## Getting started with Dockers

### Objective:

Let’s execute a few Docker commands to experiment with Dockers. You will download and install a Docker image from the Docker Hub site <https://hub.docker.com/>

Docker images are read-only templates that contain instructions for creating a container.

A Docker image is a snapshot of a library and its dependencies required inside a container needed to run.

|  |
| --- |
| 1. First, make sure "Docker Desktop Service" is running Start a Terminal session (best as Administrator)   **You can stop and start this service by typing:  net stop** com.docker.service  **net start** com.docker.service  Best to just run Docker desktop which starts the docker service and enables the docker CLI.   1. Lets try to install a Docker on your PC.  **docker run -d -p 80:80 nginx**   This command will pull an nginx image and then creates a container to run it. If the image exists, it will not download (pull) the image. It may take a minute to run this particular container!  **NGINX** is an open-source web server software used for reverse proxy, load balancing, and caching.   1. Test this docker by browsing to: <http://localhost/>   (port 80 is assumed)     1. Type the following command in the Terminal window to view all the images installed on your PC: **docker images**  Take a note of the **nginx’s** image ID. 2. Type the following command to view all the Dockers that are currently running:  **docker** **ps**   Take a note of the **nginx’s** container ID.   1. Type **docker stop** <nginx ID> Tip: Just type a few starting letters of the **nginx**’s container ID. 2. Refresh the browser at <http://localhost/> You will encounter “This site can’t be reached” 3. Start the container by typing **docker start** <nginx ID> 4. Refresh the browser at <http://localhost/> and notice the web page is fine. 5. Remove the **nginx’s** container by name – Type **docker ps** This command shows all the currently running dockers   (please note the random name created for this container! We can change it)  Type**: docker stop <docker name>**  (from the right column)  **docker rm <docker name>**  (from the right column)  Tip: You can see it is easier to remove a docker using a few letters of its ID!   1. Type: **docker images** to see all the images on your PC. 2. Remove the **nginx’s** image by typing: **docker rmi** imageid (just type just part of the image ID) 3. Type: **docker pull nginx**   To get the docker image only   1. Type **docker ps -a** and notice there is no **nginx** container 2. Practice removing this image before moving to the next section  Search Docker Hub from the command line  1. To search your docker hub repo, you must first login. Login to docker hub by typing**: docker login** 2. Search for possible images of Nginx 3. Type the following command and view the list:  **docker search nginx**      1. Pull an image down with the command  **docker pull nginx**      1. Let’s tag this image. Tag lets us edit an image and save it to our own personal registry **docker tag** **nginx** *your-Docker Hub-user name /my-nginx*      1. view the images which are saved locally  **docker images** |

1. Push this image to your Docker Hub.   
   **docker push** *your-Docker Hub-user name /my-nginx*You can now search for your copy of nginx either in the Docker hub site or in command line: **docker search** *your-Docker Hub-user name /my-nginx*

1. You can delete our local image(s)  
   **docker ps**  
   **docker rmi** nginx  
   **docker rmi** *your-Docker Hub-user name/my-nginx*
2. Let’s install Nginx with a name of **nginxDemo**But this time run it on client port 8080 instead  
   **docker run -d -p 8080:80 --name nginxDemo nginx**  
     
   -d : detached, logs aren't outputted to the terminal  
   -p : port publishing, <host port (your PC)>:<container port (nginx docker)>  
   --name : what you want to name the container, if no name is given a random name is given  
   The last word (**nginx**) is what image to build the container from

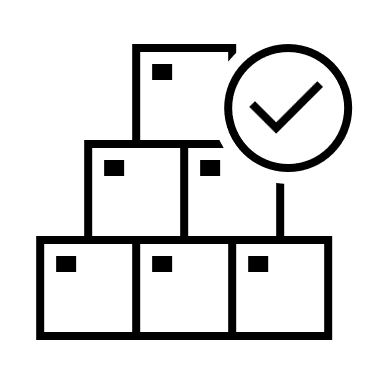
1. Check all the images running: **docker ps**
2. Try Test the Nginx docker in a browser:  
   <http://localhost>:8080

### Make changes to your Nginx container

1. Interact with the nginx container   
   You can execute a command within a running Docker container and run commands inside a container to interact with the processes running in it. The following command runs a bash command which is the Unix’s shell command (terminal)  
     
   **docker exec** -it nginxDemo bash  
   **-it** flag means interactive

1. Update package manager and install the Nano editor.   
   Type the following command in the Nginx Terminal:  
     
   **apt update  
   apt install nano -y**  
     
   **apt update:** update the package index files on the system, which contain information about available packages and their versions. It downloads the most recent package information from the sources listed in the "/etc/apt/sources.list.d" file that contains your sources

1. Modify the Nginx’s start up screen:  
   **cd /usr/share/nginx/html**   
     
   and then edit it by typing: **nano index.html**  
     
   Make some changes (style/text), save and then exit the nano editor  
   (view the bottom of the screen for how to save)  
     
   Escape the terminal interaction by typing: **exit**
2. Test the changes that you’ve made  
   <http://localhost:8080/>

Congratulations, you have successfully created a Docker, modified the code inside the docker and learned about the basic docker CLI command.