Guided Exercise: Accessing Containers

Use the `podman exec` and `podman cp` commands to debug and correct container configuration.

Outcomes

You should be able to:

- Use the `podman exec` to execute commands inside of a container.

- Use the `podman cp` command to copy files from and into a container.

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

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

Instructions

1. Create a new container with the following parameters:

   - Container name: `nginx`

   - Container image: `registry.ocp4.example.com:8443/redhattraining/podman-nginx-helloworld`

     Route traffic from port `8080` on your machine to port `8080` inside of the container. Use the `-d` option to run the container in the background.

     ```
     [student@workstation ~]$ podman run --name nginx -d -p 8080:8080 \
       registry.ocp4.example.com:8443/redhattraining/podman-nginx-helloworld
     Trying to pull registry.ocp4.example.com:8443/redhattraining/podman-nginx-helloworld:latest...
     ...output omitted...
     3c2b...fa84
     ```

2. In a web browser, go to http://localhost:8080. You are presented with a `404 Not Found` error page.

3. Troubleshoot the issue.

   Use the `podman cp` command to copy the `/var/log/nginx/error.log` log file in the `nginx` container to your local machine.

   In your terminal, execute the following command:

   ```
   [student@workstation ~]$ podman cp nginx:/var/log/nginx/error.log error.log
   no output expected
   ```

   View the contents of the log file.

   ```
   [student@workstation ~]$ gedit error.log
   2022/04/26 12:19:17 [error] 2#0: *2 "/usr/share/nginx/html/public/index.html" is not found (2: No such file or directory),
   client: 10.0.2.100, server: _, request: "GET / HTTP/1.1", host: "localhost:8080"
   2022/04/26 12:19:17 [error] 2#0: *2 open() "/usr/share/nginx/html/public/404.html" failed (2: No such file or directory),
   client: 10.0.2.100, server: _, request: "GET / HTTP/1.1", host: "localhost:8080"
   ```

   The server cannot read the `/usr/share/nginx/html/public/index.html` file.

   Verify the contents of the `/usr/share/nginx/html/public` directory.

   ```
   [student@workstation ~]$ podman exec nginx ls /usr/share/nginx/html/public
   ls: cannot access '/usr/share/nginx/html/public': No such file or directory
   ```

   The directory does not exist in the container.

   Verify the contents of the `/usr/share/nginx/html` directory.

   ```
   [student@workstation ~]$ podman exec nginx ls /usr/share/nginx/html
   404.html
   50x.html
   index.html
   nginx-logo.png
   poweredby.png
   ```

   The `index.html` page exists in the `/usr/share/nginx/html` directory.

4. Correct the server configuration.

   Copy the `/etc/nginx/nginx.conf` file from the container to your machine.

   ```
   [student@workstation ~]$ podman cp nginx:/etc/nginx/nginx.conf .
   no output expected
   ```

   Open the `nginx.conf` file in a text editor, such as VSCodium or Gedit.

   In the `server` directive, change the `root` parameter to the `/usr/share/nginx/html/` value.

   ```
   [student@workstation ~]$ gedit nginx.conf
   ...file omitted...
       server {
           listen       8080 default_server;
           server_name  _;
           root         /usr/share/nginx/html/;
   ...file omitted...
   ```

   Save the change in the `nginx.conf` file.

   Replace the `/etc/nginx/nginx.conf` file in the container with the modified `nginx.conf` file.

   ```
   [student@workstation ~]$ podman cp nginx.conf nginx:/etc/nginx
   no output expected
   ```

   > **NOTE**
   >
   > You can also use the `podman exec` command with the `--interactive` and `--tty` options to open an interactive shell to modify files directly in a running container. For example, you can use the `podman exec -ti nginx vim /etc/nginx/nginx.conf` command to interactively modify the `nginx.conf` file with the `vim` text editor.
   >
   > Keep in mind that changes applied to a running container are temporary and do not persist if you delete the container. For persistent changes, you must rebuild the container image. Rebuilding container images is covered elsewhere in this course.

5. Verify the functionality of the running container.

   Reload the server configuration.

   ```
   [student@workstation ~]$ podman exec nginx nginx -s reload
   no output expected
   ```

   In a web browser, go to `localhost:8080`. You are presented with the `index.html` page contents.

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 basics-accessing
```