Q1-What is meant by FlumeNG?

Answer:

1. To solve certain known issues and limitations, Flume requires a refactoring of some core classes and systems.

2. This bug is a parent issue to track the development of a "Flume NG" - a poorly named, but necessary refactoring.

3. Subtasks should be added to track individual systems and components.

4. The following known issues are specifically to be addressed:

* Code complexity; Flume has evolved over the last few years and has a fair amount of extraneous code.
* Core component lifecycle standardization and control code (e.g. anything that can be start()ed or stop()ed, sources, sinks).
* (Static) Configuration access throughout the code base.
* Drastic simplification of common data paths (e.g. durability as an element of the source rather than a disconnected sink).
* Heartbeat and master rearchitecture.
* Renaming packages to org.apache.flume

5. Flume NG's high level architecture solidifies a few concepts from Flume OG and drastically simplifies others.

Q2- Can Flume provides 100 % reliability to the data flow?

Answer:

1. **Yes**, Flume provides end-to-end reliability of the flow.

2. By default Flume uses a transactional approach in the data flow.

3. Sources and sinks encapsulated in a transactional repository provides by the channels.

4. This channel responsible to pass reliably from end to end in the flow. So it provides 100% reliability in data flow.

Q3- Can Flume can distributes data to multiple destinations?

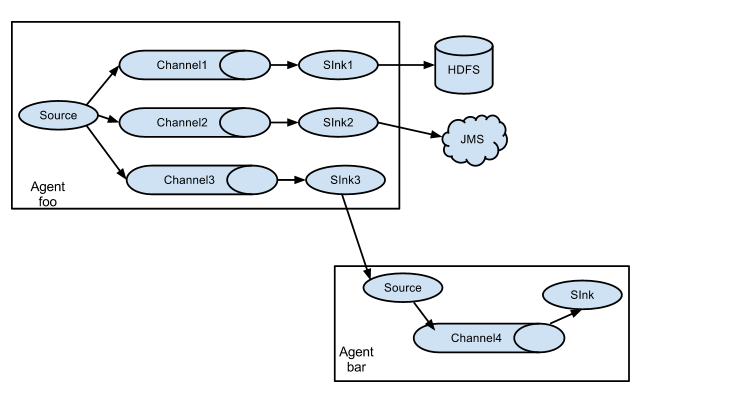
Answer:

1. **Yes,** it supports multiplexing flow.

**2.** The event flows from one source to multiple channels and multiple destinations.

3. It is achieved by defining a flow multiplexer.

4. In the example below, data flows and replicated to HDFS and another sink to destination and another destination is input to another agent.



Q4- Explain about the different channel types in Flume. And which channel type is faster?

Answer:

The 3 different built in channel types available in Flume are-  
1. MEMORY Channel:

Events are read from the source into memory and passed to the sink.  
2. JDBC Channel:

JDBC Channel stores the events in an embedded Derby database.  
3. FILE Channel:

File Channel writes the contents to a file on the file system after reading the event from a source. The file is deleted only after the contents are successfully delivered to the sink.

**MEMORY Channel is the fastest channel among the three however has the risk of data loss.**