Step 1: Prepare the Environment

Ensure Podman is installed on your RHEL/CentOS system. Run the following commands to install Podman if it's not already installed:

$ sudo dnf update -y

$ sudo dnf install -y podman

Verify the installation:

podman --version

Step 2: Run a Web Server Container (Nginx Example)

Run an Nginx container and expose it on port 8080.

$ podman run -d --name web-server -p 8080:80 nginx

This command:

- Runs the Nginx container in detached mode (-d).
- Names the container "web-server."
- Maps port 8080 on the host to port 80 in the container (-p 8080:80).

Step 3: Run a Database Container (MySQL Example)

Run a MySQL container and expose it on port 3306.

$ podman run -d --name mariadb-server -e MARIADB_ROOT_PASSWORD=my-secret-pw -p 3306:3306 rhel8/mariadb-105

Explanation

- **-d**: Runs the container in detached mode (in the background).
- **--name mariadb-server**: Names the container "mariadb-server" for easy identification.
- **-e MARIADB_ROOT_PASSWORD=my-secret-pw**: Sets the root password for MariaDB.
- **-p 3306:3306**: Maps port 3306 on the host to port 3306 in the container, making it accessible.

Step 4: List Running Containers

List all currently running containers:

$ podman ps

This will display the container ID, names, status, ports, and other details of running containers.

Step 5: Access the Nginx Web Server

To access the Nginx server, open a web browser and navigate to:

$ http://localhost:8080

You should see the default Nginx welcome page if everything is configured correctly.

Step 6: Inspecting Container Networking

To inspect the network configuration of the Nginx container:

```
$ podman inspect web-server --format '{{ .NetworkSettings }}'
```

This command provides detailed information about the container's networking setup, including IP address, ports, and network mode.

Step 7: Stopping and Removing Containers

To stop and remove the running containers:

```
$ podman stop web-server db-server
```

```
$ podman rm web-server db-server
```

This will stop the Nginx and MySQL containers, then remove them from the system.

Step 8: Run Multiple Containers with Different Ports

Run additional containers with different ports to explore multi-container setups. For example, if you want to run another Nginx instance on port 9090:

```
$ podman run -d --name web-server-2 -p 9090:80 nginx
```

Now, you can access this second Nginx instance at http://localhost:9090.

Step 9: Explore Security Practices for Running Containers

Here are a few security practices when running containers with Podman:

1. **Use Non-Root User**: Avoid running containers as root. Use a non-root user by specifying --user flag:

   ```
   $ podman run -d --name secure-web-server -p 8081:80 --user 1001 nginx
   ```

2. **Limit Container Capabilities**: Limit the capabilities of the container by adding --cap-drop:

   ```
   $ podman run -d --name limited-web-server -p 8082:80 --cap-drop all nginx
   ```

3. **Enable SELinux**: Ensure SELinux policies are in place and enforce them:

   ```
   $ sudo setenforce 1
   ```

4. **Network Isolation**: Consider using custom networks for better isolation:

   ```
   $ podman network create my-network
   ```

   ```
   $ podman run -d --name isolated-web-server --network my-network -p 8083:80 nginx
   ```

```
[student@workstation ~]$ podman run -d --name web-server -p 8080:80 nginx

Resolving "nginx" using unqualified-search registries (/home/student/.config/containers/registries.conf)
Trying to pull registry.lab.example.com/nginx:latest...
Getting image source signatures
Copying blob 655316c160af done
Copying blob d121f8d1c412 done
Copying blob d15953c0e0f8 done
Copying blob ebd81fc8c071 done
Copying blob 2ee525c5c3cc done
Copying config 7e4d58f0e5 done
Writing manifest to image destination
Storing signatures
cd0f20e083ac100b7da3f1fcd9fc85716bd4275be74d61c1cedfb6ec66ab73d7
```

```
[student@workstation ~]$ podman login registry.lab.example.com
Username: admin
Password:
Login Succeeded!
```

Password: redhat321

```
[student@workstation ~]$ podman search registry.lab.example.com/
NAME                                        DESCRIPTION
registry.lab.example.com/rhel8/mariadb-103
registry.lab.example.com/rhel8/mariadb-105
registry.lab.example.com/rhel8/httpd-24
registry.lab.example.com/library/nginx
registry.lab.example.com/ubi7/ubi
registry.lab.example.com/ubi8/ubi
registry.lab.example.com/ubi9-beta/ubi
registry.lab.example.com/ubi8/python-38
registry.lab.example.com/ubi8/httpd-24
registry.lab.example.com/rhel8/php-74
```

```
[student@workstation ~]$ podman run -d --name mariadb-server -e MARIADB_ROOT_PASSWORD=my-secret-pw -p 3306:3306
rhel8/mariadb-105
Resolving "rhel8/mariadb-105" using unqualified-search registries (/home/student/.config/containers/registries.c
onf)
Trying to pull registry.lab.example.com/rhel8/mariadb-105:latest...
Getting image source signatures
Copying blob 6dfd6932feef done
Copying blob 3de00bb8554b done
Copying blob c530010fb61c done
Copying blob 6c155a0c493b done
Copying config 8ddd2fba74 done
Writing manifest to image destination
Storing signatures
474e2805ccdc0ebcfd59b8dfb1438ea6f8747bbc4ae41b13b3070d5ecea2977a
[student@workstation ~]$
```

```
[student@workstation ~]$ podman run -d --name isolated-web-server --network my-network -p 8083:80 nginx
352ac976ff3e63dc88ea997887827e253d8620a6fedfc900714d2f52ed677b1d
[student@workstation ~]$ podman ps
CONTAINER ID  IMAGE                                  COMMAND               CREATED        STATUS            PORT
S                 NAMES
3c984bc71109  registry.lab.example.com/nginx:latest  nginx -g daemon o...  6 minutes ago  Up 6 minutes ago  0.0.
0.0:9090->80/tcp  web-server-2
352ac976ff3e  registry.lab.example.com/nginx:latest  nginx -g daemon o...  4 minutes ago  Up 4 minutes ago  0.0.
0.0:8083->80/tcp  isolated-web-server
[student@workstation ~]$
```

```
[student@workstation ~]$ podman ps

CONTAINER ID  IMAGE                                  COMMAND               CREATED         STATUS             PO
RTS                 NAMES
cd0f20e083ac  registry.lab.example.com/nginx:latest  nginx -g daemon o...  39 minutes ago  Up 39 minutes ago  0.
0.0.0:8080->80/tcp  web-server
[student@workstation ~]$
[student@workstation ~]$ podman inspect web-server --format '{{ .NetworkSettings }}'
{{   0 []   0 []  []}   false  0 map[80/tcp:[{ 8080}]] /run/user/1000/netns/netns-95090751-8d7a-d424-8bf2-11fa1b
72949e map[]}
[student@workstation ~]$ podman stop web-server
web-server
[student@workstation ~]$ podman rm web-server
cd0f20e083ac100b7da3f1fcd9fc85716bd4275be74d61c1cedfb6ec66ab73d7
[student@workstation ~]$ podman run -d --name web-server-2 -p 9090:80 nginx
3c984bc71109e13c893acaba1a931f4733653d84c2373bb2cb2919ba81e7c414
[student@workstation ~]$ podman run -d --name secure-web-server -p 8081:80 --user 1001 nginx
0177f72136a408c2a98144f2c4ee0b81bfac5a55454ca99caac96492d0187269
[student@workstation ~]$ podman ps
CONTAINER ID  IMAGE                                  COMMAND               CREATED         STATUS             PO
RTS                 NAMES
3c984bc71109  registry.lab.example.com/nginx:latest  nginx -g daemon o...  48 seconds ago  Up 48 seconds ago  0.
0.0.0:9090->80/tcp  web-server-2
[student@workstation ~]$ podman run -d --name limited-web-server -p 8082:80 --cap-drop all nginx
109908b9856c33130f6b6edcf2e6a14a9ff83ba16d4ec80d641052574f03327b
[student@workstation ~]$ sudo setenforce 1
[sudo] password for student:
[student@workstation ~]$ podman network create my-network
my-network
[student@workstation ~]$ podman run -d --name isolated-web-server --network my-network -p 8083:80 nginx
352ac976ff3e63dc88ea997887827e253d8620a6fedfc900714d2f52ed677b1d
[student@workstation ~]$
```