

# データベースのセットアップ (MariaDB のインストールからテーブル暗号化まで)

2017 年 1 月 28 日 溝上友貴

## 目次

|                                      |   |
|--------------------------------------|---|
| MariaDB の暗号化について.....                | 1 |
| 1. 現在の MariaDB 5.5.50 をアンインストール..... | 4 |
| 2. MariaDB 10.1(stable)をインストール.....  | 4 |
| 3. MariaDB に暗号化設定.....               | 5 |
| 4. MariaDB の起動.....                  | 6 |

初期状態 (CentOS7 に MariaDB-5.5.50 がインストールされている)

```
[tomoki@localhost ~]$ rpm -qa | grep -i "mariadb"
mariadb-5.5.50-1.el7_2.x86_64
mariadb-server-5.5.50-1.el7_2.x86_64
mariadb-libs-5.5.50-1.el7_2.x86_64
```

Figure. MariaDB 存在確認 01281705

MySQL5.7 で透過的データ暗号化 (テーブルごとの暗号化) ができるらしいが、その前に、MariaDB について調べてみる。

<https://mariadb.com/kb/en/mariadb/data-at-rest-encryption/> によれば

テーブルの暗号化は MariaDB10.1.3 に追加された。MariaDB 10.1.4 では大きな変更が加えられた。以下の説明は MariaDB 10.1.4 についてのものである。

## MariaDBの暗号化について

### 概要

- ・暗号化で 3~5%のオーバーヘッドあり

### ストレージエンジン

- ・XtraDB と InnoDB ストレージエンジンでサポート
- ・ARI ストレージエンジンでは、ROW\_FORMAT=PAGE で作成されたテーブルのみ
- ・暗号化は、(すべて) (個々のテーブル) (個々のテーブル以外のすべて) で行うことができる
- ・XtraDB と InnoDB のログファイルを暗号化することもできる

### 暗号化キー

- ・複数の暗号化キーをサポート
- ・暗号化キーは、32 ビットのキー識別子で判別
- ・同じキーの古いバージョンから新しいバージョンのデータを自動的に再暗号化する
- ・キー管理は、プラグインによって管理 (file\_key\_management)

### file\_key\_management プラグイン

- ・ファイルから暗号鍵を読み込む
- ・file\_key\_management\_filename : 暗号化キーのファイルがある場所。オプション必須
- ・file\_key\_management\_filekey : キーファイルを解読するオプションのキー。キーが FILE: で始まる場合は、残りの値はキーを格納したファイルへのパスと解釈
- ・暗号化キーには適切なパーミッション必要 (mysql はそれを読める。権限のないユーザは読むことができない)
- ・file\_key\_management\_encryption\_algorithm : 使用する暗号化アルゴリズム

file\_key\_management\_filename

- ・識別子 (32 ビット整数) と 128, 192, または 256 ビットのキーをサポート
- ・コメントはハッシュ文字#から始まる
- ・識別子とキーはセミコロン ; で分けられる

(例)

```
# this is a comment
```

```
1;770A8A65DA156D24EE2A093277530142
```

・システムの XtraDB/InnoDB テーブルスペースとログは常にキー番号 1 を使用するため、キー番号 1 は存在する必要がある。

- ・キー番号 2 はオプション。一時表と一時ファイルに使用される。
- ・キーローテーション (キーの変更) をサポートしておらず、すべてのキーはバージョン 1
- ・サーバ起動時にキーファイルを読み取れないと、暗号化は機能せず、テーブルが読み取れなくなる

file\_key\_management\_filekey

・キーファイルは暗号化され、それを復号するキーはオプションの file\_key\_management\_filekey パラメータで与えられる。キーファイルを暗号化するには、OpenSSL コマンドラインユーティリティを使用する

(例) openssl enc -aes-256-cbc -md sha1 -k secret -in keys.txt -out keys.enc

- ・-aes-256-cbc と sha1 を指定する必要がある

file\_key\_management\_encryption\_algorithm

- ・2つの暗号化アルゴリズム (AES/CBC または AES/CTR) をサポート
- ・AES/CTR は推奨されているが、最新の OpenSSL が必要なため利用できない場合あり
- ・AES/CBC : CBC モードで 128 ビットのキーを持つ AES
- ・AES/CTR : テーブルスペースページ (InnoDB, XtraDB, or Aria) を暗号化するために CTR モードで 128 ビットのキーを持つ AES、テンポラリファイルの場合は認証された GCM モードで AES を使用

(例) my.cnf

[mysqld]

```
file_key_management_encryption_algorithm=aes_cbc
```

```
file_key_management_filename = /home/mdb/keys.enc
```

```
file_key_management_filekey = secret
```

## データの暗号化 (XtraDB/InnoDB)

- ・暗号化プラグインをロードし、ストレージエンジンを使用するように設定する必要がある
- ・次の変数の設定を行う

画像の引用元を忘れてしまったので利用できません。

- ・innodb-encrypt-tables : テーブルを暗号化するかどうか
- ・innodb-encrypt-log : ログファイルの暗号化を可能にする
- ・innodb-encryption-threads : キーの変更とスクラブを変更するスレッドの数

- ・ テーブルの暗号化 (innodb-encrypt-tables) を行い, ログの暗号化 (innodb-encrypt-log) は行わない, このようにはすべきではない. 逆の場合は構わない.
- (例) my.cnf (XtraDB の暗号化を可能にする例)

```
[mysqld]

plugin-load-add=file_key_management.so

file-key-management

file-key-management-filename = /mount/usb1/keys.txt

innodb-encrypt-tables

innodb-encrypt-log

innodb-encryption-threads=4
```

### 暗号化するテーブルの指定

- ・ innodb\_file\_per\_table をオン (デフォルト) に設定
- ・ CREATE TABLE または ALTER TABLE に次のオプションを指定

利用できません

- ・ ENCRYPTED ・ ENCRYPTION\_KEY\_ID

(例)

```
CREATE TABLE T (id int, value varchar(255)) ENCRYPTED=YES ENCRYPTION_KEY_ID=17;
```

テーブル T は, キー 17 で暗号化される

```
ALTER TABLE T ENCRYPTED=YES ENCRYPTION_KEY_ID=18;
```

テーブル T は, キー 18 で再暗号化される

- ・ innodb\_file\_per\_table が OFF の場合, または ENCRYPTION\_KEY\_ID が見つからない場合は, errno : 140 の " Wrong create options " でエラーを起こす

- ・ そのほかの関係のある変数

利用できません.

- ・ innodb\_encrypt\_tables
- ・ innodb\_default\_encryption\_key\_id

こちらは, 実際にデータベースを暗号化するための設定を行った人のウェブページである.

<https://orebibou.com/2016/02/mariadb%E3%81%A7%E6%9A%97%E5%8F%B7%E5%8C%96%E3%81%97%E3%81%9F%E3%83%86%E3%83%BC%E3%83%96%E3%83%AB%E3%82%92%E4%BD%9C%E6%88%90%E3%81%99%E3%82%8B/>

MySQL 5.7.12(11) でもデータベース暗号化を行うことができる（参考はこちら  
<http://qiita.com/hmatsu47/items/476d446887244de17ae4> ）

次に、暗号化を行うために、現在のMariaDB をアンインストールし、 MariaDB 10.1.4 以降のものをインストールする。

こちらのサイト <http://server.etutstplus.com/centos-7-mariadb-install-and-mysql-secure-installation/> を参考にしてMariaDB 10.1 のインストールを行う。

## 1. 現在のMariaDB 5.5.50 をアンインストール

アンインストール

```
$ sudo yum remove mariadb mariadb-libs
```

```
[tomoki@localhost ~]$ rpm -qa | grep -i "mariadb"
[tomoki@localhost ~]$
```

Figure. 削除確認（成功）

## 2. MariaDB 10.1(stable)をインストール

<https://mariadb.com/kb/ja/yum/> に従って、MariaDB をインストールする。

yum でインストールするために、 /etc/yum.repos.d/MariaDB.repo を作成する。

※以降、毎回 sudo するのがめんどうなので管理者権限で行う

```
# vim /etc/yum.repos.d/MariaDB.repo
```

/etc/yum.repos.d/MariaDB.repo

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.1/centos7-amd64
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```

(a) [a] で編集

(b) [Esc] でコマンドモードに移行

(c) [Shift] + [z] 2回で保存して vim エディタ終了

MariaDB-server, MariaDB-client, MariaDB-devel をインストール

```
# yum install --enablerepo=mariadb MariaDB-server MariaDB-client MariaDB-devel
```

```
[root@localhost tomoki]# rpm -qa | grep -i "mariadb"
MariaDB-client-10.1.21-1.el7.centos.x86_64
MariaDB-devel-10.1.21-1.el7.centos.x86_64
MariaDB-common-10.1.21-1.el7.centos.x86_64
MariaDB-server-10.1.21-1.el7.centos.x86_64
```

Figure. インストール確認

設定ファイル /etc/my.cnf.d/server.cnf を編集して MariaDB の設定を行う。

/etc/my.cnf.d/server.cnf にはほとんどなにも書かれていないので、MySQL の設定ファイルをコピーすることにする。

```
[root@localhost mysql]# ls -l /usr/share/mysql/my-*.cnf
-rw-r--r--. 1 root root 4920 1月 18 04:49 /usr/share/mysql/my-huge.cnf
-rw-r--r--. 1 root root 20438 1月 18 04:49 /usr/share/mysql/my-innodb-heavy-4G.cnf
-rw-r--r--. 1 root root 4907 1月 18 04:49 /usr/share/mysql/my-large.cnf
-rw-r--r--. 1 root root 4920 1月 18 04:49 /usr/share/mysql/my-medium.cnf
-rw-r--r--. 1 root root 2846 1月 18 04:49 /usr/share/mysql/my-small.cnf
```

Figure. MySQL の設定ファイル一覧

my-small.cnf : 64MB 以下の使用用途

my-medium.cnf : 使うメモリ数が少ないシステム (32 - 64MB) , または Web サーバなどの他のプログラムと同時に使用されるシステム (-128MB)

my-large.cnf : 主に MariaDB がメモリを使うシステム (512MB)

my-huge.cnf : 1G-2G

今回は、my-large.cnf を選択する。

```
# cp -p /usr/share/mysql/my-large.cnf /etc/my.cnf.d/server.cnf
cp: '/etc/my.cnf.d/server.cnf' を上書きしますか? y
```

次に、データベースの保存先を (/var/lib/mysql) に変更し、文字化けしないよう文字コードを UTF-8 に変更。

```
# vim /etc/my.cnf.d/server.cnf
```

[client] に追加

```
default-character-set = utf8
```

[mysqld] に追加

```
datadir=/var/lib/mysql
character-set-server = utf8
```

これで基本のインストールは完了である。

### 3. MariaDB に暗号化設定

鍵ファイルを作成するために、OpenSSL のコマンドを使って鍵を生成する。

```
# openssl enc -aes-256-cbc -k secret -P -md sha1
salt=38301872F2B4FDC8
key=7979BE40ED44A2E0C7A19437F06F55BACA6500AE32D4D9512BF0CBF7D00733C6
iv =78B3D98BD3C8E813EB7F7724909C4B91
```

この内容をフォーマット「キー番号;キー」でテキストファイルに書き込む。今回は/opt/MariaDB/key.txt に書き込んだ。

/opt/MariaDB/key.txt

```
1;78B3D98BD3C8E813EB7F7724909C4B91
```

このテキストファイルから OpenSSL を使ってキーのファイル (/opt/MariaDB/key.enc) を生成する。

```
# openssl enc -aes-256-cbc -md sha1 -k secret -in /opt/MariaDB/key.txt -out  
/opt/MariaDB/key.enc
```

確認

```
# ls -la /opt/MariaDB/key.enc  
-rw-r--r--. 1 root root 64  1月 28 21:43 /opt/MariaDB/key.enc  
# file /opt/MariaDB/key.enc  
/opt/MariaDB/key.enc: data
```

次に、暗号化の設定を設定ファイル/etc/my.cnf.d/server.cnf に追記する。

[mysqld] に追加

```
plugin-load-add=file_key_management.so  
file-key-management  
file_key_management_encryption_algorithm=aes_cbc  
file_key_management_filename = /opt/MariaDB/key.enc  
file_key_management_filekey = secret  
innodb-encrypt-tables  
innodb-encrypt-log
```

## 4. MariaDB の起動

次のコマンドで起動

```
# systemctl enable mariadb  
# systemctl start mariadb  
# mysql  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 3  
Server version: 10.1.21-MariaDB MariaDB Server  
  
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB[(none)] >
```

データベース test に暗号化テーブルを作成する

```
MariaDB [test]> CREATE TABLE TEST_ENCRYPTION(
```

```
-> TEST_ID INT PRIMARY KEY,  
-> TEST_VAR VARCHAR(80))  
-> ENGINE=InnoDB ENCRYPTED=YES ENCRYPTION_KEY_ID=1;  
Query OK, 0 rows affected (0.10 sec)
```

テストデータを挿入する

```
MariaDB [test]> INSERT INTO TEST_ENCRYPTION VALUES(1, "test");  
Query OK, 1 row affected (0.10 sec)  
MariaDB [test]> INSERT INTO TEST_ENCRYPTION VALUES(2, "test");  
Query OK, 1 row affected (0.02 sec)
```

テストデータの確認

```
MariaDB [test]> SELECT * FROM TEST_ENCRYPTION;  
+-----+-----+  
| TEST_ID | TEST_VAR |  
+-----+-----+  
|      1 | test     |  
|      2 | test     |  
+-----+-----+  
2 rows in set (0.00 sec)
```

次に、テーブルのデータが格納されているファイルを確認してみる。このファイル内で暗号化がされていれば良い。

```
# strings /var/lib/mysql/test/TEST_ENCRYPTION.ibd  
ycz^  
F{\c  
s?`a|  
"p"9  
]u=0)  
v%~i  
'FVvkt  
wW?)  
>Z^m  
-}i/  
p&x\  
cnME  
Rr{@  
IUy[  
>T}im#mv!S  
*``_  
kM^;  
...
```

かなり長いファイルになっているが、暗号化はされている。

以上、MariaDB 10.1 のインストールからテーブルの暗号化までの手順を示した。