

Cloud Computing: laborator 1

Honorius Gâlmeanu
galmeanu@unitbv.ro

October 25, 2023

1 Instalare Fedora Linux

Platforma Hadoop va fi instalată pe un sistem Fedora Linux. Recomandăm Fedora 36, ale cărei imagini pot fi descărcate de la <https://getfedora.org/workstation/download/>. Pentru sistemele de tip Intel veți descărca imagine de tip **x86_64**, iar pentru MacBook M1 de exemplu, arhitectura **aarch64**.

Pentru sistemele de tip Intel se va instala VirtualBox¹. Va trebui să instalați apoi și Virtualbox guest additions, respectiv Virtualbox Extension Pack. Acestea sunt necesare pentru a putea lucra în mod full screen, respectiv a putea accesa dispozitive periferice din mașina virtuală guest².

Dacă aveți la dispoziție un MacBook M1, aveți nevoie să instalați UTM³. Un exemplu de configurare prezentat în figurile 1-3.

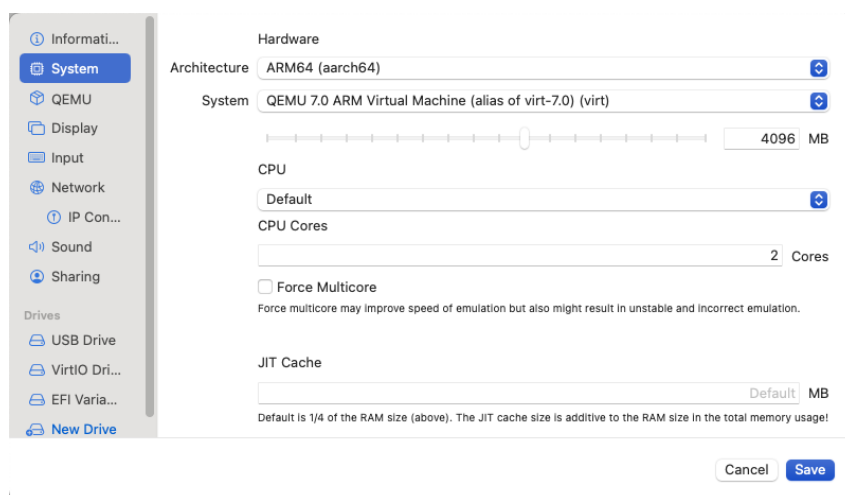


Figure 1: Setări sistem pentru mașina virtuală Fedora 36 sub UTM.

Odată instalat, este recomandat să faceți un update pentru sistem:

Command Line

```
$ sudo dnf update
```



Notice: Opriti mașina virtuală doar prin Shutdown, și nu forțat, din VirtualBox / UTM, prin închiderea ferestrei! Alminteri puteți pierde date din mașina guest, deoarece prin închidere forțată se află într-o stare inconsistentă.

¹https://www.virtualbox.org/wiki/Download_Old_Builds_6_0

²Mașina virtuală instalată în VirtualBox se numește mașină **guest**; ea funcționează pe mașina **host**, adică mașina originală unde este instalat VirtualBox.

³Imaginea .DMG este disponibilă la <https://mac.getutm.app/>

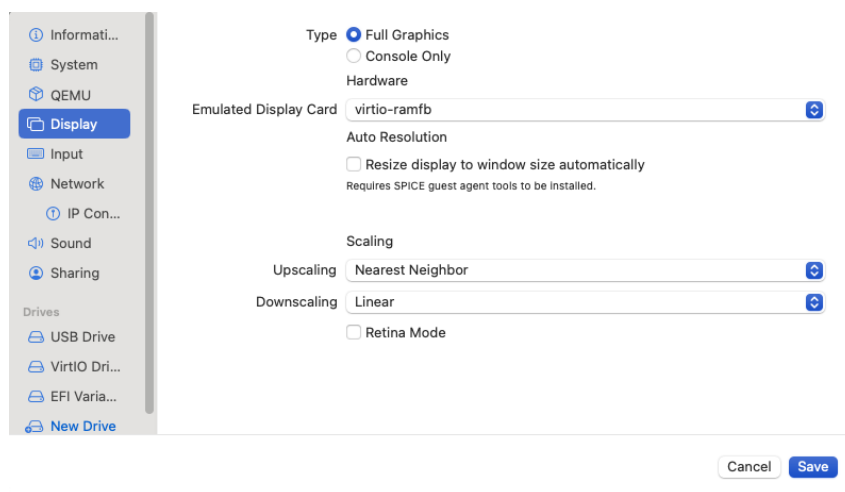


Figure 2: Setări display pentru mașina virtuală Fedora 36 sub UTM.

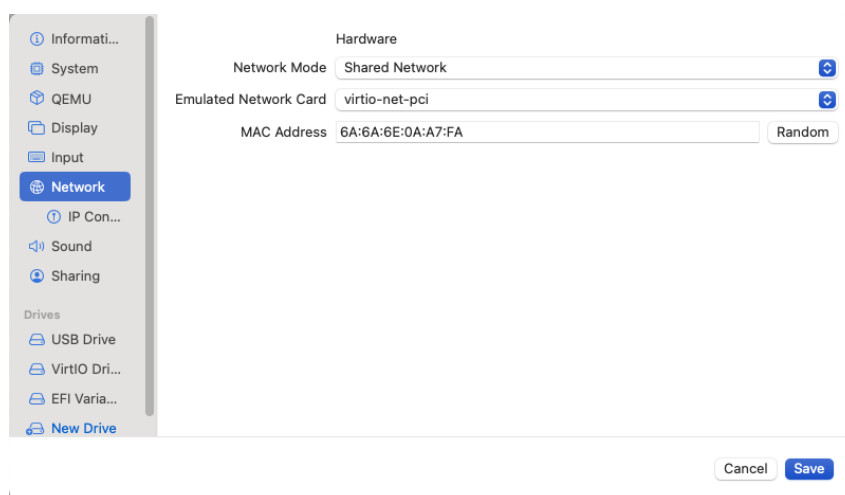


Figure 3: Setări de rețea pentru mașina virtuală Fedora 36 sub UTM.

2 Instalarea framework-ului Hadoop

2.1 Dependințe și dezarhivare

Hadoop are nevoie de Java OpenJDK, astfel că realizăm instalarea Java:

Command Line

```
$ sudo dnf install java-11-openjdk
```

```
[user@fedora ~]$ sudo dnf install java-17-openjdk
[sudo] password for user:
Last metadata expiration check: 1:05:12 ago on Sun 02 Oct 2022 12:45:39 PM EEST.
Dependencies resolved.
=====
Package                        Arch      Version              Repository    Size
=====
Installing:
  java-17-openjdk              aarch64   1:17.0.4.1.1-1.fc36 updates      231 k
Upgrading:
  java-17-openjdk-headless     aarch64   1:17.0.4.1.1-1.fc36 updates      39 M
Installing dependencies:
  ttmkfdirdir                  aarch64   3.0.9-65.fc36       fedora       52 k
  xorg-x11-fonts-Type1         noarch    7.5-33.fc36         fedora      500 k
Transaction Summary
=====
Install  3 Packages
Upgrade  1 Package

Total download size: 40 M
Is this ok [y/N]:
```

Figure 4: Instalarea OpenJDK.

Se va descărca arhiva **hadoop-3.3.4.tar.gz** de la <https://downloads.apache.org/hadoop/common/stable/>, respectiv ultima versiune stabilă, și se va instala în `/home/user/apps/hadoop`:

Command Line

```
$ mkdir /home/user/apps
$ tar -zxvf Downloads/hadoop-3.3.4.tar.gz -C /home/user/apps
$ ln -s /home/user/apps/hadoop-3.3.4 /home/user/apps/hadoop
```

Folosiți un editor de text precum **nano** pentru a edita fișierul **.bashrc**, care este apelat la login și setează variabilele de mediu (sistem); adăugați următoarele linii:

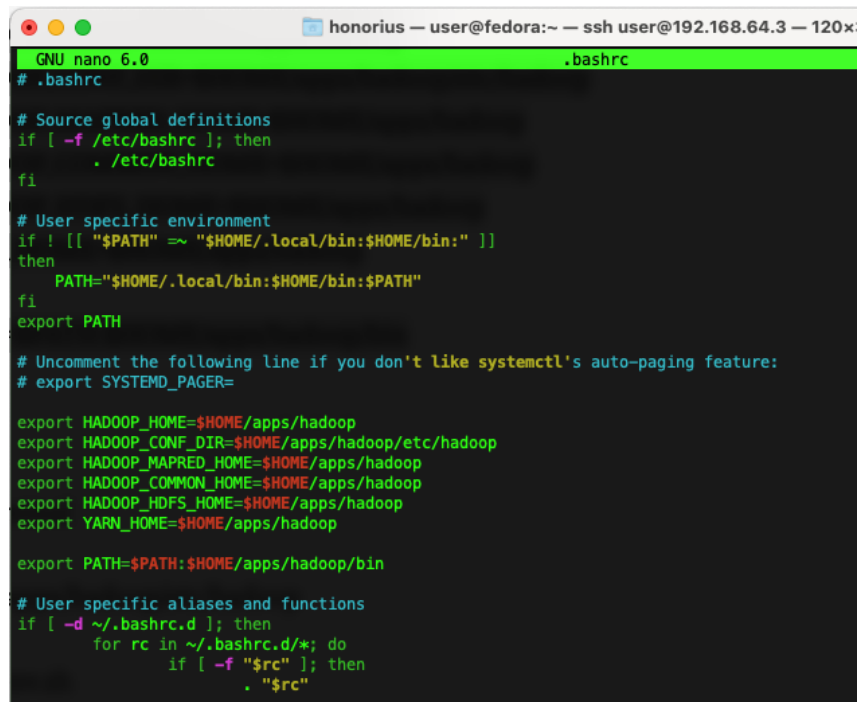
Command Line

```
export HADOOP_HOME=$HOME/apps/hadoop
export HADOOP_CONF_DIR=$HOME/apps/hadoop/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/apps/hadoop
export HADOOP_COMMON_HOME=$HOME/apps/hadoop
export HADOOP_HDFS_HOME=$HOME/apps/hadoop
export YARN_HOME=$HOME/apps/hadoop

export PATH=$PATH:$HOME/apps/hadoop/bin
```

Rezultatul este prezentat în figura 5.

Mai departe, identificăm calea pentru OpenJDK, care în cazul meu este `/usr/lib/jvm/java-11-openjdk-11.0.16.1.1-1.fc36.aarch64`:



```
GNU nano 6.0 .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

export HADOOP_HOME=$HOME/apps/hadoop
export HADOOP_CONF_DIR=$HOME/apps/hadoop/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/apps/hadoop
export HADOOP_COMMON_HOME=$HOME/apps/hadoop
export HADOOP_HDFS_HOME=$HOME/apps/hadoop
export YARN_HOME=$HOME/apps/hadoop

export PATH=$PATH:$HOME/apps/hadoop/bin

# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
    for rc in ~/.bashrc.d/*; do
        if [ -f "$rc" ]; then
            . "$rc"
        fi
    done
fi
```

Figure 5: Variabile sistem.

Command Line

```
$ ls -al /usr/lib/jvm/
drwxr-xr-x. 1 root root 44 Oct 2 13:53 java-11-openjdk-11.0.16.1.1-1.fc36.aarch64
```

Setăm această cale în secțiunea "Generic settings for Hadoop" din fișierul **hadoop-env.sh**:

Command Line

```
$ cd $HOME/apps/hadoop/etc/hadoop
$ nano hadoop-env.sh
...
export JAVA_HOME="/usr/lib/jvm/java-11-openjdk-11.0.16.1.1-1.fc36.aarch64"
```

2.2 Editarea fișierelor de configurare

Vom configura Hadoop ca o instalare de tip single-node. Edităm fișierul de configurare **core-site.xml** și adăugăm următoarea configurare:

Command Line

```
$ cd $HOME/apps/hadoop/etc/hadoop
$ nano core-site.xml
. . .
<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>

<property>
<name>hadoop.tmp.dir</name>
<value>/home/user/apps/hadoop/tmp</value>
</property>

</configuration>
```

Configurăm HDFS (Hadoop Distributed File System):

Command Line

```
$ cd $HOME/apps/hadoop/etc/hadoop
$ nano hdfs-site.xml
. . .
<configuration>

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

<property>
<name>dfs.name.dir</name>
<value>file:///home/user/apps/hadoop/hadoopdata/hdfs/namenode</value>
</property>

<property>
<name>dfs.data.dir</name>
<value>file:///home/user/apps/hadoop/hadoopdata/hdfs/datanode</value>
</property>

</configuration>
```

Configurăm daemon-ul de map-reduce:

Command Line

```
$ cd $HOME/apps/hadoop/etc/hadoop
$ nano mapred-site.xml
. . .
<configuration>

<property>
<name>mapred.job.tracker</name>
<value>localhost:9001</value>
</property>

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

<property>
<name>mapreduce.application.classpath</name>
<value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:
$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
</property>

</configuration>
```

Configurăm daemon-ul de alocare de resurse:

Command Line

```
$ cd $HOME/apps/hadoop/etc/hadoop
$ nano yarn-site.xml
. . .
<configuration>

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

<property>
<name>yarn.nodemanager.env-whitelist</name>
<value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,
CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>

</configuration>
```

Creăm structura de directoare:

Command Line

```
$ cd $HOME/apps/hadoop/

$ mkdir -p hadoopdata/hdfs/namenode
$ mkdir -p hadoopdata/hdfs/datanode

$ mkdir cache
$ mkdir logs
$ mkdir tmp

$ mkdir input
```

Creăm sistemul de fișiere pe nod:

Command Line

```
$ bin/hdfs namenode -format
2022-10-02 15:09:25,261 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = fedora/192.168.64.3
STARTUP_MSG:  args = [-format]
STARTUP_MSG:  version = 3.3.4
. . .
2022-10-02 15:09:25,894 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at fedora/192.168.64.3
*****/
```

2.3 Configurarea daemonului OpenSSH

Hadoop are nevoie, pentru comunicarea între noduri, de protocolul openssh. Vom instala serverul și îl vom porni:

Command Line

```
$ sudo dnf install openssh
$ sudo systemctl enable sshd
$ sudo systemctl start sshd
```

Pentru a se conecta la noduri într-un mod securizat, sistemul Hadoop trebuie să folosească chei. Generăm astfel perechea de chei publică și privată, și setăm daemon-ul să o recunoască:

Command Line

```
$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 600 ~/.ssh/authorized_keys
```

Dacă totul a mers bine, ar trebui ca prin conectarea, din terminal, la localhost, să nu mai fie cerută parola:

Command Line

```
$ ssh user@localhost
Last login: Sun Oct  2 15:33:29 2022 from 192.168.64.1
$
```

3 Pornirea și oprirea serviciilor Hadoop

Pornirea și oprirea va trebui să le faceți de fiecare dată când porniți/opriți mașina virtuală.

Pornirea serviciilor:

Command Line

```
$ pwd
/home/user/apps/hadoop

$ sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as user in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [fedora]
2022-10-02 15:50:31,274 WARN util.NativeCodeLoader: Unable to load native-hadoop
  library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
```

Oprirea serviciilor:

Command Line

```
$ sbin/stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as user in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [fedora]
2022-10-02 16:02:33,091 WARN util.NativeCodeLoader: Unable to load native-hadoop
  library for your platform... using builtin-java classes where applicable
Stopping nodemanagers
Stopping resourcemanager
```

4 Verificarea corectitudinii instalării

Verificăm în continuare că instalarea realizată funcționează.

Exemplul următor pune o serie de fișiere pe sistemul de fișiere distribuit (HDFS), lansează un executabil Hadoop (care este în fapt un program Java) și citește rezultatele.

Sistemul de fișiere inițial este gol. Nu există nici măcar un fișier în el. În HDFS, fiecare utilizator are un "home folder". Acesta este mapat din rădăcină într-o cale de forma /user/«nume utilizator». În cazul nostru, utilizatorul se numește chiar "user", așadar calea va fi /user/user.

Command Line

```
$ hdfs dfs -mkdir /user
$ hdfs dfs -mkdir /user/user
$ hdfs dfs -ls /
Found 2 items
drwx----- - user supergroup          0 2022-10-02 16:14 /tmp
drwxr-xr-x - user supergroup          0 2022-10-02 16:12 /user
```

Creăm folderul "input" și copiem acolo o serie de fișiere XML:

Command Line

```
$ hdfs dfs -mkdir input
$ hdfs dfs -put etc/hadoop/*.xml input
```

Pornim comanda, care face parte din biblioteca de exemple, și se numește "grep". Acesta caută cuvinte în fișierele din folder-ul "input" din HDFS, folosind șablonul specificat de expresia regulată, și le scrie în folder-ul "output":

Command Line

```
$ hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar grep
input output 'dfs[a-z.]+'
. . .
2022-10-02 16:15:03,762 INFO mapreduce.Job: map 0% reduce 0%
2022-10-02 16:15:13,853 INFO mapreduce.Job: map 10% reduce 0%
2022-10-02 16:15:14,857 INFO mapreduce.Job: map 60% reduce 0%
2022-10-02 16:15:20,886 INFO mapreduce.Job: map 100% reduce 0%
2022-10-02 16:15:21,893 INFO mapreduce.Job: map 100% reduce 100%
2022-10-02 16:15:21,916 INFO mapreduce.Job: Job job_1664715923306_0001
completed successfully
2022-10-02 16:15:21,997 INFO mapreduce.Job: Counters: 55
. . .
```

Inspectăm ieșirea programului:

Command Line

```
$ hdfs dfs -cat output/*
1 dfsadmin
1 dfs.replication
1 dfs.name.dir
1 dfs.data.dir
```

Bibliografie

1. J.W. Eckert, "Using Linux on Your M1-based Mac",
<https://www.comptia.org/blog/using-linux-on-your-m1-based-mac>
2. M.C. Bozoglan, "Hadoop install on Linux",
<https://cevheri.medium.com/hadoop-install-on-linux-fedora-or-any-distros-5185e7c1db64>
3. "Hadoop: Setting up a Single Node Cluster",
<https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>