|  |
| --- |
| 入户走访信息采集系统 |
| 安装部署手册 |

版权说明：

此文档的版权归中国电子科学研究院所有，作为本系统的最终用户，可以拥有该份文档的使用权，但未征得中国电子科学研究院的书面批准，不得向第三方借阅、出让、出版该文档。

文档属性

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称： |  | | |
| 项目名称： |  | | |
| 版本号： |  | 文档编号： |  |
| 撰写者： |  | 日期： |  |
| 复核者： |  | 日期： |  |
| 负责者： |  | 日期： |  |
| 审批者： |  | 日期： |  |

版本历史：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 修改内容 | 撰写者 | 发布日期 |
|  |  |  |  |

目 录

[项目安装部署文档 1](#_Toc516607465)

[1. 引言 3](#_Toc516607466)

[1.1. 编写目的 3](#_Toc516607467)

[1.2. 范围 3](#_Toc516607468)

[2. 项目安装部署 3](#_Toc516607469)

[2.1. 外网服务安装部署 3](#_Toc516607470)

[2.1.1. Docker服务安装 3](#_Toc516607471)

[2.1.2. FastDFS集群安装 3](#_Toc516607472)

[2.1.3. redis与sentinel集群安装 16](#_Toc516607473)

[2.1.4. Mysql数据库安装 17](#_Toc516607474)

[3.1.5. ftp服务安装 19](#_Toc516607475)

[3.1.6 Zookeeper集群安装(3个节点) 20](#_Toc516607476)

[3.1.7 kafka安装 21](#_Toc516607477)

[3.1.8 TCP服务安装 21](#_Toc516607478)

[3.1.9 cetc服务 22](#_Toc516607479)

[3.1.10 grxx-sync服务 22](#_Toc516607480)

[3.1.11 nginx 服务安装 24](#_Toc516607481)

[3.1.12 keepalived服务安装 25](#_Toc516607482)

[2.2. 内网服务安装部署 29](#_Toc516607483)

[3.2.1.docker服务安装 29](#_Toc516607484)

[3.2.2.FastDFS集群安装 29](#_Toc516607485)

[3.2.3. redis与sentinel集群安装 31](#_Toc516607486)

[3.2.4.ftp服务安装 32](#_Toc516607487)

[3.2.5.nginx安装 33](#_Toc516607488)

[3.2.6.grxx-sync程序服务安装 33](#_Toc516607489)

# 引言

## 编写目的

本文档将帮助入户走访信息采集系统的运维人员了解内网和外网服务器集群安装、部署全过程，包括Docker服务安装、FastDFS集群安装、Mysql数据库安装及各种服务部署过程，帮助服务器的运维人员进行服务器迁移和运维管理工作。

## 范围

本文档描述了入户走访信息采集系统软件平台的内网和外网服务器集群安装、部署全过程。

# 项目安装部署

## 外网服务安装部署

### Docker服务安装

1. **分别进入内网ip是192.168.30.7, 192.168.30.8, 192.168.30.9, 192.168.30.11，192.168.30.52，192.168.30.53，192.168.30.54, 192.168.30.73, 192.168.30.74, 192.168.30.75, 192.168.30.76 的11台服务器执行以下步骤安装服务**
2. **在线安装**

**输入yum -y install docker-io指令在线安装,安装过程根据网络快慢需要一段时间完成。**

1. **启动docker服务。**

**systemctl start docker.service**

1. **将docker服务加入开机启动项**

**systemctl enable docker.service**

### FastDFS集群安装

#### 3.1.2.1电信fastdfs安装

1.进入部署包的FastDFS文件夹中，修改mod\_fastdfs.conf跟storage.conf文件中的参数（修改为计划安装为tracker服务器的公网IP和端口）修改参数如下：

tracker\_server=220.171.11.200:22522

tracker\_server=220.171.11.200:22622

tracker\_server=220.171.11.200:22722

tracker\_server=220.171.11.200:22822

tracker\_server=220.171.11.200:22922

tracker\_server=220.171.11.201:22522

tracker\_server=220.171.11.201:22622

tracker\_server=220.171.11.201:22722

tracker\_server=220.171.11.201:22822

tracker\_server=220.171.11.201:22922

2. 分别上传FastDFS的整个文件夹目录到ip为220.171.11.200, 220.171.11.201, 220.171.11.202, 220.171.11.204四台服务器的/opt路径下，然后分别进入到四台服务器的FastDFS文件中，安装fastdfs镜像,此安装过程需要几分钟，安装fastdfs镜像指令是：docker build -t fastdfs --rm=true .

3. fastdfs镜像安装完后,先启动tracker，后启动storage

4. 在ip为220.171.11.200, 220.171.11.201的两台服务器中，每台启动5个tracker服务，如下指令所示：

docker run -d --name tracker1 -v /home/tracker/data01:/fastdfs/tracker/data --net=host -e TR\_PORT=22522 fastdfs tracker

docker run -d --name tracker2 -v /home/tracker/data02:/fastdfs/tracker/data --net=host -e TR\_PORT=22622 fastdfs tracker

docker run -d --name tracker3 -v /home/tracker/data03:/fastdfs/tracker/data --net=host -e TR\_PORT=22722 fastdfs tracker

docker run -d --name tracker4 -v /home/tracker/data04:/fastdfs/tracker/data --net=host -e TR\_PORT=22822 fastdfs tracker

docker run -d --name tracker5 -v /home/tracker/data05:/fastdfs/tracker/data --net=host -e TR\_PORT=22922 fastdfs tracker

5. 在ip为220.171.11.202, 220.171.11.204的两台服务器中，每台启动20个storage服务，如下指令所示：**\*\*\*\*\*\*注意以下每个组都是一行指令\*\*\*\*\*\*\***

storage1组:

**docker run -d --name storage1new -v /home/storage/data01:/fastdfs/storage/data -v /home/storage/store\_path01:/fastdfs/store\_path --net=host -e ST\_PORT=23001 -e NGX\_PORT=3001 -e GROUP\_NAME=group1 fastdfs storage**

**storage2组：**

**docker run -d --name storage2new -v /home/storage/data02:/fastdfs/storage/data -v /home/storage/store\_path02:/fastdfs/store\_path --net=host -e ST\_PORT=23002 -e NGX\_PORT=3002 -e GROUP\_NAME=group2 fastdfs storage**

**storage3组：**

**docker run -d --name storage3new -v /home/storage/data03:/fastdfs/storage/data -v /home/storage/store\_path03:/fastdfs/store\_path --net=host -e ST\_PORT=23003 -e NGX\_PORT=3003 -e GROUP\_NAME=group3 fastdfs storage**

**storage4组：**

**docker run -d --name storage4new -v /home/storage/data04:/fastdfs/storage/data -v /home/storage/store\_path04:/fastdfs/store\_path --net=host -e ST\_PORT=23004 -e NGX\_PORT=3004 -e GROUP\_NAME=group4 fastdfs storage**

**storage5组：**

**docker run -d --name storage5new -v /home/storage/data05:/fastdfs/storage/data -v /home/storage/store\_path05:/fastdfs/store\_path --net=host -e ST\_PORT=23005 -e NGX\_PORT=3005 -e GROUP\_NAME=group5 fastdfs storage**

**storage6组：**

**docker run -d --name storage6new -v /home/storage/data06:/fastdfs/storage/data -v /home/storage/store\_path06:/fastdfs/store\_path --net=host -e ST\_PORT=23006 -e NGX\_PORT=3006 -e GROUP\_NAME=group6 fastdfs storage**

**storage7组：**

**docker run -d --name storage7new -v /home/storage/data07:/fastdfs/storage/data -v /home/storage/store\_path07:/fastdfs/store\_path --net=host -e ST\_PORT=23007 -e NGX\_PORT=3007 -e GROUP\_NAME=group7 fastdfs storage**

**storage8组：**

**docker run -d --name storage8new -v /home/storage/data08:/fastdfs/storage/data -v /home/storage/store\_path08:/fastdfs/store\_path --net=host -e ST\_PORT=23008 -e NGX\_PORT=3008 -e GROUP\_NAME=group8 fastdfs storage**

**storage9组：**

**docker run -d --name storage9new -v /home/storage/data09:/fastdfs/storage/data -v /home/storage/store\_path09:/fastdfs/store\_path --net=host -e ST\_PORT=23009 -e NGX\_PORT=3009 -e GROUP\_NAME=group9 fastdfs storage**

**storage10组：**

**docker run -d --name storage10new -v /home/storage/data10:/fastdfs/storage/data -v /home/storage/store\_path10:/fastdfs/store\_path --net=host -e ST\_PORT=23010 -e NGX\_PORT=3010 -e GROUP\_NAME=group10fastdfs storage**

**storage11组：**

**docker run -d --name storage11new -v /home/storage/data11:/fastdfs/storage/data -v /home/storage/store\_path11:/fastdfs/store\_path --net=host -e ST\_PORT=23011 -e NGX\_PORT=3011 -e GROUP\_NAME=group11 fastdfs storage**

**storage12组：**

**docker run -d --name storage12new -v /home/storage/data12:/fastdfs/storage/data -v /home/storage/store\_path12:/fastdfs/store\_path --net=host -e ST\_PORT=23012 -e NGX\_PORT=3012 -e GROUP\_NAME=group12 fastdfs storage**

**storage13组：**

**docker run -d --name storage13new -v /home/storage/data13:/fastdfs/storage/data -v /home/storage/store\_path13:/fastdfs/store\_path --net=host -e ST\_PORT=23013 -e NGX\_PORT=3013 -e GROUP\_NAME=group13 fastdfs storage**

**storage14组：**

**docker run -d --name storage3new -v /home/storage/data14:/fastdfs/storage/data -v /home/storage/store\_path14:/fastdfs/store\_path --net=host -e ST\_PORT=23014 -e NGX\_PORT=3014 -e GROUP\_NAME=group14 fastdfs storage**

**storage15组：**

**docker run -d --name storage15new -v /home/storage/data15:/fastdfs/storage/data -v /home/storage/store\_path15:/fastdfs/store\_path --net=host -e ST\_PORT=23015 -e NGX\_PORT=3015 -e GROUP\_NAME=group15fastdfs storage**

**storage16组：**

**docker run -d --name storage16new -v /home/storage/data16:/fastdfs/storage/data -v /home/storage/store\_path16:/fastdfs/store\_path --net=host -e ST\_PORT=23016 -e NGX\_PORT=3016 -e GROUP\_NAME=group16 fastdfs storage**

**storage17组：**

**docker run -d --name storage17new -v /home/storage/data17:/fastdfs/storage/data -v /home/storage/store\_path17:/fastdfs/store\_path --net=host -e ST\_PORT=23017 -e NGX\_PORT=3017 -e GROUP\_NAME=group17 fastdfs storage**

**storage18组：**

**docker run -d --name storage18new -v /home/storage/data18:/fastdfs/storage/data -v /home/storage/store\_path18:/fastdfs/store\_path --net=host -e ST\_PORT=23018 -e NGX\_PORT=3018 -e GROUP\_NAME=group18 fastdfs storage**

**storage19组：**

**docker run -d --name storage19new -v /home/storage/data19:/fastdfs/storage/data -v /home/storage/store\_path19:/fastdfs/store\_path --net=host -e ST\_PORT=23019 -e NGX\_PORT=3019 -e GROUP\_NAME=group19 fastdfs storage**

storage20组：

**docker run -d --name storage20new -v /home/storage/data20:/fastdfs/storage/data -v /home/storage/store\_path20:/fastdfs/store\_path --net=host -e ST\_PORT=23020 -e NGX\_PORT=3020 -e GROUP\_NAME=group20 fastdfs storage**

#### 3.1.2.2联通fastdfs安装

1.进入部署包的FastDFS文件夹中，修改mod\_fastdfs.conf跟storage.conf文件中的参数（修改为计划安装为tracker服务器的公网IP和端口）修改参数如下：

tracker\_server=124.88.86.244:22522

tracker\_server=124.88.86.244:22622

tracker\_server=124.88.86.244:22722

tracker\_server=124.88.86.244:22822

tracker\_server=124.88.86.244:22922

2. 分别上传FastDFS的整个文件夹目录到ip为124.88.86.244, 124.88.86.245,两台服务器的/home/packages路径下，然后分别进入到两台服务器的FastDFS文件中，安装fastdfs镜像,此安装过程需要几分钟，安装fastdfs镜像指令是：

docker build -t fastdfs --rm=true .

3. fastdfs镜像安装完后,先启动tracker，后启动storage

4. 在ip为124.88.86.244的服务器中，启动5个tracker服务，如下指令所示：

docker run -d --name tracker6 -v /home/tracker/data6:/fastdfs/tracker/data --net=host -e TR\_PORT=22522 fastdfs tracker

docker run -d --name tracker7 -v /home/tracker/data7:/fastdfs/tracker/data --net=host -e TR\_PORT=22622 fastdfs tracker

docker run -d --name tracker8-v /home/tracker/data8:/fastdfs/tracker/data --net=host -e TR\_PORT=22722 fastdfs tracker

docker run -d --name tracker9-v /home/tracker/data9:/fastdfs/tracker/data --net=host -e TR\_PORT=22822 fastdfs tracker

docker run -d --name tracker10 -v /home/tracker/data10:/fastdfs/tracker/data --net=host -e TR\_PORT=22922 fastdfs tracker

5. 在ip为124.88.86.245的服务器中，启动20个storage服务，如下指令所示：

**\*\*\*\*\*\*注意以下每个组都是一行指令\*\*\*\*\*\*\*\*\*\*\*\***

storage21组:

**docker run -d --name storage21 -v /home/storage/data21:/fastdfs/storage/data -v /home/storage/store\_path21:/fastdfs/store\_path --net=host -e ST\_PORT=23001 -e NGX\_PORT=3001 -e GROUP\_NAME=group21 fastdfs storage**

storage22组:

**docker run -d --name storage22 -v /home/storage/data22:/fastdfs/storage/data -v /home/storage/store\_path22:/fastdfs/store\_path --net=host -e ST\_PORT=23002 -e NGX\_PORT=3002 -e GROUP\_NAME=group22 fastdfs storage**

storage23组:

**docker run -d --name storage23 -v /home/storage/data23:/fastdfs/storage/data -v /home/storage/store\_path23:/fastdfs/store\_path --net=host -e ST\_PORT=23003 -e NGX\_PORT=3003-e GROUP\_NAME=group23 fastdfs storage**

storage24组:

**docker run -d --name storage24 -v /home/storage/data24:/fastdfs/storage/data -v /home/storage/store\_path24:/fastdfs/store\_path --net=host -e ST\_PORT=23004 -e NGX\_PORT=3004 -e GROUP\_NAME=group24 fastdfs storage**

storage25组:

**docker run -d --name storage25 -v /home/storage/data25:/fastdfs/storage/data -v /home/storage/store\_path25:/fastdfs/store\_path --net=host -e ST\_PORT=23005 -e NGX\_PORT=3005 -e GROUP\_NAME=group25fastdfs storage**

storage26组:

**docker run -d --name storage26 -v /home/storage/data26:/fastdfs/storage/data -v /home/storage/store\_path26:/fastdfs/store\_path --net=host -e ST\_PORT=23006 -e NGX\_PORT=3006 -e GROUP\_NAME=group26 fastdfs storage**

storage27组:

**docker run -d --name storage27 -v /home/storage/data27:/fastdfs/storage/data -v /home/storage/store\_path27:/fastdfs/store\_path --net=host -e ST\_PORT=23007 -e NGX\_PORT=3007 -e GROUP\_NAME=group27 fastdfs storage**

storage28组:

**docker run -d --name storage28 -v /home/storage/data28:/fastdfs/storage/data -v /home/storage/store\_path28:/fastdfs/store\_path --net=host -e ST\_PORT=23008 -e NGX\_PORT=3008 -e GROUP\_NAME=group28 fastdfs storage**

storage29组:

**docker run -d --name storage29 -v /home/storage/data29:/fastdfs/storage/data -v /home/storage/store\_path29:/fastdfs/store\_path --net=host -e ST\_PORT=23009 -e NGX\_PORT=3009 -e GROUP\_NAME=group29fastdfs storage**

storage30组:

**docker run -d --name storage30 -v /home/storage/data30:/fastdfs/storage/data -v /home/storage/store\_path30:/fastdfs/store\_path --net=host -e ST\_PORT=23030 -e NGX\_PORT=3010-e GROUP\_NAME=group30 fastdfs storage**

storage31组:

**docker run -d --name storage31 -v /home/storage/data31:/fastdfs/storage/data -v /home/storage/store\_path31:/fastdfs/store\_path --net=host -e ST\_PORT=23011 -e NGX\_PORT=3011 -e GROUP\_NAME=group31 fastdfs storage**

storage32组:

**docker run -d --name storage32 -v /home/storage/data32:/fastdfs/storage/data -v /home/storage/store\_path32:/fastdfs/store\_path --net=host -e ST\_PORT=23012 -e NGX\_PORT=3012-e GROUP\_NAME=group32 fastdfs storage**

storage33组:

**docker run -d --name storage33 -v /home/storage/data33:/fastdfs/storage/data -v /home/storage/store\_path33:/fastdfs/store\_path --net=host -e ST\_PORT=23013 -e NGX\_PORT=3013-e GROUP\_NAME=group33 fastdfs storage**

storage34组:

**docker run -d --name storage34 -v /home/storage/data34:/fastdfs/storage/data -v /home/storage/store\_path34:/fastdfs/store\_path --net=host -e ST\_PORT=23014 -e NGX\_PORT=3014 -e GROUP\_NAME=group34 fastdfs storage**

storage35组:

**docker run -d --name storage35 -v /home/storage/data35:/fastdfs/storage/data -v /home/storage/store\_path35:/fastdfs/store\_path --net=host -e ST\_PORT=23015 -e NGX\_PORT=3015 -e GROUP\_NAME=group35 fastdfs storage**

storage36组:

**docker run -d --name storage36 -v /home/storage/data36:/fastdfs/storage/data -v /home/storage/store\_path36:/fastdfs/store\_path --net=host -e ST\_PORT=23016 -e NGX\_PORT=3016-e GROUP\_NAME=group36 fastdfs storage**

storage37组:

**docker run -d --name storage**37 **-v /home/storage/data**37**:/fastdfs/storage/data -v /home/storage/store\_path**37**:/fastdfs/store\_path --net=host -e ST\_PORT=23017 -e NGX\_PORT=3017 -e GROUP\_NAME=group**37 **fastdfs storage**

storage38组:

**docker run -d --name storage**38 **-v /home/storage/data**38**:/fastdfs/storage/data -v /home/storage/store\_path**38**:/fastdfs/store\_path --net=host -e ST\_PORT=23018 -e NGX\_PORT=3018-e GROUP\_NAME=group**38 **fastdfs storage**

storage39组:

**docker run -d --name storage**39 **-v /home/storage/data**39**:/fastdfs/storage/data -v /home/storage/store\_path**39**:/fastdfs/store\_path --net=host -e ST\_PORT=23019 -e NGX\_PORT=3019 -e GROUP\_NAME=group**39 **fastdfs storage**

storage40组：

**docker run -d --name storage40 -v /home/storage/data40:/fastdfs/storage/data -v /home/storage/store\_path40:/fastdfs/store\_path --net=host -e ST\_PORT=23020 -e NGX\_PORT=3020 -e GROUP\_NAME=group40 fastdfs storage**

#### 3.1.2.3移动fastdfs安装

1.进入部署包的FastDFS文件夹中，修改mod\_fastdfs.conf跟storage.conf文件中的参数（修改为计划安装为tracker服务器的公网IP和端口）修改参数如下：

tracker\_server=117.146.60.19:22522

tracker\_server=117.146.60.19:22622

tracker\_server=117.146.60.19:22722

tracker\_server=117.146.60.19:22822

tracker\_server=117.146.60.19:22922

2. 分别上传FastDFS的整个文件夹目录到ip为117.146.60.19, 117.146.60.20,两台服务器的/home/packages路径下，然后分别进入到两台服务器的FastDFS文件中，安装fastdfs镜像,此安装过程需要几分钟，安装fastdfs镜像指令是：

docker build -t fastdfs --rm=true .

3. fastdfs镜像安装完后,先启动tracker，后启动storage

4. 在ip为117.146.60.19的服务器中，启动5个tracker服务，如下指令所示：

docker run -d --name tracker11 -v /home/tracker/data11:/fastdfs/tracker/data --net=host -e TR\_PORT=22522 fastdfs tracker

docker run -d --name tracker12 -v /home/tracker/data12:/fastdfs/tracker/data --net=host -e TR\_PORT=22622 fastdfs tracker

docker run -d --name tracker13 -v /home/tracker/data13:/fastdfs/tracker/data --net=host -e TR\_PORT=22722 fastdfs tracker

docker run -d --name tracker14 -v /home/tracker/data14:/fastdfs/tracker/data --net=host -e TR\_PORT=22822 fastdfs tracker

docker run -d --name tracker15 -v /home/tracker/data15:/fastdfs/tracker/data --net=host -e TR\_PORT=22922 fastdfs tracker

5. 在ip为117.146.60.20的服务器中，启动20个storage服务，如下指令所示：

storage41组:

**docker run -d --name storage41 -v /home/storage/data41:/fastdfs/storage/data -v /home/storage/store\_path41:/fastdfs/store\_path --net=host -e ST\_PORT=23001 -e NGX\_PORT=3001 -e GROUP\_NAME=group41 fastdfs storage**

storage42组:

**docker run -d --name storage42 -v /home/storage/data42:/fastdfs/storage/data -v /home/storage/store\_path42:/fastdfs/store\_path --net=host -e ST\_PORT=23002 -e NGX\_PORT=3002 -e GROUP\_NAME=group42 fastdfs storage**

storage43组:

**docker run -d --name storage43 -v /home/storage/data43:/fastdfs/storage/data -v /home/storage/store\_path43:/fastdfs/store\_path --net=host -e ST\_PORT=23003 -e NGX\_PORT=3003 -e GROUP\_NAME=group43 fastdfs storage**

storage44组:

**docker run -d --name storage44 -v /home/storage/data44:/fastdfs/storage/data -v /home/storage/store\_path44:/fastdfs/store\_path --net=host -e ST\_PORT=23004 -e NGX\_PORT=3004 -e GROUP\_NAME=group44fastdfs storage**

storage45组:

**docker run -d --name storage45 -v /home/storage/data45:/fastdfs/storage/data -v /home/storage/store\_path45:/fastdfs/store\_path --net=host -e ST\_PORT=23005 -e NGX\_PORT=3005 -e GROUP\_NAME=group45 fastdfs storage**

storage46组:

**docker run -d --name storage46 -v /home/storage/data46:/fastdfs/storage/data -v /home/storage/store\_path46:/fastdfs/store\_path --net=host -e ST\_PORT=23006 -e NGX\_PORT=3006 -e GROUP\_NAME=group46 fastdfs storage**

storage47组:

**docker run -d --name storage47 -v /home/storage/data47:/fastdfs/storage/data -v /home/storage/store\_path47:/fastdfs/store\_path --net=host -e ST\_PORT=23007 -e NGX\_PORT=3007-e GROUP\_NAME=group47 fastdfs storage**

storage48组:

**docker run -d --name storage48 -v /home/storage/data48:/fastdfs/storage/data -v /home/storage/store\_path48:/fastdfs/store\_path --net=host -e ST\_PORT=23008 -e NGX\_PORT=3008 -e GROUP\_NAME=group48 fastdfs storage**

storage49组:

**docker run -d --name storage49 -v /home/storage/data49:/fastdfs/storage/data -v /home/storage/store\_path49:/fastdfs/store\_path --net=host -e ST\_PORT=23009 -e NGX\_PORT=3009-e GROUP\_NAME=group49 fastdfs storage**

storage50组:

**docker run -d --name storage50 -v /home/storage/data50:/fastdfs/storage/data -v /home/storage/store\_path50:/fastdfs/store\_path --net=host -e ST\_PORT=23010 -e NGX\_PORT=3010 -e GROUP\_NAME=group50 fastdfs storage**

storage51组:

**docker run -d --name storage51 -v /home/storage/data51:/fastdfs/storage/data -v /home/storage/store\_path51:/fastdfs/store\_path --net=host -e ST\_PORT=23011 -e NGX\_PORT=3011 -e GROUP\_NAME=group51 fastdfs storage**

storage52组:

**docker run -d --name storage52 -v /home/storage/data52:/fastdfs/storage/data -v /home/storage/store\_path52:/fastdfs/store\_path --net=host -e ST\_PORT=23012 -e NGX\_PORT=3012 -e GROUP\_NAME=group52 fastdfs storage**

storage53组:

**docker run -d --name storage53 -v /home/storage/data53:/fastdfs/storage/data -v /home/storage/store\_path53:/fastdfs/store\_path --net=host -e ST\_PORT=23013 -e NGX\_PORT=3013 -e GROUP\_NAME=group53 fastdfs storage**

storage54组:

**docker run -d --name storage54 -v /home/storage/data54:/fastdfs/storage/data -v /home/storage/store\_path54:/fastdfs/store\_path --net=host -e ST\_PORT=23014 -e NGX\_PORT=3014 -e GROUP\_NAME=group54 fastdfs storage**

storage55组:

**docker run -d --name storage55 -v /home/storage/data55:/fastdfs/storage/data -v /home/storage/store\_path55:/fastdfs/store\_path --net=host -e ST\_PORT=23015 -e NGX\_PORT=3015 -e GROUP\_NAME=group55fastdfs storage**

storage56组:

**docker run -d --name storage56 -v /home/storage/data56:/fastdfs/storage/data -v /home/storage/store\_path56:/fastdfs/store\_path --net=host -e ST\_PORT=23016 -e NGX\_PORT=3016 -e GROUP\_NAME=group56 fastdfs storage**

storage57组:

**docker run -d --name storage57 -v /home/storage/data57:/fastdfs/storage/data -v /home/storage/store\_path57:/fastdfs/store\_path --net=host -e ST\_PORT=23017 -e NGX\_PORT=3017 -e GROUP\_NAME=group57 fastdfs storage**

storage58组:

**docker run -d --name storage**58 **-v /home/storage/data**58**:/fastdfs/storage/data -v /home/storage/store\_path**58**:/fastdfs/store\_path --net=host -e ST\_PORT=23018 -e NGX\_PORT=3018 -e GROUP\_NAME=group**58 **fastdfs storage**

storage59组:

**docker run -d --name storage**59 **-v /home/storage/data**59**:/fastdfs/storage/data -v /home/storage/store\_path**59**:/fastdfs/store\_path --net=host -e ST\_PORT=23019 -e NGX\_PORT=3019 -e GROUP\_NAME=group**59 **fastdfs storage**

storage60组：

**docker run -d --name storage60 -v /home/storage/data60:/fastdfs/storage/data -v /home/storage/store\_path60:/fastdfs/store\_path --net=host -e ST\_PORT=23020 -e NGX\_PORT=3020 -e GROUP\_NAME=group60 fastdfs storage**

### redis与sentinel集群安装

1. **分别进入内网ip为192.168.30.58，192.168.30.59两台服务器**
2. **下载安装包：wget http://download.redis.io/redis-stable.tar.gz**
3. **解压安装**

**tar -xvzf redis-stable.tar.gz**

**cd redis-stable**

**make**

**make install**

1. **Redis配置修改**

**cp /redis-stable/redis.conf /etc/redis.conf**

**修改redis.conf文件 vim /etc/redis.conf（修改redis.conf文件的以下参数）**

**daemonize yes**

**bind 本机的内网IP + 空格 + 127.0.0.1**

**requirepass redis密码**

**masterauth redids密码**

**如果是slave机器，还需要配置**

**slaveof master的ip +空格+master的端口号**

1. **配置完成后启动redis（先启动master）**

**/usr/local/bin/redis-server /etc/redis.conf**

1. **Sentinel配置修改**

**cp /redis-stable/sentinel.conf /etc/sentinel.conf**

**修改sentinel.conf 文件 vim /etc/sentinel.conf（修改sentinel.conf 文件以下参数）**

**protected-mode no**

**daemonize yes**

**sentinel monitor mymaster [Redis mater的ip] [Redis Master的端口号] 2**

**sentinel down-after-milliseconds mymaster 5000**

**sentinel failover-timeout mymaster 10000**

**sentinel auth-pass mymaster [Redis的密码]**

1. **配置完成后启动Sentinel（启动Sentinel之前先将Redis的所有节点启动）**

**/usr/local/bin/redis-sentinel /etc/sentinel.conf**

### Mysql数据库安装

**1.将部署包中的mysql安装包文件夹下的mysql-community-release-el6-5.noarch.rpm文件上传到内网ip为192.168.30.9，192.168.30.11的服务器中**

**2.分别在ip是192.168.30.9，192.168.30.11两台服务器中执行以下命令安装**

**yum localinstall mysql-community-release-el6-5.noarch.rpm**

**yum install mysql-community-server**

**3.安装完后,分别修改/etc/my.cnf配置文件以下参数**

**[mysqld]**

**character\_set\_server=utf8**

**default-time-zone='+8:00'**

**max\_connections=1024**

**default-character-set=utf8**

**4.启动MySQL并添加开机启动项**

**systemctl start mysqld.service**

**systemctl enable mysqld.service**

**5.命令行连接MySQL：mysql -u root**

**select user,host,password from mysql.user;**

**6.将user不为空的户密码全部设置⼀遍**

**set password for root@'localhost'=password('Inspeeding123456');**

**7.删除user为空的记录**

**delete from mysql.user where user='';**

**8.授权账号,远程登陆**

**grant all privileges on \*.\* to root@"%" identified by 'Inspeeding123456' with grant option;**

**flush privileges;**

**quit;**

**9.master配置**

**vim /etc/my.cnf**

**[mysqld]**

**server\_id=1（两个节点不能⼀样）**

**binlog-ignore-db=mysql**

**log-bin=sibosen-bin**

**binlog\_cache\_size=1M**

**binlog\_format=mixed**

**expire\_logs\_days=7**

**slave\_skip\_errors=1062**

**relay\_log=sibosen-relay-bin**

**log\_slave\_updates=1**

**auto\_increment\_increment=2**

**auto\_increment\_offset=1**

**10.配置完master重启MySQL**

**systemctl restart mysqld.service**

**11.** **配置Slave户**

**命令行登录MySQL：mysql -u root -p**

**grant replication slave, replication client on \*.\* to 'repl'@'slave ip' identified by 'password';**

**flush privileges;**

**quit;**

**12. Slave配置**

**命令登录MySQL：mysql -u root -p**

**输入:** **change master to master\_host='master ip',master\_user='user for slave(ext. repl)',**

**master\_password='password', master\_port=3306, master\_log\_file='sibosen-bin.000004', master\_log\_pos=439, master\_connect\_retry=30;**

**quit;**

**其中master\_log\_file、master\_log\_pos的值通过以下命令在master机器上查看：**

**show master status;**

**13.** **启动slave**

**命令登录MySQL：mysql -u root -p**

**start slave;**

**查看状态**

**show slave status\G;**

**quit;**

### 3.1.5. ftp服务安装

**1.进入内网ip为192.168.30.57的服务器**

**2.下载安装包 yum -y install vsftpd**

**3.安装完成后，修改vsftpd.conf文件以下参数**

**anonymous\_enable=NO**

**chroot\_local\_user=YES**

**allow\_writeable\_chroot=YES**

**pasv\_enable=YES**

**pasv\_min\_port=60000**

**pasv\_max\_port=62000**

**4.创建FTP虚拟宿主帐户**

**mkdir /opt/ftp**

**useradd -d /opt/ftp/ryxx -g ftp -s /sbin/nologin ryxx**

**passwd ryxx**

**chown -R ryxx /opt/ftp**

**chown -R 777 /opt/ftp**

**mkdir /opt/ftp/ryxx/out2in**

**mkdir /opt/ftp/ryxx/in2out**

**cd /opt/ftp**

**chmod -R 777 \***

**5.启动FTP服务并添加开机自启动**

**systemctl start vsftpd.service**

**systemctl enable vsftpd.service**

### 3.1.6 Zookeeper集群安装(3个节点)

**1.分别进入内网ip为192.168.30.52, 192.168.30.53, 192.168.30.54的3台服务器**

**2.下载安装包 wget http://mirror.bit.edu.cn/apache/zookeeper/zookeeper-3.4.10/**

**3.解压安装**

**tar -zxvf zookeeper-3.4.10.tar.gz -C /opt/**

**mkdir /opt/zoologs**

**mkdir /opt/zoostorage**

**cp /opt/zookeeper-3.4.10/conf/zoo\_sample.cfg /opt/zookeeper-3.4.10/conf/zoo.cfg**

4.**配置Zookeeper集群**

**设置severid**

**echo 1 /opt/zoostorage/myid**

**修改配置文件：vim /opt/zookeeper-3.4.10/conf/zoo.cfg以下参数**

**dataDir=/opt/zoostorage**

**dataLogDir=/opt/zoologs**

**maxClientCnxns=1024**

**server.1=192.168.30.52:2888:3888**

**server.2=192.168.30.53:2888:3888**

**server.3=192.168.30.54:2888:3888**

**其中1、2、3对应echo的serverid（int类型）**

**5.** **依次启动Zookeeper**

**/opt/zookeeper-3.4.10/bin/zkServer.sh start**

### 3.1.7 kafka安装

**1.进入内网ip为192.168.30.57的服务器**

**2.下载安装包**

**wget http://apache.fayea.com/kafka/0.11.0.0/kafka\_2.11-0.11.0.0.tgz**

**3.解压配置Kafka**

**tar -zxvf kafka\_2.11-0.11.0.0.tgz -C /opt/**

**vim /opt/kafka\_2.11-0.11.0.0/config/server.properties**

**delete.topic.enable=true**

**log.dirs=/opt/kaf-kalogs**

**zookeeper.connect=192.168.30.52:2181,192.168.30.53:2181,192.168.30.54:2181**

**4.** **启动Kafka（启动前需要先启动Zookeeper集群）**

**/opt/kafka\_2.11-0.11.0.0/bin/kafka-server-start.sh -daemon /opt/kafka\_2.11-0.11.0.0/config/server.properties**

**5.** **创建Topic（该Topic将用于TCP Server,注意以下指令为一行）**

**/opt/kafka\_2.11-0.11.0.0/bin/kafka-topics.sh --create --topic SibosenPushsTopic --replication-factor 1 --partitions 1**

**--zookeeper 192.168.30.52:2181,192.168.30.53:2181,192.168.30.54:2181**

### 3.1.8 TCP服务安装

3.1.8.1.**comet集群安装**

**1.进入内网IP为192.168.30.7与192.168.30.8的两台服务器**

**2.将部署包中的/TCP/comet下的comet,comet-log.xml,comet.conf文件上传到/opt/tcpserver目录下**

3.1.8.2**. logic、router、job安装**

**1. 进入内网ip为192.168.30.57的服务器**

**2. 将部署包中的/TCP/ tcpserver整个文件夹上传到服务器的/opt路径下**

**3. 启动logic、router、Job服务,按照以下先后顺序启动**

**注意：comet启动的时候，需要的在IP为192.168.30.7与192.168.30.8的两台服务器中启动**

**启动router nohup /opt/tcpserver/router -c /opt/tcpserver/router.conf 2>&1 > /opt/tcpserver/logs/router.log &**

**启动logic nohup /opt/tcpserver/logic -c /opt/tcpserver/logic.conf 2>&1 > /opt/tcpserver/logs/logic.log &**

**启动comet nohup /opt/tcpserver/comet -c /opt/tcpserver/comet.conf 2>&1 > /opt/tcpserver/logs/comet.log &**

**启动job nohup /opt/tcpserver/job -c /opt/tcpserver/job.conf 2>&1 > /opt/tcpserver/logs/job.log &**

### 3.1.9 cetc服务

**1.分别进入内网IP为192.168.30.52，192.168.30.53，192.168.30.54的三台服务器的/opt/HTTP路径下**

**2.分别在三台服务器的/opt/HTTP路径下，运行以下指令构建cetc镜像**

**docker build -t web --rm=true .（web是镜像名称）**

**3.分别在三台服务器中,按照以下指令在每台服务器中运行4个cetc服务,运行的服务数量可以根据情况改变**

**docker run -d --name web1 -p 3001:8080 web（web1是服务名称，web是镜像名称）**

**docker run -d --name web2 -p 3002:8080 web（web2是服务名称，web是镜像名称）**

**docker run -d --name web3 -p 3003:8080 web（web3是服务名称，web是镜像名称）**

**docker run -d --name web4 -p 3004:8080 web（web4是服务名称，web是镜像名称）**

**4.查看启动的所有服务**

**docker ps**

**5停止运行的服务**

**docker stop 服务名称**

**6.重启运行的服务**

**docker restart 服务名称**

### 3.1.10 grxx-sync服务

#### 3.1.10.1老版本服务

**1.分别进入**

**内网ip为192.168.30.52服务器的/opt/javapross/old\_server/FtpCacheServer路径下,**

**内网ip为192.168.30.53服务器的/opt/javaprocess/old\_server/SyncDataFromIn路径下,**

**内网ip为192.168.30.54服务器的/opt/javaprocess/old\_server/SyncOutFromInProxy路径下**

**2.进入/opt/javapross/old\_server/FtpCacheServer路径下，按照以下指令构建镜像，启动服务**

**docker build -t fcs --rm=true .（构建镜像, fcs是镜像名称）**

**docker run -d --name FtpCacheServer fcs（启动服务,FtpCacheServer是服务名称）**

**3.进入/opt/javaprocess/old\_server/SyncDataFromIn路径下, 按照以下指令构建镜像，启动服务**

**docker build -t sdfi --rm=true . （构建镜像, sdfi是镜像名称）**

**docker run -d --name SyncDataFromIn1 sdfi （启动服务, SyncDataFromIn1是服务名称）**

**4.进入/opt/javaprocess/old\_server/SyncOutFromInProxy路径下,按照以下指令构建镜像，启动服务**

**docker build -t sofi --rm=true .（构建镜像, sofi是镜像名称）**

**docker run -d –name SyncOutFromInProxy1 sofi （启动服务，SyncOutFromInProxy1是服务名称）**

#### 3.1.10.2新版本服务

**1.进入内网ip为192.168.30.52服务器的/opt/javapross/new\_server/sync-userzone-o2i路径下,按照以下指令构建镜像，启动1个新版本的服务**

**docker build -t userzone --rm=true . （构建镜像, userzone是镜像名称）**

**docker run -d --name usersync userzone （启动服务, usersync是服务名称）**

**2.进入内网ip为192.168.30.53服务器的/opt/javaprocess/new\_server/sync-o2i路径下, 按照以下指令构建镜像，启动2个o2i新版本的服务**

**docker build -t o2i --rm=true . （构建镜像, o2i是镜像名称）**

**docker run -d --name o2i1 o2i （启动服务, o2i1是服务名称）**

**docker run -d --name o2i2 o2i （启动服务, o2i2是服务名称）**

**3.进入内网ip为192.168.30.53服务器的/opt/javaprocess/new\_server/sync-i2o\_new路径下，按照以下指令构建镜像，启动1个i2o新版本的服务**

**docker build -t i2o --rm=true . （构建镜像, i2o是镜像名称）**

**docker run -d --name i2o1 i2o （启动服务, i2o1是服务名称）**

**4.进入内网ip为192.168.30.53服务器的/opt/javaprocess/new\_server/sync-i2o\_tcp路径下，按照以下指令构建镜像，启动1个i2otcp新版本的服务**

**docker build -t i2otcp --rm=true . （构建镜像, i2otcp是镜像名称）**

**docker run -d --name i2otcp1 i2otcp （启动服务, i2otcp1是服务名称）**

**5.进入ip为192.168.30.54服务器的** **/opt/javaprocess/new\_server/sync-o2i的路径下,按照以下指令构建镜像，启动2个o2i新版本的服务**

**docker build -t o2i --rm=true . （构建镜像, o2i是镜像名称）**

**docker run -d --name o2i1 o2i （启动服务, o2i1是服务名称）**

**docker run -d --name o2i2 o2i （启动服务, o2i2是服务名称）**

**6.查看启动的所有服务**

**docker ps**

**7.停止运行的服务**

**docker stop 服务名称**

**8.重启运行的服务**

**docker restart 服务名称**

### 3.1.11 nginx 服务安装

**1.进入内网ip为192.168.30.12，192.168.30.51两台服务器**

**2.下载安装包** **rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-releasecentos-7-0.el7.ngx.noarch.rpm**

**2. yum install nginx**

**3.修改nginx的配置文件**

**将部署包里面nginx文件夹下面的http.conf跟nginx.conf文件替换掉当前服务器/etc/nginx路径下的nginx.conf与/etc/nginx/conf.d 路径下的http.conf文件**

**4.启动nginx服务**

**systemctl start nginx.service**

**5.nginx服务加入开机启动项**

**systemctl enable nginx.service**

### 3.1.12 keepalived服务安装

**1.分别进入内网ip为192.168.30.12,** **192.168.30.51,** **192.168.30.9,** **192.168.30.11四台服务器执行以下操作**

**yum install ipvsadm**

**yum install keepalived**

**2.修改keepalived.conf文件(vim /etc/keepalived/keepalived.conf)**

**\*\*\*\*\*内网ip为192.168.30.12的服务器具体修改内容如下\*\*\*\*\*\***

**global\_defs {**

**}**

**vrrp\_script chk\_nginx {**

**script "killall -0 nginx"**

**interval 1**

**weight -15**

**}**

**vrrp\_instance VI\_1 {**

**state MASTER**

**interface eno1**

**virtual\_router\_id 52**

**priority 100**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass Inspeeding123456**

**}**

**virtual\_ipaddress {**

**192.168.30.72**

**}**

**track\_script {**

**chk\_nginx**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*内网ip为192.168.30.51的服务器具体修改内容如下\*\*\*\*\*\*\*\***

**global\_defs {**

**}**

**vrrp\_script chk\_nginx {**

**script "killall -0 nginx"**

**interval 1**

**weight -15**

**}**

**vrrp\_instance VI\_1 {**

**state BACKUP**

**interface eno1**

**virtual\_router\_id 52**

**priority 99**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass Inspeeding123456**

**}**

**virtual\_ipaddress {**

**192.168.30.72**

**}**

**track\_script {**

**chk\_nginx**

**}**

**}**

**\*\*\*\*\*\*\*\*内网ip为192.168.30.9的服务器具体修改内容如下\*\*\*\*\*\*\*\***

**global\_defs {**

**}**

**vrrp\_script chk\_mysqld {**

**script "killall -0 mysqld"**

**interval 1**

**weight -15**

**}**

**vrrp\_instance VI\_1 {**

**state MASTER**

**interface eno1**

**virtual\_router\_id 51**

**priority 100**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass Inspeeding123456**

**}**

**virtual\_ipaddress {**

**192.168.30.71**

**}**

**track\_script {**

**chk\_mysqld**

**}**

**}**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*内网ip为192.168.30.11的服务器具体修改内容如下\*\*\*\*\*\*\*\*\***

**global\_defs {**

**}**

**vrrp\_script chk\_mysqld {**

**script "killall -0 mysqld"**

**interval 1**

**weight -15**

**}**

**vrrp\_instance VI\_1 {**

**state BACKUP**

**interface eno1**

**virtual\_router\_id 51**

**priority 99**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass Inspeeding123456**

**}**

**virtual\_ipaddress {**

**192.168.30.71**

**}**

**track\_script {**

**chk\_mysqld**

**}}**

**3.分别在192.168.30.12,** **192.168.30.51,** **192.168.30.9,** **192.168.30.11四台服务器中修改/etc/sysconfig/keepalived文件的参数如下:**

**KEEPALIVED\_OPTIONS="-D -d -S 0"**

**4. 分别在192.168.30.12,** **192.168.30.51,** **192.168.30.9,** **192.168.30.11四台服务器中修改/etc/rsyslog.conf文件, 在文件的最后追加参数如下:**

**local0.\*/var/log/keepalived.log**

**5.分别启动keepalived**

**systemctl start keepalived.service**

**6.分别将keepalived加入开机启动项**

**systemctl enable keepalived.service**

## 内网服务安装部署

### 3.2.1.docker服务安装

**1.分别进入内网ip是176.100.13.164, 176.100.13.164, 176.100.13.164, 176.100.13.164 的4台服务器执行以下步骤安装服务**

**2.在线安装**

**输入yum -y install docker-io指令在线安装,安装过程根据网络快慢需要一段时间完成。**

**3.启动docker服务。**

**systemctl start docker.service**

**4.将docker服务加入开机启动项**

**systemctl enable docker.service**

### 3.2.2.FastDFS集群安装

**1. 进入部署包的FastDFS文件夹中，修改mod\_fastdfs.conf跟storage.conf文件中的参数如下**

**tracker\_server=176.100.13.164:22522**

**tracker\_server=176.100.13.164:22622**

**tracker\_server=176.100.13.165:22522**

**tracker\_server=176.100.13.165:22622**

**2.分别上传FastDFS的整个文件夹目录到ip为176.100.13.164, 176.100.13.165, 176.100.13.172, 176.100.13.173四台服务器的/opt路径下，然后分别进入到四台服务器的FastDFS文件中，安装fastdfs镜像,此安装过程需要几分钟，安装fastdfs镜像指令是： docker build -t fastdfs --rm=true .**

**3. fastdfs镜像安装完后,先启动tracker，后启动storage**

**4. 在ip为176.100.13.164, 176.100.13.165的两台服务器中，每台启动2个tracker服务，如下指令所示：**

**docker run -d --name tracker1 -v /home/tracker/data01:/fastdfs/tracker/data --net=host -e TR\_PORT=22522 fastdfs tracker**

**docker run -d --name tracker2 -v /home/tracker/data02:/fastdfs/tracker/data --net=host -e TR\_PORT=22622 fastdfs tracker**

**5. 在ip为176.100.13.172, 176.100.13.173的两台服务器中，每台启动10个storage服务，如下指令所示:\*\*\*\*\*\*注意以下每个组都是一行指令\*\*\*\*\*\*\*\*\*\*\*\***

**storage1组:**

**docker run -d --name storage1 -v /home/storage/data01:/fastdfs/storage/data -v /home/storage/store\_path01:/fastdfs/store\_path --net=host -e ST\_PORT=23001 -e NGX\_PORT=3001 -e GROUP\_NAME=group1 fastdfs storage**

**storage2组：**

**docker run -d --name storage2 -v /home/storage/data02:/fastdfs/storage/data -v /home/storage/store\_path02:/fastdfs/store\_path --net=host -e ST\_PORT=23002 -e NGX\_PORT=3002 -e GROUP\_NAME=group2 fastdfs storage**

**storage3组：**

**docker run -d --name storage3 -v /home/storage/data03:/fastdfs/storage/data -v /home/storage/store\_path03:/fastdfs/store\_path --net=host -e ST\_PORT=23003 -e NGX\_PORT=3003 -e GROUP\_NAME=group3 fastdfs storage**

**storage4组：**

**docker run -d --name storage4 -v /home/storage/data04:/fastdfs/storage/data -v /home/storage/store\_path04:/fastdfs/store\_path --net=host -e ST\_PORT=23004 -e NGX\_PORT=3004 -e GROUP\_NAME=group4 fastdfs storage**

**storage5组：**

**docker run -d --name storage5 -v /home/storage/data05:/fastdfs/storage/data -v /home/storage/store\_path05:/fastdfs/store\_path --net=host -e ST\_PORT=23005 -e NGX\_PORT=3005 -e GROUP\_NAME=group5 fastdfs storage**

**storage6组：**

**docker run -d --name storage6 -v /home/storage/data06:/fastdfs/storage/data -v /home/storage/store\_path06:/fastdfs/store\_path --net=host -e ST\_PORT=23006 -e NGX\_PORT=3006 -e GROUP\_NAME=group6 fastdfs storage**

**storage7组：**

**docker run -d --name storage7 -v /home/storage/data07:/fastdfs/storage/data -v /home/storage/store\_path07:/fastdfs/store\_path --net=host -e ST\_PORT=23007 -e NGX\_PORT=3007 -e GROUP\_NAME=group7 fastdfs storage**

**storage8组：**

**docker run -d --name storage8 -v /home/storage/data08:/fastdfs/storage/data -v /home/storage/store\_path08:/fastdfs/store\_path --net=host -e ST\_PORT=23008 -e NGX\_PORT=3008 -e GROUP\_NAME=group8 fastdfs storage**

**storage9组：**

**docker run -d --name storage9 -v /home/storage/data09:/fastdfs/storage/data -v /home/storage/store\_path09:/fastdfs/store\_path --net=host -e ST\_PORT=23009 -e NGX\_PORT=3009 -e GROUP\_NAME=group9 fastdfs storage**

**storage10组：**

**docker run -d --name storage10 -v /home/storage/data10:/fastdfs/storage/data -v /home/storage/store\_path10:/fastdfs/store\_path --net=host -e ST\_PORT=23010 -e NGX\_PORT=3010 -e GROUP\_NAME=group10 fastdfs storage**

### 3.2.3. redis与sentinel集群安装

**1.分别进入内网ip为176.100.13.160，176.100.13.172两台服务器**

**2.下载安装包：wget http://download.redis.io/redis-stable.tar.gz**

**3.解压安装**

**tar -xvzf redis-stable.tar.gz**

**cd redis-stable**

**make**

**make install**

**4.Redis配置修改**

**cp /redis-stable/redis.conf /etc/redis.conf**

**修改redis.conf文件 vim /etc/redis.conf（修改redis.conf文件的以下参数）**

**daemonize yes**

**bind 本机的内网IP + 空格 + 127.0.0.1**

**requirepass redis密码**

**masterauth redids密码**

**如果是slave机器，还需要配置**

**slaveof master的ip +空格+master的端口号**

**5.配置完成后启动redis（先启动master）**

**/usr/local/bin/redis-server /etc/redis.conf**

**6.Sentinel配置修改**

**cp /redis-stable/sentinel.conf /etc/sentinel.conf**

**修改sentinel.conf 文件 vim /etc/sentinel.conf（修改sentinel.conf 文件以下参数）**

**protected-mode no**

**daemonize yes**

**sentinel monitor mymaster [Redis mater的ip] [Redis Master的端口号] 2**

**sentinel down-after-milliseconds mymaster 5000**

**sentinel failover-timeout mymaster 10000**

**sentinel auth-pass mymaster [Redis的密码]**

**7.配置完成后启动Sentinel（启动Sentinel之前先将Redis的所有节点启动）**

**/usr/local/bin/redis-sentinel /etc/sentinel.conf**

### 3.2.4.ftp服务安装

**1.进入内网ip为176.100.13.173的服务器**

**2.下载安装包 yum -y install vsftpd**

**3.安装完成后，修改vsftpd.conf文件以下参数**

**anonymous\_enable=NO**

**chroot\_local\_user=YES**

**allow\_writeable\_chroot=YES**

**pasv\_enable=YES**

**pasv\_min\_port=60000**

**pasv\_max\_port=62000**

**4.创建FTP虚拟宿主帐户**

**mkdir /opt/ftp**

**useradd -d /opt/ftp/ryxx -g ftp -s /sbin/nologin ryxx**

**passwd ryxx**

**chown -R ryxx /opt/ftp**

**chown -R 777 /opt/ftp**

**mkdir /opt/ftp/ryxx/out2in**

**mkdir /opt/ftp/ryxx/in2out**

**cd /opt/ftp**

**chmod -R 777 \***

**5.启动FTP服务并添加开机自启动**

**systemctl start vsftpd.service**

**systemctl enable vsftpd.service**

### 3.2.5.nginx安装

1. **rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-releasecentos-7-0.el7.ngx.noarch.rpm**

**2. yum install nginx**

**3.启动nginx服务**

**systemctl start nginx.service**

**4.nginx服务加入开机启动项**

**systemctl enable nginx.service**

### 3.2.6.grxx-sync程序服务安装

#### 3.2.6.1老版本服务

**1.进入内网ip为176.100.13.164的服务器**

**2.进入/opt/SyncInFromOut路径下,运行以下指令构建服务镜像**

**docker build -t old --rm=true . (old指的是镜像名称,根据实际情况改变)**

**3.根据镜像名称,启动1个老服务**

**docker run -d --name SyncInFromOut --restart=unless-stopped old**

**4.查看启动的所有服务**

**docker ps**

**5.停止运行的服务**

**docker stop 服务名称**

**6.重启运行的服务**

**docker restart 服务名称**

#### 3.2.6.2新版本服务

**1. 进入内网ip为176.100.13.164，176.100.13.165,** **176.100.13.172,** **176.100.13.173的四台服务器**

**2.分别进入ip为176.100.13.164,176.100.13.165服务器的 /opt/SYNC-INNER/reader 与/opt/SYNC-INNER/writer路径下**

**3.在/opt/SYNC-INNER/reader路径下，运行以下指令构建reader服务镜像**

**docker build -t reader --rm=true .**

**4.在/opt/SYNC-INNER/writer路径下，运行以下指令构建writer服务镜像**

**docker build -t writer --rm=true .**

**5.根据writer服务镜像，按照以下指令分别在176.100.13.164,176.100.13.165两台服务器中，每个服务器启动2个writer服务**

**docker run -d --name writer1 --restart=unless-stopped writer**

**docker run -d --name writer2 --restart=unless-stopped writer**

6.**根据reader服务镜像，按照以下指令分别在176.100.13.164,176.100.13.165两台服务器中，每个服务器启动10个reader服务**

**docker run -d --name reader1 --restart=unless-stopped reader**

**docker run -d --name reader2 --restart=unless-stopped reader**

**docker run -d --name reader3 --restart=unless-stopped reader**

**docker run -d --name reader4 --restart=unless-stopped reader**

**docker run -d --name reader5 --restart=unless-stopped reader**

**docker run -d --name reader6 --restart=unless-stopped reader**

**docker run -d --name reader7 --restart=unless-stopped reader**

**docker run -d --name reader8 --restart=unless-stopped reader**

**docker run -d --name reader9 --restart=unless-stopped reader**

**docker run -d --name reader10 --restart=unless-stopped reader**

**7.分别进入ip为176.100.13.172,176.100.13.173服务器的/opt/** **javaserver/reader下**

**8.在/opt/** **javaserver/reader路径下，运行以下指令构建reader服务镜像**

**docker build -t reader --rm=true .**

9.**根据reader服务镜像，按照以下指令分别在176.100.13.172,176.100.13.173两台服务器中，每个服务器启动10个reader服务**

**docker run -d --name reader1 --restart=unless-stopped reader**

**docker run -d --name reader2 --restart=unless-stopped reader**

**docker run -d --name reader3 --restart=unless-stopped reader**

**docker run -d --name reader4 --restart=unless-stopped reader**

**docker run -d --name reader5 --restart=unless-stopped reader**

**docker run -d --name reader6 --restart=unless-stopped reader**

**docker run -d --name reader7 --restart=unless-stopped reader**

**docker run -d --name reader8 --restart=unless-stopped reader**

**docker run -d --name reader9 --restart=unless-stopped reader**

**docker run -d --name reader10 --restart=unless-stopped reader**

**10.查看启动的所有服务**

**docker ps**

**11停止运行的服务**

**docker stop 服务名称**

**12.重启运行的服务**

**docker restart 服务名称**