

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ APACHE
HADOOP ΜΕ APACHE SPARK ΓΙΑ
ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΓΑΛΩΝ
ΔΕΔΟΜΕΝΩΝ

Δημητριάδης Πρόδρομος (1359)
Νάστος Βασίλης (1525)

ΠΕΡΙΕΧΟΜΕΝΑ

BIG DATA

ΕΡΓΑΛΕΙΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ
ΔΕΔΟΜΕΝΩΝ

ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

ΣΥΓΚΡΙΣΗ HADOOP ΜΕ SPARK

ΕΦΑΡΜΟΓΕΣ



BIG DATA

ΟΡΙΣΜΟΣ



- Σύνολα δομημένων, ημιδομημένων και αδόμητων δεδομένων με τεράστια μεγέθη, τα οποία είναι πολύ μεγάλα για επεξεργασία, αποθήκευση, διαχείριση και ανάλυση από ένα μόνο υπολογιστικό σύστημα.

ΤΑ 3 V



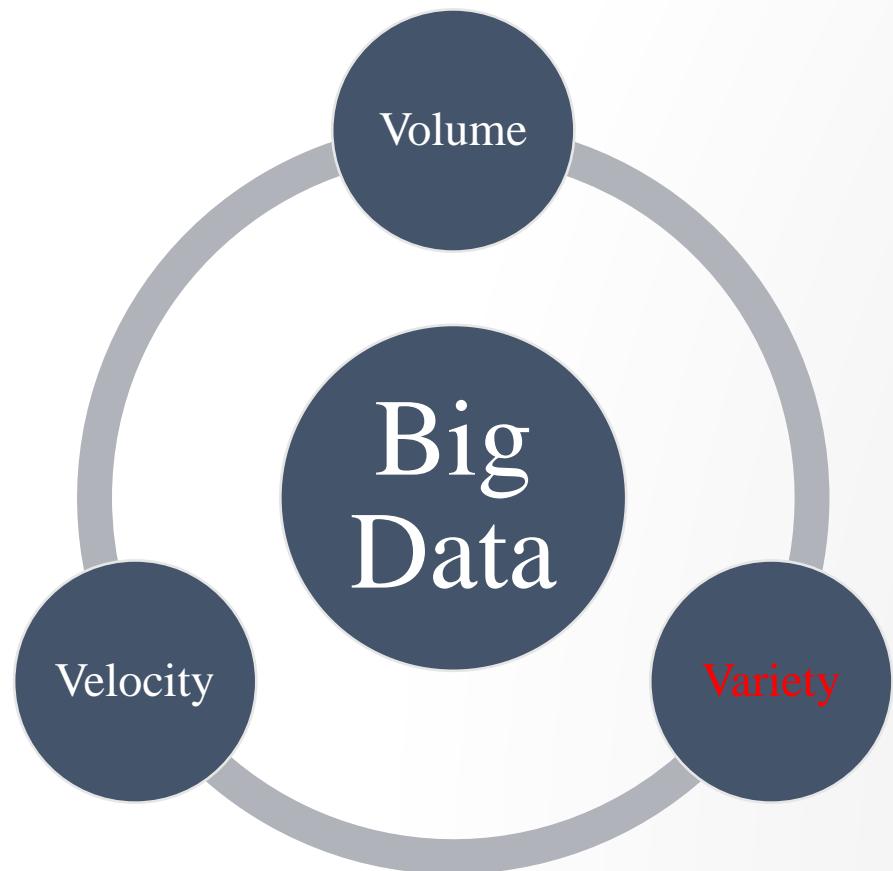
- Αποθήκευση και επεξεργασία Terabytes και Petabytes δεδομένων.
- Παραδείγματα:
 - Facebook
 - YouTube
 - Twitter
 - Instagram
 - Netflix

TA 3 V



- Υψηλός αριθμός με τον οποίο παράγονται νέα δεδομένα.
- Παραδείγματα:
 - Facebook
 - Αισθητήρες

ΤΑ 3 V



- Συγκέντρωση δεδομένων από διάφορες πηγές και σε διάφορες μορφές.
- Ορισμένες μορφές δεδομένων:
 - Φωτογραφίες
 - Δεδομένα αισθητήρων
 - Tweets
 - Κρυπτογραφημένα πακέτα

AKOMA 2 V

Veracity

Value

Διασφάλιση
ποιότητας,
αξιοπιστίας και
ακρίβειας των
δεδομένων.

Οικονομική αξία
των δεδομένων και
στο πόσο χρήσιμα
είναι στη λήψη
αποφάσεων.

ΤΕΧΝΟΛΟΓΙΕΣ



Ανάλυση, επεξεργασία
και εξαγωγή
πληροφοριών από ένα
περίπλοκο και μεγάλο
σύνολο δεδομένων.

Ανάλυση τεράστιων
δεδομένων σε
πραγματικό χρόνο.

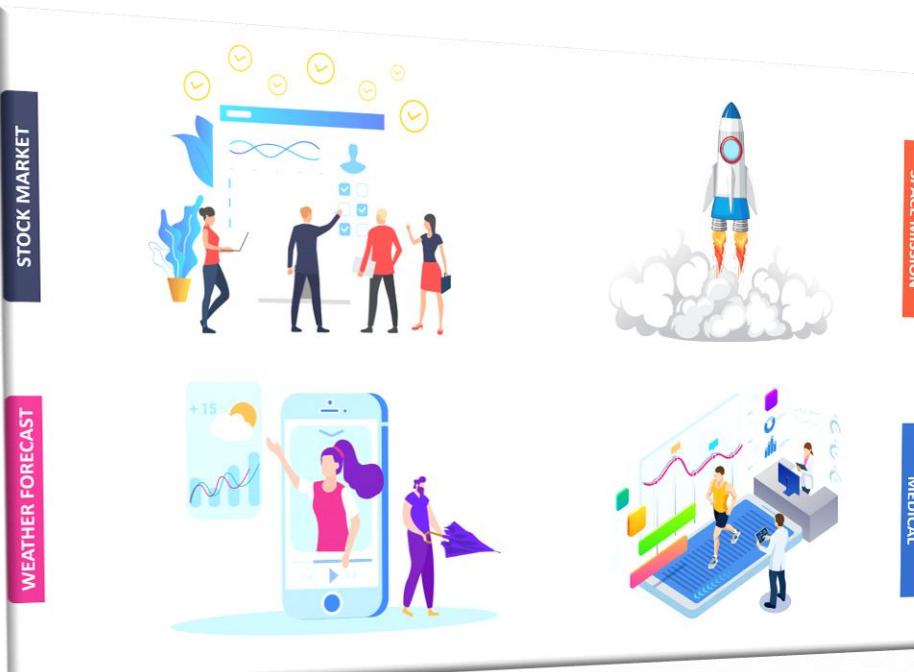
ΤΕΧΝΟΛΟΓΙΕΣ



Τεχνολογίες Μεγάλων Δεδομένων που χρησιμοποιούνται στην πράξη:

- Καθημερινά δεδομένα που δημιουργούμε

ΤΕΧΝΟΛΟΓΙΕΣ



Αναλυτική Μεγάλων Δεδομένων:

- Κρίσιμες επιχειρηματικές αποφάσεις που λαμβάνονται σε πραγματικό χρόνο με την ανάλυση Μεγάλων Δεδομένων

ΤΕΧΝΟΛΟΓΙΕΣ

+

•

○



ΕΡΓΑΛΕΙΑ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ

APACHE HADOOP



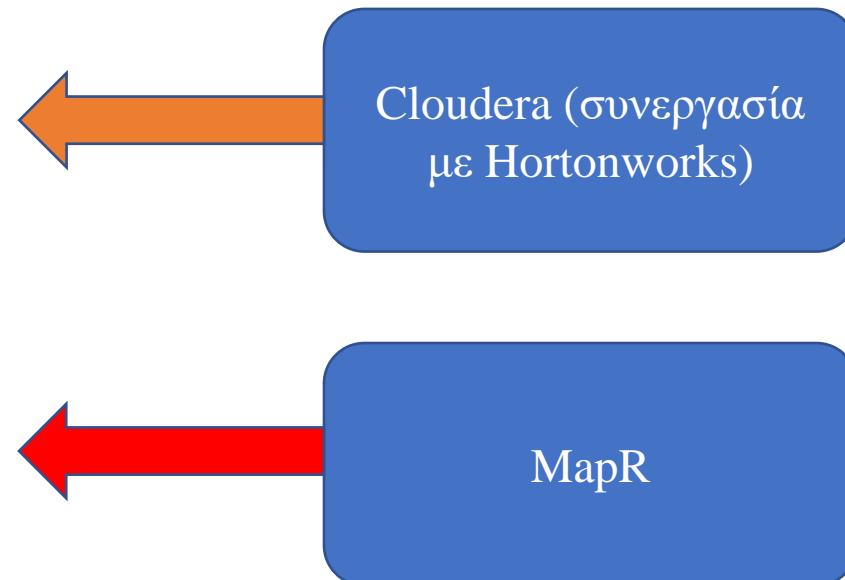
APACHE HADOOP

- ❑ Framework ανοιχτού κώδικα
- ❑ Σχεδιασμένο για να κάνει scale up από μεμονωμένους διακομιστές σε χιλιάδες μηχανήματα
- ❑ Εντοπίζει και χειρίζεται αποτυχίες στο επίπεδο των εφαρμογών.
- ❑ Αποτελείται από:
 - ❑ Hadoop Common
 - ❑ HDFS (Hadoop Distributed File System)
 - ❑ Hadoop YARN
 - ❑ Hadoop MapReduce

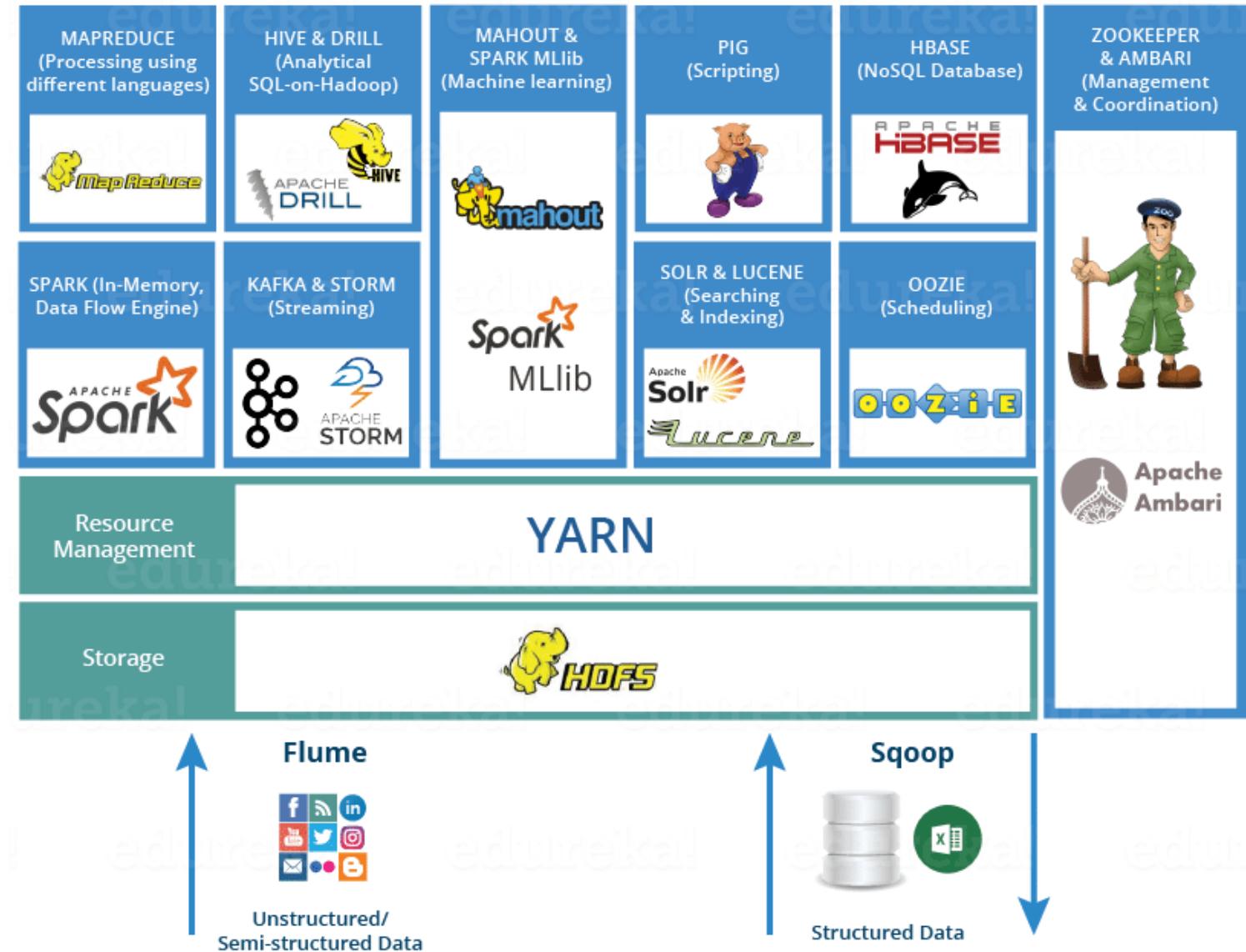
ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΤΡΟΠΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Ανεπτυγμένο κυρίως σε γλώσσα προγραμματισμού Java
- Αρχεία παραμετροποίησης σε μορφή XML
- Εγκατάσταση σε λογισμικό Linux, Windows, macOS (συνηθέστερος τρόπος σε Linux, ή στα υπόλοιπα μέσω VM)

CLOUDERA



ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

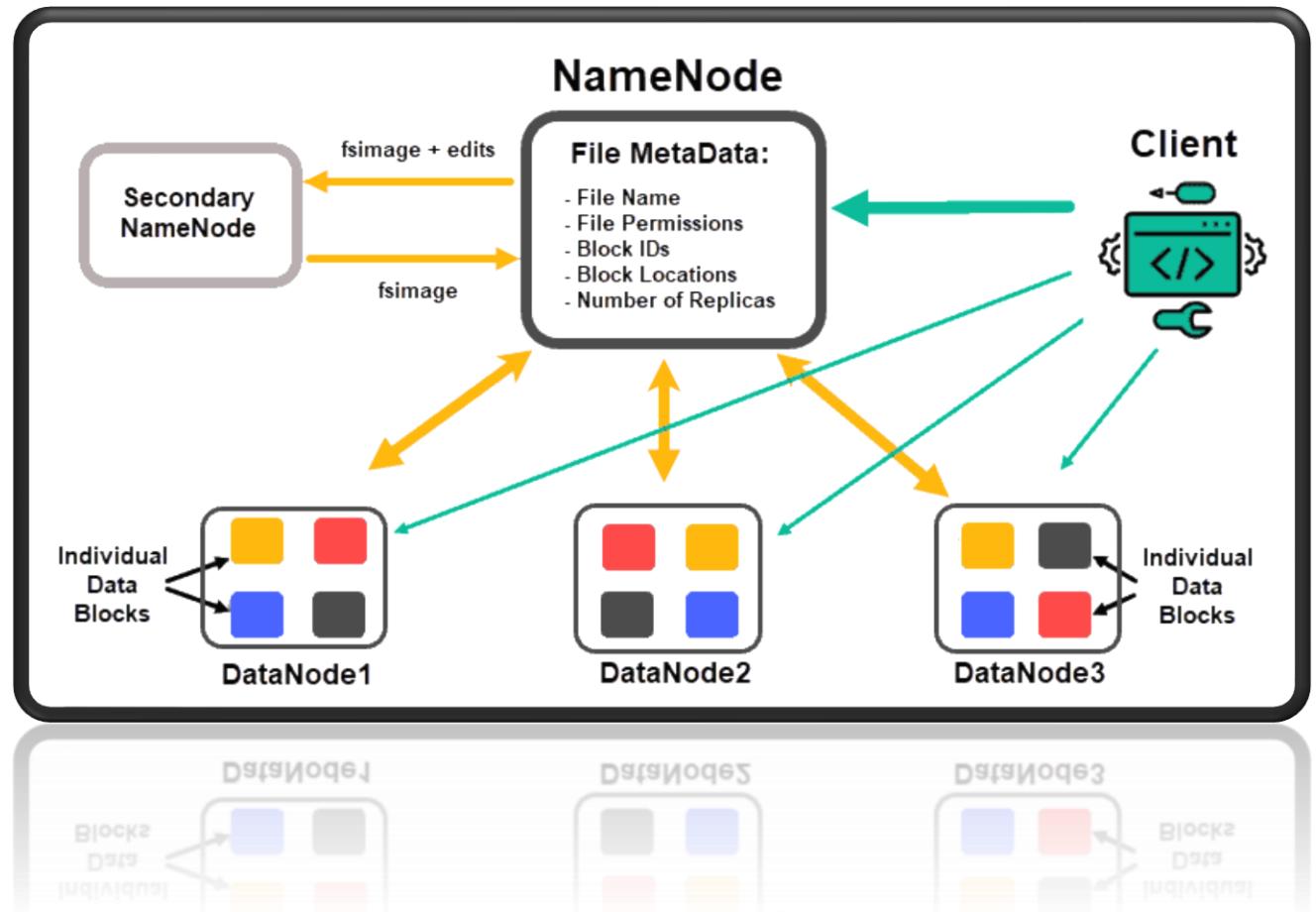


ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- HDFS (Hadoop Distributed File System):
 - Σύστημα κατανεμημένων αρχείων, σχεδιασμένο για να τρέχει πάνω από το σύστημα ενός Linux συστήματος.
 - Ανοχή σε σφάλματα υλικού.
 - Σχεδιασμένο για την ανάπτυξη σε υλικό χαμηλού κόστους.
 - Βοηθά στην αποθήκευση των δεδομένων σε διάφορους κόμβους και στη διατήρηση του αρχείου καταγραφής σχετικά με τα αποθηκευμένα δεδομένα.

ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- HDFS Αρχιτεκτονική:
 - NameNode
 - Κύριος κόμβος
 - Δεν αποθηκεύει τα πραγματικά δεδομένα
 - Περιέχει μεταδεδομένα
 - Secondary NameNode
 - Παρακολουθεί τις αλλαγές που πραγματοποιούνται στο NameNode
 - Παρέχει στο NameNode γρήγορη εκκίνηση
 - DataNode
 - Αποθηκεύονται όλα τα δεδομένα
 - Βασικό υλικό (επιτραπέζιοι υπολογιστές)
 - Προσφέρει αποδοτικές λύσεις στο Hadoop



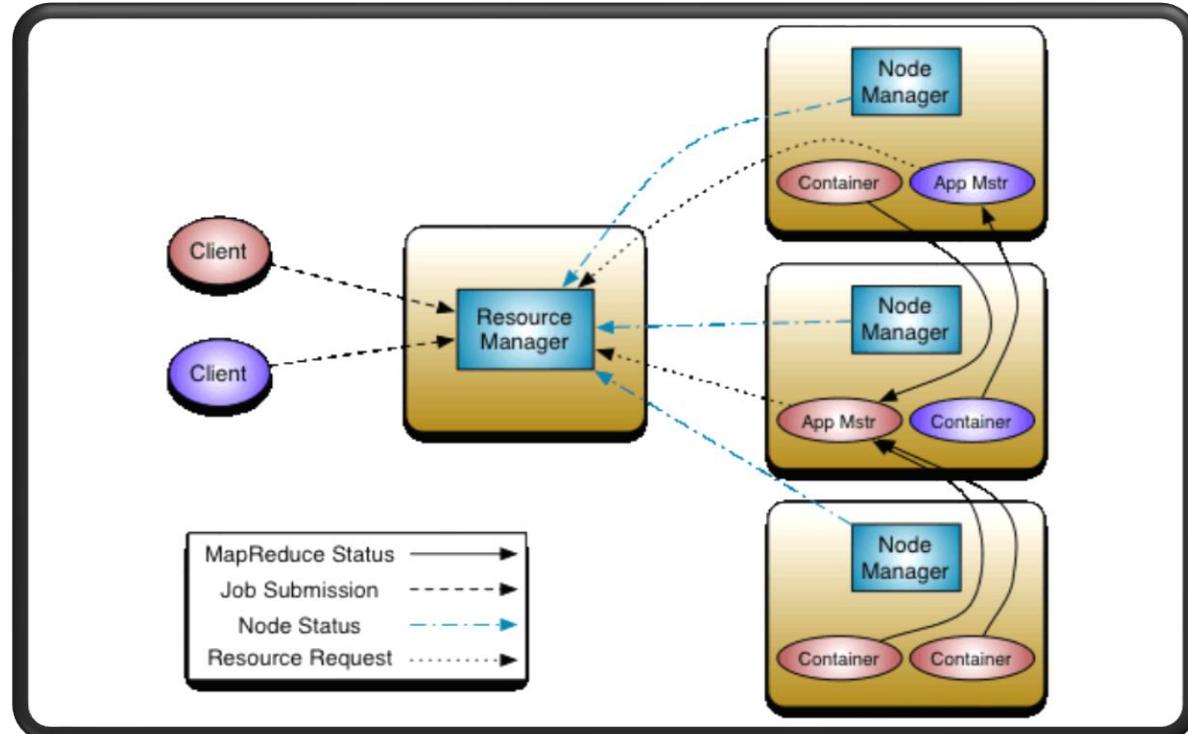


ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- YARN (Yet Another Resource Negotiator):
 - Διαχωρισμός των λειτουργιών της διαχείρισης πόρων
 - Προγραμματισμός εργασιών
 - Παρακολούθηση εργασιών σε ξεχωριστά daemons

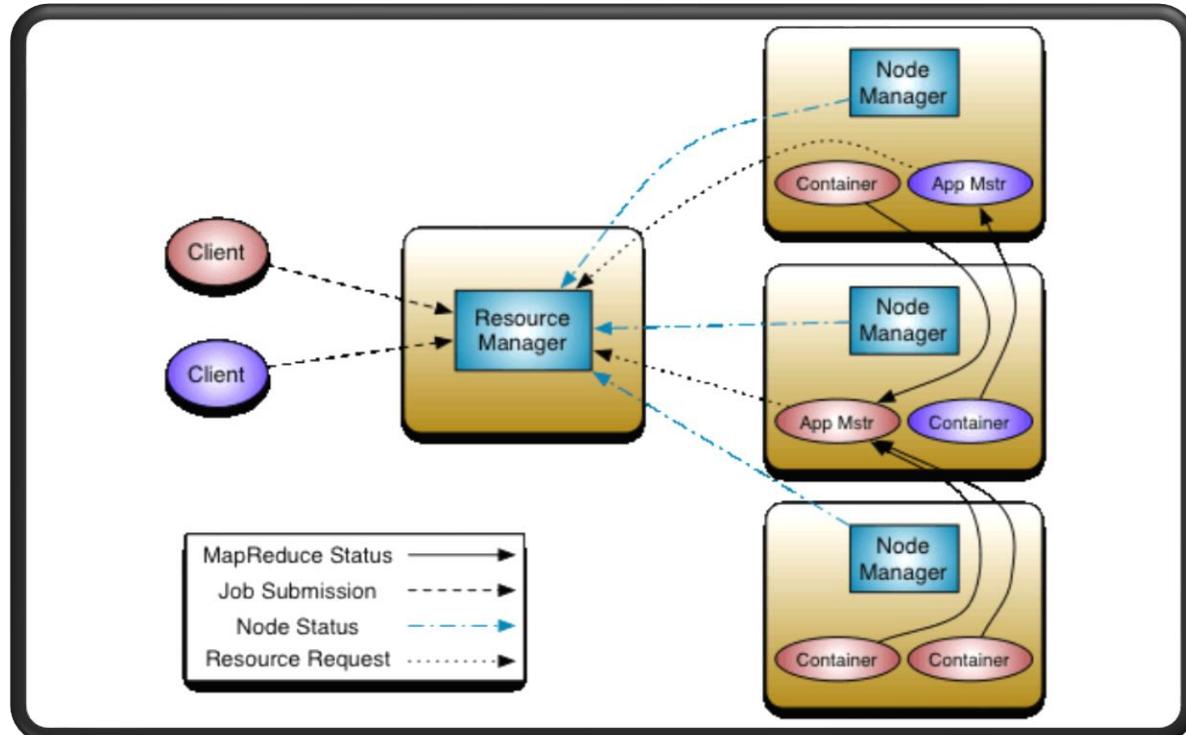
ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- YARN Αρχιτεκτονική:
 - ResourceManager
 - Κατανομή των πόρων
 - Scheduler (βασικό συστατικό ResourceManager)
 - Κατανομή των πόρων σε διάφορες εφαρμογές
 - Δεν προσφέρει καμία εγγύηση για την επανεκκίνηση αποτυχημένων εργασιών
 - Εκτελεί τη λειτουργία προγραμματισμού βάσει των απαιτήσεων σε πόρους εφαρμογών
 - ApplicationsManager (βασικό συστατικό ResourceManager)
 - Αποδοχή της εργασίας που το υποβλήθηκε
 - Διαπραγμάτευση του πρώτου πλαισίου πόρων για την εκτέλεση της συγκεκριμένης εφαρμογής ApplicationMaster
 - Παροχή υπηρεσίας επανεκκίνησης του ApplicationMaster
 - Διαπραγμάτευση των κατάλληλων πόρων από τον Scheduler



ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- YARN Αρχιτεκτονική:
 - NodeManager
 - Παρακολούθηση των πόρων
 - Αναφορά στο ResourceManager
 - ApplicationMaster (βασικό συστατικό NodeManager)
 - Διαπραγμάτευση πόρων από το ResourceManager
 - Συνεργασία με το NodeManager





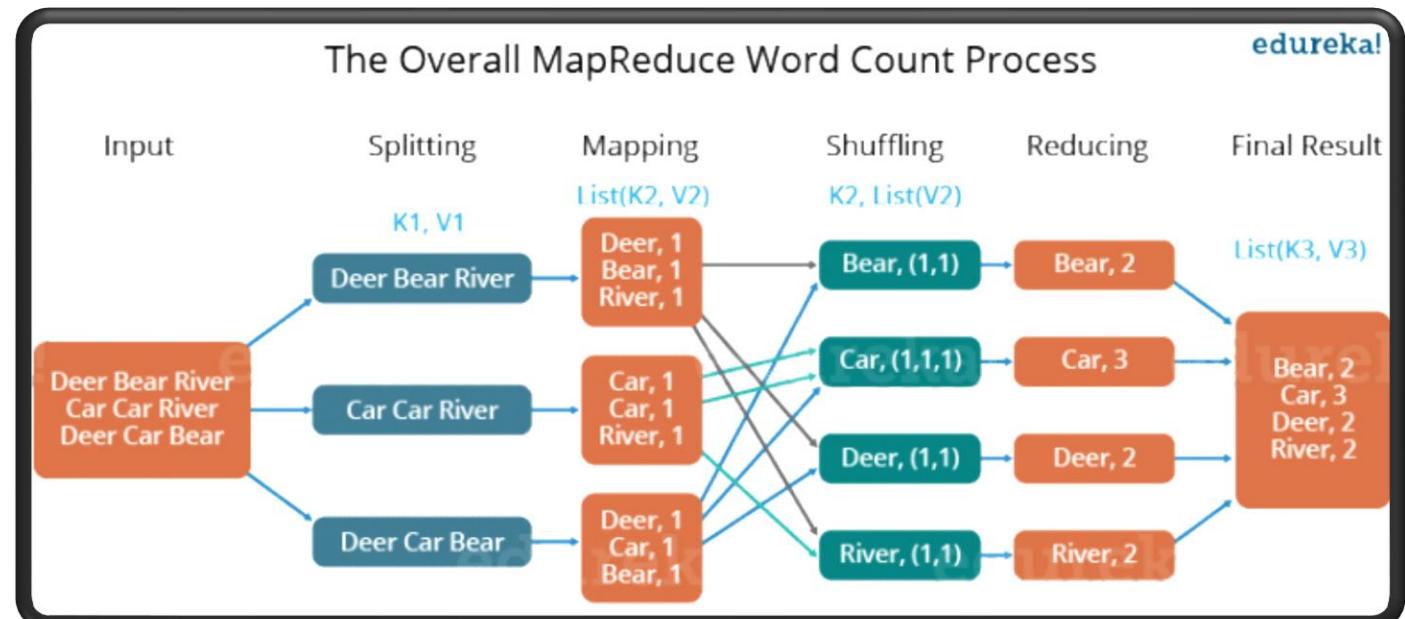
ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- MapReduce:

- Framework που βοηθά να γράφονται εφαρμογές, οι οποίες επεξεργάζονται τεράστιες ποσότητες δεδομένων
- Χωρίζει το σύνολο των δεδομένων εισόδου σε ανεξάρτητα κομμάτια
 - Επεξεργασία αντιστοίχισης (Map)
 - Ταξινόμηση εξόδων, ώστε να επιτευχθεί η εργασία της μείωσης (Reduce)

ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- MapReduce Εργασία:
 - Split
 - Map
 - Sort & Shuffle
 - Reduce



APACHE SPARK



APACHE SPARK

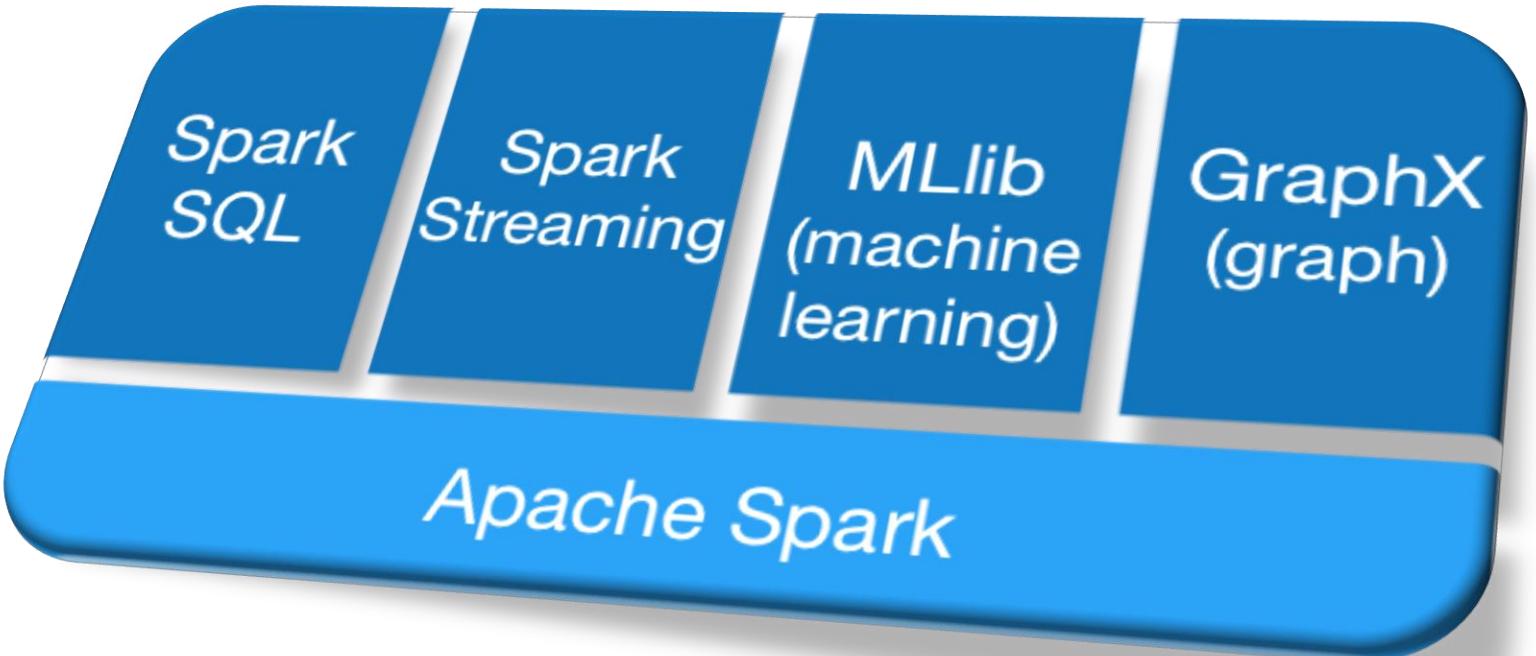
- Μηχανή επεξεργασίας και ανάλυσης δεδομένων ανοιχτού κώδικα
- Αποθήκευση, επεξεργασία, ανάλυση δεδομένων σε πραγματικό χρόνο
- Περιλαμβάνει βιβλιοθήκες για μηχανική εκμάθηση, επεξεργασία ροής δεδομένων και επεξεργασία γράφων

ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΙ ΤΡΟΠΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Γλώσσες προγραμματισμού: Java, Python, Scala, R
- Αρχεία παραμετροποίησης σε μορφή XML
- Εγκατάσταση σε λογισμικό Linux, Windows, macOS
- Μπορεί να τρέξει σε συστήματα, όπως: Hadoop, Apache Mesos, Kubernetes, standalone, cloud
- Spark Core
 - Παροχή κατανεμημένης μετάδοσης εργασιών, προγραμματισμού, I/O λειτουργιών
 - RDD (Resilient Distributed Dataset)
 - Κατανεμημένη αποθήκευση δεδομένων στις μνήμες των μηχανημάτων ενός cluster



TO STACK
TOY SPARK



ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ



ΕΓΚΑΤΑΣΤΑΣΗ APACHE HADOOP

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Προεργασίες:
 - Ενημέρωση συστήματος: `sudo apt update`
 - Εγκατάσταση OPENJDK (Java):
`sudo apt install openjdk-8-jdk -y`
 - Επαλήθευση εγκατάστασης:
`java -version; javac -version`
 - Δημιουργία Non-root χρήστη:
 - Εγκατάσταση OpenSSH server και έναν πελάτη:
`sudo apt install openssh-server openssh-client -y`
 - Δημιουργία Hadoop χρήστη:
`sudo adduser hduser`

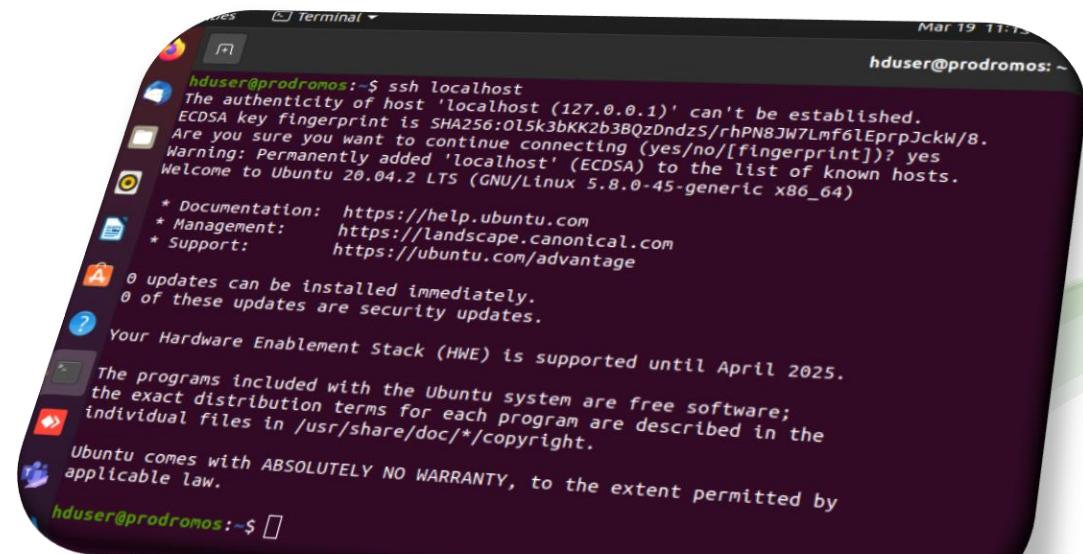


The screenshot shows a terminal window titled 'Terminal' with the command `sudo adduser hduser` being run. The output of the command is displayed, including the creation of a new user group 'hduser', a new user 'hduser' with ID 1001, and a home directory '/home/hduser'. It also shows the copying of files from '/etc/skel'. The terminal then prompts for a new password, which is confirmed. Finally, it asks for user information such as Full Name, Room Number, Work Phone, Home Phone, and Other, with the default value for each being the current value. The user is asked if the information is correct, and the response 'y' is given.

```
prodromos@prodromos:~$ sudo adduser hduser
Adding user 'hduser' ...
Adding new group 'hduser' (1001) ...
Adding new user 'hduser' (1001) with group 'hduser' ...
Creating home directory '/home/hduser' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] y
prodromos@prodromos:~$
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Προεργασίες:
 - Είσοδος Hadoop χρήστη: `su - hduser`
 - Δημιουργία ενός ζεύγους κλειδιών ssh & ορισμός τοποθεσίας αποθήκευσης:
`ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa`
 - Εξουσιοδότηση νέου κλειδιού:
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
 - Ορισμός δικαιωμάτων του χρήστη:
`chmod 0600 ~/.ssh/authorized_keys`
 - Επαλήθευση: `ssh localhost`



The terminal window shows the following text:

```
hduser@prodromos:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:0l5k3bKK2b3BQzDndzS/rhPN8JW7Lmf6LEprpJckW/8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-45-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

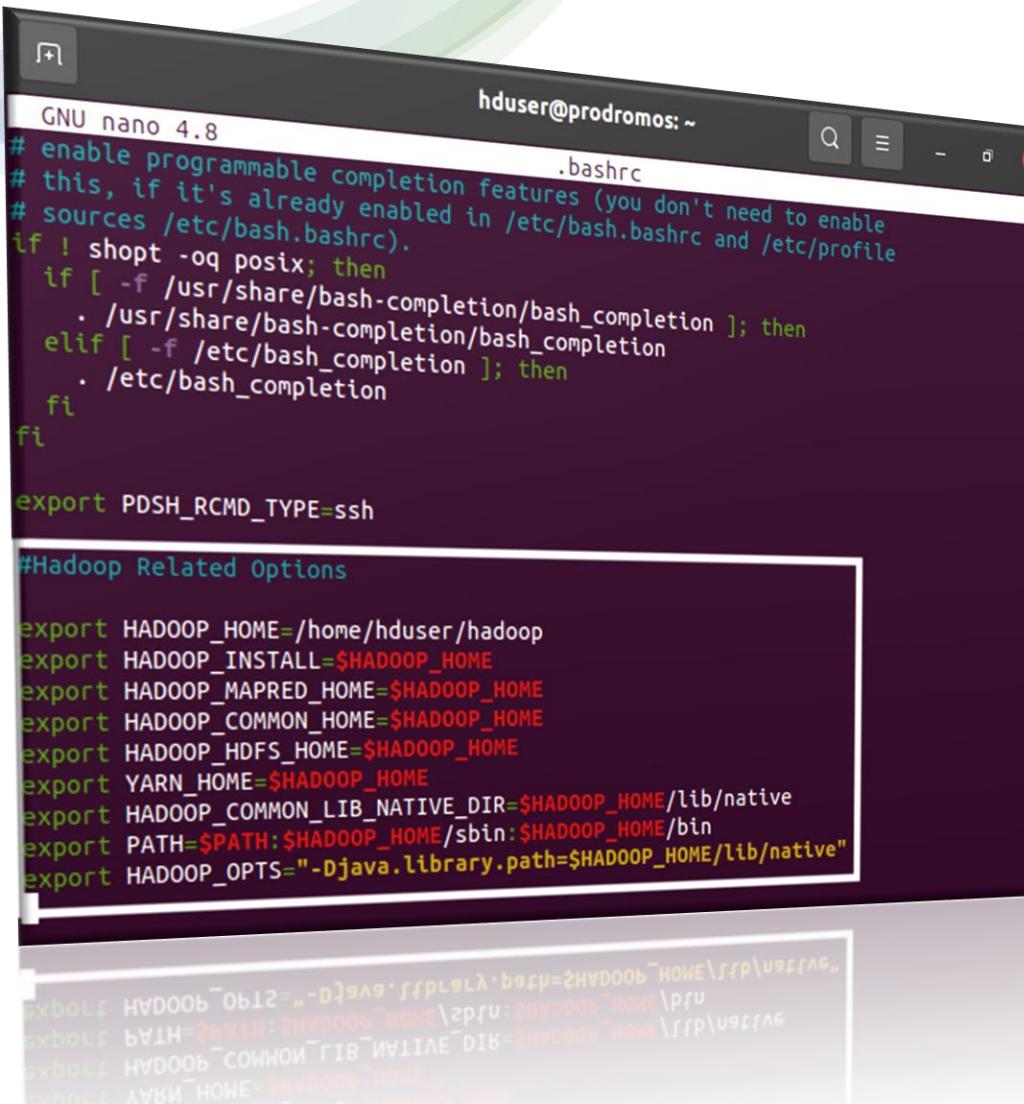
Your Hardware Enablement Stack (HWE) is supported until April 2025.

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

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

hduser@prodromos:~$
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE



A screenshot of a terminal window titled "GNU nano 4.8" showing the contents of a ".bashrc" file. The file contains shell script code for setting up Hadoop. A white rectangular box highlights the section starting with "#Hadoop Related Options".

```
hduser@prodromos: ~
GNU nano 4.8
.bashrc
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PDSH_RCMD_TYPE=ssh

#Hadoop Related Options

export HADOOP_HOME=/home/hduser/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

- Εγκατάσταση Hadoop:
`wget https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz`
- Αποσυμπίεση αρχείου Hadoop:
`tar xzf hadoop-3.2.1.tar.gz`
- Αλλαγή ονόματος από.hadoop-3.2.1 σε.hadoop:
`mv ./hadoop-3.2.1 ./hadoop`
- Μετάβαση στο bashrc και export την εντολή:
`export PDSH_RCMD_TYPE=ssh`
- Μετάβαση στο bashrc και export τις εντολές τη εικόνας:
`sudo nano .bashrc`

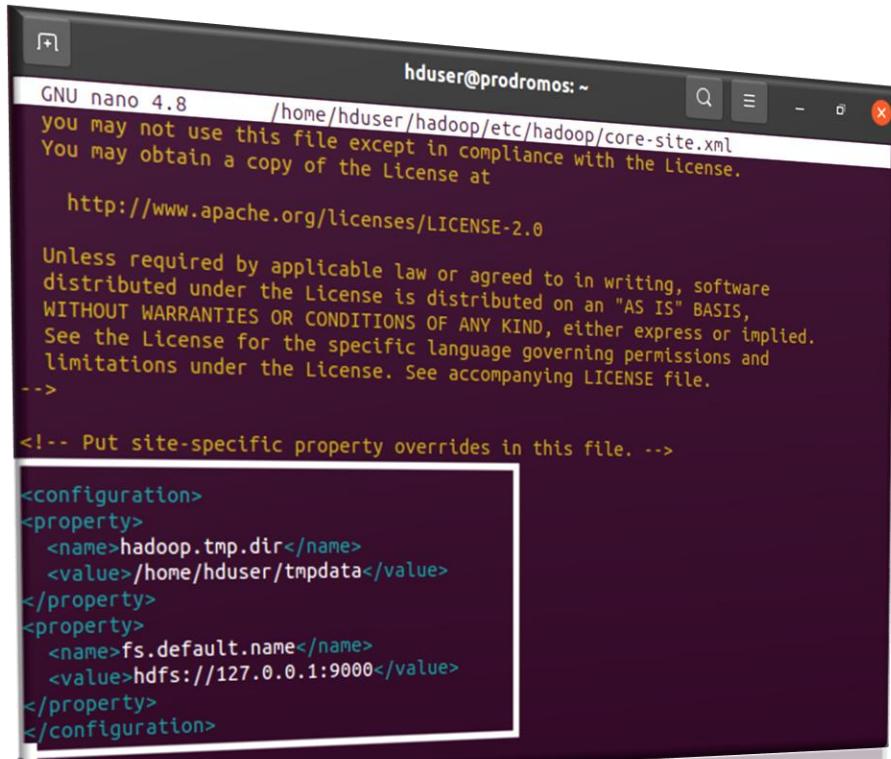
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

```
GNU nano 4.8      hduser@prodromos: ~      /home/hduser/hadoop/etc/hadoop/hadoop-env.sh
# For example:
#
# JAVA_HOME=/usr/java/testing hdfs dfs -ls
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append. If append
# is preferable, modify this file accordingly.
## Generic settings for HADOOP
##
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
# Location of Hadoop. By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=
```

- Επεξεργασία αρχείου hadoop-env.sh:
`sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh`
 - Export το path της Java

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Επεξεργασία αρχείου core-site.xml:
`sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml`
- Δημιουργία καταλόγου tmpdata: `mkdir tmpdata`



```
hduser@prodromos: ~
GNU nano 4.8 /home/hduser/hadoop/etc/hadoop/core-site.xml
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

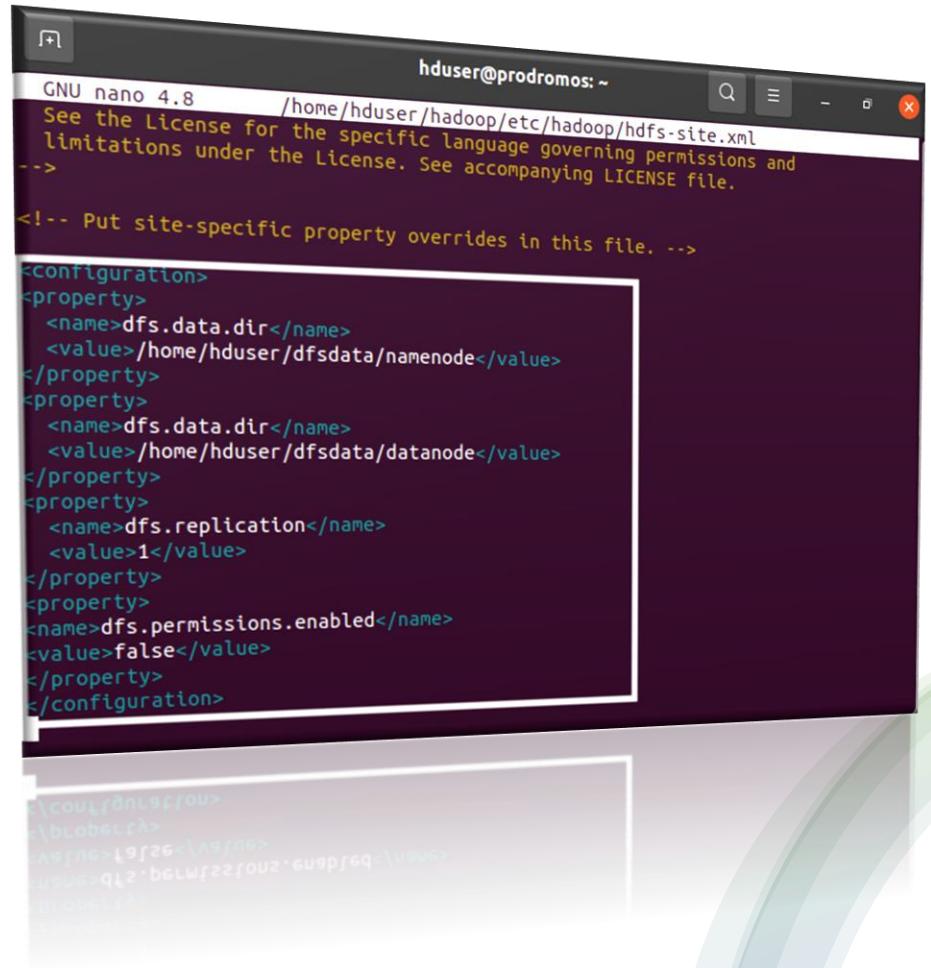
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.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

-->
<! -- Put site-specific property overrides in this file. -->

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hduser/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Επεξεργασία αρχείου hdfs-site.xml:
`sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml`
- Δημιουργία δύο καταλόγων dfsdata (ένα για το namenode και ένα για το datanode): `mkdir dfsdata`



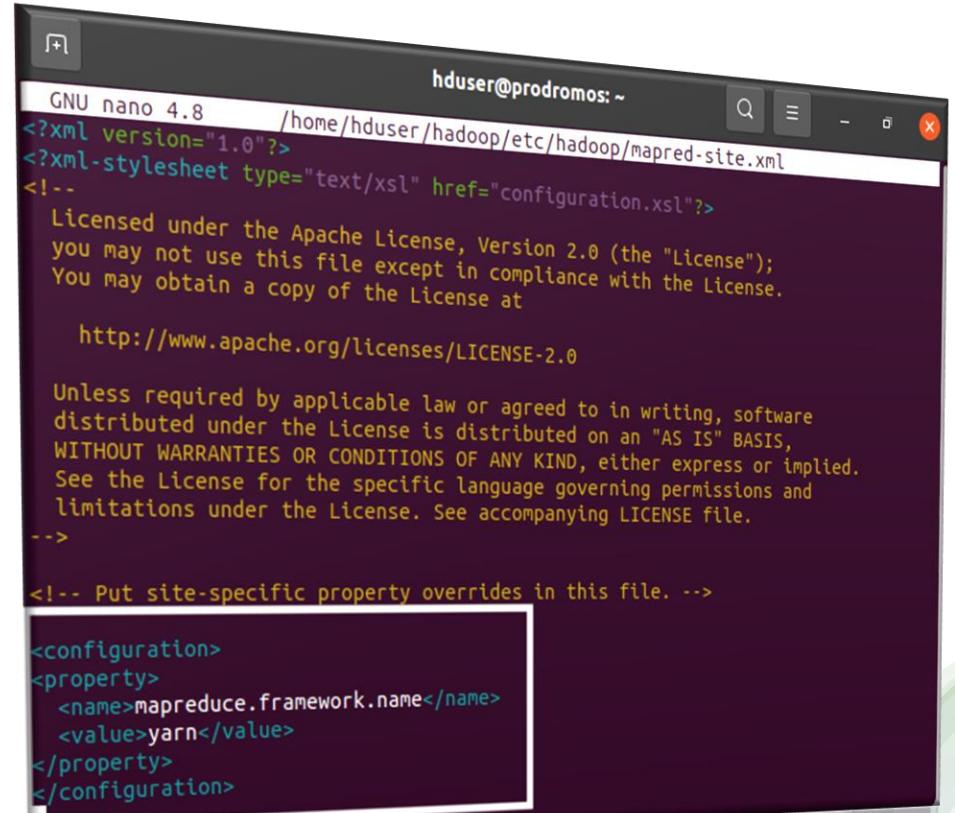
The screenshot shows a terminal window titled "GNU nano 4.8" with the command "hduser@prodromos: ~" at the top. The window displays the XML configuration for HDFS. A specific section of the XML code is highlighted with a white rectangle:

```
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hduser/dfsdata/namenode</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hduser/dfsdata/datanode</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.permissions.enabled</name>
    <value>false</value>
  </property>
</configuration>
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Επεξεργασία αρχείου mapred-site.xml:

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```



```
GNU nano 4.8 /home/hduser/hadoop/etc/hadoop/mapred-site.xml
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

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

<!-- Put site-specific property overrides in this file. -->

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

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Επεξεργασία αρχείου yarn-site.xml:

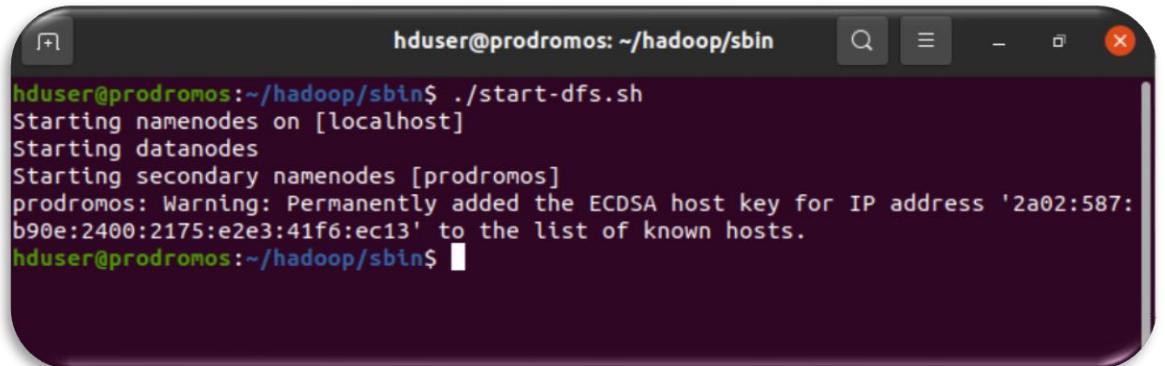
```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

The screenshot shows a terminal window titled "GNU nano 4.8" with the command "hduser@prodromos: ~ /home/hduser/hadoop/etc/hadoop/yarn-site.xml Modified" at the top. The window displays XML code for the "yarn-site.xml" configuration file. The code includes properties for nodemanager aux-services, resource manager hostname, and yarn acl enable. It also contains a property for the environment whitelist, listing Java home, Hadoop common home, HDFS home, Hadoop conf dir, classpath perpend distcache, Hadoop YARN home, and Hadoop Mapred home.

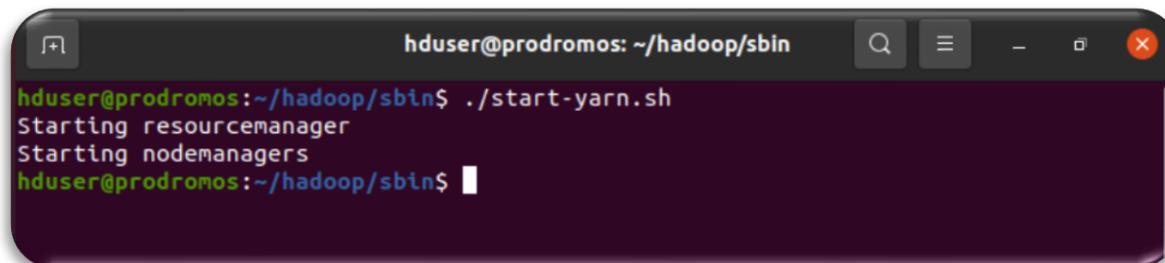
```
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,
CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Μορφοποίηση για NameNode:
`hdfs namenode -format`
- Μετάβαση στον φάκελο hadoop/sbin και εκκίνηση του NameNode και DataNode:
`./start-dfs.sh`
- Εκκίνηση του ResourceManager και NodeManagers: `./start-yarn.sh`
- Επαλήθευση πως όλα τρέχουν σωστά: `jps`



```
hduser@prodromos:~/hadoop/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [prodromos]
prodromos: Warning: Permanently added the ECDSA host key for IP address '2a02:587:b90e:2400:2175:e2e3:41f6:ec13' to the list of known hosts.
hduser@prodromos:~/hadoop/sbin$
```



```
hduser@prodromos:~/hadoop/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hduser@prodromos:~/hadoop/sbin$
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Προεργασίες:
 - Εγκατάσταση ssh: `sudo apt install ssh`
 - Εγκατάσταση pdsh : `sudo apt install pdsh`
 - Μετάβαση στο bashrc και export την εντολή:
`export PDSH_RCMD_TYPE=ssh`
 - Δημιουργία ενός ζεύγους κλειδιών ssh & ορισμός τοποθεσίας αποθήκευσης:
`ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa`
 - Εξουσιοδότηση νέου κλειδιού:
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
 - Επαλήθευση: `ssh localhost`
 - Έλεγχος εγκατάστασης java:
`java -version; javac -version`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Εγκατάσταση Hadoop:
`sudo wget http://apache.cs.utah.edu/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz`
- Αποσυμπίεση αρχείου Hadoop:
`tar xzf hadoop-3.2.1.tar.gz`
- Αλλαγή ονόματος από hadoop-3.2.1 σε hadoop:
`mv ./hadoop-3.2.1 ./hadoop`
- Επεξεργασία αρχείου hadoop-env.sh:
`sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh`
 - Export το path της Java: `export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Μετακίνηση του hadoop στον κατάλογο /usr/local:
`sudo mv hadoop /usr/local/hadoop`
- Μετάβαση στο περιβάλλον του συστήματος:
`sudo nano /etc/environment`
 - `PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64/jre"`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Δημιουργία Hadoop χρήστη: `sudo adduser hadoopuser`
- Πληκτρολογούμε τις εντολές:
 - `sudo usermod -aG hadoopuser hadoopuser`
 - `sudo chown hadoopuser:root -R /usr/local/hadoop/`
 - `sudo chmod g+rwx -R /usr/local/hadoop/`
 - `sudo adduser hadoopuser sudo`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Μετάβαση στο αρχείο των hosts:
`sudo nano .../..etc/hosts`
- Ορισμός master/slaves για αντίστοιχες ip (η ίδια διαδικασία γίνεται και για τους υπολογιστές που θα χρησιμοποιηθούν ως workers)
- Για τις ip: `ip addr`
- Επανεκκίνηση συστήματος:
`sudo reboot`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Ατιγραφή ssh κλειδιού σε όλους τους χρήστες:
 - `ssh-copy-id hadoopuser@hadoop-master`
 - `ssh-copy-id hadoopuser@hadoop-slave1`
 - `ssh-copy-id hadoopuser@hadoop-slave2`
- Μετάβαση στο bashrc και export τις εντολές: `sudo nano .bashrc`
 - `export HADOOP_HOME=/home/hadoopuser/hadoop`
 - `export PATH=$PATH:$HADOOP_HOME/bin`
 - `export PATH=$PATH:$HADOOP_HOME/sbin`
 - `export HADOOP_MAPRED_HOME=${HADOOP_HOME}`
 - `export HADOOP_COMMON_HOME=${HADOOP_HOME}`
 - `export HADOOP_HDFS_HOME=${HADOOP_HOME}`
 - `export YARN_HOME=${HADOOP_HOME}`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Επεξεργασία αρχείου hadoop-env.sh:
`sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh`
 - Export το Java Home: `export JAVA_HOME=$JAVA_HOME`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Επεξεργασία αρχείου core-site.xml:

```
sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

```
• <configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop-master:9000</value>
  </property>
</configuration>
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Επεξεργασία αρχείου hdfs-site.xml:

```
sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

- ```
<configuration>
<property>
 <name>dfs.namenode.name.dir</name><value>/usr/local/hadoop/data/nameNode</value>
</property>
<property>
 <name>dfs.datanode.data.dir</name><value>/usr/local/hadoop/data/dataNode</value>
</property>
<property>
 <name>dfs.replication</name>
 <value>2</value>
</property>
</configuration>
```

- Δημιουργία δύο καταλόγων dfsdata (ένα για το namenode και ένα για το datanode): `mkdir dfsdata`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Επεξεργασία αρχείου yarn-site.xml:

```
sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

- ```
<configuration>
  <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.resourcemanager.hostname</name>
    <value>localhost</value>
  </property>
</configuration>
```

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Επεξεργασία αρχείου mapred-site.xml:

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- ```
<configuration>
 <property>
 <name>mapreduce.jobtracker.address</name>
 <value>pd-master:54311</value>
 </property>
 <property>
 <name>mapreduce.framework.name</name>
 <value>yarn</value>
 </property>
</configuration>
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Μετάβαση στο αρχείο των workers:  
`sudo nano ~/hadoop/etc/hadoop/workers`
  - `hadoop1@pd-slave01`
  - `hadoop2@pd-slave02`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Μορφοποίηση για NameNode:  
`hdfs namenode -format`
- Μετάβαση στον φάκελο hadoop/sbin και εκκίνηση του NameNode και DataNode:  
`./start-dfs.sh`
- Επαλήθευση πως όλα τρέχουν σωστά: `jps`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

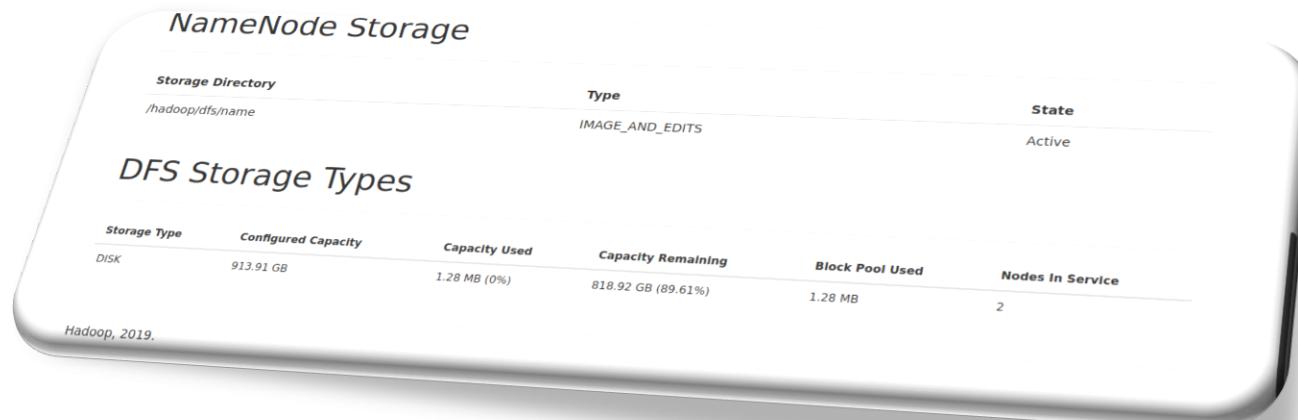
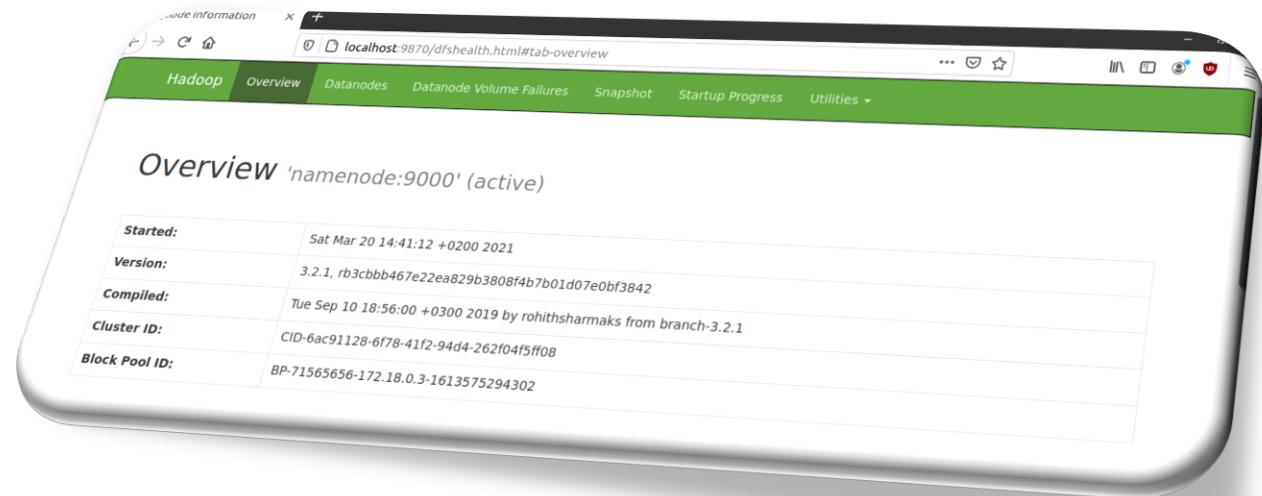
- Ρύθμιση YARN
  - `export HADOOP_HOME="/home/hadoop"`
  - `export HADOOP_COMMON_HOME=$HADOOP_HOME`
  - `export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop`
  - `export HADOOP_HDFS_HOME=$HADOOP_HOME`
  - `export HADOOP_MAPRED_HOME=$HADOOP_HOME`
  - `export HADOOP_YARN_HOME=$HADOOP_HOME`
- Επεξεργασία αρχείου `yarn-site.xml` των slaves:  
`sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml`
  - `<property>`  
`<name>yarn.resourcemanager.hostname</name>`  
`<value>hadoop-master</value>`  
`</property>`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Μετάβαση στον φάκελο hadoop/sbin και εκκίνηση του ResourceManager και NodeManagers:  
`./start-yarn.sh`
- Επαλήθευση πως όλα τρέχουν σωστά: `jps`

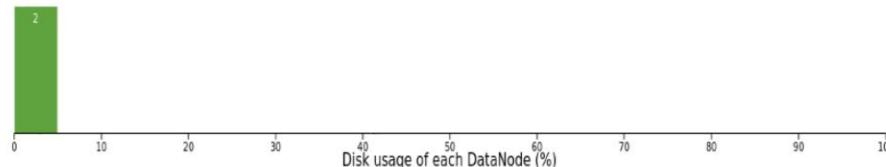
# ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

- Πρόσβαση στο Hadoop NameNode UI: <http://localhost:9870>



# ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

Datanode usage histogram



In operation

A table titled "In operation" listing two DataNodes. The columns include Node, Http Address, Last contact, Last Block Report, Capacity, Blocks, Block pool used, and Version.

Node	Http Address	Last contact	Last Block Report	Capacity	Blocks	Block pool used	Version
✓7ab139f3d91e:9866 (172.18.0.5:9866)	http://7ab139f3d91e:9864	1s	128m	456.96 GB	26	656 KB (0%)	3.2.1
✓prodromos-9866 (172.18.0.1:9866)	http://prodromos:9864	1s	5m	456.96 GB	26	656 KB (0%)	3.2.1

Show 25 entries Search: /

Showing 1 to 2 of 2 entries

## Browse Directory

A table titled "Browse Directory" showing the contents of the root directory. The columns include Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name.

	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
✓	drwxrwxrwt	root	root	0 B	Feb 17 17:30	0	0 B	app-logs
✓	drwxr-xr-x	root	supergroup	0 B	Feb 17 17:21	0	0 B	rmstate
✓	drwx-----	root	supergroup	0 B	Feb 17 17:29	0	0 B	tmp
✓	drwxr-xr-x	root	supergroup	0 B	Feb 25 12:59	0	0 B	user

Show 25 entries Search: /

Showing 1 to 4 of 4 entries

Hadoop, 2019.

# ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

- Πρόσβαση στο Hadoop DataNode  
UI: <http://localhost:9864>

The screenshot shows a web browser window with the URL `localhost:9864/datanode.html`. The page title is "DataNode on prodromos:9866".  
**Cluster Information:**

Cluster ID:	CID-6ac91128-6f78-41f2-94d4-262f04f5ff08
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842

  
**Block Pools:**

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	BP-71565656-172.18.0.3-1613575294302	RUNNING	2s	8 minutes	243 B (64 MB)

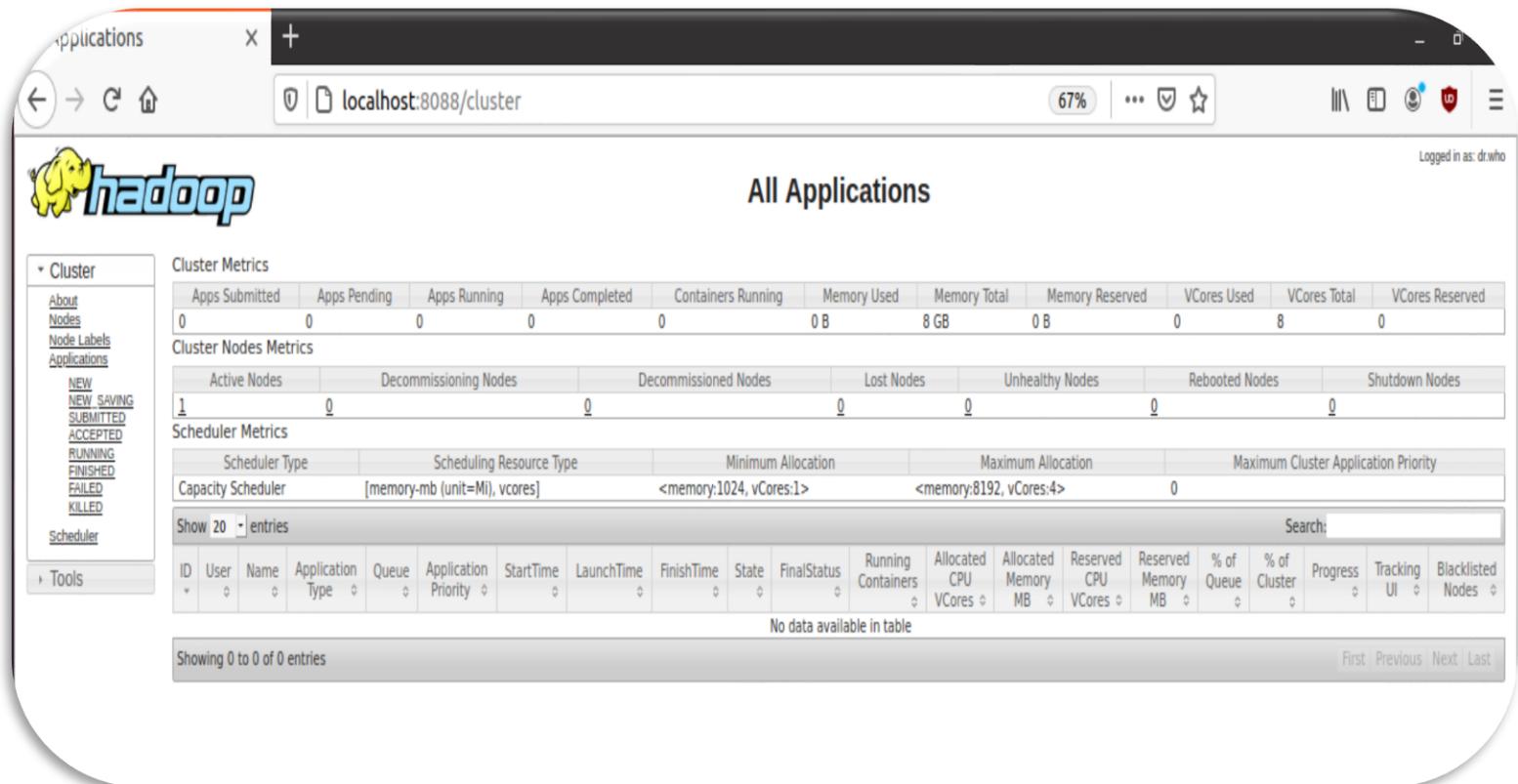
  
**Volume Information:**

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hduser/dfsdata/datanode	DISK	656 KB	409.46 GB	0 B	0 B	26

Hadoop, 2019.

# ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

- Πρόσβαση στο Hadoop YARN UI:  
<http://localhost:8080>



The screenshot shows the Hadoop YARN User Interface (UI) running at <http://localhost:8080>. The title bar says "applications". The main content area is titled "All Applications". It features several metrics tables:

- Cluster Metrics:** Shows 0 Apps Submitted, 0 Apps Pending, 0 Apps Running, 0 Apps Completed, 0 Containers Running, 0 B Memory Used, 8 GB Memory Total, 0 B Memory Reserved, 0 VCores Used, 8 VCores Total, and 0 VCores Reserved.
- Cluster Nodes Metrics:** Shows 1 Active Nodes, 0 Decommissioning Nodes, 0 Decommissioned Nodes, 0 Lost Nodes, 0 Unhealthy Nodes, 0 Rebooted Nodes, and 0 Shutdown Nodes.
- Scheduler Metrics:** Shows the Scheduler Type as Capacity Scheduler, Scheduling Resource Type as [memory-mb (unit=Mi), vcores], Minimum Allocation as <memory:1024, vCores:1>, Maximum Allocation as <memory:8192, vCores:4>, and Maximum Cluster Application Priority as 0.

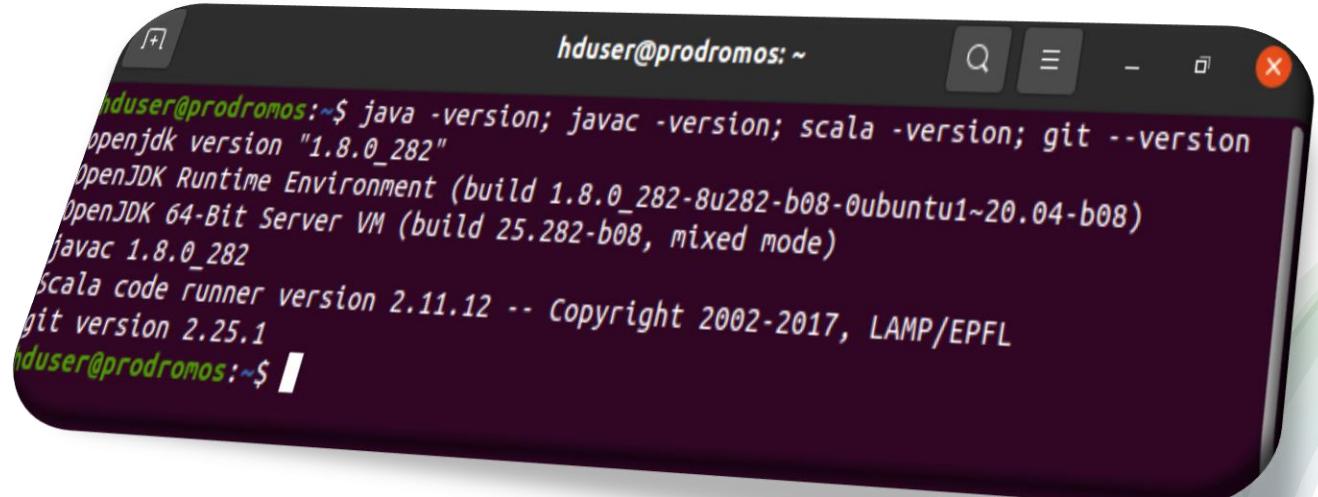
On the left, there is a sidebar with navigation links: Cluster (About, Nodes, Node Labels, Applications, Scheduler), Tools, and a status bar showing "Logged in as: dr.who". At the bottom, there is a table header for "Show 20 entries" and a search bar. The message "No data available in table" is displayed below the table headers.



# ΕΓΚΑΤΑΣΤΑΣΗ APACHE SPARK

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Προεργασίες:
  - Εγκατάσταση Java, Scala, Git:  
`sudo apt install default-jdk scala git -y`
  - Επαλήθευση εγκαταστάσεων:  
`java -version; javac -version; scala -version; git -version`

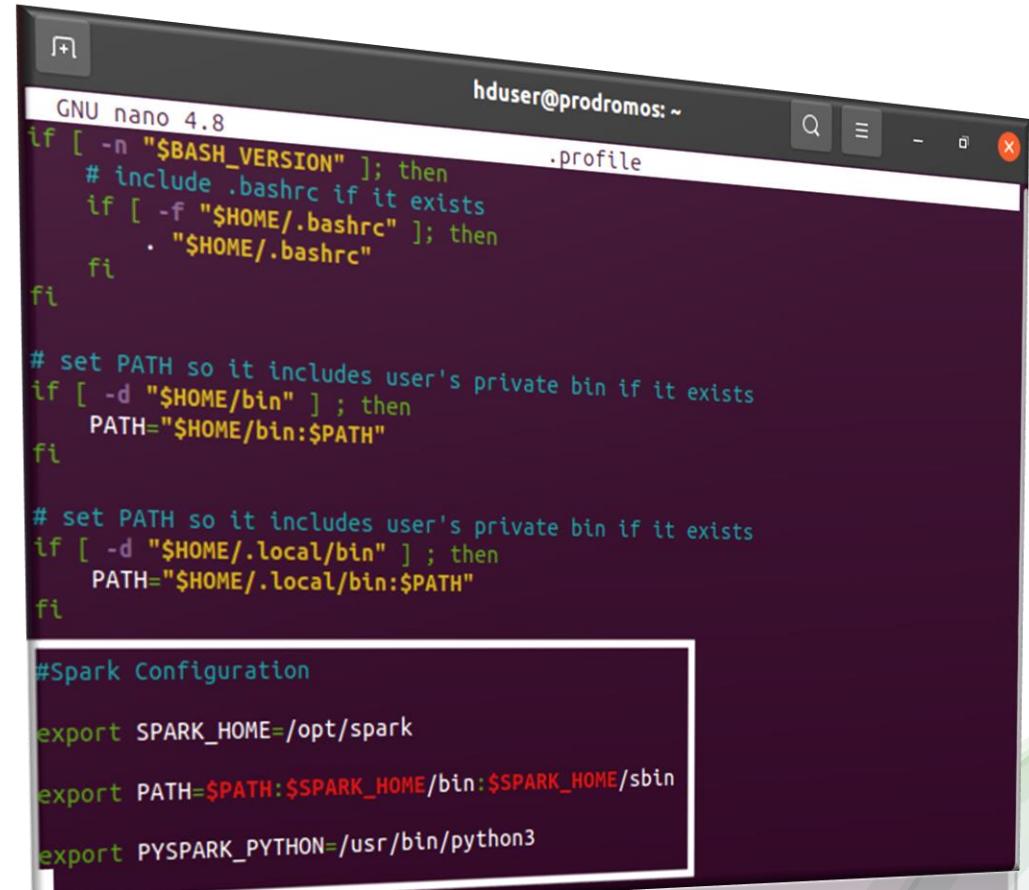


A screenshot of a terminal window titled "hduser@prodromos: ~". The window shows the following command and its output:

```
hduser@prodromos:~$ java -version; javac -version; scala -version; git --version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
javac 1.8.0_282
scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
git version 2.25.1
hduser@prodromos:~$
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Εγκατάσταση Spark:  
`wget https://downloads.apache.org/spark/spark-3.1.1/spark-3.1.1-bin-hadoop3.2.tgz`
- Αποσυμπίεση αρχείου Spark: `tar xvf spark-*`
- Αλλαγή ονόματος από `spark-3.0.1-bin-hadoop2.7` σε `spark`: `mv ./spark-3.0.1-bin-hadoop2.7 ./spark`
- Μετάβαση αρχείου στον κατάλογο `opt/spark`:  
`sudo mv spark /opt/spark`
- Μετάβαση στο `profile` και `export` τις εντολές της εικόνας: `sudo nano .profile`



```
GNU nano 4.8
hduser@prodromos: ~ .profile
if [-n "$BASH_VERSION"]; then
include .bashrc if it exists
if [-f "$HOME/.bashrc"]; then
. "$HOME/.bashrc"
fi
fi

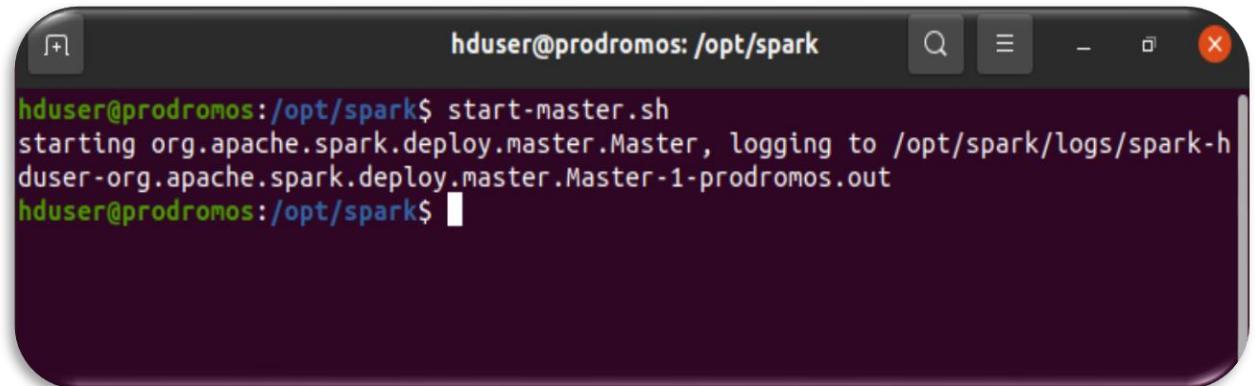
set PATH so it includes user's private bin if it exists
if [-d "$HOME/bin"] ; then
PATH="$HOME/bin:$PATH"
fi

set PATH so it includes user's private bin if it exists
if [-d "$HOME/.local/bin"] ; then
PATH="$HOME/.local/bin:$PATH"
fi

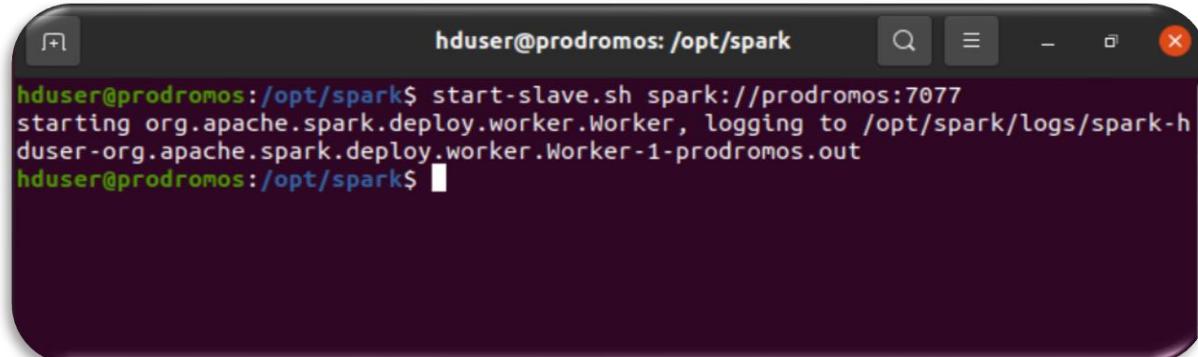
#Spark Configuration
export SPARK_HOME=/opt/spark
export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin
export PYSPARK_PYTHON=/usr/bin/python3
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Μετάβαση στον κατάλογο του spark:  
`cd ../../opt/spark`
- Εκκίνηση master server: `start-master.sh`
- Εκκίνηση slave server:  
`start-slave.sh spark://prodromos:7077`



```
hduser@prodromos:/opt/spark$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spark-hduser-org.apache.spark.deploy.master.Master-1-prodromos.out
hduser@prodromos:/opt/spark$
```



```
hduser@prodromos:/opt/spark$ start-slave.sh spark://prodromos:7077
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spark-hduser-org.apache.spark.deploy.worker.Worker-1-prodromos.out
hduser@prodromos:/opt/spark$
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Εκκίνηση spark-shell (αν θέλουμε να ορίσουμε πόσοι πυρήνες θα χρησιμοποιηθούν συμπληρώνουμε την εντολή -master local[x]):  
**spark-shell**

```
hduser@prodromos:/opt/spark$ spark-shell
21/03/20 16:24:31 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
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://192.168.1.23:4040
Spark context available as 'sc' (master = local[*], app id = local-1616250285113).
Spark session available as 'spark'.
Welcome to

 / \ \ / \
 / \ \ / \
 / \ \ / \
 / \ \ / \
\ \ \ / \
 \ \ \ / \
 \ \ \ / \
 \ \ \ / \
 \ \ \ / \
 \ \ / \
 \ / \
 \ /
version 3.0.2

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_282)
Type in expressions to have them evaluated.
Type :help for more information.

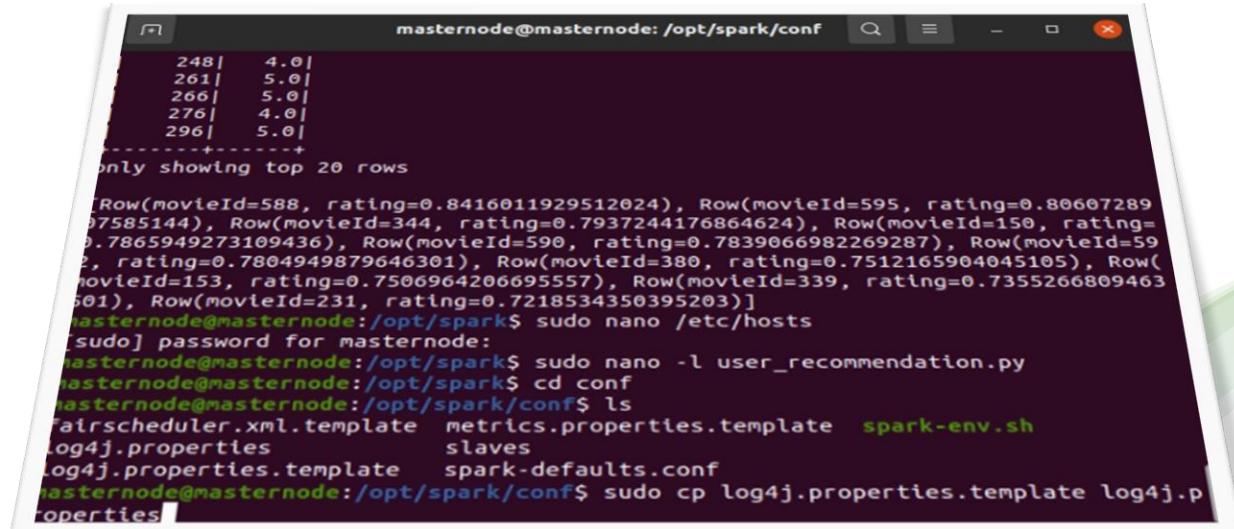
scala> █
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Εκκίνηση pyspark: `pyspark`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Αντιγραφή των περιεχομένων των template αρχείων σε καινούργια shell αρχεία (.sh) με το ίδιο όνομα:
  - `sudo nano spark-env.template spark-env.sh`
- Εκτελούμε επίσης τις παρακάτω εντολές:
  - `sudo cp spark-defaults.conf.template spark-defaults.conf`
  - `sudo cp slaves.template slaves`



```
248| 4.0|
261| 5.0|
266| 5.0|
276| 4.0|
296| 5.0|
-----+-----+
only showing top 20 rows

[Row(movieId=588, rating=0.8416011929512024), Row(movieId=595, rating=0.80607289
07585144), Row(movieId=344, rating=0.7937244176864624), Row(movieId=150, rating=
0.7865949273109436), Row(movieId=590, rating=0.7839066982269287), Row(movieId=59
2, rating=0.7804949879646301), Row(movieId=380, rating=0.7512165904045105), Row(
movieId=153, rating=0.7506964206695557), Row(movieId=339, rating=0.7355266809463
501), Row(movieId=231, rating=0.7218534350395203)]
[masternode@masternode:/opt/spark$ sudo nano /etc/hosts
[sudo] password for masternode:
[masternode@masternode:/opt/spark$ sudo nano -l user_recommendation.py
[masternode@masternode:/opt/spark$ cd conf
[masternode@masternode:/opt/spark/conf$ ls
fairscheduler.xml.template metrics.properties.template spark-env.sh
log4j.properties slaves
log4j.properties.template spark-defaults.conf
[masternode@masternode:/opt/spark/conf$ sudo cp log4j.properties.template log4j.p
roperties]
```

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

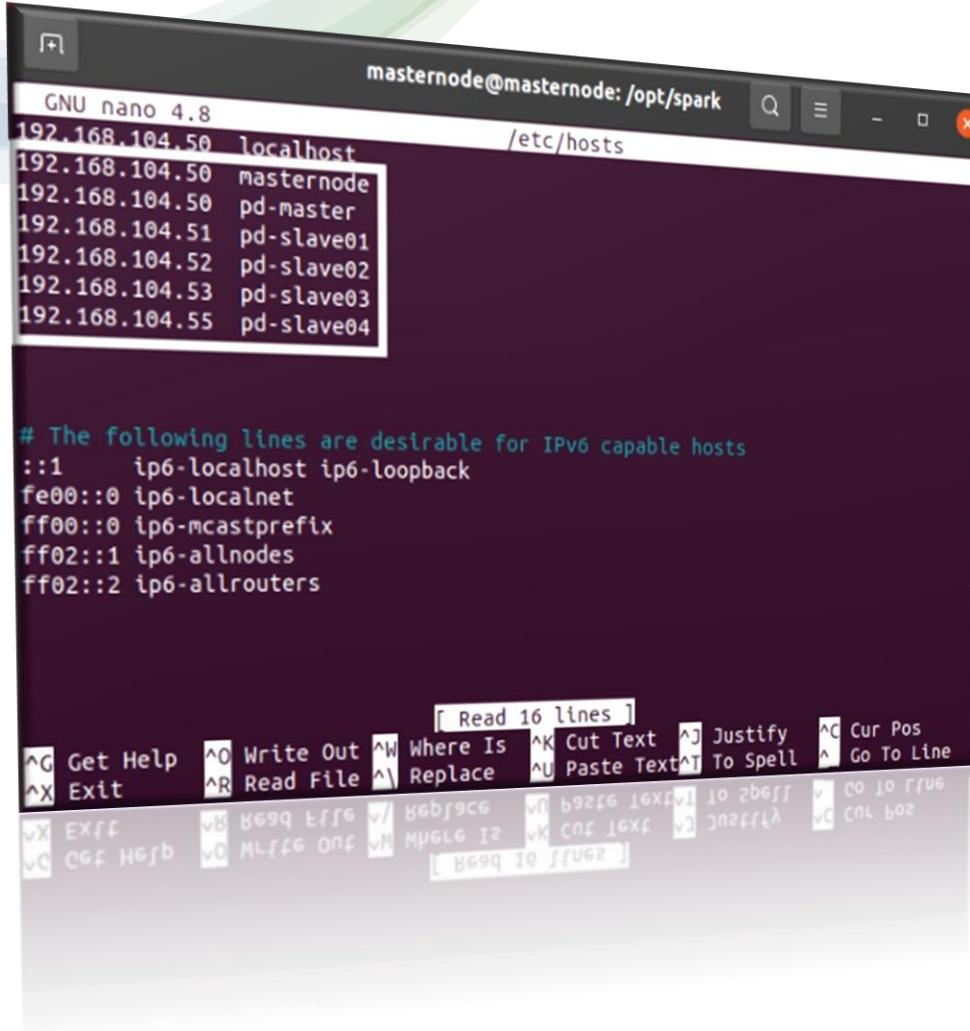
```
GNU nano 4.8 masternode@centralnode: ~/..../opt/spark/conf/spark-env.sh
Options for the daemons used in the standalone deploy mode
export SPARK_MASTER_HOST='192.168.104.50' ←
- SPARK_MASTER_HOST, to bind the master to a different IP address or hostname
- SPARK_MASTER_PORT / SPARK_MASTER_WEBUI_PORT, to use non-default ports for t>
- SPARK_MASTER_OPTS, to set config properties only for the master (e.g. "-Dx=>
- SPARK_WORKER_CORES, to set the number of cores to use on this machine
- SPARK_WORKER_MEMORY, to set how much total memory workers have to give exec>
- SPARK_WORKER_PORT / SPARK_WORKER_WEBUI_PORT, to use non-default ports for t>
- SPARK_WORKER_DIR, to set the working directory of worker processes
- SPARK_WORKER_OPTS, to set config properties only for the worker (e.g. "-Dx=>
- SPARK_DAEMON_MEMORY, to allocate to the master, worker and history server t>
- SPARK_HISTORY_OPTS, to set config properties only for the history server (e>
- SPARK_SHUFFLE_OPTS, to set config properties only for the external shuffle >
- SPARK_DAEMON_JAVA_OPTS, to set config properties for all daemons (e.g. "-Dx=>
- SPARK_DAEMON_CLASSPATH, to set the classpath for all daemons
- SPARK_PUBLIC_DNS, to set the public dns name of the master or workers

Options for launcher
- SPARK_LAUNCHER_OPTS, to set config properties and Java options for the laun>
```

- Δημιουργία του Master:

- Μετάβαση στο spark-env.sh:  
`sudo nano .../..../opt/spark/conf/spark-env.sh`
- Ορισμός master host μαζί με την ip του:  
`export SPARK_MASTER_HOST='MASTER_HOST_IP'`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE



```
masternode@masternode: /opt/spark
GNU nano 4.8
192.168.104.50 localhost
192.168.104.50 masternode
192.168.104.50 pd-master
192.168.104.51 pd-slave01
192.168.104.52 pd-slave02
192.168.104.53 pd-slave03
192.168.104.55 pd-slave04

The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[Read 16 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^L Go To Line
^V EXFF ^B B699 E756 ^V B699CE5 ^V B92F6 TFXF ^V 79950 TFXF ^V CO 10 F7U6
^C C6F H9jB ^V B699 E756 ^V B699CE5 ^V B92F6 TFXF ^V 79950 TFXF ^V CO 10 F7U6
^D C6F H9jB ^V B699 E756 ^V B699CE5 ^V B92F6 TFXF ^V 79950 TFXF ^V CO 10 F7U6
```

- Δημιουργία του Master:
  - Μετάβαση στο αρχείο των hosts:  
`sudo nano ../../etc/hosts`
  - Ορισμός master/slaves για αντίστοιχες ip (η ίδια διαδικασία γίνεται και για τους υπολογιστές που θα χρησιμοποιηθούν ως workers)
  - Για τις ip: `ip addr`
  - Επανεκκίνηση συστήματος:  
`sudo reboot`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Έλεγχος εγκατάστασης Java & Scala:  
`java -version; javac -version; scala -version`
- Έλεγχος ssh: `ssh`
- Εγκατάσταση ssh:  
`sudo apt-get install openssh-server openssh-client`
- Δημιουργία ζεύγους κλειδιών:  
`ssh-keygen -t rsa -P ''`
- Εξουσιοδότηση νέου κλειδιού:  
`cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
- Αντιγραφή περιεχομένου `.ssh/id_rsa.pub` στο `.ssh/authorized_keys` όλων των workers:  
`ssh-copy-id user@pd-master`  
`ssh-copy-id user@pd-slave1`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- Είσοδος στον 1<sup>o</sup> slave: `ssh user@pd-slave1`
- Επιστροφή στον master: `exit`
- Είσοδος στο bashrc: `sudo nano ~/.bashrc`
- Export το path του spark:  
`export PATH=$PATH:/opt/spark/bin`

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

The screenshot shows a terminal window titled "masternode@masternode:/opt/spark" running the "nano" editor on file "conf/slaves". The file contains the Apache Software Foundation license and the configuration for starting workers on four slave nodes. The configuration section lists four hosts: node\_1@pd-slave01, node\_2@pd-slave02, node\_3@pd-slave03, and node\_4@pd-slave04.

```
GNU nano 4.8
masternode@masternode:/opt/spark
conf/slaves

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

A Spark Worker will be started on each of the machines listed below.
masternode@192.168.104.50
node_1@pd-slave01
node_2@pd-slave02
node_3@pd-slave03
node_4@pd-slave04
```

- Ορισμός των workers

# ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

```
GNU nano 4.8
masternode@masternode:/opt/spark
```

	/etc/hosts
192.168.104.50	localhost
192.168.104.50	masternode
192.168.104.50	pd-master
192.168.104.51	pd-slave01
192.168.104.52	pd-slave02
192.168.104.53	pd-slave03
192.168.104.55	pd-slave04

```
The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- Ρύθμιση ενός worker:
  - Αντιγραφή των template αρχείων παρόμοια με την αντιγραφή που έγινε στον κεντρικό κόμβο
  - Μετάβαση στο αρχείο spark-env.sh:  
`sudo nano ../../opt/spark/conf/spark-env.sh`
  - Export:  
`export SPARK_MASTER_HOST = 'YOUR_MASTER_HOST_IP'`
  - Μετάβαση στο αρχείο των hosts:  
`sudo nano ../../etc/hosts`
  - Ορισμός ip με τα ονόματα των υπολογιστών

# ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

- Εκκίνηση του δικτύου από τον κεντρικό κόμβο
    - `./start-all.sh`
  - Πρόσβαση στο Spark Web:
    - [Cluster Demo](#)

Apache Spark 3.0.2									
Spark Master at spark://192.168.104.50:7077									
URL: spark://192.168.104.50:7077									
<b>Alive Workers:</b> 4									
<b>Cores in use:</b> 12 Total, 0 Used									
<b>Memory in use:</b> 22.7 GiB Total, 0.0 B Used									
<b>Resources in use:</b>									
<b>Applications:</b> 0 <a href="#">Running</a> , 2 <a href="#">Completed</a>									
<b>Drivers:</b> 0 Running, 0 Completed									
<b>Status:</b> ALIVE									
<a href="#">Workers (4)</a>									
Worker Id	Address			State	Cores	Memory	Resources		
worker-20210509183425-192.168.104.51-36793	192.168.104.51:36793			ALIVE	3 (0 Used)	6.0 GiB (0.0 B Used)			
worker-20210509183425-192.168.104.52-43281	192.168.104.52:43281			ALIVE	4 (0 Used)	6.7 GiB (0.0 B Used)			
worker-20210509183425-192.168.104.53-37735	192.168.104.53:37735			ALIVE	3 (0 Used)	6.0 GiB (0.0 B Used)			
worker-20210509183441-192.168.104.50-35905	192.168.104.50:35905			ALIVE	2 (0 Used)	4.0 GiB (0.0 B Used)			
<a href="#">Running Applications (0)</a>									
Application ID	Name	Cores	Memory per Executor		Resources Per Executor		Submitted Time		User
									State
									Duration
<a href="#">Completed Applications (2)</a>									
Application ID	Name		Cores	Memory per Executor		Resources Per Executor		Submitted Time	User
app-20210509194009-0001	PySparkShell		12	1024.0 MiB				2021/05/09 19:40:09	masternode
app-20210509183503-0000	Movie Recommendation		12	3.0 GiB				2021/05/09 18:35:03	masternode
									FINISHED
									3.7 min
									11 min

# ΣΥΓΚΡΙΣΗ HADOOP ME SPARK



Performance

## Hadoop

- Γενικά αργό (εκτελεί τις εργασίες στο δίσκο)
- Αποθηκεύει τα δεδομένα σε πολλές διαφορετικές τοποθεσίες και στη συνέχεια τα επεξεργάζεται σε παρτίδες χρησιμοποιώντας το MapReduce

## Spark

- Αποθηκεύει τα δεδομένα σε RAM



Cost

## Hadoop

- Χαμηλό κόστος εκτέλεσης  
(αποθήκευση δεδομένων στο δίσκο)
- Υπολογιστικά συστήματα χαμηλού  
κόστους

## Spark

- Μεγάλο κόστος εκτέλεσης  
(αποθήκευση δεδομένων σε RAM)



## Fault Tolerance

### Hadoop

- Ανοχή σε σφάλματα υλικού
- Αναπαράγει τα δεδομένα στους κόμβους
- Ο κύριος κόμβος παρακολουθεί την κατάσταση των δευτερευόντων κόμβων

### Spark

- Ανοχή σε σφάλματα υλικού
- Χρησιμοποιεί μπλοκ RDD
- Παρακολουθεί τον τρόπο δημιουργίας του αμετάβλητου συνόλου δεδομένων
- Μπορεί να αναδημιουργήσει δεδομένα σε ένα cluster



## Data Processing

### Hadoop

- Χρησιμοποιεί MapReduce
- Βασίζεται σε απλό υλικό για την αποθήκευση
- Κατάλληλο για γραμμική επεξεργασία δεδομένων

### Spark

- Χρησιμοποιεί RDD
- Εκτελεί τις λειτουργίες παράλληλα
- Ανάλυση δεδομένων σε πραγματικό χρόνο



## Ease of Use & Language Support

### Hadoop

- Περίπλοκο (δεν έχει διαδραστική λειτουργία)
- Βασίζεται σε γλώσσα προγραμματισμού Java
- Σύνταξη κώδικα για MR εργασίες με Java ή Python

### Spark

- Φιλικό προς τον χρήστη
- Υποστήριξη API σε αρκετές γλώσσες προγραμματισμού
- Μητρική γλώσσα Scala, υποστηρίζει και Java, Python, R, Spark SQL
- Spark-shell, pyspark, spark-submit



Scalability

## Hadoop

- Εύκολη κλιμάκωση
- Υποστήριξη δεκάδων χιλιάδων κόμβων

## Spark

- Λιγότερο εύκολη κλιμάκωση
- Υποστήριξη χιλιάδων κόμβων σε ένα cluster



## Security

### Hadoop

- Εξαιρετικά ασφαλές
- Έλεγχος ταυτότητας με Kerberos και LDAP

### Spark

- Καθόλου ασφαλές (απενεργοποιημένη η ασφάλεια από προεπιλογή)
- Ασφαλές μόνο ενσωματώνοντας το Hadoop



## Machine Learning

### Hadoop

- Mahout για επεξεργασία δεδομένων, ομαδοποίησης, ταξινόμησης
- Samsara για αλγεβρικές λειτουργίες στη μνήμη

### Spark

- MLlib για επαναληπτικούς υπολογισμούς στη μνήμη
- Εργαλεία για εκτέλεση παλινδρόμησης, κατηγοριοποίησης, αξιολόγησης

## Scheduling & Resource Management

### Hadoop

- Εξωτερικές λύσεις για προγραμματισμό και διαχείριση πόρων
- Oozie, διαθέσιμο εργαλείο για τον προγραμματισμό ροών εργασιών
- YARN, διαθέσιμο εργαλείο για τη διαχείριση πόρων

### Spark

- Ενσωματωμένα εργαλεία για κατανομή πόρων, προγραμματισμό, παρακολούθηση
- DAG (Directed Acyclic Graph), κύριο εργαλείο για διαχωρισμό των λειτουργιών σε στάδια

# ΕΦΑΡΜΟΓΕΣ



# ΕΦΑΡΜΟΓΗ APACHE HADOOP WORD COUNT

# ΕΦΑΡΜΟΓΗ APACHE SPARK

MOVIE  
RECOMMENDATION  
SYSTEM

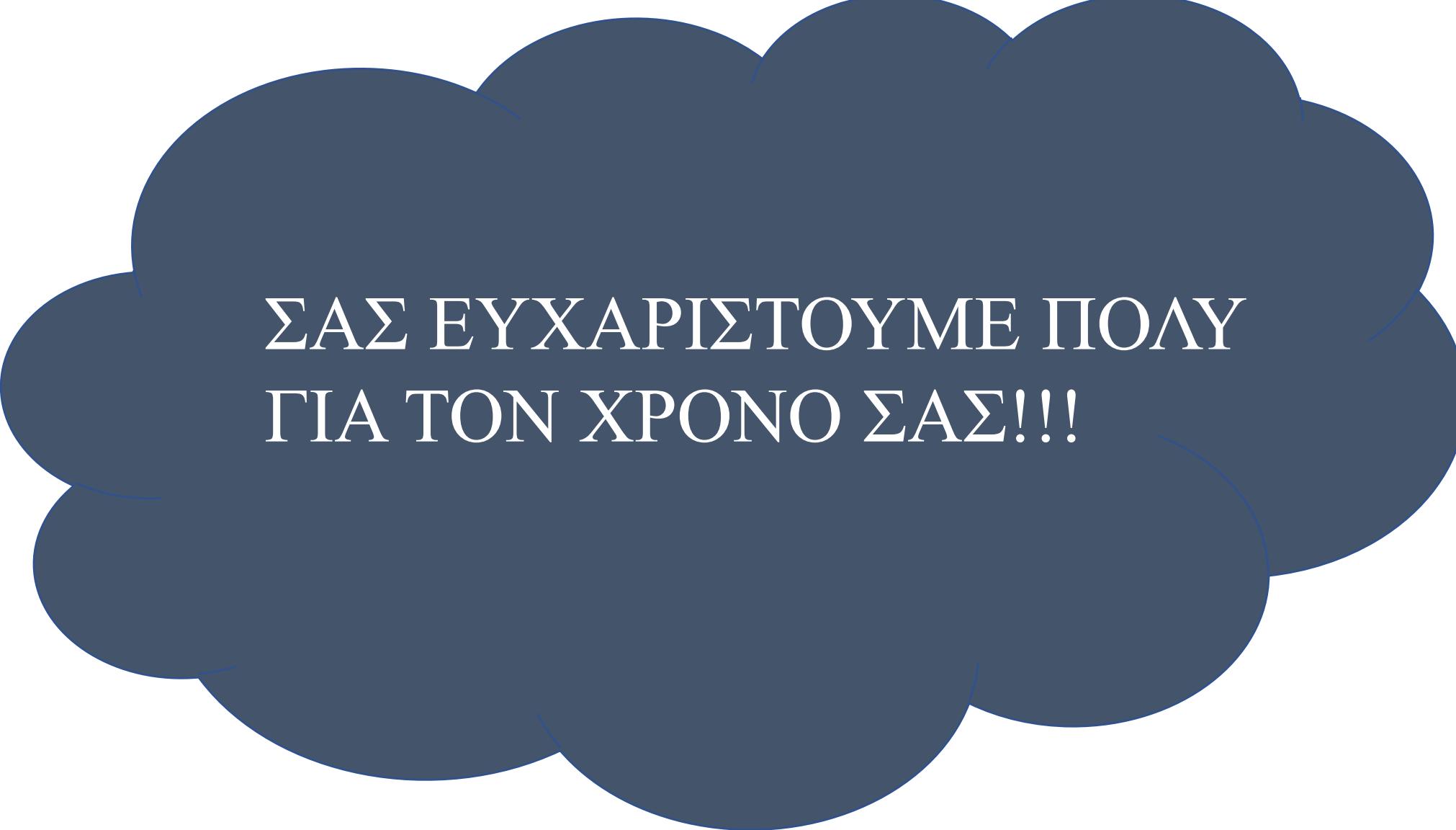
---



# ΒΙΒΛΙΟΓΡΑΦΙΑ

# ΒΙΒΛΙΟΓΡΑΦΙΑ

- D. Gewirtz, “Volume, velocity, and variety: Understanding the three V’s of big data.” 2018, [Online]. Available: <https://www.zdnet.com/article/volume-velocity-and-variety-understanding-the-three-vs-of-big-data/>.
- C. P. Kasula, “Netflix Recommender System — A Big Data Case Study.” 2020, [Online]. Available: <https://towardsdatascience.com/netflix-recommender-system-a-big-data-case-study-19cfaf6d56ff5>.
- S. Gutta, “Data Science: The 5 V’s of Big Data.” 2020, [Online]. Available: <https://suryagutta.medium.com/the-5-vs-of-big-data-2758bfcc51d>.
- R. Kiran, “Top Big Data Technologies that you Need to know.” 2020, [Online]. Available: <https://www.edureka.co/blog/top-big-data-technologies/>.
- Apache Software Foundation, “Apache Hadoop.” 2021, [Online]. Available: <http://hadoop.apache.org/>.
- Apache Software Foundation, “HDFS Architecture.” 2021, [Online]. Available: <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>.
- Apache Software Foundation, “Apache Hadoop YARN.” 2021, [Online]. Available: <https://hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html>.
- Apache Software Foundation, “MapReduce Tutorial.” 2021, [Online]. Available: <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>.
- D. Kvasnicka and G. Roda, “BIG DATA ON VSC,” p. 63, 2021.
- R. Kiran, “MapReduce Tutorial – Fundamentals of MapReduce with MapReduce Example.” 2020, [Online]. Available: <https://www.edureka.co/blog/mapreduce-tutorial/>.
- Databricks, “Apache Spark.” 2021, [Online]. Available: <https://databricks.com/glossary/what-is-apache-spark>.
- Databricks, “Apache Spark Ecosystem.” 2021, [Online]. Available: <https://databricks.com/spark/about>.
- phoenixNAP, “Hadoop vs Spark - Detailed Comparison.” 2020, [Online]. Available: <https://phoenixnap.com/kb/hadoop-vs-spark>.



ΣΑΣ ΕΥΧΑΡΙΣΤΟΥΜΕ ΠΟΛΥ  
ΓΙΑ ΤΟΝ ΧΡΟΝΟ ΣΑΣ!!!