**Explain in detail.**

**●What is meant by FlumeNG?**

This is the top level section for all Flume NG documentation. Flume NG is a refactoring of Flume and was originally tracked in [FLUME-728](https://issues.apache.org/jira/browse/FLUME-728). From the JIRA's description:

To solve certain known issues and limitations, Flume requires a refactoring of some core classes and systems. This bug is a parent issue to track the development of a "Flume NG" - a poorly named, but necessary refactoring. Subtasks should be added to track individual systems and components.

 Flume NG retains Flume OG's general approach to data transfer and handling

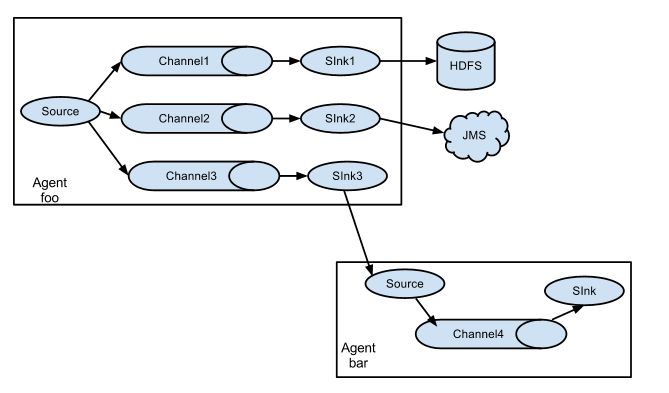
**● Can Flume provides 100 % reliability to the data flow?**

Yes, it provides end-to-end reliability of the flow. By default flume uses a transactional approach in the data flow. Sources and sink encapsulated in a transactional repository provides by the channels. This channels responsible to pass reliably from end to end in the flow. So it provides 100% reliability to the data flow.

**● Can Flume distribute data to multiple destinations?**

Yes, it supports multiplexing flow. The event flows from one source to multiple channels and multiple destinations. It’s achieved by defining a flow multiplexer.

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



**● Explain about the different channel types in Flume. And which channel type is faster?**

The 3 different built in channel types available in Flume are-

MEMORY Channel – Events are read from the source into memory and passed to the sink.

JDBC Channel – JDBC Channel stores the events in an embedded Derby database.

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. The channel that you choose completely depends on the nature of the big data application and the value of each event.