

ΟΔΗΓΙΕΣ ΓΙΑ ΤΗΝ ΔΗΜΙΟΥΡΓΙΑ ΕΙΚΟΝΙΚΗΣ ΜΗΧΑΝΗΣ LINUX (UBUNTU) ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ SPARK

Σε όσα ακολουθούν παρουσιάζονται οδηγίες για την δημιουργία μιας εικονικής μηχανής Linux και συγκεκριμένα Ubuntu και την εγκατάσταση του λογισμικού SPARK.

Για την δημιουργία και την διαχείριση της εικονικής μηχανής έχει χρησιμοποιηθεί το λογισμικό Oracle VM VirtualBox και συγκεκριμένα η έκδοση 7.2. Η εγκατάσταση έγινε σε ένα σταθμό εργασίας με λειτουργικό σύστημα Windows 10 64bit.

Παρατήρηση: Υπάρχουν διάφοροι τρόποι για να εγκαταστήσετε και να παραμετροποιήσετε ένα περιβάλλον στο οποίο μπορείτε να τρέξετε το λογισμικό Spark. Οι παρακάτω οδηγίες περιγράφουν τα βήματα που ακολουθήθηκαν για την δημιουργία της εικονικής μηχανής που χρησιμοποιήθηκε στο σεμινάριο του μαθήματος. Μπορείτε να τα ακολουθήσετε για να δημιουργήσετε την δική σας εικονική μηχανή.

1. Εγκατάσταση του λογισμικού Virtual Box

Από τον παρακάτω σύνδεσμο κατεβάζουμε και εγκαθιστούμε το λογισμικό VirtualBox. Η τελευταία έκδοση είναι η **7.2**

<https://www.virtualbox.org/>

2. Εγκατάσταση Ubuntu 24.04.3 LTS

Από τον παρακάτω σύνδεσμο κατεβάζουμε το λειτουργικό Ubuntu (Version 24.04.3 LTS):

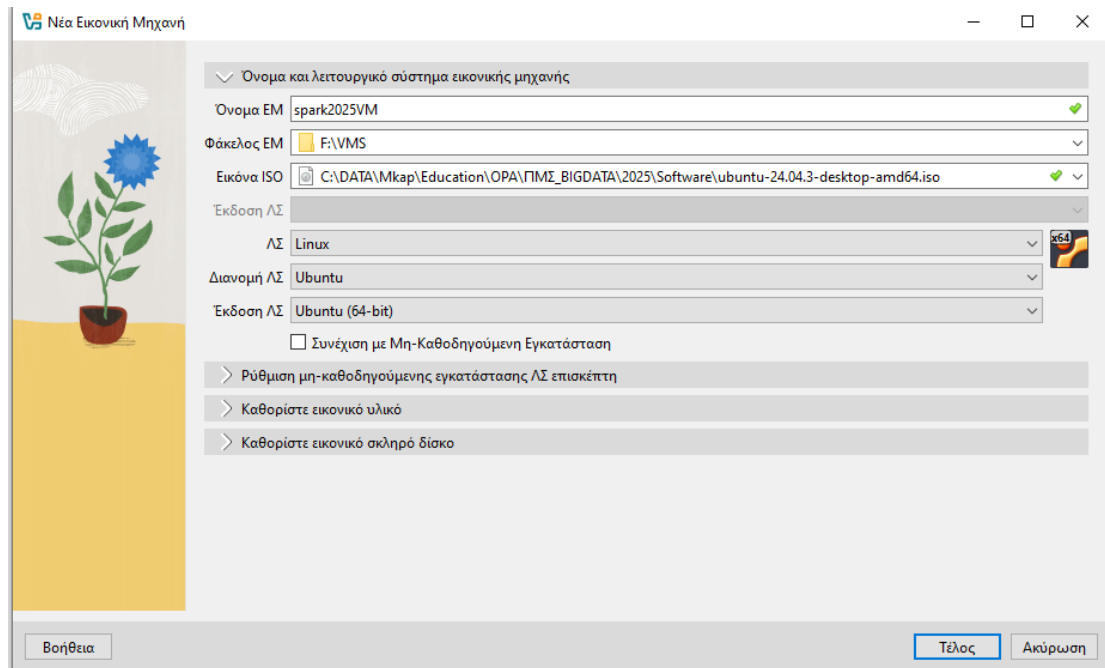
<https://ubuntu.com/download/desktop>

Επιλέξτε το image: **ubuntu-24.04.3-desktop-amd64.iso**

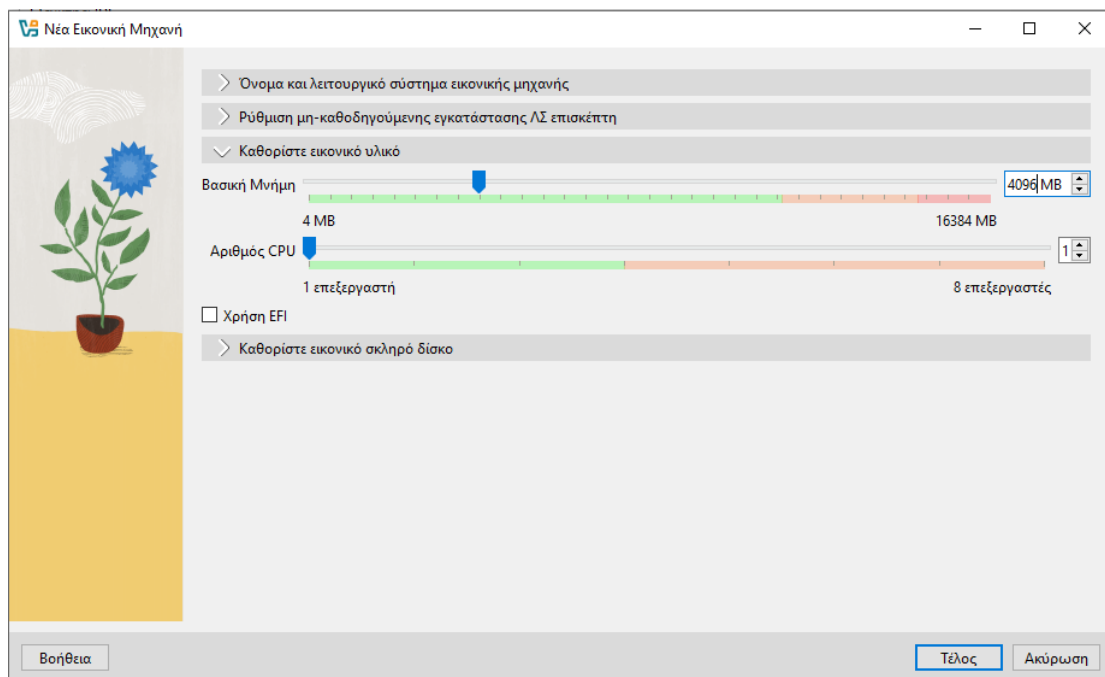
Ξεκινάμε το VirtualBox και από το μενού επιλέγουμε **Μηχανή-->Νέα** στο παράθυρο που εμφανίζεται δίνουμε τα στοιχεία της εικονικής μηχανής που θα δημιουργήσουμε.

Συγκεκριμένα προσδιορίζουμε το Όνομα, τον φάκελο στον οποίο θα δημιουργηθούν τα αρχεία της εικονικής μηχανής και επιλέγουμε το IMAGE του Ubuntu που έχουμε κατεβάσει.

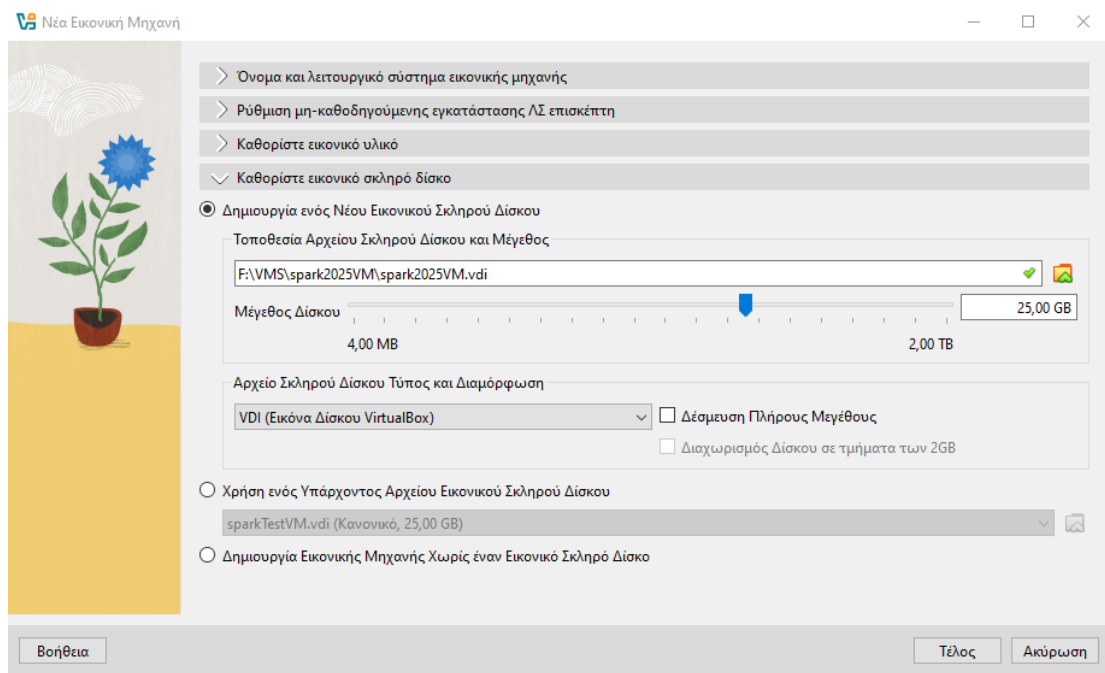
ΠΡΟΣΟΧΗ: Μην επιλέξετε “Συνέχιση με Μη-Καθοδηγούμενη Εγκατάσταση”.



Στη συνέχεια επιλέγουμε «Καθορίστε εικονικό υλικό». Προτεινόμενη Βασική μνήμη **4096 MB**.

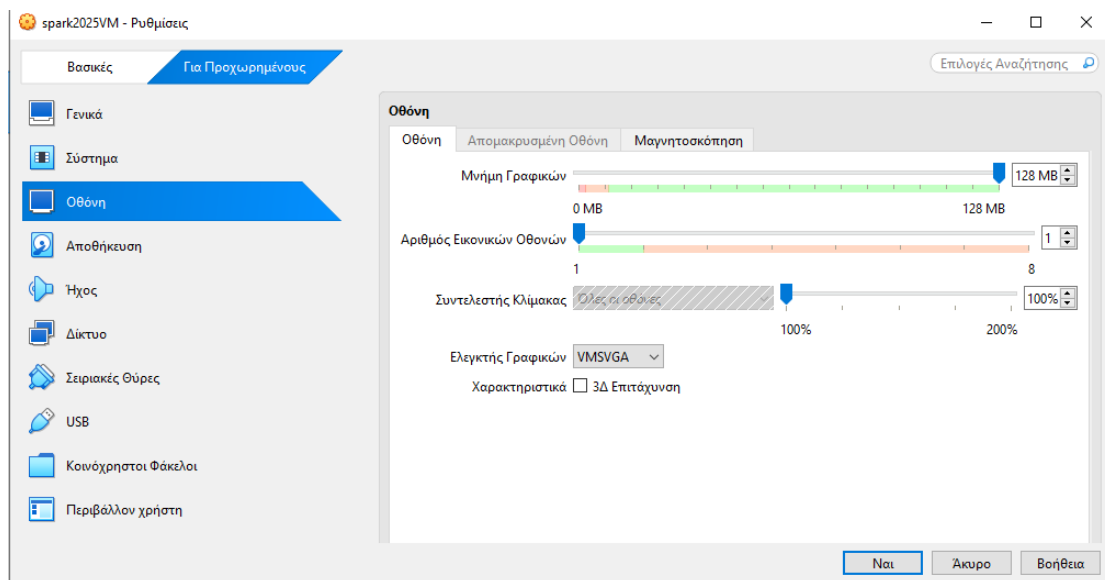


Στην συνέχεια επιλέγουμε «Καθορίστε εικονικό σκληρό δίσκο».

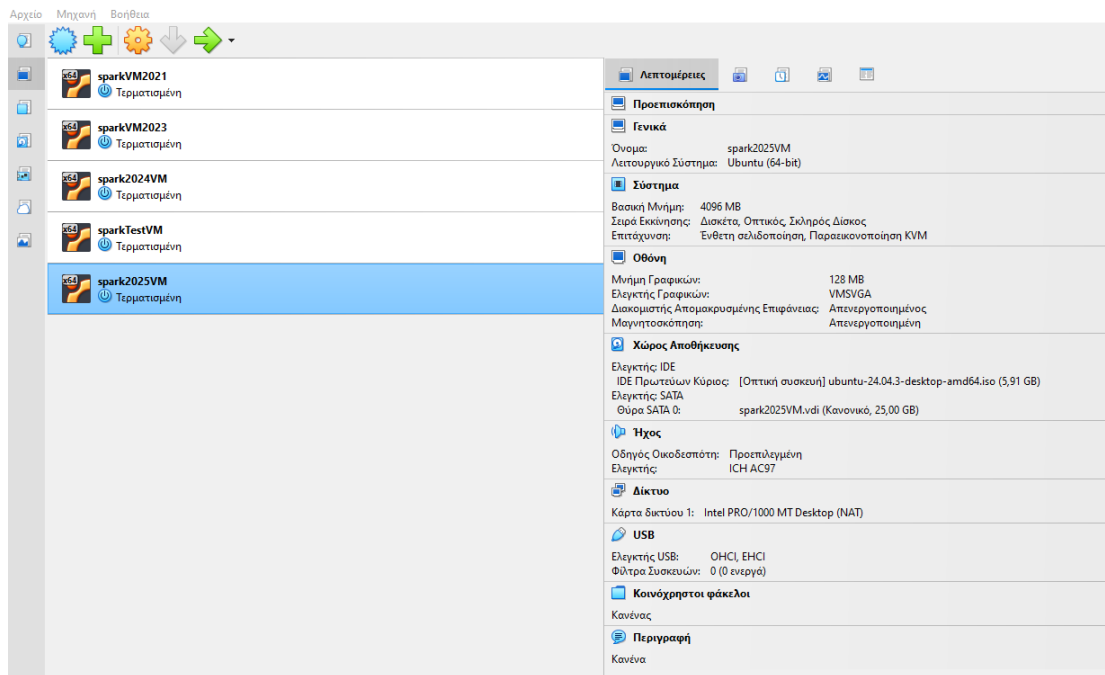


Μετά από τις παραπάνω ρυθμίσεις επιλέγουμε «Τέλος»

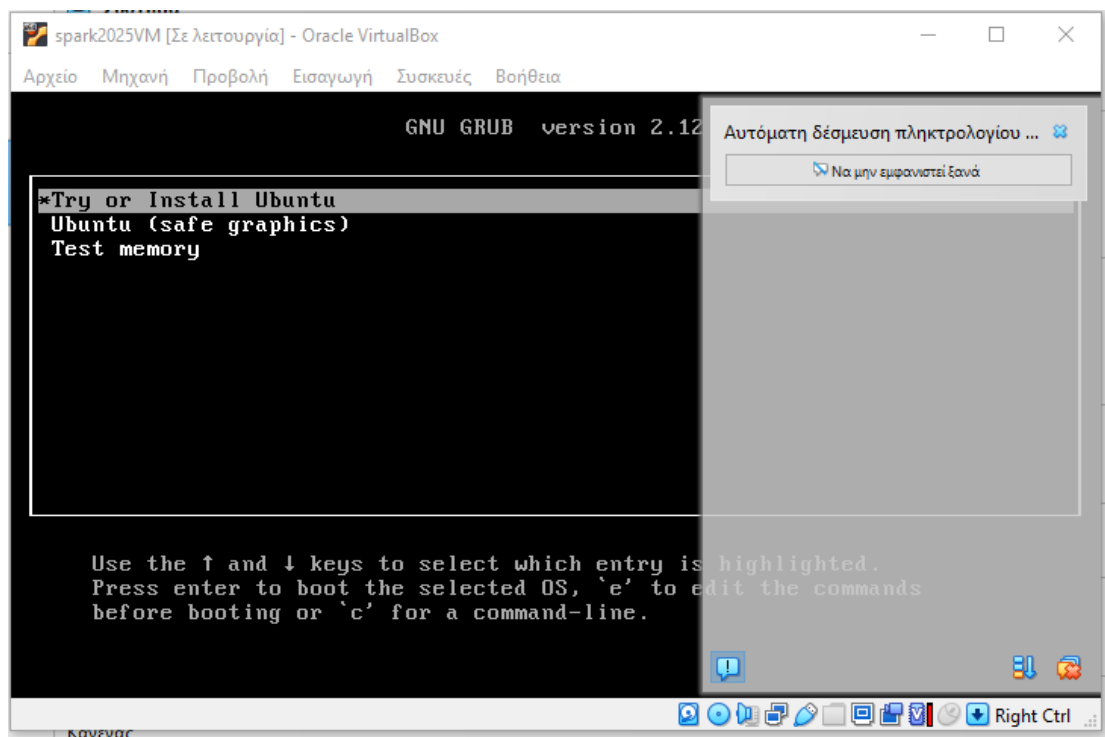
Από τις ρυθμίσεις της εικονικής μηχανής επιλέξτε οθόνη και αυξήστε την μνήμη σε 128MB.

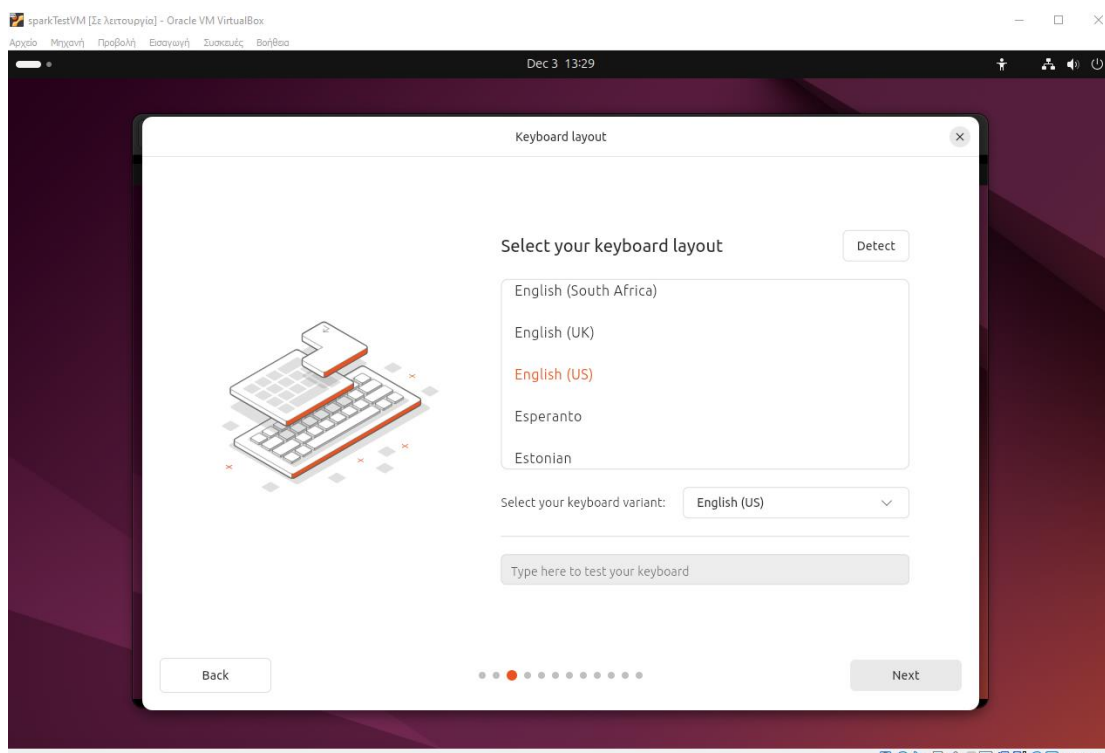
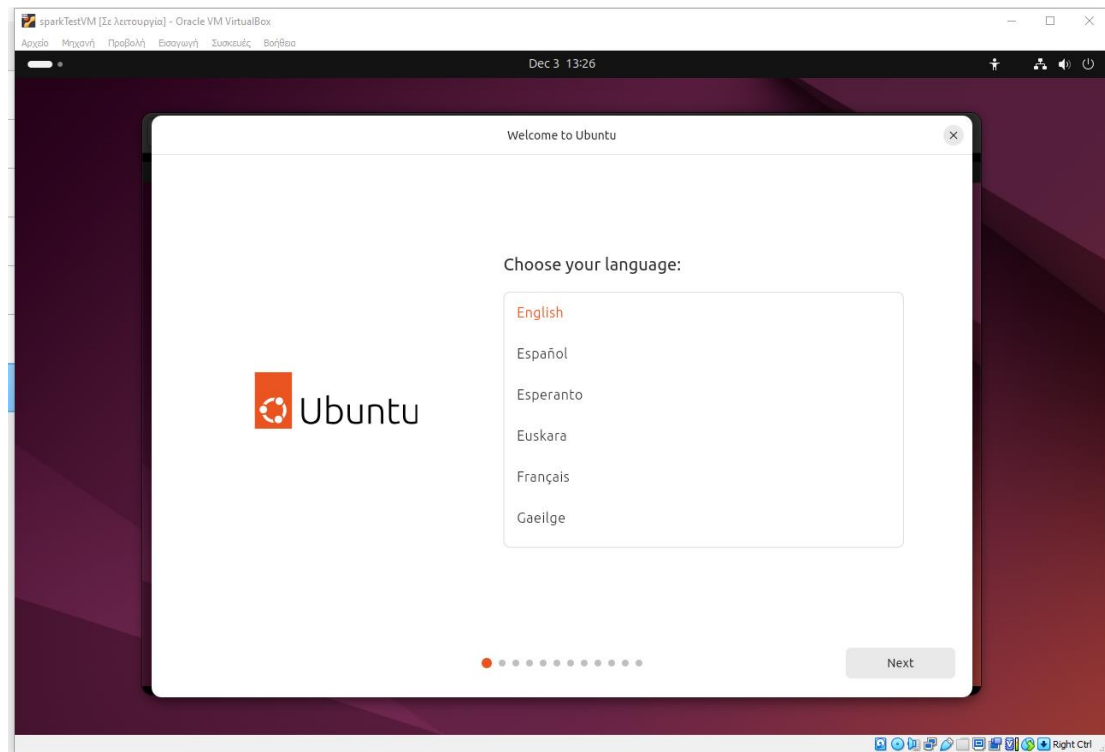


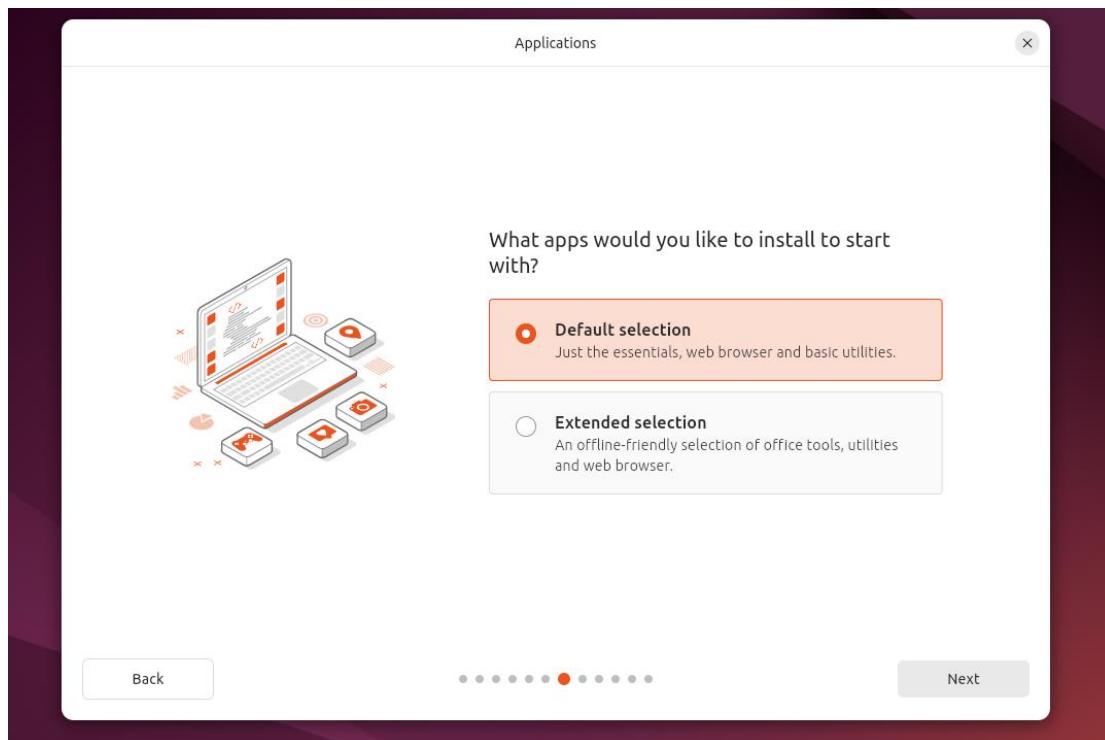
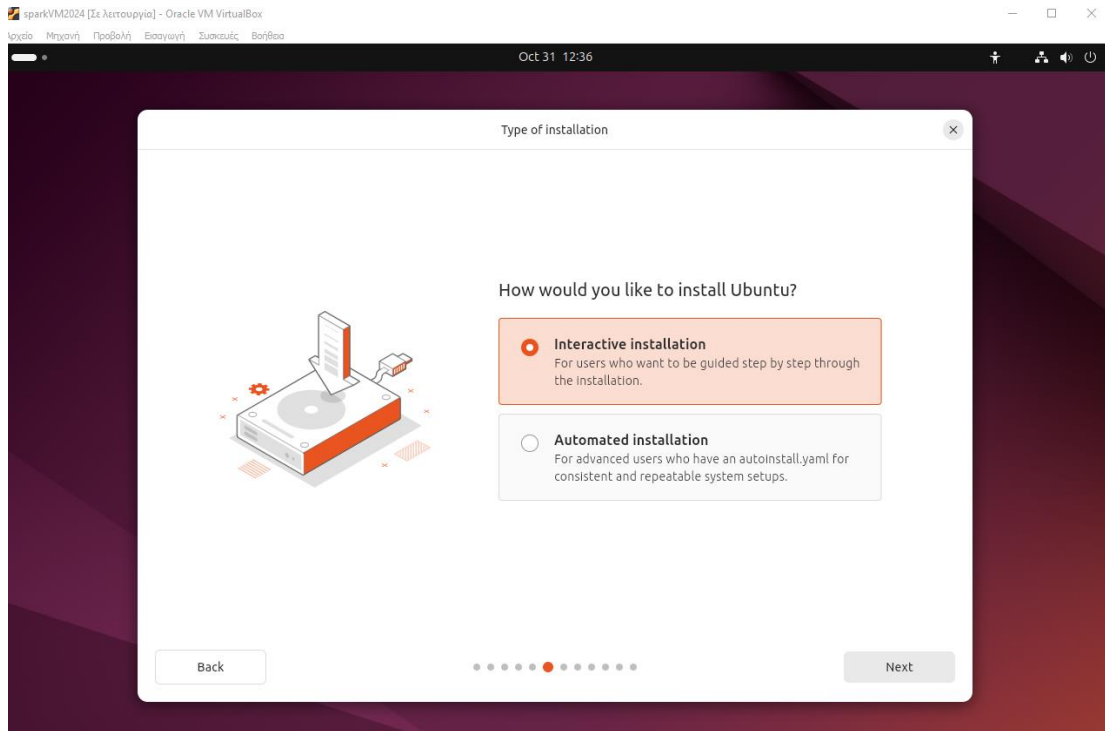
Στο σημείο αυτό επιλέγεται την εικονική μηχανή που δημιουργήσατε και κάνετε click στο πράσινο βελάκι (εκκίνηση) ώστε να ξεκινήσει η εικονική μηχανή.

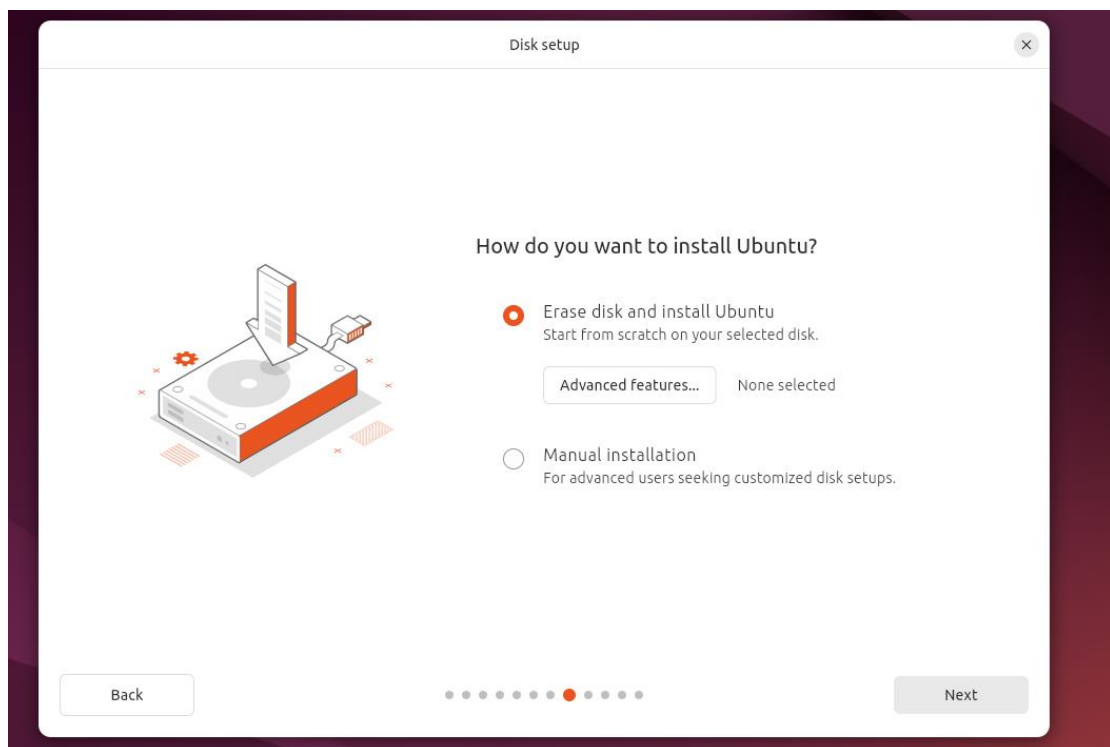
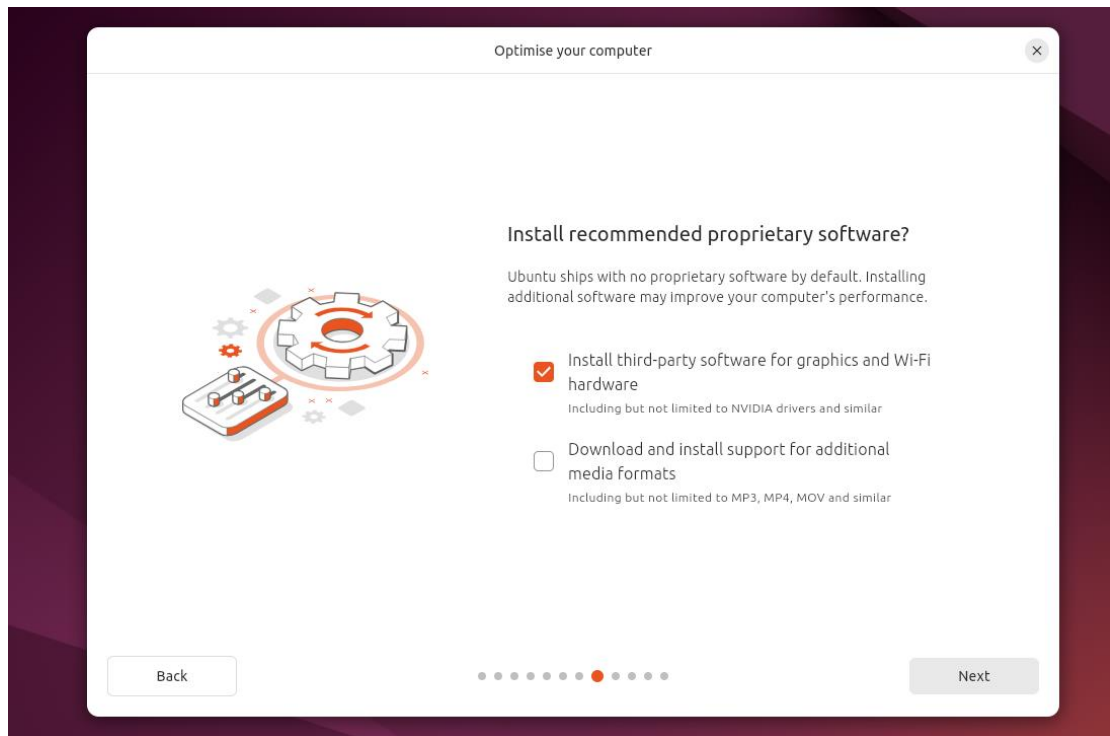


Η εγκατάσταση του λειτουργικού Ubuntu-20.04 θα ξεκινήσει. Μόλις εμφανιστεί η παρακάτω οθόνη επιλέξτε **install Ubuntu**.









Create your account

Create your account

Your name
lab ✓

Your computer's name
pclab ✓

Your username
lab ✓

Password
••••• Show Weak password

Confirm password
••••• ✓

☒ Require my password to log in

☐ Use Active Directory

Back

Next

Τα χαρακτηριστικά της εικονικής μηχανής που χρησιμοποιήθηκε στο εργαστήριο είναι:

your name = lab

computer name = pclab

username = lab

password = lab2025 (είναι σημαντικό να ορίσετε ένα συνθηματικό)

Ακολουθεί η εγκατάσταση του συστήματος και στο τέλος γίνεται επανεκκίνηση.

Η εικονική μηχανή είναι έτοιμη. Στο διάλογο του login θα δείτε το όνομα του χρήστη (lab) θα πληκτρολογήσετε το συνθηματικό (lab2025) και θα αποκτήσετε πρόσβαση στο λειτουργικό Ubuntu. Αν ανοίξετε ένα τερματικό (terminal) και πληκτρολογήσετε την εντολή **pwd** θα εμφανιστεί το **path** του home directory (**/home/lab**) στην συγκεκριμένη περίπτωση.

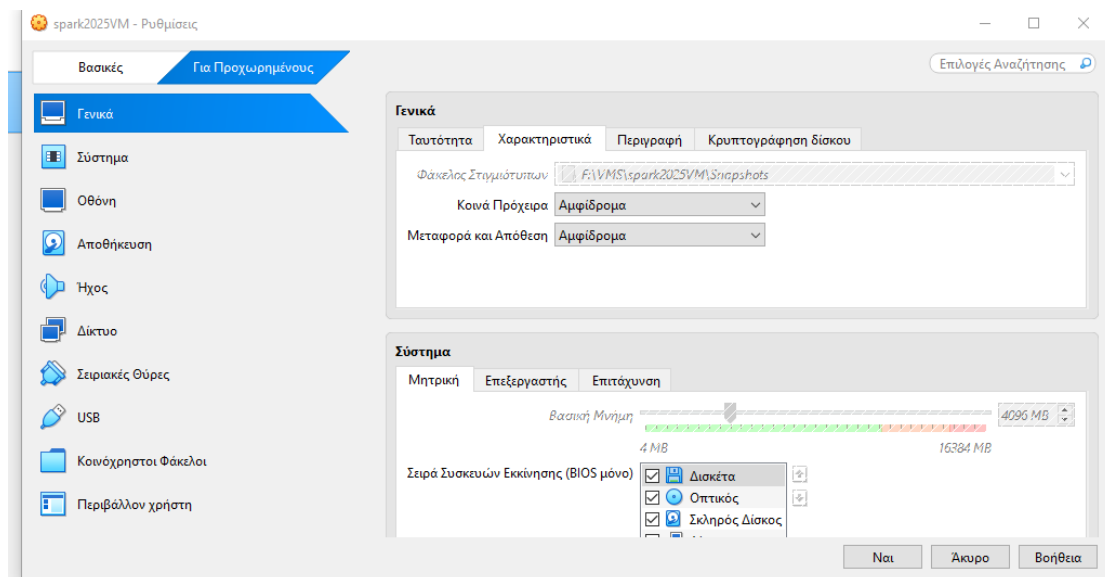
Εισαγωγή Δίσκου με προσθήκες επισκέπτη

Κατόπιν της εγκατάστασης του ubuntu από το μενού επιλογών της εικονικής μηχανής επιλέγουμε **Συσκευές --> Εισαγωγή Δίσκου με τις προσθήκες επισκέπτη**.

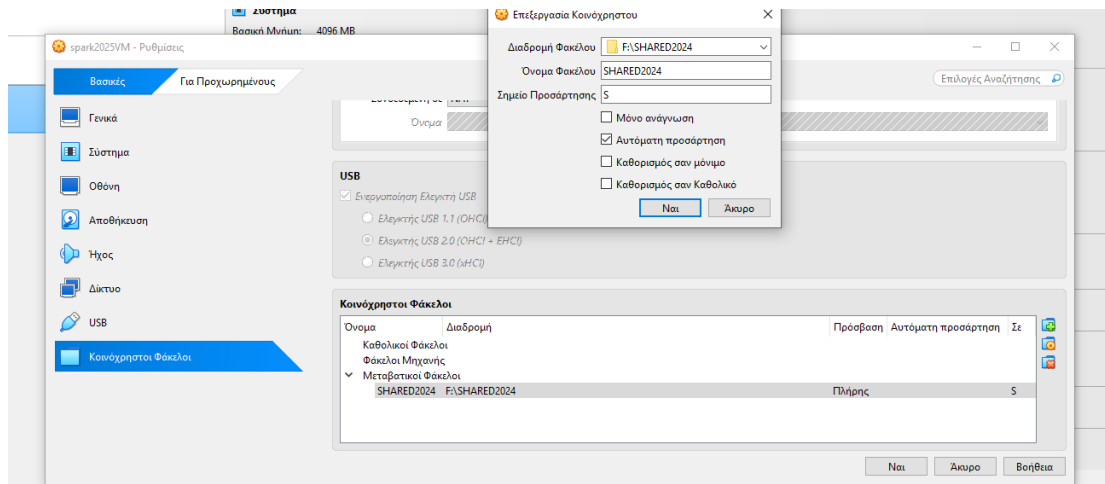
Στην συνέχεια ανοίγουμε ένα τερματικό (terminal) και πληκτρολογούμε τις παρακάτω εντολές:

```
$ sudo apt update  
$ sudo apt install -y build-essential dkms linux-headers-$(uname -r)  
$ sudo mkdir -p /mnt/cdrom  
$ sudo mount /dev/cdrom /mnt/cdrom  
$ sudo /mnt/cdrom/VBoxLinuxAdditions.run  
$ sudo reboot
```

Στην συνέχεια κάνουμε επανεκκίνηση της εικονικής μηχανής. Αυτό για να έχουμε καλύτερη προσαρμογή της επιφάνειας εργασίας καθώς επίσης και δυνατότητες copy-paste μεταξύ της εικονικής μηχανής και του συστήματός μας (π.χ. windows). Απαιτείται επίσης να ενεργοποιήσουμε από τις ρυθμίσεις την επιλογή **Γενικά -- Χαρακτηριστικά --> Κοινά πρόχειρα**



Επιπλέον από τις ρυθμίσεις της εικονικής μηχανής μπορούμε να ορίσουμε κοινόχρηστους φακέλους για να μεταφέρουμε αρχεία από την εικονική μηχανή στον υπολογιστή μας και αντίστροφα.



Αφού επιλέξουμε τον κοινόχρηστο φάκελο (π.χ. SHARED2024) στην συνέχεια δημιουργούμε ένα σημείο προσάρτησης (mount) στο σύστημα αρχείων της εικονικής μηχανής (π.χ. /mnt/S).

```
$ sudo mkdir -p /mnt/S
$ sudo mount -t vboxsf SHARED2024 /mnt/S
```

3. Εγκατάσταση SPARK

Εγκατάσταση προαπαιτούμενων πακέτων

Πριν εγκαταστήσουμε το λογισμικό SPARK πρέπει να εγκαταστήσουμε ορισμένα απαιτούμενα πακέτα και συγκεκριμένα:

- **JDK**
- **Scala**

Ανοίγουμε ένα terminal και από περιβάλλον φλοιού (Shell) και το home directory (/home/lab στην περίπτωση μας) δίνουμε τις εντολές:

Ubuntu Update

```
$ sudo apt update && sudo apt upgrade -y
```

```
$ sudo apt install -y git curl wget unzip build-essential
```

Install Java 17

```
$ sudo apt install -y openjdk-17-jdk
```

```
$ java -version
```

Install Scala 2.13

```
$ curl -fL https://github.com/coursier/coursier/releases/latest/download/cs-x86_64-pc-linux.gz | gzip -d > cs
```

```
$ chmod +x cs && ./cs setup
```

Στο σημείο αυτό κάνουμε **logout** και μετά **login**. Στη συνέχεια πληκτρολογούμε τις παρακάτω εντολές:

```
$ cs install scala:2.13.13
```

```
$ cs install scalac:2.13.13
```

```
$ scala -version
```

Εγκατάσταση του Spark 4.0.1 (Hadoop 3)

Δίνουμε τις παρακάτω εντολές

```
$ cd ~
```

```
$ wget https://downloads.apache.org/spark/spark-4.0.1/spark-4.0.1-bin-hadoop3.tgz
```

Στον φάκελο εργασίας (/home/lab) θα κατέβει το αρχείο **spark-4.0.1-bin-hadoop3.tgz**

Στη συνέχεια κάνουμε extract τα αρχεία με την εντολή:

```
$ tar xvf spark-4.0.1-bin-hadoop3.tgz
```

Δημιουργείται ο φάκελος **spark-4.0.1-bin-hadoop3** τον οποίο θα μετακινήσουμε και θα μετονομάσουμε με εντολή **mv** ως εξής:

```
$ sudo mv spark-4.0.1-bin-hadoop3 /opt/spark
```

Πριν ξεκινήσουμε το Spark πρέπει να διαμορφώσουμε κατάλληλα το περιβάλλον ορίζοντας τις εξής μεταβλητές **στο τέλος** του αρχείου **.profile** το οποίο βρίσκεται στο home directory του χρήστη.

```
$ echo 'export SPARK_HOME=/opt/spark' >> ~/.profile
```

```
$ echo 'export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin' >> ~/.profile
```

```
$ echo 'export PYSARK_PYTHON=/usr/bin/python3' >> ~/.profile
```

Όταν ολοκληρώσουμε τις αλλαγές εκτελούμε το αρχείο **.profile** ως εξής:

```
$ source ~/.profile (η αποσυνδεόμαστε και ξανακάνουμε login)
```

```
$ source ~/.profile
```

Μπορείτε να ελέγξετε την εγκατάσταση του spark με τις παρακάτω εντολές

```
$ spark-shell --version  
$ pyspark --version
```

4. Εκκίνηση του Spark

Εκκίνηση του Master (Standalone Spark Master Server)

```
$ start-master.sh
```

Μπορούμε να δούμε την web διεπαφή του Server με έναν από τους παρακάτω τρόπους:

```
$ http://127.0.0.1:8080/  
$ localhost:8080  
$ pclab:8080 (devicename:8080)
```

Εκκίνηση του Slave (Start a Worker Process)

```
start-worker.sh spark://master:port
```

Στην περίπτωση μας:

```
start-worker.sh spark://pclab:7077
```

Εξ ορισμού η διεργασία worker χρησιμοποιεί όλους τους διαθέσιμους πυρήνες της cpu.

```
start-worker.sh -c 1 spark://pclab:7077
```

(Θέλουμε η διεργασία worker να χρησιμοποιεί έναν πυρήνα της CPU)

```
start-worker.sh -m 512M spark://pclab:7077
```

(Θέλουμε η διεργασία worker να χρησιμοποιεί 512MB RAM)

Δοκιμή του Spark Shell

```
$ spark-shell (SCALA)
```


5. Εγκατάσταση του εργαλείου SBT (Scala Build Tool)

Στο σημείο αυτό θα ανοίξετε ένα τερματικό (terminal) και από το φλοιό (\$) θα δώσετε τις παρακάτω εντολές:

```
$ echo "deb https://repo.scala-sbt.org/scalasbt/debian all main" |  
sudo tee /etc/apt/sources.list.d/sbt.list  
  
$ curl -sL  
"https://keyserver.ubuntu.com/pks/lookup?op=get&search=0x2EE0EA64E40A  
89B84B2DF73499E82A75642AC823" | sudo apt-key add -  
  
$ sudo apt update  
$ sudo apt install -y sbt
```

Την πρώτη φορά που τρέχει το sbt θα εγκαταστήσει ορισμένες βιβλιοθήκες οπότε η όλη διαδικασία μπορεί να διαρκέσει ορισμένα λεπτά. Τις επόμενες φορές ανοίγει γρήγορα.

6. Εγκατάσταση των βιβλιοθηκών matplotlib και pandas

Μπορείτε να εγκαταστήσετε τις βιβλιοθήκες matplotlib και pandas που χρησιμοποιήσαμε στο εργαστήριο για να κάνουμε plot ως εξής:

```
sudo apt install -y python3-matplotlib python3-pandas
```

7. Δημιουργία Spark Project με το εργαλείο SBT

Με το εργαλείο SBT έχουμε την δυνατότητα να κάνουμε να μεταγλωττίσουμε μια εφαρμογή και να δημιουργήσουμε το αρχείο .jar

Έστω μια απλή εφαρμογή η οποία έχει γραφεί σε scala και ο κώδικας βρίσκεται στο αρχείο wordCount.scala.

```
/* wordCount.scala */  
import org.apache.spark._  
import org.apache.spark.SparkContext._  
  
object wordCount {  
  def main(args: Array[String]) {  
    val inputFile = args(0)  
    val outputFile = args(1)  
    val conf = new SparkConf().setAppName("wordCount")  
    // Create a Scala Spark Context.  
    val sc = new SparkContext(conf)  
    // Load our input data.  
    //val input = sc.textFile(inputFile)  
    val input =  
sc.textFile("file:///home/lab/myprj/wordCount/"+inputFile)  
    // Split up into words.  
    val words = input.flatMap(line => line.split(" "))  
    // Transform into word and count.
```

```

    val counts = words.map(word => (word, 1)).reduceByKey ( (x, y)
=> x + y)
    // Save the word count back out to a text file, causing
evaluation.

counts.saveAsTextFile("file:///home/lab/myprj/wordCount/"+outputFile)
}
}

```

Πρέπει να δημιουργήσουμε ένα αρχείο build.sbt (configuration file) στο οποίο να ορίσουμε τις εξαρτήσεις (library dependencies) καθώς η εφαρμογή μας χρησιμοποιεί το API του SPARK. Το αρχείο build.sbt έχει την παρακάτω μορφή:

```

name := "wordCount"
version := "1.0"
scalaVersion := "2.13.16"

libraryDependencies += "org.apache.spark" %% "spark-core" % "4.0.1"

```

Το εργαλείο sbt για να λειτουργήσει σωστά απαιτεί την δημιουργία μιας συγκεκριμένης ιεραρχικής δομής καταλόγων.

Έστω ότι ο κατάλογος εργασίας μας είναι :

```
/home/lab/myprj/wordCount
```

Μέσα στον κατάλογο wordCount πρέπει να υπάρχει το αρχείο build.sbt. Στην συνέχεια δημιουργούμε την εξής ιεραρχική δομή καταλόγων:

```

/home/lab/myprj/wordCount
/home/lab/myprj/wordCount/src
/home/lab/myprj/wordCount/src/main
/home/lab/myprj/wordCount/src/main/scala

```

Το αρχείο με τον κώδικα της εφαρμογής "wordCount.scala" πρέπει να βρίσκεται στον κατάλογο /home/lab/myprj/wordCount/src/main/scala

Πηγαίνουμε στον κατάλογο /home/lab/myprj/wordCount και δίνουμε την εντολή:

```
sbt package
```

Η εντολή αυτή θα δημιουργήσει τους καταλόγους project και target μέσα στον φάκελο wordCount και θα παράγει το αρχείο .jar στον κατάλογο wordCount/target/scala-2.13

Στην συνέχεια από τον φάκελο wordCount μπορούμε να τρέξουμε την εφαρμογή ως εξής:

```
/opt/spark/bin/spark-submit --class "wordCount" --master local[1] target/scala-2.13/wordcount_2.13-1.0.jar
```