

# 3. docker image in-depth understanding and publishing images

---

## 3. docker image in-depth understanding and publishing images

- 3.1 Understanding of mirroring
- 3.2 UnionFS (Union File System)
- 3.3. Mirror layering
  - 3.3.1 Understanding Layers
  - 3.3.2, docker image to use layered benefits
- 3.4 Creating and distributing mirrors
  - 3.4.1 Making mirrors
  - 3.4.2 Publishing Images

The operating environment and hardware and software reference configuration are as follows:

- Reference model: ROSMASTER X3
- Robot hardware configuration: Arm series main control, Silan A1 LiDAR, AstraPro Plus depth camera.
- Robot system: Ubuntu (version not required) + docker (version 20.10.21 and above)
- PC virtual machine: Ubuntu (20.04) + ROS2 (Foxy)
- Usage scenario: use on a relatively clean 2D plane

## 3.1 Understanding of mirroring

---

1, the image is a lightweight, executable independent software package, which contains all the content needed to run a particular software. We will be applications, configurations packaged into a molded, deliverable, deployable runtime environment, including code, libraries needed to run, environment variables and configuration files, etc., this large package of good runtime environment is image image file.

2, only through the image file can generate docker container instances.

## 3.2 UnionFS (Union File System)

---

1, Union file system (UnionFS) is a layered, lightweight, high-performance file system, which is the basis of docker image, and support for modification of the file system as a commit to layer by layer, at the same time, you can hang different directories under the same virtual file system.

2, the image can be inherited through layering, based on the base image, you can create a variety of specific application images.

Union file system features: one-time simultaneous loading of multiple file systems , but from the outside , only one file system can be seen ; Union loading will stack the layers of the file system , so that the final file system will contain all the stratum of files and directories .

### 3.3. Mirror layering

When downloading an image, pay attention to the log output of the download, you can see that it is downloading one layer at a time:

```
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:a43f6e7e7f3a5e5b90f857fbcd4e3103ece771b19f0f75880f767cf66bbb6577
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
jetson@ubuntu:~$
```

```
# 查看镜像分层的方式可以通过命令: docker image inspect 镜像名称
# view image layering can be done with the command: docker image inspect Image
name
jetson@ubuntu:~$ docker image inspect mysql:latest
[
  {
    "Id":
    "sha256:5371f8c3b63eec64a33b35530be5212d6148e0940111b57b689b5ba1ffe808c8",
    .....
    "RootFS": {
      "Type": "layers",
      "Layers": [

        "sha256:d6d4fc6aef875958d6186f85f03d88e6bb6484ab2dd56b30a79163baceff2f6d",
        "sha256:05c3b0b311a02bc56ca23105a76d16bc9b8c1d3e6eac808f4efb1a2e8350224b",
        "sha256:7b80f7f05642477ebc7d93de9539af27caab7c41a768db250fe3fe2b5506ca2c",
        "sha256:50e037faefab22cb1c75e60abb388b823e96a845650f3abd6d0a27e07a5a1d5e",
        "sha256:66040abb3f7201d2cc64531349a8225412db1029447a9431d59d999c941d56f6",
        "sha256:857162425652837a362aa5f1c3d4974cc83702728793de52ba48176d5367a89b",
        "sha256:7eebed3016f6b6ab68aa8e6be35f0689a3c18d331b7b542984a0050b859eaf26",
        "sha256:2fc4c142633d57d795edc0f3fd457f99a35fa611eab8b8c5d75c66e6eb729bc2",
        "sha256:7fde2d12d484f0c14dabd9ca845da0bcdaf60bd773a58ca2d73687473950e7fe",
        "sha256:9319848a00d38e15b754fa9dcd3b6e77ac8506850d32d8af493283131b9745a3",
        "sha256:5ff94d41f068ea5b52244393771471edb6a9a10f7a4ebafda9ef6629874a899b"

      ]
    },
  ]
}
```

```

    "Metadata": {
      "LastTagTime": "0001-01-01T00:00:00Z"
    }
  }
]

```

### 3.3.1 Understanding Layers

- All docker images start with a base image layer. When you make changes or add new content, you create a new image layer on top of the current one.
- For example, if you create a new image based on ubuntu 20.04, this is the first layer of the new image; if you add a python package to the image, this creates a second layer on top of the base layer; and if you continue to add a security patch, this creates a third layer.
- The 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 refer to as the container layer; anything below the container is called the image layer!

### 3.3.2, docker image to use layered benefits

Resource sharing, for example, there are multiple mirrors are built from the same Base image, then the host only need to keep a copy of the Base image on disk, at the same time only need to load a copy of the Base image in memory, so that it can serve all the containers, and each layer of the image can be shared.

## 3.4 Creating and distributing mirrors

### 3.4.1 Making mirrors

Way 1: Commit an image from the container:

```

# 命令
docker commit -m="提交的描述信息" -a="作者" 容器id 要创建的目标镜像名:[标签名] 【也可省略 -m -a 参数】

# Command
docker commit -m="commit description" -a="author" container id target image name
to be created:[tag name] [you can also omit the -m -a parameter].

# 测试# Testing
jetson@ubuntu:~$ docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c54bf9efae47	ubuntu:latest	"/bin/bash"	3 hours ago	Up 24 minutes
	funny_hugle			
3b9c01839579	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago
	jovial_brown			

```
jetson@ubuntu:~$ docker commit c54bf9efae47 ubuntu:1.0
sha256:78ca7be949b6412f74ba12e8d16bd548aaa7c3fa25134326db3a67784f848f8f
jetson@ubuntu:~$ docker images # 生成了ubuntu:1.0镜像
# generated ubuntu:1.0 image
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	1.0	78ca7be949b6	5 seconds ago	69.2MB
yahboomtechnology/ros-foxy	3.4.0	49581aa78b6b	5 hours ago	24.3GB
yahboomtechnology/ros-foxy	3.3.9	cefb5ac2ca02	4 days ago	20.5GB
yahboomtechnology/ros-foxy	3.3.8	49996806c64a	4 days ago	20.5GB
yahboomtechnology/ros-foxy	3.3.7	8989b8860d17	5 days ago	17.1GB
yahboomtechnology/ros-foxy	3.3.6	326531363d6e	5 days ago	16.1GB
ubuntu	latest	bab8ce5c00ca	6 weeks ago	69.2MB
hello-world	latest	46331d942d63	13 months ago	9.14kB

Way 2, dockerfile to make an image:

```
# 命令
docker build -f dockerfile文件路径 -t 新镜像名字:TAG . # docker build 命令最后有一个
. 表示当前目录
# Command
docker build -f dockerfile file path -t new image name:TAG . # The docker build
command ends with a . at the end of the command.
# 测试# Testing
docker build -f dockerfile-ros2 -t yahboomtechnology/ros-foxy:1.2 .
```

关于dockerfile的编写请参考:

For more information on writing dockerfiles see:

[https://docs.docker.com/develop/develop-images/dockerfile\\_best-practices/](https://docs.docker.com/develop/develop-images/dockerfile_best-practices/)

## 3.4.2 Publishing Images

A docker repository is a centralized place to store image files. The largest public repository is docker hub (<https://hub.docker.com/>), which stores a huge number of images for users to download. Domestic public repositories include AliCloud, NetEaseCloud, and so on.

Steps to publish an image to docker hub:

- 1, the address: <https://hub.docker.com/> , first registered account
- 2, to ensure that the account can be normally logged in



3. Use the tag command to modify the image name

The specification for publishing an image to docker hub is:

```
docker push 注册用户名/镜像名
docker push register username/image name
```

For example, my registered user name here is: pengan88, then you have to change the mirror name first

```
# 命令:
docker tag 镜像ID 修改后的镜像名称

# command:
docker tag image id modified image name

# 测试# Testing
jetson@ubuntu:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
ubuntu              1.0         78ca7be949b6     5 seconds ago    69.2MB
ubuntu              latest      bab8ce5c00ca     6 weeks ago      69.2MB
hello-world         latest      46331d942d63     13 months ago    9.14kB
jetson@ubuntu:~$ docker tag 78ca7be949b6 pengan88/ubuntu:1.0
jetson@ubuntu:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED           SIZE
pengan88/ubuntu     1.0         78ca7be949b6     23 minutes ago    69.2MB
ubuntu              1.0         78ca7be949b6     23 minutes ago    69.2MB
ubuntu              latest      bab8ce5c00ca     6 weeks ago      69.2MB
hello-world         latest      46331d942d63     13 months ago    9.14kB
```

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

```
jetson@ubuntu:~$ docker login -u pengan88
Password:      # 这里输入docker hub注册的账号密码
# Enter the password of your docker hub account here.
WARNING! Your password will be stored unencrypted in
/home/jetson/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
jetson@ubuntu:~$ docker push pengan88/ubuntu:1.0
The push refers to repository [docker.io/pengan88/ubuntu]
ca774712d11b: Pushed
874b048c963a: Mounted from library/ubuntu
1.0: digest:
sha256:6767d7949e1c2c2adffbc5d3c232499435b95080a25884657fae366ccb71394d size:
736
```

5, visit docker hub can be viewed to have been released successfully

Want faster and simpler Kubernetes development? Test out Telepresence for Docker today.

docker hub

[Explore](#) [Repositories](#) [Organizations](#) [Help](#)

Upgrade

pengan88

pengan88

Repositories

ubuntu

General

Using 0 of 1 private repositories. [Get more](#)

General

Tags

Builds

Collaborators

Webhooks

Settings

Add a short description for this repository

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

[Update](#)

pengan88 / ubuntu

Description

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

Public View

To push a new tag to this repository,

```
docker push pengan88/ubuntu:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
1.0		Image	---	a few seconds ago

[See all](#) [Go to Advanced Image Management](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

Upgrade