Thailand Big Data Challenge #2/2016
Apache Spark in Actions

18-19 June 2016

Dr.Thanachart Numnonda
IMC Institute
thanachart@imcinstitute.com

Outline

- Launch Azure Instance
- Install Docker on Ubuntu
- Pull Cloudera QuickStart to the docker
- HDFS
- Spark
- Spark SQL
- Spark Streaming

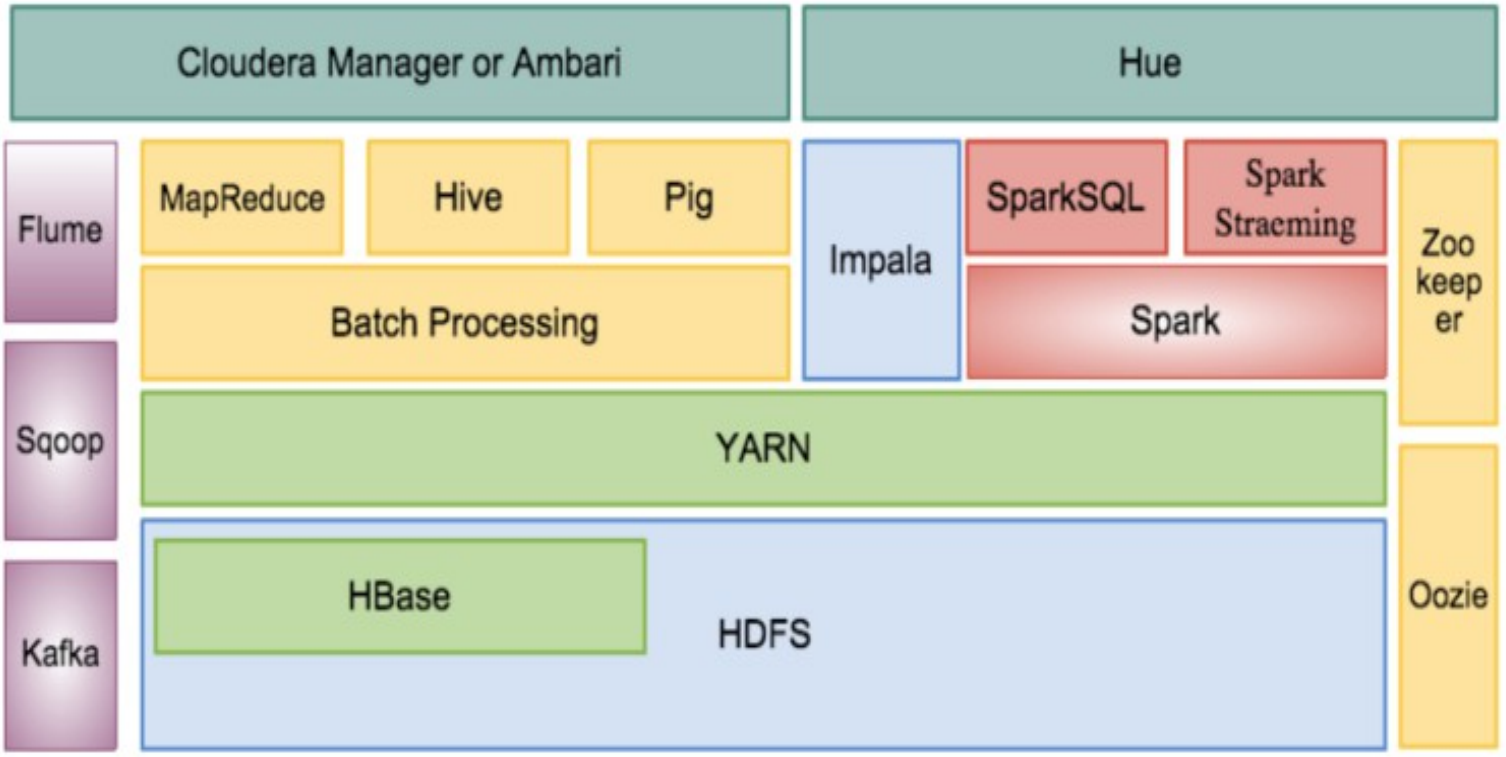
Hive.apache.org

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

Hadoop Ecosystem



Spark Streaming

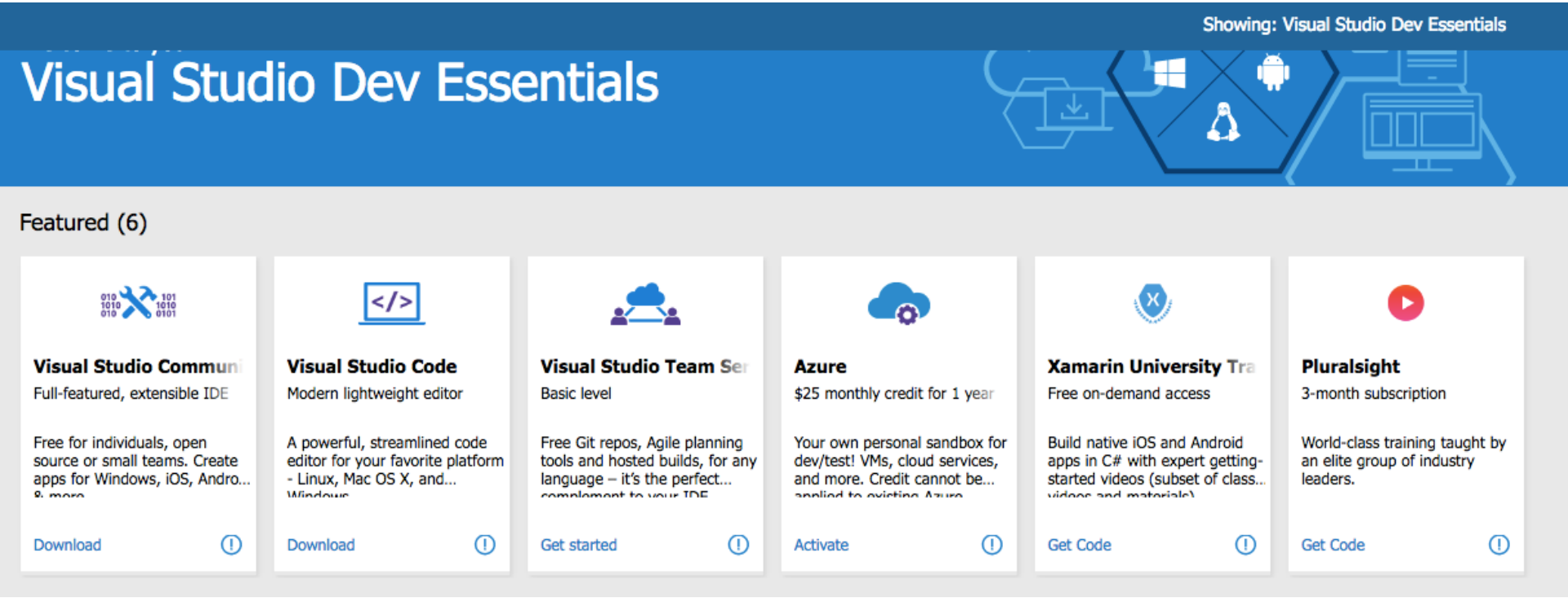
Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Hands-On: Launch a virtual server on Microsoft Azure

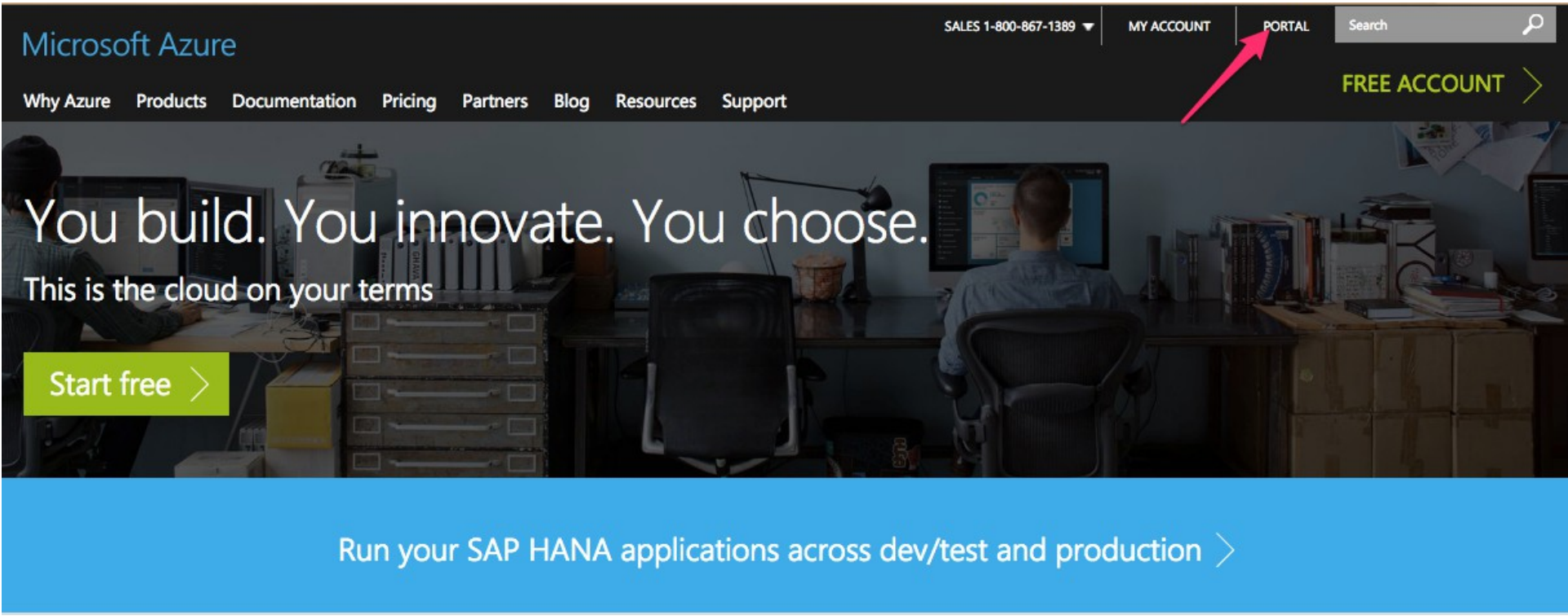
(Note: You can skip this session if you use your own computer or another cloud service)

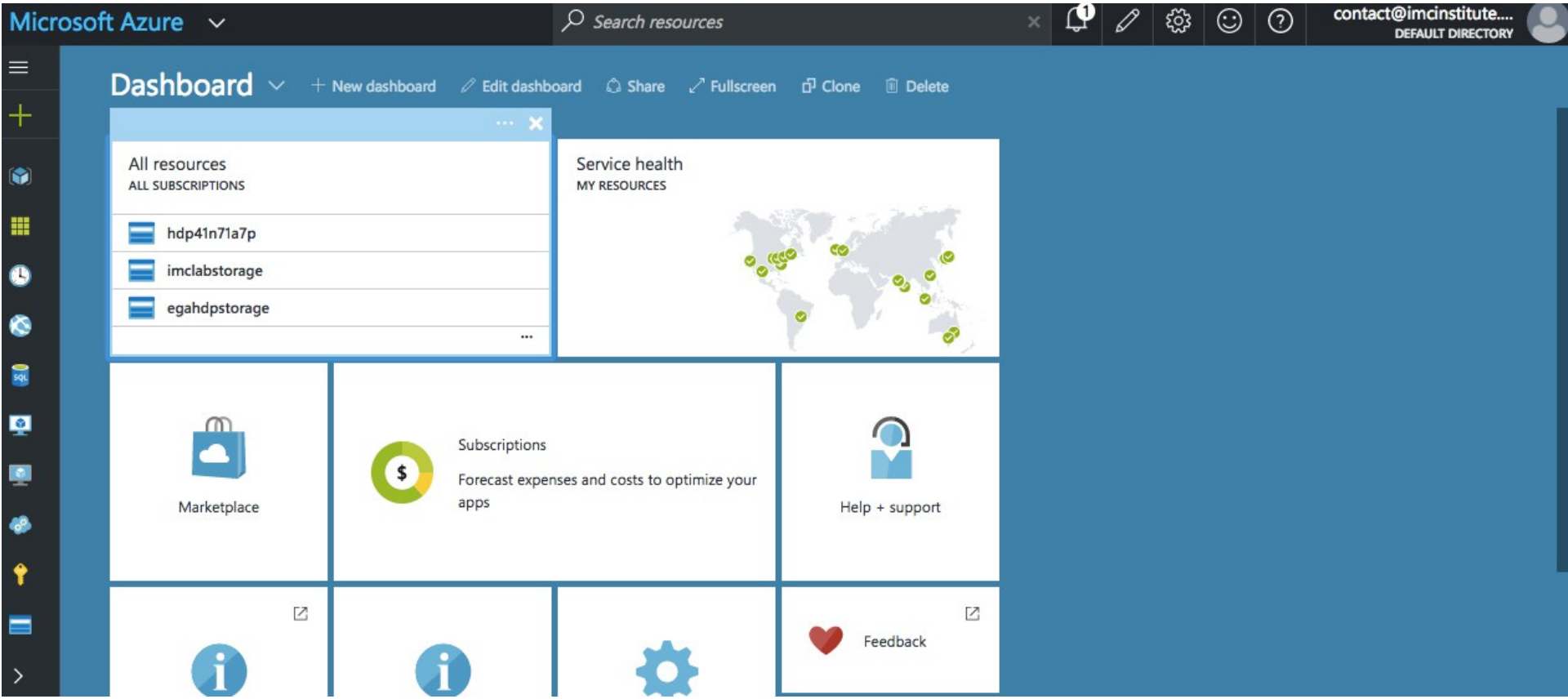
Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Sign up for Visual Studio Dev Essential to get free Azure credit.



Sign in to Azure Portal



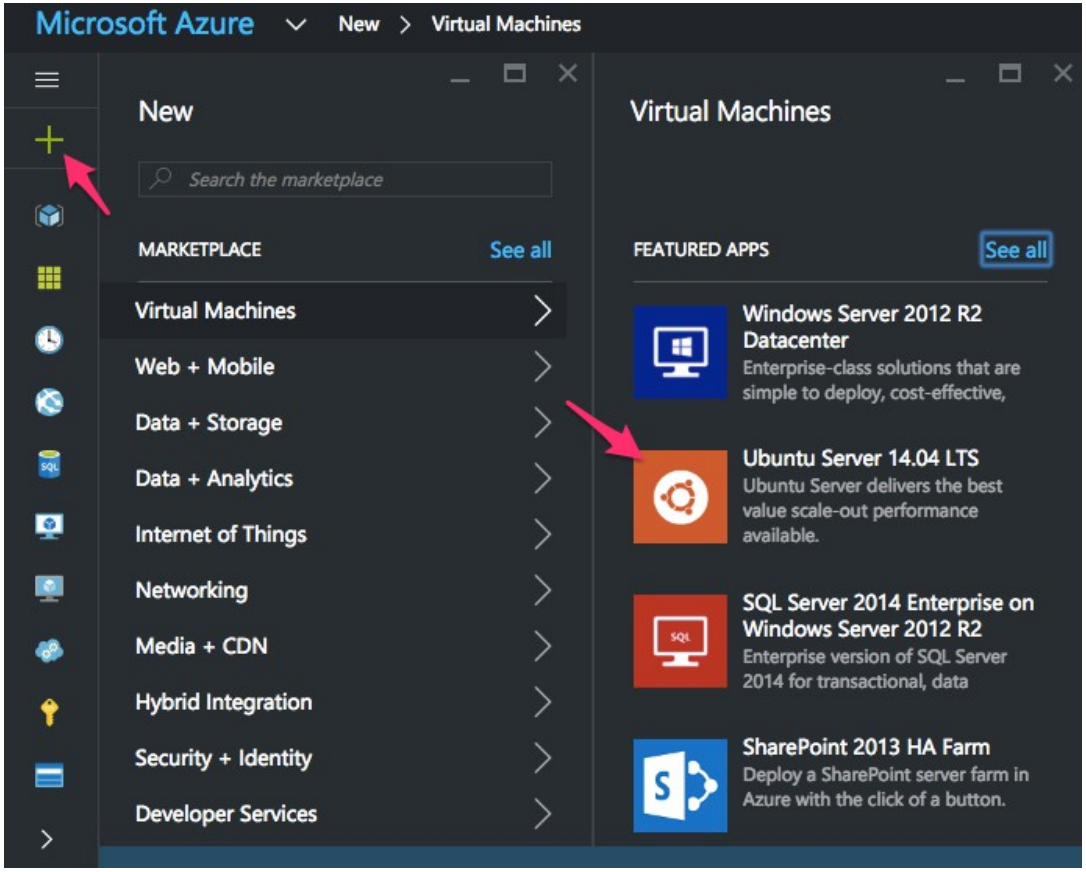

Microsoft Azure
Search resources
contact@imcinstitute....
DEFAULT DIRECTORY
Dashboard
New dashboard
Edit dashboard
Share
Fullscreen
Clone
Delete
All resources
ALL SUBSCRIPTIONS
hdp41n71a7p
imclabstorage
egahdpstorage
Service health
MY RESOURCES
Marketplace
Subscriptions
Forecast expenses and costs to optimize your apps
Help + support
Feedback

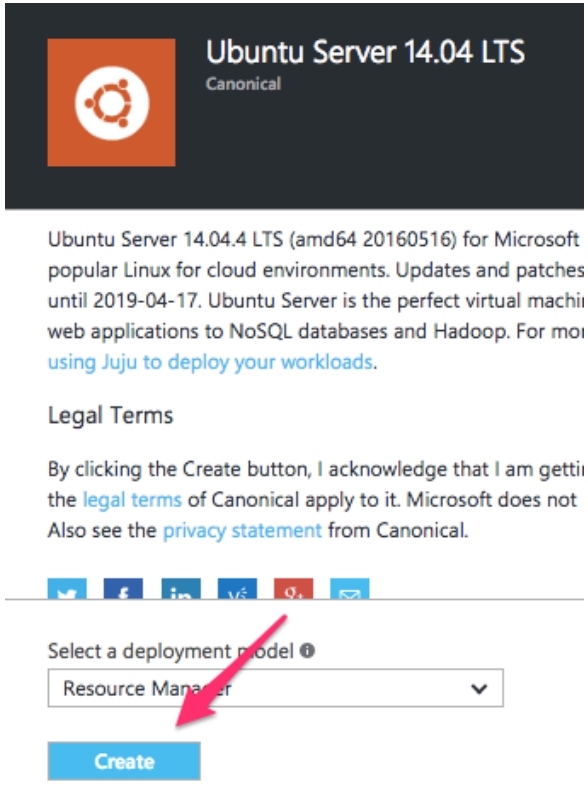
Virtual Server

This lab will use an Azure virtual server to install a Cloudera Quickstart Docker using the following features:

- Ubuntu Server 14.04 LTS
- DS3_V2 Standard 4 Core, 14 GB memory,28 GB SSD

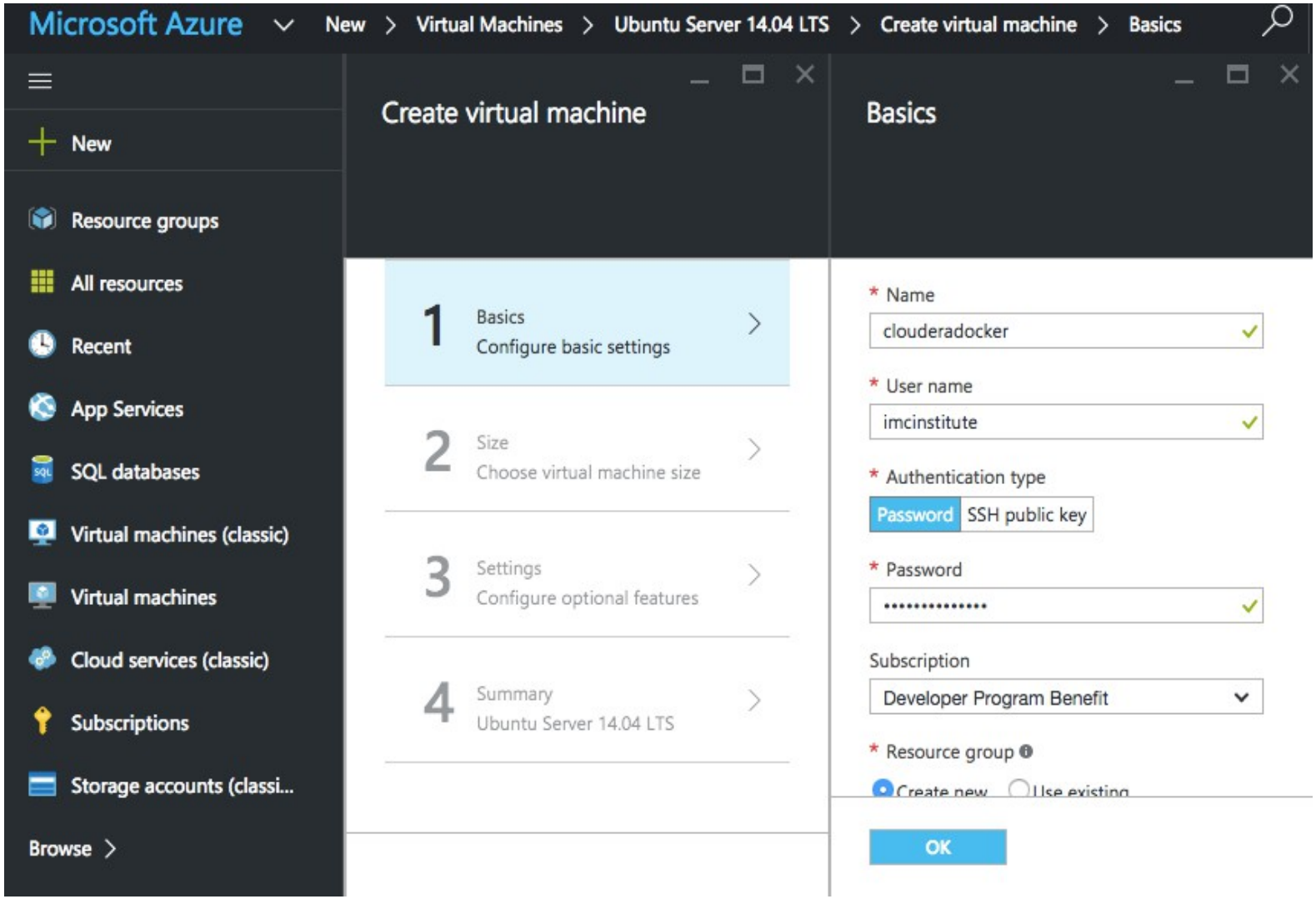
Select New => Virtual Machines => Virtual Machines



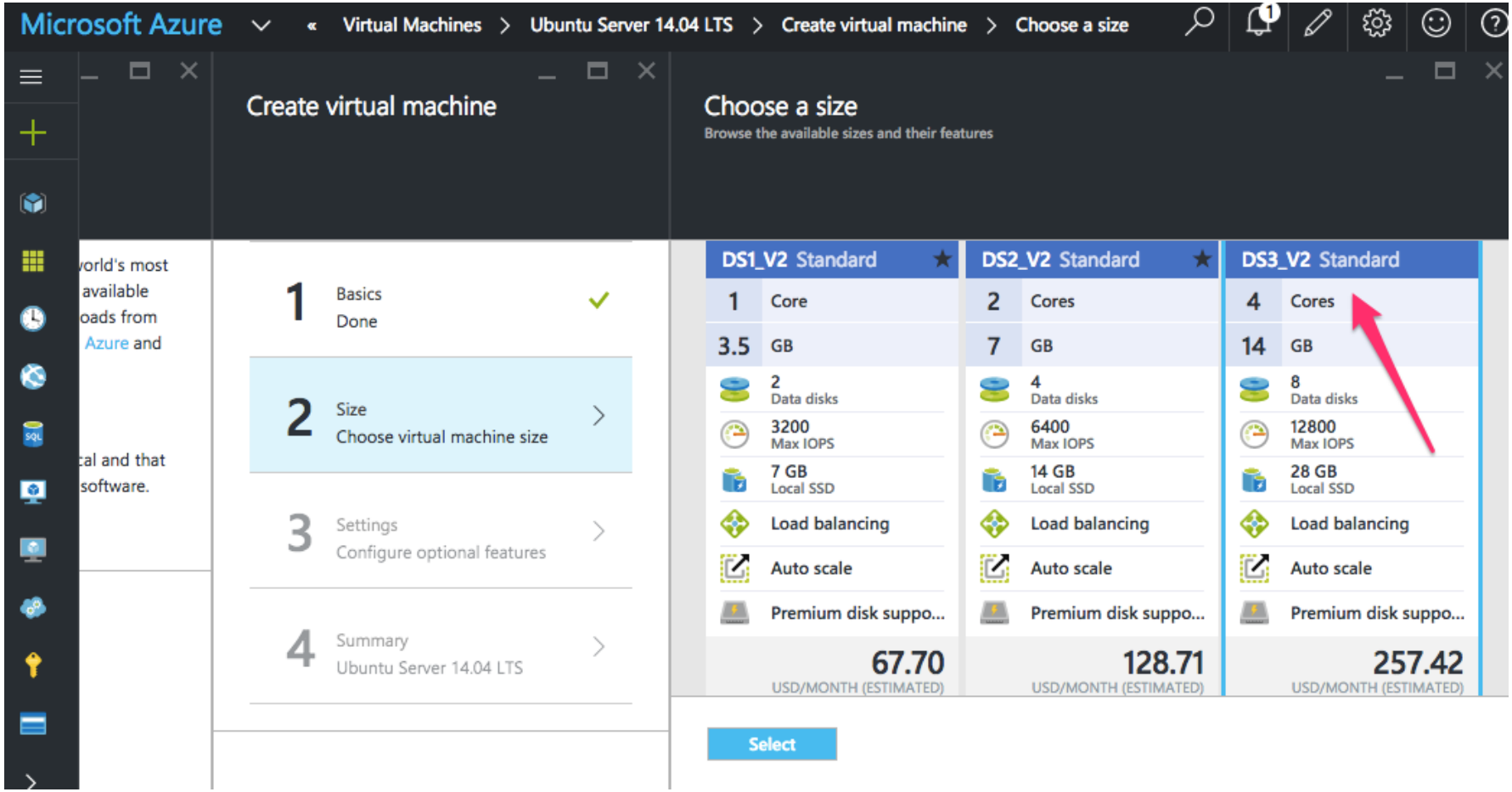


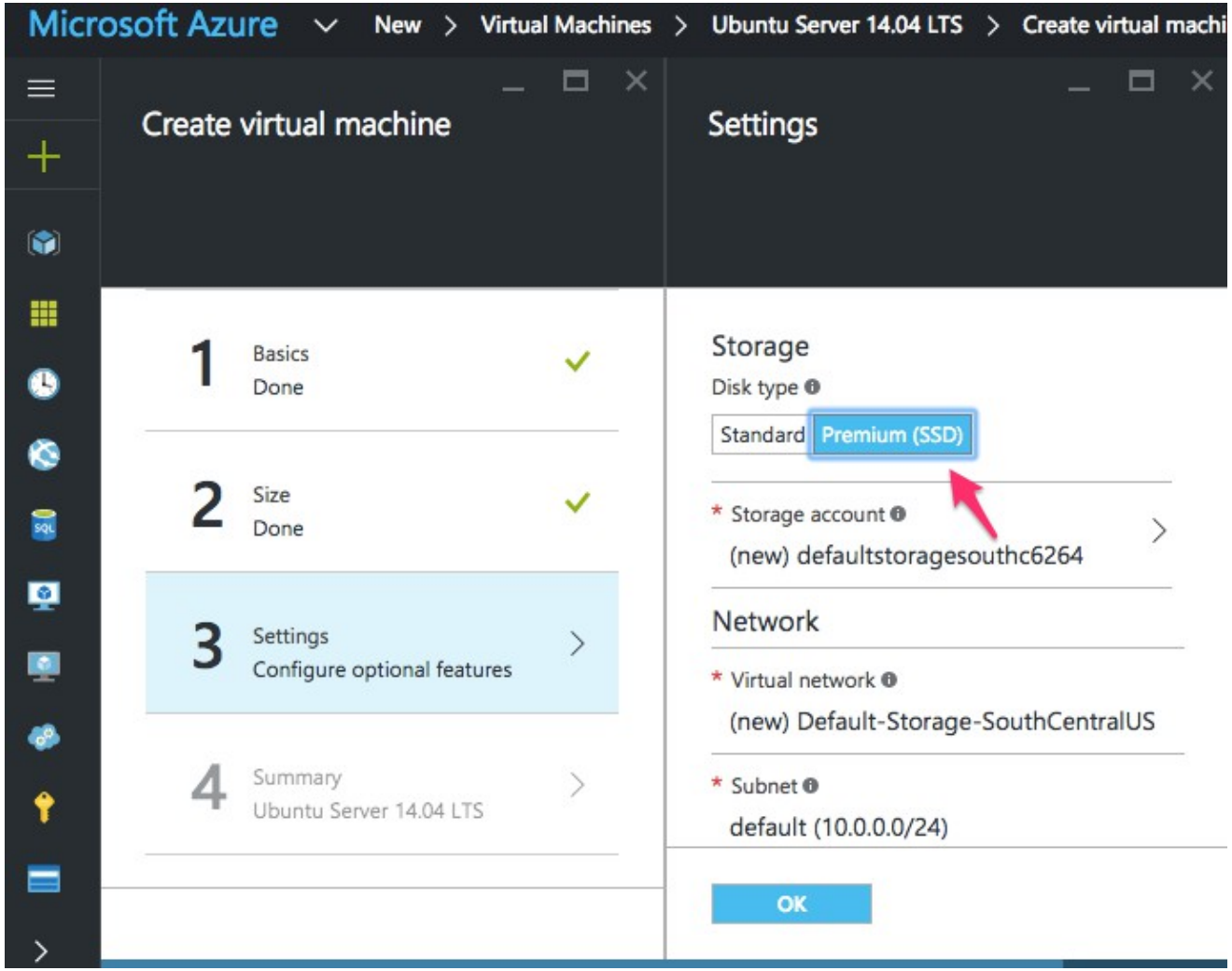
On the Basics page, enter:

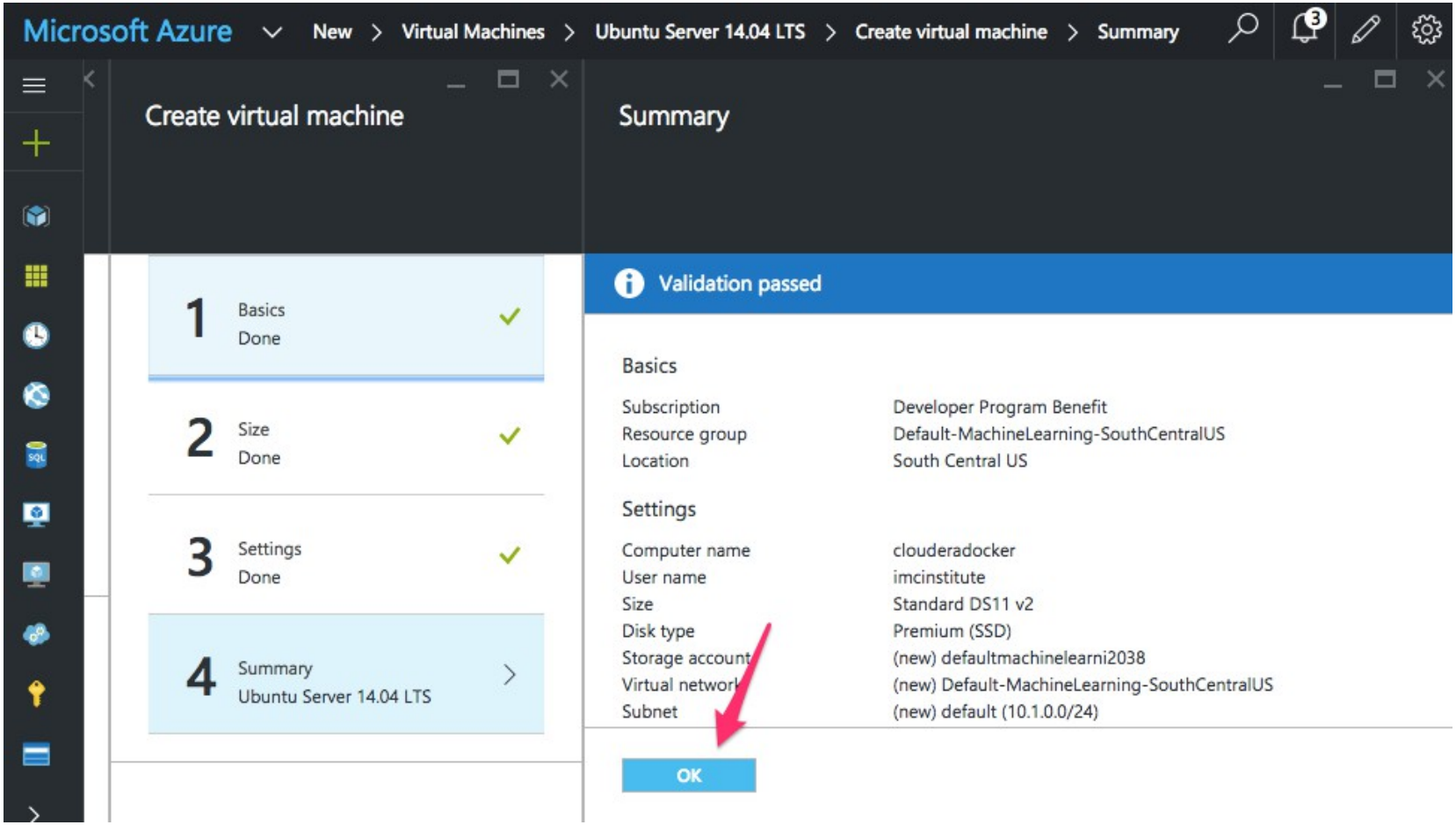
- a name for the VM
- a username for the Admin User
- the Authentication Type set to password
- a password
- a resource group name

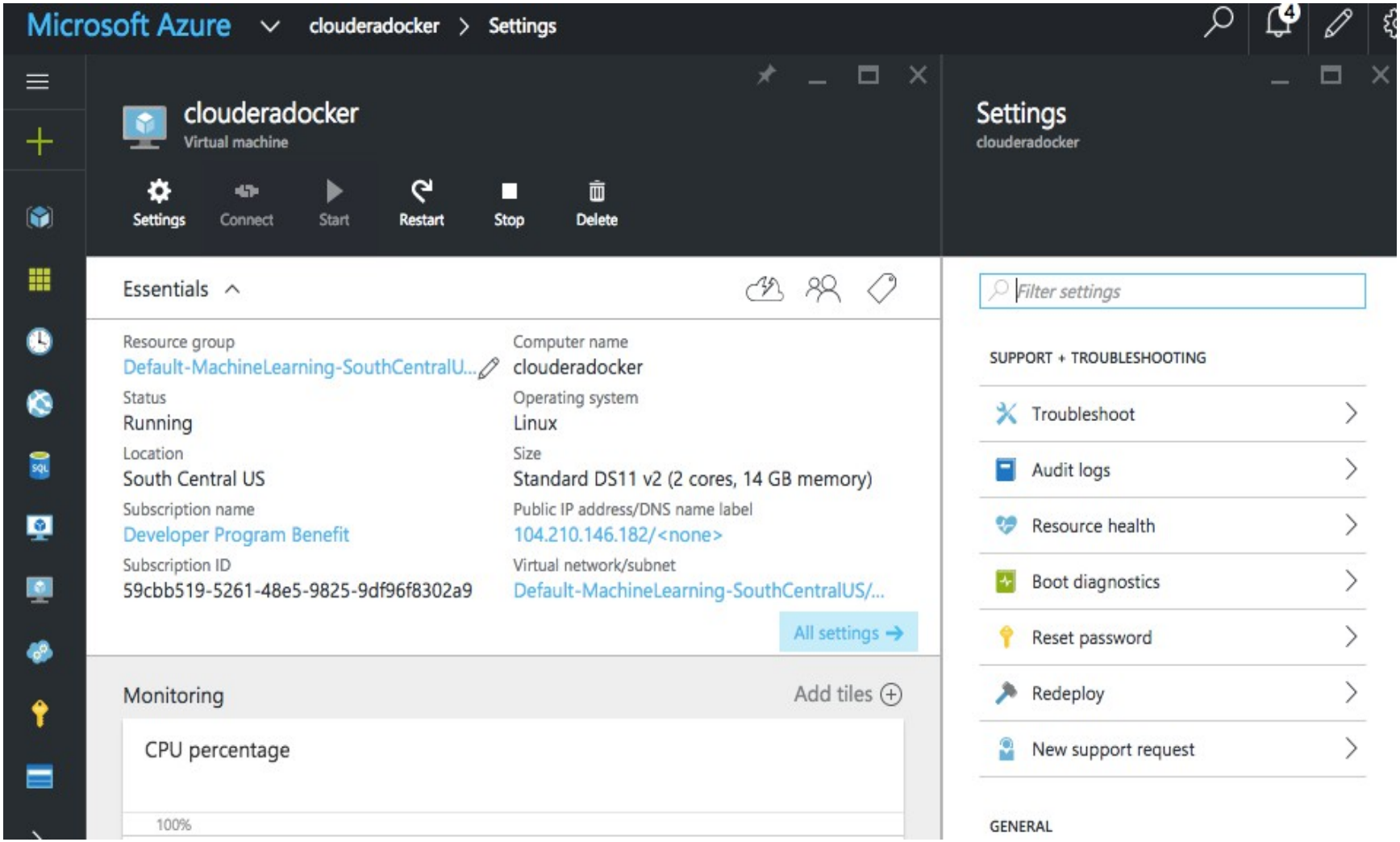

Microsoft Azure
New
Virtual Machines
Ubuntu Server 14.04 LTS
Create virtual machine
Basics
New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Storage accounts (classi...
Browse
Create virtual machine
1 Basics
Configure basic settings
2 Size
Choose virtual machine size
3 Settings
Configure optional features
4 Summary
Ubuntu Server 14.04 LTS
Basics
* Name
clouderadocker
* User name
imcinstitute
* Authentication type
Password
SSH public key
* Password
Subscription
Developer Program Benefit
* Resource group
Create new
Use existing
OK

Choose DS3_v2 Standard

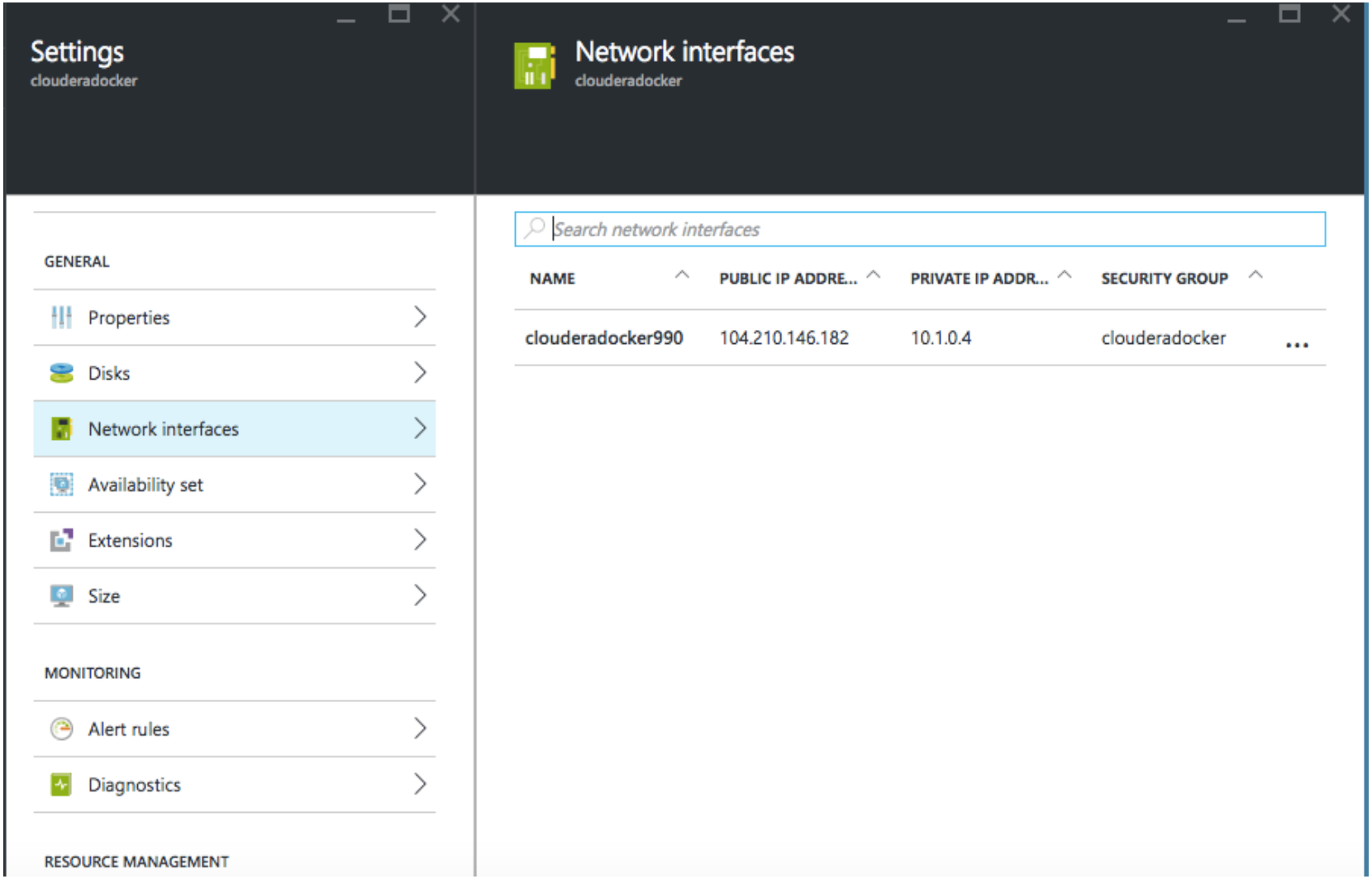


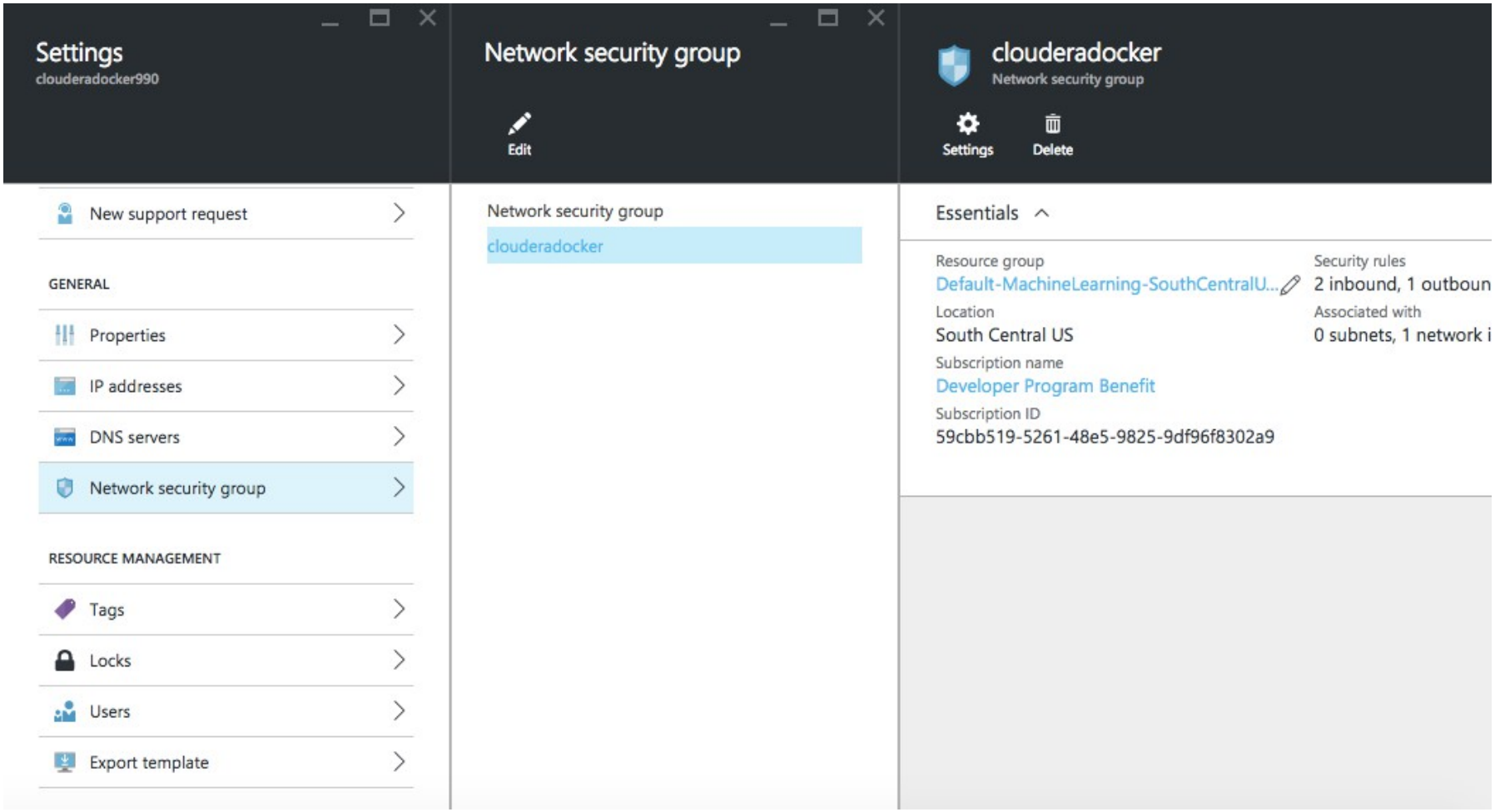

Microsoft Azure
New
Virtual Machines
Ubuntu Server 14.04 LTS
Create virtual machi
Create virtual machine
Settings
1
Basics
Done
2
Size
Done
3
Settings
Configure optional features
4
Summary
Ubuntu Server 14.04 LTS
Storage
Disk type
Standard
Premium (SSD)
* Storage account
(new) defaultstoragesouthc6264
Network
* Virtual network
(new) Default-Storage-SouthCentralUS
* Subnet
default (10.0.0.0/24)
OK

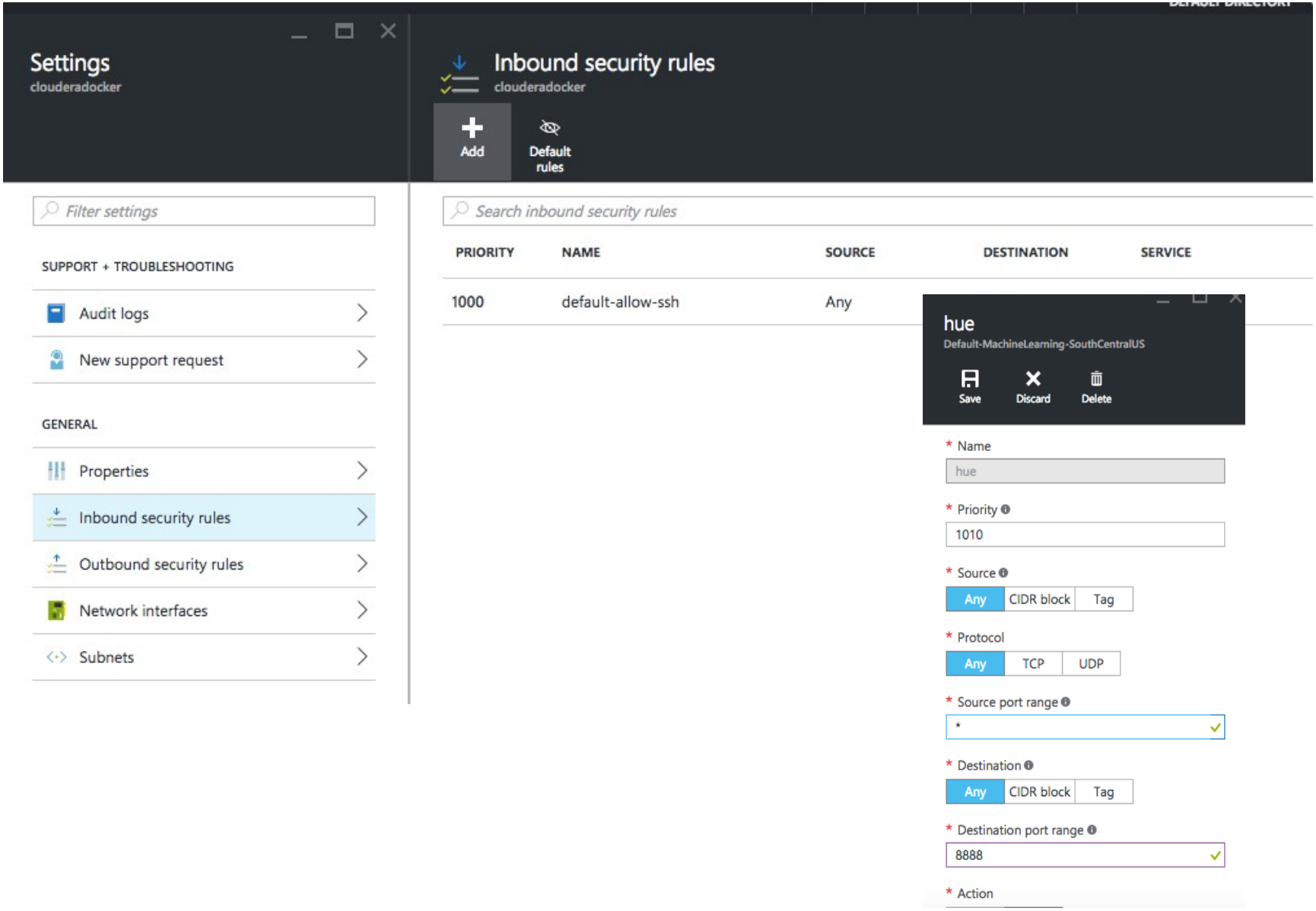

Microsoft Azure
New
Virtual Machines
Ubuntu Server 14.04 LTS
Create virtual machine
Summary
Create virtual machine
Summary
1 Basics Done
2 Size Done
3 Settings Done
4 Summary Ubuntu Server 14.04 LTS
Validation passed
Basics
Subscription Developer Program Benefit
Resource group Default-MachineLearning-SouthCentralUS
Location South Central US
Settings
Computer name clouderadocker
User name imcinstitute
Size Standard DS11 v2
Disk type Premium (SSD)
Storage account (new) defaultmachinelearni2038
Virtual network (new) Default-MachineLearning-SouthCentralUS
Subnet (new) default (10.1.0.0/24)
OK

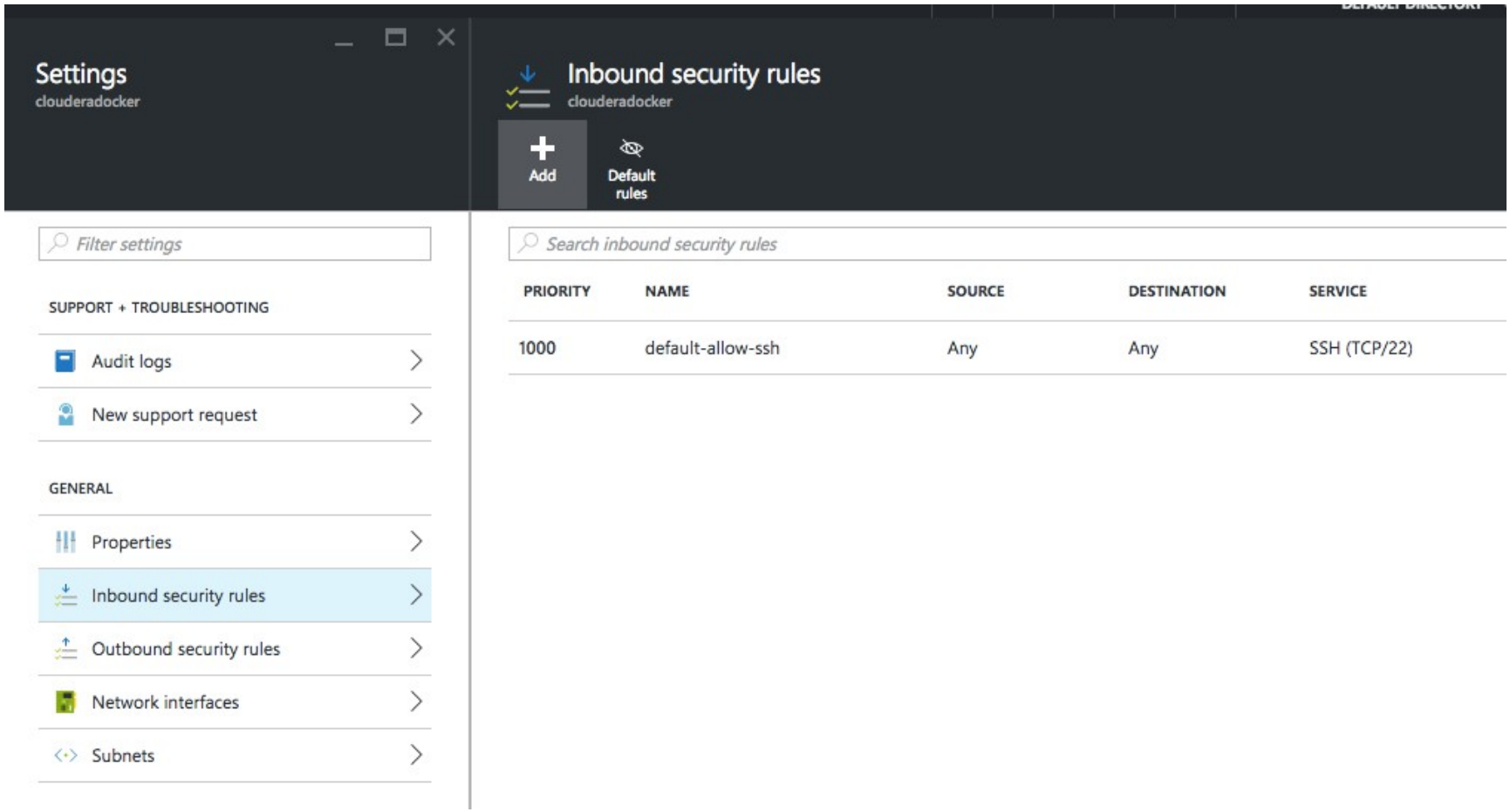

Microsoft Azure
cloderadocker
Settings
cloderadocker
Virtual machine
Settings
Connect
Start
Restart
Stop
Delete
Essentials
Resource group
Default-MachineLearning-SouthCentralU...
Computer name
cloderadocker
Status
Running
Operating system
Linux
Location
South Central US
Size
Standard DS11 v2 (2 cores, 14 GB memory)
Subscription name
Developer Program Benefit
Public IP address/DNS name label
104.210.146.182/<none>
Subscription ID
59cbb519-5261-48e5-9825-9df96f8302a9
Virtual network/subnet
Default-MachineLearning-SouthCentralUS/...
All settings
Monitoring
Add tiles
CPU percentage
100%
Settings
cloderadocker
Filter settings
SUPPORT + TROUBLESHOOTING
Troubleshoot
Audit logs
Resource health
Boot diagnostics
Reset password
Redeploy
New support request
GENERAL

Setting the inbound port for Hue (8888)

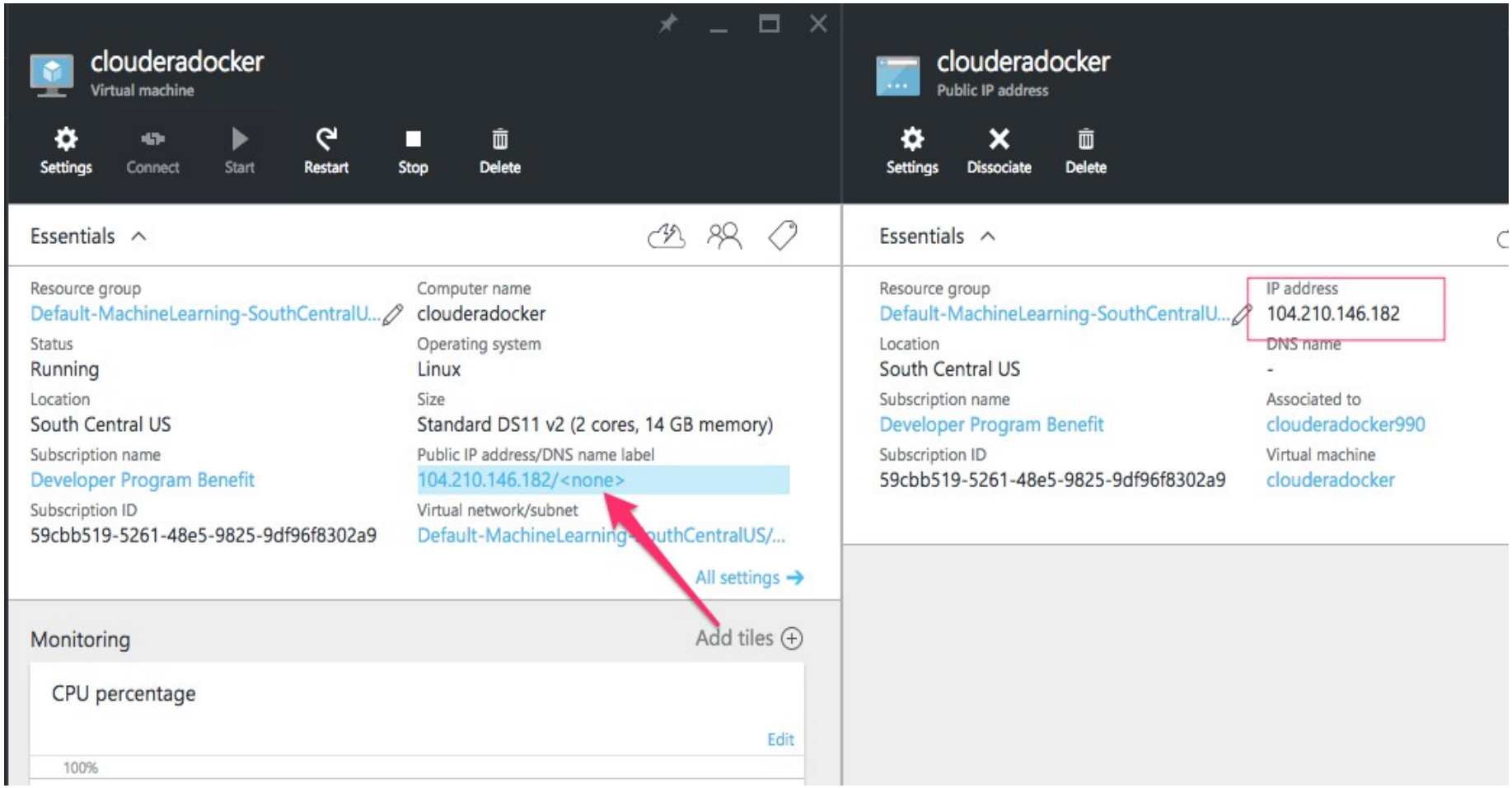



Settings
clouderadocker990
New support request
GENERAL
Properties
IP addresses
DNS servers
Network security group
RESOURCE MANAGEMENT
Tags
Locks
Users
Export template
Network security group
Edit
Network security group
clouderadocker
clouderadocker
Network security group
Settings
Delete
Essentials
Resource group
Default-MachineLearning-SouthCentralU...
Location
South Central US
Subscription name
Developer Program Benefit
Subscription ID
59cbb519-5261-48e5-9825-9df96f8302a9
Security rules
2 inbound, 1 outboun
Associated with
0 subnets, 1 network i


Settings
clouderadocker
Inbound security rules
clouderadocker
Add
Default rules
Filter settings
SUPPORT + TROUBLESHOOTING
Audit logs
New support request
GENERAL
Properties
Inbound security rules
Outbound security rules
Network interfaces
Subnets
Search inbound security rules
PRIORITY
NAME
SOURCE
DESTINATION
SERVICE
1000
default-allow-ssh
Any
hue
Default-MachineLearning-SouthCentralUS
Save
Discard
Delete
* Name
hue
* Priority
1010
* Source
Any
CIDR block
Tag
* Protocol
Any
TCP
UDP
* Source port range
*
* Destination
Any
CIDR block
Tag
* Destination port range
8888
* Action


Settings
clouderadocker
Inbound security rules
clouderadocker
Add
Default rules
Filter settings
Search inbound security rules
SUPPORT + TROUBLESHOOTING
Audit logs
New support request
GENERAL
Properties
Inbound security rules
Outbound security rules
Network interfaces
Subnets
PRIORITY
NAME
SOURCE
DESTINATION
SERVICE
1000
default-allow-ssh
Any
Any
SSH (TCP/22)

Get the IP address

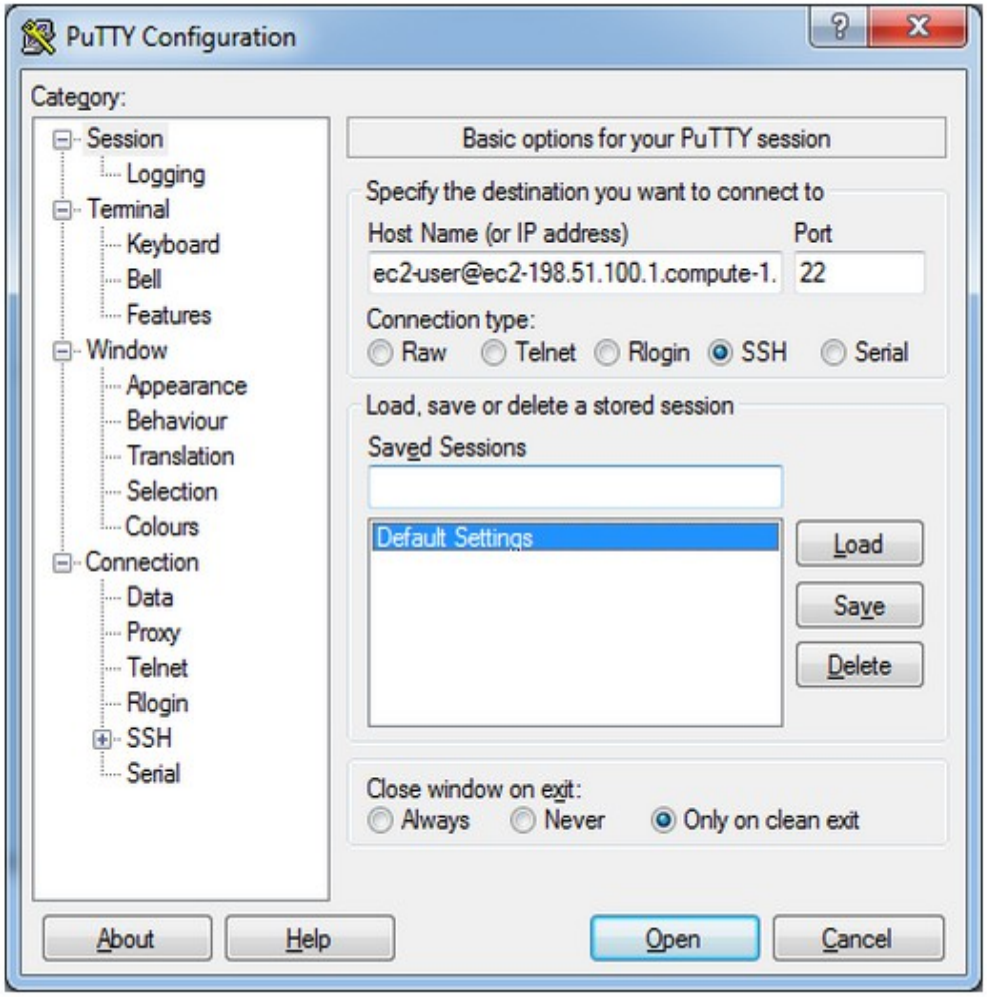


Connect to an instance from Mac/Linux
ssh -i ~/.ssh/id_rsa imcinstitute@104.210.146.182

```
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
```

```
imcinstitute@clouderadocker:~$
```

Connect to an instance from Windows using Putty



Hands-On: Installing Cloudera Quickstart on Docker Container

Thanachart Numnonda, thanachart@imcinstitute.com June 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
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

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
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Docker commands:

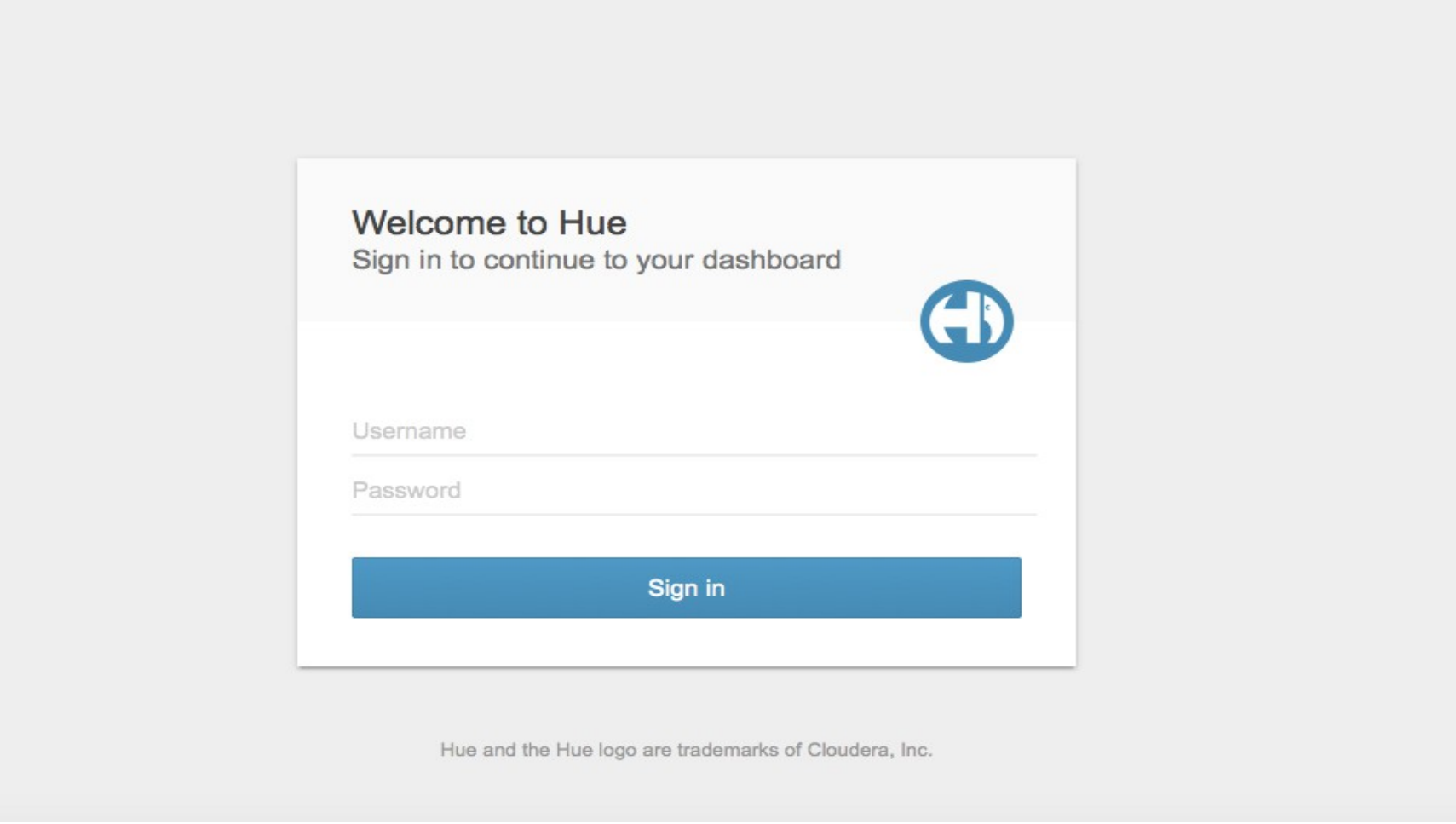
- *docker images*
- *docker ps*
- *docker attach id*
- *docker kill id*

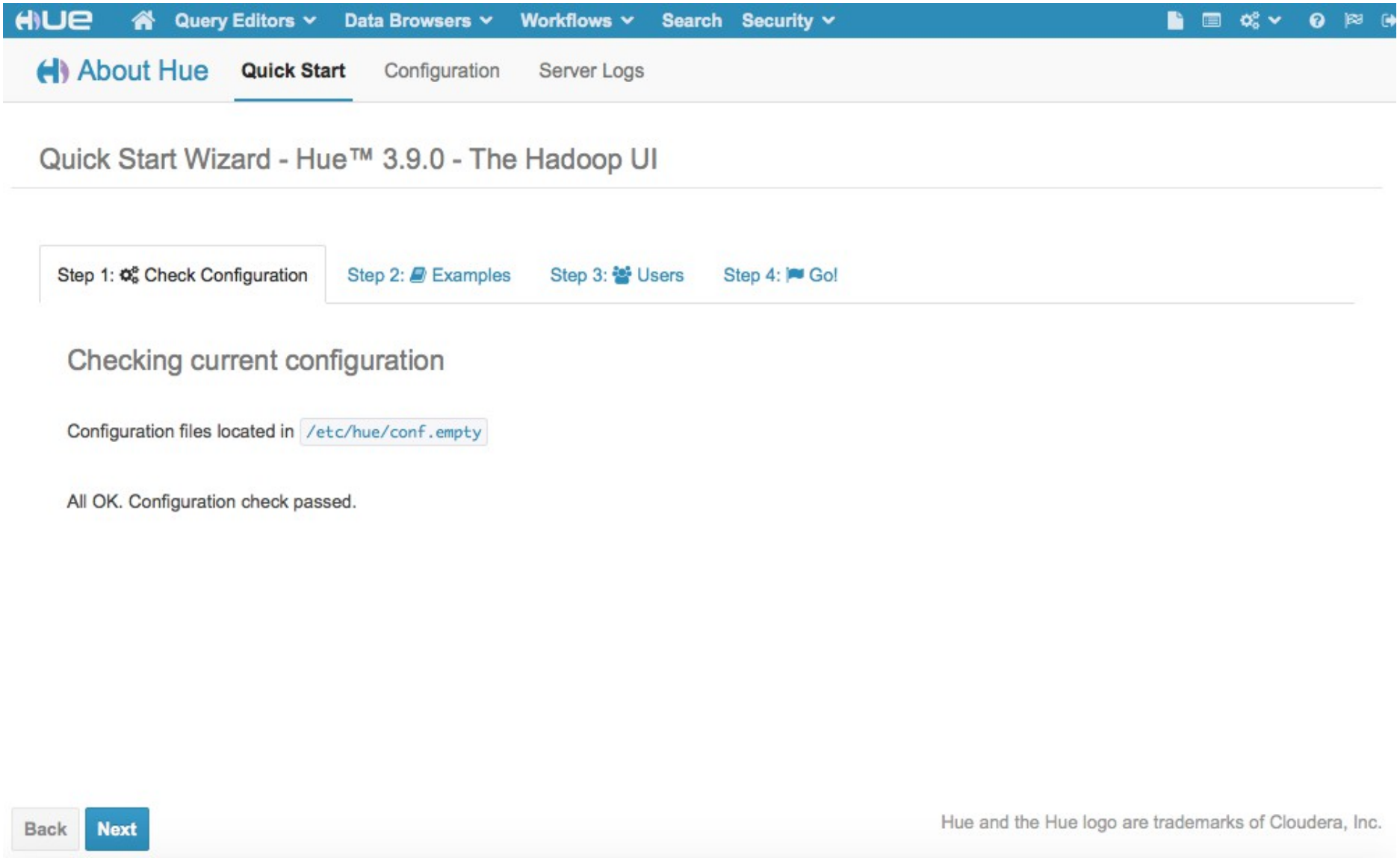
- *Exit from container*
  - *exit (exit & kill the running image)*
  - *Ctrl-P, Ctrl-Q (exit without killing the running image)*

Hive.apache.org

Login to Hue

http://104.210.146.182:8888




HUE
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.

Hands-On: Importing/Exporting Data to HDFS

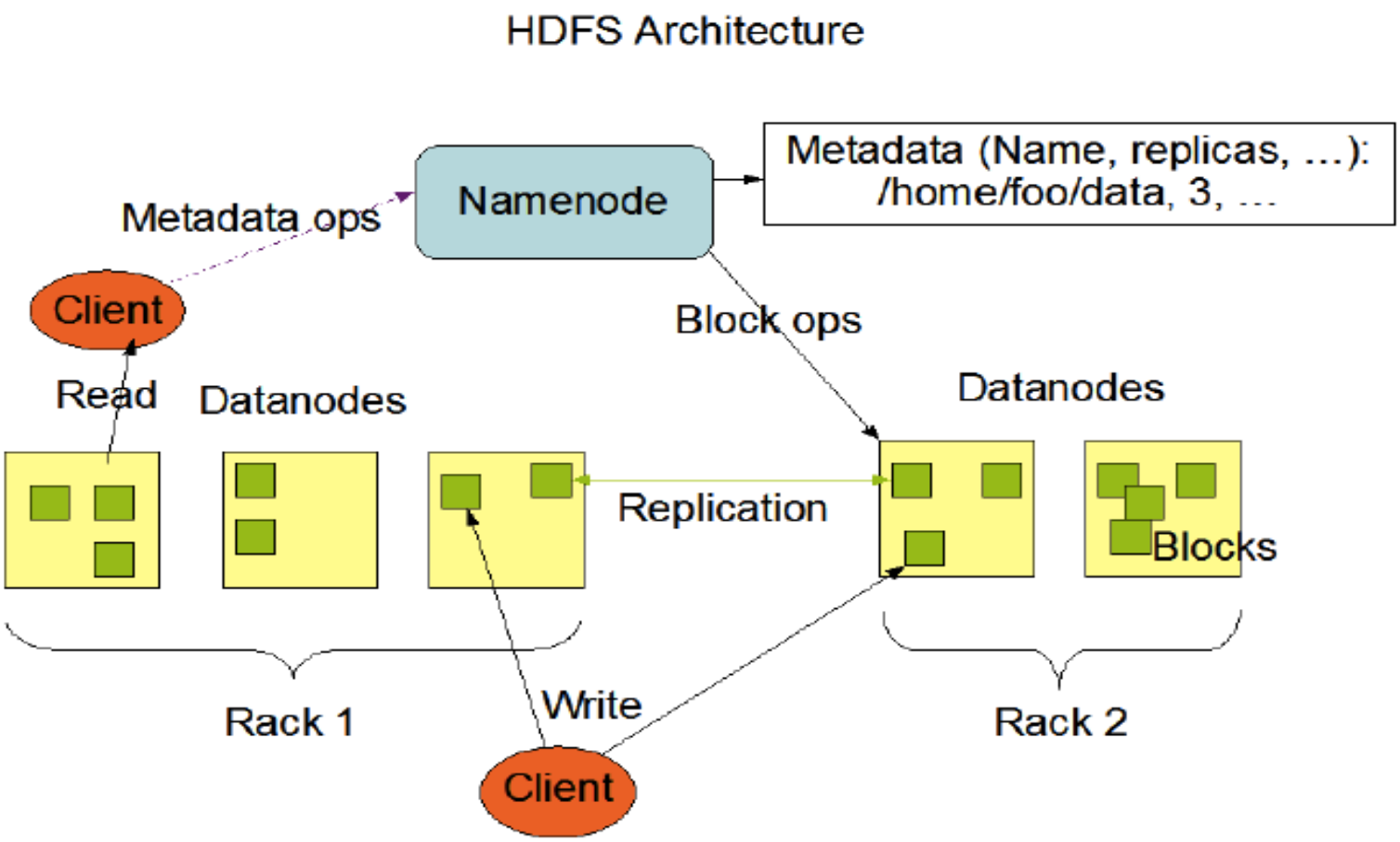
Thanachart Numnonda, thanachart@imcinstitute.com June 2016

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

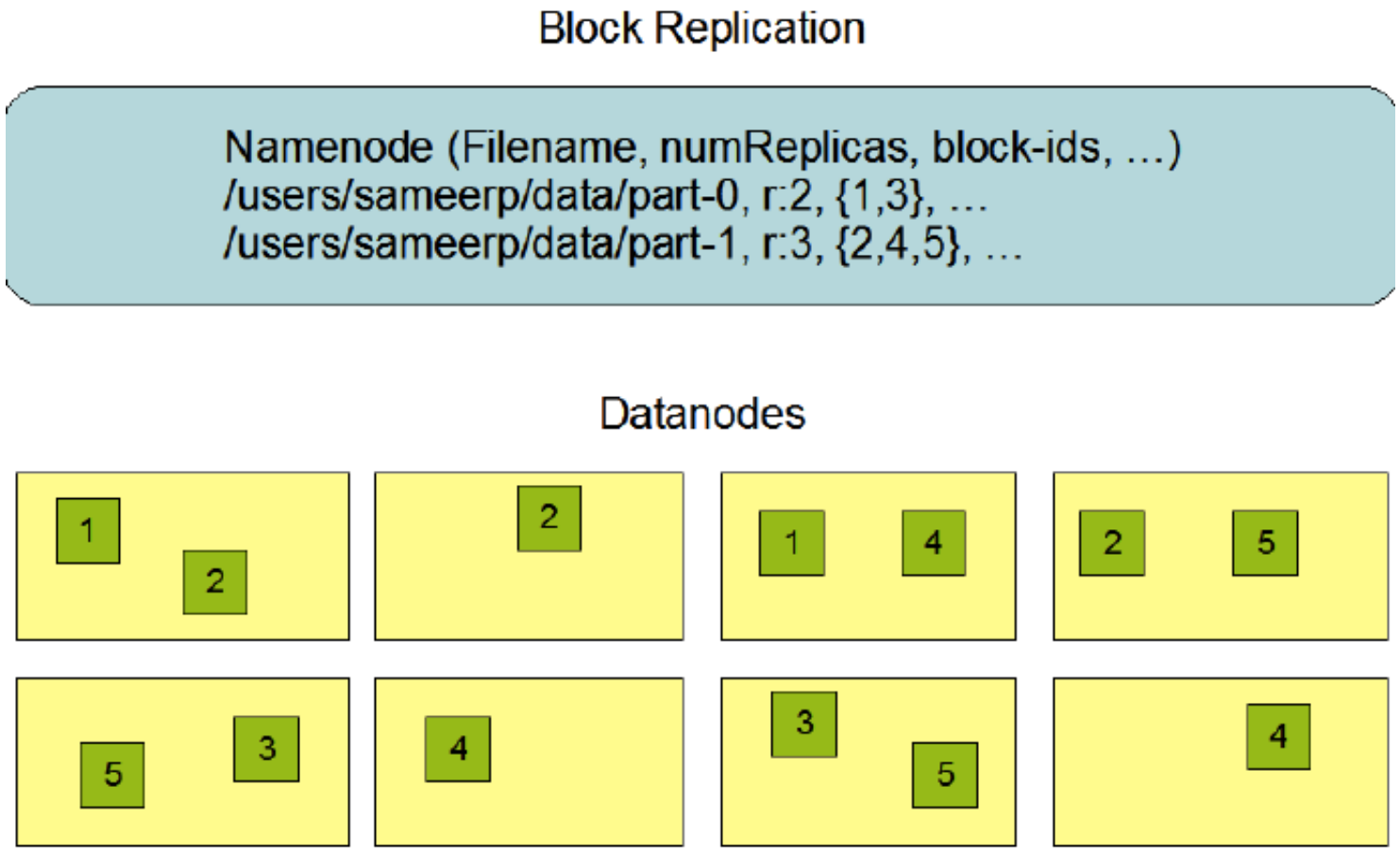
Hive.apache.org

HDFS Architecture



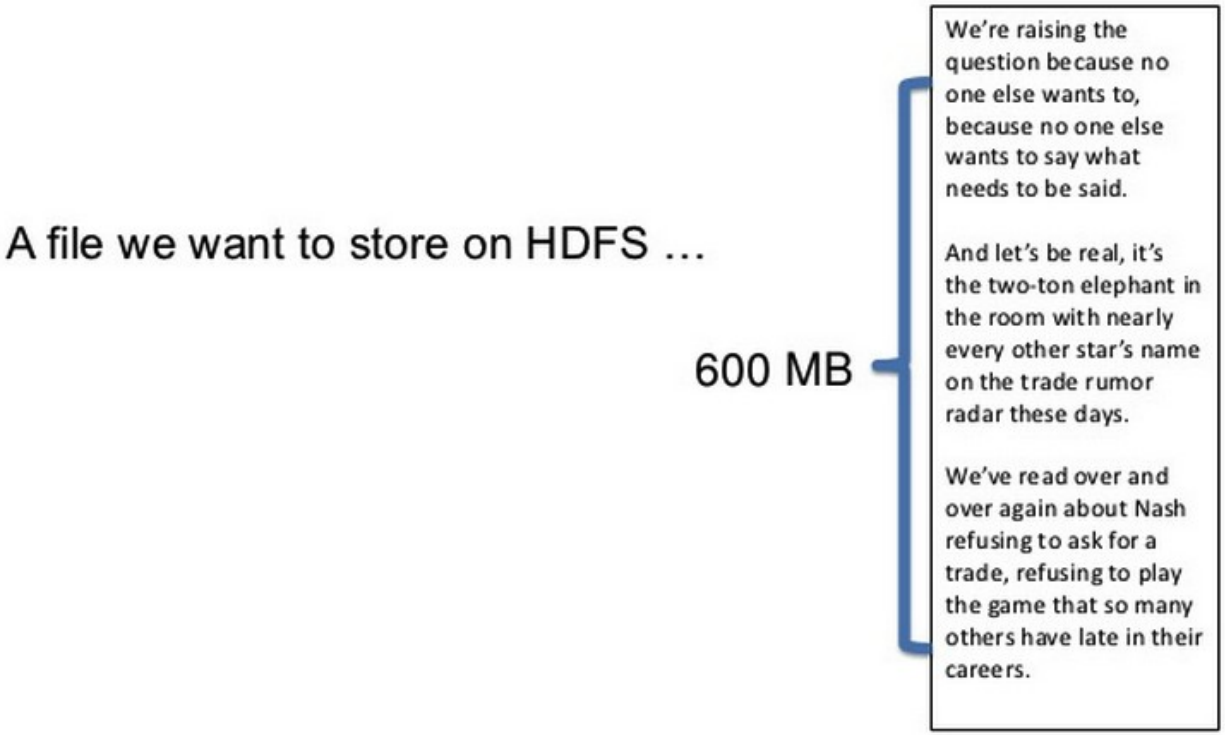
Source Hadoop: Shashwat Shriparv

Data Replication in HDFS



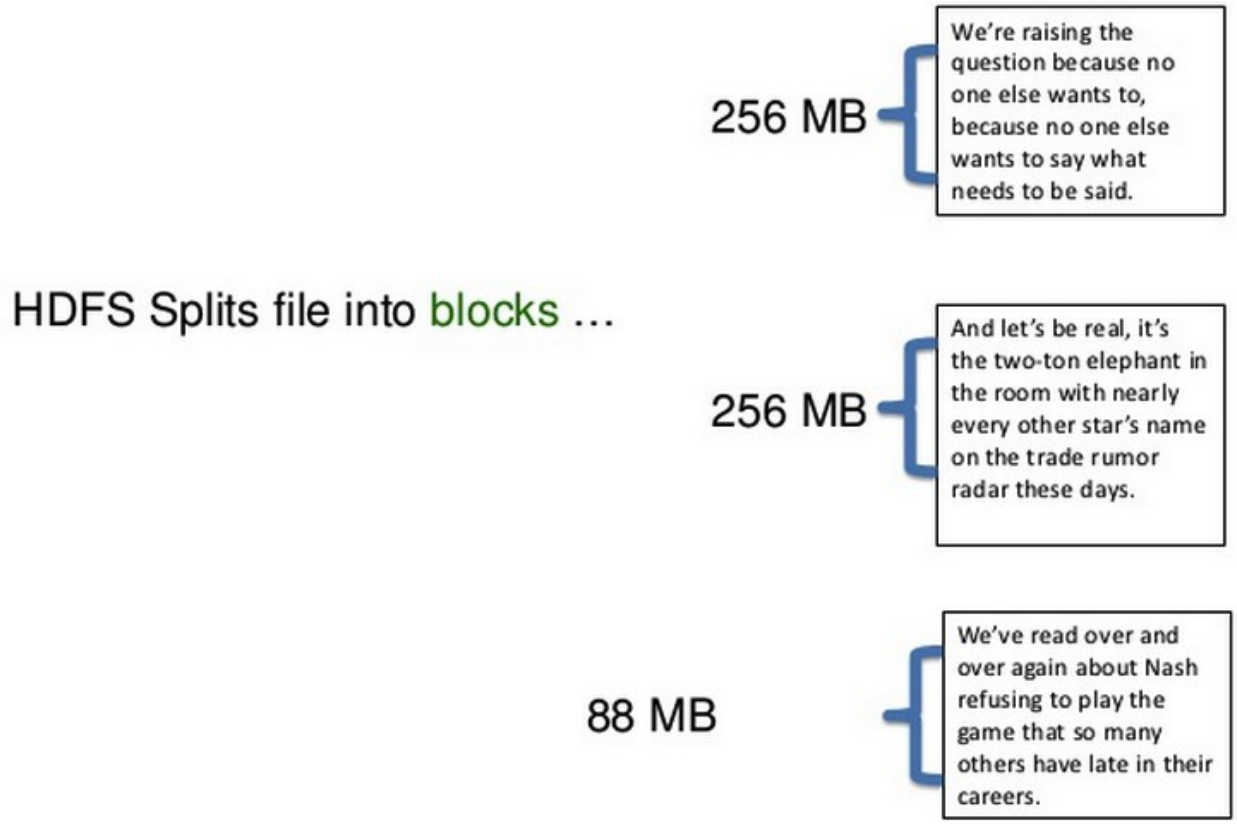
Source Hadoop: Shashwat Shriparv

How does HDFS work?



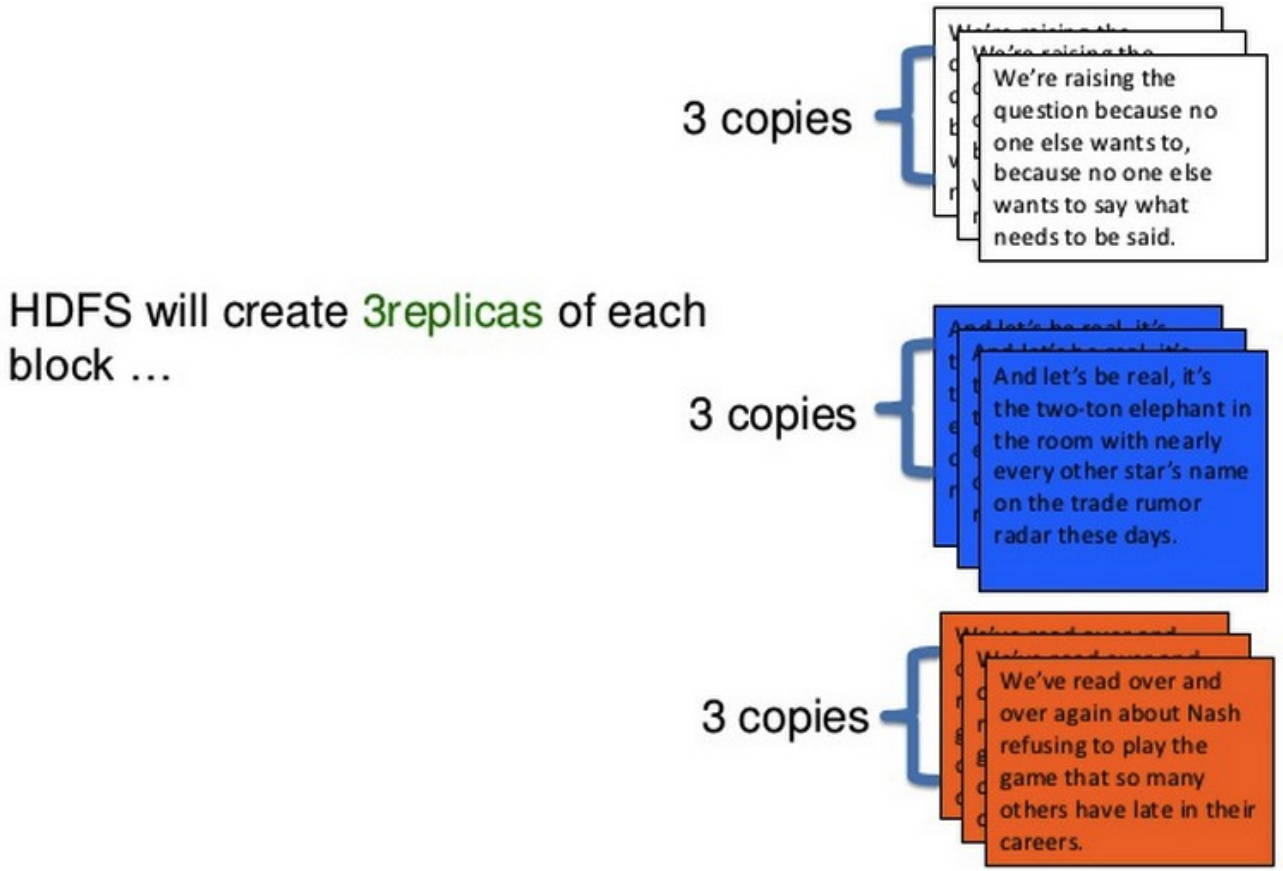
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?



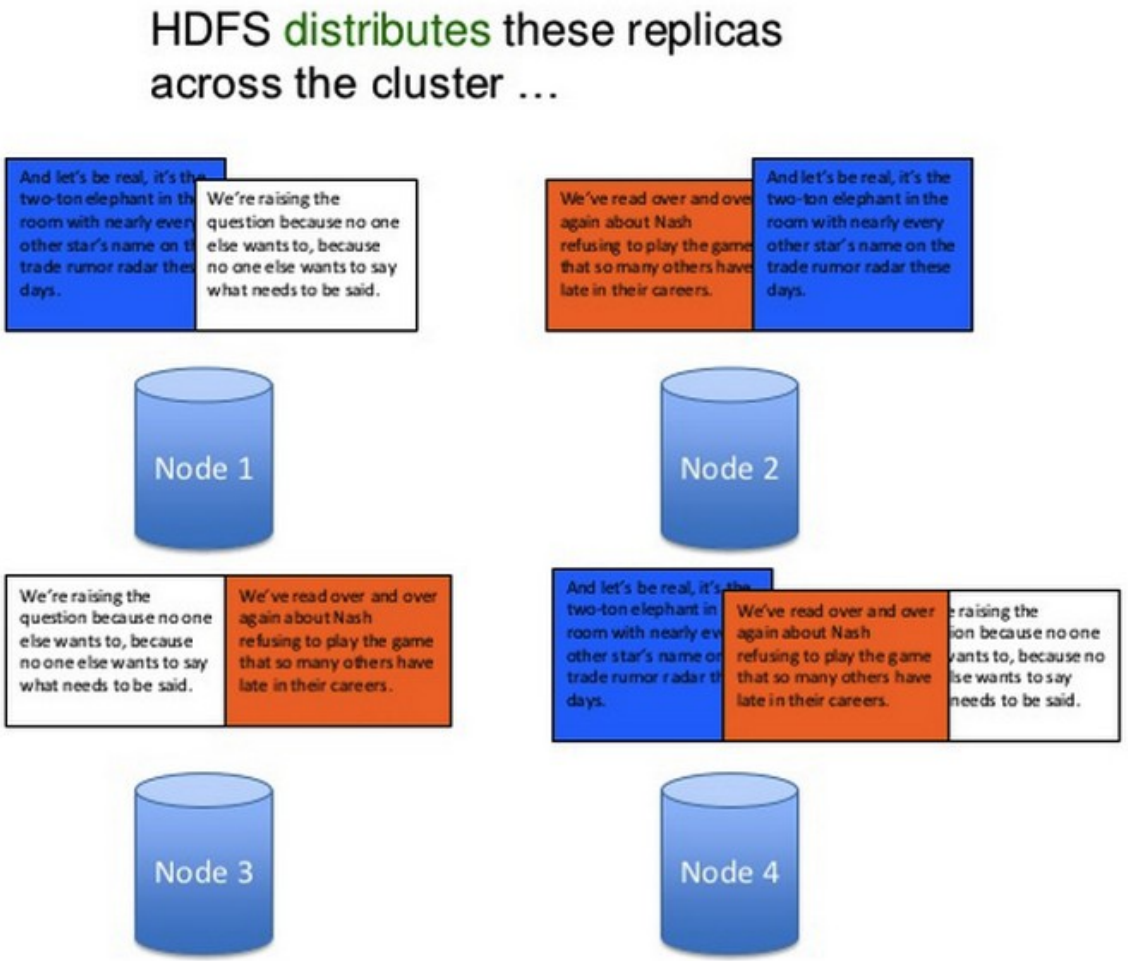
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?



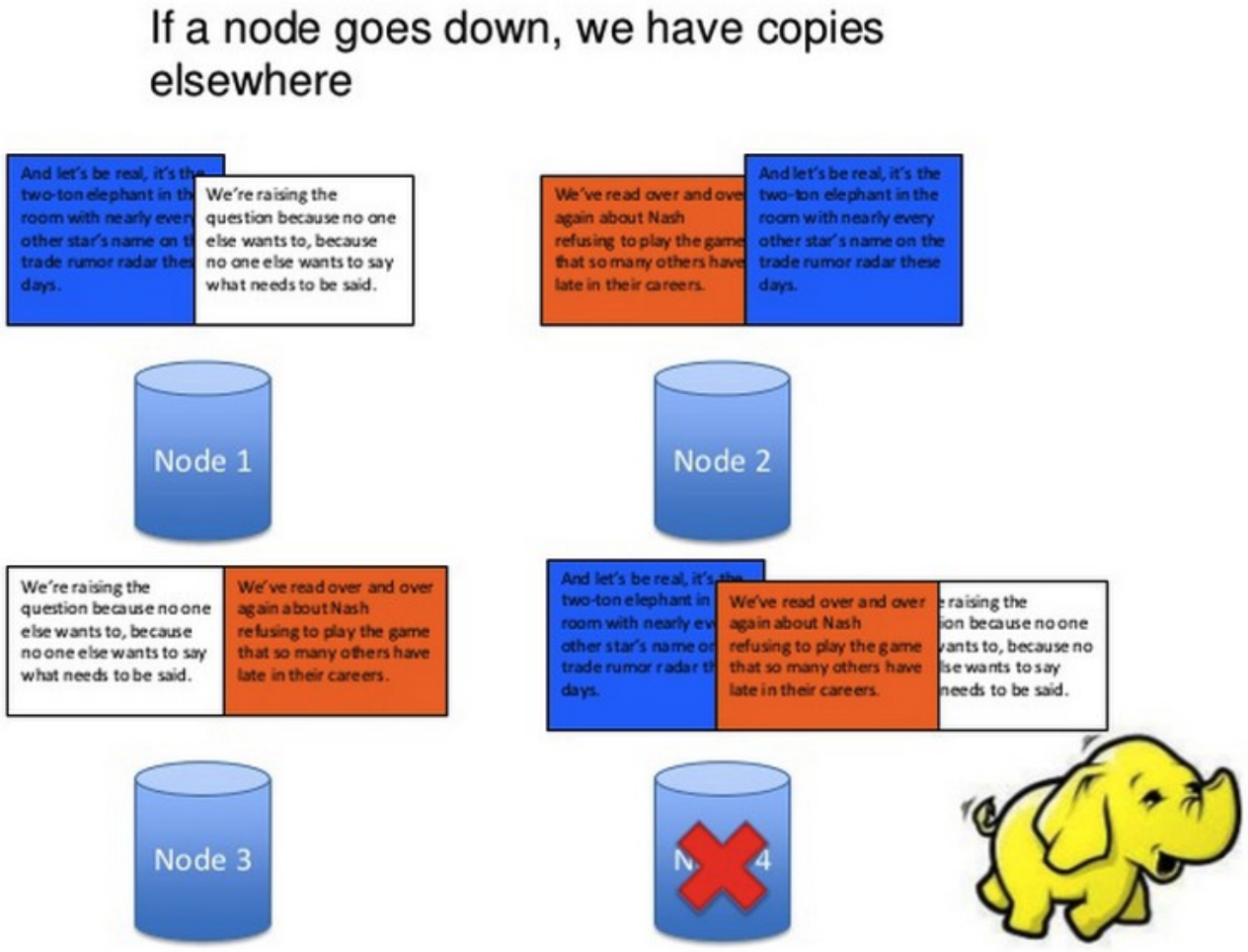
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?



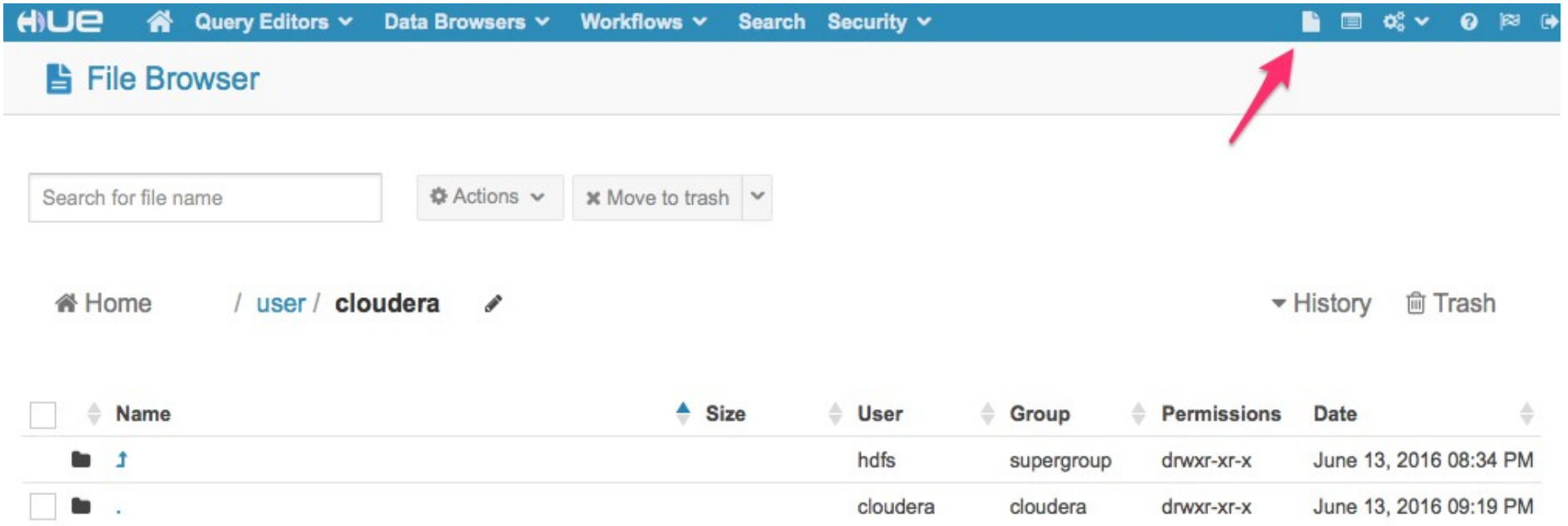
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

How does HDFS work?

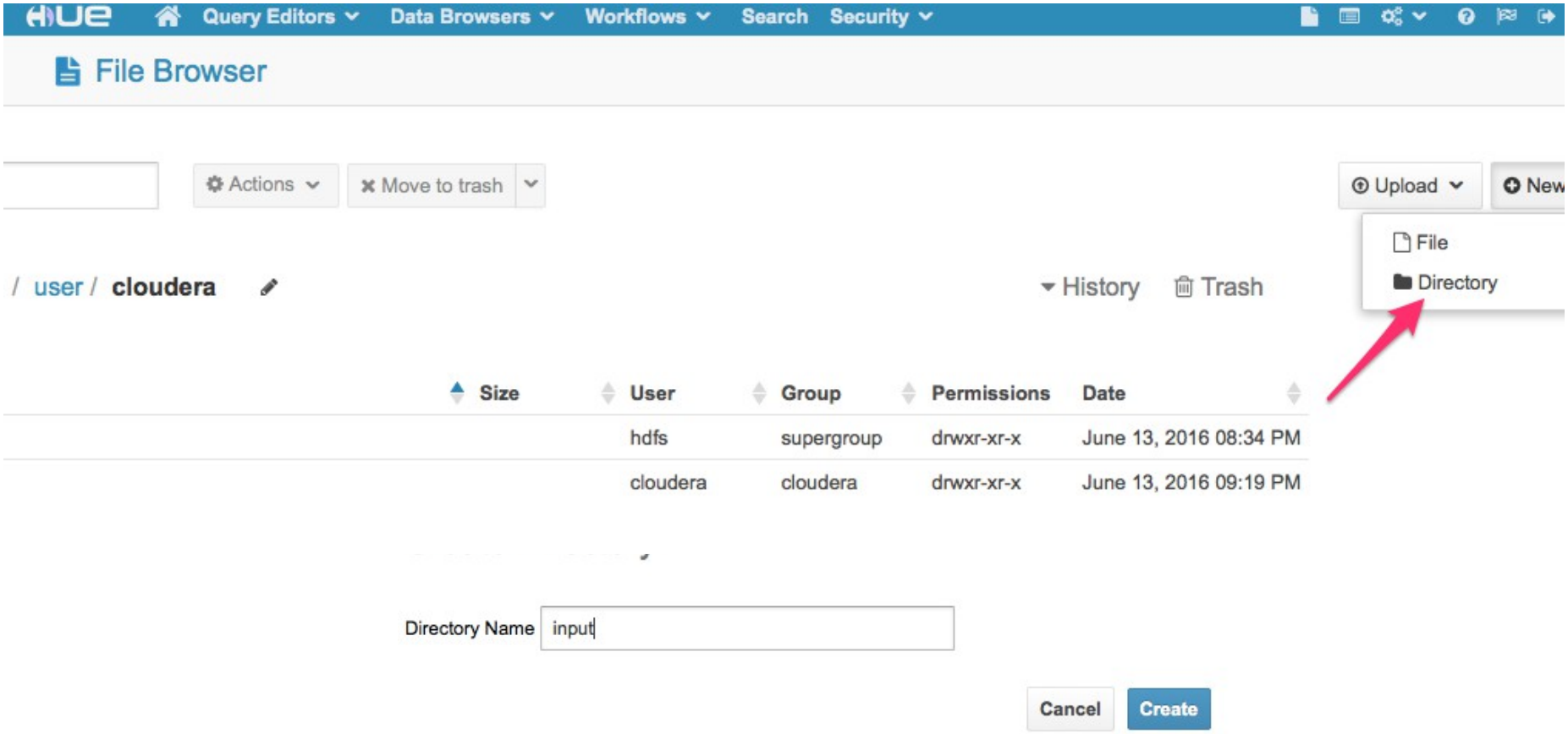


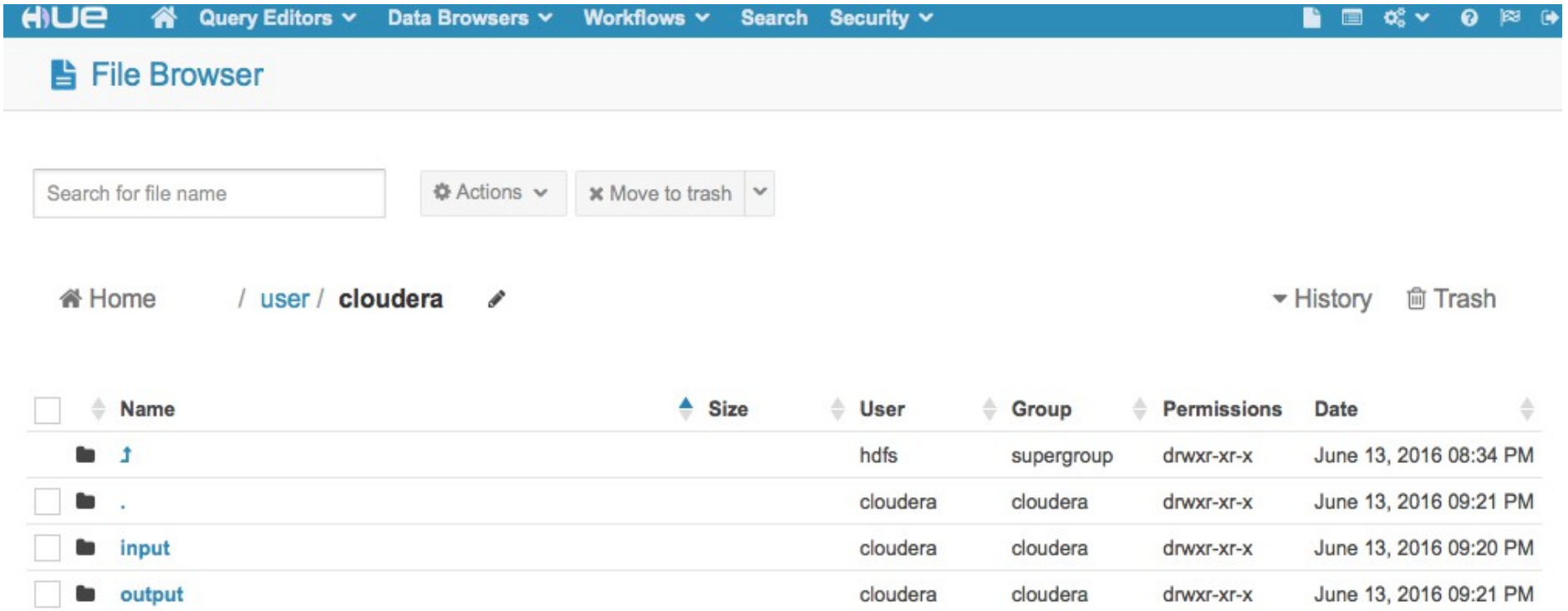
Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

Review file in Hadoop HDFS using File Browse

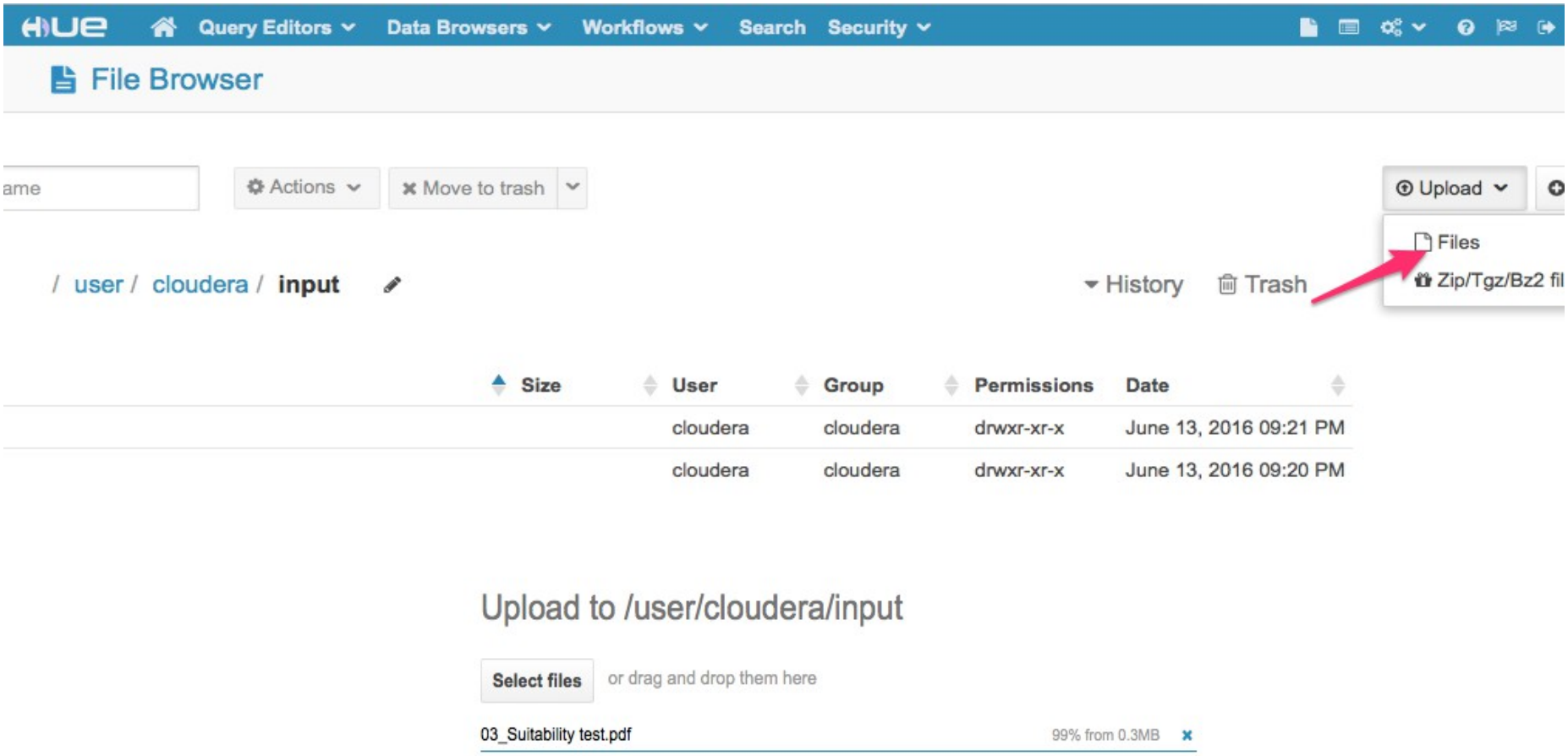


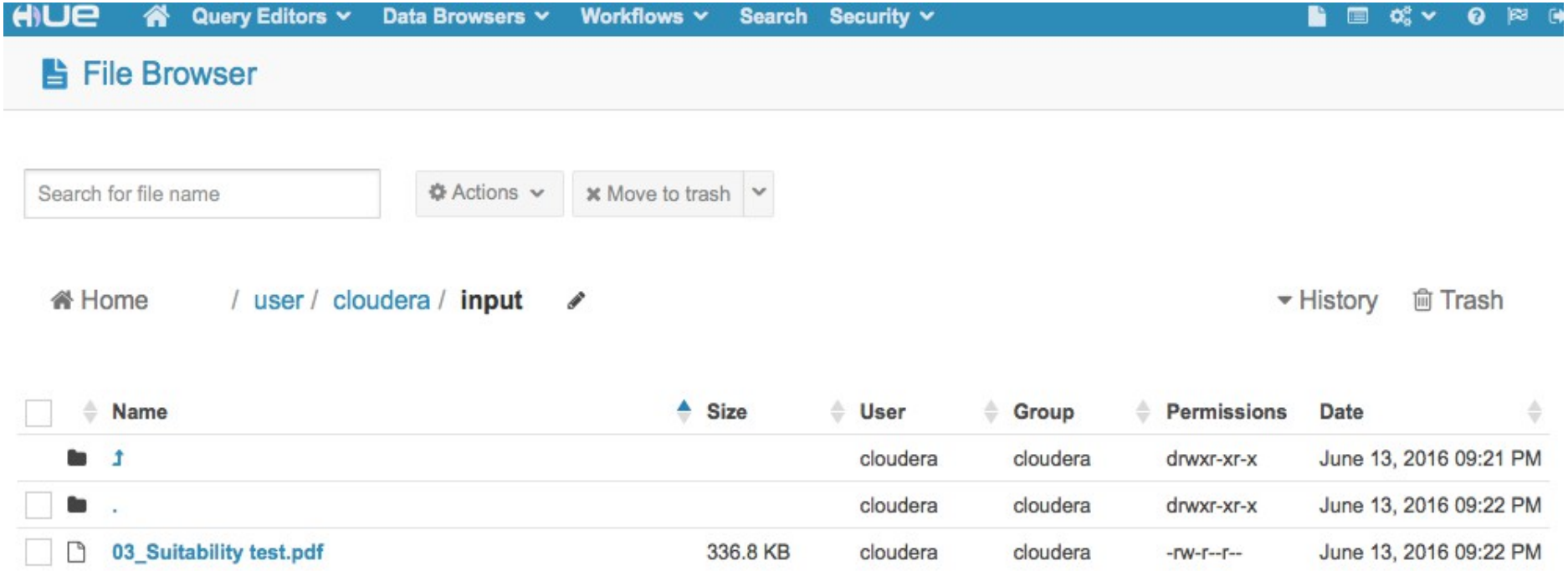
Create a new directory name as: input & output




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

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:~$
```

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
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

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

Note: you login as ubuntu, so you need to a sudo command to Switch user to hdfs

```
$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]#
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Lecture
Understanding Spark

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

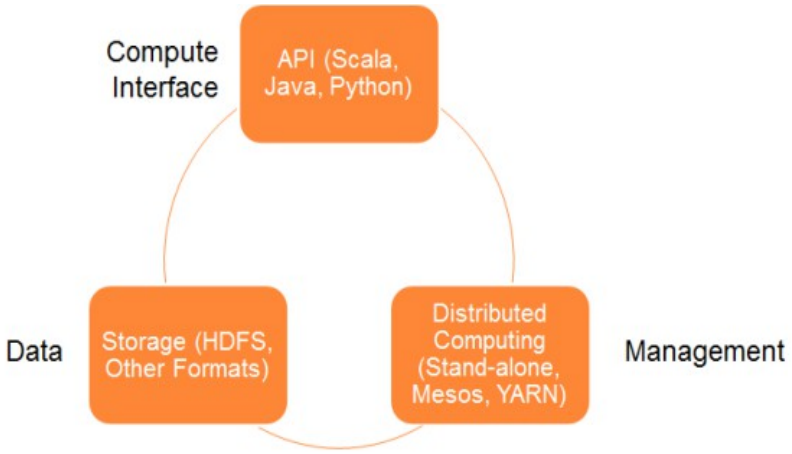
Introduction

A fast and general engine for large scale data processing

An open source big data processing framework built around speed, ease of use, and sophisticated analytics. Spark enables applications in Hadoop clusters to run up to 100 times faster in memory and 10 times faster even when running on disk.

What is Spark?

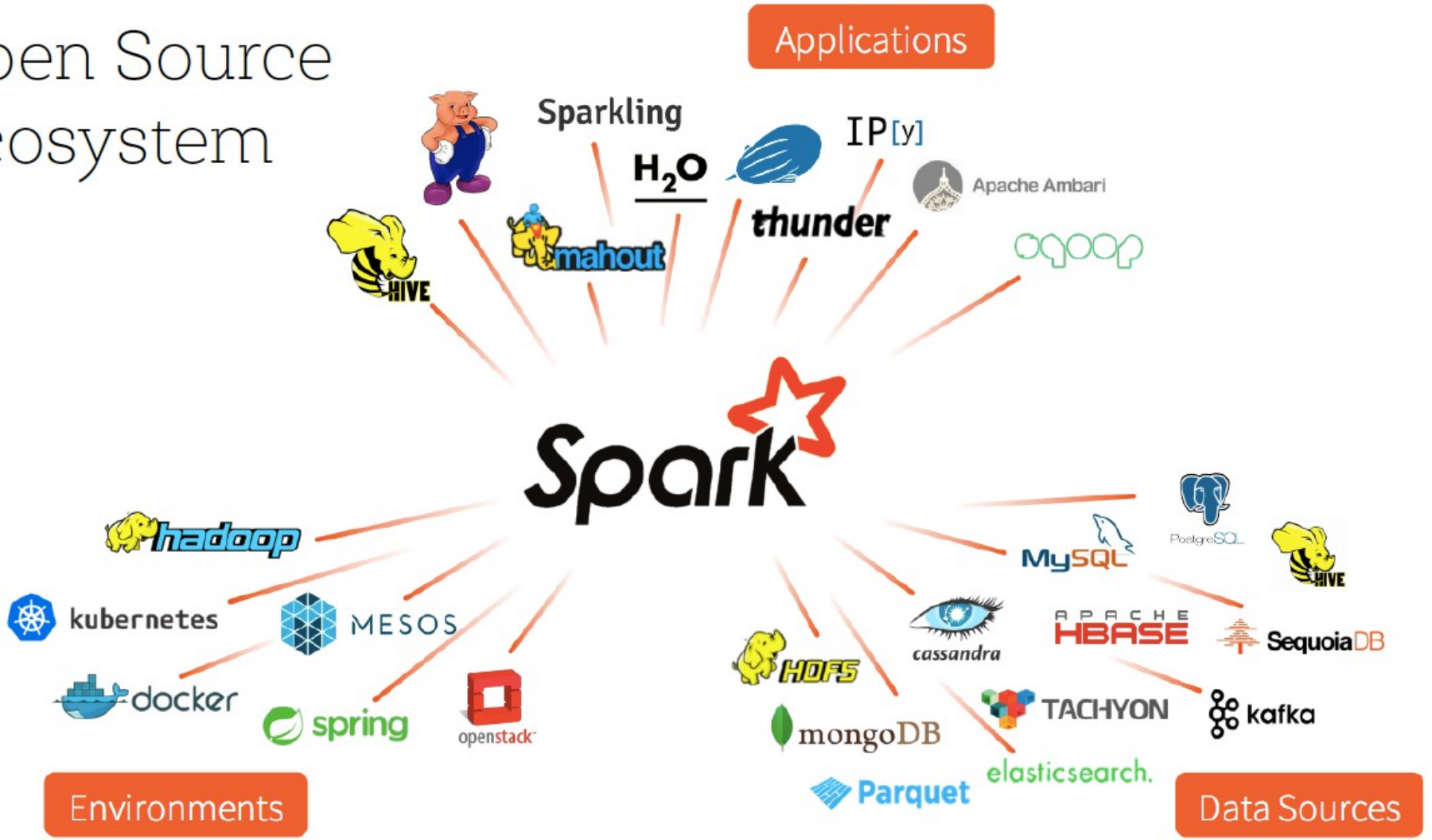
- Framework for distributed processing.
- In-memory, fault tolerant data structures
- Flexible APIs in Scala, Java, Python, SQL, R
- Open source



Why Spark?

- Handle Petabytes of data
- Significant faster than MapReduce
- Simple and intutive APIs
- General framework
  - Runs anywhere
  - Handles (most) any I/O
  - Interoperable libraries for specific use-cases

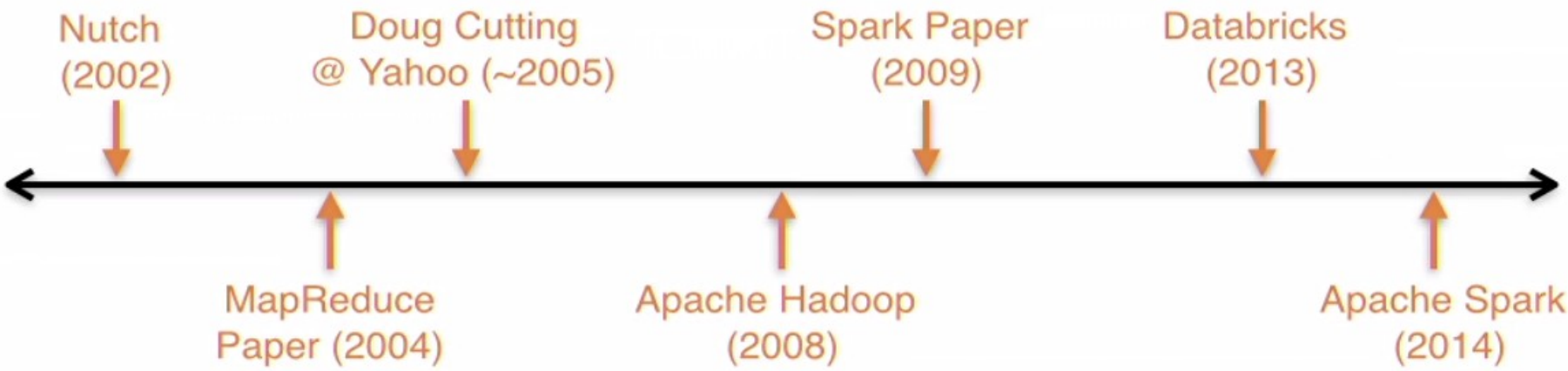
Open Source Ecosystem

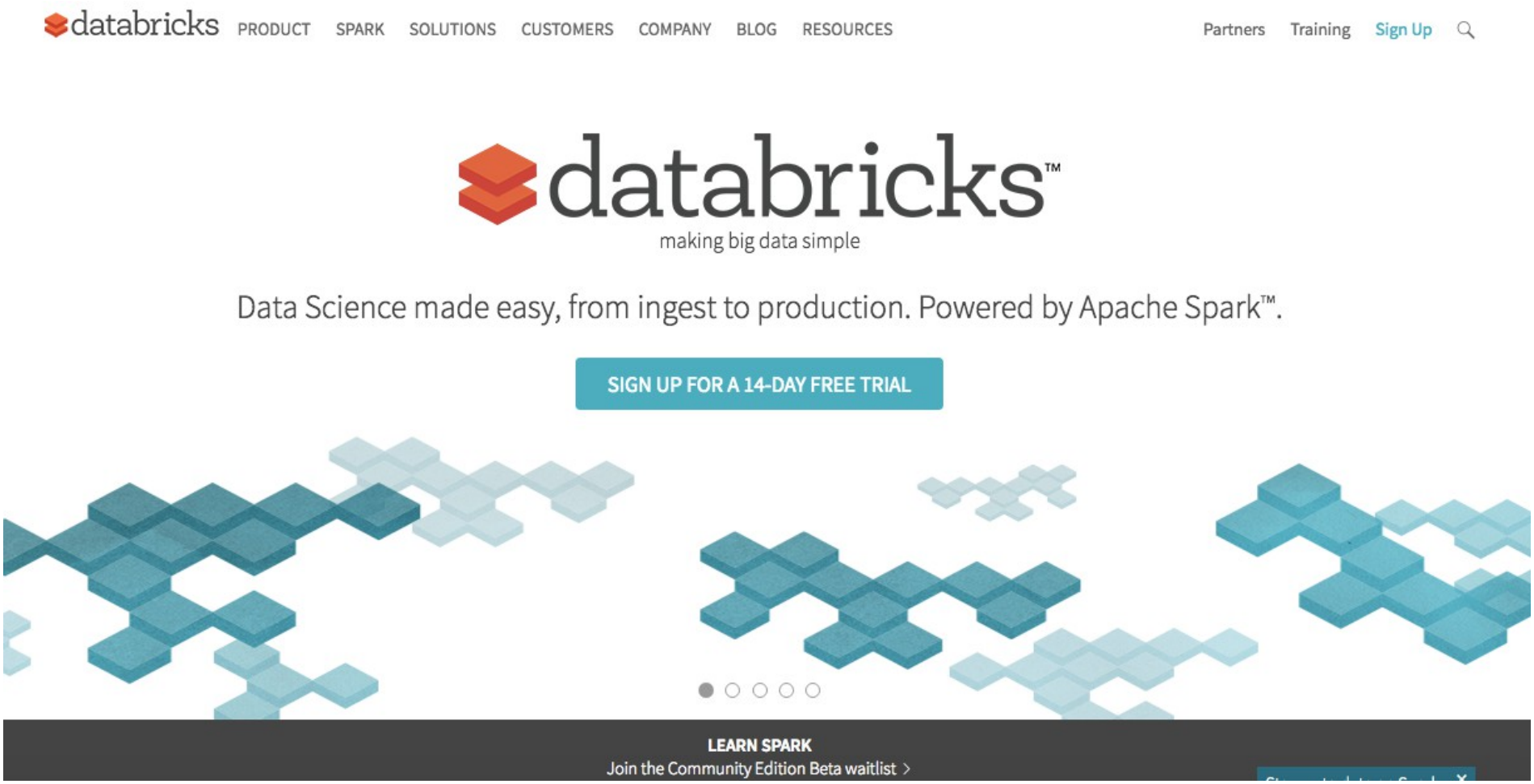


Source: Jump start into Apache Spark and Databricks

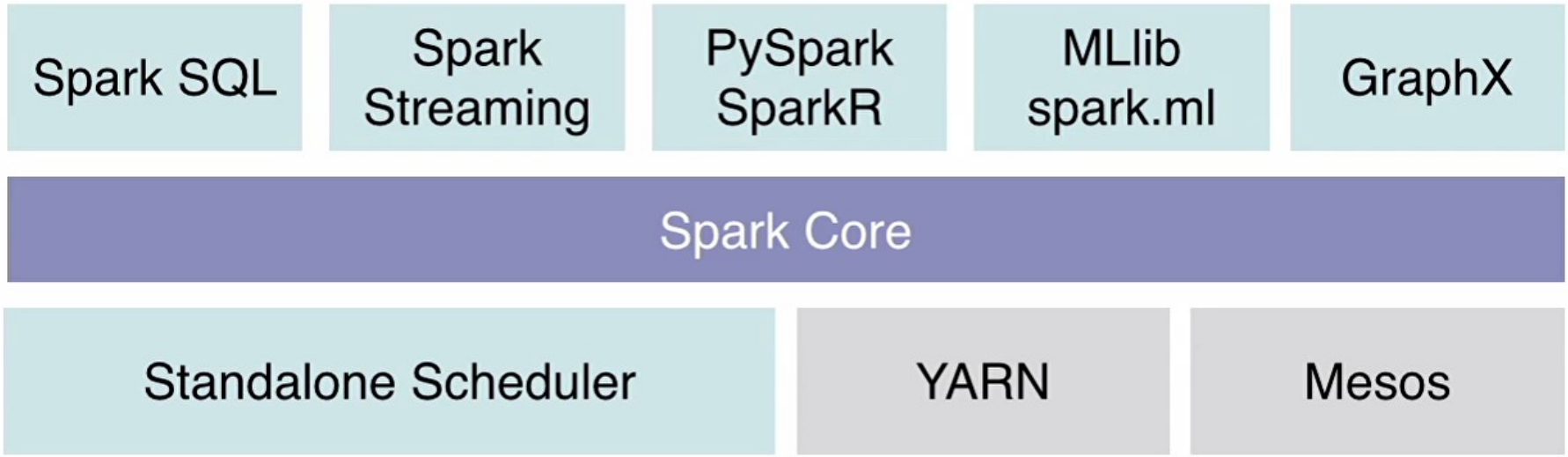
Spark: History

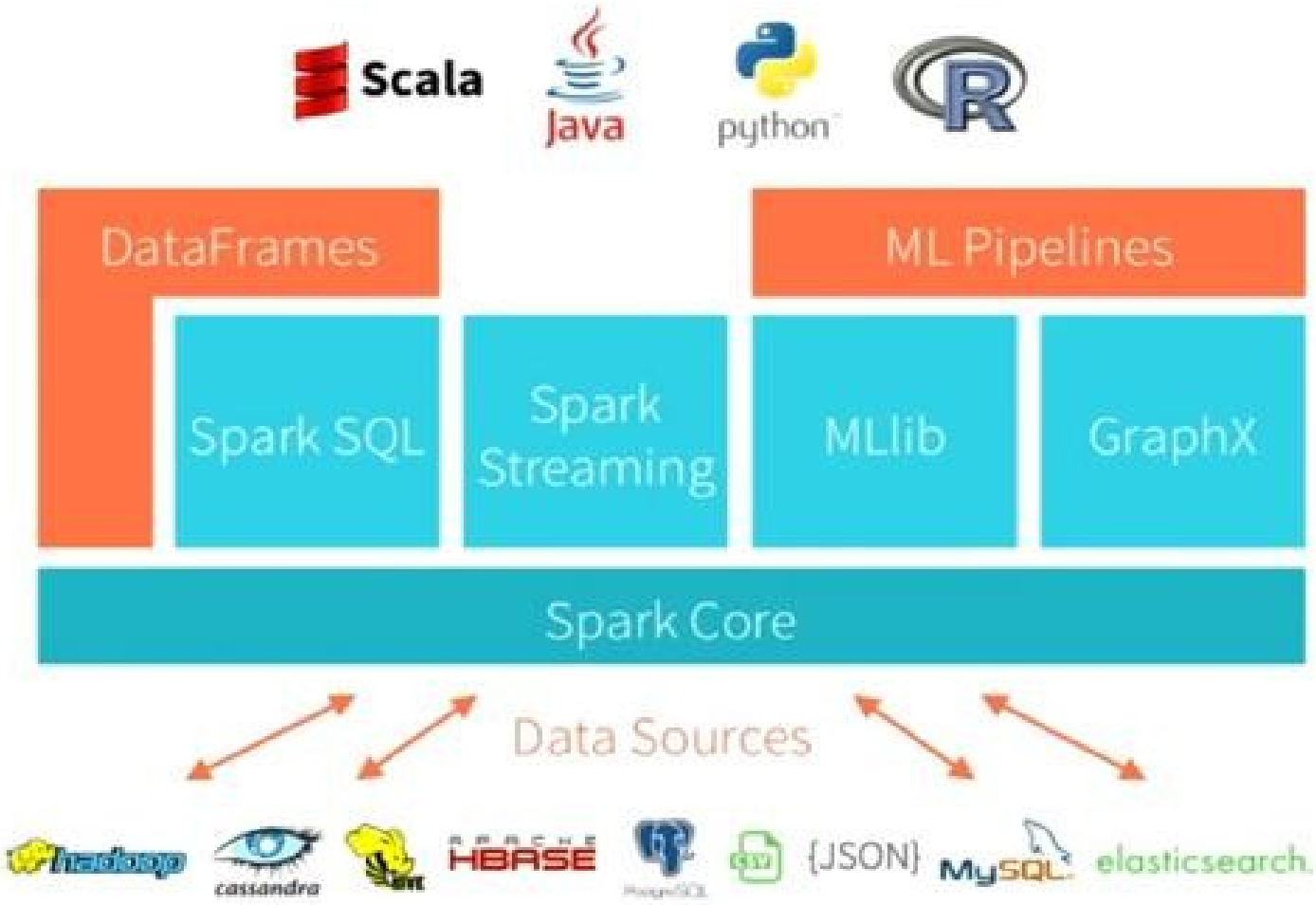
- Founded by AMPlab, UC Berkeley
- Created by Matei Zaharia (PhD Thesis)
- Maintained by Apache Software Foundation
- Commercial support by Databricks




databricks
PRODUCT
SPARK
SOLUTIONS
CUSTOMERS
COMPANY
BLOG
RESOURCES
Partners
Training
Sign Up
databricks™
making big data simple
Data Science made easy, from ingest to production. Powered by Apache Spark™.
SIGN UP FOR A 14-DAY FREE TRIAL
LEARN SPARK
Join the Community Edition Beta waitlist >

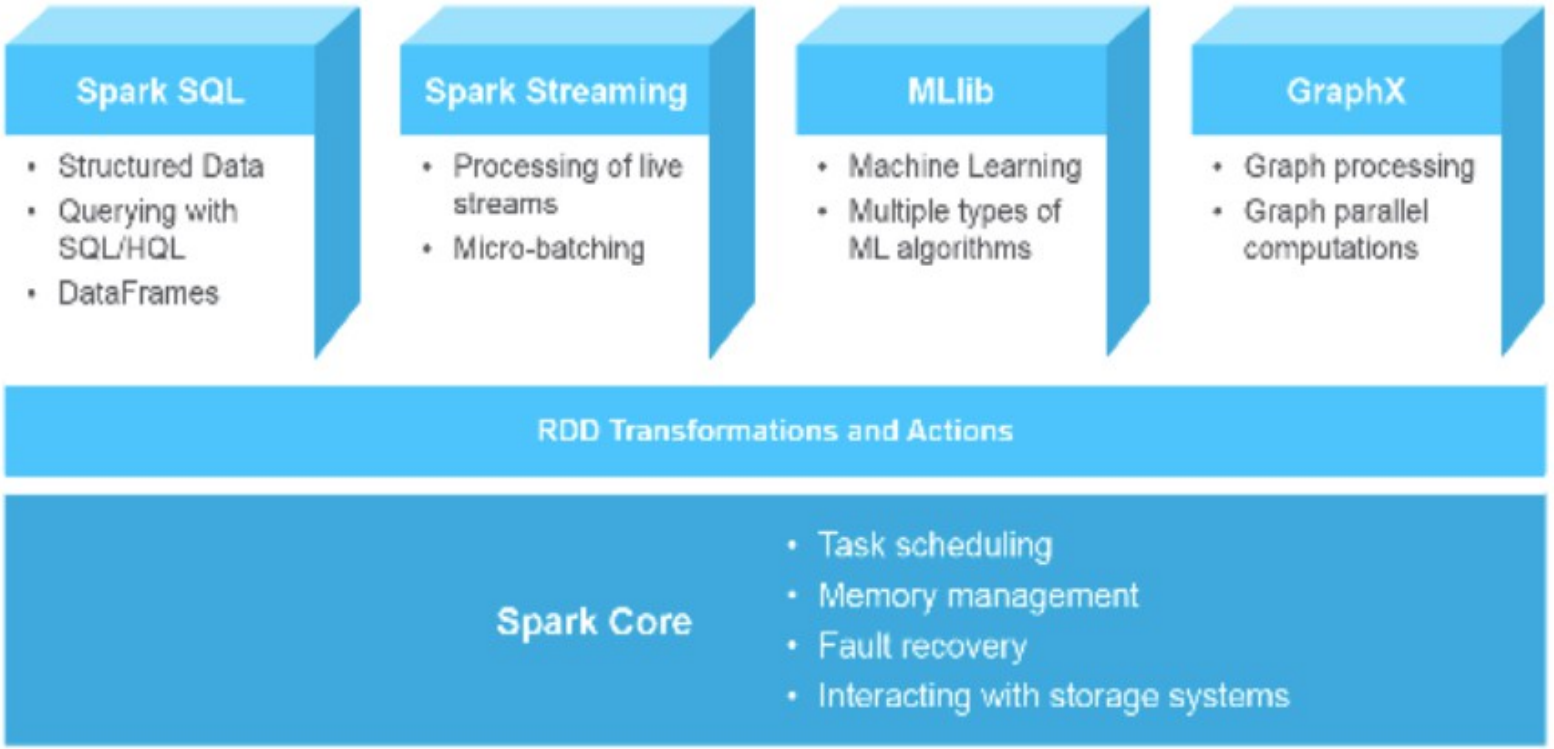
Spark Platform





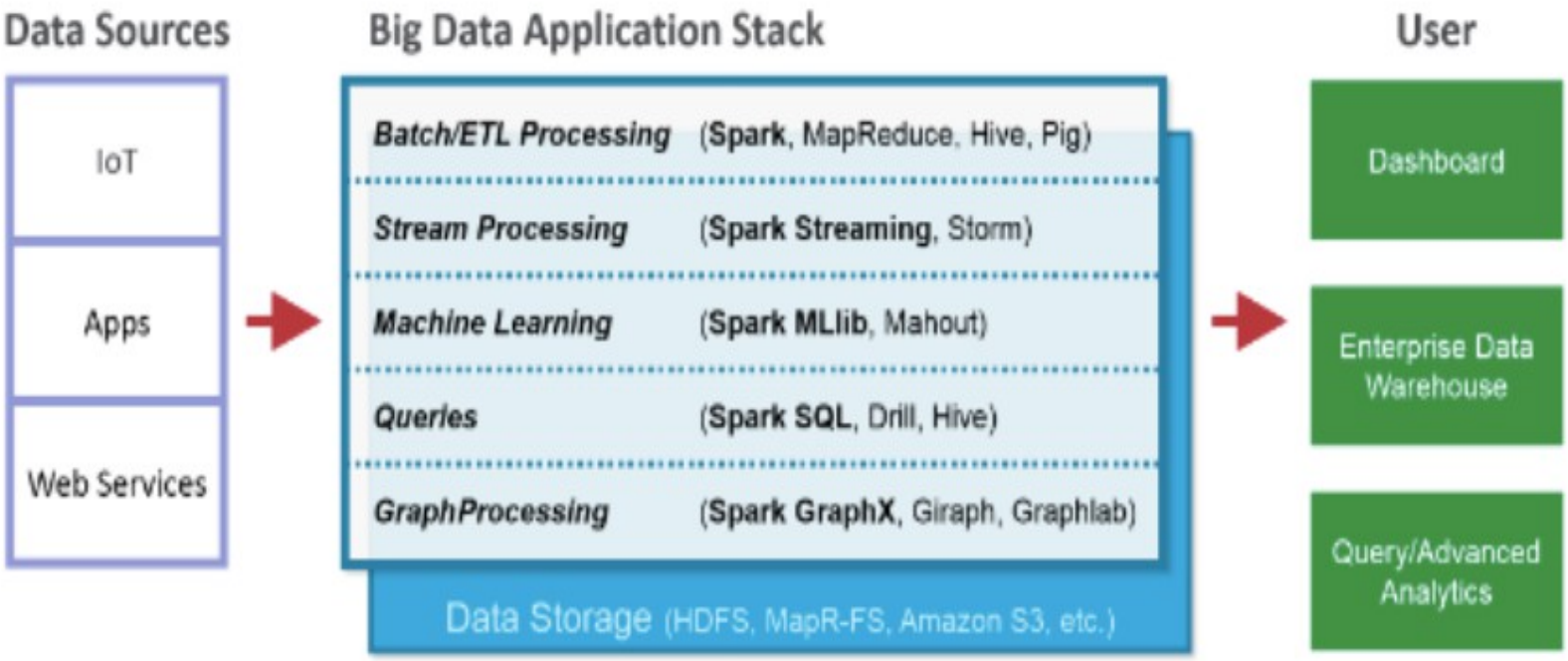
Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark Platform



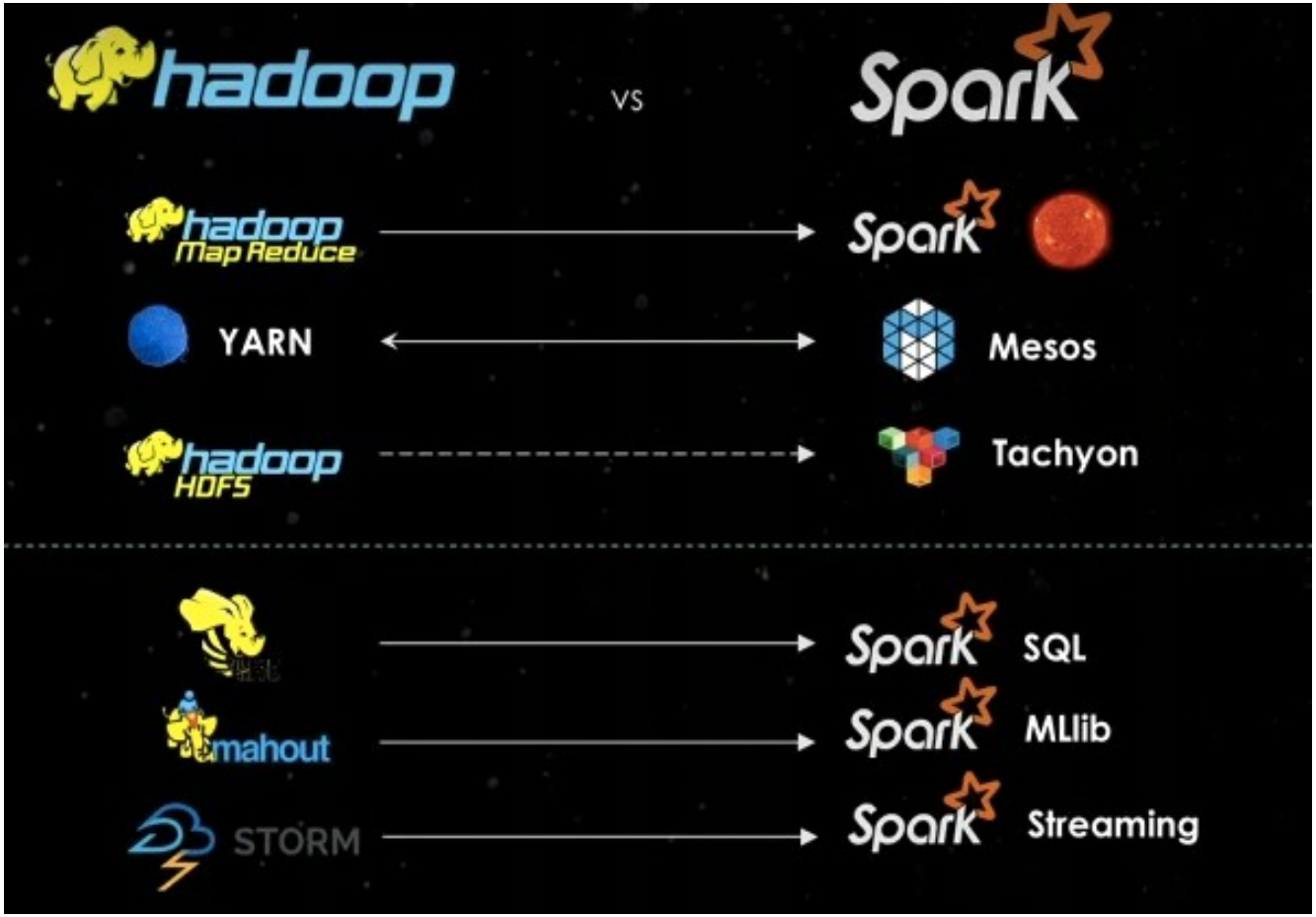
Source: MapR Academy

thanachart@imcinstitute.com



Source: MapR Academy

thanachart@imcinstitute.com



Source: TRAINING Intro to Apache Spark - Brian Clapper

thanachart@imcinstitute.com

Large-Scale Usage

Largest cluster
8000 Nodes (Tencent)

Largest single job
1 PB (Alibaba, Databricks)

Top Streaming Intake
1 TB/hour (HHMI Janelia Farm)

2014 On-Disk Sort Record
Fastest Open Source Engine for sorting a PB

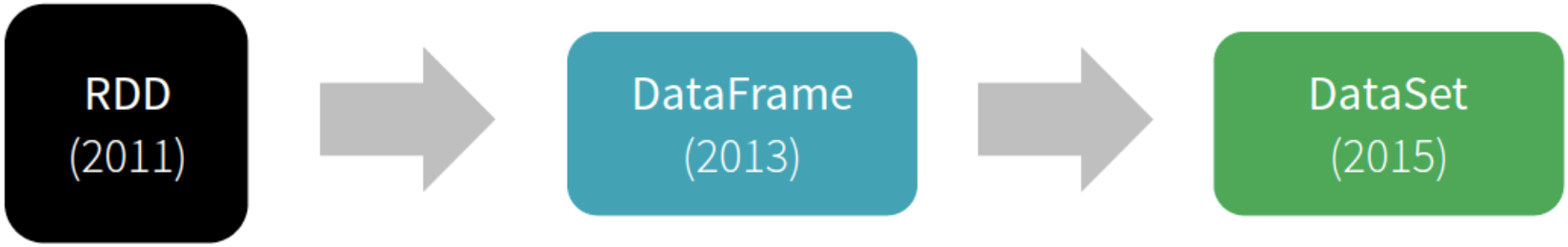
Source: Jump start into Apache Spark and Databricks

Notable Users

Companies That Presented at Spark Summit 2015 in San Francisco

Source: Jump start into Apache Spark and Databricks

History of Spark APIs



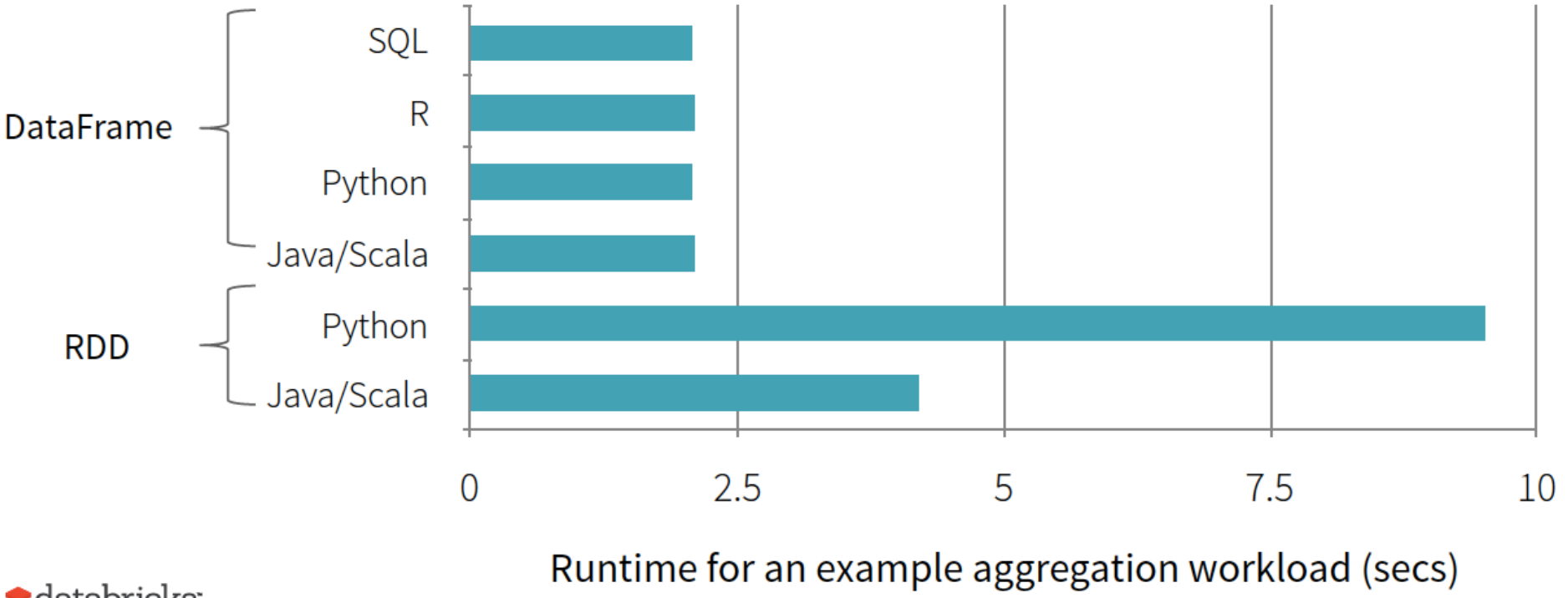
- Distribute collection of JVM objects
- Functional Operators (map, filter, etc.)

- Distribute collection of Row objects
- Expression-based operations and UDFs
- Logical plans and optimizer
- Fast/efficient internal representations

- Internally rows, externally JVM objects
- "Best of both worlds" **type safe + fast**

Source: Jump start into Apache Spark and Databricks

Benefit of Logical Plan: Performance Parity Across Languages



Source: Jump start into Apache Spark and Databricks

What is a RDD?

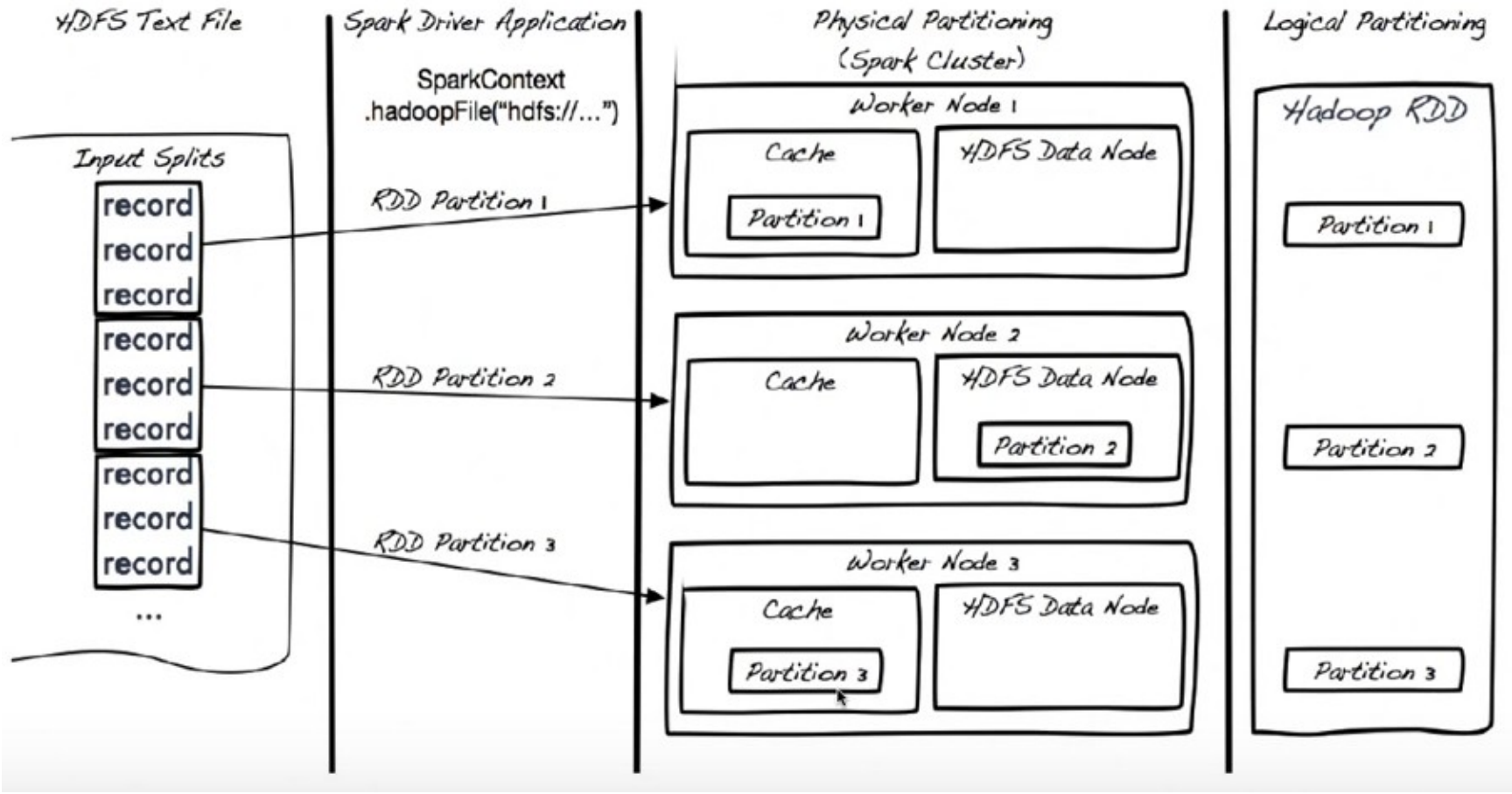
- Resilient: if the data in memory (or on a node) is lost, it can be recreated.
- Distributed: data is chucked into partitions and stored in memory acress the custer.
- Dataset: initial data can come from a table or be created programmatically

RDD:

- Fault tollerant
- Immutable
- Three methods for creating RDD:
  - Parallelizing an existing correction
  - Referencing a dataset
  - Transformation from an existing RDD
- Types of files supported:
  - Text files
  - SequenceFiles
  - Hadoop InputFormat

RDD Creation

```
hdfsData = sc.textFile("hdfs://data.txt")
```

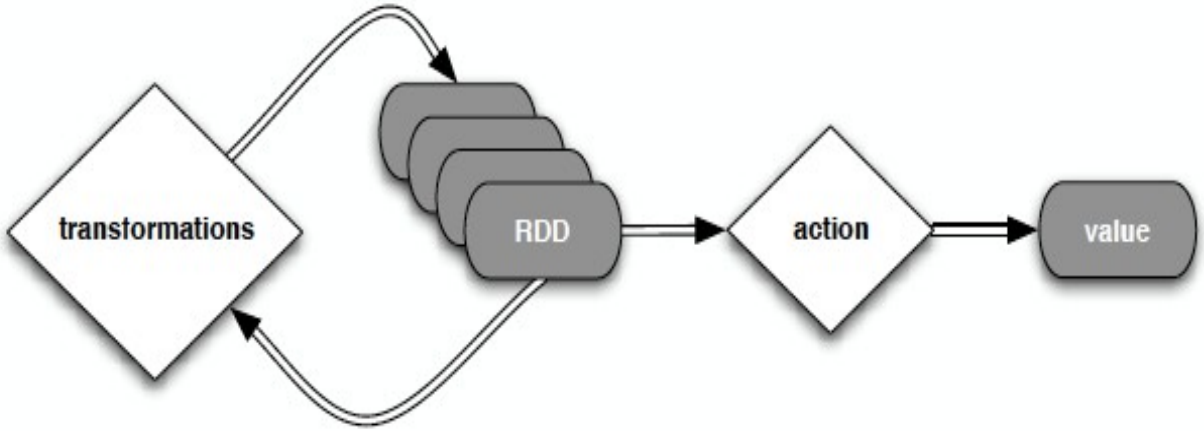


Source: Pspark: A brain-friendly introduction

RDD: Operations

- Transformations: transformations are lazy (not computed immediately)
- Actions: the transformed RDD gets recomputed

  when an action is run on it (default)



Direct Acyclic Graph (DAG)

- View the DAG
  linesLength.toDebugString
- Sample DAG

```
res5: String =
MappedRDD[4] at map at <console>:16 (3 partitions)
  MappedRDD[3] at map at <console>:16 (3 partitions)
    FilteredRDD[2] at filter at <console>:14 (3 partitions)
      MappedRDD[1] at textFile at <console>:12 (3 partitions)
        HadoopRDD[0] at textFile at <console>:12 (3 partitions)
```

Functions Deconstructed

```
import random
flips = 1000000

# lazy eval
coins = xrange(flips)

# lazy eval, nothing executed
heads = sc.parallelize(coins) \
          .map(lambda i: random.random()) \
          .filter(lambda r: r < 0.51) \
          .count()
```

Python Generator

Create RDD

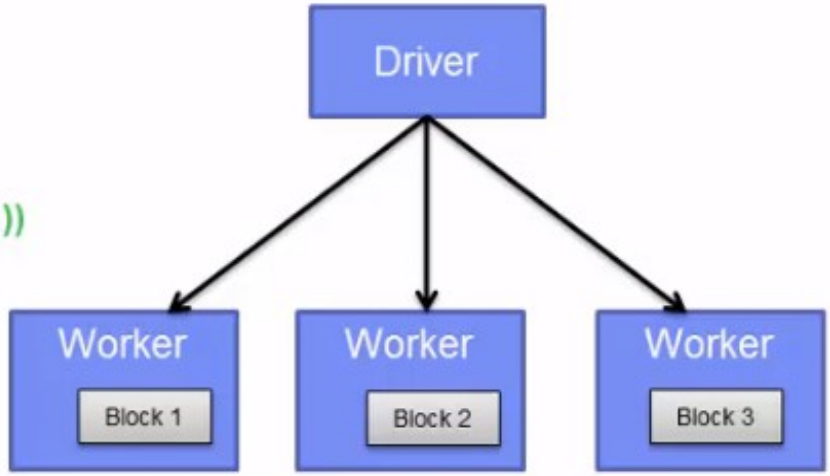
Transformations

Action (materialize result)

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

What happens when an action is executed

```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Cache
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```

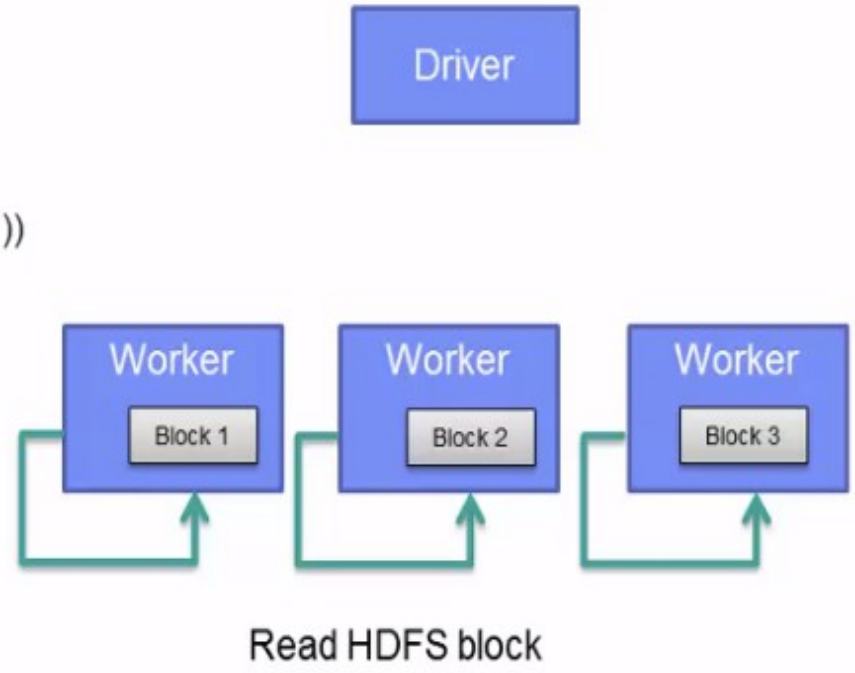


Driver sends the code to be executed on each block

Source: Spark Fundamentals I Big Data Usibersity

What happens when an action is executed

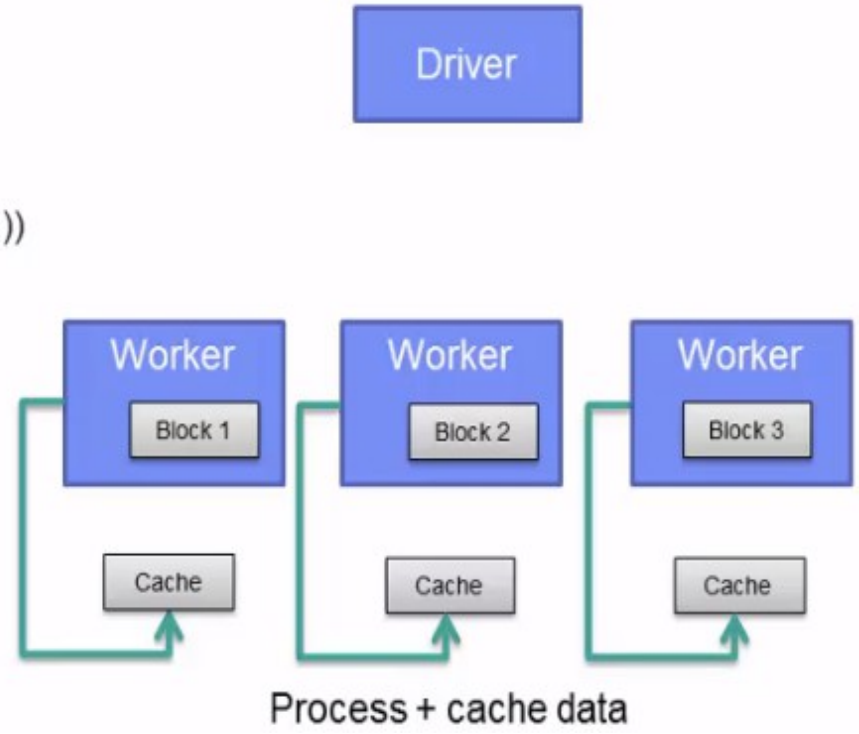
```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Caching
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```



Source: Spark Fundamentals I Big Data Usibersity

What happens when an action is executed

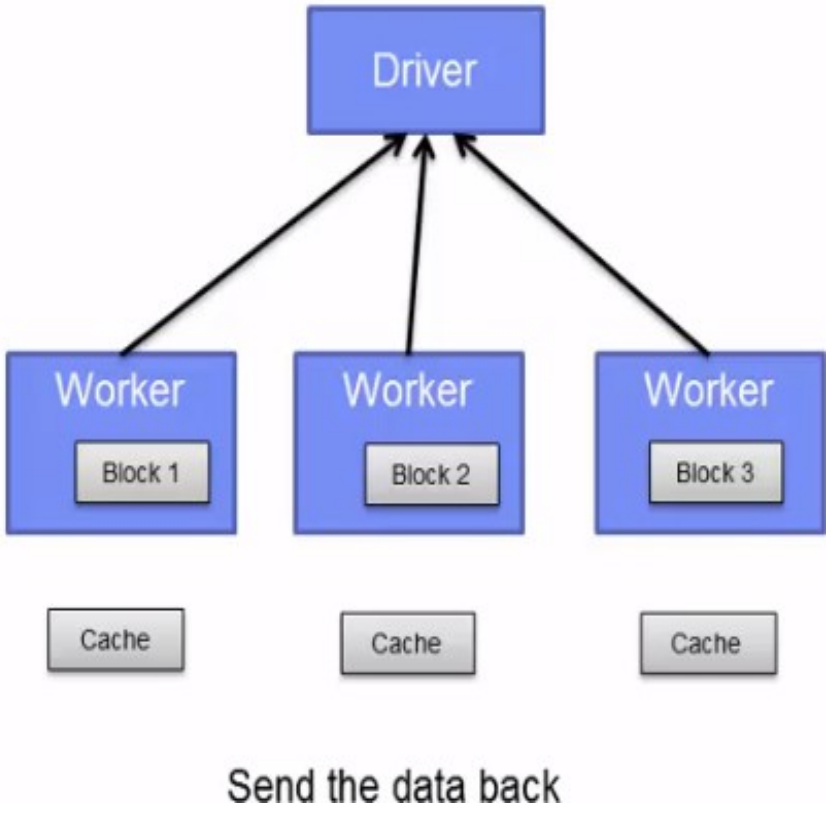
```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Caching
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```



Source: Spark Fundamentals I Big Data Usibersity

What happens when an action is executed

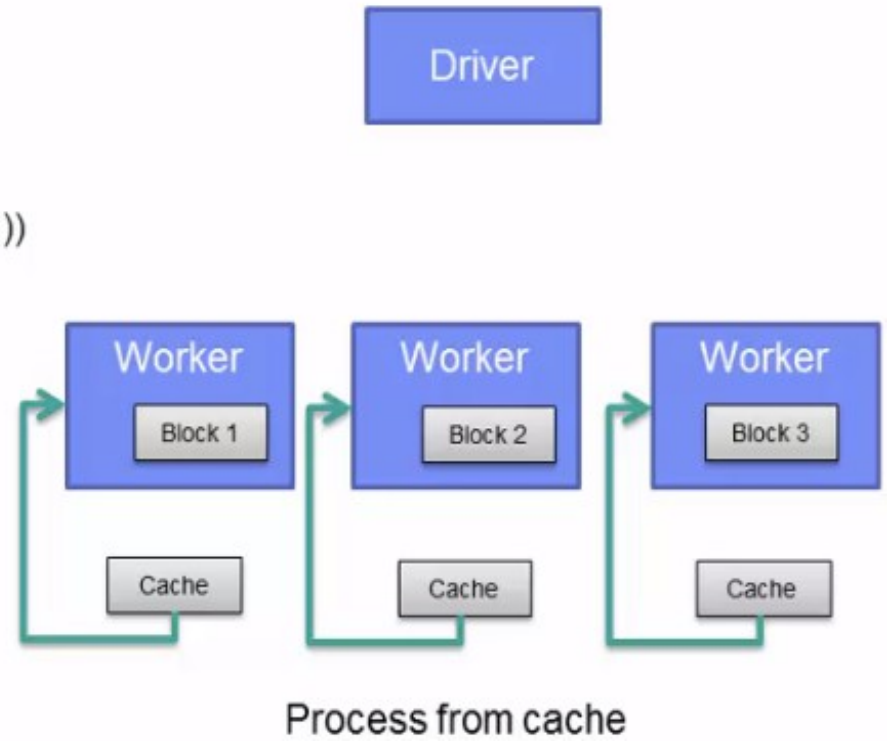
```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Caching
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```



Source: Spark Fundamentals I Big Data Usibersity

What happens when an action is executed

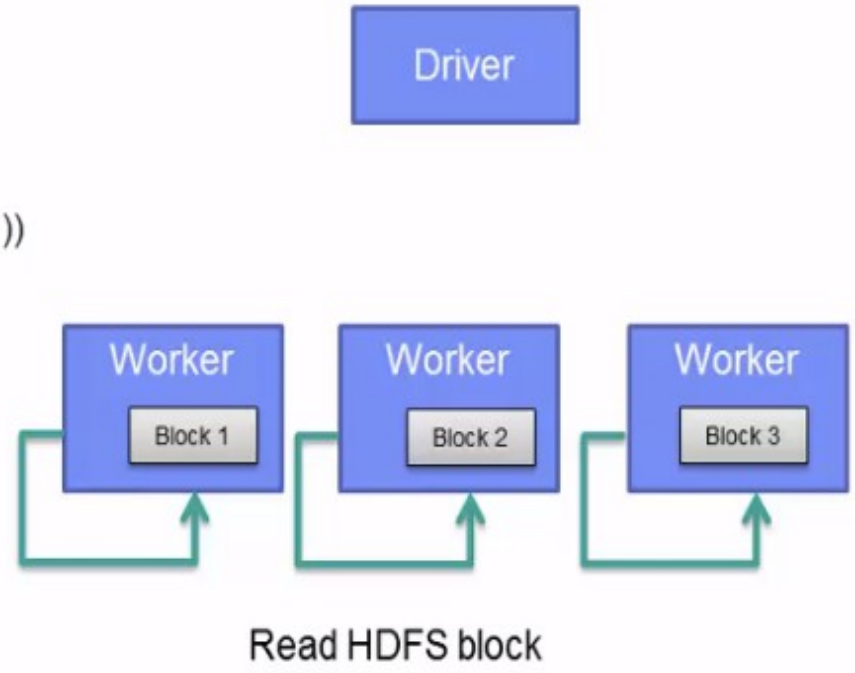
```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Caching
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```



Source: Spark Fundamentals I Big Data Usibersity

What happens when an action is executed

```
// Creating the RDD
val logFile = sc.textFile("hdfs://...")
// Transformations
val errors = logFile.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
//Caching
messages.cache()
// Actions
messages.filter(_.contains("mysql")).count()
messages.filter(_.contains("php")).count()
```



Spark:Transformation

transformation	*description*
map(*func*)	return a new distributed dataset formed by passing each element of the source through a function *func*
filter(*func*)	return a new dataset formed by selecting those elements of the source on which *func* returns true
flatMap(*func*)	similar to map, but each input item can be mapped to 0 or more output items (so *func* should return a Seq rather than a single item)
sample(*withReplacement*, *fraction*, *seed*)	sample a fraction *fraction* of the data, with or without replacement, using a given random number generator *seed*
union(*otherDataset*)	return a new dataset that contains the union of the elements in the source dataset and the argument
distinct([*numTasks*]))	return a new dataset that contains the distinct elements of the source dataset

Spark:Transformation

transformation	description
groupByKey([*numTasks*])	when called on a dataset of (K, V) pairs, returns a dataset of (K, Seq[V]) pairs
reduceByKey(*func*, [*numTasks*])	when called on a dataset of (K, V) pairs, returns a dataset of (K, V) pairs where the values for each key are aggregated using the given reduce function
sortByKey([*ascending*], [*numTasks*])	when called on a dataset of (K, V) pairs where K implements Ordered, returns a dataset of (K, V) pairs sorted by keys in ascending or descending order, as specified in the boolean ascending argument
join(*otherDataset*, [*numTasks*])	when called on datasets of type (K, V) and (K, W), returns a dataset of (K, (V, W)) pairs with all pairs of elements for each key
cogroup(*otherDataset*, [*numTasks*])	when called on datasets of type (K, V) and (K, W), returns a dataset of (K, Seq[V], Seq[W]) tuples – also called groupWith
cartesian(*otherDataset*)	when called on datasets of types T and U, returns a dataset of (T, U) pairs (all pairs of elements)

Single RDD Transformation

filter females to analyze female buying patterns

```
male1, male2, female1 -> female1
```

map squared values

```
2, 5, 6 -> 4, 25, 36
```

flatMap to break up a sentence into words

```
my name is ray -> my, name, is, ray
```

find the distinct values in a dataset

```
apple, apple, banana -> apple, banana
```

sample two values at random

```
apple, banana, guava -> banana, apple
```

Source: Jspark: A brain-friendly introduction

Multiple RDD Transformation

union

```
apple, orange, banana, guava,
banana, pear
```

intersection

```
banana
```

subtract anything shown in Dataset B from Dataset A

```
apple, orange
```

cartesian (every possible pair combo)

```
(apple, guava), (apple, banana), ...
```

Dataset A

apple
orange
banana

Dataset B

guava
banana
pear

Source: Jspark: A brain-friendly introduction

Pair RDD Transformation

- reduceByKey
- groupByKey
- combineByKey
- mapValues
- flatMapValues
- keys
- values
- subtractByKey
- join
- rightOuterJoin
- leftOuterJoin
- cogroup
- sortByKey

Source: Jspark: A brain-friendly introduction

Spark:Actions

action	description
reduce(*func*)	aggregate the elements of the dataset using a function *func* (which takes two arguments and returns one), and should also be commutative and associative so that it can be computed correctly in parallel
collect()	return all the elements of the dataset as an array at the driver program – usually useful after a filter or other operation that returns a sufficiently small subset of the data
count()	return the number of elements in the dataset
first()	return the first element of the dataset – similar to *take(1)*
take(*n*)	return an array with the first *n* elements of the dataset – currently not executed in parallel, instead the driver program computes all the elements
takeSample(*withReplacement, fraction, seed*)	return an array with a random sample of *num* elements of the dataset, with or without replacement, using the given random number generator seed

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark:Actions

action	description
saveAsTextFile(*path*)	write the elements of the dataset as a text file (or set of text files) in a given directory in the local filesystem, HDFS or any other Hadoop-supported file system. Spark will call `toString` on each element to convert it to a line of text in the file
saveAsSequenceFile(*path*)	write the elements of the dataset as a Hadoop `SequenceFile` in a given path in the local filesystem, HDFS or any other Hadoop-supported file system. Only available on RDDs of key-value pairs that either implement Hadoop's `Writable` interface or are implicitly convertible to `Writable` (Spark includes conversions for basic types like `Int`, `Double`, `String`, etc).
countByKey()	only available on RDDs of type `(K, V)`. Returns a `Map` of `(K, Int)` pairs with the count of each key
foreach(*func*)	run a function *func* on each element of the dataset – usually done for side effects such as updating an accumulator variable or interacting with external storage systems

Spark: Persistence

transformation	*description*
`MEMORY_ONLY`	Store RDD as deserialized Java objects in the JVM. If the RDD does not fit in memory, some partitions will not be cached and will be recomputed on the fly each time they're needed. This is the default level.
`MEMORY_AND_DISK`	Store RDD as deserialized Java objects in the JVM. If the RDD does not fit in memory, store the partitions that don't fit on disk, and read them from there when they're needed.
`MEMORY_ONLY_SER`	Store RDD as serialized Java objects (one byte array per partition). This is generally more space-efficient than deserialized objects, especially when using a fast serializer, but more CPU-intensive to read.
`MEMORY_AND_DISK_SER`	Similar to MEMORY_ONLY_SER, but spill partitions that don't fit in memory to disk instead of recomputing them on the fly each time they're needed.
`DISK_ONLY`	Store the RDD partitions only on disk.
`MEMORY_ONLY_2`, `MEMORY_AND_DISK_2`, etc	Same as the levels above, but replicate each partition on two cluster nodes.

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Accumulators

- Similar to a MapReduce “Counter”
- A global variable to track metrics about your Spark program for debugging.
- Reasoning: Excutors do not communicate with each other.
- Sent back to driver

Broadcast Variables

- Similar to a MapReduce “Distributed Cache”
- Sends read-only values to worker nodes.
- Great for lookup tables, dictionaries, etc.

Hands-On: Spark Programming

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Functional tools in Python

- map
- filter
- reduce
- lambda
- IterTools
    - Chain, flatmap

Map

```
>>> a= [1,2,3]

>>> def add1(x) : return x+1

>>> map(add1, a)

     Result: [2,3,4]
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Filter

```
>>> a= [1,2,3,4]

>>> def isOdd(x) : return x%2==1

>>> filter(isOdd, a)

    Result: [1,3]
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Reduce

```
>>> a= [1,2,3,4]

>>> def add(x,y) : return x+y

>>> reduce(add, a)

     Result: 10
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

lambda

```
>>> (lambda x: x + 1)(3)
      Result: 4

>>> map((lambda x: x + 1), [1,2,3])

     Result: [2,3,4]
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Exercises

- (lambda x: 2*x)(3) => ?
- map(lambda x: 2*x, [1,2,3]) =>
- map(lambda t: t[0], [(1,2), (3,4), (5,6)]) =>
- reduce(lambda x,y: x+y, [1,2,3]) =>
- reduce(lambda x,y: x+y, map(lambda t: t[0], [(1,2), (3,4), (5,6)]))=>

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Start Spark-shell

```
$spark-shell
```

```
[root@quickstart 201402_babs_open_data]# spark-shell
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
      /_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)
Type in expressions to have them evaluated.
Type :help for more information.
```

Testing SparkContext

Spark-context

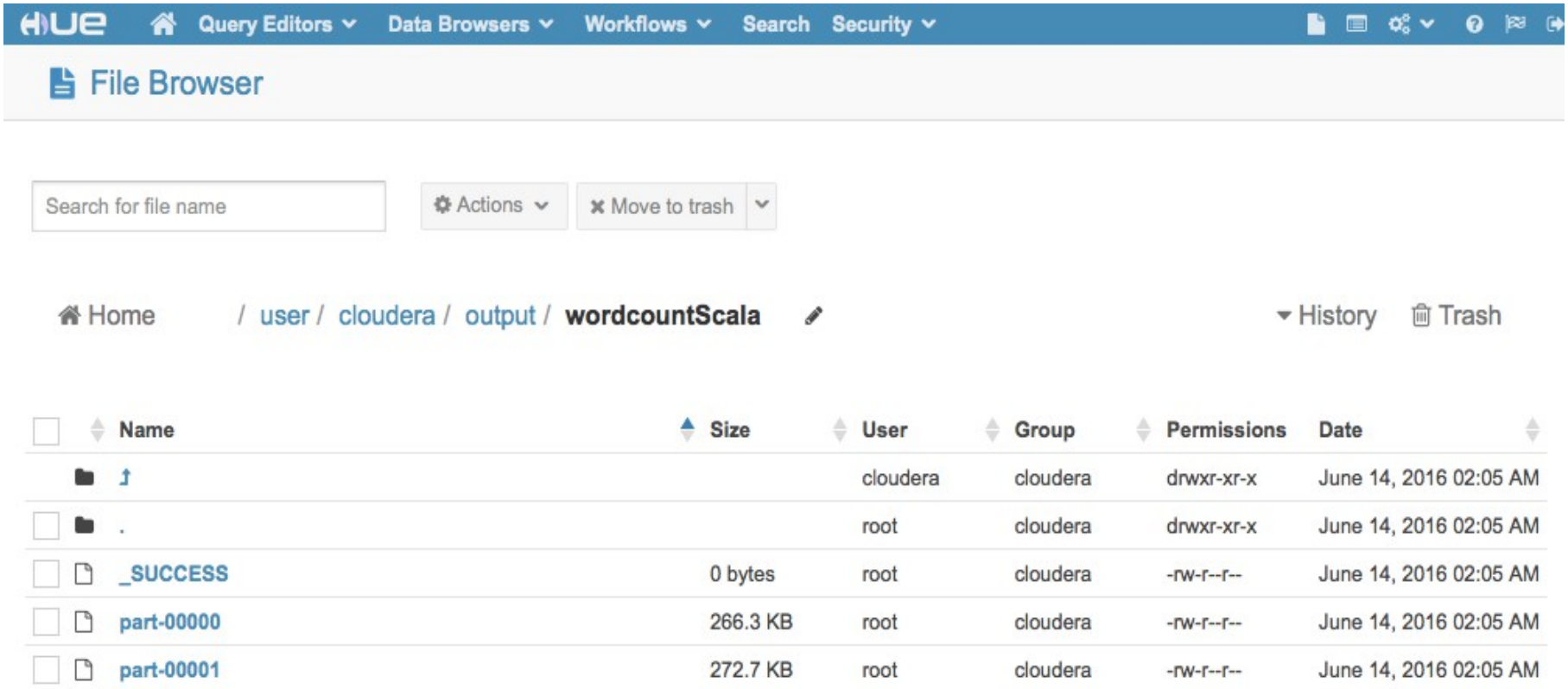
```
scala> sc
```

```
scala> sc
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@18c07e25
```

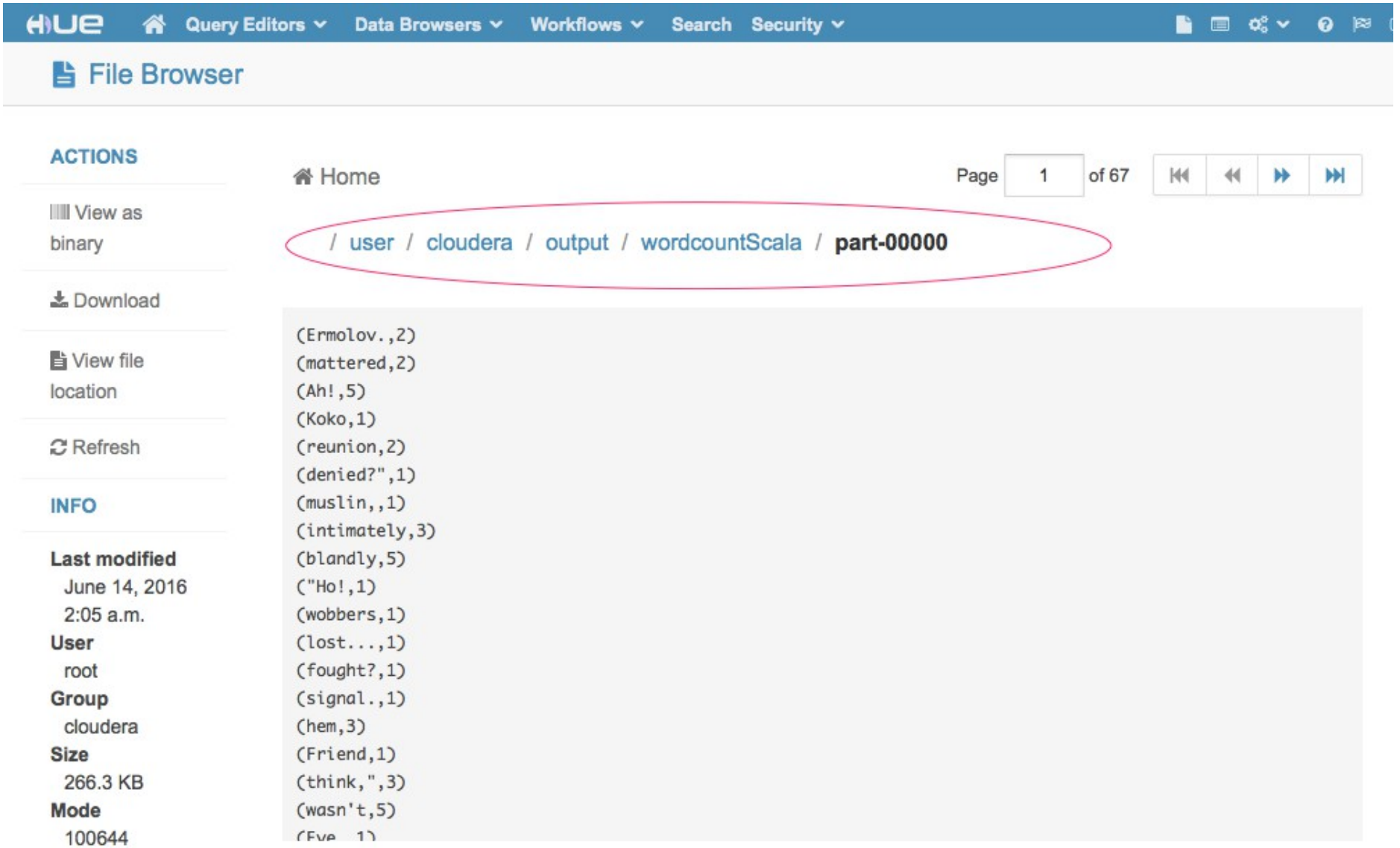
Spark Program in Scala: WordCount

```
scala> val file =
sc.textFile("hdfs:///user/cloudera/input/pg2600.txt")


scala> val wc = file.flatMap(l => l.split(" ")).map(word =>
(word, 1)).reduceByKey(_ + _)


scala>
wc.saveAsTextFile("hdfs:///user/cloudera/output/wordcountScala
")
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

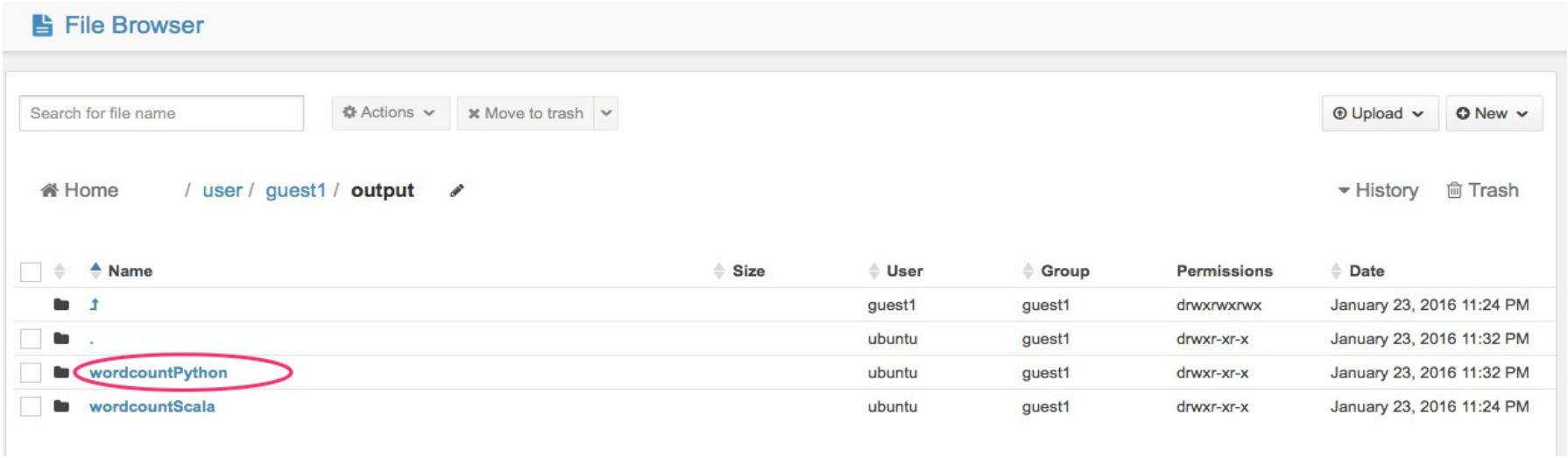

HUE
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Search for file name
Actions
Move to trash
Home / user / cloudera / output / wordcountScala
History
Trash
Name
Size
User
Group
Permissions
Date
cloudera cloudera drwxr-xr-x June 14, 2016 02:05 AM
. root cloudera drwxr-xr-x June 14, 2016 02:05 AM
_SUCCESS 0 bytes root cloudera -rw-r--r-- June 14, 2016 02:05 AM
part-00000 266.3 KB root cloudera -rw-r--r-- June 14, 2016 02:05 AM
part-00001 272.7 KB root cloudera -rw-r--r-- June 14, 2016 02:05 AM

WordCount output



Spark Program in Python: WordCount

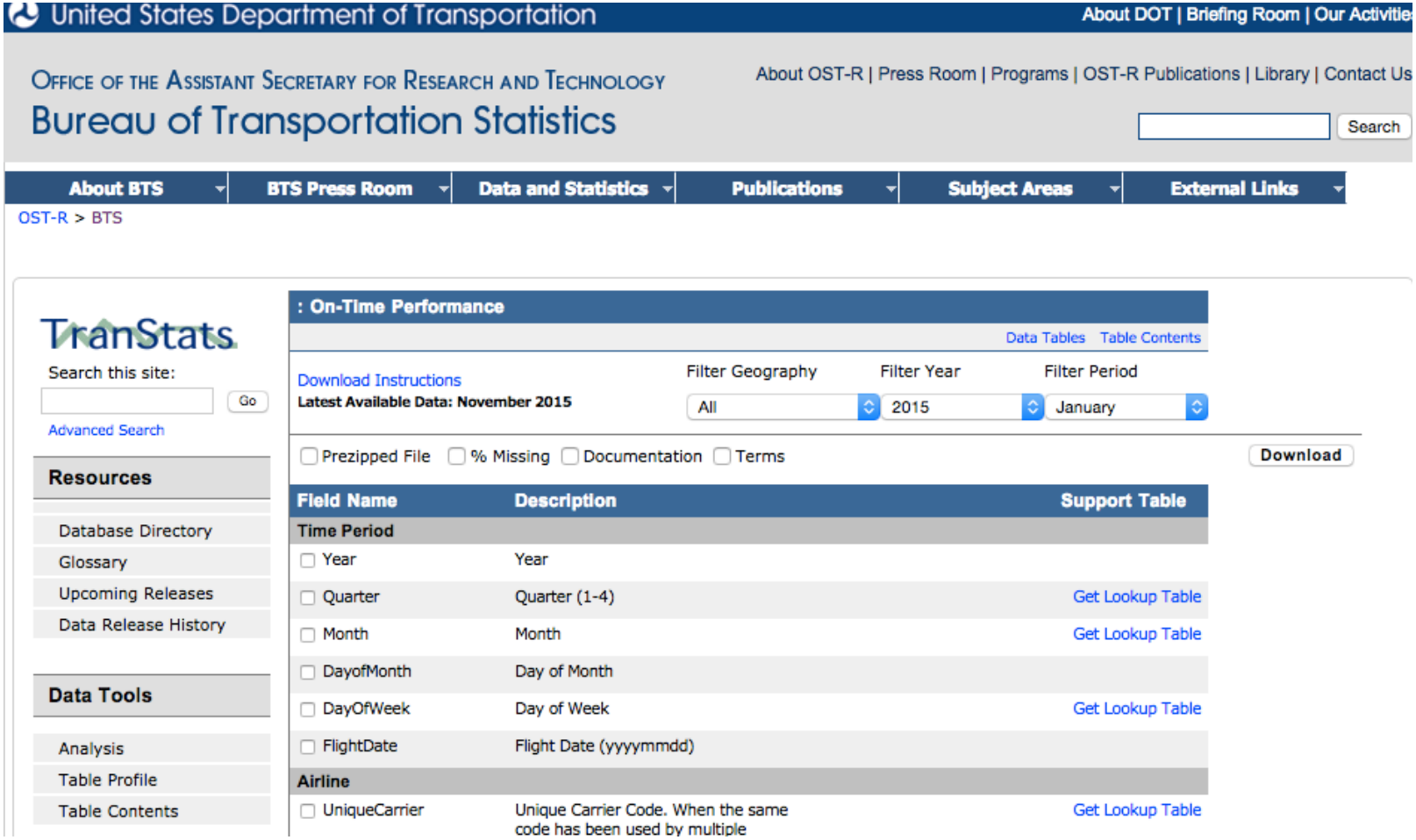
```
$ pyspark
>>> from operator import add
>>> file =
sc.textFile("hdfs:///user/cloudera/input/pg2600.txt")
>>> wc = file.flatMap(lambda x: x.split(' ')).map(lambda x:
(x, 1)).reduceByKey(add)
>>> wc.saveAsTextFile("hdfs:///user/cloudera/output/
wordcountPython")
```



Project: Flight

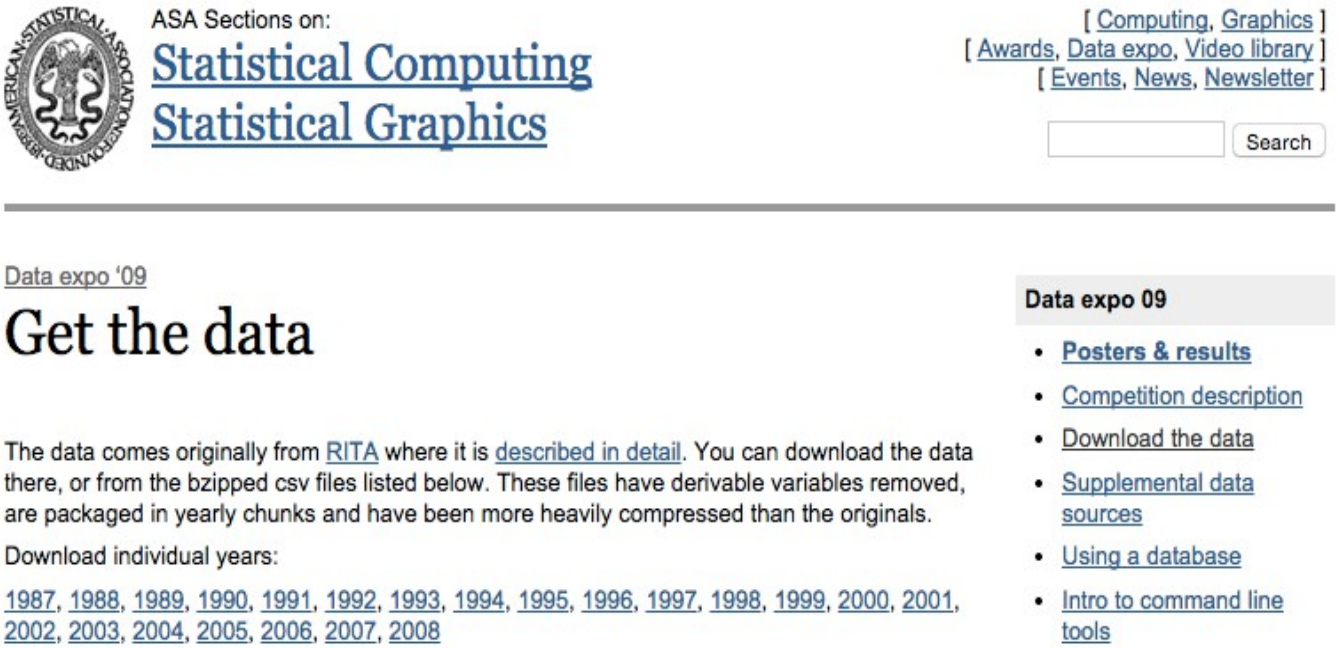
Flight Details Data

http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236



Flight Details Data

http://stat-computing.org/dataexpo/2009/the-data.html



ASA Sections on:
Statistical Computing
Statistical Graphics

[Computing, Graphics]
[Awards, Data expo, Video library]
[Events, News, Newsletter]

Search

Data expo '09

Get the data

The data comes originally from RITA where it is described in detail. You can download the data there, or from the bzipped csv files listed below. These files have derivable variables removed, are packaged in yearly chunks and have been more heavily compressed than the originals.

Download individual years:

1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008

Data expo 09

- **Posters & results**
- Competition description
- Download the data
- Supplemental data sources
- Using a database
- Intro to command line tools

Data Description

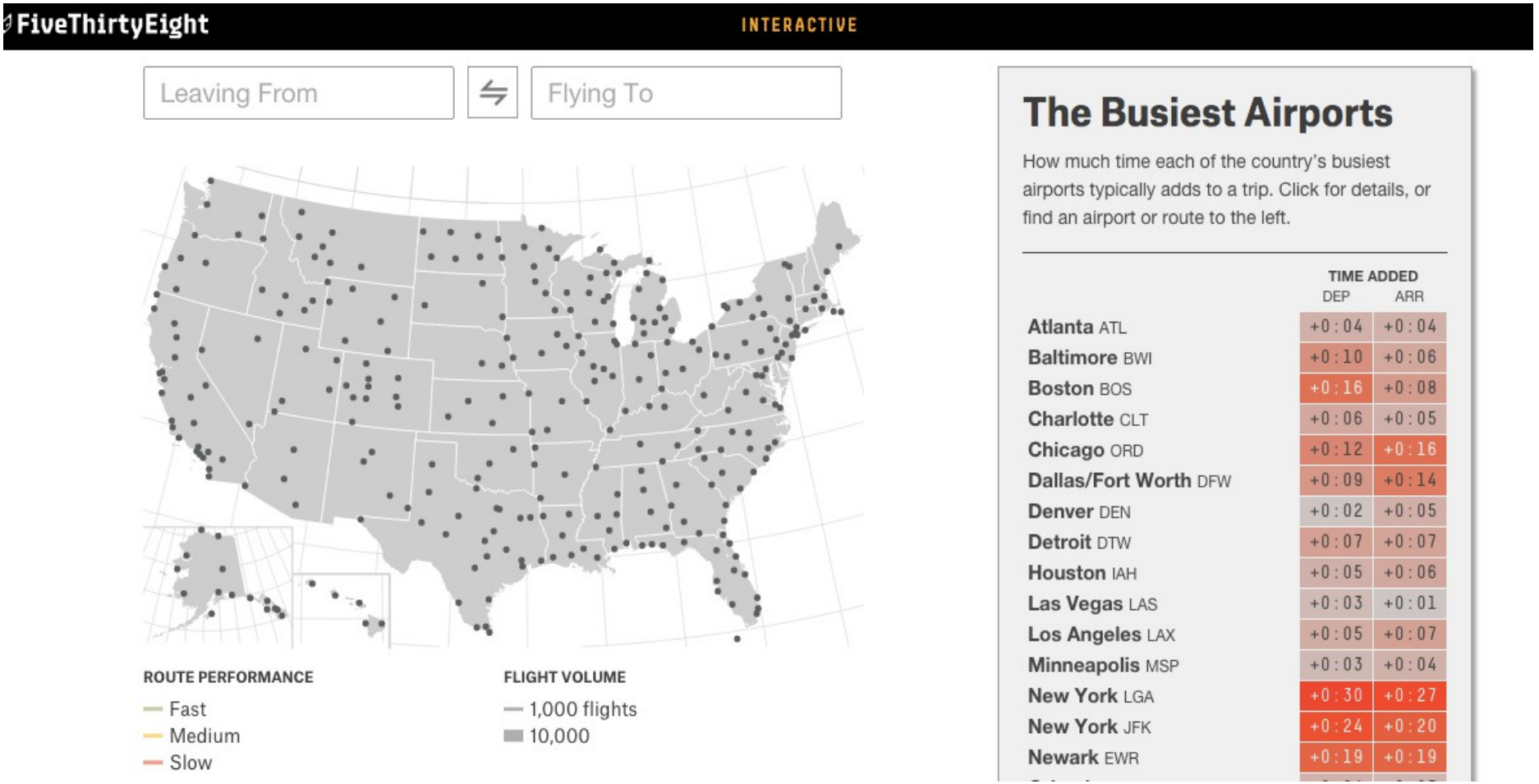
	Name	Description
1	Year	1987-2008
2	Month	1-12
3	DayofMonth	1-31
4	DayOfWeek	1 (Monday) - 7 (Sunday)
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plane tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin IATA airport code
18	Dest	destination IATA airport code
19	Distance	in miles
20	TaxiIn	taxi in time, in minutes
21	TaxiOut	taxi out time in minutes
22	Cancelled	was the flight cancelled?
23	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecurityDelay	in minutes
29	LateAircraftDelay	in minutes

Snapshot of Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Year	Month	DayofMor	DayOfWe	DepTime	CRSDepTi	ArrTime	CRSArrTim	UniqueCa	FlightNun	TailNum	ActualEla	CRSElapse	AirTime	ArrDelay	DepDelay	Origin	Dest	Distance	TaxiIn	TaxiO
2	2008	1	5	6	2243	1415	45	1625	WN	1684	N347SW	62	70	41	500	508	SAN	PHX	304	2	
3	2008	1	5	6	1940	1220	2111	1350	WN	1684	N347SW	91	90	64	441	440	SFO	SAN	447	5	
4	2008	1	7	1	111	1845	308	2045	WN	405	N644SW	117	120	103	383	386	MDW	JAN	666	4	
5	2008	1	7	1	2213	1700	2317	1655	WN	1827	N759GS	124	55	75	382	313	IND	MDW	162	10	
6	2008	1	7	1	2143	1720	26	1820	WN	1430	N644SW	163	60	83	366	263	STL	MDW	251	24	
7	2008	1	7	1	117	2020	302	2135	WN	490	N651SW	105	75	87	327	297	STL	TUL	351	5	
8	2008	1	7	1	2358	1855	105	2000	WN	490	N651SW	67	65	50	305	303	MDW	STL	251	4	
9	2008	1	3	4	2245	1730	2354	1850	WN	186	N792SW	69	80	59	304	315	JAN	HOU	359	3	
10	2008	1	7	1	2219	1730	35	1935	WN	2474	N710SW	76	65	67	300	289	MDW	CMH	284	2	
11	2008	1	5	6	2129	1620	2246	1750	WN	1924	N408WN	77	90	56	296	309	SFO	LAS	414	4	
12	2008	1	3	4	1615	1130	1623	1135	WN	10	N617SW	68	65	56	288	285	MAF	ABQ	332	4	
13	2008	1	3	4	1736	1305	2031	1555	WN	1837	N761RR	295	290	268	276	271	MDW	SFO	1855	4	
14	2008	1	5	6	2236	1805	2400	1930	WN	646	N283WN	84	85	71	270	271	LAX	SFO	337	6	
15	2008	1	3	4	2021	1700	2303	1835	WN	2005	N302SW	162	95	73	268	201	LAS	SFO	414	4	
16	2008	1	3	4	2059	1620	2216	1750	WN	1924	N761RR	77	90	60	266	279	SFO	LAS	414	6	
17	2008	1	7	1	2348	2105	307	2250	WN	3137	N358SW	259	165	244	257	163	MCO	MDW	989	1	
18	2008	1	3	4	2255	1820	509	55	WN	1924	N761RR	194	215	176	254	275	LAS	IND	1591	9	
19	2008	1	9	3	1458	1040	1725	1315	WN	2556	N501SW	87	95	76	250	258	BNA	BWI	588	4	
20	2008	1	7	1	2300	1835	113	2105	WN	2804	N420WN	253	270	240	248	265	MDW	PDX	1751	5	
21	2008	1	5	6	47	2040	151	2145	WN	505	N435WN	64	65	51	246	247	BWI	PVD	328	5	
22	2008	1	5	6	1558	1225	14	2010	WN	505	N442WN	316	285	250	244	213	SAN	BWI	2295	5	
23	2008	1	5	6	1931	1540	2104	1705	WN	1179	N718SW	93	85	77	239	231	SAN	OAK	446	7	
24	2008	1	4	5	1822	1425	2003	1605	WN	753	N726SW	101	100	88	238	237	PDX	OAK	543	6	

FiveThirtyEight

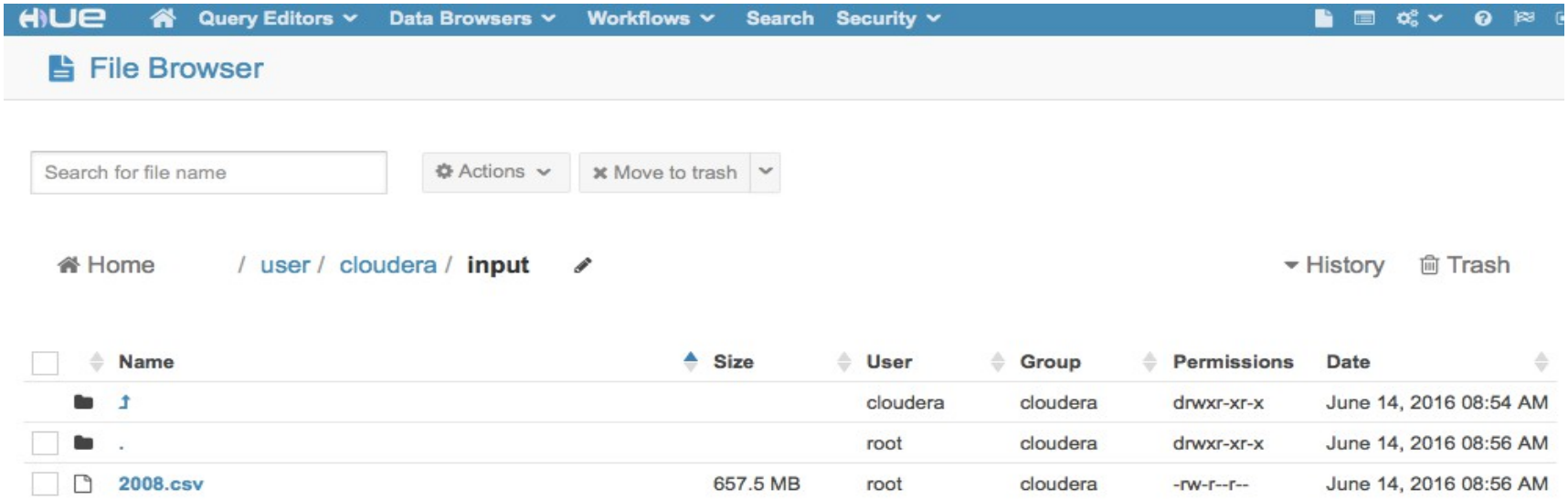
http://projects.fivethirtyeight.com/flights/



Spark Program : Upload Flight Delay Data

Upload a data to HDFS

```
$ wget
https://s3.amazonaws.com/imcbucket/data/flights/2008.csv
$ hadoop fs -put 2008.csv /user/cloudera/input
```



Spark Program : Navigating Flight Delay Data

```
>>> airline =
sc.textFile("hdfs:///user/cloudera/input/2008.csv")
>>> airline.take(2)
```

```
[u'Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueC
arrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDe
lay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,Carr
ierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay', u'2008,1,3,4,20
03,1955,2211,2225,WN,335,N712SW,128,150,116,-14,8,IAD,TPA,810,4,8,0,,0,NA,NA,NA,
NA,NA']
```

Spark Program : Preparing Data

```
>>> header_line = airline.first()
>>> header_list = header_line.split(',')
>>> airline_no_header = airline.filter(lambda row: row != header_line)
>>> airline_no_header.first()
>>> def make_row(row):
...     row_list = row.split(',')
...     d = dict(zip(header_list,row_list))
...     return d
...
>>> airline_rows = airline_no_header.map(make_row)
>>> airline_rows.take(5)
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark Program : Define convert_float function

```
>>> def convert_float(value):
...     try:
...             x = float(value)
...             return x
...     except ValueError:
...             return 0
...
>>>
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark Program : Finding best/worst airline

```
>>> carrier_rdd = airline_rows.map(lambda row:
(row['UniqueCarrier'],convert_float(row['ArrDelay'])))
>>> carrier_rdd.take(2)
```

```
16/06/17 17:48:06 INFO scheduler.DAGScheduler: Job 5 finished: runJob at PythonRDD.s
cala:393, took 0.062345 s
[(u'WN', -14.0), (u'WN', 2.0)]
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark Program : Finding best/worst airlines

```
>>> mean_delays_dest =
carrier_rdd.groupByKey().mapValues(lambda delays:
sum(delays.data)/len(delays.data))

>>> mean_delays_dest.sortBy(lambda t:t[1],
ascending=True).take(10)

>>> mean_delays_dest.sortBy(lambda t:t[1],
ascending=False).take(10)
```

```
[(u'AQ', -2.8708974358974357), (u'HA', 1.2518519716624075), (u'US', 2.800998260539827
8), (u'9E', 3.9874908469611912), (u'AS', 4.721360405553864), (u'WN', 5.115703380225
9031), (u'F9', 6.0841356696810847), (u'OO', 6.4389386397817896), (u'NW', 7.293465879
6727758), (u'DL', 7.7161646357519178)]
```

```
[(u'AA', 12.202853434950445), (u'OH', 11.404110178283158), (u'YV', 11.32256697917075
3), (u'UA', 11.001550560048052), (u'B6', 10.859381613638567), (u'CO', 10.80982057596
6226), (u'XE', 10.320298523403915), (u'EV', 10.00033146217589), (u'MQ', 9.4969706109
522658), (u'FL', 8.98815747237125 61)]
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Spark SQL

DataFrame

- A distributed collection of rows organied into named columns.
- An abstraction for selecting, filtering, aggregating, and plotting structured data.
- Previously => SchemaRDD

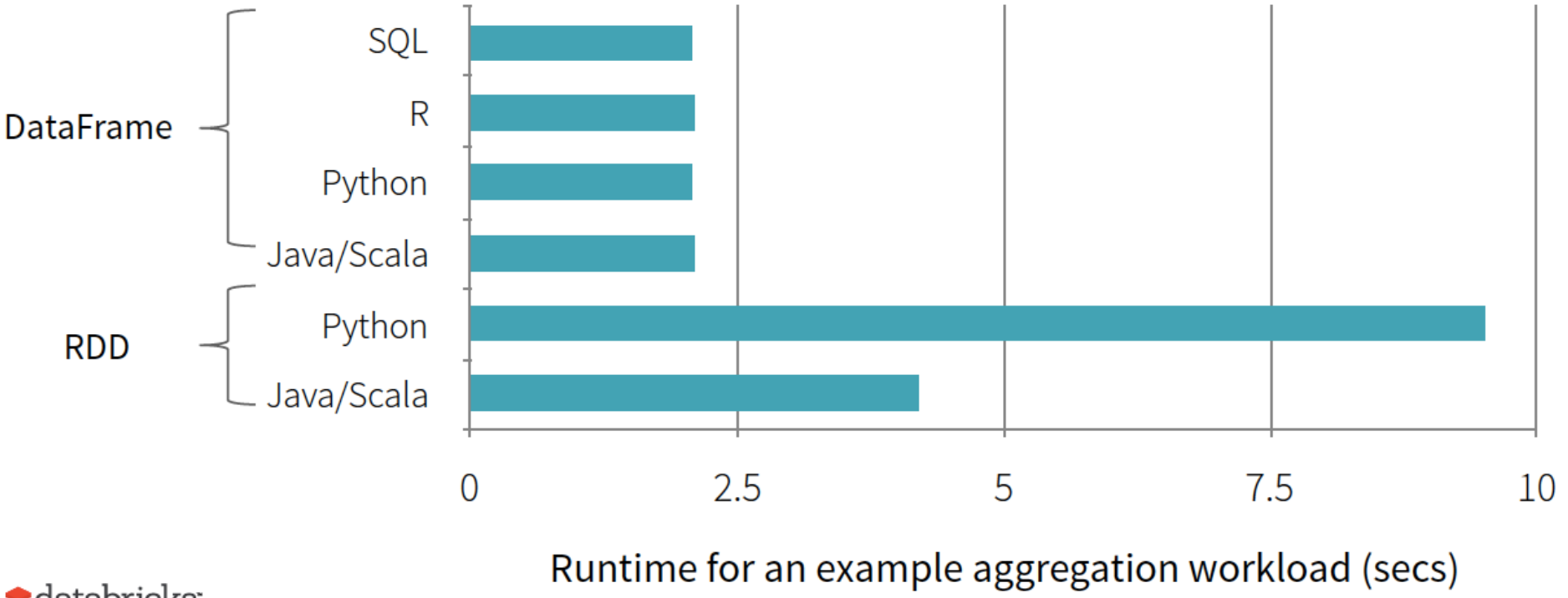
thanachart@imcinstitute.com

SparkSQL

- Creating and running Spark program faster
  - Write less code
  - Read less data
  - Let the optimizer do the hard work

thanachart@imcinstitute.com

Benefit of Logical Plan: Performance Parity Across Languages



databricks

Source: Jump start into Apache Spark and Databricks

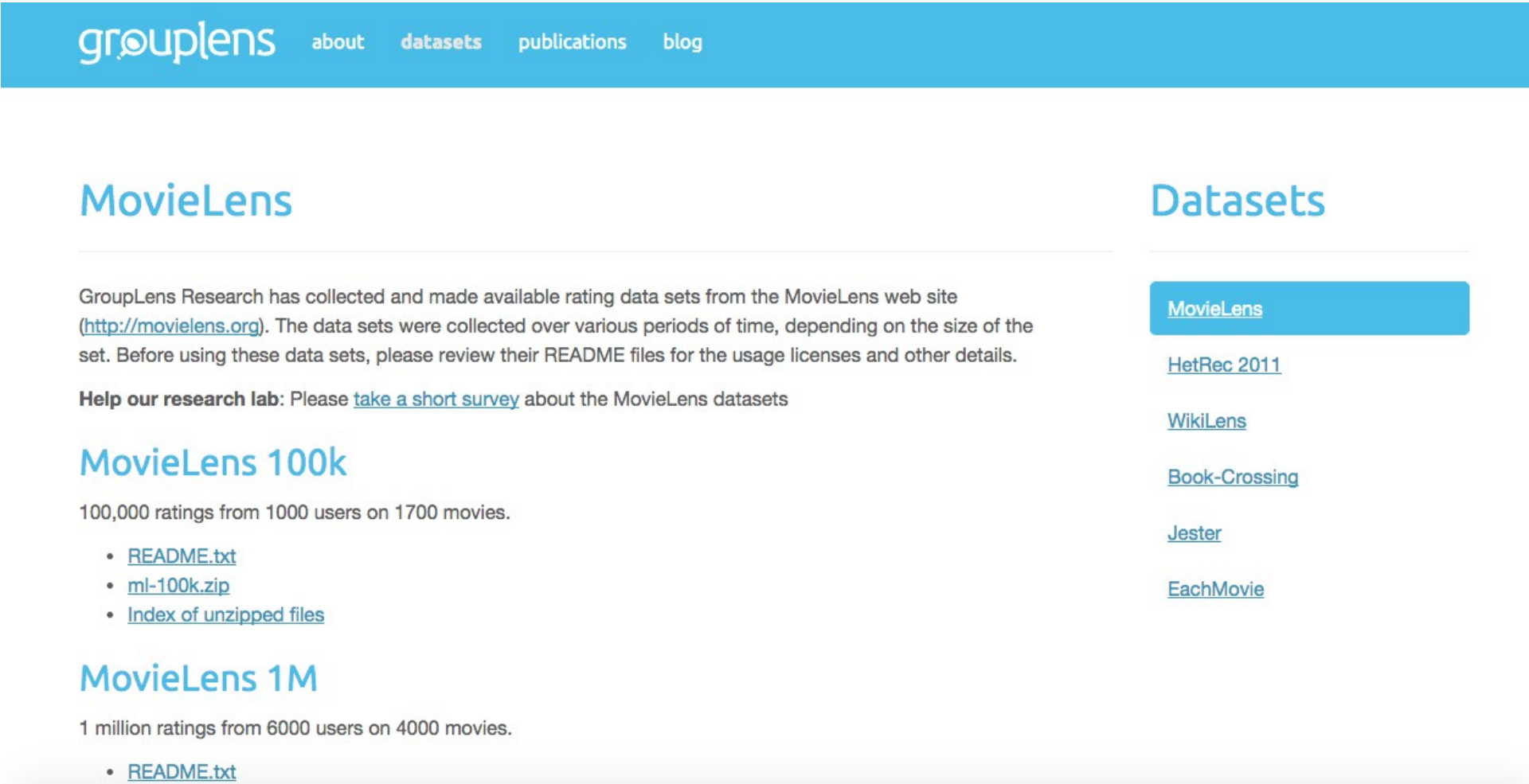
thanachart@imcinstitute.com

SparkSQL

```
context = ps.HiveContext(sc)

# query with SQL
results = context.sql(
  "SELECT * FROM people")

# apply Python transformation
names = results.map(lambda p: p.name)
```

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
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

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]#
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

SQL Spark MovieLens

Upload a data to HDFS then

```
$ pyspark --packages com.databricks:spark-csv_2.10:1.2.0
```

```
>>> df =
sqlContext.read.format('com.databricks.spark.csv').options(hea
der='true').load('hdfs:///user/cloudera/u.user')
```

```
>>> df.registerTempTable('user')
```

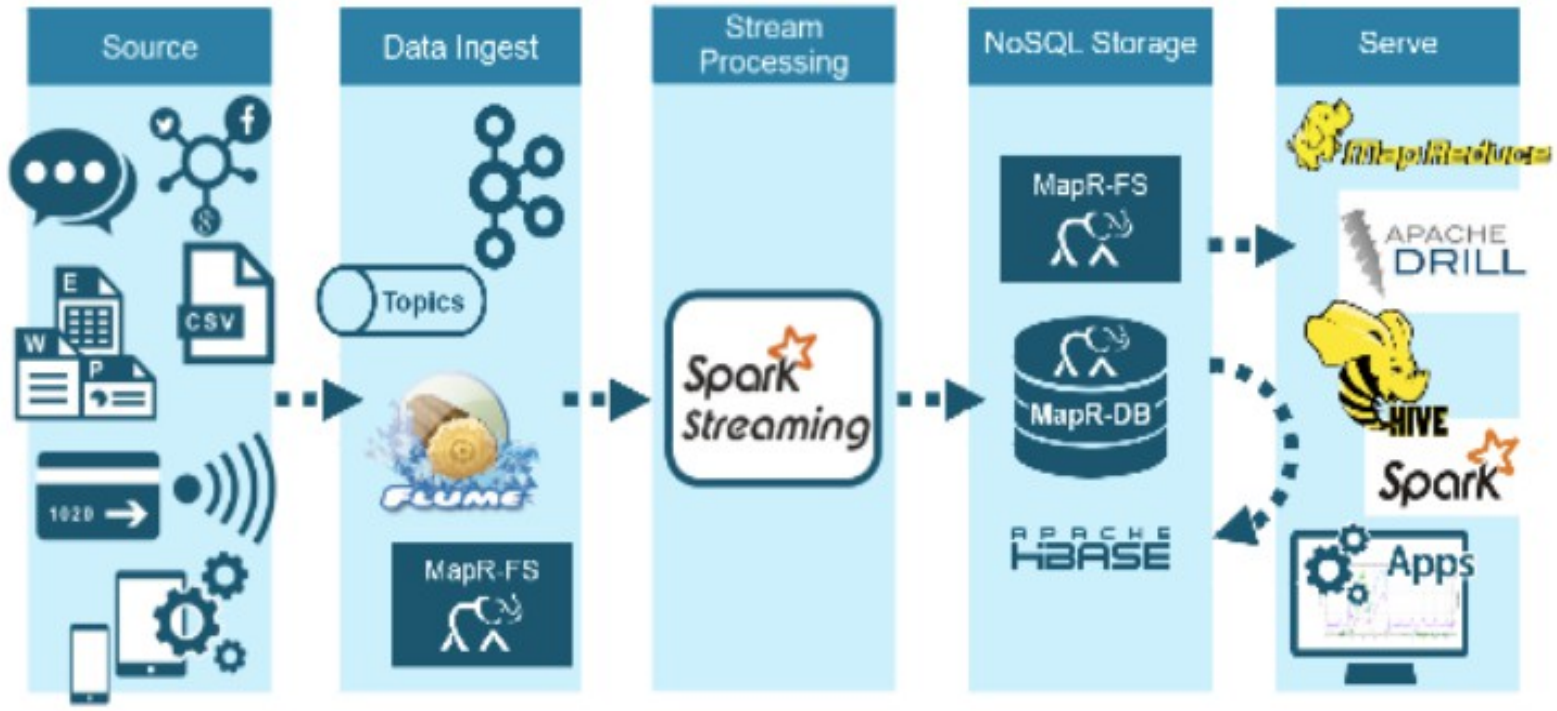
```
>>> sqlContext.sql("SELECT * FROM user").collect()
```

```
16/06/17 09:47:14 INFO scheduler.DAGScheduler: Job 1 finished: coll
ect at <stdin>:1, took 0.295085 s
[Row(1|24|M|technician|85711=u'2|53|F|other|94043'), Row(1|24|M|tec
hnician|85711=u'3|23|M|writer|32067'), Row(1|24|M|technician|85711=
u'4|24|M|technician|43537'), Row(1|24|M|technician|85711=u'5|33|F|o
ther|15213'), Row(1|24|M|technician|85711=u'6|42|M|executive|98101'
), Row(1|24|M|technician|85711=u'7|57|M|administrator|91344'), Row(
1|24|M|technician|85711=u'8|36|M|administrator|05201'), Row(1|24|M|
technician|85711=u'9|29|M|student|01002'), Row(1|24|M|technician|85
711=u'10|53|M|lawyer|90703'), Row(1|24|M|technician|85711=u'11|39|F
|other|30329'), Row(1|24|M|technician|85711=u'12|28|F|other|06405')
, Row(1|24|M|technician|85711=u'13|47|M|educator|29206'), Row(1|24|
M|technician|85711=u'14|45|M|scientist|55106'), Row(1|24|M|technici
an|85711=u'15|49|F|educator|97301'), Row(1|24|M|technician|85711=u'
16|21|M|entertainment|10309'), Row(1|24|M|technician|85711=u'17|30|
```

Spark Streaming

Thanachart Numnonda, thanachart@imcinstitute.com June 2016

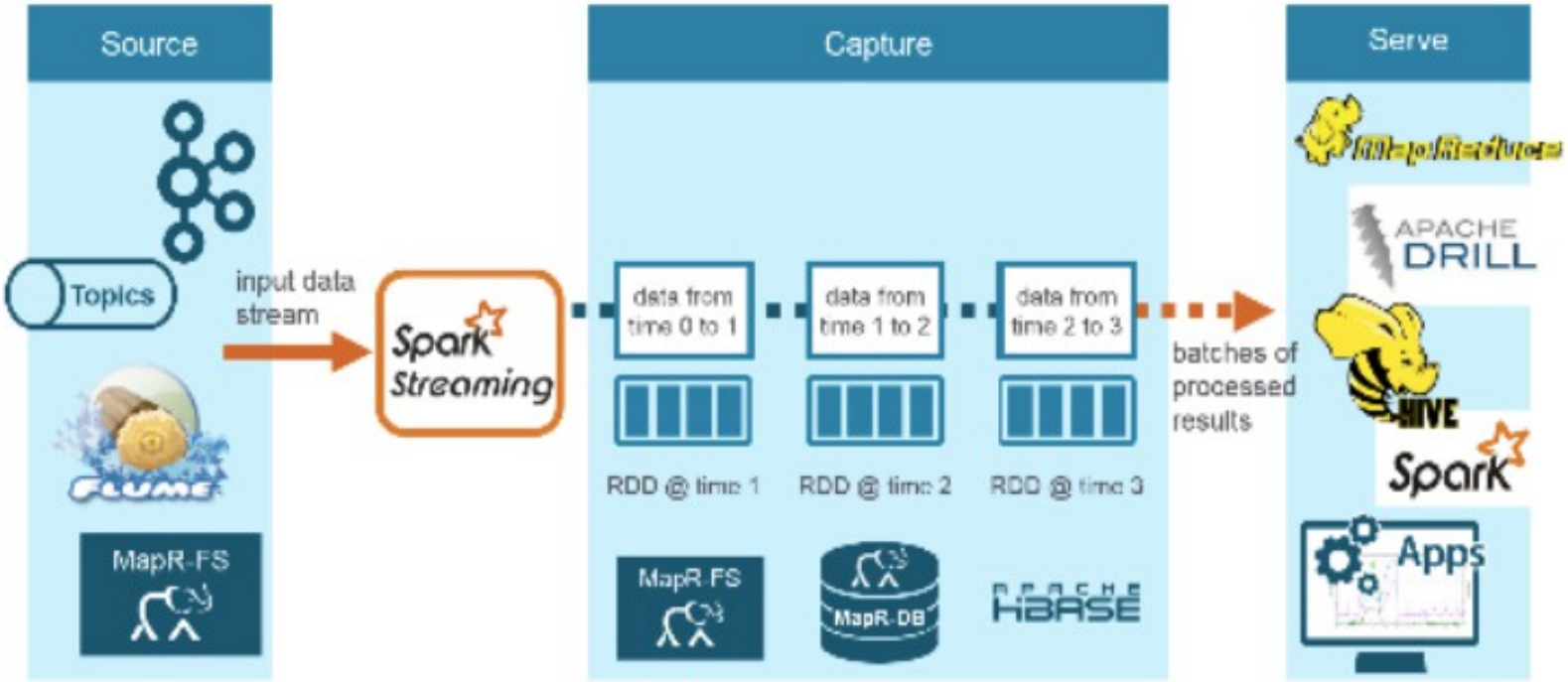
Stream Process Architecture



Source: MapR Academy

thanachart@imcinstitute.com

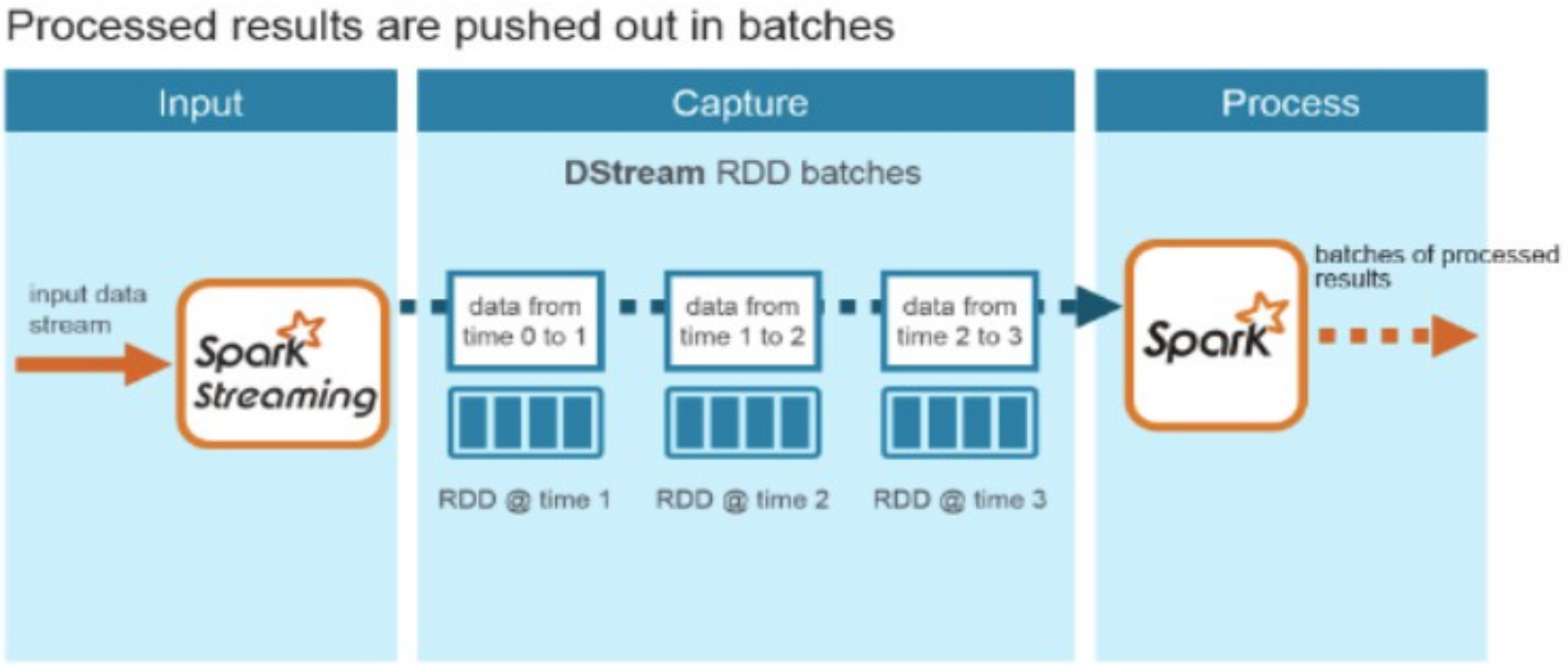
Spark Streaming Architecture



Source: MapR Academy

thanachart@imcinstitute.com

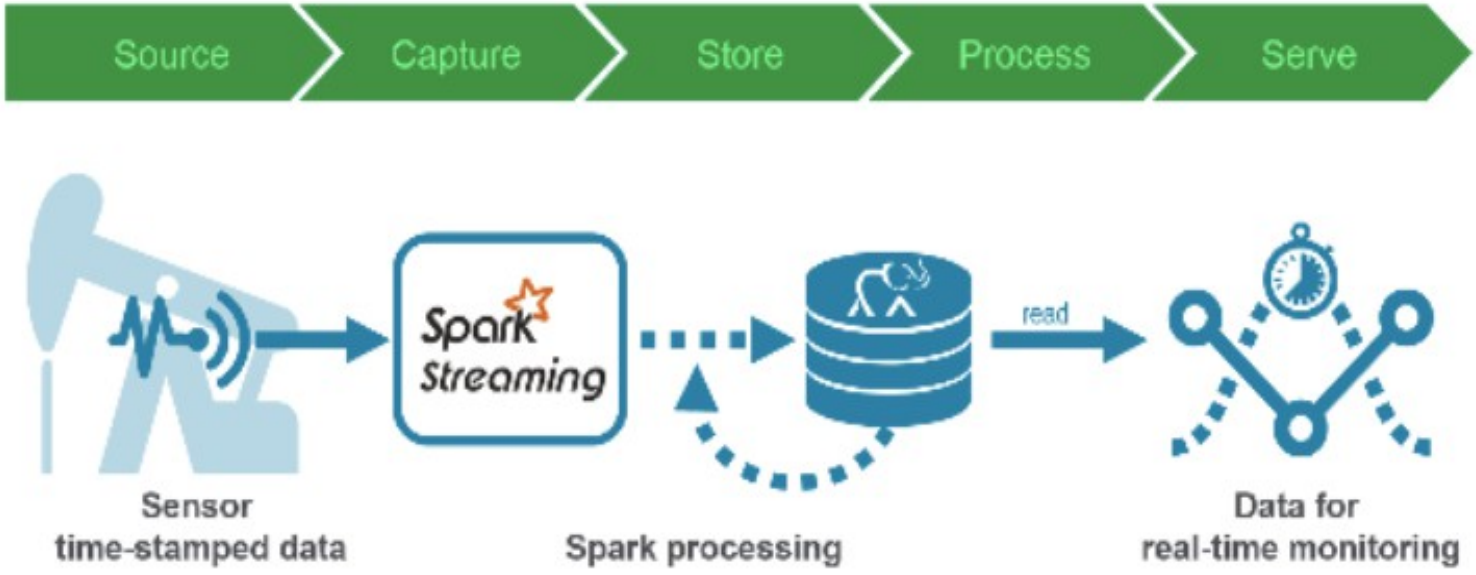
Processing Spark DStreams



Source: MapR Academy

thanachart@imcinstitute.com

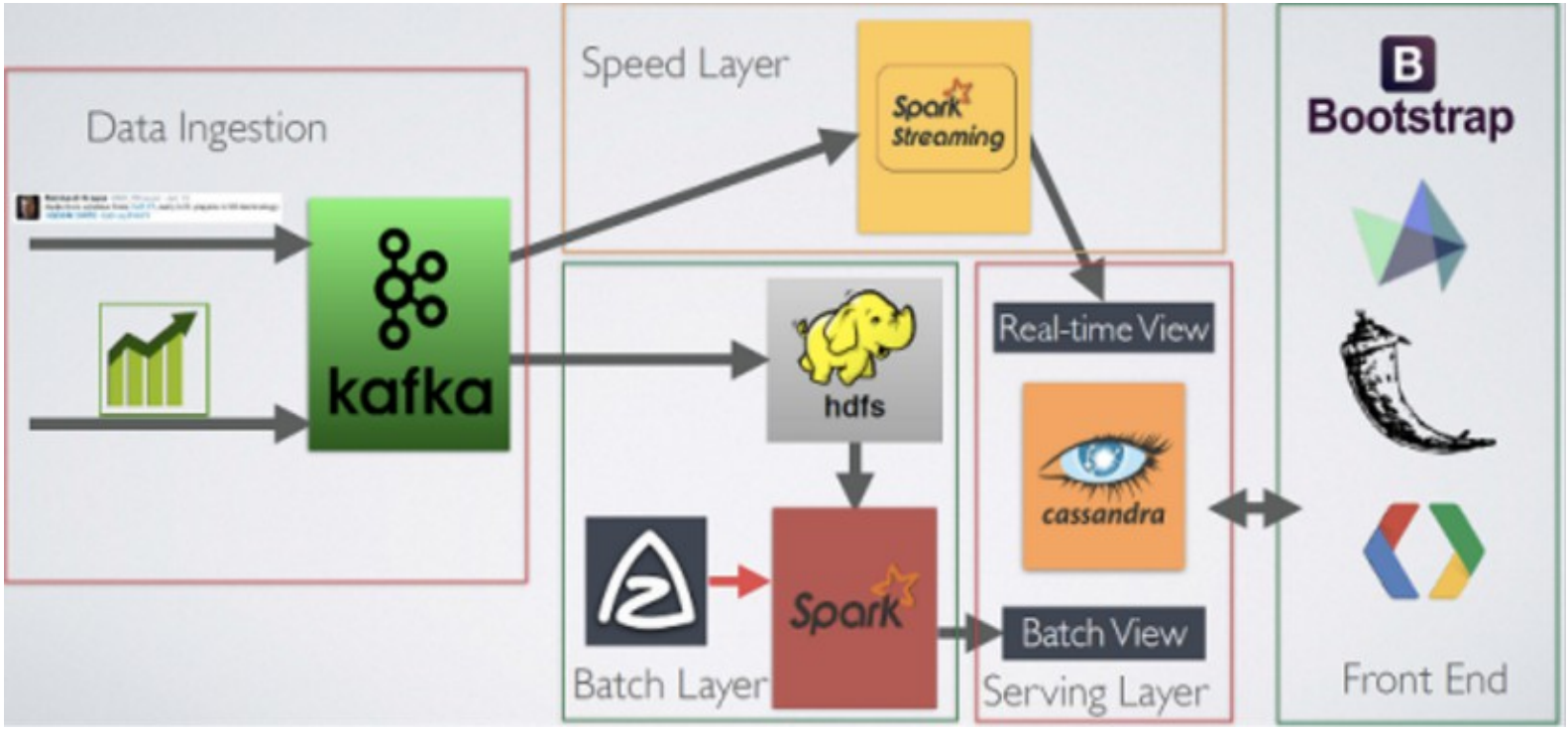
Use Case: Time Series Data



Source: MapR Academy

thanachart@imcinstitute.com

Use Case



Source: http://www.insightdataengineering.com/

thanachart@imcinstitute.com

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
      /_/
```

WordCount using Spark Streaming

```
$ 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._
val ssc = new StreamingContext(sc, Seconds(2))
val lines = ssc.socketTextStream("localhost",8585,MEMORY_ONLY)
val wordsFlatMap = lines.flatMap(_.split(" "))
val wordsMap = wordsFlatMap.map( w => (w,1))
val wordCount = wordsMap.reduceByKey( (a,b) => (a+b))
wordCount.print
ssc.start
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Running the netcat server on another window

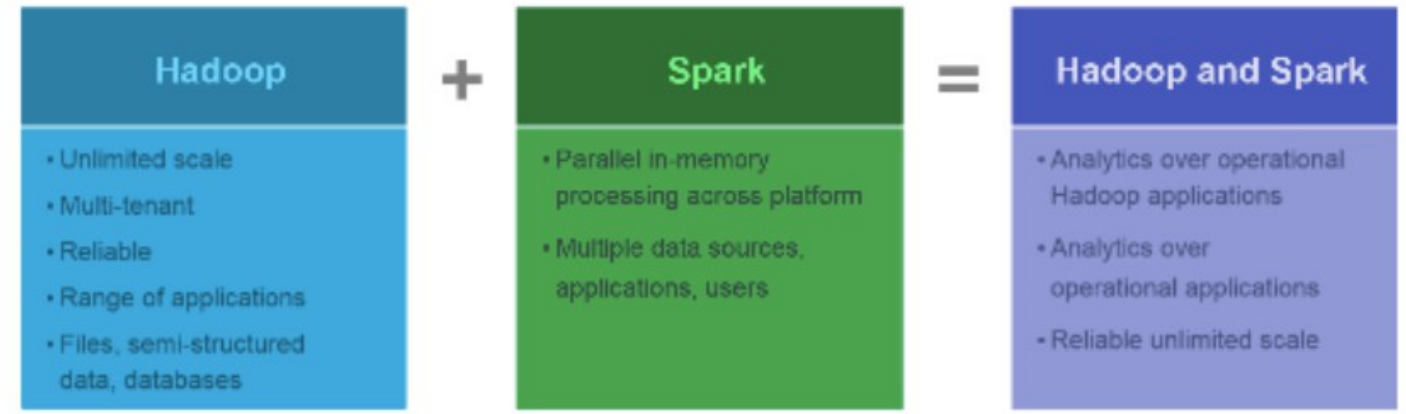
```
ubuntu@ip-172-31-30-238:~$ sudo su
root@ip-172-31-30-238:/home/ubuntu# docker ps
CONTAINER ID        IMAGE                              COMMAND                  CREATED
        STATUS              PORTS                      NAMES
581c45e85683        cloudera/quickstart:latest   "/usr/bin/docker-qui   About an hou
r ago   Up About an hour    0.0.0.0:8888->8888/tcp   backstabbing_lumiere
root@ip-172-31-30-238:/home/ubuntu# docker exec -i -t 581c45e85683 nc -lk 8585

test this
It is another test on Spark streaming. It is great
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

```
-------------------------------------------
Time: 1465924608000 ms
-------------------------------------------
(streaming.,1)
(Spark,1)
(great,1)
(is,2)
(test,1)
(another,1)
(on,1)
(It,2)
```


Thanachart Numnonda, thanachart@imcinstitute.com June 2016

Hadoop + Spark



Source: MapR Academy

thanachart@imcinstitute.com

Challenge: Dataset

NYC's Taxi Trip Data

http://www.andresmh.com/nyctaxitrips/

NYC Taxi Trips

Data obtained through a FOIA request

View the Project on GitHub
andresmh/nyctaxitrips

View On **GitHub** | Read the **story** | Check the **viz**

NYC Taxi Data

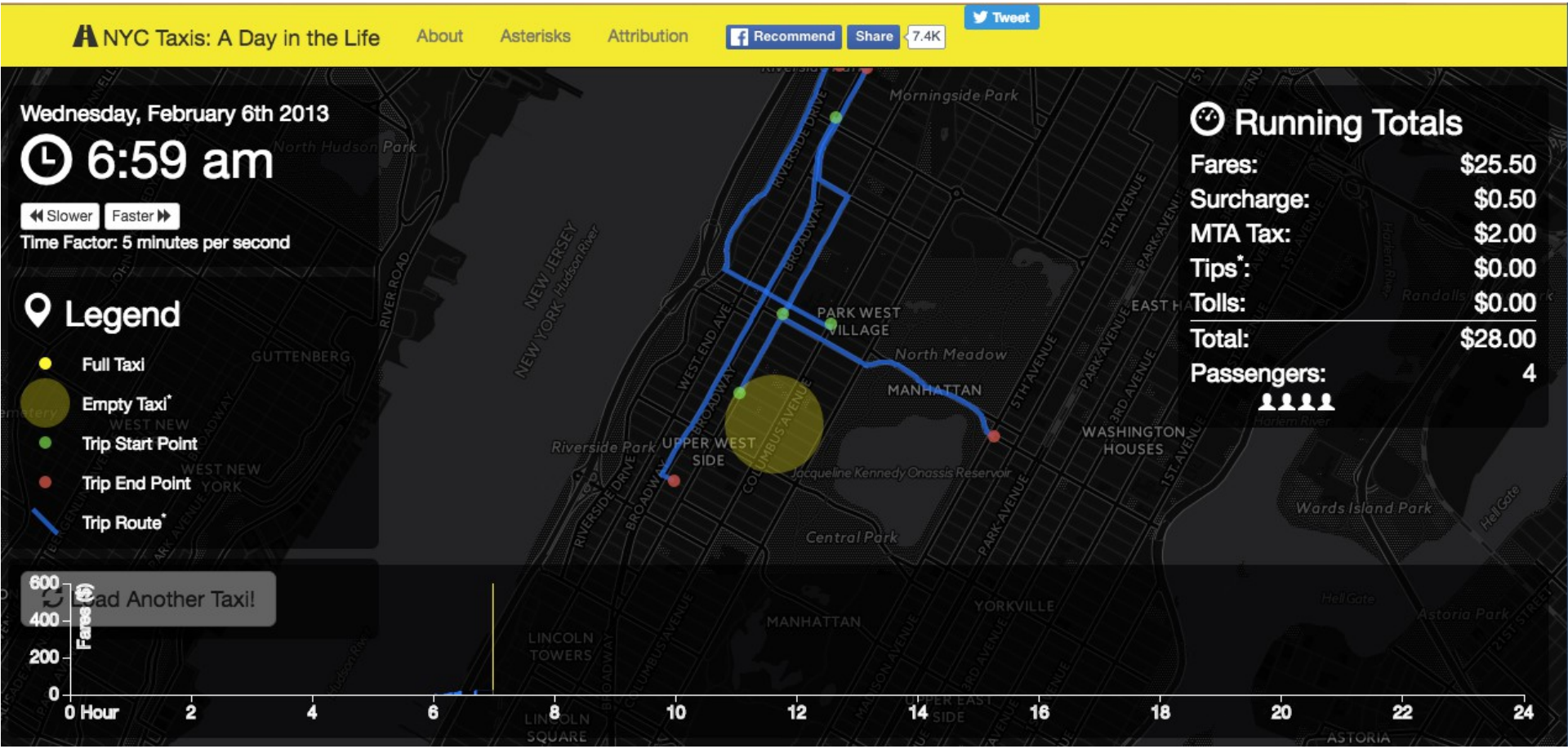
Trips

1. trip_data_1.csv.zip
2. trip_data_2.csv.zip
3. trip_data_3.csv.zip
4. trip_data_4.csv.zip
5. trip_data_5.csv.zip
6. trip_data_6.csv.zip
7. trip_data_7.csv.zip
8. trip_data_8.csv.zip
9. trip_data_9.csv.zip
10. trip_data_10.csv.zip
11. trip_data_11.csv.zip
12. trip_data_12.csv.zip

thanachart@imcinstitute.com

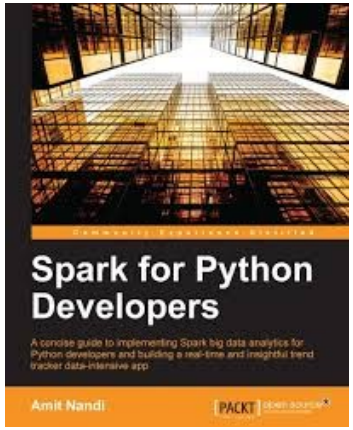
NYC Taxi : A Day in Life

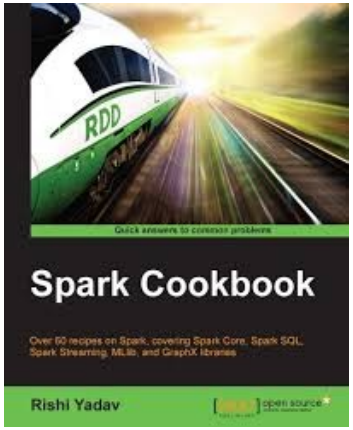
http://nyctaxi.herokuapp.com/

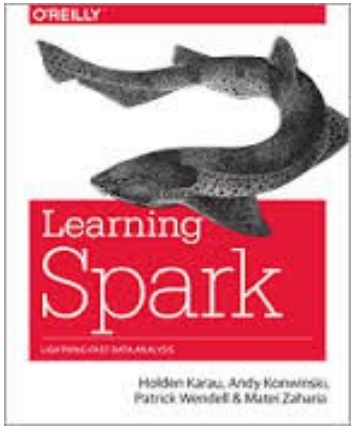


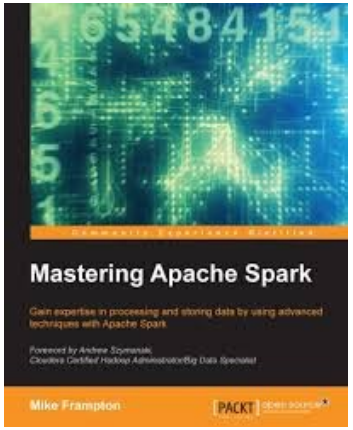
thanachart@imcinstitute.com

Recommended Books

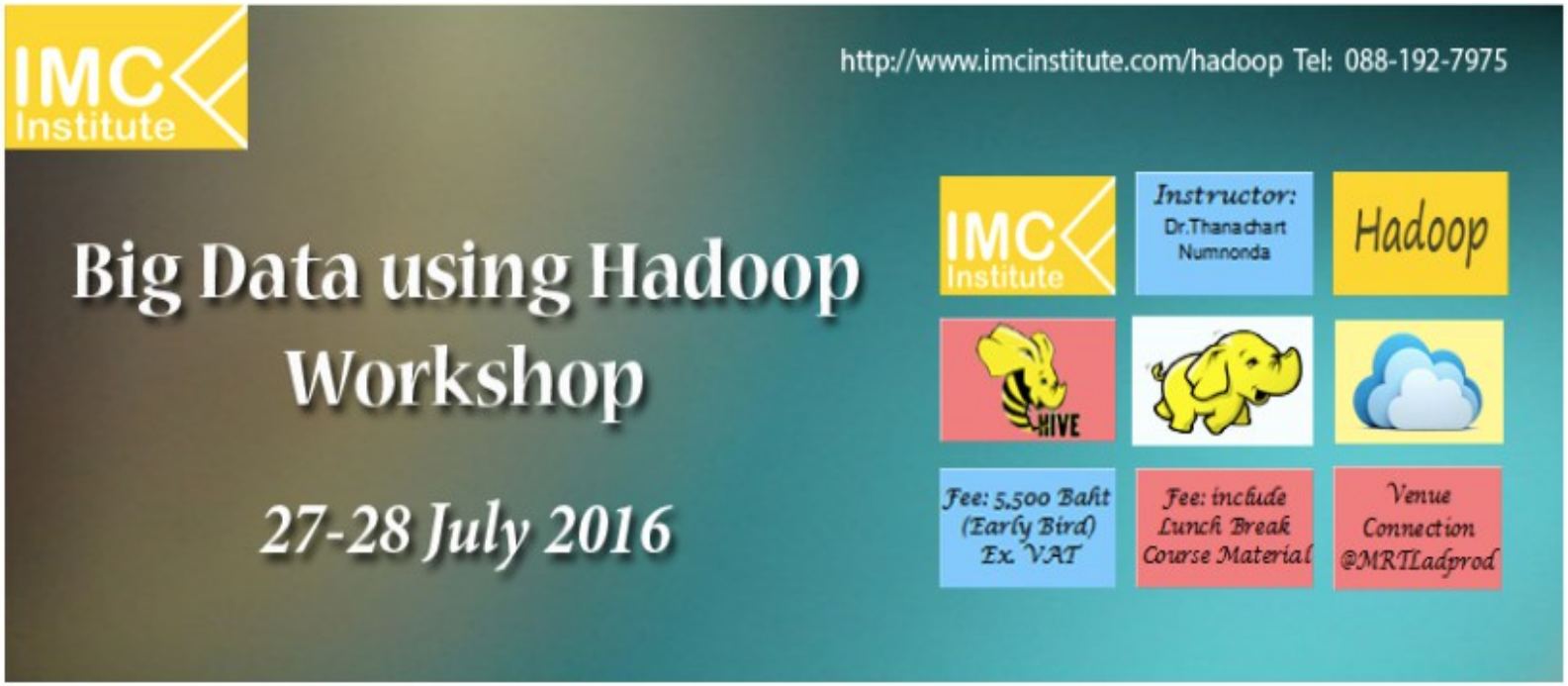








thanachart@imcinstitute.com



thanachart@imcinstitute.com

Thank you

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