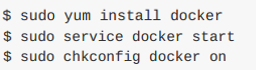
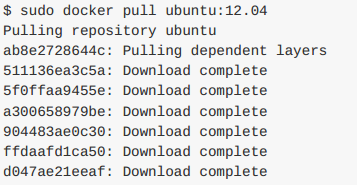
5.1 Docker安装

本题目是在阿里云（CentOS 7.0）平台下进行的。由于CentOS 7.0库中已经带有Docker，所以可以直接安装，如果是低于7.0版本，需要添加Docker源。

Docker的安装与自启动



获取Docker镜像，本题目中使用到的镜像为Ubuntu:12.04



执行上面的命令之后，会从Docker Hub仓库下载一个Ubuntu:12.04操作系统的镜像，由于在阿里云平台下面配置了阿里云的镜像，配置方式见图5-1，所以上面会从阿里云上面获取上面版本的系统。

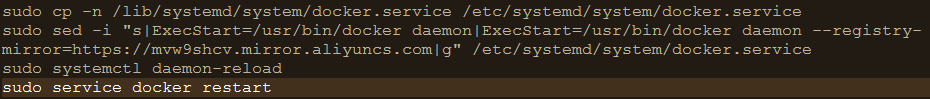


图5-1. 配置阿里云镜像与加速器[5-1]

如果需要查看我们刚刚下载的镜像，可以使用docker images 命令来查看。

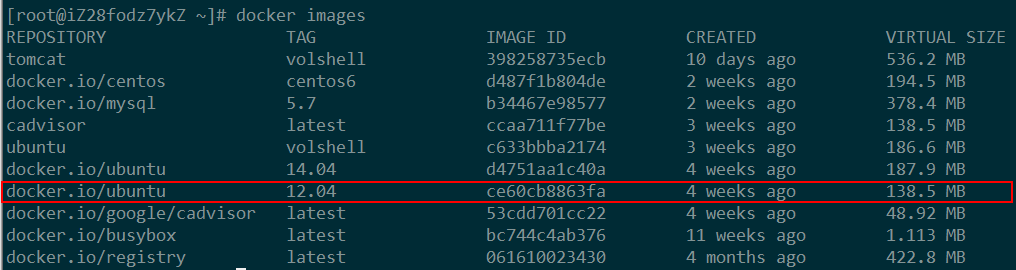


图5-2. Docker镜像列表

5.2 运行容器

当下载完一个镜像之后，我们就可以基于该镜像创建一个容器，创建并启动命令

sudo docker run –ti Ubuntu:12.04 /bin/bash

并通过sudo docker ps –a 命令查看正在运行中的容器。

具体一些命令的使用，不在本文的范围内，相关命令查看Docker文档。

5.3 Docker数据卷

在一个应用中往往有些数据是需要在一个或者多个容器中共享，例如日志文件，Docker提供了数据的共享机制——数据卷/数据卷容器。

5.3.1 数据卷

数据卷是能够实现一个或多个容器使用的特殊目录，它绕过UFS，从功能上讲类似与Linux的mount命令，它提供了多种功能性的特性：

1). 数据卷可以在容器之间共享和重用

2). 对数据卷的修改，在容器中会立马生效

3). 对数据卷的变更不会对镜像有影响

4). 数据卷会一直存在，直到没有任何容器使用它

在运行一个Docker 容器的时候，使用-v选项，同时指定主机源目录和容器目标目录，二者使用“:”分割，多次使用-v标记可以创建多个数据卷。

5.3.2 数据卷容器

有些时候，我们需要在多个容器之间共享数据，而且这些数据是需要持续更新的，这种情况下，如果让每一个容器挂载一个数据卷尤为显得复杂。为了解决这个问题，Docker提出了容器卷容器的概念。

数据卷容器其实是一个普通的容器，专门用来提供数据卷供其它容器挂载。 创建数据卷容器的方式和创建普通挂载了数据卷的容器的方式是一样的，使用-v 命令。之后使用

—volumes-from命令将创建的数据卷容器挂载到其他容器中，同样多次使用该命令可以挂载多个数据卷容器。需要特别注意的是如果需要删除数据卷需要使用-v命令删除，默认是不会删除的。除此之外，数据卷容器也为数据的热备提供方便，只需使用上面命令挂载数据卷容器同时指定tar -cvf /backup/backup.tar /data既可将/data中的数据热备为backup.rar。

5.4 Docker网络

在正确安装完Docker之后，会在本机生成一个名为Docker0的网桥，如下图

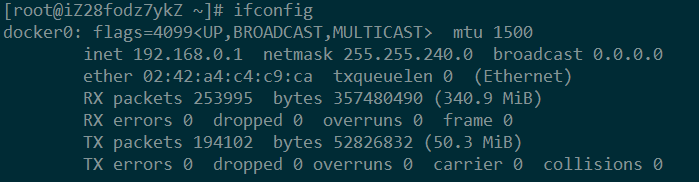


图5-3.Docker0网桥

Docker0桥接模式网络是Docker提供给使用者默认的网络模式，在该种模式下面，Docker 后台程序会创建 docker0，一个虚拟的以太网桥，用于自动转发与之连接的任意网络接口间的数据包。默认情况下，该后台进程会创建一对对等接口(peer interface)，分派其一为容器的 eth0 接口，另一个则分派在主机的命名空间下，默认的名称为vethXXX，同时从私有 IP 地址范围中为该桥接分配一个 IP 地址或子网络，由此，同一主机上的所有容器全部连接到该网桥上。如图4所示，

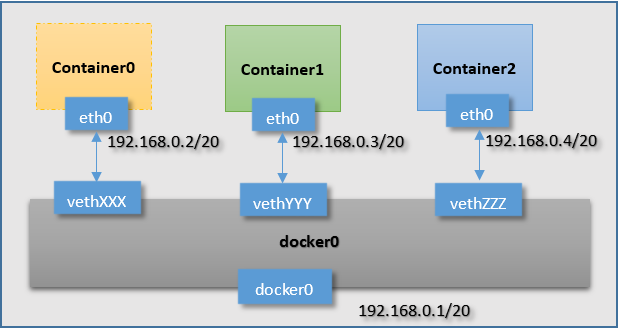


图5-4.桥接模式网络示意图

在该种模式下可以使用-p/P命令来指定容器对外接口。

除了上面Docker默认的网络模式之外，还提供其他三种网络模式：

1).主机网络模式，该模式下禁用容器自身网络隔离，与主机共享主机的网络命名空间。

2).容器网络模式，该模式下，能够重用其他正在运行的容器的网络命名空间。

3).无网络模式，该模式下对容器的进行配置，使用者可以自行对网络进行定义。

5.5 Docker容器互联

Docker容器给复杂的应用带来的生机，尤其是微服务概念的出现，开发者可以更新应用程序的单个组件，而不影响其他的部分。但是各个组件之间的交互是必需的，Docker允许使用者使用相关命令实现勇气之间的互联。

5.5.1 同主机容器互联

通过link方式创建容器，然后我们可以使用被link容器的别名进行访问，从而解除应用对IP的依赖，这样会在容器的/etc/hosts文件中生成相应的DNS解析。不幸的是link方式只能解决单机容器间的互联。跨主机情况下，容器的互联需要其他的方式。

5.5.2 跨主机容器互联

容器的跨主机互联的基本思想可以使用下图来表示：

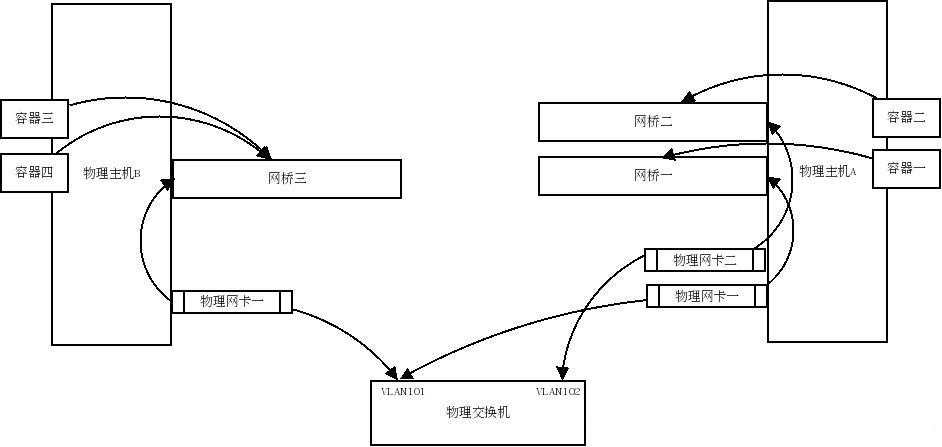


图5-5.跨主机容器互联的原理[5-2]

如果要实现跨主机的容器互联可以使用下面的方式：

1).修改主机中的hosts文件的方式。

2). Docker官方提供了一种ambassador的agent方案。需要借助svendowideit/ambassador的image，将不同host进行解耦合。如图5-6所示，

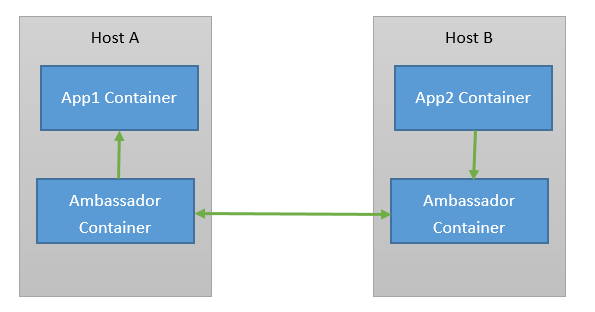


图5-6. Ambassador方式跨主机容器互联

5.6 Docker容器监控与编排

5.6.1 Docker容器监控

随着Docker的发展，Docker容器中性能越来越收到重视，比如容器的CPU，Memory等性能指标的控制显得尤为重要，对应用的部署有着相当重要的决策作用。

本系统中的Docker容器监控方案选择使用Google cAdvisor，这是一个 Docker 容器内封装的实用工具，能够搜集、集料、处理和导出运行中的容器的信息。通过它可以看到 CPU 的使用率、内存使用率、网络吞吐量以及磁盘空间利用率。cAdvisor只能监控同一主机的容器的运行状况，对与跨主机的多节点是无效的。cAdvisor的启动方式如下图5-7所示，

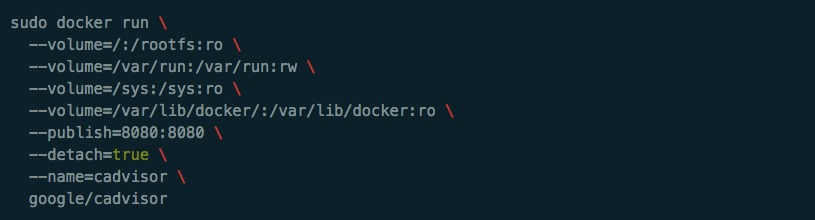


图5-7. cAdvisor的启动

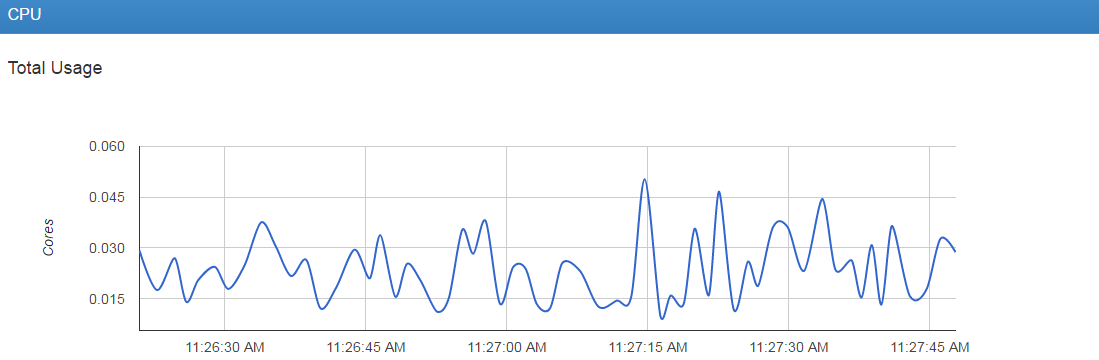


图5-8. cAdvisor对CPU的监控

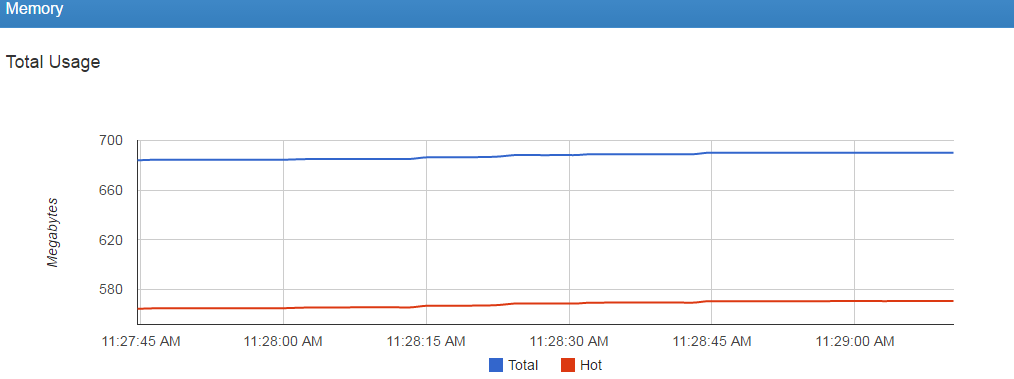


图5-9. cAdvisor对Memory的监控

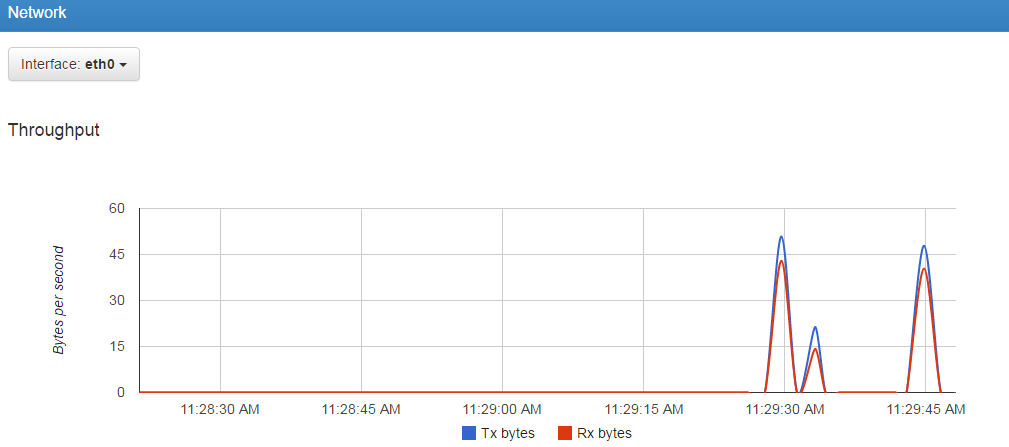


图5-10. cAdvisor对Network的监控

另外，[Cloud Insight](http://www.oneapm.com/ci/feature.html?utm_source=Community&utm_medium=Article&utm_term=201512DockerComparison&utm_campaign=CiArti&from=jscwftqp) 是国内唯一一款的Docker监控软件，支持多种操作系统、云主机、数据库和中间件的监控，原理是在平台服务仪表盘和自定义仪表盘中，采集并处理 Metric，对数据进行聚合与分组等计算，提供曲线图、柱状图等多样化的展现形式。

### 5.6.1 Docker容器编排

5.7 Dockerfile的使用

除了使用docker pull命令从Docker Hub或其他镜像库拉去镜像之外，Docker允许使用者自行在基础镜像的基础上制作自己需要的镜像。Dockerfile由一行行命令语句组成，并且支持以#开头的注释行。一般的，Dockerfile分为四部分：基础镜像信息、 维护者信息、 镜像操作指令和容器启动时执行指令。Docker中常用指令说明见下表1所示，

|  |  |  |  |
| --- | --- | --- | --- |
| 命令名称 | 简介 | 命令名称 | 简介 |
| FROM | 首个命令，指定来自基础镜像 | ADD/COPY | 拷贝文件到容器 |
| MAINTAINER | 维护作者信息 | CMD | 指定容器启动后命令 |
| RUN | 指定运行的命令 | ENV | 设置容器环境变量 |
| VOLUME | 指定容器挂载目录 | EXPOSE | 指定对外端口 |

表5-1.Dockerfile命令及简介

[5-1]. 阿里云Docker镜像配置.

[<https://cr.console.aliyun.com/?spm=5176.1971733.0.2.JbKBiF#/docker/booster>].阿里云官网

[5-2]. [**美味小鱼**](http://my.csdn.net/qcpm1983). [ <http://blog.csdn.net/smallfish1983/article/details/38845351>] CSDN博客