将覆盖压缩编码默认的扩展名)。

如果需要，HDFS 可以使用 Kerberos[Kerberos] 来确保 HDFS 的安全。HDFS Sink 可以使用在配置文件中指定的认证信息写入到安全 HDFS 集群上。hdfs.kerberosPrincipal 参数指明了登录到 KDC 使用的主体。hdfs.kerberosKeytab 参数应该指明到达 Kerberos keytab 文件的完整路径。该文件对于运行 Flume Agent 的用户应该是可读的，而且必须包含使用的委托人对应的 keytab。HDFS 允许用户模拟其他的用户。Flume 通过 hdfs.proxyUser 参数支持该特征。为了模拟其他用户，指定用户的用户名为这个参数的值即可。Flume 将以那个用户写数据到 HDFS。详细的配置见 HDFS 文档 [impersonation]。为了使用这个功能，运行 Flume Agent 的用户必须在 HDFS 配置中被授权给可以模拟 Flume 写入的用户。

Kerberos、HDFS 和 HBase Sink

相同 Agent 中的所有 HDFS Sink 必须使用相同的 Kerberos 认证信息去登录。另外，所有的 HBase Sink 也必须使用相同的认证信息。对不同的 HDFS 和 HBase Sink 使用不同的认证信息，会导致它们中的一个或多个不能写入到 HDFS 或 HBase。如果多个 HDFS 或 HBase 集群需要相同的数据但是有不同的认证信息，那么数据必须从起源的 Flume Agent（接收或生成数据的 Agent）路由到不同的 Flume Agent，每次写入使用多个认证信息中的一个。这可以通过以下方法来简单实现，将接收数据的 Source 绑定多个 Channel，由 Avro Sink 每个 Channel 拉取数据，并且将它们发送到不同的写 HDFS 的 Flume Agent。

在配置的时间周期之后，HDFS Sink 可以对每个 HDFS 操作超时。这能确保 Sink 不会停止处理事件以防 DataNode 挂起。hdfs.callTimeout 参数值就是配置好的毫秒值的超时时间。该参数的最佳值取决于用户具体的部署，需要谨慎设置以避免太多的超时发生或 Sink 等待时间过长，这可能会影响吞吐量。这些操作使用独立的线程池执行，线程池的大小可以使用 hdfs.threadsPoolSize 参数进行设置，但是该值很少需要改变。

为了触发滚动和闲置超时，Flume 使用单独的线程池，它的大小也可以配置，虽然很少需要改变。这可以使用 hdfs.rollTimerPoolSize 参数调整。

HDFS 客户端 API 对于打开的文件保持了内部缓冲区，尤其对于压缩的文件。为了限制打开文件的数量，从而节约使用的资源，一旦打开文件的数量达到了上限，Sink 就会自动关闭最早写入的文件。这个上限通过 hdfs.maxOpenFiles 参数指定。

hdfs.useLocalTimestamp 参数如果设置为 true，会使用用于基于时间分桶的托管 Agent 的机器上的本地时间。