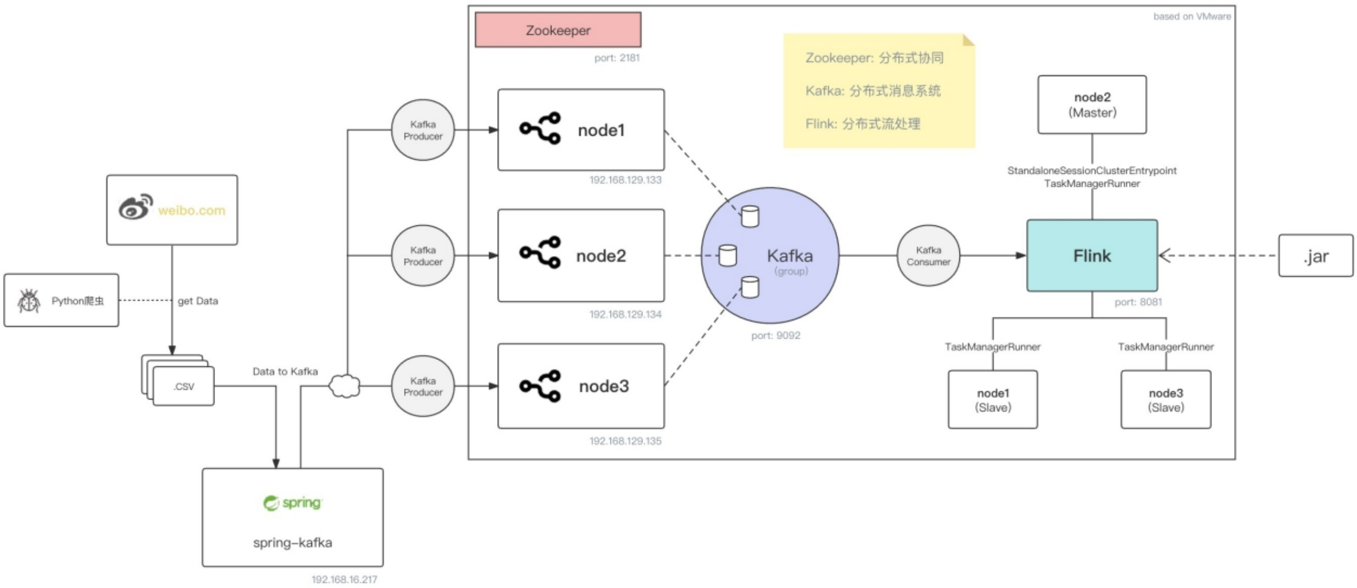

# 配置相关

**操作系统**：CentOS7

**docker容器**：node1，node2，node3

**Zookeeper**：node1 (Follower)，node2 (Follower)，node3 (Leader)

**Kafka**：node1，node2，node3

**Flink**：node1 (TaskExecutor)，node2 (Standalonesession, TaskExecutor)，node3 (TaskExecutor)

**keys**：root:123123

# 测试步骤

1. 分别导入三个Docker容器：

```
docker import node1.ova docker-node1:node1
```

```
docker import node2.ova docker-node2:node2
```

```
docker import node3.ova docker-node3:node3
```

2. 分别启动三个容器

```
docker run -it docker-node1:node1 /bin/bash
docker run -it docker-node2:node2 /bin/bash
```

```
docker run -it docker-node3:node3 /bin/bash
```

3. 分别在三个容器中启动Zookeeper、Kafka

   ```
   su root
   systemctl stop firewalld
   zkServer.sh start
   kafka-server-start.sh config/server.properties
   ```

4. 创建topic：weibo、flinkresult

   ```
   kafka-topics.sh --bootstrap-server node1:9092,node2:9092,node3:9092 --partition 3 --topic weibo
   kafka-topics.sh --bootstrap-server node1:9092,node2:9092,node3:9092 --partition 3 --topic flinkresult
   ```

5. 在node1开启了一个weibo (topic)的consumer用于观察流式数据的输入(new terminal)

   ```
   kafka-console-consumer.sh --bootstrap-server node1:9092,node2:9092,node3:9092 --topic weibo
   ```

6. 在node3开启了一个flinkresult (topic)的consumer用于观察Flink输出结果(new terminal)

   ```
   kafka-console-consumer.sh --bootstrap-server node1:9092,node2:9092,node3:9092 --topic flinkresult
   ```

7. 在node2中启动flink

   ```
   start-cluster.sh
   ```

8. 进入Flink Web UI：localhost:8081 --> submit new job --> Add new --> 选择**jar包** --> submit
9. 在主机端解压**kafkaProducer.zip**，指定kafkaProducer.java中的filePath为**data_test.csv**路径，并运行该Java项目用于模拟流式数据，并作为Kafka Producer向weibo topic中发送数据
10. 在node3中观察输出结果（flinkresult topic）

# 其他

docker镜像是用VMware虚拟机上导出的ova文件创建的，在部署时若有任何问题请联系我！

微信：ddtheluckyone

邮箱：xydong@stu.pku.edu.cn