

VTUModule 2
Pulse.com

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Hadoop Ecosystem

- **Sqoop:** It is used to import and export data to and from between HDFS and RDBMS.
- **Pig:** It is a procedural language platform used to develop a script for MapReduce operations.
- **Hbase:** HBase is a distributed column-oriented database built on top of the Hadoop file system.
- **Hive:** It is a platform used to develop SQL type scripts to do MapReduce operations.
- **Flume:** Used to handle streaming data on the top of Hadoop.
- **Oozie:** Apache Oozie is a workflow scheduler for Hadoop.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Soop VTUPulse.com Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Apache Sqoop

- In real-world there are many sources which are responsible for generating big data.
- For Example: Twitter, Facebook, Blogs, System Logs, IoT application etc.
- The traditional application management system, that is, the interaction of applications with relational database using RDBMS, is one of the sources that generate Big Data.
- Such Big Data, generated by RDBMS, is stored in **Relational Database Servers** in the relational database structure.
- When Big Data storages and analyzers such as MapReduce, Hive, HBase, Cassandra, Pig, etc. of the Hadoop ecosystem came into picture, they required a tool to interact with the relational database servers for importing and exporting the Big Data residing in them.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Apache Swoop

- Swoop is a tool designed to transfer data between Hadoop and relational databases.
- You can use Swoop to import data from a relational database management system (RDBMS) into the Hadoop Distributed File System (HDFS), transform the data in Hadoop, and then export the data back into an RDBMS.

Swoop – “SQL to Hadoop and Hadoop to SQL”

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

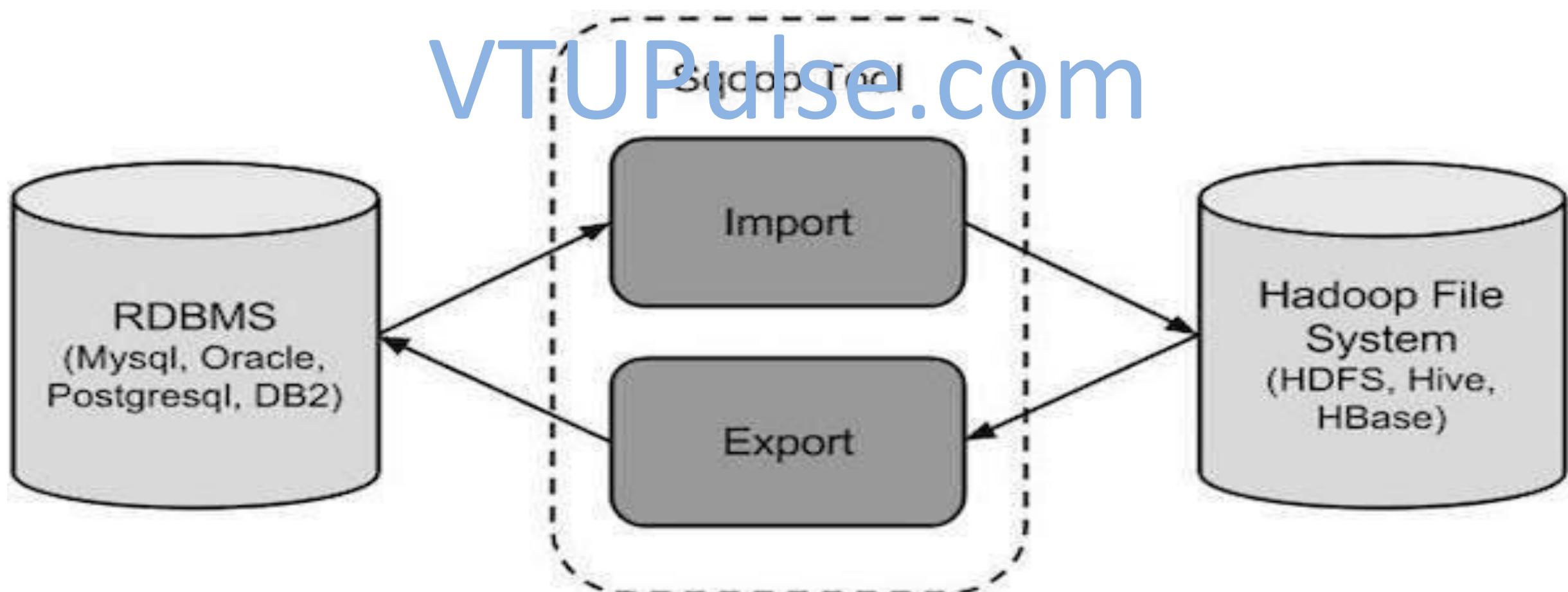
Apache Swoop

- Swoop can be used with any Java Database Connectivity (JDBC) - compliant database and has been tested on Microsoft SQL Server, PostgreSQL, MySQL, and Oracle.
- In version 1 of Swoop, data were accessed using connectors written for specific databases.
- Version 2 does not support connectors, instead it uses generic JDBC connector for data transfer.

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

How Sqoop Works?



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Sqoop Import Method

- The data import is done in two steps.
- In the first step, shown in the figure, Sqoop examines the database to gather the necessary metadata for the data to be imported.
- The second step is a map-only (no reduce step) Hadoop job that Sqoop submits to the cluster.
- This job does the actual data transfer using the metadata captured in the previous step.
- Note that each node doing the import must have access to the database.
- The imported data are saved in an HDFS directory.

Apache Sqoop Import Method

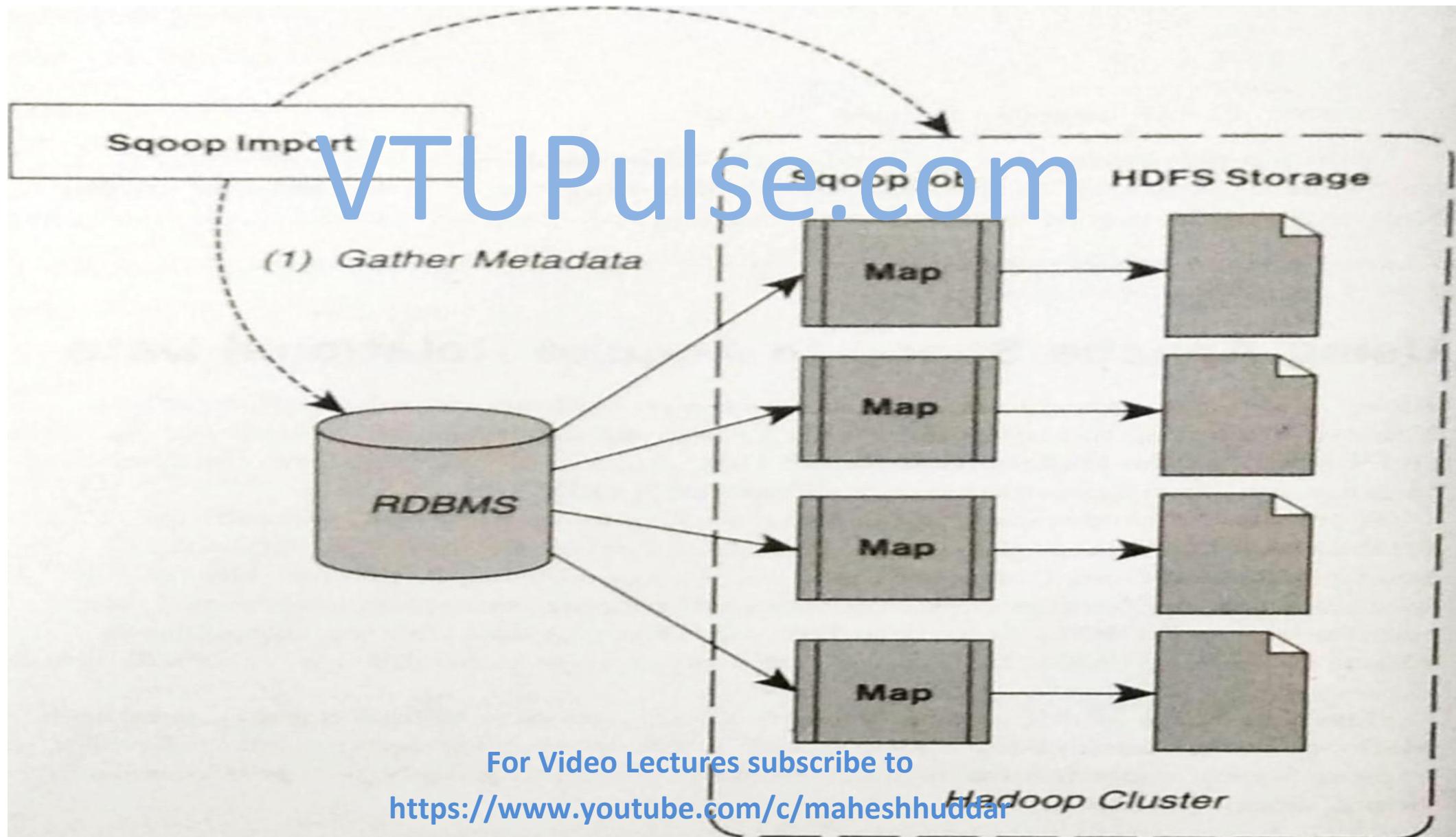
- Sqoop will use the database name for the directory, or the user can specify any alternative directory where the files should be populated.
- By default, these files contain comma-delimited fields, with new lines separating different records. User can also specify the delimiter of his choice.
- For Example:

Name	Age	Dept.
Mahesh Huddar	30	CSE
Rahul	30	ECE

- Content of HDFS is: **Mahesh Huddar,30,CSE\nRahul,30,ECE**
- Once placed in HDFS, the data are ready for processing.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

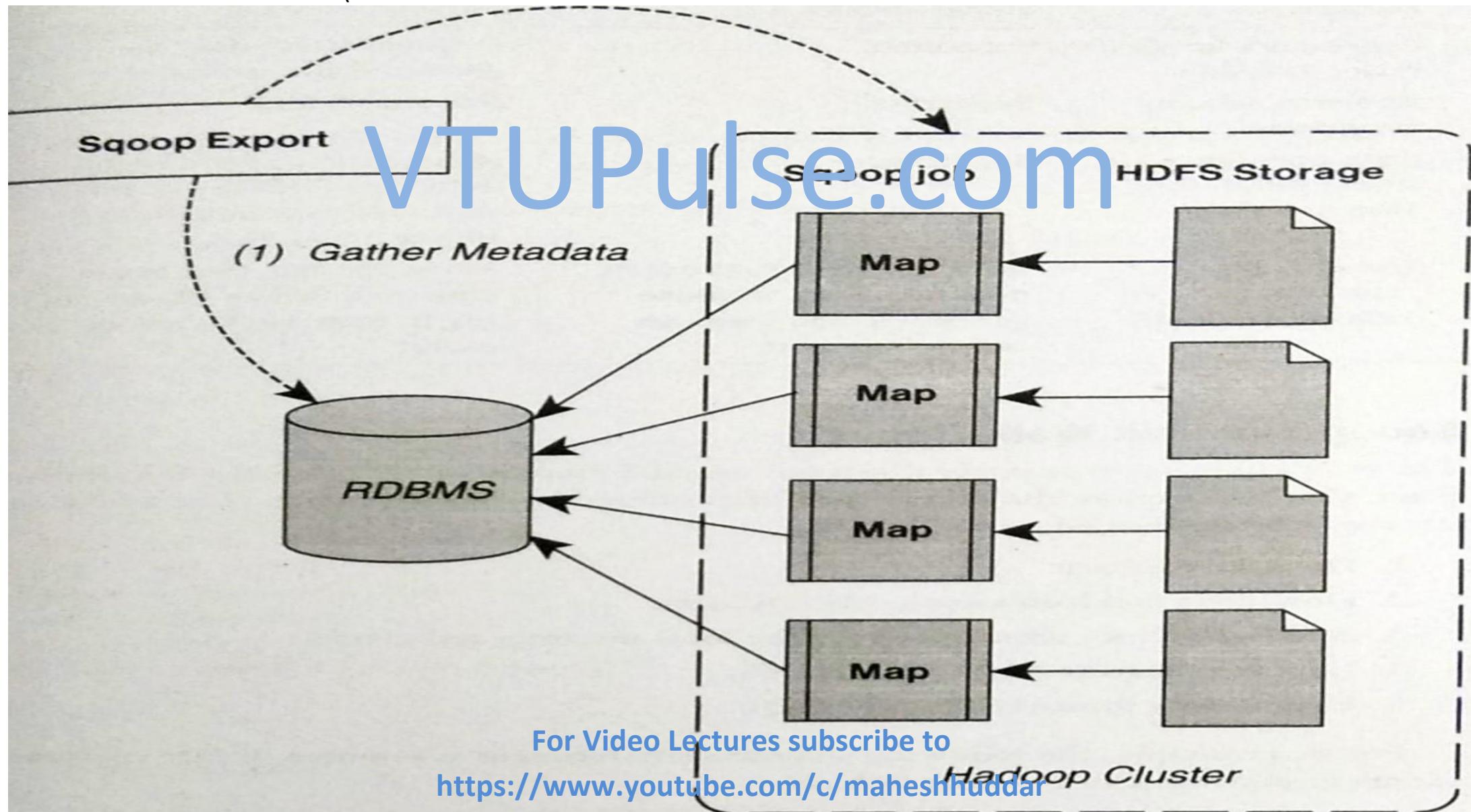
Apache Sqoop Import Method



Apache Sqoop Export Method

- Data export from the HDFS cluster works in a similar fashion.
- The export is done in two steps
- As in the import process, the first step is to examine the database for metadata.
- The export step again uses a map-only Hadoop job to write the data to the database.
- Sqoop divides the input data set into splits, then uses individual map tasks to push the splits to the database.
- Again, this process assumes the map tasks have access to the database.

Apache Sqoop Export Method



Apache Swoop Version Changes

Feature	Swoop Version 1	Swoop Version 2
Connectors for all major RDBMSs	Supported.	Not supported. Use the generic JDBC connector.
Kerberos security integration	Supported.	Not supported.
Data transfer from RDBMS to Hive or HBase	Supported.	Not supported. First import data from RDBMS into HDFS, then load data into Hive or HBase manually.
Data transfer from Hive or HBase to RDBMS	Not supported. First export data from Hive or Hbase into HDFS, and then use Swoop for export.	Not supported. First export data from Hive or HBase into HDFS, then use Swoop for export

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Apache Sqoop Walk-Through Example

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Sqoop Example Walk-Through

- The following steps are performed for data import and export using Sqoop:

VTUPulse.com

1. Download Sqoop and load sample MySQL data.
2. Add Sqoop user permissions for the local machine and cluster.
3. Import data from MySQL to HDFS.
4. Export data from HDFS to MySQL.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 1: Download Sqoop and Load Sample MySQL Database

- If you have not done so already, make sure Sqoop is installed on your cluster.
- Sqoop is needed on only a single node in your cluster.
- This Sqoop node will then serve as an entry point for all connecting Sqoop clients.
- Because the Sqoop node is a Hadoop MapReduce client, it requires both a Hadoop installation and access to HDFS.
- To install Sqoop using the HDP distribution RPM files, simply enter:

yum install sqoop sqoop-metastore

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 1: Download Sqoop and Load Sample MySQL Database

- For demonstration purpose, we use world example database (world_innodb.sql) from MySQL site.
- Contents of data base are:
 - Country: Information about countries of the world.
 - City: Information about some of the cities of world.
 - CountryLanguage: Languages spoken in each country.
- \$wget http://downloads.mysql.com/docs/world_innodb.sql.gz
- Extract
- Gunzip world_innodb.sql.gz

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Step 1: Download Sqoop and Load Sample MySQL Database

```
Mysql -u root -p
```

```
mysql> create database world;
```

```
mysql> use world;
```

```
mysql> source world_innodb.sql
```

```
mysql> show tables;
```

Tables in world

City

Country

CountryLanguage

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 2: Add sqoop user permissions

```
Mysql -u root -p
```

```
mysql>CREATE USER 'sqoop'@'localhost' IDENTIFIED BY 'M@heshg1';
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'Sqoop'@'localhost';
```

```
mysql> quit;
```

```
$ mysql -u sqoop -p
```

```
mysql> use world;
```

```
mysql> show tables;
```

Tables in world

City

Country

CountryLanguage

```
mysql> quit
```

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 3: Import data using Sqoop

- sqoop list-databases --connect jdbc:mysql://localhost --username sqoop --password M@heshg1
 - Information_schema
 - Test
 - world
- sqoop list-tables --connect jdbc:mysql://localhost/world --username sqoop --password M@heshg1
 - City
 - Country
 - CountryLanguage

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 3: Import data using Sqoop

To import data we need to make directory in HDFS

VTUPulse.com

- `hdfs dfs -mkdir /sqoop-mysql-import`

The following command imports the City into HDFS

- `sqoop import --connect jdbc:mysql://localhost/world --username sqoop --password M@heshg1 --table City -m 1 --target-dir /sqoop-mysql-import/city`

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 3: Import data using Sqoop

The Imports can be confirmed by examining the HDFS using **ls** command

VTUPulse.com

- hdfs dfs -ls /sqoop-mysql-import/city

The file can be viewed using the hdfs dfs –cat command

- hdfs dfs -cat /sqoop-mysql-import/city/part-m-00000

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 3: Import data using Sqoop

Options file content – conf.txt

```
import  
--connect  
jdbc:mysql://localhost/world  
--username  
sqoop  
--password  
M@heshg1
```

VTUPulse.com

```
sqoop --options-file conf.txt --table City -m 1 --target-dir /sqoop-mysql-  
import/city
```

For Video Lectures [Subscribe to](#)

<https://www.youtube.com/c/maheshhuddar>

Step 3: Import data using Sqoop

SQL Query Select can be used while importing

VTUPulse.com

- `sqoop import --connect jdbc:mysql://localhost/world --username sqoop --password M@heshg1 -m 1 --target-dir /sqoop-mysql-import/city1 --query "select ID, Name from City where CountryCode='CAN' AND \$CONDITIONS"`
- `sqoop --options-file conf.txt -m 1 --target-dir /sqoop-mysql-import/city1 --query "select ID, Name from City where CountryCode='CAN' AND \$CONDITIONS"`

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 4: Export Data from HDFS to MySQL

- There are two tables are required to exporting tables.
- The first table holds the exported data and second is used for staging the exported data.

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Step 4: Export Data from HDFS to MySQL

- **Actual Table**
- CREATE TABLE `CityExport`(`ID` int(11) NOT NULL AUTO_INCREMENT,
`Name` char(35) NOT NULL DEFAULT "", `CountryCode` char(3) NOT NULL
DEFAULT "", `District` char(20) NOT NULL DEFAULT "", `Population` int(11)
NOT NULL DEFAULT '0', PRIMARY KEY (`ID`));
- **Staging Table**
- CREATE TABLE `CityExportStaging`(`ID` int(11) NOT NULL
AUTO_INCREMENT, `Name` char(35) NOT NULL DEFAULT "", `CountryCode`
char(3) NOT NULL DEFAULT "", `District` char(20) NOT NULL DEFAULT "",
`Population` int(11) NOT NULL DEFAULT '0', PRIMARY KEY (`ID`));

Step 4: Export Data from HDFS to MySQL

- `sqoop export --connect jdbc:mysql://localhost/world --username sqoop --password M@heshg1 --table CityExport --staging-table CityExportStaging --clear-staging-table -m 4 --export-dir /sqoop-mysql-import/city`
- `sqoop export --options-file conf.txt --table CityExport --staging-table CityExportStaging --clear-staging-table -m 4 --export-dir /sqoop-mysql-import/city`

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Flume

VTUPulse.com

Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

What is Flume?

- Apache Flume is a tool used for collecting, aggregating and transporting large amounts of streaming data to a centralized data store.
- Streaming Data is real-time continuously generated data.
- For example: log files, social media generated data, email messages or just any continuous data
- Flume is a highly reliable, distributed, and configurable tool.
- It is principally designed to copy streaming data (continuous data) from various web servers to Hadoop Distributed File System (HDFS).

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

HDFS put Command

- Hadoop File System Shell provides commands to insert data into Hadoop and read from it.
- We can use the **put** command of Hadoop to transfer data to HDFS.
- You can insert data into Hadoop using the **put** command as shown below.

```
$ hadoop dfs –put /path_of_the_required_file /path_in_HDFS  
where_to_save_the_file
```

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

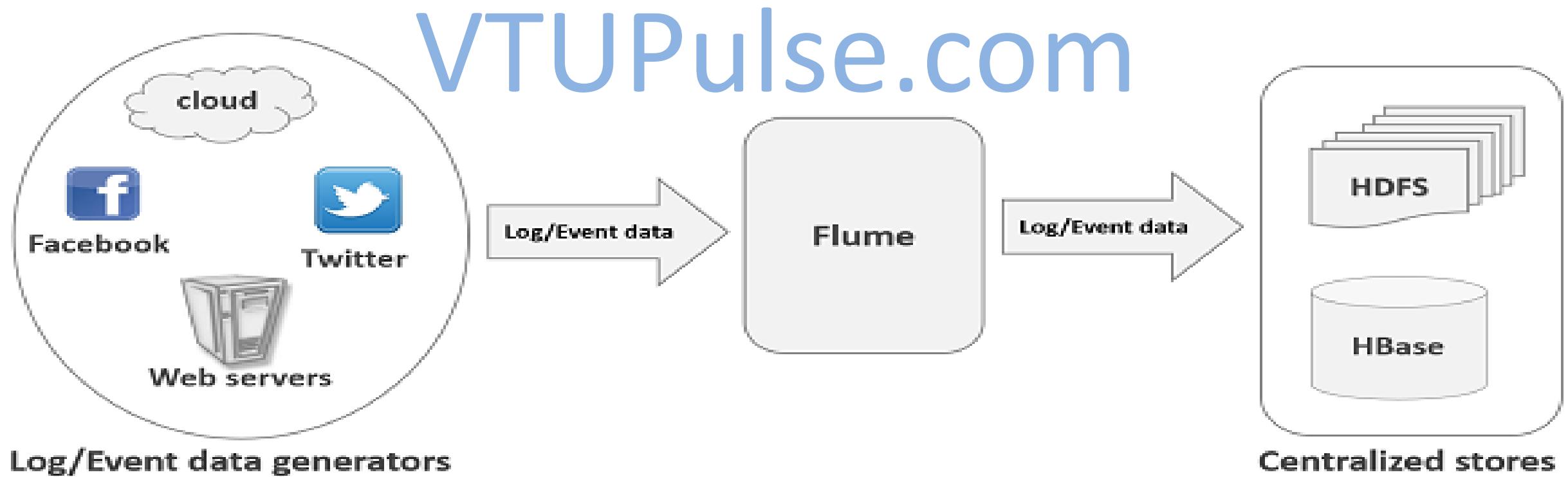
Problem with HDFS put Command

But, it suffers from the following drawbacks

- Using **put** command, we can transfer only one file at a time.
- If we use **put** command, the data is needed to be packaged and should be ready for the upload. Since the web servers generate data continuously, it is a very difficult task.

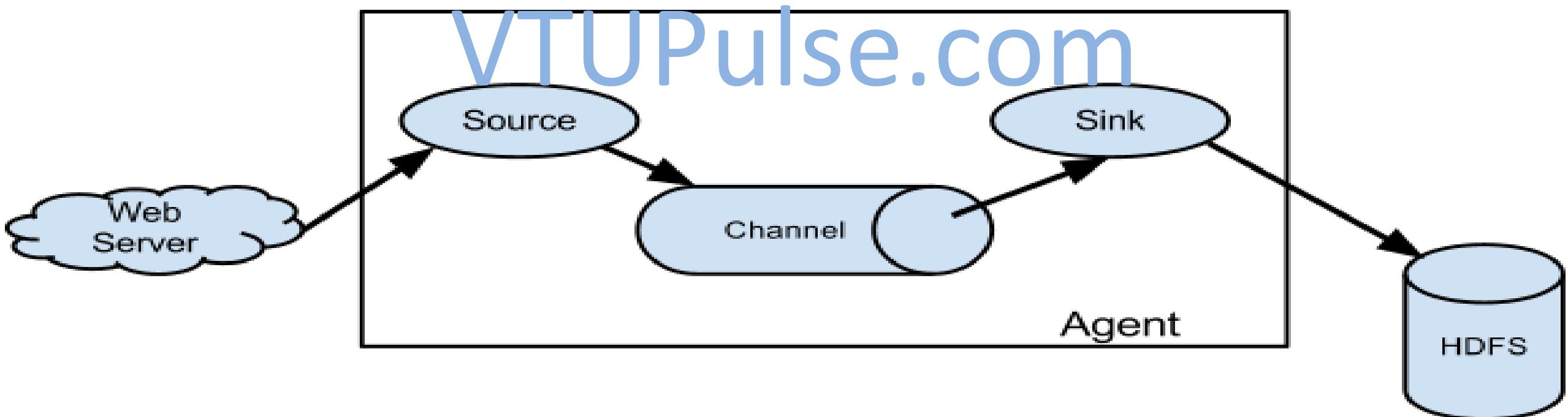
What we need here is a solutions that can overcome the drawbacks of **put** command and transfer the "streaming data" from data generators to centralized stores (especially HDFS) with less delay.

Overview of Data transfer over Apache Flume



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Flume Agent



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Flume Agent

Source

- The source component receives the data and sends it to channel. It can send the data to more than one channel. The input data can be real-time source (example: tweets from twitter, system log file etc.) or another flume agent.

Channel

- A channel is a data queue that forwards the source data to sink destination. It can be thought of as a buffer that manages the input and output data flow rates.
- **Example** – JDBC channel, File system channel, Memory channel, etc.

Sink

- A **sink** delivers the data to centralized stores like HBase and HDFS or another flume agent.

For Video Lectures subscribe to

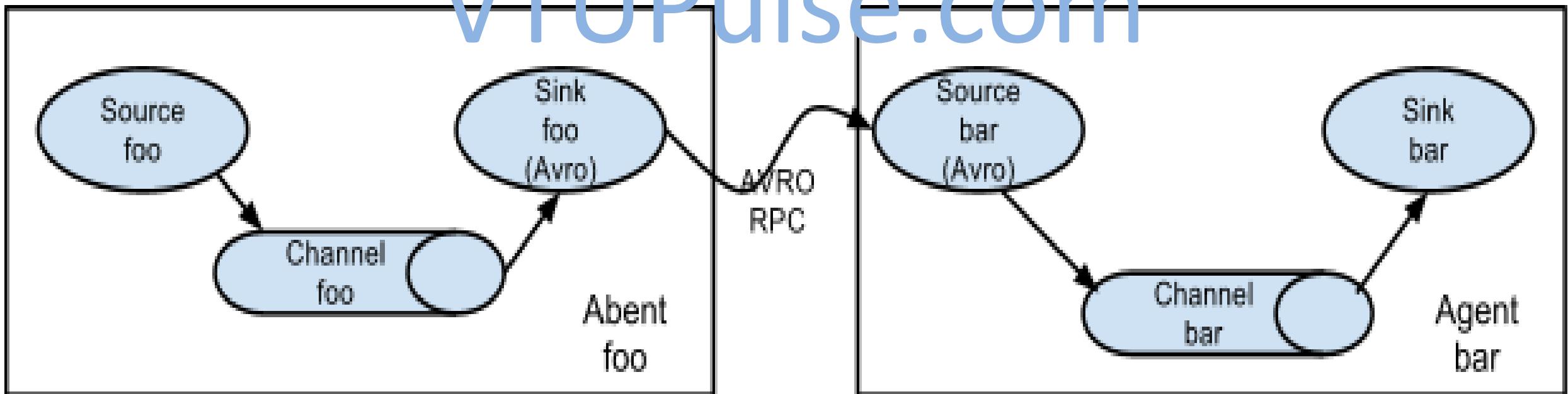
<https://www.youtube.com/c/maheshhuddar>

Setting multi-agent flow

- In order to flow the data across multiple agents or hops, the sink of the previous agent and source of the current agent need to be AVRO type with the sink pointing to the hostname (or IP address) and port of the source.
- Within Flume, there can be multiple agents and before reaching the final destination, an event may travel through more than one agent.
- This is known as **multi-agent flow**.

Setting multi-agent flow

VTUPulse.com



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Consolidation

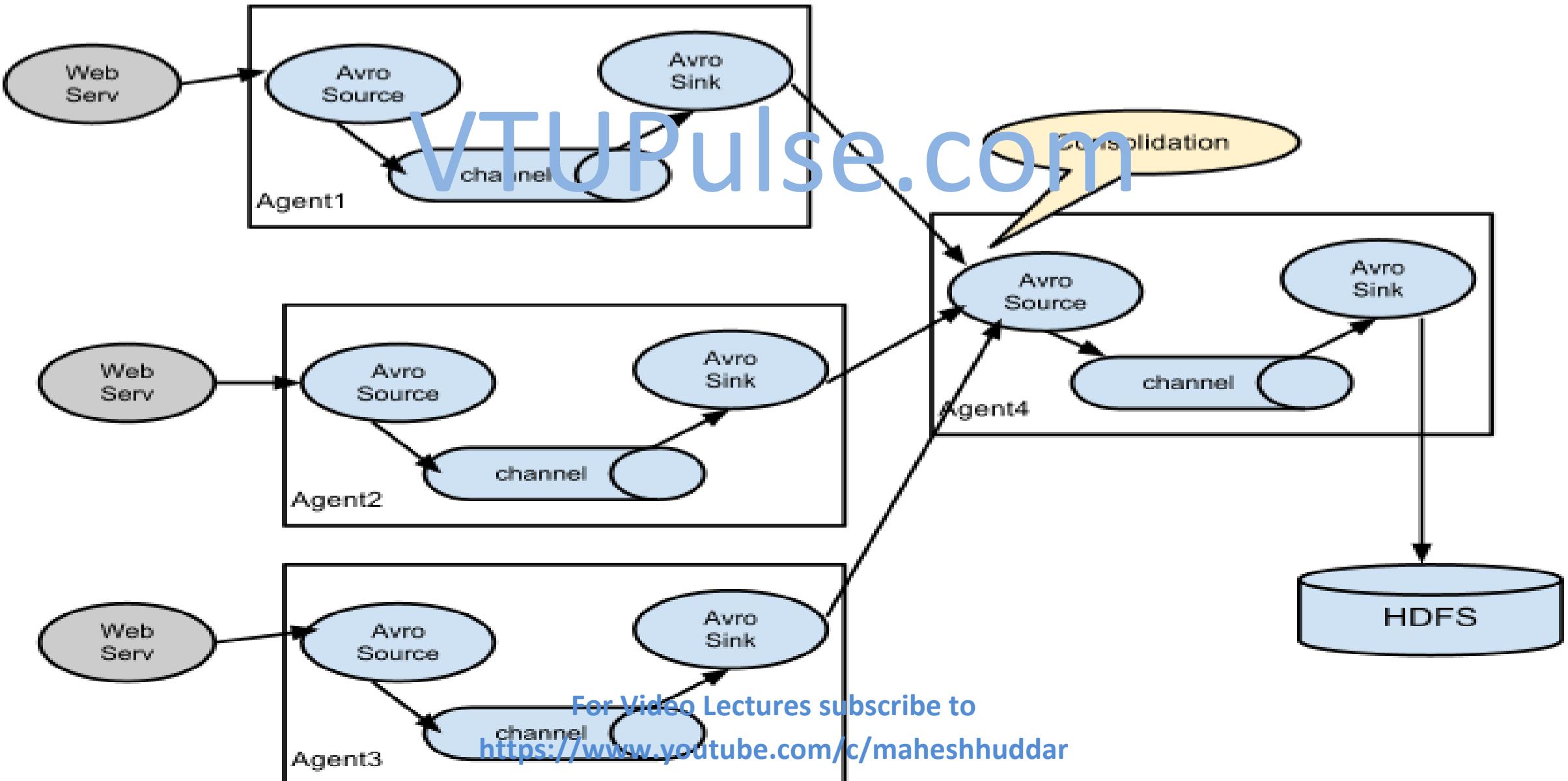
- A very common scenario in log collection is a large number of log producing clients sending data to a few consumer agents that are attached to the storage subsystem.
- For example, logs collected from hundreds of web servers sent to a dozen of agents that write to HDFS cluster.
- This can be achieved in Flume by configuring a number of first tier agents with an avro sink, all pointing to an avro source of single agent .
- This source on the second tier agent consolidates the received events into a single channel which is consumed by a sink to its final destination.

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Consolidation



VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Pig

VTUPulse.com

Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Objectives

- What is Apache Pig..?
- Why Apache Pig is so popular..? 
- Data Types
- Apache Pig usage modes
- Pig Example Walk-Through
- Running Pig Programs
 - Local mode
 - Cluster mode

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Apache Pig

- Apache pig a high-level language that enable the programmers to write complex MapReduce programs using a simple scripting language.
- Pig Latin (actual name) defines a set of transformations on data set such as aggregate, join, and sort.
- Pig is often used to extract, transform and load data pipelines, quick research of raw data and iterative data processing.
- It is an open source platform developed by yahoo.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Why Apache Pig is so popular...?

- Easy to learn read and write and implement if you know SQL.
- It implements a new approach of multi query.
- It consist of different user groups for instance up to 90% of Yahoo's MapReduce is done by Pig and up to 80% of Twitter's MapReduce is also done by Pig and various other companies like Sales force, LinkedIn and Nokia etc. are majoritively using the Pig.
- Provides a large number of nested data types such as Maps, Tuples and Bags which are not easily available in MapReduce along with some other data operations like Filters, Ordering and Joins.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Data Types

The scalar data types in pig are in the form of int, float, double, long, chararray, and byte array. The complex data types in Pig are namely the map, tuple, and bag.

- **Map:** The data element consisting the data type chararray where element has pig data type include complex data type

Example- ['city' # 'bang' , 'pin' # 560001]

- In this city and pin are data element mapping the values here.
- **Tuple:** Collection of data types and it has defined fixed length. It consists of multiple fields and those are ordered in sequence.
- **Bag:** It is a huge collection of tuples, unordered sequence, tuples arranged in the bag are separated by comma.

Example: {('Bangalore', 560001), ('Mysore', 570001), ('Mumbai', 400001)}

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Apache Pig usage modes

- Local Mode
 - All processing is done at a local machine
- Non-local mode (Cluster)
 - Processing is done on a cluster of machines (HDFS and Tez)
- Iterative and batch mode
 - Here the small amount of data is used to test the application at development stage before running the application into large cluster.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Running Pig Programs

There are namely 3 ways of executing Pig programs which works on both local and MapReduce mode:

Script



- Pig can run a script file that contains Pig commands. For example, `pig script.pig` runs the commands in the local file `script.pig`. Alternatively, for very short scripts, you can use the `-e` option to run a script specified as a string on the command line.

Grunt

- Grunt is an interactive shell programming for running Pig commands. Grunt is started when no file is specified for Pig to run, and the `-e` option apparently not used. It is also possible to run Pig scripts from within Grunt using `run` and `exec`.

Embedded

- You can execute all the Pig programs from Java and can use JDBC to run SQL programs from Java.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Pig Example Walk-Through

Installation

1. Download **VTUPulse.com**
<https://pig.apache.org/releases.html>
2. Extract
Extract the zip file into local machine or in cluster directory
3. Set Path
Set the path of directory as home directory of apache pig

Pig Example Walk-Through

Local Mode

Use following command to start pig

pig -x local

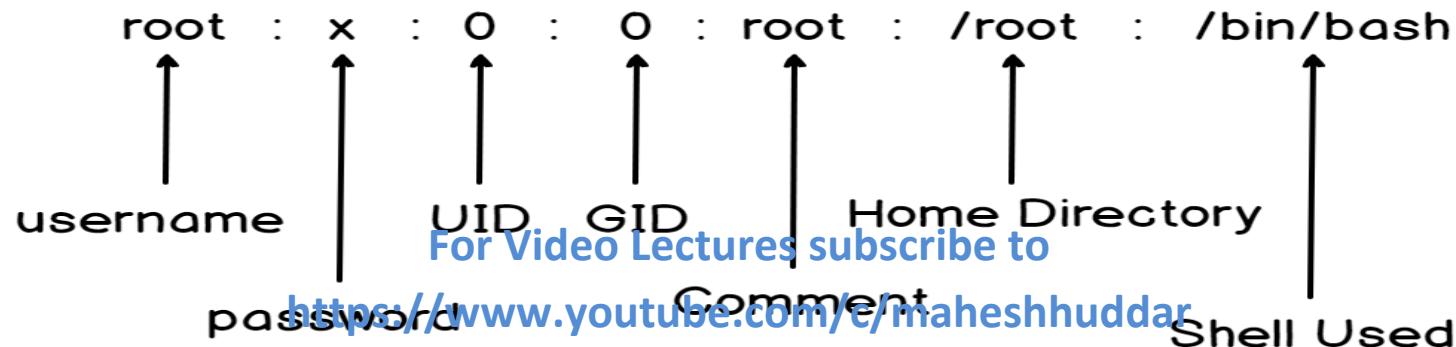
VTUPulse.com

If pig starts correctly then, you will get grunt> prompt.

grunt> A = load 'passwd' using PigStorage(':');

the /etc/passwd file is used for demonstration

/etc/passwd columns



Pig Example Walk-Through

Local Mode

VTUPulse.com

get all the usernames from /etc/passwd file

grunt> B = foreach A generate \$0 as id;

dump the username onto the terminal

grunt> dump B;

to quit the the grunt prompt.

grunt> quit

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Pig Example Walk-Through

Cluster Mode

VTUPulse.com

To use Hadoop MapReduce start the pig as follows

pig -x mapreduce

Write sequence of commands in a file and save file as filename.pig

A = load 'passwd' using PigStorage(':'');

B = foreach A generate \$0 as id;

dump B;

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Pig Example Walk-Through

First Start → Start Hadoop

VTUPulse.com

Store input file in HDFS

```
Hdfs dfs -mkdir /passwdDIR
```

```
hdfs dfs -put passwd passwdDIR
```

Run using command

```
pig -x mapreduce filename.pig
```

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Hive

VTUPulse.com

Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Objectives

- What is Apache Hive..?
- Features of Apache Hive
- Architecture of Hive
- Hive Example Walk-Through

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

What is Apache Hive ?

- Hive is a data warehouse infrastructure tool to process structured data in Hadoop.
- It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.
- Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive.
- It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

VTUPulse.com

For Video Lectures subscribe to

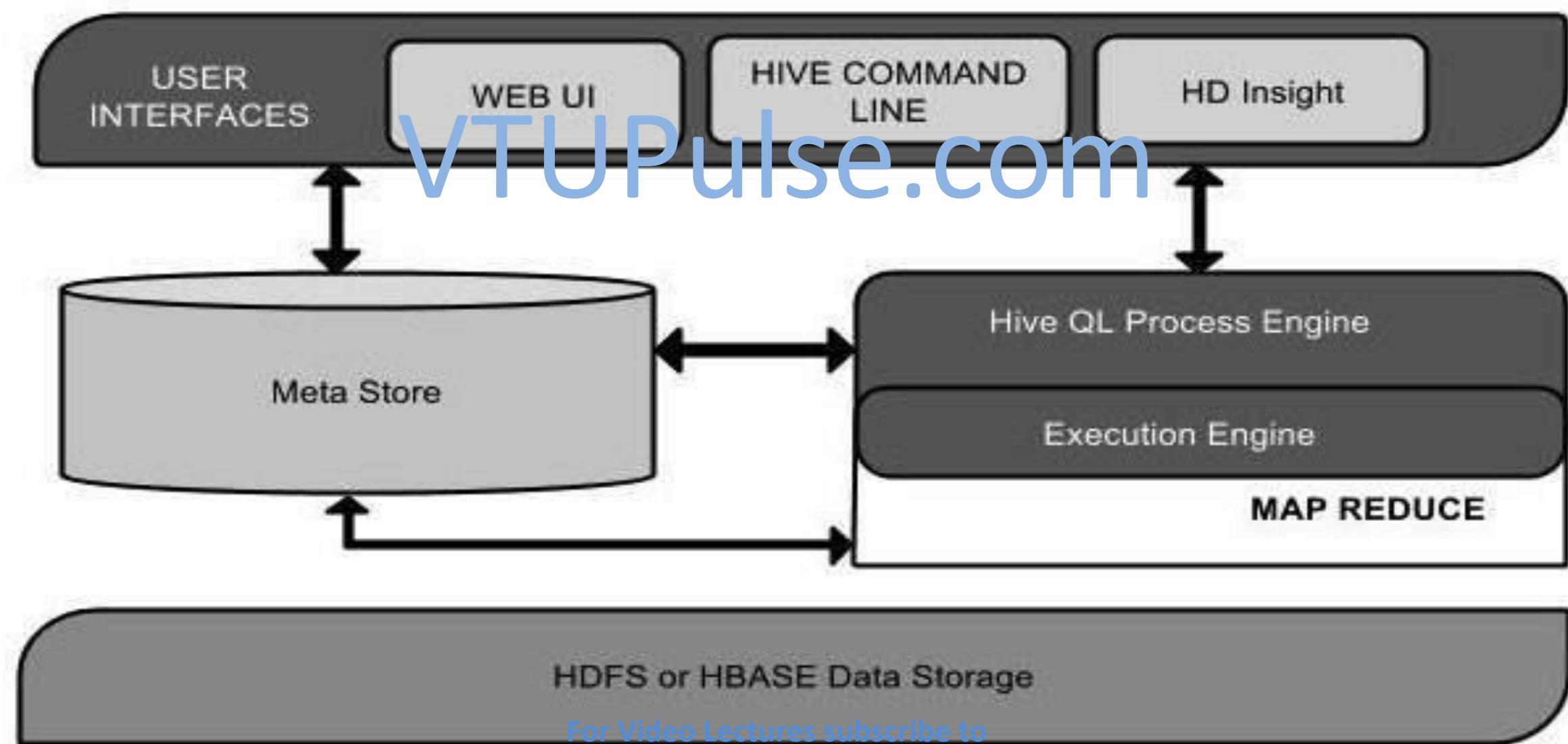
<https://www.youtube.com/c/maheshhuddar>

Features of Hive

Features of Apache Hive

- Tool enable the easy data extraction, transformation and loading (ETL)
- A mechanism to impose structure on various data formats.
- Query execution via MapReduce and Tez, provides SQL type language for querying called HiveQL or HQL.
- Hive is fast, scalable, and extensible.

Architecture of Hive



For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Architecture of Hive

User Interface	Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line and Hive HDInsight (In Windows server).
Meta Store	Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.
HiveQL Process Engine	HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.
Execution Engine	The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.
HDFS or HBASE	Hadoop distributed file system or HBASE are the data storage techniques to store data into file system. <small>For Video Lectures subscribe to https://www.youtube.com/c/maheshhuddar</small>

Hive Example Walk-Through

To start Hive simply enter the command `hive`. If Hive starts correctly then you should get `hive` prompt.

`$hive`

`hive>`

VTUPulse.com

As a simple test we try to create and drop tables.

`hive> CREATE TABLE pokes (foo INT, bar string);`

`hive> SHOW TABLES;`

`Pokes`

`hive> DROP TABLE pokes;`

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hive Example Walk-Through

Another example

- hive> CREATE DATABASE IF NOT EXISTS userdb;
- hive> SHOW DATABASES;
- hive> DROP DATABASE IF EXISTS userdb;
- hive> DROP SCHEMA userdb;
- hive> CREATE TABLE IF NOT EXISTS employee (eid int, name String, salary String, destination String) COMMENT 'Employee details' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
- hive> SELCT eid, name from employee where salary > 40000;
- hive> ALTER TABLE employee RENAME TO emp;
- hive> DROP TABLE IF EXISTS employee;

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache Oozie

VTUPulse.com

Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Objectives

- What is Apache Oozie..?
- Types of nodes in Oozie Workflow
- Example Workflow Diagram
- Complex Example Workflow Diagram

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

What is Apache Oozie..?

- Apache Oozie is a workflow director designed to run and manage multiple related Apache Jobs.
- For instance, complete data input and analysis may require several related Hadoop jobs to be run as a workflow in which the output of one job may input to the successive job.
- Oozie is designed to construct and manage these workflows.
- Ozzie is not a substitute for YARN scheduler.
- YARN manages the resources of individual jobs and Oozie provides a mechanism to connect and control Hadoop jobs on the HDFS cluster.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

What is Apache Oozie..?

Oozie workflow jobs are represented using **Directed Acyclic Graphs**.

VTUPulse.com

Oozie Supports three types of Jobs:

- **Workflow engine:** Responsibility of a workflow engine is to store and run workflows composed of Hadoop jobs e.g., MapReduce, Pig, Hive.
- **Coordinator engine:** It runs workflow jobs based on predefined schedules and availability of data.
- **Bundle:** Higher level abstraction that will batch a set of coordinator jobs

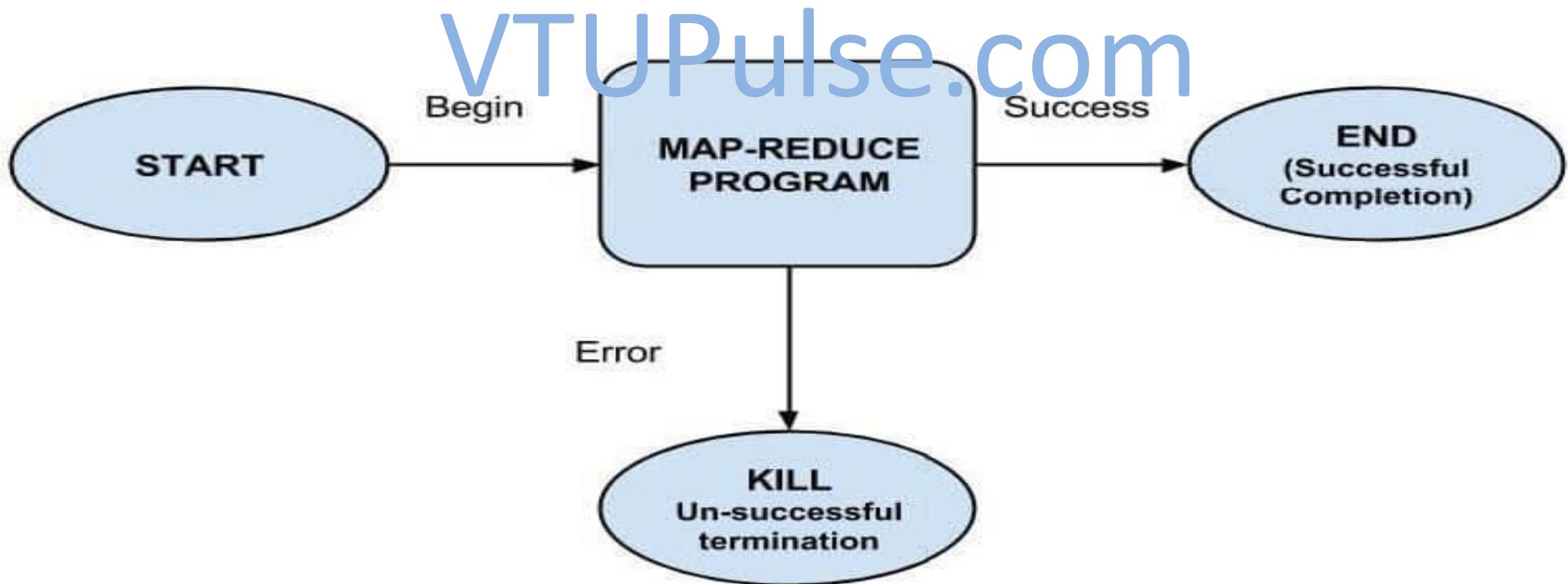
For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Types of nodes in Oozie Workflow

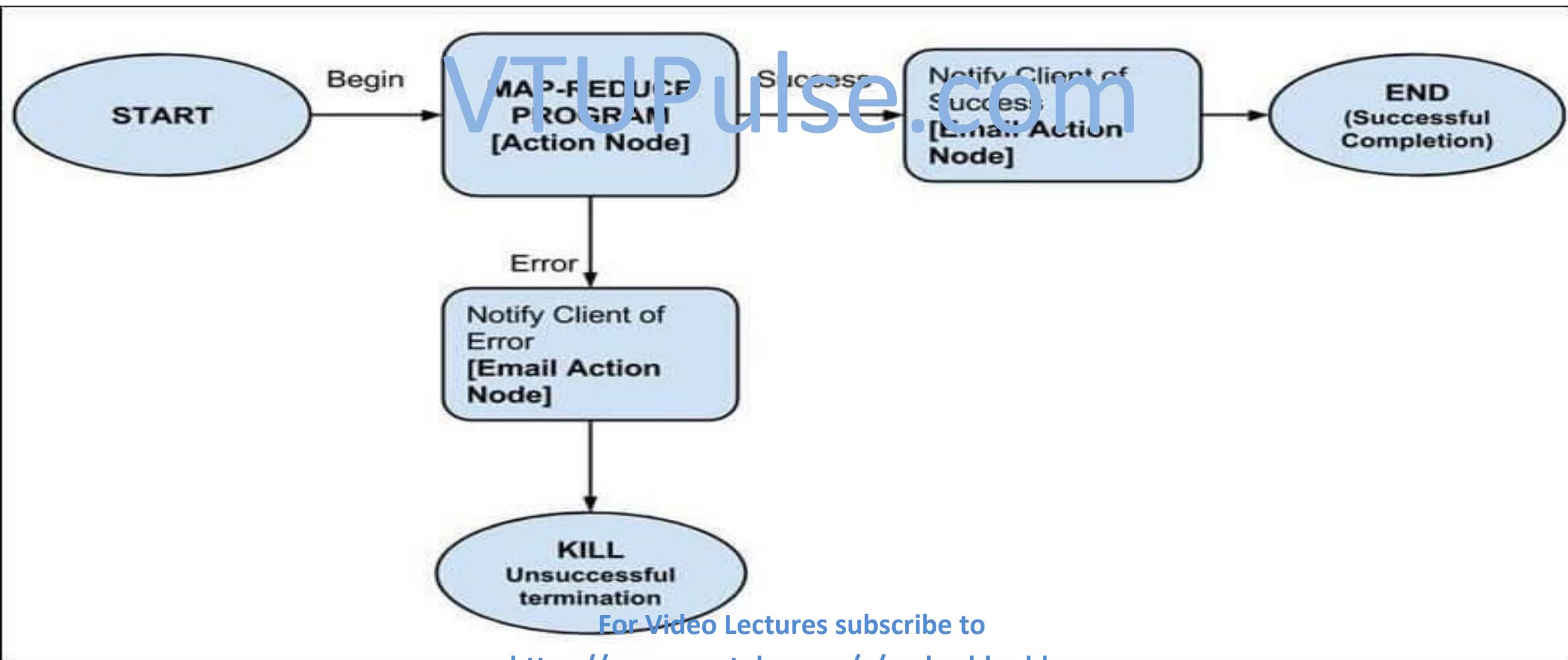
- **Control Flow nodes:** define the beginning and the end of the workflow. They include start, end and optional fail nodes.
- **Action Nodes:** are where the actual processing tasks are defined. When an action node finishes, the remote systems notify Oozie and the next node in the workflow is executed. Action nodes can also include HDFS commands.
- **Fork/Join nodes:** enable parallel execution of tasks in the workflow. The fork node enables two or more tasks to run at the same time. A join node represents a rendezvous point that must wait until all forked tasks complete.
- **Control flow nodes:** enable decisions to be made about the previous task. Control decisions are based on the results of the previous actions. Decision nodes are essentially switch-case statements that use JSP EL (Java Server Pages – Expression Language) that evaluate to either true or false.

Example Workflow Diagram



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

More Complex Example Workflow Diagram



VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Apache HBase

VTUPulse.com

Hadoop Ecosystem

Mahesh G Huddar

Asst. Professor

Dept. of CSE, HIT, Nidasoshi

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Objectives

- What is Apache HBase..?
- Features of Apache Hbase
- Limitations of Hadoop
- Hbase Data Model Overview
- Hbase Installation
- Hbase Example Walk-Through
 - Create Database and insert data
 - Inspect Database
 - Get a row
 - Get table cell
 - Delete Cell, Row and Table [For Video Lectures subscribe to](#)
<https://www.youtube.com/c/maheshhuddar>
 - Scripting

What is Apache Hbase..?

- HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable.
- HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS).
- It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.
- One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

Features of Apache Hbase

Some of the more important features include the following capabilities:

- Linear and modular capabilities
- Strictly consistent reads and writes
- Automatics and configurable sharding of tables
- Automatics failover support between RegionServers
- Convenient base classes for backing Hadoop MapReduce jobs with Apache Hbase tables
- Easy-to-use Java API for client access

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Limitations of Hadoop

- Hadoop can perform only batch processing, and data will be accessed only in a sequential manner. That means one has to search the entire dataset even for the simplest of jobs.
- A huge dataset when processed results in another huge data set, which should also be processed sequentially. At this point, a new solution is needed to access any point of data in a single unit of time (random access).
- Applications such as HBase, Cassandra, couchDB, Dynamo, and MongoDB are some of the databases that store huge amounts of data and access the data in a random manner.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Hbase Data Model Overview

- A table in Hbase is similar to other databases, having rows and columns. Column in Hbase are grouped into column families, all with the same prefix.
- For example, consider a table of daily stock prices.
- There may be a column family called “price” that has four members – price:open, price:cose, price:low and price:high
- A column does not need to be a family.
- For instance, the stock table may have a column named ‘volume’ indicating how many shares were traded.
- All column family members are stores together in the physical file system.

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Hbase Data Model Overview

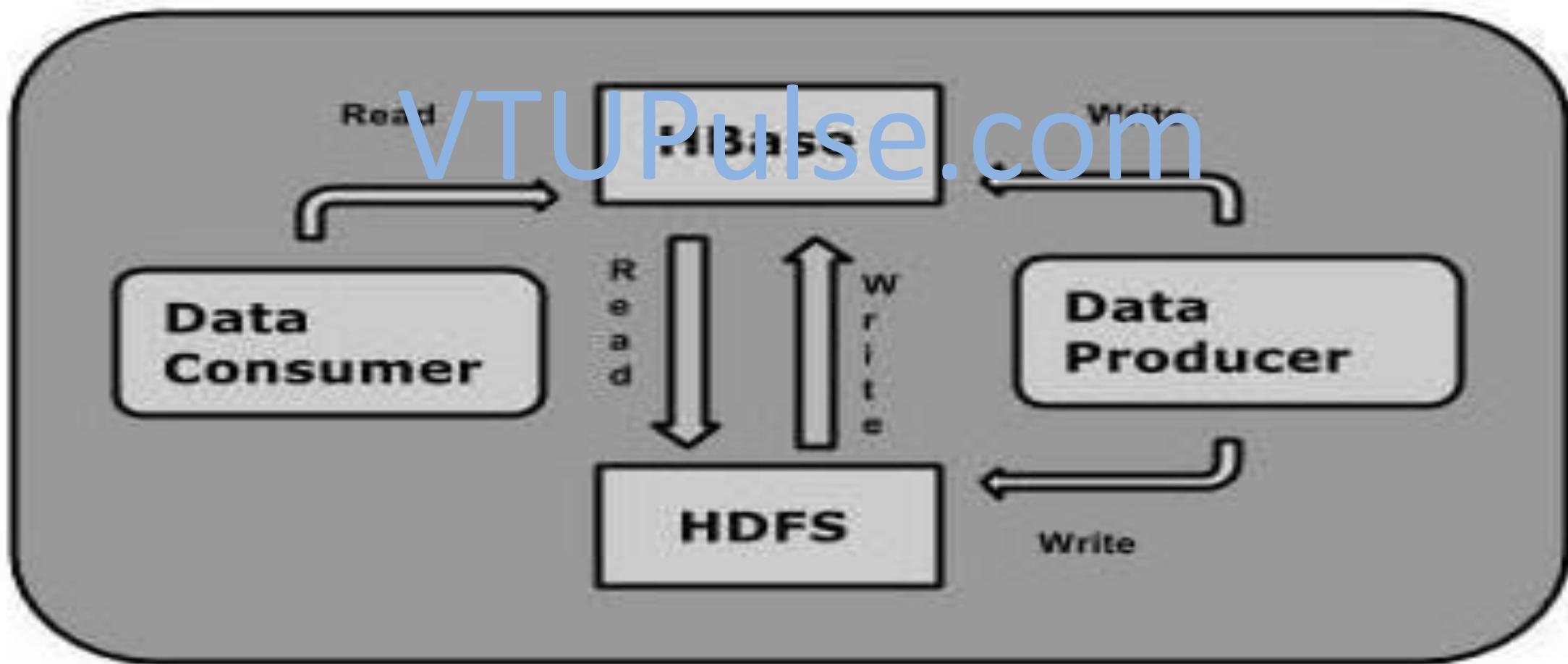
- Specific Hbase cell values are identified by row key, column (column family and column), version (timestamp).
- It is possible to have many versions of data within an Hbase cell.
- A version is specified as a timestamp and is created each time data are written to a cell.
- Almost anything can serve as a row key, from strings to binary representations of longs to serialized data structures.
- Rows are lexicographically sorted with the lowest order appearing first in a table.
- The empty byte array denotes both the start and the end of a tables namespace.
- All table accesses are via the table row key, which is considered its primary key.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

What is HBase?



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

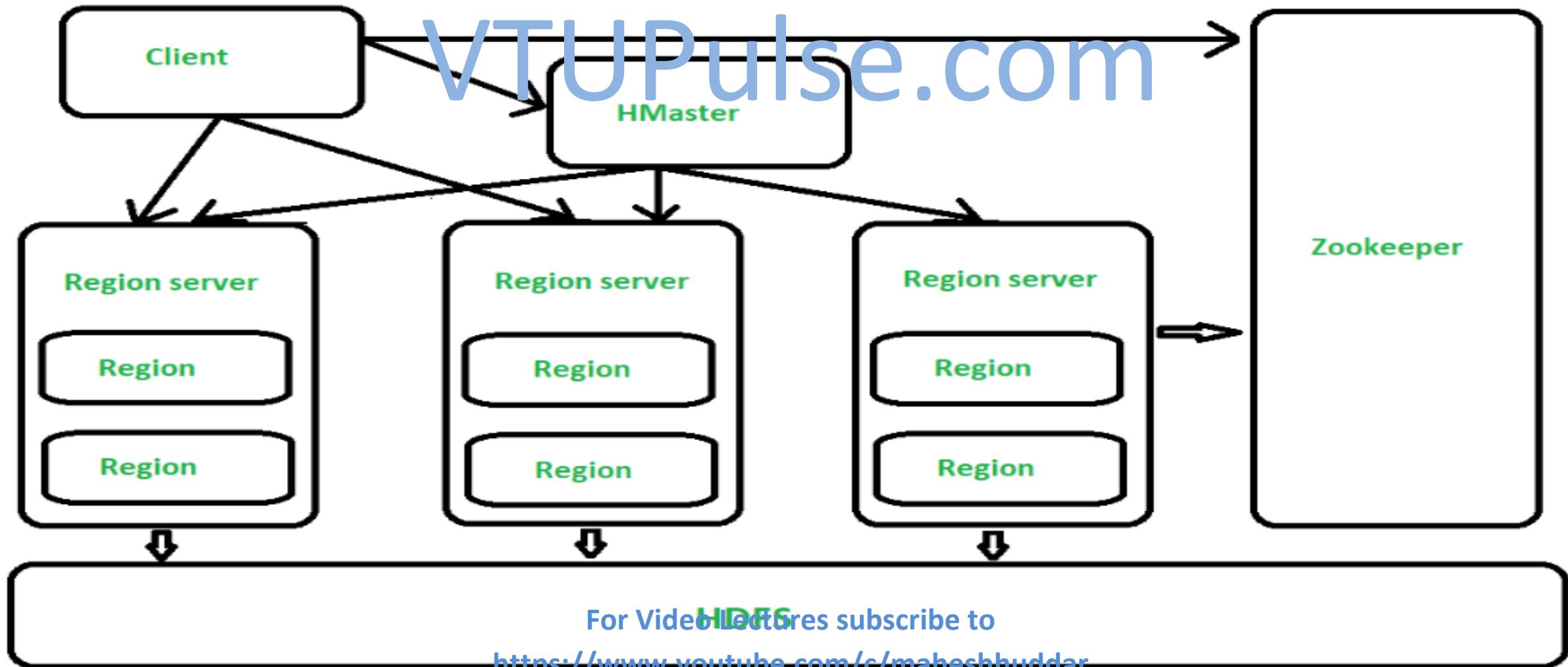
Storage Mechanism in HBase

VTUPulse.com

Row key	personal data		professional data	
	name	city	designation	salary
empid				
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hbase Architecture



Hbase Installation

- Download
- <http://archive.apache.org/dist/hbase/0.98.24/>
- Extract
- sudo tar -zxvf hbase-0.98.24-hadoop2-bin.tar.gz
- Move
- sudo mv hbase-0.98.24-hadoop2 /usr/local/Hbase
- cd /usr/local/Hbase/
- sudo gedit hbase-env.sh
 - export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
- cd conf/
- sudo gedit hbase-site.xml
- cd /usr/local/Hbase/bin
- Edit ~/.bashrc
- ./start-hbase.sh

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hbase Installation

- <property>
- <name>hbase.rootdir</name>
- <value>file:/usr/local/hadoop/HBase/HFiles</value>
- </property>

- //Here you have to set the path where you want HBase to store its built in zookeeper files.
- <property>
- <name>hbase.zookeeper.property.dataDir</name>
- <value>/usr/local/hadoop/zookeeper</value>
- </property>

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hbase Example Walk-Through

- To enter the hbase shell use the following command
 - **hbase shell**
- To exit the shell you can type exit
 - **hbase> exit**
- To know the system status use status command which gives the number of servers running and dead.
 - **hbase> status**
 - **4 servers, 0 dead, 1.000 average load**

Hbase Example Walk-Through

- Create Database and insert data
 - create 'apple', 'price', 'volume'
 - In this case table name is apple and two columns are defined.
 - The price column is the family of four values (open, close, low and high)
 - put command is used to add data to the database within the shell.
 - put 'apple', '17-April-19', 'price:open', '125'
 - put 'apple', '17-April-19', 'price:high', '126'
 - put 'apple', '17-April-19', 'price:low', '124'
 - put 'apple', '17-April-19', 'price:close', '125.5'
 - put 'apple', '17-April-19', 'volume', '1000'

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hbase Example Walk-Through

Inspect Database

- The entire database can be listed using the scan command.
- **scan 'apple'**
- ROW COLUMN+CELL
- 17-April-19 column=price:close, timestamp=1555508855040, value=122.5
- 17-April-19 column=price:high, timestamp=1555508840180, value=126
- 17-April-19 column=price:low, timestamp=1555508846589, value=124
- 17-April-19 column=price:open, timestamp=1555508823773, value=125
- 17-April-19 column=volume:, timestamp=1555508892705, value=1000

VTUPulse.com

Hbase Example Walk-Through

Get a row

- You can use row key to access the individual rows.
- `get 'apple', '17-April-19'`
- COLUMN CELL
- `price:close` `timestamp=1555508855040, value=122.5`
- `price:high` `timestamp=1555508840180, value=126`
- `price:low` `timestamp=1555508846589, value=124`
- `price:open` `timestamp=1555508823773, value=125`
- `volume:` `timestamp=1555508892705, value=1000`

Hbase Example Walk-Through

Get table Cell

VTUPulse.com

- get 'apple', '17-April-19', {COLUMN => price:low'}
- COLUMN CELL
- price:low timestamp=1555508846589, value=124

- get 'apple', '17-April-19', {COLUMN => ['price:low', 'price:close']}
- COLUMN CELL
- price:close timestamp=1555508855040, value=122.5
For Video Lectures subscribe to
<https://www.youtube.com/c/maheshkuddar>
- price:low timestamp=1555508846589, value=124

Hbase Example Walk-Through

Get table Cell

VTUPulse.com

- get 'apple', '17-April-19', { COLUMN => ['volume', 'price:low'] }
 - COLUMN CELL
 - price:low timestamp=1555508846589, value=124
 - volume: timestamp=1555508892705, value=1000

Hbase Example Walk-Through

Delete Cell, Row and Table

VTUPulse.com

- delete 'apple', '17-April-19', 'price.now'
- deleteall 'apple', '17-April-19'
- disable 'apple'
- drop 'apple'

Hbase Example Walk-Through

Scripting

VTUPulse.com

- echo "create 'apple', 'price', 'volume'"
- echo "create 'apple', 'price', 'volume'" | hbase shell
- Create test.sh file with contents
- echo "create 'mango', 'price', 'volume'"
- echo "put 'mango', '123', 'price', '100'"
- Then run the following commands
- sudo sh test.sh | hbase shell

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Why YARN?

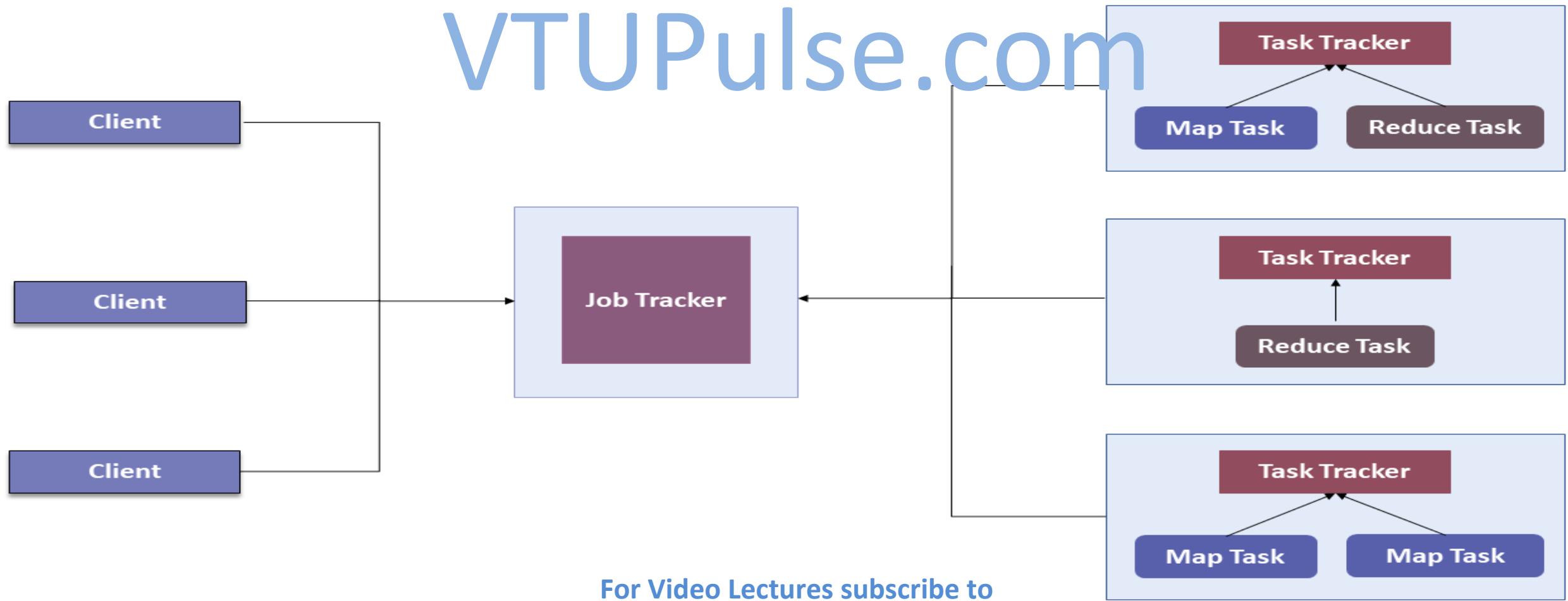
- In Hadoop version 1.0 which is also referred to as MRV1(MapReduce Version 1), MapReduce performed both processing and resource management functions.
- It consisted of a Job Tracker which was the single master.
- The Job Tracker allocated the resources, performed scheduling and monitored the processing jobs.
- It assigned map and reduce tasks on a number of subordinate processes called the Task Trackers.
- The Task Trackers periodically reported their progress to the Job Tracker.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Why YARN?

VTUPulse.com



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Why YARN?

- This design resulted in scalability bottleneck due to a single Job Tracker. IBM mentioned in its article that according to Yahoo!, the practical limits of such a design are reached with a cluster of 5000 nodes and 40,000 tasks running concurrently.
- Apart from this limitation, the utilization of computational resources is inefficient in MRV1. Also, the Hadoop framework became limited only to MapReduce processing paradigm.

Why YARN?

- To overcome all those issues, YARN was introduced in Hadoop version 2.0 in the year 2012 by Yahoo and Hortonworks.
- The basic idea behind YARN is to relieve MapReduce by taking over the responsibility of Resource Management and Job Scheduling.
- YARN started to give Hadoop the ability to run non-MapReduce jobs within the Hadoop framework.

Why YARN ?

- MapReduce is a powerful distributed framework and **programming model** that allows batch-based parallelized work to be performed on a cluster of multiple nodes.
- Despite being very efficient at what it does, though, MapReduce has some disadvantages; principally that it's **batch-based**, and as a result isn't suited to real-time or even near-real-time data processing.
- Historically this has meant that processing models such as graph, iterative, and real-time data processing are not a natural fit for **MapReduce**.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Introduction to Hadoop YARN



Hadoop v1.0

MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage

VTUPulse.com



Hadoop v2.0

MapReduce

**Other Data
Processing
Frameworks**

YARN

Resource Management

HDFS

Distributed File Storage

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Components of YARN

Apart from Resource Management, YARN also performs Job Scheduling. YARN performs all your processing activities by allocating resources and scheduling tasks.

VTUPulse.com

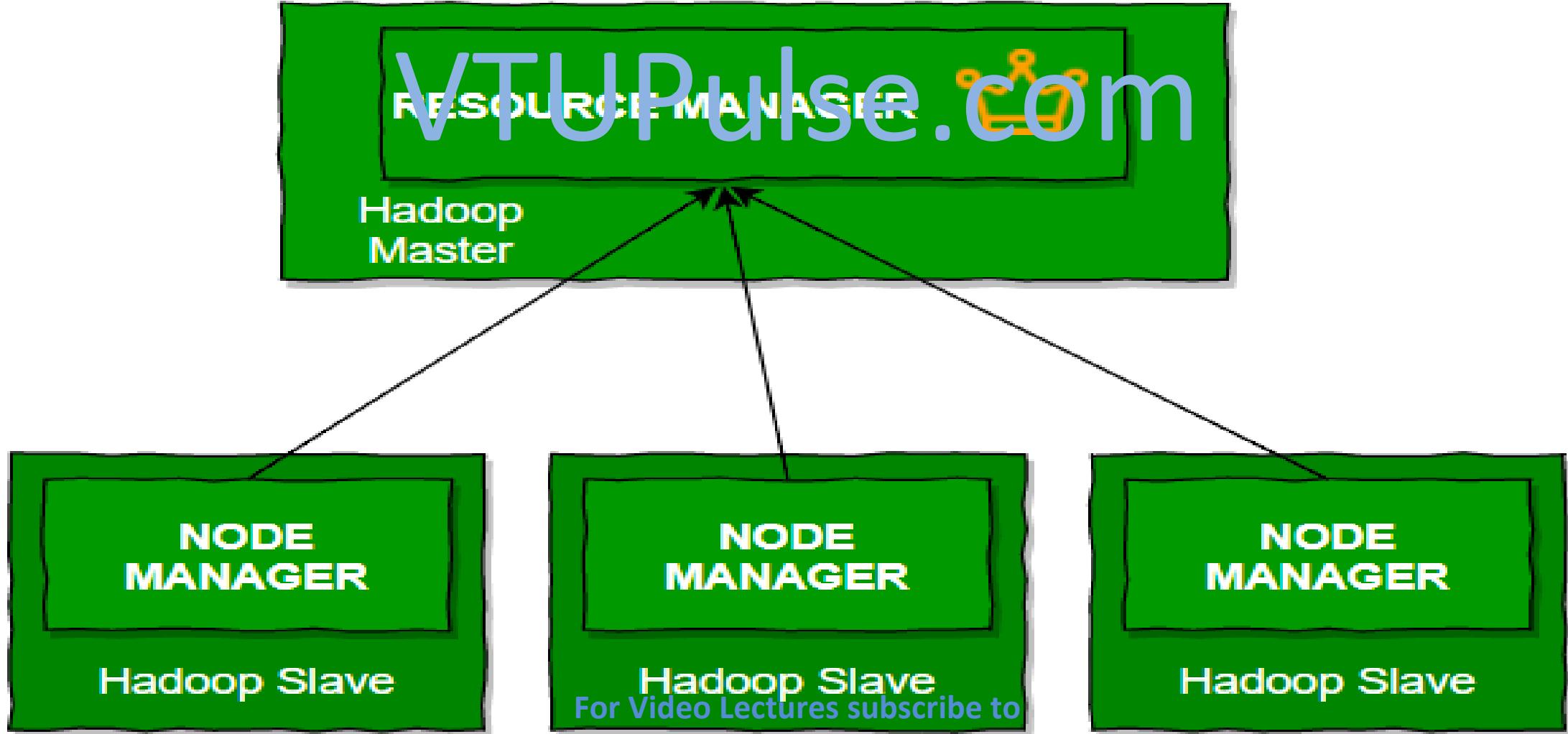
Apache Hadoop YARN Architecture consists of the following main components :

- **Resource Manager:** Runs on a master daemon and manages the resource allocation in the cluster.
- **Node Manager:** They run on the slave daemons and are responsible for the execution of a task on every single Data Node.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Components of YARN



Components of YARN - ResourceManager

- The ResourceManager is the YARN master process.
- A Hadoop cluster has a single Resource Manager (RM) for the entire cluster. Its sole function is to arbitrate all the available resources on a Hadoop cluster.
- ResourceManager tracks usage of resources, monitors the health of various nodes in the cluster, enforces resource-allocation invariants, and arbitrates conflicts among users.
- The components of resource manager
 - Scheduler
 - ApplicationsManager

Components of YARN - NodeManager

- The NodeManager is the slave process of YARN.
- It runs on every data node in a cluster.
- Its job is to create, monitor, and kill containers.
- It services requests from the ResourceManager and ApplicationMaster to create containers, and it reports on the status of the containers to the ResourceManager. The ResourceManager uses the data contained in these status messages to make scheduling decisions for new container requests.
- On start-up, the NodeManager registers with the ResourceManager; it then sends heartbeats with its status and waits for instructions. Its primary goal is to manage application containers assigned to it by the ResourceManager.

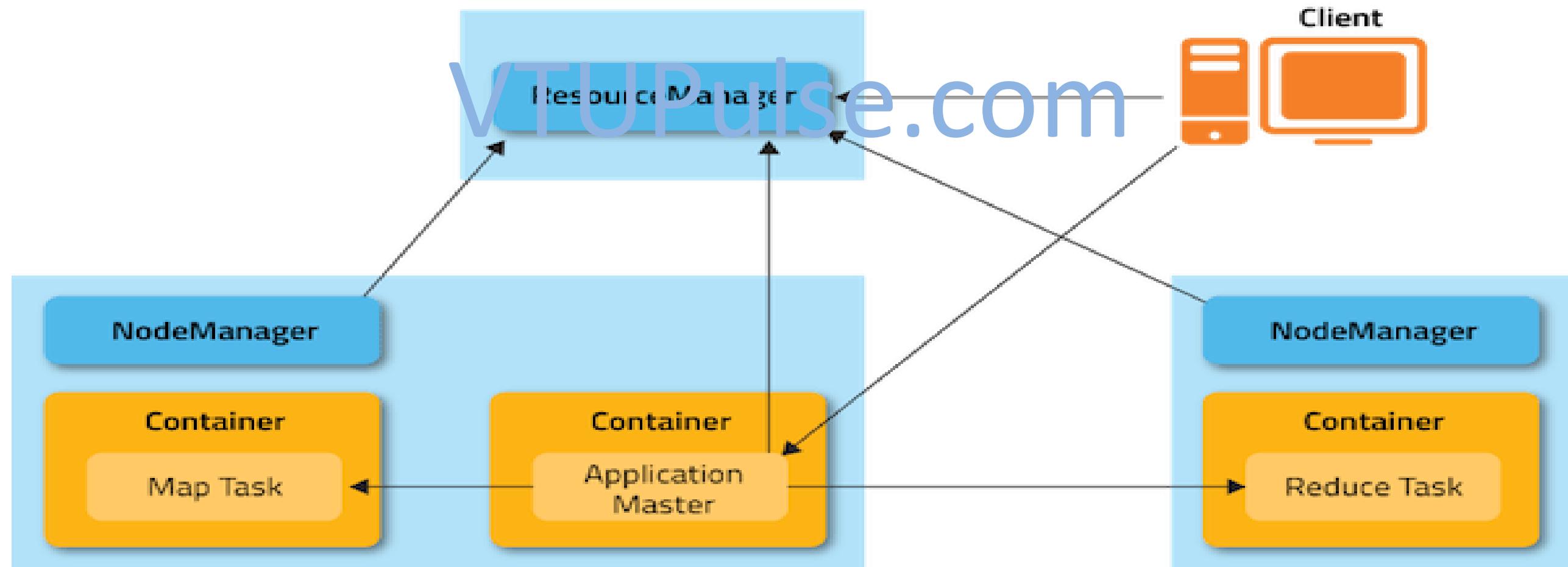
VTUPulse.com

YARN Applications

- The YARN framework/platform exists to manage applications, so let's take a look at what components a YARN application is composed of.
- A YARN application implements a specific function that runs on Hadoop. A YARN application involves 3 components:
 - Client
 - ApplicationMaster(AM)
 - Container

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

YARN Applications



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

YARN Applications - YARN Client

- Launching a new YARN application starts with a YARN client communicating with the ResourceManager to create a new YARN ApplicationMaster instance.
- Part of this process involves the YARN client informing the ResourceManager of the ApplicationMaster's physical resource requirements.

YARN Applications - YARN ApplicationMaster

- The ApplicationMaster is the master process of a YARN application.
- It doesn't perform any application-specific work, as these functions are delegated to the containers. Instead, it's responsible for managing the application-specific containers.
- Once the ApplicationMaster is started (as a container), it will periodically send heartbeats to the ResourceManager to affirm its health and to update the record of its resource demands.

YARN Applications - YARN Container

- A container is an application-specific process that's created by a NodeManager on behalf of an ApplicationMaster.
- At the fundamental level, a container is a collection of physical resources such as RAM, CPU cores, and disks on a single node.

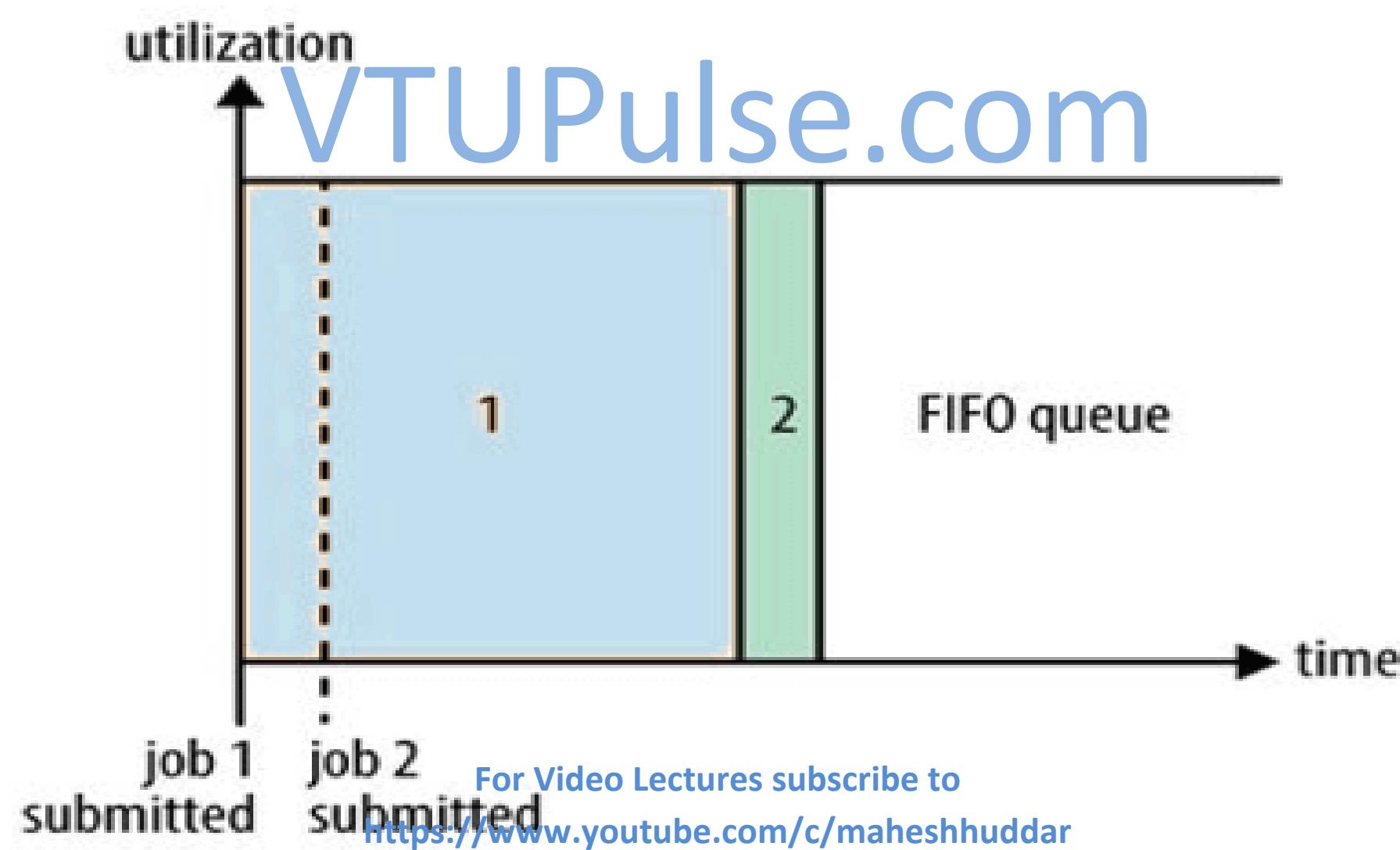
VTUPulse.com

YARN scheduler policies

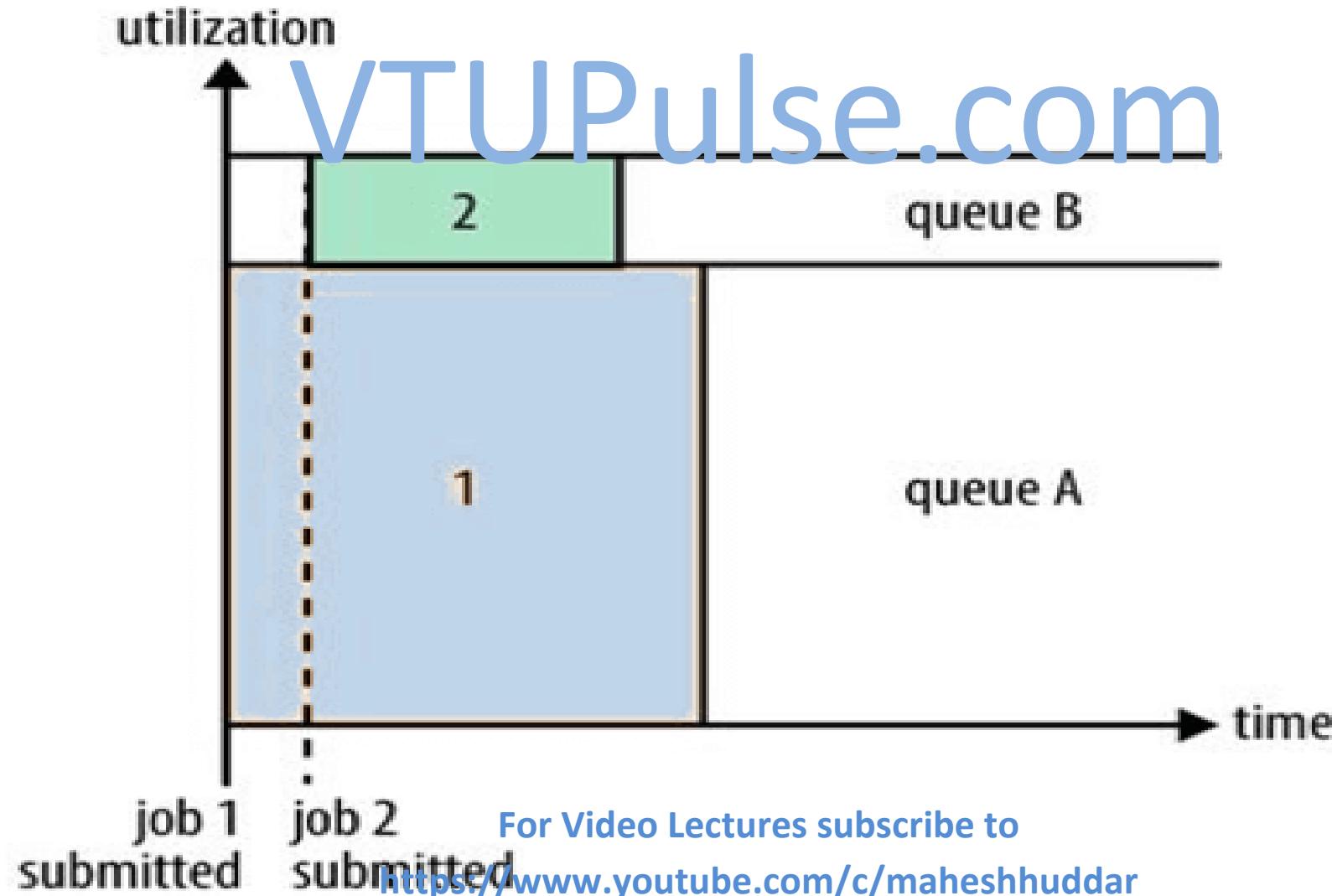
- In an ideal world, the requests that a YARN application makes would be granted immediately.
- In the real world, however, resources are limited, and on a busy cluster, an application will often need to wait to have some of its requests fulfilled.
- The FIFO scheduler
- The Capacity scheduler
- The Fair scheduler

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

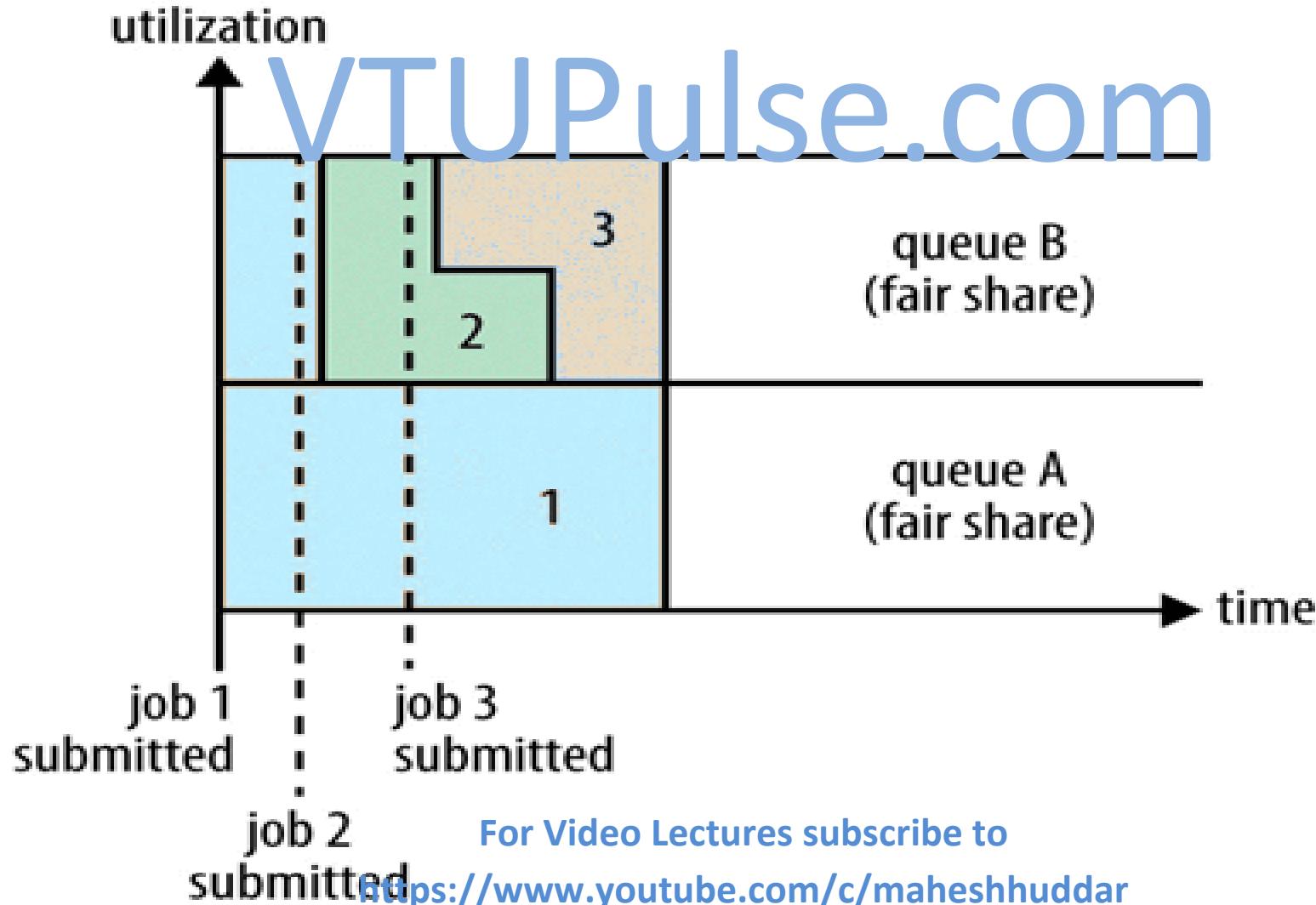
YARN scheduler policies - The FIFO scheduler



YARN scheduler policies - The Capacity scheduler



YARN scheduler policies - The Fair scheduler



Apache Ambari

- Managing a Hadoop installation by hand can be tedious and time consuming. In addition to keeping configuration files synchronized across a cluster, starting, stopping, and restarting Hadoop services and dependent services in the right order is not a simple task.
- The Apache Ambari graphical management tool is designed to help you easily manage these and other Hadoop administrative issues. This chapter provides some basic navigation and usage scenarios for Apache Ambari

Apache Ambari

- Along with being an installation tool, Ambari can be used as a centralized point of administration for a Hadoop cluster. Using Ambari, the user can configure cluster services, monitor the status of cluster hosts (nodes) or services, visualize hotspots by service metric, start or stop services, and add new hosts to the cluster.
- All of these features infuse a high level of agility into the processes of managing and monitoring a distributed computing environment. Ambari also attempts to provide real-time reporting of important metrics.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Dashboard View

- Moving: Click and hold a widget while it is moved about the grid.
- Edit: Place the mouse on the widget and click the gray edit symbol in the upper-right corner of the widget. You can change several different aspects (including thresholds) of the widget.
- Remove: Place the mouse on the widget and click the X in the upper-left corner.
- Add: Click the small triangle next to the Metrics tab and select Add. The available widgets will be displayed. Select the widgets you want to add and click Apply.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>



Ambari

hadoop2

0 rpm

Dashboard

- HDFS
- MapReduce2
- YARN
- Tez
- Nagios
- Ganglia
- Hive
- HBase
- Pig
- Sqoop
- Oozie
- ZooKeeper
- Flume

[Metrics](#) [Heatmaps](#) [Config History](#)

HDFS Disk Usage



CPU Usage



DataNodes Live

4/4

Cluster Load



HDFS Links

 NameNode
 Secondary NameNode
 4 DataNodes
[More...](#)

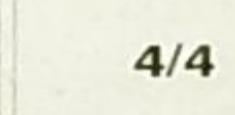
NameNode Heap



HBase Links

 HBase Master
 4 RegionServers
 Master Web UI
[More...](#)

ResourceManager Heap



ResourceManager Uptime

18.2 d

Memory Usage



NameNode RPC

0.08 ms

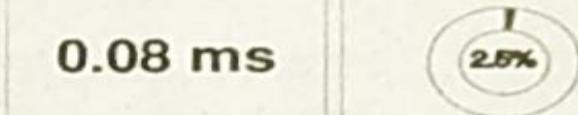
HBase Ave Load

1.33

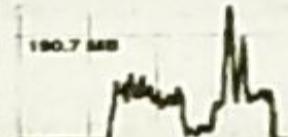
YARN Memory



NameNode CPU WIO



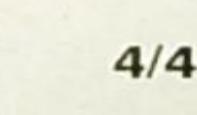
Network Usage



HBase Master Uptime

52.0 d

Flume Live



NameNode Uptime

52.0 d

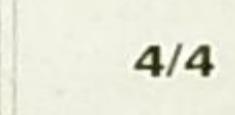
HBase Master Heap



HBase Links

 HBase Master
 4 RegionServers
 Master Web UI
[More...](#)

NodeManagers Live



CPU Usage

X

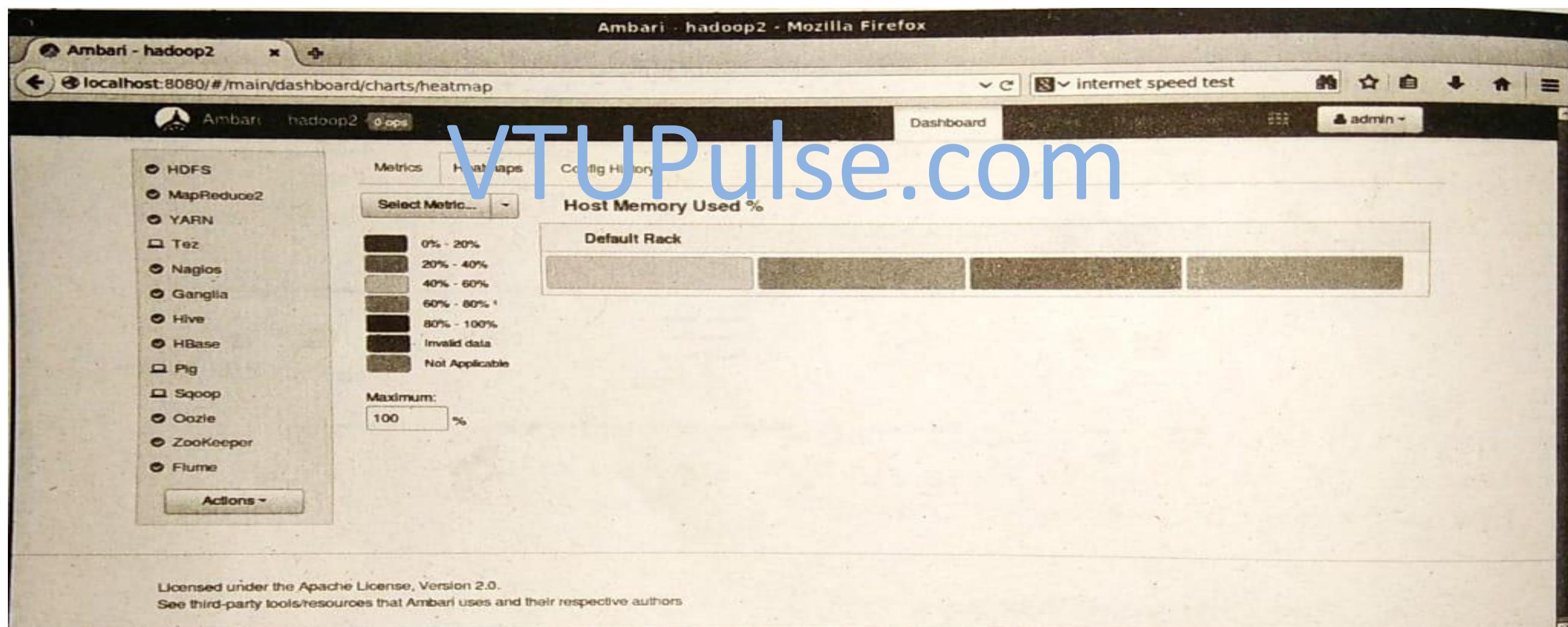
Last 1 hour



	min	avg	max
Idle	0.35%	75.156%	96.31%
Wait	0.675%	4.582%	18.472%
User	2.175%	15.278%	84.25%
System	1.22%	4.981%	16.092%
Nice	0 %	0 %	0 %

OK

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Ambari - hadoop2

localhost:8080/#/main/dashboard/config_history

Ambari hadoop2 0 ops

Metrics Healthchecks Config History

Dashboard Services Help Admin admin

Service: All Config: All Created: Any Author: Any Notes: Any

YARN YARN Default Current Thu, May 28, 2015 12:39 admin No notes

YARN YARN Default Thu, May 28, 2015 12:23 admin No notes

YARN YARN Default Thu, May 28, 2015 12:16 admin No notes

YARN YARN Default Thu, May 28, 2015 12:07 admin No notes

YARN YARN Default Thu, May 28, 2015 11:19 admin No notes

YARN YARN Default Thu, May 28, 2015 08:28 admin No notes

YARN YARN Default Wed, May 27, 2015 19:16 admin No notes

YARN YARN Default Tue, May 26, 2015 15:02 admin Created from service config version V1

YARN YARN Default Tue, May 26, 2015 14:02 admin No notes

YARN YARN Default Tue, May 26, 2015 13:50 admin No notes

23 of 23 versions showing - clear filters

Show: 10 1 - 10 of 23

VTUPulse.com

Version	Service	Config Type	Created	Author	Notes
V11	YARN	YARN Default	Thu, May 28, 2015 12:39	admin	No notes
V10	YARN	YARN Default	Thu, May 28, 2015 12:23	admin	No notes
V9	YARN	YARN Default	Thu, May 28, 2015 12:16	admin	No notes
V8	YARN	YARN Default	Thu, May 28, 2015 12:07	admin	No notes
V7	YARN	YARN Default	Thu, May 28, 2015 11:19	admin	No notes
V6	YARN	YARN Default	Thu, May 28, 2015 08:28	admin	No notes
V5	YARN	YARN Default	Wed, May 27, 2015 19:16	admin	No notes
V4	YARN	YARN Default	Tue, May 26, 2015 15:02	admin	Created from service config version V1
V3	YARN	YARN Default	Tue, May 26, 2015 14:02	admin	No notes
V2	YARN	YARN Default	Tue, May 26, 2015 13:50	admin	No notes

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Services View

- The Services menu provides a detailed look at each service running on the cluster. It also provides a graphical method for configuring each service (i.e., instead of hand editing the /etc/hadoop/conf XML files).
- The summary tab provides a current Summary view of important service metrics and an Alerts and Health Checks sub-window.
- Similar to the Dashboard view, the currently installed services are listed on the leftside menu. To select a service, click the service name in the menu. When applicable, each service will have its own Summary, Alerts and Health Monitoring, and Service Metrics windows.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>



Ambari

hadoop2

4 op/s

Services

Logs

Metrics

Help

?

Logout

admin

- HDFS**
 - MapReduce2
 - YARN
 - Tez
 - Nagios
 - Ganglia
 - Hive
 - HBase
 - Pig
 - Sqoop
 - Oozie
 - ZooKeeper
 - Flume
- Actions**

Summary

Configs

Quick Links

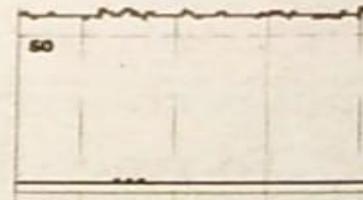
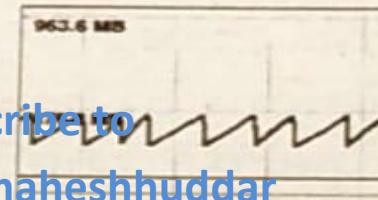
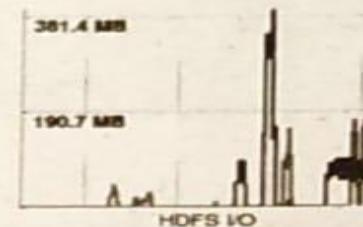
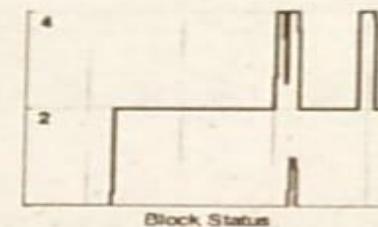
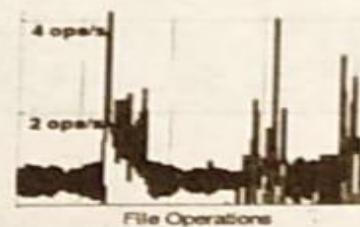
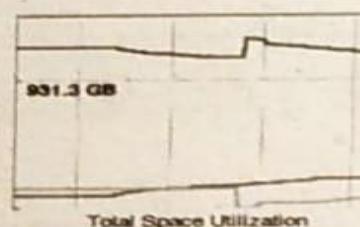
Service Actions

Summary

NameNode Started
SNameNode Started
DataNodes 4/4 DataNodes Alive
NameNode Uptime 51.5 days
NameNode Health 317.4 MB / 1004.0 MB (0.3% used)
DataNodes Status 4 live / 0 dead / 0 decommissioning
Disk Usage (DFS Used) 78.9 GB / 1.4 TB (5.64%)
Disk Usage (Non DFS Used) 211.7 GB / 1.4 TB (15.12%)
Disk Usage (Remaining) 1.1 TB / 1.4 TB (79.25%)
Blocks (total) 1350
Block Errors 0 corrupt / 0 missing / 2 under replicated
Total Files + Directories 300622
Upgrade Status No pending upgrade
Safe Mode Status Not in safe mode

Alerts and Health Checks

- Percent DataNodes with space available OK for 2 months OK: total<4>, affected:<0>
- Percent DataNodes live OK for 2 months OK: total<4>, affected:<0>
- Security NameNode process OK for 2 months TCP OK - 0.001 second response time on port 50090
- HDFS capacity utilization OK for 2 months OK: DFISUsedGB<78.7>, DFSTotalGB<1158.4>
- Block health OK for 2 months OK: missing_blocks:<0>, total_blocks:<1330>
- NameNode RPC latency on limulus OK for 2 months OK: RpcQueueTime_avg_time:<0> Secs, RpcProcessingTime_avg_time:<0> Secs
- NameNode process on limulus OK for 2 months TCP OK - 0.000 second response time on port 8020

HDFS Service Metrics

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



Ambari

hadoop2



Services

admin

- HDFS
 - MapReduce2
 - YARN
 - Tez
 - Nagios
 - Ganglia
 - Hive
 - HBase
 - Pig
 - Sqoop
 - Oozie
 - ZooKeeper
 - Flume
- Actions** -

Summary Configs Quick Links - Service Actions -

Group: HDFS Default (4) Manage Config Groups Filter...

V1 admin 2 months ago Current

NameNode hosts: limulus

NameNode directories: /hdfs1/hadoop/hdfs/namenode/

NameNode Java heap size: 1024 MB

NameNode new generation size: 200 MB

NameNode maximum new generation size: 200 MB

NameNode permanent generation size: 128 MB

NameNode maximum permanent generation size: 256 MB

Secondary NameNode

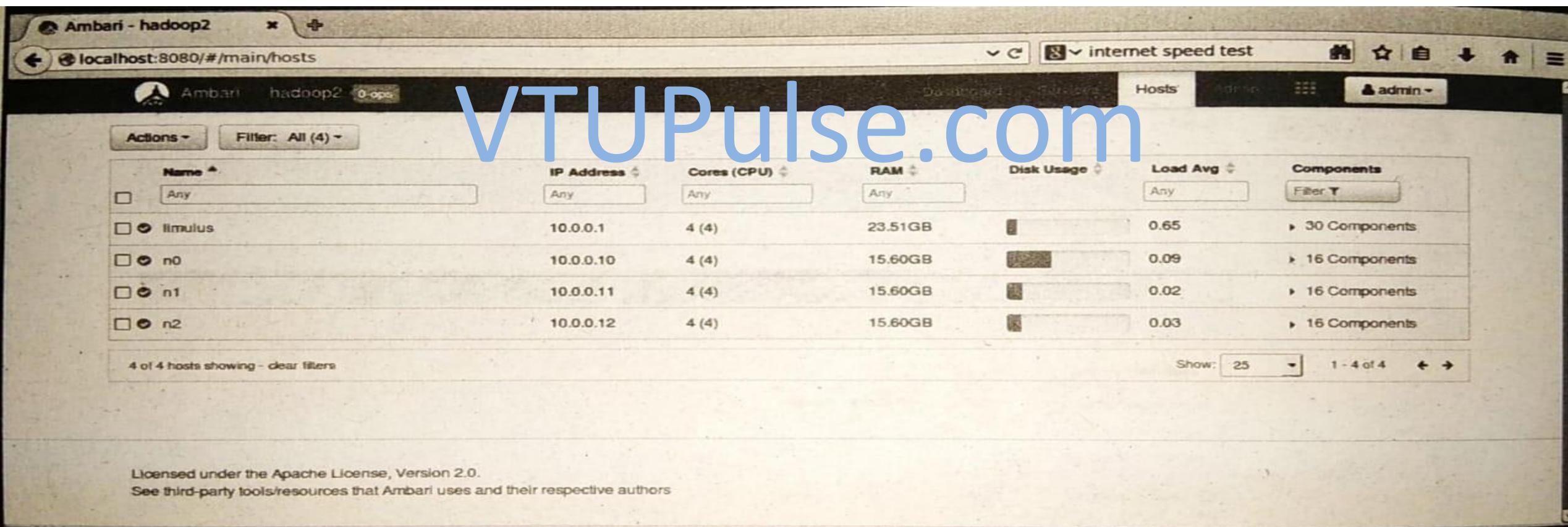
SNameNode host: limulus

SecondaryNameNode: /var/lib/hadoop/hdfs/namesecondary

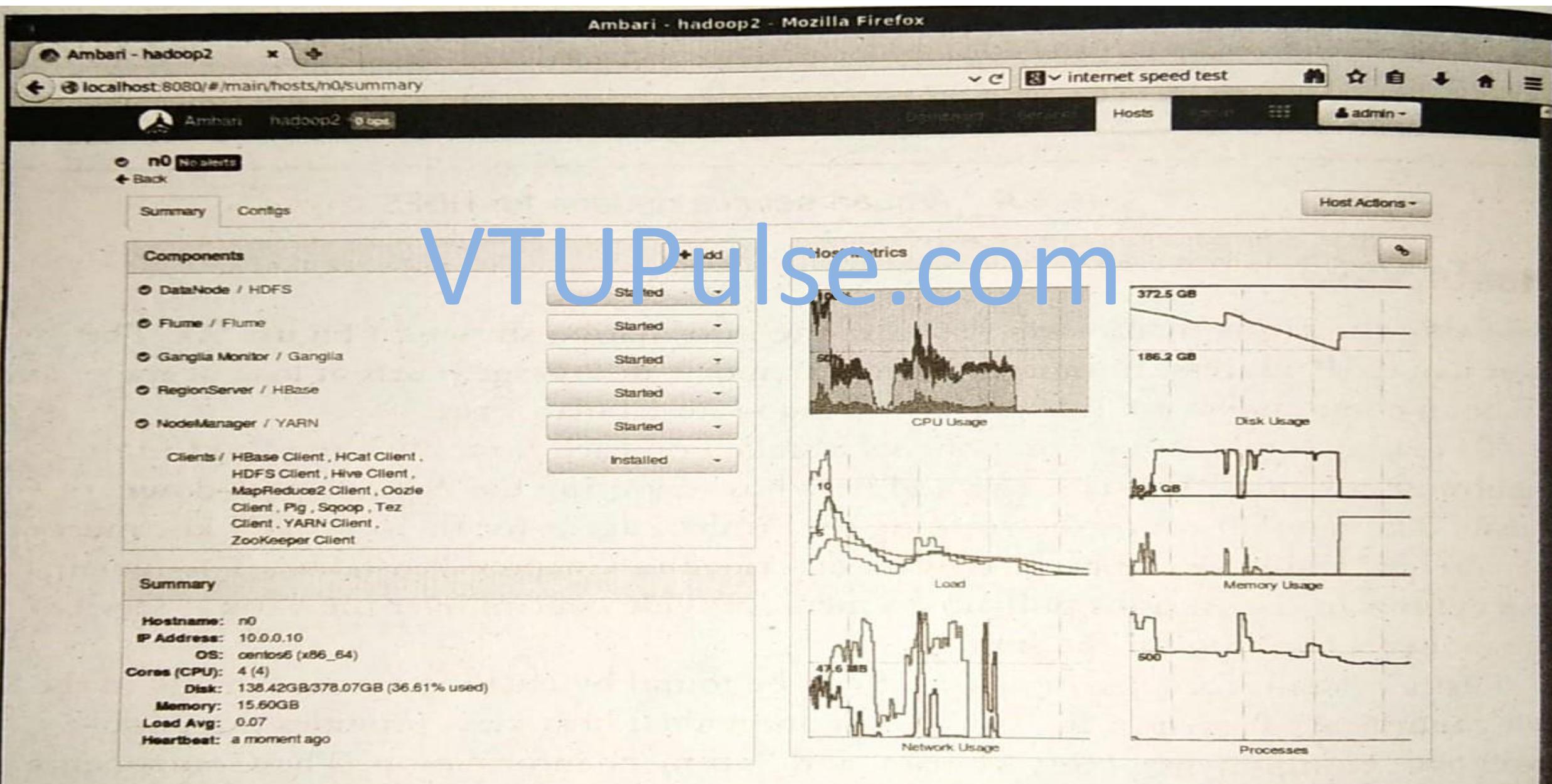
For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Hosts View

- The host name, IP address, number of cores, memory, disk usage, current load average, and Hadoop components are listed in this window in tabular form.
- To display the Hadoop components installed on each host, click the links in the rightmost columns. You can also add new hosts by using the Actions pull-down menu.



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Licensed under the Apache License, Version 2.0.

See third-party tools/resources that Ambari uses and their respective authors

Admin View

- The Administration (Admin) view provides three options. The first, as shown in Figure, displays a list of installed software. This Repositories listing generally reflects the version of Hortonworks Data Platform (HDP) used during the installation process.
- Second, the Service Accounts option lists the service accounts added when the system was installed. These accounts are used to run various services and tests for Ambari.
- The third option, Security, sets the security on the cluster. A fully secured Hadoop cluster is important in many instances and should be explored if a secure environment is needed. This aspect of Ambari is beyond the scope of this book.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



Ambari

hadoop2



Dashboard

Services

Logs

Admin



admin

Repositories

Service Accounts

Security

Cluster Stack Version: HDP-2.2

Service	Version	Description
Falcon	0.6.0.2.2.0.0	Data management and processing platform
Flume	1.5.2.2.2.0.0	A distributed service for collecting, aggregating, and moving large amounts of streaming data into HDFS
Ganglia	3.5.0	Ganglia Metrics Collection system (RRD Tool will be installed too)
HBase	0.96.4.2.2.0.0	No-relational distributed database and centralized service for configuration management & synchronization
HDFS	2.6.0.2.2.0.0	Apache Hadoop Distributed File System
Hive	0.14.0.2.2.0.0	Data warehouse system for ad-hoc queries & analysis of large datasets and table & storage management service
Kafka	0.8.1.2.2.0.0	A high-throughput distributed messaging system
Knox	0.5.0.2.2.0.0	Provides a single point of authentication and access for Apache Hadoop services in a cluster
Nagios	3.5.0	Nagios Monitoring and Alerting system
Oozie	4.1.0.2.2.0.0	System for workflow coordination and execution of Apache Hadoop jobs. This also includes the installation of the optional Oozie Web Console which relies on and will install the ExtJS Library.
Pig	0.14.0.2.2.0.0	Scripting platform for analyzing large datasets
Slider	0.60.0.2.2.0.0	A framework for deploying, managing and monitoring existing distributed applications on YARN.
Sqoop	1.4.5.2.2.0.0	Tool for transferring bulk data between Apache Hadoop and structured data stores such as relational databases
Storm	0.9.3.2.2.0.0	Apache Hadoop Stream processing framework
Tez	0.5.2.2.2.0.0	Tez is the next generation Hadoop Query Processing framework written on top of YARN.
YARN + MapReduce2	2.6.0.2.2.0.0	Apache Hadoop NextGen MapReduce (YARN)
ZooKeeper	3.4.6.2.2.0.0	Centralized service which provides highly reliable distributed coordination

Repositories

2.2

OS

Name

Base URL

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Managing Hadoop Services

- During the course of normal Hadoop cluster operation, services may fail for any number of reasons.
- Ambari monitors all of the Hadoop services and reports any service interruption to the dashboard.
- In addition, when the system was installed, an administrative email for the Nagios monitoring system was required. All service interruption notifications are sent to this email address.
- Figure 9.10 shows the Ambari dashboard reporting a down DataNode.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Ambari - hadoop2 x Namenode Information x

localhost:8080/#/main/services/HDFS/summary

Ambari hadoop2 ops Services admin

HDFS 2

MapReduce2

YARN

Tez

Nagios

Ganglia

Hive

HBase

Pig

Sqoop

Oozie

ZooKeeper

Flume

Actions

Summary Configs Quick Links Service Actions

Summary

NameNode: Started
DataNodes: 3/4 DataNodes Live
NameNode Uptime: 52.86 days
NameNode Heap: 155.6 MB / 104.0 MB (35.1% used)
DataNodes Status: 3 live / 1 dead / 0 decommissioning

Disk Usage (DFS Used): 73.8 GB / 1.0 TB (7.03%)
Disk Usage (Non DFS Used): 126.8 GB / 1.0 TB (12.08%)
Disk Usage (Remaining): 849.5 GB / 1.0 TB (80.90%)
Blocks (total): 1352
Block Errors: 0 corrupt / 0 missing / 385 under replicated
Total Files + Directories: 300574
Upgrade Status: No pending upgrade
Safe Mode Status: Not in safe mode

Alerts and Health Checks

- Percent DataNodes live: WARN for 5 minutes (WARNING: total:<4>, affected:<1>)
- Percent DataNodes with space available: WARN for 7 minutes (WARNING: total:<4>, affected:<1>)
- Secondary NameNode process: OK for 2 months (TCP OK - 0.000 second response time on port 50090)
- HDFS capacity utilization: OK for 2 months (OK: DFSUsedGB:<79.3>, DFSTotalGB:<1237.7>)
- Blocks health: OK for 2 months (OK: missing_blocks:<0>, total_blocks:<1352>)
- NameNode RPC latency on llmnlus: OK for 2 months (OK: RpcQueueTime_avg_time:<0> Secs, RpcProcessingTime_avg_time:<0> Secs)
- NameNode process on llmnlus: OK for 2 months (TCP OK - 0.000 second response time on port 8020)

HDFS Service Metrics

Total Space Utilization: 931.3 GB

File Operations: 0.5 op/s

Block Status: 2

HDFS IO: 555.1 PB

444.0 PB

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

VTUPulse.com

The screenshot shows the Ambari web interface for a Hadoop cluster named 'hadoop2'. The main title bar says 'Ambari - hadoop2' and the sub-title bar says 'localhost:8080/#/main/hosts'. The top navigation bar includes links for 'Dashboard', 'Services', 'Hosts' (with a count of 1), 'Admin', and a user 'admin'. The main content area displays a table of hosts:

Name	IP Address	Cores (CPU)	RAM	Disk Usage	Load Avg	Components
Any	Any	Any	Any	Any	Any	Filter T
stimulus	10.0.0.1	4 (4)	23.51GB	2.28	30 Components	
n0	10.0.0.10	4 (4)	15.60GB	0.03	16 Components	
n1	10.0.0.11	4 (4)	15.60GB	0.06	16 Components	
n2	10.0.0.12	4 (4)	15.60GB	0.00	16 Components	

At the bottom left, it says '4 of 4 hosts showing - clear filters'. On the right, there are buttons for 'Show: 25' and '1 - 4 of 4'.

Licensed under the Apache License, Version 2.0.
See third-party tools/resources that Ambari uses and their respective authors

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Ambari - hadoop2 x Namenode Information x

localhost:8080/#/main/hosts/n1/summary

Ambari hadoop2 Ops

Dashboard Services Hosts Admin admin

n1 Back

Summary Configs Host Actions

Components

- ▲ DataNode / HDFS
- Flume / Flume
- Ganglia Monitor / Ganglia
- RegionServer / HBase
- NodeManager / YARN

Clients / HBase Client , HCat Client ,
HDFS Client , Hive Client ,
MapReduce2 Client , Oozie
Client , Pig , Sqoop , Tez
Client , YARN Client ,
ZooKeeper Client

Host Metrics

CPU Usage Disk Usage

Load Memory Usage

Network Usage Processes

VTUPulse.com

Licensed under the Apache License, Version 2.0.

See third-party tools/resources that Ambari uses and their respective authors

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

1 Background Operations Running

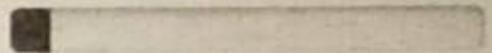
x

Operations

Start Time

Duration

Show: All (10)

⚙ Start DataNode  9% 

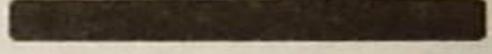
Today 14:29

6.08 secs

✓ Stop DataNode  100% 

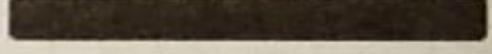
Today 14:28

7.75 secs

✓ Start DataNode  100% 

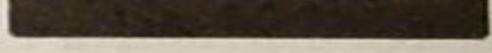
Today 14:28

14.64 secs

✓ Stop DataNode  100% 

Today 14:13

11.72 secs

✓ Restart all components with Stale
Configs for Oozie  100% 

Mon Jun 15 2015 16:27

8.32 secs

✓ Restart components with Stale Configs
on IIMulus  100% 

Mon Jun 01 2015 20:52

32.36 secs

✓ Restart all components with Stale  100% 

Thu May 28 2015 12:39

50.50 secs

 Do not show this dialog again when starting a background operation

OK

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Changing Hadoop Properties

- One of the challenges of managing a Hadoop cluster is managing changes to Clusterwide configuration properties. In addition to modifying a large number of properties, making changes to a property often requires restarting daemons (and dependent daemons) across the entire cluster. This process is tedious and time consuming. Fortunately, Ambari provides an easy way to manage this process.
- As described previously, each service provides a Configs tab that opens a form displaying all the possible service properties. Any service property can be changed (or added) using this interface. As an example, the configuration properties for the YARN scheduler are shown in Figure.

Ambari - hadoop2 x Namenode information x

localhost:8080/#/main/services/YARN/configs

Ambari hadoop2 0 ops Services admin

Summary Configs Quick Links - Service Actions

YARN Default (4) Manage Config Groups Filter...

V11 admin V10 admin V9 admin V8 admin V7 admin V6 admin

19 days ago 19 days ago

Resource Manager

ResourceManager limulus
ResourceManager Java heap size 1024 MB

yarn.acl.enable

yarn.admin.acl

yarn.log-aggregation-enable

Node Manager

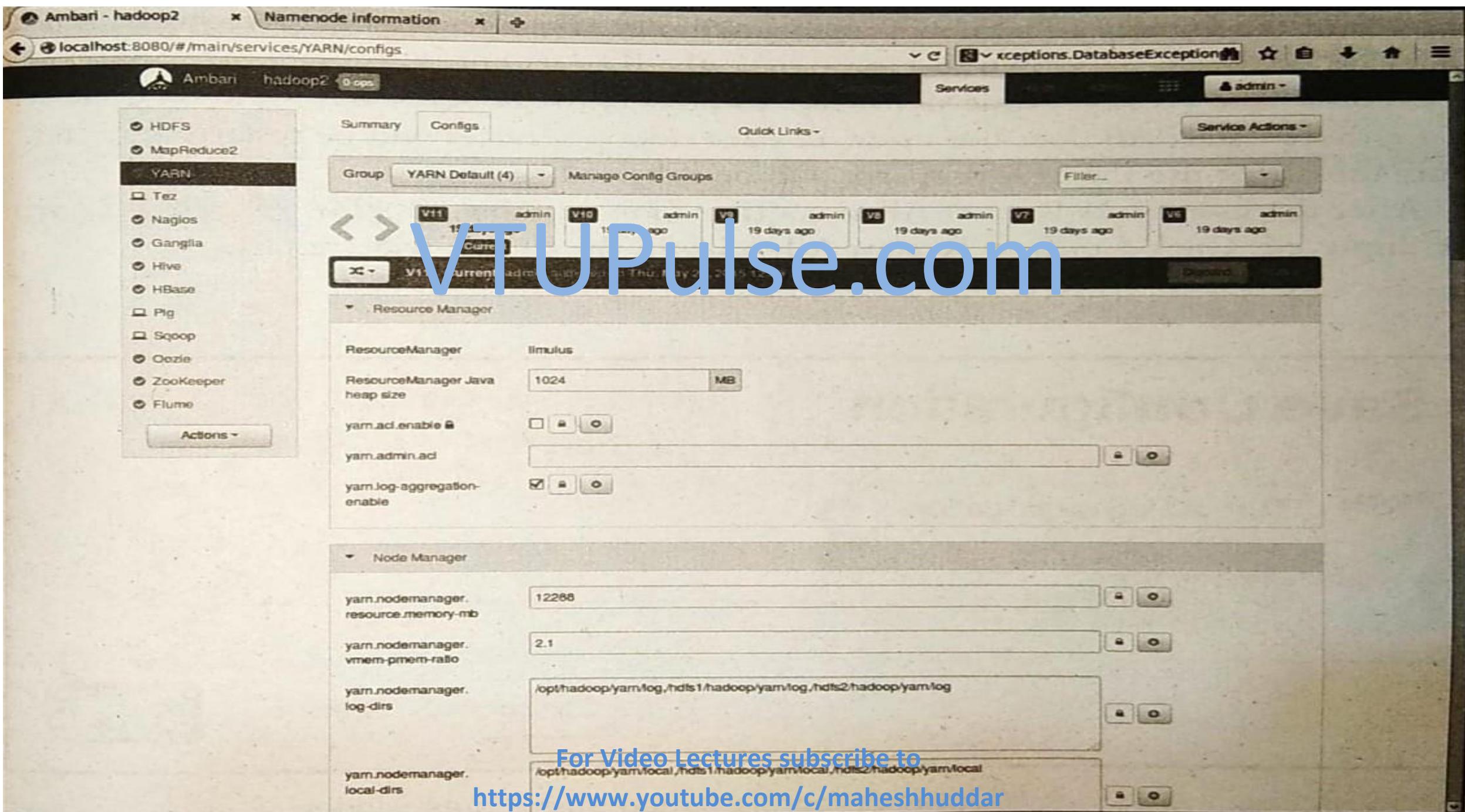
yarn.nodemanager.resource.memory-mb 12288

yarn.nodemanager.vmem-pmem-ratio 2.1

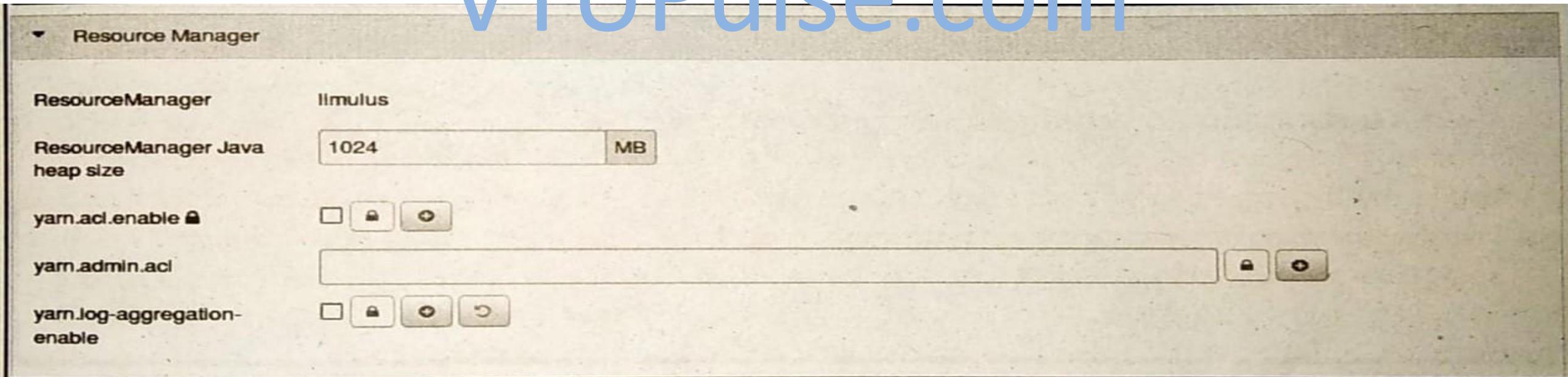
yarn.nodemanager.log-dirs /opt/hadoop/yarn/log/hdfs1/hadoop/yarn/log/hdfs2/hadoop/yarn/log

yarn.nodemanager.local-dirs /opt/hadoop/yarn/local/hdfs1/hadoop/yarn/local/hdfs2/hadoop/yarn/local

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



VTUPulse.com



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Save Configuration

X

Notes

Turn off log aggregation

Cancel

Discard

Save

Save Configuration Changes

x

VTUPulse.com

Service configuration changes saved successfully.

OK

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>



Figure 9.21 Ambari Restart function appears after changes in service properties

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Confirmation

X

You are about to restart YARN

VTUPulse.com

This will trigger alerts as the service is restarted. To suppress alerts, turn on Maintenance Mode for YARN prior to running restart all

Cancel

Confirm Restart All

1 Background Operations Running

X

Operations**Start Time****Duration****Show:** All (10)

Restart all components with Stale Configs for YARN

Today 14:52

18.13 secs

35%

Start DataNode

Today 14:29

14.50 secs

100%

Stop DataNode

Today 14:28

7.75 secs

100%

Start DataNode

Today 14:28

14.64 secs

100%

Stop DataNode

Today 14:13

11.72 secs

100%

Restart all components with Stale Configs for Oozie

Mon Jun 15 2015 16:27

8.32 secs

100%

Restart components with Stale Configs

Mon Jun 01 2015 20:52

32.36 secs

100%

 Do not show this dialog again when starting a background operation

OK

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Summary Configs Quick Links ▾ Service Actions ▾

Group YARN Default (4) Manage Config Groups Filter...

V12 admin a moment ago V11 admin 19 days ago V10 admin 19 days ago V9 admin 19 days ago V8 admin 9 days ago V7 admin 19 days ago

V12 Current admin authorized on Tue, Jun 16, 2015 14:49

Resource Manager

ResourceManager limulus

ResourceManager Java heap size 1024 MB

yarn.acl.enable

yarn.admin.acl

yarn.log-aggregation-enable



For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Summary **Configs**

V11 **YARN Default**
admin authored on Thu, May 28, 2015 12:39
No notes

Service Actions ▾

Group **YARN Default (4)**

V12 admin 7 minutes ago **V11** admin 19 days ago **V1** admin 19 days ago **V9** admin 19 days ago **V8** admin 19 days ago **V7** admin 19 days ago

View **Compare** **Make Current** **Filter...**

Current

x **V11** admin authored on Thu, May 28, 2015 12:39 **Make V11 Current**

Resource Manager

ResourceManager **lilimulus**

ResourceManager Java heap size **1024 MB**

yarn.acl.enable

yarn.admin.acl

yarn.log-aggregation-enable

VTUPulse.com

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Make Current Confirmation

X

Notes

restore log aggregation

Cancel

Discard

Make Current

Basic Hadoop Administration Procedures

- Hadoop has two main areas of administration: the YARN resource manager and the HDFS file system. Other application frameworks (e.g., the MapReduce framework) and tools have their own management files.
- Hadoop configuration is accomplished through the use of XML configuration files. The basic files and their function are as follows:
 - **core-default.xml**: System-wide properties
 - **hdfs-default.xml**: Hadoop Distributed File System properties
 - **mapred-default.xml**: Properties for the YARN MapReduce framework
 - **yarn-default .xml**: YARN properties

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

- YARN has several built-in administrative features and commands.
- To find out more about them, examine the YARN commands documentation at

https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YarnCommands.html#Administration_Commands

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

Decommissioning YARN Nodes

- If a NodeManager host/node needs to be removed from the cluster, it should be decommissioned first.
- Assuming the node is responding, you can easily decommission it from the Ambari web UI.
- Simply go to the Hosts view, click on the host, and select Decommission from the pull-down menu next to the NodeManager component.

Basic Hadoop YARN Administration

YARN WebProxy

- The Web Application Proxy is a separate proxy server in YARN that addresses security issues with the cluster web interface on ApplicationMasters.
- By default, the proxy runs as part of the Resource Manager itself, but it can be configured to run in a stand-alone mode by adding the configuration property `yarn.web-proxy.address` to `yarn-site.xml`

Basic Hadoop YARN Administration

Using the JobHistoryServer

- The removal of the JobTracker and migration of MapReduce from a system to an application-level framework necessitated creation of a place to store MapReduce job history.
- The JobHistoryServer provides all YARN MapReduce applications with a central location in which to aggregate completed jobs for historical reference and debugging.
- The settings for the JobHistoryServer can be found in the mapred-site.xml file.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

Managing YARN Jobs

- YARN jobs can be managed using the `yarn` application command. The following options, including `-kill`, `-list`, and `-status`, are available to the administrator with this command.
- MapReduce jobs can also be controlled with the `mapred` job command.
- Usage

application –option [application ID]

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

Setting Container Memory

- YARN manages application resource containers over the entire cluster. Controlling the amount of container memory takes place through three important values in the `yarn-site.xml` file:
- **`yarn.nodemanager.resource.memory-mb`** is the amount of memory the NodeManager can use for containers.
- **`Yarn.scheduler.minimum-allocation-mb`** is the smallest container allowed by the Resource Manager. A requested container smaller than this value will result in an allocated container of this size (default 1024MB).
- **`yarn.scheduler.maximum-allocation-mb`** is the largest container allowed by the Resource Manager (default 8192MB)

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

Setting Container Cores

- **yarn.scheduler.minimum-allocation-vcores:** The minimum allocation for every container request at the Resource Manager, in terms of virtual CPU cores. Requests smaller than this allocation will not take effect, and the specified value will be allocated the minimum number of cores. The default is 1 core.
- **yarn.scheduler.maximum-allocation-vcores:** The maximum allocation for every container request at the Resource Manager, in terms of virtual CPU cores. Requests larger than this allocation will not take effect, and the number of cores will be capped at this value. The default is 32.
- **yarn.nodemanager.resource.cpu-vcores:** The number of CPU cores that can be allocated for containers. The default is 8.
For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Basic Hadoop YARN Administration

Setting MapReduce Properties

- As noted throughout this book, MapReduce now runs as a YARN application. Consequently, it may be necessary to adjust some of the mapred-site.xml properties as they relate to the map and reduce containers. The following properties are used to set some Java arguments and memory size for both the map and reduce containers:
 - mapred.child.java.opts** provides a larger or smaller heap size for child JVMs of maps
 - mapreduce.map.memory.mb** provides a larger or smaller resource limit for maps (default = 1536MB).
 - mapreduce.reduce.memory.mb** provides a larger heap size for child JVMs of maps (default = 3072MB).
 - mapreduce.reduce.java.opts** provides a larger or smaller heap size for child reducers.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

The NameNode User Interface

- Monitoring HDFS can be done in several ways. One of the more convenient ways to get a quick view of HDFS status is through the NameNode user interface.
- This web-based tool provides essential information about HDFS and offers the capability to browse the HDFS namespace and logs.
- URL to open UI: <http://10ca1host:50070>
- There are five tabs on the UI: Overview, Datanodes, Snapshot, Startup Progress, and Utilities.
- The **Overview** page provides much of the essential information that the command-line tools also offer, but in a much easier-to-read format.
- The **Datanodes** tab displays node information like that shown in Figure.
- The **Snapshot** window lists the "snapshottable" directories and the snapshots.
- **Startup progress** gives you the details process timings
- The **utility** menu offers two options: First, file system browser, where you can easily explore the HDFS namespace and second, Links to various name node logs.
For Video Lectures subscribe to
<https://www.youtube.com/c/maneshhuddar>

Overview 'limulus:8020' (active)

Started:	Tue Jun 23 21:36:29 EDT 2015
Version:	2.6.0.2.2.4.0-2122a563cbe4c8949d7f02ee8f3ac1cc52b28
Compiled:	2015-03-31 19:49:21 by jenkins from (no branch)
Cluster ID:	CID-b24693ba-9cc7-4750-bbc8-08c4d88f26e7
Block Pool ID:	BP-1208979959-10.0.0.1-1429639633828

Summary

Security is off.

Safemode is off.

351446 files and directories, 1063 blocks = 352509 total filesystem object(s).

Heap Memory used 221.75 MB of 1004 MB Heap Memory. Max Heap Memory is 1004 MB.

Non Heap Memory used 48.97 MB of 134.56 MB Committed Non Heap Memory. Max Non Heap Memory is 304 MB.

Configured Capacity:	1.37 TB
DFS Used:	55.35 GB
Non DFS Used:	89.95 GB
DFS Remaining:	1.23 TB
DFS Used%:	3.95%

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Ambari - hadoop2 All Applications Namenode information

limulus:50070/dfshealth.html#tab-datanode

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
n1 (10.0.0.11:50010)	1	In Service	350.1 GB	7.87 GB	24.48 GB	317.75 GB	352	7.87 GB (2.25%)	0	2.6.0.2.2.4.2-2
n0 (10.0.0.10:50010)	1	In Service	350.1 GB	11.55 GB	24.52 GB	314.03 GB	174	11.55 GB (3.3%)	0	2.6.0.2.2.4.2-2
limulus (10.0.0.1:50010)	1	In Service	349.85 GB	19.06 GB	18.47 GB	314.31 GB	956	19.06 GB (5.45%)	0	2.6.0.2.2.4.2-2
n2 (10.0.0.12:50010)	1	In Service	350.1 GB	16.86 GB	24.48 GB	308.77 GB	634	16.86 GB (4.83%)	0	2.6.0.2.2.4.2-2

Decommissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction
------	--------------	-------------------------	------------------------------	--

Hadoop, 2014.

Legacy interface

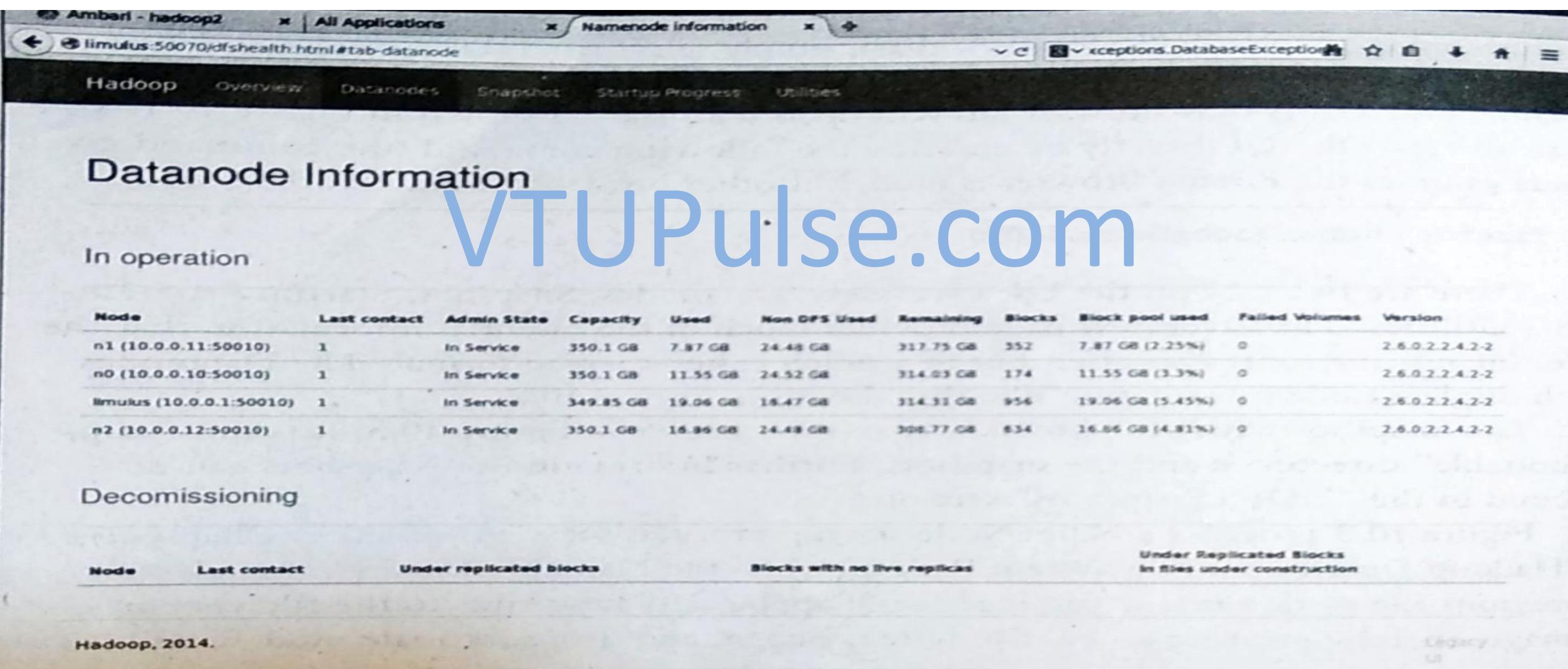


Figure 10.2 NameNode web interface showing status of DataNodes

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Startup Progress

Elapsed Time: 2 mins, 6 sec, Percent Complete: 100%

Phase	Completion	Elapsed Time
Loading fsimage /hdfs1/hadoop/hdfs/namenode/current/fsimage_000000000000099638-000000000000099793 1 MB (637 : 635 MB)	100%	2 sec
inodes (0/0)	100%	
delegation tokens (0/0)	100%	
cache pools (0/0)	100%	
Loading edits	100%	0 sec
/hdfs1/hadoop/hdfs/namenode/current/edits_000000000000099638-000000000000099793 1 MB (156/156)	100%	
Saving checkpoint	100%	0 sec
Safe mode	100%	2 mins, 3 sec
awaiting reported blocks (971/971)	100%	

Hadoop, 2014.

Lega
UI

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Browse Directory

/

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxrwxrwx	yarn	hadoop	0 B	0	0 B	app-logs
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	apps
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	benchmarks
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	hdp
drwxr-xr-x	mapred	hdfs	0 B	0	0 B	mapred
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	mr-history
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	system
drwxrwxrwx	hdfs	hdfs	0 B	0	0 B	tmp
drwxr-xr-x	hdfs	hdfs	0 B	0	0 B	user
drwx-wx-wx	hdfs	hdfs	0 B	0	0 B	var

Hadoop, 2014.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

Adding Users to HDFS

- To quickly create user accounts manually on a Linux-based system, perform the following steps:
- Add the user to the group for your operating system on the HDFS client system. In most cases, the groupname should be that of the HDFS superuser, which is often hadoop or hdfs.

useradd -G <groupname> <username>

- Create the username directory in HDFS.

hdfs dfs -mkdir / user/<username>

- Give that account ownership over its .directory in HDFS.

hdfs dfs -chown <username>:<groupname> /user/<username>

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

Perform an FSCK on HDFS

- To check the health of HDFS, you can issue the `hdfs fsck` (file system check) command.
- The entire HDFS namespace can be checked, or a subdirectory can be entered as an argument to the command.
- The following example checks the entire HDFS namespace.

```
hdfs fsck /
```
- Other options provide more detail, include snapshots and open files, and management of corrupted files.
 - move** moves corrupted files to / lost+found.
 - delete** deletes corrupted files.
 - files** prints out files being checked.
 - openforwrite** prints out files opened for writes during check.
 - list-corruptfileblocks** prints out a list of missing blocks
 - blocks** prints out a block report.
 - locations** prints out locations for every block.
 - racks** prints out network topology for data-node locations.

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>

.....

Status: HEALTHY

Total size: 100433565781 B (Total open files size: 498 B)

Total dirs: 201331

Total files: 1003

Total symlinks: 0 (Files currently being written: 6)

Total blocks (validated): 1735 (avg. block size 57836781 B) (Total open file blocks (not validated): 6)

Minimally replicated blocks: 1735 (100.0 %)

Over-replicated blocks: 0 (0.0 %)

Under-replicated blocks: 0 (0.0 %)

Mis-replicated blocks: 0 (0.0 %)

Default replication factor: 2

Average block replication: 1.7850144

Corrupt blocks: 0

Missing replicas: 0 (0.0 %)

Number of data-nodes: 4

Number of racks: 1

FSCK ended at Fri May 29 14:48:03 EDT 2015 in 1853 milliseconds

Basic HDFS Administration

Balancing HDFS

- Based on usage patterns and DataNode availability, the number of data blocks across the DataNodes may become unbalanced.
- To avoid over-utilized DataNodes, the HDFS balancer tool rebalances data blocks across the available DataNodes.
- Data blocks are moved from over-utilized to under-utilized nodes to within a certain percent threshold.
- Rebalancing can be done when new DataNodes are added or when a DataNode is removed from service.
- This step does not create more space in HDFS, but rather improves efficiency.

hdfs balancer

VTUPulse.com

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

HDFS Safe Mode

- When the NameNode starts, it loads the file system state from the fsimage and then applies the edits log file.
- It then waits for DataNodes to report their blocks. During this time, the NameNode stays in a read-only Safe Mode. The NameNode leaves Safe Mode automatically after the DataNodes have reported that most file system blocks are available.
- The administrator can place HDFS in Safe Mode by giving the following command:
hdfs dfsadmin -safemode enter
- Entering the following command turns off Safe Mode:
hdfs dfsadmin -safemode leave
- HDFS may drop into Safe Mode if a major issue arises within the file system. The file system will not leave Safe Mode until the situation is resolved. To check whether HDFS is in Safe Mode, enter the following command:

hdfs dfsadmin -safemode get

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

Decommissioning HDFS Nodes

- If you need to remove a DataNode host/node from the cluster, you should decommission it first.
- Assuming the node is responding, it can be easily decommissioned from the Ambari web UI.
- Simply go to the Hosts view, click on the host, and select Decommission from the pull-down menu next to the DataNode component.
- Note that the host may also be acting as a Yarn NodeManager. Use the Ambari Hosts view to decommission the YARN host in a similar fashion.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

SecondaryNameNode

VTUPulse.com

- To avoid long NameNode restarts and other issues, the performance of the SecondaryNameNode should be verified.
- The SecondaryNameNode takes the previous file system image file (`fsimage*`) and adds the NameNode file system edits to create a new file system image file for the NameNode to use when it restarts.
- The **hdfs-site.xml** defines a property called **fs.checkpoint.period**.
- This property provides the time in seconds between the SecondaryNameNode checkpoints.

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

HDFS Snapshots

- HDFS snapshots are read-only, point-in-time copies of HDFS. Snapshots can be taken on a subtree of the file system or the entire file system.
- Some common use-cases for snapshots are data backup, protection against user errors, and disaster recovery.
- Snapshots can be taken on any directory once the directory has been set as snapshottable. A snapshottable directory is able to accommodate 65,536 simultaneous snapshots.
- There is no limit on the number of snapshottable directories.
- Administrators may set any directory to be snapshottable, but nested snapshottable directories are not allowed.
- For example, a directory cannot be set to snapshottable if one of its ancestors/descendants is a snapshottable directory

For Video Lectures subscribe to

<https://www.youtube.com/c/maheshhuddar>

Basic HDFS Administration

HDFS Snapshots

VTUPulse.com

- The following example walks through the procedure for creating a snapshot. The first step is to declare a directory as "snapshotable" using the following command:

hdfs dfsadmin -allowSnapshot / user/hdfs/war-and-peace-input

- Allowing snapshot on / user/hdfs/war-and-peace-input succeeded
- Once the directory has been made snapshotable, the snapshot can be taken with the following command. The command requires the directory path and a name for the snapshot—in this case, wapi-snap-1.

hdfs dfs -createSnapshot /user/hdfs/war-and-peace-input wapi-snap-1

- Created snapshot / user/hdfs/war-and-peace-input/.snapshot/wapi-snap-1

For Video Lectures subscribe to
<https://www.youtube.com/c/maheshhuddar>