Docker Swarm

Docker Swarm是一个原生的Docker集群管理工具，将一组Docker主机作为一个虚拟的Docker主机来管理，对外提供标准Docker API的形式提供服务。可以使用Docker来启动一个Swarn Container，然后管理Docker集群。Docker Swarm也是一个容器编排工具，和Docker Compose不同的是Compose是在单个服务器或者主机上创建多个容器，而Docker Swarm则在多个服务器或者主机上创建容器集群服务。

# Swarm的安装

1. 配置Docker Engine

所有节点，配置/etc/docker/daemon.json，不存在则创建该文件，文件中配置：

*{*

*"hosts": ["tcp://0.0.0.0:2375", "unix:///var/run/docker.sock"]*

*}*

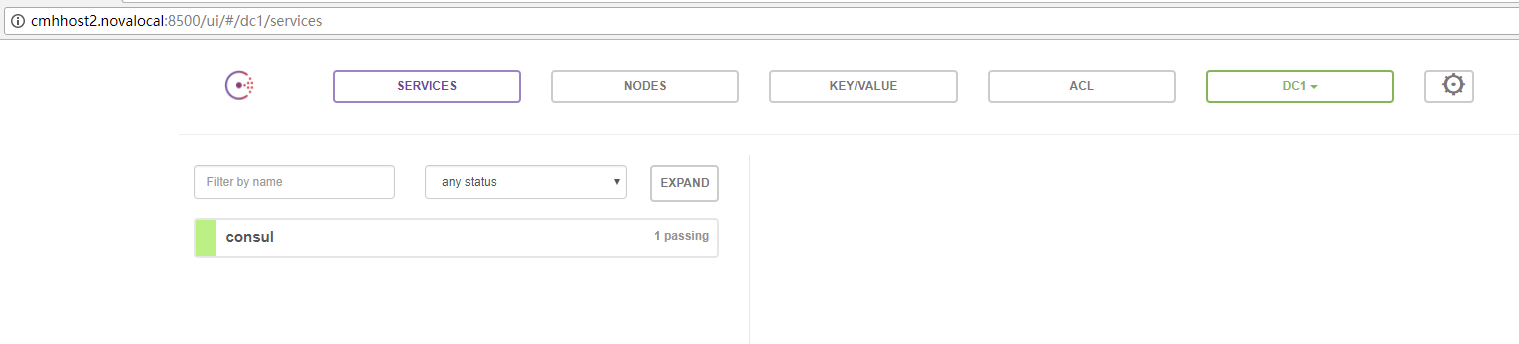
重启docker.

*$systemctl restart docker*

1. 启动集群discovery backend，使用consul作为Discovery，选择一个节点运行命令如下：

*docker run -d -p 8500:8500 --name=consul progrium/consul -server -bootstrap -advertise=10.139.4.83*

打开页面如下：

3）在所有节点中获取docker swarm镜像

*# docker pull swarm*

执行成功后，查看镜像如下：

*# docker images swarm*

*REPOSITORY TAG IMAGE ID CREATED SIZE*

*swarm latest 59c0df55980b 2 months ago 15.8MB*

4）创建Swarm Manager，执行命令如下：

*docker run -d -p 4000:4000 swarm manage -H :4000 --advertise 10.139.4.94:4000 consul://10.139.4.83:8500*

查看container如下：

*# docker ps*

*CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES*

*020e47abfd5b swarm "/swarm manage -H :4…" 38 minutes ago Up 37 minutes 2375/tcp, 0.0.0.0:4000->4000/tcp heuristic\_goodall*

5）添加两个Swarm Worker，分别在对应的主机上，执行命令如下：

*docker run -d swarm join --advertise=10.139.4.82:2375 consul://10.139.4.83:8500*

*docker run -d swarm join --advertise=10.139.4.83:2375 consul://10.139.4.83:8500*

6）在Swarm Manager中查询Info，命令如下：

*# docker -H :4000 info*

*Nodes: 2*

*cmhhost1.novalocal: 10.139.4.82:2375*

*└ ID: WT6J:K4QG:3RFV:YNZM:WZG4:TVKW:ULEL:4HZW:JEDM:GB3F:VPSA:ELEB|10.139.4.82:2375*

*└ Status: Healthy*

*└ Containers: 6 (1 Running, 0 Paused, 5 Stopped)*

*└ Reserved CPUs: 0 / 8*

*└ Reserved Memory: 0 B / 16.29 GiB*

*└ Labels: kernelversion=3.10.0-123.20.1.el7.x86\_64, operatingsystem=CentOS Linux 7 (Core), ostype=linux, storagedriver=devicemapper*

*└ UpdatedAt: 2018-01-29T10:32:45Z*

*└ ServerVersion: 17.12.0-ce*

*cmhhost2.novalocal: 10.139.4.83:2375*

*└ ID: WQVD:NGE3:XRXS:KAHL:SN6V:SECA:TK5K:SJJQ:7OPJ:6AMM:CDO7:JRNF|10.139.4.83:2375*

*└ Status: Healthy*

*└ Containers: 5 (2 Running, 0 Paused, 3 Stopped)*

*└ Reserved CPUs: 0 / 8*

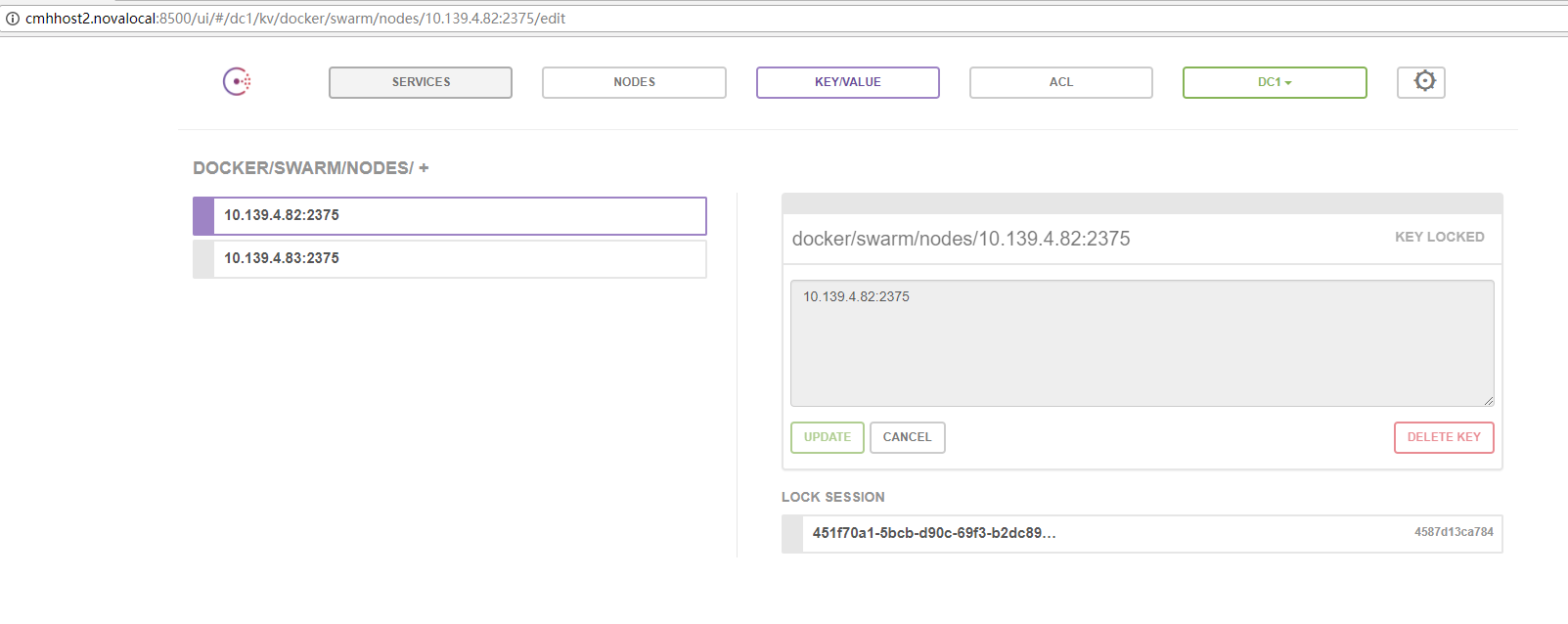
*└ Reserved Memory: 0 B / 16.29 GiB*

*└ Labels: kernelversion=3.10.0-123.20.1.el7.x86\_64, operatingsystem=CentOS Linux 7 (Core), ostype=linux, storagedriver=devicemapper*

*└ UpdatedAt: 2018-01-29T10:32:59Z*

*└ ServerVersion: 17.12.0-ce*

7）Consul中查看如下：



8）通过Swarm启动应用，测试命令如下：

*[root@cmhhost3 docker]# docker -H :4000 run hello-world*

*Hello from Docker!*

*This message shows that your installation appears to be working correctly.*

查看swarm worker节点，命令如下：

*[root@cmhhost1 docker]# docker ps -a*

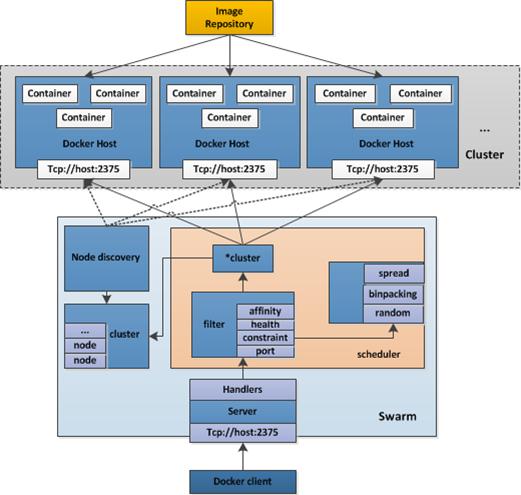
*CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES*

*1d5ea7b1008d hello-world "/hello" 42 minutes ago Exited (0) 42 minutes ago hungry\_minsky*

在该节点中执行了hello-world docker应用

# Swarm系统架构和概念

Swarm是Docker原生的容器编排工具，使用标准的Docker API和网络，系统架构如下图所示：



Swarm由管理器和运行服务的工作节点组成：

1. 管理器，在集群中分配任务，协调众多构成集群的工作节点
2. 工作节点，运行管理器分配的Docker容器
3. 服务，在集群中运行的某组特定的Docker容器接口
4. 任务，特定服务所需要的命令及运行镜像的Docker容器

Swarm设置简单，使用Docker Engine来启动主节点，并可提供一个可在每个工作节点上使用的命令将这些节点添加到集群。集群一旦运行，可使用Docker compose指定服务，启动后，服务部署在集群的各个主机上。

# Swarm高级特性

## 3.1 Swarm过滤器

过滤器是告知Swarm优先在哪个节点上运行容器：

* 约束过滤器，contraint filter
* 亲和过滤器，affinity filter
* 依赖过滤器，dependency filter
* 端口过滤器，port filter
* 健康过滤器，health filter

1. 约束过滤器

依赖于用户给各个节点赋予的标签，例如用户为使用特殊存储类型或者指定操作系统的节点来分组。在启动Docker Daemon进程时，设置键值对标签label，示例如下：

*docker -H :4000 run –e contraints:datacenter==user-east1 –d ==name www-user1 –p 80 nginx*

1. 亲和过滤器

让容器运行更互相接近，比如让容器web1挨着haproxy1容器或者挨着指定ID的容器运行，示例如下：

*docker -H :4000 run –d ==name www-user1 –e affinity:container==www-east1 nginx*

告诉这个容器在www-east1容器所在的Swarm节点上启动。

*-e affinity:container!=www-east1*

让容器不在www-east1所在的Swarm节点上启动

1. 依赖过滤器

在具备卷或容器连接的节点上启动容器

1. 端口过滤器

通过网络端口进行调度，在具有指定端口可用的节点上启动容器，示例如下：

*docker -H :4000 run –d ==name www-user1 -p 80:80 nginx*

5) 健康过滤器

Swarm不会在将容器调度到认为不健康的节点上。

## 3.2 调度策略

调度策略运行用户用集群节点更隐式的特性来对容器进行调度，比如该节点可用资源的数量等。只在拥有足够内存或者CPU的节点上启动容器。Docker Swarm有三种策略：平铺策略（Spread）、紧凑策略（BinPacking）和随机策略（Random）。但是只有平铺策略和紧凑策略才真正的策略。默认策略是平铺策略

在执行swarm manage命令是，通过—strategy标志设置用户想选用的策略

1. 平铺策略，选在已运行容器数量最少的节点，让所有容器比较平均的分配到集群中的每个节点上
2. 紧凑策略，根据节点上可用的CPU和内存资源为节点打分，先返回使用最紧凑的节点。保证节点最大程度被使用，避免碎片化，并确保在需要启动更大的容器时有最大数量的空间可用
3. 随机策略，随机选择一个随机的节点运行容器，主要用于调试中