Guided Exercise: Build Images with Advanced Containerfile Instructions

Use a simple Python application to create a Containerfile with advanced instructions.

Outcomes

You should be able to work with:

- Multistage builds.
- The `USER` instruction.
- The `WORKDIR` instruction.
- The `ENV` instruction.
- The `VOLUME` instruction.

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

```
[student@workstation ~]$ lab start custom-advanced
```

Instructions

1. Examine the exercise application.

   Go to the `/home/student/DO188/labs/custom-advanced` directory.

   ```
   [student@workstation ~]$ cd ~/DO188/labs/custom-advanced
   no output expected
   ```

   Examine the `main.py` file, which contains a basic Python application. The application reads the `numbers.txt` file, prints its content, and appends another number to the file. The application uses an environment variable to locate the `numbers.txt` file.

   ```
   ...output omitted...

   FILE = environ.get('FILE')

   ...output omitted...
   ```

   Examine the incomplete `Containerfile` file.

2. Use a multistage build to complete the `Containerfile` file.

   Add a stage that uses the `registry.ocp4.example.com:8443/redhattraining/podman-random-numbers` base image. The base image contains the `random_generator.py` script. Use the script to generate a `numbers.txt` file and copy the file to the final stage.

   Create a stage by adding a `FROM` instruction.

   ```
   FROM registry.ocp4.example.com:8443/redhattraining/podman-random-numbers as generator

   FROM registry.ocp4.example.com:8443/ubi8/python-38:1-96
   ...output omitted...
   ```

   Generate the `numbers.txt` file.

   ```
   FROM registry.ocp4.example.com:8443/redhattraining/podman-random-numbers as generator
   RUN python3 random_generator.py

   FROM registry.ocp4.example.com:8443/ubi8/python-38:1-96
   ...output omitted...
   ```

   In the second build stage, copy the `numbers.txt` file by using the `--from=generator` option of the `COPY` instructon. Use the `--chown` option to make the user `default` the new owner.

```
...output omitted...
WORKDIR /redhat

COPY --from=generator --chown=default /app/numbers.txt materials/numbers.txt
COPY main.py .
...output omitted...
```

3. Add the `FILE` environment variable to set the path of the `numbers.txt` file.

```
FROM registry.ocp4.example.com:8443/redhattraining/podman-random-numbers as generator
RUN python3 random_generator.py

FROM registry.ocp4.example.com:8443/ubi8/python-38:1-96

ENV FILE="/redhat/materials/numbers.txt"
USER default
WORKDIR /redhat

COPY --from=generator --chown=default /app/numbers.txt materials/numbers.txt
COPY main.py .

CMD python3 main.py
```

4. Build and test the image.

   In a command-line terminal, build the image from the Containerfile.

```
[student@workstation custom-advanced]$ podman build -t \
  redhat-local/custom-advanced .
...output omitted...
Successfully tagged localhost/redhat-local/custom-advanced:latest
3360...cc21
```

   Run the image.

```
[student@workstation custom-advanced]$ podman run --rm \
  --name=custom-advanced redhat-local/custom-advanced
Current content: ['17 72 97 8 32 15 63 97 57']
Writing another number...
Current content: ['17 72 97 8 32 15 63 97 57 4']
```

   The application uses the environment variable to read from the file, and write content to the file. The numbers that you get might differ because they are generated randomly, however, a `4` is always appended to the end.

5. Add a mount point to the `/redhat/materials` directory.

   In the Containerfile, add a `VOLUME` instruction.

```
...output omitted...
COPY main.py .

VOLUME /redhat/materials

CMD python3 main.py
```

6. Build and test the image.

   In your command-line terminal, build the image from the Containerfile.

```
[student@workstation custom-advanced]$ podman build -t \
 redhat-local/custom-advanced .
...output omitted...
Successfully tagged localhost/redhat-local/custom-advanced:latest
4872...0ee0
```

   Create a container from the image.

```
[student@workstation custom-advanced]$ podman run --name=custom-advanced \
  redhat-local/custom-advanced
Current content: ['46 37 98 39 70 53 81 44 59']
Writing another number...
Current content: ['46 37 98 39 70 53 81 44 59 4']
```

> **NOTE**
>
> The previous command does not have the `--rm` option. If you use the `--rm` option, then Podman removes the anonymous volumes that the container uses.

7. Read the `numbers.txt` file on your local file system.

   Inspect the container to identify the anonymous volume path on the file system.

```
[student@workstation custom-advanced]$ podman inspect custom-advanced
...output omitted...
  "Mounts": [
       {
            "Type": "volume",
            "Name": "3aa7...22ac",
            "Source": "/home/student/.local/share/containers/storage/volumes/3aa7...22ac/_data",
            "Destination": "/redhat/materials",
            "Driver": "local",
...output omitted...
```

   You can limit the output of the `podman inspect` command by using the `--format` option.

```
[student@workstation custom-advanced]$ podman inspect custom-advanced \
 --format="{{ (index .Mounts 0).Source}}"
/home/student/.local/share/containers/storage/volumes/40ed...96bf/_data
```

   Use the `Source` path to read the `numbers.txt` file from your local file system.

```
[student@workstation custom-advanced]$ cat SOURCE/numbers.txt
46 37 98 39 70 53 81 44 59 4
```

Finish

On the `workstation` machine, use the `lab` command to complete this exercise. This is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish custom-advanced
```