

Docker contd

Containerizing Applications

- Dockerfile is an instruction based approach to create docker images.
- In this approach we create a file with name **Dockerfile**
- Dockerfile contains series of instructions

```
<INSTRUCTION> <VALUE>
```

- [Refer Here](#) for all the dockerfile instructions

Most widely used instructions

- FROM: This instruction specifies the base image
- RUN: This instruction executes commands as part of image building
- EXPOSE: This instruction specifies the ports to be exposed
- CMD: This instruction will have command that is used when container is started
- LABEL: This instruction is used to add metadata

command to build the docker image

- cd in to folder where you have Dockerfile
- command:

```
docker image build -t <image-name>:<tag> .
```

Dockerfile and image building

- The Dockerfile which we have written is

```
# take amazon corretto 17 as a base image
FROM amazoncorretto:17
# add metadata
LABEL author="shaikkhajaibrahim"
LABEL project="lt-learning"
# download spring petclinic
RUN curl -O https://khajareferenceapps.s3.ap-south-1.amazonaws.com/spring-petclinic-3.2.0-SNAPSHOT.jar
# expose 8080 port as spring petclinic needs 8080
EXPOSE 8080
# command to start the application
CMD ["java", "-jar", "spring-petclinic-3.2.0-SNAPSHOT.jar"]
```

- Build the image

```

ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker image build -t spc:1.0 .
[+] Building 12.1s (6/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 468B
=> [internal] load metadata for docker.io/library/amazoncorretto:17
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/2] FROM docker.io/library/amazoncorretto:17
=> [2/2] RUN curl -O https://khajareferenceapps.s3.ap-south-1.amazonaws.com/spring-petclinic-3.2.0-SNAPS 11.5s
=> exporting to image
=> => exporting layers
=> => writing image sha256:e80c428f90bfb16b5f2323c3a80270cba5e0bf3ec330ea2bf507fbd8c81adbdf
=> => naming to docker.io/library/spc:1.0
ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker image ls
REPOSITORY          TAG          IMAGE ID      CREATED        SIZE
spc                  1.0          e80c428f90bf 9 seconds ago  525MB
amazoncorretto      17           33387c4b66b5 5 weeks ago   465MB
trail               1.0          2714d3b9bf8b 5 weeks ago   465MB
ubuntu@ip-172-31-22-23:~/spring-petclinic$

```

- Lets try creating the container with image

```
docker container run -d --name spc1 -P spc:1.0
```

```

ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker container run -d --name spc1 -P spc:1.0
8bde0b8b36e9b888b85fe1c43abc7a670d157992a3b7c702fa4383edff4248fe
ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
8bde0b8b36e9   spc:1.0    "java -jar spring-pe..." 6 seconds ago  Up 5 seconds  0.0.0.0:32768->8080/tcp, :::3276
8->8080/tcp    spc1
ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker container rm -f spc1
spc1
ubuntu@ip-172-31-22-23:~/spring-petclinic$

```

Port-forwarding in containers

- Applications running in containers needs to be accessed externally.
- Generally containers will be running in an internal network

- To access application in container we use port forwarding



- Port forwarding can be done in two ways
 - static port forwarding: `docker container run -p 10000:8080 -d spc:1.0` In this case we are forwarding 8080 port of application to 10000 port on vm/host. users can access application by using `ip address host on 10000 => http://192.168.10.11:10000`
 - dynamic port forwarding: in this docker engine will expose the application port (specified in Dockerfile) to available port on host(ec2/vm).

Passing values while building the image

- ARG: [Refer Here](#) This instruction is used to pass values that effect the image building

```

ubuntu@ip-172-31-22-23:~/spring-petclinic$ docker image build --build-arg "DOWNLOAD_URL=https://khajareferenceapps.s3.ap-south-1.amazonaws.com/spring-petclinic-3.2.0-SNAPSHOT.jar" -t spc:1.1 .
[+] Building 12.2s (6/6) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.0s
=> => transferring dockerfile: 516B                                              0.0s
=> [internal] load metadata for docker.io/library/amazoncorretto:17             0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> CACHED [1/2] FROM docker.io/library/amazoncorretto:17                      0.0s
=> [2/2] RUN curl -O https://khajareferenceapps.s3.ap-south-1.amazonaws.com/spring-petclinic-3.2.0-SNAPS 11.6s
=> exporting to image                                                            0.4s
=> => exporting layers                                                            0.4s
=> => writing image sha256:429a540aee3464a223eed53bbceca414a08cc3bfe5ced1f3dc7aa66a3c21c68b 0.0s
=> => naming to docker.io/library/spc:1.1                                       0.0s
ubuntu@ip-172-31-22-23:~/spring-petclinic$

```