

Big Data

Hadoop in Google Colab



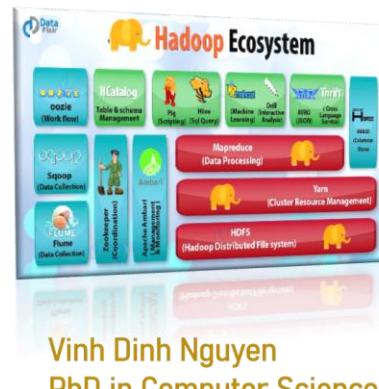
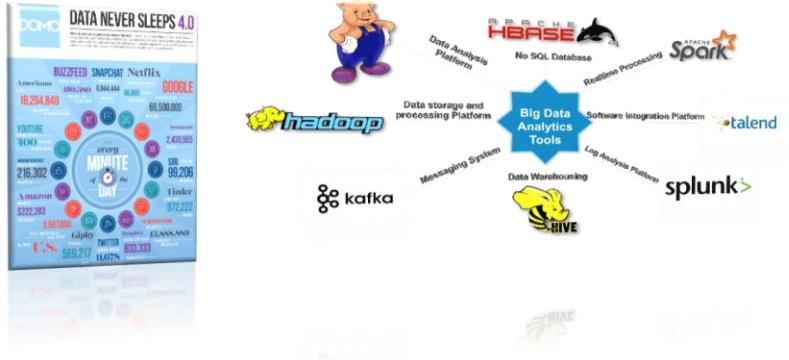
Vinh Dinh Nguyen
PhD in Computer Science

Big Data

From Zero To Mastery

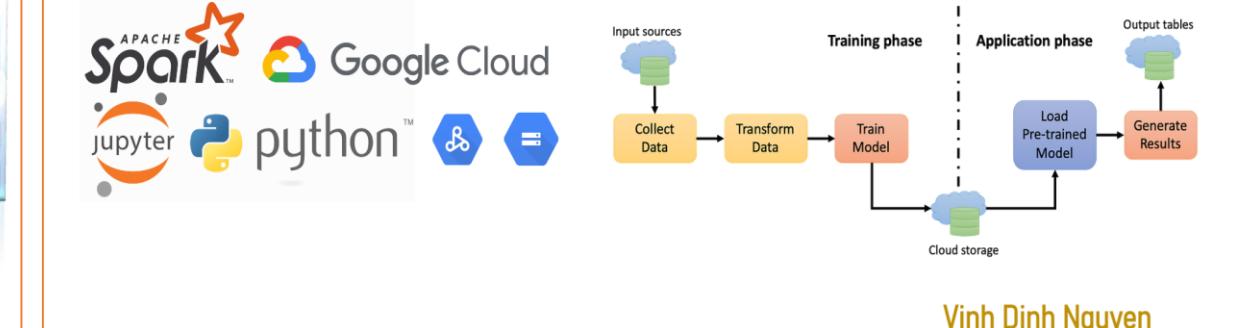
Big Data

(Zero To Mastery: Big Data Analytics Tools)

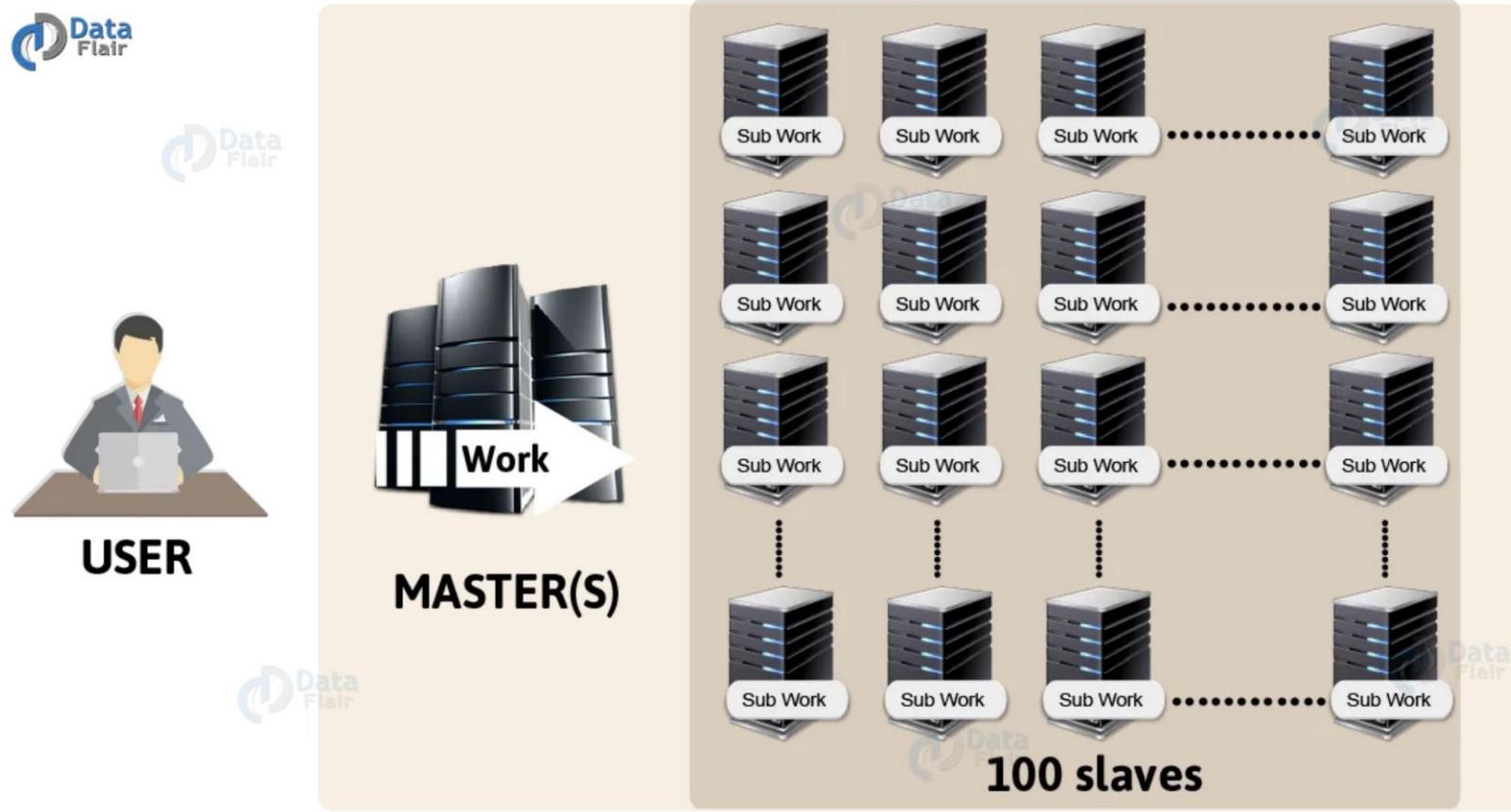


Big Data

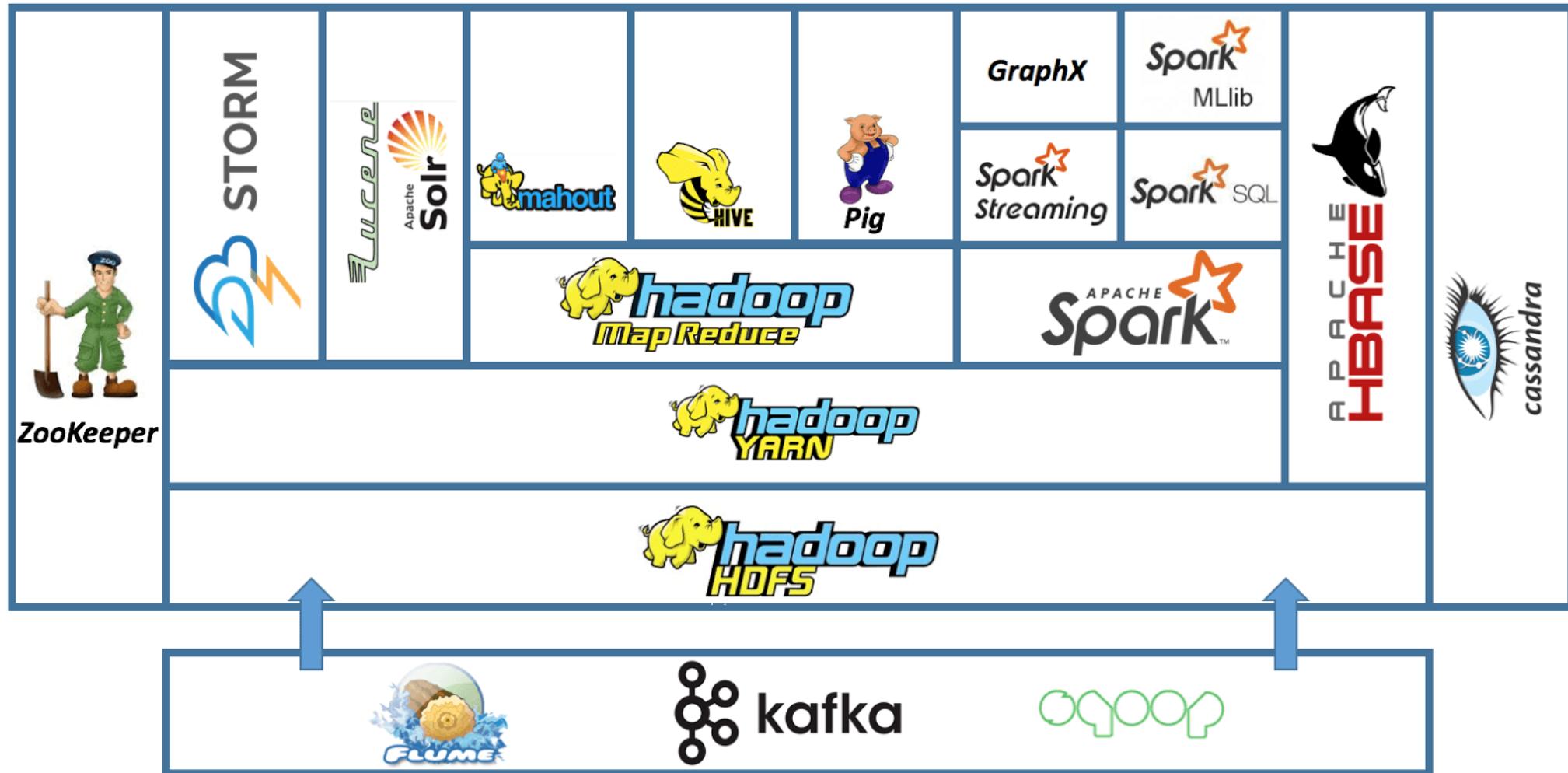
Models Using PySpark in Cloud Platform)



Hadoop Architecture Review



Hadoop Architecture Review



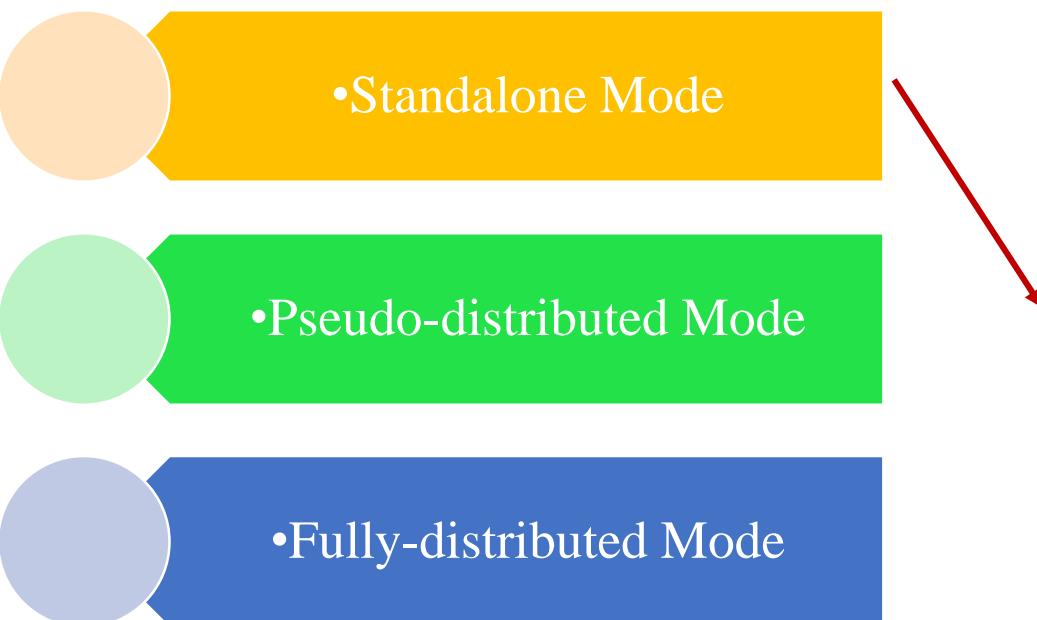
Hadoop Architecture Review

Hadoop is an open-source framework which is mainly used for storage purposes and maintaining and analyzing a large amount of data or datasets on the clusters of commodity hardware, which means it is actually a data management tool.

- Standalone Mode

- Pseudo-distributed Mode

- Fully-distributed Mode



This mode is useful for debugging and there isn't any need to configure core-site.xml, hdfs-site.xml, mapred-site.xml, masters & slaves. Stand-alone mode is usually the fastest mode in Hadoop.

Hadoop Architecture Review

Hadoop is an open-source framework which is mainly used for storage purposes and maintaining and analyzing a large amount of data or datasets on the clusters of commodity hardware, which means it is actually a data management tool.

- Standalone Mode

- Pseudo-distributed Mode

- Fully-distributed Mode

In this mode, each daemon runs on separate java processes. In this mode custom configuration is required (core-site.xml, hdfs-site.xml, mapred-site.xml). Here HDFS is utilized for input and output. This mode of deployment is useful for testing and debugging purposes.

Hadoop Architecture Review

Hadoop is an open-source framework which is mainly used for storage purposes and maintaining and analyzing a large amount of data or datasets on the clusters of commodity hardware, which means it is actually a data management tool.

- Standalone Mode

- Pseudo-distributed Mode

- Fully-distributed Mode

In this mode typically one machine in the cluster is designated as NameNode and another as Resource Manager exclusively. These are masters. All other nodes act as Data Node and Node Manager. These are the slaves. Configuration parameters and environment need to be specified for Hadoop Daemons.



Hello World in Hadoop using Hortonworks Sandbox

Hello World program in Hadoop using Hortonworks Sandbox



Step 1:
Load Data into HDFS

HDFS is a Hadoop component where data is stored



HCatalog

Step 2:
Register data with HCatalog

HCatalog provides data in tabular format



Step 3:
Process data using Hive

Hive is a SQL-like language to query the data

Step 1: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

Download and install Oracle Virtual Box

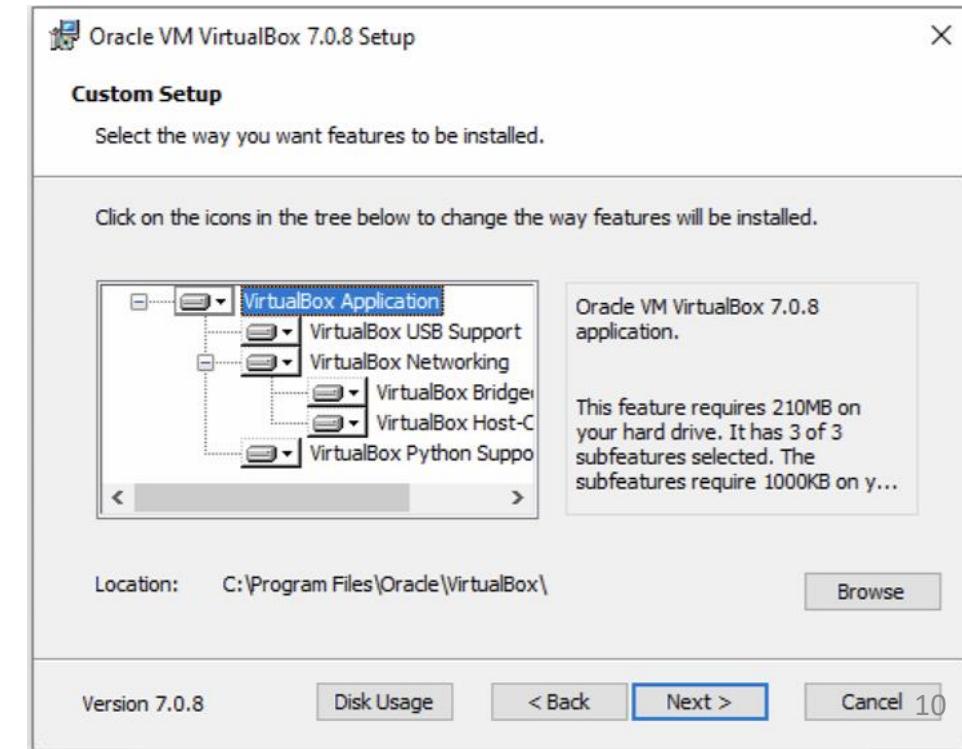


The screenshot shows a web browser displaying the VirtualBox download page at [virtualbox.org/wiki/Downloads](https://www.virtualbox.org/wiki/Downloads). The page features a large blue header with the 'VirtualBox' logo. Below the header, there's a navigation menu with links like 'About', 'Screenshots', 'Downloads', 'Documentation', 'End-user docs', 'Technical docs', 'Contribute', and 'Community'. The main content area is titled 'Download VirtualBox' and contains text about finding binaries and source code, a section for 'VirtualBox binaries' with a note about license terms, and a list of 'VirtualBox 7.0.8 platform packages' including Windows hosts, macOS / Intel hosts, developer preview for macOS / Arm64 (M1/M2) hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. At the bottom, it states that binaries are released under the terms of the GPL version 3. The URL <https://www.virtualbox.org/wiki/Downloads> is visible in the bottom right corner.

Step 1: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

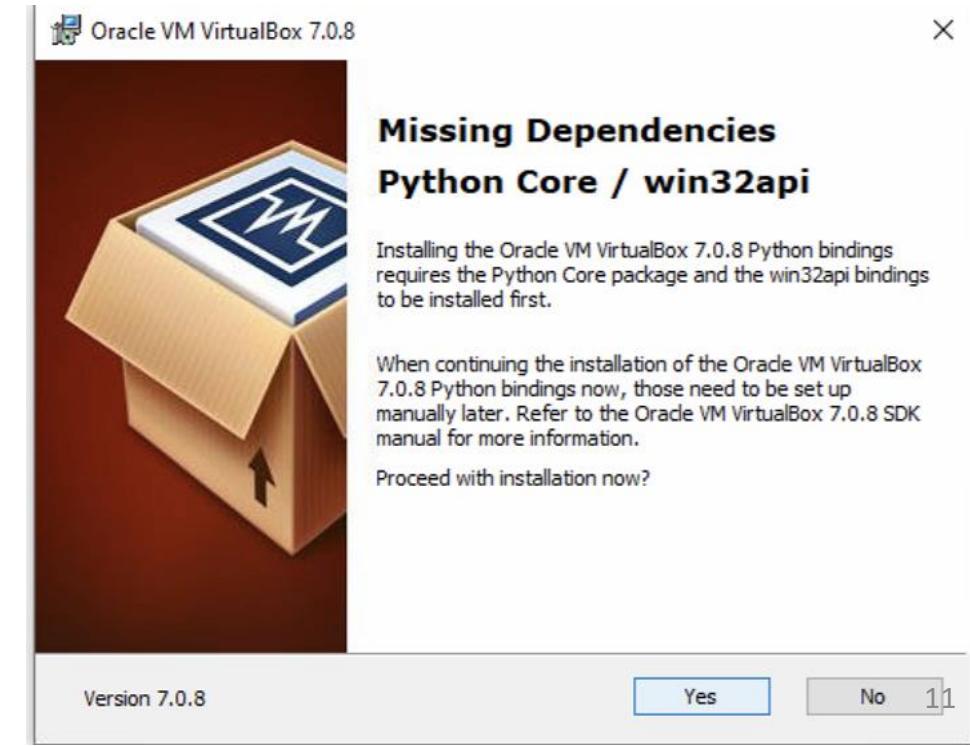
Download and install Oracle Virtual Box



Step 1: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

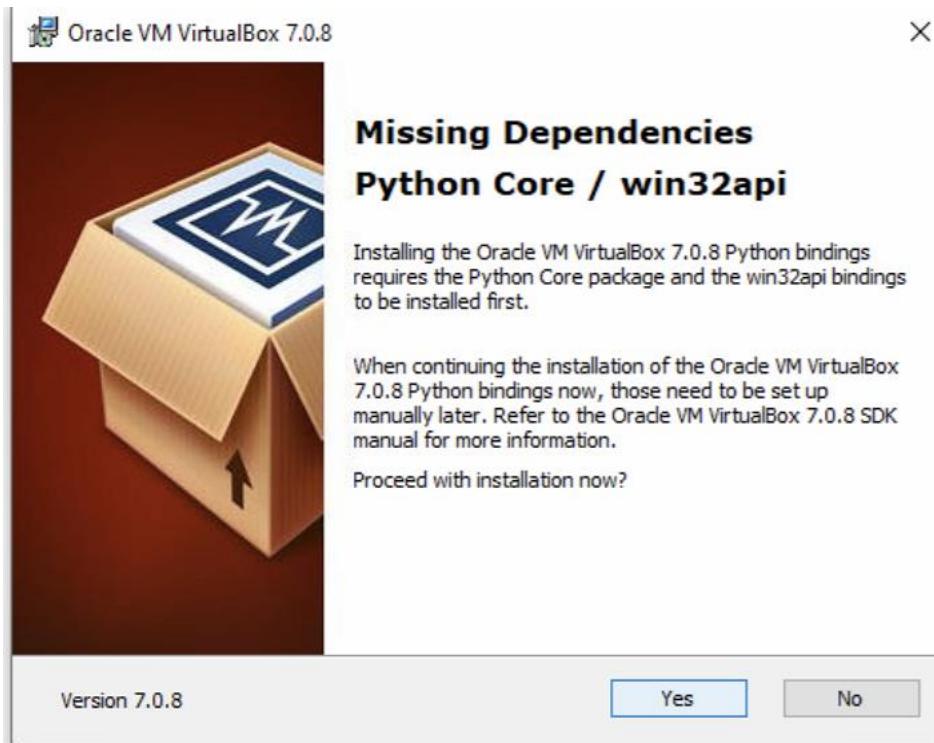
Download and install Oracle Virtual Box



Step 1: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

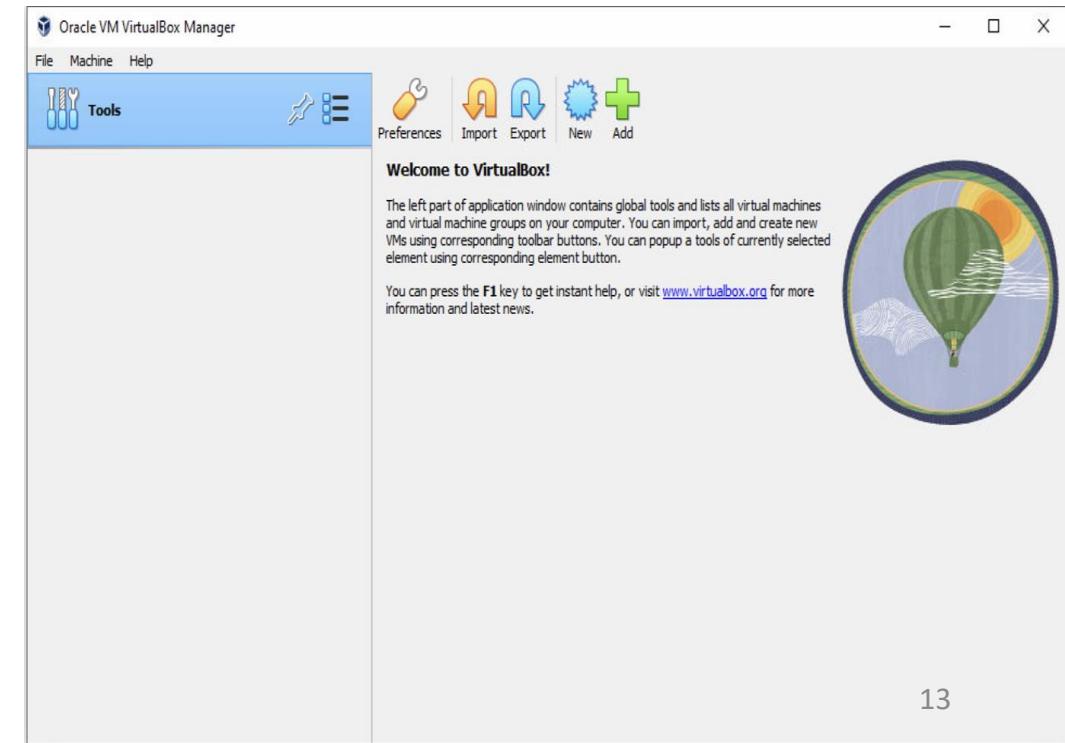
Download and install Oracle Virtual Box



Step 1: Prepare Hadoop Environment

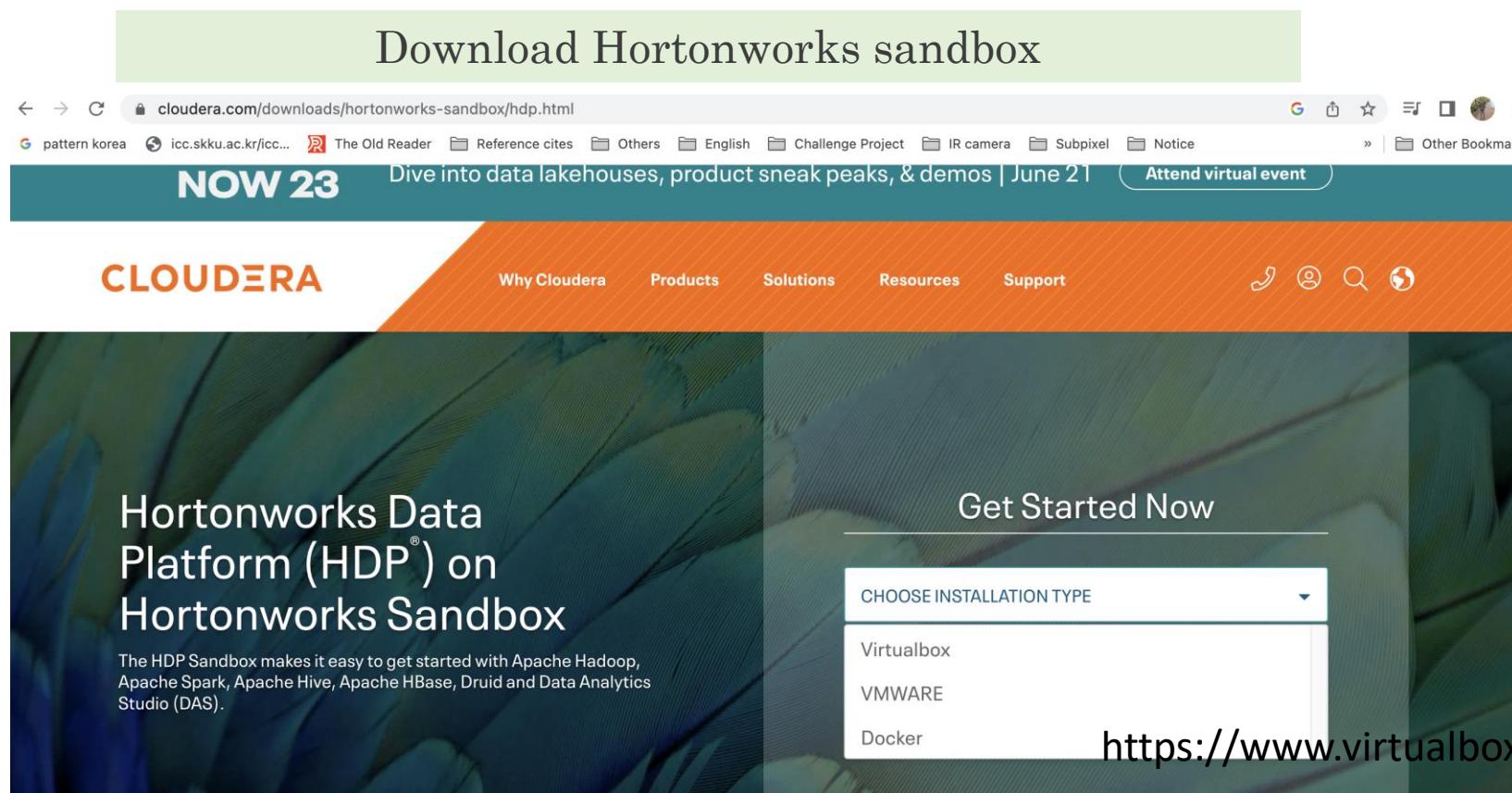
Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

Download and install Oracle Virtual Box



Step 2: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.



Step 2: Prepare Hadoop Environment

Instead of installing Hadoop from scratch, we will be using sandbox system provided by Hortonworks. The Hortonworks Sandbox is a personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials.

Download Hortonworks sandbox

Thank you for choosing Hortonworks Data Platform (HDP) on Sandbox

Sandbox HDP Virtualbox Downloads

[HDP Sandbox 3.0.1 \(Latest\)](#)

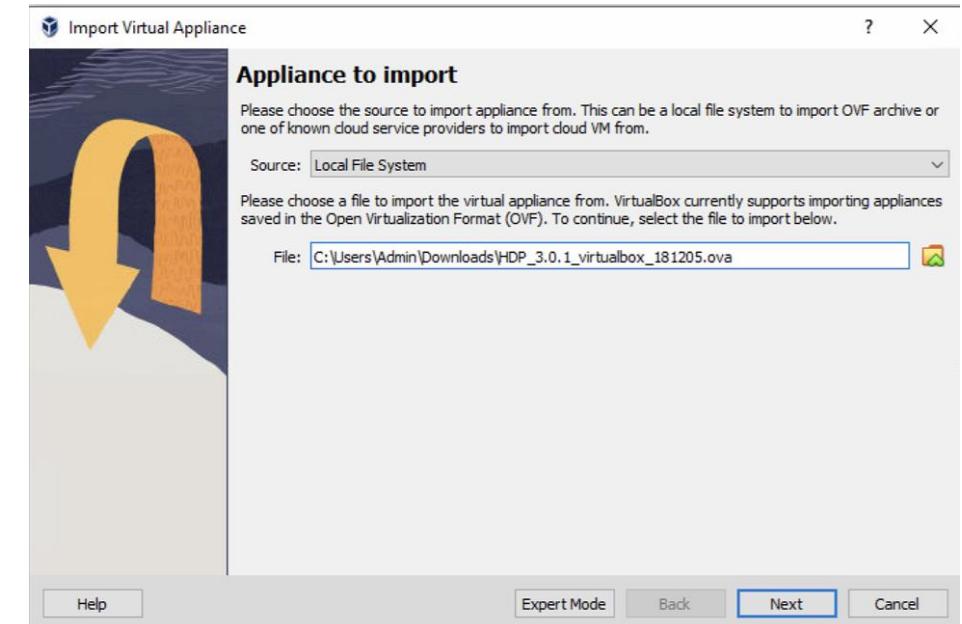
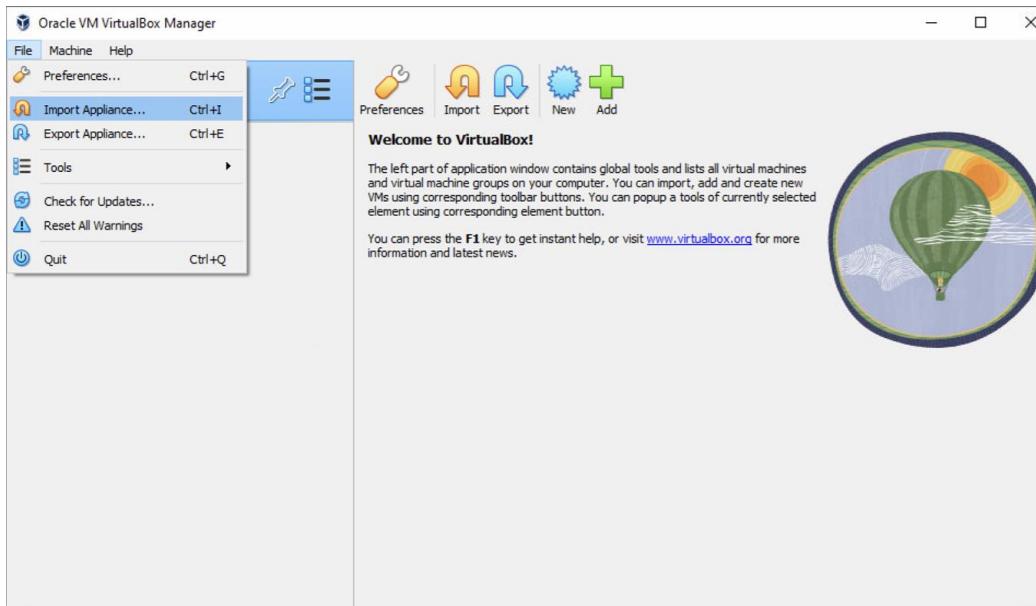
[Install Guide on VirtualBox](#)

Older Versions

- [2.6.5](#)
- [2.5.0](#)

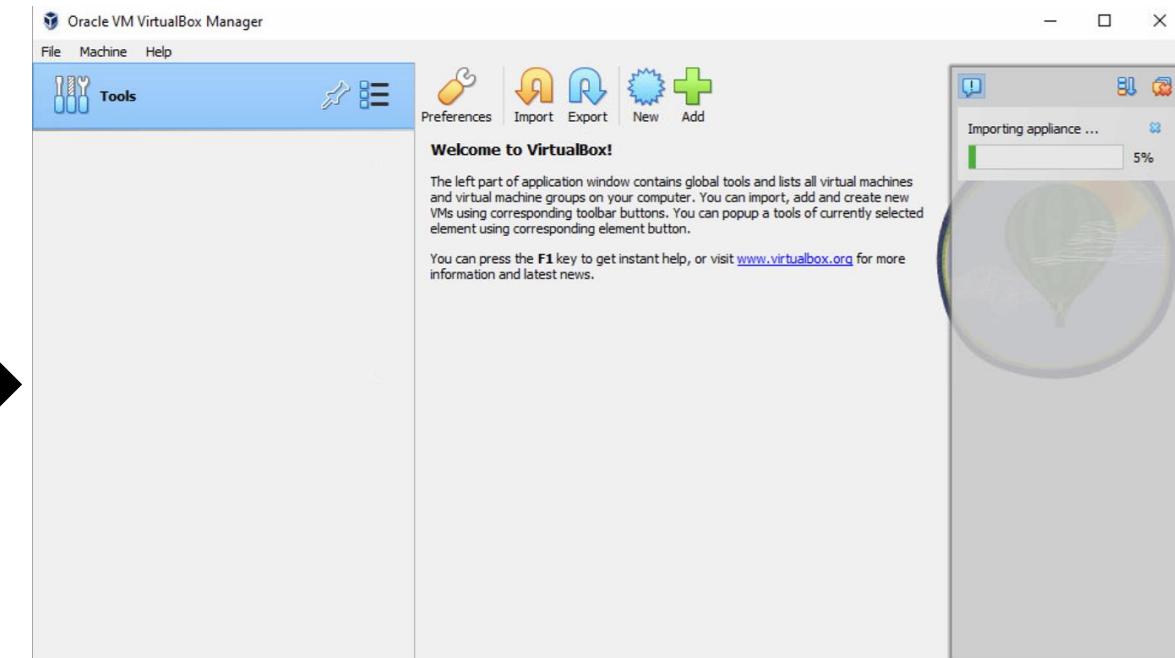
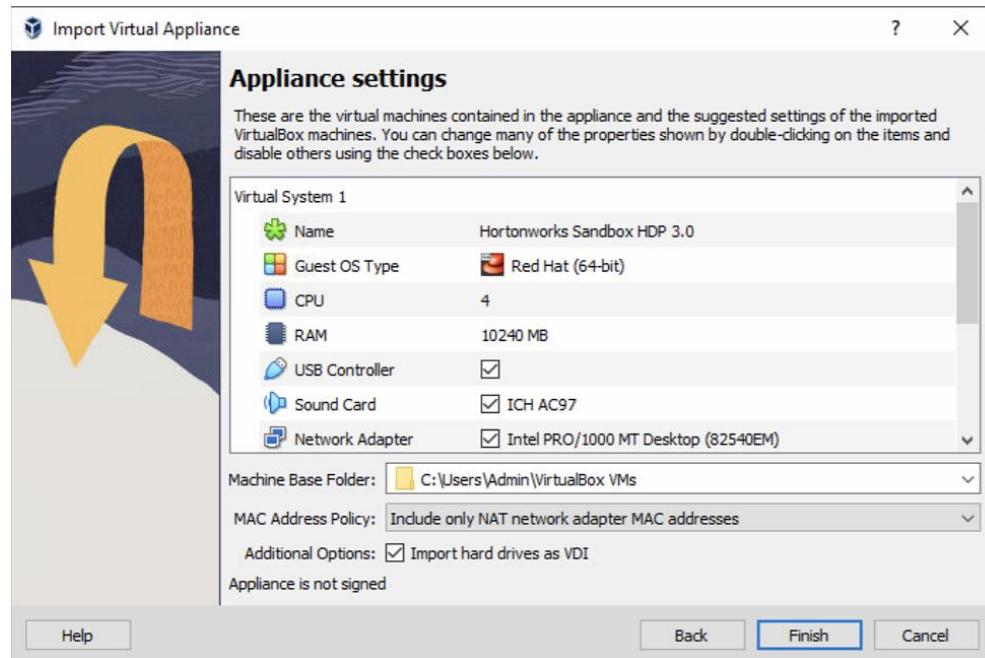
Step 3: Prepare Hadoop Environment

Open Oracle VM VirtualBox: File ->Import Appliance and select the .ova file



Step 3: Prepare Hadoop Environment

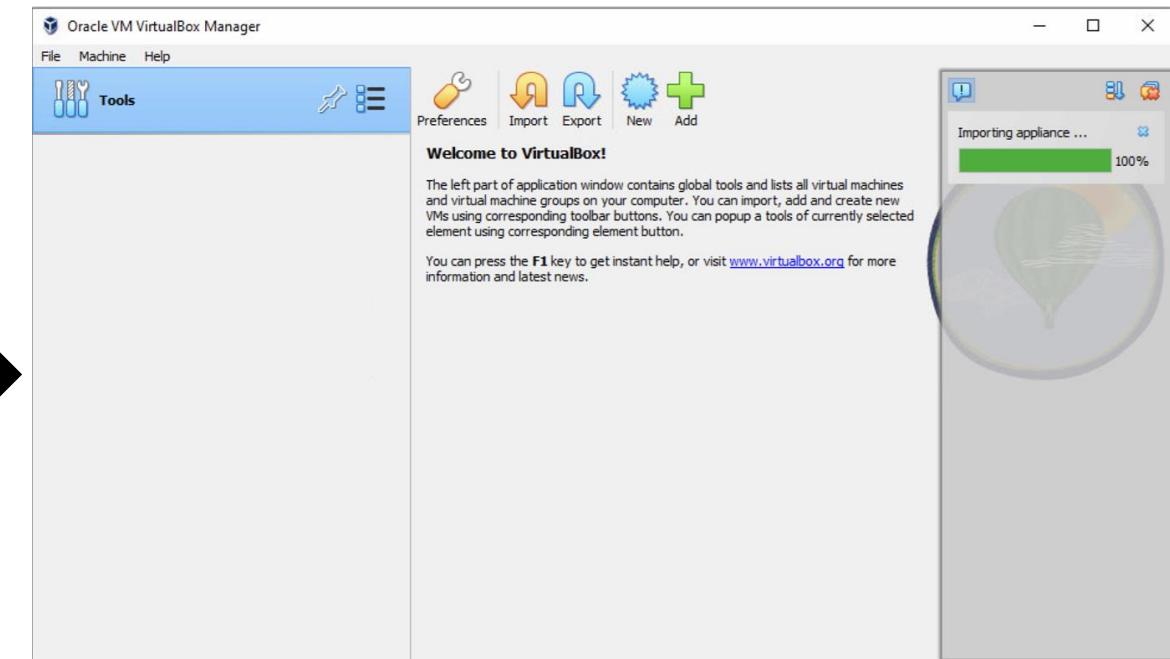
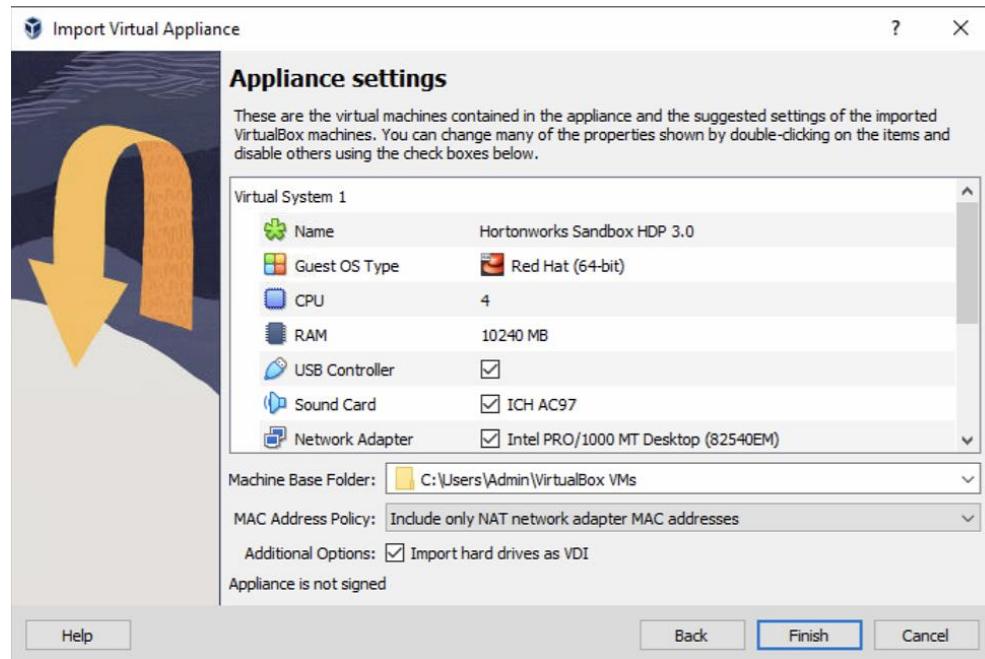
Open Oracle VM VirtualBox: File ->Import Appliance and select the .ova file



Have a quick look on Appliance Settings which would tell you that what is the guest OS on which you would be running Sandbox.

Step 3: Prepare Hadoop Environment

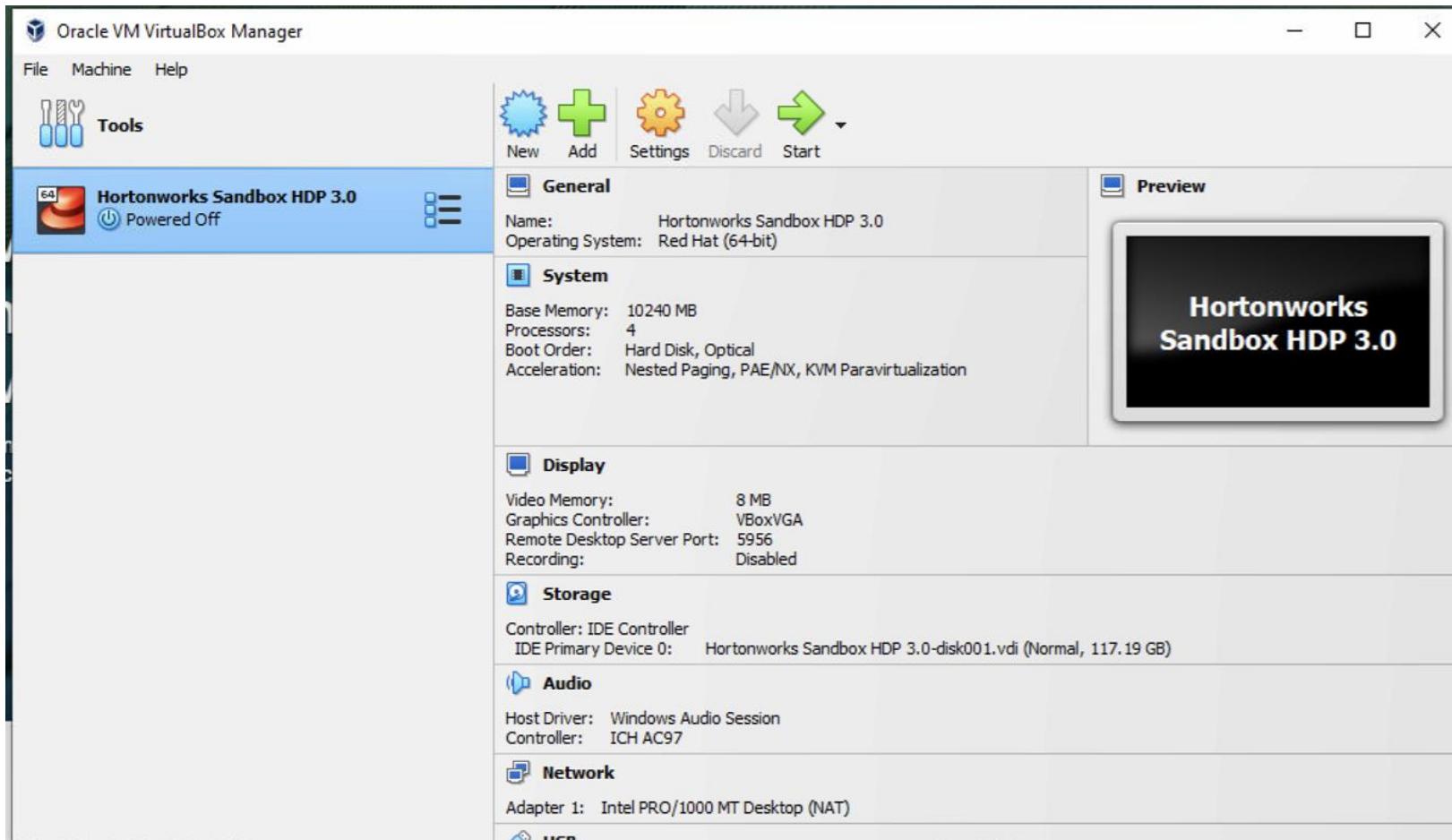
Open Oracle VM VirtualBox: File ->Import Appliance and select the .ova file



Have a quick look on Appliance Settings which would tell you that what is the guest OS on which you would be running Sandbox.

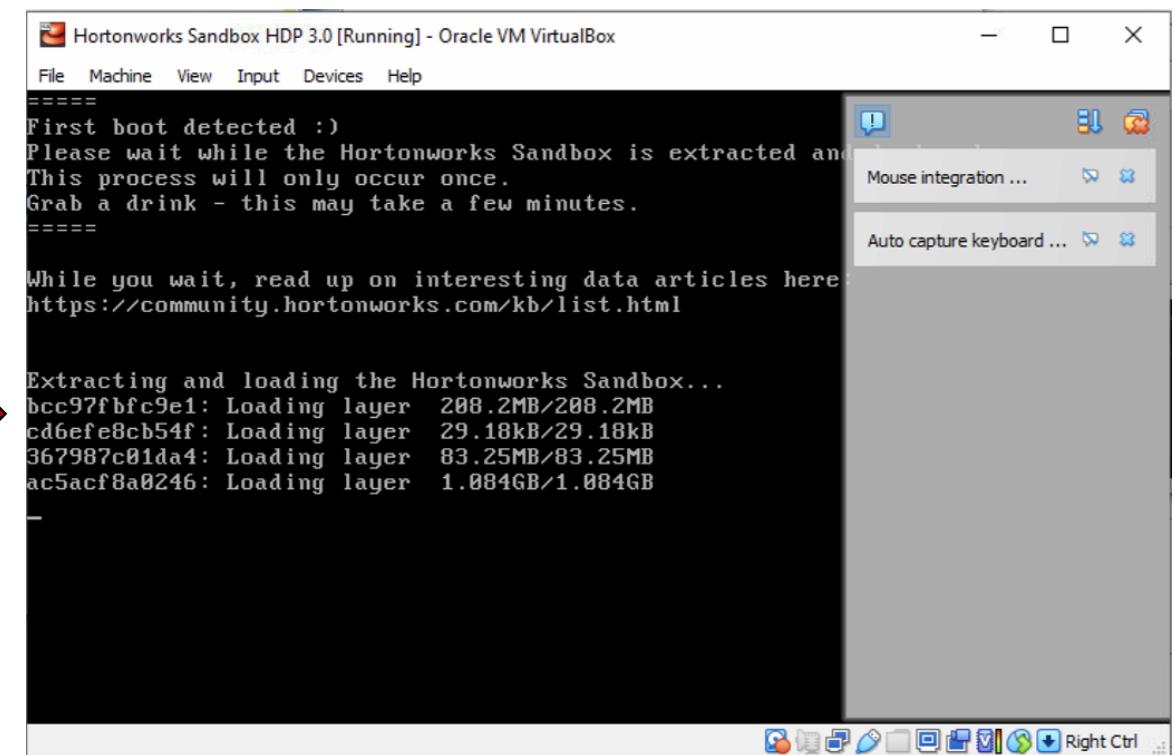
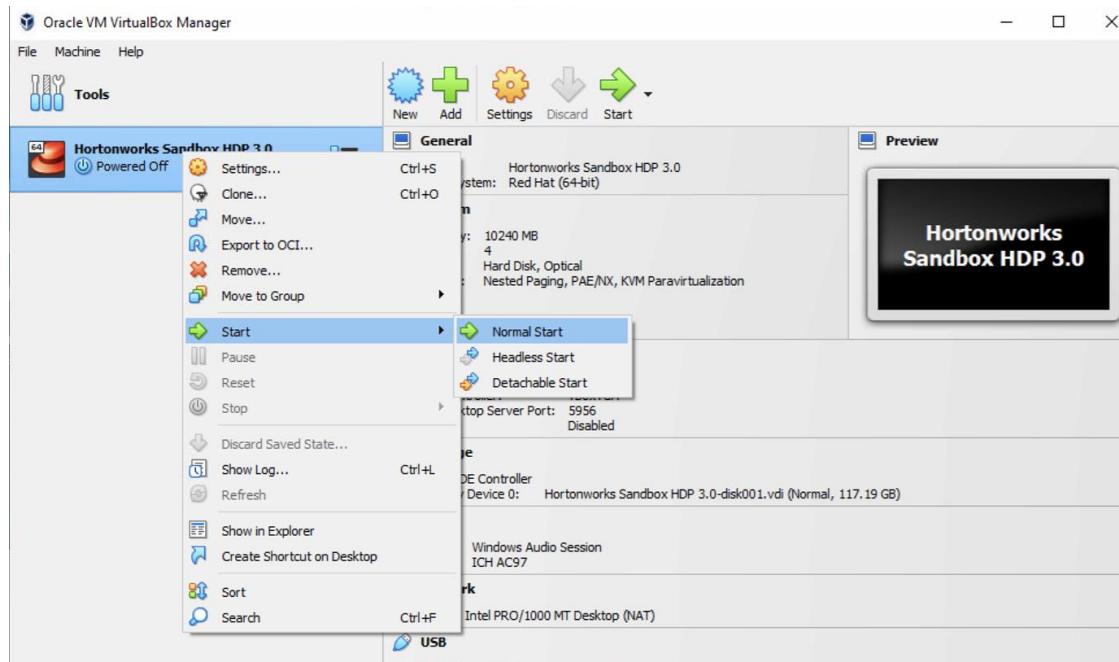
Step 3: Prepare Hadoop Environment

Open Oracle VM VirtualBox: File ->Import Appliance and select the .ova file



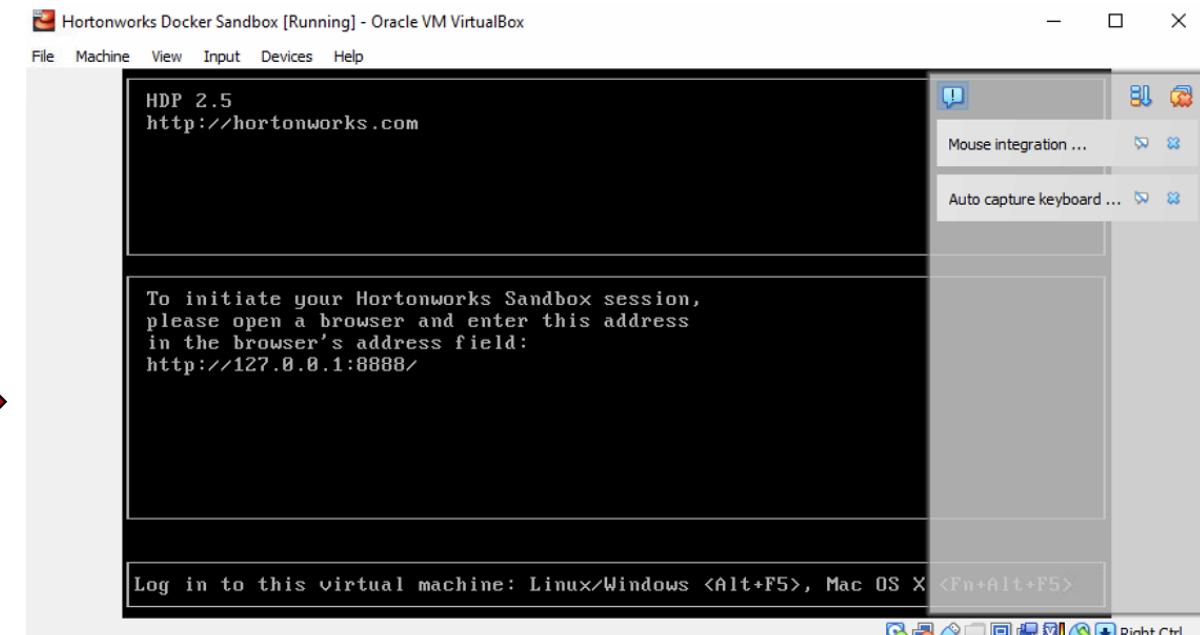
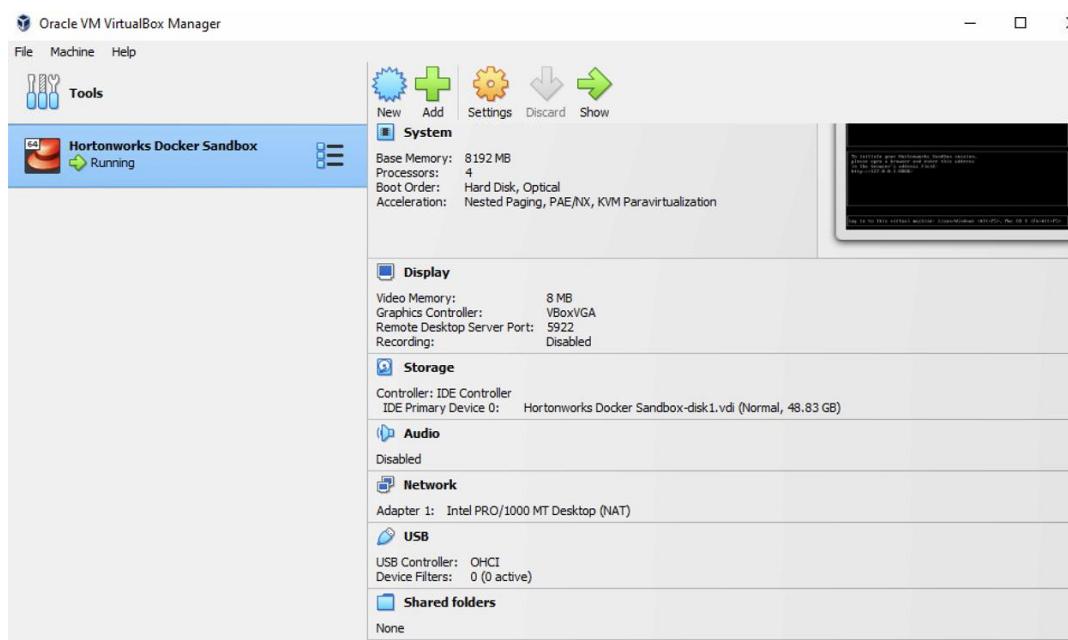
Step 3: Prepare Hadoop Environment

You will notice that virtual machine starts booting up and loading different configuration



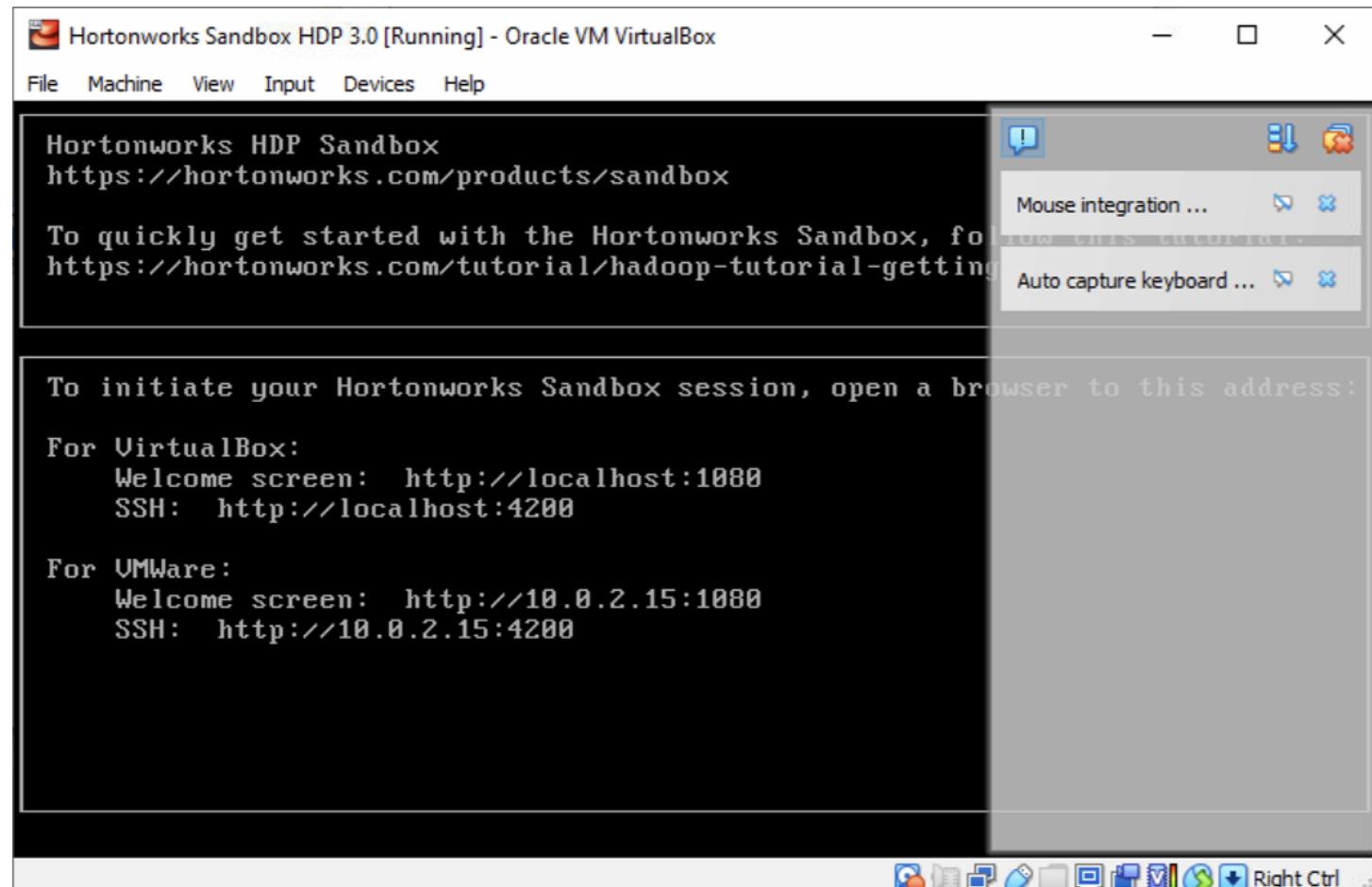
Step 3: Prepare Hadoop Environment

You will notice that virtual machine starts booting up and loading different configuration



Step 3: Prepare Hadoop Environment

Done!! We have successfully installed and configured HortonWorks SandBox



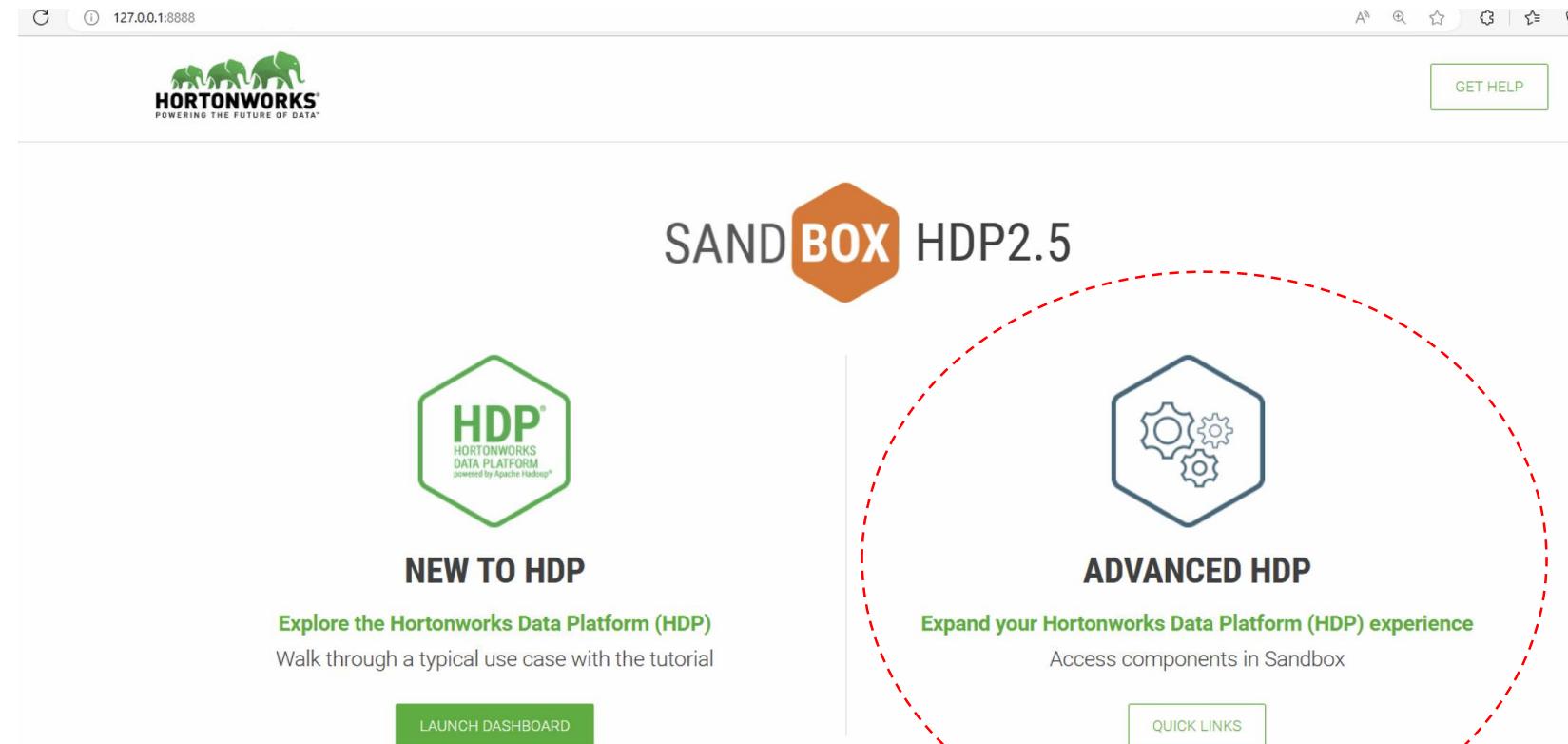
Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

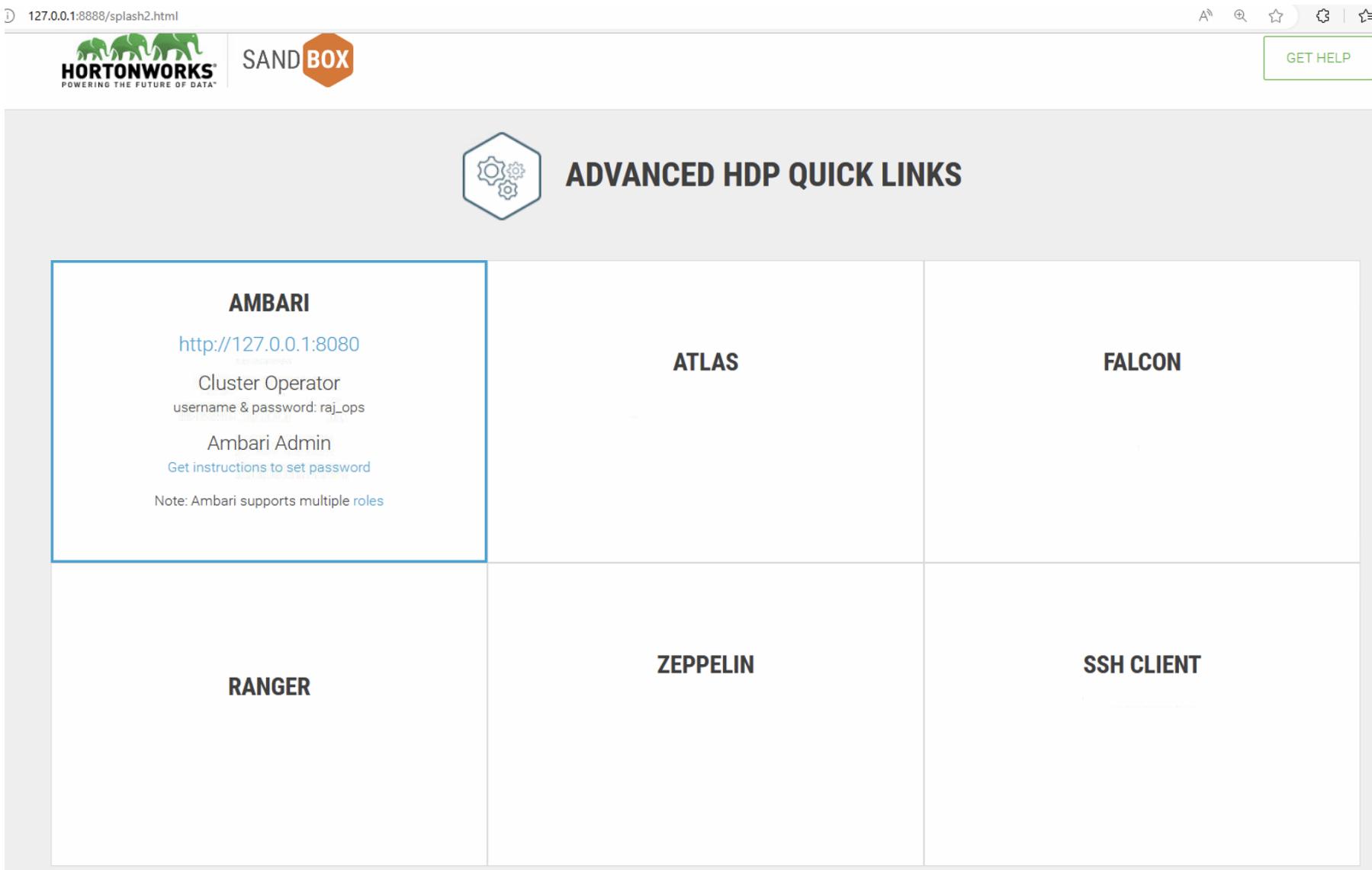
The screenshot shows a web browser window with the URL 127.0.0.1:1080/splash.html. At the top, there is a navigation bar with back, forward, and search icons. The main header features the Hortonworks logo (three green elephants) and the text "SANDBOX HDP 3.0". To the right of the header is a "GET HELP" button. Below the header, there are two main sections. The left section, enclosed in a red dashed circle, is titled "NEW TO HDP" and contains the text "Explore the Hortonworks Data Platform (HDP)" and "Walk through a typical use case with the tutorial". It includes a "LAUNCH DASHBOARD" button. The right section is titled "ADVANCED HDP" and contains the text "Expand your Hortonworks Data Platform (HDP) experience" and "Access components in Sandbox". It includes a "QUICK LINKS" button.

Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

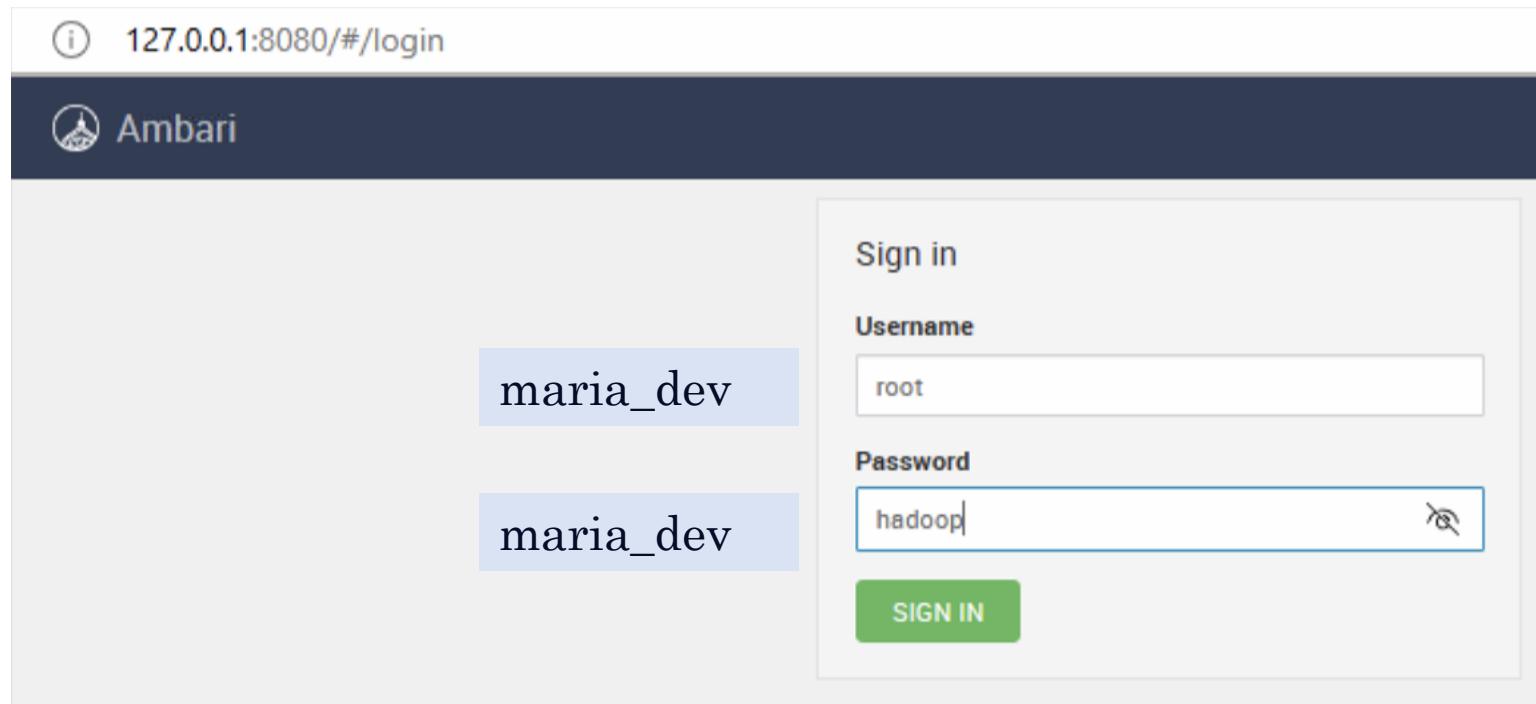


Step 4: First glance at Hortonworks sandbox



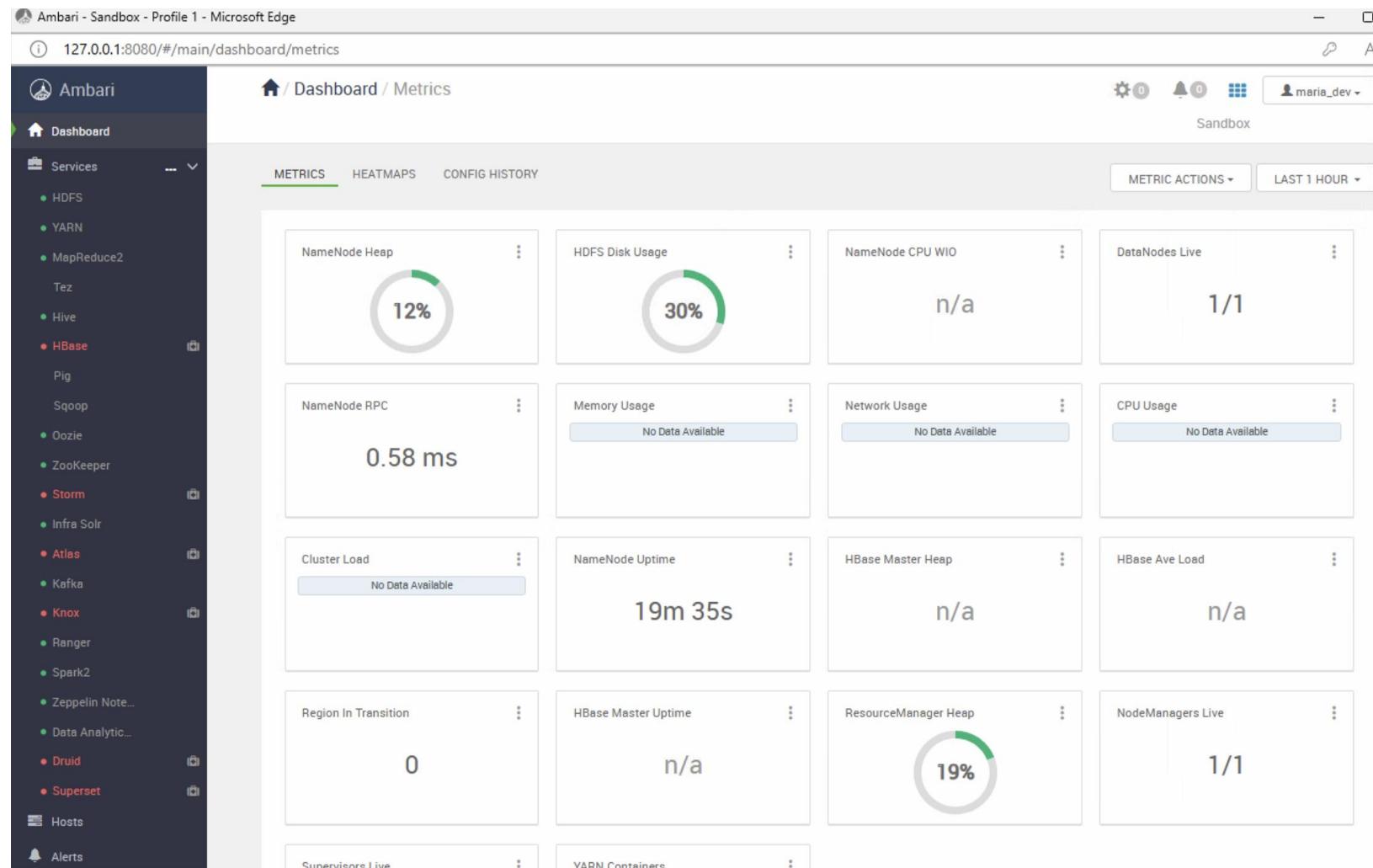
Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox



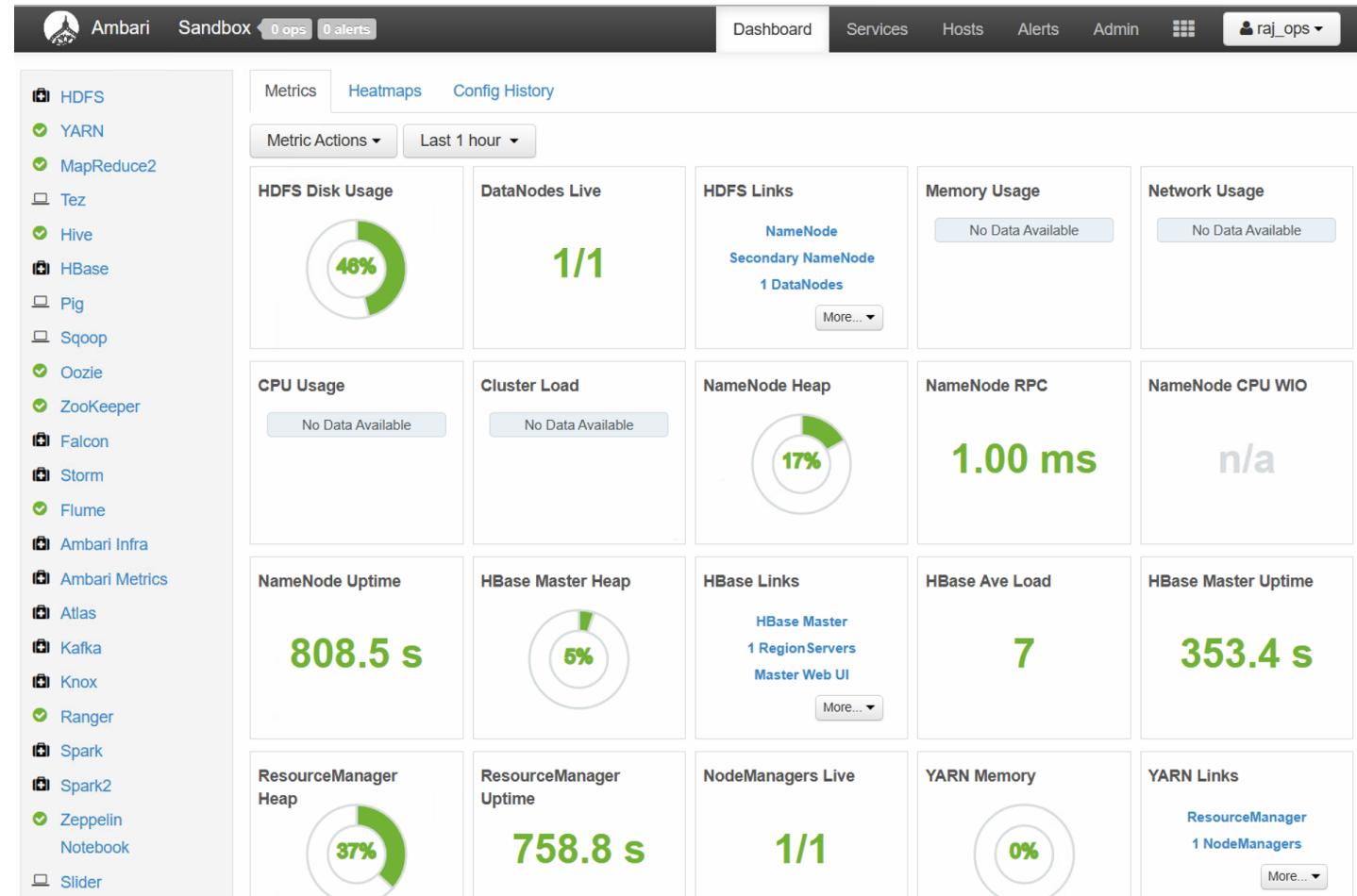
Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox



Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox



Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

The screenshot shows the Ambari interface with a red border around the main content area. At the top, there's a header bar with the Ambari logo, the text "Ambari Sandbox 0 ops 0 alerts", and navigation links for Dashboard, Services, Hosts, Alerts, Admin, and a user dropdown for "raj_ops". A sidebar on the right contains a list of views: YARN Queue Manager, Files View (selected), Hive View (highlighted in dark grey), Pig View, Storm View, and Tez View.

Your Views

- YARN Queue Manager** (1.0.0)
Manage YARN Capacity Scheduler Queues
- Files View** (1.0.0)
This view instance is auto created when the HDFS service is added to a cluster.
- Hive View** (1.5.0)
This view instance is auto created when the Hive service is added to a cluster.
- Pig View** (1.0.0)
User Interface to write and execute Pig scripts
- Storm View** (0.1.0)
Manage Storm
- Tez View** (0.7.0.2.5.0.0-1225)
Monitor and debug all Tez jobs, submitted by Hive queries and Pig scripts (auto-created)

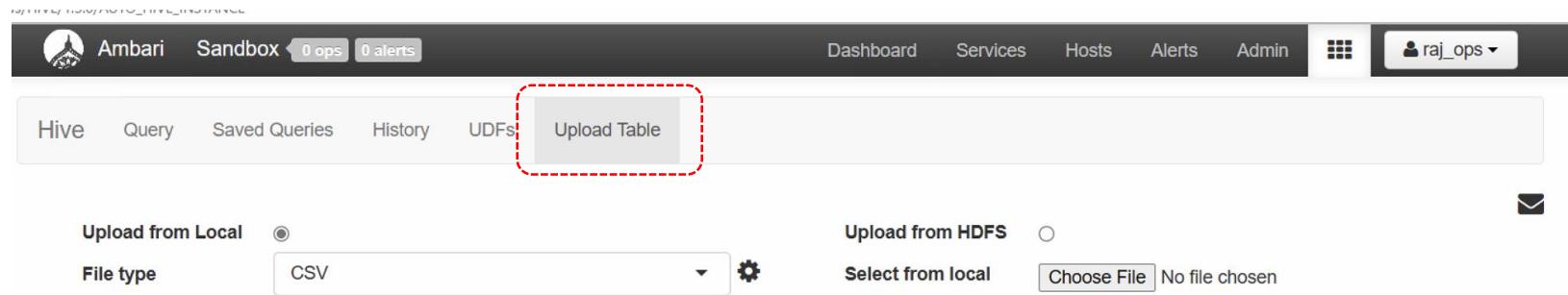
Step 4: First glance at Hortonworks sandbox

Create AIVN database

The screenshot shows the Hortonworks Data Viz interface. At the top, there are tabs for Hive, Query (which is selected), Saved Queries, History, UDFs, and Upload Table. On the left, the Database Explorer panel shows the 'aivn_database' selected, with a list of databases including 'aivn database', 'default', 'foodmart', and 'xademo'. In the center, the Query Editor panel contains a worksheet with the SQL command 'CREATE DATABASE aivn_database;'. Below the editor are buttons for Execute, Explain, Save as..., and New Worksheet. To the right, a sidebar provides options for SQL, HDFS, Tez, and a message center with 27 notifications. At the bottom, a status bar indicates 'Query Process Results (Status: UNKNOWN)'.

Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox



Advertising Dataset

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2
8.6	2.1	1	4.8
199.8	2.6	21.2	15.6

Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

The screenshot shows the Hortonworks SandBox interface with the 'Upload Table' tab selected in the top navigation bar. The page is divided into two main sections: 'Upload from Local' and 'Upload from HDFS'.
Upload from Local: This section is active, indicated by a radio button. It includes fields for 'File type' (set to 'CSV'), 'Database' (set to 'aivn_database'), and 'Stored as' (a dropdown menu showing 'aivn_database' as the current selection, along with 'default', 'foodmart', and 'xademo'). Below these are four columns labeled 'TV', 'Radio', 'Newspaper', and 'Sales', each with a 'FLOAT' dropdown menu.
Upload from HDFS: This section is inactive, indicated by an empty radio button. It includes fields for 'Select from local' (with a 'Choose File' button and 'No file chosen' message), 'Table name' (set to 'advertising'), and a checkbox for 'Contains endlines?' which is unchecked.
Data Preview: At the bottom, there is a preview of the data in the 'advertising' table:

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2

Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

The screenshot shows the Hortonworks Ambari Sandbox interface. At the top, there's a navigation bar with links for Ambari, Sandbox (0 ops, 0 alerts), Dashboard, Services, Hosts, Alerts, Admin, and a user dropdown for 'raj_ops'. Below the navigation is a toolbar with tabs for Hive, Query, Saved Queries, History, UDFs, and Upload Table. The 'Upload Table' tab is active. On the left, there are form fields for 'Upload from Local' (radio button selected), 'File type' (CSV), 'Database' (aivn_database), and 'Stored as' (ORC). A 'Contains endlines?' checkbox is also present. A modal window titled 'Upload Progress' displays a list of successful actions: 'Successfully created Actual table.', 'Successfully created Temporary table.', 'Successfully uploaded file.', and 'Waiting for insertion of rows from temporary table to actual table.' To the right of the modal, there's a note 'No file chosen' and a large empty text area. Below the form, there are four input fields labeled TV, Radio, Newspaper, and Sales, each with a dropdown menu set to 'FLOAT'. A 'Upload Table' button is located to the right of these fields. Below these fields is a table with four columns corresponding to the inputs: TV, Radio, Newspaper, and Sales. The table contains the following data:

TV	Radio	Newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12
151.5	41.3	58.5	16.5
180.8	10.8	58.4	17.9
8.7	48.9	75	7.2
57.5	32.8	23.5	11.8
120.2	19.6	11.6	13.2

Step 4: First glance at Hortonworks sandbox

Done!! We have successfully installed and configured HortonWorks SandBox

The screenshot shows the Hortonworks Data Vessel interface. At the top, there are tabs: Hive, Query (which is selected), Saved Queries, History, UDFs, and Upload Table. The main area is divided into two panes. The left pane is the Database Explorer, which lists databases: aivn_database, aivn database, advertising, tv, radio, newspaper, sales, default, foodmart, and xademo. The 'aivn_database' database is selected. The right pane is the Query Editor, showing a worksheet with the SQL command: CREATE DATABASE aivn_database;. Below the editor are buttons for Execute, Explain, Save as..., and New Worksheet. To the right of the editor is a sidebar with icons for i (Information), SQL, Settings, and TEZ, with a red notification badge showing 31.

Step 4: First glance at Hortonworks sandbox

The screenshot shows the Hortonworks Data Platform interface. The top navigation bar includes tabs for Hive, Query (which is selected), Saved Queries, History, UDFs, and Upload Table. The Database Explorer on the left shows the 'aivn_database' selected, with tables advertising, tv, radio, newspaper, and sales listed, each with a FLOAT data type. The Query Editor panel contains a 'Worksheet' section with a red dashed border, showing the query: `1 select * from advertising where sales > 10;`. Below the worksheet are buttons for Execute, Explain, Save as..., and New Worksheet. The bottom panel displays the 'Query Process Results (Status: SUCCEEDED)' with a 'Logs' tab selected and a 'Results' tab. The results table has columns: advertising.tv, advertising.radio, advertising.newspaper, and advertising.sales. The data is as follows:

advertising.tv	advertising.radio	advertising.newspaper	advertising.sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.2	45.9	69.3	12.0

Adding new DataNode to the cluster using Ambari

The screenshot shows the Ambari web interface at the URL `127.0.0.1:8080/#/main/hosts`. The top navigation bar includes links for 'Ambari', 'Sandbox', 'Dashboard', 'Services', 'Hosts' (which is highlighted with a red dashed box), 'Alerts', 'Admin', and a user dropdown for 'raj_ops'. The main content area displays a table of host information. On the left, a sidebar titled 'Actions' contains a dropdown menu with three options: 'Selected Hosts (0)', 'Filtered Hosts (1)', and 'All Hosts (1)'. The 'Add New Hosts' option is highlighted with a red dashed box and has a blue arrow pointing from it to the 'Hosts' tab in the navigation bar. The table columns include IP Address, Rack, Cores, RAM, Disk Usage, Load Avg, Versions, and Components. One host entry is visible: 'works.com' with IP '172.17.0.2', Rack '/default-rack', Cores '4 (4)', RAM '7.64GB', and Components 'HDP-2.5.0.0-1245'. The bottom right of the table area shows pagination controls: 'Show: 10' and '1 - 1 of 1'.

Adding new DataNode to the cluster using Ambari

Add Host Wizard

ADD HOST WIZARD

- Install Options** (selected)
- Confirm Hosts
- Assign Slaves and Clients
- Configurations
- Review
- Install, Start and Test
- Summary

Install Options

Enter the list of hosts to be included in the cluster and provide your SSH key.

Target Hosts

Enter a list of hosts using the Fully Qualified Domain Name (FQDN), one per line. Or use [Pattern Expressions](#)

aivn.edu.vn

Host Registration Information

Provide your [SSH Private Key](#) to automatically register hosts

No file chosen

SSH User Account

SSH Port Number

Perform [manual registration](#) on hosts and do not use SSH

Register and Confirm →

Confirm Hosts

Registering your hosts.

Please confirm the host list and remove any hosts that you do not want to include in the cluster.

Show: All (1) Installing (0) Registering (1) Success (0) Fail (0)				
<input type="checkbox"/>	Host	Progress	Status	Action
<input type="checkbox"/>	aivn.edu.vn	<div style="width: 100%; background-color: #0072bc; height: 15px;"></div>	Registering	<button> Remove</button>
Show: 25 ▾ 1 - 1 of 1 ⏪ ⏴ ⏵ ⏩ ⏹				

← Back

Next →

Hadoop on CoLab: Step by Step

- Installing Java 8
- Installing Secure Shell Server (SSHD)
- Installing Hadoop 3.2.3
- Running Hadoop in standalone mode
- Running Hadoop in Pseudo-distributed mode

Installing Java 8

```
#Kiểm tra version java hiện tại  
!java -version
```

```
openjdk version "11.0.20" 2023-07-18  
OpenJDK Runtime Environment (build 11.0.20+8-post-Ubuntu-1ubuntu122.04)  
OpenJDK 64-Bit Server VM (build 11.0.20+8-post-Ubuntu-1ubuntu122.04, mixed mode,
```

```
#Tiến hành cài đặt JDK 8  
!apt-get install openjdk-8-jdk-headless -  
qq > /dev/null
```

```
# Cấu hình sử dụng java 11  
!update-alternatives --config java
```

```
There are 2 choices for the alternative java (providing /usr/bin/java).
```

Selection	Path	Priority	Status
0	/usr/lib/jvm/java-11-openjdk-amd64/bin/java	1111	auto mode
1	/usr/lib/jvm/java-11-openjdk-amd64/bin/java	1111	manual mode
* 2	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1081	manual mode

```
Press <enter> to keep the current choice[*], or type selection number: 2
```

```
# cấu hình javac mặc định (chọn số 2)  
!update-alternatives --config javac
```

```
There are 2 choices for the alternative javac (providing /usr/bin/javac).
```

Selection	Path	Priority	Status
* 0	/usr/lib/jvm/java-11-openjdk-amd64/bin/javac	1111	auto mode
1	/usr/lib/jvm/java-11-openjdk-amd64/bin/javac	1111	manual mode
2	/usr/lib/jvm/java-8-openjdk-amd64/bin/javac	1081	manual mode

```
Press <enter> to keep the current choice[*], or type selection number: 2  
update-alternatives: using /usr/lib/jvm/java-8-openjdk-amd64/bin/javac to provide /usr/bin/javac (javac)
```

Installing Secure Shell Server (SSHD)

```
#Installing openssh-server
!apt-get install openssh-server -qq >
/dev/null
```

```
#Starting the server
!service ssh start
```

```
# xem cấu hình port
!grep Port /etc/ssh/sshd_config
```

```
#Creating a new rsa key pair with empty
password
!ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
```

```
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:dz8zNm5GNTSwNEEWHQE3gNjnTzNF2QnuVCi7zUAihNs root@ccb14c0b25d0
The key's randomart image is:
+---[RSA 3072]---+
|   o .+B#%|
|   . o =.o+B*|
|   o . o =oo o|
|   . E   oo. =.|
|       S . *o.+|
|       . o +..|
|           .B|
|               oo=|
|                   o.|
+---[SHA256]---+
```

Installing Hadoop 3.2.3

```
#Downloading Hadoop 3.2.3
```

```
!wget -q
https://archive.apache.org/dist/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz
```

```
#Untarring the file
```

```
!sudo tar -xzf hadoop-3.2.3.tar.gz
#Removing the tar file
!rm hadoop-3.2.3.tar.gz
```

```
#Copying the hadoop files to
```

```
user/local
```

```
!cp -r hadoop-3.2.3/ /usr/local/
#-r copy directories recursively
```

```
#Exploring hadoop-3.2.3/etc/hadoop directory
```

```
!ls /usr/local/hadoop-3.2.3/etc/hadoop
#we can see various configuration files of
hadoop
```

```
capacity-scheduler.xml
configuration.xsl
container-executor.cfg
core-site.xml
hadoop-env.cmd
hadoop-env.sh
hadoop-metrics2.properties
hadoop-policy.xml
hadoop-user-functions.sh.example
hdfs-site.xml
httpfs-env.sh
httpfs-log4j.properties
httpfs-signature.secret
httpfs-site.xml
kms-acls.xml
kms-env.sh
```

```
kms-log4j.properties
kms-site.xml
log4j.properties
mapred-env.cmd
mapred-env.sh
mapred-queues.xml.template
mapred-site.xml
shellprofile.d
ssl-client.xml.example
ssl-server.xml.example
user_ec_policies.xml.template
workers
yarn-env.cmd
yarn-env.sh
yarnservice-log4j.properties
yarn-site.xml
```

Running Hadoop in standalone mode

```
#Exploring mapreduce tools
```

```
!ls
```

```
$HADOOP_HOME/share/hadoop/mapreduce/*.jar
```

```
#Exploring the examples of programs available
```

```
!$HADOOP_HOME/bin/hadoop jar
```

```
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar
```

```
#Usage of the wordcount MapReduce program
```

```
!$HADOOP_HOME/bin/hadoop jar
```

```
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.3.jar wordcount
```

```
/usr/local/hadoop-3.2.3/share/hadoop/mapreduce/hadoop-mapreduce-client-app-3.2.3.jar  
/usr/local/hadoop-3.2.3/share/hadoop/mapreduce/hadoop-mapreduce-client-common-3.2.3.jar  
/usr/local/hadoop-3.2.3/share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.2.3.jar  
/usr/local/hadoop-3.2.3/share/hadoop/mapreduce/hadoop-mapreduce-client-hs-3.2.3.jar
```

An example program must be given as the first argument.

Valid program names are:

aggregatewordcount: An Aggregate based map/reduce program that counts the words in a file.
aggregatewordhist: An Aggregate based map/reduce program that computes histograms of word frequencies.
bbp: A map/reduce program that uses Bailey–Borwein–Plouffe to compute pi to many digits.
dbcount: An example job that counts the pageview counts from a database.
distbbp: A map/reduce program that uses a BBP-type formula to compute pi to many digits.
grep: A map/reduce program that counts the matches of a regex in the input files.
join: A job that effects a join over sorted, equally partitioned datasets.
multifilewc: A job that counts words from several files.
pentomino: A map/reduce tile laying program to find solutions to pentomino puzzles.
pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
randomtextwriter: A map/reduce program that writes 10GB of random text.
randomwriter: A map/reduce program that writes 10GB of random data per node.
secondarysort: An example defining a secondary sort to the reduce.
sort: A map/reduce program that sorts the data written by the random writer.
sudoku: A sudoku solver.
teragen: Generate data for the terasort.
terasort: Run the terasort.
teravalidate: Checking results of terasort.
wordcount: A map/reduce program that counts the words in the input file.
wordmean: A map/reduce program that counts the average length of the words in the input file.
wordmedian: A map/reduce program that counts the median length of the words in the input file.
wordstandarddeviation: A map/reduce program that counts the standard deviation of the word lengths in the input file.



Running Hadoop in standalone mode

```
#Running MapReduce program wordcount
#the output directory will be created
automatically
!${HADOOP_HOME}/bin/hadoop jar
${HADOOP_HOME}/share/hadoop/mapreduce/hadoop-
mapreduce-examples-3.2.3.jar wordcount
/content/101.txt /content/output
```

```
2023-08-04 12:21:33,098 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2023-08-04 12:21:33,229 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2023-08-04 12:21:33,229 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2023-08-04 12:21:33,423 INFO input.FileInputFormat: Total input files to process : 1
2023-08-04 12:21:33,474 INFO mapreduce.JobSubmitter: number of splits:1
2023-08-04 12:21:33,674 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1552352151_0001
2023-08-04 12:21:33,674 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-08-04 12:21:33,923 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2023-08-04 12:21:33,924 INFO mapreduce.Job: Running job: job_local1552352151_0001
```

Running Hadoop in Pseudo-distributed mode

```
#Adding required property to core-site.xml file
!sed -i '/<configuration>/a\
<property>\n\
<name>fs.defaultFS</name>\n\
<value>hdfs://localhost:9000</value>\n\
</property>' \
$HADOOP_HOME/etc/hadoop/core-site.xml
```

```
<configuration>
<property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

```
#Adding required property to hdfs-site.xml file
#Since we are running Hadoop in only one machine,
# a replication factor greater than 1 does not make
# sense
!sed -i '/<configuration>/a\
<property>\n\
<name>dfs.replication</name>\n\
<value>1</value>\n\
</property>' \
$HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

```
<configuration>
<property>
    <name>dfs.replication</name>
    <value>1</value>
</property>

</configuration>
```

Running Hadoop in Pseudo-distributed mode

```
#Adding required properties to mapred-site.xml file
!sed -i '/<configuration>/a\
<property>\n\
<name>mapreduce.framework.name</name>\n\
<value>yarn</value>\n\
</property>\n\
<property>\n\
<name>mapreduce.application.classpath</name>\n\
<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>\n\
</property>' \
$HADOOP_HOME/etc/hadoop/mapred-site.xml
```

```
<configuration>
<property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
</property>
<property>
    <name>mapreduce.application.classpath</name>
    <value>$HAD00P_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
</property>

</configuration>
```

Running Hadoop in Pseudo-distributed mode

```
#Adding required properties to yarn-site.xml file
!sed -i '</configuration>/a\
<property>\n\
<description>The hostname of the RM.</description>\n\
<name>yarn.resourcemanager.hostname</name>\n\
<value>localhost</value>\n\
</property>\n\
<property>\n\
<name>yarn.nodemanager.aux-services</name>\n\
<value>mapreduce_shuffle</value>\n\
</property>\n\
<property>\n\
<name>yarn.nodemanager.env-whitelist</name>\n\
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_YARN_HOME,HADOOP_YARN_HOME</value>
</property>' \
$HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Hadoop vs. Spark

Performance



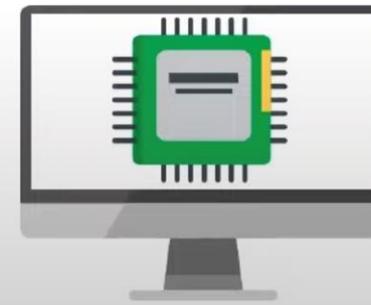
Hadoop is generally **slow** as it performs operations on the disk and **cannot deliver** near **real-time** analytics from the data



No real-time analytics



Spark runs **100 times faster** in-memory, and **10 times faster** on disk. If Spark runs on YARN with other resources demanding services, there could be major degradation



Faster in-memory processing

Hadoop vs. Spark

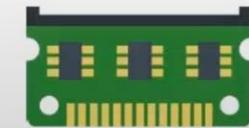
Cost



Hadoop is less expensive as it is an **open-source software**. It requires more memory on disk which is relatively an **inexpensive commodity**



Spark is **open-source** but requires a lot of RAM to run in-memory. This **increases the cluster size** and its cost

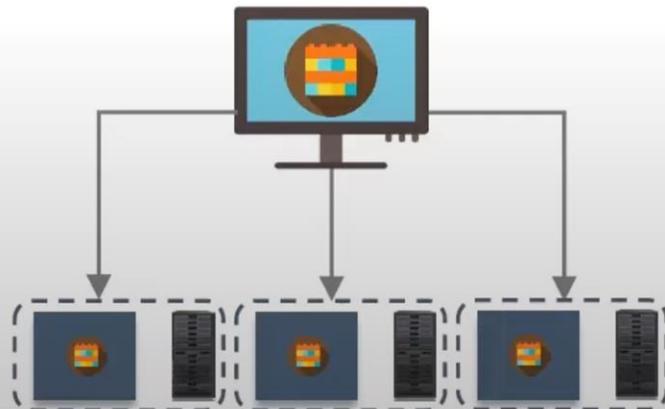


Hadoop vs. Spark

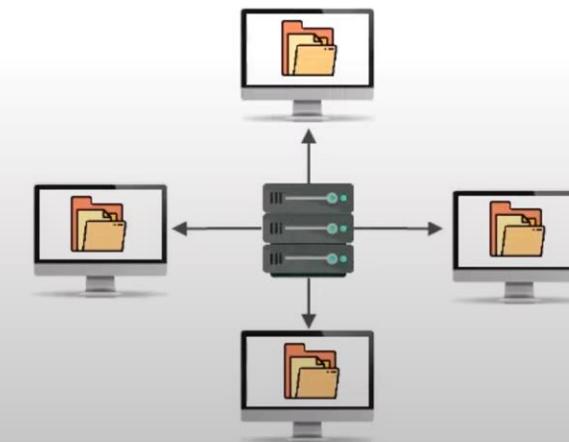
Fault Tolerance



Hadoop is **highly fault-tolerant** because it was designed to **replicate data** across many nodes. Each file is split into blocks and replicated numerous times across many machines.



Spark uses **Resilient Distributed Datasets** (RDDs), which are fault-tolerant collections of elements that can be operated on in parallel.

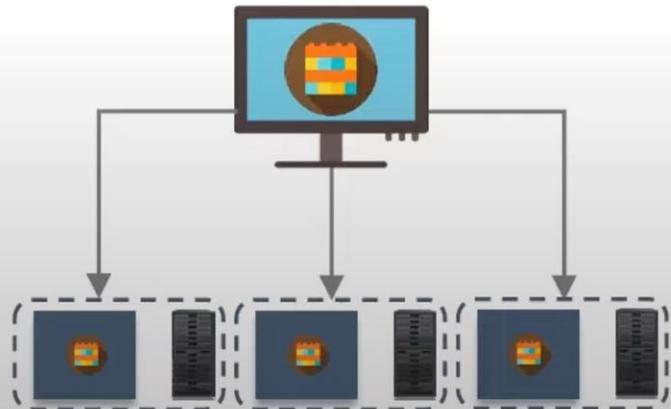


Hadoop vs. Spark

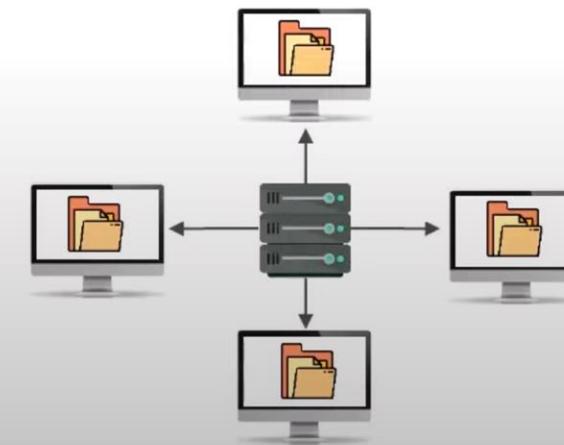
Fault Tolerance



Hadoop is **highly fault-tolerant** because it was designed to **replicate data** across many nodes. Each file is split into blocks and replicated numerous times across many machines.



Spark uses **Resilient Distributed Datasets** (RDDs), which are fault-tolerant collections of elements that can be operated on in parallel.



Hadoop vs. Spark

Ease of Use



Hadoop's MapReduce has **no interactive mode** and is complex. It needs to handle low-level APIs to process the data, which requires lots of coding



Spark supports **user-friendly APIs** for different languages. It has an **interactive mode** and provides intermediate feedback for queries and actions

```
[Select Command Prompt - spark-shell]
Press any key to continue . . .
07/31/2019 11:52 PM          1,180 spark-submit.cmd
07/31/2019 11:52 PM          1,125 spark-submitZ.cmd
07/31/2019 11:52 PM          1,039 spark-kill4k
07/31/2019 11:52 PM          1,168 sparkR.cmd
07/31/2019 11:52 PM          1,097 sparkR2.cmd
27 File(s)      47,335 bytes
2 Dir(s)   913,701,969,232 bytes free

C:\spark-2.4.3-bin-hadoop2.6\bin\spark-shell
19/09/01 00:59:01 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-Java class instead, which may not perform well.
Using Spark's default Log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://10.0.0.10:4040
Spark context available as `sc` (master = local[*], app id = local[1564613946783]),
Spark session available as `spark`.
Welcome to

    \_____
   /       \
  /  V   V  \
 /  / \ / \  \
/  /  /  /  \ \
\  \  \  \  / \
 \  \  \  /  /
  \  \  /  /
   \  /  /
    \/
version 2.4.3

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_221)
Type in expressions to have them evaluated.
Type help for more information.
```

Hadoop vs. Spark

Language Support



Hadoop framework is developed in **Java** programming language. While, MapReduce applications can be written in **Python**, **R** and **C++**



MapReduce supports programming languages



Apache Spark is developed in **Scala** language and supports other programming languages like **Python**, **R**, and **Java**



Spark supports other programming languages

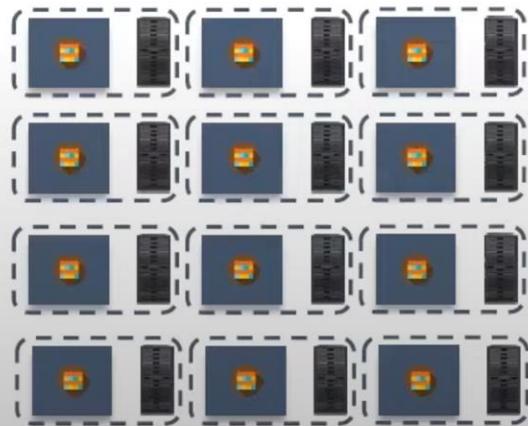


Hadoop vs. Spark

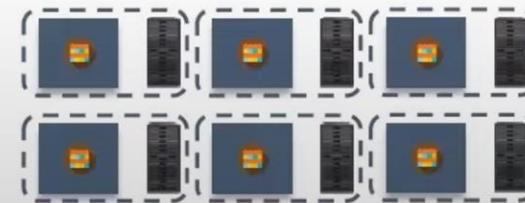
Scalability



Hadoop is **highly scalable** as we can add n number of nodes in the cluster. Yahoo reportedly used a **42,000** node Hadoop cluster



The largest known Spark cluster has **8,000** nodes. But as big data grows, it's expected that cluster sizes will increase to maintain throughput expectations.



Hadoop vs. Spark

Security



Hadoop supports [Kerberos](#) and [LDAP](#) for [authentication](#). It also supports [access control lists](#) (ACLs) and a traditional file permissions model



Spark's security is a bit sparse as it supports [authentication via passwords](#). If you run Spark on HDFS, it can use [HDFS ACLs](#) and [file-level permissions](#). Additionally, Spark can run on YARN, giving it the capability of using Kerberos authentication.



Hadoop vs. Spark

Machine Learning



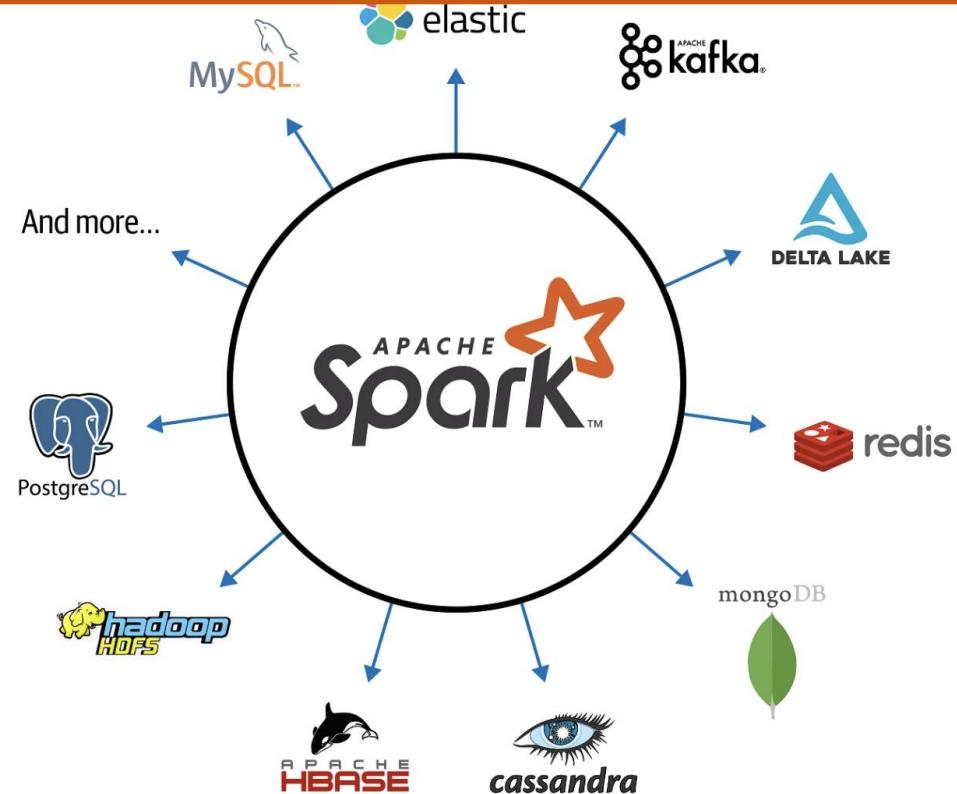
Hadoop uses **Mahout** for **processing data** and **building models**. Also, **Samsara**, a Scala-backed DSL language can be used for in-memory algebraic operations and allows users to write their own algorithms



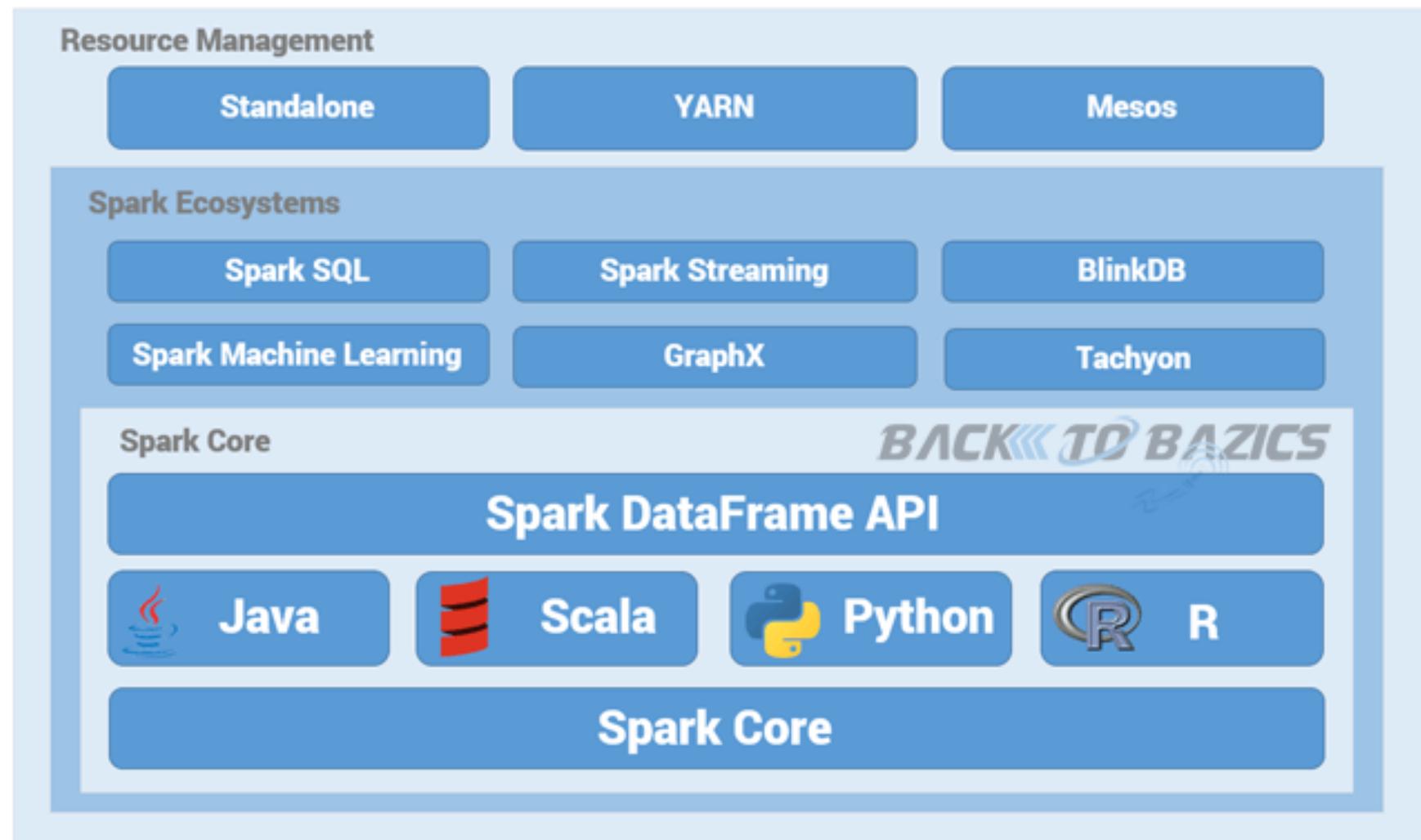
Spark has a built-in **machine learning library** that can be used for classification, and regression. It can also build machine-learning pipelines with hyperparameter tuning



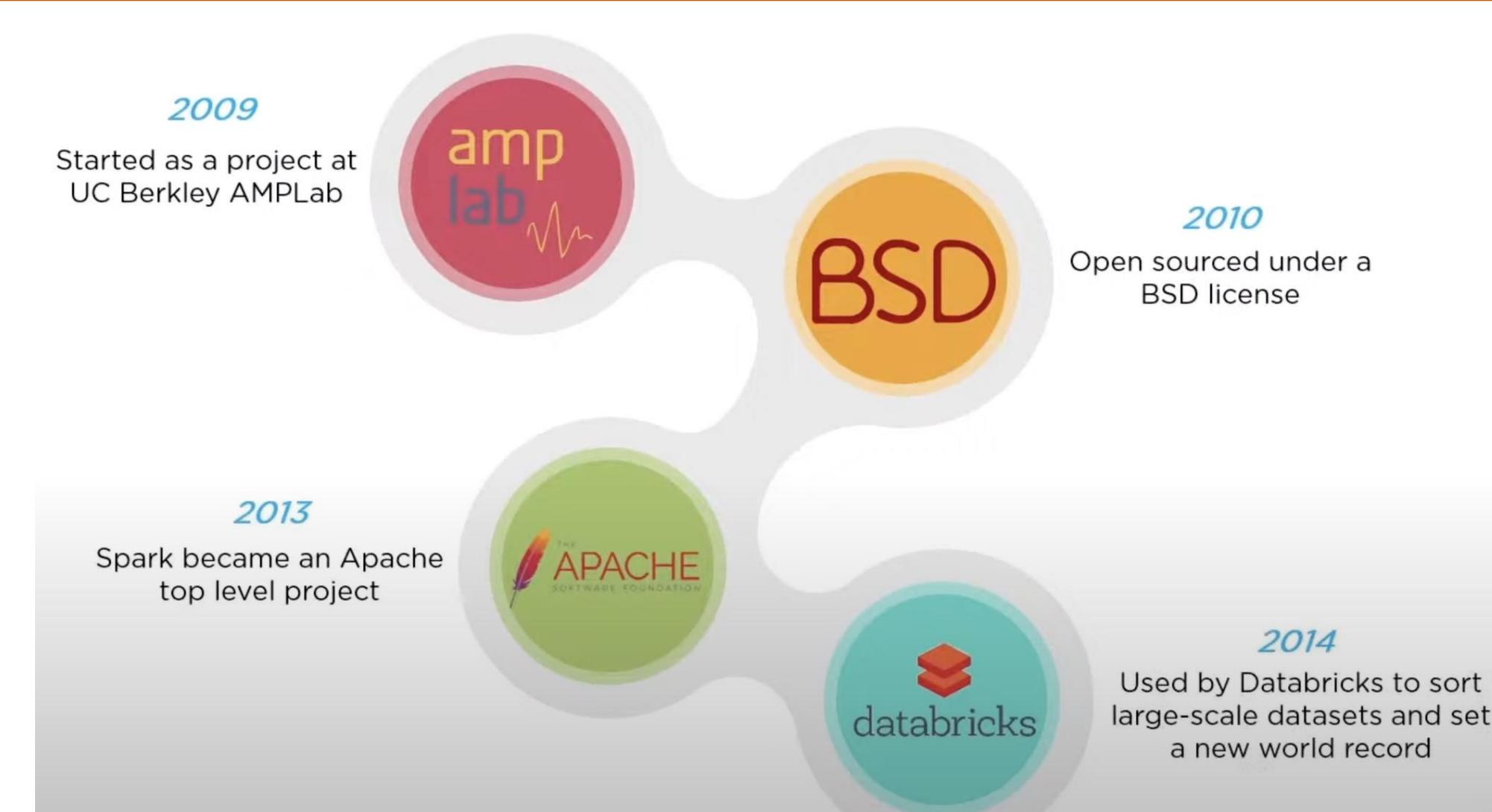
Data Storage Technologies



Whereas Hadoop reads and writes files to HDFS, Spark processes data in RAM using a concept known as an RDD, Resilient Distributed Dataset.



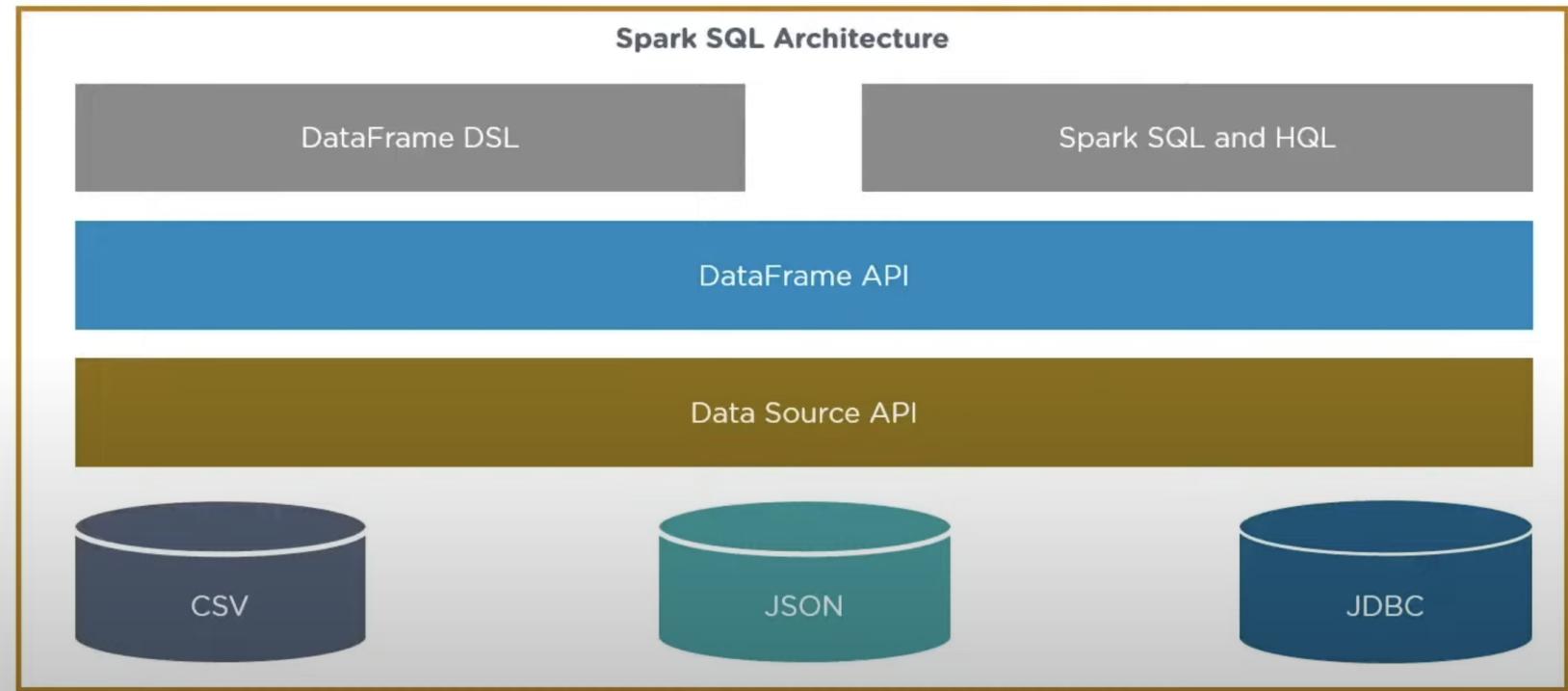
History of Apache Spark



Spark SQL

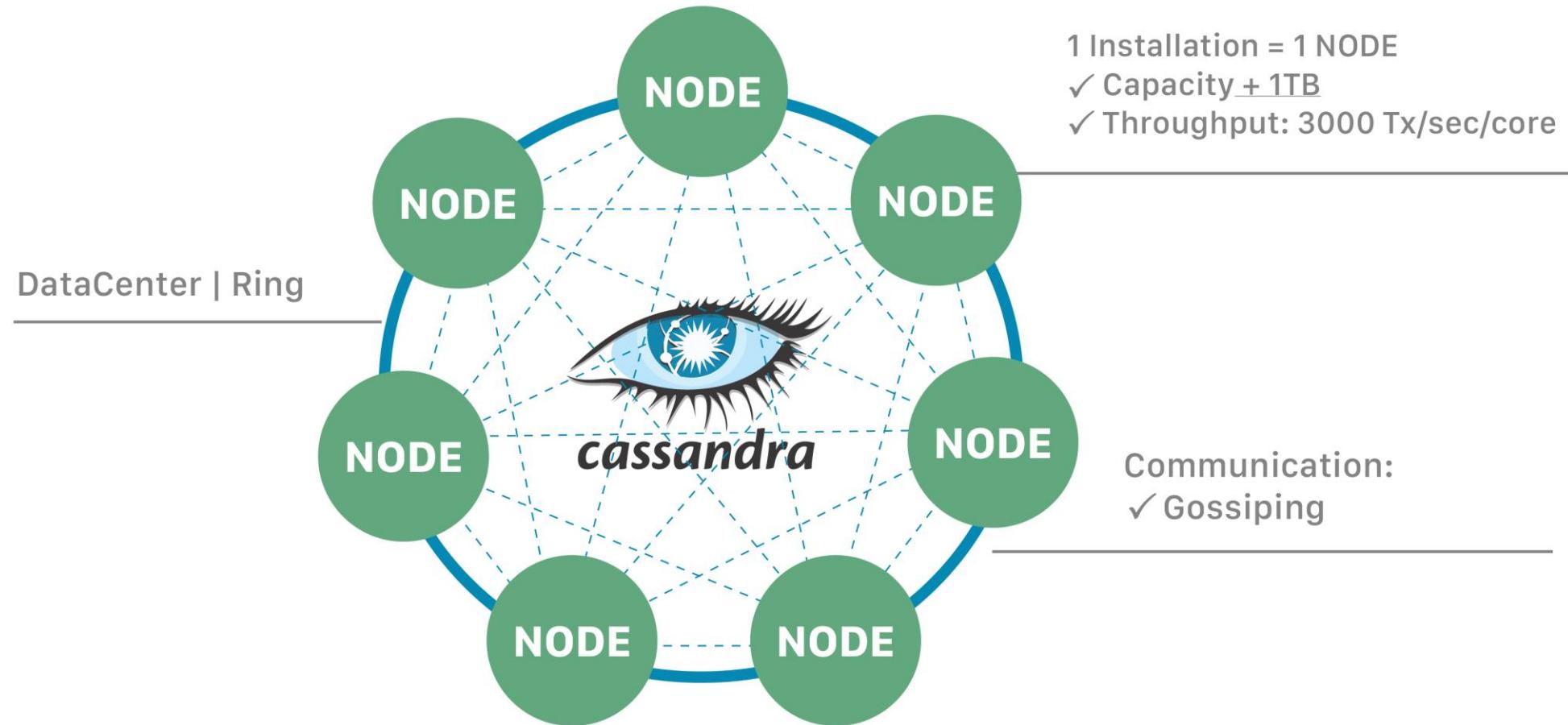
Spark SQL

Spark SQL framework component is used for structured and semi-structured data processing



Data Storage Technologies

ApacheCassandra™ = NoSQL Distributed Database



Summary



Summary

Why is Hadoop used ?

- ❑ it can handle a large amount of data quickly
- ❑ all the steps mentioned above are automatic
- ❑ it is fully open source
- ❑ it is compatible with all platforms since written in Java
- ❑ we can add servers and remove some dynamically
- ❑ it handles failure cases

Summary

When to use Hadoop?

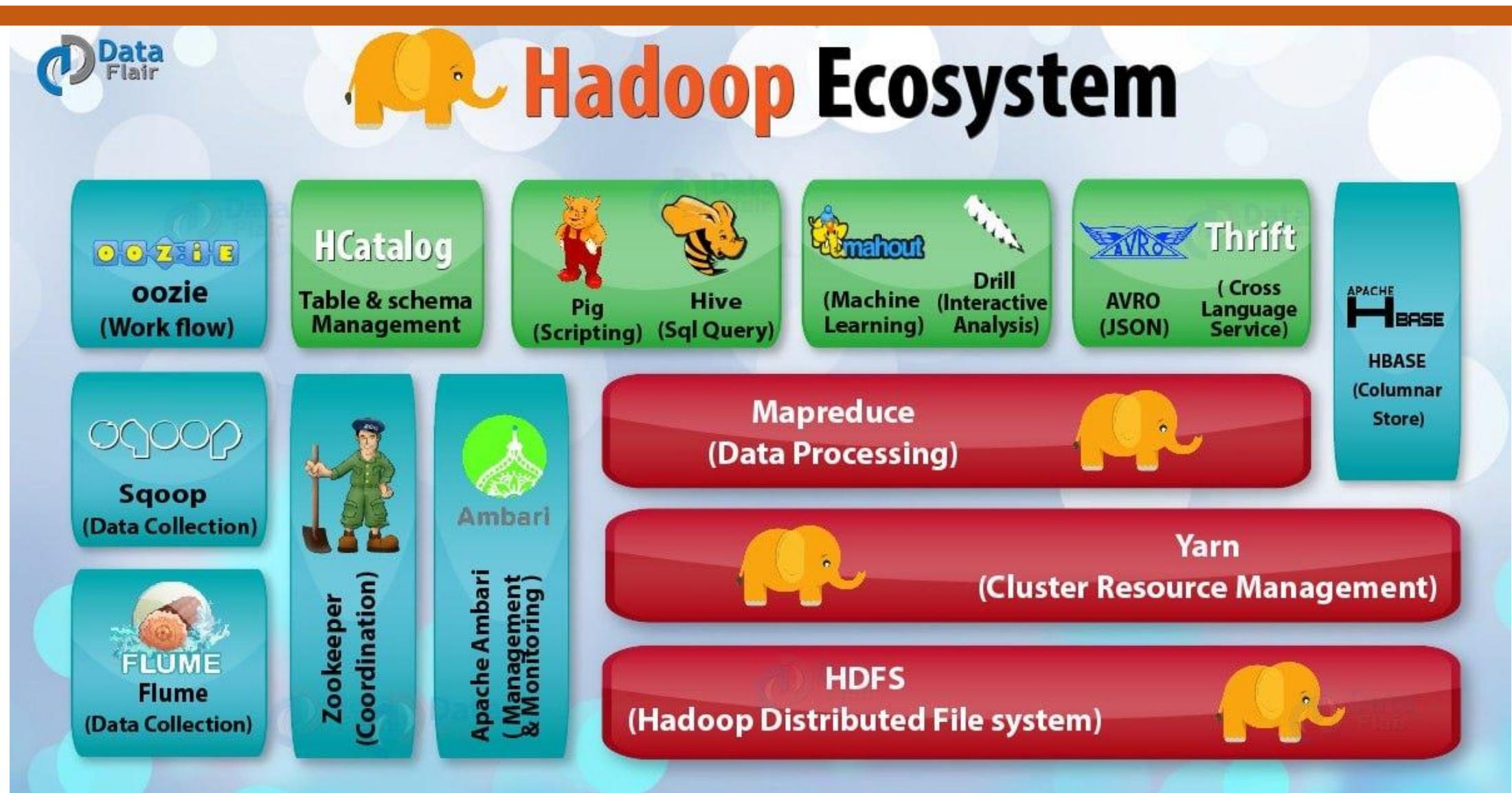
- your local machine can't handle the computation
- the computation can be parallelized
- you can't use Spark
- you have no other choice

Summary

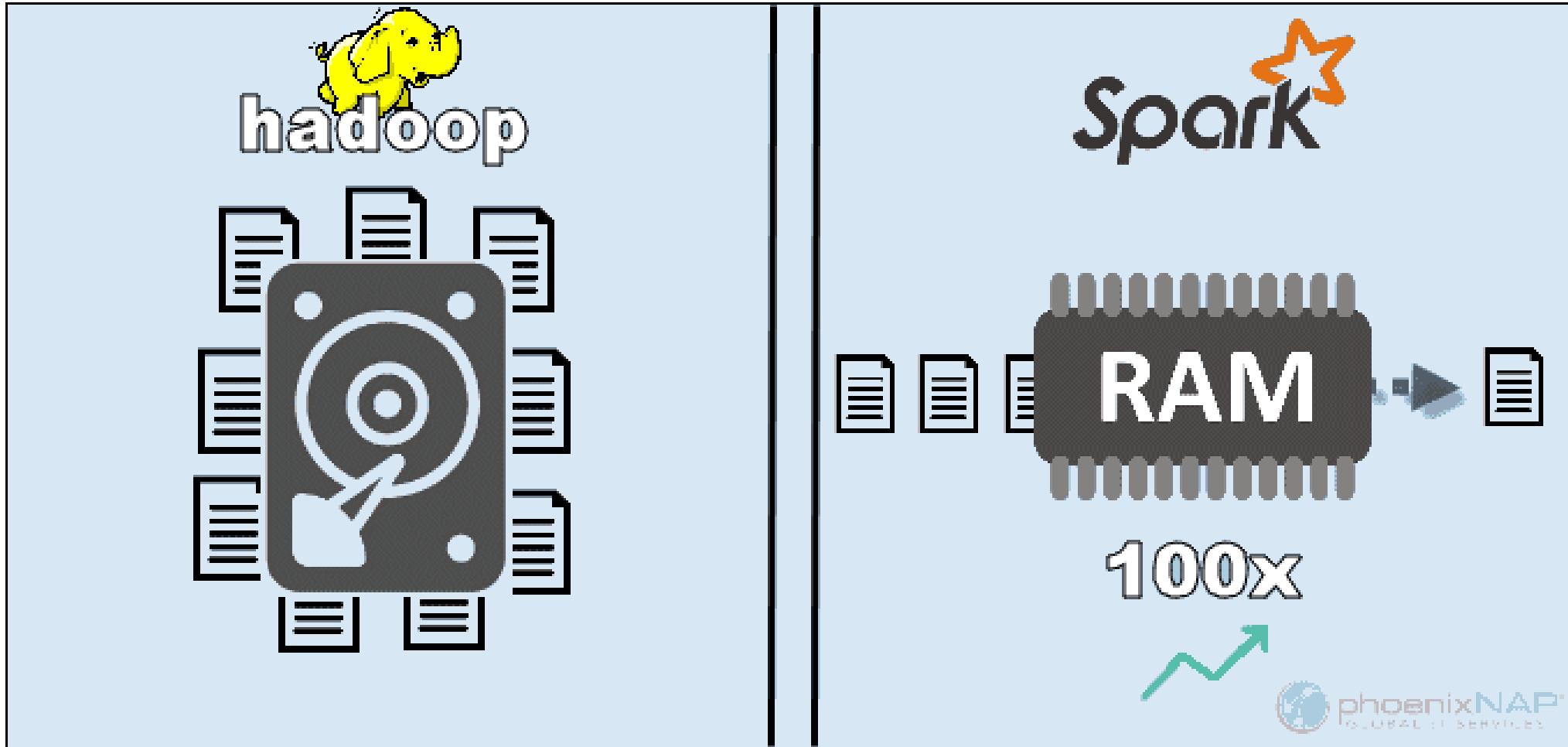
What are the limits of Hadoop ?

- The shuffle part is really expensive in terms of computations
- The files should be transferred to HDFS, and this is expensive
- The community around Hadoop is not active anymore
- Spark is almost systematically chosen

Summary



Summary

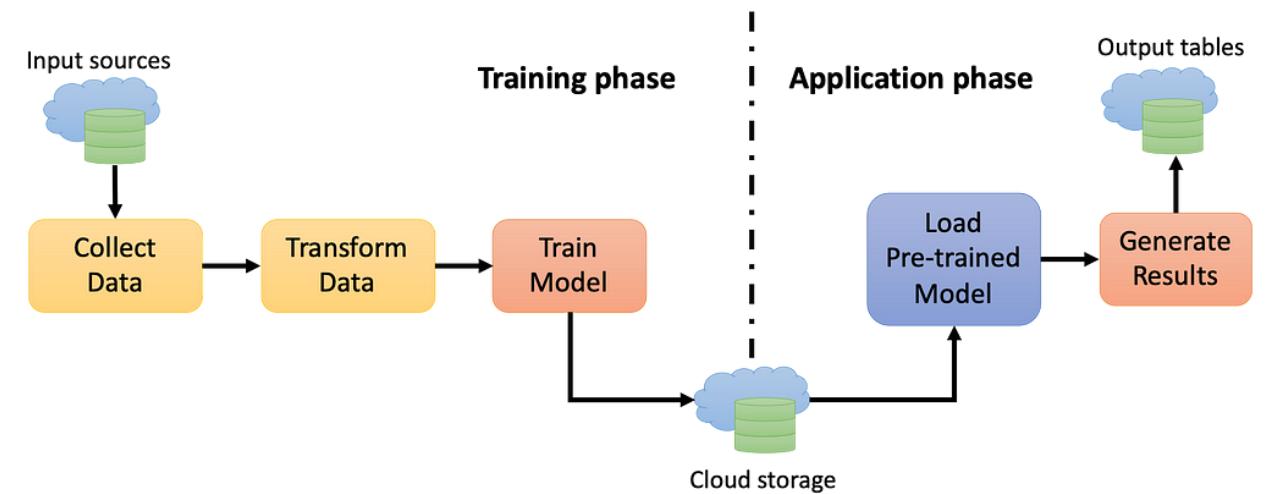




NEXT TOPIC

Big Data 2

(Zero To Mastery: ML Models Using PySpark in Cloud Platform)



Vinh Dinh Nguyen
PhD in Computer Science