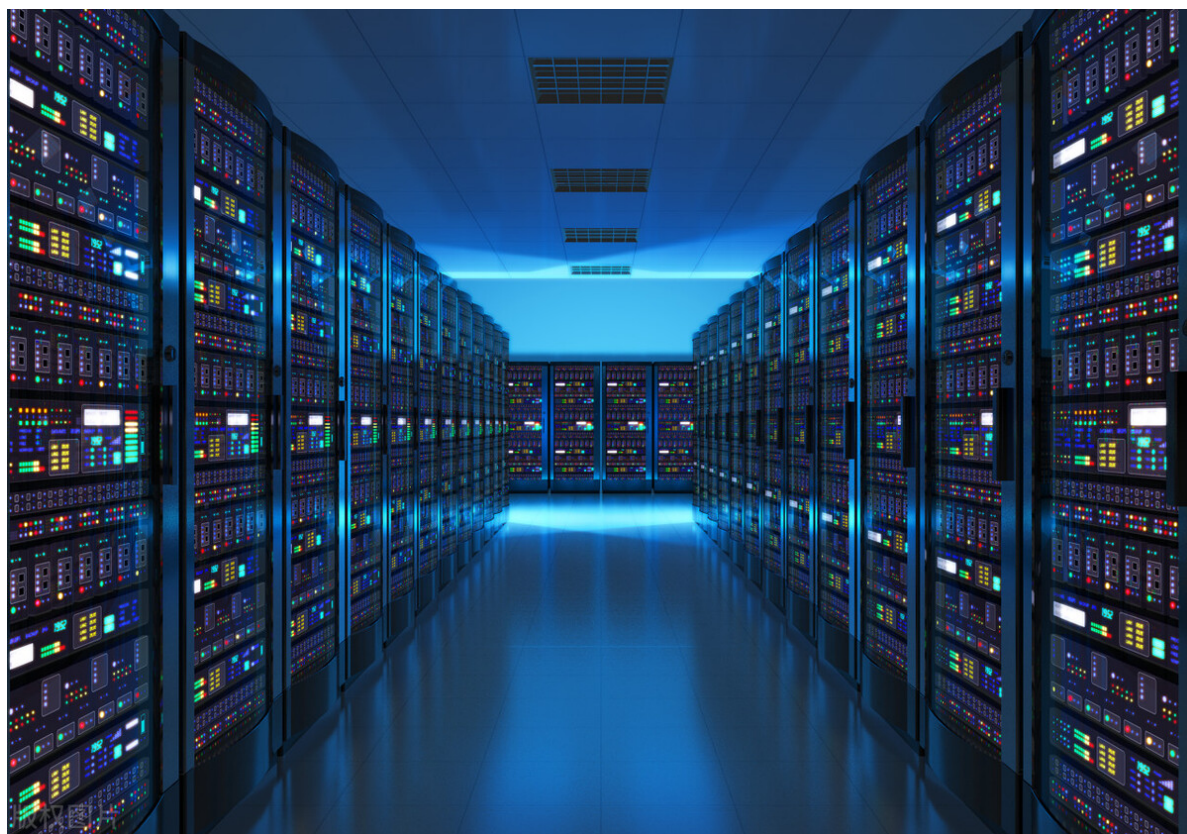


什么是虚拟化技术？

对于一个服务器来说，大多数时候资源都是闲置状态，并没有得到充分利用。所以，对于一些初创公司来说，还没有稳定业务，就自建机房，配置大量服务器，还有高昂的运维成本。

因此，现在大多数公司在初期都将业务放到云端，需要多少资源，就租用相应规模的服务器。云服务提供商会将服务器划分成多份，并且相互隔离，这就是虚拟化技术。



比如，刚毕业的学生，开始工资不高，不仅无法买房，也租不起大的房子，很多选择合租。房东将整个房子划分成几个单间，分别出租，这也类似于虚拟化的原理。

什么是Docker？

Docker是一个开源的容器引擎，它基于LXC容器技术，使用Go语言开发。

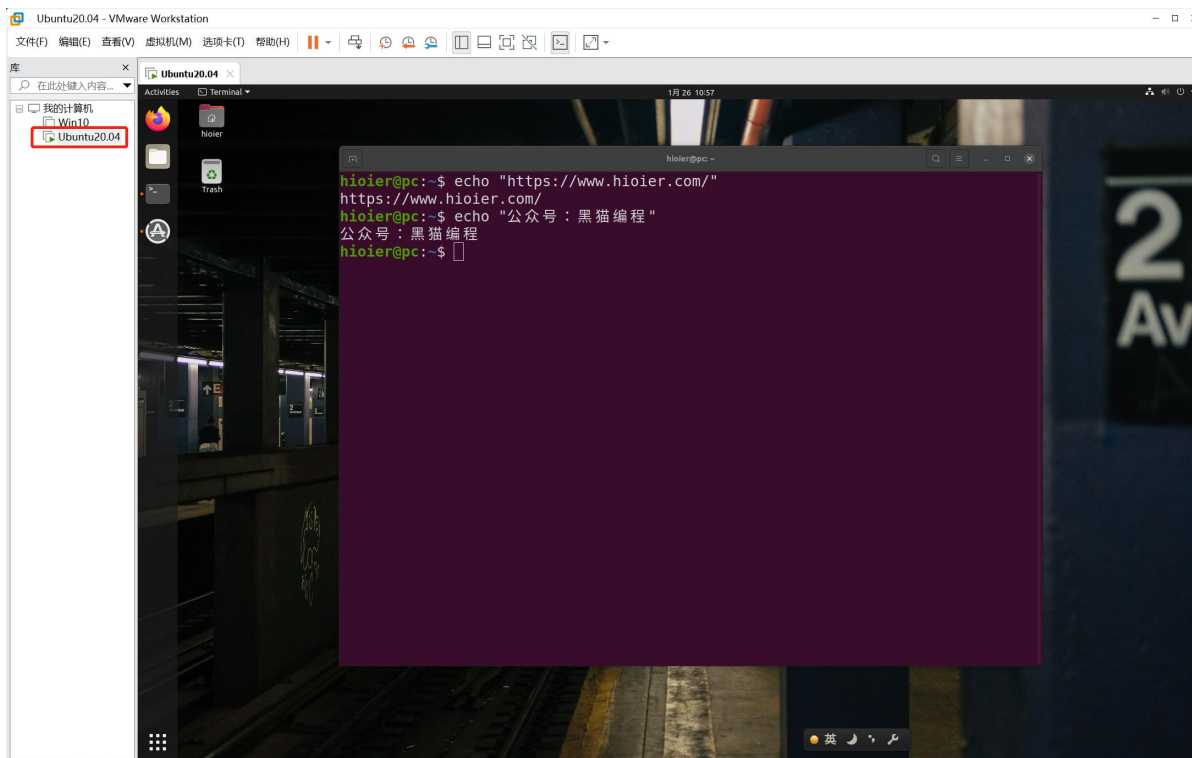
源代码托管在Github上，并遵从Apache2.0协议。

Docker采用C/S架构，其可以轻松的为任何应用创建一个轻量级的、可移植的、自给自足的容器。

简单来说：Docker就是一种快速解决生产问题的一种技术手段。

Docker是容器的一种，而容器不仅仅是Docker。

早期的虚拟化比如VMWare虚拟机，一个虚拟机可以同时运行多个操作系统。但是，**每个虚拟机都要虚拟化单独的硬件资源，这样，就会消耗更多的资源，主服务器可以虚拟化出的独立空间就越小。**



而容器技术最显著的特点是所有容器都可以共有主服务器的硬件资源，这样就使得容器更加轻量化，启动速度也更快。但是Docker只能支持Linux系统，所有容器都共用一个Linux内核。

容器：顾名思义，就是可以装物品的东西。每个容器可以按照我们的需求装物品，不同容器间相互隔离，携带方便。比如，一家人外出旅行，每个人都在自己的旅行包装上自己需要的生活物品。

或者说生活中很痛苦的事情就是电脑系统重装，要重复配置各种软件，但是只要提前制作一个系统盘，把常用的软件提前安装好，重装系统后自动包括所有的软件。



在云平台上，容器装载的物品就是不同应用。比如，公司在A云平台部署了自己的项目，现在发现B云平台价格更加优惠，性能也更好，直接将A云平台容器打包切换到B云平台，而不需要重复配置环境，安装软件。

ubuntu安装Docker

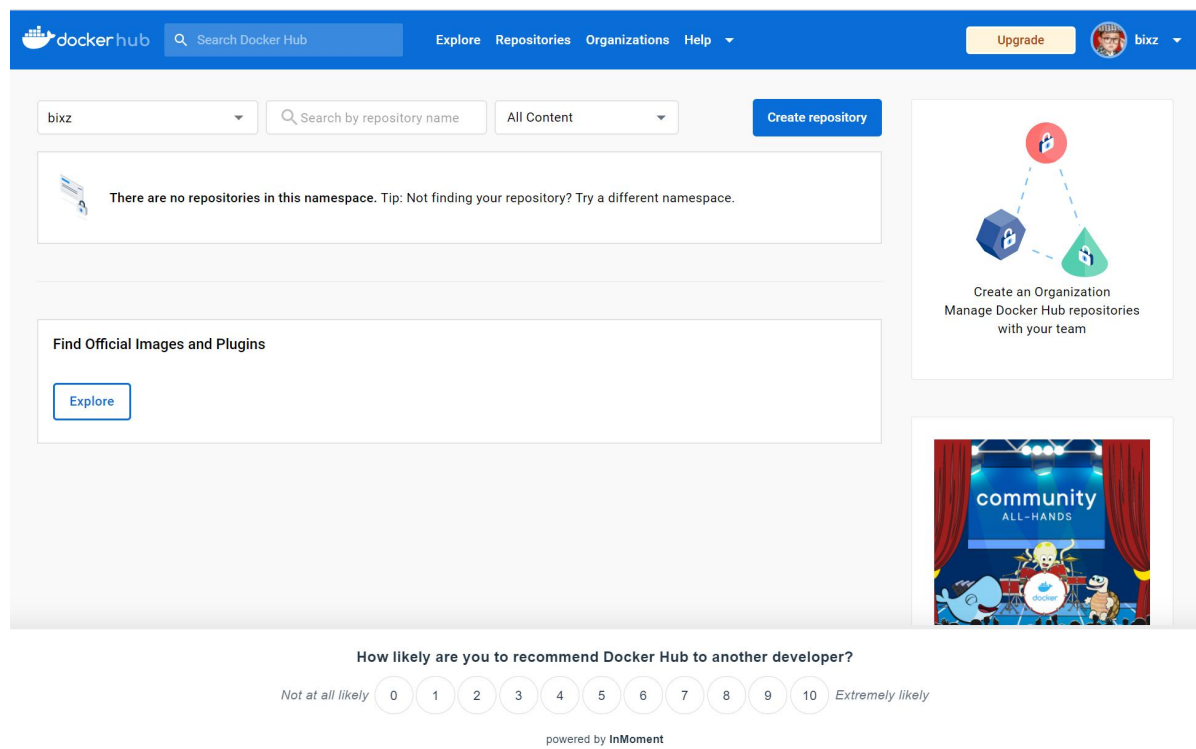
```
1 sudo apt install docker.io
2 docker --version
```

```
hioier@pc:~$ docker --version
Docker version 20.10.12, build 20.10.12-0ubuntu2~20.04.1
```

```
hioier@pc:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor prese
   Active: active (running) since Thu 2023-01-26 11:13:45 CST; 9min ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 3603 (dockerd)
      Tasks: 7
     Memory: 36.3M
    CGroup: /system.slice/docker.service
            └─3603 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/con
```

注册Docker账号

docker官网:



docker加速器：在国内使用docker的官方镜像源，会因为网络的原因，造成无法下载，或者一直处于超时，所以我们使用 daocloud 的方法进行加速配置。



注册 DaoCloud 帐号

用户名

邮箱

密码

验证码



注册

或使用以下帐号注册



Github



微信

已经注册过帐号? [立即登录](#) >



[产品](#) [客户](#) [动态](#) [合作伙伴](#) [关于我们](#)

[我的账户](#)

[EN](#)

配置 Docker 镜像站

Linux

```
curl -sSL https://get.daocloud.io/daotools/set_mirror.sh | sh -s http://f1361db2.m.daocloud.io
```

该脚本可以将 --registry-mirror 加入到你的 Docker 配置文件 /etc/docker/daemon.json 中。适用于 Ubuntu14.04、Debian、CentOS6、CentOS7、Fedora、Arch Linux、openSUSE Leap 42.1，其他版本可能有细微不同。更多详情请访问[文档](#)。

macOS

Docker For Mac

右键点击桌面顶栏的 docker 图标，选择 Preferences，在 Daemon 标签（Docker 17.03 之前版本为 Advanced 标签）下的 Registry mirrors 列表中加入下面的镜像地址：

```
http://f1361db2.m.daocloud.io
```

点击 Apply & Restart 按钮使设置生效。

Docker Toolbox 等配置方法请参考[帮助文档](#)。

Windows

Docker For Windows

在桌面右下角状态栏中右键 docker 图标，修改在 Docker Daemon 标签页中的 json，把下面的地址：

```
http://f1361db2.m.daocloud.io
```

加到 "registry-mirrors" 的数组里。点击 Apply。

公众号：黑猫编程

网址：<https://noi.hioier.co>

```
1 curl -sSL https://get.daocloud.io/daotools/set_mirror.sh | sh -s  
http://f1361db2.m.daocloud.io
```

```
root@pc:~# curl -sSL https://get.daocloud.io/daotools/set_mirror.sh | sh -s http://f1361db2.m.daocloud.io  
docker version >= 1.12  
{"registry-mirrors": ["http://f1361db2.m.daocloud.io"]}  
Success.  
You need to restart docker to take effect: sudo systemctl restart docker.service
```

查看国内源:

```
root@pc:~# cat /etc/docker/daemon.json  
{"registry-mirrors": ["http://f1361db2.m.daocloud.io"]}
```

Docker镜像

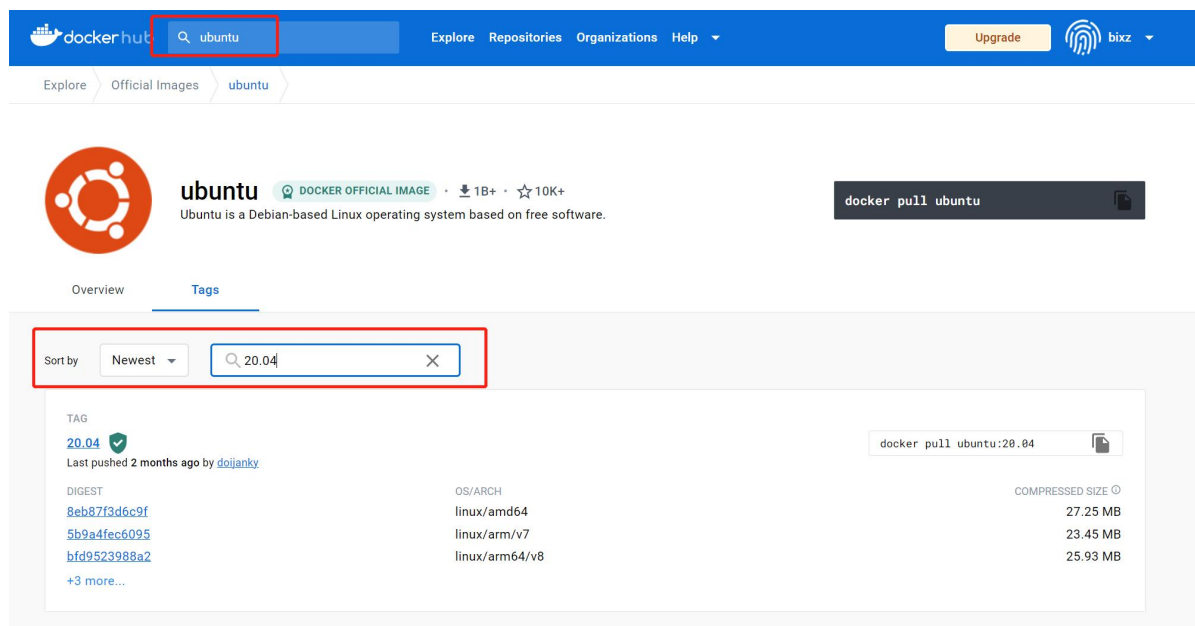
Docker镜像是一个只读的文件，就类似于我们安装操作系统时候所需要的那个iso光盘镜像，通过运行这个镜像来完成各种应用的部署。

通过 `docker search ubuntu` 查找镜像:

```
root@pc:~# docker search ubuntu
```

NAME	DESCRIPTION	STARS
ubuntu	Ubuntu is a Debian-based Linux operating sys...	15494
websphere-liberty	WebSphere Liberty multi-architecture images ...	291
ubuntu-upstart	DEPRECATED, as is Upstart (find other proces...	112
neurodebian	NeuroDebian provides neuroscience research s...	98
ubuntu/nginx	Nginx, a high-performance reverse proxy & we...	75
open-liberty	Open Liberty multi-architecture images based...	56
ubuntu/apache2	Apache, a secure & extensible open-source HT...	53
ubuntu-debootstrap	DEPRECATED; use "ubuntu" instead	50
ubuntu/squid	Squid is a caching proxy for the Web. Long-t...	49
ubuntu/bind9	BIND 9 is a very flexible, full-featured DNS...	44
ubuntu/mysql	MySQL open source fast, stable, multi-thread...	41
ubuntu/prometheus	Prometheus is a systems and service monitori...	35

如果获取镜像具体版本可以官网上搜索Ubuntu，点击Tags，在后面搜索版本号:



The screenshot shows the Docker Hub interface for the 'ubuntu' image. The 'Tags' tab is selected, and a search bar is highlighted with a red box. The search results show the '20.04' tag, which is the latest version. The page also displays the Ubuntu logo, the 'DOCKER OFFICIAL IMAGE' badge, and a list of tags with their respective OS/ARCH and compressed sizes.

- 1 `docker pull ubuntu:20.04` # 拉取镜像
- 2 `docker images` # 列出本地所有镜像

公众号：黑猫编程
网址：https://noi.hioier.co


```

root@pc:~# docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
846c0b181fff: Pull complete
Digest: sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
root@pc:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
ubuntu        20.04      d5447fc01ae6  6 weeks ago  72.8MB
root@pc:~#

```

删除镜像:

```

1 docker image rm nginx
2 docker rmi django # 简写形式

```

```

root@pc:~# docker image rm nginx
Untagged: nginx:latest
Untagged: nginx@sha256:b8f2383a95879e1ae064940d9a200f67a6c79e710ed82ac42263397367e7cc4e
Deleted: sha256:a99a39d070bfd1cb60fe65c45dea3a33764dc00a9546bf8dc46cb5a11b1b50e9
Deleted: sha256:937f740376014d8e951ae4f0ff0b0ca64fd958c176510fc0a86dd1b491226a71
Deleted: sha256:a73369e624d6b4fbae1e6046a2f3cf3c7a15c6e0e0012659f410380d8dc70594
Deleted: sha256:bc2dd4cf50f065c3e884872cd30eb888a71f6ab545d8bec196a7662bfd033a3
Deleted: sha256:c9372c5c9249380700f7af0441de14f14c361b5561ecc876da46f58a7ad71b7b
Deleted: sha256:deea3af05bc845b217a415b240422c19933218422b6d60414c2caeff976e5430
Deleted: sha256:67a4178b7d47beb6a1f697a593bd0c6841c67eb0da00f2badeffb05fd30671490

```

```

root@pc:~# docker rmi django
Untagged: django:latest
Untagged: django@sha256:5bfd3f442952463f5bc97188b7f43cfcd6c2f631a017ee2a6fca3cb8992501e8
Deleted: sha256:eb40dcf64078249a33f68fdd8d80624cb81b524c24f50b95fff5c2b40bdc3fde
Deleted: sha256:16e6b55f22cca357d6d1c156ffde240ebd570d707409b959bdd5791ebe8d6968
Deleted: sha256:9c313ef12b5130c2a4d1d8ed5672b1cdf975851ad91c4893b23b809f1029096c
Deleted: sha256:776c17f61bc2b89f37cb5acc7d28f3ba9b19f5b44dee0d8d10c497aa46b99d9b
Deleted: sha256:81c42ad66fd6acd4f23fac6487913a931b741c3bb45de49be2d593ec40a1e63c
Deleted: sha256:53694fb1f2bf8ecef0e91bed100e5dbd5e832b33c29396fa6f0a6246f09bbfde
Deleted: sha256:b6ca02dfe5e62c58dacb1dec16eb42ed35761c15562485f9da9364bb7c90b9b3

```

镜像资源改名操作: `docker tag ubuntu:20.04 myubuntu:1.0`

该操作会保留原有镜像, 不需要可以自行删除。

```

root@pc:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
ubuntu        20.04      d5447fc01ae6  6 weeks ago  72.8MB
root@pc:~# docker tag ubuntu:20.04 myubuntu:1.0
root@pc:~# docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
myubuntu      1.0        d5447fc01ae6  6 weeks ago  72.8MB
ubuntu        20.04      d5447fc01ae6  6 weeks ago  72.8MB

```

压缩镜像, 以后用到直接解压: `docker save -o ubuntu.tar ubuntu`

```

root@pc:~# docker save -o ubuntu.tar ubuntu
root@pc:~# ls
ubuntu.tar

```

导入ubuntu.tar压缩包: `docker load -i ubuntu.tar`

```

root@pc:~# docker rmi ubuntu:20.04
Untagged: ubuntu:20.04
Untagged: ubuntu@sha256:0e0402cd13f68137edb0266e1d2c682f217814420f2d43d300ed8f65479b14fb
root@pc:~#
root@pc:~# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
myubuntu      1.0       d5447fc01ae6  6 weeks ago   72.8MB
root@pc:~# docker load -i ubuntu.tar
Loaded image: ubuntu:20.04
root@pc:~# docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
myubuntu      1.0       d5447fc01ae6  6 weeks ago   72.8MB
ubuntu        20.04     d5447fc01ae6  6 weeks ago   72.8MB

```

查看指定镜像历史: `docker image history ubuntu:20.04`

```

root@pc:~# docker image history ubuntu:20.04
IMAGE          CREATED          CREATED BY                                      SIZE      COMMENT
d5447fc01ae6   6 weeks ago     /bin/sh -c #(nop) CMD ["bash"]                0B
<missing>      6 weeks ago     /bin/sh -c #(nop) ADD file:9d282119af0c42bc8... 72.8MB

```

容器命令

利用ubuntu20.04镜像创建一个名为myrq的容器，并进入容器，执行/bin/bash:

```
1 | docker run -it --name myrq ubuntu:20.04
```

```

root@pc:~# docker run -it --name myrq ubuntu:20.04
root@3cb40093af53:/#

```

```

root@pc:~# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS          NAMES
d7b3cad9f52a   ubuntu:20.04  "/bin/bash"             About a minute ago  Up About a minute              myrq

```

exit退出容器:

利用镜像ubuntu:20.04创建一个容器: `docker create -it ubuntu:20.04`

启动容器: `docker start 容器`

停止容器: `docker stop 容器`

重启容器: `docker restart 容器`

列出本地所有容器: `docker ps -a`

```

root@cat:~# docker create -it ubuntu:20.04
35eb26784de467b76364fee9afdc4bf7f81119a732c403afb744e1a31bce642c
root@cat:~# docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS          NAMES
35eb26784de4   ubuntu:20.04  "bash"                 9 seconds ago    Created                               silly_aryabhata
cf90c356015d   ubuntu:20.04  "bash"                 21 seconds ago   Created                               charming_proskuriako
va
root@cat:~# docker start 35eb26784de4
35eb26784de4

```

创建并启动一个容器: `docker run -itd ubuntu:20.04`

进入容器: `docker attach 容器`

挂起容器: 先按Ctrl-p, 再按Ctrl-q


```

root@cat:~# docker run -itd ubuntu:20.04
2a51fd27448b7426cbf4baccf76b5c15da5f579526bee81234bd052c0442f5b5
root@cat:~# docker ps
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS
2a51fd27448b   ubuntu:20.04   "bash"        About a minute ago    Up About a minute
ess_wilbur
35eb26784de4   ubuntu:20.04   "bash"        10 minutes ago      Up 5 minutes
aryabhata
root@cat:~# docker attach 35eb26784de4
root@35eb26784de4:/# read escape sequence

```

在容器中执行命令： `docker exec 容器 命令`

```

root@blackcat:/home/cat/docker# docker start 857c7a204a2f
857c7a204a2f
root@blackcat:/home/cat/docker# docker exec 857c7a204a2f pwd
/

```

删除容器（先关闭容器）： `docker rm 容器`

将容器CONTAINER导出到本地文件xxx.tar中： `docker export -o xxx.tar 容器`

```

root@blackcat:/home/cat/docker# docker ps -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS        PORTS        NAMES
07a2530ce393   ubuntu:20.04   "bash"        About an hour ago    Created
root@blackcat:/home/cat/docker# docker export -o tmp.tar 07a2530ce393
root@blackcat:/home/cat/docker# ls
tmp.tar  ubuntu_20_04.tar
root@blackcat:/home/cat/docker# chmod +r tmp.tar
root@blackcat:/home/cat/docker# su cat
cat@blackcat:~/docker$ scp tmp.tar rootserver:
tmp.tar                                           100%   72MB   11.1MB/s   00:06
cat@blackcat:~/docker$ ssh rootserver
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-86-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Sun Nov 14 11:03:07 2021 from 58.37.50.235
root@cat:~# ls
mysite  others  tmp.tar  ubuntu_20_04.tar

```

将本地文件xxx.tar导入成镜像，并将镜像命名为image_name:tag： `docker import xxx.tar image_name:tag`

```

root@cat:~# docker import tmp.tar ubuntu:my
sha256:eec496bd9a94b9f9be910557f5a2fa9eb62f71557ef0e3354b1d0b0d8e30dae3c
root@cat:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        my        eec496bd9a94   4 seconds ago  72.8MB

```

通过容器生成的镜像创建容器： `run -it --name myrq4 ubuntu:1.0 **/bin/bash**`

查看所有容器的统计信息，包括CPU、内存、存储、网络等信息： `docker stats`

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
2a51fd27448b	priceless_wilbur	0.00%	1.191MiB / 1.862GiB	0.06%	1.22kB / 0B	0B / 0B	1
35eb26784de4	silly_aryabhata	0.00%	1.434MiB / 1.862GiB	0.08%	1.74kB / 0B	352kB / 0B	1

在本地和容器间复制文件： `docker cp xxx CONTAINER:xxx` 或 `docker cp CONTAINER:xxx xxx`

重命名容器： `docker rename CONTAINER1 CONTAINER2`


```

root@cat:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS        PORTS        NAMES
2a51fd27448b   ubuntu:20.04   "bash"        2 hours ago   Up 2 hours                priceless_wilbur
35eb26784de4   ubuntu:20.04   "bash"        2 hours ago   Up 2 hours                silly_aryabhata
cf90c356015d   ubuntu:20.04   "bash"        2 hours ago   Exited (0) 2 hours ago    charming_proskuriakova
root@cat:~# docker rename priceless_wilbur docker_newname
root@cat:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS        PORTS        NAMES
2a51fd27448b   ubuntu:20.04   "bash"        2 hours ago   Up 2 hours                docker_newname
35eb26784de4   ubuntu:20.04   "bash"        2 hours ago   Up 2 hours                silly_aryabhata
cf90c356015d   ubuntu:20.04   "bash"        2 hours ago   Exited (0) 2 hours ago    charming_proskuriakova
root@cat:~#

```

删除所有已停止容器: `docker container prune`

批量删除所有容器, 先用命令列出所有容器id并赋值给变量a:

```

root@pc:~# docker ps -a
CONTAINER ID   IMAGE          COMMAND        CREATED        STATUS        PORTS        NAMES
67c8bcf4e84c   ubuntu:20.04   "bash"        13 seconds ago   Exited (0) 10 seconds ago
3cb40093af53   ubuntu:20.04   "bash"        5 minutes ago   Exited (127) 28 seconds ago    myrq2
root@pc:~#
root@pc:~# docker ps -a -q
67c8bcf4e84c
3cb40093af53
root@pc:~# a=$(docker ps -a -q)
root@pc:~# echo $a
67c8bcf4e84c 3cb40093af53

```

强制删除所有容器:

```

root@pc:~# docker rm -f $a
67c8bcf4e84c
3cb40093af53

```

```

1 # run.sh
2 a=$(docker ps -a -q)
3 docker rm -f $a

```