Transcript name: Flume – Part 2

English
Much like Hadoop, Flume supports three modes of operation: single node, pseudo-distributed, and fully distributed. Single node is useful for basic testing and getting up and running quickly, pseudo-distributed is a more production like environment that lets you build more complicated flows while testing on a single physical machine, and fully distributed is the mode you run in for production. The fully-distributed mode offers two further sub-modes: a standalone mode with a single master and a distributed mode with multiple masters. Let us start by examining the single node mode. The dump command gives you a source and sink all in one command. You run the flume command and specify the word "dump" and an argument. The console argument echos stdin to stdout. The text argument lets you dump the contents of a text file to stdout. Tail is much like the unix tail command, streaming data out as it arrives. Multitail is simply a version of tail that reads from multiple files at once. There are many other sources. All of these sources are the same as you would use in the pseudo-distributed and fully distributed modes. Pseudo-distributed mode runs like a real production Flume environment, except on a single machine. You use it by starting up a master daemon and at least one node daemon, both of which can be configured through an http interface. Here we have the web interface for a Flume master node. You configure it by clicking the config link at the top. This takes you to a "Flume Master: Configure Nodes" page. To configure a node, you choose the node from the dropdown list. You specify a source and a sink for the node. In this case, we will simply configure it like the "flume dump console" command and make both the source and sink a console. There are many other event sinks than consoles. There is the null sink for simply discarding events, the text sink for writing to a text file, and a dfs sink for writing to a distributed file system such as HDFS. One limitation with writing directly from source to a dfs sink is that there is no durability guarantee until the file is closed. There is a way to compensate for this, but it involves introducing a collector tier into our configuration. So this time, we start a master and two nodes: one node to serve as a collector and another to serve as an agent. Using the -n option, we can give them names when we

start them. Now we simply have to connect them to each other, connect the agent to the source, and connect the collector to the sink. You see the number to use to identify the collector when you start it. In this case it is 35853.

You specify this configuration on the same page as before, but under the "Configure multiple nodes" heading.

Finally, fully-distributed mode can be configured by editing the flume-site.xml file in flume's conf directory. You specify the names of the machines you want to use as master servers. You have the option to specify more than one.

Data flows through Flume in the form of events. Events have two parts: the body and the metadata. The body is just a string of bytes containing the content of the event. The size limit is under your control, but defaults to 32 kilobytes. The metadata is in the form of a table of key/value pairs. Flume can inspect the metadata to make routing decisions. Some examples of metadata include the time of creation of the event and the machine that created the event.

This concludes this lesson. Thank you for watching.