



Docker – Building images

Containerization – Key terms.

- Image: a standalone, executable package that includes everything needed to run a piece of software (application code + dependencies + system tools + libraries + configuration files).
 - The entire filesystem and metadata (e.g. environment variables) to run an app.
 - An immutable template for a containers.
- Container: The runtime instance of an image.
 - An OS process isolated from the rest of the system through abstractions created by the OS.
 - A container's filesystem comes from an image.

Build A Docker Image.

- Steps:

1. Create a Dockerfile (set of commands/directives), e.g.

```
FROM nginx
```

```
COPY index.html /usr/share/nginx/html/
```

2. Build the image and give it a name, e.g.

```
$ docker build -t my-nginx-website:1.0 . (the . at the end means  
current folder)
```

- Important:

1. Place the Dockerfile in project's base folder
2. Each command in the file creates a new temporary container
3. Every creation step is cached, so repeated builds are fast.

DockerFile instructions.

- FROM
 - Sets the Base Image for the newly created image
e.g. FROM nginx:15:04
- COPY - copy files from the host to to the image filesystem, i.e. COPY <src> <dest>
 - Source can contain wildcards
 - dest is created if it does not exist.
 - Example:
COPY ./service.config /etc/service/
COPY ./service.config /etc/service/myconfig.conf
COPY ./*.config /etc/service/
COPY ./cfg/ /etc/service/

DockerFile instructions.

- CMD - specifies the default start-up command to execute at container runtime.
 - Form: CMD ["executable","param1","param2"]
 - Example: CMD ["nginx", "-g", "daemon off;"]
 - If supplied, the docker run arguments overwrite those of the CMD.
docker run <options> image executable params
- RUN <command> - execute command(s) inside the container during the build process.
 - It is common to tie related commands together into one RUN command, using && (and).
RUN apt-get update && \
apt-get install -y ca-certificates && \
rm -rf /var/lib/apt/lists/*

DockerFile instructions.

- ENV - sets environment variables which are present during building and remain in the final image. Form:

ENV <key1>=<value> <key2>=<value> ...

- They can be overwritten at container runtime with the -e option:

\$ docker run -e key1=new_value image-name

e.g. \$ docker run \

-e message='The answer is' -e answer=42 \

ubuntu:latest \

bash -c 'echo \$message \$answer'

The answer is 42

DockerFile instructions.

- ADD - the same as COPY with the following additions:
 1. If src is a URL, the file is downloaded, e.g.
ADD <https://download.elasticsearch.org/elasticsearch/elasticsearch-1.4.4.tar.gz> /es/
RUN cd /es && tar xvfz elasticsearch-1.4.4.tar.gz
 2. If src is a local tar archive, it will be extracted to destination, e.g.
ADD configs.tar.gz /etc/service/

DockerFile instructions.

- WORKDIR path - set path as default directory inside the container, e.g.
WORKDIR /usr/app
COPY ./src . # Copy project src folder to /usr/app/src