第四章：Docker容器升华--swarm集群

**一、Docker Swarm集群概述；**

**二、Docker Swarm核心架构；**

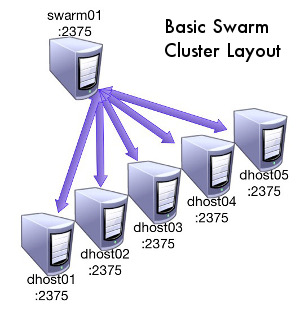
**三、案例：实现Docker Swarm高可用集群；**

**四、扩展学习；**

**一、Docker Swarm集群概述；**

**概述：**当有多台物理机的时候，就要考虑使用集群的模式了，那么docker如何来使用集群来进行管理呢？在这里主要使用的是docker自带的swarm mode，也就是docker集群的管理和编排。所谓的编排就是指多台集群的管理，主机的配置，容器的调度等；

Docker swarm 是docker官方的三大项目之一，提供docker容器集群服务，是docker官方对容器云生态支持的核心方案，Docker Swarm项目开始于2014年，是Docker公司推出的第一个容器集群项目。使用docker swarm可以将多个docker主机封装为一个大型的虚拟docker主机，swarm集群提供给用户管理集群内所有容器的操作接口与使用一台docker主机几乎相同，从而可以快速打造一套容器云服务，Swarm和Kubernetes比较类似，但是更加轻量级，具有的功能也较kubernetes更少一些；



**项目特点：**

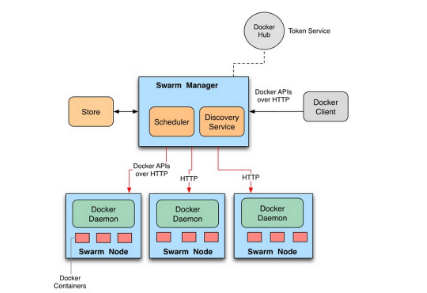
swarm v1是典型的mater-slave结构，需要通过发现服务来选举出manager，而manager是中心管理节点，而各个节点通过运行agent接受manager的统一管理；

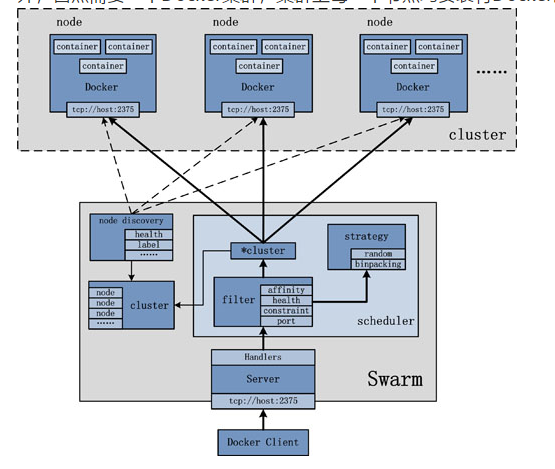
swarm v2中，swarm集群会自动通过Raft协议分布式选举出manager节点，因此无需配置额外的发现服务，从而避免了单点瓶颈；且swarm v2内置了DNS负载均衡和对外部负载均衡机制的支持；

**二、Docker Swarm核心架构；**

**Docker Cluster：**Docker Cluster由多个Docker Node组成，而负责给Swarm发送请求的是Docker Client；

**Swarm deamon：**只是一个调度器（Scheduler）加路由器（route），Swarm自己不运行容器，它只是接受docker客户端发送过来的请求，调度适合的节点来运行容器，这意味着，即使Swarm由于某些原因挂掉了，集群中的节点也会照常运行，当Swarm重新恢复运行之后，它会收集重建集群信息；





**总结：**Swarm守护进程本身相当于是一个调度器和一个路由器。它实际上并没有运行容器，也就是说，如果Swarm服务停止了，它在终端Docker主机上已分配好的容器仍然是开启的。另外，由于它不处理任何网络路由（网络连接需要被直接发送到后端的Docker主机上），即使Swarm守护进程意外终止，运行的容器仍然可用。当Swarm从这样的崩溃中恢复，它依然能够查询终端以重建其元数据的列表；

**三、案例：实现Docker Swarm高可用集群；**

**核心配置：**

docker swarm init ##初始化集群，注意复制--token（票据）

docker swarm join --token 主管理节点（swarm）的票据及ip:2375 ##加入集群

docker service inspect swarm\_cluster --pretty ##查看集群

docker node ls ##查看加入的集群节点

docker service ps swarm\_cluster ##查看集群的进程

docker service scale swarm\_cluster=3 ##设置集群中生效的节点数量

**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统类型 | IP地址 | 主机名 | 所需软件 |
| Centos 7.4 1708 64bit | 192.168.100.101 | swarm.linuxfan.cn | docker-engine |
| Centos 7.4 1708 64bit | 192.168.100.102 | docker1.linuxfan.cn | docker-engine |
| Centos 7.4 1708 64bit | 192.168.100.103 | docker2.linuxfan.cn | docker-engine |

**案例步骤：**

* 配置基础环境（网络、解析等，在此列举swarm节点的配置）；
* 在所有节点安装docker，并且配置加速器（在此列举swarm节点的配置）；
* 在所有节点禁用“live-restore”，swarm不兼容活动存储（在此列举swarm节点的配置）；
* 配置swarm节点成为管理节点：
* 配置所有docker工作节点加入到swarm集群中：
* 在swarm节点上进行验证集群状态：
* 在所有节点上进行构建httpd镜像（在此列举swarm节点的配置）：
* 在swarm节点上配置集群；
* 客户端节点访问测试swarm集群：
* 添加swarm节点到集群中作为node节点：
* 客户端节点访问测试swarm集群：
* 关闭docker1节点的容器；
* 客户端节点访问测试swarm集群：
* **配置基础环境（网络、解析等，在此列举swarm节点的配置）；**

[root@swarm ~]# cat <<END >>/etc/hosts

192.168.100.101 swarm.linuxfan.cn

192.168.100.102 docker1.linuxfan.cn

192.168.100.103 docker2.linuxfan.cn

END

* **在所有节点安装docker，并且配置加速器（在此列举swarm节点的配置）；**

参考教案五十六章；

* **在所有节点禁用“live-restore”，swarm不兼容活动存储（在此列举swarm节点的配置）；**

[root@swarm ~]# vi /etc/docker/daemon.json

{"registry-mirrors": ["http://9a248fb4.m.daocloud.io"]}

{

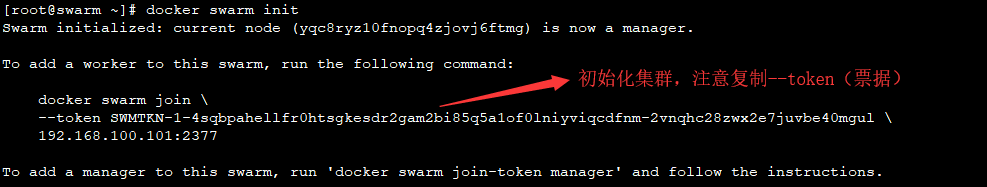
"live-restore":false

}

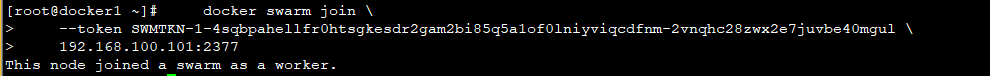
:wq

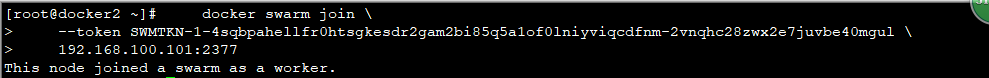
[root@swarm ~]# systemctl restart docker

* **配置swarm节点成为管理节点：**

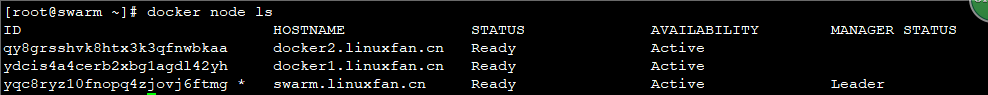


* **配置所有docker工作节点加入到swarm集群中：**





* **在swarm节点上进行验证集群状态：**



* **在所有节点上进行构建httpd镜像（在此列举swarm节点的配置）：**

[root@swarm ~]# mkdir apache

[root@swarm ~]# vi apache/Dockerfile

FROM centos

MAINTAINER The Centos Project <cloud@linuxfan.cn>

RUN yum -y install httpd

EXPOSE 80

ADD index.html /var/www/html/index.html

ADD run.sh /run.sh

RUN chmod 755 /run.sh

CMD ["/run.sh"]

:wq

[root@swarm ~]# vi apache/run.sh

#!/bin/bash

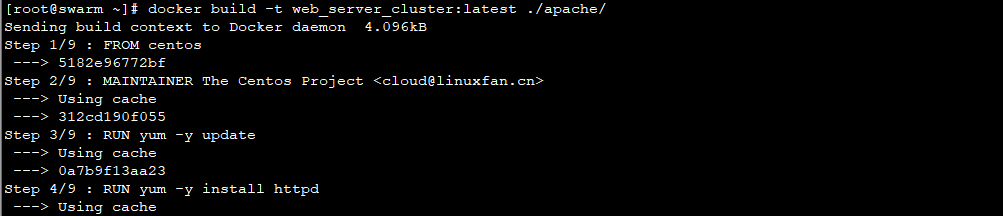
rm -rf /run/httpd/\*

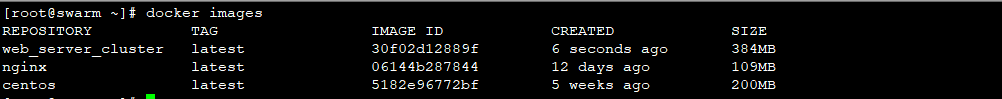
exec /usr/sbin/apachectl -D FOREGROUND

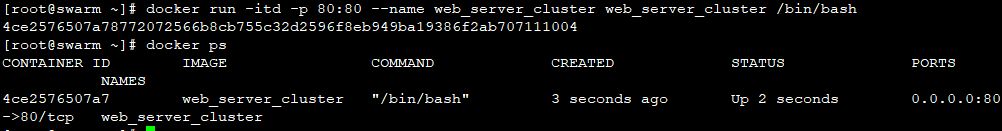
:wq

[root@swarm ~]# echo "swarm.linuxfan.cn" >apache/index.html

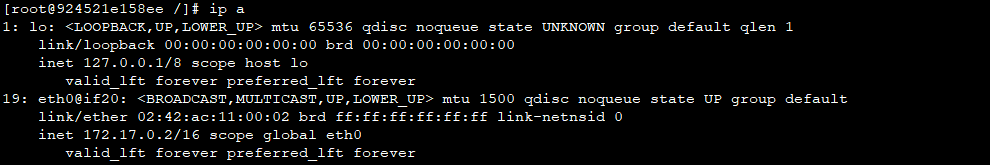


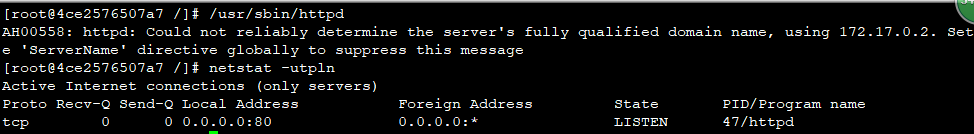


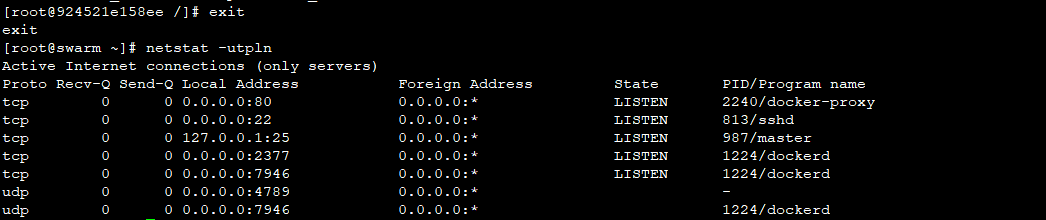






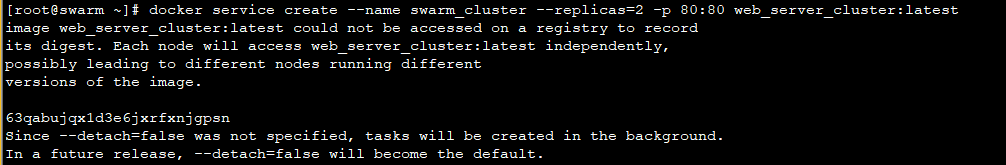




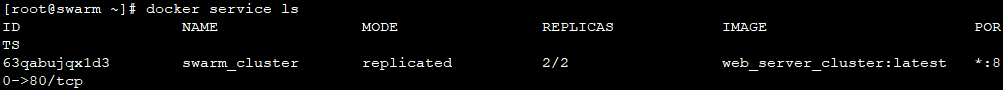


* **在swarm节点上配置集群；**

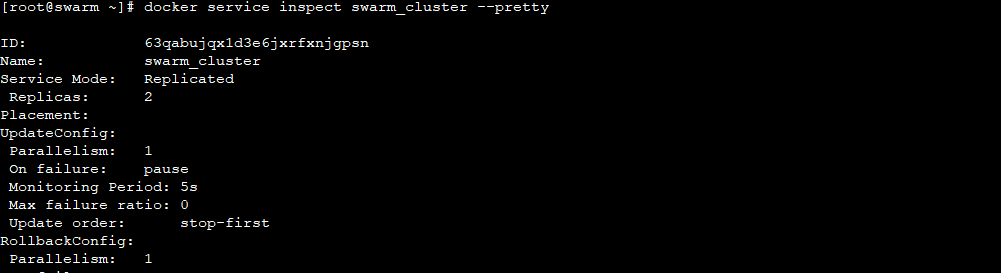
创建集群服务并且设置端口映射：

****

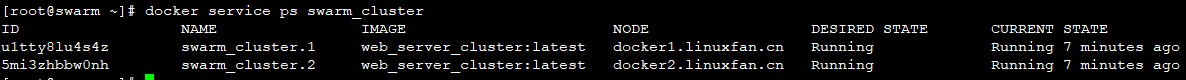
查看集群信息：



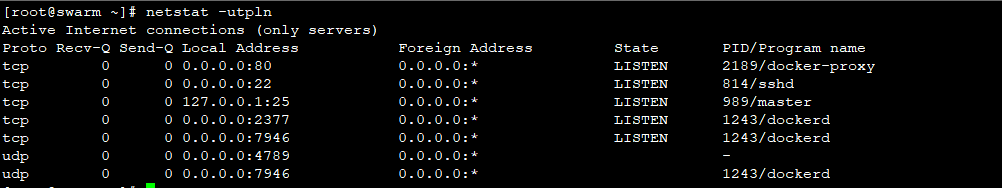
查看集群详细信息：



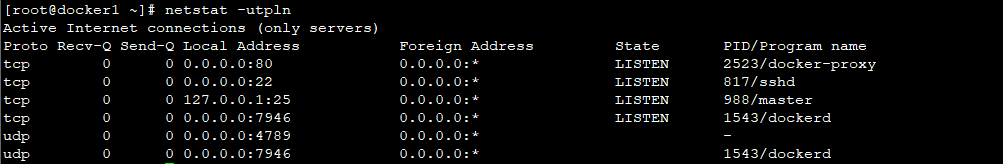
查看集群节点数量：

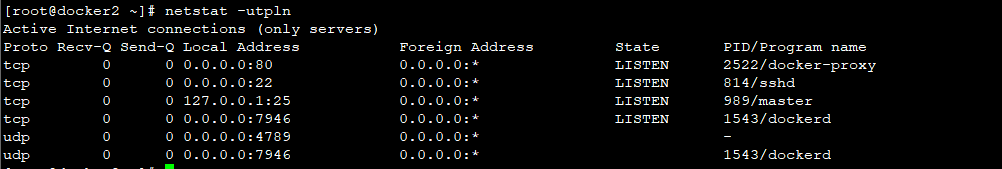


查看swarm节点端口监听状态：



查看docker1和docker2节点的端口监听状态：





注解：Docker service 常用命令：

docker service create ##部署服务

docker service inspect ##查看服务详情

docker service logs ##查看某个服务日志

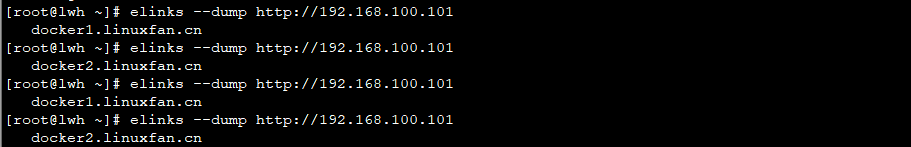
docker service ls ##查看所有服务详情

docker service rm ##删除某个服务（-f 强制删除）

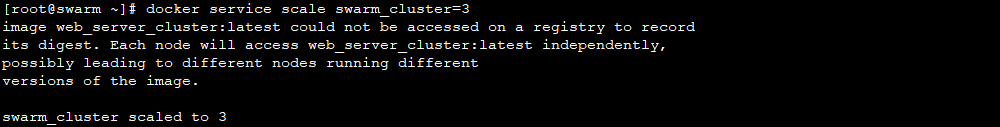
docker service scale ##设置某个服务的个数

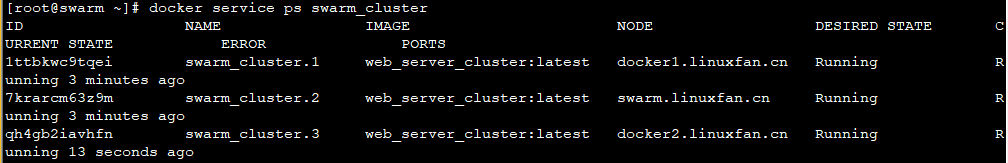
docker service update ##更新某个服务

* **客户端节点访问测试swarm集群：**

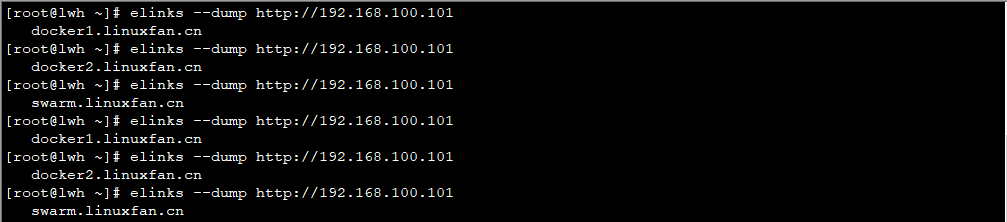


* **添加swarm节点到集群中作为node节点：**



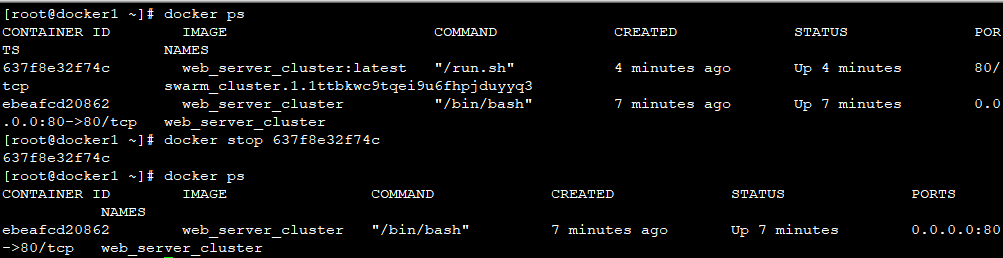


* **客户端节点访问测试swarm集群：**

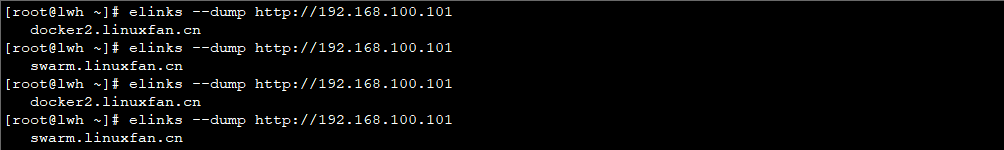


* **关闭docker1节点的容器；**

在关闭docker1节点的容器的时候，发现除手动启动的一个http容器之外，又出现了一个web\_server\_cluster:latest容器，此容器就是docker swarm容器，swarm集群技术及来源于容器又基于容器；



* **客户端节点访问测试swarm集群：**

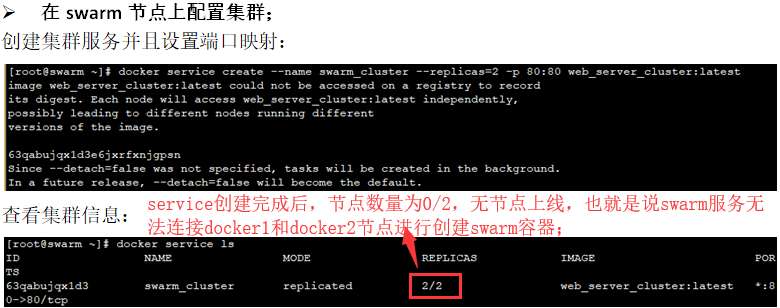


**四、扩展学习：**

**扩展一：**退出swarm集群用的命令是docker swarm leave，然而存在这样一种情况：没有完全退出swarm集群时就关掉了Docker服务。随后网络环境变化了（主机的IP变了）。此时若再systemctl start docker将会报错，通过systemctl status docker -l可以查看完整的报错信息，提示找不到老IP地址。其实这是启动swarm时报的错误；

比较方便的解决办法是手动修改/var/lib/docker/swarm下面的docker-state.json和state.json两个文件。把这两个json文件中原来的老地址都改成现在的新地址。应该就可以顺利启动了；

**扩展二：在创建swarm集群service时，出现的错误：**







解决：经如下方法，查看swarm的service日志，发现出现容器中的httpd服务启动问题，此问题是由于在docker1和docker2的http容器镜像的Dockerfile最下边写的是CMD ["/usr/sbin/httpd"]，那么swarm服务在连接容器时，出现httpd启动异常，则连接终止，最终导致swarm服务无法再docker1和docker2节点上启动swarm相关的容器，所以出现无法启动；



http://dockone.io/article/662

http://blog.daocloud.io/swarm\_analysis\_part1/

**Docker云厂商：**

刀云：http://www.daocloud.io/

灵雀云：http://www.alauda.cn/

希云：https://csphere.cn/training

网易蜂巢：https://c.163.com/product/service

有容云：http://www.youruncloud.com/

腾讯云：https://www.qcloud.com/community/article/266172