```
核的cgroup,namespace,以及OverlayFS类的Union FS等技术,对进程进行
                                                                   封装隔离,属于操作系统层面的虚拟化技术。由于隔离的进程独立于宿主和其
                                                                   他的隔离进程,因此也称其为容器。
                                                                            Architecture In Linux
                                                                                       REST Interface
                                                                          libcontainerd libnetwork graph plugins
                                                                         containerd + runc
                                                                                            Union Filesystems AUF:
btrfs, vfs, zfs*,
DeviceMapper
                                                                                                                       runc是linux命令行工具,它根据OCI容器运行时规范创建和运行容器。
                                                                                                                       containerd是一个守护程序,它管理容器生命周期,提供了在一个节点上执
                                               01 docker简介与安装
                                                                                                                       行容器和管理镜像的最小功能集。
                                                                   docker的技术优势(相比传统虚拟化技术): 容器内的应用进程直接运行于
                                                                   宿主的内核,容器内没有自己的内核,而且也没有进行硬件虚拟。
                                                                                                   操作系统分为内核和用户空间,在linux中,挂载的root文件系统提供了用户
                                                                                                                                                        静态数据
                                                                                                   空间。docker的镜像是特殊的文件系统,好比root文件系统。
                                                                                      镜像lmage
                                                                                                                                                        分层存储
                                                                                                     1\镜像是静态的定义,容器是镜像的运行实体。2\容器实质是进程,但直接运
                                                                   Docker三大基本概念
                                                                                                     行于宿主的内核,运行于独立的命名空间。因此它拥有自己的root文件系统、
                                                                                      容器Container
                                                                                                     网络配置、进程空间等。
                                                                                                                                                          容器=镜像基础层+容器存储层
                                                                                      仓库Repository
                                                                                                      一个集中的存储、分发镜像的服务。
                                                                                                                                  镜像名=<仓库名>:<标签> 。<标签>默认为latest。
                                                                   <u>Docker安装</u>
                                                                                                                            修改为国内仓库的方法
                                                                                      获取镜像: docker pull [选项] <仓库名>:<标签>。
                                                                                                                            镜像采用分层存储,因此也是一层层下载的。
                                                                                      列出本地的镜像: docker image ls
                                                                                                                  镜像 ID 则是镜像的唯一标识
                                                                                                                                   由<仓库名>:<标签> 或 镜像ID 或 镜像hash 来指出目标镜像
                                                                                                                                   因为一些依赖关系的存在,可能导致删除镜像失败。
                                                                                      删除本地镜像: docker image rm [选项] <镜像1> [<镜像2>...]
                                                                                                                                                      批量删除mongo镜像: $ docker image rm $(docker image ls -q mongo)
                                                                                                                                   配合其他命令来删除
                                                                                                                                                     批量删除mongo:3.2之前的镜像: $ docker image rm $(docker image ls -q -f before=mongo:3.2)
                                                                                                         Dockerfile是文本文件,包含了
                                                                   Docker镜像操作命令
                                                                                                                                  注意:每一层构建的最后一定要清理掉无关文件。
                                                                                                         一条条指令,每条指令描述如
                                                                                                         何构建镜像中的一层内容。
                                                                                                                                              FROM指令:指定基础镜像,特别地,FROM scratch 指定一个空白的镜像。
                                                                                      Dockerfile制作镜像
                                                                                                                                              RUN指令:执行Linux命令行的命令。
                                                                                                         构建镜像:docker build [选项] <上下文路径/URL/->
                                                                                      跨平台构建镜像:比如希望在不同架构CPU的设备上运行程序,可使用
                                                                                      QEMU的用户态模式 (User mod) 来构建跨平台镜像。
                                                                                                     查看默认存储位置: docker info | grep "Docker Root Dir"
                                                                                      镜像存储位置
                                                                                                                     方法1、使用软连接
                                                                                                                     方法2、在配置文件中指定容器启动的参数。
                                                                                                     修改默认存储位置
                                               02 Docker镜像与容器
                                                                                                                      方法3、System 下创建配置文件
                                                                                      新建并启动容器: docker run [选项] 镜像
                                                                                      启动已终止的容器: docker container start
                                                                                                                   stop和kill的本质是向容器这个特殊进程发送SIGTERM信号和SIGKILL信号
                                                                                      停止容器:docker stop、docker kill
                                                                                                                   docker ps 列出运行中的容器,docker ps -a可列出所有容器。
                                                                                      重启容器:docker start、docker restart
                                                                                                                    docker start会保留上一次启动的所有参数启动。
                                                                                                                  docker container Is 命令来查看容器信息,加-a参数查看所有容器信息。
                                                                                      后台运行容器: docker run -d 镜像
                                                                                                                  docker container logs 获取容器的输出信息
                                                                   Docker容器操作命令
                                                                                      进入容器: docker exec -it 69d1 bash
                                                                                                                      -t参数给容器分配一个伪终端; -i参数让容器标准输入保持打开。
                                                                                      暂停容器:docker pause 容器 [容器...]
                                                                                                                                    添加 -f 参数,可删除运行中的容器。Docker 会发送 SIGKILL 信号给容器。
                                                                                      删除容器: docker container rm 来删除一个处于终止状态的容器
                                                                                                                                    $ docker container prune 清理所有处于终止状态的容器。
                                                                                                                                    $ docker rm -v $(docker ps -aq -f status=exited) 批量删除所有已经退出的容器。
                                                                                      导出容器: docker export 7691a814370e > ubuntu.tar
                                                                                                            cat ubuntu.tar | docker import - test/ubuntu:v1.0
                                                                                      导入容器:docker import
                                                                                                            $ docker import http://example.com/exampleimage.tgz
                                                                                                             example/imagerepo
                                                                                                                  查看所有数据卷: $ docker volume ls
                                                                              创建数据卷:$ docker volume create datawhale
                                                                                                                  查看指定数据卷: $ docker volume inspect datawhale
                                                                              启动一个挂载数据卷的容器
                                                                  数据卷
                                                                              查看数据卷的具体信息: $ docker inspect 容器名
                                                                                                              无主的数据卷清理:$ docker volume prune
                                                                              删除数据卷:$ docker volume rm 数据卷名
                                               03 Docker数据管理
                                                                                挂载一个主机目录作为数据卷
                                                                                                       使用 --mount 标记可以指定挂载一个本地主机的目录到容器中
                                                                  挂载主机目录
                                                                                查看数据卷的具体信息: $ docker inspect 容器名
                                                                                挂载一个本地主机文件作为数据卷
                                                                                                         --mount 标记也可以从主机挂载单个文件到容器中
                                                                                                 通过-p 参数来设置端口映射,通过docker container ls可查看主机端口到容器端口的映射关系。
                                                                                                 映射所有接口地址: $ docker run -d -p 80:80 nginx:alpine
                                                                                                                                           -d 参数:后台运行容器,并返回容器id
                                                                                  外部访问容器
                                                                                                 映射到指定地址的指定端口: $ docker run -d -p 127.0.0.1:80:80 nginx:alpine
datawhale docker
                                                                                                 映射到指定地址的任意端口: $ docker run -d -p 127.0.0.1::80 nginx:alpine
docker从入门到实践
                                                               Docker基础网络介绍
                                                                                              应该使用自定义docker网络来连接多个容器,不建议使用--link参数; 如果有
                                                                                  容器互联
                                                                                             多个容器互联,建议docker compose方法。
                                                                                                                            默认配置/etc/resolv.conf文件 或者 添加配置/etc/docker/daemon.json文件
                                                                                  配置DNS: 利用虚拟文件来挂载容器的 3个相关配置文件
                                                                                                                            查看dns配置文件是否生效: $ docker run -it --rm ubuntu:18.04 cat etc/resolv.conf
                                                                                 备注: docker network Is 命令查看网络;容器内部 ip addr命令查看Ip
                                                                                 Bridge模式:为每一个容器分配、设置 IP等,并将容器连接到一个 docker0 虚拟网桥,默认为该模式。
                                                               Docker的网络模式
                                                                                                                                                                   ipv4 ip_forward
                                                                                 Host模式:容器将不会虚拟出自己的网卡,配置自己的 IP 等,而是使用宿主机的 IP 和端口。
                                                                                 None模式:容器有独立的 Network namespace,但并没有对其进行任何网
                                                                                 络设置,如分配 veth pair 和网桥连接,IP 等。
                                                                                 Container模式:新创建的容器不会创建自己的网卡和配置自己的 IP,而是和
                                                                                 一个指定的容器共享 IP、端口范围等。
                                                                                                                 -b BRIDGE 或 --bridge=BRIDGE 指定容器挂载的网桥。
                                               04 Docker网络
                                                                                                                 --bip=CIDR定制docker0的掩码。
                                                                                                                 -H SOCKET... 或 --host=SOCKET... Docker 服务端接收命令的通道
                                                                                                 启动容器时配置
                                                                                                                 --icc=true|false 是否支持容器之间进行通信
                                                                                                                 --ip-forward=true|false 请看下文容器之间的通信
                                                                                                                 --iptables=true|false 是否允许 Docker 添加 iptables 规则
                                                                                                                 --mtu=BYTES 容器网络中的 MTU
                                                                                  快速配置指南
                                                                                                                    --dns=IP_ADDRESS... 使用指定的DNS服务器
                                                                                                 启动或运行容器时配置
                                                                                                                     --dns-search=DOMAIN... 指定DNS搜索域
                                                                                                                 -h HOSTNAME 或 --hostname=HOSTNAME 配置容器主机名
                                                                                                                 --link=CONTAINER_NAME:ALIAS 添加到另一个容器的连接
                                                                                                 运行容器时配置
                                                                                                                 --net=bridge|none|container:NAME_or_ID|host 配置容器的桥接模式
                                                                                                                 -p SPEC 或 --publish=SPEC`映射容器端口到宿主主机
                                                                                                              -P or --publish-all=true|false 映射容器所有端口到宿主主机
                                                                                                 容器访问控制的介绍:容器的访问控制,主要通过 Linux 上的 iptables 防火
                                                                                                 墙来进行管理和实现。
                                                                                                                  检查本地转发是否打开: $sysctl net.ipv4.ip_forward
                                                                                                 容器访问外部网络
                                                                                                                 打开本地转发: $sysctl -w net.ipv4.ip_forward=1 或者 容器启动时指定参数 --ip-forward=true
                                                                                                               条件1:容器的网络拓扑是否互联,默认采用bright模式桥接到docker0网桥
                                                                                  容器访问控制
                                                                                                容器之间访问
                                                                                                              上。条件2: 本地系统防火墙软件--iptables是否允许通过。
                                                                                                               默认情况下,不同容器之间是允许网络互通的。如果为了安全考虑,可以在
                                                                                                              /etc/docker/daemon.json 文件中配置 {"icc": false} 来禁止它。
                                                                                                 访问所有端口
                                                               Docker高级网络配置
                                                                                                              详细过程
                                                                                                 访问指定端口
                                                                                                              详细过程
                                                                                                 端口映射实现的介绍:默认情况下,容器可以主动访问到外部网络的连接,但
                                                                                                 是外部网络无法访问到容器。
                                                                                                                                                     查看主机的NAT规则的命令如下: $ sudo iptables -t nat -nL
                                                                                                 容器访问外部
                                                                                                              详细过程
                                                                                  端口映射实现
                                                                                                               详细过程
                                                                                                 外部访问容器
                                                                                                              docker run -p参数指定映射 (见网络基础部分)
                                                                                                  详细过程
                                                                                  配置docker网桥
                                                                                                  brctl show 命令来查看网桥和端口连接信息
                                                                                  自定义网桥
                                                                                               详细过程
                                                                                               pipework
                                                                                  工具和示例
                                                                                               playground
                                                                                                   Docker 1.2.0 开始支持在运行中的容器里编辑 /etc/hosts, /etc/hostname
                                                                                                   和 /etc/resolv.conf 文件。
                                                                                  编辑网络配置文件
                                                                                                                                                        这些修改是临时的,仅容器运行时保留,也不会被docker commit提交。
                                                                                  实例: 创建一个点到点的连接
                                                                                                          详细过程
                                                05 Docker Compose
                                                                  docker compose介绍:多个容器互联
```

Docker是在GitHub上维护的开源项目,使用Go语言开发实现,基于Linux内