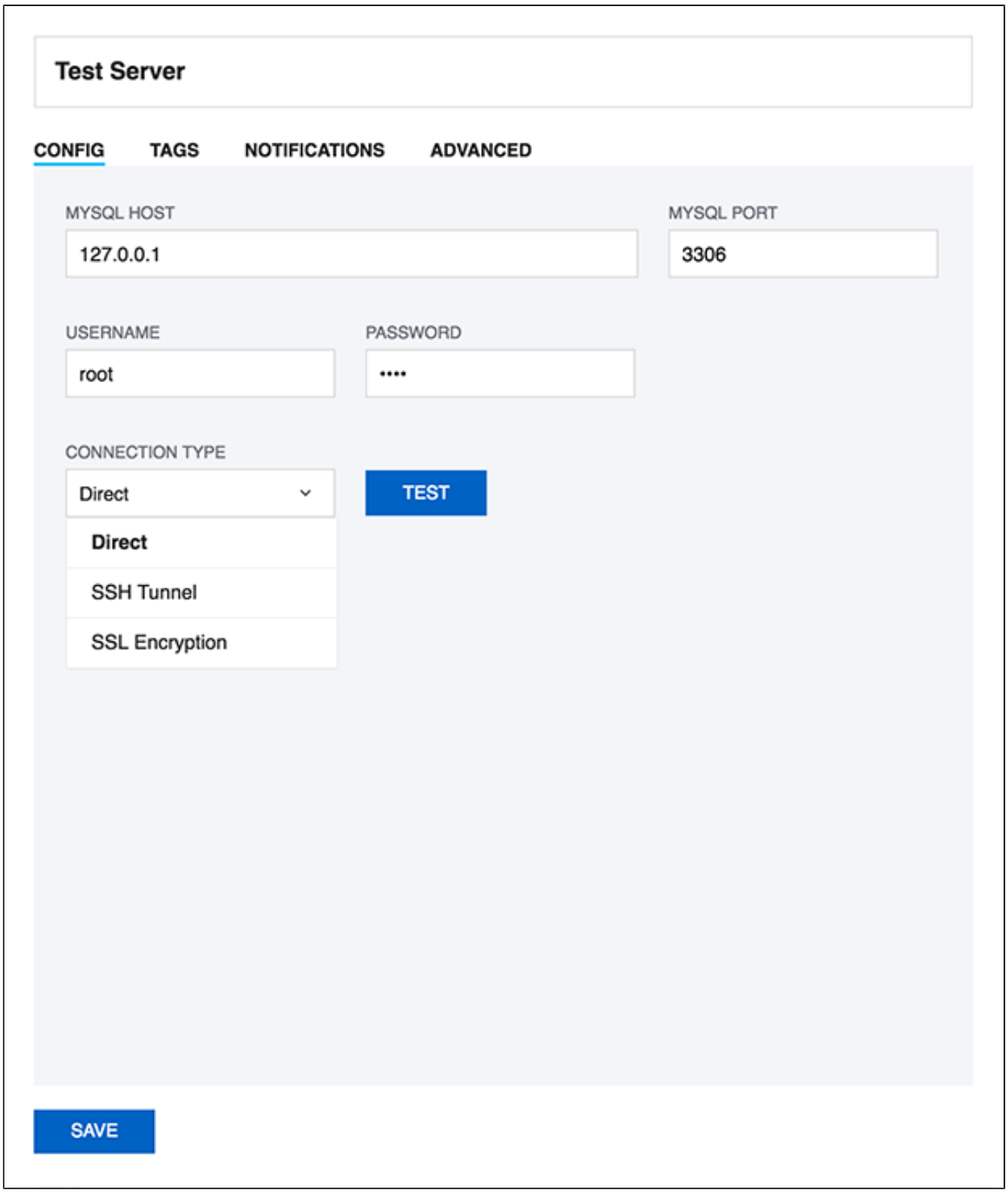
Connecting to MySQL Server

Registering Servers

First you must register the MySQL servers to which SQL DM for MySQL connects to. For every server you specify here, an embedded database is created. SQL DM for MySQL connects to those servers to retrieve and store information from that server.



For a **Direct connection**, enter the connection information as you would do from any client.

To retrieve system information (CPU, memory load etc.) you need to give the information required to create a remote command shell (Linux) from the server machine on which SQL DM for MySQL is running.

SSH tunneling

SQL DM for MySQL can send and receive encrypted authentication information as well as monitored data from MySQL using SSH tunneling. Also, it is possible to connect to MySQL with SSH tunneling even if the MySQL port (normally 3306) is blocked by a firewall or if users are not allowed to connect from remote hosts. An operating system user is required so that SQL DM for MySQL SSH client functionalities can use this user to connect to the SSHD daemon on the server.

If you want to use SSH tunneling to your MySQL server, you have to provide details for creating a SSH connection. Please refer to Using SSH connections, below.

To connect using SSH tunneling

- TCP port forwarding must be allowed in the SSH server. For openSSH servers it should be done in the "sshd_config" configuration file. In Linux usually it is in, `/etc/ssh/sshd_config`.

Set the "AllowTcpForwarding" option to "yes". So it should look like,

```
AllowTcpForwarding yes
```

- MySQL host should be specified 'relative' to the SSH server. Say, your SSH server is running in M1. Then you should ask yourself the following question: How should I connect to the target MySQL server from M1?

If your MySQL server is running in the same system M1, you can choose "**localhost**", "**127.0.0.1**" or the IP address of MySQL server which M1 can see.

SSH will listen on a specified port on the client machine, encrypt the data it receives, and forward it to the remote SSH host on port 22 (the SSH protocol port). The remote SSH host decrypts the data and forwards it to the MySQL server. The SSH host and the MySQL server do not have to be on separate machines, but separate SSH and MySQL servers are supported.

Using SSH connections

To create a SSH connection you need the following:

- **SSH Host**: Host of the machine on which SSH server is running.
- **SSH Port**: Port on which SSH server is listening. By default, it is 22.
- **SSH Username**: Username to access the SSH server (Note: not the MySQL server).

- **Authentication type**: Specify the type of authentication to use. This can be either key based or password based.
- If you have specified authentication type as Password - Provide the password.

- If you have specified authentication type as Key - You should note that SQL DM for MySQL only supports "OpenSSH standard key format" for key based authentication in SSH connections.
    - **Private Key**: Paste the content of your private key file. Again, do not specify the path to your private key file.
    - **Passphrase**: Enter the passphrase for your private key file (if any). This can be left blank, if no passphrase was given for the private key.

Using SSL Connection

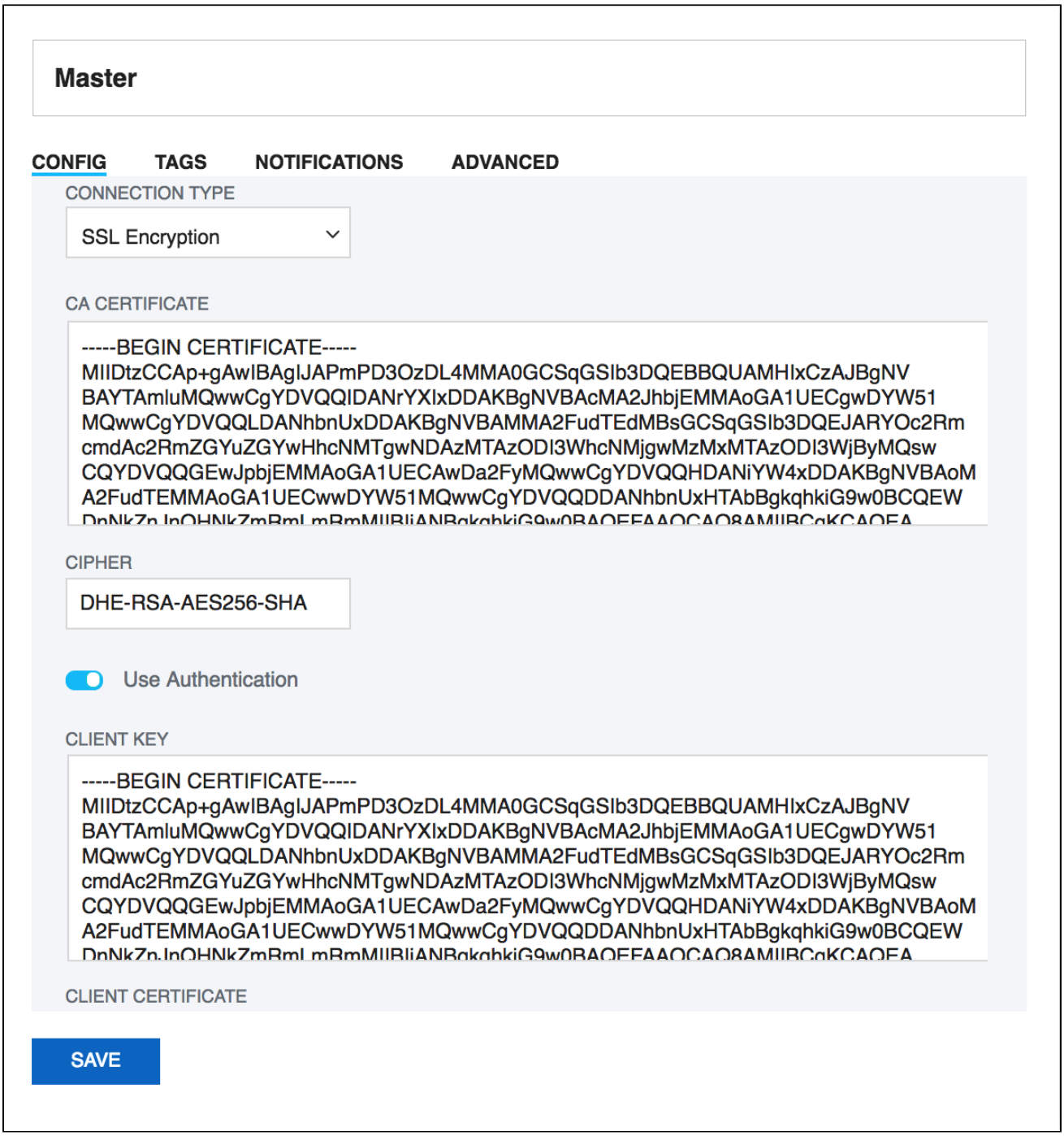
The Secure Sockets Layer (SSL) is a commonly-used protocol for managing the security of a message transmission on the Internet. SQL DM for MySQL provides native MySQL SSL encryption for direct MySQL connections.

To use SSL encryption you will be asked for:

- **CA Certificate**: The digital certificate issued by CA.
- **Cipher**: Encryption algorithm like DES, AES etc.

If you need client authentication then,

- **Client Key**: Private key of the client that is needed for encryption.
- **Client Certificate**: The client certificate.



SQL Diagnostic Manager for MySQL agentless and cost-effective performance monitoring for MySQL and MariaDB.

IDERA | Products | Purchase | Support | Community | Resources | About Us | Legal