

KeyTalk Server administrator manual notes

Last updated 4 March 2024

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	6
2.5 Harden security of Db setup	8
2.6. Troubleshooting ‘SSL connection error: error:unsupported protocol’ error for MySQL 5.7	9
3. Setting up KeyTalk MySQL authentication module backend.....	10
3.1. Db requirements.....	10
3.2. Quick Db setup (existing MySQL server)	10
3.3. Setting up Db from scratch (Linux)	10
3.4. Use SSL Certificate and Key for admin authentication [Optional]	12
4. Configuring KeyTalk with load balancers.....	14
4.1. General information	14
4.2. Quick setup.....	14
4.3. Setting up HAProxy load balancer.....	14
5. Configuring KeyTalk with S/MIME certificates.....	16
5.1 LDAP service in KeyTalk.....	16

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 to be used KeyTalk administrators to configure KeyTalk server installations.

This document refers to KeyTalk server v7

2. Setting up KeyTalk Db

KeyTalk server is shipped with the build-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 setup and hook up 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!*). Jump directly to the section [2.4](#) to setup 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 suite your needs)
- 4GB RAM (8 recommended)
- 100 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 KeyTalk Db.
3. Jump to the section [2.4](#) to setup Db connection from KeyTalk application server.

2.3 Setting up Db from scratch (Linux)

This section is for the ones who has chosen to set up KeyTalk Db server from scratch. The guidelines below were tested on Ubuntu 22.04 yet they should still apply for other Linux distributions, possibly with minor adjustments. We begin with the setup which is functional. Once it works, we provide guidelines to harden security to make the Db ready for use in production.

Install MySQL 8.x

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

[Optional] 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
```

Enable SSL

If you have MySQL Db SSL certificates and key, copy them to `/etc/mysql` directory.

Otherwise, generate them:

```
sudo rm -f /var/lib/mysql/ca.pem /var/lib/mysql/server-cert.pem  
/var/lib/mysql/server-key.pem  
sudo mysql_ssl_rsa_setup --uid=mysql
```

Notice that removing the existing certificates before executing `'mysql_ssl_rsa_setup'` is essential since the latter does not overwrite the existing certificates.

Effectuate the generated certificates

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

Enable SSL by adding/uncommenting the following lines under the `[mysqld]` section of `/etc/mysql/mysql.conf.d/mysqld.cnf`

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

Make sure the certificates and keys are readable by MySQL

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

Effectuate the changes

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

Creating KeyTalk database from CLI

This section applies to admins who use command-line interface (CLI) to manage Db. If you make use of MySQL Workbench to manage your database, jump directly to the next section skipping this section all over.

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

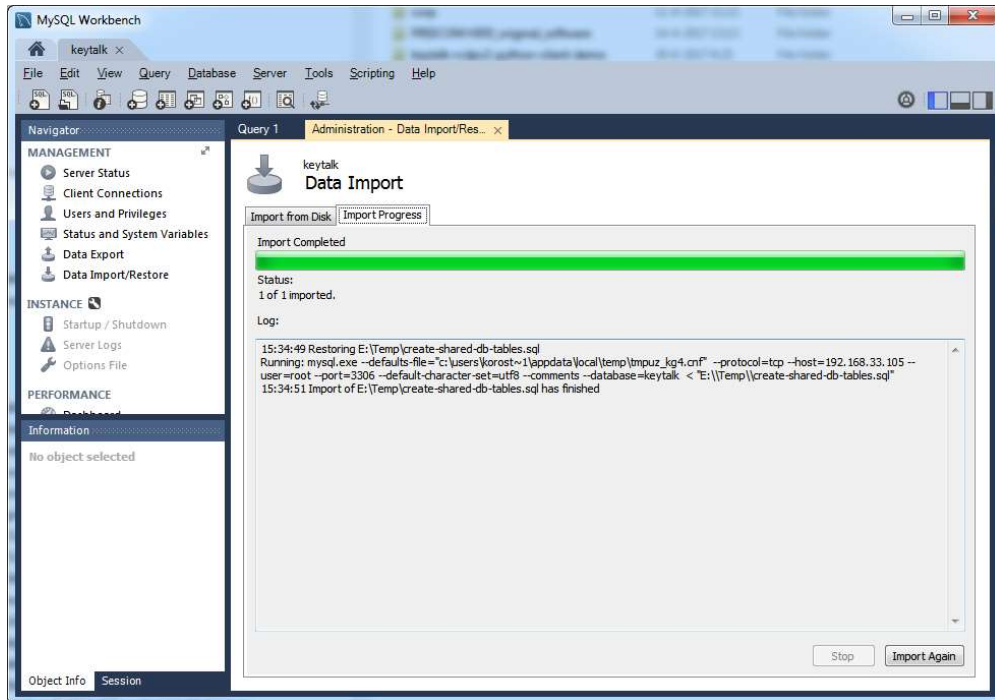
```
DROP DATABASE IF EXISTS `keytalk`;
CREATE DATABASE `keytalk`;
DROP USER IF EXISTS `keytalk-admin`@`%`;
CREATE USER `keytalk-admin`@`%` IDENTIFIED BY 'secret';
GRANT ALL ON `keytalk%`.* TO `keytalk-admin`@`%`;
GRANT RELOAD ON *.* TO `keytalk-admin`@`%`;
FLUSH PRIVILEGES;
EXIT;
```

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

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

Create database with MySQL Workbench¹

¹ In this manual we use MySQL Workbench version 6.3




2.4. Configure Db connection on KeyTalk server

Login to the KeyTalk administrator panel, go to *SYSTEM->Database*. Set Db connection parameters and 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.

Validate and save your Db connection settings.



MAIN

SERVICES

AUTHENTICATION

DEVID USERS

ACCOUNTING

Database | Time | Daemons | Settings | Upgrade | Shut Down | Report Problem

DB CONNECTION SUCCESSFUL

Configure Database Connection Settings

Db Host: *192.168.33.105

Db Port: *3306

Db Name: *keytalk

Db User: *keytalk-admin

Db Password: ******

OK

VALIDATE

CANCEL

Db Server Verification CA ⓘ

Subject:CN=MySQL_Server_5.7.18_Auto_Generated_CA_Certificate

Issuer:CN=MySQL_Server_5.7.18_Auto_Generated_CA_Certificate

Serial Number:01

Subject Alternative Names:

Valid From:14-06-2017 14:46 (14-06-2017 14:46 GMT)

Valid To:12-06-2027 14:46 (12-06-2027 14:46 GMT)

Signature Algorithm:sha256WithRSAEncryption

Public Key:RSA (2048 bits)

SHA1 Fingerprint:b024823d941e93b01fd92577a37bdc92cffb432

Choose File

No file chosen

UPLOAD

REMOVE

DOWNLOAD

Use KeyTalk admin panel to create KeyTalk service with authentication connector of your choice. Test your setup by authenticating with KeyTalk client.

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.%``

2.6. Troubleshooting ‘SSL connection error: error:unsupported protocol’ error for MySQL 5.7

Should you happen to choose to keep the existing MySQL v5.7 Db from KeyTalk v5 by upgrading the application server only, you might encounter the following error in KeyTalk logs:

```
SSL connection error: error:1425F102:SSL
routines:ssl_choose_client_version:unsupported protocol
```

The error is caused by the fact that your MySQL Db version is too old and does not support TLSv1.3 which is a minimal TLS protocol required by KeyTalk 7. Normally MySQL Db server should be kept up to date using Ubuntu update system, which would automatically upgrade MySQL to v5.7.28 built against OpenSSL and featuring TLSv1.3. Should the Db server not kept up-to-date e.g. because no outbound Internet access was ever configured on it, the server would keep using the original MySQL v5.7.20 built against yaSSL and featuring TLS 1.0 and TLS v.1.1 only. To resolve it you should manually upgrade the Db server with:

```
sudo apt update
sudo apt -y upgrade
sudo apt clean autoclean
sudo apt autoremove --purge -y
```

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 5.7 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-5.7 and enable SSL

The instructions are identical to setting up KeyTalk main Db from the 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 < /path/to/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 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]

Enable SSL

If you have MySQL Db SSL certificates and key, copy them to */etc/mysql* directory.

Otherwise you might choose to make use of the sample certificates and keys pre-generated by MySQL under */var/lib/mysql* (`sudo mysql_ssl_rsa_setup --uid=mysql` to re-generate)

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

Enable SSL by uncommenting the following lines in */etc/mysql/mysql.conf.d/mysqld.cnf*

```
[mysqld]
ssl-ca=/etc/mysql/cacert.pem
ssl-cert=/etc/mysql/server-cert.pem
ssl-key=/etc/mysql/server-key.pem
```

Make sure 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.²

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
```

² 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
&~
```

5. Configuring KeyTalk with S/MIME certificates

KeyTalk server can be configured to store S/MIME certificates to LDAP servers. S/MIME certificates are used for two purposes: signing and encrypting emails. Signing is used to sign an email, providing the sender's identity. Encryption is used to encrypt an email so that only the intended recipient can decrypt and read the email. If so configured KeyTalk can generate the required certificates and can store them as a user attribute in LDAP.

KeyTalk has been tested against Windows Server 2019 Active Directory and OpenLDAP version 2.4.42.

5.1 LDAP service in KeyTalk

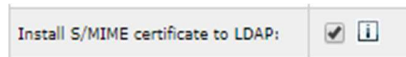
Prerequisites


- LDAP server (Active Directory or OpenLDAP) set up.
- KeyTalk LDAP service set up. This service will be further configured in this chapter.
- All users in LDAP server must have their email address set in an attribute.

Set up 'Install S/MIME certificate to LDAP'

On the *SERVICES* page in the admin panel, edit the desired service using the pencil icon. To ensure certificate S/MIME compatibility, set these settings in the Certificate Settings section:

- Check 'Install S/MIME certificate to LDAP'



Install S/MIME certificate to LDAP: ☒ 

- Ensure 'Subject Email' is set
- Set the 'Time To Live (sec)' value to the desired duration
- If any 'Key Usage' checkbox is selected: ensure that either 'digitalSignature', 'nonRepudiation' or both are selected
- If any 'Extended Key Usage' checkbox is selected: ensure that 'emailProtection' is entered in the 'Additional OIDs' field

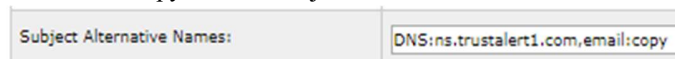


Key Usage: ☒ digitalSignature ☒ nonRepudiation ☒ keyEncipherment 
☒ dataEncipherment ☒ keyAgreement ☐ keyCertSign

Extended Key Usage: ☒ clientAuth ☐ serverAuth ☒ emailProtection

Additional OIDs:
OID1,OID2,...

- Add 'email:copy' to the 'Subject Alternative Names' field




Subject Alternative Names:

- Click 'OK' at the bottom to save
- Navigate to AUTHENTICATION -> LDAP Modules, choose the service and click 'Configure'
- Click Change under the 'Certificate to LDAP attribute mappings' section
- Filter should be set, e.g.: "(cn=\$(userid))"

- For S/MIME it is necessary to have the user's email address known. Map the following values unless users log in to KeyTalk with their email address as their account name.
 - > Map 'Email' to "mail" (& check the box). This may be required to successfully use the S/MIME certificate
 - > Map 'Common Name' to "mail" (& check the box)

Configure certificate to LDAP attribute mappings for Service CUST_PASSWD_LDAP

Filter: * 

Mapped	Certificate attribute	LDAP attribute
<input type="checkbox"/>	Country	<input type="text"/>
<input type="checkbox"/>	State	<input type="text"/>
<input type="checkbox"/>	City/Locality	<input type="text"/>
<input type="checkbox"/>	Organization	<input type="text"/>
<input type="checkbox"/>	Organization Unit	<input type="text"/>
<input checked="" type="checkbox"/>	Common Name	mail
<input checked="" type="checkbox"/>	Email	mail
<input type="checkbox"/>	Time To Live (sec)	<input type="text"/>
<input type="checkbox"/>	Time For Correction (sec)	<input type="text"/>
<input type="checkbox"/>	Basic Constraints	<input type="text"/>
<input type="checkbox"/>	Key Usage	<input type="text"/>
<input type="checkbox"/>	Extended Key Usage	<input type="text"/>
<input type="checkbox"/>	Subject Alternative Name	<input type="text"/>

- Click 'OK' to save.