**Lab Exercise 1- Docker Basic Commands**

**Objective**

Learn and practice basic Docker commands for managing containers and images.

**Prerequisites**

* Install Docker on your machine. You can find installation instructions for different operating systems on the official Docker website.
* Basic knowledge of the command line interface (CLI).

**Exercise Steps**

**1. Verify Docker Installation**

After installing Docker, verify that it is correctly installed by running the following command in your terminal or command prompt:

docker --version

This should return the version of Docker that is installed.

**2. Pull an Image**

Docker images are the basis of containers. Pull an official image from Docker Hub. In this exercise, we'll use the Ubuntu image.

docker pull ubuntu

**3. List Available Images**

Check that the Ubuntu image has been successfully downloaded:

docker images

This command lists all Docker images on your local machine.

**4. Run a Container**

Run a container using the Ubuntu image. The -it flag allows you to interact with the container in an interactive terminal.

docker run -it ubuntu

Once inside the container, you can run commands as if you were in a normal Ubuntu environment.

**5. Install a Package Inside the Container**

While in the Ubuntu container, update the package list and install a package (for example, curl):

apt-get update

apt-get install curl

**6. Exit the Container**

To exit the interactive shell of the container, type:

exit

This will stop the container.

**7. List Running Containers**

Check which containers are currently running:

docker ps

Since you just exited the container, it won’t show up here. To see all containers (including stopped ones), use:

docker ps -a

**8. Restart a Container**

You can restart the container you just exited. Use the container ID or name obtained from the previous command:

docker start <container\_id>

**9. Stop a Running Container**

If you want to stop a running container, you can use the following command:

docker stop <container\_id>

**10. Remove a Container**

To remove a stopped container, use the following command:

docker rm <container\_id>

**11. Remove an Image**

If you want to remove the Ubuntu image you pulled earlier, you need to first ensure that there are no containers running from that image. Use the following command:

docker rmi ubuntu

**12. Clean Up Resources**

You can remove all stopped containers and unused images with the following commands:

docker container prune

docker image prune

**13. Summary of Basic Commands**

Here’s a quick recap of the basic Docker commands you practiced:

* **Check Docker version**: docker --version
* **Pull an image**: docker pull <image\_name>
* **List images**: docker images
* **Run a container**: docker run -it <image\_name>
* **List running containers**: docker ps
* **List all containers**: docker ps -a
* **Start a container**: docker start <container\_id>
* **Stop a container**: docker stop <container\_id>
* **Remove a container**: docker rm <container\_id>
* **Remove an image**: docker rmi <image\_name>
* **Clean up resources**: docker container prune, docker image prune

**Conclusion**

By completing this lab exercise, you should now have a foundational understanding of basic Docker commands. Practice using these commands to become more comfortable with Docker and container management. You can also explore further by trying out additional Docker features, such as networking, volumes, and Docker Compose!