

9 Cosas Que Se Deben Supervisar En MySQL

Los servicios que se deben de **supervisar en MySQL** no es cosa del otro mundo, es necesario pero no complicado. **MySQL** ofrece una lista completa de variables a **monitorizar el estado de su servidor y el rendimiento**. El sistema se guiará a través de las variables cruciales que deben someterse a control. Vamos a suponer que usted tiene uno o más servidores MySQL, que se hayan configurado y funcionando bien. Aquí están **9 cosas que debes supervisar en MySQL**.

1. Disponibilidad MySQL

Sí, esta es la primera cosa que usted debe buscar!. No tendría sentido supervisar su MySQL si no es aún disponible. El tiempo de inactividad de MySQL simplemente no es aceptable. Al mismo tiempo garantizar cero tiempo de inactividad no garantiza el máximo rendimiento.

Puede ejecutar

```
[sql]-mysqladmin -h 192.168.1.95 -u root -p status[/sql]
```

Para saber si MySQL se está ejecutando ó simplemente puede hacer ping a MySQL utilizando el servicio de estado

```
[sql]service mysqld status[/sql]
```

Si esta usando algún Sistema Operativo Linux.

2. Presencia de los usuarios y la seguridad de la base de datos

¿Alguno de sus usuarios de MySQL tiene '%' como anfitrión?

Lo que significa que el usuario puede conectarse a MySQL desde cualquier lugar del mundo, Si es así, los servidores son propensos a los ataques. Para mayor seguridad, sería mejor dar valores literales de host en lugar de '%'.
Si, por ejemplo, su anfitrión es localhost entonces el hacker tendrá que acceder a localhost antes de toparse con los servidores.

Tiene usuarios de MySQL con privilegios excesivos?

como MySQL, por defecto viene con un usuario que tenga privilegios de root, por desgracia, la palabra root es demasiada conocida pero afortunadamente no tiene ningún significado en MySQL en absoluto. Usted puede simplemente dar otro nombre en lugar del root.

```
[sql]
CREATE USER 'obsure_name'@'%' IDENTIFIED BY 'password';
GRANT ALL ON *.* TO 'obsure_name'@'%' WITH GRANT OPTION;
DROP USER 'root'@'%;
FLUSH PRIVILEGES;
[/sql]
```

Además, asegúrese de que todos los usuarios tengan contraseñas de MySQL.

La instalación por defecto de MySQL viene con una base de datos llamada “**test**” en la cual, cualquiera puede acceder. Esta base de datos es sólo para fines de pruebas o tutorial. La base de datos “**test**” se puede acceder por los usuarios que no tienen privilegios explícitos otorgados a esa base de datos. Por lo tanto, las bases de datos con nombre “test” se debe evitar en los servidores con aplicativos de producción.

3. Abortar conexiones

aborted_connects da el número total de intentos fallidos de conectarse a MySQL. El exceso de aborted_connects indica que el cliente no tiene suficientes privilegios, o el cliente utiliza una contraseña incorrecta, o que alguien está tratando

de cortar en su servidor.

`aborted_connects` un valor de estado global y puede ser recuperada mediante:

```
[sql]SHOW GLOBAL STATUS LIKE 'aborted_connects';[/sql]
```

4. Registro de errores

Los registros Log MySQL no sólo contiene información del servidor de arranque y tiempo de parada, sino también los errores críticos que ocurren mientras el servidor está en ejecución. En algunos sistemas operativos, el registro de errores contiene también seguimiento de la pila de ejecuciones de MySQL antes de bloquearse el servicio. Cualquier entrada de tipo (**error**) obviamente necesita su atención.

5. Puntos muertos InnoDB

InnoDB revierte las transacciones si hay un punto muerto. Conociendo su frecuencia es importante rastrear el origen de un punto muerto. Usted debe saber lo ocurrido en bloqueos y verificar que las solicitudes se han manejado correctamente o emprender las acciones pertinentes.

Para detectar bloqueos, utilice

```
[sql]SHOW ENGINE INNODB STATUS;[/sql]
```

6. Cambio en la configuración del servidor

Alguna vez se preguntó por qué su servidor estaba funcionando perfectamente hace unas semanas pero no ahora? Entonces usted debe buscar en los cambios de configuración de MySQL que se hicieron recientemente. La configuración de MySQL desempeña un papel crítico en el rendimiento de su servidor.

7. Slave lag

La mayoría de los servidores tienen uno o más esclavos de replicación y esclavos de monitoreo que son igualmente importantes. Si `slave_sql_running`, `slave_io_running` esta en NO, entonces el esclavo ha dejado de replicar y debe solucionarse en primer lugar. Si `seconds_behind_master` esta alto, más lento es su esclavo.

Mediante la ejecución:

```
[sql]SHOW SLAVE STATUS;[/sql]
```

8. Porcentaje máximo de conexiones permitidas

Si hay un valor alto en el porcentaje máximo de conexiones permitidas $\text{max_used_connections} / \text{max_connections}$ le dice que pronto podría quedarse sin conexiones de clientes y saturar el servidor. En otras palabras, las nuevas peticiones de conexión simplemente se negaran y cargarán de procesos el servidor. Así que asegúrese de que su `max_connections` es lo suficientemente grande para satisfacer su solicitud. Al ejecutar:

```
[sql]  
SHOW GLOBAL VARIABLES LIKE 'max_connections';  
SHOW GLOBAL STATUS LIKE 'max_used_connections';  
[/sql]
```

con esto, usted puede realizar un seguimiento máximo de conexiones permitidas.

9. Porcentaje de escaneos de tablas completo

El porcentaje de escaneos de tablas completa se representa mejor mediante

$$\left(\frac{\text{Handler_read_rnd_next} + \text{Handler_read_rnd}}{\text{Handler_read_rnd_next} + \text{Handler_read_rnd} + \text{Handler_read_first} + \text{Handler_read_next} + \text{Handler_read_key} + \text{Handler_read_prev}} \right).$$

Un aumento en escaneos de tablas completas (o en la creación de tablas temporales o similares) puede resultar en un desempeño no muy satisfactorio. Usted debe entender las consultas que causan esto y considerar realizar los cambios en los esquemas o las consultas si es necesario. Puede recuperar éstas métricas mediante la ejecución:

```
[sql]SHOW GLOBAL STATUS LIKE "Handler_read%";[/sql]
```