# 3、Docker images deeply understand and publish images



## 3.1、image understanding

1、An image is a lightweight, executable stand-alone software package that contains everything needed to run a piece of software. We package applications and configurations into a ready, deliverable, deployable runtime environment, including code, libraries, environment variables and configuration files required for runtime, and this large package runtime environment is the image image file.

2、Docker container instances can only be generated through image files.

## 3.2、UnionFS（Federated file systems）

1.Union file system (UnionFS) is a hierarchical, lightweight, high-performance file system, it is the basis of docker images, and supports the modification of the file system as a commit to overlay layer by layer, while different directories can be mounted under the same virtual file system.

2.The image can be inherited through layering, and based on the basic image, various specific application images can be made.

Features of the Union file system: load multiple file systems at the same time, but from the outside, only one file system can be seen; Federated loading overlays the layers of file systems so that the final file system contains files and directories for all layers.

## 3.3、image layering

When downloading an image, pay attention to the downloaded log output, you can see that it is downloading layer by layer:

```
jetson@ubuntu:~$ docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
6425367b44c9: Pull complete
7cef374d113a: Pull complete
1751ddbc0d77: Pull complete
f41e9e3c6d9a: Pull complete
c26e9c11cd2d: Pull complete
949ad8819238: Pull complete
3028a5ad3fd0: Pull complete
a41584bf2c82: Pull complete
f413abbd4b9d: Pull complete
da7c55c30cf5: Pull complete
038fc84e09b5: Pull complete
Digest: sha256:a43f6e7e7f3a5e5b90f857fbed4e3103ece771b19f0f75880f767cf66bbb6577
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
jetson@ubuntu:~$
```

To view the image layering, you can use the command: `docker image inspect image name`, taking the supporting virtual machine as an example, enter in the terminal

```
docker image inspect microros/micro-ros-agent:humble
```

```
yahboom@yahboom-VM:~$ docker image inspect microros/micro-ros-agent:humble
[
    {
        "Id": "sha256:b88b7a8b049eac29cbe1439a485a5e28284bd33de91a987a4f2c1e7a36a6861c",
        "RepoTags": [
            "microros/micro-ros-agent:humble"
        ],
        "RepoDigests": [
            "microros/micro-ros-agent@sha256:21448c90ecd0c02ce392e9231b11c3a91f836c9e0a4e64ed1fc48798da53161b"
        ],
        "Parent": "",
        "Comment": "buildkit.dockerfile.v0",
        "Created": "2023-08-02T07:52:52.538949774Z",
        "Container": "",
        "ContainerConfig": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": null,
            "Cmd": null,
            "Image": "",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
            "OnBuild": null,
            "Labels": null
        },
        "DockerVersion": "",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
```

### 3.3.1、hierarchical understanding

- All docker images start from a base image layer, and when modifications or additions are made, a new image layer will be created on top of the current image layer.
- For a simple example, if a new image is created based on Ubuntu 20.04, this is the first layer of the new image; If you add a Python package to the image, a second image layer is created on top of the base image layer; If you continue to add a security patch, a third mirror layer is created.

- Docker images are all read-only, and when the container starts, a new writable layer is loaded on top of the image! This layer is what we usually call the container layer, and what is under the container is called the image layer!

### 3.3.2、Docker images should use layering benefits

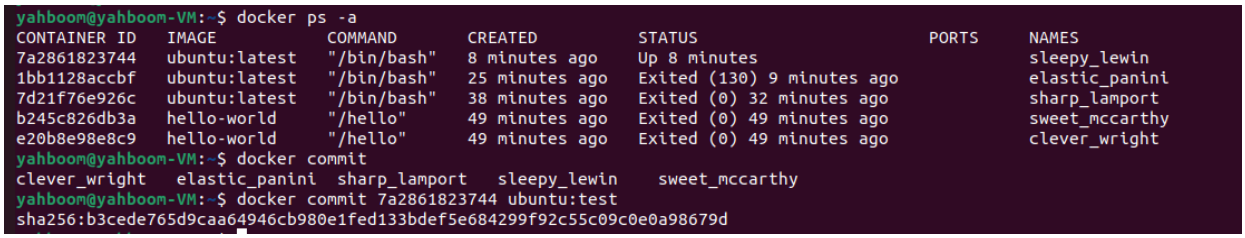
Resource sharing, for example, if there are multiple images built from the same base image, then the host only needs to keep a base image on disk, and only one base image needs to be loaded in memory, so that all containers can be served, and each layer of the image can be shared.

## 3.4、Make and publish images

### 3.4.1、Make an image

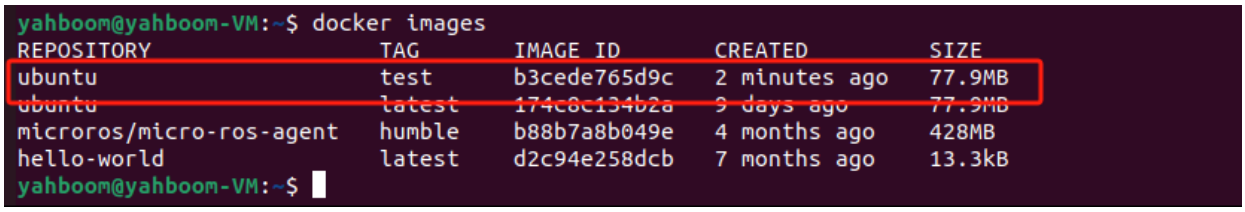
To submit an image from a container, the command used is `docker commit -m="submission description information" -a="author" container id target image name to be created: [label name] [-m -a can also be omitted Parameters]`, taking the supporting virtual machine as an example, input in the terminal.

```
docker commit 7a2861823744 ubuntu:test
```




You can use docker images to check whether it is generated. Enter the following command in the terminal.

```
docker images
```




### 3.4.2、Release image

The docker repository is a centralized place for storing image files. The largest public repository is docker hub (https://hub.docker.com/), which stores a large number of images for users to download. Domestic public warehouses include Alibaba Cloud, NetEase Cloud, etc.

Steps to publish the image to docker hub:

1、Address: https://hub.docker.com/, register an account first

2、Ensure that the account can be logged in normally

3、Use the tag command to modify the image name

The specifications for publishing images to docker hub are as follows:

```
docker push Register username/image name
```

4、Log in to docker hub to publish the image.

```
docker login -u Register username
```

You will then be prompted to enter a password. The password is the account password registered with docker hub. After the password is correct, enter the following command to publish the image.

```
docker push Register username/image name
```

You can visit docker hub to see that it has been published successfully.