**Python与Kafka集群的交互**

**搭建Kafka集群环境**

先从容易的开始吧。首先我们需要搭建Kafka的集群环境。可以参考[http://www.cnblogs.com/luotianshuai/p/5206662.html](http://www.cnblogs.com/luotianshuai/p/5206662.html" \t "_blank)

在配置完成后，写两个脚本用于自动启动和关闭所有节点上的Kafka实例。

start\_kafka\_all.sh

#!/bin/sh

SERVERS="amaster anode1 anode2"

start\_kafka\_all() {

for SERVER in $SERVERS

do

ssh $SERVER "source ~/.bashrc;/root/apps/kafka\_2.12-0.10.2.0/bin/kafka-server-stop.sh"

ssh $SERVER "source ~/.bashrc;/root/apps/kafka\_2.12-0.10.2.0/bin/kafka-server-start.sh -daemon /root/apps/kafka\_2.12-0.10.2.0/config/server.properties"

done

}

start\_kafka\_all

stop\_kfaka\_all.sh

#!/bin/sh

SERVERS="amaster anode1 anode2"

stop\_kafka\_all() {

for SERVER in $SERVERS

do

ssh $SERVER "source ~/.bashrc;/root/apps/kafka\_2.12-0.10.2.0/bin/kafka-server-stop.sh"

done

}

stop\_kafka\_all

**使用Python与Kafka集群交互**

使用下面的代码安装kafka-python:

pip install kafka-python

然后写一个生产者程序。

#-\*- coding: utf-8 -\*-

import json

from kafka import KafkaProducer

def func():

producer = KafkaProducer(value\_serializer=lambda m: json.dumps(m).encode('utf-8'),

bootstrap\_servers=['amaster:9092','anode1:9092','anode2:9092'])

for \_ in range(100):

producer.send('json-topic', {'ip': '117.90.1.130','port':'9000','anonymity':'高匿名','type':'HTTP,HTTPS','location':'中国 江苏省 镇江市 电信'})

producer.close()

if \_\_name\_\_ == '\_\_main\_\_':

func()

代码很简单，没什么需要解释的。

然后写一个客户端。

#-\*- coding: utf-8 -\*-

import json

from kafka import KafkaConsumer

def func():

consumer = KafkaConsumer('json-topic',

group\_id='group3',

bootstrap\_servers=['amaster:9092'],

auto\_offset\_reset='earliest', enable\_auto\_commit=False)

for message in consumer:

# message value and key are raw bytes -- decode if necessary!

# e.g., for unicode: `message.value.decode('utf-8')`

print message

if \_\_name\_\_ == '\_\_main\_\_':

func()

有一个地方需要解释一下，auto\_offset\_reset='earliest', enable\_auto\_commit=False 这两个参数的设置是指不向ZooKeeper集群提交进度信息，所以下次启动的时候还可以获得最前面的消息。

但是这个跟控制台的客户端不同，这个是没办法回到最开始的，也就是说只能回到最后记录的那个位置。这点需要注意。

输出结果如下：

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

{u'ip': u'117.90.1.130', u'anonymity': u'\u9ad8\u533f\u540d', u'type': u'HTTP,HTTPS', u'location': u'\u4e2d\u56fd \u6c5f\u82cf\u7701 \u9547\u6c5f\u5e02 \u7535\u4fe1', u'port': u'9000'}

看着有点乱，但是是正确的，只是显示Unicode编码而已。