

PANDORAFMS
E N T E R P R I S E

**Manual de Administración
para monitorizar PostgreSQL en Linux
Pandora Plugin psqlserver v1**



Manual de Administración para monitorizar PostgreSQL

© Artica Soluciones Tecnológicas 2005-2013

Indice

1	Changelog.....	3
2	Introducción.....	4
3	Requisitos.....	5
4	Matriz de compatibilidad.....	6
5	Clasificación de los módulos generados por el agente.....	7
5.1.	Módulos de sistemas	7
5.2.	Módulos definidos por el usuario.....	7
6	Instalación y configuración.....	8
6.1.	Configuración de PostgreSQL.....	8
6.2.	Instalación del plugin sobre el Agente Software de Pandora FMS.....	10
7	Parametrización del Plugin.....	11
7.1.	Configurando el acceso a Postgresql.....	11
7.1.1.	Credenciales.....	11
7.2.	Monitorización del Sistema.....	12
7.2.1.	Estado del servidor Postgres.....	12
7.2.2.	Uso de CPU de Postgres.....	12
7.2.3.	Uso de Memoria de Postgres.....	13
7.3.	Parámetros de rendimiento.....	13
8	Solución de problemas.....	15
8.1.	Revisa el documento pandora_agent.conf.....	15
8.2.	Reinicia el pandora_agent_daemon.....	15
8.3.	Revisa los permisos del plugin.....	15
8.4.	Valida la salida.....	15
8.5.	Valida el XML resultante.....	16
8.6.	Modo Debug.....	16
8.7.	Foro.....	16

1 CHANGELOG

Fecha	Autor	Cambio	Versión
20/04/2013	Carlos	Primera versión	v1r1

2 INTRODUCCIÓN

Este documento proporciona información sobre como usar el plugin de agente para monitorizar Postgres a través de Pandora FMS. Empezaremos por distinguir tres elementos esenciales del funcionamiento del mismo:

- **Utilización de software ya instalado en el sistema para la monitorización:** PostgreSQL, Agente Software de Pandora FMS, comandos del sistema, consultas definidas por el usuario en el archivo de configuración.
- **Archivo de configuración externo:** donde se definen los parámetros de configuración para la conexión al servidor de Postgres y donde el usuario define las consultas personalizadas que se ejecutarán contra la bases de datos según sus necesidades.
- **El propio código del plugin:** archivo que se ejecutará para la monitorización de la base de datos.



3 REQUISITOS

Los requisitos para que funcione correctamente la monitorización son:

- Tener instalado y configurado el Agente Software de Pandora.
- Tener la última versión de Perl Instalada.
- Servidor y cliente PostgreSQL instalado y con conexión.
- Nombre de la base de datos y dirección ip del equipo.
- Usuario con suficientes privilegios para acceder y monitorizar Postgres con su contraseña.

4 MATRIZ DE COMPATIBILIDAD

La matriz de compatibilidad del plugin es la siguiente:

Sistemas donde funciona	<ul style="list-style-type: none">• Ubuntu 12.04 con PostgreSQL• OpenSuse 11 con PostgreSQL• Debian 6.07 con PostgreSQL• Fedora 18 con PostgreSQL
Sistemas donde debería funcionar	<ul style="list-style-type: none">• Resto de sistemas Linux con PostgreSQL

5 CLASIFICACIÓN DE LOS MÓDULOS GENERADOS POR EL AGENTE

A modo general podemos diferenciar dos grupos de módulos que se crean durante la ejecución del plugin dentro del Agente software de Pandora FMS.

5.1. Módulos de sistemas

Se ejecutan automáticamente sin que el usuario intervenga directamente, es decir, los que siempre se ejecutarán mientras el plugin este instalado en el agente:

- Estado del servidor
- Uso de CPU de Postgres en la máquina donde está instalado
- Uso de memoria de Postgres en la máquina donde está instalado

5.2. Módulos definidos por el usuario

Estos son definidos en el archivo de configuración por el usuario administrador, el cuál deberá tener las credenciales necesarias para poder ejecutar las consultas SQL que más se adapten a sus necesidades.

6 INSTALACIÓN Y CONFIGURACIÓN

Primero se debe instalar las siguientes dependencias tecleando en un terminal los siguientes comandos:

- *zypper install perl perl-DBI perl-DBD-Pg* (Suse)
- *apt-get install perl libdbi-perl libdbd-perl* (Debian)
- *yum install perl perl-DBI perl-DBD-Pg* (Fedora)

6.1. Configuración de PostgreSQL

Para configurar el sistema correctamente de cara a la monitorización mediante el proceso colector de estadísticas de Pandora FMS tenemos que buscar y editar en el servidor de PostgreSQL los archivos de configuración siguientes:

- **postgresql.conf:** este es el fichero principal de configuración de Postgres y se deberá descomentar y/o editar las siguientes directivas:

```
track_activities = on
track_counts = on
track_functions = all
update_process_title = on
stats_temp_directory = 'pg_stat_tmp'
log_parser_stats = on
log_planner_stats = on
log_executor_stats = on
```


- **pg_hba.conf:** este archivo de configuración controla la autenticación y permisos de conexión de clientes con el servidor PostgreSQL. Su formato general es una línea por cada registro. Por lo tanto, hay que indicar la dirección del agente de Pandora FMS que ejecuta el plugin para que pueda realizar y recoger los datos. Una configuración de ejemplo podría ser la siguiente:

```
# Database administrative login by Unix domain socket
local  all                postgres                md5
# TYPE  DATABASE      USER          ADDRESS              METHOD
# "local" is for Unix domain socket connections only
local  all                all                md5
# IPv4 local connections:
host   all            all            127.0.0.1/32        md5
host   all            all            192.168.1.1/24      md5
host   all            all            0.0.0.0 0.0.0.0    md5
# IPv6 local connections:
host   all            all            ::1/128             md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local  replication    postgres                peer
#host   replication    postgres                127.0.0.1/32        md5
#host   replication    postgres                ::1/128             md5
```

Nota: una vez hechos los cambios es necesario reiniciar el servidor Posgres para que los cambios surtan efecto.

6.2. Instalación del plugin sobre el Agente Software de Pandora FMS

Copie el archivo ejecutable **Pandora_Plugin_psqlserver_v1.pl** (debemos asegurarnos que este archivos tenga permisos de ejecución) y el archivo de configuración **Plugin_psqlserver.conf** en la siguiente ruta:

```
/etc/pandora/plugins
```

Buscamos el archivo de configuración del Agente software de Pandora FMS que se encuentra en la siguiente ruta:

```
/etc/pandora/pandora_agent.conf
```

Agregamos una línea con la directiva **module_plugin** seguida de la ruta donde se encuentra el plugin, la ruta del archivo de configuración que se pasa como argumento y el archivo **psql_error.log** que recibirá los posibles errores de la ejecución del plugin y mantendrá limpia la salida estándar.

*Nota: podemos dejar sin indicar la ruta hacia el plugin si la colocamos en **/etc/pandora/plugins** ya que el Agente de Pandora FMS lo buscará en ese lugar por defecto, sin embargo se recomienda colocar siempre su ruta absoluta.*

```
module_plugin /etc/pandora/plugins/Pandora_Plugin_psqlserver_v1  
/etc/pandora/plugins/Plugin_psqlserver.conf 2> /etc/pandora/plugins/psql_error.log
```

Reiniciamos el Agente de Pandora FMS tecleando por terminal lo siguiente:

```
/etc/init.d/pandora_agent_daemon restart
```

7 PARAMETRIZACIÓN DEL PLUGIN

La configuración del plugin se hace a través del archivo de configuración *Plugin_psqserver.conf*

7.1. Configurando el acceso a Postgres

Para que el plugin pueda monitorizar la base de datos, debe proporcionar las credenciales de acceso y algunos datos generales:

7.1.1. Credenciales

Las credenciales de inicio de sesión se completarán de la siguiente manera:

```
#####
#                               DATA CONECTION DATABASE SECTION
#####
#
# Name of PostgreSQL database
dbname = postgres

# Ip address or single domain name may consist of letters, numbers and special
# characters like "." or "_" or "-"
host = localhost

# Default port of PostgreSQL
port = 5432

# Recommended user with administrator privileges
username = postgres

# Password to login
password = postgres
```

Donde:

- **dbname:** es el nombre de la base de datos a monitorizar
- **host:** es la dirección ip o nombre de dominio de la máquina donde está instalado Postgres
- **port:** el puerto donde escucha el servidor de Postgres (por defecto el 5432)
- **username:** el nombre del usuario con cierto privilegios para realizar las consultas necesarias para la monitorización
- **password:** la contraseña del usuario

7.2. Monitorización del Sistema

Por defecto el plugin generará tres chequeos básicos en cada intervalo de monitorización y que serán devueltos en forma de *módulos* al servidor de Pandora FMS los cuales ya están configurado internamente dentro del archivo ejecutable. Estos son:

7.2.1. Estado del servidor Postgres

Chequea mediante un valor *booleano* si el servidor está activo o no. Para ello el plugin ejecuta internamente el comando necesario para conocer el estado del proceso que es responsable de la ejecución de la base de datos. Al ser un valor *booleano* el servidor de Pandora discriminará automáticamente el estado de Postgres entre “*CRITICAL*” o “*NORMAL*” según sea el caso.

Nota: es posible modificar y/o adaptar este chequeo a un sistema Linux que no interprete adecuadamente los comandos que vienen establecidos por defecto en el código del plugin, para esto usted debe poseer ciertos conocimientos de Perl y de programación general.

7.2.2. Uso de CPU de Postgres

Chequea el uso de CPU de todos los procesos lanzados por el servidor, ya sea el principal o los siguientes:

- *postgres: logger process*
- *postgres: writer process*
- *postgres: wal writer process*
- *postgres: autovacuum launcher process*
- *postgres: stats collector process*

Para este módulo (como en el caso del módulo de *estado del servidor*) no es necesario definir manualmente desde la consola de administración de Pandora FMS los valores de los distintos estados (posibles) del módulo ya que estos vienen predefinidos internamente en el archivo ejecutable del plugin como se muestra a continuación en la siguiente tabla:

<i>Uso de CPU de Postgres</i>	<i>Estado</i>
Entre 0 y 49	NORMAL
Entre 50 y 75	WARNING
Entre 76 y 100	CRITICAL

Nota: es posible modificar y/o adaptar este modulo a un sistema Linux que no interprete adecuadamente los comandos que vienen establecidos por defecto en el código del plugin,

adicionalmente, puede cambiar los umbrales de los estados mostrados en la tabla anterior, para esto usted debe poseer ciertos conocimientos de Perl y de programación general.

7.2.3. Uso de Memoria de Postgres

Chequea el uso de memoria de todos los procesos lanzados por el servidor, ya sea el principal o los siguientes:

- *postgres: logger process*
- *postgres: writer process*
- *postgres: wal writer process*
- *postgres: autovacuum launcher process*
- *postgres: stats collector process*

Nota: es posible modificar y/o adaptar este chequeo a un sistema Linux que no interprete adecuadamente los comandos que vienen establecidos por defecto en el código del plugin, para esto usted debe poseer ciertos conocimientos de Perl y de programación general.

7.3. Monitorización mediante SQL

7.3.1. Crear módulos predefinidos

El administrador podrá crear cuantos módulos necesite según sus necesidades de tal forma que el plugin los ejecute de manera secuencial. Estos deberán configurarse desde el archivo de configuración utilizando para ello el siguiente formato:

```

usermod
name = "user module"
description = "user module"
query := "SELECT count(*) from pg_stat_user_tables;"
type = "generic"
end
    
```

Donde:

- **usermod:** es la etiqueta que define el módulo personalizado
- **name:** el nombre del módulo y el que se mostrará en la consola de Pandora FMS
- **description:** descripción del módulo
- **query:** consulta SQL que definirá el administrador
- **type:** el tipo de dato que será enviado
- **end:** cierre de la etiqueta del módulo
-

En el caso de el campo *query* hay que tener especial cuidado que las consultas devuelvan un dato atómico ya que si estas retornan mas de un valor (por ejemplo un registro o incluso varios) el Servidor de Pandora no podrá interpretarlos.

Si el campo *type* no se especifica o está comentado se interpreta por defecto como “generic”. A continuación se muestran los distintos tipos de datos que maneja el plugin:

- **"generic" = generic_data** : dato de tipo numérico, por ejemplo el número de usuarios conectados a la base de datos
- **"incremental" = generic_data_inc** : este tipo de dato al aumentar constantemente (a delta) cogerá al menos dos valores para calcular la diferencia entre ellos
- **"string" = genric_data_string** : dato pensado para devolver cadenas de texto, por ejemplo la versión del servidor de Postgres
- **"boolean" = generic_proc** : este tipo de dato devuelve 0 si es falso y un valor distinto de 0 si es verdadero, usado con frecuencia para saber si algo se cumple o no, por ejemplo saber si el servidor Postgres está activo

7.3.2. Desactivar módulos

Para desactivar un módulo basta con comentar la línea “usermod” correspondiente al como se muestra a continuación:

```
#usermod
name = "user module"
description = "user module"
query := "SELECT count(*) from pg_stat_user_tables;"
type = "generic"
end
```

7.3.3. Módulos predefinidos

De serie el plugin viene con varios módulos preconfigurados para monitorizar aspectos generales de Postgres (algunos deberá activarlos):

- Número de Roles en la base de datos
- Número de conexiones activas a la base de datos
- Versión de Postgres
- Espacio en disco de todas las bases de datos
- Procesos activos
- Número de bases de datos
- Número de *commits* (debe activarlo)
- Número de *rollbacks* (debe activarlo)

- Número de registros consultados (debe activarlo)
- Número de registros buscados (debe activarlo)
- Número de registros insertados (debe activarlo)
- Número de registros actualizados (debe activarlo)
- Número de registros borrados (debe activarlo)

8 SOLUCIÓN DE PROBLEMAS

Si Pandora FMS no reconoce tu plugin de agente, no consigues la información esperada o el agente simplemente no quiere funcionar, aquí hay varias cosas que deberías tener en cuenta:

8.1. Revisa el documento *pandora_agent.conf*

El Agente de Software requiere de una línea en este archivo con la **ruta correcta** del plugin.

Ejemplo:

```
module_plugin /etc/pandora/plugins/MyMonitor.pl /etc/pandora/plugins/MyMonitor.conf
2> /etc/pandora/plugins/MyMonitor.err
```

MyMonitor.pl es el plugin de agente, **MyMonitor.conf** es el archivo de configuración pasado como argumento, y **MyMonitor.err** es un archivo que recibirá posibles errores de la ejecución del plugin y mantendrá limpia la salida estándar.

8.2. Reinicia el *pandora_agent_daemon*

Si tienes la versión básica de Pandora FMS (no Enterprise), el Agente de Software ejecutará los plugins cada 5 minutos. Para aquellas personas que no pueden esperar, es posible reiniciar el Agente de Software desde la línea de comandos.

Ejemplo:

```
/etc/init.d/pandora_agent_daemon restart
```

8.3. Revisa los permisos del plugin

El plugin y los archivos que va a usar, **deben tener los permisos correctos** de lectura, escritura y ejecución. En UNIX, esto debería bastar:

```
chmod 755 <ruta_plugin>
```

8.4. Valida la salida

Una forma fácil de encontrar los errores, es **ejecutar el plugin manualmente** en línea de comandos. Siéntate y **revisa la salida** cuidadosamente.

Por ejemplo:

```
popeye:/etc/pandora/plugins # ./pandora_df
```



```
<module>
<name><![CDATA[/dev/sda2]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[19]]></data>
<description>% of usage in this volume</description>
</module>
<module>
<name><![CDATA[udev]]></name>
<type><![CDATA[generic_data]]></type>
<data><![CDATA[1]]></data>
<description>% of usage in this volume</description>
</module>
```

8.5. Valida el XML resultante

El XML que imprime el plugin **debe tener una sintaxis de XML válida**. El XML, además, debe estar **bien formado**. Para comprobar si lo está, puedes seguir estos dos pasos desde la línea de comandos:

1. Crea un documento XML con la salida del plugin: `./Plugin.pl > Plugin.xml`
2. Comprueba el documento XML: `xmllint Plugin.xml`

8.6. Modo *Debug*

Puedes activar el modo de desarrollo cambiando el valor de la etiqueta *debug* en el archivo *pandora_agent.conf* de 0 a 1. Una vez hayas hecho esto, cuando el Agente de Software ejecute el plugin, los resultados se guardarán en un documento XML con toda la información del agente.

El nombre del documento será el nombre del agente con la extensión *.data*, y estará localizado en el directorio */tmp* (echa un vistazo al log del agente de pandora en */var/log/pandora/pandora_agent.log*). Revisando el documento, podrás ver si los datos de tu plugin están siendo recogidos y si son como esperas.



Cuando habilitas el modo *Debug*, el agente se ejecuta solo una vez y posteriormente sale

8.7. Foro

Si después de todo, el error persiste, siéntete libre de preguntar en los [foros](#) oficiales de Pandora FMS.