2. docker image container common command

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

2.1. Not using sudo commands

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

```
sudo docker version
```

But after you add a docker usergroup, you can do so without the sudo prefix. docker usergroups are added (by running commands within the host where docker is running):

```
sudo groupadd docker          # 添加docker用户组
 # Add docker user group
sudo gpasswd -a $USER docker  # 将当前用户添加至docker用户组，其中$USER可以自动解析到当前登陆的用户
# Add the current user to the docker user group, where $USER can be automatically resolved to the currently logged in user
newgrp docker                 # 更新docker用户组
# Update the docker user group
```

After adding the above commands, use the [docker images] command to test, if no error is reported, it means that you can not use the sudo command. 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 machine to solve the problem:

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

2.2. Help command

```
docker info    # 显示 Docker 系统信息，包括镜像和容器数。。
 # Display Docker system information, including the number of images and
containers.
docker --help  # 帮助# Help
```

2.3. Mirror commands

1, docker pull download image

```
# 下载镜像# Download mirrors
jetson@ubuntu:~$ docker pull ubuntu
Using default tag: latest                   # 不写tag，默认是latest  # No tag,
default is latest
latest: Pulling from library/ubuntu
cd741b12a7ea: Pull complete                 # 分层下载
# Layered downloads
Digest: sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babc295a0428a6d21
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest             # 真实位置
                                       # True location
```

2. docker images lists the mirrors

```
# 列出本地主机上的镜像
# List mirrors 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

# 解释
REPOSITORY 镜像的仓库源
TAG 镜像的标签
IMAGE ID 镜像的ID
CREATED 镜像创建时间
SIZE 镜像大小

# 同一个仓库源可以有多个TAG，代表这个仓库源的不同版本，我们使用REPOSITORY：TAG 定义不同的镜
像，如果你不定义镜像的标签版本，docker将默认使用 lastest镜像！
```

```
# 可选项
-a： 列出本地所有镜像
-q： 只显示镜像id
--digests： 显示镜像的摘要信息
# Explanation
REPOSITORY The repository source of the image
TAG The tag of the image
IMAGE ID The ID of the image
CREATED The time the mirror was created
SIZE The size of the mirror

# The same repository source can have multiple TAGs representing different
versions of this repository source, we use REPOSITORY: TAG to define different
mirrors, if you don't define the tagged version of the mirror, docker will use
the lastest mirror by default!

# Optional
-a: list all local mirrors
-q: show only mirror ids
--digests: show digests of the mirrors
```

3, docker search search mirror

```
# 搜索镜像# Search mirrors
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 某个镜像的名称 对应DockerHub仓库中的镜像
# 可选项
--filter=stars=50 ： 列出收藏数不小于指定值的镜像。
# docker search Name of an image Corresponding to an image in the DockerHub
repository
# Optional
--filter=stars=50 : Lists mirrors with a favorite number not less than the
specified value.
```

4. docker rmi delete image

```
# 删除镜像
docker rmi -f 镜像id # 删除单个
docker rmi -f 镜像名:tag 镜像名:tag # 删除多个
docker rmi -f $(docker images -qa) # 删除全部
# Delete a mirror
docker rmi -f mirror id # Delete single
docker rmi -f mirror name:tag mirror name:tag # Delete Multiple
docker rmi -f $(docker images -qa) # delete all
```

2.4 Container Commands

You can create a container only if you have a mirror, we use ubuntu's mirror to test here, download the mirror:

```
docker pull ubuntu
```

1. docker run Run the image to start the container

```
# 命令# Command
docker run [OPTIONS] IMAGE [COMMAND][ARG...]
# 常用参数说明
--name="Name" # 给容器指定一个名字
-d # 后台方式运行容器，并返回容器的id!
-i # 以交互模式运行容器，通过和 -t 一起使用
-t # 给容器重新分配一个终端，通常和 -i 一起使用
-P # 随机端口映射（大写）
-p # 指定端口映射（小结），一般可以有四种写法
ip:hostPort:containerPort
ip::containerPort
hostPort:containerPort (常用)
containerPort
# Description of common parameters
--name="Name" # Assign a name to the container.
-d # Run the container in background mode and return the container's id!
-i # Runs 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), which can be written in four general ways
ip:hostPort:containerPort
ip::containerPort
hostPort:containerPort (common)
containerPort

#测试#Testing
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

#使用ubuntu进行用交互模式启动容器，在容器内执行/bin/bash命令！
# Use ubuntu to start the container in interactive mode and execute the /bin/bash
command inside 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       # 使用 exit 退出容器回到宿主机
# Use exit to exit the container back to the host
exit
jetson@ubuntu:~$
```

2. docker ps lists all running containers

```
# 命令
docker ps [OPTIONS]
# 常用参数说明
-a # 列出当前所有正在运行的容器 + 历史运行过的容器
-l # 显示最近创建的容器
-n=? # 显示最近n个创建的容器
-q # 静默模式，只显示容器编号。

# Command
docker ps [OPTIONS]
# Description of common parameters
-a # List all currently running containers + history of running containers
-l # Show recently created containers
-n=? # Show the last n containers created
-q # Silent mode, only container numbers are shown.

#测试#Testing
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE           COMMAND       CREATED       STATUS
  PORTS     NAMES
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        # 容器停止退出
ctrl+P+Q    # 容器不停止退出
exit # The container stops exiting
ctrl+P+Q # container does not stop exit
```

4. Multi-terminal access to running containers

```
# 命令1
```

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

# 测试# Testing
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE            COMMAND       CREATED       STATUS
  PORTS     NAMES
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   # 容器的id可以简写，只要是能唯一标识这个容器就行
# The id of the container can be abbreviated, as long as it uniquely identifies the container
root@c54bf9efae47:/#

# 命令2
docker attach 容器id

# Command 2
docker attach container id

# 测试# Testing
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE            COMMAND       CREATED       STATUS
  PORTS     NAMES
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    # 容器的id可以简写，只要是能唯一标识这个容器就行
# The id of the container can be abbreviated, as long as it uniquely identifies the container
root@c54bf9efae47:/#

# 区别
# exec 是在容器中打开新的终端，并且可以启动新的进程
# attach 直接进入容器启动命令的终端，不会启动新的进程
# Difference
# exec opens a new terminal in the container and starts a new process.
# attach opens a new terminal in the container and starts a new process.
```

5. Start-stop container

```
docker start (容器id or 容器名)       # 启动容器
docker restart (容器id or 容器名)     # 重启容器
docker stop (容器id or 容器名)        # 停止容器
docker kill (容器id or 容器名)        # 强制停止容器

docker start (container id or container name) # start the container
docker restart (container id or container name) # Restart the container
docker stop (container id or container name) # Stop the container
docker kill (container id or container name) # Force the container to stop
```

6. Deletion of containers

```
docker rm 容器id                    # 删除指定容器
docker rm -f $(docker ps -a -q)   # 删除所有容器
docker ps -a -q|xargs docker rm   # 删除所有容器

docker rm container id # Remove the specified container
docker rm -f $(docker ps -a -q) # Remove all containers
docker ps -a -q|xargs docker rm # Remove all containers
```

2.5 Other common commands

1. View information about the processes running in the container, supporting the ps command parameter.

```
# 命令
docker top 容器id

# command
docker top container id

# 测试# Testing
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 container / image metadata

```
# 命令
docker inspect 容器id

# command
docker inspect container id

# 测试查看容器元数据  # 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
```

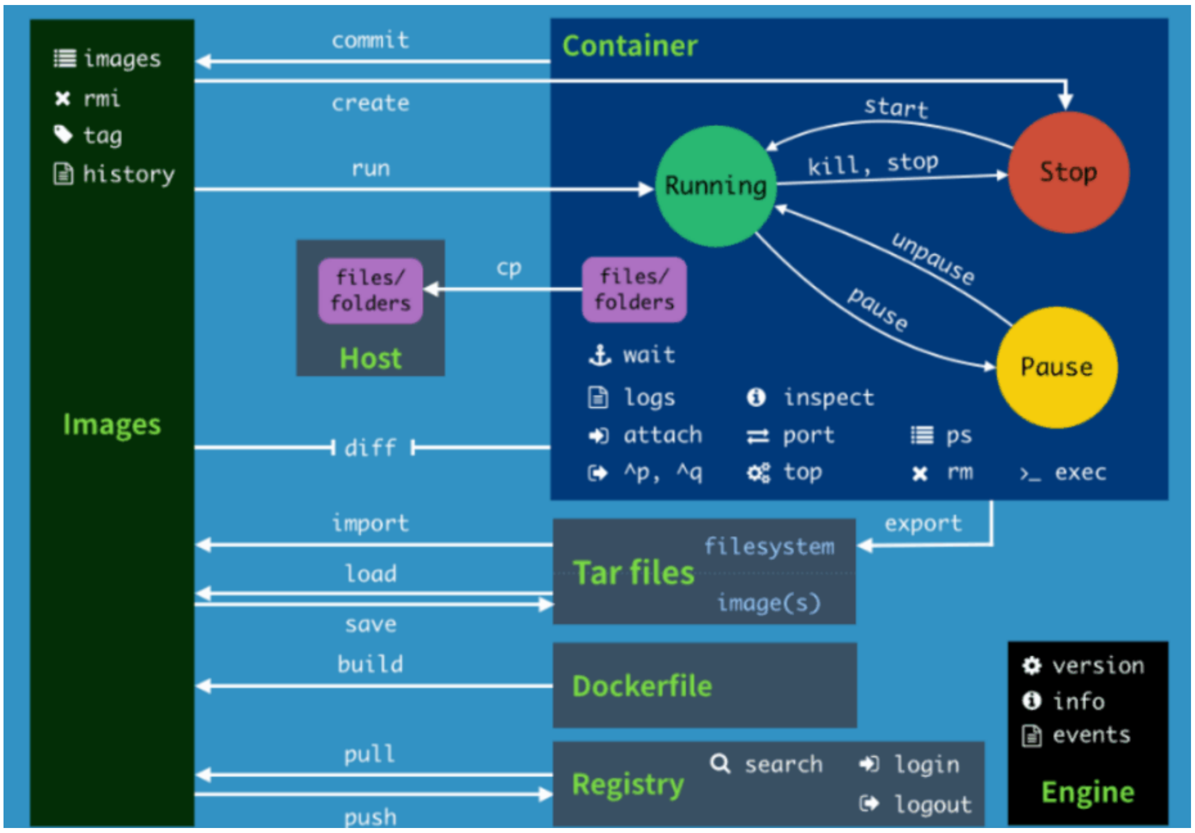
```
3b9c01839579   hello-world     "/hello"      3 hours ago   Exited (0) 3 hours ago
            jovial_brown
jetson@ubuntu:~$ docker inspect c54bf9efae47
[
    {
        # 完整的id，这里上面的容器id，就是截取的这个id前几位
        # full id, here the container id above, is the first few digits of this
id intercepted
        "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"
        },
    。。。。


# 测试查看镜像元数据 # Test to view mirror 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@sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118ef3babc295a0428a6d21"
        ],
        "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 Summary of orders


images
rmi
tag
history
Images
commit
create
run
Container
start
kill, stop
Running
Stop
unpause
pause
Pause
files/ folders
Host
cp
files/ folders
wait
logs
inspect
attach
port
ps
^p, ^q
top
rm
exec
diff
import
export
Tar files
filesystem
image(s)
load
save
build
Dockerfile
pull
push
Registry
search
login
logout
version
info
events
Engine