

## Índice

Exercícios de laboratórios Hadoop.....	2
HACK19 – Instalando o SSH Server na Máquina Virtual.....	2
HACK20 – Validando os Pré-requisitos.....	2
HACK21 – Criando um usuário para Hadoop.....	2
HACK22 – Configurando Chave SSH para usuário.....	3
HACK23 – Desabilitando o IPv6.....	3
HACK23 – Adquirindo os pacotes de instalação do Hadoop.....	4
HACK24 – Instalando o pacote Hadoop.....	4
HACK25 – Configurando duas instâncias de JAVA.....	4
HACK26 – Configurando variáveis de ambiente do Hadoop.....	5
HACK27 – Configurando o core-site.xml Hadoop.....	6
HACK28 – Configurando o yarn-site.xml.....	6
HACK29 – Configurando o mapred-site.xml.....	7
HACK30 – Criando diretórios do HDFS.....	7
HACK31 – Configurando os diretórios do HDFS ( hdfs-site.xml ).....	7
HACK32 – Formatando o HDFS Namenode.....	8
HACK33 – Inicializando os Serviços Hadoop.....	9
HACK34 – Encerrando os Serviços Hadoop.....	9
HACK35 – Testando os processos java executando.....	10
HACK36 – Testando as portas TCP dos serviços.....	10

## Exercícios de laboratórios Hadoop

Cria uma pasta em seu ambiente para armazenar estes Hacks ( Exercícios práticos de laboratórios ), os Hacks serão inicialmente usados e criados para a prática dos conhecimentos adquiridos no curso e posteriormente podem ser uma base de conhecimento para consulta.

### ***HACK19 – Instalando o SSH Server na Maquina Virtual.***

1. Acesse sua VM

```
sudo apt-get install openssh-server
```

2. Pronto!

### ***HACK20 – Validando os Pré-requisitos.***

3. Acesse sua VM por SSH

```
ssh ambientelivre@IP
```

4. Verifique se sua VM Linux tem java instalado, o java deve ter a versão acima de 1.6.
5. Deve estar instalada a JVM e a JDK
6. Para testar acesse o console de comandos ( shell ) e digite os comandos abaixo:

```
java -version
```

Este comando ira testar se existe a JVM e mostrar a versão da mesma.

```
javac -version
```

Este comando ira testar se existe um Kit de Desenvolvimento Java ( JDK ) e mostrar a versão da mesma.

7. Caso não tenha sido instada ou tenha problemas na execução dos comandos acima verifique o documento Hacks de Instalação de Ambiente no ambiente de EAD.

### ***HACK21 – Criando um usuário para Hadoop.***

1. Acesse sua VM Linux
2. Execute os comando abaixo no console

```
sudo addgroup hadoop
sudo adduser --ingroup hadoop hadoop
```

3. Adicione e senha “sejalivre” ao usuário e confirme as demais informações com vazio.

## **HACK22 – Configurando Chave SSH para usuário.**

1. acesse como usuário hadoop

```
su - hadoop
```

2. Crie a chave ssh , confirme com ENTER

```
ssh-keygen -t rsa -P ""
```

3. Adicione a chave gerada na lista de autorizados

```
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

4. Sair do console do usuário hadoop

```
exit
```

## **HACK23 – Desabilitando o IPv6.**

1. Edite o arquivo usando um editor de sua preferência.

```
sudo vi /etc/sysctl.conf
```

2. Configure o arquivo sysctl.conf adicionando as linhas conforme abaixo para desabilitar o IPv6

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

3. **Reinicie a sua VM** para receber as novas configurações.

```
sudo shutdown -r now
```

4. Teste com o comando a seguir se esta correto e desabilitar do IPv6 , o resultado deve ser 1

```
cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

5. Se o /resultado é 1 esta correto.

## **HACK23 – Adquirindo os pacotes de instalação do Hadoop.**

1. O site oficial para buscar a versão mais atual do Hadoop é <http://hadoop.apache.org/>
2. Lá poderá encontrar o download da versão 2.2 em <http://ftp.unicamp.br/pub/apache/hadoop/common/>
3. No material do treinamento já deixamos a versão que utilizaremos ( softwares-hadoop-fundamental-ambiente-livre ) no diretório hadoop.
4. Copie o arquivo **hadoop-2.2.0.tar.gz** para sua VM Linux via SSH ou SFTP para o diretório /home/ambientelivre/

## **HACK24 – Instalando o pacote Hadoop.**

1. Acessando a VM Linux e entre no diretório /usr/local

```
cd /usr/local
```

2. Mova o pacote para /usr/local

```
sudo mv /home/ambientelivre/hadoop-2.X.0.tar.gz /usr/local
```

3. descompacte o pacote do hadoop com o comando a seguir:

```
sudo tar xzf hadoop-2.X.0.tar.gz
```

4. crie um link simbólico

```
sudo ln -s hadoop-2.X.0 hadoop
```

5. Altere as permissões do diretório considerando o usuário hadoop e o grupo hadoop

```
sudo chown -R hadoop:hadoop hadoop-2.X.0
```

6. elimine o arquivo tar.gz

```
sudo rm hadoop-2.2.0.tar.gz
```

## **HACK25 – Configurando duas instâncias de JAVA.**

1. Hack opcional! Caso precise rodar duas versões de java em seu server , pode configurar o JAVA\_HOME do Hadoop em /usr/local/hadoop/etc/hadoop/**hadoop-env.sh**
2. Mude a \$JAVA\_HOME a seu gosto
3. Mode de **export JAVA\_HOME=\$JAVA\_HOME**
4. Para **export JAVA\_HOME=/opt/java/jdk1.6\_31**

## ***HACK26 – Configurando variáveis de ambiente do Hadoop.***

1. Acesso a VM com o usuário hadoop

```
su - hadoop
```

2. entre no diretório do usuário

```
cd ~
```

3. edit o arquivo .bashrc do usuário hadoop

```
vi .bashrc
```

4. Adicione no final do arquivo as variáveis abaixo

```
export JAVA_HOME=/opt/java/jdk1.6.0_31
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
```

5. Encere a sessão do usuário hadoop e e faça login novamente para testar o hadoop

```
exit
```

```
su -hadoop
```

6. execute o comando para mostrar a versão do hadoop

```
hadoop version
```

7. será apresentado algo similar a mensagem abaixo

```
Hadoop 2.2.0
Subversion https://svn.apache.org/repos/asf/hadoop/common -r 1529768
Compiled by hortonmu on 2013-10-07T06:28Z
Compiled with protoc 2.5.0
From source with checksum 79e53ce7994d1628b240f09af91e1af4
This command was run using /usr/local/hadoop-2.2.0/share/hadoop/common/hadoop-
common-2.2.0.jar
```

8. Pronto variáveis de ambiente configuradas

### **HACK27 – Configurando o core-site.xml Hadoop.**

1. Este arquivo contém no nível do sistema itens de configuração do Hadoop, como a URL HDFS, o diretório temporário Hadoop, e localizações do script para rack conscientes clusters Hadoop. Configurações neste arquivo substituir as configurações no núcleo de default.xml.
2. Acesso o diretório de configurações do hadoop

```
cd /usr/local/hadoop/etc/hadoop
```

3. Edite o arquivo core-site.xml

```
vi core-site.xml
```

4. adicione entre a tag <configuration> as configurações abaixo

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

5. Salve o arquivo.

### **HACK28 – Configurando o yarn-site.xml.**

1. Edite o arquivo yarn-site.xml
2. inclua as linhas a seguir:

```
<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>
```



3. salve o arquivo

### **HACK29 – Configurando o mapred-site.xml.**

1. Copie o arquivo template ( mapred-site.xml.template ) para o nosso arquivo mapred-site.xml

```
cp mapred-site.xml.template mapred-site.xml
```

2. Edite o arquivo mapred-site.xml

3. inclua as linhas a seguir entre a tag configuraton:

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

4. salve o arquivo

### **HACK30 – Criando diretórios do HDFS.**

1. Com o usuário hadoop acesse o home

```
cd ~
```

2. Crie o diretório para o namenode e o datanode

```
mkdir -p mydata/hdfs/namenode  
mkdir -p mydata/hdfs/datanode
```

### **HACK31 – Configurando os diretórios do HDFS ( hdfs-site.xml ).**

1. Acesse o diretório de configuração do Hadoop

```
cd /usr/local/hadoop/etc/hadoop
```

2. Edite o arquivo hdfs-site.xml

```
vi hdfs-site.xml
```

3. Adiciona o XML abaixo entre as tags configuration

```
<property>
```

```
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:/home/hadoop/mydata/hdfs/namenode</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:/home/hadoop/mydata/hdfs/datanode</value>
</property>
```

4. salve o arquivo

### **HACK32 – Formatando o HDFS Namenode.**

1. Acesse a VM com o usuário hadoop
2. execute o comando de formatação

```
hdfs namenode -format
```

3. Uma mensagem similar a esta sera emitida, aqui ela foi “comprimida”:

```
14/01/17 00:33:08 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = hadoop/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.2.0
STARTUP_MSG:   classpath = /usr/local/hadoop-2.2.0/etc/hadoop:
.....
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common -r
1529768; compiled by 'hortonmu' on 2013-10-07T06:28Z
...
/*****
SHUTDOWN_MSG: Shutting down NameNode at hadoop/127.0.1.1
*****/
```



4. o HDFS esta formatado e pronto para armazenamento.

### **HACK33 – Inicializando os Serviços Hadoop.**

1. Com o usuário hadoop em qualquer diretório execute o comando a seguir para iniciar o hdfs

```
start-dfs.sh
```

2. será emitida uma mensagem de alerta

```
14/01/17 00:51:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

```
Starting namenodes on [localhost]
```

```
localhost: starting namenode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-namenode-hadoop.out
```

```
localhost: starting datanode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-datanode-hadoop.out
```

```
Starting secondary namenodes [0.0.0.0]
```

```
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.2.0/logs/hadoop-hadoop-secondarynamenode-hadoop.out
```

```
14/01/17 00:52:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

3. Agora execute o comando para iniciar o yarn

```
start-yarn.sh
```

4. será emitida uma mensagem similar a está:

```
starting yarn daemons
```

```
starting resourcemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-resourcemanager-hadoop.out
```

```
localhost: starting nodemanager, logging to /usr/local/hadoop-2.2.0/logs/yarn-hadoop-nodemanager-hadoop.out
```

5. Serviços funcionando para nossas atividades.

### **HACK34 – Encerrando os Serviços Hadoop.**

1. Caso queira parar os serviços use os comandos abaixo ( **não executar apenas para conhecimento** )

```
stop-dfs.sh
```

```
stop-yarn.sh
```

### **HACK35 – Testando os processos java executando.**

1. Para consultar os processos java executando digite o comando a seguir

```
jps
```

2. Deve aparecer os seguintes nomes de processos:

```
6231 DataNode
7178 Jps
6657 ResourceManager
6497 SecondaryNameNode
6868 NodeManager
6024 NameNode
```

3. Se estes processos aparecer seus serviços hadoop estão todos em execução.

### **HACK36 – Testando as portas TCP dos serviços.**

1. Vamos consultar os serviços e as portas abertas.
2. Para o serviço **Hadoop IPC port**

Sintaxe:

`http://<IP-DA-SUA_VIRTUALMACHINE>:8031/`

Exemplo : `http://192.168.0.102:8031/`

3. NodeManager information

`http://192.168.0.102:8042/node`

4. Cluster Metrics

`http://192.168.0.102:8088/cluster`