# Section 8 - Build Images - The Dockerfile Basics

## 2 Build Docker Images

# docker build

- Use the `docker build` command to build an image from a Dockerfile and a context.
- Usage: docker build [OPTIONS] PATH
- The most common option is `-t` to specify a Name and optionally a tag in the *name:tag* format.
- The `PATH` argument defines the context of the build, it is usually set to "." (current working directory) and by default will search for a file named Dockerfile.



# docker build - example (1a)

- In the following example we are going to use the official nginx Dockerfile, available in the `resources/dockerfile-sample-2` directory, to build a custom image:

```
# cd resources/dockerfile-sample-2/
# ls
Dockerfile

# docker build -t custom_nginx .
Sending build context to Docker daemon  6.144kB
Step 1/9 : FROM debian:stretch-slim
stretch-slim: Pulling from library/debian
d599a449871e: Pull complete
Digest: sha256:1dbbf9306be70a879f9f1eac520b2b1f9c4fed55fcb38202d0da64ab05f
Status: Downloaded newer image for debian:stretch-slim
 ---> 2b343cb3b772
Step 2/9 : LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.
 ---> Running in 11ef9991d028
Removing intermediate container 11ef9991d028
 ---> 5f2b304a6651
```

# docker build - example (1aa)

```
Step 5/9 : RUN set -x   && apt-get update       && apt-get install --no-ins
Removing intermediate container 01d252db70e4
 ---> f6670bd7f563
Step 6/9 : RUN ln -sf /dev/stdout /var/log/nginx/access.log     && ln -sf /
 ---> Running in 5eead82e4602
Removing intermediate container 5eead82e4602
 ---> a7b9597bfb59
Step 7/9 : EXPOSE 80
 ---> Running in b7cea4d2f0c6
Removing intermediate container b7cea4d2f0c6
 ---> 67280e8db9a8
Step 8/9 : STOPSIGNAL SIGTERM
 ---> Running in 310974324a67
Removing intermediate container 310974324a67
 ---> f6d6c317a6fe
Step 9/9 : CMD ["nginx", "-g", "daemon off;"]
 ---> Running in 609cdb444b4e
Removing intermediate container 609cdb444b4e
 ---> e64fcdb04016
Successfully built e64fcdb04016
Successfully tagged custom_nginx:latest
```



# docker build - example (1b)

- Each Step corresponds to a line in the Dockerfile.
- Each Step will create an image layer that we can later refer to it by the hash number e.g. `---> c08899734c03`.
- The image with all the related layers are stored to the local cache.



# docker build - example (1c) - local cache

- The next time that the build process takes place, before actually executing every single step, it will search in the local cache if any related image layer already exists.

```
# docker build -t custom_nginx .
Sending build context to Docker daemon  6.144kB
Step 1/9 : FROM debian:stretch-slim
 ---> 2b343cb3b772
Step 2/9 : LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.
 ---> Using cache
 ---> 5f2b304a6651
Step 3/9 : ENV NGINX_VERSION 1.15.12-1~stretch
 ---> Using cache
 ---> 1c30445c11d2
Step 4/9 : ENV NJS_VERSION   1.15.12.0.3.1-1~stretch
 ---> Using cache
 ---> 21c13a34ef99
...
```

# docker build - example (1cc) - local cache

- During the build process the "Docker engine" will understand for which layers of the image is possible to use the build cache and when the build cache cannot be used because:
  1. there are changes in the Dockerfile or
  2. there are changes in the files that are included in the image.
- During the build process we can see from the output `---> Using cache` when the cached is used.

# Dockerfile - order of the commands

- The order of the instructions specified in the Dockefile is important.
- Instructions that usually will cause a layer to change should be placed at the end of the Dockefile.
- For example, a command that adds our application code should be placed at the end of the Dockefile file, since it is the one that changes more often.
- Instructions that usually build the same layer should be placed on the top.

# docker image default tag - *latest*

- By default the created image will be tagged as latest

```
# docker image ls
REPOSITORY                  TAG                 IMAGE ID            CREA
custom_nginx                latest              916effbcb643        5 mi
nginx                       1.15                27a188018e18        12 c
gerassimos/nginx            latest              27a188018e18        12 c
gerassimos/nginx            test1               27a188018e18        12 c
...
```