## 1.环境服务部署准备

### 1.1新增log4x用户及密码配置（所有规划的log4x机子都要做）

l  在分布式组网中的**每台机器**上，新建log4x用户并为log4x用户设定密码。

[root@nm-log4x-1 ~]# useradd -g root -d /data -m log4x

[root@nm-log4x-1 ~]# passwd log4x

l  以log4x用户将相关软件包拷贝到nm-log4x-1机器的log4x用户家目录下。

**!注意事项**

（1）项目现场环境请配置**root**用户密码为 **root@34#$**

（2）项目现场环境请配置**log4x**用户密码为 **log4x@34#$**

### 1.2主机名配置（flink集群主机做）

l  为**每台机器**设置主机名，这里以nm-log4x-1为例

[root@nm-log4x-1 ~]# vim /etc/hostname

|  |
| --- |
| nm-log4x-1 |

l  在分布式组网的所有节点，以root用户登录，配置主机名称，使主机之间可以互相识别。

[root@nm-log4x-1 ~]# vim /etc/hosts

|  |
| --- |
| 127.0.0.1      localhost  10.221.229.8 nm-log4x-1  10.221.229.9 nm-log4x-2  10.221.229.53 nm-log4x-3  10.221.229.45 nm-log4x-4  10.221.229.46 nm-log4x-5  10.221.229.47 nm-log4x-6 |

**!注意事项**

（1）首行必须配置为“127.0.0.1  localhost”，不能使用其他配置。

（2）所有机器均要配置hosts文件

### 1.3无密码互访（仅flink集群必须做）

l  以nmyunwei用户登录log4x节点，创建.ssh目录（如果目录存在则忽略此步，其他机器也需要检查是否存在~/.ssh目录，如果不存在，则创建）

[nmyunwei@nm-log4x-4 ~]$ mkdir ~/.ssh

l 在每台机器上输入命令：ssh-keygen -t rsa生成public key(id\_rsa.pub)和private key(id\_rsa)文件。当出现”Enter Passphrase”的提示时直接按回车

[nmyunwei@nm-log4x-4 ~]$ ssh-keygen -t rsa

[nmyunwei@nm-log4x-5 ~6$ ssh-keygen -t rsa

[nmyunwei@nm-log4x-6 ~]$ ssh-keygen -t rsa

l 将每台机器上id\_dsa.pub文件拷贝到master1机器上

[nmyunwei@nm-log4x-5 ~]$ scp ~/.ssh/id\_rsa.pub nmyunwei@nm-log4x-4:~/.ssh/nm-log4x-5.pub

[nmyunwei@nm-log4x-6 ~]$ scp ~/.ssh/id\_rsa.pub nmyunwei@nm-log4x-4:~/.ssh/nm-log4x-6.pub

l 在nm-log4x-4机器上将所有的pub公钥文件追加到authorized\_keys文件

[nmyunwei@nm-log4x-4 ~]$ cat ~/.ssh/\*.pub >> ~/.ssh/authorized\_keys

l 将.ssh目录拷贝到其他机器上，按照提示输入log4x密码

[nmyunwei@nm-log4x-4 ~]$ scp ~/.ssh/authorized\_keys nmyunwei@nm-log4x-5:~/.ssh/

[nmyunwei@nm-log4x-4 ~]$ scp ~/.ssh/authorized\_keys nmyunwei@nm-log4x-6:~/.ssh/

l 在每台机器上修改目录文件权限

[nmyunwei@nm-log4x-4 ~]$ chmod 700 ~/.ssh

[nmyunwei@nm-log4x-4 ~]$ chmod 600 ~/.ssh/authorized\_keys

l 主机之间无密码登录访问测试，满足任意主机之间无密码访问

[nmyunwei@nm-log4x-4 ~]$ ssh nmyunwei@nm-log4x-5

[nmyunwei@nm-log4x-4 ~]$ ssh nmyunwei@nm-log4x-6

### 1.4 JDK安装（所有规划的log4x机子都要做）

l  在nm-log4x-1上以log4x用户安装JDK，安装包名为jdk-8u101-linux-x64.tar.gz

[log4x@nm-log4x-1 ~]$ cd /data

[log4x@nm-log4x-1 ~]$ tar -zxvf  jdk-8u101-linux-x64.tar.gz

l  配置JAVA\_HOME环境变量

[log4x@nm-log4x-1 ~]$ vim ~/.bashrc

|  |
| --- |
| export JAVA\_HOME=/data/jdk1.8.0\_101  export PATH=$JAVA\_HOME/bin:$PATH |

l  拷贝JDK及环境变量到slave机器

[log4x@nm-log4x-1 ~]$ scp -r ~/jdk1.8.0\_101 log4x@master2:~

[log4x@nm-log4x-1 ~]$ scp -r ~/jdk1.8.0\_101 log4x@slave1:~

[log4x@nm-log4x-1 ~]$ scp -r ~/.bashrc log4x@master2:~

[log4x@nm-log4x-1 ~]$ scp -r ~/.bashrc log4x@slave1:~

l  使**每台机器**环境变量生效

[log4x@nm-log4x-1 ~]$ source ~/.bashrc

[log4x@master2 ~]$ source ~/.bashrc

[log4x@slave1 ~]$ source ~/.bashrc

l  为**每台机器**验证jdk环境变量，这里以nm-log4x-1为例

[log4x@nm-log4x-1 ~]$ java -version

openjdk version " 1.8.0\_101"

OpenJDK Runtime Environment (build 1.8.0\_161-b14)

OpenJDK 64-Bit Server VM (build 25.161-b14, mixed mode)

## 2.部署Zookeeper集群

Zookeeper环境部署列表，如下表：

|  |  |  |  |
| --- | --- | --- | --- |
| **服务地址** | **主机名** | **安装服务** | **默认端口** |
| 10.221.229.8 | nm-log4x-1 | Zookeeper（QuorumPeerMain） | 2181 |
| 10.221.229.9 | nm-log4x-2 | Zookeeper（QuorumPeerMain） | 2181 |
| 10.221.229.53 | nm-log4x-3 | Zookeeper（QuorumPeerMain） | 2181 |

**!注意事项**

**（1）安装zookeeper之前可检查zookeeper端口（默认2181端口）是否被占用**

**[log4x@nm-log4x-1 ~]$ netstat -anp|grep 2181**

**（2）若2181端口被占用，则使用2182端口，以此类推，服务端口被占用的情况下，    端口循序向上叠加+1。**

### 2.1安装Zookeeper

l  以log4x用户在nm-log4x-1节点解压zookeeper-3.4.12.tar.gz

[log4x@nm-log4x-1 ~]$ cd /data

[log4x@nm-log4x-1 ~]$ tar -zxvf zookeeper-3.4.12.tar.gz

l  修改zookeeper配置文件

[log4x@nm-log4x-1 ~]$ cd ~/zookeeper-3.4.12/conf/

[log4x@nm-log4x-1 ~]$ mv zoo\_sample.cfg zoo.cfg

[log4x@nm-log4x-1 ~]$ vim ~/zookeeper-3.4.12/conf/zoo.cfg

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/data/zookeeper-3.4.12/data  clientPort=2181   maxClientCnxns=0  server.1=nm-log4x-1:2888:3888  server.2= nmlog4x-2:2888:3888  server.3= nm-log4x-3:2888:3888 |

注：如果机器不够只部署一台zookeeper，则不需要最后三行的配置。配置完成后直接启动该台zookeeper即可。通过zkServer.sh status命令显示处于standalone状态。

l  创建数据目录并创建myid文件

[log4x@nm-log4x-1 ~]$ mkdir /data/zookeeper-3.4.12/data

[log4x@nm-log4x-1 ~]$ cd /data/zookeeper-3.4.12/data

[log4x@nm-log4x-1 ~]$ touch myid; echo 1 > myid;

l  将zookeeper目录拷贝到其他两台机器

[log4x@nm-log4x-1 ~]$ scp -r ~/zookeeper-3.4.12 master2:~

[log4x@nm-log4x-1 ~]$ scp -r ~/zookeeper-3.4.12 slave1:~

l  修改master2中的myid值

[log4x@master2 ~]$ echo 2 > /data/zookeeper-3.4.12/data/myid

l  修改slave中的myid值

[log4x@slave1 ~]$ echo 3 > /data/zookeeper-3.4.12/data/myid

l  分别在**三台机器**上启动zookeeper

[log4x@nm-log4x-1 ~]$ zkServer.sh start

[log4x@master2 ~]$ zkServer.sh start

[log4x@slave1 ~]$ zkServer.sh start

l  分别在**三台机器**上查看zookeeper状态

[log4x@nm-log4x-1 ~]$ zkServer.sh status

[log4x@master2 ~]$ zkServer.sh status

[log4x@slave1 ~]$ zkServer.sh status

如果有一台显示leader，两台显示follower则表示zookeeper集群启动成功。

l  zookeeper停止命令（需要时停止）

[log4x@nm-log4x-1 ~]$ zkServer.sh stop

[log4x@master2 ~]$ zkServer.sh stop

[log4x@slave1 ~]$ zkServer.sh stop

## 3. 部署Kafka集群

kafka集群部署列表，如下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **服务地址** | **主机名** | **服务进程** | **端口** |
| 10.221.229.8 | nm-log4x-1 | Kafka | 9092 |
| 10.221.229.9 | nm-log4x-2 | Kafka | 9092 |
| 10.221.229.53 | nm-log4x-3 | Kafka | 9092 |

**!注意事项**

**（1）安装kafka之前可检查kafka端口（默认9092端口）是否被占用**

**[log4x@nm-log4x-1 ~]$ netstat -anp|grep 9092**

**（2）若9092端口被占用，则使用9093端口，以此类推，服务端口被占用的情况下， 端口循序向上叠加+1。**

### 3.1安装Kafka

l  以log4x用户在nm-log4x-1机器下解压安装包

[log4x@nm-log4x-1 ~]$ tar -zxvf kafka\_2.11-0.10.2.1.tgz

l  将kafka拷贝到其他机器并分别解压

[log4x@nm-log4x-1 ~]$scp kafka\_2.11-0.10.2.1.tgz master2:~

[log4x@nm-log4x-1 ~]$scp kafka\_2.11-0.10.2.1.tgz slave1:~

在master2和slave1上执行

[log4x@master2 ~]$ tar -zxvf kafka\_2.11-0.10.2.1.tgz

[log4x@slave1 ~]$ tar -zxvf kafka\_2.11-0.10.2.1.tgz

### 3.2配置Kafka

l  **每台机器**的配置类似，以nm-log4x-1上配置为例

[log4x@nm-log4x-1 ~]$ vim /home/aicache/kafka\_2.11-0.10.2.1/config/server.properties

|  |
| --- |
| # The id of the broker. This must be set to a unique integer for each broker.  broker.id=0（每台机器确保不一样）  # Hostname and port the broker will advertise to producers and consumers. If not set,  # it uses the value for "listeners" if configured.  Otherwise, it will use the value  # returned from java.net.InetAddress.getCanonicalHostName().  advertised.listeners=PLAINTEXT://nm-log4x-1:9092       ###master2填master2:9092，其他同理  # A comma seperated list of directories under which to store log files  log.dirs=/data01/kafka-logs,/data02/kafka-logs,/data03/kafka-logs    ###如果有多块磁盘，并对kafka的性能有较高要求，建议在每个磁盘下都配置一个目录  # Zookeeper connection string (see zookeeper docs for details).  zookeeper.connect=nm-log4x-1:2181,master2:2181,slave1:2181 |

### 3.3启动Kafka

l  **每台机器**启动kafka服务

[log4x@nm-log4x-1 ~]$ cd /home/aicache/kafka\_2.11-0.10.2.1

[log4x@nm-log4x-1 kafka\_2.11-0.10.2.1]$ bin/kafka-server-start.sh -daemon config/server.properties

l  验证进程是否正常启动

使用 “ps -ef | grep kafka| grep -v grep” 命令观察进程是否存在。

l  验证消息发送和接收

【发送消息命令】

[log4x@nm-log4x-1 kafka\_2.11-0.10.2.1]$ bin/kafka-console-producer.sh --broker-list nm-log4x-1:9092, master2:9092, slave1:9092 --topic test

在控制台输入一些消息，按ctrl+c退出发送。

【读取消息命令】

[log4x@master2 kafka\_2.11-0.10.2.1]$ bin/kafka-console-consumer.sh --bootstrap-server nm-log4x-1:9092, master2:9092, slave1:9092 --topic test --from-beginning

控制台中看到之前发送的消息则证明kafka配置成功，按ctrl+c退出控制台。

【手动创建topic】

[nmyunwei@nm-log4x-2 bin]$ cd /data/kafka\_2.11-1.1.0/bin

[nmyunwei@nm-log4x-2 bin]$ ./kafka-topics.sh --zookeeper nm-log4x-1:2181 --create --topic LOG4X-TRACE-TOPIC --partitions 1 --replication-factor 1

Created topic "LOG4X-TRACE-TOPIC".

[nmyunwei@nm-log4x-2 bin]$ ./kafka-topics.sh --zookeeper nm-log4x-1:2181 --create --topic LOG4X-LOG-TOPIC --partitions 1 --replication-factor 1

Created topic "LOG4X-LOG-TOPIC".

[nmyunwei@nm-log4x-2 bin]$ ./kafka-topics.sh --zookeeper nm-log4x-1:2181 --create --topic LOG4X-METRIC-TOPIC --partitions 1 --replication-factor 1

Created topic "LOG4X-METRIC-TOPIC".

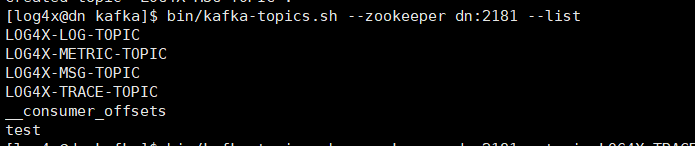
[nmyunwei@nm-log4x-2 bin]$ ./kafka-topics.sh --zookeeper nm-log4x-1:2181 --create --topic LOG4X-MSG-TOPIC --partitions 1 --replication-factor 1

Created topic "LOG4X-MSG-TOPIC".

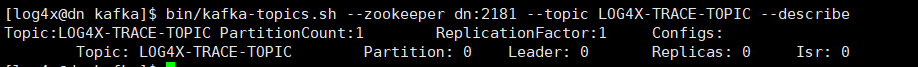
[nmyunwei@nm-log4x-2 bin]$

【检查topic】

[log4x@dn kafka]$ bin/kafka-topics.sh --zookeeper nm-log4x-1:2181 --list



[log4x@dn kafka]$ bin/kafka-topics.sh --zookeeper nm-log4x-1:2181 --topic LOG4X-TRACE-TOPIC --describe



[log4x@dn kafka]$ bin/kafka-topics.sh --zookeeper nm-log4x-1:2181 --topic LOG4X-LOG-TOPIC --describe

[log4x@dn kafka]$ bin/kafka-topics.sh --zookeeper nm-log4x-1:2181 --topic LOG4X-METRIC-TOPIC --describe

### 3.4停止kafka

分别在对应的机器上停止相应进程（需要停止时执行）

l  停止kafka进程

[log4x@nm-log4x-1 kafka\_2.11-0.10.2.1]$bin/kafka-server-stop.sh

## 4. 部署flink集群

flink集群部署列表，如下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **服务地址** | **主机名** | **服务进程** | **端口** |
| 10.221.229.45 | nm-log4x-5 | flink | 9092 |
| 10.221.229.46 | nm-log4x-6 | flink | 9092 |
| 10.221.229.47 | nm-log4x-7 | flink | 9092 |

**!注意事项**

**（1）安装kafka之前可检查kafka端口（默认9092端口）是否被占用**

**[log4x@nm-log4x-1 ~]$ netstat -anp|grep 9092**

**（2）若9092端口被占用，则使用9093端口，以此类推，服务端口被占用的情况下， 端口循序向上叠加+1。**

### 4.1安装flink

l  以log4x用户在nm-log4x-4机器下解压安装包

[log4x@nm-log4x-1 ~]$ tar -zxvf flink-1.7.2-bin-hadoop26-scala\_2.11.tgz

解压后

进入flink-1.7.2

然后mkdir logs data tmp

Cd logs 进入logs目录

mkdir -p trace traceProcess agg perflog

然后

cd flink-1.7.2/

#用来放job文件

mkdir -p log4x

### 4.2配置flink

l  **每台机器**的配置类似，以nm-log4x-4上配置为例

[log4x@nm-log4x-1 ~]$ flink-1.7.2/conf/flink-conf.yaml

|  |
| --- |
| # jobManager 的IP地址  jobmanager.rpc.address: 10.221.229.45 ###master主机IP  # jobManager 的端口号  jobmanager.rpc.port: 6213  # JobManager JVM heap 内存大小  jobmanager.heap.size: 512m  # TaskManager JVM heap 内存大小  taskmanager.heap.size: 2048m  # 每个 TaskManager 提供的任务 slots 数量大小  taskmanager.numberOfTaskSlots: 32  # 程序默认并行计算的个数  parallelism.default: 1  #日志输出目录  env.log.dir: /data/flink1.7.2/logs  #最大日志保留数  env.log.max: 1  #心跳超时设置  heartbeat.timeout: 100000  #web ui端口  rest.port: 18081  #页面提交job所存目录  web.upload.dir: /data/flink-1.7.2/log4x/web-job/webjars  #存放临时文件地址  io.tmp.dirs: /data/ flink1.7.2/tmp |
|  |
| ##### 配置master  ```shell  **在45主机上面执行：**  cd /dataflink-1.7.2/conf  vi masters  ##管理页面  10.221.229.45:18081  ##### 配置slaves  ```shell  cd /data/flink-1.7.2/conf  vi slaves  10.221.229.45  10.221.229.46  10.221.229.47 |

### 4.3启动flink

##### 启动Flink

```shell

bin/start-cluster.sh

jps验证有两个进程

[nmyunwei@nm-log4x-4 bin]$ jps

41521 TaskManagerRunner

40973 StandaloneSessionClusterEntrypoint

Slave主机上面有一个进程TaskManagerRunner

### 4.4停止flink

分别在对应的机器上停止相应进程（需要停止时执行）

l  停止kafka进程

[log4x@nm-log4x-1 kafka\_2.11-0.10.2.1]$bin/stop\_cluster.sh

## 5. 部署ES集群

ES集群部署列表，如下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| **服务地址** | **主机名** | **服务进程** | **端口** |
| 10.221.229.45 | nm-log4x-5 | Es | 9092/9302 |
| 10.221.229.46 | nm-log4x-6 | Es | 9092/9302 |
| 10.221.229.47 | nm-log4x-7 | Es | 9092/9302 |

**!注意事项**

**（1）安装ES之前可检查ES端口（默认9092端口）是否被占用**

**[log4x@nm-log4x-1 ~]$ netstat -anp|grep 9092**

**（2）若9092端口被占用，则使用9093端口，以此类推，服务端口被占用的情况下， 端口循序向上叠加+1。**

### 5.1安装ES

cd /data/data01/ailog4x

tar -xvzf elasticsearch-oss-6.8.1.tar.gz

mv elasticsearch-oss-6.8.1 es-trace-master

mkdir /data/data02/ailog4x/es/es-trace-master/es-data

/data/data02/ailog4x/es/es-trace-master/logs

```

### 5.2配置ES

[nmyunwei@nm-log4x-6 config]$cd /data/es-data/config

Vi elasticsearch.yml

每台集群服务器里的三个程序（es-master、es-client、es-data）配置里面的端口号是不一样的，只有集群名是一样的

拿一台集群服务器做样例查看如下：

|  |
| --- |
| [nmyunwei@nm-log4x-6 config]$ **cd /data/es-master/config**  [nmyunwei@nm-log4x-6 config]$ vi elasticsearch.yml    [nmyunwei@nm-log4x-6 config]$ **cd /data/es-client /config**  [nmyunwei@nm-log4x-6 config]$ vi elasticsearch.yml    [nmyunwei@nm-log4x-6 config]$ **cd /data/** **es-data /config**  [nmyunwei@nm-log4x-6 config]$ vi elasticsearch.yml |

##### 分发主机

```shell

#分发给三个trace-master机器

scp -r elasticsearch-oss-6.8.1.tar.gz ailog4x@10.143.116.175:/data/data01/ailog4x

scp conf/elasticsearch.yml ailog4x@10.143.116.175:/data/data01/ailog4x/es-trace-master/conf/

#trace-data变动，分发其他五台机器

cp es-trace-master es--datraceta

mkdir /data/data02/ailog4x/es/es-trace-data/es-data

/data/data02/ailog4x/es/es-trace-data/logs

cluster.name: log4xv4-trace-cluster

node.name: trace-data-174

node.master: false

node.data: true

transport.tcp.port: 9321 #节点通信端口

http.port: 9221

#trace-client变动，分发其他三台机器

cp es-trace-master es-trace-client

mkdir /data/data02/ailog4x/es/es-trace-client/es-data

/data/data02/ailog4x/es/es-trace-client/logs

cluster.name: log4xv4-trace-cluster

node.name: trace-client-174

node.master: false

node.data: false

transport.tcp.port: 9322 #节点通信端口

http.port: 9222

```

### 5.3启动ES

```shell

bin/elasticsearch -d

```

### 5.4停止ES

## 6. 部署USPA (若刷全量数据库脚本则不需要部署)

### 6.1部署uspa

从装有Tomcat 服务器10.221.229.15 上面SCP 一个csf-webapp过来

登录10.221.229.15

[nmyunwei@wg-zk-pc-01 data]$ scp -r csf-webapp [nmyunwei@10.221.229.47:/data/](mailto:nmyunwei@10.221.229.47:/data/)

登录10.221.229.47 ，cd /data/

[nmyunwei@nm-log4x-6 data]$ mv csf-webapp log4x-uspa

[nmyunwei@nm-log4x-6 data]$ cd log4x-uspa/

[nmyunwei@nm-log4x-6 data]$ rm -rf work

[nmyunwei@nm-log4x-6 data]$cd logs

[nmyunwei@nm-log4x-6 data]$ rm \*cp /data/uspa.war /data/

[nmyunwei@nm-log4x-6 data]$cd /data/log4x-uspa/webapps

[nmyunwei@nm-log4x-6 data]$ rm -rf \*

把提供的程序包 uspa.war 上传cp 到 /data/log4x-uspa/webapps/ 目录下面，启动Tomcat

[nmyunwei@nm-log4x-6 data]$cp /data/uspa.war /data/log4x-uspa/webapps/

[nmyunwei@nm-log4x-6 data]$cd /data/log4x-uspa/bin

[nmyunwei@nm-log4x-6 data]$./startup.sh

启动Tomcat后通过 jps 命令查看进程是否启动

### 6.2刷数据库

|  |
| --- |
| 接下来刷数据库 root mysql -uroot -pfoot\_123  /\*根据主节点添加用户\*/  create user 'log4x'@'localhost' identified by 'Log4x$123';  create user 'log4x'@'%' identified by 'Log4x$123';  /\*赋权 db为库名\*/  GRANT ALL PRIVILEGES ON \*.\* TO log4x@localhost IDENTIFIED BY 'Log4x$123' WITH GRANT OPTION;  GRANT ALL PRIVILEGES ON \*.\* TO log4x@'%' IDENTIFIED BY 'Log4x$123';  flush privileges;  log4x mysql -ulog4x -pLog4x$123  /\*刷sql\*/  create database log4xdb;  use log4xdb;  source /data/mysql/FQ/mysql/log4x/lo4x4.0-mysql.sql;  source /data/mysql/FQ/mysql/log4x/initialdata.sql;  source /data/mysql/FQ/mysql/uspa/uspa.sql;  commit;  10.221.229.13  用root登录MySQL数据库创建用户  ---  [mysql@wg-mysql-db-01 ~]$ /data/mysql/mysql/bin/mysql -u root -P 30305 -p -S /data/mysql/mysql/mysql.sock  Enter password:  Welcome to the MySQL monitor. Commands end with ; or \g.  Your MySQL connection id is 69882  Server version: 5.6.41-log MySQL Community Server (GPL)  Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its  affiliates. Other names may be trademarks of their respective  owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql> create user 'log4x'@'localhost' identified by 'Log4x$123';  Query OK, 0 rows affected (0.00 sec)  mysql> create user 'log4x'@'%' identified by 'Log4x$123';  Query OK, 0 rows affected (0.00 sec)  mysql> GRANT ALL PRIVILEGES ON \*.\* TO log4x@localhost IDENTIFIED BY 'Log4x$123' WITH GRANT OPTION;  Query OK, 0 rows affected (0.00 sec)  mysql> GRANT ALL PRIVILEGES ON \*.\* TO log4x@'%' IDENTIFIED BY 'Log4x$123';  Query OK, 0 rows affected (0.00 sec)  mysql> flush privileges;  Query OK, 0 rows affected (0.00 sec)  mysql> exit  Bye  10.221.229.13  通过上面创建的lol4x用户登录MySQL 创建数据库  [mysql@wg-mysql-db-01 ~]$/data/mysql/mysql/bin/mysql -u log4x -P 30305 -p -S /data/mysql/mysql/mysql.sock  Enter password: Log4x$123  Welcome to the MySQL monitor. Commands end with ; or \g.  Your MySQL connection id is 70278  Server version: 5.6.41-log MySQL Community Server (GPL)  Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.  Oracle is a registered trademark of Oracle Corporation and/or its  affiliates. Other names may be trademarks of their respective  owners.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  mysql> create database log4xdb;  mysql>use log4xdb;  mysql>source /data/mysql/FQ/log4x+db.sql;  mysql>commit; |

### 6.3配置uspa

|  |
| --- |
| 登录10.221.229.47  修改下面的文件中的连接数据库配置  [nmyunwei@nm-log4x-6 classes]$ cd /data/log4x-uspa/webapps/uspa/WEB-INF/classes  [nmyunwei@nm-log4x-6 classes]$ vi uspaJDBC.properties      重启Tomcat 登录页面查看  Cd /data/log4x-uspa/bin  [nmyunwei@nm-log4x-6 bin]$ pwd  /data/log4x-uspa/bin  [nmyunwei@nm-log4x-6 bin]$ ./shutdown.sh  [nmyunwei@nm-log4x-6 bin]$ ./startup.sh |

### 6.4停止uspa

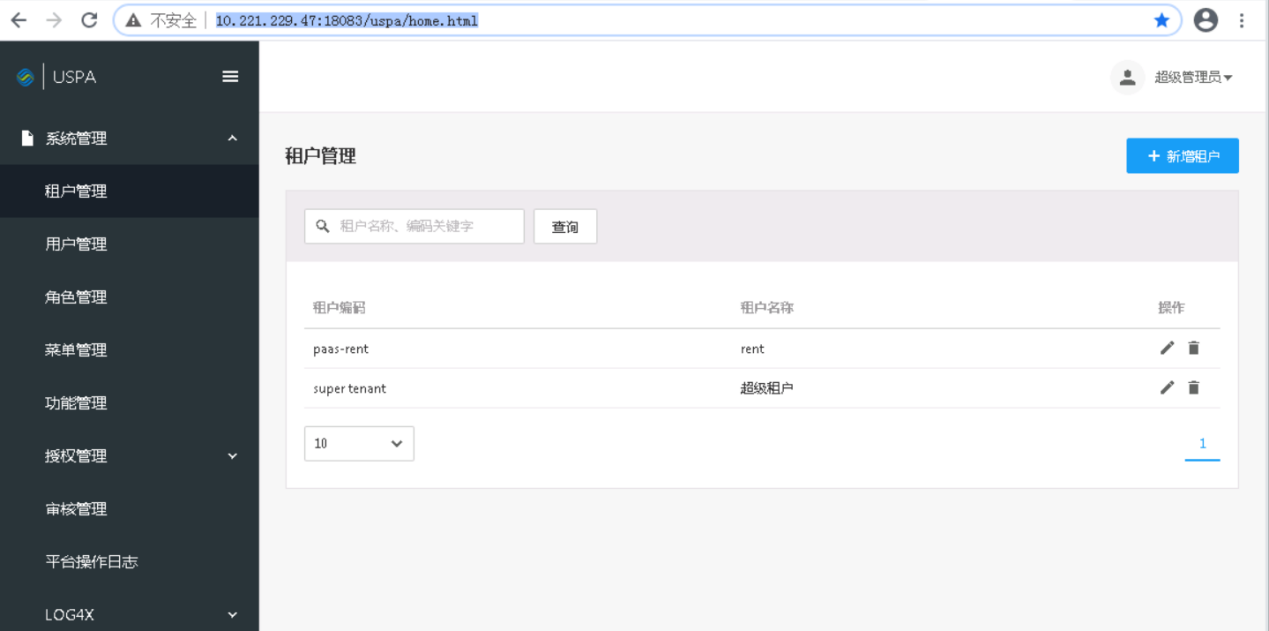
l  停止uspa进程

[nmyunwei@nm-log4x-6 bin]$ cd /data/log4x-uspa/bin

[nmyunwei@nm-log4x-6 bin]$ ./shutdown.sh

### 6.5登录uspa页面查看

<http://10.221.229.47:18083/uspa/home.html>



## 7.部署安装log4x-service

### 7.1安装解压log4x-service.zip

10.221.229.47

[nmyunwei@nm-log4x-6 data]$ unzip log4x-service.zip

### 7.2配置log4x-service

[nmyunwei@nm-log4x-6 config]$ cd /data/log4x-service/config

[nmyunwei@nm-log4x-6 config]$ vi application.yml

|  |
| --- |
| log4x:  datasource:  log4x:  driver-class-name: com.mysql.jdbc.Driver  jdbc-url: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8  username: log4x  password: Log4x$123  initialSize: 1  minIdle: 1  maxActive: 10  maxWait: 30  uspa:  driver-class-name: com.mysql.jdbc.Driver  jdbc-url: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8  username: log4x  password: Log4x$123    zk:  url:  redis:  host: 10.221.229.15  port: 6379 |

### 7.3启停log4x-service

分别在对应的机器上停止相应进程（需要停止时执行）

l  停止kafka进程

##### 启停方式

[nmyunwei@nm-log4x-6 log4x-service]$ cd /data/log4x-service

[nmyunwei@nm-log4x-6 log4x-service]$./start.sh

[nmyunwei@nm-log4x-6 log4x-service]$./stop.sh

jps 查看进程是否启动

## 8. 部署安装scheduler

### 8.1安装解压scheduler

10.221.229.47

[nmyunwei@nm-log4x-6 data]$ unzip log4x-scheduler.zip

### 8.2配置log4x-scheduler

[nmyunwei@nm-log4x-6 config]$ cd /data/log4x-scheduler/config

[nmyunwei@nm-log4x-6 config]$ vi application.yml

|  |
| --- |
| dataSource:  schedulerDataSource:  driver: com.mysql.jdbc.Driver  URL: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8  user: log4x  password: Log4x$123  datasource:  log4x:  # driver-class-name: oracle.jdbc.driver.OracleDriver  # jdbc-url: jdbc:oracle:thin:@10.1.234.62:1521:XE  driver-class-name: com.mysql.jdbc.Driver  jdbc-url: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8  username: log4x  password: Log4x$123  csf:  driver-class-name: com.mysql.jdbc.Driver  jdbc-url: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries=true&useUnicode=true&characterEncoding=utf-8  username: log4x  password: Log4x$123 |

### 8.3启停log4x-scheduler

分别在对应的机器上停止相应进程（需要停止时执行）

l  ##### 启停方式

[nmyunwei@nm-log4x-6 log4x-scheduler]$cd /data/log4x-scheduler

[nmyunwei@nm-log4x-6 log4x-scheduler]$./start.sh

[nmyunwei@nm-log4x-6 log4x-scheduler]$./stop.sh

jps 查看进程是否启动

## 9. 部署安装log4x-web

### 9.1安装解压log4x-web(log4x-web只做解压)

10.221.229.47

[nmyunwei@nm-log4x-6 data]$ unzip log4x-web.zip

解压后在 nginx-1.16.0 程序中配置

## 10.部署安装nginx

### 10.1安装解压nginx-1.16.0.tar.gz

10.221.229.47

[nmyunwei@nm-log4x-6 data]$ tar -zxvf nginx-1.16.0.tar.gz

### 10.2配置nginx-1.16.0

[nmyunwei@nm-log4x-6 config]$ cd /data/ nginx-1.16.0/config

[nmyunwei@nm-log4x-6 config]$ vi nginx.conf

新添加一段

|  |
| --- |
| upstream log4xservice{  server 10.221.229.47:8089;  }  server {  listen 8080;  server\_name 10.221.229.47;  # charset koi8-r;  # access\_log logs/host.access.log main;  location /bpcl-web {  root html;  }  location / {  # root html;  index index.html index.htm;  proxy\_pass http://log4xservice;  }  location ^~/log4x-web {  try\_files $uri $uri/ /index.html;  alias /data/log4x-web; #前端静态页面  index index.html;  }  error\_page 500 502 503 504 /50x.html;  location = /50x.html {  root html;  }  } |

配置文件样例：



### 10.3启停nginx

分[nmyunwei@nm-log4x-6 sbin]$ ./nginx -s stop

[nmyunwei@nm-log4x-6 sbin]$ ps -ef|grep ngi

nmyunwei 60043 59778 0 11:27 pts/0 00:00:00 grep --color=auto ngi

[nmyunwei@nm-log4x-6 sbin]$ ./nginx

[nmyunwei@nm-log4x-6 sbin]$ ps -ef|grep ngi

nmyunwei 60045 1 0 11:27 ? 00:00:00 nginx: master process ./nginx

nmyunwei 60046 60045 0 11:27 ? 00:00:00 nginx: worker process

nmyunwei 60172 59778 0 11:28 pts/0 00:00:00 grep --color=auto ngi

## 11.部署安装hbase（若规划中无hbase则不需要安装）

### 11.1安装解压hbase-1.2.6.1-bin.tar.gz

10.221.229.47

[nmyunwei@nm-log4x-6 data]$ tar -zxvf hbase-1.2.6.1-bin.tar.gz

[nmyunwei@nm-log4x-6 data]$**ln -s hbase-1.2.6.1 hbase**

#建⽴立HBase的⽇日志⽂文件夹

[nmyunwei@nm-log4x-6 data]$ cd /data/hbase

[nmyunwei@nm-log4x-6 hbase]$ mkdir logs

### 11.2配置hbase-site.xml

[nmyunwei@nm-log4x-6 conf]$ cd /data/hbase/conf

[nmyunwei@nm-log4x-6 config]$ vi hbase-site.xml

|  |
| --- |
| <configuration>  <!--hbase存储根⽬目录-->  <property>  <name>hbase.rootdir</name>  <value>file:///data/hbase/hbase\_data/</value> ----/当前配置为本地文件系统，也可以配置HDFS地址，例如 hdfs://hostname:8020/hbase/  </property>  <!--分布式开关-->  <property>  <name>hbase.cluster.distributed</name>  <value>true</value>  </property>  <!--zk集群地址-->  <property>  <name>hbase.zookeeper.quorum</name>  <value>10.221.229.8:2181,10.221.229.9:2181,10.221.229.53:2181</value>  </property>  </configuration> |

### 11.3配置启动参数

|  |
| --- |
| 配置启动参数  [nmyunwei@nm-log4x-6 conf]$cd /data/hbase/conf  [nmyunwei@nm-log4x-6 conf]$ vi hbase-env.sh  #关闭⾃自带的zk  export HBASE\_MANAGES\_ZK=false  #让HBase读取到HDFS的配置  #HBASE\_CLASSPATH=/  #修改HBase⽇日志输出的⽂文件夹  export HBASE\_LOG\_DIR=/data/hbase/logs  ``` |

### 11.3启停hbase

[nmyunwei@nm-log4x-6 bin]$ cd /data/hbase/bin

[nmyunwei@nm-log4x-6 bin]$ ./start-hbase.sh

[nmyunwei@nm-log4x-6 bin]$ ./stop-hbase.sh

##### 创建namespace

```shell

cd /home/log4x/hbase/bin

./hbase --config ../conf shell

list\_namespace ''

create\_namespace 'cx\_log4x'

```

创建hbase namespace

hbase :

[nmyunwei@nm-log4x-6 bin]$ ./hbase --config ../conf shell

2021-03-09 09:57:47,728 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

HBase Shell; enter 'help<RETURN>' for list of supported commands.

Type "exit<RETURN>" to leave the HBase Shell

Version 1.2.6.1, rUnknown, Sun Jun 3 23:19:26 CDT 2018

hbase(main):001:0> ps -ef|grep hbase

SyntaxError: (hbase):1: syntax error, unexpected tIDENTIFIER

ps -ef|grep hbase

^

hbase(main):002:0> list\_namespace ''

NAMESPACE

default

hbase

2 row(s) in 0.2030 seconds

hbase(main):003:0> create\_namespace 'cx\_log4x'

0 row(s) in 0.0340 seconds

hbase(main):004:0> list\_namespace ''

NAMESPACE

cx\_log4x

default

hbase

3 row(s) in 0.0300 seconds

hbase(main):005:0>

flink-job jar包中配置

[nmyunwei@nm-log4x-4 log4x]$ jar -xvf log4x-flink-4.0.0.jar log4x.yml

已解压: log4x.yml

修改编辑log4x.yml 里面的hbase 地址配置

[nmyunwei@nm-log4x-4 log4x]$ vi log4x.yml

zk.model: hbase

java.security.auth.login.config: ''

hbase:

flush.period.ms: 15000

write.buffer.size: 5242880

task.per.region: 10

task.per.server: 30

default:

namespace: "cx\_log4x"

ttl: 86400

zookeeper:

server.list: "10.221.229.8:2181,10.221.229.9:2181,10.221.229.53:2181"

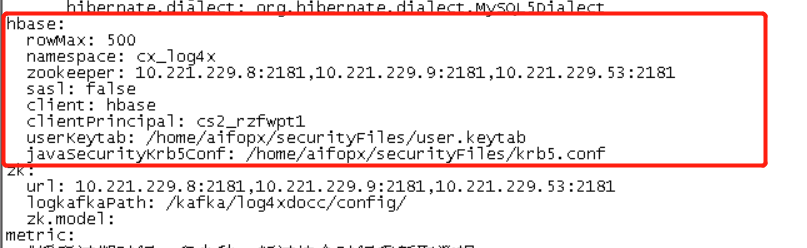
再打包进去

[nmyunwei@nm-log4x-4 log4x]$ jar uvf log4x-flink-4.0.0.jar log4x.yml

log4x-service中配置

[nmyunwei@nm-log4x-6 config]$ cd /data/log4x-service/config

[nmyunwei@nm-log4x-6 config]$ vi application.yml



## 12. 组件检查调试

## 12.1 检查ES索引状态

用以下命令查询索引状态

[nmyunwei@nm-log4x-6 config]$ curl -XGET '10.221.229.45:9220/\_cat/indices?v&pretty'

|  |
| --- |
|  |

## 12.2 配置启动flink-job

**登录flink-master主机**

[nmyunwei@nm-log4x-4 data]$ cp log4x-flink-4.0.0.jar /data/flink-1.7.2/log4x/

[nmyunwei@nm-log4x-4 data]$ cd /data/flink-1.7.2/log4x/

[nmyunwei@nm-log4x-4 log4x]$ jar -xvf log4x-flink-4.0.0.jar log4x.yml

[nmyunwei@nm-log4x-4 log4x]$ vi log4x.yml

|  |
| --- |
| 修改里面的数据库配置及kafak地址  dbc.driver: com.mysql.jdbc.Driver  jdbc.url: jdbc:mysql://10.221.229.13:30305/log4xdb?allowMultiQueries:true  jdbc.username: log4x  jdbc.password: "Log4x$123"  flink.fixedJob:  traces:  - name: trace  bootstrap.servers: 10.221.229.8:9092,10.221.229.9:9092,10.221.229.53:9092  ......    此配置有多处都 需要修改，修改完保存后执行下面的命令  [nmyunwei@nm-log4x-4 log4x]$ jar uvf log4x-flink-4.0.0.jar log4x.yml  启动脚本查看加工数据  [nmyunwei@nm-log4x-4 flink-1.7.2]$ pwd  /data/flink-1.7.2  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -log  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -osp  Starting execution of program  Job has been submitted with JobID b4089aa6637951ca1def481ef9745ed1  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -ospAgg  Starting execution of program  Job has been submitted with JobID 53ae9c14176abd8bc0fbe77514453dcb  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -agg  Starting execution of program  Job has been submitted with JobID 8483e569b92164990a55c276fef2fd18  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -traces trace  Starting execution of program  Job has been submitted with JobID 9a0ab68d194084fe1153b6d501f82fae  [nmyunwei@nm-log4x-4 flink-1.7.2]$ bin/flink run -d log4x/log4x-flink-4.0.0.jar -traces traceProcess  Starting execution of program  Job has been submitted with JobID 04a10d54aaad667fd1d2b1e218038539  [nmyunwei@nm-log4x-4 flink-1.7.2]$  执行完上面的命令,查看页面进程是否启动（此地址是在flink里面配置过的 ）  [nmyunwei@nm-log4x-6 conf]$ cat /data/flink-1.7.2/conf/masters  10.221.229.45:18081  [http://10.221.229.45:18081/#/running-jobs](http://10.221.229.45:18081/" \l "/running-jobs) |

## 12.3调试测试数据

登录log4x-web部署主机上解压log4x-mock.tar.gz

[nmyunwei@nm-log4x-6 20210306]$ cd /data/

tar -zxvf log4x-mock.tar.gz

修改log4x.yaml里面的配置，改为实际kafka集群地址，log4x-mock功能是模拟往kafak里面写数据

[nmyunwei@nm-log4x-6 configext]$ **cd /data/log4x-mock/configext**

[nmyunwei@nm-log4x-6 configext]$ vi log4x.yaml

