链滴

基于 docker 搭建 hadoop 跨主机集群

作者：Arthur
原文链接：https://ld246.com/article/1508232710946
来源网站：链滴

作者：Arthur 联系方式：ArthurHappy@qq.com 博客：blog.arthurpapa.cn

摘要：本文是基于docker 17.09.0搭建的hadoop 2.7.2 分布式跨主机集群，集群规模为一个maste和两个salve，一共使用三台物理主机（两台或者多台物理机均可模拟），集群网络使用的是docker warm搭建。

备注：中文社区中相关资料极少，相关资料请直接翻阅官方文档

环境介绍

环境要求
1. 操作系统：centos7
2. 物理机（或虚拟机）：两台或两台以上，本文使用三台
3. 网络：物理机需在同一子网中（最好给它分配固定ip），验证方式：互相ping是否能够ping通
4. 均安装了docker 17.X 版本（要求支持swarm）

本文环境介绍
1. 物理机

hostname	ip
master	192.168.0.150
slave1	192.168.0.151
slave2	192.168.0.152

以下为具体配置过程

1. 下载hadoop镜像

master 下下载镜像。slave机子不需要下载，会在后续swarm网络中自动分发。

本文使用的hadoop镜像为 kiwenlau/hadoop:1.0，相关介绍点此处。

```
[root@master ~]#  docker pull kiwenlau/hadoop:1.0
```

2. 建立Swarm网络

在master上运行以下命令，建立以master为主节点的swarm网络。注意，ip改为你自己网络中maste的ip。

```
[root@master ~]# docker swarm init --advertise-addr 192.168.1.150
```

运行之后会有如下信息提示

```
Swarm initialized: current node (ytxpz5xrzujbkmil0geiacepr) is now a manager.

To add a worker to this swarm, run the following command:

docker swarm join --token SWMTKN-1-3fbqppvsbpkg461lkeddkwjtng8ee0ufnuvfq1h8zzmhv
```

```
v54x-d4hdzdb997g6ljv4lga0r3xro 192.168.1.150:237

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instr
ctions.
```

3. 子节点加入Swarm网络

进入slave1中，运行如下命令：

```
[root@slave1 ~]# docker swarm join --token SWMTKN-1-3fbqppvsbpkg461lkeddkwjtng8ee0
fnuvfq1h8zzmhv3v54x-d4hdzdb997g6ljv4lga0r3xro 192.168.1.150:237
```

同样进入slave2中，运行相同命令

```
[root@slave2 ~]# docker swarm join --token SWMTKN-1-3fbqppvsbpkg461lkeddkwjtng8ee0
fnuvfq1h8zzmhv3v54x-d4hdzdb997g6ljv4lga0r3xro 192.168.1.150:237
```

这样，节点slave1 slave2就加入了master的swarm网络了。其中运行的命令即为第二步中创建完网提示的信息。

4. 创建一个专用网络

```
## 运行如下命令，查看现由网络。可以看到只有一个swarm网络ingress。这个是默认网络，我们一
不会直接使用
[root@master ~]# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
8b78fc47d7fb        bridge              bridge              local
9570aad7f0e0        docker_gwbridge     bridge              local
cf0d62f00408        host                host                local
xdelmzi55ifr        ingress             overlay             swarm
90c51fe9392a        none                null                local

## 创建一个专用网络
[root@master ~]# docker network create --opt encrypted --driver overlay --attachable hado
p

## 创建完成后可以看到多出了一个hadoop网络
[root@master ~]# docker network ls
8b78fc47d7fb        bridge              bridge              local
9570aad7f0e0        docker_gwbridge     bridge              local
iv1irku3ohf7        hadoop              overlay             swarm
cf0d62f00408        host                host                local
xdelmzi55ifr        ingress             overlay             swarm
90c51fe9392a        none                null                local
```

5. 启动容器服务

```
[root@master ~]# docker service create -t --name hadoop-master --hostname hadoop-maste
 --network hadoop --detach=false --replicas 1 --publish mode=host,target=8088,published=
```

```
088,protocol=tcp --publish mode=host,target=50070,published=50070,protocol=tcp kiwenla
/hadoop:1.0
[root@master ~]# docker service create -t --name hadoop-slave1 --hostname hadoop-slave1
--network hadoop --detach=false --replicas 1 kiwenlau/hadoop:1.0
[root@master ~]# docker service create -t --name hadoop-slave2 --hostname hadoop-slave2
--network hadoop --detach=false --replicas 1 kiwenlau/hadoop:1.0
## 启动成功后查看结果

[root@master ~]# docker service ls
ID              NAME              MODE              REPLICAS          IMAGE              PORTS
p79vetbllisg    hadoop-master     replicated        1/1               kiwenlau/hadoop:1.0
8d9ll46cf7p2    hadoop-slave1     replicated        1/1               kiwenlau/hadoop:1.0
ghp3ffhnlbf3    hadoop-slave2     replicated        1/1               kiwenlau/hadoop:1.0

## 查看服务情况 可以观察到三个服务已经被分配到了不同的服务器上。这样容器就启动成功了。
[root@master ~]# docker service ps p79vetbllisg 8d9ll46cf7p2 ghp3ffhnlbf3
ID              NAME              IMAGE              NODE              DESIRED STATE       CURRE
T STATE        ERROR              PORTS
oml7l4xt0ay7    hadoop-slave2.1    kiwenlau/hadoop:1.0    slave2              Running
Running 1 minutes ago
gva3i0ufqund    hadoop-slave1.1    kiwenlau/hadoop:1.0    slave1              Running
 Running 1 minutes ago
1rn1f8jwv8gp    hadoop-master.1    kiwenlau/hadoop:1.0    msater              Running
  Running 1 minutes ago                *:8088->8088/tcp,*:50070->50070/tcp
```

6. 启动hadoop

观察上一步我们发现， hadoop-master容器启动在master主机上。我们进入到master。

```
[root@master ~]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
PORTS                                      NAMES
8282e646d559        kiwenlau/hadoop:1.0   "sh -c 'service ss..."   2 hours ago         Up 2 hours
    0.0.0.0:8088->8088/tcp, 0.0.0.0:50070->50070/tcp   hadoop-master.1.1rn1f8jwv8gpmrb5i6
ijysff
##########  进入容器
[root@master ~]# docker exec -it 8282e646d559 bash
##########  进入容器后的当前目录下可以看到已经有如下脚本
root@hadoop-master:~# ls
hdfs  input  run-wordcount.sh  start-hadoop.sh
##########  ping 下两个容器，看看是否能够ping通
root@hadoop-master:~# ping hadoop-slave1
root@hadoop-master:~# ping hadoop-slave2

########## 分别测试两个容器ssh是否正常，能够正常ssh登录则表明正常。 如果出现time out异
，请查看本文最后的异常分析。
root@hadoop-master:~# ssh hadoop-slave1
root@hadoop-slave1:~# exit
root@hadoop-master:~# ssh hadoop-slave2
root@hadoop-slave2:~# exit

########## 启动hadoop
root@hadoop-master:~# ./start-hadoop.sh
```

```
Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,10.0.0.3' (ECDSA) to the list of
nown hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namen
de-hadoop-master.out
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,10.0.0.6' (ECDSA) to the list of k
own hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,10.0.0.4' (ECDSA) to the list of k
own hosts.
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode
hadoop-slave1.out
hadoop-slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode
hadoop-slave2.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secon
arynamenode-hadoop-master.out

starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-hadoo
-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,10.0.0.4' (ECDSA) to the list of k
own hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,10.0.0.6' (ECDSA) to the list of k
own hosts.
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodema
ager-hadoop-slave1.out
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodema
ager-hadoop-slave2.out

########## 跑个word count测试下
root@hadoop-master:~# ./run-wordcount.sh
17/10/17 04:17:37 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/
0.0.0.3:8032

....

input file1.txt:
Hello Hadoop

input file2.txt:
Hello Docker

wordcount output:
Docker    1
Hadoop    1
Hello    2
```

自此，使用docker的跨主机的hadoop集群搭建完成。

参考：Docker swarm mode overlay network security model

异常分析

1. swarm网络中的容器能够ping通但是无法互相ssh登录。

问题描述：笔者在搭建过程中碰到了这个问题，docker容器hadoop-master和hadoop-slave1,hadoop-slave2在一个swarm网络中，能够互相ping通，但是在ssh登录的时候出现connection time out常，等了很久最后连接超时，也没有报其他问题。笔者在碰到这个问题的时候，找到的原因是物理主slave1,slave2的防火墙没有关，直接截拦了对容器内部的ssh访问。

解决方案：

```
######进入slave1, slave2物理主机，都运行如下两条命令
# 关闭防火墙
[root@slave1 ~]# systemctl stop firewalld.service
# 关闭开机自启防火墙
[root@slave1 ~]# systemctl disable firewalld.service
```