

## 01\_Linuxサーバデータベース - Redmineのファイル一式の日次バックアップ。

### 概要

Redmineのメンテナンスの中で重要となるDBのバックアップは記載しました。

ここでは、それ以外のファイル一式をバックアップするスクリプトを作成することで不測の事態に備えます。

### バックアップ対象となるファイル群

基本的に、以下のファイル群が残っていれば復旧は(理論上)可能です。

- /Redmine格納ディレクトリ/plugins配下一式
- /Redmine格納ディレクトリ/files配下一式
- /Redmine格納ディレクトリ/public/themes配下一式
- /Redmine格納ディレクトリ/config/database.yml
- /Redmine格納ディレクトリ/config/configuration.yml (メール設定などで設定している場合)
- /Redmine格納ディレクトリ/config/additional\_environment.rb (プラグインなどで追記している場合)

### 本記事で扱うこと

1. 先に述べたバックアップ対象となるファイル群の定期バックアップを行うシェルスクリプトを作成します。
2. バックアップ時、パスワードが書かれているコンフィグに関してはその箇所をマスクします。(これもスクリプトに組み込みます)
3. 動作を確認し、cronに設定して日次でバックアップを行います。

### 動作を確認した環境

- Ubuntu 20.04 LTS
- Redmine 4.2

### 実施前提

- 利用しているRedmine環境の容量を確認し、バックアップ先に十分な空き容量があることを確認してください。
- 本スクリプトは「全てのデータを一時的なバックアップディレクトリにコピーした上で圧縮し、そのディレクトリは削除する」処理を取っています。その容量も加味してください。
- バックアップ元(Redmine格納ディレクトリ)の所有者が全てRedmine実行ユーザ(通例はwww-data)となっていることを確認してください。
- また、本記事において、Redmineの格納ディレクトリは/home/www-data/redmineと一般的な構成と異なっております。

### 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。

### さっくりとした手順

1. バックアップディレクトリを作成します。
2. バックアップスクリプトを作成します。(環境に合わせて修正します)
3. crontabに登録します。

### バックアップディレクトリ作成

```
sudo mkdir -p /home/backup/redmine
# 運用に合わせて指定ください
```

```
sudo chow -R www-data:www-data /home/backup/redmine
# Redmineディレクトリの所有者に設定します
```

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

### リストア方法のテキストファイル作成

バックアップから切り戻したとき、作業者はかなり焦るものです。そこで、簡単な手順をバックアップファイル一式に配置しておけばスムーズな復旧を行うことができます。

以下の内容を教義・信仰に沿ったエディタで作成します。(ここでは筆者が用いているテキスト内容です。必要に応じて修正してください)

- テキスト内容
  - テキスト名:how\_to\_restore.md
  - テキストの所有者はスクリプトの実行者と同じ(またはroot)である必要があります。

```
# Redmine復旧方法
```

```
## 前提環境
```

- Ubuntu 20.04系サーバ
- MySQL 8.3以上
- Apache 2.4系
- Redmineの実行ユーザはデフォルトのwww-data

```
## 【リストア/移行方法】
```

```
### 移行先にRedmineを作成
```

1. 新たにRedmineサイトを立ち上げます。(移行元と移行先のバージョンは合わせます)
2. リストア先/移行先のdb名/dbユーザはバックアップ元と同じにします。
3. バックアップされたdatabase.ymlを /redmine/config/配下に上書きします。dbパスワードを設定してください。
4. この状態で、Redmineのデータマイグレーションまで行います。

```
### バックアップされたファイル式の再配置
```

- files/ディレクトリー式は /redmine/配下に上書きしてください。
- plugins/ディレクトリー式は /redmine/配下に上書きしてください。
- themesディレクトリー式は /redmine/public/配下に格納してください。
- configuration.yml : (あるなら)/redmine/config/配下に上書き。メールパスワードを設定してください。
- additional\_environment.rb : (あるなら)/redmine/config/配下に上書きします。

それぞれのディレクトリ/ファイルの所有者を「www-data」にします。

```
sudo chown -R www-data:www-data /redmine格納ディレクトリ/  
> 例 sudo chown -R www-data:www-data /var/lib/redmine
```

```
### プラグインのDBマイグレーション
```

1. redmineを配置したディレクトリに移動します。(例; cd /var/lib/redmine/)
2. 以下のコマンドを発行して、プラグインのマイグレーションを行います。

```
sudo -u www-data bundle exec rake redmine:plugins:migrate RAILS_ENV=production
```

```
### プラグインマイグレーション後のApache再起動
```

1. 以下を実施してWebサービスを再起動します。

```
sudo systemctl restart apache2.service
```

```
### DBのリストア
```

1. 別途、バックアップしたRedmineのsqlファイルを移行先サーバの任意のディレクトリに転送します。
2. 以下のコマンドを発行して、DBをリストアします。

```
mysql -u redmineのdbユーザ -p redmineのdb名 < バックアップしたsql  
> 例 mysql -u redmine -p redmine < redmine_backup
```

```
### DBリストア後のApache再起動
```

1. 以下を実施してWebサービスを再起動します。

```
sudo systemctl restart apache2.service
```

### ### 動作確認

移行先(または復旧した)Redmineのホストにブラウザでアクセスし、バックアップした時の状態になっているかを確認します。

### 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:redmine\_daily\_backup.sh
  - 実行ユーザ:Redmineの所有者 (通例はwww-dataです)

```
#!/bin/bash
```

```
### ここからはスクリプトの変数を定義します。""の記述は自身の環境に合わせて修正ください。 ###
# バックアップ先のディレクトリを指定します。
backup_dir="/home/backup/redmine"
# バックアップ元のRedmineが格納されているディレクトリを指定します。
redmine_dir="/home/www-data/redmine"
# ファイルに付与する日付/作業ディレクトリ名/バックアップファイル名を指定します。
current_date=$(date +%Y%m%d)
backup_name="redmine_backup_${current_date}"
zip_file="redmine_backup.${current_date}.zip"
# 保存する日数を「日」で指定します。(ここでは3日にします。)
retention_period=3
### 変数はここまでです ###

# 一時的なバックアップディレクトリを作成します。
mkdir "${backup_dir}/${backup_name}"

# Redmineのユーザーデータ/プラグイン/テーマをバックアップディレクトリにコピーします。
cp -R "${redmine_dir}/plugins" "${backup_dir}/${backup_name}"
cp -R "${redmine_dir}/files" "${backup_dir}/${backup_name}"
cp -R "${redmine_dir}/public/themes" "${backup_dir}/${backup_name}"

# マイグレーション時に必要となるdatabase.ymlをコピーします。この時、パスワードが書かれている行をマスクします。
cp $redmine_dir/config/database.yml $backup_name
sed -i 's/password:*/password: "type your db passw $backup_name/database.yml

# メール設定などでconfiguration.ymlを設定している場合、これもコピーします。(存在しない場合はコピーしません)
# 同様にパスワードが書かれていたらその行はマスクします。
if [ -f $redmine_dir/config/configuration.yml ]; then
  cp $redmine_dir/config/configuration.yml $backup_name
  sed -i 's/password:*/password: "type your passwo $backup_name/configuration.yml
fi

# プラグイン設定などでadditional_environment.rbを設定している場合、これもコピーします。(存在しない場合はコピーしません)
)
if [ -f $redmine_dir/config/additional_environment.rb ]; then
  cp $redmine_dir/config/additional_environment.rb $backup_name
fi

# スクリプトと同じディレクトリにあるhow_to_restore.mdをバックアップディレクトリに流し込みます。(存在しない場合はコピーしません)
if [ -f $backup_dir/how_to_restore.md ]; then
  cp $backup_dir/how_to_restore.txt $backup_name/how_to_restore.md
fi

# バックアップディレクトリをzip形式で圧縮します。
cd "${backup_dir}"
zip -r "${zip_file}" "${backup_name}"

# 一時的なバックアップディレクトリを削除します。
rm -rf "${backup_dir}/${backup_name}"

# 上記retention_periodで指定した日数前のバックアップしたzipファイルを削除します。
find "${backup_dir}" -name "redmine_backup.*.zip" ! -type -newermt "${retention_period} days a{ -delete
```

- 実行権限の付与

```
chmod +x redmine_daily_backup.sh
```

## 動作確認

```
sudo -u www-data bash redmine_daily_backup.sh  
# 管理者権限で実行する場合は sudo bash redmine_daily_backup.sh
```

以下を確認します。

- エラーなく実行できること
- redmine\_backup.実行日付.zip形式でファイルが作成されること
- sudo -u www-data unzip redmine\_backup.実行日付.zipでファイルが解凍されディレクトリに移動できること
- ymlファイルのパスワードが「type your (db) password」と本来のパスワードが上書きされていること
- 設定したhow\_to\_restore.mdが回答したディレクトリの中にあり、参照できること

## Crontab設定

### Cron登録

```
sudo crontab -e -u www-data  
sudo crontab -e -u www-data  
# 管理者が実行する場合は sudo crontab -e -u root
```

### 登録内容例

```
5 0 * * * /home/backup/redmine/redmine_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

## ファイル

---

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