# [Kafka实战－KafkaOffsetMonitor](http://www.cnblogs.com/smartloli/p/4562551.html)

# 1.概述

　　前面给大家介绍了Kafka的背景以及一些应用场景，并附带上演示了Kafka的简单示例。然后，在开发的过程当中，我们会发现一些问题，那就是消息的监控情况。虽然，在启动Kafka的相关服务后，我们生产消息和消费消息会在终端控制台显示这些记录信息，但是，这样始终不够友好，而且，在实际开发中，我们不会有权限去一直观看终端控制台，那么今天就为大家来介绍Kafka的一个监控系统——KafkaOffsetMonitor。下面是今天所分享的目录内容：

* **KafkaOffsetMonitor简述**
* **KafkaOffsetMonitor安装部署**
* **KafkaOffsetMonitor运行预览**

　　下面开始今天的内容分享。

# 2.KafkaOffsetMonitor简述

　　KafkaOffsetMonitor是有由Kafka开源社区提供的一款Web管理界面，这个应用程序用来实时监控Kafka服务的Consumer以及它们所在的Partition中的Offset，你可以通过浏览当前的消费者组，并且每个Topic的所有Partition的消费情况都可以观看的一清二楚。它让我们很直观的知道，每个Partition的Message是否消费掉，有木有阻塞等等。

　　这个Web管理平台保留的Partition、Offset和它的Consumer的相关历史数据，我们可以通过浏览Web管理的相关模块，清楚的知道最近一段时间的消费情况。

　　该Web管理平台有以下功能：

* **对Consumer的消费监控，并列出每个Consumer的Offset数据**
* **保护消费者组列表信息**
* **每个Topic的所有Partition列表包含：Topic、Pid、Offset、LogSize、Lag以及Owner等等**
* **浏览查阅Topic的历史消费信息**

　　这些功能对于我们开发来说，已经绰绰有余了。

# 3.KafkaOffsetMonitor安装部署

## 3.1下载

　　在安装KafkaOffsetMonitor管理平台时，我们需要先下载其安装包，其资源可以在Github上找到，考虑到Github访问的限制问题，我将安装包上传到百度云盘：<http://pan.baidu.com/s/1sj0YERV>

## 3.2安装部署

　　KafkaOffsetMonitor的安装部署较为简单，所有的资源都打包到一个JAR文件中了，因此，直接运行即可，省去了我们去配置。这里我们可以新建一个目录单独用于Kafka的监控目录，我这里新建一个kafka\_monitor文件目录，然后我们在准备启动脚本，脚本内容如下所示：

**#! /bin/bash**

**java -cp KafkaOffsetMonitor-assembly-0.2.0.jar \**

**com.quantifind.kafka.offsetapp.OffsetGetterWeb \**

**--zk node01:2181,node02:2181,node03:2181 \**

**--port 8089 \**

**--refresh 10.seconds \**

**--retain 1.days**

　　给大家解释以下这条启动命令的含义，首先我们需要指明运行Web监控的类，然后需要用到ZooKeeper，所有要填写ZK集群信息，接着是Web运行端口，页面数据刷新的时间以及保留数据的时间值。

## 3.3启动

　　接下来，我们开始启动，启动步骤如下所示：

* **步骤1:启动ZK（DN1～DN3节点）**

**zkServer.sh start**

* **步骤2:启动Kafka服务（集群依次输入以下命令启动）**

**kafka-server-start.sh config/server.properties &**

* **步骤3:启动Web监控服务(如果没有写脚本的话，否则直接启动脚本)**

**java -cp KafkaOffsetMonitor-assembly-0.2.0.jar \**

**com.quantifind.kafka.offsetapp.OffsetGetterWeb \**

**--zk dn1:2181,dn2:2181,dn3:2181 \**

**--port 8089 \**

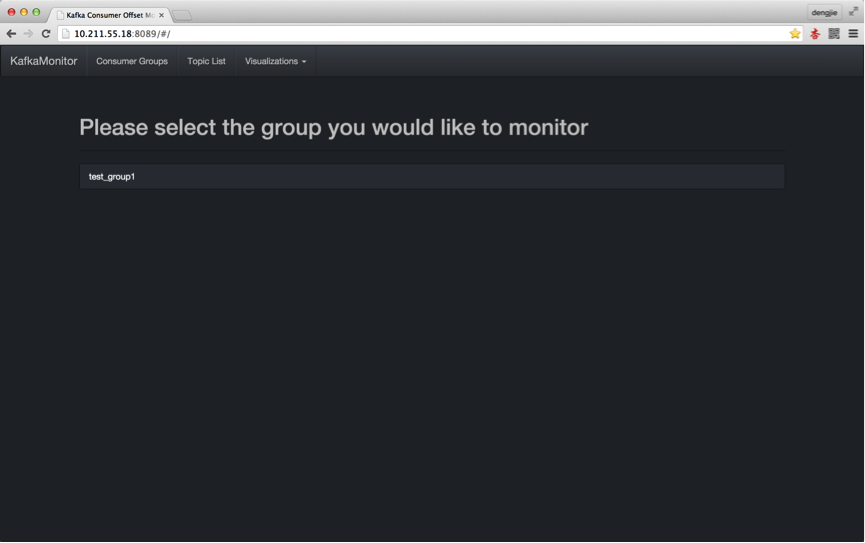
**--refresh 10.seconds \**

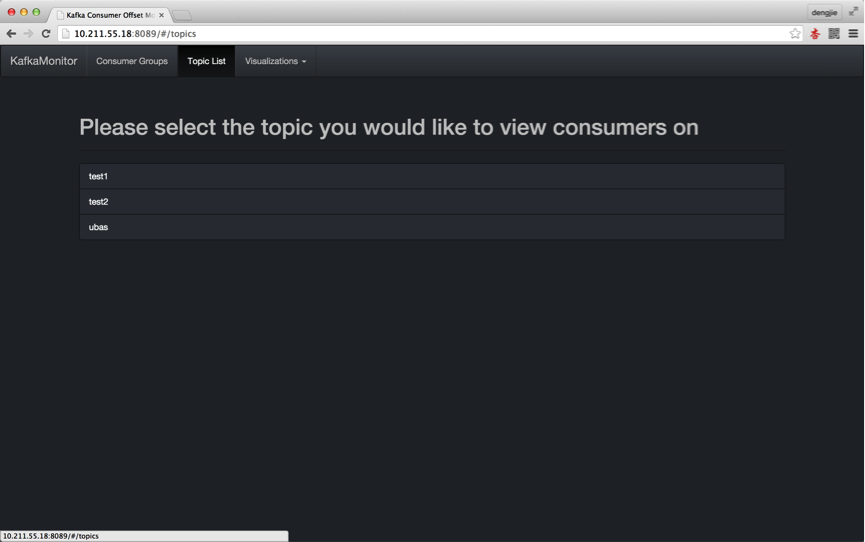
**--retain 1.days**

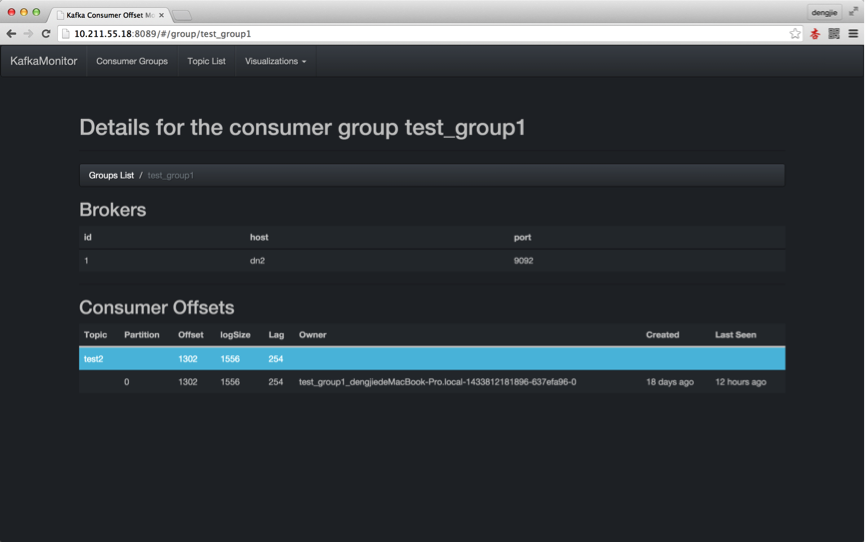
　　Web服务启动成功后，如下图所示：

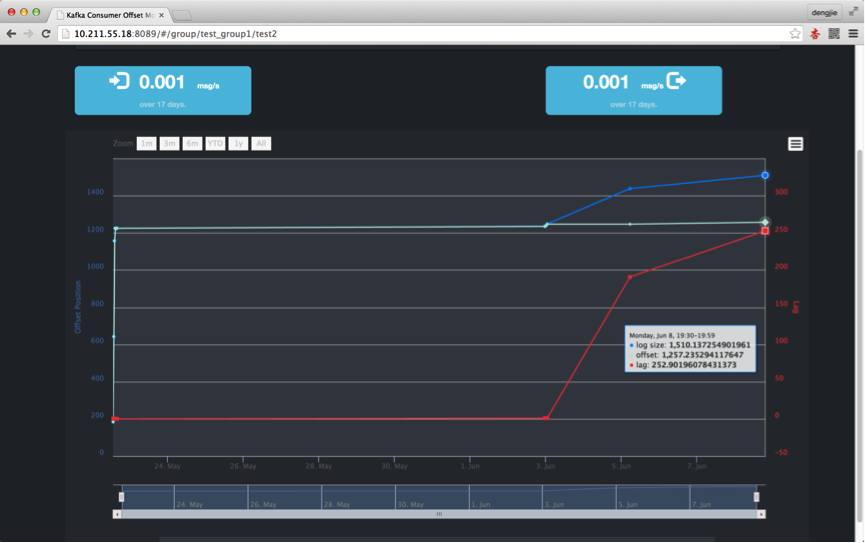
# 4.KafkaOffsetMonitor运行预览

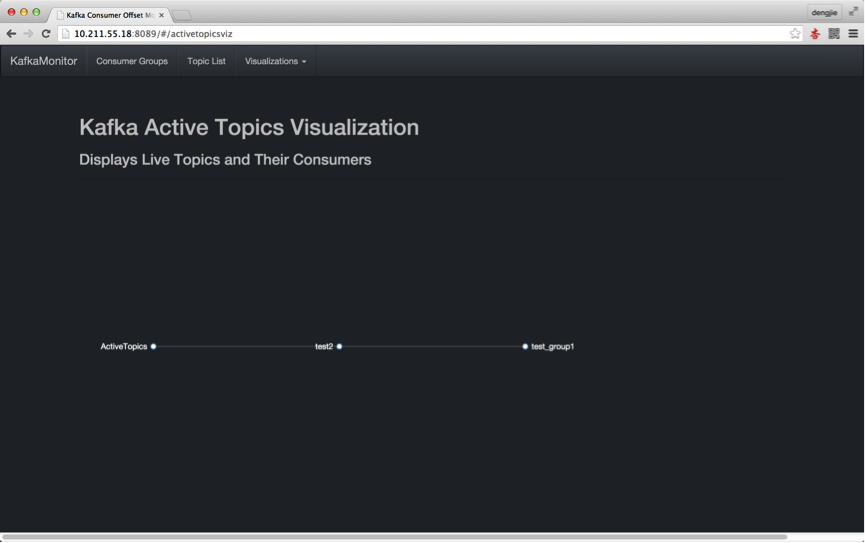
　　下面，我们来使用Kafka代码生产消费一些消息，使用Web监控来浏览消息情况。生产的代码大家可以参考前面我写的《[Kafka实战－简单示例](http://www.cnblogs.com/smartloli/p/4543211.html)》，这里直接预览演示结果，如下图所示：

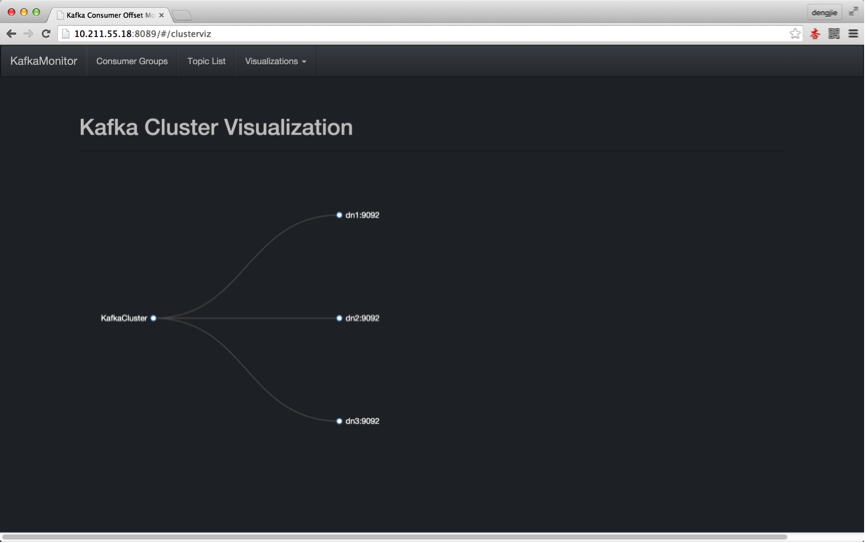












# 5.总结

　　在运行KafkaOffsetMonitor的JAR包时，需要确保启动参数的配置正确，以免启动出错，另外，Github的上的KafkaOffsetMonitor的JAR中的静态资源有些链接用到了Google的超链接，所有如果直接只用，若本地木有代理软件会启动出错，这里使用我所提供的JAR，这个JAR是经过静态资源改版后重新编译的使用本地静态资源。

　　另外图中的一些参数的含义如下：

* **Topic：创建Topic名称**
* **Partition：分区编号**
* **Offset：表示该Parition已经消费了多少Message**
* **LogSize：表示该Partition生产了多少Message**
* **Lag：表示有多少条Message未被消费**
* **Owner：表示消费者**
* **Created：表示该Partition创建时间**
* **Last Seen：表示消费状态刷新最新时间**