

[CS304] Lab13 Docker

Part 1 Introduction to Docker

Docker overview

Docker is an open-source engine that automates the deployment of applications into containers.

It is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

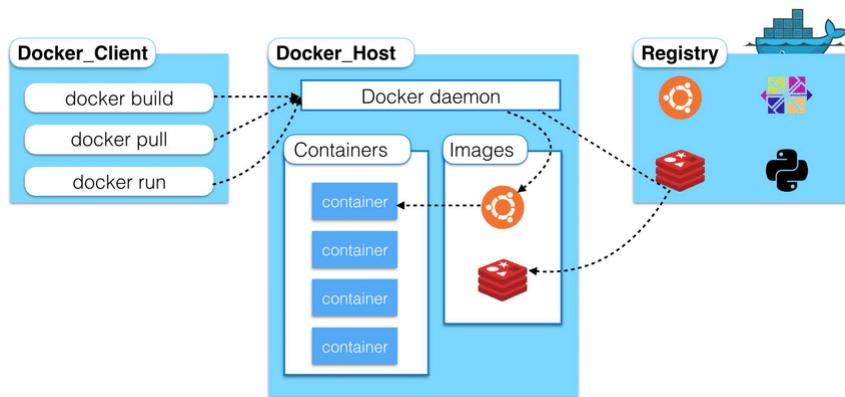
Docker : <http://www.docker.com>

Docker Hub : <https://hub.docker.com> (Registries)

What can I use Docker for:

- Fast, consistent delivery of your applications.
- Responsive deployment and scaling.
- Running more workloads on the same hardware.

Docker components



Container:

Simply put, a container is a sandboxed process on your machine that is isolated from all other processes on the host machine. That isolation leverages kernel namespaces and cgroups, features that have been in Linux for a long time. Docker has worked to make these capabilities approachable and easy to use. To summarize, a container:

- Is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI.
- Can be run on local machines, virtual machines or deployed to the cloud.
- Is portable (can be run on any OS).
- Is isolated from other containers and runs its own software, binaries, and configurations.

Image: When running a container, it uses an isolated filesystem. This custom filesystem is provided by a container image. Since the image contains the container's filesystem, it must contain everything needed to run an application - all dependencies, configurations, scripts, binaries, etc. The image also contains other configuration for the container, such as environment variables, a default command to run, and other metadata.

Registries: Docker Hub is the world's largest library and community for container images (from its website)

Part 2 Install Docker Engine on WSL(Ubuntu)

Reference: <https://docs.docker.com/engine/install/ubuntu/#installation-methods>

https://yeasy.gitbook.io/docker_practice/install/ubuntu

1. Uninstall old versions Older versions of Docker went by the names of docker, docker.io, or docker-engine, you might also have installations of containerd or runc. Uninstall any such older versions before attempting to install a new version:

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

2. Install using the convenience script(You can install Docker Engine in different ways, depending on your needs)

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

3. To create the docker group and add your user: <https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user>

Create the docker group.

```
sudo groupadd docker
```

Add your user to the docker group.

```
sudo usermod -aG docker $USER
```

Log out and log back in so that your group membership is re-evaluated

4. Verify that the Docker Engine installation is successful by running the hello-world image

```
sudo service docker start
sudo docker run hello-world
```



```
se@LAPTOP-EOHBGLP: ~
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

This message is shown once a day. To disable it please create the
/home/se/.hushlogin file.
se@LAPTOP-EOHBGLP: $ sudo apt-get remove docker docker-engine docker.io containerd runc
[sudo] password for se:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package docker-engine
se@LAPTOP-EOHBGLP: ~$ curl -fsSL https://get.docker.com -o get-docker.sh
se@LAPTOP-EOHBGLP: ~$ sudo sh get-docker.sh
# Executing docker install script, commit: a8a6b338bdfedd7ddefb96fe3e7fe7d4036d945a

WSL DETECTED: We recommend using Docker Desktop for Windows.
Please get Docker Desktop from https://www.docker.com/products/docker-desktop

You may press Ctrl+C now to abort this script.
+ sleep 20
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates
+ sh -c mkdir -p /etc/apt/keyrings && chmod -R 0755 /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /etc/
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.co
.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce docker-ce-cli containerd
ilidx-plugin >/dev/null

=====
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:
```

```
dockerd-rootless-setuptool.sh install

se@LAPTOP-EOHBGLP: ~
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/
```

```
se@LAPTOP-EOHBGLP: ~$ sudo groupadd docker
se@LAPTOP-EOHBGLP: ~$ r' already exists
se@LAPTOP-EOHBGLP: ~$ sudo usermod -aG docker $USER
se@LAPTOP-EOHBGLP: ~$ sudo service docker start
se@LAPTOP-EOHBGLP: ~$ cker
se@LAPTOP-EOHBGLP: ~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:4e83453afed1b4fa1a3500525091dbfcfa6ce1e66903fd4c01ff015dbcb1ba33e
Status: Downloaded newer image for hello-world:latest
```

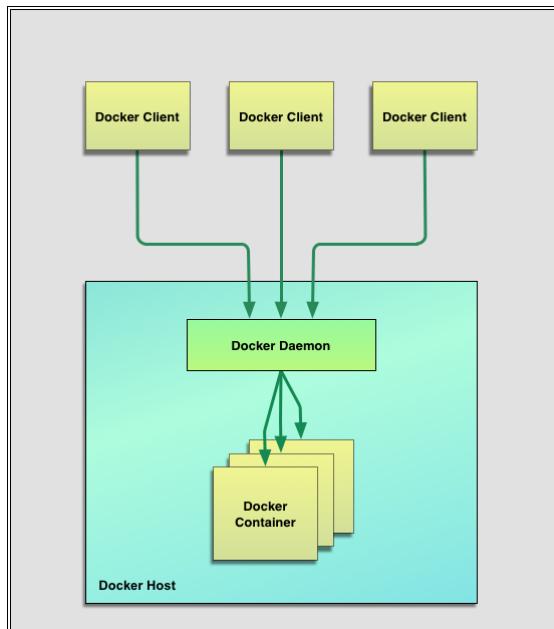
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

5. basic command

```
docker version      # you can see there are client and server
docker run hello-world #pull a docker image(if you haven't pull) and run it
docker images       #list docker images
docker ps           #list running docker containers
docker info
```



```
se@LAPTOP-EOBHBGLP:~$ docker version
Client: Docker Engine - Community
 Version:          23.0.5
 API version:     1.42
 Go version:      go1.19.8
 Git commit:      bc4487a
 Built:           Wed Apr 26 16:21:07 2023
 OS/Arch:         linux/amd64
 Context:         default

Server: Docker Engine - Community
Engine:
 Version:          23.0.5
 API version:     1.42 (minimum version 1.12)
 Go version:      go1.19.8
 Git commit:      94d3ad6
 Built:           Wed Apr 26 16:21:07 2023
 OS/Arch:         linux/amd64
 Experimental:   false
 containerd:
 Version:          1.6.20
 GitCommit:        2806fc1057397dbaeefbea0e4e17bddfb388f38
 runc:
 Version:          1.1.5
 GitCommit:        v1.1.5-0-gf19387a
 docker-init:
 Version:          0.19.0
 GitCommit:        de40ad0

se@LAPTOP-EOBHBGLP:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world     latest   feb5d9fea6a5  19 months ago  13.3kB
se@LAPTOP-EOBHBGLP:~$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
se@LAPTOP-EOBHBGLP:~$ docker info
Client:
 Context:         default
 Debug Mode:     false
 Plugins:
 buildx: Docker Buildx (Docker Inc.)
```

Part 3 Get start

1. Docker Image basic:

```
docker images
```

-a : list all images -q : only list id --digests : list image key message

name	description
REPOSITORY	Image repository
TAG	Image TAG
CREATED	Image created time
SIZE	Image size

```
se@LAPTOP-EOBHBGLP:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world    latest    feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOBHBGLP: $
```

search images:

```
docker search mysql
```

also you can search in: https://hub.docker.com/_/

The terminal window shows the command `docker search mysql` with the following output:

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
mysql	MySQL is a widely used, open-source relation...	14085	[OK]	
mariadb	MariaDB Server is a high performing open sou...	5379	[OK]	
percona	Percona Server is a fork of the MySQL relati...	606	[OK]	
phpmyadmin	phpMyAdmin – A web interface for MySQL and M...	788	[OK]	
circleci/mysql	MySQL is a widely used, open-source relation...	29		

The Docker Hub search results page for "mysql" shows the official MySQL image details. The image is described as "MySQL is a widely used, open-source relational database management system (RDBMS)". It has 14,085 stars, is official, and is automated. A red box highlights this description.

Get images: you can use `docker pull --help` to search more.

```
docker pull mysql
```

```
se@LAPTOP-EOBHBGLP:~$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
328ba678bf27: Pull complete
f3f5ff008d73: Pull complete
dd7054d6d0c7: Pull complete
70b5d4e8750e: Pull complete
cdc4a7b43bdd: Pull complete
a0608f8959e0: Pull complete
5823e721608f: Pull complete
a564ada930a9: Pull complete
539565d00e89: Pull complete
a11a06843fd5: Pull complete
92f6d4aa041d: Pull complete
Digest: sha256:a43f6e7e7f3a5e5b90f857fbcd4e3103ece771b19f0f75880f767cf66bbb6577
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
se@LAPTOP-EOBHBGLP:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED       SIZE
mysql           latest        8189e588b0e8   2 weeks ago   564MB
hello-world     latest        febd9fea6a5   19 months ago  13.3kB
```

different version:

- Maintained by:
the Docker Community and the MySQL Team
- Where to get help:
the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow

5.7.42-del
8.0.33-ora

Supported tags and respective Dockerfile links

- [8.0.33](#), [8.0](#), [8](#), [latest](#), [8.0.33-oracle](#), [8.0-oracle](#), [8-oracle](#), [oracle](#)
- [8.0.33-debian](#), [8.0-debian](#), [8-debian](#), [debian](#)
- [5.7.42](#), [5.7](#), [5](#), [5.7.42-oracle](#), [5.7-oracle](#), [5-oracle](#)
- [5.7.42-debian](#), [5.7-debian](#), [5-debian](#)

About

Docker O
open sou

Why O

These im
best prac
common

Quick reference (cont.)

- Where to file issues:
<https://github.com/docker-library/mysql/issues>
- Supported architectures: (more info)
[amd64](#), [arm64v8](#)

```
e83e8f2e82cc: Pull complete
:0f23deb01b84: Pull complete
f5bda3b184ea: Pull complete
:ed17edbc6604: Pull complete
33a94a6acfa7: Pull complete
:f153bd2953e4: Pull complete
:ab532edfb813: Pull complete
:c76bdfe4f3d0: Pull complete
:8a7ffe2f2551: Pull complete
:857ada4fbbcc: Pull complete
:b7c508404c3c: Pull complete
Digest: sha256:f7eef421000aaaf8332a91ab0b6c96b3c83ed2a981c29e6528b21ce10197cd16
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
se@LAPTOP-EOBHBGLP:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           5.7      dd6675b5cf ea    2 weeks ago   569MB
mysql           latest    8189e588b0e8  2 weeks ago   564MB
hello-world     latest    feb5d9fea6a5  19 months ago  13.3kB
se@LAPTOP-EOBHBGLP:~$ se@LAPTOP-EOBHBGLP:~$ se@LAPTOP-EOBHBGLP:~$
```

remove images

```
docker rmi -f      image ID #single image
docker rmi -f      Image name:tag image name:tag #multiple
docker rmi -f $(docker images -qa) # all
```

```
se@LAPTOP-EOHBHGLP: $ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           5.7      dd6675b5fea   2 weeks ago   569MB
mysql           latest    8189e588b0e8   2 weeks ago   564MB
hello-world     latest    feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOHBHGLP: ~
se@LAPTOP-EOHBHGLP: ~
se@LAPTOP-EOHBHGLP: ~$ docker rmi -f dd6675b5fea
Untagged: mysql:5.7
Deleted: sha256:f57eef421000aaf8332a91ab0b6c96b3c83ed2a981c29e6528b21ce10197cd16
Deleted: sha256:dd6675b5fea17abb655ea8229cbcfa5db9d0b041f839db0c24228c2e18a4bdf
Deleted: sha256:ce931a23976217d9a54389e972eb3fc823ec4c2ef51ff712f6330437bd7a373a
Deleted: sha256:64eb3560e160f706ef4994cb38a97d1ce32a219e547ead81fd51766d46c5e1c2
Deleted: sha256:9e009673657831f462ffadcbf099a070f0bb03ccb105483c69470a25ab87fe0
Deleted: sha256:a37428431f1f5ebda54ff32a40299e87a2e5e9b867bf9c1e619ff9d04850a9a
Deleted: sha256:b5463af91376b3d4772113805dbce79805129326c301a80947cbfd43c8069898
Deleted: sha256:c0a975af561fdd0b68965ca9c572d91244b6dff6124b6a4d66b3188c0cb7d9f
Deleted: sha256:38700ee823d5d82b695bb7d53973c0106384a6984b553e2af6d6fb9545d9636
Deleted: sha256:7cfca860605c4f16b1f255ebd51e2801e00ac16997c7840428f88a2246cfebb
Deleted: sha256:0d8a62cb129e3cebaaf51f977b670282775d48a9cdaa021a0cb5ed73243a446
Deleted: sha256:66e6e199acb7395ac9a10b125841eafdfa3f80a1a3f710e0a0a09519da228bce2
Deleted: sha256:e74a57638021cebbbe0ddaa98a27675f9b53258050ee2c29c3e5fb7fb1f3ab533
se@LAPTOP-EOHBHGLP: ~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest    8189e588b0e8   2 weeks ago   564MB
hello-world     latest    feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOHBHGLP: ~
se@LAPTOP-EOHBHGLP: ~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest    8189e588b0e8   2 weeks ago   564MB
hello-world     latest    feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOHBHGLP: ~$ docker images mysql:latest
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
mysql           latest    8189e588b0e8   2 weeks ago   564MB
se@LAPTOP-EOHBHGLP: ~$ docker rmi -f mysql:latest
Untagged: mysql:latest
Deleted: sha256:a43f6e7e7f3a5e5b90f857fdb4e3103ece771b19f0f75880f767cf66bbb6577
Deleted: sha256:8189e588b0e8fcc95b0d764d6f7bdb55b5b41e9249157177d73781058f603ca9
Deleted: sha256:48c450c06ed83938e899fb0b77b2e9e35094015b503bb5e88de6c2d93f445241
Deleted: sha256:c5d77efb49ec3a7a74ab898b9da9217ec78fa9bee4701840902546761d60329
Deleted: sha256:de0c0e28b37cc33347f02709e0a6a2f17637b1e761fb96616861ed345bd34f6
Deleted: sha256:c635684e233946f93fe008e0322c86e15ccb97b56c58d269a5f89da15d973d
Deleted: sha256:0b795b85d567512d79a544b1b74f21108339156ccbcb78d16921e96a2a69f687b
Deleted: sha256:16e250c36f4e01085512653edd47fd03b60d230ddb575aab8eb224d96e668f
Deleted: sha256:d023b92a46a5fa8fa8d54387e6d3cb0c73997fefc64ec9000eab0ee1c550ef45
Deleted: sha256:f1c1643119168a94089eab1c9126cda0ee6056a4bb4b18e27a7dcacdf4823972
Deleted: sha256:b147319dd21e8994e6d2fb3bb58a8278c5a72f39488e1f1cff94fc73f1089eb9
Deleted: sha256:ff7c2b28c0dfa63d0d30b7a5069bf526b0f6492143110381351bbf7d07b4baf
Deleted: sha256:caef4a4e45110eab274ebbdcb781f9227229f947f8718cee62beff1aac8f1d5b
se@LAPTOP-EOHBHGLP: ~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
hello-world     latest    feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOHBHGLP: ~
```

2. Docker Container

Sample :download Ubuntu and construct a Container

```
docker pull ubuntu:22.04
```

```
se@LAPTOP-EOBHBGLP: $ docker pull ubuntu:22.04
22.04: Pulling from library/ubuntu
2ab09b027e7f: Pull complete
Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3bab295a0428a6d21
Status: Downloaded newer image for ubuntu:22.04
docker.io/library/ubuntu:22.04
se@LAPTOP-EOBHBGLP:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
ubuntu          22.04       08d22c0ceb15   8 weeks ago   77.8MB
hello-world     latest       feb5d9fea6a5   19 months ago  13.3kB
se@LAPTOP-EOBHBGLP:~$
```

start and enter a Container: <https://docs.docker.com/engine/reference/run/>

```
docker run -it ubuntu:22.04 /bin/bash

docker run [OPTIONS] IMAGE [COMMAND][ARG...]
--name="Name" # 给容器指定一个名字
-d # 后台方式运行容器，并返回容器的id！
-i # 以交互模式运行容器，通过和 -t 一起使用
-t # 给容器重新分配一个终端，通常和 -i 一起使用
-P # 随机端口映射（大写）
-p # 指定端口映射（小写）
```

list container:

```
docker ps [OPTIONS]
-a #list all history Containers
-l #list recent Container
-n=? #list number n recently construct Containers
```

exit Container

```
exit # stop and quit
ctrl+P+Q # do not stop and quit
```

```

se@LAPTOP-EOBHBGLP:~$ docker run -it ubuntu:22.04 /bin/bash
root@9463a0a6d35b:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@9463a0a6d35b:/# exit
exit
se@LAPTOP-EOBHBGLP:~$ docker ps
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
se@LAPTOP-EOBHBGLP:~$ docker ps -a
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
9463a0a6d35b  ubuntu:22.04  "/bin/bash"  33 seconds ago  Exited (0)  26 seconds ago  relaxed_chaum
0a49aabceaae  ubuntu:22.04  "/bin/bash"  About a minute ago  Exited (0)  About a minute ago  sleepy_bose
941b171282dd  ubuntu:22.04  "/bin/bash"  4 minutes ago  Exited (0)  About a minute ago  hungry_gould
d51810dab3da  ubuntu:22.04  "/bin/bash"  8 minutes ago  Exited (0)  7 minutes ago  recursing_rho
des
ae9b82a9154a  hello-world  "/hello"  About an hour ago  Exited (0)  About an hour ago  flamboyant_vii
svesvaraya
9dfe5f9dc6d0  hello-world  "/hello"  2 hours ago  Exited (0)  2 hours ago  inspiring_hoo
ver
9de5c878e6e8  hello-world  "/hello"  2 hours ago  Exited (0)  2 hours ago  elegant_lumie
re
0c120de09774  hello-world  "/hello"  16 hours ago  Exited (0)  16 hours ago  amazing_lampo
rt
se@LAPTOP-EOBHBGLP:~$ docker ps -l
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
9463a0a6d35b  ubuntu:22.04  "/bin/bash"  2 minutes ago  Exited (0)  2 minutes ago  relaxed_chaum
se@LAPTOP-EOBHBGLP:~$ docker ps -n=2
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
9463a0a6d35b  ubuntu:22.04  "/bin/bash"  4 minutes ago  Exited (0)  4 minutes ago  relaxed_chaum
0a49aabceaae  ubuntu:22.04  "/bin/bash"  4 minutes ago  Exited (0)  4 minutes ago  sleepy_bose
se@LAPTOP-EOBHBGLP:~$
```

start and stop container:

```

docker start (Container id or Container name)
docker restart (Container id or Container name)
docker stop (Container id or Container name)
docker kill (Container id or Container name) # Forced stop

```

delete Containers:

```

docker rm (Container id) # can not delete running container
docker rm -f $(docker ps -a -q) # delete all
docker ps -a -q|xargs docker rm # delete all

```

```

se@LAPTOP-EOBHBGLP:~$ docker run -it ubuntu:22.04 /bin/bash
root@9f8bdfb8aca2:/# exit
exit
se@LAPTOP-EOBHBGLP:~$ docker ps
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
acc88f1d7bb9  ubuntu:22.04  "/bin/bash"  About a minute ago  Up About a minute  angry_
meninsky
d51810dab3da  ubuntu:22.04  "/bin/bash"  23 minutes ago  Up 7 minutes  recurs_
ing_rhodes
se@LAPTOP-EOBHBGLP:~$ docker rm -f $(docker ps -a -q)
9f8bdfb8aca2
acc88f1d7bb9
9463a0a6d35b
0a49aabceaae
941b171282dd
d51810dab3da
ae9b82a9154a
9dfe5f9dc6d0
9de5c878e6e8
0c120de09774
se@LAPTOP-EOBHBGLP:~$ docker ps
CONTAINER ID  IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
se@LAPTOP-EOBHBGLP:~$
```

test command: `docker run -d ubuntu:22.04 /bin/bash -c "while true ;do echo cs304 ;sleep 1; done"`

then check log history `docker logs -f -t --tail (Container id)`

```
se@LAPTOP-EOBHBGLP:~$ docker run -d ubuntu:22.04 /bin/bash -c "while true ;do echo cs304 ;sleep 1; done"
18ea3f69c6a93f08e2d46f8947f7d520ca205bda8cad8054d263250763f967cd
se@LAPTOP-EOBHBGLP: $ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
.04 "/bin/bash -c 'while true ;do echo cs304 ;sleep 1; done'" 5 seconds ago Up 4 seconds
naughty_khorana 18ea3f69c6a9 ubuntu:22
se@LAPTOP-EOBHBGLP:~$ ^C
se@LAPTOP-EOBHBGLP: $ docker logs -tf --tail 10 18ea3f69c6a9
2023-05-03T06:47:49.771388188Z cs304
2023-05-03T06:47:50.775842268Z cs304
2023-05-03T06:47:51.779356249Z cs304
2023-05-03T06:47:52.783774629Z cs304
2023-05-03T06:47:53.787907610Z cs304
2023-05-03T06:47:54.792011490Z cs304
2023-05-03T06:47:55.796064770Z cs304
2023-05-03T06:47:56.798836951Z cs304
2023-05-03T06:47:57.801822341Z cs304
2023-05-03T06:47:58.804506023Z cs304
```

```
se@LAPTOP-EOBHBGLP: $ docker logs --help
Command 'dokcer' not found, did you mean:
  command 'docker' from deb docker.io (20.10.21-0ubuntu1~22.04.3)
  command 'docker' from deb podman-docker (3.4.4+ds1-1ubuntu1)
Try: sudo apt install <deb name>
se@LAPTOP-EOBHBGLP: $ docker logs --help

Usage: docker logs [OPTIONS] CONTAINER

Fetch the logs of a container

Aliases:
  docker container logs, docker logs

Options:
  --details      Show extra details provided to logs
  -f, --follow   Follow log output
  --since string Show logs since timestamp (e.g. "2013-01-02T13:23:37Z") or relative
                  (e.g. "42m" for 42 minutes)
  -n, --tail string Number of lines to show from the end of the logs (default "all")
  -t, --timestamps Show timestamps
  --until string Show logs before a timestamp (e.g. "2013-01-02T13:23:37Z") or relative
                  (e.g. "42m" for 42 minutes)
se@LAPTOP-EOBHBGLP:~$
```

check process information:`docker top (Container id)`

UID	PID	PPID	C	STIME	TTY	TIME
CMD						
root :00	3701	3681	0	14:47	?	00:00
root :00	/bin/bash -c while true ;do echo cs304 ;sleep 1; done	3701	0	14:51	?	00:00
	sleep 1					

check detail information: `docker inspect (Container id)`

```
选择 se@LAPTOP-EOBHBGLP: ~
se@LAPTOP-EOBHBGLP:~$ docker inspect 18ea3f69c6a9
[{"Id": "18ea3f69c6a93f08e2d46f8947f7d520ca205bda8cad8054d263250763f967cd",
 "Created": "2023-05-03T06:47:04.781973369Z",
 "Path": "/bin/bash",
 "Args": [
   "-c",
   "while true ;do echo cs304 ;sleep 1; done"
 ],
 "State": {
   "Status": "running",
   "Running": true,
   "Paused": false,
   "Restarting": false,
   "OOMKilled": false,
   "Dead": false,
   "Pid": 3701,
   "ExitCode": 0,
   "Error": "",
   "StartedAt": "2023-05-03T06:47:05.596770123Z",
   "FinishedAt": "0001-01-01T00:00:00Z"
 },
 "Image": "sha256:08d22c0ceb150ddeb2237c5fa3129c0183f3cc6f5eeb2e7aa4016da3ad02140a",
 "ResolvConfPath": "/var/lib/docker/containers/18ea3f69c6a93f08e2d46f8947f7d520ca205bda8cad8054d263250763f967cd/r
esolv.conf"
}
```

Enter a container(open a new terminal, can start a new process):**docker exec -it (Container id) bashShell**

sample: **docker exec -it 18ea3f69c6a9 /bin/bash**

```
se@LAPTOP-EOBHBGLP:~$ docker exec -it 18ea3f69c6a9 /bin/bash
root@18ea3f69c6a9:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@18ea3f69c6a9:/#
```

Enter a container(enter running terminal, can not start new process):**docker attach (Container id)**

copy document from container to host :**docker cp Container id:path/doc tar_path**

```
root@LAPTOP-EOBHBGLP:/home# docker ps
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
(c5bc3a32a5d1)      ubuntu:22.04   "/bin/bash"   2 minutes ago   Up 2 minutes   0.0.0.0:22->22/tcp   festive_payne
root@LAPTOP-EOBHBGLP:/home# docker attach c5bc3a32a5d1
root@c5bc3a32a5d1:/# cd /home
root@c5bc3a32a5d1:/home# ls
root@c5bc3a32a5d1:/home# touch lab13.java
root@c5bc3a32a5d1:/home# ls
a.lab13.java
root@c5bc3a32a5d1:/home# exit
exit
root@LAPTOP-EOBHBGLP:/home# docker ps
Command 'doker' not found, did you mean:
  command 'docker' from deb docker.io (20.10.21-0ubuntu1~22.04.3)
  command 'docker' from deb podman-docker (3.4.4+ds1-1ubuntu1)
Try: apt install <deb name>
root@LAPTOP-EOBHBGLP:/home# docker ps
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
root@LAPTOP-EOBHBGLP:/home# docker cp c5bc3a32a5d1:/home/lab13.java /home
Successfully copied 1.54kB to /home
root@LAPTOP-EOBHBGLP:/home# ls
cs304.java  lab13.java  se
root@LAPTOP-EOBHBGLP:/home#
```

Install in container:

```
hello-world  latest  feb5d0fea6a5  10 months ago  13.3kB
se@LAPTOP-E0BHBGLP: ~$ sudo docker run -i -t ubuntu:22.04 /bin/bash
root@d10133843306:~# hostname
d10133843306
root@d10133843306:~# cat /etc/hosts
127.0.0.1      localhost
::1    localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.3      d10133843306
root@d10133843306:~# ip a
bash: ip: command not found
root@d10133843306:~# ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root         1  0.2  0.0  4624  3676 pts/0    Ss   06:44   0:00 /bin/bash
root        12  0.0  0.0  7000  1514 pts/0    R+   06:45   0:00 ps -aux
root@d10133843306:~# apt-get update && apt-get install vim
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
```

install program in container

summary:

<https://docs.docker.com/engine/reference/commandline/cli/>

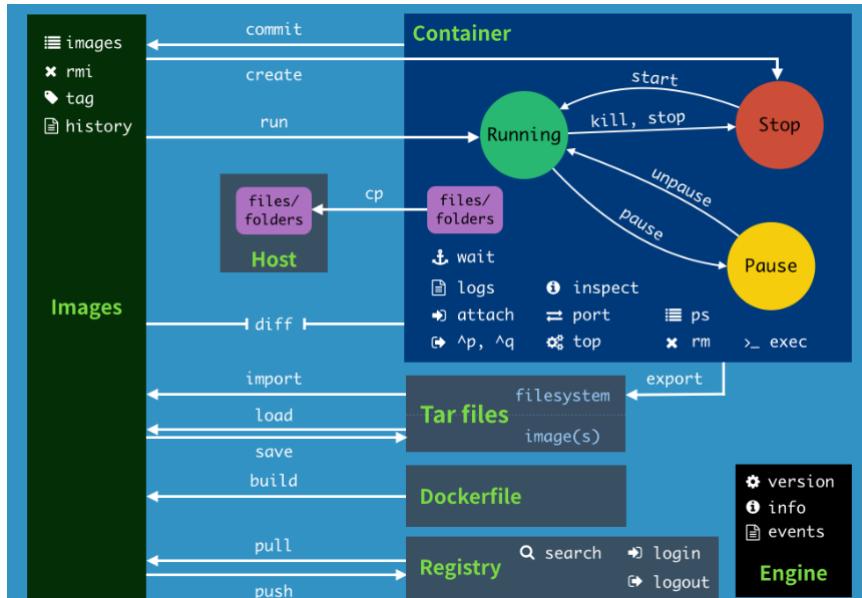
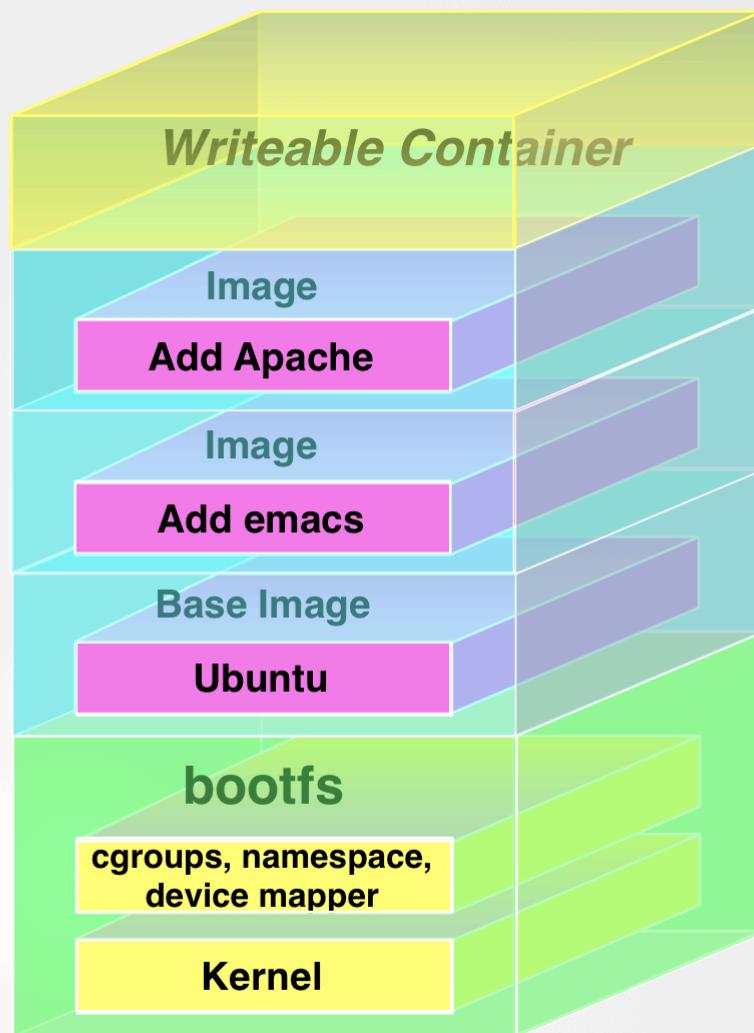


image and container

When a container is created, Docker builds from the stack of images and then adds the read-write layer on top. That layer, combined with the knowledge of the image layers below it and some configuration data, form the container.



Part 4 Teedy Image and Container construction

There are two ways to create a Docker image:

- Via the docker commit command
- Via the docker build command with a Dockerfile

Using commit command you need to create Docker Hub account ,login in then refer the container part do what you need to do,finally commit.

Building images with a Dockerfile is more suitable.

Dockerfile reference: <https://docs.docker.com/engine/reference/builder/>

command	description
---------	-------------

command	description
FROM	The FROM instruction initializes a new build stage and sets the Base Image for subsequent instructions. As such, a valid Dockerfile must start with a FROM instruction.
LABEL	The LABEL instruction adds metadata to an image. A LABEL is a key-value pair. To include spaces within a LABEL value, use quotes and backslashes as you would in command-line parsing. A few usage examples:
EXPOSE	The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime. You can specify whether the port listens on TCP or UDP, and the default is TCP if the protocol is not specified.
ADD	The ADD instruction adds files and directories from our build environment into our image
ENV	Environment variable substitution will use the same value for each variable throughout the entire instruction.
RUN	The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
CMD	The main purpose of a CMD is to provide defaults for an executing container. These defaults can include an executable, or they can omit the executable, in which case you must specify an ENTRYPOINT instruction as well.

Teedy Dockerfile

```

FROM sismics/ubuntu-jetty:9.4.36
LABEL maintainer="b.gamarr@sismics.com"

RUN apt-get update && \
    apt-get -y -q --no-install-recommends install \
    ffmpeg \
    mediainfo \
    tesseract-ocr \
    tesseract-ocr-ara \
    tesseract-ocr-ces \
    tesseract-ocr-chi-sim \
    tesseract-ocr-chi-tra \
    tesseract-ocr-dan \
    tesseract-ocr-deu \

```

Linux (Ubuntu 22.04)

```

sudo apt install \
default-jdk \
ffmpeg \
grunt \
maven \
npm \
tesseract-ocr \
tesseract-ocr-ara \
tesseract-ocr-ces \
tesseract-ocr-chi-sim \
tesseract-ocr-chi-tra \
tesseract-ocr-dan \
tesseract-ocr-deu \

```

Open <https://hub.docker.com/r/sismics/ubuntu-jetty/tags>

sismics/ubuntu-jetty:9.4.36 is a image created by simics

hub.docker.com/r/sismics/ubuntu-jetty/tags

sismics/ubuntu-jetty SPONSORED OSS

By [Sismics](#) • Updated 25 days ago

Image

Overview Tags

Sort by Newest Filter Tags

TAG	DIGEST	OS/ARCH	SCANNED	COMPRESSED SIZE
11.0.14 Last pushed 25 days ago by naku	888b119da4c4	linux/amd64	---	472.91 MB
9.4.51 Last pushed 2 months ago by naku	8f1bfff549687	linux/amd64	---	481.77 MB
9.4.36 Last pushed a year ago by naku	42153c1ed573	linux/amd64	---	456.16 MB

hub.docker.com/layers/sismics/ubuntu-jetty/9.4.36/images/sha256-42153c1ed573c7f3182a81aee1e1fad732761f07eb839934c8f889507fc36884?context=explore

[docker pull sismics/ubuntu-jetty:11...](#) [docker pull sismics/ubuntu-jetty:9...](#) [docker pull sismics/ubuntu-jetty:9.4...](#)

[Explore](#) [Pricing](#) [Sign In](#) [Register](#)

Explore / [sismics/ubuntu-jetty](#) / [9.4.36](#)

sismics/ubuntu-jetty:9.4.36

DIGEST: sha256:42153c1ed573c7f3182a81aee1e1fad732761f07eb839934c8f889507fc36884

OS/ARCH: linux/amd64 | COMPRESSED SIZE: 456.16 MB | LAST PUSHED: a year ago by [naku](#) | TYPE: Image

IMAGE LAYERS

Step	Command	Size
1	ADD file ... in /	33.22 MB
2	/bin/sh -c set -xe &&	842 B
3	/bin/sh -c rm -rf /var/lib/apt/lists/*	412 B
4	/bin/sh -c sed -i 's/^\(deb.*universe\)\$/\1/g'	850 B
5	/bin/sh -c mkdir -p /run/systemd	162 B
6	CMD ["bin/bash"]	0 B
7	MAINTAINER Benjamin Gamard <b.gamard@sismics.com>	0 B
8	ENV DEBIAN_FRONTEND=noninteractive	0 B
9	/bin/sh -c apt-get update &&	68.49 MB
10	/bin/sh -c curl -fsSL https://www.sismics.com/non	1.37 KB

```

17 /bin/sh -c apt-get -y -q          20.02 MB
18 MAINTAINER Benjamin Gamard <b.gamard@sismics.com>      0 B
19 RUN /bin/sh -c apt-get update      298.06 MB
20 ENV JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/        0 B
21 ENV JAVA_OPTS=-Duser.timezone=Europe/Paris -Dfile.encoding=UTF-8
22 MAINTAINER Jean-Marc Treméaux <jm.tremeaux@sismics.com>    0 B
23 ENV JETTY_VERSION=9.4.36.v20210114      0 B
24 RUN /bin/sh -c wget -nv           35.95 MB
25 WORKDIR /opt/jetty             0 B
26 RUN /bin/sh -c chmod +x         0 B
27 COPY opt /opt # buildkit       530 B
28 EXPOSE map[8080/tcp:{}]
29 ENV JETTY_HOME=/opt/jetty      0 B
30 ENV JAVA_OPTIONS=-Xmx512m      0 B
31 CMD ["bin/jetty.sh" "run"]

```

```

.gitattributes 33 tesseract-ocr-vie \
.gitignore       34 tesseract-ocr-vie && \
CODE_OF_CONDUCT.md 35 apt-get clean && rm -rf /var/lib/apt/lists/*
COPYING         36
Dockerfile       37 # Remove the embedded javax.mail jar from Jetty
docsxml          38 RUN rm -f /opt/jetty/lib/mail/javax.mail.glassfish-*.jar
pom.xml          39
README.md        40 ADD docs.xml /opt/jetty/webapps/docs.xml
Scratches and Consoles 41 ADD docs-web/target/docs-web-*.war /opt/jetty/webapps/docs.war
42
ENV JAVA_OPTIONS -Xmx1g
43
44

```

creat teedy images and serval containers

set `docs.xml`, `docs-web/target/docs-web-*.war`, `Dockerfile` to your files: We already have one teedy image, then we create another teedynew image;

```

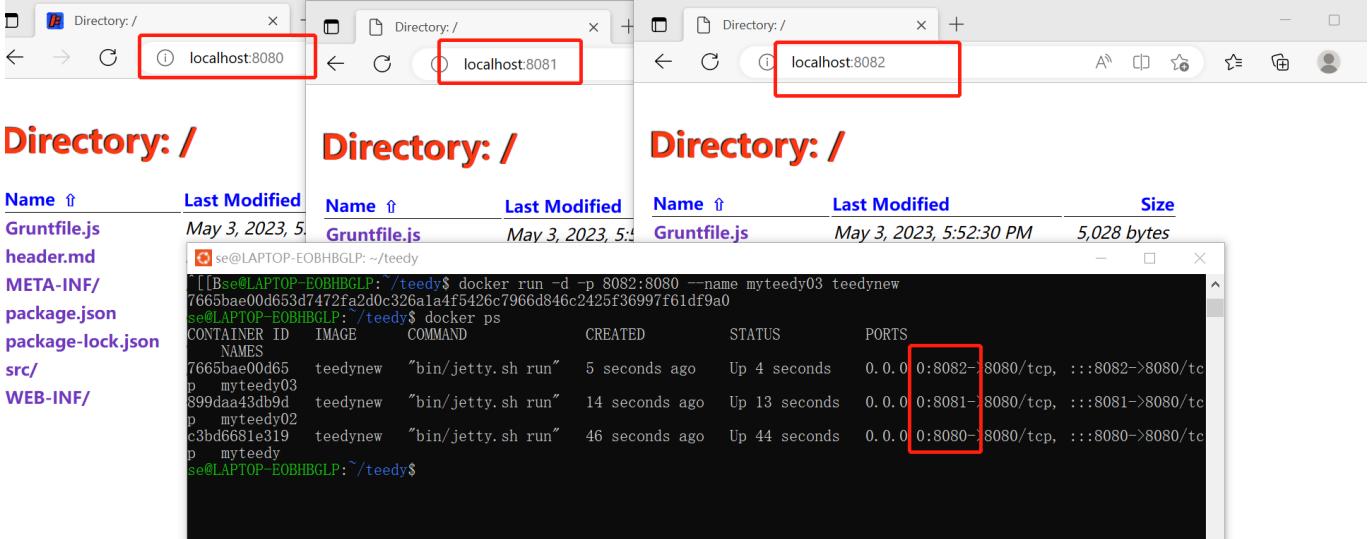
docker build -t teedynew .
docker run -d -p 8080:8080 --name myteedy teedynew
docker run -d -p 8081:8080 --name myteedy02 teedynew
docker run -d -p 8082:8080 --name myteedy03 teedynew

```

```

        0.0s
se@LAPTOP-EOBHBGLP:~/teedy$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
teedy          latest   3b5155b11e48  2 hours ago  1.18GB
teedynew        latest   3b5155b11e48  2 hours ago  1.18GB
my_first_container latest   ff8dc777f8b0  2 hours ago  172MB
lab13_demo      latest   869d9532ba4a  7 hours ago  544MB
ubuntu          22.04   08d22c0ceb15  8 weeks ago  77.8MB
hello-world     latest   feb5d9fea6a5  19 months ago 13.3kB
se@LAPTOP-EOBHBGLP:~/teedy$ docker rmi -f 3b5155b11e48
Untagged: teedy:latest
Untagged: teedynew:latest
Deleted: sha256:3b5155b11e48d6243b99c58a430cd1943ac54e05b8124bca4a3a4c61bc2eb234
se@LAPTOP-EOBHBGLP:~/teedy$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
my_first_container latest   ff8dc777f8b0  2 hours ago  172MB
lab13_demo      latest   869d9532ba4a  7 hours ago  544MB
ubuntu          22.04   08d22c0ceb15  8 weeks ago  77.8MB
hello-world     latest   feb5d9fea6a5  19 months ago 13.3kB
se@LAPTOP-EOBHBGLP:~/teedy$ docker build -t teedynew .
[+] Building 1.2s (10/10) FINISHED
  => [internal] load .dockerignore
    0.0s
  => => transferring context: 2B
    0.0s
  => [internal] load build definition from Dockerfile
    0.1s
  => => transferring dockerfile: 1.20kB
    0.0s

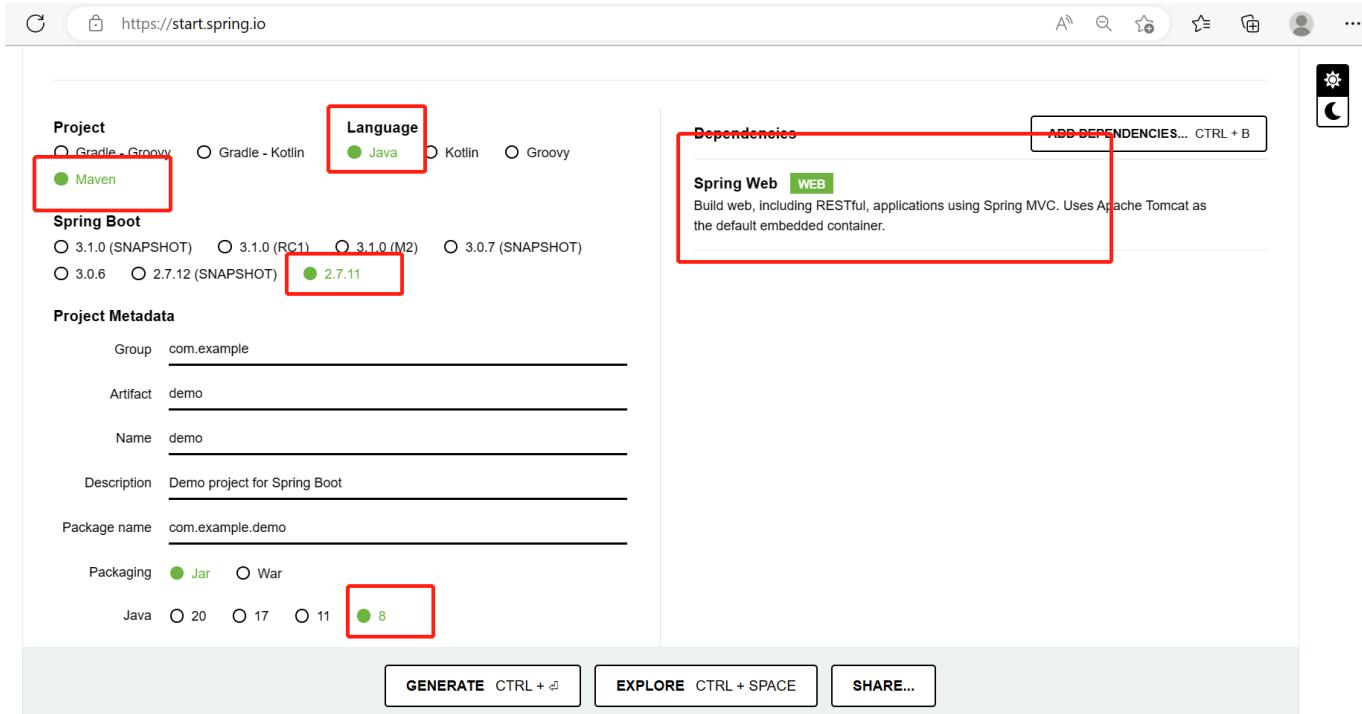
=> CACHED [2/5] RUN apt-get update && apt-get -y -q --no-install-recommends install      ffmpeg      mediainfo      tesseract-ocr      tesseract
=> CACHED [3/5] RUN rm -f /opt/jetty/lib/mail/javax.mail.glassfish-*.jar
=> CACHED [4/5] ADD docs.xml /opt/jetty/webapps/docs.xml
=> CACHED [5/5] ADD docs-web/target/docs-web-*.war /opt/jetty/webapps/docs.war
=> exporting to image
=> => exporting layers
=> => writing image sha256:3b5155b11e48d6243b99c58a430cd1943ac54e05b8124bca4a3a4c61bc2eb234
=> => naming to docker.io/library/teedynew
se@LAPTOP-EOBHBGLP:~/teedy$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
teedynew        latest   3b5155b11e48  2 hours ago  1.18GB
my_first_container latest   ff8dc777f8b0  2 hours ago  172MB
lab13_demo      latest   869d9532ba4a  7 hours ago  544MB
ubuntu          22.04   08d22c0ceb15  8 weeks ago  77.8MB
hello-world     latest   feb5d9fea6a5  19 months ago 13.3kB
se@LAPTOP-EOBHBGLP:~/teedy$ docker run -d -p 8080:8080 --name mytedy teedynew
c3bd6681e319fdfeada9cccad08a7986617faebec3fe4a361d24bdf205b3181b
se@LAPTOP-EOBHBGLP:~/teedy$ docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
c3bd6681e319  teedynew   "bin/jetty.sh run"  12 seconds ago Up 10 seconds  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  mytedy
se@LAPTOP-EOBHBGLP:~/teedy$ docker run -d -p 8081:8080 --name mytedy teedynew
docker: Error response from daemon: Conflict. The container name "/mytedy" is already in use by container "c3bd6681e319fdfeada9cccad08a7986617d24bdf205b3481b". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
se@LAPTOP-EOBHBGLP:~/teedy$ docker run -d -p 8081:8080 --name mytedy02 teedynew
899daa43db9d7cc8e620a81b334fad2c74e589f1bc60c957a200ea2775434eae
[[Bse@LAPTOP-EOBHBGLP:~/teedy$ docker run -d -p 8082:8080 --name mytedy03 teedynew
7665bae00d65d7472fa2d0c326a1a4f5426c7966d846c2425f36997f61df9a0
se@LAPTOP-EOBHBGLP:~/teedy$ docker ps
CONTAINER ID  IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
7665bae00d65  teedynew   "bin/jetty.sh run"  5 seconds ago Up 4 seconds  0.0.0.0:8082->8080/tcp, :::8082->8080/tcp  mytedy03
899daa43db9d  teedynew   "bin/jetty.sh run"  14 seconds ago Up 13 seconds  0.0.0.0:8081->8080/tcp, :::8081->8080/tcp  mytedy02
c3bd6681e319  teedynew   "bin/jetty.sh run"  46 seconds ago Up 44 seconds  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp  mytedy
se@LAPTOP-EOBHBGLP:~/teedy$
```



Part 5 Build a SpringBoot Image

1. Create a springBoot project

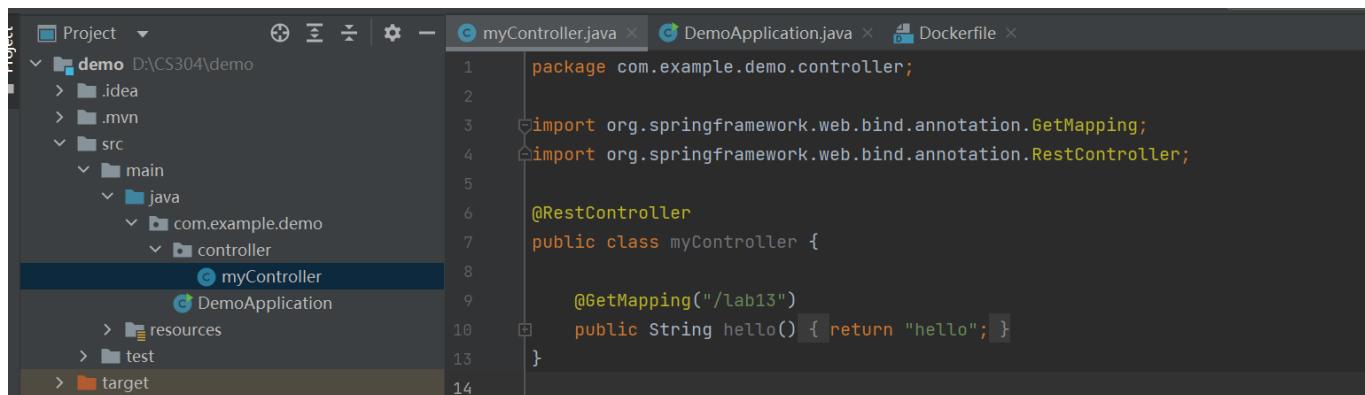
please refer lab 10 or follow below: download project from :<https://start.spring.io/>



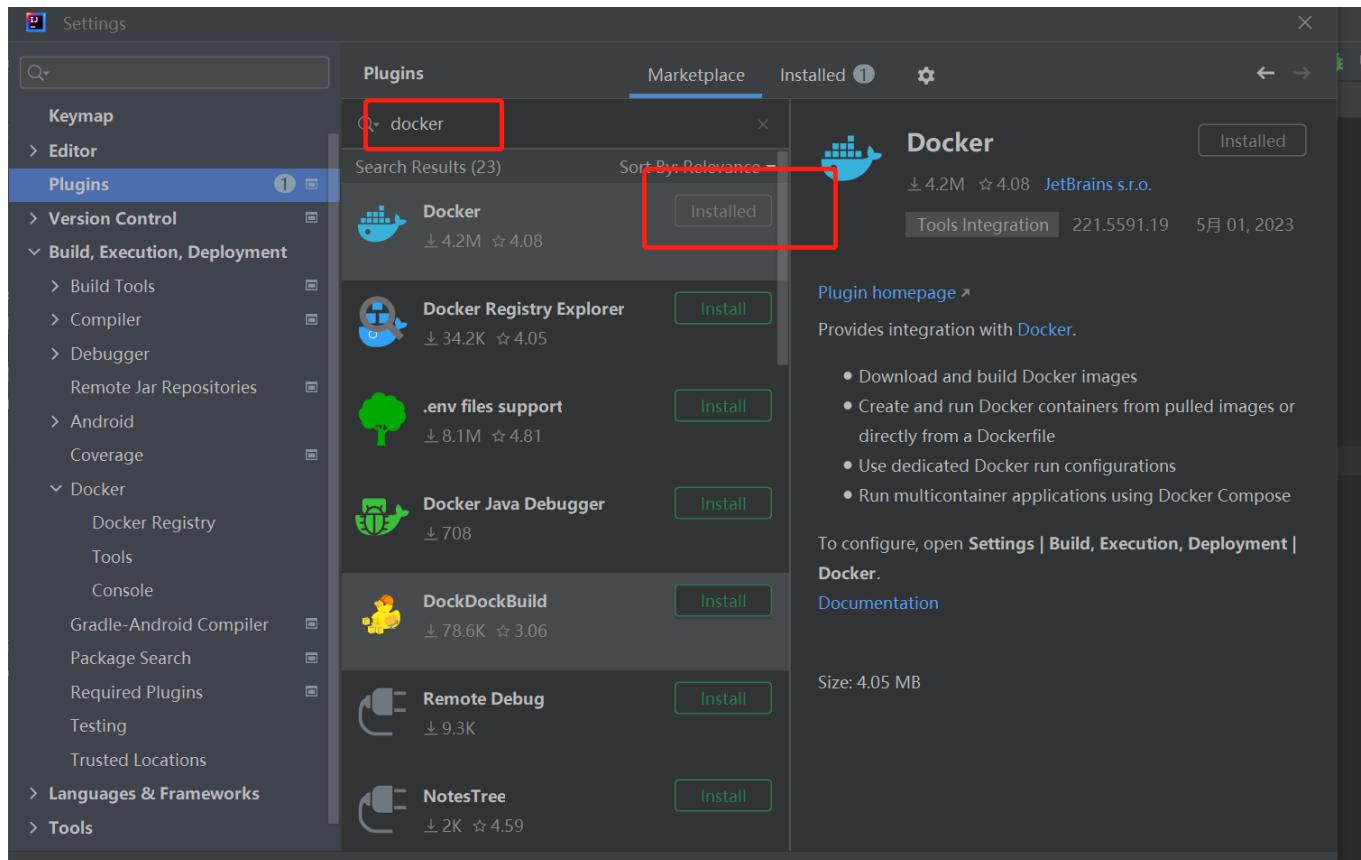
open in idea, write below in myController.java, make you can run it.

```
@RestController
public class myController {

    @GetMapping("/lab13")
    public String hello(){
        return "hello";
    }
}
```



install docker plugin:

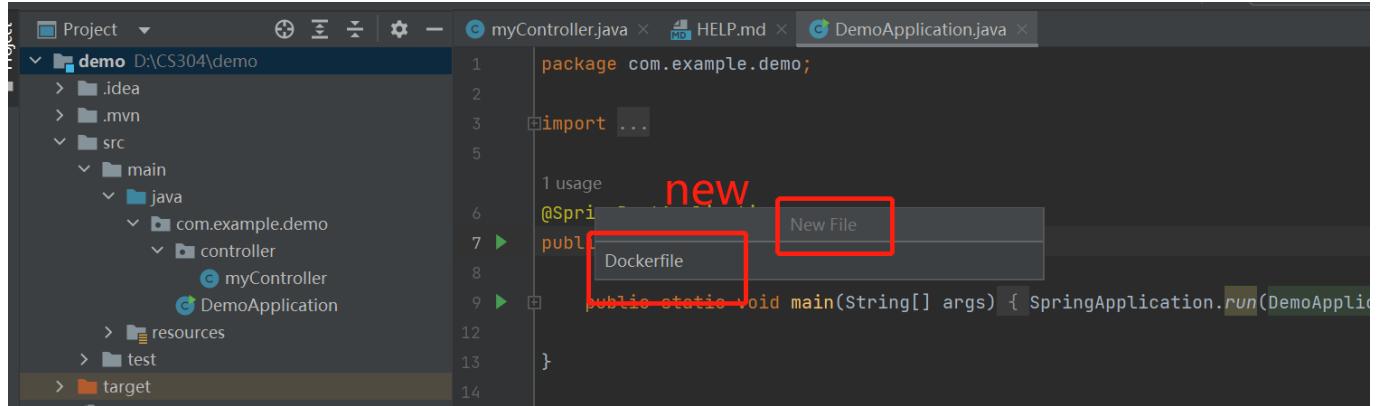


Create a new file named **Dockerfile**

```
FROM openjdk:8

COPY *.jar /app.jar
CMD ["--server.port=8080"]

EXPOSE 8080
ENTRYPOINT ["java","-jar","/app.jar"]
```



The screenshot shows the IntelliJ IDEA interface with a Java file named `myController.java` open. A code completion dropdown is displayed at the bottom of the editor, with the word `Dockerfile` highlighted. A red box surrounds the dropdown. Another red box highlights the `New File` option in the dropdown.

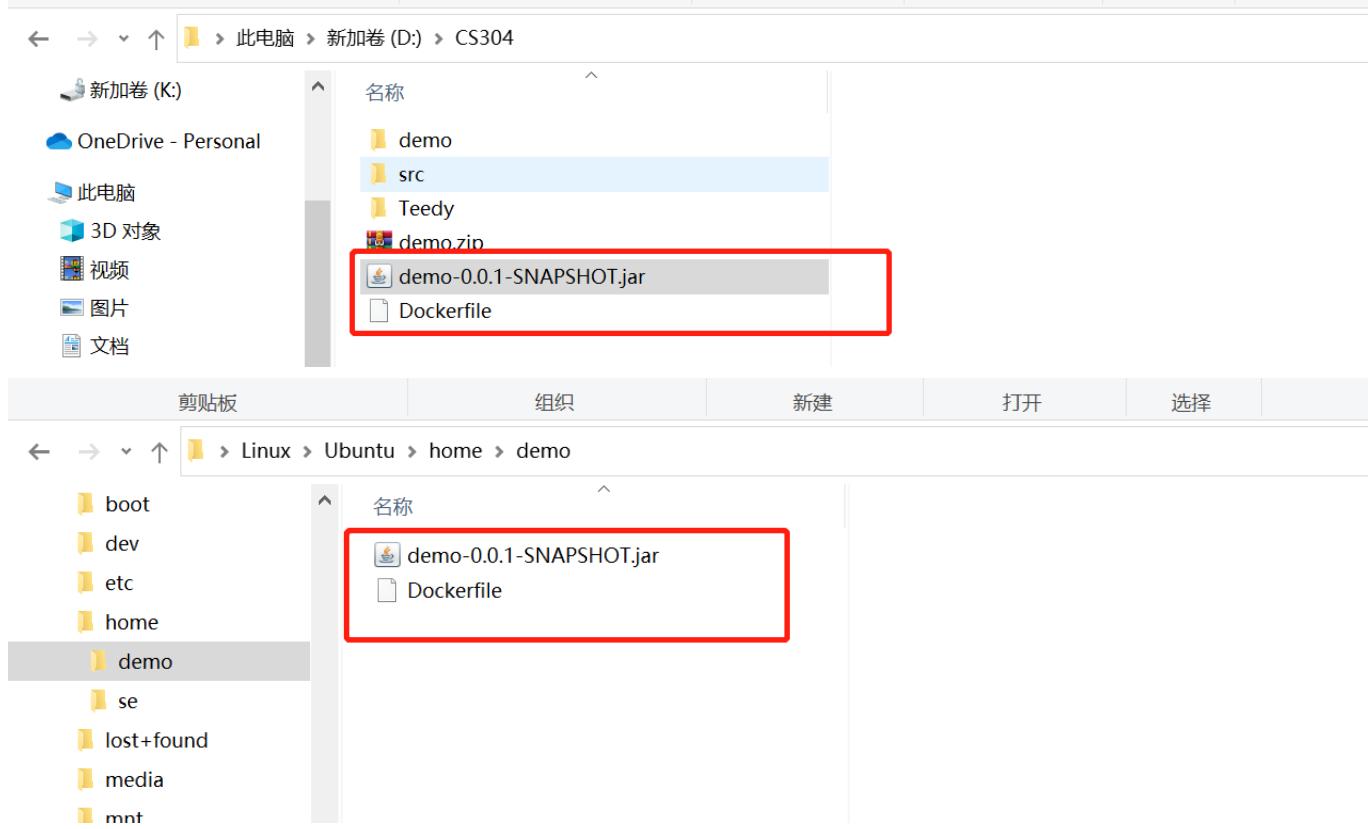
Below the editor, the project structure is visible, showing a `Dockerfile` file in the `target` directory. The `Dockerfile` content is as follows:

```
FROM openjdk:8
COPY *.jar /app.jar
CMD [ "--server.port=8080"]
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

package demo then you will get a file `demo-0.0.1-SNAPSHOT.jar`, with this two files,

maybe you need more knowledge about Linux command:<https://www.runoob.com/linux/linux-tutorial.html>

```
cp /mnt/d/cs304/Dockerfile /home/demo
```



build your own image:

```
docker build -t lab13_demo /home/demo
```

check your images:

```
docker images
```

run your images and named your running container:

```
docker run -d -P --name lab13_demo_t1 lab13_demo
```

check your program:

```
curl localhost:32768/lab13
```

```
root@LAPTOP-EOBHBGLP:~# docker build -t lab13_demo /home/demo
[+] Building 30.0s (7/7) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 241B
=> [internal] load metadata for docker.io/library/openjdk:8
=> [internal] load build context
=> => transferring context: 17.69MB
=> [1/2] FROM docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452c 17.6s
=> => resolve docker.io/library/openjdk:8@sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb 0.1s
=> sha256:d9d4b9b6e964657da49910b495173d6c4f0d9bc47b3b44273cf82fd32723d165 5.16MB / 5.16MB 0.1s
=> sha256:b273004037cc3af245d8e08cfbfa672b93ee7dcbb289736c82d0b59936fb71702 7.81kB / 7.81kB 3.8s
=> sha256:3af2ac94130765b73fc8f1b42ffc04f77996ed8210c297fcfa28ca880ff0a217 1.79kB / 1.79kB 0.0s
=> sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452 55.00MB / 55.00MB 3.4s
=> sha256:2068746827ec1b043b571e4788693eab7e9b2a95301176512791f8c317a2816a 10.88MB / 10.88MB 4.1s
=> sha256:86e863cc57215cfb181bd319736d0baf625fe8f150577f9eb58bd937f5452cb8 1.04kB / 1.04kB 0.0s
=> sha256:9daef329d35093868ef75ac8b7c6eb407fa53abccb3a264c218c2ec7bca716e6 54.58MB / 54.58MB 7.5s
=> => extracting sha256:001c52e26ad57e3b25b439ee0052f6692e5c0f2d5d982a00a8819ace5e521452 2.7s
=> => sha256:d85151f15b6683b98f21c3827ac545188b1849efb14a1049710ehbc4692de3dd5 5.42MB / 5.42MB 5.5s
```

```
root@LAPTOP-EOBHBGLP:~#
root@LAPTOP-EOBHBGLP:~# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
lab13_demo latest 869d9532ba4a 3 hours ago 544MB
ubuntu 22.04 08d22c0ceb15 8 weeks ago 77.8MB
hello-world latest fah5d9fea6a5 19 months ago 13.3kB
root@LAPTOP-EOBHBGLP:~# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
8e56be6ee2e9 lab13_demo "java -jar /app.jar ..." 3 hours ago Up 3 hours 0.0.0.0:32768->8080/tcp, :::32768->8080/tcp lab13_demo_t1
root@LAPTOP-EOBHBGLP:~# curl localhost:32768
{"timestamp": "2023-05-04T06:00:00+00:00", "status": 404, "error": "Not Found", "path": "/"}
root@LAPTOP-EOBHBGLP:~# curl localhost:32768/lab13
curl: (3) URL using bad/illegal format or missing URL
root@LAPTOP-EOBHBGLP:~# curl localhost:32768/lab13
helloroot@LAPTOP-EOBHBGLP:~#
```