

Alta disponibilidad (HA)



URL: <https://pandorafms.com/manual/!current/>
Permanent link: https://pandorafms.com/manual/!current/es/documentation/pandorafms/complex_environments_and_optimization/06_ha
4/06/10 14:36



Alta disponibilidad (HA)

Introducción

En entornos críticos y/o con mucha carga es posible que se necesite repartir la carga en varias máquinas y tener la seguridad de que si algún componente de Pandora FMS falla, el sistema se mantiene en línea.

Pandora FMS ha sido diseñado para que sea modular pero también está diseñado para trabajar en colaboración con otros componentes y ser capaz de asumir la carga de aquellos componentes que han caído.

Evidentemente, los agentes no son redundantes. La solución es redundar los sistemas críticos —independientemente de que tengan corriendo agentes de Pandora FMS o no— y así redundar la monitorización de dichos sistemas.

Se puede hablar de utilizar la Alta disponibilidad (High Availability o HA) en varios escenarios:

- Servidor de datos.
- Servidores de Red, WMI, Plugin, Web, Prediction, Recon, y similares.
- Base de datos (BBDD).
- Consola de Pandora FMS.

Alta disponibilidad del Servidor de Datos

- Para el servidor de datos de Pandora FMS se necesita montar dos máquinas con un Pandora FMS data server configurado (y diferente *hostname* y nombre del servidor).
- Se habrá de configurar un servidor Tentacle en cada uno de ellos.
- Cada máquina tendrá una dirección IP diferente.
- Si va a utilizar un balanceador externo, éste proveerá una única dirección IP a la que los agentes se conectarán para enviar sus datos.
- En el caso de usar el mecanismo de HA de los agentes, existirá un pequeño retardo en el envío de datos, ya que a cada ejecución del agente, este intentará conectar con el servidor primario, y si no contesta, lo hará contra el secundario (si se ha configurado así).

Si se quieren utilizar dos servidores de datos y que ambos manejen políticas, colecciones, y configuraciones remotas, habrá que compartir los directorios clave para que todas las instancias del data server puedan leer y escribir sobre dichos directorios. Las consolas también deberán tener acceso a dichos directorios compartidos.

- /var/spool/pandora/data_in/conf
- /var/spool/pandora/data_in/collections
- /var/spool/pandora/data_in/md5
- /var/spool/pandora/data_in/netflow
- /var/www/html/pandora_console/attachment

Alta disponibilidad en los Agentes Software

Desde los Agentes Software es posible realizar un balanceo de servidores de datos ya que es posible configurar un servidor de datos *master* (principal) y otro de *backup* (respaldo operativo). En el fichero de configuración del Agente Software `pandora_agent.conf` **se debe configurar y descomentar** la siguiente parte del archivo de configuración del agente:

- `secondary_mode`: Modo en el que debe estar el servidor secundario. Puede tener dos valores:
 - `on_error`: Envía datos al servidor secundario solo si no puede enviarlas al primario.
 - `always`: Siempre envía datos al servidor secundario, independientemente si puede contactar o no con el servidor principal.
- `secondary_server_ip`: Dirección IP del servidor secundario.
- `secondary_server_path`: Ruta donde se copian los XML en el servidor secundario, habitualmente `/var/spool/pandora/data_in`.
- `secondary_server_port`: Puerto por el que se copiarán los XML al servidor secundario, en Tentacle 41121, en SSH el puerto 22 y en FTP 21.
- `secondary_transfer_mode`: Modo de transferencia que se usará para copiar los XML al servidor secundario, Tentacle, SSH, FTP.
- `secondary_server_pwd`: Opción de contraseña para la transferencia por FTP.
- `secondary_server_ssl`: Se pondrá yes o no según se quiera usar SSL para transferir los datos por Tentacle.
- `secondary_server_opts`: En este campo se pondrán otras opciones necesarias para la transferencia.

Solamente está operativa la configuración remota del Agente Software, en el caso de estar habilitada, en el servidor principal.

Alta disponibilidad de los servidores de Red, WMI, plugin, web, prediction y similares

Necesita instalar múltiples **servidores**: servidor de red, servidor WMI, servidor de Plugin, servidor Web o prediction, en varias máquinas de la red (todas con la misma visibilidad hacia los sistemas que se quieran monitorizar) y que todas estén en el mismo segmento (para que los datos de latencia de la red sean coherentes).

Los servidores se pueden marcar como primarios. Esos servidores automáticamente recogerán los datos de todos los módulos asignados a un servidor que esté marcado como “caído”. Los propios servidores de Pandora FMS implementan un mecanismo para detectar que uno de ellos se ha caído a través de una verificación de su última fecha de contacto (`server_threshold x 2`). Basta que exista un solo servidor de Pandora FMS activo para que pueda detectar la caída del resto.

La forma evidente de implementar HA y un balanceo de carga, en un sistema de dos nodos es asignar el 50% de los módulos a cada servidor y marcar ambos servidores como principales

(*Master*). En el caso de haber más de dos servidores principales y un tercer servidor caído con módulos pendientes de ejecutar, el primero de los servidores principales que ejecute el módulo se “autoasigna” el módulo del servidor caído. En caso de recuperación de uno de los servidores caídos, se vuelven a asignar automáticamente los módulos que se habían asignado al servidor primario.

Configuración en los servidores PFMS

Un servidor de Pandora FMS puede estar corriendo en dos modos diferentes:

- Modo maestro (modo principal o modo MASTER).
- Modo no maestro.

Si un servidor se “cae” (está fuera de línea), sus módulos serán ejecutados por el servidor maestro de modo que no se pierdan datos.

En un momento dado solo puede haber un servidor maestro que se elige entre los servidores que tengan la opción de configuración *master* en `/etc/pandora/pandora_server.conf` con un valor mayor que 0:

```
master [1..7]
```

Si el servidor maestro actual se cae, se elige un nuevo maestro. Si hay más de un candidato, se elige aquel que tenga un valor mas alto en *master*.

Tenga cuidado al deshabilitar servidores. Si un servidor con módulos de red se cae y el Servidor de Red del servidor maestro está deshabilitado, esos módulos no se ejecutarán.

Dentro de `pandora_server.conf` se han introducido los siguientes parámetros:

- `ha_file`: Dirección del fichero binario temporal del HA.
- `ha_pid_file`: Proceso actual del HA.
- `pandora_service_cmd`: Control de estado del servicio de Pandora FMS.

Agregar servidores PFMS a un clúster HA DB

Si se cuenta con [Alta Disponibilidad en la Base de Datos](#) se necesitan algunas configuraciones extra para conectar más servidores Pandora FMS al *cluster* MySQL. En el archivo `pandora_server.conf` (ubicado por defecto en `/etc/pandora`) de cada uno de los servidores Pandora FMS independientes a agregar se deben configurar los siguientes parámetros:

- `dbuser`: Se debe tener el nombre de usuario de acceso al clúster MySQL. Por ejemplo:

```
dbuser pandora
```

- `dbpass`: Se debe contener la contraseña del usuario de acceso al clúster MySQL. Por ejemplo:

```
dbpass pandora
```

- `ha_hosts`: Se debe configurar el parámetro `ha_host` seguido de las direcciones IP o FQDN de los servidores MySQL que conforman el entorno HA. Ejemplo:

```
ha_hosts 192.168.80.170,192.168.80.172
```

Alta disponibilidad de la consola de Pandora FMS

Sólo hay que [instalar otra consola](#) apuntando a la base de datos. Cualquiera de las consolas podrá usarse de forma simultánea desde diferentes ubicaciones por diferentes usuarios. Se puede usar un [balanceador de carga web](#) delante de las consolas en caso de necesitar un crecimiento horizontal para el reparto de carga en la consola. El sistema de sesiones se gestiona mediante *cookies* y estas quedan almacenada en el navegador.

En el caso de estar utilizando configuración remota y para gestionarlo desde todas las consolas, tanto los servidores de datos como las consolas deben compartir el directorio de datos de entrada (por defecto: `/var/spool/pandora/data_in`) para la configuración remota de los agentes, las colecciones y otros directorios (véase el tema "[Arquitectura de seguridad](#)").

Es importante solo compartir los subdirectorios dentro de `data_in` y no el propio `data_in` ya que afectaría negativamente el rendimiento del servidor.

Actualización

A la hora de actualizar la consola de Pandora FMS en un entorno de Alta Disponibilidad es importante tener en cuenta las siguientes consideraciones al actualizar mediante OUM a través de Management → Warp update → [Update offline](#) . Se podrá descargar el paquete OUM desde la web de soporte de Pandora FMS.

Al estar en un entorno balanceado con una base de datos compartida, la actualización de la primera consola aplica los cambios correspondientes en la base de datos. Esto provoca que al actualizar la segunda consola, Pandora FMS muestre un error al encontrar la información ya insertada en la base de datos. No obstante, la actualización de la consola quedará realizada de igual forma.

Alta Disponibilidad en la Base de Datos

El objetivo de este apartado es ofrecer una solución completa para HA en entornos de Pandora FMS. Éste es el único modelo HA con soporte oficial para Pandora FMS, y se proporciona a partir de la versión 770. Este sistema reemplaza la configuración del cluster con Corosync y Pacemaker de versiones anteriores.

La nueva solución de HA de Pandora FMS está integrada en el producto (dentro del binario `pandora_ha`). Implementa un HA que soporta sedes aisladas geográficamente, con diferentes rangos de direcciones IP, cosa que con Corosync/Pacemaker no se puede realizar.

En el nuevo modelo de HA la configuración habitual es por parejas, por lo que el diseño no implementa sistema de *quorum* y simplifica la configuración y los recursos necesarios. De esta manera el sistema de monitorización funcionará siempre que exista un nodo de BB.DD. disponible y en caso de existir un **Split-Brain** de la BB.DD., el sistema funcionará en paralelo hasta fusionar de nuevo ambos nodos.

A partir de la versión 772 se cambió el HA para que fuera más sencillo y tenga menos fallos. Para éste nuevo HA es recomendable el uso de discos SSD para una mayor velocidad de escritura/lectura (IOPS), mínimo 500 Mb/s (o más, según el entorno). Se debe tener en cuenta también la *latencia* entre servidores ya que con *latencias* muy altas es complicado sincronizar ambas BB.DD. a tiempo.

Con la propuesta nueva se quiere solucionar los tres problemas actuales:

1. Complejidad y mantenibilidad del sistema actual (hasta la version NG 770).
2. Posibilidad de disponer de un entorno HA repartido en ubicaciones geográficas diferentes con un segmentación de red no local.
3. Procedimiento de recuperación de datos en caso de **Split-Brain** y funcionamiento asegurado del sistema en caso de ruptura de la comunicación entre ambos sitios separados geográficamente.

El nuevo sistema de HA para BBDD está implementado sobre Percona8, aunque en sucesivas versiones se detallará como hacerlo también en MySQL/MariaDB 8.

Pandora FMS se fundamenta en una base de datos MySQL para configurar y almacenar datos, por lo que un fallo en la base de datos puede paralizar temporalmente la herramienta de monitorización. El *cluster* de la base de datos de alta disponibilidad de Pandora FMS permite desplegar fácilmente una arquitectura robusta y tolerante a los fallos.

Se trata de una característica avanzada que requiere conocimientos en sistemas GNU/Linux. Es importante que todos los servidores tengan la

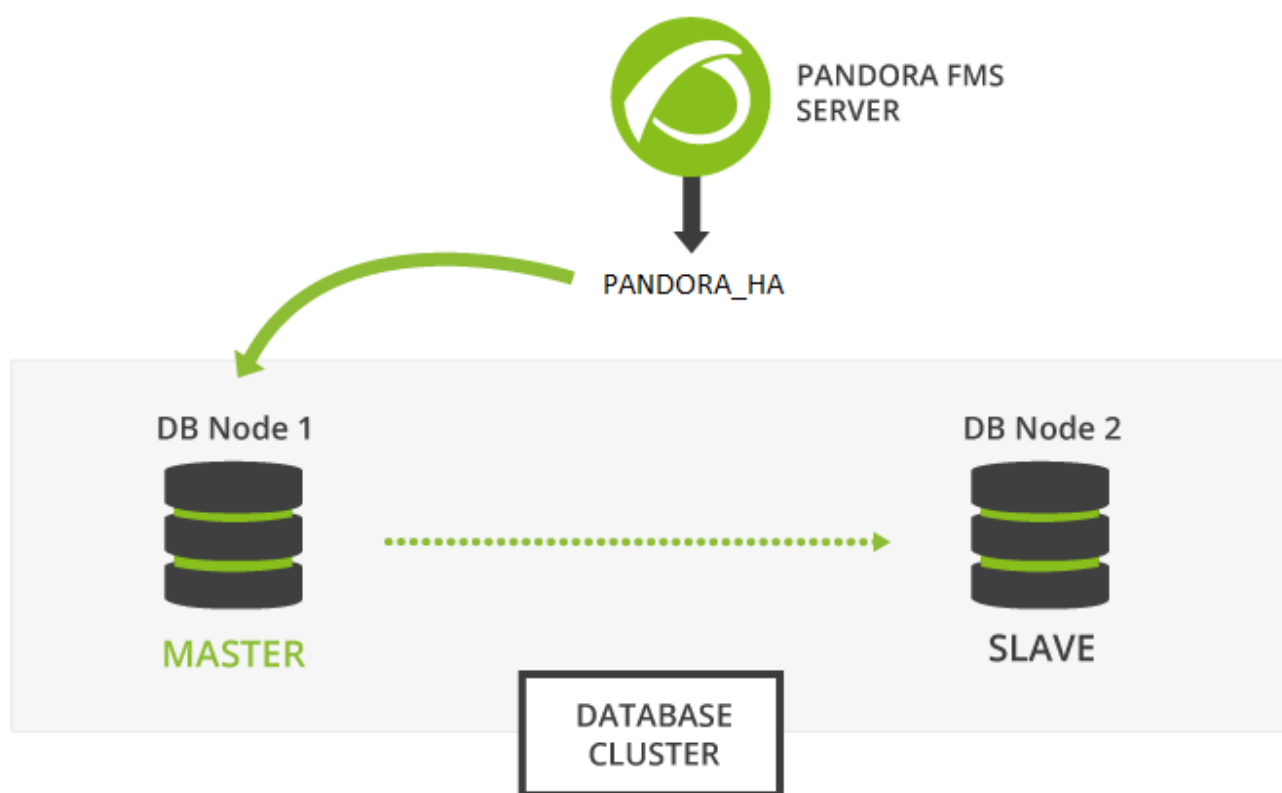
hora sincronizada con un servidor NTP (servicio chronyd en Rocky Linux 8).

Los nodos del *cluster* MySQL de replicación binaria se gestionan con el binario `pandora_ha`, a partir de la versión 770. Se eligió Percona como RDBMS por defecto por su escalabilidad, disponibilidad, seguridad y características de respaldo (*backup*).

La **replicación** activo/pasivo se desarrolla desde un nodo maestro único (con permiso de escritura) a cualquier número de secundarios (sólo lectura). Si falla el nodo maestro, uno de los secundarios asciende a maestro y `pandora_ha` se encarga de actualizar la dirección IP del nodo maestro.

El entorno quedará formado por los siguientes elementos:

- Servidores MySQL8 con replicación binaria activada (Activo - Pasivo).
- Servidor con `pandora_ha` con la configuración de todos los servidores de MySQL para llevar a cabo una monitorización continua y efectuar las promociones de esclavo-maestro y maestro-esclavo necesarias para el correcto funcionamiento del *clúster*.



Instalación de Percona 8

Versión 770 o posterior.

Instalación Percona 8 para RHEL 8 y Rocky Linux 8

Antes de nada es necesario tener instalado el repositorio de Percona en todos los nodos del entorno para poder realizar posteriormente la instalación de los paquetes del servidor de Percona.

Deberá abrir una ventana terminal con derechos de root o como usuario root. *Usted es el único responsable de dicha clave.* En las siguientes instrucciones se indica si debe ejecutar instrucciones en todos los dispositivos, en algunos o en uno en particular, preste atención a los enunciados.

Ejecutar en todos los dispositivos involucrados:

```
yum install -y https://repo.percona.com/yum/percona-release-latest.noarch.rpm
```

Active la versión 8 del repositorio de Percona en todos los dispositivos:

```
percona-release setup ps80
```

Instale el servidor de Percona junto a la herramienta de *backup* con el que se van a realizar los respaldos para la sincronización manual de ambos entornos. Ejecutar en todos los dispositivos involucrados:

```
yum install percona-server-server percona-xtrabackup-80
```

En el caso de que usted instale el servidor de Percona junto a la Consola web y el servidor PFMS, podrá usar el *deploy* indicándole la versión de MySQL 8 por medio del parámetro MYVER=80:

```
curl -Ls https://pfms.me/deploy-pandora-el8 | env MYVER=80  
bash
```

Instalación de Percona 8 en Ubuntu Server

Instale el repositorio de Percona en todos los dispositivos:

```
curl -O https://repo.percona.com/apt/percona-release_latest.generic_all.deb  
apt install -y gnupg2 lsb-release ./percona-release_latest.generic_all.deb
```

Active la versión 8 del repositorio de Percona en todos los dispositivos:

```
percona-release setup ps80
```

Instale el servidor de Percona junto a la herramienta de *backup* con el que se van a realizar los respaldos para la sincronización manual de ambos entornos. En todos los dispositivos ejecute:

```
apt install -y percona-server-server percona-xtrabackup-80
```

Configuración de la replicación binaria

Se recomienda almacenar los *binlogs* generados por la replicación en una partición adicional o disco externo su tamaño deberá ser el mismo que el que se haya reservado para la base de datos. Véase también los *token* `log-bin` y `log-bin-index`.

Una vez tenga instalado el servidor MySQL en todos los nodos del *clúster* proceda a realizar la configuración en ambos entornos para tenerlos replicados.

Primero de todo, debe configurar el fichero de configuración `my.cnf` preparándolo para que la replicación binaria funcione correctamente.

Nodo 1

Nodo 1 `/etc/my.cnf` (`/etc/mysql/my.cnf` en Ubuntu server):

```
[mysqld]
server_id=1 # It is important that it is different in all nodes.
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
# OPTIMIZATION FOR PANDORA FMS
innodb_buffer_pool_size = 4096M
innodb_lock_wait_timeout = 90
innodb_file_per_table
innodb_flush_method = O_DIRECT
innodb_log_file_size = 64M
innodb_log_buffer_size = 16M
thread_cache_size = 8
max_connections = 200
key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=128K
join_buffer_size=4M
```

```

sql_mode=""
# SPECIFIC PARAMETERS FOR BINARY REPLICATION
binlog-do-db=pandora
replicate-do-db=pandora
max_binlog_size = 100M
binlog-format=MIXED
binlog_expire_logs_seconds=172800 # 2 DAYS
#log-bin=/path # Uncomment for adding an additional storage
path for binlogs and setting a new path
#log-bin-index=/path/archive.index # Uncomment for adding an additional storage
path for binlogs and setting a new path
sync_source_info=1
sync_binlog=1
port=3306
report-port=3306
report-host=master
gtid-mode=off
enforce-gtid-consistency=off
master-info-repository=TABLE
relay-log-info-repository=TABLE
sync_relay_log = 0
replica_compressed_protocol = 1
replica_parallel_workers = 1
innodb_flush_log_at_trx_commit = 2
innodb_flush_log_at_timeout = 1800

[client]
user=root
password=pandora

```

- Los *token* a continuación del comentario OPTIMIZATION FOR PANDORA FMS realizan la configuración optimizada para Pandora FMS.
- Luego del comentario SPECIFIC PARAMETERS FOR BINARY REPLICATION se configuran los parámetros específicos para la replicación binaria.
- El *token* llamado `binlog_expire_logs_seconds` está configurado para un período de dos días.
- En la subsección `[client]` coloque el usuario y contraseña utilizada para la base de datos, por defecto al instalar PFMS son `root` y `pandora` respectivamente. *Estos valores son necesarios para la realización de los respaldos sin indicar usuario (automatizado).*

Es importante que el *token* `server_id` sea distinto en todos los nodos, en este ejemplo para el nodo 1 se utiliza dicho número.

Nodo 2

Nodo 2 `/etc/my.cnf` (`/etc/mysql/my.cnf` en Ubuntu server)

```

[mysqld]
server_id=2 # It is important that it is different in all nodes.
datadir=/var/lib/mysql

```

```
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
# OPTIMIZATION FOR PANDORA FMS
innodb_buffer_pool_size = 4096M
innodb_lock_wait_timeout = 90
innodb_file_per_table
innodb_flush_method = O_DIRECT
innodb_log_file_size = 64M
innodb_log_buffer_size = 16M
thread_cache_size = 8
max_connections = 200
key_buffer_size=4M
read_buffer_size=128K
read_rnd_buffer_size=128K
sort_buffer_size=128K
join_buffer_size=4M
sql_mode=""
# SPECIFIC PARAMETERS FOR BINARY REPLICATION
binlog-do-db=pandora
replicate-do-db=pandora
max_binlog_size = 100M
binlog-format=MIXED
binlog_expire_logs_seconds=172800 # 2 DAYS
#log-bin=/path # Uncomment for adding an additional storage
path for binlogs and setting a new path
#log-bin-index=/path/archive.index # Uncomment for adding an additional storage
path for binlogs and setting a new path
sync_source_info=1
sync_binlog=1
port=3306
report-port=3306
report-host=master
gtid-mode=off
enforce-gtid-consistency=off
master-info-repository=TABLE
relay-log-info-repository=TABLE
sync_relay_log = 0
replica_compressed_protocol = 1
replica_parallel_workers = 1
innodb_flush_log_at_trx_commit = 2
innodb_flush_log_at_timeout = 1800

[client]
user=root
password=pandora
```

- Los *token* a continuación del comentario OPTIMIZATION FOR PANDORA FMS realizan la configuración optimizada para Pandora FMS.
- Luego del comentario SPECIFIC PARAMETERS FOR BINARY REPLICATION se configuran los parámetros específicos para la replicación binaria.
- El *token* llamado binlog_expire_logs_seconds está configurado para un período de dos días.

- En la subsección [client] coloque el usuario y contraseña utilizada para la base de datos, por defecto al instalar PFMS son root y pandora respectivamente. *Estos valores son necesarios para la realización de los respaldos sin indicar usuario (automatizado).*

Es importante que el *token server_id* sea distinto en todos los nodos, en este ejemplo para el nodo 2 se utiliza dicho número correspondiente.

Configuración del nodo maestro

Una vez tenga la configuración correcta en ambos nodos, inicie la configuración del nodo *que va a tomar el rol de servidor maestro*.

1.- Inicie el servicio mysqld:

```
systemctl start mysqld
```

2.- Acceda con la contraseña temporal de root que se habrá generado en el *log*, el fichero `/var/log/mysqld.log`:

```
grep "temporary password" /var/log/mysqld.log
```

Con la *password* que aparezca acceda al servidor MySQL:

```
mysql -u root -p
```

Password: → Introducir la contraseña observada con el comando grep.

3.- Cambie la contraseña temporal por *pandora* del usuario root. Recuerde que el indicador `mysql >` (*prompt*) corresponde al intérprete de comandos de MySQL (MYSQL CLI):

```
mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Pandora4!';
```

```
mysql > UNINSTALL COMPONENT 'file://component_validate_password';
```

```
mysql > ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'pandora';
```

4.- Cree el usuario de replicación binaria y usuario root para conexiones remotas y administración del *clúster*:

```
mysql > CREATE USER slaveuser@%' IDENTIFIED WITH mysql_native_password BY 'pandora';
```

```
mysql > GRANT REPLICATION CLIENT, REPLICATION SLAVE on *.* to
```

```
slaveuser@'%';
```

```
mysql > CREATE USER root@%' IDENTIFIED WITH mysql_native_password BY  
'pandora';
```

```
mysql > GRANT ALL PRIVILEGES ON *.* to root@'%';
```

5.- Cree la base de datos de Pandora FMS:

```
mysql > create database pandora;
```

```
mysql > use pandora;
```

```
mysql > source /var/www/html/pandora_console/pandoradb.sql
```

```
mysql > source /var/www/html/pandora_console/pandoradb_data.sql
```

Para el comando source: Siempre que esté instalada la consola de Pandora FMS en el mismo servidor, *si no es así enviar ese fichero al servidor maestro.*

6.- Cree el usuario pandora y otorgue los privilegios de acceso a dicho usuario:

```
mysql > CREATE USER pandora@%' IDENTIFIED WITH mysql_native_password BY  
'pandora';
```

```
mysql > grant all privileges on pandora.* to pandora@'%';
```

En este momento ya tiene el servidor maestro preparado para empezar a replicar la base de datos de Pandora FMS.

Clonación de la base de datos

El siguiente paso consiste en hacer un clon de la base de datos maestra (MASTER) en el nodo esclavo (SLAVE). Para ello siga los siguientes pasos:

1.- Hacer un descarga (*dump*) completa de la base de datos MASTER:

```
MASTER #
```

```
xtrabackup --backup --target-dir=/root/pandoradb.bak/
```

```
MASTER #
```

```
xtrabackup --prepare --target-dir=/root/pandoradb.bak/
```

2.- Obtenga la posición del *log* binario del *backup*:

```
MASTER # cat /root/pandoradb.bak/xtrabackup_binlog_info
```

Devolverá algo parecido a lo siguiente:

```
binlog.000003 157
```

Tome nota de estos dos valores ya que son necesarios para el punto número 6.

3.- Haga una copia utilizando *rsync* con el servidor SLAVE para enviar el *backup* realizado:

```
MASTER # rsync -avpP -e ssh /root/pandoradb.bak/ node2:/var/lib/mysql/
```

4.- En el servidor SLAVE, configure los permisos para que el servidor MySQL pueda acceder sin problema a los ficheros enviados:

```
SLAVE # chown -R mysql:mysql /var/lib/mysql
```

```
SLAVE # chcon -R system_u:object_r:mysql_db_t:s0 /var/lib/mysql
```

5.- Inicie el servicio *mysqld* en el servidor SLAVE:

```
systemctl start mysqld
```

6.- Inicie el modo SLAVE en este servidor (utilice los datos anotados en el punto número 2):

```
SLAVE # mysql -u root -ppandora
```

```
SLAVE # mysql > reset slave all;
```

```
SLAVE # mysql >
```

```
CHANGE MASTER TO MASTER_HOST='nodo1',  
MASTER_USER='slaveuser',  
MASTER_PASSWORD='pandora',  
MASTER_LOG_FILE='binlog.000003',  
MASTER_LOG_POS=157;
```

```
SLAVE # mysql > start slave;
```

```
SLAVE # mysql > SET GLOBAL read_only=1;
```

Una vez haya terminado con todos estos pasos, si ejecuta el comando `show slave status` dentro de la *shell* de MySQL observará que el nodo se encuentra como *slave*. Si se ha configurado

correctamente se debe observar una salida como la siguiente:

```
***** 1. row *****
Slave_IO_State: Waiting for source to send event
Master_Host: node1
Master_User: root
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: binlog.000018
Read_Master_Log_Pos: 1135140
Relay_Log_File: relay-bin.000002
Relay_Log_Pos: 1135306
Relay_Master_Log_File: binlog.000018
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: pandora
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1135140
Relay_Log_Space: 1135519
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: fa99f1d6-b76a-11ed-9bc1-000c29cbc108
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
```



```

Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
Network_Namespace:
1 row in set, 1 warning (0,00 sec)

```

A partir de este momento se podría asegurar que tiene la replicación binaria habilitada y funcionando correctamente.

Configuración pandora_server

Versión 770 o posterior.

Es necesario configurar dentro del [fichero pandora_server.conf](#) una serie de parámetros necesarios para el correcto funcionamiento del pandora_ha.

Los parámetros que han de añadirse son los siguientes:

- `ha_hosts <IP_ADDRESS1>,<IP_ADDRESS2>` :

Debe configurar el parámetro `ha_host` seguido de las direcciones IP o FQDN de los servidores MySQL que conforman el entorno HA. La dirección IP que ponga en primer lugar tendrá preferencia para ser el servidor MASTER o al menos tiene el rol de maestro cuando se inicia por primera vez el entorno HA. Ejemplo:

```
ha_hosts 192.168.80.170,192.168.80.172
```

- `ha_dbuser` y `ha_dbpass` :

Son los parámetros donde se debe indicar el usuario y contraseña de usuario root o en su defecto un usuario de MySQL con los máximos privilegios que se encargará de realizar todas las operaciones de promoción *master - slave* en los nodos. Ejemplo:

```
ha_dbuser root
ha_dbpass pandora
```

- `repl_dbuser` y `repl_dbpass` :

Parámetros para definir el usuario de replicación que usará el SLAVE para conectarse al MASTER.

Ejemplo:

```
repl_dbuser slaveuser
repl_dbpass pandora
```

- `ha_sshuser` y `ha_sshport` :

Parámetros para definir el usuario/puerto con el que se conecta por ssh a los servidores Percona/MySQL para realizar las operaciones de recuperación. Para el correcto funcionamiento de esta opción, es necesario tener compartidas las claves ssh entre el usuario con el que se ejecuta el servicio `pandora_ha` y el usuario indicado en el parámetro `ha_sshuser`. Ejemplo:

```
ha_sshuser root
ha_sshport 22
```

- `ha_resyncPATH_SCRIPT_RESYNC` :

Por defecto el *script* para realizar la resincronización de los nodos, este se encuentra en:

```
/usr/share/pandora_server/util/pandora_ha_resync_slave.sh
```

En el caso de tener una instalación personalizada del *script*, indicar en este parámetro su ubicación para que se realice la sincronización automática o manual del nodo SLAVE cuando se necesite.

```
ha_resync /usr/share/pandora_server/util/pandora_ha_resync_slave.sh
```

- `ha_resync_log` :

Ruta del *log* donde se guardará toda la información relativa a las ejecuciones que realice el *script* de sincronización configurado en el anterior token. Ejemplo:

```
ha_resync_log /var/log/pandoraha_resync.log
```

- `ha_connect_retries` :

Número de intentos que realizará en cada comprobación con cada uno de los servidores del entorno HA antes de realizar cualquier cambio en el entorno. Ejemplo:

```
ha_connect_retries 2
```

Una vez configurados todos estos parámetros, podrá iniciar el servidor de Pandora FMS con el servicio `pandora_ha`. El servidor obtendrá una imagen del entorno y conocerá en ese momento quién es el servidor MASTER.

Cuando lo conozca, creará el fichero `pandora_ha_hosts.conf` en la carpeta `/var/spool/pandora/data_in/conf/`, donde se indicará en todo momento el servidor de Percona/MySQL que tiene el rol de MASTER.

En el caso de que el parámetro `incomingdir` del fichero `pandora_server.conf` contenga una ruta (PATH) distinta, este fichero se ubicará en ese PATH.

Este fichero se usará de intercambio con la Consola de Pandora FMS para que conozca en todo momento la dirección IP del servidor Percona/MySQL con rol MASTER.

- restart :

Se indicará con un valor de 0, ya que el *daemon* de `pandora_ha` es el encargado de reiniciar el servicio en caso de fallo y de esta forma se evitan posibles conflictos. Ejemplo:

```
# Pandora FMS will restart after restart_delay seconds on critical errors.
restart 0
```

Compartir claves SSH entre servidores

Se debe instalar y ejecutar un servidor OpenSSH en cada *host*. Suprima el mensaje de bienvenida o *banner* que muestra OpenSSH, ejecute en todos los dispositivos:

```
[ -f /etc/cron.hourly/motd_rebuild ] && rm -f /etc/cron.hourly/motd_rebuild
sed -i -e 's/^Banner.*//g' /etc/ssh/sshd_config
systemctl restart sshd
```

Comparta las claves SSH entre el `pandora_ha` y todos los servidores Percona/MySQL existentes en el entorno, ejecute en el servidor Pandora FMS:

```
printf "\n\n\n" | ssh-keygen -t rsa -P ''
ssh-copy-id -p22 root@node1
ssh-copy-id -p22 root@node2
```

- En el caso de que tenga la instalación sobre Ubuntu Server, usted debe habilitar el usuario `root` para realizar la conexión por SSH. Esto lo consigue generando una contraseña al usuario `root` al ejecutar la sentencia `sudo passwd root`.
- Habilite luego la conexión por SSH del usuario `root` al menos a través de claves compartidas "PermitRootLogin without-password" en el fichero de configuración del servicio `sshd`.

Uso del script de sincronización

Con el servidor de Pandora FMS se implementa un *script* que te permite sincronizar la base de datos SLAVE en el caso de que esté desincronizada.

La ejecución manual de este *script* es la siguiente:

```
./pandora_ha_resync_slave.sh "pandora_server.conf file" MASTER SLAVE
```

Por ejemplo, para hacer una sincronización manual del nodo 1 al nodo 2 la ejecución sería la siguiente:

```
/usr/share/pandora_server/util/pandora_ha_resync_slave.sh  
/etc/pandora/pandora_server.conf node1 node2
```

Para configurar la recuperación automática del entorno HA cuando haya algún problema de sincronización entre MASTER y SLAVE, es necesario tener el *token* de configuración `splitbrain_autofix` configurado a 1, dentro del fichero de configuración del servidor (`/etc/pandora/pandora_server.conf`).

Así, siempre que se produzca un **Split-Brain** (ambos servidores tengan el rol maestro) o haya algun problema de sincronización entre nodo MASTER y SLAVE, `pandora_ha` intentará lanzar el *script* `pandora_ha_resync_slave.sh` para sincronizar a partir de ese momento el estado del servidor MASTER en el servidor SLAVE.

Este proceso generará eventos en el sistema indicando el inicio, el final y si ha ocurrido algún error en el mismo.

Configuración de la Consola Pandora FMS

Versión 770 o posterior.

Se ha añadido un nuevo parámetro a la configuración del `config.php` indicando la ruta del directorio de intercambio que usa Pandora FMS por defecto `/var/spool/pandora/data_in`.

Si está configurado, buscará el fichero `/var/spool/pandora/data_in/conf/pandora_ha_hosts.conf` donde obtendrá la dirección IP para hacer la conexión.

```
$config["remote_config"] = "/var/spool/pandora/data_in";
```

En la consola de Pandora FMS podrá visualizar el estado del cluster accediendo a la vista de Manage HA.

```
https://PANDORA_IP/pandora_console/index.php?sec=gservers&sec2=enterprise/godmode/servers/HA_cluster
```

Servers / Manage Database HA

View nodes



IP	Node label	SQL Node status	SSH	Replication Status	DB Role	Delay	Last Update	SQL version	DB version	Actions
192.168.80.170		●	●	-	Master	-	2023-03-22 10:32:59	8.0.30-22	MR 60	
192.168.80.172		●	●	●	Slave	0	2023-03-22 10:32:59	8.0.30-22	MR 60	

Los datos de esta vista se están actualizando constantemente gracias a `pandora_ha`, no hay que realizar ningún procedimiento previo de configuración para poder visualizar esta sección siempre y cuando se encuentre configurado correctamente el `pandora_server.conf` con los parámetros citados en el anterior apartado.

Dentro de las acciones disponibles se puede configurar una etiqueta para cada uno de los nodos y se puede realizar la opción de sincronizar el nodo SLAVE mediante el icono .

Este icono puede tener los siguientes estados:

- Verde: Normal, ninguna operación por realizar.
- Azul: Pendiente por resincronización.
- Amarillo: Se está realizando resincronización.
- Rojo: *Error*, la resincronización ha fallado.

Migración de entornos HA Corosync-Pacemaker

La principal diferencia que existe entre un entorno HA usado en la versión 5 de MySQL/Percona Server con el modo HA actual es que ahora se usa `pandora_ha` para gestionar los nodos del *clúster* en detrimento de Corosync-Pacemaker, los cuales a partir de ahora dejarán de utilizarse.

La migración del entorno consistirá en:

- 1.- Actualizar Percona de la versión 5.7 a la versión 8.0: ["Instalación y actualización a MySQL 8"](#).
- 2.- Instalar `xtrabackup-80` en todos los dispositivos:

```
yum install percona-xtrabackup-80
```

Si utiliza Ubuntu server vea la sección ["Instalación Percona 8 para Ubuntu Server"](#).

- 3.- Crear todos los usuarios de nuevo con el *token* `mysql_native_password` en el nodo MASTER:

```
mysql > CREATE USER slaveuser@% IDENTIFIED WITH mysql_native_password BY 'pandora';
```

```
mysql > GRANT REPLICATION CLIENT, REPLICATION SLAVE on *.* to slaveuser@%;
```

```
mysql > CREATE USER pandora@% IDENTIFIED WITH mysql_native_password BY  
'pandora';
```

```
mysql > grant all privileges on pandora.* to pandora@%;
```

4.- Hacer el volcado (*dump*) de la base de datos desde el nodo MASTER al SLAVE:

4.1.- Hacer *dump* completo de la base de datos del MASTER:

MASTER #

```
xtrabackup --backup --target-dir=/root/pandoradb.bak/
```

MASTER #

```
xtrabackup --prepare --target-dir=/root/pandoradb.bak/
```

4.2.- Obtenga la posición del *log* binario del *backup*:

MASTER # cat /root/pandoradb.bak/xtrabackup_binlog_info

```
binlog.000003 157
```

Tome nota de estos dos valores ya que son necesarios en el punto 4.6.

4.3.- Haga una sincronización con *rsync* con el servidor SLAVE para enviar el *backup* realizado.

SLAVE # rm -rf /var/lib/mysql/*

MASTER # rsync -avpP -e ssh /root/pandoradb.bak/ node2:/var/lib/mysql/

4.4- En el servidor SLAVE, configure permisos para que el servidor MySQL pueda acceder sin problema a los ficheros enviados.

SLAVE # chown -R mysql:mysql /var/lib/mysql

SLAVE # chcon -R system_u:object_r:mysql_db_t:s0 /var/lib/mysql

4.5.- Inicie el servicio *mysqld* en el servidor SLAVE.

```
systemctl start mysqld
```

4.6.- Inicie el modo SLAVE en este servidor (utilice los datos del punto 4.2):

```
SLAVE # mysql -u root -ppandora
```

```
SLAVE # mysql > reset slave all;
```

```
SLAVE # mysql >
```

```
CHANGE MASTER TO MASTER_HOST='nodo1',  
MASTER_USER='slaveuser', MASTER_PASSWORD='pandora',  
MASTER_LOG_FILE='binlog.000003', MASTER_LOG_POS=157;
```

```
SLAVE # mysql > start slave;
```

```
SLAVE # mysql > SET GLOBAL read_only=1;
```

En el caso de que se desee instalar desde cero el entorno en un nuevo servidor, en el procedimiento de migración solamente debe instalarse desde cero tal y como indica el actual procedimiento en el nuevo entorno, y en el paso de crear la base de datos de Pandora FMS se debe importar los datos con un *backup* de la base de datos del antiguo entorno.

A su vez será necesario guardar en el nuevo entorno la configuración de Consola y Servidor de Pandora FMS indicadas en anteriores apartados.

Split-Brain

Por diversos factores, altas latencias, cortes de red, etcétera, se puede encontrar que ambos servidores MySQL han adquirido el rol maestro y no tenemos la opción *autoresync* activada en *pandora_ha* para que sea el propio servidor quien elija el servidor que va a actuar como maestro y realice la sincronización del nodo maestro al esclavo, perdiendo así toda la información que pudiese estar recopilándose en ese servidor.

Para solventar esta problemática se pueden fusionar (*merge*) los datos siguiendo este procedimiento.

Este procedimiento manual solamente cubre la recuperación de datos y eventos entre dos fechas. Asume que únicamente recupera datos de agentes/módulos que ya existen en el nodo en donde se va a realizar un *merging* de datos.

Si se crean agentes nuevos durante el tiempo de Split-Brain, o información nueva de configuración (alertas, políticas, etc.) estos no se tendrán en cuenta. Solamente se recuperarán datos y eventos. Es decir los datos relacionados con las **tablas** *tagente_datos*, *tagente_datos_string* y *tevento*.

Se ejecutarán los comandos siguientes en el nodo que quedó desconectado (el que va a ser promocionado a SLAVE), donde `yyyy-mm-dd hh:mm:ss` es la fecha y hora de inicio del Split-Brain y `yyyy2-mm2-dd2 hh2:mm2:ss2` su fecha y hora de finalización.

Ejecute el comando `mysqldump` con derechos de usuario apropiado para obtener una descarga de datos (*data dump* o simplemente *dump*)

```
mysqldump -u root -p -n -t --skip-create-options --databases pandora --tables tagente_datos --where='FROM_UNIXTIME(utimestamp)> "yyyy-mm-dd hh:mm:ss" AND FROM_UNIXTIME(utimestamp) <"yyyy2-mm2-dd2 hh2:mm2:ss2"'> tagente_datos.dump.sql
```

```
mysqldump -u root -p -n -t --skip-create-options --databases pandora --tables tagente_datos_string --where='FROM_UNIXTIME(utimestamp)> "yyyy-mm-dd hh:mm:ss" AND FROM_UNIXTIME(utimestamp) <"yyyy2-mm2-dd2 hh2:mm2:ss2"'> tagente_datos_string.dump.sql
```

```
mysqldump -u root -p -n -t --skip-create-options --databases pandora --tables tevento --where='FROM_UNIXTIME(utimestamp)> "yyyy-mm-dd hh:mm:ss" AND FROM_UNIXTIME(utimestamp) <"yyyy2-mm2-dd2 hh2:mm2:ss2"' | sed -e "s/([0-9]*,/(NULL,/gi"> tevento.dump.sql
```

Una vez obtenidos los *dumps* de esas tablas, se cargarán dichos datos en el nodo MASTER:

```
MASTER # cat tagente_datos.dump.sql | mysql -u root -p pandora
```

```
MASTER # cat tagente_datos_string.dump.sql | mysql -u root -p pandora
```

```
MASTER # cat tagente_evento.dump.sql | mysql -u root -p pandora
```

Tras cargar los datos que hA recuperado del nodo a promocionar a SLAVE, se procederá a sincronizarlo usando el siguiente procedimiento:

0.- Eliminar en el MASTER el directorio `/root/pandoradb.bak/`.

1.- Hacer *dump* completo de la DB del Maestro:

```
MASTER #
```

```
xtrabackup --backup --target-dir=/root/pandoradb.bak/
```

```
MASTER #
```

```
xtrabackup --prepare --target-dir=/root/pandoradb.bak/
```

2.- Obtenga la posición del *log* binario de los datos respaldados (*backup*):


```
MASTER # cat /root/pandoradb.bak/xtrabackup_binlog_info
```

Obtendrá algo parecido a lo siguiente (tome debida nota de estos valores):

```
binlog.000003 157
```

En este punto en SLAVE se debe limpiar el contenido de `/var/lib/mysql/` como se indica allá en el punto 4.3:

```
SLAVE # rm -rf /var/lib/mysql/*
```

3.- Haga una tarea con el comando `rsync` con el servidor *slave* para enviar así el *backup* realizado.

```
MASTER # rsync -avpP -e ssh /root/pandoradb.bak/ node2:/var/lib/mysql/
```

4.- En el servidor esclavo, configure permisos para que el servidor MySQL pueda acceder sin problema a los ficheros enviados.

```
SLAVE # chown -R mysql:mysql /var/lib/mysql
```

```
SLAVE # chcon -R system_u:object_r:mysql_db_t:s0 /var/lib/mysql
```

5.- Iniciamos servicio `mysqld` en el servidor esclavo.

```
systemctl start mysqld
```

6.- Inicie el modo esclavo en este servidor.

```
SLAVE # mysql -u root -ppandora
```

```
SLAVE # mysql > reset slave all;
```

```
SLAVE # mysql > CHANGE MASTER TO MASTER_HOST='nodo1',  
MASTER_USER='slaveuser', MASTER_PASSWORD='pandora',  
MASTER_LOG_FILE='binlog.000003', MASTER_LOG_POS=157;
```

```
SLAVE # mysql > start slave;
```

```
SLAVE # mysql > SET GLOBAL read_only=1;
```

Una vez terminados todos estos pasos, si ejecutamos el comando `show slave status` dentro de la *shell* de MySQL observará que el nodo se encuentra como modo *slave*. Si se ha configurado correctamente se debe observar una salida como la del siguiente ejemplo:

```
***** 1. row *****  
Slave_IO_State: Waiting for source to send event  
Master_Host: node1
```

```
Master_User: root
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: binlog.000018
Read_Master_Log_Pos: 1135140
Relay_Log_File: relay-bin.000002
Relay_Log_Pos: 1135306
Relay_Master_Log_File: binlog.000018
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Do_DB: pandora
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1135140
Relay_Log_Space: 1135519
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 1
Master_UUID: fa99f1d6-b76a-11ed-9bc1-000c29cbc108
Master_Info_File: mysql.slave_master_info
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Replica has read all relay log; waiting for more
updates
Master_Retry_Count: 86400
Master_Bind:
Last_IO_Error_Timestamp:
Last_SQL_Error_Timestamp:
Master_SSL_Crl:
Master_SSL_Crlpath:
Retrieved_Gtid_Set:
Executed_Gtid_Set:
```

```
Auto_Position: 0
Replicate_Rewrite_DB:
Channel_Name:
Master_TLS_Version:
Master_public_key_path:
Get_master_public_key: 0
Network_Namespace:
1 row in set, 1 warning (0,00 sec)
```

A partir de este momento se podría asegurar que se tiene la replicación binaria habilitada y funcionando correctamente de nuevo.

[Volver al Índice de Documentación Pandora FMS](#)