

KeyTalk Server administrator manual notes

Last updated 4 June 2026

Table of Contents

1. Scope.....	2
2. Setting up KeyTalk Db	3
2.1 Setup Db from pre-built Db virtual server image	3
2.2 Setup Db on the existing Db server	3
2.3 Setting up Db from scratch (Linux)	3
2.4. Configure Db connection on KeyTalk server	5
2.5 Harden security of Db setup	7
3. Setting up KeyTalk MySQL authentication module backend	8
3.1. Db requirements.....	8
3.2. Quick Db setup (existing MySQL server)	8
3.3. Setting up Db from scratch (Linux)	8
3.4. Use SSL Certificate and Key for admin authentication [Optional]	10
4. Configuring KeyTalk with load balancers.....	12
4.1. General information.....	12
4.2. Quick setup	12
4.3. Setting up HAProxy load balancer	12

1. Scope

This document explains the process of configuring KeyTalk server databases (main Db and backend MySQL connector) and load balancer. The document is intended for administrators setting up KeyTalk.

This document refers to KeyTalk server v8.1 and nigher

2. Setting up KeyTalk Db

KeyTalk server is shipped with the built-in database residing with the application server. Such a configuration is handy for quick testing but hardly suits production use. This section explains how to configure KeyTalk database to run outside KeyTalk application server.

2.1 Setup Db from pre-built Db virtual server image

KeyTalk distributes pre-built virtual images of sample Db server which you can import in your KeyTalk setup. The VM images are available in OVF and VMDK formats (CLI login: user `keytalk`, password `change!change!`). Jump to the section [2.4](#) to configure connection to the Db connection from KeyTalk application server.

2.2 Setup Db on the existing Db server

Should you run MySQL Db server you can choose to host KeyTalk Db there.

2.2.1 Db requirements

- MySQL 8.x (you can choose any edition which suits your needs)
- 8GB RAM (or more)
- 200 GB free disk space (or more)

2.2.2. Setup KeyTalk Db

This section outlines brief setup leaving out the details of creating and configuring MySQL Db server. For the detailed instructions please see the following section.

1. Create KeyTalk database structure using `create-shared-db-tables.sql` script found in this directory.
2. Create users to access using `create-db-users.sql` script found in this directory.
3. Jump to the section [2.4](#) to configure Db connection from KeyTalk application server.

2.3 Setting up Db from scratch (Linux)

This section is for the ones who have chosen to set up KeyTalk Db server from scratch. The guidelines below were tested on Ubuntu 24.04 and with minor adjustments should apply for other Linux distributions.

Install MySQL 8.x

```
sudo apt -y update  
  
sudo apt -y install mysql-server mysql-client
```

Tune some performance-related

Db parameters

```
sudo mysql

SET PERSIST max_connections = 1000;
SET PERSIST join_buffer_size = 4 * 1024 * 1024;
SET PERSIST innodb_buffer_pool_size = 1024 * 1024 * 1024;
SET PERSIST innodb_redo_log_capacity = 256 * 1024 * 1024;
exit;
```

Make MySQL accessible from the outside

```
sudo sed -r -i 's/^\s*bind-address\s*=.*$/bind-address = 0.0.0.0/'
/etc/mysql/mysql.conf.d/mysqld.cnf

sudo sed -r -i 's/^\s*mysqlx-bind-address\s*=.*$/mysqlx-bind-address =
0.0.0.0/' /etc/mysql/mysql.conf.d/mysqld.cnf
```

Enable SSL

Upload MySQL Db certificates under `/etc/mysql/` directory. For staging purposes, you can use pre-generated MySQL certificates found under `/var/lib/mysql/`

```
sudo cp /var/lib/mysql/ca.pem /etc/mysql/
sudo cp /var/lib/mysql/server-cert.pem /etc/mysql/
sudo cp /var/lib/mysql/server-key.pem /etc/mysql/
```

Specify the location of SSL certificates and key by editing `[mysqld]` section in `/etc/mysql/mysql.conf.d/mysqld.cnf`

```
ssl-ca=/etc/mysql/ca.pem
ssl-cert=/etc/mysql/server-cert.pem
ssl-key=/etc/mysql/server-key.pem
```

Ensure the certificates and keys are readable by MySQL

```
sudo chown mysql /etc/mysql/*.pem
```

Effectuate your changes

```
sudo systemctl restart mysql.service
```

Creating KeyTalk database from CLI

Copy Db creation scripts `create-shared-db-tables.sql` and `create-db-users.sql` found in the directory with this manual to the system with MySQL server.

Create Db users

```
sudo mysql < ./create-db-users.sql
```

Create Database

```
sudo mysql -e 'DROP DATABASE IF EXISTS `keytalk`; CREATE DATABASE `keytalk`;'

sudo mysql keytalk < ./create-shared-db-tables.sql
```

2.4. Configure Db connection on KeyTalk server

Login to the KeyTalk administrator panel, go to *SYSTEM->Database*. First upload MySQL CA certificate.

- if you created MySQL Db from scratch, use CA file corresponding to `ssl-ca` option in your `mysqld.cnf`.
- if you used pre-built Db VM distributed with KeyTalk server upload `db_cacert-selfsigned.pem` found in this directory.

Once the CA is uploaded, change Db connection parameters.

Validate and save your Db connection settings. Applying new Db connection settings will take a while because it will trigger restart of all KeyTalk daemons.



KEYTALK Proof of Concept



System

Database

License

Time

Customize UI

Daemons

Advanced Configuration

Settings

Upgrade

Shut Down

Report Problem

Database

Configure Database Connection Settings

Db Host: *	192.168.206.136
Db Port: *	3306
Db Name: *	keytalk
Db User: *	keytalk-admin
Db Password: *	*****

Ok **Validate** **Cancel**

Db Server Verification CA ⓘ

Subject:	CN=MySQL_Server_8.0.46_Auto_Generated_CA_Certificate
Issuer:	CN=MySQL_Server_8.0.46_Auto_Generated_CA_Certificate
Serial Number:	01
Subject Alternative Names:	
Valid From:	04-06-2026 13:42 (04-06-2026 13:42 UTC)
Valid To:	01-06-2036 13:42 (01-06-2036 13:42 UTC)
Signature Algorithm:	sha256WithRSAEncryption
Public Key:	RSA (2048 bit)
SHA1 Fingerprint:	77b352a074b57ec64191d8adb3bc60589370a948

Choose File No file chosen

Choose File No file chosen

Choose File

Upload

2.5 Harden security of Db setup

At this point we successfully have setup KeyTalk Db and connected it to KeyTalk application server. For the sake of simplicity, we didn't pay much attention to security. This section supplies instruction to straighten security of your Db setup to make it production-ready.

2.5.1 Set strong Db password

Execute from MySQL server CLI (e.g. `sudo mysql keytalk`):

To change password for all remote KeyTalk application servers:

```
SET PASSWORD FOR 'keytalk-admin'@ '%' = 'new-password';
```

To change password for the specific KeyTalk application server:

```
SET PASSWORD FOR 'keytalk-admin'@'192.168.33.101' = 'new-password';
```

2.5.2 Use public SSL certificates on the Db

The ultimate solution would be to use certificates with trusted CA, e.g. from GlobalSign. If this happens to be an overkill to certificates used within organization only, follow the steps from “Enable SSL” paragraph of the section [2.3](#).

2.5.3. Restrict remote access to the Db

At this point we allow Db to be accessed from any remote host (`keytalk-admin`@`%``). We might want to restrict the Db access from the specific IP addresses only.

Execute from MySQL server CLI (e.g. `sudo mysql keytalk`):

```
DROP USER IF EXISTS `keytalk-admin`@`%`;
CREATE USER `keytalk-admin`@`192.168.33.102` IDENTIFIED BY 'secret' REQUIRE
SSL;
GRANT ALL ON `keytalk` TO `keytalk-admin`@`192.168.33.102`;
GRANT RELOAD ON *.* TO `keytalk-admin`@`192.168.33.102`;
FLUSH PRIVILEGES;
```

To restrict access to a specific subnet use wildcards like ``@`192.168.33.%``

3. Setting up KeyTalk MySQL authentication module backend

This section explains how to create and configure KeyTalk MySQL authentication module backend.

3.1. Db requirements

- MySQL 8 of any edition

3.2. Quick Db setup (existing MySQL server)

This section outlines brief setup leaving out the details of creating and configuring MySQL Db server. For the detailed instructions please see the following section.

1. Create database structure using `create-mod-mysql-db-tables.sql` script found in this directory.
2. Create users to access the Db.
3. Login to the KeyTalk administrator panel, go to *AUTHENTICATION->MySQL Modules* and select KeyTalk service of your choice. Configure connection to the Db.
4. Test your setup by authenticating with KeyTalk client.

3.3. Setting up Db from scratch (Linux)

This section outlines the detailed instructions to configure MySQL backend Db on Linux.

Install MySQL and enable SSL

The instructions are identical to setting up KeyTalk main Db from section 2.3.

Create database

Below we use *keytalk-backend* for Db name accessible by user *keytalk-admin* with password *secret*. Execute from MySQL server CLI (`sudo mysql`):

```
DROP DATABASE IF EXISTS `keytalk-backend`;
CREATE DATABASE `keytalk-backend`;
DROP USER IF EXISTS `keytalk-backend-admin`@`%`;
CREATE USER `keytalk-backend-admin`@`%` IDENTIFIED BY 'secret' REQUIRE SSL;
GRANT ALL ON `keytalk-backend`.* TO `keytalk-backend-admin`@`%`;
FLUSH PRIVILEGES;
EXIT;
```


Copy the Db creation script `create-mod-mysql-db-tables.sql` to the system with MySQL server.

```
sudo mysql keytalk-backend < ./create-mod-mysql-db-tables.sql
```

Please contact KeyTalk support if you require assistance in migrating previous certificate attribute data to the new structure.

Configure the user table mappings

On the 'Configure MySQL User Database Structure' section of the KeyTalk administrator panel it is possible to map the MySQL Db 'user' table columns to KeyTalk.

1. Login to the KeyTalk administrator panel.
2. Go to Authentication->MySQL Modules and select the required Service.
3. Select 'CHANGE' under the 'Configure MySQL User Database Structure'
4. Change the user table name & mappings where needed. The user database column mappings should match those in the 'user' database.
5. Uncheck credentials that are unused.
6. *Optional, should your database contain values that you want to appear as attributes of the end user certificate, for example 'Organization', 'Country' or 'Organization Unit':* Perform the previous steps for the Certificate attributes, which can be found below the user table mappings.
7. Validate and then save the changes for user table and certificate attribute table separately.

[Optional] Populate database with sample values

Execute from MySQL server CLI (`sudo mysql`):

```
USE `keytalk-backend`;
SET foreign_key_checks = 1;
LOCK TABLES `user` WRITE;
INSERT INTO `user` VALUES (
  1,
  'DemoUser',
  SHA1(concat('change!', '30db2e33ab6d61d76a57ce524b4ccb4b')),
  SHA1(concat('1234', '30db2e33ab6d61d76a57ce524b4ccb4c')),
  '30db2e33ab6d61d76a57ce524b4ccb4b',
  '30db2e33ab6d61d76a57ce524b4ccb4c'
);
UNLOCK TABLES;

LOCK TABLES `user_cert_attr_mappings` WRITE;
INSERT INTO `user_cert_attr_mappings` VALUES
(1,1, 'NL', 'KeyTalk', 'test@keytalk.com');
UNLOCK TABLES;
EXIT;
```

This will add user *DemoUser* with password *change!* and pincode *change* and several custom values to store in the generated certificate.

Configure Db connection on KeyTalk server

Login to the KeyTalk administrator panel, go to *AUTHENTICATION->MySQL Modules* and select KeyTalk service of your choice. Set Db connection parameters and upload Db CA certificate you used for *ssl-ca* option in your *mysqld.cnf*. Validate and save your Db connection settings.

Test your setup by authenticating with KeyTalk client.

3.4. Use SSL Certificate and Key for admin authentication [Optional]

Upload MySQL Db certificates under */etc/mysql/* directory. For staging purposes, you can use pre-generated MySQL certificates found under */var/lib/mysql/*

```
sudo cp /var/lib/mysql/ca.pem /etc/mysql/  
sudo cp /var/lib/mysql/server-cert.pem /etc/mysql/  
sudo cp /var/lib/mysql/server-key.pem /etc/mysql/
```

Specify the location of SSL certificates and key by editing *[mysqld]* section in */etc/mysql/mysql.conf.d/mysqld.cnf*

```
ssl-ca=/etc/mysql/ca.pem  
ssl-cert=/etc/mysql/server-cert.pem  
ssl-key=/etc/mysql/server-key.pem
```

Ensure the certificates and keys are readable by MySQL

```
sudo chown mysql /etc/mysql/*.pem
```

Copy the client certificate and key

```
sudo cp /var/lib/mysql/client-cert.pem /etc/mysql/  
sudo cp /var/lib/mysql/client-key.pem /etc/mysql/
```

Copy these files to a safe location, to be used to authenticate the Admin user. After you have copied the files, don't forget to secure the private key with:

```
sudo chmod 600 /etc/mysql/client-key.pem
```

Add admin user with SSL enabled

Run MySQL with `sudo mysql` and input:

```
DROP USER IF EXISTS `keytalk-ssl-admin`@`%`;
CREATE USER `keytalk-ssl-admin`@`%` REQUIRE X509;
GRANT ALL ON `keytalk-backend`.* TO `keytalk-ssl-admin`@`%`;
FLUSH PRIVILEGES;
EXIT;
```

Login to the KeyTalk administrator panel, go to *AUTHENTICATION->MySQL Modules* and select KeyTalk service of your choice. Configure the MySQL Database connection settings by clicking the *CHANGE* button underneath *Db Host*, *Db Port*, etc.

Choose *Authenticate using: certificate* and upload both *client-cert.pem* & *client-key.pem* which you stored in the previous step from */etc/mysql/*. Use the correct username, in this example: *keytalk-ssl-admin*. Validate and save the Db connection settings.

4. Configuring KeyTalk with load balancers

KeyTalk server can be configured to run behind load balancers to improve performance and availability.

4.1. General information

Generally, there are 2 setups possible when configuring load balancing of SSL servers.

1. The first setup expects all backend servers to have SSL certificate. In this setup LB operates on TCP level (L3/L4) and doesn't have SSL certificate/key. Because LB doesn't have a key it can't affect SSL traffic, but it still can re-route it.
2. The second setup expects a load balancer to operate on HTTPS level (L7) and hence having SSL certificate and key installed. Traffic from the LB towards the backend servers is then sent unencrypted.

KeyTalk supports only the first type of configuration with LB operating on TCP level.

4.2. Quick setup

Configure load balancers to relay TCP traffic coming from KeyTalk agents to the KeyTalk server backends.

Configure the load balancer to enable session affinity based on the client's IP address. This will prevent the load balancer from changing backend KeyTalk server during ongoing client-server session which would interrupt the session.¹

4.3. Setting up HAProxy load balancer

This section outlines the instructions to configure HAProxy on Linux to balance traffic to KeyTalk servers. The guidelines below are written for Ubuntu 16.04 though should be usable for other Linux distributions, probably with minor modifications.

Install and configure HAproxy

```
sudo apt -y install haproxy
```

Add to /etc/haproxy/haproxy.cfg:

```
frontend keytalk-frontend
```

¹ Clearly that relying on source IP address to maintain session affinity will be less reliable when the requestor's IP address may change frequently (mobile clients).

```
bind :443
mode tcp
option tcplog
default_backend keytalk-backend

backend keytalk-backend
mode tcp
# enable session affinity
balance source
server s1 keytalk-backend-svr1:443 check
server s2 keytalk-backend-svr2:443 check
```

Effectuate your changes

```
sudo service haproxy restart
```

Troubleshooting

See `/var/log/haproxy.log` and **make sure** `/etc/rsyslog.d/49-haproxy.conf` **contains** the following line:

```
if $programname startswith 'haproxy' then -/var/log/haproxy.log
&~
```