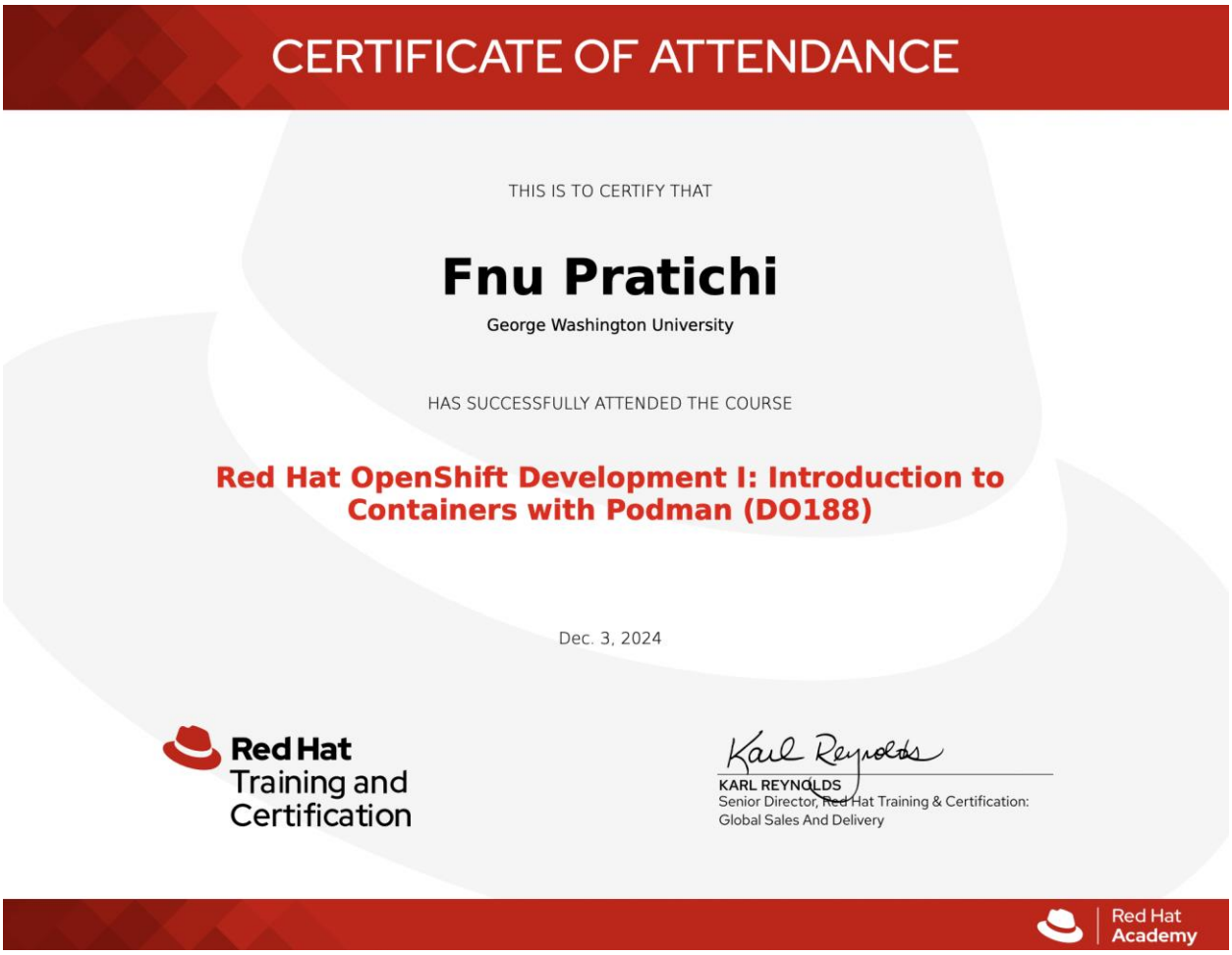
Red Hat OpenShift Development I: Introduction to Containers with Podman 4.14



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

Red Hat Training and Certification

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

Red Hat Academy

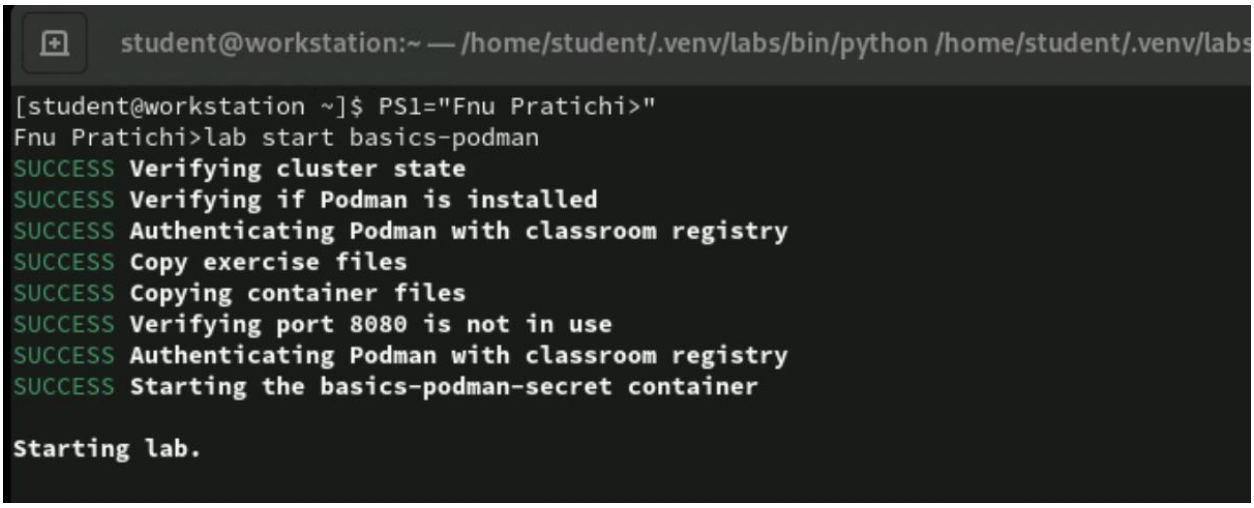
Please note that documentation (only proofs are required) is not needed for project 7 and 8 as discussed with professor in one of the classes.

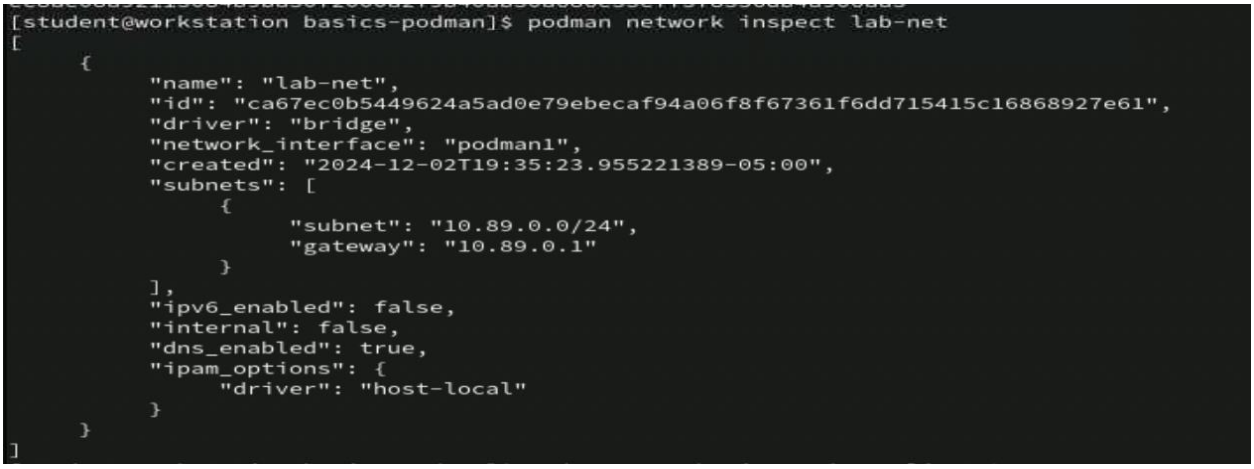
Chapter 1. Introduction and Overview of Containers

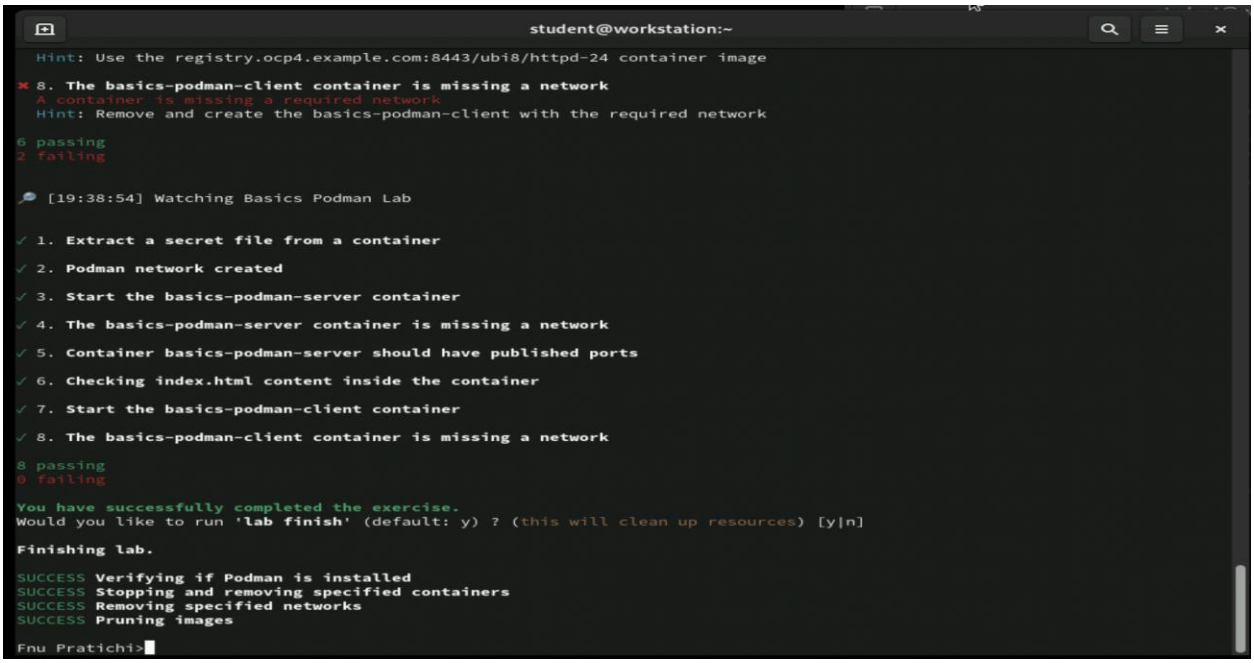
No labs

Chapter 2. Introduction and Overview of Containers

Lab: Pod man Basics

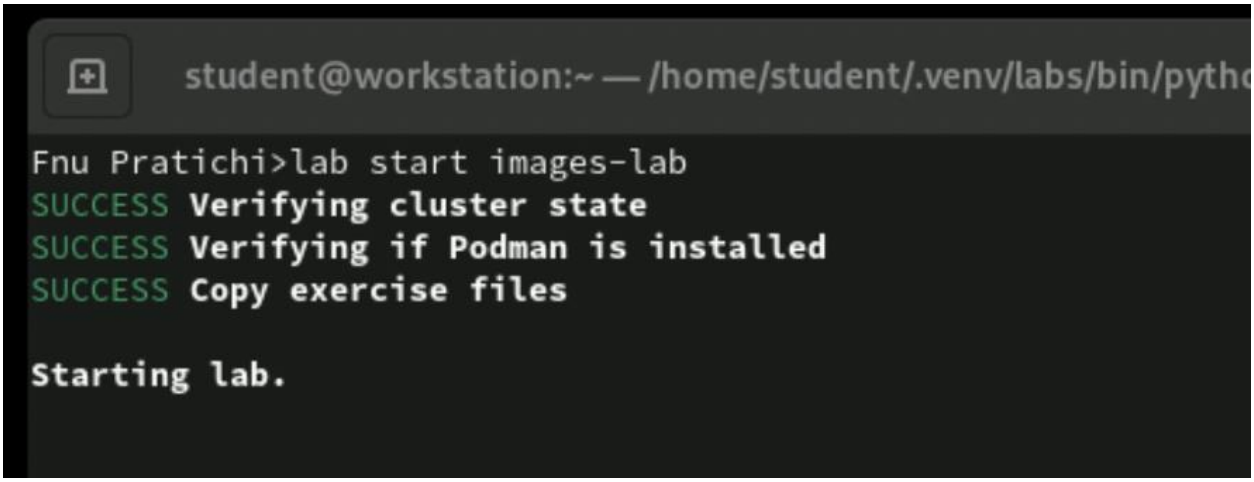






Chapter 3

Lab: Container Images



```
[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]$
```

```
student@workstation:~

1 passing
2 failing

[19:48:44] Watching Images Lab

✓ 1. Push an image to remote repository

✓ 2. Checking that the image is tagged

✖ 3. Start a container
  Container images-lab not running
  Hint: Use the images-lab:grue container image
  Hint: Name the container images-lab

2 passing
1 failing

[19:49:21] Watching Images Lab

✓ 1. Push an image to remote repository

✓ 2. Checking that the image is tagged

✓ 3. Start a container

3 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 4

Lab: Custom Container Images

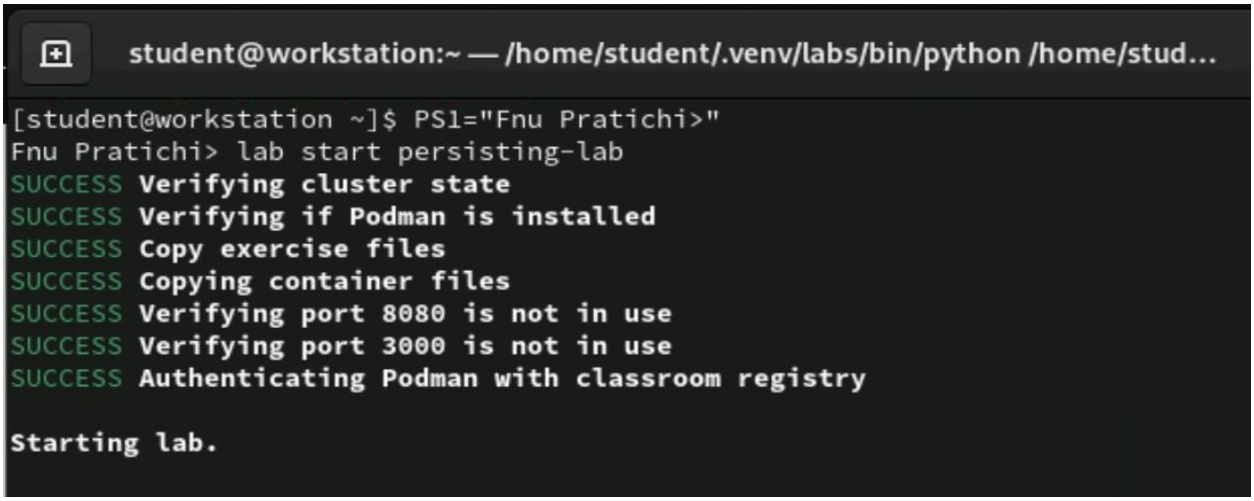
```
student@workstation:~

Fnu Pratichi>lab start custom-lab
SUCCESS Verifying cluster state
SUCCESS Verifying if Podman is installed
SUCCESS Copy 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

Starting lab.
```

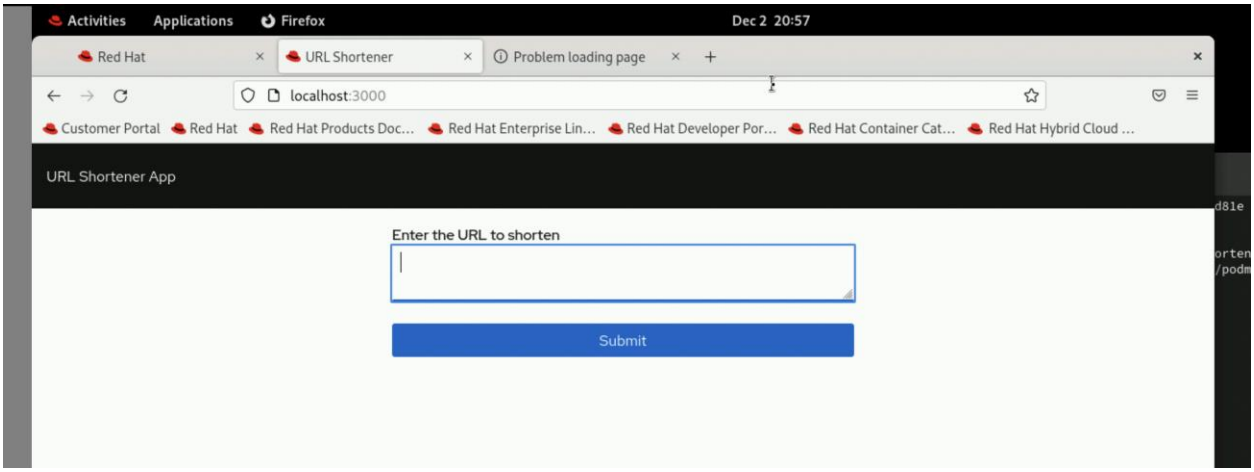
```
Fnu Pratichi>podman run --name=custom-lab \
>  -p 8080:8080 -p 8443:8443 podman-qr-app

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

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

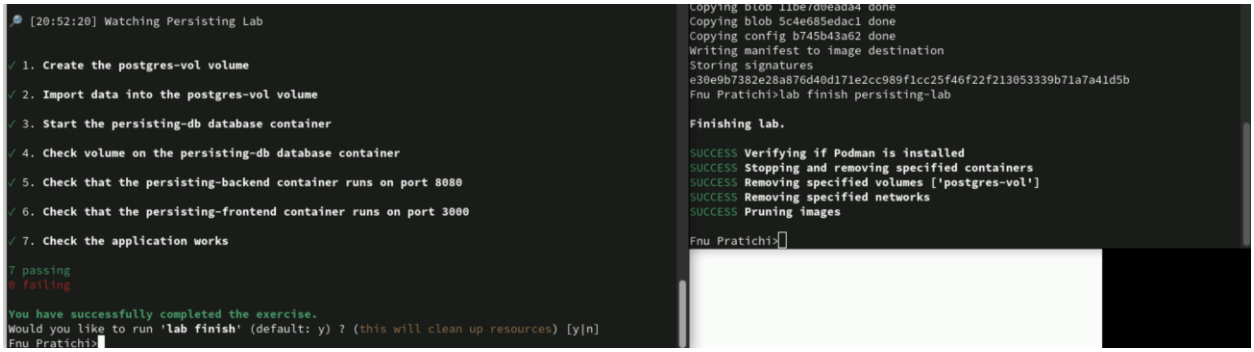
```
🔎 [20:06:30] Watching Custom Images Lab

✓ 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

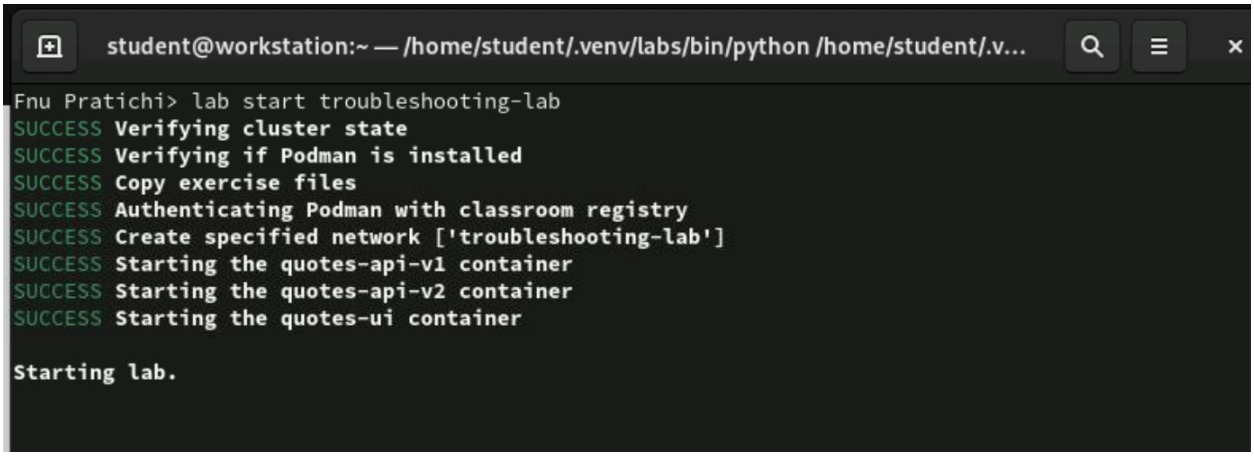


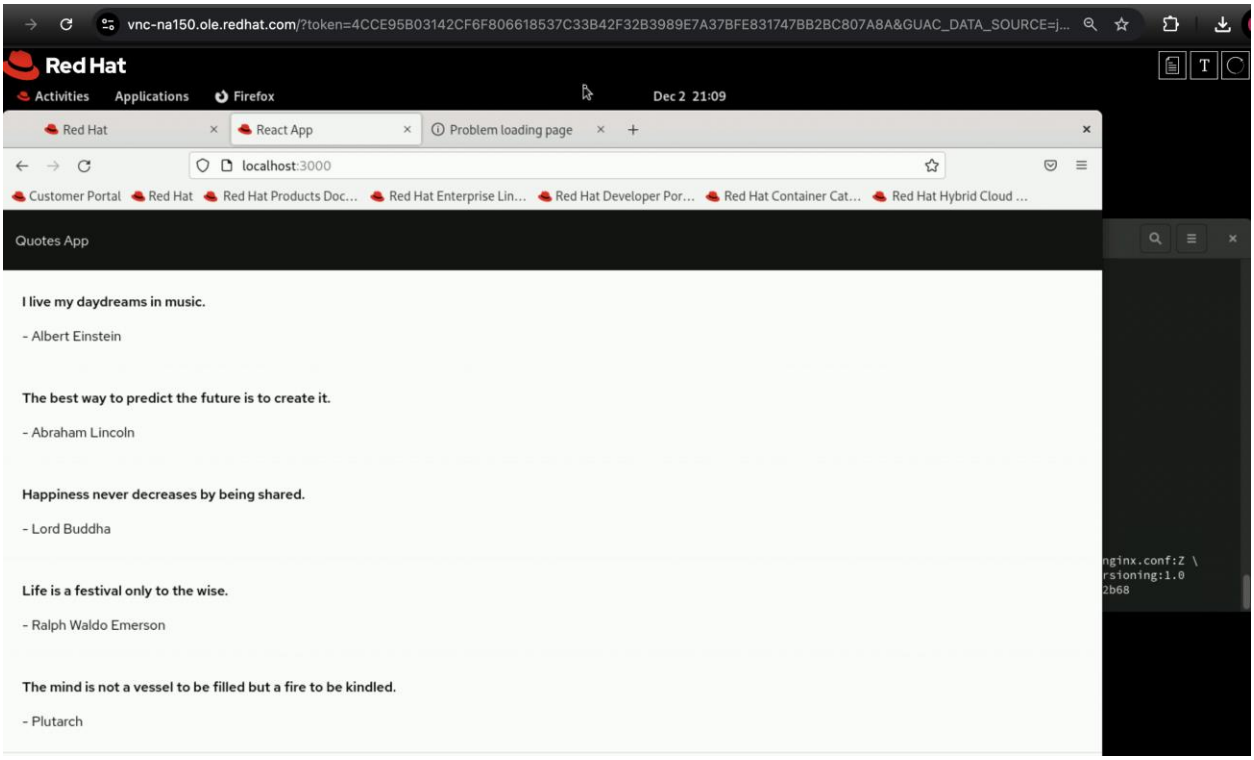


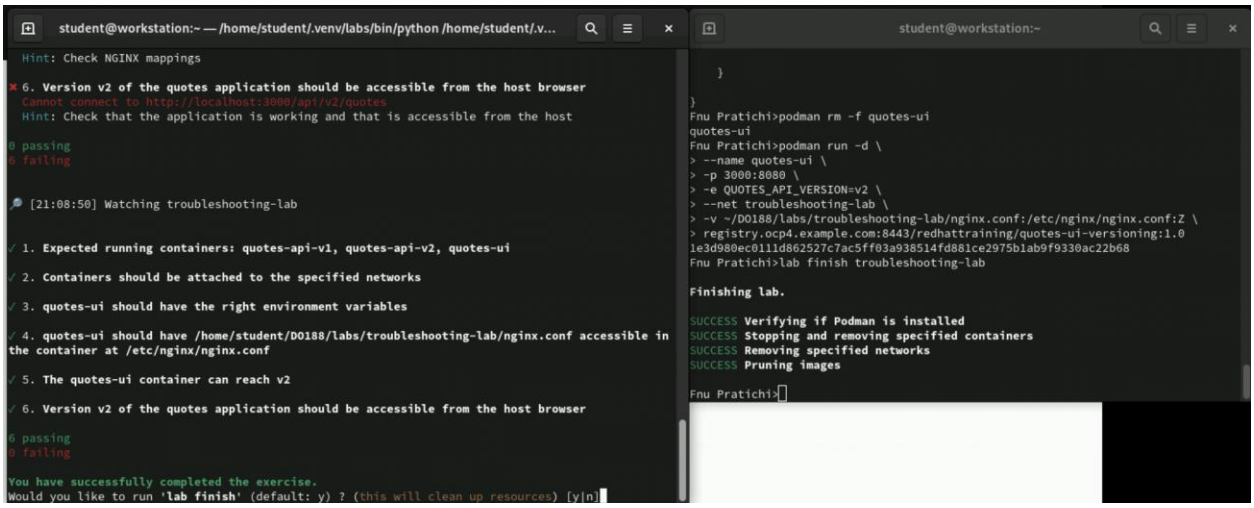


Chapter 6. Troubleshooting Containers

Lab: Troubleshooting Containers

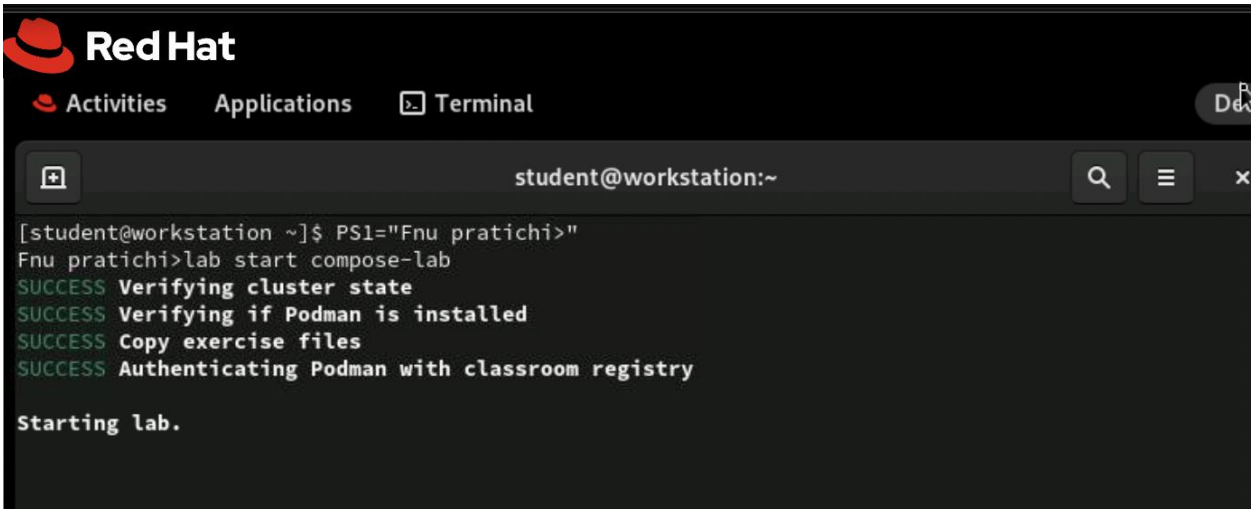


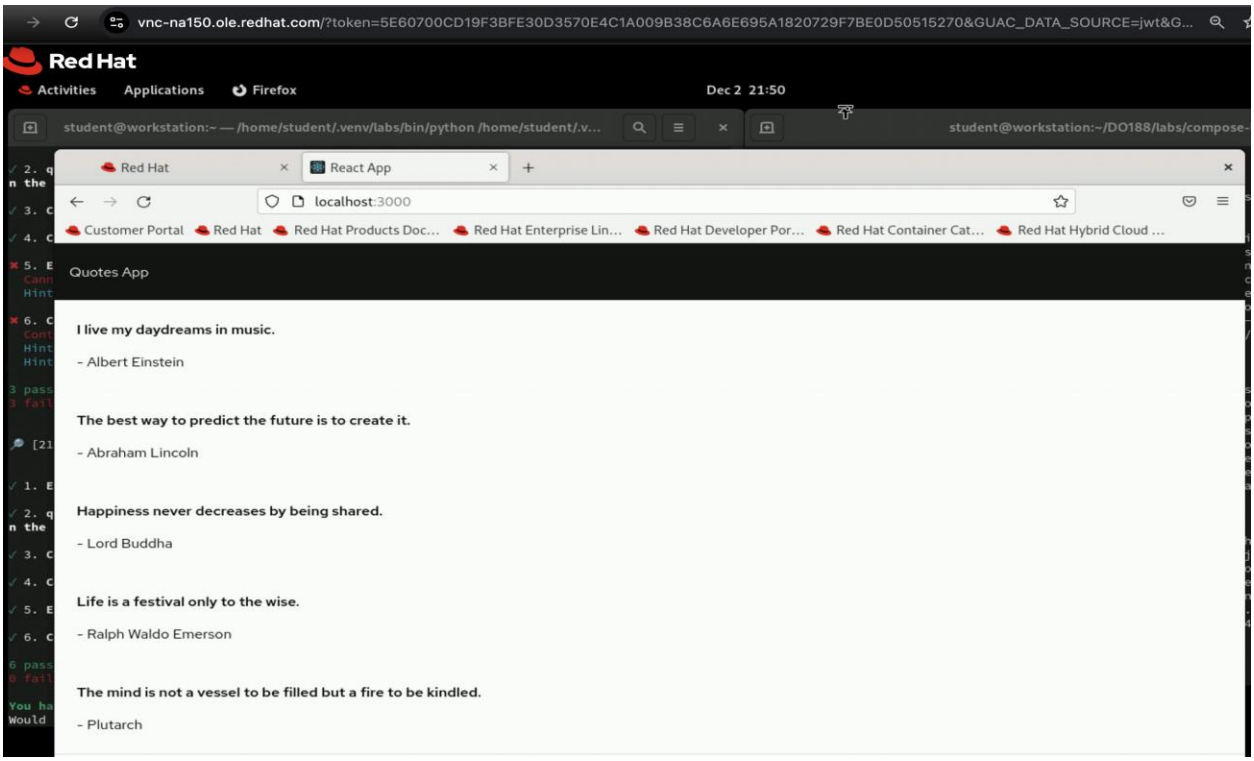




Chapter 7. Multi-container Applications with Compose

Lab: Multi-container Applications with Compose

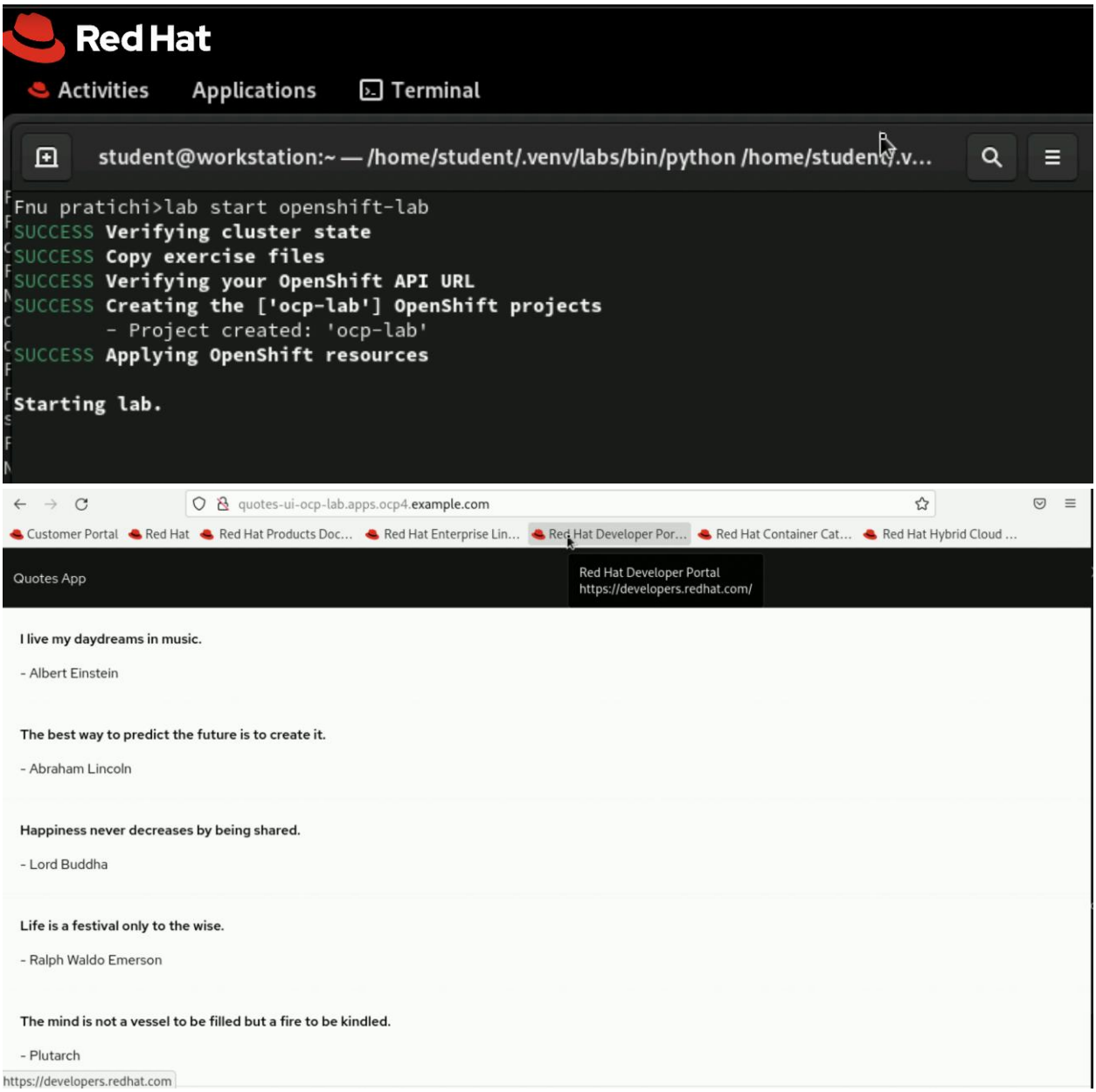


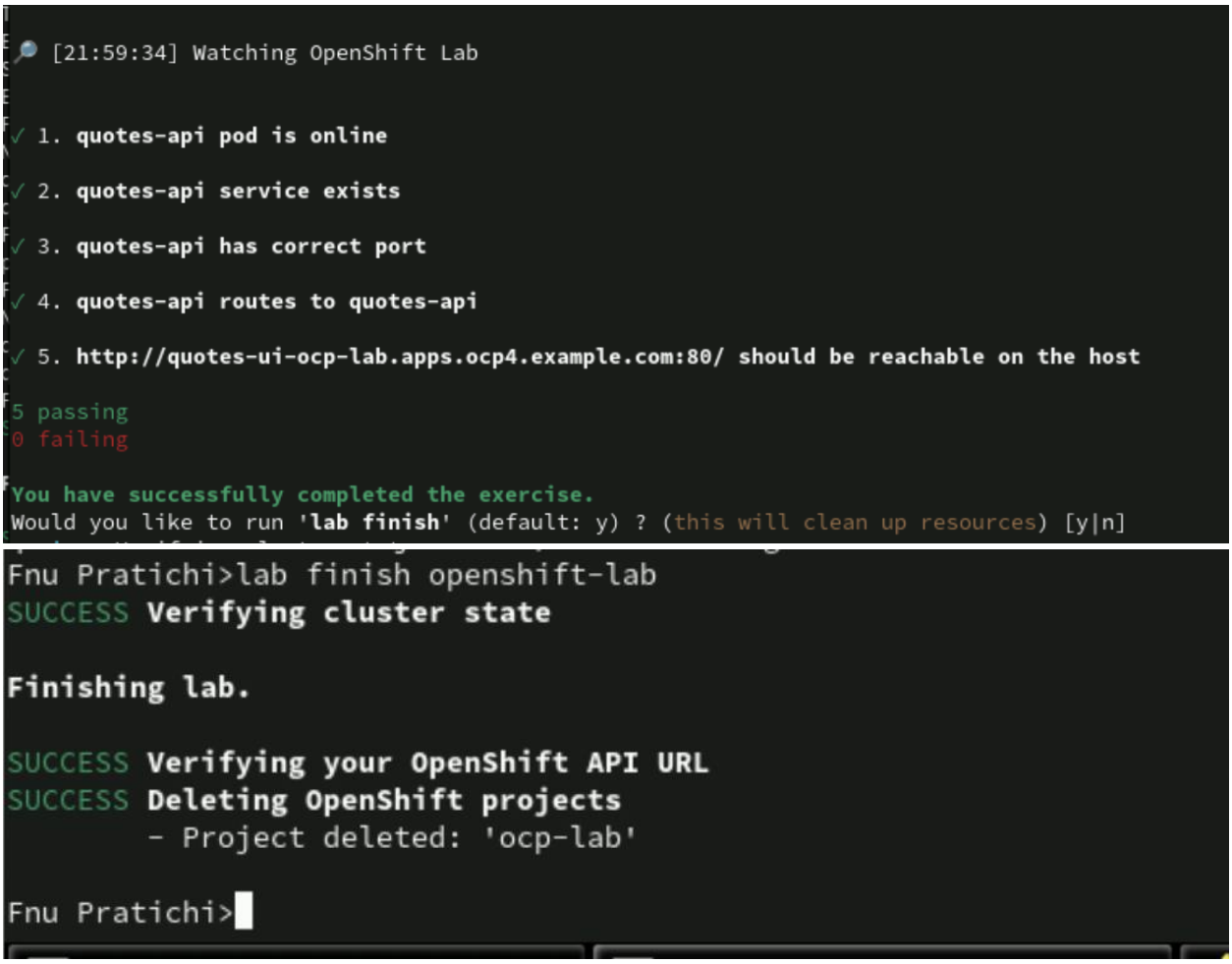



```
🔎 [21:49:53] Watching Compose Lab


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

✓ 2. quotes-provider should have /home/student/DO188/labs/compose-lab/wiremock/stubs accessible i
n 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>
Fnu pratichi>
```

Chapter 8. Container Orchestration with OpenShift and Kubernetes

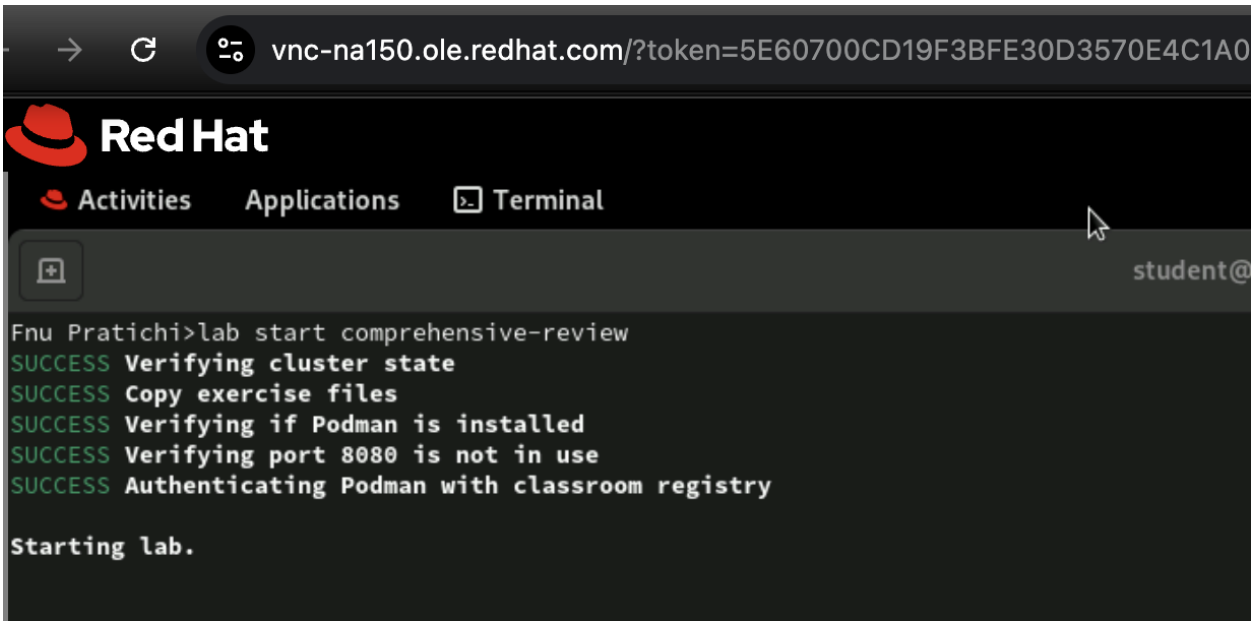
Lab: Container Orchestration with Kubernetes and OpenShift



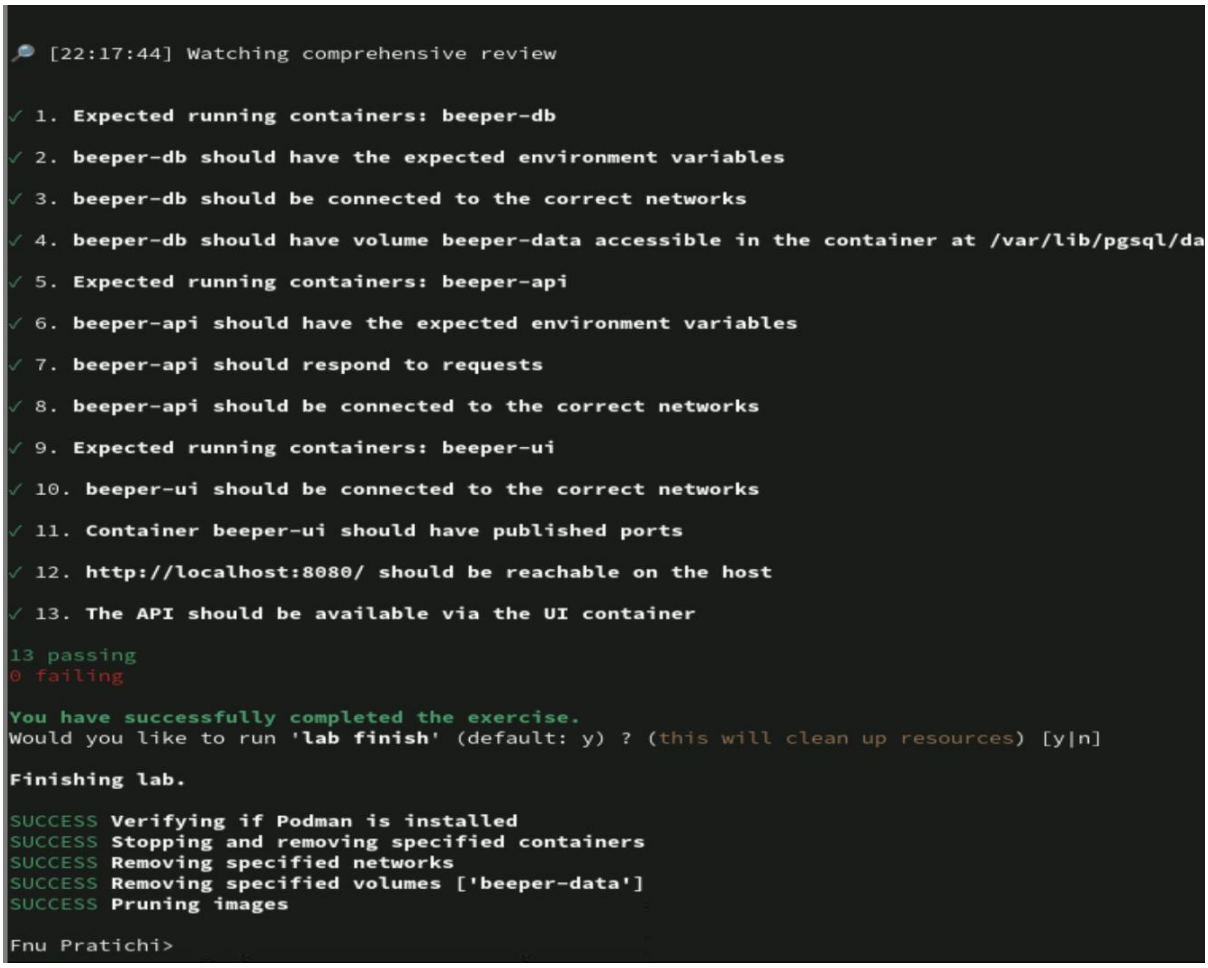


Chapter 9. Comprehensive Review

Lab: Comprehensive Review




```
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
60c053ac766ac469eb06398c57a0d592e07f6916f6da634d42d3f3e97377aa1a
Fnu Pratichi>podman run -d \
> --name beeper-api --net beeper-backend,beeper-frontend \
> -e DB_HOST=beeper-db beeper-api:v1
1b0e72f48ee94c8690aeeba29e793d99e5492f71207a778ece18ac0181804c64
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/da
✓ 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>