

# バックアップサービス利用手順書

## 第 2.1 版

2020 年 4 月 20 日  
日本医師会 ORCA 管理機構

## 目次

1	更新履歴 .....	2
2	概要 .....	3
3	動作環境 .....	3
4	事前設定 .....	3
4.1	証明書および DB バックアップ用 API キーの準備 .....	3
4.2	パッケージのインストール .....	3
4.3	設定 .....	4
5	バックアップの実行 .....	6
6	ログの確認 .....	7
7	オンプレ環境に復元手順 .....	7
8	(参考) logwatch による監視 .....	10
8.1	logwatch インストール .....	10
8.2	設定 .....	10

# 1 更新履歴

---

- 2018年2月22日第1版
- 2019年7月10日第2版  
クライアント証明書が統合されたので内容を更新
- 2020年4月20日第2.1版  
VPN 接続手順を修正

## 2 概要

---

本資料は、バックアップサービスを利用するために行う手順を以下に記述する。

## 3 動作環境

---

バックアップサービスは以下の環境で動作可能である。

- Ubuntu 16.04、日レセ 5.0.0
- Ubuntu 16.04、日レセ 5.1.0
- Ubuntu 18.04、日レセ 5.1.0

## 4 事前設定

---

### 4.1 証明書および DB バックアップ用 API キーの準備

CA 証明書(ca.crt)およびクライアント証明書(cert)、秘密鍵(pem)を準備する。  
システム管理サイトより DB バックアップ用 API キーをメモしておく。

### 4.2 パッケージのインストール

```
$ sudo apt-get install dbbackup-client
```

## 4.3 設定

1. 「4.1 証明書および DB バックアップ用 API キーの準備」で用意した証明書を `/etc/dbbackup-client/certs/` 以下に配置する。`/var/lib/dbbackup-client/config.yml` にファイル名は合わせる。

- `cert: /etc/dbbackup-client/certs/dbbackup.crt`
- `ca: /etc/dbbackup-client/certs/ca.crt`
- `cert_key: /etc/dbbackup-client/certs/dbbackup.pem`

```
$ sudo cp ca.crt /etc/dbbackup-client/certs/ca.crt

$ sudo cp 000052__JP_u00000014_client21.crt /etc/dbbackup-client/certs/dbbacku
p.crt

$ sudo cp 000052__JP_u00000014_client21.pem /etc/dbbackup-client/certs/dbbacku
p.pem
```

※上記の例では、

クライアント証明書のファイル名：`000052__JP_u00000014_client21.crt`

秘密鍵のファイル名：`000052__JP_u00000014_client21.pem`

としています。提供されるファイル名は医療機関間毎に異なりますので、読み替えてください。

2. `/var/lib/dbbackup-client/config.yml` の記載を変更する。

`dbbackup_api_key:` に「4.1 証明書および DB バックアップ用 API キーの準備」でメモした DB バックアップ用 API キーを記載する。

(例)

`dbbackup_api_key: 03c655cbefd34448acfbbb551ef3371190df85a0fdfe2ed3970079e64c8df05f`

※コロンと API キーの間に半角スペース 1 文字入れること

`wal_dir:`

Ubuntu16.04・・・`wal_dir: /var/lib/postgresql/9.5/main/pg_xlog`

Ubuntu18.04・・・`wal_dir: /var/lib/postgresql/10/main/pg_wal`

### 3. postgresql の設定を変更する。

```
Ubuntu18.04 : /etc/postgresql/10/main/postgresql.conf
```

```
Ubuntu16.04 : /etc/postgresql/9.5/main/postgresql.conf
```

の変更

```
#wal_level = minimal                # minimal, archive, or hot_standby

↓ #を削除、archive に変更

wal_level = archive

#max_wal_senders = 0                # max number of walsender processes

↓ #を削除、1 に変更

max_wal_senders = 1                # max number of walsender processes
```

Ubuntu16.04 のみ

```
/etc/postgresql/9.5/main/pg_hba.conf
```

の変更

```
#local  replication  postgres                                peer

↓ 有効にする

local  replication  postgres                                peer
```

postgrsql および jma-receipt を再起動する。

```
$ sudo service postgresql restart

$ sudo service jma-receipt restart
```

## 5 バックアップの実行

---

以下の手順でフルバックアップをおこない、アップロードをおこなう。

### <留意事項>

手順では実行コマンドのみを示します。運用に応じて、バックアップ用スクリプトを作成する、電源終了時のスクリプトに組み込む等を検討してください。

### 1 VPN 接続をおこなう。

「ORCA VPN(ソフト型) Ubuntu 接続手順書」もしくは「ORCA VPN (ルータ型) サービス 設置・操作マニュアル」に則って、VPN 接続をおこなってください。

(例) ソフト型の場合は以下のコマンドを実行する。

```
$ sudo ipsec up gam
$ sudo service xl2tpd restart
```

### 2 VPN 通信確認をおこなう。

```
$ ping sms.orca.orcamo.jp
```

ping 疎通ができない場合は VPN 接続を見直してください。

※VPN 接続が完了するまで数分程度、時間を要する場合があります。

### 3 バックアップコマンドを実行する。

```
$ sudo -u postgres /usr/bin/ginbee-dbbackup --full --work_dir=/var/lib/dbbackup-client/
```

成功時は終了コード 0 でコンソールには出力はされない。ログには出力される。

(/var/log/dbbackup-client/dbbackup-client.log)

エラー時は終了コード 0 以外で以下のような出力がされる。またログにもエラーログが出力される。

```
/usr/lib/dbbackup-client/lib/ginbee/dbbackup/uploader.rb:75:in `send_f
```

```
ile': send backup file failed: file=</var/lib/dbbackup-client/ginbee_s  
plit_base_backup/split.00000005.gz>,  
http_response=<#<HTTP::Message:0x00000001a91b08 @http_header=#  
(略)
```

4 VPN 切断をおこなう。

(例) ソフト型の場合は以下のコマンドを実行する。

```
$ sudo service xl2tpd stop
```

## 6 ログの確認

---

ログについては `/var/log/dbbackup-client/dbbackup-client.log` を確認

## 7 オンプレ環境に復元手順

---

### ▲注意

バックアップした環境とリストアする環境の OS および日レセバージョンが一致している必要があります。

1 「4 事前設定」を参考に設定をおこなう。

※「4.3 設定」をおこなう前に、`/etc/cron.d/dbbackup-client` の設定がコメントアウトされていることを確認してください。

2 VPN 接続をおこなう。

「ORCA VPN(ソフト型) Ubuntu 接続手順書」もしくは「ORCA VPN (ルータ型) サービス 設置・操作マニュアル」に則って、VPN 接続をおこなってください。



(例) ソフト型の場合は以下のコマンドを実行する。

```
$ sudo service xl2tpd start
```

※ソフト型の場合、初期設定では OS 起動時に VPN 接続されます。

バックアップ時のみ VPN 接続をおこないたない場合は、サービス設定コマンドである update-rc.d コマンド等で、VPN 接続サービスを起動しないようにしてください。

```
$ sudo update-rc.d -f xl2tpd remove
```

3 バックアップしたファイルをダウンロードします。ダウンロードするフォルダで以下を実行します。

※作業ディレクトリとして/var/tmp/postgres を利用しています。任意に変更しても構いません。

```
$ mkdir /var/tmp/postgres
```

```
$ sudo chown postgres.postgres /var/tmp/postgres
```

```
$ cd /var/tmp/postgres
```

```
$ sudo -u postgres ginbee-dbbbackup-download --work_dir=/var/lib/dbbackup-client/
```

ls で以下のファイルがダウンロードされていることを確認します。

```
$ ls
```

```
base.backup.tar.gz
```

```
concat.sha1
```

4 VPN 切断をおこなう。

(例) ソフト型の場合は以下のコマンドを実行する。

```
$ sudo service xl2tpd stop
```

5 リストアは以下の手順でおこないます。

## jma-receipt、PostgreSQL 停止

```
$ sudo service jma-receipt stop  
$ sudo service postgresql stop
```

## postgres ユーザになり postgresql のデータディレクトリに移動

```
$ sudo su postgres  
$ cd /var/lib/postgresql/??/  
# (??はUbuntu 18.04 では10、 Ubuntu 16.04 では9.5)
```

## 元のファイルを退避

```
$ mv main main.bak
```

## ダウンロードしたファイルをリストア

```
$ mkdir main  
$ chmod 700 main  
$ tar xvzf /var/tmp/postgres/base.backup.tar.gz -C main
```

## recovery.conf の用意

```
$ cd /var/lib/postgresql/??/main/  
# (??はUbuntu 18.04 では10、 Ubuntu 16.04 では9.5)  
$ echo "restore_command = ''" > recovery.conf  
$ exit
```

## Jma-receipt、PostgreSQL の起動

```
$ sudo service postgresql start  
$ sudo service jma-receipt start
```

## 8 (参考) logwatch による監視

---

バックアップが失敗した場合に通知させる方法は複数あるが、ここでは logwatch を使用しての通知方法を紹介する。

logwatch は変更可能だが 1 日 1 回 cron により実行してレポートを出力することを想定している。

### 8.1 logwatch インストール

```
$ sudo apt-get install logwatch
```

### 8.2 設定

パッケージをインストールすると /etc/logwatch/conf が作成されているので、ここに設定を追加していく。

#### 1 サービスのエントリ作成

dbbackup-client を監視することを宣言する。「LogFile」で指定した名前がログファイルの設定名になる。

/etc/logwatch/conf/services/dbbackup-client.conf に以下の設定を作成

```
Title = dbbackup-client
```

```
LogFile = dbbackup-client
```

## 2 LogFile で指定した名前 (dbbackup-client)+.conf でログファイル設定の作成

/etc/logwatch/conf/logfiles/dbbackup-client.conf

```
LogFILE = /var/log/dbbackup-client/dbbackup-client.log
```

```
Archive = /var/log/dbbackup-client/dbbackup-client.log.*
```

## 3 ログ監視スクリプトの作成

スクリプトファイルはログファイルを読み取って標準出力に出力が出来る実行ファイルであれば言語は問わない。

ここでは **Bash** スクリプトを使用して作成する

/etc/logwatch/scripts/services/dbbackup-client

```
#!/bin/bash

OLD_IFS=$IFS

IFS=$'\n'

rm_date=''

if [[ "${LOGWATCH_DATE_RANGE}" =~ "today" ]]; then

    rm_date=`date '+%Y-%m-%d'`

elif [[ "${LOGWATCH_DATE_RANGE}" =~ "yesterday" ]]; then

    rm_date=`date --date '1 days ago' '+%Y-%m-%d'`

fi
```

```
for LINE in $( cat /dev/stdin ); do

    if echo $LINE|egrep "^E, \[${rm_date}]" &> /dev/null; then

        echo ${LINE}

    fi

done

IFS=$OLD_IFS
```

以上のファイルを用意すれば、通常の `logwatch` が動作する。通知先変更等は別途設定が必要になる。

#### 4 動作確認

`/var/log/dbbackup-client/dbbackup-client.log` でのエラーは以下のような形式になっている。

```
I, [2018-02-09T07:02:06.759671 #4043] INFO -- : send backup file success: file=</var/lib/dbbackup-client/ginbee_split_base_backup/split.00000004.gz>, http_status=<200>

E, [2018-02-09T07:02:07.401292 #4043] ERROR -- : send backup file failed
```

`/var/log/dbbackup-client/dbbackup-client.log` でのエラーは以下のような形式になっている

手動でコマンドを実行することで動作確認することが出来る。上記のログが実行する日付と同じである場合は、以下のように実行する

```
$ sudo /usr/sbin/logwatch --output stdout --service dbbackup-client --range today
```

該当するエントリがあれば以下のようなレポートが出力される。該当するエントリがなかった場合は何も出力されない。

```
##### Logwatch 7.4.0 (05/29/13) #####

Processing Initiated: Fri Feb 9 07:30:06 2018

Date Range Processed: today

                        ( 2018-Feb-09 )

                        Period is day.

Detail Level of Output: 0

Type of Output/Format: stdout / text

Logfiles for Host: ubuntu-1404

#####

----- dbbackup-client Begin -----

E, [2018-02-09T07:02:07.401292 #4043] ERROR -- : send backup file fa
iled

----- dbbackup-client End -----

##### Logwatch End #####
```

## 5 通知先設定

デフォルトの実行は `/etc/cron.daily/00logwatch` に書かれていて、この場合1日1回実行される。前日 (`yesterday`)のログを対象とする。

## /etc/cron.daily/00logwatch

```
#!/bin/bash

#Check if removed-but-not-purged

test -x /usr/share/logwatch/scripts/logwatch.pl || exit 0

#execute

/usr/sbin/logwatch --output mail

#Note: It's possible to force the recipient in above command

#Just pass --mailto address@a.com instead of --output mail
```

この場合は他のサービスについてもレポートが出力される。もし **dbbackup-client** に限定する場合は **--service dbbackup-client** を追加する。

```
usr/sbin/logwatch --output mail

↓

/usr/sbin/logwatch --output mail --service dbbackup-client
```

また当日のログを監視したい場合は **--range today** を追加する。

(その場合 **/etc/crontab** の **cron.daily** の実行時間に注意すること)

上記はメールで **root** 宛に送信されるので、別途メールの設定をする必要がある。

`logwatch` の通知先はメールやファイル出力等可能なので、`man logwatch` や `--help` を参照しながら環境に合わせて設定する。