

次批处理提交 Channel 中的一个事务。

Avro Sink 在通过线路发送数据之前可以压缩数据。为了启用压缩，设置 `compression-type` 的值为 `deflate`。可以使用 `compression-level` 参数控制压缩的级别，该参数的值可以是 1 到 9 之间的任意整数，1 代表最差的压缩，9 代表最好的压缩。值得注意的是，级别越高，在 Sink 压缩和 Source 解压缩所花费的时间越多。默认值是 6。当启用了压缩，Avro Sink 发送压缩数据到的 Avro Source，也应该要设置 `compression-type` 参数的值为 `deflate`。压缩发生在批处理级别，所以在批量被发送之前，事件的整个批量立即被压缩。

Avro Sink 支持使用 SSL 进行加密通信。Source 接收数据也必须被配置为从 Avro Sink 接收加密数据。为了启用加密，应设置 `ssl` 参数的值为 `true`。默认情况下该参数值为 `false`。除非另有指定，Flume 将使用 Java 默认的 JSSE 证书授权文件，`jssecacerts/cacerts`，来判断 Avro Source 的 SSL 证书是否可信。

124

如果要使用自定义 trust store，设置 `truststore` 参数的值为各自的 trust store 文件的路径。运行 Agent 的用户应该已经读取访问完文件。`truststore-password` 参数的值为打开该文件的密码。`truststore-type` 参数设置了 trust store 的类型。默认为 JKS，但可以是任何支持 Java trust store 的类型。如果 Sink 发送数据时不需要检查 Source 的 SSL 证书，设置 `trust-all-certs` 值为 `true`。除了测试该值不应被设为 `true`。一味地信任 Source 的 SSL 证书意味着任何 Source 都可以读取发送的数据——这影响了 SSL 提供的安全。Java 中更多关于 SSL 的细节，以及如何创建 trust store 都可以在 Oracle Java CAPS 文档 [truststores] 查找到。

大多数情况下，Agent 之间在中间利用负载均衡器相互通信。因为 Agent 之间的连接是黏性的，在负载均衡器之后添加托管 Avro Source 的新 Agent，需要托管 Avro Sink 的 Agent 重新启动，以确保新 Agent 接收连接。为了避免必须这样做，用户可以通过设置 `reset-connection-interval` 参数，强制 Avro Sink 定期终止它们到 Agent 的连接并重新连接。第 3 章在“Avro Source”一节中展示了 Avro Sink 配置的一个例子。

Thrift Sink

Thrift Sink 可以用来在 Flume Agent 之间使用 Thrift RPC 进行通信。通常 Thrift Sink 与 Avro Sink 工作方式很像，但是它缺少压缩和 SSL 功能。在 Flume Agent 之间推荐使用 Avro RPC 进行通信。Thrift Sink 应该用来写数据到已经运行的 Thrift Source，或许从使用 Thrift RPC 客户端的其他语言写的 RPC 客户端接收数据。

表 5-8 展示了 Thrift Sink 的配置参数。