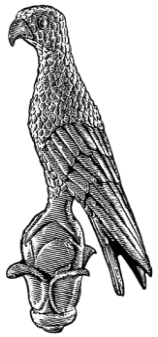


ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΣΥΓΚΡΙΤΙΚΗ ΜΕΛΕΤΗ APACHE HADOOP ΜΕ APACHE SPARK ΓΙΑ ΕΠΕΞΕΡΓΑΣΙΑ ΜΕΓΑΛΩΝ ΔΕΔΟΜΕΝΩΝ

Δημητριάδης Πρόδρομος (1359)
Νάστος Βασίλης (1525)

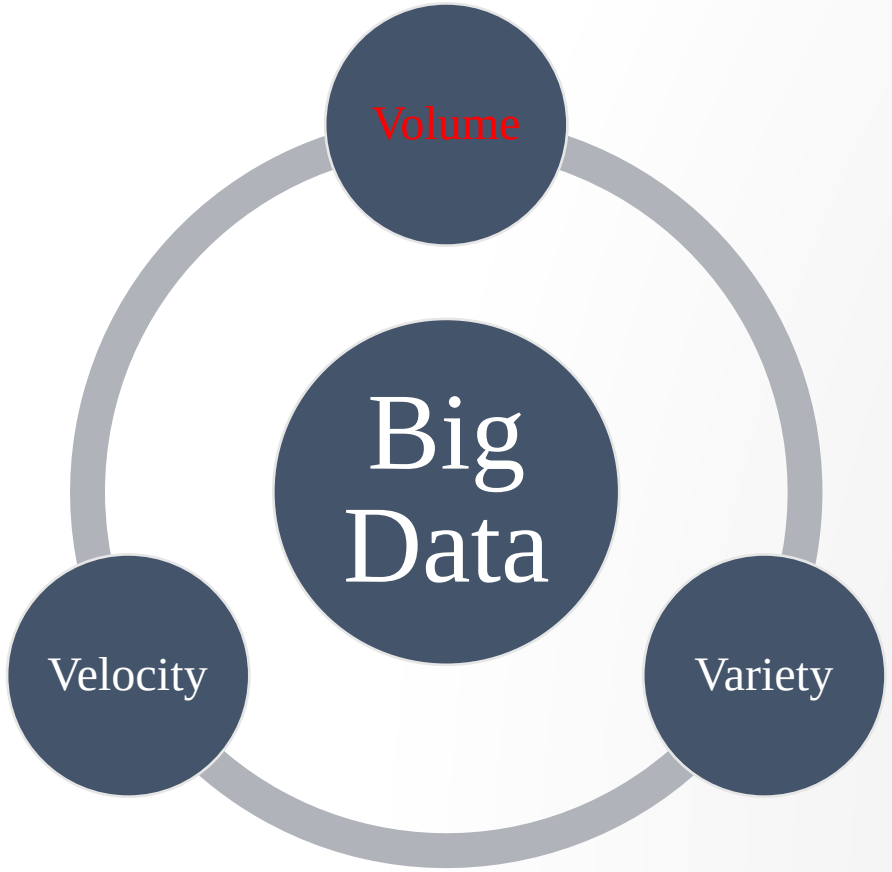
ΠΕΡΙΕΧΟΜΕΝΑ

BIG DATA

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.

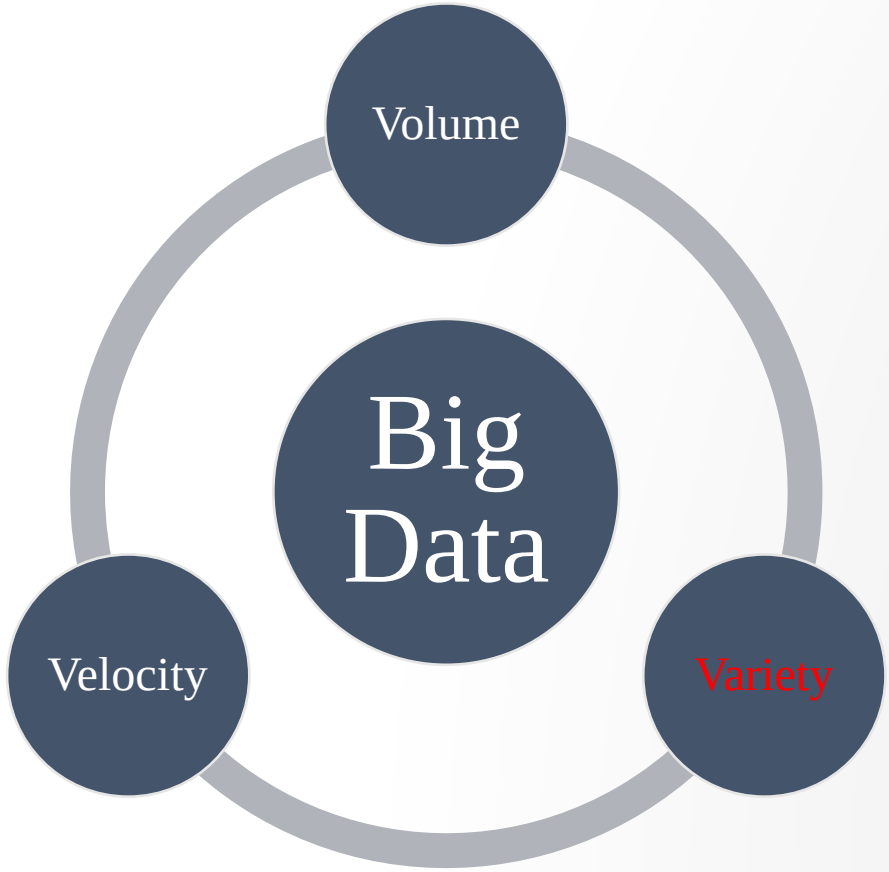
- Παραδείγματα:
  - Facebook
  - YouTube
  - Twitter
  - Instagram
  - Netflix

ΤΑ 3 V

- Υψηλός αριθμός με τον οποίο παράγονται νέα δεδομένα.

- Παραδείγματα:
  - Facebook
  - Αισθητήρες

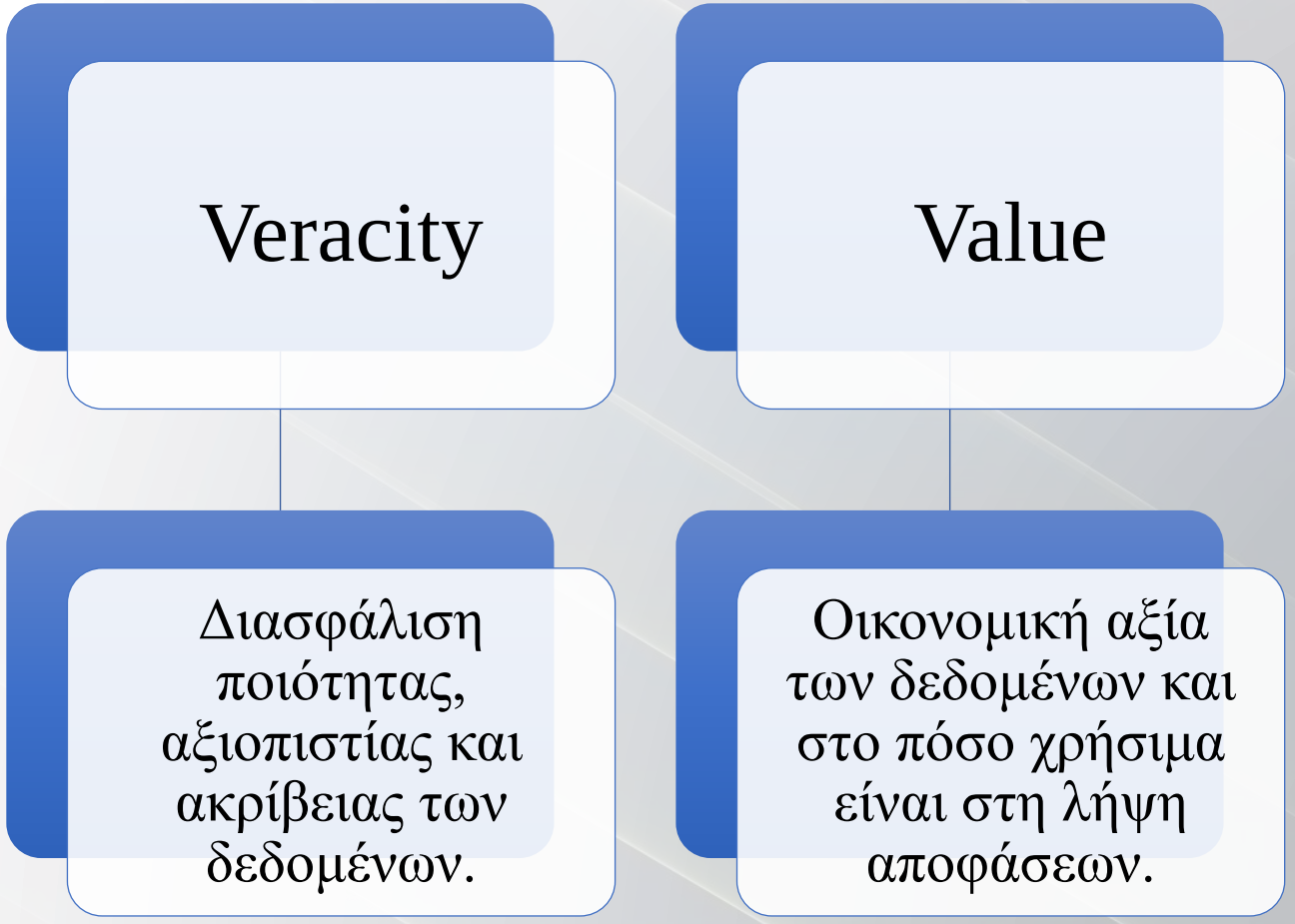
ΤΑ 3 V



- Συγκέντρωση δεδομένων από διάφορες πηγές και σε διάφορες μορφές.

- Ορισμένες μορφές δεδομένων:
  - Φωτογραφίες
  - Δεδομένα αισθητήρων
  - Tweets
  - Κρυπτογραφημένα πακέτα

ΑΚΟΜΑ 2 V


Veracity
Διασφάλιση ποιότητας, αξιοπιστίας και ακρίβειας των δεδομένων.
Value
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.

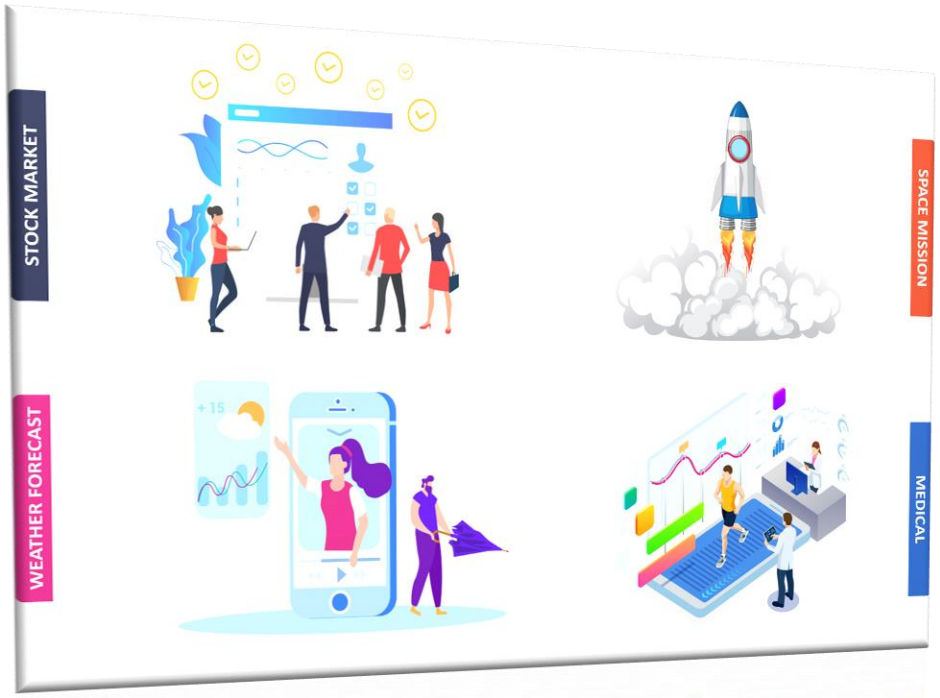
Ανάλυση τεράστιων δεδομένων σε πραγματικό χρόνο.

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 ΜΟΝΤΕΛΟΠΟΙΗΣΗΣ ΔΕΔΟΜΕΝΩΝ

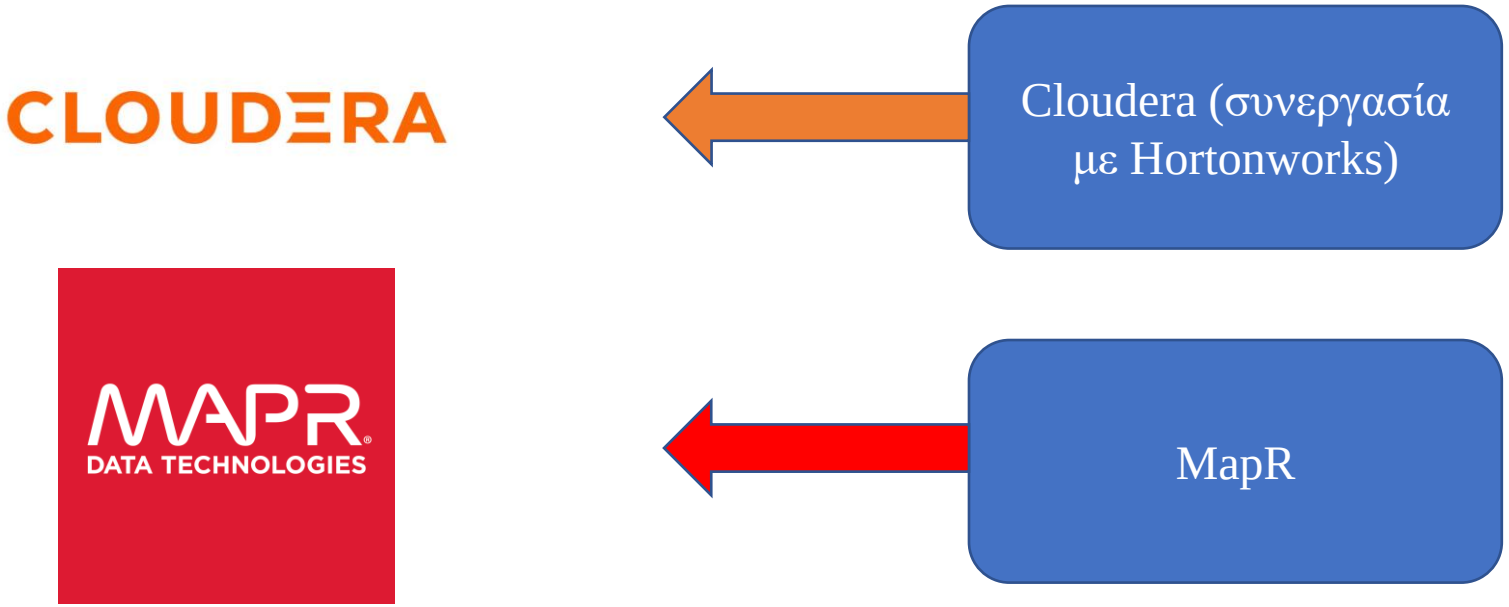
APACHE HADOOP

APACHE HADOOP

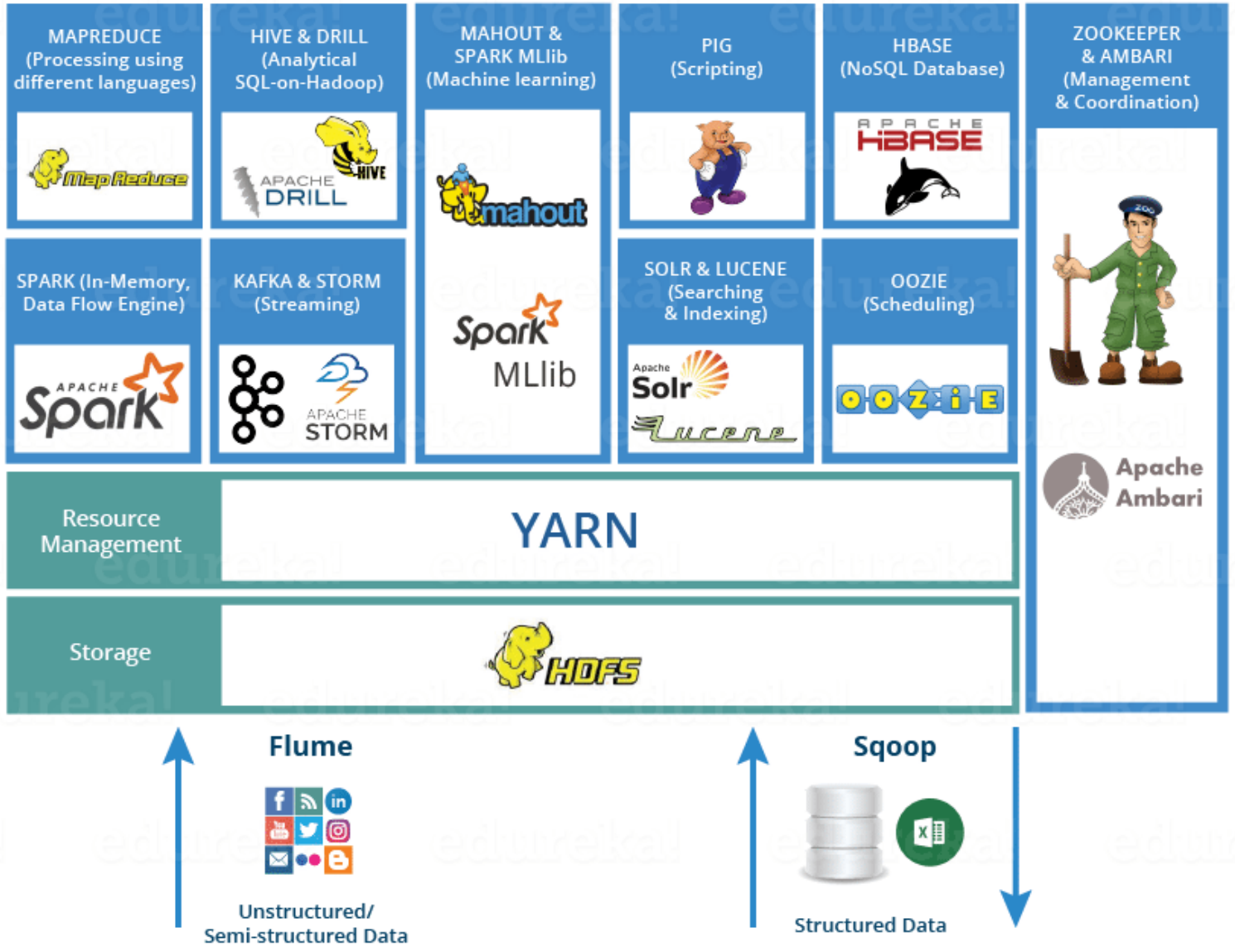
- Framework ανοιχτού κώδικα
- Σχεδιασμένο για να κάνει scale up από μεμονωμένους διακομιστές σε χιλιάδες μηχανήματα
- Εντοπίζει και χειρίζεται αποτυχίες στο επίπεδο των εφαρμογών.
- Αποτελείται από:
  - Hadoop Common
  - HDFS (Hadoop Distributed File System)
  - Hadoop YARN
  - Hadoop MapReduce

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)



ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

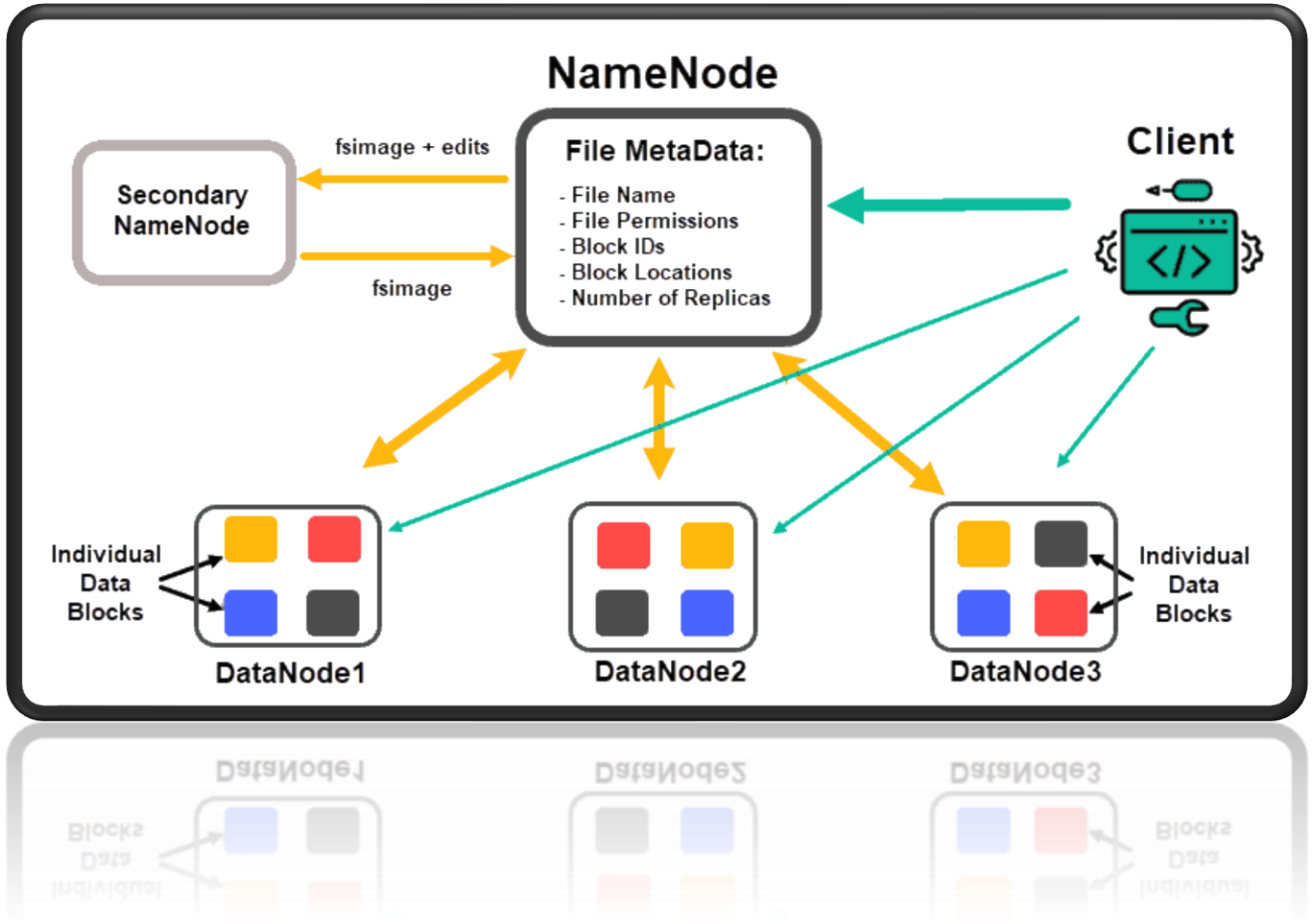


ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- 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

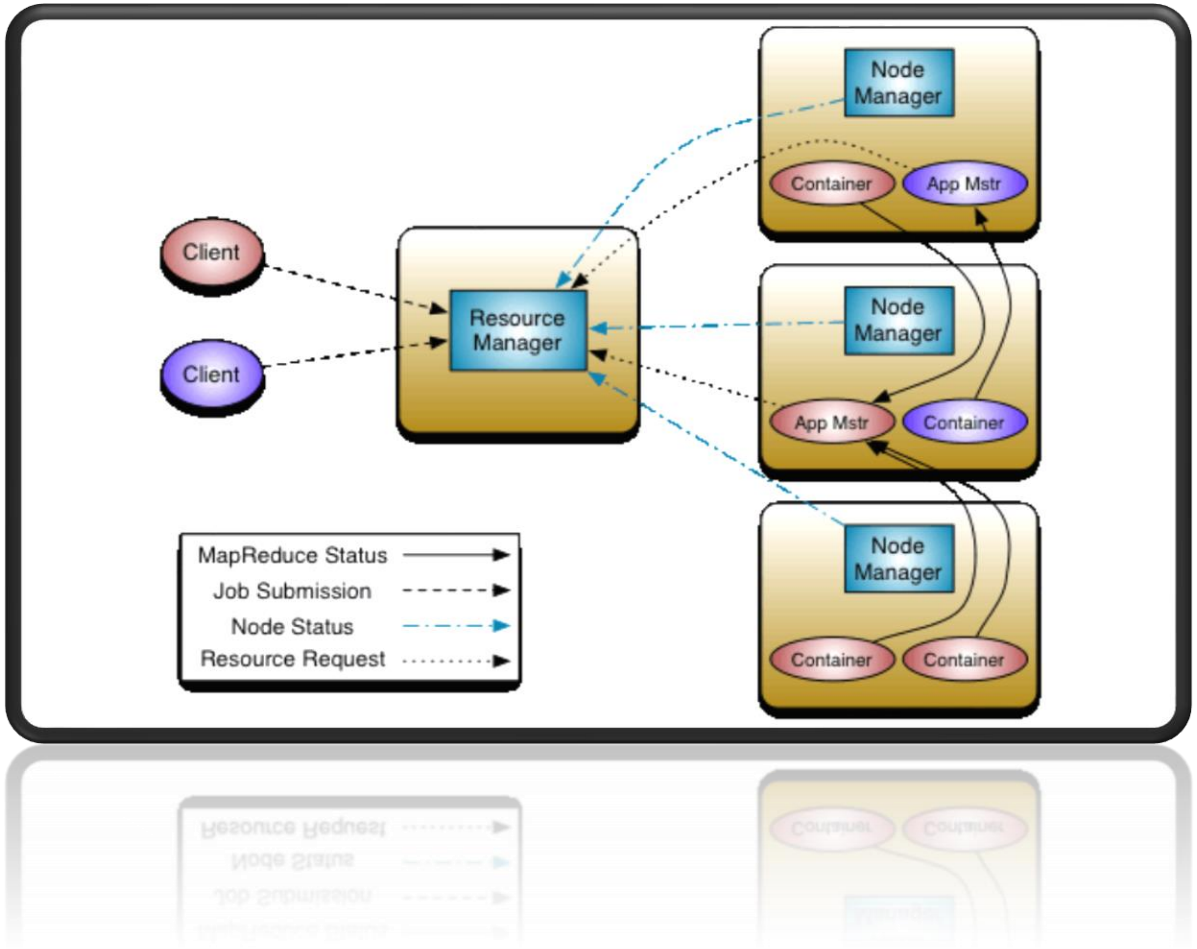


ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

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

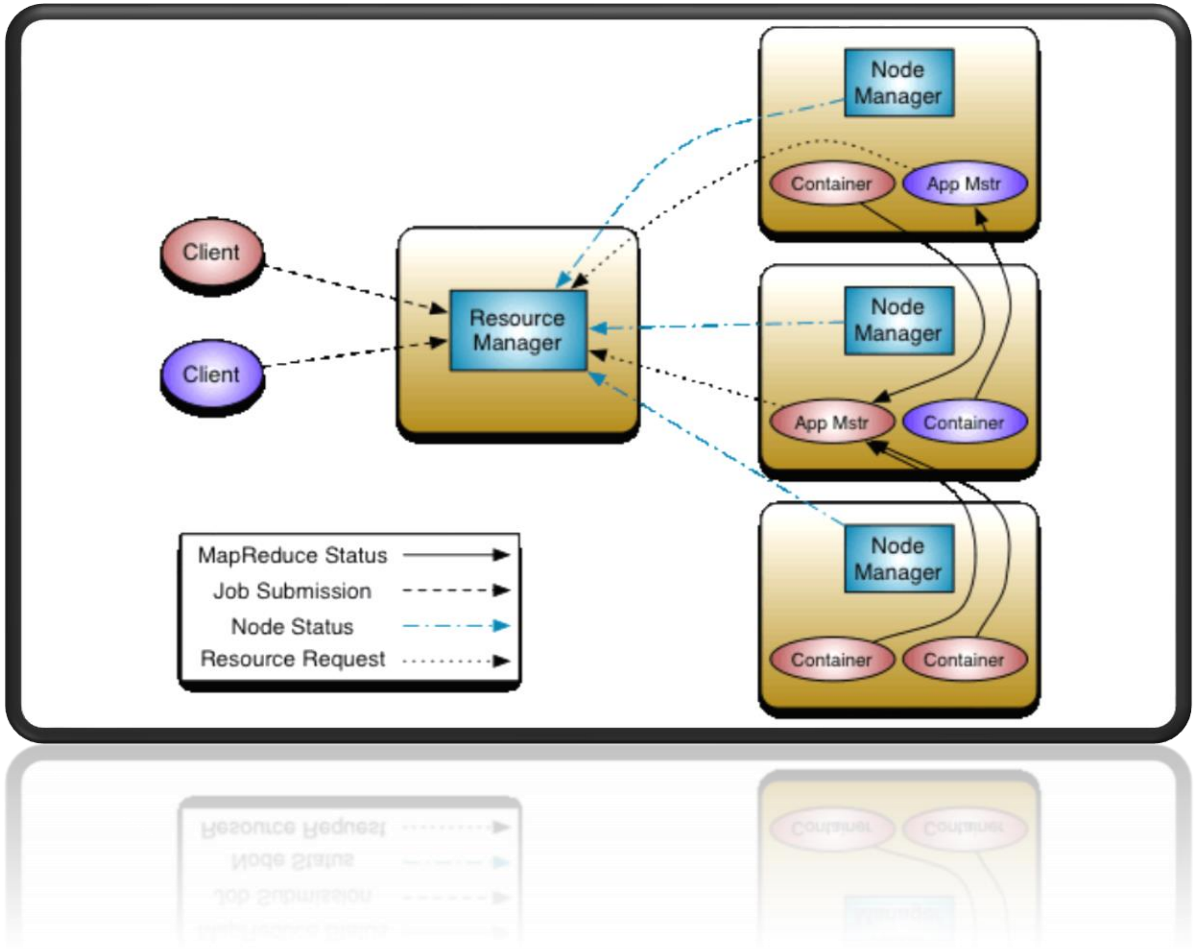
ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- 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

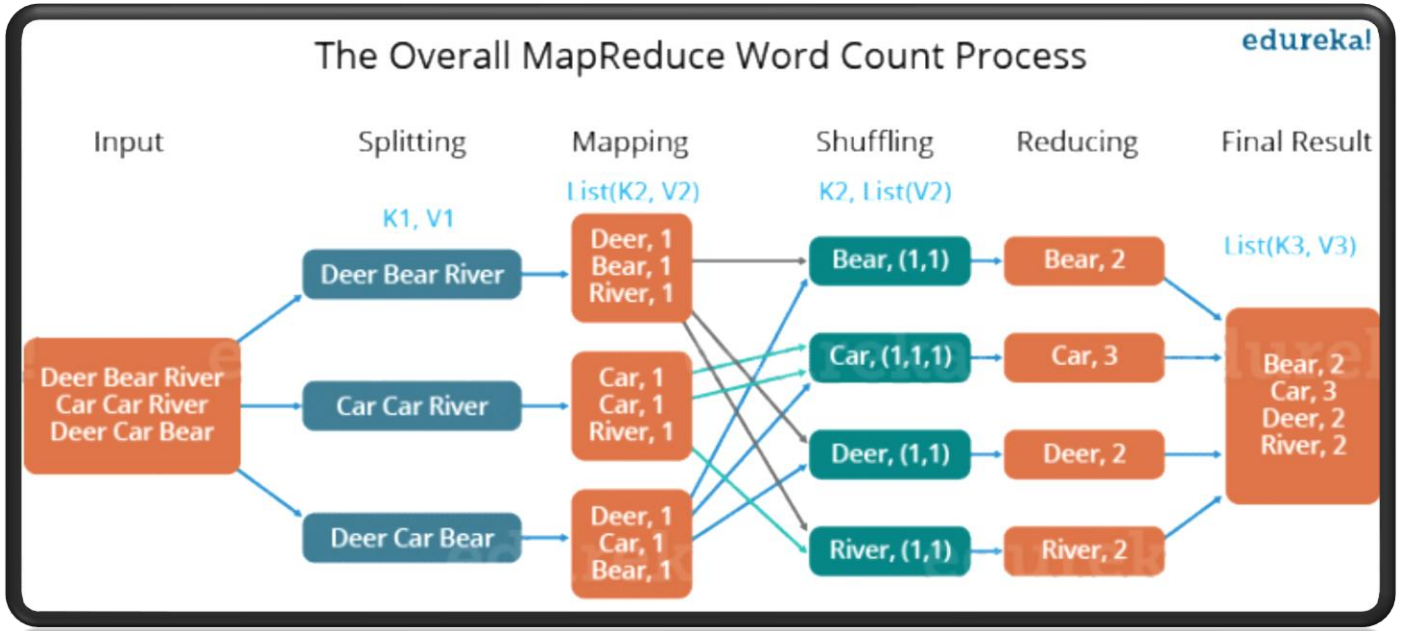


ΒΑΣΙΚΑ ΥΠΟΣΥΣΤΗΜΑΤΑ

- 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

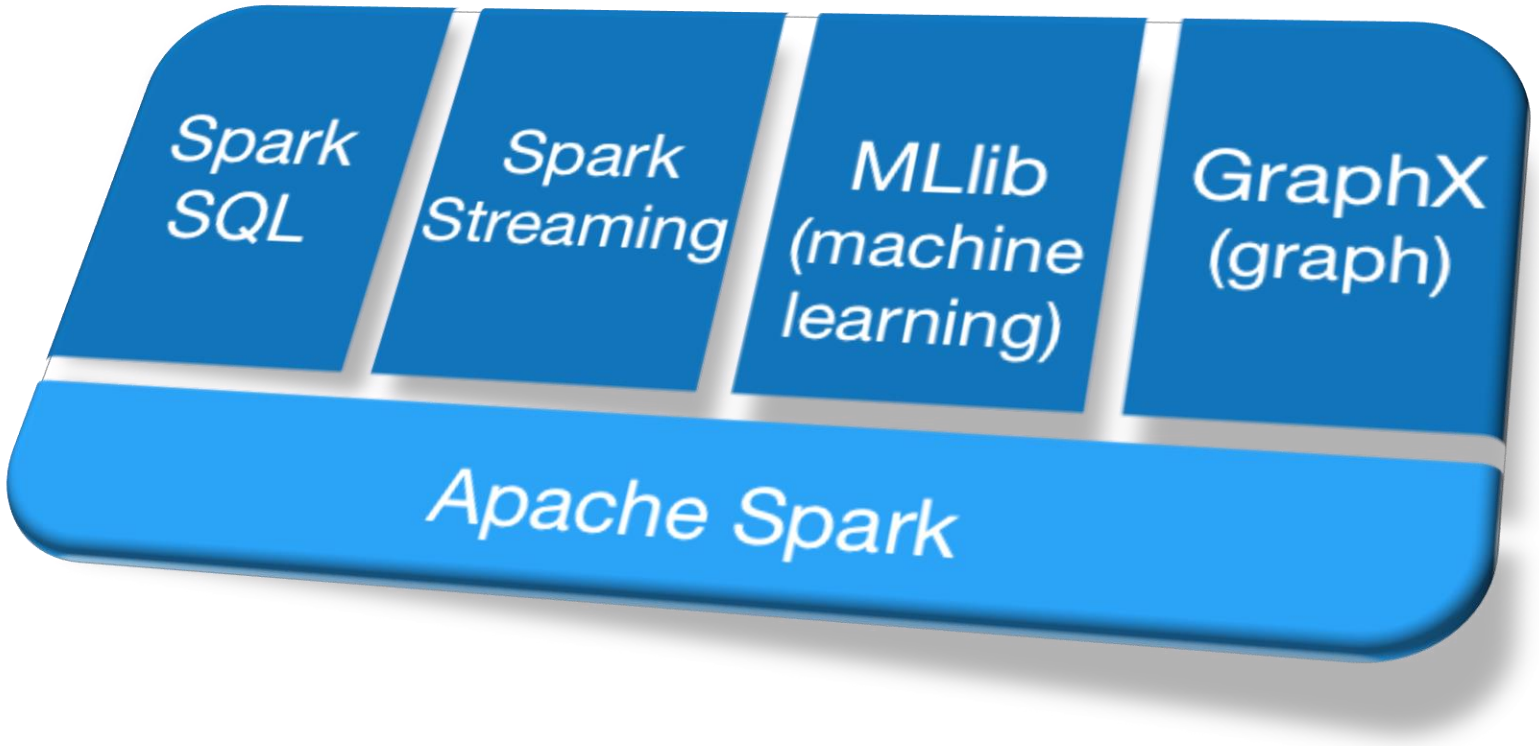
APACHE SPARK

- 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

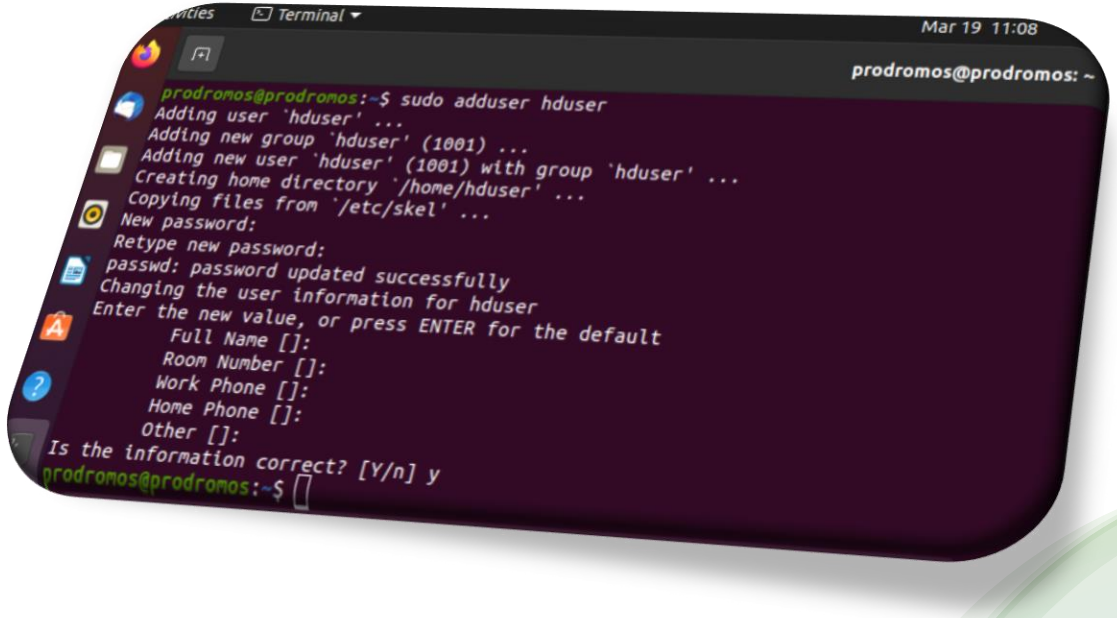


ΕΓΚΑΤΑΣΤΑΣΗ ΛΟΓΙΣΜΙΚΟΥ

ΕΓΚΑΤΑΣΤΑΣΗ APACHE HADOOP

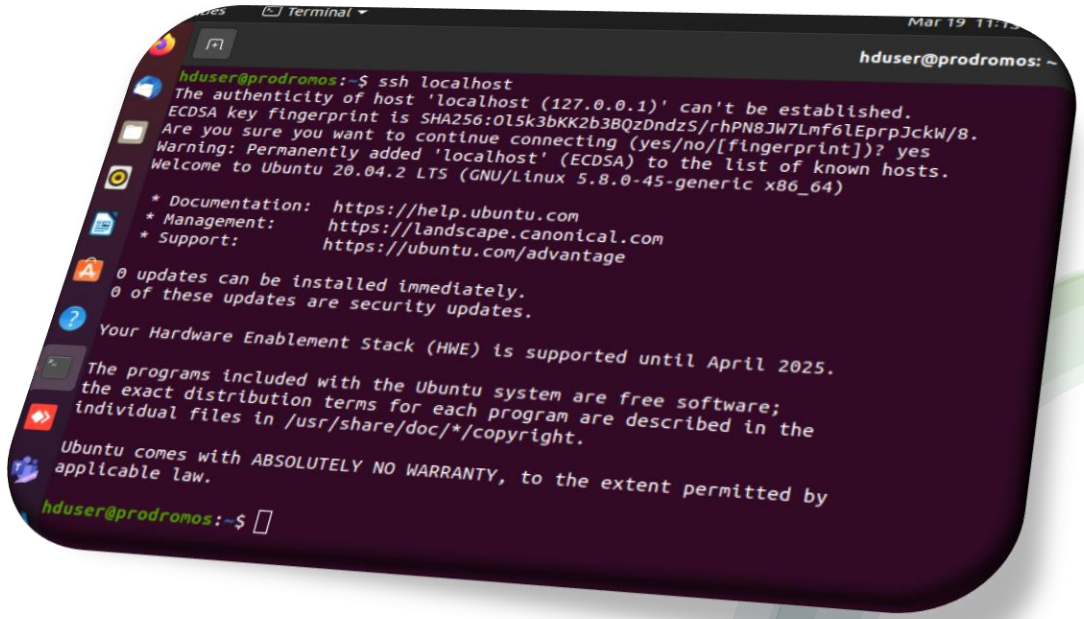
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- 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`

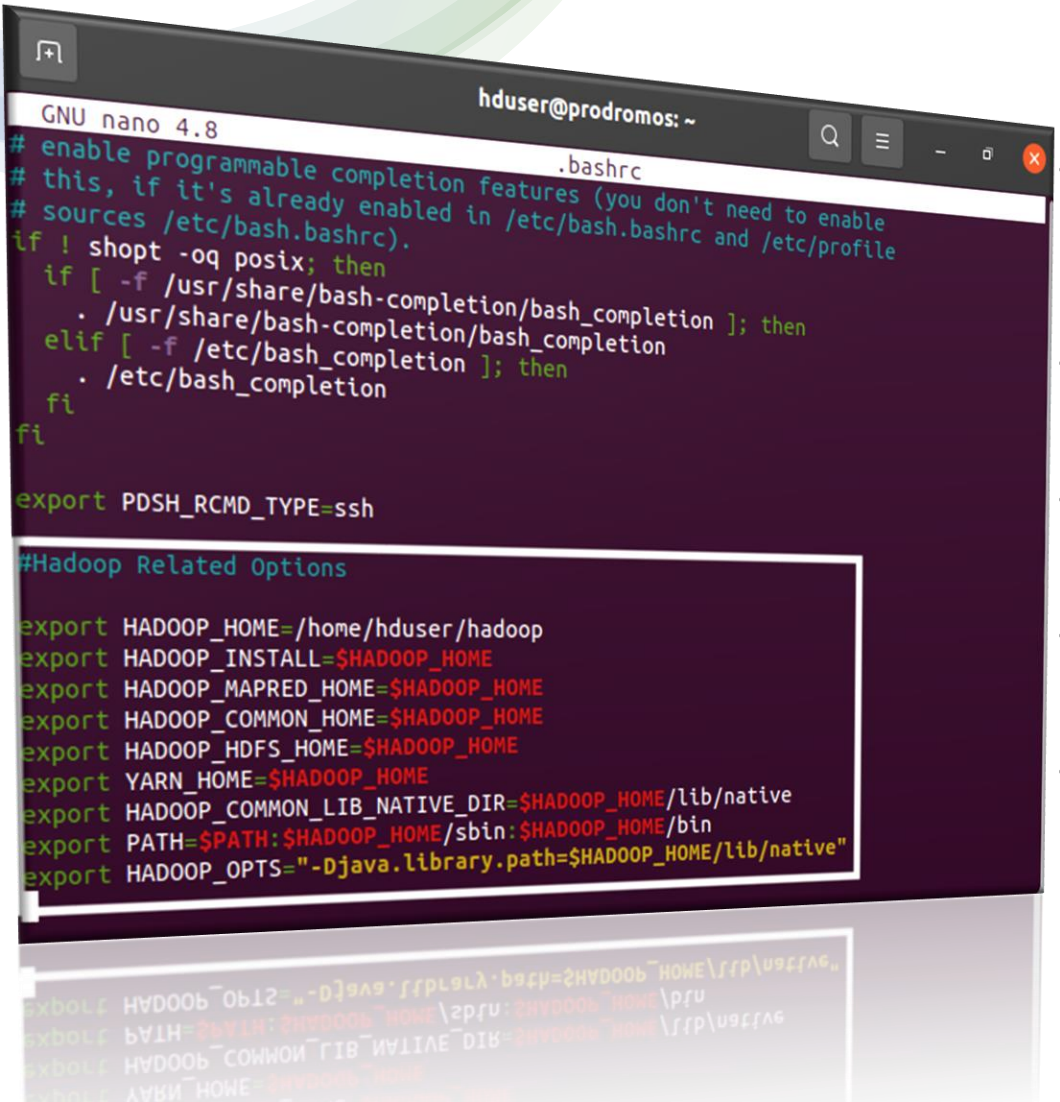


ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- 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`

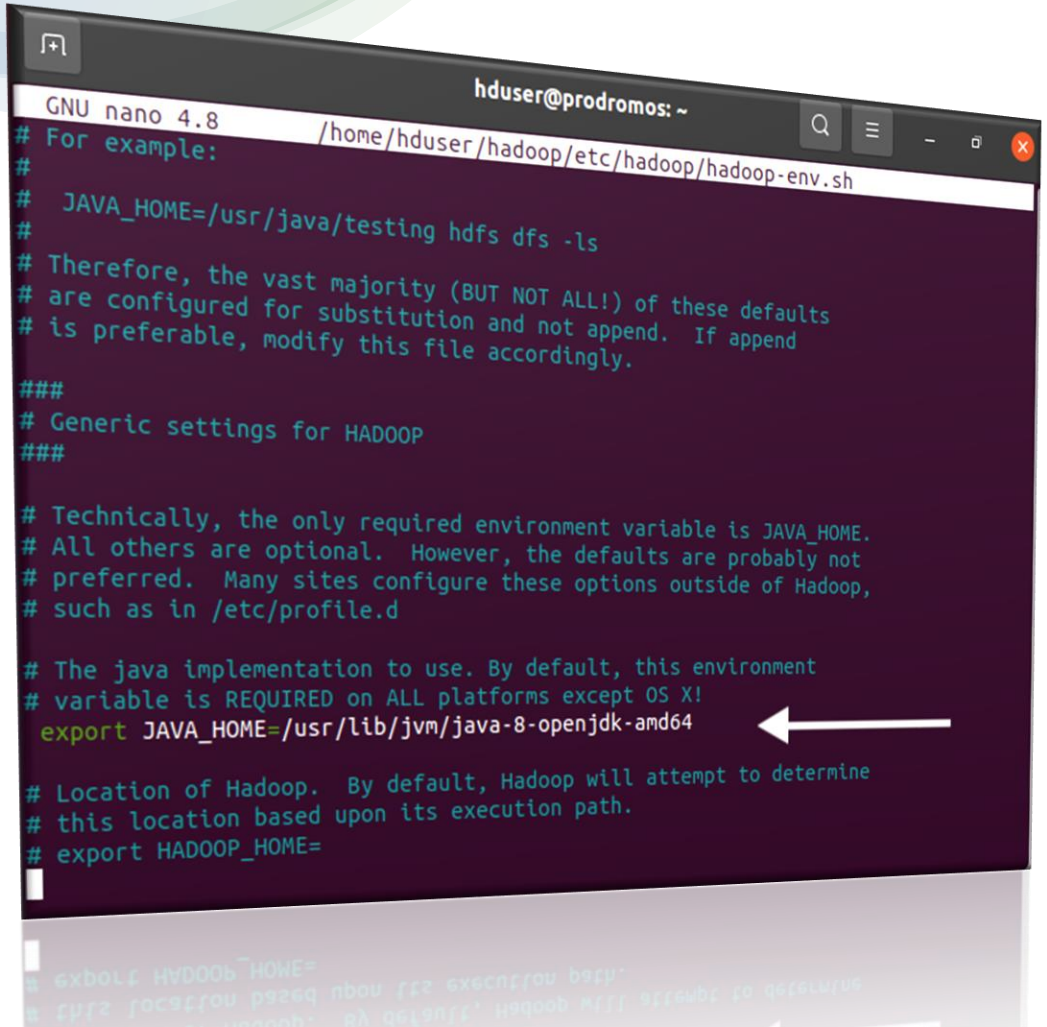


ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE



- 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-3.2.1σε 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



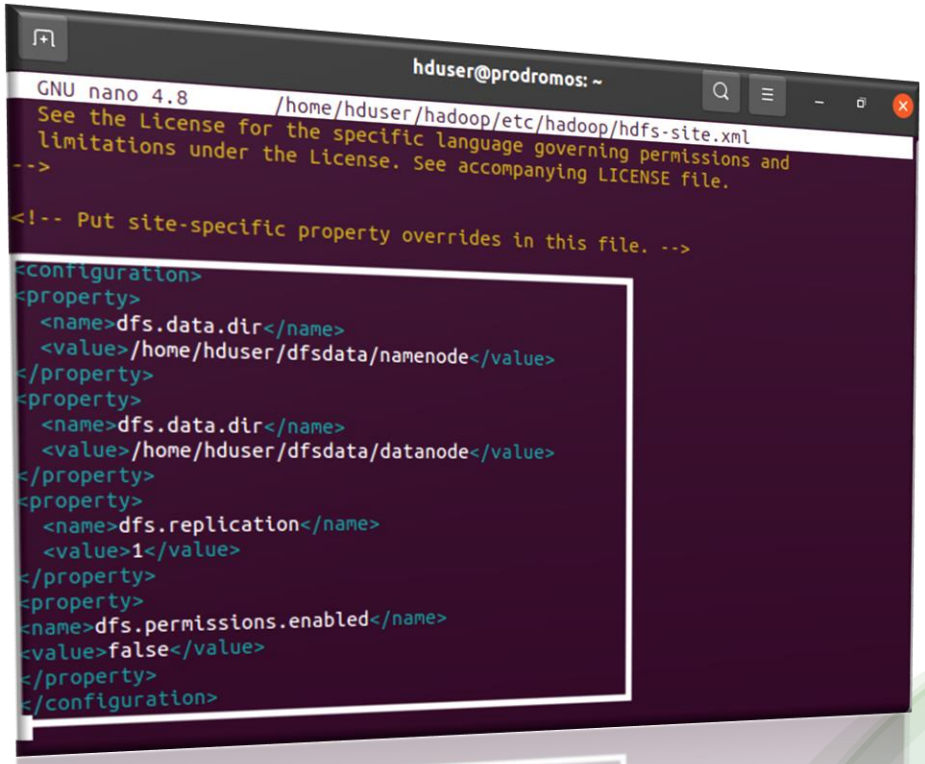
- Επεξεργασία αρχείου hadoop-env.sh:
  `sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh`
  - Export το path της Java

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`

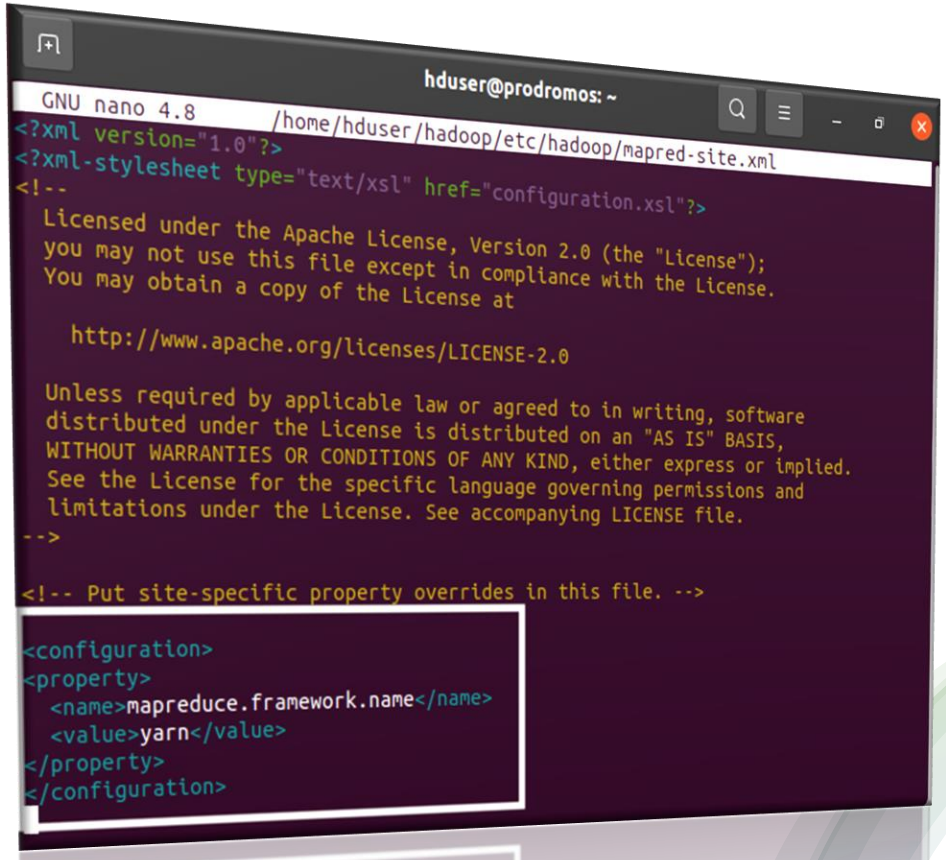
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- 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`



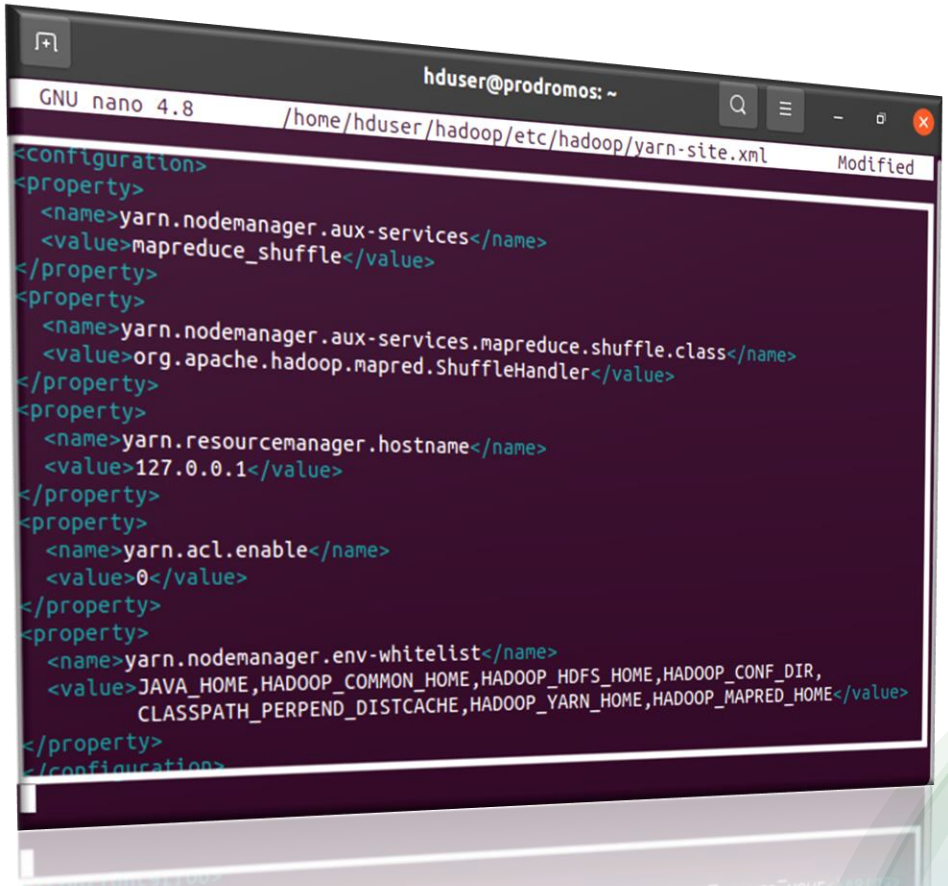
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

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



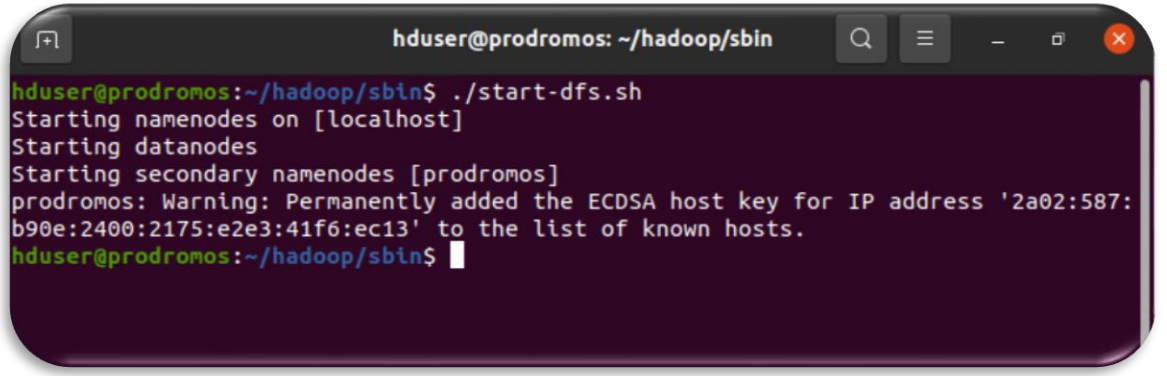
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

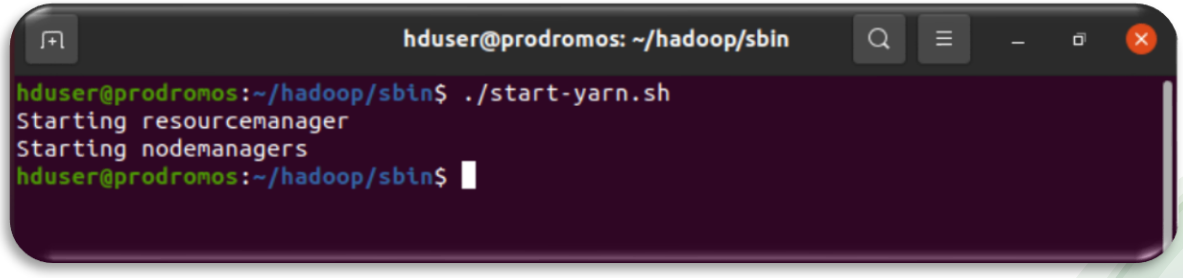
- Επεξεργασία αρχείου yarn-site.xml:
  `sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml`



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`





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-3.2.1σε 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:/sb
    in:/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`
  - `hdoop1@pd-slave01`
  - `hdoop2@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`

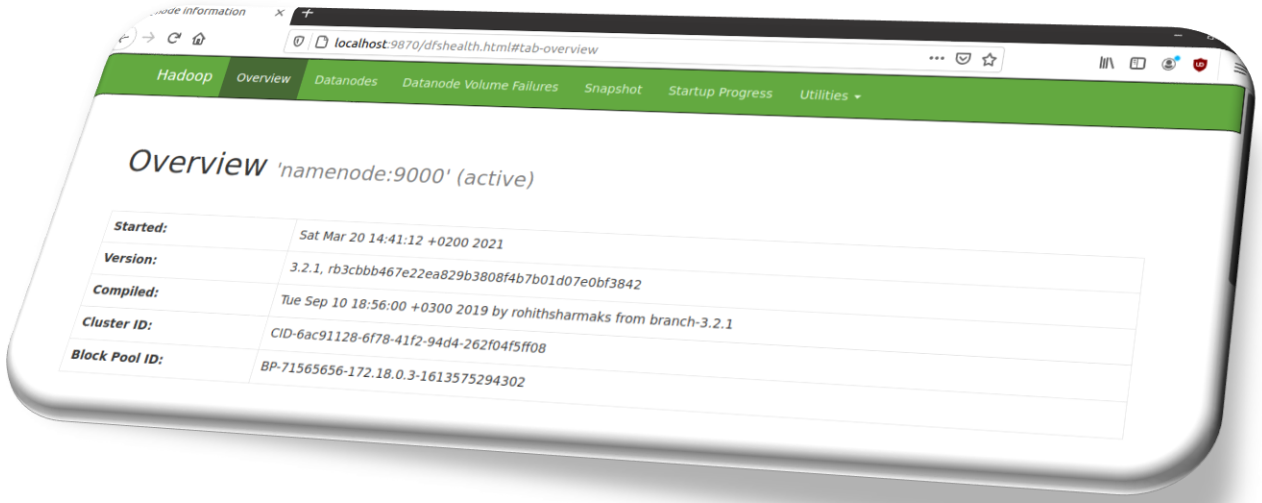
- Επεξεργασία αρχείου yarn-site.xml των slaves:
  `sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml`
  - ```
    <property>
      <name>yarn.resourcemanager.hostname</name>
      <value>hadoop-master</value>
    </property>
    ```
    ```

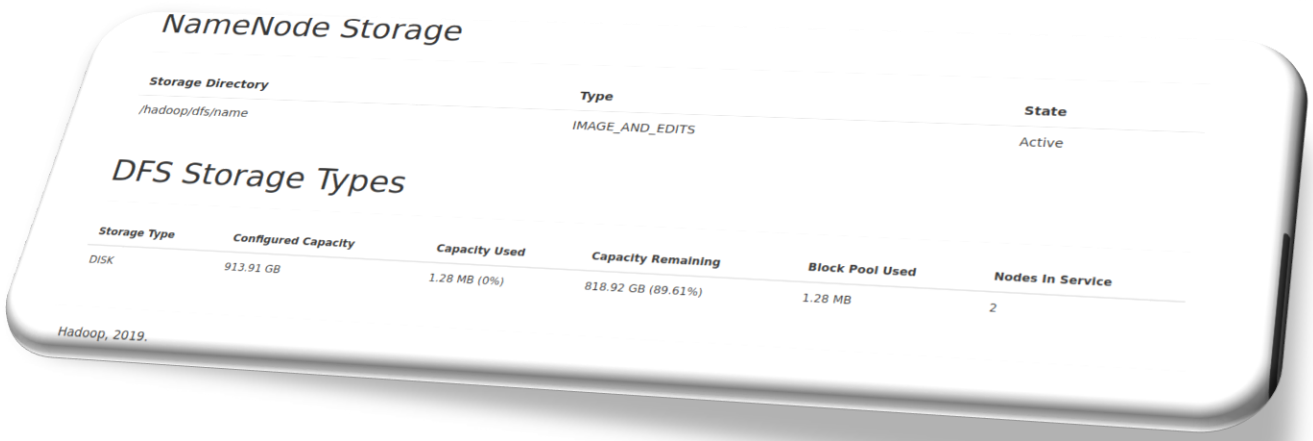
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

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

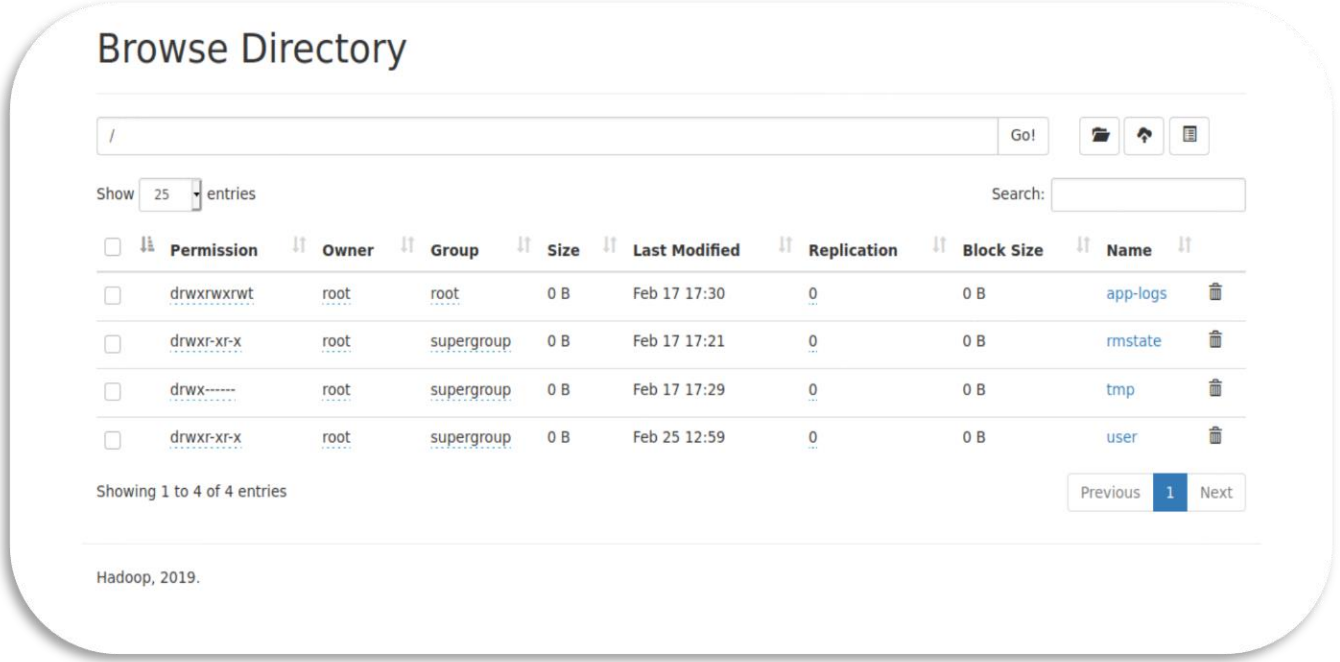
ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

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



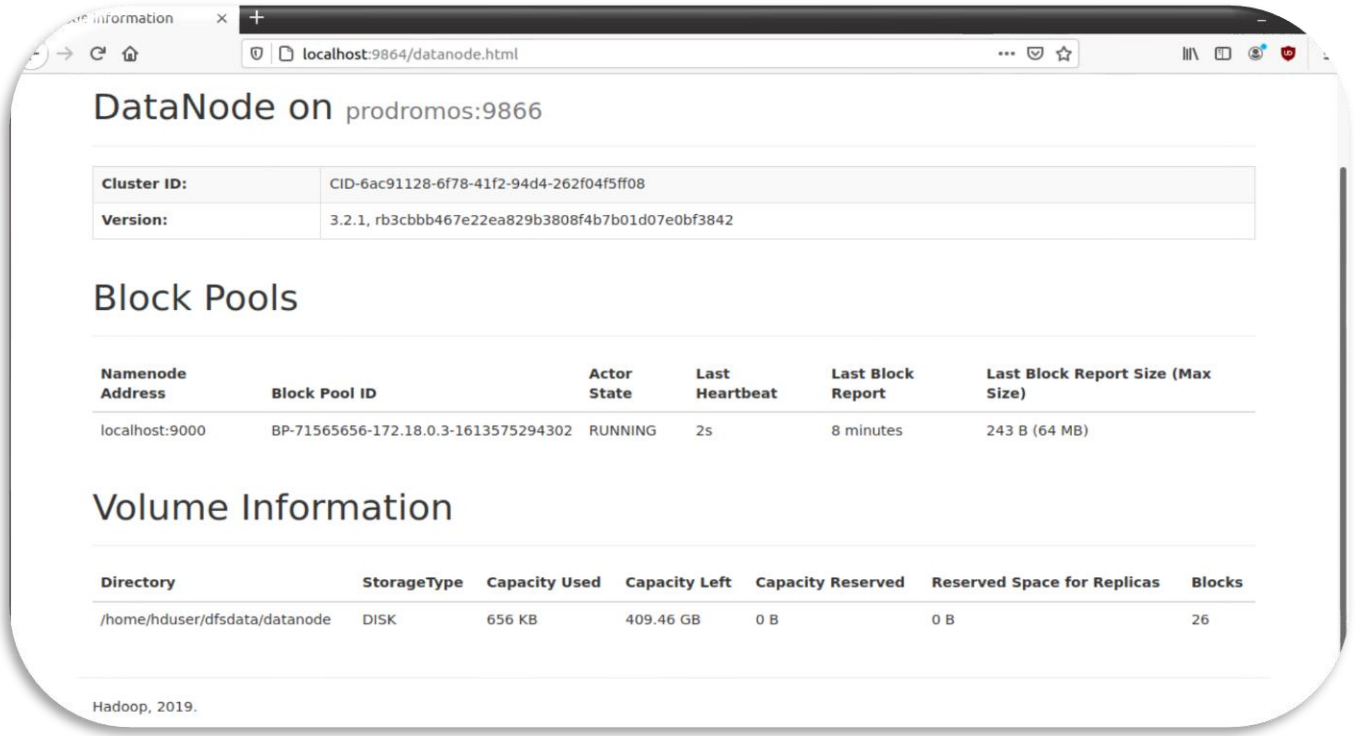


ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ



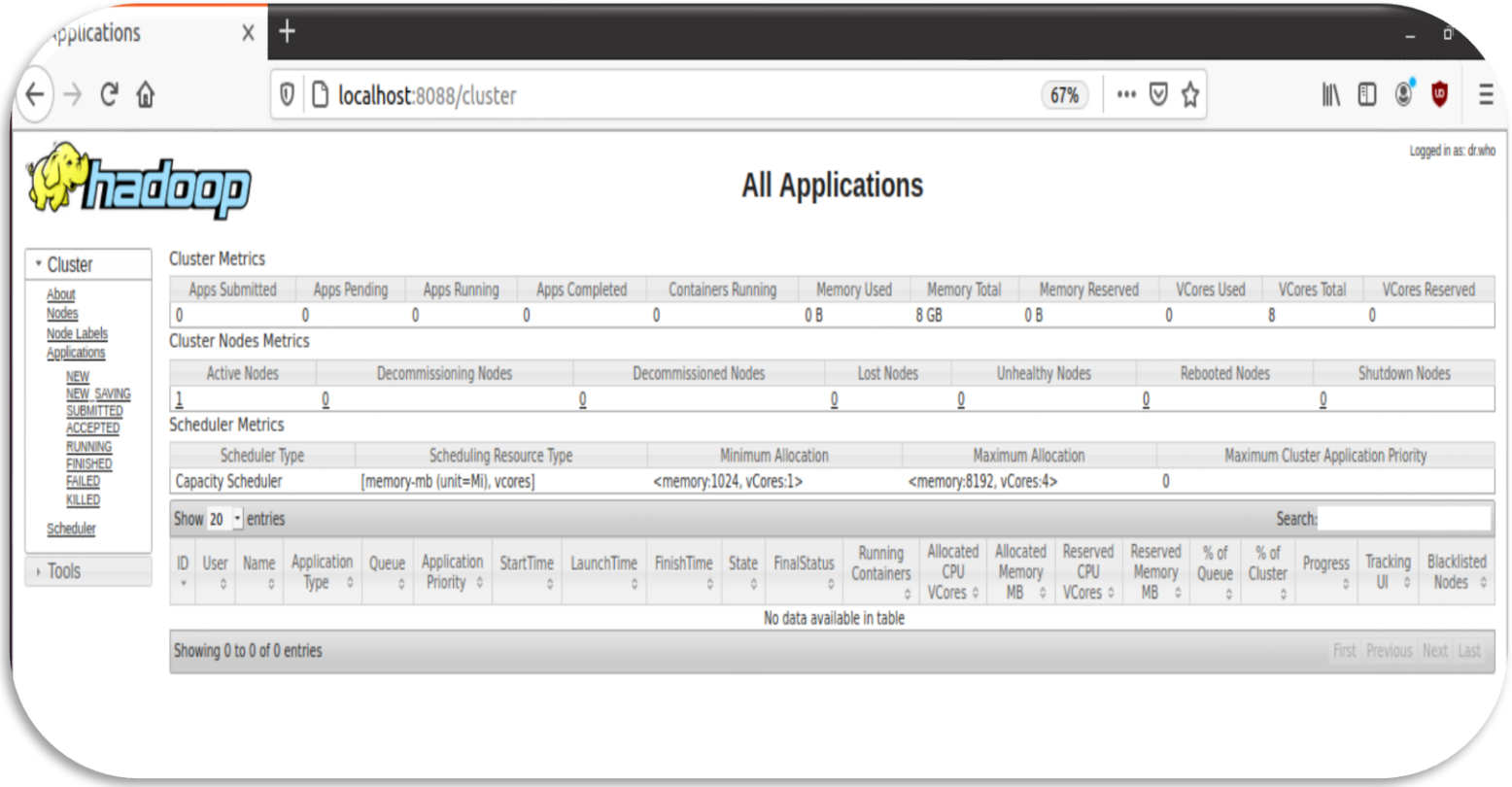
ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

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



ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ

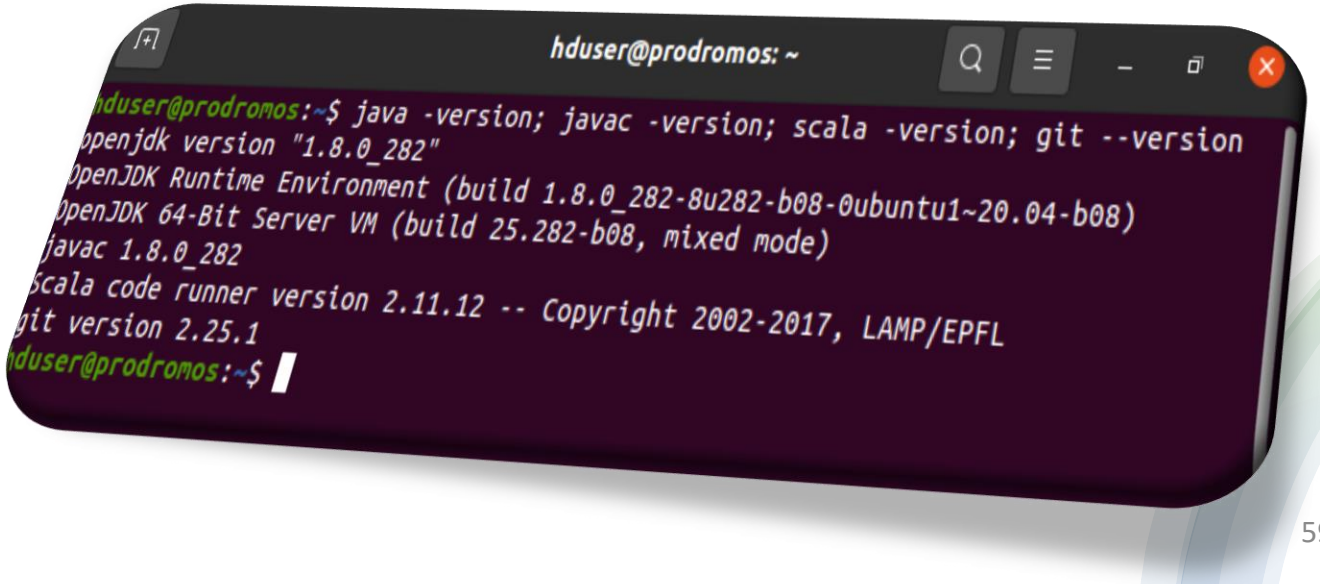
- Πρόσβαση στο Hadoop YARN UI: http://localhost:8080



ΕΓΚΑΤΑΣΤΑΣΗ APACHE SPARK

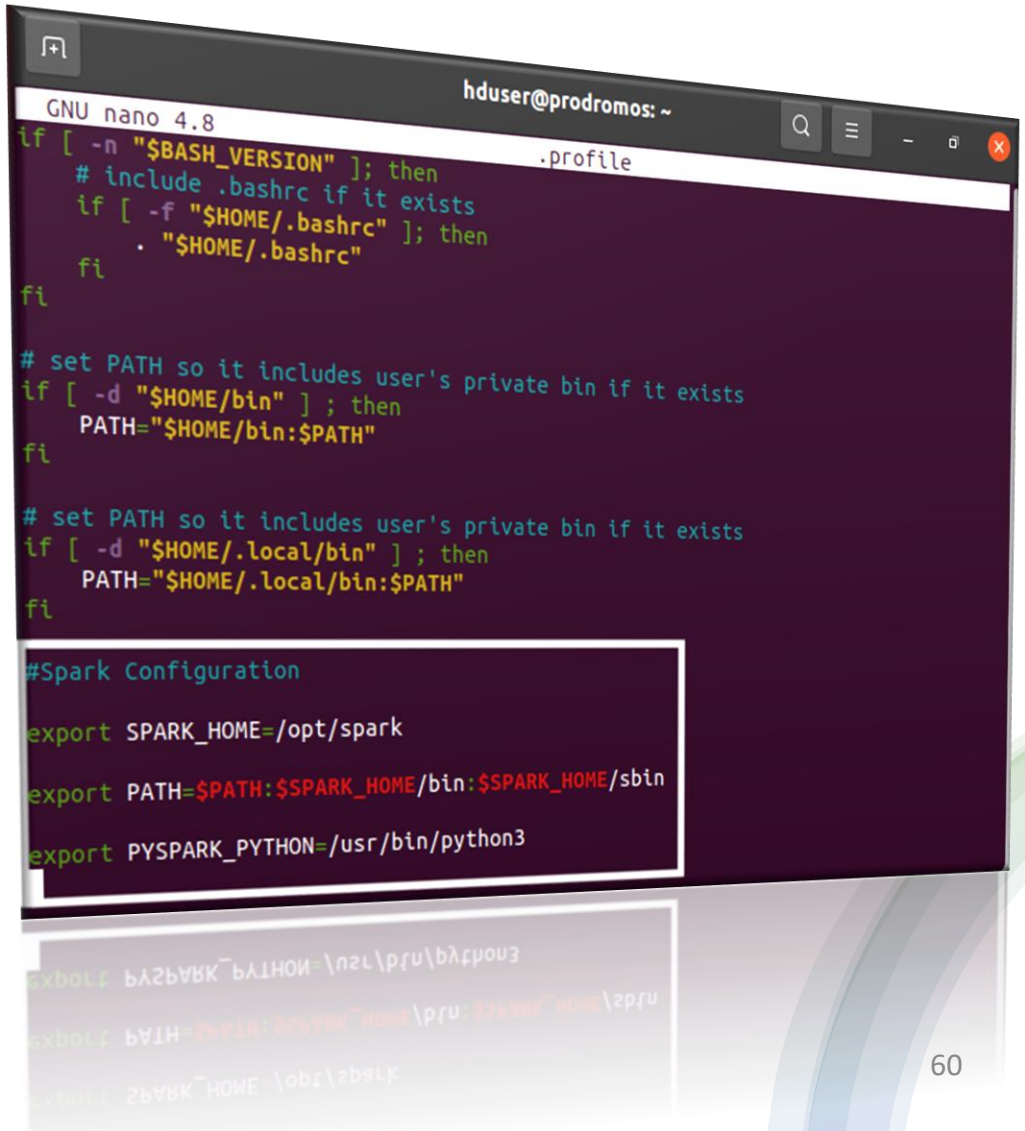
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

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



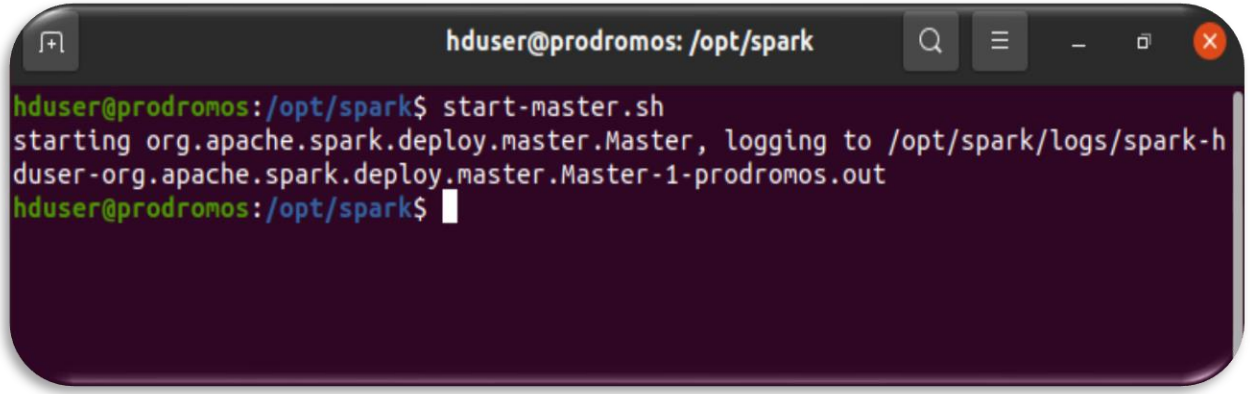
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

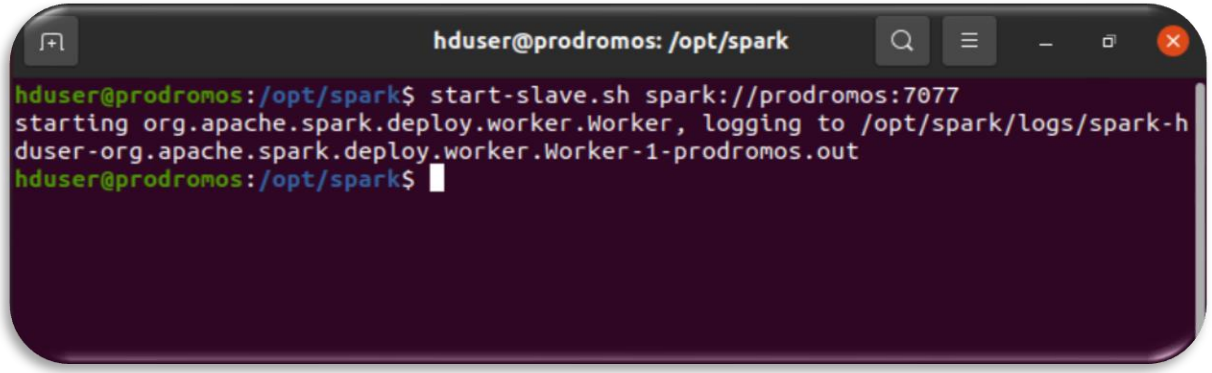
- Εγκατάσταση Spark: `wget https://downloads.apache.org/spark/spark-3.1.1/spark-3.1.1-bin-hadoop3.2.tgz`
- Αποσυμπίεση αρχείου Spark: `tar xvf spark-*`
- Αλλαγή ονόματος από spark-3.0.1-bin-hadoop2.7 σε spark: `mv ./spark-3.0.1-bin-hadoop2.7 ./spark`
- Μετάβαση αρχείου στον κατάλογο opt/spark: `sudo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



ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

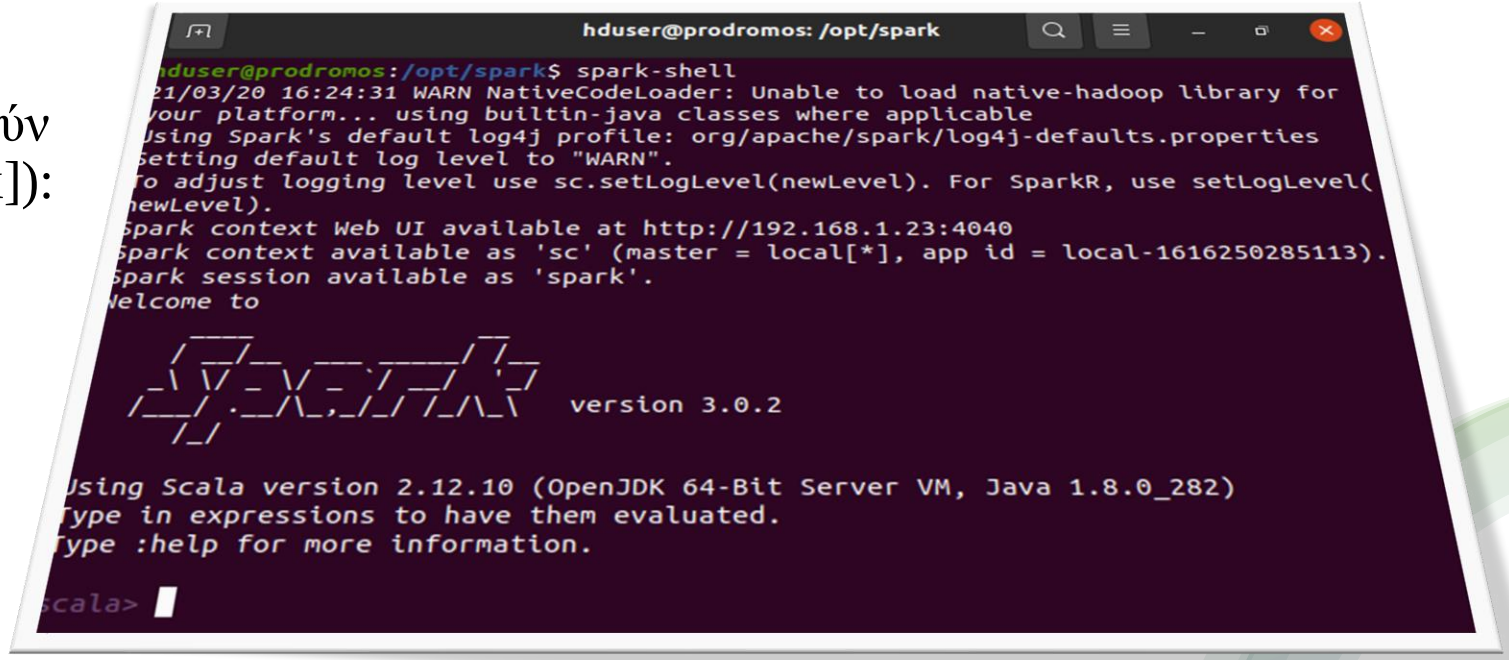
- Μετάβαση στον κατάλογο του spark:
  `cd ../../opt/spark`
- Εκκίνηση master server: `start-master.sh`
- Εκκίνηση slave server:
  `start-slave.sh spark://prodromos:7077`





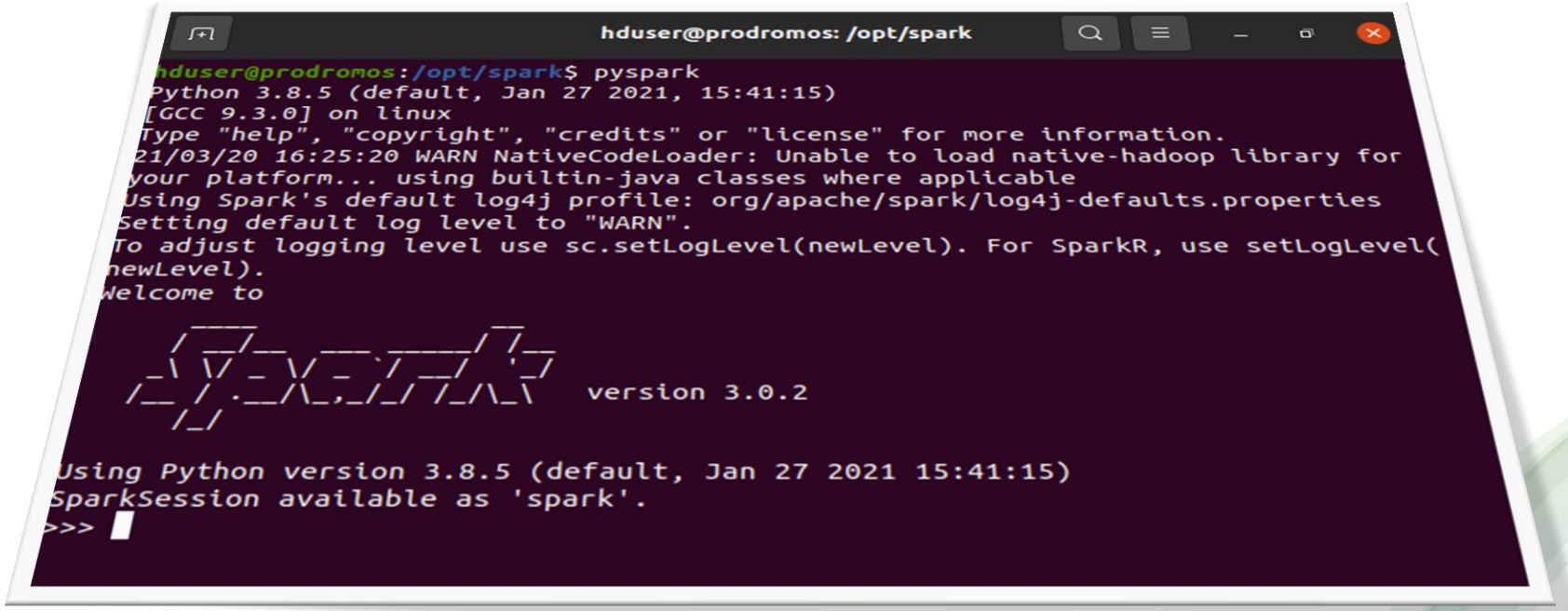
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

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



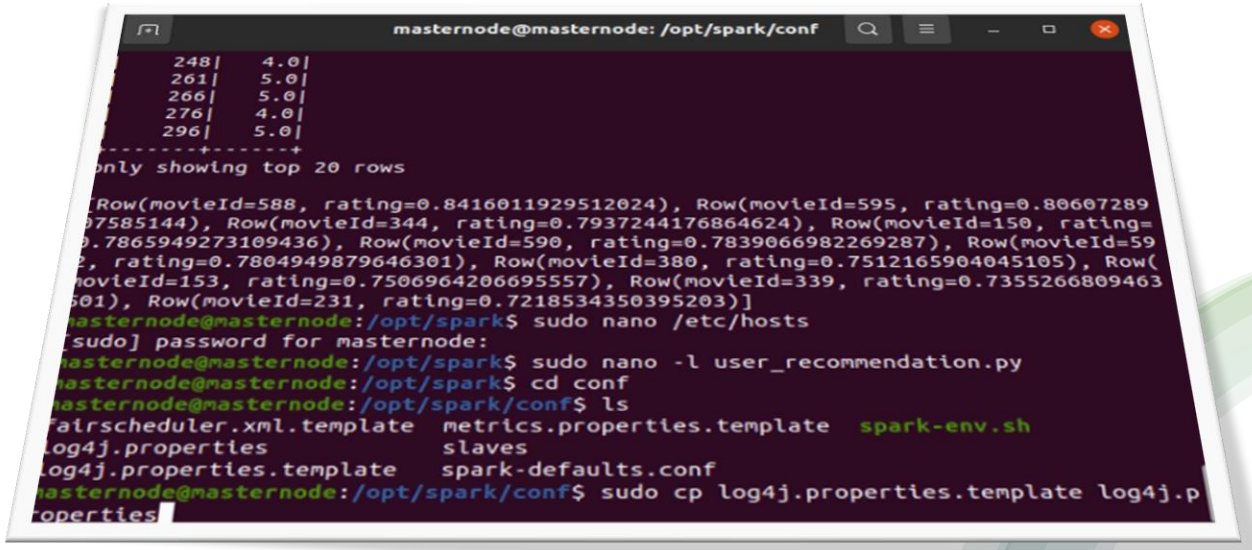
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ SINGLE NODE

- Εκκίνηση pyspark: `pyspark`



ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

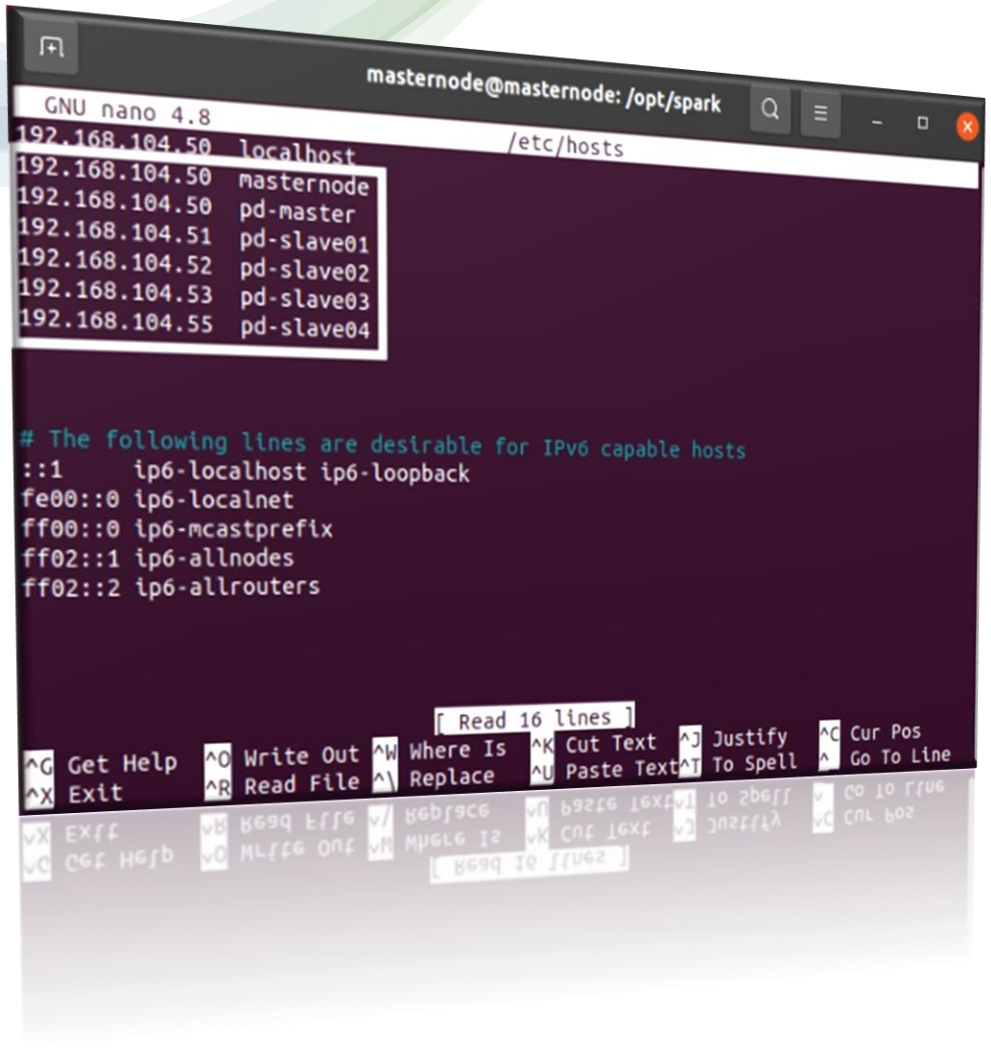
- Αντιγραφή των περιεχομένων των template αρχείων σε καινούργια shell αρχεία (.sh) με το ίδιο όνομα:
  - `sudo nano spark-env.template spark-env.sh`
- Εκτελούμε επίσης τις παρακάτω εντολές:
  - `sudo cp spark-defaults.conf.template spark-defaults.conf`
  - `sudo cp slaves.template slaves`



ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- 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



- 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`

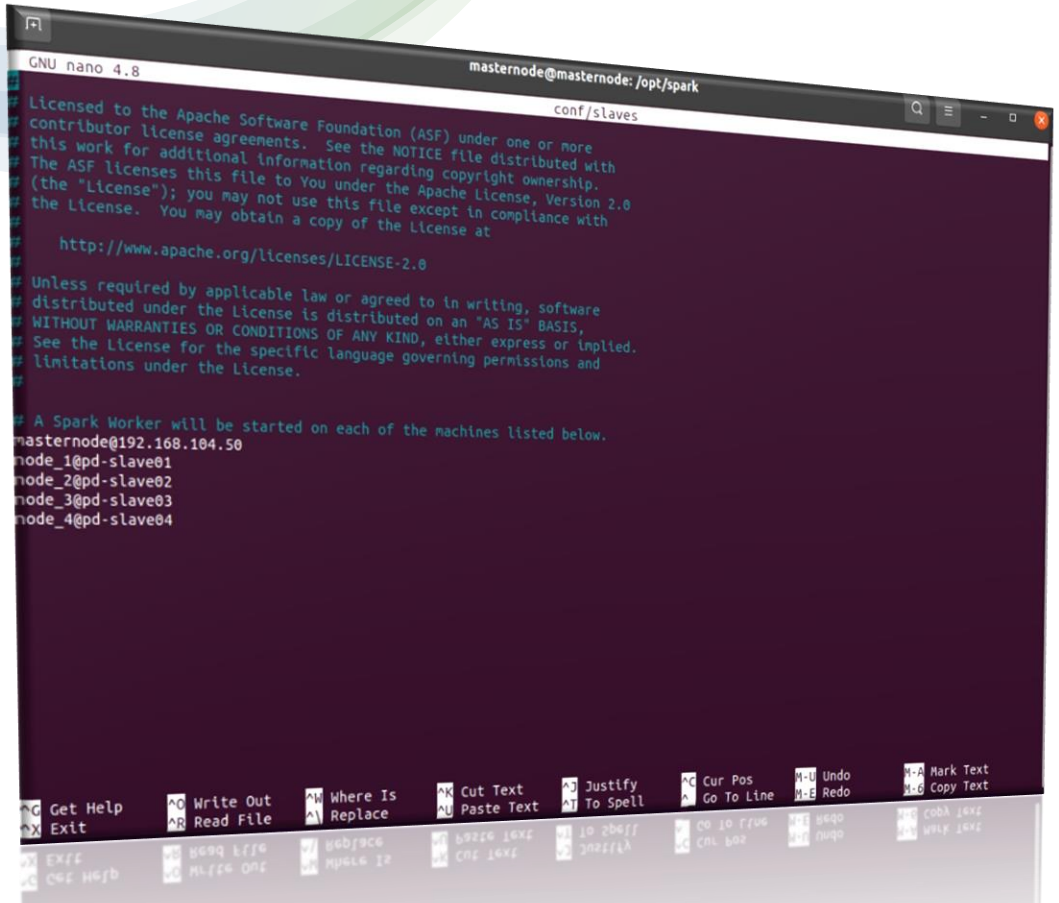
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE

- 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

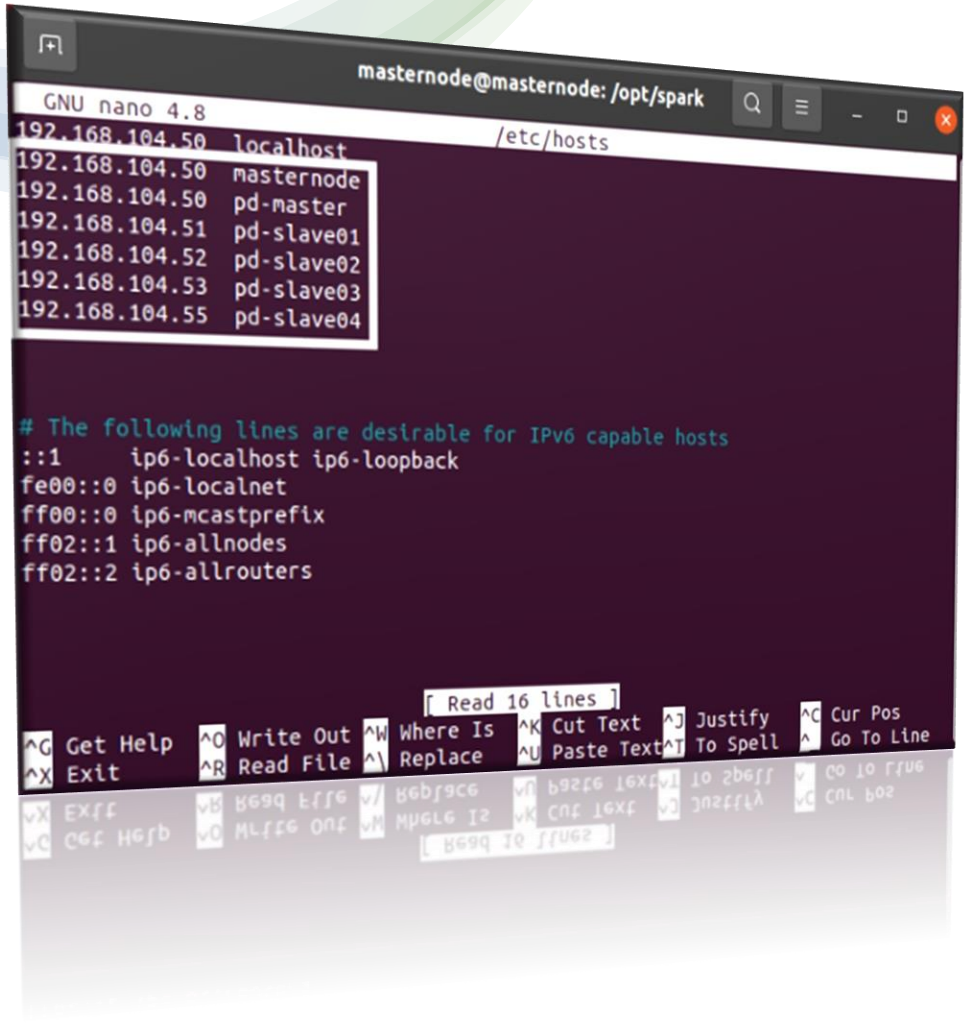
- Είσοδος στον 1ο slave: `ssh user@pd-slave1`
- Επιστροφή στον master: `exit`
- Είσοδος στο bashrc: `sudo nano ~/.bashrc`
- Export το path του spark:
  `export PATH=$PATH:/opt/spark/bin`

ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE



- Ορισμός των workers

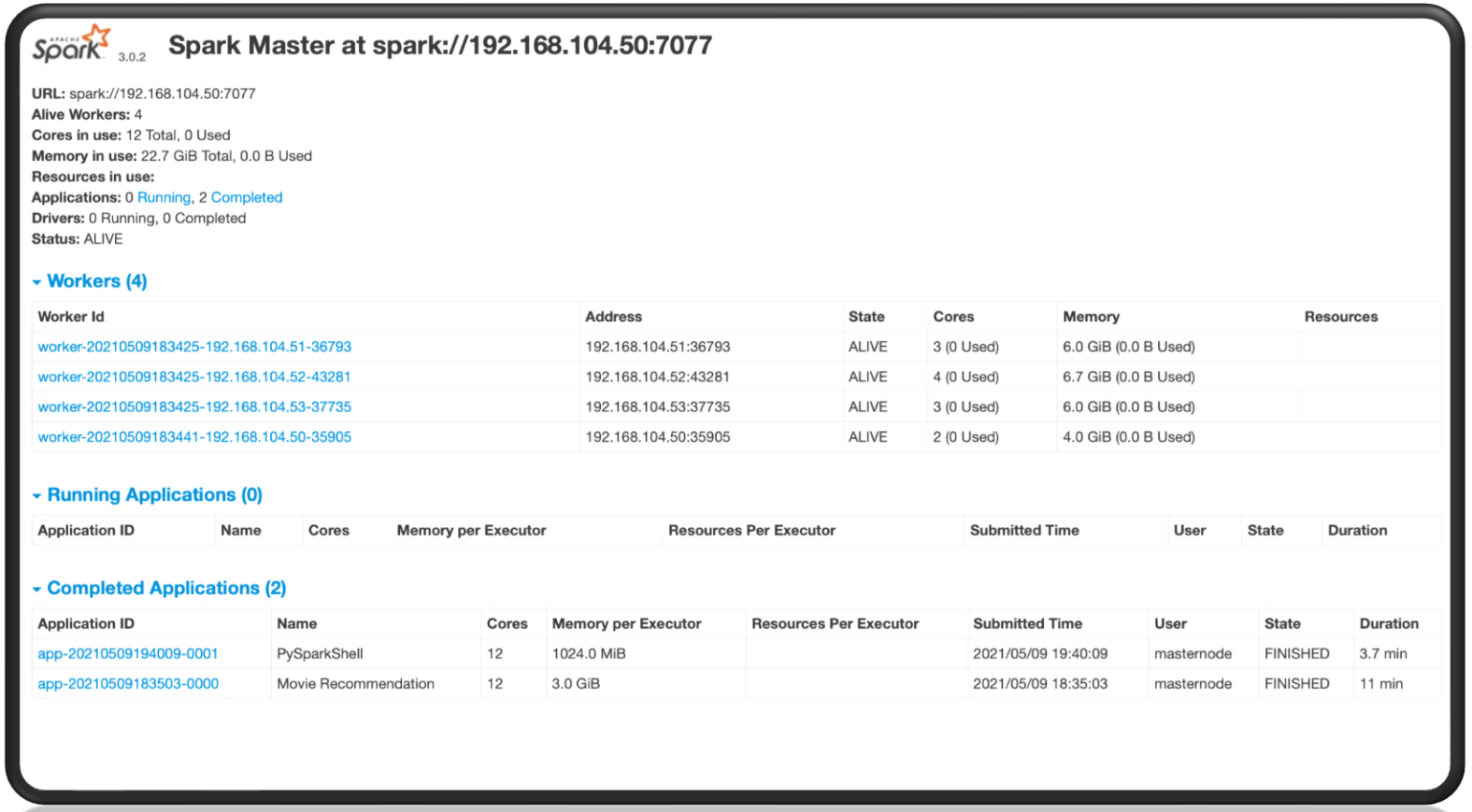
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΣΕ MULTI NODE



- 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**



ΣΥΓΚΡΙΣΗ HADOOP ΜΕ 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-19cfa6d56ff5.
- 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!!!