PGECons
PostgreSQL Enterprise Consortium

2019年度WG3活動成果報告
クラウド検証編
Azure PostgreSQL PaaS検証

PostgreSQL エンタープライズコンソーシアム
WG3 パブリッククラウド検証チーム
日鉄ソリューションズ株式会社　伊藤 春

アジェンダ

- 活動概要
- Microsoft AzureにおけるPostgreSQL PaaSについて
- 非機能要件の達成度（抜粋版）
- ベンチマークテスト結果（抜粋版）

責任範囲

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

活動概要

検証の背景

- 近年、パブリッククラウドを利用する企業が増加しており、クラウド上でPostgreSQL等を利用したRDBを構築・運用する機会も増えてきている。

- 各クラウドベンダはPostgreSQLを利用したPaaS/SaaS を提供しているが、**クラウドベンダごとに独自の仕様**となっている項目もあり、導入にあたってはサービス仕様の調査や性能評価等を通してサービス自体の評価を事前に行う事が必要になる場合が多い。

- この流れの中、パブリッククラウド班では2018年にAWSの検証に着手。
  また2019年にMicrosoft Azureの検証に着手。
  - 近年、Microsoft Azureを採用するケースも増えている中、AzureにおいてもPaaS型PostgreSQLサービスが複数提供されていることから、**これらの特徴を整理・比較することがサービス利用時の敷居を下げPostgreSQLの導入推進に繋がる**と想定。

調査・検証内容

- Azure×PostgreSQLとしてどんな選択肢があるのか？
  - サービスメニューの種類
  - 各メニューの特徴 等
- 非機能要件の達成度は？
  - 可用性、拡張性、運用保守性 等
- 性能は？
  - ベンチマークの実行結果 等

補足

- 結果の詳細については、後日公開される成果物をご参照ください。

- 検証結果は、Microsoft社にもご確認頂き、指摘・コメントを反映した内容となります。

Microsoft Azureにおける PostgreSQL PaaSについて

AzureにおけるPostgreSQL PaaS

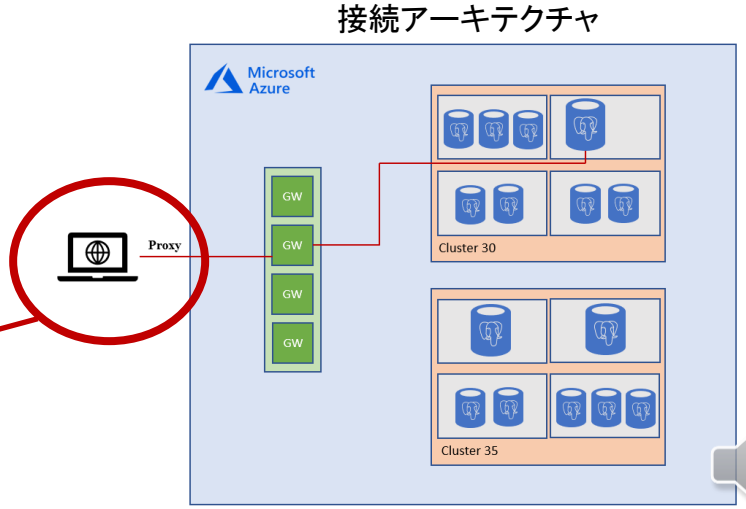
- 2020年8月時点、AzureにおけるPostgreSQL PaaSであるAzure Database for PostgreSQLでは、2種類のデプロイオプションが利用可能
  - SingleServer
    - ベーシックなマネージドRDBサービス
  - Hyperscale (Citus)
    - Citus社が提供している、PostgreSQLサーバを分散DB化（シャーディング）させる拡張モジュール「citus」が組み込まれたサービス

- 2020年8月時点で、SingleServerは全リージョンで、Hyperscaleは特定リージョンでのみ利用可能（東日本リージョンはリクエストベースで利用可能）

SingleServer

- Azure Service Fabric上のWindowsコンテナ上で動く、PostgreSQLのマネージドサービス。
- コンテナ上で動いているため、障害時の復旧（切替）も高速だが、その分性能面では少しデメリットもある
- 利用者には、DBアクセス用のエンドポイントしか公開されておらず、サーバ・コンテナへの接続は出来ない。
- クライアントからの接続経路は、
  各DBコンテナへのアクセスを振り分ける
  ゲートウェイを経由した接続となる

接続アーキテクチャ

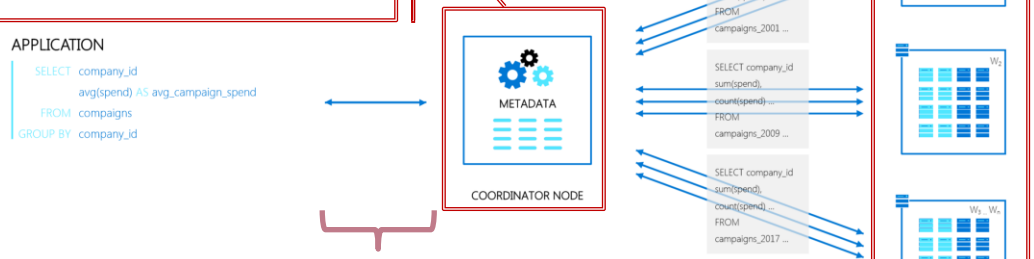


接続時に指定されたゲートウェイのIPアドレス、DB名、ユーザ名 を元にゲートウェイがコンテナへと振り分ける。

Hyperscale (citus)

- Hyperscale (citus) とは？
  - Citus Data社が提供している拡張モジュールである「citus」を組み込んだ、シェアードナッシング型のマネージド分散DBサービス。
  - データは複数ノードに分散配置（シャーディング）され、また処理実行ノードをスケールアウト方式で拡張する事が可能
  - 司令塔となり、ワーカノードへクエリを振り分けるコーディネータノードと、データが格納され、SQLクエリが実行されるワーカノードとに分かれる

コーディネータノード
・フロントアプリケーションとワーカーノードとを繋ぐ司令塔役
・対象データが格納されたワーカへのSQL投入、結果集約＋ソート処理 等を担当
・アーキテクチャ上、シングルポイントではあるが、HAモードをONにすることで二重化が可能(後述)
・拡張はスケールアップのみ対応(リクエストベース)



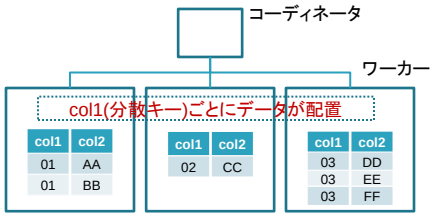
ワーカノード
・データが実際に格納され、抽出・更新等の処理もこちらで実行される
・データは、分散キーに指定されたカラムの値で格納先が判別される
・最低2台、最大20台までスケールさせることがが可能(※データリバランスによる負荷増は考慮が必要)だが、一度増やしたノードを縮小する事は出来ない
・リクエストベースで、スケールアップも可能
・可用性向上のためデータを複数ノードに重複し持たせることも可能。

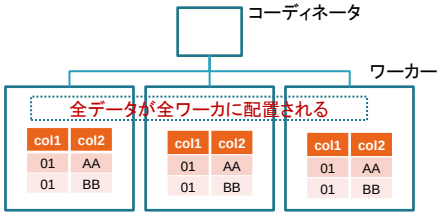
フロントアプリケーションからの接続は、すべてコーディネータノードを経由して行われる。また、直接ワーカーに接続することは出来ない点は、Azureならではの仕様。

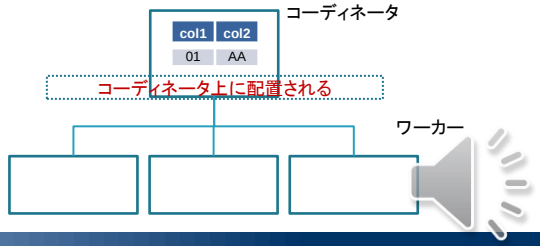
Hyperscale (citus)

- **以下3種類のテーブルを用途にあわせて利用する**

項目	分散テーブル	参照テーブル	ローカルテーブル
概要	分散キーに指定したカラムの値を元に、ワーカーノードに分散配置(シャーディング)されるテーブル	全てのワーカーノードに、テーブル上の全データが複製配置されるテーブル	分散も複製もされず、コーディネータノード上に配置されるテーブル
用途	大規模トランザクション系テーブル 等	マスタ系テーブル全般 件数規模が小さく、分散テーブルと頻繁に結合されるテーブル　等	運用管理系テーブル 等
補足	・DDLも自動で全ワーカーにカスケードされる ・分散キー列の値の更新は不可 ・オプションで、ワーカーノード間でデータを複製配置させることも可能(可用性向上のため)	・分散テーブルと結合させて利用する事が多いマスタ系テーブルを、参照テーブルとして全ワーカにあらかじめ配置しておけば、ノードを跨いだ結合処理が起きず、パフォーマンス向上となる	・テーブル作成時に分散or参照テーブルの指定が無いと、ローカルテーブルとしてコーディネータ上に作成される







非機能要件の達成度（サマリ版）

可用性

項目	比較の観点	SingleServer	Hyperscale
継続性	復元可能な直近の時間は何分前か	障害発生の5分前まで復元出来ることを保証。	2020年4月時点で、自動バックアップサービスは未実装。SingleServerと同等の機能になるように、実装を検討中
	HA構成における切り替え時間の目安	ダウンタイムの目安（数値）について公開されている情報は無いが、対象インスタンスが稼働するコンテナを別のホストで起動し直し、GWのルーティングを変更する形での切替となるため、短時間での切替は可能とのこと	高可用性オプションをONにすることで、コーディネータ/ワーカー各ノードが二重化される。 この際、データストレージ層はサーバから分離して構築されており、障害時にはストレージのアタッチ先を切り替える形で切替を実現する。切替時間の目安について公開されている情報は無し。
	SLAとして年間何%の稼働率を規定しているか	99.99% （SLAとして公表）	99.95% （SLAとして公表）
耐障害性	HAとしてサーバを多重化する仕組みは何があるか	コンテナ層は多重化されておらず、設定変更等で多重化させることも出来ない。そのため、障害時には別ノードでの再起動が必要となる。	高可用性（HA）モードをONにすることで、サーバー グループ内の全ノードを二重化することが可能。ノードがダウンすると、Hyperscale は、障害が発生したノードからそのスタンバイノードに受信接続を切り替える。これは数分で完了する。
	データの多重化の仕組みとして何があるか	実データが格納されるストレージ層は、データ同期方式で三重化されている。	高可用性（HA）モードをONにすることの他に、ワーカー側は同じデータを複数ノードに複製・配置することも可能。 （citus.shard_replication_factor パラメータを変更する）
災害対策	広域圏災害における、DRの方式は何が用意されているか	・リードレプリカを別リージョンに作成しておき、有事には手動で切り替える（自動切替機能は未実装） ・GEO冗長バックアップをONにし、有事には別リージョンでインスタンスを手動リストアする	リージョンをまたいだ切替・バックアップの仕組みは未実装。

運用・保守性

項目	比較の観点	SingleServer	Hyperscale
バックアップ	実行可能なバックアップ方式について	Azureサービス側での自動バックアップサービスあり。 ・対象はデータファイルとトランザクションログ ・頻度は、フルバックアップは週1回、スナップショットバックアップは1日1回～2回となる（サイズにより変わる） ・バックアップデータの冗長化サービスあり。（ローカル/GEO） GEOを選択すると、別リージョンにバックアップデータが保存される。	2020年4月時点で、自動バックアップサービスは未実装。SingleServerと同等の機能になるように、実装を検討中とのこと。 pg_dumpでの手動バックアップは対応可能。バックアップデータは、全ワーカーのデータが集約された状態で取得される。
リストア	実行可能なリストア方式について	上記バックアップを用いて、バックアップのリテンション期間内の任意の時点へ、ポイントインタイムリカバリが可能。 別リージョンにリストアするためには、バックアップデータの冗長化でGEOを選択する必要があり。	
障害監視	障害監視の実現方式について	サーバダウン等の障害を検知～通報してくれるサービスは無いが、Azureモニターの閾値監視を利用すれば、インスタンスの死活監視は可能。 例） DBへの接続エラー数（Failed Connection）が一定数を超過した場合、アラートメールを送信するように閾値監視の設定を入れる 等	
性能管理	性能管理の実現方式について	CPU,メモリ,IO といったリソース監視は、Azureモニターで監視可能。 クエリ ランタイム統計や待機イベントといったDB層の性能監視は、Query Performane Insightで監視可能。ただしクエリストア機能を事前にONにしておく必要があるため注意が必要。	CPU,メモリ,IO といったリソース監視は、Azureモニターで監視可能。 クエリ ランタイム統計や待機イベントといったDB層の性能監視は、Query Performane Insightで監視可能。ただしクエリストア機能を事前にONにしておく必要があるため注意が必要。
計画停止	計画停止の有無・タイミングなどについて	月一回程度、計画停止によるコンテナのマイグレーションが発生。事前に通知は来るが、アラートが飛ぶ停止タイミングの調整は不可能。ダウンタイムは1分程度。	計画停止は無し。動作するVMのセキュリティパッチ適用など、緊急性の高いメンテナンスが必要になった場合は事前に通知が来る。停止タイミングの調整も可能。

コミュニティ版PostgreSQLとの違い/制約事項

項目	比較の観点	SingleServer	Hyperscale
VACUUM/ANALYZE	VACUUM/ANALYZEの実行方法、実行タイミングなど	実行方法や実行周期・起動トリガなど、すべてコミュニティ版Postgresと同様。autovacuumも実行可能だが、逆にautovacuumをOFFにすることは出来ない。（パラメータ変更不可）	実行方法や実行周期・起動トリガなど、すべてコミュニティ版Postgresと同様。autovacuumのON/OFFも切り替え可能。
インデックスリビルド	REINDEXやpg_repackの利用可否など	reindexコマンドは実行可能。 pg_repackは未サポート。	reindexコマンドは実行可能。 pg_repackは未サポート。
psql	psqlによる接続の可否、利用制約、注意事項など	コミュニティ版Postgres同様、利用可能だが、接続先ホスト名・DB名・ユーザ名は接続時に明示的に指定する必要がある。（対話式が未対応） → 全利用者で共通のGWを経由して振り分けられるため	コミュニティ版Postgres同様利用可能だが、接続先はコーディネータノードに限られる。（コミュニティ版Citusのように、ワーカーノードに直接ログインすることは出来ない）
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	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
パラメータ設定	設定方法	変更可能なパラメータは「サーバーパラメータ」として用意されており、AzurePortalからはもちろん、Azure CLIからも変更可能。ただしALTER SYTEMコマンドは実行不可能。	
	コミュニティ版との違い/制約事項	WALアーカイブをOFFに出来ないなど、バックアップ運用を決定するようなパラメータは軒並み変更不可。またautovacuumもOFFに出来ない、shared_bufferが変更出来ないなど、全体的にAWSのPaaSと比較すると変更可能なパラメータはやや少ない（104個のみ）	コーディネータ/ワーカーそれぞれに対して「PostgreSQL」のパラメータと「Citus」のパラメータ設定が可能だが、いずれも設定可能なパラメータは制限されている。
EXTENSION	EXTENTIONの対応状況	適用可能なextension、およびそのバージョンは、MS社サイトに公開されている情報、もしくはpg_available_extensionsビューを参照すれば確認可能。 適用にはcreate extensionを用いる。	

ベンチマークテスト結果（概要）

実機検証計画（性能測定）

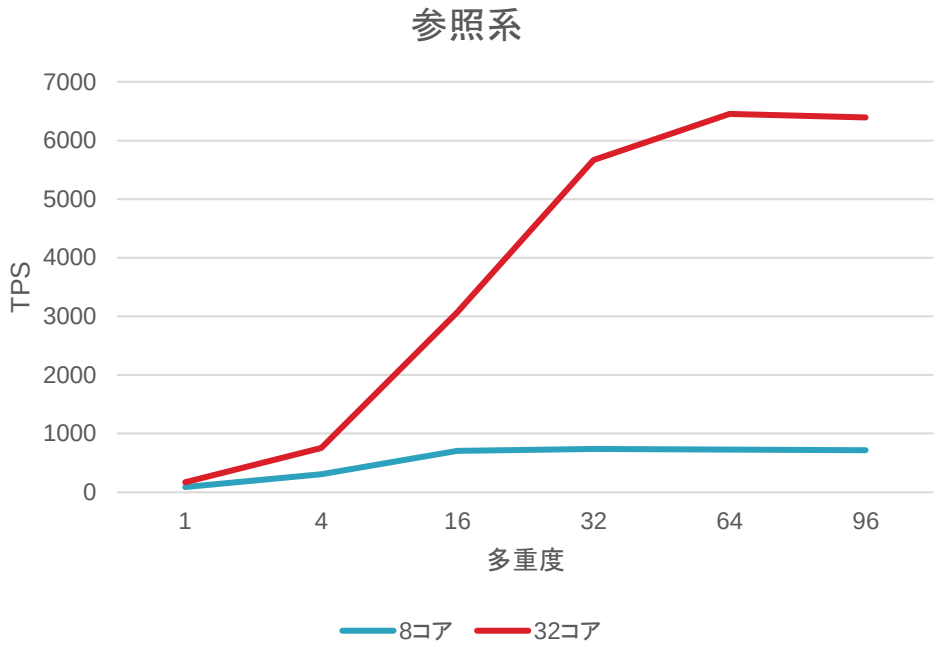
- ベンチマークテストの概要は以下の通り
  - ベンチマークシナリオ、実行方法、パラメータ設定等の条件は、AWS検証時の内容を流用

項目	SingleServer	Hyperscale
ベンチマーク種別/ツール	単純参照、単純更新 ：pgbench 業務トランザクション（TPC-C） ：HammerDB	（2020年度に実施予定）
CPU/メモリ	メモリ最適化タイプ 8コア80GB 32コア320GB	
ストレージ	500GB（IO上限：1500IOPS）	
多重度	1，4，16，32、64，96	
PostgreSQL Ver	10.6	

検証結果

- pgbench（参照系）検証

多重度	8vCPU(TPS)	32vCPU(TPS)
1	89.73353	167.3619
4	303.5597	752.7121
16	705.5004	3069.272
32	737.6824	5668.561
64	728.0791	6452.567
96	718.3075	6391.86



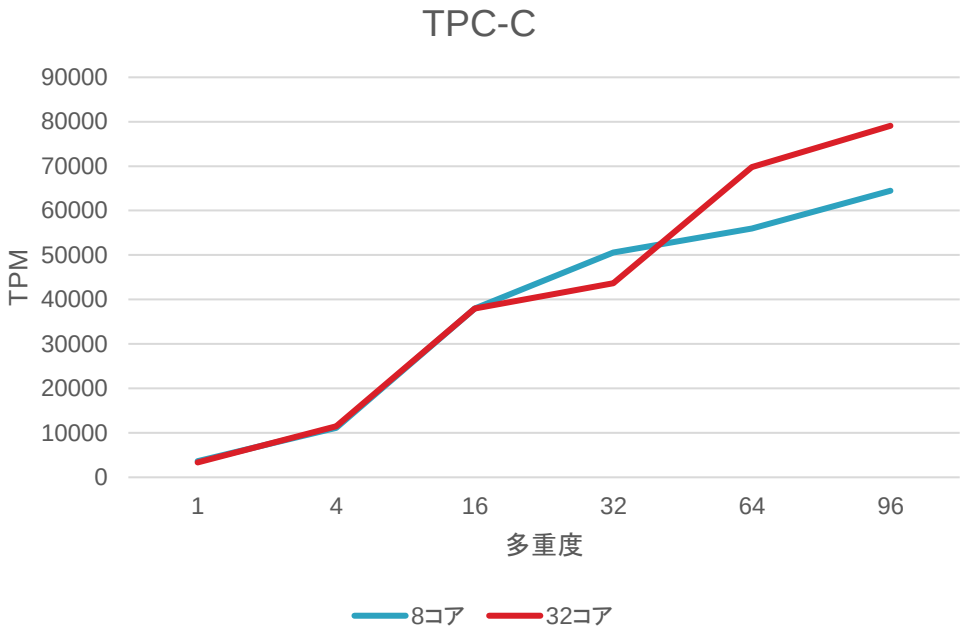
- 多重度に比例しスループットも増加。8コアでは多重度16近辺、32コアでは多重度32～64近辺を上限にスループットの上昇が止まり、それ以上は減少へと向かう

- スループットの限界地点でCPU使用率が100%付近に達しており、単純参照処理としての性能は、この多重度が上限ラインになると考えられる。

検証結果

- TPC-C (HammerDB) 検証

多重度	8vCPU(TPM)	32vCPU(TPM)
1	3585.667	3294
4	11113.67	11467.67
16	37962.33	37968.67
32	50554	43617.67
64	55976	69821.67
96	64452.67	79063



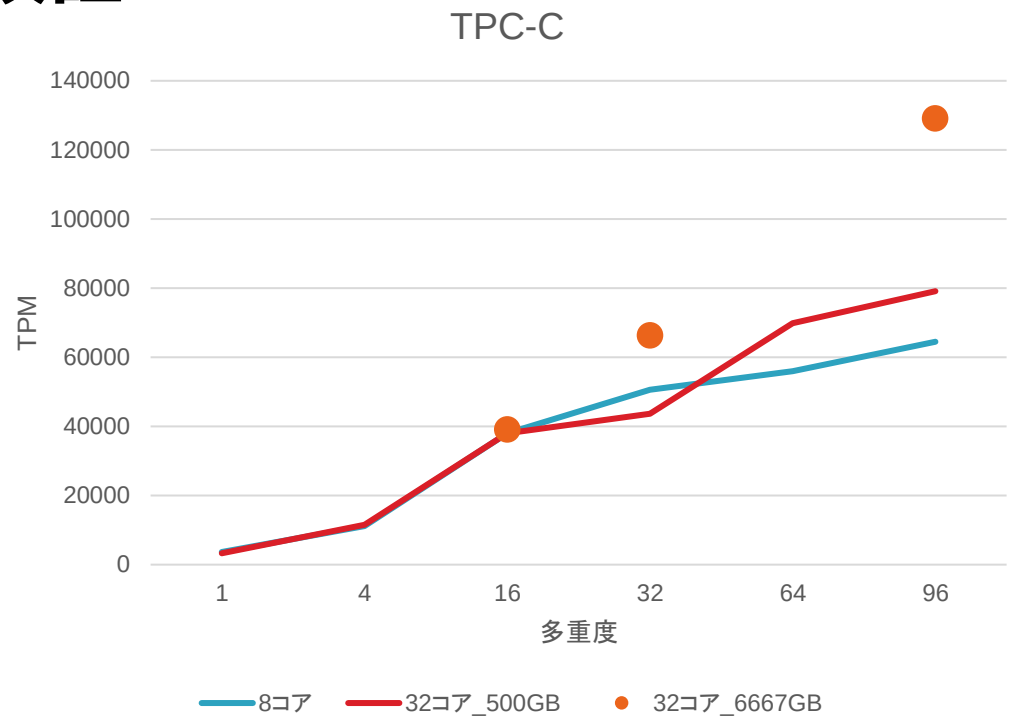
- 多重度と共にスループットは上昇し続けたが、96多重まで上げても頭打ちとなることは無かった。また64多重までの間は8コアと32コアとの間でスループットに差が見られなかった。
  - 32コア側のリソース使用率を確認すると、CPU使用率にはまだ余裕があったが、IO使用率は60～80%に近い値で推移（瞬間的に100%を記録することもあり）していた。このことから、**IOネックになっていた可能性が高く、再試験を実施。**（次項）

再検証結果

I/O制限を外した状態で再試験
(ストレージサイズを6667GBに変更)

- TPC-C (HammerDB) 検証

多重度	32vCPU(TPM) 500GB	32vCPU(TPM) 6667GB
1	3294	-
4	11467.67	-
16	37968.67	39022
32	43617.67	65309
64	69821.67	-
96	79063	126668



- I/O制限がかかっていた状態と比べると、32多重以降で大きくスループット値が上昇した。この事から、I/Oネックでスループットが伸び悩んでいたことが明確であり、SingleServer利用時にはデータサイズだけでなく、I/O制限の事も考慮したストレージサイジングが重要であることが分かった。

結果まとめ

- Microsoft AzureにおけるPostgreSQL PaaSの種類、特徴等について、Microsoft社の協力の下色々な視点で確認をすることが出来た。
  - 非機能要件の達成度としては、他のパブリッククラウドサービスと比べ同等な面もあれば、達成度が異なる点も見られるため、事前に差異を確認・認識しておくことが望ましい。
  - 性能面（SingleServer）においては概ね予想通りの結果となったが、IOPSの上限値がストレージサイズによって決まる点は注意が必要。またより高スループットを目指す場合はHyperscale（citus）の採用や今後GA予定のSingleServer Gen2を選択肢の一つとして考慮する事が望ましい。