



Hands on Lab

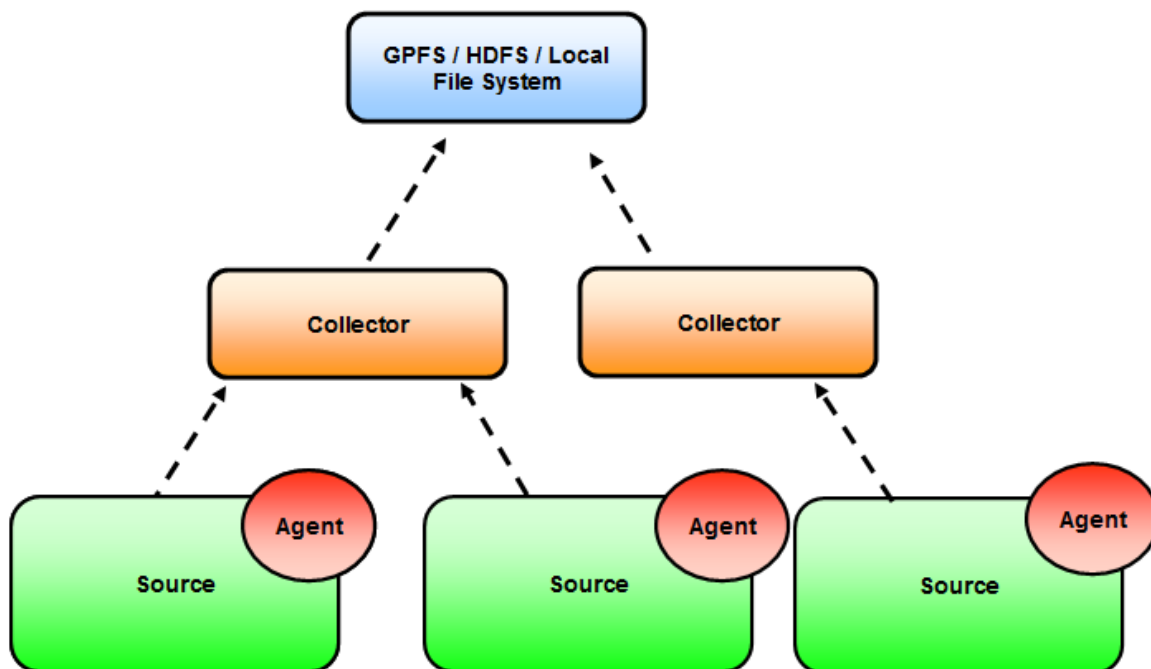
**IBM InfoSphere BigInsights V2.1
Hadoop Basics**

Unit 5: Data Movement

1 Introduction

Flume is a distributed service for efficiently moving the large amounts of data. The primary use case of Flume is to gather a set of log files on every machine in a cluster and aggregate them to a centralized persistent store such as Hadoop Distributed File System (HDFS).

The diagram below shows a deployment of Flume that collects log data from a set of application servers. The flow demonstrates it in three tiers. The first tier is the **agent** tier. Agent nodes are typically installed on the machines that generate the logs and are the initial point of contact with Flume. The second tier is the **collector** tier which is typically installed where the Flume master is. The third tier is the final destination of the file which can be either GPFS, HDFS or the local file system.



1.1 About this Lab

After completing this hands-on lab, you'll be able to:

- Move data from your local file system to HDFS using Flume
- Setup and configure Flume agents
- Start the agents to collect data.



Which services should be started?

This lab assumes that *all* InfoSphere BigInsights services have already been started.



NOTE: These instructions are based on the BigInsights QuickStart Edition 2.1, so your screenshots could vary depending on the distribution which you are using.

1.2 Getting Started

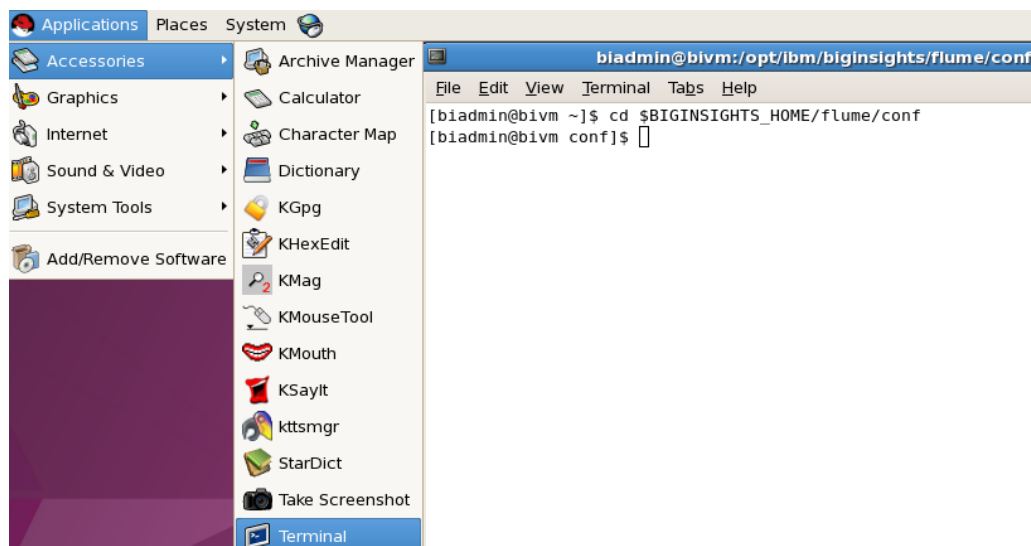
For this lab all Hadoop components should be up and running. If all components are running you may move on to Section 2 of this lab. Otherwise please refer back to Hadoop Basics Unit 1: Exploring Hadoop Distributed File System to get started. (All Hadoop components should be started.)

2 Setting up Flume configurations

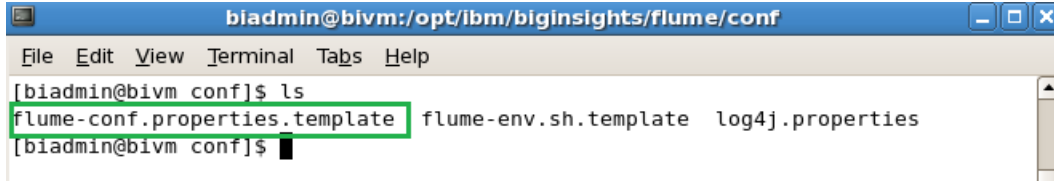
Before we can start to collect data, we must first configure our Flume agents

__1. Open a terminal and navigate to `$BIGINSIGHTS_HOME/flume/conf`

`cd $BIGINSIGHTS_HOME/flume/conf`



__2. Execute an ls command to list all the files in this directory



```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
[biadmin@bivm conf]$ ls
flume-conf.properties.template  flume-env.sh.template  log4j.properties
[biadmin@bivm conf]$
```

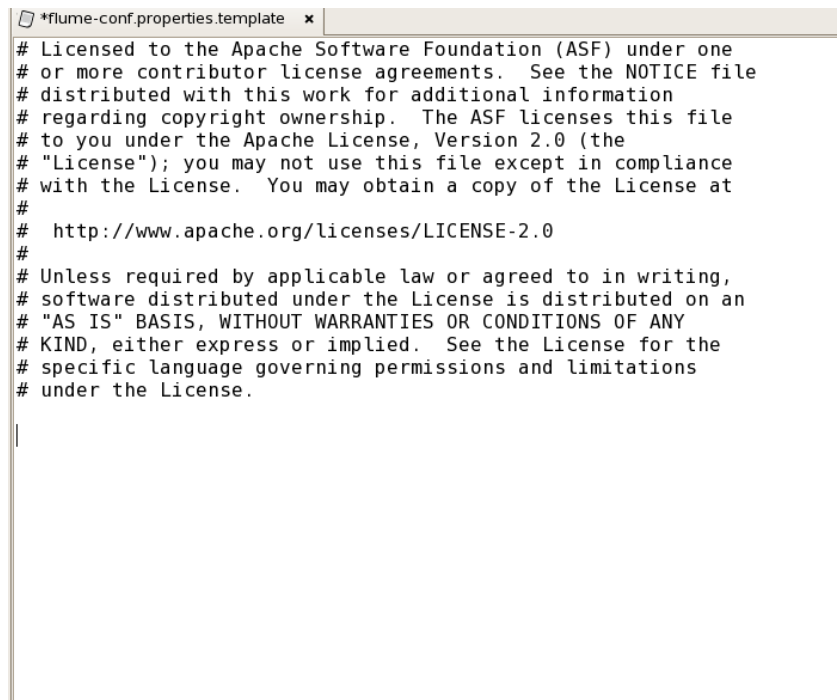
The file we will edit is called `flume-conf.properties.template`. This is the file in which you configure the Flume agents. You can add configurations for more than one agent as will be shown in later steps.

__3. To edit this file enter the following:

```
gedit flume-conf.properties.template
```

You do not need to use *gedit*, you may use *vi* or any editor you wish.

__4. Erase all the code except for what is shown in the image below



```
*flume-conf.properties.template x
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
```

We will be adding code snippets to this file and explaining them as we go

__5. Now we want to configure our source agent. Add the following code:

```
# Name the components on this agent
agent1.sources = src
agent1.sinks = snk
agent1.channels = ch
```

This snippet names the components of agent1. Each agent needs a *source*, *sinks*, and *channels* component. You can name each component anything you wish; in this case we used 'src' for sources, 'snk' for sinks and 'ch' for channels.

__6. Now we must describe and configure the source. Add the following code:

```
# Describe/configure the source
agent1.sources.src.type = netcat
agent1.sources.src.bind = localhost
agent1.sources.src.port = 44444
```

Notice the syntax:

<Agent>.sources.<SourceName>.<Property> = value.
For the first line we specify the <Property> as type and the value as netcat. Each source must have the 'type' property. On the next line we have bind with value localhost, this the hostname or ip address to listen on. The last line is the port number to bind to.

__7. Now we must describe and configure the sink. Add the following code:

```
# Describe/configure the sink
agent1.sinks.snk.type = hdfs
agent1.sinks.snk.writeFormat = Text
agent1.sinks.snk.hdfs.path = hdfs://bivm:9000/tmp/
```

Notice how the syntax is similar to that of source. In this snippet we configure the sink. In the first line we specify the <Property> as 'type' and give it the value *hdfs*. This means that it will write the file to HDFS, when ever you specify *type* as *hdfs* you must specify the <Property> *hdfs.path* as well. There is also a *writeFormat* <Property> that specifies the format of the output as text.

__8. Now we must configure the channel. Add the following code:

```
# Use a channel which buffers events in memory
agent1.channels.ch.type = memory
agent1.channels.ch.capacity = 1000
```

These two lines configure the channel component of the agent. The <Property> *type* has value *memory* which means that events will be buffered to memory. The <Property> *capacity* is the maximum number of events stored on the channel.

- ___9. Now we must bind the source and sink to the channel. Add the following code:

```
# Bind the source and sink to the channel
agent1.sources.src.channels = ch
agent1.sinks.snk.channel = ch
```

These two lines simply bind the source and the sink to the channel.

Your file should now look like the image below.

```
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.

# Name the components on this agent
agent1.sources = src
agent1.sinks = snk
agent1.channels = ch

# Describe/configure the source
agent1.sources.src.type = netcat
agent1.sources.src.bind = localhost
agent1.sources.src.port = 44444

# Describe the sink
agent1.sinks.snk.type = hdfs
agent1.sinks.snk.writeFormat = Text
agent1.sinks.snk.hdfs.path = hdfs://bivm.ibm.com:9000/tmp/

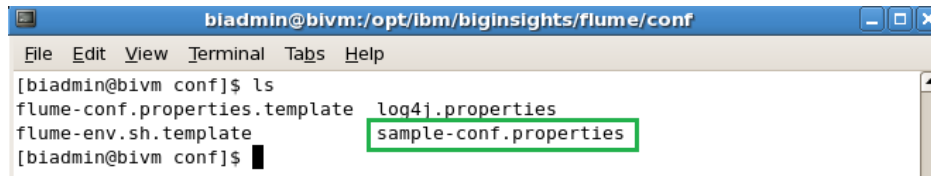
# Use a channel which buffers events in memory
agent1.channels.ch.type = memory
agent1.channels.ch.capacity = 1000

# Bind the source and sink to the channel
agent1.sources.src.channels = ch
agent1.sinks.snk.channel = ch
```

- ___10. We do not want to overwrite the previous file. Save As sample-conf.properties. and close the window.

- __11. Clear the terminal and execute the ls command to make sure your file is in the directory.

```
clear
ls
```



```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
[biadmin@bivm conf]$ ls
flume-conf.properties.template  log4j.properties
flume-env.sh.template           sample-conf.properties
[biadmin@bivm conf]$
```

2.1 Running the flume agent

Now that we have setup the flume agent configuration file we can now run the agent.

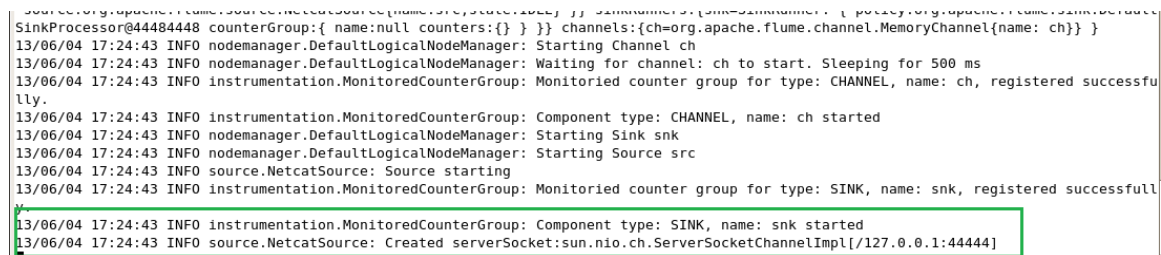
- __1. To run this agent execute the following:

```
$BIGINSIGHTS_HOME/flume/bin/flume-ng agent -f sample-
conf.properties -n agent1 -Dflume.root.logger=INFO,console
```



```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
[biadmin@bivm conf]$ $BIGINSIGHTS_HOME/flume/bin/flume-ng agent -f sample-conf.properties -n agent1 -Dflume.root.logger=INFO,console
```

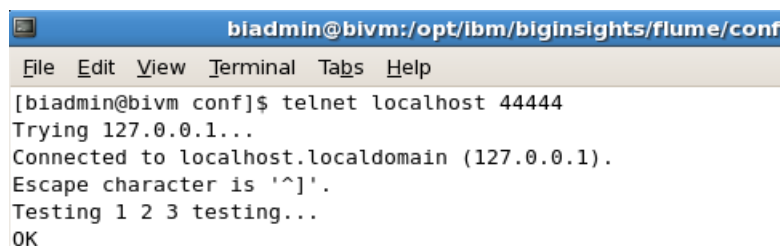
If executed correctly your terminal will look similar to the image below



```
SinkProcessor@44484448 counterGroup:{ name:null counters:{} } } channels:{ch=org.apache.flume.channel.MemoryChannel{name: ch}} }
13/06/04 17:24:43 INFO nodemanager.DefaultLogicalNodeManager: Starting Channel ch
13/06/04 17:24:43 INFO nodemanager.DefaultLogicalNodeManager: Waiting for channel: ch to start. Sleeping for 500 ms
13/06/04 17:24:43 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: CHANNEL, name: ch, registered successfu
lly.
13/06/04 17:24:43 INFO instrumentation.MonitoredCounterGroup: Component type: CHANNEL, name: ch started
13/06/04 17:24:43 INFO nodemanager.DefaultLogicalNodeManager: Starting Sink snk
13/06/04 17:24:43 INFO nodemanager.DefaultLogicalNodeManager: Starting Source src
13/06/04 17:24:43 INFO source.NetcatSource: Source starting
13/06/04 17:24:43 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SINK, name: snk, registered successfull
y.
13/06/04 17:24:43 INFO instrumentation.MonitoredCounterGroup: Component type: SINK, name: snk started
13/06/04 17:24:43 INFO source.NetcatSource: Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[/127.0.0.1:44444]
```

- __2. The agent is now listening on port 44444 on your localhost. To see that the agent is listening open a new terminal and do the following:

```
telnet localhost 44444
type anything you wish
```



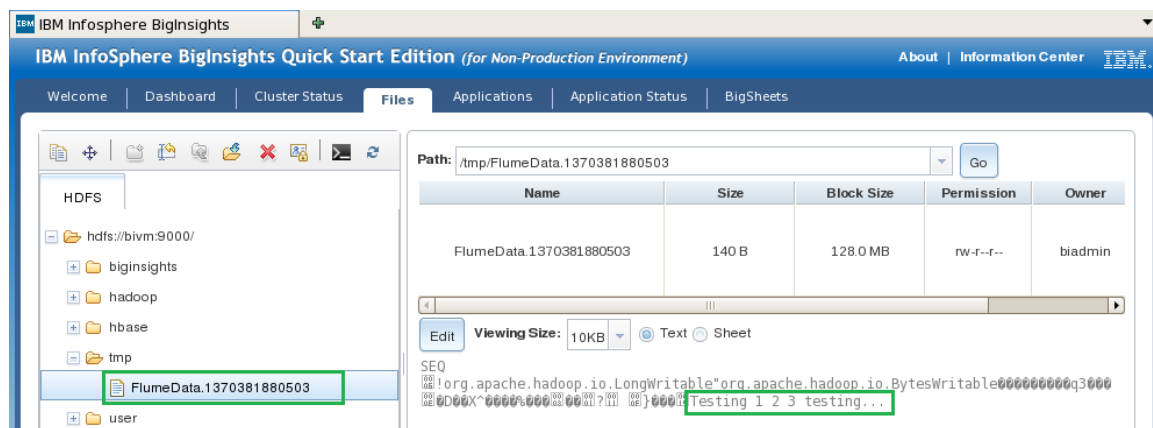
```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
[biadmin@bivm conf]$ telnet localhost 44444
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
Testing 1 2 3 testing...
OK
```

You may exit telnet by pressing **Ctrl+] then <Enter>** then type **quit**. Do not close the terminal window as you will need it in the next few steps.

The terminal in which your Flume agent was running should be updated and look similar to the image below.

```
13/06/04 17:37:37 INFO source.NetcatSource: Source starting
13/06/04 17:37:37 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SINK, name: snk, registered successfully.
13/06/04 17:37:37 INFO instrumentation.MonitoredCounterGroup: Component type: SINK, name: snk started
13/06/04 17:37:37 INFO source.NetcatSource: Created serverSocket:sun.nio.ch.ServerSocketChannelImpl[127.0.0.1:44444]
13/06/04 17:38:00 INFO hdfs.BucketWriter: Creating hdfs://bivm:9000/tmp/FlumeData.1370381880503.tmp
```

You can now open the Web Console and navigate to the tmp folder and find that the output is inside.



__3. To end this flume agent process you must manually kill it. On the terminal where you used telnet, issue the following command:

```
ps -ef | grep flume | awk '{print $2}'
```

This command lists all the flume processes.

__4. Now manually kill all the flume processes by issuing the following command:

```
kill -9 <pid> <pid>..
```

```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
[biadmin@bivm conf]$ ps -ef | grep flume | awk '{print $2}'
24161
28836
[biadmin@bivm conf]$ kill -9 24161 28836
bash: kill: (24161) - No such process
[biadmin@bivm conf]$
```


All flume processes should now be killed, if you look back to the terminal in which your flume agent was running you will see it now says *'Killed'*.

```
13/06/04 17:38:00 INFO hdfs.BucketWriter: Creating hdfs://bivm:9000/tmp//FlumeData.1370381880503.tmp
13/06/04 17:38:31 INFO hdfs.BucketWriter: Renaming hdfs://bivm:9000/tmp/FlumeData.1370381880503.tmp to
70381880503
Killed
[biadmin@bivm conf]$
```

There are no longer any flume agents listening on port 44444.

3 Collecting Data with Flume

In the previous version of Flume (0.9.4) there was the role of collector and agent for a Flume node, and there was also Flume master that played the role of centralized control and management.

In the current version of Flume (1.3.0) agent becomes the only role of Flume. There is no flume master or web page anymore. However, agent could be configured as the role of collector.

In the next few steps we will show you how to configure an agent and collector.

3.1 Configuring the agent

__1. Navigate to \$BIGINSIGHTS_HOME/flume/conf

```
cd $BIGINSIGHTS_HOME/flume/conf
```

__2. Open the file sample-conf.properties for editing.

```
gedit sample-conf.properties
```

__3. Replace the code in the file with the code below:

```
#####
# AGENT 1 "AGENT" #
#####

# Name the components on this agent
agent1.sources = avro-src
agent1.sinks = avro-sink
agent1.channels = ch

# Describe/configure the source
agent1.sources.avro-src.type = avro
agent1.sources.avro-src.bind = 127.0.0.1
agent1.sources.avro-src.port = 44444
```

```

# Describe/configure the sink
agent1.sinks.avro-sink.type = avro
agent1.sinks.avro-sink.hostname = localhost
agent1.sinks.avro-sink.port = 55555

# Use a channel which buffers events in memory
agent1.channels.ch.type = memory
agent1.channels.ch.capacity = 1000

# Bind the source and sink to the channel
agent1.sources.avro-src.channels = ch
agent1.sinks.avro-sink.channel = ch

#####
# AGENT 2 "COLLECTOR" #
#####

# Name the components on this agent
agent2.sources = avro-collection-source
agent2.sinks = hdfs-sink
agent2.channels = mem-channel

# Describe/configure the source
agent2.sources.avro-collection-source.type = avro
agent2.sources.avro-collection-source.bind = 127.0.0.1
agent2.sources.avro-collection-source.port = 55555

# Describe/configure the sink
agent2.sinks.hdfs-sink.type = hdfs
agent2.sinks.hdfs-sink.writeFormat = Text
agent2.sinks.hdfs-sink.hdfs.path = hdfs://bivm:9000/tmp/

# Use a channel which buffers events in memory
agent2.channels.mem-channel.type = memory
agent2.channels.mem-channel.capacity = 1000

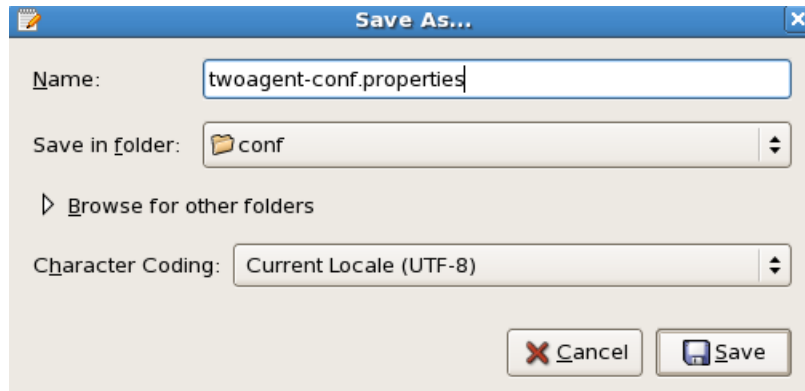
# Bind the source and sink to the channel
agent2.sources.avro-collection-source.channels = mem-channel
agent2.sinks.hdfs-sink.channel = mem-channel

```

In the code above agent1 will be listening on port 44444 for incoming avro events. It will then forward those avro events into the avro sink. The avro sink will then send the events to any agent listening on port 55555.

In the code above agent2 will be listening on port 55555 for incoming avro events. With the configurations that we have, agent1 will be sending avro events on port 55555, which agent2 will receive. Agent2 will then write the avro events to HDFS.

__4. Save this new file as twoagent-conf.properties.

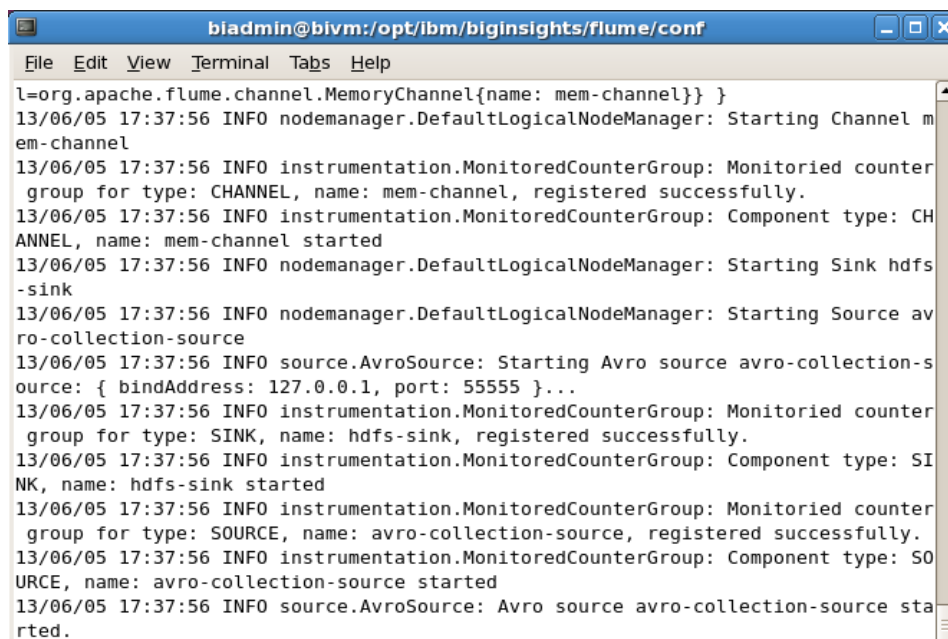


3.2 Running Two Agents

__1. Open a terminal and execute the following:

```
cd $BIGINSIGHTS_HOME/flume/conf
$BIGINSIGHTS_HOME/flume/bin/flume-ng agent -n agent2 -f
twoagent-conf.properties
```

The terminal should now appear similar to the image below

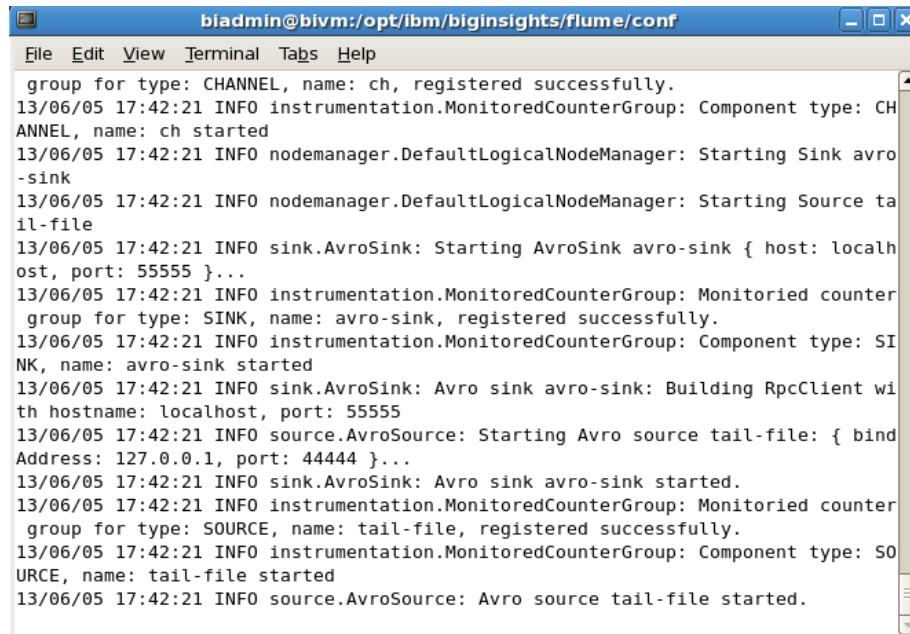


Agent2 or “collector” is now listening on port 55555 for avro events. Do not close this terminal.

__2. Open a new terminal and execute the following

```
cd $BIGINSIGHTS_HOME/flume/conf
$BIGINSIGHTS_HOME/flume/bin/flume-ng agent -n agent1 -f
twoagent-conf.properties
```

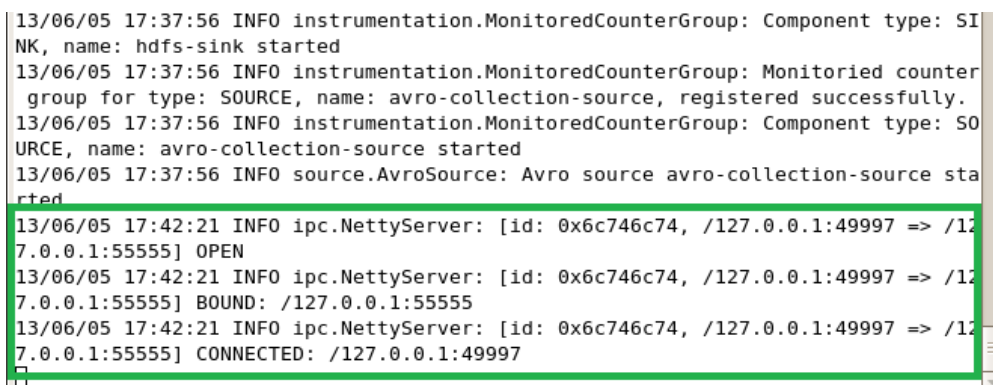
The terminal should now appear similar to the image below



```
bladmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
group for type: CHANNEL, name: ch, registered successfully.
13/06/05 17:42:21 INFO instrumentation.MonitoredCounterGroup: Component type: CH
ANNEL, name: ch started
13/06/05 17:42:21 INFO nodemanager.DefaultLogicalNodeManager: Starting Sink avro
-sink
13/06/05 17:42:21 INFO nodemanager.DefaultLogicalNodeManager: Starting Source ta
il-file
13/06/05 17:42:21 INFO sink.AvroSink: Starting AvroSink avro-sink { host: localh
ost, port: 55555 }...
13/06/05 17:42:21 INFO instrumentation.MonitoredCounterGroup: Monitored counter
group for type: SINK, name: avro-sink, registered successfully.
13/06/05 17:42:21 INFO instrumentation.MonitoredCounterGroup: Component type: SI
NK, name: avro-sink started
13/06/05 17:42:21 INFO sink.AvroSink: Avro sink avro-sink: Building RpcClient wi
th hostname: localhost, port: 55555
13/06/05 17:42:21 INFO source.AvroSource: Starting Avro source tail-file: { bind
Address: 127.0.0.1, port: 44444 }...
13/06/05 17:42:21 INFO sink.AvroSink: Avro sink avro-sink started.
13/06/05 17:42:21 INFO instrumentation.MonitoredCounterGroup: Monitored counter
group for type: SOURCE, name: tail-file, registered successfully.
13/06/05 17:42:21 INFO instrumentation.MonitoredCounterGroup: Component type: SO
URCE, name: tail-file started
13/06/05 17:42:21 INFO source.AvroSource: Avro source tail-file started.
```

Agent1 or “agent” is now listening on port 44444 for avro events. Do not close this window.

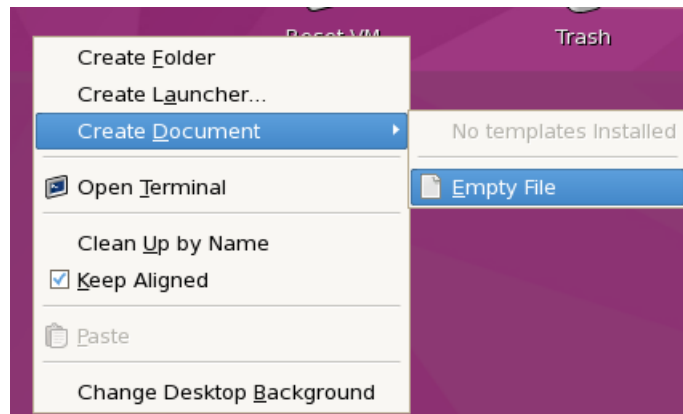
You will notice that the terminal for agent2 has changed, this is because it is now connected to agent1.



```
13/06/05 17:37:56 INFO instrumentation.MonitoredCounterGroup: Component type: SI
NK, name: hdfs-sink started
13/06/05 17:37:56 INFO instrumentation.MonitoredCounterGroup: Monitored counter
group for type: SOURCE, name: avro-collection-source, registered successfully.
13/06/05 17:37:56 INFO instrumentation.MonitoredCounterGroup: Component type: SO
URCE, name: avro-collection-source started
13/06/05 17:37:56 INFO source.AvroSource: Avro source avro-collection-source sta
rted
13/06/05 17:42:21 INFO ipc.NettyServer: [id: 0x6c746c74, /127.0.0.1:49997 => /12
7.0.0.1:55555] OPEN
13/06/05 17:42:21 INFO ipc.NettyServer: [id: 0x6c746c74, /127.0.0.1:49997 => /12
7.0.0.1:55555] BOUND: /127.0.0.1:55555
13/06/05 17:42:21 INFO ipc.NettyServer: [id: 0x6c746c74, /127.0.0.1:49997 => /12
7.0.0.1:55555] CONNECTED: /127.0.0.1:49997
```

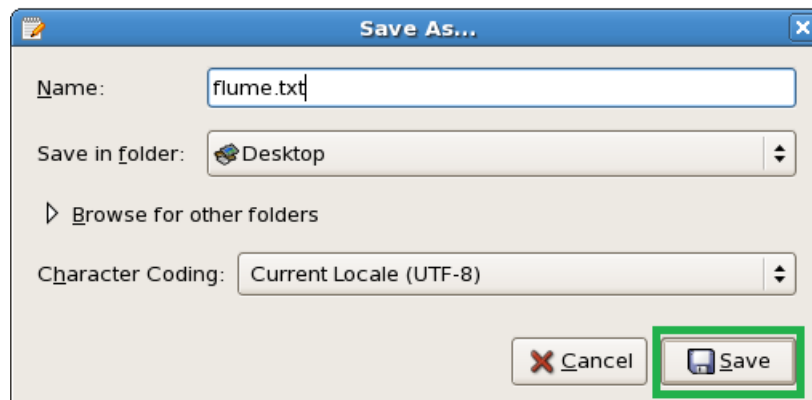
You should now have both agent2 and agent1 running.

__3. Right-click on the desktop and click Create Document -> Empty File



__4. The document will appear on the top-left corner of your desktop. Open this file and write "***Learning Flume is fun!***"

__5. Click File -> Save and save the file as flume.txt



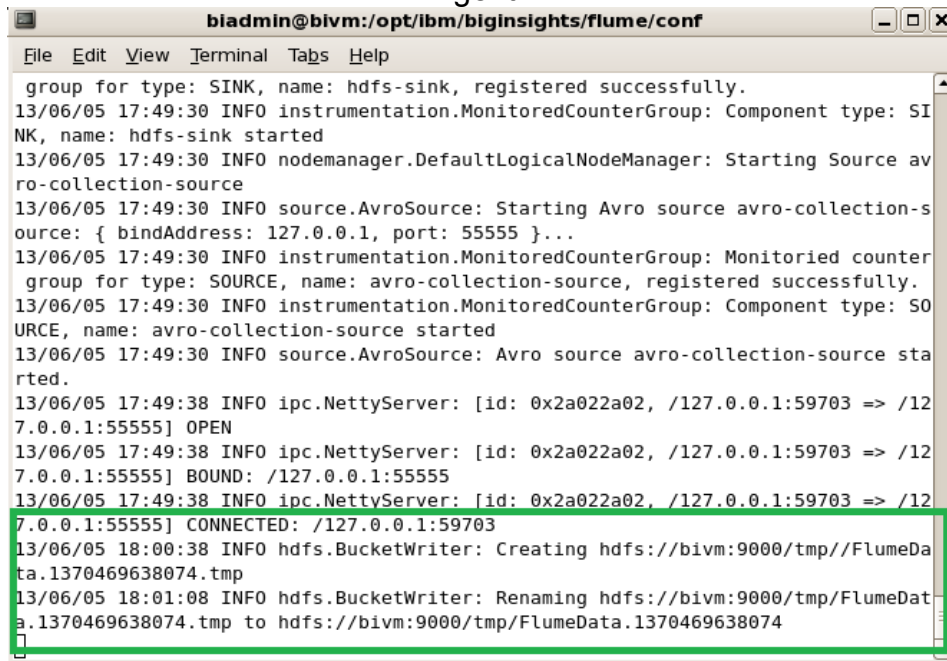
__6. Now that you have both agents running and a sample file execute the following:

```
$BIGINSIGHTS_HOME/flume/bin/flume-ng avro-client -H  
localhost -p 44444 -F /home/biadmin/Desktop/flume.txt
```

This will send an avro event to the local host on port 44444 which is where agent1 is listening on.

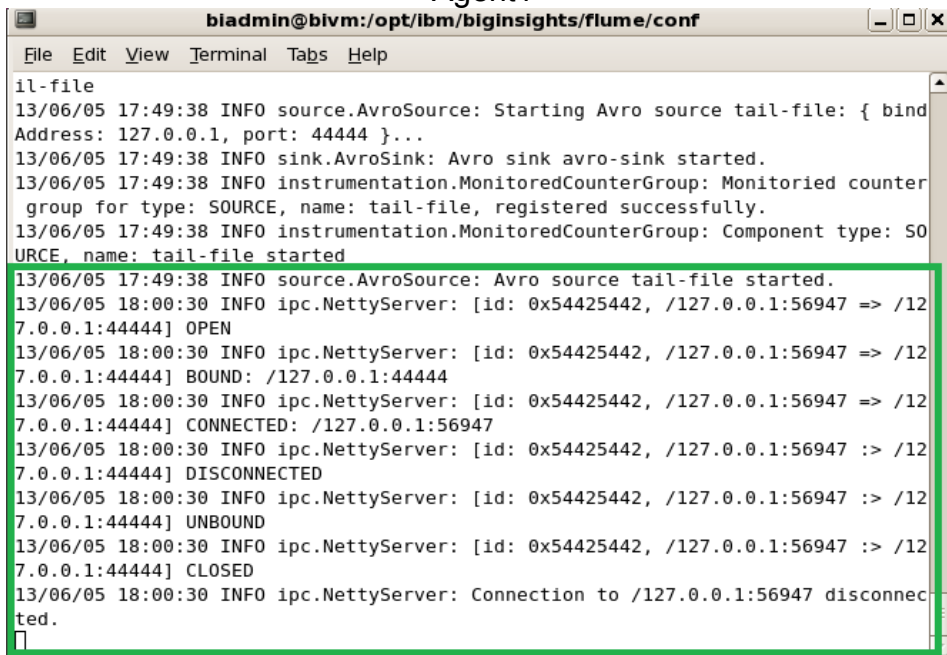
- ___7. Inspect the terminals on which agent1 and 2 were running on. You should see something similar to the images below.

Agent2



```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
group for type: SINK, name: hdfs-sink, registered successfully.
13/06/05 17:49:30 INFO instrumentation.MonitoredCounterGroup: Component type: SINK, name: hdfs-sink started
13/06/05 17:49:30 INFO nodemanager.DefaultLogicalNodeManager: Starting Source avro-collection-source
13/06/05 17:49:30 INFO source.AvroSource: Starting Avro source avro-collection-source: { bindAddress: 127.0.0.1, port: 55555 }...
13/06/05 17:49:30 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SOURCE, name: avro-collection-source, registered successfully.
13/06/05 17:49:30 INFO instrumentation.MonitoredCounterGroup: Component type: SOURCE, name: avro-collection-source started
13/06/05 17:49:30 INFO source.AvroSource: Avro source avro-collection-source started.
13/06/05 17:49:38 INFO ipc.NettyServer: [id: 0x2a022a02, /127.0.0.1:59703 => /127.0.0.1:55555] OPEN
13/06/05 17:49:38 INFO ipc.NettyServer: [id: 0x2a022a02, /127.0.0.1:59703 => /127.0.0.1:55555] BOUND: /127.0.0.1:55555
13/06/05 17:49:38 INFO ipc.NettyServer: [id: 0x2a022a02, /127.0.0.1:59703 => /127.0.0.1:55555] CONNECTED: /127.0.0.1:59703
13/06/05 18:00:38 INFO hdfs.BucketWriter: Creating hdfs://bivm:9000/tmp/FlumeData.1370469638074.tmp
13/06/05 18:01:08 INFO hdfs.BucketWriter: Renaming hdfs://bivm:9000/tmp/FlumeData.1370469638074.tmp to hdfs://bivm:9000/tmp/FlumeData.1370469638074
```

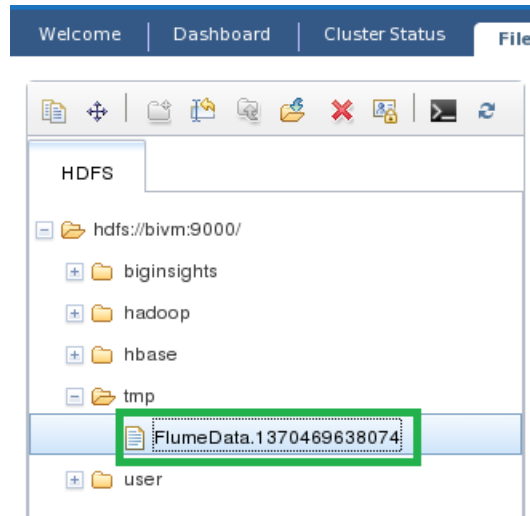
Agent1



```
biadmin@bivm:/opt/ibm/biginsights/flume/conf
File Edit View Terminal Tabs Help
1l-file
13/06/05 17:49:38 INFO source.AvroSource: Starting Avro source tail-file: { bindAddress: 127.0.0.1, port: 44444 }...
13/06/05 17:49:38 INFO sink.AvroSink: Avro sink avro-sink started.
13/06/05 17:49:38 INFO instrumentation.MonitoredCounterGroup: Monitored counter group for type: SOURCE, name: tail-file, registered successfully.
13/06/05 17:49:38 INFO instrumentation.MonitoredCounterGroup: Component type: SOURCE, name: tail-file started
13/06/05 17:49:38 INFO source.AvroSource: Avro source tail-file started.
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] OPEN
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] BOUND: /127.0.0.1:44444
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] CONNECTED: /127.0.0.1:56947
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] DISCONNECTED
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] UNBOUND
13/06/05 18:00:30 INFO ipc.NettyServer: [id: 0x54425442, /127.0.0.1:56947 => /127.0.0.1:44444] CLOSED
13/06/05 18:00:30 INFO ipc.NettyServer: Connection to /127.0.0.1:56947 disconnected.
```

- ___8. After inspecting agent2 you will notice that it says “creating hdfs://bivm:9000/tmp/FlumeData...” This means we have successfully written to HDFS. Open a Web Console and navigate to the files tab.

- ___9. Open the hdfs://bivm:9000 folder then open the tmp folder. You will now see the file that was created by flume agent2



- ___10. You may now manually kill the flume processes.

Congratulations you can now collect data using two different methods in Flume (1.3.0)