```
agent1.sinks = sink1 sink2 sink3 sink4
agent1.sinkgroups = sg1 sg2
agent1.channels = channel1 channel2
```

上面的配置片段表示名为 agent1 的 Flume Agent，带有两个 Source、两个 Sink 组、两个 Channel、四个 Sink。即使某些部件罗列出配置参数，如果它们不是在活跃列表中，则它们不创建、配置或启动。其他组件，例如拦截器和 Channel 选择器，不需要存在于活跃列表中。当和它们有关的组件（Source、Sink 等）是活跃的，它们会自动创建并激活。

对于要配置的每个组件，组件的配置使用下面格式的前缀传递：

```
<agent-name>.<component-type>.<component-name>.<configuration-parameter> = \
<value>
```

用于 Source 的 <component-type> 部分是 sources，Sink 是 sinks，Channel 是 channels，Sink 组是 sinkgroups。如拦截器、Channel 选择器和 Sink 处理器等组件是绑定到单个顶级组件，并使用相同的配置模式锚定到这些组件的。

组件名称根据其组件类型设定命名空间。因此，有可能多个组件具有相同名称，只要它们的组件类型不同即可。像拦截器这样的组件也设定独立的 Source 命名空间，多个拦截器具有相同名称是可能的。但是不推荐这种方式。

例如，可以通过下面的格式传递配置到 source1：

```
agent1.sources.source1.port = 4144
agent1.sources.source1.bind = avro.domain.com
```

对于每个组件，配置参数键的前缀被移除（包括组件的名称）。只有实际的参数和它的值通过传递到配置方法的一个 Context 类的实例来传入。Context 是类似 Map 的键值存储，并带有一些稍微复杂的方法。因此，在这种情况下，Source 只在 Context 实例中获取了两个参数，键是 port 和 bind，值是 4144 和 avro.domain.com（并不是整个配置行）。当我们讨论每个组件的配置时，这些表将只显示传递到组件的实际参数，而不是配置文件中的所有行。

15 对于 Source、Sink、Channel 和拦截器，Flume 配置使用 type 参数实例化组件。该 type 参数可以是完整类别名称完全限定类名（FQCN）或内置组件的别名。指定 type 参数的示例如下：

```
agent1.sources.source1.type = avro
```

Flume 配置系统还通过创建 Channel 处理器和为每个 Source 的处理器设定正确的 Channel，确保为每个 Source 设置正确的 Channel。它还负责拦截器的初始化，所以来自