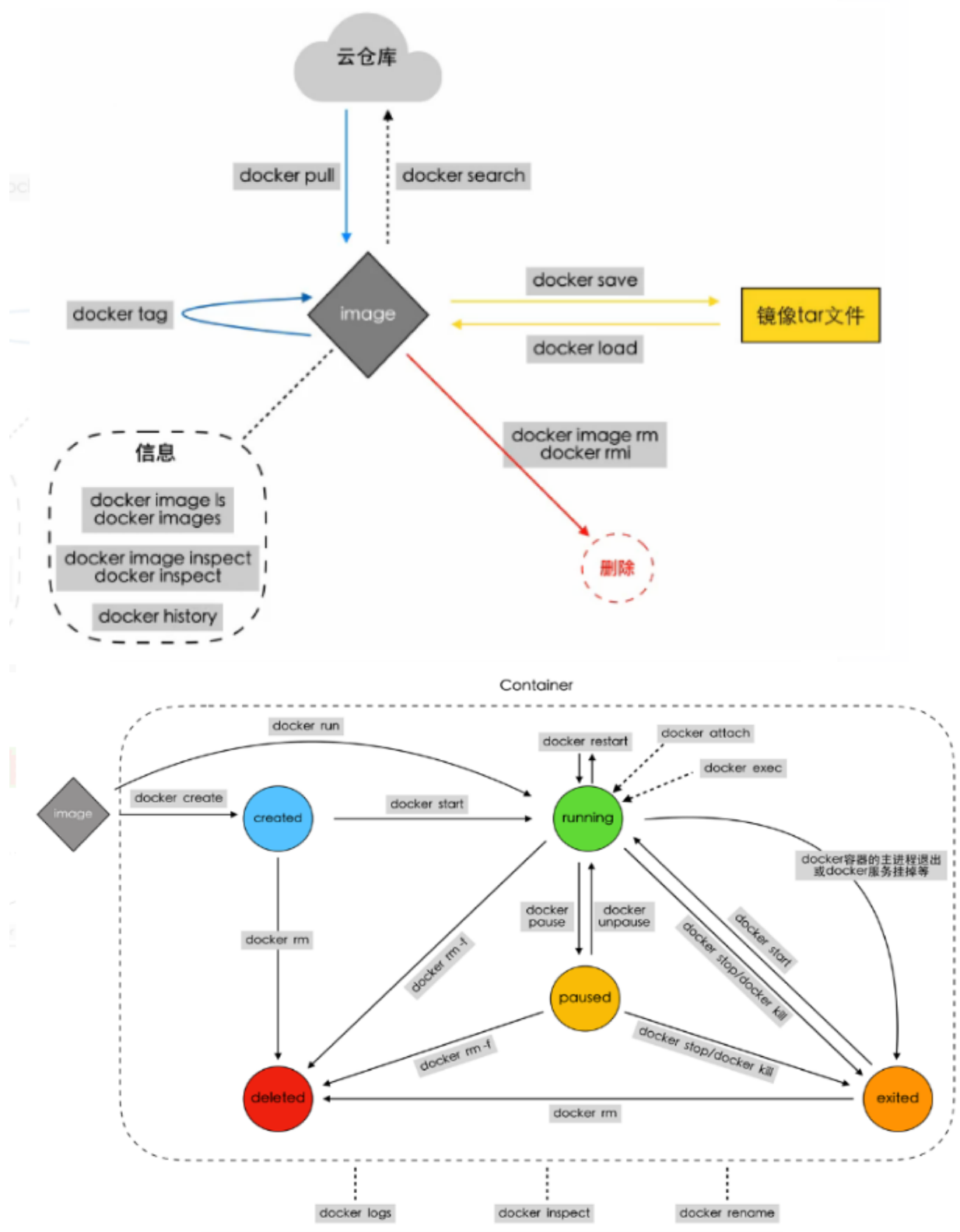
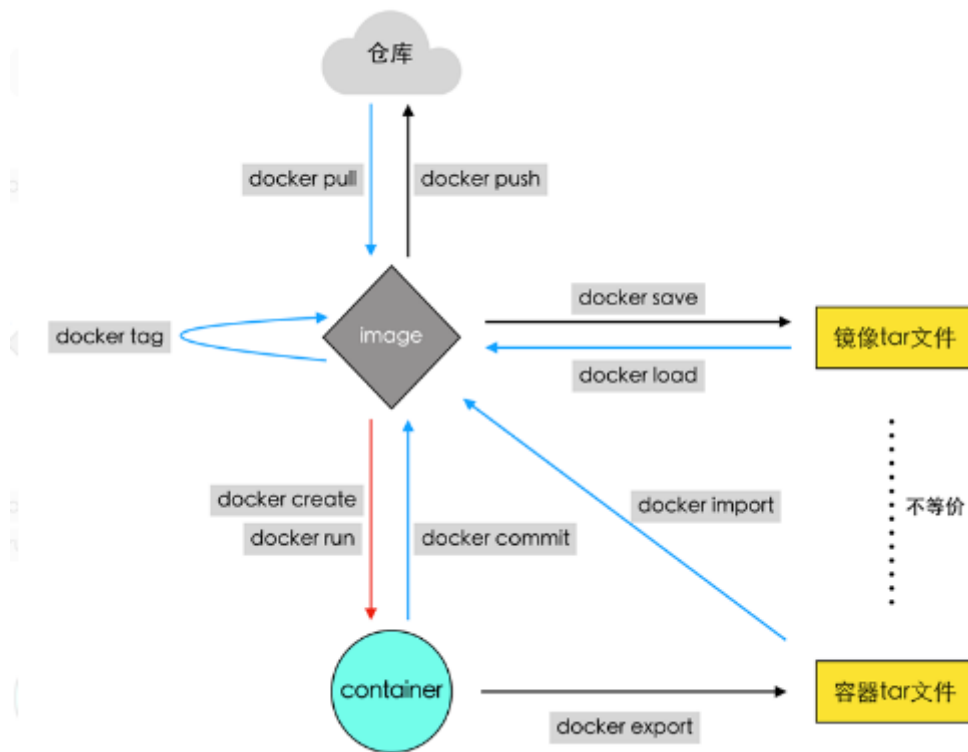


# 一, ==docker操作==





## • 1, 安装docker

- 进入源码docker文件中
- 执行命令
  - `sudo apt-key add gpg`
  - `sudo dpkg -i docker-ce_17.03.2~ce-0~ubuntu-xenial_amd64.deb`

## • 2, 检查docker是否安装成功

- `sudo docker run hello-world`
- 显示: hello from Docker! 表示已经安装成功

提示1: 如果不想每次都输入sudo, 执行, `sudo groupadd docker` --添加当前用户到docker组  
`sudo gpasswd -a ${USER} docker` --重启docker服务 `sudo service docker restart` --切换会话, 避免使用缓存的组信息 `newgrp -docker`

提示2: 如果没有镜像, 会去下载

•

## • 3, 启动, 停止docker服务

- 启动
  - `sudo service docker start`
- 停止
  - `sudo service docker stop`
- 启动docker
  - `sudo service docker restart`
- 状态检查
  - `sudo service docker status`

## • 4, 列出所有docker镜像

- `docker image ls`
- 或者: `docker images`



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
172.16.17.34:5000/my_nginx	late	53f3fd8007f7	3 weeks ago	109MB
nginx	latest	53f3fd8007f7	3 weeks ago	109MB
meiduo	latest	851f93f02b26	4 weeks ago	717MB
registry	latest	f32a97de94e1	2 months ago	25.8MB
hello-world	latest	fce289e99eb9	4 months ago	184kB
ubuntu	latest	113a43faa138	11 months ago	81.2MB
delron/fastdfs	latest	8487e86fc6ee	13 months ago	464MB
delron/elasticsearch-ik	2.4.6-1.0	095b6487fb77	13 months ago	689MB

- 5,拉取云端镜像
  - docker image pull hello-world
  - 查看镜像: docker image search hello-world
- 6,加载本地镜像
  - docker load -i 本地镜像包
- 7,删除镜像
  - docker image rm 镜像名或id
- 8,使用镜像,创建交互式容器
  - docker run [option] --name=容器名称 镜像 [传入的命的命令]
  - 例如:
    - docker run -it --name=myubuntu ubuntu
    - 解释: 使用ubuntu镜像创建交互式容器, 名字为myubuntu
    - 退出容器: ctrl + d
- 9,进入创建的容器
  - docker exec -it myubuntu /bin/bash
- 10,查看容器
  - 查看所有运行的容器
    - docker container ls
  - 查看运行和停止的容器
    - docker container ls -all
    - 或者: docker ps -a
- 11,运行,停止运行容器
  - 运行容器
    - docker container start 容器名或id
  - 停止容器
    - docker container stop 容器名或id
    - docker container kill 容器名或id
- 12,删除容器
  - docker container rm 容器名或id

## 二, ==使用Docker安装FastDFS==

- 1, 获取fastdfs镜像
  - 远程下载镜像

- docker image pull delron/fastdfs
- 本地镜像加载
  - docker load -i 文件路径/fastdfs\_docker.tar

#### ◦ 2,创建tracker容器

```
docker run -dti --network=host --name tracker -v /var/fdfs/tracker:/var/fdfs delron/fastdfs
tracker
```

- 解释: 将delron/fastdfs目录映射到/var/fdfs/目录中,创建名字为tracker的容器

#### ◦ 3,运行tracker容器

- docker container start tracker

#### ◦ 4,创建storage容器

```
docker run -dti --network=host --name storage -e TRACKER_SERVER=自己端口:22122 -v
/var/fdfs/storage:/var/fdfs delron/fastdfs storage
```

```
docker run -dti --network=host --name storage -e TRACKER_SERVER=192.168.163.143:22122
-v /var/fdfs/storage:/var/fdfs delron/fastdfs storage
```

- 解释: 将delron/fastdfs镜像目录, 映射到/var/fdfs/, 创建容器的名字为storage, 绑定追踪服务器的地址

#### ◦ 5,运行storage容器

- docker container start storage

如果不能运行删除/var/fdfs/storage/data目录下的fdfs\_storaged.pid\*\*文件