

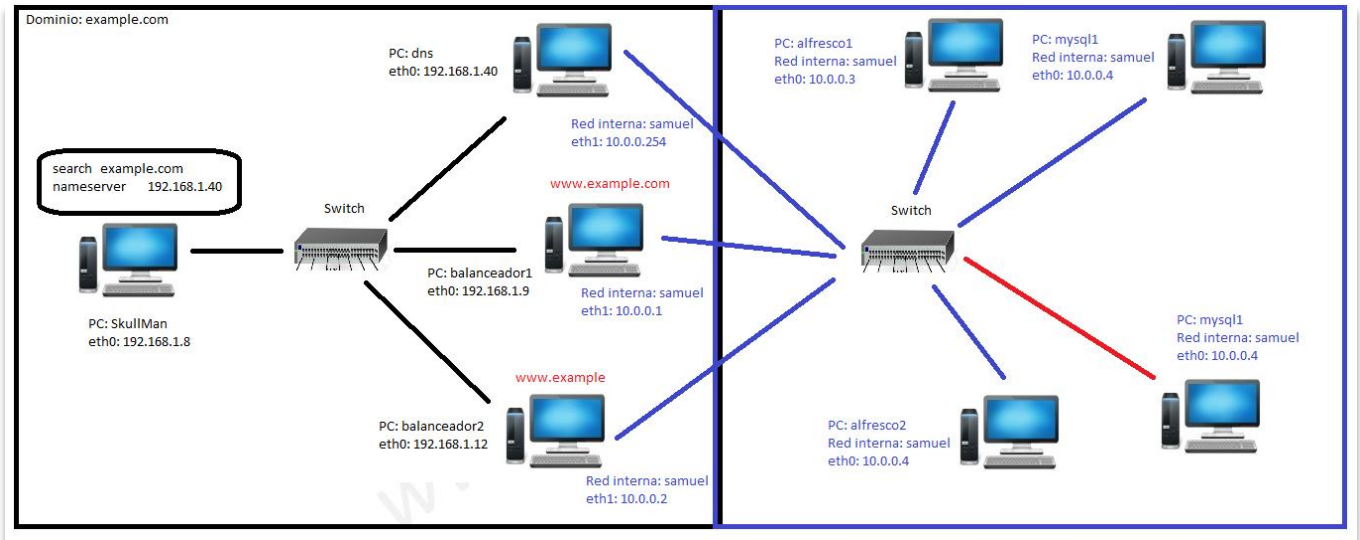


Cluster de Alta Disponibilidad con Balanceo de Carga en Alfresco

INDICE:

1. Escenario.	pág 3
2. Configuración Percona.	pág 3
2.1 Características de Percona.	pág 3
3. Configuración DRBD (RAID en Red).	pág 10
4. Configuración de Alfresco.	pág 24
4.1 Configuración idéntica en ambos nodos.	pág 24
4.2 Configuración propia de cada nodo.	pág 34
5. Configuración del Balanceador de carga HAproxy.	pág 51
6. Configuración del DNS.	pág 56

1. Escenario:



2. Configuración Percona.

2.1 Características de Percona:

Percona es un conocido software de desarrollo libre para MySQL, también se puede implementar en gestores de base de datos como MongoDB, permite aumentar rendimiento y fiabilidad, además de añadir funcionalidades adicionales como:

- Ejecuta consultas más rápido.
- Es un software consolidado en servidores potentes.
- Ahorro económico en hardware.
- Ahorro de tiempo en la puesta a punto y la administración.
- Mejora el tiempo que permanece un servidor sin caerse.

Comenzamos añadiendo en la siguiente ubicación, las máquinas con las que vamos a trabajar:

```
root@mysql2:/home/usuario# nano /etc/hosts
```

```
127.0.0.1    localhost
127.0.1.1    mysql2
10.0.0.6     mysql2
10.0.0.5     mysql1

# The following lines are desirable for IPv6 capable hosts

::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Descargamos el paquete percona en ambas máquinas:

```
root@mysql1:/home/usuario# wget
https://repo.percona.com/apt/percona-release_0.1-3.$(lsb_release
-sc)_all.deb

root@mysql1:/home/usuario# ls
percona-release_0.1-3.wheezy_all.deb
```

```
root@mysql2:/home/usuario# wget
https://repo.percona.com/apt/percona-release_0.1-3.$(lsb_release
-sc)_all.deb

root@mysql2:/home/usuario# ls
percona-release_0.1-3.wheezy_all.deb
```

Instalamos los siguientes paquetes en ambas máquinas:

```
root@mysql1:/home/usuario# dpkg -i percona-release_0.1-3.wheezy_all.deb

root@mysql1:/home/usuario# aptitude update

root@mysql1:/home/usuario# apt-get install percona-xtradb-cluster-server-5.6 percona-xtradb-cluster-client-5.6 percona-xtradb-cluster-galera-3.x -y
```

```
root@mysql2:/home/usuario# dpkg -i percona-release_0.1-3.wheezy_all.deb

root@mysql2:/home/usuario# aptitude update

root@mysql2:/home/usuario# apt-get install percona-xtradb-cluster-server-5.6 percona-xtradb-cluster-client-5.6 percona-xtradb-cluster-galera-3.x -y
```

Creamos el usuario el cual nos servirá para conectarnos con el otro servidor:

```
root@mysql1:/home/usuario# mysql -u root -p
```

```
mysql> CREATE USER 'sstuser'@'localhost' IDENTIFIED BY 's3cretPass';

Query OK, 0 rows affected (0.20 sec)

mysql> GRANT RELOAD, LOCK TABLES, REPLICATION CLIENT ON *.* TO 'sstuser'@'localhost';

Query OK, 0 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.02 sec)
```

Una vez creada, nos vamos al siguiente fichero y realizamos las siguientes configuraciones:

```
root@mysql1:/home/usuario# nano /etc/mysql/my.cnf
```

Añadimos las siguientes debajo de "[mysqld]":

```
[mysqld]

datadir=/var/lib/mysql
user=mysql

# Path to Galera library
wsrep_provider=/usr/lib/libgalera_smm.so

# Empty gcomm address is being used when cluster is getting
bootstrapped

wsrep_cluster_address=gcomm://

# Cluster connection URL contains the IPs of node#1, node#2 and
node#3

wsrep_cluster_address=gcomm://10.0.0.5,10.0.0.6

# In order for Galera to work correctly binlog format should be
ROW

binlog_format=ROW

# MyISAM storage engine has only experimental support
default_storage_engine=InnoDB

# This is a recommended tuning variable for performance
innodb_locks_unsafe_for_binlog=1

# This changes how InnoDB autoincrement locks are managed and is
a requirement for$

innodb_autoinc_lock_mode=2

# Node #1 address
wsrep_node_address=10.0.0.5

# SST method
wsrep_sst_method=xtrabackup-v2

# Cluster name
```

```
wsrep_cluster_name=cluster
#Name of node
wsrep_node_name=mysql1
# Authentication for SST method
wsrep_sst_auth="sstuser:s3cretPass"
```

Y en la siguiente línea añadimos el valor:

```
bind-address = 0.0.0.0
```

En el nodo2 realizamos la siguiente configuración:

```
root@mysql2:/home/usuario# nano /etc/mysql/my.cnf
```

Debajo del "[mysqld]" añadimos el siguiente contenido:

```
[mysqld]

datadir=/var/lib/mysql
user=mysql
wsrep_provider=/usr/lib/libgalera_smm.so
### IP addresses of your two nodes.
wsrep_cluster_address=gcomm://10.0.0.5,10.0.0.6
### This will not work unless binlog is formatted to ROW.
binlog_format=ROW

### This changes how InnoDB autoincrement locks are managed and
is a

### requirement for Galera
innodb_autoinc_lock_mode=2
```

```
### Node #2 address (The node you are ssh`d into now.)
wsrep_node_address=10.0.0.6

### SST Method
wsrep_sst_method=xtrabackup-v2

### Cluster name
wsrep_cluster_name=cluster

### Node Name, in this case we will just call it xtradb1
wsrep_node_name=mysql2

### Authentication, REMEMBER THIS.
wsrep_sst_auth="sstuser:s3cretPass"
```

Por último lanzamos mysql1 con el siguiente comando:

```
root@mysql1:/home/usuario# /etc/init.d/mysql bootstrap-pxc
[ ok ] Bootstrapping Percona XtraDB Cluster database server:
mysqld already running.
```

Y mysql2 con el siguiente comando:

```
root@mysql2:/home/usuario# /etc/init.d/mysql start
[ ok ] Starting MySQL (Percona XtraDB Cluster) database server:
mysqld . . . . .
```

Realizamos la prueba, en la máquina mysql1 vamos a apagar el servidor mysql:

```
root@mysql1:/home/usuario# /etc/init.d/mysql stop
[ ok ] Stopping MySQL (Percona XtraDB Cluster): mysqld.
```


En el nodo dos vamos a crear una base de datos llamada "samuel":

```
root@mysql2:/home/usuario# mysql -uroot -proot
Warning: Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
```

```
mysql> create database samuel;
Query OK, 1 row affected (0.09 sec)
```

A continuación volvemos a levantar el servicio mysql en mysql1 y debería replicar la base de datos "samuel":

```
root@mysql1:/home/usuario# /etc/init.d/mysql start
[....] Starting MySQL (Percona XtraDB Cluster) database server:
mysqld . . . .[...[ ok ate transfer in progress, setting sleep
higher: mysqld ..
```

Realizamos la consulta de que efectivamente la base de datos "samuel" ha sido copiada del nodo mysql2:

```
root@mysql1:/home/usuario# mysql -u root -proot
Warning: Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
```

```
mysql> show databases;
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| performance_schema |
| samuel            |
| test              |
+-----+
5 rows in set (0.07 sec)

mysql>
```

3. Configuración DRBD (RAID1 en red entre varios nodos):

DRBD es un software que permite hacer réplica de los datos de una partición entre varias máquinas. Es decir, que si tengo una partición del mismo tamaño en dos máquinas, con DRBD puedo hacer una réplica del contenido de esta partición de forma automática, para que en el caso de que una máquina falle, tenga todo el contenido de esa partición accesible desde la otra máquina. Es como un RAID1, pero entre distintas máquinas. Habitualmente, esta partición de la que se hace mirror, solamente está montada en una de las máquinas porque se utiliza un sistema de ficheros tradicional: ext3, raiserfs, xfs, ... De esta forma, solo una de las máquinas puede acceder a los datos, la que tiene la partición montada. Sirve para montar un sistema de cluster en modo "activo/pasivo", y que una de las máquinas tenga todos los datos hasta que falle, y en ese momento se puede acceder desde la otra máquina.

Pero también se puede configurar para que ambas máquinas tengan acceso a la partición en espejo, y en este caso habría que montar un sistema de ficheros para acceso en clúster, como GFS o OCFS.

De esta forma podemos montar un clúster "activo/activo" donde ambas máquinas tienen acceso simultáneo al recurso de datos.*

Comenzamos primeramente, montando un disco duro de 8 Gigas en cada uno de los nodos, y a continuación instalamos los siguientes paquetes:*

```
root@alfresco1:/home/usuario# apt-get install drbd8-utils
```

```
root@alfresco2:/home/usuario# apt-get install drbd8-utils
```

A continuación activamos drbd en ambos nodos con el siguiente comando:

```
root@alfresco1:/home/usuario# modprobe drbd
```

```
root@alfresco2:/home/usuario# modprobe drbd
```

Ahora definimos el fichero de configuración de drbd en ambos nodos, pero antes realizamos una copia del original, por si hay que volver a la configuración anterior:*

```
root@alfresco1:/home/usuario# cp /etc/drbd.conf
/etc/drbd.conf_orig
root@alfresco1:/home/usuario# cat /dev/null > /etc/drbd.conf
root@alfresco1:/home/usuario# nano /etc/drbd.conf
```

Y añadimos el siguiente contenido:

```
resource r0 {
    meta-disk internal;
    device /dev/drbd0;
```

```
disk /dev/sdb1;

syncer { rate 1000M; }

net {

    allow-two-primaries;

    after-sb-0pri discard-zero-changes;

    after-sb-1pri discard-secondary;

    after-sb-2pri disconnect;

}

startup { become-primary-on both; }

on alfresco1 { address 10.0.0.3:7789; }

on alfresco2 { address 10.0.0.4:7789; }

}
```

En nodo2:

```
root@alfresco2:/home/usuario# cp /etc/drbd.conf
/etc/drbd.conf_orig
root@alfresco2:/home/usuario# cat /dev/null > /etc/drbd.conf
root@alfresco2:/home/usuario# nano /etc/drbd.conf
```

```
resource r0 {

    meta-disk internal;

    device /dev/drbd0;

    disk /dev/sdb1;

    syncer { rate 1000M; }

    net {
```

```
        allow-two-primaries;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri discard-secondary;
        after-sb-2pri disconnect;
    }
    startup { become-primary-on both; }

    on alfresco1 { address 10.0.0.3:7789; }
    on alfresco2 { address 10.0.0.4:7789; }
}
```

Lo siguiente que vamos a hacer antes de lanzar nuestro raid en red, es crear una partición primaria en el disco duro que hemos añadido en cada uno de los nodos:

```
root@alfresco1:/home/usuario# fdisk /dev/sdb
```

Elegimos las opciones "n, p, tecla enter, tecla enter, tecla enter y w".

Comprobamos que la partición primaria ha sido creada:

```
root@alfresco1:/home/usuario# fdisk /dev/sdb

Bienvenido a fdisk (util-linux 2.25.2).

Los cambios solo permanecerán en la memoria, hasta que decida
escribirlos.

Tenga cuidado antes de utilizar la orden de escritura.

Orden (m para obtener ayuda): p
```

```
Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x371565a6
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	16777215	16775168	8G	83	Linux

Igual con el nodo2:

```
root@alfresco2:/home/usuario# fdisk /dev/sdb
```

Elegimos las opciones "n, p, tecla enter, tecla enter, tecla enter y w".

Comprobamos que la partición primaria ha sido creada:

```
root@alfresco2:/home/usuario# fdisk /dev/sdb
```

Bienvenido a fdisk (util-linux 2.25.2).

Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.

Tenga cuidado antes de utilizar la orden de escritura.

Orden (m para obtener ayuda): p

```
Disco /dev/sdb: 8 GiB, 8589934592 bytes, 16777216 sectores
```

```
Unidades: sectores de 1 * 512 = 512 bytes
```

```
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x921abe0b

Device      Boot Start          End  Sectors  Size Id Type
/dev/sdb1           2048 16777215 16775168    8G 83 Linux
```

Una vez creada la partición, lanzamos el comando de creación del raid en red en ambos nodos:

```
root@alfresco1:/home/usuario# drbdadm create-md r0

      ---- This is a new installation of DRBD ----

Please take part in the global DRBD usage count at
http://usage.drbd.org.

</pre>

<pre>
root@alfresco2:/home/usuario# drbdadm create-md r0

      ---- This is a new installation of DRBD ----

Please take part in the global DRBD usage count at
http://usage.drbd.org.
```

A continuación lanzamos el servicio, y nos mostrará un fallo, esto es porque drbd debe saber cuál de los nodos primarios va a lanzar nuestro raid en red:

```
root@alfresco1:/home/usuario# /etc/init.d/drbd start

[....] Starting drbd (via systemctl): drbd.serviceJob for
drbd.service failed. See 'systemctl status drbd.service' and
'journalctl -xn' for details.
```

```
failed!
```

```
root@alfresco2:/home/usuario# /etc/init.d/drbd start

[....] Starting drbd (via systemctl): drbd.serviceJob for
drbd.service failed. See 'systemctl status drbd.service' and
'journalctl -xn' for details.

failed!
```

Indicamos cuál de los nodos va a comenzar siendo primario:

```
root@alfresco1:/home/usuario# drbdadm -- --overwrite-data-of-
peer primary all
```

Comprobamos desde el nodo 2 que los discos se están sincronizando y esperemos hasta que se sincronice del todo:

```
root@alfresco2:/home/usuario# cat /proc/drbd
version: 8.4.3 (api:1/proto:86-101)
srcversion: 1A9F77B1CA5FF92235C2213
 0: cs:SyncTarget ro:Primary/Primary ds:Inconsistent/UpToDate C
r-----
      ns:0 nr:5215352 dw:5215232 dr:792 al:0 bm:318 lo:0 pe:0 ua:0
ap:0 ep:1 wo:f oos:3172060
      [======>.....] sync'ed: 62.3%
(3096/8188)Mfinish: 0:01:33 speed: 33,740 (32,392) want: 0 K/sec
```

A continuación instalamos el sistema de ficheros o2fs2 del que antes habíamos mencionado anteriormente:

```
root@alfresco1:/home/usuario# aptitude install ocfs2-tools
```



```
root@alfresco1:/home/usuario# aptitude install ocfs2-tools
```

Una vez instalado el sistema de ficheros, en ambos nodos debemos mencionar cuales son las máquinas que van a poner acceder al cluster, para ello definimos el siguiente fichero en ambos nodos:

```
root@alfresco1:/home/usuario# nano /etc/ocfs2/cluster.conf
```

```
cluster:
    node_count = 2
    name = web

node:
    ip_port = 7777
    ip_address = 10.0.0.3
    number = 1
    name = alfresco1
    cluster = web

node:
    ip_port = 7777
    ip_address = 10.0.0.4
    number = 2
    name = alfresco2
    cluster = web
```

```
root@alfresco2:/home/usuario# nano /etc/ocfs2/cluster.conf
```

```
cluster:
    node_count = 2
    name = web

node:
    ip_port = 7777
    ip_address = 10.0.0.3
    number = 1
    name = alfresco1
    cluster = web

node:
    ip_port = 7777
    ip_address = 10.0.0.4
    number = 2
    name = alfresco2
    cluster = web
```

Una vez que tengamos definidos ambos ficheros, realizamos los siguientes pasos en cada uno de los nodos:

Iniciamos el servicio:

```
root@alfresco1:/home/usuario# service o2cb start
```

Formateamos el raid en red con el sistema de fichero indicado anteriormente:

```
root@alfresco1:/home/usuario# mkfs.ocfs2 -L "web" /dev/drbd0
mkfs.ocfs2 1.6.4
Cluster stack: classic o2cb
Label: web
Features: sparse backup-super unwritten inline-data strict-
journal-super xattr
Block size: 4096 (12 bits)
Cluster size: 4096 (12 bits)
Volume size: 8588587008 (2096823 clusters) (2096823 blocks)
Cluster groups: 66 (tail covers 183 clusters, rest cover 32256
clusters)
Extent allocator size: 4194304 (1 groups)
Journal size: 67108864
Node slots: 4
Creating bitmaps: done
Initializing superblock: done
Writing system files: done
Writing superblock: done
Writing backup superblock: 2 block(s)
Formatting Journals: done
Growing extent allocator: done
Formatting slot map: done
Formatting quota files: done
Writing lost+found: done
mkfs.ocfs2 successful
```

Creamos el directorio donde más tarde replicaremos los documentos de alfresco:

```
root@alfresco1:/home/usuario# mkdir -p /alfresco/documentos
```

Añadimos la siguiente línea al fichero fstab para que se monte automáticamente:

```
root@alfresco1:/home/usuario# echo "/dev/drbd0  
/alfresco/documentos ocfs2 noauto,noatime 0 0" >> /etc/fstab
```

Cargamos el servicio:

```
root@alfresco1:/home/usuario# /etc/init.d/o2cb load  
Loading filesystem "configfs": OK  
Loading stack plugin "o2cb": OK  
Loading filesystem "ocfs2_dlmfs": OK  
Creating directory '/dlm': OK  
Mounting ocfs2_dlmfs filesystem at /dlm: OK
```

Le indicamos a ocfs2 que utilice replique el servicio del cluster definido anteriormente:

```
root@alfresco1:/home/usuario# /etc/init.d/o2cb online web
```

Lo montamos:

```
root@alfresco1:/home/usuario# mount /dev/drbd0  
/alfresco/documentos/
```

Añadimos la siguiente configuración al siguiente fichero, esto hace que cuando reinicie la máquina te ejecute automáticamente el servicio y te lo monte en el directorio:

```
root@alfresco1:/home/usuario# nano /etc/rc.local
```

```
/etc/init.d/o2cb online web  
  
sleep 10  
  
mount /dev/drbd0 /alfresco/documentos  
  
exit 0
```

En el nodo 2 lanzamos el servicio:

```
root@alfresco2:/home/usuario# /etc/init.d/o2cb online web  
Loading filesystem "configfs": OK  
Mounting configfs filesystem at /sys/kernel/config: OK  
Loading stack plugin "o2cb": OK  
Loading filesystem "ocfs2_dlmfs": OK  
Creating directory '/dlm': OK  
Mounting ocfs2_dlmfs filesystem at /dlm: OK  
Setting cluster stack "o2cb": OK  
Starting O2CB cluster web: OK
```

Creamos el directorio:

```
root@alfresco2:/home/usuario# mkdir -p /alfresco/documentos
```

Añadimos la línea al fstab:

```
root@alfresco2:/home/usuario# echo "/dev/drbd0  
/alfresco/documentos ocfs2 noauto,noatime 0 0" >> /etc/fstab
```

Montamos:

```
root@alfresco2:/home/usuario# mount /dev/drbd0
```

Realizamos la misma configuración que anteriormente:

```
root@alfresco2:/home/usuario# nano /etc/rc.local
```

```
/etc/init.d/o2cb online web  
  
sleep 10  
  
mount /dev/drbd0  
  
exit 0
```

Por último recargamos el servicio:

```
root@alfresco2:/home/usuario# /etc/init.d/o2cb load
```

Primera comprobación, vamos a ver si drbd está montado en el directorio que le hemos especificado:

```
root@alfresco1:/alfresco/documentos# df -h
```

S.ficheros	Tamaño	Usados	Disp	Uso%	Montado en
/dev/sda1	14G	1,3G	13G	10%	/
udev	10M	0	10M	0%	/dev
tmpfs	201M	4,5M	196M	3%	/run
tmpfs	501M	0	501M	0%	/dev/shm
tmpfs	5,0M	0	5,0M	0%	/run/lock
tmpfs	501M	0	501M	0%	/sys/fs/cgroup
tmpfs	101M	0	101M	0%	/run/user/0
tmpfs	101M	0	101M	0%	/run/user/1000

```
/dev/drbd0      8,0G   279M   7,8G   4% /alfresco/documentos
```

```
root@alfresco2:/alfresco/documentos# df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
/dev/sda1       14G    1,3G   13G  10% /
udev            10M     0    10M   0% /dev
tmpfs           201M    4,5M  196M   3% /run
tmpfs           501M     0   501M   0% /dev/shm
tmpfs           5,0M     0   5,0M   0% /run/lock
tmpfs           501M     0   501M   0% /sys/fs/cgroup
tmpfs           101M     0   101M   0% /run/user/1000
/dev/drbd0      8,0G    279M   7,8G   4% /alfresco/documentos
```

Segunda comprobación, apagamos el nodo 1, creamos un fichero en nodo 2 y cuando levantemos nodo 1 el fichero debe aparecer:

```
root@alfresco1:/alfresco/documentos# poweroff
Connection to 10.0.0.3 closed by remote host.
Connection to 10.0.0.3 closed.
```

```
root@alfresco2:/home/usuario# cd /alfresco/documentos/
root@alfresco2:/alfresco/documentos# touch 02
root@alfresco2:/alfresco/documentos# ls
01 02 lost+found
```

Arrancamos nodo1 y comprobamos que efectivamente aparece el fichero creado:

```
root@alfresco1:/home/usuario# cd /alfresco/documentos/
```

```
root@alfresco1:/alfresco/documentos# ls
01  02  lost+found
</pre>
```

4. Configuración de Alfresco:

La configuración de alfresco la vamos a dividir en dos partes:

- La primera parte de la configuración será idéntica en ambos nodos.
- La segunda parte de la configuración añadiremos los parámetros propios de cada nodo.

4.1. Configuración idéntica en ambos nodos:

Requisitos mínimos de la máquina:

1. 1GB de RAM.
2. 15GB de Disco Duro.

Pasos a realizar:

1. Instalar JRE 7.
2. Instalar Tomcat.
3. Configurar el conector de MySQL.

Comenzamos instalando el paquete java en ambos nodos:

```
root@alfresco1:/ahome/usuario# aptitude install openjdk-7-jre
```


A continuación descargamos tomcat desde la página oficial "<http://tomcat.apache.org>", lo descomprimos y procedemos a su instalación:

```
root@alfresco1:/home/usuario# wget
http://apache.rediris.es/tomcat/tomcat-7/v7.0.69/bin/apache-
tomcat-7.0.69.tar.gz
```

```
root@alfresco1:/home/usuario# tar xvzf apache-tomcat-
7.0.69.tar.gz
```

Una vez que tengamos los ficheros descomprimidos, movemos la carpeta al directorio "/opt" cambiándolo de nombre "/opt/tomcat" llamada tomcat:

```
root@alfresco1:/home/usuario# mv apache-tomcat-7.0.69
/opt/tomcat
```

Lo siguiente que vamos a hacer es modificar el fichero "tomcat-users.xml" para añadir el usuario administrador:

```
root@alfresco1:/home/usuario# nano /opt/tomcat/conf/tomcat-
users.xml
```

Y modificamos las siguientes líneas a partir de "tomcat-users":

```
<tomcat-users>
<!--
NOTE: By default, no user is included in the "manager-gui" role
required
to operate the "/manager/html" web application. If you wish to
use this app,
you must define such a user - the username and password are
arbitrary.
```

```
-->
<!--
NOTE: The sample user and role entries below are wrapped in a
comment
and thus are ignored when reading this file. Do not forget to
remove
<!-- ..> that surrounds them.
-->
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="admin" password="admin" roles="admin-gui,
manager-gui, manager-script, admin-script" />
</tomcat-users>
```

Con esto conseguimos que podamos acceder al tomcat con el usuario "admin" y la credencial "admin".

A continuación modificamos el fichero "server.xml":

```
root@alfrescol:/opt/tomcat/conf# nano
/opt/tomcat/conf/server.xml
```

Y en la línea donde aparece la etiqueta conector, modificamos los siguientes parámetros dejándolo de la siguiente forma:

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443"
           URIEncoding="UTF-8" />
```

A continuación creamos un script que permita el arranque de tomcat automáticamente, la ruta será "/etc/init.d/tomcat":

```
root@alfresco1:/opt/tomcat/conf# nano /etc/init.d/tomcat
```

E introducimos el siguiente contenido:

```
# Tomcat auto-start
#
# description: Auto-starts tomcat
# processname: tomcat
# pidfile: /var/run/tomcat.pid

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

case $1 in
start)
sh /opt/tomcat/bin/startup.sh
;;
stop)
sh /opt/tomcat/bin/shutdown.sh
;;
restart)
sh /opt/tomcat/bin/shutdown.sh
sh /opt/tomcat/bin/startup.sh
;;
esac

exit 0
```

</pre>

Una vez creado el script, le damos permisos de ejecución:

```
root@alfresco1:/opt/tomcat/conf# cd /opt/tomcat/
root@alfresco1:/opt/tomcat# chmod +x bin/*.sh
root@alfresco1:/opt# cd conf/
root@alfresco1:/opt/tomcat/conf# chmod +x /etc/init.d/tomcat
```

Lo siguiente será añadir el fichero para permitir el arranque cuando iniciemos la maquina de forma automática, esto se consigue con el siguiente comando:

```
root@alfresco1:/opt/tomcat/conf# cd /etc/init.d/
root@alfresco1:/etc/init.d# update-rc.d tomcat defaults
insserv: warning: script 'tomcat' missing LSB tags and overrides
```

A continuación nos tenemos que bajar el instalador de alfresco de la pagina web:

```
root@alfresco1:/etc/init.d# cd /home/usuario/
root@alfresco1:/home/usuario# wget
http://dl.alfresco.com/release/community/5.0.c-build-
00145/alfresco-community-5.0.c.zip
```

Lo descomprimos:

```
root@alfresco1:/home/usuario# unzip alfresco-community-5.0.c.zip
```

Cambiamos el nombre al directorio:

```
root@alfresco1:/home/usuario# mv alfresco-community-5.0.c
alfresco
```

Modificamos el siguiente fichero:

```
root@alfresco1:/home/usuario# nano /opt/tomcat/bin/catalina.sh
```

Y añadimos la siguiente línea, Esto permite usar 1GB de la memoria RAM asignada:

```
JAVA_OPTS="-XX:MaxPermSize=160m -XX:NewSize=256m -Xms512m "  
JAVA_OPTS="$JAVA_OPTS -Xmx512m -Xss512K "  
  
# OS specific support. $var _must_ be set to either true or  
false.  
  
cygwin=false  
darwin=false  
os400=false
```

Una vez que guardemos los cambios, reiniciamos tomcat:

```
root@alfresco1:/opt/tomcat/bin# /etc/init.d/tomcat start
```

Mostramos la salida del tomcat arrancado:

```
Using CATALINA_BASE:   /opt/tomcat  
Using CATALINA_HOME:   /opt/tomcat  
Using CATALINA_TMPDIR: /opt/tomcat/temp  
Using JRE_HOME:        /usr/lib/jvm/java-7-openjdk-amd64  
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar  
Tomcat started.
```

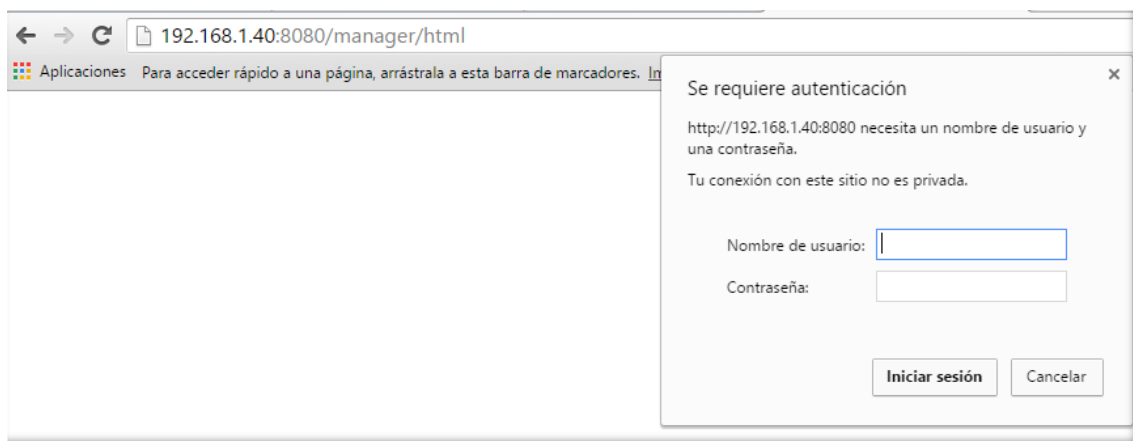
Comprobamos que el tomcat está funcionando:

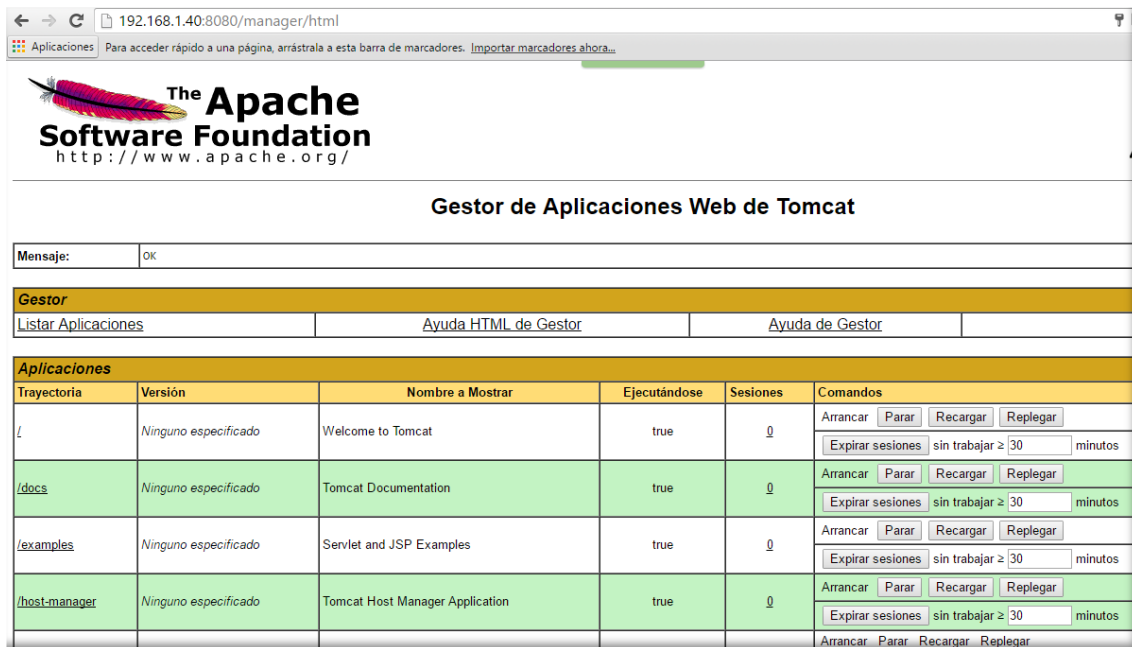
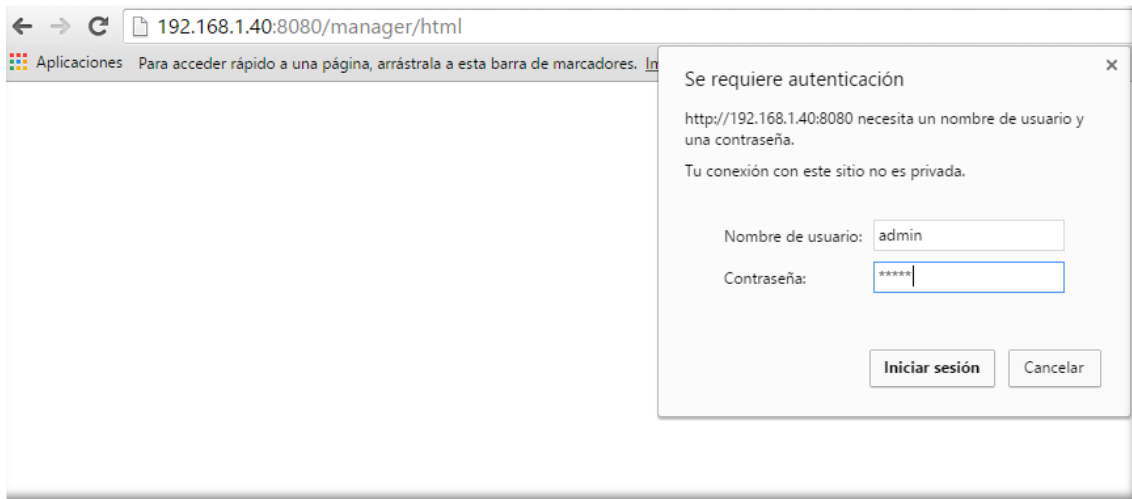
```
root@alfresco1:/home/usuario# ps -aux | grep java
root      4769  13.5  13.7 1384644 140700 pts/0    Sl   13:15   0:06
/usr/lib/jvm/java-7-openjdk-amd64/bin/java -
Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
-XX:MaxPermSize=160m -XX:NewSize=256m -Xms512m -Xmx512m -Xss512K
-Djdk.tls.ephemeralDHKeySize=2048 -
Djava.endorsed.dirs=/opt/tomcat/endorsed -classpath
/opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar -
Dcatalina.base=/opt/tomcat -Dcatalina.home=/opt/tomcat -
Djava.io.tmpdir=/opt/tomcat/temp
org.apache.catalina.startup.Bootstrap start
```

A continuación probamos entrar en la configuración de tomcat añadiendo la* siguiente URL en el navegador:

```
http://10.0.0.3:8080/manager/html
```

Ponemos las siguientes credenciales, usuario: "admin" y contraseña "admin":





Una vez dentro nos vamos al apartado Desplegar y modificamos los siguientes parámetros:

Trayectoria de contexto: **/alfresco**

URL de WAR: **/home/usuario/alfresco/web-server/webapps/alfresco.war**

Pulsamos Desplegar y nos saldrá el siguiente error:

FALLO - Aplicación desplegada en la ruta de contexto /alfresco, pero el contexto no pudo arrancar

Repetimos el proceso pero esta vez con el fichero "share.war":

Trayectoria de contexto: **/share**

URL de WAR: **/home/usuario/alfresco/web-server/webapps/share.war**

Le damos de nuevo al botón Desplegar y nos saltará con un mensajito como que es correcto:

OK - Desplegada aplicación en trayectoria de contexto /share

Lo siguiente que haremos será bajarnos el conector de MySQL para Java:

```
root@alfresco1:/home/usuario# wget
https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-
java-5.0.8.tar.gz
```

Lo descomprimos:

```
root@alfresco1:/home/usuario# tar xvzf mysql-connector-java-
5.0.8.tar.gz
```

Una vez que lo tengamos descomprimido, entramos en el directorio:

```
root@alfresco1:/home/usuario# cd mysql-connector-java-5.0.8
```


Copiamos el fichero ".jar" al siguiente directorio "/opt/tomcat/lib/":

```
root@alfresco1:/home/usuario/mysql-connector-java-5.0.8# cp
mysql-connector-java-5.0.8-bin.jar /opt/tomcat/lib/
```

Luego copiamos el directorio "/shared" dentro de tomcat:

```
root@alfresco1:/home/usuario/mysql-connector-java-5.0.8# cd
/home/usuario/alfresco/web-server/

root@alfresco1:/home/usuario/alfresco/web-server# cp -r shared/
/opt/tomcat/
```

Instalamos las siguientes dependencias:

```
root@alfresco1:/home/usuario/alfresco/web-server# aptitude
install imagemagick

root@alfresco1:/home/usuario/alfresco/web-server# aptitude
install swftools

root@alfresco1:/home/usuario/alfresco/web-server# aptitude
install libjodconverter-java
```

Ahora Nos situamos en el directorio "/opt/tomcat/shared/classes/alfresco/extension/" y copiamos todos los ficheros xml:

```
root@alfresco1:/home/usuario/alfresco/web-server# cd
/opt/tomcat/shared/classes/alfresco/extension/

root@alfresco1:/opt/tomcat/shared/classes/alfresco/extension# cp
video-thumbnail-context.xml.sample video-thumbnail-context.xml

root@alfresco1:/opt/tomcat/shared/classes/alfresco/extension# cp
video-transformation-context.xml.sample video-transformation-
context.xml
```

A continuación modificaremos el siguiente fichero `"/opt/tomcat/conf/catalina.properties"`:

```
root@alfresco1:/opt/tomcat/shared/classes/alfresco/extension#  
nano /opt/tomcat/conf/catalina.properties
```

Y modificamos el parámetro `?shared.loader?` dejándolo de la siguiente forma:

```
shared.loader=${catalina.base}/shared/classes,${catalina.base}/lib/*  
lib/*.jar
```

4.2 Configuración propia de cada nodo:

El siguiente paso será situarnos en el directorio `"/opt/tomcat/shared/classes/alfresco/extension"` para modificar los parámetros de la base de datos:

```
root@alfresco1:/opt/tomcat/shared/classes/alfresco/extension# cd  
/opt/tomcat/shared/classes/  
  
root@alfresco1:/opt/tomcat/shared/classes# cp alfresco-  
global.properties.sample alfresco-global.properties  
  
root@alfresco1:/opt/tomcat/shared/classes# nano alfresco-  
global.properties
```

Los parámetros a configurar serán:

```
#Directorio donde guardaremos los documentos, hay que crearlo.  
dir.root=/alfresco/documentos
```

```
#El usuario y la contraseña que crearemos en nuestro servidor de  
base de datos.  
db.username=alfresco
```

```
db.password=alfresco
```

```
# External locations
#-----
ooo.exe=/usr/bin/libreoffice
ooo.enabled=true
jodconverter.officeHome=/usr/lib/libreoffice/program/soffice
jodconverter.portNumbers=8100
jodconverter.enabled=true
img.root=/usr
img.dyn=${img.root}/lib
img.exe=${img.root}/bin/convert
swf.exe=/usr/bin/pdf2swf
ffmpeg.exe=/usr/bin/ffmpeg
```

#El conector de mysql, para que alfresco pueda utilizar el gestor de base de datos mysql.

```
db.driver=org.gjt.mm.mysql.Driver
db.url=jdbc:mysql://10.0.0.5:3306/alfresco?useUnicode=yes&characterEncoding=UTF-8
```

En el nodo2 realizamos la misma configuración, cambiando el parámetro de la base de datos:

```
root@alfresco2:/opt/tomcat/shared/classes/alfresco/extension# cd
/opt/tomcat/shared/classes/
root@alfresco2:/opt/tomcat/shared/classes# cp alfresco-
global.properties.sample alfresco-global.properties
root@alfresco2:/opt/tomcat/shared/classes# nano alfresco-
global.properties
```

Los parámetros a configurar serán:

```
#Directorio donde guardaremos los documentos, hay que crearlo.  
dir.root=/alfresco/documentos
```

```
#El usuario y la contraseña que crearemos en nuestro servidor de  
base de datos.
```

```
db.username=alfresco
```

```
db.password=alfresco
```

```
# External locations
```

```
#-----
```

```
ooo.exe=/usr/bin/libreoffice
```

```
ooo.enabled=true
```

```
jodconverter.officeHome=/usr/lib/libreoffice/program/soffice
```

```
jodconverter.portNumbers=8100
```

```
jodconverter.enabled=true
```

```
img.root=/usr
```

```
img.dyn=${img.root}/lib
```

```
img.exe=${img.root}/bin/convert
```

```
swf.exe=/usr/bin/pdf2swf
```

```
ffmpeg.exe=/usr/bin/ffmpeg
```

```
#El conector de mysql, para que alfresco pueda utilizar el  
gestor de base de datos mysql.
```

```
db.driver=org.gjt.mm.mysql.Driver
```

```
db.url=jdbc:mysql://10.0.0.6:3306/alfresco?useUnicode=yes&charac  
terEncoding=UTF-8
```

A continuación lo que vamos a hacer es crear el usuario "alfresco" y la base de datos "alfresco" en uno de los servidores mysql. Ya que al hacerlo en el otro se replicará automáticamente:

```
root@mysql1:/home/usuario# mysql -uroot -proot
Warning: Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.30-76.3-56 Percona XtraDB Cluster (GPL),
Release rel76.3, Revision aa929cb, WSREP version 25.16,
wsrep_25.16
Copyright (c) 2009-2016 Percona LLC and/or its affiliates
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All
rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.
mysql>
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.
```

```
mysql> CREATE DATABASE alfresco CHARACTER SET 'UTF8';
Query OK, 1 row affected (0.42 sec)
```

```
mysql> CREATE USER alfresco identified by 'alfresco';
Query OK, 0 rows affected (0.65 sec)

mysql> GRANT ALL PRIVILEGES ON alfresco.* to alfresco;
Query OK, 0 rows affected (0.09 sec)
```

Comprobamos que la base de datos "alfresco" ha sido replicada en el otro nodo:

```
root@mysql2:/home/usuario# mysql -u root -proot
Warning: Using a password on the command line interface can be
insecure.

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 6

Server version: 5.6.30-76.3-56 Percona XtraDB Cluster (GPL),
Release rel76.3, Revision aa929cb, WSREP version 25.16,
wsrep_25.16

Copyright (c) 2009-2016 Percona LLC and/or its affiliates
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All
rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or
its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
input statement.

mysql> show databases;
+-----+
```

```
| Database |
+-----+
| information_schema |
| alfresco |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.24 sec)

mysql>
```

Una vez que esté todo finalizado, reiniciamos los tomcat en ambos nodos de alfresco:

```
root@alfrescol1:/opt/tomcat/shared/classes# /etc/init.d/tomcat
restart

Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:
/opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
jun 15, 2016 7:04:15 PM org.apache.catalina.startup.Catalina
stopServer

GRAVE: Could not contact localhost:8005. Tomcat may not be
running.

jun 15, 2016 7:04:16 PM org.apache.catalina.startup.Catalina
stopServer

GRAVE: Catalina.stop:

java.net.ConnectException: Conexión rehusada

    at java.net.PlainSocketImpl.socketConnect(Native Method)
```

```
    at
java.net.AbstractPlainSocketImpl.doConnect (AbstractPlainSocketIm
pl.java:339)

    at
java.net.AbstractPlainSocketImpl.connectToAddress (AbstractPlainS
ocketImpl.java:200)

    at
java.net.AbstractPlainSocketImpl.connect (AbstractPlainSocketImpl
.java:182)

    at
java.net.SocksSocketImpl.connect (SocksSocketImpl.java:392)
    at java.net.Socket.connect (Socket.java:579)
    at java.net.Socket.connect (Socket.java:528)
    at java.net.Socket.<init> (Socket.java:425)
    at java.net.Socket.<init> (Socket.java:208)

    at
org.apache.catalina.startup.Catalina.stopServer (Catalina.java:49
8)

    at sun.reflect.NativeMethodAccessorImpl.invoke0 (Native
Method)

    at
sun.reflect.NativeMethodAccessorImpl.invoke (NativeMethodAccessor
Impl.java:57)

    at
sun.reflect.DelegatingMethodAccessorImpl.invoke (DelegatingMethod
AccessorImpl.java:43)

    at java.lang.reflect.Method.invoke (Method.java:606)

    at
org.apache.catalina.startup.Bootstrap.stopServer (Bootstrap.java:
343)

    at
org.apache.catalina.startup.Bootstrap.main (Bootstrap.java:430)

Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
```



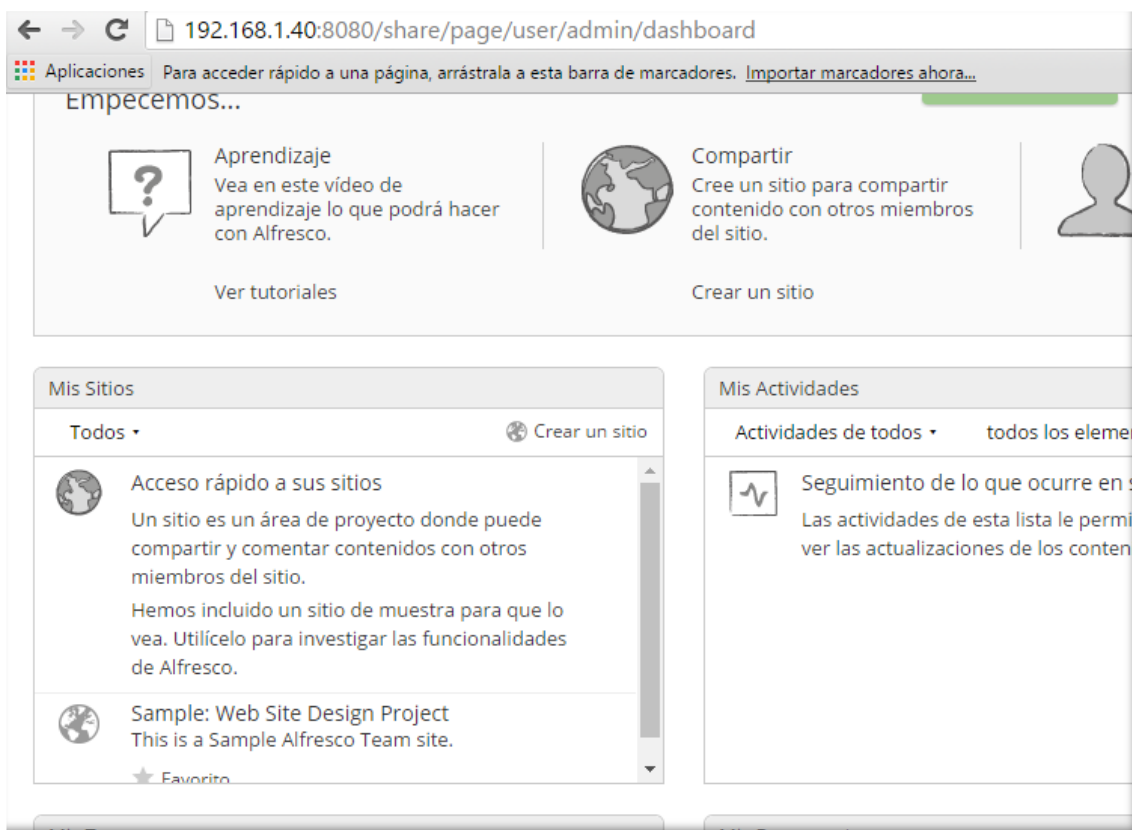
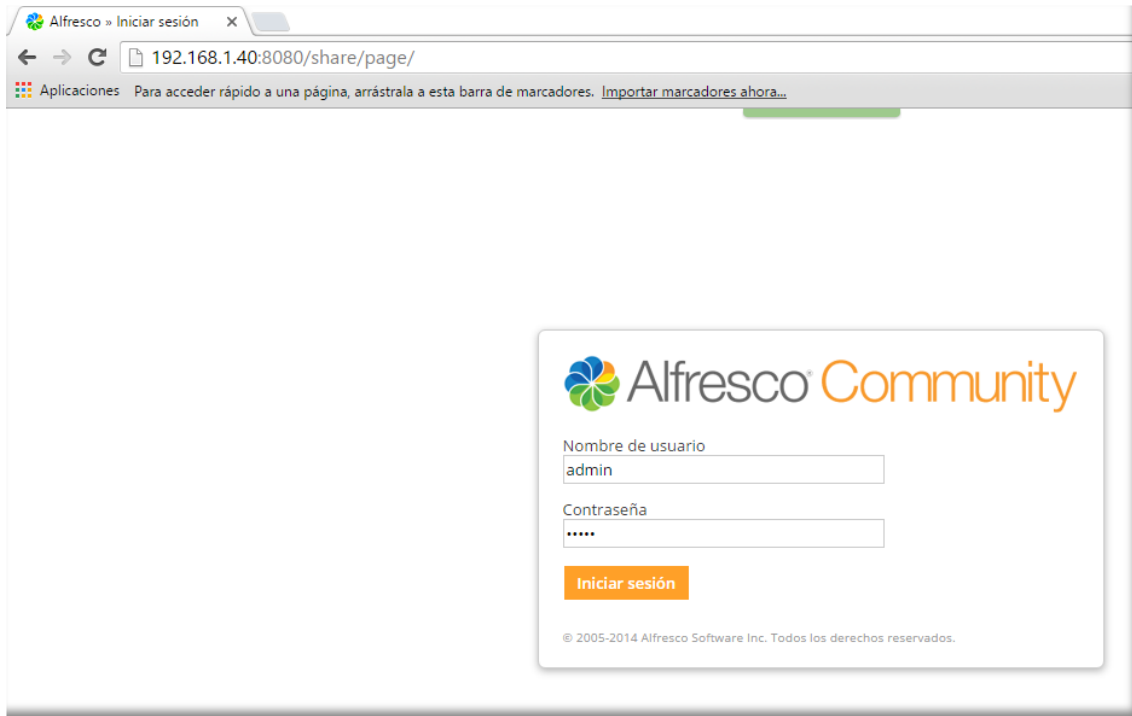
```
Using JRE_HOME:          /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:
/opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

```
root@alfresco2:/opt/tomcat/shared/classes# /etc/init.d/tomcat
restart
Using CATALINA_BASE:    /opt/tomcat
Using CATALINA_HOME:    /opt/tomcat
Using CATALINA_TMPDIR:  /opt/tomcat/temp
Using JRE_HOME:         /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:
/opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_BASE:    /opt/tomcat
Using CATALINA_HOME:    /opt/tomcat
Using CATALINA_TMPDIR:  /opt/tomcat/temp
Using JRE_HOME:         /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:
/opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

Accedemos al alfresco desde la siguiente url:

```
http://192.168.1.40:8080/share/
```

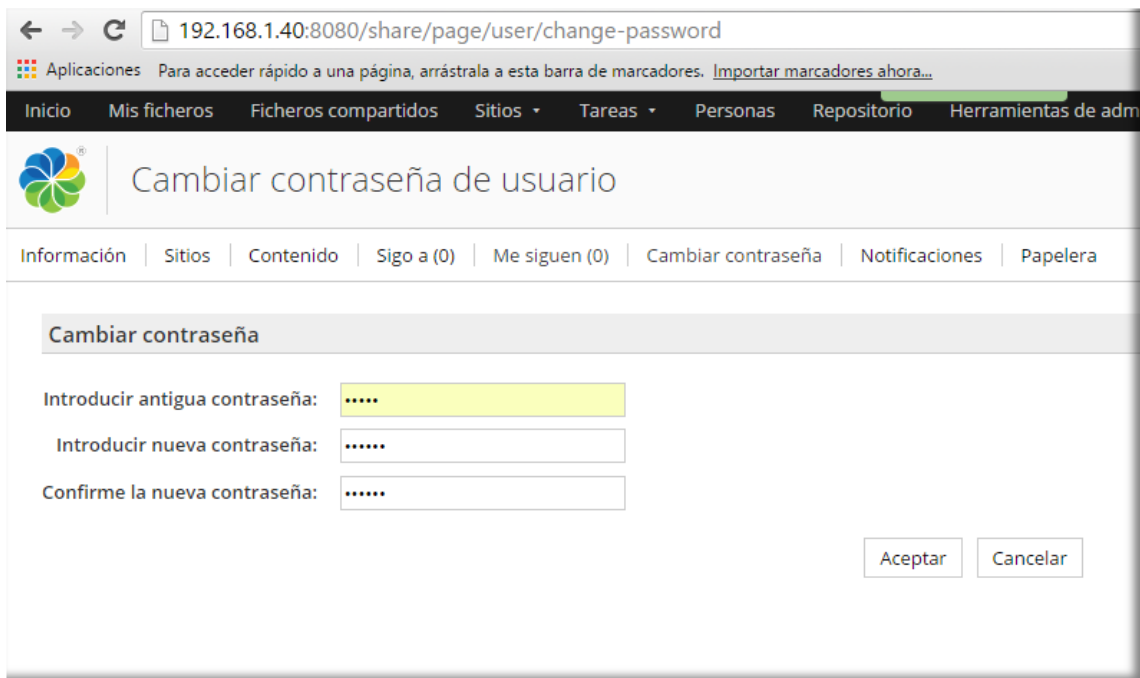
El usuario por defecto es "admin" y la contraseña "admin":*



Vamos a realizar dos pruebas:

La primera cambiamos la clave del usuario "admin" le ponemos por ejemplo "samuel" y probamos a entrar desde el nodo2:

Para ello nos vamos a las siguientes opciones "Su perfil > Editar su perfil > Cambiar contraseña" ponemos la antigua que es por defecto "admin" y cambiamos a "samuel":

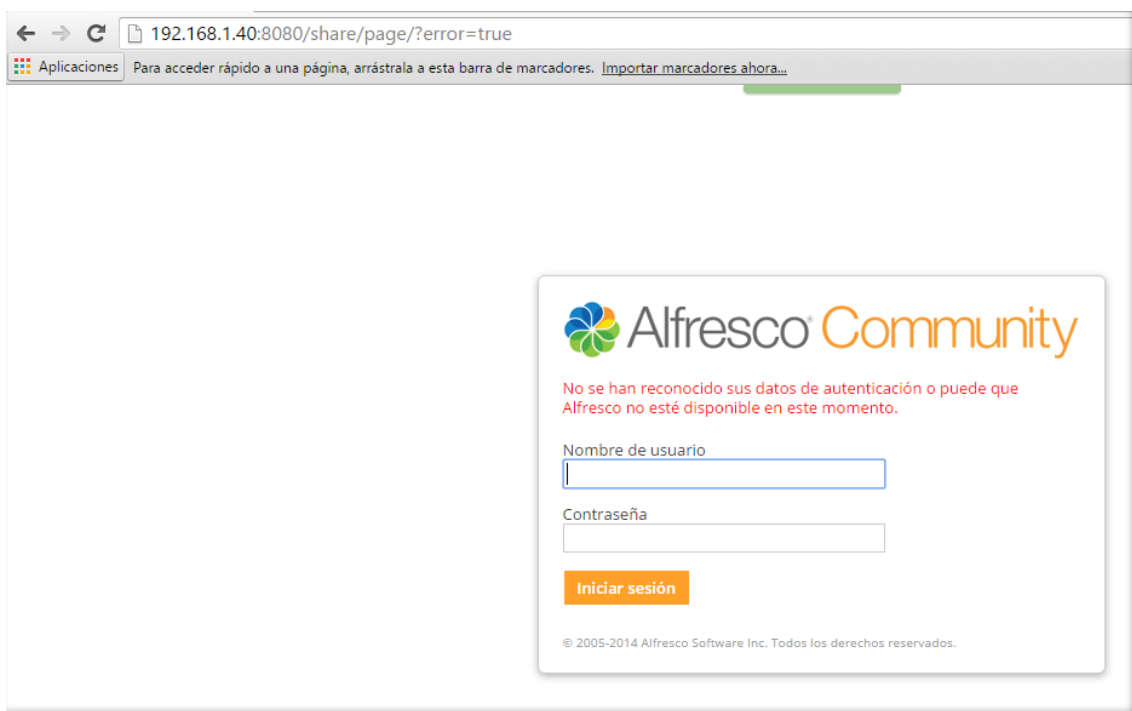


The screenshot shows a web browser window with the URL `192.168.1.40:8080/share/page/user/change-password`. The page title is "Cambiar contraseña de usuario". The navigation bar includes "Inicio", "Mis ficheros", "Ficheros compartidos", "Sitios", "Tareas", "Personas", "Repositorio", and "Herramientas de adm". The main content area is titled "Cambiar contraseña" and contains three input fields: "Introducir antigua contraseña:" (highlighted in yellow), "Introducir nueva contraseña:", and "Confirme la nueva contraseña:". All fields contain masked characters (dots). At the bottom right, there are two buttons: "Aceptar" and "Cancelar".

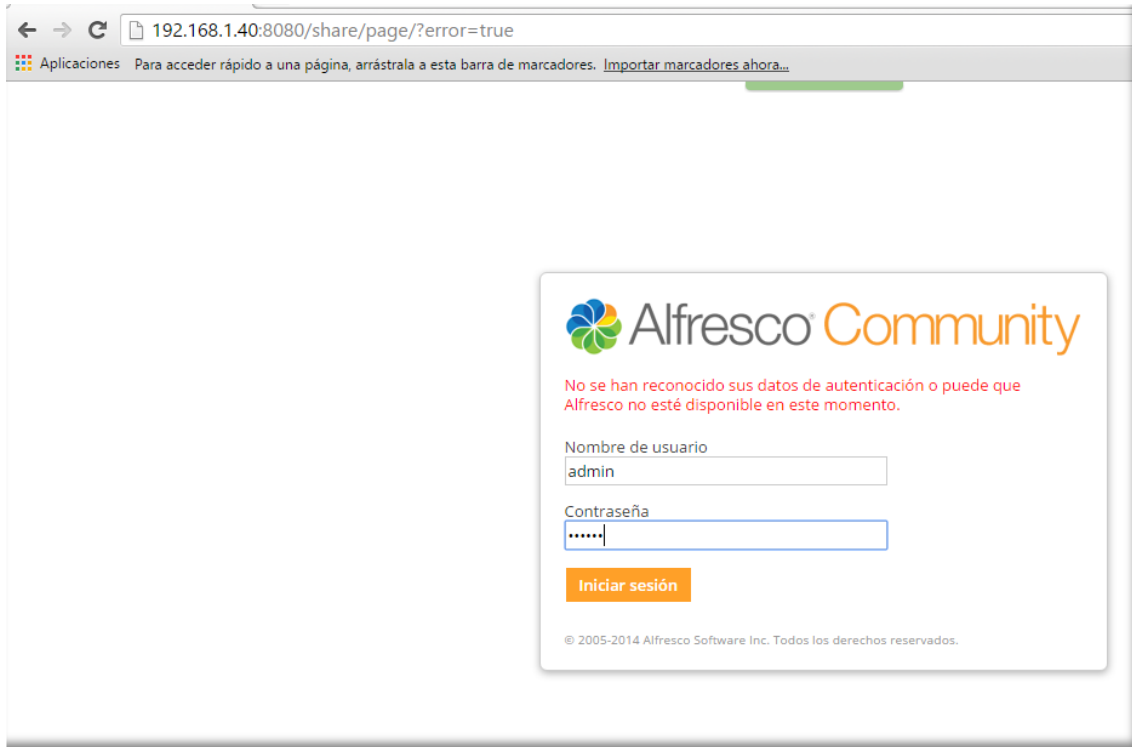
A continuación nos vamos a nodo 2 y probamos a entrar con usuario "admin" y contraseña "admin" :



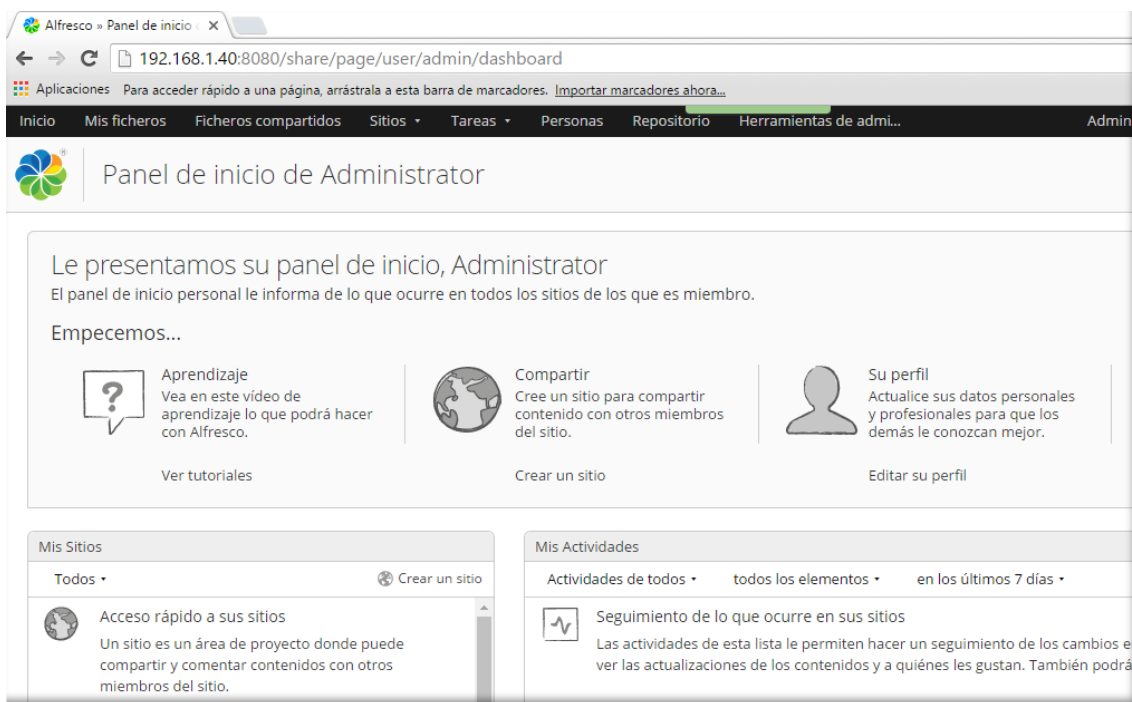
No nos deja entrar:

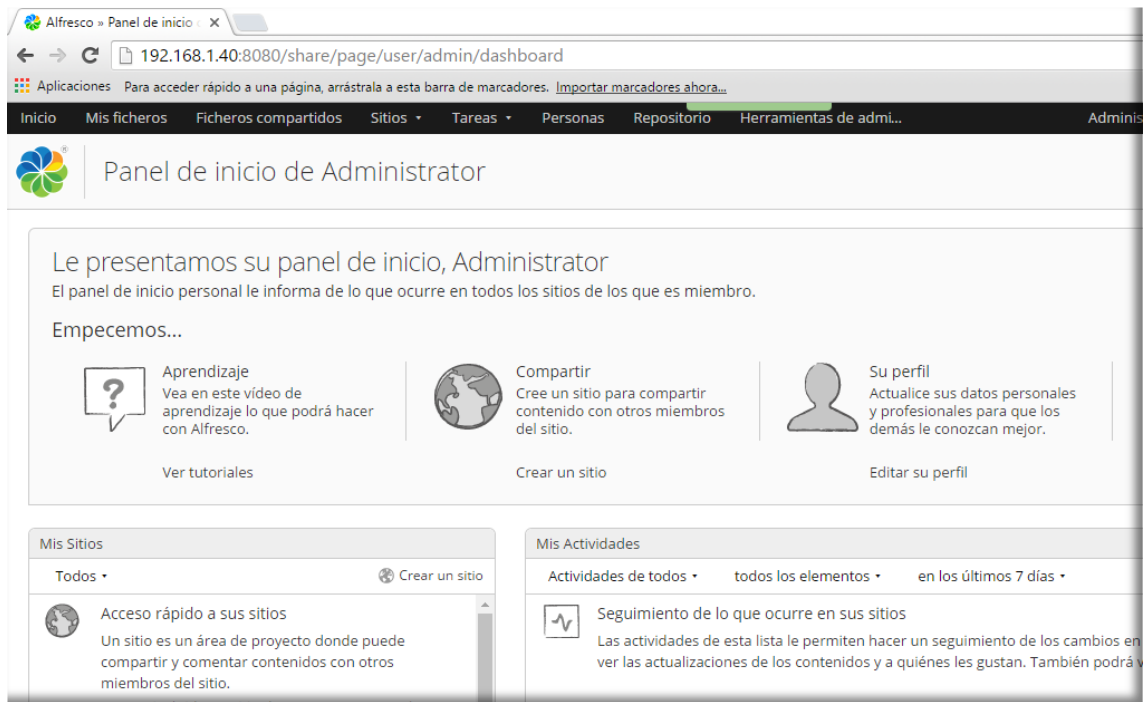


Probamos a entrar con usuario "admin" y contraseña "samuel":



Comprobamos que podemos entrar:





La segunda prueba desde nodo2 creamos el usuario "samuel" con contraseña "samuel", para ello desde panel de control accedemos a las siguientes opciones "Herramientas de administración > Usuarios > Nuevo usuario":

Nuevo usuario

Información

Nombre: *

Apellidos:

Correo electrónico: *

Acerca del usuario

Nombre de usuario: *

Contraseña: *

Verificar contraseña: *

Grupos:

Introduzca un término de búsqueda para encontrar grupos

Desde el nodo1 de msql, realizamos la consulta para comprobar que el usuario ha sido replicado:

```
root@mysql1:/home/usuario# mysql -uroot -proot
Warning: Using a password on the command line interface can be
insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 21
```

```
Server version: 5.6.30-76.3-56 Percona XtraDB Cluster (GPL),  
Release rel76.3, Re  
vision aa929cb, WSREP version 25.16, wsrep_25.16
```

Copyright (c) 2009-2016 Percona LLC and/or its affiliates

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> use alfresco;
```

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

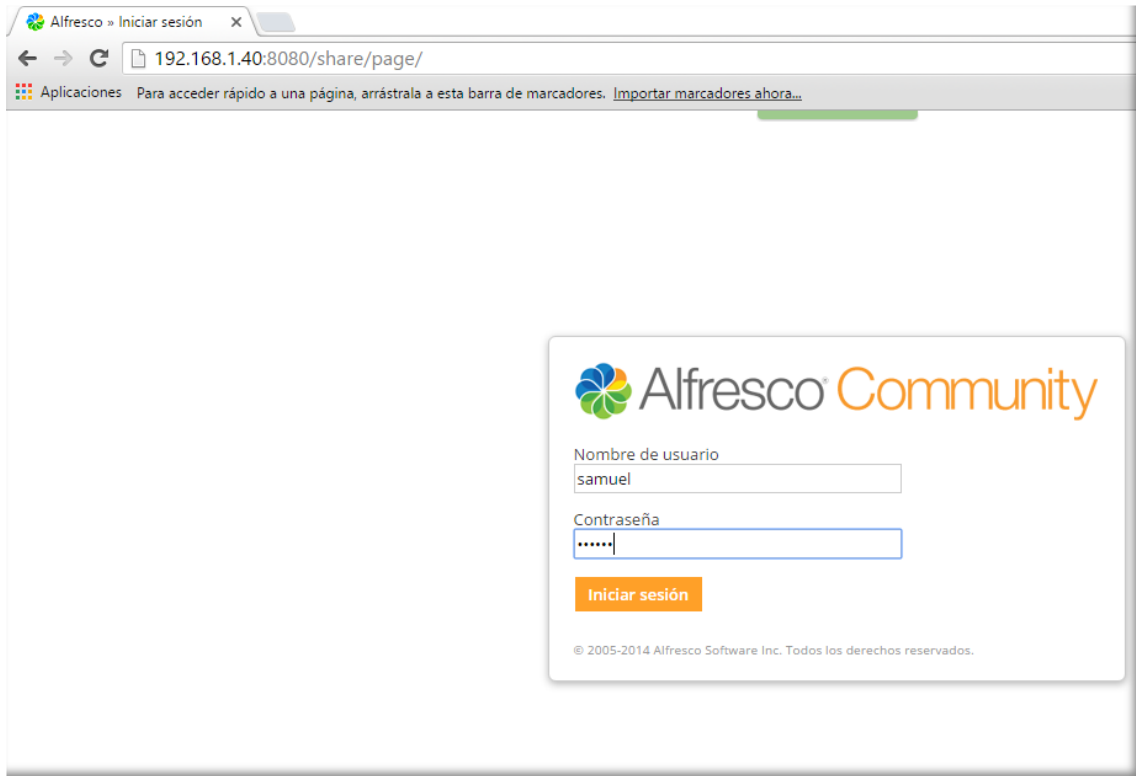
```
mysql> select * from alf_authority;
```

```
+-----+-----+-----+-----+
+
| id | version | authority | crc
+-----+-----+-----+-----+
+
| 1 | 0 | GROUP_EVERYONE | 1514782197
|
| 3 | 0 | guest | 2897713717
|
```



```
| 5 | 0 | admin | 2282622326  
|  
| 7 | 0 | ROLE_OWNER | 881792602  
|  
| 9 | 0 | GROUP_site_swsdp_SiteManager | 2553824491  
|  
| 11 | 0 | GROUP_site_swsdp_SiteCollaborator | 3822262144  
|  
| 13 | 0 | GROUP_site_swsdp_SiteContributor | 3082136708  
|  
| 15 | 0 | GROUP_site_swsdp_SiteConsumer | 4116454302  
|  
| 17 | 0 | abeecher | 2776041939  
|  
| 19 | 0 | mjackson | 4006557174  
|  
| 20 | 0 | samuel | 1608431557  
|  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
+  
11 rows in set (0.00 sec)
```

Por último probamos a entrar desde nodo1 con el usuario "samuel" y contraseña "samuel":



Comprobamos que efectivamente podemos entrar:



5. Configuración del balanceador de carga HAproxy:

En este apartado vamos a realizar la configuración de los balanceadores "balanceador1 y balanceador2":

Comenzamos instalando haproxy en ambos nodos:

```
root@balanceador1:/home/usuario# apt-get install haproxy
```

```
root@balanceador2:/home/usuario# apt-get install haproxy
```

Configuramos HAproxy en ambos balanceadores:

```
root@balanceador1:/home/vagrant# cd /etc/haproxy
```

```
root@balanceador1:/etc/haproxy# mv haproxy.cfg  
haproxy.cfg.original
```

```
root@balanceador1:/etc/haproxy# nano haproxy.cfg
```

```
global  
  
    daemon  
  
    maxconn 256  
  
    user haproxy  
  
    group haproxy  
  
    log 127.0.0.1 local0  
  
    log 127.0.0.1 local1 notice  
  
defaults  
  
    mode http  
  
    log global  
  
    timeout connect 5000ms
```

```
    timeout client 50000ms

    timeout server 50000ms

listen granja_cda

    bind 192.168.1.9:80

    mode http

    stats enable

    stats auth cda:cda

    balance roundrobin

    server uno 10.0.0.3:8080 maxconn 128

    server dos 10.0.0.4:8080 maxconn 128
```

```
root@balanceador2:/home/vagrant# cd /etc/haproxy
root@balanceador2:/etc/haproxy# mv haproxy.cfg
haproxy.cfg.original
root@balanceador2:/etc/haproxy# nano haproxy.cfg
```

```
global

    daemon

    maxconn 256

    user haproxy

    group haproxy

    log 127.0.0.1 local0

    log 127.0.0.1 local1 notice

defaults

    mode http

    log global

    timeout connect 5000ms

    timeout client 50000ms
```

```
timeout server 50000ms

listen granja_cda

bind 192.168.1.12:80

mode http

stats enable

stats auth cda:cda

balance roundrobin

server uno 10.0.0.3:8080 maxconn 128

server dos 10.0.0.4:8080 maxconn 128
```

Definimos la sección listen de un "proxy inverso" de nombre "granja_cda" que:

- Atendiendo peticiones en el puerto 80 del balanceador.
- Con balanceo round-robin.
- Que repartirá las peticiones entre dos servidores reales (de nombres uno y dos) en el puerto 80 de las direcciones 10.0.0.3 y 10.0.0.4.
- Adicionalmente, habilita la consola Web de estadísticas, accesible con las credenciales cda:cda.

Iniciamos HAproxy en balanceador pero antes de hacerlo es necesario habilitar el arranque de HAproxy desde los scripts de inicio en ambas manquinas:

```
root@balanceador1:/etc/haproxy# nano /etc/default/haproxy
```

Estableciendo la variable:

```
ENABLED=1
```

```
root@balanceador2:/etc/haproxy# nano /etc/default/haproxy
```

```
ENABLED=1
```

Reiniciamos el servicio:

```
root@balanceador1:/etc/haproxy# /etc/init.d/haproxy restart
[ ok ] Restarting haproxy (via systemctl): haproxy.service.
```

```
root@balanceador2:/etc/haproxy# /etc/init.d/haproxy restart
[ ok ] Restarting haproxy (via systemctl): haproxy.service.
```

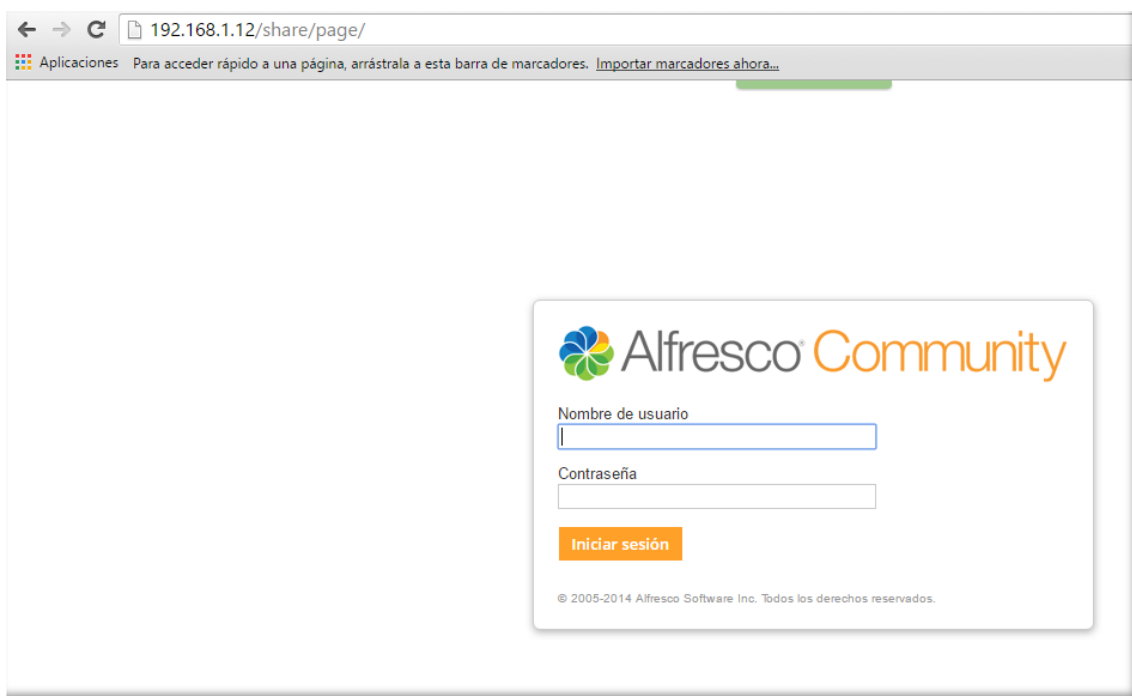
Comprobamos el funcionamiento apagando uno de los nodos:

```
root@alfresco2:/home/usuario# /etc/init.d/tomcat stop
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /usr/lib/jvm/java-7-openjdk-amd64
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
```

Realizamos el balanceo:



Error porqué ha intentado acceder a "alfresco2", volvemos a pulsar enter en el navegador:



6. Configuración DNS:

En este apartado, vamos a configurar el "dominio example.com" y un script para que cuando se nos caiga nuestro balanceador1, esté cambie al balanceador2.

Comenzamos instalando bind9 en nuestro nodo "dns":

```
root@dns:/home/usuario# apt-get install bind9
```

Ahora vamos a definir el nombre del fichero de la resolución directa. Para ello realizamos los siguientes pasos:

```
root@dns:/home/usuario# nano /etc/bind/named.conf.local
```

```
//  
  
// Do any local configuration here  
  
//  
  
// Consider adding the 1918 zones here, if they are not used in  
your  
  
// organization  
include "/etc/bind/zones.rfc1918";  
  
zone "example.com" {  
type master;  
file "db.example.com";  
};
```


A continuación creamos el fichero de resolución directa, con el siguiente contenido:

```
root@dns:/home/usuario# nano /var/cache/bind/db.example.com

<pre>
$TTL      604800
@         IN      SOA      dns.example.com.      root.example.com. (
                1          ; Serial
                                1200         ; Refresh
                                300          ; Retry
                                2419200      ; Expire
                                1200 )       ; Negative Cache TTL

@         IN      NS       dns.example.com.
@         IN      MX 10    correo.example.com

$ORIGIN   example.com.
dns       IN      A        192.168.1.40
SkullMan  IN      A        192.168.1.8
balanceador1  IN    A        192.168.1.9
balanceador2  IN    A        192.168.1.12
www       IN      CNAME    balanceador1
```

Reiniciamos el servicio dns:

```
root@dns:/home/usuario# /etc/init.d/bind9 restart
[ ok ] Restarting bind9 (via systemctl): bind9.service.
```

En el "/etc/hosts" de los nodos "balanceador1" y "balanceador2" añadimos los siguientes parámetros:

```
root@balanceador1:/home/usuario# nano /etc/hosts
```

```
127.0.0.1    localhost
127.0.1.1
10.0.0.254   dns
10.0.0.3     alfresco1
10.0.0.4     alfresco2
10.0.0.5     mysql1
10.0.0.6     mysql2
10.0.0.2     balanceador2
192.168.1.9  www.example.com balanceador1

# The following lines are desirable for IPv6 capable hosts

::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
root@balanceador2:/home/usuario# nano /etc/hosts
```

```
127.0.0.1    localhost
127.0.1.1
10.0.0.254   dns
10.0.0.3     alfresco1
10.0.0.4     alfresco2
10.0.0.5     mysql1
10.0.0.6     mysql2
```

```
10.0.0.1          balanceador1
192.168.1.12     balanceador2.www.example.com  balanceador2

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
```

A continuación vamos a crear el script, para comprobar si el nodo1 está caído o no:

```
root@dns:/home/usuario/scripts# nano script_dns.sh
```

```
#!/bin/bash

#Realizar un ping de 10 peticiones:

count=$( ping -c 10 10.0.0.1 | grep ttl | wc -l )

#Si los paquetes capturados son mayor o igual que 8 el servidor
funciona:

if [ $count -ge 8 ]
then
    echo " "
    echo "-----"
    echo "balanceador1"
    echo "-----"
    echo "El servidor "balanceador1" funciona correctamente."
    echo " "
```

```
    echo "Los paquetes recibidos son" $count "de 10"
    echo " "
else
    echo ""
    echo "El servidor "balanceador1" puede estar fallando..."
    echo "....."
    echo "Los paquetes recibidos son" $count "de 10."
    echo "....."
    echo "Activamos el nodo balanceador2."
    echo "....."

    #Eliminamos la última línea del fichero:

    sed -i '$d' /var/cache/bind/db.example.com

    sleep 5

    sed -i '$a www                IN          CNAME    balanceador2'
/var/cache/bind/db.example.com

    sleep 5

    /etc/init.d/bind9 restart
fi
</pre>
```

Ahora vamos a realizar la prueba apagando "balanceador1" y vamos a ejecutar el script:

```
root@balanceador1:/home/usuario# poweroff
Connection to 10.0.0.1 closed by remote host.
```

Ejecutamos el script:

```
root@dns:/home/usuario/scripts# sh script_dns.sh

El servidor balanceador1 puede estar fallando...
.....
Los paquetes recibidos son 0 de 10.
.....
Activamos el nodo balanceador2.
.....
[ ok ] Restarting bind9 (via systemctl): bind9.service.
```

Comprobamos que la entrada en la zona directa ha sido cambiada:

```
root@dns:/home/usuario/scripts# nano
/var/cache/bind/db.example.com
```

```
$TTL      604800
@         IN      SOA     dns.example.com.   root.example.com. (
                        1      ; Serial
                        1200    ; Refresh
                        300     ; Retry
                        2419200 ; Expire
                        1200 )  ; Negative Cache TTL
@         IN      NS     dns.example.com.
@         IN      MX     10   correo.example.com

$ORIGIN example.com.
```

dns	IN	A	192.168.1.40
SkullMan	IN	A	192.168.1.8
balanceador1	IN	A	192.168.1.9
balanceador2	IN	A	192.168.1.12
www	IN	CNAME	balanceador2

Lo siguiente que haremos, es ir al cliente "skullman", añadir el dns que vamos a utilizar y realizar la siguiente consulta:

```
root@SkullMan:/home/samuel# nano /etc/resolv.conf
```

```
# Generated by NetworkManager
search example.com
nameserver 192.168.1.40
```

```
root@SkullMan:/home/samuel# dig -t ns example.com

; <<>> DiG 9.9.5-9+deb8u6-Debian <<>> -t ns example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11711
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                IN      NS
```

```
;; ANSWER SECTION:
example.com.          604800  IN      NS      dns.example.com.

;; ADDITIONAL SECTION:
dns.example.com.     604800  IN      A       192.168.1.40

;; Query time: 3 msec
;; SERVER: 192.168.1.40#53(192.168.1.40)
;; WHEN: Fri Jun 17 20:48:12 CEST 2016
;; MSG SIZE  rcvd: 74
```

```
root@SkullMan:/home/samuel# dig www.example.com

; <<>> DiG 9.9.5-9+deb8u6-Debian <<>> www.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62238
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1,
ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.example.com.          IN      A

;; ANSWER SECTION:
www.example.com.          604800  IN      CNAME   balanceador2.example.com.
balanceador2.example.com. 604800  IN      A       192.168.1.12
```

```
;; AUTHORITY SECTION:
example.com.          604800  IN      NS      dns.example.com.

;; ADDITIONAL SECTION:
dns.example.com.     604800  IN      A       192.168.1.40

;; Query time: 3 msec
;; SERVER: 192.168.1.40#53(192.168.1.40)
;; WHEN: Fri Jun 17 20:48:46 CEST 2016
;; MSG SIZE rcvd: 121
```

Por último desde "skullman" accedemos con la siguiente url y nos debe hacer el balanceo de carga con "balanceador2":*

