

Check docker version:

```
C:\Users\trand\Downloads>docker -v
Docker version 27.3.1, build ce12230
```

Containerize hello-world image

If hello-world doesn't exist in local, Docker is going to pull from Docker Hub

We are going to run this with **docker run hello-world** command

```
C:\Users\trand>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Download complete
Digest: sha256:7e1a4e2d11e2ac7a8c3f768d4166c2defeb09d2a750b010412b6ea13de1efb19
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

Run nginx with docker

We can use docker to implement nginx without downloading it on my computer.

We can use docker to pull nginx image and run this.

```
C:\Users\trand>docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
c22eb46e871a: Download complete
417c4bccf534: Download complete
e7e0ca015e55: Download complete
373fe654e984: Download complete
6e909acdb790: Download complete
5eaa34f5b9c2: Download complete
97f5c0f51d43: Download complete
Digest: sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
Status: Downloaded newer image for nginx:latest
```

List container with docker ps:

```
C:\Users\trand>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
c05add4870c2   nginx     "/docker-entrypoint...." 15 minutes ago Up 4 seconds  80/tcp       interesting_chatelet
```

```
C:\Users\trand>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  NAMES          CREATED        STATUS        PORTS
c05add4870c2   nginx     "/docker-entrypoint...." interesting_chatelet 16 minutes ago Up 34 seconds  80/tcp
fb773a01d959   hello-world "/hello"               jovial_wiles    18 minutes ago Exited (0) 18 minutes ago
f04c782d7674   mongo:7.0 "docker-entrypoint.s..." web_ks-mongo3-1 4 days ago     Exited (255) 23 minutes ago 27017/tcp
f6f2a73b97bb   mongo:7.0 "docker-entrypoint.s..." web_ks-mongo2-1 4 days ago     Exited (255) 23 minutes ago 27017/tcp
9a77ad0c4456   mongo:7.0 "docker-entrypoint.s..." web_ks-mongo1-1 4 days ago     Exited (255) 23 minutes ago 27017/tcp
a04af6da6482   mongo:8.0.4 "docker-entrypoint.s..." my-mongo        6 weeks ago     Exited (0) 5 weeks ago
27f6f5684f5a   mysql     "docker-entrypoint.s..." some-mysql       2 months ago     Exited (255) 5 days ago 0.0.0.0:3306->3306/tcp, 33060/tcp
e4dfceafb735   rabbitmq:4.0-management rabbitmq          2 months ago     Exited (0) 2 months ago
```

List local images:

```
C:\Users\trand\Downloads>docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
java_hello_world	latest	97fd47409077	8 minutes ago	642MB
mongo	7.0	ac9e83c57e32	3 weeks ago	1.09GB
nginx	latest	124b44bfc9cc	6 weeks ago	279MB
hello-world	latest	7e1a4e2d11e2	8 weeks ago	20.4kB
rabbitmq	3	5d03a2dfceae	3 months ago	341MB
mongo	8.0.4	b5725ac74a0d	3 months ago	1.18GB
mcr.microsoft.com/mssql/server	2019-latest	9b7a16515362	4 months ago	2.02GB
testcontainers/ryuk	0.11.0	2eaec4fc2cf1	4 months ago	28.3MB
mysql	latest	0255b469f013	5 months ago	828MB
rabbitmq	3-management	29ded394fc86	6 months ago	394MB
rabbitmq	4.0-management	4a2b95d53111	6 months ago	410MB
mongo	4.0.10	7df93c5e2d14	5 years ago	585MB

Go to inside container environment:

We can configure something inside container

```
C:\Users\trand\Downloads>docker exec -it 27f6f5684f5a bin/bash
bash-5.1# ls
afs  boot  docker-entrypoint-initdb.d  home  lib64  mnt  proc  run  srv  tmp  var
bin  dev  etc                          lib   media  opt  root  sbin  sys  usr
```

Stop container:

If the container isn't used, we can stop it to reduce resources.

```
C:\Users\trand\Downloads>docker stop 27f6f5684f5a
27f6f5684f5a

C:\Users\trand\Downloads>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Delete specific container:

```
C:\Users\trand>docker rm -f c05add4870c2
c05add4870c2

C:\Users\trand>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

Restart specific container:

```
C:\Users\trand>docker restart a04af6da6482
a04af6da6482

C:\Users\trand>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
a04af6da6482	mongo:8.0.4	"docker-entrypoint.s..."	6 weeks ago	Up 3 seconds	0.0.0.0:27017->27017/tcp	my-mongo

Run java with docker file

Step 1: Create Dockerfile

```
# Use an official Java runtime as a parent image
FROM openjdk:17-slim

# Set the working directory in the container
WORKDIR /app

# Copy the Java source file(s)
COPY Helloworld.java .

# Compile the Java program
RUN javac Helloworld.java

# Run the Java program when the container launches
CMD ["java", "Helloworld"]
```

Step 2: Using **docker build -t java_hello_world .**

```
C:\Users\trand\Downloads>docker build -t java_hello_world .
[+] Building 33.0s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile                0.0s
=> => transferring dockerfile: 368B                                0.0s
=> [internal] load metadata for docker.io/library/openjdk:17-slim 1.1s
=> [internal] load .dockerignore                                   0.0s
=> => transferring context: 2B                                       0.0s
=> [1/4] FROM docker.io/library/openjdk:17-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3 27.3s
=> => resolve docker.io/library/openjdk:17-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f 0.0s
=> => sha256:6ce99fdf16e86bd02f6ad66a0e1334878528b5a4b5487850a76e0c08a7a27d56 187.90MB / 187.90MB 21.9s
=> => sha256:44d3aa8d076675d49d85180b0ced9daef210fe4fdff4b422b9cf384e591d0 1.58MB / 1.58MB 1.6s
=> => sha256:1fe172e4850f03bb45d41a20174112bc119fbfec42a650edbbd8491aee32e3c3 31.38MB / 31.38MB 10.8s
=> => extracting sha256:1fe172e4850f03bb45d41a20174112bc119fbfec42a650edbbd8491aee32e3c3 3.0s
=> => extracting sha256:44d3aa8d076675d49d85180b0ced9daef210fe4fdff4b422b9cf384e591d0 0.4s
=> => extracting sha256:6ce99fdf16e86bd02f6ad66a0e1334878528b5a4b5487850a76e0c08a7a27d56 5.4s
=> [internal] load build context                                   0.0s
=> => transferring context: 162B                                       0.0s
=> [2/4] WORKDIR /app                                             1.9s
=> [3/4] COPY Helloworld.java .                                   0.1s
=> [4/4] RUN javac Helloworld.java                               1.7s
```

Step 3: Run image with **docker run java_hello_world**

```
C:\Users\trand\Downloads>docker run java_hello_world
Hello World
```

Delete everything (image, volume, container,...)

Something, we will encounter some problem which isn't caused by you.

Maybe docker cache data like images, volume make your container doesn't work well.

We are going to delete everything and setup from zero.

```
C:\Users\trand\Downloads>docker system prune
```

```
WARNING! This will remove:
```

- all stopped containers
- all networks not used by at least one container
- all dangling images
- unused build cache

```
Are you sure you want to continue? [y/N] y
```

```
Deleted Containers:
```

```
6e7b536142befab5214e52d32db12e46b21788e33f912a24160290d0f6797292  
fb773a01d959fc593f2e8fffdfa4f888ea18d9813035a4361d5a781df57b86a7  
e4dfceafb735c8ef0d67b79d3fa8f486c0612bb6a051fd23a71809cb057fa107
```

```
Deleted build cache objects:
```

```
jkrk9wutqs22pfhb065klcuoc  
4zad070ktmy08w8luj4zmyjpk  
0usj68fkgb57j6xqjuc9m0t09
```

```
Total reclaimed space: 45.06kB
```