



PGECons
PostgreSQL Enterprise Consortium

2019年度WG3活動成果報告 クラウド検証編 ①机上調査編

PostgreSQLエンタープライズコンソーシアム
WG3(課題検討WG)

責任範囲

- 本資料は、PGECconsが独自に検証した結果であり、結果はPGECconsの責任の元、公開しています。

概要

■ 目的

- 2017年頃より、パブリッククラウドの企業利用が促進されつつあり、PGEConsセミナー等のアンケートでも、調査してほしいという要望が増えている。そのようなニーズにこたえるため、2018年、2019年とクラウド調査を実施した。
- 本編では、PaaS利用のシェアが特に高いAWSのマネージドサービスであるAmazon RDSについて、机上調査を行った結果を整理して報告する。なおクラウドサービスは、頻繁に更新されるため、最新情報はマニュアル等で確認していただきたい。

■ 調査タイミング

- 今回の机上調査は、2020年1月に実施した。

AWSにおけるPostgreSQL PaaS

- 2020年1月時点、AWSにおけるPaaSであるAWS Relational Database ServiceのPostgreSQL系エンジンは、2種類のデータベースエンジンが利用可能
 - Amazon RDS for PostgreSQL(以下、RDS)
 - ベーシックなマネージドRDBサービス
 - Amazon Aurora with PostgreSQL compatibility(以下、Aurora)
 - クラウド向けに再設計されたPostgreSQLと互換性のあるマネージドRDBサービス
- 机上調査では、マネージドRDBサービスのRDS、Auroraの2種類とオンプレ環境に自力で構築した場合とで、IPAから公開されている「非機能要求グレード」の項目をベースに比較調査を実施した。

机上調査結果

可用性

■ 継続性

項目	比較の観点	オンプレ	RDS	Aurora
最新の復元地点	復元可能な直近の時間は何分前か	PITRで実現する場合、最新のWALアーカイブの取得タイミングに依存する	5分毎にWALアーカイブをS3に自動保存しており、最大でも5分前までの地点にPITRで復元できる。障害時の即時復旧を目的とするならレプリケーションをmultiAZで組んでいれば、接続先の切り替え/付け替えのみで対応可能	独自にS3へ継続的かつ増分的なバックアップをとっており(WALアーカイブではない)、最も直近の復元時点は現在時刻の5分以内まで、と仕様面はRDSと同等。障害時の即時復旧を目的とする場合はmultiAZを用いる点もRDSと同様
ダウンタイム	HA構成における切り替え時間の目安	構成手段やダウン時の状況、死活監視のタイミングなどに依るため、一概に規定することは本来難しい	multiAZレプリケーションを組んでいる場合は、1～2分で完了すると規定。 仕組みとしてはスタンバイ側は同期ストリーミングレプリケーションでありスタンバイ側へのアクセスはできない仕様である	multiAZレプリケーションを組んでいる場合は、1分以内で完了すると規定。 仕組みとしてはスタンバイ側はリードレプリカであるが、リードのみのフラグを外し、接続先エンドポイントの付け替えを行っている
SLAにおける稼働率保証	SLAとして年間何%の稼働率を規定しているか	—	月間稼働率99.95% (ダウンタイム約21.6分) を規定。 達成できない場合、SLAに基づいてCreditが付与される	月間稼働率99.99% (ダウンタイム約4.3分) を規定。 達成できない場合、SLAに基づいてCreditが付与される

可用性

■ 耐障害性

項目	比較の観点	オンプレ	RDS	Aurora
サーバの多重化方式	HAとしてサーバを多重化する仕組みは何かがあるか	FTサーバやディスクミラーリング、ストリーミングレプリケーション、などの手法で多重化	multiAZによる多重化が可能。 また、リージョン間のDRも可能。	
データの多重化方式	データの多重化の仕組みとして何かがあるか	同上	ストレージ部分はEBSを利用しているため、多重化はEBS自体の仕様として担保されている。単独AZ内での多重化であるため、AZ内の対象ストレージがすべて破損するような事態が起きればロストする	AZをまたいで複数ストレージに書き込むようになっており、1AZあたり2か所への書き込みを3AZに対し実施する。 計6か所のうちQuorumで書き込み:4、読み込み:3で整合性を担保している

■ 災害対策(DR)

項目	比較の観点	オンプレ	RDS	Aurora
DRサイトの種類	広域圏災害における、DRの方式は何か用意されているか	ストリーミングレプリケーションや、遠隔ディスクミラーリングなどの手法が考えられる	リージョン間でリードレプリカを立てることで、DRが可能 内部的には非同期によるストリーミングレプリケーションが実施されている	GlobalDatabaseに対応(2020年3月20日)し、リージョン間の複製とDRを実現。また、リージョン間でロジカルレプリケーションを構成することでも対応可能

バックアップ/リストア

項目	比較の観点	オンプレ	RDS	Aurora
バックアップ	実行可能なバックアップ方式について	様々な方式のバックアップが利用可能(例えば、pg_basebackupによる物理バックアップ、pg_dumpによる論理バックアップ、ストレージ機能を用いたバックアップ…等)	RDSはEBS上のベースバックアップを1日1回、WALアーカイブ転送をS3に実施しておりPITRを可能としている。任意のタイミングでバックアップを取る場合は、スナップショット機能を用いる。RDSはEBS上のその時点のスナップショットを取得する。 ※pg_dumpは利用可能、pg_basebackupは利用不可	Auroraは常時ストリーミングで継続的にS3にバックアップ取得しており、頻繁なスナップショットを取る必要はなく任意の時間に復元できる。任意のタイミングでバックアップを取る場合は、スナップショットの機能を用いる。ストリーミングバックアップのその時点のメタデータを取るだけなので高速で取得可能である。 ※pg_dumpは利用可能、pg_basebackupは利用不可
リストア	実行可能なリストア方式について	物理バックアップではPGDATAのコピー、論理バックアップではpsql、pg_restore等で復元可能	スナップショットからの復元と、自動バックアップの保持期間の範囲内で任意の時刻にPITRでの復元が可能。いずれも、新たなインスタンスを作成するという動作となり、既存のインスタンスをロールバックさせることはできない ※pg_restore利用可能	
保存期間	自動バックアップの保存期間	—	デフォルトでは7日間、最大35日間まで設定可能	
復旧可能時点	自動バックアップの復旧可能時点	—	PITRを用いる場合は最大でも5分前まで	

バックアップ/リストア

■ スナップショット取得によるバックアップ処理時間の特性

- RDSは、ファイル単位で書くため、スナップショット取得時間は、取得時のEBSの容量に依存する
- Auroraは、継続的にS3に自動バックアップされているデータに対し、メタデータを取得するだけなので高速（数十秒）

■ スナップショット復元によるリストア処理時間の特性

- RDSは、スナップショット取得時の容量に依存する
- Auroraは、データはディスクストライピングにより分散保持されているため、復元時も分散処理が可能である。そのため、スナップショットの容量や復元時点の容量には影響を受けにくい。

データロード

■ DBインスタンスへのインポートについて

ここで、データロードについて、紹介する。

- インポート方法は、次の2種類がある。
 - PostgreSQLのcopyコマンドを使用してインポートする
 - S3にいったんデータロードし、そこからインポートする
- S3経由の場合、RDSが提供するaws_s3 PostgreSQL拡張機能を使用してインポートすることができる。ただし、S3へのデータロードに時間がかかる点は、注意が必要である。
 - https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/PostgreSQL.Procedural.Importing.html#PostgreSQL.Procedural.Importing.Copy

拡張性

■ (1) CPU

項目	比較の観点	オンプレ	RDS	Aurora
CPUの特性と拡張性	CPUの特性や拡張性について(換算ルールやブーストの方法、拡張時の無停止性など)	IaaSの場合の特性は、各クラウドサービスに依り異なる	ハイパースレッドで分割されたCPUリソースを1vCPUとしているので、2vCPUで1物理CPUコアに該当する。割り当てられる物理CPUの種類や特性はインスタンスタイプによって異なるため、業務要件にあったインスタンス選択が必要である。 https://aws.amazon.com/jp/rds/instance-types/ 拡張する場合は、インスタンスタイプの変更となるためサーバの停止が必要となる。リードレプリカを立ててF0することで、ダウンタイムを最小にすることは可能	ハイパースレッドで分割されたCPUリソースを1vCPUとしているので、2vCPUで1物理CPUコアに該当する。割り当てられる物理CPUの種類や特性はインスタンスタイプによって異なるが、RDSより種類は少ない。 https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/AuroraUserGuide/Concepts.DBInstanceClass.html 拡張する場合は、インスタンスタイプの変更となるためサーバの停止が必要となる。リードレプリカを立ててF0することで、ダウンタイムを最小にすることは可能

拡張性

■ (2) メモリ

項目	比較の観点	オンプレ	RDS	Aurora
メモリの特性と拡張性	メモリの特性や拡張性について(換算ルールやブーストの方法、拡張時の無停止性など)	IaaSの場合の特性は、各クラウドサービスに依り異なる	インスタンスに設定された値でメモリが確保される。拡張する場合は、インスタンスタイプの変更となるためサーバの停止が必要となる。リードレプリカを立ててF0することで、ダウンタイムを最小にすることは可能	インスタンスに設定された値でメモリが確保される。拡張する場合は、インスタンスタイプの変更となるためサーバの停止が必要となる。リードレプリカを立ててF0することで、ダウンタイムを最小にすることは可能。その他Auroraの仕様としてダブルバッファリングを使用せずファイルキャッシュを用いない特性があり、shared_buffersの値が大きく設定されている。 https://aws.amazon.com/jp/premiumsupport/knowledge-center/rds-aurora-postgresql-shared-buffers/ また、クラスタキャッシュ管理により、リードレプリカにおけるプライマリのキャッシュを同期できたり、再起動時でもキャッシュを保持できるなどの特性がある。 https://aws.amazon.com/jp/blogs/news/introduction-to-aurora-postgresql-cluster-cache-management/

拡張性

■ (3)ストレージ

項目	比較の観点	オンプレ	RDS	Aurora
ストレージの特性と拡張性	ストレージの特性や拡張性について(換算ルールやブーストの方法、拡張時の無停止性など)	IaaSの場合の特性は、各クラウドサービスに依り異なる	EBSを利用しているため、EBSの仕様に沿う。 無停止による拡張が可能であり、20GiB～64TiBで設定する	自動でスケールアップする仕様であり、10GiB～64TiBまで10GB単位で自動拡張する。 ただし、Auroraは一度拡張すると、シュリンクされず、確保分の費用が発生し続ける(VACUUMにより空き領域となれば再利用されるが、ストレージとしての確保領域はシュリンクされない、また、VACUUM FULLやpg_repackなど、再編成処理により一時的に拡張された領域も、再編成処理後にはシュリンクされない)。 シュリンクするには、一度別のAuroraクラスタを作成し、元のクラスタからデータをエクスポートしたうえで削除する必要がある

拡張性

■ (4)read/write

項目	比較の観点	オンプレ	RDS	Aurora
ディスク I/O	ディスク I/O の性能、上限・制限ラインなどについて (換算ルールやブーストの方法、拡張時の無停止性など)	IaaS の場合の特性は、各クラウドサービスに依り異なる	基本的にはインスタンスタイプと、ストレージ容量の確保量によって IOPS の上限と下限が規定される。 また、CPU 同様クレジットという概念があり、アイドル時に貯められる無料分で IOPS をバーストさせられるほか、課金で増幅させながら向上させることができる https://docs.aws.amazon.com/ja_jp/AmazonRDS/latest/UserGuide/CHAP_Storage.html	RDS のような IOPS の制限はなく、確保容量による IOPS のリミットは受けない。 また、Aurora は独自のディスクストライピングを実施しており、10GB 単位で分割されたストレージに分散してデータを格納する。この分散がうまく機能するデータ量、データ構造であるほど、高い性能を発揮する
読み取り専用インスタンスの利用可否	リードレプリカの作成可否、マスタ障害時にフェイルオーバー可能か？等	ストリーミングレプリケーションなどを設定すれば利用可能	いずれも可能。ただし、multiAZ のセカンダリは参照不可。内部的には同期レプリケーションが実施される	いずれも可能。内部的にはストレージノードへの更新を multiAZ で実施することにより実現している

障害監視

項目	比較の観点	オンプレ	RDS	Aurora
監視方式	障害監視の方式について	監視ソフトウェアを選定し、自前で設定する必要がある。	インスタンス自体の死活監視はAWS側が自動で実施し、問題があればフェイルオーバーも自動で行われる (multiAZを組んでいた場合のみ)。EC2のステータスチェック(statuscheck)のような監視はユーザ側ではできない。 監視間隔は、ドキュメントでの記載はないが、Blackbeltの資料から15-20secで動いていると思われる。 https://d1.awsstatic.com/webinars/jp/pdf/services/20190828_AWSBlackBelt2019_Aurora_PostgreSQL2.pdf ※CloudWatchからCLIやAPIを用いて情報を転送すれば、自前の監視ソフトで監視することも可能	
監視項目	障害監視の監視項目について	監視ソフトウェアによるが、基本はサーバやOS、プロセスの死活監視を実施する	multiAZを組んでいる場合は、以下のイベントでフェイルオーバーが自動的に発動する ・プライマリ利用可能ゾーンの可用性損失 ・プライマリに対するネットワーク接続の喪失 ・プライマリ上でのコンピュートユニット障害 ・プライマリのストレージ不良	
通報方式	障害監視の通報方式について	監視ソフトウェアによるが、種類は多い	あらかじめAWS側で定義された「イベント」(フェイルオーバーが発生した、等)が発生すると、Amazon SNSを通して通知が可能	

性能管理

項目	比較の観点	オンプレ	RDS	Aurora
監視方式	性能監視の方式について	監視ソフトウェアやpg_statsinfoなどのツールを用いる	インスタンスの作成や修正時に「拡張モニタリング」(1秒～60秒で設定可能)と「Performance Insights」(1秒おきにサンプリング)をそれぞれ有効にしておくことで、CloudWatch上にデータが送られ任意のメトリクスを表示させることが可能となる。	
監視項目	性能監視の監視項目について	監視ソフトウェア/ツールによる	「拡張モニタリング」はいわゆるOS関連のメトリクスを収集可能であり、「Performance Insights」ではアクティブなセッション数やセッションによるCPU負荷の内容の割合、各セッションが発行しているSQLステートメントなどを表示させて、スロークエリとその原因を把握することができる。	
通報方式	性能監視の通報方式について	監視ソフトウェア/ツールによる	CloudWatchにて、特定のメトリクスに対し閾値を設定すれば、値を超えた際にアラームをAmazon SNSメッセージを通じて通知が可能	

保守運用

項目	比較の観点	オンプレ	RDS	Aurora
計画停止の有無	計画メンテなどについて	—	関連コンポーネントのアップデートのため、計画再起動がなされることがある。インスタンスの再起動により内部のコンポーネントがアップデート、メンテナンスされたものに置き換わるという運用になっている。予告も入るし再起動タイミングはずらせるが、ずっと回避することは不可能。	
運用保守ツール	運用保守向けにはどのようなツールが用意されているか	—	運用監視としてはCloudWatchに集約されている。CloudWatch Logs等でAWS全体のログを収集監視しつつ、得られた情報をもとにどのように保守するかはPJ運用次第であるが、別途「AWS Trusted Advisor」というサービスで、AWSがユーザへのサービス提供の中で積み上げたベストプラクティスに基づくAWS環境の検査、推奨事項がある場合のガイドを受けることができる	
通報方式	パッチ情報の展開とパッチ適用のポリシーに関する項目。	—	メンテナンスの提供という形で順次、AWSが細かなパッチ適用を行い、場合によってはリソースがオフラインになる。適用タイミングについてはメンテナンスタイムの設定やオフラインを伴うものは延期も可能	
停止対応	利用しないタイミングで停止し、費用を抑えることは可能か	可能	インスタンスの停止は可能だが、7日以上経つと自動起動される。長期停止させたい場合は手動でスナップショットを取得し、インスタンスは削除する運用が必要 (自動スナップショットはインスタンス削除と連動し削除される)	

バージョン管理

項目	比較の観点	オンプレ	RDS	Aurora
バージョンアップの方式	バージョンアップはどのように実施するか、または自動で実施されるか	手動でバージョンアップを実施可能	メジャーバージョン/マイナーバージョンそれぞれについて、AWS上のバージョンアップ機能/コマンドを用いると実施可能。自動バージョンアップは、マイナーバージョンのみ設定項目としてON/OFFの設定ができる。pg_upgradeコマンドは制限がかかっており、実行はできなくなっている。	
ライフサイクル管理	ライフサイクル管理とEOSについて	PostgreSQLのVersioning Policyに従う(メジャーバージョンは5年) https://www.postgresql.org/support/versioning/	FAQ上では、メジャーバージョンはRDSによるサポート開始から少なくとも3年、マイナーバージョンはRDSによるサポート開始からは少なくとも1年はサポートすると規定されている。そこから先は不定であるが、EOSについては6か月以上の猶予期間を持って通知される	
バージョン追跡	コミュニティ版PostgreSQLのリリースにどれだけ追跡できているか	—	メジャーバージョン、マイナーバージョンともにコミュニティの公開から一般的に5か月以内にサポート開始すると規定されているが、マイナーバージョンについてはスキップされるものも存在する(10.3の次が10.5の提供となる場合など)。Beta版であれば、さらに早く試行することも可能だが、特定のリージョンに限定され、東京リージョンは対象外となっている。	

セキュリティ

項目	比較の観点	オンプレ	RDS	Aurora
ネットワーク制御	IP指定/ポート指定による接続制限の可否など	PostgreSQL本体の設定としてはpg_hba.confを利用する	VPCやセキュリティグループなどAWSのNW設定機能でかなり細かく設定が可能。基本はそこで制御する。pg_hba.confは設定できないが、ポートの変更は可能	
データ暗号化	pgcrypto関数の利用可否、その他TDEなどの暗号化手段について	pgcrypto利用可能。TDEはTDEforPGなど好みのツールを利用可能	pgcryptoは利用可能。TDE機能はなし。インスタンスの設定で暗号化を有効にするとストレージレベルの暗号化が実施される	
DB層権限管理	ロール管理と追加、スーパーユーザの利用可否など	—	PostgreSQLのスーパーユーザは設定/利用できない rds_superuserを最高権限として、ロール管理、ロール追加が可能	
不正追跡・監視	セキュリティログの取得 ログ保管期間 ログ監視、およびアラート通知など	postgresql.confでlog関連のパラメータを設定(log_connectionsなど)したり、pgauditや商用製品などを用いて自前で構築する	DBアクセスに関してはpostgresql.logにあたるものをDBログとして取得可能。また、AWS CloudTrail ログにて、RDS含むAWS全般に関するリクエストを記録させることができる CloudTrailは90日間～S3での設定により無制限に保存可能、DBログは最大7日間まで。それ以上の保存は別途CloudWatch Logsに送信しそちらで保持するよう設定する CloudTrail、DBログともにCloudWatch側に送信しアラート設定することで、Amazon SNSメッセージを通して通知可能	

コミュニティ版PostgreSQLとの違い/制約事項

■ DB運用コマンドと接続について

項目	比較の観点	RDS	Aurora
VACUUM/ ANALYZE	VACUUM/ANALYZEの実行方法、 実行タイミングなど	コミュニティ版と同じように利用可能。(Auroraについては、VACUUMにかかる時間が短縮されるケースの報告あり) autovacuumもON/OFF設定可能	
インデックスリビルド	REINDEXや pg_repackの利用可否など	REINDEX / pg_repack いずれも利用可能	
psql	psqlによる接続の可否、利用制約、注意事項など	接続先としてDNSによるエンドポイントURLが生成され払い出される。これを指定してつなぎにすればコミュニティ版と特に変わりなく使える。SSL強制などの制約もない	
アプリケーション接続	アプリケーションからの接続方法、 接続用ドライバ等	エンドポイントURLを指定する以外、コミュニティ版PostgreSQLとの違いは特 にない	

コミュニティ版PostgreSQLとの違い/制約事項

■ パラメータ設定について

項目	比較の観点	RDS	Aurora
設定方法	設定方法とSET/ALTER SYSTEMの利用可否など	AWSポータル上でGUIで設定:永続化にはパラメータグループを用いる 制約のないものはSETコマンドで設定可能。ALTER SYSTEMコマンドは利用不可	
コミュニティ版との違い/制約事項	設定変更可能なパラメータ、不可可能なパラメータ、PaaSならではのパラメータ等	大体のパラメータは変更可能だが、以下のような特性のあるパラメータやRDS独自のパラメータが存在する ①変更できないもの(WALアーカイブ関連など) ②インスタンスのスペックに応じて自動で設定してくれるもの(shared_buffersなど) ③RDS独自のパラメータ(rds.で開始) この中で②については、それが変更できるものであれば任意の値を設定することもできる。また、①であるが、③に連動して変更されるため、③を通してのみ変更が可能といったパラメータも存在する 詳細はWeb上のリファレンスで確認する、または管理画面から変更可否や設定範囲を確認できる また、Auroraについてはクラスタ単位とインスタンス単位で設定が分類されている。クラスタパラメータはAuroraのクラスタ単位で統一されるパラメータとなり、インスタンスパラメータは、クラスタ内のインスタンスにおいて個別に設定可能なパラメータとなる	

コミュニティ版PostgreSQLとの違い/制約事項

その他

項目	比較の観点	RDS	Aurora
EXTENSION	利用可能なEXTENSIONと適用方法など	rds.extensionsのパラメータで記載されており、パラメータ設定画面やSHOWクエリで対応EXTENSIONを確認できる。適用にはcreate extension文を用いる	
PGDATAとWAL保存領域の分離	性能改善の方法としてよく使われる、PGDATA領域とWAL保存領域の分離	分離不可。I/O性能については必要に応じてブーストやプロビジョンドで確保する	分離不可であるが、Auroraはディスクストライピングにより10GB単位でのI/O分散が行われる設計となっており高いIOPSが発揮される
バイナリの違い	コミュニティ版PostgreSQLとバイナリ上の違いはあるか	RDSはバイナリレベルで同一。 Auroraは独自の改造が入っているが、APやPSQLからの入力に対してはコミュニティ版と同じものが出力されるという仕様になっている。プランも同一のものが出る。エグゼキュータから先は異なる。 もし、コミュニティ版でこう動くものがAuroraでは違う、といった場合はサポートへの報告となる	
エクスポート	postgresql.confやlogなど、どこまでエクスポートが可能か	confファイルとしてのエクスポートはできない。 postgresql.logはCloudWatchなどを通してエクスポート可能	

その他

■ サポート体制について

- 利用料金の10%程でサポート契約が可能であり、開発者/ビジネス/エンタープライズでレベルが分かれ、それぞれサポートレベルと下限となる料金が異なる。
- 開発者以上の有償サポート契約であれば、RDS、Auroraに関する問い合わせ(技術的な質問や障害など)は、技術サポートケースに登録することで対応可能。
 - なお、サポートプランによって、対応品質の差はない。
- PostgreSQLの一般的な操作方法に関する問い合わせ(インデックスの作り方など)やチューニング方法に関する問い合わせに関しては、PostgreSQLのドキュメントを案内するレベルの対応となる(ベストエフォート)。
- PostgreSQLのバグについては、AWSからコミュニティにフィードバックするのが基本方針。RDS、Auroraに独自にパッチを当てて修正するということはない。

まとめ

- コミュニティ版における設計／運用と比較して、基本的には極端な制限や制約、注意事項はないという印象
- OSログインができないため、各種監視機能（障害／運用／性能）はAWSに備わっている機能をよく理解して活用するか、CloudWatchのエクスポート機能を利用して自前のSWと連携させる必要がある
- 可用性とバックアップに関してはAWSの機能でかなり充実しており、設計構築と運用の手間を省くことができる
- スケールアップが気軽にできることはクラウドの大きなメリットであるが、ストレージ領域の拡張を除き、ダウンタイムを必要とすることは注意

備考

ライセンス

本作品はCC-BYライセンスによって許諾されています。ライセンスの内容を知りたい方は[こちら](#)でご確認ください。文書の内容、表記に関する誤り、ご要望、感想等につきましては、[PGEConsのサイト](#)を通じてお寄せいただきますようお願いいたします。

- Amazon Web Services、"Powered by Amazon Web Services"ロゴ、Amazon EC2、Amazon S3、Amazon Relational Database Service (Amazon RDS) およびAmazon Auroraは、米国その他の諸国における、Amazon.com, Inc.またはその関連会社の商標です。
- IBMおよびDb2は、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。
- Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。
- Red HatおよびShadowman logoは、米国およびその他の国におけるRed Hat,Inc.の商標または登録商標です。
- Microsoft、Windows Server、SQL Server、米国 Microsoft Corporationの米国及びその他の国における登録商標または商標です。
- MySQLは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- Oracleは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。
- PostgreSQLは、PostgreSQL Community Association of Canadaのカナダにおける登録商標およびその他の国における商標です。
- TPC, TPC Benchmark, TPC-B, TPC-C, TPC-E, tpmC, TPC-H, TPC-DS, QphHは米国Transaction Processing Performance Councilの商標です。
- その他、本資料に記載されている社名及び商品名はそれぞれ各社が 商標または登録商標として使用している場合があります。

著者

(企業・団体名順)

版	所属企業・団体名	部署名	氏名
第1.0版 (2019年度 WG3)	NECソリューションイノベータ株式会社	サポートサービス事業部	湯村 昇平
	株式会社日立製作所	Software CoE OSSソリューションセンタ	稲垣 毅
	株式会社日立製作所	Software CoE OSSソリューションセンタ	牧田 伸行