FUJITSU Hybrid IT Service FJcloud-O IaaS

データベースサービス
ユーザーズガイド

V1.20

2025年3月17日

富士通株式会社

・ 本資料の無断複製、転載を禁じます。

・ 本資料は仕様変更等により予告なく内容を変更する場合がございます。予めご注意願います。

K5IA-DC-M-012-001J

まえがき

本書の目的

本書は、FUJITSU Hybrid IT Service FJcloud-O IaaS（以降、IaaS）が提供する IaaS データベースサービス（以降、データベースサービス）を構築、運用するための基本的な操作手順を示しています。

本書は東日本第 1／東日本第 2／西日本第 1／西日本第 2 リージョンを対象としています。

本書の読者

本書は、データベースサービスを運用される方を対象としています。

なお、本書は、以下についての一般的な知識があることを前提に書かれています。

- IaaS の サービスの操作
- PostgreSQL
- Linux

マニュアル体系

目的・用途に合わせて、以下の関連マニュアルもお読みください

マニュアル名称	目的・用途
機能説明書	本サービスが提供する機能詳細を解説した資料です。
API ユーザーズガイド	REST API の使い方について、API 実行環境の構築方法、利用シーケンスにあわせたサンプルスクリプトなどを説明した資料です。
API リファレンスマニュアル	REST API を利用する際の詳細リファレンスとしてご参照・ご利用ください。
IaaS ポータルユーザーズガイド	IaaS ポータル(Web GUI)を使用して、本サービスが提供する機能の使い方を説明した資料です。
HEAT テンプレート解説書	オーケストレーション機能を利用する際に記述する、Heat Orchestration Template (HOT)の書式について説明した資料です。
PostgreSQL 9.6.2 文書	文中で参照するドキュメント「PostgreSQL 9.6.2 文書」については、http://software.fujitsu.com/jp/manual/manualindex/p17000156.html「PostgreSQL 9.6.2 文書」を参照してください。
PostgreSQL 9.2.4 文書	文中で参照するドキュメント「PostgreSQL 9.2.4 文書」については、http://software.fujitsu.com/jp/manual/manualindex/p14000255.html「PostgreSQL 9.2.4 文書」を参照してください。
Enterprise Postgres のマニュアル	文中で参照するドキュメント「Enterprise Postgres のマニュアル」について、

	サーバについては http://software.fujitsu.com/jp/manual/manualindex/p17000156.html クライアントについては Enterprise Postgres Client に同梱されている 「FUJITSU Enterprise Postgres 9.6 導入ガイド(クライアント編)」 を参照してください。
Symfoware Server のマニュアル	文中で参照するドキュメント「Symfoware Server のマニュアル」について、 サーバについては http://software.fujitsu.com/jp/manual/manualindex/p14000255.html クライアントについては Symfoware Server Client に同梱されている 「Symfoware Server 導入ガイド(クライアント編)」 を参照してください。
データベースユーザーズガイド（本書）	本サービスが提供するデータベースサービスの基本的な操作方法を説明した資料です。

商標

- Adobe、Adobe ロゴ、Acrobat、および Reader は、Adobe Systems Incorporated の米国またはその他の国における商標または登録商標です。
- Apache、Tomcat は、The Apache Software Foundation の米国およびその他の国における商標または登録商標です。
- Microsoft、Windows、および Windows Server は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。
- Linux®は米国及びその他の国における Linus Torvalds の登録商標です。
- Red Hat、Red Hat Enterprise Linux は米国およびその他の国において登録された Red Hat, Inc.の商標です。
- UNIX は、米国およびその他の国におけるオープン・グループの登録商標です。

- そのほか、本書に記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

なお、本書では、システム名または製品名に付記される登録表示（™または®）は、省略しています。

輸出管理規制

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

お願い

- 本書は、予告なしに変更されることがあります。
- 本書を無断で他に転用しないようお願いします。

- 本書に記載されたデータの使用に起因する第三者の特許権およびその他の権利の侵害については、当社はその責を負いません。

変更履歴

版数	更新日	変更箇所	概要
1.0	2017 年 6 月 16 日	全体	初版作成
1.0.1	2017 年 8 月 7 日	図 1.2 マルチ AZ のシステム構成図	誤記修正
1.1	2017 年 12 月 4 日	付録 B サポートしているクライアント	サポートする DB エンジンのクライアントのバージョンを更新
1.2	2017 年 12 月 15 日	全体	DB エンジンを追加
1.3	2018 年 3 月 1 日	付録 E：各 API を実行可能なロール	DB インスタンスに対する操作のキャンセルを追加
1.4	2018 年 6 月 29 日	付録 A：変更できる DB パラメータの一覧	仕様変更
1.5	2018 年 7 月 31 日	3.4 性能監視 付録 A：変更できる DB パラメータの一覧	説明追加
1.6	2018 年 9 月 25 日	3.4 性能監視	説明追加
1.7	2018 年 11 月 22 日	付録 A：変更できる DB パラメータの一覧 付録 I：リージョン内のリードレプリカの作成	説明追加
1.8	2019 年 3 月 20 日	2.1 事前準備	注意事項追記
1.9	2019 年 8 月 22 日	4.1 フレーバーID の変更 4.3 シングル構成からマルチ構成への変更 付録 I：制限事項・注意事項一覧	注意事項追記
1.10	2019 年 10 月 16 日	2.2 DB インスタンスの作成 付録 I：DB インスタンスの状態	DB インスタンス状態一覧の追加
1.11	2019 年 12 月 16 日	まえがき 4.1 フレーバーID の変更 付録 I：制限事項・注意事項一覧	海外リージョン記事案の削除
1.12	2020 年 3 月 17 日	付録 J：制限事項・注意事項一覧 No.4、No.5、No.12	説明修正
1.13	2020 年 4 月 15 日	2.5 性能検証	性能検証の記事追加
1.14	2020 年 7 月 16 日	付録 J：制限事項・注意事項一覧 No.20	注意事項追加
1.15	2020 年 8 月 20 日	付録 J：制限事項・注意事項一覧 No.8 東日本リージョン 2 におけるフレーバー変更制限	制限解除に伴う記事削除
1.16	2021 年 11 月 16 日	4.1 フレーバーID の変更 5.1 DB インスタンスの復旧 付録 J：制限事項・注意事項一覧 - No.20	東日本リージョン 2 におけるフレーバーに関する注意を削除 パターン 1、およびパターン 2 への記事追加 注意事項追記
1.17	2023 年 4 月 3 日	2.2 DB インスタンスの作成	DB エンジンの提供終了

		付録 B サポートしているクライアント 付録 I：DB インスタンスの状態 付録 J：制限事項・注意事項一覧 No.21	
1.18	2023 年 9 月 19 日	付録 J：制限事項・注意事項一覧 No.21	制限解除に伴う注意事項削除
1.19	2024 年 9 月 17 日	付録 J：制限事項・注意事項一覧 No.21	注意事項追加
1.20	2025 年 3 月 17 日	付録 J：制限事項・注意事項一覧 No.7	注意事項修正

目次

第 1 章　IaaS データベースサービスの運用

IaaS データベースサービスの運用について説明します。
本マニュアルは下図のようなシステム構成を想定し記載しています。

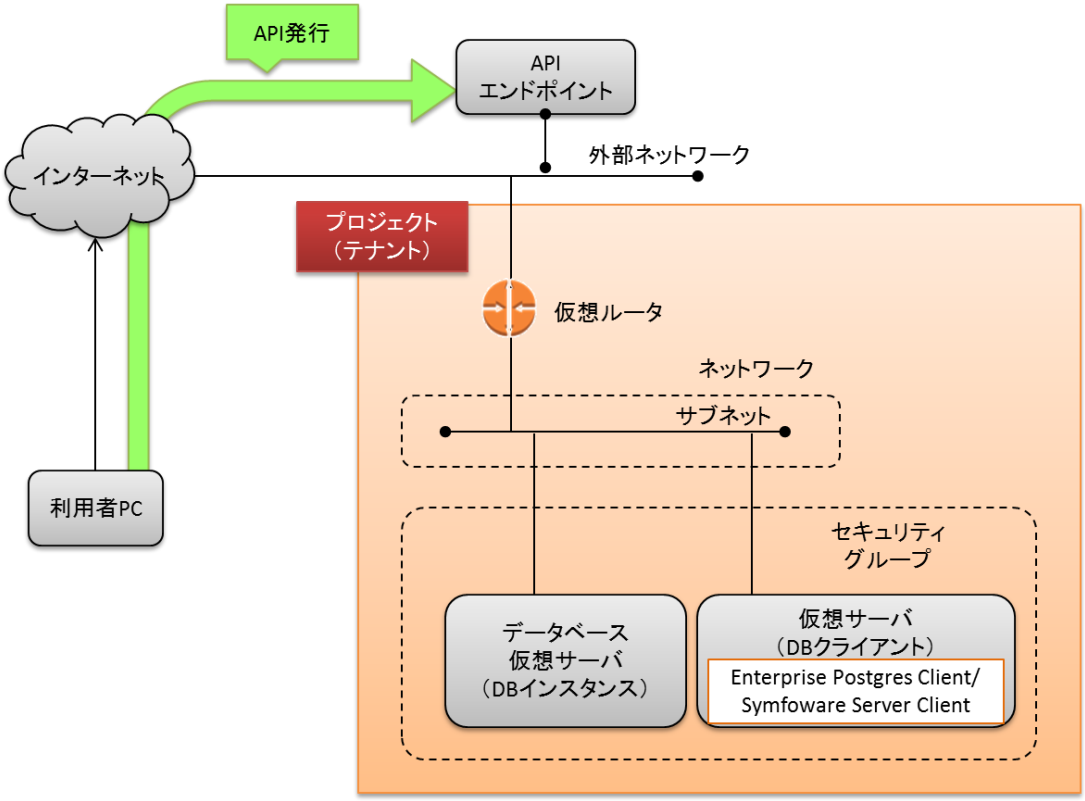


図 1.1　システム構成図

IaaS データベースサービスではデータベース仮想サーバを作成します。データベース仮想サーバとは、データベースが作成される仮想サーバです。以下では、データベース仮想サーバを DB インスタンスと表記します。
DB インスタンスを作成するためには、以下のリソースをあらかじめ作成しておく必要があります。

リソース	説明
ネットワーク／サブネット／IP アドレス	ネットワークは、仮想サーバなどのリソースを配備するためにプロジェクト内に作成します。サブネットは、ネットワークに接続するリソースに対するプライベート IP アドレスの管理、DHCP による IP アドレスの自動設定に必要です。
仮想ルータ	外部ネットワークとネットワークを接続するために作成します。データベース仮想サーバへの接続に必要な FQDN の名前解決や、外部からのインターネット経由接続の場合に使用します。
セキュリティグループ	関連付けられた DB インスタンスのファイアウォールとして動作します。DB インスタンス上のデータベースへの接続には、DB インスタンスへの接続を許可するルールを設定する必要があります。
ネットワークコネクター／コネクターエンドポイント	マルチ AZ 構成の場合に、各 AZ のネットワークを接続するために作成します。
DB サブネットグループ	DB インスタンス配備先のサブネットを定義します。DB サブネットグループを作成するために、上述のサブネットを事前に作成しておく必要があります。

DB パラメータグループ	DB インスタンスに適用できる DB パラメータを設定できます。

［参照］

DB インスタンスは、DB インスタンス作成時に設定した DB サブネットグループに属するサブネットのどれか 1 つに所属します。
DB インスタンス作成時に設定したアベイラビリティゾーンに所属します。

［参照］

DB サブネットグループに属するサブネットは変更できます。ただし DB インスタンスが配備されているサブネットは変更できません。

［参照］

マルチ AZ では下図のようなシステム構成になります。

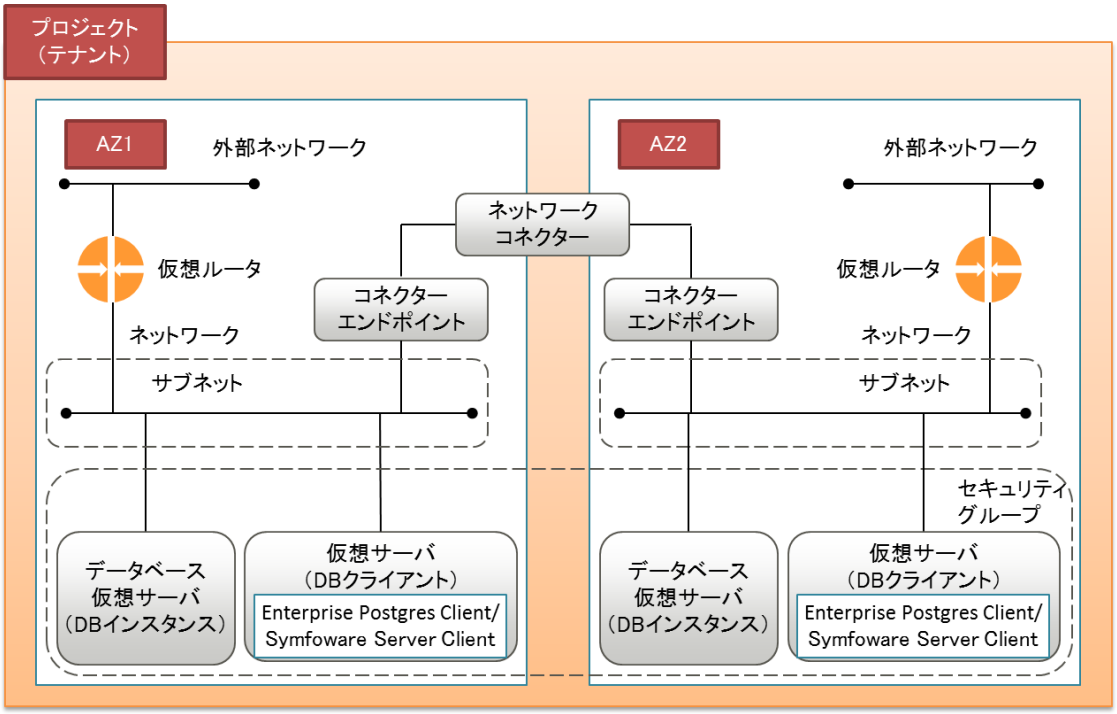


図 1.2　マルチ AZ のシステム構成図

IaaS データベースサービスは REST API を発行して利用します。
REST API を発行する際に、共通に設定するパラメータについて説明します。

・ リクエストヘッダ

Header	説明
X-Auth-Token	ユーザー認証を行った際に取得するトークンを設定する。

・ レスポンスタイプ

Header	説明
Content-Type	POST/PUT の場合、application/json を設定

REST API は、データベースサービスのエンドポイント URL に対して発行します。

・ RESTAPI RequestURI

URL	説明
https://database.[region-name].cloud.global.fujitsu.com/[API_version_information]/[tenant_id]/...	[region-name] リージョン名を設定します。 例： jp-east-1 [API_version_information] API のバージョンを"vx.x"のように設定します。 例： v1.0 [tenant_id] プロジェクト ID を設定します。 例：1234 <エンドポイント URL の例> https://database.jp-east-1.cloud.global.fujitsu.com/v1.0/1234

以下の章では、DB インスタンスの作成と接続、DB インスタンスの運用、DB インスタンスの変更、異常時の対処、DB インスタンスの削除手順について、REST API のコマンド例と合わせて説明します。
本マニュアル中のコマンド例は全て、環境変数 TENANTID にプロジェクト ID が設定されているものとして記載します。

［参照］

※TOKEN 取得について

データベースサービスへ REST API を発行する際は、認証のため、TOKEN を予め取得しておく必要があります。

本マニュアル中のコマンド例は全て、環境変数 TOKEN に予め取得した TOKEN が設定されているものとして記載します。

IaaS データベースサービスの運用は、REST API を実行するユーザーと、データベースの管理者で実施します。
FUJITSU Hybrid IT Service では利用者管理サービスの機能を使用してユーザーにロールを付与することで、権限の管理ができます。IaaS データベースサービスの REST API を実行できるプリセットロールとして、下記の 4 種類があります。

・ IaaS データベースサービスの REST API を実行できるプリセットロール

ロール	実施可能な API
全体管理者ロール	IaaS データベースサービスの全ての API が実行できます。
設計・構築者ロール	IaaS データベースサービスの全ての API が実行できます。
運用者ロール	IaaS データベースサービスの API のうち、資源の追加、削除を除いた運用操作が実行できます。 ただし、DB インスタンスの復旧に必要な API は実行できます。
監視者ロール	IaaS データベースサービスの監視に必要な API が実行できます。

各ロールの API の実行可否については、付録 E に記載します。
ロールの詳細、および、ユーザーにロールを付与する方法については、「機能説明書」の「12.3 利用者管理」の章を参照してください。

第 2 章 DB インスタンスの作成と接続

本章では DB インスタンスの作成からデータベースの接続までの、基本的な操作に関して説明します。
事前に DB インスタンスを配備するサブネット、および、データベースに接続する仮想サーバ（以下、DB クライアントと記載します。）を配備済みであることを前提としています。
本章の内容は、全体管理者ロールまたは設計・構築者ロールで実施します。

2.1 事前準備

DB インスタンス作成の事前準備について説明します。

(1) ネットワーク／サブネットの作成

ネットワーク／サブネットの作成方法は、「 API ユーザーズガイド」の「3.2.1 ネットワーク作成」～「3.2.2 サブネット作成」の章を参照してください。
IP アドレスはサブネット作成時に自動的に作成されます。
ここで作成したサブネットは、2.2 章(1)で DB サブネットグループを作成する際に使用します。
DB インスタンスを配備したい全ての AZ でネットワーク／サブネットを作成してください。

［注意］

DB インスタンス作成後、DB インスタンスが所属するサブネットについて、仮想ネットワークサービスの「Update subnet」API により host_routes の更新を行う場合、更新後、該当の DB インスタンスに反映させるためには、DB インスタンスの再作成が必要です。

［参照］

DB インスタンス作成時、IP アドレスは以下の数だけ消費されます。

サブネット毎の privateIP 消費数

privateIP は、サブネットのネットワークアドレスから消費されます。
サブネット毎の消費 privateIP 数は下記になります。

DB インスタンスの構成	インターネット経由での接続を許可する	インターネット経由での接続を許可しない
シングル	9 個	5 個
AZ 内冗長	10 個	6 個
AZ 間冗長	9 個	5 個

全体の publicIP 消費数

publicIP は、インターネット経由での接続を許可する場合のみ消費されます。

publicIP は、外部ネットワークから消費されます。

消費 publicIP 数は下記になります。

シングル構成の場合：5 個

冗長構成の場合：10 個

※インターネット経由での接続許可について

DB インスタンス作成時、publiclyAccessible パラメータに true を設定すると、外部からのインターネット経由での接続を許可する DB インスタンスを作成できます。publiclyAccessible パラメータについては、「API リファレンス（Application Platform Service 編）」を参照してください。

・・・

(2) 仮想ルータの作成

(1)で作成したサブネットは、以下の要件を満たす必要があります。

外部からのインターネット経由接続の場合：全てのサブネットが外部ネットワークにアタッチされている

マルチ AZ の場合：各サブネットのデフォルトゲートウェイから、他の全てのサブネットの IP に到達するルーティングがある

サブネットが上記の要件を満たすために、仮想ルータを作成します。

仮想ルータの作成方法は、「API ユーザーズガイド」の「3.2.3 仮想ルータ作成」～「3.2.5 仮想ルータの情報変更(サブネットにアタッチ)」の章を参照してください。

DB インスタンスを配備したい全ての AZ で仮想ルータを作成してください。

マルチ AZ の場合は、各 AZ の仮想ルータに、もう一方のサブネット上の IP に到達するルーティングを設定してください。

ルーティングの設定方法は、「API リファレンス（Network 編）」の「1.4.6.8 Update extra route」の章を参照してください。

(3) セキュリティグループの作成

DB インスタンスへの接続を許可するセキュリティグループを作成します。

ここで作成したセキュリティグループは DB インスタンス作成時に DB インスタンスに設定します。

コマンド例
SECURITY_GROUP_NAME=[作成するセキュリティグループ名] ENDPOINT=[Network サービスのエンドポイント] curl -X POST -i ${ENDPOINT}/v2.0/security-groups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group¥": {¥"name¥": ¥"${SECURITY_GROUP_NAME}¥"}}"
実行結果例
$ SECURITY_GROUP_NAME=test-security-group $ ENDPOINT=https://networking.jp-east-1.cloud.global.fujitsu.com

```
$ curl -X POST -i ${ENDPOINT}/v2.0/security-groups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group¥": {¥"name¥": ¥"${SECURITY_GROUP_NAME}¥"}}"
HTTP/1.1 201 Created
（中略）
{"security_group": {"tenant_id": "XXXXXXXX", "description": "", "name": "test-security-group", "security_group_rules": [{"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null, "ethertype": "IPv6", "port_range_max": null, "security_group_id": "XXXXXXXX", "port_range_min": null, "tenant_id": "XXXXXXXX", "id": "XXXXXXXX"}, {"remote_group_id": null, "direction": "egress", "remote_ip_prefix": null, "protocol": null, "ethertype": "IPv4", "port_range_max": null, "security_group_id": "XXXXXXXX", "port_range_min": null, "tenant_id": "XXXXXXXX", "id": "XXXXXXXX"}], "id": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"}}  ★security_group の id を SECGRP_ID とし、(4)で使用します
```

セキュリティグループ作成方法の詳細は、「API ユーザーズガイド」の「3.3.1 セキュリティグループ作成」の章を参照してください。

(4) セキュリティグループのルール作成

DB インスタンスに設定するセキュリティグループに、以下のルールを設定します。

ルールの目的	Direction	Protocol	許可相手※	ポート
DB クライアントからの通信	ingress	tcp	DB クライアント	DB インスタンス作成時に設定するポート番号 (default では 26500)
DB インスタンス内部通信	ingress	tcp	DB インスタンスを配備するサブネット	DB インスタンス作成時に設定するポート番号 (default では 26500)
DB インスタンス内部通信	egress	tcp	DB インスタンスを配備するサブネット	DB インスタンス作成時に設定するポート番号 (default では 26500)

セキュリティグループの接続許可相手には以下の 2 つの形式が選択できます。

- --remote-ip-prefix
  - 接続を許可する相手の CIDR

・ --remote-group-id
  ➢ 接続を許可する相手のセキュリティグループ

コマンド例

```
CLIENT_CIDR=[DB クライアントの CIDR（事前に作成しているものを設定してください）]
SECGRP_ID=[(1)で作成したセキュリティグループの ID]
PORT=[DB インスタンス作成時に設定するポート番号（任意の値を設定してください）]
ENDPOINT=[Network サービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"ingress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${CLIENT_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"
```

実行結果例

```
$ CLIENT_CIDR=XXX.XXX.XXX.XXX/XX
$ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
$ PORT=26500
$ ENDPOINT=https://networking.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"ingress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${CLIENT_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"
HTTP/1.1 201 Created
（省略）
```

以下のコマンドは、2.2 章(1)で作成する DB サブネットグループに所属させる全てのサブネットについて繰り返し実施してください。
※2.2 章では、2 つのサブネットが DB サブネットグループに含まれる例を説明しています。この例では、以下のコマンド例は各サブネットについて実施し、計 2 回実施する必要があります。

コマンド例

```
INSTANCE_CIDR=[DB インスタンスの CIDR（DB インスタンスを配備しようとしているサブネットのネットワーク ID）]
SECGRP_ID=[(1)で作成したセキュリティグループの ID]
PORT=[DB インスタンス作成時に設定するポート番号（任意の値を設定してください）]
ENDPOINT=[Network サービスのエンドポイント]

curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"ingress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${INSTANCE_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"

curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"egress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${INSTANCE_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"
```

実行結果例

```
$ INSTANCE_CIDR=XXX.XXX.XXX.XXX/XX
$ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
$ PORT=26500
$ ENDPOINT=https://networking.jp-east-1.cloud.global.fujitsu.com

$ curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"ingress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${INSTANCE_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"
HTTP/1.1 201 Created
（省略）

$ curl -X POST -i ${ENDPOINT}/v2.0/security-group-rules -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -H "Accept: application/json" -d "{¥"security_group_rule¥": {¥"direction¥": ¥"egress¥", ¥"port_range_min¥": ¥"${PORT}¥", ¥"ethertype¥": ¥"IPv4¥", ¥"port_range_max¥": ¥"${PORT}¥", ¥"remote_ip_prefix¥": ¥"${INSTANCE_CIDR}¥", ¥"protocol¥": ¥"tcp¥", ¥"security_group_id¥": ¥"${SECGRP_ID}¥"}}"
HTTP/1.1 201 Created
（省略）
```

2.2 DB インスタンスの作成

DB インスタンス作成手順について説明します。

(1) DB サブネットグループの作成

AZ1、AZ2 サブネットから、DB サブネットグループを作成します。

ここで作成した DB サブネットグループは、DB インスタンス作成時に DB インスタンスに設定します。

コマンド例

```
SUBNET1=[DB インスタンスを配備する AZ1 サブネットの ID（事前に作成しているものを設定してください）]
SUBNET2=[DB インスタンスを配備する AZ2 サブネットの ID（事前に作成しているものを設定してください）]
SUBNETGROUPID=[DB サブネットグループの ID（任意の値を設定してください）]
SUBNETGROUPNAME=[DB サブネットグループの名前（任意の値を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"subnetgroup¥": {¥"name¥": ¥"${SUBNETGROUPNAME}¥", ¥"id¥": ¥"${SUBNETGROUPID}¥", ¥"subnetIds¥": [{¥"subnetId¥": ¥"${SUBNET1}¥"}, {¥"subnetId¥": ¥"${SUBNET2}¥"}] }}"
```

実行結果例

```
$ SUBNET1=YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY
$ SUBNET2=ZZZZZZZZ-ZZZZ-ZZZZ-ZZZZ-ZZZZZZZZZZZZ
$ SUBNETGROUPID=test-subnetgroup-id
$ SUBNETGROUPNAME=test-subnetgroup-name
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"subnetgroup¥": {¥"name¥": ¥"${SUBNETGROUPNAME}¥", ¥"id¥": ¥"${SUBNETGROUPID}¥", ¥"subnetIds¥": [{¥"subnetId¥": ¥"${SUBNET1}¥"}, {¥"subnetId¥": ¥"${SUBNET2}¥"}] }}"
HTTP/1.1 200 OK
（省略）
```

作成した DB サブネットグループについて、作成が完了したことを確認します。

以下の DB サブネットグループの情報参照 API を実行し、status が Available になっていることを確認します。

コマンド例

```
SUBNETGROUPID=[上記で作成した DB サブネットグループの ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups/${SUBNETGROUPID} -H "X-Auth-Token: ${TOKEN}" -H
```

```
"Content-Type: application/json"
```

実行結果例

```
$ SUBNETGROUPID=test-subnetgroup-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups/${SUBNETGROUPID} -H "X-Auth-Token: ${TOKEN}" -H
"Content-Type: application/json"
HTTP/1.1 200 OK
（省略）
{"subnetgroup":{"name":"test-subnetgroup-name","id":"test-subnetgroup-
id","status":"Available","subnetIds":[{"subnetId":"YYYYYYYY-YYYY-YYYY-YYYY-
YYYYYYYYYYYY"},{"subnetId":"ZZZZZZZZ-ZZZZ-ZZZZ-ZZZZ-ZZZZZZZZZZZZ"}],"created":"YYYY-MM-
DDThh:mm:ssZ","description":null}}
```

(2) DB パラメータグループの作成

DB インスタンスに適用する DB パラメータグループを作成します。

ここで作成した DB パラメータグループが、データベースのパラメータに適用されます。

パラメータの変更については、4.5 章で説明します。

DB パラメータグループを作成する際には、作成する DB インスタンスの DB エンジンバージョンに合わせて、DB パラメータグループファミリを指定します。

engine	engineVersion	DB パラメータグループファミリ
enterprisepostgres	9.6	enterprisepostgres_v9.6
symfoware	12.1	symfoware_v12.1

［参照］

DB インスタンスを作成する際に DB エンジンバージョンを指定しない場合は、デフォルトの DB エンジンバージョンが選択されます。2017 年 11 月時点では、enterprisepostgres 9.6 がデフォルトのため、enterprisepostgres_v9.6 を指定してください。

コマンド例

```
PARAMG_ID=[DB パラメータグループの ID（任意の値を設定してください）]
PARAMG_NAME=[DB パラメータグループの名前（任意の値を設定してください）]
PARAMG_FAMILY=[DB パラメータグループファミリ]
ENDPOINT=[データベースサービスのエンドポイント]
```

```
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"parametergroup¥": {¥"parameterGroupFamily¥": ¥"${PARAMG_FAMILY}¥", ¥"name¥": ¥"${PARAMG_NAME}¥", ¥"id¥": ¥"${PARAMG_ID}¥"}}"
```

実行結果例

```
$ PARAMG_ID=test-paramg-id
$ PARAMG_NAME=test-paramg-name
$ PARAMG_FAMILY="enterprisepostgres_v9.6"
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"parametergroup¥": {¥"parameterGroupFamily¥": ¥"${PARAMG_FAMILY}¥", ¥"name¥": ¥"${PARAMG_NAME}¥", ¥"id¥": ¥"${PARAMG_ID}¥"}}"
HTTP/1.1 200 OK
（省略）
{"parameterGroup":{"id":"test-paramg-id","description":null,"name":"test-paramg-name","parameterGroupFamily":"enterprisepostgres_v9.6"}}
```

(3) DB インスタンスの作成

DB インスタンスを作成します。

API 実行時、設定するオプションについて説明します。

- AZ 名(availabilityZone)
  DB インスタンス作成先の AZ 名として、公開されているアベイラビリティゾーンから任意の一つを選択し、設定してください。
- DB インスタンスの ID(id)／名前(name)
  DB インスタンスの ID と名前に任意の値を設定してください。
- セキュリティグループ ID(securityGroupIds)／DB サブネットグループ ID(subnetGroupId)
  ／DB パラメータグループ ID(parameterGroupId)
  事前に作成したセキュリティグループの ID、DB サブネットグループの ID、DB パラメータグループの ID を設定してください。
- フレーバーID(flavorRef)
  付録 C を参照し、適切なものを設定してください。

- データベースの管理者名／管理者パスワード
  データベースの管理者名と管理者パスワードとして、任意の値を設定してください。
- ポート番号
  データベースのポート番号を設定してください。
  2.1 章(2)でセキュリティグループのルールに設定したポート番号と同じ値を設定してください。
- データディスクのサイズ(volume)
  DB インスタンスのディスクサイズを GB 単位で設定します。10～10240 の任意の値を設定してください。

・ 自動バックアップ保持期間(backupRetentionPeriod)

自動バックアップ保持期間は自動バックアップを保持する世代数を設定してください。

0～10 の任意の値を設定してください。

自動バックアップ保持期間が 1 以上の場合に自動バックアップが実施されます。0 の場合は自動バックアップを行いません。

1 以上を設定した場合は、DB インスタンスの作成後に初回のバックアップが実行されます。バックアップが完了するまで status は Build のままになります。バックアップが完了すると status が Active になります。

・ 自動バックアップの時間帯(preferredBackupWindow)

自動バックアップの時間帯には、日次の自動バックアップが行われる時間帯を設定してください。

上記で設定した時間帯の開始時間になると、自動バックアップが開始されます。

データ量によっては、バックアップ作成が上記で設定した時間帯の終了時間を越えて継続されることがあります。

バックアップ作成中は SQL 文発行の性能が低下するため、自動バックアップの時間帯は、業務の少ない時間帯に設定することを推奨します。

・ 自動メンテナンスの実施有無(autoMaintenance)／自動メンテナンス時間帯(preferredMaintenanceWindow)

自動メンテナンスでは、DB インスタンスのパッチ適用、および、適用に再起動が必要な DB インスタンスの変更がある場合、DB インスタンスの再起動が実施されます。

適用に再起動が必要な DB インスタンスの変更については、4.4 章を参照してください。

自動メンテナンスの実施有無に true または false を設定してください。

本パラメータを設定しない場合は、デフォルトで true が設定されます。false にすると、自動メンテナンスを実施しません。

自動メンテナンス時間帯には、週次の自動メンテナンスが行われる時間帯を設定してください。

上記で設定した曜日、設定した時間帯の開始時間になると、メンテナンスが自動的に開始されます。

自動メンテナンス中は SQL 文発行が出来なくなるため、自動メンテナンス時間帯は、業務の最も少ない時間帯に設定し、必要に応じて変更してください。変更手順は 4.4 章で説明します。

・ DB エンジン(engine)とバージョン(engineVersion)

DB エンジンとバージョンは、デフォルトでは最新の DB エンジンとバージョンが選択されます。

2023 年 4 月時点で作成できる DB エンジンのバージョンは、下記です。

engine	engineVersion
enterprisepostgres（デフォルト）	9.6（デフォルト）

なお、DB エンジンバージョン（engineVersion）は、提供開始から 5 年で公開停止となるため、最新の DB エンジンとバージョンを使用することを推奨します。

※2023 年 3 月をもって symfoware DB エンジンは提供を終了しました。

詳細は付録 B を参照してください。

その他オプションの設定によって、DB インスタンスの構成は、シングル構成かマルチ構成を選択できます。
以下では、シングル構成の DB インスタンスの作成例を記載します。

コマンド例

```
AZ=[DB インスタンスの作成先として設定する AZ 名（公開されているアベイラビリティゾーンから任意の一つを選択）]
INSTANCEID=[DB インスタンスの ID（任意の値を設定してください）]
INSTANCENAME=[DB インスタンスの名前（任意の値を設定してください）]
SUBNETGROUPID=[(1)で作成した DB サブネットグループの ID]
FLAVOR=[フレーバーID（公開されているフレーバーID から任意の一つを選択）]
MASTERUSERNAME=[データベースの管理者名（任意の値を設定してください）]
MASTERUSERPASSWORD=[データベースの管理者パスワード（任意の値を設定してください）]
PORT=[DB インスタンス作成時に設定するポート番号（2.1 章(2)で設定した PORT と同じ値を設定してください）]
SECGRP_ID=[2.1 章(1)で作成したセキュリティグループの ID]
PARAMG_ID=[(2)で作成した DB パラメータグループの ID]
SIZE=[データディスクのサイズ［GB］（10～10240 の任意の値を設定してください）]
BACKUP_RETENTION_PERIOD=[バックアップ保持期間［日］（0～10 の任意の値を設定してください）]
PREFERRED_BACKUP_WINDOW=[バックアップ時間帯[UTC 時刻]（30 分間以上の任意の時間帯を設定してください）]
AUTO_MAINTENANCE=[自動メンテナンスの実施有無（true/false のいずれかを設定してください）]
PREFERRED_MAINTENANCE_WINDOW=[メンテナンス時間帯[UTC 時刻]（30 分間以上の任意の曜日・時間帯を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"multiAZ¥": ¥"false¥", ¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"name¥": ¥"${INSTANCENAME}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥", ¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${INSTANCEID}¥", ¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}}"
```

実行結果例

```
$ AZ=jp-east-1a
$ INSTANCEID=test-instance-id
$ INSTANCENAME=test-instance-name
$ SUBNETGROUPID=test-subnetgroup-id
$ FLAVOR=1101
$ MASTERUSERNAME="masterusername"
$ MASTERUSERPASSWORD="masteruserpassword"
$ PORT=26500
$ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
$ PARAMG_ID=test-paramg-id
$ SIZE=10
$ BACKUP_RETENTION_PERIOD=5
$ PREFERRED_BACKUP_WINDOW="16:00-16:30"
$ AUTO_MAINTENANCE=true
$ PREFERRED_MAINTENANCE_WINDOW="Sat:18:00-Sat:18:30"
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"multiAZ¥": ¥"false¥", ¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"name¥": ¥"${INSTANCENAME}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥", ¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${INSTANCEID}¥", ¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

上記コマンド実行後、DB インスタンス作成が完了すると、status が Active になります。

DB インスタンスの状態については、付録 I または「API リファレンス（Application Platform Service 編）」を参照してください。

［参照］

※マルチ構成の DB インスタンス作成例

マルチ構成の DB インスタンスは、DB インスタンスの作成 API の multi、および、multiAZ パラメータを true にすることで作成できます。各パラメータと DB インスタンスの構成は以下の関係になっています。

multi パラメータ	multiAZ パラメータ	DB インスタンスの構成
false	false	シングル構成の DB インスタンスが作成されます。
true	false	同一 AZ に 2 台の DB インスタンスが作成され、冗長化されます。
false	true	設定できません。
true	true	設定した DB サブネットグループに含まれる 2 つの AZ それぞれに 1 台ずつ、計 2 台の DB インスタンスが作成され、冗長化されます。

以下は、multi=true、multiAZ=true でマルチ構成の DB インスタンスを作成するコマンド例です。

※フレーバーID については付録 C を参照し、適切なものを設定してください。

コマンド例

```
AZ=[DB インスタンスの作成先として設定する AZ 名（公開されているアベイラビリティゾーンから任意の一つを選択）]
INSTANCEID=[DB インスタンスの ID（任意の値を設定してください）]
INSTANCENAME=[DB インスタンスの名前（任意の値を設定してください）]
SUBNETGROUPID=[(1)で作成した DB サブネットグループの ID]
FLAVOR=[フレーバーID（公開されているフレーバーID から任意の一つを選択）]
MASTERUSERNAME=[データベースの管理者名（任意の値を設定してください）]
MASTERUSERPASSWORD=[データベースの管理者パスワード（任意の値を設定してください）]
PORT=[DB インスタンス作成時に設定するポート番号（2.1 章(2)で設定した PORT と同じ値を設定してください）]
SECGRP_ID=[2.1 章(1)で作成したセキュリティグループの ID]
PARAMG_ID=[(2)で作成した DB パラメータグループの ID]
SIZE=[データディスクのサイズ［GB］（10～10240 の任意の値を設定してください）]
BACKUP_RETENTION_PERIOD=[バックアップ保持期間［日］（0～10 の任意の値を設定してください）]
PREFERRED_BACKUP_WINDOW=[バックアップ時間帯[UTC 時刻]（30 分間以上の任意の時間帯を設定してください）]
AUTO_MAINTENANCE=[自動メンテナンスの実施有無（true/false のいずれかを設定してください）]
PREFERRED_MAINTENANCE_WINDOW=[メンテナンス時間帯[UTC 時刻]（30 分間以上の任意の曜日・時間帯を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"multiAZ¥": ¥"true¥", ¥"multi¥": ¥"true¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"name¥": ¥"${INSTANCENAME}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"flavorRef¥": ¥"${FLAVOR}¥",
```

```
¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥",
¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥",
¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}],
¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${INSTANCEID}¥", ¥"preferredBackupWindow¥":
¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}}"
```

2.3 DB インスタンスへの接続

DB インスタンスへの接続方法を説明します。

(1) DB インスタンスの情報参照

接続したい DB インスタンスの状態を確認します。ここでは、以下の 2 点を確認します。

- DB インスタンスの状態が Active になっていることを確認
- データベースへの接続に必要な FQDN を確認

コマンド例

```
INSTANCEID=[2.2 章(3)で作成した DB インスタンスの ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ INSTANCEID= test-instance-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
X-Fcx-Endpoint-Request: EXECUTED_REQ016533324_200
Date: Tue, 12 Jul 2016 02:31:33 GMT
Server: Apache/2.2.15 (CentOS)
Pragma: no-cache
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Cache-Control: no-cache, no-store
Content-Type: application/json;charset=UTF-8
Content-Length: 1601

{"instance":{"volume":{"size":10,"type":"M1"},"id":"test-instance-id","name":"test-instance-
name","description":null,"multiAZ":false,"multi":false,"port":26500,"preferredBackupWindow":"17:50-
18:20","preferredMaintenanceWindow":"mon:01:46-
mon:02:16","preferredRecoveryTime":{"applyImmediately":true,"time":null},"securityGroupIds":[{"securityGrou
pId":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"}],"parameterGroupId":"DefaultGroup-
12.1","backupRetentionPeriod":0,"autoMinorVersionUpgrade":true,"engineVersion":"9.6","engineMinorVersion":"
0","autoMaintenance":true,"availabilityZone":"XXXXXXXX","subnetGroupId":"test-subnetgroup-
```

```
id","publiclyAccessible":false,"engine":"enterprisepostgres","masterUserName":"masterusername","characterSet":"UTF8","collate":"C","created":"2016-07-12T02:09:01Z","flavor":{"id":"1101","links":[{"href":"http://XXXXXXXX/v1.0/XXXXXXXX/flavors/1101","rel":"SELF"},{"href":"http://XXXXXXXX/XXXXXXXX/flavors/1101","rel":"BOOKMARK"}]},"links":[{"href":"http://XXXXXXXX/v1.0/XXXXXXXX/instances/test-instance-id","rel":"SELF"},{"href":"http://XXXXXXXX/XXXXXXXX/instances/test-instance-id","rel":"BOOKMARK"}],"status":"Active","updated":null,"privateAddress":"XXXXXXXX.XXX.XXX","privateIp":"XX.XX.XX.XX","publicAddress":null,"publicIp":null,"subPrivateIp":null,"subPublicIp":null,"pendingModifiedValues":null,"readReplicaDBInstanceIdentifiers":[],"readReplicaSrcDBInstanceIdentifier":null,"secondaryAvailabilityZone":null,"replicaStatus":null}}
```


データベースへの接続には、DB インスタンス毎に提供される FQDN を使用します。
FQDN は、上記 API のレスポンス中の privateAddress に記載されています。この FQDN を控えておき、(2)で使用します。

※ 本章で確認した FQDN、および、2.2 章(3)で DB インスタンス作成時に設定したポート番号、データベースの管理者名／管理者パスワードは、(2)でデータベースへ接続するために使用します。DB インスタンスを作成したユーザーからデータベースの管理者へ情報を引き継いでください。

(2) DB インスタンスへの接続

本章はデータベースの管理者が実施します。

本章では、DB インスタンス上のデータベースへの接続方法の例を説明します。

データベースへの接続は、DB エンジンのクライアント（Enterprise Postgres Client または Symfoware Server Client）がインストールされた DB クライアントで実行します。

DB エンジンのクライアントの入手方法については、付録 B を参照してください。

DB エンジンのクライアントの機能については、DB エンジンのマニュアル（Enterprise Postgres のマニュアルまたは Symfoware Server のマニュアル）を参照してください。

※pgAdmin を利用する場合は、DB エンジンのマニュアルを参照してください。
（pgAdmin に設定するサーバ情報には、(1)で確認したパラメータを使用してください。
「ホスト」には(1)で確認した FQDN を設定してください。）

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[(1)で確認した FQDN 形式のアドレス]
PORT=[(1)で確認したポート]
MASTERUSERNAME=[(1)で確認したデータベースの管理者名]
SQL_STR=[発行したい SQL 文]
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
```

実行結果例

```
$ CLIENT_DIR="/XXXXXXXX/XXXXXXXX"
$ export PATH=${CLIENT_DIR}/bin:${PATH}
$ export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$ FQDN="XXXXXXXX.XXX.XXX"
$ PORT="26500"
$ MASTERUSERNAME="masterusername"
$ SQL_STR="CREATE DATABASE testdb;"
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
Password for user masterusername:
CREATE DATABASE
```

［参照］

データベースに格納するデータは透過的データ暗号化機能を使用することで、暗号化することができます。
透過的データ暗号化機能を使用するために DB インスタンス作成時に必要な初期設定手順はありません。
透過的データ暗号化機能については、付録 D を参照してください。

2.4 イベント通知の登録

DB インスタンスへのイベント通知登録方法を説明します。
イベント通知を登録しておくと、DB インスタンスに発生したイベントを確認することができます。
イベントを確認することで、DB インスタンスに発生した異常を検知することができます。

(1) イベント通知登録の作成

2.2 章で作成した DB インスタンスにイベント通知を登録します。

コマンド例

```
EVENTSUB_ID=[イベント通知登録の ID（任意の値を設定してください）]
SOURCEID=[2.2 章(3)で作成した DB インスタンスの ID]
EVENTSUB_NAME=[イベント通知登録の名前（任意の値を設定してください）]
SOURCE_TYPE=[イベント通知を行うリソースの種類（db-instance または db-snapshot を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/eventnotifications -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"eventnotification¥": {¥"id¥": ¥"${EVENTSUB_ID}¥", ¥"sourceIds¥": [{¥"sourceId¥": ¥"${SOURCEID}¥"}], ¥"enabled¥": ¥"false¥", ¥"name¥": ¥"${EVENTSUB_NAME}¥", ¥"sourceType¥": ¥"${SOURCE_TYPE}¥"}}"
```

実行結果例

```
$ EVENTSUB_ID=test-eventsub-id
$ SOURCEID=test-instance-id
$ EVENTSUB_NAME=test-eventsub-name
$ SOURCE_TYPE=db-instance
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/eventnotifications -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"eventnotification¥": {¥"id¥": ¥"${EVENTSUB_ID}¥", ¥"sourceIds¥": [{¥"sourceId¥": ¥"${SOURCEID}¥"}], ¥"enabled¥": ¥"false¥", ¥"name¥": ¥"${EVENTSUB_NAME}¥", ¥"sourceType¥": ¥"${SOURCE_TYPE}¥"}}"
HTTP/1.1 200 OK
（省略）
```

(2) イベント通知登録の情報参照

イベント通知が登録されたことを確認します。

API のレスポンス中、sourceId にイベント通知を登録した DB インスタンスの ID が表示されることを確認してください。

コマンド例

```
EVENTSUB_ID=[(1)で作成したイベント通知登録の ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/eventnotifications/${EVENTSUB_ID} -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ EVENTSUB_ID=test-eventsub-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/eventnotifications/${EVENTSUB_ID} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"eventSubscription":{"enabled":false,"sourceType":"db-instance","id":"test-eventsub-id","name":"test-eventsub-name","eventCategories":null,"sourceIds":[{"sourceId":"test-instance-id"}],"description":null,"status":"Active","created":"YYYY-MM-DDThh:mm:ssZ"}}
```

2.5 性能検証

性能検証について説明します。

アプリケーション単体での性能検証だけでなく、業務全体の性能評価を実施することが重要となります。そのため以下での性能評価を実施する必要があります。

1） 本番運用を想定した負荷による性能検証

将来的な業務拡張を見据えた負荷を設定します。

2）性能目標の達成確認

業務全体のスループット、アプリケーションごとのレスポンスを確認します。

3）性能目標達成に向けたチューニング

・業務全体のスループットを改善する場合は、データベース活動状況のモニタリング情報などを元に、
必要に応じてチェックポイント間隔の見直しやデータベースバッファの変更などのチューニングを実施します。

・アプリケーションレスポンスを改善する場合は、SQL 文の実行計画をベースに、
必要に応じてインデックスの見直しや pg_hint_plan による実行計画の手動変更（※）などのチューニングを実施します。
※大量の更新クエリなどで統計情報の最新化が間にあわない場合や、刻々と統計情報が変化する場合に、
プランナ（オプティマイザ）が作成する実行計画が不安定になることがあります。その結果、レスポンスが平準化されない事象が発生する可能性があります。該当 SQL 文の実行計画を確認し、pg_hint_plan を利用してチューニングしてください。
なお、pg_hint_plan は使用するセッションにてロードしてください。

第 3 章 DB インスタンスの運用（日々の運用）

本章の内容は、全体管理者ロール、設計・構築者ロールまたは運用者ロールで実施できます。
ただし、3.3 章のデータベースのログ確認については、監視者ロールでも実施できます。

3.1 DB インスタンスの停止・起動

DB インスタンスの停止・起動手順について説明します。

運用方針に応じて実施してください。（例：DB インスタンスを使用しない時間帯にはデータベース仮想サーバの料金が発生しないようにしたい場合。）

(1) DB インスタンスの停止

DB インスタンスを停止します。

DB インスタンス停止中はデータベースへの SQL 接続はできません。

DB インスタンスを停止すると、SQL 接続が切断されるため、DB インスタンス停止前には、アプリケーションを停止してください。

また、停止している DB インスタンスについては、データベース仮想サーバの料金は発生しません。

コマンド例
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"stop": ""}}'
実行結果例
$ INSTANCEID=test-instance-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H

```
"Content-Type: application/json" -d '{"action": {"stop": ""}}'
HTTP/1.1 202 Accepted
（省略）
```

上記コマンド実行後、停止処理が完了すると status が Shutdown になります。
DB インスタンスの状態は DB インスタンスの情報参照 API で確認できます。（「API リファレンス（Application Platform Service 編）」、または、2.3 章(1)を参照してください。）

(2) DB インスタンスの起動

停止中の DB インスタンスを起動します。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"start": ""}}'
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"start": ""}}'
HTTP/1.1 202 Accepted
（省略）
```

DB インスタンスの状態は DB インスタンスの情報参照 API で確認できます。（「API リファレンス（Application Platform Service 編）」、または、2.3 章(1)を参照してください。）

3.2 手動でのバックアップ作成方法

DB インスタンスのバックアップを手動で作成する方法を説明します。

(1) DB スナップショットの作成

DB インスタンスのバックアップを手動で作成します。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
```

```
DESCRIPTION=[DB スナップショットの説明（任意の内容を設定してください）]
SNAPSHOT_ID=[DB スナップショットの ID（任意の値を設定してください）]
SNAPSHOT_NAME=[DB スナップショットの名前（任意の値を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"snapshot¥": {¥"instanceId¥": ¥"${INSTANCEID}¥", ¥"description¥": ¥"${DESCRIPTION}¥", ¥"name¥": ¥"${SNAPSHOT_NAME}¥", ¥"id¥": ¥"${SNAPSHOT_ID}¥"}}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ DESCRIPTION="this is sample"
$ SNAPSHOT_ID=test-snapshot-id
$ SNAPSHOT_NAME=test-snapshot-name
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"snapshot¥": {¥"instanceId¥": ¥"${INSTANCEID}¥", ¥"description¥": ¥"${DESCRIPTION}¥", ¥"name¥": ¥"${SNAPSHOT_NAME}¥", ¥"id¥": ¥"${SNAPSHOT_ID}¥"}}"
HTTP/1.1 202 Accepted
（省略）

{"snapshot":{"instanceId":"test-instance-id","id":"test-snapshot-id","name":"test-snapshot-name","snapshotType":"manual","status":"In_Progress","created":null,"description":"this is sample"}}
```

(2) DB スナップショットの一覧参照

作成した DB スナップショットについて、作成が完了したことを確認します。

以下の DB スナップショットの一覧参照 API を実行し、status が Available になっていることを確認します。

DB スナップショット作成が完了するまでの時間は、データ量に依存します。

コマンド例

```
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots?snapshotType=manual -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots?snapshotType=manual -H "X-Auth-Token: ${TOKEN}"
```

```
HTTP/1.1 200 OK
（省略）
{"snapshots":[{"instanceId":"test-instance-
id","id":"XXXXXXXX","name":"XXXXXXXX","snapshotType":"manual","status":"Available"}]}
```

［参照］

不要になった DB スナップショットは、スナップショット・自動バックアップの料金が発生しないよう、削除することを推奨します。
DB スナップショットの削除 API のコマンド例は以下のようになります。

コマンド例
SNAPSHOT_ID=[不要になった DB スナップショットの ID] ENDPOINT=[データベースサービスのエンドポイント] curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots/${SNAPSHOT_ID} -H "X-Auth-Token: ${TOKEN}"
実行結果例
$ SNAPSHOT_ID=test-snapshot-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/snapshots/${SNAPSHOT_ID} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 202 Accepted

DB スナップショットが削除されたことは、DB スナップショットの一覧参照 API で確認できます。
DB スナップショットの一覧参照 API で、削除した DB スナップショットが一覧に表示されなくなることを確認してください。

3.3 データベースのログ確認

データベースのログはデータベースのエラー発生時や作成するアプリケーションの動作確認に使用できます。

(1) DB ログファイルの一覧参照

データベースのログファイル一覧を取得します。

コマンド例
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）] LIMIT=[表示するログファイル数（20～100 の任意の値を設定してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"

実行結果例

```
$ INSTANCEID=test-instance-id
$ LIMIT=100
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"describeDBLogFiles":[{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX},{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX},{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX}]}
```

［参照］

特定の日付の DB ログファイルを参照したい場合、filelastwritten パラメータを設定することで、設定した日時以降に書き込みが行われた DB ログファイルの一覧を参照できます。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
LIMIT=[表示するログファイル数（20～100 の任意の値を設定してください）]
FILELASTWRITTEN=[ログファイルの最終書き込み日時（POSIX time stamp）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}?limit=${LIMIT}&filelastwritten=${FILELASTWRITTEN} -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ LIMIT=100
$ FILELASTWRITTEN=1485907200
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}?limit=${LIMIT}&filelastwritten=${FILELASTWRITTEN} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"describeDBLogFiles":[{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX},{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX},{"lastWritten":XXXXXXXX,"logFileName":"postgresql.log-YYYY-MM-DD_XXXXXX","size":XXXX}]}
```

(2) DB ログファイルの参照

(1)で取得した DB ログファイル一覧から、確認したい DB ログファイルを参照します。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
LOGFILENAME=[(1)で確認した DB ログファイルの名前]
LIMIT=[表示する行数（20～100 の任意の値を設定してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}/${LOGFILENAME}?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ LOGFILENAME=postgresql.log-YYYY-MM-DD_XXXXXX
$ LIMIT=100
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}/${LOGFILENAME}?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"downloadDBLogFilePortion":{"additionalDataPending":false,"logFileData":"LOG:  XXXXXXXX¥nLOG:  XXXXXXXX¥nLOG:  XXXXXXXX¥nLOG:  XXXXXXXX¥n","marker":"-1"}}
```

［参照］

ページネーションのため、marker を使用することで、どの位置からログファイルを取得するかを設定できます。

marker には前回の DB ログファイルの参照 API で返却された値を設定します。

marker を設定した場合、marker の次の行から DB ログファイルを表示します。

（marker を設定しない場合、先頭から DB ログファイルを表示します。）

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
LOGFILENAME=[(1)で確認した DB ログファイルの名前]
LIMIT=[表示する行数（20～100 の任意の値を設定してください）]
MARKER=[どの位置からリストを取得するか設定する文字列（設定した marker の次の行からログを表示します）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}/${LOGFILENAME}?marker=${MARKER}&limit=${LIMIT} -H "X-
```

```
Auth-Token: ${TOKEN}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ LOGFILENAME=postgresql.log-YYYY-MM-DD_XXXXXX
$ LIMIT=20
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}/${LOGFILENAME}?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"downloadDBLogFilePortion":{"additionalDataPending":false,"logFileData":"LOG: XXXXXXXX¥nLOG: XXXXXXXX¥nLOG: XXXXXXXX¥nLOG: XXXXXXXX¥n","marker":"20"}} ★この marker を MARKER に設定し、DB ログファイルの参照を行います

$ MARKER=20
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/logfiles/${INSTANCEID}/${LOGFILENAME}?marker=${MARKER}&limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"downloadDBLogFilePortion":{"additionalDataPending":false,"logFileData":"LOG: XXXXXXXX¥nLOG: XXXXXXXX¥nLOG: XXXXXXXX¥nLOG: XXXXXXXX¥n","marker":"null"}} ★前回結果の次の行から DB ログファイルを表示します
```

3.4 性能監視

監視サービスを利用して、DB インスタンスの性能情報を取得することができます。

データベース仮想サーバのディスク容量については、お客様にて性能情報を定期的に確認するようにしてください。20%程度のディスク空き容量を保つことを推奨します。ディスク容量が枯渇した場合は、リストアまたはデータベースの再作成が必要になるためご注意ください。

ディスクの空き容量が少なくなってきた場合、ディスクサイズの拡張をご検討ください（4.2 章 データディスクサイズ変更参照）。また、ディスク使用率が 80%を超えた場合、イベント通知機能のイベントカテゴリ「low storage」によりメール通知を実施します。通知を受け取った場合、ディスクの空き容量を確認し、ディスクサイズの拡張を検討してください。

性能情報の取得方法については、「API リファレンス（Management Administration 編）」の「第 4 章: 監視サービス」を参照してください。

取得できるデータベースサービスの監視項目は「機能説明書」の「A.7 監視項目一覧」を参照してください。

3.5 イベント通知の確認

FUJITSU Hybrid IT Service FJcloud-O の契約者と全体管理者にイベント通知がメールで通知されます。

必要に応じて、DB インスタンスの復旧対処をしてください。DB インスタンスの復旧については第 5 章を参照してください。

3.6 透過的データ暗号化で使用する暗号化キーの変更

透過的データ暗号化機能（付録 D 参照）を使用すると、データベースに格納するデータを暗号化することができます。

透過的データ暗号化機能を使用する場合、マスタ暗号化キーとキーストアのパスフレーズは定期的に変更することを推奨します。

第 4 章　DB インスタンスの変更（必要に応じて実施）

本章の内容は、全体管理者ロールまたは設計・構築者ロールで実施できます。

4.1　フレーバーID の変更

運用の必要に応じて、フレーバーを変更してください。

※フレーバーID については付録 C を参照し、適切なものを設定してください。

以下の例では、DB インスタンスの変更を即時適用します。即時適用を実施する場合、API 実行後に DB インスタンスが再起動します。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
FLAVOR=[フレーバーID（公開されているフレーバーID から任意の一つを選択）]
APPLY_IMMEDIATELY=[DB インスタンスの変更の即時適用の有無（true または false を選択）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ FLAVOR=1102
$ APPLY_IMMEDIATELY=true
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

4.2　データディスクのサイズ変更

運用の必要に応じて、データディスクのサイズを変更してください。

データディスクのサイズは、サイズを大きくする変更のみ可能です。サイズを小さくすることはできません。

自動バックアップを取得していない DB インスタンスのデータディスクのサイズを変更する場合、事前に DB スナップショットを作成してください。DB スナップショットの作成方法については、3.2 章を参照してください。

以下の例では、DB インスタンスの変更を即時適用します。即時適用を実施する場合、API 実行後に DB インスタンスが再起

動します。

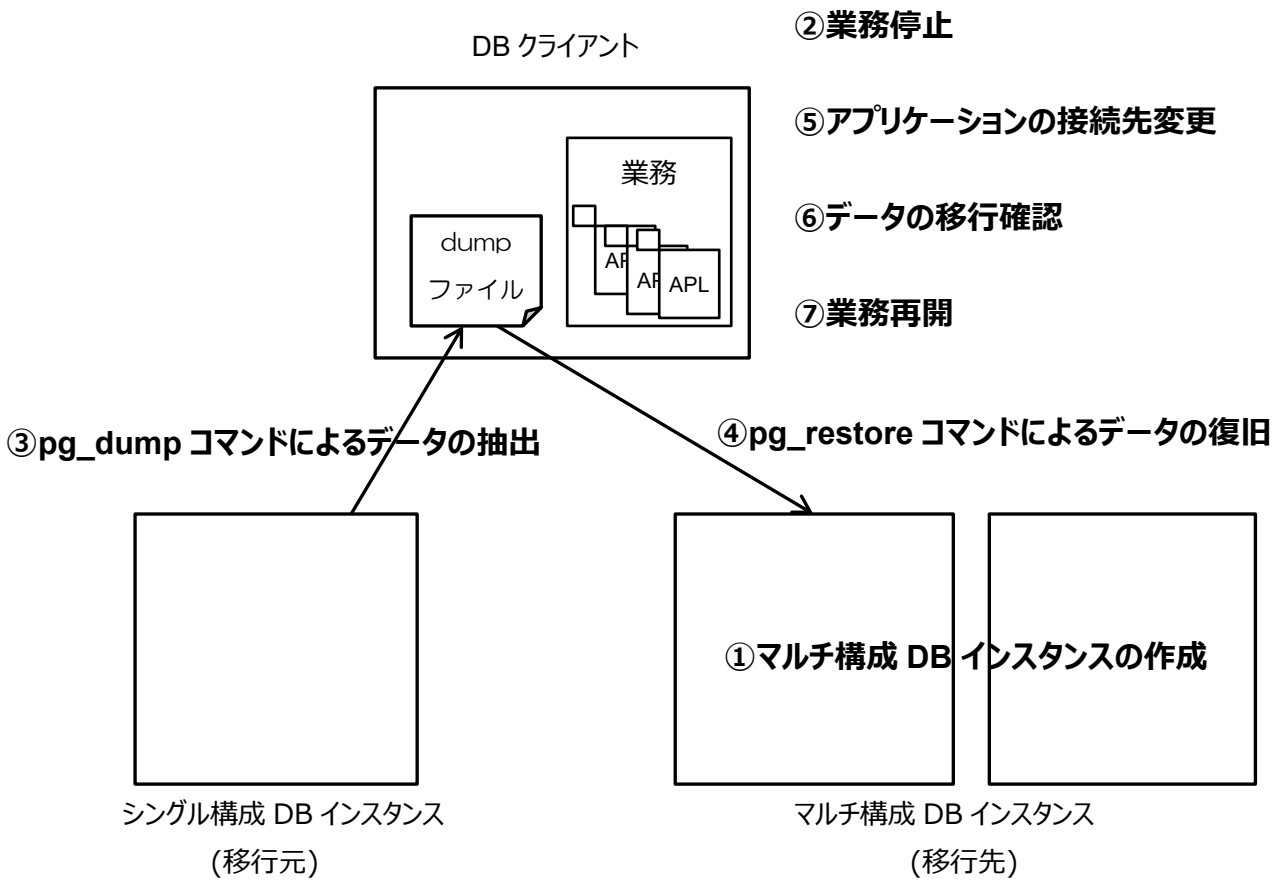
コマンド例
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）] SIZE=[データディスクのサイズ［GB］（10～10240 の任意の値を設定してください）] APPLY_IMMEDIATELY=[DB インスタンスの変更の即時適用の有無（true または false を選択）] ENDPOINT=[データベースサービスのエンドポイント] curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"volume¥": {¥"size¥": ${SIZE}}, ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}"

実行結果例
$ INSTANCEID=test-instance-id $ SIZE=20 $ APPLY_IMMEDIATELY=true $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"volume¥": {¥"size¥": ${SIZE}}, ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}" HTTP/1.1 202 Accepted （省略）

4.3 シングル構成からマルチ構成への変更

シングル構成の DB インスタンスをマルチ構成の DB インスタンスに移行する手順を以下に示します。

なお、本手順により IP アドレス・FQDN が変更されます。



［注意］

DB インスタンスの変更 API では、データベース仮想サーバの冗長構成の変更(シングル→冗長化、冗長化→シングル)は実行できません。シングル構成を冗長構成に変更したい場合、本章の手順により、元のデータをバックアップ後、冗長構成のデータベース仮想サーバを新規に作成し、バックアップからデータをリストアしてください。

(1) 新規にマルチ構成 DB インスタンスを作成

DB インスタンスの作成 API に以下のオプションを設定して作成してください。

- multi
- multiAZ (AZ 間で冗長化する場合)

コマンド例は、2.2 章(3) 「※マルチ構成の DB インスタンス作成例」 を参照してください。

(2) 業務停止

運用中のシングル構成 DB インスタンスへの業務を停止してください。

(3) データの抽出

DB エンジンのクライアント（Enterprise Postgres Client または Symfoware Server Client）がインストールされた DB クライアントで実行します。

移行元の DB インスタンスからデータベースのデータ、スキーマを以下のコマンドを実行し取得してください。

移行元で使用しているデータベース数分取得を実施してください。

例：データベースからデータ、スキーマの抽出（カスタムアーカイブ形式）

コマンド例
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス] export PATH=${CLIENT_DIR}/bin:${PATH} export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH} FQDN=[移行元の DB インスタンスの接続先（FQDN の確認方法は 2.3 章(1)を参照してください）] PORT=[移行元の DB インスタンスのポート番号（ポート番号の確認方法は 2.3 章(1)を参照してください）] MASTERUSERNAME=[移行元の DB インスタンスのデータベースの管理者名] DB=[バックアップする対象の DB 名] pg_dump -h ${FQDN} -p ${PORT} -U ${MASTERUSERNAME} --format=custom ${DB} > K5DB_dump.custom

(4) マルチ構成 DB インスタンスへのデータの挿入

DB エンジンのクライアントがインストールされた DB クライアントで実行します。

(1)で作成したマルチ構成 DB インスタンスに、(3)で抽出したスキーマ、データの復旧をします。

例：データ、スキーマの挿入

コマンド例
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス] export PATH=${CLIENT_DIR}/bin:${PATH} export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH} FQDN=[移行先の DB インスタンスの接続先（FQDN の確認方法は 2.3 章(1)を参照してください）] PORT=[移行先の DB インスタンスのポート番号（ポート番号の確認方法は 2.3 章(1)を参照してください）] MASTERUSERNAME=[移行先の DB インスタンスのデータベースの管理者名] pg_restore -h ${FQDN} -p ${PORT} -U ${MASTERUSERNAME} -C -d postgres < K5DB_dump.custom

(5) アプリケーションの修正

アプリケーションのデータベースへの接続先を、(1)で新規に配備したマルチ構成 DB インスタンスに変更してください。

(6) 業務再開

業務を再開してください。

［参照］

- pg_dump コマンド、および pg_restore コマンドの詳細については「PostgreSQL9.6.2 文書」または「PostgreSQL9.2.4 文書」の以下を参照してください。
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「pg_dump」
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「pg_restore」
- (3)の手順で全てのデータベースのデータ、スキーマを一括して取得する場合は pg_dumpall コマンドを使用してください。pg_dumpall コマンドの詳細については「PostgreSQL9.6.2 文書」または「PostgreSQL9.2.4 文書」の以下を参照してください。
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「pg_dumpall」
- (3)の手順で取得のダンプファイルの形式を SQL スクリプトファイルで取得した場合は、(4)の手順では psql コマンドを使用して復旧してください。psql コマンドの使用方法は「PostgreSQL9.6.2 文書」または「PostgreSQL9.2.4 文書」の以下を参照してください。
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「psql」

4.4 DB インスタンスの変更

DB インスタンスの変更 API では、本章で説明したもの以外にも、必要に応じて、DB インスタンスの構成を変更できます。
変更できるパラメータは「API リファレンス（Application Platform Service 編）」を参照してください。

DB インスタンスの変更では、パラメータによって、変更の適用に DB インスタンスの再起動が必要なものがあります。
適用に再起動が必要な DB インスタンスの変更は、以下のいずれかです。

- 以下のいずれかのパラメータを変更した場合
  - flavorRef
  - volume
  - multi
  - multiAZ
  - port
  - engineVersion

DB インスタンスの変更 API 実行時に、”applyImmediately”パラメータに true を設定することで、API 実行後に DB インスタンスが再起動し、変更を即時適用できます。
変更を即時適用しない場合は、”applyImmediately”パラメータに false を設定してください。この場合、DB インスタンスの変更は、自動メンテナンスが実行されたとき、または、手動で DB インスタンスを再起動したときに適用されます。

DB インスタンスの変更 API については「API リファレンス（Application Platform Service 編）」、または、4.1 章、4.2 章を参照してください。

4.5 データベースのパラメータ変更

データベースのパラメータ変更について説明します。

パラメータ変更は、以下の 2 通りの方法があります。

(A) 新規に DB パラメータグループを作成し、DB インスタンスに適用する

(B) DB インスタンスに適用済みの、既存の DB パラメータグループを変更する

(A)では、個別に DB インスタンスのパラメータを変更できます。

(B)では、対象の DB パラメータグループを適用済みの全ての DB インスタンスのパラメータを変更できます。

(A)の場合は、以下の(1)～(5)を全て実施してください。 (B)の場合は、以下の(2)、(3)、(5)を実施してください。

(1) DB パラメータグループの作成

DB パラメータグループの作成 API は「API リファレンス（Application Platform Service 編）」、または、2.2 章(2)を参照してください。

(2) パラメータ値の変更

DB パラメータグループのパラメータ値を変更します。運用上変更が必要なパラメータを変更してください。

変更できるパラメータは付録 A を参照してください。

以下は、”shared_buffers”というパラメータを変更する例を記載します。

コマンド例

```
PARAMG_ID=[事前に作成した DB パラメータグループの ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"parametergroup¥": {¥"parameters¥": [{¥"applyMethod¥": ¥"pending-reboot¥", ¥"name¥": ¥"shared_buffers¥", ¥"value¥": ¥"10000¥"}]}}"
```

実行結果例

```
$ PARAMG_ID=test-paramg-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"parametergroup¥": {¥"parameters¥": [{¥"applyMethod¥": ¥"pending-reboot¥", ¥"name¥": ¥"shared_buffers¥", ¥"value¥": ¥"10000¥"}]}}"
HTTP/1.1 200 OK
（省略）
{"parameterGroup":{"id":"XXXXXXXX","description":null,"name":"XXXXXXXX",
"parameters":[{"name":"shared_buffers","value":"10000","allowedValues":"","applyMethod":"pending-reboot","applyType":"static","dataType":"integer","description":"(8kB) Sets the number of shared memory buffers used by the
```

server.","minimumEngineVersion":"12.1","source":"user","isModifiable":true}],"parameterGroupFamily":"enterp
risepostgres_v9.6","appliedInstances":null}}

［参照］

PostgreSQL の SET コマンドでもパラメータを変更できますが、SET コマンドで変更したパラメータは変更したセッションのみ反映されるため、DB パラメータグループのパラメータ値の変更には影響しません。

(3) DB パラメータグループの情報参照

DB パラメータグループを参照し、(2)の DB パラメータグループへの変更が適用されていることを確認します。

この時点では、"applyMethod"が"pending-reboot"のものは DB インスタンスには反映されていません。

（"applyMethod"については付録 A を参照してください。）

コマンド例

```
PARAMG_ID=[(2)で変更した DB パラメータグループの ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}"
```

実行結果例

```
$ PARAMG_ID=test-paramg-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}"
HTTP/1.1 200 OK
（省略）
{"parameterGroup":{"id":"XXXXXXXX","description":null,"name":"XXXXXXXX","parameters":[{（省略）
,{"name":"shared_buffers","value":"10000",（省略）
}],"parameterGroupFamily":"enterprisepostgres_v9.6","appliedInstances":null}}
```

［参照］

現在の DB インスタンスに設定されているパラメータ値は、DB インスタンスに接続して、"SHOW ALL;"という SQL を実行することで確認できます。

SQL の実行方法については、2.3 章の(2)を参照してください。

(4) DB パラメータグループの適用

既存の DB インスタンスに、上記で新たに作成した DB パラメータグループを適用します。

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
PARAMG_ID=[(1)で作成した DB パラメータグループの ID]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"parameterGroupId¥": ¥"${PARAMG_ID}¥"}}"
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ PARAMG_ID=test-paramg-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"parameterGroupId¥": ¥"${PARAMG_ID}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

［参照］

現在の DB インスタンスに設定されているパラメータ値は、DB インスタンスに接続して、”SHOW ALL;”という SQL を実行することで確認できます。
SQL の実行方法については、2.3 章の(2)を参照してください。

(5) DB インスタンスの再起動

"applyMethod"が"pending-reboot"のパラメータを変更した場合は、DB インスタンスの再起動を行います。
（"applyMethod"については付録 A を参照してください。）

コマンド例

```
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"reboot": ""}}'
```

実行結果例

```
$ INSTANCEID=test-instance-id
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"reboot": ""}}'
```

```
HTTP/1.1 202 Accepted
（省略）
```

［参照］

・・

現在の DB インスタンスに設定されているパラメータ値は、DB インスタンスに接続して、"SHOW ALL;"という SQL を実行することで確認できます。

SQL の実行方法については、2.3 章の(2)を参照してください。

・・

［参照］

・・

誤ったパラメータ値を設定した状態で DB インスタンスの再起動を実施した場合、下記のイベント通知が実施されます。

source-type	category	message
db-instance	configuration change	Modification of db parameters failed. See DB logfiles.

このとき、DB インスタンスは起動していますが、誤ったパラメータ、および、"applyMethod"が"pending-reboot"のパラメータは反映されていない状態です。

誤ったパラメータに関する DB ログは、start_error.log という DB ログファイルに出力されます。
start_error.log を参照して、誤っていたパラメータを特定してください。
DB ログファイルの参照方法については、3.3 章の(2)の手順で、LOGFILENAME に start_error.log を指定することで参照できます。

誤ったパラメータを特定した後、4.5 章の(2)を参照してパラメータ値の変更を実施してください。
"applyMethod"が"pending-reboot"のパラメータを変更した場合は、4.5 章の(5)を参照し、DB インスタンスの再起動を実施してください。

・・

第 5 章　異常時の対処（リカバリ）

本章の内容は、全体管理者ロール、設計・構築者ロールまたは運用者ロールで実施します。

5.1 DB インスタンスの復旧

DB インスタンスの復旧方法について説明します。

DB インスタンスを復旧すると、データベースがリストアされます。

DB インスタンスのバックアップ・リストア時には、透過的データ暗号化機能（付録 D 参照）で使用するマスタ暗号化キー、および、自動オープンキーも自動的にバックアップ・リストアされます。

(1) DB インスタンスのリカバリ

DB インスタンスの復旧方法は以下の 2 通りがあります。

運用方針に応じて、パターン 1 、パターン 2 のいずれかを実施して DB インスタンスを復旧してください。

パターン 1：DB スナップショットからの DB インスタンス復旧

事前に作成した DB スナップショットから DB インスタンスを復旧します。

DB インスタンスを復旧すると、DB スナップショットの created の時刻に復旧されます。

DB スナップショットからの DB インスタンス復旧では、DB スナップショット取得元の DB インスタンスとは別の新規 DB インスタンスが作成されます。新規 DB インスタンスとなるため、FQDN/IP は新規に設定され、DB スナップショット取得元の DB インスタンスとは別に課金が発生します。また、作成される新規 DB インスタンスは、最新のマイナーバージョンで作成されます。

コマンド例

```
AZ=[DB インスタンスの作成先として設定する AZ 名（復旧元の DB インスタンスと同じ値を設定）]
SNAPSHOTID=[バックアップ元となるスナップショット ID（事前に作成しているものから任意の一つを選択してください。
スナップショット ID の確認方法は 3.2 章(2)を参照してください）]
RESTOREINSTID=[リストアで作成する DB インスタンスの ID（任意の値を設定してください）]
SUBNETGROUPID=[事前に作成した DB サブネットグループの ID（復旧元の DB インスタンスと同じ値を設定）]
FLAVOR=[フレーバーID（復旧元の DB インスタンスと同じ値を設定）]
MASTERUSERNAME=[データベースの管理者名（復旧元の DB インスタンスと同じ値を設定）]
MASTERUSERPASSWORD=[データベースの管理者パスワード（復旧元の DB インスタンスと同じ値を設定）]
PORT=[DB インスタンス作成時に設定するポート番号（復旧元の DB インスタンスと同じ値を設定）]
SECGRP_ID=[事前に作成したセキュリティグループの ID（復旧元の DB インスタンスと同じ値を設定）]
PARAMG_ID=[事前に作成した DB パラメータグループの ID（復旧元の DB インスタンスと同じ値を設定）]
SIZE=[データディスクのサイズ［GB］（復旧元の DB インスタンスと同じ値を設定）]
BACKUP_RETENTION_PERIOD=[バックアップ保持期間［日］（復旧元の DB インスタンスと同じ値を設定）]
PREFERRED_BACKUP_WINDOW=[バックアップ時間帯[UTC 時刻]（復旧元の DB インスタンスと同じ値を設定）]
AUTO_MAINTENANCE=[自動メンテナンスの実施有無（復旧元の DB インスタンスと同じ値を設定）]
PREFERRED_MAINTENANCE_WINDOW=[メンテナンス時間帯[UTC 時刻]（復旧元の DB インスタンスと同じ値を設定）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type:
application/json" -d "{¥"action¥": {¥"restoresnapshot¥": ¥"¥"}, ¥"instance¥": {¥"multiAZ¥": ¥"false¥",
¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥",
¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"flavorRef¥": ¥"${FLAVOR}¥",
¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥",
¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥",
¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}],
¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${RESTOREINSTID}¥",
¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"},
¥"snapshot¥": {¥"id¥": ¥"${SNAPSHOTID}¥"}}"
```

※MASTERUSERPASSWORD には SNAPSHOTID に指定するスナップショットの作成日時での masterUserPassword と同じ値を指定してください。

BACKUP_RETENTION_PERIOD に 1 以上を設定した場合は、DB インスタンスのリカバリ後に初回のバックアップが実行されます。バックアップが完了するまで status は Build のままになります。バックアップが完了すると status が Active になります。

実行結果例

```
$ AZ=jp-east-1a
$ SNAPSHOTID=test-snapshot-id
$ RESTOREINSTID=test-restore-instance-id
$ SUBNETGROUPID=test-subnetgroup-id
$ FLAVOR=1101
$ MASTERUSERNAME="masterusername"
$ MASTERUSERPASSWORD="masteruserpassword"
$ PORT=26500
$ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
$ PARAMG_ID=test-paramg-id
$ SIZE=10
$ BACKUP_RETENTION_PERIOD=5
$ PREFERRED_BACKUP_WINDOW="16:00-16:30"
$ AUTO_MAINTENANCE=true
$ PREFERRED_MAINTENANCE_WINDOW="Sat:18:00-Sat:18:30"
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"action¥": {¥"restoresnapshot¥": ¥"¥"}, ¥"instance¥": {¥"multiAZ¥": ¥"false¥", ¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥", ¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${RESTOREINSTID}¥", ¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}, ¥"snapshot¥": {¥"id¥": ¥"${SNAPSHOTID}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

パターン 2：ポイントイン･タイムリカバリーによる DB インスタンスの復旧

バックアップ保持期間中の任意の時点を指定して、その時点の状態に DB インスタンスを復旧することができます。
最新のリストア可能時間は、通常現在時刻～5 分前の間となっています。
ポイントイン･タイムリカバリーによる DB インスタンスの復旧では、リカバリ元の DB インスタンスとは別の新規 DB インスタンスが作成されます。新規 DB インスタンスとなるため、FQDN/IP は新規に設定され、リカバリ元の DB インスタンスとは別に課金が発生します。また、作成される新規 DB インスタンスは、最新のマイナーバージョンで作成されます。

コマンド例

```
AZ=[DB インスタンスの作成先として設定する AZ 名（復旧元の DB インスタンスと同じ値を設定）]
INSTANCEID=[リカバリ元となる DB インスタンスの ID（事前に作成したもの。(DB インスタンスの作成については 2.2 章(3)を参照してください）]
PITR_INSTID=[ポイントイン・タイムリカバリーで作成する DB インスタンスの ID（任意の値を設定してください）]
USE_LATEST_RESTORABLE_TIME=[最新のリストア可能時点へのリストアの有無（true または false を選択）]
SUBNETGROUPID=[事前に作成した DB サブネットグループの ID（復旧元の DB インスタンスと同じ値を設定）]
FLAVOR=[フレーバーID（復旧元の DB インスタンスと同じ値を設定）]
MASTERUSERNAME=[データベースの管理者名（復旧元の DB インスタンスと同じ値を設定）]
MASTERUSERPASSWORD=[データベースの管理者パスワード（復旧元の DB インスタンスと同じ値を設定）]
PORT=[DB インスタンス作成時に設定するポート番号（復旧元の DB インスタンスと同じ値を設定）]
SECGRP_ID=[事前に作成したセキュリティグループの ID（復旧元の DB インスタンスと同じ値を設定）]
PARAMG_ID=[事前に作成した DB パラメータグループの ID（復旧元の DB インスタンスと同じ値を設定）]
SIZE=[データディスクのサイズ［GB］（復旧元の DB インスタンスと同じ値を設定）]
BACKUP_RETENTION_PERIOD=[バックアップ保持期間［日］（復旧元の DB インスタンスと同じ値を設定）]
PREFERRED_BACKUP_WINDOW=[バックアップ時間帯[UTC 時刻]（復旧元の DB インスタンスと同じ値を設定）]
AUTO_MAINTENANCE=[自動メンテナンスの実施有無（復旧元の DB インスタンスと同じ値を設定）]
PREFERRED_MAINTENANCE_WINDOW=[メンテナンス時間帯[UTC 時刻]（復旧元の DB インスタンスと同じ値を設定）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"action¥": {¥"restoretopointintime¥": ¥"¥"}, ¥"restore¥": {¥"useLatestRestorableTime¥": ¥"${USE_LATEST_RESTORABLE_TIME}¥"}, ¥"instance¥": {¥"multiAZ¥": ¥"false¥", ¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥", ¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${PITR_INSTID}¥", ¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}}"
```

※ restoreTime を指定した場合、MASTERUSERPASSWORD には指定した時刻での DB インスタンスの masterUserPassword と同じ値を指定してください。useLatestRestorableTime に true を指定した場合、

MASTERUSERPASSWORD には現時点での DB インスタンスの masterUserPassword と同じ値を指定してください。

BACKUP_RETENTION_PERIOD に 1 以上を設定した場合は、DB インスタンスのリカバリ後に初回のバックアップが実行されます。バックアップが完了するまで status は Build のままになります。バックアップが完了すると status が Active になります。

実行結果例

```
$ AZ=jp-east-1a
$ INSTANCEID=test-instance-id
$ PITR_INSTID=test-pitr-instance-id
$ USE_LATEST_RESTORABLE_TIME=true
$ SUBNETGROUPID=test-subnetgroup-id
$ FLAVOR=1101
$ MASTERUSERNAME="masterusername"
$ MASTERUSERPASSWORD="masteruserpassword"
$ PORT=26500
$ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
$ PARAMG_ID=test-paramg-id
$ SIZE=10
$ BACKUP_RETENTION_PERIOD=5
$ PREFERRED_BACKUP_WINDOW="16:00-16:30"
$ AUTO_MAINTENANCE=true
$ PREFERRED_MAINTENANCE_WINDOW="Sat:18:00-Sat:18:30"
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"action¥": {¥"restoretopointintime¥": ¥"¥"}, ¥"restore¥": {¥"useLatestRestorableTime¥": ¥"${USE_LATEST_RESTORABLE_TIME}¥"}, ¥"instance¥": {¥"multiAZ¥": ¥"false¥", ¥"multi¥": ¥"false¥", ¥"availabilityZone¥": ¥"${AZ}¥", ¥"publiclyAccessible¥": ¥"false¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"masterUserName¥": ¥"${MASTERUSERNAME}¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"masterUserPassword¥": ¥"${MASTERUSERPASSWORD}¥", ¥"port¥": ${PORT}, ¥"volume¥": {¥"type¥": ¥"M1¥", ¥"size¥": ${SIZE}}, ¥"preferredMaintenanceWindow¥": ¥"${PREFERRED_MAINTENANCE_WINDOW}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"backupRetentionPeriod¥": ${BACKUP_RETENTION_PERIOD}, ¥"id¥": ¥"${PITR_INSTID}¥", ¥"preferredBackupWindow¥": ¥"${PREFERRED_BACKUP_WINDOW}¥", ¥"autoMaintenance¥": ¥"${AUTO_MAINTENANCE}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

(2) DB インスタンスの復旧確認

(1)実行直後は、復旧した DB インスタンスの status は Build となっているため、DB インスタンスの情報参照 API により、復旧した DB インスタンスの status が Active になったことを確認した後、(3)の手順を実施します。
DB インスタンスの情報参照 API については、「API リファレンス（Application Platform Service 編）」、または、2.3 章(1)を参照してください。

DB インスタンスの復旧が完了するまでの時間は、データ量に依存します。
status が Error となった場合は、復旧に失敗しています。Error となった DB インスタンスを削除し、(1)から再実行してください。
DB インスタンスの削除については、「API リファレンス（Application Platform Service 編）」、または、6.1 章(1)を参照してください。

(3) アプリケーションの修正
アプリケーションのデータベースへの接続先を、(1)で配備した DB インスタンスに変更してください。

(4) 復旧元 DB インスタンスの削除
不要になった復旧元の DB インスタンスを削除してください。
DB インスタンスの削除手順は、6.1 章(1)を参照してください。

5.2 マルチ構成の DB インスタンスの復旧
マルチ構成の DB インスタンスの復旧について説明します。

(1) 異常発生時の動作
マルチ構成の DB インスタンスで異常が発生すると、片系運用になります。
異常発生から片系運用になるまでに約 2 分かかります。
この間、SQL 接続できなくなるため、アプリ作成時は SQL を 2 分以上リトライするように作成してください。
片系で縮退運用中、および、冗長化復旧中は DB インスタンスの状態は Degenerated となります。

(2) DB インスタンスの自動復旧
マルチ構成の DB インスタンスは、片系運用になると自動復旧が動作します。
自動復旧は、DB インスタンス毎に設定される、preferredRecoveryTime の時刻に開始されます。
デフォルトでは、即時復旧に設定されています。
preferredRecoveryTime は、DB インスタンスの変更 API により、設定できます。
自動復旧中は SQL 文発行の性能が落ちるため、運用方針に合わせて、設定してください。
冗長化復旧時間は、データ量に依存し、30 分以上の時間がかかる場合があります。

(3) 自動復旧後の運用

正系の DB インスタンスで異常が発生し、自動復旧が実施された場合は、自動復旧後、DB の正系と副系が逆転して運用されます。このとき、DB インスタンスの状態は Switched となります。

Switched 状態で DB インスタンスの冗長化は復旧されており、運用を継続できます。

Switched 状態から再度、正系と副系を逆転させる場合は、DB インスタンスの再起動 API でフェイルオーバーを実施します。

この間、SQL 文発行ができなくなるため、再起動を実施する際は注意してください。

コマンド例
INSTANCEID=[事前にマルチ構成で作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"failover": "true", "reboot": ""}}'
実行結果例
$ INSTANCEID=test-multi-instance-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID}/action -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d '{"action": {"failover": "true", "reboot": ""}}' HTTP/1.1 202 Accepted （省略）

第 6 章　DB インスタンスの削除

本章の内容は、全体管理者ロールまたは設計・構築者ロールで実施します。

6.1　DB インスタンスの削除

DB インスタンスの削除手順を説明します。

(1)　DB インスタンスの削除

DB インスタンスを削除します。

コマンド例
INSTANCEID=[事前に作成した DB インスタンスの ID（DB インスタンスの作成については 2.2 章(3)を参照してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}"

実行結果例
$ INSTANCEID=test-instance-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 202 Accepted

(2)の手順実施前に、DB インスタンスの情報参照 API により、DB インスタンスが削除されたことを確認してください。

DB インスタンスの情報参照 API を実行し、DB インスタンスの status が Deleted になること、または、DB インスタンスの情報が取得できなくなる（リクエストのステータスが 404 で返却される）ことを確認してください。

DB インスタンスの情報参照 API については、「API リファレンス（Application Platform Service 編）」、または、2.3 章(1)を参照してください。

(2)　DB パラメータグループの削除

DB パラメータグループを削除します。

DB パラメータグループを使用中の DB インスタンスが存在する場合、その DB パラメータグループは削除出来ません。

（レスポンスのステータスが 400 で返却されます。）

どの DB インスタンスからも使用されていない場合のみ、DB パラメータグループを削除してください。

コマンド例
PARAMG_ID=[事前に作成した DB パラメータグループの ID（DB パラメータグループの作成については 2.2 章(2)を参照してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}"

実行結果例
$ PARAMG_ID=test-paramg-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/parametergroups/${PARAMG_ID} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 200 OK

(3) DB サブネットグループの削除

DB サブネットグループを削除します。

DB サブネットグループを使用中の DB インスタンスが存在する場合、その DB サブネットグループは削除出来ません。
（レスポンスのステータスが 400 で返却されます。）

どの DB インスタンスからも使用されていない場合のみ、DB サブネットグループを削除してください。

コマンド例
SUBNETGROUPID=[事前に作成した DB サブネットグループの ID（DB サブネットグループの作成については 2.2 章(1)を参照してください）] ENDPOINT=[データベースサービスのエンドポイント] curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups/${SUBNETGROUPID} -H "X-Auth-Token: ${TOKEN}"

実行結果例
$ SUBNETGROUPID=test-subnetgroup-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X DELETE -i ${ENDPOINT}/v1.0/${TENANTID}/subnetgroups/${SUBNETGROUPID} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 200 OK

付録 A：変更できる DB パラメータの一覧

A.1 変更できる DB パラメータの一覧

データベースに設定するパラメータの一覧を記載します。
4.5 章の手順に従い、データベースのパラメータを変更できます。
ただし、"ismodifiable" が "f" のパラメータは変更できません。
"applymethod" が pending-reboot のパラメータは DB インスタンスの再起動時に変更が適用されます。
"applymethod" が immediate のパラメータはパラメータ値の変更 API 発行後、即時変更が適用されます。
パラメータの詳細については、Enterprise Postgres のマニュアルまたは Symfoware Server のマニュアルを参考に、運用方針に適した値を設定してください。

※“allowedvalues”が “'・・・'”のように、ダブルクォートとシングルクォートで囲まれた値になっているパラメータの値を変更する場合は、パラメータ値の変更 API で設定する設定値についても、“'・・・'”のように、ダブルクォートとシングルクォートで囲む形で設定してください。
※下記表で「 -（ハイフン） 」は「制限なし」を表します。
※下記表で、黄色の網掛けになっている箇所は、メモリの使用量に関わるパラメータとなります。
　そのため、メモリ不足とならないよう、メモリの使用量を見積もってから設定してください。
　メモリの使用量の見積もりについては、A.2 章を参照してください。
※下記表で、max_connections のパラメータは、データベースに同時に接続するコネクション数を指定します。
　データベースサービスもコネクションを使用するため、実際に使用するコネクション数に加えて、シングル構成の場合に 1 コネクション、冗長化構成の場合に 3 コネクションを加算して設定してください。

以下のパラメータは DB パラメータグループのパラメータ値変更時に単位を一緒に指定すると異常になるため、以下に示す単位で数値だけを指定をしてください。

パラメータ名	単位
authentication_timeout	秒
maintenance_work_mem	KB
shared_buffers	8KB のブロック単位
temp_buffers	8KB のブロック単位
work_mem	KB
bgwrite_delay	ミリ秒
wal_writer_delay	ミリ秒
checkpoint_timeout	秒
checkpoint_warning	秒
effective_cache_size	8KB のブロック単位
autovacuum_naptime	秒
deadlock_timeout	ミリ秒

(1) Enterprise Postgres 9.6 で変更できる DB パラメータの一覧

name	default value	min	max	allowedvalues	applymethod	ismodifiable
archive_mode	on	-	-	{on,off}	pending-reboot	f
default_transaction_isolation	"'read committed'"	-	-	{serializable,"'repeatable read'","'read committed'","'read uncommitted'"}	immediate	t
bonjour	off	-	-	-	pending-reboot	f
bonjour_name	''	-	-	-	pending-reboot	f
krb_caseins_users	true	-	-	-	pending-reboot	f
default_text_search_config	'pg_catalog.english'	-	-	-	pending-reboot	t
data_directory	'/userdata/data'	-	-	-	pending-reboot	f
listen_addresses	'*'	-	-	-	pending-reboot	f
dynamic_library_path	'$libdir'	-	-	-	pending-reboot	f
ssl	off	-	-	{on,off}	pending-reboot	t
hot_standby	off	-	-	{on,off}	pending-reboot	f
hot_standby_feedback	off	-	-	{on,off}	pending-reboot	f
krb_server_keyfile	''	-	-	-	pending-reboot	f
local_preload_libraries	''	-	-	-	pending-reboot	f
logging_collector	on	-	-	{on,off}	pending-reboot	f
max_connections	100	7	262143	-	pending-reboot	t
max_files_per_process	1000	25	-	-	pending-reboot	t
max_locks_per_transaction	64	10	1000000	-	pending-reboot	t
max_pred_locks_per_transaction	64	10	2000000	-	pending-reboot	t
max_prepared_transactions	0	0	1000000	-	pending-reboot	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
log_checkpoints	off	-	-	{on,off}	immediate	t
max_wal_senders	12	-	-	-	pending-reboot	f
shared_buffers	4096	16	7602176	-	pending-reboot	t
shared_preload_libraries	'pgx_datamasking,block_alter_system'	-	-	-	pending-reboot	f
ssl_ca_file	''	-	-	-	pending-reboot	f
ssl_ciphers	'HIGH:MEDI-UM:+3DES:!aNULL'	-	-	-	pending-reboot	f
ssl_crl_file	''	-	-	-	pending-reboot	f
ssl_prefer_server_ciphers	on	-	-	-	pending-reboot	f
ssl_ecdh_curve	'prime256v1'	-	-	-	pending-reboot	f
superuser_reserved_connections	5	-	-	-	pending-reboot	f
synchronous_standby_names	''	-	-	-	pending-reboot	f
syslog_ident	'postgres'	-	-	-	pending-reboot	f
tablespace_path_prefix	'/userdata/tblspc'	-	-	-	pending-reboot	f
enable_indexonlyscan	on	-	-	{on,off}	pending-reboot	t
temp_file_limit	-1	-1	2147483647	-	pending-reboot	t
timezone_abbreviations	'Default'	-	-	-	pending-reboot	f
ssl_key_file	'/opt/dbaas/ssl_keys/server.key'	-	-	-	pending-reboot	f
max_standby_archive_delay	30000	-1	2147483647	-	immediate	t
track_activity_query_size	1024	100	102400	-	pending-reboot	t
unix_socket_directory	''	-	-	-	pending-reboot	f
unix_socket_group	''	-	-	-	pending-reboot	f
hba_file	'/userdata/data/pg_hba.conf'	-	-	-	pending-reboot	f
ident_file	'/userdata/data/pg_ident.co	-	-	-	pending-reboot	f

name	default value	min	max	allowedvalues	applymethod	ismodifiable
	nf'					
autovacuum_freeze_max_age	200000000	100000000	2000000000	-	pending-reboot	t
autovacuum_max_workers	3	1	1000	-	pending-reboot	t
unix_socket_permissions	0777	-	-	-	pending-reboot	f
wal_buffers	-1	-1	7602176	-	pending-reboot	t
wal_level	'replica'	-	-	{replica,logical}	pending-reboot	t
external_pid_file	''	-	-	-	pending-reboot	f
track_counts	on	-	-	{on,off}	immediate	t
archive_command	"'/opt/dbaas/client_config/local_backup_script.sh %p %f'"	-	-	-	pending-reboot	f
keystore_location	'/userdata/data'	-	-	-	pending-reboot	f
ssl_cert_file	'/opt/dbaas/ssl_keys/server.crt'	-	-	-	pending-reboot	f
archive_timeout	300	-	-	-	pending-reboot	f
autovacuum_vacuum_threshold	50	0	2147483647	-	immediate	t
backslash_quote	safe_encoding	-	-	{safe_encoding,on,off}	immediate	t
bgwriter_delay	200	10	10000	-	immediate	t
bgwriter_lru_maxpages	100	0	1000	-	immediate	t
bgwriter_lru_multiplier	2.0	0	10	-	immediate	t
bgwriter_flush_after	64	0	256	-	immediate	t
bytea_output	'hex'	-	-	{escape,hex}	immediate	t
check_function_bodies	on	-	-	{on,off}	immediate	t
vacuum_freeze_min_age	50000000	0	1000000000	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
checkpoint_completion_target	0.5	0	1	-	immediate	t
checkpoint_timeout	3600	30	3600	-	immediate	t
checkpoint_warning	30	0	2147483647	-	immediate	t
checkpoint_flush_after	32	0	256	-	immediate	t
client_encoding	sql_ascii	-	-	-	immediate	t
client_min_messages	notice	-	-	{debug5,debug4,debug3,debug2,debug1,log,notice,warning,error}	immediate	t
commit_delay	0	0	100000	-	immediate	t
commit_siblings	5	0	1000	-	immediate	t
constraint_exclusion	partition	-	-	{partition,on,off}	immediate	t
cpu_index_tuple_cost	0.005	0	1.797690000000001e+308	-	immediate	t
cpu_operator_cost	0.0025	0	1.797690000000001e+308	-	immediate	t
cpu_tuple_cost	0.01	0	1.797690000000001e+308	-	immediate	t
cursor_tuple_fraction	0.1	0	1	-	immediate	t
db_user_namespace	off	-	-	{on,off}	immediate	f
deadlock_timeout	1000	1	2147483647	-	immediate	t
debug_pretty_pri	on	-	-	{on,off}	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
nt						
debug_print_parse	off	-	-	{on,off}	immediate	t
debug_print_plan	off	-	-	{on,off}	immediate	t
debug_print_rewritten	off	-	-	{on,off}	immediate	t
default_statistics_target	100	1	10000	-	immediate	t
default_tablespace	''	-	-	-	immediate	f
default_transaction_deferrable	off	-	-	{on,off}	immediate	t
full_page_writes	on	-	-	{on,off}	immediate	t
default_transaction_read_only	off	-	-	{on,off}	immediate	t
default_with_oids	off	-	-	{on,off}	immediate	t
effective_cache_size	16384	1	2147483647	-	immediate	t
effective_io_concurrency	1	0	1000	-	immediate	t
enable_bitmapscan	on	-	-	{on,off}	immediate	t
enable_hashagg	on	-	-	{on,off}	immediate	t
enable_hashjoin	on	-	-	{on,off}	immediate	t
enable_indexscan	on	-	-	{on,off}	immediate	t
enable_material	on	-	-	{on,off}	immediate	t
enable_mergejoin	on	-	-	{on,off}	immediate	t
enable_nestloop	on	-	-	{on,off}	immediate	t
enable_seqscan	on	-	-	{on,off}	immediate	t
enable_sort	on	-	-	{on,off}	immediate	t
enable_tidscan	on	-	-	{on,off}	immediate	t
escape_string_warning	on	-	-	{on,off}	immediate	t
exit_on_error	off	-	-	{on,off}	immediate	f
extra_float_digits	0	-15	-3	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
from_collapse_limit	8	1	2147483647	-	immediate	t
fsync	on	-	-	{on,off}	immediate	t
geqo_effort	5	1	10	-	immediate	t
geqo_generations	0	0	2147483647	-	immediate	t
geqo_pool_size	0	0	2147483647	-	immediate	t
geqo_seed	0.0	0	1	-	immediate	t
geqo_selection_bias	2.0	1.5	2	-	immediate	t
geqo_threshold	12	2	2147483647	-	immediate	t
intervalstyle	'postgres'	-	-	{postgres,postgres_verbose,sql_standard,iso_8601}	immediate	t
join_collapse_limit	8	1	2147483647	-	immediate	t
lc_messages	'en_US.UTF-8'	-	-	-	immediate	t
lc_monetary	'en_US.UTF-8'	-	-	-	immediate	t
lc_numeric	'en_US.UTF-8'	-	-	-	immediate	t
lc_time	'en_US.UTF-8'	-	-	-	immediate	t
geqo	on	-	-	{on,off}	immediate	t
lo_compat_privileges	off	-	-	{on,off}	immediate	f
log_autovacuum_min_duration	-1	-1	2147483647	-	immediate	t
log_connections	off	-	-	{on,off}	immediate	t
log_destination	'stderr'	-	-	-	immediate	f
log_directory	'pg_log'	-	-	-	immediate	f
log_disconnections	off	-	-	{on,off}	immediate	t
log_duration	off	-	-	{on,off}	immediate	t
log_error_verbosity	default	-	-	{terse,default,verbose}	immediate	t
log_executor_sta	off	-	-	{on,off}	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
ts						
log_file_mode	0600	-	-	-	immediate	f
log_filename	'postgresql.log-%Y-%m-%d_%H%M%S'	-	-	{postgresql.log.%Y-%m-%d,postgresql.log.%Y-%m-%d-%H}	immediate	t
log_hostname	off	-	-	{on,off}	immediate	t
log_line_prefix	''	-	-	-	immediate	t
log_lock_waits	off	-	-	{on,off}	immediate	t
log_min_duration_statement	-1	-1	2147483647	-	immediate	t
log_min_error_statement	error	-	-	{debug5,debug4,debug3,debug2,debug1,info,notice,warning,error,log,fatal,panic}	immediate	t
log_min_messages	warning	-	-	{debug5,debug4,debug3,debug2,debug1,info,notice,warning,error,log,fatal,panic}	immediate	t
log_planner_stats	off	-	-	{on,off}	immediate	t
log_replication_commands	off	-	-	{on,off}	immediate	t
log_rotation_age	1440	1	1440	-	immediate	t
log_rotation_size	10240	0	2097151	-	immediate	t
log_statement	'none'	-	-	{none,ddl,mod,all}	immediate	t
log_statement_stats	off	-	-	{on,off}	immediate	t
log_temp_files	-1	-1	2147483647	-	immediate	t
log_truncate_on_rotation	on	-	-	{on,off}	immediate	f
maintenance_work	16384	1024	2147483	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
_mem			647			
log_parser_stats	off	-	-	{on,off}	immediate	t
max_stack_depth	2048	100	9216	-	immediate	t
max_standby_streaming_delay	30000	-1	2147483647	-	immediate	t
password_encryption	on	-	-	{on,off}	immediate	f
quote_all_identifiers	off	-	-	{on,off}	immediate	t
random_page_cost	4.0	0	1.797690000000001e+308	-	immediate	t
restart_after_crash	on	-	-	{on,off}	immediate	f
seq_page_cost	1.0	0	1.79769e+308	-	immediate	t
session_replication_role	'origin'	-	-	{origin,replica,local}	immediate	t
sql_inheritance	on	-	-	{on,off}	immediate	t
standard_conforming_strings	on	-	-	{on,off}	immediate	t
statement_timeout	0	0	2147483647	-	immediate	t
stats_temp_directory	'pg_stat_tmp'	-	-	-	immediate	f
synchronize_seqscans	on	-	-	{on,off}	immediate	t
synchronous_commit	on	-	-	{local,on,off,remote_write, remote_apply}	immediate	t
syslog_facility	'LOCAL0'	-	-	-	immediate	f
tcp_keepalives_count	0	0	2147483647	-	immediate	t
tcp_keepalives_idle	0	0	2147483647	-	immediate	t
tcp_keepalives_interval	0	0	2147483647	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
temp_buffers	1024	0	1073741823	-	immediate	t
temp_tablespaces	''	-	-	-	immediate	f
track_activities	on	-	-	{on,off}	immediate	t
timezone	UTC	-	-	-	immediate	t
track_functions	none	-	-	{none,pl,all}	immediate	t
track_io_timing	off	-	-	{on,off}	immediate	t
track_commit_timestamp	off	-	-	{on,off}	pending-reboot	t
transform_null_equals	off	-	-	{on,off}	immediate	t
array_nulls	on	-	-	{on,off}	immediate	t
authentication_timeout	60	1	600	-	immediate	t
autovacuum	on	-	-	{on,off}	immediate	t
autovacuum_analyze_scale_factor	0.1	0	100	-	immediate	t
autovacuum_analyze_threshold	50	0	2147483647	-	immediate	t
autovacuum_vacuum_cost_delay	20	-1	100	-	immediate	t
autovacuum_naptime	60	1	2147483	-	immediate	t
autovacuum_multixact_freeze_max_age	400000000	10000	2000000000	-	pending-reboot	t
autovacuum_vacuum_cost_delay	20	-1	100	-	immediate	t
autovacuum_vacuum_cost_limit	-1	-1	10000	-	immediate	t
autovacuum_vacuum_scale_factor	0.2	0	100	-	immediate	t
autovacuum_work_mem	-1	-1	2147483647	-	immediate	t
update_process_title	on	-	-	-	immediate	f
vacuum_cost_dela	0	0	100	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
y						
vacuum_cost_limit	200	1	10000	-	immediate	t
vacuum_cost_page_dirty	20	0	10000	-	immediate	t
vacuum_cost_page_hit	1	0	10000	-	immediate	t
vacuum_cost_page_miss	10	0	10000	-	immediate	t
vacuum_defer_cleanup_age	0	0	1000000	-	immediate	t
vacuum_freeze_table_age	150000000	0	2000000000	-	immediate	t
vacuum_multixact_freeze_min_age	5000000	0	1000000000	-	immediate	t
vacuum_multixact_freeze_table_age	150000000	0	2000000000	-	immediate	t
wal_keep_segments	64	-	-	-	immediate	f
wal_receiver_status_interval	10	0	2147483	-	immediate	f
wal_sync_method	fdatasync	-	-	-	immediate	f
wal_writer_delay	200	1	10000	-	immediate	t
wal_writer_flush_after	128	0	2147483647	-	immediate	t
work_mem	1024	64	2147483647	-	immediate	t
xmlbinary	'base64'	-	-	{base64,hex}	immediate	t
xmloption	'content'	-	-	{content,document}	immediate	t
datestyle	"'iso,mdy'"	-	-	-	immediate	t
search_path	"'$user,public'"	-	-	-	immediate	t
log_timezone	UTC	-	-	-	immediate	f
row_security	on	-	-	{on,off}	immediate	t
huge_pages	'try'	-	-	{try,on,off}	pending-reboot	t
dynamic_shared_memory_type	'posix'	-	-	{posix,sysv,mmap,none}	pending-reboot	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
replacement_sort_tuples	150000	0	2147483647	-	immediate	t
max_worker_processes	8	0	1000	-	pending-reboot	t
max_parallel_workers_per_gather	0	0	1024	-	immediate	t
old_snapshot_threshold	-1	-1	86400	-	pending-reboot	t
backend_flush_after	0	0	256	-	immediate	t
wal_compression	off	-	-	{on,off}	immediate	t
wal_log_hints	off	-	-	-	pending-reboot	f
max_wal_size	64	2	2147483647	-	immediate	t
min_wal_size	5	2	2147483647	-	immediate	t
wal_sender_timeout	60000	0	2147483647	-	immediate	f
wal_receiver_timeout	60000	0	2147483647	-	immediate	f
wal_retrieve_retry_interval	5000	1	2147483647	-	immediate	f
max_replication_slots	6	6	262143	-	pending-reboot	t
parallel_tuple_cost	0.1	0	1.797690000000001e+308	-	immediate	t
parallel_setup_cost	1000	0	1.797690000000001e+308	-	immediate	t
min_parallel_relation_size	1024	0	715827882	-	immediate	t
force_parallel_mode	off	-	-	{on,off,regress}	immediate	t
syslog_sequence_numbers	on	-	-	-	immediate	f

name	default value	min	max	allowedvalues	applymethod	ismodifiable
syslog_split_messages	on	-	-	-	immediate	f
cluster_name	''	-	-	-	pending-reboot	f
lock_timeout	0	0	2147483647	-	immediate	t
idle_in_transaction_session_timeout	0	0	2147483647	-	immediate	t
gin_fuzzy_search_limit	0	0	2147483647	-	immediate	t
gin_pending_list_limit	4096	64	2147483647	-	immediate	t
session_preload_libraries	''	-	-	-	immediate	f
operator_precedence_warning	off	-	-	{on,off}	immediate	f

(2) Symfoware Server V12.1 で変更できる DB パラメータの一覧

name	default value	min	max	allowedvalues	applymethod	ismodifiable
archive_mode	on	-	-	{on,off}	pending-reboot	f
default_transaction_isolation	"'read committed'"	-	-	{serializable,"'repeatable read'","'read committed'","'read uncommitted'"}	immediate	t
bonjour	off	-	-	-	pending-reboot	f
bonjour_name	''	-	-	-	pending-reboot	f
krb_caseins_users	true	-	-	-	pending-reboot	f
default_text_search_config	'pg_catalog.english'	-	-	-	pending-reboot	t
data_directory	'/userdata/data'	-	-	-	pending-reboot	f
listen_addresses	'*'	-	-	-	pending-reboot	f
dynamic_library_path	'$libdir'	-	-	-	pending-reboot	f
ssl	off	-	-	{on,off}	pending-reboot	t
hot_standby	off	-	-	{on,off}	pending-reboot	f
hot_standby_feedback	off	-	-	{on,off}	pending-reboot	f
krb_server_keyfile	''	-	-	-	pending-reboot	f
krb_srvname	''	-	-	-	pending-reboot	f
local_preload_libraries	''	-	-	-	pending-reboot	f
logging_collector	on	-	-	{on,off}	pending-reboot	f
max_connections	100	7	2000	-	pending-reboot	t
max_files_per_process	1000	25	-	-	pending-reboot	t
max_locks_per_transaction	64	10	1000000	-	pending-reboot	t
max_pred_locks_per_transaction	64	10	2000000	-	pending-reboot	t
max_prepared_transactions	0	0	1000000	-	pending-reboot	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
log_checkpoints	off	-	-	{on,off}	immediate	t
max_wal_senders	12	-	-	-	pending-reboot	f
replication_timeout	60000	0	2147483647	-	pending-reboot	t
shared_buffers	4096	16	7602176	-	pending-reboot	t
shared_preload_libraries	''	-	-	-	pending-reboot	f
ssl_ca_file	''	-	-	-	pending-reboot	f
ssl_ciphers	'ALL:!ADH:!LOW:!EXP:!MD5:@STRENGTH'	-	-	-	pending-reboot	f
ssl_crl_file	''	-	-	-	pending-reboot	f
superuser_reserved_connections	5	-	-	-	pending-reboot	f
synchronous_standby_names	''	-	-	-	pending-reboot	f
syslog_ident	'postgres'	-	-	-	pending-reboot	f
tablespace_path_prefix	'/userdata/tblspc'	-	-	-	pending-reboot	f
enable_indexonlyscan	on	-	-	{on,off}	pending-reboot	t
temp_file_limit	-1	-1	2147483647	-	pending-reboot	t
timezone_abbreviations	'Default'	-	-	-	pending-reboot	f
ssl_key_file	'/opt/dbaas/ssl_keys/server.key'	-	-	-	pending-reboot	f
max_standby_archive_delay	30000	-1	2147483647	-	immediate	t
track_activity_query_size	1024	100	102400	-	pending-reboot	t
unix_socket_directory	''	-	-	-	pending-reboot	f
unix_socket_group	''	-	-	-	pending-reboot	f
hba_file	'/userdata/data/pg_hba.conf'	-	-	-	pending-reboot	f
ident_file	'/userdata/data/pg_ident.co	-	-	-	pending-reboot	f

name	default value	min	max	allowedvalues	applymethod	ismodifiable
	nf'					
autovacuum_freeze_max_age	200000000	100000000	2000000000	-	pending-reboot	t
autovacuum_max_workers	3	1	1000	-	pending-reboot	t
unix_socket_permissions	0777	-	-	-	pending-reboot	f
wal_buffers	-1	-1	7602176	-	pending-reboot	t
wal_level	hot_standby	-	-	-	pending-reboot	f
external_pid_file	''	-	-	-	pending-reboot	f
track_counts	on	-	-	{on,off}	immediate	t
archive_command	"'/opt/dbaas/client_config/local_backup_script.sh %p %f'"	-	-	-	pending-reboot	f
keystore_location	'/userdata/data'	-	-	-	pending-reboot	f
ssl_cert_file	'/opt/dbaas/ssl_keys/server.crt'	-	-	-	pending-reboot	f
archive_timeout	300	-	-	-	pending-reboot	f
autovacuum_vacuum_threshold	50	0	2147483647	-	immediate	t
backslash_quote	safe_encoding	-	-	{safe_encoding,on,off}	immediate	t
bgwriter_delay	200	10	10000	-	immediate	t
bgwriter_lru_maxpages	100	0	1000	-	immediate	t
bgwriter_lru_multiplier	2.0	0	10	-	immediate	t
bytea_output	'hex'	-	-	{escape,hex}	immediate	t
check_function_bodies	on	-	-	{on,off}	immediate	t
vacuum_freeze_min_age	50000000	0	1000000000	-	immediate	t
checkpoint_completion_target	0.5	0	1	-	immediate	t
checkpoint_segme	20	1	2147483	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
nts			647			
checkpoint_timeout	3600	30	3600	-	immediate	t
checkpoint_warning	30	0	2147483647	-	immediate	t
client_encoding	sql_ascii	-	-	-	immediate	t
client_min_messages	notice	-	-	{debug5,debug4,debug3,debug2,debug1,log,notice,warning,error}	immediate	t
commit_delay	0	0	100000	-	immediate	t
commit_siblings	5	0	1000	-	immediate	t
constraint_exclusion	partition	-	-	{partition,on,off}	immediate	t
cpu_index_tuple_cost	0.005	0	1.797690000000001e+308	-	immediate	t
cpu_operator_cost	0.0025	0	1.797690000000001e+308	-	immediate	t
cpu_tuple_cost	0.01	0	1.797690000000001e+308	-	immediate	t
cursor_tuple_fraction	0.1	0	1	-	immediate	t
db_user_namespace	off	-	-	{on,off}	immediate	f
deadlock_timeout	1000	1	2147483647	-	immediate	t
debug_pretty_print	on	-	-	{on,off}	immediate	t
debug_print_parse	off	-	-	{on,off}	immediate	t
debug_print_plan	off	-	-	{on,off}	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
debug_print_rewritten	off	-	-	{on,off}	immediate	t
default_statistics_target	100	1	10000	-	immediate	t
default_tablespace	''	-	-	-	immediate	f
default_transaction_deferrable	off	-	-	{on,off}	immediate	t
full_page_writes	on	-	-	{on,off}	immediate	t
default_transaction_read_only	off	-	-	{on,off}	immediate	t
default_with_oids	off	-	-	{on,off}	immediate	t
effective_cache_size	16384	1	2147483647	-	immediate	t
effective_io_concurrency	1	0	1000	-	immediate	t
enable_bitmapscan	on	-	-	{on,off}	immediate	t
enable_hashagg	on	-	-	{on,off}	immediate	t
enable_hashjoin	on	-	-	{on,off}	immediate	t
enable_indexscan	on	-	-	{on,off}	immediate	t
enable_material	on	-	-	{on,off}	immediate	t
enable_mergejoin	on	-	-	{on,off}	immediate	t
enable_nestloop	on	-	-	{on,off}	immediate	t
enable_seqscan	on	-	-	{on,off}	immediate	t
enable_sort	on	-	-	{on,off}	immediate	t
enable_tidscan	on	-	-	{on,off}	immediate	t
escape_string_warning	on	-	-	{on,off}	immediate	t
exit_on_error	off	-	-	{on,off}	immediate	f
extra_float_digits	0	-15	-3	-	immediate	t
from_collapse_limit	8	1	2147483647	-	immediate	t
fsync	on	-	-	{on,off}	immediate	t
geqo_effort	5	1	10	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
geqo_generations	0	0	2147483647	-	immediate	t
geqo_pool_size	0	0	2147483647	-	immediate	t
geqo_seed	0.0	0	1	-	immediate	t
geqo_selection_bias	2.0	1.5	2	-	immediate	t
geqo_threshold	12	2	2147483647	-	immediate	t
intervalstyle	'postgres'	-	-	{postgres,postgres_verbose,sql_standard,iso_8601}	immediate	t
join_collapse_limit	8	1	2147483647	-	immediate	t
lc_messages	'en_US.UTF-8'	-	-	-	immediate	t
lc_monetary	'en_US.UTF-8'	-	-	-	immediate	t
lc_numeric	'en_US.UTF-8'	-	-	-	immediate	t
lc_time	'en_US.UTF-8'	-	-	-	immediate	t
geqo	on	-	-	{on,off}	immediate	t
lo_compat_privileges	off	-	-	{on,off}	immediate	f
log_autovacuum_min_duration	-1	-1	2147483647	-	immediate	t
log_connections	off	-	-	{on,off}	immediate	t
log_destination	'stderr'	-	-	-	immediate	f
log_directory	'pg_log'	-	-	-	immediate	f
log_disconnections	off	-	-	{on,off}	immediate	t
log_duration	off	-	-	{on,off}	immediate	t
log_error_verbosity	default	-	-	{terse,default,verbose}	immediate	t
log_executor_stats	off	-	-	{on,off}	immediate	t
log_file_mode	0600	-	-	-	immediate	f
log_filename	'postgresql.log-%Y-%m-%d_%H%M%S'	-	-	{postgresql.log.%Y-%m-%d,postgresql.log.%Y-%	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
				m-%d-%H}		
log_hostname	off	-	-	{on,off}	immediate	t
log_line_prefix	''	-	-	-	immediate	t
log_lock_waits	off	-	-	{on,off}	immediate	t
log_min_duration_statement	-1	-1	2147483647	-	immediate	t
log_min_error_statement	error	-	-	{debug5,debug4,debug3,debug2,debug1,info,notice,warning,error,log,fatal,panic}	immediate	t
log_min_messages	warning	-	-	{debug5,debug4,debug3,debug2,debug1,info,notice,warning,error,log,fatal,panic}	immediate	t
log_planner_stats	off	-	-	{on,off}	immediate	t
log_rotation_age	1440	1	1440	-	immediate	t
log_rotation_size	10240	0	2097151	-	immediate	t
log_statement	'none'	-	-	{none,ddl,mod,all}	immediate	t
log_statement_stats	off	-	-	{on,off}	immediate	t
log_temp_files	-1	-1	2147483647	-	immediate	t
log_truncate_on_rotation	on	-	-	{on,off}	immediate	f
maintenance_work_mem	16384	1024	2147483647	-	immediate	t
log_parser_stats	off	-	-	{on,off}	immediate	t
max_stack_depth	2048	100	9216	-	immediate	t
max_standby_streaming_delay	30000	-1	2147483647	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
password_encryption	on	-	-	{on,off}	immediate	f
quote_all_identifiers	off	-	-	{on,off}	immediate	t
random_page_cost	4.0	0	1.797690000000001e+308	-	immediate	t
restart_after_crash	on	-	-	{on,off}	immediate	f
seq_page_cost	1.0	0	1.79769e+308	-	immediate	t
session_replication_role	'origin'	-	-	{origin,replica,local}	immediate	t
sql_inheritance	on	-	-	{on,off}	immediate	t
ssl_renegotiation_limit	0	0	2147483647	-	immediate	f
standard_conforming_strings	on	-	-	{on,off}	immediate	t
statement_timeout	0	0	2147483647	-	immediate	t
stats_temp_directory	'pg_stat_tmp'	-	-	-	immediate	f
synchronize_seqscans	on	-	-	{on,off}	immediate	t
synchronous_commit	on	-	-	{local,on,off,remote_write}	immediate	t
syslog_facility	'LOCAL0'	-	-	-	immediate	f
tcp_keepalives_count	0	0	2147483647	-	immediate	t
tcp_keepalives_idle	0	0	2147483647	-	immediate	t
tcp_keepalives_interval	0	0	2147483647	-	immediate	t
temp_buffers	1024	0	1073741823	-	immediate	t
temp_tablespaces	''	-	-	-	immediate	f
track_activities	on	-	-	{on,off}	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
timezone	UTC	-	-	-	immediate	t
track_functions	none	-	-	{none,pl,all}	immediate	t
track_io_timing	off	-	-	{on,off}	immediate	t
transform_null_equals	off	-	-	{on,off}	immediate	t
array_nulls	on	-	-	{on,off}	immediate	t
authentication_timeout	60	1	600	-	immediate	t
autovacuum	on	-	-	{on,off}	immediate	t
autovacuum_analyze_scale_factor	0.1	0	100	-	immediate	t
autovacuum_analyze_threshold	50	0	2147483647	-	immediate	t
autovacuum_naptime	60	1	2147483	-	immediate	t
autovacuum_vacuum_cost_delay	20	-1	100	-	immediate	t
autovacuum_vacuum_cost_limit	-1	-1	10000	-	immediate	t
autovacuum_vacuum_scale_factor	0.2	0	100	-	immediate	t
update_process_title	on	-	-	{on,off}	immediate	t
vacuum_cost_delay	0	0	100	-	immediate	t
vacuum_cost_limit	200	1	10000	-	immediate	t
vacuum_cost_page_dirty	20	0	10000	-	immediate	t
vacuum_cost_page_hit	1	0	10000	-	immediate	t
vacuum_cost_page_miss	10	0	10000	-	immediate	t
vacuum_defer_cleanup_age	0	0	1000000	-	immediate	t
vacuum_freeze_table_age	150000000	0	2000000000	-	immediate	t

name	default value	min	max	allowedvalues	applymethod	ismodifiable
wal_keep_segments	64	-	-	-	immediate	f
wal_receiver_status_interval	10	0	2147483	-	immediate	f
wal_sync_method	fdatasync	-	-	-	immediate	f
wal_writer_delay	200	1	10000	-	immediate	t
work_mem	1024	64	2147483647	-	immediate	t
xmlbinary	'base64'	-	-	{base64,hex}	immediate	t
xmloption	'content'	-	-	{content,document}	immediate	t
datestyle	"'iso,mdy'"	-	-	-	immediate	t
search_path	"'$user,public'"	-	-	-	immediate	t
log_timezone	UTC	-	-	-	immediate	f

A.2 メモリの使用量の見積もり式

DB インスタンスで使用するメモリ量は、次の式で見積もってください。

DB インスタンスの使用メモリ量 = 基本使用メモリ量 + DB エンジンの使用メモリ量

(1) 基本使用メモリ量

基本使用メモリ量は、以下になります。

シングル構成の場合：300MB
冗長構成の場合：500MB

(2) DB エンジンの使用メモリ量

DB エンジンで使用するメモリ量は、次の式で見積もってください。

DB エンジンの使用メモリ量 = 共用メモリ量 + ローカルメモリ量

共用メモリ量は、以下を参照してください。

＜Enterprise Postgres 9.6 の場合＞

「PostgreSQL 9.6.2 文書」の「III. サーバの管理」-「18.4 カーネルリソースの管理」-「18.4.1 共有メモリとセマフォ」

＜Symfoware Server 12.1 の場合＞

「PostgreSQL 9.2.4 文書」の「III. サーバの管理」-「17.4 カーネルリソースの管理」-「17.4.1 共有メモリとセマフォ」-「表 17-2. PostgreSQL の共有メモリ使用量」

ローカルメモリ量は、以下を参照してください。

＜Enterprise Postgres 9.6 の場合＞

「FUJITSU Enterprise Postgres 9.6 導入ガイド(サーバ編)」の「付録 H メモリの見積り」-「H.1 FUJITSU Enterprise Postgres で使用するメモリの見積り式」-「ローカルメモリ量」

＜Symfoware Server 12.1 の場合＞

「Symfoware Server 導入ガイド(サーバ編)」の「付録 J メモリの見積り」-「J.1 Symfoware Server で使用するメモリの見積り式」-「ローカルメモリ量」

付録 B：サポートしているクライアント

IaaS データベースサービスでは、以下のバージョンの DB エンジンのクライアントをサポートしています。
DB エンジンのクライアントは、下記からダウンロードしてください。
https://doc.cloud.global.fujitsu.com/jp/iaas/index.html

DB インスタンスのエンジン	サポートする DB エンジンのクライアント
Enterprise Postgres 9.6	Enterprise Postgres Client 9.6
Symfoware Server V12.1	Symfoware Server Client V12.1.1

DB エンジンのクライアントが対応している OS は「FUJITSU Enterprise Postgres 9.6 導入ガイド(クライアント編)」または「Symfoware Server 導入ガイド(クライアント編)」を参照してください。

IaaS で仮想サーバ（DB クライアント）を作成し、DB エンジンのクライアントを導入する場合は、下記の OS を利用できます。
※下記は、2017 年 11 月時点の情報です。

OS の名称	Enterprise Postgres Client 9.6	Symfoware Server Client V12.1.1	備考
Windows Server 2008 Standard Edition R2 64bit 日本語版	○	○	東日本リージョン、西日本リージョンのみ
Windows Server 2008 Enterprise Edition R2 64bit 日本語版	○	○	
Windows Server 2012 Standard Edition 64bit 日本語版	○	○	
Windows Server 2012 R2 Standard Edition 64bit 日本語版	○	○	
CentOS 6.8 64bit (English)	○	○	
CentOS 7.2 64bit (English)	○	○	
CentOS 7.3 64bit (English)	○	○	
Red Hat Enterprise Linux 6.8 64bit (English)	○	○	
Red Hat Enterprise Linux 7.2 64bit (English)	○	○	
Red Hat Enterprise Linux 7.3 64bit (English)	○	○	

※2023 年 3 月をもって Symfoware DB エンジンは提供を終了しました。
作成済の Symfoware DB インスタンスは継続して利用できますが、下記の制限があります。

・DB インスタンスをリストアする場合、リストア元 DB インスタンスの状態が、Shutdown/Error/Deleting/Deleted のどれかの場合だけ実施できます。

・リストア実施後は、リストア元 DB インスタンスは Restored 状態となり、DB インスタンスの情報参照、DB インスタンスの削除だけ実施できます。

・DB インスタンスの変更において、flavorRef パラメータを変更し CPU 数が増加する場合、変更できません。

・DB インスタンスをリストアする場合、リストア元 DB インスタンスよりリストア先 DB インスタンスの CPU 数が増加する flavorRef

パラメータを指定することはできません。

・DB インスタンスのリストアにおいて、リストア元 DB インスタンスがシングル構成の場合は、リストア先 DB インスタンスを冗長構成に指定できません。

・DB インスタンスをリストアし、復旧を確認した後は、速やかにリストア元 DB インスタンスを削除してください。

※Enterprise Postgres DB インスタンスは継続して提供します。

作成済の Enterprise Postgres DB インスタンスにおいて、マイナーバージョンが 3 以下の場合は、マイナーバージョンアップを実施して利用してください。

付録 C：フレーバーID 一覧

IaaS データベースサービスが提供しているフレーバーの一覧は flavor の一覧参照 API で確認できます。
DB インスタンス作成時は flavor の一覧参照 API で取得できるいずれかのフレーバーID を設定し必要に応じて変更してください。

コマンド例
ENDPOINT=[データベースサービスのエンドポイント] LIMIT=[表示するフレーバー数（20～100 の任意の値を設定してください）] curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/flavors?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}"
実行結果例
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ LIMIT=100 $ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/flavors?limit=${LIMIT} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 200 OK （省略） {"flavors":[{"id":"XXXXXXXX", （省略） },{"id":"1101","links":[{"href":"https://XXXXXXXX/flavors/1101","rel":"SELF"},{"href":"https://XXXXXXXX/flavors/1101","rel":"BOOKMARK"}],"name":"S-1"},{"id":"1102","links":[{"href":"https://XXXXXXXX/flavors/1102","rel":"SELF"},{"href":"https://XXXXXXXX/flavors/1102","rel":"BOOKMARK"}],"name":"S-2"}, （省略）]} ★　flavors の name に仮想サーバタイプ※　が出力されます。適切なものを選択し、対応する id を使用します。

※ IaaS が提供している仮想サーバタイプ（フレーバー）は、http://jp.fujitsu.com/solutions/cloud/k5/terms/ の「IaaS サービス仕様書」から確認できます。

付録 D：透過的データ暗号化

D.1 透過的データ暗号化機能について

データベースに格納するデータの暗号化について説明します。

Enterprise Postgres または Symfoware Server の透過的データ暗号化機能を使用すると、OS ファイル内のデータが暗号化されます。

そのため、たとえ不正アクセスによりファイルが盗まれたとしても貴重な情報は保護されます。

Enterprise Postgres または Symfoware Server の透過的データ暗号化機能では、PostgreSQL の暗号化機能に対して下記の違いがあります。

- アプリケーション（SQL 文）の修正が不要です。
- 暗号化してもデータサイズが変わりません。
- 列のデータ型を変更する必要はありません。

データベースに格納するデータは、ファイルに書き出されるときに暗号化され、読み出されるときに復号されます。

これはインスタンスによって自動的に行われるため、ユーザーやアプリケーションが意識することなくキーの管理や暗号化／復号化の処理を実行できます。

［参照］

初回のマスタ暗号化キーの設定、および、キーストアの自動オープンは DB インスタンス作成時に自動的に設定されています。
IaaS データベースサービスでは、マスタ暗号化キーの設定、および、キーストアの自動オープンの設定は不要です。
透過的データ暗号化の詳細については Enterprise Postgres のマニュアルまたは Symfoware Server のマニュアルを参照してください。

D.2 暗号化の範囲について

暗号化の範囲について説明します。

暗号化されるデータ	説明
指定したテーブル空間内のすべてのユーザーデータ	・暗号化はテーブル空間単位で指定します。 ・暗号化テーブル空間内に作成されるテーブル、インデックス、一時テーブル、一時インデックス全てが暗号化されます。
バックアップデータ	・データベースサービスの機能で作成したバックアップデータも暗号化されたままになります。
WAL と一時ファイル	・暗号化されたテーブルとインデックスの更新で生成される WAL も暗号化されます。 ・大きな結合やソートを実行するときには、暗号化データは一時ファイルにも暗号化された形で書き出されます。
冗長化された DB インスタン	・プライマリサーバで暗号化されたデータ（暗号化されたテーブル空間に格納されている

スのデータ同期	テーブル、インデックス）と WAL は、暗号化されたままスタンバイサーバに 転送／格納されます。

［参照］

以下のファイルについては暗号化されません。

- pg_dump および pg_dumpall コマンドの出力ファイル
- COPY コマンドの出力　※
- LISTEN/NOTIFY コマンドでやりとりする通知イベントのペイロード

※　データベースのインポートとエクスポートについて

COPY TO コマンドの出力結果は暗号化されません。

COPY FROM コマンドのインポート先が暗号化テーブル空間にある場合、インポートされたデータは自動的に暗号化され、格納されます。

D.3 テーブル空間の暗号化

本章はデータベースの管理者が実施します。

テーブル空間の暗号化手順について説明します。

暗号化するテーブル空間を作成するとき、実行時パラメータに暗号化アルゴリズムを設定します。

暗号化アルゴリズムは、キー長が 128 または 256 ビットの AES を使用できます。※256 ビットの AES を推奨します。

有効な値は AES128、 AES256、または none（デフォルト値）です。none を設定した場合、暗号化は行いません。

テーブル空間 pg_default および pg_global を暗号化することはできません。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[2.3 章(1)で確認した FQDN 形式のアドレス]
PORT=[2.3 章(1)で確認したポート]
MASTERUSERNAME=[2.3 章(1)で確認したデータベースの管理者名]
ALGORITHM=[暗号化アルゴリズム（AES128 または AES256 を設定）]
SECURE_TABLESPACE=[作成したい暗号化テーブル空間の名前（任意）]
SECURE_DATA=[テーブル空間を格納するための識別名（任意）（半角の英数字と記号で構成される最大長 82 バイトの文字列を作成し、その先頭にスラッシュ(/)を 1 文字付加した文字列を設定してください。）]

# 以降に作成するテーブル空間の暗号化アルゴリズムを設定する SQL
echo "SET tablespace_encryption_algorithm = '${ALGORITHM}';" > /tmp/temp.sql
```

```
echo "CREATE TABLESPACE ${SECURE_TABLESPACE} LOCATION '${SECURE_DATA}';" >> /tmp/temp.sql
# 以降に作成するテーブル空間は暗号化しないようにする SQL
echo "SET tablespace_encryption_algorithm = 'none';" >> /tmp/temp.sql
# 暗号化されているテーブル空間の確認
SQL_STR="SELECT spcname, spcencalgo FROM pg_tablespace ts, pgx_tablespaces tsx WHERE ts.oid = tsx.spctablespace;"

psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -f /tmp/temp.sql
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"

rm /tmp/temp.sql
```

実行結果例

```
$ CLIENT_DIR="/XXXXXXXX/XXXXXXXX"
$ export PATH=${CLIENT_DIR}/bin:${PATH}
$ export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$ FQDN="XXXXXXXX.XXX.XXX"
$ PORT="26500"
$ MASTERUSERNAME="masterusername"
$ ALGORITHM="AES256"
$ SECURE_TABLESPACE="secure_tablespace"
$ SECURE_DATA="/SecureData"

$ echo "SET tablespace_encryption_algorithm = '${ALGORITHM}';" > /tmp/temp.sql
$ echo "CREATE TABLESPACE ${SECURE_TABLESPACE} LOCATION '${SECURE_DATA}';" >> /tmp/temp.sql
$ echo "SET tablespace_encryption_algorithm = 'none';" >> /tmp/temp.sql
$  SQL_STR="SELECT spcname, spcencalgo FROM pg_tablespace ts, pgx_tablespaces tsx WHERE ts.oid = tsx.spctablespace;"

$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -f /tmp/temp.sql
SET
CREATE TABLESPACE
SET
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
      spcname       | spcencalgo
--------------------+------------
 pg_default         | none
 pg_global          | none
 secure_tablespace  | AES256        ★作成したテーブル空間の spcencalgo が AES256 になっていることを確認
```

```
(3 rows)
$ rm /tmp/temp.sql
```

上記暗号化テーブル空間に作成されたテーブルやインデックスは自動的に暗号化されます。

D.4 マスタ暗号化キーの変更

本章はデータベースの管理者が実施します。
マスタ暗号化キーの変更方法について説明します。

同じ暗号化キーを長期間使い続けると、攻撃者に解読の機会を与えてしまうため、一定期間ごとにマスタ暗号化キーを変更することを推奨します。
以下のような SQL 関数を実行してマスタ暗号化キーを設定します。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[2.3 章(1)で確認した FQDN 形式のアドレス]
PORT=[2.3 章(1)で確認したポート]
MASTERUSERNAME=[2.3 章(1)で確認したデータベースの管理者名]

PASSPHRASE=[マスタ暗号化キーのパスフレーズ　※1]
SQL_STR="SELECT pgx_set_master_key('${PASSPHRASE}');"  ※2
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
```

実行結果例

```
$ CLIENT_DIR="/XXXXXXXX/XXXXXXXX"
$ export PATH=${CLIENT_DIR}/bin:${PATH}
$ export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$ FQDN="XXXXXXXX.XXX.XXX"
$ PORT="26500"
$ MASTERUSERNAME="masterusername"

$ PASSPHRASE="XXXXXXXX"
$ SQL_STR="SELECT pgx_set_master_key('${PASSPHRASE}');"
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
pgx_set_master_key
--------------------

(1 row)                    ★このような結果が返ることを確認
```

※1 マスタ暗号化キーの初期パスフレーズには DB インスタンス作成時に設定した管理者パスワードが設定されています。
初期パスフレーズは 8～200 バイトの文字列です。

［注意］

・ DB インスタンス作成時、管理者パスワードが 8 バイトより少なかった場合、初期パスフレーズには管理者パスワードに 0 を右詰めして 8 バイトにした値が設定されます。

例）

管理者パスワード	初期パスフレーズ
pass	pass0000

・ 管理者パスワードが 200 バイトより多かった場合、初期パスフレーズには 201 バイト目以降を除いた 200 バイトの文字列が設定されます。

※2 pgx_set_master_key 関数は、トランザクションブロック内で実行することはできません。

D.5 キーストアのパスフレーズの変更

本章はデータベースの管理者が実施します。
キーストアのパスフレーズ変更方法について説明します。
以下のような SQL 関数を実行してパスフレーズを変更します。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[2.3 章(1)で確認した FQDN 形式のアドレス]
PORT=[2.3 章(1)で確認したポート]
MASTERUSERNAME=[2.3 章(1)で確認したデータベースの管理者名]

OLD_PASSPHRASE=[マスタ暗号化キーのパスフレーズ　※1]
NEW_PASSPHRASE=[今後使用したいマスタ暗号化キーのパスフレーズ（8～200 バイトの半角英数字、全角文字、記号で構成される文字列）（任意）]
SQL_STR="SELECT pgx_set_keystore_passphrase('${OLD_PASSPHRASE}', '${NEW_PASSPHRASE}');"　※2
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
```

実行結果例

```
$ CLIENT_DIR="/XXXXXXXX/XXXXXXXX"
$ export PATH=${CLIENT_DIR}/bin:${PATH}
$ export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$ FQDN="XXXXXXXX.XXX.XXX"
$ PORT="26500"
$ MASTERUSERNAME="masterusername"
```

```
$ OLD_PASSPHRASE="XXXXXXXX"
$ NEW_PASSPHRASE="YYYYYYYY"
$ SQL_STR="SELECT pgx_set_keystore_passphrase('${OLD_PASSPHRASE}', '${NEW_PASSPHRASE}');"
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR}"
pgx_set_keystore_passphrase
------------------------------

(1 row)                          ★このような結果が返ることを確認
```

※1 マスタ暗号化キーの初期パスフレーズには DB インスタンス作成時に設定した管理者パスワードが設定されています。
初期パスフレーズは 8～200 バイトの文字列です。

［注意］

・ DB インスタンス作成時、管理者パスワードが 8 バイトより少なかった場合、初期パスフレーズには管理者パスワードに 0 を右詰めして 8 バイトにした値が設定されます。

例）

管理者パスワード	初期パスフレーズ
pass	pass0000

・ 管理者パスワードが 200 バイトより多かった場合、初期パスフレーズには 201 バイト目以降を除いた 200 バイトの文字列が設定されます。

※2 pgx_set_master_key 関数は、トランザクションブロック内で実行することはできません。

［参照］

ここで設定したパスフレーズを忘れてしまうと、マスタ暗号化キーの変更、キーストアのパスフレーズの変更ができなくなります。
忘れたパスフレーズを取り戻す方法や、データを復号する方法はありません。パスフレーズを忘れないようにしてください。

D.6 データベースのバックアップとリストア/リカバリ

DB インスタンスのバックアップ・リストア時には、透過的データ暗号化機能で使用するマスタ暗号化キー、および、自動オープンキーも自動的にバックアップ・リストアされます。必要な設定手順はありません。

D.7 構築済みアプリケーションの導入

透過的データ暗号化では、アプリケーションを変更することなく、アプリケーション全体のデータを容易に暗号化できます。

構築済みアプリケーションの所有者ユーザーのデフォルトテーブル空間として、暗号化テーブル空間を設定します。

アプリケーションの SQL スクリプトを実行するユーザーのデフォルトテーブル空間を暗号化テーブル空間に設定しておくことで、その SQL スクリプトが生成するオブジェクトは暗号化テーブル空間に配置されます。

D.3 章で作成した暗号化テーブル空間を設定する場合、コマンド例は以下のようになります。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[2.3 章(1)で確認した FQDN 形式のアドレス]
PORT=[2.3 章(1)で確認したポート]
MASTERUSERNAME=[2.3 章(1)で確認したデータベースの管理者名]
APP_USER=[構築済みアプリケーションの所有者ユーザー]
SECURE_TABLESPACE=[D.3 章で作成した暗号化テーブル空間の名前]

SQL_STR1="ALTER USER ${APP_USER} SET default_tablespace = '${SECURE_TABLESPACE}';"
SQL_STR2="ALTER USER ${APP_USER} SET temp_tablespaces = '${SECURE_TABLESPACE}';"
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR1}"
psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR2}"
```

実行結果例

```
$ CLIENT_DIR="/XXXXXXXX/XXXXXXXX"
$ export PATH=${CLIENT_DIR}/bin:${PATH}
$ export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$ FQDN="XXXXXXXX.XXX.XXX"
$ PORT="26500"
$ MASTERUSERNAME="masterusername"
$ APP_USER="appuser"
$ SECURE_TABLESPACE="secure_tablespace"

$ SQL_STR1="ALTER USER ${APP_USER} SET default_tablespace = '${SECURE_TABLESPACE}';"
$ SQL_STR2="ALTER USER ${APP_USER} SET temp_tablespaces = '${SECURE_TABLESPACE}';"
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR1}"
ALTER ROLE
$ psql -p ${PORT} -U ${MASTERUSERNAME} -h ${FQDN} -d postgres -c "${SQL_STR2}"
ALTER ROLE
```

D.8 その他注意事項

・ セキュリティに関する注意事項

DB ログの内容は暗号化されないため、SQL 文に定数を設定していると、その値が DB ログに出力される場合があります。
「マスタ暗号化キー変更」や「キーストアのパスフレーズの変更」の SQL を実行するときには、パスフレーズを含む SQL 文が DB ログに出力される場合があります。
上記を防ぐために、log_min_error_statement などのパラメータ設定を検討して下さい。設定するレベルはデフォルト値「ERROR」以上を推奨します。このとき、正常・警告状態の SQL は DB ログに出力されず、SQL 中の定数も出力されません。

・ エラー時の対処について

付録 D 各章のコマンド実行時にエラーメッセージが返却された場合、下記エラーメッセージが出力された場合は実行した SQL を見直してください。

メッセージ番号	メッセージ内容
15204	passphrase is too short or too long
15207	passphrase of the keystore has been changed
15214	could not open keystore "@1@": passphrase is wrong, or the auto-open keystore was created by another computer or user

上記以外のメッセージについては Enterprise Postgres のマニュアルまたは Symfoware Server のマニュアルの「メッセージ集」を参照してください。
エラー種別(ERROR, FATAL, PANIC)のメッセージが出力され SQL で対処できないと判断した場合は、第 5 章に従い DB インスタンスの復旧を検討してください。
上記の対処を行っても問題が解決しない場合は、サービス窓口まで御連絡ください。

付録 E：各 API を実行可能なロール

IaaS データベースサービスの各 API を実行可能なロールは、下記のようになります。

API	全体管理者	設計・構築者	運用者	監視者
DB インスタンスの作成	○	○	×	×
DB インスタンスの削除	○	○	○	×
DB インスタンスの変更	○	○	○	×
DB スナップショットからの DB インスタンス復旧	○	○	○	×
ポイントイン・タイムリカバリーによる DB インスタンス復旧	○	○	○	×
DB インスタンスの起動	○	○	○	×
DB インスタンスの停止	○	○	○	×
DB インスタンスの再起動	○	○	○	×
参照レプリカ DB インスタンスの作成	○	○	×	×
DB インスタンスの一覧参照	○	○	○	○
DB インスタンスの情報参照	○	○	○	○
DB インスタンスに対する操作のキャンセル	○	○	○	×
DB スナップショットの作成	○	○	○	×
DB スナップショットの削除	○	○	○	×
DB スナップショットの複製	○	○	×	×
DB スナップショットの一覧参照	○	○	○	○
DB スナップショットの情報参照	○	○	○	○
DB ログファイルの一覧参照	○	○	○	○
DB ログファイルの参照	○	○	○	○
DB サブネットグループの作成	○	○	×	×
DB サブネットグループの削除	○	○	×	×
DB サブネットグループの変更	○	○	×	×
DB サブネットグループの一覧参照	○	○	○	○
DB サブネットグループの情報参照	○	○	○	○
DB パラメータグループの作成	○	○	×	×

DB パラメータグループの削除	○	○	×	×
DB パラメータグループの一覧参照	○	○	○	○
DB パラメータグループの情報参照	○	○	○	○
DB パラメータ値の変更	○	○	×	×
イベント通知登録の作成	○	○	×	×
イベント通知登録の削除	○	○	×	×
イベント通知登録の属性変更	○	○	×	×
イベント監視対象の追加/削除	○	○	×	×
イベント通知登録の一覧参照	○	○	○	○
イベント通知登録の情報参照	○	○	○	○
イベント通知カテゴリー覧参照	○	○	○	○
イベント通知の一覧参照	○	○	○	○
DB エンジン情報参照	○	○	○	○
flavor の一覧参照	○	○	○	○
flavor の情報参照	○	○	○	○

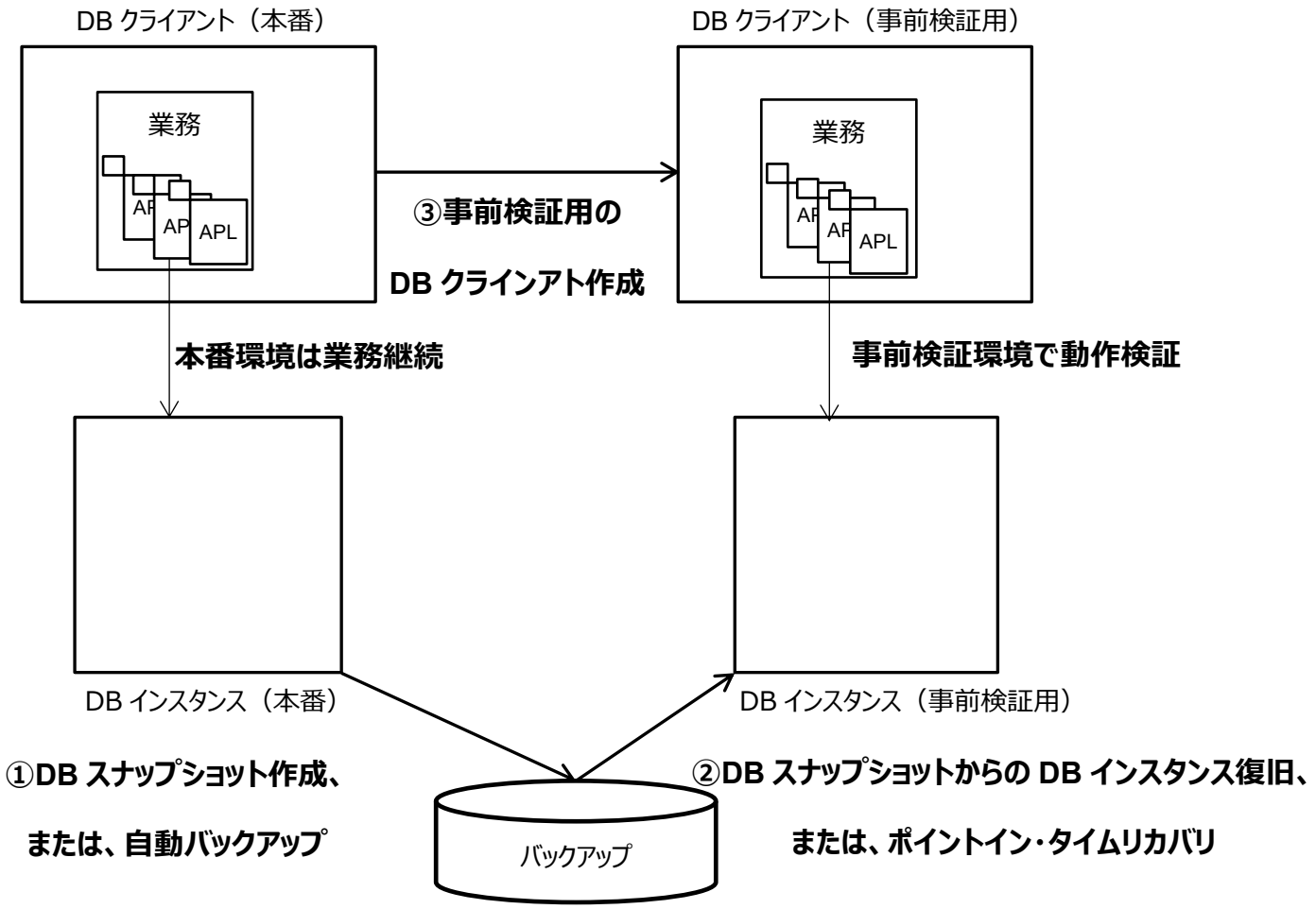
付録 F：既存 DB インスタンスの DB エンジンバージョンアップ

Symfoware Server V12.1 の DB インスタンスから Enterprise Postgres 9.6 の DB インスタンスに移行する場合など、既存 DB インスタンスの DB エンジンのバージョンアップを実施するためには、下記の手順を実施します。

［注意］

・ Symfoware Server V12.1 の DB インスタンスは PostgreSQL 9.2 相当、Enterprise Postgres 9.6 の DB インスタンスは PostgreSQL 9.6 相当であり、既存のアプリケーションに修正が必要な場合があります。そのため、DB エンジンバージョンアップを実施する場合には、複製した DB インスタンスで事前検証を実施し、アプリケーションが正常に動作することを確認してから、本番の DB インスタンスの移行を実施して下さい。事前検証用の DB インスタンスは、5.1 章(1)「DB インスタンスのリカバリ」に記載の、DB スナップショットからの DB インスタンス復旧、または、ポイントイン・タイムリカバリーによる DB インスタンス復旧の手順で作成できます。

[事前検証環境の作成のイメージ]

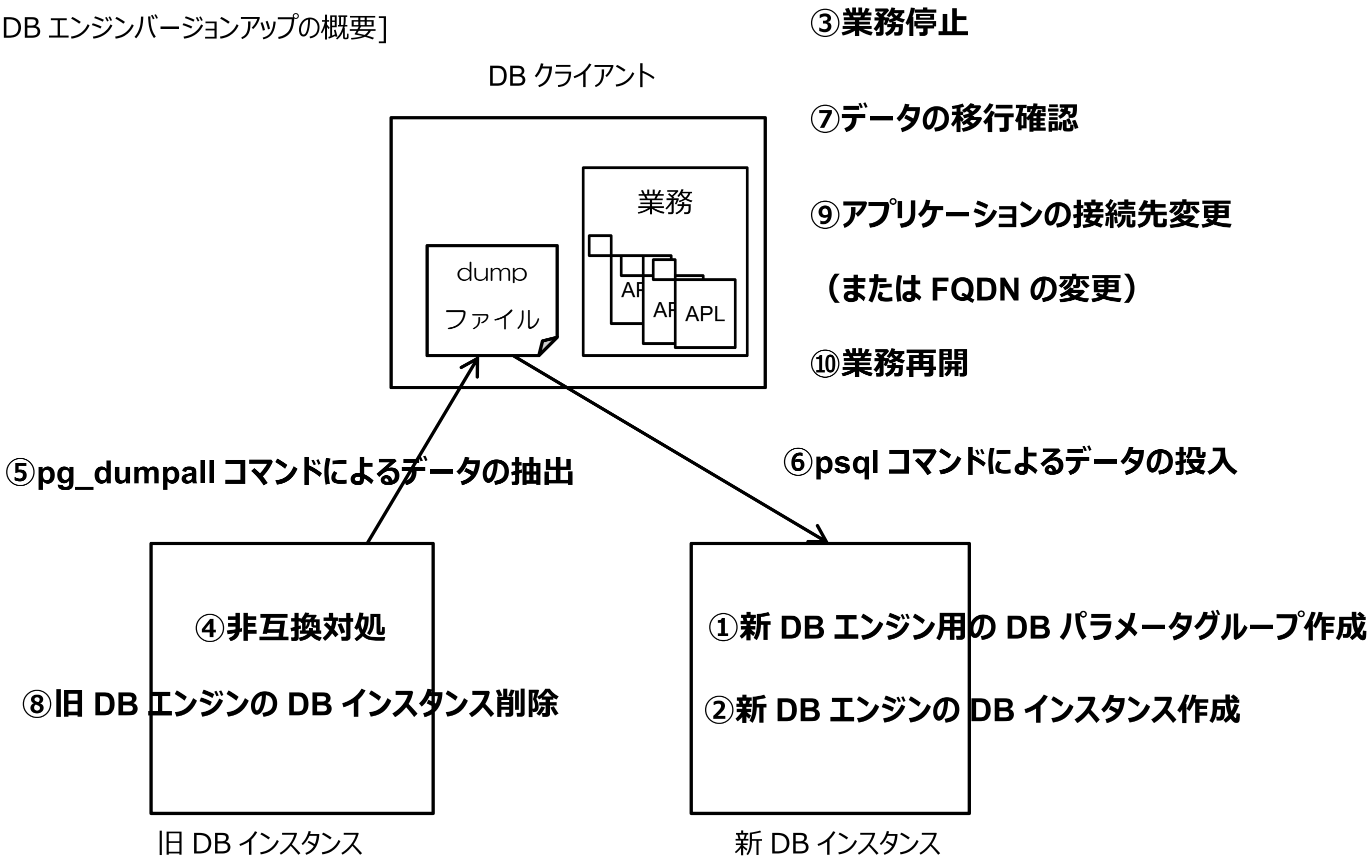


・ DB エンジンバージョンアップを実施する際には、アプリケーションを停止する必要があります。事前検証により付録 F.(3)～(10)の移行時間を把握し、業務の停止時間を見積もってください。

・ 本手順を実施すると、移行元の DB インスタンスとは別に、新規に移行先の DB インスタンスを作成するため、移行元の DB インスタンスとは別に課金が発生します。

・・・

[DB エンジンバージョンアップの概要]



(1) 新 DB エンジン用の DB パラメータグループ作成

4.5 章(1)「DB パラメータグループの作成」、および、(2)「パラメータ値の変更」を参照し、新 DB エンジン用の DB パラメータグループを作成し、パラメータ値を変更してください。

各 DB エンジンで変更可能な DB パラメータは、付録 A.1「変更できる DB パラメータの一覧」を参照してください。

(2) 新 DB エンジンの DB インスタンス作成

2.2 章「DB インスタンスの作成」を参照し、新 DB エンジンで DB インスタンスを作成してください。

ここで、DB パラメータグループは、(1)で作成したものを指定してください。

(3) 業務停止

運用中の旧 DB インスタンスへの業務を停止してください。

(4) 非互換対応

＜Symfoware Server から Enterprise Postgres へバージョンアップする場合＞

Enterprise Postgres では、"pg_"で始まるロール名が予約語となりました。旧 DB で"pg_"で始まるロール名が存在する場合、バージョンアップに失敗するため、"pg_"で始まるロール名を変更・削除してください。

(5) 旧 DB インスタンスから DB データの抽出

旧 DB エンジンに対応するクライアント（Enterprise Postgres Client または Symfoware Server Client）がインストールされた DB クライアントで実行します。

旧 DB インスタンスからデータベースのデータ、スキーマを以下のコマンドを実行し取得してください。

［注意］

・ DB に接続し以下 SQL を実行することで、事前に DB サイズを確認してください。

```
SELECT pg_size_pretty(sum(pg_database_size(datname))) AS dbsize FROM pg_database;
```

なお、下記で使用する pg_dumpall コマンドは、DB のデータを SQL コマンドとして出力するため、実際に作成されるファイルは、上記で確認した DB サイズよりも大きくなることがあります（例えば、integer 型の 2147483647 は、DB データとしては 4 バイトですが、SQL コマンドでは文字列として出力されるため、10 バイトになります）。そのため、ディスク領域に十分余裕のある場所で実行してください。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[旧 DB インスタンスの接続先（FQDN の確認方法は 2.3 章(1)を参照してください）]
PORT=[旧 DB インスタンスのポート番号（ポート番号の確認方法は 2.3 章(1)を参照してください）]
MASTERUSERNAME=[旧 DB インスタンスのデータベースの管理者名]
pg_dumpall -h ${FQDN} -p ${PORT} -U ${MASTERUSERNAME} > db.dump
```

(6) 新 DB インスタンスに DB データの投入

新 DB エンジンのクライアントがインストールされた DB クライアントで実行します。

(2)で作成した新 DB インスタンスに、(5)で抽出したスキーマ、データを投入します。

コマンド例

```
CLIENT_DIR=[DB エンジンのクライアントをインストールしたディレクトリパス]
export PATH=${CLIENT_DIR}/bin:${PATH}
```

```
export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
FQDN=[新 DB インスタンスの接続先（FQDN の確認方法は 2.3 章(1)を参照してください）]
PORT=[新 DB インスタンスのポート番号（ポート番号の確認方法は 2.3 章(1)を参照してください）]
MASTERUSERNAME=[新 DB インスタンスのデータベースの管理者名]
psql -h ${FQDN} -p ${PORT} -U ${MASTERUSERNAME} -d postgres -f db.dump
```

［注意］

・ DB データの投入の際に、下記の拡張機能の登録に失敗したメッセージが出力されますが、本メッセージは無視してください。

- pgx_cpu
- pgx_disk
- pgx_io
- pgx_log
- pgx_memory
- pgx_network
- pgx_network_err
- pgx_paging
- pgx_process
- pg_stat_statements

(7) データの移行確認

新 DB インスタンスに接続し、データの移行が正常に完了していることを確認してください。

DB インスタンスへの接続方法は、2.3 章「DB インスタンスへの接続」を参照してください。

(8) 旧 DB エンジンの DB インスタンス削除

6.1 章「DB インスタンスの削除」を参照し、旧 DB エンジンの DB インスタンスを削除してください。

(9) アプリケーションの接続先変更または FQDN の変更

下記のいずれかの方法で、アプリケーションから新 DB インスタンスに接続できるように変更します。

＜アプリケーションの接続先を変更する場合＞

アプリケーションに設定している接続先を、新 DB インスタンスの FQDN に変更してください。

＜FQDN を変更する場合＞

新 DB エンジンの DB インスタンスの FQDN を、旧 DB エンジンの DB インスタンスの FQDN に変更してください。

また、Heat で作成した DB インスタンスの場合は、DB インスタンスの ID も旧 DB インスタンスの値に変更してください。

コマンド例

```
INSTANCEID=[新 DB エンジンの DB インスタンスの ID]
PRIVATE_ADDRESS=[旧 DB エンジンの DB インスタンスの FQDN]
OLD_INSTANCEID=[旧 DB エンジンの DB インスタンスの ID]
APPLY_IMMEDIATELY=[DB インスタンスの変更の即時適用の有無（true または false を選択）]
ENDPOINT=[データベースサービスのエンドポイント]
curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"privateAddress¥": ¥"${PRIVATE_ADDRESS}¥", ¥"id¥": ¥"${OLD_INSTANCEID}¥", ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}"
```

実行結果例

```
$ INSTANCEID=enterprisepostgres-instance-id
$ PRIVATE_ADDRESS="XXXXXXXX.XXX.XXX"
$ OLD_INSTANCEID=symfoware-instance-id
$ APPLY_IMMEDIATELY=true
$ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com
$ curl -X PUT -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"instance¥": {¥"privateAddress¥": ¥"${PRIVATE_ADDRESS}¥", {¥"id¥": ¥"${OLD_INSTANCEID}¥"}}, ¥"applyImmediately¥": ¥"${APPLY_IMMEDIATELY}¥"}}"
HTTP/1.1 202 Accepted
（省略）
```

(10) 業務再開

業務を再開してください。

[参照]

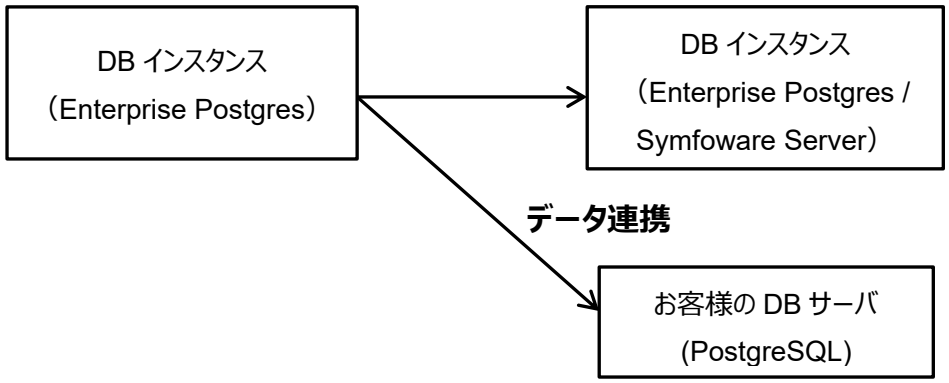
- pg_dumpall コマンドの詳細については、移行元の DB エンジンのドキュメント（「PostgreSQL9.6.2 文書」または「PostgreSQL9.2.4 文書」）の以下を参照してください。
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「pg_dumpall」
- psql コマンドの詳細については、移行先 DB エンジンのドキュメント（「PostgreSQL9.6.2 文書」または「PostgreSQL9.2.4 文書」）の以下を参照してください。
  ・「Ⅵ. リファレンス」-「Ⅱ. PostgreSQL クライアントアプリケーション」-「psql」

付録 G：postgres_fdw

G.1 postgres_fdw について

postgres_fdw を使用することで、DB インスタンスと他の DB インスタンスや外部の PostgreSQL の DB と連携できます。
postgres_fdw は Enterprise Postgres の DB インスタンスで使用できます。

[postgres_fdw 構成例]



postgres_fdw で接続可能な DB は以下です。
- DB インスタンス（Enterprise Postgres および Symfoware Server）
- DB インスタンス以外の DB（Enterprise Postgres、Symfoware Server (Open インタフェース)、PostgreSQL 8.3 以降）

G.2 使用方法

postgres_fdw を使用するためには、以下条件が必要となります。

- 接続元から接続先 DB への通信経路が存在すること（※1）。
- 接続元から接続先 DB の FQDN の名前解決ができること（※2）。（接続先に FQDN を指定する場合）
- 接続元から接続先 DB へのアクセス権限があること。（接続先が DB インスタンスの場合は考慮不要）

（※1）
接続元 DB インスタンスのセキュリティグループに以下のルール設定が必要となります。

ルールの目的	Direction	Protocol	許可相手	ポート
postgres_fdw による DB の通信	egress	tcp	接続先 DB インスタンスのセキュリティグループまたは、接続先のサーバ	接続先 DB のポート番号

名前解決	egress	udp	DNS サーバ	53

接続先が DB インスタンス、もしくは、IaaS 内の VM の場合は、セキュリティグループに以下のルール設定が必要となります。

ルールの目的	Direction	Protocol	許可相手	ポート
postgres_fdw による DB の通信	ingress	tcp	接続元 DB インスタンスのセキュリティグループ	接続先 DB のポート番号

（※2）

DB インスタンス作成時に指定した DB サブネットグループに含まれる全てのサブネットについて、DNS サーバを設定してください。

［参照］

- postgres_fdw の詳細については「PostgreSQL9.6.2 文書」の「F.33. postgres_fdw」を参照してください。

付録 H：リードレプリカの作成

DB インスタンスの負荷を分散し、参照系業務をリードレプリカで実施するなどの用途で、リードレプリカを作成します。下記の手順を実施します。

(1) DB インスタンス作成

2.1 章および 2.2 章を参照し、下記を作成し、DB インスタンスを作成します。

- ネットワーク・サブネット
- 仮想ルータ
- セキュリティグループ
- DBサブネットグループ
- DBパラメータグループ
- DBインスタンス

リードレプリカを作成するためには、DB インスタンスが下記の条件を満たしている必要があります。

- DBエンジンとしてEnterprise Postgres 9.6以降を使用していること
- 冗長化DBインスタンスであること
- 自動バックアップを有効化していること

なお、既に上記条件を満たす DB インスタンスが、テナント上にある場合は、再作成は不要です。
上記条件を満たさない DB インスタンスに対して新たにリードレプリカを作成する場合には、下記を実行してください。

- DBエンジンとしてSymfoware Server V12.1を使用している場合には、付録Fを参照して、Enterprise Postgresにバージョンアップを実施してください。
- シングル構成のDBインスタンスを使用している場合には、4.3章を参照して、冗長化構成のDBインスタンスに移行してください。
- 自動バックアップが有効化されていない場合には、4.4章を参照して、自動バックアップを有効化してください。

(2) DB クライアントから DB インスタンスへ接続

2.3 章(2)を参照し、DB クライアントから DB インスタンスに接続します。

(3) リードレプリカの作成

リードレプリカでは参照系の業務を実施するため、下記について DB インスタンスとは異なる設定にすべきかどうか検討します。

- フレーバー
- DBパラメータグループ

その後、リードレプリカ作成の API により、リードレプリカを作成します。

コマンド例
AZ=[AZ 名（公開されているアベイラビリティゾーンから任意の一つを選択）] INSTANCEID=[リードレプリカの DB インスタンスの ID（任意の値を設定してください）] INSTANCENAME=[リードレプリカの名前（任意の値を設定してください）] SUBNETGROUPID=[DB サブネットグループの ID] SECGRP_ID=[セキュリティグループの ID] PARAMG_ID=[リードレプリカ用の DB パラメータグループの ID] FLAVOR=[フレーバーID（公開されているフレーバーID から任意の一つを選択）] SOURCE_INSTANCEID=[元の DB インスタンス ID] ENDPOINT=[災対側リージョンのデータベースサービスのエンドポイント] curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"action¥":{¥"readreplica¥":¥"¥"}, ¥"instance¥": {¥"availabilityZone¥": ¥"${AZ}¥", ¥"name¥": ¥"${INSTANCENAME}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"id¥": ¥"${INSTANCEID}¥", ¥"readReplicaSrcDBInstance¥": {¥"id¥": ¥"${SOURCE_TENANTID}¥"}}}"
実行結果例
$ AZ=jp-east-1a $ INSTANCEID=replica-instance-id $ INSTANCENAME=replica-instance-name $ SUBNETGROUPID=test-subnetgroup-id $ FLAVOR=1101 $ SECGRP_ID=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX $ PARAMG_ID=replica-paramg-id $ SOURCE_INSTANCEID=test-instance-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X POST -i ${ENDPOINT}/v1.0/${TENANTID}/instances -H "X-Auth-Token: ${TOKEN}" -H "Content-Type: application/json" -d "{¥"action¥":{¥"readreplica¥":¥"¥"}, ¥"instance¥": {¥"availabilityZone¥": ¥"${AZ}¥", ¥"name¥": ¥"${INSTANCENAME}¥", ¥"subnetGroupId¥": ¥"${SUBNETGROUPID}¥", ¥"flavorRef¥": ¥"${FLAVOR}¥", ¥"parameterGroupId¥": ¥"${PARAMG_ID}¥", ¥"securityGroupIds¥": [{¥"securityGroupId¥": ¥"${SECGRP_ID}¥"}], ¥"id¥": ¥"${INSTANCEID}¥", ¥"readReplicaSrcDBInstance¥": {¥"id¥": ¥"${SOURCE_TENANTID}¥"}}}" HTTP/1.1 202 Accepted （省略）

リードレプリカ作成 API を実行すると、元の DB インスタンスからデータがコピーされてリードレプリカが作成されます。そのため、リードレプリカの作成時間は、データ量に依存します。

リードレプリカの作成が完了したかどうかの確認は、リードレプリカに対して DB インスタンスの情報参照の API を実行してください。

作成が完了すると、status が Active になり、replicaStatus が Replicating となります。リードレプリカの作成完了後に DB インスタンスでデータを更新すると、リードレプリカに随時更新ログが送信され、非同期でデータが反映されます。

コマンド例
INSTANCEID=[リードレプリカの DB インスタンスの ID] ENDPOINT=[データベースサービスのエンドポイント] curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}"
実行結果例
$ INSTANCEID=replica-instance-id $ ENDPOINT=https://database.jp-east-1.cloud.global.fujitsu.com $ curl -X GET -i ${ENDPOINT}/v1.0/${TENANTID}/instances/${INSTANCEID} -H "X-Auth-Token: ${TOKEN}" HTTP/1.1 200 OK （省略） {"instance":{"volume":{"size":10,"type":"M1"},"id":"test-instance-id","name":"test-instance-name","description":null,"multiAZ":false,"multi":true,"port":26500,"preferredBackupWindow":"17:50-18:20","preferredMaintenanceWindow":"mon:01:46-mon:02:16","preferredRecoveryTime":{"applyImmediately":true,"time":null},"securityGroupIds":[{"securityGroupId":"XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"}],"parameterGroupId":"DefaultGroup-12.1","backupRetentionPeriod":0,"autoMinorVersionUpgrade":true,"engineVersion":"9.6","engineMinorVersion":"0","autoMaintenance":true,"availabilityZone":"XXXXXXXX","subnetGroupId":"test-subnetgroup-id","publiclyAccessible":false,"engine":"enterprisepostgres","masterUserName":"masterusername","characterSet":"UTF8","collate":"C","created":"2016-07-12T02:09:01Z","flavor":{"id":"1101","links":[{"href":"http://XXXXXXXX/v1.0/XXXXXXXX/flavors/1101","rel":"SELF"},{"href":"http://XXXXXXXX/XXXXXXXX/flavors/1101","rel":"BOOKMARK"}]},"links":[{"href":"http://XXXXXXXX/v1.0/XXXXXXXX/instances/test-instance-id","rel":"SELF"},{"href":"http://XXXXXXXX/XXXXXXXX/instances/test-instance-id","rel":"BOOKMARK"}],"status":"Active","updated":null,"privateAddress":"XXXXXXXX.XXX.XXX","privateIp":"XX.XX.XX.XX","publicAddress":null,"publicIp":null,"subPrivateIp":null,"subPublicIp":null,"pendingModifiedValues":null,"readReplicaDBInstanceIdentifiers":[],"readReplicaSrcDBInstanceIdentifier":null,"secondaryAvailabilityZone":null,"readReplicaDBInstances":[],"readReplicaSrcDBInstance":{"region":"jp-east-1","tenantId":"YYYYYYYY-YYYY-YYYY-YYYY-YYYYYYYYYYYY","id":"test-instance-id"},"replicaStatus":"Replicating"}}

(4) DB クライアントからリードレプリカへ接続

2.3 章(2)を参照し、DB クライアントからリードレプリカに接続します。このとき使用する FQDN は、リードレプリカの FQDN となります。

付録 I：DB インスタンスの状態

DB インスタンス（データベース仮想サーバ）の状態は、下記の表のとおりです。

状態	内容
Active	【説明】 データベース仮想サーバが正常で、使用可能な状態です。 【データベース接続】 データベースに接続できます。 【API リクエスト】 本状態のデータベース仮想サーバに対して、すべての API を実行できます。
Backup	【説明】 データベース仮想サーバをバックアップ中です。 【データベース接続】 データベースに接続できます。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下の操作はできません。 ・データベース仮想サーバの起動/停止/再起動 ・データベース仮想サーバの変更 ・参照レプリカデータベース仮想サーバの作成 ・DB スナップショットの作成
Restart_Required	【説明】 反映されていないデータベース仮想サーバの変更やセキュリティアップデート、マイナーバージョンアップがある状態です。反映する場合は、データベース仮想サーバの再起動が必要です。 【データベース接続】 データベースに接続できる状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、すべての API を実行できます。
Switched	【説明】 データベースがフェイルオーバーしたことで、正系と副系が逆転して運用しています。 【データベース接続】 データベースに接続できる状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、すべての API を実行できます。

Degenerated	【説明】 データベース仮想サーバが片系で縮退運用中、および、冗長化復旧中の状態です。 冗長化復旧時間は、データ量に依存し、30 分以上の時間がかかる場合があります。 preferredRecoveryTime に設定した時刻に自動復旧が開始されるとイベント通知により復旧開始の通知を行います。 【データベース接続】 データベースに接続できる状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下の操作はできません。 ・データベース仮想サーバの変更 ・参照レプリカデータベース仮想サーバの作成
Build	【説明】 データベース仮想サーバを作成中です。 【データベース接続】 データベースに接続できません。 【API リクエスト】 本状態のデータベース仮想サーバに対して、POST/PUT/DELETE の操作はできません。
Deleting	【説明】 データベース仮想サーバを削除中です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、POST/PUT/DELETE の操作はできません。
Deleted	【説明】 データベース仮想サーバは削除済みですが、そのデータベース仮想サーバに紐付いた DB スナップショットが存在していることを示している状態です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、POST/PUT/DELETE の操作はできません。

Failed	【説明】 データベース仮想サーバが不正な状態です。 データベース仮想サーバが異常な状態になっている可能性があるため、「DB スナップショットからのデータベース仮想サーバ復旧」API、または、「ポイントイン・タイムリカバリーによるデータベース仮想サーバの復旧」API により復旧してください。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 ・DB ログファイルの一覧参照/参照 ・データベース仮想サーバの停止 ・データベース仮想サーバの再起動
Error	【説明】 データベース仮想サーバが不正な状態です。 「データベース仮想サーバの作成」API、「DB スナップショットからのデータベース仮想サーバ復旧」API、「ポイントイン・タイムリカバリーによるデータベース仮想サーバの復旧」API のうちのどれかを実行し、Error となった場合は、再作成を実施してください。 その他の場合は、サービス窓口まで連絡してください。 【データベース接続】 データベースに接続できる場合もありますが、不正な状態の為、再作成または復旧が必要です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 ・DB ログファイルの一覧参照/参照
Modifying	【説明】 データベース仮想サーバの変更中です。 【データベース接続】 変更したパラメータによってデータベース仮想サーバの再起動を伴い、再起動中はデータベースに接続できません。再起動を伴うパラメータは「4.4　DB インスタンスの変更」を参照してください。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT/DELETE の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 ・DB ログファイルの一覧参照/参照

Resize	【説明】 データベース仮想サーバのデータディスクのサイズ変更中です。 【データベース接続】 データベース仮想サーバの再起動を伴い、再起動中はデータベースに接続できません。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT/DELETE の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 ・DB ログファイルの一覧参照/参照
Shutdown	【説明】 データベース仮想サーバを停止済みの状態です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下の操作はできません。 ・データベース仮想サーバの変更（applyImmediately=true）　※ ・参照レプリカデータベース仮想サーバの作成 ・DB スナップショットの作成 ※Degenerated 状態からデータベース仮想サーバを停止した場合は、applyImmediately パラメータに関わらず、「データベース仮想サーバの変更」API を実行できません。
Reboot	【説明】 データベース仮想サーバを再起動中です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT/DELETE の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧
Starting	【説明】 データベース仮想サーバを起動中です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT/DELETE の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧

Stopping	【説明】 データベース仮想サーバを停止中です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、以下を除く POST/PUT/DELETE の操作はできません。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧
Restoring	【説明】 Symfoware データベース仮想サーバのリストア中です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、POST/PUT/DELETE の操作はできません。
Restored	【説明】 Symfoware データベース仮想サーバのリストア実施済みの状態です。 【データベース接続】 データベースに接続できない状態です。 【API リクエスト】 本状態のデータベース仮想サーバに対して、POST/PUT の操作はできません。

付録 J：制限事項・注意事項一覧

IaaS データベースサービスの制限事項・注意事項については、下記の内容を参照してください。

No	内容
1	データベースは FQDN でアクセスしてください。
2	自動メンテナンス設定で指定した時間にパッチが適用された場合、約 10 分間の SQL エラーとなる時間帯が発生します。
3	データベース仮想サーバの冗長構成の変更(シングル→冗長化、冗長化→シングル)は実行できません。シングル構成を冗長構成に変更したい場合、元のデータをバックアップ後、冗長構成のデータベース仮想サーバを新規に作成し、バックアップからデータをリストアしてください。
4	冗長構成のデータベース仮想サーバの名前(name)を変更しないでください。 名前を変更したい場合、希望する名前でデータベース仮想サーバを新規に作成し、バックアップからデータをリストアしてください。
5	冗長構成のデータベース仮想サーバにおいて、データベース仮想サーバの変更 API は主系運用に戻した状態で実施してください。従系運用時にデータベース仮想サーバの変更 API が実施された場合、データベース仮想サーバのステータスが Error となり、バックアップデータからのリカバリが必要となります。
6	フェイルオーバーを設定("failover": true)してデータベース仮想サーバを再起動する場合は、必ずデータベース仮想サーバの起動中に実行してください。停止中に実行された場合、データベース仮想サーバは SQL を受け付けなくなり、バックアップデータからのリカバリが必要となります。
7	冗長構成のデータベース仮想サーバにおいて DB パラメータを変更する場合、パラメータ変更時は再起動が必要ですが、この際、フェイルオーバーを設定("failover": true)して再起動をしないでください。また、再起動中はデータベースに接続しないでください。これらの操作が実行された場合、設定変更が反映されないおそれがあります。
8	「low storage」を指定したイベント通知設定はご利用できません。
9	再起動が必要なパラメータを含むパラメータグループ変更後は、データベース仮想サーバのステータスが Restart_Required になりませんが、データベース仮想サーバの再起動が必要です。 再起動が行われない場合、変更は反映されません。
10	パラメータグループの参照 API では利用者が変更したパラメータ値は表示されますが、デフォルト値は表示されません。デフォルト値を確認する場合、SQL 文「show all;」を発行して確認してください。
11	DB ログファイルの一覧参照 API にて取得するログファイル名は 100 件以上表示できません。72 時間の間に出力されるログファイル数が 100 件以下になるようにログローテーションを設定してください。
12	DB パラメータにはデータベース仮想サーバの再起動を必要とするものがあります。 それらのパラメータを変更する際には、「即時適用(applyImmediately : true)」を指定するか、変更操作後にデータベース仮想サーバを再起動してください。 再起動が行われない場合、設定は反映されません。 設定値を元の値に戻す場合にも、必ず再起動を実施して一度設定が反映された後に実施してください。
13	データベース仮想サーバの作成、データベース仮想サーバの変更などの API について、ブロックストレージのタイプは「M1」を必ず指定してください。
14	リストアによって FQDN および IP アドレスが変更されます。

15	ポイントイン・タイムリカバリーによる復旧を実施する際に、指定した時点以降の更新量が 16MB に満たない場合はリカバリが行われません。ポイントイン・タイムリカバリー前に「select pg_switch_xlog();」SQL を実施することでリカバリが可能になります。
16	データベースは 20%程度の余裕を持った容量で運用してください。ディスク容量が枯渇した場合、ステータスが Error になり、リストアまたはデータベースの再作成が必要となります。
17	データベース仮想サーバに割り当てるセキュリティグループには、そのセキュリティグループ自身に対するデータベース用ポートの送受信を許可する設定を含めてください。
18	1 つのデータベースサービスごとに 1 つのサブネットに対してシングル構成の場合は 5 個、冗長化構成の場合は 6 個の IP アドレスを確保できる構成にしてください。 また、AZ 間冗長化構成の場合は、主系データベースを配備した AZ 側のサブネットから 5 個、従系データベースを配備した AZ 側のサブネットから 5 個の IP アドレスを確保できる構成にしてください。
19	下記のいずれかの操作を実施した場合は、データベース仮想サーバの再起動を実施してください。再起動が行われない場合、DB ログが文字化けする場合があります。 DB パラメータ「lc_messages」を変更した DB パラメータグループを使用して下記の操作を実施した場合 ・データベース仮想サーバの作成 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 ・参照レプリカデータベース仮想サーバの作成 データベース仮想サーバに適用している DB パラメータグループの、DB パラメータ「lc_messages」を変更後に下記の操作を実施した場合 ・データベース仮想サーバのデータディスクのサイズの変更、および、変更の適用 ・データベース仮想サーバのポート番号の変更、および、変更の適用（シングル構成の場合） ・データベース仮想サーバのパッチ適用
20	下記いずれかの操作によりリストアを実施すると、リストアで作成されたデータベース仮想サーバは、最新のマイナーバージョンで作成されます。 ・DB スナップショットからのデータベース仮想サーバ復旧 ・ポイントイン・タイムリカバリーによるデータベース仮想サーバ復旧 復旧元のデータベース仮想サーバで最新のマイナーバージョンアップが実施されていなかった場合、リストアで作成されたデータベース仮想サーバの情報参照を実施すると、表示上、誤ったマイナーバージョンが表示されます。
21	データベース仮想サーバのパッチ適用の実施中は、Starting/Reboot 状態となります。 パッチ適用の実施中は、データベースに接続しないでください。 パッチ適用の実施中にデータベースに接続すると、パッチ適用に失敗する場合があります。 データベースへの接続は、パッチ適用が終了し、Active/Switched/Restart_Required 状態に遷移したあとで実施してください。

FUJITSU Hybrid IT Service FJcloud-O IaaS
データベースサービス ユーザーズガイド 1.２０版
発行日 2025 年３月