



# **EDB Postgres for Kubernetes**

リリース *1.15.0*

**EDB**

2022 年 05 月 19 日



# 目次

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <b>第 1 章 EDB Postgres for Kubernetes Release notes</b>          | <b>3</b>  |
| 1.1 EDB Postgres for Kubernetes 1.15.0 release notes . . . . .  | 4         |
| 1.2 EDB Postgres for Kubernetes 1.14.0 release notes . . . . .  | 5         |
| 1.3 EDB Postgres for Kubernetes 1.13.0 release notes . . . . .  | 6         |
| 1.4 EDB Postgres for Kubernetes 1.12.0 release notes . . . . .  | 7         |
| 1.5 EDB Postgres for Kubernetes 1.11.0 release notes . . . . .  | 8         |
| 1.6 EDB Postgres for Kubernetes 1.10.0 release notes . . . . .  | 10        |
| 1.7 EDB Postgres for Kubernetes 1.9.2 release notes . . . . .   | 11        |
| 1.8 EDB Postgres for Kubernetes 1.9.1 release notes . . . . .   | 12        |
| 1.9 EDB Postgres for Kubernetes 1.9.0 release notes . . . . .   | 12        |
| 1.10 EDB Postgres for Kubernetes 1.8.0 release notes . . . . .  | 13        |
| 1.11 EDB Postgres for Kubernetes 1.7.1 release notes . . . . .  | 14        |
| 1.12 EDB Postgres for Kubernetes 1.7.0 release notes . . . . .  | 14        |
| 1.13 EDB Postgres for Kubernetes 1.6.0 release notes . . . . .  | 15        |
| 1.14 EDB Postgres for Kubernetes 1.5.1 release notes . . . . .  | 16        |
| 1.15 EDB Postgres for Kubernetes 1.5.0 release notes . . . . .  | 16        |
| 1.16 EDB Postgres for Kubernetes 1.4.0 release notes . . . . .  | 17        |
| 1.17 EDB Postgres for Kubernetes 1.3.0 release notes . . . . .  | 18        |
| 1.18 EDB Postgres for Kubernetes 1.2.1 release notes . . . . .  | 18        |
| 1.19 EDB Postgres for Kubernetes 1.2.0 release notes . . . . .  | 19        |
| 1.20 EDB Postgres for Kubernetes 1.1 release notes . . . . .    | 19        |
| 1.21 EDB Postgres for Kubernetes 1.0 release notes . . . . .    | 19        |
| <b>第 2 章 ‘Differences with CloudNativePG v1’</b>                | <b>21</b> |
| <b>第 3 章 ‘Differences with CloudNativePG’</b>                   | <b>23</b> |
| <b>第 4 章 ‘Installation and upgrades’</b>                        | <b>25</b> |
| 4.1 Kubernetes でのインストール . . . . .                               | 25        |
| 4.1.1 演算子マニフェストを直接使用する . . . . .                                | 25        |
| 4.2 展開に関する詳細 . . . . .                                          | 26        |
| <b>第 5 章 ‘Quickstart’</b>                                       | <b>27</b> |
| 5.1 パート 1-ローカル Kubernetes / Openshift プレイグラウンドのセットアップ . . . . . | 28        |
| 5.1.1 CodeReady コンテナ (CRC) . . . . .                            | 28        |
| 5.2 パート 2-Kubernetes 用の EDB Postgres のインストール . . . . .          | 28        |
| 5.3 パート 3- PostgreSQL クラスターをデプロイする . . . . .                    | 28        |
| <b>第 6 章 Installation, Configuration and Deployment Demo</b>    | <b>31</b> |
| 6.1 Kubernetes 用 EDB Postgres をインストールする . . . . .               | 32        |
| 6.2 PostgreSQL クラスターをデプロイする . . . . .                           | 33        |

|               |                                             |           |
|---------------|---------------------------------------------|-----------|
| 6.3           | kubectl-cnp プラグインをインストールする                  | 37        |
| 6.4           | フェイルオーバーのテスト                                | 38        |
| 6.4.1         | さらに読む                                       | 40        |
| <b>第 7 章</b>  | <b>‘Operator configuration’</b>             | <b>41</b> |
| 7.1           | 利用可能オプション                                   | 41        |
| 7.2           | 演算子構成マップの定義                                 | 43        |
| 7.3           | 演算子シークレットの定義                                | 43        |
| 7.4           | 構成をリロードするために演算子を再起動する                       | 43        |
| 7.5           | PPROF HTTP サーバー                             | 44        |
| <b>第 8 章</b>  | <b>EDB Audit logs</b>                       | <b>47</b> |
| <b>第 9 章</b>  | <b>‘EDB Postgres for Kubernetes Plugin’</b> | <b>49</b> |
| 9.1           | 機能                                          | 49        |
| 9.2           | インストール                                      | 49        |
| 9.2.1         | サポートされているアーキテクチャ                            | 49        |
| 9.3           | 使用                                          | 50        |
| 9.3.1         | ステータス                                       | 50        |
| 9.3.2         | プロモート                                       | 54        |
| 9.3.3         | 証明書                                         | 54        |
| 9.3.4         | 再起動                                         | 55        |
| 9.3.5         | リロード                                        | 55        |
| 9.3.6         | メンテナンス                                      | 55        |
| 9.3.7         | レポート                                        | 56        |
| 9.3.8         | レポート演算子                                     | 56        |
| 9.3.9         | レポートクラスター                                   | 58        |
| <b>第 10 章</b> | <b>‘Red Hat OpenShift’</b>                  | <b>61</b> |
| 10.1          | OpenShift の重要な概念                            | 62        |
| 10.1.1        | クラスターサービスバージョン (CSV)                        | 63        |
| 10.1.2        | マルチテナント管理の制限                                | 63        |
| 10.2          | Web コンソールからのインストール                          | 63        |
| 10.2.1        | クラスター全体のインストール                              | 67        |
| 10.2.2        | 単一プロジェクトのインストール                             | 67        |
| 10.3          | oc CLI を介したインストール                           | 69        |
| 10.3.1        | oc を使用したクラスター全体のインストール                      | 70        |
| 10.3.2        | OpenShift CLI (oc) のインストール                  | 71        |
| 10.4          | 事前定義された RBAC オブジェクト                         | 72        |
| 10.4.1        | カスタムリソース定義 (CRD)                            | 72        |
| 10.4.2        | サービスアカウント                                   | 73        |
| 10.4.3        | クラスターの役割                                    | 74        |
| 10.4.4        | Kubernetes CRD の EDB Postgres のクラスターロール     | 74        |
| 10.4.5        | Kubernetes CRD のクラスターロール                    | 75        |
| 10.5          | ユーザーと権限                                     | 78        |
| 10.6          | プーラーイメージのカスタマイズ                             | 79        |
| <b>第 11 章</b> | <b>‘Free evaluation’</b>                    | <b>81</b> |

|               |                                            |           |
|---------------|--------------------------------------------|-----------|
| 11.1          | PostgreSQL の評価                             | 81        |
| 11.2          | EDB Postgres Advanced Server の評価           | 81        |
| <b>第 12 章</b> | <b>‘License and license keys’</b>          | <b>83</b> |
| 12.1          | 会社レベルのライセンスキー                              | 83        |
| 12.1.1        | YAML マニフェストを介した Kubernetes インストール          | 83        |
| 12.1.2        | OLM を使用した OpenShift および Kubernetes のインストール | 84        |
| 12.2          | クラスターレベルのライセンスキー                           | 84        |
| 12.3          | クラスターレベルでのライセンスキーシークレット                    | 85        |
| 12.4          | ライセンスの有効期限が切れるとどうなりますか                     | 85        |
| <b>第 13 章</b> | <b>‘API Reference’</b>                     | <b>87</b> |
| 13.1          | AffinityConfiguration                      | 89        |
| 13.2          | AzureCredentials                           | 90        |
| 13.3          | バックアップ                                     | 91        |
| 13.4          | BackupConfiguration                        | 91        |
| 13.5          | BackupList                                 | 92        |
| 13.6          | BackupSource                               | 92        |
| 13.7          | BackupSpec                                 | 92        |
| 13.8          | BackupStatus                               | 92        |
| 13.9          | BarmanObjectStoreConfiguration             | 93        |
| 13.10         | BootstrapConfiguration                     | 94        |
| 13.11         | BootstrapInitDB                            | 95        |
| 13.12         | BootstrapPgBaseBackup                      | 96        |
| 13.13         | BootstrapRecovery                          | 96        |
| 13.14         | 証明書の構成                                     | 97        |
| 13.15         | 証明書ステータス                                   | 98        |
| 13.16         | クラスター                                      | 98        |
| 13.17         | ClusterCondition                           | 99        |
| 13.18         | ClusterList                                | 99        |
| 13.19         | ClusterSpec                                | 99        |
| 13.20         | ClusterStatus                              | 102       |
| 13.21         | ConfigMapKeySelector                       | 103       |
| 13.22         | ConfigMapResourceVersion                   | 103       |
| 13.23         | DataBackupConfiguration                    | 104       |
| 13.24         | EPASConfiguration                          | 104       |
| 13.25         | EmbeddedObjectMetadata                     | 105       |
| 13.26         | ExternalCluster                            | 105       |
| 13.27         | GoogleCredentials                          | 105       |
| 13.28         | InstanceID                                 | 106       |
| 13.29         | LDAPBindAsAuth                             | 106       |
| 13.30         | LDAPBindSearchAuth                         | 106       |
| 13.31         | LDAPConfig                                 | 107       |
| 13.32         | LocalObjectReference                       | 107       |
| 13.33         | MonitoringConfiguration                    | 107       |
| 13.34         | NodeMaintenanceWindow                      | 108       |
| 13.35         | PgBouncerIntegrationStatus                 | 108       |

|                                             |     |
|---------------------------------------------|-----|
| 13.36 PgBouncerSecrets . . . . .            | 108 |
| 13.37 PgBouncerSpec . . . . .               | 108 |
| 13.38 PodMeta . . . . .                     | 109 |
| 13.39 PodTemplateSpec . . . . .             | 110 |
| 13.40 プーラー . . . . .                        | 110 |
| 13.41 PoolerIntegrations . . . . .          | 111 |
| 13.42 PoolerList . . . . .                  | 111 |
| 13.43 PoolerSecrets . . . . .               | 111 |
| 13.44 PoolerSpec . . . . .                  | 111 |
| 13.45 PoolerStatus . . . . .                | 112 |
| 13.46 PostgresConfiguration . . . . .       | 112 |
| 13.47 RecoveryTarget . . . . .              | 112 |
| 13.48 ReplicaClusterConfiguration . . . . . | 113 |
| 13.49 RollingUpdateStatus . . . . .         | 113 |
| 13.50 S3Credentials . . . . .               | 113 |
| 13.51 ScheduledBackup . . . . .             | 114 |
| 13.52 ScheduledBackupList . . . . .         | 114 |
| 13.53 ScheduledBackupSpec . . . . .         | 115 |
| 13.54 ScheduledBackupStatus . . . . .       | 115 |
| 13.55 SecretKeySelector . . . . .           | 115 |
| 13.56 SecretVersion . . . . .               | 116 |
| 13.57 SecretsResourceVersion . . . . .      | 116 |
| 13.58 StorageConfiguration . . . . .        | 117 |
| 13.59 WalBackupConfiguration . . . . .      | 117 |

EDB EDB Postgres for Kubernetes は、演算子によって設計、開発、サポートされ、ネイティブストリーミングレプリケーションを使用して、プライマリ/スタンバイアーキテクチャを備えた高可用性 Postgres データベースクラスターのライフサイクル全体をカバーしています。これは、オープンソースの CloudNativePG 演算子に基づいており、EDB Postgres Advanced Server を使用したオラクルとの互換性や、IBM Power や OpenShift などの追加サポートプラットフォームなどの追加価値を提供します。

利用可能な主な機能は次のとおりです：-高可用性のための Kubernetes API 統合-フェイルオーバーおよびレプリカの自動再作成による自己修復-スケールアップ/ダウン機能による容量管理-スケジュールされたメンテナンスのための計画的なスイッチオーバー-Kubernetes サービス定義の読み書き Postgres マイナーバージョンと演算子アップグレードのローリング更新-継続的なバックアップとポイントインタイムリカバリ-PgBouncer を使用した接続プーリング-すぐに使用可能な統合メトリックエクスポート-マルチプルの Kubernetes クラスターにわたる PostgreSQL レプリケーション-OpenShift の Red Hat 認定演算子

演算子は Cloud Native PostgreSQL から名前が変更されました。Cloud Native PostgreSQL の既存のユーザーは、基礎となるコンポーネントとリソースが変更されていないため、変更は発生しません。



# 第1章 EDB Postgres for Kubernetes Release notes

Kubernetes 用 EDB Postgres の文書には、マイナーリリースやパッチなど、Kubernetes 用 EDB Postgres のメジャーバージョンが記載されています。リリースノートには、各リリースの新機能に関する情報が記載されています。マイナーリリースまたはパッチリリースで導入された新しい機能の場合、コンテンツにはその機能を導入したリリースも示されます。

| Version                                                 | Release date | Upstream merges                                                  |
|---------------------------------------------------------|--------------|------------------------------------------------------------------|
| <i>EDB Postgres for Kubernetes 1.15.0 release notes</i> | 2022 Apr 21  | Upstream <i>EDB Postgres for Kubernetes 1.15.0 release notes</i> |
| <i>EDB Postgres for Kubernetes 1.14.0 release notes</i> | 2022 Mar 25  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.13.0 release notes</i> | 2022 Feb 17  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.12.0 release notes</i> | 2022 Jan 11  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.11.0 release notes</i> | 2021 Dec 15  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.10.0 release notes</i> | 2021 Nov 11  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.9.2 release notes</i>  | 2021 Oct 15  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.9.1 release notes</i>  | 2021 Sep 30  | NA                                                               |
| <i>EDB Postgres for Kubernetes 1.9.0 release notes</i>  | 2021 Sep 28  | NA                                                               |

次のページに続く

表 1 – 前のページからの続き

| Version                                                | Release date | Upstream merges |
|--------------------------------------------------------|--------------|-----------------|
| <i>EDB Postgres for Kubernetes 1.8.0 release notes</i> | 2021 Sep 13  | NA              |
| <i>EDB Postgres for Kubernetes 1.7.1 release notes</i> | 2021 Aug 11  | NA              |
| <i>EDB Postgres for Kubernetes 1.7.0 release notes</i> | 2021 Jul 28  | NA              |
| <i>EDB Postgres for Kubernetes 1.6.0 release notes</i> | 2021 Jul 12  | NA              |
| <i>EDB Postgres for Kubernetes 1.5.1 release notes</i> | 2021 Jun 11  | NA              |
| <i>EDB Postgres for Kubernetes 1.5.0 release notes</i> | 2021 Jun 17  | NA              |
| <i>EDB Postgres for Kubernetes 1.4.0 release notes</i> | 2021 May 18  | NA              |
| <i>EDB Postgres for Kubernetes 1.3.0 release notes</i> | 2021 Apr 23  | NA              |
| <i>EDB Postgres for Kubernetes 1.2.1 release notes</i> | 2021 Apr 6   | NA              |
| <i>EDB Postgres for Kubernetes 1.2.0 release notes</i> | 2021 Mar 31  | NA              |
| 1.1.0                                                  | 2021 Mar 3   | NA              |
| 1.0.0                                                  | 2021 Feb 4   | NA              |

## 1.1 EDB Postgres for Kubernetes 1.15.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type           | Description                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Upstream merge | コミュニティ CloudNativePG 1.15 と統合されました。コミュニティ <a href="#">:ref:`EDB Postgres for Kubernetes Release notes`</a> <EDB Postgres for Kubernetes Release notes>を参照してください。 |

## 1.2 EDB Postgres for Kubernetes 1.14.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type             | Description                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Feature          | Barman Cloud 2.19 で導入された機能を活用することにより、バックアップとリカバリのために Google Cloud Storage をネイティブにサポートします。                                             |
| Feature          | Cluster オブジェクトに LastBackupSucceeded 条件を序文することにより、バックアップの視認性が向上しました。                                                                   |
| Feature          | ホットスタンバイの重要なパラメーターの更新のサポート: max_connections、max_prepared_transactions、max_locks_per_transaction、max_wal_senders、max_worker_processes。 |
| Feature          | Cluster オブジェクトに進行中のオンラインアップグレードフェーズを追加して、演算子のオンラインアップグレードが進行中であることを示します。                                                              |
| Feature          | S3 オブジェクトストレージの認証情報を提供する代替方法として AWS IAM ロールを継承する機能。                                                                                   |
| Feature          | Pooler の authQuerySecret および証明書の不透明なシークレットのサポート。                                                                                      |
| Feature          | デフォルトの PostgreSQL バージョンを 14.2 に更新しました。                                                                                                |
| Feature          | メンテナンス名前付けの kubectl cnp プラグインに新しいコマンドを追加して、メンテナンスウィンドウを Kubernetes クラスター全体の 1 つまたはすべてのネームスペースのクラスターに設定します。                            |
| Container images | 最新の PostgreSQL および EDB Postgres Advanced Server コンテナには、Barman Cloud 2.19 が含まれています。                                                    |
| Security fix     | OpenShift を含む、Operator Lifecycle Manager を使用した名前空間付き演算子のインストールに対する RBAC の強化。OpenShift ユーザーは、このバージョンに更新することをお勧めします。                    |
| Bug fix          | インスタスマネージャが、オリジナルの pg_control ファイルのコピーを保存することにより、中断された pg_rewind をリトライできるようにします。                                                      |
| Bug fix          | pg_rewind を実行する前に、古い PID ファイルをクリーンアップします。                                                                                             |
| Bug fix          | primary_conninfo のキーによる並べ替えを強制して、13 より前の PostgreSQL バージョンでのランダム再起動を回避します。                                                             |
| Bug fix          | 調整時に ServiceAccount の変更（ラベル、注釈など）を保持します。                                                                                              |
| Bug fix          | imagePullPolicy デフォルト値の強制を無効にします。                                                                                                     |
| Bug fix          | WAL セグメントサイズの initDB バリデーションを改善します。                                                                                                   |
| Bug fix          | LSN が指定されたクラスターを回復するときに、targetLSN オプションを適切にハンドルします。                                                                                   |
| Bug fix          | サーバー証明書と CA 証明書の両方で証明書チェーンを許可することにより、カスタム TLS 証明書のバリデーションを修正します。                                                                      |

## 1.3 EDB Postgres for Kubernetes 1.13.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type             | Description                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Feature          | Snappy 圧縮のサポート。Snappy は、より低い圧縮率を使用してオブジェクトストアへのアップロード速度を向上させるバックアップ用の高速圧縮オプションです。                                                         |
| Feature          | Barman オブジェクトストアにアップロードされたファイルのタグ付けのサポート。この機能を使用するには、オペランドイメージに Barman 2.18 が必要です。クラスタ削除後のバックアップの。                                        |
| Feature          | status.conditions を使用したクラスタのステータスの拡張。条件 ContinuousArchiving は、クラスタが WAL ファイルのアーカイブを開始したことを示します。                                           |
| Feature          | 追加情報を使用して kubectl の cnp プラグインの status コマンドを改善します。クラスタのステータスを示す Cluster Summary セクションと、クラスタで使用されている証明書のステータスとエキスパイアまでの証明書ステータスセクションを追加します。 |
| Feature          | 新しい barman-cloud-check-wal- アーカイブコマンドをサポートして、新しいクラスタを作成するときに空でないバックアップ先を検出します。                                                            |
| Feature          | MONITORING_QUERIES_SECRET 構成変数を使用して、シークレットを使用してデフォルトのモニタリングクエリを追加するサポートを追加します。                                                            |
| Feature          | ユーザが AppArmor を使用してコンテナの権限を制限できるようにします (AppArmor サポートでデプロイされた Kubernetes クラスタ上)。                                                          |
| Feature          | kubectl の cnp プラグインに Windows プラットフォームのサポートを追加し、Windows x86 および ARM でプラグインを使用できるようになりました。                                                  |
| Feature          | Kubernetes 1.18 および非推奨の API バージョンのサポートを終了                                                                                                 |
| Container images | PostgreSQL コンテナには、Barman 2.18 が含まれます。                                                                                                     |
| Security fix     | 所有者とスーパーユーザのシークレット内にユーザー名フィールドの一貫性チェックを追加します。以前は、悪意のあるユーザがシークレットを使用して任意の PostgreSQL ユーザのパスワードを変更できました。                                    |
| Bug fix          | Postgres ポッドからステータスを取得するコードのメモリーリークを修正しました。                                                                                               |
| Bug fix          | クラッシュ後の PostgreSQL の自己再起動を無効にします。インスタンスコントローラは、PostgreSQL インスタンスのライフサイクルを処理します。                                                           |
| Bug fix          | バリデーション webhook の spec.postgresUID および spec.postgresGID フィールドの変更を防ぎます。クラスタの作成後にこれらのフィールドを変更すると、PostgreSQL をスタートできなくなります。                  |
| Bug fix          | バックアップおよび WAL アーカイビングハンドリングコードからのログの冗長性を減らします。                                                                                            |
| Bug fix          | まだ初期化されていないときにクラスタが「正常」とマークされるバグを修正します。                                                                                                   |
| Bug fix          | 非常に高い WAL 稼働レートを備えたクラスタ内のスタンバイサーバが調整されたら、ストリーミングにスイッチことができます。                                                                             |

次のページに続く

表 4 – 前のページからの続き

| Type    | Description                                                  |
|---------|--------------------------------------------------------------|
| Bug fix | まれにクラッシュにつながる可能性のある PostgreSQL ポッドのスタートアップ中の競合条件を修正します。      |
| Bug fix | クラスター内の最初の PVC の初期化に失敗する可能性のある競合条件を修正します。                    |
| Bug fix | クラスターをレプリカとして参加させる前に、降格したばかりのプライマリポッドの余分なリスタートを削除します。        |
| Bug fix | バックアップから復旧するときに、レプリケーションに敏感な PostgreSQL 構成パラメーターを正しくハンドルします。 |
| Bug fix | クラスター作成中の PostgreSQL 設定のバリデーションの欠落を修正。                       |

## 1.4 EDB Postgres for Kubernetes 1.12.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                                                                                     |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature | サポートされている Kubernetes ディストリビューションのリストに Kubernetes 1.23 を追加し、1.17 のエンドツーエンドテストを削除して、2020 年 12 月に Kubernetes プロジェクトによるサポートを終了しました。                                                 |
| Feature | 2 秒の接続タイムアウトと 30 秒の通信タイムアウトを追加して、ネットワークに問題がある場合のポッドステータスチェックの応答性を改善します。この変更により、演算子がポッドが失敗と宣言する前にポッドのステータスをレポートするのを待つ時間に制限が設定され、フェイルオーバーオペレーションの堅牢性と予測可能性が向上します。                 |
| Feature | .spec.inheritedMetadata フィールドをクラスターに導入して、ユーザがクラスターによって生成されたすべてのオブジェクトに適用されるラベルと注釈を指定できるようにします。                                                                                  |
| Feature | インスタンスのステータスを計算するときに実行されるクエリの数減らします。                                                                                                                                            |
| Feature | PgBouncer の準備プロンプトを追加します。                                                                                                                                                       |
| Feature | Azure プロトコルを使用する場合、Barman のバックアップオブジェクトストアのエンドポイントのカスタム証明機関のサポートを追加します。                                                                                                         |
| Bug fix | フェイルオーバー中に、すべての WAL ストリーミング接続が閉じられるまで、新しいプライマリが選択されるのを待ちます。演算子は、デフォルトで wal_sender_timeout および wal_receiver_timeout を 5 秒に設定して、プライマリにネットワークの問題がある場合にスタンバイ・ノードが迅速に通知 make ようにします。 |

次のページに続く

表 5 – 前のページからの続き

| Type    | Description                                                                                                                                                                                             |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bug fix | レプリカクラスターの PostgreSQL インスタンスに対して「archive_mode」を「always」に設定することにより、レプリカクラスターの WAL アーカイビング戦略を変更してローリング更新を修正します。次に、現在およびターゲットの指定プライマリからのみ WAL のアップロードを制限します。WAL は切り替え中に 2 回アップロードされる場合がありますが、これは問題ではありません。 |
| Bug fix | レプリカクラスターソース内の Barman のバックアップオブジェクトストアのエンドポイントのカスタム証明機関のサポートを修正します。                                                                                                                                     |
| Bug fix | クラスター名前空間のデフォルトのモニタリング構成マップに固定名前を使用します。                                                                                                                                                                 |
| Bug fix | 何らかの理由でデフォルトの Webhook が機能していない場合、演算子は調整サイクル中にもデフォルトでクラスターを更新するようになりました。                                                                                                                                 |
| Bug fix | リソース要求と制限の比較を修正して、すべての調整サイクルですべてのポッドの更新につながるまれな問題を修正します。                                                                                                                                                |
| Bug fix | Webhook からのログメッセージを改善して、オブジェクトのネームスペースも含めます。                                                                                                                                                            |
| Bug fix | すべての調整ループのスタートに「デフォルト」メッセージのロギングを停止します。                                                                                                                                                                 |
| Bug fix | enablePodMonitor が false の場合、すべての調整サイクルで PodMonitor の削除のロギングを停止します。                                                                                                                                     |
| Bug fix | ポッドに到達できない場合、アーキテクチャの不一致の可能性について文句を言わないでください。                                                                                                                                                           |

## 1.5 EDB Postgres for Kubernetes 1.11.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                                                                                                                                                                                                        |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature | 並列 WAL アーカイビングと復元：backupObjectStore.maxParallel オプションを導入して、WAL アーカイビング（PostgreSQL の archive_command による）と WAL 復元の両方で実行される並列ジョブの最大数を設定することにより、データベースが高書き込みシステムでの WAL 生成に対応できるようにします（restore_command によって）。並列復元オプションを使用すると、必要な WAL ファイルをフェッチして、順次ではなく並列にリプレイすることにより、新しく昇格したスタンバイをより早く準備状態にすることができます。 |
| Feature | モニタリング用のデフォルトのメトリックセット：default-monitoring と呼ばれる新しい ConfigMap は、演算子の同じネームスペースに自動的にデプロイされ、デフォルトで既存の Postgres クラスターに追加されます。そのような動作は、オペレーターの構成で MONITORING_QUERIES_CONFIGMAP パラメータを設定するか、クラスターレベルで.spec を使用してグローバルに変更できます。モニタリング。                                                                      |

次のページに続く

表 6 – 前のページからの続き

| Type    | Description                                                                                                                                                                                                       |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature | disableDefaultQueries オプション（デフォルトで false に設定）。                                                                                                                                                                    |
| Feature | クラスターのモニタリングセクションに enablePodMonitor オプションを導入して、PodMonitor リソースを自動的に管理し、Prometheus とシームレスに統合します。                                                                                                                   |
| Feature | 目的の stopDelay 時間の前半にスマートシャットダウンを実行し、残りの半分に高速シャットダウンを実行してから、ポッドが Kubernetes によって殺される前に、PostgreSQL のシャットダウンプロセスを改善します。                                                                                              |
| Feature | switchoverDelay オプションを追加して、新しいプライマリを昇格する前に元のプライマリに与えられる時間を制御して正常にシャットダウンし、すべての WAL ファイルをアーカイブします（デフォルトでは、Cloud Native PostgreSQL は権限データの永続性まで無期限に待機します）。                                                          |
| Feature | ローリング更新を発行して、PostgreSQL クラスターのリソース要求と制限の変更を処理します。                                                                                                                                                                 |
| Feature | 追加情報を使用して、kubectl の cnf プラグインの status コマンドを改善します。ストリーミングレプリケーションステータス、データベースの合計サイズ、クラスター内のインスタンスのロール。                                                                                                             |
| Feature | PostgreSQL の共有メモリ管理のために dynamic_shared_memory_type および shared_memory_type パラメーターの設定を有効にすることにより、多くの並列ワーカーでワークロードのサポートを強化します。                                                                                       |
| Feature | クラスターレベルで定義されたラベルと注釈を、ポッドなどの関連リソースに伝播します（削除はサポートされていません）。                                                                                                                                                         |
| Feature | Kubelet によって削除されたポッドを自動的に削除します。                                                                                                                                                                                   |
| Feature | ENABLE_AZURE_PVC_UPDATES 演算子構成オプションを使用して、必要に応じてクラスターのローリング更新を発行することにより、Azure の永続ボリュームの自動サイズ変更を管理します（デフォルトで無効）。                                                                                                    |
| Feature | 特定の Postgres クラスターで無効に設定されている場合、調整ループが実行されないようにする k8s.enterprisedb.io/reconciliationLoop アノテーションを導入します。                                                                                                           |
| Feature | initdb ブートストラップメソッドのパートとして postInitApplicationSQL オプションを導入して、クラスターが作成された直後にスーパーユーザとしてメインアプリケーションデータベースで実行される SQL クエリのリストを指定します。                                                                                   |
| Feature | EDB Postgres Advanced 14.2 のサポート。                                                                                                                                                                                 |
| Bug fix | Liveness プローブは、PostgreSQL サーバーのスタートアッププロセスを正しく処理するようになりました。これにより、少数のお客様から報告された問題が修正され、WAL ファイルをリカバリして整合性のある状態に戻すニーズがある再起動されたスタンバイサーバに影響がありますが、ライブプロブのタイムアウトが発生して CrashLoopBackOff にポッドが残る前にそれを行うことができませんでしたステータス。 |
| Bug fix | Liveness プローブは、タイムラインの転換後に現在のプライマリと再調整するために pg_rewind を使用するニーズがある以前のプライマリのケースを正しく処理するようになりました。これにより、新しいスタンバイのポッドが Kubernetes によって繰り返し殺されるのを修正します。                                                                 |

次のページに続く

表 6 – 前のページからの続き

| Type    | Description                                                                           |
|---------|---------------------------------------------------------------------------------------|
| Bug fix | Postgres ポッドからのクライアント側の調整を削減します（例：優先度と公平性ではなく、クライアント側の調整のために 1.182388649 を待機、要求：GET）。 |
| Bug fix | 提供されているものを使用して、OpenShift および OLM インストールで公開キーインフラストラクチャ（PKI）の初期化を無効にします。               |
| Bug fix | リスタートが必要な構成パラメータを変更する場合は、常にプライマリを最後のままにしてください。                                        |
| Bug fix | ジョブが正常に完了した後にのみ PVC を準備完了としてマークし、PVC 初期化の競合条件を防ぎます。                                   |
| Bug fix | 期限切れの Webhook TLS シークレットを更新するときは、正しい公開キーを使用してください。                                    |
| Bug fix | LSN をパースするときのオーバーフローを修正します。                                                           |
| Bug fix | スタートアップに古い PID ファイルを削除します。                                                            |
| Bug fix | Pooler リソースは、演算子で定義された imagePullSecret を継承します（存在する場合）。                                |

## 1.6 EDB Postgres for Kubernetes 1.10.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                                                                                                                                                                           |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature | PgBouncer による接続プーリング：Pooler リソースとコントローラを導入して、ローカル PostgreSQL クラスターのコネクションプーラとして使用される PgBouncer デプロイメントを自動的に管理します。この機能には、TLS クライアント/サーバー接続、パスワード認証、高可用性、ポッドテンプレートサポート、キー PgBouncer パラメーターの構成、PAUSE / RESUME、JSON フォーマットでのロギング、統計、プール、およびリストの Prometheus エクスポーターが含まれます。 |
| Feature | バックアップ保持ポリシー：バックアップのリカバリウィンドウ保持ポリシーの定義をサポートします（30 日間のリカバリウィンドウを保証の「30d」など）。                                                                                                                                                                                           |
| Feature | 演算子のインプレース更新：インスタスマネージャのインプレースオンライン更新を導入します。これにより、演算子の更新後にクラスター全体のローリング更新を実行する必要がなくなります。デフォルトでは、このオプションは無効になっています（詳細については、文書を参照してください）。                                                                                                                               |
| Feature | initdb ブートストラップメソッドでカスタマイズできるオプションのリストを dataChecksums、エンコーディング、localeCollate、localeCTYPE、walSegmentSize に制限します。これにより、オプション配列は廃止され、v2 API で削除される予定です。                                                                                                                  |

次のページに続く

表 7 – 前のページからの続き

| Type         | Description                                                                                                                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature      | initdb ブートストラップメソッドのパートとして postInitTemplateSQL オプションを導入して、クラスターが作成された直後にスーパーユーザとして template1 データベースで実行される SQL クエリのリストを指定します。この機能により、クラスターで作成されたすべてのアプリケーションデータベースにデフォルトオブジェクトを含めることができます。 |
| Feature      | インスタンス Prometheus エクスポーターに追加された新しいデフォルトメトリック：Postgres バージョン、クラスター名前、およびバックアップカタログに基づく回復性の最初のポイント。                                                                                         |
| Feature      | 障害が発生した後、バックアップの取得を再試行してください。                                                                                                                                                             |
| Feature      | 演算子が頻繁に更新されないオペランドイメージに存在しない最近導入された機能（保持ポリシーや Azure Blob Container ストレージなど）を呼び出さないオーダーに、Barman Cloud の機能に関する認識を構築します。                                                                     |
| Feature      | cnp プラグインの status コマンドの出力を、バックアップに関する情報と統合します。                                                                                                                                            |
| Feature      | PVC のステータス（初期化中または準備中）を報告する新しい注釈を導入します。                                                                                                                                                   |
| Feature      | バックアップオブジェクトを含む、クラスターで生成されたすべてのオブジェクトの k8s.enterprisedb.io/clusterlabel にクラスター名前を設定します。                                                                                                   |
| Feature      | 非推奨 API バージョン postgresql.k8s.enterprisedb.io/v1alpha1 のサポートを、クラスター、バックアップ、およびスケジュールされたバックアップの種類にドロップします。                                                                                  |
| Feature      | デフォルトのオペランドイメージを PostgreSQL 14.2 に設定します。                                                                                                                                                  |
| Security fix | 演算子コンテナ securityContext の allowPrivilegeEscalation を false に設定します。                                                                                                                        |
| Bug fix      | 単一インスタンスクラスターのメンテナンス中は、プライマリ PodDisruptionBudget を無効にします。                                                                                                                                 |
| Bug fix      | リカバリ操作中は正しい証明書証明権限（CA）を使用してください。                                                                                                                                                          |
| Bug fix      | バックアップをとる前に WAL アーカイビングステータスをチェックするとき、Postgres 接続のリークを防ぎます。                                                                                                                               |
| Bug fix      | そうでなければログをあふれさせる一時的なエラーに続いて、WAL アーカイブ/リストアを 100 ミリ秒間スリープさせます。                                                                                                                             |

## 1.7 EDB Postgres for Kubernetes 1.9.2 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                     |
|---------|-----------------------------------------------------------------------------------------------------------------|
| Feature | 2 つの新しいロガーで JSON ログを強化します。PostgreSQL の archive_command の wal-archive と、スタンバイの restore_command の wal-restore です。 |

次のページに続く

表 8 – 前のページからの続き

| Type    | Description                                                                                |
|---------|--------------------------------------------------------------------------------------------|
| Bug fix | スタンバイプロモーション中に WAL アーカイビングを有効にします (.history ファイルがアーカイブされないようにします)。                         |
| Bug fix | 古いオペランドのエラーを回避するために、Barman 2.13 以降を使用する場合にのみ、-cloud-provider オプションを Barman Cloud ツールに渡します。 |
| Bug fix | バックアップを開始する前に、プライマリのポッドが準備できるまで待ちます。                                                       |

## 1.8 EDB Postgres for Kubernetes 1.9.1 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                             |
|---------|-----------------------------------------------------------------------------------------|
| Feature | このリリースは、特定のイメージ名前を定義せずに新しいクラスターが作成されたときにデフォルトのメジャーバージョンにすることで、PostgreSQL 14 の発売を祝うことです。 |
| Bug fix | プライマリポッドのログに barman エンドポイント CA シークレットメッセージが表示されているときにエラーが発生し、バックアップが正常に機能しなかった問題を修正します。 |
| Bug fix | インスタスマネージャとの一時的な通信の問題に備えて、新しいバックアップの要求を適切にリトライします。                                      |

## 1.9 EDB Postgres for Kubernetes 1.9.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                              |
|---------|--------------------------------------------------------------------------------------------------------------------------|
| Feature | サポートされている Kubernetes ディストリビューションのリストに Kubernetes 1.22 を追加し、1.16 を削除します。                                                  |
| Feature | 必要に応じて、バックアップアーカイブから WAL ファイルをフェッチオーダーに、pg_rewind で--restore-target-wal オプションのサポートを導入します (PostgreSQL/EPAS 13 以降でのみ利用可能)。 |
| Feature | バージョン 1.9.0 は、リリースプロセスとその後のバージョン 1.9.1 のリリースの遅延のため、OpenShift では使用できません。                                                  |
| Feature | 各データベースの pg_catalog.pg_largeobject テーブルのページ数を推定する Prometheus エクスポートのデフォルトメトリックを公開します。                                    |
| Feature | ローカルのメモリ内キャッシュにより、WAL のアーカイビングとフェッチのパフォーマンスを向上させます。                                                                      |
| Bug fix | pg_isready を呼び出すときに postgres ユーザを明示的に設定します-OpenShift の制限付き SCC で必要です。                                                    |

次のページに続く

表 10 – 前のページからの続き

| Type    | Description                                              |
|---------|----------------------------------------------------------|
| Bug fix | ステータスの FirstRecoverabilityPoint を適切に更新します。               |
| Bug fix | バックアップが要求される場合、指定されたプライマリで archive_mode = always を設定します。 |

## 1.10 EDB Postgres for Kubernetes 1.8.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                                                                                                                                         |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature | 外部クラスタセクションで定義されたオブジェクトストアから直接フルまたはポイントインタイムリカバリを介して新しいクラスタをブートストラップし、バックアップ CR を定義するという以前の要件を排除します。                                                                                                                                |
| Feature | スケジュールされたバックアップに即時オプションを導入して、最初の Postgres インスタンスの実行直後にバックアップを要求し、Point-In-Time Recovery が構成されているときにクラスタの最初に巻き戻す機能を追加します。                                                                                                            |
| Feature | クラスタステータスに firstRecoverabilityPoint を追加して、バックアップオブジェクトストアのコンテンツに基づいてリカバリを要求するために、最も古い一貫した時点レポートします。                                                                                                                                 |
| Feature | 次の新しいメトリックを公開することにより、PostgreSQL インスタンスのデフォルトの Prometheus エクスポーターを強化します：ディスク上の WAL ファイルの数と計算された合計サイズ、ディスクディスクの WAL ファイルの数と計算された合計サイズ、アーカイブステータスの .ready および .done ファイルの数フォルダー、レプリカモードのフラグ、要求された最小/最大同期的レプリカの数、および予想されるものと実際に観測されたもの。 |
| Feature | Prometheus エクスポーターでカスタムメトリックを定義するときに runonserver オプションのサポートを追加して、メトリックのコレクションを PostgreSQL バージョンのレンジに制限します。                                                                                                                          |
| Feature | Barman Cloud 向けに Barman 2.13 で導入された機能を活用することにより、バックアップとリカバリのために Azure Blob Storage をネイティブにサポートします。                                                                                                                                  |
| Feature | liveness プロローブは pg_isready に依存してください。                                                                                                                                                                                               |
| Feature | リカバリターゲット時間のタイムスタンプ仕様の RFC3339 フォーマットをサポートします。                                                                                                                                                                                      |
| Feature | .spec.imagePullPolicy を導入して、クラスタ用に作成されたすべてのポッドとジョブのイメージコンテナのプルポリシーを制御します。                                                                                                                                                           |
| Feature | OpenShift 4.5 に代わる OpenShift 4.8 のサポートを追加します。                                                                                                                                                                                       |
| Feature | PostgreSQL 14 (ベータ) をサポートします。                                                                                                                                                                                                       |
| Feature | ストリーミング接続を必要とせず、外部クラスタセクションで定義されたオブジェクトストアからのクラスタ間レプリケーションでレプリカクラスタ機能を強化します (実験的)。                                                                                                                                                  |

次のページに続く

表 11 – 前のページからの続き

| Type         | Description                                                                       |
|--------------|-----------------------------------------------------------------------------------|
| Feature      | クラスターの仕様に logLevel オプションを導入して、エラー、情報、デバッグ、トレースのいずれかのレベルを指定します。                    |
| Security fix | .spec.enableSuperuserAccess を導入して、パスワード認証を通じて postgres ユーザとのネットワークアクセスを有効/無効にします。 |
| Security fix | spec.licenseKeySecret を使用して、秘密のライセンスキーの指定を有効にします。                                 |
| Bug fix      | クラスターが回復不能な状態になり、人間の介入が必要になったときに、ユーザーに適切に通知します。                                   |

## 1.11 EDB Postgres for Kubernetes 1.7.1 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------|
| Feature | 同期レプリケーションに関しては、構成よりも自己修復を優先し、クラスターが自己修復操作中に要件を満たせない場合に、minSyncReplicas および演算子設定を一時的にオーバーライドできるようにします。 |
| Feature | initdb ブートストラップメソッドのパートとして postInitSQL オプションを導入して、クラスターが作成された直後にスーパーユーザとして実行される SQL クエリのリストを指定します。     |
| Bug fix | プライマリの準備ができていないときに演算子がフェイルオーバーできるようにします (1.7.0 で導入されたバグ)。                                              |
| Bug fix | LOCAL 同期的コミットレベルを使用して管理クエリを実行します。                                                                      |
| Bug fix | PGAudit で複数行のログエントリを正しく解析します。                                                                          |

## 1.12 EDB Postgres for Kubernetes 1.7.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                      |
|---------|------------------------------------------------------------------------------------------------------------------|
| Feature | JSON 出力で直接利用できる pgaudit と呼ばれる新しいタイプのロガーを使用して、PGAudit にネイティブサポートを追加します。                                           |
| Feature | pg_stat_statements および auto_explain 拡張のネイティブサポート。                                                                |
| Feature | 1 つ以上のデータベースでユーザー定義のメトリッククエリを実行する Prometheus エクスポートの target_databases オプション (シェルのようなパターンマッチングによるデータベースの自動検出を含む)。 |
| Feature | primaryUpdateStrategy が監視対象に設定されているクラスターでマニュアルスイッチオーバーが必要かどうかを迅速にレポートするための manual_switchover_required メトリックの公開。  |

次のページに続く

表 13 – 前のページからの続き

| Type         | Description                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Feature      | pg_audit、auto_explain、pg_stat_statements の shared_preload_libraries を透過的にハンドルします。                                                 |
| Feature      | pg_stat_statements、pgaudit、または auto_explain オプションが postgresql パラメーターセクションに追加された場合の PostgreSQL の shared_preload_libraries の自動構成。   |
| Feature      | k8s.enterprisedb.io/ リロード label をサポートして、Prometheus エクスポートのカスタムモニタリング/アラートメトリックの使用や証明書の保存に使用されるものを含む、構成マップとシークレットの自動リロードを細かく制御します。 |
| Feature      | リロードコマンドを kubectl の cnp プラグインに追加して、インスタンスで調整ループをトリガーします。                                                                          |
| Feature      | additionalPodAffinity および additionalPodAntiAffinity を使用して、ポッドアフィニティおよび非アフィニティ構成の制御を改善します。                                         |
| Feature      | 少なくとも 1 つのプライマリインスタンスをいつでも実行することを要求することにより、プライマリインスタンスに個別の PodDisruptionBudget を導入します。                                            |
| Security fix | .spec.certificates.clientCASecret および spec.certificates.replicationTLSSecret オプションを追加して、カスタムクライアント認証を定義します。                       |
| Security fix | PostgreSQL サーバーの認証局と証明書。クライアント証明書の認証と PostgreSQL ノード間の安全な通信に使用されます。                                                               |
| Security fix | .spec を追加します。バックアップ.barmanObjectStore.endpointCA オプションは、Barman のバックアップオブジェクトストアのエンドポイントのカスタム証明機関バンドルを定義します。                       |
| Bug fix      | Prometheus エクスポートでヒストグラムを正しく解析します。                                                                                                |
| Bug fix      | 演算子がクラスター用に作成したサービスを調整します。                                                                                                        |

## 1.13 EDB Postgres for Kubernetes 1.6.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                          |
|---------|------------------------------------------------------------------------------------------------------|
| Feature | レプリカモード（実験的）：ソースクラスターのレプリカとしてクラスターを作成できるようにします。レプリカクラスターには、指定されたプライマリと任意の数のスタンバイがあります。               |
| Feature | EDB EDB Postgres Advanced Server イメージでの EDB Audit サポート。                                              |
| Feature | .spec.postgresql.promotionTimeout パラメータを追加して、インスタンスをプライマリに昇格するときに待機する最大秒数を指定します。デフォルトは 40000000 秒です。 |

次のページに続く

表 14 – 前のページからの続き

| Type         | Description                                                                                                                                                                                         |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature      | .spec.affinity.podAntiAffinityType パラメータを追加します。 preferred（デフォルト）に設定すると、preferredDuringSchedulingIgnoredDuringExecution が使用されるか、required に設定され、requiredDuringSchedulingIgnoredDuringExecution になります。 |
| Security fix | ライセンスキーがログに表示されないようにします。                                                                                                                                                                            |
| Change       | 演算子が新しいポッドを作成できないようにする PVC とポッドを削除するときの競合条件を修正しました。                                                                                                                                                 |
| Change       | 構成の変更時にマネージャが PostgreSQL のリスタートの必要性を検出できない競合条件を修正しました。                                                                                                                                              |
| Change       | 注釈のないクラスターでの kubectl-cnnp のパニックを修正しました。                                                                                                                                                             |
| Change       | デバッグするログメッセージのレベルを下げました。                                                                                                                                                                            |
| Change       | サーバー CA および TLS インジェクションの E2E テスト。                                                                                                                                                                  |

## 1.14 EDB Postgres for Kubernetes 1.5.1 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type   | Description                                                                |
|--------|----------------------------------------------------------------------------|
| Change | Red Hat OpenShift 上の Operator Deployments での自動更新を妨げる CRD バリデーションのバグを修正します。 |
| Change | シークレットを使用してオペレーターの構成を渡すことを許可します。                                           |

## 1.15 EDB Postgres for Kubernetes 1.5.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type         | Description                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| Feature      | pg_basebackup ブートストラップメソッドを導入して、Kubernetes 以外でも、同じメジャーバージョンの既存の PostgreSQL インスタンスのコピーとして新しい PostgreSQL クラスターを作成します。 |
| Feature      | Cluster リソースの Affinity セクションで Kubernetes の許容範囲のサポートを追加し、ユーザーが必要な汚染で Kubernetes ノードに PostgreSQL インスタンスを配布できるようにします。  |
| Feature      | より決定的で反復可能なデプロイメントのために、`<image>:<tag>@sha256:<digestValue>` フォーマットを介して、イメージ名前へのダイジェストの指定を有効にします。                    |
| Security fix | サーバー証明書とそれに署名した関連する認証局のシークレットを定義することにより、TLS 証明書をカスタマイズして PostgreSQL サーバーを認証します。                                     |

次のページに続く

表 16 – 前のページからの続き

| Type              | Description                                                                               |
|-------------------|-------------------------------------------------------------------------------------------|
| Security fix      | require から verify-ca への内部および自動管理ストリーミングレプリカの WAL レシーバプロセスの sslmode を上げます。                 |
| Change            | kubectrl のプロモートプラグインの Promo サブコマンドを拡張して、ポッドの名前全体ではなくノード番号のみを受け入れるようにします。                  |
| Change            | (サービス名が継承される) クラスター名に DNS-1035 バリデーションスキームを採用します。                                         |
| Change            | スタンバイインスタンスのクローンを作成するとき、または pg_basebackup メソッドを使用してブートストラップするときに、ストリーミングレプリケーション接続を強制します。 |
| Change            | 物理的ベースバックアップに関して、Backup リソースを beginWal、endWal、beginLSN、endLSN、startedAt、stoppedAt と統合します。 |
| Documentation fix | 演算子およびオペランドコンテナーによって公開されるポートのリストを提供します。                                                   |
| Documentation fix | データベースワークロード用のストレージと PostgreSQL のベンチマークの cnp-bench ヘルムチャートとガイドラインを紹介します。                  |
| E2E test fix      | Kubernetes 1.21 をテストします。                                                                  |
| E2E test fix      | 演算子の高可用性のテストを追加します。                                                                       |
| E2E test fix      | ノードドレインのテストを追加します。                                                                        |
| Bug fix           | リカバリ操作中の pg_ctl スタートのタイムアウトが短すぎます。                                                        |
| Bug fix           | オペレーターは、PVC の直接イベントを見守っていません。                                                             |
| Bug fix           | barmanObjectStore バックアップの immediateCheckpoint および jobs パラメータのハンドリングを修正しました。               |
| Bug fix           | バックアップから回復するときに空のログ。                                                                      |

## 1.16 EDB Postgres for Kubernetes 1.4.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type         | Description                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Feature      | JSON フォーマットの PostgreSQL エラーメッセージの標準出力ロギング。                                                                                                                                                                                                         |
| Feature      | Prometheus エクスポーター用の PostgreSQL メトリックの基本セットを提供します。                                                                                                                                                                                                 |
| Feature      | kubectrl の cnp プラグインにリスタートコマンドを追加して、特定の PostgreSQL クラスターのポッドをロールアウト方式でリスタートします。                                                                                                                                                                    |
| Security fix | ポッドの readOnlyRootFilesystem セキュリティコンテキストを設定します。                                                                                                                                                                                                    |
| Change       | 重要：YAML マニフェストを使用して Cloud Native PostgreSQL 演算子を以前にデプロイした場合、新しいバージョンをインストールする前に既存の演算子デプロイメントを削除する必要があります。これは、演算子によって直接管理されるラベルと label セレクタの変更による他の Kubernetes API との競合を回避するために必要です。1.4.0 へのアップグレード処理の詳細については、Cloud Native PostgreSQL の文書を参照してください。 |

次のページに続く

表 17 – 前のページからの続き

| Type                | Description                                                                                                              |
|---------------------|--------------------------------------------------------------------------------------------------------------------------|
| Change              | 演算子によって自動的に定義されるラベルを修正し、control-plane : controller-manager から app.kubernetes.io/ 名前 : cloud-native-postgresql に名前を変更します。 |
| Change              | Prometheus エクスポートの TCP ポートにメトリック名前を割り当てます。                                                                               |
| Change              | cnp_metrics_exporter を application_name として、PostgreSQL のメトリックエクスポート接続に設定します。                                             |
| Change              | 利用可能な場合、postgres データベースの代わりに、Prometheus エクスポートのクエリをモニタリングするためにアプリケーションデータベースを使用します。                                      |
| Documentation fixes | モニタリングクエリのカスタマイズ。                                                                                                        |
| Documentation fixes | オペレーターのアップグレード手順。                                                                                                        |
| Bug fix             | pg_basebackup を呼び出すときに -R を使用しないでください。                                                                                   |
| Bug fix             | ステータスを取得するときに、エラーログからスタックトレースを削除します。                                                                                     |

## 1.17 EDB Postgres for Kubernetes 1.3.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type         | Description                                                                                         |
|--------------|-----------------------------------------------------------------------------------------------------|
| Feature      | ラベルと注釈の継承。                                                                                          |
| Feature      | すべてのコンテナにリソース制限を設定します。                                                                              |
| Security fix | Red Hat OpenShift での制限付きセキュリティコンテキスト制約のサポートにより、ポッドの実行を UID および SELinux コンテキストが割り当てられたネームスペースに制限します。 |
| Security fix | 非ルート、非特権、および権限エスカレーションなしで実行するために演算子によって明示的に定義されたポッドセキュリティコンテキスト。                                    |
| Change       | ポート 9187 でリッスンしている Prometheus エクスポートエンドポイント (ポート 8000 は API サーバーとのインスタンス調整用に予約しています)。               |

## 1.18 EDB Postgres for Kubernetes 1.2.1 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type         | Description                                                                            |
|--------------|----------------------------------------------------------------------------------------|
| Feature      | ScheduledBackup はバックアップの所有者ではなくなりました。つまり、ScheduledBackup オブジェクトが削除されても、バックアップは削除されません。 |
| Security fix | ubi8-minimal イメージを更新して、RHSA-2021 : 1024 を解決します (セキュリティアドバイザリ : 重要)。                    |

## 1.19 EDB Postgres for Kubernetes 1.2.0 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                                                       |
|---------|-------------------------------------------------------------------------------------------------------------------|
| Feature | Prometheus の postgres_exporter と互換性のある構文を使用して、ConfigMap および Secret オブジェクトとしてカスタムモニタリングクエリの実験的サポートを導入します。          |
| Feature | オペレーターライフサイクルマネージャー（OLM）の展開をサポートし、その後 OperatorHub.io に存在します。                                                      |
| Feature | 全社的に使用するためのライセンスキーサポートを拡張します（以前の制限は単一のクラスター名前空間のみに制限されていました）。                                                     |
| Feature | 米国国防総省（DoD）の防衛情報システム局（DISA）およびインターネットセキュリティセンター（CIS）のガイドラインを適用し、Dockle を使用してパイプラインで直接検証することにより、コンテナのセキュリティを強化します。 |

## 1.20 EDB Postgres for Kubernetes 1.1 release notes

Kubernetes 用 EDB Postgres のこのリリースには、次のものが含まれます。

| Type    | Description                                                                             |
|---------|-----------------------------------------------------------------------------------------|
| Feature | JSON および YAML 出力を含むクラスターのステータスをきれいに出力するには、kubectrl cnp status を追加します。                   |
| Feature | kubectrl cnp 証明書を追加して、クライアントアプリケーションの TLS 認証を有効にします。                                    |
| Feature | -ro サービスを追加して、利用可能なホットスタンバイレプリカのみに接続をルーティングし、クラスターのプライマリインスタンスから読み取り専用クエリのオフロードを有効にします。 |
| Feature | クラスターを maxSyncReplicas より低い値にスケールダウンするロールバック。                                           |
| Feature | 元のプライマリを降格する前にチェックポイントを要求します。                                                           |
| Feature | SIGTERM で PostgreSQL プロセスに SIGINT シグナル（高速シャットダウン）を送信します。                                |

## 1.21 EDB Postgres for Kubernetes 1.0 release notes

最初のメジャーリリースでは、API グループ postgresql.k8s.enterprisedb.io/v1 に Cluster、Backup、および ScheduledBackup が実装されています。これらのリソースを使用して、以下の主な機能を備えた Kubernetes 内で PostgreSQL クラスターを作成および管理します。

| Type    | Description                                        |
|---------|----------------------------------------------------|
| Feature | 外部ツールを必要とせずに、高可用性のために Kubernetes API サーバーと直接統合します。 |
| Feature | 最も整合したレプリカを昇格させることによるプライマリインスタンスのフェールオーバー。         |
| Feature | レプリカの自動作成。                                         |
| Feature | 選択したレプリカを昇格させることによる、プライマリインスタンスの計画的な切り替え。          |
| Feature | スケールアップ/スケールダウン機能。                                 |
| Feature | 任意の数のインスタンスの定義（最小 1-1 つのプライマリーサーバ）。                |
| Feature | アプリケーションをクラスターの唯一のプライマリーサーバに接続するための読み書きサービスの定義。    |
| Feature | ワークロードを読み取るためにアプリケーションを任意のインスタンスに接続する読み取りサービスの定義。  |
| Feature | PVC テンプレートを使用したローカル永続ボリュームのサポート。                   |
| Feature | ポッドでの永続ボリュームストレージの再利用。                             |
| Feature | PostgreSQL のマイナーバージョンと演算子のアップグレードのローリング更新。         |
| Feature | TLS 接続とクライアント証明書認証。                                |
| Feature | S3 互換オブジェクトストアへの継続的なバックアップ。                        |
| Feature | S3 互換オブジェクトストアバックアップからの完全リカバリとポイントインタイムリカバリ。       |
| Feature | 同期的レプリカのサポート。                                      |
| Feature | nodeSelector プロパティを介したノードアフィニティのサポート。              |
| Feature | PostgreSQL エラーメッセージの標準出力ロギング。                      |

## 第2章 ‘Differences with CloudNativePG v1’

Kubernetes 演算子用の EDB Postgres は、CloudNativePG に基づくフォークです。

どちらも Cluster という顧客リソースを定義していますが、2つの異なる API グループがあります。その結果、同じ Kubernetes クラスターに共存できます。具体的には：

- CloudNativePG は postgresql.cnpg.io/v1 API バージョンを使用します
- Kubernetes の EDB Postgres は postgresql.k8s.enterprisedb.io/v1 API バージョンを使用します

---

**重要：** Cloud Native PostgreSQL を展開している既存の EDB のお客様は、Kubernetes 用の EDB Postgres の序文によってバージョン 1 の API グループが変更されていないことに気付いているかもしれません。

---

一般的なルールとして、オープンソースの CloudNativePG で利用可能なすべての機能は、Kubernetes の EDB Postgres でも利用可能です-正しい API バージョンが使用されている場合。

利用可能な機能の詳細については、CloudNativePG の文書を参照してください。



## 第3章 ‘Differences with CloudNativePG’

Kubernetes 演算子用の EDB Postgres は、CloudNativePG に基づくフォークです。

両方とも `Cluster` という顧客リソースを定義していますが、2つの異なる API グループがあります。その結果、同じ Kubernetescluster に共存できます。具体的には：

- CloudNativePG は `postgresql.cnpg.io/v1` API バージョンを使用します
- Kubernetes の EDB Postgres は `postgresql.k8s.enterprisedb.io/v1` API バージョンを使用します

---

**重要：** Cloud Native PostgreSQL を展開している既存の EDB のお客様は、Kubernetes 用の EDB Postgres の序文によってバージョン 1 の API グループが変更されていないことに気付いているかもしれません。

---

一般的なルールとして、オープンソースの CloudNativePG で利用可能なすべての機能は、Kubernetes 用の EDB Postgres でも利用可能です-正しい API バージョンが使用されている場合。

利用可能な機能の詳細については、[CloudNativePG documentation](#) を参照してください。



## 第4章 ‘Installation and upgrades’

以下は、アップストリーム CloudNativePG プロジェクトの 'Installation and upgrades' トピックを補足するものです。

### 4.1 Kubernetes でのインストール

#### 4.1.1 演算子マニフェストを直接使用する

演算子は、`kubectl` を介して適用される YAML マニフェストを介して、Kubernetes の他のリソースと同様にインストールできます。

`latest operator manifest` は次のようにインストールできます。

```
kubectl apply -f \
  https://get.enterprisedb.io/cnp/postgresql-operator-1.15.0.yaml
```

`kubectl` コマンドを実行すると、Kubernetes の EDB Postgres が Kubernetes クラスターにインストールされます。

次の方法で確認できます。

```
kubectl get deploy -n postgresql-operator-system postgresql-operator-controller-
→manager
```

<!-- この作業は必要です### Operator Lifecycle Manager (OLM) を使用する

OperatorHub は、オペレーター向けのパッケージ管理システムである `Operator Lifecycle Manager` を介して利用可能な、オペレーターがコミュニティから入手したインデックスです。

そのページにリストされているインストール手順に従って、`OperatorHub.io website` の `EDB Postgres for Kubernetes page` で利用可能なメタデータを使用して、Kubernetes 用の EDB Postgres をインストールできます。

-

## 4.2 展開に関する詳細

CloudNativePG とは異なり、演算子はデフォルトで `postgresql-operator-controller-manager` と呼ばれる Kubernetes Deployment として `postgresql-operator-system` 名前空間にインストールされます。以下を実行することにより、詳細情報を取得できます。

```
kubectl describe deploy \
  -n postgresql-operator-system \
  postgresql-operator-controller-manager
```

## 第5章 ‘Quickstart’

CloudNativePG 文書の 'Installation and upgrades' トピックを参照し、Kubernetes 用 EDB Postgres の使用に関する詳細についてはこのトピックを使用してください。

< !-fileProcessor / processors / add-quickstart-content.js によって挿入された以下のセクション-ここで行われた変更は失われます->

< !-挿入セクションの終了->

Red Hat OpenShift Container Platform ユーザーは、OpenShift の [Red Hat CodeReady Containers \(CRC\)](#) で Kubernetes の EDB Postgres の認定演算子をテストできます。

**警告:** このセクションに含まれる手順は、デモンストレーション、テスト、および実践のみを目的としたものであり、実稼動環境では使用しないでください。

他の Kubernetes アプリケーションと同様に、EDM EDB Postgres for Kubernetes は、YAML で記述された通常のマニフェストを使用して展開されます。

これらの指示に従うことにより、ローカル Kubernetes / Openshift インストールで PostgreSQLcluster をスタートし、実験することができるはずです。

---

**重要:** Kubernetes クラスターに接続するオーダーにマシンに *kubectl* がインストールされていることを確認します。OpenShift に CRC を使用している場合は *oc* をインストールします。 *how to install 'kubectl'* <<https://kubernetes.io/docs/tasks/tools/install-kubectl/>>`\_\_ の Kubernetes の文書または *how to install 'oc'* <[https://docs.openshift.com/container-platform/4.6/cli\\_reference/openshift\\_cli/getting-started-cli.html](https://docs.openshift.com/container-platform/4.6/cli_reference/openshift_cli/getting-started-cli.html)>`\_\_ の Openshift の文書に従ってください。

---

---

**注釈:** Openshift を実行している場合、この文書で *kubectl* について言及するたびに *oc* を使用します。 *kubectl* コマンドは、 *oc* コマンドと互換性があります。

---

## 5.1 パート 1-ローカル Kubernetes / Openshift プレイグラウンドのセットアップ

このセクションでは、CRC のインストールについて説明します。Kind または Minikube については、Cloud-NativePG のアップストリーム文書を参照してください。

### 5.1.1 CodeReady コンテナ (CRC)

1. Download Red Hat CRC を実行し、PATH

内のディレクトリ内にバイナリを移動します。

1. 次のコマンドを実行します。

```
crc setup
crc start
```

`crc start` の出力は、処理方法を説明しています。1. `crc oc-env` コマンドの出力を実行します。印刷された `oc login` コマンドを使用して、`kubeadmin` としてログインします。 `crc console` を実行している Web コンソールを開くこともできます。ユーザーとパスワードは `oc login` コマンドの場合と同じです。

1. CRC には `StorageClass` が付属していないため、設定する必要があります。 [Dynamic volume provisioning wiki page](#) に従って、`rancher/local-path-provisioner` をインストールします。

## 5.2 パート 2-Kubernetes 用の EDB Postgres のインストール

Kubernetes または OpenShift のインストールがラップトップで実行されたので、Kubernetes の EDB Postgres インストールを続行できます。

CloudNativePG の文書の 'Installation and upgrades' トピックを参照して、PostgreSQL クラスターの展開に進みます。

## 5.3 パート 3- PostgreSQL クラスターをデプロイする

Kubernetes の他の展開と同様に、PostgreSQL クラスターを展開するには、目的の `Cluster` を定義する構成ファイルを適用する必要があります。

`cluster-example.yaml` サンプルファイルは、デフォルトのストレージクラスを使用してシンプル

`Cluster` を定義し、ディスクスペースを割り当てます。

```
# Example of PostgreSQL cluster
apiVersion: postgresql.k8s.enterprisedb.io/v1
kind: Cluster
metadata:
  name: cluster-example
spec:
  instances: 3

  # Require 1Gi of space
  storage:
    size: 1Gi
```



## 第6章 Installation, Configuration and Deployment Demo

Kubernetes Operator の EDB Postgres を起動して実行するために必要なものを確認したいですか？このセクションでは、次のことを示します。

1. Kubernetes Operator2 の EDB Postgres をインストールします。3 ノードの PostgreSQL クラスターの展開 3。 kubectl-cnp plugin4 のインストールと使用。 フェイルオーバーのテストによるクラスターの復元力の検証

作業に約 5～10 分かかります。

```
k3d cluster create
__OUTPUT__
INFO[0000] Prep: Network
INFO[0000] Created network k3d-k3s-default
INFO[0000] Created image volume k3d-k3s-default-images
INFO[0000] Starting new tools node...
INFO[0001] Pulling image ghcr.io/k3d-io/k3d-tools:5.4.1
INFO[0001] Creating node k3d-k3s-default-server-0
INFO[0002] Pulling image docker.io/rancher/k3s:v1.22.7-k3s1
INFO[0003] Starting Node k3d-k3s-default-tools
INFO[0007] Creating LoadBalancer k3d-k3s-default-serverlb
INFO[0008] Pulling image ghcr.io/k3d-io/k3d-proxy:5.4.1
INFO[0011] Using the k3d-tools node to gather environment information
INFO[0011] HostIP: using network gateway 172.17.0.1 address
INFO[0011] Starting cluster k3s-default
INFO[0011] Starting servers...
INFO[0011] Starting Node k3d-k3s-default-server-0
INFO[0016] All agents already running.
INFO[0016] Starting helpers...
INFO[0017] Starting Node k3d-k3s-default-serverlb
INFO[0024] Injecting records for hostAliases (incl. host.k3d.internal) and for 2
↪network members into CoreDNS configmap...
INFO[0026] Cluster k3s-default created successfully!
INFO[0026] You can now use it like this:
kubectl cluster-info
```

これにより Kubernetes クラスターが作成され、使用する準備が整います。次のコマンドで動作することを確認します。

```
kubectl get nodes
__OUTPUT__
NAME                                STATUS    ROLES                                AGE    VERSION
k3d-k3s-default-server-0           Ready     control-plane,master                35s    v1.22.7+k3s1
```

k3d-k3s-default-server-0 という 1 つのノードが表示されます。ステータスがまだ「準備完了」でない場合は、数秒待ってから上記のコマンドを再度実行します。

## 6.1 Kubernetes 用 EDB Postgres をインストールする

Kubernetes クラスターが実行されたので、'Installation and upgrades' セクションの説明に従って Kubernetes の EDB Postgres インストールを続行できます。

```
kubectl apply -f https://get.enterprisedb.io/cnp/postgresql-operator-1.15.0.yaml
__OUTPUT__
namespace/postgresql-operator-system created
customresourcedefinition.apiextensions.k8s.io/backups.postgresql.k8s.enterprisedb.io
→created
customresourcedefinition.apiextensions.k8s.io/clusters.postgresql.k8s.enterprisedb.io
→created
customresourcedefinition.apiextensions.k8s.io/poolers.postgresql.k8s.enterprisedb.io
→created
customresourcedefinition.apiextensions.k8s.io/scheduledbackups.postgresql.k8s.
→enterprisedb.io created
serviceaccount/postgresql-operator-manager created
clusterrole.rbac.authorization.k8s.io/postgresql-operator-manager created
clusterrolebinding.rbac.authorization.k8s.io/postgresql-operator-manager-rolebinding
→created
configmap/postgresql-operator-default-monitoring created
service/postgresql-operator-webhook-service created
deployment.apps/postgresql-operator-controller-manager created
mutatingwebhookconfiguration.admissionregistration.k8s.io/postgresql-operator-
→mutating-webhook-configuration created
validatingwebhookconfiguration.admissionregistration.k8s.io/postgresql-operator-
→validating-webhook-configuration created
```

そして、正常にインストールされたことを確認します。

```
kubectl get deploy -n postgresql-operator-system postgresql-operator-controller-
→manager
__OUTPUT__
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
postgresql-operator-controller-manager 1/1      1              1            52s
```

## 6.2 PostgreSQL クラスターをデプロイする

Kubernetes の他の展開と同様に、PostgreSQL クラスターを展開するには、目的の Cluster を定義する構成ファイルを適用する必要があります。

cluster-example.yaml サンプルファイルは、デフォルトのストレージクラスを使用してシンプル Cluster を定義し、ディスクスペースを割り当てます。

```
cat <<EOF > cluster-example.yaml
# Example of PostgreSQL cluster
apiVersion: postgresql.k8s.enterprisedb.io/v1
kind: Cluster
metadata:
  name: cluster-example
spec:
  instances: 3

  # Example of rolling update strategy:
  # - unsupervised: automated update of the primary once all
  #                   replicas have been upgraded (default)
  # - supervised: requires manual supervision to perform
  #                   the switchover of the primary
  primaryUpdateStrategy: unsupervised

  # Require 1Gi of space
  storage:
    size: 1Gi
EOF
```

3 ノードの PostgreSQL クラスターを作成するには、オーダーのコマンドを実行する必要があります。

```
kubectl apply -f cluster-example.yaml
__OUTPUT__
cluster.postgresql.k8s.enterprisedb.io/cluster-example created
```

get pods コマンドを使用して、ポッドが作成されていることを確認できます。初期化にはビット時間がかかるため、クラスター構成を適用した直後に実行すると、Init: または PodInitializing のステータスが表示されます。

```
kubectl get pods
__OUTPUT__
```

| NAME                              | READY | STATUS  | RESTARTS | AGE |
|-----------------------------------|-------|---------|----------|-----|
| cluster-example-1-initdb--1-2cqfw | 0/1   | Pending | 0        | 3s  |

…少し待ってから、もう一度確認してください。

```
kubectl get pods
__OUTPUT__
NAME                READY   STATUS    RESTARTS   AGE
cluster-example-1   1/1     Running   0           56s
cluster-example-2   1/1     Running   0           35s
cluster-example-3   1/1     Running   0           19s
```

これで、クラスターのステータスを確認できます。

```
kubectl get cluster cluster-example -o yaml
__OUTPUT__
apiVersion: postgresql.k8s.enterprisedb.io/v1
kind: Cluster
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"postgresql.k8s.enterprisedb.io/v1","kind":"Cluster","metadata":{"
      ↪ "annotations":{"},"name":"cluster-example","namespace":"default"},"spec":{"instances
      ↪ ":3,"primaryUpdateStrategy":"unsupervised","storage":{"size":"1Gi"}}}
  creationTimestamp: "2022-04-21T18:33:12Z"
  generation: 1
  name: cluster-example
  namespace: default
  resourceVersion: "1868"
  uid: bf85fdbf-4404-46a0-a16a-63676a377d90
spec:
  affinity:
    podAntiAffinityType: preferred
    topologyKey: ""
  bootstrap:
    initdb:
      database: app
      encoding: UTF8
      localeCTYPE: C
      localeCollate: C
      owner: app
    enableSuperuserAccess: true
  imageName: quay.io/enterprisedb/postgresql:14.2
  instances: 3
  logLevel: info
  maxSyncReplicas: 0
  minSyncReplicas: 0
  monitoring:
    customQueriesConfigMap:
      - key: queries
        name: postgresql-operator-default-monitoring
```

(次のページに続く)

(前のページからの続き)

```

    disableDefaultQueries: false
    enablePodMonitor: false
  postgresGID: 26
  postgresUID: 26
  postgresql:
    parameters:
      archive_mode: "on"
      dynamic_shared_memory_type: posix
      log_destination: csvlog
      log_directory: /controller/log
      log_filename: postgres
      log_rotation_age: "0"
      log_rotation_size: "0"
      log_truncate_on_rotation: "false"
      logging_collector: "on"
      max_parallel_workers: "32"
      max_replication_slots: "32"
      max_worker_processes: "32"
      shared_memory_type: mmap
      shared_preload_libraries: ""
      wal_keep_size: 512MB
      wal_receiver_timeout: 5s
      wal_sender_timeout: 5s
    primaryUpdateMethod: switchover
    primaryUpdateStrategy: unsupervised
  resources: {}
  startDelay: 30
  stopDelay: 30
  storage:
    resizeInUseVolumes: true
    size: 1Gi
  switchoverDelay: 40000000
status:
  certificates:
    clientCASecret: cluster-example-ca
  expirations:
    cluster-example-ca: 2022-07-20 18:28:12 +0000 UTC
    cluster-example-replication: 2022-07-20 18:28:12 +0000 UTC
    cluster-example-server: 2022-07-20 18:28:12 +0000 UTC
  replicationTLSSecret: cluster-example-replication
  serverAltDNSNames:
    - cluster-example-rw
    - cluster-example-rw.default
    - cluster-example-rw.default.svc
    - cluster-example-r

```

(次のページに続く)

```

- cluster-example-r.default
- cluster-example-r.default.svc
- cluster-example-ro
- cluster-example-ro.default
- cluster-example-ro.default.svc
serverCASecret: cluster-example-ca
serverTLSSecret: cluster-example-server
cloudNativePostgresqlCommitHash: 0531b739
cloudNativePostgresqlOperatorHash: 1dd0fcb2f3fa9e7293fa5696f0cd7dd560923cc9f7d714b75c10d282072fcf2d
configMapResourceVersion:
  metrics:
    postgresql-operator-default-monitoring: "848"
currentPrimary: cluster-example-1
currentPrimaryTimestamp: "2022-04-21T18:33:49Z"
healthyPVC:
- cluster-example-1
- cluster-example-2
- cluster-example-3
instances: 3
instancesStatus:
  healthy:
  - cluster-example-1
  - cluster-example-2
  - cluster-example-3
latestGeneratedNode: 3
licenseStatus:
  isImplicit: true
  isTrial: true
  licenseExpiration: "2022-05-21T18:33:12Z"
  licenseStatus: Implicit trial license
  repositoryAccess: false
  valid: true
phase: Cluster in healthy state
poolerIntegrations:
  pgBouncerIntegration: {}
pvcCount: 3
readService: cluster-example-r
readyInstances: 3
secretsResourceVersion:
  applicationSecretVersion: "813"
  clientCaSecretVersion: "809"
  replicationSecretVersion: "811"
  serverCaSecretVersion: "809"
  serverSecretVersion: "810"

```

(前のページからの続き)

```

superuserSecretVersion: "812"
targetPrimary: cluster-example-1
targetPrimaryTimestamp: "2022-04-21T18:33:13Z"
writeService: cluster-example-rw

```

**注釈:** デフォルトでは、演算子は、リリースされたときに、PostgreSQL の最新メジャーバージョンの利用可能な最新マイナーバージョンをインストールし演算子。[the `imageName` key in the `spec` section of the `Cluster` definition](`api_reference/#clusterspec`)を設定することでこれをオーバーライドできます。

**重要:** 不変のインフラストラクチャパラダイムでは、コンテナイメージの特定のバージョンを常にポイントする必要があります。実稼動環境では、`latest` や `13` などのタグを使用しないでください。クラスター内の更新ポリシーとバージョンの一貫性に関して予測できないシナリオが発生する可能性があります。

## 6.3 kubectl-cnp プラグインをインストールする

Kubernetes の EDB Postgres は、Kubernetes のクラスターを管理するための a plugin for kubectl と、それをインストールするスクリプトを提供します。

```

curl -sSfL \
  https://github.com/EnterpriseDB/kubectl-cnp/raw/main/install.sh | \
  sudo sh -s -- -b /usr/local/bin
__OUTPUT__
EnterpriseDB/kubectl-cnp info checking GitHub for latest tag
EnterpriseDB/kubectl-cnp info found version: 1.15.0 for v1.15.0/linux/x86_64
EnterpriseDB/kubectl-cnp info installed /usr/local/bin/kubectl-cnp

```

`cnp` コマンドが `kubectl` で利用可能になりました。

```

kubectl cnp status cluster-example
__OUTPUT__
Cluster Summary
Name:          cluster-example
Namespace:     default
System ID:     7089127092443762706
PostgreSQL Image: quay.io/enterprisedb/postgresql:14.2
Primary instance: cluster-example-1
Status:        Cluster in healthy state
Instances:     3
Ready instances: 3
Current Write LSN: 0/4000060 (Timeline: 1 - WAL File: 00000001000000000000000004)

```

(次のページに続く)

(前のページからの続き)

## Certificates Status

| Certificate Name            | Expiration Date               | Days Left Until Expiration |
|-----------------------------|-------------------------------|----------------------------|
| -----                       | -----                         | -----                      |
| →-                          |                               |                            |
| cluster-example-replication | 2022-07-20 18:28:12 +0000 UTC | 89.99                      |
| cluster-example-server      | 2022-07-20 18:28:12 +0000 UTC | 89.99                      |
| cluster-example-ca          | 2022-07-20 18:28:12 +0000 UTC | 89.99                      |

## Continuous Backup status

Not configured

## Streaming Replication status

| Name              | Sent LSN  | Write LSN  | Flush LSN     | Replay LSN | Write Lag | Flush Lag |
|-------------------|-----------|------------|---------------|------------|-----------|-----------|
| →Replay Lag       | State     | Sync State | Sync Priority |            |           |           |
| ---               | -----     | -----      | -----         | -----      | -----     | -----     |
| cluster-example-2 | 0/4000060 | 0/4000060  | 0/4000060     | 0/4000060  | 00:00:00  | 00:00:00  |
| →00:00:00         | streaming | async      | 0             |            |           |           |
| cluster-example-3 | 0/4000060 | 0/4000060  | 0/4000060     | 0/4000060  | 00:00:00  | 00:00:00  |
| →00:00:00         | streaming | async      | 0             |            |           |           |

## Instances status

| Name              | Database Size | Current LSN | Replication role | Status | QoS        |
|-------------------|---------------|-------------|------------------|--------|------------|
| →Manager Version  |               |             |                  |        |            |
| ---               | -----         | -----       | -----            | -----  | ---        |
| →-----            |               |             |                  |        |            |
| cluster-example-1 | 33 MB         | 0/4000060   | Primary          | OK     | BestEffort |
| →1.15.0           |               |             |                  |        |            |
| cluster-example-2 | 33 MB         | 0/4000060   | Standby (async)  | OK     | BestEffort |
| →1.15.0           |               |             |                  |        |            |
| cluster-example-3 | 33 MB         | 0/4000060   | Standby (async)  | OK     | BestEffort |
| →1.15.0           |               |             |                  |        |            |

## 6.4 フェイルオーバーのテスト

ステータスチェックが示すように、2つのレプリカを実行しています PostgreSQL のプライマリインスタンスに何かが発生した場合、クラスターはそのうちの1つにフェイルオーバーします。プライマリポッドを殺すことでこれを実証しましょう：

```
kubectl delete pod --wait=false cluster-example-1
__OUTPUT__
pod "cluster-example-1" deleted
```

これは、サーバーのハードシャットダウン-何かがうまくいかないシナリオをシミュレートします。

```
ステータスを確認したら... ``shell kubectl cnp status cluster-example OUTPUT Cluster Summary
Name: cluster-example Namespace: default System ID: 7089127092443762706 PostgreSQL Image:
quay.io/enterprisedb/postgresql:14.2 Primary instance: cluster-example-2 Status: Failing over Failing over from
cluster-example-1 to cluster-example-2 Instances: 3 Ready instances: 2 Current Write LSN: 0/5000F58 (Timeline:
2 - WAL File: 00000002000000000000000005)
```

```
Certificates Status Certificate Name Expiration Date Days Left Until Expiration - - - - -
- - - - - - cluster-example-replication 2022-07-20 18:28:12 +0000 UTC 89.99 cluster-example-server
2022-07-20 18:28:12 +0000 UTC 89.99 cluster-example-ca 2022-07-20 18:28:12 +0000 UTC 89.99
```

Continuous Backup status Not configured

Streaming Replication status Not available yet

```
Instances status Name Database Size Current LSN Replication role Status QoS Manager Version - - - - -
- - - - - cluster-example-3 33 MB 0/50000A0 Standby (file based) OK Best-
Effort 1.15.0 cluster-example-2 33 MB 0/5000F58 Primary OK BestEffort 1.15.0 cluster-example-1 - - - pod not
available BestEffort -
```

... フェイルオーバープロセスが開始され、2 番目のポッドがプライマリーに昇格しました。失敗したポッドが再起動すると、新しいプライマリーのレプリカになります。

```
``shell
kubectl cnp status cluster-example
__OUTPUT__
Cluster Summary
Name: cluster-example
Namespace: default
System ID: 7089127092443762706
PostgreSQL Image: quay.io/enterprisedb/postgresql:14.2
Primary instance: cluster-example-2
Status: Cluster in healthy state
Instances: 3
Ready instances: 3
Current Write LSN: 0/5004CD8 (Timeline: 2 - WAL File: 00000002000000000000000005)

Certificates Status
Certificate Name Expiration Date Days Left Until Expiration
- - - - -
- - - - -
cluster-example-ca 2022-07-20 18:28:12 +0000 UTC 89.99
cluster-example-replication 2022-07-20 18:28:12 +0000 UTC 89.99
cluster-example-server 2022-07-20 18:28:12 +0000 UTC 89.99

Continuous Backup status
Not configured
```

(次のページに続く)

(前のページからの続き)

| Streaming Replication status |               |             |                  |            |            |           |
|------------------------------|---------------|-------------|------------------|------------|------------|-----------|
| Name                         | Sent LSN      | Write LSN   | Flush LSN        | Replay LSN | Write Lag  | Flush Lag |
| →Replay Lag                  | State         | Sync State  | Sync Priority    |            |            |           |
| -----                        | -----         | -----       | -----            | -----      | -----      | -----     |
| cluster-example-1            | 0/5004CD8     | 0/5004CD8   | 0/5004CD8        | 0/5004CD8  | 00:00:00   | 00:00:00  |
| →00:00:00                    | streaming     | async       | 0                |            |            |           |
| cluster-example-3            | 0/5004CD8     | 0/5004CD8   | 0/5004CD8        | 0/5004CD8  | 00:00:00   | 00:00:00  |
| →00:00:00                    | streaming     | async       | 0                |            |            |           |
| Instances status             |               |             |                  |            |            |           |
| Name                         | Database Size | Current LSN | Replication role | Status     | QoS        |           |
| →Manager Version             |               |             |                  |            |            |           |
| -----                        | -----         | -----       | -----            | -----      | ---        | ---       |
| cluster-example-3            | 33 MB         | 0/5004CD8   | Standby (async)  | OK         | BestEffort |           |
| →1.15.0                      |               |             |                  |            |            |           |
| cluster-example-2            | 33 MB         | 0/5004CD8   | Primary          | OK         | BestEffort |           |
| →1.15.0                      |               |             |                  |            |            |           |
| cluster-example-1            | 33 MB         | 0/5004CD8   | Standby (async)  | OK         | BestEffort |           |
| →1.15.0                      |               |             |                  |            |            |           |

### 6.4.1 さらに読む

PostgreSQL クラスターを起動して実行するのに必要なことはこれだけです。もちろん、もっと多くの可能性があります-もちろん、稼働環境にデプロイする前に、もっと多くのことが賢明です！

- CloudNativePG オペレーターが提供する設計目標と可能性： [サポートされているアーキテクチャ](#) および [Use cases](#) セクションを確認してください。
- 安全で信頼できるシステムの構成： [Security](#)、[Failure Modes](#)、および [Backup and Recovery](#) セクションを読んでください。

## 第7章 ‘Operator configuration’

EDB Postgresfor Kubernetes の演算子は、標準の展開マニフェストからインストールされ、構成パラダイムよりも慣習に従います。ほとんどの場合これは問題ありませんが、次のようなデフォルトの動作を変更したいシナリオがいくつかあります。

- 管理されるすべての展開で共有される会社のライセンスキーを設定する演算子による
- 作成されたすべてのリソースに継承されるアノテーションとラベルの定義演算子によって、クラスタリソースに設定されている PostgreSQL 用の別のデフォルトイメージまたは追加のプルシークレットを定義する

デフォルトでは、演算子は `postgresql-operator-controller-manager` という Kubernetes Deployment として `postgresql-operator-system` namespace にインストールされます。

---

**注釈:** 以下の例では、演算子展開のデフォルトの名前と名前空間を想定しています。

---

演算子のデプロイメントと同じ名前空間にあり、名前として `with postgresql-operator-controller-manager-config` にある ConfigMap / Secret を使用してカスタマイズできます。

---

**重要:** 構成の ConfigMap / Secret への変更は、演算子によって自動的に検出されないため、再読み込みするニーズがあります（以下を参照）。さらに、変更は構成が再ロードされた後に作成されたリソースにのみ適用されます。

---



---

**重要:** 演算子は最初に ConfigMap 値を処理し、次にシークレットの値をこのオーダーで処理します。その結果、両方の場所でパラメータが定義されている場合、シークレットのパラメーターが使用されます。

---

### 7.1 利用可能オプション

演算子は、ConfigMap / Secret で定義される以下の環境変数を探します。

| Name                         | Description                                             |
|------------------------------|---------------------------------------------------------|
| <code>EDB_LICENSE_KEY</code> | デフォルトのライセンスキー（クラスターで定義されていない場合のみ、できれば `Secret` で使用されます） |

次のページに続く

表 1 – 前のページからの続き

| Name                                | Description                                                                                                            |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <i>ENABLE_REDWOOD_BY_DEFAULT</i>    | EPAS を使用する場合、デフォルトで Redwood 互換性を有効にします。                                                                                |
| <i>INHERITED_ANNOTATIONS</i>        | `Cluster` メタデータで定義されている場合、ポッドを含むすべての生成されたリソースに継承される注釈名のリスト                                                             |
| <i>INHERITED_LABELS</i>             | `Cluster` メタデータで定義されている場合、ポッドを含むすべての生成されたリソースに継承される label 名のリスト                                                        |
| <i>PULL_SECRET_NAME</i>             | オペレーターの名前空間で定義され、イメージのダウンロードに使用される追加のプルシークレットの名前                                                                       |
| <i>ENABLE_AZURE_PVC_UPDATE</i>      | PVC がサイズ変更状態でスタックしている場合、Postgres ポッドを削除できません。この機能は主に Azure 環境用です（デフォルトは `false`）                                       |
| <i>ENABLE_INSTANCE_MANAGEMENT</i>   | <i>true</i> に設定すると、 <b>演算子の更新後にインスタスマネージャのインプレース更新を有効にします</b> 。この機能は主に Azure クラスターのローリング更新を回避します（デフォルトの <i>false</i> ） |
| <i>MONITORING_QUERIES_CONFIGMAP</i> | 作成されたすべてのクラスターに適用される一連のデフォルトクエリ（キー `queries` で指定される）を持つオペレーターの名前スペース内の ConfigMap の名前                                   |
| <i>MONITORING_QUERIES_SECRET</i>    | 作成されたすべてのクラスターに適用される、一連のデフォルトクエリ（キー `queries` で指定される）を持つオペレーターの名前スペース内のシークレットの名前                                       |

*INHERITED\_ANNOTATIONS* および *INHERITED\_LABELS* の値は、パスのようなワイルドカードをサポートしています。例、値 `example.com/*` は値 `example.com/one` と `example.com/two` の両方に一致します。

*PULL\_SECRET\_NAME* パラメータを使用して追加のプルシークレット名前を指定すると、演算子はそのシークレットを使用して、作成された PostgreSQLcluster ごとにプルシークレットを作成します。その秘密の名前付けは `<cluster-name>-pull` になります。

演算子が *PULL\_SECRET\_NAME* シークレットを探す名前空間は、演算子をインストールした場所です。演算子とその秘密を見つけることができない場合、設定パラメータを無視します。

**警告:** 以前のバージョンの演算子は、PostgreSQL クラスターをデプロイするネームスペース内に *PULL\_SECRET\_NAME* シークレットをコピーしました。バージョン "1.11.0" から、以前の説明とマッチするように動作が変更されました。演算子の以前のバージョンで作成されたプルシークレットは使用されていません。

## 7.2 演算子構成マップの定義

以下の例では、デフォルトのライセンスキー（つまり、会社キー）、後で展開される Cluster オブジェクトによって作成されたリソースによって継承される label/注釈名を定義し、in-place updates for the instance manager を有効にすることにより、演算子の動作をカスタマイズします。

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: postgresql-operator-controller-manager-config
  namespace: postgresql-operator-system
data:
  INHERITED_ANNOTATIONS: categories
  INHERITED_LABELS: environment, workload, app
  ENABLE_INSTANCE_MANAGER_INPLACE_UPDATES: true
```

## 7.3 演算子シークレットの定義

次の例では、デフォルトのライセンスキーを定義することにより、演算子の動作をカスタマイズしています。

```
apiVersion: v1
kind: Secret
metadata:
  name: postgresql-operator-controller-manager-config
  namespace: postgresql-operator-system
type: Opaque
data:
  EDB_LICENSE_KEY: <YOUR_BASE64_ENCODED_EDB_LICENSE_KEY_HERE>
```

## 7.4 構成をリロードするために演算子を再起動する

変更を有効にするには、演算子ポッドを再作成して構成マップを再ロードする必要があります。マニフェストを使用して Kubernetes に演算子をインストールした場合は、次を発行することでそれを行うことができます。

```
kubectl rollout restart deployment \
  -n postgresql-operator-system \
  postgresql-operator-controller-manager
```

それ以外の場合、OLM を使用して演算子をインストールした場合、または onOpenshift を実行している場合は、演算子がインストールされているネームスペースを指定して次のコマンドを実行します。

```
kubectl delete pods -n [NAMESPACE_NAME_HERE] \
-l app.kubernetes.io/name=cloud-native-postgresql
```

**警告:** カスタマイズは、演算子デプロイメントのリロード後に作成された *Cluster* リソースにのみ適用されます。

上記の例に従って、Cluster 定義に categories 注釈と environment、workload、または app ラベルのいずれかが含まれている場合、これらはデプロイメントによって生成されるすべてのリソースに継承されます。

## 7.5 PPROF HTTP サーバー

演算子は、ローカルホスト:6060 上の以下のエンドポイントで PPROF HTTP サーバーを公開できます。

- `/debug/pprof/``. Responds to a request for `"/debug/pprof/"` with an HTML page.
  - listing the available profiles
- `/debug/pprof/cmdline``. Responds with the running programs command line, with
  - arguments separated by NUL bytes.
- `/debug/pprof/profile``. Responds with the pprof-formatted cpu profile. Profiling
  - lasts for duration specified in seconds GET parameter, or for 30 seconds if not
  - specified.
- `/debug/pprof/symbol``. Looks up the program counters listed in the request,
  - responding with a table mapping program counters to function names.
- `/debug/pprof/trace``. Responds with the execution trace in binary form. Tracing
  - lasts for duration specified in seconds GET parameter, or for 1 second if not
  - specified.

演算子を有効にするには、演算子のデプロイメントを編集して、`--pprof-server=true` フラグを追加する必要があります。

これを行うには、次のコマンドを実行します。

```
kubectl edit deployment -n postgresql-operator-system postgresql-operator-controller-
→manager
```

次に、編集ページでコンテナの引数を下にスクロールして、`--pprof-server=true`、例を追加します。

### containers:

- **args:**
  - controller
  - --enable-leader-election
  - --config-map-name=postgresql-operator-controller-manager-config
  - --secret-name=postgresql-operator-controller-manager-config

(次のページに続く)

(前のページからの続き)

```
- --log-level=info
- --pprof-server=true # relevant line
command:
- /manager
```

変更を保存すると、展開でロールアウトが実行され、新しいポッドで PPROF サーバーが有効になります。ポッドが実行されたら、次のようにしてコンテナ内で実行できます。

```
kubectl exec -ti -n postgresql-operator-system <pod name> -- bash
```

一度実行と：

```
curl localhost:6060/debug/pprof/
```



## 第8章 EDB Audit logs

EDB Postgres Advanced Server で実行されているクラスターは、次のように *EDB Audit logs* を有効にできます。

```
apiVersion: postgresql.k8s.enterprisedb.io/v1
kind: Cluster
metadata:
  name: cluster-example
spec:
  instances: 3
  imageName: quay.io/enterprisedb/edb-postgres-advanced:13
  licenseKey: <LICENSE>

  postgresql:
    epas:
      audit: true

  storage:
    size: 1Gi
```

`.spec.postgresql.epas.audit: true` を設定すると、次のパラメーターが強制されます。

```
edb_audit = csv
edb_audit_destination = file
edb_audit_directory = /controller/log
edb_audit_filename = edb_audit
edb_audit_rotation_day = none
edb_audit_rotation_seconds = 0
edb_audit_rotation_size = 0
edb_audit_tag =
edb_log_every_bulk_value = false
```

他のパラメーターは、通常どおり `.spec.postgresql.parameters` を介して渡すことができます。

監査 CSV ログは、残りのすべてのログと同様に、JSON フォーマットで解析され、stdout にルーティングされます。

- `.logger` を `edb_audit` に設定
- `.msg` を `record` に設定 JSON オブジェクトとして解析されたレコード全体を含む `.record`

以下の例を参照してください。

```
{
  "level": "info",
  "ts": 1624629110.7641866,
  "logger": "edb_audit",
  "msg": "record",
  "record": {
    "log_time": "2021-06-25 13:51:50.763 UTC",
    "user_name": "postgres",
    "database_name": "postgres",
    "process_id": "68",
    "connection_from": "[local]",
    "session_id": "60d5df76.44",
    "session_line_num": "5",
    "process_status": "idle in transaction",
    "session_start_time": "2021-06-25 13:51:50 UTC",
    "virtual_transaction_id": "3/93",
    "transaction_id": "1183",
    "error_severity": "AUDIT",
    "sql_state_code": "00000",
    "message": "statement: GRANT EXECUTE ON function pg_catalog.pg_read_binary_
↪file(text) TO \"streaming_replica\"",
    "detail": "",
    "hint": "",
    "internal_query": "",
    "internal_query_pos": "",
    "context": "",
    "query": "",
    "query_pos": "",
    "location": "",
    "application_name": "",
    "backend_type": "client backend",
    "command_tag": "GRANT",
    "audit_tag": "",
    "type": "grant"
  },
  "logging_pod": "cluster-example-1",
}
```

レコードのフィールドの詳細については、**EDB** Audit file を参照してください。

ロギングの詳細については、[CloudNativePG documentation](#) を参照してください。

## 第9章 ‘EDB Postgres for Kubernetes Plugin’

Kubernetes の EDB Postgres は、Kubernetes でクラスターを管理するための `kubectl` のプラグインを提供します。このプラグインは [the CloudNativePG Plugin](#) に基づいており、OpenShift 環境で `oc` と連携できます。

### 9.1 機能

Kubernetes プラグインの EDB Postgres は、CloudNativePG プラグインと同じ機能にプラスで、以下を提供します。

- OpenShift 環境の `oc` で動作します。
- 管理できる

### 9.2 インストール

以下を使用して、システムにプラグインをインストールできます。

```
curl -sSfL \
  https://github.com/EnterpriseDB/kubectl-cnp/raw/main/install.sh | \
  sudo sh -s -- -b /usr/local/bin
```

#### 9.2.1 サポートされているアーキテクチャ

Kubernetes プラグインの EDB Postgres は現在、次のオペレーティングシステムとアーキテクチャ向けにビルドされています。

- Linux
- amd64
- アーム 5/6/7
- arm64
- s390x
- ppc64le
- macOS

- amd64
- arm64 Windows
- 386
- amd64
- アーム 5/6/7
- arm64

## 9.3 使用

プラグインをインストールしてデプロイしたら、次のように使用をスタートできます。

```
kubectl cnp <command> <args...>
```

### 9.3.1 ステータス

`status` コマンドは、以下を含むクラスターの現在のステータスの概要を提供します。

- **一般情報** : クラスターの名前、PostgreSQL のシステム ID、インスタンス、現在のタイムラインおよび WAL 内の位置
- **バックアップ** : リカバリ可能ポイント、および以下によって返される WAL アーカイビングステータスプライマリからのプライマリビュー-またはレプリカクラスターの場合
- **ストリーミングレプリケーション** : `pg_stat_replication` から直接取得した情報 プライマリインスタンスのビュー
- **instances** : 各 Postgres インスタンスに関する情報。各インスタンスが直接取得しますインスタンスマネージャ; スタンバイの場合、**Current LSN** フィールドは対応しますリカバリ中に再生された最新のログ先行書き込みの場所 (リプレイ LSN)。

**重要:** 上記のステータス情報は、異なる時間および異なる場所で取得されるため、戻り値がわずかに矛盾します。例、メインヘッダーの *Current Write LSN* の場所は、2つの異なる時間間隔で取得されるため、インスタンスステータスの *Current LSN* フィールドと異なる場合があります。

```
kubectl cnp status sandbox
```

```
Cluster in healthy state
Name:          sandbox
Namespace:     default
System ID:     7039966298120953877
PostgreSQL Image: quay.io/enterprisedb/postgresql:14.2
```

(次のページに続く)

(前のページからの続き)

```

Primary instance:  sandbox-2
Instances:         3
Ready instances:   3
Current Write LSN: 3AF/EAF6168 (Timeline: 8 - WAL File: 00000008000003AF00000075)

Continuous Backup status
First Point of Recoverability: Not Available
Working WAL archiving:      OK
Last Archived WAL:          00000008000003AE00000079 @ 2021-12-14T10:16:29.
➔340047Z
Last Failed WAL: -

Certificates Status
Certificate Name          Expiration Date          Days Left Until Expiration
-----
➔-
cluster-example-ca       2022-05-05 15:02:42 +0000 UTC 87.23
cluster-example-replication 2022-05-05 15:02:42 +0000 UTC 87.23
cluster-example-server    2022-05-05 15:02:42 +0000 UTC 87.23

Streaming Replication status
Name      Sent LSN      Write LSN      Flush LSN      Replay LSN      Write Lag      ⬇
➔Flush Lag      Replay Lag      State          Sync State      Sync Priority
-----
➔-----
sandbox-1  3AF/EB0524F0  3AF/EB011760  3AF/EAFED50  3AF/EAFED50  00:00:00.004461 ⬇
➔00:00:00.007901 00:00:00.007901 streaming quorum 1
sandbox-3  3AF/EB0524F0  3AF/EB030B00  3AF/EB030B00  3AF/EB011760  00:00:00.000977 ⬇
➔00:00:00.004194 00:00:00.008252 streaming quorum 1

Instances status
Name      Database Size  Current LSN      Replication role  Status  QoS      Manager⬇
➔Version
-----
➔-----
sandbox-1  302 GB        3AF/E9FFFFE0    Standby (sync)    OK      Guaranteed 1.11.0
sandbox-2  302 GB        3AF/EAF6168     Primary           OK      Guaranteed 1.11.0
sandbox-3  302 GB        3AF/EBAD5D18    Standby (sync)    OK      Guaranteed 1.11.0

```

また、`--verbose` または単に `-v` を追加することで、より詳細なバージョンのステータスを取得することもできます

```
kubectl cnp status sandbox --verbose
```

```

Cluster in healthy state
Name:                sandbox
Namespace:           default
System ID:            7039966298120953877
PostgreSQL Image:    quay.io/enterprisedb/postgresql:14.2
Primary instance:    sandbox-2
Instances:           3
Ready instances:     3
Current Write LSN:    3B1/61DE3158 (Timeline: 8 - WAL File: 0000000080000003B1000000030)

PostgreSQL Configuration
archive_command = /controller/manager wal-archive --log-destination /controller/log/
↳ postgres.json %p
archive_mode = on
archive_timeout = 5min
checkpoint_completion_target = 0.9
checkpoint_timeout = 900s
cluster_name = sandbox
dynamic_shared_memory_type = sysv
full_page_writes = on
hot_standby = true
jit = on
listen_addresses = *
log_autovacuum_min_duration = 1s
log_checkpoints = on
log_destination = csvlog
log_directory = /controller/log
log_filename = postgres
log_lock_waits = on
log_min_duration_statement = 1000
log_rotation_age = 0
log_rotation_size = 0
log_statement = ddl
log_temp_files = 1024
log_truncate_on_rotation = false
logging_collector = on
maintenance_work_mem = 2GB
max_connections = 1000
max_parallel_workers = 32
max_replication_slots = 32
max_wal_size = 15GB
max_worker_processes = 32
pg_stat_statements.max = 10000
pg_stat_statements.track = all
port = 5432

```

(次のページに続く)

(前のページからの続き)

```

shared_buffers = 16GB
shared_memory_type = sysv
shared_preload_libraries = pg_stat_statements
ssl = on
ssl_ca_file = /controller/certificates/client-ca.crt
ssl_cert_file = /controller/certificates/server.crt
ssl_key_file = /controller/certificates/server.key
synchronous_standby_names = ANY 1 ("sandbox-1","sandbox-3")
unix_socket_directories = /controller/run
wal_keep_size = 512MB
wal_level = logical
wal_log_hints = on
cnpg.config_sha256 = 3cfa683e23fe513afaae7c97b50ce0628e0cc634bca8b096517538a9a4428efc

```

#### PostgreSQL HBA Rules

```

# Grant local access
local all all peer map=local

# Require client certificate authentication for the streaming_replica user
hostssl postgres streaming_replica all cert
hostssl replication streaming_replica all cert
hostssl all cnpg_pooler_pgrounder all cert

# Otherwise use the default authentication method
host all all all scram-sha-256

```

#### Continuous Backup status

```

First Point of Recoverability: Not Available
Working WAL archiving: OK
Last Archived WAL: 000000080000003B000000001D @ 2021-12-14T10:20:42.
→272815Z
Last Failed WAL: -

```

#### Streaming Replication status

| Name             | Sent LSN        | Write LSN    | Flush LSN    | Replay LSN    | Write Lag       |       |
|------------------|-----------------|--------------|--------------|---------------|-----------------|-------|
| →Flush Lag       | Replay Lag      | State        | Sync State   | Sync Priority |                 |       |
| -----            | -----           | -----        | -----        | -----         | -----           | ----- |
| →-----           | -----           | -----        | -----        | -----         | -----           | ----- |
| sandbox-1        | 3B1/61E26448    | 3B1/61DF82F0 | 3B1/61DF82F0 | 3B1/61DF82F0  | 00:00:00.000333 | └     |
| →00:00:00.000333 | 00:00:00.005484 | streaming    | quorum       | 1             |                 |       |
| sandbox-3        | 3B1/61E26448    | 3B1/61E26448 | 3B1/61DF82F0 | 3B1/61DF82F0  | 00:00:00.000756 | └     |
| →00:00:00.000756 | 00:00:00.000756 | streaming    | quorum       | 1             |                 |       |

#### Instances status

(次のページに続く)

(前のページからの続き)

| Name      | Database Size | Current LSN  | Replication role | Status | QoS        | Manager |
|-----------|---------------|--------------|------------------|--------|------------|---------|
| ↪Version  |               |              |                  |        |            |         |
| -----     | -----         | -----        | -----            | -----  | ---        | -----   |
| ↪-----    |               |              |                  |        |            |         |
| sandbox-1 |               | 3B1/610204B8 | Standby (sync)   | OK     | Guaranteed | 1.11.0  |
| sandbox-2 |               | 3B1/61DE3158 | Primary          | OK     | Guaranteed | 1.11.0  |
| sandbox-3 |               | 3B1/62618470 | Standby (sync)   | OK     | Guaranteed | 1.11.0  |

このコマンドは、yaml および json フォーマットの出力もサポートしています。

### 9.3.2 プロモート

このコマンドの意味は、クラスター内のポッドをプライマリに **promote** することです。したがって、メンテナンス作業をスタートしたり、クラスターのスイッチオーバーシチュエーションをテストしたりできます。

```
kubectl cnp promote cluster-example cluster-example-2
```

または、インスタンスノード番号を使用してプロモートさせることができます

```
kubectl cnp promote cluster-example 2
```

### 9.3.3 証明書

EDB Postgres for Kubernetes 演算子を使用して作成されたクラスターは、CA と連携して TLS 認証証明書に署名します。

証明書を取得するには、資格情報を保存するシークレットの名前、クラスター名前、およびこの証明書のユーザを指定する必要があります

```
kubectl cnp certificate cluster-cert --cnp-cluster cluster-example --cnp-user appuser
```

Secret が作成されたら、**kubectl** を使用して取得できます

```
kubectl get secret cluster-cert
```

また、次のコマンドを使用したプレーンテキストでの同じコンテンツ：

```
kubectl get secret cluster-cert -o json | jq -r .data | map(@base64d) | .[]
```

### 9.3.4 再起動

`kubectl cnp restart` コマンドは、次の 2 つの場合に使用できます。

- 演算子にロールアウトのリスタートを調整するよう要求する特定のクラスターに対して。これは適用するのに便利です ConfigMap などのクラスター依存オブジェクトの構成変更カスタムモニタリングクエリを含む。
- 単一インスタンスのリスタートを要求します。インスタンスがクラスターのプライマリまたはポッドの削除と再作成 それはレプリカです。

```
# this command will restart a whole cluster in a rollout fashion
kubectl cnp restart [clusterName]

# this command will restart a single instance, according to the policy above
kubectl cnp restart [clusterName] [pod]
```

インプレースリスタートが要求されているが、スイッチオーバーなしでは変更を適用できない場合、スイッチオーバーはインプレースリスタートよりも優先されます。これの一般的なケースは、PostgreSQL イメージのマイナーアップグレードです。

---

**注釈:** ConfigMaps と Secrets をインスタンスによって\*\*自動的に\*\*リロードする場合、`k8s.enterprisedb.io/reload` キーを持つ label を追加できます。

---

### 9.3.5 リロード

`kubectl cnp reload` コマンドは、特定のクラスターの調整ループをトリガーするよう演算子に要求します。これは、カスタムモニタリングクエリを含む ConfigMap など、クラスターに依存するオブジェクトに構成の変更を適用するのに役立ちます。

次のコマンドは、特定のクラスターのすべての構成をリロードします。

```
kubectl cnp reload [cluster_name]
```

### 9.3.6 メンテナンス

`kubectl cnp maintenance` コマンドは、ネームスペース全体で 1 つ以上のクラスターを変更し、メンテナンスウィンドウの値を設定するのに役立ちます。次のフィールドが変更されます。

- `.spec.nodeMaintenanceWindow.inProgress`
- `.spec.nodeMaintenanceWindow.reusePVC`

これを引数 `set` および `unset` として受け入れ、`set` の場合は `inProgress` を `true` に、`unset` の場合は `false` に設定します。

デフォルトでは、`--reusePVC` フラグが渡されない限り、`reusePVC` は常に `false` に設定されます。

プラグインは、変更するクラスターのリストとそれらの新しい値の確認を求めます。これが受け入れられると、このアクションはリスト内のすべてのクラスターに適用されます。

Kubernetes クラスター内のすべての PostgreSQL をメンテナンスで設定する場合は、次のコマンドを記述する必要があります。

```
kubect1 cnp maintenance set --all-namespaces
```

そして、更新するすべてのクラスターのリストがあります

```
The following are the new values for the clusters
Namespace Cluster Name      Maintenance reusePVC
-----
default    cluster-example true        false
default    pg-backup      true        false
test       cluster-example true        false
Do you want to proceed? [y/n]: y
```

### 9.3.7 レポート

`kubect1 cnp report` コマンドは、さまざまな情報を ZIP ファイルにバンドルします。稼働環境のクラスターの問題をデバッグするために必要なコンテキストを提供することを目的としています。

`operator` と `cluster` の 2 つのサブコマンドがあります。

### 9.3.8 レポート演算子

`operator` サブコマンドは、演算子のデプロイメント、構成、およびイベントに関する情報を提供するよう演算子に要求します。

---

**重要:** Secrets および ConfigMaps のすべての機密情報は編集済みです。データマップには `** keys **` が表示されますが、値は空になります。フラグ `-S / --stopRedaction` は、リダクションを無効にし、値を表示します。ご自身のリスクでのみ使用してください。これはプライベートデータを共有します。

---



---

**注釈:** デフォルトでは、演算子ログは収集されませんが、`--logs` フラグを使用して演算子ログ収集を有効にできます

---

- **デプロイメント情報** : 演算子のデプロイメントと演算子のポッド
- **構成** : 演算子名前空間の秘密と構成マップ
- **イベント** : 演算子名前空間のイベント

- **webhook 設定** : webhook 設定の変更と検証
- **webhook サービス** : webhook サービス
- **logs** : JSON-lines フォーマットの演算子 Pod (オプション、デフォルトではオフ) のログ

このコマンドは、YAML フォーマットのさまざまなマニフェストを含む ZIP ファイルを生成します (デフォルトでは、ただし `-o` フラグを使用して JSON に設定可能)。 `-f` フラグを使用して、結果ファイルに明示的に名前を付けます。 `-f` フラグを使用しない場合、デフォルトのタイムスタンプ付きファイル名が zip ファイル用に作成されます。

```
kubectl cnp report operator
```

結果として

```
Successfully written report to "report_operator_<TIMESTAMP>.zip" (format: "yaml")
```

`-f` フラグセットしている場合:

```
kubectl cnp report operator -f reportRedacted.zip
```

ファイルを解凍すると、タイムスタンプ付きの最上位フォルダーが作成され、ディレクトリが整理されます。

```
unzip reportRedacted.zip
```

結果として:

```
Archive:  reportRedacted.zip
  creating: report_operator_<TIMESTAMP>/
  creating: report_operator_<TIMESTAMP>/manifests/
 inflating: report_operator_<TIMESTAMP>/manifests/deployment.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/operator-pod.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/events.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/validating-webhook-configuration.
→yaml
 inflating: report_operator_<TIMESTAMP>/manifests/mutating-webhook-configuration.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/webhook-service.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/postgresql-operator-ca-secret.yaml
 inflating: report_operator_<TIMESTAMP>/manifests/postgresql-operator-webhook-cert.
→yaml
```

機密情報が編集されていることを確認できます。

```
cd report_operator_<TIMESTAMP>/manifests/
head postgresql-operator-ca-secret.yaml
```

```
data:
  ca.crt: ""
  ca.key: ""
```

(次のページに続く)

(前のページからの続き)

```

metadata:
  creationTimestamp: "2022-03-22T10:42:28Z"
  managedFields:
    - apiVersion: v1
      fieldsType: FieldsV1
      fieldsV1:

```

-S (--stopRedaction) オプションをアクティブにすると、秘密が表示されます：

```
kubectl cnp report operator -f reportNonRedacted.zip -S
```

機密情報をビューしようとしていることを通知します。

```

WARNING: secret Redaction is OFF. Use it with caution
Successfully written report to "reportNonRedacted.zip" (format: "yaml")

```

```

unzip reportNonRedacted.zip
head postgresql-operator-ca-secret.yaml

```

```

data:
  ca.crt: LS0tLS1CRUdJTiBD...
  ca.key: LS0tLS1CRUdJTiBF...
metadata:
  creationTimestamp: "2022-03-22T10:42:28Z"
  managedFields:
    - apiVersion: v1
      fieldsType: FieldsV1

```

### 9.3.9 レポートクラスター

cluster サブコマンドは次を収集します。

- クラスターリソース：クラスター情報、`kubectl get cluster -o yaml` と同じ
- クラスターポッド：クラスター名にマッチングするクラスター名前空間のポッド
- クラスタージョブ：クラスター名とマッチングするクラスター名前空間にジョブがある場合は、ジョブ
- イベント：クラスター名前空間のイベント
- podlogs：JSON-lines フォーマットのクラスターポッド（オプション、デフォルトではオフ）のログ
- ジョブブログ：ジョブによって作成されたポッドのログ（オプション、デフォルトではオフ）JSON フォーマット

cluster サブコマンドは、operator が行うように、-f および -o フラグを受け入れます。-f フラグを使用しない場合、デフォルトのタイムスタンプ付きレポート名前が使用されます。無効です。

**注釈:** デフォルトでは、クラスターログは収集されませんが、`--logs` フラグを使用してクラスターログの収集を有効にできます

使用法:

```
kubectl-cnp report cluster [clusterName] -f <filename.zip> [flags]
```

`operator` サブコマンドとは異なり、`cluster` サブコマンドでは、クラスターがデフォルトの名前にない限り、クラスター名前と名前空間を指定する必要があることに注意してください。

```
kubectl cnp report cluster cluster-example-full -f report.zip -n example2
```

そして:

```
unzip report.zip
```

```
Archive:  report.zip
  creating: report_cluster_<TIMESTAMP>/
  creating: report_cluster_<TIMESTAMP>/manifests/
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster-pods.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster-jobs.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/events.yaml
```

`--logs` フラグを使用して、ポッドとジョブのログを ZIP に追加できることに注意してください。

```
kubectl cnp report cluster cluster-example-full -n example2 --logs
```

結果として:

```
Successfully written report to "report_cluster_<TIMESTAMP>.zip" (format: "yaml")
```

```
unzip report_cluster_<TIMESTAMP>.zip
```

```
Archive:  report_cluster_<TIMESTAMP>.zip
  creating: report_cluster_<TIMESTAMP>/
  creating: report_cluster_<TIMESTAMP>/manifests/
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster-pods.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/cluster-jobs.yaml
 inflating: report_cluster_<TIMESTAMP>/manifests/events.yaml
  creating: report_cluster_<TIMESTAMP>/logs/
 inflating: report_cluster_<TIMESTAMP>/logs/cluster-example-full-1.jsonl
  creating: report_cluster_<TIMESTAMP>/job-logs/
```

(次のページに続く)

(前のページからの続き)

```
inflating: report_cluster_<TIMESTAMP>/job-logs/cluster-example-full-1-initdb-qnnvw.  
→jsonl  
inflating: report_cluster_<TIMESTAMP>/job-logs/cluster-example-full-2-join-tvj8r.  
→jsonl
```

## 第10章 ‘Red Hat OpenShift’

Kubernetes の EDB Postgres は Red Hat OpenShift Container Platform (OCP) version 4.x での実行が認定されており、[Red Hat Catalog](#) から直接入手できます。

このセクションの目標は、組織のセキュリティおよびアクセス制御ポリシーに基づいて、Kubernetes 用の EDB Postgres の最適なインストールメソッドを決定するのをヘルプすることです。

Kubernetes の EDB Postgres は、次の方法でインストールおよび管理できます。

- OpenShift web console
- oc と呼ばれる OpenShift command-line interface (CLI)、フルコントロール

Kubernetes の EDB Postgres は、OpenShift で定義されているすべての利用可能なインストールモードをサポートしています。

- すべてのネームスペースでのクラスタ全体
- ローカル、単一の名前空間
- local、マルチプルのネームスペースを監視します（ローカルを使用した場合のみ使用可能）

---

**注釈:** プロジェクトは、注釈が追加された Kubernetes 名前空間であり、通常のユーザーのリソースへのアクセスを管理する中心的な手段です。

---

ほとんどの場合、Kubernetes の EDB Postgres のデフォルトのクラスタ全体のインストールは、PostgreSQL クラスターの中央管理または委任管理（RBAC 定義に従って特定のユーザー/プロジェクトに制限されます-以下の [OpenShift の重要な概念](#) および [ユーザーと権限](#) を参照）を推奨します。

---

**重要:** インストールとアップグレードの両方のプロセスでは、`cluster-admin` 権限を持つアカウントを使用して OpenShift Container Platform クラスターにアクセスする必要があります。 [Default cluster roles](#) から、`cluster-admin` は\* "任意のプロジェクトで任意のアクションを実行できるスーパーユーザーです。ローカルバインディングでユーザにバインドされると、プロジェクト内のすべてのリソースに対するクォータとすべてのアクションを完全に制御できます\*"。

---

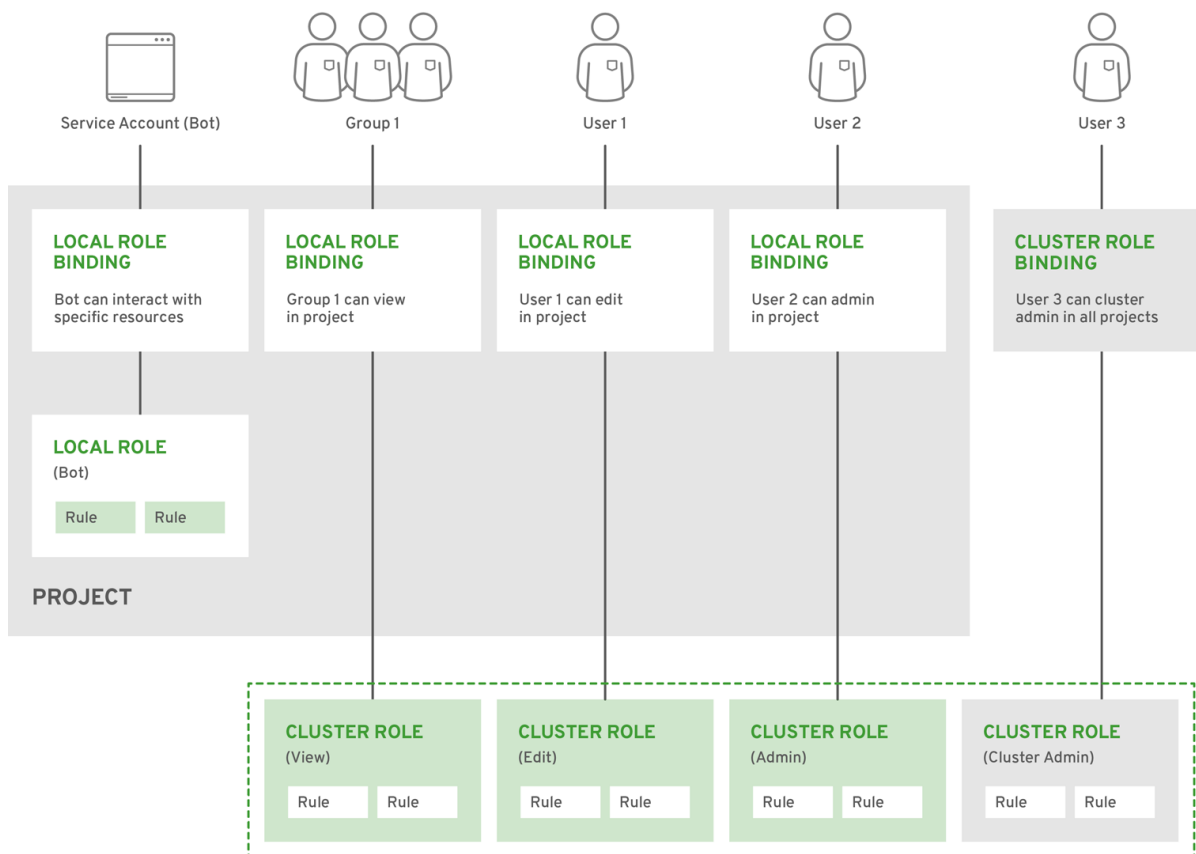
## 10.1 OpenShift の重要な概念

Kubernetes 向けの EDB Postgres 演算子が OpenShift 環境にどのように適合するかを理解するには、以下の Kubernetes 関連のトピックに精通する必要があります。

- オペレーター
- 認証
- 役割ベースのアクセス制御（RBAC）による承認
- サービスアカウントとユーザー
- ルール、ロール、およびバインディング
- プロジェクトを通じたクラスター RBAC とローカル RBAC

これは、演算子のデフォルトのクラスター全体のインストールに必要な高い権限に満足できない場合に特に当てはまります。

また、クラスターの役割、ローカルの役割、クラスターのロールのバインディング、ローカルのロールのバインディング、ユーザー、グループ、およびサービスアカウント間の関係を明確に示しているため、OpenShift の文書から以下の図を選択しました。



OPENSIFT\_415489\_0218

The 事前定義された RBAC オブジェクト below contains important information about how EDB Postgres for Kubernetes adheres to Kubernetes and OpenShift RBAC implementation, covering default installed cluster roles, roles, service accounts.

上記の概念に精通している場合は、選択したインストールメソッドに直接進むことができます。そうでない場合は、OpenShift の文書と Red Hat ブログから入手した以下のリソースを読むことをお勧めします。

- [Operator Lifecycle Manager \(OLM\) concepts and resources](#)
- [Understanding authentication](#)
- [Role-based access control \(RBAC\)](#)、  
承認のためのルール、ロール、バインディング、およびプロジェクトを介したクラスター RBAC  
とローカル RBAC のカバー
- [Default project service accounts and roles](#)
- [With Kubernetes Operators comes great responsibility](#)

### 10.1.1 クラスターサービスバージョン (CSV)

技術的には、演算子は、EDB によって定義されたクラスターサービスバージョン (CSV) に従って、Operator Lifecycle Manager (OLM) を介して EDB で実行するように設計されています。

CSV は、ユーザインターフェイス (Web ダッシュボードから利用可能) だけでなく、オペレーターが必要とする RBAC ルールと、演算子が定義および所有するカスタムリソース (例、`Cluster` など) も定義する YAML マニフェストです。CSV は、演算子で使用可能な `installModes` も定義します。つまり、`AllNamespaces` (クラスター全体)、`SingleNamespace` (単一プロジェクト)、`MultiNamespace` (マルチプロジェクト)、および `OwnNamespace` です。

### 10.1.2 マルチテナント管理の制限

Red Hat OpenShift Container Platform は、単一クラスターに異なるバリエーションの演算子を同時にインストールするための限定的なサポートを提供します。他の演算子と同様に、EDB Postgres for Kubernetes はコントロールプレーンの拡張になります。コントロールプレーンは OpenShift クラスターのすべてのテナント (プロジェクト) で共有されるため、オペレーターもマルチテナント環境で共有リソースになります。

Operator Lifecycle Manager (OLM) は、異なる名前空間にマルチプル回オペレーターをインストールできますが、1 つの重要な制限があります：すべて同じ演算子の同じ API バージョンを共有する必要があります。

詳細については、OpenShift 文書の [Operator groups](#) を参照してください。

## 10.2 Web コンソールからのインストール

Kubernetes 向けの EDB Postgres 演算子は、OpenShift ダッシュボードから直接 Red Hat OperatorHub で見つけることができます。

1. Web コンソールで **Operators** -> **OperatorHub** ページに移動します。
2. **Database** セクションをスクロールするか、**Filter by keyword** ボックスにキーワード (この場合は「PostgreSQL」) を入力して Kubernetes オペレーターの EDB Postgres を見つけ、それを選択します。

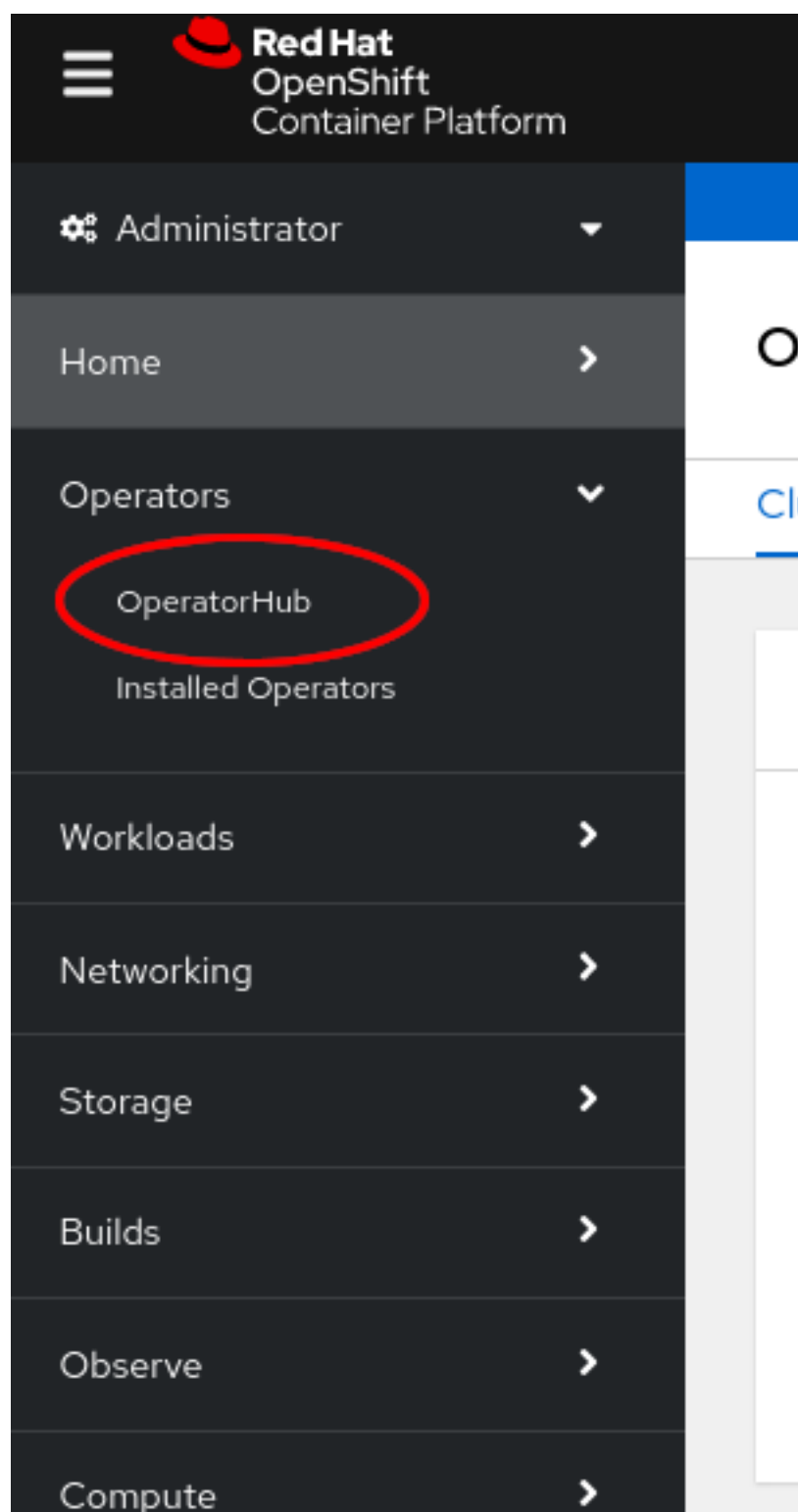


図 1: Menu OperatorHub

The screenshot displays the OpenShift OperatorHub interface. On the left, a dark sidebar contains navigation links: Administrator, Home, Operators (selected), Installed Operators, Workloads, Networking, Storage, Builds, Observe, Compute, and User Management. Under 'Operators', 'OperatorHub' is selected. The main content area is titled 'OperatorHub' and includes a description: 'Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service interface for installing Operators.' Below this, a list of categories is shown: All Items, AI/Machine Learning, Application Runtime, Big Data, Cloud Provider, Database (highlighted with a red circle), Developer Tools, Development Tools, Drivers and plugins, Integration & Delivery, Logging & Tracing, Modernization & Migration, Monitoring, Networking, OpenShift Optional, Security, Storage, Streaming & Messaging, and Other. A 'Source' section at the bottom left shows 'Red Hat (1)' and 'Certified (15)'. On the right, a 'Database' filter is active, and a search bar says 'Filter by keyword...'. The main content area shows two operator cards. The first is '[DEPRECATED] Hazelcast Operator' provided by Hazelcast, Inc. The second is 'Cloud Native PostgreSQL' provided by EnterpriseDB Corporation, which is highlighted with a red circle. This card includes the EDB logo and a description: 'Operator to manage Postgres high availability clusters with a primary/standby architecture.'

図 2: Install OperatorHub

3. オペレーターに関する情報を読み、`Install` を選択します。

4. 次の `Operator installation` ページでは、選択することを想定しています。

- インストールモード： [クラスター全体のインストール](#) または `single namespace` インストール-更新チャンネル：現在安定（stable）のみが利用可能-Red Hat によって認定された演算子の新しいリリースの市場での入手可能性に続く承認戦略：- `Automatic`：OLM が自動的にアップグレード新しいバージョンの実行中の演算子- `Manual`：OpenShift は、`Installed Operators` セクションでの承認を要求することにより、人間の介入を待機します

---

**重要：** 演算子のアップグレードのプロセスは 'Installation and upgrades' で説明されています。

---

---

**重要：** 同じクラスター内の単一プロジェクト（技術的には `OwnNamespace` インストールモード）に演算子をマルチプル回インストールすることが可能です。API が同じである限り、アップグレードポリシーが異なるすべてのネームスペースに演算子がインストールされます（[マルチテナント管理の制限](#) を参照）。

---

演算子のクラスター全体でのインストールとローカルでのインストールを選択することが重要な転換点です。既存のローカルインストールで演算子をグローバルにインストールしようとすると、以下のエラーがスローされ、ブロックされます。続行する場合は、最初に演算子のローカルインストールをすべて削除する必要があります。

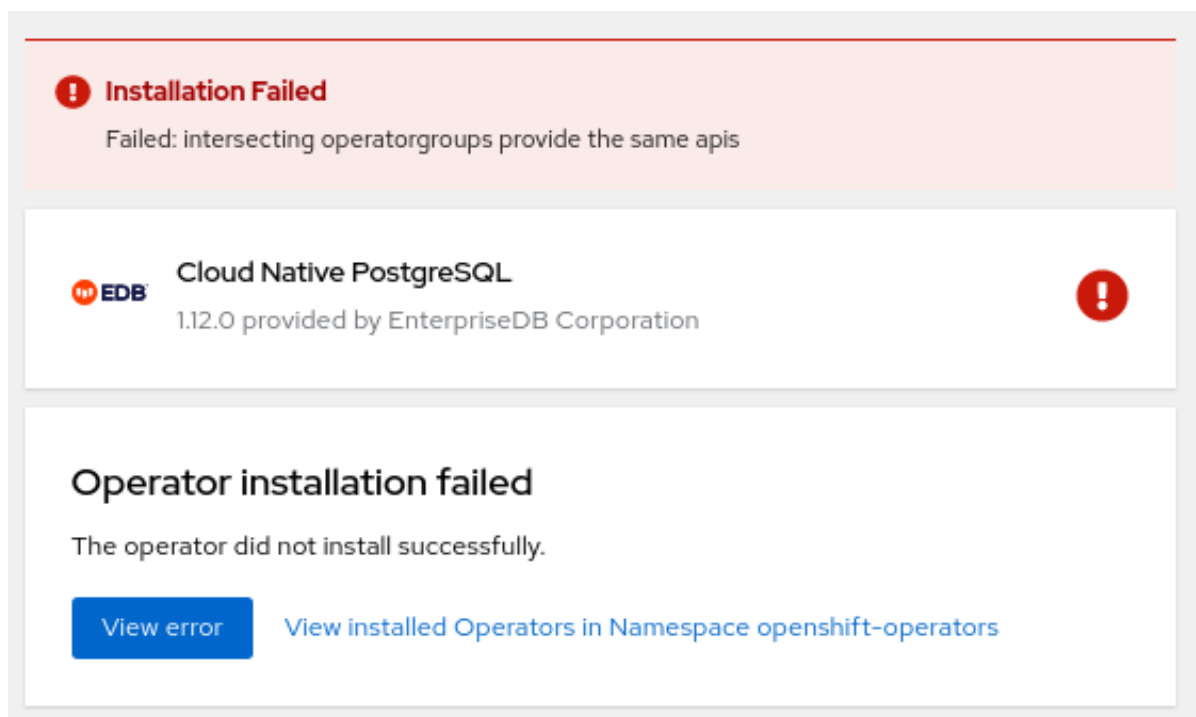


図 3: Install failing

## 10.2.1 クラスター全体のインストール

クラスター全体のインストールでは、OpenShift にデフォルトの `openshift-operators` 名前空間に theOperator をインストールし、クラスター内のすべてのプロジェクトで使用できるように make 求めています。これはデフォルトであり、Kubernetes 用 EDB Postgres をインストールするための通常推奨されるアプローチです。

**警告:** これは、OpenShift クラスター内のすべてのユーザが Cloud Native PostgreSQL Operator を使用したり、*Cluster* オブジェクトをデプロイしたり、独自のネームスペースで実行されている *Cluster* オブジェクトを見たりできることを意味しません。Kubernetes の管理されたカスタムリソース（主に *Cluster* ロール）の EDB Postgres と対話オーダーに、ユーザがネームスペースで必要とする特別なロールがいくつかあります。詳細については、[ユーザーと権限](#) を参照してください。

Web コンソールから、`All namespaces on the cluster (default)` as Installation mode を選択します。

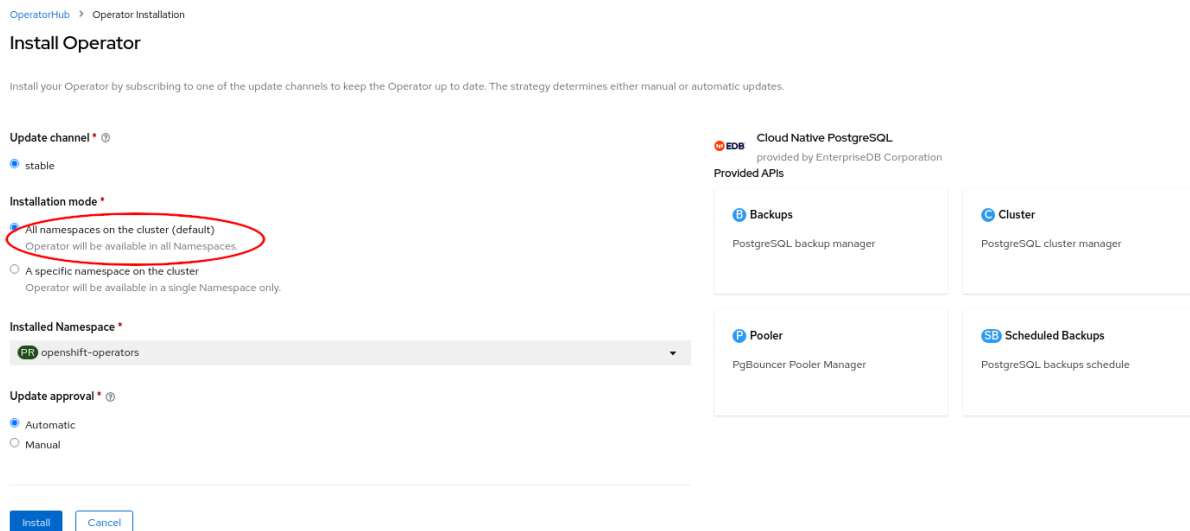


図 4: Install all namespaces

その結果、演算子はすべての名前空間で可視されます。それ以外の場合、他の OpenShift 演算子と同様に、問題を報告している `Workloads` → `Pods` ページの `openshift-operators` project にあるポッドのログを確認して、さらにトラブルシューティングを行います。

## 10.2.2 単一プロジェクトのインストール

単一プロジェクトのインストールでは、OpenShift に指定されたネームスペースに theOperator をインストールし、そのプロジェクトでのみ使用可能に make に要求しています。

**警告:** これは、ネームスペース内のすべてのユーザが Cloud Native PostgreSQL Operator を使用したり、*Cluster* オブジェクトをデプロイしたり、ネームスペースで実行されている *Cluster* オブジェクトを見たりすることもできるという意味ではありません。クラスター全体のインストールモードと同様に、Kubernetes

の管理されたカスタムリソース（主に *Cluster* の 1 つ）の EDB Postgres と対話オーダーに、ユーザーがネームスペースで持つ必要がある特別なロールがいくつかあります。詳細については、[ユーザーと権限](#) を参照してください。

Web コンソールから、**A specific namespace on the cluster as Installation mode** を選択し、ターゲットの名前空間を選択します（この例では、`ed proj-dev`）。

OperatorHub > Operator Installation

### Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

**Update channel \***

- ☒ stable

**Installation mode \***

- ☐ All namespaces on the cluster (default)  
Operator will be available in all namespaces.
- ☒ A specific namespace on the cluster  
Operator will be available in a single Namespace only.

**Installed Namespace \***

ed proj-dev

**Update approval \***

- ☒ Automatic
- ☐ Manual

**Cloud Native PostgreSQL**  
provided by EnterpriseDB Corporation

**Provided APIs**

- Backups**  
PostgreSQL backup manager
- Cluster**  
PostgreSQL cluster manager
- Pooler**  
PgBouncer Pooler Manager
- Scheduled Backups**  
PostgreSQL backups schedule

**Install** **Cancel**

図 5: Install one namespace

その結果、演算子は選択した名前空間でのみ可視されます。これは、**Installed operators** ページから確認できます。

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: proj-dev

### Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#). Or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

| Name                                                                         | Managed Namespaces | Status                  | Last updated  | Provided APIs                                                                                                     |
|------------------------------------------------------------------------------|--------------------|-------------------------|---------------|-------------------------------------------------------------------------------------------------------------------|
| <b>Cloud Native PostgreSQL</b><br>112.0 provided by EnterpriseDB Corporation | <b>proj-dev</b>    | Succeeded<br>Up to date | 4 minutes ago | <a href="#">Backups</a><br><a href="#">Cluster</a><br><a href="#">Pooler</a><br><a href="#">Scheduled Backups</a> |

図 6: Install one namespace list

問題が発生した場合は、**Workloads** → **Pods** ページから、問題を報告している選択したインストールネームスペース内のすべてのポッドのログを確認して、さらにトラブルシューティングを行います。

このインストールプロセスは、同じ OpenShift クラスター内のマルチプルネームスペースで繰り返すことができ、異なるプロジェクトの演算子の独立したインストールを可能にします。この場合、**make マルチテナント管理の制限** を読んでください。

## 10.3 oc CLI を介したインストール

**重要:** `oc` コマンドラインインタフェースのインストール方法については、OpenShift CLI (``oc``) のインストール<OpenShift CLI (`oc`) のインストール> を参照してください。

OpenShift Container Platform Web コンソールを使用する代わりに、OperatorHub から Kubernetes Operator 用の EDB Postgres をインストールし、`oc` コマンドラインインタフェースを使用してサブスクリプションを作成できます。`oc` CLI を使用して、すべての名前空間、単一の名前空間、または複数の名前空間に演算子をインストールできます。

**警告:** OpenShift は現在、複数のネームスペースのインストールをサポートしています。ただし、[definition of multiple target namespaces for an operator may be removed in future versions of OpenShift](#)。

このセクションでは、主に、`OperatorGroup` および `Subscription` オブジェクトを作成することにより、複数のプロジェクトに演算子をシンプル例でインストールする方法を説明します。

**注釈:** この例では、演算子を `my-operators` 名前空間にインストールし、`web-staging`、`web-prod`、`bi-staging`、および `make` 名前空間でのみ使用できるようにします。フリーにプロジェクトの名前を変更したり、名前空間を追加/削除したりできます。

1. `cloud-native-postgresql` 演算子が OperatorHub から利用できることを確認します。

```
oc get packagemanifests -n openshift-marketplace cloud-native-postgresql
```

2. 演算子を調べて、インストールモード（特に `MultiNamespace`）と使用可能なチャネルを確認します。

```
oc describe packagemanifests -n openshift-marketplace cloud-native-postgresql
```

3. `web-staging`、`web-prod`、`bi-staging`、および `bi-prod` 名前空間を対象とするように、`my-operators` 名前空間に `OperatorGroup` オブジェクトを作成します。

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cloud-native-postgresql
  namespace: my-operators
spec:
  targetNamespaces:
    - web-staging
    - web-prod
    - bi-staging
    - bi-prod
```

**重要:** または、[Target namespace selection](#) で説明されているように、label セレクターを使用して名前空間をリストすることもできます。

4. openshift-marketplace の certified-operators ソースで使用可能な cloud-native-postgresql 演算子の stable チャンネルにサブスクライブするために、my-operators 名前空間に Subscription オブジェクトを作成します（以前に手順 1 および 2 で見つけたように）。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cloud-native-postgresql
  namespace: my-operators
spec:
  channel: stable
  name: cloud-native-postgresql
  source: certified-operators
  sourceNamespace: openshift-marketplace
```

5. OperatorGroup および Subscription オブジェクトの上記の YAML ファイル定義で `oc apply -f` を使用します。

このセクションで説明するメソッドは、Kubernetes の事前定義された ClusterRoles の EDB Postgres を選択したネームスペースの特定のユーザーにマッピングできるため、適切な RoleBinding オブジェクトと組み合わせると非常に強力です。

**注釈:** 上記の手順は、単一プロジェクトのバインドにも使用できます。唯一の違いは、指定されたターゲット名前空間の数（1 つ）と、場合によっては演算子グループの名前空間（理想的には、ターゲット名前空間と同じ）です。

上記のオペレーションの結果は、下のイメージに示すように、Web コンソールから確認することもできます。

### 10.3.1 oc を使用したクラスター全体のインストール

必要に応じて、`oc` を使用して演算子をグローバルにインストールすることもできます。openshift-operators 名前空間で global-operators と呼ばれるデフォルトの OperatorGroup を利用し、同じ名前空間で cloud-native-postgresql 演算子の新しい Subscription オブジェクトを作成します：

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cloud-native-postgresql
  namespace: openshift-operators
```

(次のページに続く)

Project: my-operators ▼

## Installed Operators

Installed Operators

Understanding Operators

Namespace. For more information, see the ClusterServiceVersion using the [Operator SDK](#).

Managed Namespaces

- NS bi-prod
- NS bi-staging
- NS web-prod
- NS web-staging

| Name                                                            | Namespace    | Status                  | Provided APIs                                                                                                     |
|-----------------------------------------------------------------|--------------|-------------------------|-------------------------------------------------------------------------------------------------------------------|
| Cloud Native PostgreSQL<br>provided by EnterpriseDB Corporation | 4 Namespaces | Succeeded<br>Up to date | <a href="#">Backups</a><br><a href="#">Cluster</a><br><a href="#">Pooler</a><br><a href="#">Scheduled Backups</a> |

図 7: Multi namespace installation displayed in the web console

(前のページからの続き)

```
spec:
  channel: stable
  name: cloud-native-postgresql
  source: certified-operators
  sourceNamespace: openshift-marketplace
```

上記の YAML ファイルを使用して `oc apply -f` を実行すると、すべてのネームスペースで演算子が使用可能になります。

### 10.3.2 OpenShift CLI ( oc ) のインストール

`oc` コマンドは、OpenShift コマンドラインインタフェース (CLI) を表します。システムにインストールすることを強くお勧めします。以下に、OpenShift ダッシュボードから `oc` をインストールするための基本的な手順を示します。

まず、ダッシュボードの右上隅にあるマークを選択します。

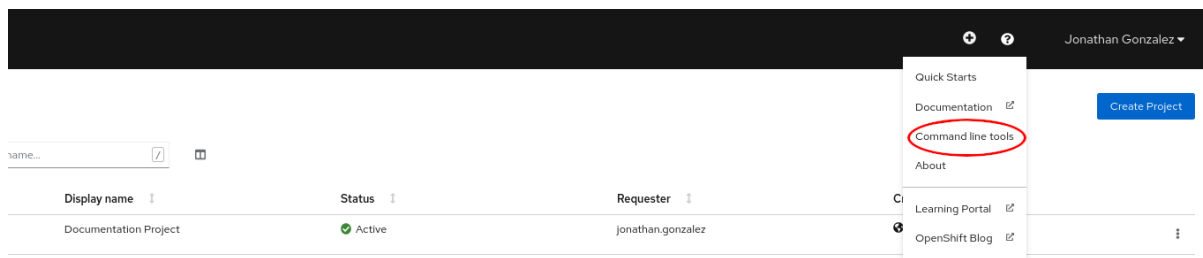


図 8: Command Line Tools menu

次に、オペレーティングシステムとアーキテクチャの観点からニーズに合ったバイナリをダウンロードして、指示に従ってください。

### Command Line Tools

[Copy Login Command](#)

#### oc - OpenShift Command Line Interface (CLI)

With the OpenShift command line interface, you can create applications and manage OpenShift projects from a terminal.

The oc binary offers the same capabilities as the kubectl binary, but it is further extended to natively support OpenShift Container Platform features.

- [Download oc for Linux for x86\\_64](#)
- [Download oc for Mac for x86\\_64](#)
- [Download oc for Windows for x86\\_64](#)
- [Download oc for Linux for ARM 64](#)
- [Download oc for Linux for IBM Power, little endian](#)
- [Download oc for Linux for IBM Z](#)
- [LICENSE](#)

#### helm - Helm 3 CLI

Helm 3 is a package manager for Kubernetes applications which enables defining, installing, and upgrading applications packaged as Helm Charts.

[Download Helm](#)

#### odo - Developer-focused CLI for OpenShift

odo is a fast, iterative, and straightforward CLI tool for developers who write, build, and deploy applications on OpenShift.

odo abstracts away complex Kubernetes and OpenShift concepts, thus allowing developers to focus on what is most important to them: code.

[Download odo](#)

図 9: Command Line Tools

## 10.4 事前定義された RBAC オブジェクト

EDB Postgres for Kubernetes には、RBAC ポリシーの構成に関して重要なロール事前定義されたリソースのセットが付属しています。

### 10.4.1 カスタムリソース定義 (CRD)

Kubernetes の EDB Postgres 演算子は、次のカスタムリソース定義 (CRD) を所有しています。

- Backup
- Cluster
- Pooler
- ScheduledBackup

これを確認するには、次を実行します。

```
oc get customresourcedefinitions.apiextensions.k8s.io | grep postgresql
```

次のような結果が返されます。

|                                                 |                      |
|-------------------------------------------------|----------------------|
| backups.postgresql.k8s.enterprisedb.io          | 20YY-MM-DDTHH:MM:SSZ |
| clusters.postgresql.k8s.enterprisedb.io         | 20YY-MM-DDTHH:MM:SSZ |
| poolers.postgresql.k8s.enterprisedb.io          | 20YY-MM-DDTHH:MM:SSZ |
| scheduledbackups.postgresql.k8s.enterprisedb.io | 20YY-MM-DDTHH:MM:SSZ |

### 10.4.2 サービスアカウント

演算子がインストールされているネームスペース（デフォルトでは `b`）には、次の定義済みのサービスアカウントが含まれています：`builder`、`default`、`deployer`、および最も重要なのは、CSV によって管理されている

---

**重要：** Kubernetes のサービスアカウントは名前空間付きのリソースです。明示的に許可されない限り、定義された名前空間の外部ではサービスアカウントにアクセスできません。

---

これを確認するには、次を実行します。

```
oc get serviceaccounts -n openshift-operators
```

次のような結果が返されます。

| NAME                        | SECRETS | AGE |
|-----------------------------|---------|-----|
| builder                     | 2       | ... |
| default                     | 2       | ... |
| deployer                    | 2       | ... |
| postgresql-operator-manager | 2       | ... |

`default` サービスアカウントは Kubernetes によって自動的に作成され、すべてのネームスペースに存在します。`builder` および `deployer` サービスアカウントは、OpenShift によって自動的に作成されます（[Default project service accounts and roles](#) を参照）。

`postgresql-operator-manager` サービスアカウントは、CloudNative PostgreSQL 演算子が PostgreSQL クラスターの管理において Kubernetes / OpenShift コントロールプレーンのパートとして使用するために使用するアカウントです。

---

**重要：** `postgresql-operator-manager` ServiceAccount を削除しないでください。削除すると、演算子の作業が停止する可能性があります。

---

### 10.4.3 クラスターの役割

Operator Lifecycle Manager (OLM) は一連の clusterrole オブジェクトを自動的に作成して、ロールバインドの定義と RBAC ポリシーの詳細な実装を促進します。Kubernetes の EDB Postgres のパートであるカスタム ResourceDefinitions に適用されるルールを持つクラスターロールと、より広い Kubernetes / OpenShift レールの一部であるクラスターロールがあります。

### 10.4.4 Kubernetes CRD の EDB Postgres のクラスターロール

every CRD owned by EDB Postgres for Kubernetes's CSV の場合、OLM は、顧客向けのユーザーおよびサービスアカウントが使用できる事前定義されたクラスターロールをいくつかデプロイします。特に：

- リソースの完全な管理のためのロール（`admin` サフィックス）
- リソースを編集するロール（`edit` サフィックス）
- リソースをビューするロール（`view` サフィックス）
- 実際の CRD をビューするロール（`crdview` サフィックス）

**重要:** クラスターロール自体はセキュリティの脅威ではありません。これらは、OpenShift で、特定のプロジェクトの実際のユーザーまたはグローバルに後で「バインド」されるロールのテンプレートを定義するための推奨される方法です。実際、クラスターロールは、グローバルアクセス許可の場合は *ClusterRoleBinding* オブジェクトと、ローカルアクセス許可の場合は *RoleBinding* オブジェクトと組み合わせて使用できます。これにより、ローカルロールを介して個々のプロジェクト内でカスタマイズを可能にしながら、マルチブルのプロジェクトでクラスターロールを再利用できます。

以下を実行することにより、事前定義されたクラスターの役割のリストを確認できます。

```
oc get clusterroles | grep postgresql
```

次のような結果が返されます。

|                                                    |                      |
|----------------------------------------------------|----------------------|
| backups.postgresql.k8s.enterprisedb.io-v1-admin    | YYYY-MM-DDTHH:MM:SSZ |
| backups.postgresql.k8s.enterprisedb.io-v1-crdview  | YYYY-MM-DDTHH:MM:SSZ |
| backups.postgresql.k8s.enterprisedb.io-v1-edit     | YYYY-MM-DDTHH:MM:SSZ |
| backups.postgresql.k8s.enterprisedb.io-v1-view     | YYYY-MM-DDTHH:MM:SSZ |
| cloud-native-postgresql.VERSION-HASH               | YYYY-MM-DDTHH:MM:SSZ |
| clusters.postgresql.k8s.enterprisedb.io-v1-admin   | YYYY-MM-DDTHH:MM:SSZ |
| clusters.postgresql.k8s.enterprisedb.io-v1-crdview | YYYY-MM-DDTHH:MM:SSZ |
| clusters.postgresql.k8s.enterprisedb.io-v1-edit    | YYYY-MM-DDTHH:MM:SSZ |
| clusters.postgresql.k8s.enterprisedb.io-v1-view    | YYYY-MM-DDTHH:MM:SSZ |
| poolers.postgresql.k8s.enterprisedb.io-v1-admin    | YYYY-MM-DDTHH:MM:SSZ |
| poolers.postgresql.k8s.enterprisedb.io-v1-crdview  | YYYY-MM-DDTHH:MM:SSZ |
| poolers.postgresql.k8s.enterprisedb.io-v1-edit     | YYYY-MM-DDTHH:MM:SSZ |

(次のページに続く)

(前のページからの続き)

|                                                            |                      |
|------------------------------------------------------------|----------------------|
| poolers.postgresql.k8s.enterprisedb.io-v1-view             | YYYY-MM-DDTHH:MM:SSZ |
| scheduledbackups.postgresql.k8s.enterprisedb.io-v1-admin   | YYYY-MM-DDTHH:MM:SSZ |
| scheduledbackups.postgresql.k8s.enterprisedb.io-v1-crdview | YYYY-MM-DDTHH:MM:SSZ |
| scheduledbackups.postgresql.k8s.enterprisedb.io-v1-edit    | YYYY-MM-DDTHH:MM:SSZ |
| scheduledbackups.postgresql.k8s.enterprisedb.io-v1-view    | YYYY-MM-DDTHH:MM:SSZ |

get コマンドを使用して、他の Kubernetes リソースとしての実際のロールを検査できます。例：

```
oc get -o yaml clusterrole clusters.postgresql.k8s.enterprisedb.io-v1-admin
```

以下の関連するスキミングされた出力を見ると、clusters.postgresql.k8s.enterprisedb.io-v1-admin クラスターロールが、postgresql.k8s.enterprisedb.io API グループによって定義された cluster リソースのすべてを有効にしていることがわかります。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: clusters.postgresql.k8s.enterprisedb.io-v1-admin
rules:
- apiGroups:
  - postgresql.k8s.enterprisedb.io
  resources:
  - clusters
  verbs:
  - *
```

### 10.4.5 Kubernetes CRD のクラスターロール

Subscription オブジェクトを特定の名前空間（オペレーターのクラスタ全体のインストールの場合は eg openshift-operators ）にインストールすると、OLM は、演算子が使用する postgresql-operator-manager サービスアカウントに権限を付与するために使用されるクラスターロールも作成し演算子。このクラスターロールの名前は、インストールされているオペレーターのバージョンとインストールの時間に依存するため、異なります。

次のコマンドを実行して取得できます。

```
oc get clusterrole --selector=olm.owner.kind=ClusterServiceVersion
```

次に、上記のクエリー（cloud-native-postgresql.VERSION-HASH の形式である必要があります）によって返された名前を使用して、describe コマンドでルール、リソース、および動詞を確認できます。

```
oc describe clusterrole cloud-native-postgresql.VERSION-HASH
```

```

Name:          cloud-native-postgresql.VERSION.HASH
Labels:        olm.owner=cloud-native-postgresql.VERSION
               olm.owner.kind=ClusterServiceVersion
               olm.owner.namespace=openshift-operators
               operators.coreos.com/cloud-native-postgresql.openshift-operators=
Annotations:   <none>
PolicyRule:
  Resources                                     Non-Resource URLs
  ↪Resource Names  Verbs
  -----
  ↪-----
  configmaps                                     []                []
               [create delete get list patch update watch]
  secrets                                       []                []
               [create delete get list patch update watch]
  services                                     []                []
               [create delete get list patch update watch]
  deployments.apps                             []                []
               [create delete get list patch update watch]
  poddisruptionbudgets.policy                 []                []
               [create delete get list patch update watch]
  backups.postgresql.k8s.enterprisedb.io       []                []
               [create delete get list patch update watch]
  clusters.postgresql.k8s.enterprisedb.io       []                []
               [create delete get list patch update watch]
  poolers.postgresql.k8s.enterprisedb.io        []                []
               [create delete get list patch update watch]
  scheduledbackups.postgresql.k8s.enterprisedb.io []                []
               [create delete get list patch update watch]
  persistentvolumeclaims                     []                []
               [create delete get list patch watch]
  pods/exec                                   []                []
               [create delete get list patch watch]
  pods                                         []                []
               [create delete get list patch watch]
  jobs.batch                                   []                []
               [create delete get list patch watch]
  podmonitors.monitoring.coreos.com            []                []
               [create delete get list patch watch]
  serviceaccounts                             []                []
               [create get list patch update watch]
  rolebindings.rbac.authorization.k8s.io        []                []
               [create get list patch update watch]
  roles.rbac.authorization.k8s.io              []                []
               [create get list patch update watch]

```

(次のページに続く)

(前のページからの続き)

|                                                              |    |    |
|--------------------------------------------------------------|----|----|
| leases.coordination.k8s.io                                   | [] | [] |
| [create get update]                                          |    |    |
| events                                                       | [] | [] |
| [create patch]                                               |    |    |
| mutatingwebhookconfigurations.admissionregistration.k8s.io   | [] | [] |
| [get list update]                                            |    |    |
| validatingwebhookconfigurations.admissionregistration.k8s.io | [] | [] |
| [get list update]                                            |    |    |
| customresourcedefinitions.apiextensions.k8s.io               | [] | [] |
| [get list update]                                            |    |    |
| namespaces                                                   | [] | [] |
| [get list watch]                                             |    |    |
| nodes                                                        | [] | [] |
| [get list watch]                                             |    |    |
| clusters.postgresql.k8s.enterprisedb.io/status               | [] | [] |
| [get patch update watch]                                     |    |    |
| poolers.postgresql.k8s.enterprisedb.io/status                | [] | [] |
| [get patch update watch]                                     |    |    |
| configmaps/status                                            | [] | [] |
| [get patch update]                                           |    |    |
| secrets/status                                               | [] | [] |
| [get patch update]                                           |    |    |
| backups.postgresql.k8s.enterprisedb.io/status                | [] | [] |
| [get patch update]                                           |    |    |
| scheduledbackups.postgresql.k8s.enterprisedb.io/status       | [] | [] |
| [get patch update]                                           |    |    |
| Pods/status                                                  | [] | [] |
| [get]                                                        |    |    |
| clusters.postgresql.k8s.enterprisedb.io/finalizers           | [] | [] |
| [update]                                                     |    |    |
| poolers.postgresql.k8s.enterprisedb.io/finalizers            | [] | [] |
| [update]                                                     |    |    |

**重要:** 上記の権限は、オペレーターのサービスアカウントが Kubernetes API サーバーと対話するためにのみ予約されています。Cluster、Pooler、Backup、および ScheduledBackup リソースとのみ対話する演算子のユーザーは、直接アクセスできません（[Kubernetes CRD の EDB Postgres のクラスターロール](#) を参照）。

演算子は宣言的方法で PostgreSQL の管理に関連する多くの操作を自動化します。そうでなければ、マニュアルで命令的な介入が必要になります。このような操作には、RBAC（サービスアカウントなど）、ポッド（セキュリティコンテキストコンストレインなど）、および Postgreslevels（TLS 証明書など）のセキュリティ関連事項も含まれます。

演算子がこれらの高い権限をニーズとする理由の詳細については、[Security / Cluster / RBAC](#) を参照してください。

## 10.5 ユーザーと権限

Kubernetes 演算子用の EDB Postgres を使用する非常に一般的な方法は、`cluster-admin` ロールに依存し、リソースを一元管理することです。

または、他の演算子またはリソースと同様に、Kubernetes / OpenShift によって利用可能になった RBAC フレームワークを使用できます。

例、他のクラウドネイティブアプリケーションと同様に、`clusters.postgresql.k8s.enterprisedb.io-v1-admin` クラスターロールを特定の名前空間の特定のグループまたはユーザーにバインドすることに興味があるかもしれません。次の例は、そのクラスターロールを `the web-prod` プロジェクトの特定のユーザーにバインドします。

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: web-prod-admin
  namespace: web-prod
subjects:
- kind: User
  apiGroup: rbac.authorization.k8s.io
  name: mario@cioni.org
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: clusters.postgresql.k8s.enterprisedb.io-v1-admin
```

同じプロセスは、他の事前定義された `ClusterRole` で繰り返すことができます。

一方、クラスターロールを使用しない場合は、次の例のように特定の名前空間ロールを作成できます。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: web-prod-view
  namespace: web-prod
rules:
- apiGroups:
  - postgresql.k8s.enterprisedb.io
  resources:
  - clusters
  verbs:
  - get
  - list
  - watch
```

次に、このロールを特定のユーザーに割り当てます。

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: web-prod-view
  namespace: web-prod
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: web-prod-view
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: web-prod-developer1

```

この最後の例では、そのネームスペース（web-prod）の演算子が管理するすべてのリソース管理権限（verbs は \* と等しい）を持つロールを作成します。

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: web-prod-admin
  namespace: web-prod
rules:
- apiGroups:
  - postgresql.k8s.enterprisedb.io
  resources:
  - clusters
  - backups
  - scheduledbackups
  - poolers
  verbs:
  - *

```

## 10.6 プーラーイメージのカスタマイズ

デフォルトでは、Pooler リソースは、quay.io/enterprisedb/pgbouncer イメージで実行される pgbouncer コンテナを持つポッドを作成します。

具体的には、\* “テンプレート” セクションを開き、“仕様>コンテナ”の下にある“コンテナの追加” \* を選択することにより、高度なインタフェースからイメージ名前を変更できます。

次に：

- pgbouncer をコンテナの名前として設定します（ポッドテンプレートの必須フィールド）
- 必要に応じて\* “Image” \*フィールドを設定します（下のイメージを参照）

Template

Pod Template Spec for pod to be created.

Spec

Specification of the desired behavior of the pod. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status>

Containers \*

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

+ Add Container

図 10: Pooler template

Tty

☐ tty

Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.

Image

registry.company.com/pgbouncer:1.x.x

Docker image name. More info: <https://kubernetes.io/docs/concepts/containers/images> This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Working Dir

Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

図 11: Pooler image

## 第11章 ‘Free evaluation’

Kubernetes の EDB Postgres はフリーの評価版で利用できます。Vanilla/ Community PostgreSQL と EDB Postgres Advanced ではプロセスが異なります。

用語および詳細については、'License and license keys' を参照してください。

### 11.1 PostgreSQL の評価

デフォルトでは、EDB Postgres for Kubernetes は Community PostgreSQL の利用可能な最新バージョンをインストールします。演算子は、30 日間続くクラスターの暗黙的な試用ライセンスを自動的に生成します。

このライセンスは、評価、概念実証、CI / CD パイプラインとの統合などに最適です。

PostgreSQL コンテナイメージは [quay.io/enterprisedb/postgresql](https://quay.io/enterprisedb/postgresql) で入手できます。

### 11.2 EDB Postgres Advanced Server の評価

EDB Postgres Advanced Server で Kubernetes 用の EDB Postgres を使用できます。

[EDB website](#) からトライアルライセンスキーをリクエストする必要があります。

EDB Postgres Advanced コンテナイメージは、[quay.io/enterprisedb/edb-postgres-advanced](https://quay.io/enterprisedb/edb-postgres-advanced) で入手できます。

ライセンスキーを受け取ったら、Cluster デプロイメント構成ファイルの spec セクションの [quay.io/enterprisedb/edb-postgres-advanced](https://quay.io/enterprisedb/edb-postgres-advanced) リポジトリを指すように `imageName` を設定することにより、EDB Postgres Advanced Server を使用できます。



## 第12章 ‘License and license keys’

演算子が作業するには、ライセンスキーが常に必要です。

唯一の例外は、コミュニティ PostgreSQL で演算子を実行する場合です。この場合、ライセンスキーが設定されていない場合、クラスターは defaulttrial ライセンスで開始されます。これは 30 日後に自動的に期限切れになります。

### 12.1 会社レベルのライセンスキー

ライセンスキーを使用すると、インストール環境で PostgreSQLcluster を無制限に作成できます。

ライセンスキーは、演算子がデプロイされている同じネームスペースの `Secret` で使用できるニーズがあります（`ConfigMap` も使用できますが、ライセンスキーには推奨されません）。

会社レベルのライセンスがインストールされると、ライセンスキーの有効性をクラスターステータス内で確認できます。

```
kubectl get cluster cluster_example -o yaml
[...]
status:
  [...]
  licenseStatus:
    licenseExpiration: "2021-11-06T09:36:02Z"
    licenseStatus: Trial
    valid: true
    isImplicit: false
    isTrial: true
[...]
```

#### 12.1.1 YAML マニフェストを介した Kubernetes インストール

YAML マニフェストを使用して Kubernetes に演算子をインストールすると、デフォルトで `postgresql-operator-system` 名前空間にデプロイされます。

名前空間名前とライセンスキーを指定すると、次のコマンドで構成マップを作成できます。

```
kubectl create configmap -n [NAMESPACE_NAME_HERE] \
  postgresql-operator-controller-manager-config \
  --from-literal=EDB_LICENSE_KEY=[LICENSE_KEY_HERE]
```

新しい構成を適用するには、オペレーターポッドを再作成する必要があります。次のコマンドを使用できます。

```
kubectl rollout restart deployment -n [NAMESPACE_NAME_HERE] \
  postgresql-operator-controller-manager
```

### 12.1.2 OLM を使用した OpenShift および Kubernetes のインストール

代わりに OLM を使用する場合（つまり、OpenShift または OperatorHub の Kubernetes 上）、単一のネームスペースに演算子をインストールするか、すべてのネームスペースで使用可能に make かを選択できます。2 番目のオプションはデフォルトのオプションであり、演算子は OpenShift では in openshift-operators、OperatorHub では operators にインストールされます。

次のコマンドを使用して、目的のネームスペース（openshift-operators、operators、または選択したネームスペース）にシークレットを作成することにより、クラスター全体に特定のライセンスキーを追加できます。

```
kubectl create secret generic -n [NAMESPACE_NAME_HERE] \
  postgresql-operator-controller-manager-config \
  --from-literal=EDB_LICENSE_KEY=[LICENSE_KEY_HERE]
```

現在の演算子ポッドを削除する必要があります。新しいポッドは自動的に再作成され、シークレットが使用されます：

```
kubectl delete pods -n [NAMESPACE_NAME_HERE] \
  -l app.kubernetes.io/name=cloud-native-postgresql
```

## 12.2 クラスターレベルのライセンスキー

各 Cluster リソースの定義には licenseKey パラメータがあります。クラスターステータスで、有効期限とライセンスに関する詳細情報を確認できます。

```
kubectl get cluster cluster_example -o yaml
[...]
status:
  [...]
  licenseStatus:
    licenseExpiration: "2021-11-06T09:36:02Z"
    licenseStatus: Trial
    valid: true
    isImplicit: false
    isTrial: true
  [...]
```

有効期限を延長したり、クラスターを稼動ライセンスに移動したりするために、クラスターライセンスキーはいつでも新しいものに更新できます。

## 12.3 クラスターレベルでのライセンスキーシークレット

各 Cluster リソースには、秘密の名前とキーを含む `licenseKeySecret` パラメータを含めることもできます。その秘密には、EDB によって提供されるライセンスキーが含まれています。

このフィールドは `licenseKey` よりも優先されます。有効期限を延長するオーダー、または試用ライセンスから稼動ライセンスに切り替えるために、シークレットを変更すると更新されます。

Kubernetes の EDB Postgres は、[enterprisedb.com/limited-use-license](https://enterprisedb.com/limited-use-license) で入手可能な EDB Limited Usage License Agreement の下で配布されています。

Kubernetes の EDB Postgres : Copyright (C) 2019-2022 EnterpriseDB Corporation。

## 12.4 ライセンスの有効期限が切れるとどうなりますか

ライセンスの有効期限が切れると、演算子はクラスターでの調整の試みを中止し、そのステータスを管理するために事実上停止します。これには、自動フェイルオーバーやスイッチオーバーなどの自己修復機能や高可用性機能も含まれます。

ポッドとデータは引き続き利用可能です。



## 第13章 ‘API Reference’

Kubernetes の EDB Postgres は、次のカスタムリソースを定義する Kubernetes API を拡張します。

- バックアップ
- クラスター
- プーラー
- *ScheduledBackup*

すべてのリソースは `postgresql.k8s.enterprisedb.io/v1` API で定義されています。

以下に、定義済みのリソースの説明があります。

<!--これ以降はすべて `make apidoc` を介して生成されます-->

- *AffinityConfiguration*
- *AzureCredentials*
- バックアップ
- *BackupConfiguration*
- *BackupList*
- *BackupSource*
- *BackupSpec*
- *BackupStatus*
- *BarmanObjectStoreConfiguration*
- *BootstrapConfiguration*
- *BootstrapInitDB*
- *BootstrapPgBaseBackup*
- *BootstrapRecovery*
- 証明書の構成
- 証明書ステータス
- クラスター
- *ClusterCondition*
- *ClusterList*

- *ClusterSpec*
- *ClusterStatus*
- *ConfigMapKeySelector*
- *ConfigMapResourceVersion*
- *DataBackupConfiguration*
- *EPASConfiguration*
- *EmbeddedObjectMetadata*
- *ExternalCluster*
- *GoogleCredentials*
- *InstanceID*
- *LDAPBindAsAuth*
- *LDAPBindSearchAuth*
- *LDAPConfig*
- *LocalObjectReference*
- *MonitoringConfiguration*
- *NodeMaintenanceWindow*
- *PgBouncerIntegrationStatus*
- *PgBouncerSecrets*
- *PgBouncerSpec*
- *PodMeta*
- *PodTemplateSpec*
- プーラー
- *PoolerIntegrations*
- *PoolerList*
- *PoolerSecrets*
- *PoolerSpec*
- *PoolerStatus*
- *PostgresConfiguration*
- *RecoveryTarget*
- *ReplicaClusterConfiguration*
- *RollingUpdateStatus*

- *S3Credentials*
- *ScheduledBackup*
- *ScheduledBackupList*
- *ScheduledBackupSpec*
- *ScheduledBackupStatus*
- *SecretKeySelector*
- *SecretVersion*
- *SecretsResourceVersion*
- *StorageConfiguration*
- *WalBackupConfiguration*

## 13.1 AffinityConfiguration

AffinityConfiguration には、ポッドのアフィニティルールを作成するために必要な情報が含まれています

| Name                                | Description                                                                                                                                                                                                                          | Type              |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <code>enablePodAntiAffinity`</code> | ポッドの非アフィニティをアクティブにします。このフィールドが明示的に <code>false</code> に設定されていない限り、演算子はポッドの非アフィニティを定義します                                                                                                                                              | bool              |
| <code>topologyKey`</code>           | 非アフィニティ構成に使用する TopologyKey。詳細については、k8s の文書を参照してください- <b>必須</b>                                                                                                                                                                       | string            |
| <code>nodeSelector`</code>          | NodeSelector は、ポッドを実行できるノードを定義するために使用されるキーと値のペアのマップです。詳細： <a href="https://kubernetes.io/docs/concepts/configuration/assign-pod-node/">https://kubernetes.io/docs/concepts/configuration/assign-pod-node/</a>                        | map[string]string |
| <code>tolerations`</code>           | 許容値は、汚染されたノードで実行できるようにオーダーに、すべてのポッドに設定する必要がある許容値のリストです。詳細： <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/">https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/</a> | corev1.Toleration |

次のページに続く

表 1 – 前のページからの続き

| Name                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Type                    |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| <code>podAntiAffinityType`</code>      | PodAntiAffinityType を使用すると、クラスターインスタンス間のポッドの非アフィニティをスケジューリング時に強力な要件と見なす必要があるかどうかをユーザが決定できます。許可される値は「preferred」（空の場合のデフォルト）または「required」です。「必須」に設定すると、既存のノードがすべて必要なポッドの非アフィニティルールにマッチしない場合、新しい kubernetes ノードが追加されるまでインスタンスが保留状態になる可能性があります。詳細： <a href="https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#inter-pod-affinity-and-anti-affinity">https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#inter-pod-affinity-and-anti-affinity</a> | string                  |
| <code>additionalPodAntiAffinity</code> | AdditionalPodAntiAffinity は、EnablePodAntiAffinity が true（デフォルト）に設定されている場合に演算子によって生成されたものに追加されるポッドの非アフィニティ用語を指定するか、false に設定された場合に排他的に使用されます。                                                                                                                                                                                                                                                                                                                                             | *corev1.PodAntiAffinity |
| <code>additionalPodAffinity`</code>    | AdditionalPodAffinity では、すべてのクラスターのポッドに渡されるポッドアフィニティ用語を指定できます。                                                                                                                                                                                                                                                                                                                                                                                                                           | *corev1.PodAffinity     |

## 13.2 AzureCredentials

AzureCredentials は、ファイルを Azure Blob Storage にアップロードするために使用される資格情報の種類です。接続文字列には、必要なすべての情報が含まれています。接続文字列が指定されていない場合、ストレージアカウント名と次のいずれか（1 つのみ）が必要です。

- storageKey-storageSasToken

| Name                          | Description                          | Type                     |
|-------------------------------|--------------------------------------|--------------------------|
| <code>connectionString</code> | 使用する接続文字列                            | <i>SecretKeySelector</i> |
| <code>storageAccount`</code>  | データをアップロードするストレージアカウント               | <i>SecretKeySelector</i> |
| <code>storageKey`</code>      | ストレージアカウント名前と組み合わせて使用されるストレージアカウントキー | <i>SecretKeySelector</i> |
| <code>storageSasToken`</code> | ストレージアカウント名前と組み合わせて使用される共有アクセス署名     | <i>SecretKeySelector</i> |

## 13.3 バックアップ

バックアップは、バックアップ API のスキーマです

| Name                  | Description                                                                                                                                                                                                                                                                       | Type                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <code>metadata</code> | NA                                                                                                                                                                                                                                                                                | <code>metav1.ObjectMeta</code> |
| <code>spec</code>     | バックアップの望ましい動作の仕様。詳細: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a>                                        | <code>BackupSpec</code>        |
| <code>status</code>   | バックアップの最新の状態。このデータは最新ではない可能性があります。システムによって設定されます。読み取り専用。詳細: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a> | <code>BackupStatus</code>      |

## 13.4 BackupConfiguration

BackupConfiguration は、クラスターのバックアップの取得方法を定義します。現在、サポートされている唯一のバックアップメソッドは `barmanObjectStore` です。詳細と例については、文書の「バックアップリカバリー」セクションを参照してください

| Name                           | Description                                                                                                  | Type                                        |
|--------------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------------|
| <code>barmanObjectStore</code> | barman-cloud ツールスイートの構成                                                                                      | <code>BarmanObjectStoreConfiguration</code> |
| <code>retentionPolicy</code>   | RetentionPolicy は、バックアップおよび WAL（つまり「60d」）に使用される保持ポリシーです。保持ポリシーは「XXu」のフォームで表されます。XX は正の整数で、u は [dwm]-日、週、月です。 | string                                      |

## 13.5 BackupList

BackupList にはバックアップのリストが含まれます

| Name            | Description                                                                                                                                                                                                                  | Type                   |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <i>metadata</i> | 標準リストメタデータ。詳細: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> | <i>metav1.ListMeta</i> |
| <i>items</i>    | バックアップのリスト-必須                                                                                                                                                                                                                | バックアップ                 |

## 13.6 BackupSource

BackupSource には、復元元のバックアップと、それを正しく復元するために必要な情報が含まれています。

| Name              | Description                                                                                                    | Type                     |
|-------------------|----------------------------------------------------------------------------------------------------------------|--------------------------|
| <i>endpointCA</i> | EndpointCA は、barman エンドポイントの CA バンドルを保存します。証明書発行者と barman-cloud-wal-archive のエラーを回避するために、自己署名証明書を使用する場合に役立ちます。 | <i>SecretKeySelector</i> |

## 13.7 BackupSpec

BackupSpec は、バックアップの望ましい状態を定義します

| Name           | Description   | Type                        |
|----------------|---------------|-----------------------------|
| <i>cluster</i> | バックアップするクラスター | <i>LocalObjectReference</i> |

## 13.8 BackupStatus

BackupStatus は、観測されたバックアップの状態を定義します

| Name                    | Description                                  | Type                    |
|-------------------------|----------------------------------------------|-------------------------|
| <i>s3Credentials</i>    | S3 にデータをアップロードするために使用される資格情報                 | <i>S3Credentials</i>    |
| <i>azureCredentials</i> | Azure Blob Storage にデータをアップロードするために使用される資格情報 | <i>AzureCredentials</i> |

次のページに続く

表 8 – 前のページからの続き

| Name                           | Description                                                                                                    | Type                     |
|--------------------------------|----------------------------------------------------------------------------------------------------------------|--------------------------|
| <code>googleCredentials</code> | Google Cloud Storage にデータをアップロードするために使用する資格情報                                                                  | <i>GoogleCredentials</i> |
| <code>`endpointCA`</code>      | EndpointCA は、barman エンドポイントの CA バンドルを保存します。証明書発行者と barman-cloud-wal-archive のエラーを回避するために、自己署名証明書を使用する場合に役立ちます。 | <i>SecretKeySelector</i> |
| <code>`endpointURL`</code>     | データをクラウドにアップロードするために使用されるエンドポイント。自動エンドポイント検出をオーバーライドします                                                        | string                   |
| <code>`destinationPath`</code> | バックアップを保存するパス (s3:// bucket / パス/to/folder) このパスは、異なる宛先フォルダーで、WAL とデータに使用されます-必須                               | string                   |
| <code>`serverName`</code>      | S3 上のサーバー名前。このパラメータを省略するとクラスター名前が使用されます                                                                        | string                   |
| <code>`encryption`</code>      | S3 API に必要な暗号化メソッド                                                                                             | string                   |
| <code>`backupId`</code>        | Barman バックアップの ID                                                                                              | string                   |
| <code>`phase`</code>           | 最後のバックアップステータス                                                                                                 | BackupPhase              |
| <code>`startedAt`</code>       | バックアップが開始されたとき                                                                                                 | <i>metav1.Time</i>       |
| <code>`stoppedAt`</code>       | バックアップが終了したとき                                                                                                  | <i>metav1.Time</i>       |
| <code>`beginWal`</code>        | 開始 WAL                                                                                                         | string                   |
| <code>`endWal`</code>          | エンディング WAL                                                                                                     | string                   |
| <code>`beginLSN`</code>        | 開始 xlog                                                                                                        | string                   |
| <code>`endLSN`</code>          | 終わりの xlog                                                                                                      | string                   |
| <code>`error`</code>           | 検出されたエラー                                                                                                       | string                   |
| <code>`commandOutput`</code>   | 未使用。古いバージョンとの互換性のために保持されます。                                                                                    | string                   |
| <code>`commandError`</code>    | エラーの場合のバックアップコマンド出力                                                                                            | string                   |
| <code>`instanceID`</code>      | バックアップの取得元のインスタンスを識別する情報                                                                                       | <i>InstanceID</i>        |

## 13.9 BarmanObjectStoreConfiguration

BarmanObjectStoreConfiguration には、S3 互換オブジェクトストレージに対して Barman を使用したバックアップ構成が含まれています

| Name                            | Description                                 | Type                    |
|---------------------------------|---------------------------------------------|-------------------------|
| <code>`s3Credentials`</code>    | S3 にデータをアップロードするために使用する資格情報                 | <i>S3Credentials</i>    |
| <code>`azureCredentials`</code> | Azure Blob Storage にデータをアップロードするために使用する資格情報 | <i>AzureCredentials</i> |

次のページに続く

表 9 – 前のページからの続き

| Name                           | Description                                                                                                  | Type                           |
|--------------------------------|--------------------------------------------------------------------------------------------------------------|--------------------------------|
| <code>googleCredentials</code> | Google Cloud Storage にデータをアップロードするために使用する資格情報                                                                | <i>GoogleCredentials</i>       |
| <code>endpointURL</code>       | データをクラウドにアップロードするために使用されるエンドポイント。自動エンドポイント検出をオーバーライドします                                                      | string                         |
| <code>endpointCA</code>        | EndpointCA は、barman エンドポイントの CA バンドルを保存します。証明書発行者と barman-cloud-wal-archive のエラーを回避するために自己署名証明書を使用する場合に役立ちます | <i>SecretKeySelector</i>       |
| <code>destinationPath</code>   | バックアップを保存するパス (s3:// bucket / パス/to/folder) このパスは、異なる宛先フォルダーで、WAL とデータに使用されます-必須                             | string                         |
| <code>serverName</code>        | S3 上のサーバー名前。このパラメータを省略するとクラスター名前が使用されます                                                                      | string                         |
| <code>wal</code>               | WAL ストリームのバックアップの構成。定義されていない場合、WAL ファイルは圧縮されずに保存され、バケットのデフォルトポリシーに従って、オブジェクトストアで暗号化されていない場合があります。            | <i>WalBackupConfiguration</i>  |
| <code>data</code>              | データファイルのバックアップに使用される構成定義されていない場合、ベースバックアップファイルは圧縮されずに格納され、バケットのデフォルトポリシーに従って、オブジェクトストアに暗号化されない場合があります。       | <i>DataBackupConfiguration</i> |
| <code>tags</code>              | タグは、Barman の--tags オプションに渡されるキーと値のペアのリストです。                                                                  | map[string]string              |
| <code>historyTags</code>       | HistoryTags は、Barman の--history-tags オプションに渡されるキーと値のペアのリストです。                                                | map[string]string              |

## 13.10 BootstrapConfiguration

BootstrapConfiguration には、PostgreSQL クラスターの作成方法に関する情報が含まれています。サポートされている方法の中では、1つのブートストラップメソッドのみを定義できます。指定されていない場合、`initdb` はブートストラップメソッドとして使用されます。詳細については、文書のブートストラップページを参照してください。

| Name                  | Description                 | Type                     |
|-----------------------|-----------------------------|--------------------------|
| <code>initdb</code>   | initdb 経由でクラスターをブートストラップします | <i>BootstrapInitDB</i>   |
| <code>recovery</code> | バックアップからクラスターをブートストラップします   | <i>BootstrapRecovery</i> |

次のページに続く

表 10 – 前のページからの続き

| Name                 | Description                                              | Type                         |
|----------------------|----------------------------------------------------------|------------------------------|
| <i>pg_basebackup</i> | 互換性のある別の PostgreSQL インスタンスの物理的バックアップを取るクラスターをブートストラップします | <i>BootstrapPgBaseBackup</i> |

## 13.11 BootstrapInitDB

BootstrapInitDB は、initdb を使用する場合のブートストラッププロセスの構成です。詳細については、文書のブートストラップページを参照してください。

| Name                          | Description                                                                                                            | Type                        |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| <code>`database`</code>       | アプリケーションが使用するデータベースの名前。デフォルト： <i>app</i> 。-必須                                                                          | string                      |
| <code>`owner`</code>          | アプリケーションで使用されるインスタンス内のデータベースの所有者の名前。デフォルトは <code>`database`</code> キーの値です。-必須                                          | string                      |
| <code>`secret`</code>         | ユーザデータベースの所有者の初期資格情報を含むシークレットの名前。空の場合、新しいシークレットが最初から作成されます                                                             | <i>LocalObjectReference</i> |
| <code>`redwood`</code>        | Redwood 互換性を有効/無効にする必要がある場合。EPAS が必要で、EPAS のデフォルトは true                                                                | *bool                       |
| <code>`options`</code>        | クラスターの作成時に initdb に渡す必要があるオプションのリスト。非推奨：これにより、矛盾した構成が発生する可能性があります。代わりに明示的に提供されたパラメーターを使用してください。定義されている場合、明示的な値は無視されます。 | string                      |
| <code>`dataChecksums`</code>  | -k オプションを initdb に渡して、データページのチェックサムを有効にするかどうか（デフォルト： <i>false</i> ）                                                    | *bool                       |
| <code>`encoding`</code>       | initdb のオプション <code>--encoding`</code> として渡される値（デフォルト： <i>UTF8</i> ）                                                   | string                      |
| <code>`localeCollate`</code>  | initdb のオプション <code>--lc-collate`</code> として渡される値（デフォルト： <i>C</i> ）                                                    | string                      |
| <code>`localeCTYPE`</code>    | initdb のオプション <code>--lc-ctype`</code> として渡される値（デフォルト： <i>C</i> ）                                                      | string                      |
| <code>`walSegmentSize`</code> | initdb の <code>--wal-segsize`</code> オプションに渡されるメガバイト（1～1024）の値（デフォルト：空、PostgreSQL のデフォルト：16MB）                         | int                         |
| <code>`postInitSQL`</code>    | クラスターの作成直後にスーパーユーザとして実行される SQL クエリのリスト-細心の注意を払って使用する（デフォルトでは空）                                                         | string                      |

次のページに続く

表 11 – 前のページからの続き

| Name                                 | Description                                                                 | Type   |
|--------------------------------------|-----------------------------------------------------------------------------|--------|
| <code>postInitApplication-SQL</code> | 作成直後にアプリケーションデータベースでスーパーユーザとして実行される SQL クエリのリスト-細心の注意を払って使用する（デフォルトでは空）     | string |
| <code>postInitTemplateSQL</code>     | クラスターの作成後に `template1` でスーパーユーザとして実行される SQL クエリのリスト-細心の注意を払って使用する（デフォルトでは空） | string |

## 13.12 BootstrapPgBaseBackup

BootstrapPgBaseBackup には、既存の PostgreSQL クラスターの物理的バックアップを取るために必要な構成が含まれています

| Name                | Description                 | Type   |
|---------------------|-----------------------------|--------|
| <code>source</code> | 物理的バックアップを取る必要があるサーバーの名前-必須 | string |

## 13.13 BootstrapRecovery

BootstrapRecovery には、指定された名前でバックアップを復元するために必要な構成が含まれており、スーパーユーザ用に選択したパスワードでパスワードを変更した後、復元されたプライマリからすべてのインスタンスを複製する完全なクラスターをブートストラップするためにそれを使用します。詳細については、文書のブートストラップページを参照してください。

| Name                | Description                                                                    | Type                |
|---------------------|--------------------------------------------------------------------------------|---------------------|
| <code>backup</code> | 復元する必要があるバックアップ                                                                | <i>BackupSource</i> |
| <code>source</code> | バックアップを復元する外部クラスター。これは、バックアップが保存されるフォルダーの名前としても使用されるため、ソースクラスターの名前に設定する必要があります | string              |

次のページに続く

表 13 – 前のページからの続き

| Name                  | Description                                                                                                                                                                                                                                                                                                                                                                                       | Type                  |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| <i>recoveryTarget</i> | デフォルトでは、リカバリプロセスはアーカイブ内の利用可能なすべての WAL ファイルを適用します (完全リカバリ)。ただし、PostgreSQL で期待されるようにリカバリオブジェクトを指定することで、一貫性のある状態に到達するか、ポイントインタイム (PITR) に回復する (タイムスタンプ、トランザクション ID、LSN など)。)。詳細: <a href="https://www.postgresql.org/docs/current/runtime-config-wal.html#RUNTIME-CONFIG-WAL-RECOVERY-TARGET">https://www.postgresql.org/docs/current/runtime-config-wal.html#RUNTIME-CONFIG-WAL-RECOVERY-TARGET</a> | <i>RecoveryTarget</i> |

## 13.14 証明書の構成

CertificatesConfiguration には、サーバ証明書をハンドルするために必要な構成が含まれています。

| Name                          | Description                                                                                                                                                                                                                                                                  | Type   |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <code>serverCASecret`</code>  | サーバー CA 証明書を含むシークレット。定義されていない場合、自己署名 CA を使用して新しいシークレットが作成され、TLS 証明書 <code>ServerTLSSecret</code> の生成に使用されます。<br>次を含む: <code>-ca.crt</code> : CA クライアント接続文字列で <code>sslrootcert</code> として使用されるサーバー証明書を検証するために使用する必要があります。<br><code>-ca.key</code> : サーバー SSL 証明書の生成に使用されるキー。 | string |
| <code>serverTLSSecret`</code> | クライアントが postgres に安全に接続できるように、 <code>ssl_cert_file`</code> および <code>ssl_key_file`</code> として設定されるサーバー TLS 証明書とキーを含む <code>kubernetes.io/tls</code> タイプの秘密。定義されていない場合、 <code>ServerCASecret</code> は <code>ca.key`</code> も提供する必要があり、提供された CA を使用して新しいシークレットが作成されます。         | string |
| <i>replicationTLSSecret</i>   | <code>streaming_replica`</code> ユーザとして認証するクライアント証明書を含む <code>kubernetes.io/tls</code> タイプの秘密。定義されていない場合、 <code>ClientCASecret</code> は <code>ca.key`</code> も提供する必要があり、提供された CA を使用して新しいシークレットが作成されます。                                                                       | string |

次のページに続く

表 14 – 前のページからの続き

| Name                            | Description                                                                                                                                                                                                                                                           | Type   |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <code>clientCASecret`</code>    | クライアント CA 証明書を含むシークレット。定義されていない場合、自己署名 CA で新しいシークレットが作成され、すべてのクライアント証明書の生成に使用されます。<br />次を含む：<br /><br />-ca.crt：CA すべてのインスタンスの <code>ssl_ca_file`</code> として使用されるクライアント証明書の検証に使用する必要があります。<br />-ca.key：クライアント証明書の生成に使用されるキー。ReplicationTLSSecret が提供されている場合、これは省略可能 | string |
| <code>serverAltDNSNames`</code> | 必要に応じて、生成されたサーバー TLS 証明書に追加されるサーバーの代替 DNS 名のリスト。                                                                                                                                                                                                                      | string |

## 13.15 証明書ステータス

CertificatesStatus には、構成証明書と関連する有効期限が含まれます。

| Name                      | Description   | Type              |
|---------------------------|---------------|-------------------|
| <code>expirations`</code> | すべての証明書の有効期限。 | map[string]string |

## 13.16 クラスター

Cluster は PostgreSQL API のスキーマです

| Name                   | Description                                                                                                                                                                                                                                                                                    | Type              |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <code>metadata`</code> | NA                                                                                                                                                                                                                                                                                             | metav1.ObjectMeta |
| <code>spec`</code>     | クラスターの望ましい動作の仕様。詳細：<br><a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a>                                                   | ClusterSpec       |
| <code>status`</code>   | クラスターの最近観察されたステータス。このデータは最新ではない可能性があります。システムによって設定されます。読み取り専用。詳細：<br><a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https</a><br>://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status | ClusterStatus     |

## 13.17 ClusterCondition

ClusterCondition は、特定の時点でのクラスターオブジェクトの状態を記述します

| Name                            | Description                        | Type                 |
|---------------------------------|------------------------------------|----------------------|
| <code>`type`</code>             | 条件のタイプ。                            | ClusterConditionType |
| <code>`status`</code>           | 条件のステータス。True、False、Unknown のいずれか。 | ConditionStatus      |
| <code>lastTransitionTime</code> | 条件がある状態から別の状態に最後に遷移した時間。           | metav1.Time          |
| <code>`reason`</code>           | 条件の最後の遷移の理由。                       | string               |
| <code>`message`</code>          | トランジションに関する詳細を示す人間が読めるメッセージ。       | string               |

## 13.18 ClusterList

ClusterList にはクラスターのリストが含まれます

| Name                  | Description                                                                                                                                                                                                                  | Type            |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <code>metadata</code> | 標準リストメタデータ。詳細: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> | metav1.ListMeta |
| <code>`items`</code>  | クラスターのリスト-必須                                                                                                                                                                                                                 | クラスター           |

## 13.19 ClusterSpec

ClusterSpec は、クラスターの望ましい状態を定義します

| Name                             | Description                                                                                                                                                          | Type                   |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <code>`description`</code>       | この PostgreSQL クラスターの説明                                                                                                                                               | string                 |
| <code>`inheritedMetadata`</code> | クラスターに関連するすべてのオブジェクトに継承されるメタデータ                                                                                                                                      | EmbeddedObjectMetadata |
| <code>`imageName`</code>         | コンテナイメージの名前。タグ ( <code>&lt;image&gt;:&lt;tag&gt;</code> ) と確定的で反復可能なデプロイメントのダイジェスト ( <code>&lt;image&gt;:&lt;tag&gt;@sha256:&lt;digestValue&gt;</code> ) の両方をサポートします | string                 |

次のページに続く

表 19 – 前のページからの続き

| Name                               | Description                                                                                                                                                                                                                                                                                          | Type                               |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| <code>`imagePullPolicy`</code>     | イメージプルポリシー。 <i>Always</i> 、 <i>Never</i> 、または <i>`IfNotPresent`</i> のいずれか。定義されていない場合、デフォルトは <i>`IfNotPresent`</i> になります。更新できません。詳細：<br><a href="https://kubernetes.io/docs/concepts/containers/images#updating-images">https</a><br>://kubernetes.io/docs/concepts/containers/images#updating-images | <code>corev1.PullPolicy</code>     |
| <code>`postgresUID`</code>         | イメージ内の <code>`postgres`</code> ユーザの UID、デフォルトは <code>`26`</code>                                                                                                                                                                                                                                     | <code>int64</code>                 |
| <code>`postgresGID`</code>         | イメージ内の <code>`postgres`</code> ユーザの GID、デフォルトは <code>`26`</code>                                                                                                                                                                                                                                     | <code>int64</code>                 |
| <code>`instances`</code>           | クラスターに必要なインスタンスの数-必須                                                                                                                                                                                                                                                                                 | <code>int32</code>                 |
| <code>`minSyncReplicas`</code>     | プライマリとの同期レプリケーションに必要なインスタンスの最小数。未定義または 0 の場合、スタンバイが利用できないときに書き込みを完了できます。                                                                                                                                                                                                                             | <code>int32</code>                 |
| <code>`maxSyncReplicas`</code>     | 同期レプリケーションクォーラムの目標値。準備完了スタンバイの数がこれより少ない場合は、減らすことができます。未定義または 0 は、同期レプリケーションを無効にします。                                                                                                                                                                                                                  | <code>int32</code>                 |
| <code>`postgresql`</code>          | PostgreSQL サーバーの構成                                                                                                                                                                                                                                                                                   | <i>PostgresConfiguration</i>       |
| <code>`bootstrap`</code>           | このクラスターをブートストラップする手順                                                                                                                                                                                                                                                                                 | <i>BootstrapConfiguration</i>      |
| <code>`replica`</code>             | レプリカクラスタ構成                                                                                                                                                                                                                                                                                           | <i>ReplicaClusterConfiguration</i> |
| <code>`superuserSecret`</code>     | スーパーユーザのパスワードを含むシークレット。定義されていない場合、ランダムに生成されたパスワードで新しいシークレットが作成されます                                                                                                                                                                                                                                   | <i>LocalObjectReference</i>        |
| <code>enableSuperuserAccess</code> | このオプションを有効にすると、演算子は <code>`SuperuserSecret`</code> を使用して <code>`postgres`</code> ユーザパスワードを更新します（シークレットが存在しない場合、演算子は自動的にパスワードを作成します）。このオプションを無効にすると、演算子は <code>`SuperuserSecret`</code> コンテンツを無視し、自動作成時に削除し、 <code>`NULL`</code> に設定して <code>`postgres`</code> ユーザのパスワードをブランクにします。デフォルトで有効。           | <code>*bool</code>                 |
| <code>`certificates`</code>        | CA および関連証明書の構成                                                                                                                                                                                                                                                                                       | <b>証明書の構成</b>                      |
| <code>`imagePullSecrets`</code>    | 画像のプルに使用されるプルシークレットのリスト。ライセンスキーにプルシークレットが含まれている場合、そのシークレットは自動的に含まれます。                                                                                                                                                                                                                                | <i>LocalObjectReference</i>        |
| <code>`storage`</code>             | インスタンスのストレージの構成                                                                                                                                                                                                                                                                                      | <i>StorageConfiguration</i>        |

次のページに続く

表 19 – 前のページからの続き

| Name                               | Description                                                                                                                                                                                                         | Type                                     |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| <code>startDelay`</code>           | PostgreSQL インスタンスが正常にスタートするために許可される秒単位の時間（デフォルト 30）                                                                                                                                                                 | int32                                    |
| <code>stopDelay`</code>            | PostgreSQL インスタンスが正常にシャットダウンできる秒数（デフォルトは 30）                                                                                                                                                                        | int32                                    |
| <code>switchoverDelay`</code>      | スイッチオーバー中にプライマリ PostgreSQL インスタンスが正常にシャットダウンするのに許可される秒単位の時間。デフォルトより大きいのは 40000000 で、1 年は秒単位で、無限の遅延をシミュレートするのに十分な大きさです                                                                                               | int32                                    |
| <code>affinity`</code>             | ポッドのアフィニティ/アンチアフィニティルール                                                                                                                                                                                             | <i>AffinityConfiguration</i>             |
| <code>resources`</code>            | 生成されたすべてのポッドのリソース要件。詳細については、 <a href="https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/">https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</a> を参照してください。 | <code>corev1.ResourceRequirements</code> |
| <code>primaryUpdateStrategy</code> | すべてのレプリカが正常に更新された後、ローリング更新プロシージャ中にプライマリーサーバをアップグレードするために従う戦略：自動化（ <i>unsupervised</i> -デフォルト）またはマニュアル（ <i>supervised</i> ）が可能                                                                                       | PrimaryUpdateStrategy                    |
| <code>primaryUpdateMethod`</code>  | すべてのレプリカが正常に更新された後、ローリング更新プロシージャ中にプライマリーサーバをアップグレードする方法：スイッチオーバー（ <i>switchover</i> -デフォルト）またはインプレース（ <i>restart</i> ）で可能                                                                                           | PrimaryUpdateMethod                      |
| <code>backup`</code>               | バックアップに使用される構成                                                                                                                                                                                                      | <i>BackupConfiguration</i>               |
| <code>nodeMaintenanceWindow</code> | Kubernetes ノードのメンテナンスウィンドウを定義する                                                                                                                                                                                     | <i>NodeMaintenanceWindow</i>             |
| <code>licenseKey`</code>           | クラスターのライセンスキー。空の場合、クラスターは試用モードで動作し、有効期限（デフォルトは 30 日）が経過すると、演算子は調整の試みを中止します。詳細については、演算子に付属のライセンス契約を参照してください。                                                                                                         | string                                   |
| <code>licenseKeySecret`</code>     | ライセンスキーへのリファレンス。これを設定すると、LicenseKey よりも優先されます。                                                                                                                                                                      | <code>corev1.SecretKeySelector</code>    |
| <code>monitoring`</code>           | このクラスターのモニタリングインフラストラクチャの構成                                                                                                                                                                                         | <i>MonitoringConfiguration</i>           |
| <code>externalClusters`</code>     | 構成で使用される外部クラスターのリスト                                                                                                                                                                                                 | <i>ExternalCluster</i>                   |
| <code>logLevel`</code>             | インスタンスのログレベル、次の値のいずれか：エラー、ワーニング、情報（デフォルト）、デバッグ、トレース                                                                                                                                                                 | string                                   |

## 13.20 ClusterStatus

ClusterStatus は、クラスターの観測状態を定義します

| Name                                           | Description                                                                                 | Type                            |
|------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------|
| <code>`instances`</code>                       | クラスター内のインスタンスの総数                                                                            | int32                           |
| <code>`readyInstances`</code>                  | クラスター内の準備完了インスタンスの総数                                                                        | int32                           |
| <code>`instancesStatus`</code>                 | インスタンスのステータス                                                                                | map[utils.PodStatus]string      |
| <code>`latestGeneratedNode`</code>             | 生成された最新のノードの ID（ノード名前の衝突を避けるために使用）                                                          | int32                           |
| <code>`currentPrimary`</code>                  | 現在のプライマリインスタンス                                                                              | string                          |
| <code>`targetPrimary`</code>                   | ターゲットプライマリインスタンス。これは、スイッチオーバーまたはフェイルオーバー中の以前のインスタンスとは異なります。                                 | string                          |
| <code>`pvcCount`</code>                        | このクラスターによって作成された PVC の数                                                                     | int32                           |
| <code>`jobCount`</code>                        | このクラスターによって作成されたジョブの数                                                                       | int32                           |
| <code>`danglingPVC`</code>                     | このクラスターによって作成され、まだ使用可能なポッドに接続されていないすべての PVC のリスト                                            | string                          |
| <code>`resizingPVC`</code>                     | ResizingPVC 状態のすべての PVC のリスト。                                                               | string                          |
| <code>`initializingPVC`</code>                 | このクラスターによって初期化されているすべての PVC のリスト                                                            | string                          |
| <code>`healthyPVC`</code>                      | ぶら下がりも初期化もされていないすべての PVC のリスト                                                               | string                          |
| <code>`licenseStatus`</code>                   | ライセンスの状態                                                                                    | licensekey.Status               |
| <code>`writeService`</code>                    | 現在の書き込みポッド                                                                                  | string                          |
| <code>`readService`</code>                     | 読み取りポッドの現在のリスト                                                                              | string                          |
| <code>`phase`</code>                           | クラスターの現在のフェーズ                                                                               | string                          |
| <code>`phaseReason`</code>                     | 現在のフェーズの理由                                                                                  | string                          |
| <code>`secretsResourceVersion`</code>          | 演算子によって管理される秘密のリソースバージョンのリスト。ここでのすべての変更は、秘密データをリフレッシュするインスタスマネージャの利益のために行われます               | <i>SecretsResourceVersion</i>   |
| <code>`configMapResourceVersion`</code>        | 演算子が管理する configmap のリソースバージョンのリスト。ここでのすべての変更は、configmap データをリフレッシュするインスタスマネージャの利益のために行われます。 | <i>ConfigMapResourceVersion</i> |
| <code>`certificates`</code>                    | デフォルトで初期化された CA および関連証明書の構成。                                                                | <b>証明書ステータス</b>                 |
| <code>`firstRecoverabilityPoint`</code>        | RFC3339 フォーマットの日付として保存された最初の回復可能ポイント                                                        | string                          |
| <code>`cloudNativePostgresqlCommitHash`</code> | この演算子が実行しているコミットハッシュ番号                                                                      | string                          |

次のページに続く

表 20 – 前のページからの続き

| Name                                         | Description                                                         | Type                      |
|----------------------------------------------|---------------------------------------------------------------------|---------------------------|
| <code>`currentPrimary-Timestamp`</code>      | プライマリへの最後の実際のプロモーションが発生したときのタイムスタンプ                                 | string                    |
| <code>`targetPrimaryTimes-tamp`</code>       | 新しいプライマリの最後のリクエストが発生したときのタイムスタンプ                                    | string                    |
| <code>`poolerIntegrations`</code>            | クラスターを参照するプーラーが必要とする統合                                              | <i>PoolerIntegrations</i> |
| <code>cloudNativePostgresOperatorHash</code> | 演算子のバイナリのハッシュ                                                       | string                    |
| <code>`onlineUpdateEnabled`</code>           | OnlineUpdateEnabled は、クラスター内でオンラインアップグレードが有効になっているかどうかを示します         | bool                      |
| <code>`azurePVCUpdateEnabled`</code>         | AzurePVCUpdateEnabled は、このクラスターで PVC オンラインアップグレードが有効になっているかどうかを示します | bool                      |
| <code>`conditions`</code>                    | クラスターオブジェクトの条件                                                      | <i>ClusterCondition</i>   |

## 13.21 ConfigMapKeySelector

ConfigMapKeySelector には、ConfigMap のキーを見つけるのに十分な情報が含まれています

| Name             | Description | Type   |
|------------------|-------------|--------|
| <code>key</code> | 選択するキー-必須   | string |

## 13.22 ConfigMapResourceVersion

ConfigMapResourceVersion は、演算子が管理するシークレットのリソースバージョンです

| Name                 | Description                                                    | Type              |
|----------------------|----------------------------------------------------------------|-------------------|
| <code>metrics</code> | メトリックスを渡すために使用されるすべての構成マップのバージョンを含むマップ。マップキーは構成マップ名、マップ値はバージョン | map[string]string |

## 13.23 DataBackupConfiguration

DataBackupConfiguration は、データディレクトリのバックアップの構成です。

| Name                             | Description                                                                                                                                                                                                          | Type                |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <code>`compression`</code>       | オブジェクトストアにストリーミングしながら、バックアップファイル（テーブルスペースごとの tar ファイル）を圧縮します。使用可能オプションは、空の文字列（圧縮なし、デフォルト）、 <code>gzip</code> 、 <code>`bzip2`</code> または <code>`snappy`</code> です。                                                    | CompressionType     |
| <code>`encryption`</code>        | ファイルの暗号化を強制するとき（バケットがそのためにまだ構成されていない場合）。許可されるオプションは空の文字列（バケットポリシー、デフォルトを使用）、 <code>`AES256`</code> および <code>`aws:kms`</code> です。                                                                                    | EncryptionType      |
| <code>immediateCheckpoint</code> | PostgreSQL サーバーの <code>`checkpoint_completion_target`</code> 設定に従って、バックアップ初期チェックポイントの I/O ワークロードを制限するかどうかを制御します。true に設定すると、即時チェックポイントが使用されます。つまり、PostgreSQL はチェックポイントをできるだけ早く完了します。デフォルトでは <code>`false`</code> です。 | bool                |
| <code>`jobs`</code>              | バックアップのアップロードに使用される並列ジョブの数、デフォルトは 2                                                                                                                                                                                  | <code>*int32</code> |

## 13.24 EPASConfiguration

EPASConfiguration には EDB Postgres Advanced Server 固有の構成が含まれます

| Name               | Description                     | Type |
|--------------------|---------------------------------|------|
| <code>audit</code> | true の場合、edb_audit ロギングが有効になります | bool |

## 13.25 EmbeddedObjectMetadata

EmbeddedObjectMetadata には、クラスターに関連するすべてのリソースに継承されるメタデータが含まれます

| Name                     | Description | Type              |
|--------------------------|-------------|-------------------|
| <code>labels`</code>     | NA          | map[string]string |
| <code>annotations</code> | NA          | map[string]string |

## 13.26 ExternalCluster

ExternalCluster は、構成の他のセクションで使用される外部クラスターへの接続パラメーターを表します

| Name                              | Description                             | Type                                  |
|-----------------------------------|-----------------------------------------|---------------------------------------|
| <code>name`</code>                | サーバー名前、必須-必須                            | string                                |
| <code>connectionParameters</code> | dbname、ホスト、username などの接続パラメータのリスト      | map[string]string                     |
| <code>sslCert`</code>             | このインスタンスへの接続に使用される SSL 証明書へのリファレンス      | corev1.SecretKeySelector              |
| <code>sslKey`</code>              | このインスタンスへの接続に使用される SSL プライベートキーへのリファレンス | corev1.SecretKeySelector              |
| <code>sslRootCert`</code>         | このインスタンスへの接続に使用される SSL CA 公開キーへのリファレンス  | corev1.SecretKeySelector              |
| <code>password`</code>            | サーバーへの接続に使用されるパスワードへのリファレンス             | corev1.SecretKeySelector              |
| <code>barmanObjectStore`</code>   | barman-cloud ツールスイートの構成                 | <i>BarmanObjectStoreConfiguration</i> |

## 13.27 GoogleCredentials

GoogleCredentials は、Google Cloud Storage の資格情報のタイプです。GKE 環境内で実行する場合でも、これを指定ニーズがあります。

| Name                         | Description                                            | Type |
|------------------------------|--------------------------------------------------------|------|
| <code>gkeEnvironment`</code> | true に設定すると、GKE 環境内で実行されていると想定されます。デフォルトは false です。-必須 | bool |

次のページに続く

表 27 – 前のページからの続き

| Name                                       | Description                                         | Type                     |
|--------------------------------------------|-----------------------------------------------------|--------------------------|
| <i>applicationCredent-</i><br><i>tials</i> | 資格情報を持つ Google Cloud Storage JSON ファ<br>イルを含むシークレット | <i>SecretKeySelector</i> |

## 13.28 InstanceID

InstanceID には、インスタンスを識別する情報が含まれます

| Name               | Description | Type   |
|--------------------|-------------|--------|
| <i>podName`</i>    | ポッド名前       | string |
| <i>ContainerID</i> | コンテナ ID     | string |

## 13.29 LDAPBindAsAuth

LDAPBindAsAuth は、LDAP のバインド認証を使用するために必要なフィールドを提供します

| Name          | Description         | Type   |
|---------------|---------------------|--------|
| <i>prefix</i> | バインド認証オプションのプレフィックス | string |
| <i>suffix</i> | バインド認証オプションのサフィックス  | string |

## 13.30 LDAPBindSearchAuth

LDAPBindSearchAuth は、バインド+検索 LDAP 認証プロセスを使用するために必要なフィールドを提供します

| Name                   | Description                      | Type                            |
|------------------------|----------------------------------|---------------------------------|
| <i>baseDN`</i>         | ユーザ検索を開始するルート DN                 | string                          |
| <i>bindDN`</i>         | ディレクトリにバインドするユーザの DN             | string                          |
| <i>bindPassword`</i>   | ユーザがディレクトリにバインドするためのパ<br>スワードの秘密 | <i>corev1.SecretKeySelector</i> |
| <i>searchAttribute</i> | ユーザー名とマッチする属性                    | string                          |
| <i>searchFilter`</i>   | 検索+バインド認証を行うときに使用する検索<br>フィルター   | string                          |

## 13.31 LDAPConfig

LDAPConfig には、LDAP 認証に必要なパラメーターが含まれています

| Name                        | Description                                                           | Type                               |
|-----------------------------|-----------------------------------------------------------------------|------------------------------------|
| <code>`server`</code>       | LDAP ホスト名または IP アドレス                                                  | string                             |
| <code>`port`</code>         | LDAP サーバーのポート                                                         | int                                |
| <code>`scheme`</code>       | 使用する LDAP スキーマ、可能なオプションは <code>`ldap`</code> および <code>`ldaps`</code> | LDAPScheme                         |
| <code>`tls`</code>          | LDAP over TLS を有効にするには 1 に設定します                                       | bool                               |
| <code>`bindAsAuth`</code>   | 認証設定としてバインド                                                           | <a href="#">LDAPBindAsAuth</a>     |
| <code>bindSearchAuth</code> | バインド+検索認証の構成                                                          | <a href="#">LDAPBindSearchAuth</a> |

## 13.32 LocalObjectReference

LocalObjectReference には、同じ名前空間内で既知のタイプのローカルオブジェクトを見つけるのに十分な情報が含まれています

| Name              | Description         | Type   |
|-------------------|---------------------|--------|
| <code>name</code> | 指示対象の名前。 <b>-必須</b> | string |

## 13.33 MonitoringConfiguration

MonitoringConfiguration は、特定のクラスターのすべてのモニタリング構成を含むタイプです

| Name                                  | Description                                                                            | Type                                  |
|---------------------------------------|----------------------------------------------------------------------------------------|---------------------------------------|
| <code>`disableDefault-Queries`</code> | デフォルトのクエリを挿入するかどうか。デフォルトのクエリをクラスターに注入したくない場合は、 <code>`true`</code> に設定します。デフォルト：false。 | <b>*bool</b>                          |
| <code>customQueriesConfigMap</code>   | カスタムクエリを含む構成マップのリスト                                                                    | <a href="#">ConfigMapKeySelector</a>  |
| <code>`customQueriesSecret`</code>    | カスタムクエリを含むシークレットのリスト                                                                   | SecretKeySelector[#SecretKeySelector] |
| <code>`enablePodMonitor`</code>       | <code>`PodMonitor`</code> を有効または無効にします                                                 | bool                                  |

## 13.34 NodeMaintenanceWindow

NodeMaintenanceWindow には、基になるノードのアップグレード処理中に演算子が使用する情報が含まれています。

このオプションは、選択したストレージにより、ポッドがノード間で自由に移動できない場合にのみ役立ちます。

| Name              | Description                                                              | Type         |
|-------------------|--------------------------------------------------------------------------|--------------|
| <i>inProgress</i> | 進行中のノードメンテナンスアクティビティはありますか？ -必須                                          | bool         |
| <i>reusePVC`</i>  | 既存の PVC を再使用する（ノードが再び起動するのを待つ）または使用しない（別の場所で再作成する -instances> 1 の場合） -必須 | <i>*bool</i> |

## 13.35 PgBouncerIntegrationStatus

PgBouncerIntegrationStatus は、クラスターを参照する pgbouncer プーラーに必要な統合をカプセル化します

| Name           | Description | Type   |
|----------------|-------------|--------|
| <i>secrets</i> | NA          | string |

## 13.36 PgBouncerSecrets

PgBouncerSecrets には、pgbouncer によって使用されるシークレットのバージョンが含まれています

| Name             | Description       | Type                 |
|------------------|-------------------|----------------------|
| <i>authQuery</i> | 認証クエリーシークレットバージョン | <i>SecretVersion</i> |

## 13.37 PgBouncerSpec

PgBouncerSpec は、PgBouncer の構成方法を定義します

| Name             | Description | Type              |
|------------------|-------------|-------------------|
| <i>poolMode`</i> | プールモード-必須   | PgBouncerPoolMode |

[次のページに続く](#)

表 37 – 前のページからの続き

| Name                         | Description                                                                                                                                                                                                                         | Type                        |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| <code>authQuerySecret</code> | 認証クエリーに使用する必要があるユーザの資格情報。指定されている場合は、AuthQuery (「SELECT username、passwd FROM pg_shadow WHERE username = \$1」) も指定する必要があり、自動 CNP クラスター統合はトリガーされません。                                                                                   | <i>LocalObjectReference</i> |
| <code>`authQuery`</code>     | 特定のユーザのパスワードのハッシュをダウンロードするために使用されるクエリー。デフォルト:「SELECT username、passwd FROM user_search (\$1)」。                                                                                                                                       | string                      |
| <code>`parameters`</code>    | PgBouncer に渡される追加のパラメータ-設定可能なオプションのリストについては CNP の文書を確認してください                                                                                                                                                                        | map[string]string           |
| <code>`paused`</code>        | <code>`true`</code> に設定すると、PgBouncer は PostgreSQL サーバーから切断し、まずすべてのクエリが完了するまで待機し、この値が <code>`false`</code> (デフォルト) に設定されるまですべての新しいクライアント接続を一時停止します。内部的に、演算子は PgBouncer の <code>`PAUSE`</code> および <code>`RESUME`</code> コマンドを呼び出します。 | bool                        |

## 13.38 PodMeta

PodMeta は、metav1.ObjectMeta に似た構造ですが、まだユーザに適した CRD を作成するために controller-gen によって解析可能です。PodTemplateSpec のコメントには、コアデータ型を使用しない理由が説明されています。

| Name                  | Description                                                                                                                                                                                 | Type              |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <code>`labels`</code> | オブジェクトの整理と分類 (スコープと選択) に使用できる文字列キーと値のマップ。レプリケーションコントローラーおよびサービスのセクターとマッチする場合があります。詳細: <a href="http://kubernetes.io/docs/user-guide/labels">http://kubernetes.io/docs/user-guide/labels</a> | map[string]string |

次のページに続く

表 38 – 前のページからの続き

| Name               | Description                                                                                                                                                                                                                                                                                          | Type              |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <i>annotations</i> | 注釈は、任意のメタデータを保存および取得するために外部ツールで設定できるリソースとともに保存される非構造化キー値マップです。これらはクエリ可能ではなく、オブジェクトを変更ときに保存する必要があります。詳細： <a href="http://kubernetes.io/docs/user-guide/annotations">http:  &lt;a href="http://kubernetes.io/docs/user-guide/annotations"&gt;//kubernetes.io/docs/user-guide/annotations&lt;/a&gt;</a> | map[string]string |

## 13.39 PodTemplateSpec

PodTemplateSpec は、ユーザがポッド生成用のテンプレートを設定できる構造です。

残念ながら、生成された CRD にはメタデータセクションのフィールドがないため、corev1.PodTemplateSpec タイプを使用できません。

参 照：<https://github.com/kubernetes-sigs/controller-tools/issues/385>      <https://github.com/kubernetes-sigs/controller-tools/issues/448> <https://github.com/prometheus-operator/prometheus-operator/issues/3041>

| Name            | Description                                                                                                                                                                                                                                                                                                                                                                                                                     | Type           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <i>metadata</i> | 標準オブジェクトのメタデータ。詳細： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata</a> <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#メタデータ</a> | <i>PodMeta</i> |
| <i>spec</i>     | ポッドの望ましい動作の仕様。詳細： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a>                                                                                                                                                                                         | corev1.PodSpec |

## 13.40 プーラー

プーラーは、プーラー API のスキーマです

| Name            | Description | Type                |
|-----------------|-------------|---------------------|
| <i>metadata</i> | NA          | metav1.ObjectMeta   |
| <i>spec</i>     | NA          | <i>PoolerSpec</i>   |
| <i>status</i>   | NA          | <i>PoolerStatus</i> |

## 13.41 PoolerIntegrations

PoolerIntegrations は、クラスターを参照するプーラーに必要な統合をカプセル化します

| Name                        | Description | Type                              |
|-----------------------------|-------------|-----------------------------------|
| <i>pgBouncerIntegration</i> | NA          | <i>PgBouncerIntegrationStatus</i> |

## 13.42 PoolerList

PoolerList には、Pooler のリストが含まれています

| Name            | Description | Type                   |
|-----------------|-------------|------------------------|
| <i>metadata</i> | NA          | <i>metav1.ListMeta</i> |
| <i>items</i>    | -必須         | プーラー                   |

## 13.43 PoolerSecrets

PoolerSecrets には、使用されるすべてのシークレットのバージョンが含まれます

| Name                    | Description                 | Type                    |
|-------------------------|-----------------------------|-------------------------|
| <i>serverTLS</i>        | サーバー TLS シークレットバージョン        | <i>SecretVersion</i>    |
| <i>serverCA</i>         | サーバー CA シークレットバージョン         | <i>SecretVersion</i>    |
| <i>clientCA</i>         | クライアント CA シークレットバージョン       | <i>SecretVersion</i>    |
| <i>pgBouncerSecrets</i> | PgBouncer が使用するシークレットのバージョン | <i>PgBouncerSecrets</i> |

## 13.44 PoolerSpec

PoolerSpec は、プーラーの望ましい状態を定義します

| Name             | Description                                                               | Type                        |
|------------------|---------------------------------------------------------------------------|-----------------------------|
| <i>cluster</i>   | これは、プーラーが動作するクラスターリファレンスです。プーラー名前は、同じ名前空間内のどのクラスター名前とも決してマッチしてはなりません。 -必須 | <i>LocalObjectReference</i> |
| <i>type</i>      | どのインスタンスにトラフィックを転送する必要がありますか？ -必須                                         | PoolerType                  |
| <i>instances</i> | 必要なレプリカの数-必須                                                              | int32                       |
| <i>template</i>  | 作成するポッドのテンプレート                                                            | <i>PodTemplateSpec</i>      |

次のページに続く

表 44 – 前のページからの続き

| Name             | Description     | Type                 |
|------------------|-----------------|----------------------|
| <i>pgbouncer</i> | PgBouncer 設定-必須 | <i>PgBouncerSpec</i> |

## 13.45 PoolerStatus

PoolerStatus は、プーラーの監視状態を定義します

| Name                   | Description         | Type                 |
|------------------------|---------------------|----------------------|
| <code>`secrets`</code> | 構成オブジェクトのリソースバージョン  | <i>PoolerSecrets</i> |
| <i>instances</i>       | スケジュールしようとしているポッドの数 | int32                |

## 13.46 PostgresConfiguration

PostgresConfiguration は PostgreSQL の構成を定義します

| Name                            | Description                                                                                       | Type                     |
|---------------------------------|---------------------------------------------------------------------------------------------------|--------------------------|
| <code>`parameters`</code>       | PostgreSQL 設定オプション (postgresql.conf)                                                              | map[string]string        |
| <code>`pg_hba`</code>           | PostgreSQL ホストベース認証ルール (pg_hba.conf ファイルに追加される行)                                                  | string                   |
| <code>`epas`</code>             | EDB Postgres Advanced Server 固有の構成                                                                | <i>EPASConfiguration</i> |
| <code>`promotionTimeout`</code> | インスタンスをプライマリに昇格させるときに待機する最大秒数を指定します。デフォルトより大きいのは 40000000 で、1 年は秒単位で、無限のタイムアウトをシミュレートするのに十分な大きさです | int32                    |
| <i>shared_preload_libraries</i> | デフォルトのライブラリに追加する共有プリロードライブラリのリスト                                                                  | string                   |
| <code>`ldap`</code>             | LDAP 構成を指定するオプション                                                                                 | <i>LDAPConfig</i>        |

## 13.47 RecoveryTarget

RecoveryTarget を使用すると、リカバリプロセスが停止する瞬間を構成できます。TargetTLI を除くすべてのターゲットオプションは相互に排他的です。

| Name                     | Description               | Type   |
|--------------------------|---------------------------|--------|
| <code>`targetTLI`</code> | ターゲットタイムライン (「最新」または正の整数) | string |
| <code>`targetXID`</code> | ターゲットトランザクション ID          | string |

次のページに続く

表 47 – 前のページからの続き

| Name                         | Description                                                | Type          |
|------------------------------|------------------------------------------------------------|---------------|
| <code>`targetName`</code>    | ターゲット名前（ <code>`pg_create_restore_point`</code> で以前に作成される） | string        |
| <code>`targetLSN`</code>     | ターゲット LSN（ログシーケンス番号）                                       | string        |
| <code>`targetTime`</code>    | PostgreSQL で許可されている明確な表現形式でのターゲット時間                        | string        |
| <code>targetImmediate</code> | 一貫した状態に達するとすぐにリカバリを終了します                                   | <b>*</b> bool |
| <code>`exclusive`</code>     | ターゲットを排他的に設定します（デフォルトは true）                               | <b>*</b> bool |

## 13.48 ReplicaClusterConfiguration

ReplicaClusterConfiguration は、レプリカクラスターの構成をカプセル化します

| Name                  | Description                                                                                                                                                    | Type   |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| <code>enabled</code>  | レプリカモードが有効な場合、このクラスターは既存のクラスターのレプリカになります。レプリカクラスターは、リカバリオブジェクトストアから、または <code>pg_basebackup</code> を介したストリーミングを介して作成できます。詳細については、文書の複製ページを参照してください。 <b>-必須</b> | bool   |
| <code>`source`</code> | レプリケーションのオリジン- <b>必須</b>                                                                                                                                       | string |

## 13.49 RollingUpdateStatus

RollingUpdateStatus には、更新中のインスタンスに関する情報が含まれます

| Name                   | Description            | Type        |
|------------------------|------------------------|-------------|
| <code>imageName</code> | ポッドに入れたイメージ- <b>必須</b> | string      |
| <code>startedAt</code> | 更新が開始されたとき             | metav1.Time |

## 13.50 S3Credentials

S3Credentials は、S3 にファイルをアップロードするために使用される資格情報のタイプです。次の 2 つの方法で提供できます。

- 明示的に `accessKeyId` と `secretAccessKey` を渡す
- `inheritFromIAMRole` を `true` に設定して、ポッド環境からロールを継承する

| Name                            | Description                      | Type                     |
|---------------------------------|----------------------------------|--------------------------|
| <code>`accessKeyId`</code>      | アクセスキー ID へのリファレンス               | <i>SecretKeySelector</i> |
| <code>`secretAccessKey`</code>  | シークレットアクセスキーへのリファレンス             | <i>SecretKeySelector</i> |
| <code>`sessionToken`</code>     | セッションキーへの参照                      | <i>SecretKeySelector</i> |
| <code>inheritFromIAMRole</code> | キーを明示的に提供せずに、ロールベースの認証を使用します。-必須 | bool                     |

## 13.51 ScheduledBackup

ScheduledBackup は、scheduledbackups API のスキーマです

| Name                  | Description                                                                                                                                                                                                                                                                                      | Type                         |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| <code>metadata</code> | NA                                                                                                                                                                                                                                                                                               | <i>metav1.ObjectMeta</i>     |
| <code>`spec`</code>   | ScheduledBackup の望ましい動作の仕様。詳細： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a>                                             | <i>ScheduledBackupSpec</i>   |
| <code>`status`</code> | ScheduledBackup の最新の監視ステータス。このデータは最新ではない可能性があります。システムによって設定されます。読み取り専用。詳細： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#spec-and-status</a> | <i>ScheduledBackupStatus</i> |

## 13.52 ScheduledBackupList

ScheduledBackupList には、ScheduledBackup のリストが含まれています

| Name                  | Description                                                                                                                                                                                                                  | Type                   |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <code>metadata</code> | 標準リストメタデータ。詳細： <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a> | <i>metav1.ListMeta</i> |
| <code>`items`</code>  | クラスターのリスト-必須                                                                                                                                                                                                                 | <i>ScheduledBackup</i> |

## 13.53 ScheduledBackupSpec

ScheduledBackupSpec は、ScheduledBackup の望ましい状態を定義します

| Name                   | Description                                                                                                                                                                                                                     | Type                        |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| <code>suspend</code>   | このバックアップが中断されているかどうか                                                                                                                                                                                                            | <i>*bool</i>                |
| <code>immediate</code> | 最初のバックアップを作成後すぐにスタートする必要があるかどうか                                                                                                                                                                                                 | <i>*bool</i>                |
| <code>schedule</code>  | スケジュールは Kubernetes CronJobs で使用されているのと同じフォーマットに従います<br><a href="https://pkg.go.dev/github.com/robfig/cron#hdr-CRON_Expression_Format">. https :</a> //pkg.go.dev/github.com/robfig/cron#hdr-CRON_Expression_Format を参照してください-必須 | string                      |
| <code>cluster</code>   | バックアップするクラスター                                                                                                                                                                                                                   | <i>LocalObjectReference</i> |

## 13.54 ScheduledBackupStatus

ScheduledBackupStatus は、ScheduledBackup の監視状態を定義します

| Name                          | Description                    | Type               |
|-------------------------------|--------------------------------|--------------------|
| <code>lastCheckTime</code>    | スケジュールの最新時刻                    | <i>metav1.Time</i> |
| <code>lastScheduleTime</code> | バックアップが正常にスケジュールされた最後の時間である情報。 | <i>metav1.Time</i> |
| <code>nextScheduleTime</code> | 次回バックアップを実行します                 | <i>metav1.Time</i> |

## 13.55 SecretKeySelector

SecretKeySelector には、秘密のキーを見つけるのに十分な情報が含まれています

| Name             | Description | Type   |
|------------------|-------------|--------|
| <code>key</code> | 選択するキー-必須   | string |

## 13.56 SecretVersion

SecretVersion にはシークレット名前とその ResourceVersion が含まれます

| Name                 | Description             | Type   |
|----------------------|-------------------------|--------|
| <code>`name`</code>  | 秘密の名前                   | string |
| <code>version</code> | シークレットの ResourceVersion | string |

## 13.57 SecretsResourceVersion

SecretsResourceVersion は、演算子が管理するシークレットのリソースバージョンです

| Name                                  | Description                                                      | Type              |
|---------------------------------------|------------------------------------------------------------------|-------------------|
| <code>`superuserSecretVersion`</code> | 「postgres」 ユーザシークレットのリソースバージョン                                   | string            |
| <code>replicationSecretVersion</code> | 「streaming_replica」 ユーザシークレットのリソースバージョン                          | string            |
| <code>applicationSecretVersion</code> | 「アプリ」 ユーザシークレットのリソースバージョン                                        | string            |
| <code>`caSecretVersion`</code>        | 未使用。古いバージョンとの互換性のために保持されます。                                      | string            |
| <code>`clientCaSecretVersion`</code>  | PostgreSQL クライアント側 CA シークレットバージョンのリソースバージョン                      | string            |
| <code>`serverCaSecretVersion`</code>  | PostgreSQL サーバサイドの CA シークレットバージョンのリソースバージョン                      | string            |
| <code>`serverSecretVersion`</code>    | PostgreSQL サーバサイドのシークレットバージョンのリソースバージョン                          | string            |
| <code>`barmanEndpointCA`</code>       | Barman エンドポイント CA のリソースバージョン（提供されている場合）                          | string            |
| <code>`metrics`</code>                | メトリックスを渡すために使用されるすべてのシークレットのバージョンを含むマップ。マップキーはシークレット名、マップ値はバージョン | map[string]string |

## 13.58 StorageConfiguration

StorageConfiguration は、PostgreSQL インスタンスのストレージの構成です

| Name                            | Description                                                                                                                 | Type                                          |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| <code>storageClass</code>       | データベースデータに使用する StorageClass ( <i>PG-DATA</i> )。使用可能な場合、PVC テンプレートを評価した後に適用されます。指定しない場合、生成された PVC はデフォルトのストレージクラスによって満たされます。 | string                                        |
| <code>size</code>               | ストレージのサイズ。PVC テンプレートでまだ指定されていない場合は必須です。このフィールドへの変更は、作成された PVC に自動的に再適用されます。サイズを小さくすることはできません。-必須                            | string                                        |
| <code>resizeInUseVolumes</code> | 既存の PVC のサイズを変更、デフォルトは true                                                                                                 | bool                                          |
| <code>pvcTemplate</code>        | 永続ボリューム要求の生成に使用されるテンプレート                                                                                                    | <code>corev1.PersistentVolumeClaimSpec</code> |

## 13.59 WalBackupConfiguration

WalBackupConfiguration は、WAL ストリームのバックアップの構成です。

| Name                     | Description                                                                                                                      | Type            |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <code>compression</code> | オブジェクトストアに送信する前に、WAL ファイルを圧縮します。使用可能オプションは、空の文字列（圧縮なし、デフォルト）、 <code>gzip</code> 、 <code>bzip2</code> または <code>snappy</code> です。 | CompressionType |
| <code>encryption</code>  | ファイルの暗号化を強制するとき（バケットがそのためにまだ構成されていない場合）。許可されるオプションは空の文字列（バケットポリシー、デフォルトを使用）、 <code>AES256</code> および <code>aws:kms</code> です。    | EncryptionType  |

次のページに続く

表 59 – 前のページからの続き

| Name               | Description                                                                                                                                                                                           | Type |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| <i>maxParallel</i> | 並行してアーカイブする（PostgreSQL インスタンスがバックアップオブジェクトストアにアーカイビングしている場合）か、並行して復元する（PostgreSQL スタンバイがリカバリオブジェクトストアから WAL ファイルをフェッチしている場合）WAL ファイルの数。指定しない場合、WAL ファイルは 1 つずつ処理されます。値として正の整数を受け入れます-1 が受け入れられる最小値です。 | int  |