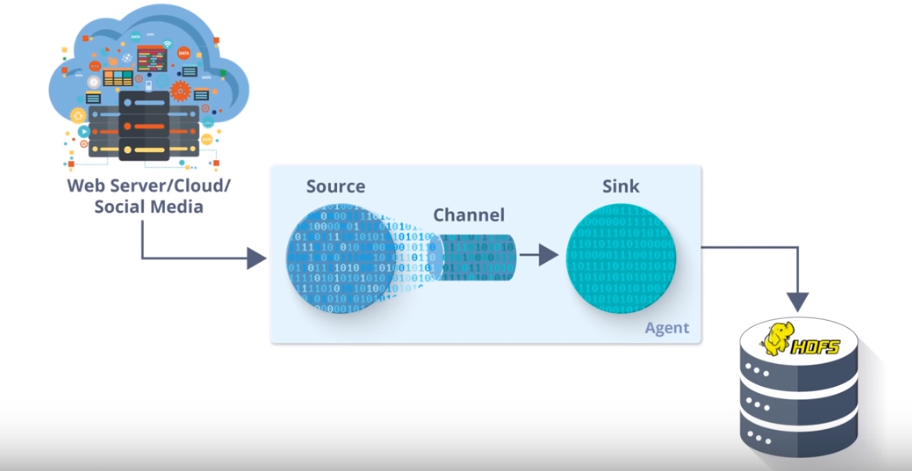
### Apache Flume – Hands On

**Arquitetura:**

****

**Hello World (hello.conf):**

|  |
| --- |
| # Exemplo de Arquivo de Configuração do Flume  # Nome dos componentes do agente  agente1.sources = fonte1  agente1.sinks = destino1  agente1.channels = canal1  # Configuração da fonte  agente1.sources.fonte1.type = netcat  agente1.sources.fonte1.bind = localhost  agente1.sources.fonte1.port = 44444  # Configuração do destino  agente1.sinks.destino1.type = logger  # Canal configurado para utilizar memória  agente1.channels.canal1.type = memory  agente1.channels.canal1.capacity = 1000  agente1.channels.canal1.transactionCapacity = 100  # Associação entre a origem e o destino no canal.  agente1.sources.fonte1.channels = canal1  agente1.sinks.destino1.channel = canal1 |

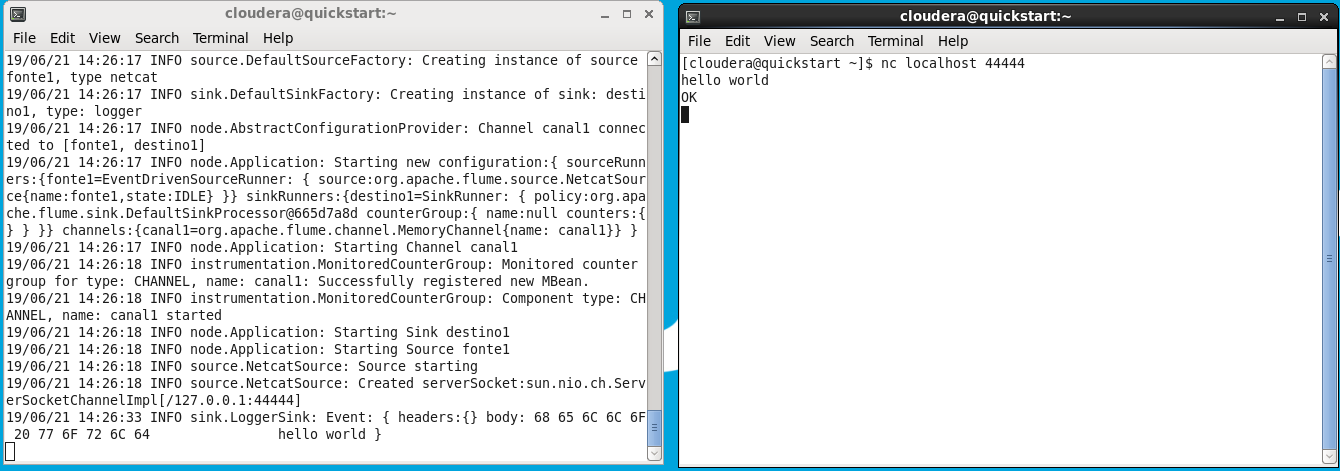
Inicialize o Agente:

|  |
| --- |
| flume-ng agent --conf-file hello.conf --name agente1 -Dflume.root.logger=INFO,console |

Envie um evento via porta 44444:

|  |
| --- |
| nc localhost 44444 <enter>  Hello World <enter>  OK |

Observe que o flume conduzirá o evento até o seu destino (sink):

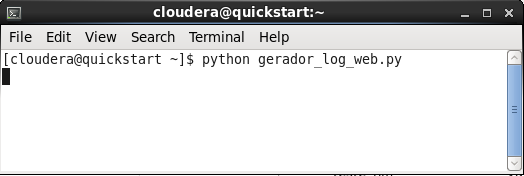


### Streaming de Dados Serializados

**Passo 1:** Simule um streaming de dados web através de um código em python.

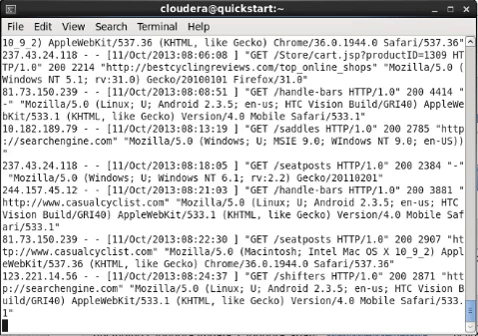
1. Copie o arquivo gerador\_log\_web.py para /home/cloudera
2. Abra o terminal e execute o código em python para gerar o streaming de dados:

|  |
| --- |
| python gerador\_log\_web.py |



1. Em um novo terminal (Ctrl+Shift+N), verifique se o streaming de dados está sendo gerado:

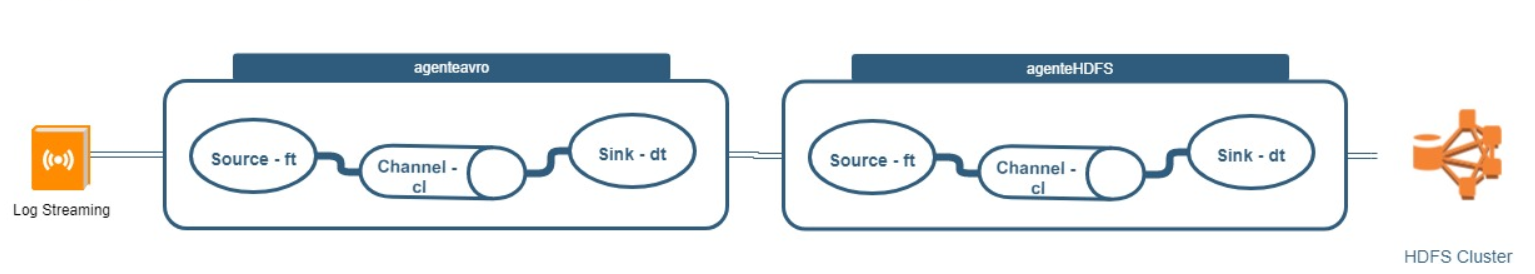
|  |
| --- |
| tail -F /tmp/access\_log |



1. Minimize os terminais.

**Passo 2:** Crie dois agentes, o primeiro deverá buscar os dados de streaming, e via memória, transformá-los em Avro (serialização). O segundo receberá o Avro e fará a ingestão no HDFS.

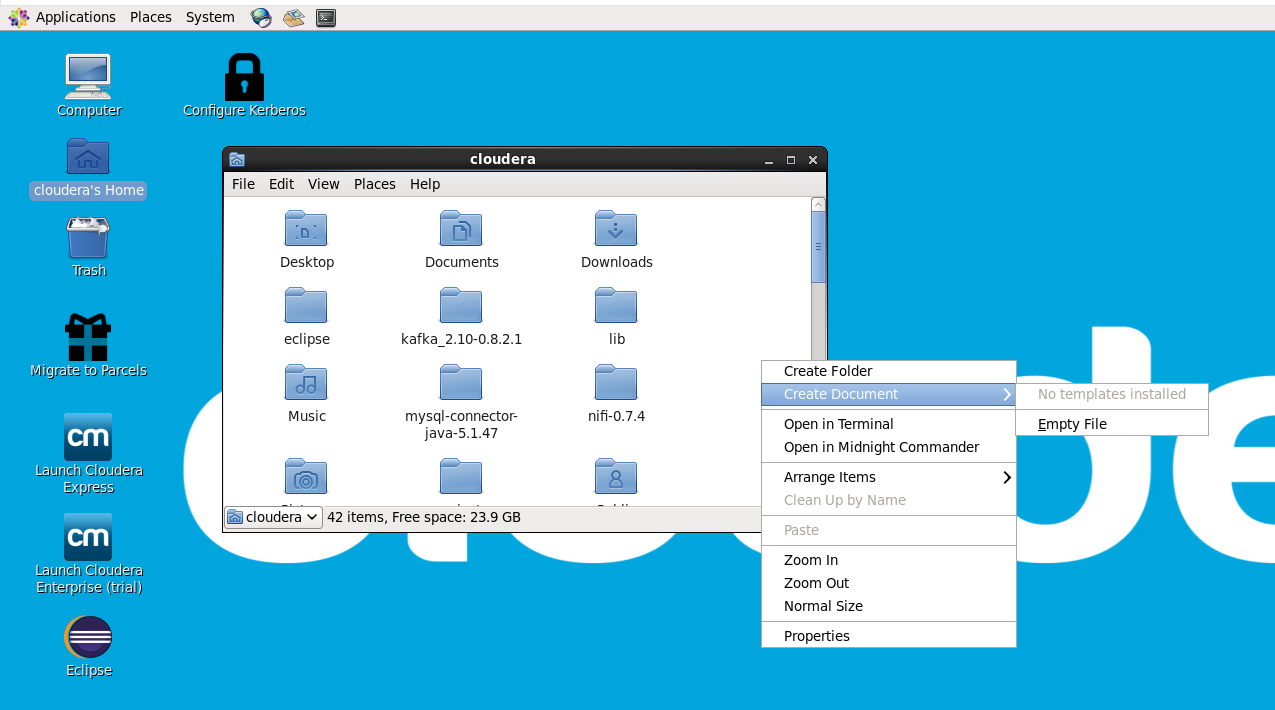
* Arquitetura da Ingestão



1. Crie no HDFS um diretório chamado flume:

|  |
| --- |
| sudo -u hdfs hadoop fs -mkdir /user/flume |

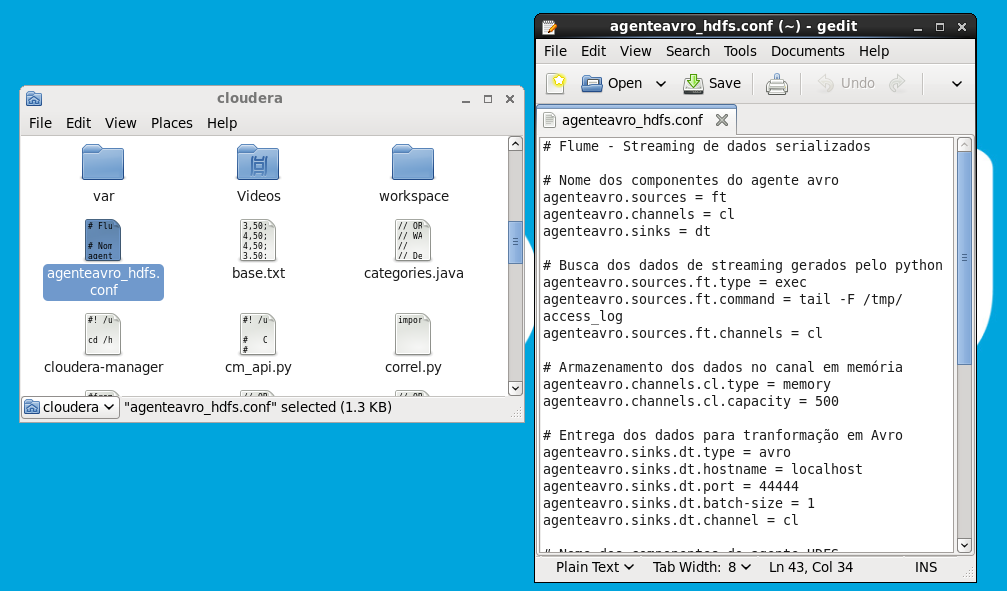
1. Crie um novo arquivo de configuração em /home/cloudera chamado agenteavro\_hdfs.conf:



Ao dar nomes aos arquivos, observe letras maiúsculas e minúsculas, pois o Flume é sensível a essas diferenças.

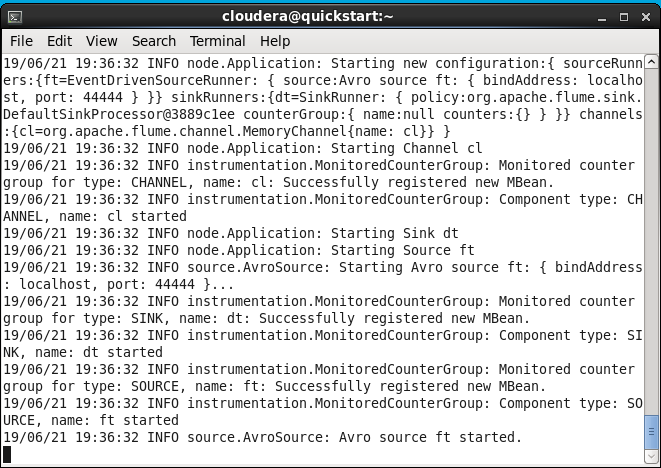
1. Insira nele, o seguinte código:

|  |
| --- |
| # Flume - Streaming de dados serializados  # Nome dos componentes do agente avro  agenteavro.sources = ft  agenteavro.channels = cl  agenteavro.sinks = dt  # Busca dos dados de streaming gerados pelo python  agenteavro.sources.ft.type = exec  agenteavro.sources.ft.command = tail -F /tmp/access\_log  agenteavro.sources.ft.channels = cl  # Armazenamento dos dados no canal em memória  agenteavro.channels.cl.type = memory  agenteavro.channels.cl.capacity = 500  # Entrega dos dados para tranformação em Avro  agenteavro.sinks.dt.type = avro  agenteavro.sinks.dt.hostname = localhost  agenteavro.sinks.dt.port = 44444  agenteavro.sinks.dt.batch-size = 1  agenteavro.sinks.dt.channel = cl  # Nome dos componentes do agente HDFS  agentehdfs.sources = ft  agentehdfs.channels = cl  agentehdfs.sinks = dt  # Recepção dos dados em Avro  agentehdfs.sources.ft.type = avro  agentehdfs.sources.ft.bind = localhost  agentehdfs.sources.ft.port = 44444  agentehdfs.sources.ft.channels = cl  # Armazenamento dos dados no canal em memória  agentehdfs.channels.cl.type = memory  agentehdfs.channels.cl.capacity = 500  # Ingestão dos dados no HDFS  agentehdfs.sinks.dt.type = hdfs  agentehdfs.sinks.dt.hdfs.path = /user/flume  agentehdfs.sinks.dt.hdfs.filePrefix = log\_web  agentehdfs.sinks.dt.channel = cl |



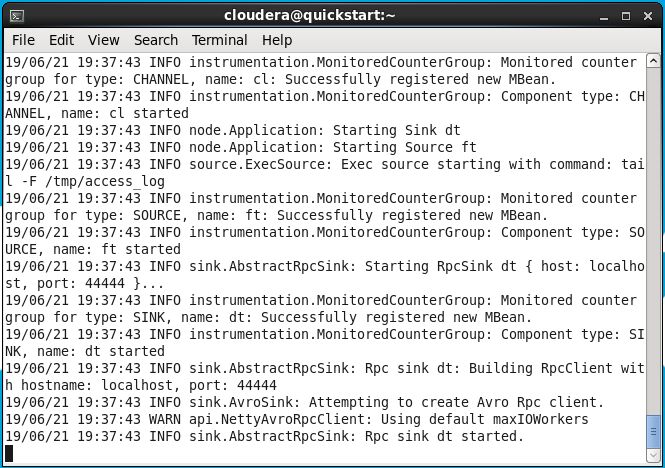
1. Abra um novo terminal para a execução do “agentehdfs” e insira:

|  |
| --- |
| sudo flume-ng agent -n agentehdfs -f agenteavro\_hdfs.conf |



1. Mantenha o terminal do “agentehdfs” aberto e abra um novo terminal para a execução do “agenteavro”. Insira:

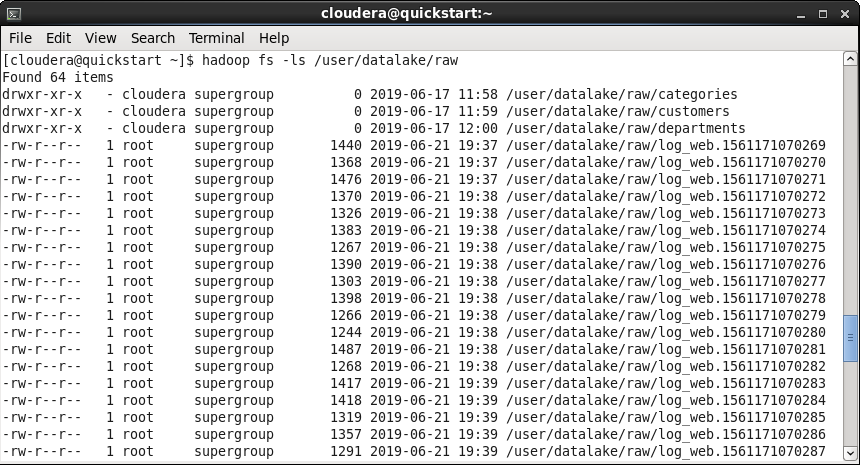
|  |
| --- |
| sudo flume-ng agent -n agenteavro -f agenteavro\_hdfs.conf |



1. Nesse instante, as ingestões já começaram a ocorrer. Para certificar-se, liste o conteúdo do lake na raw zone. Abra um novo terminal:

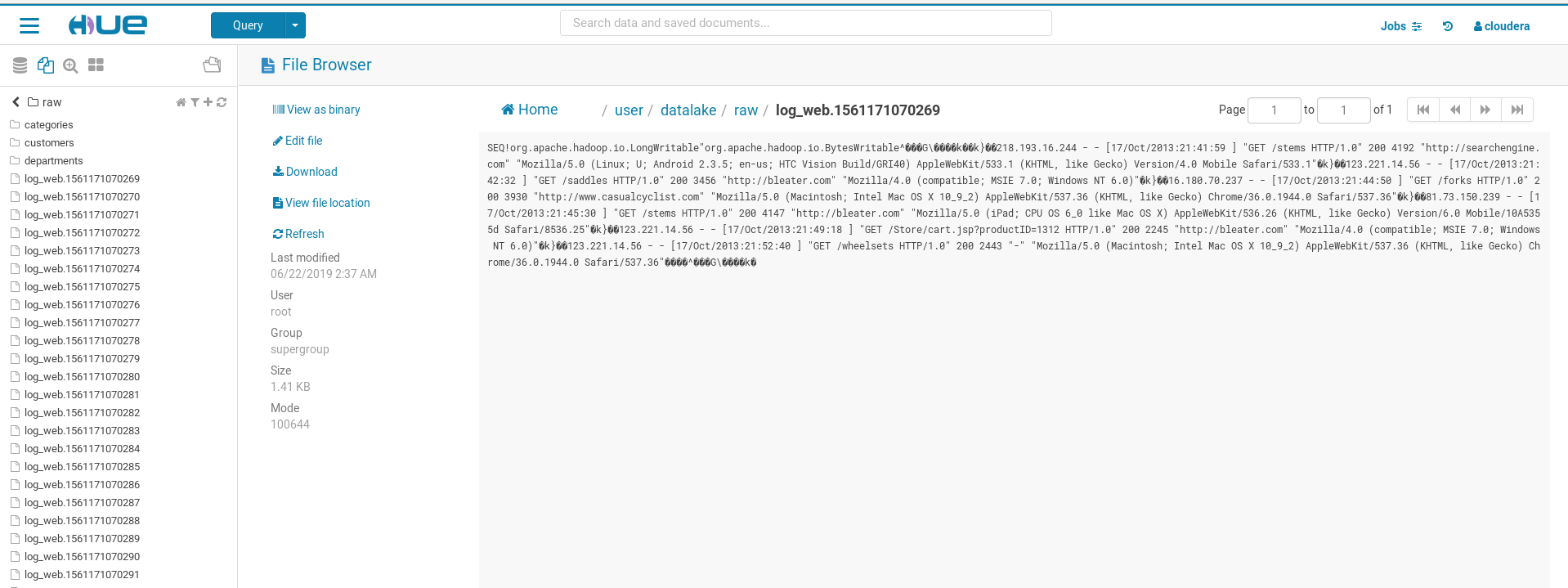
|  |
| --- |
| hadoop fs -ls /user/flume |

.



.

Se preferir você pode também visualizar o mesmo resultado no Hue. Basta acessá-lo e procurar pelo diretório Flume que lá estarão os arquivos:



Enquanto os agentes estiverem no ar, a ingestão ocorrerá initerruptamente. Vamos parar os agentes. No terminal de log de cada um deles tecle Ctrl+C, pare primeiro o “agenteavro” e depois o “agentehdfs”. Pare também a execução do streaming via python.

Uau, fizemos juntos uma ingestão de streaming de dados serializados! Toda essa lógica pode ser aplicada para qualquer necessidade que envolva fluxo contínuo.

*Insira o print do Hue em “3.2. Ingestão e Transformação dos Dados” do documento “Especificação da Solução de DW”.*

### Apache Flume – Hand On – Desafio

Que tal interferir na ingestão que acabamos de fazer, acrescentando um filtro. Vamos especificar exatamente o rastro de qual IP desejamos ver.

Acrescente no “agenteavro”, um *interceptor* que filtre apenas os logs do ip 16.180.70.237 (as novas linhas estão em negrito):

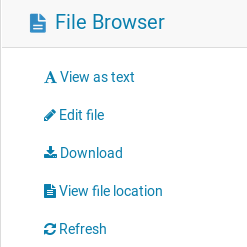
|  |
| --- |
| agenteavro.sources.ft.command = tail -F /tmp/access\_log  **agenteavro.sources.ft.interceptors=e1**  **agenteavro.sources.ft.interceptors.e1.type=regex\_filter**  **agenteavro.sources.ft.interceptors.e1.regex=16\.180\.70\.237**  agenteavro.sources.ft.channels = cl |

Dispare novamente o código em python, assim como os dois agentes (agentehdfs e agenteavro) e verifique no Raw Zone se o log está sendo filtrado.

Para ajudar a localizar os arquivos filtrados, sugiro que altere o prefixo de seus nomes:

|  |
| --- |
| agentehdfs.sinks.dt.hdfs.filePrefix = filter\_log\_web |

No Hue, ao clicar sobre um arquivo filtrado, cheque seu conteúdo através da opção “View as text”. Verifique se somente o IP 16.180.70.237 está sendo ingerido.



*Insira o print do Hue em “3.2 Ingestão e Transformação dos Dados” do documento “Especificação da Solução de DW”.*