

4.8 实战案例——Kafka 集群部署

4.8.1 案例目标

- （1）了解 Kafka 分布式发布订阅消息系统。
- （2）搭建 Kafka 分布式发布订阅消息系统。
- （3）使用 Kafka 分布式发布订阅消息系统。

4.8.2 案例分析

1. 规划节点

Kafka 集群的节点规划，见表 4-8-1。

表 4-8-1 节点规划

IP	主机名	节点
172.16.51.23	zookeeper1	集群节点
172.16.51.32	zookeeper2	集群节点
172.16.51.41	zookeeper3	集群节点

2. 基础准备

使用 ZooKeeper 集群搭建的 3 个节点来构建 Kafka 集群，因为 Kafka 服务依赖于 ZooKeeper 服务，所以不再多创建云主机来进行试验。软件包使用提供的 kafka_2.11-1.1.1.tgz 软件包。

4.8.3 案例实施

搭建 Kafka 集群

（1）解压 Kafka 软件包

将提供的 kafka_2.11-1.1.1.tgz 软件包，上传至 3 个节点的/root 目录下，并执行解压操作，3 个节点执行的解压命令如下：

```
# tar -zxvf kafka_2.11-1.1.1.tgz
```

（2）修改 3 个节点配置文件

在 zookeeper1 节点，进入 kafka_2.11-1.1.1/config 目录下，编辑 server.properties 文件。

在配置文件中找到以下两行并注释掉（在文本前加#）如下所示：

```
#broker.id=0
```

```
#zookeeper.connect=localhost:2181
```

然后在配置文件的底部添加如下 3 个配置。

zookeeper1 节点：

```
broker.id=1
```

```
zookeeper.connect=172.16.51.23:2181,172.16.51.32:2181,172.16.51.41:2181
```

```
listeners = PLAINTEXT://172.16.51.23:9092
```

命令解析：

- broker.id：每台机器不能一样。
- zookeeper.connect：因为有 3 台 ZooKeeper 服务器，所以在这里 zookeeper.connect 设置为 3 台。

- listeners：在配置集群的时候，必须设置，不然以后的操作会报找不到 leader 的错误。

另外两台服务器，zookeeper.connect 的配置跟这里的一样，但是 broker.id 和 listeners 不能一样。

zookeeper2 节点：

```
broker.id=2
```

```
zookeeper.connect=172.16.51.23:2181,172.16.51.32:2181,172.16.51.41:2181
```

```
listeners = PLAINTEXT://172.16.51.32:9092
```

zookeeper3 节点：

```
broker.id=3
```

```
zookeeper.connect=172.16.51.23:2181,172.16.51.32:2181,172.16.51.41:2181
```

```
listeners = PLAINTEXT://172.16.51.41:9092
```

（3）启动服务

在 zookeeper1 节点，进入 kafka_2.11-1.1.1/bin 目录下，启动服务命令如下。

zookeeper1 节点：

```
[root@zookeeper1 bin]# ./kafka-server-start.sh -daemon ../config/server.properties
```

```
[root@zookeeper1 bin]# jps
```

```
11416 Kafka
```

```
11464 Jps
```

```
10479 QuorumPeerMain
```

zookeeper2 节点：

```
[root@zookeeper2 bin]# ./kafka-server-start.sh -daemon ../config/server.properties
```

```
[root@zookeeper2 bin]# jps
```

```
11121 Kafka
```

```
11188 Jps
```

```
10175 QuorumPeerMain
```

zookeeper3 节点:

```
[root@zookeeper3 bin]# ./kafka-server-start.sh -daemon ../config/server.properties
```

```
[root@zookeeper3 bin]# jps
```

```
11080 Kafka
```

```
10188 QuorumPeerMain
```

```
11149 Jps
```

（4）测试服务

在 zookeeper1 节点，进入 kafka_2.11-1.1.1/bin 目录下，创建 topic 命令如下。

zookeeper1 节点:

```
[root@zookeeper1 bin]# ./kafka-topics.sh --create --zookeeper 172.16.51.23:2181
--replication-factor 1 --partitions 1 --topic test

Created topic "test".
```

如果成功的话，会输出 “Created topic "test".”。

查看 topic，虽然 topic 是在 172.16.51.23 上创建的，但是在其他机器上也能看到。例如在任意启动的机器 kafka_2.11-1.1.1/bin 的目录中执行命令如下：

zookeeper2 节点:

```
[root@zookeeper2 bin]# ./kafka-topics.sh --list --zookeeper 172.16.51.32:2181

test
```

zookeeper3 节点:

```
[root@zookeeper3 bin]# ./kafka-topics.sh --list --zookeeper 172.16.51.41:2181

test
```

测试成功。