Exercise 9

Containerfile

Q1.

Create a file “source” which will contain the content as
“CREATING OUR FIRST CONTAINERFILE”

Then, Create a Containerfile with the following specifications:

Use the ubi9 image by Red Hat for this Containerfile. Write your name and mail ID as the Maintainer. Add the label : description = “This is a custom Apache Image”. You are to define an environment variable TRAINING=redhat. The next step is to install Apache package and expose the 80 port. Copy the previously created “source” file to the document root. Lastly

```
ENTRYPOINT  ["/usr/sbin/httpd"]
CMD         ["-D", "FOREGROUND"]
```

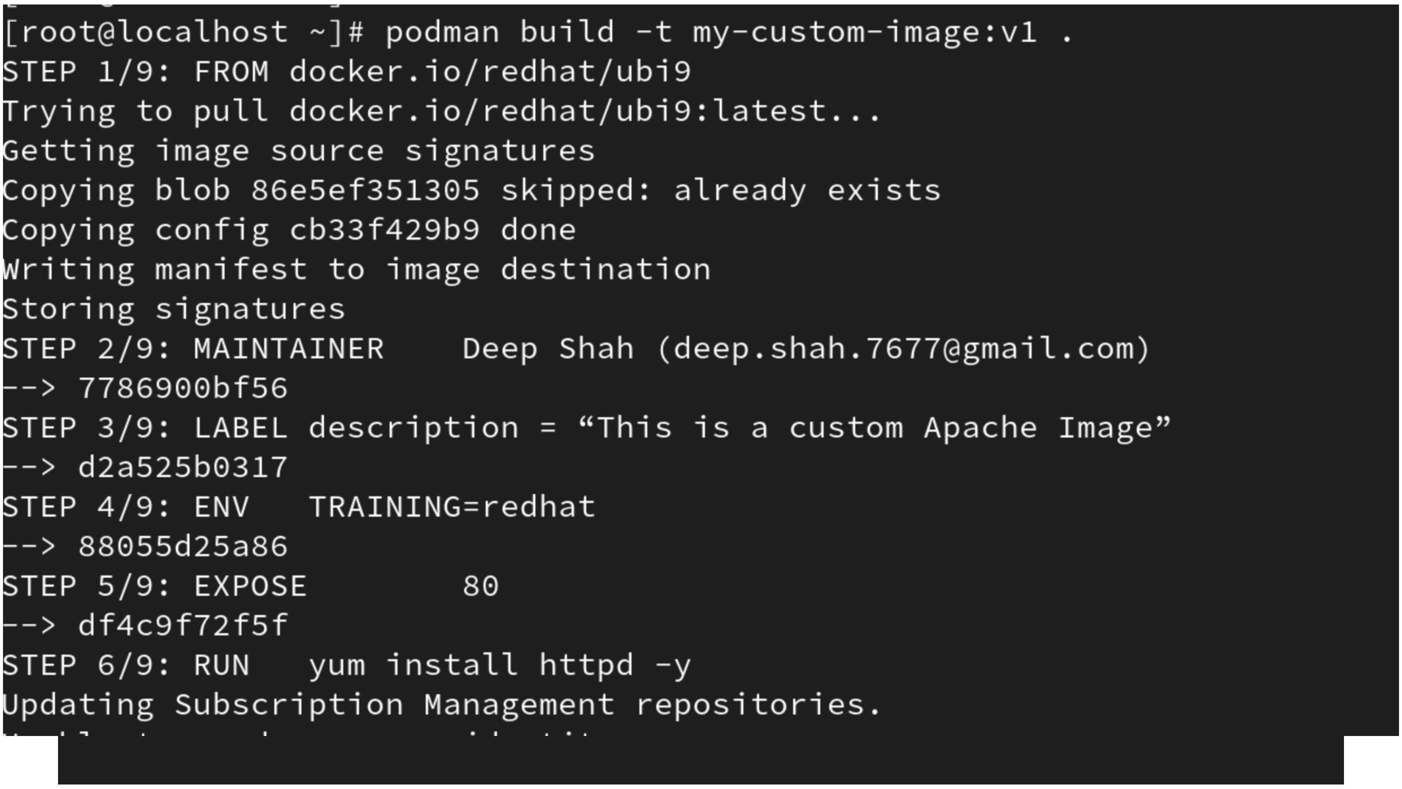
Should be added.

Build this Containerfile to an image with the name “my-custom-image:v1”
Then launch a container using this image and on doing a curl, the content:
“CREATING OUR FIRST CONTAINERFILE”
should be visible.

Q2.

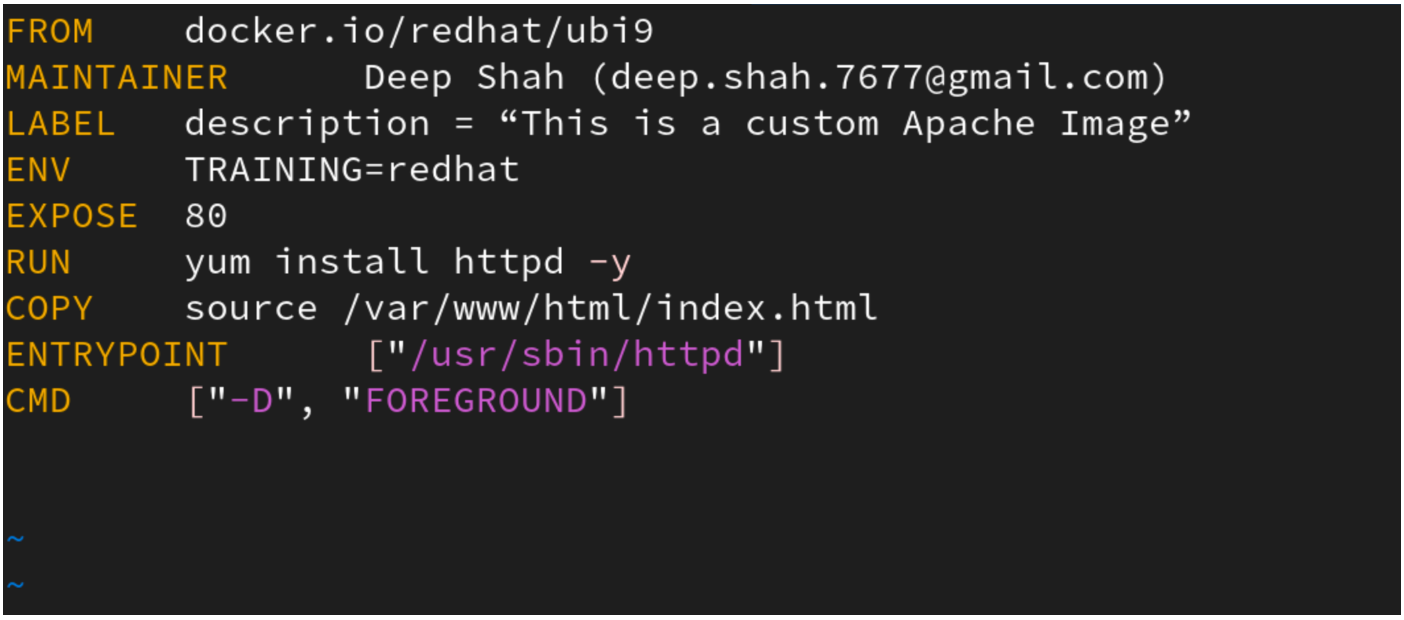
Once done with all of this, you are to now push this image onto your docker registry server with the image name as “custom-container:v1.0” and perform all the required steps to do so.

You can verify everything was completed properly by removing the container and image from your machine and pulling the image you pushed and executing it to create container which will show the content on doing curl.



```
[root@localhost ~]# podman build -t my-custom-image:v1 .
STEP 1/9: FROM docker.io/redhat/ubi9
Trying to pull docker.io/redhat/ubi9:latest...
Getting image source signatures
Copying blob 86e5ef351305 skipped: already exists
Copying config cb33f429b9 done
Writing manifest to image destination
Storing signatures
STEP 2/9: MAINTAINER    Deep Shah (deep.shah.7677@gmail.com)
--> 7786900bf56
STEP 3/9: LABEL description = “This is a custom Apache Image”
--> d2a525b0317
STEP 4/9: ENV   TRAINING=redhat
--> 88055d25a86
STEP 5/9: EXPOSE        80
--> df4c9f72f5f
STEP 6/9: RUN   yum install httpd -y
Updating Subscription Management repositories.
```

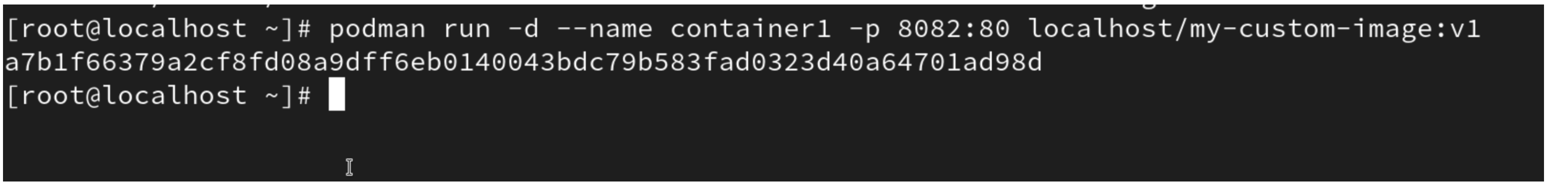
Solutions



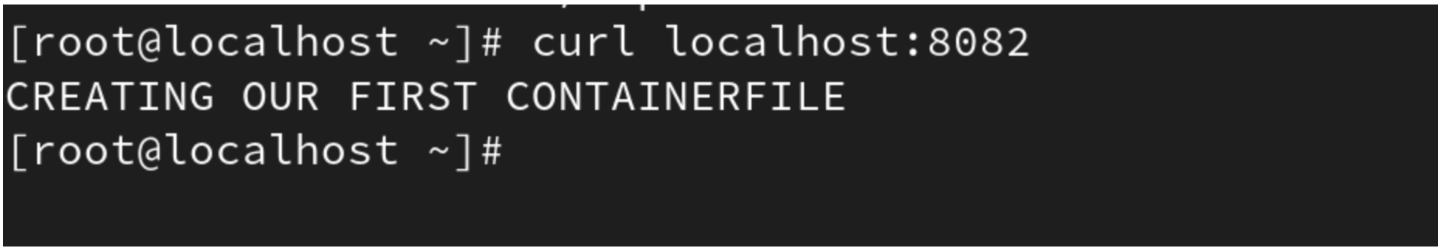
```
FROM    docker.io/redhat/ubi9
MAINTAINER      Deep Shah (deep.shah.7677@gmail.com)
LABEL   description = “This is a custom Apache Image”
ENV     TRAINING=redhat
EXPOSE  80
RUN     yum install httpd -y
COPY    source /var/www/html/index.html
ENTRYPOINT      ["/usr/sbin/httpd"]
CMD     ["-D", "FOREGROUND"]


~
~
```

Q1.

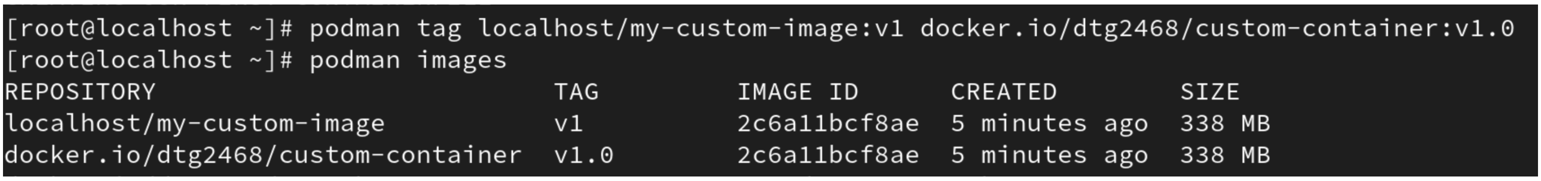


```
[root@localhost ~]# podman run -d --name container1 -p 8082:80 localhost/my-custom-image:v1
a7b1f66379a2cf8fd08a9dff6eb0140043bdc79b583fad0323d40a64701ad98d
[root@localhost ~]# 
```

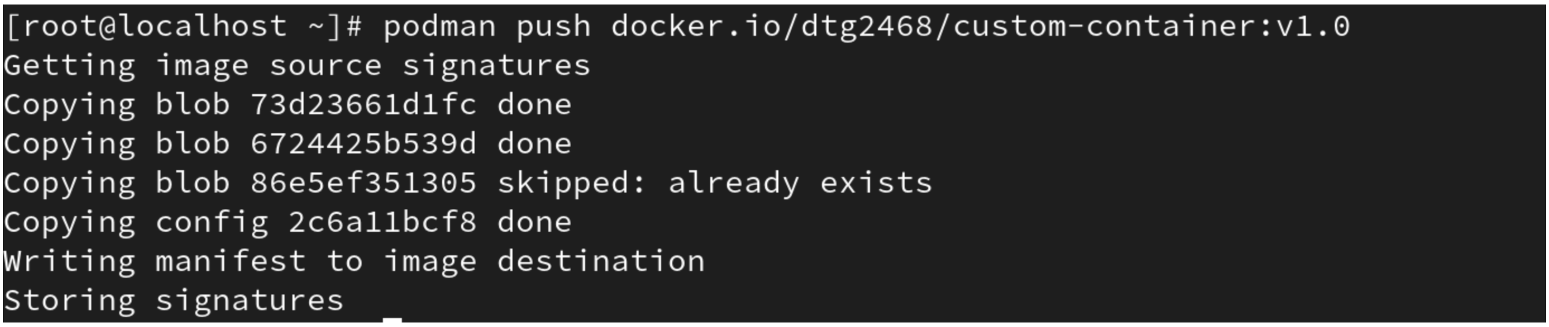


```
[root@localhost ~]# curl localhost:8082
CREATING OUR FIRST CONTAINERFILE
[root@localhost ~]#
```

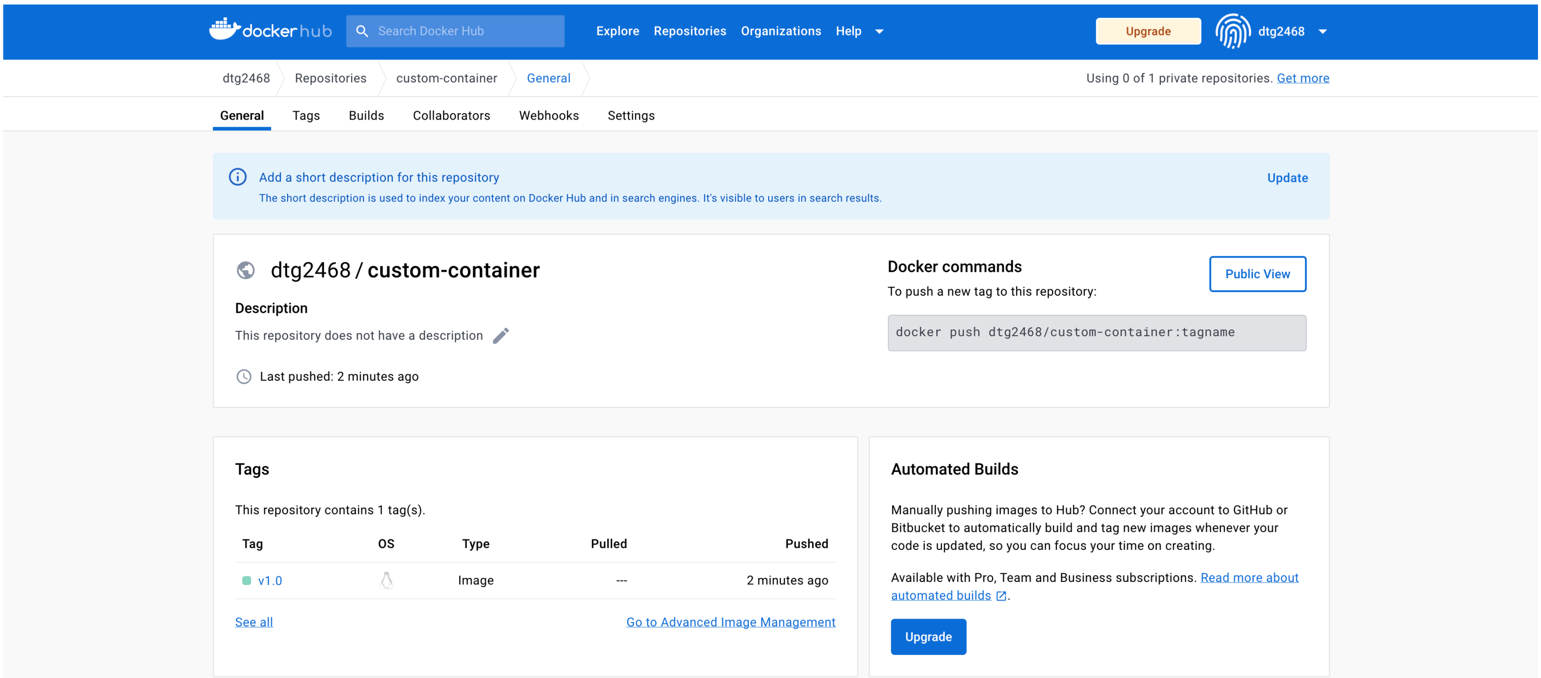
Q2.



```
[root@localhost ~]# podman tag localhost/my-custom-image:v1 docker.io/dtg2468/custom-container:v1.0
[root@localhost ~]# podman images
REPOSITORY                         TAG         IMAGE ID      CREATED        SIZE
localhost/my-custom-image          v1          2c6a11bcf8ae  5 minutes ago  338 MB
docker.io/dtg2468/custom-container v1.0        2c6a11bcf8ae  5 minutes ago  338 MB
```



```
[root@localhost ~]# podman push docker.io/dtg2468/custom-container:v1.0
Getting image source signatures
Copying blob 73d23661d1fc done
Copying blob 6724425b539d done
Copying blob 86e5ef351305 skipped: already exists
Copying config 2c6a11bcf8 done
Writing manifest to image destination
Storing signatures
```



Using 0 of 1 private repositories. Get more

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

dtg2468 / **custom-container**

Description

This repository does not have a description

Last pushed: 2 minutes ago

Docker commands

To push a new tag to this repository:

```
docker push dtg2468/custom-container:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
v1.0		Image	---	2 minutes ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.