Docker Certified Associate (DCA)

Create by: **Hadi Tayanloo**

Phone : **+**98–912–8387233

Linkedin: linkedin. com/in/htayanloo/

Session 4

- Docker Voulme
-
- Advanced Docker File[add, copy, volumes, expose, entrypoint, etc)**+ Lab**
- 5 lab DockerFile

DockerFile :: volume

- docker volume create --name vol1
- ll /var/lib/docker/volumes/
- docker run -it -v vol1:/root:ro centos:latest
- Note: Path_in_Docker:Path_in_Container:Permision

DockerFile :: volume

- docker inspect Container_id | grep -i mounts -A 11

DockerFile :: volume

- 1- base centos 7
  2- name mycontainer
  3- create volume /root host to /data container
- 4- create file and import ping 4.2.2.4 to file
  4- run mycontainer container with the above
- 5- view online file in container and docker host
- 6- pause and unpause new container
  7- test it…

DockerFile :: volume

- docker run -itd --name mycontainer -v /root:/data centos:latest /bin/bash -c 'ping 4.2.2.4 > /data/ping.txt'

Docker File

- Below are some dockerfile commands you must know:
- **FROM**
- The base image for building a new image. This command must be on top of the dockerfile.
- **MAINTAINER**
- Optional, it contains the name of the maintainer of the image.
- **RUN**
- **ADD**
- **ENV**
- **CMD**
- **ENTRYPOINT**
- Define the default command that will be executed when the container is running.
- **WORKDIR**
- This is directive for CMD command to be executed.
- **USER**
- Set the user or UID for the container created with the image.
- **VOLUME**
- Enable access/linked directory between the container and the host machine.

Dockerfile :: Simple Build

- docker build .

Dockerfile :: Dockerfile anywhere in your file system

- $ docker build -f /path/to/a/Dockerfile .

Dockerfile :: tag image

- docker build -t shykes/myapp .
- $ docker build -t shykes/myapp:1.0.2 -t shykes/myapp:latest .
-

DockerFile :: FROM

- FROM <image> [AS <name>]
- OR
- FROM <image>[:<tag>] [AS <name>]
- OR
- FROM <image>[@<digest>] [AS <name>]

DockerFile :: ARG

- ARG CODE_VERSION=latest
- FROM base:${CODE_VERSION}
- CMD /code/run-app
- FROM extras:${CODE_VERSION}
- CMD /code/run-extras

- ARG VERSION=latest
- FROM busybox:$VERSION
- ARG VERSION
- RUN echo $VERSION > image_version

DockerFile :: RUN

- RUN <command> (*shell* form, the command is run in a shell, which by default is /bin/sh -c on Linux or cmd /S /C on Windows)
- RUN ["executable", "param1", "param2"] (*exec* form)

DockerFile :: RUN

- RUN /bin/bash -c 'source $HOME/.bashrc; \
- echo $HOME'

- RUN /bin/bash -c 'source $HOME/.bashrc; echo $HOME'

DockerFile :: LABEL

- LABEL <key>=<value> <key>=<value> <key>=<value> ...

- LABEL "com.example.vendor"="ACME Incorporated"
- LABEL com.example.label-with-value="foo"
- LABEL version="1.0"
- LABEL description="This text illustrates \ that label-values can span multiple lines."

DockerFile :: EXPOSE

- EXPOSE <port> [<port>/<protocol>...]
- By default, EXPOSE assumes TCP. You can also specify UDP:
- EXPOSE 80/udp
- docker run -p 80:80/tcp -p 80:80/udp ...
-

DockerFile :: ENV

- ENV <key> <value>
- ENV <key>=<value> …

- ENV myName="John Doe" myDog=Rex\ The\ Dog \ myCat=fluffy

- ENV myName Ali
- ENV myDog Rex
- ENV myCat Janson

DockerFile :: ADD

- ADD [--chown=<user>:<group>] <src>... <dest>
- ADD [--chown=<user>:<group>] ["<src>",... "<dest>"] (this form is required

- ADD hom* /mydir/ # adds all files starting with "hom"
- ADD hom?.txt /mydir/ # ? is replaced with any single character, e.g., "home.txt”

- ADD test relativeDir/ # adds "test" to `WORKDIR`/relativeDir/
- ADD test /absoluteDir/ # adds "test" to /absoluteDir/

DockerFile :: ADD

- ADD --chown=55:mygroup files* /somedir/
- ADD --chown=bin files* /somedir/
- ADD --chown=1 files* /somedir/
- ADD --chown=10:11 files* /somedir/

DockerFile :: COPY

- COPY [--chown=<user>:<group>] <src>… <dest>
- COPY [--chown=<user>:<group>] ["<src>",… "<dest>"] (this form is required

- COPY hom* /mydir/ # adds all files starting with "hom"
- COPY hom?.txt /mydir/ # ? is replaced with any single character, e.g., "home.txt”

- COPY test relativeDir/ # adds "test" to `WORKDIR`/relativeDir/
- COPY test /absoluteDir/ # adds "test" to /absoluteDir/

DockerFile :: CMD

- CMD ["executable","param1","param2"] (*exec* form, this is the preferred form)
- CMD ["param1","param2"] (as *default parameters to ENTRYPOINT*)
- CMD command param1 param2 (*shell* form)

- Note: **There can only be one CMD** instruction in a Dockerfile. If you list more tha
  one CMDthen only the **last CMD** will take effect.

DockerFile :: ENTRYPOINT

- ENTRYPOINT ["executable", "param1", "param2"] (*exec* form, preferred)
- ENTRYPOINT command param1 param2 (*shell* form)

- FROM debian:stable
- RUN apt-get update && apt-get install -y --force-yes apache2
- EXPOSE 80 443
- VOLUME ["/var/www", "/var/log/apache2", "/etc/apache2"]

  ENTRYPOINT ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]

DockerFile :: USER

- USER <user>[:<group>]
- or
- USER <UID>[:<GID>]

- **Warning**: When the user doesn't have a primary group then the image (or the next instructions) will be run with the root group.

DockerFile :: WORKDIR

- WORKDIR /path/to/workdir
-

- **Warning**: When the user doesn't have a primary group then the image (or the next instructions) will be run with the root group.

DockerFile :: scratch

- https://github.com/htayanloo/Docker_Centos_Base_Image.git