COURSES	LESSONS	ENROLLED
9	418	2,202

From 0 to 1 : Hive for Big Data Processing

Connect the Dots Between SQL & Hive to Enhance Your Big Data Processing Skills

1. You, Us & This Course
    1. You, Us & This Course (2:02)
2. Introducing Hive
    1. Hive: An Open-Source Data Warehouse (12:59)
    2. Hive and Hadoop (9:19)
    3. Hive vs Traditional Relational DBMS (13:52)
    4. HiveQL and SQL (7:20)
3. Hadoop and Hive Install
    1. Hadoop Install Modes (8:32)
    2. Setting up a Virtual Linux Instance - For Windows Users (13:50)
    3. Hadoop Install Step 1 : Standalone Mode (9:33)
    4. Hadoop Install Step 2 : Pseudo-Distributed Mode (14:25)
    5. Hive install (12:05)
    6. Code-Along: Getting started (6:24)
4. Hadoop and HDFS Overview
    1. What is Hadoop? (7:25)
    2. HDFS or the Hadoop Distributed File System (11:01)
5. Hive Basics
    1. Primitive Datatypes (17:07)
    2. Collections*Arrays*Maps (9:28)
    3. Structs and Unions (5:57)
    4. Create Table (13:15)
    5. Insert Into Table (12:05)
    6. Insert into Table 2 (6:51)
    7. Alter Table (7:22)
    8. HDFS (9:25)
    9. HDFS CLI - Interacting with HDFS (10:59)
    10. Code-Along: Create Table (9:54)
    11. Code-Along : Hive CLI (3:06)
6. Built-in Functions
    1. Three types of Hive functions (6:45)
    2. The Case-When statement, the Size function, the Cast function (10:09)
    3. The Explode function (13:07)
    4. Code-Along : Hive Built - in functions (4:28)
7. Sub-Queries
    1. Quirky Sub-Queries (7:13)
    2. More on subqueries: Exists and In (15:13)
    3. Inserting via subqueries (5:23)

4. Code-Along : Use Subqueries to work with Collection Datatypes (5:57)
5. Views (12:18)

8. Partitioning

    1. Indices (6:40)
    2. Partitioning Introduced (6:36)
    3. The Rationale for Partitioning (6:16)
    4. How Tables are Partitioned (9:52)
    5. Using Partitioned Tables (5:27)
    6. Dynamic Partitioning: Inserting data into partitioned tables (12:44)
    7. Code-Along : Partitioning (4:03)

9. Bucketing

    1. Introducing Bucketing (11:56)
    2. The Advantages of Bucketing (4:54)
    3. How Tables are Bucketed
    4. Using Bucketed Tables (7:22)
    5. Sampling (11:13)

10. Windowing

    1. Windowing Introduced (12:59)
    2. Windowing - A Simple Example: Cumulative Sum (9:39)
    3. Windowing - A More Involved Example: Partitioning (11:55)
    4. Windowing - Special Aggregation Functions (15:08)

11. Understanding MapReduce

    1. The basic philosophy underlying MapReduce (8:49)
    2. MapReduce - Visualized and Explained (9:03)
    3. MapReduce - Digging a little deeper at every step (10:21)

12. MapReduce logic for queries: Behind the scenes

    1. MapReduce Overview: Basic Select-From-Where (11:33)
    2. MapReduce Overview: Group-By and Having (9:12)
    3. MapReduce Overview: Joins (14:17)

Learn By Example: Hadoop & MapReduce for Big Data Problems

Discover Mass Data Processing Methods by Using the Leading Data Frameworks

1. Introduction
    1. You, this course and Us (1:52)
2. Why is Big Data a Big Deal
    1. The Big Data Paradigm (14:20)
    2. Serial vs Distributed Computing (8:37)
    3. What is Hadoop? (7:25)
    4. HDFS or the Hadoop Distributed File System (11:01)
    5. MapReduce Introduced (11:39)
    6. YARN or Yet Another Resource Negotiator (4:00)
3. Installing Hadoop in a Local Environment
    1. Hadoop Install Modes (8:32)
    2. Setup a Virtual Linux Instance (For Windows users) (15:31)
    3. Hadoop Standalone mode Install (9:33)
    4. Hadoop Pseudo-Distributed mode Install (14:25)
4. The MapReduce "Hello World"
    1. The basic philosophy underlying MapReduce (8:49)
    2. MapReduce - Visualized And Explained (9:03)
    3. MapReduce - Digging a little deeper at every step (10:21)
    4. "Hello World" in MapReduce (10:29)
    5. The Mapper (9:48)
    6. The Reducer (7:46)
    7. The Job (12:28)
5. Run a MapReduce Job
    1. Get comfortable with HDFS (10:59)
    2. Run your first MapReduce Job (14:30)
6. Juicing your MapReduce - Combiners, Shuffle and Sort and The Streaming API
    1. Parallelize the reduce phase - use the Combiner (14:40)
    2. Not all Reducers are Combiners (14:31)
    3. How many mappers and reducers does your MapReduce have? (8:23)
    4. Parallelizing reduce using Shuffle And Sort (14:55)
    5. MapReduce is not limited to the Java language - Introducing the Streaming API (5:05)
    6. Python for MapReduce (12:19)
7. HDFS and Yarn
    1. HDFS - Protecting against data loss using replication (15:32)
    2. HDFS - Name nodes and why they're critical (6:48)
    3. HDFS - Checkpointing to backup name node information (11:10)
    4. Yarn - Basic components (8:33)
    5. Yarn - Submitting a job to Yarn (13:10)

6. Yarn - Plug in scheduling policies (14:21)
7. Yarn - Configure the scheduler (12:26)

8. Setting up a Hadoop Cluster

    1. Manually configuring a Hadoop cluster (Linux VMs) (13:50)
    2. Getting started with Amazon Web Servicies (6:25)
    3. Start a Hadoop Cluster with Cloudera Manager on AWS (13:04)

9. MapReduce Customizations For Finer Grained Control

    1. Setting up your MapReduce to accept command line arguments (13:47)
    2. The Tool, ToolRunner and GenericOptionsParser (12:36)
    3. Configuring properties of the Job object (10:41)
    4. Customizing the Partitioner, Sort Comparator, and Group Comparator (15:16)

10. The Inverted Index, Custom Data Types for Keys, Bigram Counts and Unit Tests!

    1. The heart of search engines - The Inverted Index (14:41)
    2. Generating the inverted index using MapReduce (10:25)
    3. Custom data types for keys - The Writable Interface (10:23)
    4. Represent a Bigram using a WritableComparable (13:13)
    5. MapReduce to count the Bigrams in input text (8:26)
    6. Test your MapReduce job using MRUnit (13:41)

11. Input and Output Formats and Customized Partitioning

    1. Introducing the File Input Format (12:48)
    2. Text And Sequence File Formats (10:21)
    3. Data partitioning using a custom partitioner (7:11)
    4. Make the custom partitioner real in code (10:25)
    5. Total Order Partitioning (10:10)
    6. Input Sampling, Distribution, Partitioning and configuring these (9:04)
    7. Secondary Sort (14:34)

12. Recommendation Systems using Collaborative Filtering

    1. Introduction to Collaborative Filtering (7:25)
    2. Friend recommendations using chained MR jobs (17:15)
    3. Get common friends for every pair of users - the first MapReduce (14:50)
    4. Top 10 friend recommendation for every user - the second MapReduce (13:46)

13. Hadoop as a Database

    1. Structured data in Hadoop (14:08)
    2. Running an SQL Select with MapReduce (15:31)
    3. Running an SQL Group By with MapReduce (14:02)
    4. A MapReduce Join - The Map Side (14:20)
    5. A MapReduce Join - The Reduce Side (13:08)
    6. A MapReduce Join - Sorting and Partitioning (8:49)
    7. A MapReduce Join - Putting it all together (13:46)

14. K-Means Clustering

    1. What is K-Means Clustering? (14:04)
    2. A MapReduce job for K-Means Clustering (16:33)

3. K-Means Clustering - Measuring the distance between points (13:52)
4. K-Means Clustering - Custom Writables for Input/Output (8:26)
5. K-Means Clustering - Configuring the Job (10:50)
6. K-Means Clustering - The Mapper and Reducer (11:23)
7. K-Means Clustering : The Iterative MapReduce Job (3:40)

Hide Full Curriculum

From 0 to 1 : Spark for Data Science in Python

Make Your Data Fly Using Spark for Analytics, Machine Learning, & Data Science

1. You, This Course and Us
    1. You, This Course and Us (2:15)
2. Introduction to Spark
    1. What does Donald Rumsfeld have to do with data analysis? (8:45)
    2. Why is Spark so cool? (12:23)
    3. An introduction to RDDs - Resilient Distributed Datasets (9:39)
    4. Built-in libraries for Spark (15:37)
    5. Installing Spark (6:42)
    6. The PySpark Shell (4:51)
    7. Transformations and Actions (13:33)
    8. See it in Action : Munging Airlines Data with PySpark - I (10:13)
3. Resilient Distributed Datasets
    1. RDD Characteristics: Partitions and Immutability (12:35)
    2. RDD Characteristics: Lineage, RDDs know where they came from (6:06)
    3. What can you do with RDDs? (11:09)
    4. Create your first RDD from a file (16:11)
    5. Average distance travelled by a flight using map() and reduce() operations (5:50)
    6. Get delayed flights using filter(), cache data using persist() (5:24)
    7. Average flight delay in one-step using aggregate() (15:10)
    8. Frequency histogram of delays using countByValue() (3:26)
    9. See it in Action : Analyzing Airlines Data with PySpark - II (6:25)
4. Advanced RDDs: Pair Resilient Distributed Datasets
    1. Special Transformations and Actions (14:45)
    2. Average delay per airport, use reduceByKey(), mapValues() and join() (18:11)
    3. Average delay per airport in one step using combineByKey() (11:53)
    4. Get the top airports by delay using sortBy() (4:34)
    5. Lookup airport descriptions using lookup(), collectAsMap(), broadcast() (14:03)
    6. See it in Action : Analyzing Airlines Data with PySpark - III (4:58)
5. Advanced Spark: Accumulators, Spark Submit, MapReduce , Behind The Scenes
    1. Get information from individual processing nodes using accumulators (13:35)
    2. See it in Action : Using an Accumulator variable (2:41)
    3. Long running programs using spark-submit (5:58)
    4. See it in Action : Running a Python script with Spark-Submit (3:58)
    5. Behind the scenes: What happens when a Spark script runs? (14:30)
    6. Running MapReduce operations (13:44)
    7. See it in Action : MapReduce with Spark (2:05)
6. Java and Spark
    1. The Java API and Function objects (15:59)
    2. Pair RDDs in Java (4:49)

3. Running Java code (3:49)
4. Installing Maven (2:20)
5. See it in Action : Running a Spark Job with Java (5:09)

7. PageRank: Ranking Search Results

    1. What is PageRank? (16:44)
    2. The PageRank algorithm (6:15)
    3. Implement PageRank in Spark (12:01)
    4. Join optimization in PageRank using Custom Partitioning (7:27)
    5. See it Action : The PageRank algorithm using Spark (3:46)

8. Spark SQL

    1. Dataframes: RDDs + Tables (16:05)
    2. See it in Action : Dataframes and Spark SQL (4:50)

9. MLlib in Spark: Build a recommendations engine

    1. Collaborative filtering algorithms (12:19)
    2. Latent Factor Analysis with the Alternating Least Squares method (11:39)
    3. Music recommendations using the Audioscrobbler dataset (7:51)
    4. Implement code in Spark using MLlib (16:05)

10. Spark Streaming

    1. Introduction to streaming (9:55)
    2. Implement stream processing in Spark using Dstreams (10:54)
    3. Stateful transformations using sliding windows (9:26)
    4. See it in Action : Spark Streaming (4:17)

11. Graph Libraries

    1. The Marvel social network using Graphs (18:01)

Scalable Programming with Scala & Spark

Get Rich Using Scala & Spark for Data Analysis, Machine Learning & Analytics

1. You, This Course and Us
    1. You, This Course and Us (2:16)
2. Introducing Scala
    1. Scala - A "better Java"? (10:13)
    2. Installing Scala and Hello World (9:43)
    3. How do Classes work in Scala? (11:02)
    4. Classes in Scala - continued (15:50)
    5. Functions are different from Methods (7:31)
    6. Collections in Scala (10:12)
    7. Map, Flatmap - The Functional way of looping (11:36)
    8. First Class Functions revisited (8:46)
    9. Partially Applied Functions (7:31)
    10. Closures (8:07)
    11. Currying (10:34)
3. Introduction to Spark
    1. What does Donald Rumsfeld have to do with data analysis? (8:45)
    2. Why is Spark so cool? (12:23)
    3. An introduction to RDDs - Resilient Distributed Datasets (9:39)
    4. Built-in libraries for Spark (15:37)
    5. Installing Spark (11:44)
    6. The Spark Shell (6:55)
    7. See it in Action : Munging Airlines Data with Spark (3:44)
    8. Transformations and Actions (17:06)
4. Resilient Distributed Datasets
    1. RDD Characteristics: Partitions and Immutability (12:35)
    2. RDD Characteristics: Lineage, RDDs know where they came from (6:06)
    3. What can you do with RDDs? (11:09)
    4. Create your first RDD from a file (14:54)
    5. Average distance travelled by a flight using map() and reduce() operations (6:59)
    6. Get delayed flights using filter(), cache data using persist() (6:11)
    7. Average flight delay in one-step using aggregate() (12:21)
    8. Frequency histogram of delays using countByValue() (2:10)
5. Advanced RDDs: Pair Resilient Distributed Datasets
    1. Special Transformations and Actions (14:45)
    2. Average delay per airport, use reduceByKey(), mapValues() and join() (13:35)
    3. Average delay per airport in one step using combineByKey() (8:23)
    4. Get the top airports by delay using sortBy() (2:51)
    5. Lookup airport descriptions using lookup(), collectAsMap(), broadcast() (10:57)
6. Advanced Spark: Accumulators, Spark Submit, MapReduce , Behind The Scenes

1. Get information from individual processing nodes using accumulators (9:25)
2. Long running programs using spark-submit (7:11)
3. Spark-Submit with Scala - A demo (6:10)
4. Behind the scenes: What happens when a Spark script runs? (14:30)
5. Running MapReduce operations (10:53)

7. PageRank: Ranking Search Results

    1. What is PageRank? (16:44)
    2. The PageRank algorithm (6:15)
    3. Implement PageRank in Spark (9:45)
    4. Join optimization in PageRank using Custom Partitioning (6:28)

8. Spark SQL

    1. Dataframes: RDDs + Tables (15:48)

9. MLlib in Spark: Build a recommendations engine

    1. Collaborative filtering algorithms (12:19)
    2. Latent Factor Analysis with the Alternating Least Squares method (11:39)
    3. Music recommendations using the Audioscrobbler dataset (5:38)
    4. Implement code in Spark using MLlib (14:45)

10. Spark Streaming

    1. Introduction to streaming (9:55)
    2. Implement stream processing in Spark using Dstreams (9:19)
    3. Stateful transformations using sliding windows (8:17)

11. Graph Libraries

    1. The Marvel social network using Graphs (14:30)

Learn by Example: HBase - The Hadoop Database

Create More Flexible Databases by Mastering HBase

1. You, This Course and Us
    1. You, This Course and Us (1:50)
2. Introduction to HBase
    1. The problem with distributed computing (7:17)
    2. Installing HBase (10:57)
    3. The Hadoop ecosystem (8:01)
    4. The role of HBase in the Hadoop ecosystem (9:42)
    5. How is HBase different from RDBMS? (3:10)
    6. HBase Data Model (10:44)
    7. Introducing CRUD operations (8:32)
    8. HBase is different from Hive (4:48)
3. CRUD operations using the HBase Shell
    1. Example1 - Creating a table for User Notifications (5:24)
    2. Example 2 - Inserting a row (19:52)
    3. Example 3 - Updating a row (19:15)
    4. Example 4 - Retrieving a row (20:25)
    5. Example 5 - Retrieving a range of rows (3:48)
    6. Example 6 - Deleting a row (2:11)
    7. Example 7 - Deleting a table (2:17)
4. CRUD operations using the Java API
    1. Example 8 - Creating a table with HBaseAdmin (6:36)
    2. Example 9 - Inserting a row using a Put object (8:33)
    3. Example 10 - Inserting a list of Puts (3:30)
    4. Example 11 - Retrieving data - Get and Result objects (10:55)
    5. Example 12 - A list of Gets (3:34)
    6. Example 13 - Deleting a row (2:25)
    7. Example 14 - A list of Deletes (2:36)
    8. Example 15 - Mix and match with batch operations (6:02)
    9. Example 16 - Scanning a range of rows (8:06)
    10. Example 17 - Deleting a table (3:51)
5. HBase Architecture
    1. HBase Architecture (9:20)
6. Advanced operations - Filters and Counters
    1. Example 18 - Filter by Row id - RowFilter (8:56)
    2. Example 19 - Filter by column value - SingleColumnValueFilter (5:13)
    3. Example 20 - Apply multiple conditions - Filterlist (4:31)
    4. Example 21 - Retrieve rows within a time range (2:11)
    5. Example 22 - Atomically incrementing a value with Counters (7:31)

7. MapReduce with HBase

    1. Example 23 : A MapReduce task to count Notifications by Type (10:24)
    2. Example 23 continued: Implementing the MapReduce in Java (13:35)
    3. Demo : Running a MapReduce task (2:21)

8. Build a Notification Service

    1. Example 24 : Implement a Notification Hierarchy (13:30)
    2. Example 25: Implement a Notifications Manager (12:05)

9. Installing Hadoop in a Local Environment

    1. Hadoop Install Modes (8:32)
    2. Setup a Virtual Linux Instance (For Windows users) (15:31)
    3. Hadoop Standalone mode Install (9:33)
    4. Hadoop Pseudo-Distributed mode Install (14:25)

Pig for Wrangling Big Data

Become a Well-Paid Data Handler by Learning to Load, Transform & Extract Data Using Pig

1. You, This Course and Us
    1. You, This Course and Us (1:46)
2. Where does Pig fit in?
    1. Pig and the Hadoop ecosystem (9:37)
    2. Install and set up (8:50)
    3. How does Pig compare with Hive? (10:15)
    4. Pig Latin as a data flow language (6:17)
    5. Pig with HBase (5:18)
3. Pig Basics
    1. Operating modes, running a Pig script, the Grunt shell (9:52)
    2. Loading data and creating our first relation (8:45)
    3. Scalar data types (9:55)
    4. Complex data types - The Tuple, Bag and Map (13:45)
    5. Partial schema specification for relations (10:00)
    6. Displaying and storing relations - The dump and store commands
4. Pig Operations And Data Transformations
    1. Selecting fields from a relation (10:22)
    2. Built-in functions (5:08)
    3. Evaluation functions (10:31)
    4. Using the distinct, limit and order by keywords (5:04)
    5. Filtering records based on a predicate (11:01)
5. Advanced Data Transformations
    1. Group by and aggregate transformations (12:12)
    2. Combining datasets using Join (16:19)
    3. Concatenating datasets using Union (4:32)
    4. Generating multiple records by flattening complex fields (5:24)
    5. Using Co-Group, Semi-Join and Sampling records (9:26)
    6. The nested Foreach command (13:47)
    7. Debug Pig scripts using Explain and Illustrate (12:55)
6. Optimizing Data Transformations
    1. Parallelize operations using the Parallel keyword (8:02)
    2. Join Optimizations: Multiple relations join, large and small relation join (10:34)
    3. Join Optimizations: Skew join and sort-merge join (8:51)
    4. Common sense optimizations (5:25)
7. A real-world example
    1. Parsing server logs (7:55)
    2. Summarizing error logs (8:47)

8. Installing Hadoop in a Local Environment

    1. Hadoop Install Modes (8:32)
    2. Setup a Virtual Linux Instance (For Windows users) (15:31)
    3. Hadoop Standalone mode Install (9:33)
    4. Hadoop Pseudo-Distributed mode Install (14:25)

From 0 to 1 : The Cassandra Distributed Database

Learn the Cassandra Distributed Database & Greatly Improve Your Big Data Resume

1. You, This Course and Us
    1. You, This Course and Us (1:45)
2. Introduction: Cassandra as a distributed, decentralized, columnar database
    1. A Column Oriented Database (10:40)
    2. Requirements For A Product Catalog System (8:08)
    3. What Is Cassandra (8:33)
    4. Cassandra Vs HBase (4:37)
3. Install And Set Up
    1. Install Cassandra (Mac and Unix based systems) (9:54)
    2. Install the Cassandra Cluster Manager (Mac and Unix) (2:21)
    3. Install Maven On Your Machine (2:20)
4. The Cassandra Cluster Manager
    1. Create A Cassandra Cluster On Your Local Machine (11:54)
    2. Basic CCM Commands (7:04)
5. The Cassandra Data Model
    1. Column And Column Family (8:03)
    2. Super Column Family And Keyspace (7:18)
    3. Comparing Cassandra With A Relational Database (4:20)
6. Shell Commands
    1. Connecting To Cassandra And Creating A Keyspace (6:55)
    2. Column Families And Their Properties (12:02)
    3. Modifying Column Families (2:42)
    4. Insert Data Into A Column Family (6:52)
    5. Advanced Data Types Collections And Counters (10:56)
    6. Update Simple And Collection Data Types (15:54)
    7. Manage Cluster Roles (5:01)
7. Keys And Indexes: Primary Keys, Partition Keys, Clustering Key, Secondary Indexe
    1. Partition Keys: Distributing Data Across Cluster Nodes (12:15)
    2. Partition Keys: Properties (5:08)
    3. Clustering Keys: Data Layout On A Node (3:36)
    4. Restrictions On Partition Keys (14:38)
    5. Restrictions On Clustering Keys (9:12)
    6. Secondory Indexes (8:32)
    7. Restrictions On Secondary Indexes (8:52)
    8. Allow Filtering (2:27)
8. Tunable Consistency

1. Write Consistency Levels And Hinted Handoff (12:18)
2. Read Consistency Levels (11:19)
3. Replication Factors And Quorum Value (8:14)

9. Storage Systems

    1. Overview Of Cassandra Storage Components (6:38)
    2. The SS Table And Its Components (9:44)
    3. Row Cache And Key Cache (3:14)
    4. Anatomy Of A Write Request (8:33)
    5. Anatomy Of A Read Request And The Gossip Protocol (7:25)

10. A Mini-Project: A Miniature Catalog Management System

    1. Overview And Basic Setup (4:29)
    2. Creating A Session And Executing Our First Query (7:40)
    3. Create A Column Family (3:27)
    4. Check If A Column Family Has Been Created (4:59)
    5. Insert Data Into The Listings Column Family (9:13)
    6. Insert Data Into The Products Column Family (9:59)
    7. Search For Products (13:32)
    8. Delete A Listing (4:17)
    9. Update Mulitple Column Families Using Logged Batch (14:42)

Oozie: Workflow Scheduling for Big Data Systems

Streamline Your Big Data Workflow by Learning to Use Workflows, Coordinators & Bundles in Oozie

1. Introduction
    1. You, This Course and Us (1:38)
2. A Brief Overview Of Oozie
    1. What is Oozie? (11:16)
    2. Oozie architectural components (10:46)
3. Oozie Install And Set Up
    1. Installing Oozie on your machine (16:29)
4. Workflows: A Directed Acyclic Graph Of Tasks
    1. Running MapReduce on the command line (4:41)
    2. The lifecycle of a Workflow (6:12)
    3. Running our first Oozie Workflow MapReduce application (11:15)
    4. The job.properties file (8:45)
    5. The workflow.xml file (24:14)
    6. A Shell action Workflow (7:46)
    7. Control nodes, Action nodes and Global configurations within Workflows (9:57)
5. Coordinators: Managing Workflows
    1. Running our first Coordinator application (12:27)
    2. A time-triggered Coordinator definition (8:52)
    3. Coordinator control mechanisms (7:09)
    4. Data availability triggers (10:03)
    5. Running a Coordinator which waits for input data (6:11)
    6. Coordinator configuration to use data triggers (15:25)
6. Bundles: A Collection Of Coordinators For Data Pipelines
    1. Bundles and why we need them (9:15)
    2. The Bundle kick-off time (11:12)
7. Installing Hadoop in a Local Environment
    1. Hadoop Install Modes (8:32)
    2. Setup a Virtual Linux Instance (For Windows users) (15:31)
    3. Hadoop Standalone mode Install (9:33)
    4. Hadoop Pseudo-Distributed mode Install (14:25)

Flume & Sqoop for Ingesting Big Data

Efficiently Import Data to HDFS, HBase & Hive From a Variety of Sources & Watch Your Job Prospects Grow

1. You, This Course and Us
    1. You, This Course and Us (1:46)
2. Why do we need Flume and Sqoop?
    1. Why do we need Flume and Sqoop? (18:23)
3. Flume
    1. Installing Flume (2:43)
    2. Flume Agent - the basic unit of Flume (10:57)
    3. Example 1 : Spool to Logger (14:34)
    4. Flume Events are how data is transported (6:07)
    5. Example 2 : Spool to HDFS (9:08)
    6. Example 3: HTTP to HDFS (9:24)
    7. Example 4: HTTP to HDFS with Event Bucketing (5:40)
    8. Example 5: Spool to HBase (6:22)
    9. Example 6: Using multiple sinks and Channel selectors (9:43)
    10. Example 7: Twitter Source with Interceptors (10:48)
4. Sqoop
    1. Installing Sqoop (4:25)
    2. Example 8: Sqoop Import from MySQL to HDFS (7:49)
    3. Example 9: Sqoop Import from MySQL to Hive (4:26)
    4. Example 10: Incremental Imports using Sqoop Jobs (5:24)