

概要

Redmineのメンテナンスの中で、「データベースのバックアップ」は非常に重要なものです。

そこで、改めて、シェルスクリプトとCronによるバックアップ手順を整理しました。

動作を確認した環境

- Ubuntu 20.04 LTS
- Redmine 4.2
- Redmine 5.0
- MySQL 8.0.32

実施前提

- MySQLに管理者権限でログインできること。
- Redmine用のDBとDBユーザ、DBパスワードを把握していること。
- また、DBサーバはローカルホストです。

確認した手順

- Redmineが稼働しているUbuntuサーバのターミナル上での操作です。
- MySQL以外は全て一般権限で実行します。

さっくりとした手順

1. Redmineのデータベースユーザに権限を付与します。
2. バックアップディレクトリを作成します。
3. アカウントファイルを作成します。
4. バックアップスクリプトを作成します。
5. crontabに登録します。

データベース設定

管理者権限でMySQLにログインします。

```
mysql -u root -p
```

データベースのユーザ権限を変更します。

これを行わないとDump処理ができませんでした。

```
GRANT RELOAD ON *.* TO '【RedmineのDBユーザ】'@'localhost';  
FLUSH PRIVILEGES;  
EXIT
```

ディレクトリとスクリプト作成

バックアップディレクトリ作成

```
sudo mkdir -p /home/backup/mysql  
# 運用に合わせて指定ください。ファイルサーバや別パーティションにマウントしている方がサーバ事態の障害発生でも冗長化を持たせられます。
```

```
sudo chown -R hoge:hoge /home/backup/mysql  
# ディレクトリの所有者をログインユーザに修正します
```

```
cd /home/backup/mysql&& pwd  
# 指定したディレクトリに移動します
```

アカウントファイル作成

このファイルを作成しないと、「安全ではない」とエラーが出ます。

以下の内容を教義・信仰に沿ったエディタで作成します。(【】内は取り除き、自分の設定に合わせます)

- アカウントファイル内容
 - ファイル名:account.txt

```
[client]
user = 【RedmineのDBユーザ】
password = "【RedmineのDBユーザ用パスワード】"
```

アカウントファイルでアクセスできることを確認

```
mysql --defaults-extra-file=account.txt
```

MySQLのプロンプトが出れば成功です。exitで抜けます。

スクリプト作成

以下の内容を教義・信仰に沿ったエディタで作成します。

- スクリプト内容
 - スクリプト名:mysql_daily_backup.sh

```
#!/bin/bash

# スクリプトの実行日時を指定します
date=$(date "%Y%m%d")

# バックアップディレクトリを定義します
# 上記手順で示したディレクトリを指定してください
backup_dir="/home/backup/mysql"

# アカウントファイルを指定します
credentials_file="$backup_dir/account.txt"

# バックアップ時に指定するオプションを定義します
options="--defaults-extra-file=$credentials_file --no-tablespaces --single-transaction"

# バックアップファイル名を定義します
backup_file="$backup_dir/redmine.sql.$date.gz"

# バックアップを実行し、.gz形式でバックアップをします
mysqldump $options -h localhost redminegzip > $backup_file

# 10世代前の圧縮ファイルを削除します(運用に合わせて指定ください)
find $backup_dir -type f -name "redmine.sql.*.gz" -mtime +10 -delete
```

- 実行権限の付与

```
chmod +x mysql_daily_backup.sh
```

動作確認

```
sh mysql_daily_backup.sh
```

以下を確認します。

- エラーなく実行できること
- redmine.sql.実行日付.gz形式でファイルが作成されること
- gunzip redmine.sql.実行日付.gzでファイルが解凍できること

Crontab設定

Cron登録

```
crontab -e
```

登録内容例

```
0 0 * * * /home/backup/mysql/mysql_daily_backup.sh  
# 実行時刻、頻度などは自分の運用形態に合わせてます。
```

Cron登録確認

```
sudo ta -20 /var/log/cron.log
```

操作時刻に

- BEGIN EDIT
- REPLACE
- END EDIT

が表示されれば設定は完了です。

動作確認日

2023/02/08

関連記事

<https://atelier.reisalin.com/projects/zettel/knowledgebase/articles/43>

こちらは上記スクリプトにパスワードによる暗号化付与を施したバックアップスクリプトです。

ファイル

Redmine運用-2-.jpeg	207 KB	2024/01/05	手動人形
-------------------	--------	------------	------