

如果事件被成功地写出，Source 返回 `Status.OK` 状态，表明数据被提交给 Channel。

当 Agent 停止或被重新配置时，调用 `stop` 方法。这个调用授权了 `doStop` 方法，这个方法必须做一些所需的清理工作。在这种情况下，该方法停止 `NettyServer` 实例，并在返回前等待它终止。如果 Event-driven Source 启动线程或创建线程池，或打开套接字或文件，它们必须在 `stop` 方法中分别关闭。

## 总结

本章中，我们讨论了两种类型的 Flume Source（Pollable 和 Event-driven）。我们还了解了 Flume 自带的各种 Source，以及它们的配置和它们的插件等。我们还研究了一些关于 Source 的最佳实践，主要是关于批处理。最后，我们通过编写自定义 Event-driven 和 Pollable Source，对如何写 Source 有了更好的理解。

在下一章，我们将讨论 Channel 的基本概念、Flume 自带的两类 Channel，以及如何配置它们。

## 参考文献

78

- [netty] The Netty Project, <http://netty.io>
- [nio] Java New I/O API, <http://bit.ly/1wxlcQC>
- [zlib\_ch3] zlib compression library, <http://www.zlib.net>
- [keystore] Java KeyStore API, <http://bit.ly/1wxlgQg>
- [keystore-type] Java KeyStore types, <http://bit.ly/1wxll6y>
- [Java-security] Customizing Java security, <http://bit.ly/1wxlmrj>
- [thrift\_ch3] Apache Thrift project, <http://thrift.apache.org>
- [jms\_src\_code] Apache Flume JMS Source, <http://bit.ly/1wxlreC>
- [line-deserializer] Line deserializer, <http://bit.ly/1wxlyXC>
- [schema-fp] Avro Schema Fingerprints, <http://bit.ly/1wxlF5s>
- [blob] Binary large object, [http://en.wikipedia.org/wiki/Binary\\_large\\_object](http://en.wikipedia.org/wiki/Binary_large_object)
- [avro-idl] Avro IDL documentation, <http://avro.apache.org/docs/current/idl.html>