5c6a489c8 /bin/bash
bash-4.4#
```

16. Now, login into data base using set credential (In last step)-

```
[root@rhel9-test ~]# podman exec -it 5af5c6a489c8 /bin/bash
bash-4.4#
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

17. Show existing database-

```
mysql> show DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

18. Create a new DB & verify it. Check warnings as well & then exit from container-

```
mysql> create DATABASE cricbuzz;
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> show DATABASES;
+--------------------+
| Database           |
+--------------------+
| cricbuzz           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql>
mysql> show WARNINGS;
Empty set (0.00 sec)

mysql>
mysql> exit
Bye
bash-4.4# exit
exit
[root@rhel9-test ~]#
```

19. Next, stop the container & verify it-

```
[root@rhel9-test ~]# podman stop -a
5af5c6a489c8ddd45051a212d2cc0de2e12ee87453785f3f5b7290e5724ccc27
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman rm -a
5af5c6a489c8ddd45051a212d2cc0de2e12ee87453785f3f5b7290e5724ccc27
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman ps -a
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9-test ~]#
[root@rhel9-test ~]#
```

## Root Less Container:

20. Now we will do container Lab in standard user. Login with it & verify the existing container images if any. For the root less lab, ignore the warnings & focus on blue arrow-

```
[root@rhel9-test ~]# su - john
[john@rhel9-test ~]$
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman images
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
WARN[0000] Failed to add pause process to systemd sandbox cgroup: exec: "dbus-launch": executable file not found in $PATH
[john@rhel9-test ~]$
```

21. Search for a container image-

```
[john@rhel9-test ~]$ podman search myubi
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
NAME                                DESCRIPTION
docker.io/myubi/votingapp_result-app
docker.io/myubi/votingapp_voting-app
docker.io/myage/myubi
docker.io/17dockerid27/myubimage
docker.io/marianorean/myubi
docker.io/myubifoundation/blockscout
docker.io/hanlinkyaw/myubi8
[john@rhel9-test ~]$
```

22. Pull the container image-

```
[john@rhel9-test ~]$ podman pull docker.io/marianorean/myubi
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
Trying to pull docker.io/marianorean/myubi:latest...
Getting image source signatures
Copying blob 556ecd0ceec5 done
Copying blob e96e057aae67 done
Copying config cddb5af51d done
Writing manifest to image destination
Storing signatures
cddb5af51ddfb9975cc35844da1d4970fe28305594637867588f3c930dff9b37
[john@rhel9-test ~]$
```

23. Verify it & inspect if require more detail about this container image-

```
[john@rhel9-test ~]$ podman images
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
REPOSITORY                  TAG         IMAGE ID      CREATED      SIZE
docker.io/marianorean/myubi  latest      cddb5af51ddf  3 weeks ago  182 MB
[john@rhel9-test ~]$
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman inspect cddb5af51ddf
```

24. To get information about podman, use command as shown-

```
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman info
```

25. If we want to go to storage location of containers, use path shown after running previous command-

```
[john@rhel9-test ~]$ cd /home/john/.local/share/containers/storage
[john@rhel9-test storage]$
[john@rhel9-test storage]$ ls
defaultNetworkBackend  libpod  mounts  networks  overlay  overlay-containers  overlay-images  overlay-layers  storage.lock  tmp  userns.lock
[john@rhel9-test storage]$
```

26. Check current container images-

```
[john@rhel9-test ~]$ podman images
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
REPOSITORY                   TAG         IMAGE ID      CREATED      SIZE
docker.io/marianorean/myubi  latest      cddb5af51ddf  3 weeks ago  182 MB
[john@rhel9-test ~]$
```

27. Run this container image in background & verify-

```
[john@rhel9-test ~]$ podman run -dit --name=myubi cddb5af51ddf
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
d29f7c0b4e5f24fc11f80bbfe74b5dda585586c0167d7d8b11a3e7842069160e
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman ps
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
CONTAINER ID  IMAGE                                COMMAND     CREATED        STATUS            PORTS       NAMES
d29f7c0b4e5f  docker.io/marianorean/myubi:latest  /bin/bash   8 seconds ago  Up 8 seconds ago              myubi
[john@rhel9-test ~]$
```

28. Go inside container image using interactive shell-

```
[john@rhel9-test ~]$ podman exec -it d29f7c0b4e5f /bin/bash
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
root@d29f7c0b4e5f:/#
```

29. Run few commands in this container to get some detail about it-

```
root@d29f7c0b4e5f:/# cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
root@d29f7c0b4e5f:/#
root@d29f7c0b4e5f:/#
root@d29f7c0b4e5f:/#
root@d29f7c0b4e5f:/# uname -a
Linux d29f7c0b4e5f 5.14.0-162.6.1.el9_1.x86_64 #1 SMP PREEMPT_DYNAMIC Fri Sep 30 07:36:03 EDT 2022 x86_64 x86_64 x86_64 GNU/Linux
root@d29f7c0b4e5f:/#
root@d29f7c0b4e5f:/# uname -r
5.14.0-162.6.1.el9_1.x86_64
root@d29f7c0b4e5f:/#
```

30. Now exit it & check the status. Check if we can mount this container in standard user or not-

```
root@d29f7c0b4e5f:/# exit
exit
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman ps
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
CONTAINER ID  IMAGE                              COMMAND     CREATED        STATUS            PORTS       NAMES
d29f7c0b4e5f  docker.io/marianorean/myubi:latest  /bin/bash   2 minutes ago  Up 2 minutes ago              myubi
[john@rhel9-test ~]$
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman mount d29f7c0b4e5f
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
Error: cannot run command "podman mount" in rootless mode, must execute `podman unshare` first
[john@rhel9-test ~]$
```

## Limitations of root less container:

(i). We are getting error while trying to mount it in root less mode. This is the limitation of root less mode.

(ii). Also we can't assign any port before 1024 in root less mode. We will verify it in next few snapshots.

31. Pull httpd image & verify it-

```
[john@rhel9-test ~]$ podman pull docker.io/library/httpd
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
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
[john@rhel9-test ~]$
```

```
[john@rhel9-test ~]$ podman images
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
REPOSITORY                  TAG         IMAGE ID      CREATED      SIZE
docker.io/library/httpd     latest      73c10eb9266e  11 days ago  150 MB
docker.io/marianorean/myubi  latest      cddb5af51ddf  3 weeks ago  182 MB
[john@rhel9-test ~]$
```

32. Now run it in background & try to use default http port (Which is less than 1024)-

```
[john@rhel9-test ~]$ podman run -dit --name=myhttpd -p 80:80 docker.io/library/httpd
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
Error: rootlessport cannot expose privileged port 80, you can add 'net.ipv4.ip_unprivileged_port_start=80' to /etc/sysctl.conf (currently 1024), or
 choose a larger port number (>= 1024): listen tcp 0.0.0.0:80: bind: permission denied
[john@rhel9-test ~]$
```

Here, it is not allowing us to use port 80 in root less mode.

33. Remove this container-

```
[john@rhel9-test ~]$ podman rm myhttpd
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
a8d8d7f553086117b7fbd72d9bf778ec7f2e4236bba4ed1c97200d952b9d3a55
[john@rhel9-test ~]$
```

34. Run it in background & this time with port no. greater than 1024 & verify it-

```
[john@rhel9-test ~]$ podman run -dit --name=myhttpd -p 8080:80 docker.io/library/httpd
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
b2ef38397aed793b23c6f19dcc293bc17807660fd9ee0deede1db72ca8a74a82
[john@rhel9-test ~]$
[john@rhel9-test ~]$ curl 192.168.111.128:8080
<html><body><h1>It works!</h1></body></html>
[john@rhel9-test ~]$
```

This time we are successful in running this container using custom port & able to see webpage output.

35. Stop & remove all running images-

```
[john@rhel9-test ~]$ podman rm -a -f
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0010] StopSignal SIGTERM failed to stop container myubi in 10 seconds, resorting to SIGKILL
b2ef38397aed793b23c6f19dcc293bc17807660fd9ee0deede1db72ca8a74a82
d29f7c0b4e5f24fc11f80bbfe74b5dda585586c0167d7d8b11a3e7842069160e
[john@rhel9-test ~]$
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman ps -a
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[john@rhel9-test ~]$
```

```
[john@rhel9-test ~]$ podman rmi -a
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
Untagged: docker.io/marianorean/myubi:latest
Untagged: docker.io/library/httpd:latest
Deleted: cddb5af51ddfb9975cc35844da1d4970fe28305594637867588f3c930dff9b37
Deleted: 73c10eb9266e7e3850d5368a05e4bdd823d6f4cec0fd03a2b19c0118645a49ea
[john@rhel9-test ~]$
[john@rhel9-test ~]$
[john@rhel9-test ~]$ podman images
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
WARN[0000] The cgroupv2 manager is set to systemd but there is no systemd user session available
WARN[0000] For using systemd, you may need to login using an user session
WARN[0000] Alternatively, you can enable lingering with: `loginctl enable-linger 1000` (possibly as root)
WARN[0000] Falling back to --cgroup-manager=cgroupfs
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
[john@rhel9-test ~]$
```

36. Exit from standard user-

```
[john@rhel9-test ~]$ exit
logout
[root@rhel9-test ~]#
```

37. Remove container images from root user login as well which we pulled at start-

```
[root@rhel9-test ~]# podman rmi -a
Untagged: docker.io/library/httpd:latest
Untagged: docker.io/library/mysql:latest
Deleted: 73c10eb9266e7e3850d5368a05e4bdd823d6f4cec0fd03a2b19c0118645a49ea
Deleted: 7484689f290f1defe06b65befc54cb6ad91a667cf0af59a265ffe76c46bd0478
[root@rhel9-test ~]#
[root@rhel9-test ~]#
[root@rhel9-test ~]# podman images
REPOSITORY  TAG         IMAGE ID    CREATED     SIZE
[root@rhel9-test ~]#
```

This is it about Lecture 5!!!