Applied Computer Science, Faculty of Science, KMUTT

CSS 223 Operating Systems (2566/1)

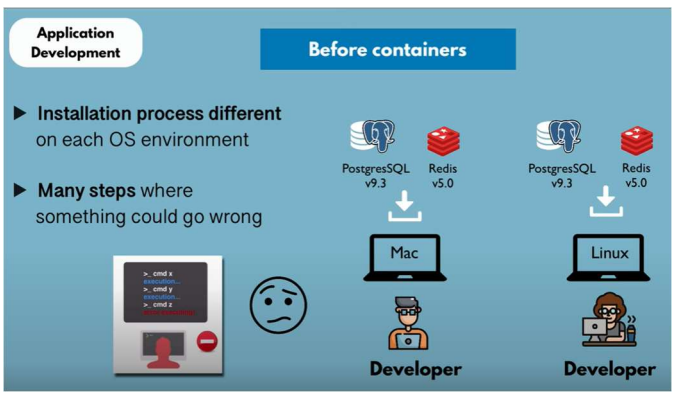
Chukiat Worasucheep

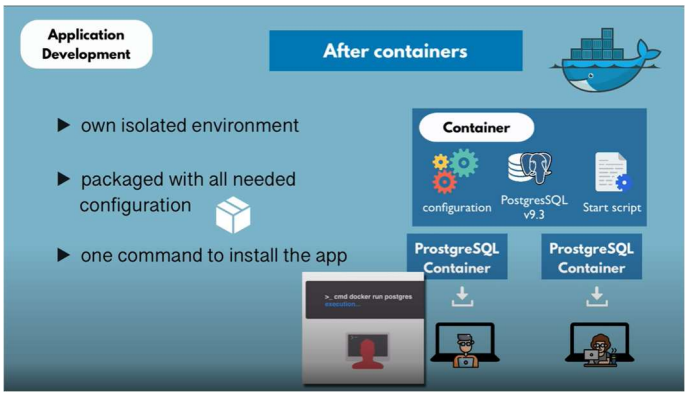
Basic Docker Container and Linux

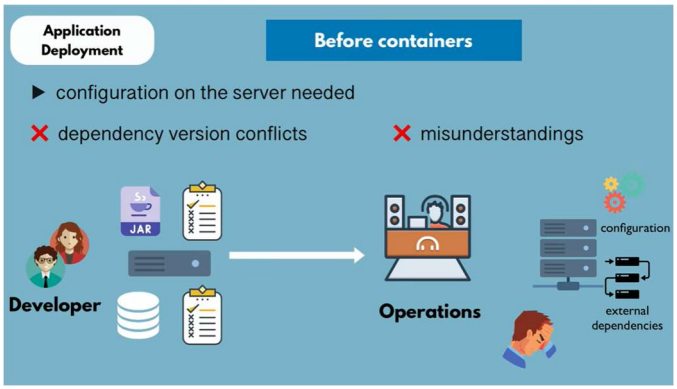
Part 0 Introduction to Docker Containerization

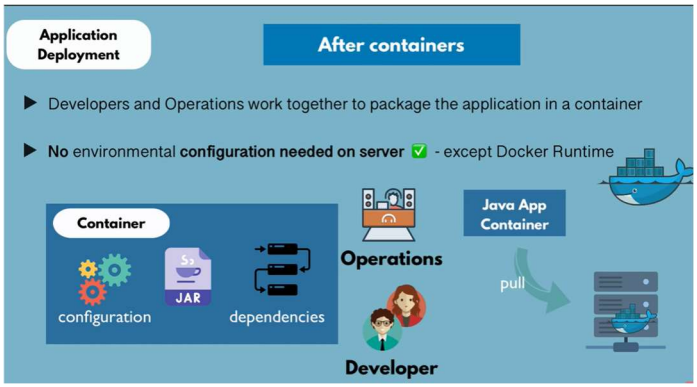
Docker is *an open-source project that automates the deployment of software applications inside* ***containers*** *by providing an additional layer of abstraction and automation of* ***OS-level virtualization*** *on Linux.*

The key benefit of Docker is that it allows users to **package an application with all of its dependencies into a standardized unit** for software development. Unlike virtual machines, containers do not have high overhead and hence enable more efficient usage of the underlying system and resources.









What are containers?

A Docker container is a lightweight, stand-alone, and executable package that includes everything needed to run an application. This includes the application code, system tools, libraries, and runtime. Containers are isolated from each other and provide an environment that is more secure and efficient than traditional virtual machines.

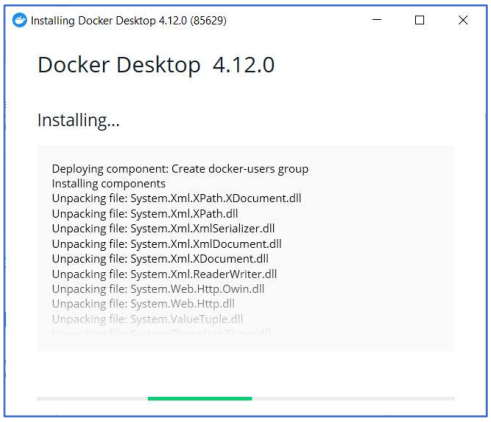
Why use containers?

Some of the main advantages of using containers include:

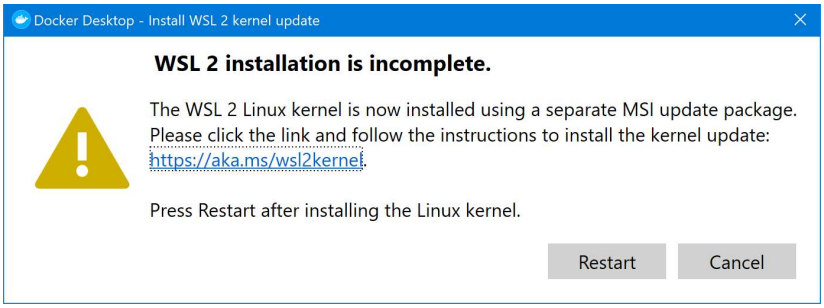
- *Efficiency*: Containers use fewer resources than traditional virtual machines, because they do not require a separate operating system for each application. This makes them more lightweight and allows them to start faster and use less memory.
- *Isolation*: Containers provide isolation at the application level, meaning that each container runs in its own environment and does not have access to the host system's resources or other containers. This makes them more secure, because a problem with one container does not affect the others.
- *Portability*: Containers can be easily moved between different environments, such as from a developer's laptop to a staging or production server. This makes it easier to deploy and manage applications.
- *Scalability*: Containers can be easily scaled up or down to meet the changing needs of an application. This makes them well-suited for use in cloud computing environments, where applications may need to scale quickly to meet demand.
- *Ease of use*: Containers can be easily created and managed using tools like Docker, which provide a standard format and API for working with containers. This makes it easier for developers to collaborate and deploy applications.

Part 1 Installation of Docker Desktop

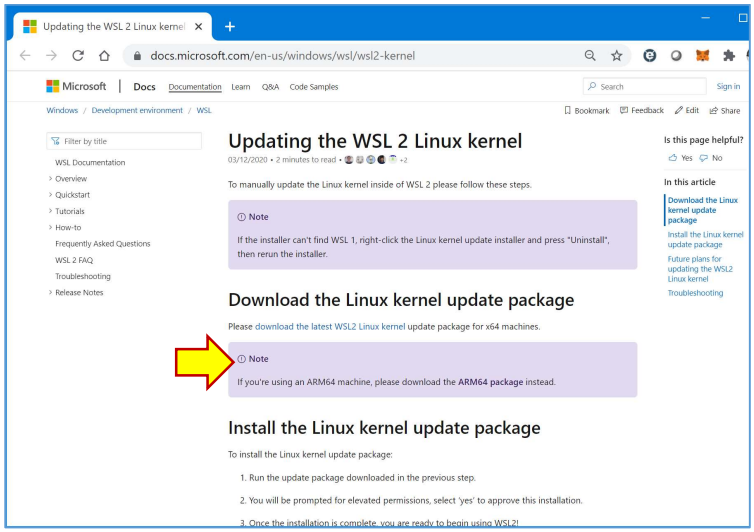
Go to web of Docker to install *Docker Desktop* onto your PC.

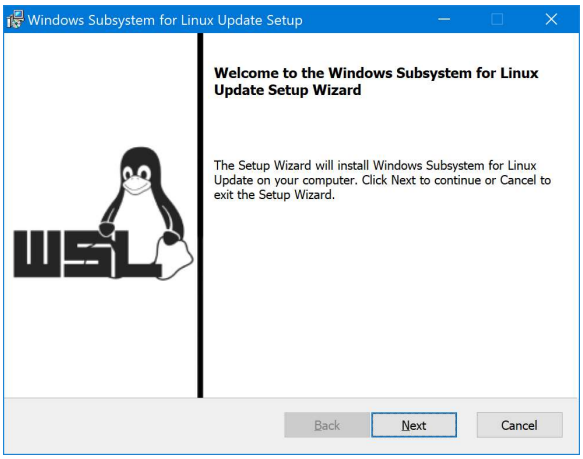


เมื่อ install แล้วอาจพบปัญหาว่า WSL 2 Installation is incomplete... ดังรูปต่อไปนี้

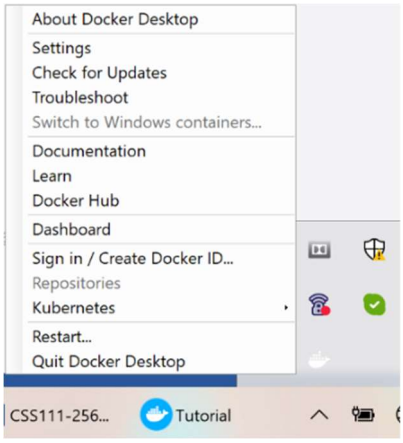


หากพบดังรูป ให้ click ไปที่ link aka.ms/wsl2kernel เพื่อติดตั้ง WSL2 Linux Kernel ให้เรียบร้อย (หลังการ restart เครื่อง)

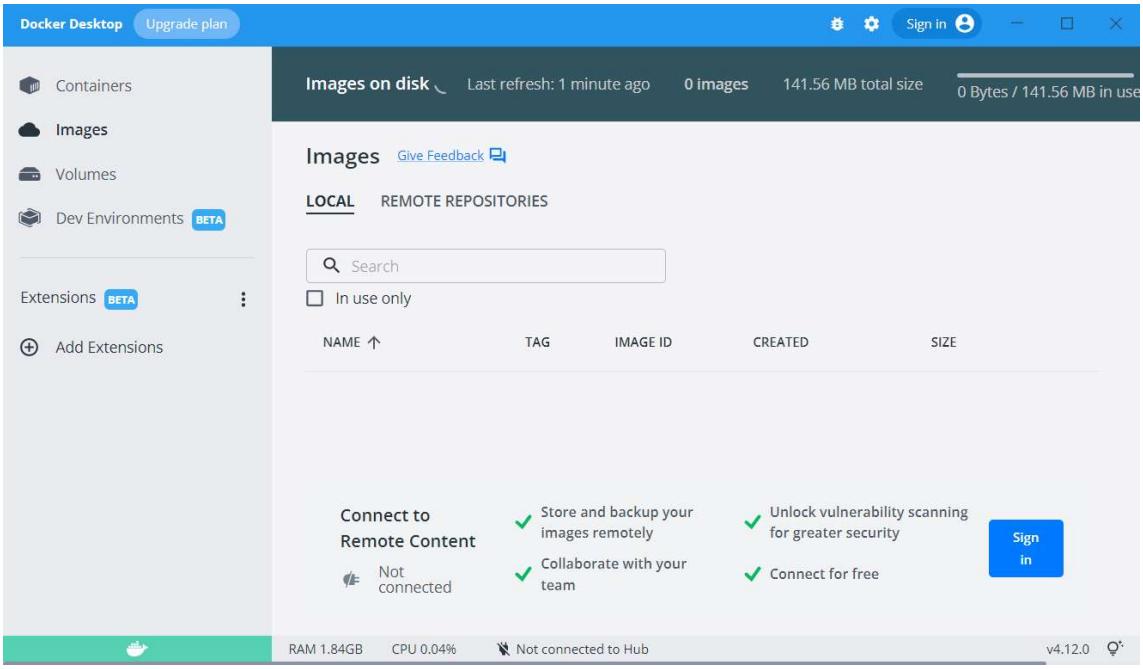




เมื่อติดตั้งแล้วเสร็จจะพบ Docker icon แจ้งว่ากำลัง Starting....



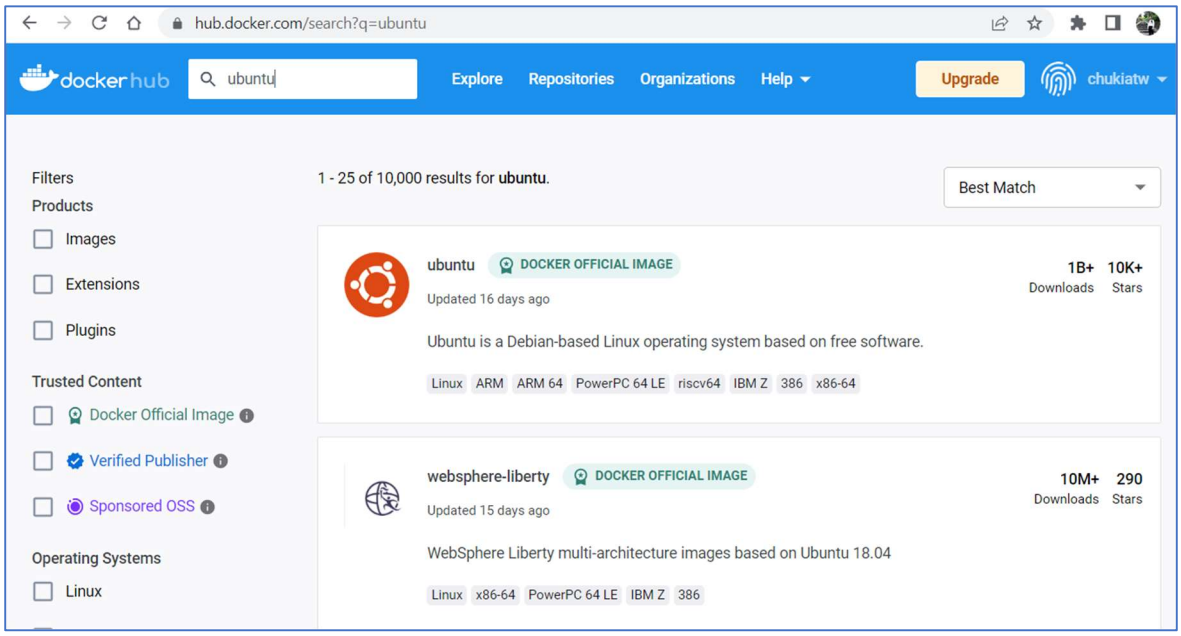
เมื่อ Docker start เรียบร้อย ให้ลอง click ขวาที่ icon นั้น จะพบเมนูคำสั่ง *Docker Desktop* ดังรูปด้านขวา. ตอนนี้ สั่งรันให้ทำงาน ก็จะเห็นหน้าจอ Docker Desktop ดังต่อไปนี้



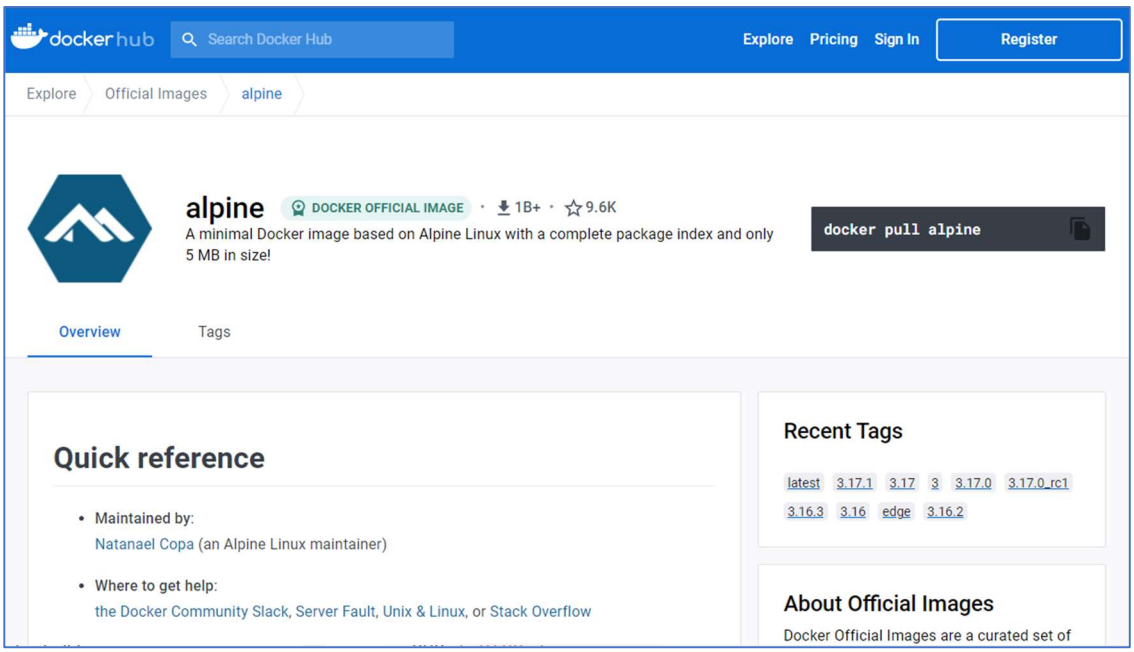
Terminology

- *Images* - The blueprints of our application which form the basis of containers. In the demo, we used the `docker pull` command to download the *alpine* image.
- *Containers* - Created from Docker images and run the actual application. We create a container using `docker run` which we did using the alpine image that we downloaded. A list of running containers can be seen using the `docker ps` command.
- *Docker Daemon* - The background service running on the host that manages building, running and distributing Docker containers. The daemon is the process that runs in the operating system to which clients talk to.
- *Docker Hub* - A registry of Docker images. The registry serves a directory of all available Docker images. If required, one can host their own Docker registries and can use them for pulling images.

Docker hub



Both Alpine and BusyBox are lightweight, minimalistic options for creating small Docker images, but Alpine Linux is more popular due to its package manager and security features.



Basic Architecuterr of Docker

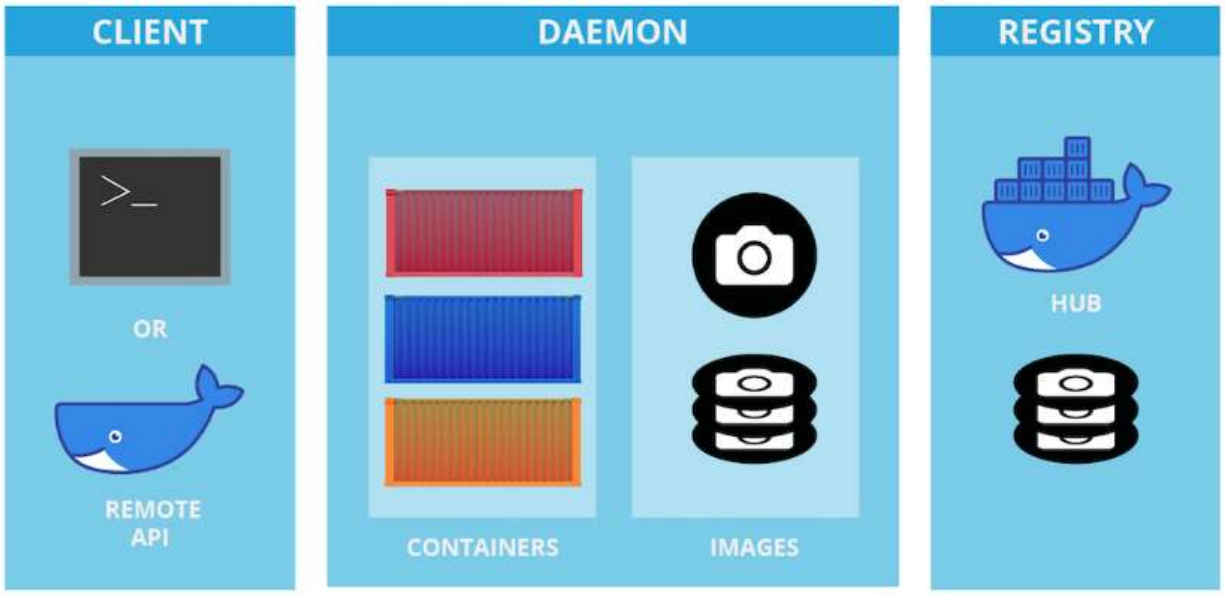
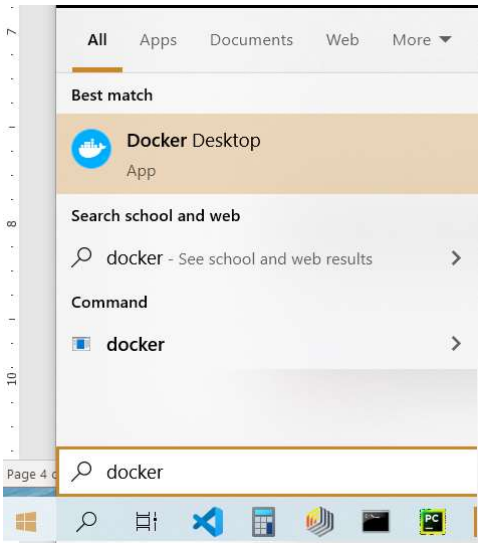


Image Source: https://devopslearners.com/the-architecture-of-docker-engine-a0a6e5ad4de0

Part 2: Basic Docker usage

Once you are done installing Docker, to get started, let's run the *Docker Desktop* in your Windows terminal (cmd program):




```
D:\docker>docker version
Client:
 Cloud integration: v1.0.29
 Version:           20.10.17
 API version:       1.41
 Go version:        go1.17.11
 Git commit:        100c701
 Built:             Mon Jun  6 23:09:02 2022
 OS/Arch:           windows/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop 4.12.0 (85629)
 Engine:
  Version:          20.10.17
  API version:      1.41 (minimum version 1.12)
  Go version:       go1.17.11
  Git commit:       a89b842
  Built:            Mon Jun  6 23:01:23 2022
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.6.8
  GitCommit:        9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6
 runc:
  Version:          1.1.4
  GitCommit:        v1.1.4-0-g5fd4c4d
```

```
 docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
```

Alpine Linux is a lightweight Linux distribution that is commonly used as the base image for Docker containers. It's smaller than other popular Linux distributions like Ubuntu or Fedora. The Alpine Linux distribution is also known for its security features, which make it a popular choice for running containers in production environments.

```
D:\docker>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest

D:\docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
alpine       latest    042a816809aa   12 days ago   7.05MB
busybox      latest    66ba00ad3de8   2 weeks ago   4.87MB
ubuntu       latest    6b7dfa7e8fdb   6 weeks ago   77.8MB
mysql        latest    7484689f290f   6 weeks ago   538MB
```

The *docker images* is a command used to list all the *images* that are currently available on a Docker host. This command will show the repository name, tag, image ID, created date and size of the image.

```
D:\docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
alpine       latest    042a816809aa   12 days ago   7.05MB
```

The *docker ps* is a command used to list the existing Docker *containers* on a system. The command will display information about each running container, including the container ID, image name, command that is running, the container's status, and the ports that are exposed.

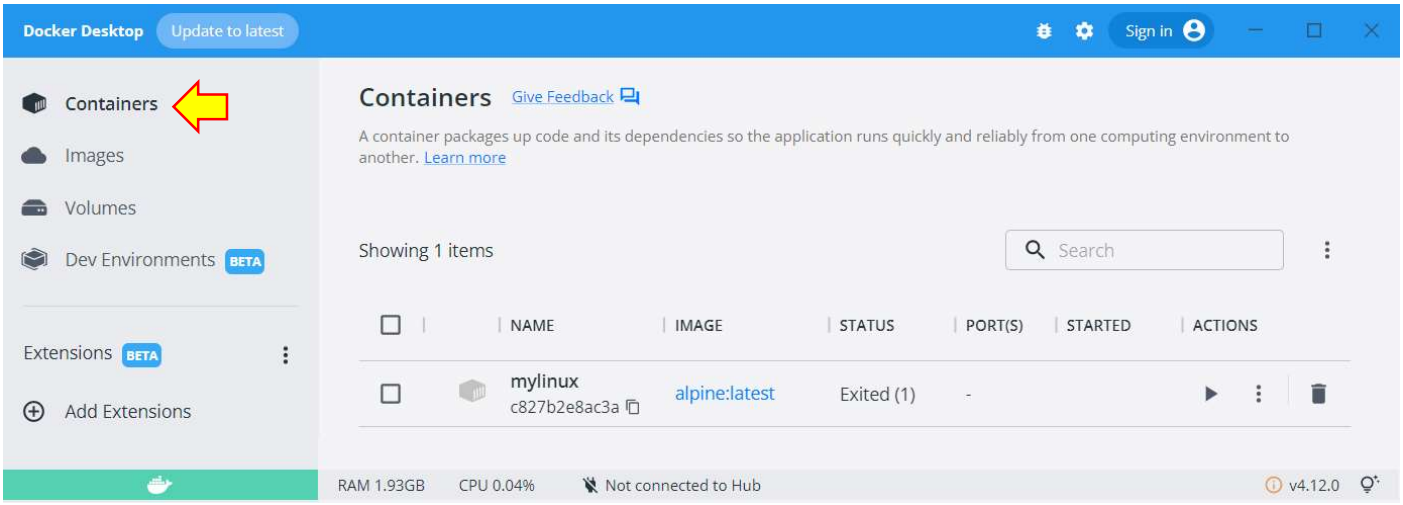
```
D:\docker>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

D:\docker>docker run alpine

D:\docker>docker ps -a
CONTAINER ID   IMAGE     COMMAND     CREATED         STATUS                     PORTS     NAMES
955f251065ab   alpine    "/bin/sh"   5 seconds ago   Exited (0) 4 seconds ago             modest_mcnulty

D:\docker>docker run -it --name mylinux -h chukiat-99 alpine
/ # uname -a
Linux chukiat-99 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021
x86_64 Linux
/ # ps -a
PID   USER     TIME  COMMAND
    1 root      0:00 /bin/sh
    8 root      0:00 ps -a
/ # ls -l
total 56
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 bin
drwxr-xr-x    5 root     root           360 Jan 22 14:37 dev
```

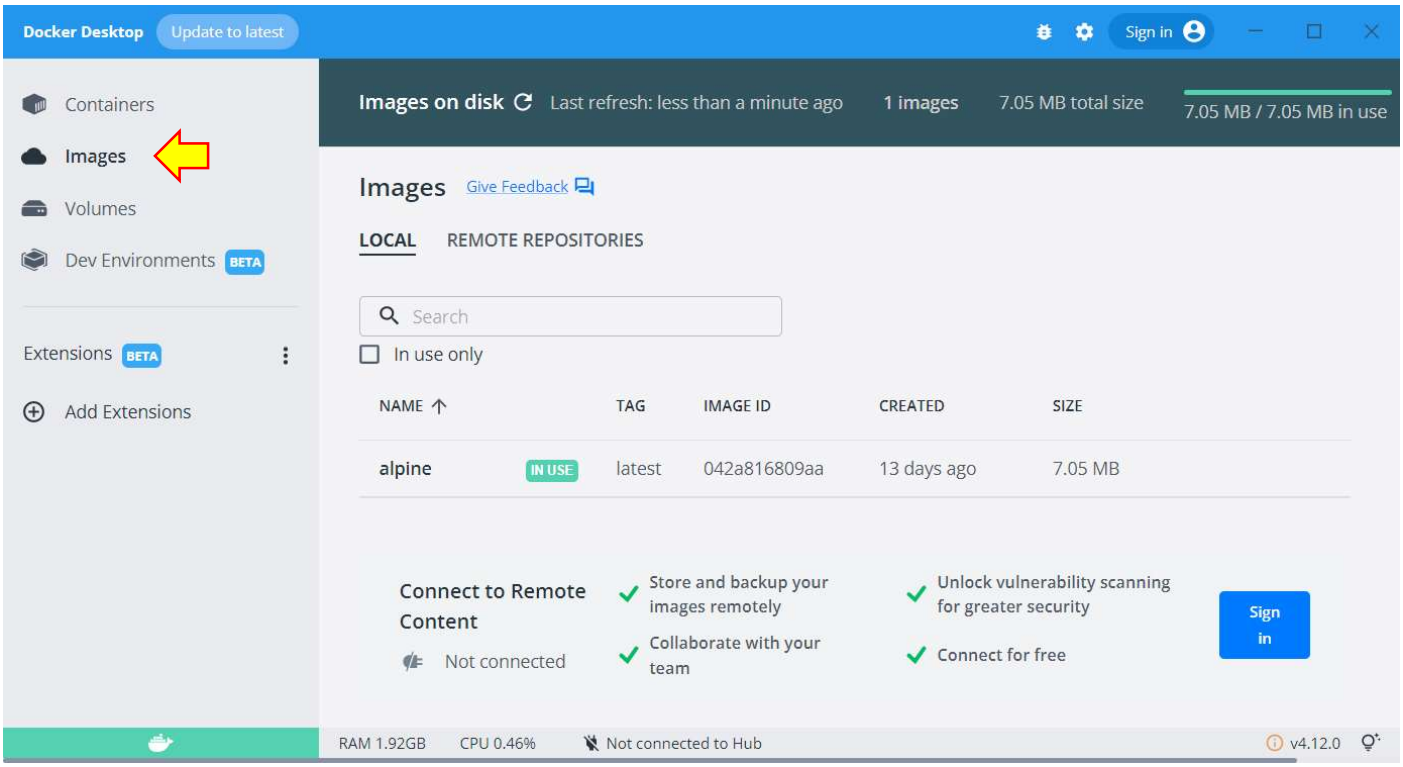
```
drwxr-xr-x    1 root     root          4096 Jan 22 14:37 etc
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 home
drwxr-xr-x    7 root     root          4096 Jan  9 12:46 lib
drwxr-xr-x    5 root     root          4096 Jan  9 12:46 media
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 mnt
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 opt
dr-xr-xr-x  296 root     root             0 Jan 22 14:37 proc
drwx------    1 root     root          4096 Jan 22 14:37 root
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 run
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 sbin
drwxr-xr-x    2 root     root          4096 Jan  9 12:46 srv
dr-xr-xr-x   11 root     root             0 Jan 22 14:37 sys
drwxrwxrwt    2 root     root          4096 Jan  9 12:46 tmp
drwxr-xr-x    7 root     root          4096 Jan  9 12:46 usr
drwxr-xr-x   12 root     root          4096 Jan  9 12:46 var
/ # pwd
/
/ # cd
~ # ls -l
total 0
~ # pwd
/root
~ # cc
/bin/sh: cc: not found

~ # apk add build-base   # install commonly-used packages in Alpine Linux
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.17/community/x86_64/APKINDEX.tar.gz
(1/20) Installing libgcc (12.2.1_git20220924-r4)
(2/20) Installing libstdc++ (12.2.1_git20220924-r4)
(3/20) Installing binutils (2.39-r2)
(4/20) Installing libmagic (5.43-r0)
(5/20) Installing file (5.43-r0)
(6/20) Installing libgomp (12.2.1_git20220924-r4)
(7/20) Installing libatomic (12.2.1_git20220924-r4)
(8/20) Installing gmp (6.2.1-r2)
(9/20) Installing isl25 (0.25-r0)
(10/20) Installing mpfr4 (4.1.0-r0)
(11/20) Installing mpc1 (1.2.1-r1)
(12/20) Installing gcc (12.2.1_git20220924-r4)
(13/20) Installing libstdc++-dev (12.2.1_git20220924-r4)
(14/20) Installing musl-dev (1.2.3-r4)
(15/20) Installing libc-dev (0.7.2-r3)
(16/20) Installing g++ (12.2.1_git20220924-r4)
(17/20) Installing make (4.3-r1)
(18/20) Installing fortify-headers (1.1-r1)
(19/20) Installing patch (2.7.6-r8)
(20/20) Installing build-base (0.5-r3)
Executing busybox-1.35.0-r29.trigger
OK: 244 MiB in 35 packages
~ # cc
cc: fatal error: no input files
compilation terminated.
~ # exit
```

```
D:\docker>docker ps
CONTAINER ID    IMAGE         COMMAND       CREATED       STATUS       PORTS        NAMES
```

Since no containers are running, we see a blank line. Let's try a more useful variant: docker ps -a

```
D:\docker>docker ps -a
CONTAINER ID   IMAGE     COMMAND     CREATED           STATUS                       PORTS   NAMES
c827b2e8ac3a   alpine    "/bin/sh"   19 minutes ago    Exited (1) 9 minutes ago             mylinux
```





The *docker rm* command is used for removal of one or more containers.
Use option *-f* to force the removal of a running container (uses SIGKILL).

```
D:\docker>docker rm mylinux
mylinux

D:\docker>docker ps -a
CONTAINER ID    IMAGE         COMMAND       CREATED       STATUS       PORTS        NAMES
```

```
D:\docker>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
alpine       latest    042a816809aa   12 days ago   7.05MB
```

The '*docker image rm*' or '*docker rmi*' command is used for removal of one or more images.

```
D:\docker>docker image rm alpine
Untagged: alpine:latest
Untagged:
alpine@sha256:f271e74b17ced29b915d351685fd4644785c6d1559dd1f2d4189a5e851ef753a
Deleted: sha256:042a816809aac8d0f7d7cacac7965782ee2ecac3f21bcf9f24b1de1a7387b769
Deleted: sha256:8e012198eea15b2554b07014081c85fec4967a1b9cc4b65bd9a4bce3ae1c0c88

D:\docker>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

D:\docker>
```

You can simply run the `docker container prune` command to remove all stopped containers.

```
D:\docker>docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
4a7f7eebae0f63178aff7eb0aa39f0627a203ab2df258c1a00b456cf20063
f98f9c2aa1eaf727e4ec9c0283bcaa4762fbdba7f26191f26c97f64090360

Total reclaimed space: 212 B
```

Part 3 Basic Ubuntu Linux inside Docker

Start by calling cmd or command of Windows 10

```
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.
```

- เรียกใช้ Ubuntu บน Docker พร้อมทั้งให้เรียก bash shell ขึ้นใช้งานในนั้น
- กำหนด hostname เป็น yourname ทั้งนี้ ให้ yourname เป็นชื่อต้น-ID สองหลักท้าย เช่น chukiat-99

```
D:\>docker run -h chukiat-99 -it ubuntu bash
root@chukiat-99:/# date
Tue Sep 13 06:35:13 UTC 2022
root@chukiat-99:/# uname
Linux
root@chukiat-99:/# uname -a
Linux chukiat-99 5.10.16.3-microsoft-standard-WSL2 #1 SMP Fri Apr 2 22:23:49 UTC 2021
x86_64 x86_64 x86_64 GNU/Linux

root@chukiat-99:/# ls -a
.   .dockerenv  boot  etc   lib    lib64   media  opt   root  sbin  sys  usr
..  bin         dev   home  lib32  libx32  mnt    proc  run   srv   tmp  var
root@chukiat-99:/# ls -l
total 48
lrwxrwxrwx   1 root root    7 Aug 15 11:50 bin -> usr/bin
drwxr-xr-x   2 root root 4096 Apr 18 10:28 boot
drwxr-xr-x   5 root root  360 Sep 13 06:35 dev
drwxr-xr-x   1 root root 4096 Sep 13 06:35 etc
drwxr-xr-x   2 root root 4096 Apr 18 10:28 home
```

ดู Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, OS

list all items including hidden items (beginning with dot .)

```
lrwxrwxrwx   1 root root    7 Aug 15 11:50 lib -> usr/lib
lrwxrwxrwx   1 root root    9 Aug 15 11:50 lib32 -> usr/lib32
lrwxrwxrwx   1 root root    9 Aug 15 11:50 lib64 -> usr/lib64
lrwxrwxrwx   1 root root   10 Aug 15 11:50 libx32 -> usr/libx32
drwxr-xr-x   2 root root 4096 Aug 15 11:50 media
drwxr-xr-x   2 root root 4096 Aug 15 11:50 mnt
drwxr-xr-x   2 root root 4096 Aug 15 11:50 opt
dr-xr-xr-x 299 root root    0 Sep 13 06:35 proc
drwx------   2 root root 4096 Aug 15 11:53 root
drwxr-xr-x   5 root root 4096 Aug 15 11:53 run
lrwxrwxrwx   1 root root    8 Aug 15 11:50 sbin -> usr/sbin
drwxr-xr-x   2 root root 4096 Aug 15 11:50 srv
dr-xr-xr-x  11 root root    0 Sep 13 06:35 sys
drwxrwxrwt   2 root root 4096 Aug 15 11:53 tmp
drwxr-xr-x  14 root root 4096 Aug 15 11:50 usr
drwxr-xr-x  11 root root 4096 Aug 15 11:53 var

root@chukiat-99:/# pwd
/
root@chukiat-99:/# cd
root@chukiat-99:~# pwd
/root
root@chukiat-99:~# ls
root@chukiat-99:~# mkdir mydir
root@chukiat-99:~# cd mydir
root@chukiat-99:~/mydir# cat /etc/passwd > etc.passwd
root@chukiat-99:~/mydir# ls -l
total 4
-rw-r--r-- 1 root root 922 Sep 13 06:36 etc.passwd

root@chukiat-99:~/mydir# more etc.passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin

root@chukiat-99:~/mydir# cd ..
root@chukiat-99:~# rm -r mydir
root@chukiat-99:~# ls -l
total 0
```

Print working directory

Change to home directory

Make or create a directory (folder)

Redirect standard output to a file

View contain of a file like cat, but page by page.

Remove recursively

```
root@chukiat-99:~# ps
  PID TTY          TIME CMD
    1 pts/0    00:00:00 bash
   18 pts/0    00:00:00 ps
root@chukiat-99:~# ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 06:39 pts/0    00:00:00 bash
root          19     1  0 06:39 pts/0    00:00:00 ps -ef
root@chukiat-99:~# ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.5  0.0   4624  3796 pts/0    Ss   06:39   0:00 bash
root          20  0.0  0.0   7056  1652 pts/0    R+   06:39   0:00 ps -aux
```

View process status

```
root@chukiat-99:~# history
    1  date
    2  uname
    3  uname -a
    4  ls -a
    5  ls -l
    6  pwd
    7  cd
    8  pwd
    9  ls
   10  mkdir mydir
   11  cd mydir
   12  cat /etc/passwd > etc.passwd
   13  ls -l
   14  more etc.passwd
   15  cd ..
   16  rm -r mydir
   17  ls -l
   18  ps
   19  ps -ef
   20  ps -aux
   21  history
```

ดูรายการคำสั่งทั้งหมดที่เรียกไป ในการ login session ปัจจุบัน

Where are the wget and sudo?

apt stands for "Advanced Package Tool", and it is a powerful *package manager* for managing packages (applications and software libraries) on a Linux system. apt can be used to install, remove, and update packages, as well as to upgrade the entire system.

apt-get is a command-line tool that is used to install, remove, and *manage packages* on a Linux system. It is similar to apt, but apt-get is more powerful and has more options.

wget is a command-line utility that is used to *download* files from the internet. It is commonly used to download files from the command line, but it can also be used in scripts to download files automatically.

sudo is a command-line utility that allows users to run programs with the security privileges of another user, usually the superuser (*root*), the special user account on a Unix-like system that has administrative privileges and can perform actions that could potentially damage the system.

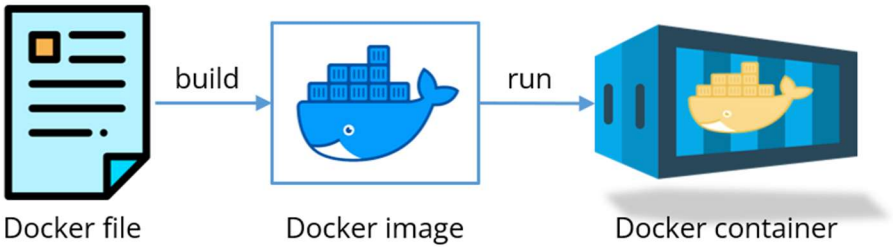
```
root@chukiat-99:/# wget
bash: wget: command not found

root@chukiat-99:/# sudo
bash: sudo: command not found

root@chukiat-99:/# apt-get update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4732 B]
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [593 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [266 kB]
Get:8 http://archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [17.5 MB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [781 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [667 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy/main amd64 Packages [1792 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy/restricted amd64 Packages [164 kB]
Get:13 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [973 kB]
Get:14 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [977 kB]
Get:15 http://archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [641 kB]
Get:16 http://archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [8150 B]
Get:17 http://archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [7291 B]
Get:18 http://archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3520 B]
Fetched 24.9 MB in 37s (668 kB/s)
Reading package lists... Done

root@chukiat-99:/# apt-get -y install wget sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ca-certificates libpsl5 openssl publicsuffix sudo wget
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 2702 kB of archives.
After this operation, 6485 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssl amd64 3.0.2-0ubuntu1.7
[1183 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates all
20211016ubuntu0.22.04.1 [144 kB]
Fetched 2702 kB in 14s (194 kB/s)
Selecting previously unselected package sudo.
Preparing to unpack .../2-sudo_1.9.9-1ubuntu2.1_amd64.deb ...
Unpacking sudo (1.9.9-1ubuntu2.1) ...
Selecting previously unselected package wget.
Preparing to unpack .../5-wget_1.21.2-2ubuntu1_amd64.deb ...
Unpacking wget (1.21.2-2ubuntu1) ...
Updating certificates in /etc/ssl/certs...
124 added, 0 removed; done.
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for ca-certificates (20211016ubuntu0.22.04.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

```
root@chukiat-99:/# wget
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.
root@chukiat-99:/# sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-ABbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h
host] [-p prompt] [-R directory] [-T timeout]
            [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-ABknS] [-r role] [-t type] [-C num] [-D directory] [-g group] [-h
host] [-p prompt] [-R directory] [-T timeout]
            [-u user] file ...

root@chukiat-99:~# exit
exit
```

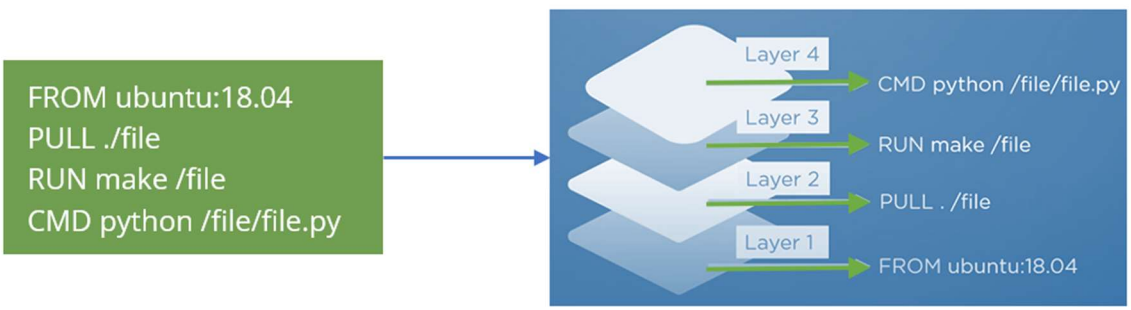
Part 4 Create, build and test a simple Dockerfile

A *dockerfile* is a text document that contains all the commands a user could call on the command line to assemble an image. The instructions in the Dockerfile specify what base image to use, what additional software to install, and how to configure the software. The docker build command builds an image from a Dockerfile. Using docker build, users can create an automated build that executes several command-line instructions in succession.

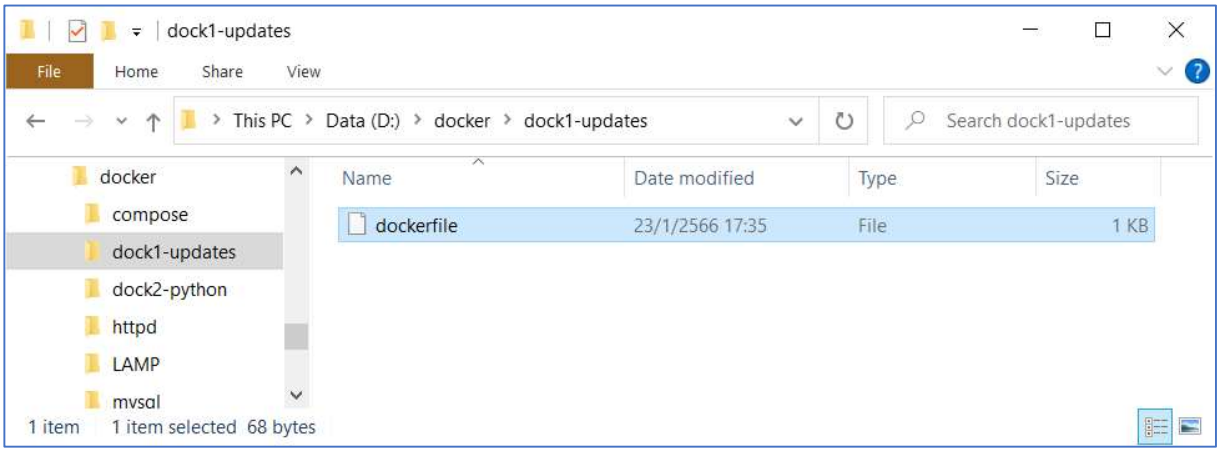
- Each layer is an image itself, just one without a human-assigned tag. They have auto-generated IDs though.
- Each layer stores the changes compared to the image it's based on.
- Each instruction in a Dockerfile results in a layer.




```
1  FROM python:3.5
2  RUN pip install Flask==0.11.1 redis==2.10.5
3  RUN useradd -ms /bin/bash admin
4  USER admin
5  COPY app /app
6  WORKDIR /app
7  CMD ["python", "app.py"]
```



Example 1 An Alpine image with automatic updates



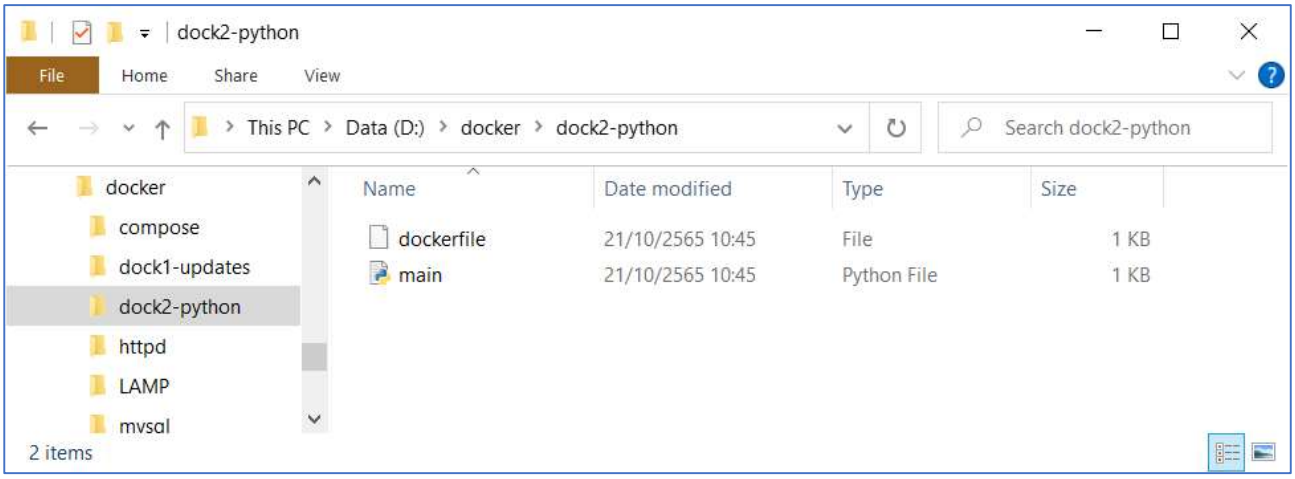
dockerfile

```
FROM alpine:latest
RUN apk update
RUN apk add curl wget sudo nginx
```

```
D:\docker\dock1-updates>docker build -t alpineupd .
[+] Building 0.1s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                    0.0s
 => => transferring dockerfile: 31B                                     0.0s
 => [internal] load .dockerignore                                       0.0s
 => => transferring context: 2B                                         0.0s
 => [internal] load metadata for docker.io/library/alpine:latest        0.0s
 => [1/3] FROM docker.io/library/alpine:latest                          0.0s
 => CACHED [2/3] RUN apk update                                         0.0s
 => CACHED [3/3] RUN apk add curl wget sudo nginx                       0.0s
 => exporting to image                                                  0.0s
 => => exporting layers                                                 0.0s
 => => writing image
sha256:861d61b1b3fff062a0ab61528e48439b32576cc7d672bd016efb3ca689454e4c 0.0s
 => => naming to docker.io/library/alpineupd                            0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn
how to fix them

D:\docker\dock1-updates>docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
alpineupd      latest    861d61b1b3ff   5 minutes ago    18.3MB
dock2          latest    27f7224fcc95   48 minutes ago   178MB
hello-docker   latest    250017d164a5   9 hours ago      47MB
alpine         latest    042a816809aa   13 days ago      7.05MB
ubuntu         latest    6b7dfa7e8fdb   6 weeks ago      77.8MB

D:\docker\dock1-updates>docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
alpineupd    latest    861d61b1b3ff   6 minutes ago    18.3MB
alpine       latest    042a816809aa   13 days ago      7.05MB
```

```
D:\docker\dock1-updates>docker run -it alpine
/ # curl
/bin/sh: curl: not found
/ # sudo
/bin/sh: sudo: not found
/ # exit

D:\docker\dock1-updates>docker run -it alpineupd
/ # curl
curl: try 'curl --help' or 'curl --manual' for more information
/ # sudo
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABkNnS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABkNnS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-ABbEHkNnPS] [-C num] [-D directory] [-g group] [-h host] [-p prompt] [-R
directory] [-T timeout] [-u user]
            [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-ABkNnS] [-C num] [-D directory] [-g group] [-h host] [-p prompt] [-R
directory] [-T timeout] [-u user]
            file ...
/ #
```

Example 2 Run a Python script with an Alpine image



dockerfile

```
FROM python:3.8.15-alpine3.15
ADD main.py /
CMD ["python", "./main.py"]
```

main.py

```
print('Hello world from inside a docker.')

for i in range(5):
   print(i, '\t', i*i)
```

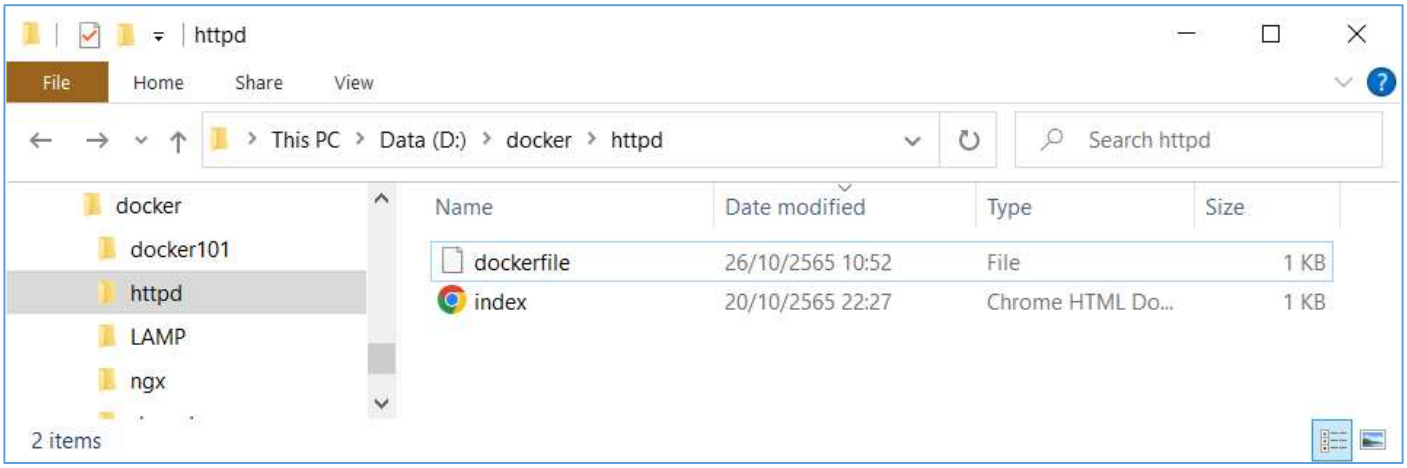
```
D:\docker\dock2-python>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    2dc39ba059dc   6 weeks ago   77.8MB

# Let's try disconnect from the Internet to see its error message.
D:\docker\dock2-python>docker build -t hello-docker .
[+] Building 0.1s (3/3) FINISHED
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 116B                                0.0s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => ERROR [internal] load metadata for docker.io/library/python:3.8.15-alpine3.15
0.0s
------
 > [internal] load metadata for docker.io/library/python:3.8.15-alpine3.15:
------
failed to solve with frontend dockerfile.v0: failed to create LLB definition: failed to
do request: Head "https://registry-1.docker.io/v2/library/python/manifests/3.8.15-
alpine3.15": Failed to lookup host: registry-1.docker.io

D:\docker\dock2-python>docker build -t hello-docker .
[+] Building 2.6s (7/7) FINISHED
 => [internal] load build definition from Dockerfile                0.0s
 => => transferring dockerfile: 31B                                 0.0s
 => [internal] load .dockerignore                                   0.0s
 => => transferring context: 2B                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.8.15-alpine3.15 2.4s
 => [internal] load build context                                   0.0s
 => => transferring context: 120B                                   0.0s
 => CACHED [1/2] FROM docker.io/library/python:3.8.15-
alpine3.15@sha256:f8b1331535670105735a5549f37b11a4b5d3a53e686391f8a75  0.0s
 => [2/2] ADD main.py /                                             0.0s
 => exporting to image                                              0.0s
 => => exporting layers                                             0.0s
 => => writing image
sha256:fddc9da952fbf444f7ba3f044b52db0bd6127ab695abd1a5450d518676479068 0.0s
 => => naming to docker.io/library/hello-docker                     0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn
how to fix them

D:\docker\dock2-python>docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
hello-docker   latest    fddc9da952fb   4 seconds ago    47MB
ubuntu         latest    2dc39ba059dc   6 weeks ago      77.8MB

D:\docker\dock2-python>docker run hello-docker
Hello world from inside a docker.
0       0
1       1
2       4
3       9
4       16
D:\docker\dock2-python>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

```
D:\docker\dock2-python>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                     PORTS
NAMES
cd5cd01d7f7f   hello-docker "python ./main.py" 3 minutes ago Exited (0) 3 minutes ago
wizardly_brahmagupta

D:\docker\dock2-python>docker rm wizardly_brahmagupta
wizardly_brahmagupta

D:\docker\dock2-python>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

D:\docker\dock2-python>docker history hello-docker
IMAGE          CREATED         CREATED BY                    SIZE      COMMENT
fddc9da952fb   5 minutes ago   CMD ["python" "./main.py"]   0B        buildkit.dockerfile.v0
<missing>      5 minutes ago   ADD main.py / # buildkit      86B       buildkit.dockerfile.v0

D:\docker\dock2-python>docker system df
TYPE            TOTAL     ACTIVE    SIZE      RECLAIMABLE
Images          2         0         124.8MB   124.8MB (100%)
Containers      0         0         0B        0B
Local Volumes   11        0         216.5MB   216.5MB (100%)
Build Cache     17        0         5.068kB   5.068kB

D:\docker\dock2-python>docker images
REPOSITORY     TAG       IMAGE ID       CREATED          SIZE
hello-docker   latest    fddc9da952fb   7 minutes ago    47MB
ubuntu         latest    2dc39ba059dc   6 weeks ago      77.8MB

D:\docker\dock2-python>docker run --name mypython hello-docker
Hello world from inside a docker.
0        0
1        1
2        4
3        9
4        16

D:\docker\dock2-python>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS
PORTS     NAMES
5943d548956b   hello-docker   "python ./main.py" 6 seconds ago   Exited (0) 4 seconds ago
mypython

D:\docker\dock2-python>docker rmi -f hello-docker
Untagged: hello-docker:latest
Deleted: sha256:fddc9da952fbf444f7ba3f044b52db0bd6127ab695abd1a5450d518676479068

D:\docker\dock2-python>docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
ubuntu       latest    2dc39ba059dc   6 weeks ago   77.8MB


D:\docker\dock2-python>docker system info
Client:
 Context:    default
```

```
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc., v0.9.1)
  compose: Docker Compose (Docker Inc., v2.10.2)
  extension: Manages Docker extensions (Docker Inc., v0.2.9)
  sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore
Inc., 0.6.0)
  scan: Docker Scan (Docker Inc., v0.19.0)

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 1
 Server Version: 20.10.17
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6
 runc version: v1.1.4-0-g5fd4c4d
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 5.10.16.3-microsoft-standard-WSL2
 Operating System: Docker Desktop
 OSType: linux
 Architecture: x86_64
 CPUs: 16
 Total Memory: 12.28GiB
 Name: docker-desktop
 ID: USUB:XDGF:AOL2:XPD3:MXGG:AX6C:DBJA:YDLQ:BPMK:YJNT:2T4Z:PC5X
 Docker Root Dir: /var/lib/docker
 Debug Mode: false
 HTTP Proxy: http.docker.internal:3128
 HTTPS Proxy: http.docker.internal:3128
 No Proxy: hubproxy.docker.internal
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
 Insecure Registries:
```

```
  hubproxy.docker.internal:5000
  127.0.0.0/8
 Live Restore Enabled: false

WARNING: No blkio throttle.read_bps_device support
WARNING: No blkio throttle.write_bps_device support
WARNING: No blkio throttle.read_iops_device support
WARNING: No blkio throttle.write_iops_device support
```

Part 5 Test a simple docker file with Apache httpd



dockerfile

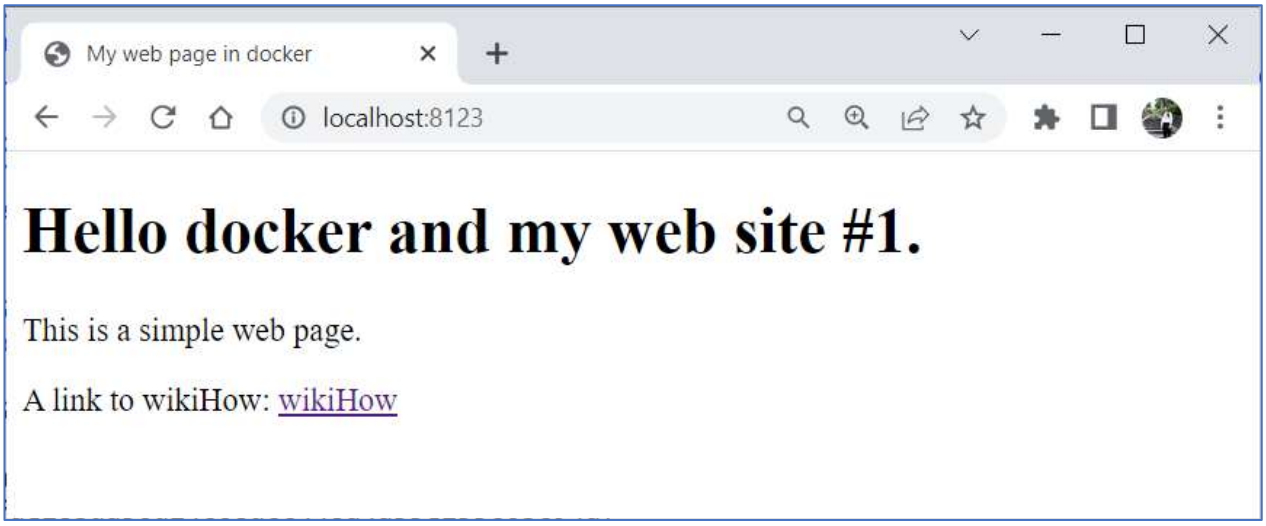
```
# A simple web page with Apache hhtpd
FROM httpd:2.4
LABEL AUTHOR=chukiatwr@gmail.com
LABEL VERSION=0.1
WORKDIR /usr/local/apache2
COPY index.html htdocs/index.html
```

index.html

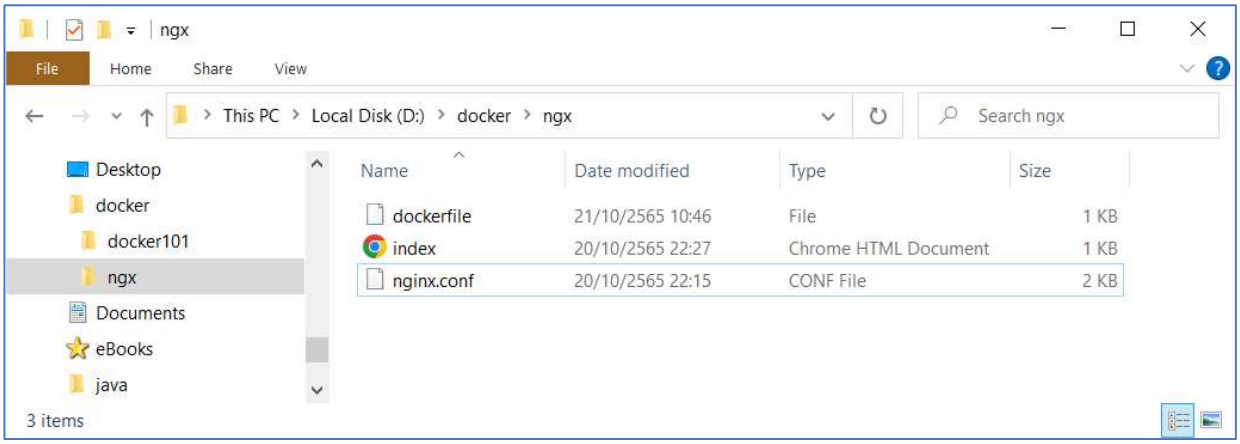
```
<!DOCTYPE html>
<html lang="en">
<head> <title>My web page in docker</title> </head>
<h1>Hello docker and my web site #1.</h1>
<p>This is a simple web page.</p>
<p>A link to wikiHow: <a href="http://www.wikihow.com">wikiHow</a></p>
</html>
```

```
D:\docker\httpd>docker build -t myhttpd .
[+] Building 3.6s (2/3)
 => [internal] load build definition from Dockerfile      0.0s
 => => transferring dockerfile: 214B                      0.0s
[+] Building 3.7s (2/3)
 => [internal] load build definition from Dockerfile      0.0s
[+] Building 3.9s (2/3)
 => [internal] load build definition from Dockerfile      0.0s
 => => transferring dockerfile: 214B                      0.0s
[+] Building 4.1s (4/7)
```

```
...
[+] Building 7.3s (8/8) FINISHED
 => [internal] load build definition from Dockerfile      0.0s
 => => transferring dockerfile: 214B                      0.0s
 => [internal] load metadata for docker.io/library/httpd:2.4  3.9s
 => [1/3] FROM docker.io/library/httpd:2.4@sha256:5fa965      3.1s>
 => [2/3] WORKDIR /usr/local/apache2                      0.2s
 => [3/3] COPY index.html htdocs/index.html               0.0s
 => exporting to image                                    0.0s
 => => exporting layers                                   0.0s
 => => writing image sha256:39c0bb0d64f904654ac5119d0d10  0.0s
 => => naming to docker.io/library/myhttpd                0.0s
 Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn
how to fix them

D:\docker\httpd>docker history myhttpd
IMAGE        CREATED        CREATED BY                                      SIZE  COMMENT
39c0bb0d64f9 11 seconds ago COPY index.html htdocs/index.html # buildkit    254B
buildkit.dockerfile.v0
<missing>    11 seconds ago WORKDIR /usr/local/apache2                      0B
buildkit.dockerfile.v0
<missing>    11 seconds ago LABEL VERSION=0.1                               0B
buildkit.dockerfile.v0
<missing>    11 seconds ago LABEL AUTHOR=chukiatwr@gmail.com                0B
buildkit.dockerfile.v0
<missing>    24 hours ago   /bin/sh -c #(nop)  CMD ["httpd-foreground"]     0B
<missing>    24 hours ago   /bin/sh -c #(nop)  EXPOSE 80                    0B
<missing>    24 hours ago   /bin/sh -c #(nop) COPY file:c432ff61c4993ecd…   138B
<missing>    24 hours ago   /bin/sh -c #(nop)  STOPSIGNAL SIGWINCH          0B
<missing>    24 hours ago   /bin/sh -c set -eux;   savedAptMark="$(apt-m…   59.9MB
<missing>    24 hours ago   /bin/sh -c #(nop)  ENV HTTPD_PATCHES=           0B
<missing>    24 hours ago   /bin/sh -c #(nop)  ENV HTTPD_SHA256=eb397fee…   0B
<missing>    24 hours ago   /bin/sh -c #(nop)  ENV HTTPD_VERSION=2.4.54     0B
<missing>    24 hours ago   /bin/sh -c set -eux;  apt-get update;  apt-g…   4.76MB
<missing>    24 hours ago   /bin/sh -c #(nop) WORKDIR /usr/local/apache2    0B
<missing>    24 hours ago   /bin/sh -c mkdir -p "$HTTPD_PREFIX"  && chow…   0B
<missing>    24 hours ago   /bin/sh -c #(nop)  ENV PATH=/usr/local/apach…   0B
<missing>    24 hours ago   /bin/sh -c #(nop)  ENV HTTPD_PREFIX=/usr/loc…   0B
<missing>    26 hours ago   /bin/sh -c #(nop)  CMD ["bash"]                  0B
<missing>    26 hours ago   /bin/sh -c #(nop) ADD file:8644a8156a07a656a…   80.5MB

D:\docker\httpd>docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
myhttpd      latest    39c0bb0d64f9   21 seconds ago   145MB

D:\docker\httpd>docker run -d -p8123:80 myhttpd
6de5413e33a22c0419ec1dbb55105e09fda2c96fb3d19a3cd803e0028570ebc6

D:\docker\httpd>docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED       STATUS       PORTS
NAMES
82271a926415   myhttpd   "httpd-foreground" 3 minutes ago Up 3 minutes 0.0.0.0:8123->80/   tcp
brave_edison
```



Part 6 Create a simple web site container from NginX



dockerfile

```
FROM nginx:1.21.1-alpine

#config
copy ./nginx.conf /etc/nginx/nginx.conf

#content, comment out the ones you don't need!
copy ./*.html /usr/share/nginx/html/
#copy ./*.css /usr/share/nginx/html/
#copy ./*.png /usr/share/nginx/html/
#copy ./*.js /usr/share/nginx/html/
```

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
   <title>My web page in docker</title>
</head>
```

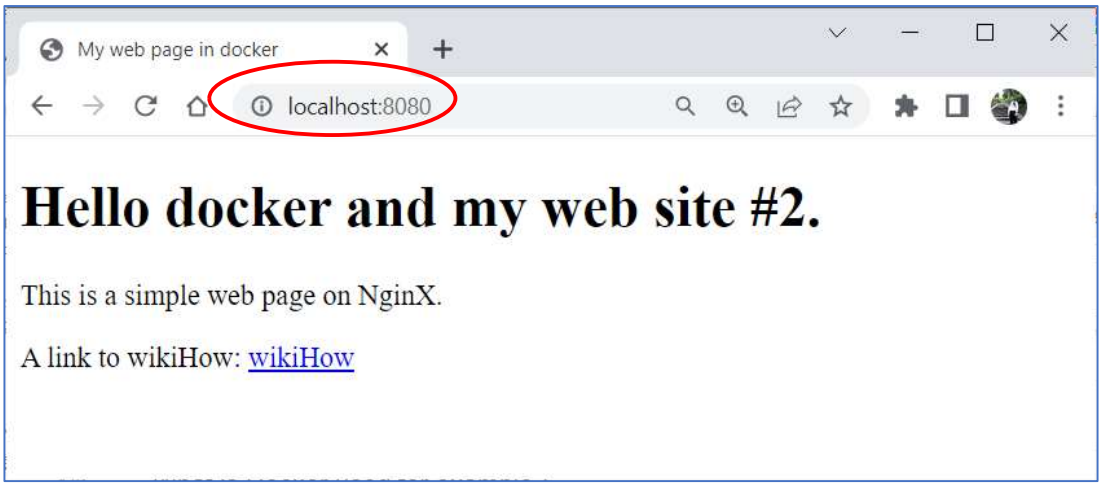
```
<h1>Hello docker and my web site #1.</h1>
<p>This is a simple web page.</p>
<p>A link to wikiHow: <a href="http://www.wikihow.com">wikiHow</a></p>

</html>
```

nginx.conf

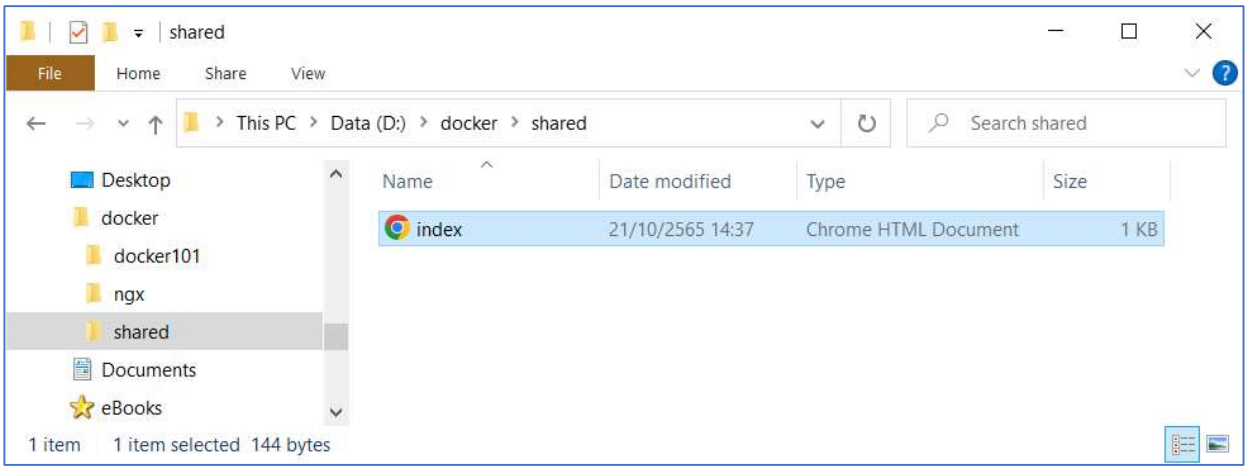
```
user  nginx;
worker_processes  auto;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
    worker_connections  512;
}
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    server {
        listen 80;

         location = /status {
             access_log off;
             default_type text/plain;
             add_header Content-Type text/plain;
             return 200 "alive";
        }

         location / {
            gzip off;
            root /usr/share/nginx/html/;
            index  index.html;
        }

        location ~* \.(js|jpg|png|css)$ {
            root /usr/share/nginx/html/;
        }
    }
    sendfile        on;
    keepalive_timeout  65;
}
```

```
D:\docker\ngx>docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

D:\docker\ngx>docker build -t mywebsite .
[+] Building 5.8s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                    0.0s
 => => transferring dockerfile: 315B                                    0.0s
 => [internal] load .dockerignore                                       0.0s
```

```
 => => transferring context: 2B                                              0.0s
 => [internal] load metadata for docker.io/library/nginx:1.21.1-alpine       3.8s
 => [1/3] FROM docker.io/library/nginx:1.21.1-alpine@sha256:bfe377bdeb...    1.8s
    ...
 => [internal] load build context                                            0.0s
 => => transferring context: 62B                                             0.0s
 => [2/3] COPY ./nginx.conf /etc/nginx/nginx.conf                            0.0s
 => [3/3] COPY ./*.html /usr/share/nginx/html/                               0.0s
 => exporting to image                                                       0.0s
 => => exporting layers                                                      0.0s
 => => writing image sha256:c63a24c2808023a792f635c5558bb828c108725c0283..  0.0s
 => => naming to docker.io/library/mywebsite                                 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn
how to fix them

D:\docker\ngx>docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
mywebsite    latest    c63a24c28080   15 seconds ago   22.8MB

D:\docker\ngx>docker run -d -t -p8080:80 --name nginx101 mywebsite
a8bc5d481bab14877c6e8b626453a5ba3655d0bcc18ed916b9846bd8a015b3f7

D:\docker\ngx>docker ps -a
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS
PORTS                  NAMES
a8bc5d481bab   mywebsite   "/docker-entrypoint.…"   5 seconds ago   Up 3 seconds
0.0.0.0:8080->80/tcp   nginx101

D:\docker\ngx>
```



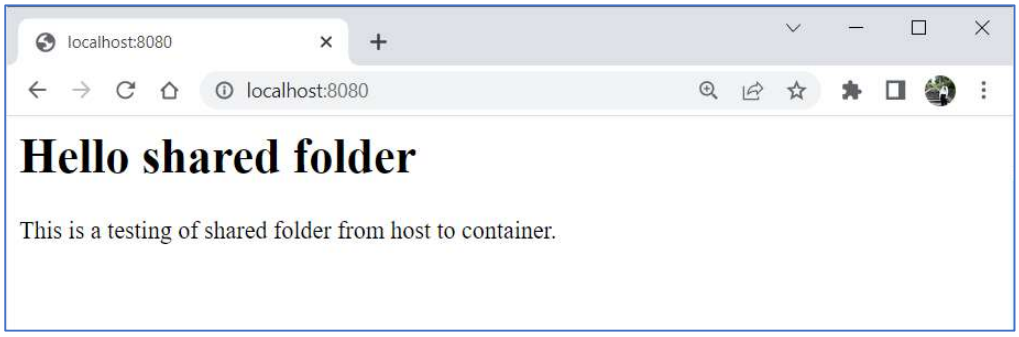
Part 7 Share folder of the host with the container



index.html

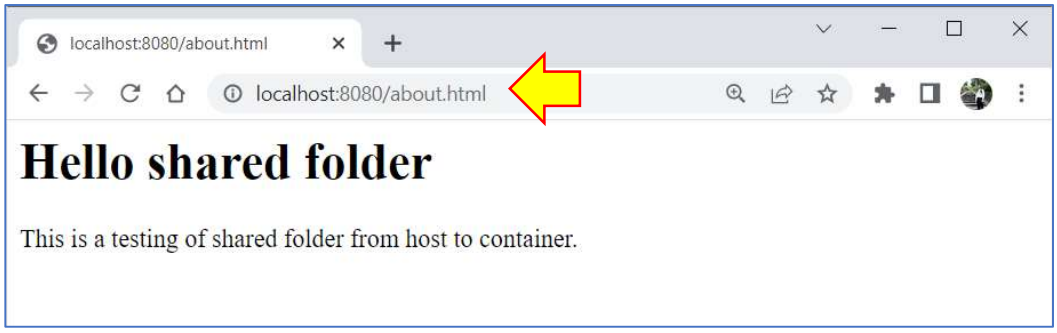
```
<html>
<h1>Hello shared folder.</h1>
<p>This is a testing of shared folder from host to container.</p>
</html>
```

```
/**
 In cmd, use %cd% to get the current working directory inside the docker run command.
 In PowerShell, use ${PWD}.
 On Linux, use $(pwd).
 */
D:\docker>docker run -d -p 8080:80 -v %cd%:/usr/share/nginx/html nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
bd159e379b3b: Pull complete
6659684f075c: Pull complete
679576c0baac: Pull complete
22ca44aeb873: Pull complete
b45acafbea93: Pull complete
bcbbe1cb4836: Pull complete
Digest: sha256:5ffb682b98b0362b66754387e86b0cd31a5cb7123e49e7f6f6617690900d20b2
Status: Downloaded newer image for nginx:latest
8d1783d20396d9d12593003aafe3e780d4b6a018a3984080bf4062590f408201
```




```
D:\docker\shared>ren index.html about.html
```

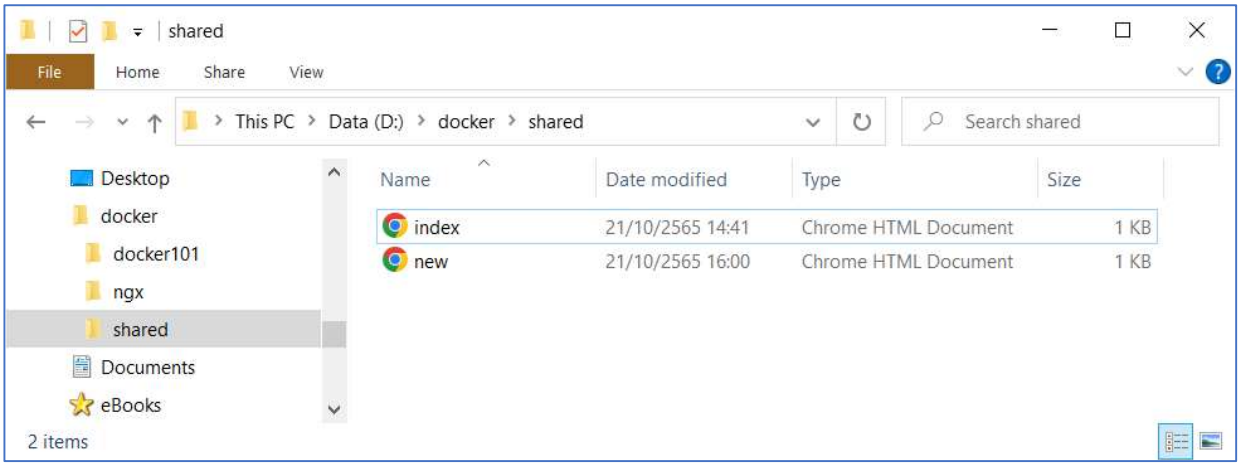
```
D:\docker\shared>dir
 Volume in drive D is Data
 Volume Serial Number is A234-7DEE
 Directory of D:\docker\shared

10/21/2022  02:47 PM    <DIR>          .
10/21/2022  02:47 PM    <DIR>          ..
10/21/2022  02:41 PM               112 about.html
               1 File(s)            112 bytes
               2 Dir(s)  158,114,881,536 bytes free
```




```
D:\docker\shared>ren about.html index.html      # rename it back to index.html
```

Share a folder from the host with the docker Ubuntu

```
D:\docker\shared>docker run -i -t -v %cd%:/home ubuntu

root@aa86fff8b10d:/# cd /home

root@aa86fff8b10d:/home# ls -l
total 0
-rwxrwxrwx 1 root root 112 Oct 21 07:41 index.html

root@aa86fff8b10d:/home# cp index.html new.html
root@aa86fff8b10d:/home# ls -l
total 0
-rwxrwxrwx 1 root root 112 Oct 21 07:41 index.html
-rwxr-xr-x 1 root root 112 Oct 21 09:00 new.html
root@aa86fff8b10d:/home# exit
```



Share a folder with read-only permission :ro

```
D:\docker\shared>docker run -i -t -v %cd%:/home:ro ubuntu

root@f68c83e15938:/# cd /home

root@f68c83e15938:/home# ls -l
total 0
-rwxrwxrwx 1 root root 112 Oct 21 07:41 index.html
-rwxr-xr-x 1 root root 112 Oct 21 09:00 new.html

root@f68c83e15938:/home# cp index.html backup.html
cp: cannot create regular file 'backup.html': Read-only file system

root@f68c83e15938:/home# exit
exit
```

Part 8 Docker Compose

Goal: Run a Python code for counting web page access with Redis, an in-memory data structure.

Key concepts

- *Docker Compose* is used for starting multiple Docker containers on the same host – so you don't have to start each container separately. You may do so by configuring a from within a single *YAML* file.
- *Docker swarm* is a container orchestration tool that allows you to run and connect containers on multiple hosts, for a scalable application.
- *Kubernetes* is a container orchestration tool that is similar to Docker swarm, but has a wider appeal due to its ease of automation and ability to handle higher demand.

References:

1. https://doc4dev.com/en/create-a-web-site-php-apache-mysql-in-5-minutes-with-docker/
2. https://www.techrepublic.com/article/simplifying-the-mystery-when-to-use-docker-docker-compose-and-kubernetes/

Steps:

1. Install Docker Desktop which includes both Docker Engine and Docker Compose.
2. Prepare app.py, requirements.txt, dockerfile, docker-compose.yml files as following.

app.py

```
import time
import redis
from flask import Flask

app = Flask(__name__)  # set name of the application package

cache = redis.Redis(host='redis', port=6379) # create connection to redis

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/') # entry point for home
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)
```

requirements.txt

```
flask
redis
```

dockerfile

```
# syntax=docker/dockerfile:1
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run"]
```

docker-compose.yml

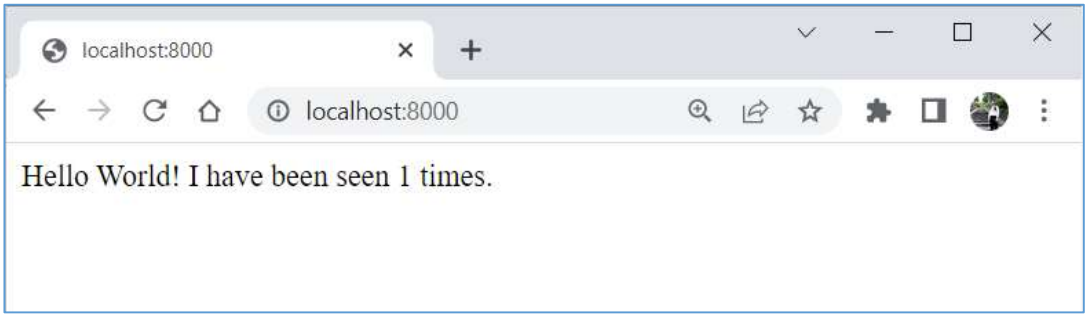
```
version: "3.9"
services:
```

```
  web:
    build: .
    ports:
      - "8000:5000"
    volumes:
      - .:/code
    environment:
      FLASK_DEBUG: True
  redis:
    image: "redis:alpine"
```

3. Run the following script.

```
D:\docker\compose>docker compose up -d
[+] Running 7/7
 - redis Pulled                                                    7.2s
   - ca7dd9ec2225 Pull complete                                    1.3s
   - 83276aa4de36 Pull complete                                    1.4s
   - 731cc432e6da Pull complete                                    1.6s
   - 862de9590cc6 Pull complete                                    2.6s
   - a26b23e71d57 Pull complete                                    2.7s
   - 4b937ee5a2e0 Pull complete                                    2.7s
[+] Building 2.3s (15/15) FINISHED
 => [internal] load build definition from dockerfile               0.0s
...
=> => writing image
sha256:edc77f380ee0ea078d7f21d00b5d1ffdba90c4d950aa06124a5a0d66a111e97  0.0s
 => => naming to docker.io/library/compose-web                     0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn
how to fix them
[+] Running 3/3
 - Network compose_default    Created                              0.7s
 - Container compose-redis-1  Started                              1.4s
 - Container compose-web-1    Started                              1.7s
```

4. Test running the app.py and Redis for a few times.




```
compose-web-1    |  * Debug mode: on
compose-web-1    | WARNING: This is a development server. Do not use it in a production deployment.
ead.
compose-web-1    |  * Running on all addresses (0.0.0.0)
compose-web-1    |  * Running on http://127.0.0.1:5000
compose-web-1    |  * Running on http://172.19.0.2:5000
compose-web-1    | Press CTRL+C to quit
compose-web-1    |  * Restarting with stat
compose-web-1    |  * Debugger is active!
compose-web-1    |  * Debugger PIN: 120-804-403
compose-web-1    | 172.19.0.1 - - [30/Nov/2022 07:12:29] "GET / HTTP/1.1" 200 -
compose-web-1    | 172.19.0.1 - - [30/Nov/2022 07:13:47] "GET / HTTP/1.1" 200 -
compose-web-1    | 172.19.0.1 - - [30/Nov/2022 07:13:48] "GET / HTTP/1.1" 200 -
compose-web-1    | 172.19.0.1 - - [30/Nov/2022 07:13:49] "GET / HTTP/1.1" 200 -
```

```
D:\docker\compose>docker compose stop
[+] Running 2/2se>
 - Container compose-redis-1  Stopped                                    0.3s
 - Container compose-web-1    Stopped                                    0.6s

[+] Running 2/0se>docker compose stop
 - Container compose-redis-1  Stopped                                    0.0s
 - Container compose-web-1    Stopped                                    0.0s

D:\docker\compose>docker compose down --volume
time="2022-11-30T21:25:58+07:00" level=warning msg="--volume is deprecated, please use
--volumes"
time="2022-11-30T21:25:58+07:00" level=warning msg="--volume is deprecated, please use
--volumes"
time="2022-11-30T21:25:58+07:00" level=warning msg="--volume is deprecated, please use
--volumes"
[+] Running 3/3
 - Container compose-redis-1  Removed                                    0.0s
 - Container compose-web-1    Removed                                    0.0s
 - Network compose_default    Removed                                    0.6s

D:\docker\compose>
```

Part 9 Use MySQL on Docker

Goal: Run mysql inside a docker container.

Ref: https://ostechnix.com/setup-mysql-with-docker-in-linux/

load_data.sql

```
CREATE DATABASE IF NOT EXISTS football;

USE football;

CREATE TABLE IF NOT EXISTS players (
    player_name     VARCHAR(16)     NOT NULL,
    player_age      INT             NOT NULL,
    player_club     VARCHAR(16)     NOT NULL,
    player_country  VARCHAR(16)     NOT NULL
);

INSERT INTO players VALUES ("Messi",34,"PSG","Argentina");
INSERT INTO players VALUES ("Ronaldo",36,"MANU","Portugal");
INSERT INTO players VALUES ("Neymar",29,"PSG","Brazil");
INSERT INTO players VALUES ("Kane",28,"SPURS","England");
INSERT INTO players VALUES ("E Hazard",30,"MADRID","Belgium");
```

```
D:\docker\mysql>docker pull mysql:latest
latest: Pulling from library/mysql
0ed027b72ddc: Pull complete
0296159747f1: Pull complete
3d2f9b664bd3: Pull complete
df6519f81c26: Pull complete
36bb5e56d458: Pull complete
054e8fde88d0: Pull complete
f2b494c50c7f: Pull complete
132bc0d471b8: Pull complete
135ec7033a05: Pull complete
5961f0272472: Pull complete
75b5f7a3d3a4: Pull complete
Digest: sha256:3d7ae561cf6095f6aca8eb7830e1d14734227b1fb4748092f2be2cfbccf7d614
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

D:\docker\mysql>docker images mysql
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
mysql        latest    7484689f290f   4 weeks ago   538MB

D:\docker\mysql>docker run --name mysql -p 3306:3306 -v mysqlvol:/var/lib/mysql/ -d -e
"MYSQL_ROOT_PASSWORD=temp123" mysql
932553197dea0ccc9ba72bd2297e2ca4583b3482a84388afdb84c9b73ce0cc4c
```

```
D:\docker\mysql>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
PORTS                               NAMES
932553197dea   mysql     "docker-entrypoint.s…"   28 seconds ago   Up 27 seconds
0.0.0.0:3306->3306/tcp, 33060/tcp   mysql

D:\docker\mysql>docker cp load_data.sql mysql:/tmp

D:\docker\mysql>docker exec -it mysql bash
bash-4.4# cd /tmp
bash-4.4# ls -l
total 4
-rwxr-xr-x 1 root root 594 Jan  4 08:21 load_data.sql

bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source /tmp/load_data.sql
Query OK, 1 row affected (0.01 sec)

Database changed
Query OK, 0 rows affected (0.02 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| football           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> select * from players;
+-------------+------------+-------------+----------------+
| player_name | player_age | player_club | player_country |
+-------------+------------+-------------+----------------+
```

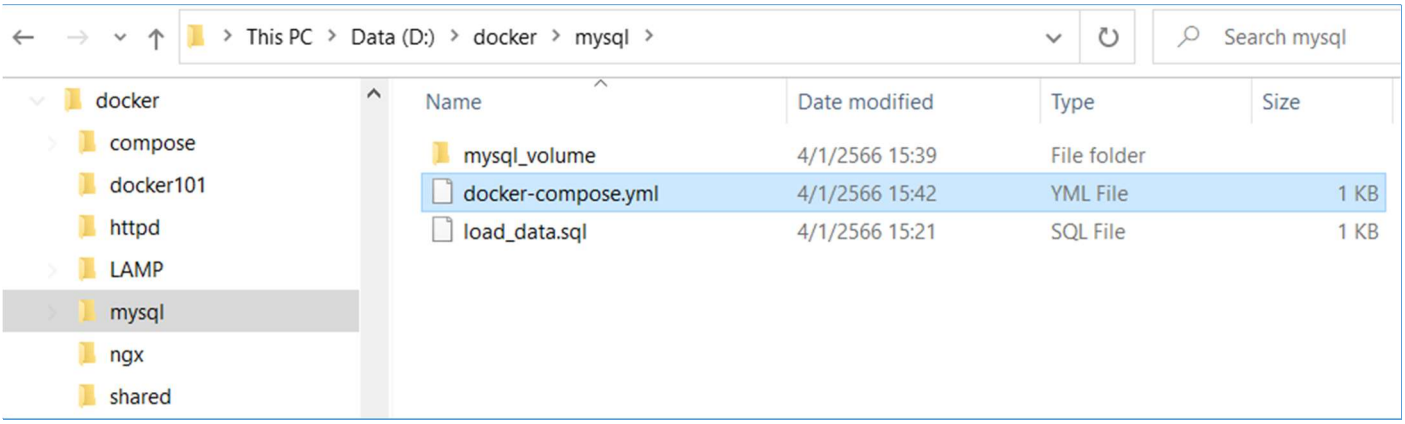
```
| Messi       |         34 | PSG         | Argentina      |
| Ronaldo     |         36 | MANU        | Portugal       |
| Neymar      |         29 | PSG         | Brazil         |
| Kane        |         28 | SPURS       | England        |
| E Hazard    |         30 | MADRID      | Belgium        |
+-------------+------------+-------------+----------------+
5 rows in set (0.01 sec)
```

Part 10 Using MySQL on Docker with Docker Compose

Goal: Use Docker Compose to run mysql

Ref: https://ostechnix.com/setup-mysql-with-docker-in-linux/

Notice: Remove mysql images from previous Part, otherwise it'll have a conflict.



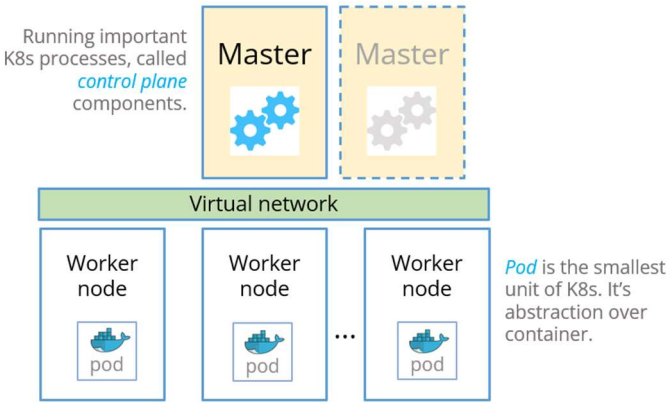

```yaml
docker-compose.yml

version: '3.8'
services:
  database:
    image: mysql:latest
    container_name: mysql
    environment:
      MYSQL_ROOT_PASSWORD: temp1234
    ports:
      - "3306:3306"
    volumes:
      - ./mysql_volume:/var/lib/mysql
volumes:
  mysql_compose_volume:
```

```
D:\docker\mysql>docker compose up -d
time="2023-01-04T15:38:53+07:00" level=warning msg="Found multiple config files with
supported names: D:\\docker\\mysql\\docker-compose.yml, D:\\docker\\mysql\\docker-
compose.yaml"
time="2023-01-04T15:38:53+07:00" level=warning msg="Using D:\\docker\\mysql\\docker-
compose.yml"
time="2023-01-04T15:38:53+07:00" level=warning msg="Found multiple config files with
supported names: D:\\docker\\mysql\\docker-compose.yml, D:\\docker\\mysql\\docker-
compose.yaml"
time="2023-01-04T15:38:53+07:00" level=warning msg="Using D:\\docker\\mysql\\docker-
compose.yml"
[+] Running 1/0
 - Container mysql  Created
0.1s
```

```
Attaching to mysql
mysql  | 2023-01-04 08:38:54+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL
Server 8.0.31-1.el8 started.
mysql  | 2023-01-04 08:38:54+00:00 [Note] [Entrypoint]: Switching to dedicated user
'mysql'
...
mysql  | 2023-01-04T08:39:09.676220Z 0 [System] [MY-011323] [Server] X Plugin ready for
connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.sock
mysql  | 2023-01-04T08:39:09.676350Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld:
ready for connections. Version: '8.0.31'  socket: '/var/run/mysqld/mysqld.sock'  port:
3306  MySQL Community Server - GPL.

[+] Running 1/1
 - Container mysql  Started
1.1s

D:\docker\mysql>docker cp load_data.sql mysql:/tmp

D:\docker\mysql>docker exec -it mysql bash
bash-4.4# cd tmp
bash-4.4# ls -la
total 12
drwxrwxrwt 1 root root 4096 Jan  4 08:50 .
drwxr-xr-x 1 root root 4096 Jan  4 08:50 ..
-rwxr-xr-x 1 root root  594 Jan  4 08:21 load_data.sql

bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> source /tmp/load_data.sql
Query OK, 1 row affected (0.00 sec)

Database changed
```

```
Query OK, 0 rows affected (0.02 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.01 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| football           |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use football;
Database changed
mysql> select * from players;
+-------------+------------+-------------+----------------+
| player_name | player_age | player_club | player_country |
+-------------+------------+-------------+----------------+
| Messi       |         34 | PSG         | Argentina      |
| Ronaldo     |         36 | MANU        | Portugal       |
| Neymar      |         29 | PSG         | Brazil         |
| Kane        |         28 | SPURS       | England        |
| E Hazard    |         30 | MADRID      | Belgium        |
+-------------+------------+-------------+----------------+
5 rows in set (0.00 sec)

mysql>
```

Part 11 Basic usage of Kubernetes



kubeadm is used to set up and manage the Kubernetes cluster.

- Set up the control plane components (API server, etcd, scheduler, and controller-manager) on the master nodes and join the worker nodes to the cluster.
- Integrate with cloud providers and network plugins
- Support for multi-master and high-availability setups

kubelet is the primary node agent that runs on every worker node in a Kubernetes cluster.

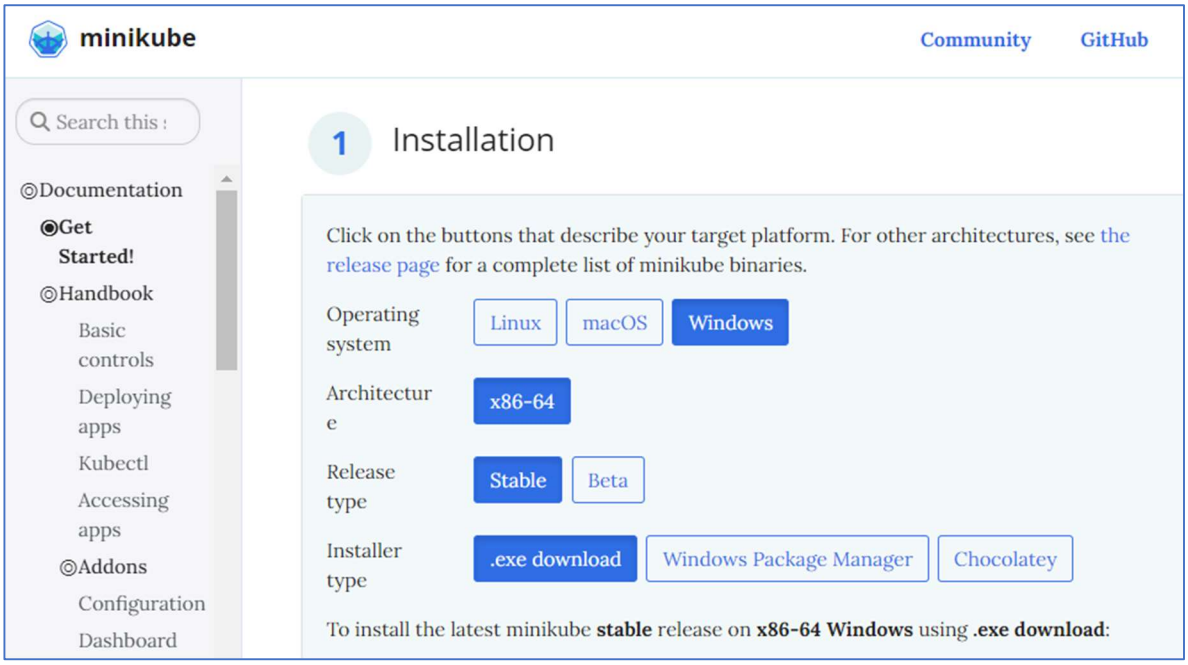
- communicate with the control plane's API server to receive instructions on which containers to run and how to run them.
- manage the lifecycle of pods on the node; ensuring that the containers specified in the pod are running and healthy.
- manage the volumes that are attached to containers on the node; mounting the volumes and ensuring that they are available to the containers as expected.

kubectl is a command-line tool for interacting to deploy, manage, and troubleshoot applications on the cluster.

- kubectl is necessary to interact with the cluster
- To deploy, inspect, update, and manage services running on a K8s cluster.
- Kubectl inherently comes with docker, no need to install.

minikube is a tool that helps you run *a single-node Kubernetes cluster locally* on your machine.

- Minikube is a way to set up a cluster on your local machine to test and experiment with.
- Primarily used for development and testing purposes.
- We may need to install minikube from https://minikube.sigs.k8s.io/docs/start/




```
D:\docker>kubectl version --short
Flag --short has been deprecated, and will be removed in the future. The --short output
will become the default.
Client Version: v1.25.0
Kustomize Version: v4.5.7
Unable to connect to the server: dial tcp 127.0.0.1:63718: connectex: No connection
could be made because the target machine actively refused it.

D:\docker>kubectl version
WARNING: This version information is deprecated and will be replaced with the output
from kubectl version --short.  Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.0",
GitCommit:"a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2", GitTreeState:"clean",
BuildDate:"2022-08-23T17:44:59Z", GoVersion:"go1.19", Compiler:"gc",
Platform:"windows/amd64"}
Kustomize Version: v4.5.7
Unable to connect to the server: dial tcp 127.0.0.1:63718: connectex: No connection
could be made because the target machine actively refused it.

D:\docker>minikube version
minikube version: v1.28.0
commit: 986b1ebd987211ed16f8cc10aed7d2c42fc8392f

D:\docker>minikube start --nodes=2
* minikube v1.28.0 on Microsoft Windows 10 Home Single Language 10.0.19044 Build 19044
* Using the docker driver based on existing profile
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Restarting existing docker container for "minikube" ...
* Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
* Configuring CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
! The cluster minikube already exists which means the --nodes parameter will be
ignored. Use "minikube node add" to add nodes to an existing cluster.
```

```
* Starting worker node minikube-m02 in cluster minikube
* Pulling base image ...
* Restarting existing docker container for "minikube-m02" ...
* Found network options:
  - NO_PROXY=192.168.49.2
  - no_proxy=192.168.49.2
* Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
  - env NO_PROXY=192.168.49.2
* Verifying Kubernetes components...
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by
default
```

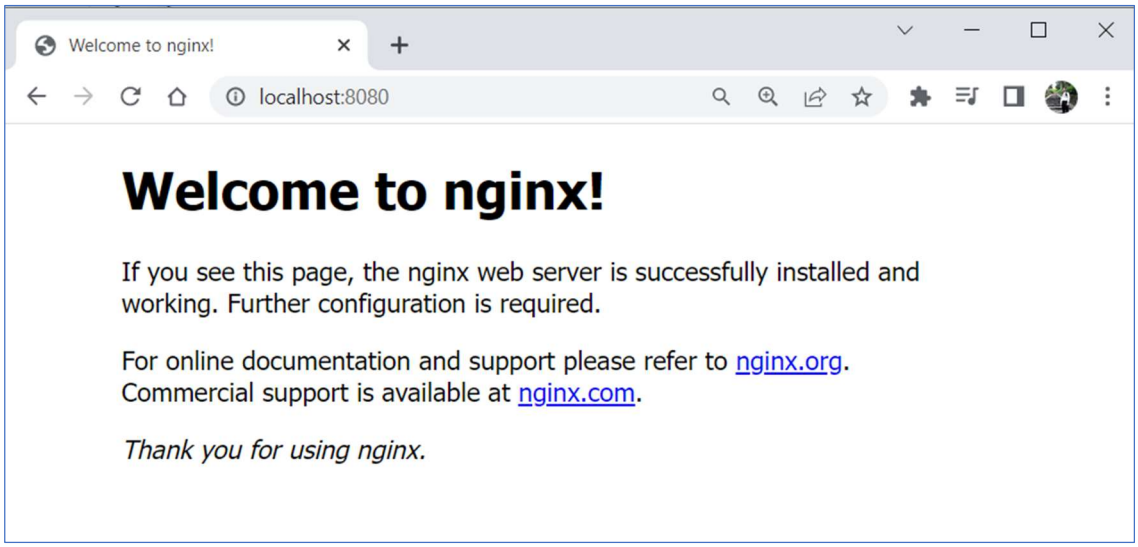
```
D:\docker>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

minikube-m02
type: Worker
host: Running
kubelet: Running
```

```
D:\docker>docker ps
CONTAINER ID   IMAGE                                  COMMAND                  CREATED
STATUS          PORTS
NAMES
cb02ae271b01   gcr.io/k8s-minikube/kicbase:v0.0.36   "/usr/local/bin/entr…"   13 hours
ago   Up 38 minutes   127.0.0.1:62772->22/tcp, 127.0.0.1:62773->2376/tcp,
127.0.0.1:62775->5000/tcp, 127.0.0.1:62776->8443/tcp, 127.0.0.1:62774->32443/tcp
minikube-m02
ba0e7e394a03   gcr.io/k8s-minikube/kicbase:v0.0.36   "/usr/local/bin/entr…"   13 hours
ago   Up 38 minutes   127.0.0.1:62635->22/tcp, 127.0.0.1:62636->2376/tcp,
127.0.0.1:62633->5000/tcp, 127.0.0.1:62634->8443/tcp, 127.0.0.1:62637->32443/tcp
minikube
```

```
D:\docker>kubectl get nodes
NAME           STATUS   ROLES           AGE   VERSION
minikube       Ready    control-plane   13h   v1.25.3
minikube-m02   Ready    <none>          39m   v1.25.3
```

```
D:\docker>kubectl get pods -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
kube-system   coredns-565d847f94-4l7nr           1/1     Running   1 (40m ago)   13h
kube-system   etcd-minikube                      1/1     Running   1 (40m ago)   13h
kube-system   kindnet-pq5wz                      1/1     Running   1 (40m ago)   13h
kube-system   kindnet-zcpg4                      1/1     Running   1 (39m ago)   13h
kube-system   kube-apiserver-minikube            1/1     Running   1 (40m ago)   13h
kube-system   kube-controller-manager-minikube   1/1     Running   1 (40m ago)   13h
kube-system   kube-proxy-7jjpf                   1/1     Running   1 (40m ago)   13h
kube-system   kube-proxy-cc9bw                   1/1     Running   1 (39m ago)   13h
kube-system   kube-scheduler-minikube            1/1     Running   1 (40m ago)   13h
```

```
kube-system   storage-provisioner                1/1     Running   2 (39m ago)   13h

D:\docker>kubectl delete deployment mynginx
deployment.apps "mynginx" deleted

D:\docker>kubectl create deployment mynginx --image nginx:latest --replicas 3
deployment.apps/mynginx created

D:\docker>kubectl get pods
NAME                       READY   STATUS              RESTARTS        AGE
mynginx-55b5b97fd6-rnd89   0/1     ContainerCreating   0               13s
mynginx-55b5b97fd6-sqgsh   0/1     ContainerCreating   0               13s
mynginx-55b5b97fd6-wsc5h   1/1     Running             0               13s

D:\docker>kubectl get deployment
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
mynginx   2/3     3            2           34s


D:\docker>kubectl scale deployment mynginx --replicas 5
deployment.apps/mynginx scaled

D:\docker>kubectl get pods
NAME                       READY   STATUS              RESTARTS   AGE
mynginx-55b5b97fd6-lr5v9   0/1     ContainerCreating   0          3s
mynginx-55b5b97fd6-rnd89   1/1     Running             0          51s
mynginx-55b5b97fd6-rrdqp   0/1     ContainerCreating   0          3s
mynginx-55b5b97fd6-sqgsh   0/1     ContainerCreating   0          51s
mynginx-55b5b97fd6-wsc5h   1/1     Running             0          51s

D:\docker>kubectl get pods
NAME                       READY   STATUS              RESTARTS   AGE
mynginx-55b5b97fd6-lr5v9   1/1     Running             0          23s
mynginx-55b5b97fd6-rnd89   1/1     Running             0          71s
mynginx-55b5b97fd6-rrdqp   1/1     Running             0          23s
mynginx-55b5b97fd6-sqgsh   1/1     Running             0          71s
mynginx-55b5b97fd6-wsc5h   1/1     Running             0          71s

D:\docker>kubectl expose deployment/mynginx --port 80
service/mynginx exposed

D:\docker>kubectl get services
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1        <none>        443/TCP   24m
mynginx      ClusterIP   10.100.137.111   <none>        80/TCP    21s

D:\docker>kubectl port-forward service/mynginx 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
Handling connection for 8080
Handling connection for 8080
```




```
D:\docker>kubectl get deployments,pods,services
NAME                        READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/mynginx     5/5     5            5           16m
NAME                                READY   STATUS    RESTARTS   AGE
pod/mynginx-55b5b97fd6-lr5v9        1/1     Running   0          15m
pod/mynginx-55b5b97fd6-rnd89        1/1     Running   0          16m
pod/mynginx-55b5b97fd6-rrdqp        1/1     Running   0          15m
pod/mynginx-55b5b97fd6-sqgsh        1/1     Running   0          16m
pod/mynginx-55b5b97fd6-wsc5h        1/1     Running   0          16m
NAME                    TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
service/kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP   38m
service/mynginx         ClusterIP   10.100.137.111   <none>        80/TCP    14m

D:\docker>kubectl delete service mynginx
service "mynginx" deleted

D:\docker>kubectl delete deployment nginx
deployment.apps "nginx" deleted
```

Part 12 Kubernetes's YAML file

The YAML file is the main configuration file for Kubernetes. YAML (which stands for YAML Ain't Markup Language) is a language used to provide configuration for software, and is the main type of input for Kubernetes configurations. It is human-readable and can be authored in any text editor.

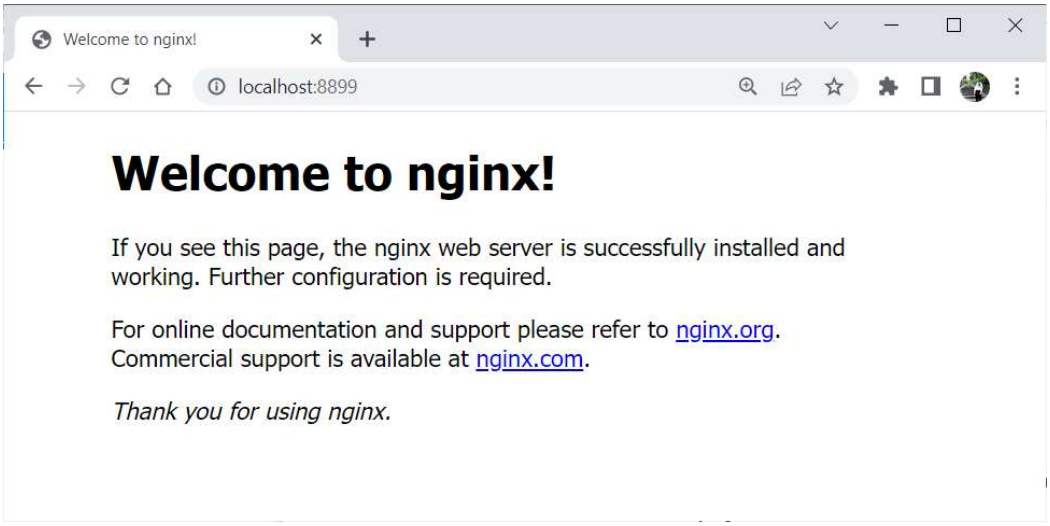
A Kubernetes user or administrator specifies data in a YAML file, typically to define a Kubernetes object. The YAML configuration is called a "manifest", and when it is "applied" to a Kubernetes cluster, Kubernetes creates an object based on the configuration.

nginx-app.yaml

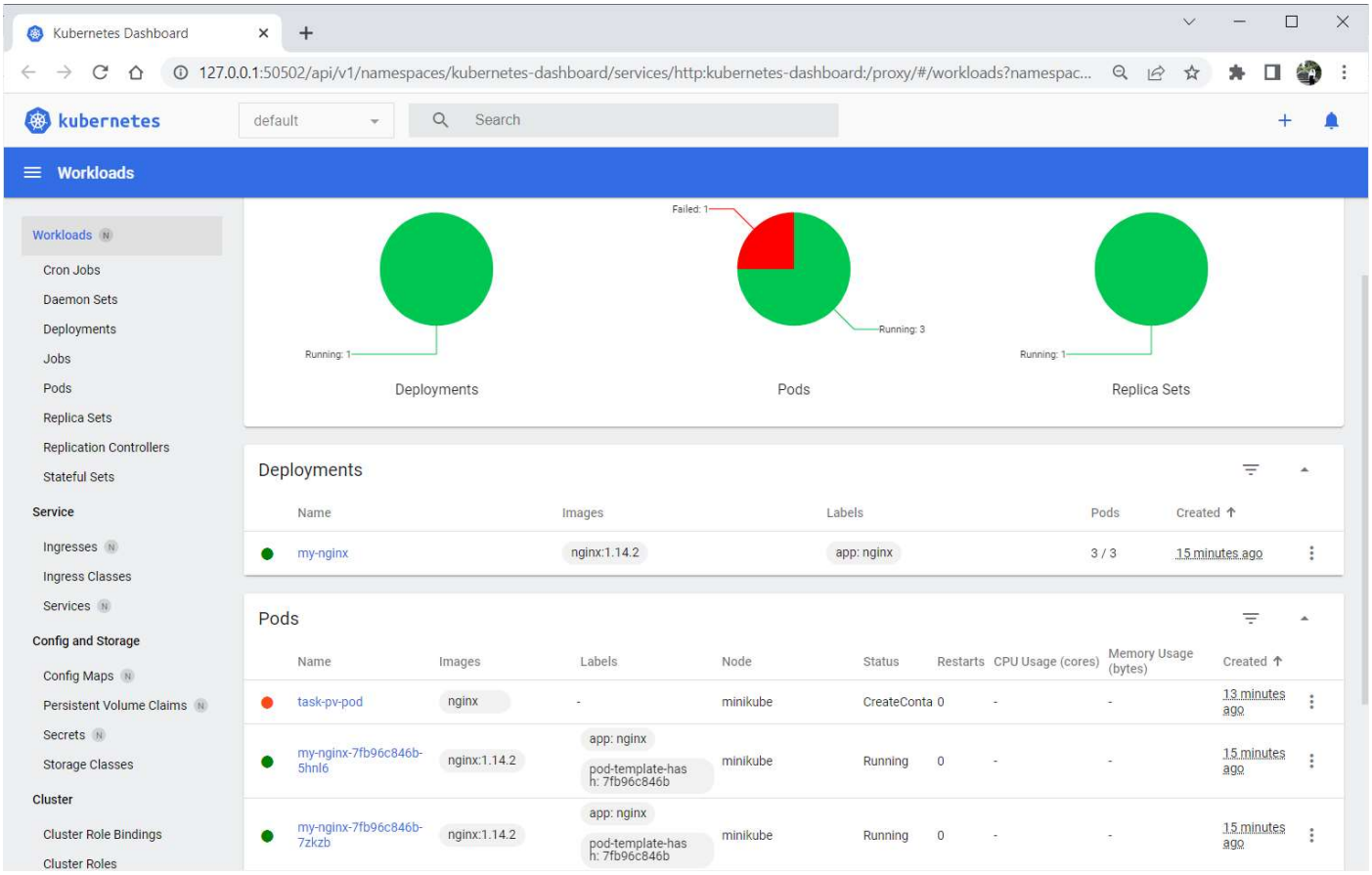
```yaml
apiVersion: v1
kind: Service

metadata:
  name: my-nginx-svc
  labels:
    app: nginx

spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: nginx

---

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx
  labels:
    app: nginx

spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

```
D:\docker\kubernetes>kubectl apply -f nginx-app.yaml
service/my-nginx-svc created
deployment.apps/my-nginx created

D:\docker\kubernetes>kubectl get deployments,pods,services
NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/my-nginx      3/3     3            3           60s
NAME                              READY   STATUS    RESTARTS   AGE
pod/my-nginx-7fb96c846b-cb5c5     1/1     Running   0          60s
pod/my-nginx-7fb96c846b-g5wc4     1/1     Running   0          60s
pod/my-nginx-7fb96c846b-rvlzh     1/1     Running   0          60s
NAME                    TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
service/kubernetes      ClusterIP      10.96.0.1        <none>        443/TCP
2m12s
service/my-nginx-svc    LoadBalancer   10.111.234.181   <pending>     80:31771/TCP   60s

D:\docker\kubernetes>kubectl get pods -A
NAMESPACE     NAME                                       READY   STATUS    RESTARTS      AGE
default       my-nginx-7fb96c846b-cb5c5                  1/1     Running   0             3m21s
default       my-nginx-7fb96c846b-g5wc4                  1/1     Running   0             3m21s
default       my-nginx-7fb96c846b-rvlzh                  1/1     Running   0             3m21s
kube-system   coredns-565d847f94-bg6fk                   1/1     Running   0             4m18s
kube-system   etcd-minikube                              1/1     Running   0             4m30s
kube-system   kube-apiserver-minikube                    1/1     Running   0             4m30s
kube-system   kube-controller-manager-minikube           1/1     Running   0             4m30s
kube-system   kube-proxy-566bh                           1/1     Running   0             4m18s
kube-system   kube-scheduler-minikube                    1/1     Running   0             4m30s
kube-system   storage-provisioner                        1/1     Running   1 (3m57s ago) 4m28s
D:\docker\kubernetes>kubectl autoscale deployment/my-nginx --min=1 --max=4
horizontalpodautoscaler.autoscaling/my-nginx autoscaled

D:\docker\kubernetes>kubectl port-forward deployment/my-nginx 8899:80
Forwarding from 127.0.0.1:8899 -> 80
```

```
Forwarding from [::1]:8899 -> 80
Handling connection for 8899
Handling connection for 8899
```




```
D:\docker\kubernetes>minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:50502/api/v1/namespaces/kubernetes-
dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```



More to Learn

- Kubernetes Fundamentals https://kubebyexample.com/learning-paths/kubernetes-fundamentals/what-kubernetes-3-minutes

- A simple example for deployment https://k8s-examples.container-solutions.com/examples/Deployment/Deployment.html
- Linux essentials https://kubebyexample.com/learning-paths/linux-essentials/what-are-linux-open-source-software-and-distribution
- Youtube: Kubernetes Tutorial https://www.youtube.com/watch?v=yznvWW_L7AA