



Haute disponibilité (HA)



m:
<https://pandorafms.com/manual/!779/>
manent link:
https://pandorafms.com/manual/!779/fr/documentation/pandorafms/complex_environments_and_optimization/06_ha
5/01/22 19:13



Haute disponibilité (HA)

Présentation

Dans des environnements critiques et/ou avec beaucoup de charge, il peut être nécessaire de répartir la charge sur plusieurs machines et de s'assurer qu'en cas de défaillance d'un composant de Pandora FMS, le système reste en ligne.

Pandora FMS a été conçu pour être modulaire mais il est également conçu pour fonctionner en collaboration avec d'autres composants et pour pouvoir assumer la charge de ces composants qui ne fonctionnent pas.

Évidemment, les agents ne sont pas redondants. La solution consiste à redondance des systèmes critiques - qu'ils aient ou non des agents Pandora FMS en cours d'exécution - et donc à redondance de la supervision desdits systèmes.

Vous pouvez parler de l'utilisation de la haute disponibilité (High Availability ou HA) dans plusieurs scénarios:

- Serveur de données.
- Serveurs réseau, WMI, Plugin, Web, Prédiction, Recon, etc.
- Base de données (BBDD).
- Console Pandora FMS.

Haute disponibilité du serveur de données

- Pour le serveur de données Pandora FMS, il est nécessaire de monter deux machines avec un serveur de données Pandora FMS configuré (et un nom d'hôte et un nom de serveur différents).
- Un serveur Tentacle devra être configuré dans chacun d'eux.
- Chaque machine aura une adresse IP différente.
- Si vous allez utiliser un équilibreur externe, il fournira une adresse IP unique à laquelle les agents se connecteront pour envoyer leurs données.
- Dans le cas de l'utilisation du mécanisme HA des agents, il y aura un petit retard dans l'envoi des données, car à chaque exécution de l'agent, il essaiera de se connecter avec le serveur primaire, et s'il ne répond pas, il le fera contre le secondaire (s'il a été configuré de cette façon).

Si vous souhaitez utiliser deux serveurs de données et que les deux gèrent les stratégies, les collections et les configurations à distance, vous devrez partager les répertoires de clés afin que toutes les instances de serveur de données puissent lire et écrire dans ces répertoires. Les consoles doivent également avoir accès à ces répertoires partagés.

- /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
- /var/spool/pandora/data_in/discovery

Haute disponibilité dans les agents logiciels

A partir des Agents Logiciels, il est possible d'effectuer un équilibrage des serveurs de données puisqu'il est possible de configurer un serveur de données maître (principal) et un de sauvegarde (sauvegarde opérationnelle). Dans le fichier de configuration de l'agent logiciel `pandora_agent.conf` vous devez configurer et décommenter la partie suivante du fichier de configuration de l'agent:

- `secondary_server_ip`: L'adresse IP du serveur secondaire.
- `secondary_server_path`: Le chemin où le XML est copié sur le serveur secondaire, généralement `/var/spool/pandora/data_in`.
- `secondary_server_port`: Le port par lequel le XML sera copié sur le serveur secondaire, dans Tentacle 41121, dans le port SSH 22 et dans FTP 21.
- `secondaire_transfer_mode`: Le mode de transfert qui sera utilisé pour copier le XML vers le serveur secondaire, Tentacle, SSH, FTP.
- `secondary_server_pwd`: L'option de mot de passe pour le transfert FTP.
- `secondary_server_ssl`: Définissez `yes` ou `no` selon que vous souhaitez utiliser SSL pour transférer des données via Tentacle.
- `secondary_server_opts`: Les autres options nécessaires au transfert seront mises dans ce champ.
- `secondary_mode`: Le mode dans lequel le serveur secondaire doit être. Il peut avoir deux valeurs:
 1. `on_error`: Il envoie des données au serveur secondaire uniquement s'il ne peut pas les envoyer au serveur principal.
 2. `always`: Il envoie toujours des données au serveur secondaire, qu'il puisse ou non contacter le serveur principal.

Seule la configuration à distance de l'agent logiciel est opérationnelle, si elle est activée, sur le serveur principal.

Haute disponibilité des serveurs réseau, WMI, plugin, web, prédiction et autres

Vous devez installer plusieurs **serveurs**: Serveur réseau, serveur WMI, serveur de plug-in, serveur Web ou prédiction, sur différentes machines du réseau (toutes avec la même visibilité sur les systèmes que vous souhaitez surveiller) et qu'elles soient toutes dans le même segment (afin que

les données de latence du réseau soient cohérentes).

Les serveurs peuvent être marqués comme principal. Ces serveurs collecteront automatiquement les données de tous les modules affectés à un serveur marqué « down ». Les serveurs Pandora FMS implémentent eux-mêmes un mécanisme pour détecter que l'un d'entre eux est tombé grâce à une vérification de sa date de dernier contact (`server_threshold x 2`). Il suffit qu'il n'y ait qu'un seul serveur Pandora FMS actif pour qu'il puisse détecter la panne du reste.

La manière évidente d'implémenter la haute disponibilité et l'équilibrage de charge dans un système à deux nœuds consiste à attribuer 50 % des modules à chaque serveur et à marquer les deux serveurs comme maîtres. Dans le cas de plus de deux serveurs principaux et d'un troisième serveur en panne avec des modules en attente d'exécution, le premier des serveurs principaux qui exécute le module « s'auto-attribue » le module du serveur en panne. En cas de récupération d'un des serveurs en panne, les modules qui avaient été affectés au serveur primaire sont automatiquement réaffectés.

Configuration sur les serveurs

Un serveur Pandora FMS peut fonctionner dans deux modes différents:

- Mode maître (mode principal ou mode MASTER).
- Mode non maître.

Si un serveur « tombe en panne » (est hors ligne), ses modules seront exécutés par le serveur maître afin qu'aucune donnée ne soit perdue.

À tout moment, il ne peut y avoir qu'un seul serveur maître, choisi parmi les serveurs qui ont l'option de configuration maître dans `/etc/pandora/pandora_server.conf` avec une valeur supérieure à 0:

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

Si le serveur maître actuel tombe en panne, un nouveau maître est élu. S'il y a plus d'un candidat, celui avec la valeur maître la plus élevée est choisi.

Soyez prudent lors de la désactivation des serveurs. Si un serveur avec des modules réseau tombe en panne et que le serveur réseau du serveur maître est désactivé, ces modules ne fonctionneront pas.

Dans `pandora_server.conf`, les paramètres suivants ont été introduits:

- `ha_file`: L'adresse du fichier binaire temporaire HA.

- `ha_pid_file`: Le processus HA actuel.
- `pandora_service_cmd`: Le contrôle de l'état du service Pandora FMS.

Ajouter des serveurs PFMS à un cluster HA DB

Si vous disposez d'une [Haute disponibilité dans la base de données](#), certaines configurations supplémentaires sont nécessaires pour connecter davantage de serveurs Pandora FMS au cluster MySQL. Dans le fichier `pandora_server.conf` (situé par défaut dans `/etc/pandora`) pour chacun des serveurs Pandora FMS indépendants à ajouter, les paramètres suivants doivent être configurés:

- `dbuser`: Vous devez avoir le nom d'utilisateur pour accéder au cluster MySQL. Par exemple:

[/etc/pandora/pandora_server.conf](#)

```
dbuser pandora
```

- `dbpass`: Il doit contenir le mot de passe de l'utilisateur pour accéder au cluster MySQL. Par exemple:

[/etc/pandora/pandora_server.conf](#)

```
dbpass pandora
```

- `ha_hosts`: Le paramètre `ha_host` doit être configuré suivi des adresses IP ou FQDN des serveurs MySQL qui composent l'environnement HA. Exemple:

```
ha_hosts 192.168.80.170,192.168.80.172
```

Haute disponibilité de la console Pandora FMS

Juste [installer une autre console](#) pointant vers la base de données. Chacune des consoles peut être utilisée simultanément à partir de différents emplacements par différents utilisateurs. Un équilibreur de charge Web peut être utilisé devant les consoles si vous devez vous développer horizontalement pour le partage de charge sur la console. Le système de session est géré par des cookies et ceux-ci sont stockés dans le navigateur.

Dans le cas de l'utilisation de la configuration à distance et pour la gérer à partir de toutes les consoles, les serveurs de données et les consoles doivent partager le répertoire des données d'entrée (par défaut: `/var/spool/pandora/data_in`) pour la configuration à distance des agents, collections et autres répertoires (voir thème « [Architecture de sécurité](#) »).

Il est important de ne partager que les sous-répertoires à l'intérieur de `data_in` et non le `data_in` lui-même car cela affecterait négativement les performances du serveur.

Mise à jour

Lors de la mise à jour de la console Pandora FMS dans un environnement à haute disponibilité, il est important de prendre en compte les considérations suivantes lors de la **mise à jour via OUM** via Management → Warp update → Update offline. Peuvent télécharger le package OUM à partir du site Web d'assistance de Pandora FMS.

Étant dans un environnement équilibré avec une base de données partagée, la première mise à niveau de la console applique les modifications correspondantes à la base de données. Cela fait que lors de la mise à jour de la deuxième console, Pandora FMS affiche une erreur lors de la recherche des informations déjà insérées dans la base de données. Cependant, la mise à jour de la console s'effectuera de la même manière.

Haute disponibilité dans la base de données

L'objectif de cette section est d'offrir une solution complète pour la haute disponibilité dans les environnements Pandora FMS. Il s'agit du seul modèle HA avec prise en charge officielle de Pandora FMS, et il est fourni à partir de la version 770. Ce système remplace la configuration en cluster avec Corosync et Pacemaker des versions précédentes.

La nouvelle solution Pandora FMS HA est intégrée au produit (au sein du binaire `pandora_ha`). Elle implémente une HA qui prend en charge des sites géographiquement isolés, avec différentes plages d'IP, ce qui ne peut pas être fait avec Corosync/Pacemaker.

Dans le nouveau modèle HA, la configuration habituelle est par paires, de sorte que la conception ne met donc pas en œuvre un système de *quorum* et simplifie la configuration et les exigences en matière de ressources. De cette manière, le système de supervision fonctionnera chaque fois qu'un nœud de base de données sera disponible et s'il existe un **Split-Brain** de la base de données, le système fonctionnera en parallèle jusqu'à ce que les deux nœuds soient à nouveau fusionnés.

À partir de la version 772, l'HA a été modifié pour le rendre plus simple et moins bogué. Pour cette nouvelle HA, il est recommandé d'utiliser des disques SSD pour une vitesse d'écriture/lecture (IOPS) plus élevée, au minimum 500 Mb/s (ou plus, en fonction de l'environnement). La latence entre les serveurs doit également être prise en compte, car avec des latences très élevées, il est difficile de synchroniser les deux bases de données

à temps.

Avec la nouvelle proposition, nous voulons résoudre les trois problèmes actuels:

1. Complexité et maintenabilité du système actuel (jusqu'à la version NG 770).
2. Possibilité d'avoir un environnement HA distribué dans différentes zones géographiques avec une segmentation réseau non locale.
3. Procédure de récupération des données en cas de **Split-Brain** et garantie de fonctionnement du système en cas de rupture de communication entre les deux sites géographiquement séparés.

Le nouveau système HA pour les bases de données est implémenté sur Percona 8, bien que les versions successives détailleront également comment le faire sur MySQL/MariaDB 8.

Pandora FMS est basé sur une base de données MySQL pour configurer et stocker les données, donc une défaillance de la base de données peut temporairement paralyser l'outil de supervision. Le cluster de bases de données haute disponibilité Pandora FMS vous permet de déployer facilement une architecture robuste et tolérante aux pannes.

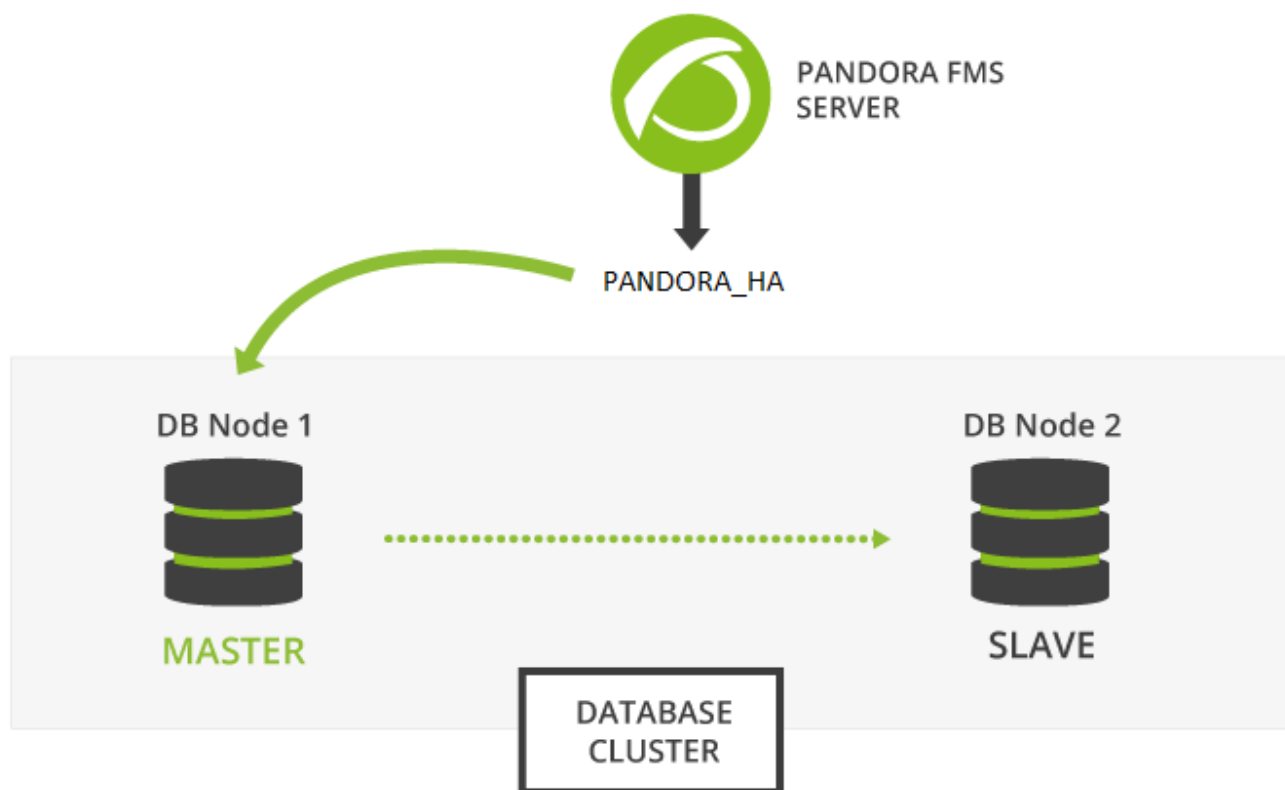
Il s'agit d'une fonctionnalité avancée qui nécessite une connaissance des systèmes GNU/Linux. Il est important que tous les serveurs aient l'heure synchronisée avec un serveur NTP (service chronyd dans Rocky Linux 8).

Les nœuds du cluster de réplication binaire MySQL sont gérés avec le binaire `pandora_ha`, à partir de la version 770. Percona a été choisi comme RDBMS par défaut pour ses fonctionnalités d'évolutivité, de disponibilité, de sécurité et de sauvegarde.

Actif/passif **replication** actif/passif se développe à partir d'un seul nœud maître (avec autorisation d'écriture) vers un nombre quelconque de nœuds secondaires (en lecture seule). Si le nœud maître tombe en panne, l'un des nœuds secondaires devient maître et `pandora_ha` se charge de mettre à jour l'adresse IP du nœud maître.

L'environnement sera composé des éléments suivants:

- Serveurs MySQL8 avec réplication binaire activée (Actif - Passif).
- Serveur avec `pandora_ha` avec la configuration de tous les serveurs MySQL pour effectuer une supervision continue et effectuer les promotions esclave-maître et maître-esclave nécessaires au bon fonctionnement du cluster.



Installation de Percona 8

Version 770 ou ultérieure.

Installation Percona 8 pour RHEL 8 et Rocky Linux 8

Tout d'abord, il est nécessaire d'avoir le référentiel Percona installé sur tous les nœuds de l'environnement afin d'installer ultérieurement les packages du serveur Percona.

Vous devrez ouvrir une fenêtre de terminal avec les droits root ou en tant qu'utilisateur root. Vous êtes seul responsable de ladite clé. Dans les instructions suivantes, il est indiqué si vous devez exécuter des instructions dans tous les appareils, dans certains ou dans un en particulier, faites attention aux instructions.

Exécutez sur tous les appareils concernés:

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

Activez la version 8 du référentiel Percona sur tous les appareils:

```
percona-release setup ps80
```

Installez le serveur Percona avec l'outil de sauvegarde avec lequel les sauvegardes seront effectuées pour la synchronisation manuelle des deux environnements. Exécutez sur tous les appareils concernés:

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

Dans le cas où vous installez le serveur Percona avec la console Web et le serveur PFMS, vous pouvez utiliser le `deploy` indiquant la version de MySQL 8 via le paramètre `MYVER=80`:

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

Installer Percona 8 sur Ubuntu Server

Installez le dépôt Percona sur tous les appareils:

```
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
```

Activez la version 8 du référentiel Percona sur tous les appareils:

```
percona-release setup ps80
```

Installez le serveur Percona avec l'outil de sauvegarde avec lequel les sauvegardes seront effectuées pour la synchronisation manuelle des deux environnements. Sur tous les appareils exécutez-les:

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

Configuration de la réplication binaire

Il est recommandé que stocker les *binlogs* générés par la réplication sur une partition supplémentaire ou un disque externe, leur taille doit être la même que celle réservée à la base de données. Voir aussi les jetons `log-bin` et `log-bin-index`.

Une fois que vous avez installé le serveur MySQL sur tous les nœuds du cluster, procédez à la configuration des deux environnements pour les répliquer.

Tout d'abord, vous devez configurer le fichier de configuration `my.cnf` en vue du bon fonctionnement de la réplication binaire.

Nœud 1

Noeud 1 `/etc/my.cnf` (`/etc/mysql/my.cnf` sur le serveur Ubuntu):

```
[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/mysql.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=ROW
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
```

- Les jetons après le commentaire OPTIMISATION POUR PANDORA FMS effectuent la configuration optimisée pour Pandora FMS.
- Après le commentaire PARAMETRES SPECIFIQUES A LA REPLICATION BINAIRE les paramètres spécifiques à la réplication binaire sont configurés.
- Le jeton nommé binlog_expire_logs_seconds est défini sur une période de deux jours.
- Dans la sous-section [client], placez le nom d'utilisateur et le mot de passe utilisés pour la base de données, par défaut lors de l'installation de PFMS, ils sont respectivement root et pandora. Ces valeurs sont nécessaires pour effectuer des sauvegardes sans indiquer l'utilisateur (automatisé).

Il est important que le jeton server_id soit différent dans tous les nœuds, dans cet exemple pour le nœud 1, ce numéro est utilisé.

Nœud 2

Noeud 2 /etc/my.cnf (/etc/mysql/my.cnf sur le serveur Ubuntu):

```
[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/mysql.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=ROW
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
```

- Les jetons après le commentaire OPTIMIZATION FOR PANDORA FMS effectuent la configuration optimisée pour Pandora FMS.
- Après le commentaire SPECIFIC PARAMETERS FOR BINARY REPLICATION les paramètres spécifiques à la réplication binaire sont configurés.
- Le jeton nommé `binlog_expire_logs_seconds` est défini sur une période de deux jours.
- Dans la sous-section `[client]`, placez le nom d'utilisateur et le mot de passe utilisés pour la base de données, par défaut lors de l'installation de PFMS, ils sont respectivement `root` et `pandora`. Ces valeurs sont nécessaires pour effectuer des sauvegardes sans indiquer l'utilisateur (automatisé).

Il est important que le jeton `server_id` soit différent dans tous les nœuds, dans cet exemple pour le nœud 2, le numéro correspondant est utilisé.

Configuration du nœud maître

Une fois que vous avez la bonne configuration sur les deux nœuds, lancez la configuration du nœud qui jouera le rôle de serveur maître.

1.- Démarrez le service `mysqld`:

```
systemctl start mysqld
```

2.- Accédez avec le mot de passe temporaire de root qui aura été généré dans le log, au fichier `/var/log/mysqld.log`:

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

Avec le mot de passe qui apparaît, accédez au serveur MySQL:

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

Password: → Saisir le mot de passe observé avec la commande grep.

3.- Changez le mot de passe temporaire dans pandora pour l'utilisateur root. Rappelez-vous que le `mysql>` (prompt) correspond à l'interpréteur de commandes MySQL (MYSQL CLI):

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Pandor4!';
```

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

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

4.- Créez l'utilisateur de réplication binaire et l'utilisateur root pour les connexions à distance et la gestion du cluster:

```
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.- Créez la base de données 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
```

Pour la commande source : Tant que la console Pandora FMS est installée sur le même

serveur, sinon, envoyez ce fichier au serveur maître.

6.- Créez l'utilisateur pandora et accordez les privilèges d'accès audit utilisateur:

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

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

À ce moment, vous avez déjà le serveur maître prêt à commencer à répliquer la base de données Pandora FMS.

Cloner la base de données

L'étape suivante consiste à créer un clone de la base de données maître (MASTER) sur le nœud esclave (SLAVE). Pour le faire, suivez ces étapes:

1.- Effectuez un téléchargement complet (dump) de la base de données MASTER :

MASTER #

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

MASTER #

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

2.- Obtenez la position du journal binaire de sauvegarde:

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

Il renverra quelque chose de similaire à ce qui suit:

```
binlog.000003 157
```

Prenez note de ces deux valeurs car elles sont nécessaires pour le point numéro 6.

3.- Faites une copie en utilisant rsync avec le serveur SLAVE pour envoyer la sauvegarde effectuée:

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

4.- Sur le serveur SLAVE, configurez les permissions pour que le serveur MySQL puisse accéder sans problème aux fichiers envoyés:

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

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

5.- Démarrez le service mysqld sur le serveur SLAVE :

```
systemctl start mysqld
```

6.- Démarrez le mode SLAVE sur ce serveur (utilisez les données notées au point numéro 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;
```

Une fois que vous avez terminé toutes ces étapes, si vous exécutez la commande `show slave status` dans le shell MySQL, vous verrez que le nœud est trouvé comme esclave. S'il a été configuré correctement, une sortie comme celle-ci doit être observée:

```
***** 1. row *****  
Slave_IO_State: Waiting for source to send event  
Master_Host: nodo1  
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)
```

À partir de ce moment, vous pouvez vous assurer que la réplication binaire est activée et fonctionne correctement.

Configuration pandora_server

Version 770 ou ultérieure.

Il est nécessaire de configurer au sein du [fichier pandora_server.conf](#) une série de paramètres nécessaires au bon fonctionnement de pandora_ha.

Les paramètres à ajouter sont les suivants:

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

Vous devez configurer le paramètre `ha_host` suivi des adresses IP ou FQDN des serveurs MySQL qui composent l'environnement HA. L'adresse IP que vous mettez en premier aura la priorité pour être le serveur MASTER ou au moins avoir le rôle de maître lors du premier démarrage de l'environnement HA. Par exemple:

```
ha_hosts 192.168.80.170,192.168.80.172
```

- `ha_dbuser` et `ha_dbpass` :

Ce sont les paramètres où doivent être indiqués le nom d'utilisateur et le mot de passe de l'utilisateur root ou, à défaut, un utilisateur MySQL avec les privilèges les plus élevés qui sera chargé d'effectuer toutes les opérations de promotion maître-esclave dans les nœuds. Par exemple:

```
ha_dbuser root
ha_dbpass pandora
```

- `repl_dbuser` et `repl_dbpass` :

Paramètres pour définir l'utilisateur de réplication que le SLAVE utilisera pour se connecter au MASTER. Par exemple:

```
repl_dbuser slaveuser
repl_dbpass pandora
```

- `ha_sshuser` et `ha_sshport` :

Paramètres pour définir l'utilisateur/port avec lequel se connecter par ssh aux serveurs Percona/MySQL pour effectuer des opérations de restauration. Pour que cette option fonctionne correctement, il est nécessaire que les clés ssh soient partagées entre l'utilisateur avec lequel le service pandora_ha est exécuté et l'utilisateur indiqué dans le paramètre `ha_sshuser`. Par exemple:

```
ha_sshuser root
ha_sshport 22
```

- `ha_resync PATH_SCRIPT_RESYNC:`

Par défaut, le script pour effectuer la resynchronisation des nœuds se trouve à:

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

Dans le cas d'une installation personnalisée du script, indiquez son emplacement dans ce paramètre afin que la synchronisation automatique ou manuelle du nœud SLAVE soit effectuée lorsque cela est nécessaire.

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

- `ha_resync_log :`

Chemin du journal où seront enregistrées toutes les informations relatives aux exécutions effectuées par le script de synchronisation configuré dans le jeton précédent. Par exemple:

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

- `ha_connect_retries :`

Le nombre de tentatives pour effectuer chaque vérification sur chacun des serveurs de l'environnement HA avant d'apporter des modifications à l'environnement. Par exemple:

```
ha_connect_retries 2
```

Une fois tous ces paramètres configurés, vous pourrez démarrer le serveur Pandora FMS avec le service `pandora_ha`. Le serveur obtiendra une image de l'environnement et saura alors qui est le serveur MASTER.

Lorsqu'il le saura, il créera le fichier `pandora_ha_hosts.conf` dans le dossier `/var/spool/pandora/data_in/conf/`, où le serveur Percona/MySQL qui a le rôle de MASTER sera indiqué à tout moment.

Dans le cas où le paramètre `incomingdir` du fichier `pandora_server.conf` contient un chemin différent (PATH), ce fichier sera situé dans ce PATH.

Ce fichier sera utilisé pour l'échange avec la console Pandora FMS afin qu'elle connaisse à tout moment l'adresse IP du serveur Percona/MySQL avec le rôle MASTER.

- `restart :`

Il sera indiqué avec une valeur de 0, car le démon `pandora_ha` est chargé de redémarrer le service en cas d'échec et de cette façon d'éventuels conflits sont évités. Par exemple:

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

Partage des clés SSH entre les serveurs

Un serveur OpenSSH doit être installé et en cours d'exécution sur chaque hôte. Supprimer le message ou la bannière de bienvenue affichés par OpenSSH, exécutez sur tous les appareils:

```
[ -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
```

Partagez les clés SSH entre `pandora_ha` et tous les serveurs Percona/MySQL existants dans l'environnement, exécutez sur le serveur Pandora FMS :

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

- Dans le cas où vous avez l'installation sur Ubuntu Server, vous devez activer l'utilisateur `root` pour établir la connexion via SSH. Pour ce faire, il génère un mot de passe pour l'utilisateur `root` en exécutant l'instruction `sudo passwd root`.
- Activez ensuite la connexion SSH de l'utilisateur `root` au moins via les clés partagées « `PermitRootLogin without-password` » dans le fichier de configuration du service `sshd`.

Utilisation du script de synchronisation

Avec le serveur Pandora FMS, un script est implémenté qui permet de synchroniser la base de données SLAVE en cas de désynchronisation.

L'exécution manuelle de ce script est la suivante:

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

Par exemple, pour effectuer une synchronisation manuelle du nœud 1 vers le nœud 2, l'exécution serait la suivante:

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

Pour configurer la récupération automatique de l'environnement HA lorsqu'il y a un problème de synchronisation entre MASTER et SLAVE, il est nécessaire d'avoir le jeton de configuration `splitbrain_autofix` défini sur 1, dans le fichier de configuration du serveur (`/etc/pandora/pandora_server.conf`).

Ainsi, chaque fois qu'un **Split-Brain** se produit (les deux serveurs ont le rôle de maître) ou qu'il y a un problème de synchronisation entre les nœuds MASTER et SLAVE, `pandora_ha` essaiera de lancer le script `pandora_ha_resync_slave.sh` pour synchroniser à partir de ce moment l'état du serveur MASTER dans le serveur SLAVE.

Ce processus générera des événements dans le système indiquant le début, la fin et si une erreur s'est produite.

Configuration de la console Pandora FMS

Version 770 ou ultérieure.

Un nouveau paramètre a été ajouté à la configuration `config.php` indiquant le chemin du répertoire d'échange que Pandora FMS utilise par défaut `/var/spool/pandora/data_in`.

S'il est configuré, il cherchera le fichier


`/var/spool/pandora/data_in/conf/pandora_ha_hosts.conf` où il obtiendra l'adresse IP pour établir la connexion.




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

Dans la console Pandora FMS, vous pouvez afficher l'état du cluster en accédant à la vue 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	 

Les données de cette vue sont constamment mises à jour grâce à `pandora_ha`, aucune procédure de configuration préalable n'est nécessaire pour pouvoir visualiser cette section tant que `pandora_server.conf` est correctement configuré avec les paramètres mentionnés dans la section précédente.

Parmi les actions disponibles, une étiquette peut être configurée pour chacun des nœuds et

l'option de synchronisation du nœud SLAVE peut être effectuée via l'icône   .

Cette icône peut avoir les états suivants:

- Vert: Normal, aucune opération à effectuer.
- Bleu: En attente en raison d'une resynchronisation.
- Jaune: La resynchronisation est en cours.
- Rouge: Erreur, la resynchronisation a échoué.

Migration des environnements HA Corosync-Pacemaker

La principale différence entre un environnement HA utilisé dans la version 5 de MySQL/Percona Server avec le mode HA actuel est que `pandora_ha` est désormais utilisé pour gérer les nœuds du cluster au lieu de Corosync-Pacemaker, qui ne sera plus utilisé à partir de maintenant.

La migration de l'environnement consistera en:

- 1.- Mettre à niveau Percona de la version 5.7 vers la version 8.0: « [Installation et mise à niveau vers MySQL 8](#) ».
- 2.- Installez `xtrabackup-80` sur tous les appareils:

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

Si vous utilisez le serveur Ubuntu, consultez la section « [Installation Percona 8 pour Ubuntu Server](#) ».

- 3.- Créez à nouveau tous les utilisateurs avec le token `mysql_native_password` dans le nœud 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.- Vider la base de données du nœud MASTER vers le SLAVE :

- 4.1.- Faites un dump complet de la base de données MASTER :

MASTER #

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

MASTER #

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

4.2.- Obtenir la position du journal binaire de sauvegarde:

MASTER # cat /root/pandoradb.bak/xtrabackup_binlog_info

```
binlog.000003 157
```

Prenez note de ces deux valeurs car elles sont nécessaires en 4.6.

4.3.- Effectuez une synchronisation avec rsync avec le serveur SLAVE pour envoyer la sauvegarde effectuée.

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

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

4.4- Sur le serveur SLAVE, définissez les autorisations afin que le serveur MySQL puisse accéder librement aux fichiers soumis.

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

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

4.5.- Démarrez le service mysqld sur le serveur SLAVE.

```
systemctl start mysqld
```

4.6.- Démarrer le mode SLAVE sur ce serveur (utiliser les données du point 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;

Dans le cas où vous souhaitez installer l'environnement à partir de zéro sur un nouveau serveur, dans la procédure de migration, vous devez uniquement l'installer à partir de zéro comme indiqué par la procédure actuelle dans le nouvel environnement, et à l'étape de création de la base de données Pandora FMS, vous devez importer les données avec une *backup* de l'ancienne base de données de l'environnement.

À son tour, il sera nécessaire de sauvegarder la configuration de la console et du serveur Pandora FMS indiquée dans les sections précédentes dans le nouvel environnement.

Split-Brain

En raison de plusieurs facteurs, latences élevées, pannes de réseau, etc., il se peut que les deux serveurs MySQL aient acquis le rôle de maître et que l'option `autoresync` ne soit pas activée dans `pandora_ha`, de sorte que le serveur lui-même choisisse le serveur qui va agir en tant que maître et effectue la synchronisation du nœud maître vers l'esclave, perdant ainsi toutes les informations qui pourraient être collectées dans ce serveur.

Pour résoudre ce problème, vous pouvez fusionner (*merge*) les données en suivant cette procédure.

Cette procédure manuelle couvre uniquement la récupération de données et d'événements entre deux dates. Elle suppose qu'elle ne récupère que les données des agents/modules qui existent déjà sur le nœud où la *merge* des données doit être effectuée.

Si de nouveaux agents sont créés pendant le temps de Split-Brain, ou de nouvelles informations de configuration (alertes, politiques, etc.) ils ne seront pas pris en compte. Seules les données et les événements seront récupérés. C'est-à-dire les données relatives aux `tables`. `tagente_datos`, `tagente_datos_string` et `tevento`.

Les commandes suivantes seront exécutées sur le nœud qui a été déconnecté (celui qui va être promu en SLAVE), où `yyyy-mm-dd hh:mm:ss` est la date et l'heure de début du Split-Brain y `yyyy2-mm2-dd2 hh2:mm2:ss2` sa date et son heure de fin.

Exécutez la commande `mysqldump` avec les droits d'utilisateur appropriés pour obtenir un vidage de données (*data dump* ou simplement *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
```

Une fois les *dumps* de ces tables obtenus, les données sont chargées dans le nœud 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
```

Après avoir téléchargé les données que vous avez récupérées du nœud à promouvoir vers SLAVE, vous procéderez à leur synchronisation en suivant la procédure suivante:

0.- Supprimer dans le MASTER le répertoire `/root/pandoradb.bak/`.

1.- Effectuer une *dump* complète de la base de données du Maître:

```
MASTER #
```

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

```
MASTER #
```

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

2.- Obtenir la position du binaire *log* des données sauvegardées (*backup*):

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

Vous obtiendrez quelque chose de similaire à ce qui suit (notez bien ces valeurs):

```
binlog.000003 157
```

À ce stade de SLAVE, le contenu de `/var/lib/mysql/` **comme indiqué dans 4.3** doit être effacé:

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

3.- Exécuter une tâche avec la commande `rsync` avec le serveur *slave* pour envoyer la *backup* effectuée.

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

4.- Sur le serveur esclave, configurez les autorisations de manière à ce que le serveur MySQL puisse accéder sans problème aux fichiers envoyés.

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

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

5.- Nous démarrons le service mysqld sur le serveur esclave.

```
systemctl start mysqld
```

6.- Démarrer le mode esclave sur ce serveur.

```
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;
```

Une fois toutes ces étapes terminées, si vous exécutez la commande `show slave status` dans la *shell* de MySQL, vous verrez que le nœud est en mode *slave*. S'il a été configuré correctement, vous devriez voir une sortie comme dans l'exemple suivant:

```
***** 1. row *****  
Slave_IO_State: Waiting for source to send event  
Master_Host: nodo1  
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)
```

À partir de ce moment, vous pouvez être sûr que la réplication binaire est activée et qu'elle fonctionne à nouveau correctement.

[Retour à l'index de la documentation du Pandora FMS](#)