1.帮助命令

`docker version` 查看当前docker的版本号

`docker info` 显示 Docker 系统信息，包括镜像和容器数。。

`docker --help` 帮助命令类似于Linux的man命令

2.镜像命令

docker images　　后面可以追加参数

　　图解：　![在这里插入图片描述](https://img-blog.csdnimg.cn/20181130112330412.png)

　-a　　列出本地所有的镜像（含中间映像层）

　-q　　只显示镜像ID。

　--digests　　显示镜像的摘要信息

　--no-trunc　　显示完整的镜像信息

docker search ：查看镜像

　推荐官网学习：https://hub.docker.com

　格式：`docker search [OPTIONS] 镜像名字`

　例子： `docker search tomcat`

docker pull 下载镜像

　　格式：`docker pull [OPTIONS] 镜像名字`

docker rmi 删除镜像

　　备注：删除单个

　　格式：`docker rmi -f [OPTIONS] 镜像名字`

　　备注：删除多个

　　格式：`docker rmi -f 镜像名1:TAG 镜像名2:TAG `

　　备注：删除全部

　　格式：`docker rmi -f $(docker images -qa)`

3.容器命令

　　有镜像才能创建容器，这是根本的前提：

　　 　　下载一个centos镜像演示 命令：`docker pull centos`

　　 　　查看镜像

　　新建并启动容器

　　　　格式：`docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`

　　　　　　 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；

　　　　　　　-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；

　　　　　　　-P: 随机端口映射；

　　　　　　　-p: 指定端口映射，有以下四种格式

　　　　　　　 ip:hostPort:containerPort

　　　　　　　ip::containerPort

　　　　　　　 hostPort:containerPort

　　　　　　　 containerPort

　　列出当前所有正在执行的容器

　　　　格式：`docker ps [OPTIONS]`

　　　　　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。

　　　　　　--no-trunc :不截断输出。

　　退出容器

　　　１.`exit`：容器停止退出

　　　２.`ctrl+P+Q`：容器不停止退出

　　启动容器

　　　格式：`docker start 容器ID或者容器名`

　　重启容器

　　　格式：`docker restart 容器ID或者容器名`

　　停止容器

　　　格式：`docker stop 容器ID或者容器名`

　　强制停止容器

　　　格式：`docker kill 容器ID或者容器名`

　　删除已停止的容器

　　　格式：`docker rm 容器ID`　只删除一个

　　　格式：`docker rm -f $(docker ps -a -q)`　一次删除多个

　　　格式：`docker ps -a -q | xargs docker rm`　一次删除多个

　　启动守护容器

　　　格式：`docker run -d 容器名`

　　查看容器日志

　　　格式：`docker logs -f -t --tail 容器ID`

　　　　　 `-t` 是加入时间戳

　　　　　 `-f` 跟随最新的日志打印

　　　　　 `--tail` 数字 显示最后多少条

　　查看容器内运行的进程

　　 格式：`docker top 容器ID`

　　查看容器内部细节

　　　格式：`docker inspect 容器ID`

　　进入正在运行的容器并以命令行交互

　　　方式１：格式：`docker exec -it 容器ID bashShell`

　　　方式２：格式：`docker attach 容器ID`

　　　　区别：

　　　　　　attach 直接进入容器启动命令的终端，不会启动新的进程

　　　　　　exec 是在容器中打开新的终端，并且可以启动新的进程

　　从容器拷贝文件到主机上

　　　格式：`docker cp 容器ID:容器内路径 目的主机路径`

`attach Attach to a running container` # 当前 shell 下 attach 连接指定运行镜像

`build Build an image from a Dockerfile` # 通过 Dockerfile 定制镜像

`commit Create a new image from a container changes` # 提交当前容器为新的镜像

`cp Copy files/folders from the containers filesystem to the host path` #从容器中拷贝指定文件或者目录到宿主机中

`create Create a new container` # 创建一个新的容器，同 run，但不启动容器

`diff Inspect changes on a container's filesystem` # 查看 docker 容器变化

`events Get real time events from the server` # 从 docker 服务获取容器实时事件

`exec Run a command in an existing container` # 在已存在的容器上运行命令

`export Stream the contents of a container as a tar archive` # 导出容器的内容流作为一个 tar 归档文件[对应 import ]

`history Show the history of an image` # 展示一个镜像形成历史

`images List images` # 列出系统当前镜像

`import Create a new filesystem image from the contents of a tarball` # 从tar包中的内容创建一个新的文件系统映像[对应export]

`info Display system-wide information` # 显示系统相关信息

`inspect Return low-level information on a container` # 查看容器详细信息

`kill Kill a running container` # kill 指定 docker 容器

`load Load an image from a tar archive` # 从一个 tar 包中加载一个镜像[对应 save]

`login Register or Login to the docker registry server` # 注册或者登陆一个 docker 源服务器

`logout Log out from a Docker registry server` # 从当前 Docker registry 退出

`logs Fetch the logs of a container` # 输出当前容器日志信息

`port Lookup the public-facing port which is NAT-ed to PRIVATE\_PORT` # 查看映射端口对应的容器内部源端口

`pause Pause all processes within a container` # 暂停容器

`ps List containers` # 列出容器列表

`pull Pull an image or a repository from the docker registry server` # 从docker镜像源服务器拉取指定镜像或者库镜像

`push Push an image or a repository to the docker registry server` # 推送指定镜像或者库镜像至docker源服务器

`restart Restart a running container` # 重启运行的容器

`rm Remove one or more containers` # 移除一个或者多个容器

`rmi Remove one or more images` # 移除一个或多个镜像[无容器使用该镜像才可删除，否则需删除相关容器才可继续或 -f 强制删除]

`run Run a command in a new container` # 创建一个新的容器并运行一个命令

`save Save an image to a tar archive` # 保存一个镜像为一个 tar 包[对应 load]

`search Search for an image on the Docker Hub` # 在 docker hub 中搜索镜像

`start Start a stopped containers` # 启动容器

`stop Stop a running containers` # 停止容器

`tag Tag an image into a repository` # 给源中镜像打标签

`top Lookup the running processes of a container` # 查看容器中运行的进程信息

`unpause Unpause a paused container` # 取消暂停容器

`version Show the docker version information` # 查看 docker 版本号

`wait Block until a container stops, then print its exit code` # 截取容器停止时的退出状态值