

# Docker资料

官网: <https://www.docker.com/>

官方文档: <https://docs.docker.com/>

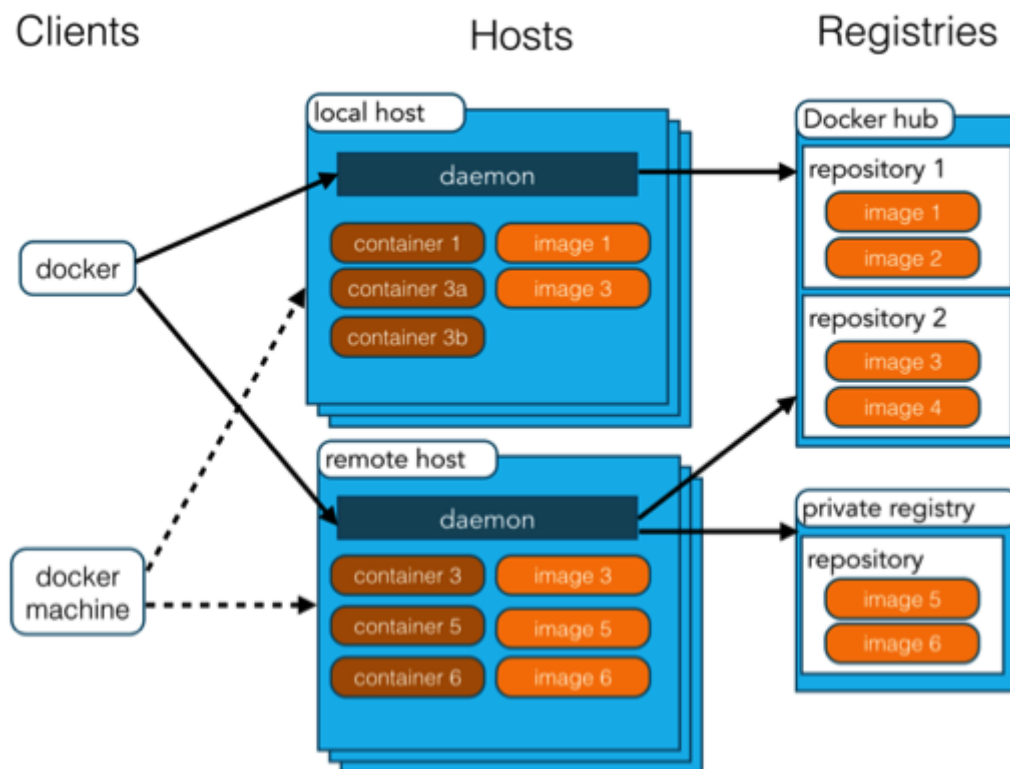
官方Hub: <https://registry.hub.docker.com/>

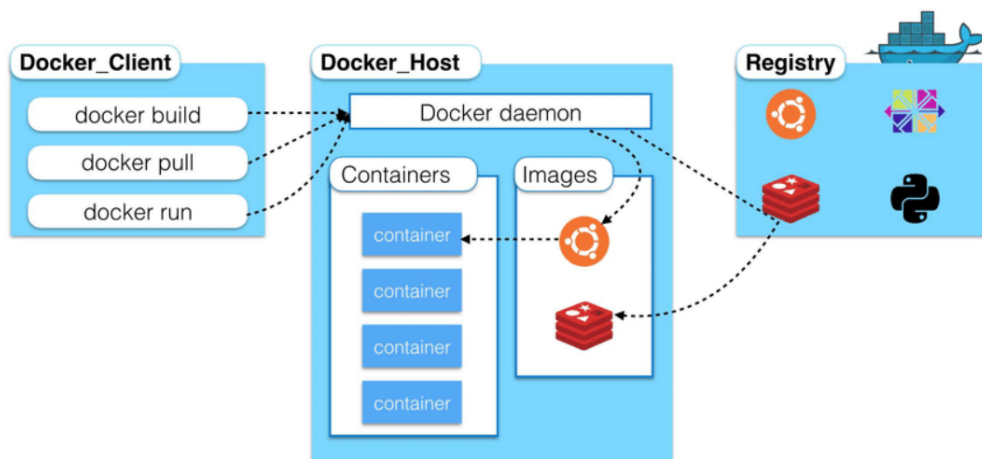
# Docker概念

镜像image=模板=类class

容器container=系统=对象object

仓库repository存放镜像的地方





# 安装Docker

## 配置Linux

关闭防火墙，关闭防火墙开机自启

```
systemctl stop firewalld
```

```
systemctl disable firewalld.service
```

修改静态IP

```
vim /etc/sysconfig/network-scripts/ifcfg-ens33
```

## Linux版本

```
[user@localhost ~]$ uname -r  
3.10.0-1160.el7.x86_64
```

```
[user@localhost ~]$ cat /etc/os-release  
NAME="CentOS Linux"  
VERSION="7 (Core)"  
ID="centos"  
ID_LIKE="rhel fedora"  
VERSION_ID="7"  
PRETTY_NAME="CentOS Linux 7 (Core)"  
ANSI_COLOR="0;31"  
CPE_NAME="cpe:/o:centos:centos:7"  
HOME_URL="https://www.centos.org/"  
BUG_REPORT_URL="https://bugs.centos.org/"  
  
CENTOS_MANTISBT_PROJECT="CentOS-7"  
CENTOS_MANTISBT_PROJECT_VERSION="7"
```

```
REDHAT_SUPPORT_PRODUCT="centos"  
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

## 安装Docker

### 卸载旧版本

```
yum remove docker \  
    docker-client \  
    docker-client-latest \  
    docker-common \  
    docker-latest \  
    docker-latest-logrotate \  
    docker-logrotate \  
    docker-engine
```

### 下载依赖安装包

```
yum install -y yum-utils
```

### 配置镜像仓库

```
#国外的地址  
yum-config-manager \  
    --add-repo \  
    https://download.docker.com/linux/centos/docker-ce.repo
```

#设置阿里云的Docker镜像仓库

```
yum-config-manager \  
    --add-repo \  
    https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

```
[root@iz1608aqb7ntn9Z /]# yum-config-manager \  
    --add-repo \  
    https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo  
添加仓库自: https://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.repo
```

### 更新yum软件包

```
yum makecache fast
```

### 下载docker

```
yum install docker-ce docker-ce-cli containerd.io # 安装社区版
yum install docker-ee docker-ee-cli containerd.io # 安装企业版
```

一般情况下安装 [社区版](#)

## 启动Docker

```
systemctl start docker # 启动Docker
docker version # 查看当前版本号，是否启动成功
systemctl enable docker # 设置开机自启动
#Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service
to /usr/lib/systemd/system/docker.service.
```

## Docker的HelloWorld

```
docker run hello-world
```

## Docker卸载

```
yum remove docker-ce docker-ce-cli containerd.io # 卸载依赖
rm -rf /var/lib/docker # 删除资源 . /var/lib/docker是docker的默认工作路径
```

## 配置阿里云镜像加速

### 进入阿里云官网，搜索[容器镜像服务](#)

阿里云 最新活动 产品 解决方案 云市场 合作伙伴 支持与服务 开发者 了解阿里云

容器镜像服务 ACR

阿里云容器镜像服务（简称 ACR）是面向容器镜像、Helm Chart 等符合 OCI 标准的云原生制品安全托管及高效分发平台。ACR 支持全球同步加速、大规模大镜像分发加速、多代码源构建加速等全链路提效，与容器服务 ACK 无缝集成，帮助企业降低交付复杂度，打造云原生应用一站式解决方案。

[立即购买](#) [管理控制台](#) [产品价格](#) [帮助与文档](#)

产品规格

ACR EE 新客包年享受 8.5 折，包月享 9.5 折！

ACR EE - 基础版	ACR EE - 标准版	ACR EE - 高级版
企业级平台提供 OCI 镜像的安全托管和高效分发。	企业级平台提供 OCI 镜像的安全托管和高效分发。	企业级平台提供 OCI 镜像的安全托管和高效分发。

开发者不要钱！点击[管理控制台](#)即可。

容器镜像服务

实例列表

镜像中心

镜像工具

镜像加速器

使用加速器可以提升获取Docker官方镜像的速度

加速器

加速器地址

<https://4b3b3pix.mirror.aliyuncs.com> 复制

操作文档

UbuntuCentOSMacWindows

1. 安装 / 升级Docker客户端

推荐安装 1.10.0 以上版本的Docker客户端，参考文档[docker-ce](#)

2. 配置镜像加速器

针对Docker客户端版本大于 1.10.0 的用户

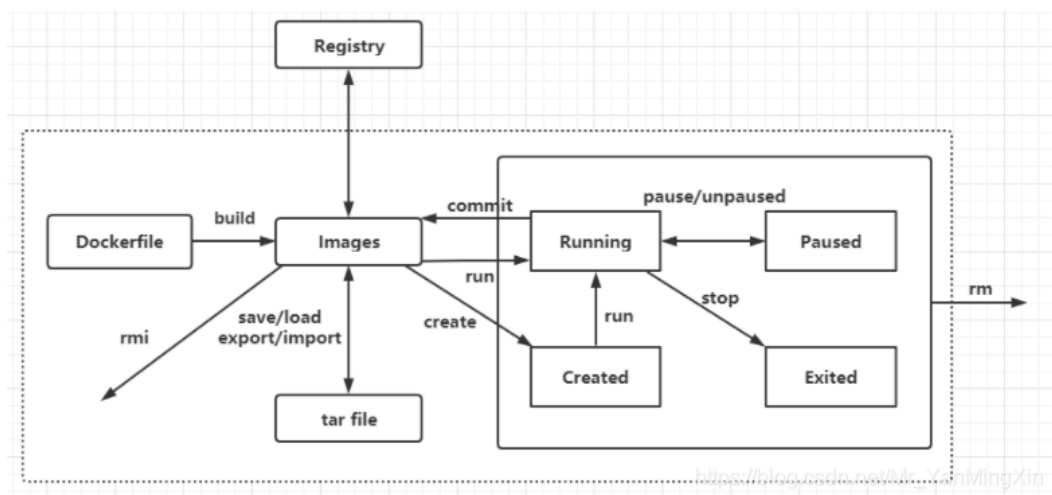
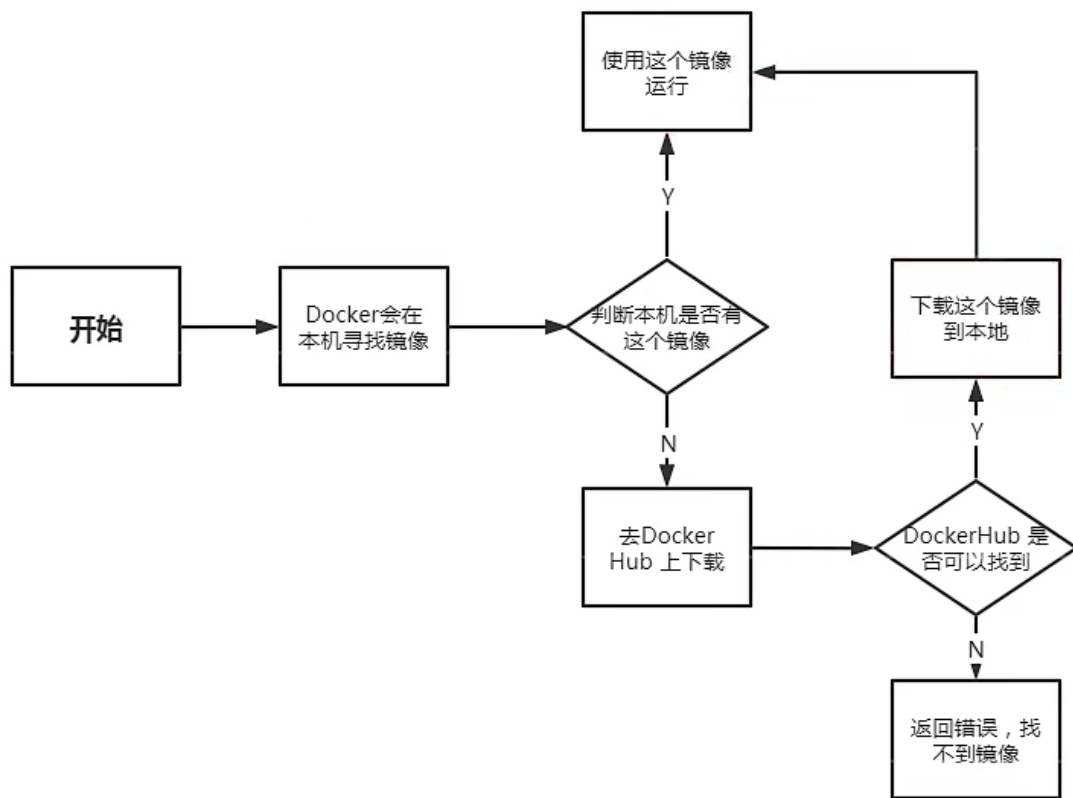
您可以通过修改daemon配置文件 `/etc/docker/daemon.json` 来使用加速器

```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://4b3b3pix.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

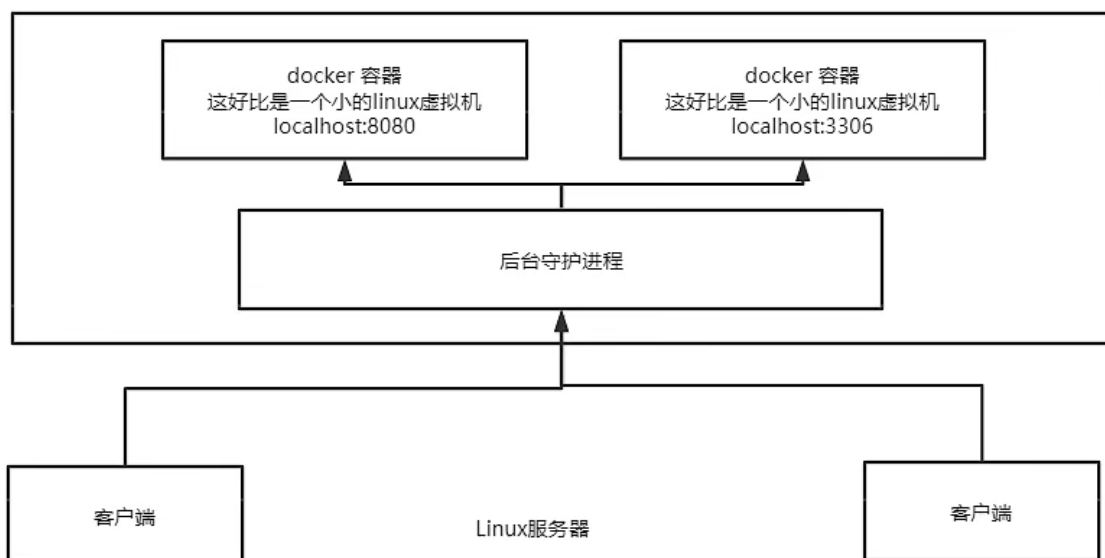
```
sudo mkdir -p /etc/docker
sudo tee /etc/docker/daemon.json <<-'EOF'
{
  "registry-mirrors": ["https://4b3b3pix.mirror.aliyuncs.com"]
}
EOF
sudo systemctl daemon-reload
sudo systemctl restart docker
```

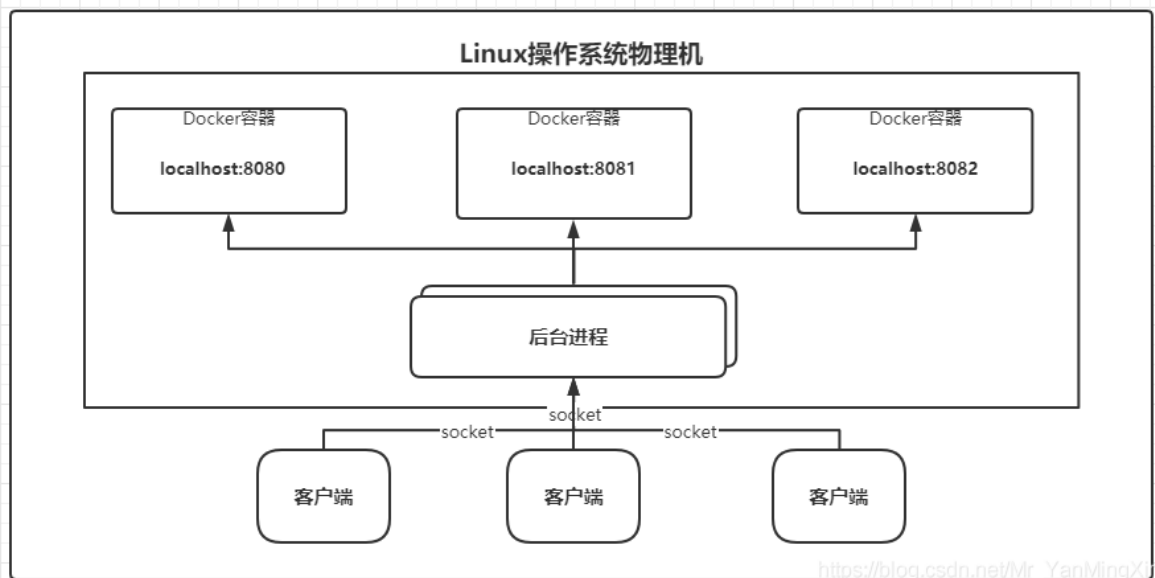
## 启动Docker

### run的运行流程图

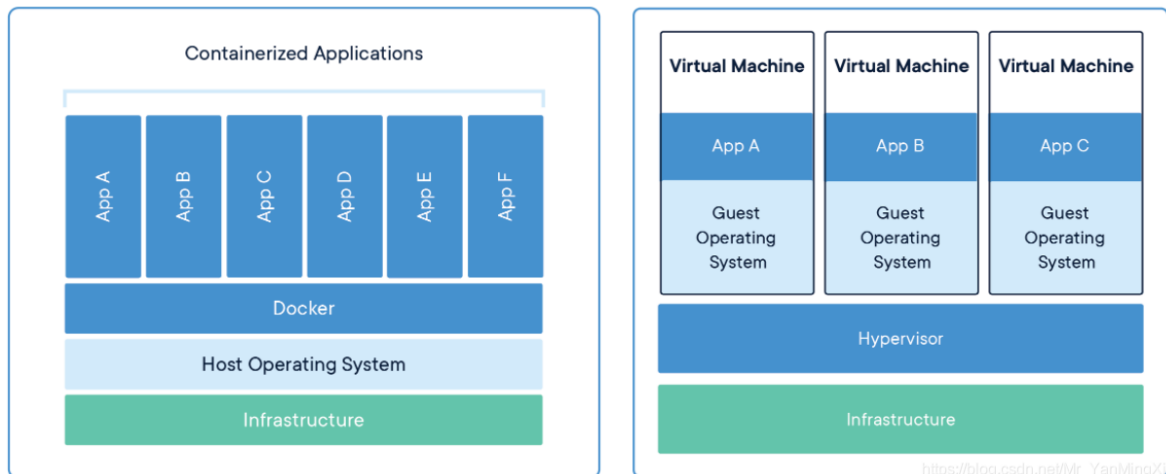


## 运行原理





## docker整体架构



## Docker常用命令

命令的帮助文档地址:<https://docs.docker.com/engine/reference/commandline/docker/>

### 基本命令

```
docker version      #查看docker的版本信息
docker info         #查看docker的系统信息,包括镜像和容器的数量
docker 命令 --help  #帮助命令(可查看可选的参数)
docker COMMAND --help
```

### 镜像命令

#### 查看本地主机的所有镜像

```
[root@localhost user]# docker images
REPOSITORY    TAG        IMAGE ID    CREATED    SIZE
```

```
# 解释:
1.REPOSITORY 镜像的仓库源
2.TAG 镜像的标签
3.IMAGE ID 镜像的id
4.CREATED 镜像的创建时间
5.SIZE 镜像的大小
# 可选参数
-a/--all 列出所有镜像
-q/--quiet 只显示镜像的id
```

## 搜索镜像

```
[root@localhost user]# docker search xxx
```

```
#可选参数
Search the Docker Hub for images
Options:
  -f, --filter filter      Filter output based on conditions provided
  --format string          Pretty-print search using a Go template
  --limit int              Max number of search results (default 25)
  --no-trunc               Don't truncate output

#搜索收藏数大于3000的镜像
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker search mysql --filter=STARS=3000
NAME                DESCRIPTION                                STARS     OFFICIAL
AUTOMATED
mysql               MySQL is a widely used, open-source relation... 10308     [OK]
mariadb             MariaDB is a community-developed fordockerk of Mys... 3819      [OK]
```

## docker pull 镜像名[:tag] 下载镜像

```
[root@iz1608aqb7ntn9Z /]# docker pull mysql:5.7
5.7: Pulling from library/mysql
33847f680f63: Pull complete
5cb67864e624: Pull complete
1a2b594783f5: Pull complete
b30e406dd925: Pull complete
48901e306e4c: Pull complete
603d2b7147fd: Pull complete
802aa684c1c4: Pull complete
5b5a19178915: Pull complete
f9ce7411c6e4: Pull complete
f51f6977d9b2: Pull complete
aeb6b16ce012: Pull complete
Digest: sha256:be70d18aedc37927293e7947c8de41ae6490ecd4c79df1db40d1b5b5af7d9596
Status: Downloaded newer image for mysql:5.7
docker.io/library/mysql:5.7
```

## 删除镜像



### #1. 删除指定的镜像id

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker rmi -f 镜像id
```

### #2. 删除多个镜像id

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker rmi -f 镜像id 镜像id 镜像id
```

### #3. 删除全部的镜像id

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker rmi -f $(docker images -aq)
```

## 容器基本命令

有了镜像才可以创建容器，linux，下载一个centos镜像测试学习

```
docker pull centos
```

## docker run [可选参数] image 运行容器

```
docker run [可选参数] image
```

### #参数说明

<b>--name="名字"</b>	指定容器名字
<b>-d</b>	后台方式运行（没有提供服务会被自动停止）
<b>-it</b>	使用交互方式运行,进入容器查看内容
<b>-p</b>	指定容器的端口
( <b>-p ip:主机端口:容器端口</b>	配置主机端口映射到容器端口
<b>-p 主机端口:容器端口</b>	
<b>-p 容器端口)</b>	
<b>-P</b>	随机指定端口(大写的P)

## 进入容器

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker run -it [容器ID] /bin/bash
进入linux容器
[root@localhost user]# docker run -it centos /bin/bash
[root@b838788eb56c /]#
```

## 退出容器

```
exit #停止容器并退出
ctrl + P + Q #不停止容器并退出
```

## docker ps列出容器

```
#docker ps
# 列出当前正在运行的容器
-a # 列出所有容器的运行记录
-n=? # 显示最近创建的n个容器
-q # 只显示容器的编号
```

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker ps -a						
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
bca129320bb5	centos	"/bin/bash"	4 minutes ago	Exited (0)		3 minutes ago
bd1b8900c547	centos	"/bin/bash"	6 minutes ago	Exited (0)		5 minutes ago
cf6adb1b506	bf756fb1ae65	"/hello"	5 hours ago	Exited (0)		5 hours ago

## 删除容器

```
docker rm 容器id #删除指定的容器,不能删除正在运行的容器,强制删除使用 rm -f
docker rm -f $(docker ps -aq) #删除所有的容器
docker ps -a -q|xargs docker rm #删除所有的容器
```

## 启动和重启容器命令

```
docker start 容器id #启动容器
docker restart 容器id #重启容器
docker stop 容器id #停止当前运行的容器
docker kill 容器id #强制停止当前容器
```

```
docker run -it --rm xxx #阅后即焚
docker pull xxx #下载再启动
docker run -d -p 3355:8080 --name xxx别名 xxx #启动运行
```

## 其他命令

### 查看日志

```
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker logs --help
```

Usage: docker logs [OPTIONS] CONTAINER

Fetch the logs of a container

Options:

- details Show extra details provided to logs
- f, --follow Follow log output
- since string Show logs since timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)
- n, --tail string Number of lines to show from the end of the logs (default "all")
- t, --timestamps Show timestamps
- until string Show logs before a timestamp (e.g. 2013-01-02T13:23:37Z) or relative (e.g. 42m for 42 minutes)

常用:

```
docker logs -tf 容器id
```

```
docker logs --tail number 容器id #num为要显示的日志条数
```

#docker容器后台运行，必须要有一个前台的进程，否则会自动停止

#编写shell脚本循环执行，使得centos容器保持运行状态

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker run -d centos /bin/sh -c "while true;do echo hi;sleep 5;done"
```

```
c703b5b1911ff84d584390263a35707b6024816e1f46542b61918a6327a570dc
```

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
c703b5b1911f	centos	"/bin/sh -c 'while t..."	13 seconds ago	Up 10 seconds
	pedantic_banach			

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker logs -tf --tail 10 c703b5b1911f
```

```
2020-12-27T03:34:07.255599560Z hi
```

```
2020-12-27T03:34:12.257641517Z hi
```

```
2020-12-27T03:34:17.259706294Z hi
```

```
2020-12-27T03:34:22.261693707Z hi
```

```
2020-12-27T03:34:27.262609289Z hi
```

```
2020-12-27T03:34:32.267862677Z hi
```

```
2020-12-27T03:34:37.270382873Z hi
```

```
2020-12-27T03:34:42.272414182Z hi
```

```
2020-12-27T03:34:47.274823243Z hi
```

```
2020-12-27T03:34:52.277419274Z hi
```

## 查看容器中进程信息

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker top c703b5b1911f
```

UID	PID	PPID	C
STIME	TTY	TIME	CMD
root	11156	11135	0
11:31	?	00:00:00	/bin/sh -c while true;do echo hi;sleep 5;done
root	11886	11156	0
11:43	?	00:00:00	/usr/bin/coreutils -coreutils-prog-shebang=sleep /usr/bin/sleep 5

## 查看容器的元数据

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker inspect 容器id
```

## 进入当前正在运行的容器

方法一

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker exec -it c703b5b1911f /bin/bash
[root@c703b5b1911f /]# ls
bin dev etc home lib lib64 lost+found media mnt opt proc root run
sbin srv sys tmp usr var
[root@c703b5b1911f /]# ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0  03:31 ?           00:00:00 /bin/sh -c while true;do echo
hi;sleep 5;done
root          279     0  0  03:54 pts/0       00:00:00 /bin/bash
root          315     1  0  03:56 ?           00:00:00 /usr/bin/coreutils --coreutils-
prog-shebang=sleep /usr/bin/sleep 5
root          316    279  0  03:56 pts/0       00:00:00 ps -ef
```

## 方法二

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker attach c703b5b1911f
```

docker exec 进入容器后开启一个新的终端，可以在里面操作

docker attach 进入容器正在执行的终端，不会启动新的进程

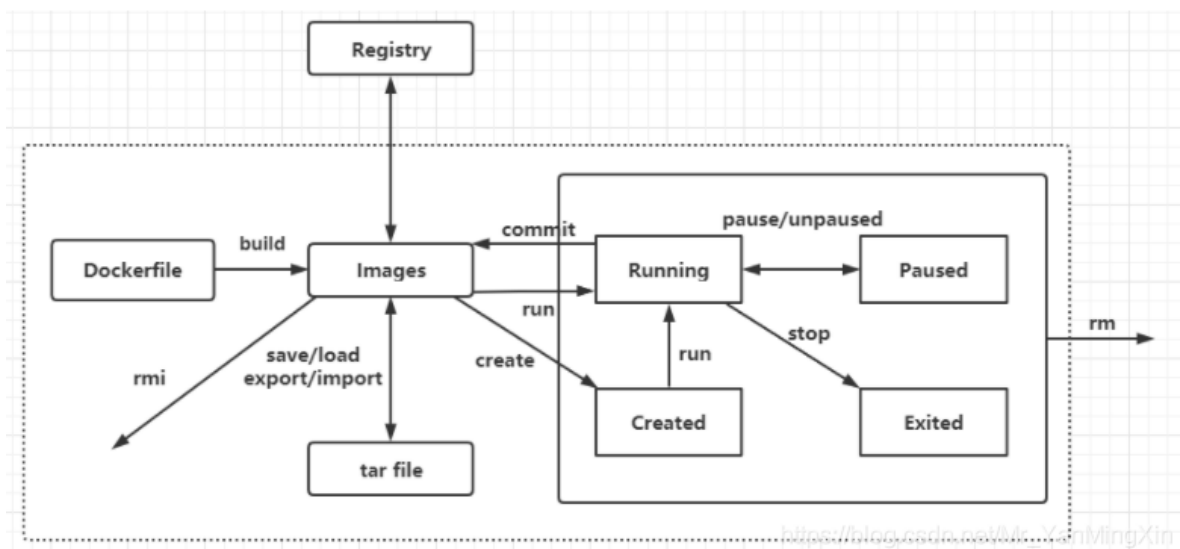
## 拷贝容器文件到主机

拷贝容器的文件到主机中

```
docker cp 容器id:容器内路径 目的主机路径
```

```
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker exec -it c703b5b1911f /bin/bash
[root@c703b5b1911f /]# cd home
[root@c703b5b1911f home]# ls
#touch 新建文件
[root@c703b5b1911f home]# touch test.java
[root@c703b5b1911f home]# ls
test.java
[root@c703b5b1911f home]# exit
exit
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
c703b5b1911f   centos    "/bin/sh -c 'while t..." 35 minutes ago Up 35 minutes
pedantic_banach
[root@iZwz99sm8v95sckz8bd2c4Z ~]# docker cp c703b5b1911f:/home/test.java /home
[root@iZwz99sm8v95sckz8bd2c4Z ~]# ls /home
hai pan test.java
```

## 常用命令小结



## Docker图形化管理工具

### Docker UI

```
docker search dockerui
docker pull abhinav/dockerui
docker run -d --privileged --name dockerui -p 9000:9000 -v
/var/run/docker.sock:/var/run/docker.sock abhinav/dockerui
#放开物理机的9000端口对应docker容器的9000端口
```

使用

访问: <http://ip地址:9000>

### Shipyards

### Portainer

```
docker search portainer
docker pull portainer/portainer
docker run -d --name portainerUI -p 9000:9000 -v
/var/run/docker.sock:/var/run/docker.sock portainer/portainer
```

使用

访问: <http://ip地址:9000>

## 常见容器部署—Nginx,Tomcat,ES

### Nginx

```
[root@iz1608aqb7ntn9Z ~]# docker search nginx # 查找
[root@iz1608aqb7ntn9Z ~]# docker pull nginx # 下载
[root@iz1608aqb7ntn9Z ~]# docker run -d --name nginx -p 9000:80 nginx # 启动

# 备注
-d 后台运行
--name 给容器命名
-p 3334:80 将宿主机的端口3334映射到该容器的80端口
```

## 测试



## Tomcat

```
[root@iz1608aqb7ntn9Z ~]# docker search tomcat # 查找
[root@iz1608aqb7ntn9Z ~]# docker pull tomcat # 下载
[root@iz1608aqb7ntn9Z ~]# docker run -d --name tomcat -p 9000:8080 tomcat # 启动
```



## ES

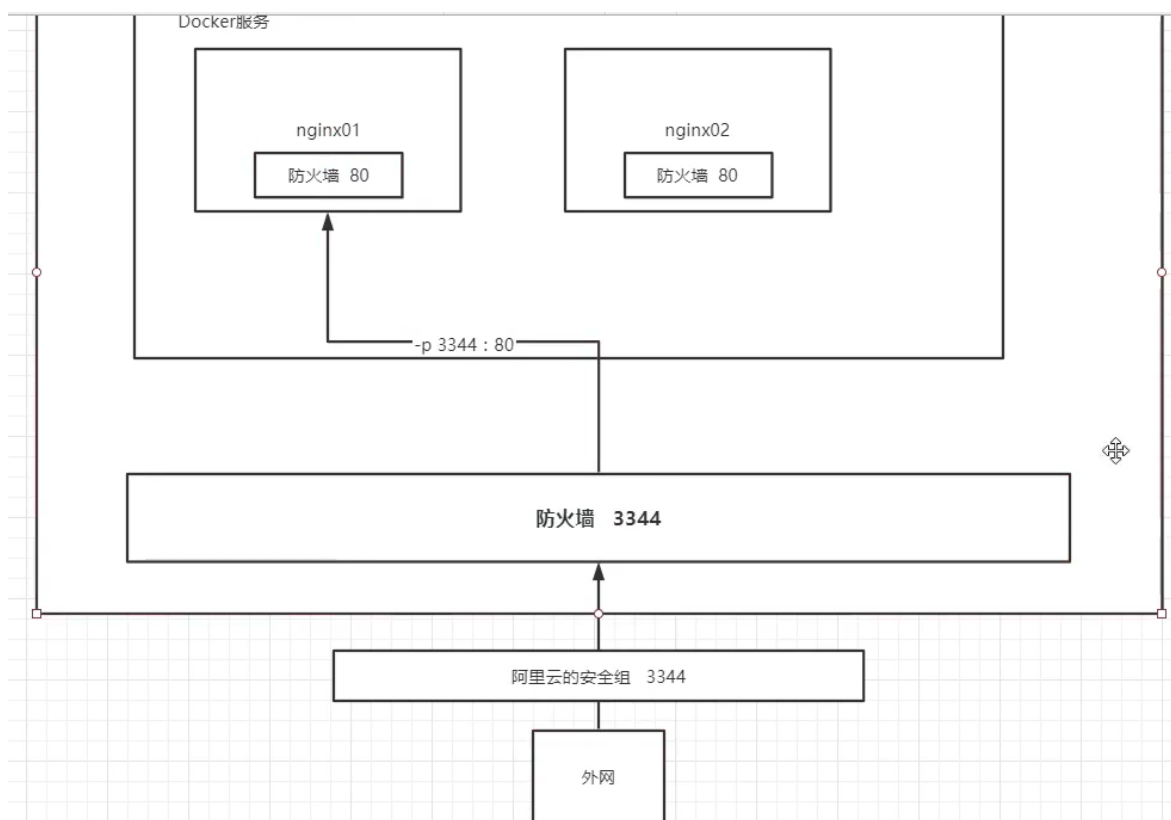
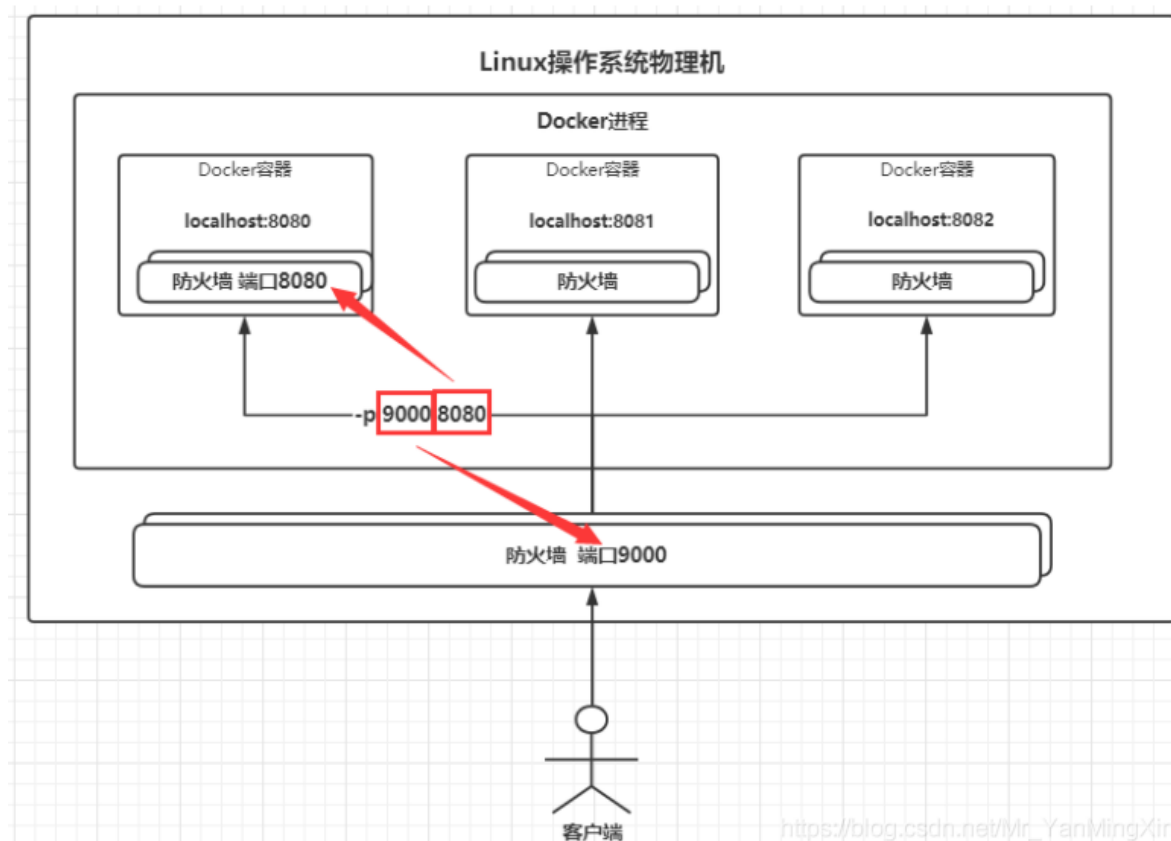
添加 '-e ES\_JAVA\_OPTS="-Xms128m -Xmx512m"' 配置ElasticSearch的虚拟机占用的内存大小。

docker stats 查看资源占用情况

```
$ docker run -d --name elasticsearch -p 9200:9200 -p 9300:9300 -e
"discovery.type=single-node" -e ES_JAVA_OPTS="-Xms128m -Xmx512m"
elasticsearch:7.6.2

[root@izwz99sm8v95sckz8bd2c4Z ~]# docker run -d --name elasticsearch01 -p
9200:9200 -p 9300:9300 -e "discovery.type=single-node" -e ES_JAVA_OPTS="-Xms128m
-Xmx512m" elasticsearch:7.6.2
3b8cd4991814896c523ee67b84ce198e32bd82b1a62d512b198138a58ca946f1
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
3b8cd4991814      elasticsearch:7.6.2  "/usr/local/bin/dock...  10 seconds ago
Up 6 seconds      0.0.0.0:9200->9200/tcp, 0.0.0.0:9300->9300/tcp  elasticsearch01
[root@izwz99sm8v95sckz8bd2c4Z ~]# docker stats
```

## 端口暴露的原理



联合文件系统

分层管理

commit镜像

## 容器数据卷

# DockerFile

---

## Docker网络

---

## 企业实战

---

[https://www.bilibili.com/video/BV1og4y1q7M4?p=21&spm\\_id\\_from=pageDriver](https://www.bilibili.com/video/BV1og4y1q7M4?p=21&spm_id_from=pageDriver)