**Stream Processing & Analytics – Flume**

**Apache Flume Getting Started**

1. Objective:
2. Get familiarity with the Apache Flume system
3. Get hands-on experience with Flume Agents
4. Configure the data flows involving simple and interceptor based data flows

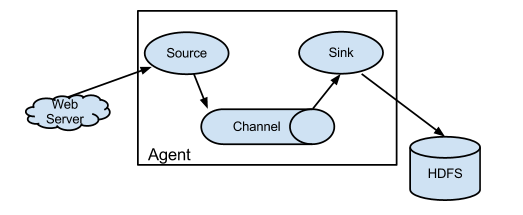
This lab sheet provides a quick introduction to Apache Flume agents. This exercise will introduce Flume components and configuring the Flume agents to see the event flow from source to the sink.

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. It uses a simple extensible data model that allows for online analytic application.

The use of Apache Flume is not only restricted to log data aggregation. Since data sources are customizable, Flume can be used to transport massive quantities of event data including but not limited to network traffic data, social-media-generated data, email messages and pretty much any data source possible.

A Flume event is defined as a unit of data flow having a byte payload and an optional set of string attributes. A Flume agent is a (JVM) process that hosts the components through which events flow from an external source to the next destination (hop).

A Flume source consumes events delivered to it by an external source like a web server. The external source sends events to Flume in a format that is recognized by the target Flume source. For example, an Avro Flume source can be used to receive Avro events from Avro clients or other Flume agents in the flow that send events from an Avro sink. A similar flow can be defined using a Thrift Flume Source to receive events from a Thrift Sink or a Flume Thrift Rpc Client or Thrift clients written in any language generated from the Flume thrift protocol.When a Flume source receives an event, it stores it into one or more channels. The channel is a passive store that keeps the event until it’s consumed by a Flume sink. The file channel is one example – it is backed by the local filesystem. The sink removes the event from the channel and puts it into an external repository like HDFS (via Flume HDFS sink) or forwards it to the Flume source of the next Flume agent (next hop) in the flow. The source and sink within the given agent run asynchronously with the events staged in the channel.



Flume allows a user to build multi-hop flows where events travel through multiple agents before reaching the final destination. It also allows fan-in and fan-out flows, contextual routing and backup routes (fail-over) for failed hops.

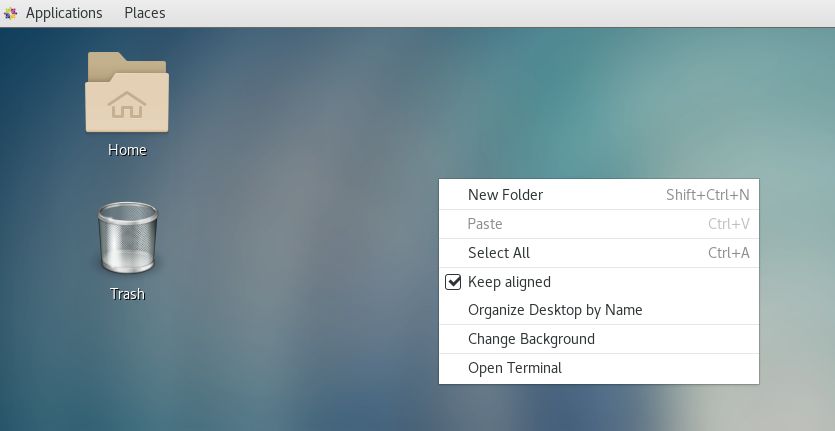
1. Steps to be performed:

Note - It’s assumed that student has made a slot reservation using the slot booking interface where Apache Hadoop framework was selected. The details of the Apache Hadoop systems to be used is received through an email. If not, please contact the administrators for the same.

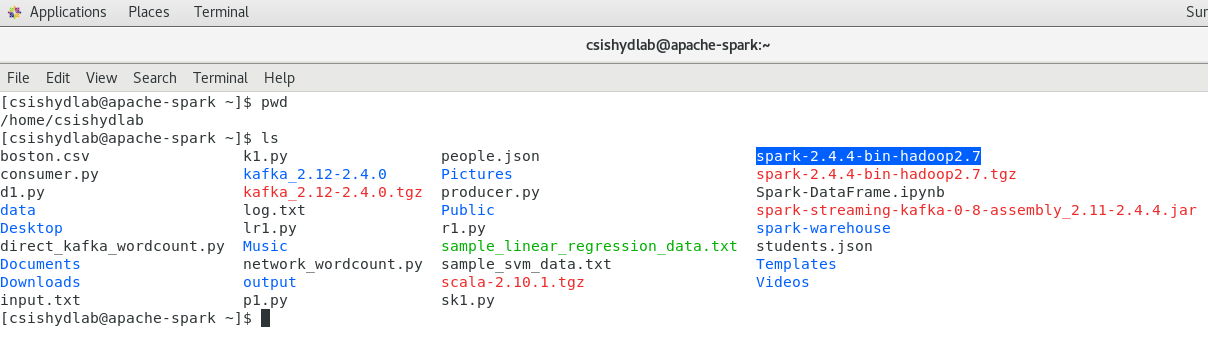
Also it’s assumed that we are aware of the process of logging into these virtual machines. If not, then get access to the user manual maintained for the usage of remote lab setup.

**Getting started with simple Flume Agent -**

1. Open the terminal by right clicking on the desktop of the virtual machine.



1. Look at the current directory and also file listings in it. It must have a Flume named “Flume” installation directory present in it. Commands like pwd, ls can be used for it.



1. If “Flume” installation directory is not present, then follow the steps to install it on the system.

* Create a directory named “Flume” and into it

mkdir Flume

cd Flume

* Download the Flume installer

wget <https://downloads.apache.org/flume/1.9.0/apache-flume-1.9.0-bin.tar.gz>

* Extract the tarball into current directory

tar zxvf apache-flume-1.9.0-bin.tar.gz

1. Set the following environment variables into .bashrc file present in your home directory

export FLUME\_HOME=/home/hadoop/Flume

export PATH=$PATH:/FLUME\_HOME/bin

export CLASSPATH=$CLASSPATH:/FLUME\_HOME/bin

1. Install the netcat on the system using the root user previleges

sudo

yum install -y nc

1. Create a netcat.conf file in Flume config directory

vi Flume/conf/netcat.conf

1. Add the following configuration properties into it which defines the data flow from netcat source to the logger sink through the memory channel.

#Define the components

agent1.sources=src1

agent1.channels=chn1

agent1.sinks=snk1

#Define the source type and properties

agent1.sources.src1.type=netcat

agent1.sources.src1.bind=localhost

agent1.sources.src1.port=12345

#Define the sink type and properties

agent1.sinks.snk1.type=logger

#Define the channel type and properties

agent1.channels.chn1.type=memory

agent1.channels.chn1.capacity=10000

#Connect the source and sink with the channel

agent1.sources.src1.channels=chn1

agent1.sinks.snk1.channel=chn1

1. From the Flume directory , start the flume agent as follows

./bin/flume-ng agent -n agent1 -c conf -f conf/netcat.conf -Dflume.root.logger=INFO,console

1. Open another terminal to start a netcat client into it.

nc localhost 12345

1. Insert some data into the netcat terminal

This is demo message

This is next event

This is working

1. In the terminal where Flume agent is running, you can note that these messages are shown, which means the events are flowing from your netcat source to console logger sink through the memory channel.

**Using SpoolDirectory source Flume Agent –**

Apache Flume source is the component of the Flume agent which receives data from external sources and passes it on to the one or more channels. It consumes data from an external source like a web server. The external data source sends data to Apache Flume in a format that is recognizable by the target Flume source.

Spooling Directory Source

This Apache Flume source allows us to ingest data by placing files that are to be ingested into a “spooling” directory on disk. The Spooling Directory source will look at the specified directory for new files.This source will parse data out of new files as they appear. The data parsing logic is pluggable. When a given file is fully read into the channel, then by default the completion is indicated either by renaming the file or deleting the file. The trackerDir is used for keeping track of processed files. Spooling Directory Source is reliable and does not lead to data loss even if the Apache Flume is restarted or killed.

File Channel

It is the Flume’s persistent channel. File channel writes out all the flume events to the disk. It does not lose data even when the process or machine shuts down or crashes. This channel ensures that any events which are committed into the channel are removed from it only when a sink consumes the events and commits the transaction. It does this even if the machine/agent crashed and was restarted.

The file channel is highly concurrent and handles several flume sources and sinks at the same time. It is best for flows where we require data durability and can’t tolerate data loss.

1. Remove the /tmp/ws\_logs directory if its present.

rm -rf /tmp/ws\_logs

1. Create the /tmp/ws\_logs

mkdir /tmp/ws\_logs

1. Remove the /tmp/flume\_file\_logs directory if its present.

rm -rf /tmp/flume\_file\_logs

1. Create the /tmp/ws\_logs

mkdir /tmp/flume\_file\_logs

1. Remove the /tmp/flume\_logs\_dir directory if its present.

rm -rf /tmp/flume\_logs\_dir

1. Create the /tmp/ flume \_logs\_dir

mkdir /tmp/flume\_logs\_dir

1. Create a spool-dir.conf file

vi Flume/conf/spool.conf

1. Add the following configuration into it.

#Define the components

agent1.sources=src1

agent1.channels=chn1

agent1.sinks=snk1

#Define the source type and properties

agent1.sources.spoolsrc.type = spooldir

agent1.sources.spoolsrc.spoolDir = /tmp/ws\_logs

#Define the sink type and properties

agent1.sinks.snk1.type=logger

#Define the channel type and properties

agent1.channels.chn1.type = file

agent1.channels.chn1.checkpointDir = /tmp/flume\_file\_logs

agent1.channels.chn1.dataDirs = /tmp/flume\_logs\_dir

#Connect the source and sink with the channel

agent1.sources.src1.channels=chn1

agent1.sinks.snk1.channel=chn1

1. From the Flume directory , start the flume agent as follows

./bin/flume-ng agent -n agent1 -c conf -f conf/spool.conf -Dflume.root.logger=INFO,console

1. Open another terminal and change directory to /tmp/ws\_logs

cd /tmp/ws\_logs

1. Create a file into it.

vi l1.log

1. In the terminal where Flume agent is running, you can note the messages related to operations those are happening related to file channel. As this channel writes data onto the disk, so it does not lose any data even on crash or failure. This is also advantageous for having a very large capacity, especially when compared to the Memory Channel.

**Using Interceptors with Flume Agent –**

Let’s see how interceptors can be used in the Flume agents to direct the selected messages to the channel.

1. Create a regex-filter.conf file

vi Flume/conf/regex-filter.conf

1. Add the following configuration into it.

#Define the components

agent1.sources=src1

agent1.channels=chn1

agent1.sinks=snk1

#Define the source type and properties

agent1.sources.src1.type=netcat

agent1.sources.src1.bind=localhost

agent1.sources.src1.port=44444

#Define the sink type and properties

agent1.sinks.snk1.type=logger

#Define the channel type and properties

agent1.channels.chn1.type=memory

agent1.channels.chn1.capacity=10000

#Connect the source and sink with the channel

agent1.sources.src1.channels=chn1

agent1.sinks.snk1.channel=chn1

## Describe regex\_filter interceptor and configure exclude events attribute

agent1.sources.src1.interceptors = int1

agent1.sources.src1.interceptors.int1.type = regex\_filter

agent1.sources.src1.interceptors.int1.regex = ios

agent1.sources.src1.interceptors.int1.excludeEvents = true

1. From the Flume directory , start the flume agent as follows

./bin/flume-ng agent -n agent1 -c conf -f conf/regex-filter.conf -Dflume.root.logger=INFO,console

1. Open another terminal to start a netcat client into it.

nc localhost 44444

1. Insert some data into the netcat terminal

apple, ios, app1, 340

google, android, a1, 287

google, android, a2, 87

apple, ios, app2, 40

google, android, a1, 871

google, android, a2, 287

apple, ios, app3, 240

google, android, a1, 287

google, android, a2, 87

google, android, a3, 123

apple, ios, app1, 340

google, android, a1, 287

apple, ios, app2, 340

1. In the terminal where Flume agent is running, you can note that the messages coming from google phones based on “android” are shown ,the one which are coming from apple phones based on “ios” are not coming to the sink, which means the events are flowing from your netcat source to console logger sink through the memory channel and in between are intercepted to filter out the messages.
2. Outputs/Results:

* install Apache Flume on the system
* configure the agent for simple data flow
* configure the agent for data flow involving interceptors

1. Observations:

Carefully needs to observe

* Source, Sink and Channel configuration properties mentioned in the Flume agent configuration
* Interceptor configuration in Agent definition
* Data flow from netcat source to logger console sink via memory channel

1. References:
2. [Flume documentation](https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html)
3. [Flume Agent Configuration](https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html#setting-up-an-agent)