

# Dockerfiles: Building Images

---

- FROM: where to pull a base image from. It will be the base of your particular image.
- CMD: Instruction for running a shell command inside the container, or argument to another command
- RUN: Executes a command to help build image
- EXPOSE: Opening Network Ports
- VOLUME: allow you to specify a disk share
- COPY: used for copy files into the image from local disk
- LABEL: Add metadata as key-value pair
- ENV: Environment variable to a container
- ENTRYPOINT: Should be used almost at the end of the file to specify which executable will run when your container starts.
  - the following CMD entry after ENTRYPOINT should then pass options to the executable if required.

```
FROM debian:latest
LABEL maintainer='Your Name'
ENTRYPOINT ['/bin/ping']
CMD ['google.com']
```

```
# Building the dockerfile
docker build -t username/dockerping:latest .
# Docker_ID/Repository:Tag
```

1. **docker build** expects a path to where it can find the Dockerfile that it will use to build the image. First docker downloads the base image, which came from FROM instruction in the Dockerfile. Next, docker ran all the commands and instructions in the dockerfile to create the resulting image.

```
# To upload it to Docker Hub we need exec ``docker login`` command.
docker login
docker push Docker_ID/Repository:Tag
```

```
# 1. Docker Engine downloads the base image from the specified path
FROM debian:latest
LABEL maintainer='John Doe'
# 2. Docker engine runs ENTRYPOINT command /bin/ping
ENTRYPOINT ['/bin/ping']
# 3. Docker uses www.google.com as the default command argument.
CMD ['www.google.com']
```

```
# The CMD instruction can be overridden by supplying an option on the
command line
```

```
FROM debian:latest
LABEL maintainer=name
RUN apt update && \
    apt install -y traceroute && \
    apt install -y curl && \
    apt clean
ENTRYPOINT ['/bin/bash']
```

```
docker build -t user/nettools nettools/
#           #ImageBase      #DirectoryPath
# -t option for tag
```

## Override Entry Point

```
docker run --entrypoint /usr/bin/curl -it username/nettools docker.com
```

## Dockerfile Default File Name

- **-f** option provide the name of the file when dockerfile has a diferent name tha 'Dockerfile'

```
docker build -t username/nettools:AlternativeTag -f
DockerfileAlternativeNam&& \e
```

## Exercise: Build and push an image

1. Create an account on Docker Hub to host your own images: <https://hub.docker.com/signup>

## Creating a Dockerfile

```
mkdir tempbuild
cd tempbuild
nano Dockerfile
```

```
# Creating the Dockerfile
FROM debian:stable
LABEL maintainer="ENTER_YOUR_NAME_HERE"
```

```
RUN apt update && apt install -y nano && apt clean  
CMD ["/bin/bash"]
```

```
docker build -t YOUR_DOCKER_ID/nanotest:latest .           # build image  
from dockerfile  
docker images                                             # see if nanotest  
image is present  
docker run --name nanotest -it YOUR_DOCKER_ID/nanotest    # run image  
which nano                                              # testing if nano  
is available  
exit
```

```
# Pushing the image to Docker Hub  
docker login  
docker push YOUR_DOCKER_ID/nanotest:latest
```