

Guided Exercise: 01

1. Open a terminal on base machine and start a container by using the image available at `httpd:2.4`. the `-p` option allows you to specify a redirect port. In the case, Docker forwards incoming request on TCP port 8180 to the TCP port 8080.
2. Create an HTML page on the official-`httpd` container.
 - a. Access the shell of the container by using the `exec` option and create an html page.
 - b. Exit the container.
 - c. Ensure that the HTML file is reachable from the base machine VM by using `curl` command.
3. Examine the differences in the container between the image and the new layer created by the container. To do so, use the `diff` option.
4. Create a new image with the changes created by the running container.
 - a. Stop the official-`httpd` container.
 - b. Commit the change to a new container image. Use your name as the author of the changes.
 - c. List the available container images.
 - d. The new container image has neither a name, as listed in the `REPOSITORY` column, nor a tag. Tag the image with a custom name of `do180/custom-httpd`.
 - e. List the available container image again to ensure that the name and tag were applied to the correct image.
5. Publish the saved container image to the Docker registry.
 - a. To tag the image with the registry host name and port.
 - b. Run the `docker image` command to ensure that the new name has been added to the cache.
 - c. Publish the image to the private registry on `hub.docker.com`
6. Create a container from the newly published image.
 - a. Use the `docker run` command to start a new container. Use `do180/custom-httpd:v1.0` as the base image.
 - b. Use the `curl` to access the HTML page. Make use to use the port 8280. Curl <http://localhost:8280/do180.html>.
7. Delete the containers and images created in the exercise.
 - a. Use the `docker stop` command to stop the running container.
 - b. Remove the container image
 - c. Delete the exported container image.
 - d. Remove the committed container image.