211720106071

1.What is Flume?

Apache Flume is a distributed, reliable, and available software for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows.

2.Explain the core components of Flume.

A Flume data flow is made up of five main components: Events, Sources, Channels, Sinks, and Agents. An event is the basic unit of data that is moved using Flume. It is similar to a message in JMS and is generally small. It is made up of headers and a byte-array body.

3.What is an Agent?

An agent, in legal terminology, is a person who has been legally empowered to act on behalf of another person or an entity. An agent may be employed to represent a client in negotiations and other dealings with third parties. The agent may be given decision-making authority.

4.What is a channel?

Flume channel is one of the components of a Flume agent. It sits in between flume sources and flume sinks. Channels ensure no loss of data in Flume. Sources write data to the Flume channel. The data written to the Flume channel are consumed by Flume sinks.

211720106071

5.What is Kafka?

Apache Kafka is a distributed event store and stream-processing platform. It is an open-source system developed by the Apache Software Foundation written in Java and Scala. The project aims to provide a unified, high-throughput, lowlatency platform for handling real-time data feeds.

6.List the various components in Kafka.

The main Kafka components are topics, producers, consumers, consumer groups, clusters, brokers, partitions, replicas, leaders, and followers.

7.What is the role of the ZooKeeper?

ZooKeeper is an open source Apache project that provides a centralized service for providing configuration information, naming, synchronization and group services over large clusters in distributed systems. The goal is to make these systems easier to manage with improved, more reliable propagation of changes.

8.Why are Replications critical in Kafka?

The purpose of adding replication in Kafka is for stronger durability and higher availability. We want to guarantee that any successfully published message will not be lost and can be consumed, even when there are server failures.