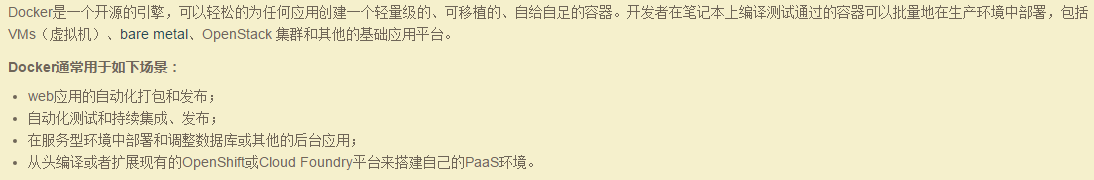
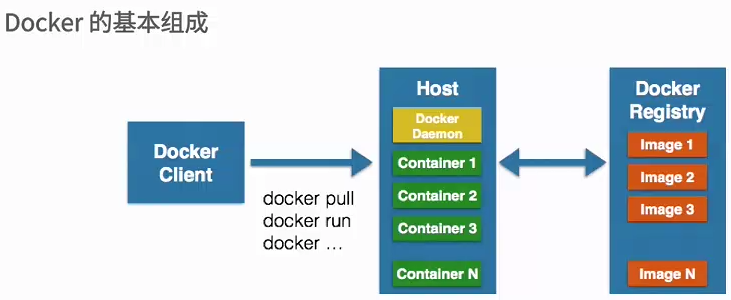
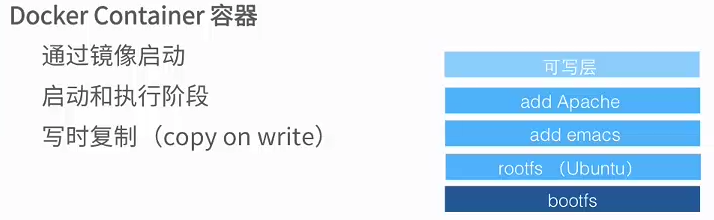
# 1，Docker简介与配置

## 1.1 Docker简介





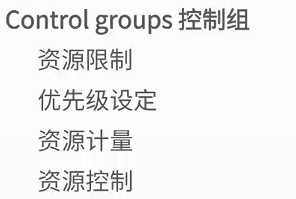


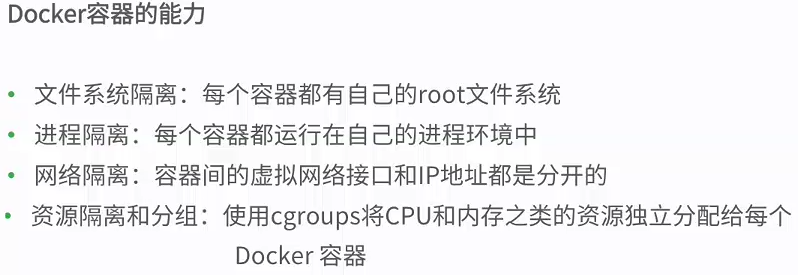




## 1.2 Docker容器相关技术

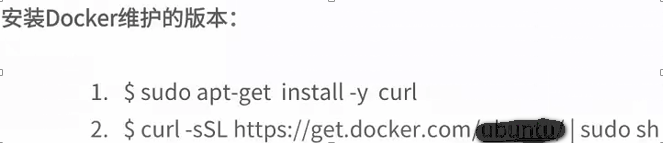


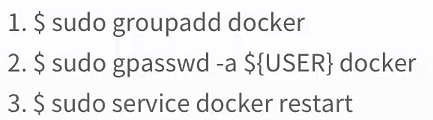




## 1.3 ubuntu安装docker

直接下载docker维护的安装脚本安装，省去了复杂的流程



配置docker用户组：

# 2，Docker 容器

## 2.1 交互式容器操作

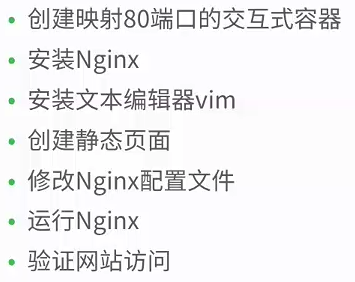
|  |  |
| --- | --- |
| 启动容器：单一命令容器，交互容器 | |
|  |  |
| 查看容器：容器和某个容器详细信息 | |
|  | +容器名 |
| 自定义容器名： | |
|  | |
| 重新启动已停止的容器： |  |
| 删除已经停止的容器： |  |

## 2.2 守护式容器操作

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

## 2.3 部署静态网站

基本部署流程：

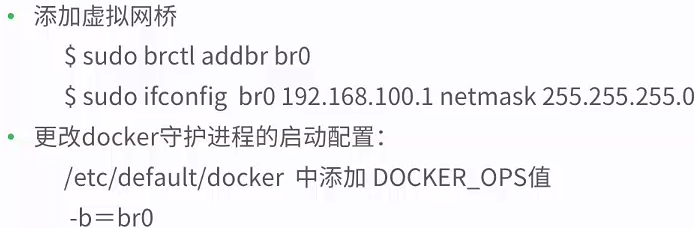


创建带有映射端口的交互容器：



## 2.4 容器网络连接

自定义容器的虚拟网桥IP：



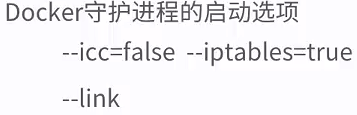
指定相互可以通信的容器：



拒绝所有容器间的通信：

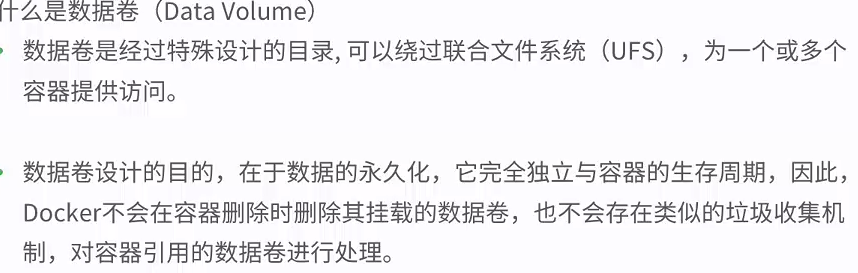


允许特定容器间通信：



## 2.5 容器数据管理

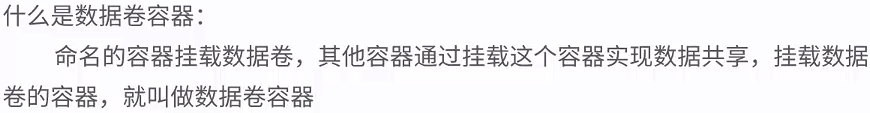
容器内数据卷：

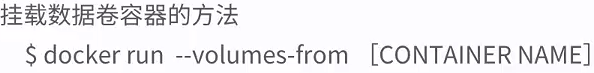


数据卷操作：

|  |
| --- |
|  |
|  |
|  |
|  |

数据卷容器：





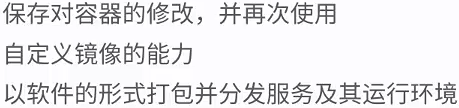
# 3，Docker镜像

## 3.1 镜像基本操作

|  |
| --- |
| 查看镜像： |
| 查看镜像详细信息： |
| 删除镜像： |
| 查找hub中的镜像： |
| 从hub下载镜像：    配置第三方镜像地址：加速下载 |
| 提交镜像： |
|  |

## 3.2 创建镜像

构建镜像的作用：



构建镜像有两种方式：第二种更适合高级操作

|  |
| --- |
| Commit：修改已有镜像提交 |
| 通过Dockerfile来build镜像：类似makefile  Dockerfile包含了一系列的命令，例如：    Docker build命令：    命令执行的样子： |

## 3.3 Dockerfile文件

Dockerfile指令：

|  |  |
| --- | --- |
|  |  |
|  | |
|  |  |
|  | 仍需要在运行容器的时候使用-p指定端口，这里expose仅告诉docker镜像要使用的端口 |
|  | 指定镜像作为容器运行时的默认命令 |
|  | 指定镜像作为容器运行时的默认命令，当用户自定义运行命令时，默认命令依旧执行 |
|  | 复制文件到镜像中，add默认带有解压缩功能 |
|  | 指定后续操作使用的工作目录 |
|  | 指定镜像运行的用户，默认不指定是root |
|  |  |

Docker build的过程：

