

# Red Hat OpenShift Development I: Introduction to Containers with Podman 4.14

## CERTIFICATE OF ATTENDANCE

THIS IS TO CERTIFY THAT

**Fnu Pratichi**

George Washington University

HAS SUCCESSFULLY ATTENDED THE COURSE

**Red Hat OpenShift Development I: Introduction to  
Containers with Podman (DO188)**

Dec. 3, 2024



*Karl Reynolds*

KARL REYNOLDS  
Senior Director, Red Hat Training & Certification:  
Global Sales And Delivery



Please note that documentation (only proofs are required) is not needed for project 7 and 8 as discussed with professor in one of the classes.

# Chapter 1. Introduction and Overview of Containers

No labs

# Chapter 2. Introduction and Overview of Containers

Lab: Pod man Basics

```
[student@workstation:~]— /home/student/.venv/labs/bin/python /home/student/.venv/labs

[student@workstation ~]$ PS1="Fnu Pratichi>"  
Fnu Pratichi>lab start basics-podman  
SUCCESS Verifying cluster state  
SUCCESS Verifying if Podman is installed  
SUCCESS Authenticating Podman with classroom registry  
SUCCESS Copy exercise files  
SUCCESS Copying container files  
SUCCESS Verifying port 8080 is not in use  
SUCCESS Authenticating Podman with classroom registry  
SUCCESS Starting the basics-podman-secret container  
  
Starting lab.
```

```
[student@workstation basics-podman]$ podman network inspect lab-net  
[  
  {  
    "name": "lab-net",  
    "id": "ca67ec0b5449624a5ad0e79ebecaf94a06f8f67361f6dd715415c16868927e61",  
    "driver": "bridge",  
    "network_interface": "podman1",  
    "created": "2024-12-02T19:35:23.955221389-05:00",  
    "subnets": [  
      {  
        "subnet": "10.89.0.0/24",  
        "gateway": "10.89.0.1"  
      }  
    ],  
    "ipv6_enabled": false,  
    "internal": false,  
    "dns_enabled": true,  
    "ipam_options": {  
      "driver": "host-local"  
    }  
  }  
]
```

```
student@workstation:~  
Hint: Use the registry.ocp4.example.com:8443/ubi8/httpd-24 container image  
✖ 8. The basics-podman-client container is missing a network  
  A container is missing a required network  
  Hint: Remove and create the basics-podman-client with the required network  
  0 passing  
  2 failing  
  
⌚ [19:38:54] Watching Basics Podman Lab  
  
✓ 1. Extract a secret file from a container  
✓ 2. Podman network created  
✓ 3. Start the basics-podman-server container  
✓ 4. The basics-podman-server container is missing a network  
✓ 5. Container basics-podman-server should have published ports  
✓ 6. Checking index.html content inside the container  
✓ 7. Start the basics-podman-client container  
✓ 8. The basics-podman-client container is missing a network  
  8 passing  
  0 failing  
  
You have successfully completed the exercise.  
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]  
Finishing lab.  
SUCCESS Verifying if Podman is installed  
SUCCESS Stopping and removing specified containers  
SUCCESS Removing specified networks  
SUCCESS Pruning images  
Fnu Pratichi>
```

## Chapter 3

### Lab: Container Images

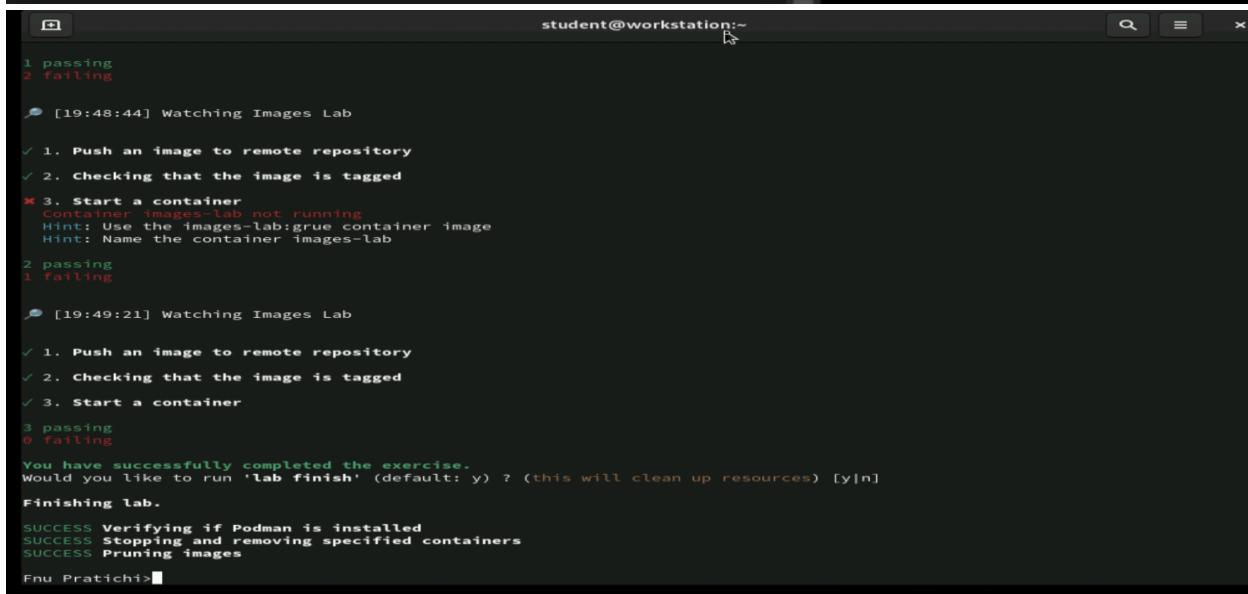
```
student@workstation:~ — /home/student/.venv/labs/bin/python  
  
Fnu Pratichi>lab start images-lab  
SUCCESS Verifying cluster state  
SUCCESS Verifying if Podman is installed  
SUCCESS Copy exercise files  
  
Starting lab.
```

```
[student@workstation images-lab]$ curl localhost:8080
It is pitch black. You are likely to be eaten by a grue.
[student@workstation images-lab]$ lab finish images-lab
```

**Finishing lab.**

```
SUCCESS Verifying if Podman is installed
SUCCESS Stopping and removing specified containers
SUCCESS Pruning images
```

```
[student@workstation images-lab]$ █
```



The screenshot shows a terminal window titled 'student@workstation:~'. It displays the results of the 'lab finish' command for the 'images-lab' exercise. The output includes:

- Exercise summary: 1 passing, 2 failing.
- Task 1: Push an image to remote repository (✓)
- Task 2: Checking that the image is tagged (✓)
- Task 3: Start a container (✗)
  - Container 'images-lab' not running
  - Hint: Use the 'images-lab:grue' container image
  - Hint: Name the container 'images-lab'
- Total: 2 passing, 1 failing.

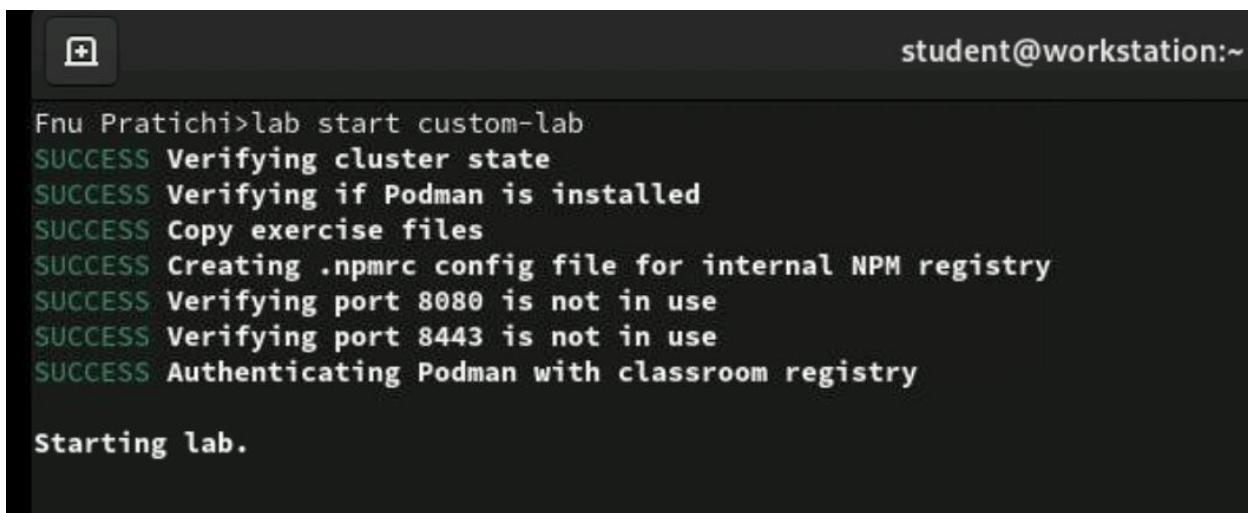
At the end, it asks if you want to run 'lab finish' again, with options [y|n]. The user types 'y' and the terminal shows the final cleanup steps:

```
You have successfully completed the exercise.
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]
Finishing lab.

SUCCESS Verifying if Podman is installed
SUCCESS Stopping and removing specified containers
SUCCESS Pruning images
```

## Chapter 4

### Lab: Custom Container Images



The screenshot shows a terminal window titled 'student@workstation:~'. It displays the output of the 'lab start custom-lab' command. The output includes:

- Starting the 'custom-lab' exercise.
- Verifying cluster state (SUCCESS)
- Verifying if Podman is installed (SUCCESS)
- Copying exercise files (SUCCESS)
- Creating .npmrc config file for internal NPM registry (SUCCESS)
- Verifying port 8080 is not in use (SUCCESS)
- Verifying port 8443 is not in use (SUCCESS)
- Authenticating Podman with classroom registry (SUCCESS)

Finally, it starts the lab with the message: "Starting lab."

```
Fnu Pratichi>podman run --name=custom-lab \
> -p 8080:8080 -p 8443:8443 podman-qr-app
```

```
> custom-images-lab@1.0.0 start
> node index.js
```

```
TLS Server running on port 8443
Server running on port 8080
```

```
[20:06:30] Watching Custom Images Lab

✓ 1. Set the environment variables
✓ 2. Set the /app working directory
✓ 3. Install application dependencies
✓ 4. Set user
✓ 5. Set entrypoint command
✓ 6. Test container ports

6 passing
0 failing

You have successfully completed the exercise.
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]

Finishing lab.

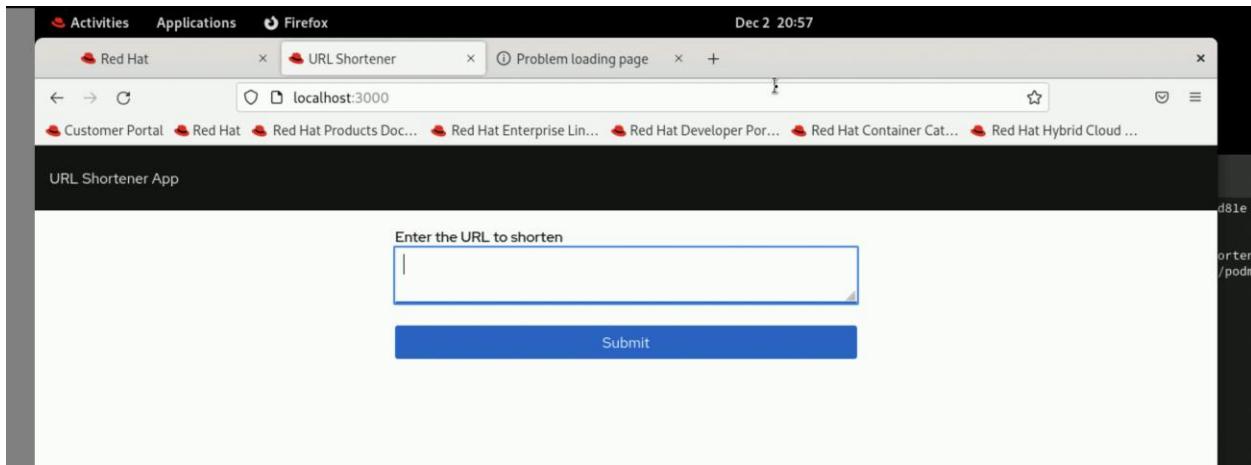
SUCCESS Verifying if Podman is installed
SUCCESS Stopping and removing specified containers
SUCCESS Pruning images

Fnu Pratichi>
```

# Chapter 5. Persisting Data

## Lab: Persisting Data

```
student@workstation:~ — /home/student/.venv/labs/bin/python /home/stud...
[student@workstation ~]$ PS1="Fnu Pratichi>"  
Fnu Pratichi> lab start persisting-lab  
SUCCESS Verifying cluster state  
SUCCESS Verifying if Podman is installed  
SUCCESS Copy exercise files  
SUCCESS Copying container files  
SUCCESS Verifying port 8080 is not in use  
SUCCESS Verifying port 3000 is not in use  
SUCCESS Authenticating Podman with classroom registry  
  
Starting lab.
```



```
[20:52:20] Watching Persisting Lab  
  
✓ 1. Create the postgres-vol volume  
✓ 2. Import data into the postgres-vol volume  
✓ 3. Start the persisting-db database container  
✓ 4. Check volume on the persisting-db database container  
✓ 5. Check that the persisting-backend container runs on port 8080  
✓ 6. Check that the persisting-frontend container runs on port 3000  
✓ 7. Check the application works  
  
7 passing  
0 failing  
  
You have successfully completed the exercise.  
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]  
Fnu Pratichi>
```

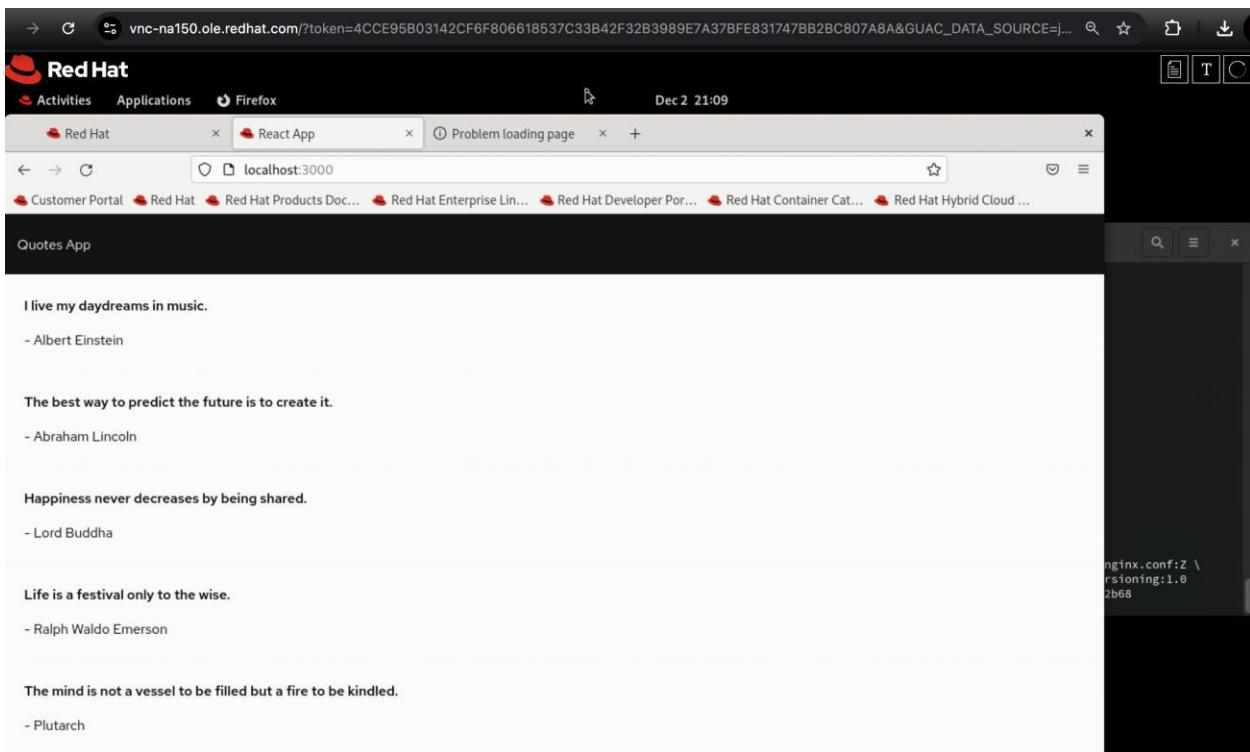
```
Copying blob libcurl/dbeada4 done  
Copying blob 5c4e685edac1 done  
Copying config b745b43a62 done  
Writing manifest to image destination  
Storing signatures  
e30e9b7382e28a876d40d171e2cc989f1c25f46f22f213053339b71a7a41d5b  
Fnu Pratichi>lab finish persisting-lab  
  
Finishing lab.  
  
SUCCESS Verifying if Podman is installed  
SUCCESS Stopping and removing specified containers  
SUCCESS Removing specified volumes ['postgres-vol']  
SUCCESS Removing specified networks  
SUCCESS Pruning images  
  
Fnu Pratichi>[ ]
```

# Chapter 6. Troubleshooting Containers

## Lab: Troubleshooting Containers

```
[+] student@workstation:~ — /home/student/.venv/labs/bin/python /home/student/v... x
Fnu Pratichi> lab start troubleshooting-lab
SUCCESS Verifying cluster state
SUCCESS Verifying if Podman is installed
SUCCESS Copy exercise files
SUCCESS Authenticating Podman with classroom registry
SUCCESS Create specified network ['troubleshooting-lab']
SUCCESS Starting the quotes-api-v1 container
SUCCESS Starting the quotes-api-v2 container
SUCCESS Starting the quotes-ui container

Starting lab.
```



The image shows two terminal windows side-by-side. The left terminal window displays the output of a command-line exercise. It includes hints, test cases, and a summary of results. The right terminal window shows the final steps of a troubleshooting lab, including stopping and removing containers, and pruning images.

```
student@workstation:~ — /home/student/.venv/labs/bin/python /home/student/v... student@workstation:~  
Hint: Check NGINX mappings  
✖ 6. Version v2 of the quotes application should be accessible from the host browser  
  Cannot connect to http://localhost:3000/api/v2/quotes  
  Hint: Check that the application is working and that is accessible from the host  
0 passing  
6 failing  
⌚ [21:08:50] Watching troubleshooting-lab  
✓ 1. Expected running containers: quotes-api-v1, quotes-api-v2, quotes-ui  
✓ 2. Containers should be attached to the specified networks  
✓ 3. quotes-ui should have the right environment variables  
✓ 4. quotes-ui should have /home/student/D0188/labs/troubleshooting-lab/nginx.conf accessible in the container at /etc/nginx/nginx.conf  
✓ 5. The quotes-ui container can reach v2  
✓ 6. Version v2 of the quotes application should be accessible from the host browser  
6 passing  
0 failing  
You have successfully completed the exercise.  
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n] [ ]
```

```
Fnu Pratichi>podman rm -f quotes-ui  
Fnu Pratichi>podman run -d \  
  --name quotes-ui \  
  -p 3000:8080 \  
  --env QUOTES_API_VERSION=v2 \  
  --net troubleshooting-lab \  
  -v ~/D0188/labs/troubleshooting-lab/nginx.conf:/etc/nginx/nginx.conf:Z \  
  registry.ocp4.example.com:8443/redhattraining/quotes-ui-versioning:1.0  
Fnu Pratichi>lab finish troubleshooting-lab  
Finishing lab.  
SUCCESS Verifying if Podman is installed  
SUCCESS Stopping and removing specified containers  
SUCCESS Removing specified networks  
SUCCESS Pruning images  
Fnu Pratichi:>[ ]
```

## Chapter 7. Multi-container Applications with Compose

### Lab: Multi-container Applications with Compose

The image shows a Red Hat desktop environment with a terminal window open. The terminal window displays the command-line steps to start a multi-container application using Compose. It includes verifying cluster state, installing Podman, copying exercise files, and authenticating Podman with a classroom registry. The terminal also shows the start of the lab.

```
Red Hat Activities Applications Terminal student@workstation:~  
[student@workstation ~]$ PS1="Fnu pratichi>"  
Fnu pratichi>lab start compose-lab  
SUCCESS Verifying cluster state  
SUCCESS Verifying if Podman is installed  
SUCCESS Copy exercise files  
SUCCESS Authenticating Podman with classroom registry  
Starting lab.
```

vnc-na150.ole.redhat.com/?token=5E60700CD19F3BFE30D3570E4C1A009B38C6A6E695A1820729F7BE0D50515270&GUAC\_DATA\_SOURCE=jwt&G...

## Red Hat

Activities Applications Firefox Dec 2 21:50 student@workstation:~/D0188/labs/compose-lab

student@workstation:~ — /home/student/.venv/labs/bin/python /home/student/v...

React App localhost:3000

Red Hat Products Documentation Red Hat Enterprise Linux Red Hat Developer Portal Red Hat Container Catalog Red Hat Hybrid Cloud ...

Red Hat Quotes App

✓ 2. q n the Red Hat React App

✓ 3. c

✓ 4. c

✗ 5. E Com Hint

✗ 6. C Com Hint Hint

3 pass 3 fail

I live my daydreams in music.  
— Albert Einstein

The best way to predict the future is to create it.  
— Abraham Lincoln

Happiness never decreases by being shared.  
— Lord Buddha

Life is a festival only to the wise.  
— Ralph Waldo Emerson

The mind is not a vessel to be filled but a fire to be kindled.  
— Plutarch

[21:49:53] Watching Compose Lab

✓ 1. Expected running containers: quotes-api, quotes-provider, quotes-ui

✓ 2. quotes-provider should have /home/student/D0188/labs/compose-lab/wiremock/stubs accessible in the container at /home/wiremock

✓ 3. Container quotes-api should have published ports

✓ 4. Containers should be attached to the specified networks

✓ 5. Expect quotes-api to consume quotes-provider

✓ 6. Container quotes-ui should have published ports

6 passing  
0 failing

You have successfully completed the exercise.

Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]

Finishing lab.

SUCCESS Verifying if Podman is installed  
SUCCESS Stopping and removing specified containers  
SUCCESS Removing specified networks  
SUCCESS Pruning images

Fnu pratichi>

# Chapter 8. Container Orchestration with OpenShift and Kubernetes

## Lab: Container Orchestration with Kubernetes and OpenShift

The screenshot shows a Red Hat workstation desktop environment. At the top, there is a header bar with the Red Hat logo and navigation links for Activities, Applications, and Terminal. Below the header is a terminal window titled "student@workstation:~ — /home/student/.venv/labs/bin/python /home/student/v...". The terminal displays the output of a command to start an OpenShift lab, showing steps like verifying cluster state, copying exercise files, and creating projects. It ends with the message "Starting lab.". Below the terminal is a web browser window displaying a quote from Albert Einstein: "I live my daydreams in music." - Albert Einstein. The browser also shows the URL [quotes-ui-ocp-lab.apps.ocp4.example.com](https://quotes-ui-ocp-lab.apps.ocp4.example.com). The status bar at the bottom of the browser window shows the URL <https://developers.redhat.com/>.

```
Fnu pratichi>lab start openshift-lab
F SUCCESS Verifying cluster state
C SUCCESS Copy exercise files
F SUCCESS Verifying your OpenShift API URL
F\ SUCCESS Creating the ['ocp-lab'] OpenShift projects
C - Project created: 'ocp-lab'
C SUCCESS Applying OpenShift resources
F
F Starting lab.
```

I live my daydreams in music.  
- Albert Einstein

The best way to predict the future is to create it.  
- Abraham Lincoln

Happiness never decreases by being shared.  
- Lord Buddha

Life is a festival only to the wise.  
- Ralph Waldo Emerson

The mind is not a vessel to be filled but a fire to be kindled.  
- Plutarch

```
[21:59:34] Watching OpenShift Lab

✓ 1. quotes-api pod is online
✓ 2. quotes-api service exists
✓ 3. quotes-api has correct port
✓ 4. quotes-api routes to quotes-api
✓ 5. http://quotes-ui-ocp-lab.apps.ocp4.example.com:80/ should be reachable on the host

5 passing
0 failing

You have successfully completed the exercise.
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]

Fnu Pratichi>lab finish openshift-lab
SUCCESS Verifying cluster state

Finishing lab.

SUCCESS Verifying your OpenShift API URL
SUCCESS Deleting OpenShift projects
    - Project deleted: 'ocp-lab'

Fnu Pratichi>
```

## Chapter 9. Comprehensive Review

### Lab: Comprehensive Review

→ C vnc-na150.ole.redhat.com/?token=5E60700CD19F3BFE30D3570E4C1A0

 Red Hat

Activities Applications Terminal

student@

```
Fnu Pratichi>lab start comprehensive-review
SUCCESS Verifying cluster state
SUCCESS Copy exercise files
SUCCESS Verifying if Podman is installed
SUCCESS Verifying port 8080 is not in use
SUCCESS Authenticating Podman with classroom registry

Starting lab.
```

```
student@workstation:~/DO188/labs/comprehensive-review/be
3/3.7/commons-lang3-3.7.jar (500 kB at 922 kB/s)
Downloaded from internal-repository: http://nexus-infra.apps.ocp4.example.com/repositor
roid/guava-28.2-android.jar (2.6 MB at 4.4 MB/s)
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:10 min
[INFO] Finished at: 2024-12-03T03:12:15Z
[INFO] -----
--> cc6df287133
[2/2] STEP 1/3: FROM registry.ocp4.example.com:8443/ubi8/openjdk-17-runtime:1.12
Trying to pull registry.ocp4.example.com:8443/ubi8/openjdk-17-runtime:1.12...
Getting image source signatures
Copying blob 54e56e6f8572 skipped: already exists
Copying blob 4f8ddd7f5a75 skipped: already exists
Copying blob 9790122be13e done
Copying config a7113ef97b done
Writing manifest to image destination
Storing signatures
[2/2] STEP 2/3: COPY --from=builder /home/jboss/target/beeper-1.0.0.jar .
--> f6b616cc0e0
[2/2] STEP 3/3: ENTRYPOINT ["java", "-jar", "beeper-1.0.0.jar"]
[2/2] COMMIT beeper-api:v1
--> 60c053ac766
Successfully tagged localhost/beeper-api:v1
60c053ac766ac469eb06398c57a0d592e07f6916f6da634d42d3f3e97377aala
Fnu Pratichi>podman run -d \
> --name beeper-api --net beeper-backend,beeper-frontend \
> -e DB_HOST=beeper-db beeper-api:v1
1b0e72f48ee94c8690aeeba29e793d99e5492f71207a778ec18ac0181804c64
Fnu Pratichi>cd \
> ~/DO188/labs/comprehensive-review/beeper-ui
Fnu Pratichi>vim Containerfile
Fnu Pratichi>podman build -t beeper-ui:v1 .
[1/2] STEP 1/4: FROM registry.ocp4.example.com:8443/ubi9/nodejs-18:1 AS builder
Trying to pull registry.ocp4.example.com:8443/ubi9/nodejs-18:1...
Getting image source signatures
Copying blob ed459618bd73 skipped: already exists
Copying blob a20d5f0bec2b skipped: already exists
```

```
⌚ [22:17:44] Watching comprehensive review

✓ 1. Expected running containers: beeper-db
✓ 2. beeper-db should have the expected environment variables
✓ 3. beeper-db should be connected to the correct networks
✓ 4. beeper-db should have volume beeper-data accessible in the container at /var/lib/pgsql/data
✓ 5. Expected running containers: beeper-api
✓ 6. beeper-api should have the expected environment variables
✓ 7. beeper-api should respond to requests
✓ 8. beeper-api should be connected to the correct networks
✓ 9. Expected running containers: beeper-ui
✓ 10. beeper-ui should be connected to the correct networks
✓ 11. Container beeper-ui should have published ports
✓ 12. http://localhost:8080/ should be reachable on the host
✓ 13. The API should be available via the UI container

13 passing
0 failing

You have successfully completed the exercise.
Would you like to run 'lab finish' (default: y) ? (this will clean up resources) [y|n]

Finishing lab.

SUCCESS Verifying if Podman is installed
SUCCESS Stopping and removing specified containers
SUCCESS Removing specified networks
SUCCESS Removing specified volumes ['beeper-data']
SUCCESS Pruning images

Fnu Pratichi>
```