How to Change the MySQL Timeout on a Server

Article ID: **3630** | Last updated on **March 10, 2014** | Authored by: **Rosie Contreras**

A MySQL server timeout can occur for many reasons, but happens most often when a command is sent to MySQL over a closed connection. The connection could have been closed by the MySQL server because of an idle-timeout; however, in most cases it is caused by either an application bug, a network timeout issue (on a firewall, router, etc.), or due to the MySQL server restarting. Rarely does the `wait_timeout` value cause the problem, and changing the value does not fix the problem. For cases where an application fails to close a connection it is no longer using, a low `wait_timeout` value can help to avoid hitting `max_connections` simply due to "sleeping" idle connections that are not in a transaction and will not be reused.

Follow these steps to resolve the issue:

1. Login to your server using SSH.
2. Edit `my.cnf` (the MySQL configuration file).

```
sudo vi /etc/my.cnf
```

3. Locate the timeout configuration and adjust it to fit your server.

```
wait_timeout = 28800
interactive_timeout = 28800
```

   - The interactive timeout does not affect any web application connections. A high `interactive_timeout` but a low `wait_timeout` is normal and is the best practice.
   - Choose a reasonable `wait_timeout` value. Stateless PHP environments do well with a 60 second timeout or less. Stateful applications that use a connection pool (Java, .NET, etc.) will need to adjust `wait_timeout` to match their connection pool settings. The default 8 hours (`wait_timeout = 28800`) works well with properly configured connection pools.
   - Configure the `wait_timeout` to be slightly longer than the application connection pool's expected connection lifetime. This is a good safety check.
   - Consider changing the `wait_timeout` value online. This does not require a MySQL restart, and the `wait_timeout` can be adjusted in the running server without incurring downtime. You would issue `set global wait_timeout=60` and any new sessions created would inherit this value. Be sure to preserve the setting in `my.cnf`. Any existing connections will need to hit the old value of `wait_timeout` if the application abandoned the connection. If you do have reporting jobs that will do longer local processing while in a transaction, you might consider having such jobs issue `set session wait_timeout=3600` upon connecting.

4. Save the changes and exit the editor.
5. Restart MySQL to apply the changes as follows:

```
sudo /etc/init.d/mysql restart
```

Once the restart completes, the new changes are applied.

Was this content helpful? Yes No

4 Comments Rackspace Knowledge Center Login

Sort by Newest Share Favorite

Join the discussion…

Anurag • 3 months ago
interactive_timeout ~8 hours, for web applications, if app is not accessed in night time time or so,
then database connection from app will get reset I reckon, how do we handle such situation? Resetting connection again?
4 • Reply • Share ›

vivek • 6 months ago
Hie
I need help when i m running this command for increasing max_connection in ssh Console then i m getting the error -bash Command not found error is coming
plz help me fast and thank you in Advance
1 • Reply • Share ›

jonathanl • 10 months ago
Hi
I need MySql connection to stay open forever with interactions, either very rapidly or perhaps several days. The connection times out overnight, giving an error at the next attempted database write. Any suggestions

Thanks
• Reply • Share ›

H123.m • a year ago
Great Post.
Is reducing wait_timeout value can be dangerous when we have minidle configured in our jdbc connexion pool ?
• Reply • Share ›