

从所有教程的词条中查询...

全部开发者教程

5 项目核心复盘

第14周-消息队列之Kafka从入门到小牛

1 初识Kafka

2 Kafka集群安装部署

3 Kafka使用初体验

4 Kafka核心扩展内容

5 Kafka核心之存储和容错机制

6 Kafka生产消费者实战

7 Kafka技巧篇

8 Kafka小试牛刀实战篇

9 Kafka核心复盘

第15周-极速上手内存数据库

Java / 2 Kafka集群安装部署



徐老师 • 更新于 2020-09-25

上一节 1 初识Kafka 3 Kafka使用初体... 下一节

前面我们对Kafka有了一个基本的认识，下面我们就想使用一下Kafka  
在使用之前，需要先把Kafka安装部署起来  
Kafka是支持单机和集群模式的，建议大家在学习阶段使用单机模式即可，单机和集群在操作上没有任何区别

注意：由于Kafka需要依赖于Zookeeper，所以在这我们需要先把Zookeeper安装部署起来

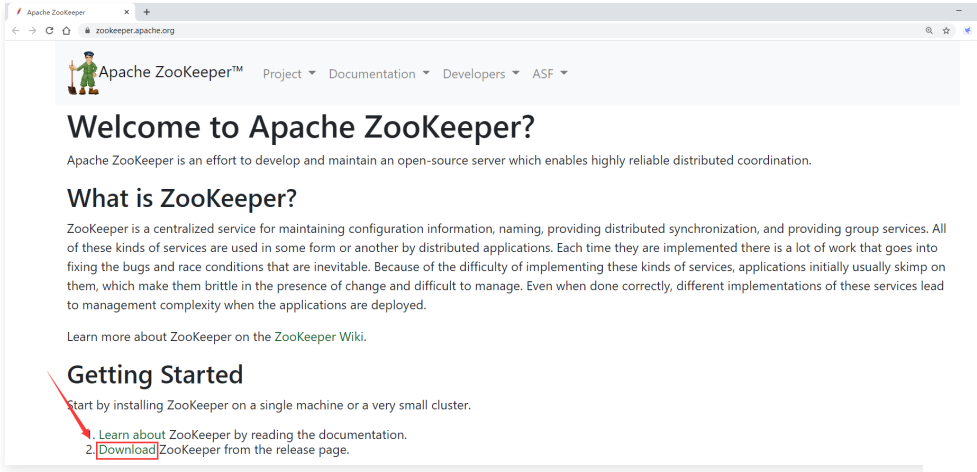
zookeeper安装部署

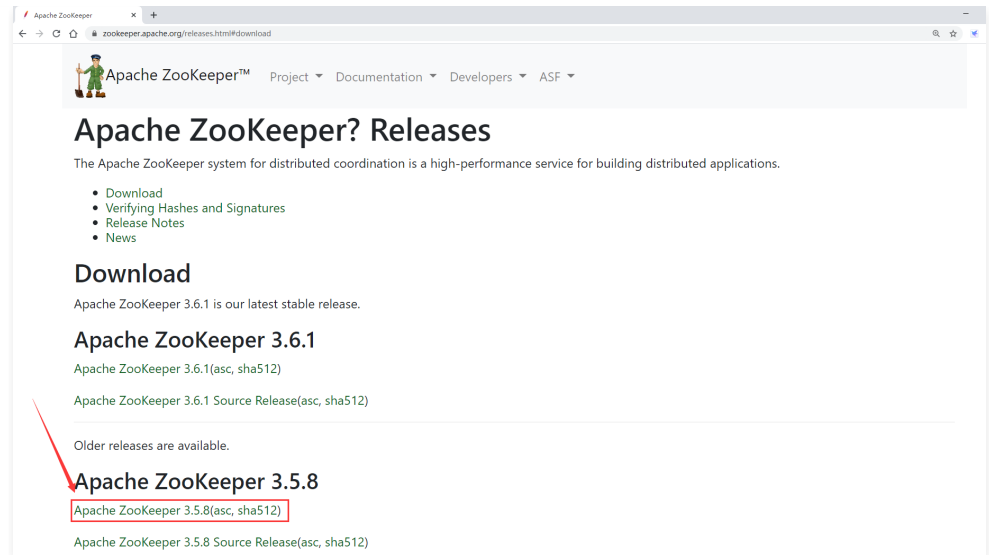
针对Zookeeper前期不需要掌握太多，只需要掌握Zookeeper的安装部署以及它的基本操作即可。  
Zookeeper也支持单机和集群安装，建议大家在学习阶段使用单机即可，单机和集群在操作上没有任何区别。  
在这里我们会针对单机和集群这两种方式分别演示一下

zookeeper单机安装

zookeeper需要依赖于jdk，只要保证jdk已经正常安装即可。  
具体安装步骤如下：

- 1：下载zookeeper的安装包
- 进入Zookeeper的官网





最终下载链接如下：

<https://mirrors.bfsu.edu.cn/apache/zookeeper/zookeeper-3.5.8/apache-zookeeper-3.5.8-bin.tar.gz>

2: 把安装包上传到bigdata01机器的/data/soft目录下

<> 代码块

```
1 [root@bigdata01 soft]# ll
2 -rw-r--r--. 1 root root 9394700 Jun 2 21:33 apache-zookeeper-3.5.8-bin.t
```

3: 解压安装包

<> 代码块

```
1 [root@bigdata01 soft]# tar -zxvf apache-zookeeper-3.5.8-bin.tar.gz
```

4: 修改配置文件

首先将 zoo\_sample.cfg 重命名为 zoo.cfg

然后修改 zoo.cfg 中的 dataDir 参数的值，dataDir 指向的目录存储的是zookeeper的核心数据，所以这个目录不能使用tmp目录

<> 代码块

```
1 [root@bigdata01 soft]# cd apache-zookeeper-3.5.8-bin/conf
2 [root@bigdata01 conf]# mv zoo_sample.cfg zoo.cfg
3 [root@bigdata01 conf]# vi zoo.cfg
4 dataDir=/data/soft/apache-zookeeper-3.5.8-bin/data
```

5: 启动zookeeper服务

<> 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh start
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Starting zookeeper ... STARTED
```

6: 验证

如果能看到QuorumPeerMain进程就说明zookeeper启动成功

<> 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# jps
2 1701 QuorumPeerMain
```

意见反馈

收藏教程

标记书签

注意：如果执行jps命令发现没有QuorumPeerMain进程，则需要到logs目录下去查看zookeeper-\*.out这个日志文件

也可以通过 zkServer.sh 脚本查看当前机器中zookeeper服务的状态

注意：使用zkServer.sh默认会连接本机2181端口的zookeeper服务，默认情况下zookeeper会监听2181端口，这个需要注意一下，因为后面我们在使用zookeeper的时候需要知道它监听的端口是哪个。

最下面显示的Mode信息，表示当前是一个单机独立集群

<> 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh status
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Client port found: 2181. Client address: localhost.
5 Mode: standalone
```

如果没有启动成功的话则会提示连不上服务not running

## 7: 操作zookeeper

首先使用zookeeper的客户端工具连接到zookeeper里面，使用bin目录下面的 zkCli.sh 脚本，默认会连接本机的zookeeper服务

<> 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# bin/zkCli.sh
2 Connecting to localhost:2181
3 .....
4 WATCHER::
5
6 WatchedEvent state:SyncConnected type:None path:null
7 [zk: localhost:2181(CONNECTED) 0]
```

这样就进入zookeeper的命令行了。

在这里面可以操作Zookeeper中的目录结构

zookeeper中的目录结构和Linux文件系统的目录结构类似

zookeeper里面的每一个目录我们称之为节点(ZNode)

正常情况下我们可以把ZNode认为和文件系统中的目录类似，但是有一点需要注意：ZNode节点本身是可以存储数据的。

zookeeper中提供了一些命令可以对它进行一些操作

在命令行下随便输入一个字符，按回车就会提示出zookeeper支持的所有命令

<> 代码块

```
1 [zk: localhost:2181(CONNECTED) 8] aa
2 ZooKeeper -server host:port cmd args
3     addauth scheme auth
4     close
5     config [-c] [-w] [-s]
6     connect host:port
7     create [-s] [-e] [-c] [-t ttl] path [data] [acl]
8     delete [-v version] path
9     deleteall path
10    delquota [-n|-b] path
11    get [-s] [-w] path
```

意见反馈

收藏教程

标记书签

```
14      listquota path
15      ls [-s] [-w] [-R] path
16      ls2 path [watch]
17      printwatches on|off
18      quit
19      reconfig [-s] [-v version] [[-file path] | [-members serverID=host:po
20      redo cmdno
21      removewatches path [-c|-d|-a] [-l]
22      rmr path
23      set [-s] [-v version] path data
24      setAcl [-s] [-v version] [-R] path acl
25      setquota -n|-b val path
26      stat [-w] path
27      sync path
28      Command not found: Command not found aa
```

下面我们来具体看一些比较常用的功能：

- 查看根节点下面有什么内容

这里显示根节点下面有一个zookeeper节点

<> 代码块

```
1      [zk: localhost:2181(CONNECTED) 0] ls /
2      [zookeeper]
```

- 创建节点

在根节点下面创建一个test节点，在test节点上存储数据hello

<> 代码块

```
1      [zk: localhost:2181(CONNECTED) 9] create /test hello
2      Created /test
```

- 查看节点中的信息

查看/test节点中的内容

<> 代码块

```
1      [zk: localhost:2181(CONNECTED) 10] get /test
2      hello
```

- 删除节点

这个删除命令可以递归删除，这里面还有一个delete命令，也可以删除节点，但是只能删除空节点，节点下面还有子节点，想一次性全部删除建议使用deleteall

<> 代码块

```
1      [zk: localhost:2181(CONNECTED) 6] deleteall /test
```

直接按 `ctrl+c` 就可以退出这个操作界面，想优雅一些的话可以输入 `quit` 退出

## 8: 停止zookeeper服务

<> 代码块

意见反馈

收藏教程

标记书签

```
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Stopping zookeeper ... STOPPED
```

## zookeeper集群安装

1: 集群节点规划, 使用三个节点搭建一个zookeeper集群

<> 代码块

```
1 bigdata01
2 bigdata02
3 bigdata03
```

2: 首先在bigdata01节点上配置zookeeper

解压

<> 代码块

```
1 [root@bigdata01 soft]# tar -zxvf apache-zookeeper-3.5.8-bin.tar.gz
```

修改配置

<> 代码块

```
1 [root@bigdata01 soft]# cd apache-zookeeper-3.5.8-bin/conf/
2 [root@bigdata01 conf]# mv zoo_sample.cfg zoo.cfg
3 dataDir=/data/soft/apache-zookeeper-3.5.8-bin/data
4 server.0=bigdata01:2888:3888
5 server.1=bigdata02:2888:3888
6 server.2=bigdata03:2888:3888
```

创建目录保存 myid 文件, 并且向 myid 文件中写入内容

myid 中的值其实是和 zoo.cfg 中 server 后面指定的编号是一一对应的

编号 0 对应的是 bigdata01 这台机器, 所以在这里指定 0

在这里使用 echo 和 重定向 实现数据写入

<> 代码块

```
1 [root@bigdata01 conf]# cd /data/soft/apache-zookeeper-3.5.8-bin
2 [root@bigdata01 apache-zookeeper-3.5.8-bin]# mkdir data
3 [root@bigdata01 apache-zookeeper-3.5.8-bin]# cd data
4 [root@bigdata01 data]# echo 0 > myid
```

3: 把修改好配置的zookeeper拷贝到其它两个节点

<> 代码块

```
1 [root@bigdata01 soft]# scp -rq apache-zookeeper-3.5.8-bin bigdata02:/data/
2 [root@bigdata01 soft]# scp -rq apache-zookeeper-3.5.8-bin bigdata03:/data/
```

4: 修改bigdata02和bigdata03上zookeeper中myid文件的内容

首先修改bigdata02节点上的myid文件

<> 代码块

```
1 [root@bigdata02 ~]# cd /data/soft/apache-zookeeper-3.5.8-bin/data/
2 [root@bigdata02 data]# echo 1 > myid
```

然后修改bigdata03节点上的myid文件

&lt;&gt; 代码块

```
1 [root@bigdata03 ~]# cd /data/soft/apache-zookeeper-3.5.8-bin/data/
2 [root@bigdata03 data]# echo 2 > myid
```

### 5: 启动zookeeper集群

分别在 bigdata01、bigdata02、bigdata03 上启动zookeeper进程

在bigdata01上启动

&lt;&gt; 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh start
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Starting zookeeper ... STARTED
```

在bigdata02上启动

&lt;&gt; 代码块

```
1 [root@bigdata02 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh start
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Starting zookeeper ... STARTED
```

在bigdata03上启动

&lt;&gt; 代码块

```
1 [root@bigdata03 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh start
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Starting zookeeper ... STARTED
```

### 6: 验证

分别在bigdata01、bigdata02、bigdata03上执行jps命令验证是否有 QuorumPeerMain 进程

如果都有就说明zookeeper集群启动正常了

如果没有就到对应的节点的 logs 目录下查看 zookeeper\*-.out 日志文件

执行 bin/zkServer.sh status 命令会发现有一个节点显示为 leader，其他两个节点为 follower

&lt;&gt; 代码块

```
1 [root@bigdata01 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh status
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Client port found: 2181. Client address: localhost.
5 Mode: follower
```

&lt;&gt; 代码块

```
1 [root@bigdata02 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh status
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Client port found: 2181. Client address: localhost.
5 Mode: leader
```

&lt;&gt; 代码块

```
1 [root@bigdata03 apache-zookeeper-3.5.8-bin]# bin/zkServer.sh status
2 ZooKeeper JMX enabled by default
3 Using config: /data/soft/apache-zookeeper-3.5.8-bin/bin/../conf/zoo.cfg
4 Client port found: 2181. Client address: localhost.
```

[意见反馈](#)[收藏教程](#)[标记书签](#)

7: 操作zookeeper  
和上面单机的操作方式一样

8: 停止zookeeper集群

在bigdata01、bigdata02、bigdata03三台机器上分别执行bin/zkServer.sh stop命令

## kafka安装部署

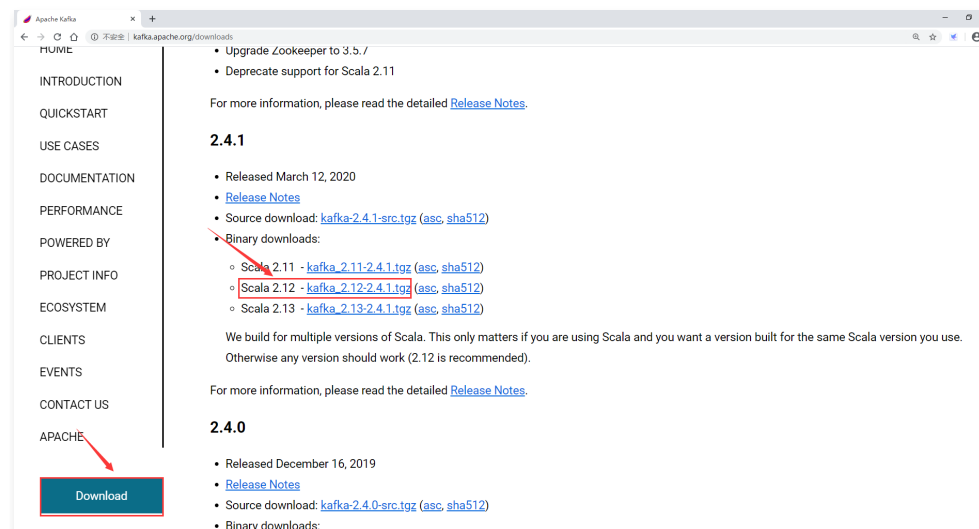
zookeeper集群安装好了以后就可以开始安装kafka了。

注意：在安装kafka之前需要先确保zookeeper集群是启动状态。

kafka还需要依赖于基础环境jdk，需要确保jdk已经安装到位。

## kafka单机安装

1: 下载kafka安装包



最终下载链接如下：

[https://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.4.1/kafka\\_2.12-2.4.1.tgz](https://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.4.1/kafka_2.12-2.4.1.tgz)

注意：kafka在启动的时候不需要安装scala环境，只有在编译源码的时候才需要，因为运行的时候是在jvm虚拟机上运行的，只需要有jdk环境就可以了

2: 把kafka安装包上传到bigdata01的/data/soft目录下

<> 代码块

```
1 [root@bigdata01 soft]# ll
2 -rw-r--r--. 1 root root 62358954 Jun 2 21:22 kafka_2.12-2.4.1.tgz
```

3: 解压

<> 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# tar -zxvf kafka_2.12-2.4.1.tgz
```

4: 修改配置文件

主要参数：

&lt;&gt; 代码块

```
1 broker.id: 集群节点id编号, 单机模式不用修改
2 listeners: 默认监听9092端口
3 log.dirs: 注意: 这个目录不是存储日志的, 是存储Kafka中核心数据的目录, 这个目录默认是:
4 zookeeper.connect: kafka依赖的zookeeper
```

针对单机模式, 如果kafka和zookeeper在同一台机器上, 并且zookeeper监听的端口就是那个默认的2181端口, 则 `zookeeper.connect` 这个参数就不需要修改了。

只需要修改一下 `log.dirs` 即可

&lt;&gt; 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# cd kafka_2.12-2.4.1/config/
2 [root@bigdata01 config]# vi server.properties
3 log.dirs=/data/soft/kafka_2.12-2.4.1/kafka-logs
```

#### 5: 启动kafka

&lt;&gt; 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# bin/kafka-server-start.sh -daemon config/s
```

#### 6: 验证

启动成功之后会产生一个kafka进程

&lt;&gt; 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# jps
2 2230 QuorumPeerMain
3 3117 Kafka
4 3182 Jps
```

#### 7: 停止kafka

&lt;&gt; 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# bin/kafka-server-stop.sh
```

#### 2.1.2.2: kafka集群安装

##### 1: 集群节点规划, 使用三个节点搭建一个kafka集群

&lt;&gt; 代码块

```
1 bigdata01
2 bigdata02
3 bigdata03
```

注意: 针对Kafka集群而言, 没有主从之分, 所有节点都是一样的。

##### 2: 首先在bigdata01节点上配置kafka

解压:

```
[root@bigdata01 soft]# tar -zxvf kafka_2.12-2.4.1.tgz
```

修改配置文件

注意: 此时针对集群模式需要修改 `broker.id`、`log.dirs`、以及 `zookeeper.connect`

`broker.id` 的值默认是从0开始的, 集群中所有节点的 `broker.id` 从0开始递增即可

[意见反馈](#)[收藏教程](#)[标记书签](#)



`log.dirs` 的值建议指定到一块存储空间比较大的磁盘上面, 因为在实际工作中kafka中会存储很多数据, 我这个虚拟机里面就一块磁盘, 所以就指定到/data目录下面了

`zookeeper.connect` 的值是zookeeper集群的地址, 可以指定集群中的一个节点或者多个节点地址, 多个节点地址之间使用逗号隔开即可

&lt;&gt; 代码块

```
1 [root@bigdata01 soft]# cd kafka_2.12-2.4.1/config/
2 broker.id=0
3 log.dirs=/data/kafka-logs
4 zookeeper.connect=bigdata01:2181,bigdata02:2181,bigdata03:2181
```

3: 将修改好配置的kafka安装包拷贝到其它两个节点

&lt;&gt; 代码块

```
1 [root@bigdata01 soft]# scp -rq kafka_2.12-2.4.1 bigdata02:/data/soft/
2 [root@bigdata01 soft]# scp -rq kafka_2.12-2.4.1 bigdata03:/data/soft/
```

4: 修改bigdata02和bigdata03上kafka中broker.id的值

首先修改bigdata02节点上的broker.id的值为1

&lt;&gt; 代码块

```
1 [root@bigdata02 ~]# cd /data/soft/kafka_2.12-2.4.1/config/
2 [root@bigdata02 config]# vi server.properties
3 broker.id=1
```

然后修改bigdata03节点上的broker.id的值为2

&lt;&gt; 代码块

```
1 [root@bigdata03 ~]# cd /data/soft/kafka_2.12-2.4.1/config/
2 [root@bigdata03 config]# vi server.properties
3 broker.id=2
```

5: 启动集群

分别在bigdata01、bigdata02、bigdata03上启动kafka进程

在bigdata01上启动

&lt;&gt; 代码块

```
1 [root@bigdata01 kafka_2.12-2.4.1]# bin/kafka-server-start.sh -daemon config/s
```

在bigdata02上启动

&lt;&gt; 代码块

```
1 [root@bigdata02 kafka_2.12-2.4.1]# bin/kafka-server-start.sh -daemon con
```

在bigdata03上启动

&lt;&gt; 代码块

```
1 [root@bigdata03 kafka_2.12-2.4.1]# bin/kafka-server-start.sh -daemon config/s
```

6: 验证

分别在bigdata01、bigdata02、bigdata03上执行jps命令验证是否有kafka进程

如果都有就说明kafka集群启动正常了

 我要提出意见反馈



 意见反馈

 收藏教程

 标记书签