

PROCESSO DE PREPARAÇÃO DO AMBIENTE HADOOP

Disciplina “Processamento Massivo Paralelo Hadoop e Mapreduce”.

Versão: 2.3

Introdução

O objetivo desse documento é orientar o aluno na configuração de um cluster de Hadoop contendo 4 nodes, utilizando a VM de Treinamento da Cloudera e o VMware Player.

Total serão 5 atividades intercaladas entre a aula.

Parte 1 - Primeiro download da VMware Player 15

Realize o download da VM Player no link:

<https://www.vmware.com/br/products/workstation-player.html>

Versão para Download:

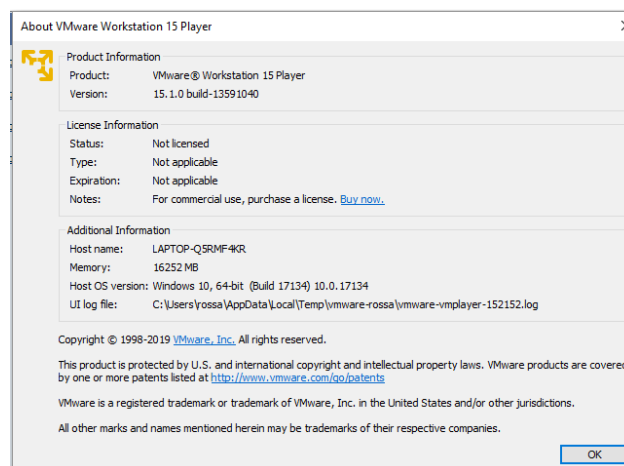
Faça download agora

VMware Workstation 15.1.0 Player for
Windows 64-bit Operating Systems

(exe | 134.64 MB)

[+ Show Details](#)

Download ↓



Parte 2 – Download e Configuração do Cluster Hadoop

ATIVIDADE - 1

Nesta sessão iremos realizar o download da VM de treinamento da Cloudera e iniciar a configuração dos nodes.

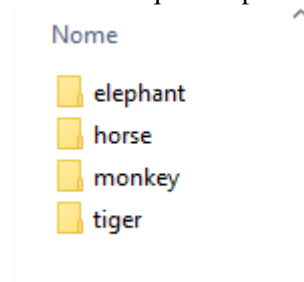
Cada node terá uma função no cluster. Ao final de todo esse documento, iremos rodar um job MapReduce no cluster e averiguar o seu resultado.

Siga o passo-a-passo a seguir, prestando bastante atenção em cada atividade e em qual **hostname** estas devem ser aplicadas:

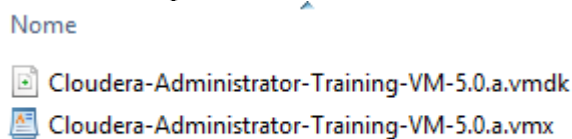
1. Realize o download da VM de treinamento de Administrator da Cloudera no link:

<http://training.cloudera.com/cloudera/VMs/Cloudera-Administrator-Training-VM-5.0.a-vmware-5.0.a.zip>

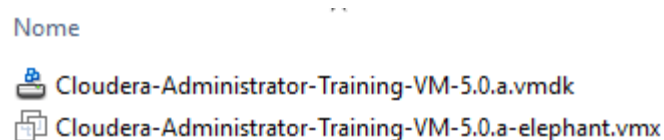
2. Assim que o download concluir, extraia a VM do arquivo .zip.
3. Crie 4 diretórios, por exemplo no C:\, cada um com o nome dos hostnames das VMs: **elephant, horse, monkey e tiger**. Você terá na sua tela uma lista de diretórios igual a figura abaixo. Onde o primeiro diretório é seu arquivo zip descompactado:



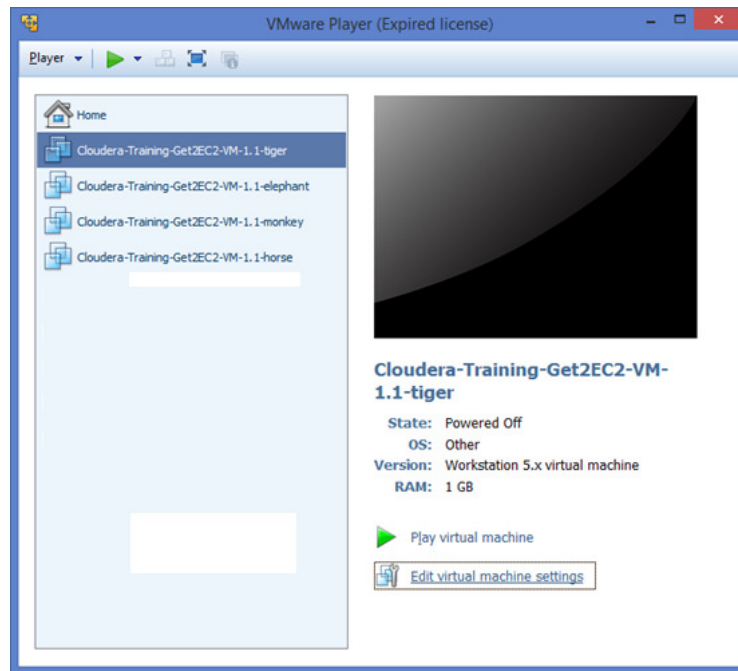
4. Copie os arquivos que estão dentro da pasta Cloudera-Administrator-Training-VM-5.0.a-vmware-5.0.a para cada um dos respectivos diretórios dos hostnames:



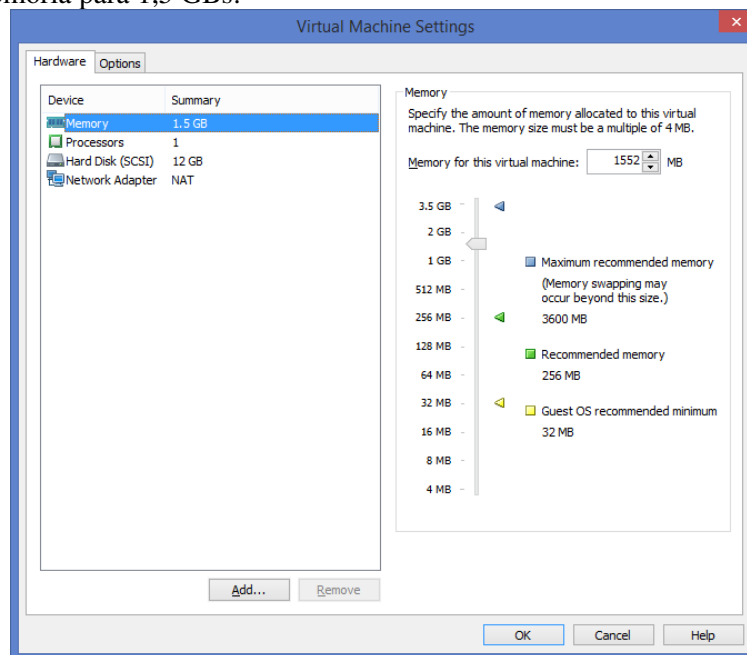
5. Na pasta **elephant**, altere o nome do arquivo com **extensão .vmx**, colocando **-hostname** no final do nome, conforme exemplo abaixo:



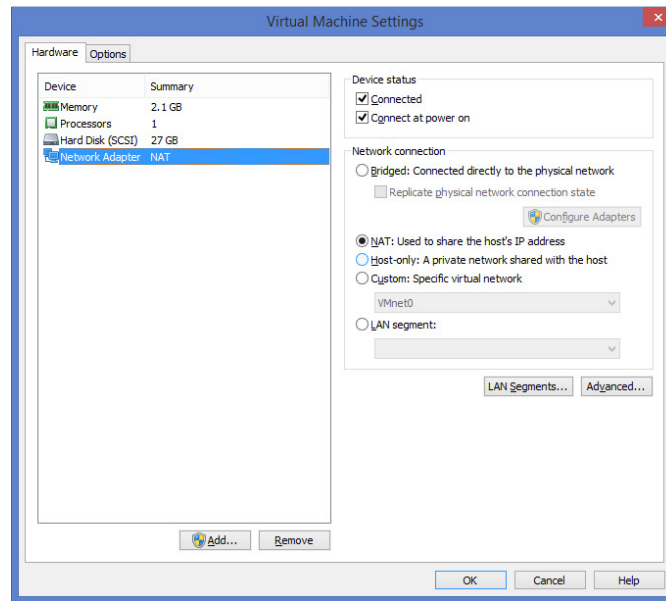
6. Abra o arquivo **Cloudera-Administrator-Training-VM-5.0.a-elephant.vmx** e altere a linha displayName, colocando no final o nome hostname:
displayName = "Cloudera-Administrator-Training-VM-5.0.a-**elephant**"
7. Salve e feche o arquivo.
8. Abra o VM player e cliquei em:
File -> Open
Selecione o arquivo com extensão .vmx que você acabou de alterar.
Clique em "Abrir"
9. Clique em "edit virtual machine settings":



10. Aumente a memória para 1,5 GBs:



11. Nas configurações de rede da máquina virtual garanta que a opção de rede selecionada seja NAT:



12. Clique em “Ok”, depois em “Play virtual machine” para inicializar a VM.
13. Repita esse mesmo procedimento para outras 3 VMs (**tiger, horse e monkey**).

Conferindo as VMs antes de continuar:

- Todas as VMs devem ter alocado 1.5GB de Memória
- Todas as VMs devem ter o “Network Adapter” ativo com “NAT”

ATIVIDADE - 2

Nesta sessão iremos subir as 4 máquinas virtuais e realizar as configurações necessários em cada *Node*.

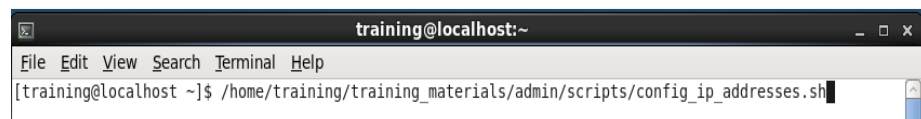
Siga o passo-a-passo a seguir, prestando bastante atenção em cada atividade e em qual *hostname* estas configurações devem ser aplicadas:

1. Suba as 4 VMs.

Depois que as 4 VMs estiverem ativas, rode o script abaixo no **tiger**, no **horse** e no **monkey**.
NÃO RODE NO elephant.

/home/training/training_materials/admin/scripts/config_ip_addresses.sh

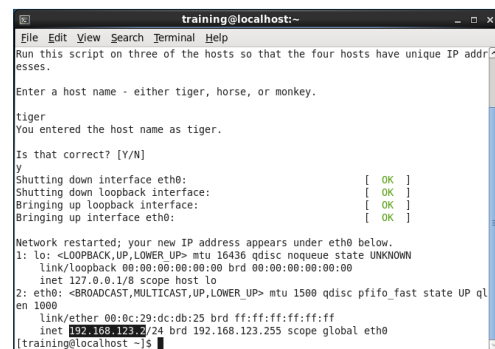
Exemplo:



```
training@localhost:~  
File Edit View Search Terminal Help  
[training@localhost ~]$ /home/training/training_materials/admin/scripts/config_ip_addresses.sh
```

2. Quando for perguntado por um hostname, digite: tiger, horse e monkey respectivamente.
Atenção: Verificar se o hostname está de acordo com VM.
3. Digite Y e tecla ENTER para confirmar.
4. Confira os IPs que aparecem no final do processo:
 - No **Tiger**, deve aparecer o IP 192.168.123.2
 - No **Horse**, 192.168.123.3
 - No **Monkey**, 192.168.123.4

Exemplo:



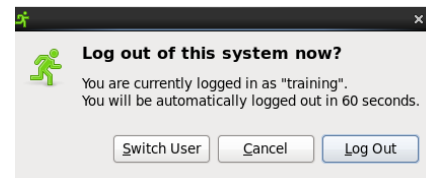
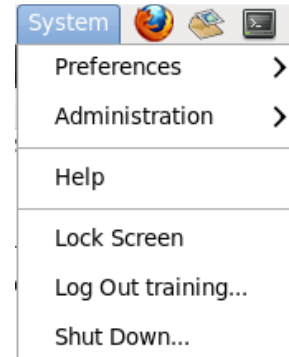
```
training@localhost:~  
File Edit View Search Terminal Help  
Run this script on three of the hosts so that the four hosts have unique IP addresses.  
Enter a host name - either tiger, horse, or monkey.  
tiger  
You entered the host name as tiger.  
Is that correct? [Y/N]  
Y  
Shutting down interface eth0: [ OK ]  
Shutting down loopback interface: [ OK ]  
Bringing up loopback interface: [ OK ]  
Bringing up interface eth0: [ OK ]  
Network restarted; your new IP address appears under eth0 below.  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000  
    link/ether 00:0c:29:dc:db:25 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.123.2/24 brd 192.168.123.255 scope global eth0  
[training@localhost ~]$
```

5. Execute no **elephant** o script:
/home/training/training_materials/admin/scripts/config_hosts.sh
6. Digite “yes” para cada confirmação de conexão com um novo host (tiger, monkey e horse)
7. Verifique que você pode dar um ping, do elephant nos 4 hostnames.

Exemplo abaixo, ping no monkey:

```
[training@localhost ~]$ ping monkey  
PING monkey (192.168.123.4) 56(84) bytes of data.  
64 bytes from monkey (192.168.123.4): icmp_seq=1 ttl=64 time=0.418 ms  
64 bytes from monkey (192.168.123.4): icmp_seq=2 ttl=64 time=0.420 ms  
64 bytes from monkey (192.168.123.4): icmp_seq=3 ttl=64 time=0.290 ms  
64 bytes from monkey (192.168.123.4): icmp_seq=4 ttl=64 time=0.649 ms
```

8. Execute “**uname-n**” nos 4 animais para ver se está tudo certo. Deve aparecer o nome do hostname de cada animal ao executar esse comando.
9. Importante, no topo **Menu**, clique em **System ->Log Out training** e clique **Log Out**”. Ao sair, você automaticamente deve voltar com o mesmo usuário.



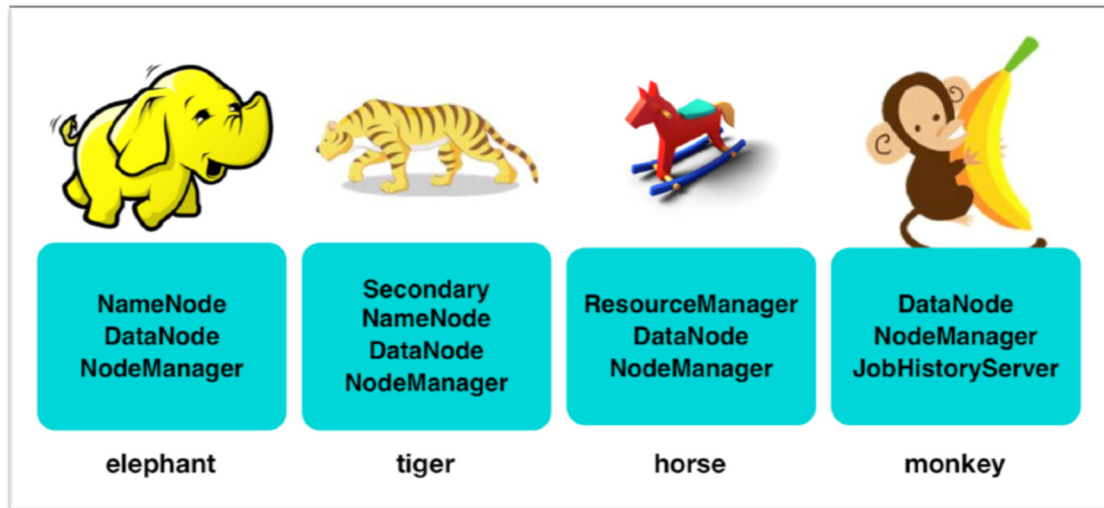
10. Isso irá configurar a variável de ambiente \$HOSTNAME corretamente. Favor executar esse procedimento para **todas** as 4 VMs.
11. Verifique o HOSTNAME nas 4 VMs, digitando:
`echo $HOSTNAME`

Importante: em cada VM devesa aparecer o seu respectivo animal. Exemplo: horse, monkey, Tiger ou elephant.

ATIVIDADE - 3

Nesta sessão iremos instalar o Cluster com os serviços do HDFS:

Siga o passo-a-passo a seguir, prestando bastante atenção em cada atividade e em qual *hostname* estas configurações devem ser aplicadas:



1) No **elephant**, execute:

```
sudo yum install --assumeyes hadoop-hdfs-namenode
sudo yum install --assumeyes hadoop-hdfs-datanode
sudo yum install --assumeyes hadoop-yarn-nodemanager
sudo yum install --assumeyes hadoop-mapreduce
```

2) No **tiger**, execute:

```
sudo yum install --assumeyes hadoop-hdfs-secondarynamenode
sudo yum install --assumeyes hadoop-hdfs-datanode
sudo yum install --assumeyes hadoop-yarn-nodemanager
sudo yum install --assumeyes hadoop-mapreduce
```

3) No **horse**, execute:

```
sudo yum install --assumeyes hadoop-yarn-resourcemanager
sudo yum install --assumeyes hadoop-hdfs-datanode
sudo yum install --assumeyes hadoop-yarn-nodemanager
sudo yum install --assumeyes hadoop-mapreduce
```

4) No **monkey**, execute:

```
sudo yum install --assumeyes hadoop-yarn-resourcemanager
sudo yum install --assumeyes hadoop-hdfs-datanode
sudo yum install --assumeyes hadoop-mapreduce
sudo yum install --assumeyes hadoop-yarn-nodemanager
sudo yum install --assumeyes hadoop-mapreduce-historyserver
```

5) Ir para o diretório no **elephant**:

```
cd ~/training_materials/admin/stubs
```

6) No **elephant**, copie os arquivos de parâmetros do hdfs, mapreduce e yarn para o diretório de configuração:

```
sudo cp core-site.xml /etc/hadoop/conf/
```

```
sudo cp hdfs-site.xml /etc/hadoop/conf/
sudo cp yarn-site.xml /etc/hadoop/conf/
sudo cp mapred-site.xml /etc/hadoop/conf/
```

Exemplo:

```
[training@elephant ~]$ cd ~/training_materials/admin/stubs
[training@elephant stubs]$ sudo cp core-site.xml /etc/hadoop/conf/
[training@elephant stubs]$ sudo cp hdfs-site.xml /etc/hadoop/conf/
[training@elephant stubs]$ sudo cp yarn-site.xml /etc/hadoop/conf/
[training@elephant stubs]$ sudo cp mapred-site.xml /etc/hadoop/conf/
[training@elephant stubs]$ █
```

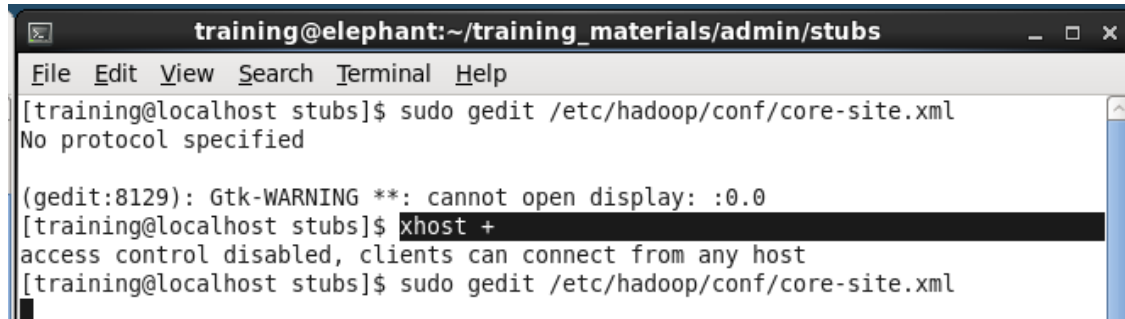
7) Edite o arquivo `/etc/hadoop/conf/core-site.xml` no **elephant** substituindo as propriedades no XML (lembre-se de dar sudo, exemplo) pelo conteúdo abaixo:

```
sudo gedit /etc/hadoop/conf/core-site.xml
```

- **Substituir** o conteúdo do arquivo `core-site.xml` a seguinte configuração:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://elephant:8020</value>
  </property>
</configuration>
```

Em caso de erro:



Isso `xhost +` permite que os clientes se conectem a partir de qualquer host usando `xhost +`

8) Edite o arquivo do hdfs `/etc/hadoop/conf/hdfs-site.xml` no **elephant**, adicionando os seguintes valores de configuração ao arquivo:

```
sudo gedit /etc/hadoop/conf/hdfs-site.xml
```

```
<property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///disk1/dfs/nn,file:///disk2/dfs/nn</value>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///disk1/dfs/dn,file:///disk2/dfs/dn</value>
</property>
```


- 9) Edite o arquivo do yarn/etc/hadoop/conf/yarn-site.xml no elephant, substituindo pelo seguinte conteúdo:

```
sudo gedit /etc/hadoop/conf/yarn-site.xml
```

```
<configuration>
  <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>horse</value>
  </property>

  <property>
    <name>yarn.application.classpath</name>
    <value>
      $HADOOP_CONF_DIR,
      $HADOOP_COMMON_HOME/*,$HADOOP_COMMON_HOME/lib/*,
      $HADOOP_HDFS_HOME/*,$HADOOP_HDFS_HOME/lib/*,
      $HADOOP_MAPRED_HOME/*,$HADOOP_MAPRED_HOME/lib/*,
      $HADOOP_YARN_HOME/*,$HADOOP_YARN_HOME/lib/*
    </value>
  </property>

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

  <property>
    <name>yarn.nodemanager.local-dirs</name>
    <value>file:///disk1/nodemgr/local,file:///disk2/nodemgr/local<
    /value>
  </property>

  <property>
    <name>yarn.nodemanager.log-dirs</name>
    <value>/var/log/hadoop-yarn/containers</value>
  </property>

  <property>
    <name>yarn.nodemanager.remote-app-log-dir</name>
    <value>/var/log/hadoop-yarn/apps</value>
  </property>

  <property>
    <name>yarn.log-aggregation-enable</name>
    <value>true</value>
  </property>
</configuration>
```

- 10) Edite o arquivo do mapred/etc/hadoop/conf/mapred-site.xml no elephant e substituindo pelo seguinte conteúdo:

```
sudo gedit /etc/hadoop/conf/mapred-site.xml
```

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>monkey:10020</value>
  </property>
</configuration>
```

```

        <name>mapreduce.jobhistory.webapp.address</name>
        <value>monkey:19888</value>
    </property>

    <property>
        <name>yarn.app.mapreduce.am.staging-dir</name>
        <value>/user</value>
    </property>
</configuration>

```

- 11) Como estamos rodando Hadoop em VMs, é necessário diminuir o gasto de memória das JVMs. Para isso, é necessário editar o arquivo `/etc/hadoop/conf/hadoop-env.sh` no **elephant**, copiando e colando o seguinte conteúdo:

```

sudo gedit /etc/hadoop/conf/hadoop-env.sh

export HADOOP_NAMENODE_OPTS="-Xmx64m"
export HADOOP_SECONDARYNAMENODE_OPTS="-Xmx64m"
export HADOOP_DATANODE_OPTS="-Xmx64m"
export YARN_RESOURCEMANAGER_OPTS="-Xmx64m"
export YARN_NODEMANAGER_OPTS="-Xmx64m"
export HADOOP_JOB_HISTORYSERVER_OPTS="-Xmx64m"

```

- 12) No **elephant**, Execute o script:

```

/home/training/training_materials/admin/scripts/copy_configuration.sh

```

- 13) No **elephant**, crie os diretórios para o namenode e datanode, depois altere as permissões conforme abaixo:

```

sudo mkdir -p /disk1/dfs/nn
sudo mkdir -p /disk2/dfs/nn
sudo mkdir -p /disk1/dfs/dn
sudo mkdir -p /disk2/dfs/dn
sudo mkdir -p /disk1/nodemgr/local
sudo mkdir -p /disk2/nodemgr/local

sudo chown -R hdfs:hadoop /disk1/dfs/nn
sudo chown -R hdfs:hadoop /disk2/dfs/nn
sudo chown -R hdfs:hadoop /disk1/dfs/dn
sudo chown -R hdfs:hadoop /disk2/dfs/dn

sudo chown -R yarn:yarn /disk1/nodemgr/local
sudo chown -R yarn:yarn /disk2/nodemgr/local

```

- 14) No **elephant**, copie essa estrutura de diretórios para as outras máquinas rodando o script:

```

/home/training/training_materials/admin/scripts/set_up_directories.sh

```

- 15) No **elephant**, formate o Namenode:

```

sudo -u hdfs hdfs namenode -format

```

- 16) Volte a utilizar o usuário training a partir deste passo, inicie o Namenode no **elephant**:

```

sudo service hadoop-hdfs-namenode start

```

- 17) Verifique se deu certo com o comando:

```
sudo jps -v
```

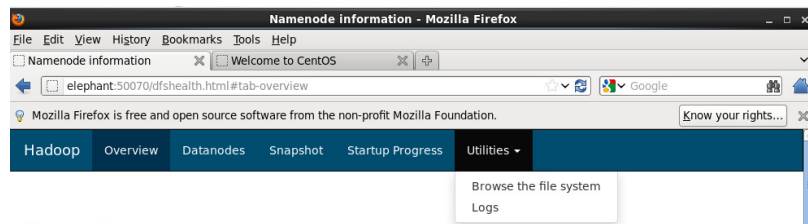
18) Verifique nos logs se deu certo ou se não deu algum erro ao iniciar:

```
less /var/log/hadoop-hdfs/hadoop-hdfs-namenode-elephant.log
less /var/log/hadoop-hdfs/hadoop-hdfs-namenode-elephant.out
```

19) Abra o firefox do Namenode e verifique a interface web acessando a:

```
http://elephant:50070
```

20) No Firefox clique em “**Utilities> Logs**”. Uma lista de arquivos do HDFS aparecerá:



Resultado:

Directory: /logs/

SecurityAuth-hdfs.audit	0 bytes	Sep 18, 2019 4:00:40 PM
hadoop-hdfs-namenode-elephant.log	30730 bytes	Sep 18, 2019 4:07:19 PM
hadoop-hdfs-namenode-elephant.out	718 bytes	Sep 18, 2019 4:00:40 PM

21) No **tigre**, inicie o SecondaryNameNode. Valide sempre cada serviço com o comando `sudo jps` para averiguar se a configuração ocorreu com sucesso.

```
sudo service hadoop-hdfs-secondarynamenode start
sudo jps
```

22) Nos quatro hosts (**tiger, monkey, horse e elephant**), inicie os datanodes:

```
sudo service hadoop-hdfs-datanode start
sudo jps
```

23) No **elephant** de acesso ao grupo **training** para o **HDFS**:

```
sudo -u hdfs hadoop fs -mkdir -p /user/training
sudo -u hdfs hadoop fs -chown training /user/training
```

24) Teste o HDFS subindo um arquivo com os comandos abaixo:

```
hadoop fs -mkdir weblog

cd ~/training_materials/admin/data

gunzip -c access_log.gz \
| hadoop fs -put - weblog/access_log
```

- 25) Verifique se o arquivo aparece na interface do NameNode via web ou com o comando `hadoopfs -ls weblog`. Se você conseguir, você já instalou um cluster com HDFS.

```
hadoop fs -ls /user/training/weblog
```

Resultado:

```
[training@localhost data]$ hadoop fs -ls /user/training/weblog
Found 1 items
-rw-r--r-- 3 training supergroup 504941532 2019-09-19 08:15 /user/training/weblog/access_log
[training@localhost data]$
```

ATIVIDADE - 4

Nesta sessão iremos instalar o **Cluster com serviços para o YARN e MapReduce** e rode **um job mapreduce**.

Siga o passo-a-passo a seguir, prestando bastante atenção em cada atividade e em qual *hostname* estas configurações devem ser aplicadas:

1. No **Elephant**, crie os diretórios necessários para o Yarn e MapReduce:

```
sudo -u hdfs hadoop fs -mkdir /tmp
sudo -u hdfs hadoop fs -chmod -R 1777 /tmp
sudo -u hdfs hadoop fs -mkdir -p /var/log/hadoop-yarn
sudo -u hdfs hadoop fs -chown yarn:mapred /var/log/hadoop-yarn
sudo -u hdfs hadoop fs -mkdir /user/history
sudo -u hdfs hadoop fs -chmod 1777 /user/history
sudo -u hdfs hadoop fs -chown mapred:hadoop /user/history
```

2. No **Horse**, inicie o Resource Manager e lembre-se de checar os logs, se está tudo ok com o comando `sudojps`

```
sudo service hadoop-yarn-resourcemanager start
```

3. Verifique se está tudo ok na interface web:

```
http://horse:8088
```

4. Selecione a opção: **Tools/Local logs**

5. Reveja os logs utilizando a interface web.

6. Nos **4 nodes** do seu cluster, rode o comando para iniciar os **NodeManager**:

```
sudo service hadoop-yarn-nodemanager start
```

7. Inicie o **JobHistoryServer** no **monkey**:

```
sudo service hadoop-mapreduce-historyserver start
```

8. Verifique a interface web do **JobHistory Server** via navegador acessando o endereço:

```
http://monkey:19888
```

ATIVIDADE - 5

Nesta sessão iremos testar o Cluster que configuramos.

Siga o passo-a-passo a seguir, prestando bastante atenção em cada atividade e em qual *hostname* estas configurações devem ser aplicadas:

1. Faça carga de dados para o cluster, no **elephant** digite:

```
cd ~/training_materials/admin/data
gunzip shakespeare.txt.gz
tail shakespeare.txt
hadoop fs -mkdir input
hadoop fs -put shakespeare.txt input
hadoop fs -ls /user
hadoop fs -ls input
hadoop fs -tail input/shakespeare.txt
```

Resultado:

```
[training@localhost data]$ hadoop fs -ls
Found 2 items
drwxr-xr-x - training supergroup          0 2019-09-19 08:40 input
drwxr-xr-x - training supergroup          0 2019-09-19 08:15 weblog
[training@localhost data]$ hadoop fs -ls /user
Found 2 items
drwxrwxrwt - mapred hadoop                0 2019-09-19 08:34 /user/history
drwxr-xr-x - training supergroup          0 2019-09-19 08:39 /user/training
[training@localhost data]$ hadoop fs -ls /user/training
Found 2 items
drwxr-xr-x - training supergroup          0 2019-09-19 08:40 /user/training/input
drwxr-xr-x - training supergroup          0 2019-09-19 08:15 /user/training/weblog
[training@localhost data]$ hadoop fs -ls /user/training/input
Found 1 items
-rw-r--r-- 3 training supergroup    5447165 2019-09-19 08:40 /user/training/input/shakespeare.txt
[training@localhost data]$
```

2. Acesse a interface web do Namenode

`http://elephant:50070`

3. Clique em **Utilities > Browse the Filesystem**

4. Navegue para o diretório:

/user/training/input e selecione o arquivo shakespeare.txt

5. Verifique se o Hadoop replicou o arquivo 3 vezes. Para ver os hosts, veja a informação de disponibilidade para o bloco 0.

6. Execute o comando abaixo e veja as opções de parâmetro desta Lib:

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
```

7. MapReduce – Teste 1 - Execute a lib com o parametro PI 100 20:

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar pi
100 20
```

8. MapReduce – Teste 2 - Execute um job Mapreduce, o Wordcount:

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar  
wordcount input counts
```

9. Verifique o resultado:

```
hadoop fs -ls counts
```

```
[training@elephant ~]$ hadoop fs -ls counts  
Found 2 items  
-rw-r--r--  3 training supergroup          0 2019-09-19 11:54 counts/_SUCCESS  
-rw-r--r--  3 training supergroup    713496 2019-09-19 11:54 counts/part-r-00000  
[training@elephant ~]$ █
```

PROCESSO ENCERRADO – AULA 1
