

# IERG 4330

## Tutorial 2

# Docker

In last tutorial, we have seen the various Image can be download from the **Docker Hub**.

Docker also gives you the capability to create your own Docker images, and it can be done with the help of **Docker Files**.

A **Docker File** is a simple text file with instructions on how to build your images.

# Docker File

# for comments

**From:** from which base image you want to base your image

**Run:** Image build step, run the instructions to build the Image

**CMD:** the command the container executes by default when you launch the built image.

Dockerfile will only use the final **CMD** defined. It can also be overridden when using “docker run” with commands.

```
#This is a sample Image  
  
FROM ubuntu:latest  
  
MAINTAINER CUHK-IERG4330  
  
RUN apt-get update  
RUN apt-get install -y vim  
  
CMD echo "Image created"
```

# Docker Build

```
docker build -t ImageName:TagName dir
```

**ImageName:** name of your Image

**TagName:** tag for the Image. Usually used to define Image version

The name of your docker file must be “Dockerfile”.

# More on Docker File

**ENV:** sets the environment variable <key> to the value <value>

**EXPOSE:** informs Docker that the container listens on the specified network ports at runtime. You can specify whether the port listens on TCP or UDP.

**WORKDIR:** sets the working directory

```
ENV PATH=/miniconda/bin:$PATH
EXPOSE 8888/tcp
WORKDIR /home
```

# More on Docker

## Expose port(-p, --expose)

```
$ docker run -p 127.0.0.1:80:8080/tcp ubuntu bash
```

This binds port **8080** of the container to TCP port **80** on **127.0.0.1** of the host machine

## Mount volume (-v)

Mounts the current working directory into the container.

```
docker run -v $dir:/home
```

# References

<https://docs.docker.com/engine/reference/builder/>

[https://www.tutorialspoint.com/docker/building\\_docker\\_files.htm](https://www.tutorialspoint.com/docker/building_docker_files.htm)

<https://docs.docker.com/engine/reference/commandline/run/>

Thank you!