cloudera

IMC Institute

Big Data Processing Using Cloudera Quickstart with a Docker Container

July 2016

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

Modifiy from Original Version by Danairat T.
Certified Java Programmer, TOGAF – Silver
danairat@gmail.com

Outline

- Launch AWS EC2 Instance
- Install Docker on Ubuntu
- Pull Cloudera QuickStart to the docker
- HDFS
- HBase
- MapReduce
- Hive
- Pig
- Impala
- Sqoop

Hive.apache.org

Cloudera VM

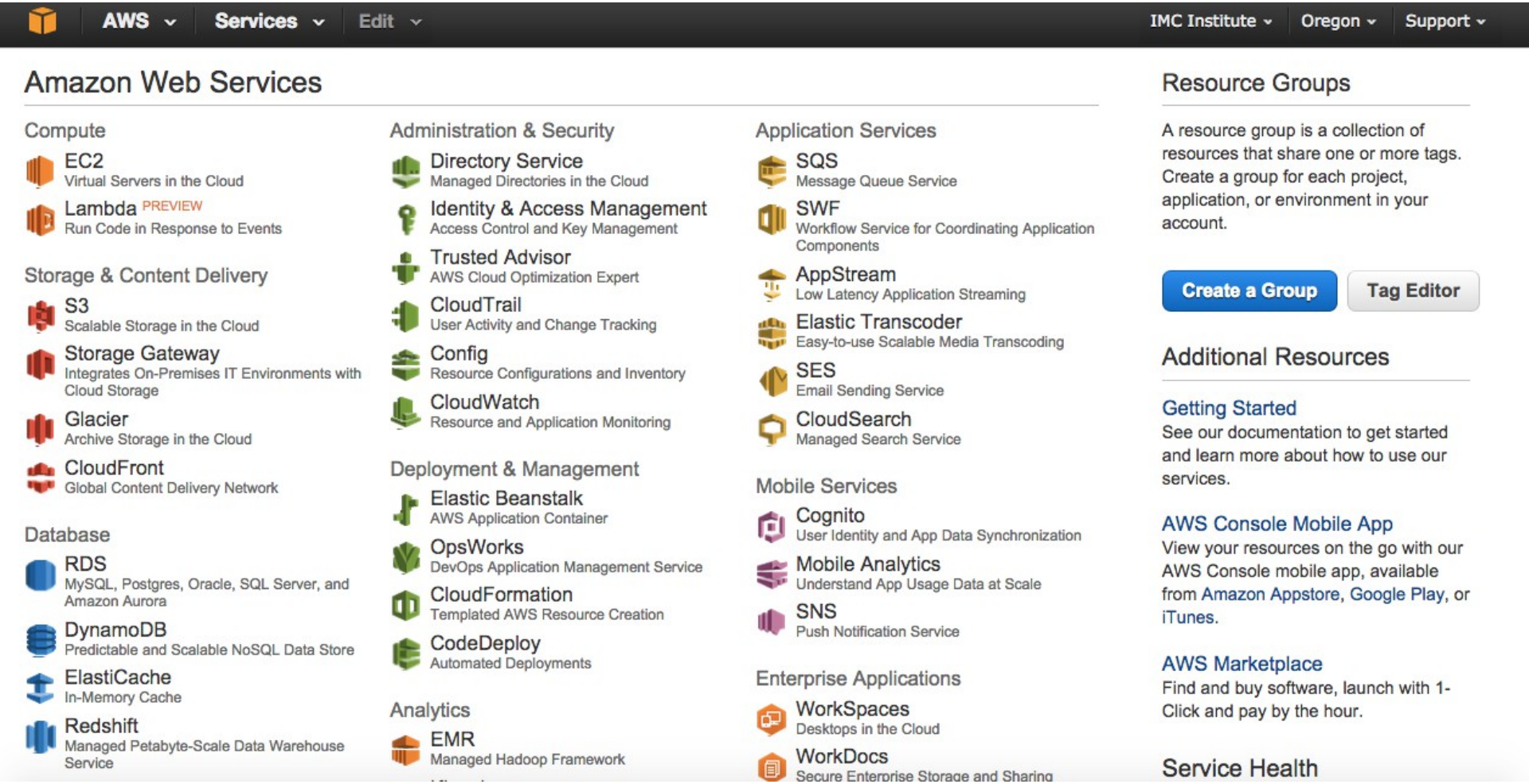
This lab will use a EC2 virtual server on AWS to install Cloudera, However, you can also use Cloudera QuickStart VM which can be downloaded from:

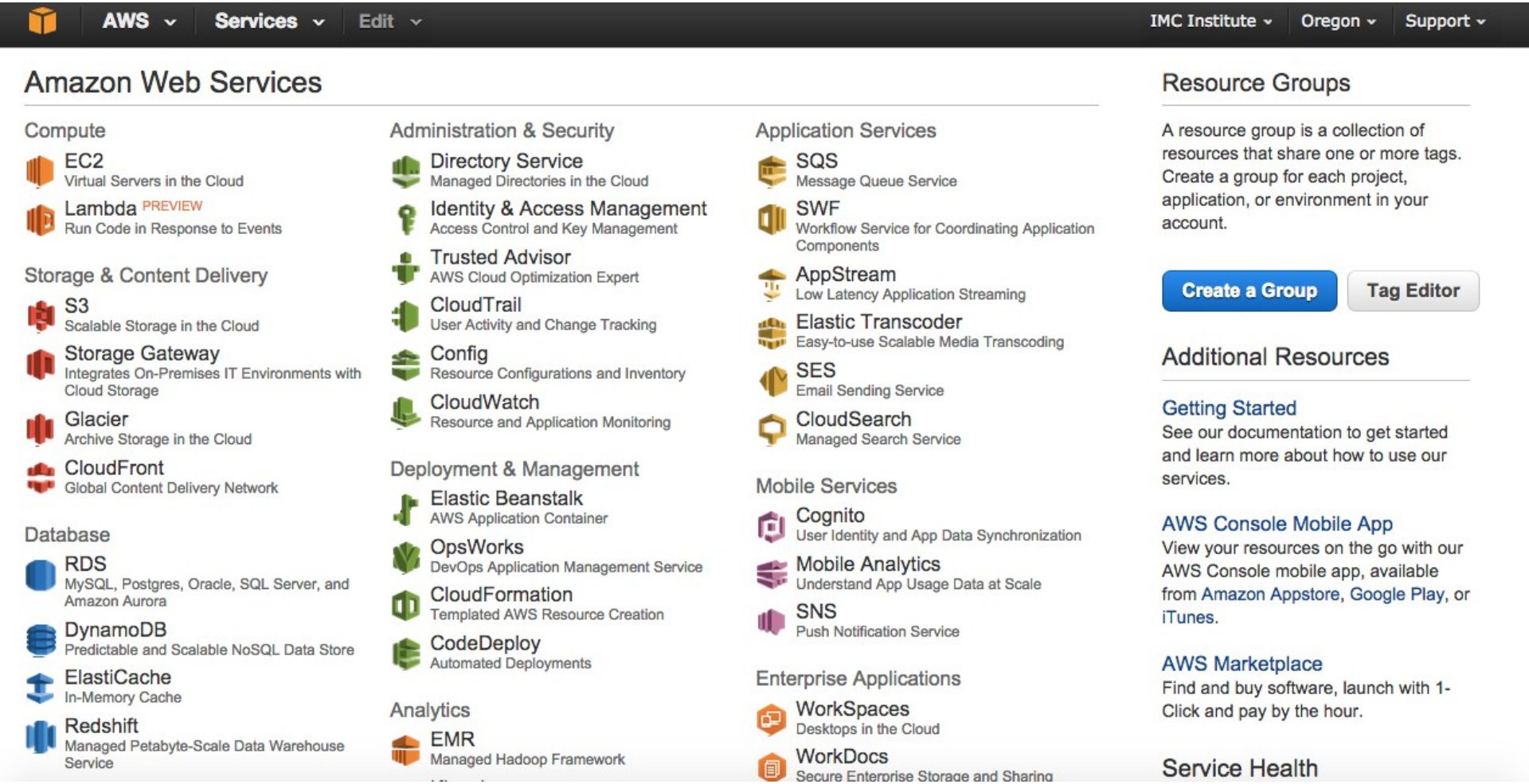
http://www.cloudera.com/content/www/en-us/downloads.html

Hands-On: Launch a virtual server on EC2 Amazon Web Services

(Note: You can skip this session if you use your own computer or another cloud service)

Thanachart Numnonda, thanachart@imcinstitute.com July 2016


AWS
Services
Edit
IMC Institute
Oregon
Support
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
Lambda PREVIEW
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
Storage Gateway
Integrates On-Premises IT Environments with Cloud Storage
Glacier
Archive Storage in the Cloud
CloudFront
Global Content Delivery Network
Database
RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
DynamoDB
Predictable and Scalable NoSQL Data Store
ElastiCache
In-Memory Cache
Redshift
Managed Petabyte-Scale Data Warehouse Service
Administration & Security
Directory Service
Managed Directories in the Cloud
Identity & Access Management
Access Control and Key Management
Trusted Advisor
AWS Cloud Optimization Expert
CloudTrail
User Activity and Change Tracking
Config
Resource Configurations and Inventory
CloudWatch
Resource and Application Monitoring
Deployment & Management
Elastic Beanstalk
AWS Application Container
OpsWorks
DevOps Application Management Service
CloudFormation
Templated AWS Resource Creation
CodeDeploy
Automated Deployments
Analytics
EMR
Managed Hadoop Framework
Application Services
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
AppStream
Low Latency Application Streaming
Elastic Transcoder
Easy-to-use Scalable Media Transcoding
SES
Email Sending Service
CloudSearch
Managed Search Service
Mobile Services
Cognito
User Identity and App Data Synchronization
Mobile Analytics
Understand App Usage Data at Scale
SNS
Push Notification Service
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing
Resource Groups
A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.
Create a Group
Tag Editor
Additional Resources
Getting Started
See our documentation to get started and learn more about how to use our services.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
Service Health

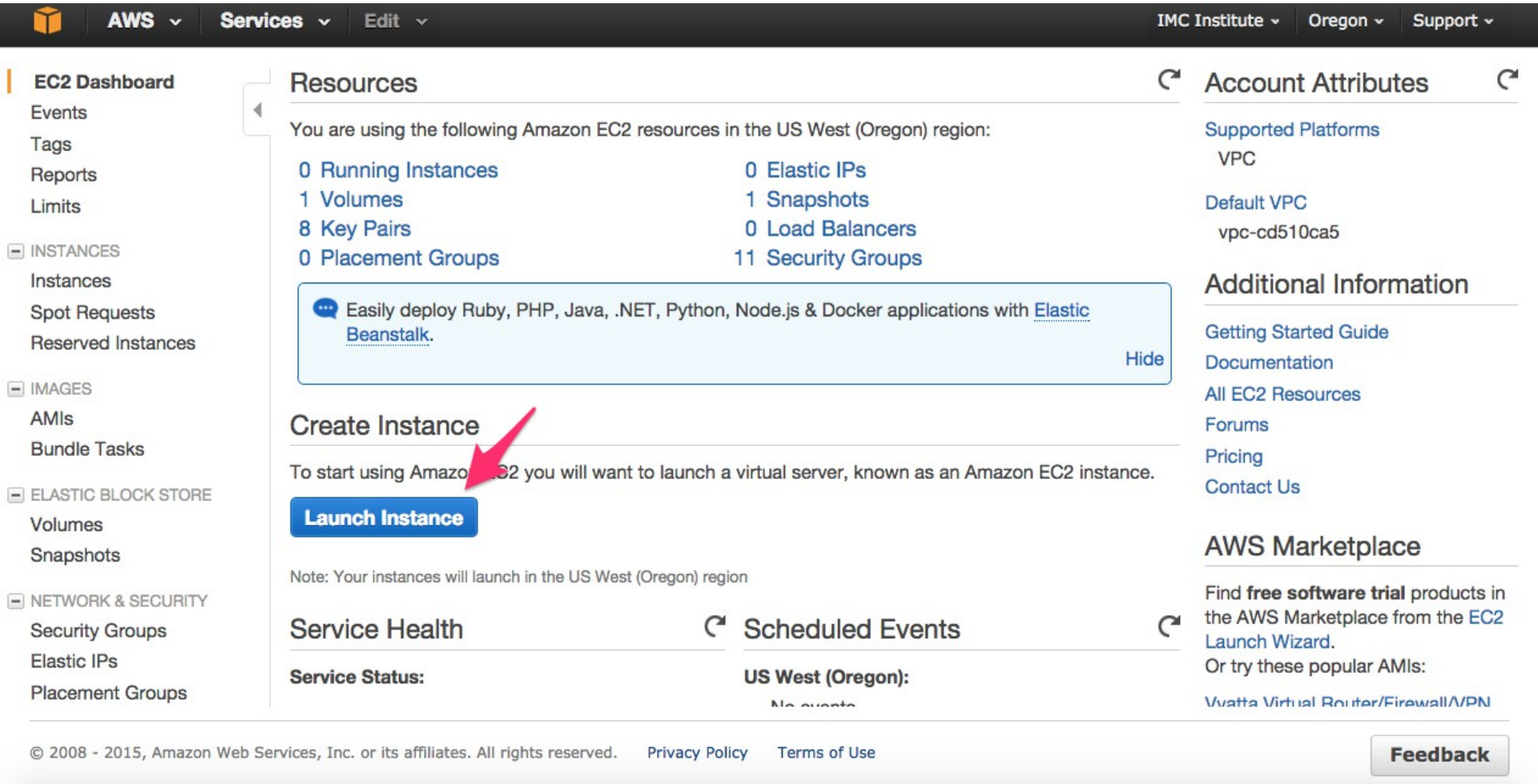

AWS
Services
Edit
IMC Institute
Oregon
Support
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
Lambda PREVIEW
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
Storage Gateway
Integrates On-Premises IT Environments with Cloud Storage
Glacier
Archive Storage in the Cloud
CloudFront
Global Content Delivery Network
Database
RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
DynamoDB
Predictable and Scalable NoSQL Data Store
ElastiCache
In-Memory Cache
Redshift
Managed Petabyte-Scale Data Warehouse Service
Administration & Security
Directory Service
Managed Directories in the Cloud
Identity & Access Management
Access Control and Key Management
Trusted Advisor
AWS Cloud Optimization Expert
CloudTrail
User Activity and Change Tracking
Config
Resource Configurations and Inventory
CloudWatch
Resource and Application Monitoring
Deployment & Management
Elastic Beanstalk
AWS Application Container
OpsWorks
DevOps Application Management Service
CloudFormation
Templated AWS Resource Creation
CodeDeploy
Automated Deployments
Analytics
EMR
Managed Hadoop Framework
Application Services
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
AppStream
Low Latency Application Streaming
Elastic Transcoder
Easy-to-use Scalable Media Transcoding
SES
Email Sending Service
CloudSearch
Managed Search Service
Mobile Services
Cognito
User Identity and App Data Synchronization
Mobile Analytics
Understand App Usage Data at Scale
SNS
Push Notification Service
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing
Resource Groups
A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.
Create a Group
Tag Editor
Additional Resources
Getting Started
See our documentation to get started and learn more about how to use our services.
AWS Console Mobile App
View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.
AWS Marketplace
Find and buy software, launch with 1-Click and pay by the hour.
Service Health

Virtual Server

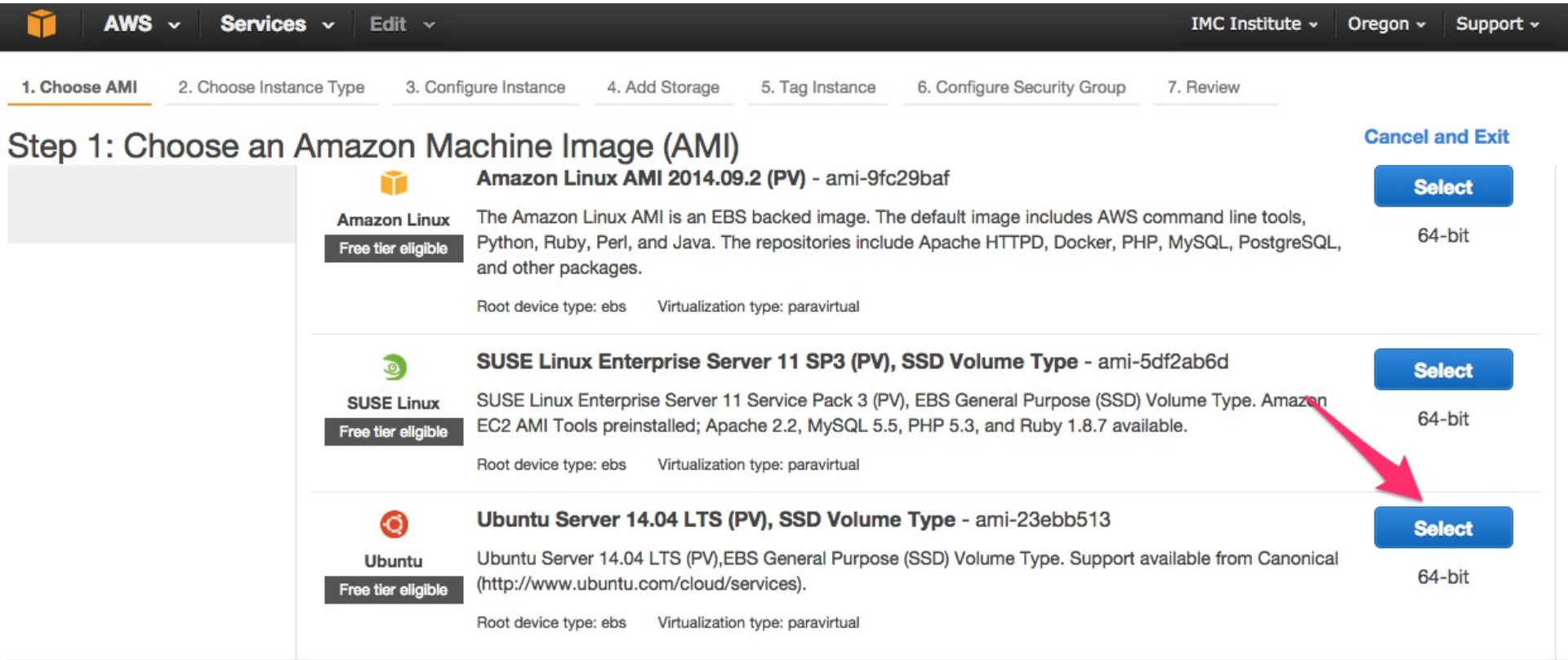
This lab will use a EC2 virtual server to install a Cloudera Cluster using the following features:

- Ubuntu Server 14.04 LTS
- Four m3.xLarge 4vCPU, 15 GB memory,80 GB SSD
- Security group: default
- Keypair: imchadoop

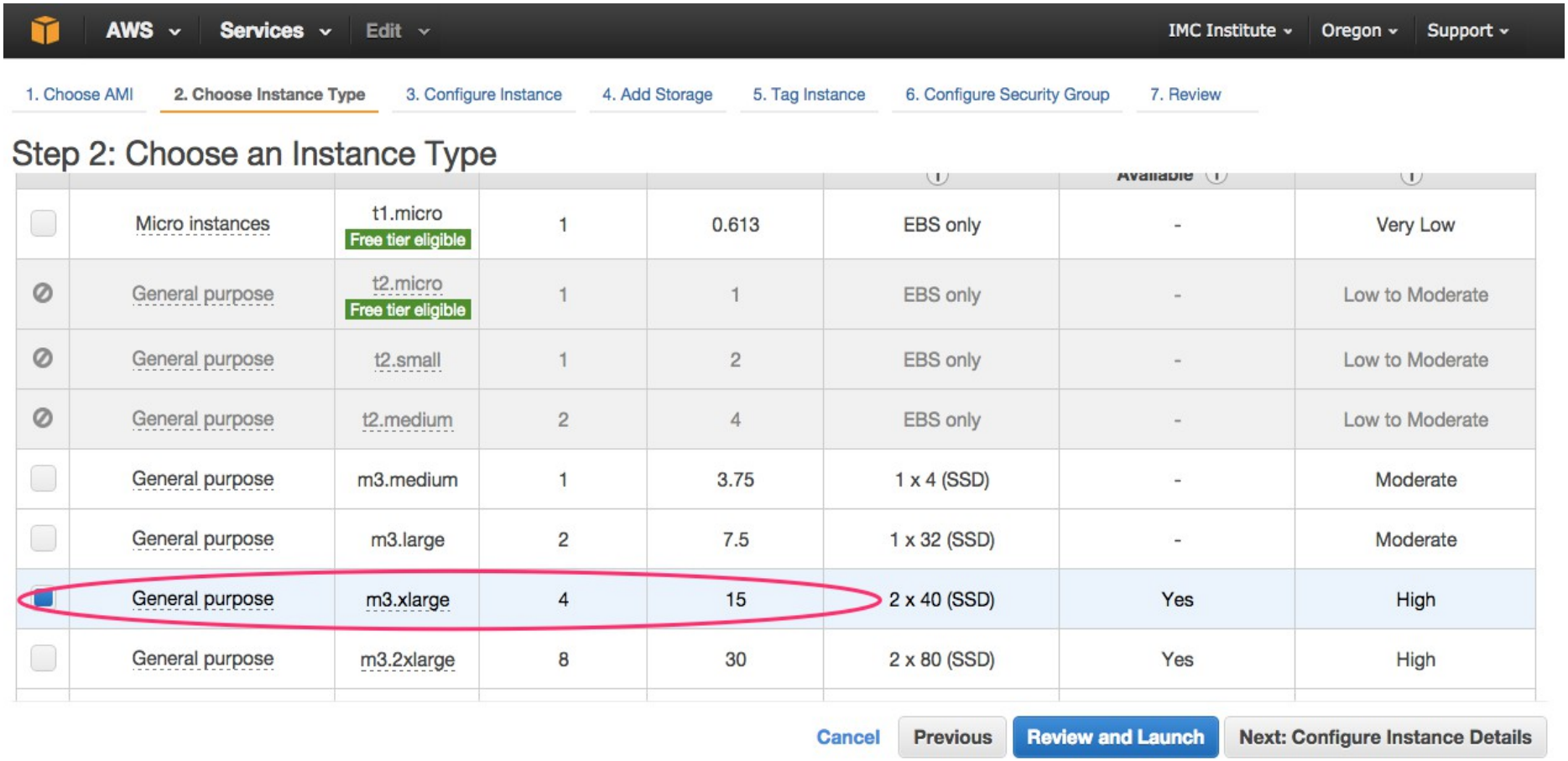
Select a EC2 service and click on Lunch Instance

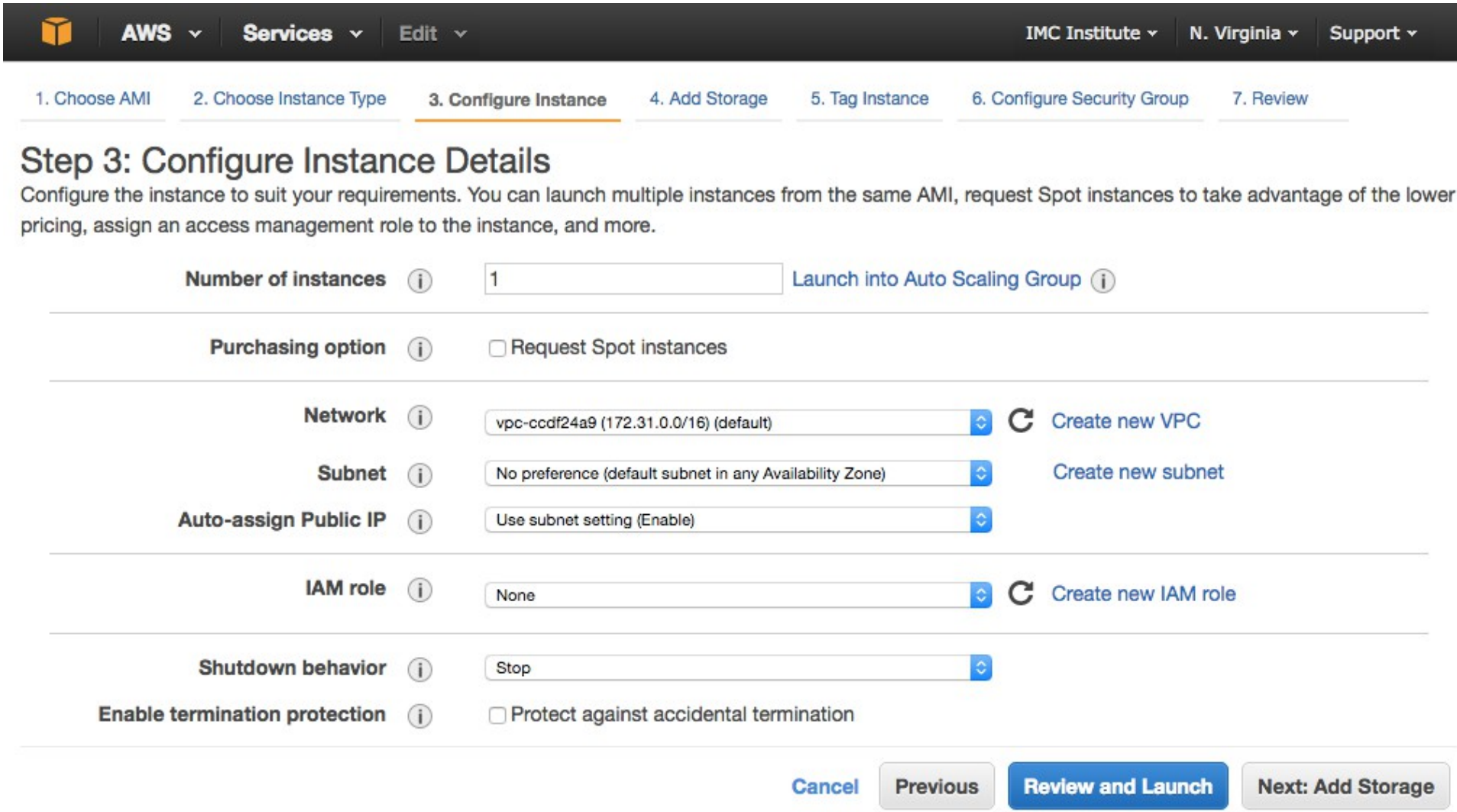


Select an Amazon Machine Image (AMI) and Ubuntu Server 14.04 LTS (PV)

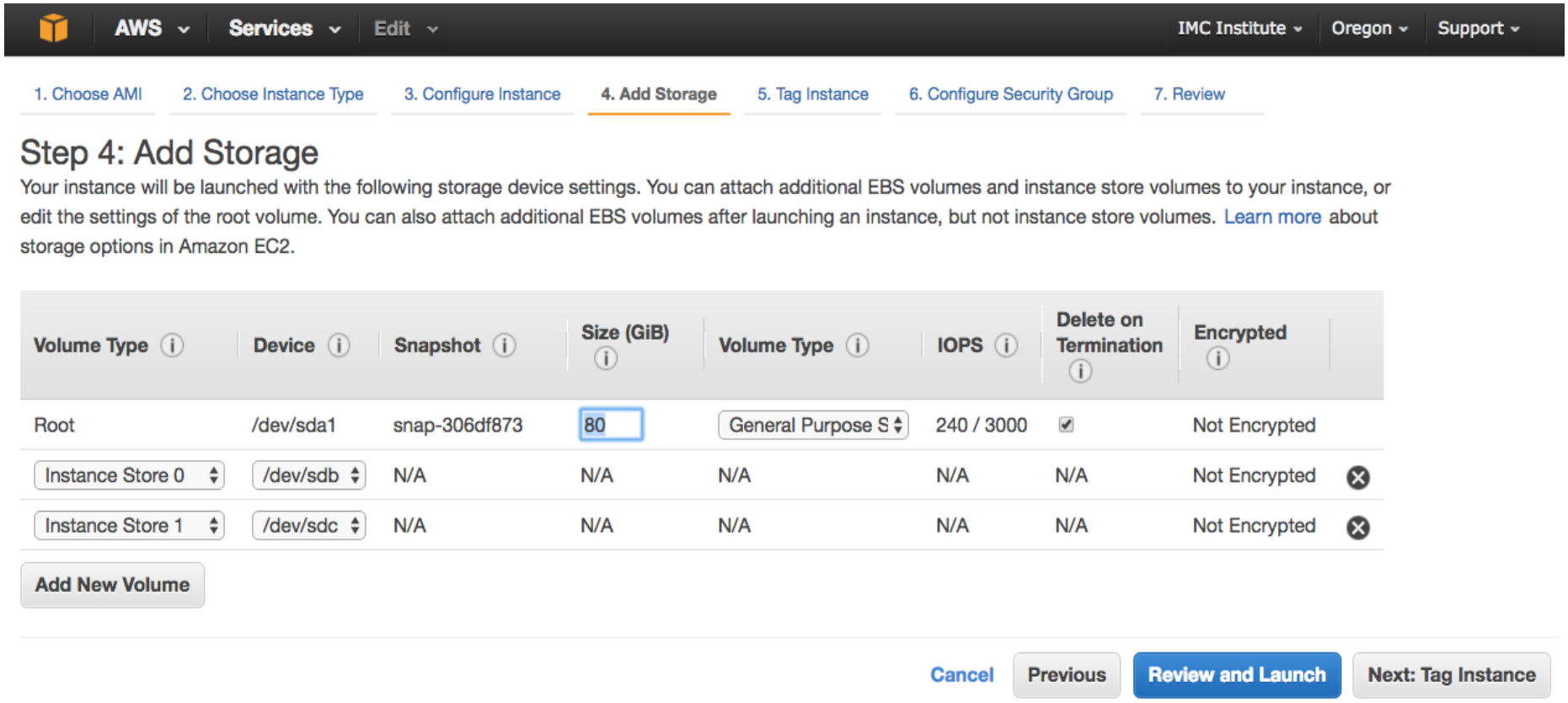


Choose m3.xlarge Type virtual server

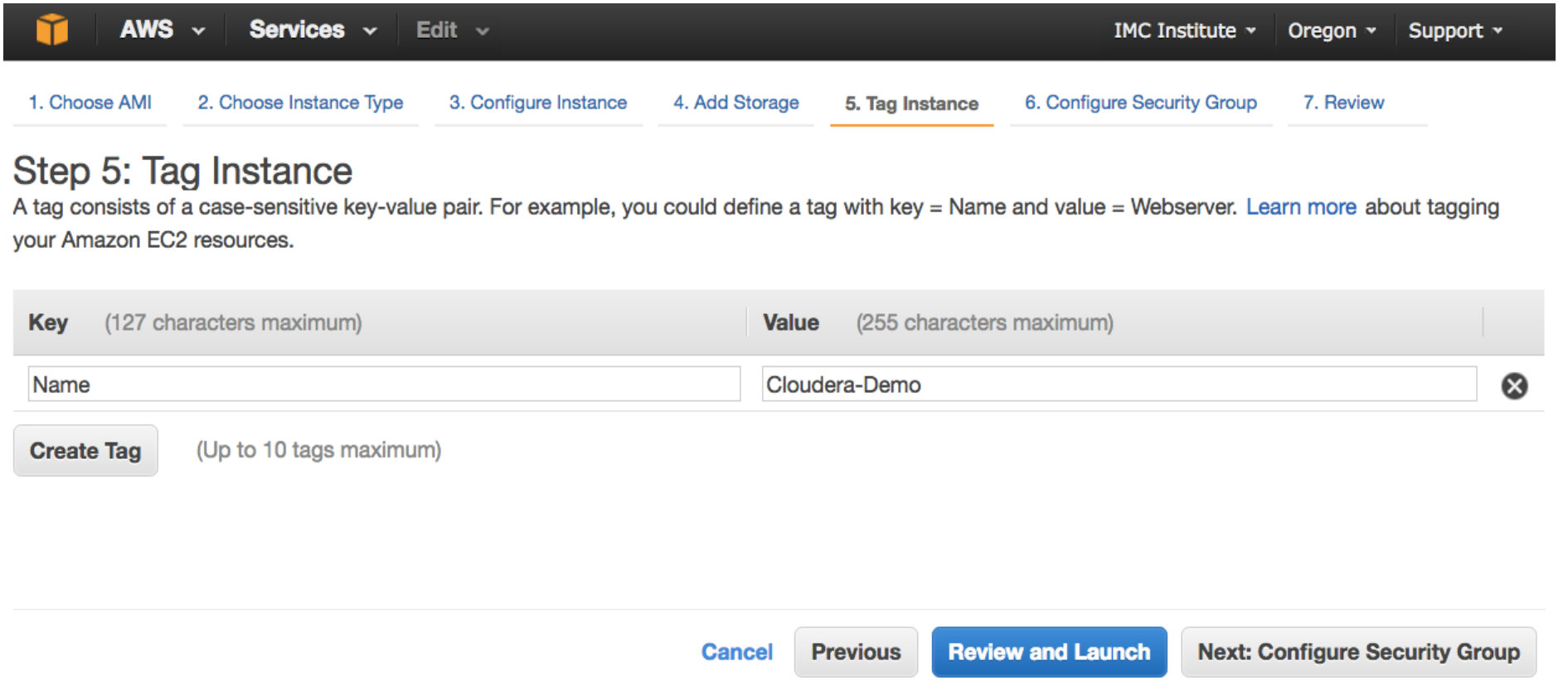



AWS
Services
Edit
IMC Institute
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Tag Instance
6. Configure Security Group
7. Review
Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances
1
Launch into Auto Scaling Group
Purchasing option
Request Spot instances
Network
vpc-ccdf24a9 (172.31.0.0/16) (default)
Create new VPC
Subnet
No preference (default subnet in any Availability Zone)
Create new subnet
Auto-assign Public IP
Use subnet setting (Enable)
IAM role
None
Create new IAM role
Shutdown behavior
Stop
Enable termination protection
Protect against accidental termination
Cancel
Previous
Review and Launch
Next: Add Storage

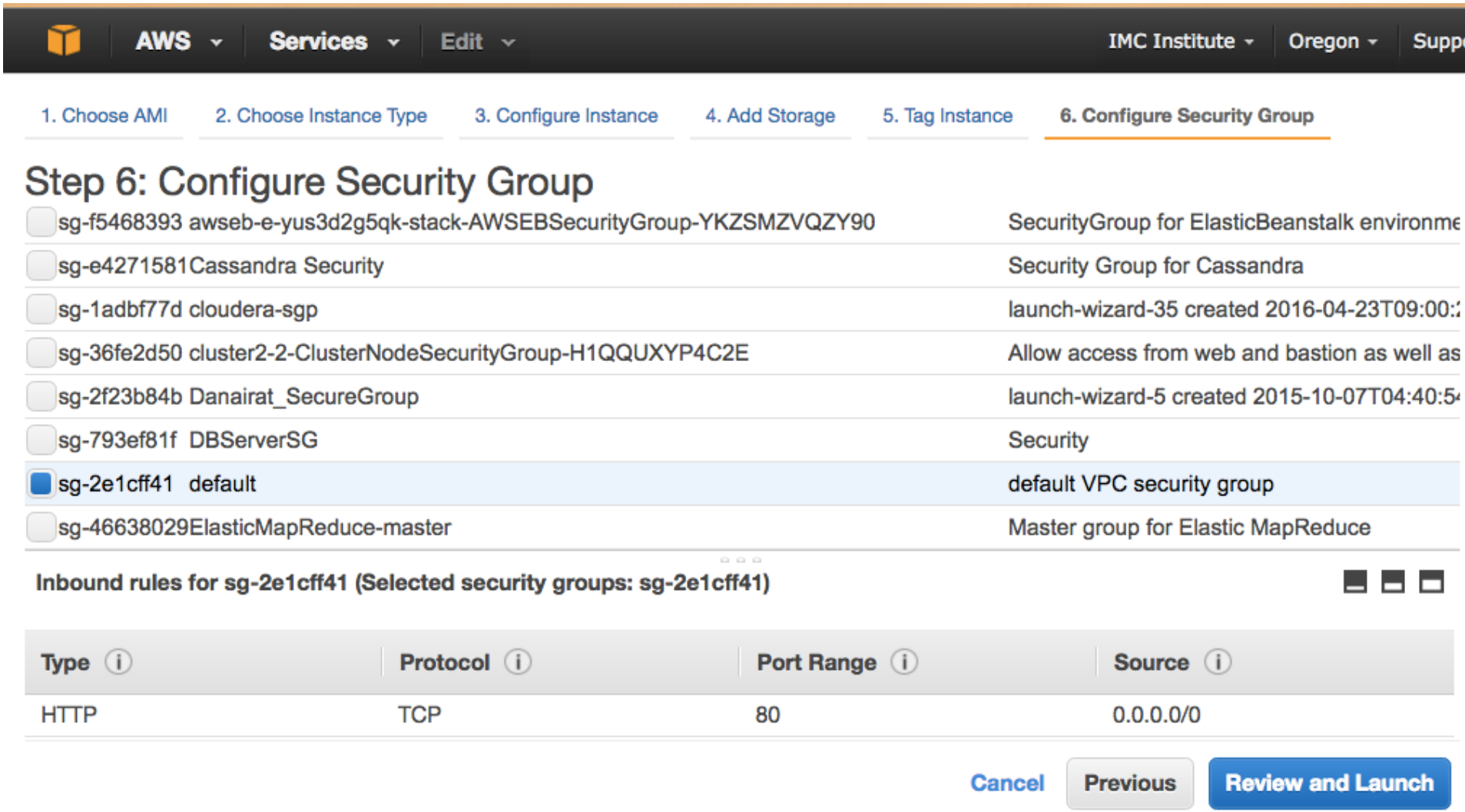
Add Storage: 80 GB



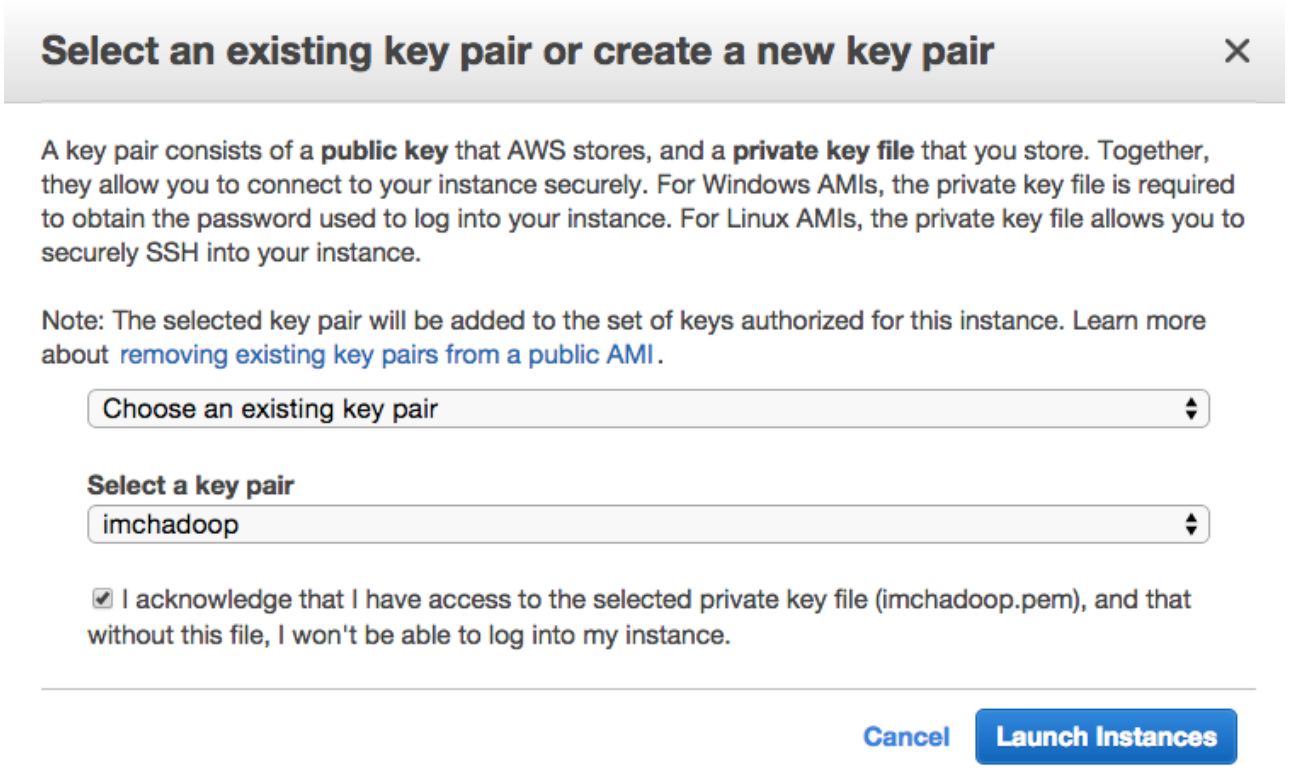
Name the instance



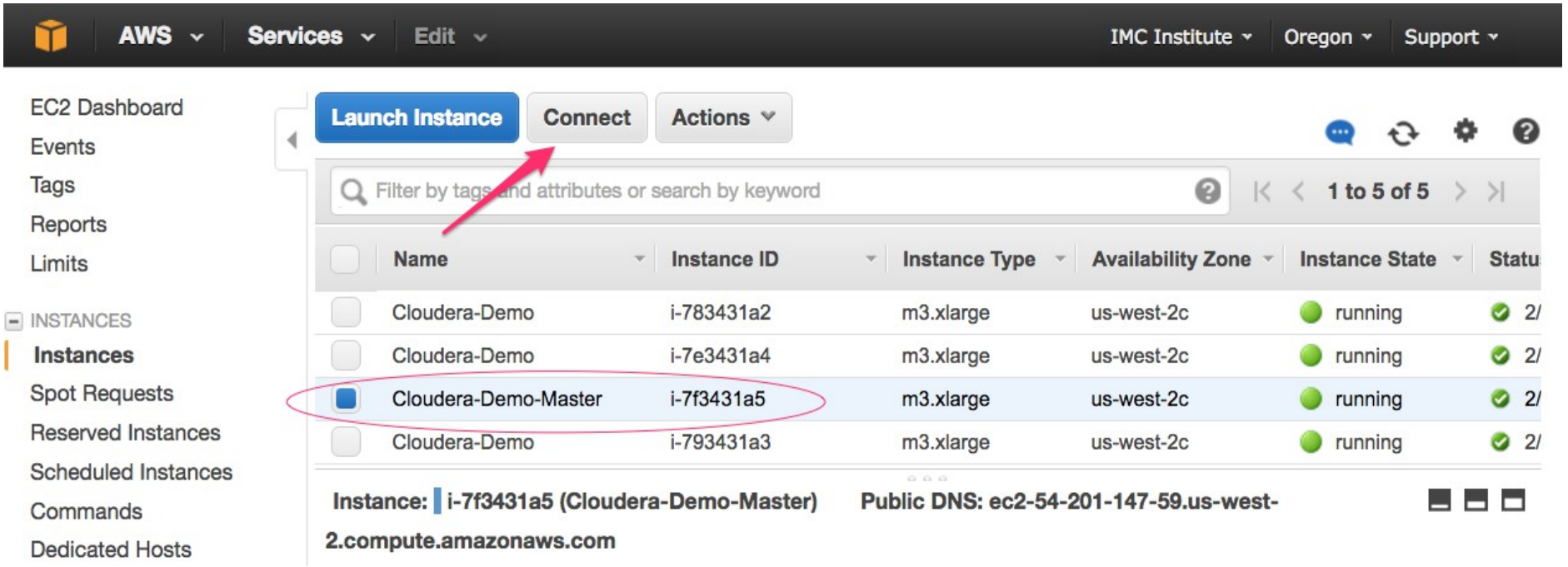
Select Create an existing security group > Default



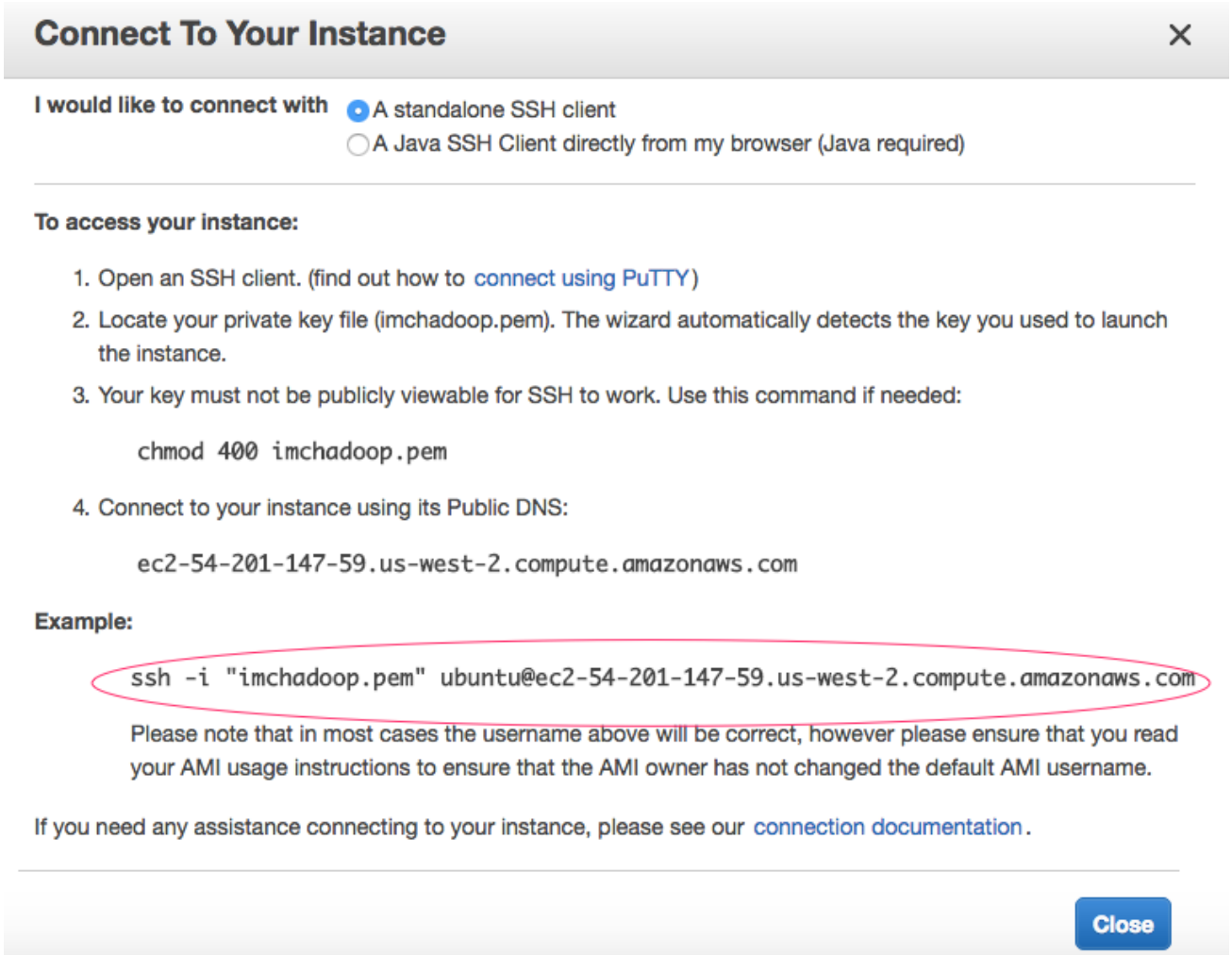
Click Launch and choose imchadoop as a key pair



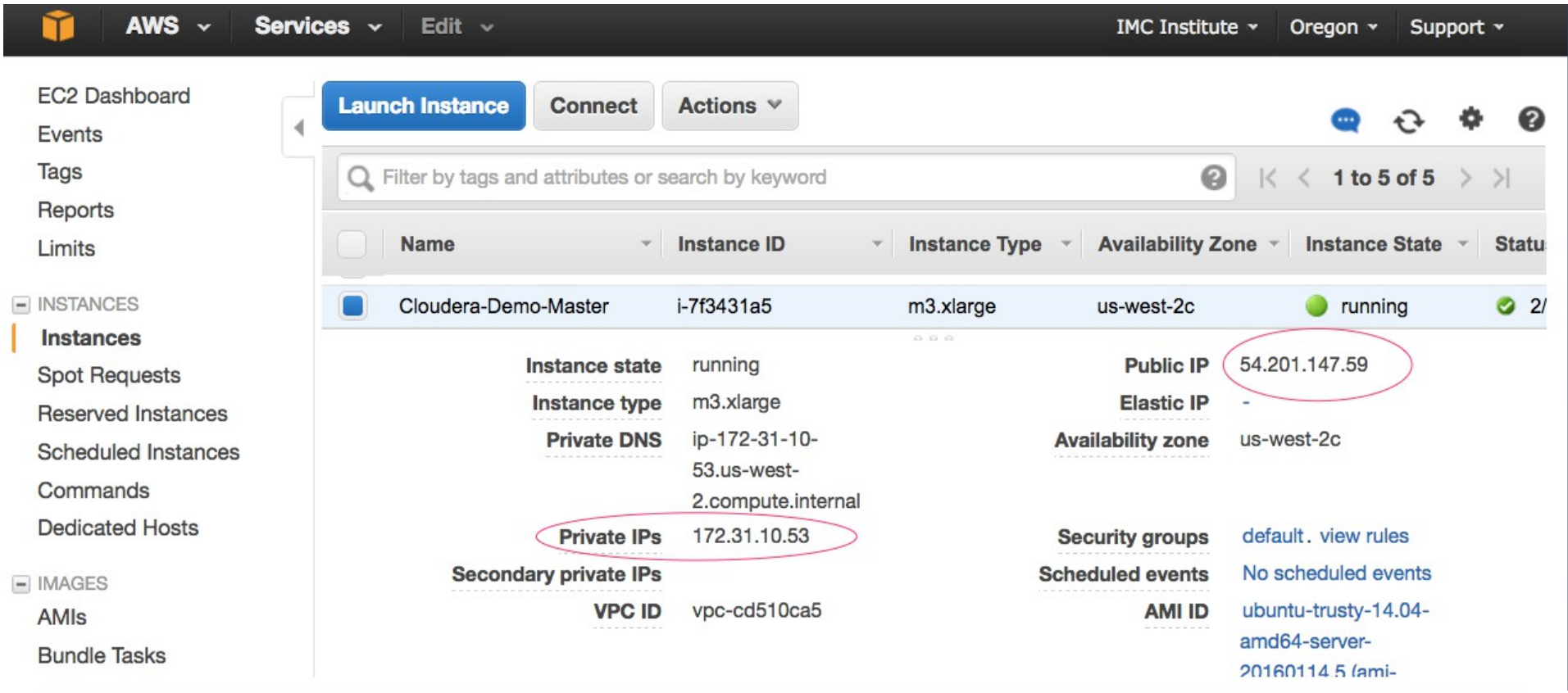
Review an instance and rename one instance as a master / click Connect for an instruction to connect to the instance



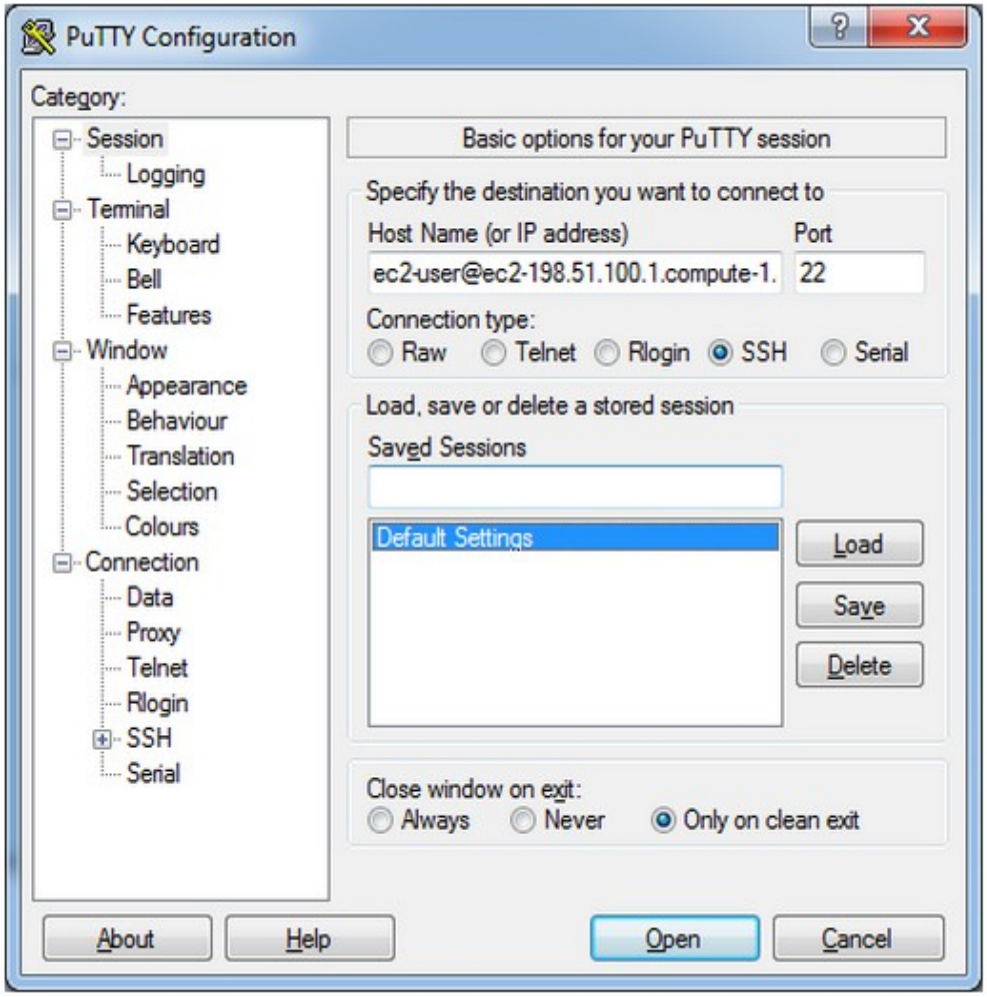
Connect to an instance from Mac/Linux



Can also view details of the instance such as Public IP and Private IP



Connect to an instance from Windows using Putty



Connect to the instance

```
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

_____________________________________________________________________
WARNING! Your environment specifies an invalid locale.
 This can affect your user experience significantly, including the
 ability to manage packages. You may install the locales by running:

   sudo apt-get install language-pack-UTF-8
     or
   sudo locale-gen UTF-8

To see all available language packs, run:
   apt-cache search "^language-pack-[a-z][a-z]$"
To disable this message for all users, run:
   sudo touch /var/lib/cloud/instance/locale-check.skip
_____________________________________________________________________

ubuntu@ip-172-31-1-242:~$
```

Hands-On: Installing Cloudera Quickstart on Docker Container

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Installation Steps

- Update OS
- Install Docker
- Pull Cloudera Quickstart
- Run Cloudera Quickstart
- Run Cloudera Manager

Hive.apache.org

Update OS (Ubuntu)

- Command: sudo apt-get update

```
ubuntu@ip-172-31-30-238:~$ sudo apt-get update
Ign http://us-east-1.ec2.archive.ubuntu.com trusty InRelease
Get:1 http://us-east-1.ec2.archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com trusty-backports InRelease [65.9 kB]
Hit http://us-east-1.ec2.archive.ubuntu.com trusty Release.gpg
Hit http://us-east-1.ec2.archive.ubuntu.com trusty Release
Get:3 http://security.ubuntu.com trusty-security InRelease [65.9 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com trusty-updates/main Sources [277 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com trusty-updates/restricted Sources [535
2 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com trusty-updates/universe Sources [156 k
B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com trusty-updates/multiverse Sources [593
9 B]
Get:8 http://us-east-1.ec2.archive.ubuntu.com trusty-updates/main amd64 Packages [78
1 kB]
```

Docker Installation

- Command: sudo apt-get install docker.io

```
ubuntu@ip-172-31-30-238:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  aufs-tools cgroup-lite git git-man liberror-perl
Suggested packages:
  btrfs-tools debootstrap lxc rinse git-daemon-run git-daemon-sysvinit git-doc
  git-el git-email git-gui gitk gitweb git-arch git-bzr git-cvs git-mediawiki
  git-svn
The following NEW packages will be installed:
  aufs-tools cgroup-lite docker.io git git-man liberror-perl
0 upgraded, 6 newly installed, 0 to remove and 84 not upgraded.
Need to get 8150 kB of archives.
After this operation, 51.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ trusty/universe aufs-tools amd
64 1:3.2+20130722-1.1 [92.3 kB]
```

Pull Cloudera Quickstart

- Command: sudo docker pull cloudera/quickstart:latest

```
ubuntu@ip-172-31-30-238:~$ sudo docker pull cloudera/quickstart:latest
latest: Pulling from cloudera/quickstart
2cda82941cb7: Already exists
Digest: sha256:f91bee4cdfa2c92ea3652929a22f729d4d13fc838b00f120e630f91c941acb63
Status: Downloaded newer image for cloudera/quickstart:latest
ubuntu@ip-172-31-30-238:~$
```

Show docker images

- Command: sudo docker images

```
ubuntu@ip-172-31-30-238:~$ sudo docker images
REPOSITORY            TAG                 IMAGE ID            CREATED
  VIRTUAL SIZE
cloudera/quickstart   latest              2cda82941cb7        9 weeks ago
  6.336 GB
```

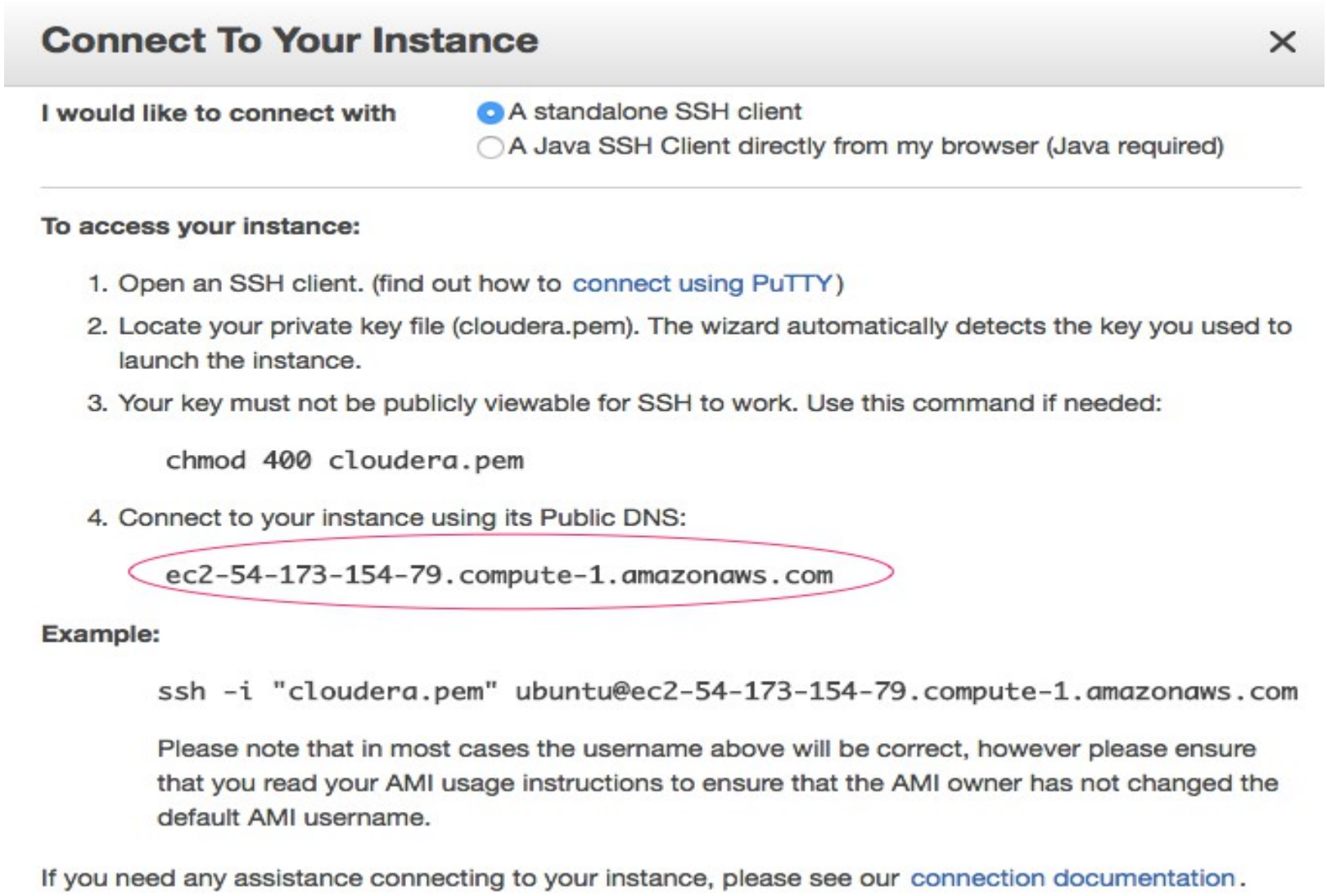
Run Cloudera quickstart

- Command: sudo docker run --hostname=quickstart.cloudera --privileged=true -t -i [OPTIONS] [IMAGE] /usr/bin/docker-quickstart

Example: sudo docker run --hostname=quickstart.cloudera --privileged=true -t -i -p 8888:8888 cloudera/quickstart /usr/bin/docker-quickstart

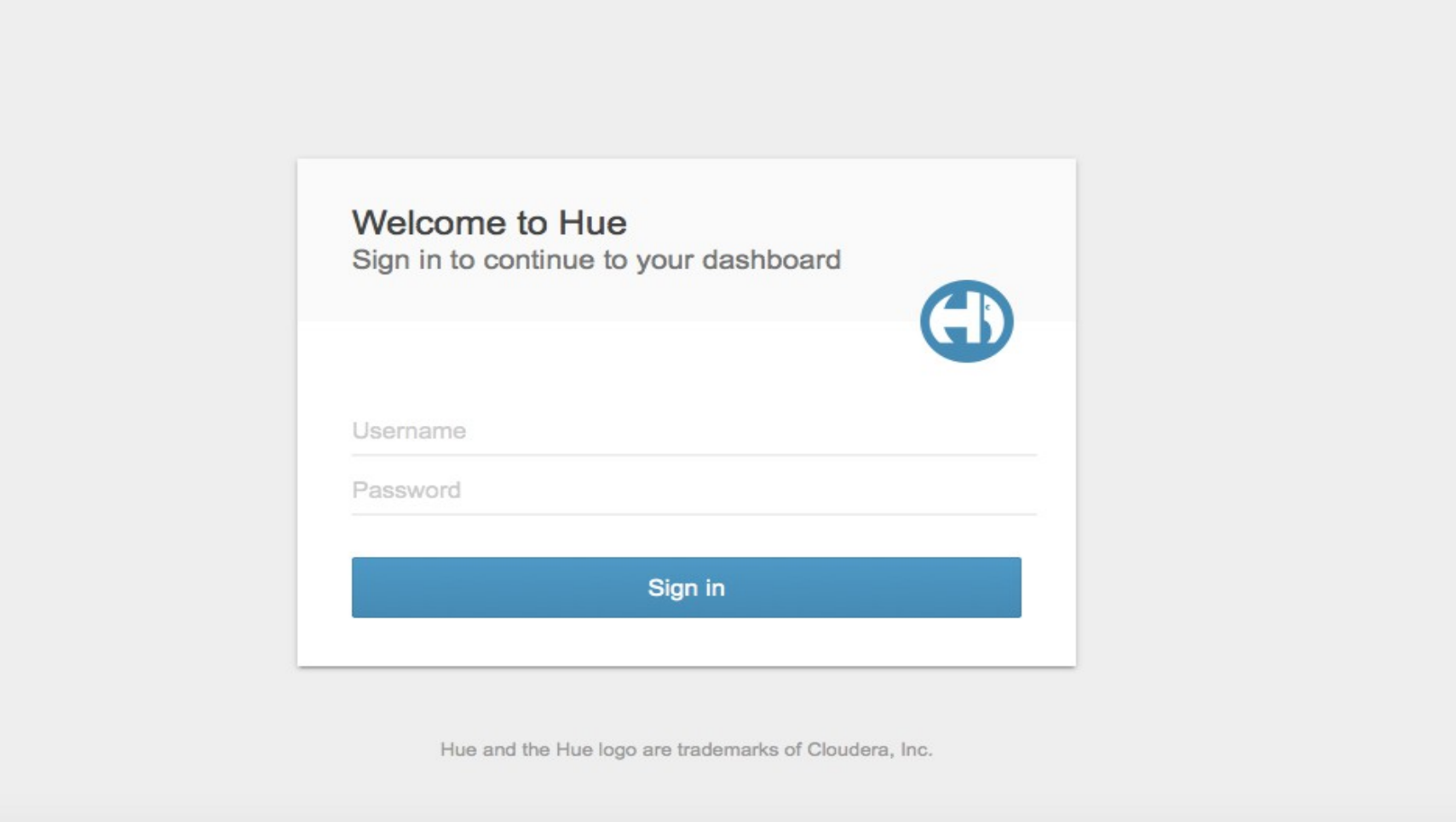
```
ubuntu@ip-172-31-30-238:~$  sudo docker run --hostname=quickstart.cloudera --pri
vileged=true -t -i  -p 8888:8888  -p 7180:7180  cloudera/quickstart  /usr/bin/do
cker-quickstart
Starting mysqld:                                           [  OK  ]

if [ "$1" == "start" ] ; then
    if [ "${EC2}" == 'true' ]; then
        FIRST_BOOT_FLAG=/var/lib/cloudera-quickstart/.ec2-key-installed
        if [ ! -f "${FIRST_BOOT_FLAG}" ]; then
            METADATA_API=http://169.254.169.254/latest/meta-data
            KEY_URL=${METADATA_API}/public-keys/0/openssh-key
```

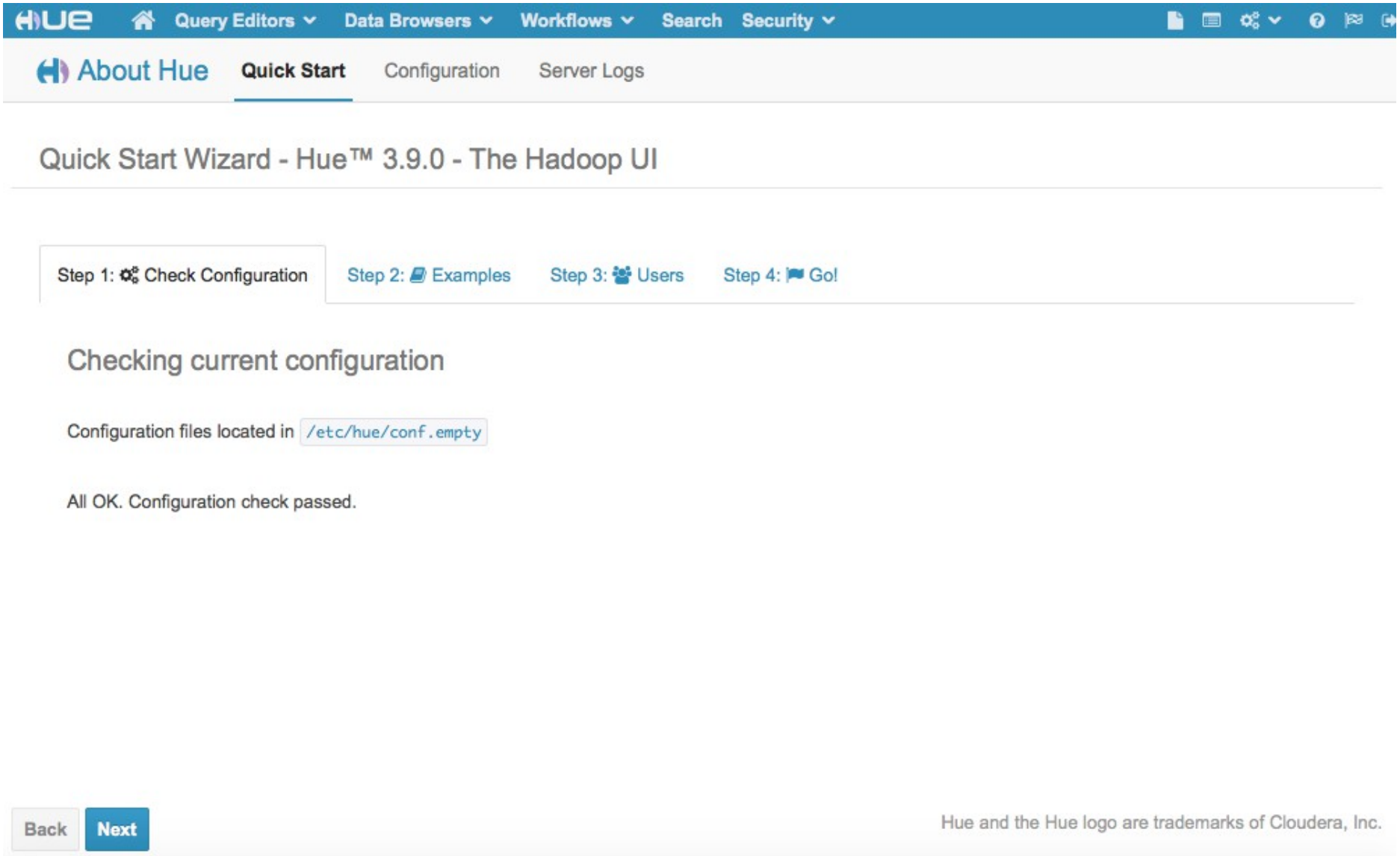
Finding the EC2 instance's DNS



Login to Hue

http://ec2-54-173-154-79.compute-1.amazonaws.com:8888




Query Editors
Data Browsers
Workflows
Search
Security
About Hue
Quick Start
Configuration
Server Logs
Quick Start Wizard - Hue™ 3.9.0 - The Hadoop UI
Step 1: Check Configuration
Step 2: Examples
Step 3: Users
Step 4: Go!
Checking current configuration
Configuration files located in /etc/hue/conf.empty
All OK. Configuration check passed.
Back
Next
Hue and the Hue logo are trademarks of Cloudera, Inc.

Hadoop File System (HDFS)

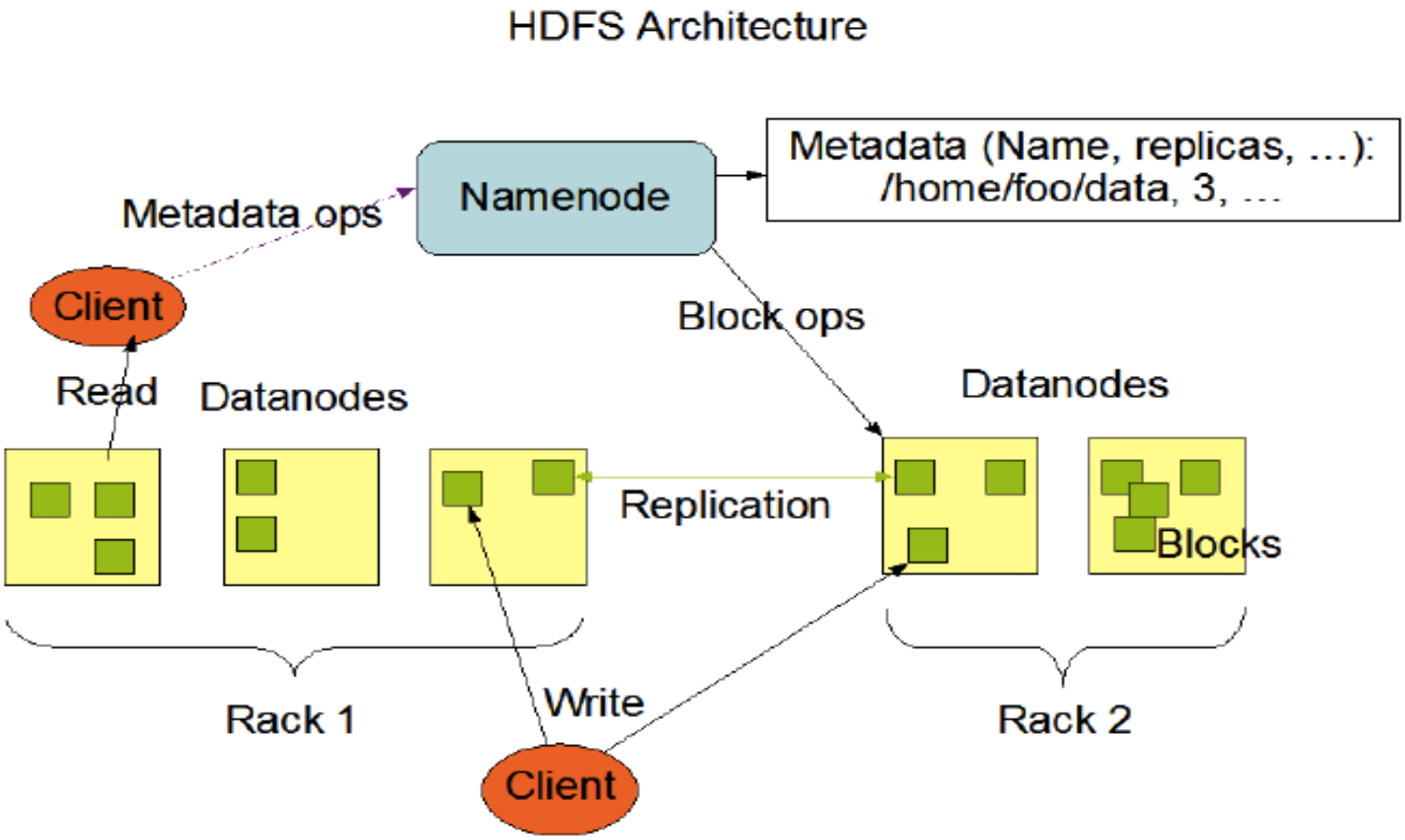
Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

HDFS

- Default storage for the Hadoop cluster
- Data is distributed and replicated over multiple machines
- Designed to handle very large files with straming data access patterns.
- NameNode/DataNode
- Master/slave architecture (1 master 'n' slaves)
- Designed for large files (64 MB default, but configurable) across all the nodes

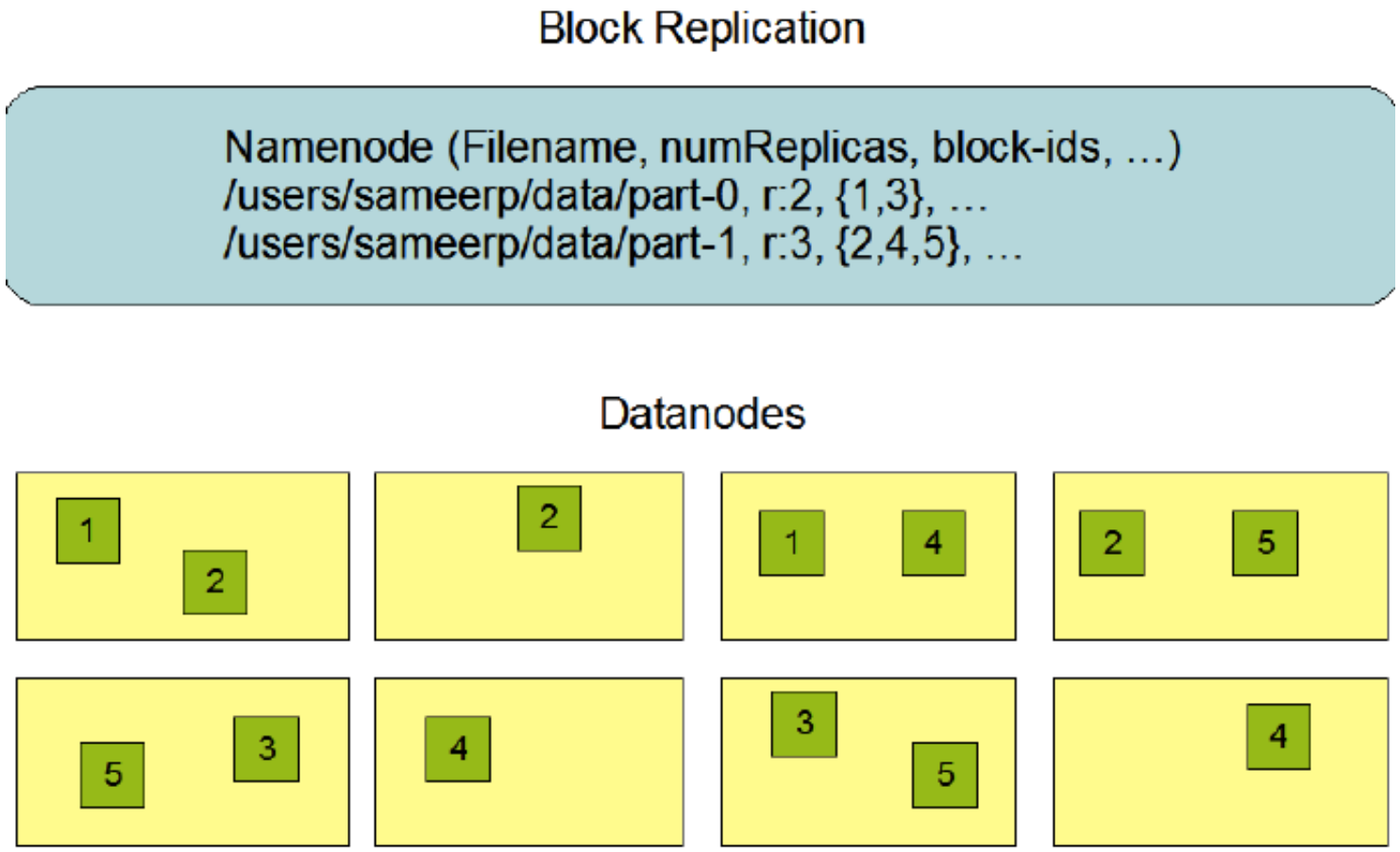
Hive.apache.org

HDFS Architecture



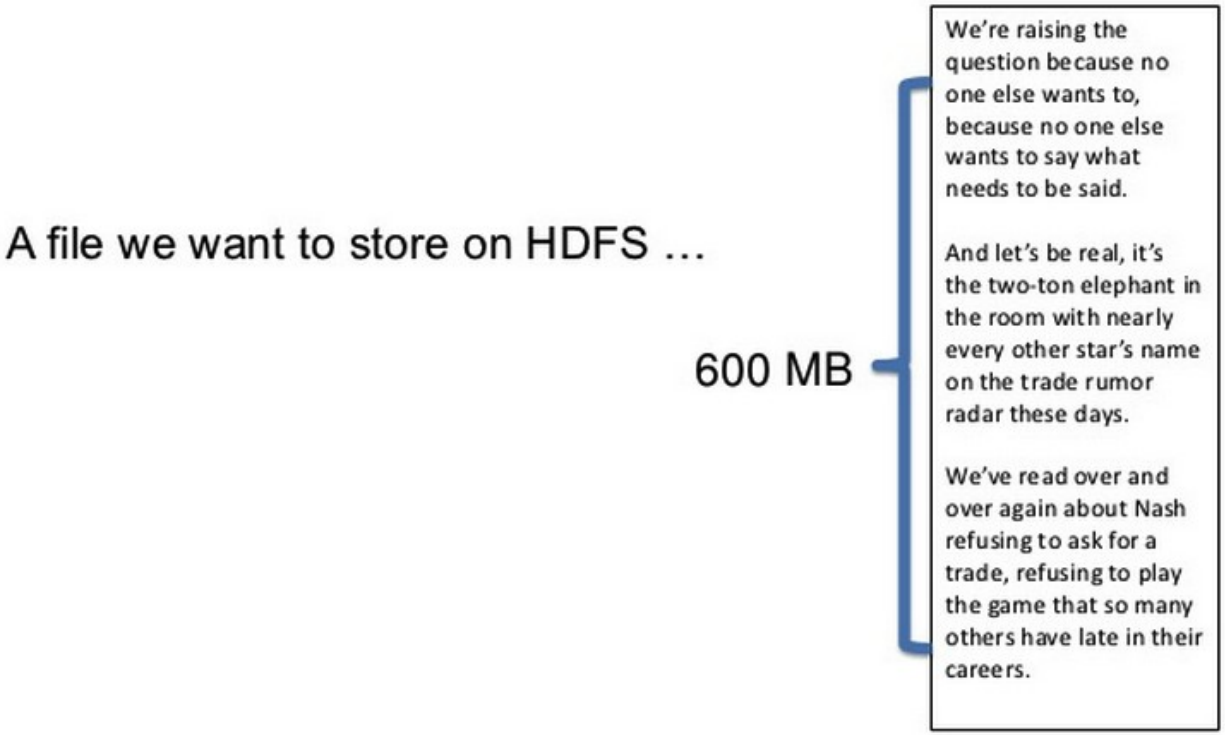
Source Hadoop: Shashwat Shriparv

Data Replication in HDFS



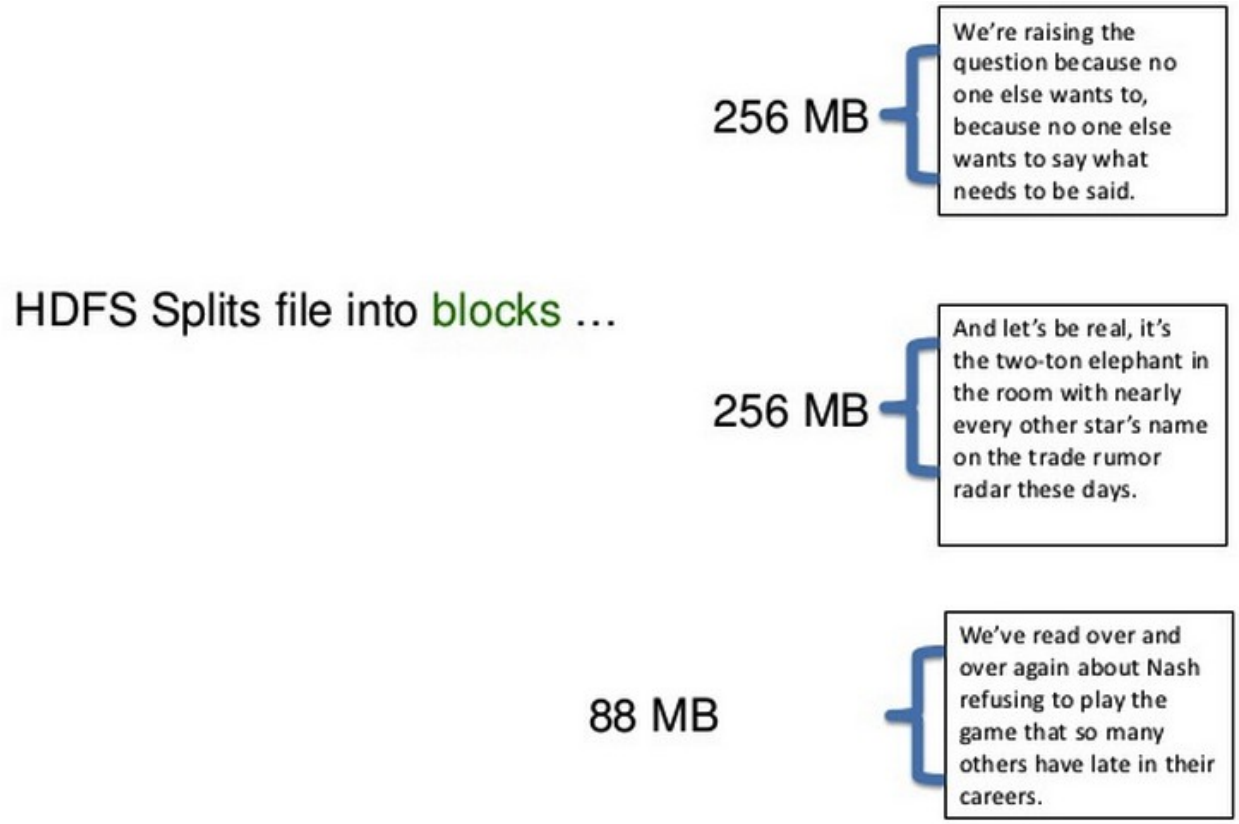
Source Hadoop: Shashwat Shriparv

How does HDFS work?



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

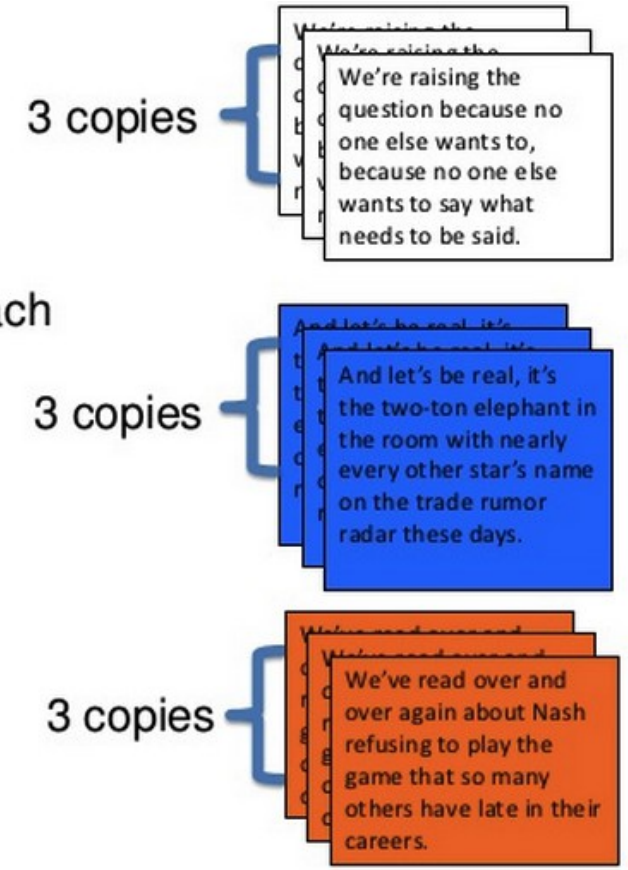
How does HDFS work?



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

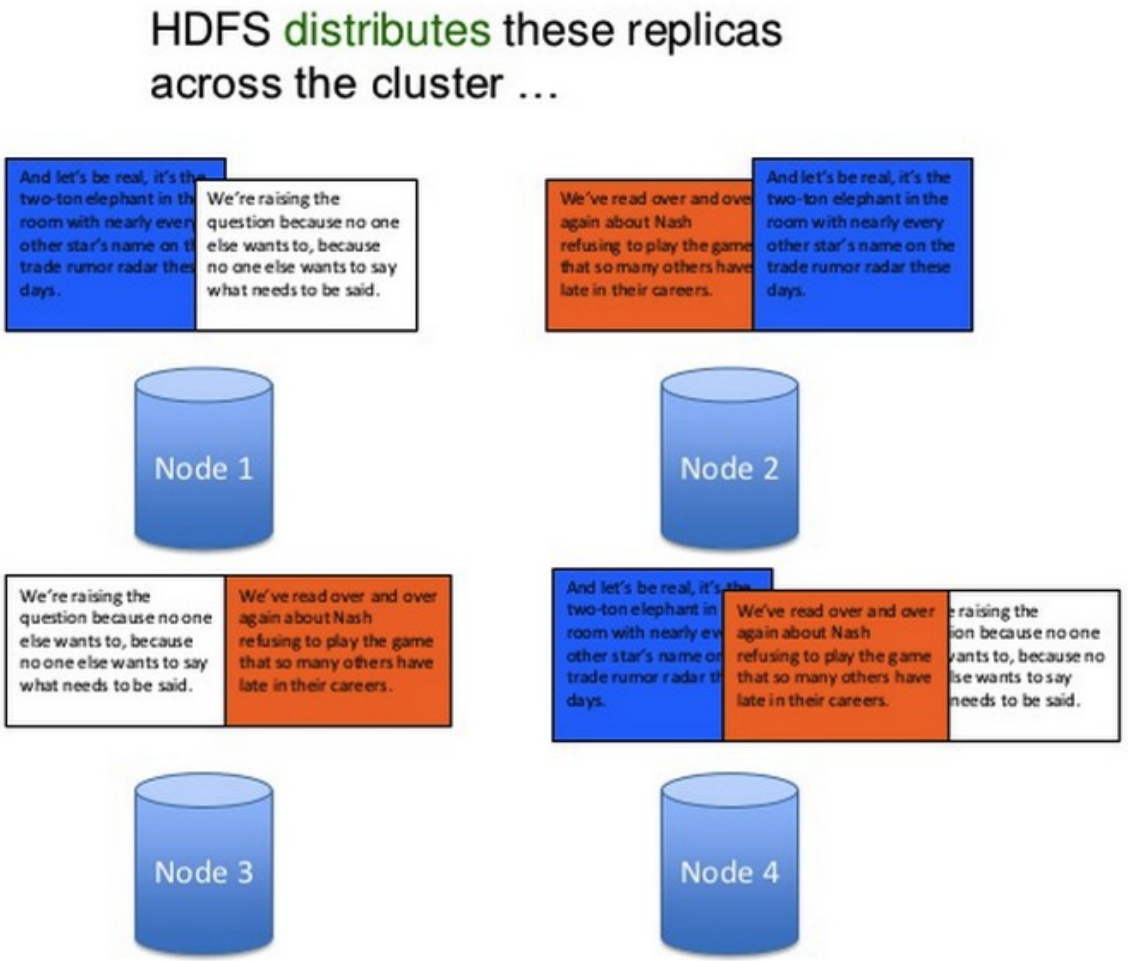
How does HDFS work?

HDFS will create 3replicas of each block …



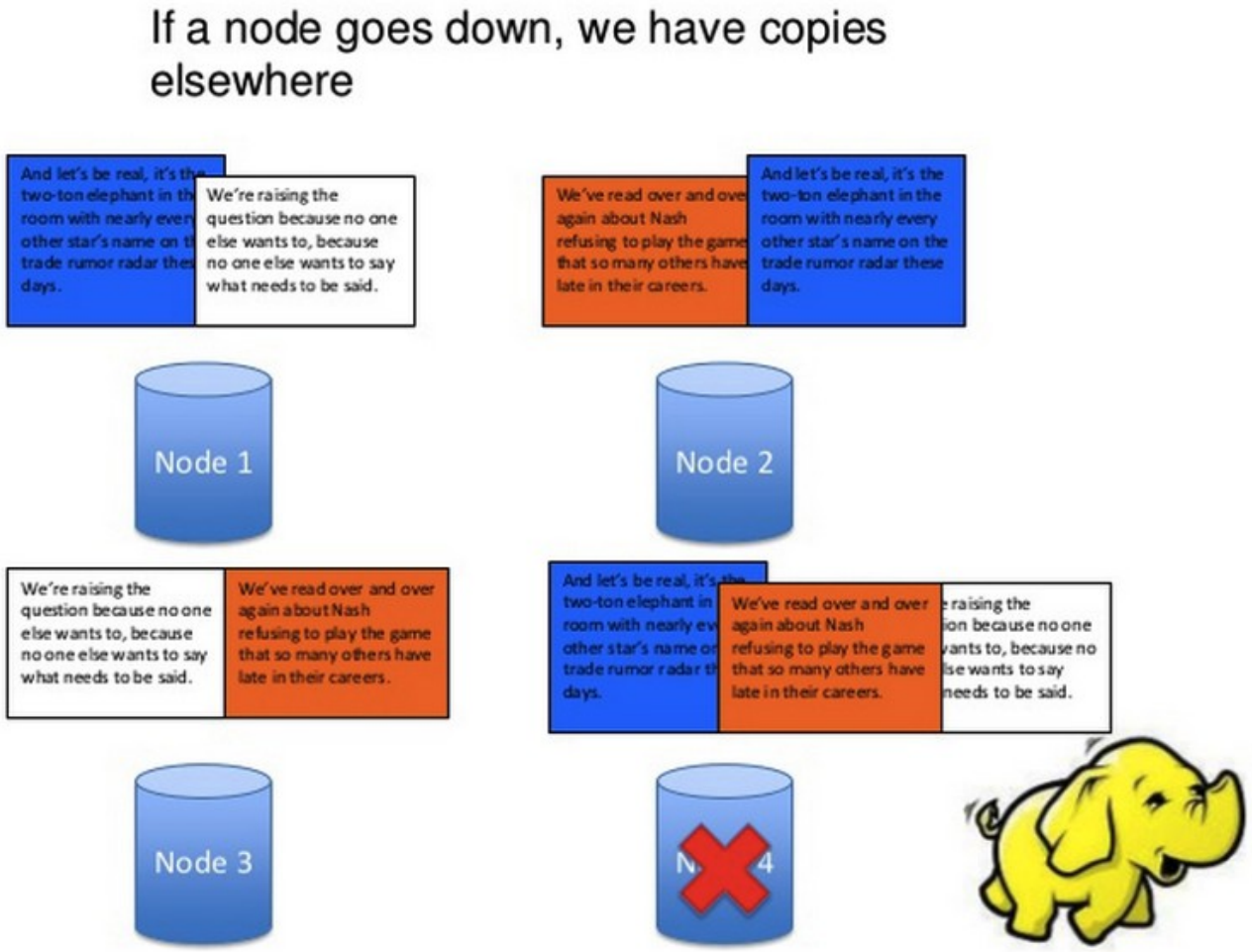
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?

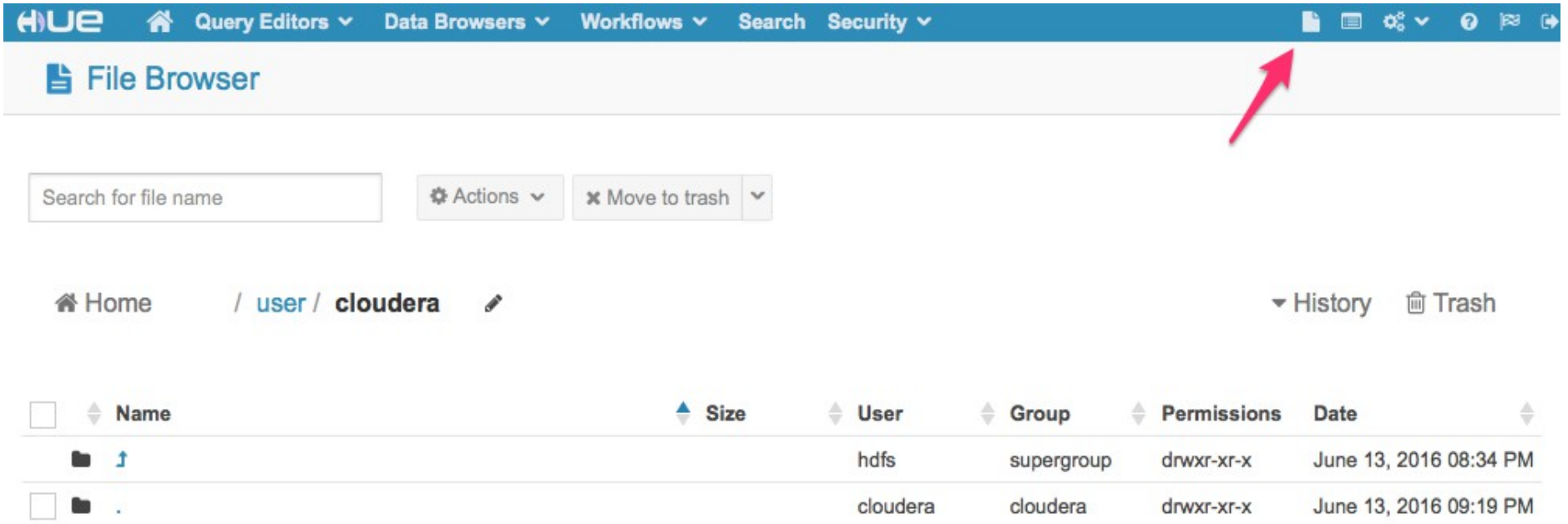


Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

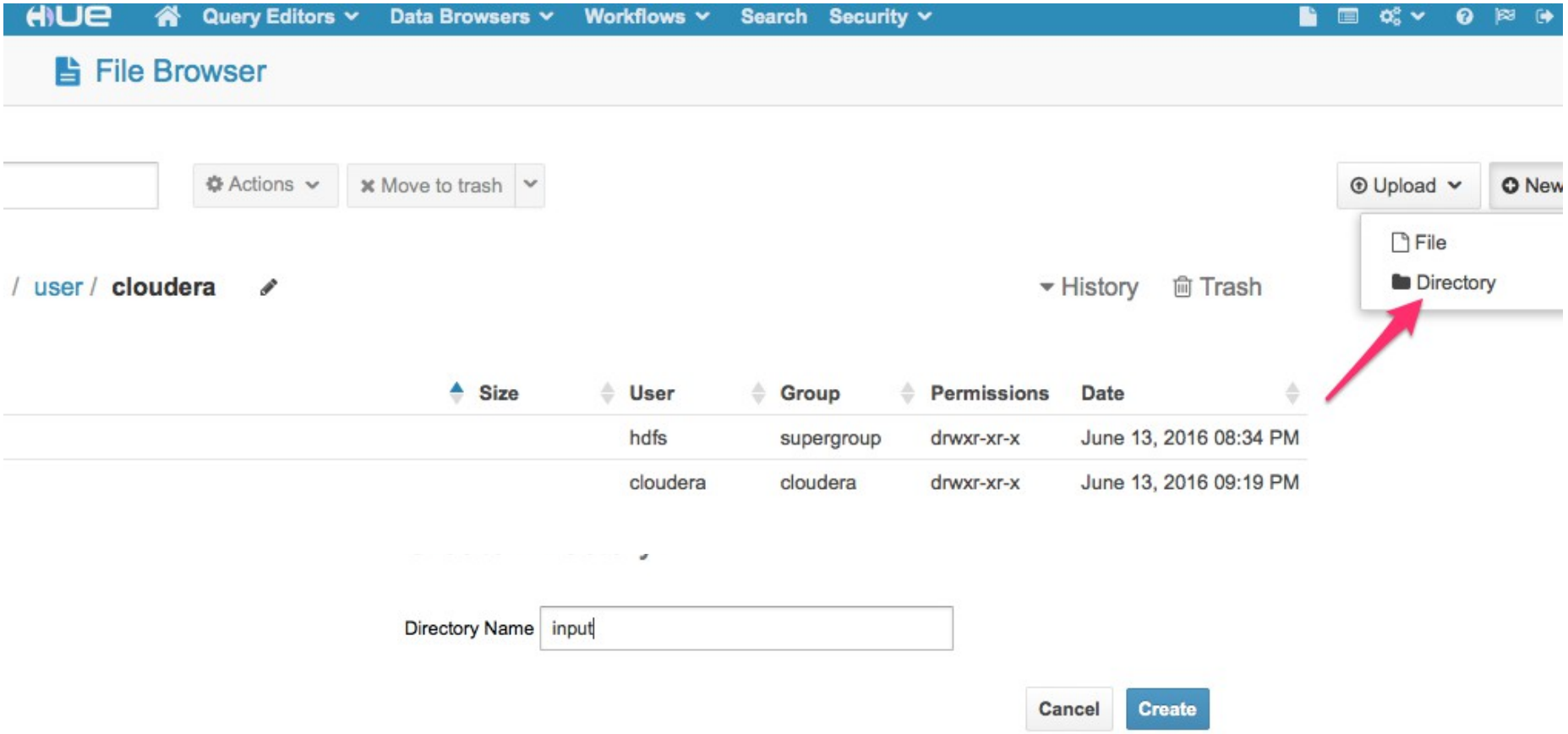
Hands-On: Importing/Exporting Data to HDFS

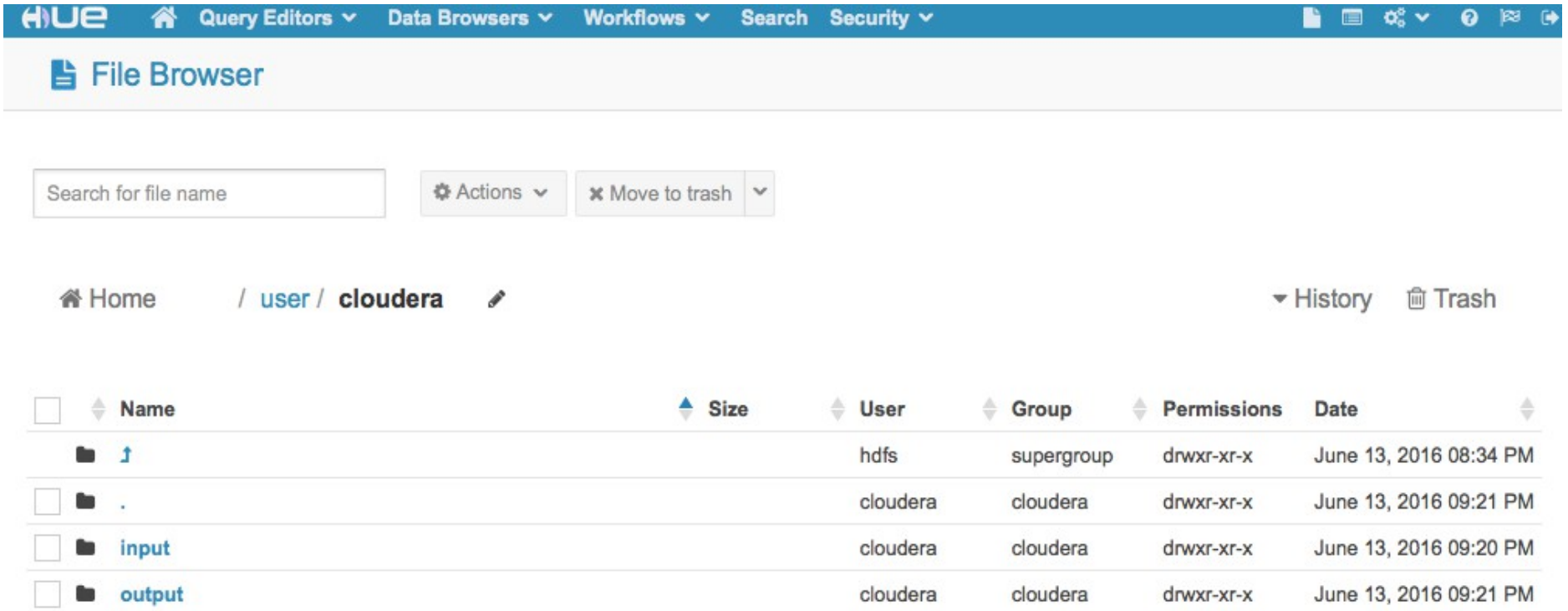
Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Review file in Hadoop HDFS using File Browse

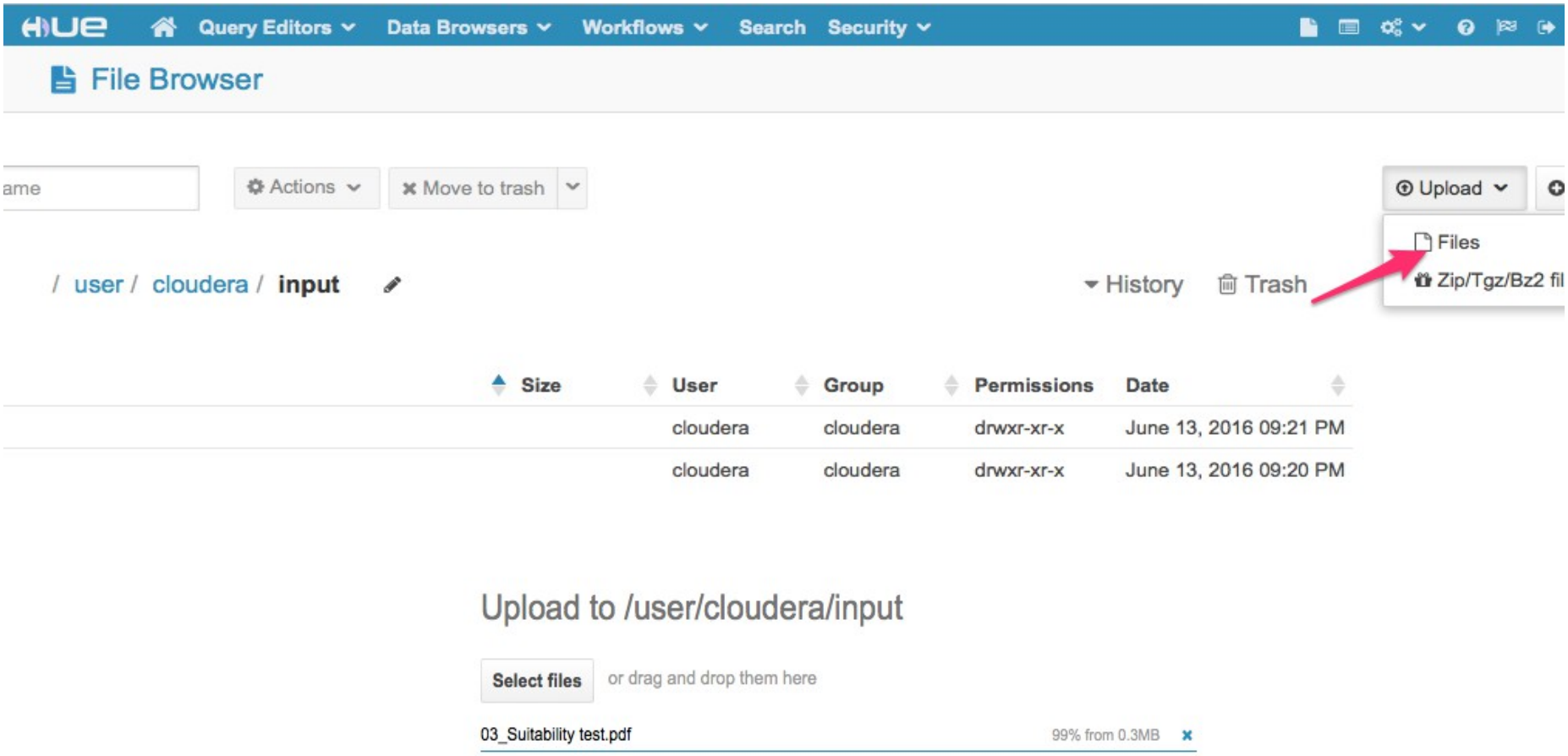


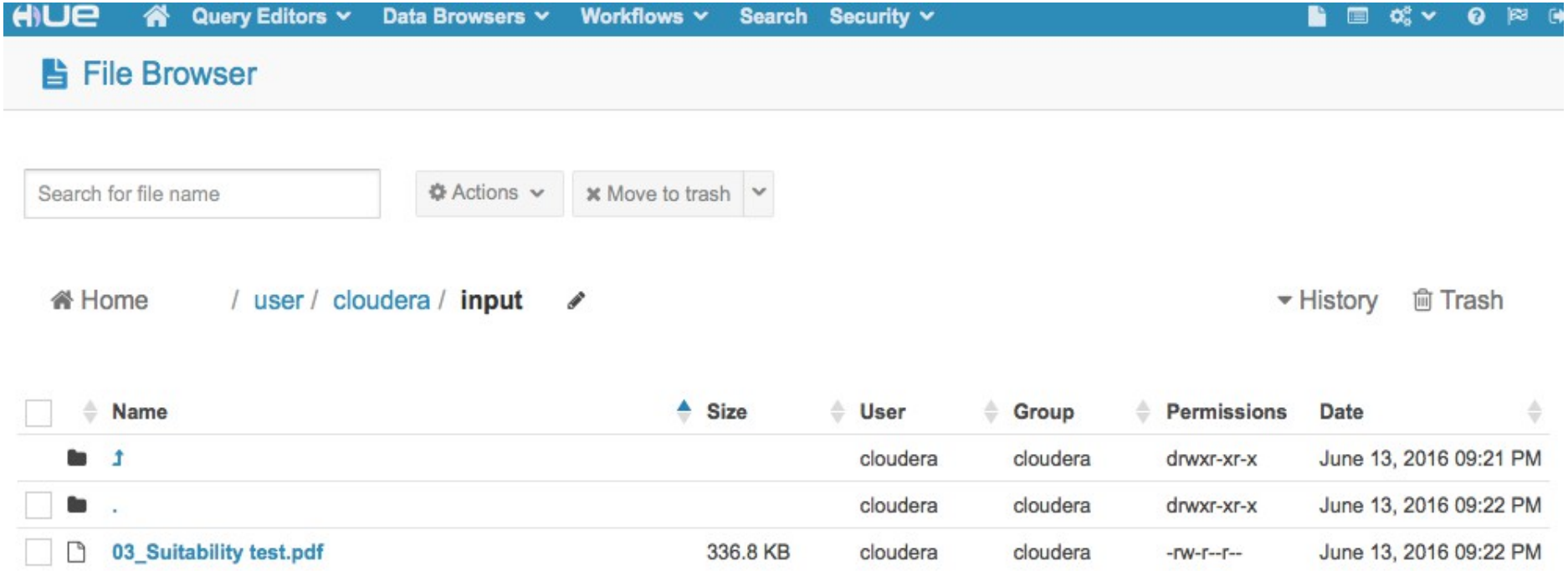
Create a new directory name as: input & output




HUE
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Search for file name
Actions
Move to trash
Home / user / cloudera
History
Trash
Name
Size
User
Group
Permissions
Date
hdfs supergroup drwxr-xr-x June 13, 2016 08:34 PM
. cloudera cloudera drwxr-xr-x June 13, 2016 09:21 PM
input cloudera cloudera drwxr-xr-x June 13, 2016 09:20 PM
output cloudera cloudera drwxr-xr-x June 13, 2016 09:21 PM

Upload a local file to HDFS




HUE
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Search for file name
Actions
Move to trash
Home / user / cloudera / input
History
Trash
Name
Size
User
Group
Permissions
Date
cloudera cloudera drwxr-xr-x June 13, 2016 09:21 PM
. cloudera cloudera drwxr-xr-x June 13, 2016 09:22 PM
03_Suitability test.pdf 336.8 KB cloudera cloudera -rw-r--r-- June 13, 2016 09:22 PM

Hands-On: Connect to a master node via SSH

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

SSH Login to a master node

```
THANACHARTs-MacBook-Air:elastic-mapreduce-cli THANACHART$ ssh -i "imchadoop.pem" ub
untu@ec2-54-201-147-59.us-west-2.compute.amazonaws.com
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-74-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sun Mar 27 09:08:01 UTC 2016

  System load:  0.0                Processes:           135
  Usage of /:   27.6% of 29.40GB   Users logged in:     0
  Memory usage: 24%                IP address for eth0: 172.31.10.53
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud


*** System restart required ***
Last login: Sun Mar 27 09:08:01 2016 from node-io5.pool-125-24.dynamic.totbb.net
ubuntu@ip-172-31-10-53:~$
```


Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Hadoop syntax for HDFS

Command	Syntax
Listing of files in a directory	`hadoop fs -ls /user`
Create a new directory	`hadoop fs -mkdir /user/guest/newdirectory`
Copy a file from a local machine to Hadoop	`hadoop fs -put C:\Users\Administrator\Downloads\localfile.csv /user/rajn/newdirectory/hadoopfile.txt`
Copy a file from Hadoop to a local machine	`hadoop fs –get /user/rajn/newdirectory/hadoopfile.txt C:\Users\Administrator\Desktop\`
Tail last few lines of a large file in Hadoop	`hadoop fs -tail /user/rajn/newdirectory/hadoopfile.txt`
View the complete contents of a file in Hadoop	`hadoop fs -cat /user/rajn/newdirectory/hadoopfile.txt`
Remove a complete directory from Hadoop	`hadoop fs –rm -r /user/rajn/newdirectory`
Check the Hadoop filesystem space utilization	`hadoop fs –du /`

Install wget

- Command: yum install wget

```
[root@quickstart /]# yum install wget
Loaded plugins: fastestmirror
Setting up Install Process
Determining fastest mirrors
epel/metalink                                    |  13 kB     00:00
 * base: mirrors.evowise.com
 * epel: mirror.cogentco.com
 * extras: mirror.us.leaseweb.net
 * updates: mirror.cs.pitt.edu
base                                             | 3.7 kB     00:00
base/primary_db                                  | 4.7 MB     00:06
```

Download an example text file

Make your own durectory at a master node to avoid mixing with others

```
$mkdir guest1
$cd guest1
$wget https://s3.amazonaws.com/imcbucket/input/pg2600.txt
```

```
--2016-03-27 09:58:48--  https://s3.amazonaws.com/imcbucket/input/pg2600.txt
Resolving s3.amazonaws.com (s3.amazonaws.com)... 54.231.19.187
Connecting to s3.amazonaws.com (s3.amazonaws.com)|54.231.19.187|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3291648 (3.1M) [text/plain]
Saving to: 'pg2600.txt'

100%[======================================>] 3,291,648   3.14MB/s   in 1.0s

2016-03-27 09:58:50 (3.14 MB/s) - 'pg2600.txt' saved [3291648/3291648]
```

Upload Data to Hadoop

```
$hadoop fs -ls /user/cloudera/input
$hadoop fs -rm /user/cloudera/input/*
$hadoop fs -put pg2600.txt /user/cloudera/input/
$hadoop fs -ls /user/cloudera/input
```

```
[root@quickstart guest1]# hadoop fs -ls /user/cloudera/input
Found 1 items
-rw-r--r--   1 root cloudera    3291648 2016-06-14 04:29 /user/clou
dera/input/pg2600.txt
[root@quickstart guest1]#
```

Lecture
Understanding HBase

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Introduction

An open source, non-relational, distributed database

HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (, providing BigTable-like capabilities for Hadoop. That is, it provides a fault-tolerant way of storing large quantities of sparse data.

HBase Features

- Hadoop database modelled after Google's Bigtable
- Column oriented data store, known as Hadoop Database
- Support random realtime CRUD operations (unlike HDFS)
- No SQL Database
- Opensource, written in Java
- Run on a cluster of commodity hardware

Hive.apache.org

When to use HBase?

- When you need high volume data to be stored
- Un-structured data
- Sparse data
- Column-oriented data
- Versioned data (same data template, captured at various time, time-elapse data)
- When you need high scalability

Hive.apache.org

Which one to use?

- HDFS
  - Only append dataset (no random write)
  - Read the whole dataset (no random read)
- HBase
  - Need random write and/or read
  - Has thousands of operation per second on TB+ of data
- RDBMS
  - Data fits on one big node
  - Need full transaction support
  - Need real-time query capabilities

Hive.apache.org

HBase vs. RDBMS

	HBase	RDBMS
Hardware architecture	Similar to Hadoop. Clustered commodity hardware. Very affordable.	Typically large scalable multiprocessor systems. Very expensive.
Fault Tolerance	Built into the architecture. Lots of nodes means each is relatively insignificant. No need to worry about individual node downtime.	Requires configuration of the HW and the RDBMS with the appropriate high availability options.
Typical Database Size	Terabytes to Petabytes - hundred of millions to billions of rows.	Gigabytes to Terabytes – hundred of thousands to millions of rows.
Data Layout	A sparse, distributed, persistent, multidimensional sorted map.	Rows or column oriented.
Data Types	Bytes only.	Rich data type support.
Transactions	ACID support on a single row only	Full ACID compliance across rows and tables
Query Language	API primitive commands only, unless combined with Hive or other technology	SQL
Indexes	Row-Key only unless combined with other technologies such as Hive or IBM's BigSQL	Yes
Throughput	Millions of queries per second	Thousands of queries per second

- Given this RDBMS:

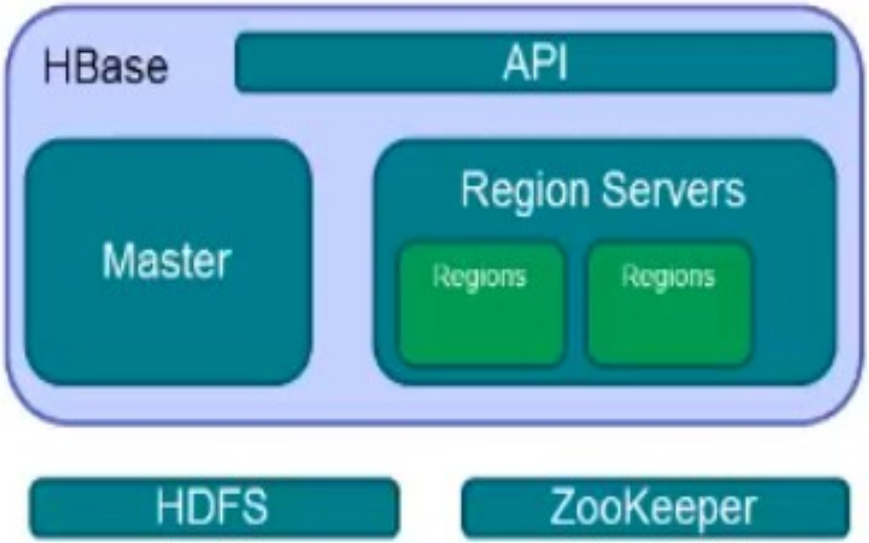
ID (Primary key)	Last name	First name	Password	Timestamp
1234	Smith	John	Hello, world!	20130710
5678	Cooper	Joyce	wysiwyg	20120825
5678	Cooper	Joyce	wisiwig	20130916

- Logical view in HBase:

Row-Key	Value (CF, Qualifier, Version)
1234	info {'lastName': 'Smith', 'firstName': 'John'} pwd {'password': 'Hello, world!'}
5678	info {'lastName': 'Cooper', 'firstName': 'Joyce'} pwd {'password': 'wysiwyg'@ts 20130916, 'password': 'wisiwig'@ts 20120825}

HBase Components

- Region
  - Row of table are stores
- Region Server
  - Hosts the tables
- Master
  - Coordinating the Region Servers
- ZooKeeper
- HDFS
- API
  - The Java Client API



Hive.apache.org

HBase Shell Commands

- See the list of the tables

```
list
```

- Create a table:

```
create 'testTable', 'cf'
```

- Insert data into a table:

  Insert at rowA, column "cf:columnName" with a value of "val1"

```
put 'testTable', 'rowA', 'cf:columnName', 'val1'
```

- Retrieve data from a table:

  Retrive "rowA" from the table "testTable"

```
get 'testTable', 'rowA'
```

- Iterate through a table:

```
- scan 'testTable'
```

- Delete a table:

```
disable 'testTable'
drop 'testTable'
```

Hive.apache.org

Hands-On: Running HBase

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

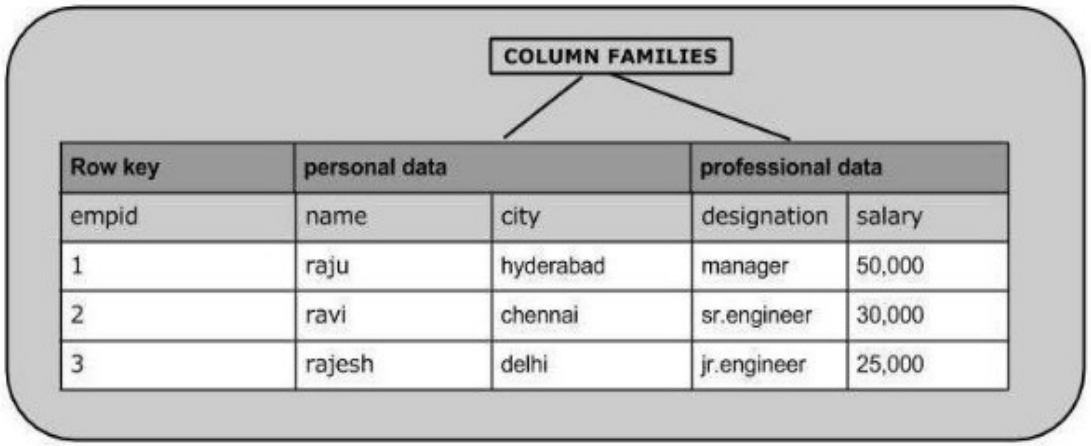
Hbase shell

Row key	personal data	professional data

```
$hbase shell
hbase(main):001:0> create 'employee', 'personal data',
'professional data'
hbase(main):002:0> list
```

```
TABLE
employee
1 row(s) in 0.0310 seconds
```

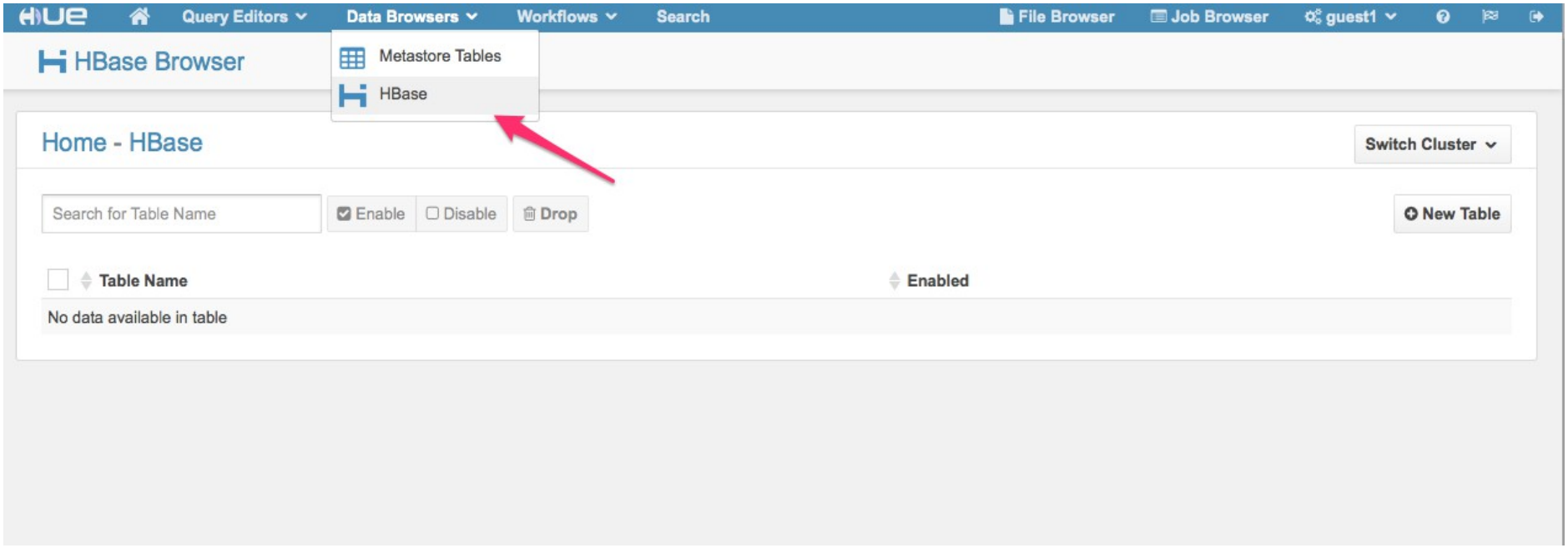
Create Data



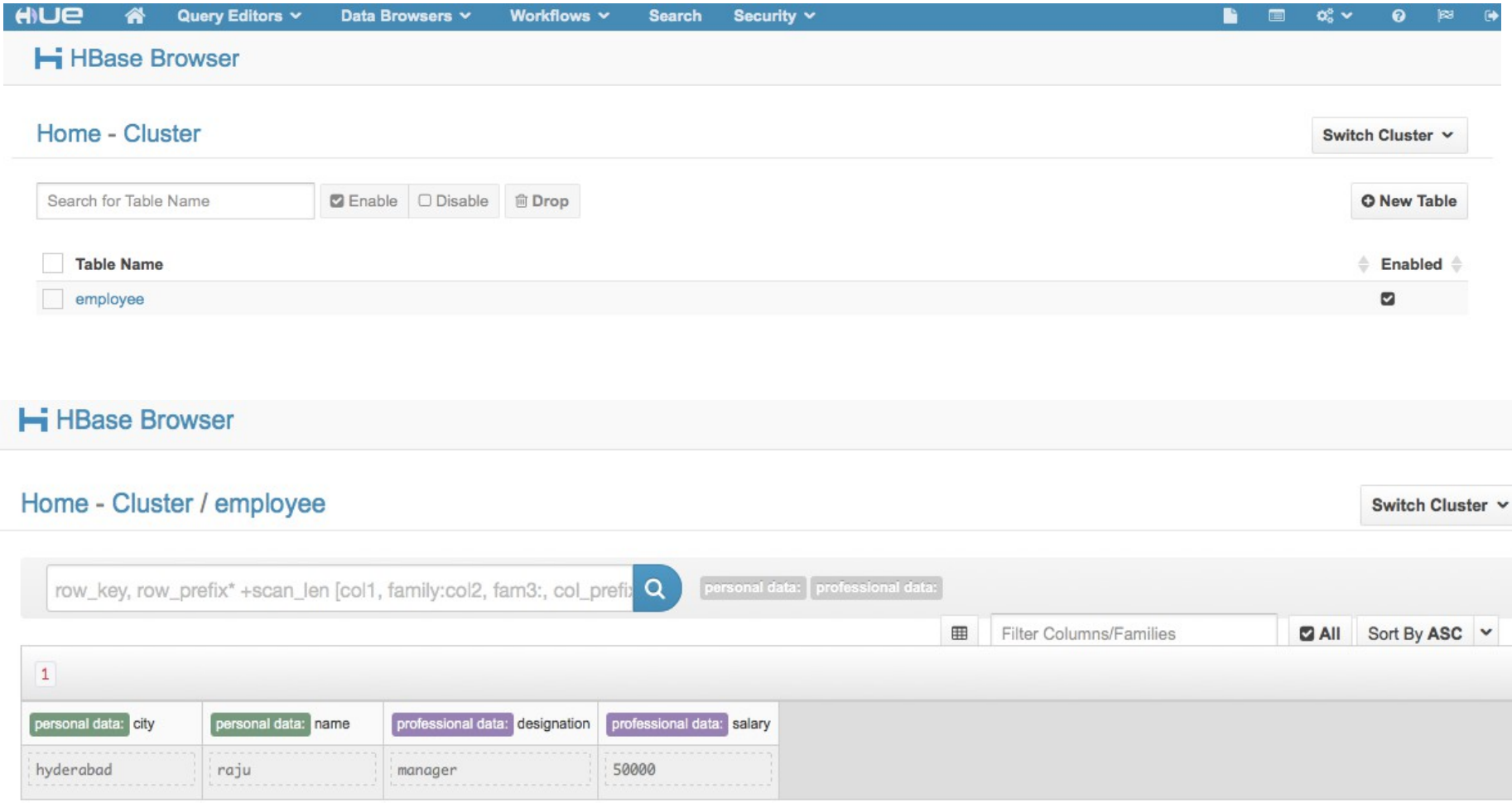
Row key	personal data		professional data	
empid	name	city	designation	salary
1	raju	hyderabad	manager	50,000
2	ravi	chennai	sr.engineer	30,000
3	rajesh	delhi	jr.engineer	25,000

```
hbase(main):010:0> put 'employee','1','personal data:name','raju'
0 row(s) in 0.1720 seconds

hbase(main):011:0> put 'employee','1','personal data:city','hyderabad'
0 row(s) in 0.0140 seconds


hbase(main):018:0> put 'employee','1','professional data:designation','manager'
0 row(s) in 0.0110 seconds

hbase(main):019:0> put 'employee','1','professional data:salary','50000'
0 row(s) in 0.0070 seconds
```

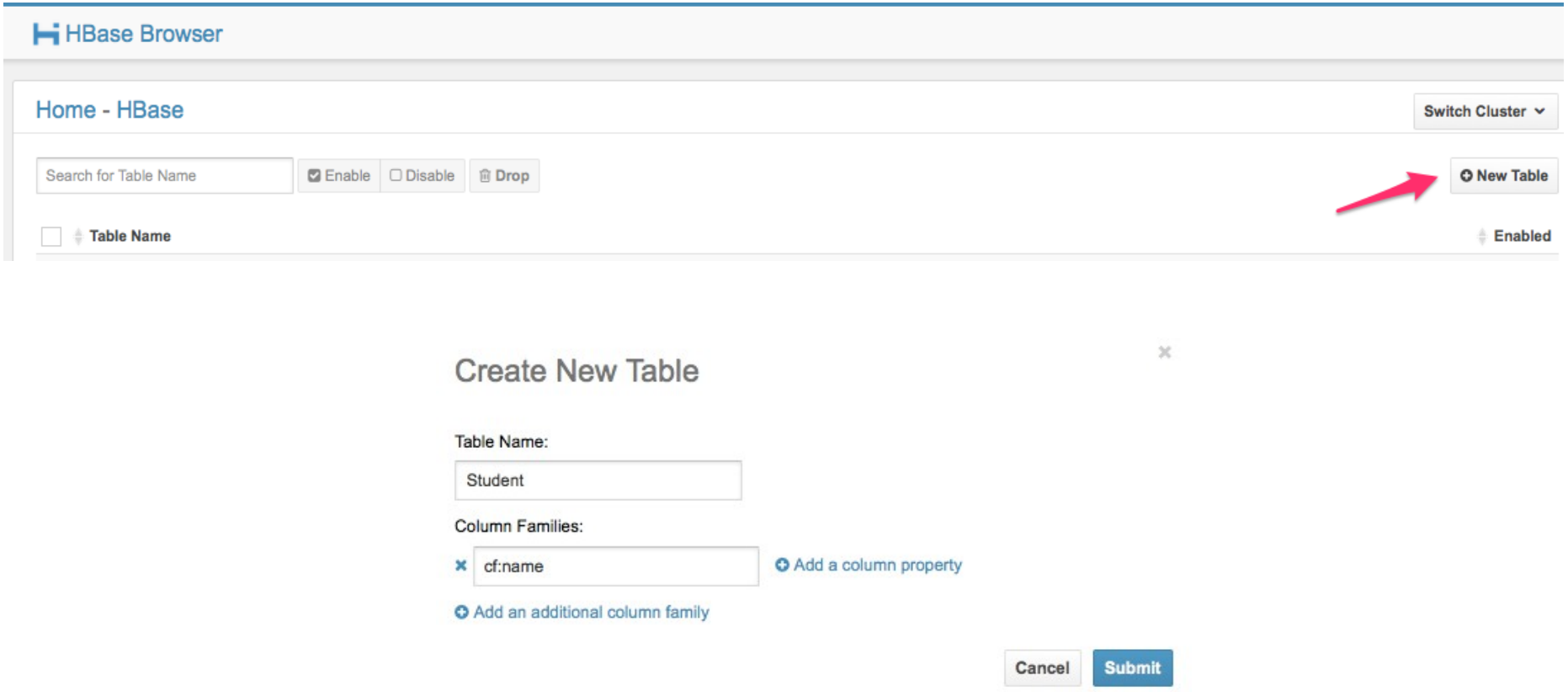
Running HBase Browser



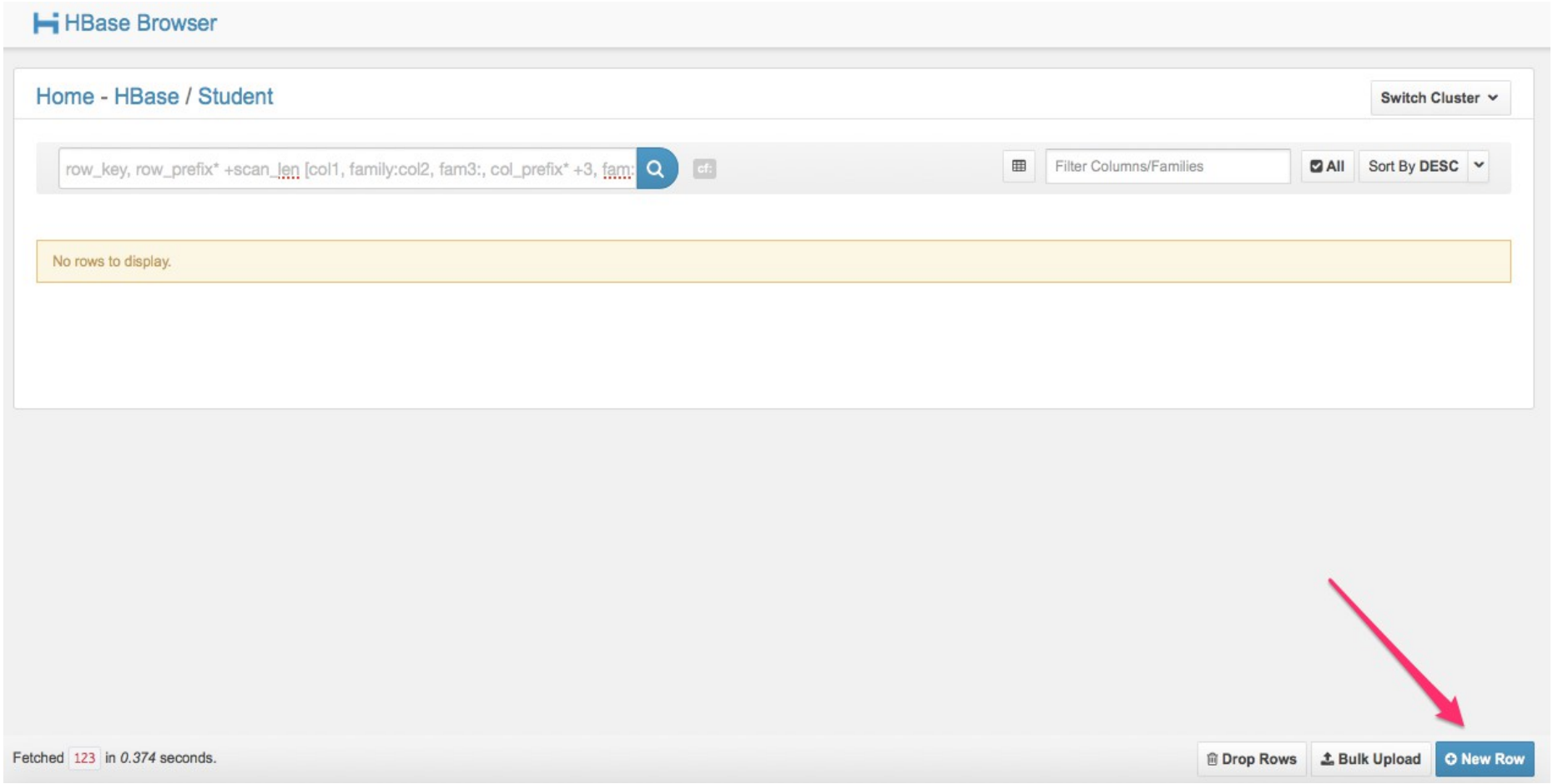
Viewing Employee Table



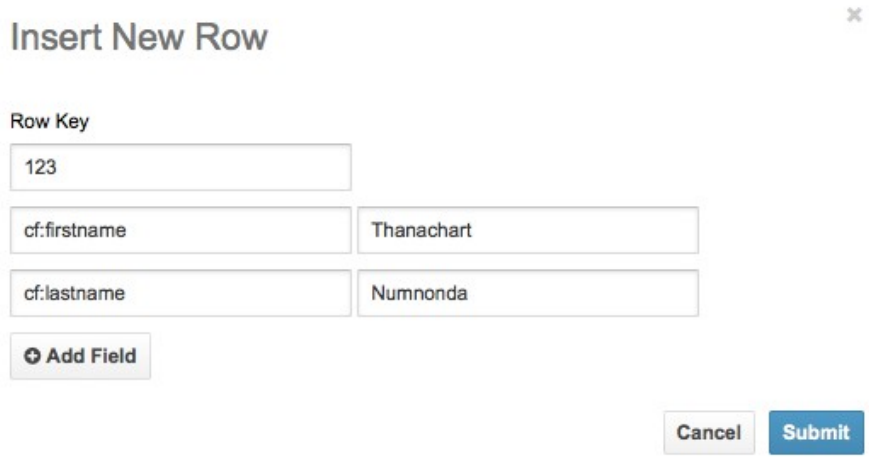
Create a table in HBase

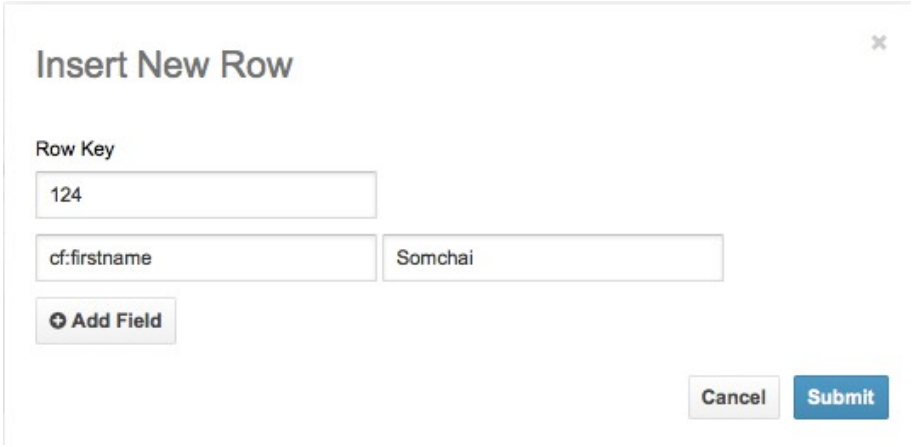


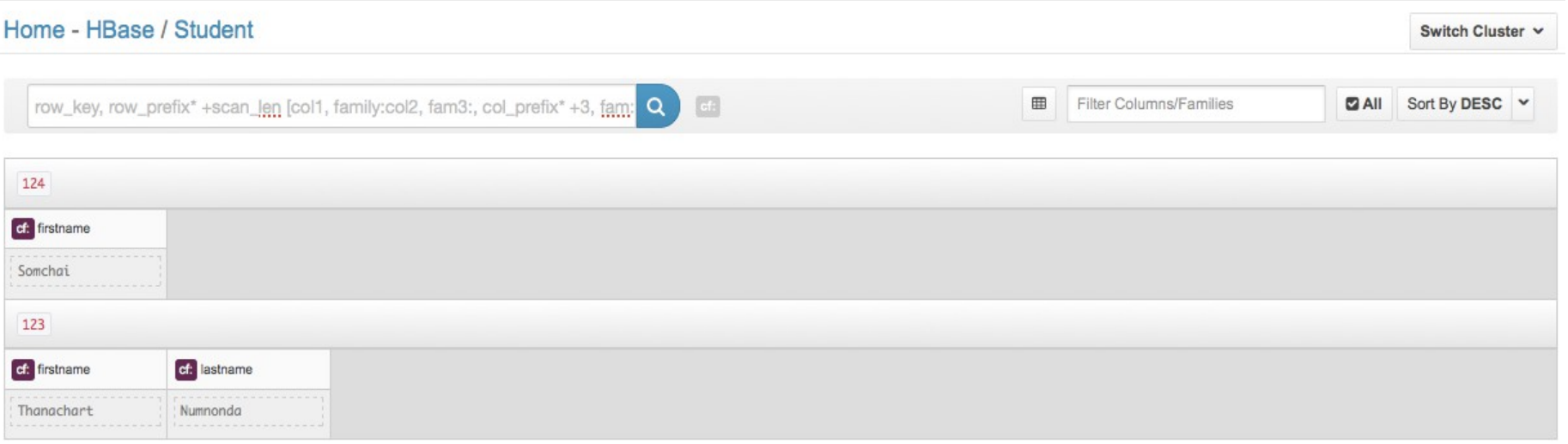
Insert a new row in a table

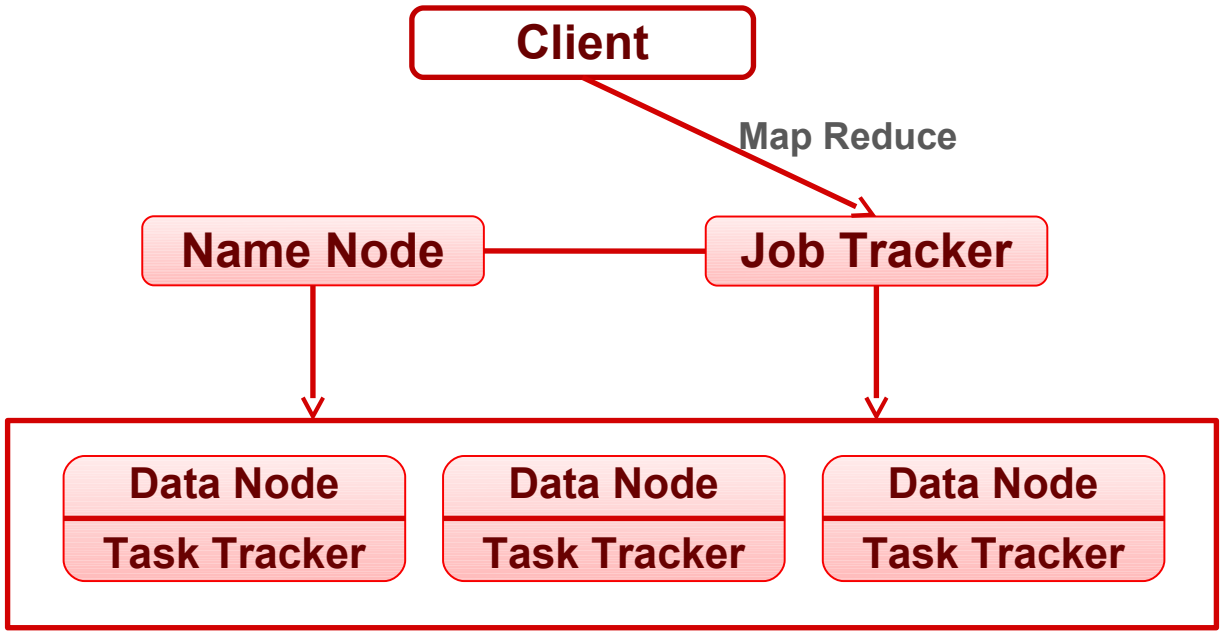


Add field into a new row



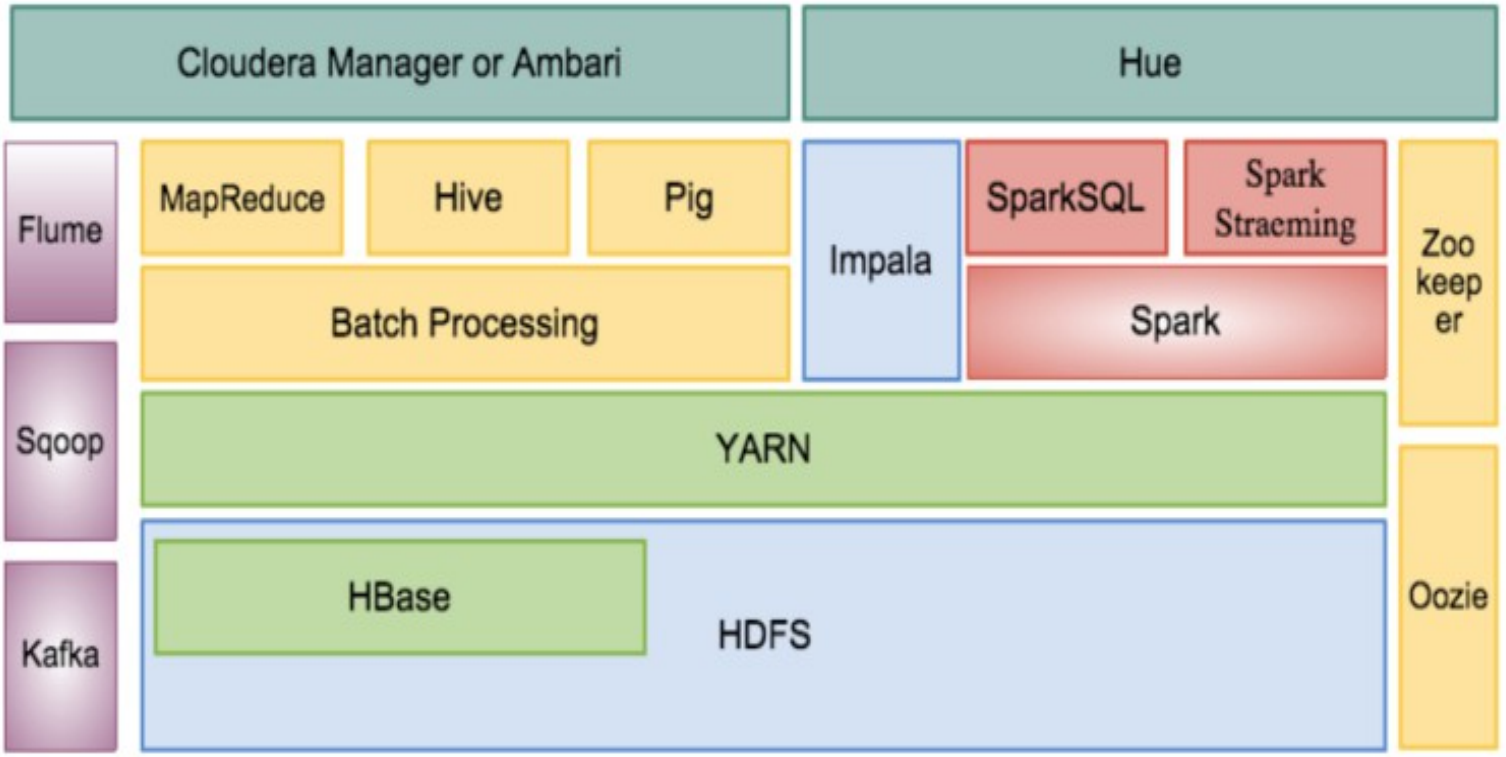






Lecture: Understanding Map Reduce Processing

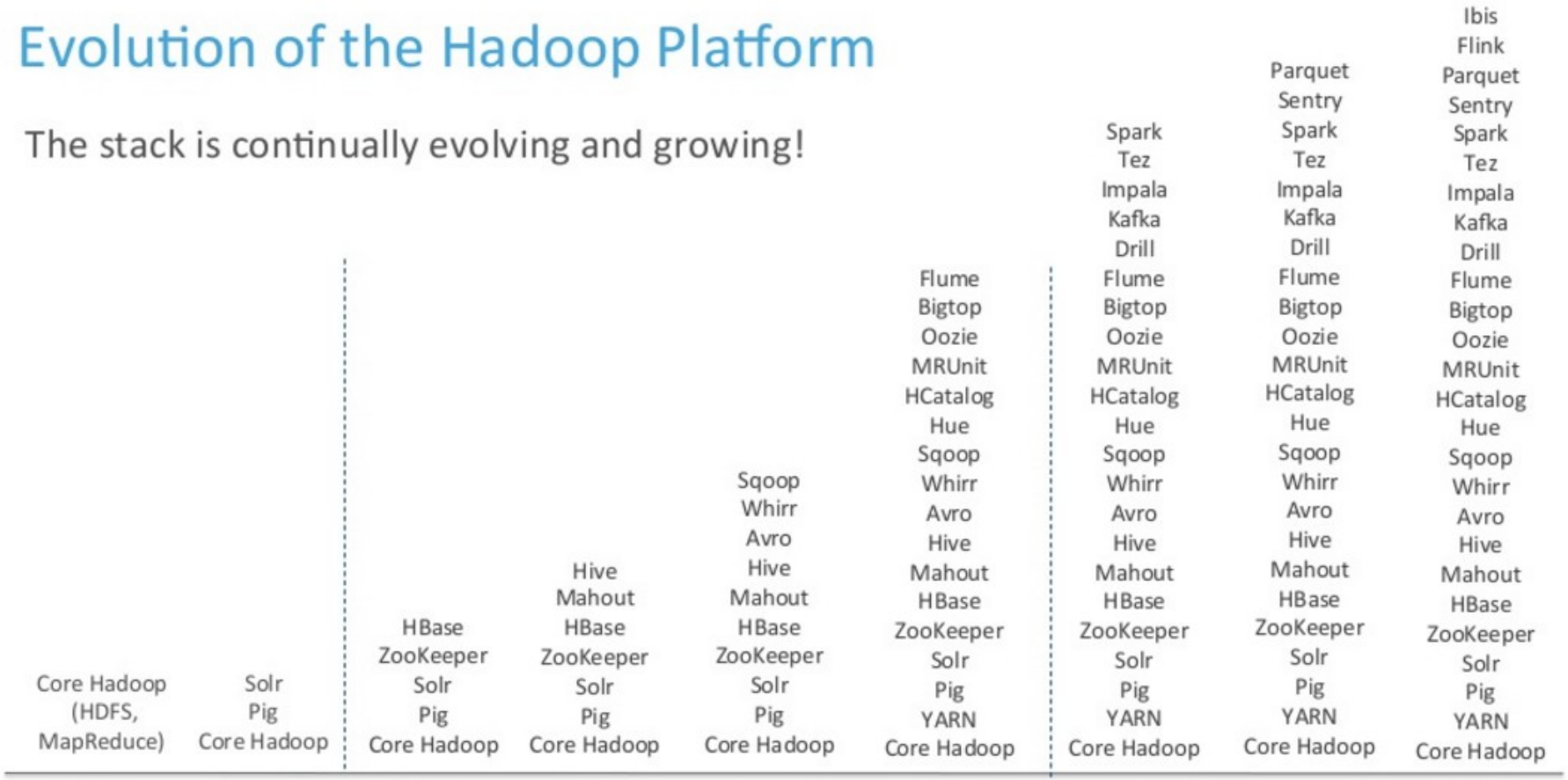
Hadoop Ecosystem



thanachart@imcinstitute.com

Evolution of the Hadoop Platform

The stack is continually evolving and growing!



cloudera

Source: The evolution and future of Hadoop storage: Cloudera

thanachart@imcinstitute.com

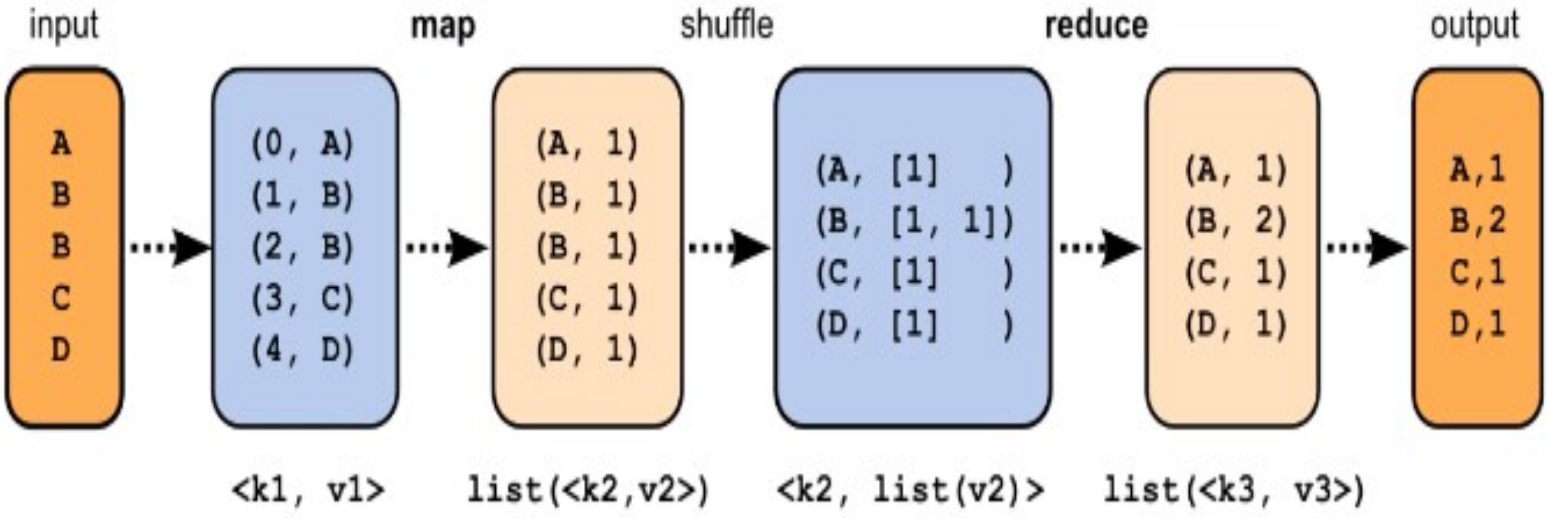
Before MapReduce…

- Large scale data processing was difficult!
  - Managing hundreds or thousands of processors
  - Managing parallelization and distribution
  - I/O Scheduling
  - Status and monitoring
  - Fault/crash tolerance
- MapReduce provides all of these, easily!

Source: http://labs.google.com/papers/mapreduce-osdi04-slides/index-auto-0002.html

thanachart@imcinstitute.com

MapReduce Framework



Source: www.bigdatauniversity.com

How Map and Reduce Work Together

- Map returns information
- Reduces accepts information
- Reduce applies a user defined function to reduce the amount of data

thanachart@imcinstitute.com

Map Abstraction

- Inputs a key/value pair
  - Key is a reference to the input value
  - Value is the data set on which to operate
- Evaluation
  - Function defined by user
  - Applies to every value in value input
    - Might need to parse input
- Produces a new list of key/value pairs
  - Can be different type from input pair

thanachart@imcinstitute.com

Reduce Abstraction

- Starts with intermediate Key / Value pairs
- Ends with finalized Key / Value pairs

- Starting pairs are sorted by key
- Iterator supplies the values for a given key to the Reduce function.

thanachart@imcinstitute.com

Reduce Abstraction

- Typically a function that:
  - Starts with a large number of key/value pairs
    - One key/value for each word in all files being greped (including multiple entries for the same word)
  - Ends with very few key/value pairs
    - One key/value for each unique word across all the files with the number of instances summed into this entry
- Broken up so a given worker works with input of the same key.

thanachart@imcinstitute.com

Why is this approach better?

- Creates an abstraction for dealing with complex overhead
  - The computations are simple, the overhead is messy
- Removing the overhead makes programs much smaller and thus easier to use
  - Less testing is required as well. The MapReduce libraries can be assumed to work properly, so only user code needs to be tested
- Division of labor also handled by the MapReduce libraries, so programmers only need to focus on the actual computation

thanachart@imcinstitute.com

Hands-On: Writing you own Map Reduce Program

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Example MapReduce: WordCount

```
package org.apache.hadoop.examples;

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
      StringTokenizer itr = new StringTokenizer(value.toString());
      while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
      }
    }
  }
```


thanachart@imcinstitute.com

```
 public static class IntSumReducer
      extends Reducer<Text,IntWritable,Text,IntWritable> {
   private IntWritable result = new IntWritable();

   public void reduce(Text key, Iterable<IntWritable> values,
                      Context context
                      ) throws IOException, InterruptedException {
     int sum = 0;
     for (IntWritable val : values) {
       sum += val.get();
     }
     result.set(sum);
     context.write(key, result);
   }
 }
```

```
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length != 2) {
      System.err.println("Usage: wordcount <in> <out>");
      System.exit(2);
    }
    Job job = new Job(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
    FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

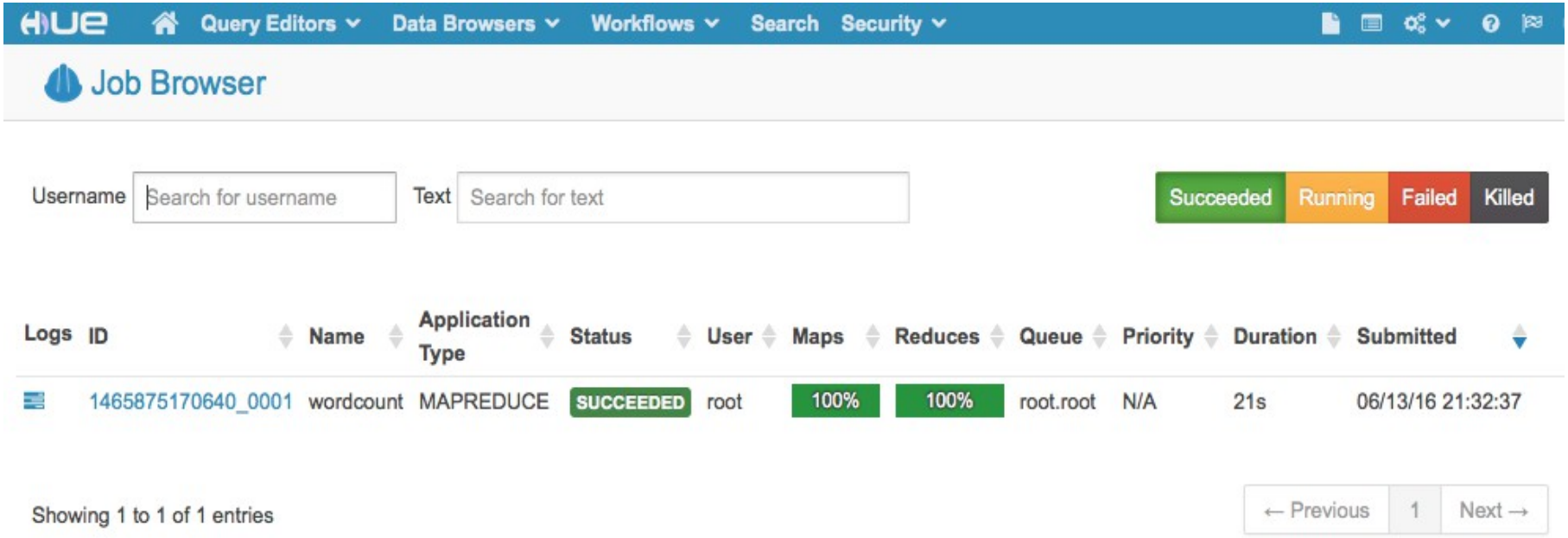

thanachart@imcinstitute.com

Running Map Reduce Program

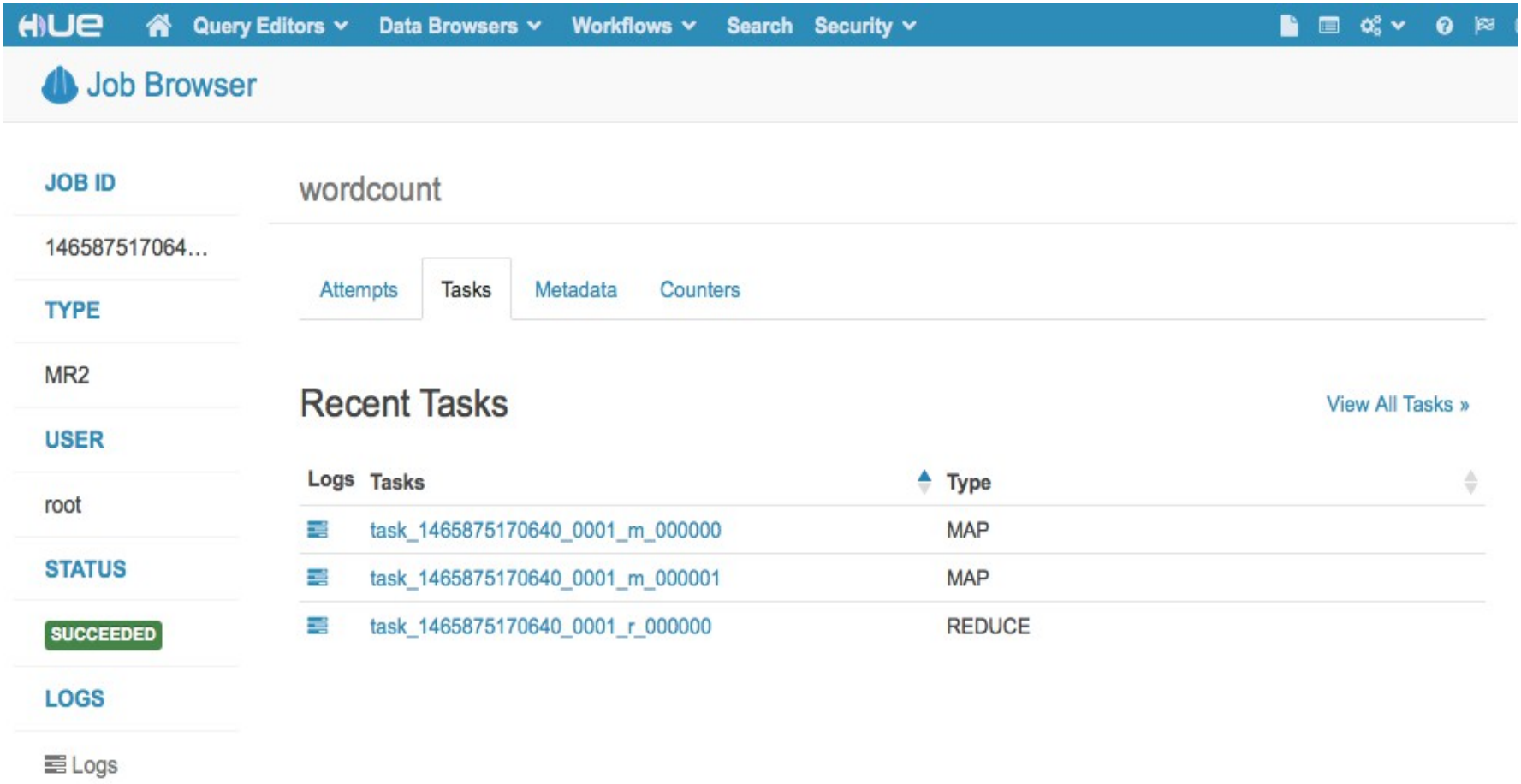
```
[cloudera@quickstart ~]$ cd workspace/
[cloudera@quickstart workspace]$ hadoop jar wordcount.jar org.myorg.WordCount input/*  output/wordcount_output
15/02/08 10:30:31 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
15/02/08 10:30:32 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
15/02/08 10:30:33 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool i
ation with ToolRunner to remedy this.
15/02/08 10:30:33 INFO mapred.FileInputFormat: Total input paths to process : 1
15/02/08 10:30:34 INFO mapreduce.JobSubmitter: number of splits:2
15/02/08 10:30:34 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1423408479621_0009
15/02/08 10:30:35 INFO impl.YarnClientImpl: Submitted application application_1423408479621_0009
15/02/08 10:30:35 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_142
15/02/08 10:30:35 INFO mapreduce.Job: Running job: job_1423408479621_0009
15/02/08 10:30:52 INFO mapreduce.Job: Job job_1423408479621_0009 running in uber mode : false
15/02/08 10:30:52 INFO mapreduce.Job:  map 0% reduce 0%
15/02/08 10:31:22 INFO mapreduce.Job:  map 58% reduce 0%
15/02/08 10:31:25 INFO mapreduce.Job:  map 100% reduce 0%
15/02/08 10:31:52 INFO mapreduce.Job:  map 100% reduce 100%
15/02/08 10:31:53 INFO mapreduce.Job: Job job_1423408479621_0009 completed successfully
15/02/08 10:31:53 INFO mapreduce.Job: Counters: 49
```

```
$cd /root/guest1
$wget https://dl.dropboxusercontent.com/u/12655380/wordcount.jar
$hadoop jar wordcount.jar org.myorg.WordCount
/user/cloudera/input/* /user/cloudera/output/wordcount
```

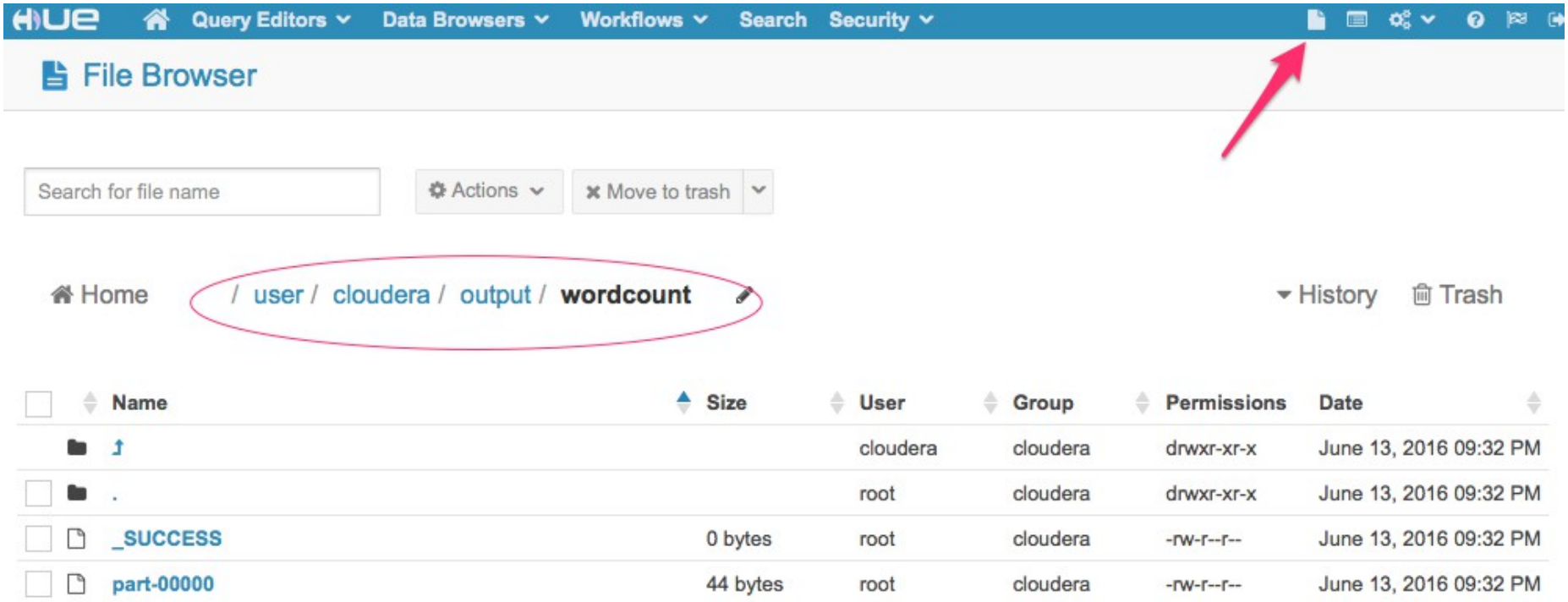
Reviewing MapReduce Job in Hue



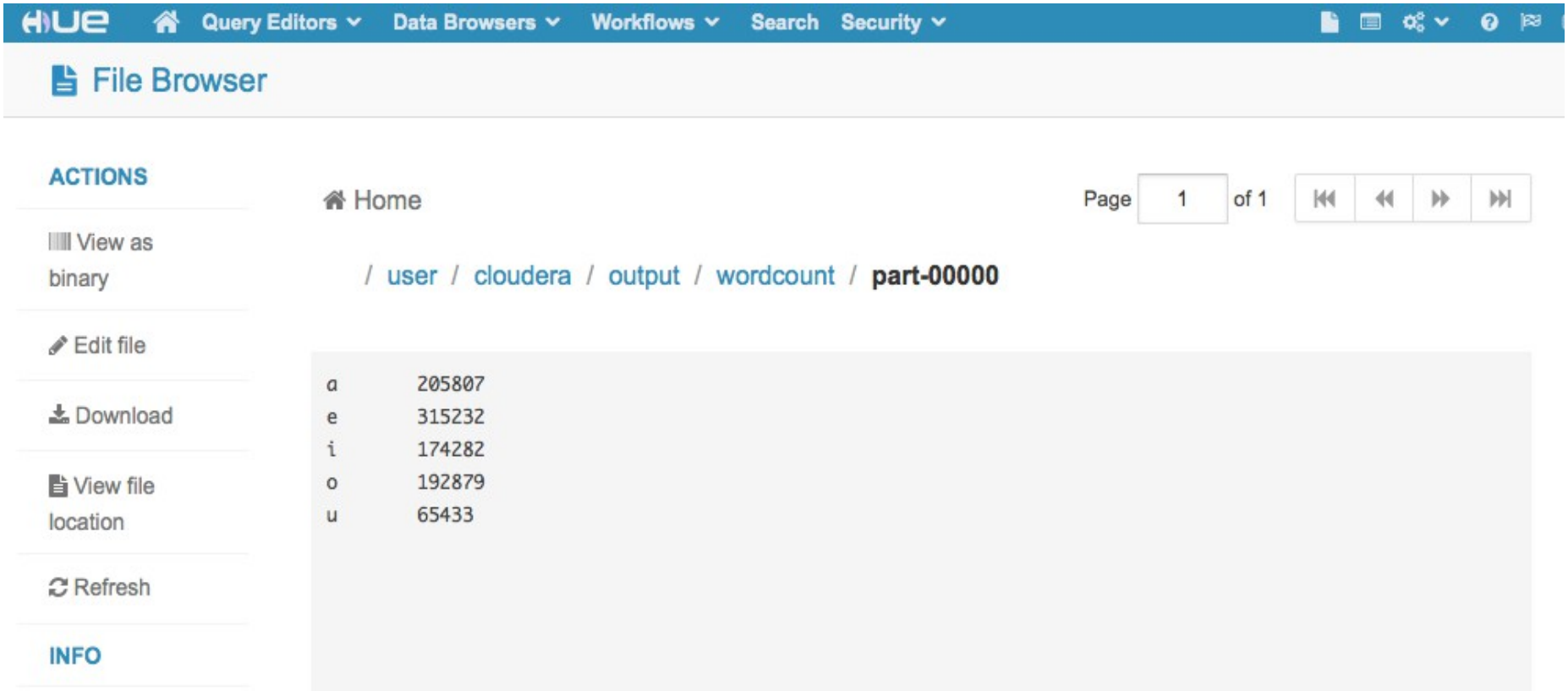
Reviewing MapReduce Job in Hue



Reviewing MapReduce Output Result



Reviewing MapReduce Output Result



Lecture
Understanding Oozie

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Introduction

Workslow scheduler for Hadoop

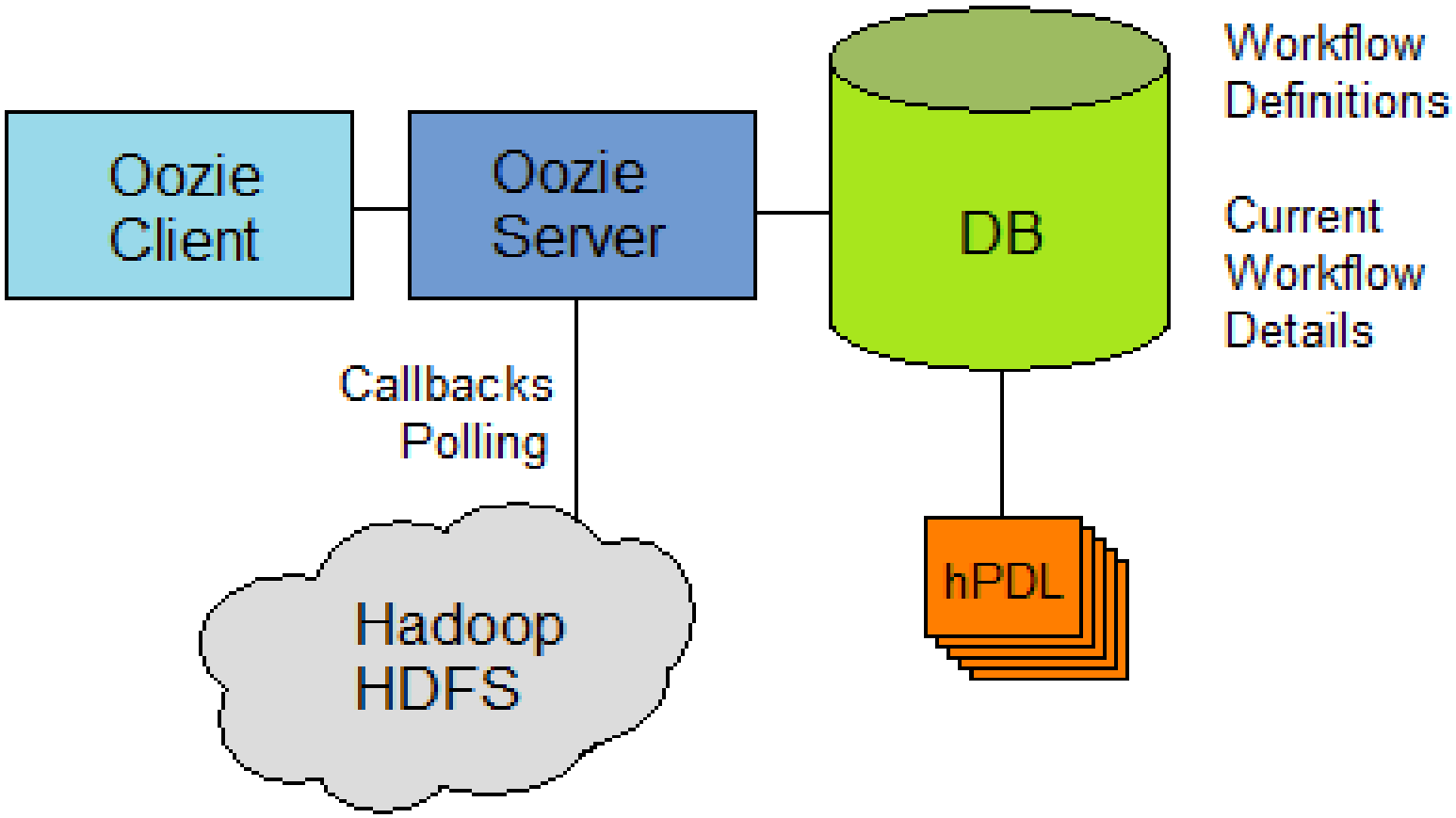
Oozie is a workflow scheduler system to manage Apache Hadoop jobs. Oozie is integrated with the rest of the Hadoop stack supporting several types of Hadoop jobs out of the box (such as Java map-reduce, Streaming map-reduce, Pig, Hive, Sqoop and Distcp) as well as system specific jobs (such as Java programs and shell scripts).

What is Oozie?

- Work flow scheduler for Hadoop
- Manages Hadoop Jobs
- Integrated with many Hadoop apps i.e. Pig, Hive
- Scaleable
- Schedule jobs
- A work flow is a collection of actions.
- A work flow is
  - Arranged as a DAG (direct acyclic graph)
  - Graph stored as hPDL (XML process definition)

thanachart@imcinstitute.com

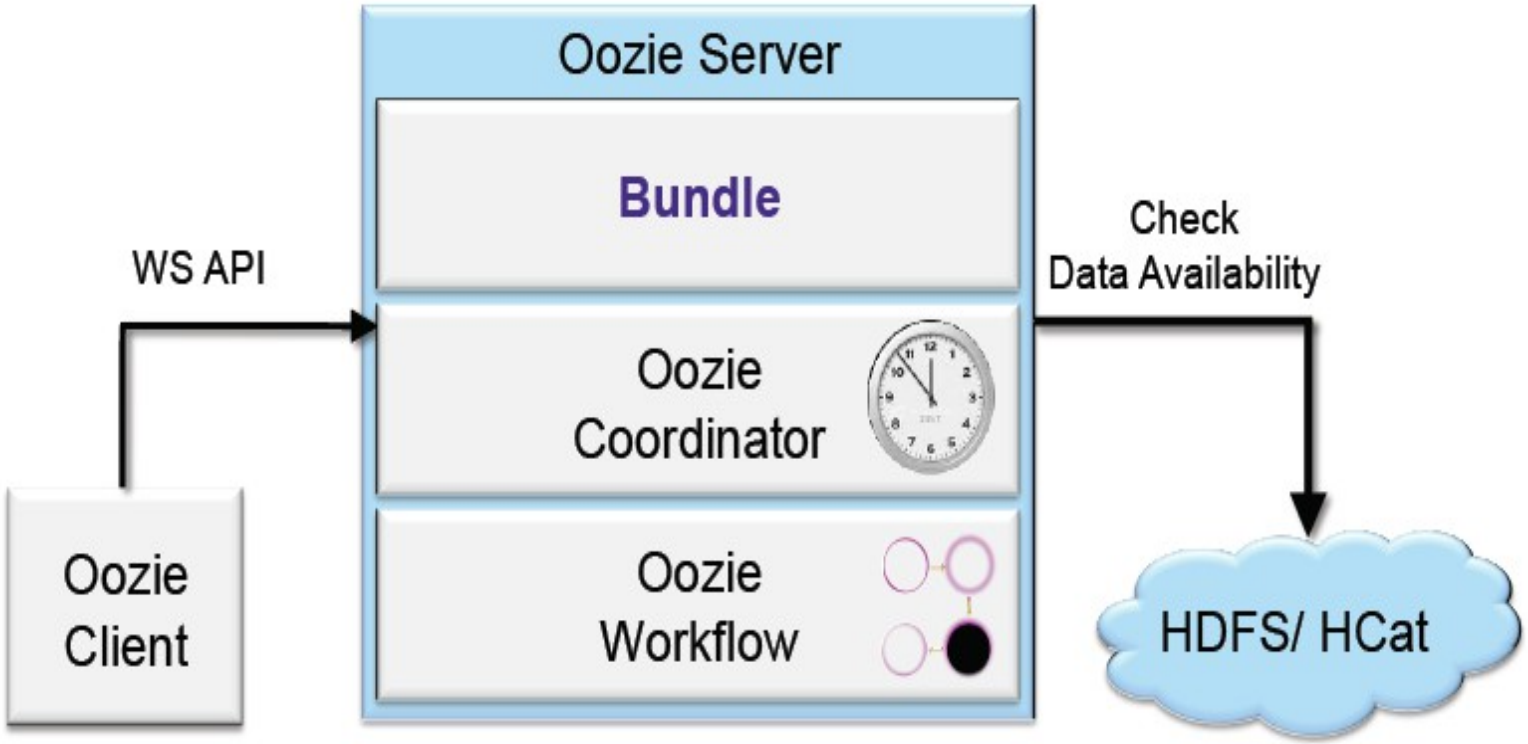
Oozie Architecture



Source: info@semtech-solutions.co.nz

thanachart@imcinstitute.com

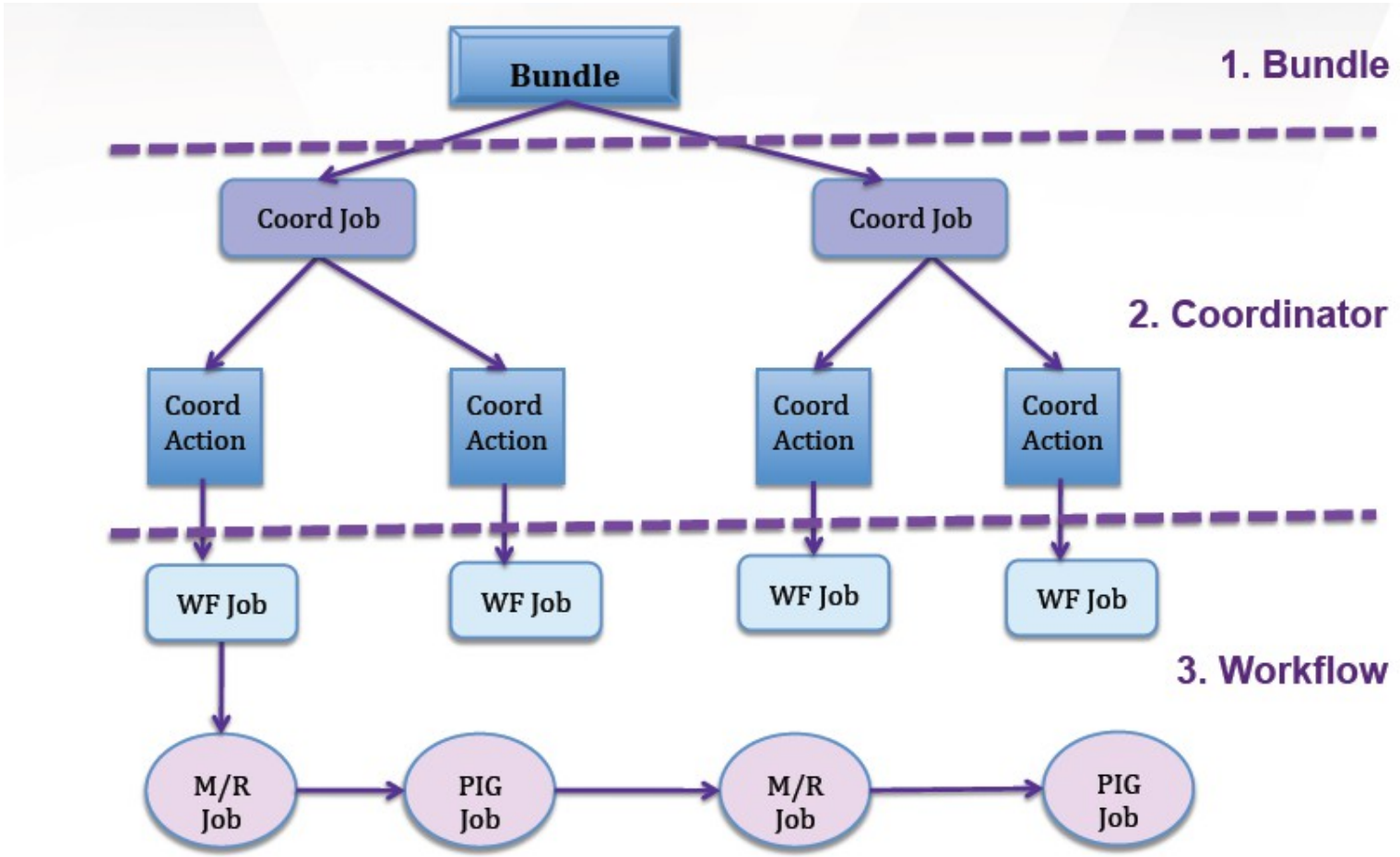
Oozie Server



Source: Oozie – Now and Beyond, Yahoo, 2013

thanachart@imcinstitute.com

Layer of Abstraction in Oozie



Source: Oozie – Now and Beyond, Yahoo, 2013

thanachart@imcinstitute.com

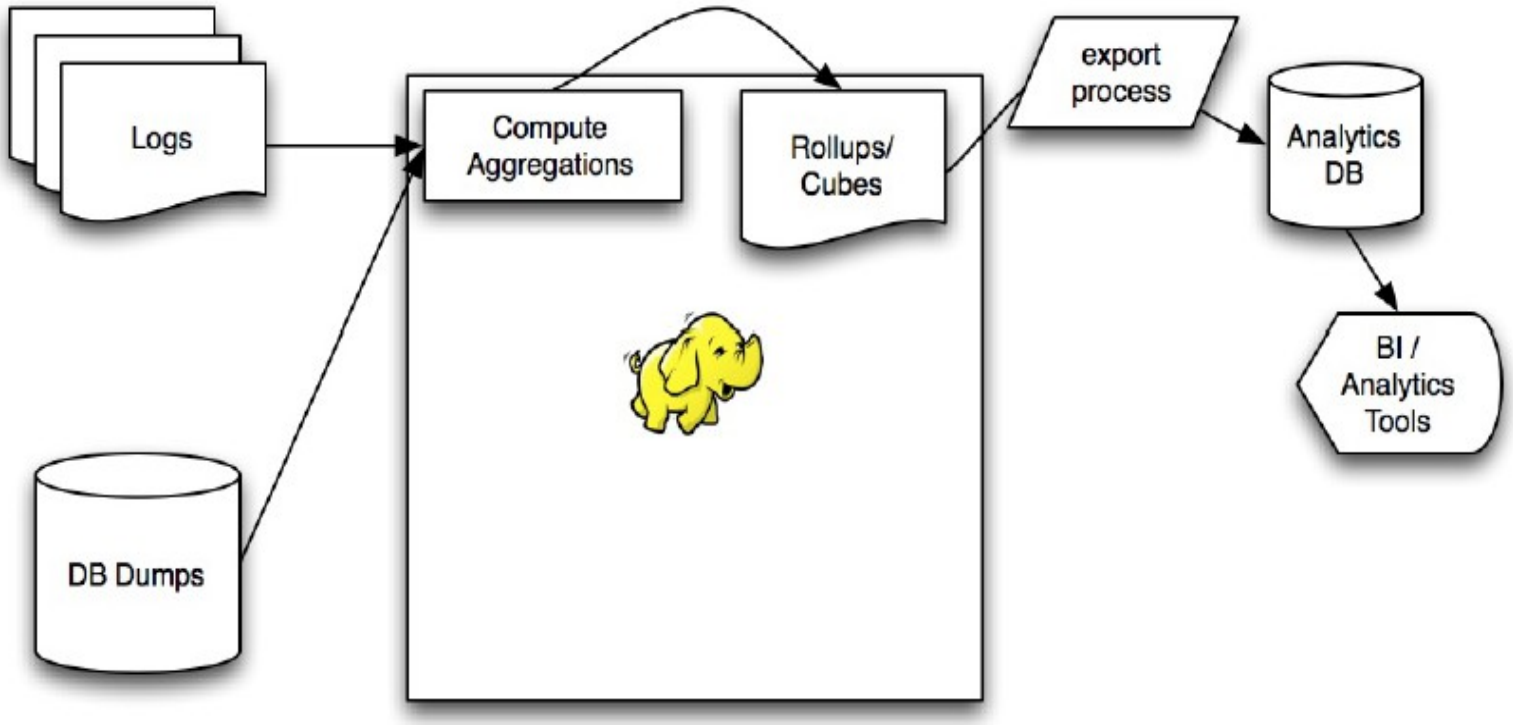
Workflow Example: Data Analytics

- Logs => fact table(s)
- Database backup => Dimension tables
- Complete rollups/cubes
- Load data into a low-latency storage (e.g. Hbae, HDFS)
- Dashboard & BI tools

Source: Workflow Engines for Hadoop, Joe Crobak, 2013

thanachart@imcinstitute.com

Workflow Example: Data Analytics



Source: Workflow Engines for Hadoop, Joe Crobak, 2013

thanachart@imcinstitute.com

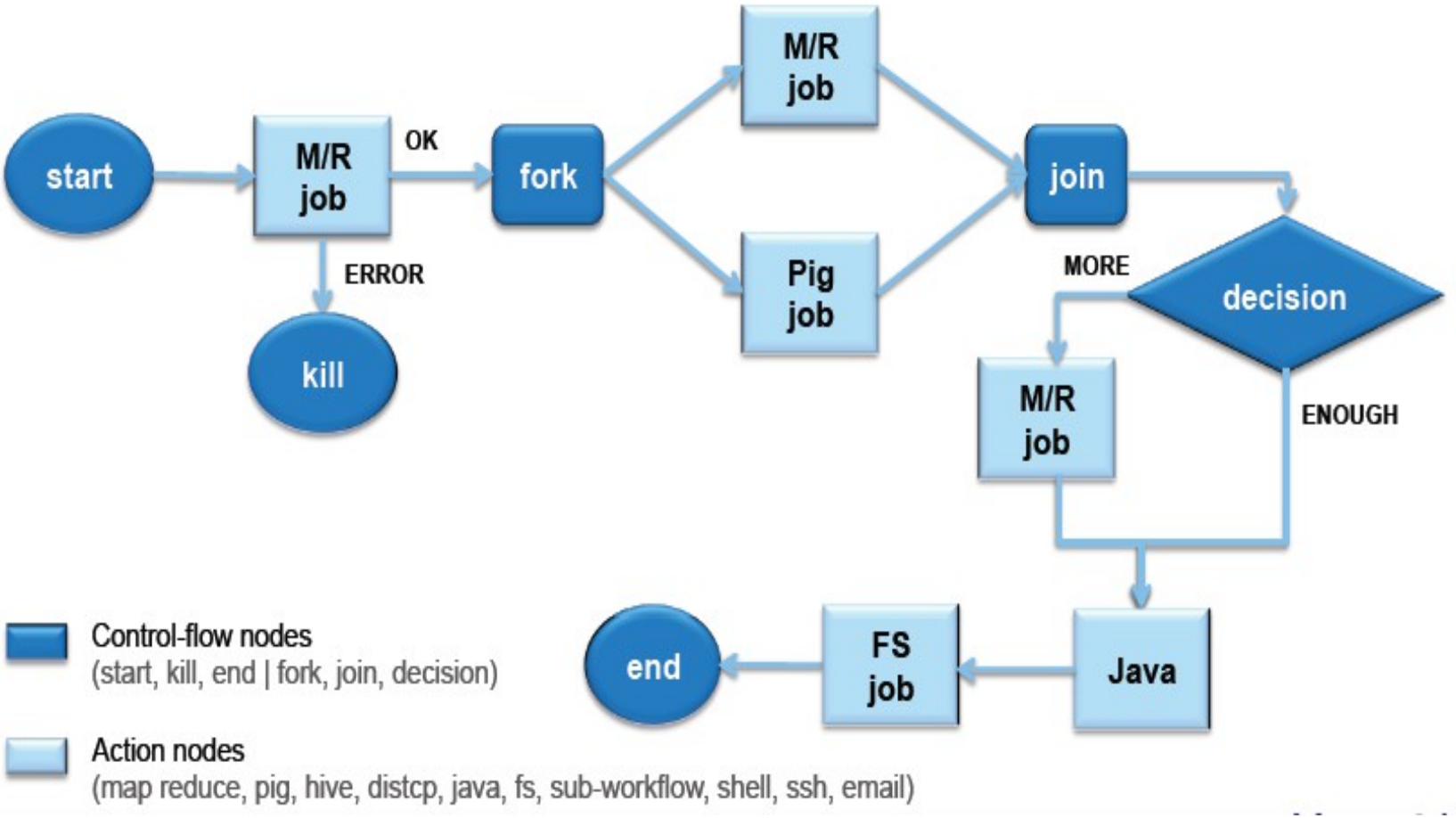
Workflow Example: Data Analytics

- What happens if there is a failure?
  - Rebuild the failed day
  - .. and any downstream datasets
- With Hadoop Workflow
  - Possible OK to skip a day
  - Workflow tends to be self-contained, so you do not need to run downstream.
  - Sanity check your data before pushing to production.

Source: Workflow Engines for Hadoop, Joe Crobak, 2013

thanachart@imcinstitute.com

Oozie Workflow



Source: Oozie – Now and Beyond, Yahoo, 2013

thanachart@imcinstitute.com

Oozie Use Cases

- Time Triggers
  - Execute your workflow every 15 minutes
- Time and Data Triggers
  - Materialize your workflow every hour, but only run them when the input data is ready (that is loaded to the grid every hour)
- Rolling Window
  - Access 15 minute datasets and roll them up into hourly datasets

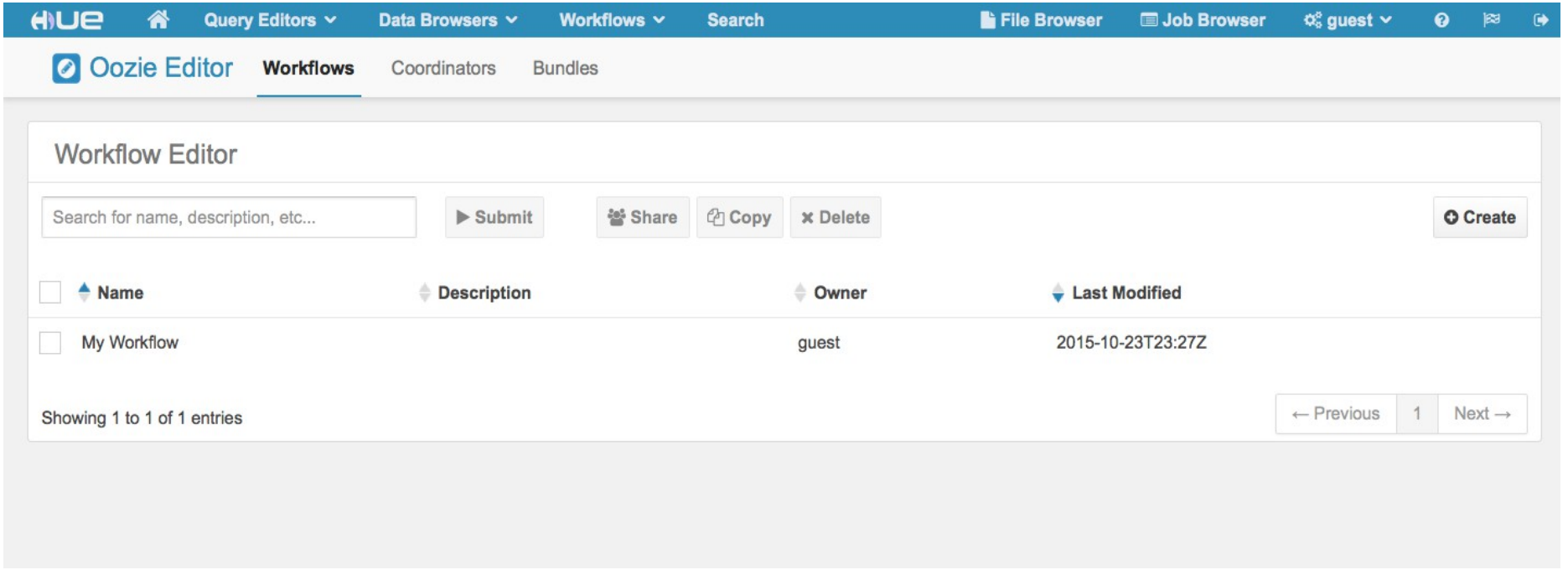
Source: Oozie – Now and Beyond, Yahoo, 2013

thanachart@imcinstitute.com

Hands-On: Running Map Reduce using Oozie workflow

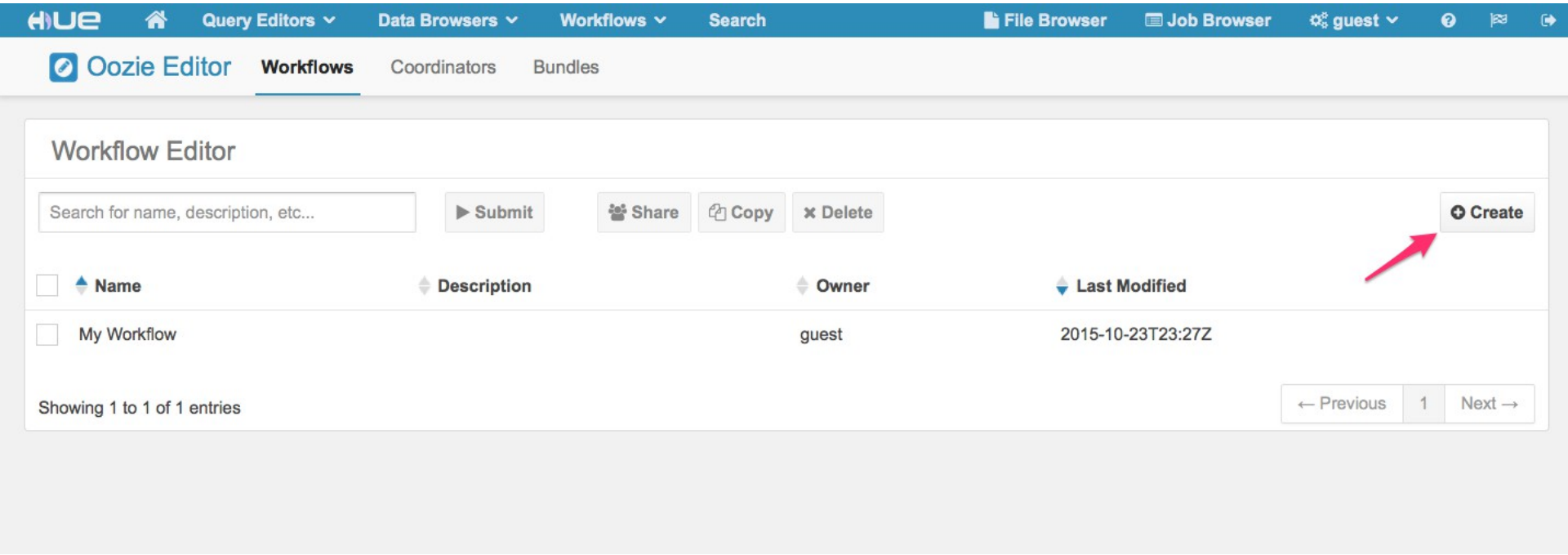
Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Using Hue: select WorkFlows >> Editors >> Workflows

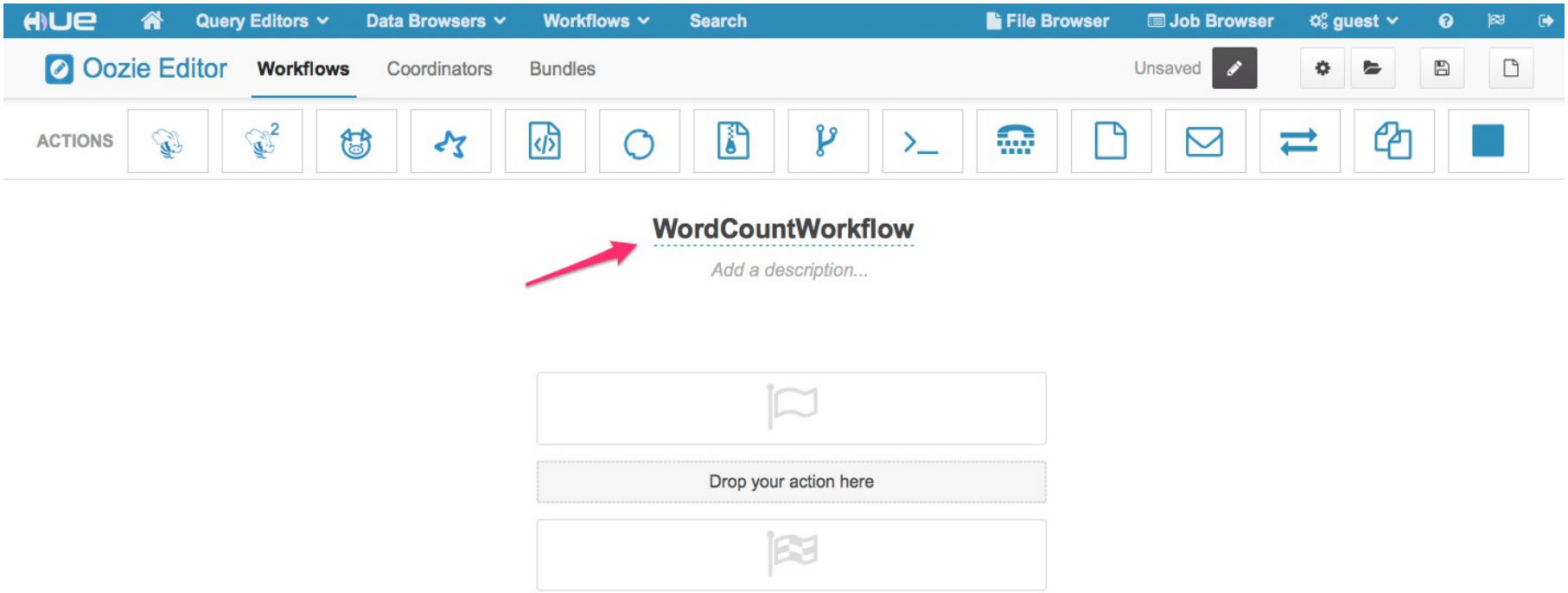


Create a new workflow

- Click Create button; the following screen will be displayed
- Name the workflow as WordCountWorkflow



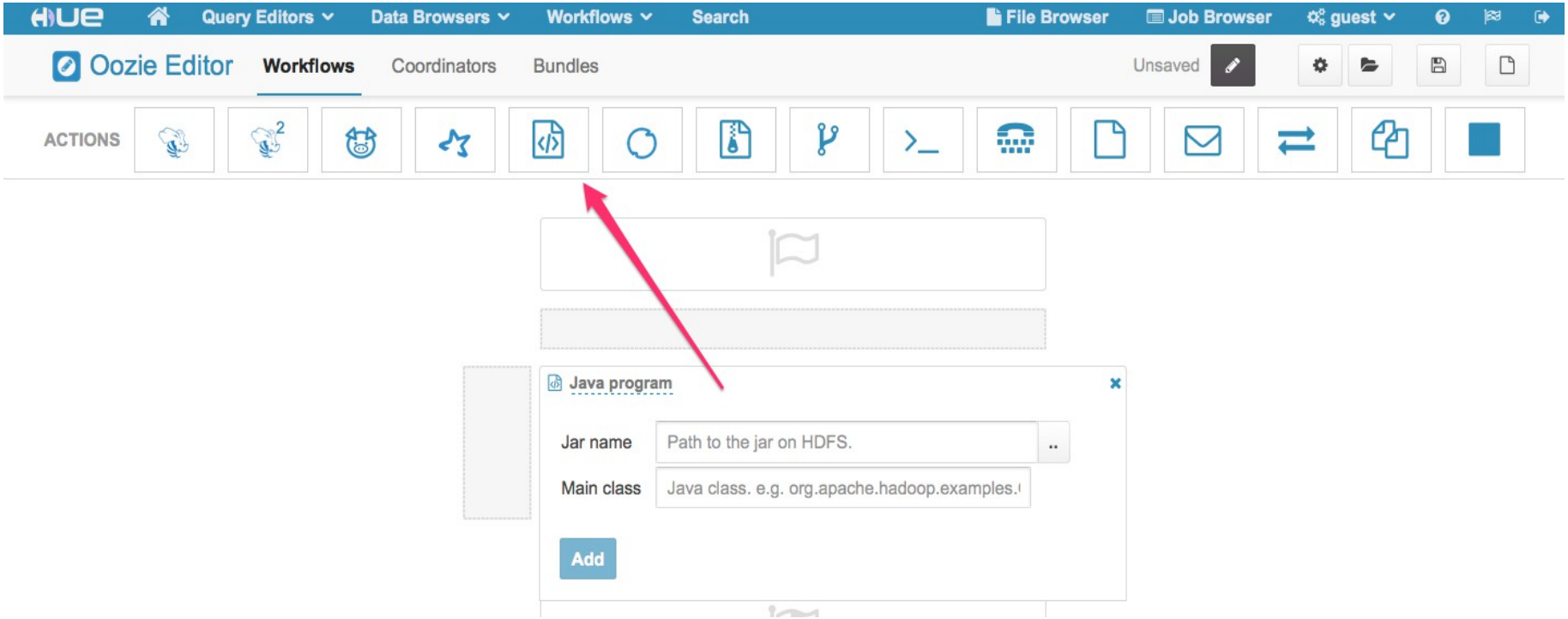
thanachart@imcinstitute.com



thanachart@imcinstitute.com

Select a Java job for the workflow

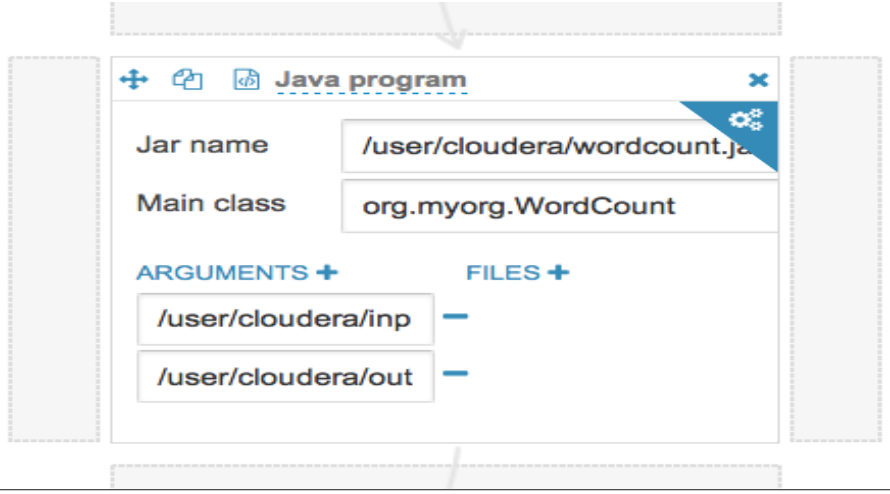
- From the Oozie editor, drag **Java Program** and drop between start and end



thanachart@imcinstitute.com

Edit the Java Job

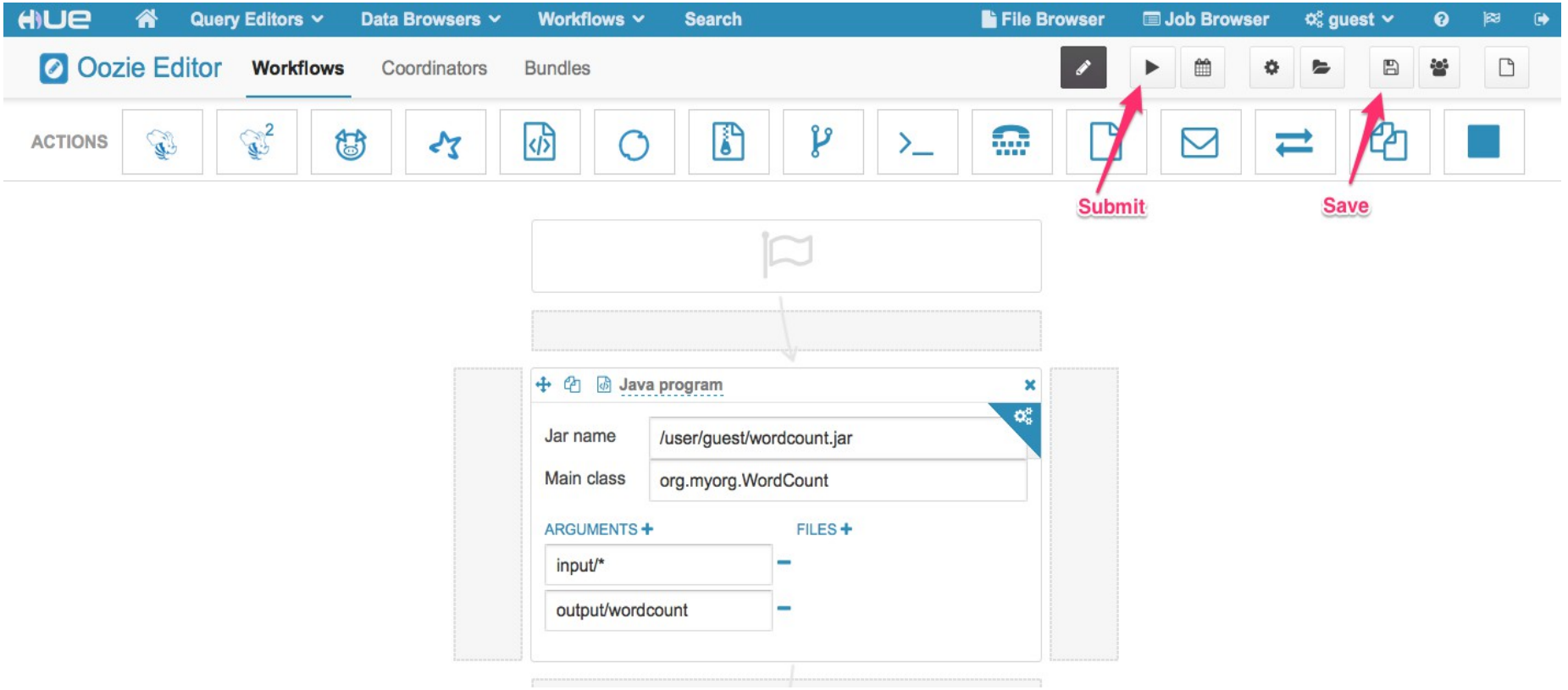
- Assign the following value
  -
  - Jar name: wordcount.jar (select … choose upload from local machine)
  - Main Class: org.myorg.WordCount
  - Arguments: /user/cloudera/input/*
  - /user/cloudera/output/wordcount



thanachart@imcinstitute.com

Submit the workflow

- Click Done, follow by Save
- Then click submit



thanachart@imcinstitute.com

Introduction

A Petabyte Scale Data Warehouse Using Hadoop

Hive is developed by Facebook, designed to enable easy data summarization, ad-hoc querying and analysis of large volumes of data. It provides a simple query language called Hive QL, which is based on SQL

What Hive is NOT

Hive is not designed for online transaction processing and does not offer real-time queries and row level updates. It is best used for batch jobs over large sets of immutable data (like web logs, etc.).

Sample HiveQL

The Query compiler uses the information stored in the metastore to convert SQL queries into a sequence of map/reduce jobs, e.g. the following query

```
SELECT * FROM t where t.c = 'xyz'

SELECT t1.c2 FROM t1 JOIN t2 ON (t1.c1 = t2.c1)

SELECT t1.c1, count(1) from t1 group by t1.c1
```

Hive.apache.org

Sample HiveQL

The Query compiler uses the information stored in the metastore to convert SQL queries into a sequence of map/reduce jobs, e.g. the following query

```
SELECT * FROM t where t.c = 'xyz'

SELECT t1.c2 FROM t1 JOIN t2 ON (t1.c1 = t2.c1)

SELECT t1.c1, count(1) from t1 group by t1.c1
```

Hive.apache.org

System Architecture and Components

Metastore: To store the meta data.

Query compiler and execution engine: To convert SQL queries to a sequence of map/reduce jobs that are then executed on Hadoop.

SerDe and ObjectInspectors: Programmable interfaces and implementations of common data formats and types.

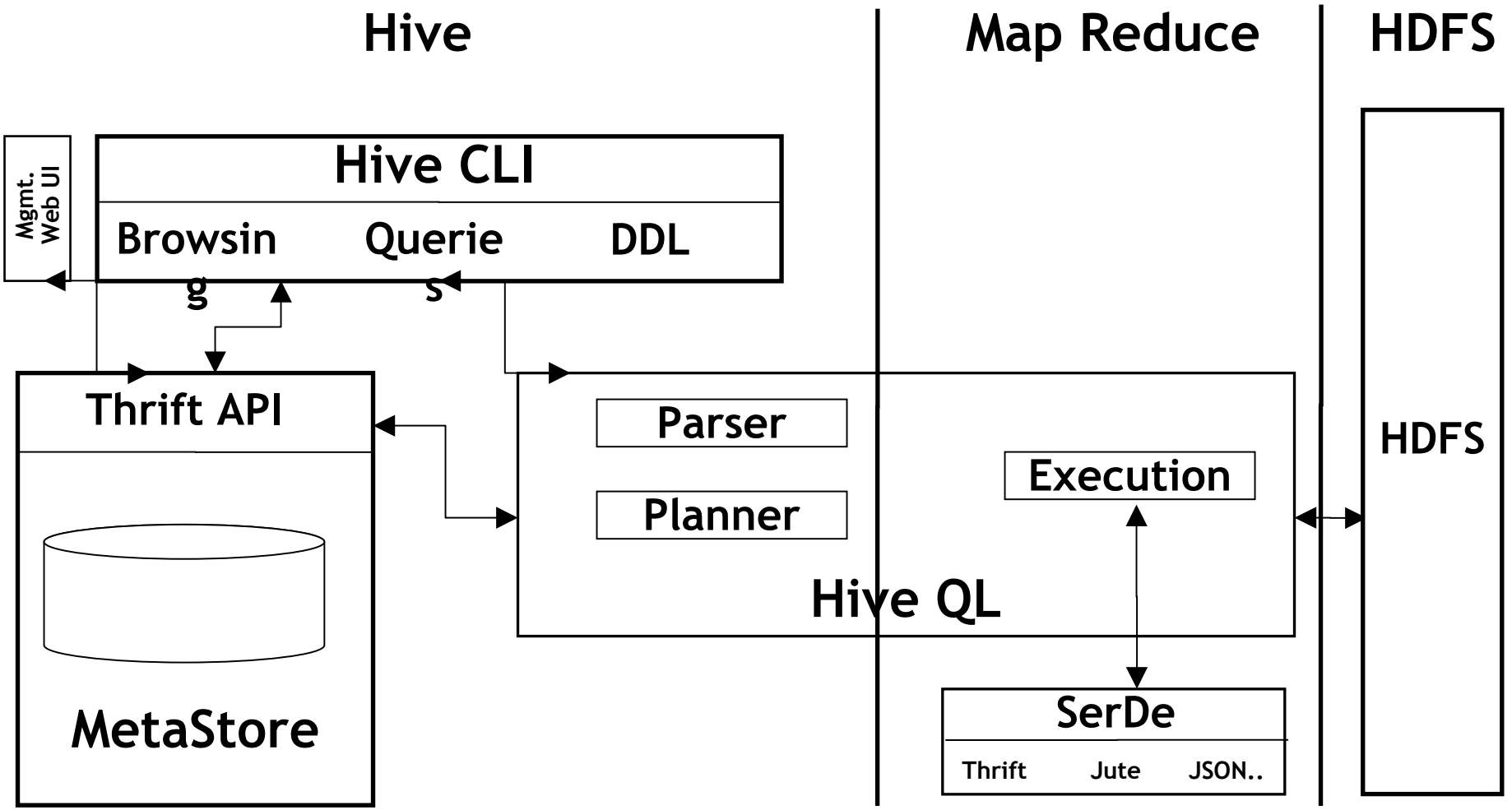
A SerDe is a combination of a Serializer and a Deserializer (hence, Ser-De). The Deserializer interface takes a string or binary representation of a record, and translates it into a Java object that Hive can manipulate. The Serializer, however, will take a Java object that Hive has been working with, and turn it into something that Hive can write to HDFS or another supported system.

UDF and UDAF: Programmable interfaces and implementations for user defined functions (scalar and aggregate functions).

Clients: Command line client similar to Mysql command line.

hive.apache.org

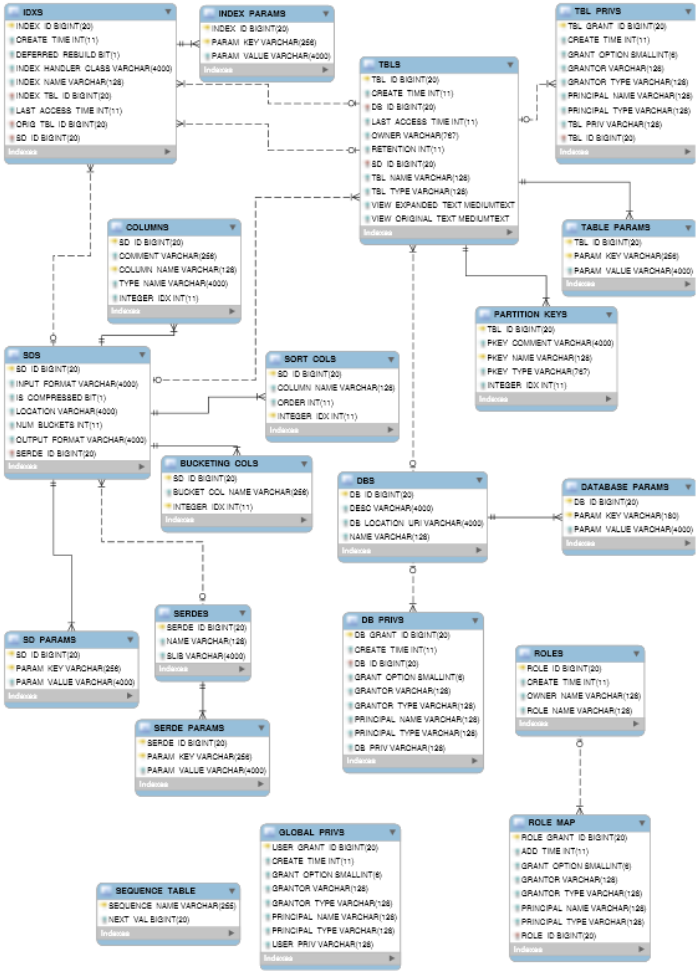
Architecture Overview



Hive.apache.org

Danairat T., danairat@gmail.com: Thanachart N., thanachart@imcinstitute.com April 2015

Hive Metastore



Hive Metastore is a repository to keep all Hive metadata; Tables and Partitions definition.

By default, Hive will store its metadata in Derby DB

Hive Built in Functions

Return Type	Function Name (Signature)	Description
BIGINT	round(double a)	returns the rounded BIGINT value of the double
BIGINT	floor(double a)	returns the maximum BIGINT value that is equal or less than the double
BIGINT	ceil(double a)	returns the minimum BIGINT value that is equal or greater than the double
double	rand(), rand(int seed)	returns a random number (that changes from row to row). Specifiying the seed will make sure the generated random number sequence is deterministic.
string	concat(string A, string B,...)	returns the string resulting from concatenating B after A. For example, concat('foo', 'bar') results in 'foobar'. This function accepts arbitrary number of arguments and return the concatenation of all of them.
string	substr(string A, int start)	returns the substring of A starting from start position till the end of string A. For example, substr('foobar', 4) results in 'bar'
string	substr(string A, int start, int length)	returns the substring of A starting from start position with the given length e.g. substr('foobar', 4, 2) results in 'ba'
string	upper(string A)	returns the string resulting from converting all characters of A to upper case e.g. upper('fOoBaR') results in 'FOOBAR'
string	ucase(string A)	Same as upper
string	lower(string A)	returns the string resulting from converting all characters of B to lower case e.g. lower('fOoBaR') results in 'foobar'
string	lcase(string A)	Same as lower
string	trim(string A)	returns the string resulting from trimming spaces from both ends of A e.g. trim(' foobar ') results in 'foobar'
string	ltrim(string A)	returns the string resulting from trimming spaces from the beginning(left hand side) of A. For example, ltrim(' foobar ') results in 'foobar '
string	rtrim(string A)	returns the string resulting from trimming spaces from the end(right hand side) of A. For example, rtrim(' foobar ') results in ' foobar'
string	regexp_replace(string A, string B, string C)	returns the string resulting from replacing all substrings in B that match the Java regular expression syntax(See Java regular expressions syntax) with C. For example, regexp_replace('foobar', 'oo|ar',) returns 'fb'
string	from_unixtime(int unixtime)	convert the number of seconds from unix epoch (1970-01-01 00:00:00 UTC) to a string representing the timestamp of that moment in the current system time zone in the format of "1970-01-01 00:00:00"
string	to_date(string timestamp)	Return the date part of a timestamp string: to_date("1970-01-01 00:00:00") = "1970-01-01"
int	year(string date)	Return the year part of a date or a timestamp string: year("1970-01-01 00:00:00") = 1970, year("1970-01-01") = 1970
int	month(string date)	Return the month part of a date or a timestamp string: month("1970-11-01 00:00:00") = 11, month("1970-11-01") = 11
int	day(string date)	Return the day part of a date or a timestamp string: day("1970-11-01 00:00:00") = 1, day("1970-11-01") = 1
string	get_json_object(string json_string, string path)	Extract json object from a json string based on json path specified, and return json string of the extracted json object. It will return null if the input json string is invalid

hive.apache.org

Hive Aggregate Functions

Return Type	Aggregation Function Name (Signature)	Description
BIGINT	count(*), count(expr), count(DISTINCT expr[, expr_.])	count(*) - Returns the total number of retrieved rows, including rows containing NULL values; count(expr) - Returns the number of rows for which the supplied expression is non-NULL; count(DISTINCT expr[, expr]) - Returns the number of rows for which the supplied expression(s) are unique and non-NULL.
DOUBLE	sum(col), sum(DISTINCT col)	returns the sum of the elements in the group or the sum of the distinct values of the column in the group
DOUBLE	avg(col), avg(DISTINCT col)	returns the average of the elements in the group or the average of the distinct values of the column in the group
DOUBLE	min(col)	returns the minimum value of the column in the group
DOUBLE	max(col)	returns the maximum value of the column in the group

hive.apache.org

Running Hive

Hive Shell

Interactive

hive

Script

hive -f myscript

Inline

*hive -e 'SELECT * FROM mytable'*

Hive.apache.org

Hive Commands

Command Line

Function	Hive
Run query	`hive -e 'select a.col from tab1 a'`
Run query silent mode	`hive -S -e 'select a.col from tab1 a'`
Set hive config variables	`hive -e 'select a.col from tab1 a' -hiveconf hive.root.logger=DEBUG,console`
Use initialization script	`hive -i initialize.sql`
Run non-interactive script	`hive -f script.sql`

Hive Shell

Function	Hive
Run script inside shell	`source file_name`
Run ls (dfs) commands	`dfs -ls /user`
Run ls (bash command) from shell	`!ls`
Set configuration variables	`set mapred.reduce.tasks=32`
TAB auto completion	`set hive.<TAB>`
Show all variables starting with hive	`set`
Revert all variables	`reset`
Add jar to distributed cache	`add jar jar_path`
Show all jars in distributed cache	`list jars`
Delete jar from distributed cache	`delete jar jar_name`

ortonworks.com

Hive Tables

- Managed- CREATE TABLE
  - LOAD- File moved into Hive's data warehouse directory
  - DROP- Both data and metadata are deleted.
- External- CREATE EXTERNAL TABLE
  - LOAD- No file moved
  - DROP- Only metadata deleted
  - Use when sharing data between Hive and Hadoop applications or you want to use multiple schema on the same data

Hive External Table

```
– CREATE EXTERNAL TABLE external_Table (dummy STRING)
– LOCATION '/user/notroot/external_table';
```

Dropping External Table using Hive:-

- Hive will delete metadata from metastore
- Hive will NOT delete the HDFS file
- You need to manually delete the HDFS file

Danairat T., danairat@gmail.com: Thanachart N., thanachart@imcinstitute.com

Java JDBC for Hive

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveJdbcClient {
  private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

  public static void main(String[] args) throws SQLException {
    try {
      Class.forName(driverName);
    } catch (ClassNotFoundException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
      System.exit(1);
    }
    Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/default", "", "");
    Statement stmt = con.createStatement();
    String tableName = "testHiveDriverTable";
    stmt.executeQuery("drop table " + tableName);
    ResultSet res = stmt.executeQuery("create table " + tableName + " (key int, value string)");
    // show tables
    String sql = "show tables '" + tableName + "'";
    System.out.println("Running: " + sql);
    res = stmt.executeQuery(sql);
    if (res.next()) {
      System.out.println(res.getString(1));
    }
    // describe table
    sql = "describe " + tableName;
    System.out.println("Running: " + sql);
    res = stmt.executeQuery(sql);
    while (res.next()) {
      System.out.println(res.getString(1) + "\t" + res.getString(2));
    }
```

Java JDBC for Hive

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.DriverManager;

public class HiveJdbcClient {
  private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

  public static void main(String[] args) throws SQLException {
    try {
      Class.forName(driverName);
    } catch (ClassNotFoundException e) {
      // TODO Auto-generated catch block
      e.printStackTrace();
      System.exit(1);
    }
    Connection con = DriverManager.getConnection("jdbc:hive://localhost:10000/default", "", "");
    Statement stmt = con.createStatement();
    String tableName = "testHiveDriverTable";
    stmt.executeQuery("drop table " + tableName);
    ResultSet res = stmt.executeQuery("create table " + tableName + " (key int, value string)");
    // show tables
    String sql = "show tables '" + tableName + "'";
    System.out.println("Running: " + sql);
    res = stmt.executeQuery(sql);
    if (res.next()) {
      System.out.println(res.getString(1));
    }
    // describe table
    sql = "describe " + tableName;
    System.out.println("Running: " + sql);
    res = stmt.executeQuery(sql);
    while (res.next()) {
      System.out.println(res.getString(1) + "\t" + res.getString(2));
    }
```

HiveQL and MySQL Comparison

Metadata

Function	MySQL	HiveQL
Selecting a database	`USE database;`	`USE database;`
Listing databases	`SHOW DATABASES;`	`SHOW DATABASES;`
Listing tables in a database	`SHOW TABLES;`	`SHOW TABLES;`
Describing the format of a table	`DESCRIBE table;`	`DESCRIBE (FORMATTED|EXTENDED) table;`
Creating a database	`CREATE DATABASE db_name;`	`CREATE DATABASE db_name;`
Dropping a database	`DROP DATABASE db_name;`	`DROP DATABASE db_name (CASCADE);`

HiveQL and MySQL Query Comparison

Query

Function	MySQL	HiveQL
Retrieving information	`SELECT from_columns FROM table WHERE conditions;`	`SELECT from_columns FROM table WHERE conditions;`
All values	`SELECT * FROM table;`	`SELECT * FROM table;`
Some values	`SELECT * FROM table WHERE rec_name = "value";`	`SELECT * FROM table WHERE rec_name = "value";`
Multiple criteria	`SELECT * FROM table WHERE rec1="value1" AND rec2="value2";`	`SELECT * FROM TABLE WHERE rec1 = "value1" AND rec2 = "value2";`
Selecting specific columns	`SELECT column_name FROM table;`	`SELECT column_name FROM table;`
Retrieving unique output records	`SELECT DISTINCT column_name FROM table;`	`SELECT DISTINCT column_name FROM table;`
Sorting	`SELECT col1, col2 FROM table ORDER BY col2;`	`SELECT col1, col2 FROM table ORDER BY col2;`
Sorting backward	`SELECT col1, col2 FROM table ORDER BY col2 DESC;`	`SELECT col1, col2 FROM table ORDER BY col2 DESC;`
Counting rows	`SELECT COUNT(*) FROM table;`	`SELECT COUNT(*) FROM table;`
Grouping with counting	`SELECT owner, COUNT(*) FROM table GROUP BY owner;`	`SELECT owner, COUNT(*) FROM table GROUP BY owner;`
Maximum value	`SELECT MAX(col_name) AS label FROM table;`	`SELECT MAX(col_name) AS label FROM table;`
Selecting from multiple tables (Join same table using alias w/"AS")	`SELECT pet.name, comment FROM pet, event WHERE pet.name = event.name;`	`SELECT pet.name, comment FROM pet JOIN event ON (pet.name = event.name);`

ortonworks.com

Hands-On: Loading Data using Hive

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Start Hive

```
[root@quickstart guest1]# hive
2016-06-14 07:48:56,273 WARN  [main] mapreduce.TableMapReduceUtil: The
hbase-prefix-tree module jar containing PrefixTreeCodec is not present.
  Continuing without it.

Logging initialized using configuration in file:/etc/hive/conf.dist/hiv
e-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended
.
hive>
```

Quit from Hive

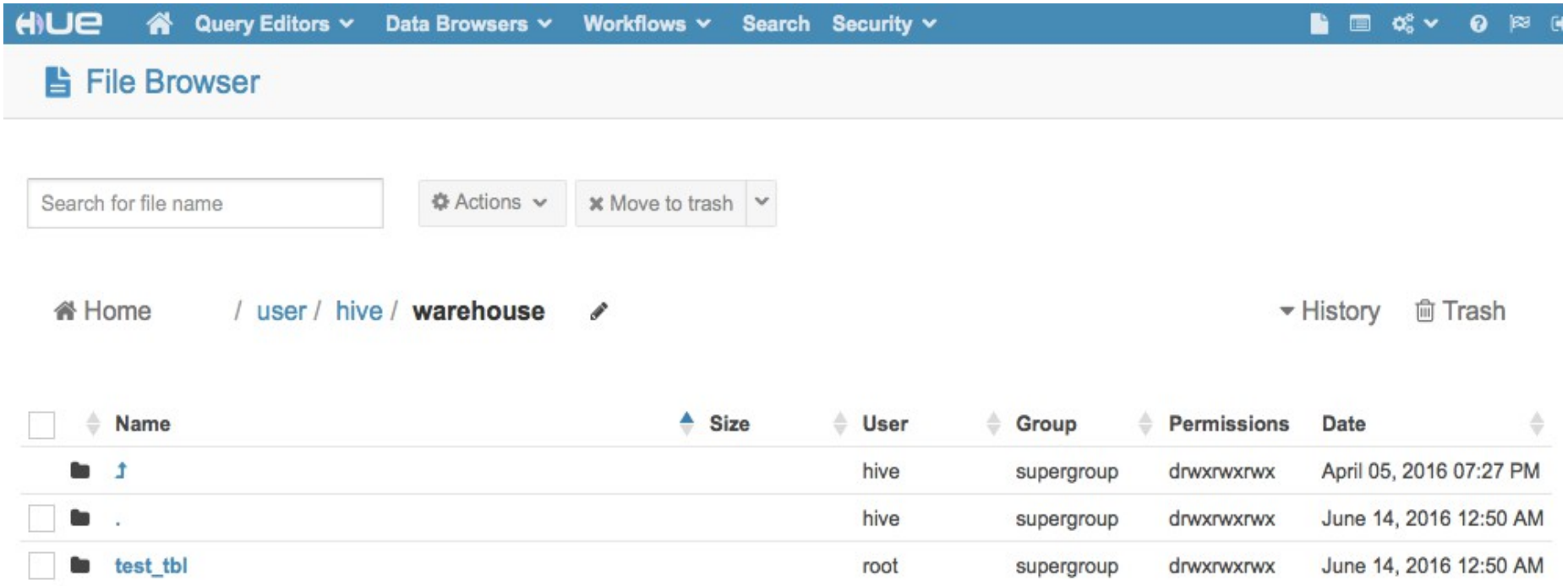
```
hive> quit;
```

Create Hive Table

```
hive> CREATE TABLE test_tbl(id INT, country STRING) ROW FORMAT DELIMITED
 FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.886 seconds
hive> show tables;
OK
test_tbl
Time taken: 0.125 seconds, Fetched: 1 row(s)
hive> describe test_tbl;
OK
id                      int
country                 string
Time taken: 0.115 seconds, Fetched: 2 row(s)
hive>
```

See also: https://cwiki.apache.org/Hive/languagemanual-ddl.html

Reviewing Hive Table in HDFS



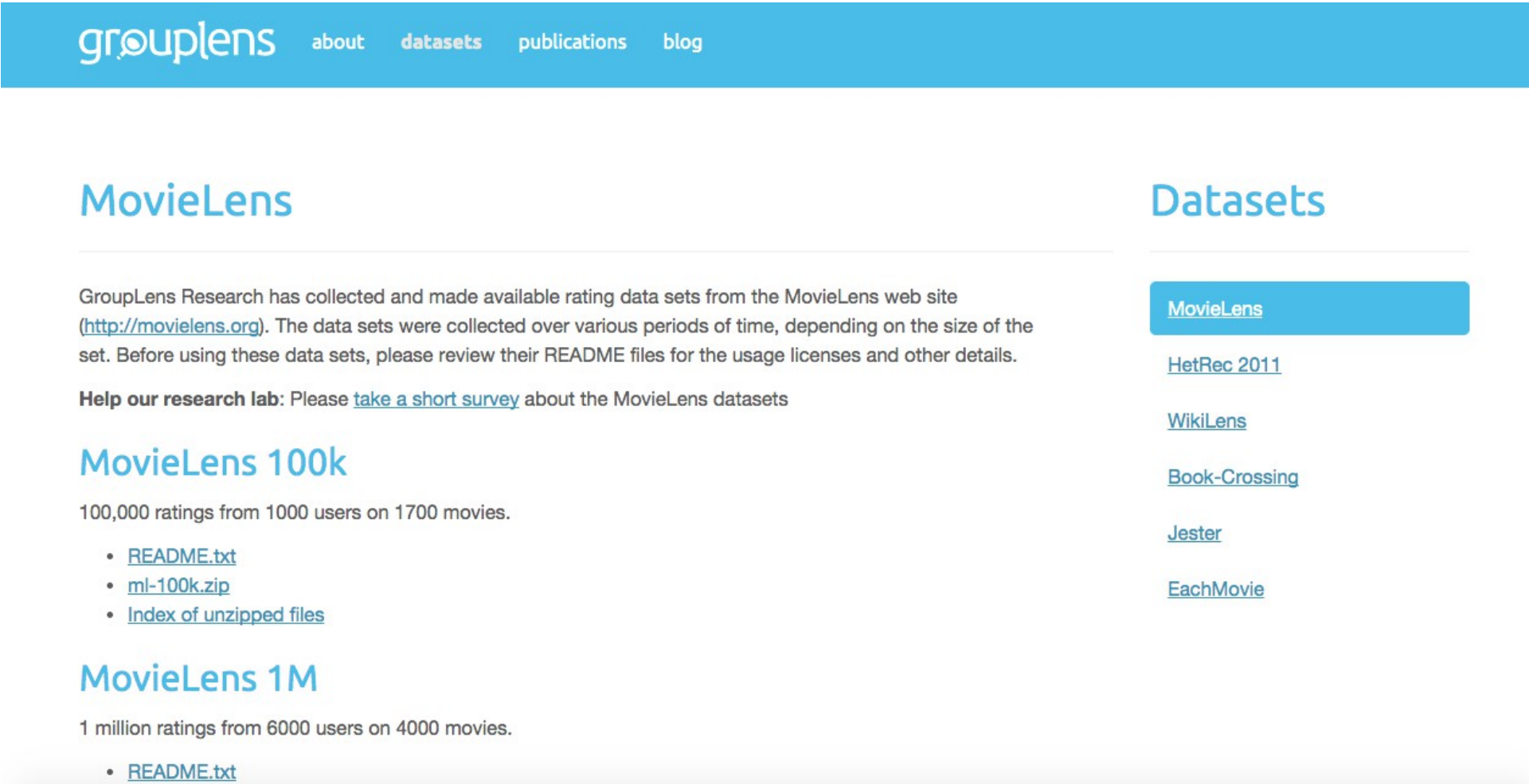
Alter and Drop Hive Table

```
Hive > alter table test_tbl add columns (remarks STRING);

hive > describe test_tbl;
OK
id   int
country  string
remarks  string
Time taken: 0.077 seconds
hive > drop table test_tbl;
OK
Time taken: 0.9 seconds
```

See also: https://cwiki.apache.org/Hive/adminmanual-metastoreadmin.html

Preparing Large Dataset

http://grouplens.org/datasets/movielens/



MovieLen Dataset

1) Type command > wget http://files.grouplens.org/datasets/movielens/ml-100k.zip

2) Type command > yum install unzip

3) Type command > unzip ml-100k.zip

4) Type command > more ml-100k/u.user

```
[root@quickstart guest1]# more ml-100k/u.user
1|24|M|technician|85711
2|53|F|other|94043
3|23|M|writer|32067
4|24|M|technician|43537
5|33|F|other|15213
6|42|M|executive|98101
7|57|M|administrator|91344
8|36|M|administrator|05201
9|29|M|student|01002
10|53|M|lawyer|90703
11|39|F|other|30329
```

Moving dataset to HDFS

1) Type command > `cd ml-100k`
2) Type command > `hadoop fs -mkdir /user/cloudera/movielens`
3) Type command > `hadoop fs -put u.user /user/cloudera/movielens`
4) Type command > `hadoop fs -ls /user/cloudera/movielens`

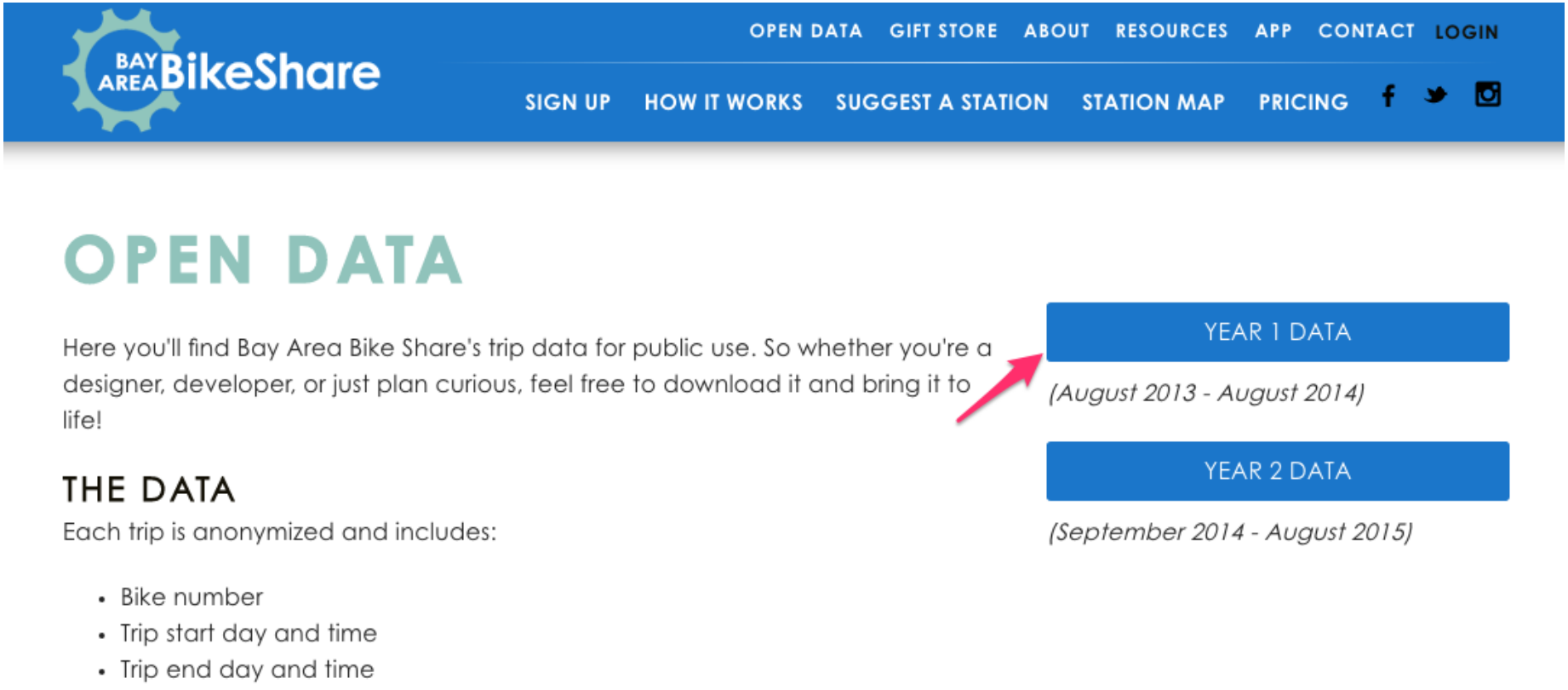
```
[root@quickstart ml-100k]# hadoop fs -ls /user/cloudera/movielens
Found 1 items
-rw-r--r--   1 root cloudera      22628 2016-06-14 08:04 /user/cloudera/
movielens/u.user
[root@quickstart ml-100k]#
```

CREATE & SELECT Table

```
hive> CREATE EXTERNAL TABLE users (userid INT,  age INT,
    >   gender STRING, occupation STRING, zipcode STRING) ROW FORMAT
    > DELIMITED FIELDS TERMINATED BY '|' STORED AS TEXTFILE
    > LOCATION '/user/cloudera/movielens';
OK
Time taken: 0.646 seconds
hive> SELECT * FROM users;
OK
1       24      M       technician      85711
2       53      F       other   94043
3       23      M       writer  32067
4       24      M       technician      43537
5       33      F       other   15213
6       42      M       executive       98101
7       57      M       administrator   91344
8       36      M       administrator   05201
```

Bay Area Bike Share (BABS)

http://www.bayareabikeshare.com/open-data



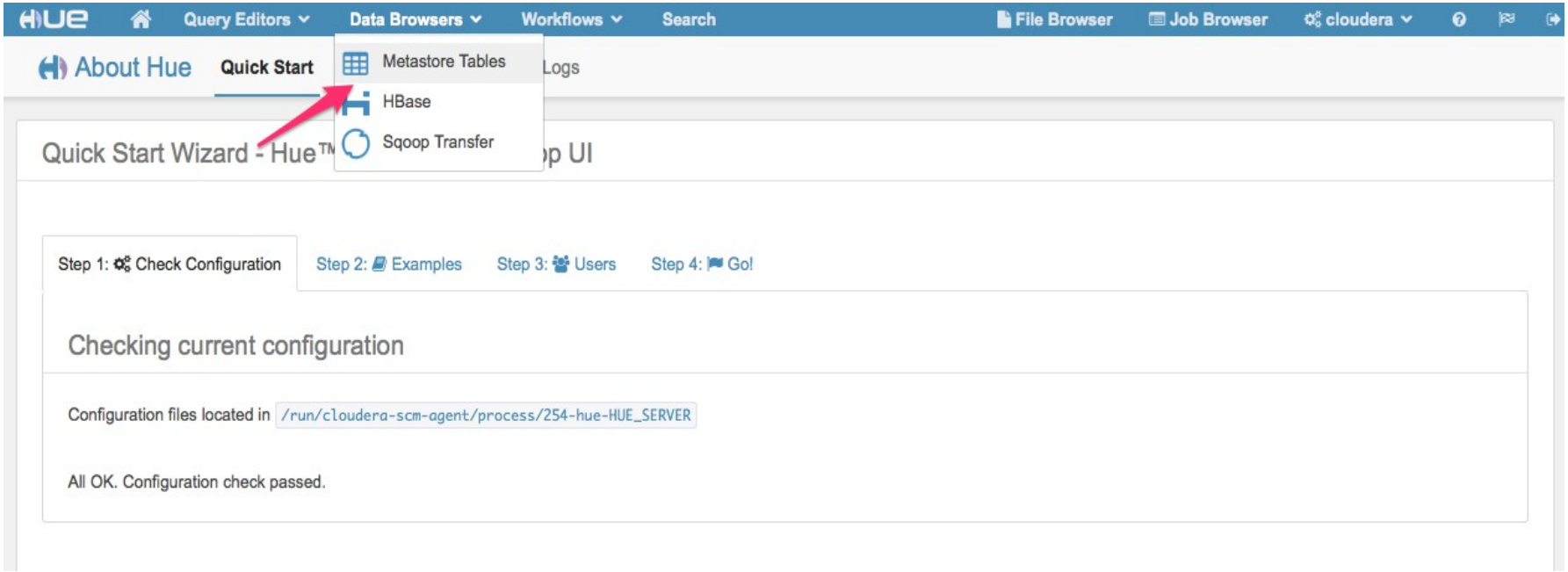
Preparing a bike data

```
$wget https://s3.amazonaws.com/babs-open-data/
babs_open_data_year_1.zip
$unzip babs_open_data_year_1.zip
$cd 201402_babs_open_data/
$hadoop fs -put 201402_trip_data.csv
/user/cloudera
$ hadoop fs -ls /user/cloudera
```

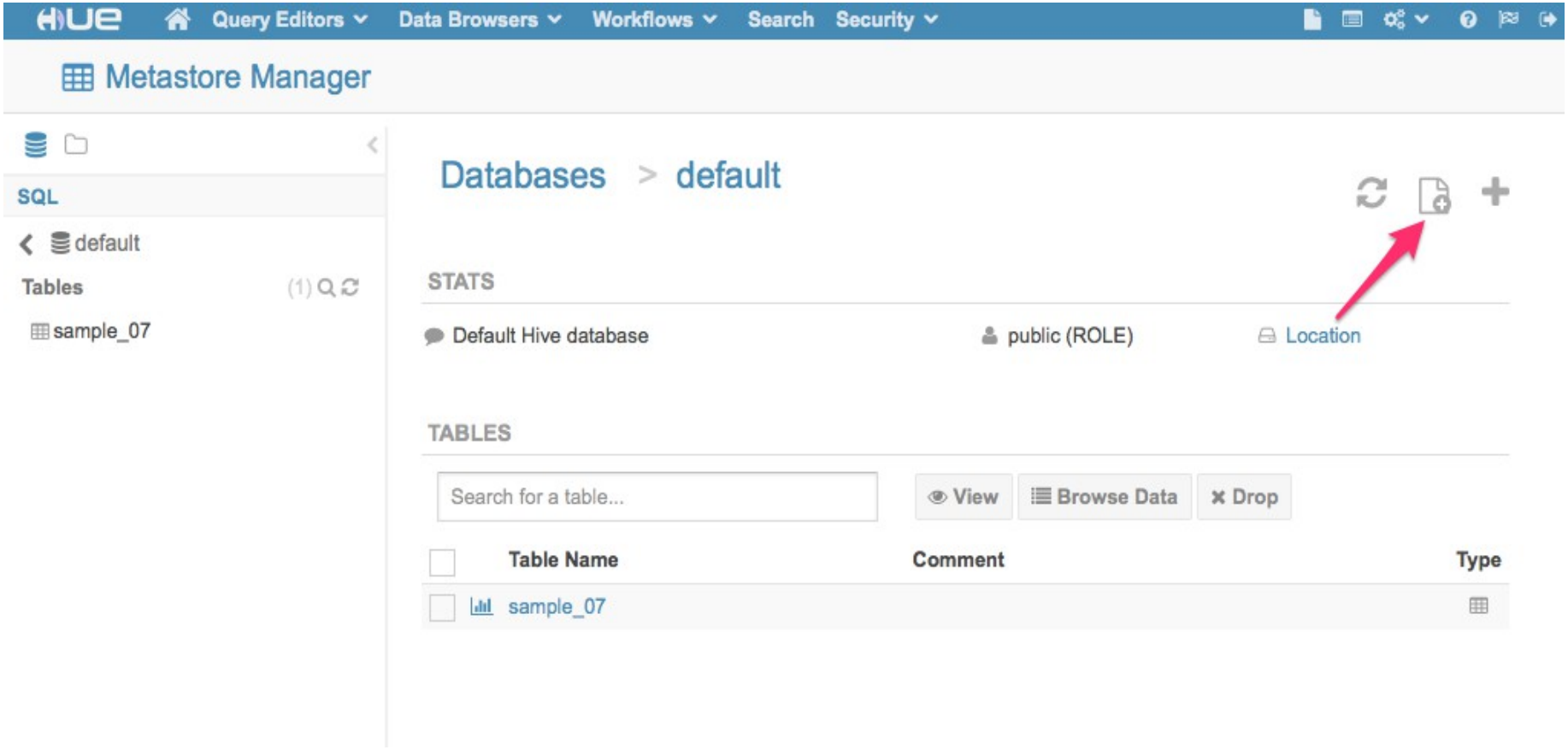
```
[root@quickstart 201402_babs_open_data]# hadoop fs -ls /user/cloudera
Found 4 items
-rw-r--r--   1 root     cloudera   17219022 2016-06-14 08:13 /user/cloudera/201402_t
rip_data.csv
drwxr-xr-x   - cloudera cloudera          0 2016-06-14 04:29 /user/cloudera/input
drwxr-xr-x   - root     cloudera          0 2016-06-14 08:04 /user/cloudera/movielen
s
drwxr-xr-x   - cloudera cloudera          0 2016-06-14 04:32 /user/cloudera/output
[root@quickstart 201402_babs_open_data]#
```

Importing CSV Data with the Metastore App

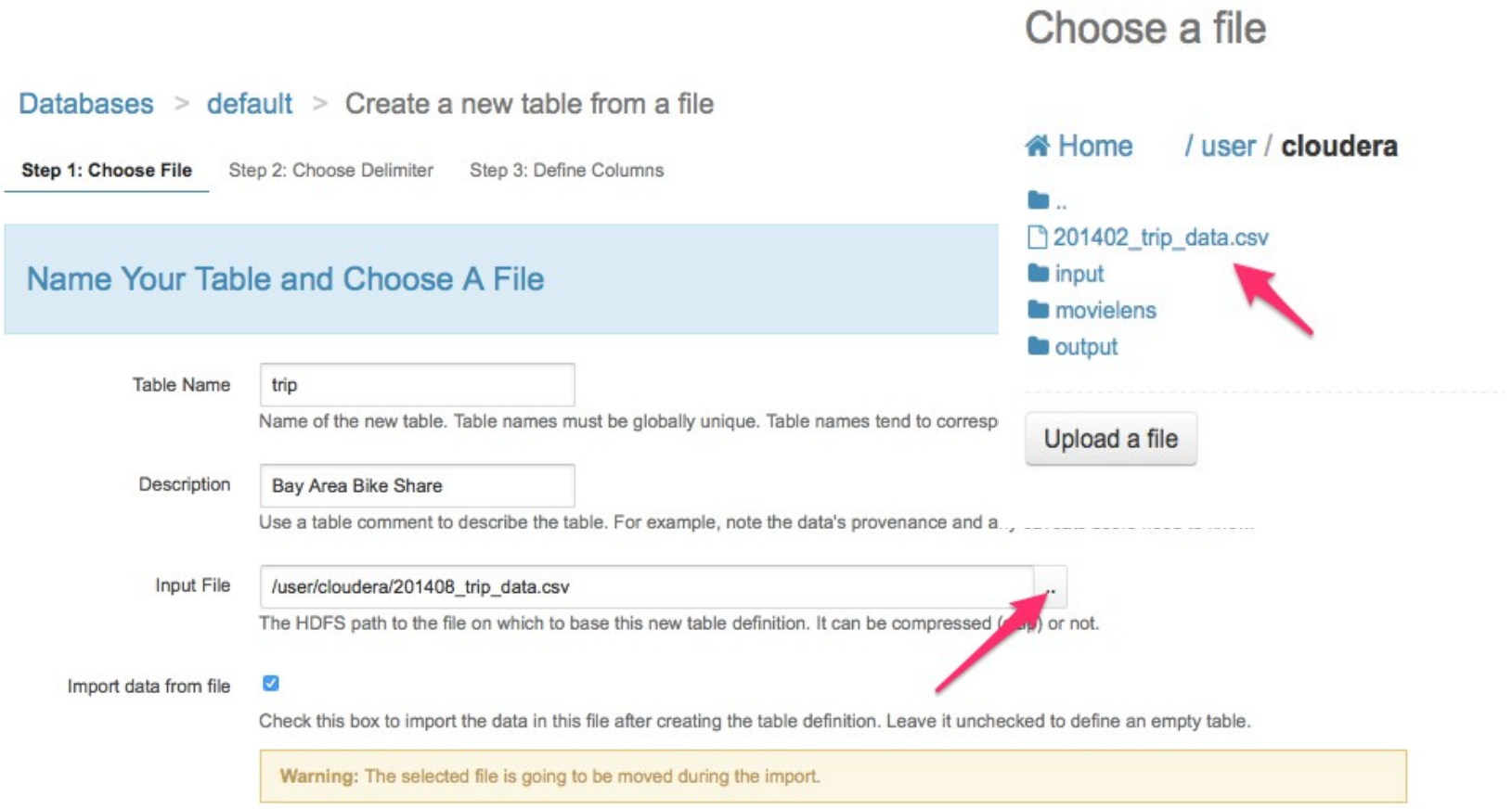
The BABS data set contains 4 CSVs that contain data for stations, trips, rebalancing (availability), and weather. We will import **trips** dataset using Metastore Tables



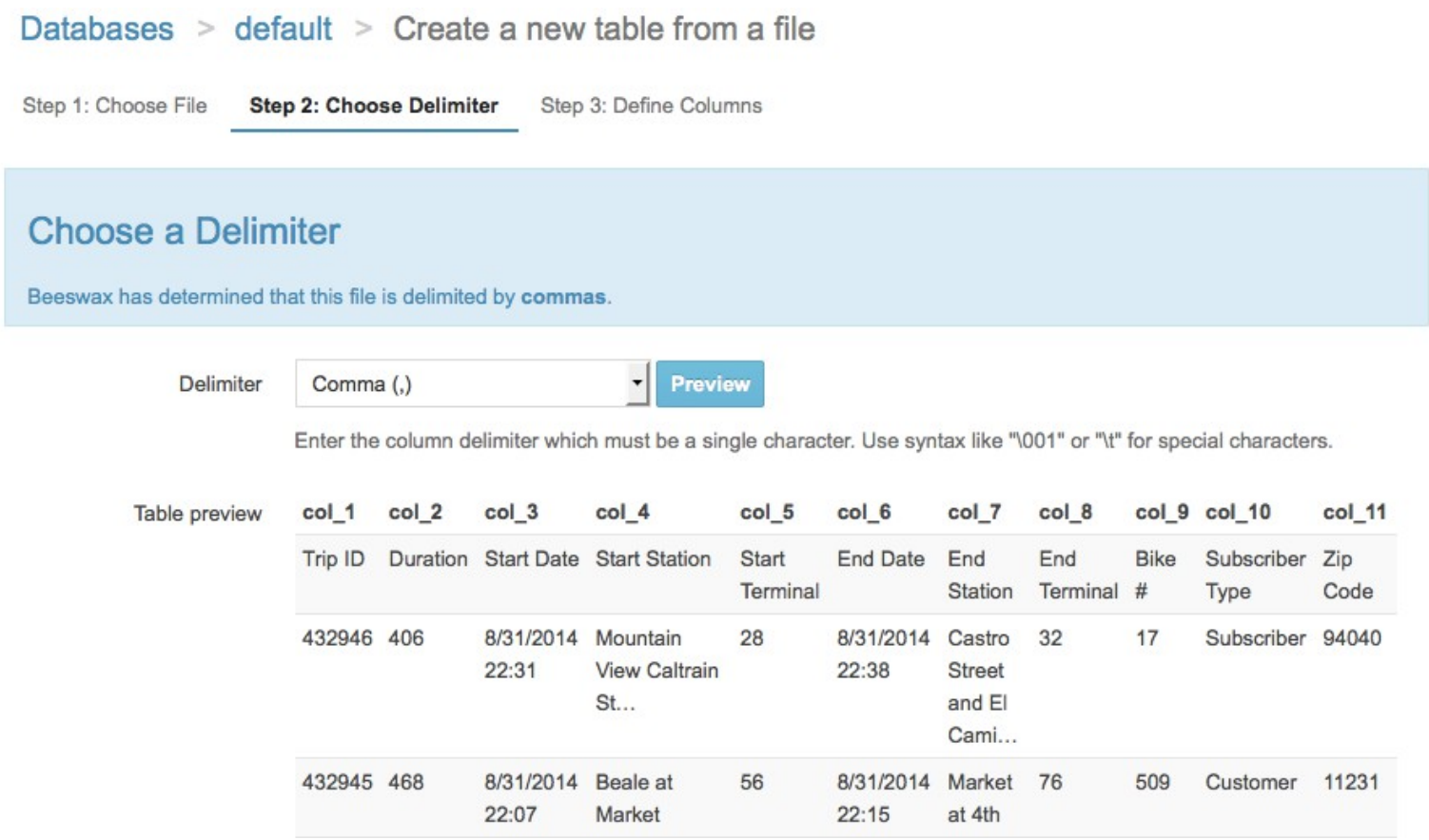
Select: Create a new table from a file



Name a table and select a file

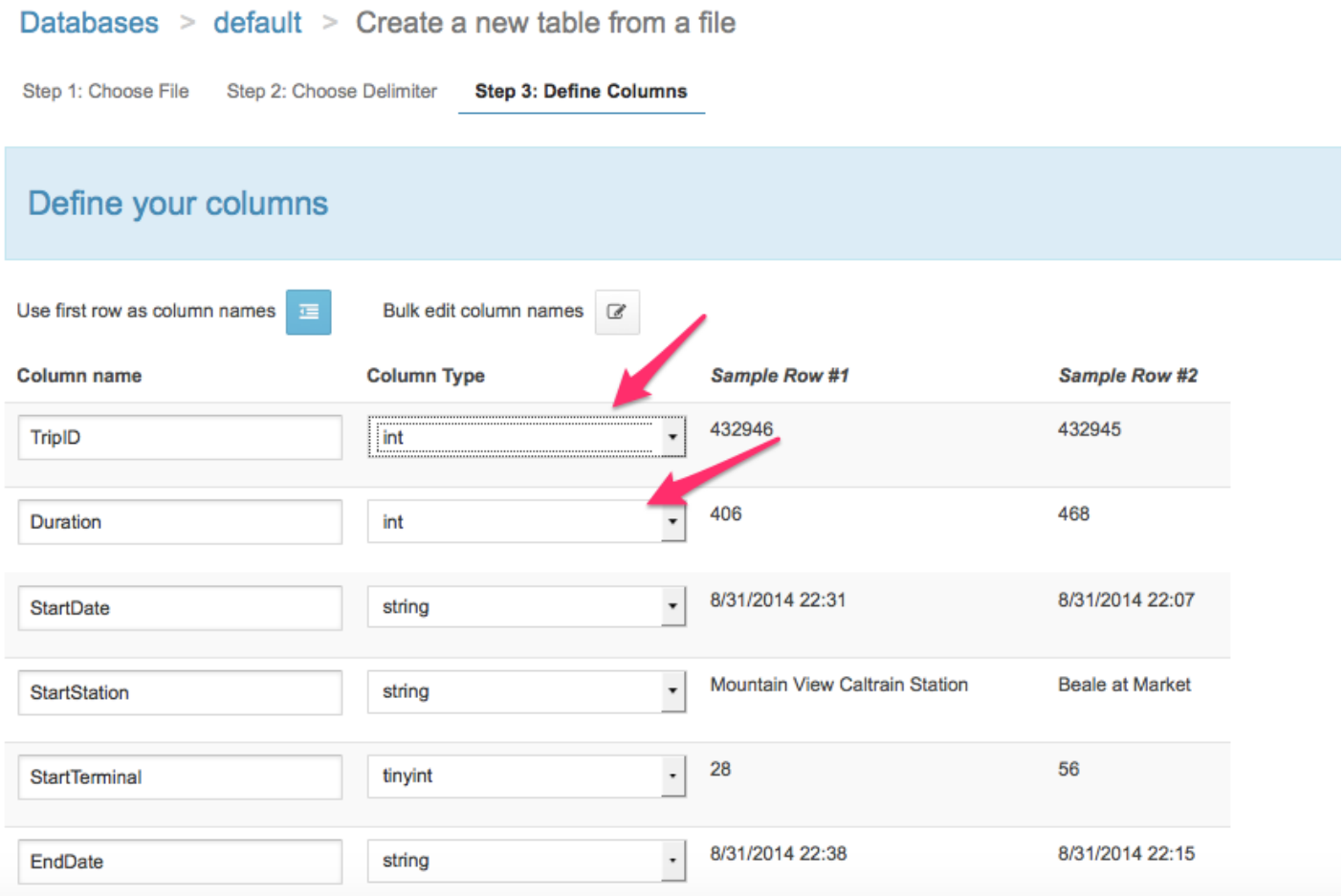


Choose Delimiter



col_1	col_2	col_3	col_4	col_5	col_6	col_7	col_8	col_9	col_10	col_11
Trip ID	Duration	Start Date	Start Station	Start Terminal	End Date	End Station	End Terminal	Bike #	Subscriber Type	Zip Code
432946	406	8/31/2014 22:31	Mountain View Caltrain St…	28	8/31/2014 22:38	Castro Street and El Cami…	32	17	Subscriber	94040
432945	468	8/31/2014 22:07	Beale at Market	56	8/31/2014 22:15	Market at 4th	76	509	Customer	11231

Define Column Types



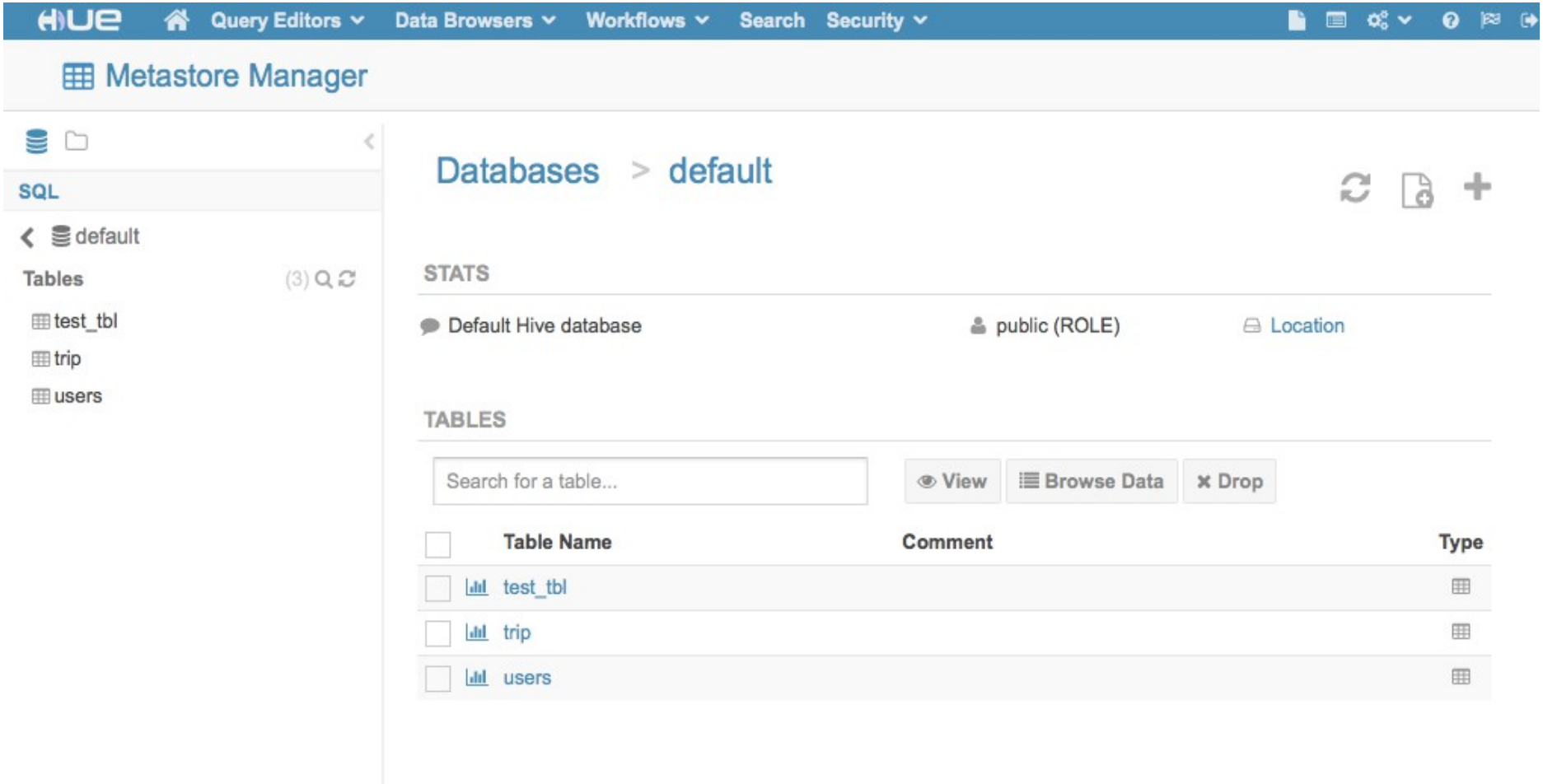
Create Table : Done

Databases > default > trip

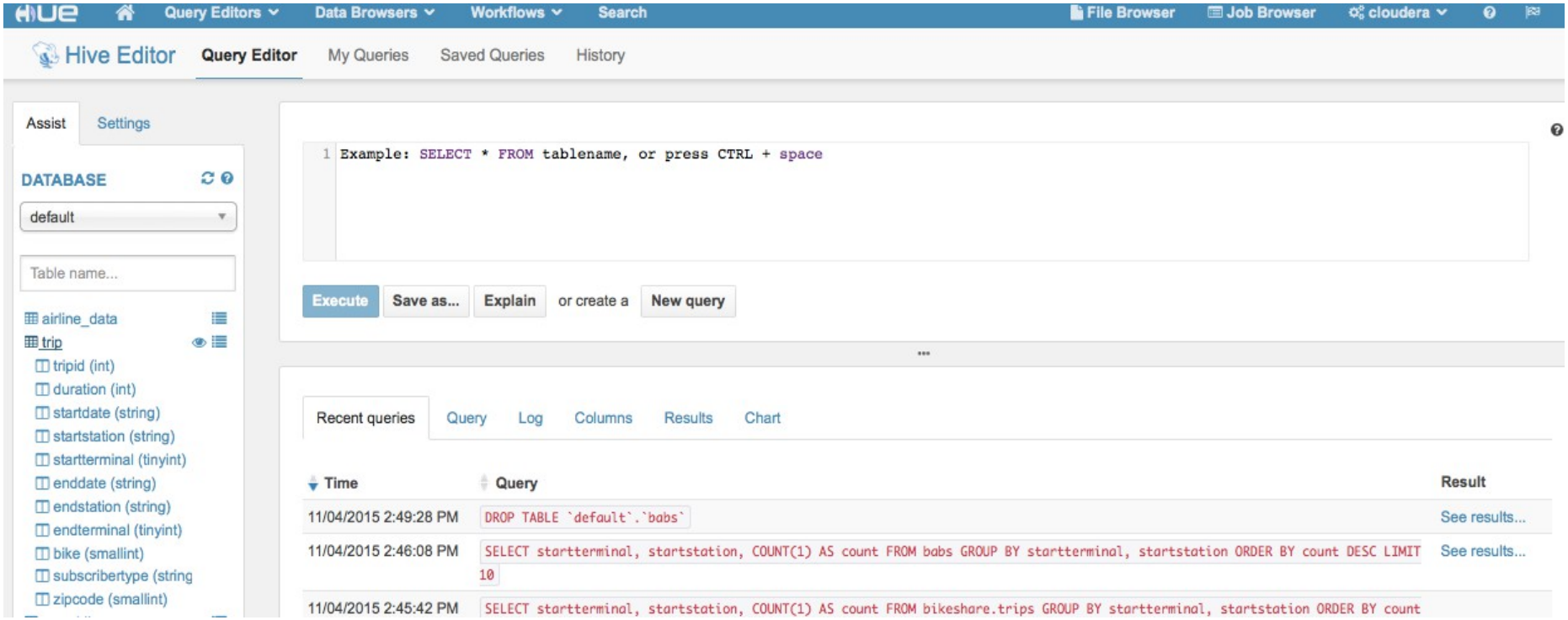
Comment: Bay Area Bike Share

Columns | Sample | Properties

	Name	Type	Comment
0	tripid	int	
1	duration	int	
2	startdate	string	
3	startstation	string	
4	startterminal	tinyint	
5	enddate	string	
6	endstation	string	
7	endterminal	tinyint	
8	bike	smallint	
9	subscribertype	string	
10	zipcode	smallint	

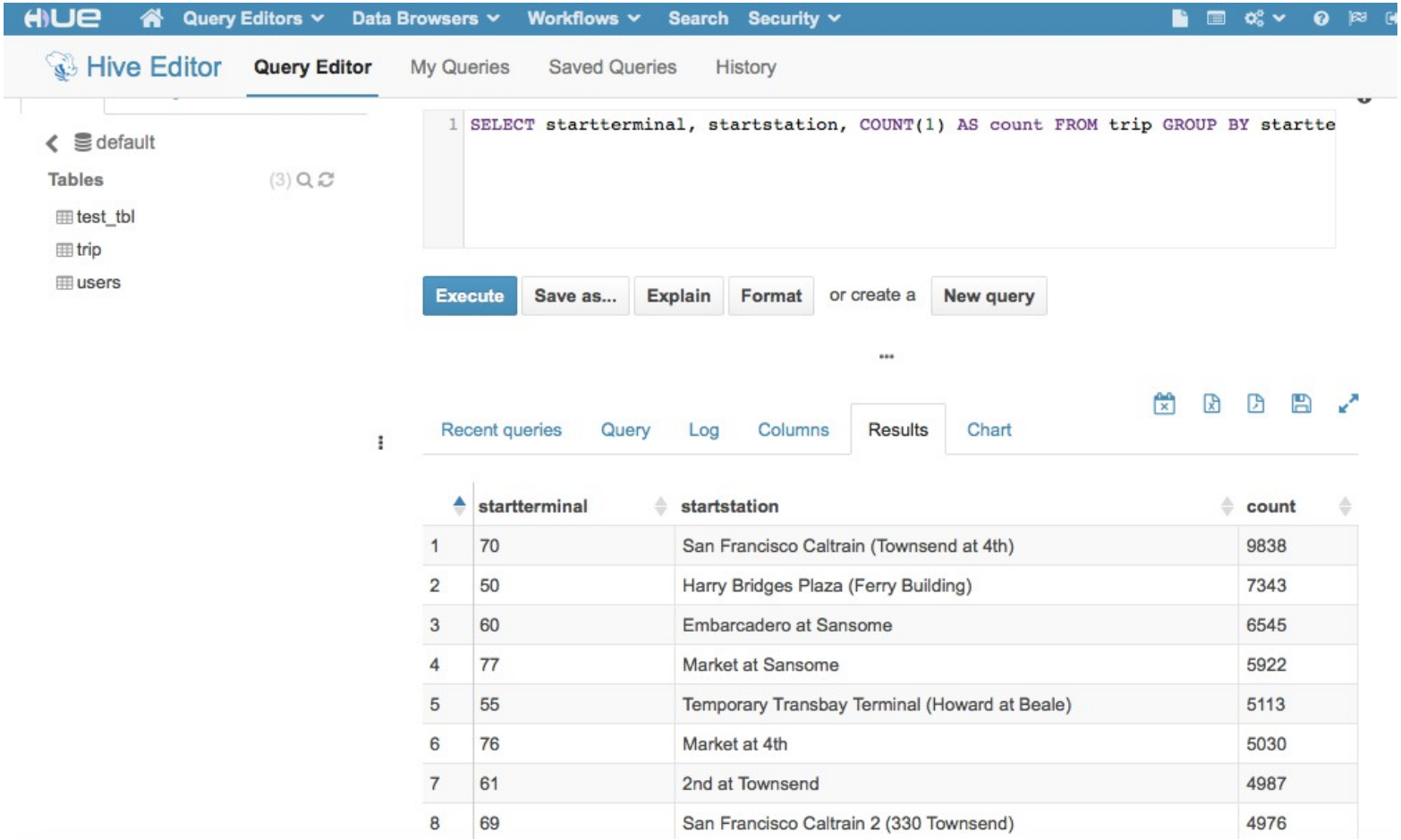

HUE
Query Editors
Data Browsers
Workflows
Search
Security
Metastore Manager
SQL
default
Tables (3)
test_tbl
trip
users
Databases > default
STATS
Default Hive database
public (ROLE)
Location
TABLES
Search for a table...
View
Browse Data
Drop
Table Name
Comment
Type
test_tbl
trip
users

Starting Hive Editor



Find the top 10 most popular start stations based on the trip data

```
SELECT startterminal, startstation, COUNT(1) AS count FROM trip
GROUP BY startterminal, startstation ORDER BY count DESC LIMIT 10
```



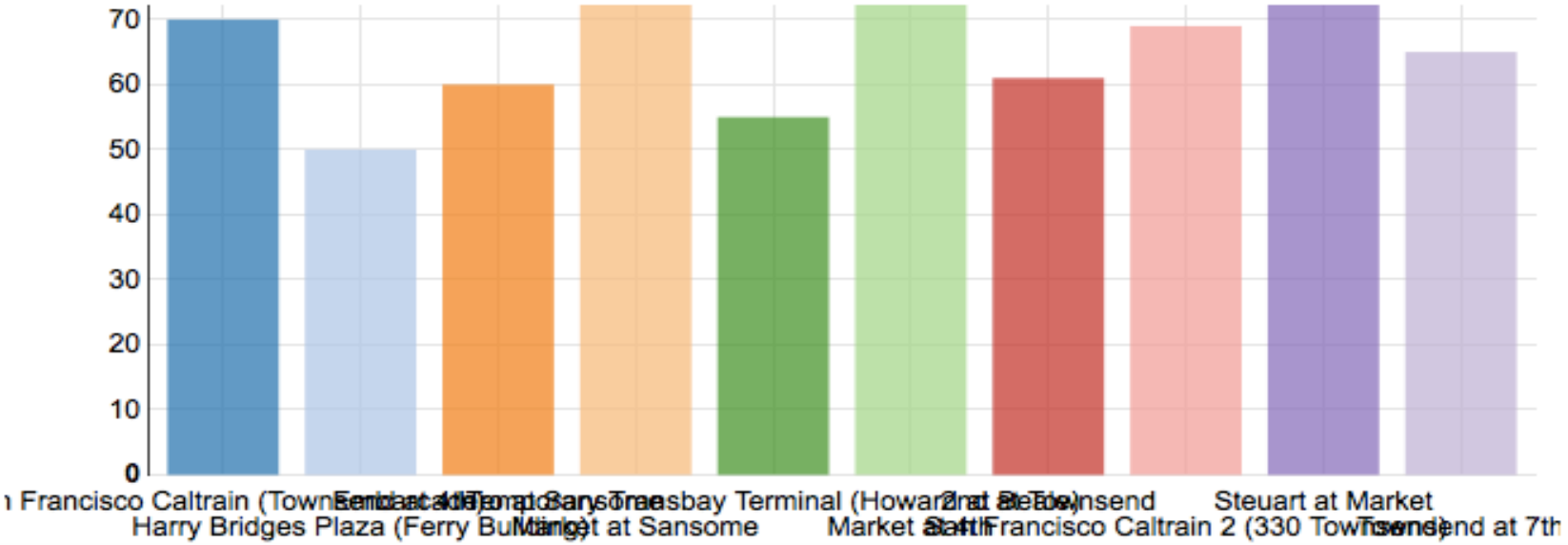
	startterminal	startstation	count
1	70	San Francisco Caltrain (Townsend at 4th)	9838
2	50	Harry Bridges Plaza (Ferry Building)	7343
3	60	Embarcadero at Sansome	6545
4	77	Market at Sansome	5922
5	55	Temporary Transbay Terminal (Howard at Beale)	5113
6	76	Market at 4th	5030
7	61	2nd at Townsend	4987
8	69	San Francisco Caltrain 2 (330 Townsend)	4976

```
1 SELECT startterminal, startstation, COUNT(1) AS count FROM trip GROUP BY startte
```

Execute | Save as... | Explain | Format | or create a | New query

...

Recent queries | Query | Log | Columns | Results | Chart



Introduction

A high-level platform for creating MapReduce programs Using Hadoop

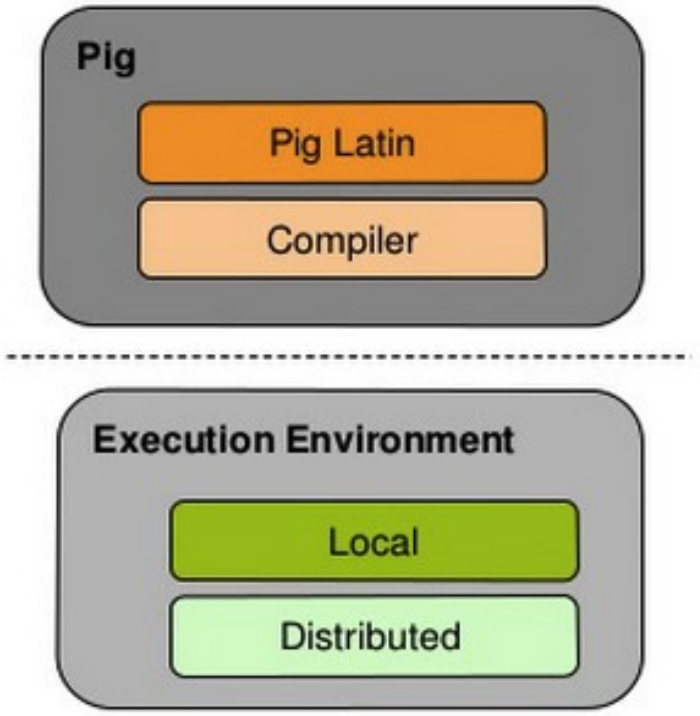
Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

Pig Components

- **Two Compnents**
  - Language (Pig Latin)
  - Compiler
- **Two Execution Environments**
  - **Local**

    pig -x local
  - **Distributed**

    pig -x mapreduce



Hive.apache.org

Running Pig

- **Script**

  pig myscript
- **Command line (Grunt)**

  pig
- **Embedded**

  Writing a java program

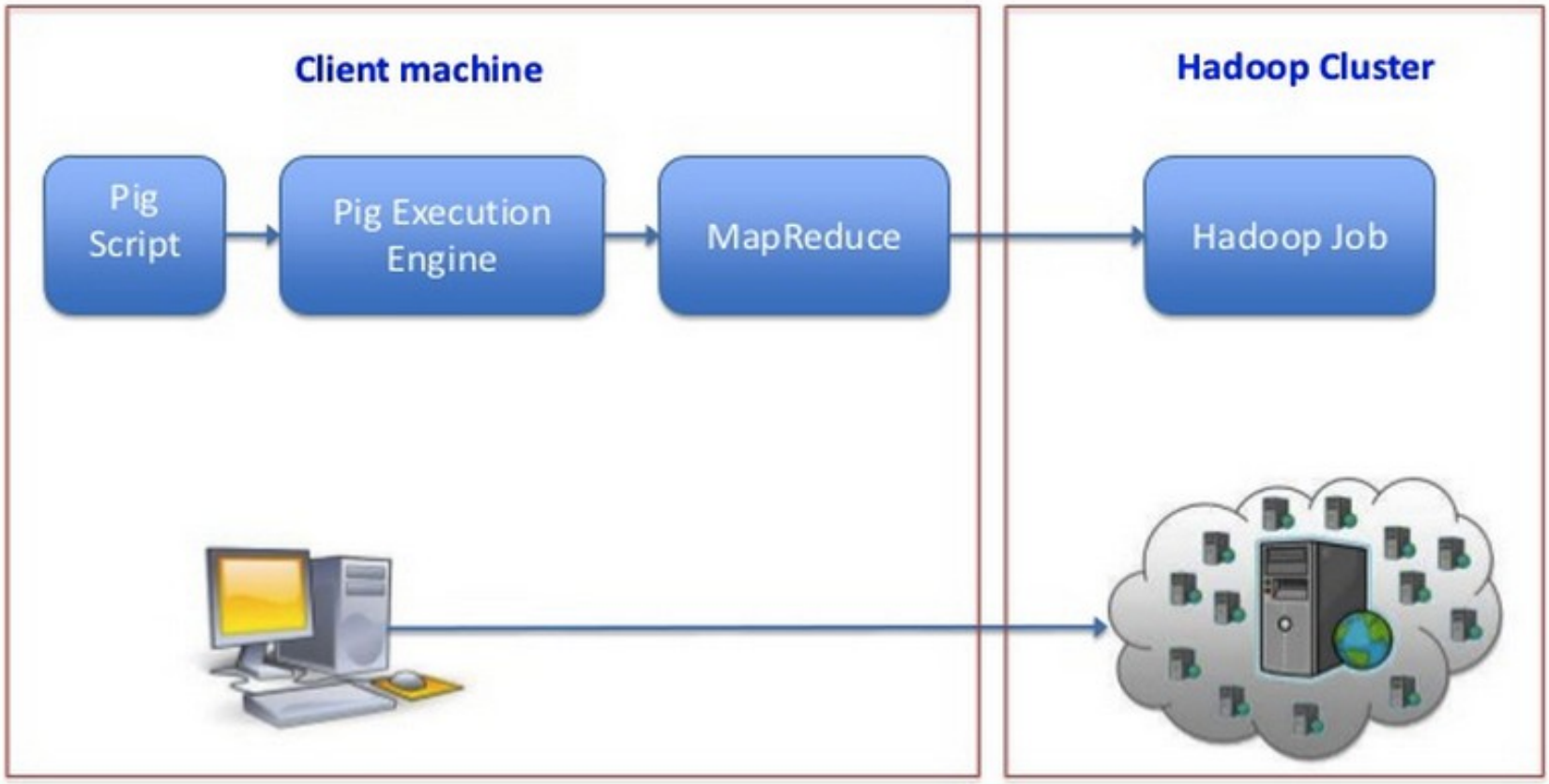
Hive.apache.org

Pig Latin

```
Users = load'users'as (name, age);
Fltrd = filter Users by
        age >= 18 and age <= 25;
Pages = load 'pages' as (user, url);
Jnd = joinFltrdby name, Pages by user;
Grpd = groupJndbyurl;
Smmd = foreachGrpdgenerate group,
COUNT(Jnd) as clicks;
Srtd = orderSmmdby clicks desc;
Top5 = limitSrtd 5;
store Top5 into'top5sites';
```

Hive.apache.org

Pig Execution Stages



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

Hive.apache.org

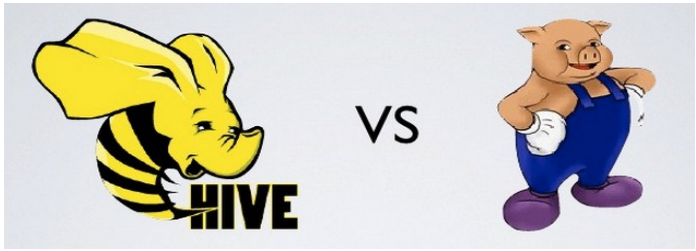
Why Pig?

- **Makes writing Hadoop jobs easier**
  - 5% of the code, 5% of the time
  - You don't need to be a programmer to write Pig scripts
- **Provide major functionality required for DatawareHouse and Analytics**
  - *Load, Filter, Join, Group By, Order, Transform*
- **User can write custom UDFs (User Defined Function)**

Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

Hive.apache.org

Pig v.s. Hive



Characteristic	Pig	Hive
Developed by	Yahoo!	Facebook
Language name	Pig Latin	HiveQL
Type of language	Data flow	Declarative (SQL dialect)
Data structures it operates on	Complex, nested	
Schema optional?	Yes	No, but data can have many schemas
Relational complete?	Yes	Yes
Turing complete?	Yes when extended with Java UDFs	Yes when extended with Java UDFs

Hive.apache.org

Hands-On: Running a Pig script

Starting Pig Command Line

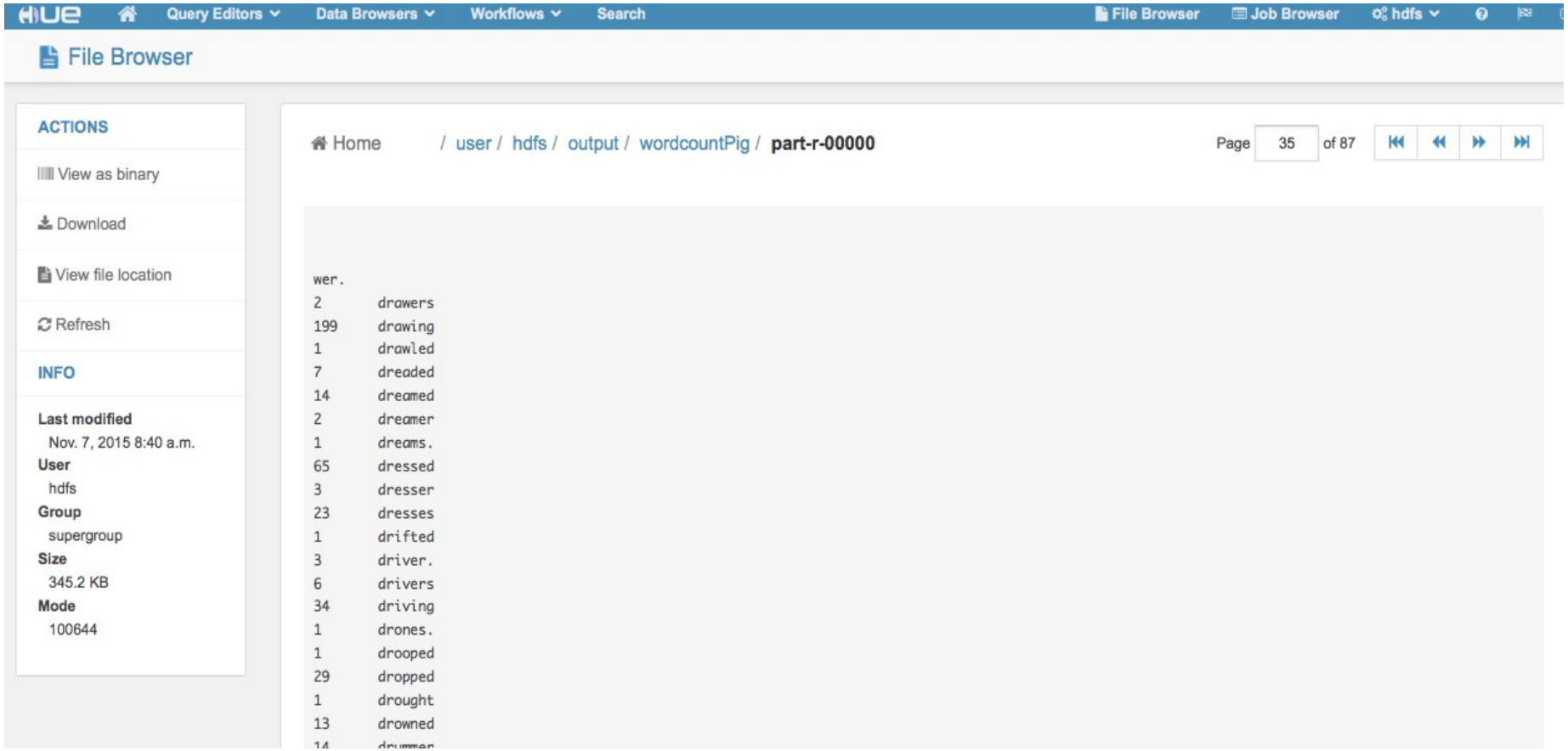
```
$  pig -x mapreduce

2013-08-01 10:29:00,027 [main] INFO  org.apache.pig.Main - Apache Pig
version 0.11.1 (r1459641) compiled Mar 22 2013, 02:13:53

2013-08-01 10:29:00,027 [main] INFO  org.apache.pig.Main - Logging error
messages to: /home/hdadmin/pig_1375327740024.log

2013-08-01 10:29:00,066 [main] INFO  org.apache.pig.impl.util.Utils -
Default bootup file /home/hdadmin/.pigbootup not found

2013-08-01 10:29:00,212 [main] INFO
org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting
to hadoop file system at: file:///

grunt>
```

Writing a Pig Script for wordcount

```
A = load '/user/cloudera/input/*';
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as
word;
C = group B by word;
D = foreach C generate COUNT(B), group;
store D into '/user/cloudera/output/wordcountPig';
```

```
2016-06-14 08:29:25,835 [main] INFO  org.apache.pig.backend.hadoop.executionengine.m
apReduceLayer.MapReduceLauncher - More information at: http://localhost:50030/jobdet
ails.jsp?jobid=job_1465875170640_0004
2016-06-14 08:29:25,871 [main] INFO  org.apache.pig.backend.hadoop.executionengine.m
apReduceLayer.MapReduceLauncher - 0% complete
2016-06-14 08:29:41,625 [main] INFO  org.apache.pig.backend.hadoop.executionengine.m
apReduceLayer.MapReduceLauncher - 50% complete
2016-06-14 08:29:51,359 [main] INFO  org.apache.pig.backend.hadoop.executionengine.m
apReduceLayer.MapReduceLauncher - 100% complete
2016-06-14 08:29:51,362 [main] INFO  org.apache.pig.tools.pigstats.SimplePigStats -
Script Statistics:

HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.7.0  0.12.0-cdh5.7.0 root    2016-06-14 08:29:18     2016-06-14 08:29:51G
ROUP_BY

Success!
```


Thanachart Numnonda, thanachart@imcinstitute.com July 2016


Query Editors
Data Browsers
Workflows
Search
File Browser
Job Browser
hdfs
File Browser
ACTIONS
View as binary
Download
View file location
Refresh
INFO
Last modified
Nov. 7, 2015 8:40 a.m.
User
hdfs
Group
supergroup
Size
345.2 KB
Mode
100644
Home / user / hdfs / output / wordcountPig / part-r-00000
Page 35 of 87
wer.
2 drawers
199 drawing
1 drawled
7 dreaded
14 dreamed
2 dreamer
1 dreams.
65 dressed
3 dresser
23 dresses
1 drifted
3 driver.
6 drivers
34 driving
1 drones.
1 drooped
29 dropped
1 drought
13 drowned

Impala

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

Introduction

open source massively parallel processing (MPP) SQL query engine

Cloudera Impala is a query engine that runs on Apache Hadoop. Impala brings scalable parallel database technology to Hadoop, enabling users to issue low-latency SQL queries to data stored in HDFS and Apache HBase without requiring data movement or transformation.

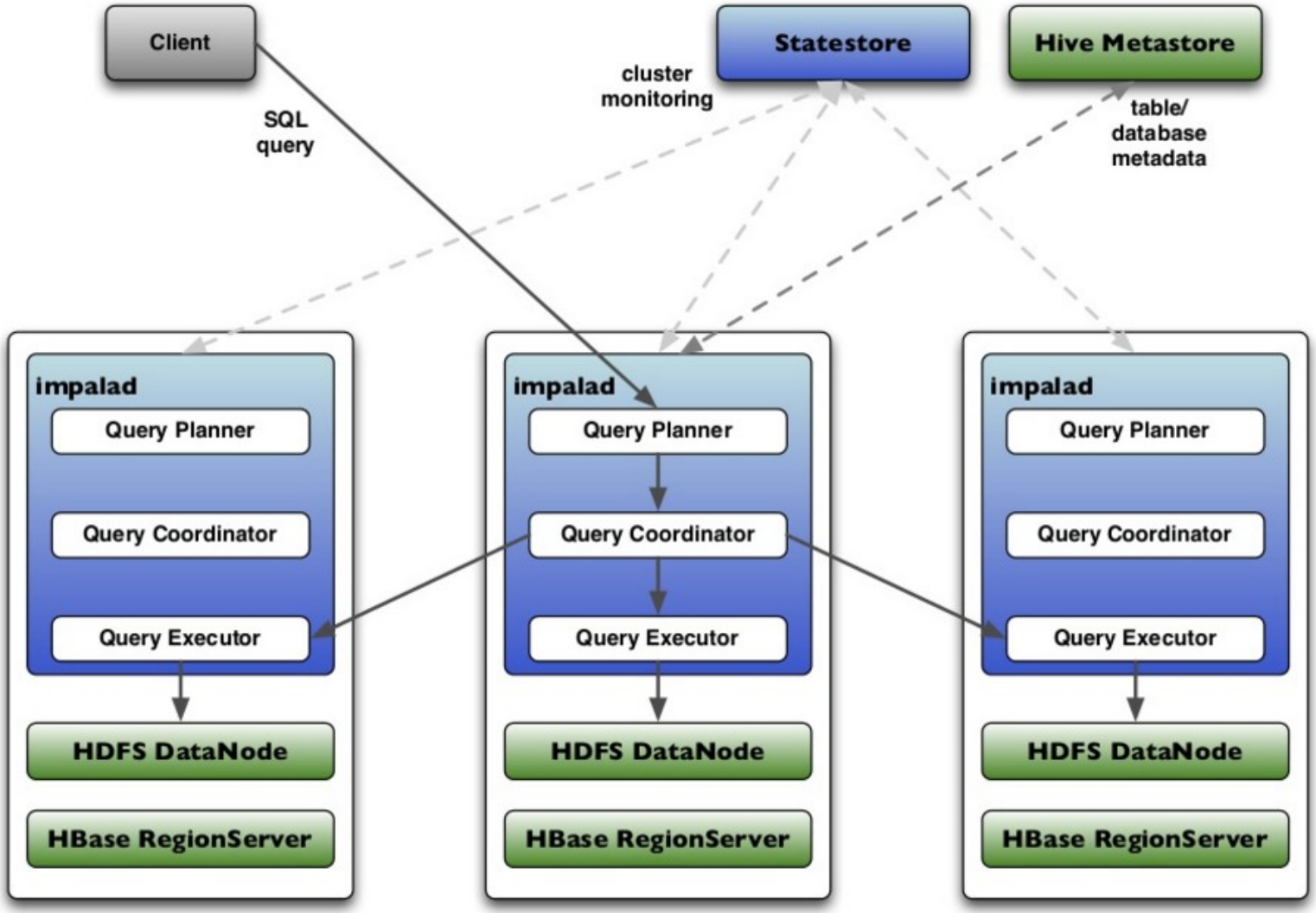
What is Impala?

- General--- purpose SQL engine
- Real--time queries in Apache Hadoop
- Opensource under Apache License
- Runs directly within Hadoop
- High performance
  - C++ instead of Java
  - Runtime code generator
  - Roughly 4-100 x Hive

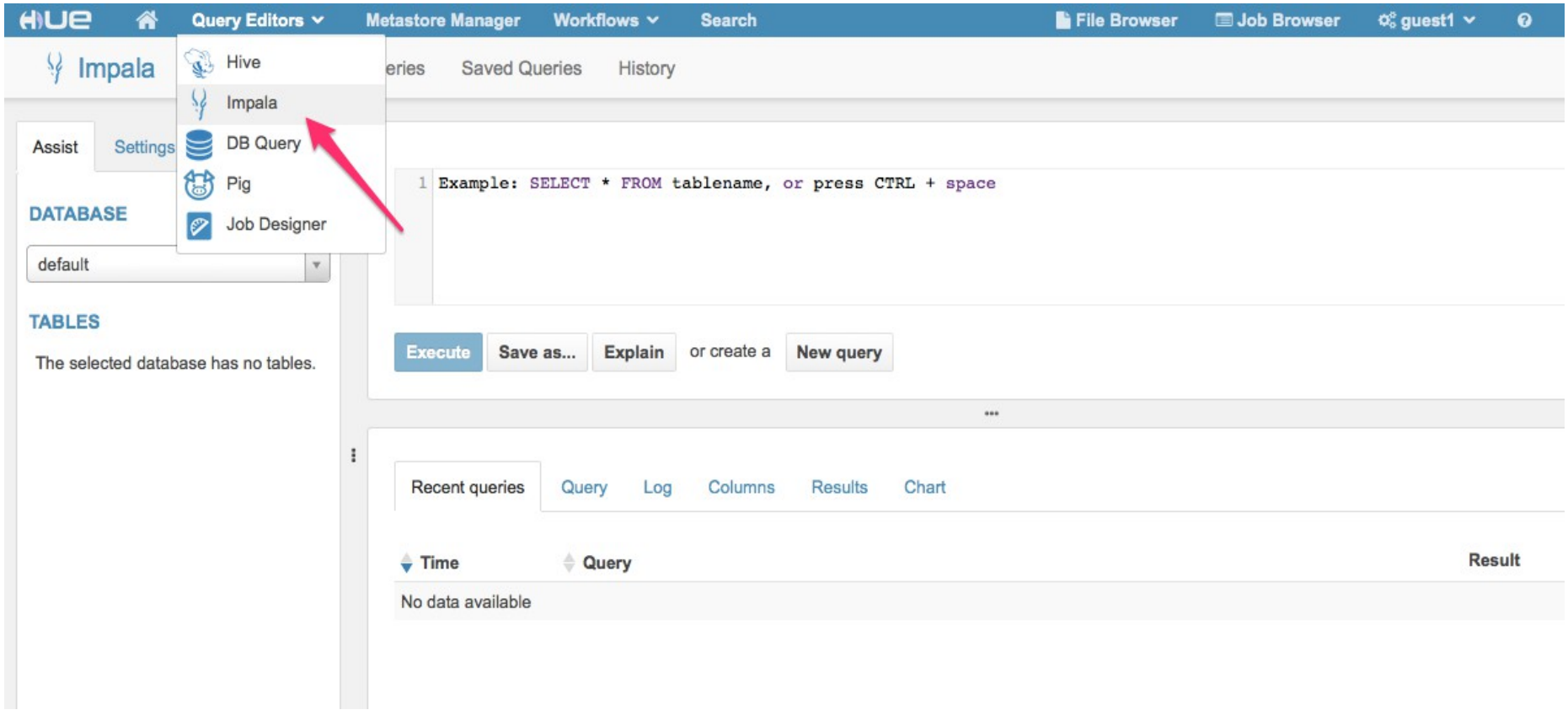
Impala Overview

- Impala daemon run on HDFS nodes
- Statestore (for cluster metadata) v.s. Metastore (for database metastore)
- Queries run on “revelants” nodes
- Support common HDFS file formats
- Submit quries via Hue/Beeswax
- No fault tolerant

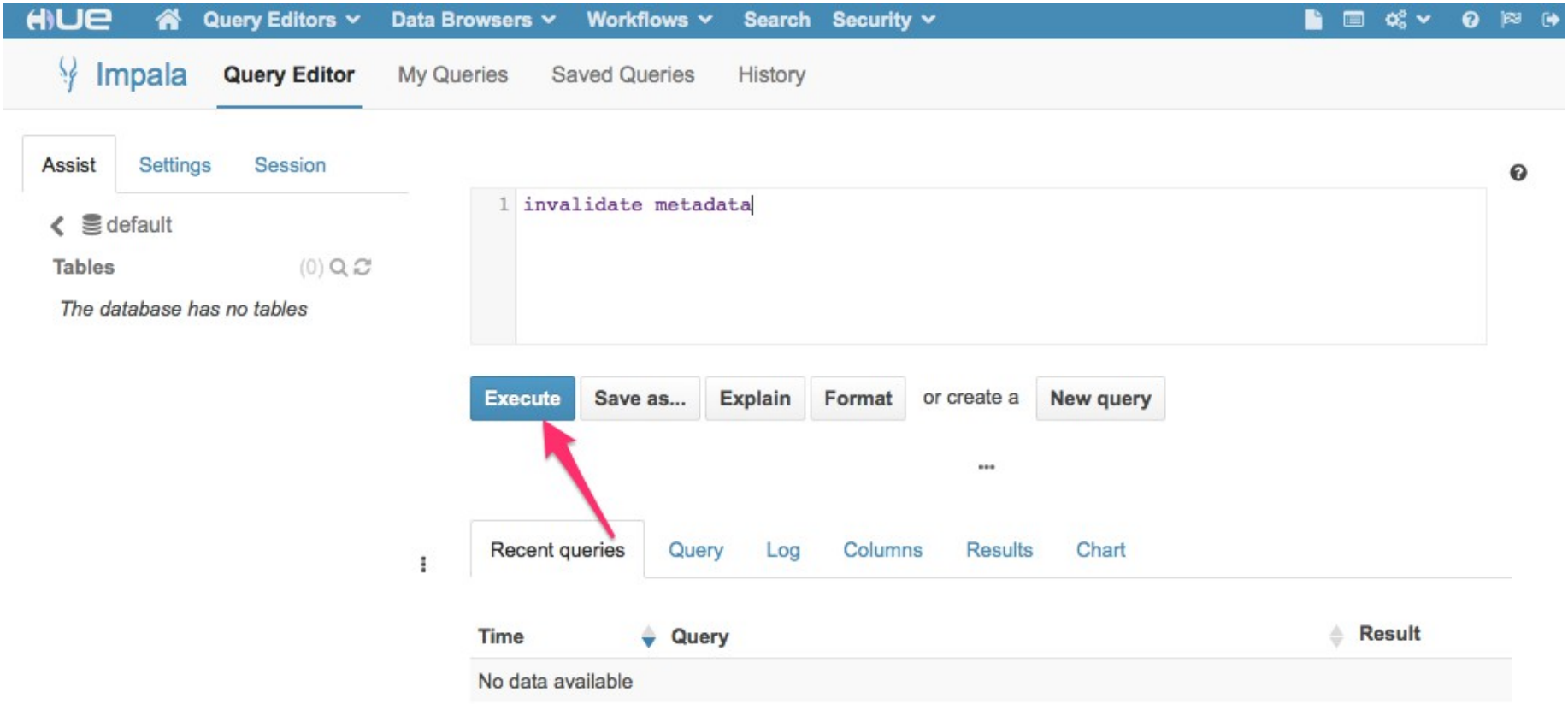
Impala Architecture



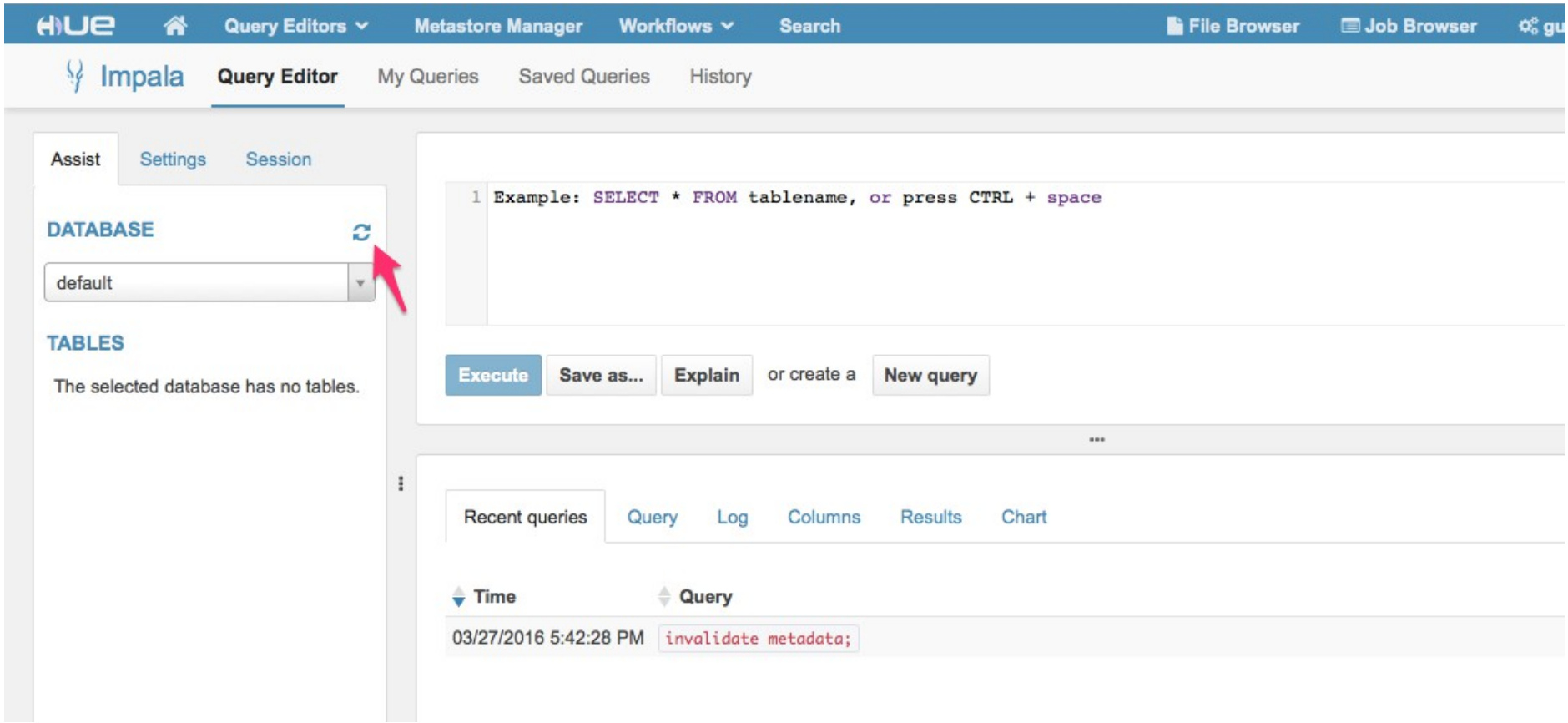
Start Impala Query Editor



Update the list of tables/metadata by excute the command invalidate metadata

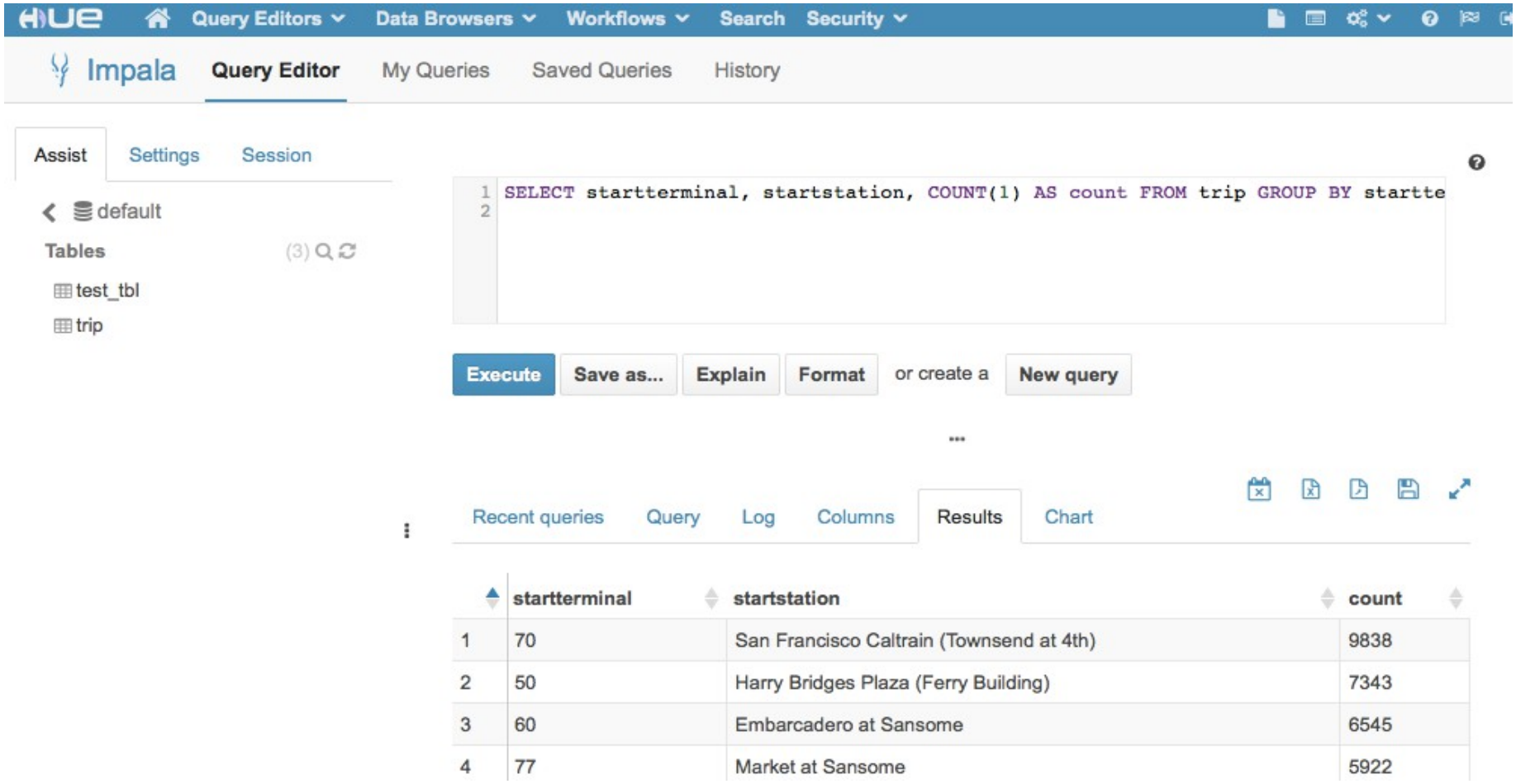


Restart Impala Query Editor and refresh the table list

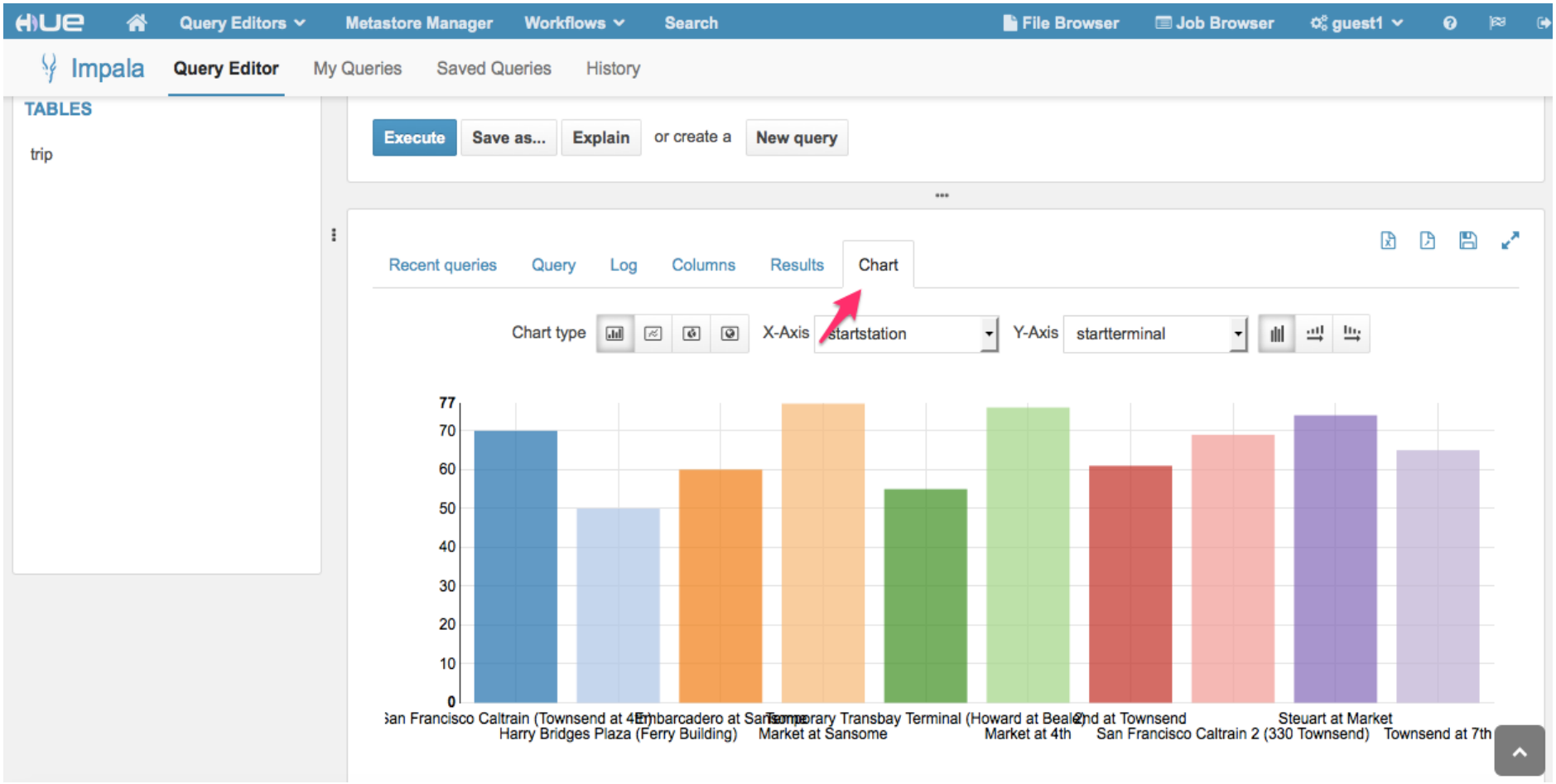


Find the top 10 most popular start stations based on the trip data: Using Impala

```
SELECT startterminal, startstation, COUNT(1) AS count FROM trip
GROUP BY startterminal, startstation ORDER BY count DESC LIMIT 10
```



	startterminal	startstation	count
1	70	San Francisco Caltrain (Townsend at 4th)	9838
2	50	Harry Bridges Plaza (Ferry Building)	7343
3	60	Embarcadero at Sansome	6545
4	77	Market at Sansome	5922


HUE
Query Editors
Metastore Manager
Workflows
Search
File Browser
Job Browser
guest1
Impala
Query Editor
My Queries
Saved Queries
History
TABLES
trip
Execute
Save as...
Explain
or create a
New query
Recent queries
Query
Log
Columns
Results
Chart
Chart type
X-Axis
startstation
Y-Axis
startterminal
77
70
60
50
40
30
20
10
0
Harry Bridges Plaza (Ferry Building)
Market at Sansome
Market at 4th
San Francisco Caltrain 2 (330 Townsend)
Steuart at Market
Townsend at 7th

Find the total number of trips and average duration (in minutes) of those trips, grouped by hour

```
SELECT
    hour,
    COUNT(1) AS trips,
    ROUND(AVG(duration) / 60) AS avg_duration
FROM (
    SELECT
        CAST(SPLIT(SPLIT(t.startdate, ' ')[1], ':')[0] AS INT) AS
hour,
        t.duration AS duration
    FROM `bikeshare`.`trips` t
    WHERE
        t.startterminal = 70
        AND
        t.duration IS NOT NULL
    ) r
GROUP BY hour
ORDER BY hour ASC;
```

Apache Sqoop

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

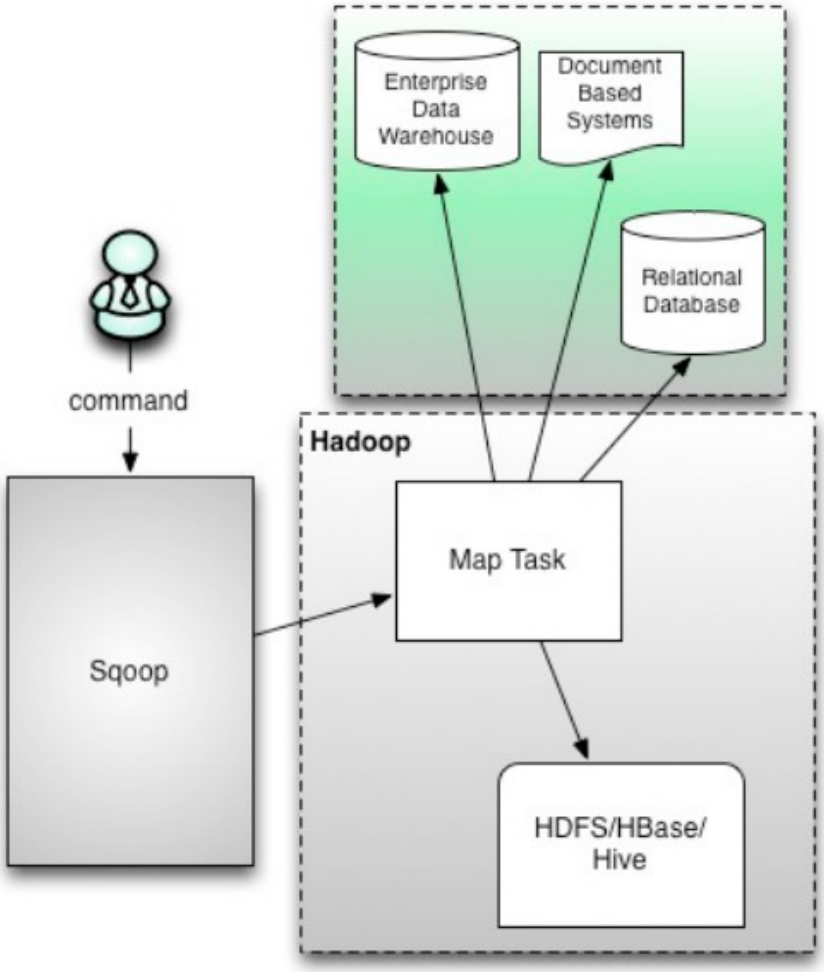
Introduction

Sqoop ("SQL-to-Hadoop") is a straightforward command-line tool with the following capabilities:

- **Imports individual tables or entire databases to files in HDFS**
- **Generates Java classes to allow you to interact with your imported data**
- **Provides the ability to import from SQL databases straight into your Hive data warehouse**

See also: http://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html

Architecture Overview



Hive.apache.org

Sqoop Benefit

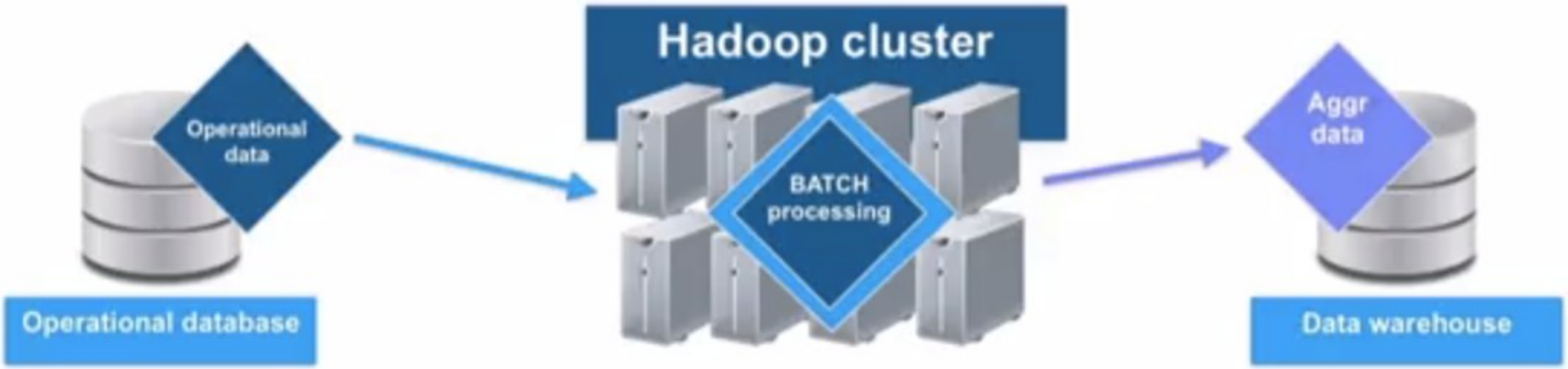
- Leverages RDBMS metadata to get the column data types
- It is simple to script and uses SQL
- It can be used to handle change data capture by importing daily transactional data to Hadoop
- It uses MapReduce for export and import that enables parallel and efficient data movement

Sqoop Mode

- Sqoop import: Data moves from RDBMS to Hadoop
- Sqoop export: Data moves from Hadoop to RDBMS

Use Case #1: ETL for Data Warehouse

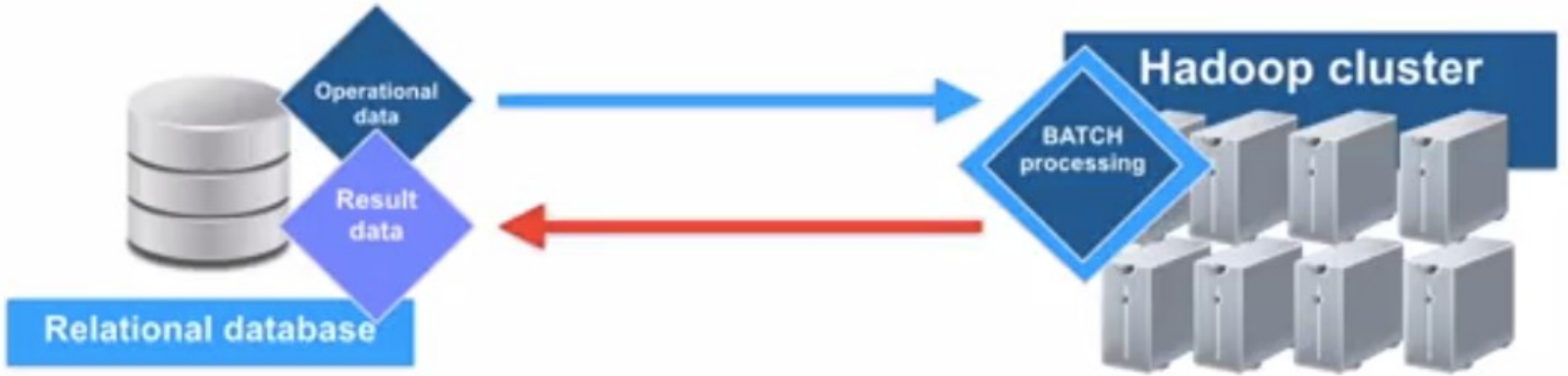
- Transform operational data for data warehouse reports in Hadoop as the batch transformation "engine"



Source: Mastering Apache Sqoop, David Yahalom, 2016

Use Case #2: ELT

- Extract operational data from RDBMS, process in Hadoop, return ***result*** to RDBMS



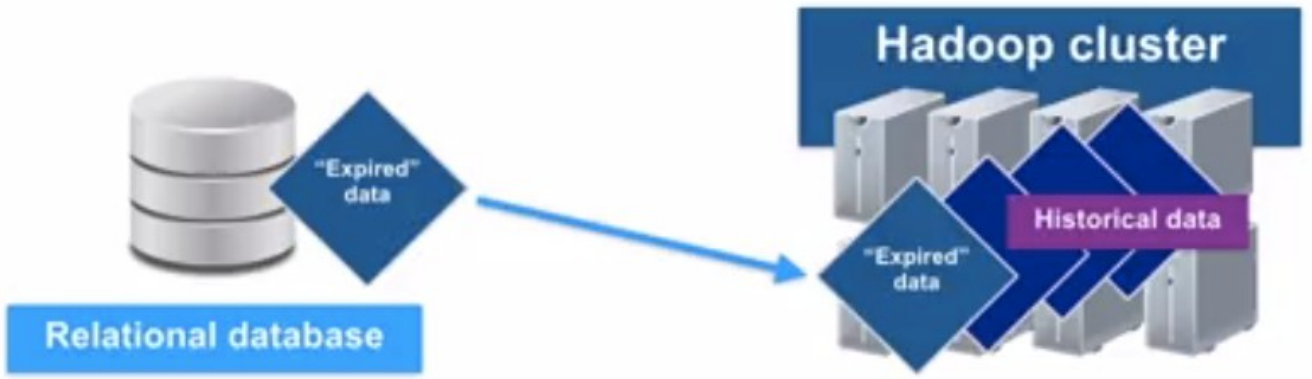
Use Case #3: Data Analysis

- Copy real-time data from RDBMS, combine with raw data on Hadoop using complex data analysis logic (not just SQL!)

Source: Mastering Apache Sqoop, David Yahalom, 2016

Use Case #4: Data Archival

- Move data from RDBMS after it expires to Hadoop, keeping the RDBMS "clean and lean"



Source: Mastering Apache Sqoop, David Yahalom, 2016

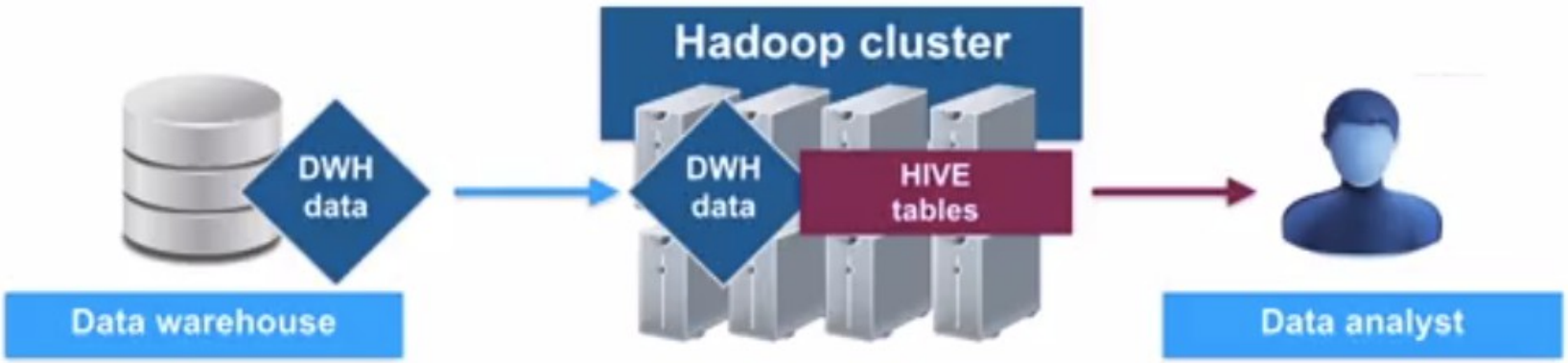
Use Case #5: Data Consolidation

- Integrate data from various organizational "data stores" to Hadoop for various data processing requirements

Source: Mastering Apache Sqoop, David Yahalom, 2016

Use Case #6: Move reports to Hadoop

- Easily allow traditional data analysis and business intelligence using Hadoop's power

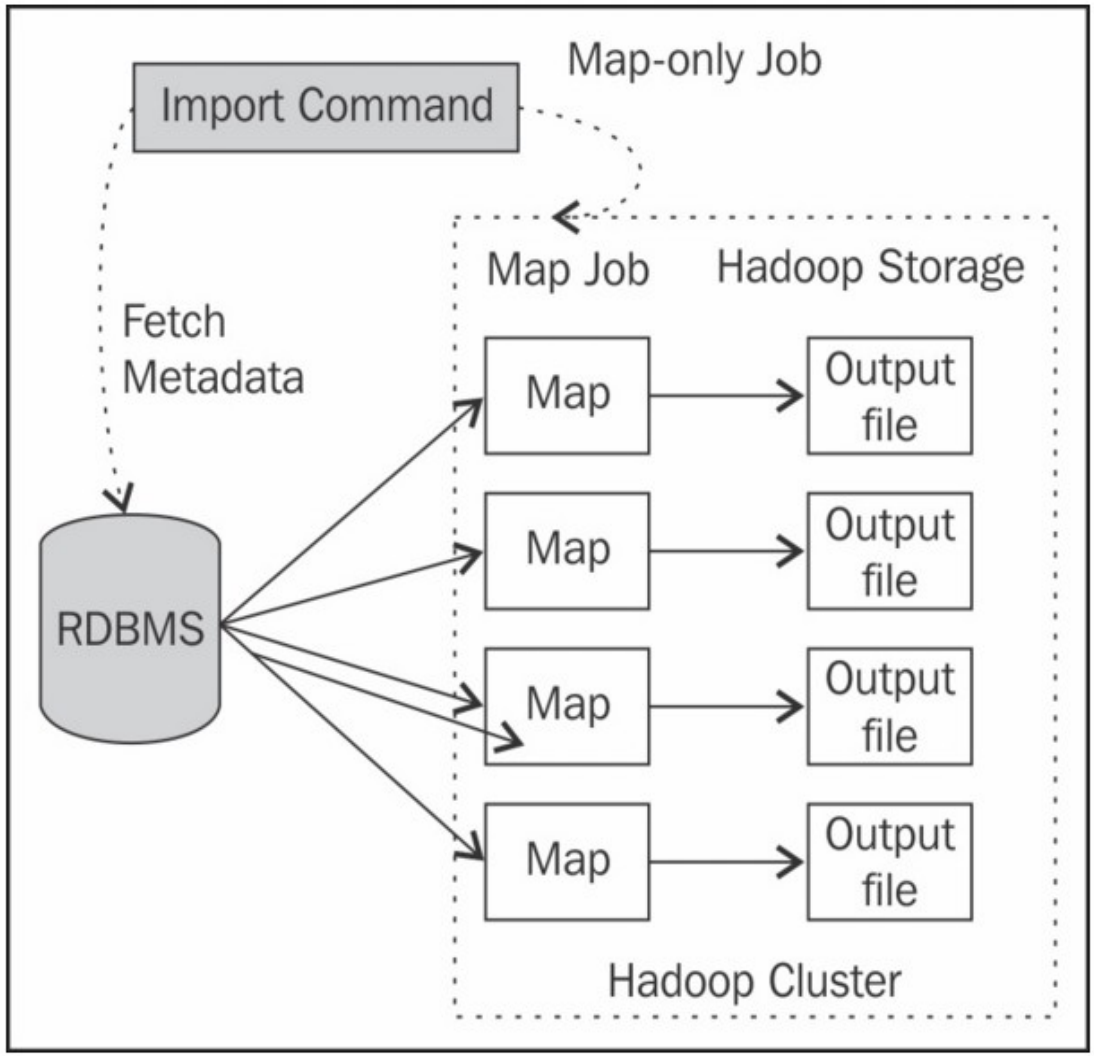


Source: Mastering Apache Sqoop, David Yahalom, 2016

Import Commands

Parameters	Description
`--connect <jdbc-uri>`	Specifies the server or database to connect to. It also specifies the port. For example: `--connect jdbc:mysql://host:port/databaseName`
`--connection-manager <class-name>`	Specifies the connection manager class name.
`--driver <class-name>`	Specifies the fully qualified name of the JDBC driver class.
`--hadoop-home <dir>`	This parameter is used to override the `$HADOOP_HOME` environment variable.
`-P`	If a user doesn't want to specify the database password along with the command, we can use the `-P` option to read the password from the console.
`--password <password>`	Sets the authentication password required to connect to the input source.
`--username <username>`	Sets the authentication username.
`--connection-param-file <properties-file>`	Specifies the connection parameter's file.
`--help`	This option will provide the usage instructions.
`--verbose`	Prints more information during a query execution.

Architecture of the import process



Incremental import

Parameter/argument	Description
`--check-column <column-name>`	The value of this column is used to determine the rows to be imported during the import process.
`--incremental <incremental-type>`	Specifies the type of incremental mode. Possible values are `append` and `lastmodified`.
`--last-value <value>`	Specifies the last value or the maximum value of the `check` column from the previous import. All the records whose `check` column value is greater than the value of the `–last-value` argument will be imported to HDFS.

```
bin/sqoop import -connect jdbc:mysql://localhost:3306/db1 -username root -password password -
table student -target-dir /user/abc/student -columns "student_id,address,name"  --incremental
lastmodified -last-value "2012-11-06 19:01:35"-check-column col4
```

Export Commands

Parameters	Description
`--direct`	Use the direct mode to perform the export quickly. Note that it is only supported for MySQL.
`--export-dir<dir>`	The location of input files in HDFS.
`--table <table-name>`	Name of the output table (the RDBMS table).
`-m,--num-mappers <n>`	Refers to the number of map tasks.
`--update-mode <mode>`	Specifies how updates are performed when new rows are found with non-matching keys in the database. Legal values for the mode include `updateonly` (default) and `allowinsert`.
`--update-key <col-name>`	The value of this column is used to identify the records that a user wants to update during the update mode. Use a comma-separated list of columns if there is more than one column.
`--staging-table <staging-table-name>`	Specifies the name of the staging table. The staging table is used to stage the data before inserting it into the destination table.
`--clear-staging-table`	This argument is used to clean the data from the staging table.

Hands-On: Loading Data from RDBMS to Hadoop

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Running MySQL Docker

- Command: sudo docker pull mysql
- Command: sudo docker run --name imcMysql -e MYSQL_ROOT_PASSWORD=imcinstitute -p 3306:3306 -d mysql
- Command: sudo docker exec -it imcMysql bash

```
root@f1922a70e09c:/# mysql -uroot -p"imcinstitute"
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Prepare a test database table

```
mysql> CREATE DATABASE imc_db;
mysql> USE imc_db;
mysql> CREATE TABLE country_tbl(id INT NOT NULL, country
VARCHAR(50), PRIMARY KEY  (id));


mysql> INSERT INTO country_tbl VALUES(1, 'USA');
mysql> INSERT INTO country_tbl VALUES(2, 'CANADA');
mysql> INSERT INTO country_tbl VALUES(3, 'Mexico');
mysql> INSERT INTO country_tbl VALUES(4, 'Brazil');
mysql> INSERT INTO country_tbl VALUES(61, 'Japan');
mysql> INSERT INTO country_tbl VALUES(65, 'Singapore');
mysql> INSERT INTO country_tbl VALUES(66, 'Thailand');
```

View data in the table

```
mysql> SELECT * FROM country_tbl;
```

```
+----+-----------+
| id | country   |
+----+-----------+
|  1 | USA       |
|  2 | CANADA    |
|  3 | Mexico    |
|  4 | Brazil    |
| 61 | Japan     |
| 65 | Singapore |
| 66 | Thailand  |
+----+-----------+
7 rows in set (0.00 sec)
```

```
mysql> exit;
```

Then exit from the container by press Ctrl-P & Ctrl-Q

Restart the Cloudera docker with linking to the MySQL Docker

Command: sudo run --hostname=quickstart.cloudera --privileged=true --link imcMysql:mysqldb -t -i -p 8888:8888 cloudera/quickstart /usr/bin/docker-quickstart

If both of these Dockers are up and running, you can find out the internal IP address of each of them by running this command. This gets the IP for imcMysql.

- Command: sudo docker inspect imcMysql | grep IPAddress

```
"IPAddress": "172.17.0.7",
```

Restart the Cloudera docker with linking to the MySQL Docker

Command: sudo docker run --hostname=quickstart.cloudera --privileged=true --link imcMysql:mysqldb -t -i -p 8888:8888   cloudera/quickstart /usr/bin/docker-quickstart

If both of these Dockers are up and running, you can find out the internal IP address of each of them by running this command. This gets the IP for imcMysql.

- Command: sudo docker inspect imcMysql | grep IPAddress

```
"IPAddress": "172.17.0.7",
```

Check a MySQL driver for Sqoop

```
$ cd /var/lib/sqoop
$ ls
```

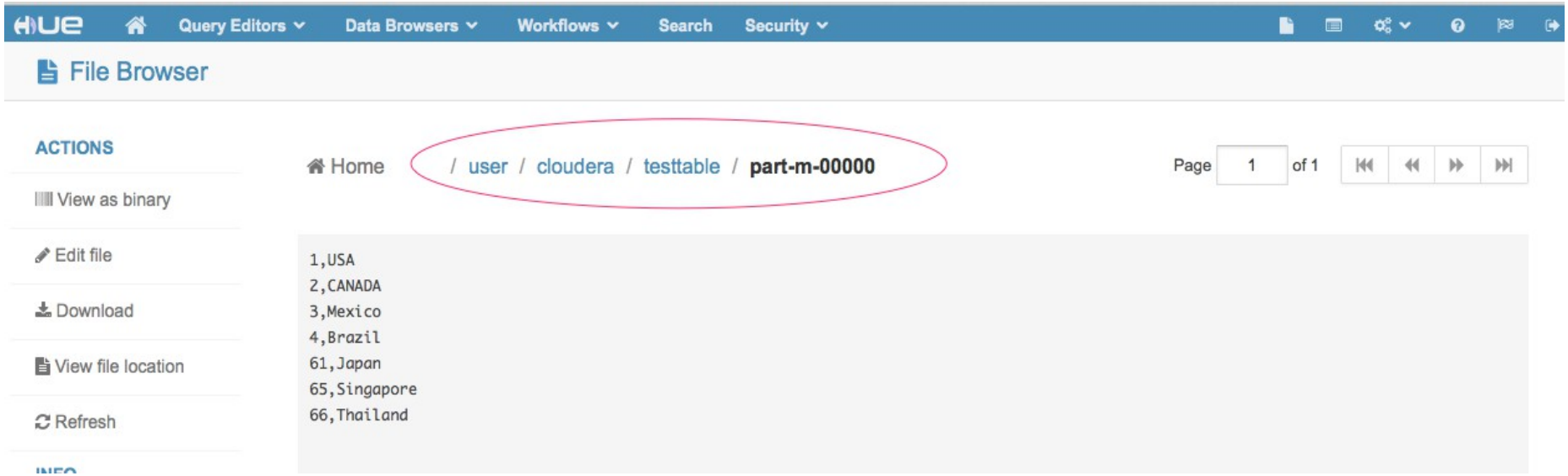
```
[root@quickstart sqoop]# ls
mysql-connector-java.jar
```

Note: If you do not see the driver file, you need to install one by using the following command

```
$ wget https://s3.amazonaws.com/imcbucket/apps/mysql-connector-java-5.1.23-bin.jar
```

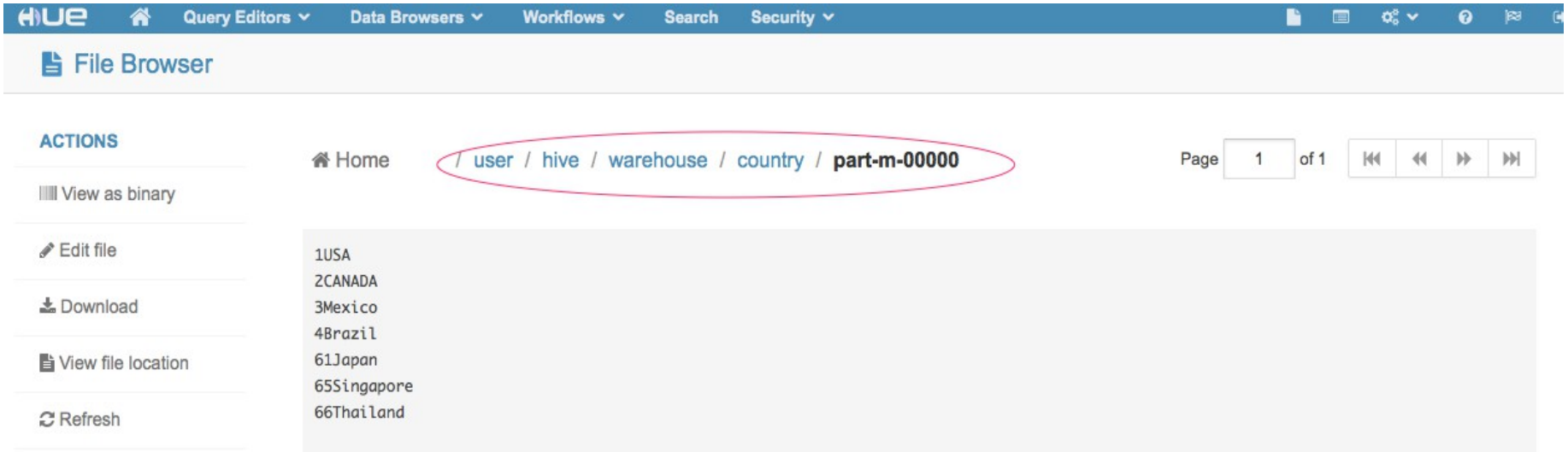
Importing data from MySQL to HDFS

```
$sqoop import --connect jdbc:mysql://172.17.0.7/imc_db
--username root --password imcinstitute --table country_tbl
--target-dir /user/cloudera/testtable -m 1
```



Importing data from MySQL to Hive Table

```
$sqoop import --connect jdbc:mysql://172.17.0.7/imc_db
--username root --password imcinstitute --table country_tbl
--hive-import --hive-table country -m 1
```

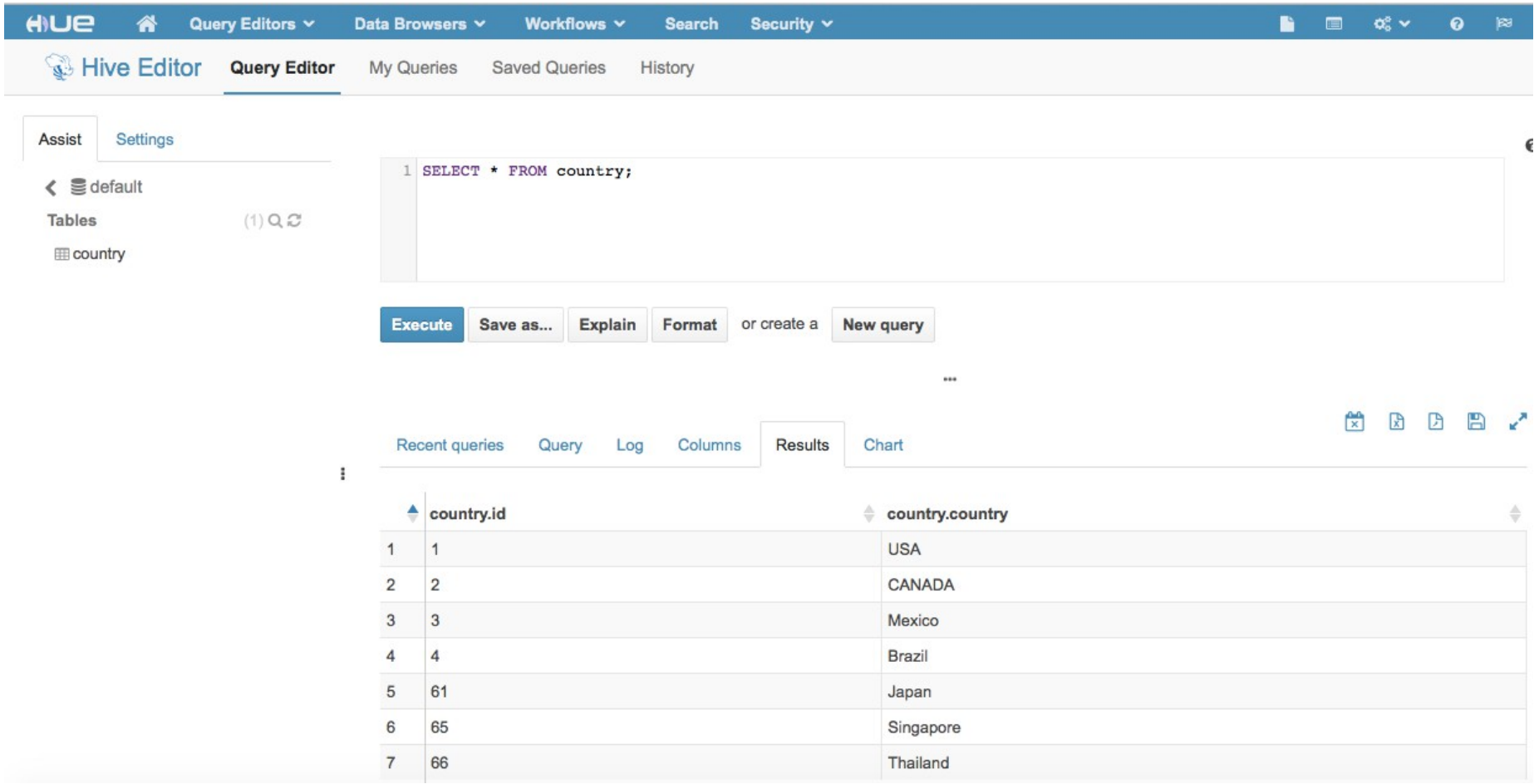


Reviewing data from Hive Table

```
[root@quickstart /]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.

hive> show tables;
hive> select * from country;
```

```
1       USA
2       CANADA
3       Mexico
4       Brazil
61      Japan
65      Singapore
66      Thailand
Time taken: 0.587 seconds, Fetched: 7 row(s)
```

Running from Hue: Beewax



	country.id	country.country
1	1	USA
2	2	CANADA
3	3	Mexico
4	4	Brazil
5	61	Japan
6	65	Singapore
7	66	Thailand

Importing data from MySQL to HBase

```
$sqoop import --connect jdbc:mysql://172.17.0.7/imc_db
--username root --password imcinstitute --table country_tbl
--hbase-table country --column-family hbase_country_cf --hbase-row-key
id --hbase-create-table  -m 1
```

Start HBase

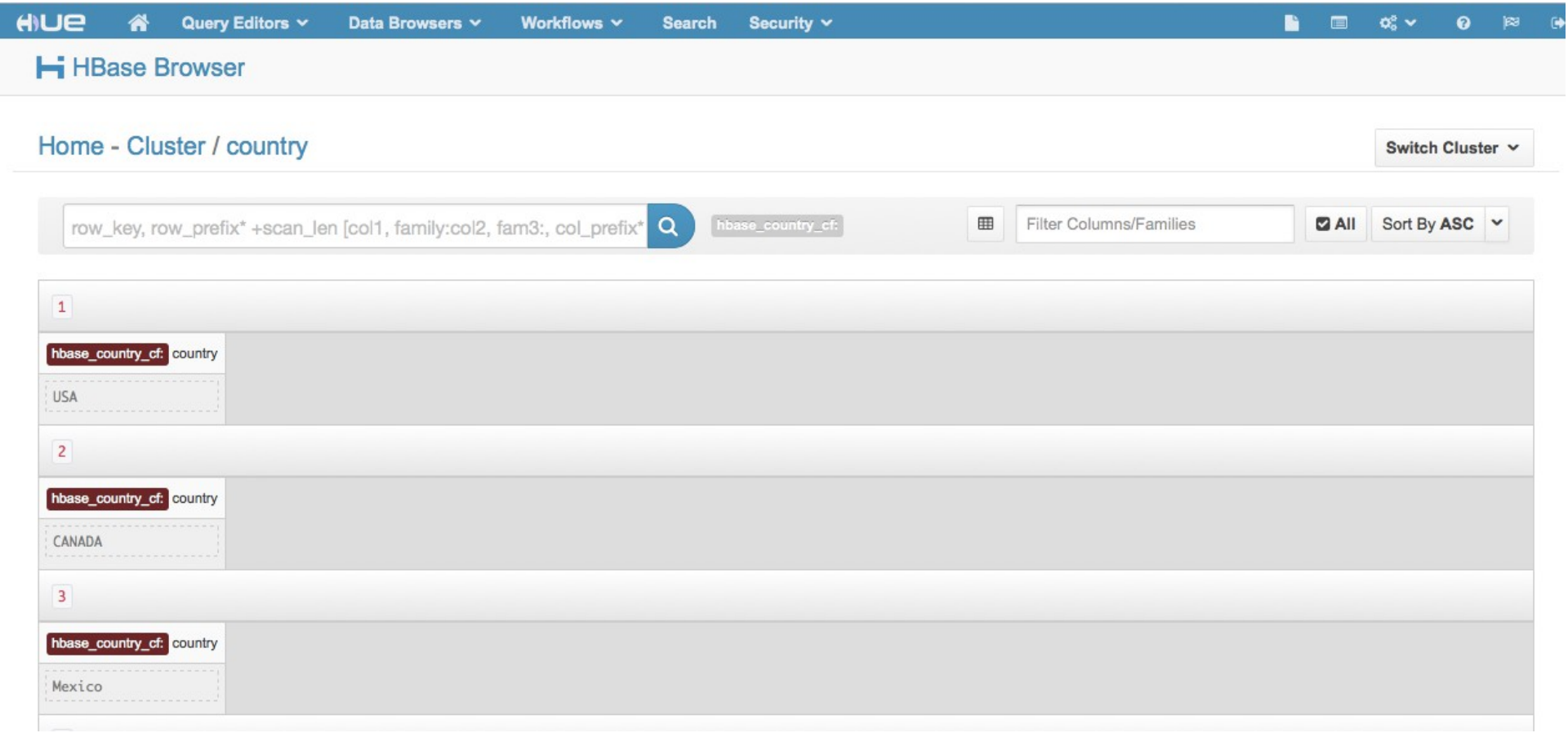
```
$hbase shell
hbase(main):001:0> list
```

```
TABLE
country
1 row(s) in 0.2570 seconds

=> ["country"]
```

Viewing Hbase data

```
hbase(main):003:0> scan 'country'
ROW                     COLUMN+CELL
 1                      column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=USA
 2                      column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=CANADA
 3                      column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=Mexico
 4                      column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=Brazil
 61                     column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=Japan
 65                     column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=Singapore
 66                     column=hbase_country_cf:country, timestamp=1468081466623, val
                        ue=Thailand
7 row(s) in 0.1670 seconds
```

Viewing data from Hbase browser



IMC Institute

Apache Flume

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

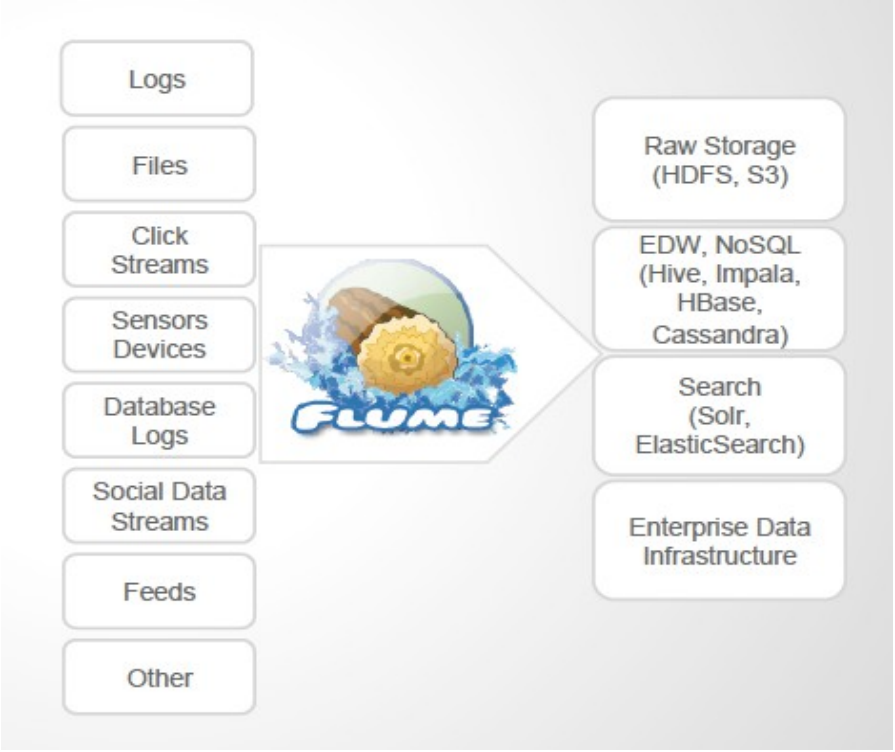
Introduction

Apache Flume is:

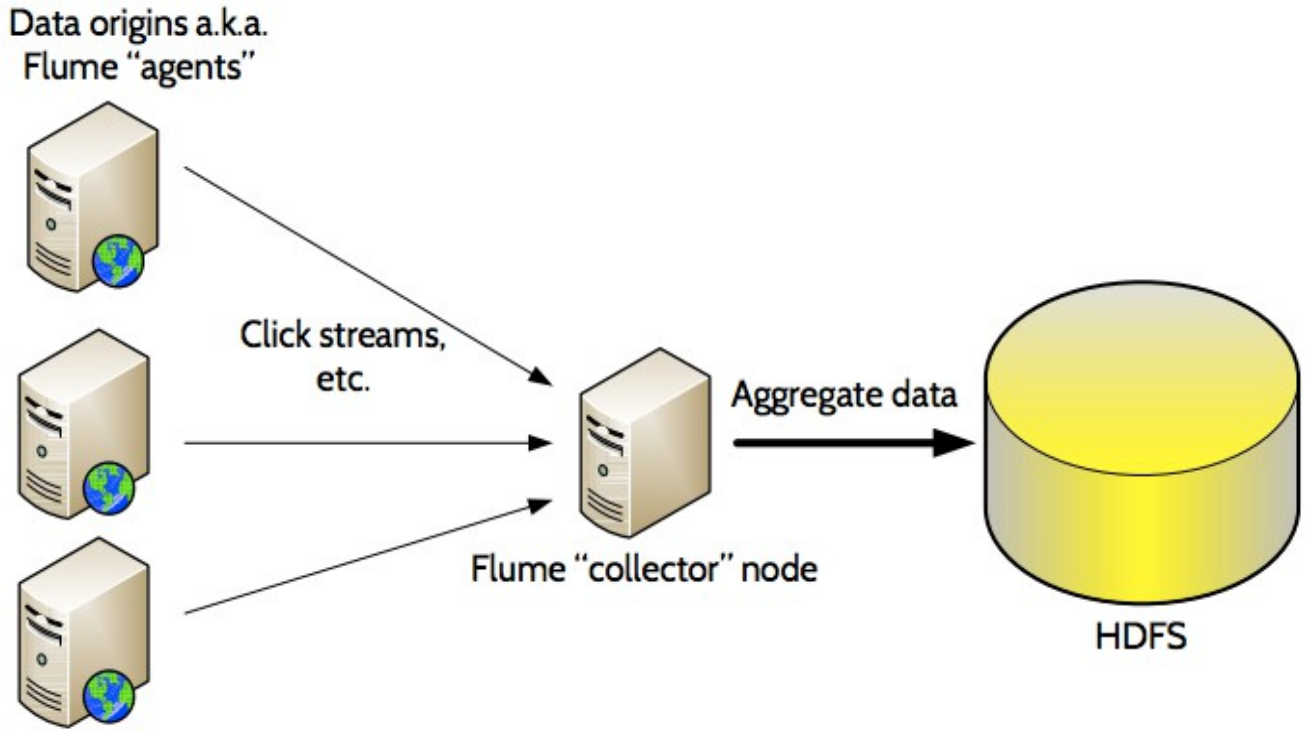
- **A distributed data transport and aggregation system for event- or log-structured data**
- **Principally designed for continuous data ingestion into Hadoop… But more flexible than that**

What is Flume?

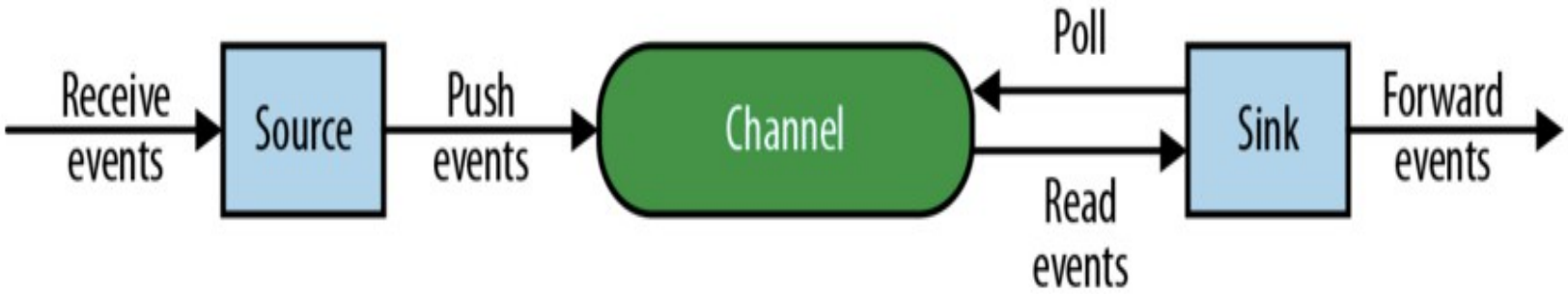
- Apache Flume is a continuous data ingestion system that is...
  - open-source,
  - reliable,
  - scalable,
  - manageable,
  - Customizable,
  - and designed for Big Data ecosystem



Architecture Overview



Flume Agent



- A source writes events to one or more channels.
- A channel is the holding area as events are passed from a source to a sink.
- A sink receives events from one channel only.
- An agent can have many channels.

Source: Using Flume, Hari Shreedharan, 2014

Sources

- Different Source types:
- Require at least one channel to function
- Specialized sources for integrating with well-known systems.
  - Example: Spooling Files, Syslog, Netcat, JMS
  - Auto-Generating Sources: Exec, SEQ
  - IPC sources for Agent-to-Agent communication: Avro, Thrift

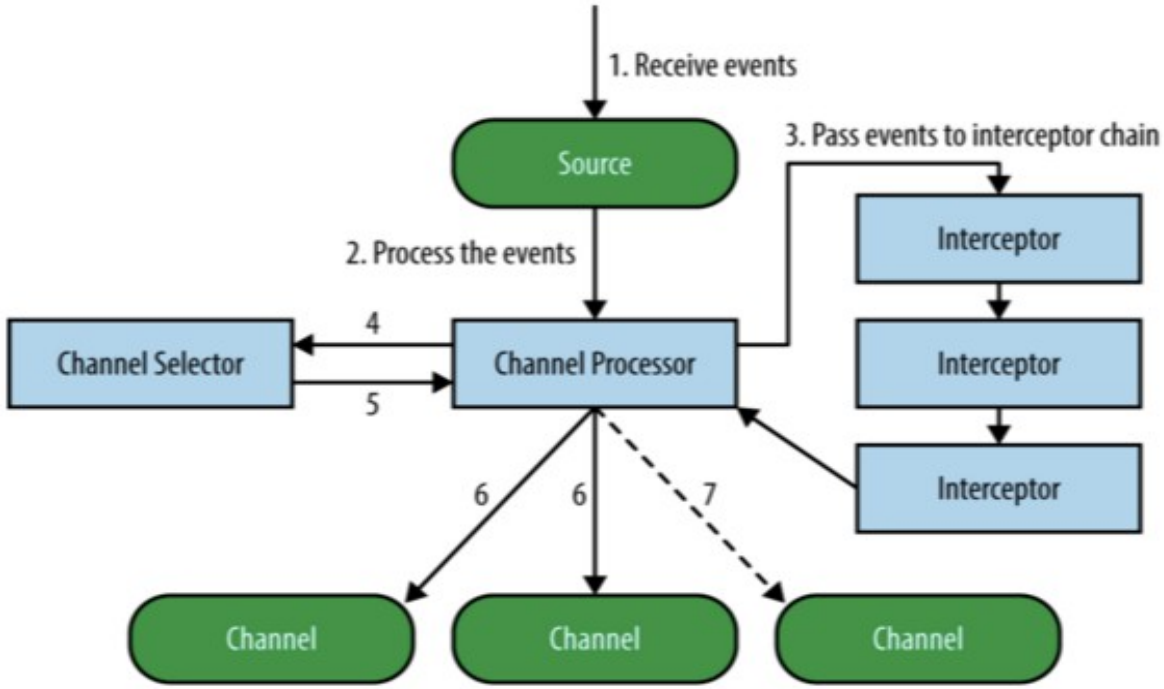
Channel

- Different Channels offer different levels of persistence:
  - Memory Channel
  - File Channel:
- Eventually, when the agent comes back data can be accessed.
- Channels are fully transactional
- Provide weak ordering guarantees
- Can work with any number of Sources and Sinks

Sink

- Different types of Sinks:
  - Terminal sinks that deposit events to their final destination. For example: HDFS, HBase, Morphline-Solr, Elastic Search
  - Sinks support serialization to user's preferred formats.
  - HDFS sink supports time-based and arbitrary bucketing of data while writing to HDFS.
  - IPC sink for Agent-to-Agent communication: Avro, Thrift
- Require exactly one channel to function

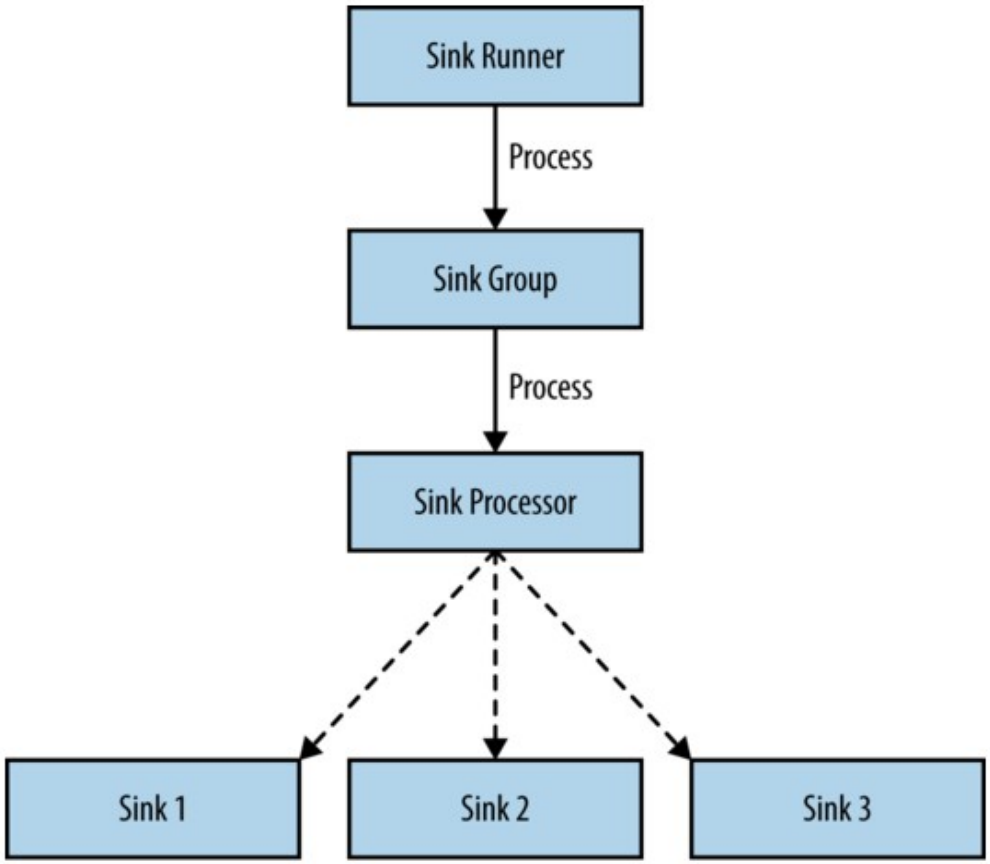
Flume Process



4. Pass each event to channel selector
5. Return list of channels the event is to be written to
6. Write all events that need to go to each required channel. Only one transaction is opened. with each channel, and all events to a channel are written as part of that transaction.
7. Repeat the same with optional channels

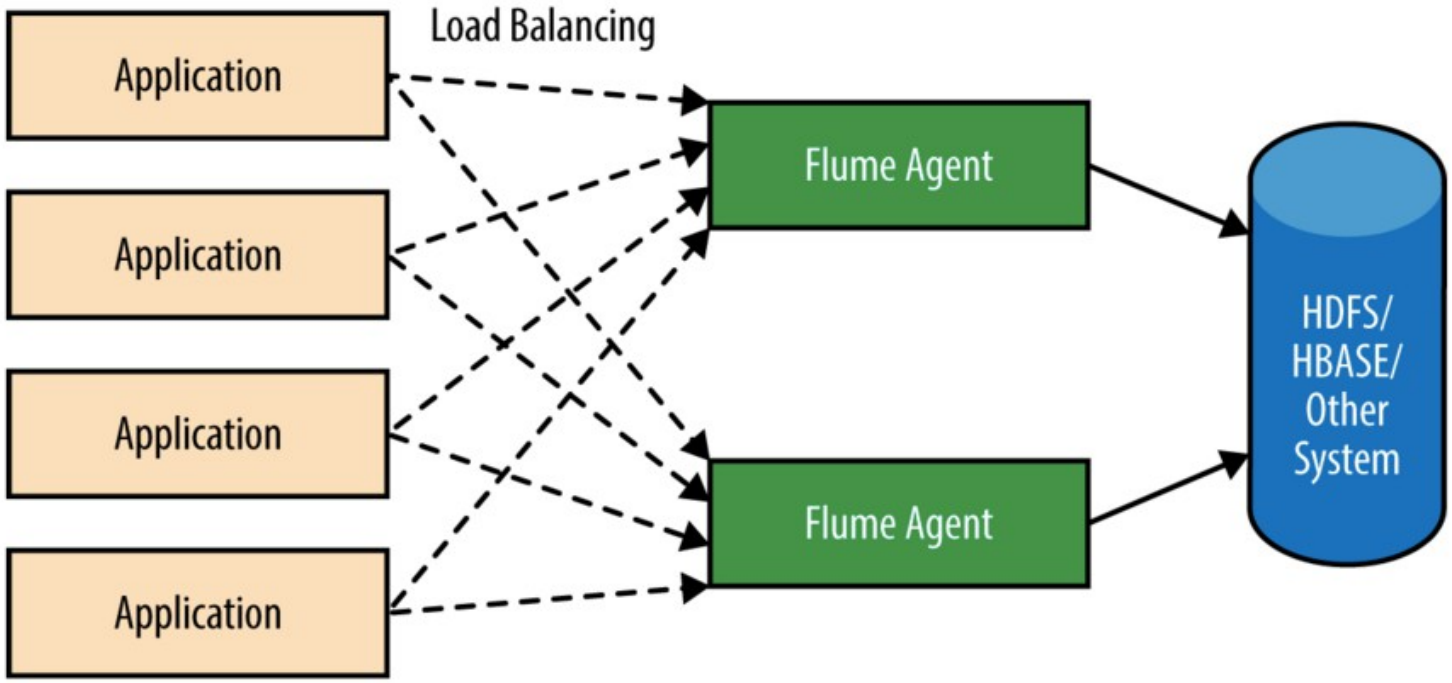
Source: Using Flume, Hari Shreedharan, 2014

Flume Process



Source: Using Flume, Hari Shreedharan, 2014

Flow



A Simple Flow

Source: Using Flume, Hari Shreedharan, 2014

Flume terminology

- A source writes events to one or more channels.
- A channel is the holding area as events are passed from a source to a sink.
- A sink receives events from one channel only.
- An agent can have many channels.

Odiago

Flume Agent Configuration : Example

```
agent.sources = httpSrc
agent.channels = memory1 memory2
agent.sinks = hdfsSink hbaseSink

agent.sources.httpSrc.type = http
agent.sources.httpSrc.channels = memory1 memory2

# Bind to all interfaces
agent.sources.httpSrc.bind = 0.0.0.0
agent.sources.httpSrc.port = 4353

# Removing this line will disable SSL
agent.sources.httpSrc.ssl = true
agent.sources.httpSrc.keystore = /tmp/keystore
agent.sources.httpSrc.keystore-password = UsingFlume

agent.sources.httpSrc.handler = usingflume.ch03.HTTPSourceXMLHandler
agent.sources.httpSrc.handler.insertTimestamp = true

agent.sources.httpSrc.interceptors = hostInterceptor
agent.sources.httpSrc.interceptors.hostInterceptor.type = host
```

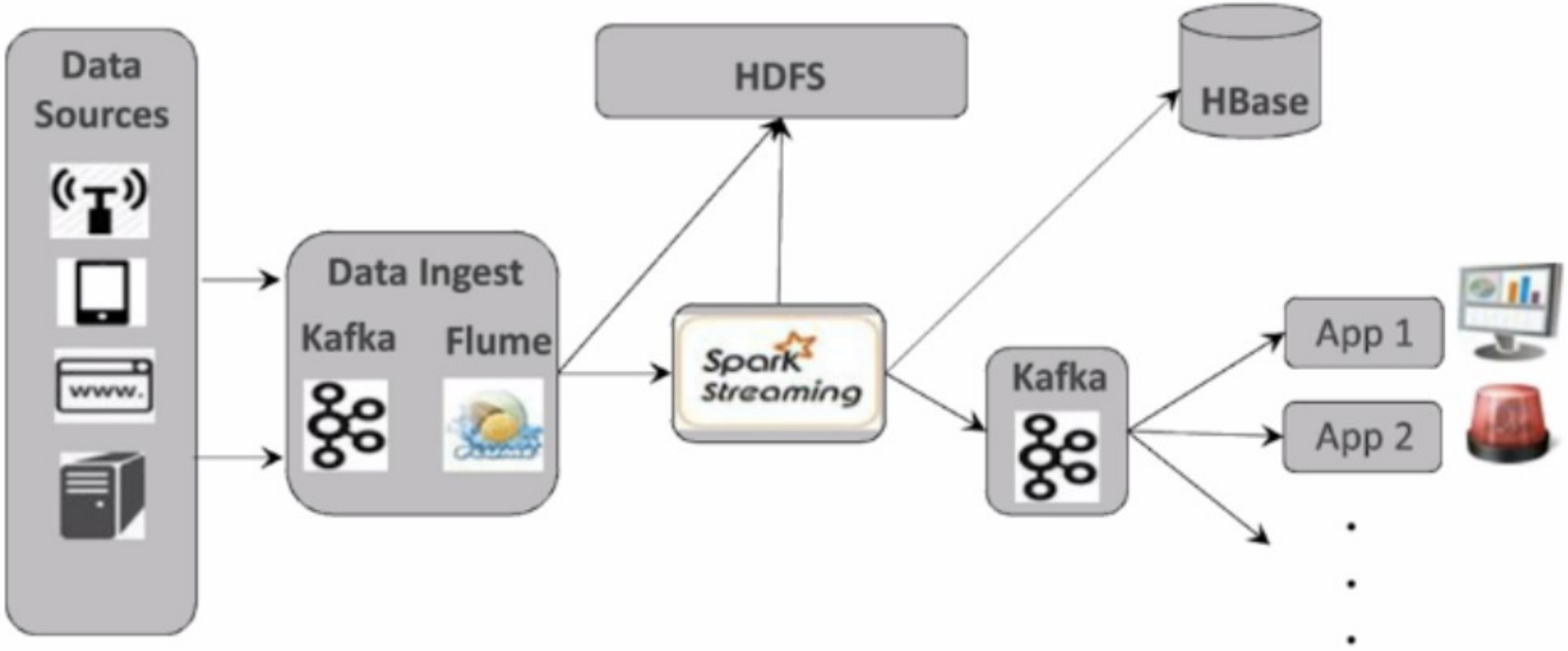
Odiago

Flume Agent Configuration : Example

```
# Initializes a memory channel with default configuration
agent.channels.memory1.type = memory

# Initializes a memory channel with default configuration
agent.channels.memory2.type = memory

# HDFS Sink
agent.sinks.hdfsSink.type = hdfs
agent.sinks.hdfsSink.channel = memory1
agent.sinks.hdfsSink.hdfs.path = /Data/UsingFlume/%{topic}/%Y/%m/%d/%H/%M
agent.sinks.hdfsSink.hdfs.filePrefix = UsingFlumeData

agent.sinks.hbaseSink.type = asynchbase
agent.sinks.hbaseSink.channel = memory2
agent.sinks.hbaseSink.serializer = usingflume.ch05.AsyncHBaseDirectSerializer
agent.sinks.hbaseSink.table = usingFlumeTable
```

Odiago

Stream Processing Architecture

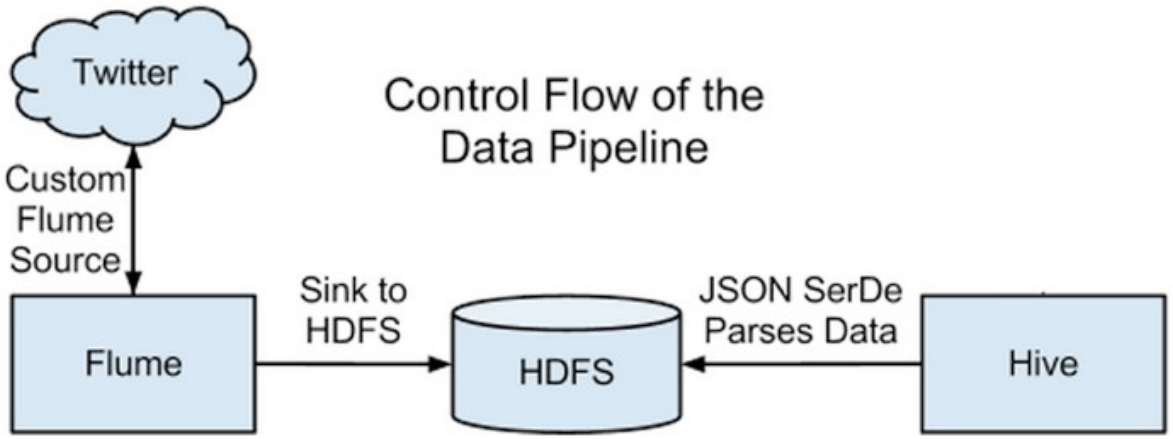


Odiago

Hands-On: Loading Twitter Data to Hadoop HDFS

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Exercise Overview



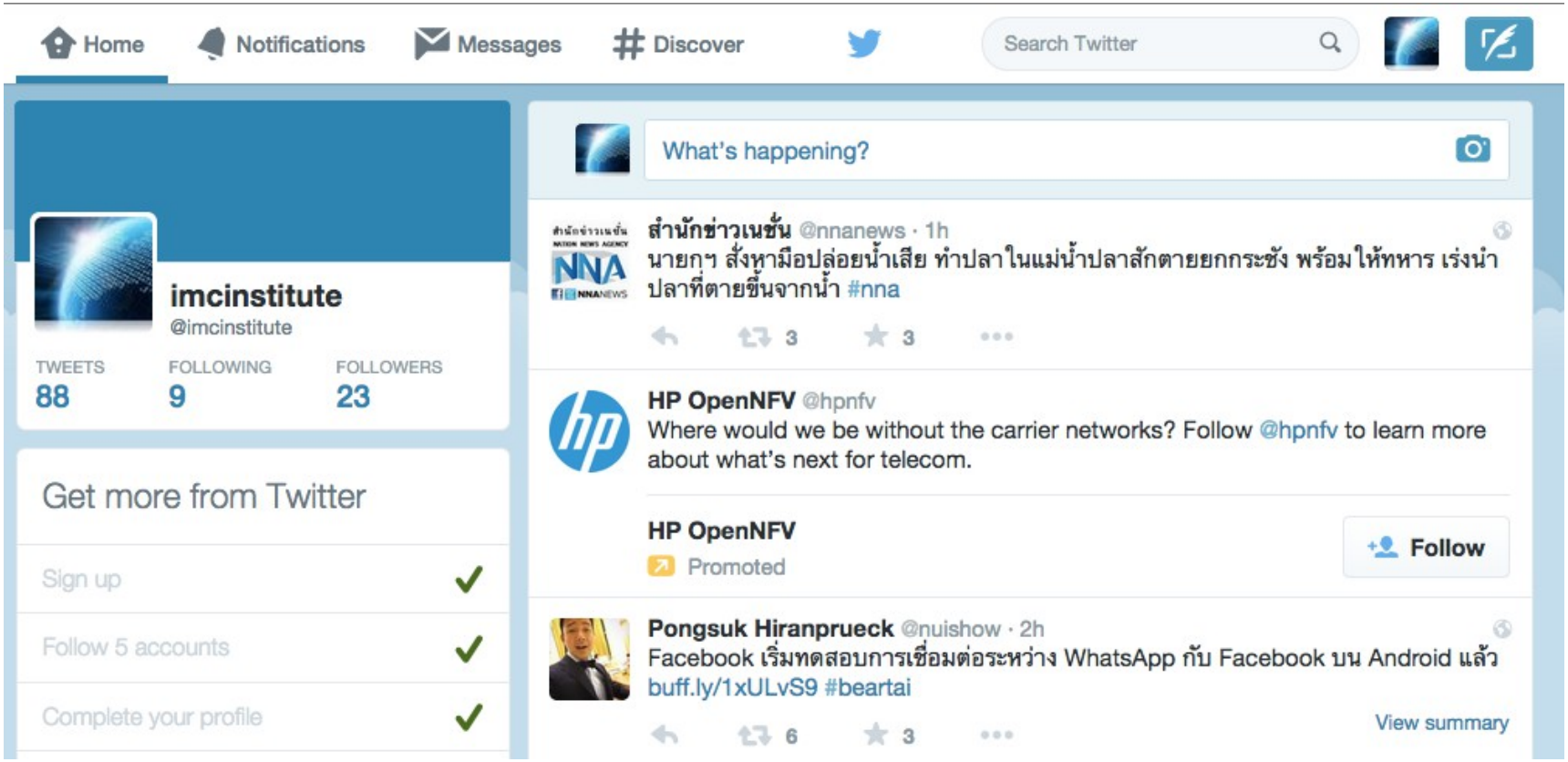
Hive.apache.org

Installing Pre-built version of flume

```
$ wget
http://files.cloudera.com/samples/flume-sources-1.0-SNAPSHOT.jar

$ sudo cp flume-sources-1.0-SNAPSHOT.jar
/opt/cloudera/parcels/CDH-5.5.1-1.cdh5.5.1.p0.11/lib/flume-
ng/lib/

$sudo cp /etc/flume-ng/conf/flume-env.sh.template
/etc/flume-ng/conf/flume-env.sh
```

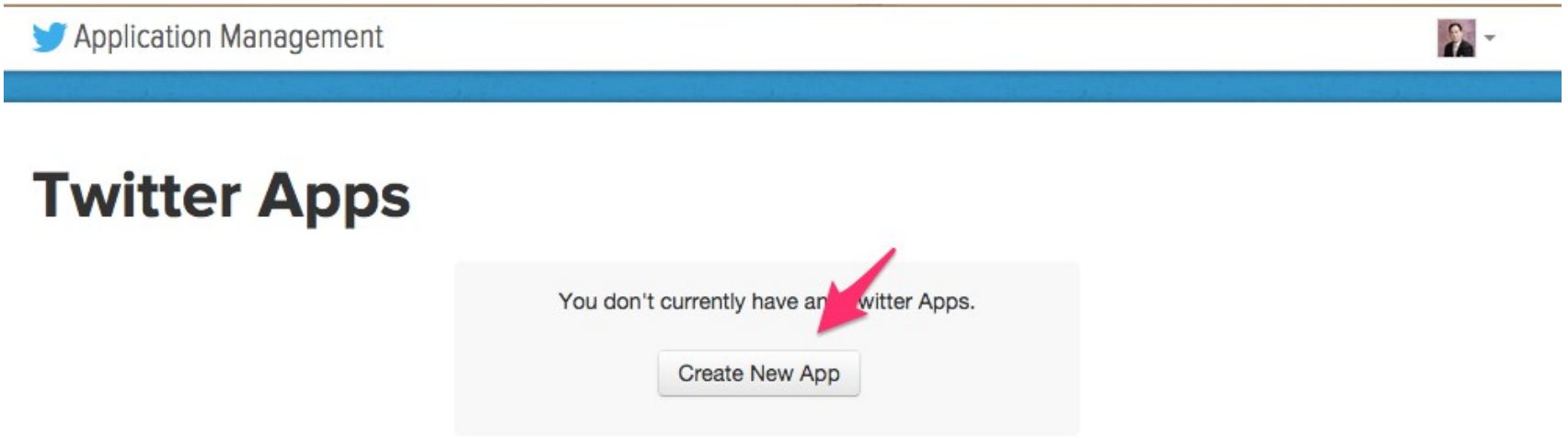
Create a new Twitter App

Login to your Twitter @ twitter.com



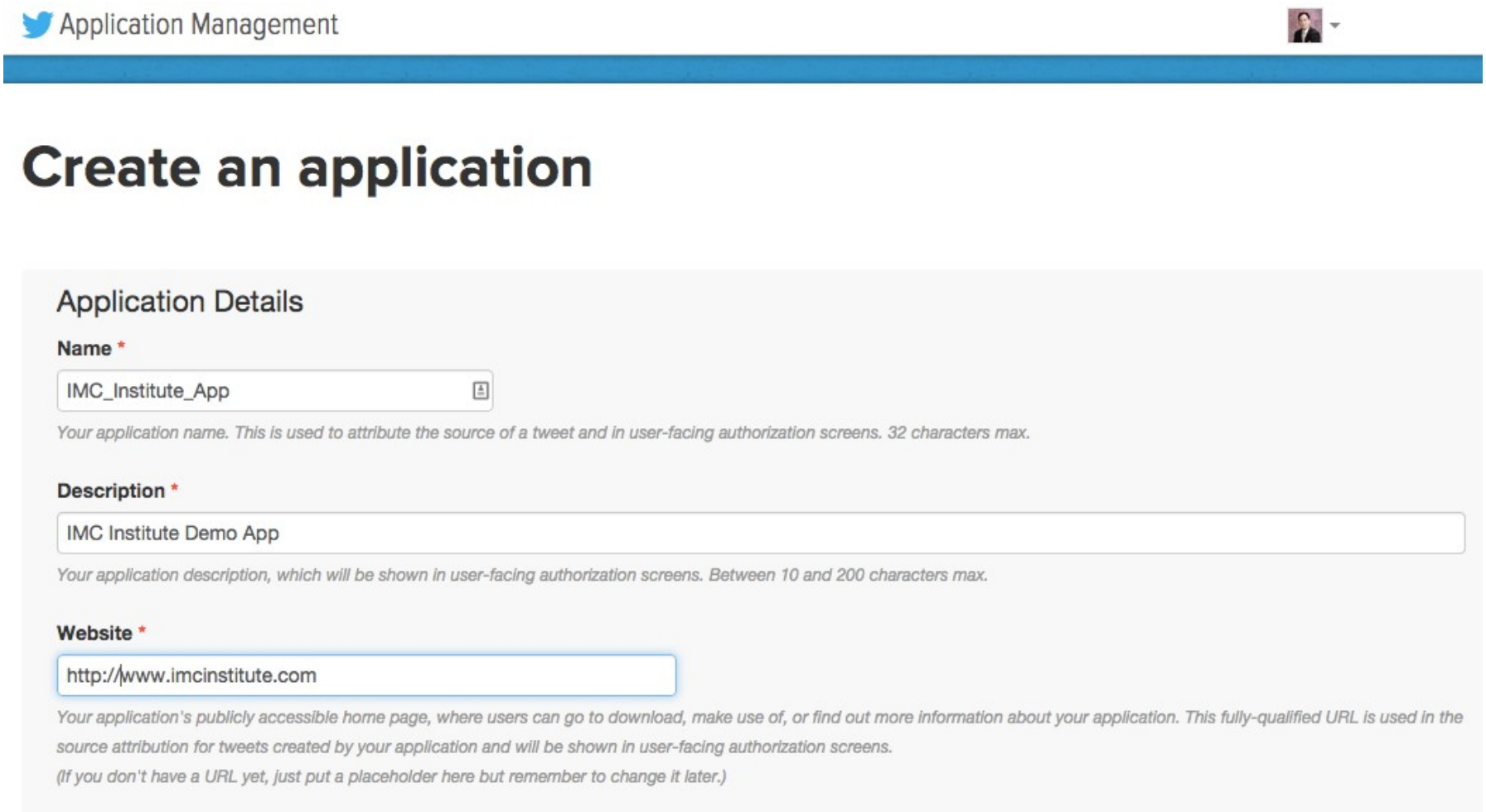
Create a new Twitter App (cont.)

Create a new Twitter App @ apps.twitter.com



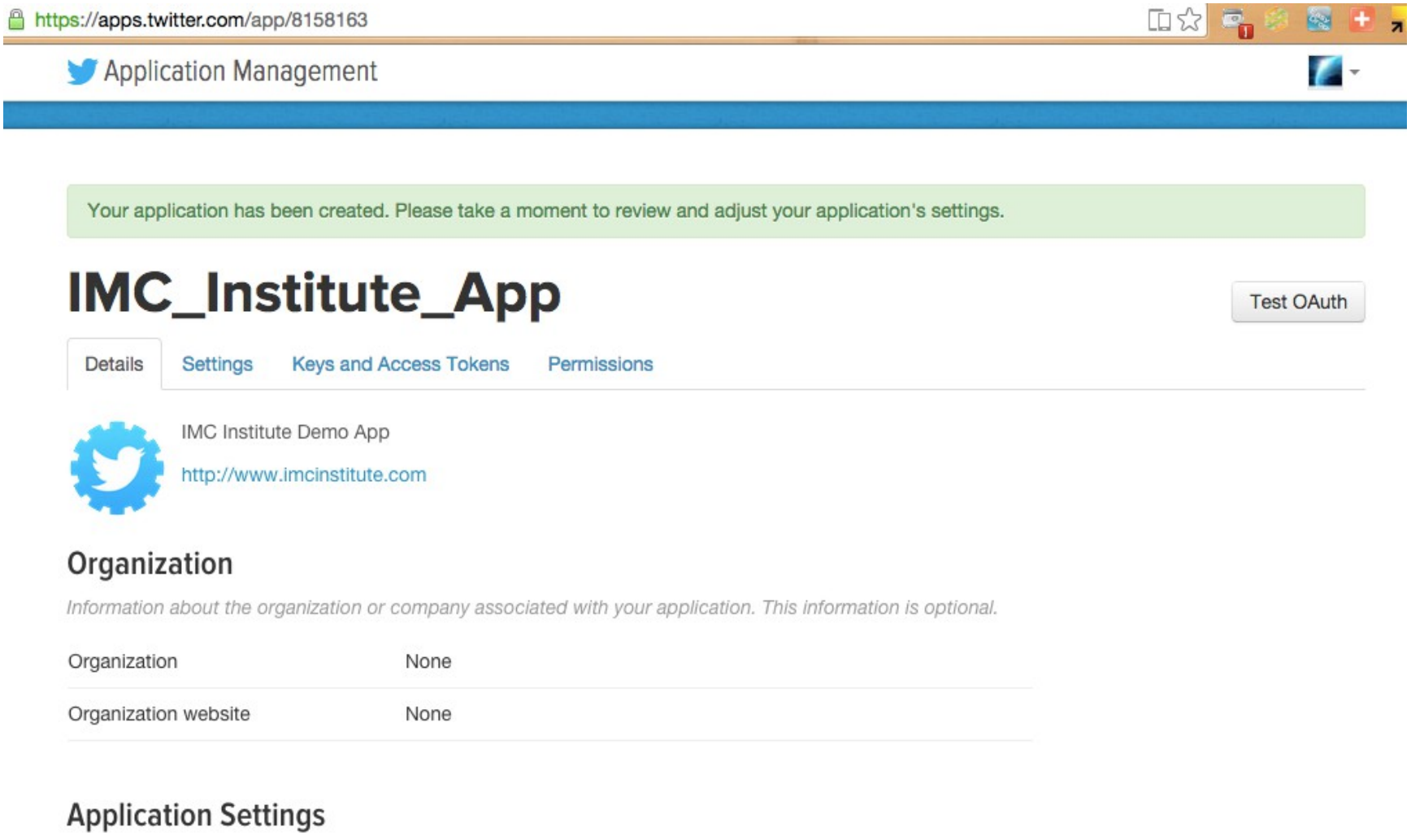
Create a new Twitter App (cont.)

Enter all the details in the application:



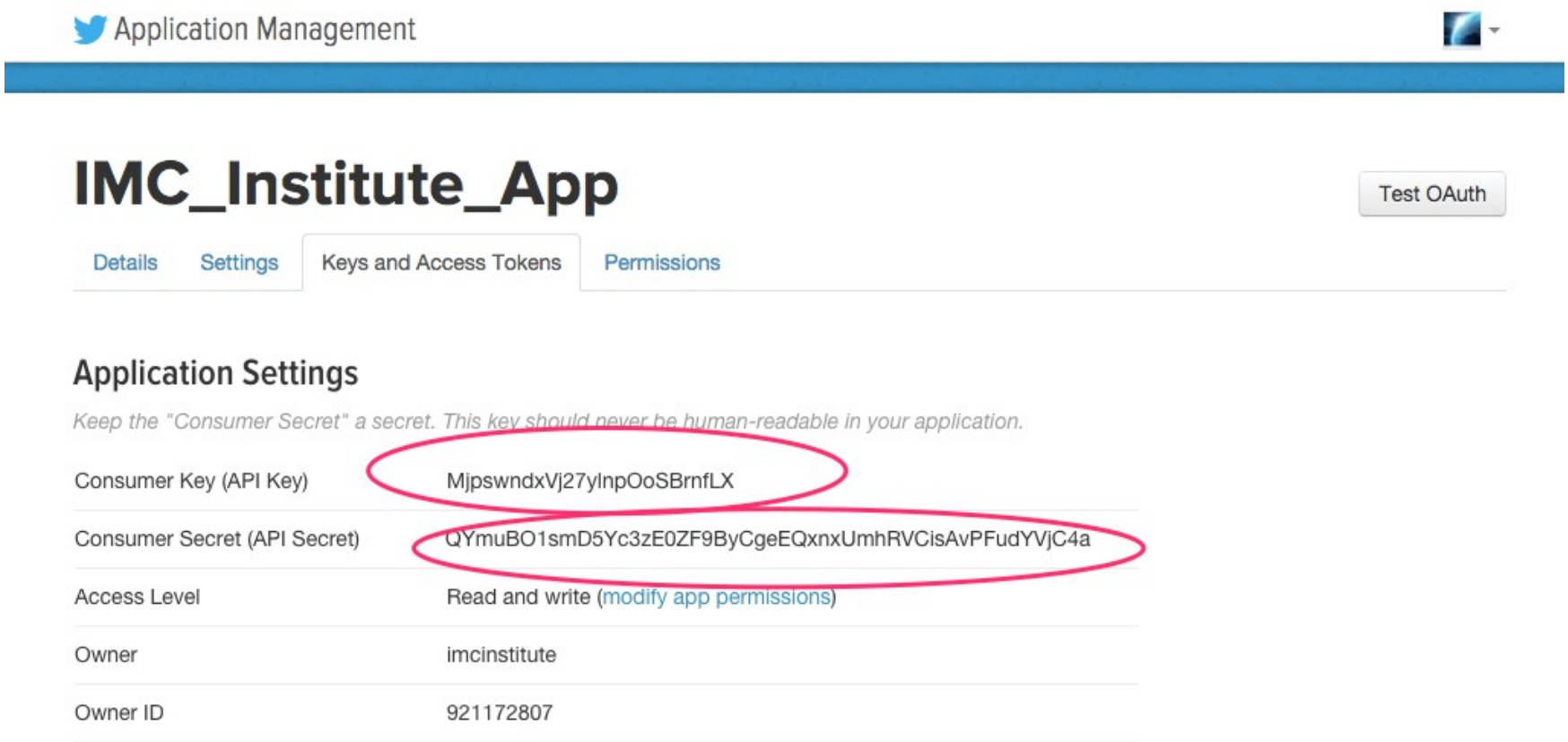
Create a new Twitter App (cont.)

Your application will be created:



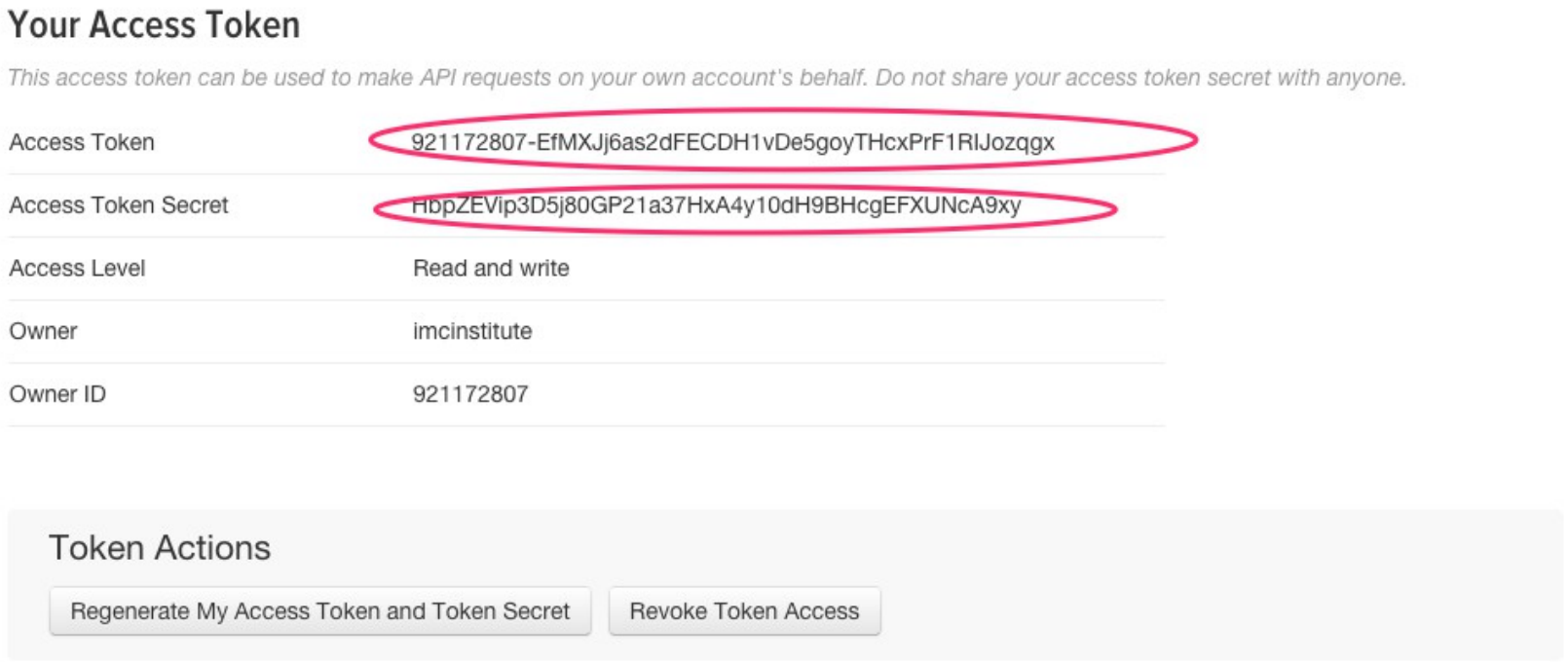
Create a new Twitter App (cont.)

Click on Keys and Access Tokens:



Create a new Twitter App (cont.)

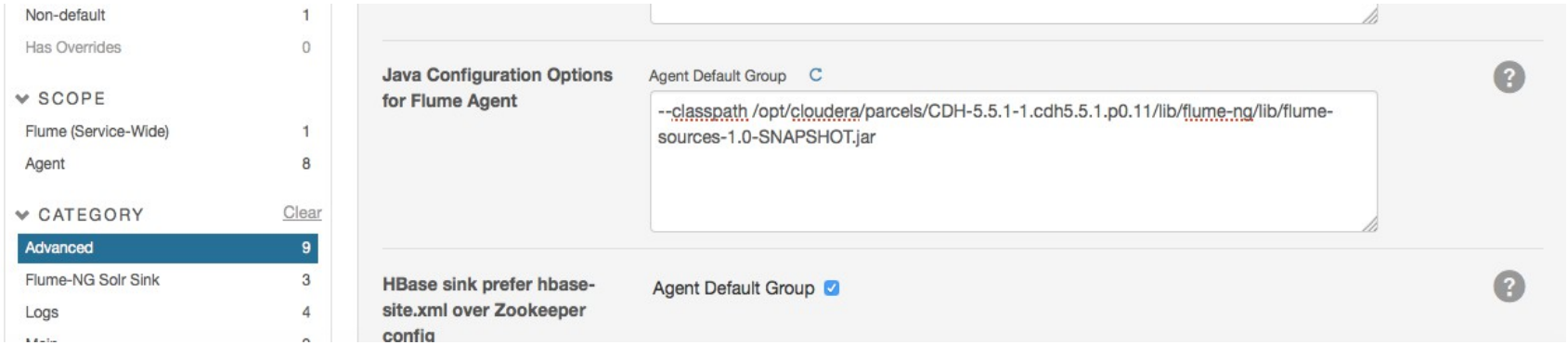
Your Access token got created:



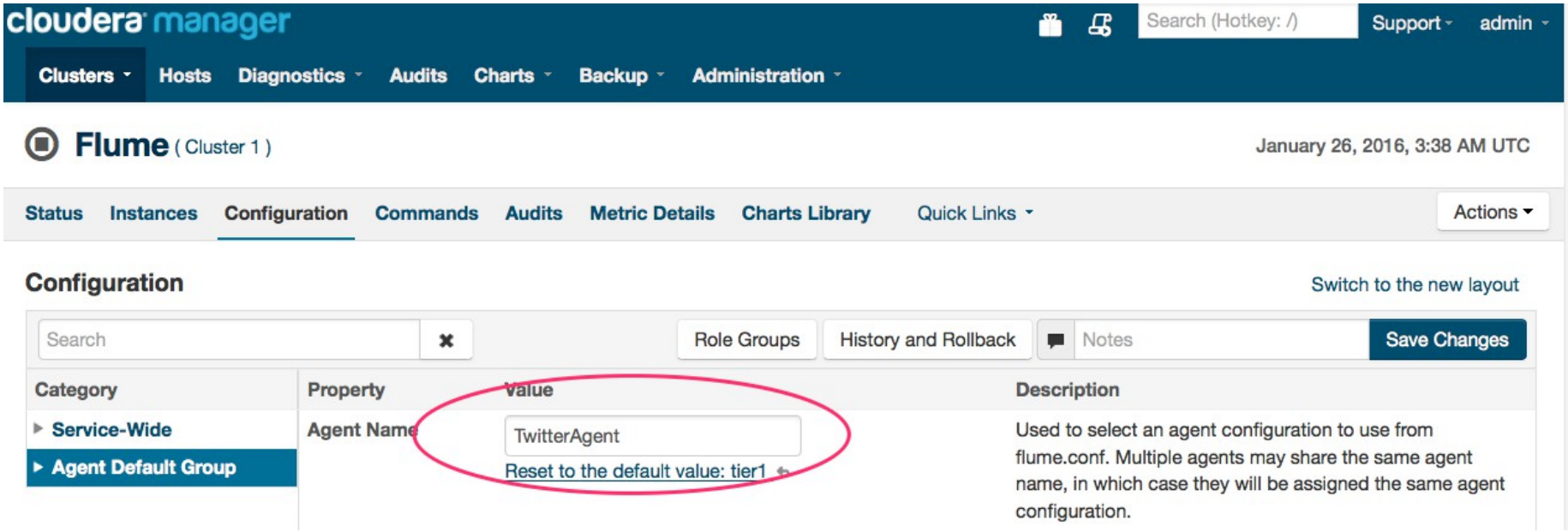
Add classpath in Cloudera Manager

"Services" -> "flume1" -> "Configuration" -> -> "Advanced" -> "Java Configuration Options for Flume Agent", add:

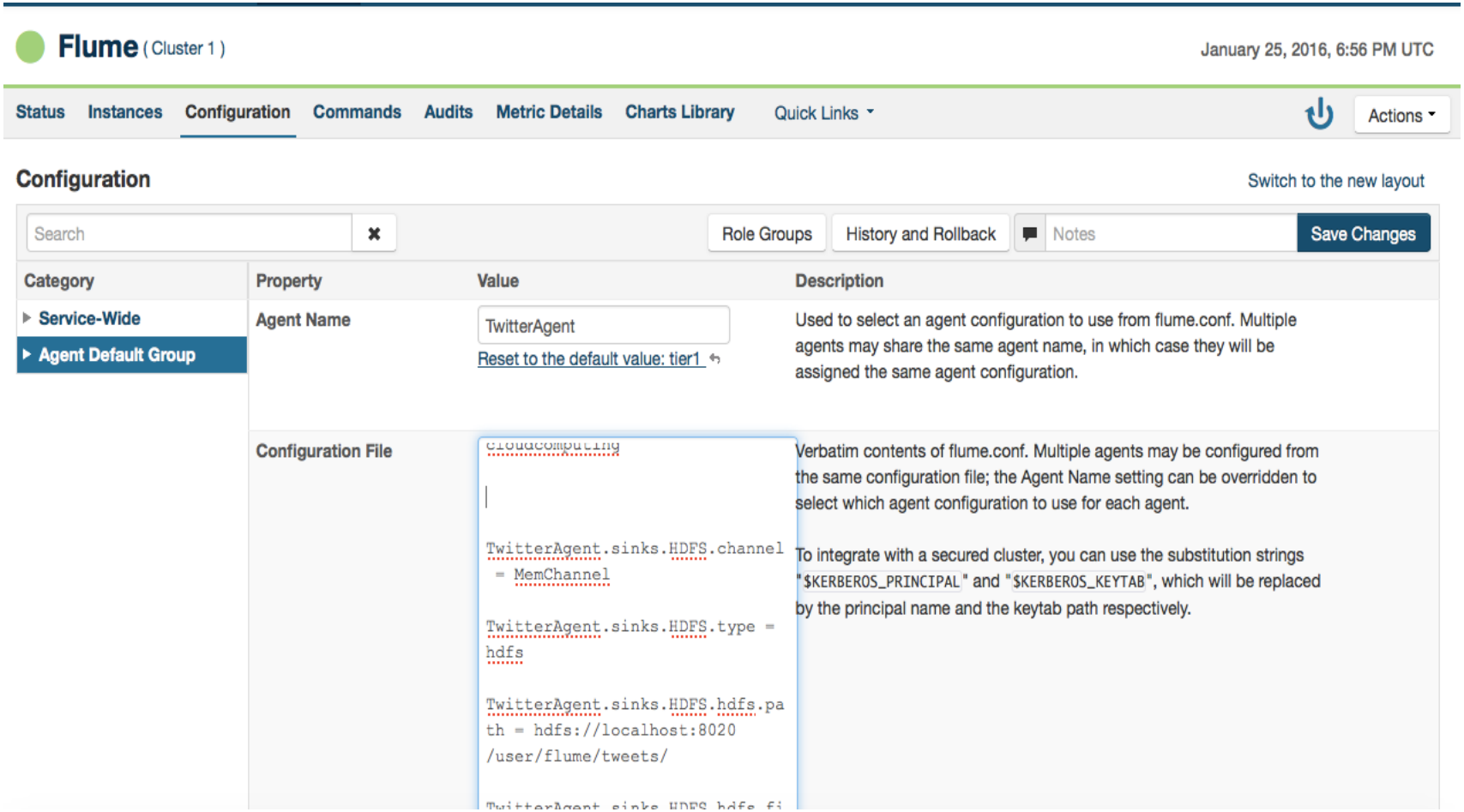
```
--classpath /opt/cloudera/parcels/CDH-5.5.1-1.cdh5.5.1.p0.11/lib/flume-ng/lib/flume-sources-1.0-SNAPSHOT.jar
```



Change the Flume Agent Name



Configuring the Flume Agent



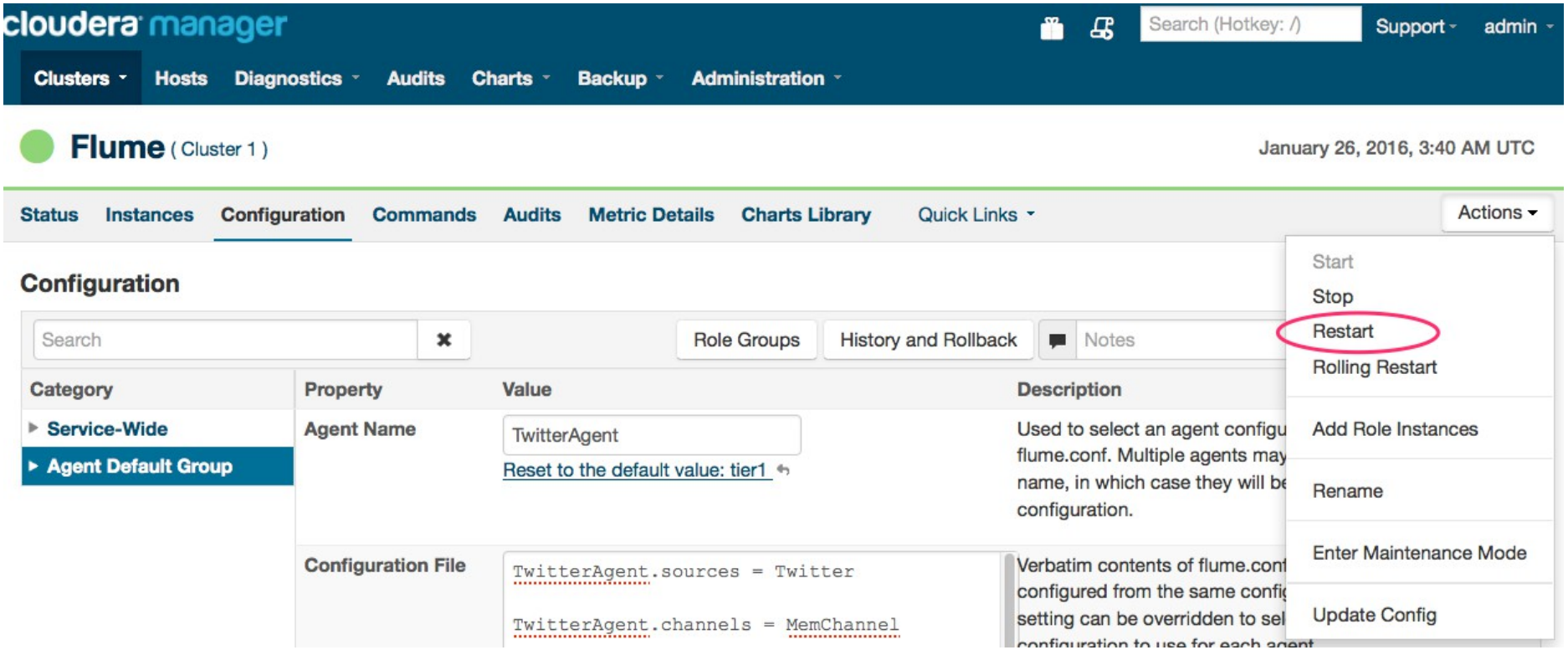
Agent Configuration

```
TwitterAgent.sources = Twitter

TwitterAgent.channels = MemChannel

TwitterAgent.sinks = HDFS


TwitterAgent.sources.Twitter.type =
org.apache.flume.source.twitter.TwitterSource

TwitterAgent.sources.Twitter.channels = MemChannel

TwitterAgent.sources.Twitter.consumerKey =
MjpswndxVj27ylnpOoSBrnfLX

TwitterAgent.sources.Twitter.consumerSecret =
QYmuBO1smD5Yc3zE0ZF9ByCgeEQxnxUmhRVCisAvPFudYVjC4a

TwitterAgent.sources.Twitter.accessToken = 921172807-
EfMXJj6as2dFECDH1vDe5goyTHcxPrF1RIJozqgx

TwitterAgent.sources.Twitter.accessTokenSecret =
HbpZEVip3D5j80GP21a37HxA4y10dH9BHcgEFXUNcA9xy
```

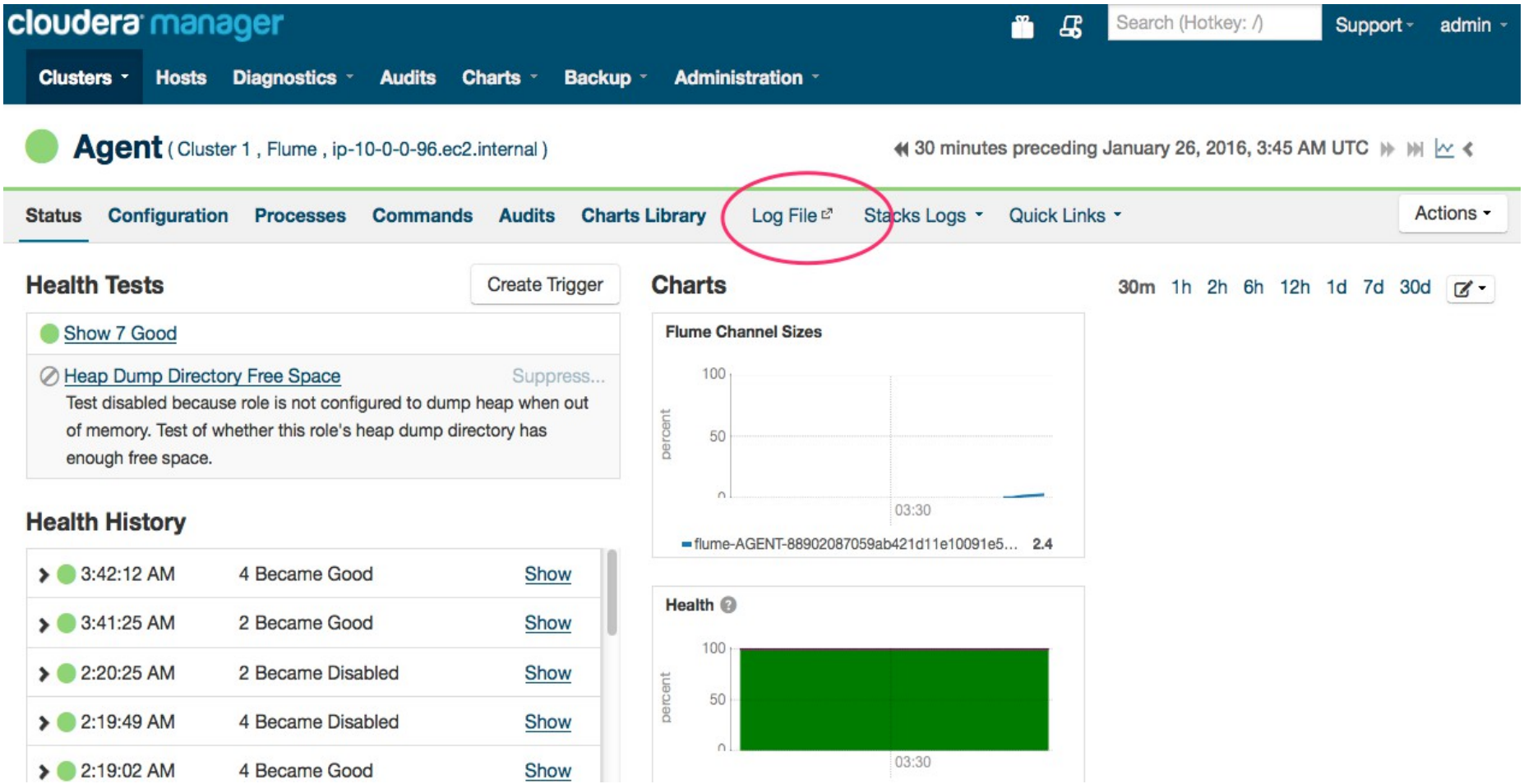
Agent Configuration

```
TwitterAgent.sources.Twitter.keywords = hadoop, big data,
analytics, bigdata, cloudera, data science, data scientiest,
business intelligence, mapreduce, data warehouse, data
warehousing, mahout, hbase, nosql, newsql,
businessintelligence, cloudcomputing

TwitterAgent.sinks.HDFS.channel = MemChannel

TwitterAgent.sinks.HDFS.type = hdfs

TwitterAgent.sinks.HDFS.hdfs.path =
hdfs://xx.xx.xx.xx:8020/user/flume/tweets/

TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream

TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text

TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000

TwitterAgent.sinks.HDFS.hdfs.rollSize = 0

TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000

TwitterAgent.channels.MemChannel.type = memory

TwitterAgent.channels.MemChannel.capacity = 10000

TwitterAgent.channels.MemChannel.transactionCapacity = 100
```

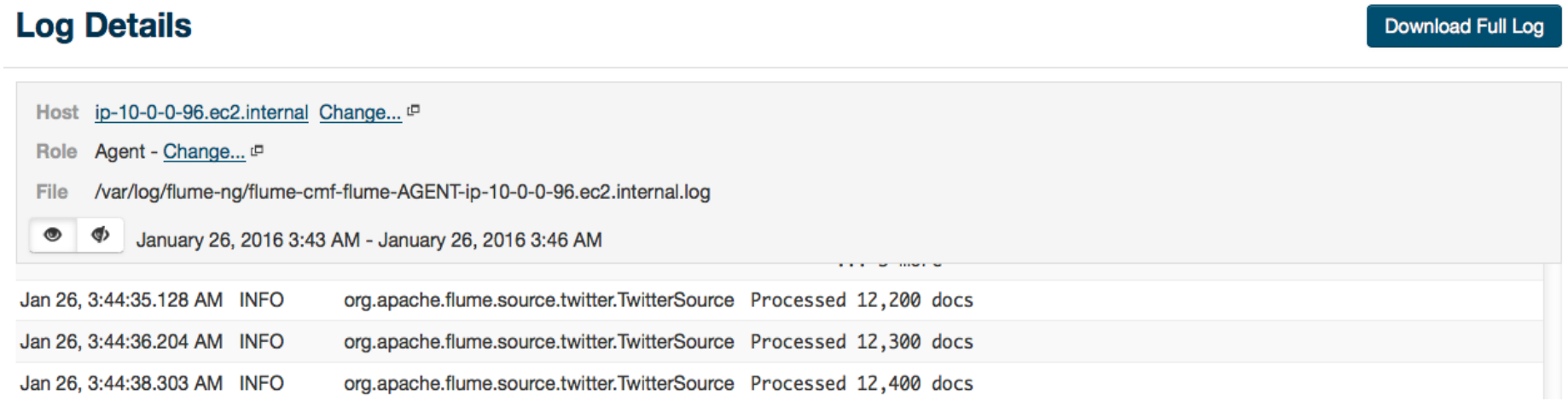
Restart Flume



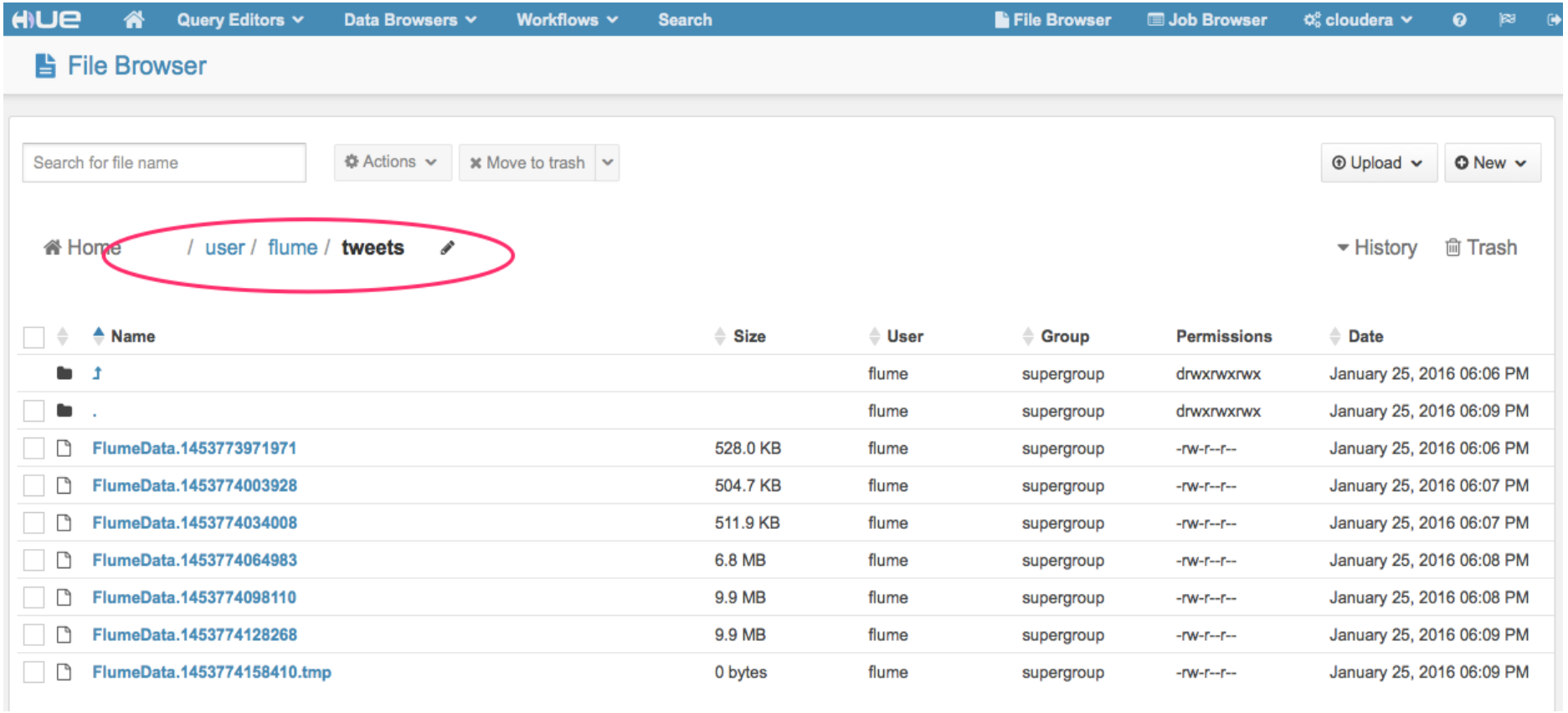
View an agent log file



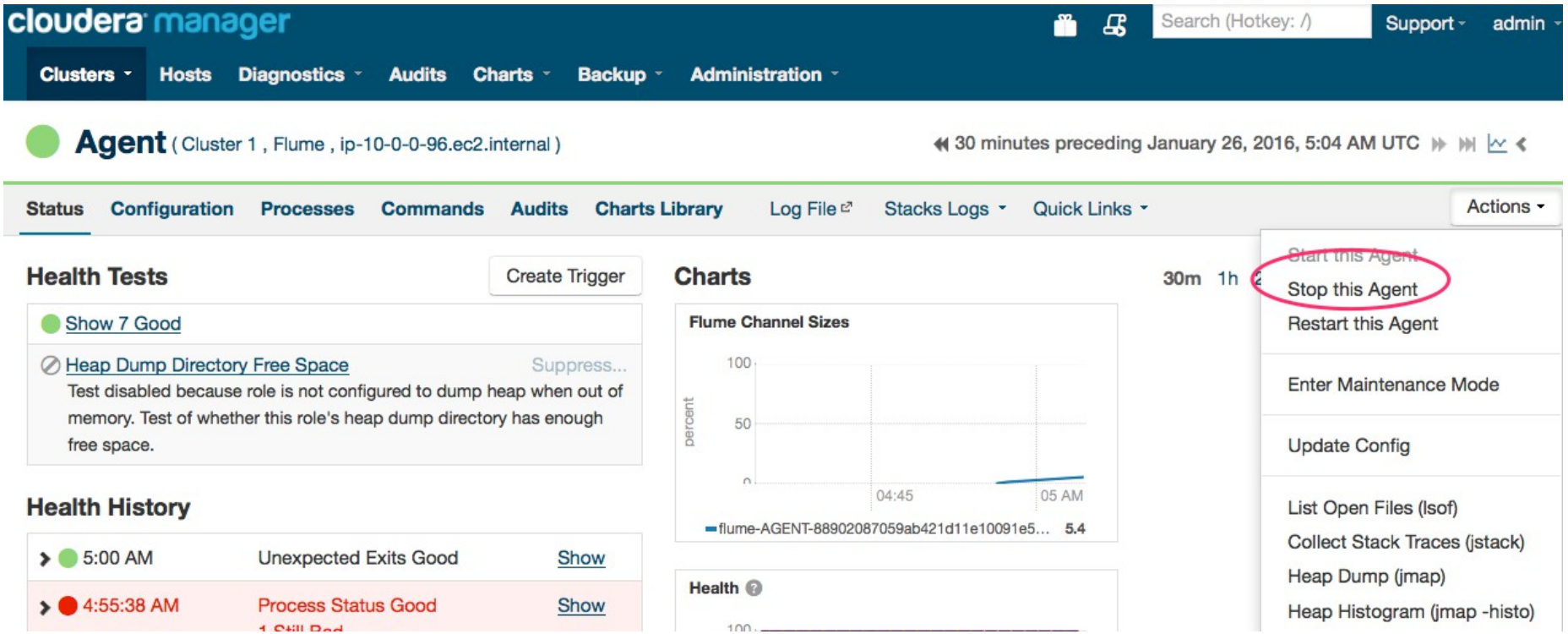
View an agent log file



View a result using Hue



Stop the agent



7. Analyse data using Hive

Get a Serde Jar File for parsing JSON file

```
$ wget
http://files.cloudera.com/samples/hive-serdes-1.0-SNAPSHOT.jar

$ mv hive-serdes-1.0-SNAPSHOT.jar  /usr/local/apache-hive-
1.1.0-bin/lib/

$ hive
```

Register the Jar file.

```
hive> ADD JAR /usr/local/apache-hive-1.1.0-bin/lib/hive-
serdes-1.0-SNAPSHOT.jar;
```

Analyse data using Hive (cont.)

Running the following hive command

```
CREATE EXTERNAL TABLE tweets (
   id BIGINT,
   created_at STRING,
   source STRING,
   favorited BOOLEAN,
   retweet_count INT,
   retweeted_status STRUCT<
      text:STRING,
      user:STRUCT<screen_name:STRING,name:STRING>>,
   entities STRUCT<
      urls:ARRAY<STRUCT<expanded_url:STRING>>,
      user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>,
      hashtags:ARRAY<STRUCT<text:STRING>>>,
   text STRING,
   user STRUCT<
      screen_name:STRING,
      name:STRING,
      friends_count:INT,
      followers_count:INT,
      statuses_count:INT,
      verified:BOOLEAN,
      utc_offset:INT,
      time_zone:STRING>,
   in_reply_to_screen_name STRING
)
ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
LOCATION '/user/flume/tweets';
```

http://www.thecloudavenue.com/2013/03/analyse-tweets-using-flume-hadoop-and.html

Analyse data using Hive (cont)

Finding user who has the most number of followers

```
hive> select user.screen_name, user.followers_count c from
tweets order by c desc;
```

```
Starting Job = job_201504051617_0010, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201504051617_0010
Kill Command = /usr/local/hadoop/libexec/../bin/hadoop job  -kill job_201504051617_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-04-06 15:37:27,782 Stage-1 map = 0%,  reduce = 0%
2015-04-06 15:37:31,837 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.27 sec
2015-04-06 15:37:39,899 Stage-1 map = 100%,  reduce = 33%, Cumulative CPU 1.27 sec
2015-04-06 15:37:40,908 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.42 sec
MapReduce Total cumulative CPU time: 2 seconds 420 msec
Ended Job = job_201504051617_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 2.42 sec   HDFS Read: 170686 HDFS Write: 687 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 420 msec
OK
vinnaum 11523
navchatterji    5485
HCITExpert      4751
NWDCScoop       4097
7wdata  3005
MotivasiMariaP  2007
WesleyBackelant 1977
IFTTTMarketing  1307
jonathangibs    968
ephraimcohen    914
feshob  716
DKajouri        713
```

Apache Kafka

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

Introduction

Open-source message broker project

An open-source message broker project developed by the Apache Software Foundation written in Scala. The project aims to provide a unified, high-throughput, low-latency platform for handling real-time data feeds. It is, in its essence, a "massively scalable pub/sub message queue architected as a distributed transaction log", making it highly valuable for enterprise infrastructures.

What is Kafka?

- An apache project initially developed at LinkedIn
- Distributed publish-subscribe messaging system
- Designed for processing of real time activity stream data e.g. logs, metrics collections
- Written in Scala
- Does not follow JMS Standards, neither uses JMS APIs

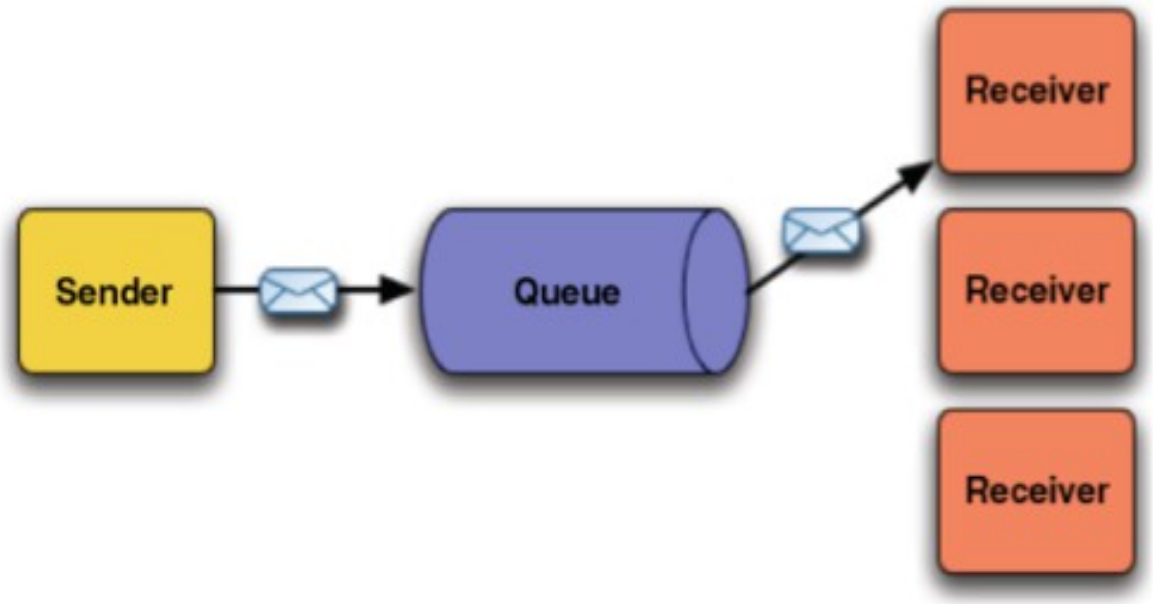
Kafka: Features

- Persistent messaging
- High-throughput
- Supports both queue and topic semantics
- Uses Zookeeper for forming a cluster of nodes (producer/consumer/broker)
- and many more…

Why Kafka?

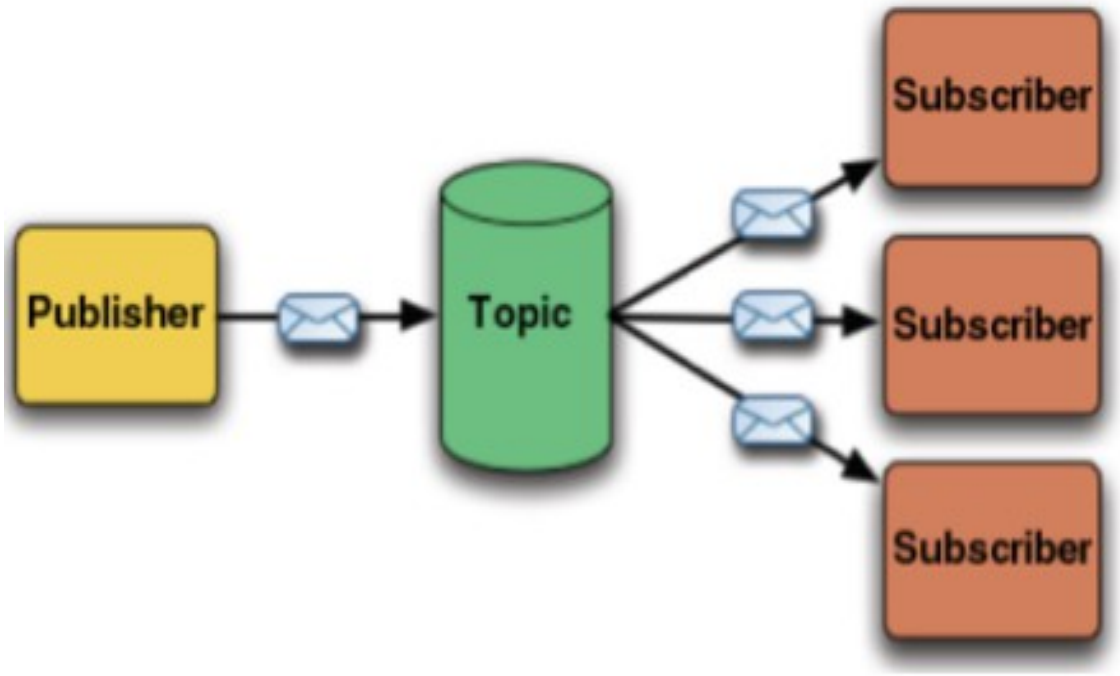
- Built with speed and scalability in mind.
- Enabled near real-time access to any data source
- Empowered hadoop jobs
- Allowed us to build real-time analytics
- Vastly improved our site monitoring and alerting capability
- Enabled us to visualize and track our call graphs.

Messaging System Concept: Queue



Source: Real time Analytics with Apache Kafka and Spark, Rahul Jain

Messaging System Concept: Topic

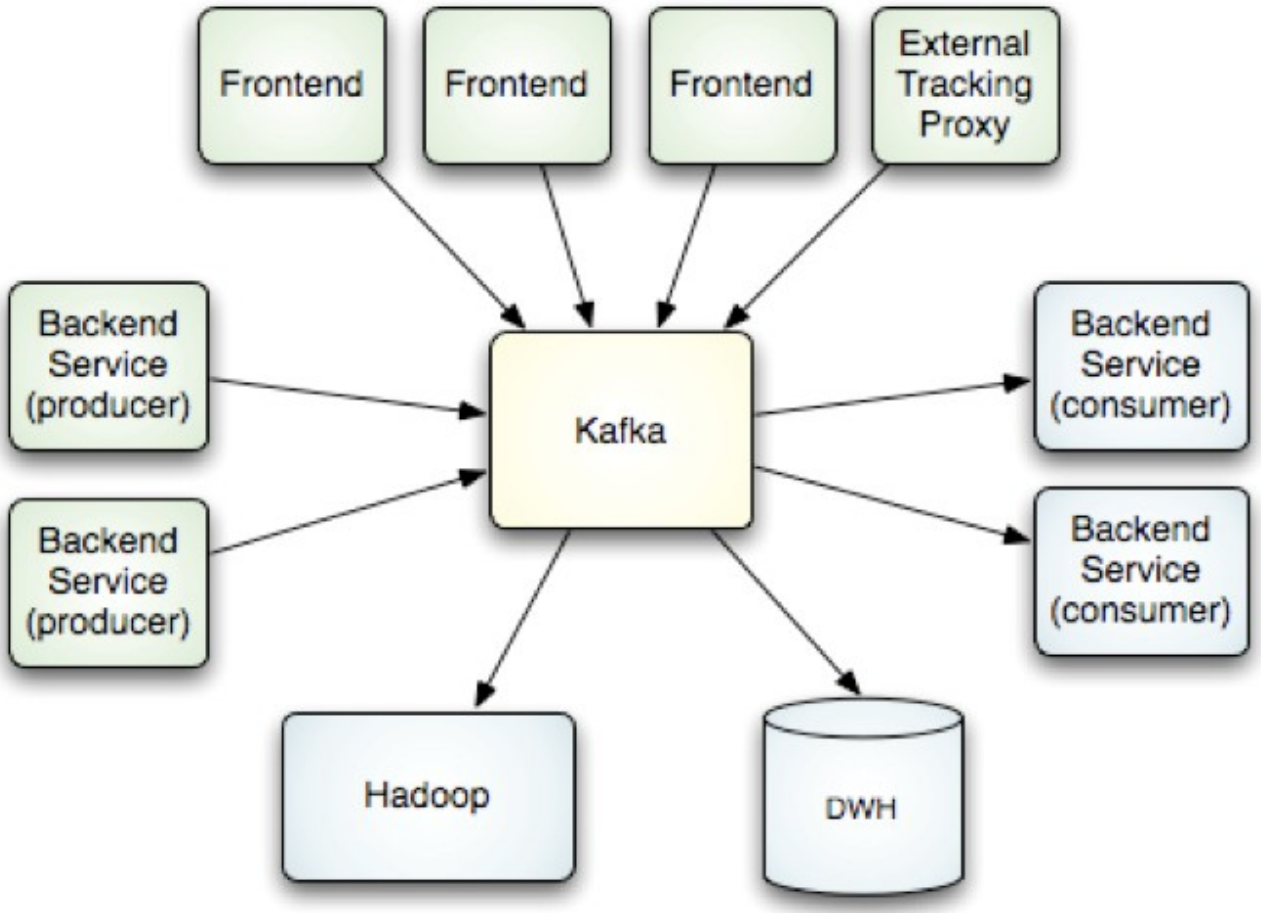


Source: Real time Analytics with Apache Kafka and Spark, Rahul Jain

Terminology

- Kafka maintains feeds of messages in categories called topics.
- Processes that publish messages to a Kafka topic are called producers.
- Processes that subscribe to topics and process the feed of published messages are called consumers.
- Kafka is run as a cluster comprised of one or more servers each of which is called a broker.

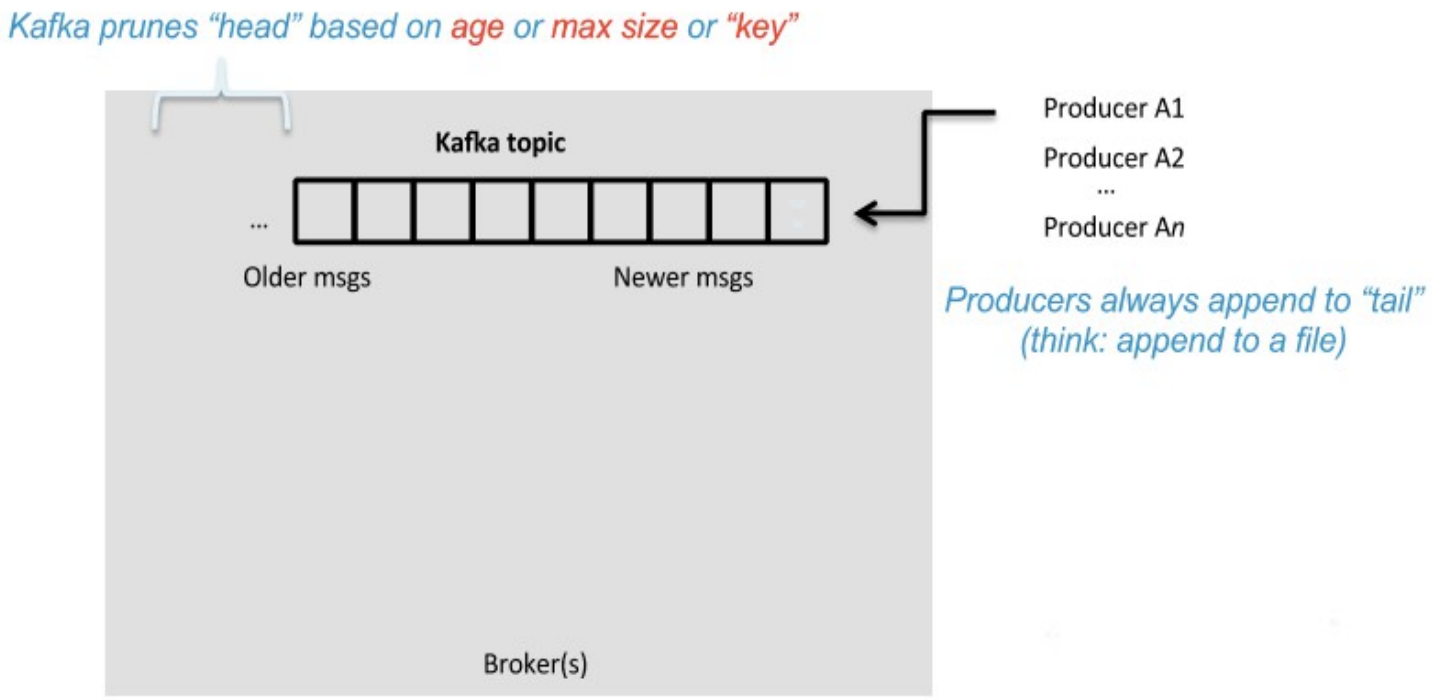
Kafka



Source: Real time Analytics with Apache Kafka and Spark, Rahul Jain

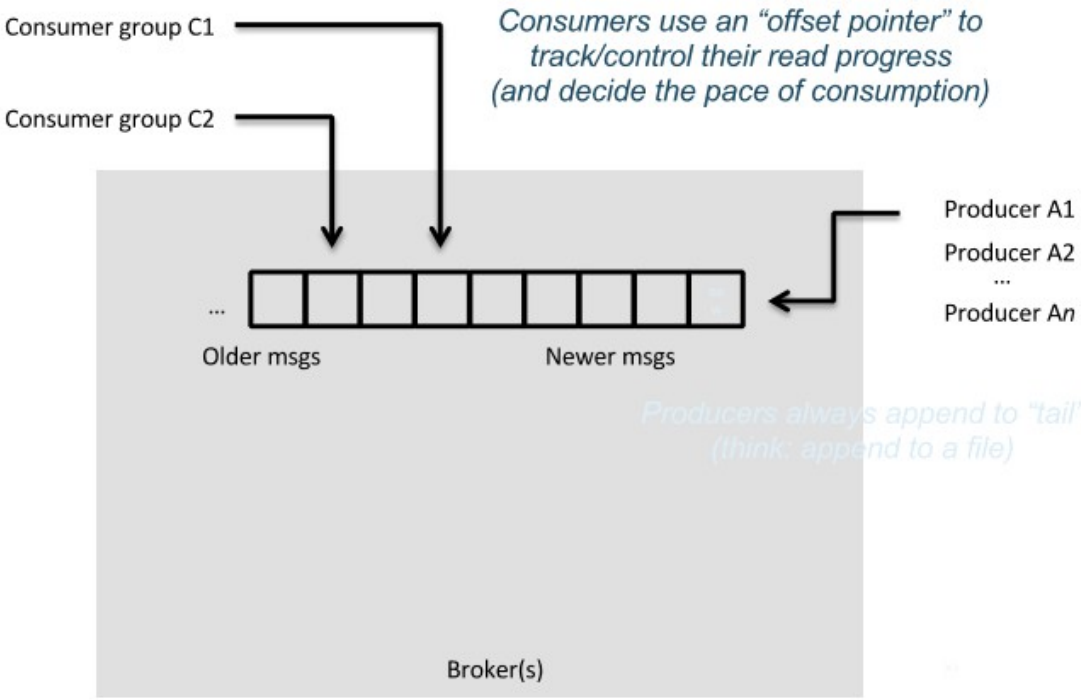
Topics

- Topic: feed name to which messages are published



Source: Apache Kafka with Spark Streaming - Real Time Analytics Redefined

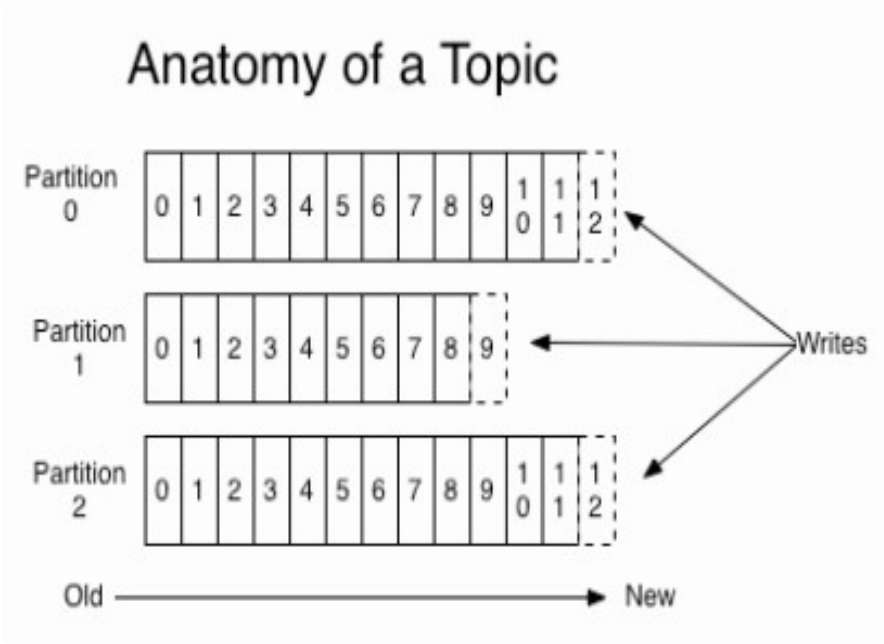
Topics



Source: Apache Kafka with Spark Streaming - Real Time Analytics Redefined

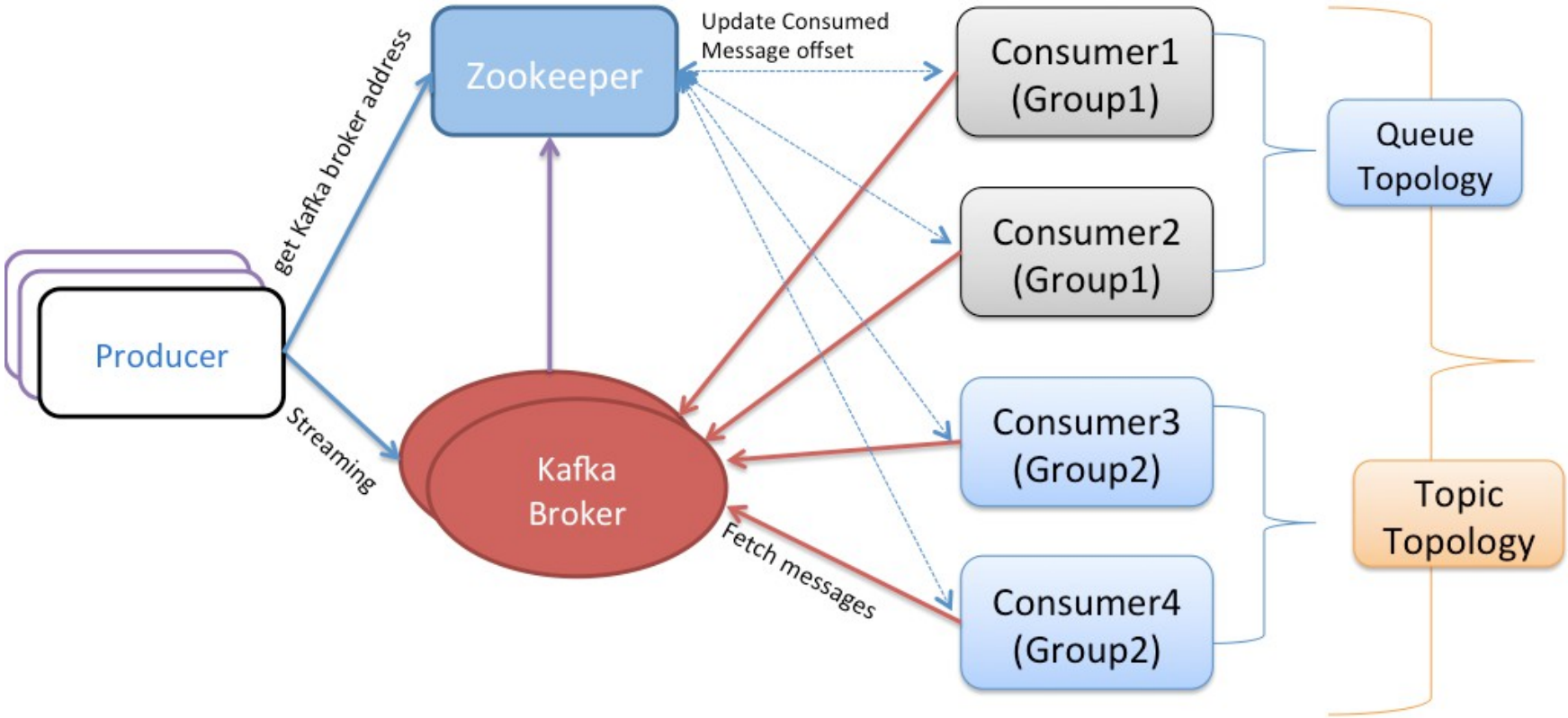
Topics

- A topic consists of partitions.
- Partition: ordered + immutable sequence of messages that is continually appended



Source: Apache Kafka with Spark Streaming - Real Time Analytics Redefined

Kafka Architecture



Source: Real time Analytics with Apache Kafka and Spark, Rahul Jain

Hands-on
SparkStreaming with Kafka

Thanachart Numnonda, thanachart@imcinstitute.com July 2016

Install & Start Kafka Server

```
# wget http://www-us.apache.org/dist/kafka/0.9.0.1/kafka_2.10-0.9.0.1.tgz
# tar xzf kafka_2.10-0.9.0.1.tgz
# cd kafka_2.10-0.9.0.1
# bin/kafka-server-start.sh config/server.properties&
```

```
[2016-06-23 04:37:21,426] INFO Kafka commitId : 23c69d62a0cabf06 (o
rg.apache.kafka.common.utils.AppInfoParser)
[2016-06-23 04:37:21,430] INFO [Kafka Server 0], started (kafka.ser
ver.KafkaServer)
[2016-06-23 04:37:21,446] INFO New leader is 0 (kafka.server.Zookee
perLeaderElector$LeaderChangeListener)
```

Running Kafka Producer

```
# bin/kafka-console-producer.sh --topic test --broker-list
localhost:9092
```

type some random messages followed by Ctrl-D to finish

```
[root@quickstart kafka_2.10-0.9.0.1]# bin/kafka-console-producer.sh
 --topic test --broker-list localhost:9092
This is a test message from IMC Institute

Big Data School
Test
[root@quickstart kafka_2.10-0.9.0.1]#
```

Running Kafka Consumer

```
# bin/kafka-console-consumer.sh --topic test --zookeeper
localhost:2181 --from-beginning
```

```
[root@quickstart kafka_2.10-0.9.0.1]# bin/kafka-console-consumer.sh
 --topic test --zookeeper localhost:2181 --from-beginning
This is a test message from IMC Institute
Big Data School
Test
```

Start Spark-shell with extra memory

```
[root@quickstart ~]# spark-shell --driver-memory 1G
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/or
g/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/jars/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/
StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/
```

Spark Streaming with Kafka

```
$ scala> :paste
import org.apache.spark.SparkConf
import org.apache.spark.streaming.{Seconds, StreamingContext}
import org.apache.spark.storage.StorageLevel
import StorageLevel._
import org.apache.spark._
import org.apache.spark.streaming._
import org.apache.spark.streaming.StreamingContext._
import org.apache.spark.streaming.kafka.KafkaUtils
val ssc = new StreamingContext(sc, Seconds(2))
val kafkaStream = KafkaUtils.createStream(ssc,
"localhost:2181","spark-streaming-consumer-group", Map("spark-
topic" -> 5))
kafkaStream.print()
ssc.start
```

Running Kafka Producer on another terminal

```
# docker ps
```

```
CONTAINER ID        IMAGE                          COMMAND
     CREATED             STATUS              PORTS
   NAMES
c77e4dc1ed9b        cloudera/quickstart:latest   "/usr/bin/docker-q
ui   22 minutes ago      Up 22 minutes       0.0.0.0:8888->8888/tcp
   trusting_newton
```

```
# docker exec -i -t c77e4dc1ed9b /bin/bash
```

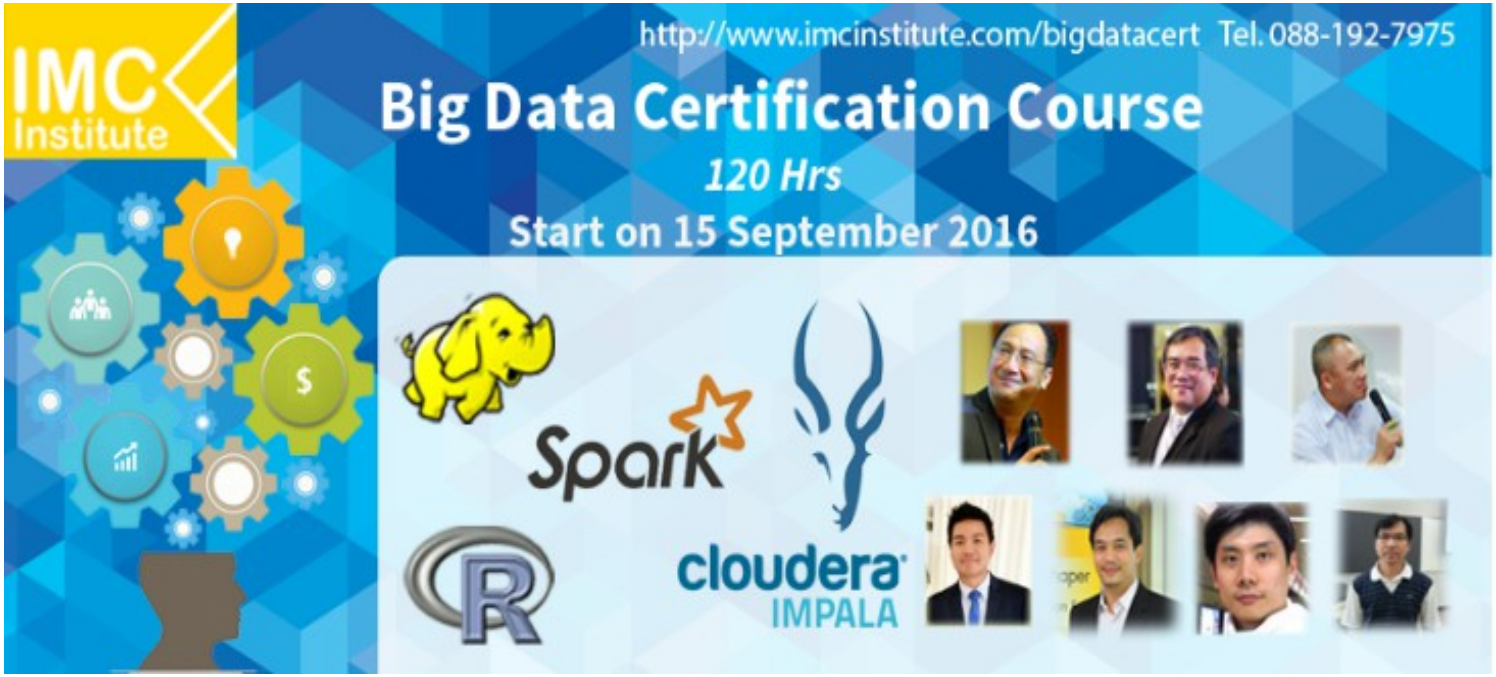
```
[root@quickstart ~]# cd /root/kafka_2.10-0.9.0.1
[root@quickstart kafka_2.10-0.9.0.1]# bin/kafka-console-
producer.sh --broker-list localhost:9092 --topic spark-topic
```

Test & View the result

```
[root@quickstart kafka_2.10-0.9.0.1]# bin/kafka-console-producer.sh
 --broker-list localhost:9092 --topic spark-topic
Hello from IMC Institute
```

Result from another terminal

```
85200 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1466658086000 ms
-------------------------------------------
(null,Hello from IMC Institute)
```


IMC Institute
http://www.imcinstitute.com/bigdatacert Tel. 088-192-7975
Big Data Certification Course
120 Hrs
Start on 15 September 2016
Spark
cloudera
IMPALA

Thank you

www.imcinstitute.com
www.facebook.com/imcinstitute