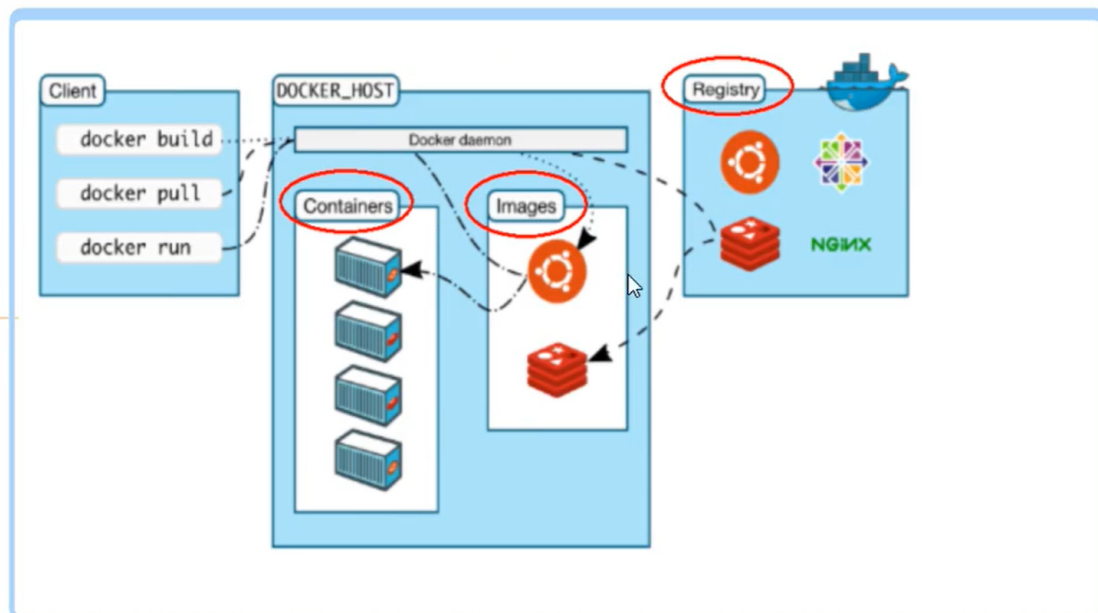


启动类命令

- 启动docker: `systemctl start docker`
- 停止docker: `systemctl stop docker`
- 重启docker: `systemctl restart docker`
- 查看docker状态: `systemctl status docker`
- 开机启动: `systemctl enable docker`
- 查看docker概要信息: `docker info`
- + 查看docker总体帮助文档: `docker --help` +
- + 查看docker命令帮助文档: `docker 具体命令 --help`

镜像类命令





```
# 查看已有镜像
docker images
docker image -a
docker image -q

# 查看仓库中的镜像
docker search [image-name]
docker search --limit [nums] [image-name]
docker search --limit 5 redis

# 从仓库中拉取镜像
docker pull image-name
docker pull image-name[:TAG]
docker pull ubuntu
docker pull redis:6.0.8

# 删除镜像
docker rmi -f 镜像ID #删除单个
docker rmi -f 镜像名1:TAG 镜像名2:TAG #删除多个
docker rmi -f $(docker images -qa) #删除全部
```

• 演示

```
[root@atquigu 桌面]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
tomcat	latest	cb315192e016	7 days ago	465.3 MB
hello-world	latest	83f0de727d85	5 weeks ago	1.848 kB

各个选项说明:

REPOSITORY: 表示镜像的仓库源
TAG: 镜像的标签版本号
IMAGE ID: 镜像ID
CREATED: 镜像创建时间
SIZE: 镜像大小

同一仓库源可以有多个 TAG 版本, 代表这个仓库源的不同个版本, 我们使用 REPOSITORY:TAG 来定义不同的镜像。
如果你不指定一个镜像的版本标签, 例如你只使用 `ubuntu`, `docker` 将默认使用 `ubuntu:latest` 镜像

案例 **docker search [image]**

参数	说明
NAME	镜像名称
DESCRIPTION	镜像说明
STARS	点赞数量
OFFICIAL	是否是官方的
AUTOMATED	是否是自动构建的

```

[root@zzyy ~]# docker search redis
NAME                DESCRIPTION                STARS     OFFICIAL    AUTOMATED
redis                Redis is an open source key-value store that... 10018     [OK]        [OK]
sameersbn/redis      Redis is an open source key-value store that... 83        [OK]        [OK]
grokzen/redis-cluster Redis cluster 3.0, 3.2, 4.0, 5.0, 6.0, 6.2     79
rediscommander/redis-commander Alpine image for redis-commander - Redis man... 66
redislabs/redisearch Redis With the RedisSearch module pre-loaded... 39
redislabs/redisinsight RedisInsight - The GUI for Redis                35
kubeguide/redis-master redis-master with "Hello World!"                33
redislabs/redis      Clustered in-memory database engine compatib... 31
oliver006/redis_exporter Prometheus Exporter for Redis Metrics. Supp... 30
redislabs/rejson     RedisJSON - Enhanced JSON data type processi... 27
arm32v7/redis        Redis is an open source key-value store that... 25
  
```

docker images

OPTIONS说明:

- a :列出本地所有的镜像 (含历史映像层)
- q :只显示镜像ID。

docker search 某个XXX镜像名字

命令

- 网站: <https://hub.docker.com>
- docker search [OPTIONS] 镜像名字
- 案例
- OPTIONS说明: 类似于限量查询 **docker search --list [nums] [image]**

docker pull 某个XXX镜像名字

7

下载镜像

docker pull 镜像名字[:TAG] 即下载指定版本的镜像文件

docker pull redis:6.0.8

没有TAG就是最新版

等价于

docker pull 镜像名字:latest

docker pull ubuntu

docker pull 镜像名字

docker pull ubuntu

docker rmi 某个XXX镜像名字ID

删除镜像

删除单个: **docker rmi -f 镜像ID**

删除多个: **docker rmi -f 镜像名1:TAG 镜像名2:TAG**

删除全部: **docker rmi -f \$(docker images -qa)**

容器命令

3.1、docker run [options]

查看使用文档: `docker run --help`

OPTIONS说明（常用）：有些是一个减号，有些是两个减号

--name="容器新名字" 为容器指定一个名称；

-d: 后台运行容器并返回容器ID，也即启动守护式容器(后台运行)；

-i: 以交互模式运行容器，通常与 **-t** 同时使用；

-t: 为容器重新分配一个伪输入终端，通常与 **-i** 同时使用；

也即启动交互式容器(前台有伪终端，等待交互)；

-P: 随机端口映射，大写P^I

-p: 指定端口映射，小写p

参数	说明
-p hostPort:containerPort	端口映射 -p 8080:80
-p ip:hostPort:containerPort	配置监听地址 -p 10.0.0.100:8080:80

示例：

```
# 查看正在运行的容器
docker ps
# 容器命令
docker run -it --name=自定义容器名称 /bin/bash
# or
docker run -it bash
```

效果：

```
[root@iZf8z3pu4d7ueh3i6pzb5fZ home]# docker run -it ubuntu /bin/bash
root@f31402b0d8cb: /#
                                方式1

[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker run -it --name=myu1 ubuntu bash
root@d5c7962a5554: /#
                                方式2

[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
f31402b0d8cb   ubuntu   "/bin/bash"   19 seconds ago   Up 19 seconds           romantic_euclid

[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
d5c7962a5554   ubuntu   "bash"     5 seconds ago   Up 4 seconds           myu1
f31402b0d8cb   ubuntu   "/bin/bash"   About a minute ago   Up About a minute       romantic_euclid

[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]#
```

查看容器

3.2、docker ps [options]

查看正在运行的容器

查看文档 `docker ps --help`

```
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps --help
Usage:  docker ps [OPTIONS]

List containers

Options:
  -a, --all            Show all containers (default shows just running)
  -f, --filter filter  Filter output based on conditions provided
  --format string      Pretty-print containers using a Go template
  -n, --last int       Show n last created containers (includes all states) (default -1)
  -l, --latest         Show the latest created container (includes all states)
  --no-trunc          Don't truncate output
  -q, --quiet          Only display container IDs
  -s, --size           Display total file sizes
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]#
```

示例：

```
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
d5c7962a5554   ubuntu   "bash"                  8 minutes ago Up 8 minutes             myu1
f31402b0d8cb   ubuntu   "/bin/bash"            10 minutes ago Up 9 minutes             romantic_euclid
c85657da3a74   hello-world "/hello"              45 minutes ago Exited (0) 45 minutes ago hardcore_shirley
7c0ec7ea0f46   hello-world "/hello"              47 minutes ago Exited (0) 47 minutes ago dreamy_yalow
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps -l
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
d5c7962a5554   ubuntu   "bash"                  8 minutes ago Up 8 minutes             myu1
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps -n 2
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
d5c7962a5554   ubuntu   "bash"                  8 minutes ago Up 8 minutes             myu1
f31402b0d8cb   ubuntu   "/bin/bash"            10 minutes ago Up 10 minutes            romantic_euclid
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]# docker ps -q
d5c7962a5554
f31402b0d8cb
[root@iZf8z3pu4d7ueh3i6pzb5fZ ~]#
```

OPTIONS说明（常用）：

- a :列出当前所有正在运行的容器+历史上运行过的
- 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。

查看容器内部的细节：

```
docker inspect 容器ID
```

3.3、docker 启动类命令



前台交互式启动和后台守护式启动

- 前台交互式启动 — `docker run -it redis:6.0.8`
- 后台守护式启动 — `docker run -d redis:6.0.8`

注意

这个是docker的机制问题,比如你的web容器,我们以nginx为例,正常情况下,我们配置启动服务只需要启动响应的service即可。例如`service nginx start`但是,这样做,nginx为后台进程模式运行,就导致docker前台没有运行的应用,这样的容器后台启动后,会立即自杀因为他觉得他没事可做了。

所以,最佳的解决方案是,将你要运行的程序以前台进程的形式运行,常见就是命令行模式,表示我还有交互操作,别中断, `O(∩_∩)O`哈哈~

关于守护式进程的小问题: 那为什么执行完: `docker run -d redis:6.0.8` 后再 `docker ps` 却发现 redis 容器并没有停止呢?

因为docker启动ubuntu时并没有进程来夯住容器,所以就会退出。而redis启动是会有进程来夯住容器,不让他退出。redis一直在监听端口啊,当然不会自杀,想不到吗? 因为ubuntu没有前台进程,redis有。

3.4、docker 重新进去已启动服务

- `docker exec -it 容器ID /bin/bash`

`exec`是在容器打开新的终端,并且可以启动新的进程,用`exit`退出不会导致容器的停止

```
# 查看当前存在的容器示例的运行情况
docker ps -a

# docker restart 容器名/ 容器ID

# 以前台交互方式重新进入正在运行的容器
docker exec -it 容器ID /bin/bash

# ctrl + p + q 切换至后台运行
exit # 退出
docker ps -a # 发现容器仍在运行
```

- docker attach 容器ID

attach直接进入容器启动命令的终端，不会启动新的进程，用exit退出会导致容器的停止

```
docker attach 容器ID
exit # 退出
docker ps -a # 发现容器就停止了
```

注意：

Ctrl + p + q 后台运行容器，但是当你在使用 run 是重新创建一个新的容器，而 exec 是进入之前创建的容器

3.5、容器文件备份与导入出

- 容器 ==> 主机

```
# 注意是在主机的命令行下
docker cp 容器ID:容器路径 目标主机的路径
docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH

# 例如
[root@izf8z3pu4d7ueh3i6pzv5fz ~]# docker cp d5c7962a5554:/secret.txt /tmp
[root@izf8z3pu4d7ueh3i6pzv5fz ~]# ls /tmp | grep secret
secret.txt
[root@izf8z3pu4d7ueh3i6pzv5fz ~]#
```

- 容器的导入导出

```
docker ps -a
d5c7962a5554    ubuntu          "bash"          2 days ago    Exited
(137) 4 seconds ago          myu1

# 导出
docker export d5c7962a5554 > myubuntu.tar
```

11

```
-rw-r--r-- 1 root root 80358912 Sep 26 09:36 myubuntu.tar
```

删除容器实例

```
docker rm -f d5c7962a5554
```

从tar包中的内容创建一个新的文件系统再导入为镜像

```
cat tar包名.tar | docker import - 镜像用户/镜像名:版本号
```

```
cat myubuntu.tar | docker import - xxr/ubuntu:1.0
```

```
[root@izf8z3pu4d7ueh3i6pzv5fz ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
xxr/ubuntu	1.0	d37efa8a6d92	8 seconds ago	77.8MB
ubuntu	latest	2dc39ba059dc	3 weeks ago	77.8MB