**Lab-1 Docker Instructions**

To get a brief introduction about docker read- <https://docs.docker.com/get-started/overview/>

Following are the steps to install Docker install on Ubuntu 20.04:

1. You can either use the lab machines or your personal computer to install and run docker.
2. The steps below have been verified on lab machines.
3. Uninstall older versions of Docker by running the following command in the shell-

sudo apt-get remove -y docker docker-engine docker.io containerd runc

*Note: It’s OK if ‘apt-get’ reports that none of these packages are installed.*

1. We will install Docker from its repository due to ease of installation. Run the following commands to set up the repository.

sudo apt-get update

sudo apt-get install -y \

apt-transport-https \

ca-certificates \

curl \

gnupg \

lsb-release \

fuse-overlayfs

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

*Note: Ignore the following warning- “gpg: WARNING: unsafe ownership on homedir '/home/\*\*/\*\*\*'”*

echo \

"deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

1. Now let’s install the latest version of Docker by running the following commands-

sudo apt-get update

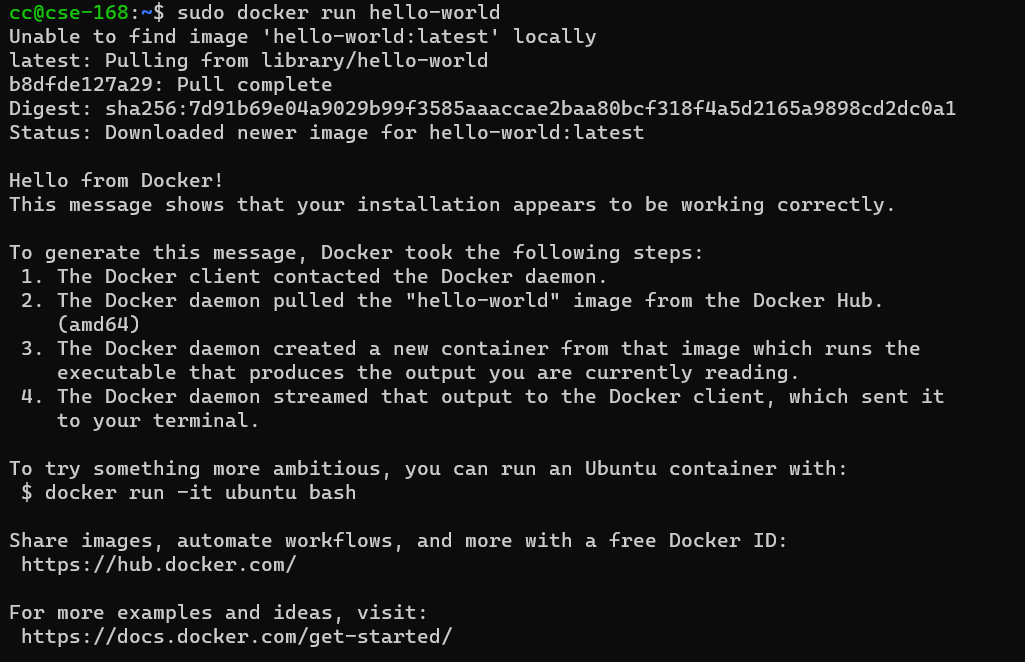
sudo apt-get install -y docker-ce docker-ce-cli containerd.io

*Note: This step might take a couple of minutes to finish.*

1. Verify whether Docker is installed correctly by running the ‘hello-world’ image as follows-

sudo docker run hello-world

The output of the command should be similar to-



If you see the above output on your screen/terminal it means that Docker is installed and running correctly. The command basically downloads a test image and runs it in a container. When the container runs, it prints a message and exits.

Remember to use *sudo* to run Docker commands. For example, if you wish to check the version of docker installed on your system, you need to run-

sudo docker --version

On my machine the following version of Docker is installed-

Docker version 20.10.8, build 3967b7d

The above instructions to install Docker on Ubuntu can be found here- <https://docs.docker.com/engine/install/ubuntu/>.

How to use gRPC docker image?

1. To avoid installing gRPC from scratch, it is better to use gRPC Docker image. This Docker image contains all the build tools and libraries ( g++ , make , protoc , and grpc ) necessary to create and run gRPC applications.

(Note: Thanks to Adam Weingram for sharing his Dockerfile and gRPC docker image with the class.)

1. Easy way to use gRPC docker image is to load it. So, first download the gRPC docker image from box-

wget -v -O /dev/shm/grpc\_image.tar.gz -L https://ucmerced.box.com/shared/static/nrltwt44xn75r8p354w0aa1lprlg9vx9

Then load the image as follows-

sudo docker load < /dev/shm/grpc\_image.tar.gz

(Other way to create a gRPC docker image using a Dockerfile. This Dockerfile is available under ‘Files’ -> ‘Lab 1 – RPC’ -> Dockerfile. After downloading this docker file run the following command- sudo docker build - < Dockerfile)

1. Run sudo docker image ls to verify if the image ‘ubuntu\_grpc’ is created.
2. Once the image is present, you can start a docker container by running the following-

sudo docker run --name grpc\_container -it ubuntu\_grpc

1. The above command launches a container and opens a bash/shell that allows you to access ‘grpc\_container’.
2. Inside the container, you will find grpc to be installed under /grpc folder.
3. Now, lets build the gRPC client and server example.

cd /grpc/examples/cpp/helloworld

mkdir -p cmake/build

pushd cmake/build

cmake -DCMAKE\_PREFIX\_PATH=$MY\_INSTALL\_DIR ../..

make -j 2

1. Now, run the server-

./greeter\_server

1. Run the client by opening another shell for the running docker container using the following command-

sudo docker exec -it grpc\_container bash

1. Now jump into the build folder of examples to run the client

cd /grpc/examples/cpp/helloworld/cmake/build

./greeter\_client

1. Now, you can modify the greeter\_server.cc and greeter\_client.cc to measure latency. To see your changes, re-run ‘make’ command.

Make changes to greeter\_client.cc and greeter\_server.cc in order to measure round trip latency and throughput.

Some docker tips-

You can copy files out of docker container into your host machine. For example, to copy the greeter\_server.cc file into the host’s current directory, use the following command-

sudo docker cp grpc\_container:/grpc/examples/cpp/helloworld/greeter\_server.cc .

If would like to save any changes you made to the docker container (for example, installed some libraries, made changes to any of the files) while running your application, run the following command-

sudo docker commit grpc\_container *<new-image-name>*

To save this image docker hub and use it again later, follow the instructions here- <https://docs.docker.com/docker-hub/>