

IT 資格道場

# DP-300 学習用テキスト

予備知識ゼロから合格ラインへ——読んで理解する1冊目

無料サンプル(第1章まで)

DP-300 出題範囲(Skills measured)2026-10-27版 準拠  
2026年10月版

## ご利用にあたって

- このテキストは「IT資格道場」([www.shikaku-doujou.com](http://www.shikaku-doujou.com))の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

# 目次

---

## この教材の使い方

試験の基本情報

試験の設計図:5 分野と配点

構成

4 ステップ学習法

## 第1章 Plan and implement data platform resources [1.1/1.2/1.3]

Azure SQL ソリューションを計画してデプロイする [1.1]

スケーリングとパフォーマンスに優れたリソースを構成する [1.2]

移行戦略を計画して実装する [1.3]

サンプルはここまでです

3冊の使い方は、ダウンロード欄の案内を参照してください。

SAMPLE

# この教材の使い方

このテキストは、Microsoft の Associate 資格 **Microsoft Certified: Azure Database Administrator Associate(試験コード DP-300、試験名 Administering Microsoft Azure SQL Solutions)** に合格することを目的に作られました。DP-300 は、Azure 上の SQL を**データベース管理者(DBA)の視点で運用する**資格。製品の名前を知っているかではなく、「この要件なら、どの製品・どの設定・どの操作を選ぶか」を問われます。

Azure の SQL には、SQL Database、SQL Managed Instance、Azure VM 上の SQL Server という 3 つの置き場があり、同じ「バックアップ」「認証」「ジョブ」でも、置き場によって使える機能と手順が違います。このテキストは、その違いを軸に、公式の出題範囲(5 分野)に沿って説明していきます。

公式の想定読者は、Azure 上の SQL Server と Azure SQL のデータベース(リレーショナル データ)を、運用の責任者として扱う人です。

とはいえ、深い開発経験は要りません。**各機能の「何をする道具か」と「選ぶ条件・設定の勘どころ」を押さえれば合格ラインに届きます**。T-SQL や PowerShell のコード例は出てきますが、**丸暗記は不要**で、「どのコマンドが何をするか」の対応が読めれば十分でしょう。

## 試験の基本情報

| 項目  | 内容                                                                  |
|-----|---------------------------------------------------------------------|
| 試験名 | Microsoft Certified: Azure Database Administrator Associate(DP-300) |

| 項目      | 内容                                                                                                                       |
|---------|--------------------------------------------------------------------------------------------------------------------------|
| 出題数     | Microsoft は公式に固定公表していません (Microsoft の認定試験は一般に 40～60 問程度とされます。このサイトのコースは 1 回 50 問と推定して作っています)                             |
| 出題形式    | 択一・複数選択のほか、ドラッグ アンド ドロップ、ホットエリア、ケーススタディ、「問題と解決策」(同じ状況に対して解決策だけを差し替え、それぞれに「はい」「いいえ」で答える形式)が含まれることがあります。試験ごとの形式は事前に公表されません |
| 試験時間    | 100 分(公式認定ページ記載)。ラボを含まない区分の試験時間です                                                                                        |
| 合格ライン   | 1,000 点満点中 <b>700 点</b> (スケールスコア)                                                                                        |
| 受験料     | 受験する国・地域により異なります(日本での価格・税の扱いは申込時に確認してください)                                                                               |
| 受験方法    | ピアソン VUE。テストセンターまたはオンライン監督試験                                                                                             |
| 言語      | 日本語で受験可(英語ほか多言語提供)                                                                                                       |
| 資格の有効期限 | 取得後 12 か月(Microsoft の無料オンライン評価で更新)                                                                                       |

分からない問題も、必ずどれかを選んで埋めてください。受験料や実施形式は、いつまでも同じとはかぎりません。申し込み時に、公式ページの最新情報を必ず確認してください。

# 試験の設計図：5 分野と配点

このテキストが準拠する Skills measured は **2026 年 10 月 27 日版**で、前の版(2026 年 4 月 24 日版)との差は 1 行だけ。「保存データと転送中データの保護」にある Always Encrypted のエンクレーブの語(VBS エンクレーブ → セキュアエンクレーブ)が変わりました。

| 分野                             | 配点     | 一言でいうと                                 |
|--------------------------------|--------|----------------------------------------|
| ① データ プラットフォーム<br>リソースの計画と実装   | 15～20% | 選ぶ・作る・移す —— 製品の<br>選択・デプロイ・スケール・移行     |
| ② セキュリティで保護され<br>た環境の実装        | 20～25% | 守る —— 認証・暗号化・ネ<br>ットワーク・監査             |
| ③ データベース リソースの<br>監視・構成・最適化    | 20～25% | 測って直す —— 監視・クエ<br>リ最適化・保守              |
| ④ タスクの自動化の構成と<br>管理            | 15～20% | 回す —— ジョブ・IaC・自動<br>化                  |
| ⑤ 高可用性と障害復旧<br>(HA/DR)環境の計画と構成 | 20～25% | 戻す・切り替える —— バッ<br>クアップ・復元・レプリケー<br>ション |

**②セキュリティ、③監視・最適化、⑤HA/DR が、それぞれ 20～25% で最大**です。この 3 分野で全体の大半を占め、①と④が 15～20% で続きます。配点の偏りはそれほど大きくないため、**どれか 1 分野を捨てる作戦は取れません**。時間が限られるなら、まず配点の重い②③⑤、次に①④の順で仕上げると効率的でしょう。

## 構成

このテキストは 5 分野を 5 つの章に分ける構成。同じ機能が複数の分野にまたがることもあります(たとえばバックアップは、③の保守でも⑤の復元でも顔を出します)、並べ方は**公式の分野順そのまま**。

- 第 1 章 … 分野①(データ プラットフォーム リソースの計画と実装)
- 第 2 章 … 分野②(セキュリティで保護された環境の実装)
- 第 3 章 … 分野③(データベース リソースの監視・構成・最適化)
- 第 4 章 … 分野④(タスクの自動化の構成と管理)
- 第 5 章 … 分野⑤(HA/DR 環境の計画と構成)

見出しの末尾の [1.1] などの数字は、公式の出題範囲の番号です。

## 4 ステップ学習法

購入者限定テキストは 3 冊で 1 セット。次の順番で使うと最短です。

- **STEP 1(理解する)**: 「学習用テキスト」を通読し、Azure SQL の地図と「選ぶ条件・設定の勘どころ」を頭に入れる
- **STEP 2(解き方を掴む)**: 「頻出度別ノート」で頻出度 A の論点を解いて、解き方を掴む
- **STEP 3(腕試し)**: 本サイトの問題演習で出題形式に慣れる
- **STEP 4(再確認)**: 「頻出度別ノート」に戻って頻出度 A の論点を再確認する(試験直前もここ)

「暗記用ノート」は STEP には数えず、空き時間や用語の再確認に使ってください。

DP-300 は操作を問う試験なので、可能なら Azure の無料アカウントで SQL Database を作って、実際に触ると定着が段違いです。時間がなければ、先に「頻出度別ノート」で頻出度 A の論点を確認し、それを含む章から読み始めても構いません。

---

SAMPLE

# 第1章 Plan and implement data platform resources [1.1/1.2/1.3]

分野① | 配点 15～20% | 出題範囲 1.1～1.3

この章では、Azure で SQL を動かす土台を決めて作る手順を扱います。どの製品を選ぶか、どう作るか、どう大きくするか、オンプレミスからどう持ってくるかの順です。配点は 15～20% (中央値 17.5%)。製品ごとの違いは、後の章のセキュリティ、監視、可用性の設問でも前提になります。

先に出す用語は 3 つ。Azure SQL Database、Azure SQL Managed Instance、SQL Server on Azure Virtual Machines です。この章の設問の大半は、要件を読んでこの 3 つのどれかを選び、続けて作り方や移し方を選ぶ形をしています。

## Azure SQL ソリューションを計画してデプロイする [1.1]

### データベース オファリング (database offering) を要件から選ぶ [1.1.1]

Azure SQL ファミリは、SQL Server のエンジンを Azure で動かす 3 つの製品の総称です[^c1-1]。製品ごとに、利用者が管理する範囲が違います。

| 製品           | 種類   | 互換性と特徴                                 | 最大サイズ  |
|--------------|------|----------------------------------------|--------|
| SQL Database | PaaS | 単一データベースとエラスティックプールの 2 つの展開形態。CLR は非対応 | 128 TB |

| 製品                      | 種類   | 互換性と特徴                                                | 最大サイズ                                              |
|-------------------------|------|-------------------------------------------------------|----------------------------------------------------|
| SQL Managed Instance    | PaaS | SQL Server とほぼ 100% の機能互換。仮想ネットワーク内のプライベート IP。CLR に対応 | 16 TB (この比較ページの値。Next-gen General Purpose は 32 TB) |
| SQL Server on Azure VMs | IaaS | SQL Server と OS を完全に制御できる。VM の管理は利用者の責任               | インスタンスあたり 256 TB                                   |

選び方は、まず「OS やエンジンの設定を自分で握る必要があるか」で分かります。握る必要があるなら SQL Server on Azure VMs (IaaS)。VM だけが、メンテナンスの開始時期、復旧モデルの simple や bulk-logged への変更、サービスの停止と開始を自分で決められます。握る必要がなく、既存の SQL Server アプリを最小の変更で移したいなら SQL Managed Instance です。SQL Server Agent、データベース間のクエリとトランザクション、リンク サーバー、CLR、Service Broker に対応しています[^c1-2]。新しく作るクラウド ネイティブなアプリで、1 つのデータベースを手軽に持ちたいなら SQL Database です。

可用性の見方も違います。PaaS は IaaS より高い SLA を保証します (ゾーン冗長にするとさらに上がります)。VM は 2 台目を可用性セットや可用性ゾーンに置き、SQL Server 側の仕組みも組んで、はじめて高い SLA に届きます。

間違えやすいのは「Managed Instance は PaaS だからサーバー レベルの機能が使えない」という思い込みです。逆で、Managed Instance は SQL Server のインスタンス単位の機能 (SQL Server Agent やリンク サーバー) を持ちます。SQL Database はデータベース単位の製品 (インスタンス単位の機能は持たない)。

SQL Server Agent、データベース間のクエリ、リンク サーバー、CLR の 4 つで、3 製品の差がはっきり出ます[^c1-59]。

| 機能               | SQL Database      | SQL Managed Instance                                                       | SQL Server on Azure VMs |
|------------------|-------------------|----------------------------------------------------------------------------|-------------------------|
| SQL Server Agent | なし(エラスティックジョブで代用) | あり                                                                         | あり                      |
| データベース間のクエリ      | なし(エラスティッククエリで代用) | あり。インスタンス内ならトランザクションも可                                                     | あり                      |
| リンク サーバー         | なし                | あり。接続先は SQL Server 系の製品 (SQL Server・SQL Database など) のみで、Oracle などの他社製品は不可 | あり                      |
| CLR              | なし                | あり。CREATE ASSEMBLY でファイル システムは使えない                                         | あり                      |

オンプレミスの SQL Server で SQL Server Agent のジョブとリンク サーバーを使っていて、アプリは変えられず、OS の管理は Azure に任せたい、という場面を考えます。合うのは SQL Managed Instance です。SQL Database には SQL Server Agent に当たる機能がありません (エラスティック ジョブで代用)。VM は OS と修正の管理が利用者に残ります。要件に「OS への管理者アクセスが必要」が加わると、答えは SQL Server on Azure VMs に変わります。

# 購入モデルとサービス レベル [1.1.1]

SQL Database には、課金の単位として 2 つの購入モデルがあります[^c1-3]。

DTU モデルは、CPU とメモリと読み書きを混ぜた 1 つの指標でまとめた、バンドル型です。サービス レベルは Basic、Standard、Premium の 3 つ (DTU モデルの区分)。vCore モデルは、論理 CPU を単位にして、ハードウェアとサービス レベルを自分で選びます。サービス レベルは General Purpose、Business Critical、Hyperscale の 3 つです。

| サービス レベル          | 構造                                                              | 選ぶ場面               |
|-------------------|-----------------------------------------------------------------|--------------------|
| General Purpose   | 計算とストレージを分離。ストレージ遅延は大きめ (5～10 ms)                               | 予算重視の一般的な業務        |
| Business Critical | ローカル SSD。高可用性のセカンダリレプリカが 3 つ自動で付く。コストは General Purpose の約 2.7 倍 | 低遅延と高い耐障害性が必要な業務   |
| Hyperscale        | 計算とストレージを分離し、ストレージを独立して伸ばす。最大 128 TB                            | データが大きい。読み取りを増やしたい |

Business Critical は、3 つのセカンダリのうち 1 つを読み取り専用の作業に使えます (読み取りスケール アウト)。Premium と Business Critical と Hyperscale では、新しいデータベースで既定で有効です。General Purpose と Basic と Standard にはレプリカがないので、読み取りスケール アウトは使えません[^c1-4]。

サーバーレスは、vCore モデルの計算レベルの 1 つです。使った計算を秒単位で課金し、使われていない間はデータベースを自動で一時停止して、ストレージだけを課金します。General Purpose と Hyperscale で使えます。常時動くアプリや、再開の待ちが許されないアプリには、プロビジョニング済みの計算を選びます<sup>[^c1-5]</sup>。

### 単一データベース、エラスティック プール、インスタンス プール [1.1.1]

単一データベースは、1 つのデータベースに計算を割り当てる形です。エラスティック プールは、使う時間帯がずれる複数のデータベースが、1 つのリソースの束 (eDTU または vCore) を分け合う形です。利用パターンが読めない多数のデータベースを、コスト効率よく運用したい場合に向きます。ピークが同じ時間帯に重なるデータベースをまとめても、同時に要る量は減らないので、必要な束は小さくなりません。常に高い負荷が続き、分け合う相手のいないデータベースは、プールに入れても得がないので単一データベースにします。典型は SaaS のマルチテナント。プールにはデータベースごとの課金がなく、プールが存在する時間ごとに、設定した最大の eDTU または vCore で課金されます。使われなくても課金は止まりません<sup>[^c1-6]</sup>。1 つのプールに入れられるデータベースの数には、サービス レベルと計算サイズごとに上限があり、超える分は別のプールに分けます。

同じ考え方を Managed Instance に当てたのがインスタンス プールです。小さなインスタンスをまとめて、1 つの仮想クラスターで動かし、コストを下げます。2 vCore のインスタンスは、プールの中でだけ作れます<sup>[^c1-7]</sup>。

### 自動デプロイの方法の使い分け [1.1.2]

デプロイを自動化する方法は 4 つあります<sup>[^c1-8]</sup>。

- ARM テンプレート: JSON で構成を宣言する。サーバー、エラスティック プール、フェールオーバー グループ、監査などの公開テンプレートがある
- Bicep: ARM テンプレートと同じ宣言型で、記述が短い。単一データベースを宣言的にデプロイできる
- Azure PowerShell: New-AzResourceGroup、New-AzSqlServer、New-AzSqlServerFirewallRule、New-AzSqlDatabase の順に実行する
- Azure CLI: az sql server create でサーバー、az sql server firewall-rule create でファイアウォール規則、az sql db create でデータベースを作る

Managed Instance は、CLI なら az sql mi create でインスタンスを、az sql midb create でその中のデータベースを作ります。PowerShell なら New-AzSqlInstance と New-AzSqlInstanceDatabase です[^c1-9]。ネットワークは New-AzVirtualNetwork 、 New-AzRouteTable 、 New-AzNetworkSecurityGroup で先に用意します。

選び方の軸は、再現性と宣言性です。同じ環境を何度も作り直す、変更を差分で管理する、という要件なら ARM テンプレートか Bicep です。手順を順に実行するスクリプトや、既存の運用に組み込む用途なら PowerShell か CLI です。ポータルは手作業なので、自動化の選択肢には入りません。

## Azure Arc 対応(Arc-enabled)の SQL サービス [1.1.3]

Azure Arc は、Azure の外にあるリソースを Azure の管理に載せる仕組みです。SQL では形が 2 つ(SQL Server と Managed Instance)。

SQL Server enabled by Azure Arc は、オンプレミスや他のクラウドにある SQL Server を Arc に接続し、Azure から一元的に管理します。詳細な在庫を Azure で確認できます。接続すると、ベスト プラクティス評価、Microsoft Entra 認証、Microsoft Defender for Cloud をオンプレミスの SQL Server に

適用できます。Entra 認証に使えるのは SQL Server 2022 (16.x) 以降だけ（それ以前は対象外）。接続の手順は 3 段です。Azure Connected Machine エージェントを入れ、マシンを Azure Arc 対応サーバーとして Azure に接続し、そのあと SQL Server 用の Azure 拡張機能を入れます。拡張機能が入ると、SQL Server の詳細な在庫を Azure で確認できます。SQL Server の従量課金 (pay-as-you-go) も、Arc に接続した SQL Server に対して使えます[^c1-10]。

SQL Managed Instance enabled by Azure Arc は、選んだインフラの上で動かす Azure SQL のデータ サービスです。最新の SQL Server エンジンとほぼ 100% 互換で、データ主権を保ったまま既存アプリを最小の変更でリフト アンド シフトできます[^c1-11]。

すでにあるオンプレミスの SQL Server を、そのまま Azure から管理したいなら前者です。Azure SQL のサービスそのものを自分のインフラで動かしたいなら後者です。

## Microsoft Fabric の SQL database [1.1.4]

SQL database in Microsoft Fabric は、Azure SQL Database に基づく、開発者向けのトランザクション データベースです[^c1-12]。Azure SQL Database と同じ SQL Database Engine を使います。特徴は、データが自動的に OneLake へほぼリアルタイムで複製され、Parquet 形式の分析向けデータに変換される点です。SQL 分析エンドポイントも一緒に作られます(自動)。

向く場面は、AI アプリ(ベクターと RAG)、リバース ETL、運用データ ストア (ODS)、そしてトランザクションと分析の両方が要る translytical なアプリ。接続には Microsoft Entra 認証を使い、接続するユーザー、サービス プリンシパル、またはそのグループには、Fabric のデータベースに対する Read item 権限が必要です。サポートされる接続ポリシーは Default のみ(現時点)。

Fabric の SQL database は、Azure SQL Database やそのミラーリング データベースとは別のものです。Fabric のワークスペースの中に作る運用データベースが欲しいなら Fabric、既存の Azure SQL のデプロイを増やすなら Azure SQL Database です(置き場所が決まります)。

## パーティション分割の計画 [1.1.5]

パーティション分割は、大きなテーブルやインデックスを、列の値で行の集まりに分ける設計です。メンテナンスや保持の操作を、テーブル全体ではなく一部のパーティションだけに絞れるので、速くなります。圧縮、インデックスの再構築、切り詰め (truncate) を、パーティション単位で行えます[^c1-13]。

部品は 2 つです。

- パーティション関数: 分割列 (partitioning column) の値に応じて、行がどのパーティションに入るかを決めるオブジェクト
- パーティション スキーム: 関数で作られた各パーティションを、ファイルグループにひもづけるオブジェクト

関数の境界値を  $n$  個書くと、パーティションは  $n + 1$  個できます。境界値は 14,999 個までなので、パーティションは最大 15,000 個です。分割列にできるのは 1 列だけです。複数の列で分けたいなら、計算列で連結して使います。text、ntext、image、xml、timestamp、varchar(max)、nvarchar(max)、varbinary(max) などは分割列にできません[^c1-14]。

RANGE LEFT と RANGE RIGHT の違いは、境界値そのものがどちら側のパーティションに入るかです。LEFT は境界値が区間の左側のパーティションに属します。RIGHT は右側、つまり各パーティションの下限。指定しなければ LEFT が既定です。datetime 系の列では、月初を境界値にして RIGHT を使うと、「2 月 1 日から 2 月末まで」が 1 つのパーティションに入り、扱いやすくなります。

LEFT で月初を境界にすると、2 月 1 日の行が 1 月のパーティションに入り、月と区間がずれます。

Azure SQL Database にはファイルグループが PRIMARY しかありません。スキームでは ALL TO ([PRIMARY]) と書いて、全パーティションを PRIMARY に置きます。PRIMARY はキーワードなので、角かっこで囲む必要があります ([PRIMARY])。

### パーティションの切り替えと削除・アーカイブの計画 [1.1.5]

ALTER TABLE ... SWITCH は、パーティションを別のテーブルへ、データを動かさずにメタデータだけで切り替えます。古い月を丸ごとアーカイブテーブルに出し、新しい月の空のパーティションを足す運用を、スライディング ウィンドウと呼びます。主な条件は 3 つです。移動先のテーブル (またはパーティション) が存在して空であること、移動元と移動先が同じファイルグループにあること、対応するインデックス (またはインデックスのパーティション) も同じファイルグループにあることです。ほかにも多くの制限があります<sup>[^c1-15]</sup>。

切り替えが速く済む前提が、整列インデックス (aligned index) です。テーブルと同じパーティション スキームで作ったインデックスは、整列していると言います。整列していれば、テーブルと非クラスター化インデックスの構造を保ったまま、パーティションを効率よく出し入れできます。一意の非クラスター化インデックスをパーティション分割するなら、キーに分割列を含める必要があります<sup>[^c1-16]</sup>。

削除については、DELETE で行を消すより、古いパーティションを SWITCH で切り離してからテーブルごと落とすほうが、ログも時間も小さく済みます。

## シャーディング(sharding)の考え方 [1.1.6]

シャーディングは、同じ構造のデータを複数の独立したデータベースに分散させる、水平スケールアウトの手法です。SaaS のテナント分割でよく使います。水平スケールはデータベースの数を増減し、垂直スケールは個々のデータベースの計算サイズを変えます(スケールアップ)[^c1-17]。

シャーディングが必要になるのは、データ総量が単一のデータベースの制約を超える場合、スループットが単一のデータベースの能力を超える場合、テナントを物理的に分けたい場合、地理的に配置したい場合などです。

Elastic Database クライアント ライブラリは、シャード化したデータベースの作成と管理を行うライブラリです。シャード マップ マネージャーが、どのキーの範囲がどのシャードにあるかを管理します。ツール群には、ほかに 4 つの道具があります。

- split-merge ツール: シャード間でデータを移す
- エラスティック ジョブ: 複数のシャードに T-SQL を流す
- エラスティック クエリ: 複数のデータベースにまたがって読む
- エラスティック トランザクション: 複数のデータベースにまたがる更新

マルチテナントのデータベースから、シングルテナントのデータベースへデータを移したいときは、split-merge を使います。

シャードを足す時は、まず新しいデータベースを作り、split-merge で移したいキー範囲のデータを移します。split-merge は、データを写し終えてから、シャード マップの対応を新しいデータベースへ更新します(移している間、そのキーの接続は一時的に使えません)。アプリはキーから接続先を引くので、対応が更新された後は、新しいデータベースへつながります。

# スケールアウトの選択肢の比較 [1.1.6]

水平に伸ばす方法は 4 つあり、役割が違います。

| 方法          | 向く要件                                                     |
|-------------|----------------------------------------------------------|
| 読み取りスケールアウト | 読み取り専用のクエリを、セカンダリレプリカへ逃がす。Business Critical と Hyperscale |
| Hyperscale  | 1 つのデータベースを最大 128 TB まで伸ばし、名前付きレプリカで読み取りを増やす             |
| シャーディング     | 書き込みも含めて、データベースを複数に分ける。テナント分離や、単一のデータベースの上限超え            |
| エラスティック クエリ | 別々のデータベースにある表を、1 つの接続から横断して読む                            |

エラスティック クエリはプレビューの機能です[^c1-18]。書き込みの負荷や容量そのものが原因なら、読み取りを逃がしても解決しません。その場合は Hyperscale かシャーディングです。

# オフリングのデプロイ [1.1.7]

SQL Database は、リソース グループ、論理サーバー、ファイアウォール規則、データベースの順に作ります[^c1-19]。論理サーバーが、ログイン、ファイアウォール規則、監査の設定をまとめる親になります。データベースを作る時に、照合順序、メンテナンス ウィンドウ、バックアップ ストレージの冗長性（ローカル冗長、ゾーン冗長、geo 冗長）を選びます。バックアップ ストレージの冗長性は、既定が geo 冗長（GRS）。開発や検証など、冗長が要らない環境ではローカル冗長にしてコストを下げます。T-SQL で作る場合は、CREATE DATABASE

の権限が必要で、サーバー管理者ログイン、サーバーの Entra 管理者、master の dbmanager ロールのメンバーが実行できます。

SQL Managed Instance は、専用のサブネット、ルートテーブル、ネットワークセキュリティグループを使って作ります。サブネットは Microsoft.Sql/managedInstances に委任します<sup>[^c1-20]</sup>。1つのインスタンスに最低 32 個の IP アドレスが要り、サブネットマスクは /27 以上の広さが必要です。作成の権限は Microsoft.Sql/managedInstances/write。サブネットがすでにそのサービスへ委任済みなら、ユーザーに必要なのは、サブスクリプションスコープのこの権限だけです。

作成には時間がかかります。General Purpose の新しいインスタンスは、95% の操作が 30 分で終わります。ゾーン冗長のインスタンスは、95% が 4 時間で終わります。同じサブネットで別の操作 (長い復元やスケール) が走っていると、デプロイは遅れます<sup>[^c1-21]</sup>。

SQL Server on Azure VM は、Azure Marketplace の SQL Server イメージを選んで仮想マシンを作ります<sup>[^c1-22]</sup>。イメージの料金には、SQL Server のライセンスが含まれます (Developer は開発とテスト用に無料、Express は無料)。手持ちのライセンスを使う場合は、Azure Hybrid Benefit を有効にして、VM の分だけを払います。SQL Server IaaS Agent 拡張機能が、ポータルからの管理、ライセンスの柔軟性、自動バックアップ、自動修正プログラムの適用を可能にします。現在の SQL 仮想マシンのギャラリー イメージには、この拡張機能が既定で入っています<sup>[^c1-23]</sup>。自分でインストールした SQL Server にも、拡張機能を登録できます。

## ハイブリッドの SQL Server の展開 [1.1.8]

オンプレミスの SQL Server を Azure とつなぐ方法は、2 つあります。

1 つ目は、Azure Arc への接続 (onboard) です。Azure から管理する手順で、接続用のスクリプトが、まず Azure Connected Machine エージェントを入れ、続けて SQL Server 用の Azure 拡張機能を入れます[^c1-24]。

2 つ目は、Managed Instance link です。SQL Server と SQL Managed Instance の間で、分散可用性グループを使ってニア リアルタイムにデータをレプリケートします[^c1-25]。対応は、SQL Server のバージョンで違います。

| SQL Server     | レプリケーション     | フェールバック                        |
|----------------|--------------|--------------------------------|
| 2016、2017、2019 | 一方向          | フェールオーバーするとリンクが切れ、フェールバックはできない |
| 2022、2025      | 双方向。災害復旧に使える | できる                            |

リンクのモードは 2 つあります。1 つのデータベースを対象にするモードと、Always On 可用性グループの全データベースを対象にする複数データベース モード (プレビュー) です。複数データベース モードには、特定の累積更新プログラム、サポートされるエディション、すべての SQL Server レプリカでのオプトインが必要です。

### IaaS の SQL Server の修正プログラム (patches) の適用 [1.1.9]

SQL Server on Azure VM の修正は、自動修正 (Automated Patching) か、Azure Update Manager で行います[^c1-26]。

自動修正は、VM にメンテナンス ウィンドウを設定し、更新をそのウィンドウの中でだけ入れる機能です。SQL Server IaaS Agent 拡張機能に依存します。対象は、Important または Critical とされた Windows と SQL Server の更新だけ (それ以外は手動)。サービス パックと累積更新プログラムは、手動で入れるか、Azure Update Manager を使います。PowerShell では New-

AzVMSqlServerAutoPatchingConfig で構成し、  
AutoPatchingSettingDayOfWeek で曜日、  
AutoPatchingSettingMaintenanceWindowDuration で分単位の長さを指定します。この機能は 2027 年 9 月 17 日に廃止の予定です。新しい環境では使わず、Azure Update Manager を使います（既存の環境も移行が必要）。自動修正と Update Manager など、複数の仕組みで同じ VM を更新すると、更新が失敗するため、1 つだけに絞ります。

Arc 対応の SQL Server の更新は、Arc 側の自動更新を設定します[^c1-27]。動くのはホスト OS のレベルで、そのホスト上のすべての SQL Server インスタンスに適用されます。現時点では Windows ホストだけです。設定項目は 3 つです。

- Automatic updates: Enable か Disable
- Maintenance schedule: Daily か、曜日
- Maintenance start hour: 0:00 から 23:00 (ローカル時刻)

自動更新が入れるのも、Important または Critical の Windows と SQL Server の更新だけ (Arc 側も同じ範囲)。そのほかの累積更新などは、手動で適用します。

IaaS の VM に修正を入れる設問では、まず対象が「Important か Critical の更新だけか、累積更新まで含むか」を見ます。累積更新まで自動で入れたいなら、Azure Update Manager (Automated Patching は対象外)。

## スケーリングとパフォーマンスに優れたリソースを構成する [1.2]

### SQL Database のスケール [1.2.1]

SQL Database は、仮想コアの数やサービス レベルを、ダウンタイムの少ない操作で変えられます。ただし、切り替えの瞬間の短い接続の切断は避けられません。スケール アップでもスケール ダウンでも、データベース エンジンのプロセスが再起動され、必要なら別の仮想マシンへ移ります。移動中は既存のデータベースを使い続けられますが、新しいエンジンが準備できると、開いている接続が切られ、コミット前のトランザクションは巻き戻ります。再試行ロジックを入れておけば、アプリにはほとんど見えません[^c1-28]。

したがって、ピーク前にスケール アップするなら、アプリに一時エラー用の再試行があることを確かめてから行います。再試行がないアプリでは、操作の後にエラーが出ます。

サーバーレスの設定値は 3 つです。自動一時停止の遅延、最小仮想コア、最大仮想コアです[^c1-29]。

| 項目            | General Purpose | Hyperscale     |
|---------------|-----------------|----------------|
| 自動一時停止の遅延の最小  | 15 分            | 60 分           |
| 最大            | 10,080 分 (7 日)  | 10,080 分 (7 日) |
| 既定            | 60 分            | 60 分           |
| 自動一時停止を無効にする値 | -1              | -1             |

Hyperscale の自動一時停止と自動再開は、プレビューの機能です (General Purpose は一般提供)。使った計算は、最小と最大の仮想コアの範囲の中で

自動的に増減し、秒単位で課金されます。一時停止中は、ストレージ分だけの課金です。次のログインなどの活動で、自動的に再開します。再開には待ちが入るので、すぐ返事が要る業務には向きません。自動一時停止の遅延に -1 を指定すると、一時停止しないので再開の待ちは出ません。計算は最小と最大の範囲で増減し、秒単位で課金されます (止まらないので、使わない間の節約は無くなります)。

Hyperscale は、ストレージを計算から切り離れた構造です。構成要素は 4 つあります[^c1-30]。

- 計算ノード: プライマリと、高可用性セカンダリ。ローカル SSD に RBPEX というキャッシュを持つ
- ページ サーバー: データをストレージから取り出して SSD に載せ、エンジンへ渡す。各ページ サーバーが管理するのは最大 128 GB。ページ サーバーも SSD のキャッシュ (RBPEX) を持つ
- ログ サービス: トランザクション ログを、レプリカとページ サーバーへ配る
- Azure Storage: データ ファイルを長期に保存する

高可用性のセカンダリ レプリカは 0 から 4 個の範囲で選びます。高可用性のセカンダリも読み取りを受けられますが、計算サイズはプライマリと同じで、障害の時のフェールオーバー先も兼ねます。読み取りを用途ごとに切り離し、別の計算サイズを割り当てたいときは、名前付きレプリカを作ります。1 つのプライマリにつき最大 30 個で、それぞれ独立した計算サイズを持てます。名前付きレプリカは読み取り専用 (書き込みはプライマリだけが受ける)。データベースの最大サイズは 128 TB、エラスティック プールでは 100 TB です。作る時に最大データ サイズは指定しません。実際の割り当て量に応じて 10 GB から自動で伸び、その量に課金されます。

エラスティック プールで決める値は 2 つの層に分かれます。プール全体の上限 (eDTU または vCore の総量とストレージ) と、データベースごとの最小と最大です[^c1-31]。データベースごとの最大を、プールの合計より小さくしておくと、1 つのデータベースがプールを独占するのを防げます。ピークが重なるデータベースが多いと、プールの総量が足りなくなります。プールの総量は、「全データベースの平均使用量の合計」と「同時にピークになるデータベースの数 × ピーク時の使用量」の、大きいほうを目安に決めます (文書の式は 2 つの値の MAX)。

使う時間帯がばらばらな 40 個のデータベースを、1 つの予算で運用したい場面を考えます。単一データベースにすると、1 つずつがピークの大きさとで課金されます。エラスティック プールにまとめれば、課金はプールの総量で決まり、ピークの重なりが小さいほど小さい総量で足りる。1 つのデータベースがプールを占有しそうなら、データベースごとの最大を設定します (プールの総量は、平均の合計と、同時にピークになる分の合計の、大きいほうから決めます)。

DTU の使用率は、CPU とデータ I/O とログ書き込みの 3 つの割合のうち、最大のものです。

```
-- avg_dtu_percent
-- = MAX(avg_cpu_percent, avg_data_io_percent, avg_log_write_percent)
SELECT TOP (5) end_time, avg_cpu_percent, avg_data_io_percent,
avg_log_write_percent
FROM sys.dm_db_resource_stats
ORDER BY end_time DESC;
```

sys.dm\_db\_resource\_stats は直近 1 時間を、sys.resource\_stats は直近 14 日 間 を 5 分 平 均 で 見 せ ます 。プ ール に は sys.elastic\_pool\_resource\_stats を使います[^c1-32]。CPU が 30% でも、ログ書き込みが 90% なら、DTU は 90% 使われています。

ハードウェアを選びたい、ライセンスの特典を使いたい、サーバーレスや Hyperscale が要る、のどれかに当てはまれば vCore モデルです。単純にまとめて買いたいなら DTU モデルです。

### SQL Managed Instance のスケール [1.2.2]

Managed Instance のサービス レベルは 2 つで、General Purpose には新しい Next-gen General Purpose があります[^c1-33]。

| 項目              | General Purpose       | Next-gen General Purpose | Business Critical  |
|-----------------|-----------------------|--------------------------|--------------------|
| 最大 vCore        | 80                    | 128                      | 128                |
| 最大ストレージ         | 16 TB                 | 32 TB                    | 16 TB              |
| インスタンスあたりの DB 数 | 100                   | 500                      | 100                |
| 読み取り専用レプリカ      | 0                     | 0                        | 1 (無料の読み取りスケールアウト) |
| ストレージ           | Azure Blob (ページ BLOB) | Elastic SAN              | ローカル SSD           |
| ストレージ遅延         | 5~10 ms               | 改善                       | 平均 1~2 ms          |

Next-gen General Purpose は、予約ストレージ 1 GB あたり 3 IOPS を無料で付けます。メモリの量も柔軟に選べます。無料枠を超える IOPS には課金さ

れます。General Purpose と Business Critical では、IOPS に課金がありません。

Business Critical は、Always On 可用性グループに似た仕組みで、セカンダリが 3 つあります。別のリージョンへのフェールオーバーには、フェールオーバー グループを構成します[^c1-34]。

スケールや更新の操作は、種類によって停止の長さが違います。操作の途中のフェールオーバーで、インスタンスが使えなくなるのは、Business Critical で最大 20 秒、General Purpose で最大 2 分です[^c1-35]。

| 操作                                | 95% の操作が終わる時間 (General Purpose) |
|-----------------------------------|---------------------------------|
| ストレージのスケール                        | 99% が 5 分以内                     |
| 計算 (vCore) のスケール                  | 60 分                            |
| Business Critical への変更            | 60 分 + データベースのシード時間             |
| 基本プロパティ (ライセンスの種類や Entra 管理者) の変更 | 最大 1 分                          |

General Purpose では、データのシードがありません。リモート ストレージを古いプロセスから外して、新しいプロセスにつなぎ直すだけだからです。Business Critical はローカル SSD なので、ほとんどの操作でデータのコピー (シード) が走ります。General Purpose から Business Critical に上げる操作も、データをリモートからローカルへ移すので、シードの分だけ長くなります (表の 60 分に加算)。

Managed Instance プールは、2 vCore などの小さなインスタンスをまとめ、共有のリソースで動かして、コストを下げる仕組みです。プールの中のインスタ

ンスの作成は、95% が 10 分未満で終わります（プールを新しく作る時は 30 分）。

### SQL Server on Azure VM のサイズとディスク [1.2.3]

VM では、サイズ、ディスク、SQL Server の設定の 3 層を、自分で決めます。決める順番は固定で、先にストレージを構成し、IOPS とスループットを決めてから、VM のサイズを選びます。必要な IOPS とスループットには、20% の余裕を持たせます[^c1-36]。

VM サイズは、メモリ最適化のシリーズを選びます。ほとんどの本番では、Ebds v5 シリーズがコストと性能の釣り合いに優れます。M シリーズは、Azure で最大のメモリを持ち、最大級のワークロード向けです。迷ったら Ebds v5 です[^c1-37]。

ディスクは、種類、キャッシュ、配置の 3 点を決めます[^c1-38]。

| 項目             | 推奨                                                            |
|----------------|---------------------------------------------------------------|
| データ ファイルのディスク  | ホスト キャッシュは読み取り専用                                              |
| ログ ファイルのディスク   | ホスト キャッシュはなし                                                  |
| 読み取り/書き込みキャッシュ | データやログのディスクでは有効にしない                                           |
| データドライブ        | キャッシュが使える Premium の P30、P40 以下のディスク。Ebds v5 では Premium SSD v2 |
| 複数ディスク         | Storage Spaces でストライプして、IOPS とスループットを合算する                     |
| 超高 IOPS        | 160,000 IOPS を超えるなら Premium SSD v2 か Ultra Disk               |

| 項目      | 推奨                                            |
|---------|-----------------------------------------------|
| ログの低遅延  | サブミリ秒が必要なら Premium SSD v2 か Ultra Disk        |
| 割り当て単位  | データ ファイル用は 64 KB でフォーマットする (一時 D:\ の既定は 4 KB) |
| ドライブの分離 | データ、ログ、tempdb を別のドライブに置く                      |

キャッシュの設定を変えると、ディスクが外されて、つなぎ直されます。変更する前に、SQL Server のサービスを止めます。止めずに変えてはいけません。データが壊れる恐れがあるからです。

tempdb は、ローカル SSD の一時ディスク (既定は D:\) に置きます。一時ディスクは、VM の再起動やホスト移動で作り直されます。再プロビジョニングの時に必要なフォルダーと権限は、SQL Server IaaS Agent 拡張機能が整えます (手作業は不要)。フェールオーバー クラスタ インスタンスでは、共有ストレージに置きます。ファイルの数は、コアあたり 1 つから始めて、最大 8 個です。デプロイ時に指定できる最大数も 8 (追加は展開後)。

チェックリストの設定は 4 つです。ページ圧縮、バックアップ圧縮、データ ファイルの即時ファイル初期化 (IFI)、メモリ内のページのロックを有効にします [^c1-39]。

Automated Backup は、SQL Server 2016 以降の Standard、Enterprise、Developer で使えます。保持期間は 1~90 日で、既定は 90 日です。この日数は、msdb でメタデータが保持される期間で、ストレージ上のファイルは消えません。対象のユーザー データベースは、完全復旧モデルである必要があります [^c1-40]。

## パーティションテーブルの作成 [1.2.4]

パーティション テーブルは、関数、スキーム、テーブルの順に作ります[^c1-41]。

```
CREATE PARTITION FUNCTION myRangePF1 (datetime2(0))
    AS RANGE RIGHT FOR VALUES ('2022-04-01', '2022-05-01', '2022-06-01');

CREATE PARTITION SCHEME myRangePS1
    AS PARTITION myRangePF1 ALL TO ([PRIMARY]);

CREATE TABLE PartitionTable (col1 datetime2(0) PRIMARY KEY, col2
    char(10))
    ON myRangePS1 (col1);
```

境界値が3つなので、パーティションは4つできます。RIGHT なので、4月1日の行は2番目のパーティションに入ります。1番目は4月1日より前、4番目は6月1日以降です (RIGHT なので境界値が下限)。

スキームの書式は2通り。ALL TO は、全パーティションを1つのファイルグループに割り当てます。ALL を付けた時に指定できるファイルグループは1つだけで、PRIMARY は [PRIMARY] と区切る (キーワードのため)。ALL を付けずにファイルグループを並べると、パーティション1から順に割り当てられます。関数のパーティション数より多くのファイルグループを書くと、最初に余った1つが NEXT USED になり、次の SPLIT RANGE で増えるパーティションが入る先になります。

```
-- 7 月 1 日の境界を足す (SPLIT RANGE)
```

```
ALTER PARTITION SCHEME myRangePS1 NEXT USED [PRIMARY];
```

```
ALTER PARTITION FUNCTION myRangePF1() SPLIT RANGE ('2022-07-01');
```

SPLIT RANGE は、パーティション関数に新しい境界値を足して、パーティションを割る操作です (ALTER PARTITION FUNCTION に付く句)。逆に、境界値を消してパーティションを併合するのは MERGE RANGE です (切り替えの SWITCH は ALTER TABLE の側)。ALL TO で作ったスキームの NEXT USED の扱いは、文書によって書き方が違います。CREATE PARTITION SCHEME のページは「ALL で指定したファイルグループは NEXT USED を保つ」と書き、ALTER PARTITION FUNCTION のページは「1 回 SPLIT した後は NEXT USED が無くなり、指定し直さないと次の SPLIT は失敗する」と書きます。SPLIT の前に毎回 ALTER PARTITION SCHEME で NEXT USED を指定しておけば、どちらでも通ります。ALL を付けずにファイルグループを並べた書式では、余りのファイルグループは最初の SPLIT RANGE で使われます。2 回目の SPLIT RANGE の前には、NEXT USED の指定し直しが必要。

インデックスは、テーブルと同じスキームで作って整列させる (aligned)。整列していれば、切り替えを効率よく行えます。

```
ALTER TABLE PartitionTable SWITCH PARTITION 1 TO PartitionArchive;
```

注文テーブルを月ごとに分け、24 か月より古い月を毎月切り離す運用を考えます。境界値は各月の 1 日にして、RANGE RIGHT で作ります。月初の行が、その月のパーティションに入ります (LEFT だと前の月に入ってしまいます)。RANGE LEFT では、境界値がその左側のパーティションの上限になります。境界値と同じ値の行は、RIGHT の場合より 1 つ前のパーティションに入ります。

新しい月は、ALTER PARTITION SCHEME で NEXT USED を指定してから、SPLIT RANGE で足します。古い月は、アーカイブ用の空のテーブルへ SWITCH で出します（移動先は存在して空で、移動元と同じファイルグループにあり、対応するインデックスも同じファイルグループにある必要があります）。条件を欠くと、切り替えは通りません。インデックスを整列させておけば、出し入れは効率よく進みます。

## データ圧縮の構成 [1.2.5]

圧縮は、ロースタ（行ストア）と列ストアで種類が違います。ロースタは行圧縮とページ圧縮です。列ストアは列ストア圧縮と列ストア アーカイブ圧縮です。列ストアのテーブルは、常に列ストア圧縮を使い、これは切り替えられません。さらに小さくしたいときだけ、アーカイブ圧縮を足します。時間と CPU が余分にかかりますが、あまり読まれない古いデータに向きます[^c1-42]。

行圧縮とページ圧縮は、パーティション単位で ROW、PAGE、NONE を設定できます。作成時の既定は NONE です。

```
ALTER TABLE dbo.Orders REBUILD PARTITION = ALL WITH (DATA_COMPRESSION = PAGE);  
  
ALTER INDEX ix_Orders_Date ON dbo.Orders REBUILD WITH (DATA_COMPRESSION = ROW);
```

圧縮の設定は REBUILD で行います。REORGANIZE には DATA\_COMPRESSION を付けられません。

非クラスター化インデックスは、テーブルの圧縮設定を引き継ぎません。テーブルに整列していないインデックスがあると、1つのパーティションだけの圧縮設定は変えられません。インデックスごとに、別に設定します。引き継がれなく

て既定が NONE のままなので、テーブルを圧縮しただけでは、インデックスは小さくなりません。

圧縮できない場合もあります。最大行サイズ 8,060 バイトを超える場合は、圧縮できません。行の外にあるデータ(大きな XML など)は、圧縮されません。ページ圧縮を設定したヒープでは、INSERT INTO ... WITH (TABLOCK) で入れたデータ(非クラスター化インデックスが無い時)か、一括最適化を有効にした一括インポートか、ALTER TABLE ... REBUILD に PAGE を付けた場合にだけ、ページ圧縮されます。

効果は、実行前に見積もれます。

```
EXEC sp_estimate_data_compression_savings
    @schema_name = 'dbo', @object_name = 'Orders',
    @index_id = NULL, @partition_number = NULL, @data_compression =
    'PAGE';
```

現在のサイズと、指定した圧縮状態での見積もりサイズが返ります[^c1-43]。この手続きは Azure SQL Database と Managed Instance でも使えます。

I/O の多いワークロードでは、読むページが減って速くなります。そのかわり、圧縮と展開に CPU が余分に要ります。CPU が限界に近い環境でページ圧縮を足しても、速くはなりません。

## 移行戦略を計画して実装する [1.3]

### 移行の要件を評価する [1.3.1]

SQL Server から Azure SQL への移行は、5 つのフェーズで進みます。検出 (Discovery)、ビジネス ケース、評価 (Assessment)、変換 (Conversion)、移

行 (Migration) です[^c1-44]。

検出は、Azure Migrate アプライアンスが担います。サーバーに何もインストールしないエージェントレスの方式で、オンプレミスの SQL Server のインスタンスとデータベースを見つけます。対象は VMware、Hyper-V、物理環境。評価のサイズ設定の基準は 2 つあります。As on-premises は、オンプレミスの構成だけを見て、同じ大きさの Azure SQL を推奨します。Performance-based は、収集した性能データを見て、実際に使っている分に合わせて推奨します。サイズを小さく見積もりたいときや、過剰な構成を避けたいときは、Performance-based を選びます (実測に合わせて推奨が決まる)。

SQL Server から Azure SQL への移行では、変換のフェーズは要りません。SQL Server Migration Assistant (SSMA) が、移行の中で変換を行います。SSMA は、Oracle、MySQL、DB2、Sybase、Microsoft Access のような別のデータベース製品が元でも、変換とデータ移行ができます。

評価の結果、SQL Managed Instance への移行を止める要因 (移行ブロッカー) が複数見つかったら、SQL Server on Azure Virtual Machines を代替りの候補にします。VM は、オンプレミスと同じ環境をそのまま再現できるためです (互換性の差が出ません)。

廃止された道具が 2 つあります。Data Migration Assistant (DMA) は 2025 年 7 月 16 日に廃止されました。もともと Managed Instance への移行には対応していません。Azure Data Studio の Azure SQL 移行拡張機能は、Azure Data Studio とともに 2026 年 2 月 28 日に廃止されました[^c1-45]。評価と移行の道具の選択肢に、この 2 つは出てきません。SQL Server enabled by Azure Arc を使えば、Azure への移行評価が自動で作られます (既定で週 1 回)。

### オフライン(offline)移行とオンライン(online)移行の比較 [1.3.2]

オフライン移行は、移行の間アプリを止めます。オンライン移行は、ほとんど止めずに、最後に切り替える時だけ短く止めます。止められる時間が、方式を決めます[^c1-46]。

Azure Database Migration Service (DMS) の Azure portal 版は、移行先ごとに対応する方式が違います。

| 移行先                    | オフライン | オンライン(最小ダウンタイム) |
|------------------------|-------|-----------------|
| SQL Database           | 対応    | なし              |
| SQL Managed Instance   | 対応    | 対応              |
| SQL Server on Azure VM | 対応    | 対応              |

この版の DMS は、評価と SKU の推奨、ログインの移行は行いません。スキーマの移行は行います。

Managed Instance link は、Business Critical への、唯一の真のオンライン移行で、ダウンタイムは最小です。ネイティブの RESTORE DATABASE FROM URL は、SQL Server のバックアップを使う方式で、ある程度のダウンタイムが要ります。

SQL Database に、止めずに近い形で移したいなら、SQL Server のトランザクション レプリケーションを使います[^c1-47]。SQL Database をサブスクライバーに設定し、同期が終わったら、アプリの接続文字列を切り替えます。対象は、本番から外せないデータベース。同期中も新しいトランザクションは流れ続けます。

Log Replay Service (LRS) は、Managed Instance へ、バックアップを継続して復元する仕組みです。2 つのモードで始められます。autocomplete モードは、バックアップ チェーン全体を、先にアップロードします。continuous モードは、新しいバックアップを継続して取り込み、手動でカットオーバーを要求して完了します[^c1-48]。

### オンライン移行の実装 [1.3.3]

DMS の Managed Instance 向けのオンライン移行は、Azure portal から、移行元と、バックアップを置く場所と、移行モードを選んで始めます[^c1-49]。

Managed Instance link による移行は、3 つの手順です。環境を準備し、SSMS かスクリプトでリンクを構成し、リンクをフェールオーバーします。技術は Always On 可用性グループで、プライマリの SQL Server からセカンダリの Managed Instance へ、ほぼリアルタイムに変更を送ります。フェールオーバー（カットオーバー）の挙動は、バージョンで違います[^c1-50]。SQL Server 2016～2019 からのカットオーバーは、リンクを切断します。SQL Server 2022 以降は、フェールオーバー時にリンクを維持するか切断するかを選べます。移行では、カットオーバーの後にリンクを削除することが推奨されます。

LRS の continuous モードは、バックアップのフォルダーを走査して、新しいファイルを次々に復元します。手動でカットオーバーを要求するまで、完了しません。バックアップ チェーン全体が、事前にそろっていないときに使います。

LRS で復元中のデータベースは、移行が完了するまで使えません。読み取り専用のアクセスもできません。Business Critical へ LRS で移すと、カットオーバーの後に、セカンダリへのシードで、長いあいだデータベースが使えなくなります。大きなデータベースで停止を短くしたいときは、先に General Purpose へ移し、後で Business Critical に上げるか、Managed Instance link で移します。

## オフライン移行の実装 [1.3.4]

LRS の autocomplete モードは、PowerShell か CLI で始めます。最後のバックアップ ファイル名を -LastBackupName で指定し、その最後のファイルの復元が終わると、自動でカットオーバーします[^c1-51]。

```
Start-AzSqlInstanceDatabaseLogReplay -ResourceGroupName  
"ResourceGroup01" `
    -InstanceName "ManagedInstance01" -Name "ManagedDatabase01" `
    -Collation "SQL_Latin1_General_CP1_CI_AS" `
    -StorageContainerUri  
"https://<account>.blob.core.windows.net/<container>" `
    -StorageContainerSasToken "<token>" `
    -AutoCompleteRestore -LastBackupName "last_backup.bak"
```

autocomplete には 2 つの制約があります。バックアップ チェーン全体を事前にアップロードする必要があり、進行中に新しいバックアップを足せません。指定した最後のファイルより後にファイルがあると、エラーになります。

LRS の前提条件と推奨は、次の 4 つです。

- Blob コンテナーへの Read と List だけを許可した SAS トークン、または同じ権限のマネージド ID。権限が多すぎると LRS は失敗する
- バックアップは、データベースごとに別のフォルダーに置く。フォルダーの中はフラットな構造で、入れ子は使えない
- バックアップは CHECKSUM 付きで取る（必須ではないが、強く推奨）。破損したデータベースを移さずに済み、復元も速くなる
- 1 つの LRS 移行ジョブは、最大 30 日以内に完了する。期限が切れると自動でキャンセルされる

バックアップのファイルの数が多すぎると、移行時間が延びて、失敗することがあるため、ファイルは統合します。メンテナンス更新があると、General Purpose では、保留中の LRS 移行がいったん止まって再開されます。Business Critical では、キャンセルされてから自動で再起動されます(やり直し)。大きなデータベースでは、メンテナンス ウィンドウを設定して、移行と更新が重ならないようにします。

バックアップと復元 (URL) の移行は、バックアップを Blob に置き、RESTORE DATABASE ... FROM URL で復元します。SSMS では、Restore Database でコンテナーと共有アクセス署名 (SAS) を指定します。Managed Instance 側に、SQL 資格情報が自動で作られます[^c1-52]。

BACPAC は、スキーマとデータをまとめたファイルです。インポートは SqlPackage で行います。

```
SqlPackage /a:Import /sf:<file>.bacpac /tdn:<db> /tsn:
<server>.database.windows.net
```

| 移行先                  | Azure portal                       | PowerShell                  | SSMS | SqlPackage |
|----------------------|------------------------------------|-----------------------------|------|------------|
| SQL Database         | 単一データベースのみ。標準ストレージの Blob 上の BACPAC | New-AzSqlDatabaseImport で対応 | 対応   | 対応         |
| SQL Managed Instance | 非対応                                | 非対応                         | 対応   | 対応         |

150 GB を超える BACPAC は、ポータルや PowerShell で失敗することがあります。大規模なデータと本番の移行には、SqlPackage が推奨です。BACPAC

のファイルは、移行先のデータセンターの近くに置きます[^c1-53]。インポートを速くしたいときは、インポートの間だけ、サービス レベルとコンピューティング サイズを上げて、成功した後で下げます。

bcp は、テーブルのデータを、ファイルに一括して出し入れするコマンド ライン ツールです。スキーマは移りません。BACPAC はデータベース単位、bcp はテーブル単位で使います。

### Azure への移行の実行 [1.3.5]

移行は、前、中、後の 3 つに分けて考えます。

- 前: 互換性の評価と、移行先の SKU の選定
- 中: 方式に応じたデータの移動と、アプリの切り替え
- 後: アプリの動作と性能の確認

SQL Server の機能の一部は、データベースの互換性レベルを最新に変更するまで、使えません。移行しただけでは、互換性レベルは上がりません。

| 移行先                    | 主な方法                                                                  |
|------------------------|-----------------------------------------------------------------------|
| SQL Server on Azure VM | バックアップと復元、URL からのデタッチ/アタッチ、VM に変換してアップロード、ログ配布                        |
| SQL Managed Instance   | Managed Instance link、Log Replay Service、ネイティブ RESTORE FROM URL の 3 つ |
| SQL Database           | オフラインは DMS か BACPAC。最小ダウンタイムはトランザクション レプリケーション                        |

SQL Database への DMS は、オフラインです[^c1-54]。止められない場合は、ここで BACPAC や DMS を選んではいけません。トランザクション レプリケー

ションです。

VM は、オンプレミスの SQL Server と環境が完全に一致します。VM への移行は、オンプレミスのサーバーから別のサーバーへ移すのと、ほとんど変わりません。Managed Instance は、移行の前に、いくつかの変更が要ることがあります。

オンプレミスの大きなデータベースを、停止を最小にして Business Critical の Managed Instance へ移す場面を考えます。LRS で移すと、カットオーバーの後にセカンダリへのシードが走り、長いあいだデータベースが使えません。ネイティブの RESTORE は、ある程度の停止が要ります。Business Critical への唯一の真のオンライン移行は Managed Instance link なので、これが答えです。止められる時間が長いなら、LRS の autocomplete やネイティブの RESTORE でも足ります。

## Azure SQL サービス間の移行 [1.3.6]

SQL Database のデータベース コピーは、トランザクション一貫性のあるスナップショットを、同じ論理サーバーか、別の論理サーバーに作ります[^c1-55]。ポータル、PowerShell、Azure CLI、T-SQL で実行できます。

```
CREATE DATABASE Database2 AS COPY OF server1.Database1;
```

PowerShell のコマンドは New-AzSqlDatabaseCopy、CLI は az sql db copy。

コピーはジオレプリケーションの技術で作られ、シードが終わるとリンクが自動で切れます。コピーの操作は非同期ですが、対象のデータベースは、すぐに作られます。進行中にやめるには、対象のデータベースを Remove-AzSqlDatabaseなどで削除します。Hyperscale のデータベースでは、同じリ

ージョン内は高速なコピー (BLOB のスナップショット)。リージョンが違う場合や、バックアップの冗長性が違う場合は、データ量に応じて時間がかかる低速なコピーになります。ポータル、PowerShell、Azure CLI は、別のサブスクリプションへのコピーに対応しません。T-SQLで行います。

所有者は、同じサーバーなら、実行したセキュリティ プリンシパルです。別のサーバーなら、対象のサーバーで操作を始めたプリンシパルです。データベース ユーザー、権限、SID はコピーされます。ログインは移りません。含まれたデータベース ユーザーを使っていれば、コピーの後も、すぐ同じ資格情報で入れます。サーバー レベルのログインでアクセスしていると、別のサーバーへのコピーの後に、入れなくなることがあります。

既存のデータベースを Hyperscale に変換することもできます。ポータル、CLI、PowerShell、T-SQL のどれでも行えます。Basic からは、直接変換できません。まず Basic 以外のレベル (General Purpose など) に変えてから変換します[^c1-56]。geo レプリケーションのデータベースは、プライマリを先に変換します。geo セカンダリは、自動で変換されます。セカンダリを直接変換することはできません。geo レプリカがチェーン構成なら、変換の前に解除します。

SQL Database と Managed Instance のあいだは、BACPAC のエクスポートとインポートで移せます。インポート先が Managed Instance なら、SSMS か SqlPackage のみ。同じ SQL Database の中でサービス レベルを変えること (General Purpose から Business Critical など) は、移行ではなくスケールの操作です。

## Managed Instance のデータベースのコピーと移動 [1.3.7]

Managed Instance のあいだで、データベースを、オンラインでコピーまたは移動できます (Always On 可用性グループの技術を使います)。コピーは、ソ

ースが残ります。移動は、操作が完了した後に、ソースが削除されます[^c1-57]。

操作を始めると、シードが進みます。終わると、状態が「準備完了」になります。完了させるまで、ソースの変更が宛先へ反映され続けます。手動で Complete を実行すると、宛先が読み書きできる状態になります(この操作が最後の手順)。

```
Copy-AzSqlInstanceDatabase -ResourceGroupName $rg -InstanceName $mi -  
Name $db `   
    -TargetInstanceName $tmi -TargetResourceGroupName $trg  
Get-AzSqlInstanceDatabaseCopyOperation -ResourceGroupName $rg -  
InstanceName $mi  
Complete-AzSqlInstanceDatabaseCopy -DatabaseName $db -InstanceName $mi  
`  
    -ResourceGroupName $rg -TargetInstanceName $tmi -  
TargetResourceGroupName $trg
```

シードが終わってから 24 時間以内に完了しないと、操作は自動でキャンセルされ、宛先のデータベースは削除されます。

別のサブスクリプションへは、-TargetSubscriptionId を使います(Az.Sql 5.1 以降)。インスタンスが別の仮想ネットワークにあるなら、ネットワーク接続が必要で、両方のサブネットで、TCP のポート 5022 と、ポート範囲 11000～11999 の、受信と送信を許可します。必要な権限は、ソースのインスタンスを含むリソース グループへの読み取りと、ソースと宛先の、両方のデータベースレベルの書き込みです。

主な制限は次のとおりです（ほかに、ソースや宛先のインスタンスにフェールオーバーグループを構成していないこと、宛先の更新ポリシーがソースと同じか新しいこと、などがあります）。

- 同じインスタンスの中では行えない。同一リージョンの中だけ
- ユーザー データベースだけが対象
- 1つのデータベースが同時に参加できる操作は1つだけ。1つのソースインスタンスが同時に動かせるのは8操作までで、超えた分は待たされる
- フェールオーバーグループや Managed Instance link を使っているデータベースは対象外
- 操作の途中で名前の変更はできない。データベースのタグもコピーされない
- ポイントインタイム リストア (PITR) のバックアップは、引き継がれない。宛先で新しいバックアップチェーンが始まる

コピーと PITR は別のものです。PITR は、過去の指定した時点の状態から、データベースを作ります。コピーと移動は、操作が完了した時点の状態から、作ります。

### 移行のトラブルシューティング [1.3.8]

DMS (classic) の SQL Server 向けのシナリオは、2026 年 3 月 15 日に廃止されました。今の移行で起きる問題は、Azure portal 版の DMS、Log Replay Service (LRS)、Managed Instance link の 3 つの仕組みごとに切り分けま

す。  
Azure portal 版の DMS で起きやすい問題は、バックアップの読み込みとカットオーバーです[^c1-58]。

| 症状                                         | 原因と対処                                                                                         |
|--------------------------------------------|-----------------------------------------------------------------------------------------------|
| バックアップがオンプレミスの SMB ネットワーク共有にあり、DMS が読み込めない | 移行元の SQL Server とバックアップの場所の両方に接続できるマシンに、セルフホステッド統合ランタイムをインストールする。バックアップが Azure Blob にあるなら要らない |
| カットオーバーの後に、最後の更新が移行先にならない                  | カットオーバーの手順の順序を守る (次の 5 つ)                                                                     |

DMS のオンライン移行のカットオーバーは、次の順に進めます。

1. 移行元のデータベースへの、新しいトランザクションを止める
2. アプリの接続先を、移行先の SQL Managed Instance に変える
3. 末尾のログ バックアップを、DMS に指定したバックアップの場所に取り
4. 監視の詳細ページで、すべてのバックアップの状態が Restored になったのを確かめる
5. Complete cutover を選ぶ

LRS で起きやすい問題は、バックアップに CHECKSUM を付けていないことと、バックアップのファイル数が多すぎること、SAS トークンの権限が広すぎることです。権限は Read と List だけにします。進まないときは、更新の時期と重なっていないかも見ます。LRS が始まらない時は、次の点も確かめます [^c1-48]。

- 移行先に、同じ名前のデータベースがない
- SAS トークンは、? の後ろの部分を渡している。有効期間が、移行の期間を覆っている

- バックアップの URI のパスに、backup や backups の語を含めていない  
(予約語なので、含むならコンテナやフォルダーの名前を変える)
- マネージド ID を使うなら、SQL Managed Instance に Storage Blob Data Reader (Read と List を含むカスタム ロールでもよい) を割り当てる。SAS トークンとマネージド ID は、同じストレージ アカウントで併用できない

Managed Instance link で起きやすい問題は、ネットワークとログの切り捨てです[^c1-60][^c1-61]。

| 症状                              | 原因と対処                                                                                                                                                                                                                     |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| リンクを作れない(接続の問題)                 | SQL Managed Instance のサブネットの NSG で、SQL Server の IP アドレスからの受信をポート 5022 とポート範囲 11000～11999 で、送信を 5022 で許可する。SQL Server 側は、ホスト OS を含むすべてのファイアウォールで、受信を 5022、送信を 5022 と 11000～11999 で許可する。SQL Managed Instance 側のポート番号は変えられない |
| 初期シードの途中で、エラー 1412 か 1408 で失敗する | 初期シードの間にログ バックアップが走り、ログが切り捨てられた。リンクを削除して作り直す。シードの間はログ バックアップを止める(対応するビルドではトレース フラグ 12381 で切り捨てを防ぐ)                                                                                                                        |
| 初期シードの進み具合を知りたい                 | sys.dm_hadr_physical_seeding_stats で見る                                                                                                                                                                                    |
| リンクの証明書の期限が切れて、リンクが失敗した         | 証明書をローテーションする                                                                                                                                                                                                             |

| 症状                              | 原因と対処                                                                                                                    |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| 強制フェールオーバーの後に、両方のレプリカがプライマリになった | 元のプライマリでバックアップを取る (SQL Server なら末尾のログ バックアップ、SQL Managed Instance ならコピーのみの完全バックアップ)。そのうえで、元のプライマリ側の分散可用性グループをセカンダリの役割にする |

ポート 1433 はクライアントの接続、UDP 1434 は名前付きインスタンスを探す SQL Server Browser のポートで、どちらもリンクの通信には使いません。

症状ごとに、見る場所は 3 つあります。

- 互換性のエラー: 評価の結果に出た移行ブロッカーから探す
- 接続の問題: ファイアウォールとポート、仮想ネットワークの設定を見る
- 性能の問題: 移行後に、互換性レベルが古いままになっていないかを見る

## サンプルはここまでです

---

続き(第2章～巻末)は、DP-300 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

SAMPLE

---

## DP-300 学習用テキスト(無料サンプル)

### 版

2026年10月版(DP-300 出題範囲(Skills measured)2026-10-27版 準拠)

### 発行

IT資格道場(<https://www.shikaku-doujou.com>)

### お問い合わせ

[info@shikaku-doujou.com](mailto:info@shikaku-doujou.com)

Microsoft、Azure、SQL Server および Microsoft の製品名・サービス名は、Microsoft Corporation の米国およびその他の国における商標または登録商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は公式シラバスの内容を基に作成したもので、Microsoft Corporation との提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点は [info@shikaku-doujou.com](mailto:info@shikaku-doujou.com) までご連絡ください。