

Dockerfile: Basics



docker®

Created by Akshay Suresh



Akshay Suresh

DevOps Engineer @ Hashedin by Deloitte

YCombinator SUS'21

 @akshayvsuresh

 akshayvsuresh27@gmail.com



docker®

What is a Dockerfile ?

A Dockerfile is a script that automatically creates containers on the Docker platform. A Dockerfile is basically a text document that contains all the commands a user could call on the command line to assemble an image. The Docker platform runs natively on Linux and allows developers to build and run containers, self-contained applications or systems with no dependencies on the underlying infrastructure. Built upon the resource-isolation features of the Linux kernel, Docker helps developers and system administrators port applications across different systems and machines by running them inside containers.



Docker Workflow

Dockerfile



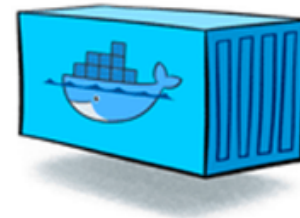
→ Build →

Image



→ Run →

Container



docker.

HOW TO BUILD AND RUN A DOCKER IMAGE



- Create a directory to store your files
`mkdir sample-docker`
- Create a Dockerfile
`touch Dockerfile`
- Copy the code snippet in slide to Dockerfile
- Build the docker image
`docker build -t sample-docker .`
- Run the docker image
`docker run -dp 3000:3000 getting-started`

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python2 g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

FROM

Usage:

FROM <image>

FROM <image>:<tag>

FROM <image>@<digest>

- **FROM must be the first non-comment instruction in the Dockerfile.**
- **FROM can appear multiple times within a single Dockerfile in order to create multiple images. Simply make a note of the last image ID output by the commit before each new FROM command.**

docker



@akshayvsuresh

MAINTAINER

Usage:

MAINTAINER <name>

- The **MAINTAINER** instruction allows you to set the **Author** field of the generated images.

docker



@akshayvsuresh

RUN

Usage:

RUN <command>

RUN ["<executable>", "<param1>", "<param2>"] (exec form)

- The exec form makes it possible to avoid shell string munging, and to RUN commands using a base image that does not contain the specified shell executable.
- The default shell for the shell form can be changed using the SHELL command.

docker



@akshayvsuresh

CMD

Usage:

CMD ["<executable>","<param1>","<param2>"]

CMD ["<param1>","<param2>"]

CMD <command> <param1> <param2> (shell form)

- The exec form makes it possible to avoid shell string munging, and to RUN commands using a base image that does not contain the specified shell executable.
- The default shell for the shell form can be changed using the SHELL command.

EXPOSE

Usage:

EXPOSE <port> [<port> ...]

- **Notifies Docker that the container listens on the specified network port(s) at runtime.**
- **EXPOSE does not make the ports of the container accessible to the host.**

docker



@akshayvsuresh

ENV

Usage:

ENV <key> <value>

ENV <key>=<value> [<key>=<value> ...]

- The ENV instruction sets the environment variable <key> to the value <value>.
- The value will be in the environment of all “descendant” Dockerfile commands and can be replaced inline as well.

docker



@akshayvsuresh

ADD

Usage:

ADD <src> [<src> ...] <dest>

ADD ["<src>", ... "<dest>"]

- **Copies new files, directories, or remote file URLs from <src> and adds them to the filesystem of the image at the path <dest>.**
- **<src> may contain wildcards and matching will be done using Go's filepath.Match rules.**

docker



@akshayvsuresh

COPY

Usage:

```
COPY <src> [<src> ...] <dest>
```

```
COPY ["<src>", ... "<dest>"]
```

- Copies new files or directories from <src> and adds them to the filesystem of the image at the path <dest>.
- <src> may contain wildcards and matching will be done using Go's filepath.Match rules.
- <src> must be relative to the source directory that is being built (the context of the build).
- <dest> is an absolute path, or a path relative to WORKDIR.

ENTRYPOINT

Usage:

ENTRYPOINT ["<executable>", "<param1>", "<param2>"] (exec form, preferred)

ENTRYPOINT <command> <param1> <param2>

- Allows you to configure a container that will run as an executable.
- Command line arguments to docker run <image> will be appended after all elements in an exec form ENTRYPOINT and will override all elements specified using CMD.

docker



REFERENCES

- <https://docs.docker.com/engine/reference/builder/>
- <https://medium.com/swlh/understand-dockerfile-dd11746ed183>
- https://kapeli.com/cheat_sheets/Dockerfile.docset/Contents/Resources/Documents/index

docker



@akshayvsuresh



Thanks you !