



Big Data using Hadoop

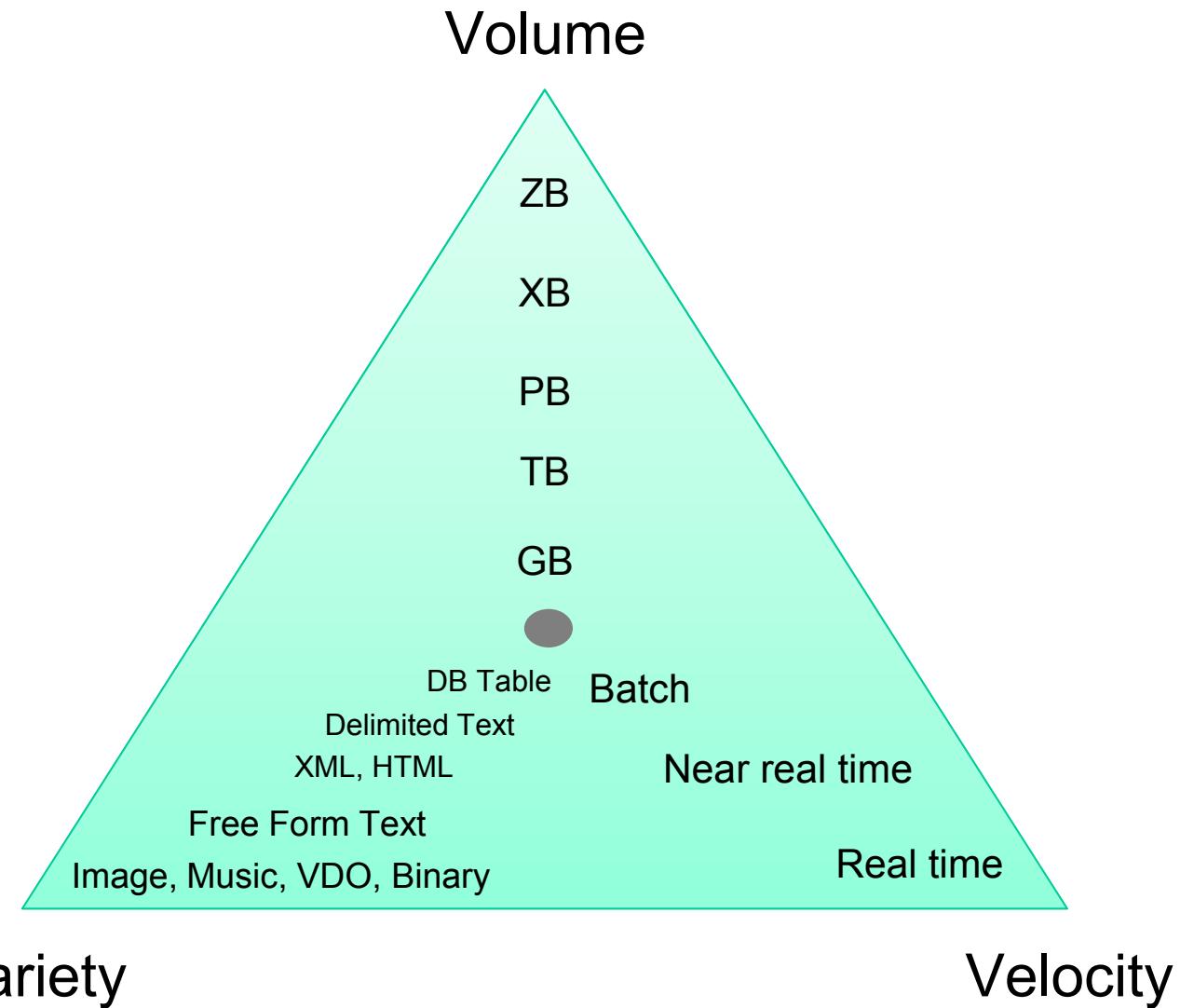
Hands On Workshop

Dr.Thanachart Numnonda
Certified Java Programmer
thanachart@imcinstitute.com

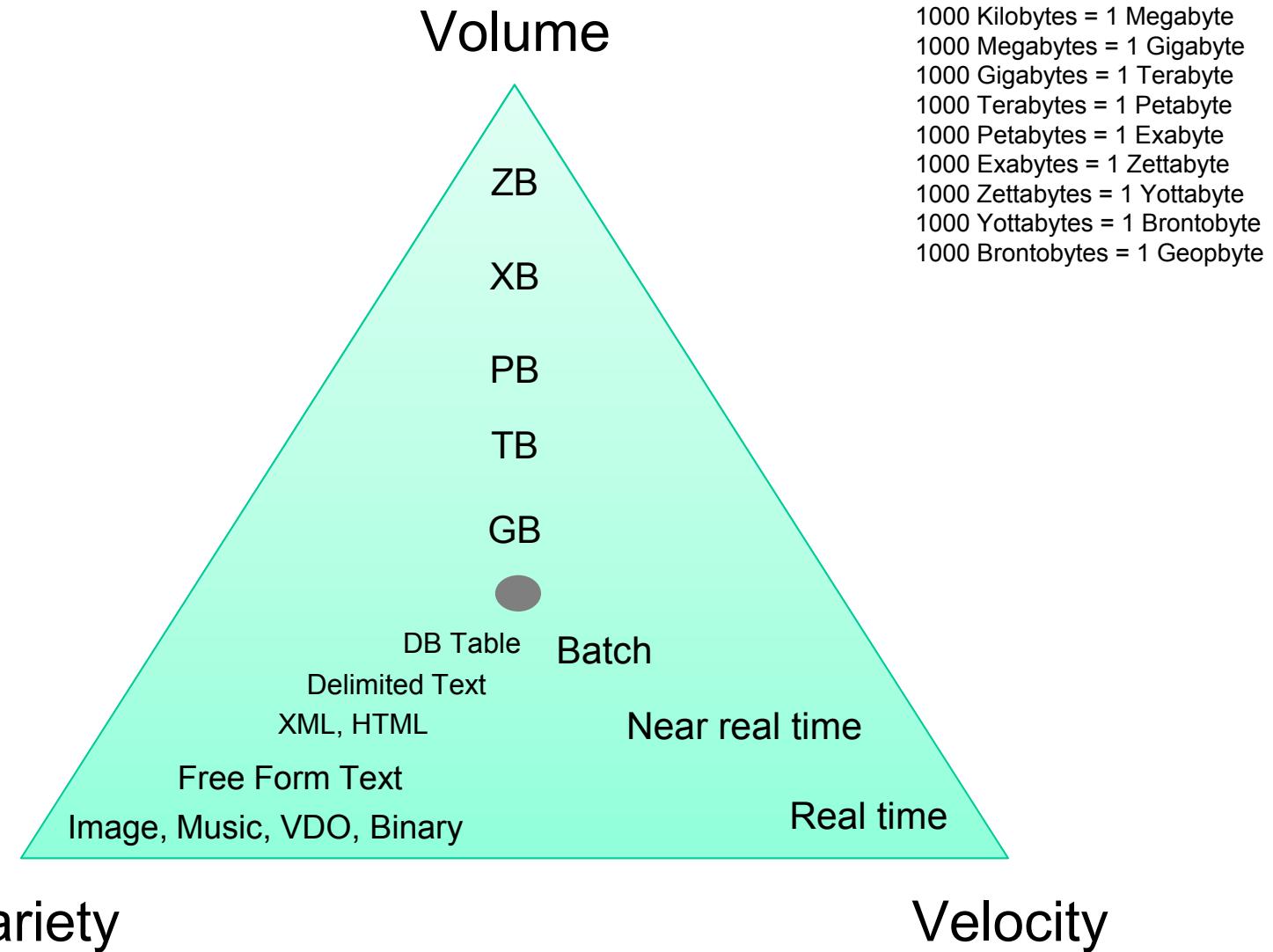
Danairat T.
Certified Java Programmer, TOGAF – Silver
danairat@gmail.com, +66-81-559-1446



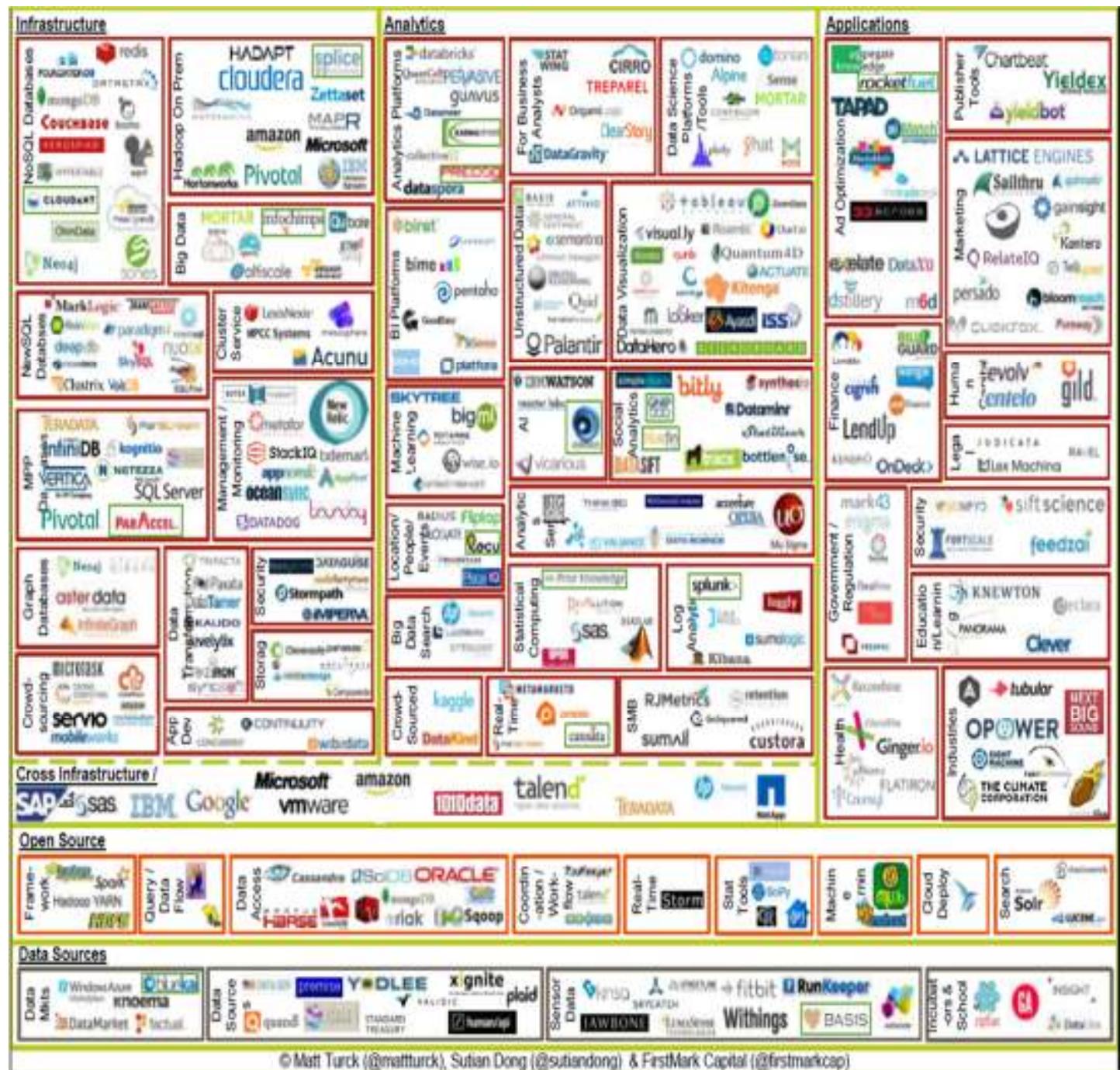
Big Data Introduction



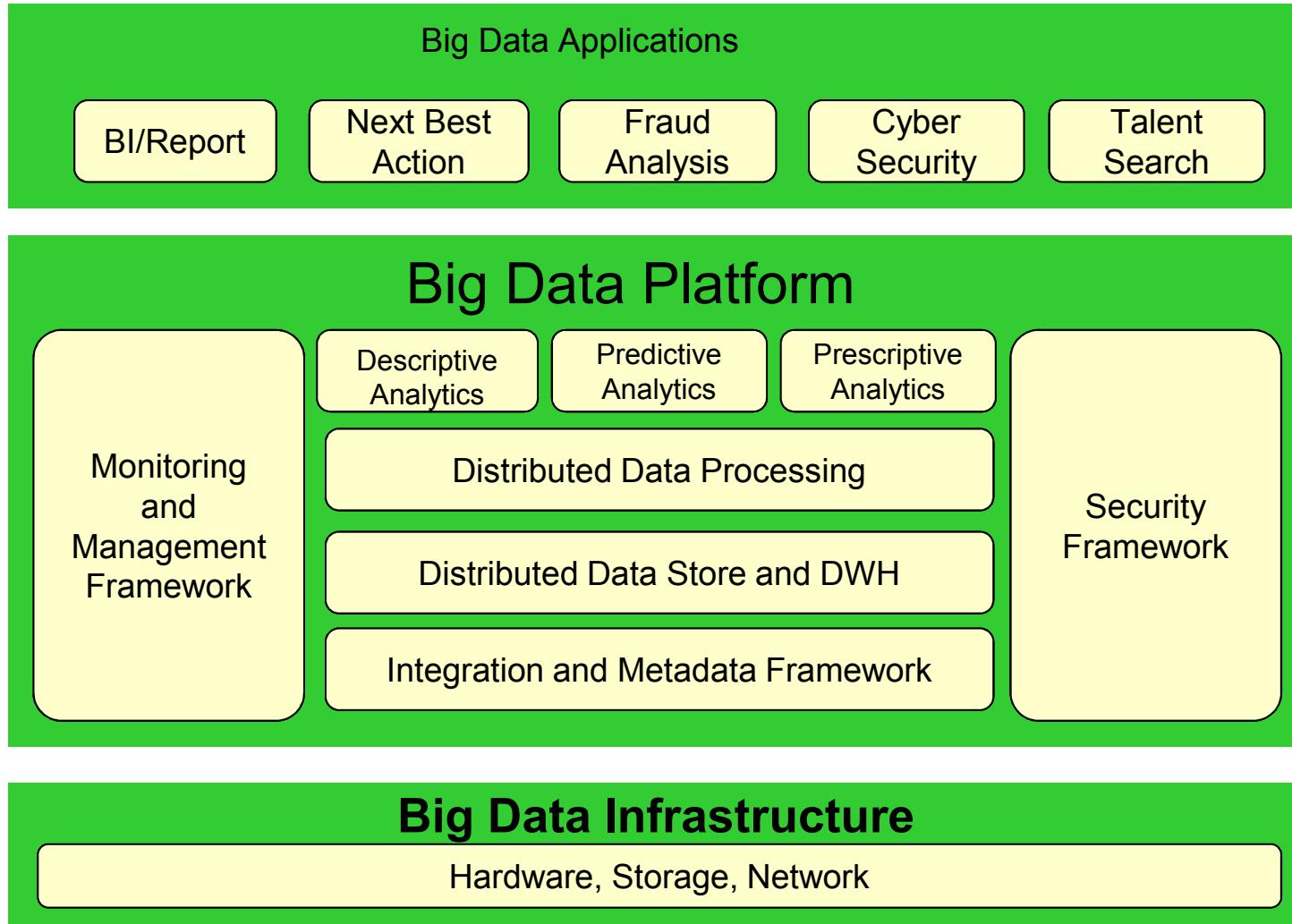
Big Data Introduction



Big Data Players



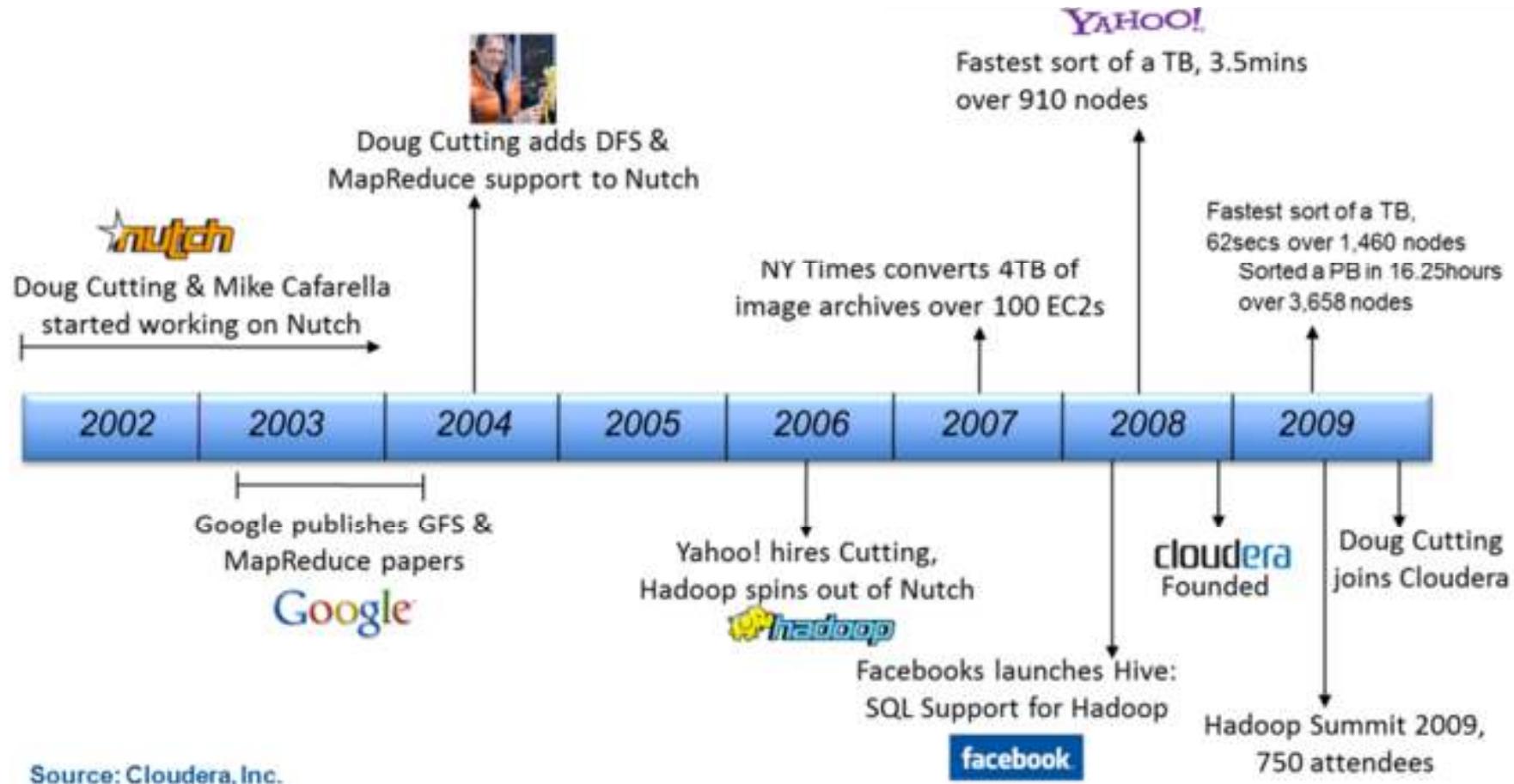
Big Data Architecture





Lecture: Hadoop

Hadoop Timeline



Apache Hadoop Core Technology



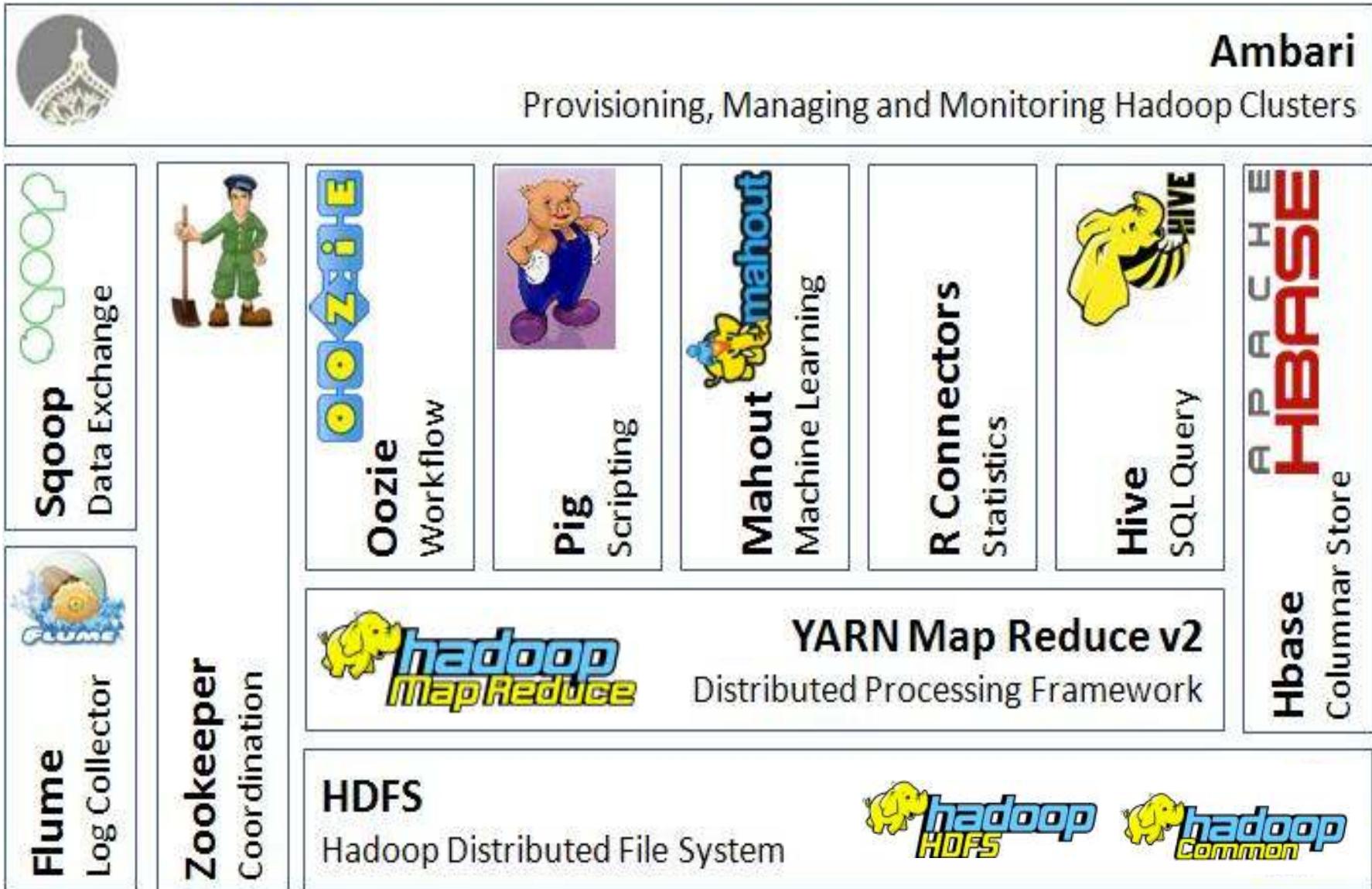
YARN Map Reduce v2
Distributed Processing Framework

HDFS
Hadoop Distributed File System



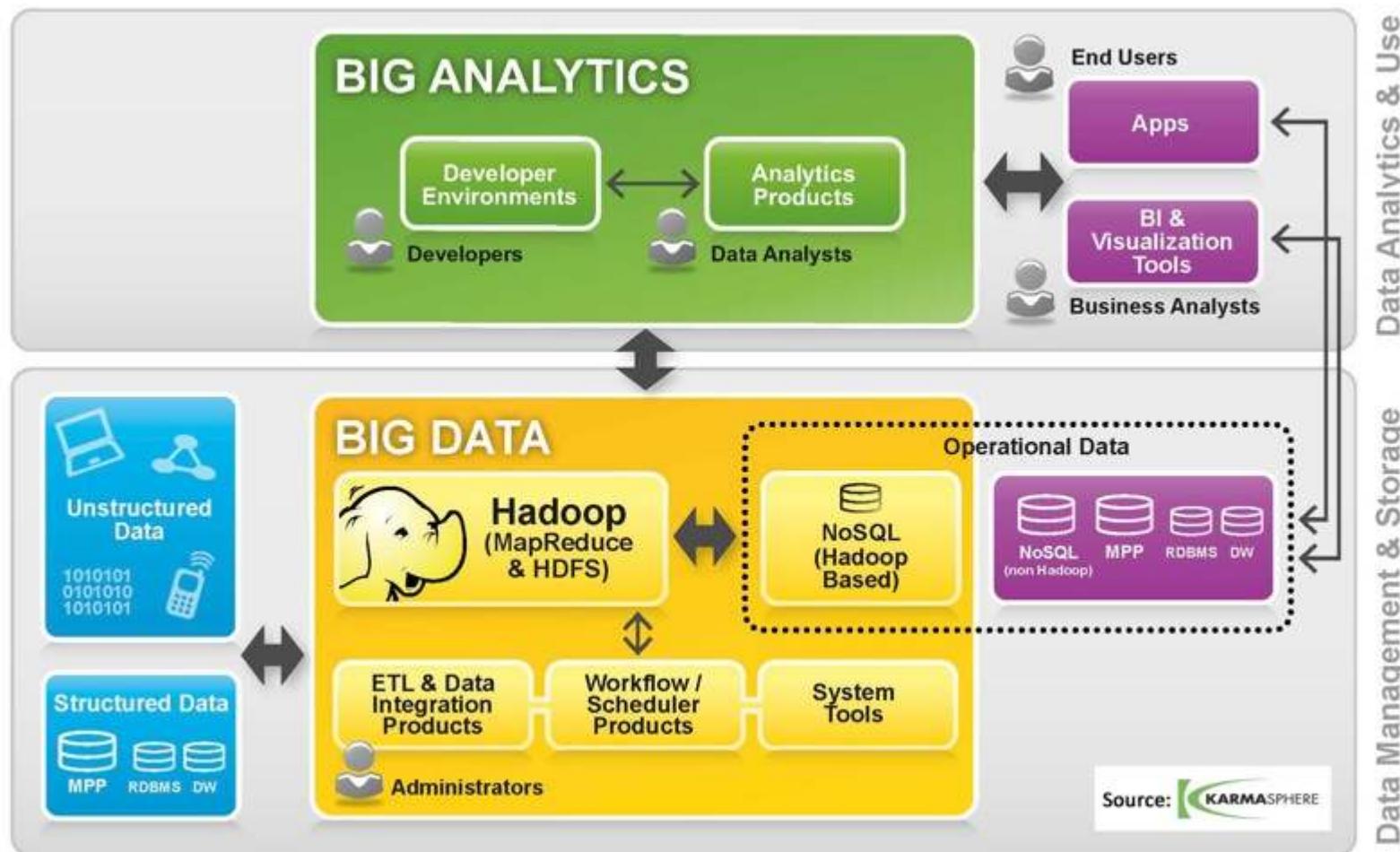
j2eedev.org/ecosystem-hadoop

Apache Hadoop Ecosystem



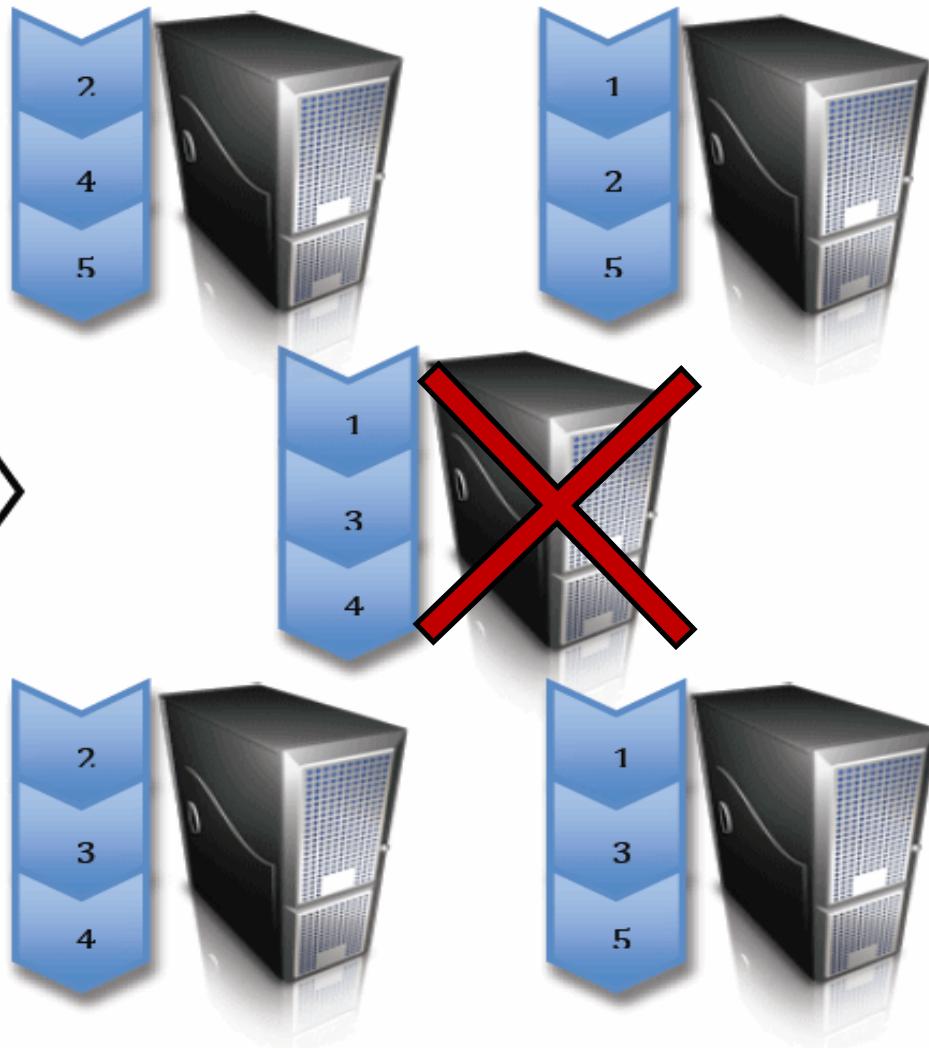
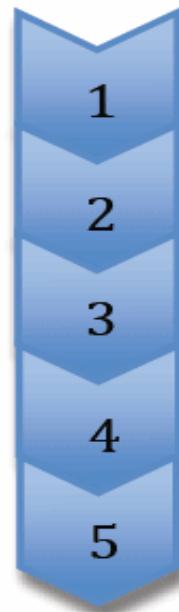
j2eedev.org/ecosystem-hadoop

Big Data Platform & Big Data Analytics Hadoop Technology



HDFS: Hadoop Distributed File System

Block Size = 64MB
Replication Factor = 3



Cost/GB is a few
¢/month vs \$/month

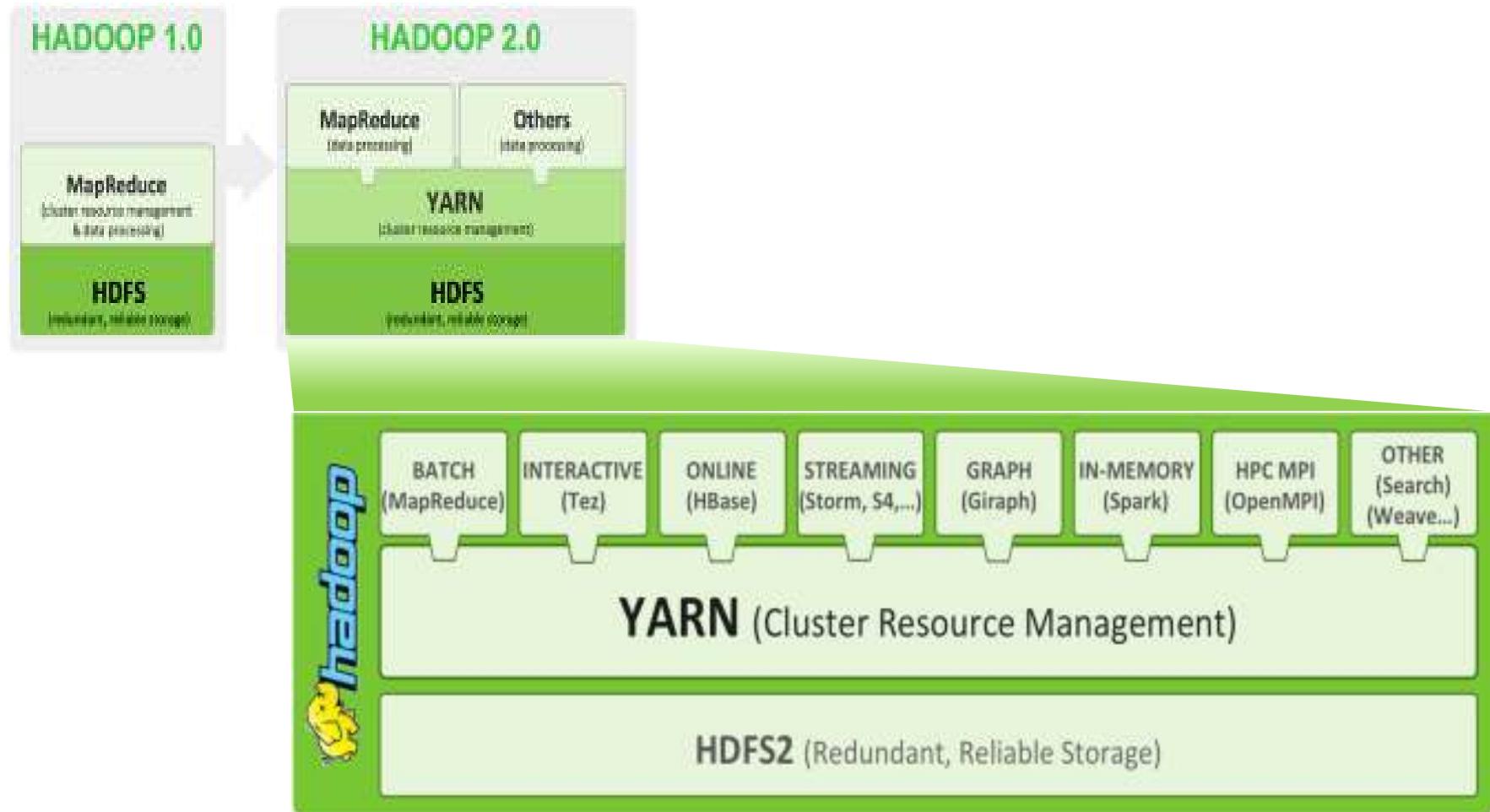
apache.org/hadoop/

Hadoop 1.0 vs Hadoop 2.0



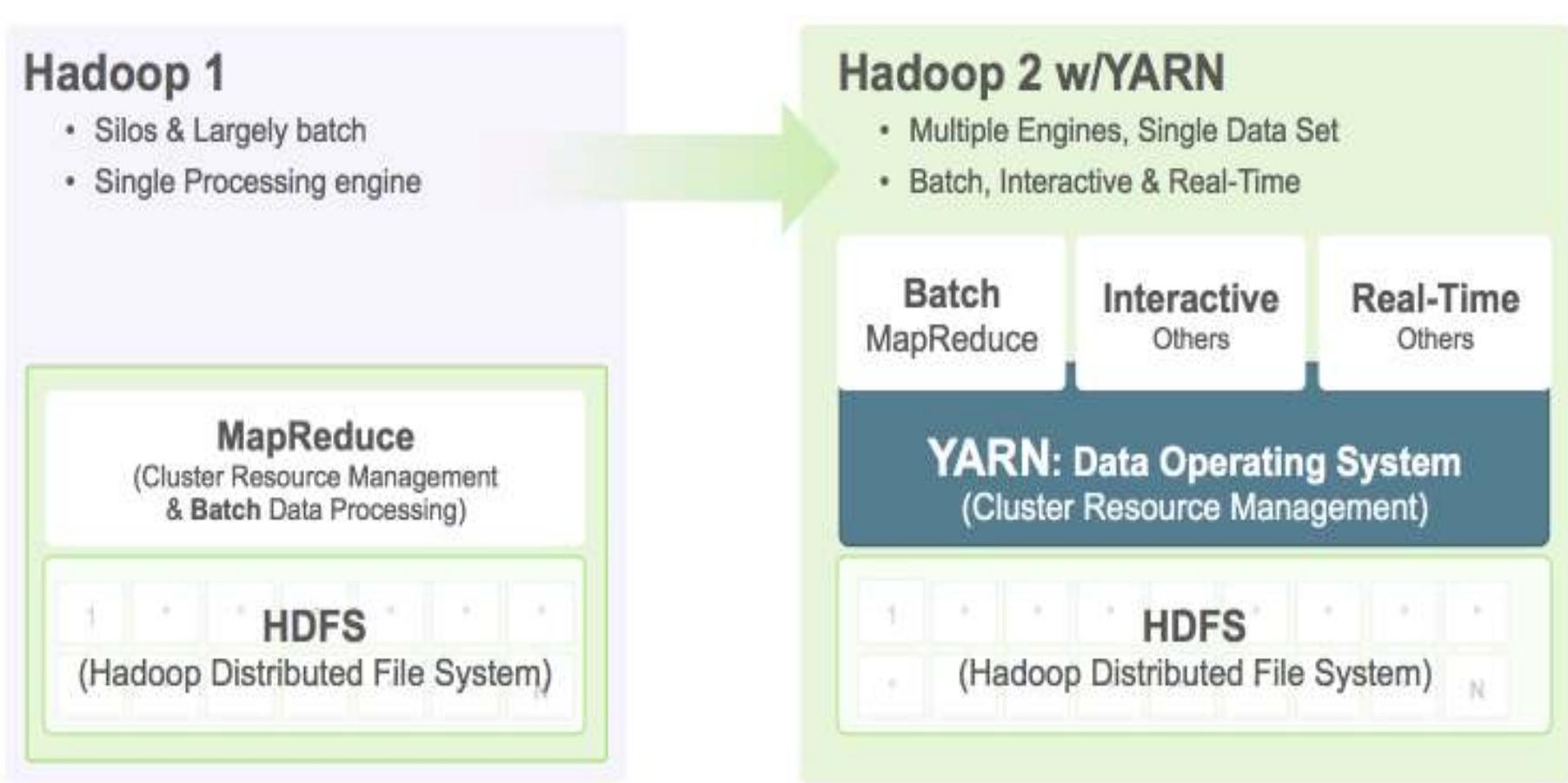
Hortonwork.com

Hadoop 1.0 vs Hadoop 2.0



Hortonwork.com

Hadoop 2



Hortonworks.com

Hadoop Symbols and Reasons Behind

Hadoop Components and Sub projects	Comparing with	Reason for the specific animal	Image
1.Hadoop distributed file system (HDFS)	Elephant	Memory of an elephant is compared with huge data storage of HDFS	
2.MapReduce	Mammoth	Mammoth means enormous, huge, massive and immense. A Mammoth's task is compared with a programming model for performing the tasks with huge volumes of data	
3.Hive	Honey Bee	Storage area of the honey in wax honeycombs inside the beehive is compared with data warehouse for storing the data in the format of table.	
4.Hbase	Horse	Running Speed of the horse indicates the real time read/write access from HBase	
5.Pig	Pig	Pigs are <u>omnivores</u> animals which means they can consume both plants and animals. This PIG consumes any type of data whether structured or unstructured or any other machine data and helps processing the same.	
6.Hue	Elephant foot prints	Elephant foot print is compared with "TREAD ON" Hadoop and explore it.	
7.Beeswax	Beeswax	Data storage happens in a structured way as honey stored in Beeswax	

Opportunity and Market Outlook

SECTOR	EXAMPLE APPLICATIONS	MAJOR DRIVER
Smart buildings	Automated monitoring of heating, ventilation and cooling	Reduced energy costs
Smart cities	Street lights that dim when roads are empty	Cost savings
Automotive	Emergency calling and accident alerts	
Leisure	Leisure vehicle and boat tracking	Safety and security
Consumer electronics	Connected satellite navigation devices to monitor traffic jams	Production innovation
Health	Remote monitoring of patients and personal health monitoring	Cheaper, home-based care
Utilities	Smart meters and energy demand response	Regulatory requirement
Transportation and logistics	Fleet optimization and supply-chain tracking and tracing	Cost savings
Retail	Wireless payments	Retail innovation
Manufacturing	Predictive maintenance through improved system monitoring	Reduced maintenance costs
Construction	Monitoring usage of equipment to improve efficiency and cut fuel usage	Cost savings
Agriculture and extraction	Remote monitoring of farm or mining operations and equipment	Proactive maintenance
Emergency services and national security	Disaster response and critical infrastructure protection	Faster response times

Source: Machina Research

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

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 Templatized 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

Hadoop Installation

Hadoop provides three installation choices:

1. **Local mode:** This is an unzip and run mode to get you started right away where all parts of Hadoop run within the same JVM
2. **Pseudo distributed mode:** This mode will be run on different parts of Hadoop as different Java processors, but within a single machine
3. **Distributed mode:** This is the real setup that spans multiple machines

Virtual Server

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

- Ubuntu Server 14.04 LTS
- m3.medium 1vCPU, 3.75 GB memory
- Security group: default
- Keypair: imchadoop

Select a EC2 service and click on Launch Instance

The screenshot shows the AWS EC2 Dashboard. On the left sidebar, under the 'INSTANCES' section, the 'Instances' link is selected. In the main content area, there's a 'Resources' summary table:

0 Running Instances	0 Elastic IPs
1 Volumes	1 Snapshots
8 Key Pairs	0 Load Balancers
0 Placement Groups	11 Security Groups

Below the table is a callout box with the text: "Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with [Elastic Beanstalk](#)".

The 'Create Instance' section contains the following text: "To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance." Below this text is a blue 'Launch Instance' button. A red arrow points to this button.

On the right side of the dashboard, there are sections for 'Account Attributes' (Supported Platforms: VPC), 'Default VPC' (vpc-cd510ca5), and 'Additional Information' (Getting Started Guide, Documentation, All EC2 Resources, Forums, Pricing, Contact Us). There's also a 'AWS Marketplace' section with links to software trials and popular AMIs.

At the bottom of the page, there are footer links for Privacy Policy and Terms of Use, and a 'Feedback' button.

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

AWS Services Edit IMC Institute Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Select

Select

Select

Image Name	Description	Type	Select Button
Amazon Linux AMI 2014.09.2 (PV) - ami-9fc29baf	The Amazon Linux AMI is an EBS backed image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Apache HTTPD, Docker, PHP, MySQL, PostgreSQL, and other packages.	64-bit	Select
SUSE Linux Enterprise Server 11 SP3 (PV), SSD Volume Type - ami-5df2ab6d	SUSE Linux Enterprise Server 11 Service Pack 3 (PV), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.	64-bit	Select
Ubuntu Server 14.04 LTS (PV), SSD Volume Type - ami-23ebb513	Ubuntu Server 14.04 LTS (PV), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).	64-bit	Select

Choose m3.medium Type virtual server

The screenshot shows the AWS CloudFormation console interface for creating a new stack. The top navigation bar includes 'AWS', 'Services', 'Edit', and account information ('IMC Institute', 'Oregon', 'Support'). Below the navigation is a progress bar with steps: 1. Choose AMI, 2. Choose Instance Type (which is underlined in blue), 3. Configure Instance, 4. Add Storage, 5. Tag Instance, 6. Configure Security Group, and 7. Review.

Step 2: Choose an Instance Type

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	Micro instances	t1.micro Free tier eligible	1	0.613	EBS only	-	Very Low
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input checked="" type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate

At the bottom of the page are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

Leave configuration details as default

The screenshot shows the AWS Launch Wizard interface for creating a new Amazon EC2 instance. The top navigation bar includes links for AWS Services, Edit, IMC Institute, Oregon, and Support. Below the navigation, a progress bar indicates Step 3: Configure Instance Details. The main section displays various configuration options:

- 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
- Monitoring:** Enable CloudWatch detailed monitoring
Additional charges apply.
- Tenancy:** Shared tenancy (multi-tenant hardware)
Additional charges will apply for dedicated tenancy.

A link to "Advanced Details" is located below the configuration options. At the bottom right are buttons for Cancel, Previous, Review and Launch (which is highlighted in blue), and Next: Add Storage.

Add Storage: 20 GB

The screenshot shows the AWS EC2 instance creation wizard at Step 4: Add Storage. The 'Size (GiB)' field for the root volume is highlighted with a red oval, containing the value '20'. The 'Volume Type' dropdown is set to 'General Purpose (SSD)'. Other visible fields include 'Delete on Termination' (unchecked), 'Encrypted' (unchecked), and 'Snapshot' (snap-0b5dab8a). Below the table, a note states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.' Navigation buttons at the bottom include 'Cancel', 'Previous', 'Review and Launch' (which is blue and bolded), and 'Next: Tag Instance'.

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0b5dab8a	20	General Purpose (SSD)	60 / 3000	<input type="checkbox"/>	Not Encrypted
Instance Store 0	/dev/sdb	N/A	N/A	N/A	N/A	<input type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Tag Instance

Name the instance

AWS Services Edit IMC Institute Oregon Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key	(127 characters maximum)	Value	(256 characters maximum)
Name		IMCInstitute Hadoop Server	<input type="button" value="X"/>

Create Tag (Up to 10 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

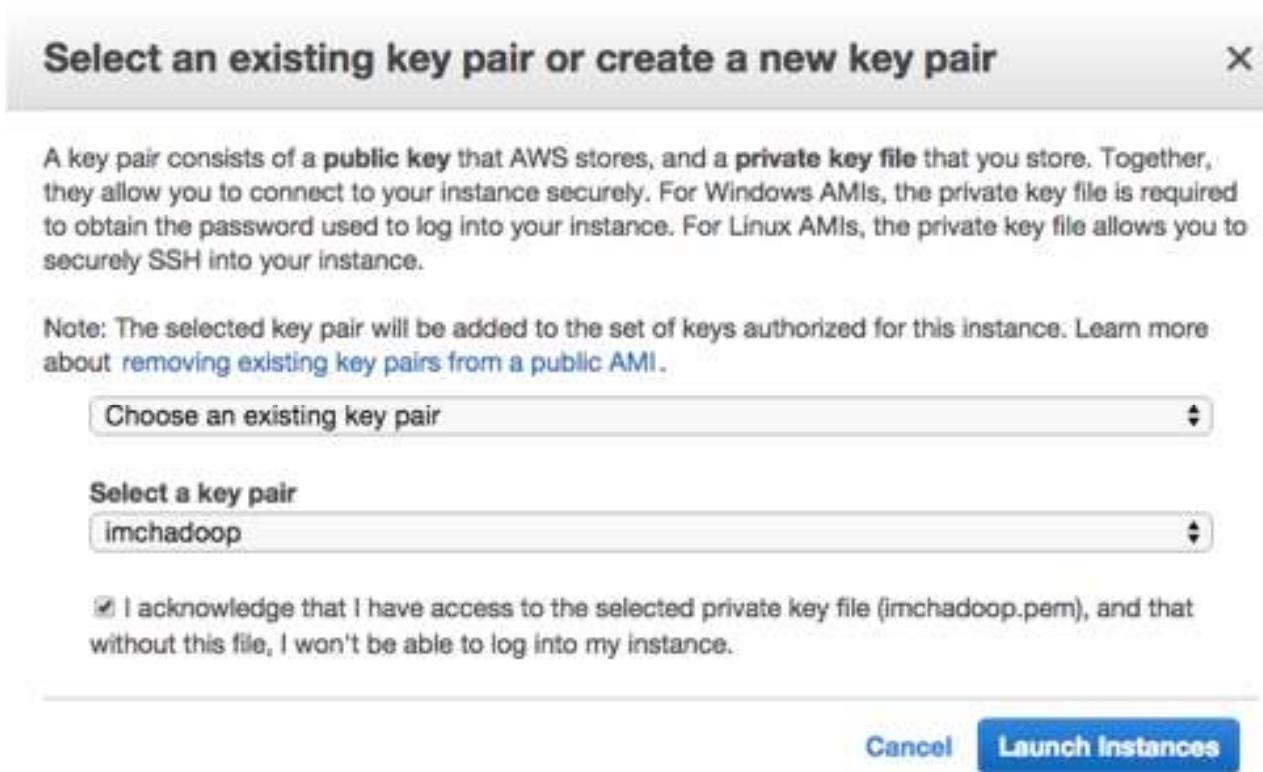
Select an existing security group > Select Security Group Name: default

The screenshot shows the AWS EC2 instance creation wizard at Step 6: Configure Security Group. The user has selected the 'Select an existing security group' option. A red circle highlights the 'default' security group in the list. The table below shows the inbound rules for the selected security group.

Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	7180	0.0.0.0/0
All TCP	TCP	0 - 65535	0.0.0.0/0

Cancel Previous Review and Launch

Click Launch and choose imchadoop as a key pair



Review an instance / click **Connect** for an instruction to connect to the instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, AMIs, Bundle Tasks, Volumes, Snapshots, Security Groups, and Elastic IPs. The 'Instances' link is currently selected. The main area has tabs for Launch Instance, Connect (which is highlighted with a red arrow), and Actions. A search bar says 'Filter by tags and attributes or search by keyword'. Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, and Status Checks. An instance named 'IMCInstitute Hadoop Ser...' is selected (indicated by a blue square icon) and circled in red. Its details are shown at the bottom: Instance ID i-2ffb89e6, Public DNS ec2-54-68-149-232.us-west-2.compute.amazonaws.com, and a status bar showing 2/2 checks passed.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Hadoop Master 01	i-d3c250df	m3.large	us-west-2b	stopped	
Cloudera Demo	i-a5b139e9	m3.large	us-west-2b	stopped	
DIW	i-233bbdce	m3.medium	us-west-2b	stopped	
IMCInstitute Hadoop Ser...	i-2ffb89e6	m3.medium	us-west-2c	running	2/2 checks passed
Cassandra	i-15475fd2	m3.medium	us-west-2c	stopped	
Hadoop Slave	i-4fcdf43	m3.medium	us-west-2b	stopped	
Hadoop Slave	i-72cd5f7e	m3.medium	us-west-2b	stopped	
Hadoop Slave	i-73cd5f7f	m3.medium	us-west-2b	stopped	

Connect to an instance from Mac/Linux

Connect To Your Instance

X

I would like to connect with A standalone SSH client A Java SSH Client directly from my browser (Java required)

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (`imchadoop.pem`). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:
`chmod 400 imchadoop.pem`
4. Connect to your instance using its Public IP:
`54.68.149.232`

Example:

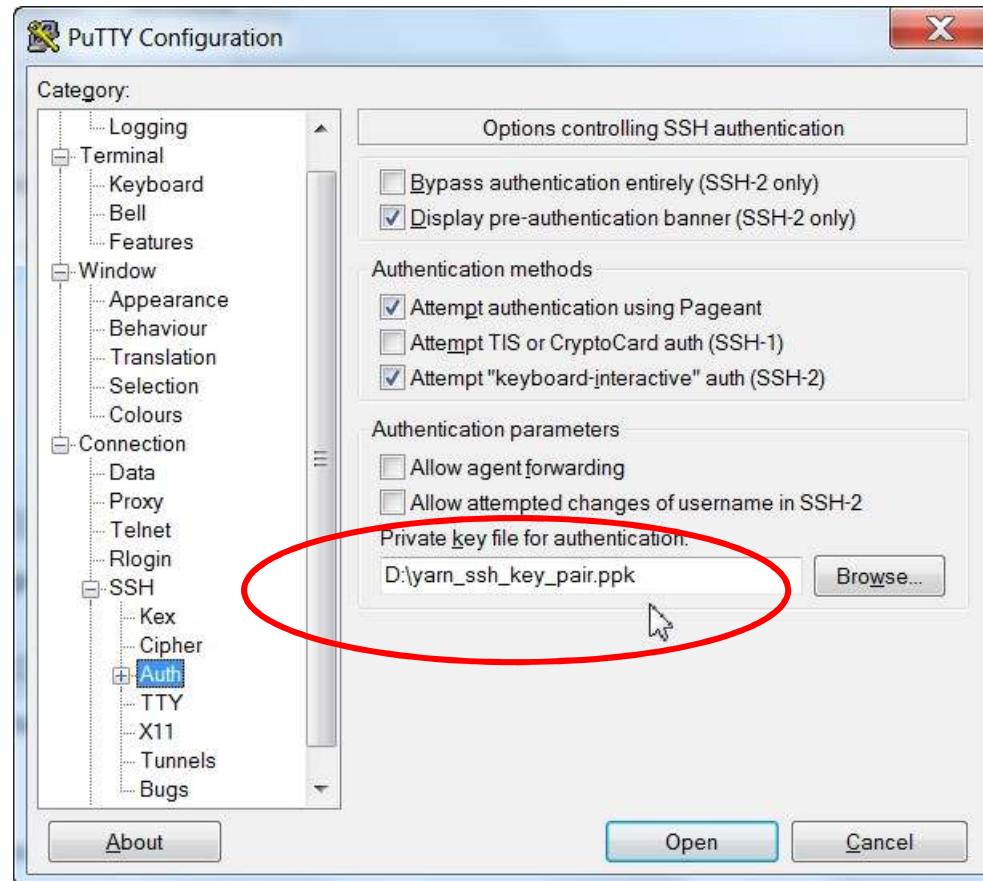
`ssh -i imchadoop.pem ubuntu@54.68.149.232`

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

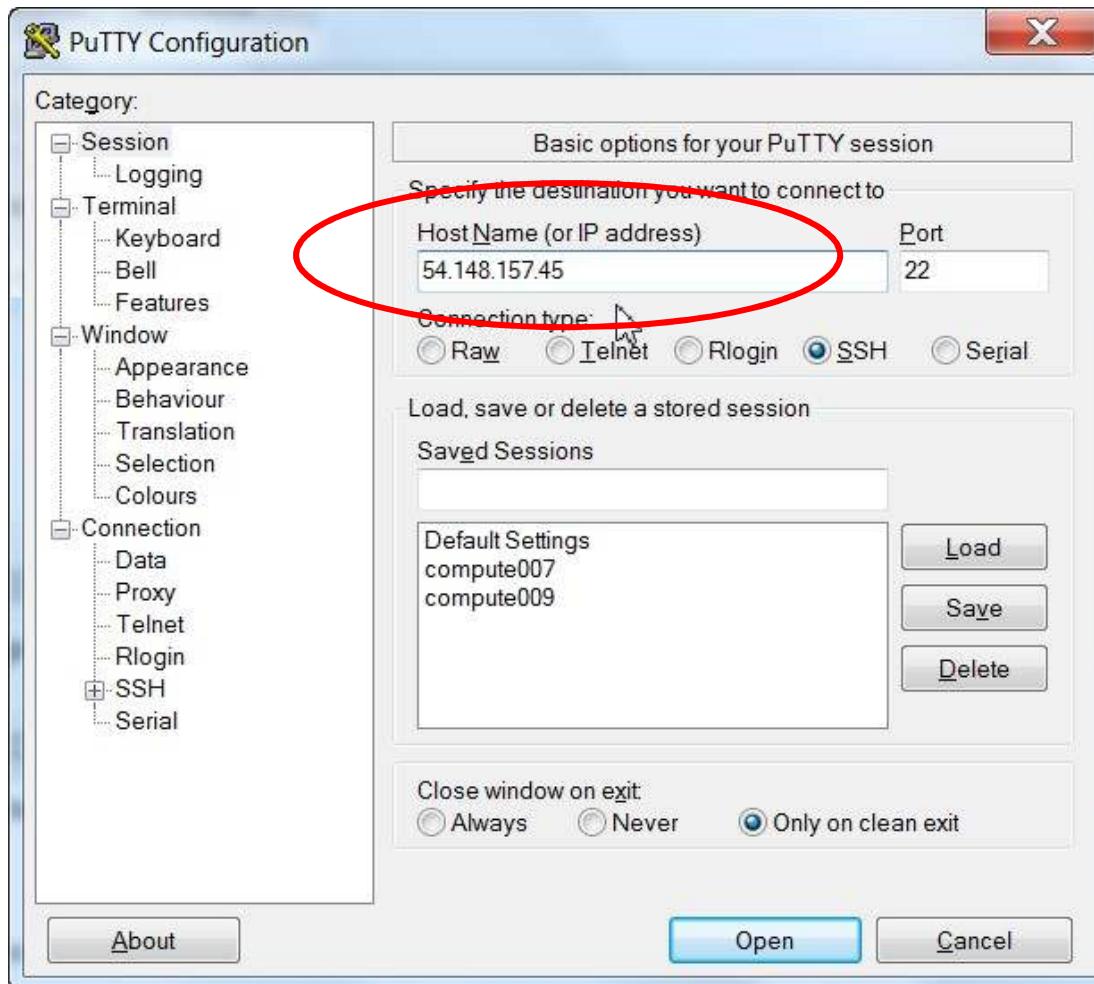
If you need any assistance connecting to your instance, please see our [connection documentation](#).

[Close](#)

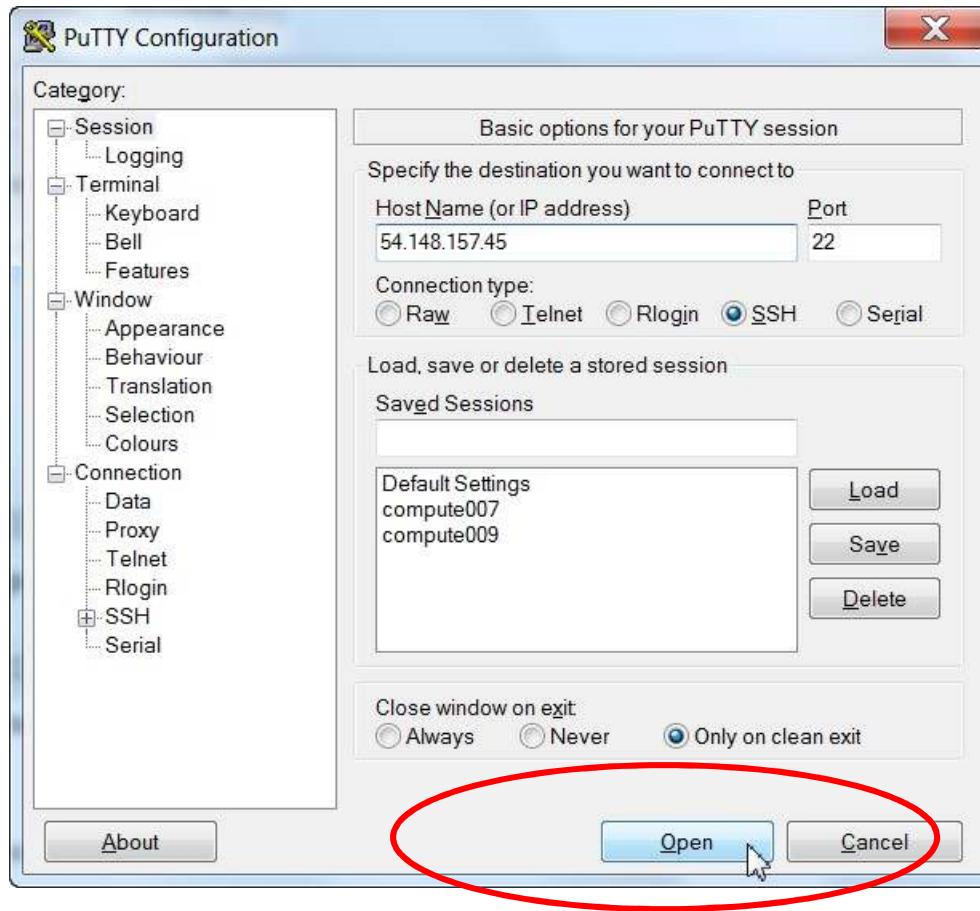
Connect to an instance from Windows using Putty



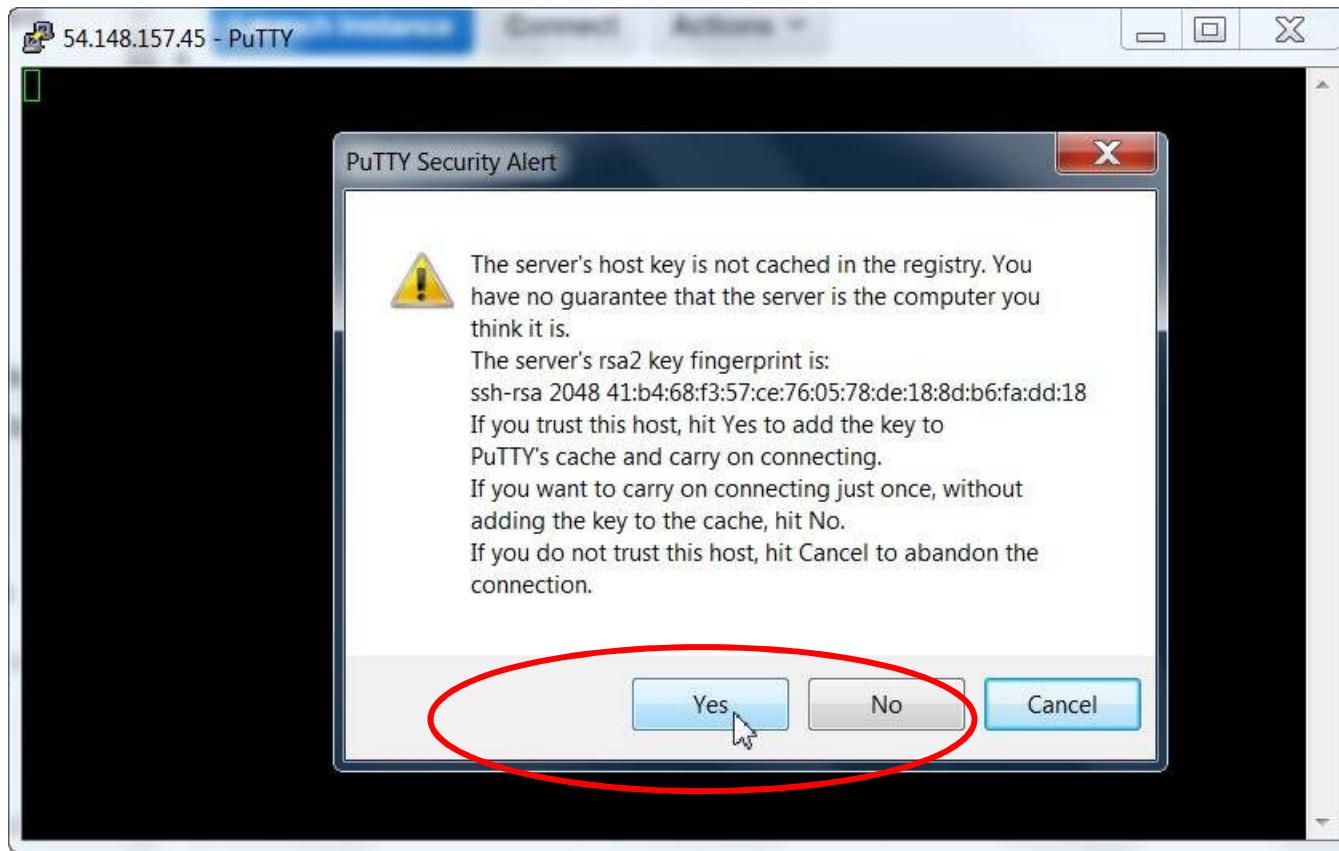
Connect to an instance from Windows using Putty



Connect to an instance from Windows using Putty



Connect to an instance from Windows using Putty



Login as: ubuntu

The screenshot shows two PuTTY windows side-by-side. The left window has a red circle around its title bar and the command line area. It displays the text "login as: ubuntu". The right window shows the system information of an Ubuntu system:

```
System information as of Thu Apr 16 09:13:11 UTC 2015

System load: 0.23          Memory usage: 1%    Processes:      48
Usage of /: 9.7% of 7.75GB Swap usage: 0%    Users logged in: 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

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

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

ubuntu@ip-172-31-4-165:~$
```

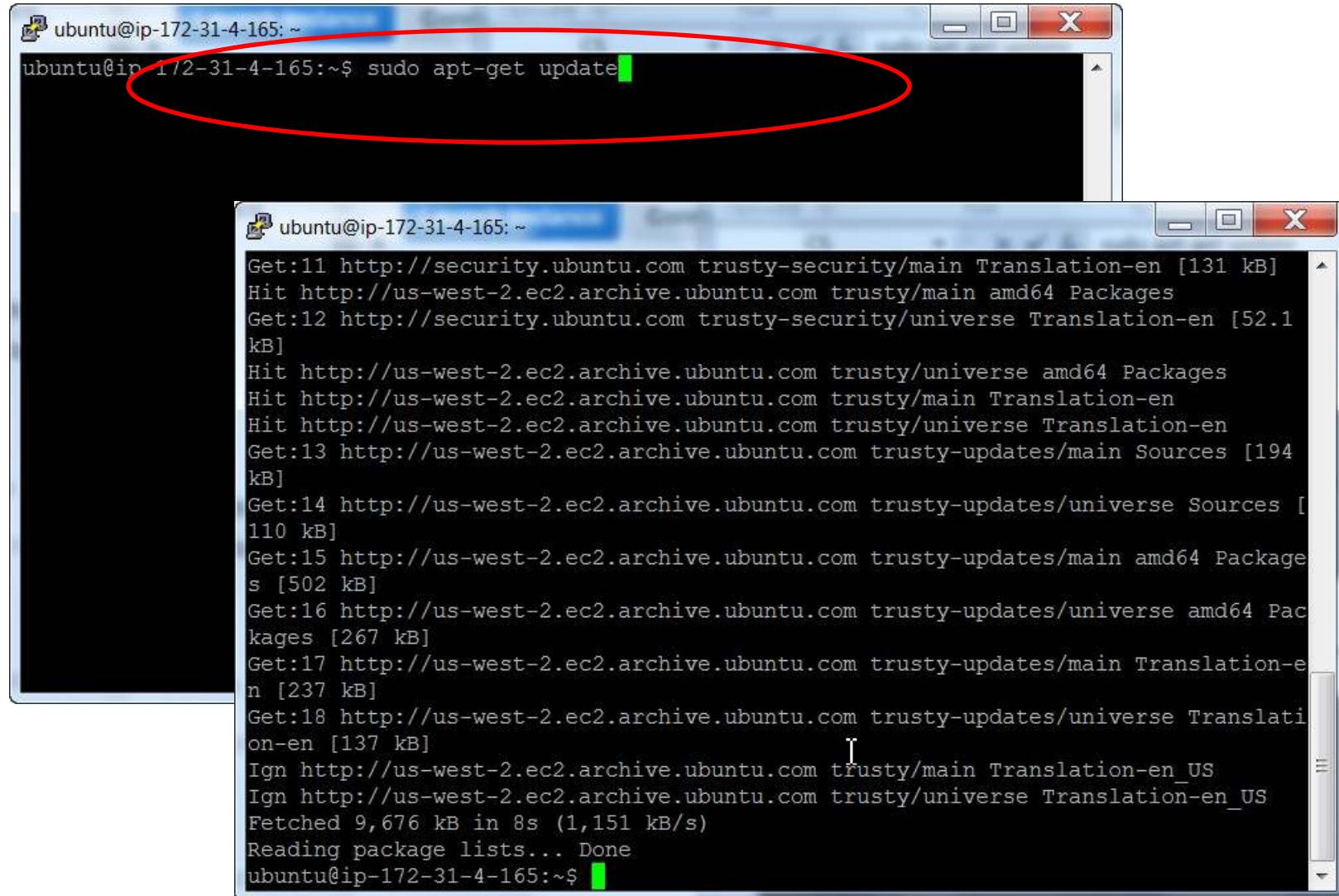
Hands-On: Installing Hadoop

Installing Hadoop and Ecosystem

- 1. Update System Software Repository**
- 2. Configuring SSH**
- 3. Installing Java**
- 4. Download/Extract Hadoop**
- 5. Installing Hadoop**
- 6. Configure Hadoop**
- 7. Formatting Namenode**
- 8. Starting Hadoop**
- 9. Accessing Hadoop Web Console**
- 10. Stopping Hadoop**

1. Update System Software Repository

sudo apt-get update



The screenshot shows two terminal windows side-by-side. The top window has a red oval highlighting the command "sudo apt-get update" in the input field. The bottom window displays the execution of the command, showing the progress of package retrieval from various Ubuntu repositories. The output includes multiple "Get:" commands for security and universe packages across different architectures (main, amd64), followed by "Ign" commands for translation files, and finally a summary message indicating the completion of the update process.

```
ubuntu@ip-172-31-4-165:~$ sudo apt-get update
Get:11 http://security.ubuntu.com trusty-security/main Translation-en [131 kB]
Hit http://us-west-2.ec2.archive.ubuntu.com trusty/main amd64 Packages
Get:12 http://security.ubuntu.com trusty-security/universe Translation-en [52.1 kB]
Hit http://us-west-2.ec2.archive.ubuntu.com trusty/universe amd64 Packages
Hit http://us-west-2.ec2.archive.ubuntu.com trusty/main Translation-en
Hit http://us-west-2.ec2.archive.ubuntu.com trusty/universe Translation-en
Get:13 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/main Sources [194 kB]
Get:14 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/universe Sources [110 kB]
Get:15 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/main amd64 Packages [502 kB]
Get:16 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/universe amd64 Packages [267 kB]
Get:17 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/main Translation-en [237 kB]
Get:18 http://us-west-2.ec2.archive.ubuntu.com trusty-updates/universe Translation-en [137 kB]
Ign http://us-west-2.ec2.archive.ubuntu.com trusty/main Translation-en_US
Ign http://us-west-2.ec2.archive.ubuntu.com trusty/universe Translation-en_US
Fetched 9,676 kB in 8s (1,151 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-4-165:~$
```

2. Configuring SSH

Install SSH, Create ssh-key

```
ubuntu@ip-172-31-4-165:~$ sudo apt-get install -y openssh-server
ubuntu@ip-172-31-4-165:~$ ssh-keygen -t dsa -P ''
ubuntu@ip-172-31-4-165:~$ ssh-keygen -t dsa -P ''
Generating public/private dsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_dsa):
ubuntu@ip-172-31-4-165:~$ ssh-keygen -t dsa -P ''
Generating public/private dsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_dsa):
Your identification has been saved in /home/ubuntu/.ssh/id_dsa.
Your public key has been saved in /home/ubuntu/.ssh/id_dsa.pub.
The key fingerprint is:
cb:e4:6d:bd:c5:e1:96:75:da:aa:9f:48:c8:2e:9a:95 ubuntu@ip-172-31-4-1
The key's randomart image is:
+--[ DSA 1024]----+
| |
| |
| |
| |
| S . o |
| + = o o * |
| E = o B . |
```

cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys

The image shows three terminal windows from a Linux desktop environment. The top window shows the command `cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys` being run. The middle window shows the SSH connection attempt to localhost, where it asks for confirmation to proceed due to unverified host fingerprints. The bottom window shows the successful SSH login to the local machine, displaying system information and package update status.

```
ubuntu@ip-172-31-4-165:~$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
ubuntu@ip-172-31-4-165:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 56:f2:7e:b4:48:fc:80:d9:00:98:08:66:7b:5f:96:ea.
Are you sure you want to continue connecting (yes/no)? yes
ubuntu@ip-172-31-4-165:~$ Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-48-generic x86_64)

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

 System information as of Thu Apr 16 09:15:42 UTC 2015

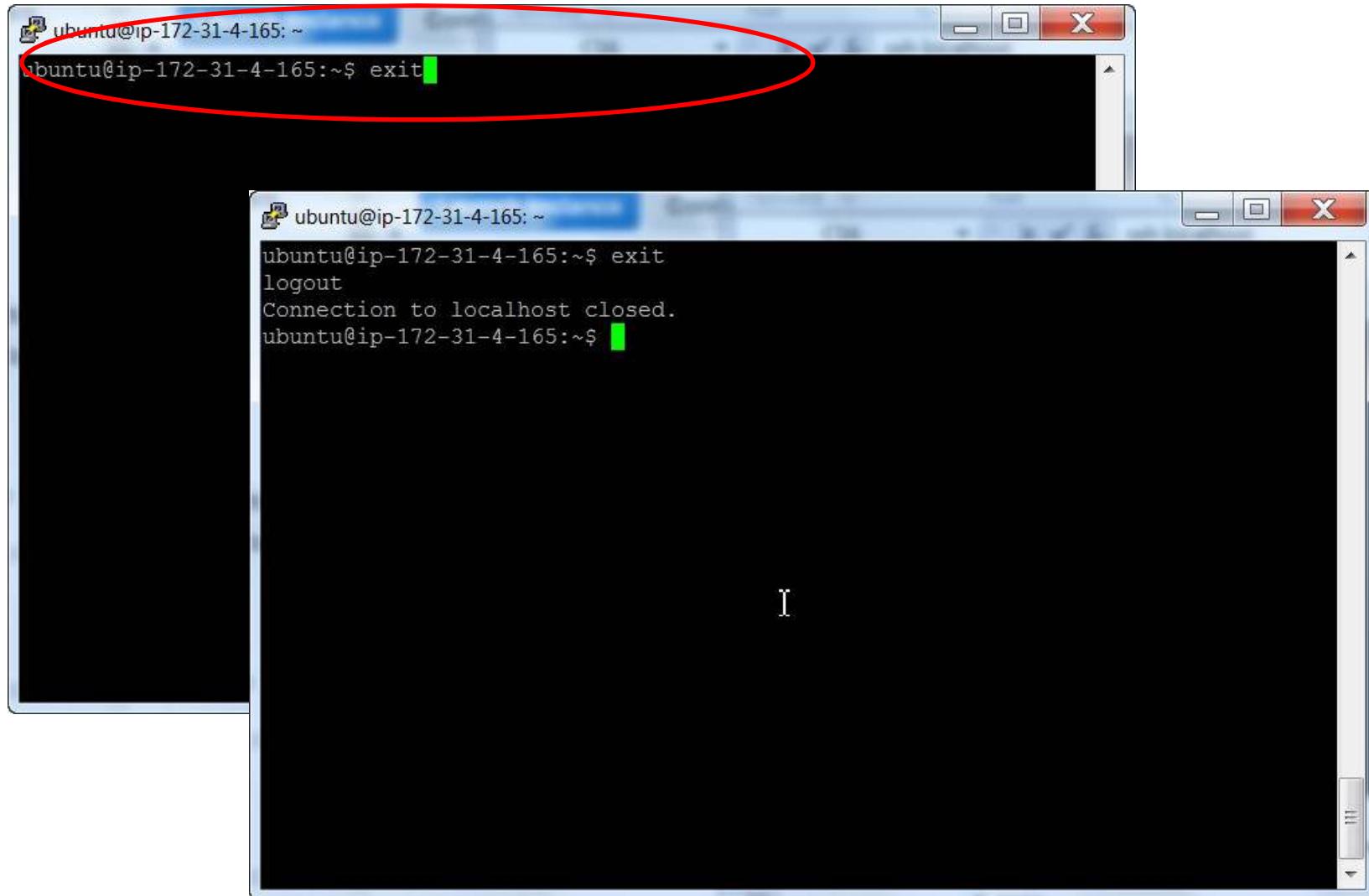
 System load:  0.12           Processes:      67
 Usage of /:   9.8% of 7.75GB  Users logged in:  0
 Memory usage: 1%            IP address for eth0: 172.31.4.165
 Swap usage:   0%

 Graph this data and manage this system at:
 https://landscape.canonical.com/
 [I] Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

 0 packages can be updated.
 0 updates are security updates.

 Last login: Thu Apr 16 09:15:51 2015 from ppp-61-90-105-6.revip.asianet.co.th
ubuntu@ip-172-31-4-165:~$
```

Exit from ssh connection



3. Installing Java

Installing Java

The screenshot shows a terminal window with three distinct sections of text output:

- Section 1 (Top):** The command `sudo apt-get install -y openjdk-7-jdk` is entered by the user.
- Section 2 (Middle):** The terminal displays the progress of certificate downloads, listing several .pem files being added.
- Section 3 (Bottom):** The command `java -version` is run to verify the Java version, which returns "java version "1.7.0_75"" followed by details about the OpenJDK Runtime Environment and Server VM.

Red circles highlight the first command and the resulting Java version output.

```
ubuntu@ip-172-31-4-165:~$ sudo apt-get install -y openjdk-7-jdk
Adding debian:AddTrust_Qualified_Certificates_Root.pem
Adding debian:Digital_Signature_Trust_Co_Global_CA_3.pem
Adding debian:Network_Solutions_Certificate_Authority.pem
Adding debian:
Adding debian:
Adding debian:
Adding debian:
ubuntu@ip-172-31-4-165:~$ java -version
java version "1.7.0_75"
OpenJDK Runtime Environment (IcedTea 2.5.4) (7u75-2.5.4-1~trusty1)
OpenJDK 64-Bit Server VM (build 24.75-b04, mixed mode)
ubuntu@ip-172-31-4-165:~$
```

4. Download/Extract Hadoop

Download/Extract Hadoop

```
ubuntu@ip-172-31-4-165:~$ wget http://mirror.cc.columbia.edu/pub/software/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
```

```
ubuntu@ip-172-31-4-165:~$ wget http://mirror.cc.columbia.edu/pub/software/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
--2015-04-16 09:25:31 -- http://mirror.cc.columbia.edu/pub/software/apache/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz
Resolving mirror.cc.columbia.edu (mirror.cc.columbia.edu)... 128.111.12.12
Connecting to mirror.cc.columbia.edu (mirror.cc.columbia.edu)|128.111.12.12|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 195257604 (192M)
Saving to: 'hadoop-2.6.0.tar.gz'

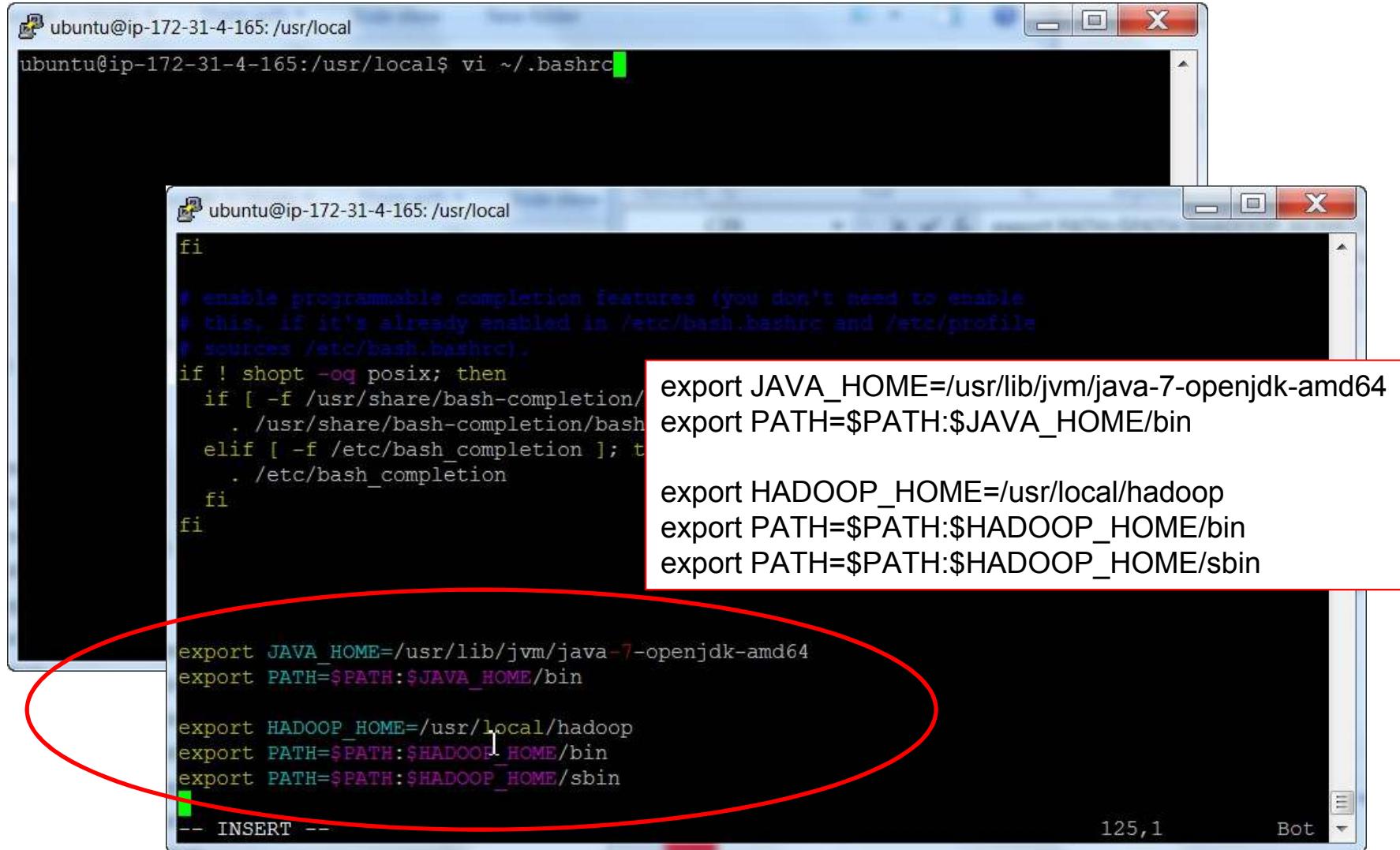
100% [=====] 2015-04-16 09:25:51
04]
```

```
ubuntu@ip-172-31-4-165:~$ tar -xvf hadoop-2.6.0.tar.gz
```

```
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-2.6.0-test-sources.jar
hadoop-2.6.0/share/hadoop/mapreduce/sources/hadoop-mapreduce-client-hs-2.6.0-test-sources.jar
hadoop-2.6.0/LICENSE.txt
hadoop-2.6.0/README.txt
hadoop-2.6.0/bin/
hadoop-2.6.0/bin/hdfs.cmd
hadoop-2.6.0/bin/test-container-executor
hadoop-2.6.0/bin/container-executor
hadoop-2.6.0/bin/hadoop.cmd
hadoop-2.6.0/bin/rcc
hadoop-2.6.0/bin/hdfs
hadoop-2.6.0/bin/mapred
hadoop-2.6.0/bin/hadoop
hadoop-2.6.0/bin/yarn.cmd
hadoop-2.6.0/bin/mapred.cmd
hadoop-2.6.0/bin/yarn
hadoop-2.6.0/include/
hadoop-2.6.0/include/TemplateFactory.hh
hadoop-2.6.0/include/StringUtils.hh
hadoop-2.6.0/include/hdfs.h
hadoop-2.6.0/include/Pipes.hh
hadoop-2.6.0/include/SerialUtils.hh
```

5. Installing Hadoop

Installing Hadoop



```
ubuntu@ip-172-31-4-165: /usr/local
ubuntu@ip-172-31-4-165:/usr/local$ vi ~/.bashrc
```

```
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/
        . /usr/share/bash-completion/bash
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

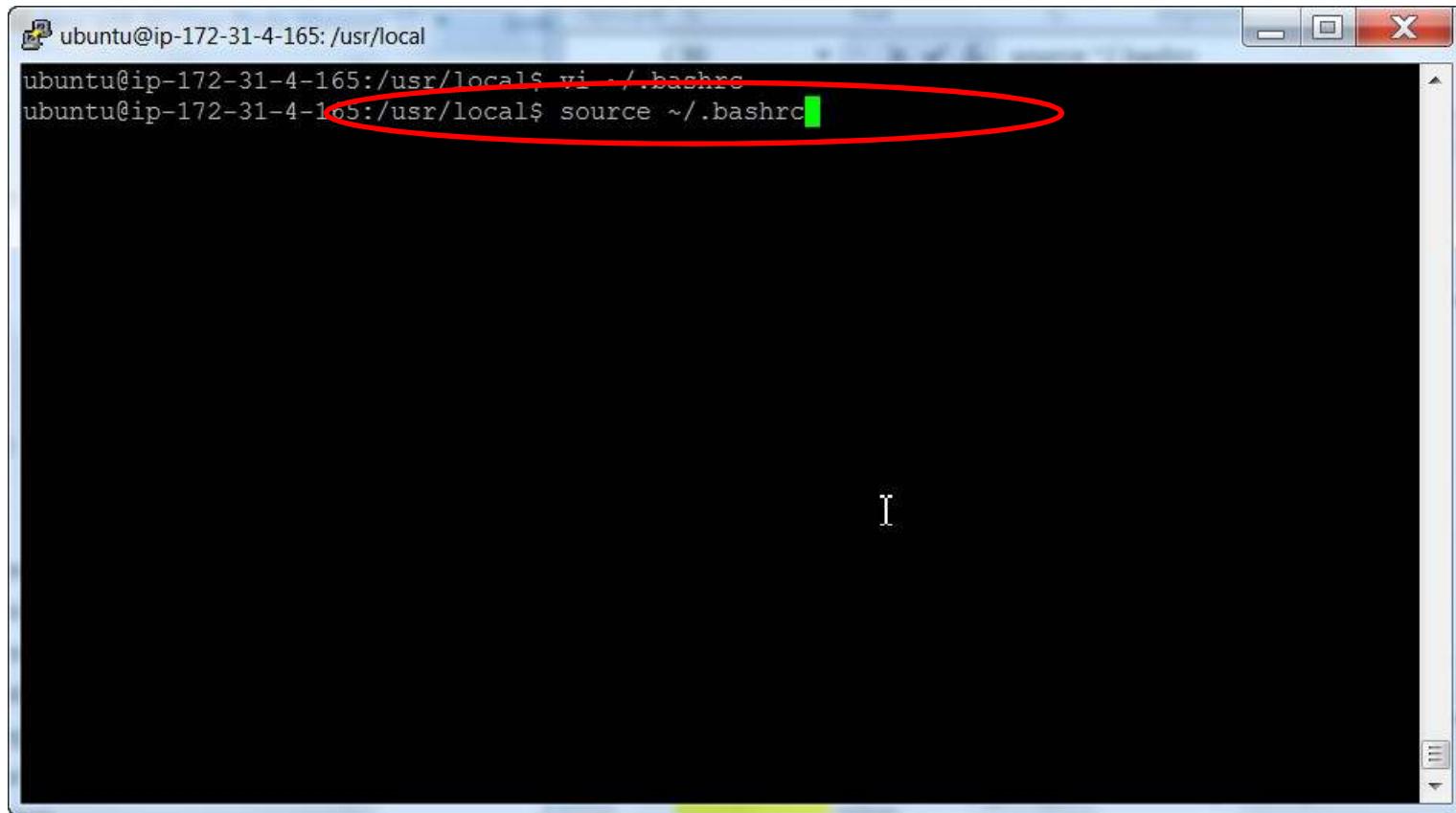
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

-- INSERT --
```

125,1 Bot

Execute environment variables: source ~/.bashrc



```
ubuntu@ip-172-31-4-165: /usr/local
ubuntu@ip-172-31-4-165:/usr/local$ vi ~/.bashrc
ubuntu@ip-172-31-4-165:/usr/local$ source ~/.bashrc
```

A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: /usr/local". The window contains two lines of text: "vi ~/.bashrc" and "source ~/.bashrc". The second line, "source ~/.bashrc", is circled with a red oval.

Edit hadoop shell script

The image shows two terminal windows side-by-side. The top window has a red circle around the command `vi hadoop-env.sh`. The bottom window has a red circle around the line `export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64`.

```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local$ vi ~/.bashrc
ubuntu@ip-172-31-4-165:/usr/local$ source ~/.bashrc
ubuntu@ip-172-31-4-165:/usr/local$ cd /usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ vi hadoop-env.sh

# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# Set Hadoop-specific environment variables here.

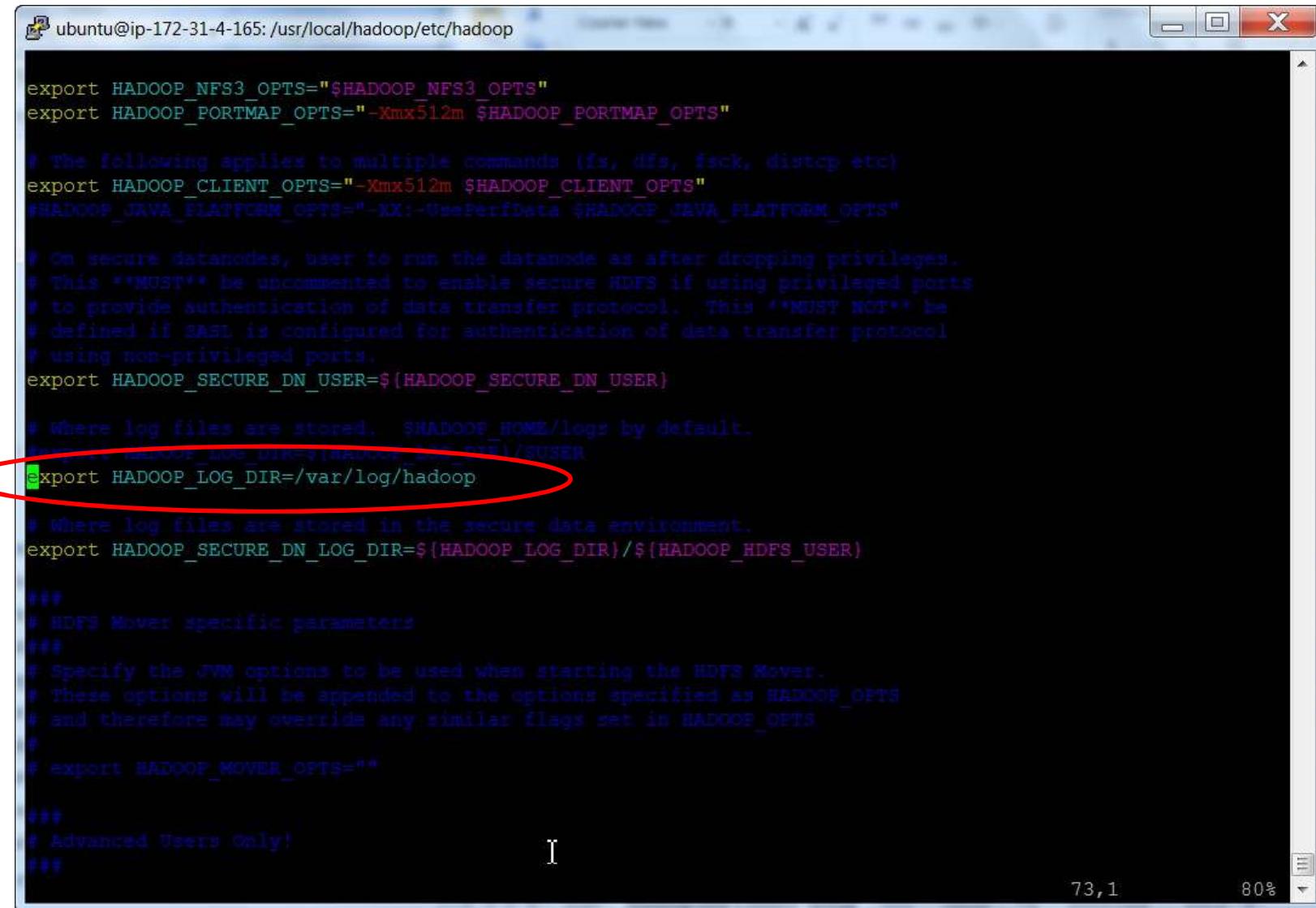
# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes

# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

# The java implementation to use. Java is required to run secure datanodes
# that bind to privileged ports to provide authentication of data transfer
# protocol. Jsvc is not required if SASL is configured for authentication of
# data transfer protocol using non-privileged ports.
#export JSVC_HOME=$JAVA_HOME

export HADOOP_CONF_DIR=${HADOOP_CONF_DIR:-"/etc/hadoop"}
```

Edit hadoop shell script – log location to another directory



```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop

export HADOOP_NFS3_OPTS="$HADOOP_NFS3_OPTS"
export HADOOP_PORTMAP_OPTS="-Xmx512m $HADOOP_PORTMAP_OPTS"

# The following applies to multiple commands (fs, dfs, fsck, distcp etc)
export HADOOP_CLIENT_OPTS="-Xmx512m $HADOOP_CLIENT_OPTS"
#HADOOP_JAVA_PLATFORM_OPTS="-XX:-UsePerfData $HADOOP_JAVA_PLATFORM_OPTS"

# On secure datanodes, user to run the datanode as after dropping privileges.
# This **MUST** be uncommented to enable secure HDFS if using privileged ports
# to provide authentication of data transfer protocol. This **MUST NOT** be
# defined if SASL is configured for authentication of data transfer protocol
# using non-privileged ports.
export HADOOP_SECURE_DN_USER=${HADOOP_SECURE_DN_USER}

# Where log files are stored. $HADOOP_HOME/logs by default.
#export HADOOP_LOG_DIR=$HADOOP_HOME/logs/${HADOOP_SECURE_DN_USER}
export HADOOP_LOG_DIR=/var/log/hadoop

# Where log files are stored in the secure data environment.
#export HADOOP_SECURE_DN_LOG_DIR=${HADOOP_LOG_DIR}/${HADOOP_HDFS_USER}

### 
# HDFS Mover specific parameters
###
# Specify the JVM options to be used when starting the HDFS Mover.
# These options will be appended to the options specified as HADOOP_OPTS
# and therefore may override any similar flags set in HADOOP_OPTS
#
# export HADOOP_MOVER_OPTS=""

###
# Advanced Users Only!
###

[ 73,1 80%]
```

Edit Yarn shell script – log location to another directory

The image shows two terminal windows side-by-side. The top window has a red circle around the command `vi yarn-env.sh`. The bottom window shows the contents of the `yarn-env.sh` file, which is a shell script. A red circle highlights the line `export YARN_LOG_DIR=/var/log/hadoop`.

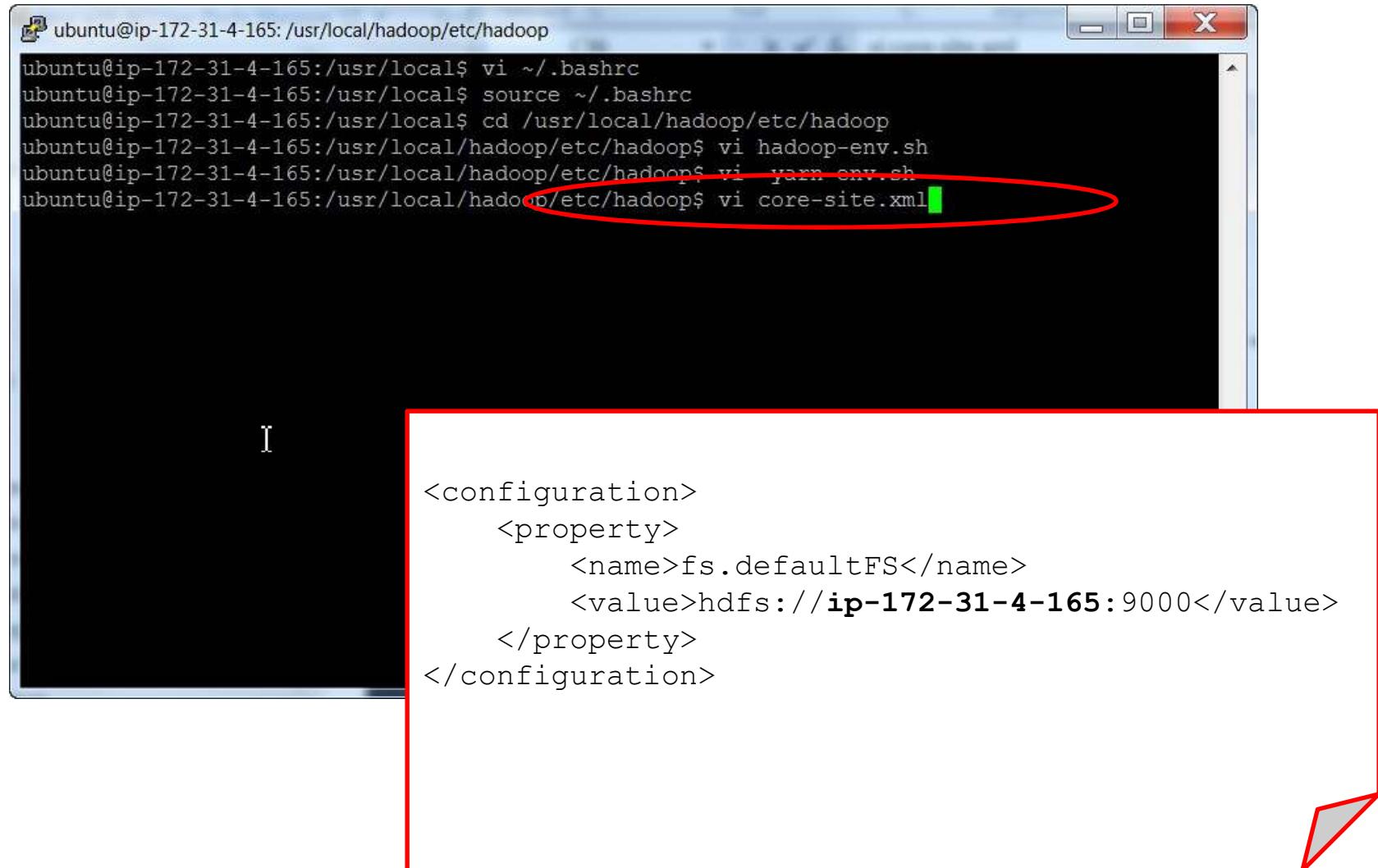
```
# restore ordinary behaviour
unset IFS

YARN_OPTS="$YARN_OPTS -Dhadoop.log.dir=$YARN_LOG_DIR"
YARN_OPTS="$YARN_OPTS -Dyarn.log.dir=$YARN_LOG_DIR"
YARN_OPTS="$YARN_OPTS -Dhadoop.log.file=$YARN_LOGFILE"
YARN_OPTS="$YARN_OPTS -Dyarn.log.file=$YARN_LOGFILE"
YARN_OPTS="$YARN_OPTS -Dyarn.home.dir=$YARN_COMMON_HOME"
YARN_OPTS="$YARN_OPTS -Dyarn.id.str=$YARN_IDENT_STRING"
YARN_OPTS="$YARN_OPTS -Dhadoop.root.logger=${YARN_ROOT_LOGGER:-INFO,console}"
YARN_OPTS="$YARN_OPTS -Dyarn.root.logger=${YARN_ROOT_LOGGER:-INFO,console}"
if [ "x$JAVA_LIBRARY_PATH" != "x" ]; then
  YARN_OPTS="$YARN_OPTS -Djava.library.path=$JAVA_LIBRARY_PATH"
fi
YARN_OPTS="$YARN_OPTS -Dyarn.policy.file=$YARN_POLICYFILE"

export YARN_LOG_DIR=/var/log/hadoop
```

6. Configuring Hadoop

Edit hadoop - core-site.xml



The screenshot shows a terminal window on an Ubuntu system. The user has run several commands to set up the environment:

```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165: /usr/local$ vi ~/.bashrc
ubuntu@ip-172-31-4-165: /usr/local$ source ~/.bashrc
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop$ vi hadoop-env.sh
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop$ vi yarn-env.sh
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop$ vi core-site.xml
```

The last command, `vi core-site.xml`, is highlighted with a red oval. A large red rectangle highlights the XML configuration code that follows:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://ip-172-31-4-165:9000</value>
  </property>
</configuration>
```

Creating directories for namenode and datanode

```
sudo mkdir -p /var/hadoop_data/namenode  
sudo mkdir -p /var/hadoop_data/datanode  
sudo chown ubuntu:ubuntu -R /var/hadoop_data
```

Edit hdfs-site.xml

```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ vi hdfs-site.xml
```

```
<!-- Put site-specific property overrides in this file. -->  
  
<configuration>  
  <property>  
    <name>dfs.replication</name>  
    <value>1</value>  
  </property>  
  <property>  
    <name>dfs.namenode.name.dir</name>  
    <value>file:/var/hadoop_data/namenode</value>  
  </property>  
  <property>  
    <name>dfs.datanode.data.dir</name>  
    <value>file:/var/hadoop_data/datanode</value>  
  </property>  
</configuration>  
-- INSERT --
```

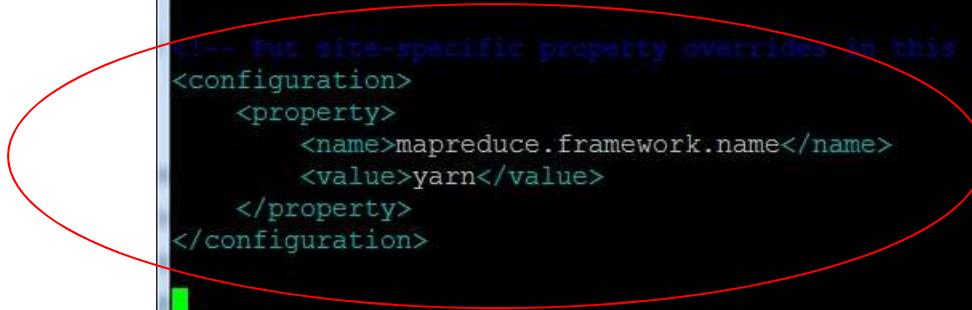
Edit yarn-site.xml

```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ vi yarn-site.xml
```

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>ip-172-31-4-165</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>ip-172-31-4-165:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>ip-172-31-4-165:8031</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>ip-172-31-4-165:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>ip-172-31-4-165:8033</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>ip-172-31-4-165:8088</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

Edit mapred-site.xml

```
cp mapred-site.xml.template mapred-site.xml  
vi mapred-site.xml
```



```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop  
<?xml version="1.0"?>  
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>  
<!--  
 Licensed under the Apache License, Version 2.0 (the "License");  
 you may not use this file except in compliance with the License.  
 You may obtain a copy of the License at  
  
     http://www.apache.org/licenses/LICENSE-2.0  
  
 Unless required by applicable law or agreed to in writing, software  
 distributed under the License is distributed on an "AS IS" BASIS,  
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 See the License for the specific language governing permissions and  
 limitations under the License. See accompanying LICENSE file.  
-->  
<!-- Put site-specific property overrides in this file. -->  
<configuration>  
    <property>  
        <name>mapreduce.framework.name</name>  
        <value>yarn</value>  
    </property>  
</configuration>
```

7. Formatting Namenode

Formatting Namenode

The screenshot shows two terminal windows. The top window has a red circle around the command `hdfs namenode -format`. The bottom window shows the execution of this command and its output, which includes configuration settings for the BlockManager and FSNamesystem, and a confirmation prompt at the end.

```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ hdfs namenode -format
```



```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop
15/04/16 10:08:31 INFO blockmanagement.BlockManager: maxReplication      = 512
15/04/16 10:08:31 INFO blockmanagement.BlockManager: minReplication     = 1
15/04/16 10:08:31 INFO blockmanagement.BlockManager: maxReplicationStreams = 2
15/04/16 10:08:31 INFO blockmanagement.BlockManager: shouldCheckForEnoughRacks = false
15/04/16 10:08:31 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
15/04/16 10:08:31 INFO blockmanagement.BlockManager: encryptDataTransfer    = false
15/04/16 10:08:31 INFO blockmanagement.BlockManager: maxNumBlocksToLog     = 1000
15/04/16 10:08:31 INFO namenode.FSNamesystem: fsOwner          = ubuntu (auth:SIMPLE)
15/04/16 10:08:31 INFO namenode.FSNamesystem: supergroup       = supergroup
15/04/16 10:08:31 INFO namenode.FSNamesystem: isPermissionEnabled = true
15/04/16 10:08:31 INFO namenode.FSNamesystem: HA Enabled       = false
15/04/16 10:08:31 INFO namenode.FSNamesystem: Append Enabled    = true
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map INodeMap
15/04/16 10:08:31 INFO util.GSet: VM type           = 64-bit
15/04/16 10:08:31 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 MB
15/04/16 10:08:31 INFO util.GSet: capacity        = 2^20 = 1048576 entries
15/04/16 10:08:31 INFO namenode.NameNode: Caching file names occurring more than 10 times
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map cachedBlocks
15/04/16 10:08:31 INFO util.GSet: VM type           = 64-bit
15/04/16 10:08:31 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
15/04/16 10:08:31 INFO util.GSet: capacity        = 2^18 = 262144 entries
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension   = 30000
15/04/16 10:08:31 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
15/04/16 10:08:31 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map NameNodeRetryCache
15/04/16 10:08:31 INFO util.GSet: VM type           = 64-bit
15/04/16 10:08:31 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
15/04/16 10:08:31 INFO util.GSet: capacity        = 2^15 = 32768 entries
15/04/16 10:08:31 INFO namenode.NNConf: ACLs enabled? false
15/04/16 10:08:31 INFO namenode.NNConf: XAttrs enabled? true
15/04/16 10:08:31 INFO namenode.NNConf: Maximum size of an xattr: 16384
Re-format filesystem in Storage Directory /var/hadoop_data/namenode ? (Y or N) y
```

Formatting Namenode

```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop
15/04/16 10:08:31 INFO namenode.FSNamesystem: Append Enabled: true
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map INodeMap
15/04/16 10:08:31 INFO util.GSet: VM type      = 64-bit
15/04/16 10:08:31 INFO util.GSet: 1.0% max memory 966.7 MB = 9.7 MB
15/04/16 10:08:31 INFO util.GSet: capacity      = 2^20 = 1048576 entries
15/04/16 10:08:31 INFO namenode.NameNode: Caching file names occurring more than 10 times
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map cachedBlocks
15/04/16 10:08:31 INFO util.GSet: VM type      = 64-bit
15/04/16 10:08:31 INFO util.GSet: 0.25% max memory 966.7 MB = 2.4 MB
15/04/16 10:08:31 INFO util.GSet: capacity      = 2^18 = 262144 entries
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
15/04/16 10:08:31 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension      = 30000
15/04/16 10:08:31 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
15/04/16 10:08:31 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
15/04/16 10:08:31 INFO util.GSet: Computing capacity for map NameNodeRetryCache
15/04/16 10:08:31 INFO util.GSet: VM type      = 64-bit
15/04/16 10:08:31 INFO util.GSet: 0.029999999329447746% max memory 966.7 MB = 297.0 KB
15/04/16 10:08:31 INFO util.GSet: capacity      = 2^15 = 32768 entries
15/04/16 10:08:31 INFO namenode.NNConf: ACLs enabled? false
15/04/16 10:08:31 INFO namenode.NNConf: XAttrs enabled? true
15/04/16 10:08:31 INFO namenode.NNConf: Maximum size of an xattr: 16384
Re-format filesystem in Storage Directory /var/hadoop_data/namenode ? (Y or N) Y
15/04/16 10:09:00 INFO namenode.FSImage: Allocated new BlockPoolId: BP-721065191-172.31.4.165-1429178940
099
15/04/16 10:09:00 INFO common.Storage: Storage directory /var/hadoop_data/namenode has been successfully formatted.
15/04/16 10:09:00 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
15/04/16 10:09:00 INFO util.ExitUtil: Exiting with status 0
15/04/16 10:09:00 INFO namenode.NameNode: SHUTDOWN_MSG:
*****SHUTDOWN_MSG: Shutting down NameNode at ip-172-31-4-165.us-west-2.compute.internal/172.31.4.165
*****ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$
```

8. Starting Hadoop

Starting Namenode and Datanode

```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
```

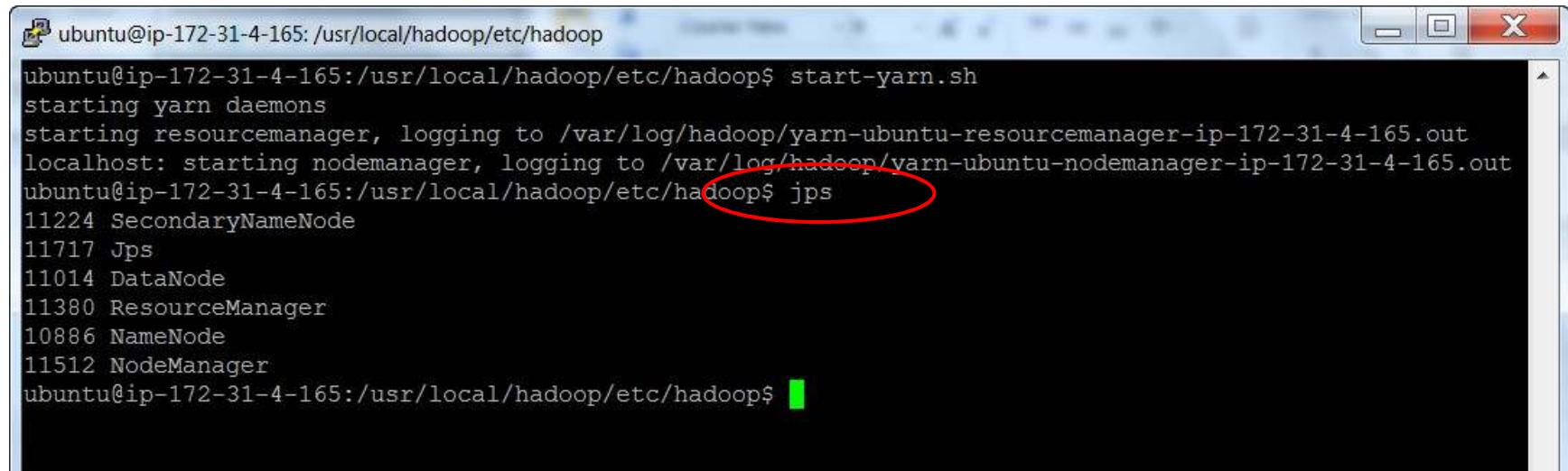
```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [ip-172-31-4-165]
ip-172-31-4-165: starting namenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-165.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is 56:f2:7e:b4:48:fc:80:d9:00:98:06:66:7b:5f:96:ea.
Are you sure you want to continue connecting (yes/no)? yes
```

```
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ start-dfs.sh
Starting namenodes on [ip-172-31-4-165]
ip-172-31-4-165: starting namenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-165.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is 56:f2:7e:b4:48:fc:80:d9:00:98:06:66:7b:5f:96:ea.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-165.out
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ jps
11224 SecondaryNameNode
11014 DataNode
11337 Jps
10886 NameNode
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$
```

Starting Yarn



```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
```

A screenshot of a terminal window on a Linux system. The window title is 'Terminal'. The command 'start-yarn.sh' is being typed into the terminal. A red oval highlights the command line.

```
ubuntu@ip-172-31-4-165: /usr/local/hadoop/etc/hadoop
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /var/log/hadoop/yarn/ubuntu-resourcemanager-ip-172-31-4-165.out
localhost: starting nodemanager, logging to /var/log/hadoop/yarn/ubuntu-nodemanager-ip-172-31-4-165.out
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ jps
11224 SecondaryNameNode
11717 Jps
11014 DataNode
11380 ResourceManager
10886 NameNode
11512 NodeManager
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$
```

A screenshot of a terminal window on a Windows system. The window title is 'Terminal'. The command 'start-yarn.sh' is run, followed by 'jps' to list running Java processes. A red oval highlights the command line.

9. Accessing Hadoop Web Console

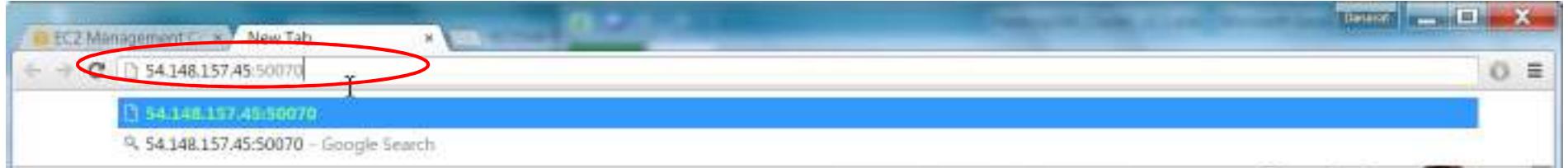
Checking Public IP Address

The screenshot shows the AWS EC2 Management Console interface. The left sidebar navigation includes 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', 'INSTANCES' (selected), 'Instances', 'Spot Requests', 'Reserved Instances', 'IMAGES' (AMIs), 'Bundle Tasks', 'ELASTIC BLOCK STORE' (Volumes, Snapshots), and 'NETWORK & SECURITY' (Security Groups, Elastic IPs, Placement Groups). The main content area displays a search bar with 'search: i-c6134c0f' and a table of instances. One instance is listed: 'hpd26_psud...' with Instance ID 'i-c6134c0f', Instance Type 'm3.medium', Availability Zone 'us-west-2c', Instance State 'running', Status Checks '2/2 checks', and Alarm Status 'None'. Below this, a detailed view for 'Instance: i-c6134c0f (hpd26_psudo001)' shows its Public DNS as 'ec2-54-148-157-45.us-west-2.compute.amazonaws.com'. The 'Description' tab of the detailed view shows the following information:

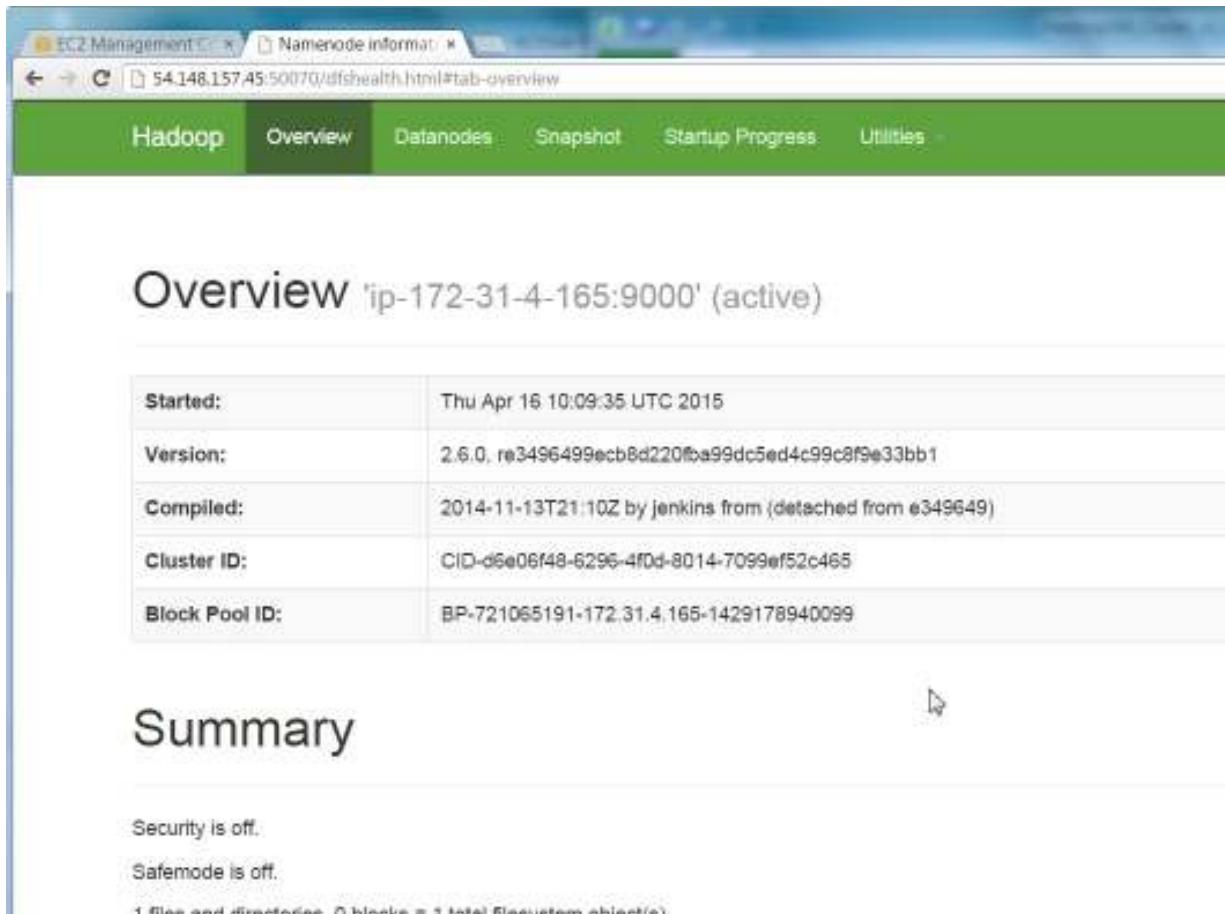
Instance ID	i-c6134c0f	Public DNS	ec2-54-148-157-45.us-west-2.compute.amazonaws.com
Instance state	running	Public IP	54.148.157.45
Instance type	m3.medium	Elastic IP	-
Private DNS	ip-172-31-4-165.us-west-	Availability zone	us-west-2c

A red oval highlights the 'Public IP' value '54.148.157.45'.

Accessing Hadoop Web Console



The screenshot shows a Microsoft Internet Explorer window with the address bar containing the URL 54.148.157.45:90070. Below the address bar, the status bar also displays the same URL. The main content area of the browser shows the Hadoop Web Console's Overview page.



Overview ip-172-31-4-165:9000 (active)

Started:	Thu Apr 16 10:09:35 UTC 2015
Version:	2.6.0, r3496499ecb8d220fb99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-d6e06f48-6296-4fd0-8014-7099ef52c465
Block Pool ID:	BP-721065191-172.31.4.165-1429178940099

Summary

Security is off.
Safemode is off.

1 file and directory. 0 blocks = 1 total file system object(s)

Hadoop Port Numbers

	Daemon	Default Port	Configuration Parameter in conf/*-site.xml
HDFS	Namenode Web	50070	dfs.http.address
	Datanodes	50075	dfs.datanode.http.address
	Secondarynamenode	50090	dfs.secondary.http.address
Yarn	ResourceManager Web	8088	yarn.resourcemanager.webapp.address

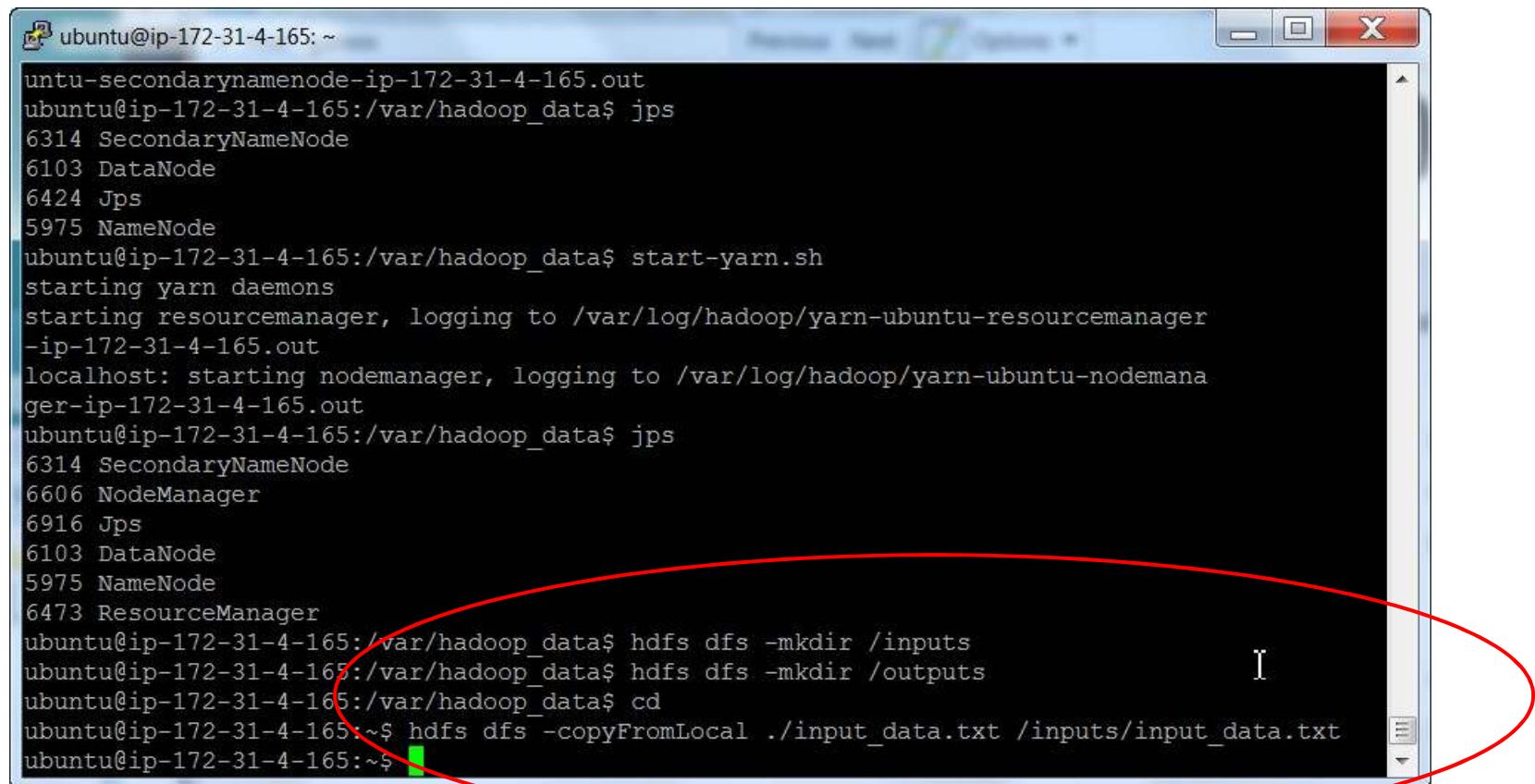
10. Stopping Hadoop

Stop Hadoop Yarn and HDFS

```
ubuntu@ip-172-31-4-165: /var/hadoop_data
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ cd /var/hadoop_data/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ ll
total 16
drwxr-xr-x  4 ubuntu ubuntu 4096 Apr 16 09:39 .
drwxr-xr-x 13 root   root   4096 Apr 16 09:39 ..
drwx----- 3 ubuntu ubuntu 4096 Apr 16 10:09 datanode/
drwxr-xr-x  3 ubuntu ubuntu 4096 Apr 16 10:09 namenode/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ stop-yarn.sh
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
no proxyserver to stop
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
11224 SecondaryNameNode
11014 DataNode
10886 NameNode
12030 Jps
ubuntu@ip-172-31-4-165:/var/hadoop_data$ stop-dfs.sh
Stopping namenodes on [ip-172-31-4-165]
ip-172-31-4-165: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
12469 Jps
ubuntu@ip-172-31-4-165:/var/hadoop_data$
```

Hands-On: Importing Data to HDFS using Hadoop Command Line

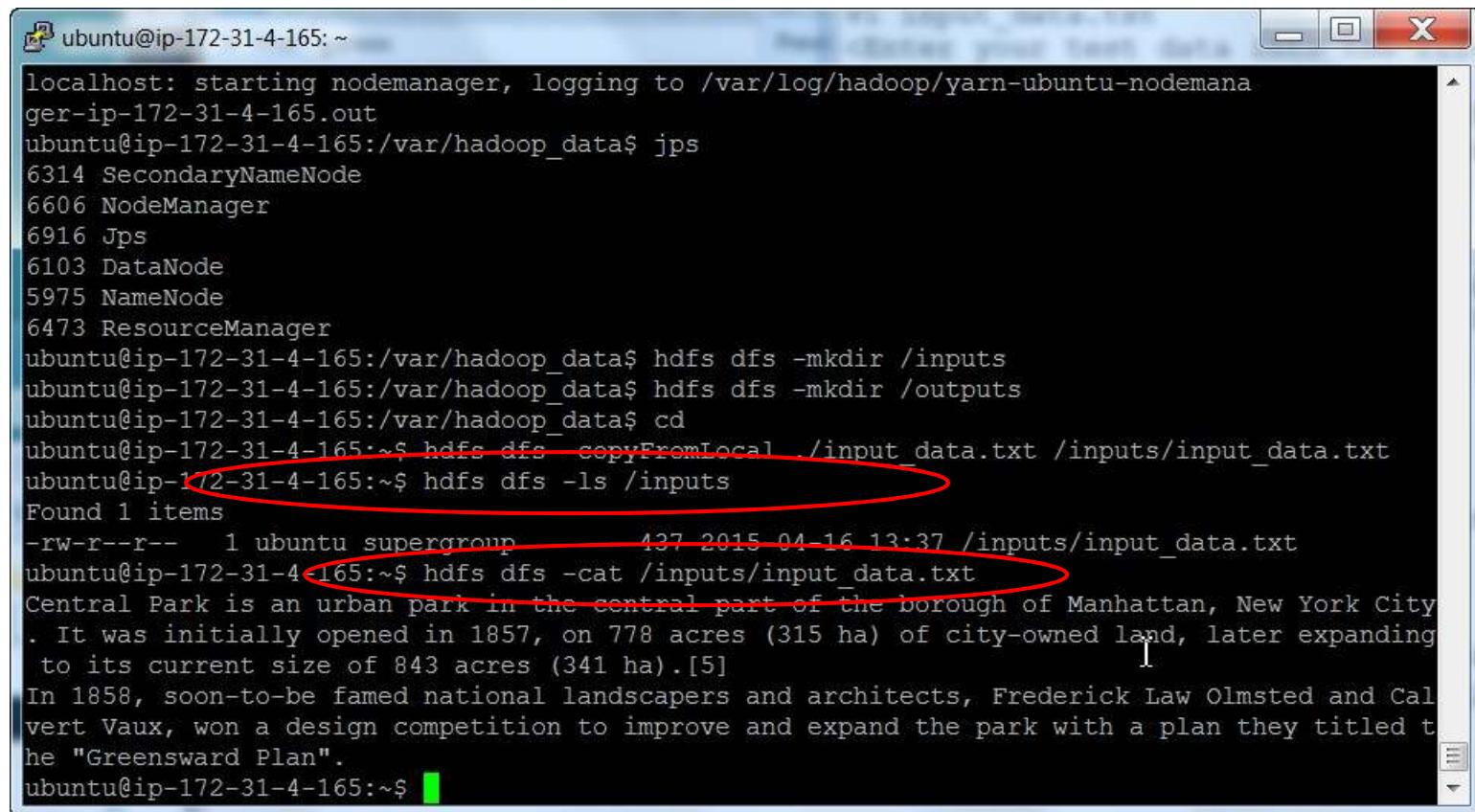
Creating Hadoop HDFS Directories and Importing file to Hadoop



```
untu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
6314 SecondaryNameNode
6103 DataNode
6424 Jps
5975 NameNode
ubuntu@ip-172-31-4-165:/var/hadoop_data$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /var/log/hadoop/yarn-ubuntu-resourcemanager
-ip-172-31-4-165.out
localhost: starting nodemanager, logging to /var/log/hadoop/yarn-ubuntu-nodemana
ger-ip-172-31-4-165.out
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
6314 SecondaryNameNode
6606 NodeManager
6916 Jps
6103 DataNode
5975 NameNode
6473 ResourceManager
ubuntu@ip-172-31-4-165:/var/hadoop_data$ hdfs dfs -mkdir /inputs
ubuntu@ip-172-31-4-165:/var/hadoop_data$ hdfs dfs -mkdir /outputs
ubuntu@ip-172-31-4-165:/var/hadoop_data$ cd
ubuntu@ip-172-31-4-165:~$ hdfs dfs -copyFromLocal ./input_data.txt /inputs/input_data.txt
ubuntu@ip-172-31-4-165:~$
```

Hands-On: Traversing, Retrieving Data from HDFS

Review data in Hadoop HDFS



```
ubuntu@ip-172-31-4-165: ~
localhost: starting nodemanager, logging to /var/log/hadoop/yarn-ubuntu-nodemanager-ip-172-31-4-165.out
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
6314 SecondaryNameNode
6606 NodeManager
6916 Jps
6103 DataNode
5975 NameNode
6473 ResourceManager
ubuntu@ip-172-31-4-165:/var/hadoop_data$ hdfs dfs -mkdir /inputs
ubuntu@ip-172-31-4-165:/var/hadoop_data$ hdfs dfs -mkdir /outputs
ubuntu@ip-172-31-4-165:/var/hadoop_data$ cd
ubuntu@ip-172-31-4-165:~/> hdfs dfs -copyFromLocal ./input_data.txt /inputs/input_data.txt
ubuntu@ip-172-31-4-165:~/> hdfs dfs -ls /inputs
Found 1 items
-rw-r--r-- 1 ubuntu supergroup 437 2015-04-16 13:37 /inputs/input_data.txt
ubuntu@ip-172-31-4-165:~/> hdfs dfs -cat /inputs/input_data.txt
Central Park is an urban park in the central part of the borough of Manhattan, New York City.
. It was initially opened in 1857, on 778 acres (315 ha) of city-owned land, later expanding
to its current size of 843 acres (341 ha).[5]
In 1858, soon-to-be famed national landscapers and architects, Frederick Law Olmsted and Calvert Vaux, won a design competition to improve and expand the park with a plan they titled the "Greensward Plan".
ubuntu@ip-172-31-4-165:~/>
```

Deleting file from HDFS

rm

Usage: hdfs dfs -rm [-f] [-r|-R] [-skipTrash] URI [URI ...]
Delete files specified as args.

Options:

The **-f** option will not display a diagnostic message or modify the exit status to reflect an error if the file does not exist.

The **-R** option deletes the directory and any content under it recursively.

The **-r** option is equivalent to **-R**.

The **-skipTrash** option will bypass trash, if enabled, and delete the specified file(s) immediately. This can be useful when it is necessary to delete files from an over-quota directory.

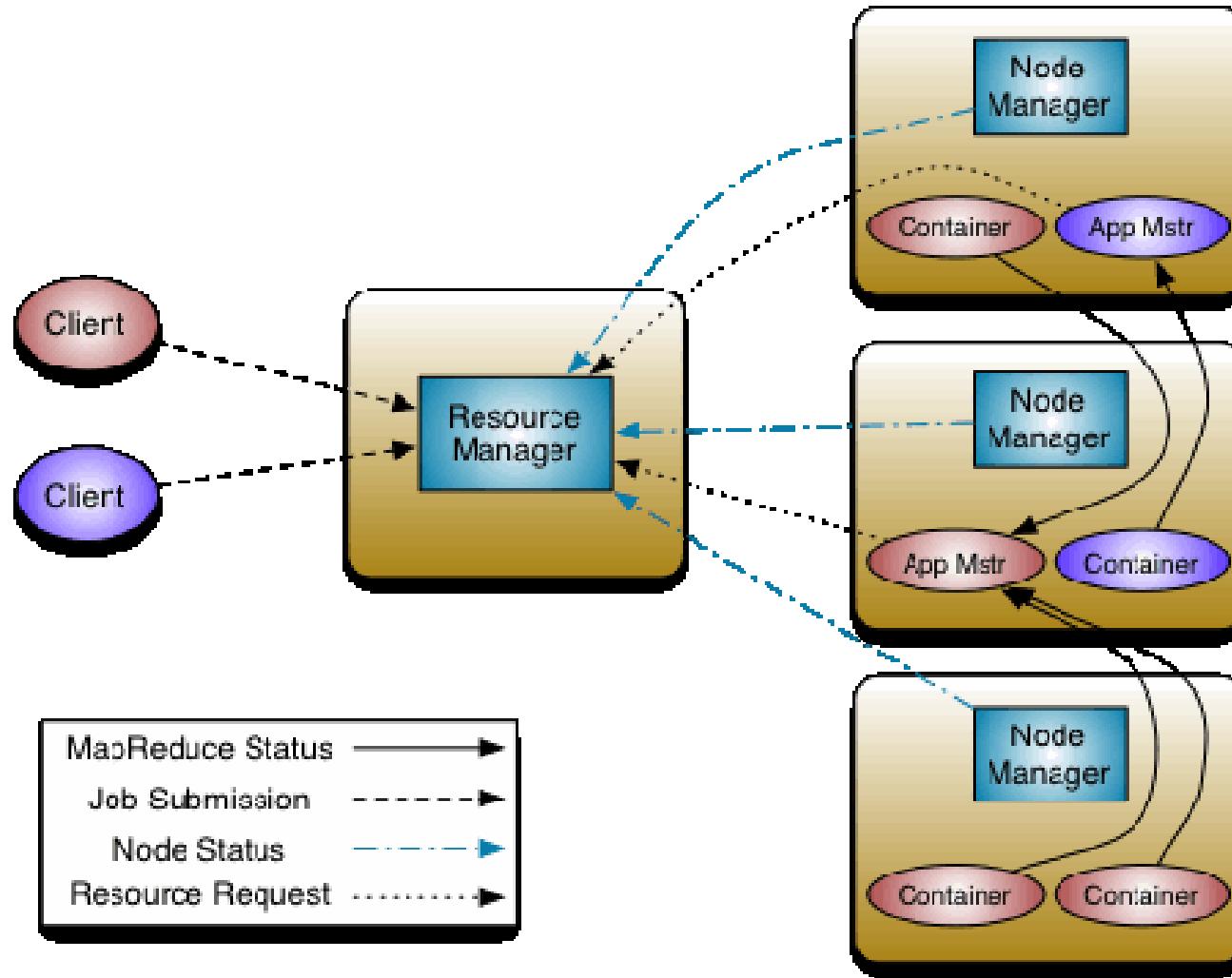
Example:

```
hdfs dfs -rm hdfs://nn.example.com/file /user/hadoop/mydir
```

YARN

Lecture: YARN

YARN: Yet Another Resource Negotiator



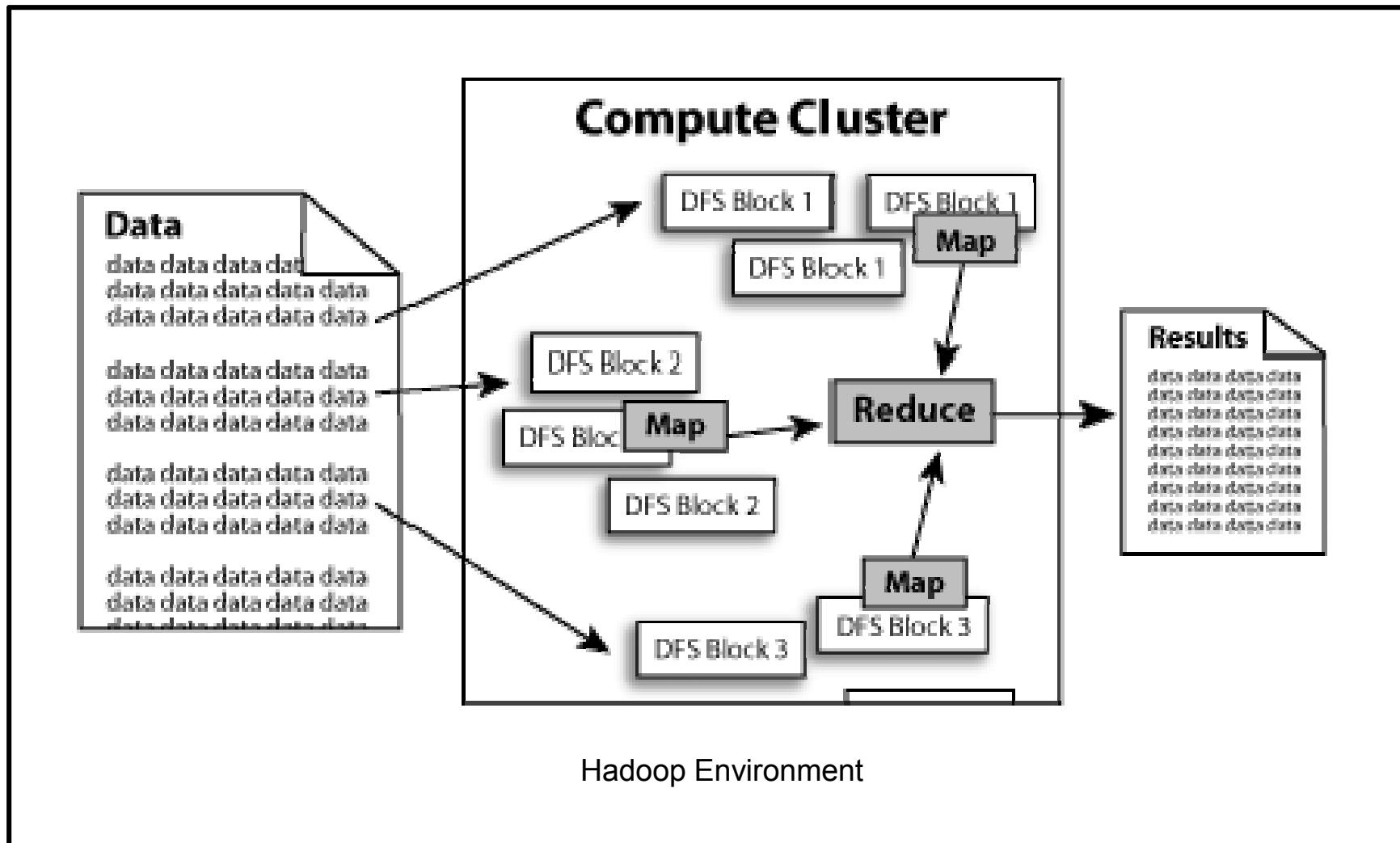
MRV2 maintains API compatibility with previous stable release (hadoop-1.x). This means that all Map-Reduce jobs should still run unchanged on top of MRv2 with just a recompile.

Hadoop.apache.org



Lecture: Map Reduce

MapReduce



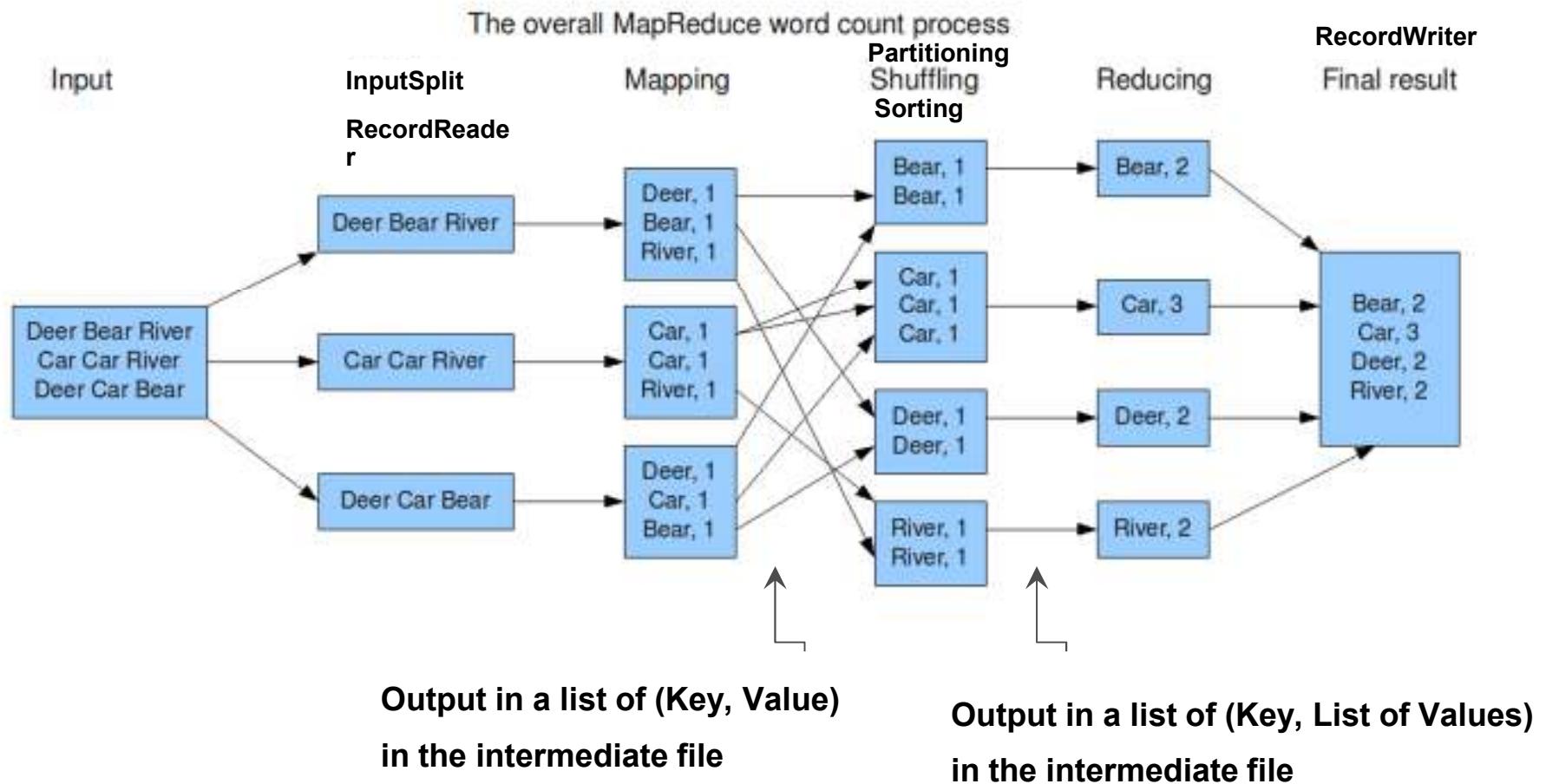
Hadoop.apache.org

MapReduce Framework

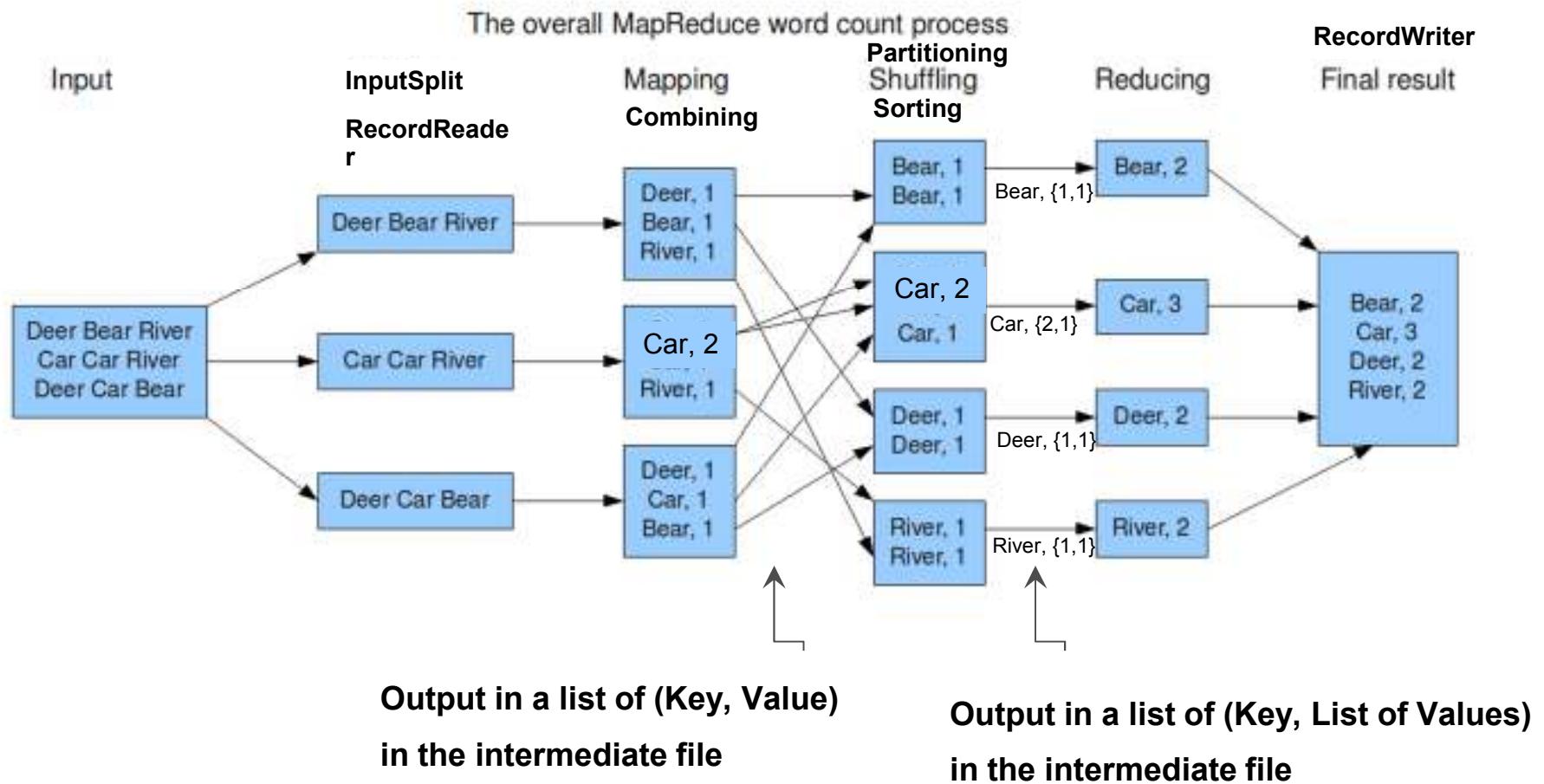
map: $(K1, V1) \rightarrow \text{list}(K2, V2)$

reduce: $(K2, \text{list}(V2)) \rightarrow \text{list}(K3, V3)$

How does the MapReduce work?



How does the MapReduce work?



MapReduce Processing – The Data flow

- 1. InputFormat, InputSplits, RecordReader**
- 2. Mapper - your focus is here**
- 3. Partition, Shuffle & Sort**
- 4. Reducer - your focus is here**
- 5. OutputFormat, RecordWriter**

InputFormat

TextInputFormat	Default format; reads lines of text files	The byte offset of the line	The line contents
KeyValueInputFormat	Parses lines into key, val pairs	Everything up to the first tab character	The remainder of the line
SequenceFileInputFormat	A Hadoop-specific high-performance binary format	user-defined	user-defined

InputSplit

An InputSplit describes a unit of work that comprises a single *map task*.

InputSplit presents a byte-oriented view of the input.

You can control this value by setting the mapred.min.split.size parameter in core-site.xml, or by overriding the parameter in the JobConf object used to submit a particular MapReduce job.

RecordReader

[RecordReader](#) reads <key, value> pairs from an InputSplit.

Typically the RecordReader converts the byte-oriented view of the input, provided by the InputSplit, and presents a record-oriented to the Mapper

Mapper

Mapper: The Mapper performs the user-defined logic to the input a key, value and emits (key, value) pair(s) which are forwarded to the Reducers.

Partition, Shuffle & Sort

After the first map tasks have completed, the nodes may still be performing several more map tasks each. But they also begin exchanging the intermediate outputs from the map tasks to where they are required by the reducers.

Partitioner controls the partitioning of map-outputs to assign to reduce task . he total number of partitions is the same as the number of reduce tasks for the job

The set of intermediate keys on a single node is automatically sorted by internal Hadoop before they are presented to the Reducer

This process of moving map outputs to the reducers is known as shuffling.

Reducer

This is an instance of user-provided code that performs read each key, iterator of values in the partition assigned. The *OutputCollector* object in Reducer phase has a method named `collect()` which will collect a (key, value) output.

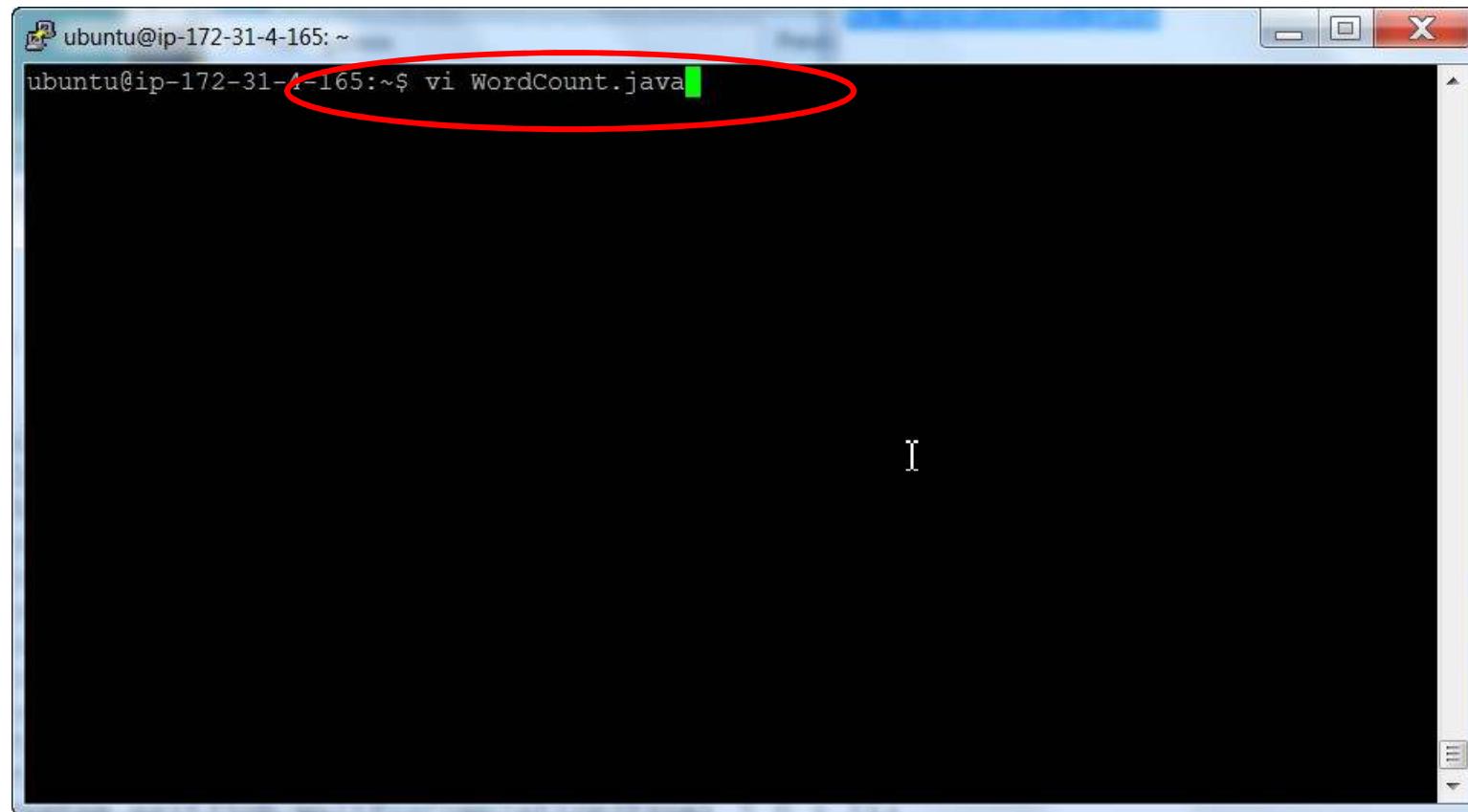
OutputFormat, Record Writer

OutputFormat governs the writing format in **OutputCollector** and **RecordWriter** writes output into HDFS.

TextOutputFormat	Default; writes lines in "key \t value" form
SequenceFileOutputFormat	Writes binary files suitable for reading into subsequent MapReduce jobs
NullOutputFormat	generates no output files

Hands-On: Writing Your Own Map Reduce

Writing Your Own Map Reduce



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the command "ubuntu@ip-172-31-4-165:~\$ vi WordCount.java". A red oval highlights the command line. The terminal has a blue header bar and a black body with a vertical scroll bar on the right.

Writing Your Own Map Reduce

```
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;

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);
            }
        }
    }
}
```

Continue (ມືດອ) ..

Writing Your Own Map Reduce

```
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);
    }
}
```

Continue (ມືດອ)..

Writing Your Own Map Reduce

```
public static void main(String[] args) throws Exception {  
    Configuration conf = new Configuration();  
    Job job = Job.getInstance(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(args[0]));  
    FileOutputFormat.setOutputPath(job, new Path(args[1]));  
    System.exit(job.waitForCompletion(true) ? 0 : 1);  
}  
}
```

Hands-On: Packaging Map Reduce and Deploying to Hadoop Runtime Environment

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

```
>Create java classes directory (สร้าง directory สำหรับ java class ที่ถูก compile แล้ว)
mkdir wordcount_classes

Typing in one single line (พิมพ์ต่อเนื่องเลย อย่ากดแป้นขึ้นบรรทัดใหม่)

javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-
2.6.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-
core-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d
wordcount_classes WordCount.java

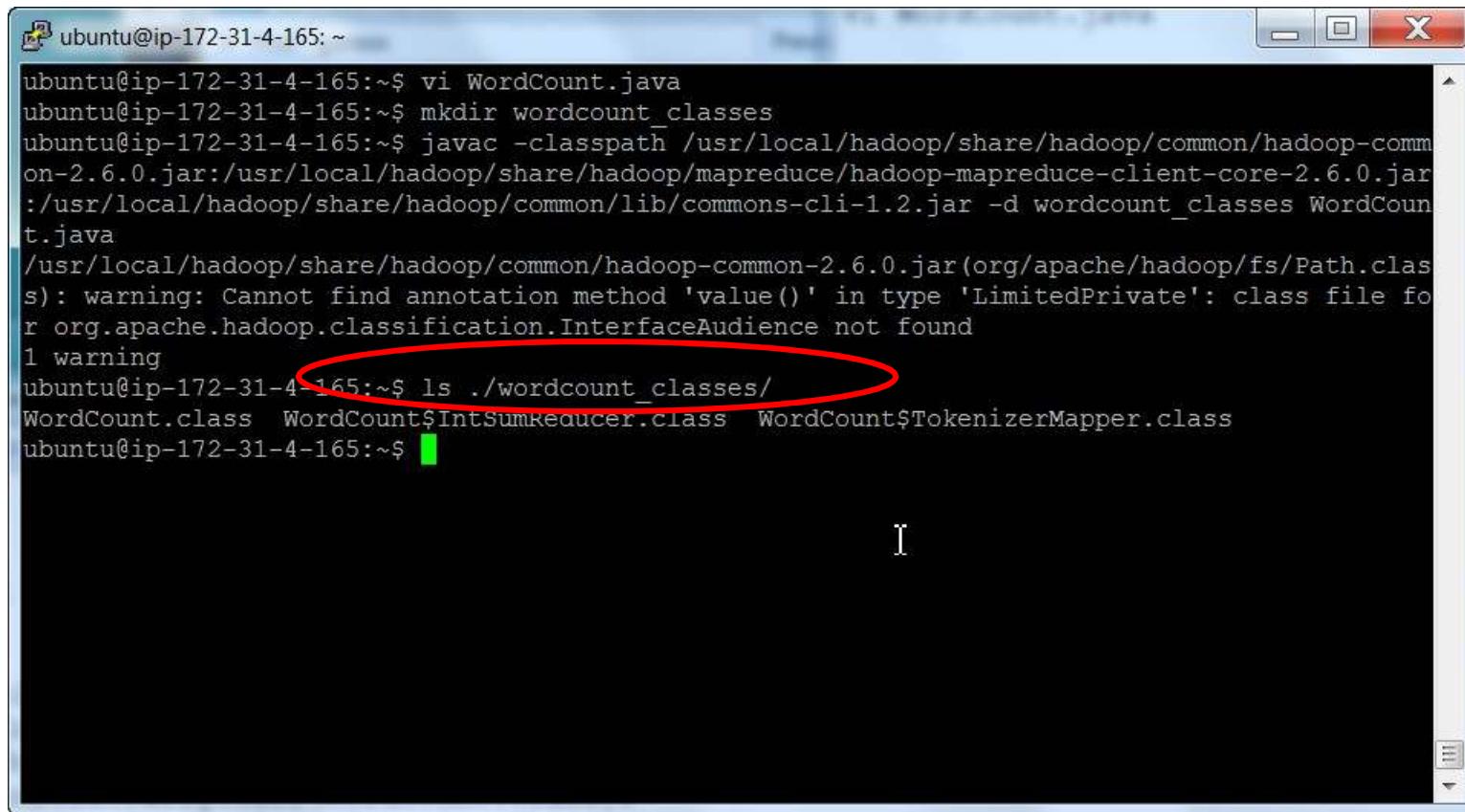
Create jar file (สร้าง jar file อย่าลืมใส่เครื่องหมายจุดด้านหลังด้วย)
jar -cvf ./wordcount.jar -C wordcount_classes/

Execute yarn (สั่งให้ yarn ทำงาน)
yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount_output_dir01

Review the result (สั่งให้พิมพ์ผลลัพธ์ออกหน้าจอ)
hdfs dfs -cat /outputs/wordcount_output_dir01/part-r-00000
```

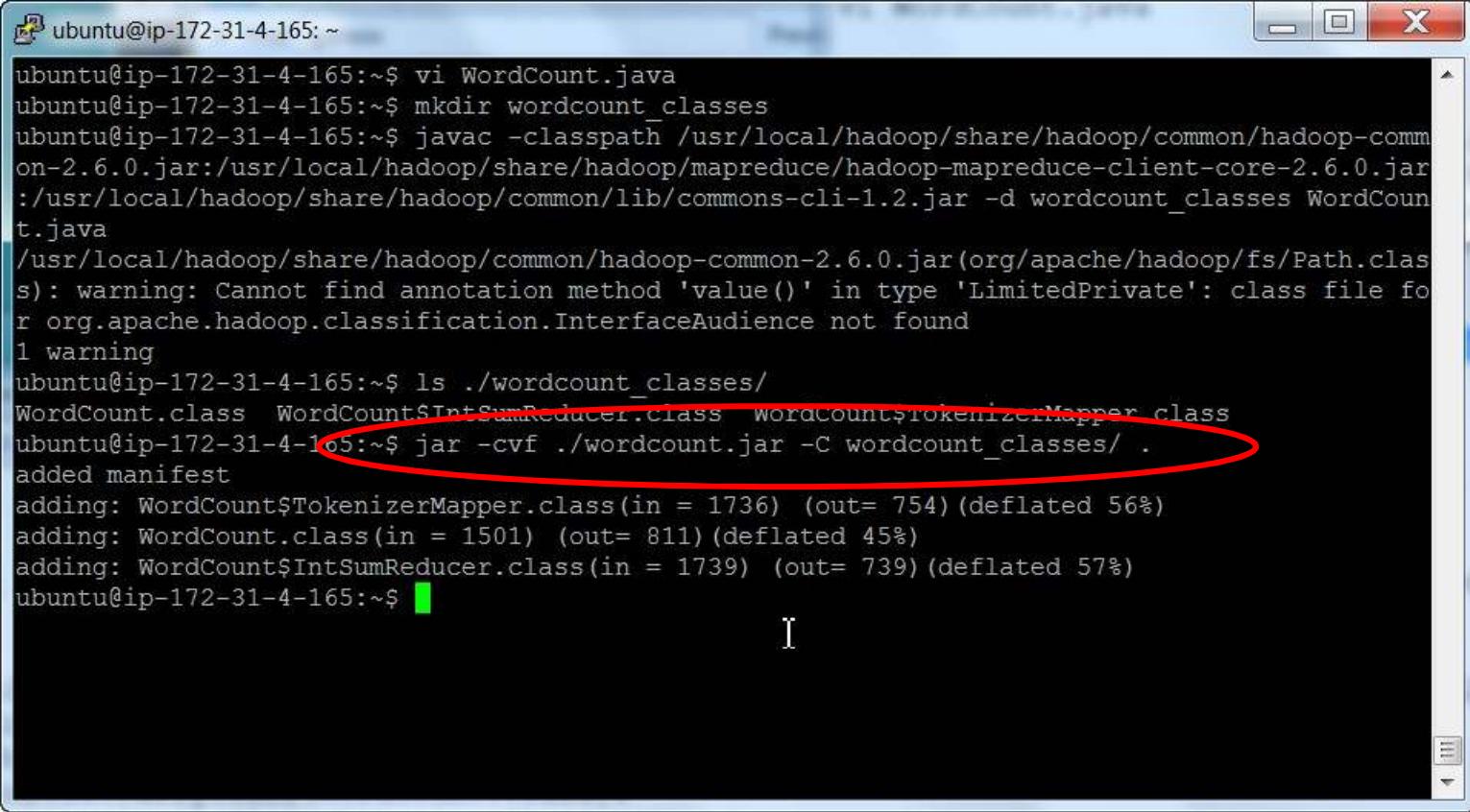
Please find next page for screen step by step (โปรดดูหน้าถัดไปสำหรับหน้าจอในแต่ละขั้นตอน)...

Packaging Map Reduce and Deploying to Hadoop Runtime Environment



```
ubuntu@ip-172-31-4-165:~$ vi WordCount.java
ubuntu@ip-172-31-4-165:~$ mkdir wordcount_classes
ubuntu@ip-172-31-4-165:~$ javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d wordcount_classes WordCount.java
/usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar(org/apache/hadoop/fs/Path.class): warning: Cannot find annotation method 'value()' in type 'LimitedPrivate': class file for org.apache.hadoop.classification.InterfaceAudience not found
1 warning
ubuntu@ip-172-31-4-165:~$ ls ./wordcount_classes/
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
ubuntu@ip-172-31-4-165:~$
```

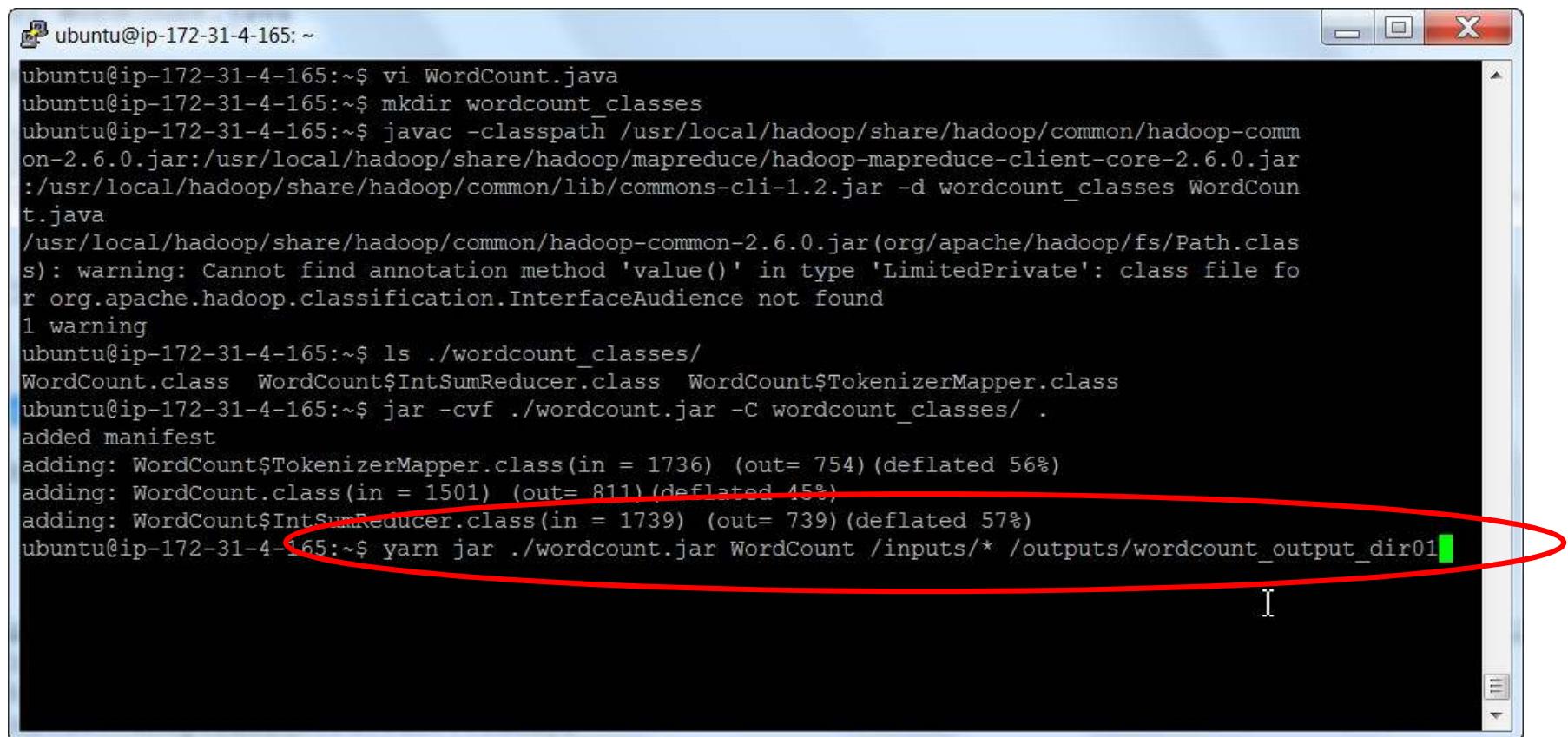
Packaging Map Reduce and Deploying to Hadoop Runtime Environment



The screenshot shows a terminal window titled "ubuntu@ip-172-31-4-165: ~". The user is performing the following steps:

- Creating a directory for the classes: `mkdir wordcount_classes`
- Compiling the Java code: `javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d wordcount_classes WordCount.java`. A warning message is displayed about a missing annotation method.
- Listing the contents of the `wordcount_classes` directory: `ls ./wordcount_classes/`. The output shows three class files: `WordCount.class`, `WordCount$IntSumReducer.class`, and `WordCount$TokenizerMapper.class`.
- Packing the classes into a JAR file: `jar -cvf ./wordcount.jar -C wordcount_classes/ .`. This command adds the manifest and compresses the three class files.

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

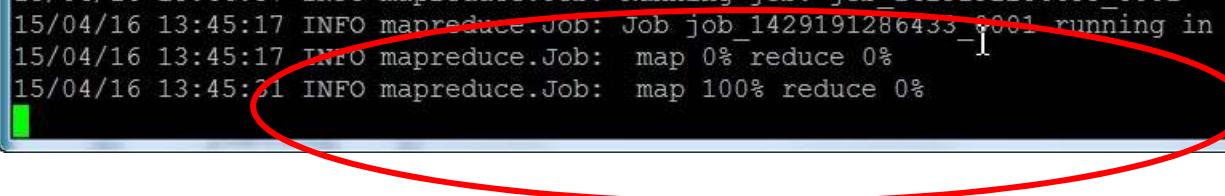


```
ubuntu@ip-172-31-4-165:~$ vi WordCount.java
ubuntu@ip-172-31-4-165:~$ mkdir wordcount_classes
ubuntu@ip-172-31-4-165:~$ javac -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.6.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d wordcount_classes WordCount.java
/usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar(org/apache/hadoop/fs/Path.class): warning: Cannot find annotation method 'value()' in type 'LimitedPrivate': class file for org.apache.hadoop.classification.InterfaceAudience not found
1 warning
ubuntu@ip-172-31-4-165:~$ ls ./wordcount_classes/
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
ubuntu@ip-172-31-4-165:~$ jar -cvf ./wordcount.jar -C wordcount_classes/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754) (deflated 56%)
adding: WordCount.class(in = 1501) (out= 811) (deflated 45%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739) (deflated 57%)
ubuntu@ip-172-31-4-165:~$ yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount_output_dir01
```

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

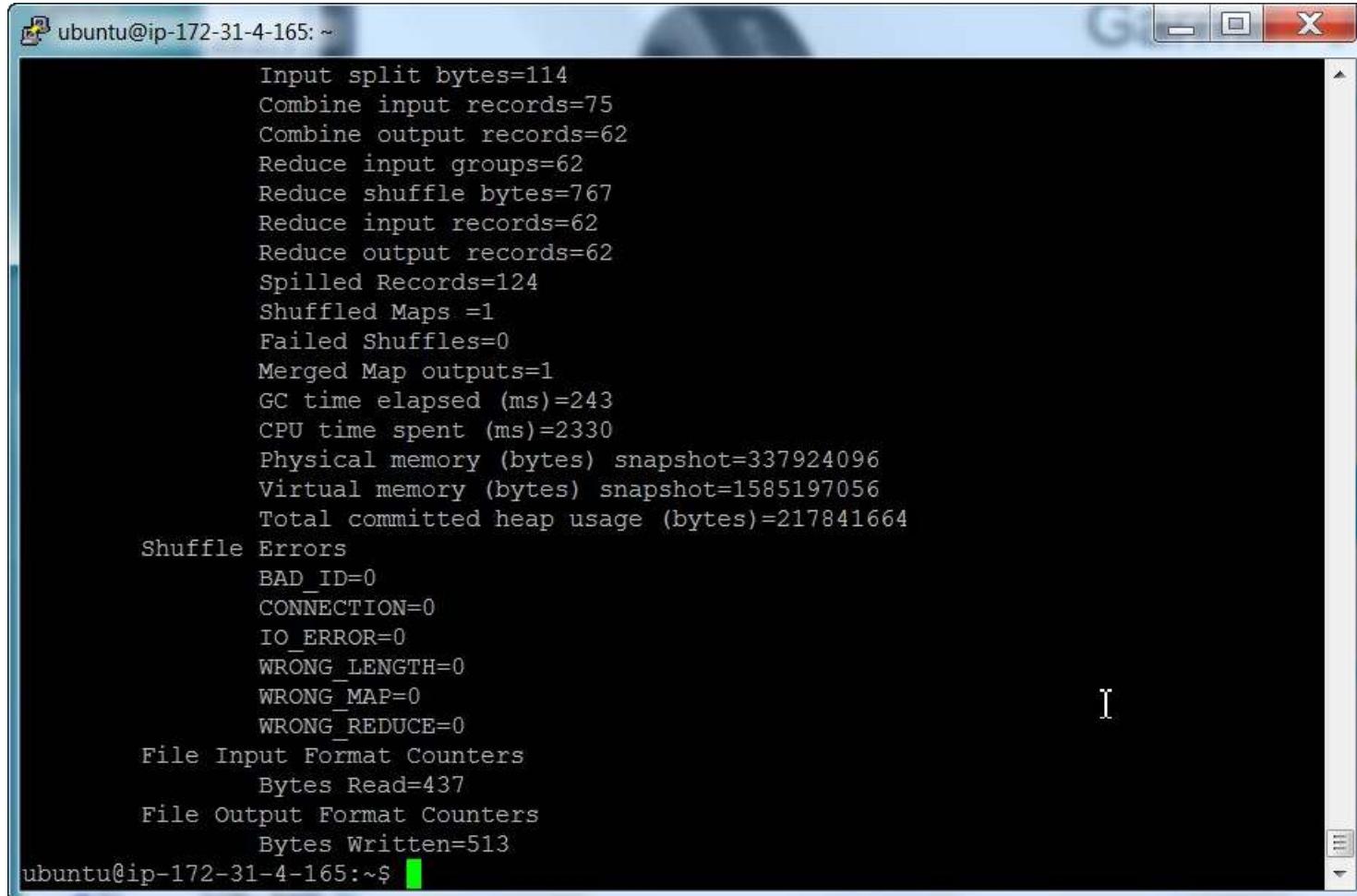
```
ubuntu@ip-172-31-4-165: ~
t.java
/usr/local/hadoop/share/hadoop/common/hadoop-common-2.6.0.jar(org/apache/hadoop/fs/Path.clas
s): warning: Cannot find annotation method 'value()' in type 'LimitedPrivate': class file fo
r org.apache.hadoop.classification.InterfaceAudience not found
1 warning
ubuntu@ip-172-31-4-165:~$ ls ./wordcount_classes/
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
ubuntu@ip-172-31-4-165:~$ jar -cvf ./wordcount.jar -C wordcount_classes/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754) (deflated 56%)
adding: WordCount.class(in = 1501) (out= 811) (deflated 45%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739) (deflated 57%)
ubuntu@ip-172-31-4-165:~$ yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount_output_dir01
15/04/16 13:44:54 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-4-165/172.31.4.165:8032
15/04/16 13:44:54 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
15/04/16 13:44:55 INFO input.FileInputFormat: Total input paths to process : 1
15/04/16 13:44:55 INFO mapreduce.JobSubmitter: number of splits:1
15/04/16 13:44:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1429191286433_0001
15/04/16 13:44:57 INFO impl.YarnClientImpl: Submitted application application_1429191286433_0001
15/04/16 13:44:57 INFO mapreduce.Job: The url to track the job: http://ip-172-31-4-165:8088/proxy/applica
tion_1429191286433_0001/
15/04/16 13:44:57 INFO mapreduce.Job: Running job: job_1429191286433_0001
```

Packaging Map Reduce and Deploying to Hadoop Runtime Environment



```
ubuntu@ip-172-31-4-165:~$ org.apache.hadoop.classification.InterfaceAudience not found
1 warning
ubuntu@ip-172-31-4-165:~$ ls ./wordcount_classes/
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
ubuntu@ip-172-31-4-165:~$ jar -cvf ./wordcount.jar -C wordcount_classes/ .
added manifest
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 754) (deflated 56%)
adding: WordCount.class(in = 1501) (out= 811) (deflated 45%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 739) (deflated 57%)
ubuntu@ip-172-31-4-165:~$ yarn jar ./wordcount.jar WordCount /inputs/* /outputs/wordcount output_dir01
15/04/16 13:44:54 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-4-165/172.31.4.165:8032
15/04/16 13:44:54 WARN mapreduce.JobSubmitter: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
15/04/16 13:44:55 INFO input.FileInputFormat: Total input paths to process : 1
15/04/16 13:44:55 INFO mapreduce.JobSubmitter: number of splits:1
15/04/16 13:44:56 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1429191286433_0001
15/04/16 13:44:57 INFO impl.YarnClientImpl: Submitted application application_1429191286433_0001
15/04/16 13:44:57 INFO mapreduce.Job: The url to track the job: http://ip-172-31-4-165:8088/proxy/application_1429191286433_0001
15/04/16 13:44:57 INFO mapreduce.Job: Running job: job_1429191286433_0001
15/04/16 13:45:17 INFO mapreduce.Job: Job job_1429191286433_0001 running in uber mode : false
15/04/16 13:45:17 INFO mapreduce.Job: map 0% reduce 0%
15/04/16 13:45:31 INFO mapreduce.Job: map 100% reduce 0%
```

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

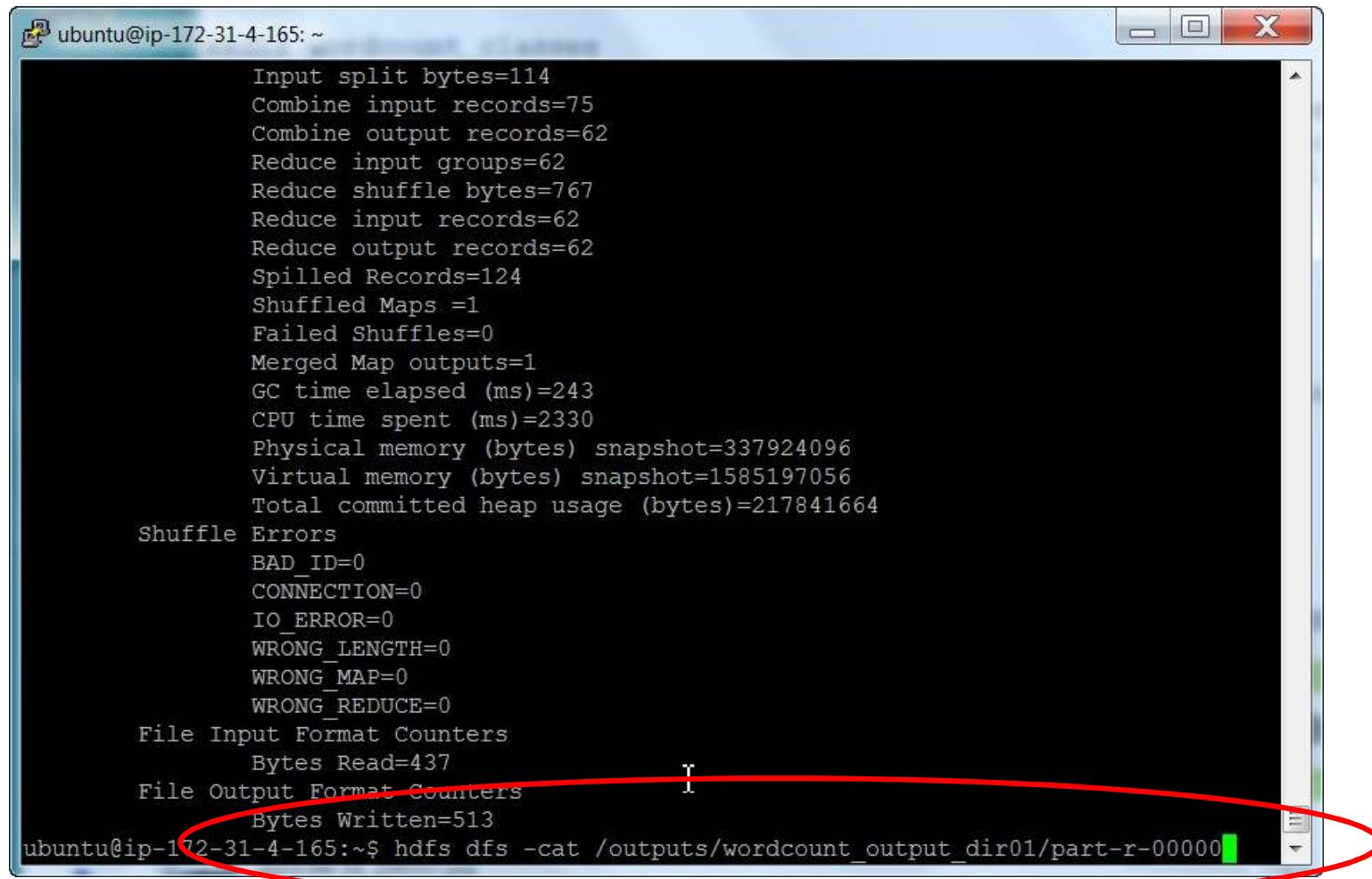
A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165:~". The window displays various Hadoop job metrics. The output includes:

```
Input split bytes=114
Combine input records=75
Combine output records=62
Reduce input groups=62
Reduce shuffle bytes=767
Reduce input records=62
Reduce output records=62
Spilled Records=124
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=243
CPU time spent (ms)=2330
Physical memory (bytes) snapshot=337924096
Virtual memory (bytes) snapshot=1585197056
Total committed heap usage (bytes)=217841664
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=437
File Output Format Counters
Bytes Written=513
```

The terminal prompt at the bottom is "ubuntu@ip-172-31-4-165:~\$".

ubuntu@ip-172-31-4-165:~\$

Packaging Map Reduce and Deploying to Hadoop Runtime Environment



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window displays various Hadoop MapReduce metrics and statistics. A red oval highlights the command at the bottom of the window.

```
Input split bytes=114
Combine input records=75
Combine output records=62
Reduce input groups=62
Reduce shuffle bytes=767
Reduce input records=62
Reduce output records=62
Spilled Records=124
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=243
CPU time spent (ms)=2330
Physical memory (bytes) snapshot=337924096
Virtual memory (bytes) snapshot=1585197056
Total committed heap usage (bytes)=217841664
Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=437
File Output Format Counters
Bytes Written=513
ubuntu@ip-172-31-4-165:~$ hdfs dfs -cat /outputs/wordcount_output_dir01/part-r-00000
```

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

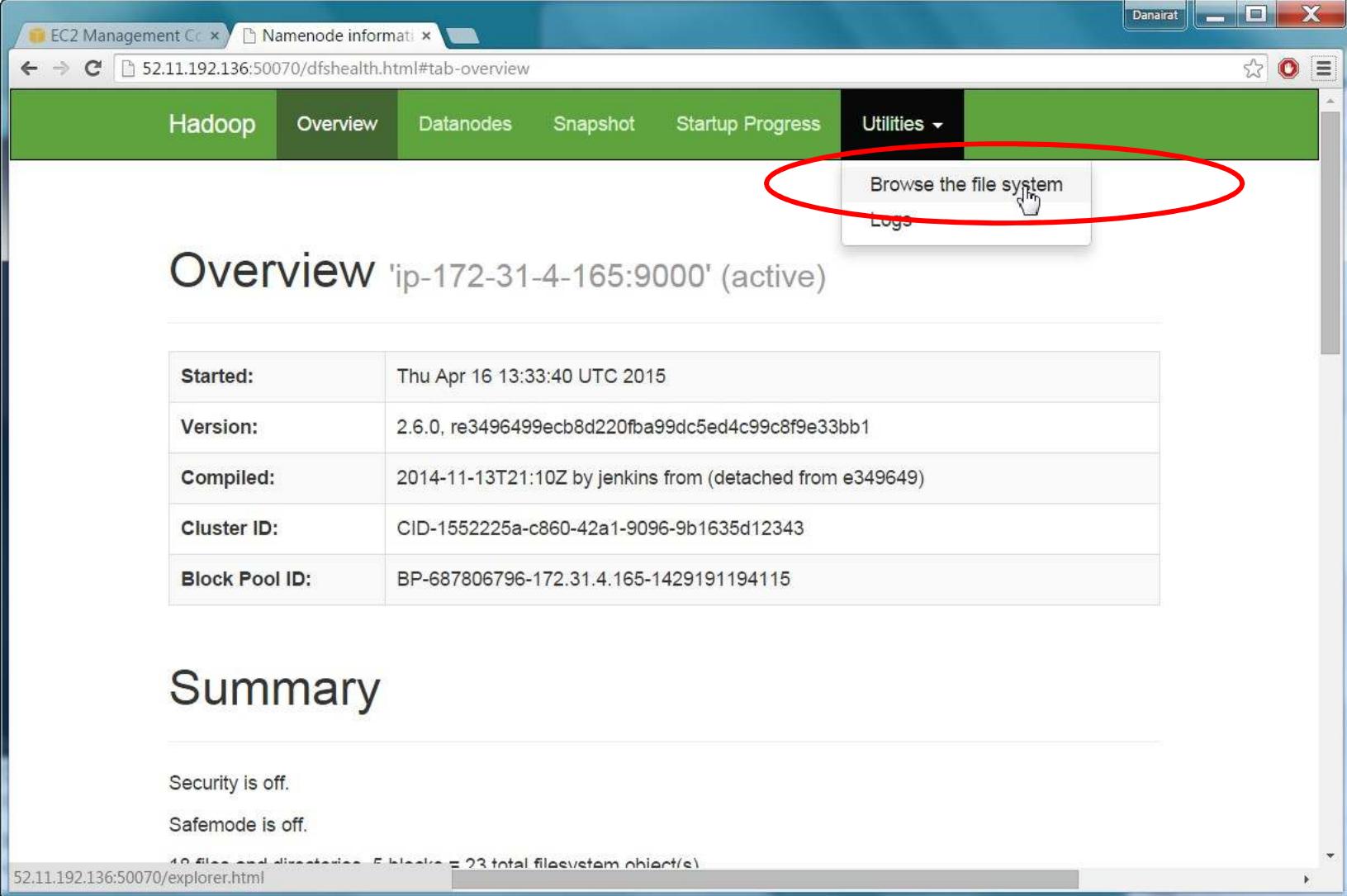


A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window displays the output of a word count program, showing word counts for various terms. The output is as follows:

```
ha)      1
ha).[5]  1
improve  1
in       2
initially    1
is       1
its      1
land,    1
landscapers  1
later    1
national  1
of       4
on       1
opened   1
park     2
part     1
plan     1
size     1
soon-to-be  1
the      4
they    1
titled  1
to      2
urban   1
was     1
with    1
won     1
```

The terminal prompt at the bottom is "ubuntu@ip-172-31-4-165:~\$".

Packaging Map Reduce and Deploying to Hadoop Runtime Environment



The screenshot shows a web browser window titled "EC2 Management Consoles" with the URL "52.11.192.136:50070/dfshealth.html#tab-overview". The page has a green header bar with tabs: Hadoop, Overview, Datanodes, Snapshot, Startup Progress, Utilities (with a dropdown arrow), and a button labeled "Logs". A red circle highlights the "Utilities" dropdown menu, specifically the "Browse the file system" option. Below the header, the page title is "Overview 'ip-172-31-4-165:9000' (active)". There is a table with the following data:

Started:	Thu Apr 16 13:33:40 UTC 2015
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1552225a-c860-42a1-9096-9b1635d12343
Block Pool ID:	BP-687806796-172.31.4.165-1429191194115

Below the table, there is a section titled "Summary" with the following status:

- Security is off.
- Safemode is off.

At the bottom of the page, it says "40 files and directories, 5 blocks = 23 total filesystem object(s)" and the URL "52.11.192.136:50070/explorer.html".

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

EC2 Management Consoles - Browsing HDFS

52.11.192.136:50070/explorer.html#/

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

Search Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	ubuntu	supergroup	0 B	0	0 B	inputs
drwxr-xr-x	ubuntu	supergroup	0 B	0	0 B	outputs
drwx-----	ubuntu	supergroup	0 B	0	0 B	tmp

Hadoop, 2014.

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

The screenshot shows a web browser window titled "Browsing HDFS" with the URL "52.11.192.136:50070/explorer.html#/outputs". The browser has a green header bar with tabs for "Hadoop", "Overview", "Datanodes", "Snapshot", "Startup Progress", and "Utilities". The main content area is titled "Browse Directory" and shows a table with the following data:

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	ubuntu	supergroup	0 B	0	0 B	wordcount_output_dir01

A red oval highlights the "Name" column of the second row, specifically the link "wordcount_output_dir01". Below the table, the text "Hadoop, 2014." is visible.

Packaging Map Reduce and Deploying to Hadoop Runtime Environment

Browsing HDFS

52.11.192.136:50070/explorer.html#/outputs/wordcount_output_dir01

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

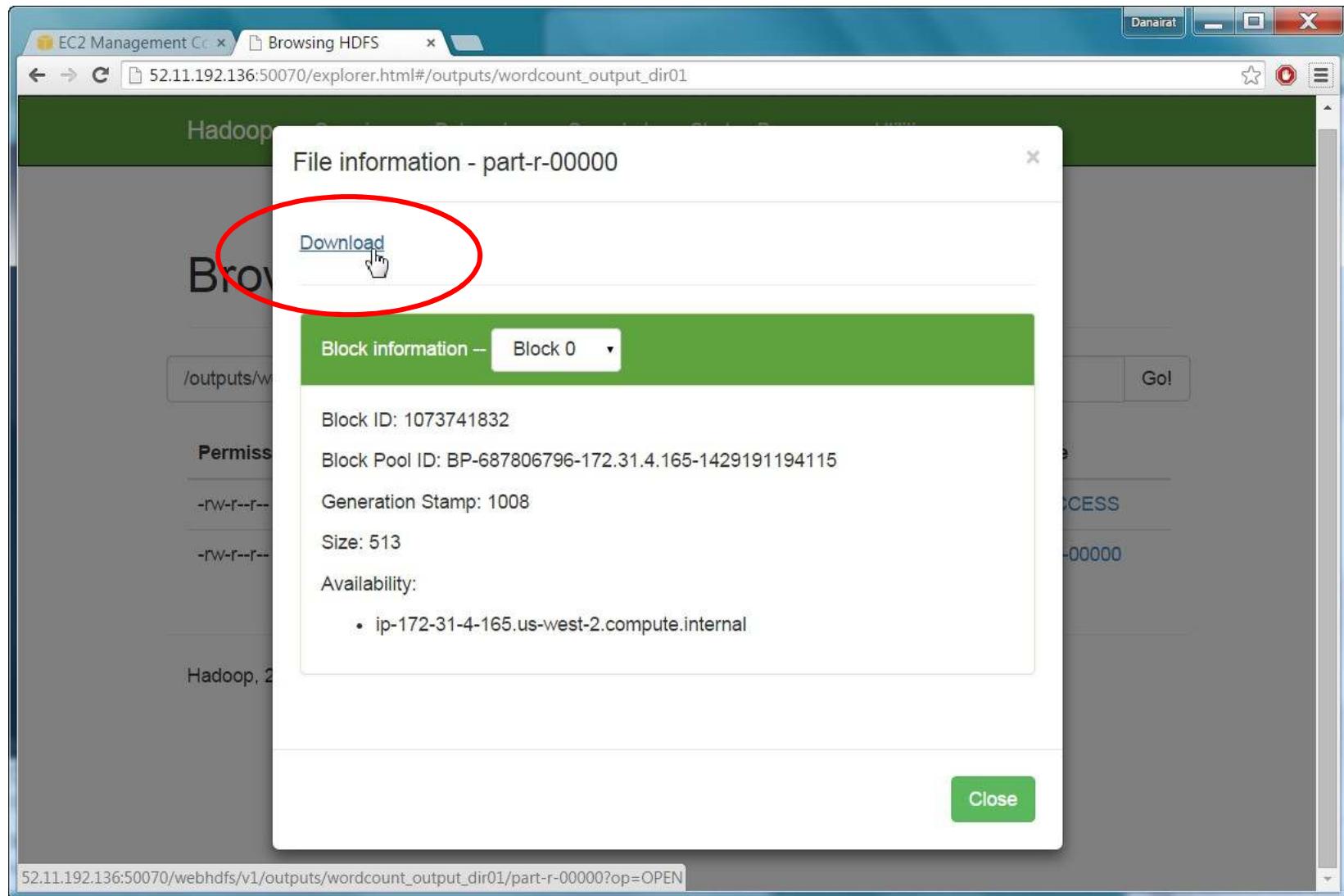
Browse Directory

/outputs/wordcount_output_dir01 Go!

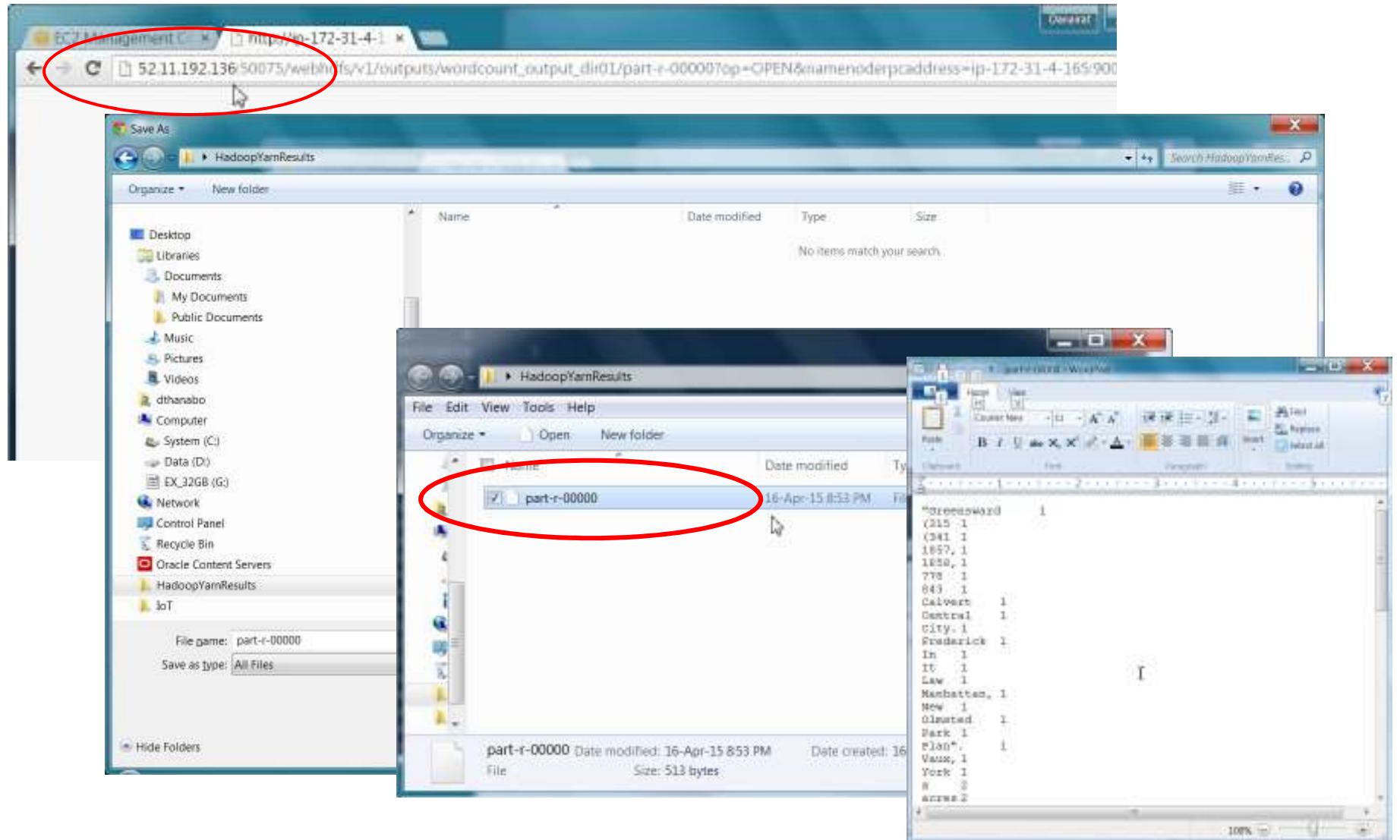
Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	ubuntu	supergroup	0 B	1	128 MB	_SUCCESS
-rw-r--r--	ubuntu	supergroup	513 B	1	128 MB	part-r-00000

Hadoop, 2014.

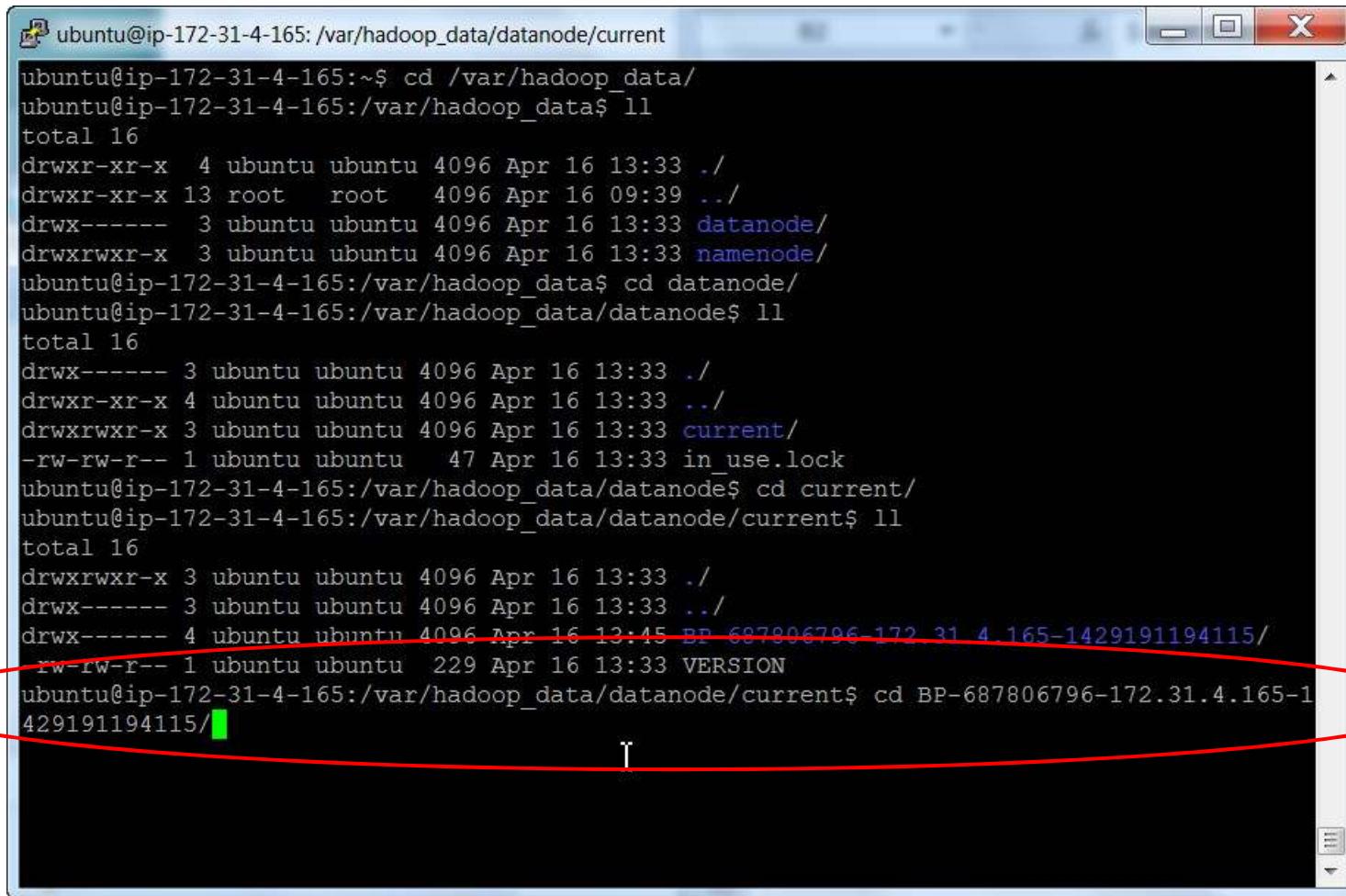
Packaging Map Reduce and Deploying to Hadoop Runtime Environment



Packaging Map Reduce and Deploying to Hadoop Runtime Environment



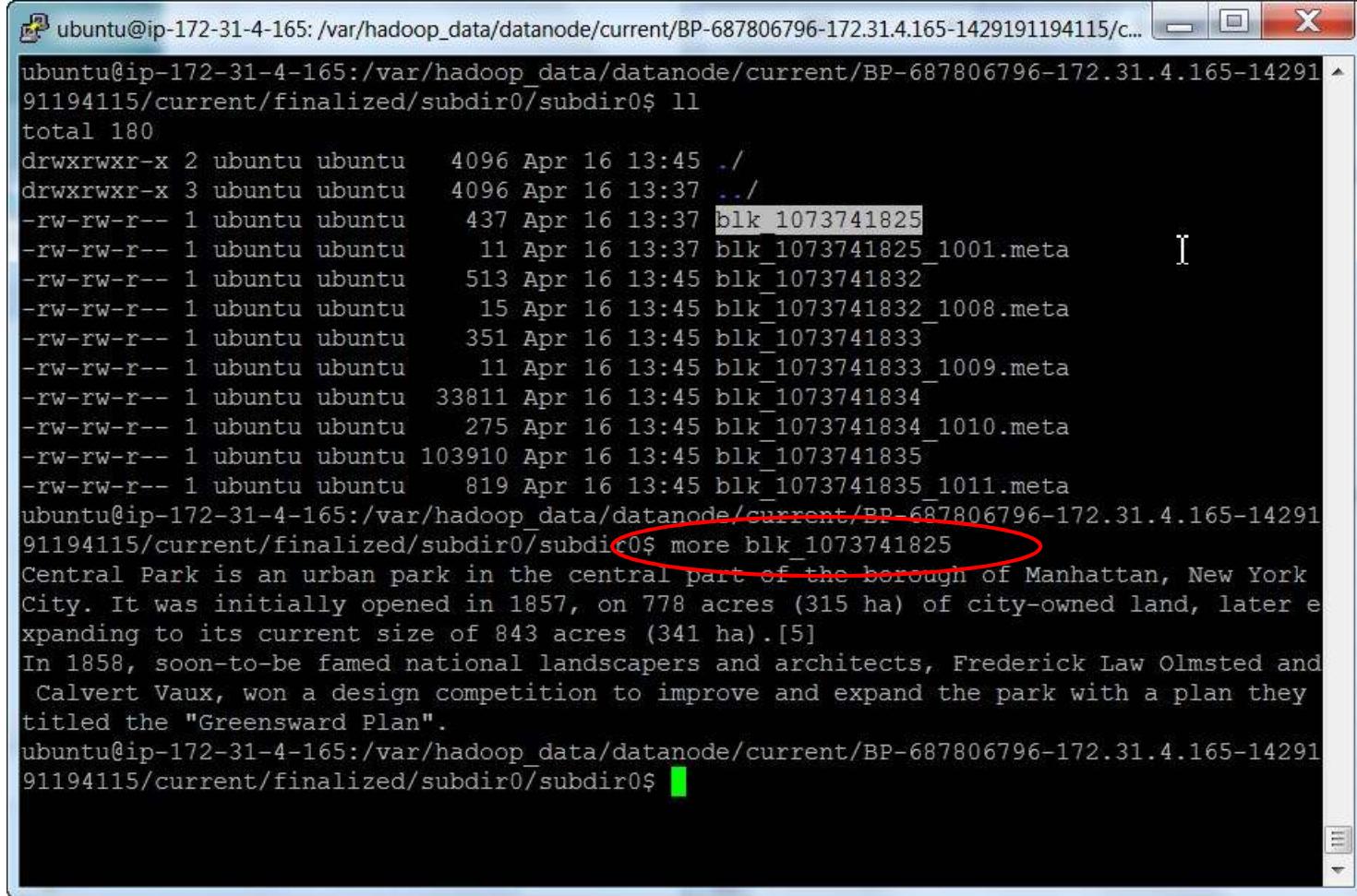
(Optional) Review OS File System



The screenshot shows a terminal window titled "ubuntu@ip-172-31-4-165: /var/hadoop_data/datanode/current". The terminal displays a series of "cd" and "ls" commands used to navigate through directory structures and list files. A red oval highlights the path "BP-687806796-172.31.4.165-1429191194115/" which contains a file named "VERSION".

```
ubuntu@ip-172-31-4-165:~$ cd /var/hadoop_data/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ ls
total 16
drwxr-xr-x  4 ubuntu  ubuntu  4096 Apr 16 13:33 .
drwxr-xr-x 13 root   root   4096 Apr 16 09:39 ..
drwx----- 3 ubuntu  ubuntu  4096 Apr 16 13:33 datanode/
drwxrwxr-x  3 ubuntu  ubuntu  4096 Apr 16 13:33 namenode/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ cd datanode/
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode$ ls
total 16
drwx----- 3 ubuntu  ubuntu  4096 Apr 16 13:33 .
drwxr-xr-x  4 ubuntu  ubuntu  4096 Apr 16 13:33 ..
drwxrwxr-x  3 ubuntu  ubuntu  4096 Apr 16 13:33 current/
-rw-rw-r--  1 ubuntu  ubuntu   47 Apr 16 13:33 in_use.lock
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode$ cd current/
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current$ ls
total 16
drwxrwxr-x  3 ubuntu  ubuntu  4096 Apr 16 13:33 .
drwx----- 3 ubuntu  ubuntu  4096 Apr 16 13:33 ..
drwx----- 4 ubuntu  ubuntu  4096 Apr 16 13:45 BP-687806796-172.31.4.165-1429191194115/
  rw-rw-r--  1 ubuntu  ubuntu  229 Apr 16 13:33 VERSION
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current$ cd BP-687806796-172.31.4.165-1429191194115/
```

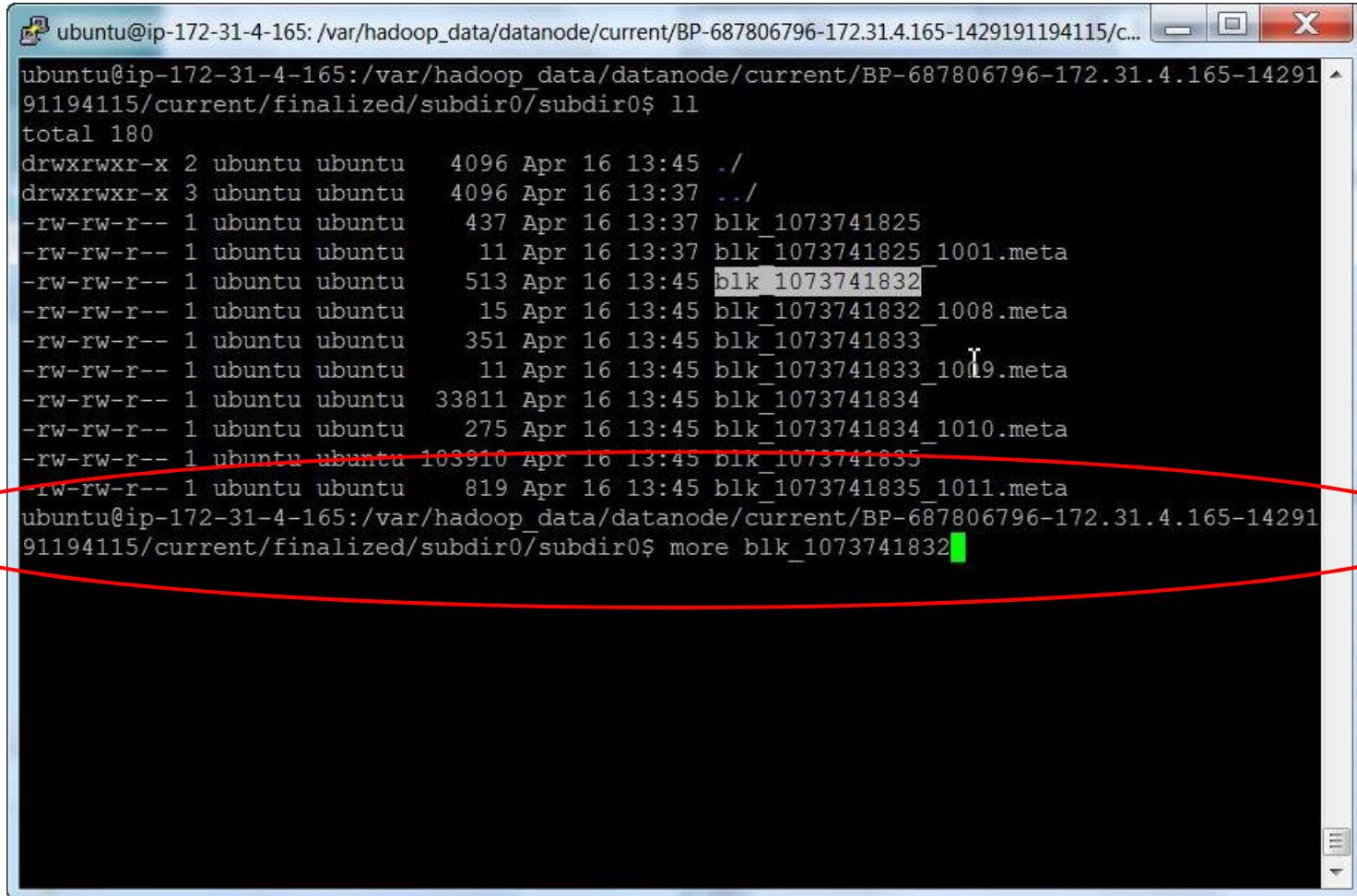
(Optional) Review OS File System



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: /var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/c...". The terminal shows a directory listing and then uses the "more" command to view the contents of a file. A red circle highlights the file name "blk_1073741825" in the listing, and another red circle highlights the command "more blk_1073741825" in the session history.

```
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/c...
91194115/current/finalized/subdir0$ ll
total 180
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 16 13:45 .
drwxrwxr-x 3 ubuntu ubuntu 4096 Apr 16 13:37 ..
-rw-rw-r-- 1 ubuntu ubuntu 437 Apr 16 13:37 blk_1073741825
-rw-rw-r-- 1 ubuntu ubuntu 11 Apr 16 13:37 blk_1073741825_1001.meta
-rw-rw-r-- 1 ubuntu ubuntu 513 Apr 16 13:45 blk_1073741832
-rw-rw-r-- 1 ubuntu ubuntu 15 Apr 16 13:45 blk_1073741832_1008.meta
-rw-rw-r-- 1 ubuntu ubuntu 351 Apr 16 13:45 blk_1073741833
-rw-rw-r-- 1 ubuntu ubuntu 11 Apr 16 13:45 blk_1073741833_1009.meta
-rw-rw-r-- 1 ubuntu ubuntu 33811 Apr 16 13:45 blk_1073741834
-rw-rw-r-- 1 ubuntu ubuntu 275 Apr 16 13:45 blk_1073741834_1010.meta
-rw-rw-r-- 1 ubuntu ubuntu 103910 Apr 16 13:45 blk_1073741835
-rw-rw-r-- 1 ubuntu ubuntu 819 Apr 16 13:45 blk_1073741835_1011.meta
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/current/finalized/subdir0$ more blk_1073741825
Central Park is an urban park in the central part of the borough of Manhattan, New York City. It was initially opened in 1857, on 778 acres (315 ha) of city-owned land, later expanding to its current size of 843 acres (341 ha). [5]
In 1858, soon-to-be famed national landscapers and architects, Frederick Law Olmsted and Calvert Vaux, won a design competition to improve and expand the park with a plan they titled the "Greensward Plan".
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/current/finalized/subdir0$
```

(Optional) Review OS File System



```
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/c... □ X
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/current/finalized/subdir0/subdir0$ ll
total 180
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr 16 13:45 .
drwxrwxr-x 3 ubuntu ubuntu 4096 Apr 16 13:37 ..
-rw-rw-r-- 1 ubuntu ubuntu 437 Apr 16 13:37 blk_1073741825
-rw-rw-r-- 1 ubuntu ubuntu 11 Apr 16 13:37 blk_1073741825_1001.meta
-rw-rw-r-- 1 ubuntu ubuntu 513 Apr 16 13:45 blk_1073741832
-rw-rw-r-- 1 ubuntu ubuntu 15 Apr 16 13:45 blk_1073741832_1008.meta
-rw-rw-r-- 1 ubuntu ubuntu 351 Apr 16 13:45 blk_1073741833
-rw-rw-r-- 1 ubuntu ubuntu 11 Apr 16 13:45 blk_1073741833_1009.meta
-rw-rw-r-- 1 ubuntu ubuntu 33811 Apr 16 13:45 blk_1073741834
-rw-rw-r-- 1 ubuntu ubuntu 275 Apr 16 13:45 blk_1073741834_1010.meta
-rw-rw-r-- 1 ubuntu ubuntu 103910 Apr 16 13:45 blk_1073741835
-rw-rw-r-- 1 ubuntu ubuntu 819 Apr 16 13:45 blk_1073741835_1011.meta
ubuntu@ip-172-31-4-165:/var/hadoop_data/datanode/current/BP-687806796-172.31.4.165-1429191194115/current/finalized/subdir0/subdir0$ more blk_1073741832
```



Lecture

Understanding Hive

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
```

Sample HiveQL

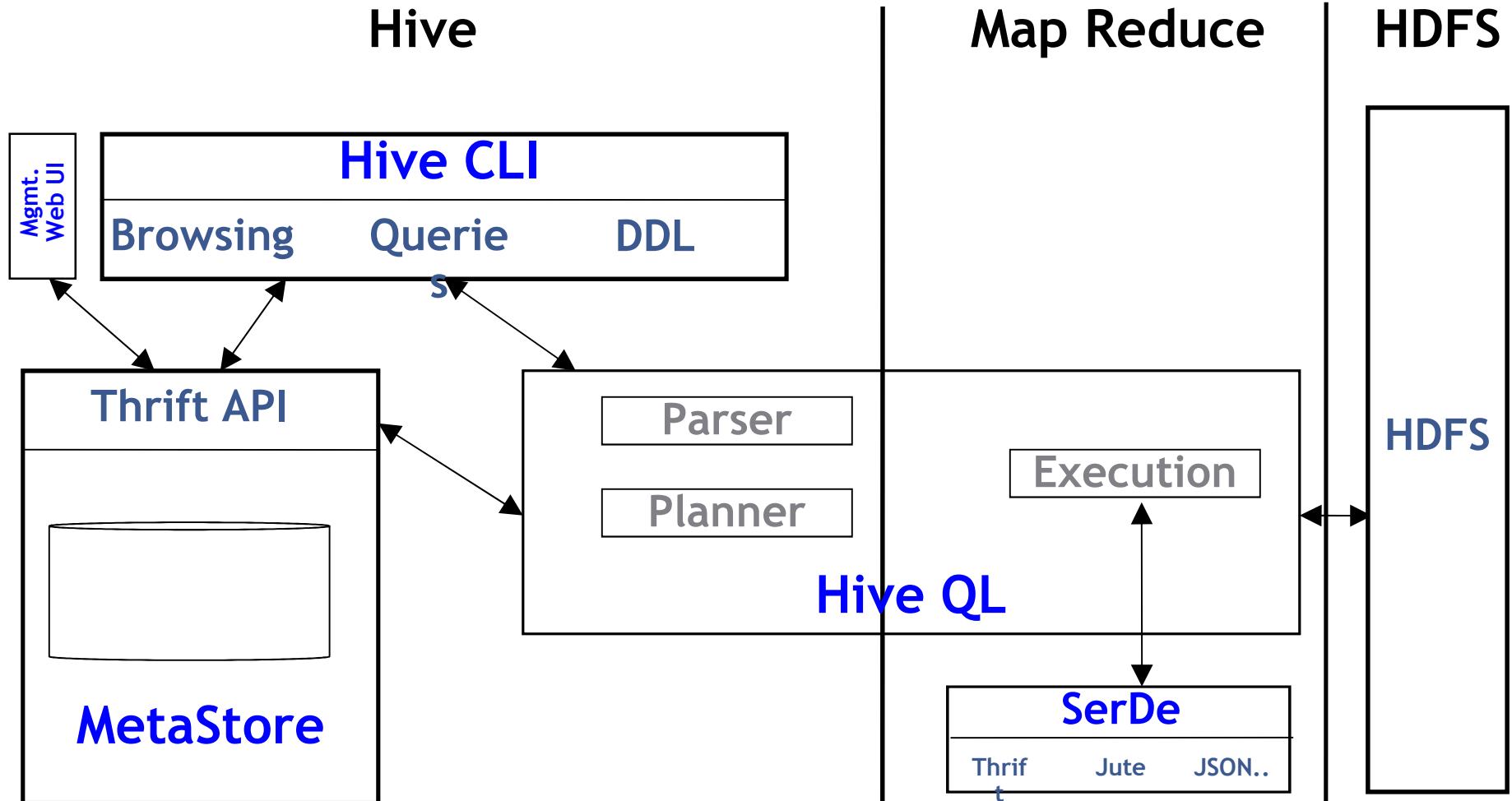
- **Aggregation**
 - `Select count (DISTINCT category) from txnrecords;`
- **Grouping**
 - `Select category, sum(amount) from txnrecords group by category`
- **Inserting Output into another table**
 - `INSERT OVERWRITE TABLE results (SELECT * from txnrecords)`
- **Inserting into local file**
 - `INSERT OVERWRITE LOCAL DIRECTORY'tmp/results' (SELECT * from txnrecords)`

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.

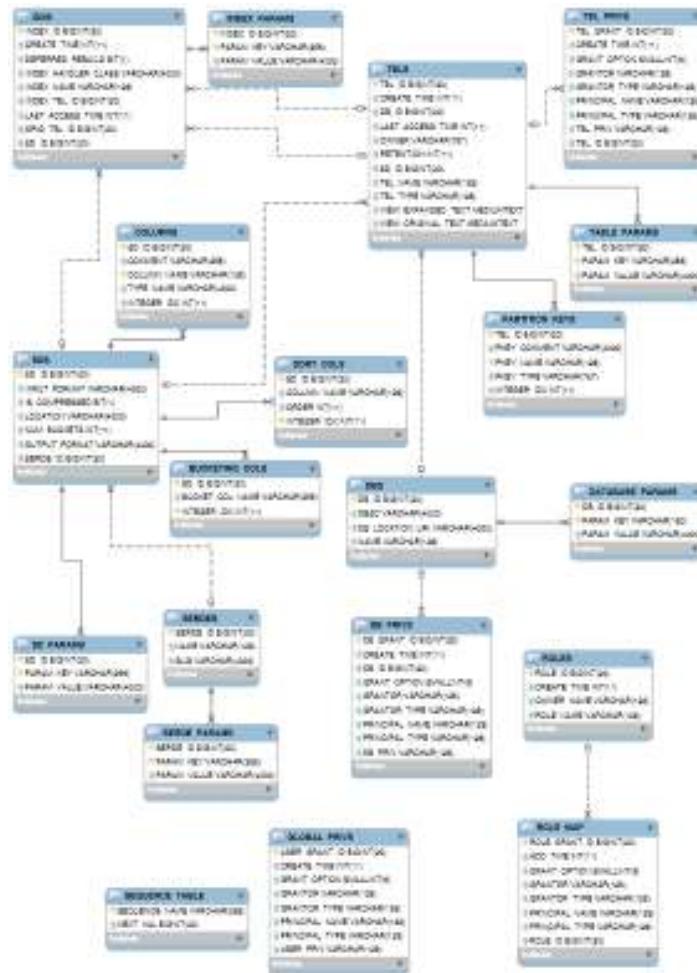
hive.apache.org

Architecture Overview



Hive.apache.org

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). Specifying 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.or
g

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

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

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

```
// load data into table
// NOTE: filepath has to be local to the hive server
// NOTE: /tmp/a.txt is a ctrl-A separated file with two fields per line
String filepath = "/tmp/a.txt";
sql = "load data local inpath '" + filepath + "' into table " + tableName;
System.out.println("Running: " + sql);
res = stmt.executeQuery(sql);

// select * query
sql = "select * from " + tableName;
System.out.println("Running: " + sql);
res = stmt.executeQuery(sql);
while (res.next()) {
    System.out.println(String.valueOf(res.getInt(1)) + "\t" + res.getString(2));
}

// regular hive query
sql = "select count(1) from " + tableName;
System.out.println("Running: " + sql);
res = stmt.executeQuery(sql);
while (res.next()) {
    System.out.println(res.getString(1));
}
}
```

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);

Hands-On: Creating Table and Retrieving Data using Hive

Hive Hands-On Labs

- 1. Installing Hive**
- 2. Configuring / Starting Hive**
- 3. Creating Hive Table**
- 4. Reviewing Hive Table in HDFS**
- 5. Alter and Drop Hive Table**
- 6. Loading Data to Hive Table**
- 7. Querying Data from Hive Table**
- 8. Reviewing Hive Table Content from HDFS Command and WebUI**

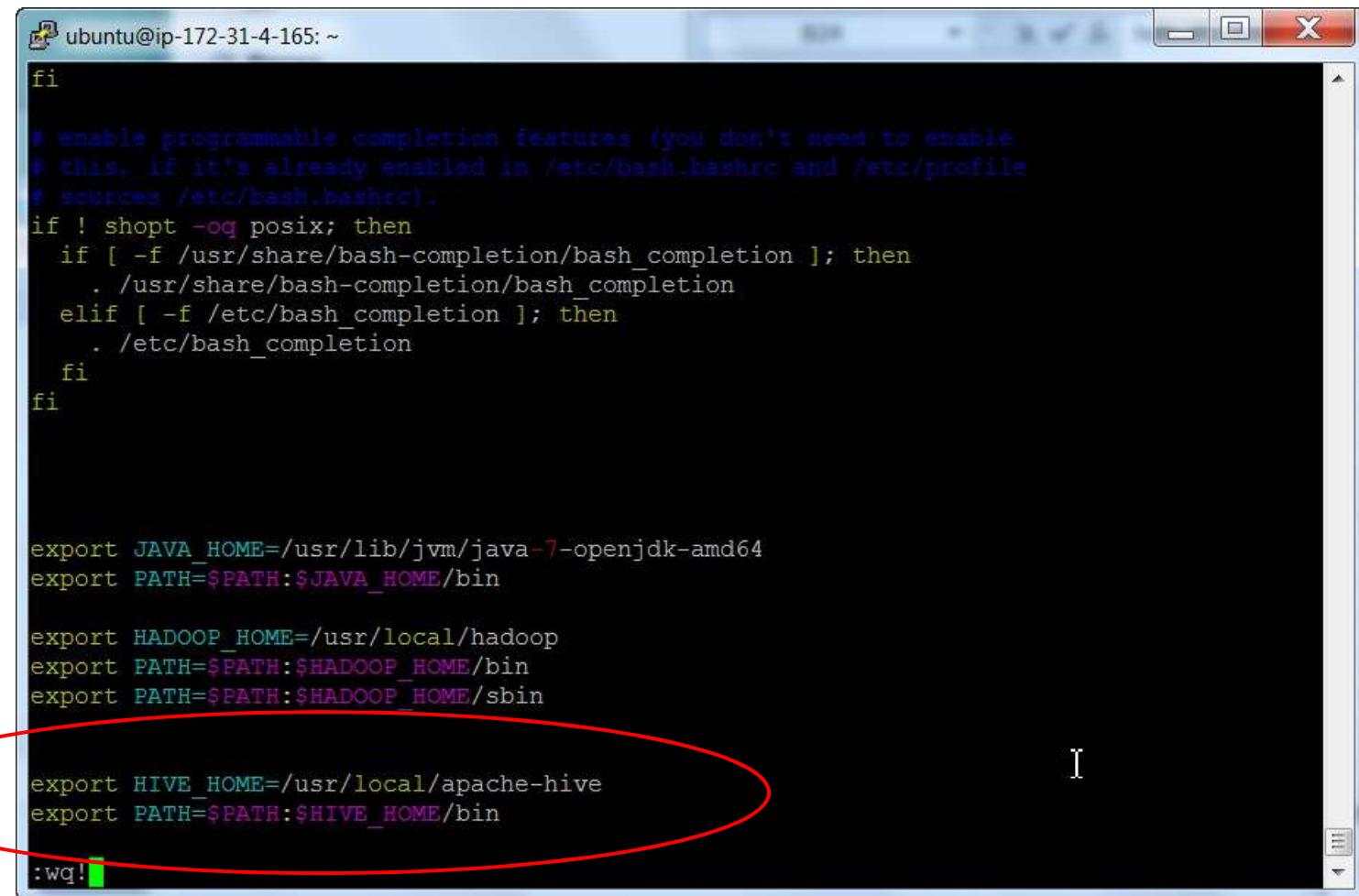
1. Installing Hive

```
$ wget http://www.us.apache.org/dist/hive/hive-  
1.0.0/apache-hive-1.0.0-bin.tar.gz# tar -xvzf apache-hive-  
1.1.0-bin.tar.gz  
  
$ tar -xvf apache-hive-1.0.0-bin.tar.gz  
  
$ mv apache-hive-1.0.0-bin apache-hive  
  
$ sudo mv ./apache-hive /usr/local/  
  
$ chown -R ubuntu:ubuntu /usr/local/apache-hive
```

1. Installing Hive

Edit \$HOME ./bashrc

```
$ vi /home/ubuntu/.bashrc
```



```
ubuntu@ip-172-31-4-165: ~
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

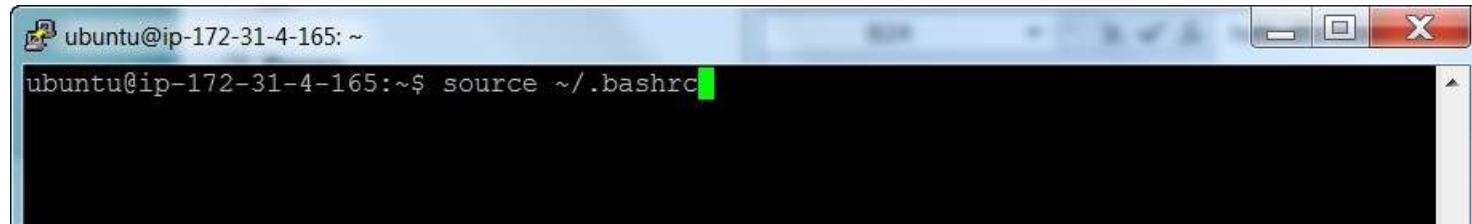
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export HIVE_HOME=/usr/local/apache-hive
export PATH=$PATH:$HIVE_HOME/bin

:wq!
```

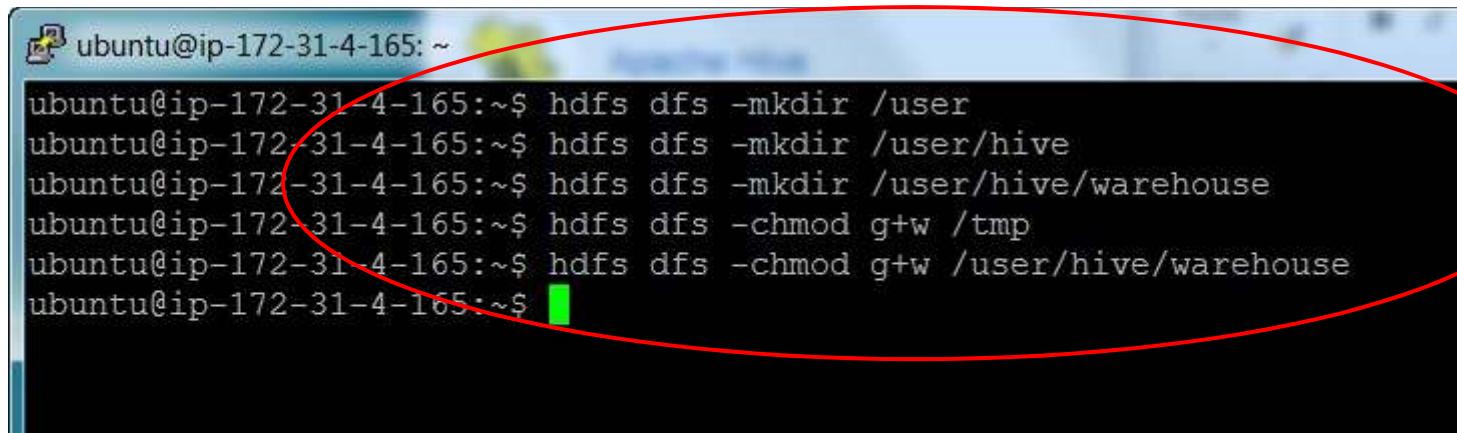
1. Installing Hive

Execute environment variable



```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ source ~/.bashrc
```

Create hive directories



```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ hdfs dfs -mkdir /user
ubuntu@ip-172-31-4-165:~$ hdfs dfs -mkdir /user/hive
ubuntu@ip-172-31-4-165:~$ hdfs dfs -mkdir /user/hive/warehouse
ubuntu@ip-172-31-4-165:~$ hdfs dfs -chmod g+w /tmp
ubuntu@ip-172-31-4-165:~$ hdfs dfs -chmod g+w /user/hive/warehouse
ubuntu@ip-172-31-4-165:~$
```

2. Configuring Hive

The image shows two terminal windows side-by-side. The top window displays a command to copy a template file and then edit it:

```
ubuntu@ip-172-31-4-165:~$ cp /usr/local/apache-hive/conf/hive-env.sh.template /usr/local/apache-hive/conf/hive-env.sh
ubuntu@ip-172-31-4-165:~$ vi /usr/local/apache-hive/conf/hive-env.sh
```

The bottom window shows the contents of the edited `hive-env.sh` file. Several lines have been highlighted with red circles:

```
# else
#   export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib -XX:NewRatio=12 -Xms1Gm -XX:MaxHeapFreeRatio=40 -XX:MinHeapFreeRatio=15 -XX:-UseGCOverheadLimit"
# fi
# fi

# the heap size of the jvm started by hive shell script can be controlled via:
#
# export HADOOP_HEAPSIZE=1024
#
# larger heap size may be required when running queries over large numbers of files or partitions.
# By default hive shell scripts use a heap size of 256 (mb). Larger heap size would also be
# appropriate for hive server (hwi etc).

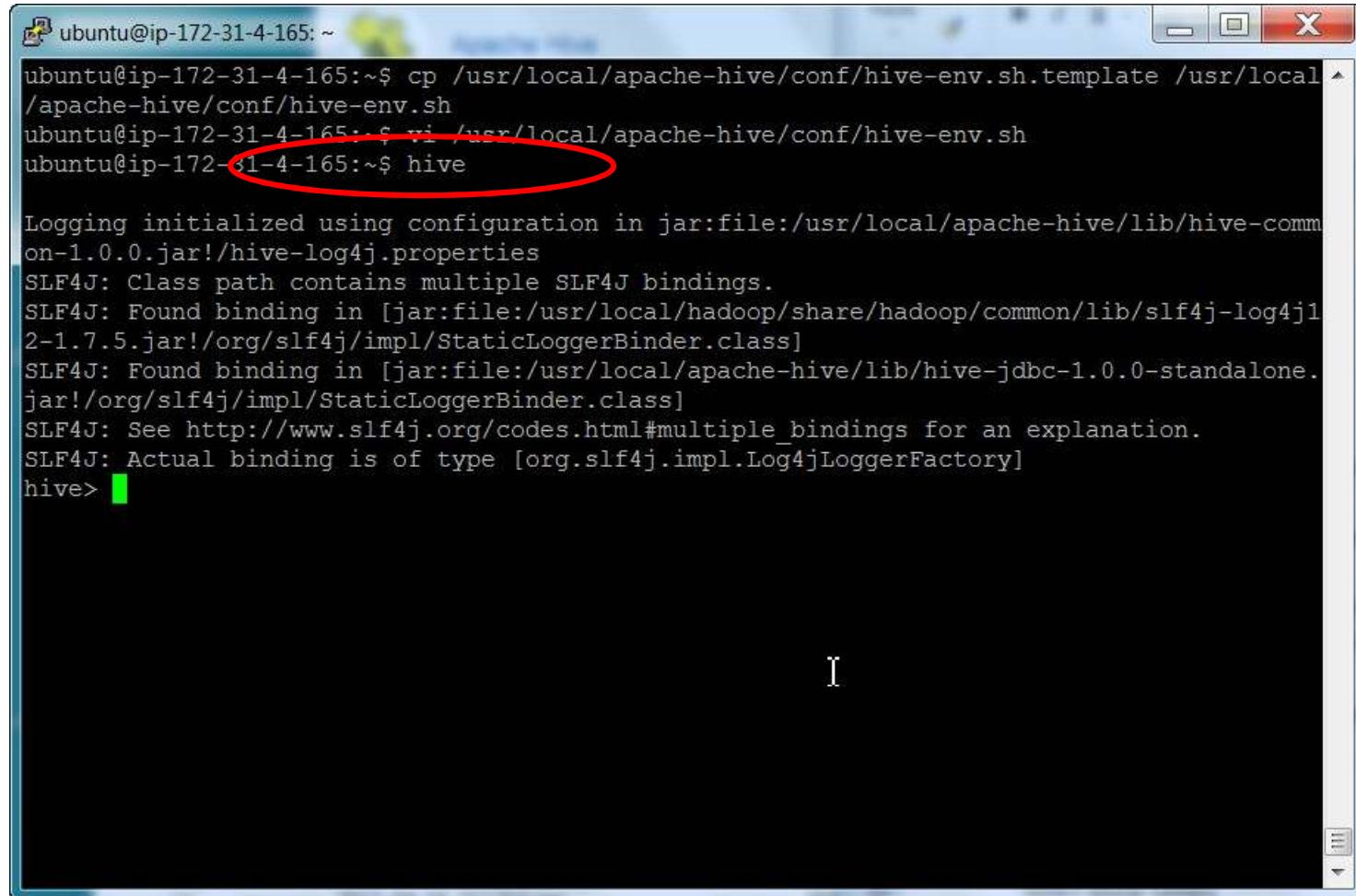
# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/usr/local/hadoop

# Hive configuration directory can be controlled by:
# export HIVE_CONF_DIR=

# Folder containing extra libraries required for hive compilation/execution can be controlled by:
# export HIVE_AUX_JARS_PATH=
```

A red circle highlights the line `HADOOP_HOME=/usr/local/hadoop`. Another red circle highlights the line `:wq!`.

2. Starting Hive

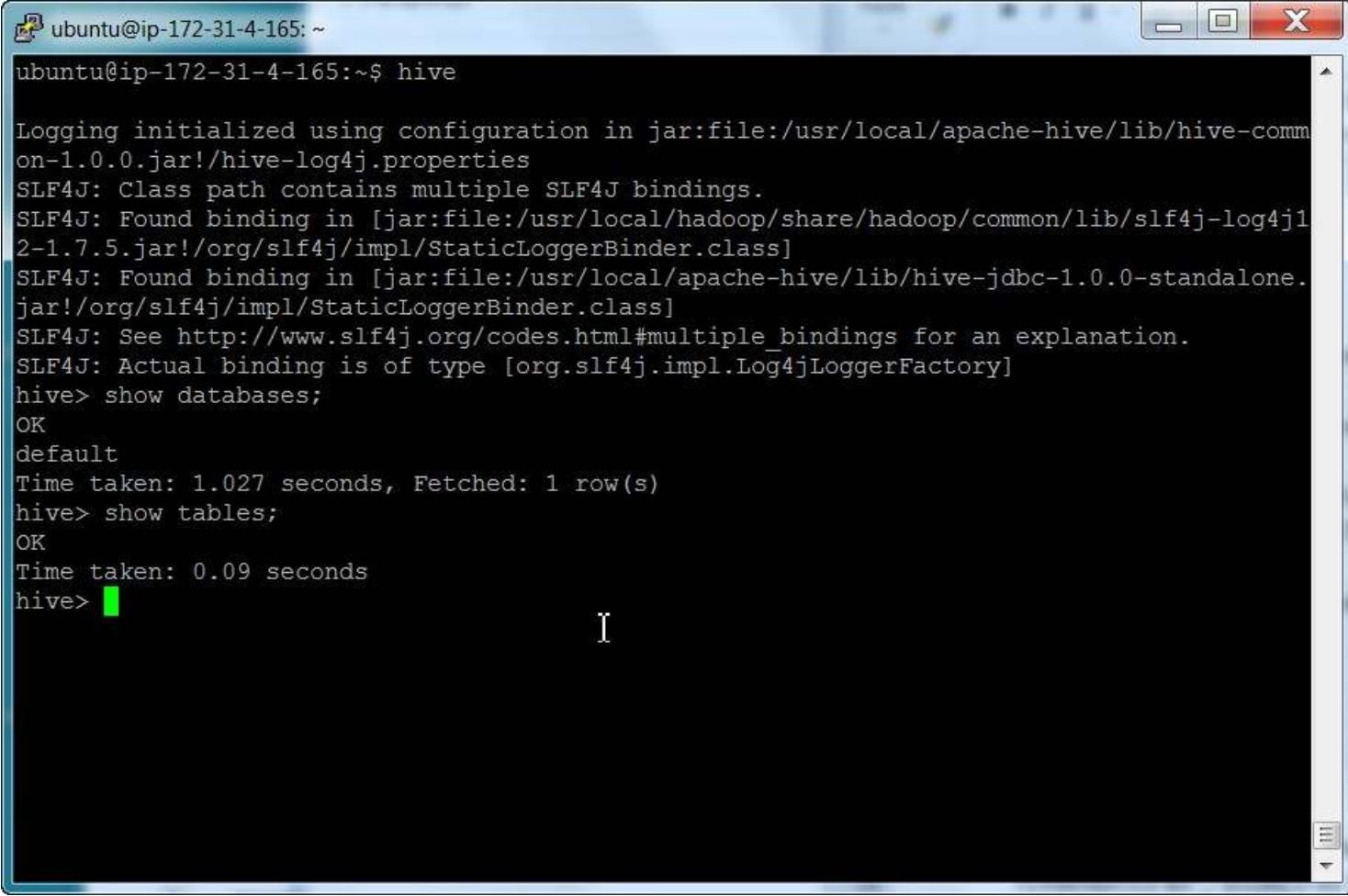


The screenshot shows a terminal window on an Ubuntu system. The user has run several commands to copy the Hive environment configuration file and edit it, then started the Hive service. A red oval highlights the command 'hive' which was just entered.

```
ubuntu@ip-172-31-4-165:~$ cp /usr/local/apache-hive/conf/hive-env.sh.template /usr/local/apache-hive/conf/hive-env.sh
ubuntu@ip-172-31-4-165:~$ vi /usr/local/apache-hive/conf/hive-env.sh
ubuntu@ip-172-31-4-165:~$ hive

Logging initialized using configuration in jar:file:/usr/local/apache-hive/lib/hive-common-1.0.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/apache-hive/lib/hive-jdbc-1.0.0-standalone.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]
hive>
```

Show Hive Table



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the following text:

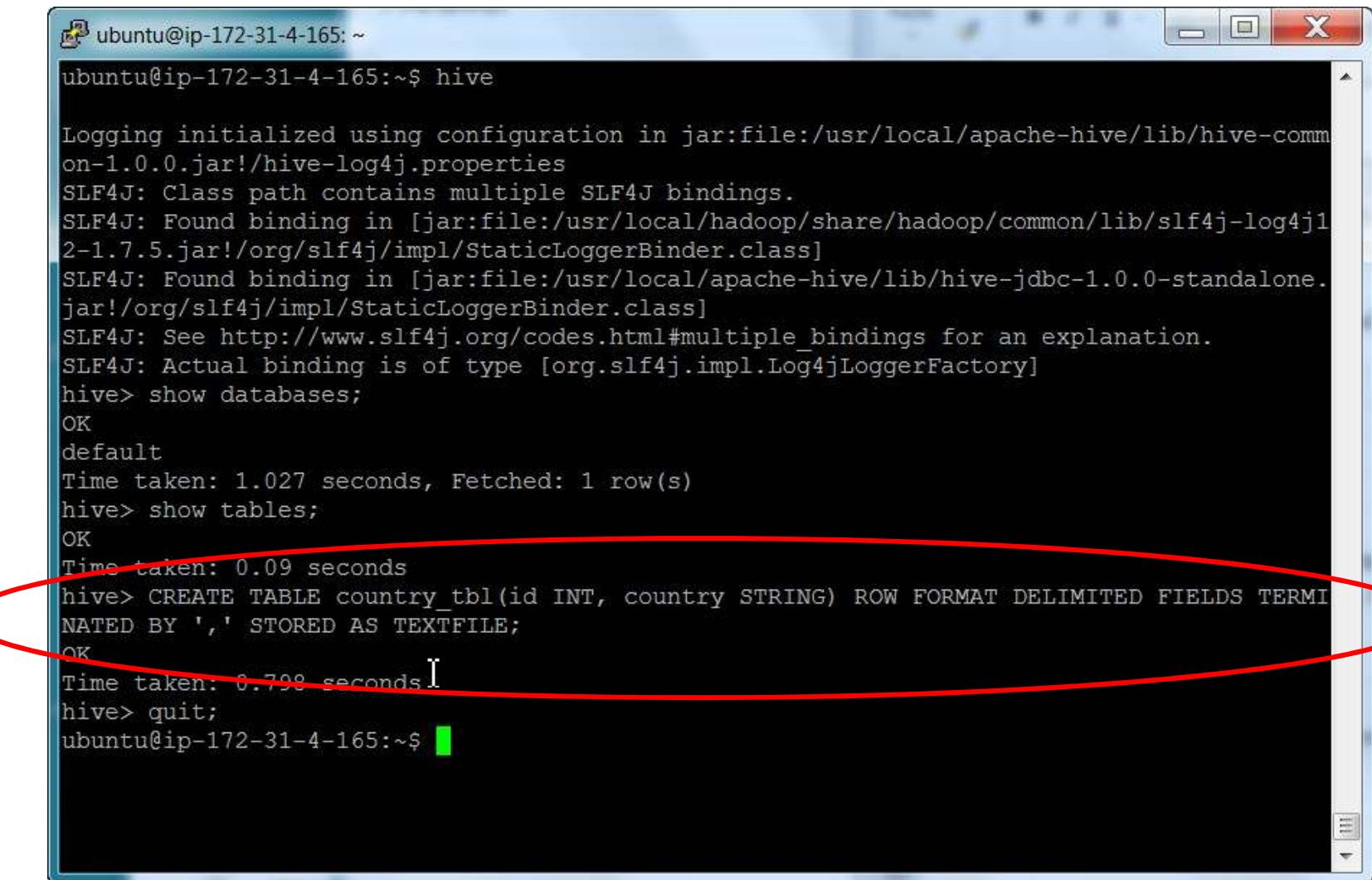
```
ubuntu@ip-172-31-4-165:~$ hive
Logging initialized using configuration in jar:/file:/usr/local/apache-hive/lib/hive-common-1.0.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:/file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:/file:/usr/local/apache-hive/lib/hive-jdbc-1.0.0-standalone.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]
hive> show databases;
OK
default
Time taken: 1.027 seconds, Fetched: 1 row(s)
hive> show tables;
OK
Time taken: 0.09 seconds
hive>
```

3. Creating Hive Table

```
hive (default)> CREATE TABLE test_tbl(id INT, country STRING) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 4.069 seconds
hive (default)> show tables;
OK
test_tbl
Time taken: 0.138 seconds
hive (default)> describe test_tbl;
OK
id      int
country string
Time taken: 0.147 seconds
hive (default)>
```

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

3. Creating Hive Table



```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ hive

Logging initialized using configuration in jar:/usr/local/apache-hive/lib/hive-comm
on-1.0.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j1
2-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:/usr/local/apache-hive/lib/hive-jdbc-1.0.0-standalone.
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]
hive> show databases;
OK
default
Time taken: 1.027 seconds, Fetched: 1 row(s)
hive> show tables;
OK
Time taken: 0.09 seconds
hive> CREATE TABLE country_tbl(id INT, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
OK
Time taken: 0.799 seconds
hive> quit;
ubuntu@ip-172-31-4-165:~$
```

4. Reviewing Hive Table in HDFS

```
[hdadmin@localhost hdadmin]$ hadoop fs -ls /user/hive/warehouse  
Found 1 items  
drwxr-xr-x  - hdadmin supergroup          0 2013-03-17 17:51  
/user/hive/warehouse/test_tbl  
[hdadmin@localhost hdadmin]$
```

Contents of directory </user/hive/warehouse>

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
test_tbl	dir				2013-03-17 17:51	rwxr-xr-x	hdadmin	supergroup

Go back to DFS home

Local logs

[Log directory](#)

This is [Apache Hadoop](#) release 0.20.205.0

Review Hive Table from HDFS WebUI

5. Alter and Drop Hive Table

```
hive (default)> alter table test_tbl add columns (remarks STRING);  
  
hive (default)> describe test_tbl;  
OK  
id      int  
country string  
remarks string  
Time taken: 0.077 seconds  
hive (default)> drop table test_tbl;  
OK  
Time taken: 0.9 seconds
```

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

6. Loading Data to Hive Table

Please get the country_data.csv from your instructor.

The screenshot shows a terminal window with two tabs. The top tab, circled in red, shows the command `vi country_data.csv`. The bottom tab, also circled in red, shows the output of the `hive` command:

```
ubuntu@ip-172-31-4-165:~$ hive

Logging initialized using configuration in jar:file:/usr/local/apache-hive/lib/hive-common-1.0.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j1-2-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/apache-hive/lib/hive-jdbc-1.0.0-standalone.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]
hive> LOAD DATA LOCAL INPATH '/home/ubuntu/country_data.csv' INTO TABLE country_tbl;
```

6. Loading Data to Hive Table

Please get the country_data.csv from your instructor.

The screenshot shows a terminal window with two tabs. The top tab, circled in red, shows the command `vi country_data.csv`. The bottom tab, also circled in red, shows the output of a Hive session. The Hive session starts with the command `hive`, followed by several SLF4J log messages about multiple bindings. Then, the command `LOAD DATA LOCAL INPATH '/home/ubuntu/country_data.csv' INTO TABLE country_tbl;` is run, which triggers a warning about the actual binding type. Finally, the data is loaded into the table, and the session ends with the command `hive>`.

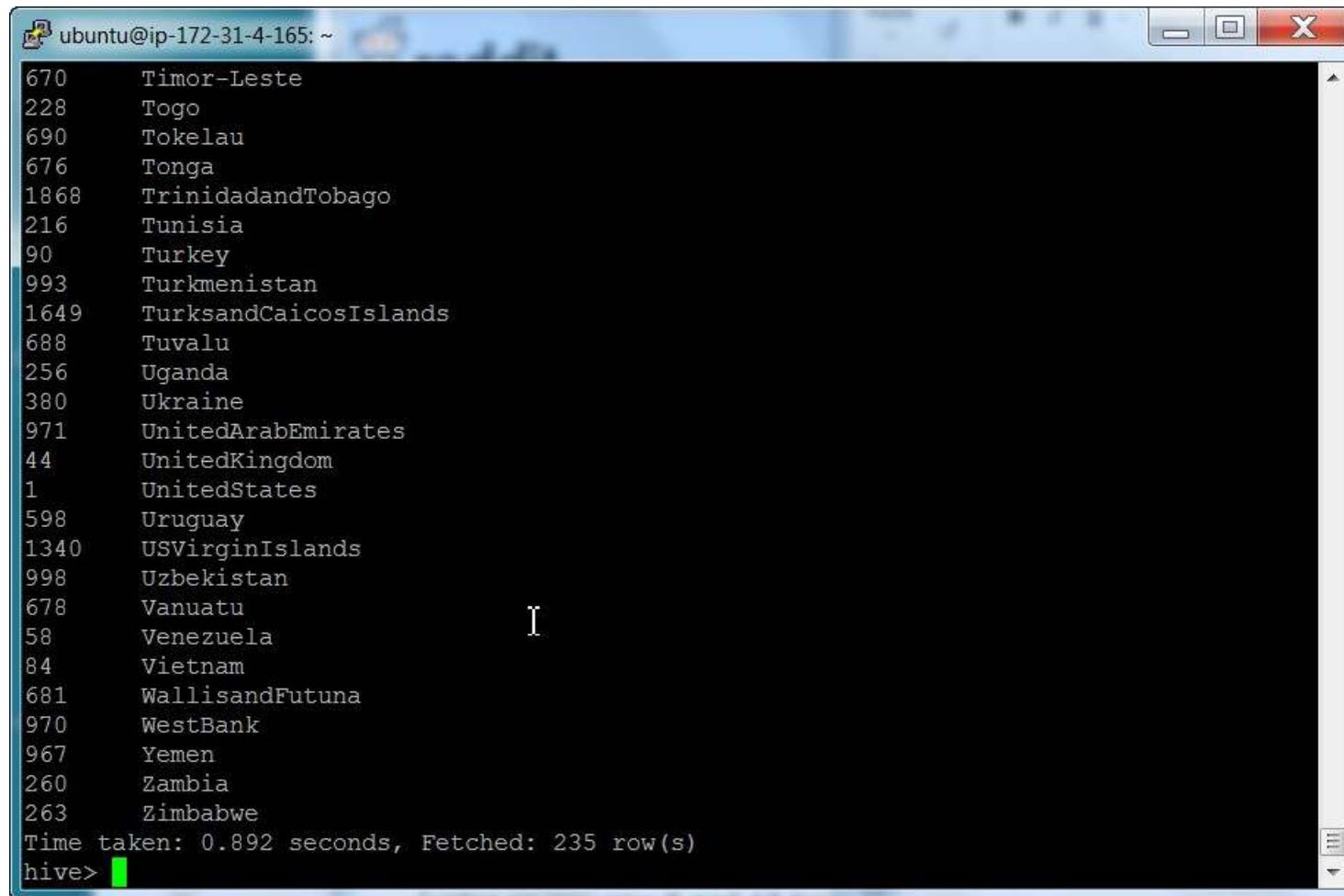
```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ vi country_data.csv

ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ hive

Logging initialized using configuration in jar:/usr/local/apache-hive/lib/hive-common-1.0.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:/file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:/file:/usr/local/apache-hive/lib/hive-jdbc-1.0.0-standalone.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]
hive> LOAD DATA LOCAL INPATH '/home/ubuntu/country_data.csv' INTO TABLE country_tbl;
Loading data to table default.country_tbl
Table default.country_tbl stats: [numFiles=1, totalSize=3250]
OK
Time taken: 3.746 seconds
hive>
```

7. Querying Data from Hive Table

```
hive> select * from country_tbl;
```



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the output of a Hive query. A red oval highlights the command "hive> select * from country_tbl;". The output lists 235 countries with their corresponding IDs. The terminal window has a standard Windows-style title bar with minimize, maximize, and close buttons.

ID	Country
670	Timor-Leste
228	Togo
690	Tokelau
676	Tonga
1868	TrinidadandTobago
216	Tunisia
90	Turkey
993	Turkmenistan
1649	TurksandCaicosIslands
688	Tuvalu
256	Uganda
380	Ukraine
971	UnitedArabEmirates
44	UnitedKingdom
1	UnitedStates
598	Uruguay
1340	USVirginIslands
998	Uzbekistan
678	Vanuatu
58	Venezuela
84	Vietnam
681	WallisandFutuna
970	WestBank
967	Yemen
260	Zambia
263	Zimbabwe

Time taken: 0.892 seconds, Fetched: 235 row(s)

```
hive>
```

8. Reviewing Hive Table Content from HDFS Command and WebUI

The screenshot shows a web browser window titled "Browsing HDFS" with the URL 52.11.192.136:50070/explorer.html#/user/hive/warehouse/country_tbl. The page has a green header bar with tabs for Hadoop, Overview, Datanodes, Snapshot, Startup Progress, and Utilities. The main content area is titled "Browse Directory" and displays a table of files in the directory `/user/hive/warehouse/country_tbl`. The table columns are Permission, Owner, Group, Size, Replication, Block Size, and Name. One file is listed: `country_data.csv`, which has permissions `-rw-r--r--`, owner `ubuntu`, group `supergroup`, size `3.17 KB`, replication `1`, and block size `128 MB`. At the bottom of the table, it says "Hadoop, 2014.". In the bottom left corner of the browser, there is a download progress bar for a file named "part-r-00000".

Permission	Owner	Group	Size	Replication	Block Size	Name
<code>-rw-r--r--</code>	ubuntu	supergroup	3.17 KB	1	128 MB	country_data.csv

Hadoop, 2014.



Lecture: Pig

Introduction

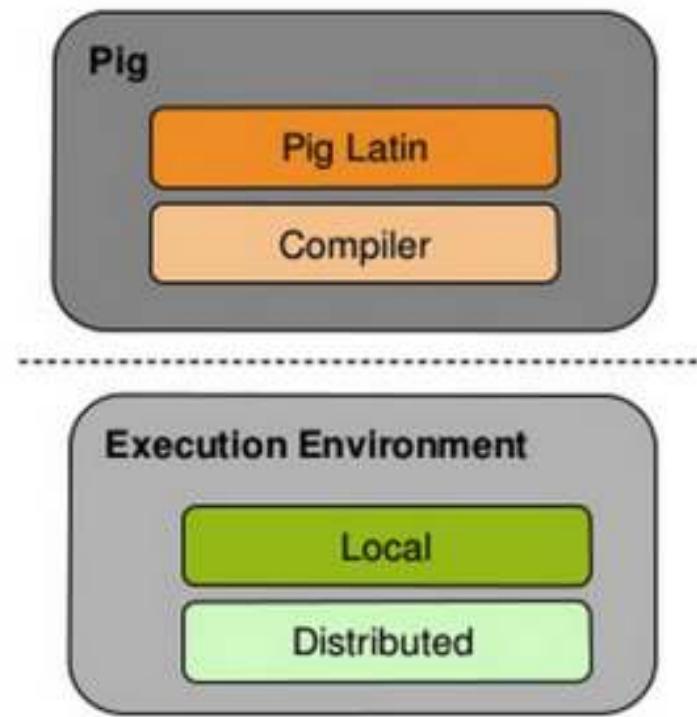
A high-level platform for creating MapReduce programs Using Hadoop



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 Components**
 - Language (Pig Latin)
 - Compiler
- **Two Execution Environments**
 - **Local**
pig -x local
 - **Distributed**
pig -x mapreduce



Running Pig

- **Script**

pig myscript

- **Command line (Grunt)**

pig

- **Embedded**

Writing a java program

Pig Dataflow Organization

1. A LOAD statement reads data from the file system.
2. A series of "transformation" statements process the data.
3. A STORE statement writes output to the file system; or, a DUMP statement displays output to the screen.

Pig Latin

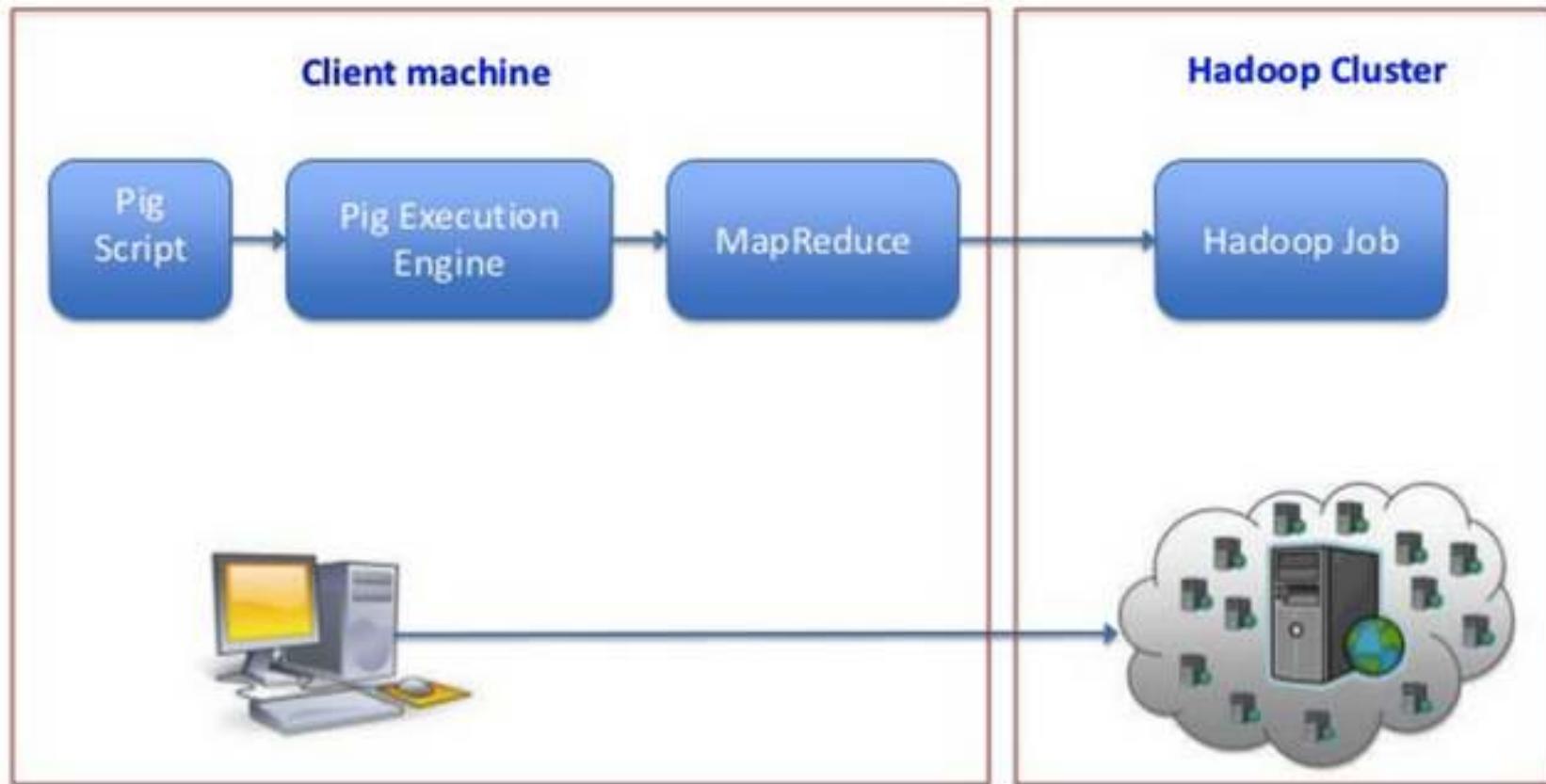
```
A = load 'hdi-data.csv' using PigStorage(',') AS  
  (id:int, country:chararray, hdi:float, lifeex:int,  
   mysch:int, eysch:int, gni:int);  
B = FILTER A BY gni > 2000;  
C = ORDER B BY gni;  
dump C;
```

Pig Command Language

Pig Command	What it does
load	Read data from file system.
store	Write data to file system.
foreach	Apply expression to each record and output one or more records.
filter	Apply predicate and remove records that do not return true.
group/cogroup	Collect records with the same key from one or more inputs.
join	Join two or more inputs based on a key.
order	Sort records based on a key.
distinct	Remove duplicate records.
union	Merge two data sets.
split	Split data into 2 or more sets, based on filter conditions.
stream	Send all records through a user provided binary.
dump	Write output to stdout.
limit	Limit the number of records.

Cloudera, 2009

Pig Execution Stages



Source Introduction to Apache Hadoop-Pig: PrashantKommireddi

Why Pig?

- **Makes writing Hadoop jobs easier**
 - 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

Pig v.s. Hive



<i>Characteristic</i>	<i>Pig</i>	<i>Hive</i>
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

Pig v.s. Hive

Pig Latin

```
countryrs = load '/user/gharriso/PIG_COUNTRIES' AS  
    (country_id, country_name , country_subregion , region);  
  
customers= load '/user/gharriso/PIG_CUSTOMERS' AS  
    (cust_id,first_name, last_name, gender, yob, marital, postcode,city,country_id);  
  
asianCountryrs = filter countryrs by region matches 'Asia';  
  
joined = join customers by country_id, asianCountryrs by country_id;  
  
grouped = group joined by country_name;  
  
agged = foreach grouped generate group, COUNT(joined.customers::cust_id);  
  
morethan500cust = filter agged by $1 > 500;  
  
ordered =order morethan500cust by $1 desc;  
  
dump ordered;
```

SQL or Hive QL

```
SELECT country_name,COUNT(cust_id) AS cust_count  
FROM countries co  
JOIN customers cu  
ON (co.country_id=cu.country_id)  
WHERE country_region='Asia'  
GROUP BY country_name  
HAVING COUNT(cust_id)>500  
ORDER BY cust_count DESC
```

<http://guyharrison.net>

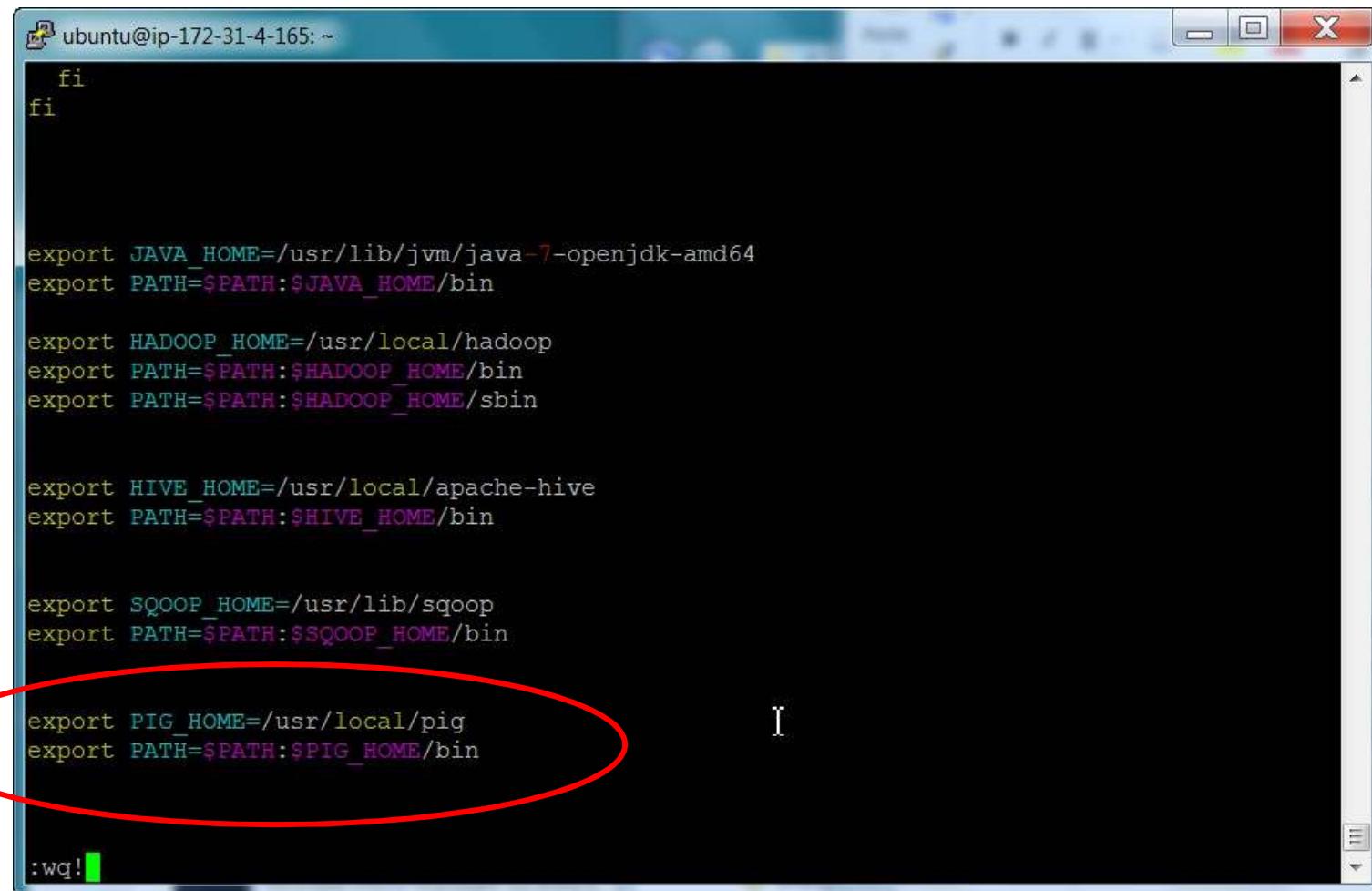
Hands-On: Running a Pig script

Installing Pig

```
$ wget http://apache.cs.utah.edu/pig/pig-0.14.0/pig-0.14.0.tar.gz  
$ tar -xvf pig-0.14.0.tar.gz  
$ mv pig-0.14.0 pig  
$ sudo mv pig /usr/local/
```

Edit System Environment Variables

```
vi ./bashrc
```



```
ubuntu@ip-172-31-4-165: ~
fi
fi

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export HIVE_HOME=/usr/local/apache-hive
export PATH=$PATH:$HIVE_HOME/bin

export SQOOP_HOME=/usr/lib/sqoop
export PATH=$PATH:$SQOOP_HOME/bin

export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin

:wq!
```

Execute System Environment

```
source ~/.bashrc
```

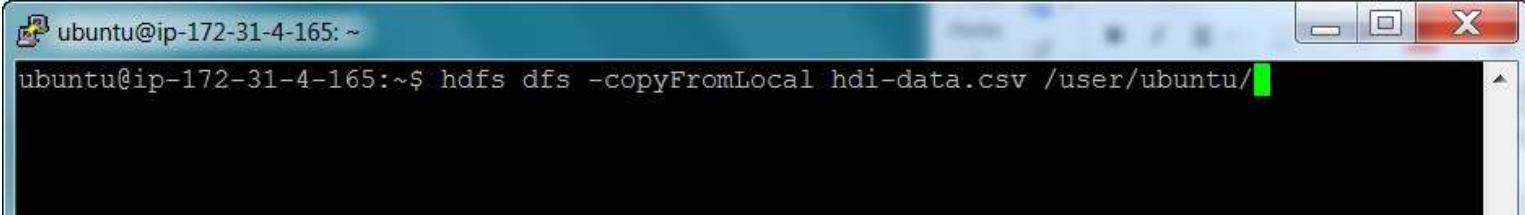
Download sample data



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the following text:

```
ubuntu@ip-172-31-4-165:~$ source ~/.bashrc
ubuntu@ip-172-31-4-165:~$ wget https://www.dropbox.com/s/pp168a6oiwqkxyu/hdi-data.csv
```

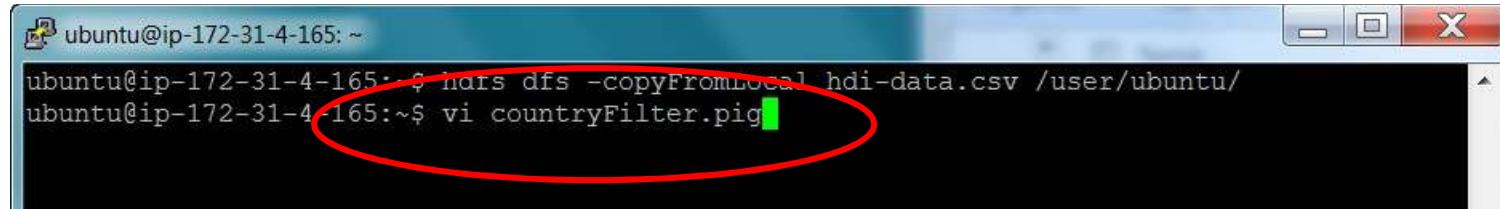
Import sample data to HDFS



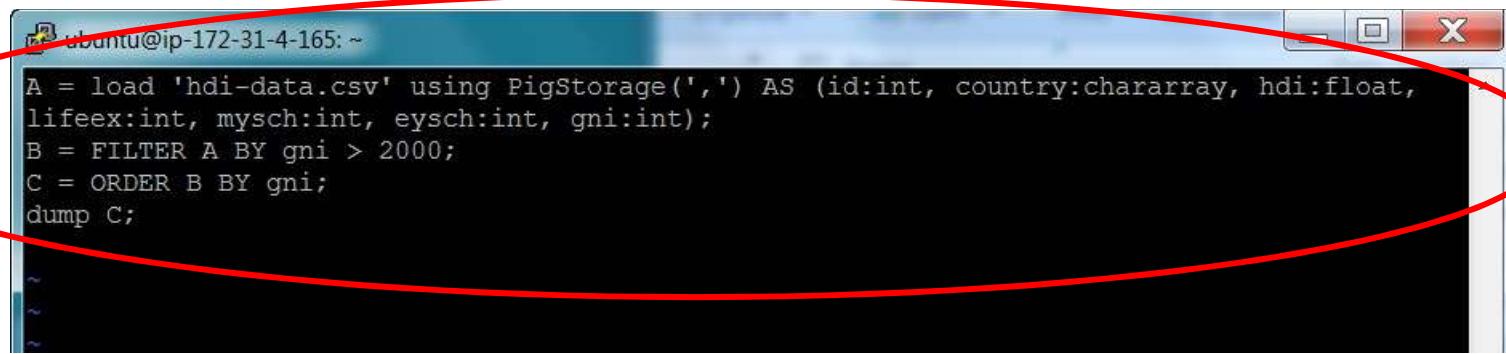
A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the following text:

```
ubuntu@ip-172-31-4-165:~$ hdfs dfs -copyFromLocal hdi-data.csv /user/ubuntu/
```

Write your own Pig script

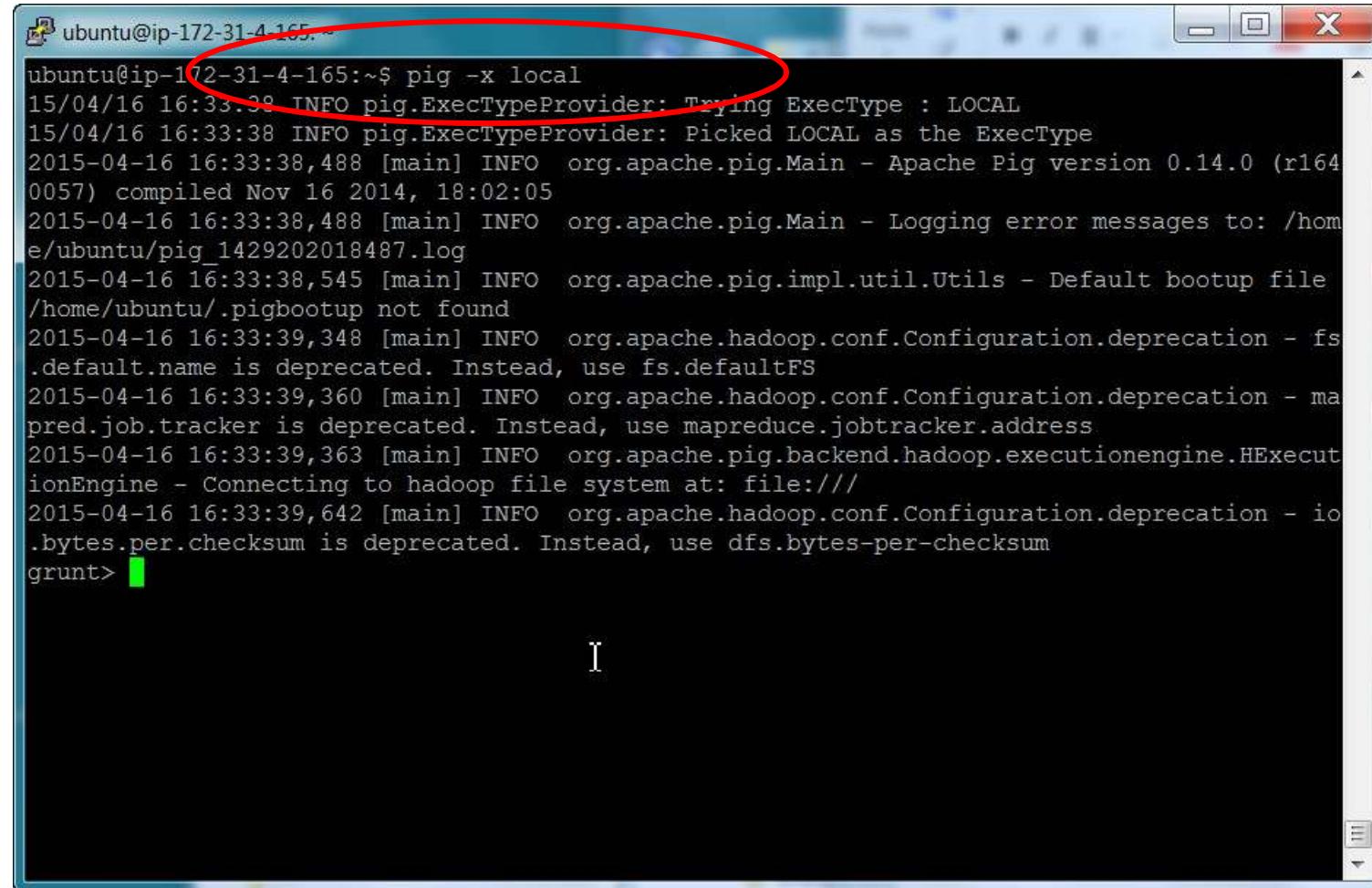


```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165: ~ hdfs dfs -copyFromLocal hdi-data.csv /user/ubuntu/
ubuntu@ip-172-31-4-165: ~$ vi countryFilter.pig
```



```
A = load 'hdi-data.csv' using PigStorage(',') AS (id:int, country:chararray, hdi:float,
lifeex:int, mysch:int, eysch:int, gni:int);
B = FILTER A BY gni > 2000;
C = ORDER B BY gni;
dump C;
```

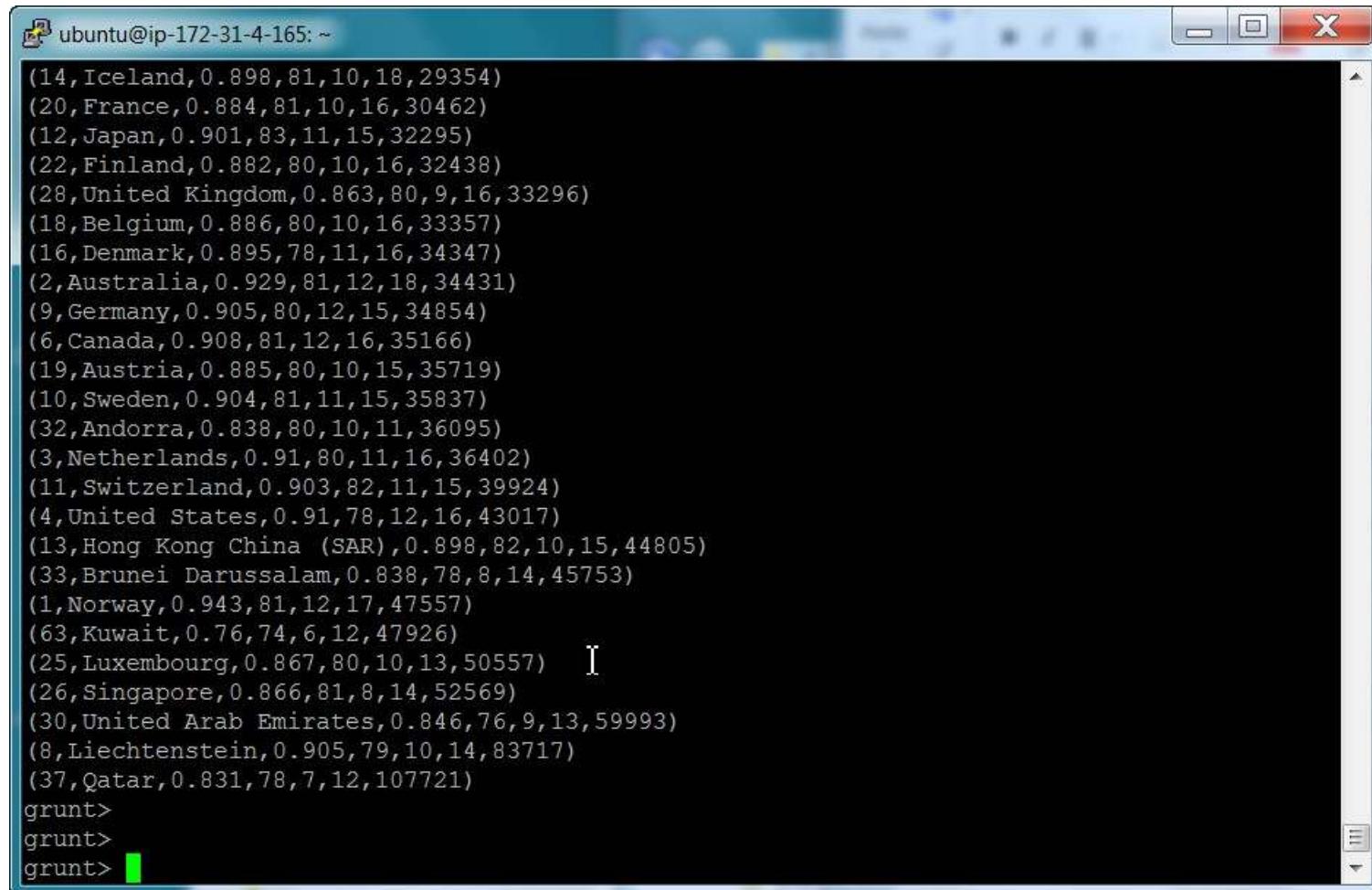
Running Pig shell



```
ubuntu@ip-172-31-4-165:~$ pig -x local
15/04/16 16:33:38 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
15/04/16 16:33:38 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2015-04-16 16:33:38,488 [main] INFO org.apache.pig.Main - Apache Pig version 0.14.0 (r164
0057) compiled Nov 16 2014, 18:02:05
2015-04-16 16:33:38,488 [main] INFO org.apache.pig.Main - Logging error messages to: /hom
e/ubuntu/pig_1429202018487.log
2015-04-16 16:33:38,545 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file
/home/ubuntu/.pigbootup not found
2015-04-16 16:33:39,348 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs
.default.name is deprecated. Instead, use fs.defaultFS
2015-04-16 16:33:39,360 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - ma
pred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2015-04-16 16:33:39,363 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecut
ionEngine - Connecting to hadoop file system at: file:///
2015-04-16 16:33:39,642 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io
.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> [REDACTED]
```

Execute Pig Script

```
run countryFilter.pig
```

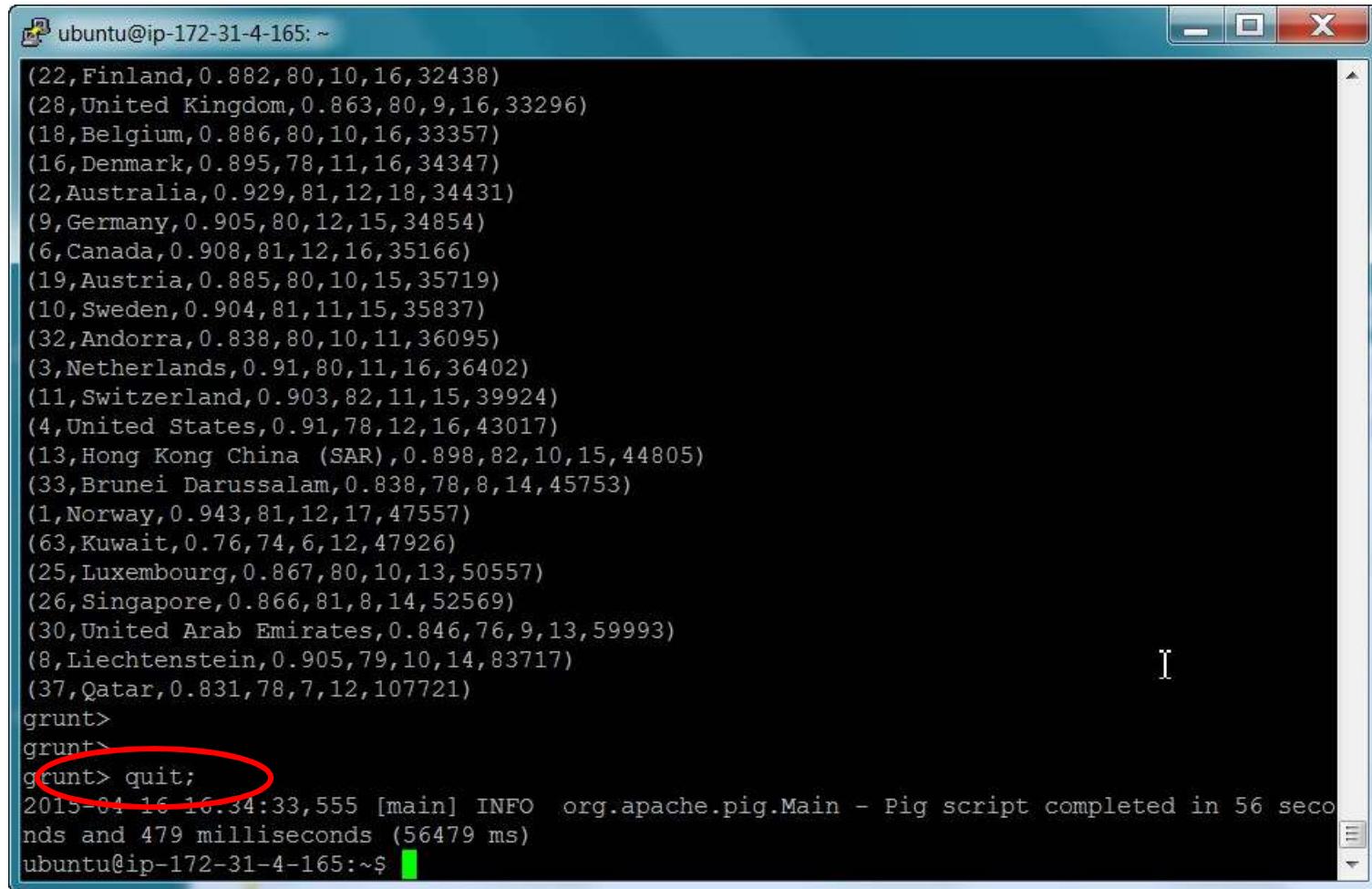


The screenshot shows a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the output of a Pig script named "countryFilter.pig". The output consists of a list of tuples, each representing a country and its corresponding values. The tuples are as follows:

- (14, Iceland, 0.898, 81, 10, 18, 29354)
- (20, France, 0.884, 81, 10, 16, 30462)
- (12, Japan, 0.901, 83, 11, 15, 32295)
- (22, Finland, 0.882, 80, 10, 16, 32438)
- (28, United Kingdom, 0.863, 80, 9, 16, 33296)
- (18, Belgium, 0.886, 80, 10, 16, 33357)
- (16, Denmark, 0.895, 78, 11, 16, 34347)
- (2, Australia, 0.929, 81, 12, 18, 34431)
- (9, Germany, 0.905, 80, 12, 15, 34854)
- (6, Canada, 0.908, 81, 12, 16, 35166)
- (19, Austria, 0.885, 80, 10, 15, 35719)
- (10, Sweden, 0.904, 81, 11, 15, 35837)
- (32, Andorra, 0.838, 80, 10, 11, 36095)
- (3, Netherlands, 0.91, 80, 11, 16, 36402)
- (11, Switzerland, 0.903, 82, 11, 15, 39924)
- (4, United States, 0.91, 78, 12, 16, 43017)
- (13, Hong Kong China (SAR), 0.898, 82, 10, 15, 44805)
- (33, Brunei Darussalam, 0.838, 78, 8, 14, 45753)
- (1, Norway, 0.943, 81, 12, 17, 47557)
- (63, Kuwait, 0.76, 74, 6, 12, 47926)
- (25, Luxembourg, 0.867, 80, 10, 13, 50557)]
- (26, Singapore, 0.866, 81, 8, 14, 52569)
- (30, United Arab Emirates, 0.846, 76, 9, 13, 59993)
- (8, Liechtenstein, 0.905, 79, 10, 14, 83717)
- (37, Qatar, 0.831, 78, 7, 12, 107721)

At the bottom of the terminal window, there are three "grunt>" prompts followed by a green cursor.

Quit from Pig Shell



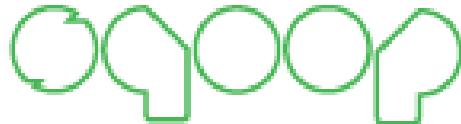
The screenshot shows a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window displays a list of tuples representing data, followed by the "grunt>" prompt, and the command "quit;". A red oval highlights the "quit;" command. The terminal concludes with the message "[main] INFO org.apache.pig.Main - Pig script completed in 56 seconds and 479 milliseconds (56479 ms)".

```
(22,Finland,0.882,80,10,16,32438)
(28,United Kingdom,0.863,80,9,16,33296)
(18,Belgium,0.886,80,10,16,33357)
(16,Denmark,0.895,78,11,16,34347)
(2,Australia,0.929,81,12,18,34431)
(9,Germany,0.905,80,12,15,34854)
(6,Canada,0.908,81,12,16,35166)
(19,Austria,0.885,80,10,15,35719)
(10,Sweden,0.904,81,11,15,35837)
(32,Andorra,0.838,80,10,11,36095)
(3,Netherlands,0.91,80,11,16,36402)
(11,Switzerland,0.903,82,11,15,39924)
(4,United States,0.91,78,12,16,43017)
(13,Hong Kong China (SAR),0.898,82,10,15,44805)
(33,Brunei Darussalam,0.838,78,8,14,45753)
(1,Norway,0.943,81,12,17,47557)
(63,Kuwait,0.76,74,6,12,47926)
(25,Luxembourg,0.867,80,10,13,50557)
(26,Singapore,0.866,81,8,14,52569)
(30,United Arab Emirates,0.846,76,9,13,59993)
(8,Liechtenstein,0.905,79,10,14,83717)
(37,Qatar,0.831,78,7,12,107721)
grunt>
grunt>
grunt> quit;
2015-04-16 10:34:33,555 [main] INFO org.apache.pig.Main - Pig script completed in 56 seconds and 479 milliseconds (56479 ms)
ubuntu@ip-172-31-4-165:~$
```



Lecture: Sqoop

Introduction

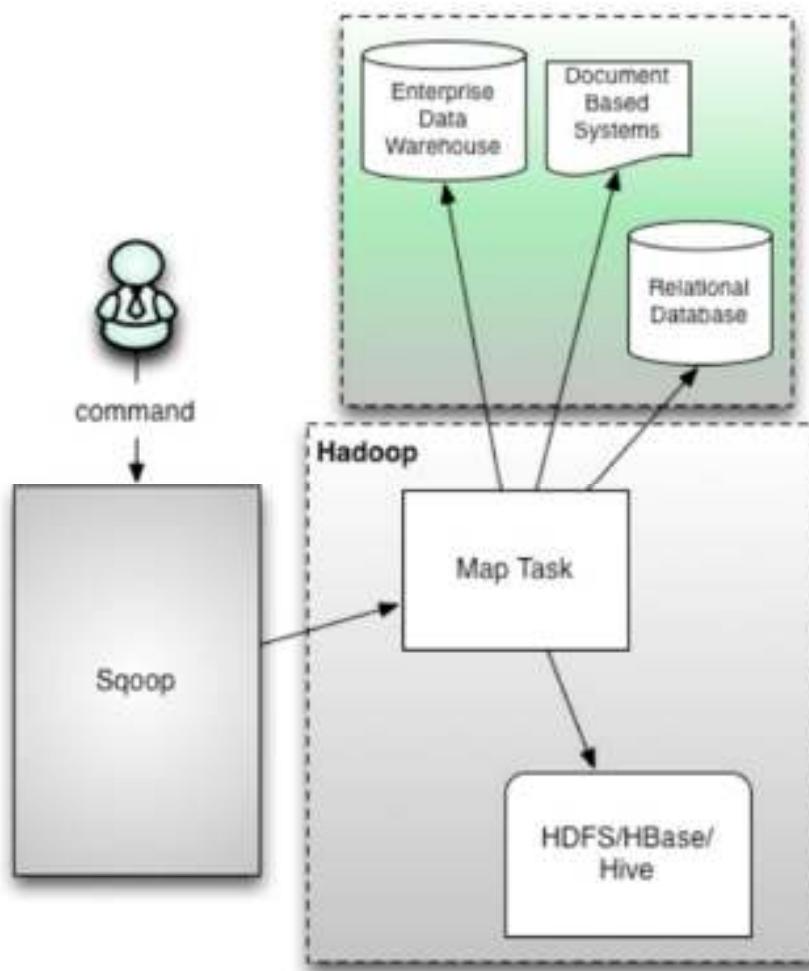


Sqoop (“SQL-to-Hadoop”) is a straightforward command-line tool with the following capabilities:

- **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.or
g



Hands-On: Loading Data from DBMS to Hadoop HDFS

Sqoop Hands-On Labs

- 1. Loading Data into MySQL DB**
- 2. Installing Sqoop**
- 3. Configuring Sqoop**
- 4. Installing DB driver for Sqoop**
- 5. Importing data from MySQL to Hadoop**
- 6. Reviewing HDFS data**

1. Loading Data into MySQL DB

Create sample sql scripts

Please get the script from your instructor

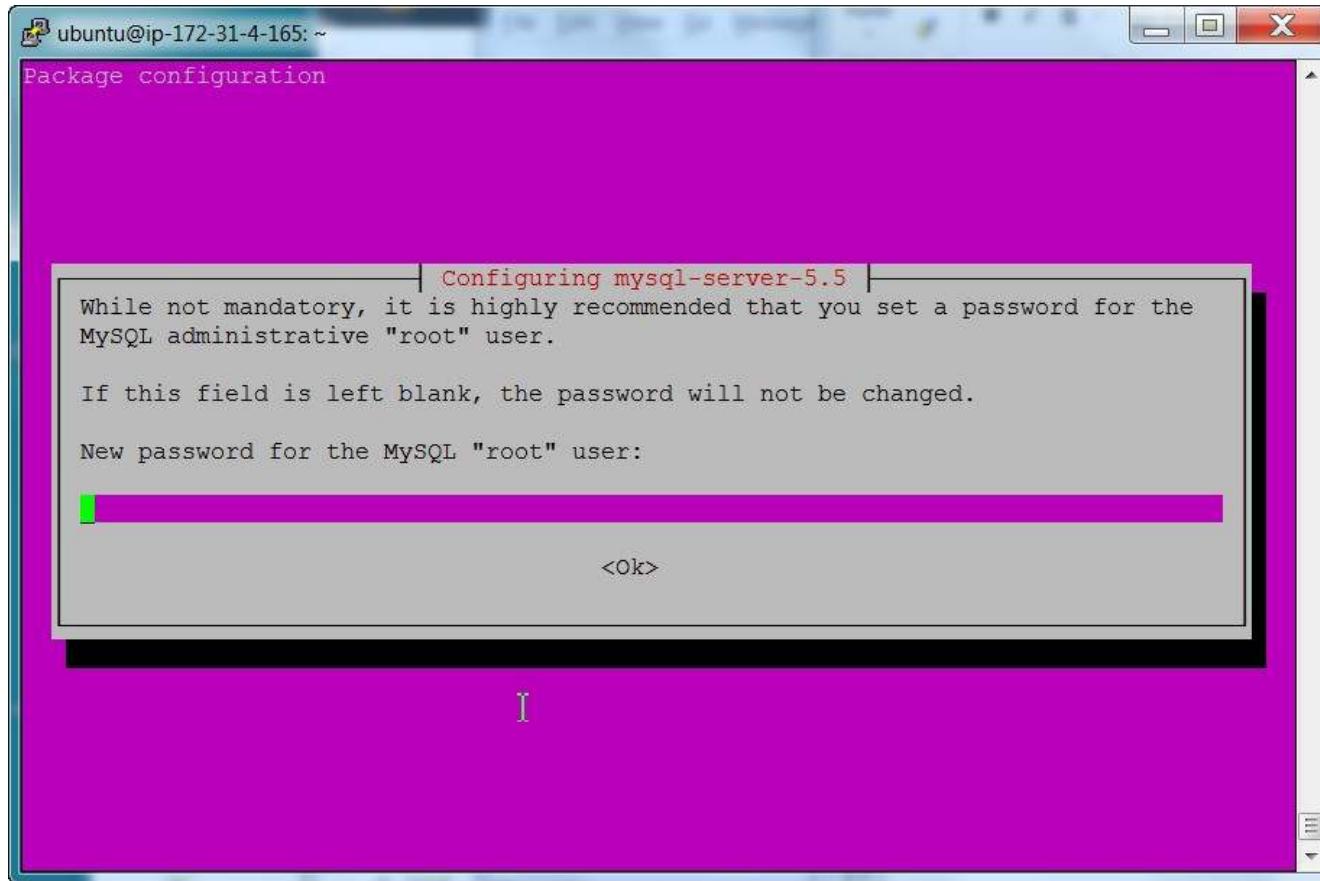
```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ cd
ubuntu@ip-172-31-4-165:~$ vi country_data_insert.sql
```

```
ubuntu@ip-172-31-4-165: ~
INSERT INTO country_tbl (id, country) VALUES ('66','Thailand');
INSERT INTO country_tbl (id, country) VALUES ('670','Timor-Leste');
INSERT INTO country_tbl (id, country) VALUES ('228','Togo');
INSERT INTO country_tbl (id, country) VALUES ('690','Tokelau');
INSERT INTO country_tbl (id, country) VALUES ('676','Tonga');
INSERT INTO country_tbl (id, country) VALUES ('1868','TrinidadandTobago');
INSERT INTO country_tbl (id, country) VALUES ('216','Tunisia');
INSERT INTO country_tbl (id, country) VALUES ('90','Turkey');
INSERT INTO country_tbl (id, country) VALUES ('993','Turkmenistan');
INSERT INTO country_tbl (id, country) VALUES ('1649','TurksandCaicosIslands');
INSERT INTO country_tbl (id, country) VALUES ('688','Tuvalu');
INSERT INTO country_tbl (id, country) VALUES ('256','Uganda');
INSERT INTO country_tbl (id, country) VALUES ('380','Ukraine');
INSERT INTO country_tbl (id, country) VALUES ('971','UnitedArabEmirates');
INSERT INTO country_tbl (id, country) VALUES ('44','UnitedKingdom');
INSERT INTO country_tbl (id, country) VALUES ('1','UnitedStates');
INSERT INTO country_tbl (id, country) VALUES ('598','Uruguay');
INSERT INTO country_tbl (id, country) VALUES ('1340','USVirginIslands');
INSERT INTO country_tbl (id, country) VALUES ('998','Uzbekistan');
INSERT INTO country_tbl (id, country) VALUES ('678','Vanuatu');
INSERT INTO country_tbl (id, country) VALUES ('58','Venezuela');
INSERT INTO country_tbl (id, country) VALUES ('84','Vietnam');
INSERT INTO country_tbl (id, country) VALUES ('681','WallisandFutuna');
INSERT INTO country_tbl (id, country) VALUES ('970','WestBank');
INSERT INTO country_tbl (id, country) VALUES ('967','Yemen');
INSERT INTO country_tbl (id, country) VALUES ('260','Zambia');
INSERT INTO country_tbl (id, country) VALUES ('263','Zimbabwe');
:wq!
```

Install MySQL

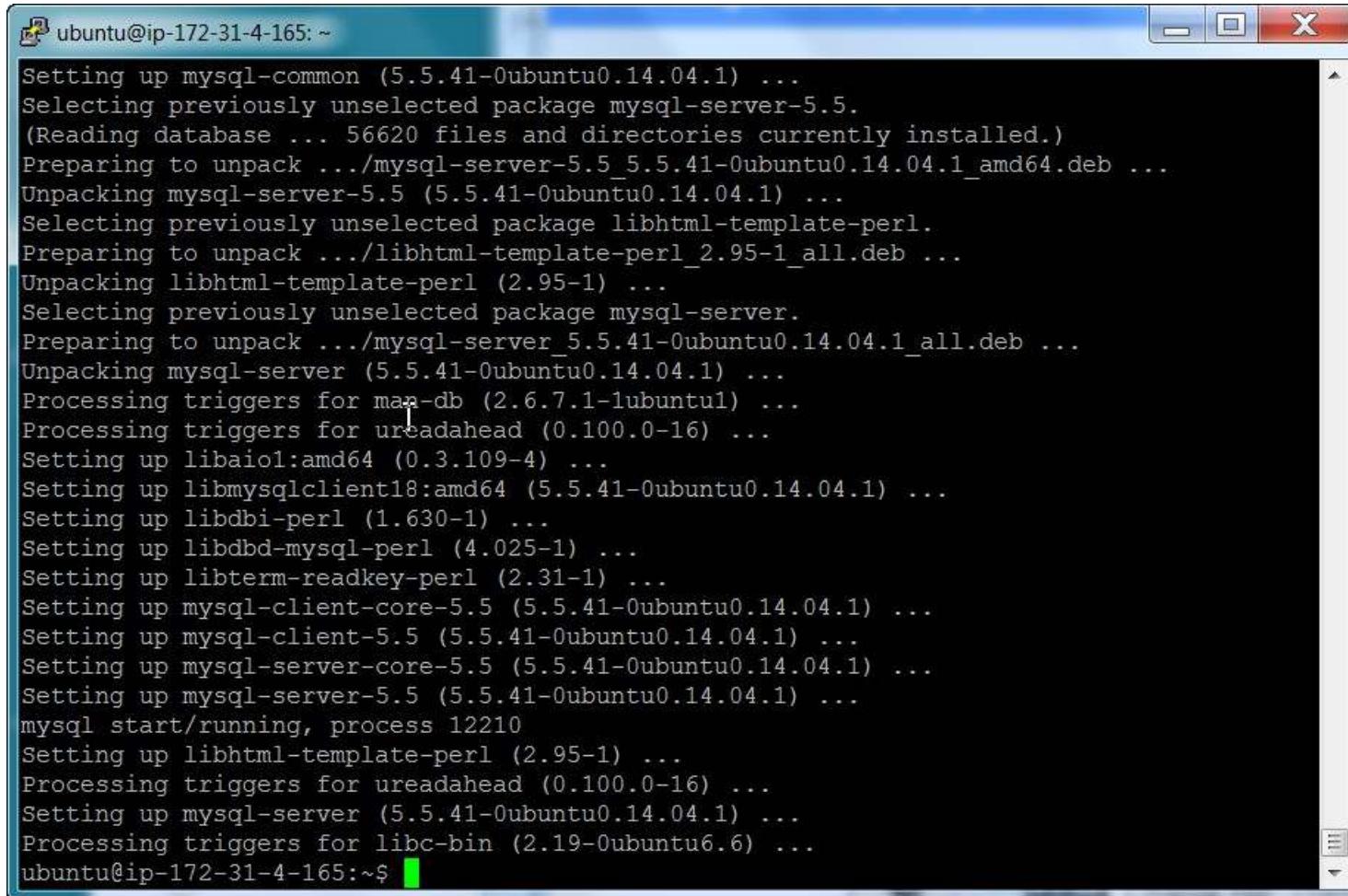
```
sudo apt-get install mysql-server
```

Leave blank for password in our class room testing



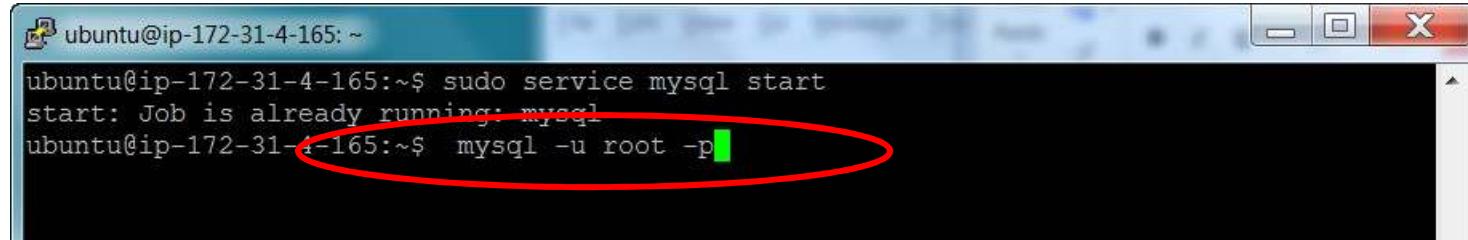
Install MySQL

Done installation

A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains a log of MySQL package installation commands and their progress. The text is as follows:

```
setting up mysql-common (5.5.41-0ubuntu0.14.04.1) ...
Selecting previously unselected package mysql-server-5.5.
(Reading database ... 56620 files and directories currently installed.)
Preparing to unpack .../mysql-server-5.5_5.5.41-0ubuntu0.14.04.1_amd64.deb ...
Unpacking mysql-server-5.5 (5.5.41-0ubuntu0.14.04.1) ...
Selecting previously unselected package libhtml-template-perl.
Preparing to unpack .../libhtml-template-perl_2.95-1_all.deb ...
Unpacking libhtml-template-perl (2.95-1) ...
Selecting previously unselected package mysql-server.
Preparing to unpack .../mysql-server_5.5.41-0ubuntu0.14.04.1_all.deb ...
Unpacking mysql-server (5.5.41-0ubuntu0.14.04.1) ...
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Setting up libaiol:amd64 (0.3.109-4) ...
Setting up libmysqlclient18:amd64 (5.5.41-0ubuntu0.14.04.1) ...
Setting up libdbi-perl (1.630-1) ...
Setting up libdbd-mysql-perl (4.025-1) ...
Setting up libterm-readkey-perl (2.31-1) ...
Setting up mysql-client-core-5.5 (5.5.41-0ubuntu0.14.04.1) ...
Setting up mysql-client-5.5 (5.5.41-0ubuntu0.14.04.1) ...
Setting up mysql-server-core-5.5 (5.5.41-0ubuntu0.14.04.1) ...
Setting up mysql-server-5.5 (5.5.41-0ubuntu0.14.04.1) ...
mysql start/running, process 12210
Setting up libhtml-template-perl (2.95-1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Setting up mysql-server (5.5.41-0ubuntu0.14.04.1) ...
Processing triggers for libc-bin (2.19-0ubuntu6.6) ...
ubuntu@ip-172-31-4-165:~$
```

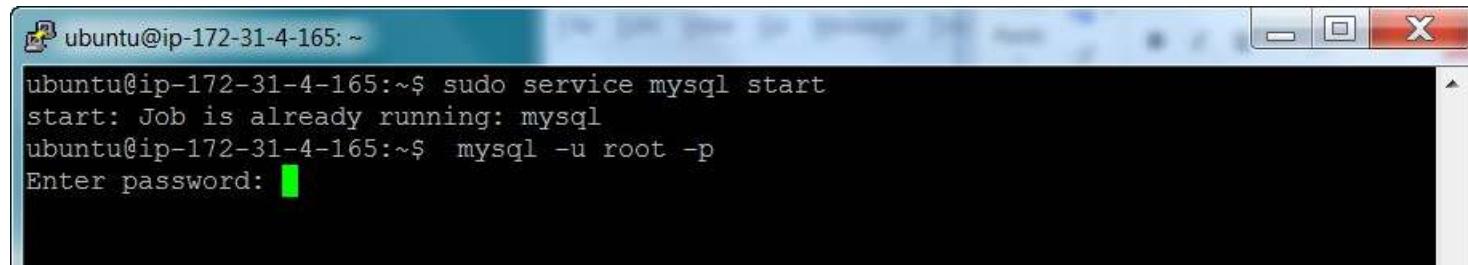
Start MySQL service



```
ubuntu@ip-172-31-4-165:~$ sudo service mysql start
start: Job is already running: mysql
ubuntu@ip-172-31-4-165:~$ mysql -u root -p
```

A terminal window titled "ubuntu@ip-172-31-4-165: ~". It shows the command "sudo service mysql start" followed by the output "start: Job is already running: mysql". Below that is the command "mysql -u root -p". A red oval highlights the "-p" option.

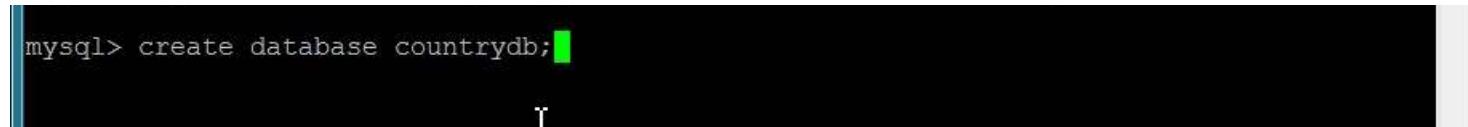
Press enter for password



```
ubuntu@ip-172-31-4-165:~$ sudo service mysql start
start: Job is already running: mysql
ubuntu@ip-172-31-4-165:~$ mysql -u root -p
Enter password:
```

A terminal window titled "ubuntu@ip-172-31-4-165: ~". It shows the same commands as the previous window, but the password prompt "Enter password:" has a green cursor block over it.

Create database name: countrydb



```
mysql> create database countrydb;
```

A terminal window titled "mysql>". It shows the command "create database countrydb;".

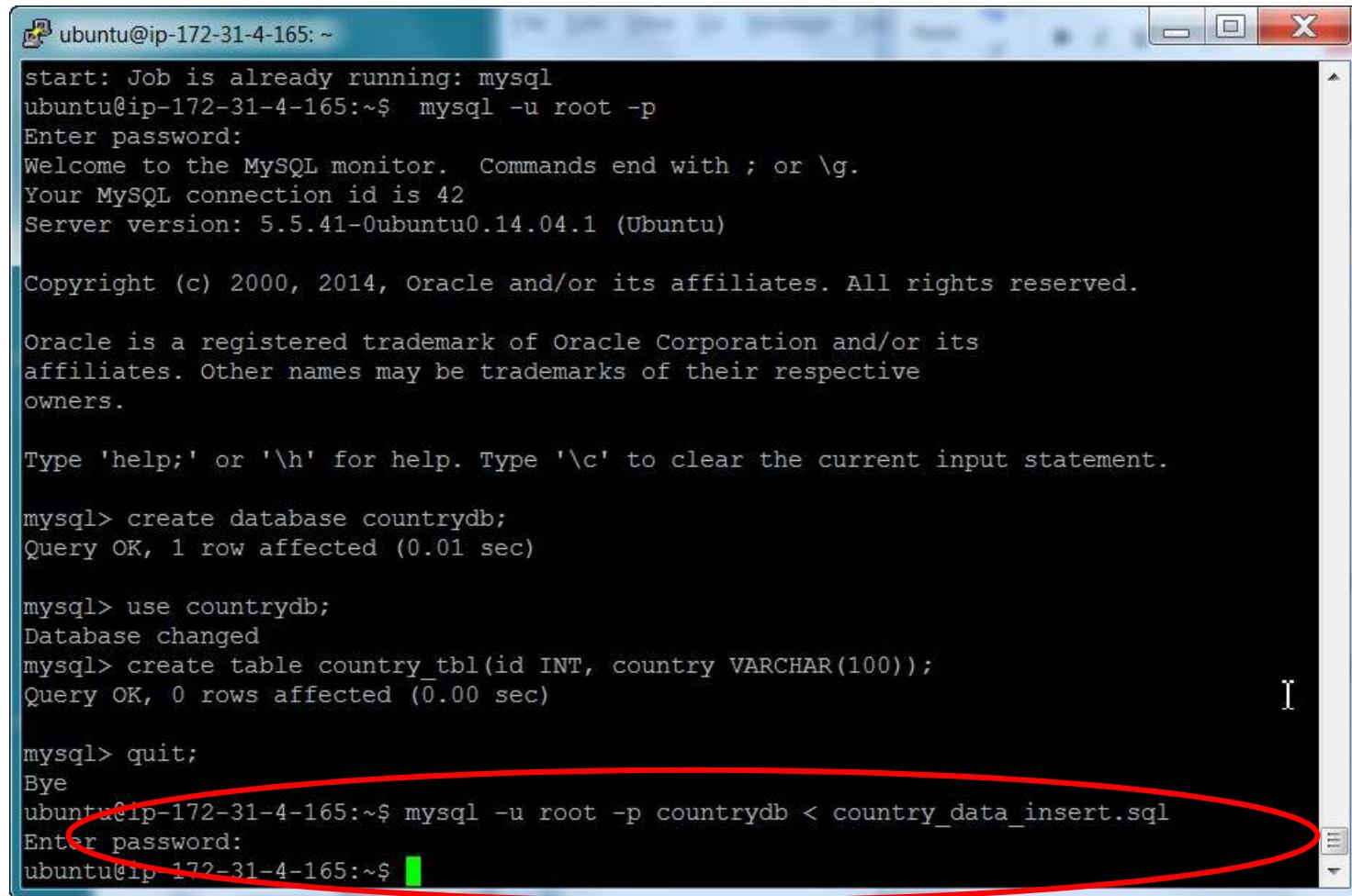
Select database name: countrydb

```
mysql> use countrydb;  
Database changed  
mysql> █
```

Create table name: country_tbl

```
mysql> create table country_tbl(id INT, country VARCHAR(100)); █
```

Import your sample data to MySQL



```
ubuntu@ip-172-31-4-165:~$ start: Job is already running: mysql
ubuntu@ip-172-31-4-165:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 42
Server version: 5.5.41-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

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

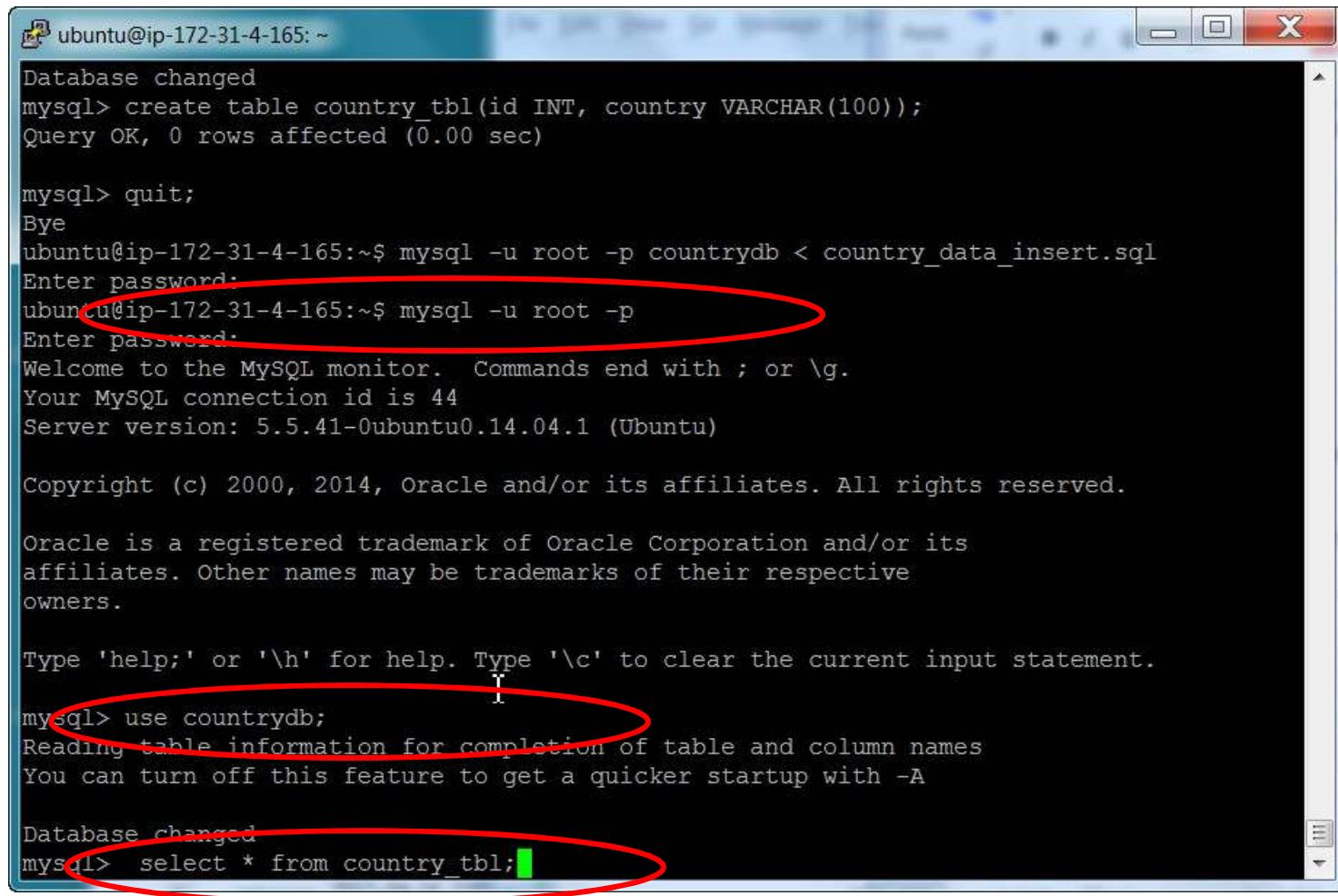
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database countrydb;
Query OK, 1 row affected (0.01 sec)

mysql> use countrydb;
Database changed
mysql> create table country_tbl(id INT, country VARCHAR(100));
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
ubuntu@ip-172-31-4-165:~$ mysql -u root -p countrydb < country_data_insert.sql
Enter password:
ubuntu@ip-172-31-4-165:~$
```

Review your sample data from MySQL



```
ubuntu@ip-172-31-4-165: ~
Database changed
mysql> create table country_tbl(id INT, country VARCHAR(100));
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
ubuntu@ip-172-31-4-165:~$ mysql -u root -p countrydb < country_data_insert.sql
Enter password:
ubuntu@ip-172-31-4-165:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.41-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

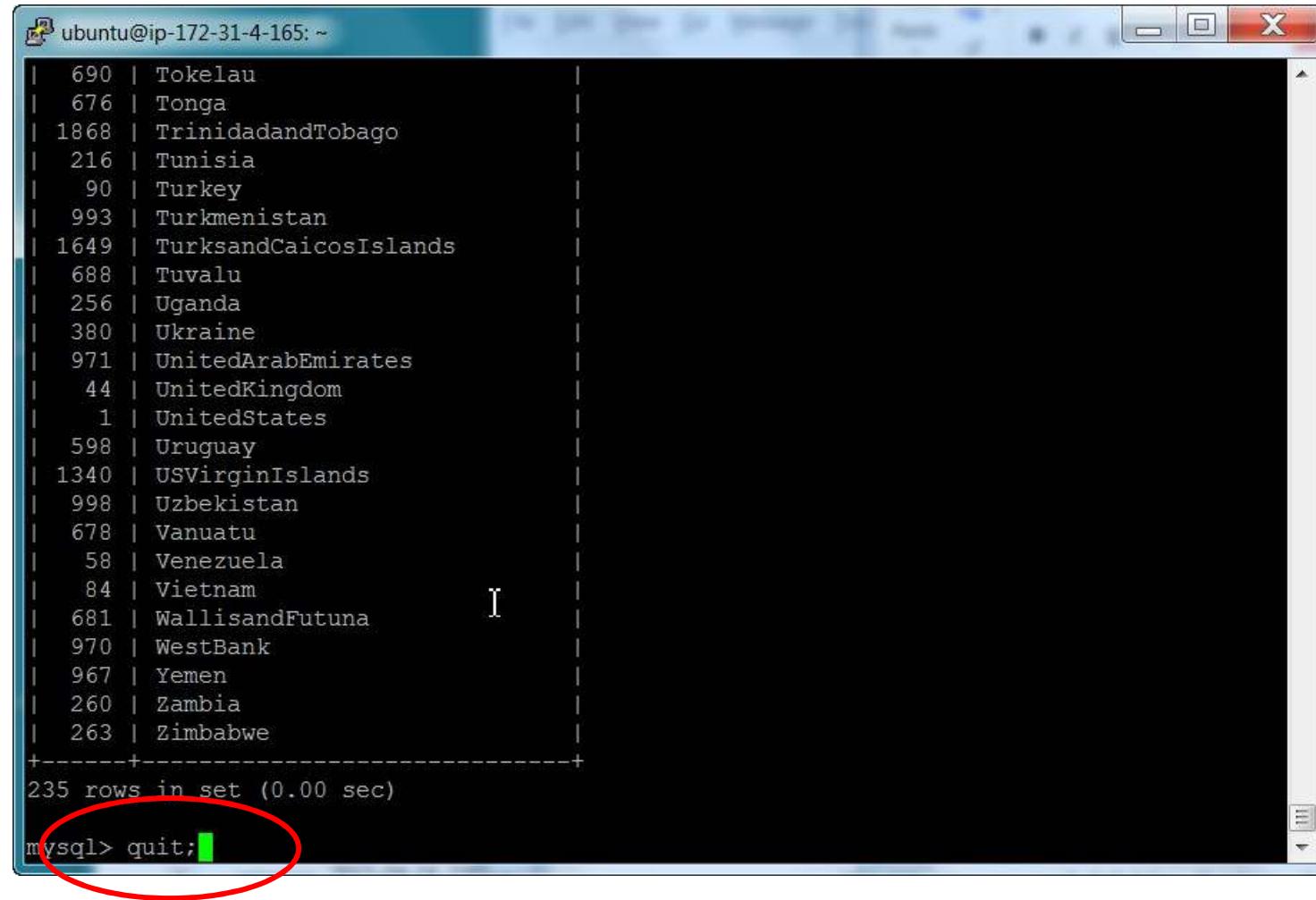
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> use countrydb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from country_tbl;
```

Review your sample data from MySQL



A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window displays the results of a MySQL query. The output shows a list of countries and their corresponding IDs. A red oval highlights the command "mysql> quit;" at the bottom of the screen.

690	Tokelau
676	Tonga
1868	TrinidadandTobago
216	Tunisia
90	Turkey
993	Turkmenistan
1649	TurksandCaicosIslands
688	Tuvalu
256	Uganda
380	Ukraine
971	UnitedArabEmirates
44	UnitedKingdom
1	UnitedStates
598	Uruguay
1340	USVirginIslands
998	Uzbekistan
678	Vanuatu
58	Venezuela
84	Vietnam
681	WallisandFutuna
970	WestBank
967	Yemen
260	Zambia
263	Zimbabwe

235 rows in set (0.00 sec)

mysql> quit;

2. Installing Sqoop

Install Sqoop

```
wget http://mirror.cc.columbia.edu/pub/software/apache/sqoop/1.4.5/sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz
```

The screenshot shows a terminal window with a red border. Inside, the user is performing two commands:

```
ubuntu@ip-172-31-4-165:~$ wget http://mirror.cc.columbia.edu/pub/software/apache/sqoop/1.4.5/sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz
--2015-04-16 16:02:01--  http://mirror.cc.columbia.edu/pub/software/apache/sqoop/1.4.5/sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz
Resolving mirror.cc.columbia.edu (mirror.cc.columbia.edu)... 128.59.59.71
Connecting to mirror.cc.columbia.edu (mirror.cc.columbia.edu)|128.59.59.71|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6459385 (6.2M) [application/x-gzip]
Saving to: 'sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz'

100%[=====] 6,459,385  4.52MB/s   in 1.4s

2015-04-16 16:02:03 (4.52 MB/s) - 'sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz' saved [6459385/6459385]

ubuntu@ip-172-31-4-165:~$ tar -xvf sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz
```

The terminal window has a red border and three red circles highlighting specific parts of the command and output. The first circle highlights the URL in the wget command. The second circle highlights the download progress bar and file size. The third circle highlights the tar command at the bottom.

Edit System Environment Variables

The image shows two terminal windows on an Ubuntu system. The top window shows the command `mv sqoop-1.4.5.bin_hadoop-2.0.4-alpha sqoop` being run, followed by `vi /home/ubuntu/.bashrc`. A red oval highlights this second command. The bottom window shows the contents of `/etc/bash.bashrc` with several environment variable assignments. A red oval highlights the last two assignments at the bottom:

```
# sourced /etc/bash.bashrc.
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin

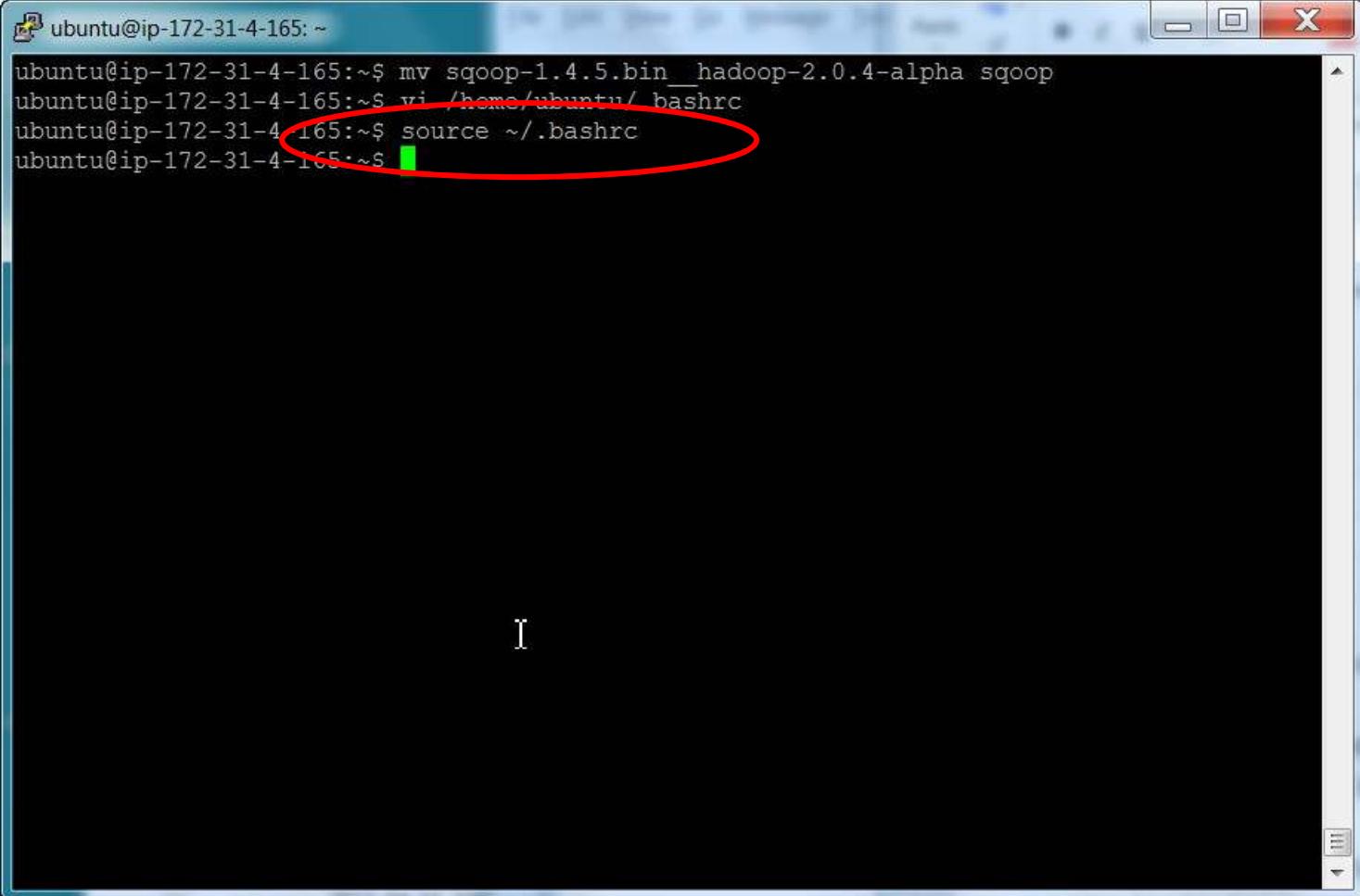
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin

export HIVE_HOME=/usr/local/apache-hive
export PATH=$PATH:$HIVE_HOME/bin

export SQOOP_HOME=/usr/lib/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
```

-- INSERT --

Execute System Environment Variables



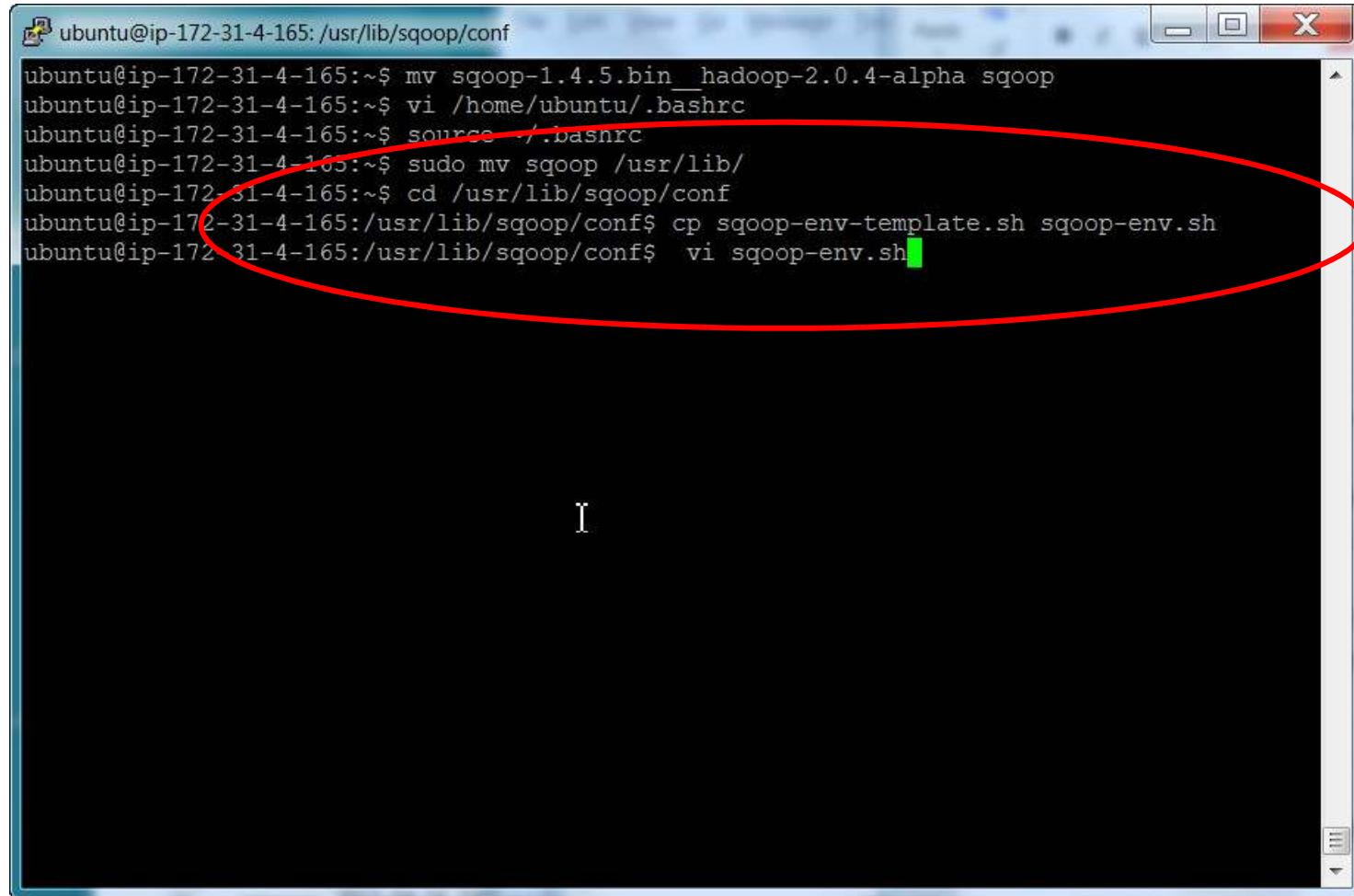
A screenshot of a terminal window titled "ubuntu@ip-172-31-4-165: ~". The window contains the following commands:

```
ubuntu@ip-172-31-4-165:~$ mv sqoop-1.4.5-bin_hadoop-2.0.4-alpha sqoop
ubuntu@ip-172-31-4-165:~$ vi /home/ubuntu/.bashrc
ubuntu@ip-172-31-4-165:~$ source ~/.bashrc
ubuntu@ip-172-31-4-165:~$
```

The command "source ~/.bashrc" is highlighted with a red oval.

3. Configuring Sqoop

Configuring Sqoop



```
ubuntu@ip-172-31-4-165:~$ mv sqoop-1.4.5-bin_hadoop-2.0.4-alpha sqoop
ubuntu@ip-172-31-4-165:~$ vi /home/ubuntu/.bashrc
ubuntu@ip-172-31-4-165:~$ source ./bashrc
ubuntu@ip-172-31-4-165:~$ sudo mv sqoop /usr/lib/
ubuntu@ip-172-31-4-165:~$ cd /usr/lib/sqoop/conf
ubuntu@ip-172-31-4-165:/usr/lib/sqoop/conf$ cp sqoop-env-template.sh sqoop-env.sh
ubuntu@ip-172-31-4-165:/usr/lib/sqoop/conf$ vi sqoop-env.sh
```

Configuring Sqoop



```
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# included in all the hadoop scripts with source command
# should not be executable directly
# also should not be passed any arguments, since we need original $*
# Set hadoop specific environment variables here.

#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/usr/local/hadoop

#Set path to where hadoop-*core.jar is available
export HADOOP_MAPRED_HOME=/usr/local/hadoop

#set the path to where bin/hbase is available
#export HBASE_HOME=

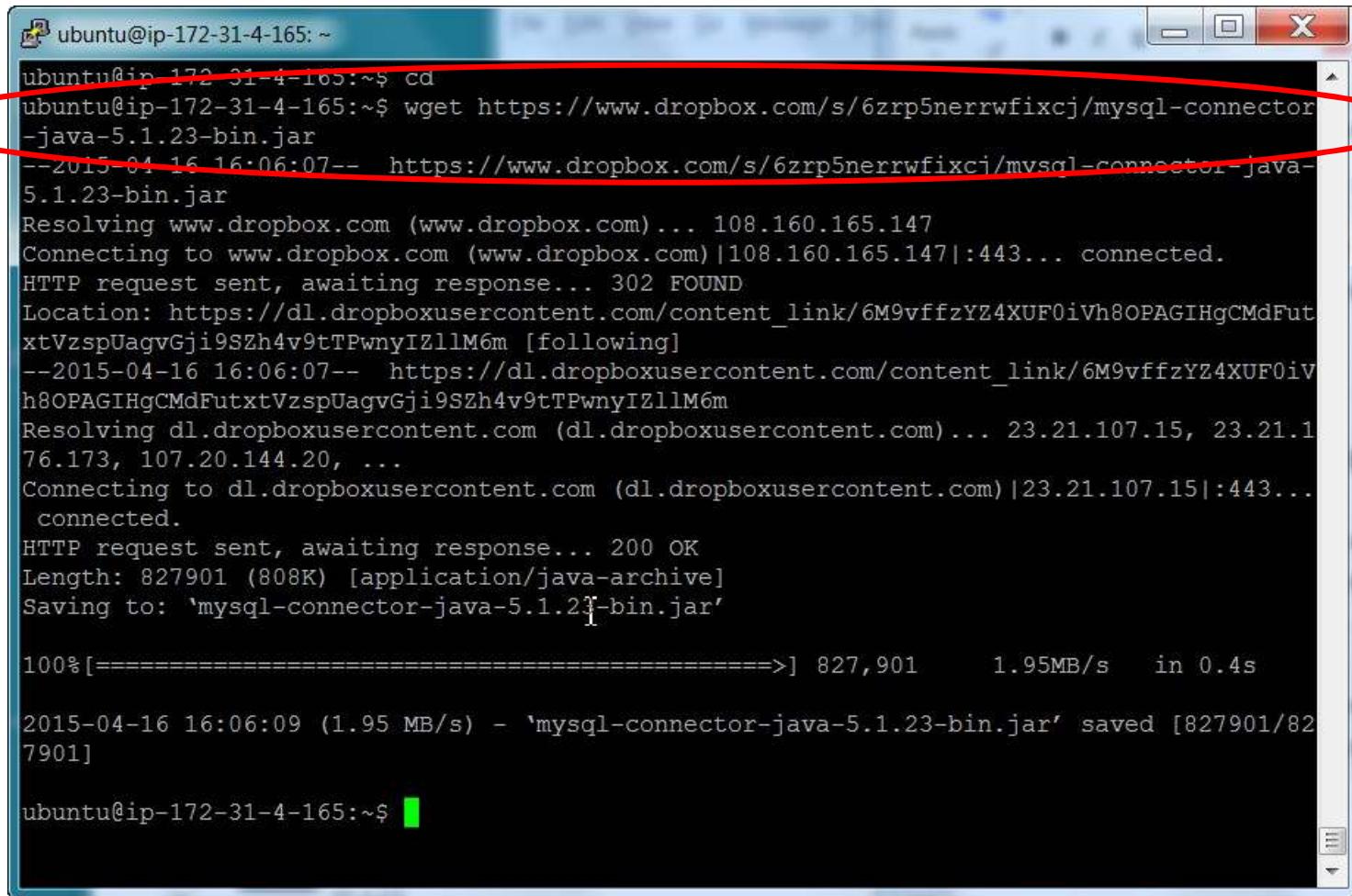
#Set the path to where bin/hive is available
export HIVE_HOME=/usr/local/apache-hive

#Set the path for where zookper config dir is
#export ZOOKEEPER_HOME=
```

4. Installing DB driver for Sqoop

Install MySQL DB Driver

```
wget https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-java-5.1.23-bin.jar
```



The screenshot shows a terminal window titled "ubuntu@ip-172-31-4-165:~". The user has run the command "wget https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-java-5.1.23-bin.jar". A red oval highlights the URL in the command line. The terminal output shows the progress of the download, including connection details, HTTP requests, and the final save location "mysql-connector-java-5.1.23-bin.jar". The download completes at 1.95MB/s in 0.4s.

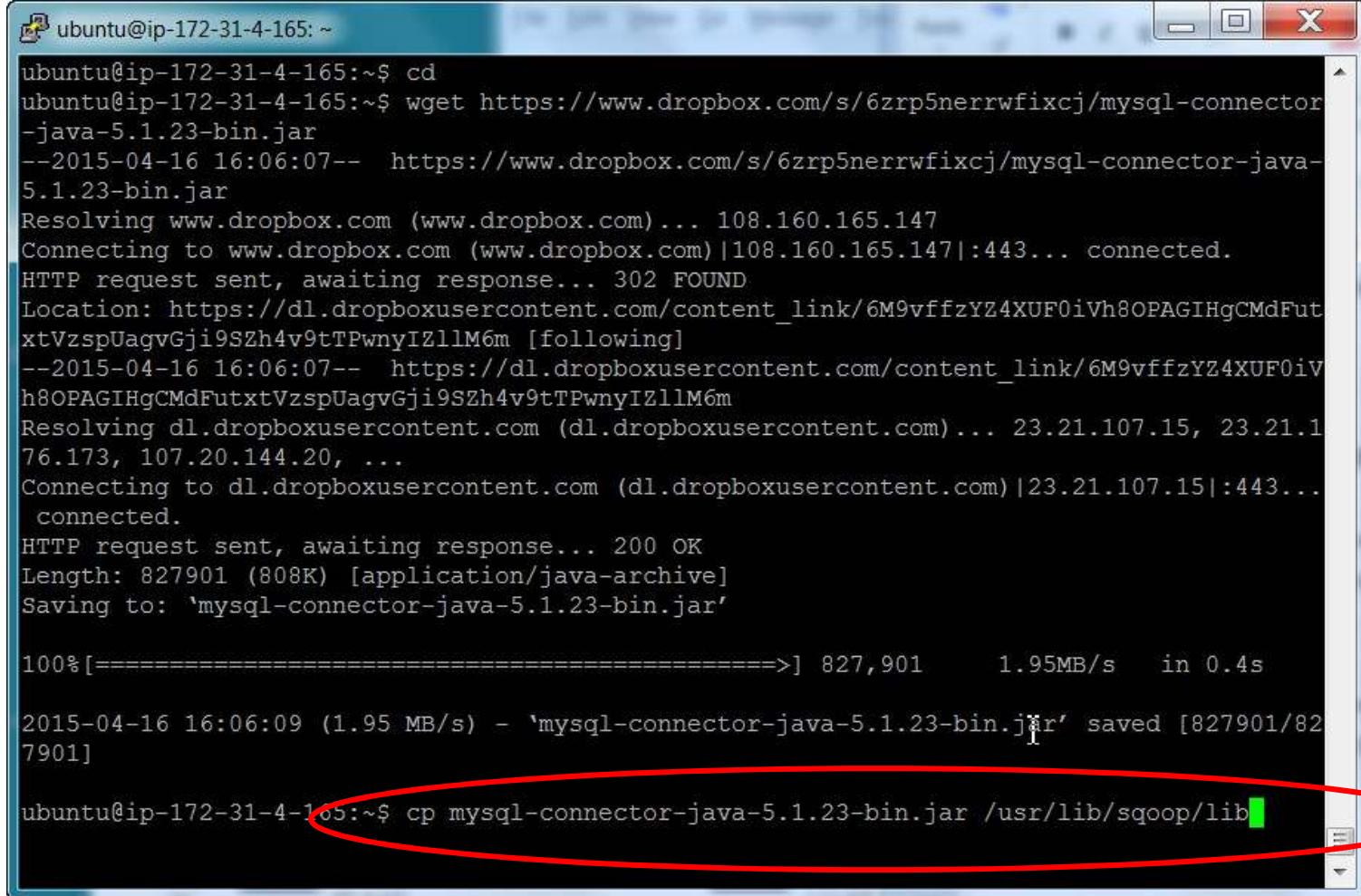
```
ubuntu@ip-172-31-4-165:~$ cd
ubuntu@ip-172-31-4-165:~$ wget https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-
-ja
--2015-04-16 16:06:07-- https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-ja-
5.1.23-bin.jar
Resolving www.dropbox.com (www.dropbox.com) ... 108.160.165.147
Connecting to www.dropbox.com (www.dropbox.com)|108.160.165.147|:443... connected.
HTTP request sent, awaiting response... 302 FOUND
Location: https://dl.dropboxusercontent.com/content_link/6M9vffzYZ4XUF0iVh8OPAGIHgCMdFut
xtVzspUagvGji9Sz
h4v9tTPwnyIZllM6m [following]
--2015-04-16 16:06:07-- https://dl.dropboxusercontent.com/content_link/6M9vffzYZ4XUF0iV
h8OPAGIHgCMdFutxtVzspUagvGji9Sz
h4v9tTPwnyIZllM6m
Resolving dl.dropboxusercontent.com (dl.dropboxusercontent.com) ... 23.21.107.15, 23.21.1
76.173, 107.20.144.20, ...
Connecting to dl.dropboxusercontent.com (dl.dropboxusercontent.com)|23.21.107.15|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 827901 (808K) [application/java-archive]
Saving to: 'mysql-connector-java-5.1.23-bin.jar'

100%[=====] 827,901      1.95MB/s   in 0.4s

2015-04-16 16:06:09 (1.95 MB/s) - 'mysql-connector-java-5.1.23-bin.jar' saved [827901/82
7901]

ubuntu@ip-172-31-4-165:~$
```

Install MySQL DB Driver



```
ubuntu@ip-172-31-4-165:~$ cd
ubuntu@ip-172-31-4-165:~$ wget https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-
-java-5.1.23-bin.jar
--2015-04-16 16:06:07-- https://www.dropbox.com/s/6zrp5nerrwfixcj/mysql-connector-java-
5.1.23-bin.jar
Resolving www.dropbox.com (www.dropbox.com)... 108.160.165.147
Connecting to www.dropbox.com (www.dropbox.com)|108.160.165.147|:443... connected.
HTTP request sent, awaiting response... 302 FOUND
Location: https://dl.dropboxusercontent.com/content_link/6M9vffzYZ4XUF0ivh8OPAGIHgCMdFut
xtVzspUagvGji9SZh4v9tTPwnyIZllM6m [following]
--2015-04-16 16:06:07-- https://dl.dropboxusercontent.com/content_link/6M9vffzYZ4XUF0iv
h8OPAGIHgCMdFutxtVzspUagvGji9SZh4v9tTPwnyIZllM6m
Resolving dl.dropboxusercontent.com (dl.dropboxusercontent.com)... 23.21.107.15, 23.21.1
76.173, 107.20.144.20, ...
Connecting to dl.dropboxusercontent.com (dl.dropboxusercontent.com)|23.21.107.15|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 827901 (808K) [application/java-archive]
Saving to: 'mysql-connector-java-5.1.23-bin.jar'

100%[=====] 827,901      1.95MB/s   in 0.4s

2015-04-16 16:06:09 (1.95 MB/s) - 'mysql-connector-java-5.1.23-bin.jar' saved [827901/82
7901]

ubuntu@ip-172-31-4-165:~$ cp mysql-connector-java-5.1.23-bin.jar /usr/lib/sqoop/lib
```

5. Importing data from MySQL to Hadoop

Sqoop Import Command

```
sqoop import -connect jdbc:mysql://localhost/countrydb -username root -table country_tbl -m 1
```

The screenshot shows two terminal windows on an Ubuntu system. The top window displays the command being run:

```
ubuntu@ip-172-31-4-165:~$ sqoop import -connect jdbc:mysql://localhost/countrydb -username root -table country_tbl -m 1
```

A red circle highlights this window. The bottom window shows the job counters and other metrics for the completed import:

```
Job Counters
    Launched map tasks=1
    Other local map tasks=1
    Total time spent by all maps in occupied slots (ms)=11757
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=11757
    Total vcore-seconds taken by all map tasks=11757
    Total megabyte-seconds taken by all map tasks=12039168

Map-Reduce Framework
    Map input records=235
    Map output records=235
    Input split bytes=87
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=142
    CPU time spent (ms)=1840
    Physical memory (bytes) snapshot=116699136
    Virtual memory (bytes) snapshot=792125440
    Total committed heap usage (bytes)=59572224

File Input Format Counters
    Bytes Read=0

File Output Format Counters
    Bytes Written=3252

15/04/16 16:08:05 INFO mapreduce.ImportJobBase: Transferred 3.1758 KB in 37.8213 seconds
(85.9833 bytes/sec)
15/04/16 16:08:05 INFO mapreduce.ImportJobBase: Retrieved 235 records.
```

A red circle highlights the bottom window.

6. Reviewing HDFS data

Review HDFS result data

The image shows two terminal windows side-by-side. The top window has a red oval highlighting the command line. The bottom window displays the output of the command.

```
ubuntu@ip-172-31-4-165: ~
ubuntu@ip-172-31-4-165:~$ hdfs dfs -cat /user/ubuntu/country_tbl/part-m-00000
```

```
66,Thailand
670,Timor-Leste
228,Togo
690,Tokelau
676,Tonga
1868,TrinidadandTobago
216,Tunisia
90,Turkey
993,Turkmenistan
1649,TurksandCaicosIslands
688,Tuvalu
256,Uganda
380,Ukraine
971,UnitedArabEmirates
44,UnitedKingdom
1,UnitedStates
598,Uruguay
1340,USVirginIslands
998,Uzbekistan
678,Vanuatu
58,Venezuela
84,Vietnam
681,WallisandFutuna
970,WestBank
967,Yemen
260,Zambia
263,Zimbabwe
```

```
ubuntu@ip-172-31-4-165:~$
```

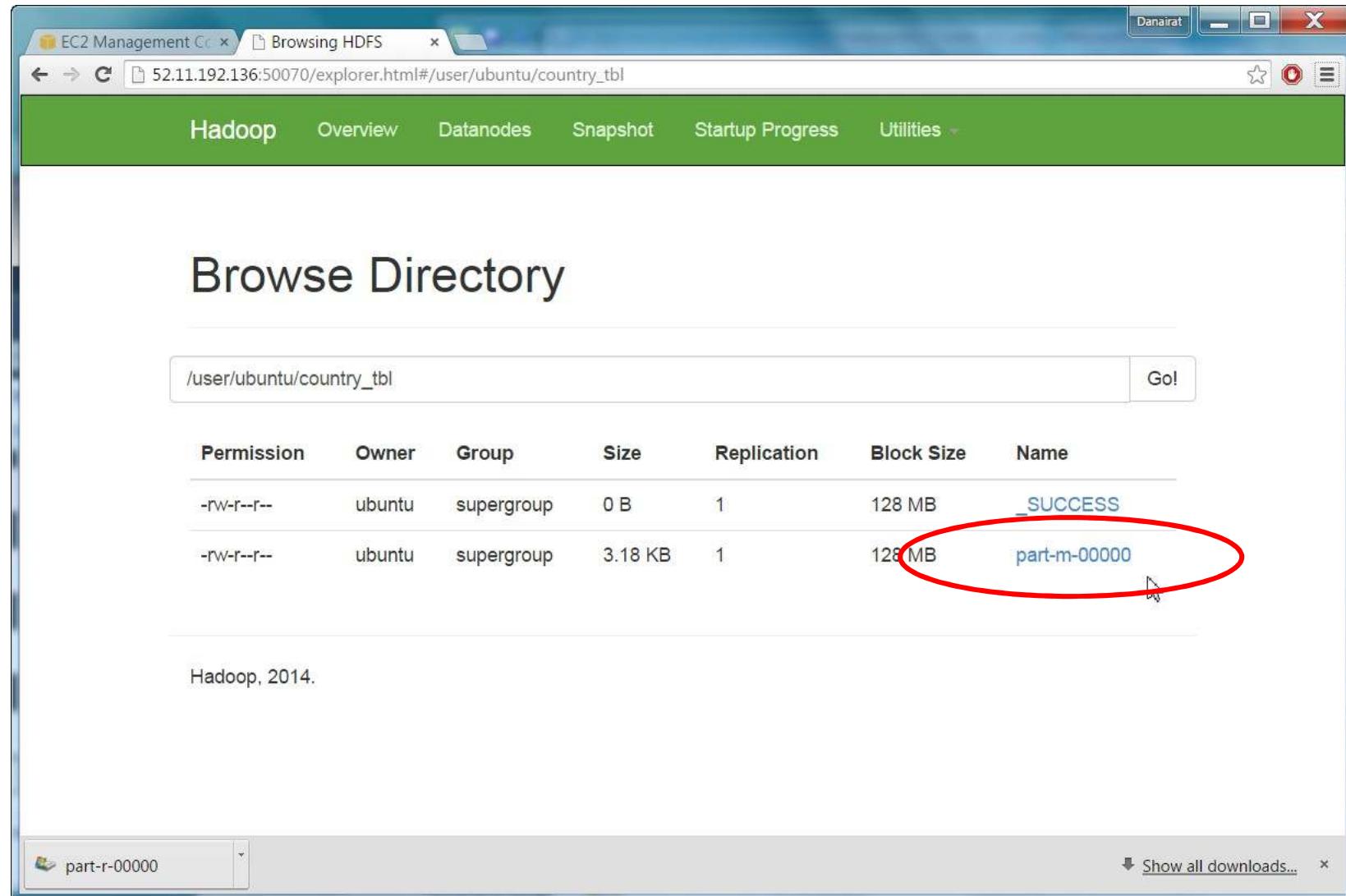
Review HDFS result data

The screenshot shows a web browser window with the URL 52.11.192.136:50070/dfshealth.html#tab-overview. The browser title bar includes 'EC2 Management C...' and 'Namenode information'. The main content area displays the 'Overview' tab of the HDFS health status. A red oval highlights the 'Utilities' dropdown menu, which contains 'Browse the file system' and 'Logs'. Below the overview, a table provides cluster metadata:

Started:	Thu Apr 16 13:33:40 UTC 2015
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-1552225a-c860-42a1-9096-9b1635d12343
Block Pool ID:	BP-687806796-172.31.4.165-1429191194115

Below the table, a 'Summary' section states 'Security is off.' At the bottom, a download bar shows a file named 'part-r-00000'.

Review HDFS result data



Browsing HDFS

52.11.192.136:50070/explorer.html#/user/ubuntu/country_tbl

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

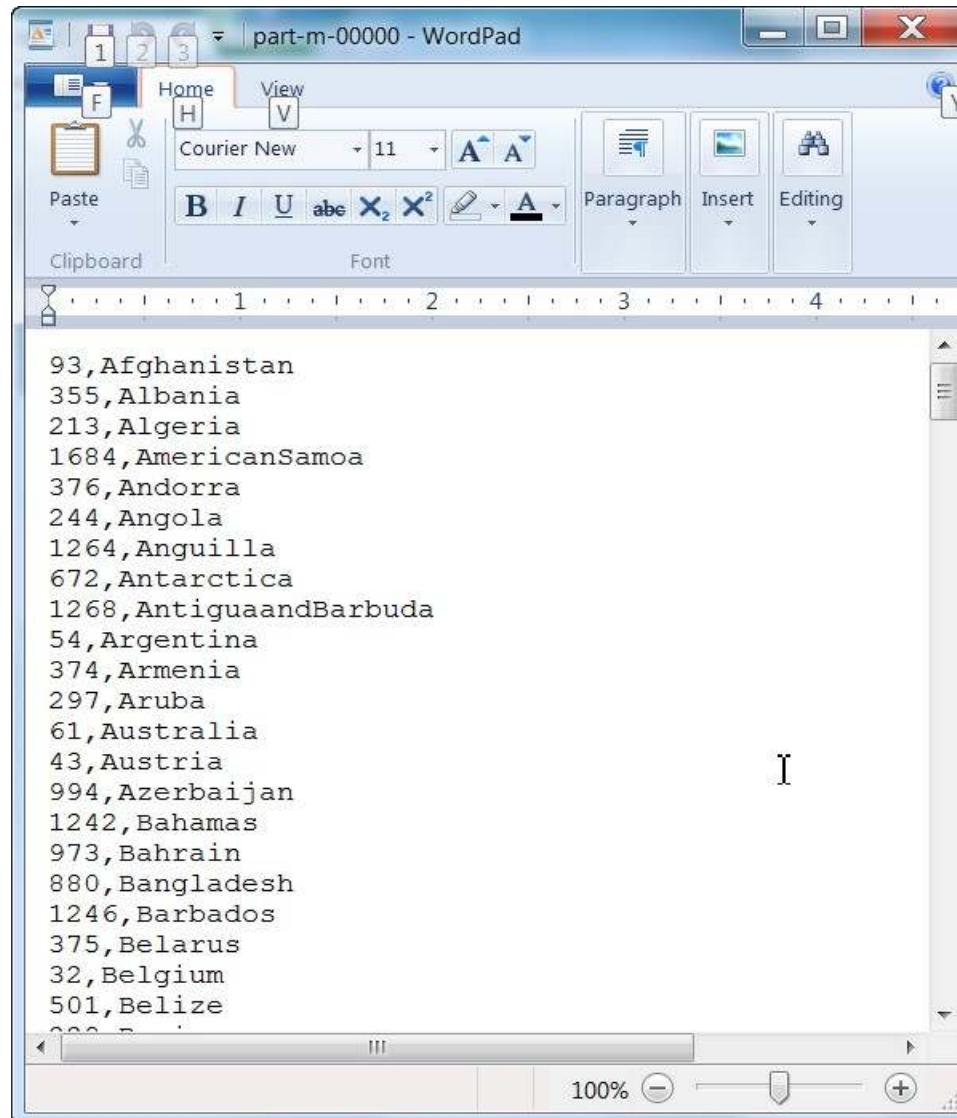
/user/ubuntu/country_tbl Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rw-r--r--	ubuntu	supergroup	0 B	1	128 MB	_SUCCESS
-rw-r--r--	ubuntu	supergroup	3.18 KB	1	128 MB	part-m-00000

Hadoop, 2014.

part-r-00000 Show all downloads...

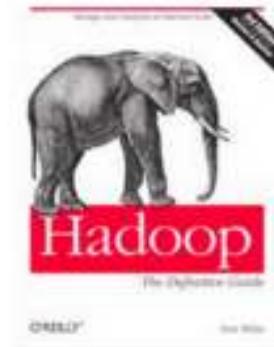
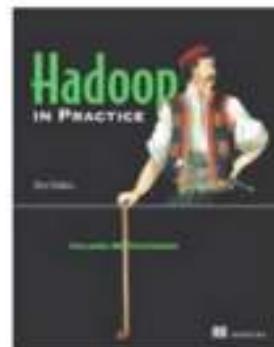
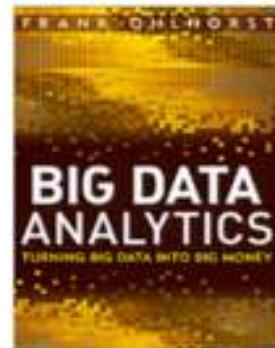
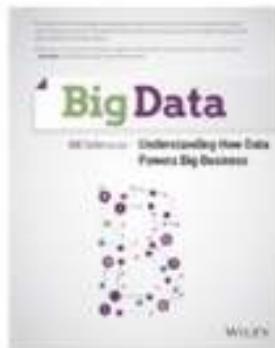
Review HDFS result data



Stop Hadoop Yarn and HDFS

```
ubuntu@ip-172-31-4-165: /var/hadoop_data
ubuntu@ip-172-31-4-165:/usr/local/hadoop/etc/hadoop$ cd /var/hadoop_data/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ ll
total 16
drwxr-xr-x  4 ubuntu ubuntu 4096 Apr 16 09:39 .
drwxr-xr-x 13 root   root   4096 Apr 16 09:39 ..
drwx----- 3 ubuntu ubuntu 4096 Apr 16 10:09 datanode/
drwxr-xr-x  3 ubuntu ubuntu 4096 Apr 16 10:09 namenode/
ubuntu@ip-172-31-4-165:/var/hadoop_data$ stop-yarn.sh
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
no proxyserver to stop
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
11224 SecondaryNameNode
11014 DataNode
10886 NameNode
12030 Jps
ubuntu@ip-172-31-4-165:/var/hadoop_data$ stop-dfs.sh
Stopping namenodes on [ip-172-31-4-165]
ip-172-31-4-165: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [0.0.0.0]
0.0.0.0: stopping secondarynamenode
ubuntu@ip-172-31-4-165:/var/hadoop_data$ jps
12469 Jps
ubuntu@ip-172-31-4-165:/var/hadoop_data$
```

Recommendation to Further Study



Thank you very much