Exercise 5

Creating and Running Containers

Q1.
Create a new container.
Run “ls” in this running container in two ways.
(i) Without entering the container
(ii) Entering the container and executing the command in it.

Q2.
Using the previous container.
Create a file in pwd named “training.txt”
Enter the line “TRAINING IN PROGRESS” inside the file created.
Copy this file inside the running container.
Enter the container to verify the copy was successful.

Q3.
Create a container using “MySQL” image
Find a file at “/docker-entrypoint-initdb.d” inside the container.
Copy that file from the container to the “/root/” directory on the Podman host.
Verify by visiting the location.

Q4.
Launch a new container “example”.
Use an Nginx Image and connect it to port 8080 of host machine, if already assigned then delete the corresponding previous container.
Make sure it is connected to a network called “example” (Create it).
Create a txt file called “content” and add a line “WEBSERVER IS WORKING”.
Copy this file inside the running container on DocumentRoot.
Simply type in “curl localhost:8080” and it should show the output as:

WEBSERVER IS WORKING

Q5.
Check all the currently podman bound ports.

Solutions

Q1.

```
[root@localhost ~]# podman run -d --name webserver -p 8080:80 httpd
d242962ebf1783f995351a694a03e99ebe6f57752bc37c1462b3ce218aed2917
[root@localhost ~]# podman exec webserver ls
bin
build
cgi-bin
conf
error
htdocs
icons
include
logs
modules
[root@localhost ~]# podman exec -it webserver bash
root@d242962ebf17:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules
root@d242962ebf17:/usr/local/apache2#
```

Q2.

```
[root@localhost ~]# pwd
/root
[root@localhost ~]# touch training.txt
[root@localhost ~]# echo "TRAINING IN PROGRESS" > training.txt
[root@localhost ~]# podman ps
CONTAINER ID  IMAGE                           COMMAND           CREATED        STATUS            PORTS
      NAMES
d242962ebf17  docker.io/library/httpd:latest  httpd-foreground  2 minutes ago  Up 2 minutes ago  0.0.0.0:8080->8
/tcp  webserver
[root@localhost ~]# podman cp ./training.txt webserver:/usr/local/apache2
[root@localhost ~]# podman exec -it webserver bash
root@d242962ebf17:/usr/local/apache2# pwd
/usr/local/apache2
root@d242962ebf17:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules  training.txt
root@d242962ebf17:/usr/local/apache2# cat training.txt
TRAINING IN PROGRESS
root@d242962ebf17:/usr/local/apache2#
root@d242962ebf17:/usr/local/apache2# exit
exit
[root@localhost ~]#
```

Q3.

```
[root@localhost ~]# podman run -d --name con1 -e MYSQL_ROOT_PASSWORD=redhat  docker.io/mysql
Trying to pull docker.io/library/mysql:latest...
Getting image source signatures
Copying blob 6effcc6561c9 done
Copying blob 06e0c37837cf done
Copying blob c6f5d3670db7 done
Copying blob d5c567b29c3e done
Copying blob 5262579e8e45 done
Copying blob 741b767e25b7 done
Copying blob 1e1493d45d9c done
Copying blob 7101609fa7d9 done
Copying blob 432a1261dc2a done
Copying blob 865a24d6d1f2 done
Copying config b2013ac991 done
Writing manifest to image destination
Storing signatures
bbdec98af68c30a9d8a3129fbab67c6bd071f62560fe8f0b9b7e8016d7e246b4
[root@localhost ~]# podman ps -a
CONTAINER ID  IMAGE                           COMMAND     CREATED         STATUS            PORTS       NAMES
bbdec98af68c  docker.io/library/mysql:latest  mysqld      10 seconds ago  Up 9 seconds ago              con1
```

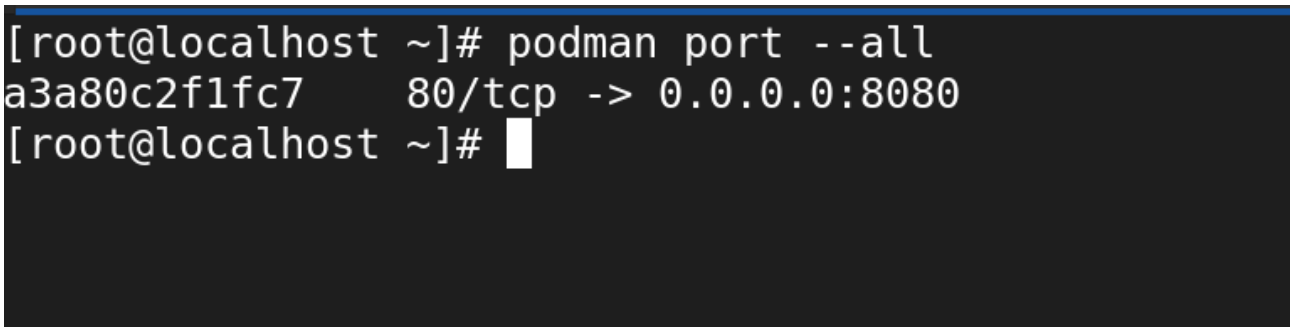
```
[root@localhost ~]# podman cp con1:/docker-entrypoint-initdb.d/ /root/
[root@localhost ~]# ls
anaconda-ks.cfg  content  docker-entrypoint-initdb.d  training.txt
[root@localhost ~]#
```

Q4.

```
[root@localhost ~]# podman rm -f webserver
fe466755fc03fce522fc279ed4efb0b78008125926a75d311282ad20fcd9935a
[root@localhost ~]# podman network create example
example
[root@localhost ~]# podman run -d --name example --network example -p 8080:80 nginx
a3a80c2f1fc7e264eb28a3b4494f5b519e20bab75e313a7cbfcc5f36f07fe35c
[root@localhost ~]# echo "WEBSERVER IS WORKING" > content
[root@localhost ~]# cat content
WEBSERVER IS WORKING
```

```
[root@localhost ~]# podman cp ./content example:/usr/share/nginx/html/index.html
[root@localhost ~]# curl localhost:8080
WEBSERVER IS WORKING
[root@localhost ~]#
```

Q5.


[root@localhost ~]# podman port --all
a3a80c2f1fc7   80/tcp -> 0.0.0.0:8080
[root@localhost ~]#