

Hands-on Workbook

Docker

Table of Contents

1. [Activities on the installation of Docker 3](#_Toc499565452)

* [Exercise 1 Installation 3](#_Toc499565453)
* [Exercise 2 Verify the Installation 4](#_Toc499565454)
* [Exercise 3 How to create a docker hub account? 6](#_Toc499565455)

1. [Activities related to images and containers 7](#_Toc499565456)

* [Exercise 4 Image Search 7](#_Toc499565457)
* [Exercise 5 Execute an Image 7](#_Toc499565458)
* [Exercise 6 Display Local Images 8](#_Toc499565459)
* [Exercise 7 Run a Container 9](#_Toc499565460)
* [Exercise 8 List a Container 10](#_Toc499565461)
* [Exercise 9 Use a container with terminal 12](#_Toc499565462)
* [Exercise 10 Run a container in detached mode 13](#_Toc499565463)
* [Exercise 11 Run a web application container 13](#_Toc499565464)
* [Exercise 12 Stop, Start, and Kill a container 14](#_Toc499565465)
* [Exercise 13 Attach/Detach client to container 15](#_Toc499565466)
* [Exercise 13.1 Attach/Detach client to container (another example) 15](#_Toc499565467)
* [Exercise 14 Execute a container 16](#_Toc499565468)
* [Exercise 15 Create a container log 17](#_Toc499565469)
* [Exercise 16 Inspect a container 17](#_Toc499565470)
* [Exercise 17 Remove a container 18](#_Toc499565471)

1. [Activities on managing images in Docker 20](#_Toc499565472)

* [Exercise 18 Commit changes 20](#_Toc499565473)
* [Exercise 19 Build Image 21](#_Toc499565474)
* [Exercise 20 Build Cache 23](#_Toc499565475)
* [Exercise 21 Build History 24](#_Toc499565476)
* [Exercise 22 CMD Command 27](#_Toc499565478)
* [Exercise 23 ENTRYPOINT 28](#_Toc499565479)
* [Exercise 24 CMD and ENTRYPOINT 29](#_Toc499565480)
* [Exercise 25 Push Image 30](#_Toc499565482)
* [Exercise 26 Add Collaborator 30](#_Toc499565483)
* [Exercise 27 Deleting Local Images 30](#_Toc499565484)

Module 4: Architecture and Installation of Docker

# Activities on the installation of Docker

## Exercise 1 Installation

|  |
| --- |
| Perform the following steps to install Docker:   * download docker engine for ubuntu 14.04 (docker 1.13.1 for xenial) * <https://apt.dockerproject.org/repo/pool/main/d/docker-engine/> * file to download (docker-engine\_1.13.1-0~ubuntu-xenial\_amd64.deb) * copy into a folder * sudo dpkg -i docker-engine\_1.13.1-0~ubuntu-xenial\_amd64.deb   OR   * $apt-get install –y docker.io//for installing docker on Ubuntu machine |

## Exercise 2 Verify the Installation

|  |
| --- |
| Perform the following steps to verify the installation:   * sudo docker version      * docker info (aufs-->advanced union file system) |

## Exercise 3 How to create a docker hub account?

|  |
| --- |
| Perform the following steps to create docker hub account:   * Sign up at docker hub * Go to <https://hub.docker.com>      * Find your email confirmation and activity your account. * Browse for some of the repositories      * Search for some images (example: java) |

Module 4: Working with Images and Containers

# Activities related to images and containers

## Exercise 4 Image Search

|  |
| --- |
| Perform the following steps to search for images:   * Search for images on the docker hub/registry * docker search java (search from client) |

## Exercise 5 Execute an Image

|  |
| --- |
| Perform the following step to execute image:   * sudo docker run hello-world |

## Exercise 6 Display Local Images

|  |
| --- |
| Perform the following steps to display local images:   * docker images//Display local images verify that the image present      * docker pull busybox (OR any other image)//Pull busybox image from docker hub   $ docker pull busybox     * Display your local images and verify your image is present $ docker images |

## Exercise 7 Run a Container

|  |
| --- |
| Perform the following steps to run a container:   * docker run ubuntu:14.04 echo “hello world” * observe the output      * docker images (ubuntu will be shown)      * docker run ubuntu:14.04 ls      * docker run ubuntu:14.04 ps -ef (display full info of all processes) * Note: Faster execution time compared to the first container that was run. Because Ubuntu:14.04 image locally available. |

## Exercise 8 List a Container

|  |
| --- |
| Perform the following steps to list a container:   * docker ps //List your running containers      * docker ps -a (all including stopped) //List the containers including the ones that have stopped      * docker ps -l (last) ) //View the last container that was started      * docker ps -aq // Listing all containers with only their short ID’s      * docker ps -lq //List the last shortID of the container started      * docker ps -a --filter “status = exited” |

## Exercise 9 Use a container with terminal

|  |
| --- |
| Perform the following steps to use a container with terminal:   1. docker run -i -t ubuntu:14.04 bash  * -i 🡪 connect to STDIN on container * -t 🡪 to get a pseudo terminal      1. Create a file called “sample” in your container      1. Try to list the file with “ls” command      1. exit to shutdown and exit a terminal     OR,  ctrl + p + q 🡪 to exit without shutdown |

## Exercise 10 Run a container in detached mode

|  |
| --- |
| Perform the following steps to run a container in detached mode:   * docker run -d ubuntu:14.04 ping localhost -c 50      * docker ps (will show the container)      * docker ps (will not show the container) after some time |

## Exercise 11 Run a web application container

|  |
| --- |
| Perform the following steps to run a web application container:   * docker run -d -P nginx (P to map container ports to host ports)      * Check the mapped port using docker ps      * Open browser and type localhost:<port> |

## Exercise 12 Stop, Start, and Kill a container

|  |
| --- |
| Perform the following steps to stop a container:   * docker run -d tomcat:8      * docker stop <containerID>      * docker start -a <containerID> (start and attach) * docker kill <containerID> * docker start <containerID> (start with same option as of initial run) * ctrl + c |

## Exercise 13 Attach/Detach client to container

|  |
| --- |
| Perform the following steps to attach/detach a container:   * docker run -d ubuntu:14.04 ping localhost -c 50      * docker attach <containerID>      * ctrl + c 🡪 for exit, ctrl + p + q 🡪 for detach * docker run -d -it ubuntu:14.04 ping localhost -c 50 ( a terminal is associated with container) * docker attach <containerID> * ctrl + c 🡪 for exit, ctrl + p + q 🡪 for detach |

## Exercise 13.1 Attach/Detach client to container (another example)

|  |
| --- |
| Check Container processes (ps command to check the processes)   * docker run -i -t ubuntu:14.04 bash      * ps -ef (will show the bash process with Pid as 1)      * ctrl + p + q |

## Exercise 14 Execute a container

|  |
| --- |
| Perform the following steps to execute a container:   * docker exec -i -t <container id> bash      * exit |

## Exercise 15 Create a container log

|  |
| --- |
| Perform the following steps to create a container log:   * docker run -d ubuntu:14.04 ping localhost -c 100 * docker logs <containerID> * docker logs -f <containerID> * ctrl + c 🡪 to stop following the log * docker logs --tail 10 <containerID> |

## Exercise 16 Inspect a container

|  |
| --- |
| Perform the following steps to inspect a container:   * docker inspect <containerID>//To display all the details about a container in JSON format      * docker inspect nginx | grep WorkingDir // Only WorkingDir information will be displayed      * docker inspect --format = ‘{{.Config}}’ <containerID> * docker inspect –format=’{{.Config.Cmd}}’ nginx      * docker inspect --format = ‘{{.Config .Image}}’ nginx |

## Exercise 17 Remove a container

|  |
| --- |
| Perform the following steps to remove a container:   * docker ps --filter = ‘status=exited’      * docker rm <containerID> |

Module 5: Managing Images

# Activities on managing images in Docker

## Exercise 18 Commit changes

|  |
| --- |
| Perform the following steps to commit changes in Docker:   * docker commit <containerID> <yourName>/myapp:1.0      * docker images      * docker run -i -t <yourName>/myapp:1.0 bash      * touch test * ls      * exit      * docker ps -a      * docker diff <containerID>      * docker commit <containerID> <yourName>/myapp:1.1 |

## Exercise 19 Build Image

|  |
| --- |
| Perform the following steps to build image in Docker:   * root@ubuntu: /home/docker5/myimage#vi Dockerfile * Content in Dockerfile: * FROM ubuntu:14.04 * RUN apt-get update * RUN apt-get install -y wget * docker build -t myimage . (. means use Dockerfile from current directory)          * docker images      * docker run myimage which wget (run and check installation of wget) |

## Exercise 20 Build Cache

|  |
| --- |
| Perform the following steps to build cache in Docker:   * docker build -t myimage . (observe the use of cache)      * modify Dockerfile to install vim   + FROM ubuntu14:04   + RUN apt-get update   + RUN apt-get install -y wget   + RUN apt-get install -y vim      * docker build -t myimage:1.0        * Edit the Dockerfile and change the sequence of wget and vim   + FROM ubuntu14:04   + RUN apt-get update   + RUN apt-get install -y vim   + RUN apt-get install -y wget * docker build -t myimage:1.0 . |

## Exercise 21 Build History

|  |
| --- |
| Perform the following steps to build history in Docker:   * docker history myimage:1.0 * note the imageID of the layer with apt-get update command      * change the Dockerfile to combine vim and wget ( RUN apt-get install -y wget vim)      * docker build –t myimage:1.0 .        * docker history myimage:1.0 |

## Exercise 22 CMD Command

|  |
| --- |
| Perform the following steps to dockerize applications using CMD:   * Add following line to end of Dockerfile * CMD [“ping”, “127.0.0.1”, “-c”, “30”]      * docker build -t <yourname>/myimage:1.0 .      * docker run <yourname>/myimage:1.0      * docker run <yourname>/myimage:1.0 echo “Hello World” (cmd is overridden) |

## Exercise 23 ENTRYPOINT

|  |
| --- |
| Perform the following steps to dockerize applications using ENTRYPOINT:   * Change the Dockerfile to * ENTRYPOINT [“ping”] (instead of CMD)      * docker build -t <yourname>/myimage:1.0 .      * docker run <yourname>/myimage:1.0 * docker run <yourname>/myimage:1.0 127.0.0.1 |

## Exercise 24 CMD and ENTRYPOINT

|  |
| --- |
| Perform the following steps to dockerize applications using CMD and ENTRYPOINT:   * Change the Dockerfile to * CMD [ “127.0.0.1”] * ENTRYPOINT [“ping”, “-c”, “30”] * docker build -t <yourname>/myimage:1.0 . * docker run <yourname>/myimage:1.0 * docker run <yourname>/myimage:1.0 localhost |

## Exercise 25 Push Image

|  |
| --- |
| Perform the following steps to push image to registry:   * docker login * docker tag myimage:1.0 <username>/myimage:1.0 * docker push <username>/myimage:1.0 verify on hub |

## Exercise 26 Add Collaborator

|  |
| --- |
| Perform the following steps to add a collaborator:   * docker pull <xyz user>/myapplication:1.0 |

## Exercise 27 Deleting Local Images

|  |
| --- |
| Perform the following steps to delete local images:   * docker rmi <username>/myapplication:1.0 |