FAQ

Table of content:

1. SQL DM for MySQL is licensed per server. Does that mean SQL DM for MySQL servers or MySQL servers?

It means MySQL servers. You may install as many instances of SQL DM for MySQL as you like as long as the total number of MySQL servers monitored does not exceed your license.

TOP

2. Do I need to install SQL DM for MySQL on the same host as MySQL?

MySQL servers, SQL DM for MySQL server(s) and Clients (browser) can be installed independently everywhere where a TCP connection (like an Internet /Intranet connection) is available. Available TCP connections is all that is required. Regarding installing SQL DM for MySQL on the same host as MySQL then SQL DM for MySQL Connects to/Monitors MySQL running on any platform.

TOP

3. What operating system does SQL DM for MySQL require?

Currently we support Windows (do not support Windows 2003) and Linux operating systems. Those are the Operating Systems where SQL DM for MySQL itself needs to be installed in. The SQL DM for MySQL client functionalities only require an Internet browser and any platform (including platforms for handheld devices like mobile phone, PDA, tablet PC, etc.). There are also no restriction as regards the platforms the MySQL servers that SQL DM for MySQL connects to - it can be any. Additionally, SQL DM for MySQL is able to retrieve OS data from Linux Operation Systems.

TOP

4. How do I upgrade SQL DM for MySQL?

This is actually a license-related question and a technical question as well.

- **License:** SQL DM for MySQL ships with 1 year of free upgrades. After that you will be offered an upgrade with discount. Our website always tells the terms and conditions. Also our website has a Portal for registered users from where you can download free upgrades and purchase upgrades after the expiry of the free upgrade period.
- **Technical:** The automatic installers (the Windows version and the RPM build for Redhat type Linux) handles everything automatically. The gz-compressed build for other Linux's requires that you run execute a few installation scripts from a command shell. We constantly improve and simplify this. After extracting the tar.gz package, you get a file called README. Please refer to that file for details.
- **Need to monitor more servers:** You can upgrade anytime your SQL DM for MySQL installation from monitoring a certain number of servers to higher numbers by opening the License Manager and enter a new license key. If you need this please first contact us through our ticket system. We will consider the value of your existing license and compensate you (details depend on what license you have and what you need and how old your existing license is).

> **Note**
>
> Do not forget to backup whatever JavaScript you have edited, as they get overwritten when you upgrade. You can take a back-up of Counters.def and Udo.def located inside SQL DM for MySQL folder: MONyog for this. Alternatively, you can directly upgrade. After upgrade you get your JS changes as a conflict in SQL DM for MySQL UI, you can resolve those and keep your changes.

TOP

5. Why should I upgrade?

Every new release adds features, fixes bugs, improves performance, stability, the GUI interface etc. Why should you NOT upgrade? Refer to Version History for details.

TOP

6. I have installed SQL DM for MySQL. What now? How do I get the reports?

What you installed was the MONyog(SQL DM for MySQL) service. This service is basically a webserver program. You connect to the service with an Internet browser by specifying the host where it is installed and the port that was specified when installing.

TOP

7. How can SQL DM for MySQL 'know all what it does'?

SQL DM for MySQL queries the MySQL servers about almost anything the server 'knows' except for data stored on those servers. The MySQL servers themselves store and maintain records of server configuration, users, history and much more. SQL DM for MySQL retrieves this information, organizes it, calculates on it and reports.

TOP

8. Where does SQL DM for MySQL store data?

This depends on the Operating system.

- Windows 2008:

Data folder:

```
C:\ProgramData\Webyog\MONyog\Data
```

MONyog.ini + MONyog.log + preferences.config are in:

```
C:\ProgramData\Webyog\MONyog\
```

- Windows Vista:

Data folder:

```
Windows Vista:{System_drive}:\ProgramData\Webyog\MONyog\Data
```

MONyog.ini + MONyog.log + preferences.config are in:>

```
Windows Vista:{System_drive}:\ProgramData\Webyog\MONyog
```

- Linux RPM build:

Data folder: The connection configuration and collected data are kept here. You can find directories named like 0001, 0002, 0003, etc.

```
/usr/local/MONyog/data/
```

Installation file:

```
/usr/local/MONyog/MONyog.ini
```

Log file:

```
/usr/local/MONyog/MONyog.log
```

Configuration file:

```
/usr/local/MONyog/preferences.config
```

- Linux .gz archive:

If you have extracted SQL DM for MySQL package in a directory called MONyog the data stored is in:

Data folder: The connection configuration and collected data are kept here. You can find directories named like 0001, 0002, 0003, etc.

```
MONyog/data/
```

Installation file:

```
 MONyog/MONyog.ini
```

Log file:

```
MONyog/MONyog.log
```

Configuration file:

```
MONyog/preferences.config
```

The data folder specified above is the default settings only. You can store in any position on any mapped/mounted drive that is writable.

TOP

9. How does SQL DM for MySQL store its data?

Except for two plain text files: the MONyog.log file and a very small .ini file (that contains information about the port on which SQL DM for MySQL listens, The SQL DM for MySQL administrator password and the path to the data folder), everything is kept in high-performance database files (SQLite format).

TOP

10. Can I move a SQL DM for MySQL installation to another computer while keeping the data stored in SQL DM for MySQL database?

Yes. Just install SQL DM for MySQL on the 2nd machine. After install, stop the running MONyog(SQL DM for MySQL) service and copy the .. \MONyog\Data folder from the old installation. You may also copy the MONyog.log if you want. All the connection configuration and the data is located in 'data' directory. The error log is MONyog.log and settings are stored in 'MONyog.ini' and 'preferences.config'. If you have made any changes to monitors they are stored in 'Counters.def' and for CSO's in 'udo.def'. Copy all of them from your old installation onto new PC. After that start the service again.

TOP

11. Can SQL DM for MySQL be configured as a virtual host in my 'ordinary' Apache webserver?

Yes, at least with Apache this is possible. In your Apache configuration file (httpd.conf) add something like this (where 'ip1.ip2.ip3.ip4' is the IP address you reserve for SQL DM for MySQL).

```
<VirtualHost *:80>
      ServerName  monyog.mydomain.com
      ServerAlias http://monyog.mydomain.com
      Redirect permanent /  https://monyog.mydomain.com
  </VirtualHost>

  NameVirtualHost *:443
  <VirtualHost *:443>
      ServerName  monyog.mydomain.com
      ProxyPreserveHost On
      ProxyPass /  http://127.0.0.1:<MONyog-Port>/
      ProxyPassReverse /  http://127.0.0.1:<MONyog-Port>/
      SSLEngine On
      SSLCertificateFile <path-to-ssl-certificate.crt>
      SSLCertificateKeyFile <path-to-ssl-key.key>
  </VirtualHost>
```

And run the following command on the machine running Apache server:

```
/usr/sbin/setsebool httpd_can_network_connect=1
```

After changing the configuration, restart the Apache server.

TOP

12. How can I access SQL DM for MySQL pages proxying through other webservers?

We can also access SQL DM for MySQL using Apache proxy. You need to follow these simple steps to configure your Apache server to support proxy.

Here, we can setup Proxy in system A and we assume that SQL DM for MySQL is installed in system B, now you can access SQL DM for MySQL using, "http:///monyog/".

Configurations on system-A:

1. Please check whether you have libxml2 installed in your system.
2. Download mod_proxy_html.c from http://apache.webthing.com/
3. Now build mod_proxy_html with apxs, apxs -c -I/usr/include/libxml2 -i mod_proxy_html.c
4. You need to load the following modules, so add the following entries in [/etc/httpd/conf/httpd.conf].

```
  LoadModule proxy_module modules/mod_proxy.so
  LoadModule proxy_http_module modules/mod_proxy_http.so
  LoadModule headers_module modules/mod_headers.so
  LoadModule deflate_module modules/mod_deflate.so
  LoadFile /usr/lib/libxml2.so
  LoadModule proxy_html_module modules/mod_proxy_html.so
```

5. Add the following configuration in your Apache configuration file [/etc/httpd/conf/httpd.conf]

ProxyRequests Off

```
<Proxy *>
            Order deny, allow
            Allow from all
            </Proxy>
            ProxyHTMLExtended On
            ProxyPass      /monyog/  http://<ip-system-B>:5555/
            ProxyHTMLURLMap http://<ip-system-B>:5555/ /monyog/
            <Location /monyog/>
                  ProxyPassReverse /
                  SetOutputFilter  proxy-html
                  ProxyHTMLURLMap  /      /monyog/
                  RequestHeader    unset  Accept-Encoding
            </Location>
```

TOP

13. How can I access SQL DM for MySQL pages proxying through nginx?

You can also access SQL DM for MySQL using nginx proxy. You need to follow these simple steps to configure your nginx server to support proxy. Here, we can setup Proxy in system A and we assume that SQL DM for MySQL is installed in system B, now you can access SQL DM for MySQL both over HTTP("http://") and HTTPS("https://").

Configuration of System A:

1. Install nginx on your system.
2. Create directories
   a. /var/log/nginx
   b. /var/www/cache

3. Configure nginx: Open nginx.conf found in /etc/nginx and add the following in the http section:

```
proxy_cache_path  /var/www/cache levels=1:2 keys_zone=my-cache:8m max_size=1000m inactive=600m;
 proxy_temp_path /var/www/cache/tmp;
```

A sample nginx.conf would like the following:

```
user  nginx;
 worker_processes  1;
 error_log  /var/log/nginx/error.log debug;
 pid        /var/run/nginx.pid;
 events {
    worker_connections  1024;
 }
 http {
   include       /etc/nginx/mime.types;
   default_type  application/octet-stream;
   log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                                '$status $body_bytes_sent "$http_referer" '
                                '"$http_user_agent" "$http_x_forwarded_for"';
   access_log  /var/log/nginx/access.log  main;
   sendfile        on;
   #tcp_nopush     on;
   keepalive_timeout  0;
   proxy_cache_path  /var/www/cache levels=1:2 keys_zone=my-cache:8m max_size=1000m inactive=600m;
   proxy_temp_path /var/www/cache/tmp;
   include /etc/nginx/conf.d/*.conf;
 }
```

4. Put your SSL certificates in /etc/nginx/conf/

5. Create a monyog.conf file inside /etc/nginx/conf.d/ and add the following:

```
server {
     server_name _;
     listen 80;
     location / {
            proxy_pass http://<ip-system-b>:5555;
            proxy_redirect     off;
            proxy_cache my-cache;
            proxy_cache_valid  200 302  0m;
            proxy_cache_valid  404      0m;
            proxy_set_header   Host             $host;
            proxy_set_header   X-Real-IP        $remote_addr;
            proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
            proxy_max_temp_file_size 0;
            client_max_body_size       10m;
            client_body_buffer_size    128k;
            proxy_connect_timeout      9000;
            proxy_send_timeout         9000;
            proxy_read_timeout         9000;
            proxy_buffer_size          4k;
            proxy_buffers              4 32k;
            proxy_busy_buffers_size    64k;
            proxy_temp_file_write_size 64k;
      }
}
server {
     server_name _;
     listen 443;
     ssl on;
     ssl_certificate /etc/nginx/conf/<certificate_name>.crt;
     ssl_certificate_key /etc/nginx/conf/<certificate_key>.key;
     location / {
            proxy_pass http://<ip-system-b>:5555;
            proxy_redirect off;
            proxy_cache my-cache;
            proxy_cache_valid 200 302 0m;
            proxy_cache_valid 404 0m;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_max_temp_file_size 0;
            client_max_body_size 10m;
            client_body_buffer_size 128k;
            proxy_connect_timeout 9000;
            proxy_send_timeout 9000;
            proxy_read_timeout 9000;
            proxy_buffer_size 4k;
            proxy_buffers 4 32k;
            proxy_busy_buffers_size 64k;
            proxy_temp_file_write_size 64k;
     }
}
```

TOP

14. Can I access SQL DM for MySQL pages using encrypted connection such as "https"?

Yes, you can access SQL DM for MySQL using "https", you may acquire a certificate from a certificate authority, such as Verisign or you may use the OpenSSL package to create your own certificate and configure your Apache webserver for "https".

Here are the steps you may follow to setup "https" in your Apache webserver.

1. Create a directory

```
mkdir sslcert
```

Now protect the directory,

```
chmod 0700 sslcert
```

2. Create two sub-directories

```
mkdir certs private
```

3. Create a database to keep track of each certificate

```
echo '100001' >serial
touch certindex.txt
```

4. Create a custom config file for OpenSSL to use similar to openssl.cnf in your /etc/pki/tls folder.

```
dir = .
   [ ca ]
   default_ca = CA_default
   [ CA_default ]
   serial = $dir/serial
   database = $dir/certindex.txt
   new_certs_dir = $dir/certs
   certificate = $dir/cacert.pem
   private_key = $dir/private/cakey.pem
   default_days = 365
   default_md = md5
   preserve = no
   email_in_dn = no
   nameopt = default_ca
   certopt = default_ca
   policy = policy_match
   [ policy_match ]
   countryName = match
   stateOrProvinceName = match
   organizationName = match
   organizationalUnitName = optional
   commonName = supplied
   emailAddress = optional
   [ req ]
   default_bits = 1024 # Size of keys
   default_keyfile = key.pem # name of generated keys
   default_md = md5 # message digest algorithm
   string_mask = nombstr # permitted characters
   distinguished_name = req_distinguished_name
   req_extensions = v3_req
   [ req_distinguished_name ]
   0.organizationName = Organization Name (company)
   organizationalUnitName = Organizational Unit Name (department, division)
   emailAddress = Email Address
   emailAddress_max = 40
   localityName = Locality Name (city, district)
   stateOrProvinceName = State or Province Name (full name)
   countryName = Country Name (2 letter code)
   countryName_min = 2
   countryName_max = 2
   commonName = Common Name (hostname, IP, or your name)
   commonName_max = 64
   0.organizationName_default = My Company
   localityName_default = My Town
   stateOrProvinceName_default = State or Providence
   countryName_default = US
   [ v3_ca ]
   basicConstraints = CA:TRUE
   subjectKeyIdentifier = hash
   authorityKeyIdentifier = keyid:always,issuer:always
   [ v3_req ]
   basicConstraints = CA:FALSE
   subjectKeyIdentifier = hash
```

5. Create a root certificate. All other certificates you create will be based of this. Since this is not a commercial certificate software may complain when they use your certificates. You may give people the "public" certificate and your certificate works like the commercial ones when they import it. To create, while in the 'sslcert' directory type:

```
openssl req -new -x509 -extensions v3_ca
-keyout private/cakey.pem -out cacert.pem -days 365 -config ./openssl.cnf
```

   You will be prompted for information and a password. Do not lose this password, make sure it is a secure one and back-up the two files that are created.
   The two files that are created are cacert.pem, which is the one you can give to others for import in their browsers, and cakey.pem, which will be in the private directory.
6. Create a key and signing request

```
openssl req -new -nodes -out name-req.pem
-keyout private/name-key.pem -config ./openssl.cnf
```

You will be prompted for information. The critical part is the "Common Name". This must be the server's hostname, such as mail.your.domain or the IP address. If you want to cover all subdomains you can enter *.your.domain. Use the "Organizational Unit" to remind you what the certificate is for, such as "Web Server". This generates two files:

- name-req.pem - the request
- name-key.pem - the private key in the private directory

7. Sign the request. This generates the certificate:

```
openssl ca -out name-cert.pem -config
./openssl.cnf -infiles name-req.pem
```

You will be prompted for the password used when creating the root certificate. Two files are created:

- <number.pem> - a copy of it in the certs directory
- name-cert.pem - which is the certificate

8. Copy to the correct location For Apache 2.x on Red Hat using the default location, the directory is:

   a. For the name-key.pem:

   ```
   cp
   name-key.pem /etc/httpd/conf/ssl.key/
   ```

   b. For the certificate:

   ```
   cp
   name-cert.pem /etc/httpd/conf/ssl.crt/
   ```

9. Create a Virtual Host

```
<VirtualHost ip-system-A>:443> DocumentRoot /var/www/html
     ServerName myserver
     ErrorLog /etc/httpd/logs/ssl_error_log
     TransferLog /etc/httpd/logs/ssl_access_log
     SSLEngine On
     SSLCertificateFile /etc/httpd/conf/ssl.crt/name-cert.pem
     SSLCertificateKeyFile /etc/httpd/conf/ssl.key/name-key.pem
      </VirtualHost>
```

10. Configure proxy in Apache described in FAQ 13 and restart Apache.
Edit the Hosts file [/etc/hosts]

```
<ip-system-A> myserver
```

TOP

15. What are the major differences between other major MySQL Monitoring Tool and SQL DM for MySQL?

They have similarities (in which they both differ from server-side scripts used for monitoring): they both make use of a HTTP service, a database and both use a web-browser for reporting. However, important differences are:

- SQL DM for MySQL needs no installation (of **'agents'**) on the server where the MySQL servers are running. Other does.
- SQL DM for MySQL **'has everything in itself'** - the webserver, the database, the MySQL client. It does not depend on the existence of other webservers, runtimes/Virtual Machines (like JAVA) and needs no separate database install. Other monitoring tools requires a full JDK (java), a TOMCAT server and a MySQL server instance for itself. Due to this simplified architecture install, configuration and first of all maintenance and upgrade is much simpler with SQL DM for MySQL. Download packages and disk storage required are much smaller with SQL DM for MySQL.

TOP

16. Can I trust the expertise of SQL DM for MySQL developers?

SQL DM for MySQL is developed by the Webyog Softworks that also created the most popular GUI for data management with the MySQL server - SQLyog Enterprise. We have more than 10 years of experience with designing MySQL related software. We have expanded our team with highly qualified developers ever since we started. We are devoted to constantly extending our knowledge and understanding of MySQL internals.

TOP

17. How does SQL DM for MySQL connect to MySQL?

SQL DM for MySQL uses the most proved and most efficient way of connecting: the native MariaDB Connector/C that is compiled into SQL DM for MySQL. Nothing else required: no separate client instance, no database abstraction layer (like ODBC/JDBC/ADO/.NET) and no webserver extensions (like PHP). Additionally, the connection can be 'wrapped' in a SSH tunnel. Also, SQL DM for MySQL implementation for this does not involve any other program (like Putty) running.

TOP

18. Windows warns after installation that SQL DM for MySQL may not have installed properly.

This can happen on some versions of Windows (Vista and higher) if you install a recent version of SQL DM for MySQL on top of an older version. The reason for this is that recent versions of Windows include a software called "Program Compatibility Assistant" (PCA) which tries to detect if an installer is running. It warns the user that the software might not have been correctly installed if the installer does not register a new uninstaller. The PCA is unable to detect changes made to an existing, registered uninstaller, which is what the new SQL DM for MySQL installers do. And thus, the warning is displayed. You can safely ignore this warning, but if it bothers you, you may just uninstall SQL DM for MySQL before upgrading to 3.5+. All collected data in the SQL DM for MySQL's data folder is still available after reinstall. However, you should:

- Backup the connections.data file before uninstalling
- Restore the old connections.data after the new install. After a restart, SQL DM for MySQL recognizes the connection settings in the old connections.data.

TOP

19. I would like to use SSH-tunnel, but my Windows server does not support it. Can that be fixed?

Yes, SSH support can be installed on Windows. You may install a complete Cygwin (Unix command line implementation for Windows). Alternatively, there are small packages available that support only a small subset of Cygwin (like SSH packages). Installation details depends on the exact Windows version.

TOP

20. SQL DM for MySQL throws an error when trying to connect to MySQL.

Please go through Error when trying to connect to MySQL. The same applies to SQL DM for MySQL as the client code is exactly the same in both programs. Observe however that everything related to HTTP-tunneling with SQLyog is not relevant for SQL DM for MySQL.

TOP

21. Failed to connect to MySQL: Can't connect to local MySQL server through socket... What can i do about this?

Ensure that the host specified resolves to an IP-address. This error occurs with some Linux distributions (most important Debian) when specifying 'localhost'. The system maps this to a Unix SOCKET file. SQL DM for MySQL connects through TCP and not to SOCKET. Try the IP '127.0.0.1' instead.

TOP

22. SQL DM for MySQL is taking up too much of system resources with the PROCESSLIST-based sniffer.

You may have noticed that, while using the PROCESSLIST-based sniffer, SQL DM for MySQL increases the load on the CPU as well as the I/O subsystem of the system on which it is installed - even when the MySQL server is idle. Do not panic: it is normal. When using the PROCESSLIST-based sniffer, SQL DM for MySQL continually queries the MySQL server at the end of each time interval, which you can specify. It then retrieves the results and stores them in an internal sniffer database before displaying the results back to you. Now, if you set a short time interval, one that almost approaches 0, then SQL DM for MySQL can get stuck in an infinite loop! Consequently, the load on the CPU and I/O subsystem increases exponentially. We generally recommend an interval of not more than 0.1 seconds times the number of servers for which Processlist-based sniffers are enabled. However, if you are worried that you may miss out on some important queries running on the MySQL server, use the Performance Schem or MySQL Proxy mode. The LUA script supplied with SQL DM for MySQL should handle the task for MySQL proxy. For more information review MySQL Proxy.

TOP

23. Why is display of queries truncated in Query Analyzer?

When using Query Analyzer feature, you may notice that sometimes queries displayed in output are incomplete. This may be due to one of the two known causes:

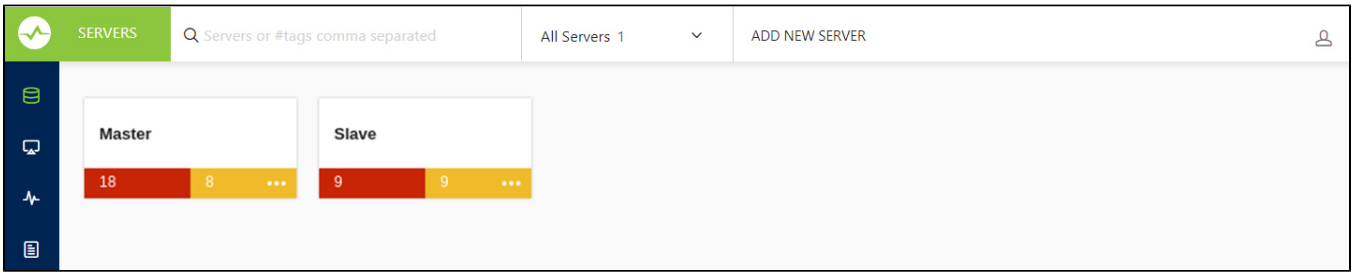
- As a security measure, SQL DM for MySQL extracts only the first 2000 characters of the query.

- MySQL does not record query delimiters in the General Log. Therefore, while analyzing the General Log, SQL DM for MySQL takes into consideration only the first line of the queries, and ignores the rest if they span over multiple lines.

TOP

24. The servers that I have registered do not display. What is wrong?

Check if the server is filtered based on a particular state, change your server filter to **All Servers** and now you can see your server between the servers if you had successfully registered it. You can also use the search bar next to the server filter to search for your server name or tag name to get to your server.



TOP

25. Now, anybody will be able to connect to my SQL DM for MySQL server and retrieve details about MySQL servers.

No, the SQL DM for MySQL authentication system will ensure that only those people that should have access have.

TOP

26. I have the same server registered twice. Metrics are reported different. Why?

For every registered server SQL DM for MySQL collects data independently. That is also the case when a server has been registered twice. Even if they were registered at the same time and even if the chosen sample interval is the same too, the connection and the server have some latency and data is not retrieved simultaneously. For that reason SQL DM for MySQL may retrieve and store slightly different values for each connection. This is most visible in the 'Delta' timeframe and least visible in the 'Current/all' timeframe. For GROUPING with 'History/Trends' the difference for each GROUP depends on the selected grouping interval. Due to laws of statistics the difference is less the longer the time interval (theoretically/statistically they converge more and more the closer time interval and/or the number of samples comes to infinity). Practically, you rarely need more than 20 samples in a GROUP for the difference to be negligible.

TOP

27. Does it affect the performance of a server if SQL DM for MySQL connects to it?

It does not affect on real 'live' servers. The queries sent by SQL DM for MySQL use almost no resources. We do not query data stored on disk and what we do query is stored in memory on the server. However if you are testing SQL DM for MySQL using a server instance that does almost nothing else and if you retrieve data at very short intervals the impact of SQL DM for MySQL may be slightly observable. The special Processlist feature (unique) may take a little more resources if there are lots of processes/client threads running. But SQL DM for MySQL only sends queries related to this when the corresponding SQL DM for MySQL client interface (the SQL DM for MySQL 'processlist' page) is open. Switching to another page or closing the browser stops sending the queries populating the SQL DM for MySQL processlist.

TOP

28. Is it possible to avoid that SQL DM for MySQL itself influences certain counters reported?

SQL DM for MySQL is a client. When it connects, the MySQL server starts a connection thread. And that connection is reported by SQL DM for MySQL. That cannot be avoided. The processlist feature has an option to 'filter out' SQL DM for MySQL connection - as well as other connections from other clients if you want - using a simple SELECT statement.

TOP

29. Can I customize SQL DM for MySQL counters?

Regarding customizing SQL DM for MySQL counters refer to Customization. For scripting examples refer Customization Scripting Examples.

TOP

30. I cannot sit watching a browser all the time - Can I get alerts if something goes wrong?

Yes, you can choose to get notifications (Email, SNMP traps, Slack, Pagerduty and Syslog) independently for every server and you can define your own warning levels and select what counters should raise an alert.

TOP

31. SQL DM for MySQL cannot identify if destination of the log file is on a "Mapped Network Drive". Why?

By default MONyog(SQL DM for MySQL) service runs under Local System Account. If you are having Slow query or General query log in a Mapped Network Drive, SQL DM for MySQL is not be able to reach it. If SQL DM for MySQL has to access the file present in a Mapped Network Drive, you have to convert the path into shared path (accessed with UNC notation: \system\share) and then follow these steps:

1. Click the **Start** menu, then click **Run** and then type,

   ```
   services.msc
   ```

2. After the Services window pops up with a list of all the services running in your system.
3. Search for Monyog and then right click --> **Properties**.
4. Click the **Log On** tab and then you can see that SQL DM for MySQL is using Local System Account.
5. You need to use **This account** option and then give the credentials that you use to log on to the system with Administrative privilege.
6. Save the settings, restart MONyog (SQL DM for MySQL) service.
7. After following the above steps try to access the file which is shared across network.

> **Note**
>
> The shared path should be accessed with UNC notation (\system\share). SQL DM for MySQL cannot identify if destination of the log file is on a Mapped Network Drive (this is a restriction with services on Windows and not with SQL DM for MySQL).

TOP

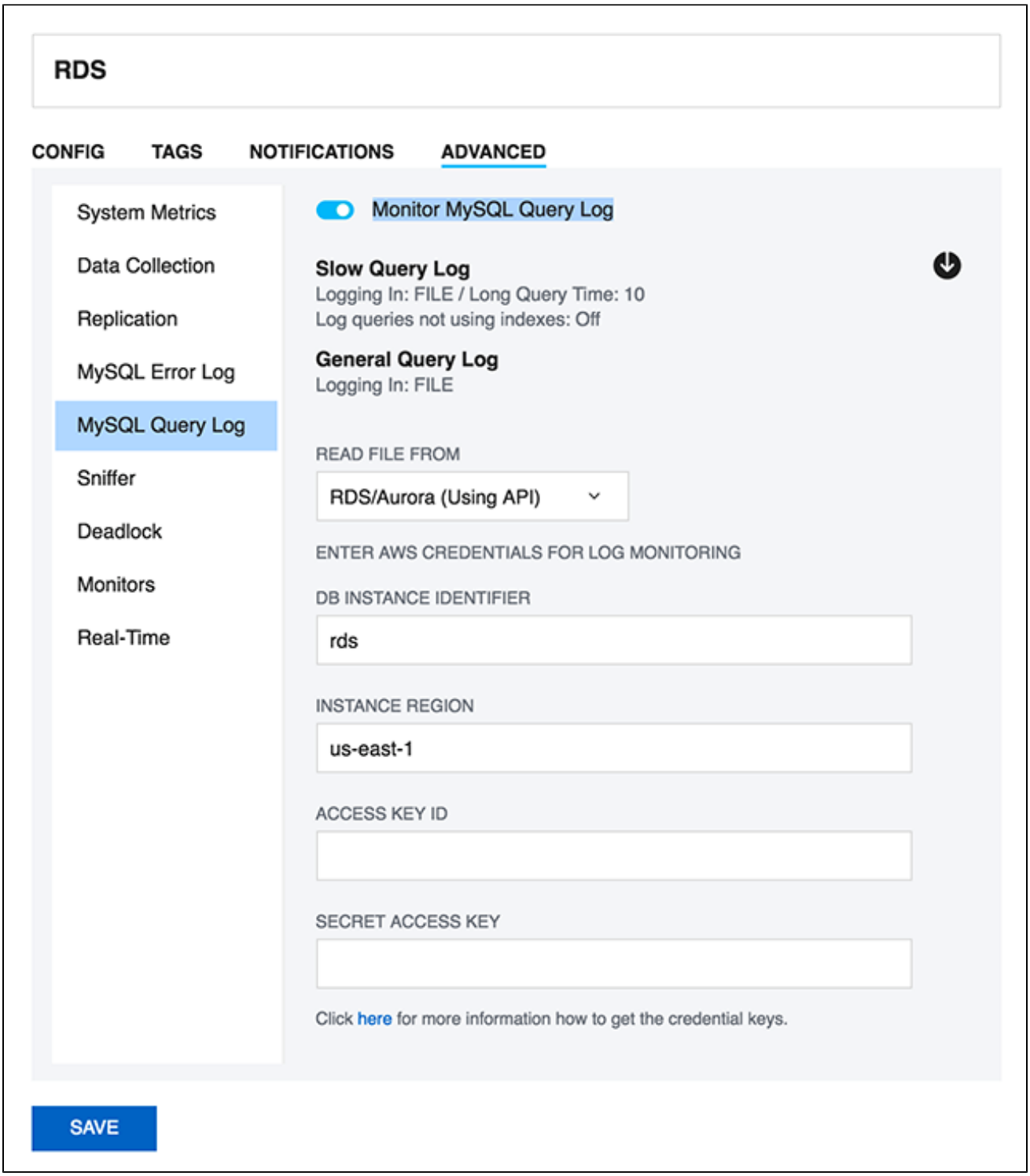
32. Failed to connect to MySQL: Unknown MySQL server host... What can I do about this?

You get this error if SQL DM for MySQL cannot resolve the hostname of a MySQL server. Ensure that other programs like ping, telnet, MySQL shell client are able to resolve the hostname to an IP-address. If yes, check "/etc/nsswitch.conf" of SQL DM for MySQL host. If the hosts section reads "files mdns4_minimal [NOTFOUND=return] dns mdns4", please change it to "files mdns4_minimal dns mdns4" or "files dns". This is introduced in some current Linux distribution. If other programs are not able to resolve the hostname, please check if host to IP resolution is properly defined inside "/etc/host" or in DNS server.

TOP

33. How can I monitor the queries from the file based RDS/Aurora Query logs?

SQL DM for MySQL can fetch the queries from the Slow Query log and General query log on Amazon RDS instance using the RDS REST APIs. SQL DM for MySQL requires the AWS access keys to fetch the file-based logs. Go to the **Edit Server->Advanced->MySQL Query log** and enable the option of "**M onitor MySQL Query Log**". Click the **Fetch logs**(down arrow) button and provide the AWS access key and secret access key to enable SQL DM for MySQL to monitor the log files.



TOP

34. What are future plans for SQL DM for MySQL?

SQL DM for MySQL is an important product for us. We plan to add new features as well as to 'refine' existing features. With the latest release we have completed what we originally planned for SQL DM for MySQL.

- It is now possible to get OS metrics from Amazon RDS/Aurora
- Added more notification channels (Slack, Pagerduty and Syslog) for SQL DM for MySQL alerts.

These features have all been requested by users. SQL DM for MySQL development has always been and continues to be very attentive to user requests. We update information here when plans for future developments have been decided.

TOP

35. How do I get help and report problems?

Four ways:

- Website Intercom https://www.idera.com/
- Post in our Forums http://community.idera.com/
- Create ticket: ideramysqlsupport@idera.com
- SQL DM for MySQL UI intercom

TOP

36. Can I use the keys generated from PuTTY for SSH connection?

SQL DM for MySQL does not support the key generated from PuTTY for SSH connection. However, you can convert the private key generated from "PuTTY key generator" into an open SSH key, and then use this key in SQL DM for MySQL to connect to the server. Here are the steps to follow:

- Go to PuTTY key generator, and generate a public/private key on your local system (refer the screenshot). Click the **Generate** button to generate the keys.

- Copy the public key generated under the "**Key**" space to the **authorized_keys** file, which is located in the **.ssh** directory on the remote host that you want to connect to.
- Go to **"Conversions"** in PuTTY key generator and click "**Export openssh key**" and save the new converted private key in a file.

- Now open the file containing the converted OpenSSH private key, copy this key and paste in the **"Private key"** field in SQL DM for MySQL (**Edit server -> SSH settings -> Private key**).

TOP

37. Steps to auto-start MONyog(SQL DM for MySQL) service with OS reboot in Ubuntu and Debian systems.

Users can make use of the "MONyog" script shipped with the Monyog(SQL DM for MySQL) package to auto-start MONyog(SQL DM for MySQL) service with OS reboot. The SQL DM for MySQL script is located at " /MONyog/bin/". Please follow the steps below:

- Copy the 'Monyog' to "/etc/init.d/" from " /MONyog/bin/"
- Open the 'Monyog' script located at "/etc/init.d/" and edit the variable "curdir" (line number 15) and set it to the path of bin. After editing, it should look like this: curdir="/home/Users/Downloads/MONyog/bin/"
- Make the script executable by 'chmod +x /etc/init.d/MONyog'
- Use debian utility update-rc.d to install the script: update- rc .d MONyog defaults

TOP

38. How to upgrade SQL DM for MySQL without losing your data or configuration?

The SQL DM for MySQL binaries are shipped in 3 packages: .tar, .rpm and .exe. The upgrade process is simple and depends on your package. Follow the steps below to upgrade to the latest version of SQL DM for MySQL:

For .rpm package :

```
rpm -Uvh <MONyog_package>.rpm
```

This command installs the latest build on top of your current installation.

For .tar package:

```
tar -xzvf <MONyog_package>.tar.gz
```

Please untar the package in the directory where the 'MONyog(SQL DM for MySQL)' package was untarred for the previous version to make sure that all your data and settings are intact.

For Windows (.exe) package:

Executing the file installs SQL DM for MySQL on top of the current installation.

All SQL DM for MySQL data and the configuration files are stored in a SQLite repository. In some of the SQL DM for MySQL GA releases, we modify /change the monitor definition in the SQLite files due to some bug or enhancements. In such cases, on upgrading, all the local changes made by the user in the previous version get replaced with the default shipped value and these local changes are shown as a conflict. You can see the conflicts as a notification on the top right-hand corner.

To resolve conflict

You can resolve these conflicts from the "**Settings -> Manage changes**" page, herein you get 2 options for all the listed conflicts: Use your changes /Discard your changes. 1- Use your changes, restores the local modifications which you had made in the previous version. 2- Discard your changes, replaces your changes with the default values.

TOP

39. How to maintain High Availability of SQL DM for MySQL?

Monit tool monitors the server process and can be used to maintain HA for SQL DM for MySQL.

Use the below commands to install monit:

apt:

```
sudo apt-get install monit
```

Yum:

```
sudo yum install monit
```

Once monit is installed, you can add programs and processes to the configuration file:

```
sudo nano /etc/monit/monitrc
```

Uncomment the below lines in the file to enable web interface. You can login to the web interface using the username ‘admin’ and password ‘monit’.

```
set httpd port 2812 and
use address localhost  # only accept connection from localhost
allow localhost        # allow localhost to connect to the server and
allow admin:monit      # require user 'admin' with password 'monit'
```

Also add the below lines in the configuration file to enable the monitoring of MONyog(SQL DM for MySQL) service:

Tar:

```
check process Monyog
    matching "MONyog"
    start program = "<MONyog extracted directory>/MONyog/bin/MONyog start"
    stop program = "<MONyog extracted directory>/MONyog/bin/MONyog stop"
```

Rpm:

```
check process Monyog
    matching "MONyog"
    start program = "/etc/init.d/MONyogd start"
    stop program = "/etc/init.d/MONyogd stop"
```

Once the above configuration changes are made, please use the below command to reload monit:

```
sudo monit reload
```

TOP