

# Common commands for docker image containers

## 2.1. Do not use sudo commands

Normally, you need to add the prefix sudo to operate docker commands, as follows:

```
sudo docker version
```

However, after adding the docker user group, you do not need to add the sudo prefix. How to add the docker user group (run the command in the host running docker):

```
sudo groupadd docker # Add docker user group
sudo gpasswd -a $USER docker # Add the current user to the docker user group,
where $USER can be automatically resolved to the currently logged in user
newgrp docker # Update the docker user group
```

After adding the above commands, use the [docker images] command to test. If there is no error, it means that the sudo command can no longer be used. If the following error is reported:

```
pi@ubuntu:~$ docker images
WARNING: Error loading config file: /home/pi/.docker/config.json: open
/home/pi/.docker/config.json: permission denied
```

Then execute the following command in the host to solve it:

```
sudo chown "$USER":"$USER" /home/"$USER"/.docker -R
sudo chmod g+rwX "/home/$USER/.docker" -R
```

## 2.2, Help command

```
docker info # Display Docker system information, including images and container
numbers. .
docker --help # Help
```

## 2.3. Image command

### 1. docker pull download image

```
# Download image
jetson@ubuntu:~$ docker pull ubuntu
Using default tag: latest # No tag, default is latest
latest: Pulling from library/ubuntu
cd741b12a7ea: Pull complete # Layered download
Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babc295a0428a6d21
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest # Real location
```

## 2. docker images list images

```
# List the images on the local host
jetson@ubuntu:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
yahboomtechnology/ros-foxy	3.4.0	49581aa78b6b	About an hour ago	24.3GB
yahboomtechnology/ros-foxy	3.3.9	cefb5ac2ca02	3 days ago	20.5GB
yahboomtechnology/ros-foxy	3.3.8	49996806c64a	4 days ago	20.5GB
yahboomtechnology/ros-foxy	3.3.7	8989b8860d17	4 days ago	17.1GB
yahboomtechnology/ros-foxy	3.3.6	326531363d6e	5 days ago	16.1GB
hello-world	latest	46331d942d63	13 months ago	9.14kB

```
# Explanation
REPOSITORY: The repository source of the image
TAG: The tag of the image
IMAGE: ID The ID of the image
CREATED: The creation time of the image
SIZE: The size of the image

# The same repository source can have multiple tags, representing different
versions of this repository source. We use REPOSITORY:TAG to define different
images. If you do not define the tag version of the image, docker will use the
latest image by default!

# Optional
-a: List all local images
-q: Show only the image id
--digests: Show summary information of the image
```

## 3. docker search Search image

```
# Search for images
jetson@ubuntu:~$ docker search ros2
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
osrf/ros2	**Experimental** Docker Images for ROS2 deve...	60	[OK]	
tiryoh/ros2-desktop-vnc	A Docker image to provide HTML5 VNC interfac...	11		
althack/ros2	An assortment of development containers for ...	7		
tiryoh/ros2	unofficial ROS2 image	6		
athackst/ros2	[Deprecated-> use althack/ros2]	5		
uobflightlabstarling/starling-mavros2	ROS2 version of MAVROS 2			
theosakamg7/ros2_java_docker	Image base	1	[OK]	

```
# docker search The name of an image corresponds to the image in the DockerHub
repository
# Optional
--filter=stars=50: List images with a collection count not less than the
specified value.
```

## 4. Docker rmi deletes images

```
# Delete images
docker rmi -f image id # Delete a single one
docker rmi -f image name: tag image name: tag # Delete multiple
docker rmi -f $(docker images -qa) # Delete all
```

## 2.4. Container commands

You can only create a container if you have an image. Here we use the ubuntu image to test. Download the image:

```
docker pull ubuntu
```

1. Docker run runs the image to start the container

```
# Command
docker run [OPTIONS] IMAGE [COMMAND][ARG...]
# Common parameter description
--name="Name" # Assign a name to the container
-d # Run the container in the background and return the container id!
-i # Run the container in interactive mode, used with -t
-t # Reassign a terminal to the container, usually used with -i
-P # Random port mapping (uppercase)
-p # Specify port mapping (summary), generally can be written in four ways
ip:hostPort:containerPort
ip::containerPort
hostPort:containerPort (common)
containerPort

# Test
jetson@ubuntu:~$ docker images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
yahboomtechnology/ros-foxy  3.4.0       49581aa78b6b  2 hours ago   24.3GB
yahboomtechnology/ros-foxy  3.3.9       cefb5ac2ca02  3 days ago    20.5GB
yahboomtechnology/ros-foxy  3.3.8       49996806c64a  4 days ago    20.5GB
yahboomtechnology/ros-foxy  3.3.7       8989b8860d17  4 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

#Use ubuntu to start the container in interactive mode and execute the /bin/bash
command in the container!
jetson@ubuntu:~$ docker run -it ubuntu:latest /bin/bash
root@c54bf9efae47:/# ls
bin boot dev etc home lib media mnt opt ••proc root run sbin srv sys tmp usr var
root@c54bf9efae47:/# exit # Use exit to exit the container and return to the
host
exit
jetson@ubuntu:~$
```

2. docker ps lists all running containers

```
# Command
docker ps [OPTIONS]

# Common parameter description
-a # List all currently running containers + historically run containers
-l # Display the most recently created container
-n=? # Display the most recently created n containers
-q # Silent mode, only display the container number.

# Test
jetson@ubuntu:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c54bf9efae47	ubuntu:latest	"/bin/bash"	2 hours ago	Up 4 seconds
	funny_hugle			
3b9c01839579	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago
	jovial_brown			

### 3. Exit the container

```
exit # Stop the container and exit
ctrl+P+Q # Exit the container without stopping
```

### 4. Enter the running container from multiple terminals

```
# Command 1
docker exec -it container id bashShell

# Test
jetson@ubuntu:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c54bf9efae47	ubuntu:latest	"/bin/bash"	2 hours ago	Up 4 seconds
	funny_hugle			
3b9c01839579	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago
	jovial_brown			

```
jetson@ubuntu:~$ docker exec -it c5 /bin/bash # The container ID can be
abbreviated as long as it can uniquely identify the container
root@c54bf9efae47:/#

# Command 2
docker attach container ID

# Test
jetson@ubuntu:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
c54bf9efae47	ubuntu:latest	"/bin/bash"	2 hours ago	Up 35 seconds
	funny_hugle			
3b9c01839579	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago
	jovial_brown			

```
jetson@ubuntu:~$ docker attach c5 # The container ID can be abbreviated as long
as it can uniquely identify the container
root@c54bf9efae47:/#

# Difference
# exec opens a new terminal in the container and can start a new process
```

```
# attach Directly enter the terminal of the container start command, and no new process will be started
```

## 5. Start and stop containers

```
docker start (container id or container name) # Start container
```

```
docker restart (container id or container name) # Restart container
```

```
docker stop (container id or container name) # Stop container
```

```
docker kill (container id or container name) # Force stop container
```

## 6. Delete container

```
docker rm container id # Delete the specified container
```

```
docker rm -f $(docker ps -a -q) # Delete all containers
```

```
docker ps -a -q|xargs docker rm # Delete all containers
```

## 2.5. Other commonly used commands

### 1. View the process information running in the container, supporting ps command parameters.

```
# Command
```

```
docker top container id
```

```
# Test
```

```
jetson@ubuntu:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
c54bf9efae47	ubuntu:latest	"/bin/bash"	2 hours ago	Up 2 minutes
	funny_hugle			
3b9c01839579	hello-world	"/hello"	3 hours ago	Exited (0) 3 hours ago
	jovial_brown			

```
jetson@ubuntu:~$ docker top c5
```

UID	PID	PPID	C
STIME	TTY	TIME	CMD
root	9667	9647	0
14:20	pts/0	00:00:00	/bin/bash

### 2. View the metadata of the container/image

```
# Command
```

```
docker inspect container id
```

```
# Test to view container metadata
```

```
jetson@ubuntu:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
c54bf9efae47	ubuntu:latest	"/bin/bash"	2 hours ago	Up 4 minutes
	funny_hugle			

```

3b9c01839579    hello-world    "/hello"        3 hours ago    Exited (0) 3 hours ago
jovial_brown
jetson@ubuntu:~$ docker inspect c54bf9efae47
[
  {
    # 完整的id, 这里上面的容器id, 就是截取的这个id前几位
    "Id":
    "c54bf9efae471071391202a8718b346d9af76cb1ff17741e206280603d6f0056",
    "Created": "2023-04-24T04:19:46.232822024Z",
    "Path": "/bin/bash",
    "Args": [],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 9667,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-04-24T06:20:58.508213216Z",
      "FinishedAt": "2023-04-24T06:19:45.096483592Z"
    },
    . . . .

# Test to view image metadata
jetson@ubuntu:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
ubuntu               latest       bab8ce5c00ca     6 weeks ago     69.2MB
hello-world         latest      46331d942d63     13 months ago   9.14kB
jetson@ubuntu:~$ docker inspect bab8ce5c00ca
[
  {
    "Id":
    "sha256:bab8ce5c00ca3ef91e0d3eb4c6e6d6ec7cffa9574c447fd8d54a8d96e7c1c80e",
    "RepoTags": [
      "ubuntu:latest"
    ],
    "RepoDigests": [
      "ubuntu@sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babbc295a0428a6d21"
    ],
    "Parent": "",
    "Comment": "",
    "Created": "2023-03-08T04:32:41.063980445Z",
    "Container":
    "094fd0c521be8c84d81524e4a5e814e88a2839899c56f654484d32d171c7195b",
    "ContainerConfig": {
      "Hostname": "094fd0c521be",
      . . . . .
      "Labels": {
        "org.opencontainers.image.ref.name": "ubuntu",
        "org.opencontainers.image.version": "22.04"
      }
    },
    "Dockerversion": "20.10.12",
    "Author": "",

```

```

    "Config": {
      "Hostname": "",
      .....
      "Labels": {
        "org.opencontainers.image.ref.name": "ubuntu",
        "org.opencontainers.image.version": "22.04"
      }
    },
    "Architecture": "arm64",
    "Variant": "v8",
    "Os": "linux",
    "Size": 69212233,
    "VirtualSize": 69212233,
    "GraphDriver": {
      "Data": {
        "MergedDir":
"/var/lib/docker/overlay2/8418b919a02d38a64ab86060969b37b435977e9bbdeb6b0840d4eb
698280e796/merged",
        "UpperDir":
"/var/lib/docker/overlay2/8418b919a02d38a64ab86060969b37b435977e9bbdeb6b0840d4eb
698280e796/diff",
        "WorkDir":
"/var/lib/docker/overlay2/8418b919a02d38a64ab86060969b37b435977e9bbdeb6b0840d4eb
698280e796/work"
      },
      "Name": "overlay2"
    },
    "RootFS": {
      "Type": "layers",
      "Layers": [

"sha256:874b048c963ab55b06939c39d59303fb975d323822a4ea48a02ac8dc635ea371"
      ]
    },
    "Metadata": {
      "LastTagTime": "0001-01-01T00:00:00Z"
    }
  }
]

```

## 2.6、命令小结

---

