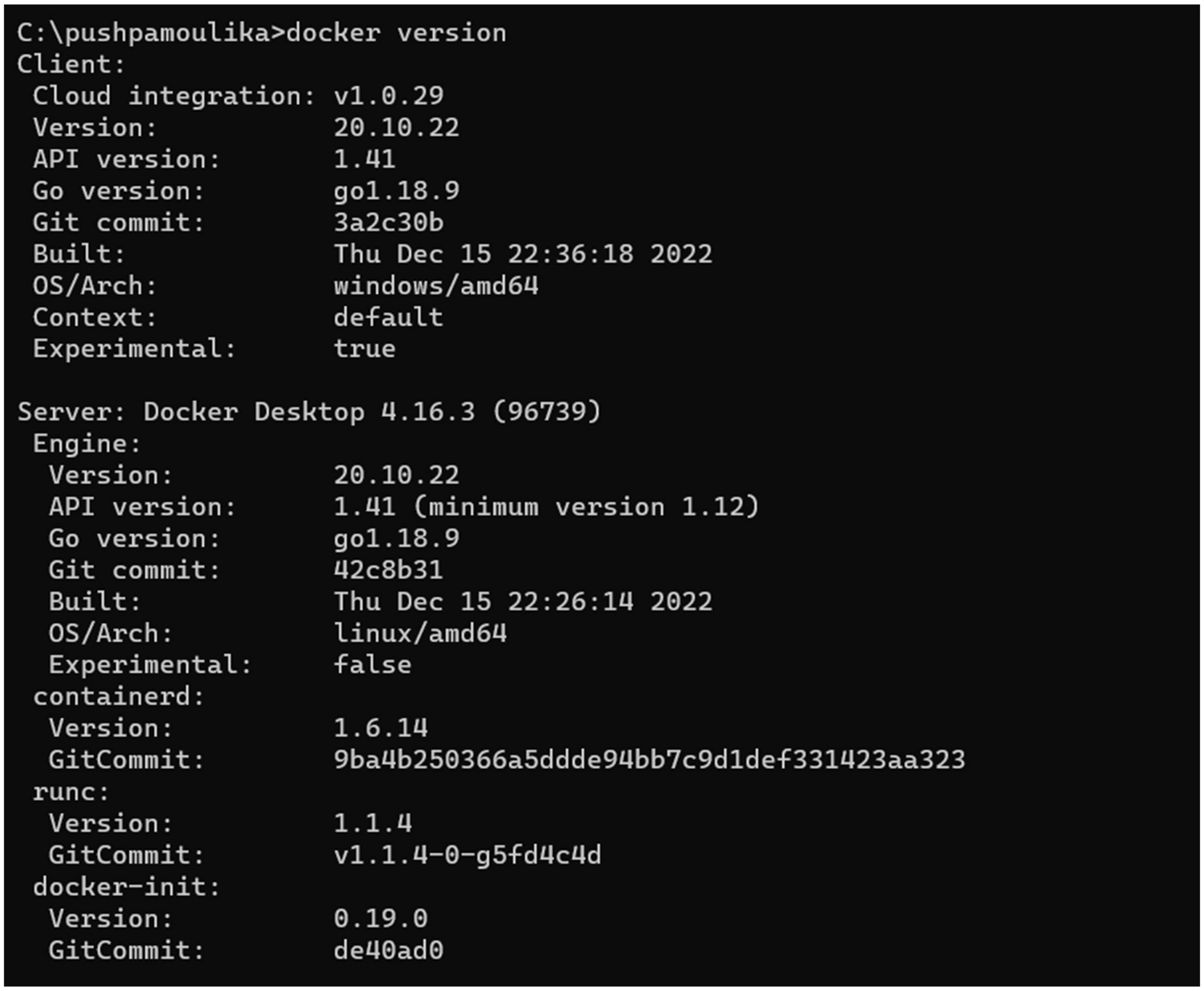
Q1) Pull any image from the docker hub, create its container, and execute it showing the output.

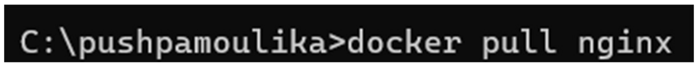
Docker is a centralized platform to create, run, execute and deploy applications on server. Docker packages software into centralized unites called containers. Docker uses the kernel of the host operating system and we can pull and push images into server known as docker hub by using an intermediate known as docker daemon.

We can know the version of docker by using docker version command



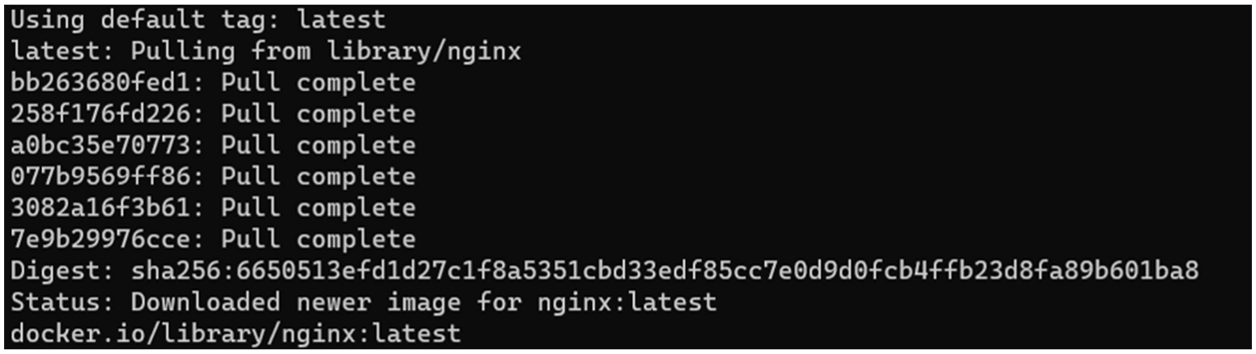
```
C:\pushpamoulika>docker version
Client:
 Cloud integration: v1.0.29
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:36:18 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.16.3 (96739)
 Engine:
  Version:          20.10.22
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.18.9
  Git commit:       42c8b31
  Built:            Thu Dec 15 22:26:14 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.14
  GitCommit:        9ba4b250366a5ddde94bb7c9d1def331423aa323
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

We can download a image by using docker pull <image name>. Docker pull downloads the image from docker hub but does not execute it.



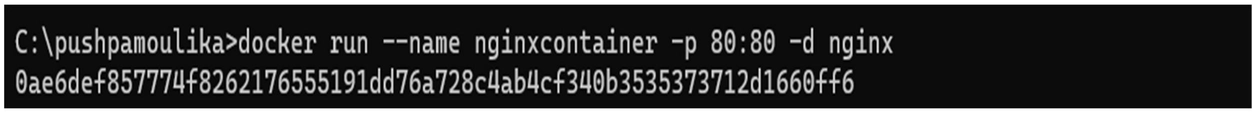
```
C:\pushpamoulika>docker pull nginx
```

Let’s download an image called nginx from the hub. After successful download of image, the terminal shoes the following lines.



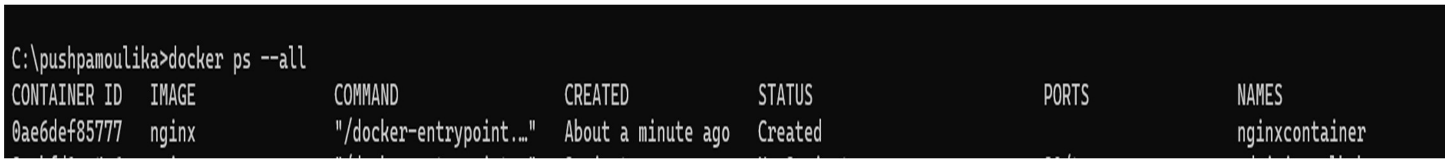
```
Using default tag: latest
latest: Pulling from library/nginx
bb263680fed1: Pull complete
258f176fd226: Pull complete
a0bc35e70773: Pull complete
077b9569ff86: Pull complete
3082a16f3b61: Pull complete
7e9b29976cce: Pull complete
Digest: sha256:6650513efd1d27c1f8a5351cbd33edf85cc7e0d9d0fcb4ffb23d8fa89b601ba8
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

To create a container for an image and expose it to port 80 using docker run –name nginxcontainer -p 80:80 -d nginx



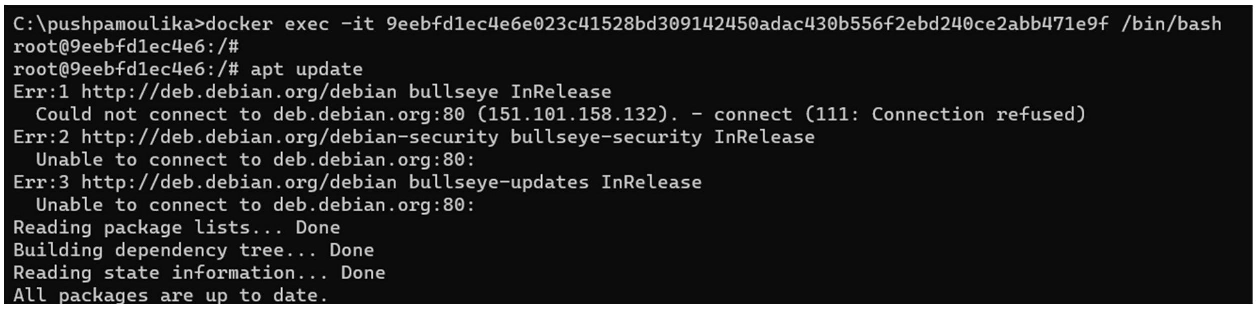
```
C:\pushpamoulika>docker run --name nginxcontainer -p 80:80 -d nginx
0ae6def857774f8262176555191dd76a728c4ab4cf340b3535373712d1660ff6
```

We can view the container by using the command docker ps -all



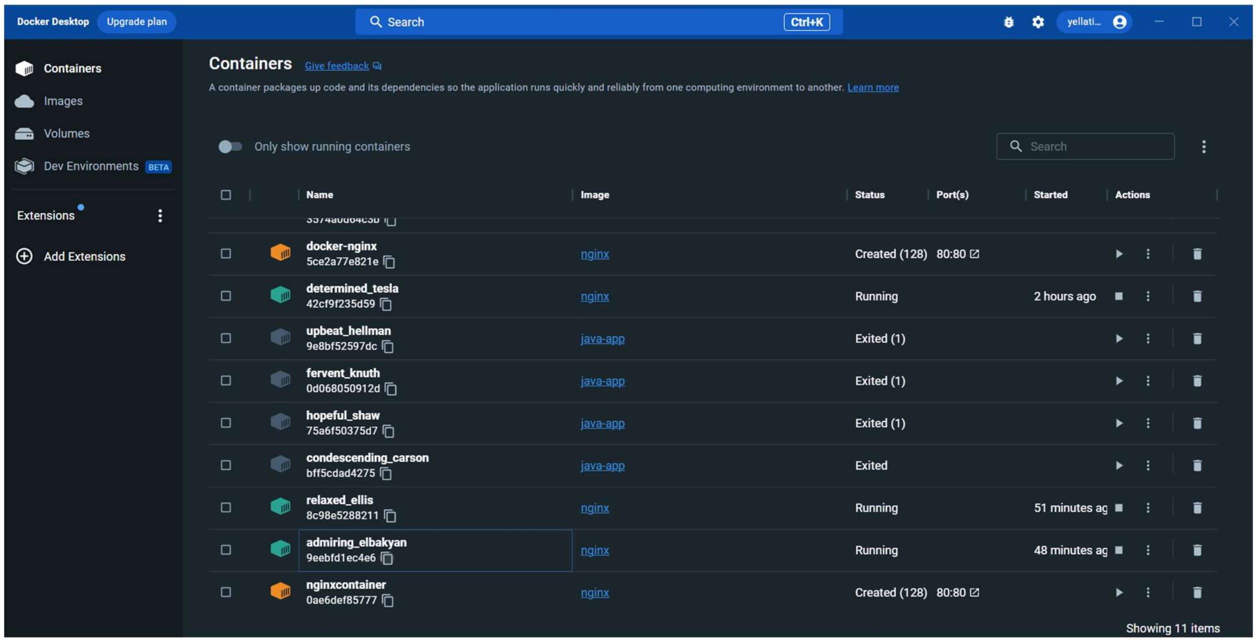
```
C:\pushpamoulika>docker ps --all
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS      PORTS     NAMES
0ae6def85777   nginx     "/docker-entrypoint.…"   About a minute ago   Created               nginxcontainer
```

We can connect to the running container with following command



```
C:\pushpamoulika>docker exec -it 9eebfd1ec4e6e023c41528bd309142450adac430b556f2ebd240ce2abb471e9f /bin/bash
root@9eebfd1ec4e6:/#
root@9eebfd1ec4e6:/# apt update
Err:1 http://deb.debian.org/debian bullseye InRelease
  Could not connect to deb.debian.org:80 (151.101.158.132). - connect (111: Connection refused)
Err:2 http://deb.debian.org/debian-security bullseye-security InRelease
  Unable to connect to deb.debian.org:80:
Err:3 http://deb.debian.org/debian bullseye-updates InRelease
  Unable to connect to deb.debian.org:80:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

We can verify whether the container is running or not on docker desktop



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Only show running containers

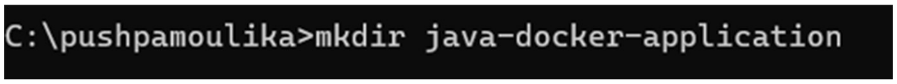
Name	Image	Status	Port(s)	Started	Actions
docker-nginx 5ce2a77e821e	nginx	Created (128)	80:80		
determined_tesla 42cf9f235d59	nginx	Running		2 hours ago	
upbeat_hellman 9e8bf52597dc	java-app	Exited (1)			
fervent_knuth 0d068050912d	java-app	Exited (1)			
hopeful_shaw 75a6f50375d7	java-app	Exited (1)			
condescending_carson bff5cdad4275	java-app	Exited			
relaxed_ellis 8c98e5288211	nginx	Running		51 minutes ag	
admiring_elbakyan 9eebfd1ec4e6	nginx	Running		48 minutes ag	
nginxcontainer 0ae6def85777	nginx	Created (128)	80:80		

Showing 11 items

Q2) Create the basic java application, generate its image with necessary files, and execute it with docker.

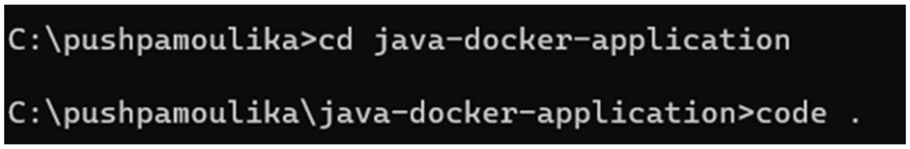
Creating a java application on docker

Step1: create a directory.



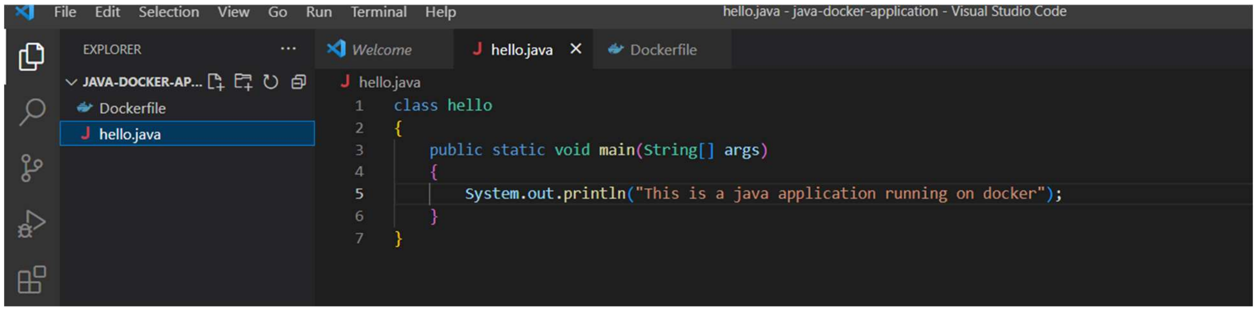
```
C:\pushpamoulika>mkdir java-docker-application
```

Step2: go to the directory you have created



```
C:\pushpamoulika>cd java-docker-application

C:\pushpamoulika\java-docker-application>code .
```

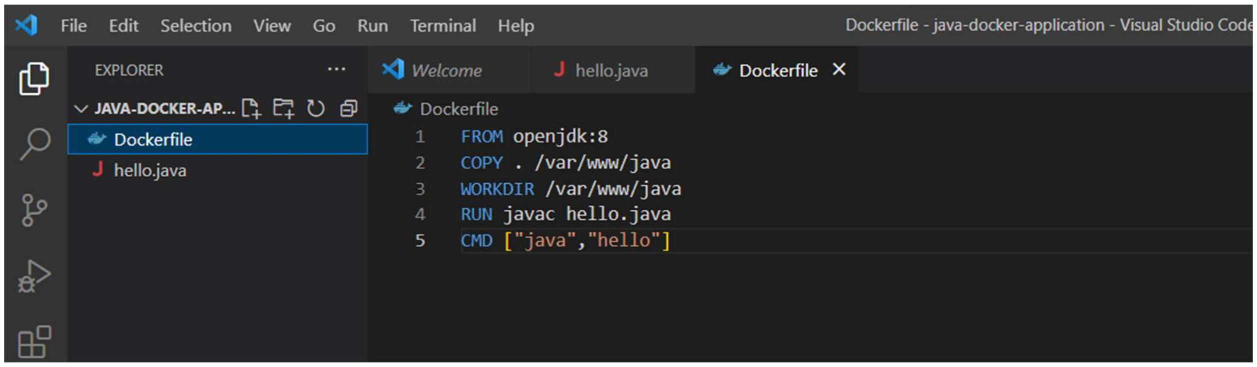
Step3: create a java file and save it as hello.java



File Edit Selection View Go Run Terminal Help
hello.java - java-docker-application - Visual Studio Code
EXPLORER
JAVA-DOCKER-AP...
Dockerfile
hello.java
Welcome
hello.java
Dockerfile

```java
class hello
{
    public static void main(String[] args)
    {
        System.out.println("This is a java application running on docker");
    }
}
```

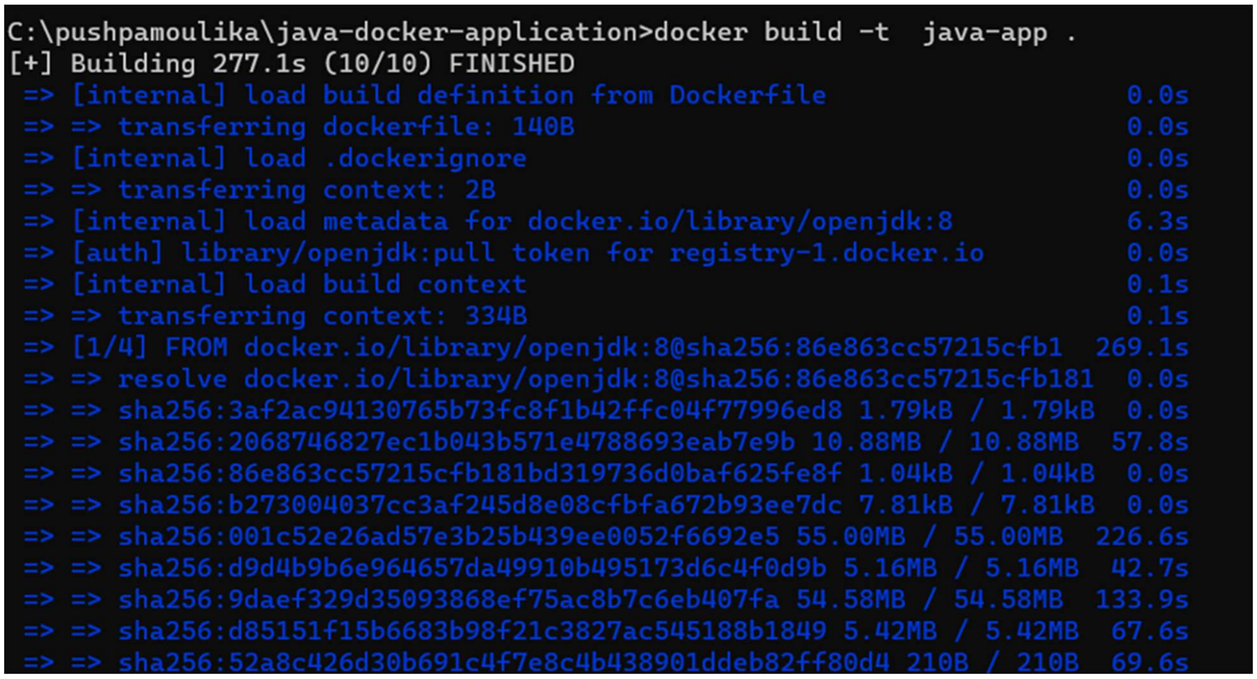
Step4: now create a docker file with commands



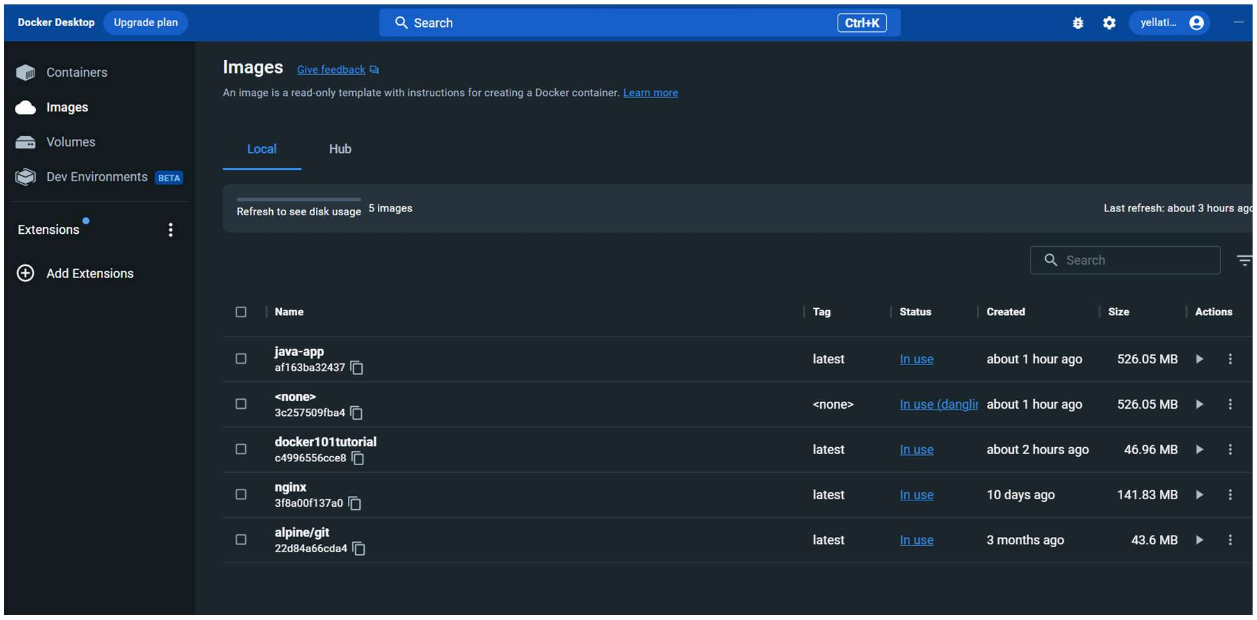
File Edit Selection View Go Run Terminal Help
Dockerfile - java-docker-application - Visual Studio Code
EXPLORER
JAVA-DOCKER-AP...
Dockerfile
hello.java
Welcome
hello.java
Dockerfile

```dockerfile
FROM openjdk:8
COPY . /var/www/java
WORKDIR /var/www/java
RUN javac hello.java
CMD ["java","hello"]
```

Step5: now build an image called java-app.



```
C:\pushpamoulika\java-docker-application>docker build -t  java-app .
[+] Building 277.1s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                      0.0s
 => => transferring dockerfile: 140B                                      0.0s
 => [internal] load .dockerignore                                         0.0s
 => => transferring context: 2B                                           0.0s
 => [internal] load metadata for docker.io/library/openjdk:8              6.3s
 => [auth] library/openjdk:pull token for registry-1.docker.io            0.0s
 => [internal] load build context                                         0.1s
 => => transferring context: 334B                                         0.1s
 => [1/4] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb1  269.1s
 => => resolve docker.io/library/openjdk:8@sha256:86e863cc57215cfb181  0.0s
 => => sha256:3af2ac94130765b73fc8f1b42ffc04f77996ed8 1.79kB / 1.79kB  0.0s
 => => sha256:2068746827ec1b043b571e4788693eab7e9b 10.88MB / 10.88MB  57.8s
 => => sha256:86e863cc57215cfb181bd319736d0baf625fe8f 1.04kB / 1.04kB  0.0s
 => => sha256:b273004037cc3af245d8e08cfbfa672b93ee7dc 7.81kB / 7.81kB  0.0s
 => => sha256:001c52e26ad57e3b25b439ee0052f6692e5 55.00MB / 55.00MB  226.6s
 => => sha256:d9d4b9b6e964657da49910b495173d6c4f0d9b 5.16MB / 5.16MB  42.7s
 => => sha256:9daef329d35093868ef75ac8b7c6eb407fa 54.58MB / 54.58MB  133.9s
 => => sha256:d85151f15b6683b98f21c3827ac545188b1849 5.42MB / 5.42MB  67.6s
 => => sha256:52a8c426d30b691c4f7e8c4b438901ddeb82ff80d4 210B / 210B  69.6s
```



Images

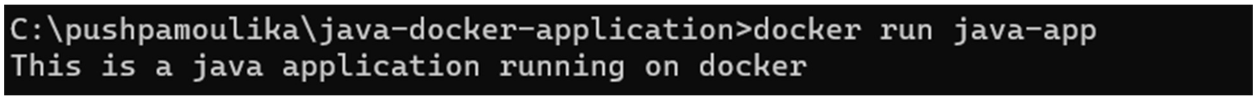
An image is a read-only template with instructions for creating a Docker container. Learn more

Refresh to see disk usage 5 images

Last refresh: about 3 hours ag

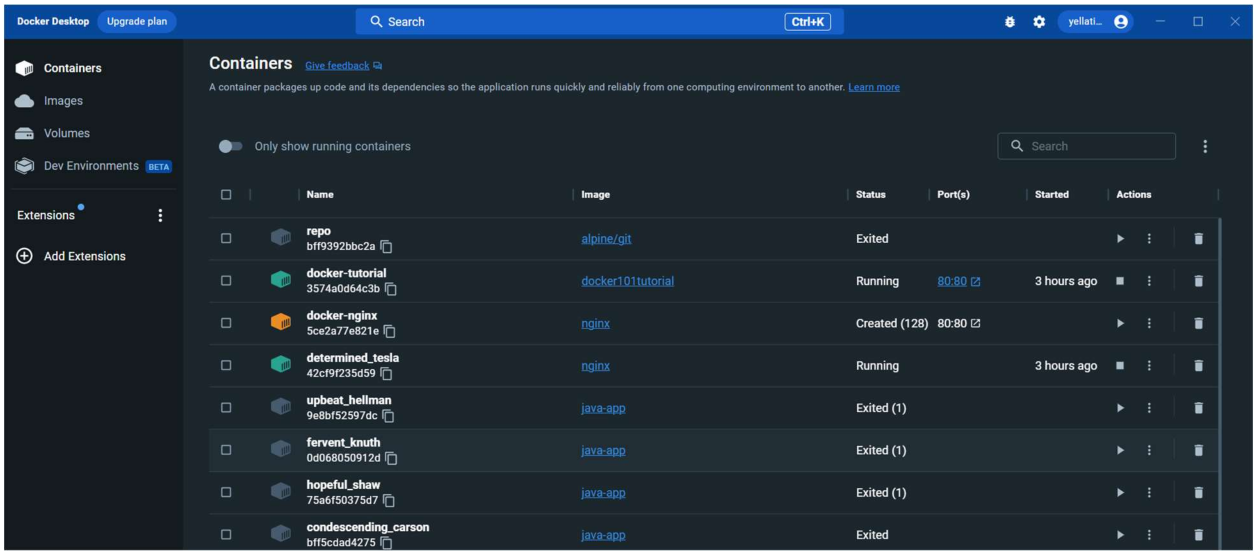
Name	Tag	Status	Created	Size	Actions
java-app af163ba32437	latest	In use	about 1 hour ago	526.05 MB	
<none> 3c257509fba4	<none>	In use (danglir	about 1 hour ago	526.05 MB	
docker101tutorial c4996556cce8	latest	In use	about 2 hours ago	46.96 MB	
nginx 3f8a00f137a0	latest	In use	10 days ago	141.83 MB	
alpine/git 22d84a66cda4	latest	In use	3 months ago	43.6 MB	

Step6: now run the java-app image by using command docker run <image name>



```
C:\pushpamoulika\java-docker-application>docker run java-app
This is a java application running on docker
```

By opening docker desktop, we the java application running.



Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more

Name	Image	Status	Port(s)	Started	Actions
repo bff9392bbc2a	alpine/git	Exited			
docker-tutorial 3574a0d64c3b	docker101tutorial	Running	80:80	3 hours ago	
docker-nginx 5ce2a77e821e	nginx	Created (128)	80:80		
determined_tesla 42cf9f235d59	nginx	Running		3 hours ago	
upbeat_hellman 9e8bf52597dc	java-app	Exited (1)			
fervent_knuth 0d068050912d	java-app	Exited (1)			
hopeful_shaw 75a6f50375d7	java-app	Exited (1)			
condescending_carson bff5cdad4275	java-app	Exited			