Lab: Podman Basics

Use Podman to manage local containers.

Outcomes

You should be able to:

- Manage local containers.
- Copy files in and out of containers.
- Run a set of application containers that connect to one another via a Podman network.
- Forward a port from a container so that it is accessible from the host machine.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command starts the `basics-podman-secret` local container, and copies the necessary files for this lab. The command also verifies that Podman is available and can pull from the required registries. You can find the source code for the `basics-podman-secret` container in the `/home/student/DO188/solutions/basics-podman/secret-container` directory.

The lab script continuously evaluates the objectives of this lab. Keep the script running in a terminal window, and complete the objectives of this lab from a new terminal window.

After each objective, return to the lab script evaluation to see if you have finished the objective successfully. When you have finished all the objectives, the `lab` command prompts you to execute the `finish` function.

```
[student@workstation ~]$ lab start basics-podman
```

Instructions

1. The lab command starts the `basics-podman-secret` container, which contains the `/etc/secret-file` file.

   Copy the `/etc/secret-file` file from the container to the `/home/student/DO188/labs/basics-podman/solution` file.

   Use the `podman ps` command to verify that the `basics-podman-secret` container is runni

   ```
   [student@workstation ~]$ podman ps --format='{{.Names}}'
   basics-podman-secret
   ```

   Use the `ls` command inside of the container to verify that the container contains the `/e`

```
[student@workstation ~]$ podman exec basics-podman-secret ls /etc/secret-file
/etc/secret-file
```

Change to the `/home/student/DO188/labs/basics-podman` directory, for example:

```
[student@workstation ~]$ cd ~/DO188/labs/basics-podman
[student@workstation basics-podman]$ ls
index.html
```

Copy the `/etc/secret-file` file from the container as the `solution` file in the `/home/stu`
`podman` directory.

```
[student@workstation basics-podman]$ podman cp \
  basics-podman-secret:/etc/secret-file solution
[student@workstation basics-podman]$ ls
index.html  solution
```

The first objective of the lab script is passing.

2. Start a new container with the following parameters:

- Name: `basics-podman-server`
- Image: `registry.ocp4.example.com:8443/ubi9/httpd-24`
- Ports: Route traffic from port `8080` on your machine to port `8080` inside of the container
- Network: `lab-net`

You can start the container in the detached mode for greater convenience.

Create the `lab-net` Podman network.

```
[student@workstation ~]$ podman network create lab-net
lab-net
```

The next objective of the lab script is passing.

Execute the `podman run` command to start the container.

```
[student@workstation basics-podman]$ podman run -d --name basics-podman-server \
  --net lab-net -p 8080:8080 registry.ocp4.example.com:8443/ubi9/httpd-24
8b747...3616
```

Additional objectives of the lab script are passing.

3. Copy the `/home/student/DO188/labs/basics-podman/index.html` file to `/var/www/html/` in the `basics-podman-server` container.

   Verify that you are in the correct directory.

   ```
   [student@workstation basics-podman]$ ls
   index.html  solution
   ```

   Copy the `index.html` file in to the container.

   ```
   [student@workstation basics-podman]$ podman cp index.html \
     basics-podman-server:/var/www/html/
   ```

   In a web browser, go to `http://localhost:8080` and verify that you see the `Hello from`

   The next objective of the lab script is passing.

4. Start a new container with the following parameters:

   - Name: `basics-podman-client`
   - Image: `registry.ocp4.example.com:8443/ubi9/httpd-24`
   - Network: `lab-net`

   You can start the container in the detached mode for greater convenience.

   Execute the `podman run` command to start the container.

   ```
   [student@workstation basics-podman]$ podman run -d --name basics-podman-client \
     --net lab-net registry.ocp4.example.com:8443/ubi9/httpd-24
   8b747...3616
   ```

5. Confirm that the `basics-podman-client` container can access the `basics-podman-server` container by its DNS name. Use the `podman exec` and `curl` commands to make a request to the `basics-podman-server` container at port `8080` from the `basics-podman-client` container.

   Confirm that DNS is enabled on the `lab-net` network.

```
[student@workstation basics-podman]$ podman network inspect lab-net
...output omitted...
          "dns_enabled": true,
...output omitted...
```

Use the `curl` command inside of the `basics-podman-client` to confirm that the `basics-` from the `lab-net` network by its DNS name.

```
[student@workstation basics-podman]$ podman exec basics-podman-client \
  curl -s http://basics-podman-server:8080
<h1>Hello from Podman Basics lab</h1>
```

Finish

As the `student` user on the `workstation` machine, change to the `student` user home directory and use the `lab` command to complete this exercise. This is important to ensure that resources from previous exercises do not impact upcoming exercises.

Press `y` when the `lab start` command prompts you to execute the `finish` function. Alternatively, execute the following command:

```
[student@workstation ~]$ lab finish basics-podman
```

All remaining objectives are passing.