Problem Statement

Explain in detail.

1. **What is meant by FlumeNG ?**

* Flume was basically designed to provide a distributed, reliable, and highly available system for efficiently collecting, aggregating and moving large amounts of log based data which is collected from many sources and stored to a centralized system generally known as HDFS.
* Architecture of Flume-NG comes into existence to achieve all these mentioned features.
* Flume has some limitations which are-
  + Code Complexity
  + Core component lifecycle standardization and control code.
  + Configuration access throughout the code base.
  + Drastic simplification of common data paths.
  + Heartbeat and master re-architecture.
  + Renaming packages to org.apache.flume.

These all the problems were addressed by Flume-NG which is a refactored version of Flume.

* Architecture of Flume NG includes following components-

1. Event- A byte payload with optional string headers that represent the unit of data that Flume can transport from its source to destination.
2. Flow- Flow is the movement of events from source to destination unit.
3. Client- It is an interface implementation which operates at the source of events and delivers them to Flume agent. Client operate in the process space of application they are consuming data from.
4. Agent- Agent is an independent process which host Flume components such as sources, channels and sinks, thus it has ability to receive, store and forward events to their next-hop destination.
5. Source- It is also an interface implementation that can consume events delivered to it.
6. Channel- It’s a store for events, where events are delivered to the channel via sources operating within the agent. An event put in a channel stays in that channel unit a sink removes it for further transport.
7. Sink- It is an interface implementation that can remove events from a channel and transmit them to the next agent in the flow. It is also known as terminal sinks
8. **Can Flume provides 100 % reliability to the data flow?**

Yes, Flume provides end-to-end reliability of the flow. It uses a transactional approach in the data flow.

Source and sink encapsulate in a transactional repository provides by the channels. This channel is responsible for end-to-end reliable flow.

Hence Flume provides 100% reliability to data flow.

1. **Can Flume can distributes data to multiple destinations?**

Yes, Flume can distribute data to multiple destinations. The event flows from one source to multiple channels and multiple destinations which is achieved by flow multiplexer.

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

Channels are the repositories where the events are staged on an agent, Source adds the events and sink removes it.

There are multiple channels of Flume-

1. Memory Channel- Events are read from the source into memory and passed to the sink. Events are stored in an in-memory queue with configurable max size.
2. JDBC Channel- JDBC channel stores the events in an embedded Derby database. The events are stored in a persistent storage that is backed by a database. This is a durable channel that is ideal for flows where recoverability is important.
3. File Channel- File channel writes the contents to a file system after reading the event from a source and these files are deleted only when the contents are successfully delivered to the sink.
4. Kafka Channel- The events are stored in a Kafka cluster. Kafka provides high availability and replication, so if in case any of the agent or Kafka crashes, the events are immediately available to other sinks.
5. Custom Channel- A custom channel is your own implementation of the channel interface. A custom channel’s class ad its dependencies must be included in the agent’s class path when starting the Flume agent.