链滴

数据 ETL 平台 Streamsets 简单应用

作者：Gaoshengyue

原文链接：https://ld246.com/article/1576837366412

来源网站：链滴

<h2 id="StreamSets介绍">StreamSets 介绍</h2>

320.png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220162320-cfbf
0e0.png?imageView2/2/interlace/1/format/jpg">
StreamSets 是最近兴起的 ETL 平台，它的特点是具有多样性的组件支持，可集成于 CDH 平台
最为吸引人的就是可视化的数据流通流程设置，多个 pipelines 的编写，RestApi 形式的自动化支持
等等等等，当然选择使用它的最终理由还是因为支持的组件多。这一篇就简单来讲讲 StreamSets 的
单使用。
<h2 id="StreamSets部署">StreamSets 部署</h2>
StreamSets 的部署有很多形式，这里列举两个最方便的。
<h3 id="Docker-compose">Docker-compose</h3>
这里我提供一份写好的 docker-compose 文件，如下
<pre><code class="language-yaml highlight-chroma"><span c
ass="highlight-cl">version:<
span> '3.1'<span cl
ss="highlight-w">
<span cla
s="highlight-w">services<span class="highligh
-p">:
<span cla
s="highlight-w"> streamsets<span class="high
ight-p">:
<span cla
s="highlight-w">   image<span class="highlig
t-p">: streamsets/da
acollector
<span cla
s="highlight-w">   restart<span class="highlig
t-p">: always</span

<span cla
s="highlight-w">   ports<span class="highligh
-p">:
<span cla
s="highlight-w">   - "18630:18630"<span cla
s="highlight-w">
<span cla
s="highlight-w">   environment<span class="h
ghlight-p">:
<span cla
s="highlight-w">   HOST_IP<span class="high
ight-p">: 0.0.0.0</s
an>
<span cla
s="highlight-w">   volumes<span class="highl
ght-p">:
<span cla
s="highlight-w">   - streamsets:/opt/steramsets</span

<span cla
s="highlight-w">   - /data/mdt/data_hbase:/data/hbas

<span cla
s="highlight-w">volumes<span class="highligh

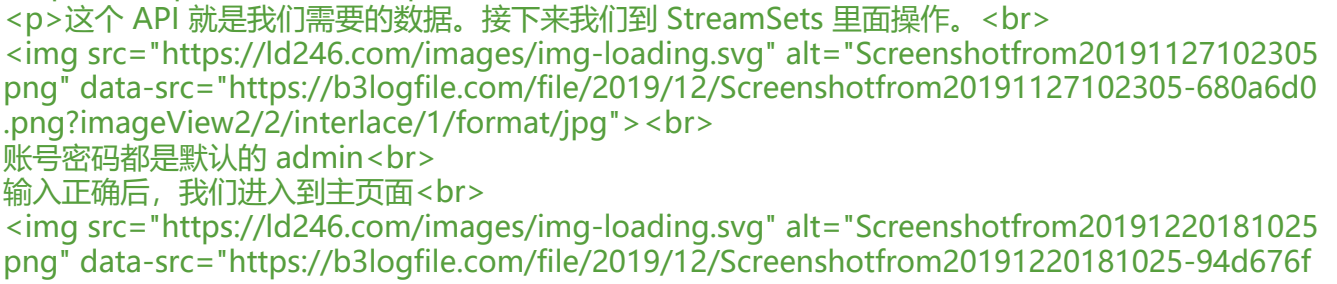
-p">:
<span cla
s="highlight-w"> streamsets<span class="high
ight-p">:
</code></pre>
这里把端口映射到了 18630 方便等下的登录使用，HOST_IP 改成万能 IP，不设置访问限制方便
问，相关数据挂到本地磁盘。
启动命令
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl">sudo docker-compose up -d
</code></pre>
直接复制 yaml 配置创建 docker-compose.yaml 就可以了
<h3 id="K8S">K8S</h3>
因为 StreamSets 内部的 data collector 重启机制问题，会导致组件无法成功添加的问题，所以
里还没有解决，暂不推荐 K8S，不过还是把配置文件给出来(组件无法附加版本)
<pre><code class="language-yaml highlight-chroma"><span c
ass="highlight-cl">apiVersion<span class="highlight-p
>: apps/v1<
pan class="highlight-w">
<span cla
s="highlight-w">kind<span class="highlight-p
>: Deployment</spa
>
<span cla
s="highlight-w">metadata<span class="highli
ht-p">:
<span cla
s="highlight-w"> name<span class="highlight
p">: streamsets-data
ollector
<span cla
s="highlight-w"> namespace<span class="hig
light-p">: default</s
an>
<span cla
s="highlight-w"> labels<span class="highlight
p">:
<span cla
s="highlight-w">   app<span class="highlight
p">: streamsets-data
ollector
<span cla
s="highlight-w">spec<span class="highlight-p
>:
<span cla
s="highlight-w"> replicas<span class="highlig
t-p">: 1<s
an class="highlight-w">
<span cla
s="highlight-w"> selector<span class="highlig
t-p">:
<span cla
s="highlight-w">   matchLabels<span class="h
ghlight-p">:

```
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">      </span><span class="highlight-nt">app</span><span class="highlight
p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets-data
ollector</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">   </span><span class="highlight-nt">template</span><span class="highli
ht-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">     </span><span class="highlight-nt">metadata</span><span class="high
ight-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">      </span><span class="highlight-nt">labels</span><span class="highli
ht-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">        </span><span class="highlight-nt">app</span><span class="highlig
t-p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets-da
acollector</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">     </span><span class="highlight-nt">spec</span><span class="highlight
p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">      </span><span class="highlight-nt">containers</span><span class="hi
hlight-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">      </span>- <span class="highlight-nt">name</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets
/span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">        </span><span class="highlight-nt">image</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets/
atacollector</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">        </span><span class="highlight-nt">env</span><span class="highlig
t-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">        </span>- <span class="highlight-nt">name</span><span class="high
ight-p">:</span><span class="highlight-w"> </span><span class="highlight-l">HOST_IP</
pan><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">          </span><span class="highlight-nt">value</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-m">0.0.0.0</s
an><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">        </span><span class="highlight-nt">resources</span><span class="h
ghlight-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">          </span><span class="highlight-nt">limits</span><span class="highl
ght-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">            </span><span class="highlight-nt">cpu</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-s1">'1'</span
<span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">            </span><span class="highlight-nt">memory</span><span class="
```

ighlight-p">: 2Gi
          requests:
            cpu: '1'
            memory: 2Gi
        volumeMounts: #容器内挂载点
          - mountPath: /opt/steramsets
            name: stp
          - mountPath: /data/hbase
            name: sthb
      volumes:
      - name: stp
        hostPath:
          path: streamsets
      - name: sthb
        hostPath:

```
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">          </span><span class="highlight-nt">path</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-l">/data/mdt/
ata_hbase</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-nn">---</span><span class="highlight-w"

</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-nt">apiVersion</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-l">v1</span>
span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-nt">kind</span><span class="highlight-p
>:</span><span class="highlight-w"> </span><span class="highlight-l">Service</span><s
an class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-nt">metadata</span><span class="highli
ht-p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span><span class="highlight-nt">labels</span><span class="highlight
p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">    </span><span class="highlight-nt">app</span><span class="highlight
p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets-data
ollector</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span><span class="highlight-nt">name</span><span class="highlight
p">:</span><span class="highlight-w"> </span><span class="highlight-l">streamsets-data
ollector</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span><span class="highlight-nt">namespace</span><span class="hig
light-p">:</span><span class="highlight-w"> </span><span class="highlight-l">default</s
an><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w"></span><span class="highlight-nt">spec</span><span class="highlight-p
>:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span><span class="highlight-nt">type</span><span class="highlight-
">:</span><span class="highlight-w"> </span><span class="highlight-l">NodePort</span
<span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span><span class="highlight-nt">ports</span><span class="highlight
p">:</span><span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">  </span>- <span class="highlight-nt">port</span><span class="highlight
p">:</span><span class="highlight-w"> </span><span class="highlight-m">18630</span
<span class="highlight-w">
</span></span></span><span class="highlight-line"><span class="highlight-cl"><span cla
s="highlight-w">    </span><span class="highlight-nt">protocol</span><span class="highl
ght-p">:</span><span class="highlight-w"> </span><span class="highlight-l">TCP</span
```

<span cla
s="highlight-w">   targetPort<span class="hig
light-p">: 18630</
pan>
<span cla
s="highlight-w">   nodePort<span class="highl
ght-p">: 31630</s
an>
<span cla
s="highlight-w"> selector<span class="highlig
t-p">:
<span cla
s="highlight-w">   app<span class="highlight
p">: streamsets-data
ollector
</code></pre>
也是把端口暴露在外部 18630
复制下来创建 StreamSets.yaml
启动命令
<pre><code class="highlight-chroma"><span class="highlight
cl">kubectl apply -f StreamSets.yaml -n data-service
</code></pre>
这里是使用 data-service 命名空间，K8S 的概念这里不多说了
<h3 id="CDH">CDH</h3>
这个安装方式我就不多说了吧，因为直接在 cloudera-manager 里面添加角色就行了，具体在
个机器上的角色看自己的需求了。
<h2 id="StreamSets应用">StreamSets 应用</h2>
<h3 id="Pipelines创建">Pipelines 创建</h3>
这里我们简单的举一个 HTTP-client 数据获取的例子。
为了方便起见，我们先创建一个 test.txt 文件，内容可以如下：
<pre><code class="language-text highlight-chroma"><span cl
ss="highlight-cl">name,age
symoon,18
willson,50
</code></pre>
这里用,分割，方便那我们取数据。
当然也可以自己设置测试数据，这里随便写了一个。
接下来我们简单开启一个 python 服务，用来启动一个 api
<pre><code class="language-bash highlight-chroma"><span c
ass="highlight-cl">python3 -m http.server 8001
</code></pre>
这个不多说了，大家应该都知道，这条命令在 txt 所在文件夹启动，这时候访问
<pre><code class="highlight-chroma"><span class="highlight
cl">http://127.0.0.1:8001/test.txt
</code></pre>
这个 API 就是我们需要的数据。接下来我们到 StreamSets 里面操作。


账号密码都是默认的 admin
输入正确后，我们进入到主页面

.png?imageView2/2/interlace/1/format/jpg">
点击 CREATE NEW PIPELINE 创建一个新的任务，Title 什么的随便填，我直接填了 test
创建成功后我们进入到任务管理界面：
<img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom20191220181127
png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220181127-9937bb
7.png?imageView2/2/interlace/1/format/jpg"></p>
<p>我们选择拖拽一个 HTTP Client 作为数据源。相关配置如下填写，这里我们用 docker 启动的 Str
amSets 的化就无法用 127.0.0.1 了，我们使用本机的 IP</p>
<p><img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom2019122018
037.png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220182037-581
5614.png?imageView2/2/interlace/1/format/jpg"></p>
<p><img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom2019122018
509.png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220181509-dca
c72e.png?imageView2/2/interlace/1/format/jpg"></p>
<p><img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom2019122018
059.png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220182059-814
c913.png?imageView2/2/interlace/1/format/jpg"></p>
<p>这里填写的信息都有说明，相信大家都能看懂就不多说了，接下来测试一下，我们点击一下右上
类似眼睛的按钮。
默认信息就好，批量可以从 10 改成 1。结果如图
<img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom20191220182144
png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191220182144-dcf8c387
png?imageView2/2/interlace/1/format/jpg"></p>
<h3 id="Mysql增量采集案例">Mysql 增量采集案例</h3>
<p>这里我们举个简单的增量采集案例，采集 Mysql 指定表中的增量数据到 Kafka 中。
接下来来实战一下，我们先创建一个 pipeline，上面已经说明了，就不多说了。
因为 mysql 的 connector 需要安装 jdbc 组件，所以我们需要进行组件安装。进入 Pipeline 中的界
，以如下方式选择。
<img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom20191221091758
png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191221091758-39499f88
png?imageView2/2/interlace/1/format/jpg"></p>
<p>点击加号 Add/Remove Stages，进入到组件安装界面。
<img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom20191221092001
png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191221092001-7503c1a
.png?imageView2/2/interlace/1/format/jpg"></p>
<p>这里我们找到 JDBC，点击右侧的配置按钮，点击 install。
因为我们需要用到 Kafka 服务，这里我们继续进入 Apache Kafka 中，安装 Kafka-V2.0 版本。安装
功后会提示重启，根据提示点击 Restart data collector 按钮即可。等待一分钟左右服务重启即可。<
p>
<p>重新进入到我们刚才创建的 Pipeline，配置 JDBC 拖动的组件及配置如下：
<img src="https://ld246.com/images/img-loading.svg" alt="Screenshotfrom20191221094547
png" data-src="https://b3logfile.com/file/2019/12/Screenshotfrom20191221094547-22adcf06
png?imageView2/2/interlace/1/format/jpg">
这里我们的数据库名字和表名字分别是 test 和 Test，偏移值由 id 字段来计算。初始值设置 0。
因为 jdbc 需要相应的驱动，所以我们需要在 External Libraries 处上传 mysql 的 jdbc 驱动。这里
接附上一份。
mysql
onnectorjava8.0.18.jar
下载好后上传，依旧需要重启服务生效。
后面还需要提供账号密码，这里就不截图多说了。
这时候依旧是点那个眼睛按钮测试一下，首先我们还是需要在数据库里面创建这张表的，具体格式如
</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight
cl">id|name|age

1|symoon|18
2|willson|19
3|lake|20
</code></pre>
大概这种表格式，测试后结果如下：


这时候我们已经拿到了增量数据，接下来配置 Kafka 的数据接收端。

这里我们在上面的接收端选择直接选择 Kafka producer 就可以了，我使用的是本机 docker 启的 Kafka 具体的配置我就不发了，在之前的一键启动 docker 程序里面有，topic 随便输入一个 test 接下来还是点击眼睛测试一下。

kafka 的 input 成功了，接下来我们直接运行起来。

数据流通的性能还有一些相关信息都在这里显示了，这个任务会一直运行着，增量数据都可以流通到 afka，其他的数据库使用其他连接器也可以支持，这里就不多说了，大家可以都一一去试试。