

一、Nifi 配置

1) nifi 基本配置

1. jdk 配置: nifi 新版本要求 jdk 版本必须 1.8 以上
目录: bin/nifi.env.sh
export JAVA_HOME=/home/user/opt/jdk1.8.0_144
忽略此项配置, nifi将使用系统默认Java环境启动。
2. nifi运行内存配置
目录: conf/bootstarp.conf
java.arg.2=-Xms512m
java.arg.3=-Xmx20480m
根据服务器性能合理设置nifi内存
3. nifi日志配置
目录: conf/logback.xml
4. nifi端口配置
目录: conf/nifi.properties
nifi.web.http.port=18083
默认为8080端口。
5. nifi基本命令
nifi.sh start 启动
nifi.sh stop 关闭
nifi.sh restart 重启
nifi.sh status 状态

以上基本配置完成, 可单节点模式启动 nifi。

2) nifi 集群配置

1. 配置 zookeeper
目录: conf/zookeeper.properties
clientPort=2181 zookeeper 端口

server.1=kube-master:2888:3888
server.2=kube-slave1:2888:3888
server.3=kube-slave2:2888:3888
zookeeper 服务器设置, nifi 集群有多少节点需要配置 server
server.1 中的数字 1 为服务器的 ID, 需要与 myid 文件中的 id 一致, 下一步将配置 myid。
2. 配置 myid

在 nifi 目录下创建文件夹/state/zookeeper/并创建文件 myid
文件内容与第一步中的 server.id 一致。



3. 配置 state-management.xml

目录: conf/state-management.xml

```
<cluster-provider>
  <id>zk-provider</id>
```

```
<class>org.apache.nifi.controller.state.providers.zookeeper.ZooKeeperStateProvider</class>
  <property name="Connect String">kube-master:2182,kube-slave1:2182,kube-
slave2:2182,kube-slave3:2182</property>
  <property name="Root Node">/nifi</property>
  <property name="Session Timeout">10 seconds</property>
  <property name="Access Control">Open</property>
</cluster-provider>
```

Connect String 配置为 nifi 节点的 ip 加 zookeeper 端口号，以逗号分隔

4. 配置 nifi 节点属性

目录: conf/nifi.properties

```
nifi.state.management.embedded.zookeeper.start=true
```

使用内置 zookeeper

```
nifi.zookeeper.connect.string=kube-master:2182,kube-slave1:2182,kube-
slave2:2182,kube-slave3:2182
```

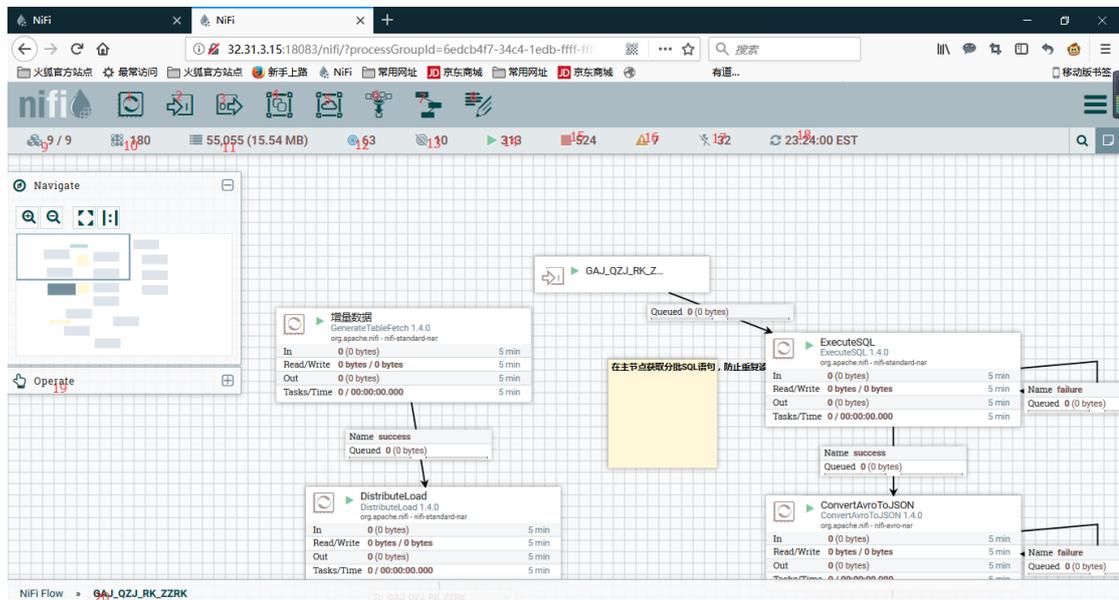
与上一步中的 connect string 一致。

下面配置需每个节点单独配，根据节点的 IP 相应配置，保持集群中节点使用的端口一致

```
nifi.cluster.is.node=true
nifi.cluster.node.address=kube-slave1
nifi.cluster.node.protocol.port=9999
nifi.remote.input.host=kube-slave1
nifi.remote.input.secure=false
nifi.remote.input.socket.port=9998
nifi.web.http.host=kube-slave1
```

配置完成后即可依次启动节点，集群将选举产生主节点。

二、Nifi 界面



1-8 为 nifi 提供的组件可以拖到下方的画布上使用。

- 1: processor : nifi 提供的多种处理器。
- 2: input port: flowfile 数据进入节点。
- 3: output port flowfile 数据输出节点。
- 4: group 组, 可以将多种处理器放入同一个组中, 组与组之间可以通过 input port 和 output port 传输 flowfile 数据。
- 5: Remote process group: 远程进程组, 可将 flowfile 发送至别的节点。
- 6: Funnel: 汇聚 flowfile 数据。
- 7: Template: 可拖放导入的模板配置。
- 8: Leble: 文字板

9-18 为 nifi 的基本状态栏

- 9: nifi 集群节点状态。9/9 表示有集群中有九个节点, 并九个节点都在线。
- 10: nifi 当前运行线程数。
- 11: nifi 当前队列中待执行的 flowfile 的 size 及大小
- 12: nifi 启动的远程进程组。
- 13: nifi 未启动的远程进程组。
- 14: nifi 启动的处理器
- 15: nifi 未启动的处理器。
- 16: nifi 配置不可用的处理器
- 17: nifi 废弃不用处理器。
- 18: nifi 界面最新刷新时间。
- 19: nifi 画布缩略图及操作按钮。
- 20: 当前位置。

1) 总览界面

Summary 摘要包含处理器，GROUP 等一系列整体摘要信息。

Counters 计数器界面，需要配合计数器处理器（UpdateCounter）使用。

Bulletin Board nifi 通讯信息页面。

Date Provenance flowfile 数据总览。

Controller Settings 设置界面，可以配置线程数等一些设置。

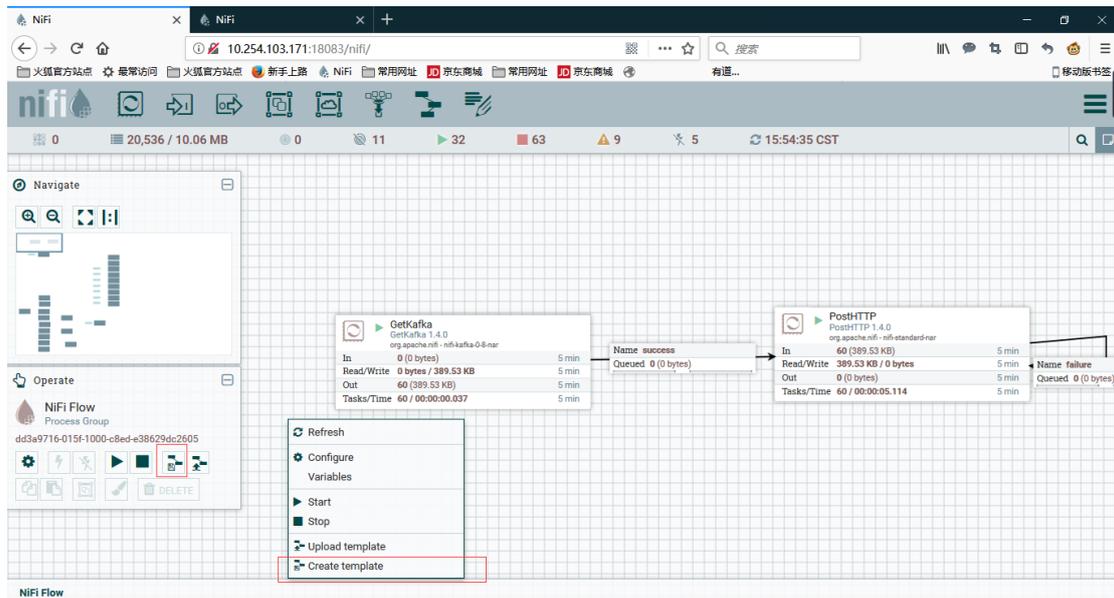
Flow conf History nifi 配置历史。

Cluster nifi 集群模式时，可以查看集群状态，管理节点。

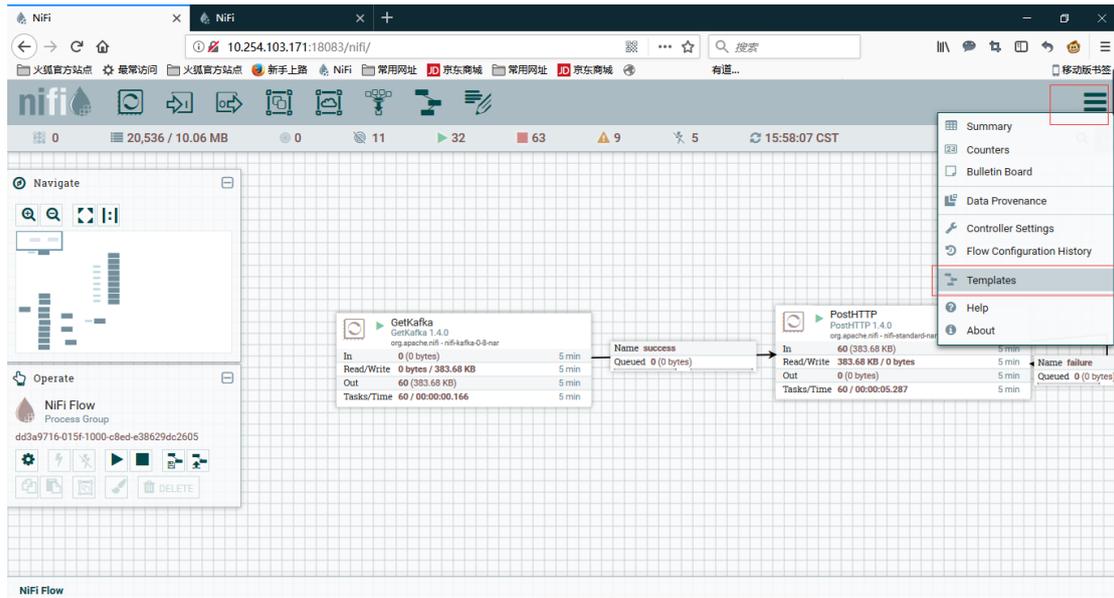
三、Nifi 使用

1) 模板

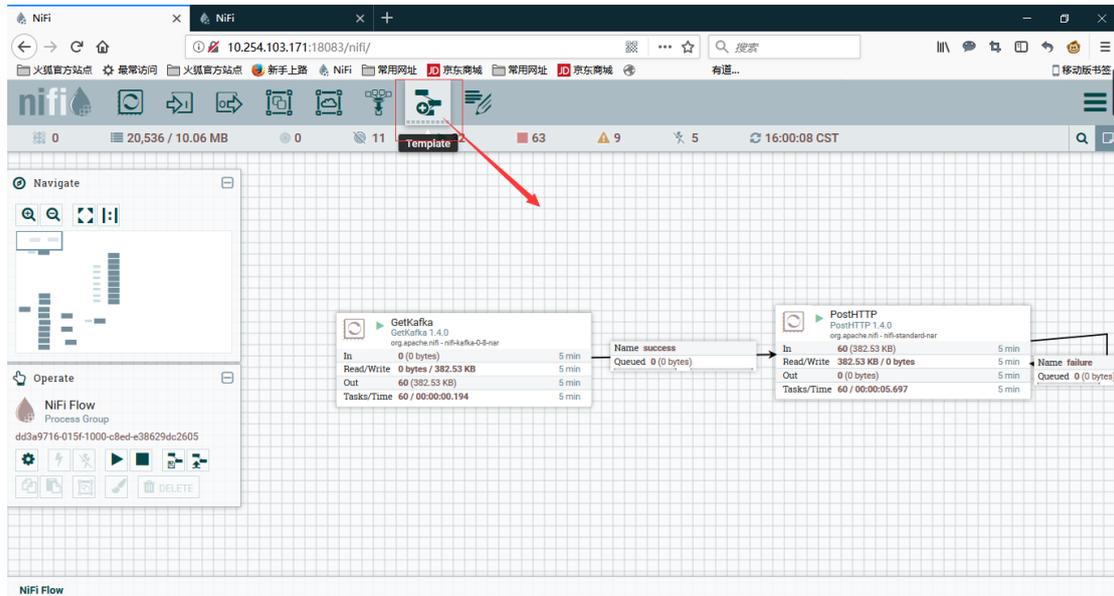
创建模板：在要创建模板的 group 中点击面板左侧的 create template 或者鼠标右键画布空白处。



下载模板：



使用模板：选择界面上分的 template 拖放至画布，选择要使用的模板。



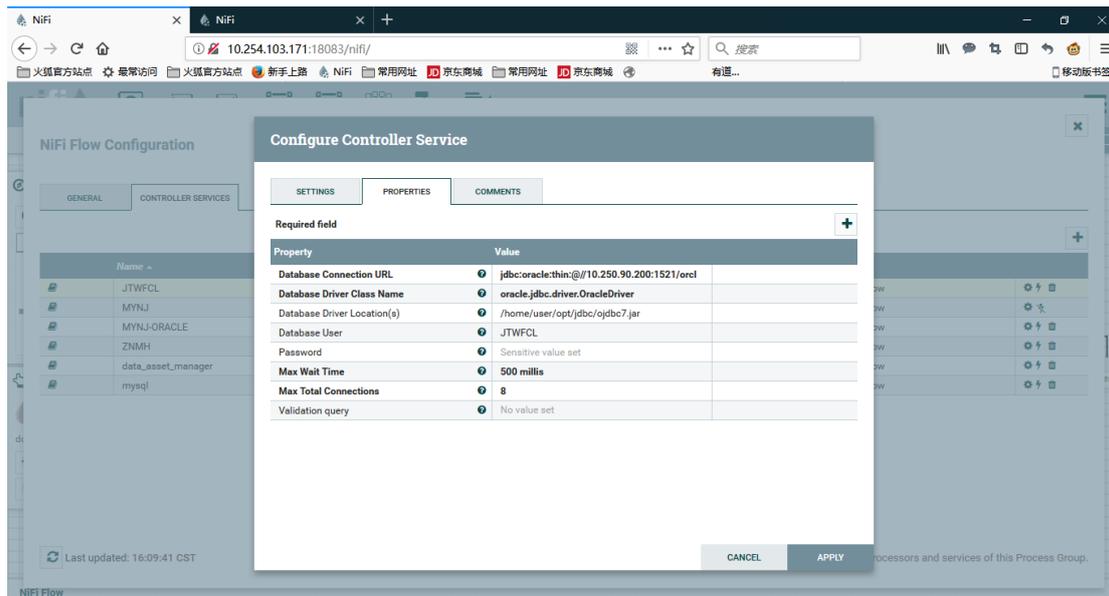
Nifi 的模板会保存组中的处理器配置及 CONTROLLER SERVERS。例如数据库连接，但是不会保存密码。

2) CONTROLLER SERVERS

Nifi 中的控制器，例如数据库连接，XML 读取，JSON 读取器。

Nifi 中子 group 可以继承父 group 中的控制器。建议在根画布及 new flow 中配置控制器。每个控制器都是一个实例，如数据库连接，交给同一个控制器管理更为节省资源及方便。在空白处右键点击 CONFIGURE：

下图为一个 DBCPConnectionPool 数据库连接控制器的配置说明：

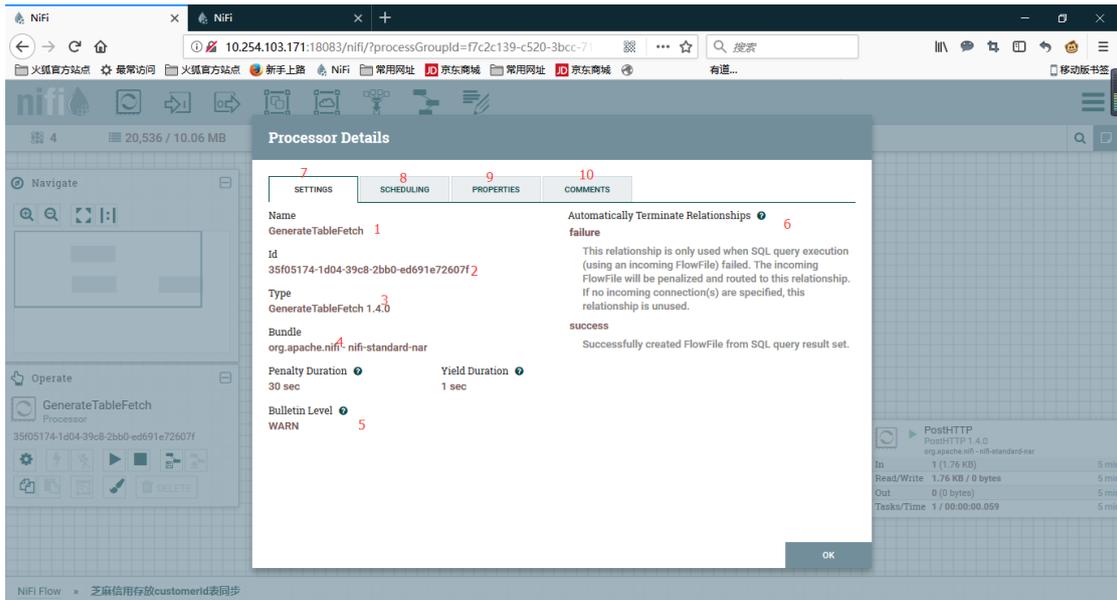


依次为：

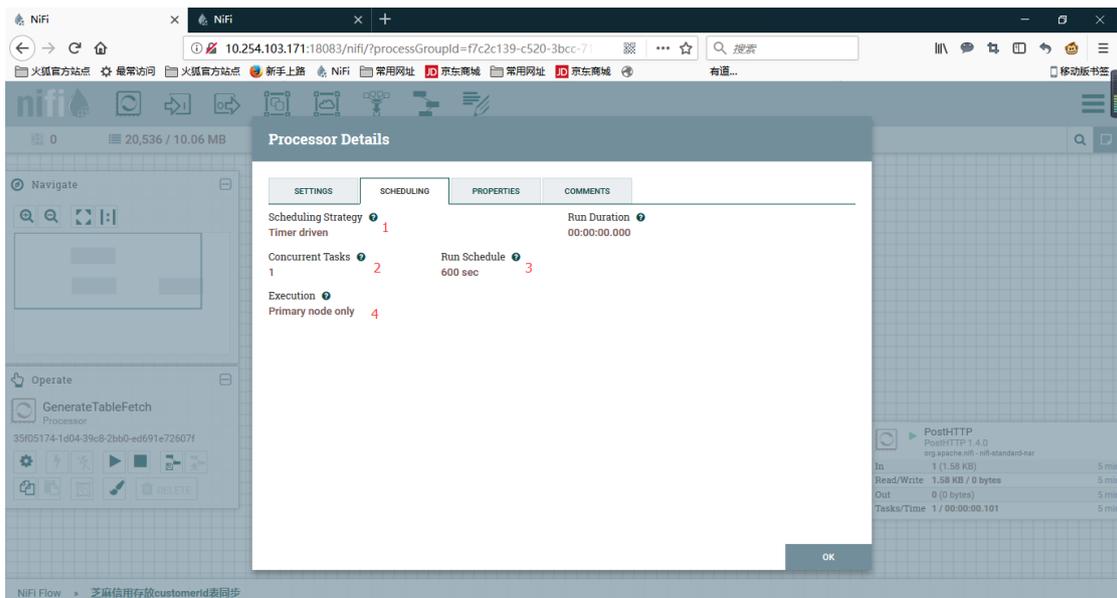
1. 数据库连接的 URL，类似于 JDBC 中的 URL。
2. 数据库驱动类名
3. 数据库驱动类目录（jar 包需要放置在 nifi 可访问到的地址，集群需要每个节点都上传 jar 包）。
4. 数据库用户名
5. 密码
6. 最大等待时间
7. 最大连接数
8. 测试 SQL。

3) PROCESSOR

Nifi 处理器：

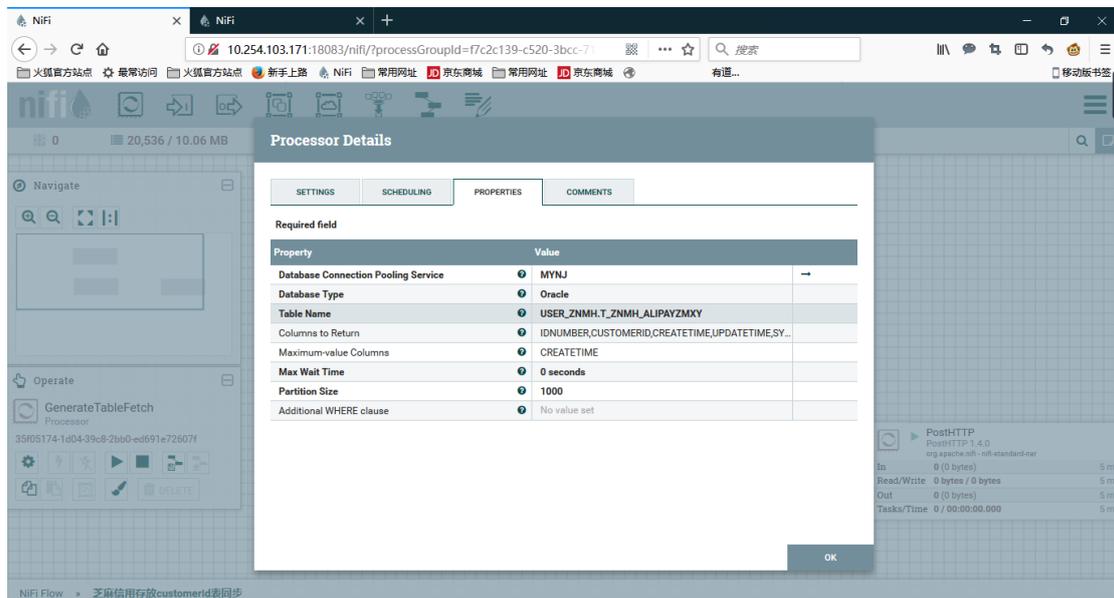


- 1: 处理器名字
- 2: 处理器的 ID
- 3: 处理器类型
- 4: 处理器所属 nar 包
- 5: 处理器输出日志级别
- 6: 处理器输出的关系，配置处理器时所有关系需要处理：拖放给下一处理器，或者选择终止关系（配置处理器时选中要自动终止的关系）。
- 7: 处理器基础设置
- 8: 处理器定时设置
- 9: 处理器属性
- 10: 备注



- 1: 处理器定时任务类型 1.timer 2.cron
- 2: 单个节点处理器并发数，如图单节点并发数为 1，集群为 9 个节点时并发数为 9
- 3: 根据定时任务类型不同配置不同，图中为 600 秒执行一次

4: 当 nifi 为集群模式时: 可以配置处理器为主节点运行或所有节点运行。一般获取数据的处理器使用主节点运行, 防止重复消费。

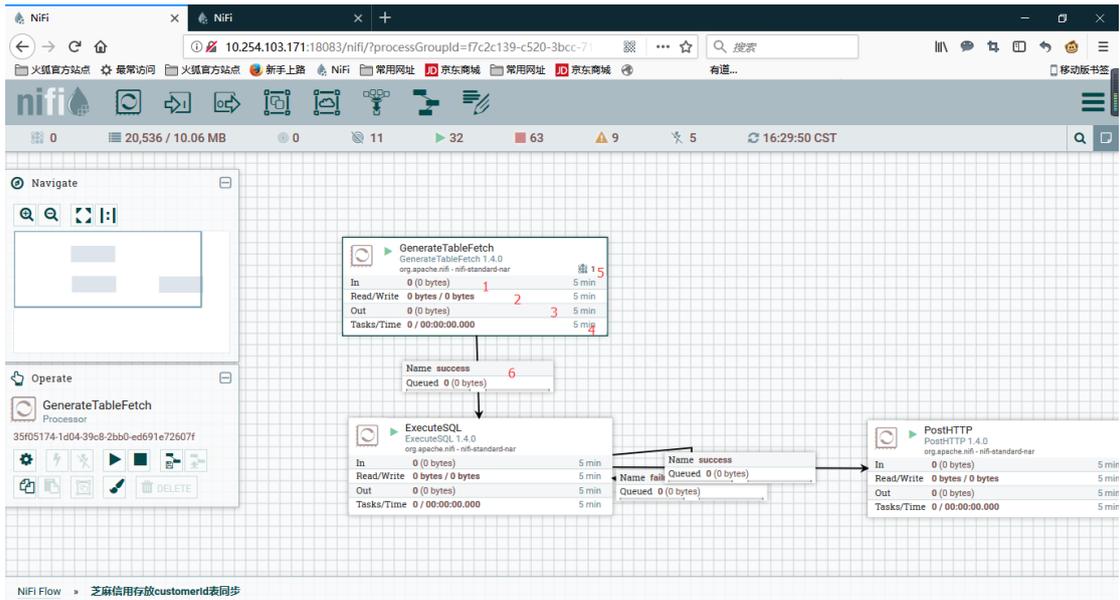


处理器属性: 某些属性需配合 CONTROLLER SERVERS 使用, 如图中的数据库连接。

依次配置项为:

- 1: 数据库连接
- 2: 数据库类型
- 3: 表名
- 4: 需要返回的字段
- 5: 最大值列, 可依据此列完成增量同步
- 6: 最大等待时间 0S 表示一直等待。
- 7: 分页条数, 1000 表示每条 SQL 查询出 1000 条记录。
- 8: 查询时的 where 条件。

GenerateTableFetch 拥有 state 属性: 记录 Maximum-value Columns 的值。



- 1: 处理器五分钟内接受到的 flowfile 数据。
- 2: 处理器五分钟读取和写的文件大小
- 3: 处理器五分钟输出的数据
- 4: 处理器五分钟内的执行次数和时间。
- 5: 处理器当前使用的线程数。如图为一个线程在运行。
- 6: 处理器的 SUCCESS 关系输出值 ExecuteSQL 处理器。其中 failure 关系为自动终止。

点击处理器左键可以查看 Data Provenance
Data Provenance 为一天内通过处理器的数据记录。

