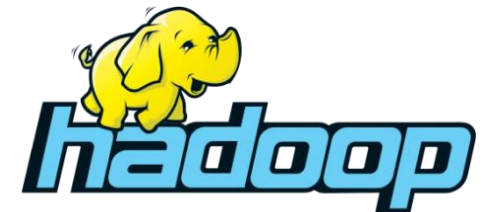




Apache Hadoop

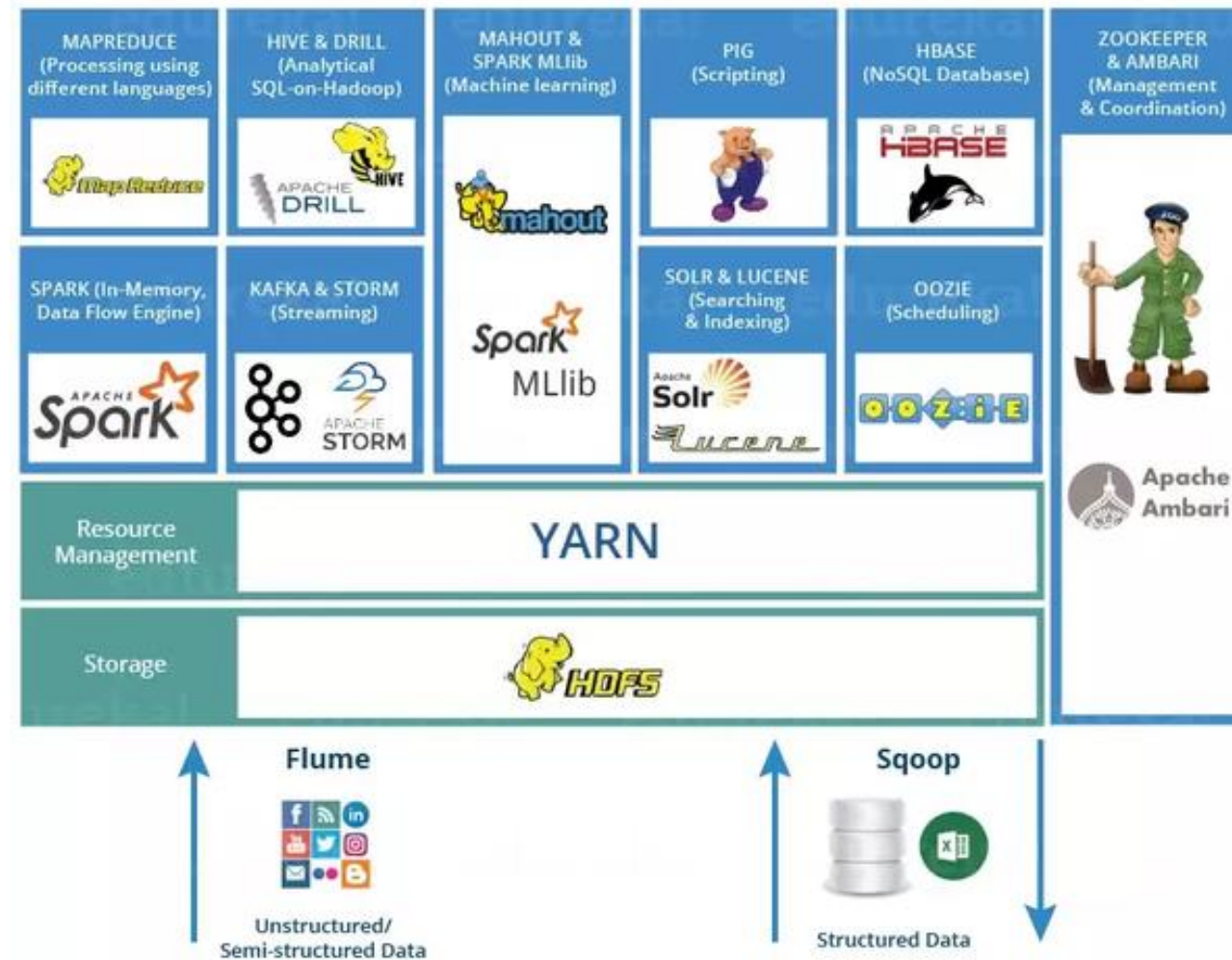
- ▶ Hadoop is an open source framework that facilitates using a network of many computers to solve problems involving massive amounts of data and computation
- ▶ Hadoop consists of two components:
 - ▶ Hadoop Distributed File System (**HDFS**)
 - ▶ Distributed computing engine that lets you implement and run programs as **MapReduce** jobs
- ▶ The Hadoop infrastructure takes care of all complex aspects of distributed processing: parallelization, scheduling, resource management, inter-machine communication, handling software and hardware failures
- ▶ Thanks to this clean abstraction, implementing distributed applications that process terabytes of data on hundreds of machines has never been so easy



Apache Hadoop

- ▶ A wide variety of companies and organizations use Hadoop for both research and production
- ▶ For example, in Yahoo More than 40,000 computers running Hadoop
 - ▶ The biggest cluster contains 4500 nodes
 - ▶ Used to support research for Ad Systems and Web Search

Hadoop EcoSystem

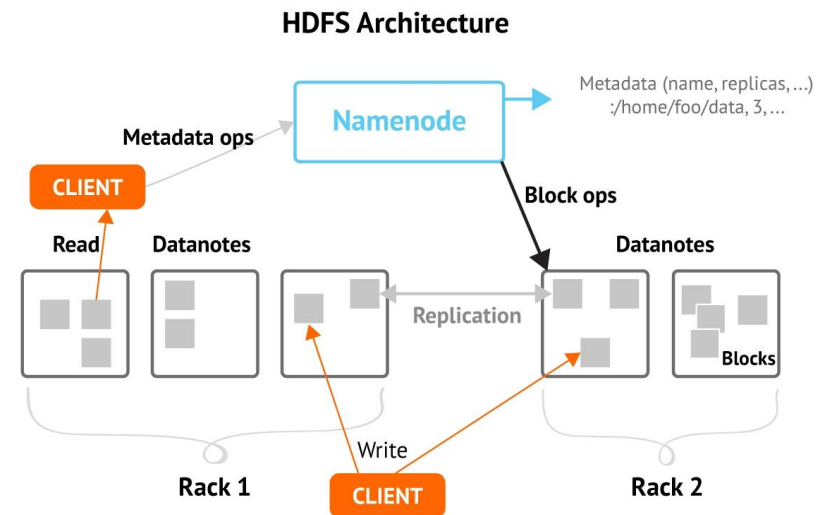


Distributed File Systems

- ▶ A distributed file system is similar to a normal file system, except that it runs on multiple servers at once
- ▶ Because it's a file system, you can do almost all the same things you'd do on a normal file system, such as storing, reading, and deleting files and adding security to files
- ▶ Distributed file systems have significant advantages:
 - ▶ They can store files larger than any one computer disk
 - ▶ Files get automatically replicated across multiple servers for redundancy or parallel operations while hiding the complexity of doing so from the user
 - ▶ The system can scale horizontally (by adding small servers to the network) instead of vertically (moving everything to a stronger server with more storage and better CPUs)
- ▶ The best-known distributed file system is the **Hadoop File System (HDFS)**
 - ▶ It is an open source implementation of the Google File System (GFS)

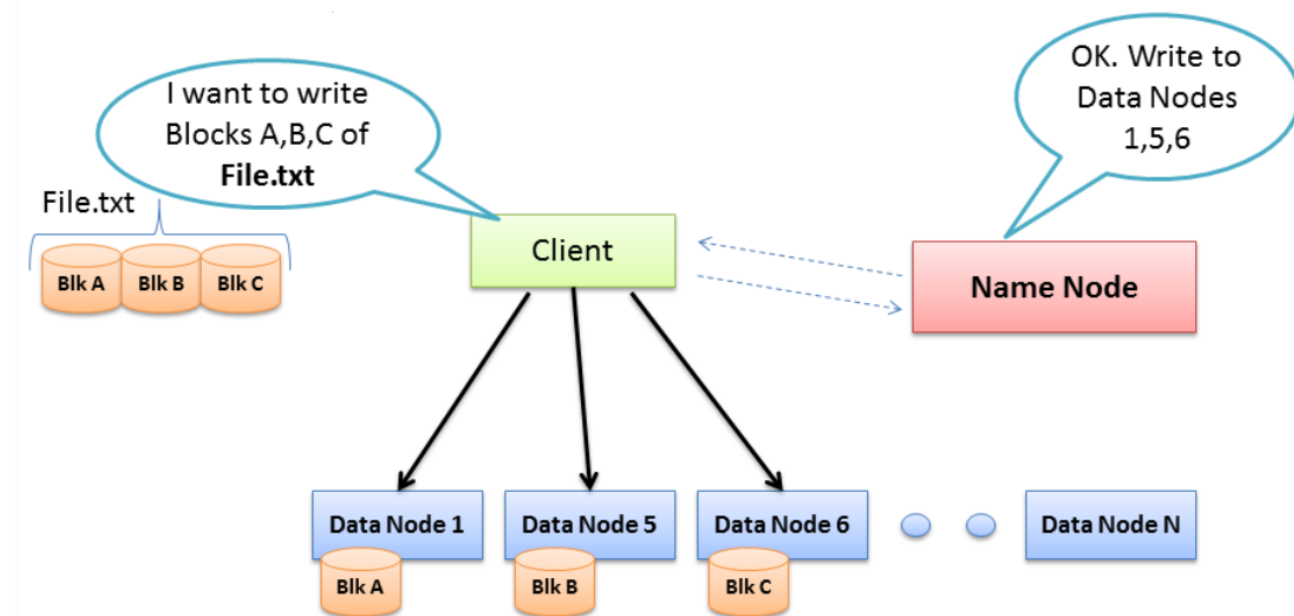
HDFS Architecture

- ▶ The master (**NameNode**) manages the file system namespace operations like opening, closing, and renaming files and directories and determines the mapping of blocks to DataNodes along with regulating access to files by clients
- ▶ The slaves (**DataNodes**) are responsible for serving read and write requests from the file system's clients along with perform block creation, deletion, and replication upon instruction from the Master (NameNode)



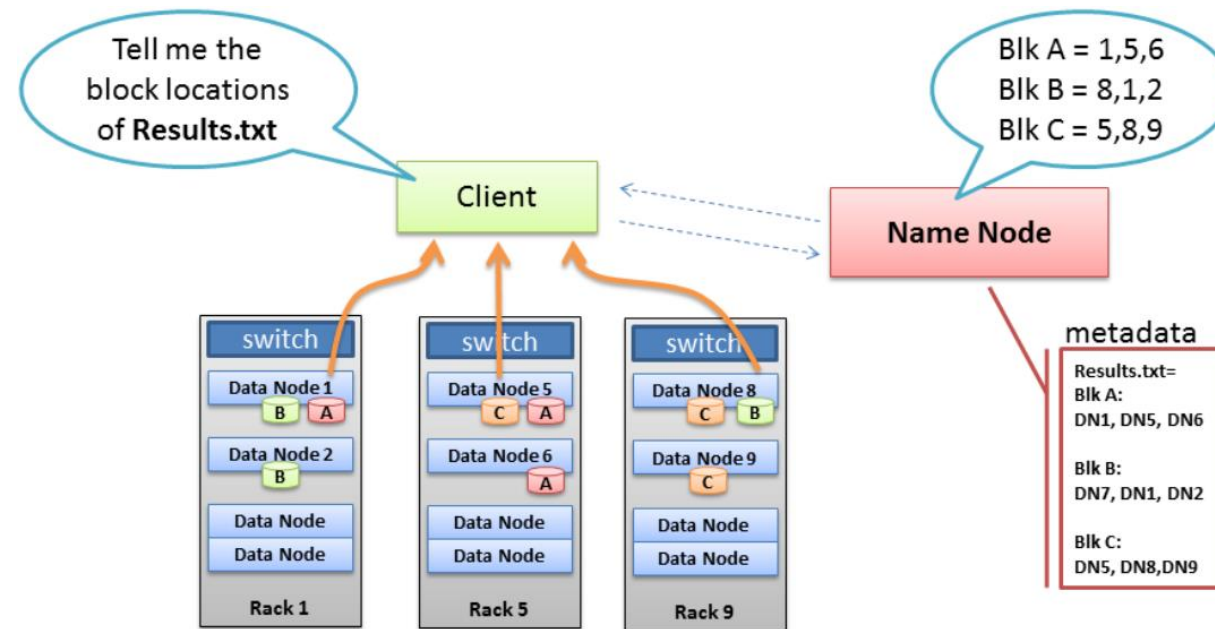
Writing a File to HDFS

- ▶ When a client wants to write a file to HDFS:
 - ▶ The client asks the NameNode for a list of data nodes that will store the file blocks
 - ▶ The client writes the blocks to the DataNodes
 - ▶ The DataNodes replicate the block
- ▶ The standard setting for Hadoop is to have 3 copies of each block in the cluster



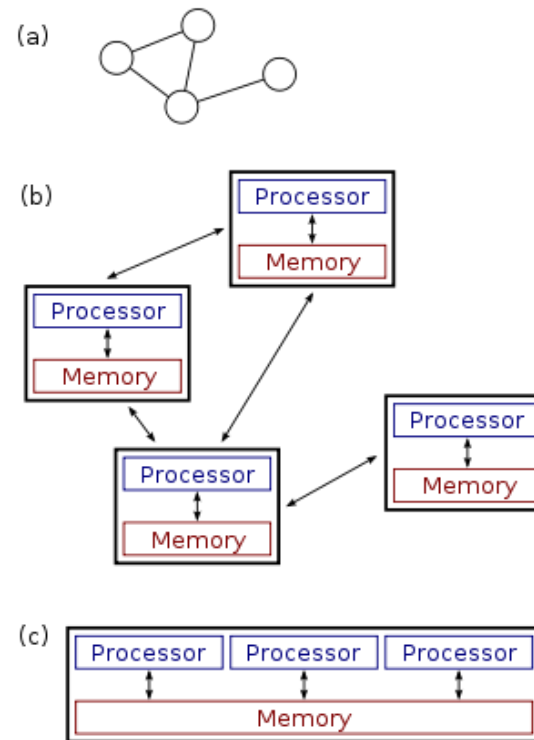
Reading a File from HDFS

- ▶ When a Client wants to retrieve a file from HDFS:
 - ▶ It consults the NameNode and asks for the block locations of the file
 - ▶ The NameNode returns a list of each DataNode holding a block, for each block
 - ▶ The client picks a DataNode from each block list and reads one block at a time with TCP on port 50010 (the default port number for the Data Node daemon)



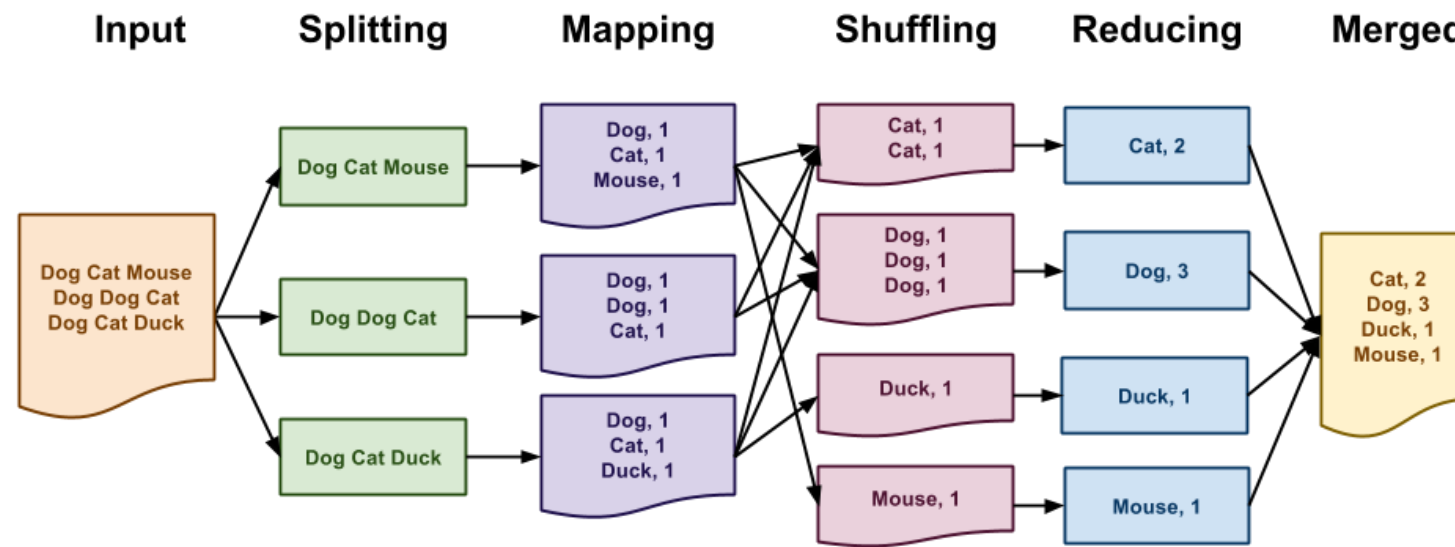
Distributed Computing

- ▶ In **distributed computing**, a problem is divided into many tasks, each of which is solved by a single computer
 - ▶ The computers communicate with each other via message passing



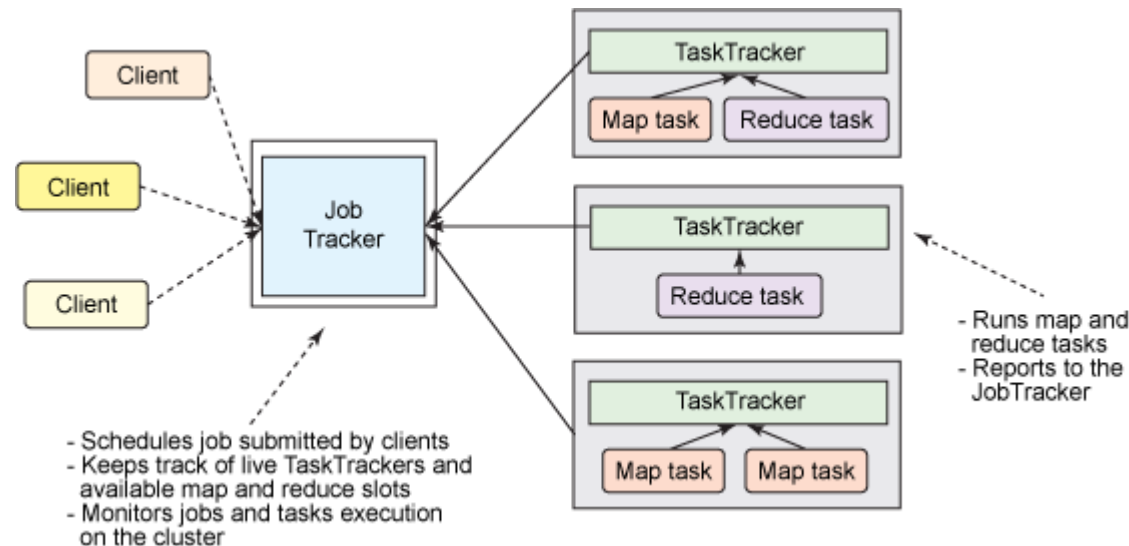
MapReduce

- ▶ MapReduce is a simple functional programming operation and it can be applied, in parallel, to gigabytes or terabytes of data
- ▶ *Map tasks* invoke map functions over subsets of input data
- ▶ After they are done, *reduce tasks* start calling reduce functions on the intermediate data, generated by map functions, to produce the final output



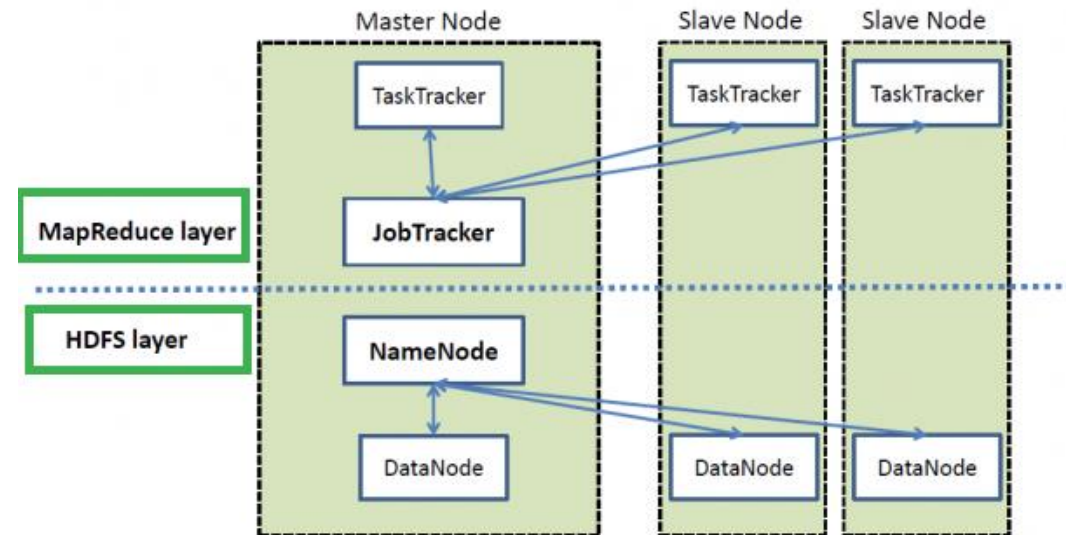
Hadoop MapReduce

- ▶ In the classical Hadoop MapReduce framework (MRv1), the job execution is controlled by two types of processes:
 - ▶ A single master process called **JobTracker**, which coordinates all jobs running on the cluster and assigns map and reduce tasks to run on the TaskTrackers
 - ▶ A number of subordinate processes called **TaskTrackers**, which run assigned tasks and periodically report the progress to the JobTracker



Hadoop MapReduce

- ▶ TaskTracker instances are typically deployed on the same servers that host DataNode instances, so that MapReduce operations are performed close to the data

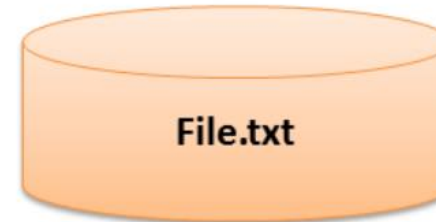


Typical Workflow

- ▶ Load data into the cluster (HDFS writes)
- ▶ Analyze the data (Map Reduce)
- ▶ Store results in the cluster (HDFS writes)
- ▶ Read the results from the cluster (HDFS reads)

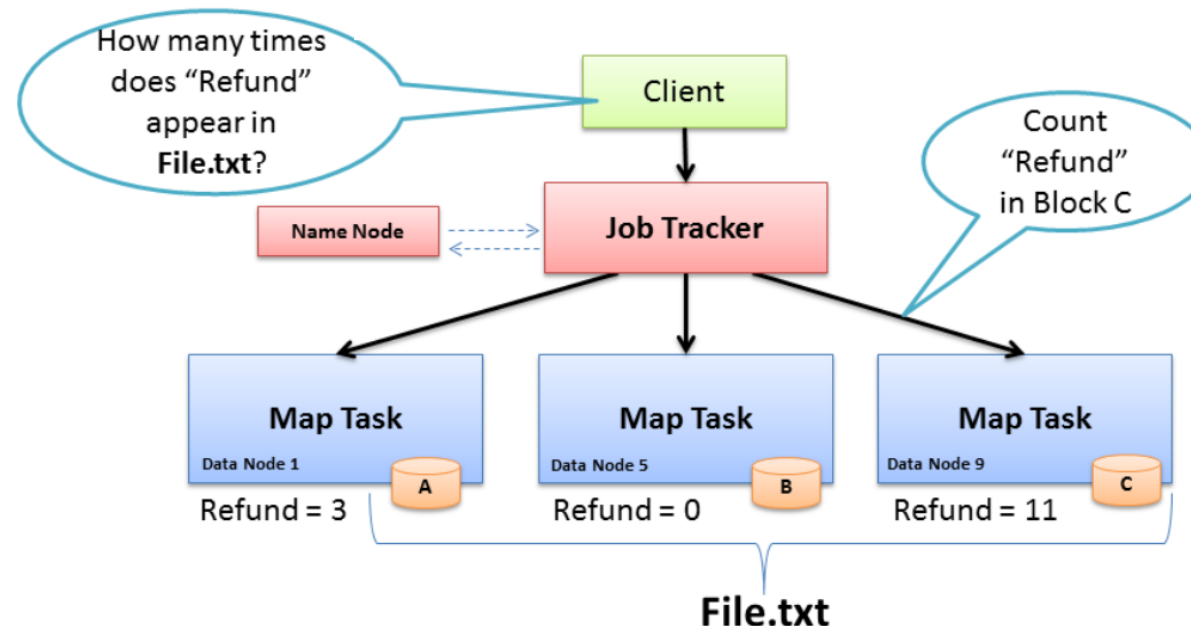
How many times did our customers type the word **"Refund"** into emails sent to customer service?

Huge file containing all emails sent
to customer service



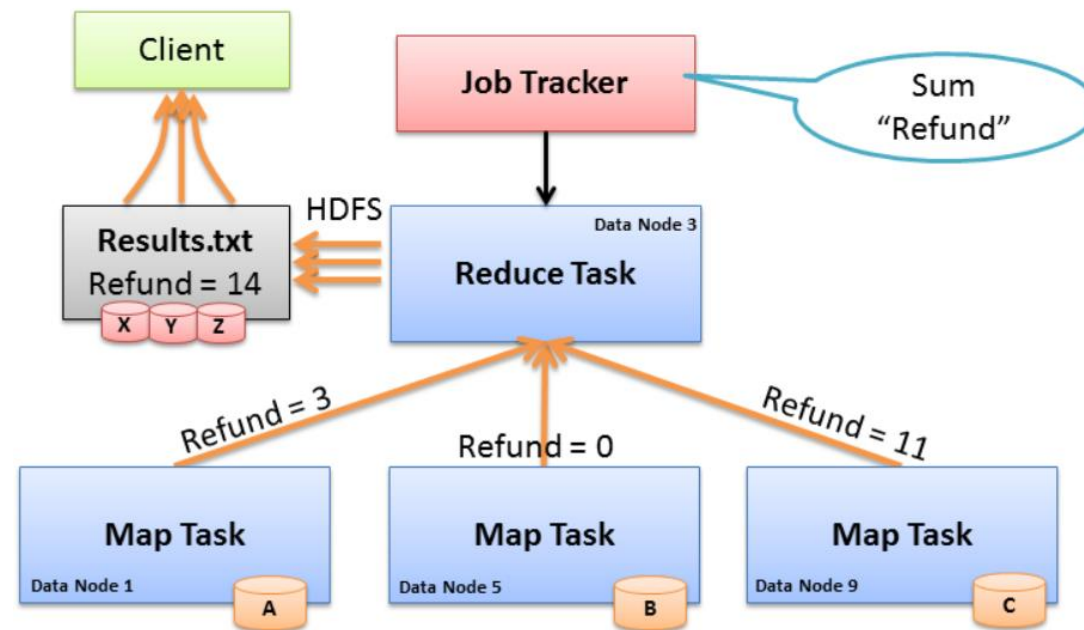
Data Processing: Map

- ▶ The Client machine submits the Map Reduce job to the Job Tracker
- ▶ The Job Tracker consults the Name Node to learn which Data Nodes have blocks of the file
- ▶ The Job Tracker then provides the Task Trackers running on those nodes with the Java code required to execute the Map computation on their local data
- ▶ Each node stores the result of its local computation in temporary local storage



Data Processing: Reduce

- ▶ The Job Tracker starts a Reduce task on any one of the nodes in the cluster
- ▶ The Reduce task grabs the intermediate data from all of the completed Map tasks.
- ▶ The Reducer summarizes the results from all the map tasks (e.g., sums up the total occurrences of the word “Refund”) and writes the final result to HDFS



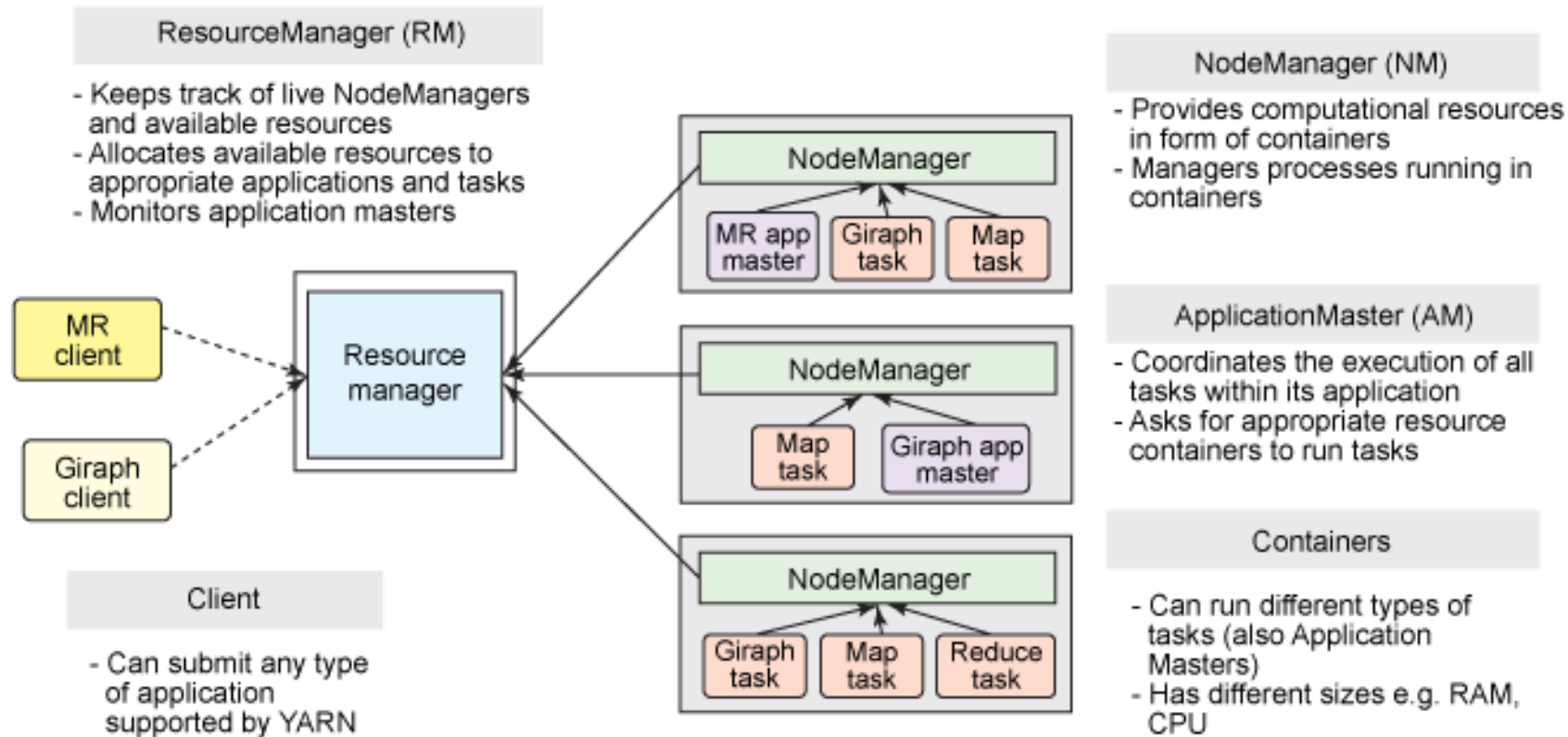
Limitations of Classical MapReduce

- ▶ The large Hadoop clusters revealed a limitation involving a scalability bottleneck caused by having a single JobTracker
- ▶ A single JobTracker had to constantly keep track of thousands of TaskTrackers, hundreds of jobs, and tens of thousands of map and reduce tasks
- ▶ According to Yahoo!, the practical limits of such a design are reached with a cluster of 5,000 nodes and 40,000 tasks running concurrently
- ▶ Due to this limitation, smaller and less-powerful clusters had to be created and maintained
- ▶ In addition, Hadoop was designed to run MapReduce jobs only
- ▶ With the advent of alternative programming models (such as graph processing), there was a need to support programming paradigms that could run on the same cluster and share resources in an efficient and fair manner

Hadoop YARN

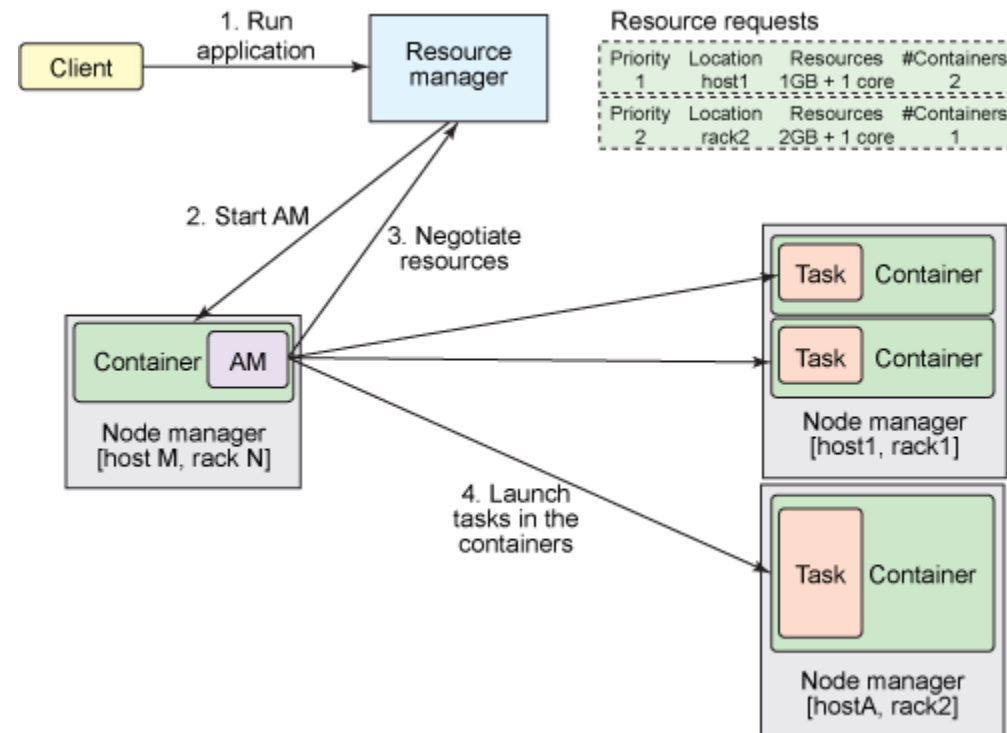
- ▶ YARN (Yet Another Resource Negotiator) is the resource management and job scheduling technology used in Hadoop from version 2.0
- ▶ YARN offers clear advantages in scalability, efficiency, and flexibility compared to the classical MapReduce engine in the first version of Hadoop
- ▶ The YARN-based architecture is not constrained to MapReduce
- ▶ The following name changes give a bit of insight into the design of YARN:
 - ▶ ResourceManager instead of a cluster manager
 - ▶ ApplicationMaster instead of a dedicated and short-lived JobTracker
 - ▶ NodeManager instead of TaskTracker
 - ▶ A distributed application instead of a MapReduce job

YARN Architecture



Application Submission in YARN

- ▶ The following diagram shows how the ResourceManager, ApplicationMaster, NodeManagers, and containers interact together when an application is submitted to a YARN cluster



Hadoop Deployment Methods

- ▶ **Standalone:** In this mode, there are no daemons running, everything runs as in a single JVM. This mode is suitable for running the MapReduce program during development, as it is easy to test and debug.
- ▶ **Pseudo-distributed:** The Hadoop daemon process runs on a local machine simulating a cluster on a small scale.
- ▶ **Fully distributed:** Hadoop runs on a cluster of machines providing a production environment.
- ▶ Client machines have Hadoop installed with all the cluster settings, but are neither a Master or a Slave

Install Java

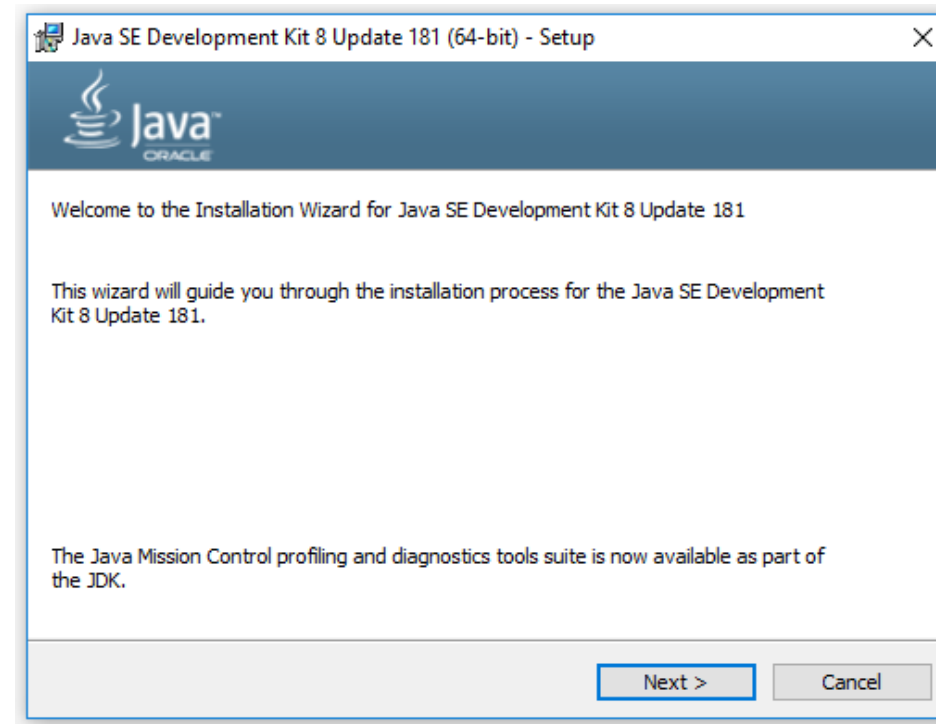
- ▶ Running Hadoop requires Java
- ▶ Download JDK (Java Development Kit) 8 from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- ▶ Note: JDK 10 is still not supported by current version of Hadoop

Java SE 8u181
Java SE 8u181 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
[Learn more](#) ▶

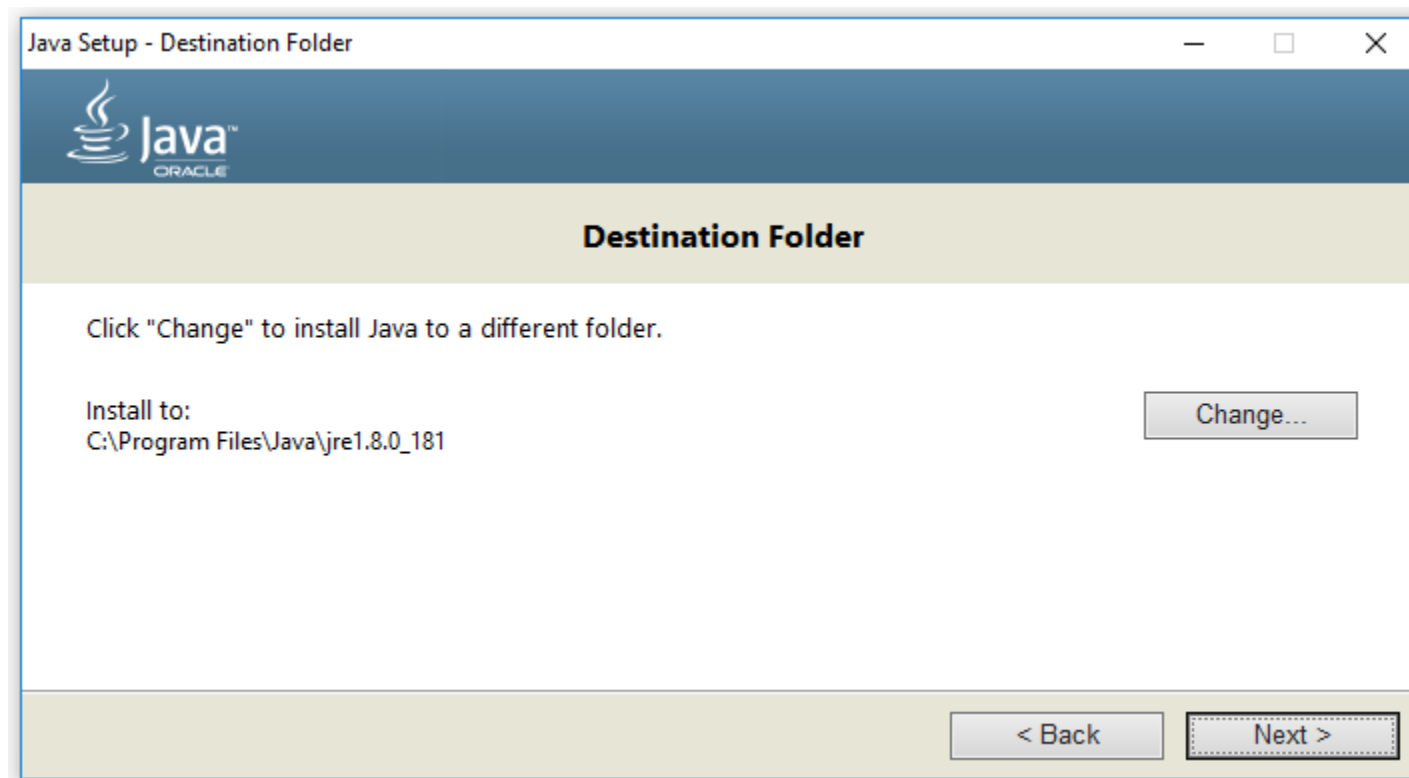
<ul style="list-style-type: none">▪ Installation Instructions▪ Release Notes▪ Oracle License▪ Java SE Licensing Information User Manual<ul style="list-style-type: none">▪ Includes Third Party Licenses▪ Certified System Configurations▪ Readme Files<ul style="list-style-type: none">▪ JDK ReadMe▪ JRE ReadMe	JDK DOWNLOAD ⬇
	Server JRE DOWNLOAD ⬇
	JRE DOWNLOAD ⬇

Install Java

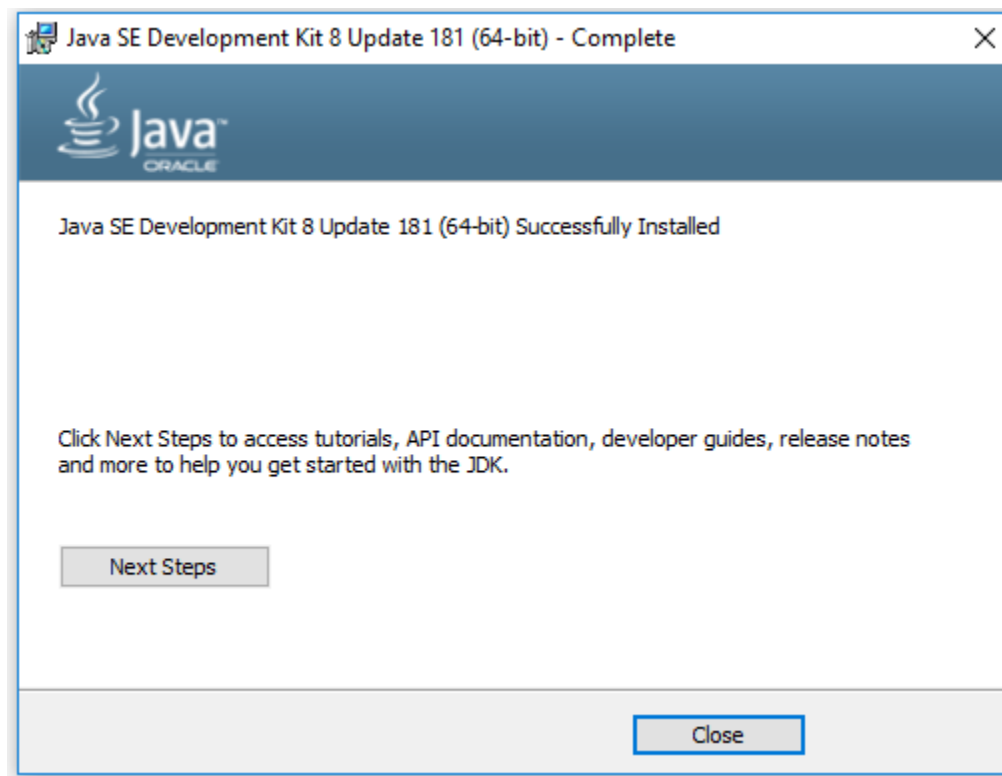
- ▶ Double-click the file `jdk-8u181-windows-x64.exe` to start installation



Install Java

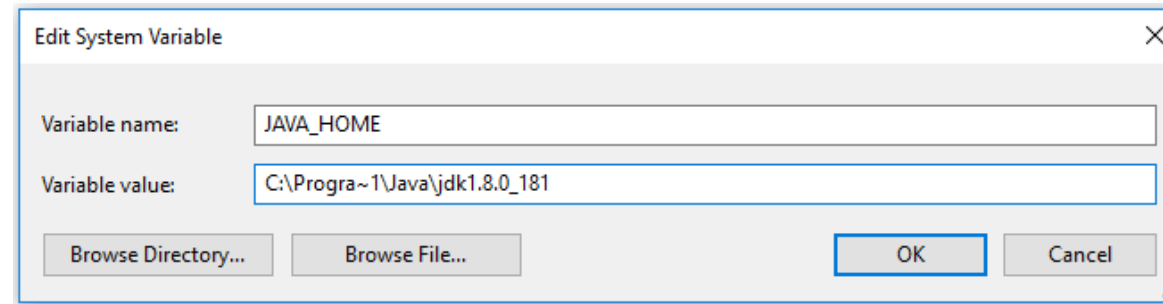


Install Java



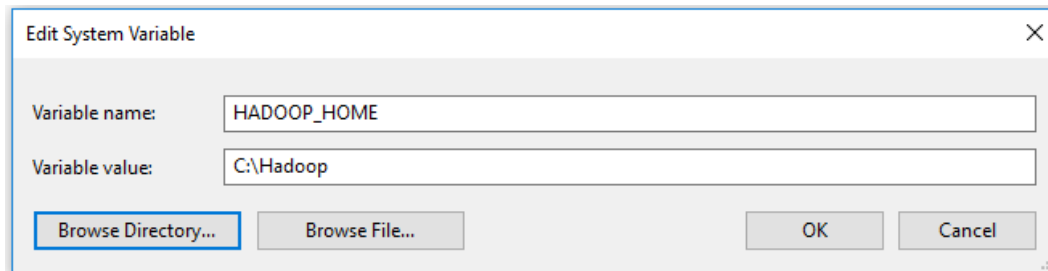
Install Java

- ▶ Set the JAVA_HOME environment variable
 - ▶ In the **Variable Name** field, enter JAVA_HOME
 - ▶ In the **Variable Value** field, enter your JDK installation path
 - ▶ Write Progra~1 instead of 'Program Files' (hadoop has some issues with spaces)



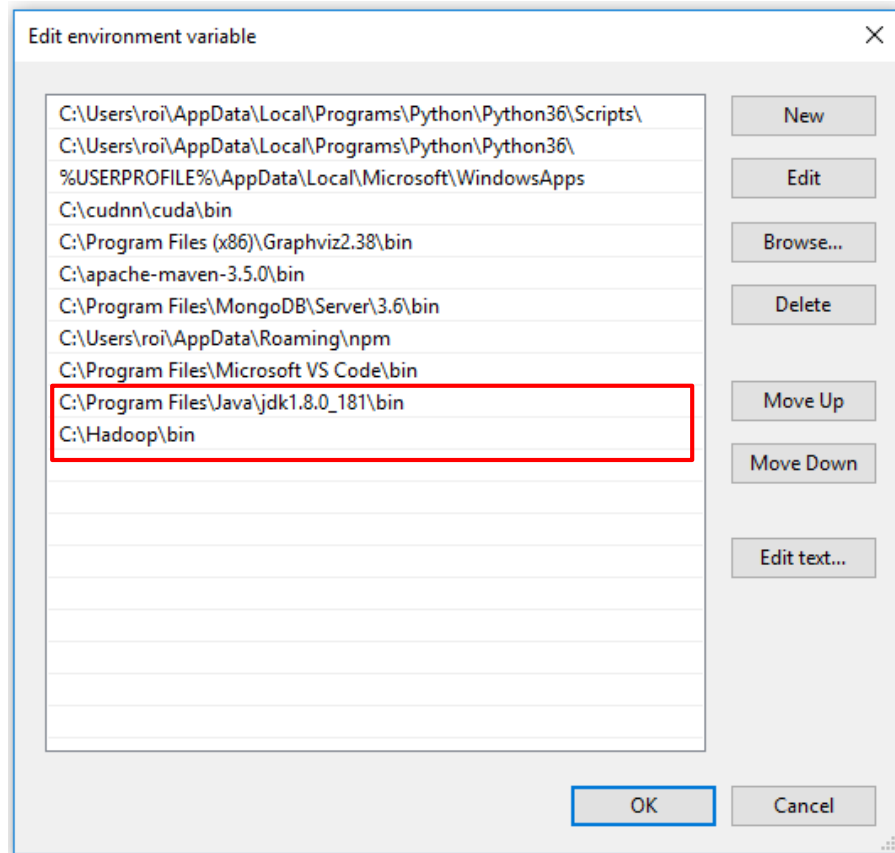
Install Hadoop

- ▶ Download the latest Hadoop binary from <http://hadoop.apache.org/releases.html>
- ▶ Extract the downloaded file (hadoop-3.0.3.tar.gz) into the folder C:\Hadoop
- ▶ You might get some warning during the extraction process, you can ignore them
- ▶ Set the HADOOP_HOME environment variable to C:\Hadoop



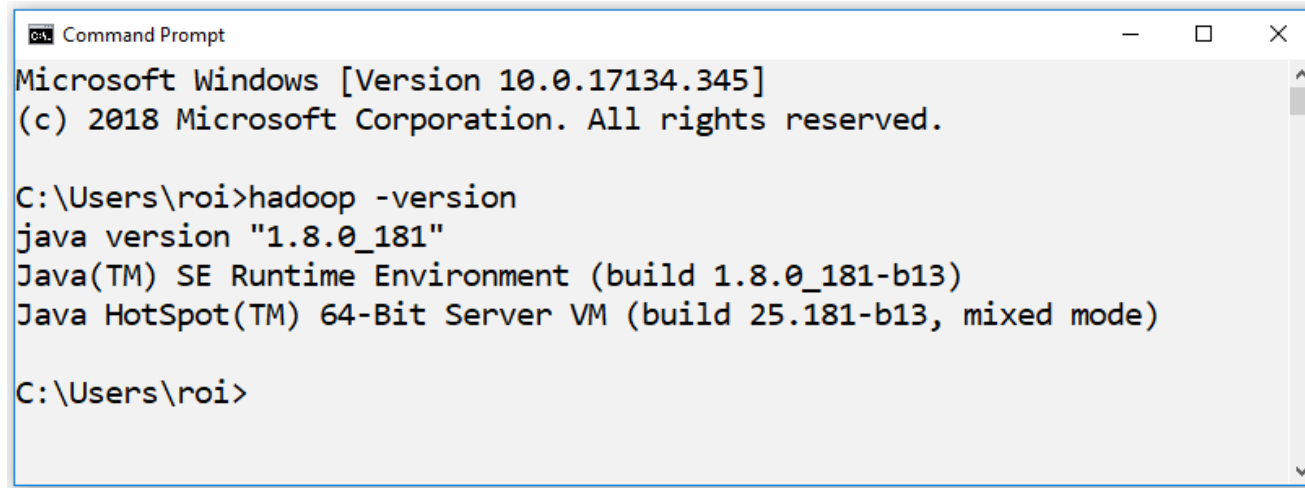
Install Hadoop

- ▶ Add %JAVA_HOME%/bin and %HADOOP_HOME%/bin into the Path variable:



Verify Setup

- ▶ You should be able to verify your settings via the following command:

A screenshot of a Windows Command Prompt window. The title bar reads 'Command Prompt'. The text inside shows the Windows version and copyright information, followed by the command 'hadoop -version' and its output: 'java version "1.8.0_181"', 'Java(TM) SE Runtime Environment (build 1.8.0_181-b13)', and 'Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)'. The prompt is currently at 'C:\Users\roi>'.

```
Command Prompt
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\roi>hadoop -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

C:\Users\roi>
```

Install WinUtils

- ▶ WinUtils are Windows binaries for Hadoop versions
- ▶ Download winutils from <https://github.com/steveloughran/winutils>

Windows binaries for Hadoop versions (built from the git commit ID used for the ASF release)

24 commits

1 branch

3 releases

2 contributors

Apache-2.0

Branch: master


New pull request

Create new file


Upload files

Find file

Clone or download


 steveloughran committed on Dec 20, 2017 improving script for adding the artifacts

Latest commit 19a39b1 on Dec 20, 2017

 hadoop-2.6.0/bin


Add Hadoop-2.6.0/HDP-2.2 windows binaries

3 years ago

 hadoop-2.6.3/bin

add gpg2 signatures

3 years ago

 hadoop-2.6.4

add 2.6.4 and 2.7.1 windows binaries

3 years ago

- ▶ Extract the file winutils-master.zip
- ▶ Choose the subfolder that corresponds to the version of Hadoop you downloaded
- ▶ Copy the files from this subfolder\bin to C:\Hadoop\bin
 - ▶ e.g., copy winutils-master\hadoop-3.0.0\bin to C:\Hadoop\bin
 - ▶ Choose replace the files in the destination

Hadoop Configuration

- ▶ Before running Hadoop, we need to configure some files in *C:\Hadoop\etc\hadoop*
- ▶ First, edit the file *C:\Hadoop\etc\hadoop\core-site.xml*
 - ▶ This file contains the name of the default file system
 - ▶ All Hadoop services and clients use this file to locate the namenode
- ▶ Add the following configuration key to the file:

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://0.0.0.0:19000</value>
  </property>
</configuration>
```

Hadoop Configuration

- ▶ Now edit the file *hdfs-site.xml*
 - ▶ This file contains the HTTP addresses for namenode and datanode
- ▶ Add the following configuration keys to the file:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>file:///C:/Hadoop/dfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>file:///C:/Hadoop/dfs/data</value>
  </property>
</configuration>
```

- ▶ Create the directories C:\Hadoop\dfs\name and C:\Hadoop\dfs\data

Hadoop Configuration

- ▶ Edit *mapred-site.xml*
 - ▶ This file contains the framework for executing MapReduce jobs
- ▶ Add the following, replacing *%USERNAME%* with your Windows username:

```
<configuration>
  <property>
    <name>mapreduce.job.user.name</name>
    <value>%USERNAME%</value>
  </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>yarn.apps.stagingDir</name>
    <value>/user/%USERNAME%/staging</value>
  </property>
  <property>
    <name>mapreduce.jobtracker.address</name>
    <value>local</value>
  </property>
</configuration>
```


Hadoop Configuration

- Edit *yarn-site.xml* and add the following entries:

```
<configuration>
  <property>
    <name>yarn.server.resourcemanager.address</name>
    <value>0.0.0.0:8020</value>
  </property>
  <property>
    <name>yarn.server.resourcemanager.application.expiry.interval</name>
    <value>60000</value>
  </property>
  <property>
    <name>yarn.server.nodemanager.address</name>
    <value>0.0.0.0:45454</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>
  <property>
    <name>yarn.server.nodemanager.remote-app-log-dir</name>
    <value>/app-logs</value>
  </property>
  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/dep/logs/userlogs</value>
  </property>
```

```
  <property>
    <name>yarn.server.mapreduce-appmanager.attempt-
listener.bindAddress</name>
    <value>0.0.0.0</value>
  </property>
  <property>
    <name>yarn.server.mapreduce-appmanager.client-
service.bindAddress</name>
    <value>0.0.0.0</value>
  </property>
  <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>
  <property>
    <name>yarn.log-aggregation.retain-seconds</name>
    <value>-1</value>
  </property>
  <property>
    <name>yarn.application.classpath</name>
    <value>%HADOOP_CONF_DIR%,%HADOOP_COMMON_HOME%/share/hadoop/common/*,%HADOOP_COMMON_HOME%/share/hadoop/common/lib/*,%HADOOP_HDFS_HOME%/share/hadoop/hdfs/*,%HADOOP_HDFS_HOME%/share/hadoop/hdfs/lib/*,%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/*,%HADOOP_MAPRED_HOME%/share/hadoop/mapreduce/lib/*,%HADOOP_YARN_HOME%/share/hadoop/yarn/*,%HADOOP_YARN_HOME%/share/hadoop/yarn/lib/*</value>
  </property>
</configuration>
```

Hadoop Configuration

- ▶ Finally, change the file `hadoop-env.cmd` to add the following lines at the end of file:

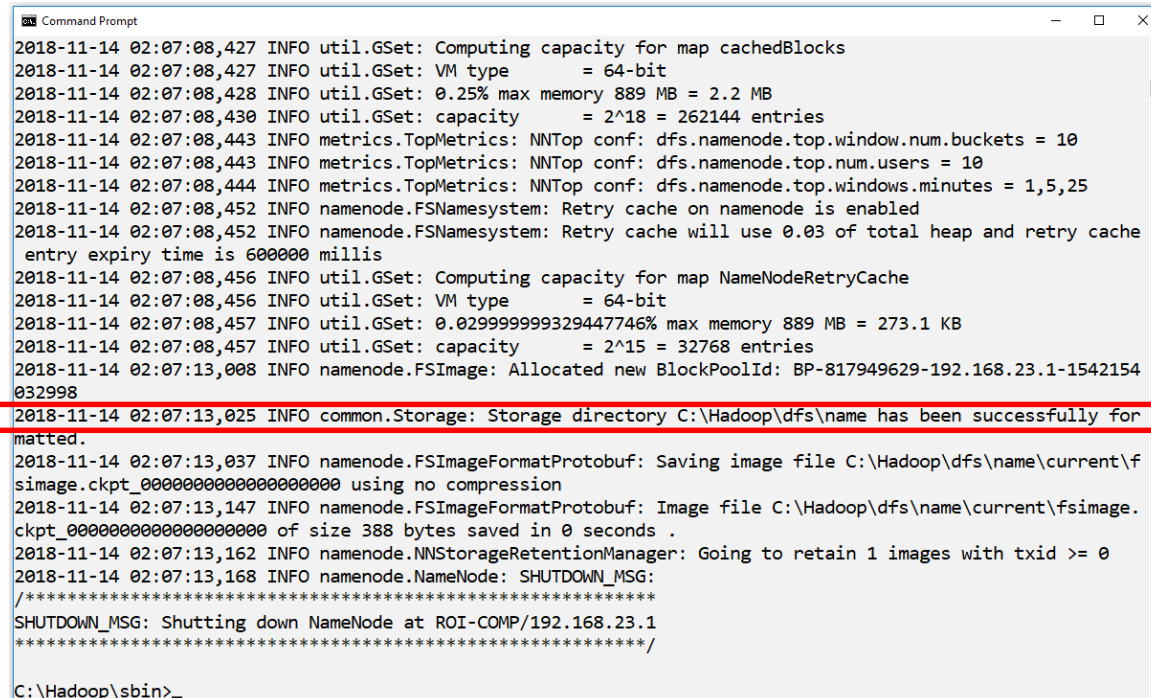
```
set HADOOP_PREFIX=%HADOOP_HOME%  
set HADOOP_CONF_DIR=%HADOOP_PREFIX%\etc\hadoop  
set YARN_CONF_DIR=%HADOOP_CONF_DIR%  
set PATH=%PATH%;%HADOOP_PREFIX%\bin
```

Format HDFS File System

- ▶ Open command prompt in administrator mode
- ▶ Run the following command to format the file system:

```
hdfs namenode -format
```

- ▶ The output should contain log messages indicating the success of the format:



```
2018-11-14 02:07:08,427 INFO util.GSet: Computing capacity for map cachedBlocks
2018-11-14 02:07:08,427 INFO util.GSet: VM type = 64-bit
2018-11-14 02:07:08,428 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
2018-11-14 02:07:08,430 INFO util.GSet: capacity = 2^18 = 262144 entries
2018-11-14 02:07:08,443 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2018-11-14 02:07:08,443 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2018-11-14 02:07:08,444 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2018-11-14 02:07:08,452 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2018-11-14 02:07:08,452 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache
entry expiry time is 600000 millis
2018-11-14 02:07:08,456 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2018-11-14 02:07:08,456 INFO util.GSet: VM type = 64-bit
2018-11-14 02:07:08,457 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
2018-11-14 02:07:08,457 INFO util.GSet: capacity = 2^15 = 32768 entries
2018-11-14 02:07:13,008 INFO namenode.FSImage: Allocated new BlockPoolId: BP-817949629-192.168.23.1-1542154
032998
2018-11-14 02:07:13,025 INFO common.Storage: Storage directory C:\Hadoop\dfs\name has been successfully for
matted.
2018-11-14 02:07:13,037 INFO namenode.FSImageFormatProtobuf: Saving image file C:\Hadoop\dfs\name\current\fs
image.ckpt_000000000000000000 using no compression
2018-11-14 02:07:13,147 INFO namenode.FSImageFormatProtobuf: Image file C:\Hadoop\dfs\name\current\fsimage.
ckpt_000000000000000000 of size 388 bytes saved in 0 seconds .
2018-11-14 02:07:13,162 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2018-11-14 02:07:13,168 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at ROI-COMP/192.168.23.1
*****/
C:\Hadoop\sbin>
```

Startup Scripts

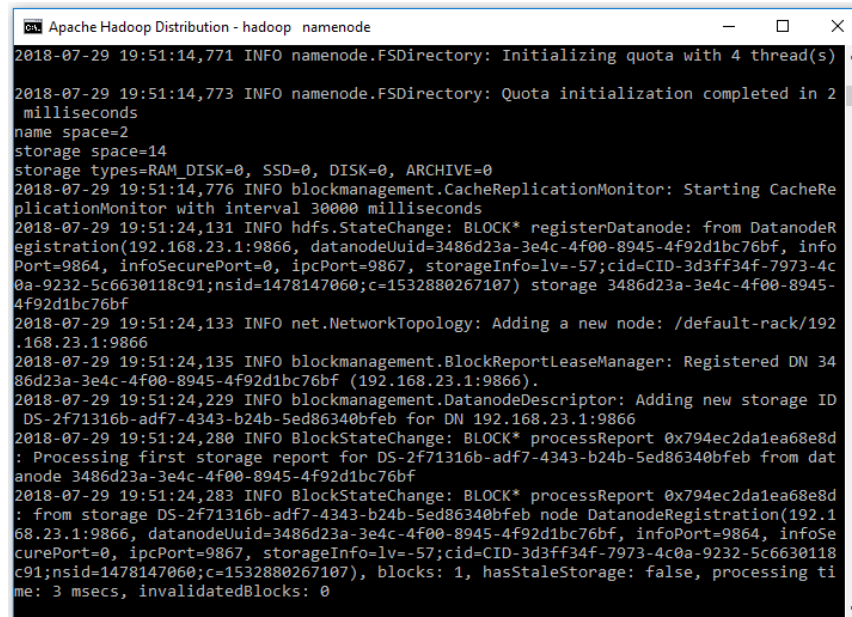
- ▶ The C:\Hadoop\sbin directory contains some scripts used to launch Hadoop DFS and Hadoop Map/Reduce daemons:
 - ▶ **start-dfs.sh** - Starts the Hadoop DFS daemons, the namenode and datanodes. Use this before start-mapred.sh
 - ▶ **stop-dfs.sh** - Stops the Hadoop DFS daemons.
 - ▶ **start-mapred.sh** - Starts the Hadoop Map/Reduce daemons, the jobtracker and tasktrackers.
 - ▶ **stop-mapred.sh** - Stops the Hadoop Map/Reduce daemons.
 - ▶ **start-all.sh** - Starts all Hadoop daemons, the namenode, datanodes, the jobtracker and tasktrackers. Deprecated; use start-dfs.sh then start-mapred.sh
 - ▶ **stop-all.sh** - Stops all Hadoop daemons. Deprecated; use stop-mapred.sh then stop-dfs.sh

Starting HDFS Daemons

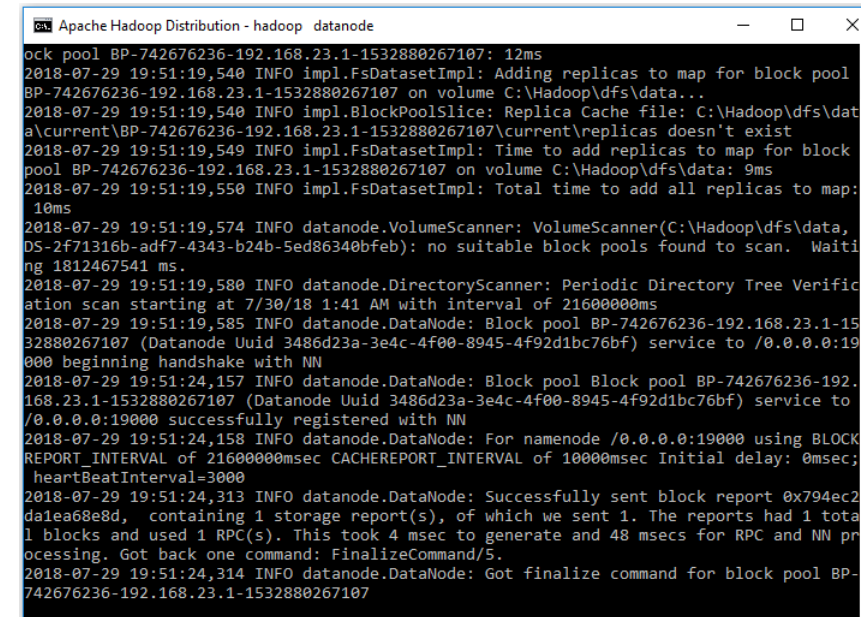
- ▶ Start namenode and datanode by running the following command:

```
cd C:\Hadoop\sbin  
C:\Hadoop\sbin>start-dfs.cmd
```

- ▶ Two command prompts named namenode and datanode will open:



```
Apache Hadoop Distribution - hadoop namenode  
2018-07-29 19:51:14,771 INFO namenode.FSDirectory: Initializing quota with 4 thread(s)  
2018-07-29 19:51:14,773 INFO namenode.FSDirectory: Quota initialization completed in 2 milliseconds  
name space=2  
storage space=14  
storage types=RAM_DISK=0, SSD=0, DISK=0, ARCHIVE=0  
2018-07-29 19:51:14,776 INFO blockmanagement.CacheReplicationMonitor: Starting CacheReplicationMonitor with interval 30000 milliseconds  
2018-07-29 19:51:24,131 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(192.168.23.1:9866, datanodeUuid=3486d23a-3e4c-4f00-8945-4f92d1bc76bf, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-3d3ff34f-7973-4c0a-9232-5c6630118c91;nsid=1478147060;c=1532880267107) storage 3486d23a-3e4c-4f00-8945-4f92d1bc76bf  
2018-07-29 19:51:24,133 INFO net.NetworkTopology: Adding a new node: /default-rack/192.168.23.1:9866  
2018-07-29 19:51:24,135 INFO blockmanagement.BlockReportLeaseManager: Registered DN 3486d23a-3e4c-4f00-8945-4f92d1bc76bf (192.168.23.1:9866).  
2018-07-29 19:51:24,229 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-2f71316b-adf7-4343-b24b-5ed86340bfeb for DN 192.168.23.1:9866  
2018-07-29 19:51:24,280 INFO BlockStateChange: BLOCK* processReport 0x794ec2da1ea68e8d: Processing first storage report for DS-2f71316b-adf7-4343-b24b-5ed86340bfeb from datanode 3486d23a-3e4c-4f00-8945-4f92d1bc76bf  
2018-07-29 19:51:24,283 INFO BlockStateChange: BLOCK* processReport 0x794ec2da1ea68e8d: from storage DS-2f71316b-adf7-4343-b24b-5ed86340bfeb node DatanodeRegistration(192.168.23.1:9866, datanodeUuid=3486d23a-3e4c-4f00-8945-4f92d1bc76bf, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-3d3ff34f-7973-4c0a-9232-5c6630118c91;nsid=1478147060;c=1532880267107), blocks: 1, hasStaleStorage: false, processing time: 3 msecs, invalidatedBlocks: 0
```



```
Apache Hadoop Distribution - hadoop datanode  
ock pool BP-742676236-192.168.23.1-1532880267107: 12ms  
2018-07-29 19:51:19,540 INFO impl.FsDatasetImpl: Adding replicas to map for block pool BP-742676236-192.168.23.1-1532880267107 on volume C:\Hadoop\dfs\data...  
2018-07-29 19:51:19,540 INFO impl.BlockPoolSlice: Replica Cache file: C:\Hadoop\dfs\data\current\BP-742676236-192.168.23.1-1532880267107\current\replicas doesn't exist  
2018-07-29 19:51:19,549 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-742676236-192.168.23.1-1532880267107 on volume C:\Hadoop\dfs\data: 9ms  
2018-07-29 19:51:19,550 INFO impl.FsDatasetImpl: Total time to add all replicas to map: 10ms  
2018-07-29 19:51:19,574 INFO datanode.VolumeScanner: VolumeScanner(C:\Hadoop\dfs\data, DS-2f71316b-adf7-4343-b24b-5ed86340bfeb): no suitable block pools found to scan. Waiting 1812467541 ms.  
2018-07-29 19:51:19,580 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting at 7/30/18 1:41 AM with interval of 2160000ms  
2018-07-29 19:51:19,585 INFO datanode.DataNode: Block pool BP-742676236-192.168.23.1-1532880267107 (Datanode Uuid 3486d23a-3e4c-4f00-8945-4f92d1bc76bf) service to /0.0.0.0:19000 beginning handshake with NN  
2018-07-29 19:51:24,157 INFO datanode.DataNode: Block pool Block pool BP-742676236-192.168.23.1-1532880267107 (Datanode Uuid 3486d23a-3e4c-4f00-8945-4f92d1bc76bf) service to /0.0.0.0:19000 successfully registered with NN  
2018-07-29 19:51:24,158 INFO datanode.DataNode: For namenode /0.0.0.0:19000 using BLOCKREPORT_INTERVAL of 2160000msec CACHEREPORT_INTERVAL of 10000msec Initial delay: 0msec; heartbeatInterval=3000  
2018-07-29 19:51:24,313 INFO datanode.DataNode: Successfully sent block report 0x794ec2da1ea68e8d, containing 1 storage report(s), of which we sent 1. The reports had 1 total blocks and used 1 RPC(s). This took 4 msec to generate and 48 msecs for RPC and NN processing. Got back one command: FinalizeCommand/5.  
2018-07-29 19:51:24,314 INFO datanode.DataNode: Got finalize command for block pool BP-742676236-192.168.23.1-1532880267107
```

HDFS Commands

- ▶ The File System (FS) shell includes various shell-like commands that directly interact with the Hadoop Distributed File System (HDFS) The FS shell is invoked by
- ▶ Most of the commands in FS shell behave like the corresponding Unix commands
- ▶ A list of HDFS commands can be found [here](#)
- ▶ For example, to copy a file from a local directory to HDFS type:

```
hdfs dfs -put /localfs/source/path /hdfs/destination/path
```

```
C:\Hadoop\sbin>hdfs dfs -put C:/test.txt /
```

- ▶ To display the contents of the HDFS root folder type:

```
hdfs dfs -ls /
```

```
C:\Hadoop\sbin>hdfs dfs -ls /
```

```
Found 1 items
```

```
-rw-r--r--    1 roi supergroup      20 2018-11-14 02:12 /test.txt
```

HDFS Commands

- ▶ To view the contents of a file in HDFS:

```
hdfs dfs -cat /hdfs/path
```

```
C:\Hadoop\sbin>hdfs dfs -cat /test.txt  
Hello world  
Goodbye
```

- ▶ To copy a file from HDFS to a local directory:

```
hdfs dfs -get /hdfs/source/path /localfs/destination/path
```

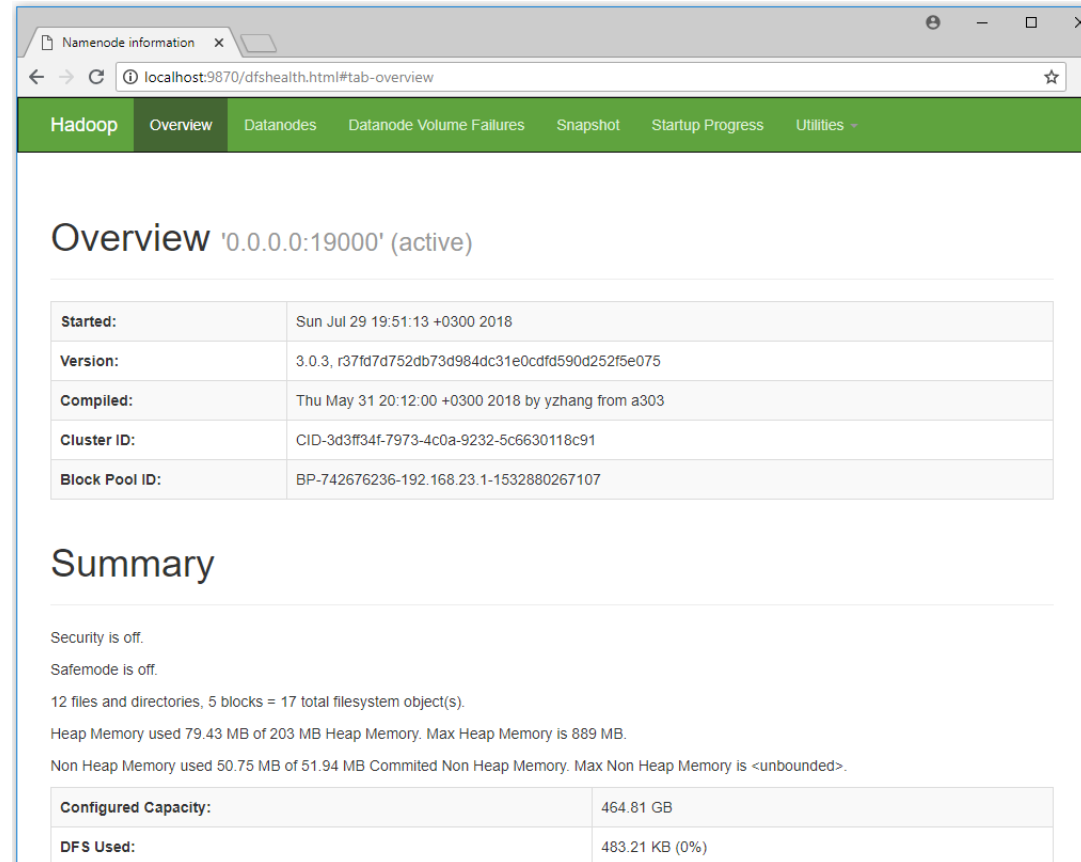
```
C:\Hadoop\sbin>hdfs dfs -get /test.txt C:\temp  
  
C:\Hadoop\sbin>more C:\temp\test.txt  
Hello world  
Goodbye
```

- ▶ To remove a file from HDFS:

```
hdfs dfs -rm /hdfs/path
```

Namenode UI

- Default URL: <http://localhost:9870>



The screenshot shows the Hadoop Namenode UI in a web browser. The browser tab is titled 'Namenode information' and the address bar shows 'localhost:9870/dfshealth.html#tab-overview'. The page has a green navigation bar with tabs: Hadoop, Overview (selected), Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area is titled 'Overview '0.0.0.0:19000' (active)'. Below this is a table with the following information:

Started:	Sun Jul 29 19:51:13 +0300 2018
Version:	3.0.3, r37fd7d752db73d984dc31e0cdfd590d252f5e075
Compiled:	Thu May 31 20:12:00 +0300 2018 by yzhang from a303
Cluster ID:	CID-3d3ff34f-7973-4c0a-9232-5c6630118c91
Block Pool ID:	BP-742676236-192.168.23.1-1532880267107

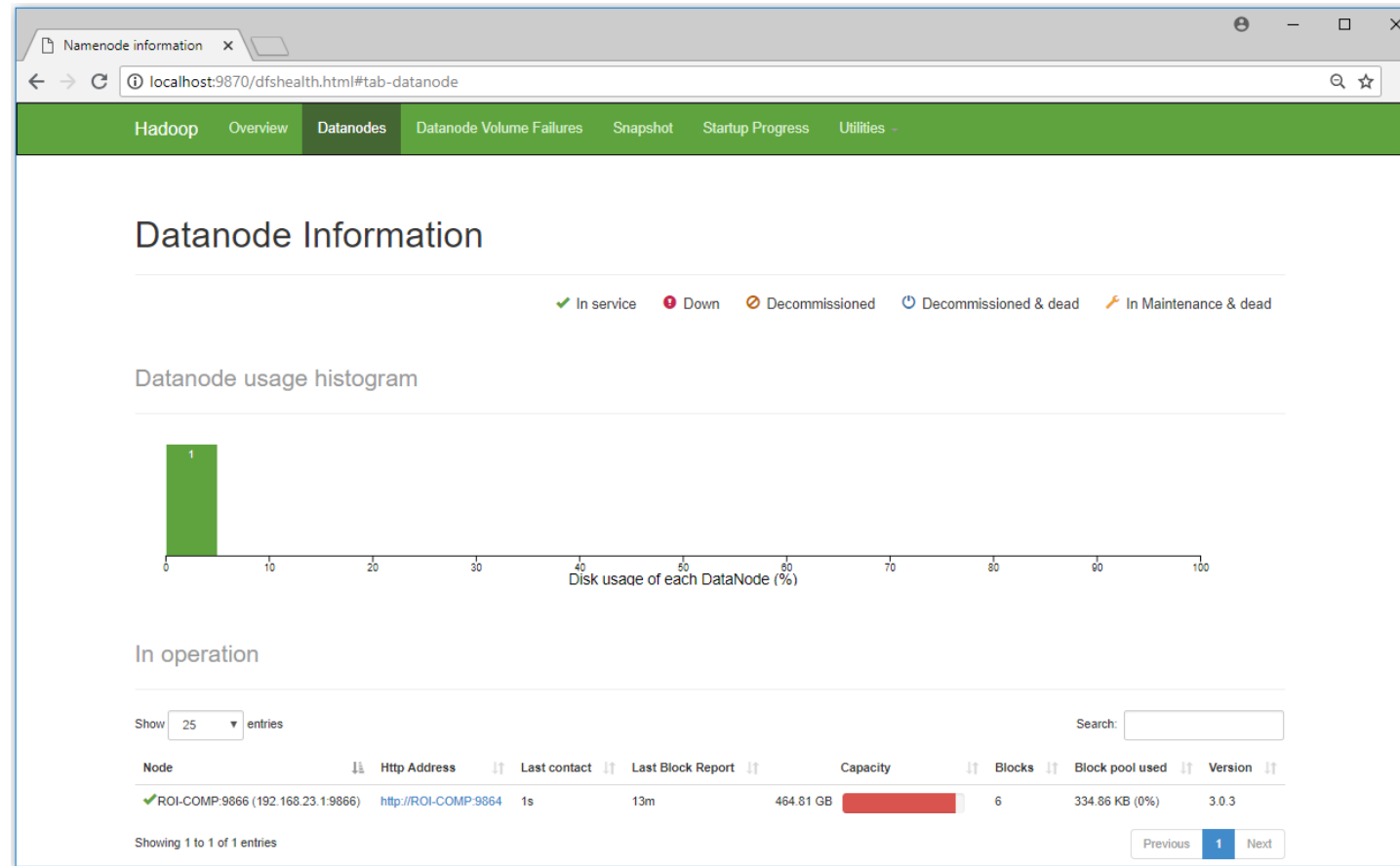
Below the table is a 'Summary' section with the following text:

Security is off.
Safemode is off.
12 files and directories, 5 blocks = 17 total filesystem object(s).
Heap Memory used 79.43 MB of 203 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 50.75 MB of 51.94 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	464.81 GB
DFS Used:	483.21 KB (0%)

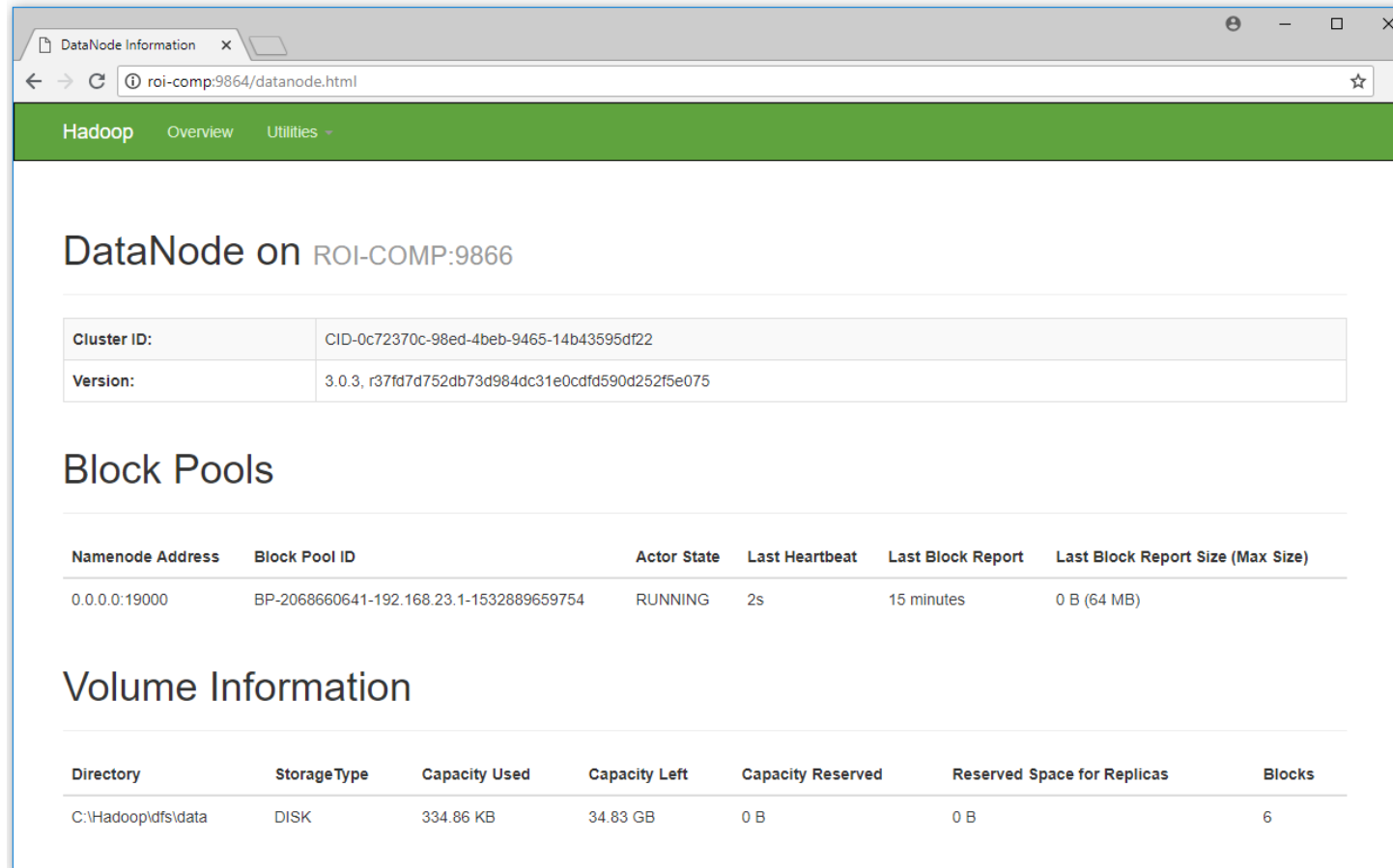
Namenode UI

- ▶ Click the Datanodes tab to find out all the data nodes:



Datanode UI

- ▶ In this case, we only have a single data node with UI URL as <http://localhost:9864>



The screenshot shows the Datanode UI in a web browser. The browser tab is titled "DataNode Information" and the address bar shows "roi-comp:9864/datanode.html". The page has a green header with "Hadoop" and "Overview" tabs. The main content area displays "DataNode on ROI-COMP:9866". Below this, there is a table with two rows: "Cluster ID" and "Version".

Cluster ID:	CID-0c72370c-98ed-4beb-9465-14b43595df22
Version:	3.0.3, r37fd7d752db73d984dc31e0cdfd590d252f5e075

Below the table is the "Block Pools" section, which contains a table with the following data:

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
0.0.0.0:19000	BP-2068660641-192.168.23.1-1532889659754	RUNNING	2s	15 minutes	0 B (64 MB)

Below the Block Pools table is the "Volume Information" section, which contains a table with the following data:

Directory	Storage Type	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
C:\Hadoop\dfs\data	DISK	334.86 KB	34.83 GB	0 B	0 B	6

Exercise

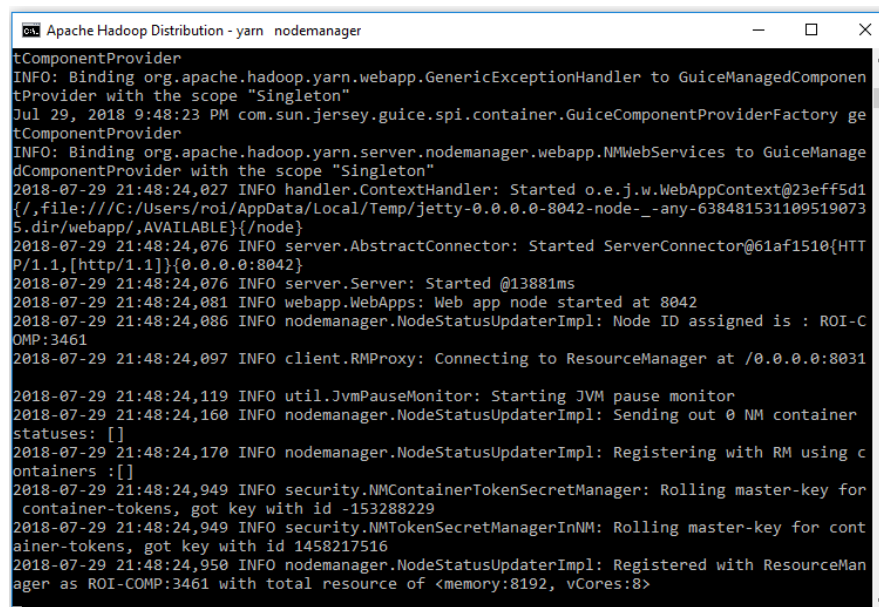
- ▶ Create a directory named mydir under HDFS root /
- ▶ Copy a few files from your local directory to mydir
- ▶ Delete a file in mydir
- ▶ Download a file from HDFS to your local directory

Start YARN Daemons

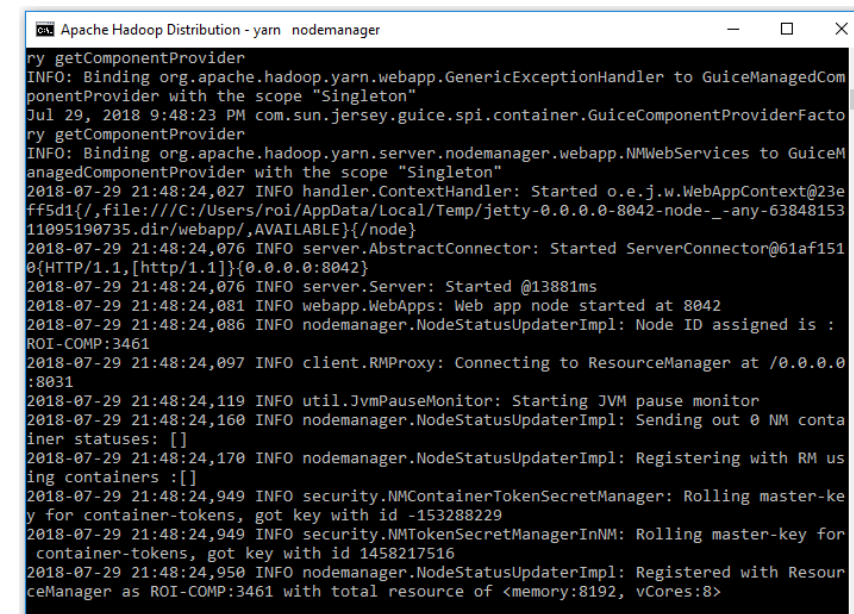
- ▶ Copy the file `hadoop-yarn-server-timelineservice-3.0.3.jar` from `C:\Hadoop\share\hadoop\yarn\timelineservice` to `C:\Hadoop\share\hadoop\yarn`
- ▶ Start YARN through the following command in admin mode:

```
C:\Hadoop\sbin\start-yarn.cmd
```

- ▶ Similar to HDFS, two windows will open:



```
Apache Hadoop Distribution - yarn nodemanager
tComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponen
tProvider with the scope "Singleton"
Jul 29, 2018 9:48:23 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory ge
tComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManag
edComponentProvider with the scope "Singleton"
2018-07-29 21:48:24,027 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@23eff5d1
[/,file:///C:/Users/roi/AppData/Local/Temp/jetty-0.0.0.0-8042-node-_-any-638481531109519073
5.dir/webapp/,AVAILABLE){/node}
2018-07-29 21:48:24,076 INFO server.AbstractConnector: Started ServerConnector@61af1510(HTT
P/1.1,[http/1.1]){0.0.0.0:8042}
2018-07-29 21:48:24,076 INFO server.Server: Started @13881ms
2018-07-29 21:48:24,081 INFO webapp.WebApps: Web app node started at 8042
2018-07-29 21:48:24,086 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : ROI-C
OMP:3461
2018-07-29 21:48:24,097 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8031
2018-07-29 21:48:24,119 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2018-07-29 21:48:24,160 INFO nodemanager.NodeStatusUpdaterImpl: Sending out 0 NM container
statuses: []
2018-07-29 21:48:24,170 INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM using c
ontainers :[]
2018-07-29 21:48:24,949 INFO security.NMContainerTokenSecretManager: Rolling master-key for
container-tokens, got key with id -153288229
2018-07-29 21:48:24,949 INFO security.NMTokenSecretManagerInNM: Rolling master-key for cont
ainer-tokens, got key with id 1458217516
2018-07-29 21:48:24,950 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceMan
ager as ROI-COMP:3461 with total resource of <memory:8192, vCores:8>
```



```
Apache Hadoop Distribution - yarn nodemanager
ry getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedCom
ponentProvider with the scope "Singleton"
Jul 29, 2018 9:48:23 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFacto
ry getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceM
anagedComponentProvider with the scope "Singleton"
2018-07-29 21:48:24,027 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@23e
ff5d1[/,file:///C:/Users/roi/AppData/Local/Temp/jetty-0.0.0.0-8042-node-_-any-63848153
11095190735.dir/webapp/,AVAILABLE){/node}
2018-07-29 21:48:24,076 INFO server.AbstractConnector: Started ServerConnector@61af151
0(HTTP/1.1,[http/1.1]){0.0.0.0:8042}
2018-07-29 21:48:24,076 INFO server.Server: Started @13881ms
2018-07-29 21:48:24,081 INFO webapp.WebApps: Web app node started at 8042
2018-07-29 21:48:24,086 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is :
ROI-COMP:3461
2018-07-29 21:48:24,097 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0
:8031
2018-07-29 21:48:24,119 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2018-07-29 21:48:24,160 INFO nodemanager.NodeStatusUpdaterImpl: Sending out 0 NM conta
iner statuses: []
2018-07-29 21:48:24,170 INFO nodemanager.NodeStatusUpdaterImpl: Registering with RM us
ing containers :[]
2018-07-29 21:48:24,949 INFO security.NMContainerTokenSecretManager: Rolling master-ke
y for container-tokens, got key with id -153288229
2018-07-29 21:48:24,949 INFO security.NMTokenSecretManagerInNM: Rolling master-key for
container-tokens, got key with id 1458217516
2018-07-29 21:48:24,950 INFO nodemanager.NodeStatusUpdaterImpl: Registered with Resour
ceManager as ROI-COMP:3461 with total resource of <memory:8192, vCores:8>
```

Start YARN Daemons

- ▶ Make sure you have at least 10% free disk space, otherwise you might get the following error:

```
Apache Hadoop Distribution - yarn  resourcemanager
2018-07-29 20:46:06,825 INFO ipc.Server: IPC Server listener on 8032: starting
2018-07-29 20:46:06,833 INFO resourcemanager.ResourceManager: Transitioned to active state
2018-07-29 20:46:07,483 INFO resourcemanager.ResourceTrackerService: NodeManager from node ROI-COMP(
cmPort: 2259 httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId RO
I-COMP:2259
2018-07-29 20:46:07,487 INFO rmnode.RMNodeImpl: ROI-COMP:2259 Node Transitioned from NEW to RUNNING
2018-07-29 20:46:07,496 INFO capacity.CapacityScheduler: Added node ROI-COMP:2259 clusterResource: <
memory:8192, vCores:8>
2018-07-29 20:46:07,548 INFO rmnode.RMNodeImpl: Node ROI-COMP:2259 reported UNHEALTHY with details:
1/1 local-dirs usable space is below configured utilization percentage/no more usable space [ /tmp/h
adoop-roi/nm-local-dir : used space above threshold of 90.0% ] ; 1/1 log-dirs usable space is below
configured utilization percentage/no more usable space [ /dep/logs/userlogs : used space above thres
hold of 90.0% ]
2018-07-29 20:46:07,549 INFO rmnode.RMNodeImpl: ROI-COMP:2259 Node Transitioned from RUNNING to UNHE
ALTHY
2018-07-29 20:46:07,551 INFO capacity.CapacityScheduler: Removed node ROI-COMP:2259 clusterResource:
<memory:0, vCores:0>
2018-07-29 20:46:41,202 INFO resourcemanager.ClientRMService: Allocated new applicationId: 1
2018-07-29 20:46:42,741 INFO capacity.CapacityScheduler: Application 'application_1532886366633_0001
' is submitted without priority hence considering default queue/cluster priority: 0
2018-07-29 20:46:42,741 INFO capacity.CapacityScheduler: Priority '0' is acceptable in queue : defau
lt for application: application_1532886366633_0001
2018-07-29 20:46:42,765 INFO resourcemanager.ClientRMService: Application with id 1 submitted by use
r roi
2018-07-29 20:46:42,765 INFO rmapp.RMAppImpl: Storing application with id application_1532886366633_
0001
2018-07-29 20:46:42,767 INFO resourcemanager.RMAuditLogger: USER=roi IP=192.168.23.1 OPERATION=Su
bmit Application Request TARGET=ClientRMService RESULT=SUCCESS APPID=application_1532886366633_
0001
2018-07-29 20:46:42,776 INFO recovery.RMStateStore: Storing info for app: application_1532886366633_
```

Start YARN Daemons

- ▶ You can increase the threshold in the config file yarn-site.xml:

```
<property>  
  <name>yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage</name>  
  <value>98</value>  
</property>
```

YARN Commands

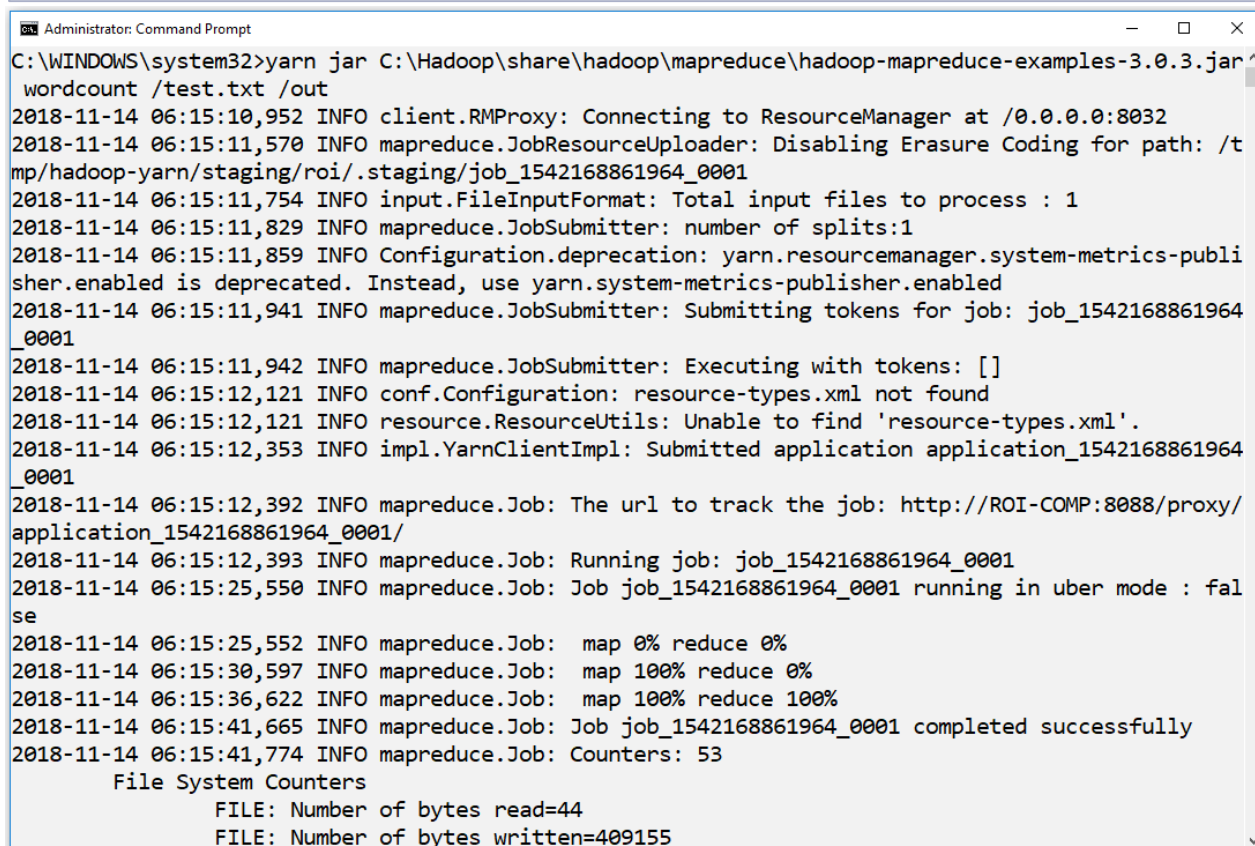
- Type yarn in command line to see a list of available commands:

```
C:\Users\roi>yarn
Usage: yarn [--config confdir] [--loglevel loglevel] COMMAND
    where COMMAND is one of:
    resourcemanager    run the ResourceManager
    nodemanager        run a nodemanager on each slave
    router             run the Router daemon
    timelineserver      run the timeline server
    timelinereader      run the timeline reader server
    rmadmin            admin tools
    version            print the version
    jar <jar>          run a jar file
    application         prints application(s) report/kill application
    applicationattempt  prints applicationattempt(s) report
    cluster            prints cluster information
    container          prints container(s) report
    node               prints node report(s)
    queue              prints queue information
    logs               dump container logs
    schedulerconf       updates scheduler configuration
    classpath           prints the class path needed to get the
                        Hadoop jar and the required libraries
    daemonlog          get/set the log level for each daemon
    or
    CLASSNAME          run the class named CLASSNAME
Most commands print help when invoked w/o parameters.
```

Running a Map-Reduce Job

- ▶ Run the following sample job (under administrator) to perform word count:

```
yarn jar C:\Hadoop\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.3.jar  
wordcount /test.txt /out
```



```
Administrator: Command Prompt
C:\WINDOWS\system32>yarn jar C:\Hadoop\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.0.3.jar  
wordcount /test.txt /out
2018-11-14 06:15:10,952 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
2018-11-14 06:15:11,570 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /t  
mp/hadoop-yarn/staging/roi/.staging/job_1542168861964_0001
2018-11-14 06:15:11,754 INFO input.FileInputFormat: Total input files to process : 1
2018-11-14 06:15:11,829 INFO mapreduce.JobSubmitter: number of splits:1
2018-11-14 06:15:11,859 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publi  
sher.enabled is deprecated. Instead, use yarn.system-metrics-publisher.enabled
2018-11-14 06:15:11,941 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1542168861964  
_0001
2018-11-14 06:15:11,942 INFO mapreduce.JobSubmitter: Executing with tokens: []
2018-11-14 06:15:12,121 INFO conf.Configuration: resource-types.xml not found
2018-11-14 06:15:12,121 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2018-11-14 06:15:12,353 INFO impl.YarnClientImpl: Submitted application application_1542168861964  
_0001
2018-11-14 06:15:12,392 INFO mapreduce.Job: The url to track the job: http://ROI-COMP:8088/proxy/  
application_1542168861964_0001/
2018-11-14 06:15:12,393 INFO mapreduce.Job: Running job: job_1542168861964_0001
2018-11-14 06:15:25,550 INFO mapreduce.Job: Job job_1542168861964_0001 running in uber mode : fal  
se
2018-11-14 06:15:25,552 INFO mapreduce.Job: map 0% reduce 0%
2018-11-14 06:15:30,597 INFO mapreduce.Job: map 100% reduce 0%
2018-11-14 06:15:36,622 INFO mapreduce.Job: map 100% reduce 100%
2018-11-14 06:15:41,665 INFO mapreduce.Job: Job job_1542168861964_0001 completed successfully
2018-11-14 06:15:41,774 INFO mapreduce.Job: Counters: 53
    File System Counters
        FILE: Number of bytes read=44
        FILE: Number of bytes written=409155
```


Running a Map-Reduce Job

- ▶ To the result of this job is stored in the /out HDFS folder:

```
C:\WINDOWS\system32>hdfs dfs -ls /out
Found 2 items
-rw-r--r--    1 roi supergroup          0 2018-11-14 06:15 /out/_SUCCESS
-rw-r--r--    1 roi supergroup        26 2018-11-14 06:15 /out/part-r-00000

C:\WINDOWS\system32>hdfs dfs -cat /out/part-r-00000
Goodbye 1
Hello   1
world   1
```

Running a Map-Reduce Job

- ▶ You can find the Java source code of this example under C:\Hadoop\share\hadoop\mapreduce\sources
- ▶ The class that implements the mapper task:

```
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);  
            }  
        }  
    }  
}
```

Running a Map-Reduce Job

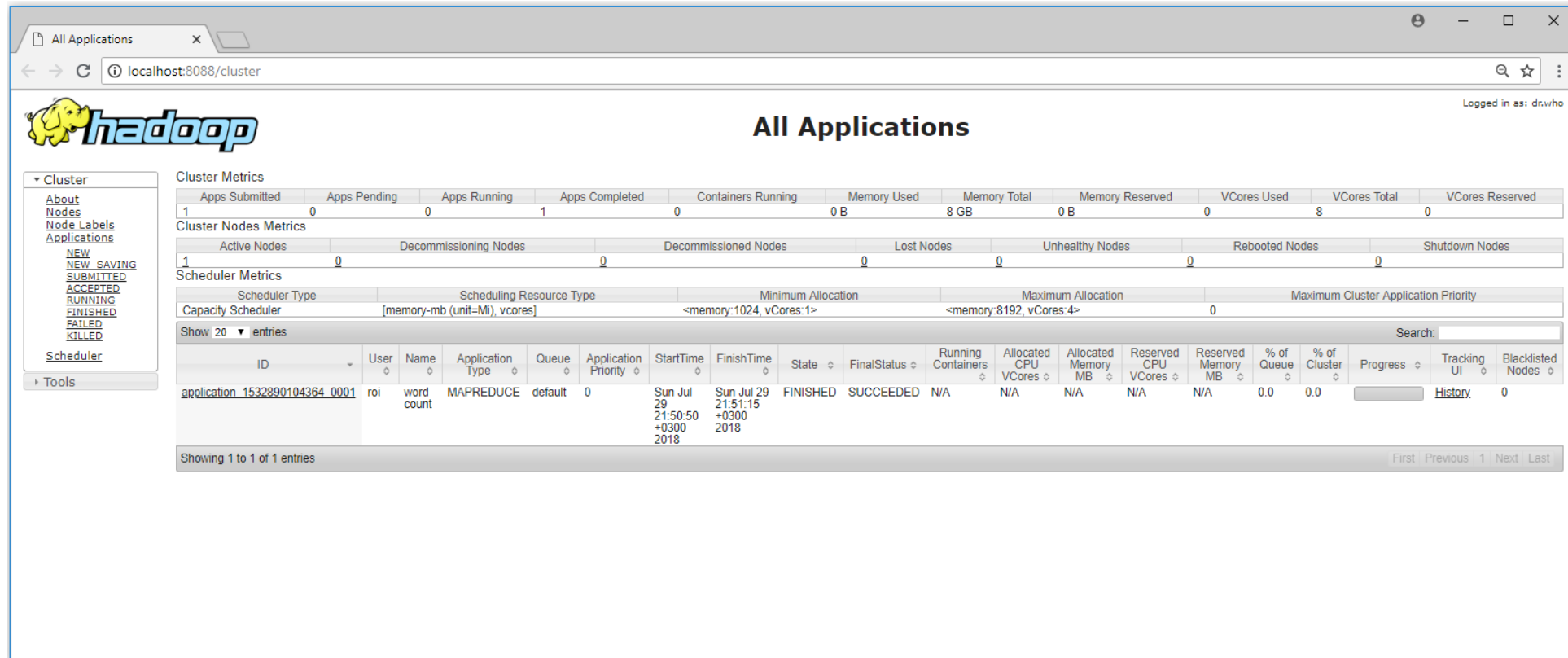
- ▶ The class implementing the reducer task:

```
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);
    }
}
```

Application Status UI

- ▶ You can also view your job status through YRAN website. The default path is <http://localhost:8088>



The screenshot shows the Hadoop YARN Application Status UI in a web browser. The browser address bar shows `localhost:8088/cluster`. The page title is "All Applications". The left sidebar contains a navigation menu with options: Cluster, About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools. The main content area displays various metrics and a table of applications.

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
1	0	0	1	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

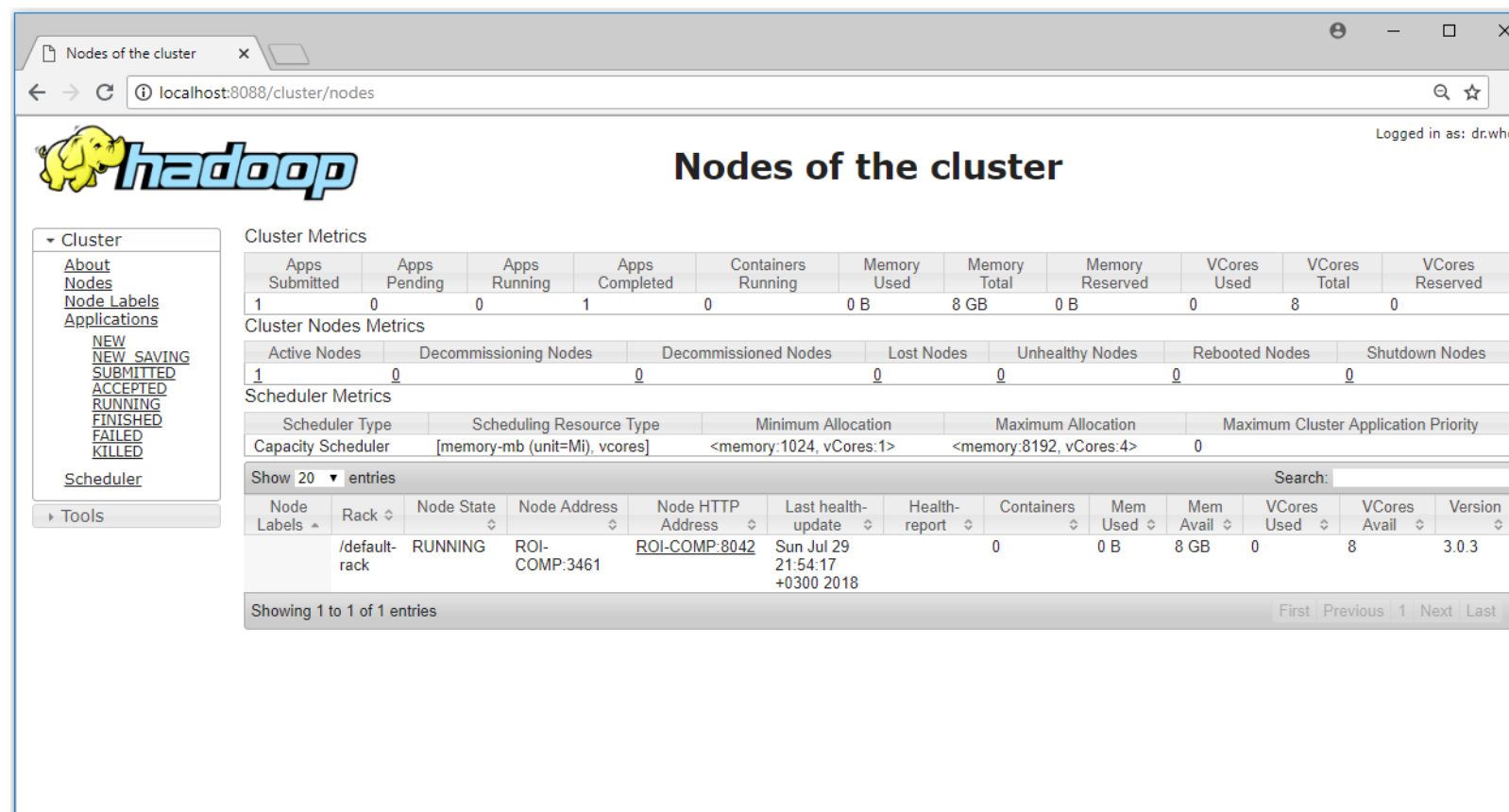
Applications Table

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1532890104364_0001	roi	word count	MAPREDUCE	default	0	Sun Jul 29 21:50:50 +0300 2018	Sun Jul 29 21:51:15 +0300 2018	FINISHED	SUCCEEDED	N/A	N/A	N/A	N/A	N/A	0.0	0.0		History	0

Showing 1 to 1 of 1 entries

Cluster Status UI

- Click Nodes to see the status of the nodes in the cluster:



The screenshot shows the Hadoop Cluster Status UI in a web browser. The browser tab is titled "Nodes of the cluster" and the address bar shows "localhost:8088/cluster/nodes". The page is logged in as "dr.who".

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
1	0	0	1	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Node Details Table

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack	RUNNING	ROI-COMP:3461	ROI-COMP:8042	Sun Jul 29 21:54:17 +0300 2018	0	0 B	8 GB	0	8	0	8	3.0.3

Showing 1 to 1 of 1 entries