Universidad Autónoma de Madrid

excelencia Campus Internacional UAM CSIC+

Escuela Politécnica Superior

Arquitecturas para tratar grandes volúmenes de información

Procesamiento de Datos a Gran Escala

Máster en Ciencia de Datos

Francisco J. Gómez Arribas

Cluster Hadoop

- Introducción a Hadoop
- Hadoop se puede instalar de tres maneras distintas:
  - Standalone
  - Pseudo-Distributed
  - Fully Distributed

Google Origins

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

2003

MapReduce: Simplified Data Processing on Large Clusters

2004

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber

{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com

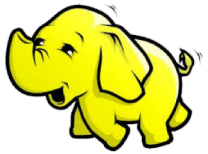
Google, Inc.

Abstract

igtable is a distributed storage system for managing
tured data that is designed to scale to a very large
petabytes of data across thousands of commodity
rs. Many projects at Google store data in Bigtable,
ding web indexing, Google Earth, and Google Fi-
e. These applications place very different demands
igtable, both in terms of data size (from URLs to

achieved scalability and high performance, but Big
provides a different interface than such systems. Big
does not support a full relational data model; inste
provides clients with a simple data model that sup
dynamic control over data layout and format, an
lows clients to reason about the locality properties o
data represented in the underlying storage. Data i
dexed using row and column names that can be arbi
strings. Bigtable also treats data as uninterpreted str

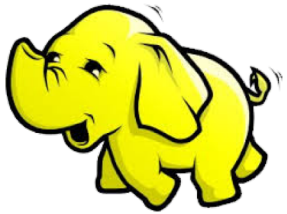
2006



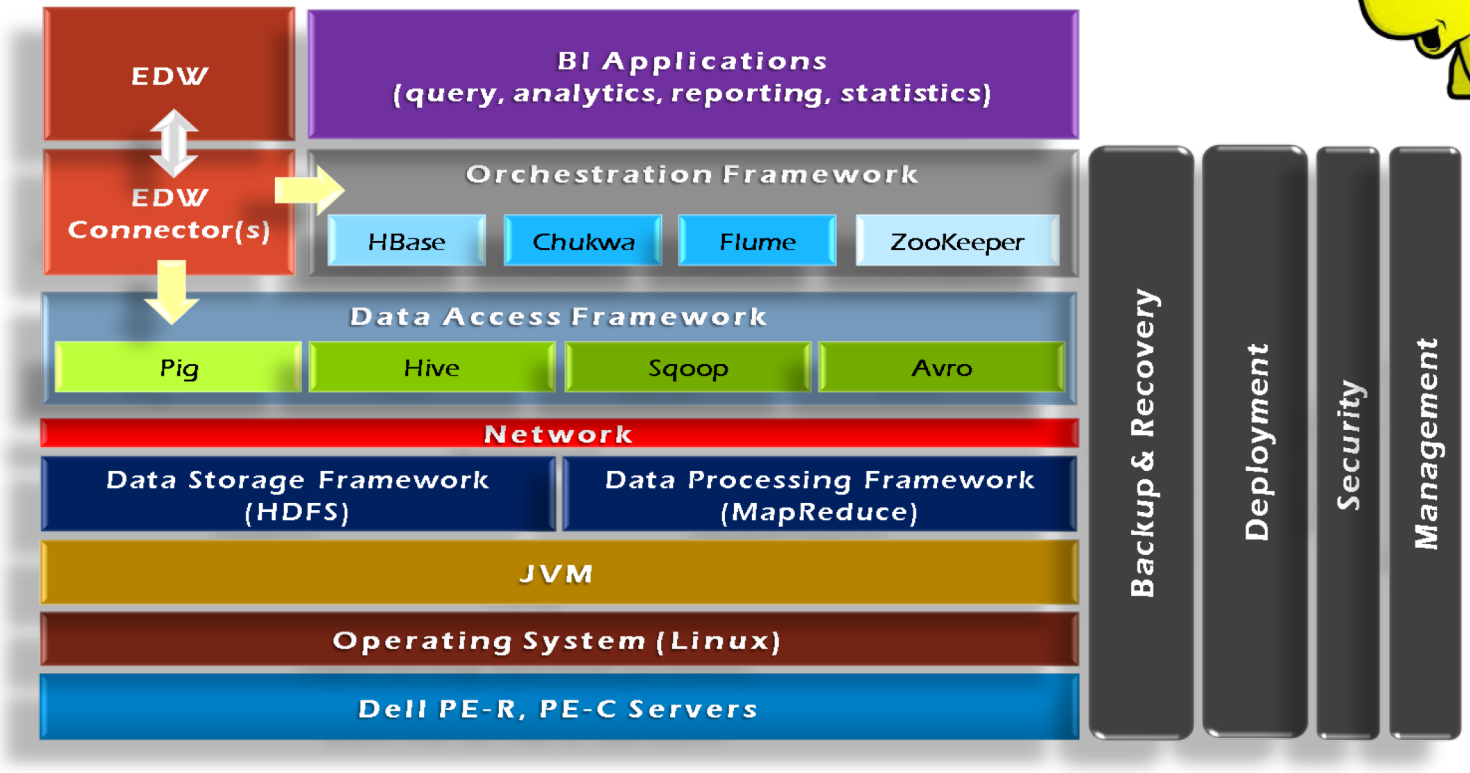
What is Hadoop?

- **Hadoop:**
  - An open-source software framework that supports data-intensive distributed applications, licensed under the Apache v2 license.

- **Goals / Requirements:**
  - Abstract and facilitate the storage and processing of large and/or rapidly growing data sets
    - Structured and non-structured data
    - Simple programming models
  - High scalability and availability
  - Use commodity (cheap!) hardware with little redundancy
  - Fault-tolerance
  - Move computation rather than data

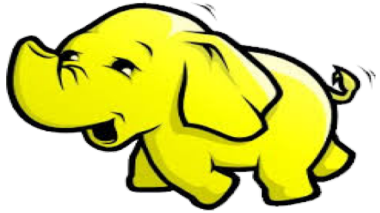


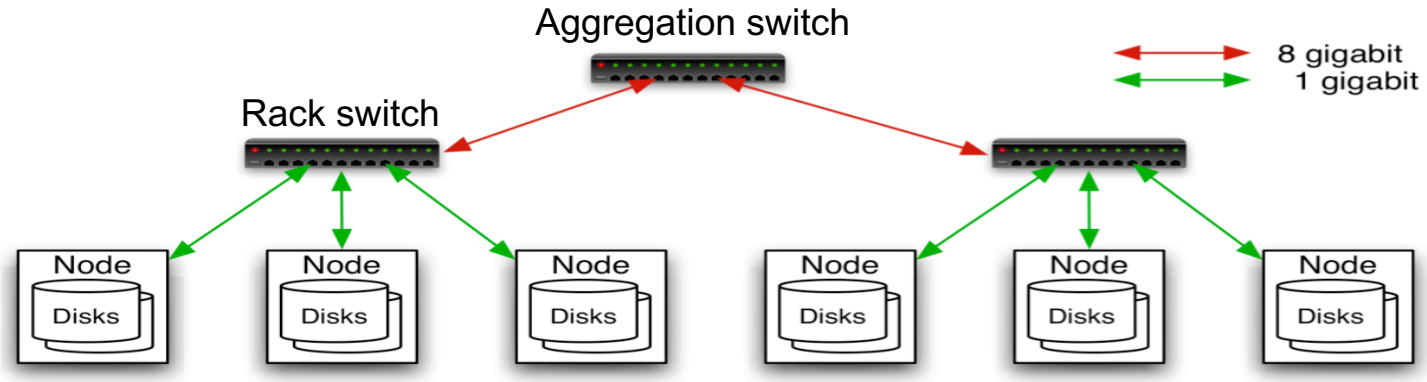
Hadoop Framework Tools



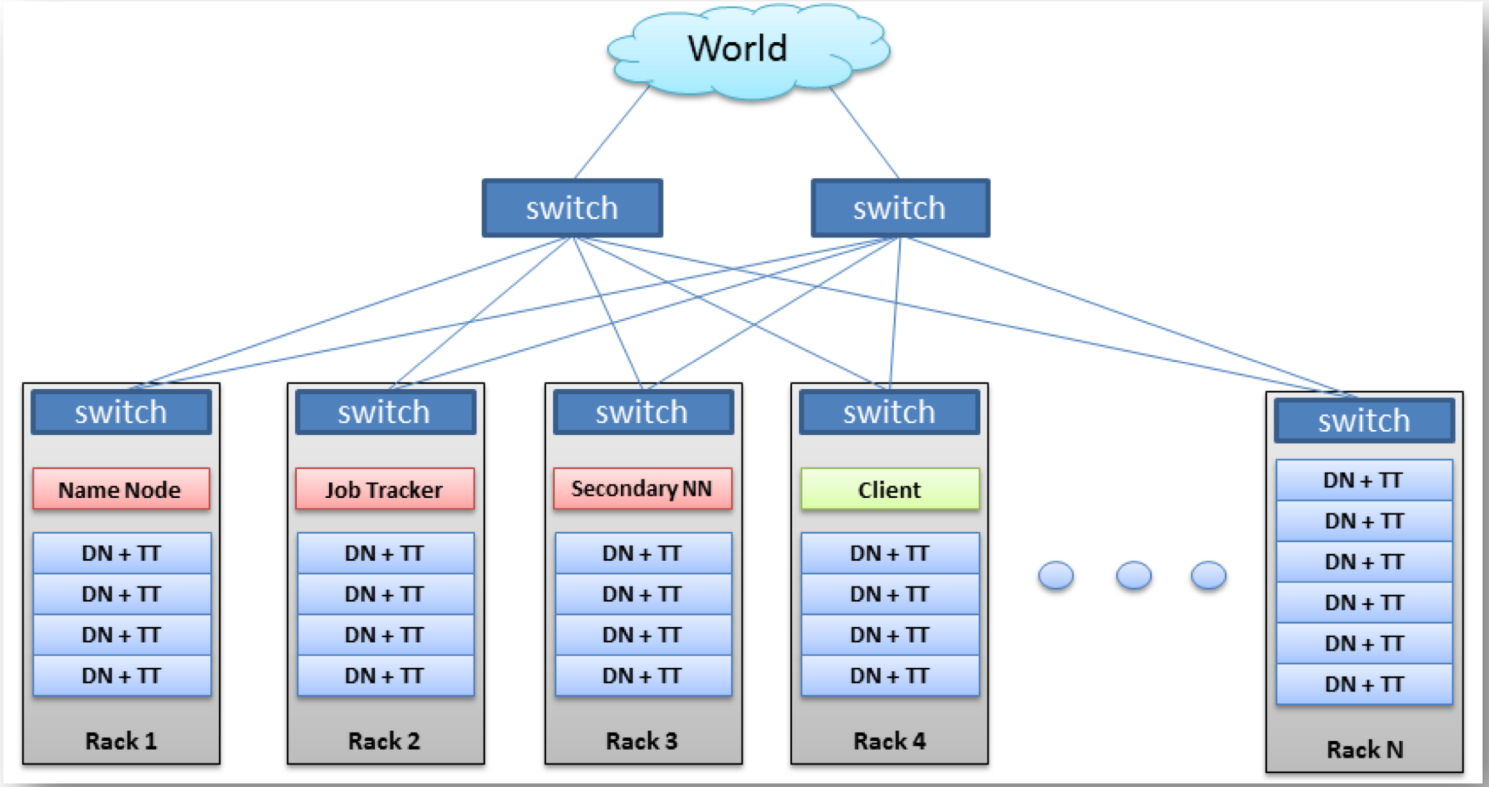
Cluster Hadoop

- Cluster creado con commodity Hardware;
  - Nodos inicalmente eran PCs
  - 30-40 nodos/rack
  - Red a1 gigabit/s en rack





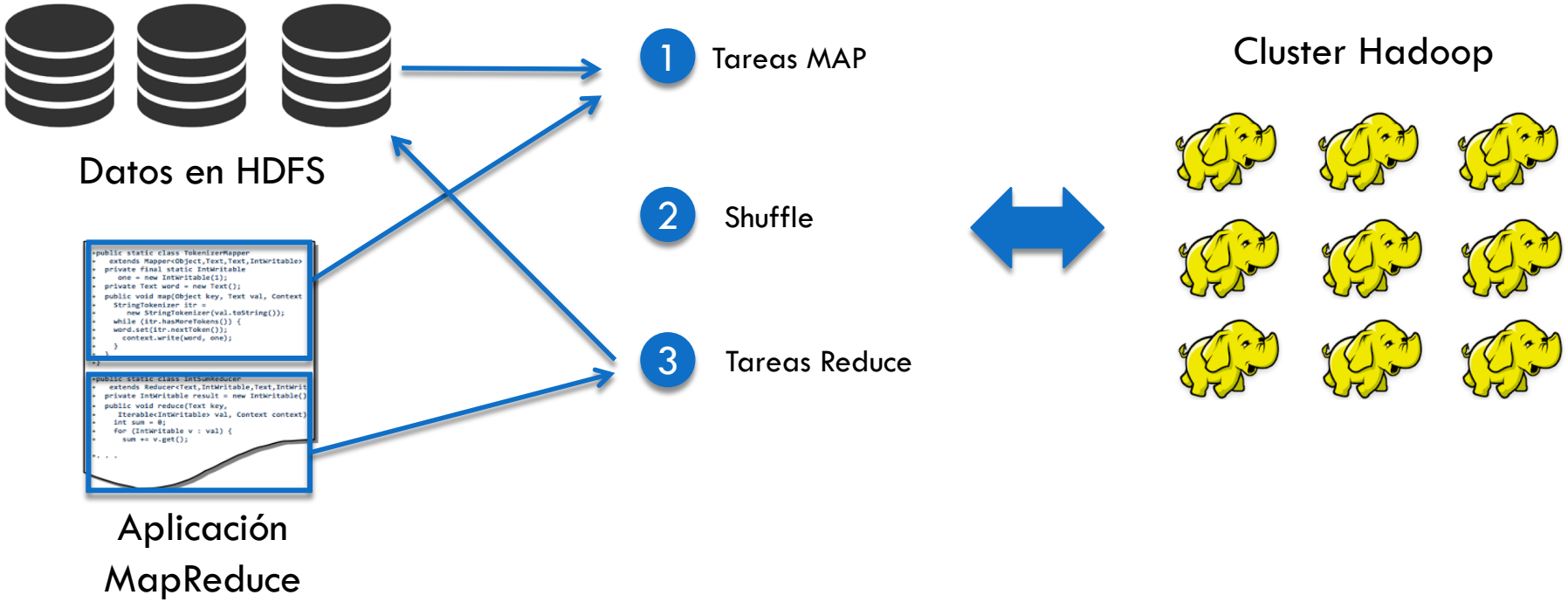
Arquitectura de almacenamiento HDFS



HDFS : Hadoop Distributed File System

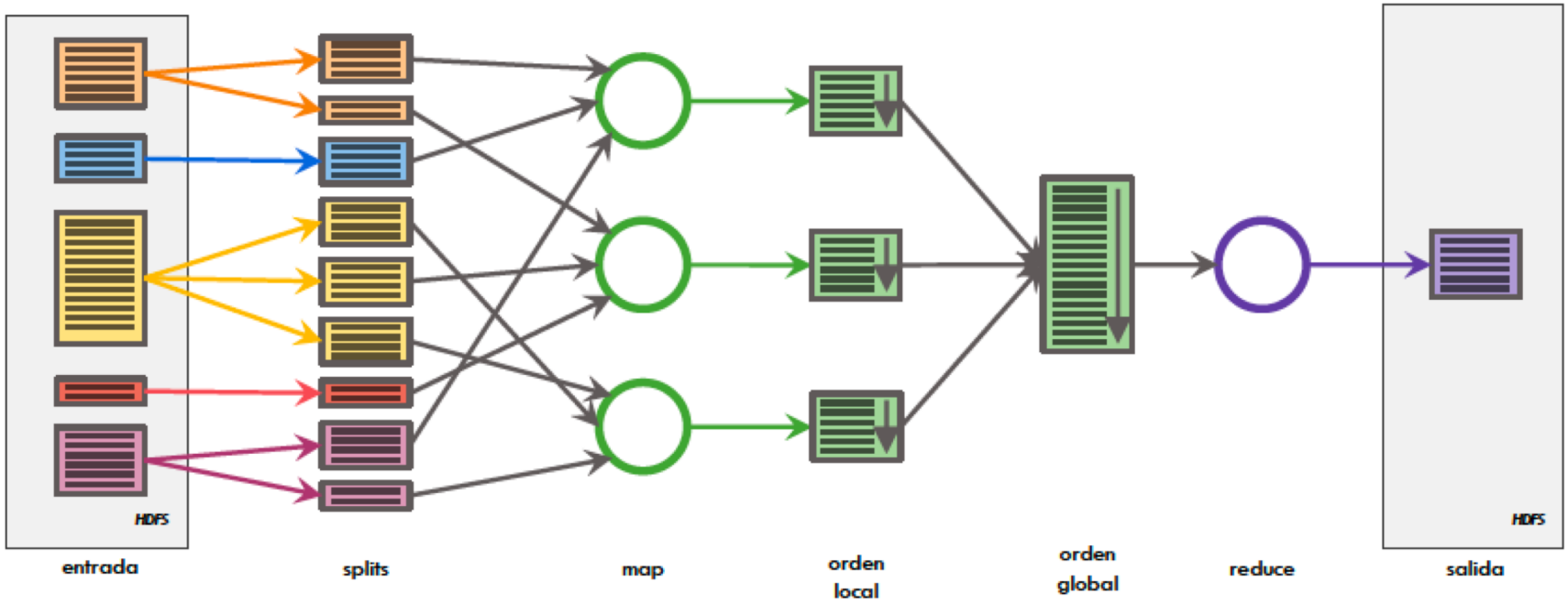
- Sistema de Ficheros distribuido muy grande
  - 10K nodos, 100 millones de ficheros 10PB
- Realizado con "*Commodity Hardware*"
  - Ficheros replicados para tolerancia a fallos
  - Detecta fallos y recupera los datos.
- Optimizado para proceso por lotes ("*Batch Processing*").
  - Expone la localización de los datos y así permite que la computación se pueda llevar cerca de los datos.
  - El ancho de banda agregado es muy alto.

Cómo se ejecuta una aplicación Hadoop



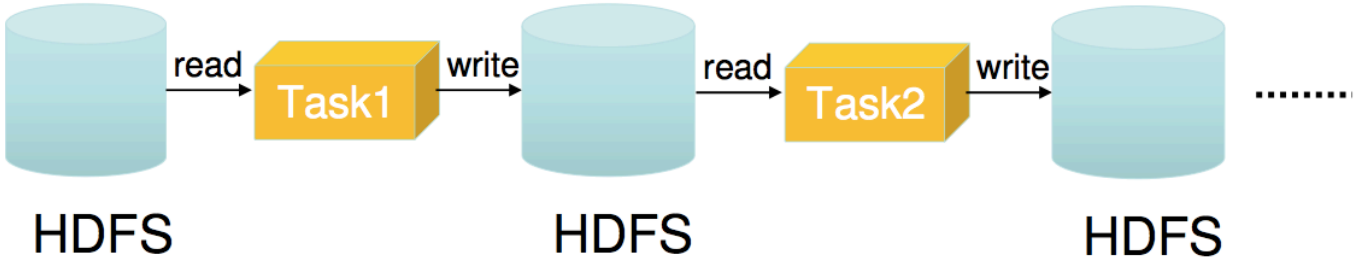
Cómo se ejecuta una aplicación Hadoop

- ¿Cómo se ejecutan las tareas?

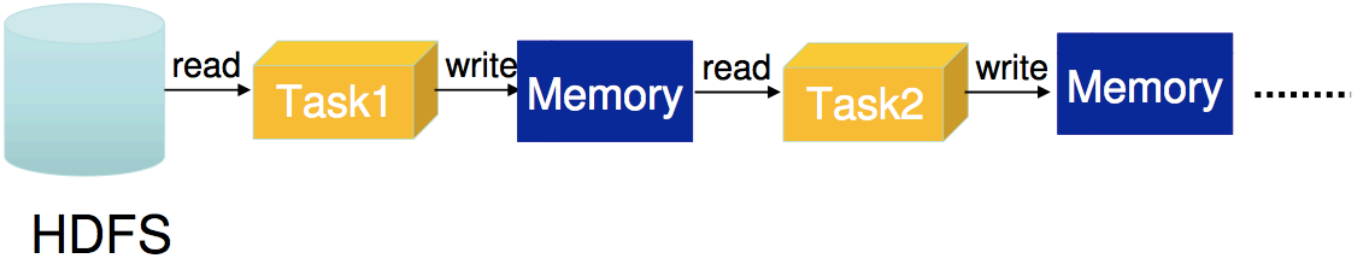


Intro: MapReduce vs Spark

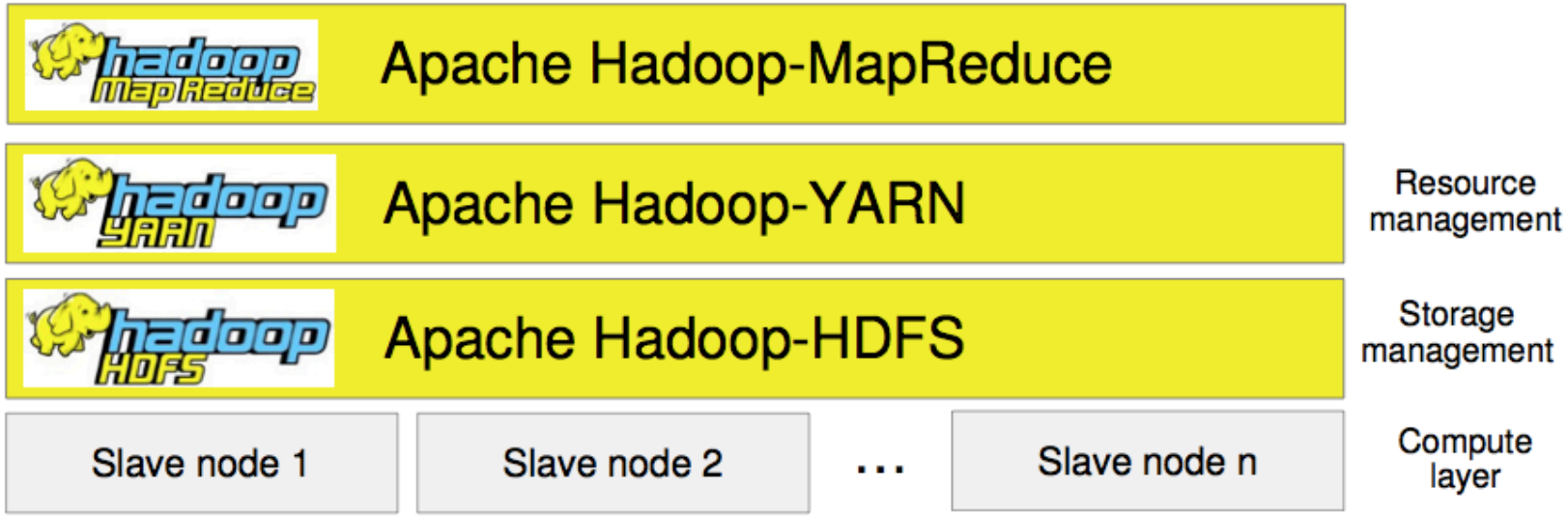
- Map/Reduce:



- Spark:



Intro: Arquitectura Hadoop



¿Qué necesitamos?

- Una ordenador con software para crear máquinas virtuales y conexión a Internet
  - La instalación en una máquina real es equivalente
  - Usaremos VMware
- Una ISO de Linux
  - Usaremos CentOS que es una RedHat “opensource”
- Hadoop
  - Este y otro software relacionado lo descargaremos una vez instalado Linux

Instalación de Hadoop

- Partimos de una imagen de MV que encontraréis en cada equipo
  - Recomendación: cambiar la configuración de la MV para "darle potencia", ya que los equipos del laboratorio nos lo permiten
    - 2 GB de RAM
    - 2 procesadores
  - Acceso:
    - bigdata (bigdata)
    - root (bigdata)

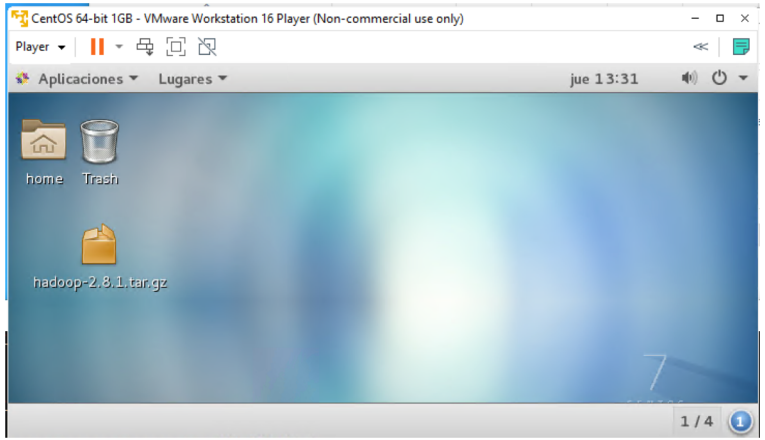
Antes de empezar, copiaros la imagen de MV tal cómo la encontréis en vuestro equipo, pues la utilizaremos varias veces como punto de partida

Máquina virtual de partida

- Instalación “minimalista”
  - CentOS 7 (ISO disponible en los equipos del laboratorio)
    - 2 GB de RAM
    - 1 procesador
  - Instalación paquete de software “Compute node”
    - Se añade:
      - Herramientas de monitorización de HW
      - Herramientas de rendimiento
      - Administración remota de Linux
      - Herramientas de desarrollo
  - Habilitar interfaz de red

Máquina virtual de partida

- En esta máquina virtual ya se ha descargado en el escritorio:
  - `wget apache.rediris.es/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz`

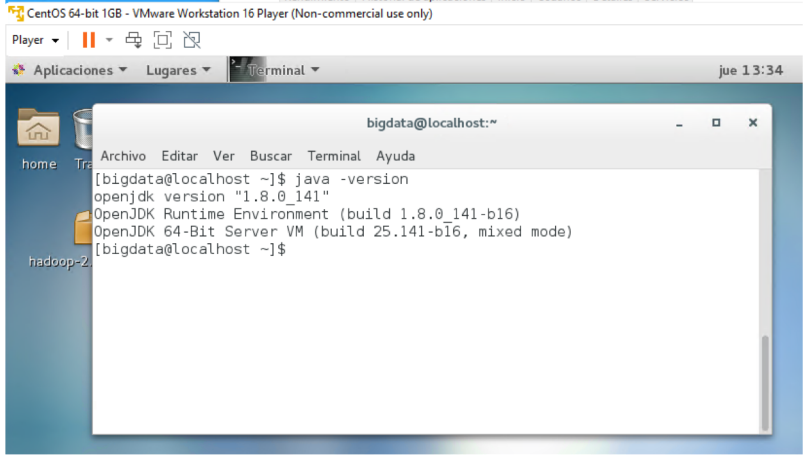


- Los requisitos de hadoop 2.8
  - Versión de java 1.7 comprobar con :
  - java -versión

  Y si no coincide instalar:
  - `sudo yum -y install java-1.7.0-openjdk`

Máquina virtual de partida

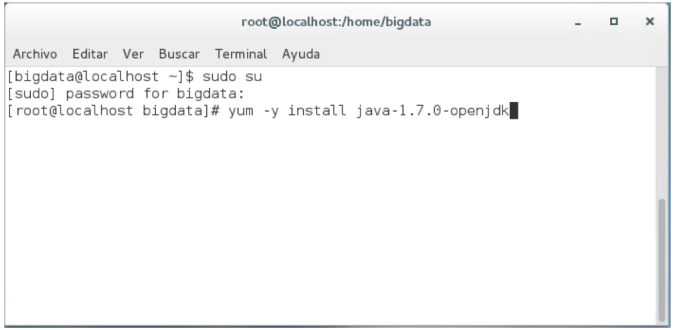
- los requisitos de hadoop 2.8
  - Versión de java 1.7 comprobar con :
  - java -versión



Y si no coincide instalar:

- `yum -y install java-1.7.0-openjdk`

Se necesita ser root.



Instalar también:

- `yum -y install rsync`

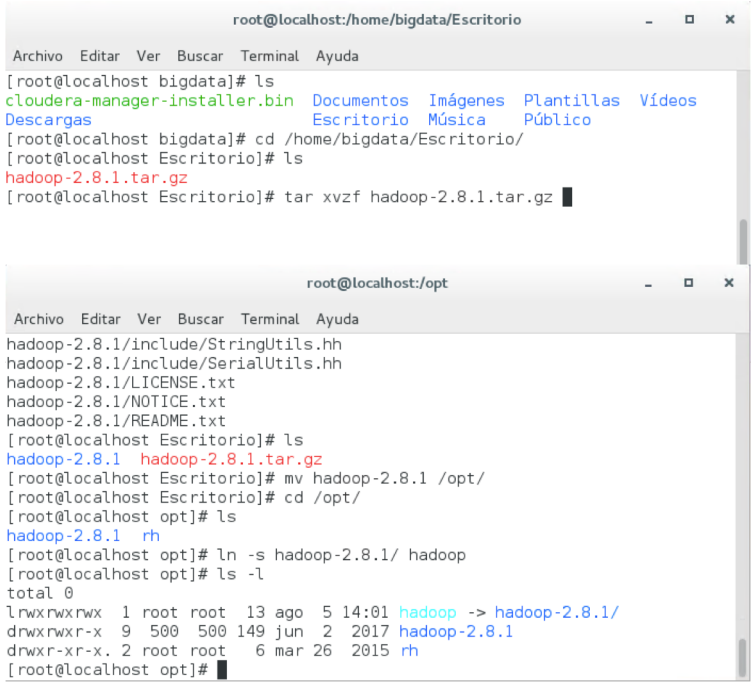
Instalar Hadoop

- La 1ª instalación es Single Node (una sola máquina)
- Recordamos los requisitos ya instalados:
  - Java 7. La opción más sencilla en OpenJDK que viene con CentOS
    - `> yum -y install java-1.7.0-openjdk`
  - SSH. Viene con Linux, pero necesitamos rsync
    - `> yum -y install rsync`
- Y también se obtuvo hadoop con:
  - `> wget apache.rediris.es/hadoop/common/hadoop-2.8.1/hadoop-2.8.1.tar.gz`

Otras versiones ver https://apache.rediris.es/hadoop/common

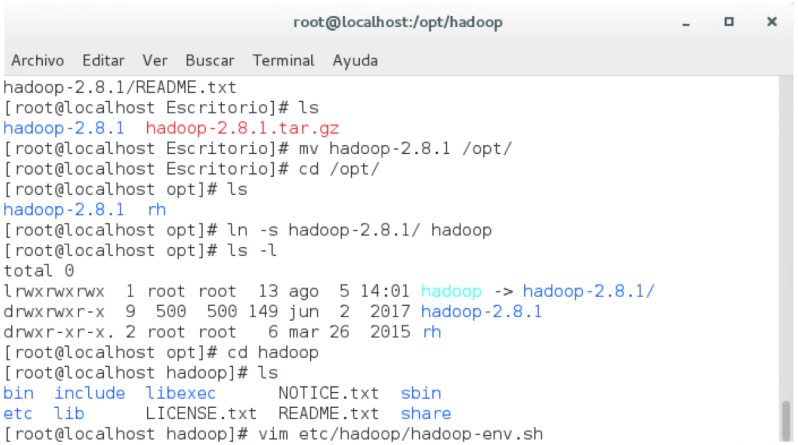
Instalar Hadoop

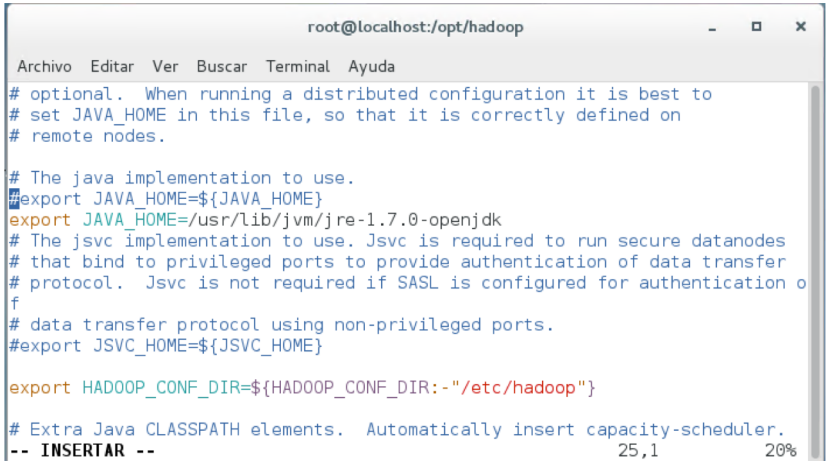
- Descomprimir hadoop
  - `> cd /home/bigdata/Escritorio`
  - `> tar xvzf hadoop-2.8.1.tar.gz`
- Movemos a /opt
  - `> mv hadoop-2.8.1 /opt`
- Creamos un link a la carpeta de hadoop
  - Permite elegir entre diferentes versiones
  - `> cd /opt`
  - `> ln -s hadoop-2.8.1 hadoop`
  - `> cd hadoop`
- Editar fichero *etc/hadoop/hadoop-env.sh*
  - *export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk*



Instalar Hadoop

- Editar fichero *etc/hadoop/hadoop-env.sh*
  - *export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk*





Instalar Hadoop

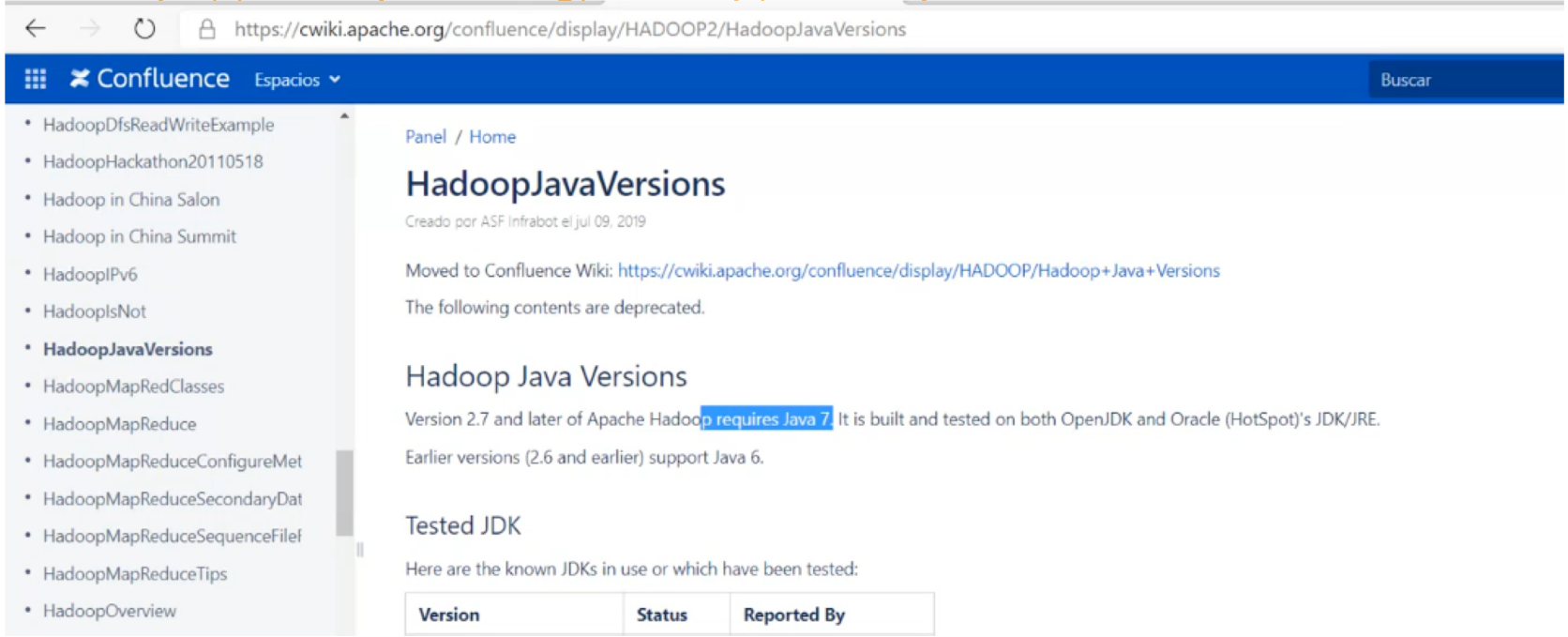
- Hemos descomprimido el código de hadoop dentro de nuestro sistema de ficheros local

- ¿Por qué no hay que compilar nada?
  - Hadoop es código JAVA!

- Conviene chequear compatibilidad con tu versión de Java
  - http://hadoop.apache.org/releases.html
  - https://wiki.apache.org/hadoop/HadoopJavaVersions

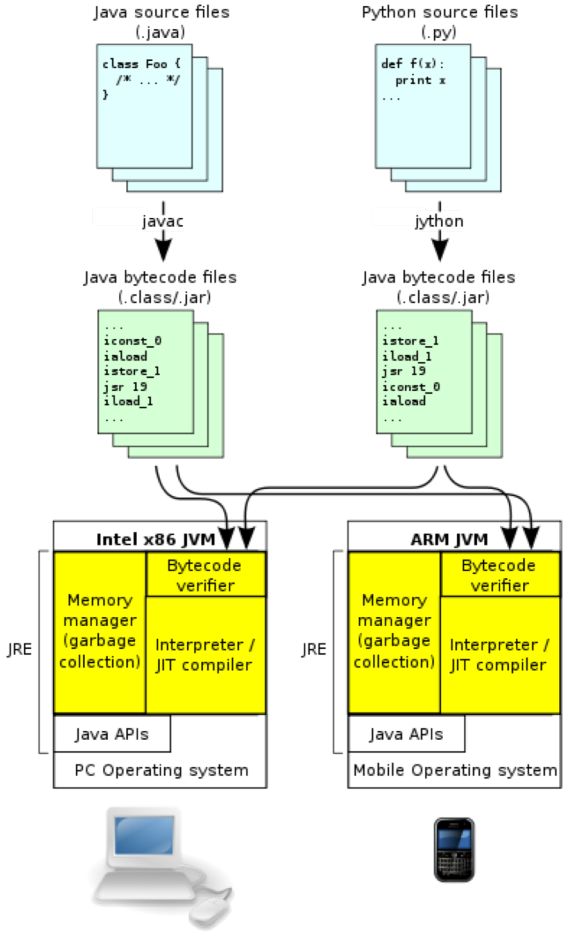
Instalar Hadoop

- http://hadoop.apache.org/releases.html
- https://wiki.apache.org/hadoop/HadoopJavaVersions



EL lenguaje de programación Java

- Máquina Virtual de Java (JVM)
  - Portabilidad
    - Cualquier dispositivo
    - Cualquier SO
    - Java > ByteCode > Código ejecutable
  - Menor rendimiento
    - Etapas intermedias de ejecución
    - Dificultad de programación MP/MC
  - Aislamiento

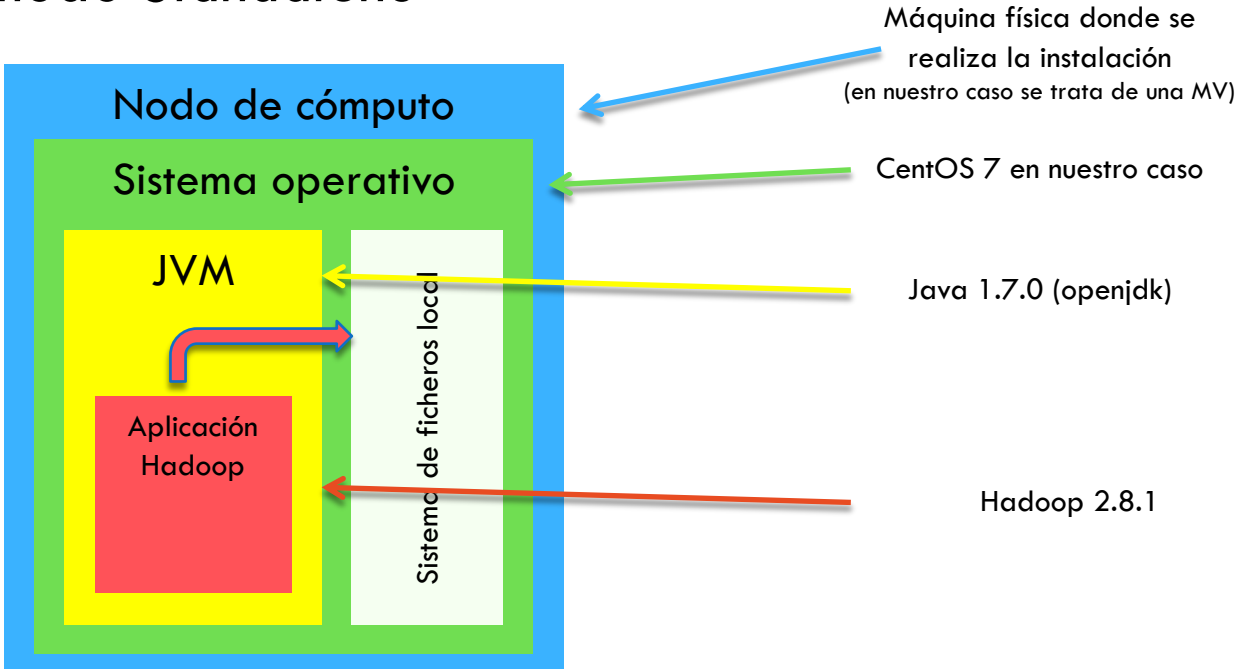


Instalación de Hadoop

- Instalación en modo *Standalone*
  - No se ejecutan demonios
  - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
  - No se usa HDFS
  - Adecuado para desarrollo y debug de aplicaciones

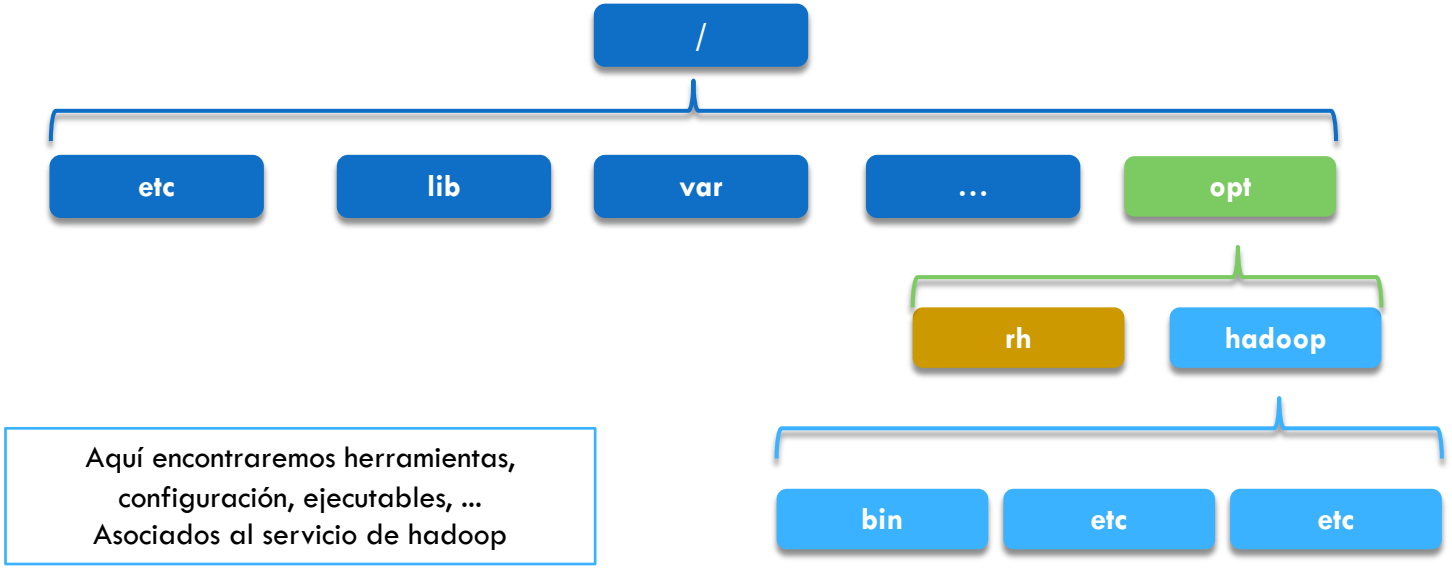
Instalación de Hadoop

- Instalación en modo *Standalone*



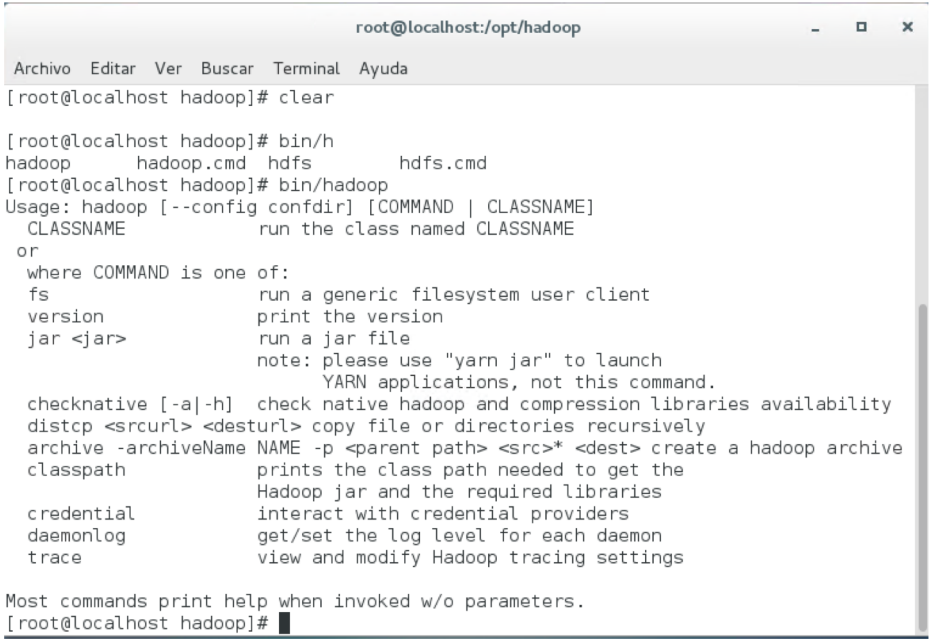
Instalación de Hadoop

- Instalación en modo *Standalone*



Instalar Hadoop - Standalone

- Por defecto Hadoop está configurado para ejecutarse en modo non-distributed, como un único proceso java
- Ejemplo1
  - `> cd /opt/hadoop`
  - `> bin/hadoop`



Instalar Hadoop - Standalone

- Ejemplo1
  - > bin/hadoop

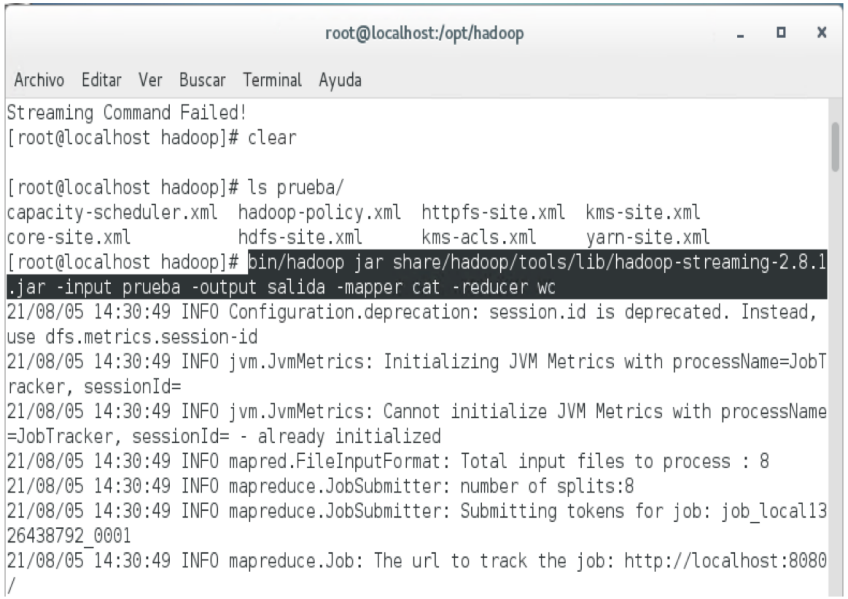
```
jar                                                    Clase a invocar
share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar

-input prueba          Ficheros (en este caso directorio)
-output salida         Directorio de salida ( no debe existir)
-mapper cat
-reducer wc            Qué hacer en las fase map y reduce
```

Parámetros de la clase

Instalar Hadoop - Standalone

- Ejemplo1: Prepara un directorio con los datos(para contar líneas|palabras|caracteres)
  - `> mkdir prueba`
  - `> cp etc/hadoop/*.xml prueba`



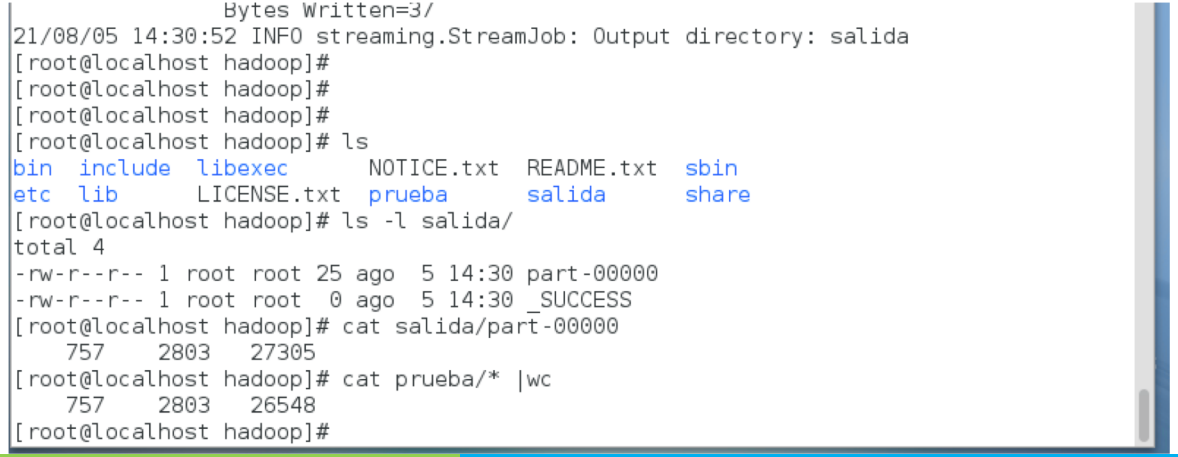
No se usa HDFS!!

Ejecuta el comando

- `> bin/hadoop jar share/hadoop/tools/lib/hadoop-streaming-2.8.1.jar –input prueba –output salida –mapper cat –reducer wc`

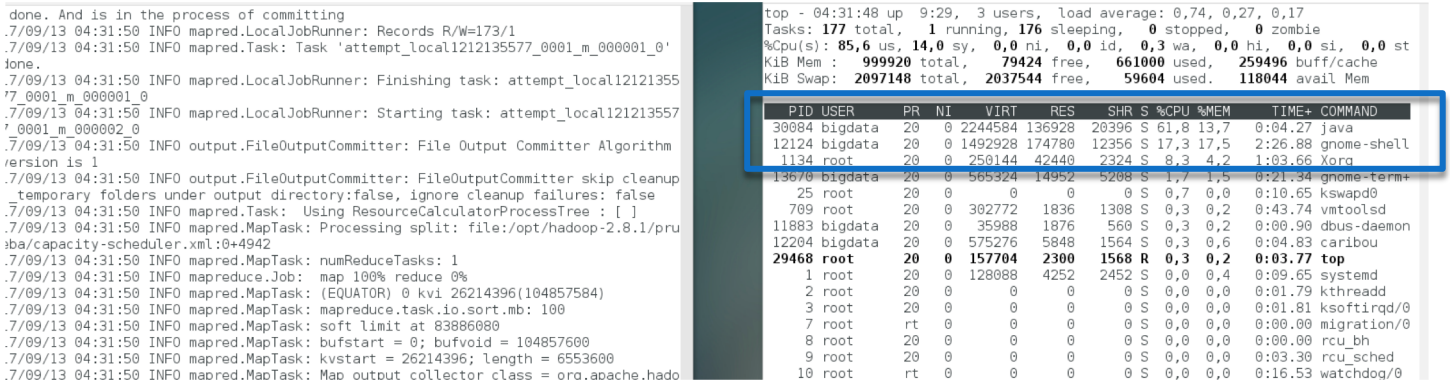
Instalar Hadoop - Standalone

- Ejemplo1
  - Comprobar la salida
    - `> cat salida/part-00000`
  - Comparar resultado con el siguiente comando
    - `> wc prueba/*`



Instalar Hadoop - Standalone

- Ejemplo1 - Observaciones
  - NO hemos utilizado HDFS en ningún momento
    - Sólo se ha utilizado el FS local
      - Directorios “prueba” y “salida”
  - Repetir la ejecución de la prueba mientras en otra terminal ejecutas el comando “top”



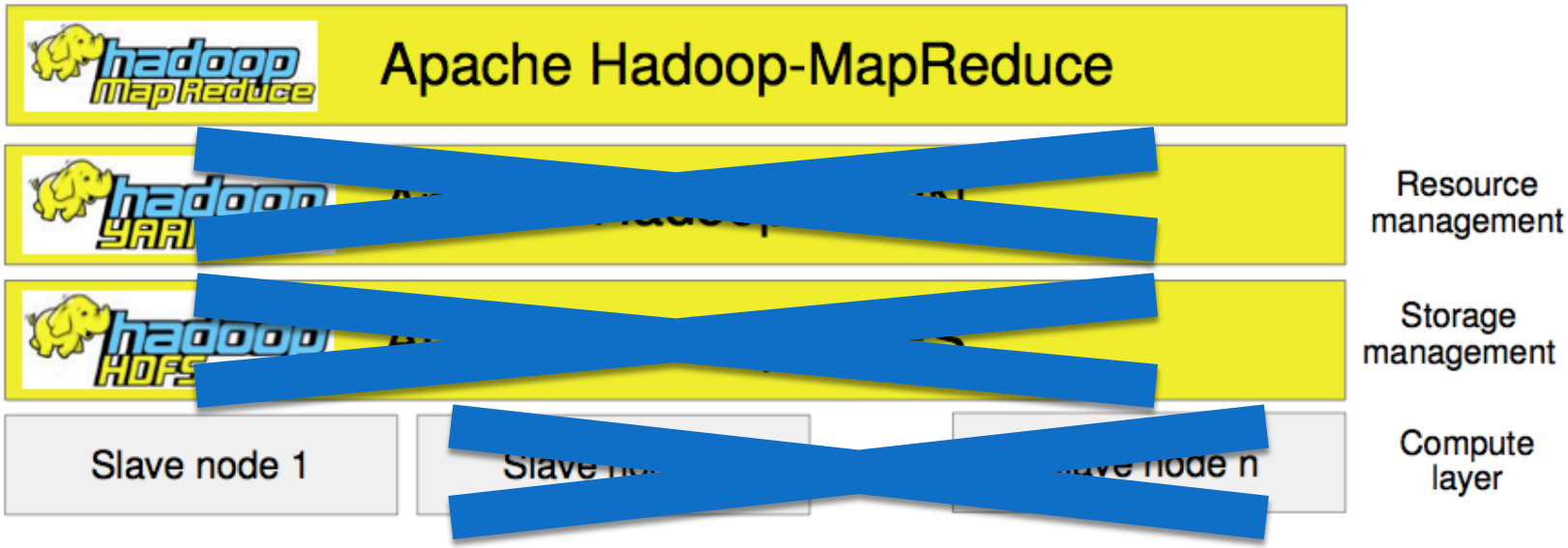
Instalar Hadoop - Standalone

- Ejemplo2
  - Ejecutamos un wordcount sobre ficheros de texto (por ejemplo los del directorio de prueba) pero con otra clase jar…
    - Pista utilizar:
    - `share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar`

```
SOL: bin/hadoop jar share/hadoop/mapreduce/hadoop-mapredu
examples-2.8.1.jar wordcount prueba salida1
```

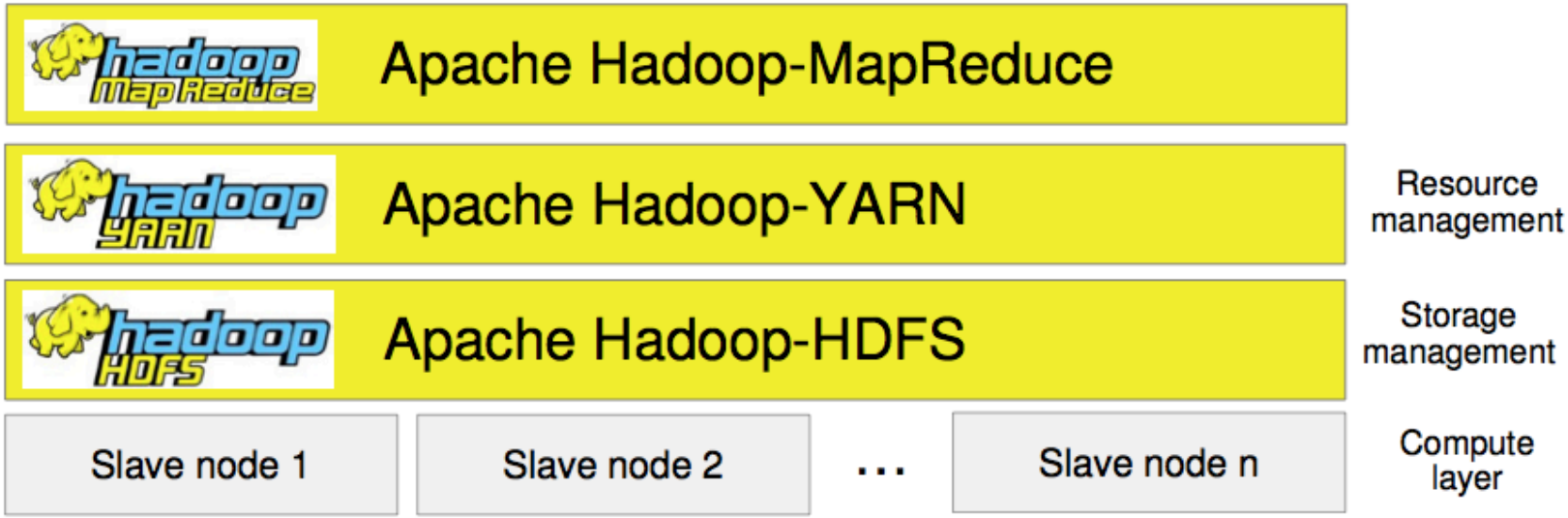
Instalación de Hadoop

- Modo *standalone instalado*



Instalación de Hadoop

- Este no el modo de ejecución de Hadoop del que estamos acostumbrados a hablar



Instalación de Hadoop

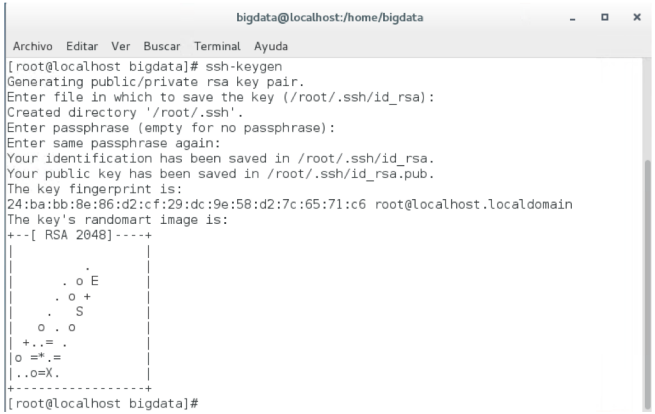
- Hadoop se puede instalar de tres maneras distintas:
  - Standalone
    - No se ejecutan demonios
    - Todo se ejecuta en una única Máquina Virtual de Java (MVJ)
    - No se usa HDFS
    - Adecuado para desarrollo y debug de aplicaciones

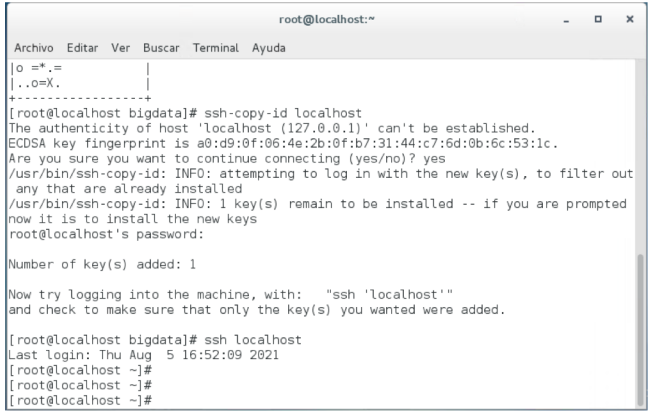
Instalación de Hadoop

- Hadoop se puede instalar de tres maneras distintas:
  - Pseudo-Distributed
    - Todos los demonios se ejecutan en la misma máquina
      - En su propia MVJ
    - Se emplea HDFS
    - Adecuado para similar un cluster en una sola máquina y para debug de programas antes de llevarlos a un “cluster real”

Instalar Hadoop – Pseudo-Distributed

- Antes de nada debemos configurar ssh para funcionar sin contraseña para conexiones a la misma máquina (localhost)
  - En esta configuración los diferentes procesos de Hadoop hacen uso de la red para conectarse aunque estén en la misma máquina
  - `> ssh-keygen` (pulso Intro a todo)
  - `> ssh-copy-id localhost` (esto vale para cualquier máquina)
- Comprobar que se puede conectar
  - `> ssh localhost`



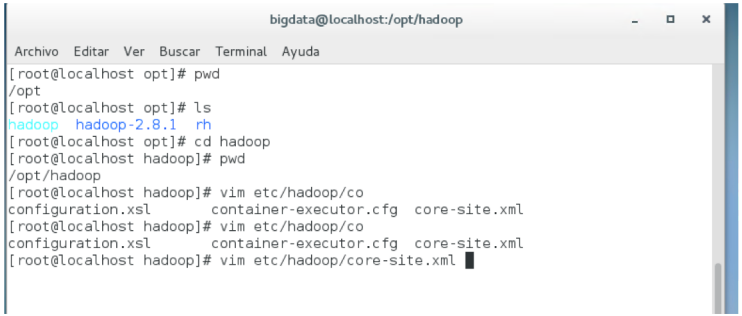


Instalar Hadoop – Pseudo-Distributed

- Añade la siguiente configuración a *etc/hadoop/core-site.xml*:

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

Importante: Se parte del directorio donde esta instalado Hadoop

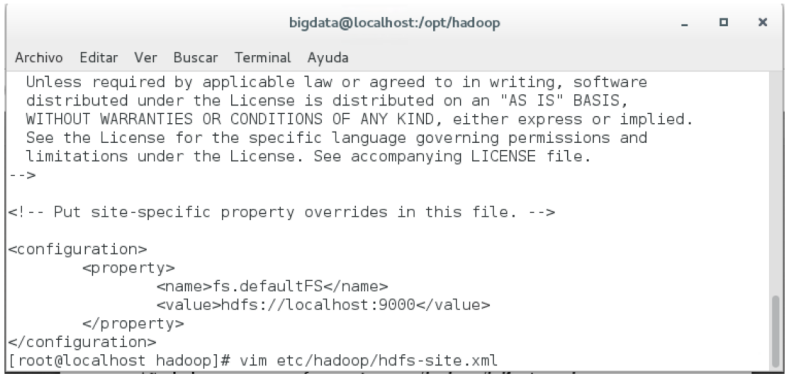


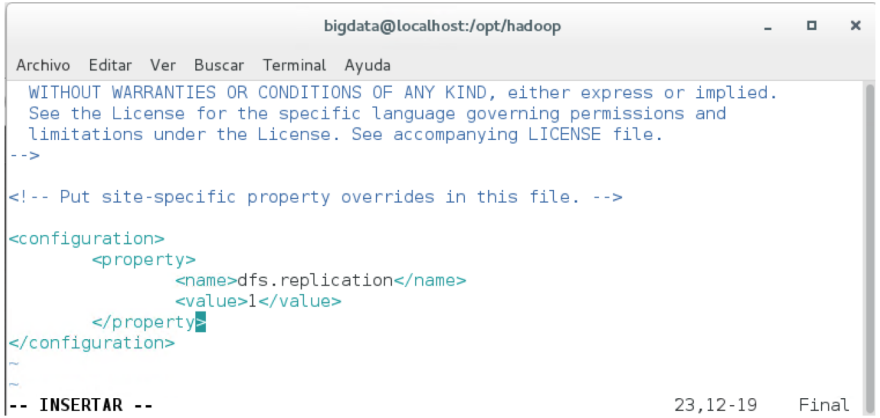
Instalar Hadoop – Pseudo-Distributed

- Añade la siguiente configuración a *etc/hadoop/hdfs-site.xml*:

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

Importante: Se parte del directorio donde esta instalado Hadoop

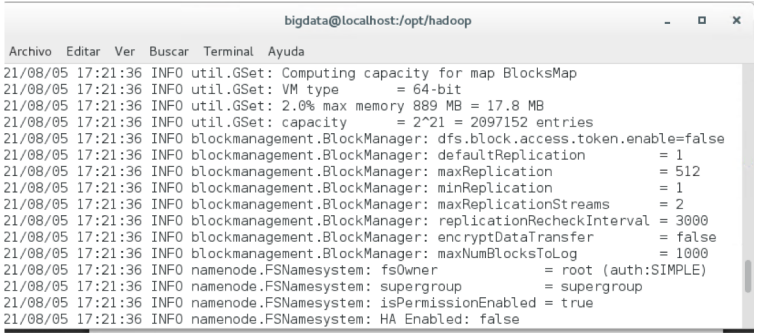


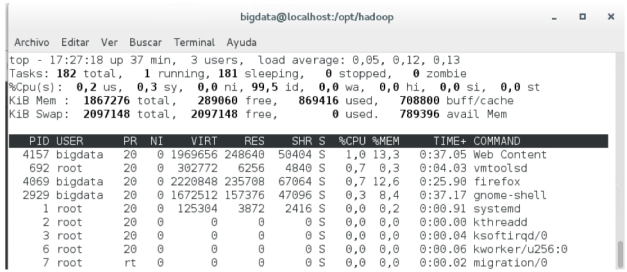


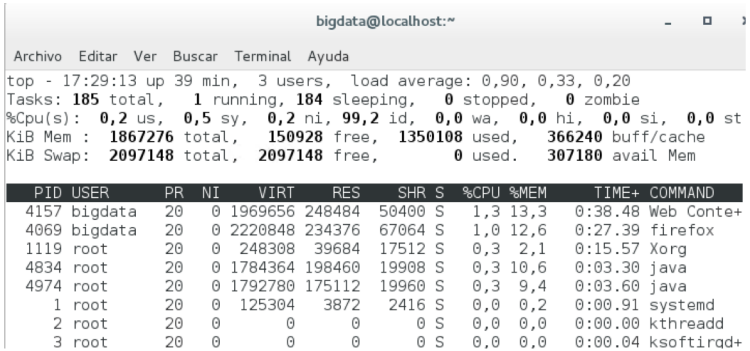
Instalar Hadoop – Pseudo-Distributed

- Formatear sistema de ficheros
  - `> bin/hdfs namenode -format`
- Iniciar el NameNode y DataNode
  - `> sbin/start-dfs.sh`

  (como root)

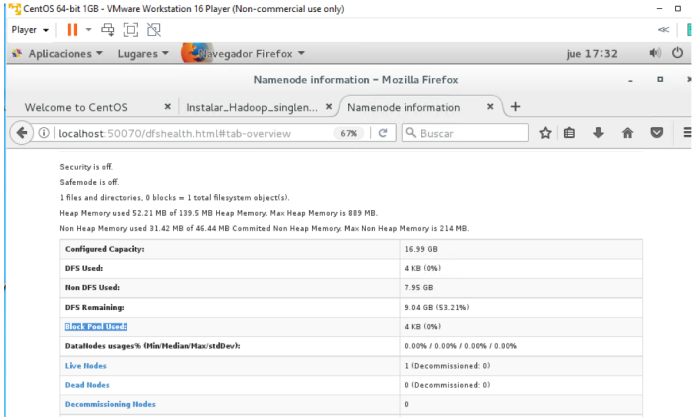


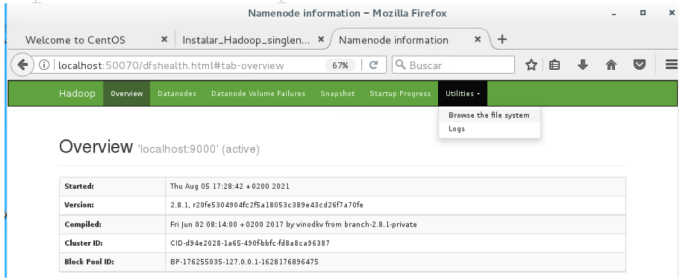


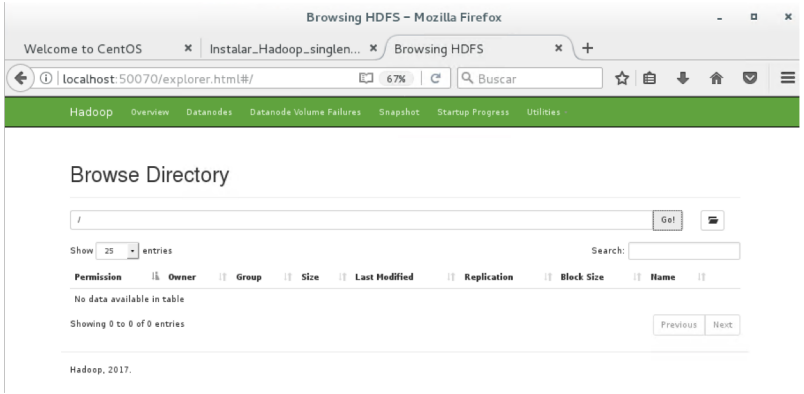


Instalar Hadoop – Pseudo-Distributed

- Deberías poder acceder a la web del NameNode en http://<ip>:50070
  - Importante: Si no accedes, es posible que el firewall esté activado
  - Desactivar firewall: `> service iptables stop`

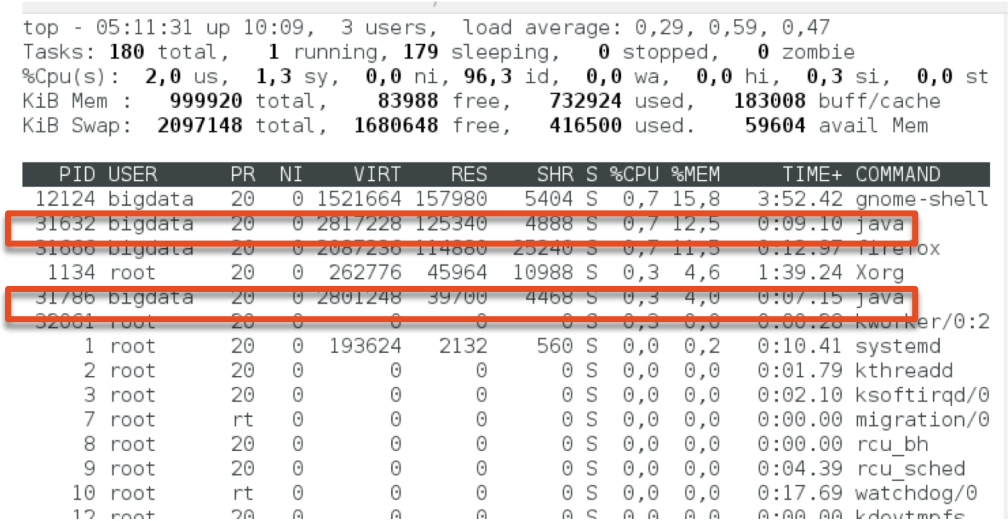




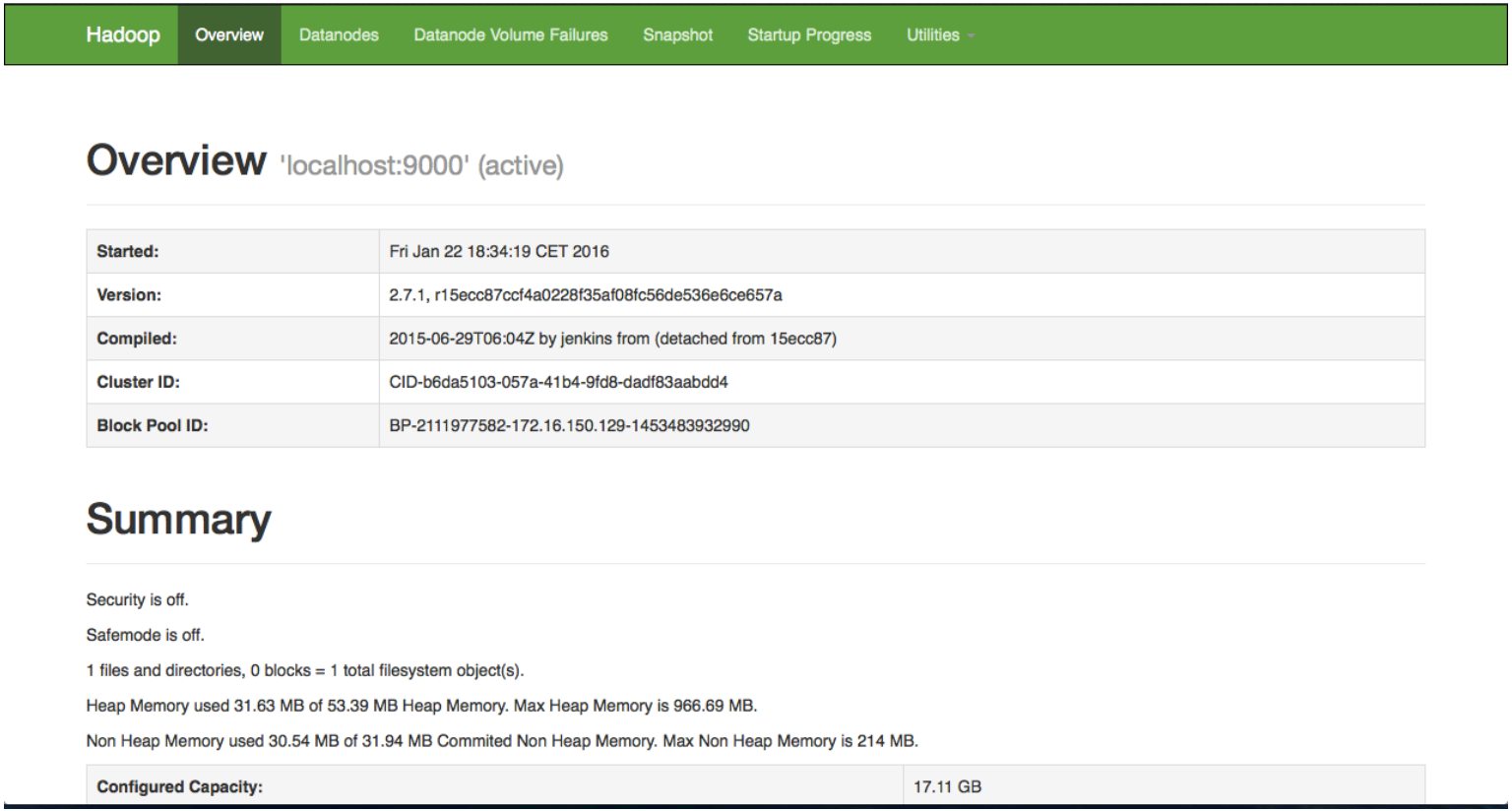


Instalar Hadoop – Pseudo-Distributed

- El servicio (demonio) de hdfs se empezará a ejecutar
  - `top`
  - `ps aux | grep dfs`
  - `free -m`



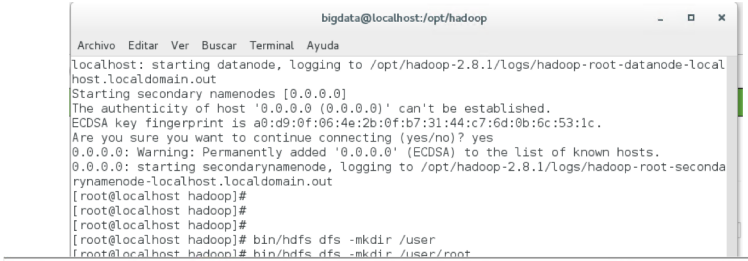
Instalar Hadoop – Pseudo-Distributed

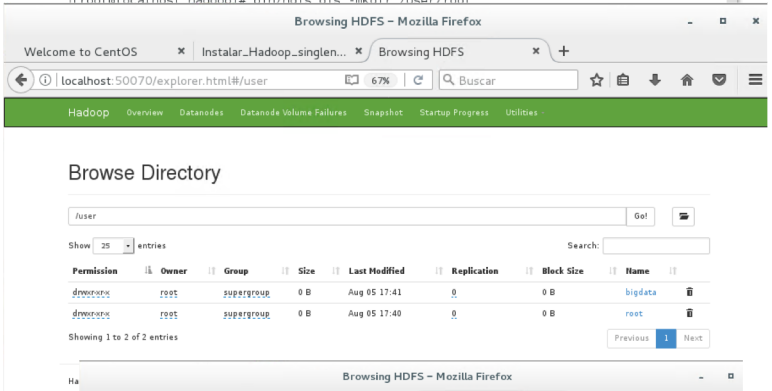


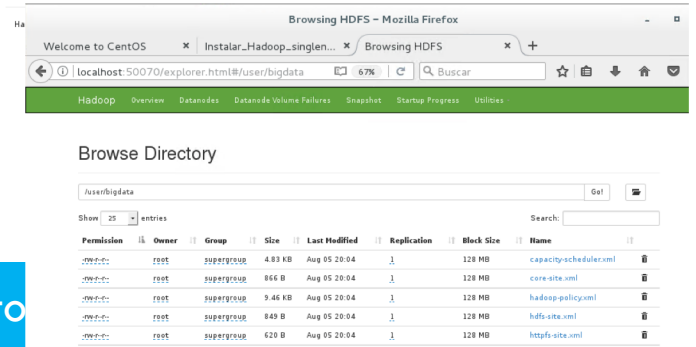
Instalar Hadoop – Pseudo-Distributed

- Ahora creamos los directorios de usuarios en HDFS
  - `> bin/hdfs dfs –mkdir /user`
  - `> bin/hdfs dfs –mkdir /user/root`
  - `> bin/hdfs dfs –mkdir /user/bigdata`
- Para probar
  - `> bin/hdfs dfs –put etc/hadoop/*.xml /user/bigdata`

  - ¿En qué directorio del HDFS se copian los ficheros?
  - ¿Qué ocurre si no hubiéramos creado el directorio?

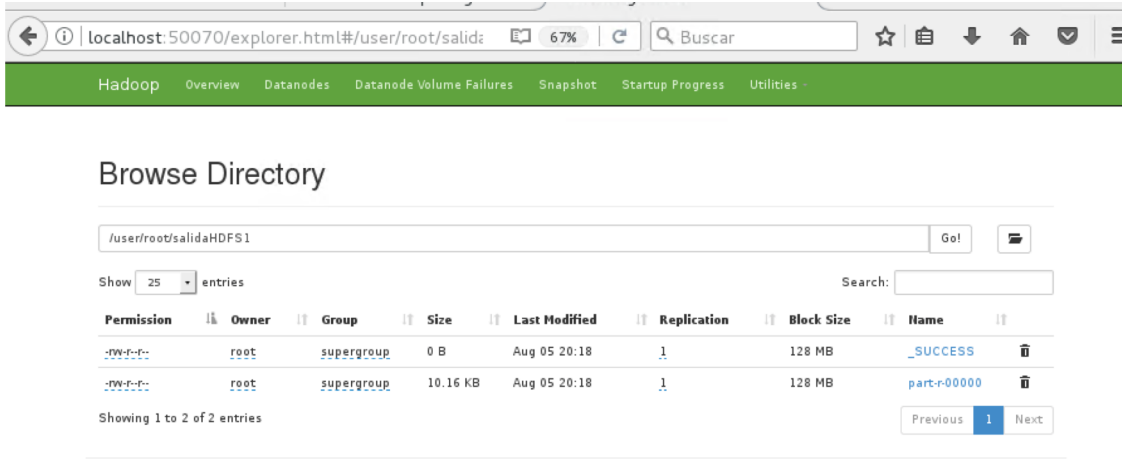






Instalar Hadoop – Pseudo-Distributed

- Para probar
  - `> bin/hdfs dfs -put etc/hadoop/*.xml /user/bigdata`
  - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.1.jar wordcount /user/root/prueba salidaHDFS1`
  - `> bin/hdfs dfs -ls /user/root/salidaHDFS1`
  - `> bin/hdfs dfs -cat /user/root/salidaHDFS1/part-r-00000`



- Para parar todos los servicios
  - `# sbin/stop-dfs.sh`

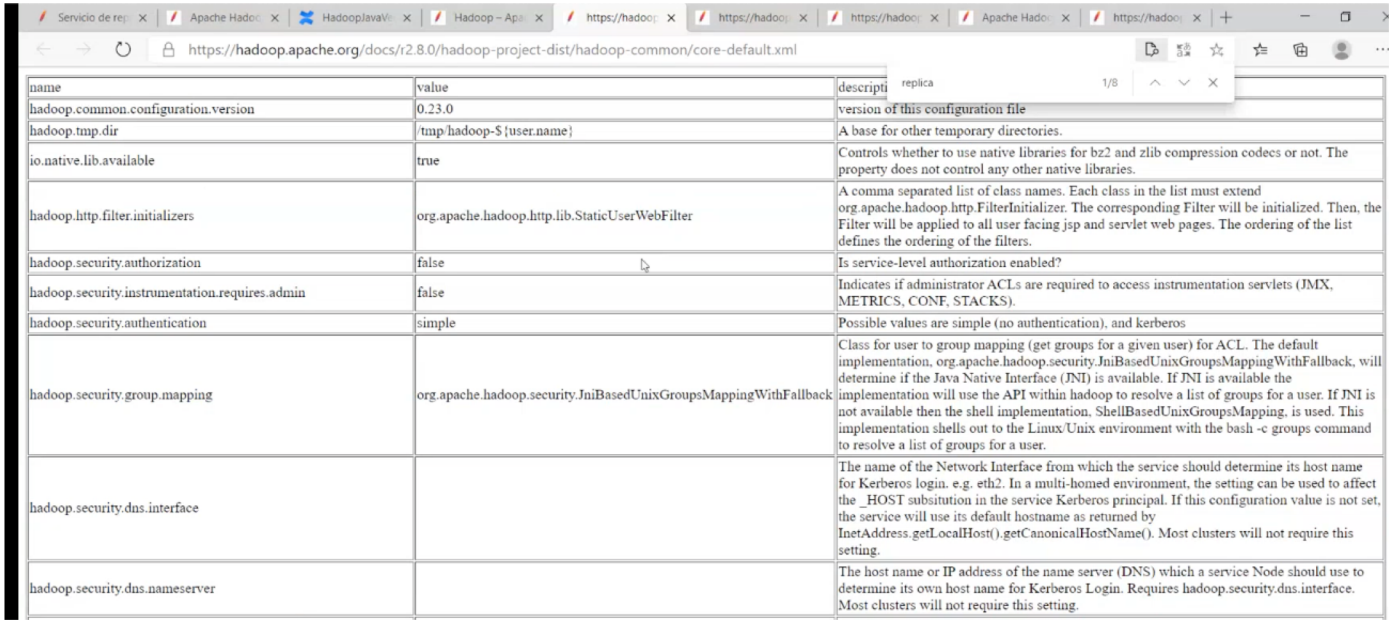
(No hacerlo de momento)

Instalar Hadoop – Pseudo-Distributed

- Para probar otro ejemplo y ver donde se crean los directorios
  - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'`
  - `> bin/hdfs dfs -cat output/*`

Instalar Hadoop – Pseudo-Distributed

- ¿Qué son todos estos ficheros de configuración que hemos editado? https://hadoop.apache.org/docs/r2.8.0/



https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/core-default.xml

name	value	descripti
hadoop.common.configuration.version	0.23.0	version of this configuration file
hadoop.tmp.dir	/tmp/hadoop-${user.name}	A base for other temporary directories.
io.native.lib.available	true	Controls whether to use native libraries for bz2 and zlib compression codecs or not. The property does not control any other native libraries.
hadoop.http.filter.initializers	org.apache.hadoop.http.lib.StaticUserWebFilter	A comma separated list of class names. Each class in the list must extend org.apache.hadoop.http.FilterInitializer. The corresponding Filter will be initialized. Then, the Filter will be applied to all user facing jsp and servlet web pages. The ordering of the list defines the ordering of the filters.
hadoop.security.authorization	false	Is service-level authorization enabled?
hadoop.security.instrumentation.requires.admin	false	Indicates if administrator ACLs are required to access instrumentation servlets (JMX, METRICS, CONF, STACKS).
hadoop.security.authentication	simple	Possible values are simple (no authentication), and kerberos
hadoop.security.group.mapping	org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback	Class for user to group mapping (get groups for a given user) for ACL. The default implementation, org.apache.hadoop.security.JniBasedUnixGroupsMappingWithFallback, will determine if the Java Native Interface (JNI) is available. If JNI is available the implementation will use the API within hadoop to resolve a list of groups for a user. If JNI is not available then the shell implementation, ShellBasedUnixGroupsMapping, is used. This implementation shells out to the Linux/Unix environment with the bash -c groups command to resolve a list of groups for a user.
hadoop.security.dns.interface		The name of the Network Interface from which the service should determine its host name for Kerberos login. e.g. eth2. In a multi-homed environment, the setting can be used to affect the _HOST subsitution in the service Kerberos principal. If this configuration value is not set, the service will use its default hostname as returned by InetAddress.getLocalHost().getCanonicalHostName(). Most clusters will not require this setting.
hadoop.security.dns.nameserver		The host name or IP address of the name server (DNS) which a service Node should use to determine its own host name for Kerberos Login. Requires hadoop.security.dns.interface. Most clusters will not require this setting.

Instalar Hadoop – Pseudo-Distributed

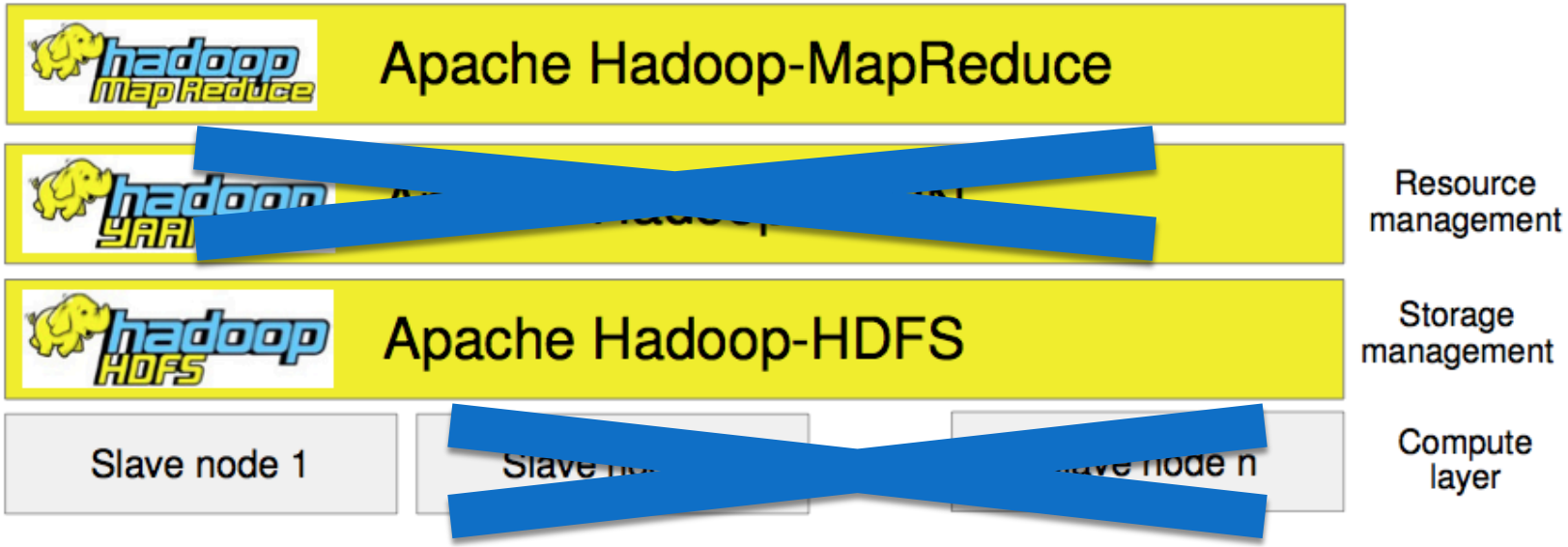
- Un pequeño resumen (http://www.big-data.tips/hadoop-configuration)
  - ***The Hadoop configuration file core-site.xml contains pieces of information about the particular 'Hadoop site' itself.*** *This includes the hostname and port number used for this particular Hadoop instance. Other optional information is the memory allocated for the file system. There can be also memory limits for storing data or more detailed configurations such as the size of read and wite buffers.*
  - Para investigar todos los parámetros disponibles:
    - https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-common/core-default.xml

Instalar Hadoop – Pseudo-Distributed

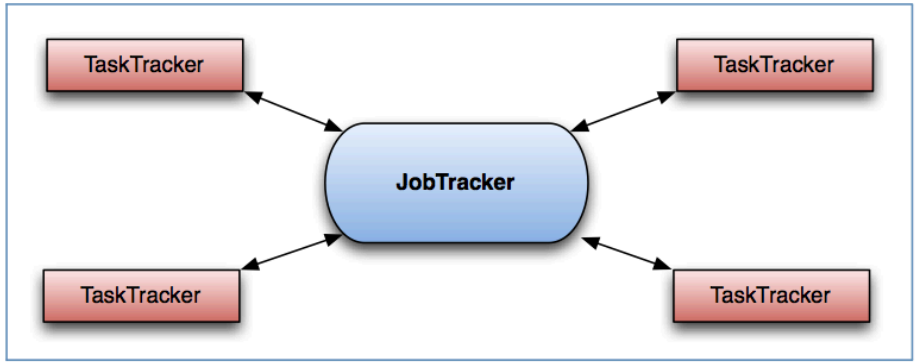
- Un pequeño resumen (http://www.big-data.tips/hadoop-configuration)
  - ***The Hadoop configuration file hdfs-site.xml file contains information about the Hadoop Distributed File System (HDFS) that is part of the Hadoop distribution.*** *It includes the value of 'replication' and the path to the 'namenode' as well as the paths to 'datanodes' based on the local file systems. This is needed in order to tell HDFS a concrete place where data in the Hadoop infrastructure is stored. Below is an example but needs to be configured according to your file system structure depending on the Hadoop infrastructure.*
  - Para investigar todos los parámetros disponibles:
    - https://hadoop.apache.org/docs/r2.8.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml

Instalación de Hadoop

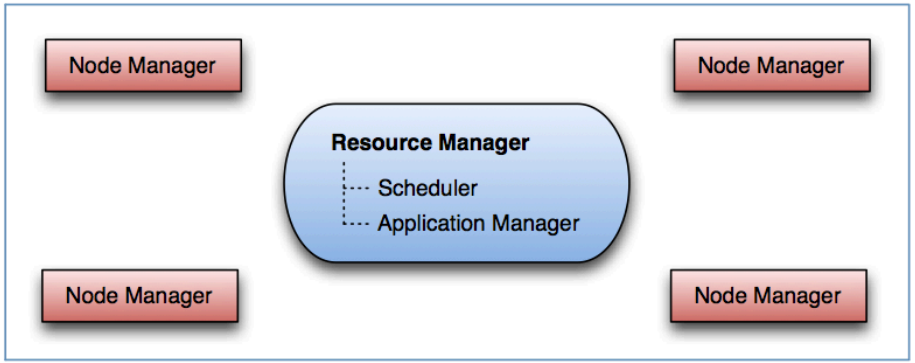
- Modo *pseudo-distributed* (hasta este punto)



MRv1 vs. MRv2 (Yarn)



- Un JobTracker (master) por cluster
- Cada esclavo ejecuta un TaskTracker



- Single Resource Manager por cluster
- Cada esclavo ejecuta un Node Manager

Imágenes obtenidas de Cloudera (http://www.cloudera.com

Instalar Hadoop – Pseudo-Distributed con YARN

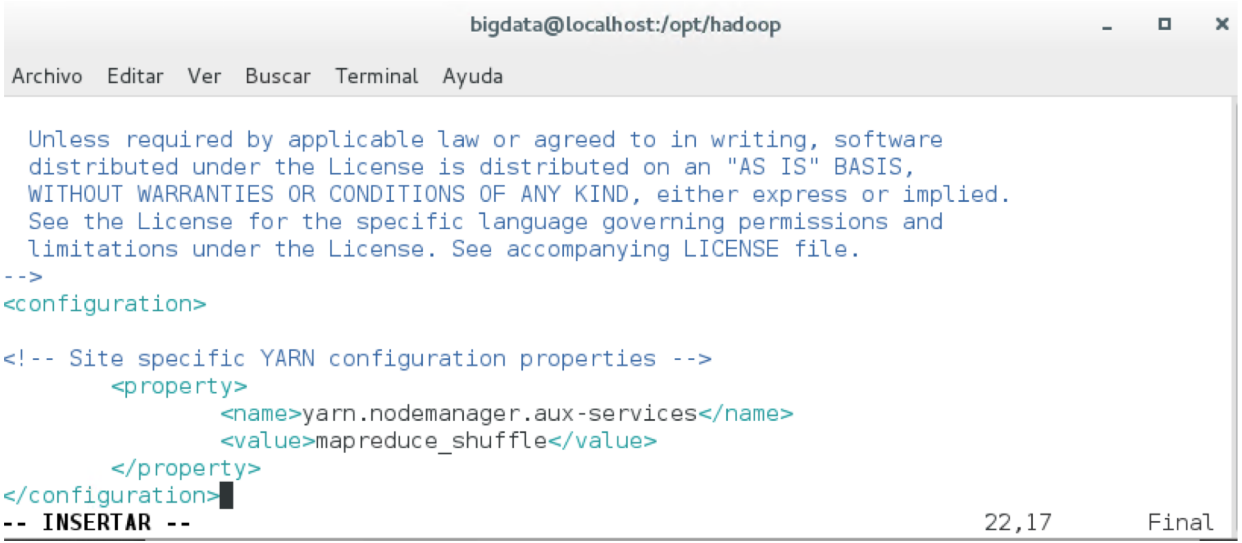
- La configuración anterior ejecuta tareas MapReduce usando MRv1. Sin embargo, es posible usar MRv2 (o YARN)
- Para ello es necesario ejecutar el servicio ResourceManager y NodeManager
- Partiendo de los pasos realizados antes… tras parar los servicios
- Añade la siguiente configuración a *etc/hadoop/mapred-site.xml*

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

Instalar Hadoop – Pseudo-Distributed con YARN

- Añade la siguiente configuración a *etc/hadoop/yarn-site.xml*

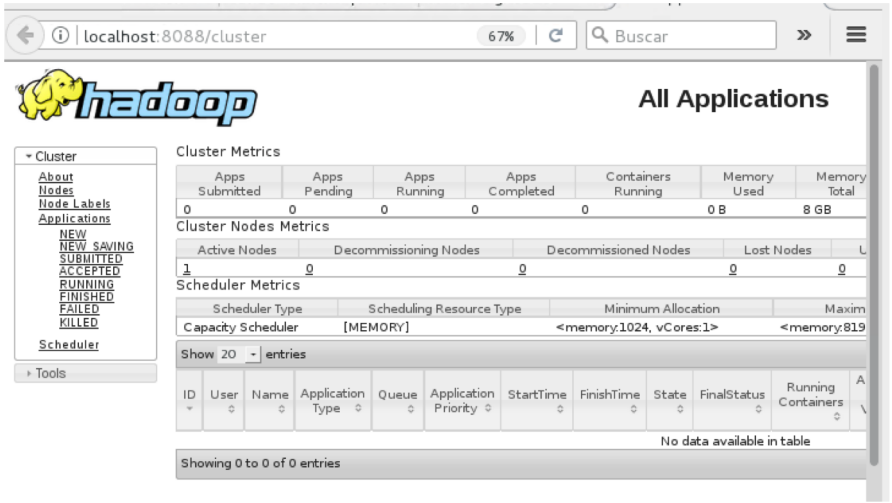
```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```



Instalar Hadoop – Pseudo-Distributed con YARN

- Iniciar el ResourceManager y NodeManager
  - `> sbin/start-yarn.sh` (para finalizar `sbin/stop-yarn.sh`)
- Deberías poder acceder a la web del ResourceManager en http://<ip>:8088

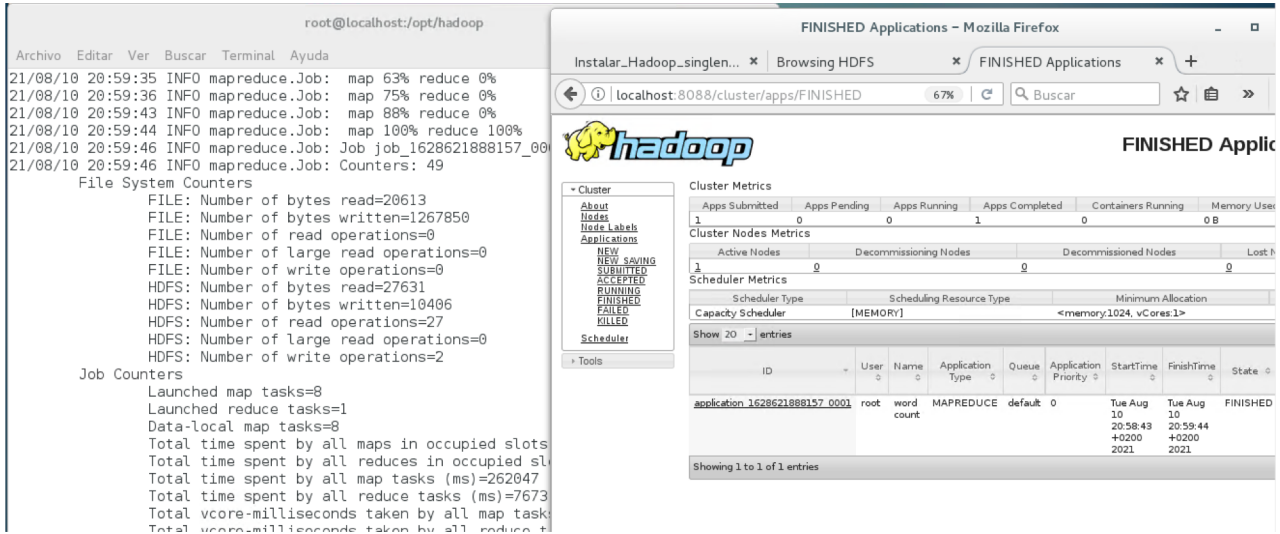
```
[root@localhost hadoop]# vim etc/hadoop/yarn-site.xml
[root@localhost hadoop]# sbin/st
start-all.cmd        start-dfs.sh         stop-all.cmd         stop-dfs.sh
start-all.sh         start-secure-dns.sh  stop-all.sh          stop-secure-dns.sh
start-balancer.sh    start-yarn.cmd       stop-balancer.sh     stop-yarn.cmd
start-dfs.cmd        start-yarn.sh        stop-dfs.cmd         stop-yarn.sh
[root@localhost hadoop]# sbin/start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /opt/hadoop-2.8.1/logs/yarn-bigdata-resourcemanager-l
ocalhost.localdomain.out
localhost: starting nodemanager, logging to /opt/hadoop-2.8.1/logs/yarn-root-nodemanager-l
ocalhost.localdomain.out
[root@localhost hadoop]#
```



Se puede bloquear
si memoria > 2GB

Instalar Hadoop – Pseudo-Distributed con YARN

- Iniciar el ResourceManager y NodeManager
  - `> sbin/start-yarn.sh` (para finalizar `sbin/stop-yarn.sh`)
- Deberías poder acceder a la web del ResourceManager en http://<ip>:8088



Se puede bloquear o ir lento si memoria <= 2GB

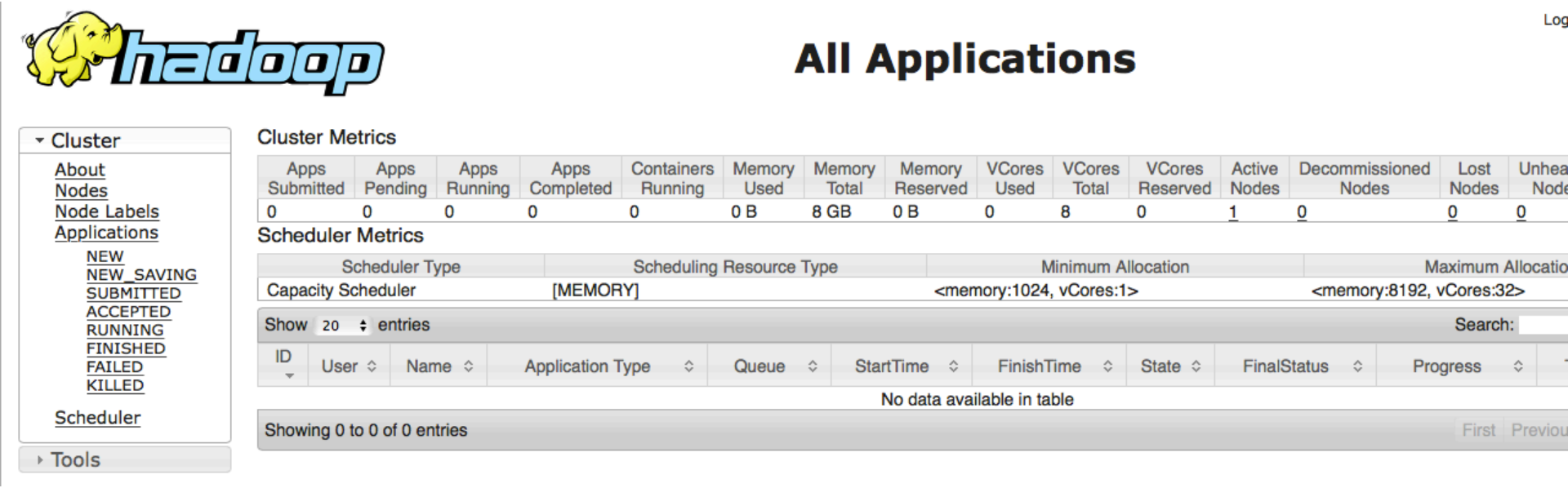
Instalar Hadoop – Pseudo-Distributed

- Un pequeño resumen (http://www.big-data.tips/hadoop-configuration)
  - ***The Hadoop configuration file mapred-site.xml specifies which map-reduce framework is used that is in our example here YARN.*** *Any Hadoop distribution contains a template of the 'mapred-site.xml' file named 'mapred-site.xml.template'. We first copy this template file to the correct name and then add lines as shown below.*
  - Para investigar todos los parámetros disponibles:
    - https://hadoop.apache.org/docs/r2.8.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml

Instalar Hadoop – Pseudo-Distributed

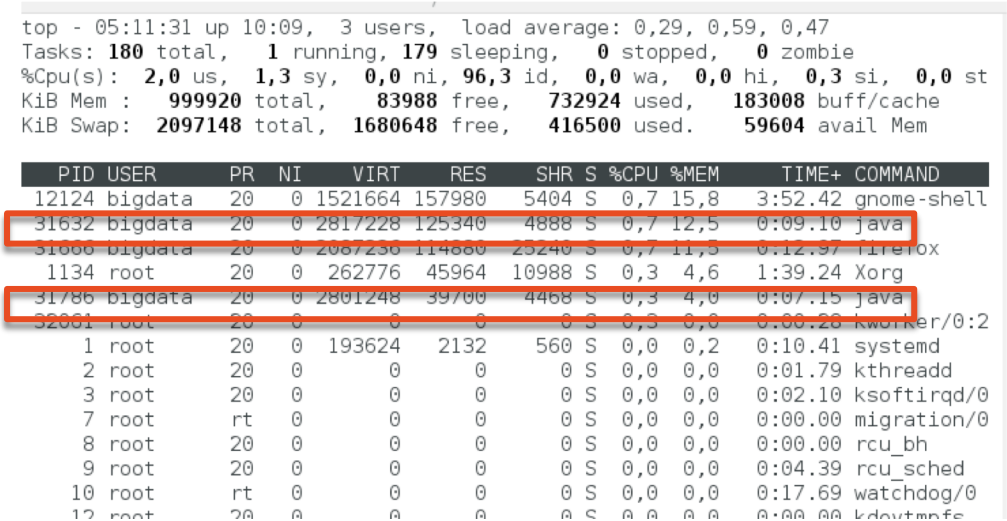
- Un pequeño resumen (http://www.big-data.tips/hadoop-configuration)
  - ***The Hadoop configuration file yarn-site.xml is used for the Hadoop scheduling system 'Yet Another Resource Negoatiator (YARN)'.*** *This component is also an integral part of Hadoop alongside HDFS.*
  - Para investigar todos los parámetros disponibles:
    - https://hadoop.apache.org/docs/r2.8.0/hadoop-yarn/hadoop-yarn-common/yarn-default.xml

Instalar Hadoop – Pseudo-Distributed con YARN



Instalar Hadoop – Pseudo-Distributed

- Añadimos más servicios ejecutándose en segundo plano
  - `top`
  - `ps aux | grep dfs`
  - `free -m`

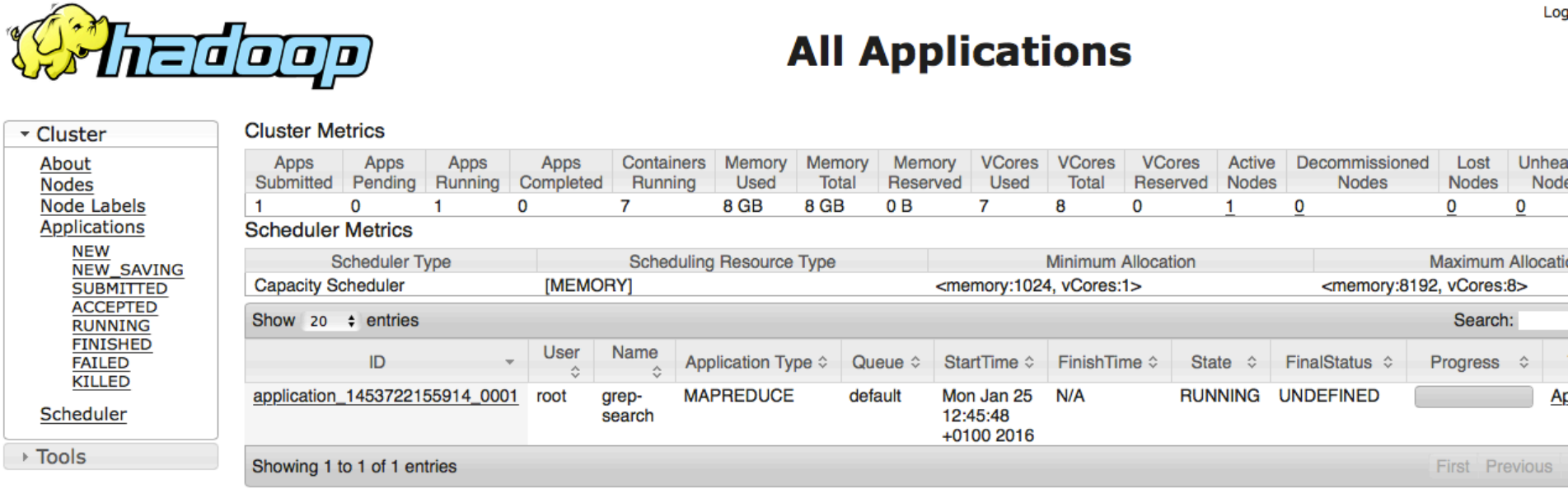


Instalar Hadoop – Pseudo-Distributed con YARN

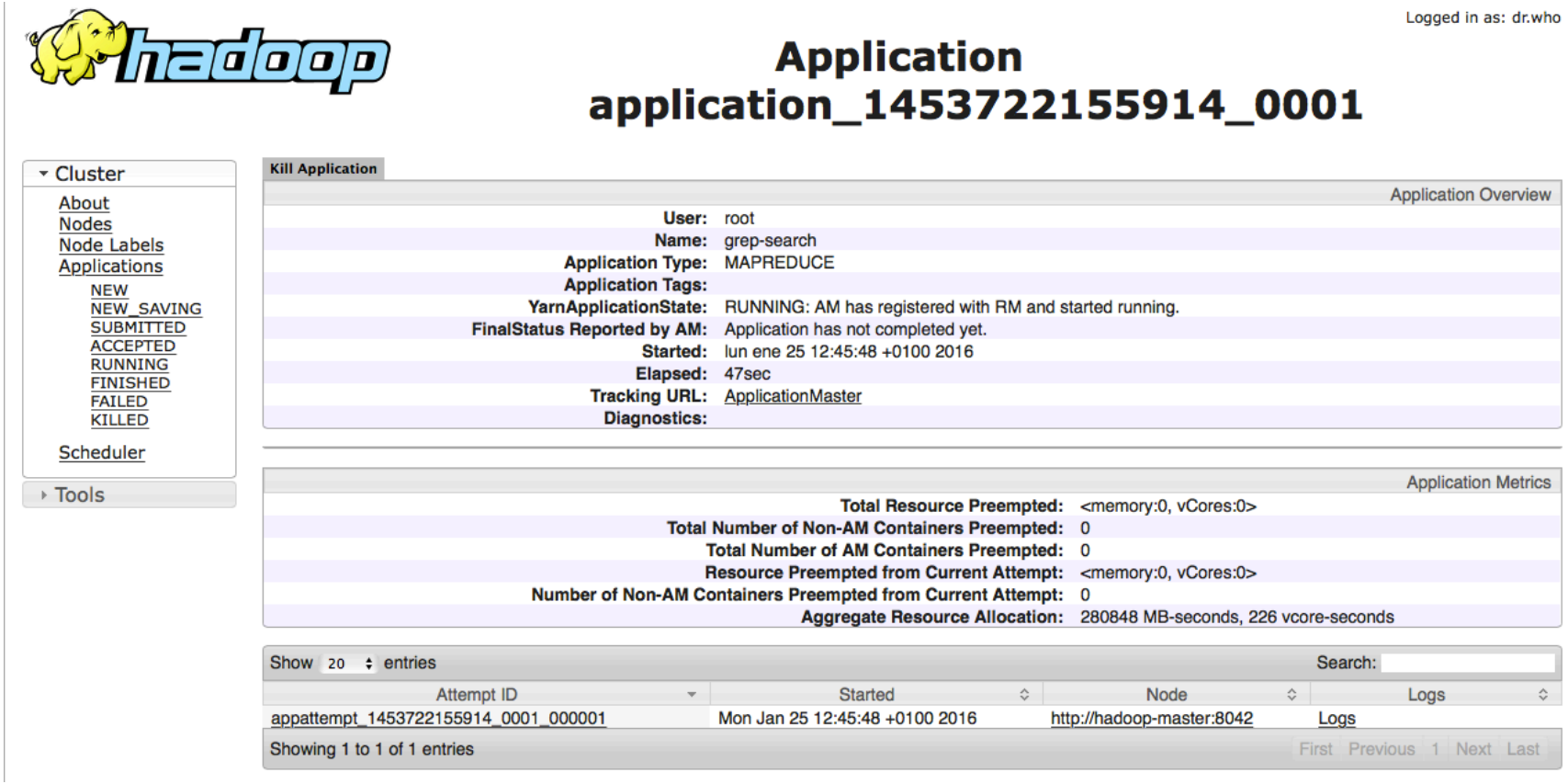
- Ahora podrías volver a lanzar la aplicación MapReduce de prueba de nuevo y ver su evolución en la web del Resource Manager
  - `> bin/hdfs dfs -rm -r output`
  - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar grep input output 'dfs[a-z.]+'`
  - `> bin/hdfs dfs -cat output/*`
- O si echas de menos el 'wordcount'…
  - `> bin/hdfs dfs -rm -r output`
  - `> bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.1.jar wordcount input output`
  - `> bin/hdfs dfs -cat output/*`

Instalar Hadoop – Pseudo-Distributed con YARN

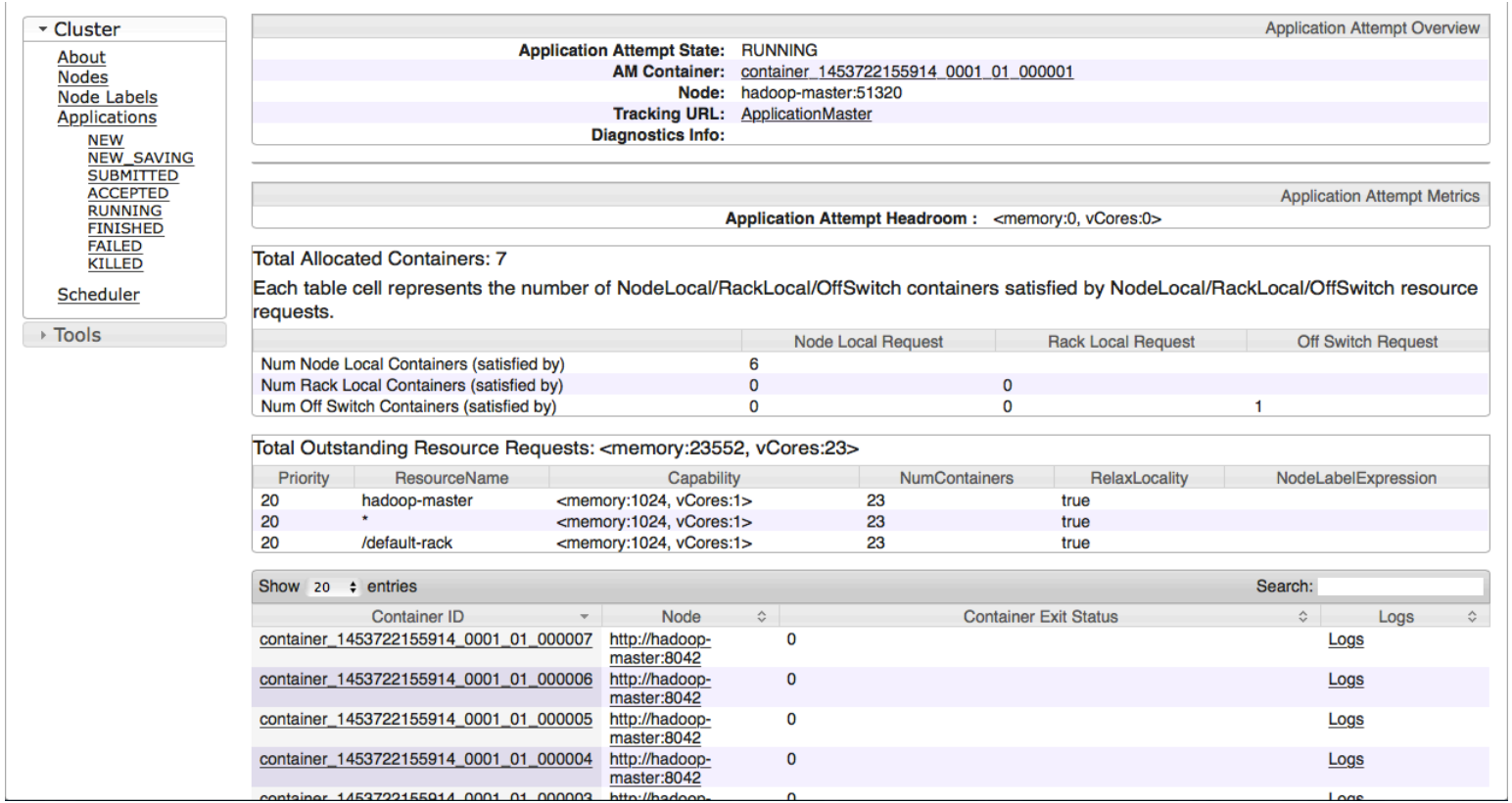
- La aplicación debe aparecer en la web…



Instalar Hadoop – Pseudo-Distributed con YARN

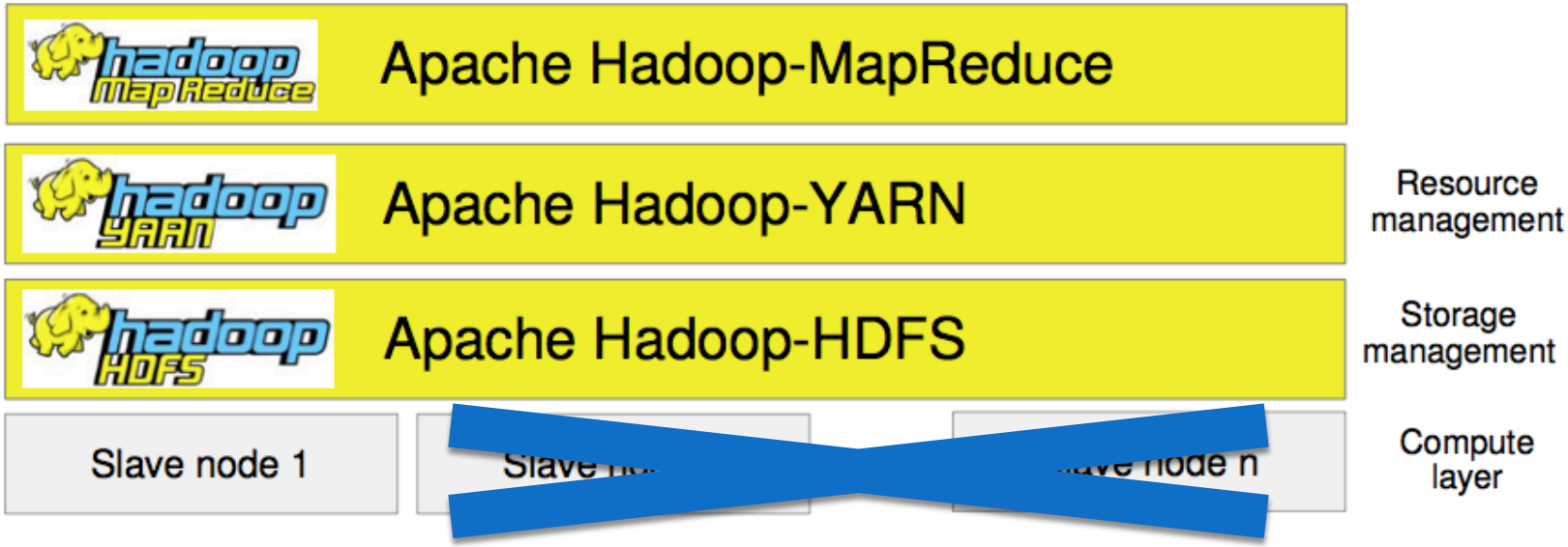


Instalar Hadoop – Pseudo-Distributed con YARN



Instalación de Hadoop

- Modo *pseudo-distributed* (ahora)



Instalación de Hadoop

- Instalación *Fully Distributed*
  - Los demonios de Hadop se ejecutan en un cluster de máquinas
  - HDFS se emplea para distribuir datos entre todos los nodos
  - A menos que se emplee un cluster pequeño (menos de 10 o 20 nodos), el NameNode y JobTracker deben ejecutarse en nodos dedicados
    - Para pequeños clusters pueden ejecutarse en el mismo nodo