IT資格道場

SC-500 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)

SC-500 出題範囲(Study guide・Last updated 2026-05-13)準拠
2026年7月版

本書のご利用にあたって

- 本書は「IT資格道場」(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。
- 無料サンプルであっても著作物です。本書の転載・改変しての再配布・二次販売を禁じます。
- 本書を AI モデルの学習データとして利用すること(学習目的の入力・データセット化を含む)を禁じます。
- 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。

不正な配布を見つけた場合は info@shikaku-doujou.com までご連絡ください。

目次

この教材群の使い方(3点セット)

種類	役割
① 学習用テキスト(本書)	これだけで問題が解けるようになる本編
② 頻出度別ノート	テキストを読んだら次はこれ。頻出順に絞った問題と解説で「解ける」に橋渡し。試験直前の最終チェックにも
③ 暗記用ノート	覚えるべき要点だけを一問一答に凝縮。空き時間の反復と用語の再確認用

使い方: ① 学習用を読む → ② 頻出度別で解き方を掴む → ③ 問題演習で腕試し → ④ 頻出度別に戻って再確認(試験直前もここ)。暗記用は空き時間や用語の再確認に。

このテキストの位置づけ: 本書は①の学習用テキストです。まずはここを通読し、これだけで問題が解けるようになることを目標にしてください。

はじめに ―― この教材の使い方

このテキストは、Microsoft が実施する **Exam SC-500: Implementing End-to-End Security Controls for Cloud and AI Workloads**（認定名 **Microsoft Certified: Cloud and AI Security Engineer Associate**）に合格することを目的に作られています。

Associate 級が問うのは「Defender for Cloud とは何か」ではなく、**与えられた要件（誰に・何を・どの経路で許すか、何を監視し、どこで止めるか）のもとで、どの機能をどう構成すれば満たせるか**です。公式の Study guide は 4 つの主題（ID・アクセス・ガバナンスの管理／ストレージ・データベース・ネットワ

ークの保護／コンピューティングの保護／セキュリティ態勢の管理と監視）から成り、本書はその順番どおりに第1章〜第4章を置いています。AI ワークロードの保護（Purview DSPM for AI・Copilot Studio・Microsoft Entra Agent ID・Microsoft Foundry の AI Gateway・Defender for AI Services）と、Microsoft Sentinel・Microsoft Security Copilot の実装は、この試験で新しく前面に出た領域です。

本書は Microsoft の公式教材ではなく、IT資格道場が独自に書き下ろしたオリジナルの教材です。実際に出題された問題を再現したものではありません。

試験の基本情報

項目	内容
試験名	Exam SC-500: Implementing End-to-End Security Controls for Cloud and AI Workloads
認定	Microsoft Certified: Cloud and AI Security Engineer Associate
運営	Microsoft（Pearson VUE のテストセンター、またはオンライン監督試験）
試験時間	**120分**
合格ライン	**700 / 1,000**
受験言語	**英語のみ**（2026年9月時点。日本語版は提供されていません）

Microsoft は問題数と出題形式の内訳を公表していません（「対話型の要素を含むことがある」とだけ案内しています）。**受験料・試験時間・出題範囲は改**

定されることがあります。お申し込みの前に、Microsoft Learn の公式ページで最新の情報をご確認ください。

試験は英語で出題されます —— 本書の書き方

本書は日本語で書いていますが、**サービス名・機能名・設定項目名は英語表記のまま**にしています（例：Conditional Access、Privileged Identity Management (PIM)、Just-in-time (JIT) VM access、Data collection rules (DCR)）。本番の設問文・選択肢に出るのはこの英語表記であり、日本語訳を覚えても画面には出てきません。本文を読みながら、英語の名前を見た瞬間に「どの主題の、どの場面の機能か」が出る状態を作ってください。

なお、公式 Study guide は「希望する言語で試験が提供されていない場合は、追加で 30 分を申請できる」と案内しています（「If the exam isn't available in your preferred language, you can request an additional 30 minutes to complete the exam.」）。SC-500 は英語のみの提供なので、日本語を希望する受験者はこの延長を申請できます。申請の手順と適用条件は、お申し込み時に Microsoft Learn の公式ページでご確認ください。

出題範囲の全体像 —— 4つの主題と本書の章

主題	分野	採点対象に占める割合（公式）	本書
1	ID、アクセス、ガバナンスの管理	**20–25%**	第1章
2	ストレージ、データベース、ネットワークの保護	**25–30%**	第2章

主題	分野	採点対象に占める割合（公式）	本書
3	コンピューティングの保護（AI・サーバー/VM・アプリ基盤）	**20–25%**	第3章
4	セキュリティ態勢の管理と監視（Defender for Cloud・Sentinel・Security Copilot）	**20–25%**	第4章

本書の章の並びは、Study guide の並びとまったく同じです。節見出しの末尾にある `[1.1]` のような番号は、Study guide の到達目標（Skills measured）の番号です。Microsoft は主題ごとの割合を幅で公表しており、到達目標単位の出題数は公表していません。本書もそれ以上の数字は書きません。

全設問に効く判断軸 ——「3つの問い」

問い	何を切り分けるか	分かれ方
問い1: どの層の話か	ID／データ／ネットワーク／コンピューティング／態勢	同じ「アクセスを制限する」でも、Conditional Access・Storage firewall・NSG・JIT は層が違い、要件がどの層を指しているかで答えが決まります
問い2: 予防か、検知か、対応か	構成で止める／Defender・Sentinel で見つける／プレイブックで動く	「侵害を検知したい」に構成系の機能を選ぶ肢は誤答です

問い	何を切り分けるか	分かれ方
問い3: 最小権限・最小露出で満たせているか	スコープ・期間・経路	要件を満たす案が複数あるとき、正解はより狭いスコープ・より短い期間・より閉じた経路のものです

4ステップ学習法

購入者限定テキストは3冊で1セットです。

- **STEP 1（理解する）**: 本書「学習用テキスト」を通読し、4つの主題の地図と3つの問いを頭に入れる
- **STEP 2（解き方を掴む）**: 「頻出度別ノート」で頻出度の高い論点を解いて、解き方を掴む
- **STEP 3（腕試し）**: 本サイトの問題演習で、択一・問題と解決策（はい/いいえ）・組み立て式に慣れる
- **STEP 4（再確認）**: 「頻出度別ノート」に戻って総ざらい。試験直前もここへ戻る

「暗記用テキスト」は本線には入れません。通勤時間や休憩時間など、**空き時間の反復と用語の再確認**に並走させてください。

それでは、第1章から始めます。

第1章 ID、アクセス、ガバナンスの管理(ドメイン1・採点対象の 20–25%)

この章は SC-500 の土台になる分野です。ここで問われるのは「Microsoft Entra ID とは何か」ではなく、**与えられた要件(誰に・何を・どの期間・どの経路で許すか、監査に何を残すか、事故が起きたときに何で戻すか)のもとで、どの構成を選ぶと要件を満たすか**という判断です。

この試験の設問は「説明する」ではなく「**実装・構成する**」粒度で書かれています。したがって本章では、機能ごとに次の4つをそろえて覚えます。

1. **何ができるか**(機能と設定項目の名前)
2. **何ができないか・制約は何か**(誤答肢はここを突いてきます)
3. **似た機能との分かれ目**(どの条件で選択が切り替わるか)
4. **設定の順序と依存関係**(並べ替え問題・対応付け問題の根拠になります)

出題形式は択一・複数選択に加えて、同じシナリオを3回繰り返して「この解決策で目標を満たしますか、はい/いいえ」を問う形式、機能と用途の**対応付け**、手順の**並べ替え**があります。長文シナリオの最後の1〜2文に「最小の管理負荷で」「最小権限で」「他の管理者の作業を止めずに」といった評価軸が置かれ、そこが正解を一意に決めます。**技術的に正しい選択肢が複数並ぶのが普通で、評価軸に照らして最良のものだけが正解**です。

このドメインは公式に **20–25%** の幅で示されており、4つの主題の中で 2 番目に重い位置にあります。試験時間は 120 分、合格ラインは 1000 点満点で **700 点**です。

1-1 Microsoft Entra ID を使用してリソースへのアクセスをセキュリティで保護する [1.1]

この到達目標は「**誰が、いつ、どの条件で、どの権限を持つか**」を設計する範囲です。扱うのは特権の時間制御(PIM)、アクセスの条件付け(Conditional Access)、本人確認の強度(authentication methods)、アプリケーションのID(enterprise applications と app registrations)、アプリに与える同意(OAuth permission grants と consent settings)、そして資格情報を持たないID(managed identities)の6本です。

この6本は独立していません。**PIM の昇格時に Conditional Access の authentication context を要求し、その Conditional Access が authentication strength で passwordless を要求する**、というように鎖でつながります。この鎖を先に頭に入れておくと、シナリオの読み違いが減ります。

1-1-1 Privileged Identity Management (PIM) の実装と構成 [1.1]

Privileged Identity Management (PIM) は、Microsoft Entra ID の機能で、重要なリソースへのアクセスを管理・制御・監視します。対象は Microsoft Entra ID のリソース、Azure のリソース、そして Microsoft 365 や Microsoft Intune といった他の Microsoft オンライン サービスです。PIM の利用には**ライセンスが必要**です。

PIM の核は「**時間ベースと承認ベースのロールのアクティブ化**」です。常時付与された過剰な権限をやめ、必要なときだけ昇格させる仕組みを提供します。

PIM が提供する主な機能は次のとおりです。試験では機能名がそのまま選択肢に並びます。

機能	何をするか
just-in-time (JIT) 特権アクセス	Microsoft Entra ID と Azure リソースへの権限を、必要なときだけ一時的に付与する
time-bound アクセス	開始日と終了日を指定して割り当てる
approval の要求	特権ロールのアクティブ化に承認を必須にする
multifactor authentication の強制	ロールのアクティブ化時に MFA を要求する
justification(理由)の要求	なぜアクティブ化したのかを記録させる
notifications	特権ロールがアクティブ化されたときに通知を出す
access reviews	ユーザーがまだそのロールを必要としているかを定期的に確認する
audit history のダウンロード	内部監査・外部監査のために履歴を取り出す
最後の管理者の保護	最後に残った有効な Global Administrator と Privileged Role Administrator の割り当ての削除を防ぐ

用語:eligible と active を取り違えない

PIM の問題で最も落としやすいのがこの2語です。**この2つの区別が付かないと、シナリオ問題はまず解けません。**

用語	分類	意味
eligible	種類	ロールを使うために**ユーザー側の操作(アクティブ化)が必要**な割り当て。付与される権限そのものは永続的な割り当てと**同じ**で、違いは「常時は要らない」という一点だけです
active	種類	ロールを使うために**何の操作も要らない**割り当て。割り当てられた時点で権限を持ちます
activate	操作	eligible なロールを使うために行う操作。MFA の確認、業務上の理由の入力、指定された承認者への承認依頼などが含まれます
assigned	状態	active なロール割り当てを持っているユーザー
activated	状態	eligible な割り当てを持ち、アクティブ化を済ませて現在 active になっているユーザー。**あらかじめ構成された期間だけ**使え、その後は再アクティブ化が必要です
permanent eligible	期間	いつでもアクティブ化できる eligible 割り当て
permanent active	期間	何もせずいつでも使える active 割り当て

用語	分類	意味
time-bound eligible	期間	開始日と終了日の範囲内でだけアクティブ化できる
time-bound active	期間	開始日と終了日の範囲内でだけ使える

> **よく出る取り違え**:「eligible にすると権限が弱くなる」は誤りです。**アクティブ化した後の権限は permanent active とまったく同じ**です。違うのは「常時持っているか、必要なときだけ持つか」だけで、これが JIT の考え方そのものです。

PIM で管理できる3種類

PIM をセットアップすると、左メニューに **Tasks**、**Manage**、**Activity** が現れます。管理対象は次の3つから選びます。

管理対象	例	割り当てを管理できるのは誰か
Microsoft Entra roles	Global Administrator、User Administrator など	Privileged Role Administrator または Global Administrator だけ。Global Administrator、Security Administrator、Global Reader、Security Reader は**参照**もできます

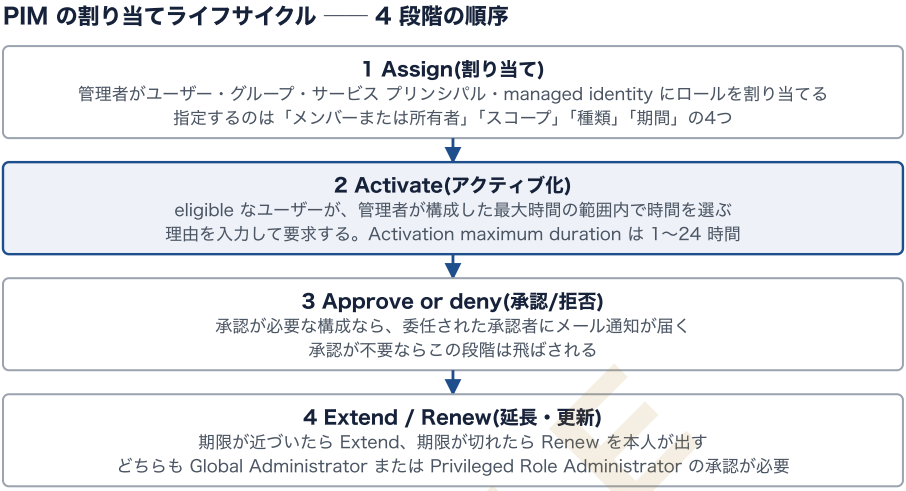
管理対象	例	割り当てを管理できるのは誰か
Azure resource roles	サブスクリプション/リソースグループ/リソースに対する Owner、Contributor など	サブスクリプション管理者、リソースの Owner、または User Access Administrator だけ。**Privileged Role Administrator、Security Administrator、Security Reader は既定では Azure リソース ロールの割り当てを参照できません**
PIM for Groups	ロール割り当て可能なグループのメンバー/所有者	グループの構成に従う

> **よく出る取り違え**:「Privileged Role Administrator なら Azure リソース ロールの PIM 割り当ても見られる」は誤りです。Microsoft Entra ロールと Azure リソース ロールは**別の権限体系**で、PIM の中でも管理者が分かれます。「Azure サブスクリプションのリソース グループに対する Contributor を JIT 化したい。誰が設定するか」と問われたら、答えは Privileged Role Administrator ではなく、そのスコープの **Owner または User Access Administrator** です。

割り当てのライフサイクル(並べ替え問題の根拠)

PIM の割り当ては次の流れで進みます。**この順序が並べ替え問題としてそのまま出ます。**

順	段階	何が起きるか
1	**Assign**(割り当て)	管理者がユーザー、グループ、サービス プリンシパル、または managed identity にロールを割り当てます。指定するのは「メンバーまたは所有者」「スコープ」「種類(eligible か active か)」「期間(開始日と終了日、または permanent)」の4つです
2	**Activate**(アクティブ化)	eligible なユーザーが、管理者が構成した**最大時間の範囲内**でアクティブ化の時間を選び、理由を入力して要求します
3	**Approve or deny**(承認/拒否)	承認が必要な構成なら、委任された承認者にメール通知が届き、承認後にメンバーがロールを使い始められます。承認が不要ならこの段階は飛ばされます
4	**Extend / Renew**(延長・更新)	期限が近づいたら **Extend**(期限前の延長要求)、期限が切れたら **Renew**(期限後の更新要求)を本人が出せます。**どちらも Global Administrator または Privileged Role Administrator の承認が必要**です



eligible = 使うためにアクティブ化が必要な割り当て。active = 何の操作も要らない割り当て。
アクティブ化した後の権限は permanent active とまったく同じ —— 違いは「常時持つか」だけ。

図 1-1　PIM の割り当てライフサイクル —— 4 段階の順序

Azure リソース ロールのロール設定(構成問題の中心)

PIM のロール設定は **PIM ポリシー**とも呼ばれ、**ロールごと・リソースごと**に定義されます。ここが極めて重要です。

- 同じロールに対するすべての割り当ては、同じロール設定に従います。
- あるロールの設定は、別のロールの設定から**独立**しています。
- あるリソースの設定は、別のリソースの設定から**独立**しています。
- **上位レベル(たとえば Subscription)で構成したロール設定は、下位レベル(たとえば Resource Group)に継承されません。**

> **よく出る取り違え**:「サブスクリプションで PIM 設定を構成したから、配下のリソース グループにも同じ承認要件が効く」は誤りです。継承しないので、リソース グループ側でも個別に構成する必要があります。ロール設定

> を管理するには **Owner または User Access Administrator** が必要です。

主なロール設定は次のとおりです。

設定	何を決めるか	押さえる値・注意
Activation maximum duration	アクティブ化の要求が有効な最大時間	**1〜24 時間**の範囲で指定します
On activation, require multifactor authentication	アクティブ化時に MFA を求める	すでにこのセッションで強い資格情報で認証済みだと、**再度は求められない場合があります**
On activation, require Microsoft Entra Conditional Access authentication context	アクティブ化時に、指定した認証コンテキストの Conditional Access ポリシー要件を満たさせる	先に Conditional Access ポリシーを作ってから PIM 側に設定するのが推奨です。ポリシーが1つも存在しない場合の保険として **MFA 要求と同じ動作**にフォールバックします
Require justification on activation	アクティブ化時に業務上の理由の入力を求める	—
Require ticket information on activation	アクティブ化時にサポートチケット番号の入力を求める	**情報を記録するだけ**で、チケット システムとの突き合わせは行われません
Require approval to activate	アクティブ化に承認を必須にする	承認者は**ロールを持っている必要はありません**。最低1名の指定が必須で、2名以上が推奨です。**既定の承認者は存在しません**

設定	何を決めるか	押さえる値・注意
Assignment duration(eligible)	`Allow permanent eligible assignment` か `Expire eligible assignment after`	後者にすると、すべての eligible 割り当てに開始日と終了日が必須になります
Assignment duration(active)	`Allow permanent active assignment` か `Expire active assignment after`	同上
Require multifactor authentication on active assignment	管理者が active 割り当てを作るときに MFA を求める	active はその瞬間から権限を持つため、**利用時には強制できません**。強制できるのは「作成時」だけです
Require justification on active assignment	active 割り当ての作成時に理由の入力を求める	—
Notifications	誰にどの通知を送るか	既定の宛先の解除、宛先の追加(セミコロン区切り)、重要な通知のみに絞る、が選べます。**1つのイベントで送られる通知は最大 1000 件**です

Conditional Access authentication context と併用したときの挙動も問われます。ある認証コンテキストで再認証を済ませると、**10 分間の猶予**が働き、その間に別の eligible ロールをアクティブ化しても再認証は求められません。この 10 分の窓は Microsoft Entra roles、Azure resource roles、PIM for Groups をまたいで適用されます。毎回必ず再認証させたい場合は、その認証コンテキストを対象とする Conditional Access ポリシーの Session controls で **sign-in frequency を Every time** に設定します。

> **誤答肢の切り方**:「PIM でロールをアクティブ化した後も、常に準拠デバイスからしか操作できないようにしたい」に対して、`On activation, require Conditional Access authentication context` だけを選ぶのは**不十分**です。この設定が縛るのは**アクティブ化の瞬間**だけで、アクティブ化後に非準拠デバイスへ移って権限を使うことは防げません。正解は、**eligible なユーザーを直接ターゲットにした Conditional Access ポリシー**で、常に準拠デバイスを要求する構成です。

PIM を使ったプログラム操作

PIM は Microsoft Graph API からも操作できます。Microsoft Entra roles 用の API と、PIM for groups 用の API が用意されています。自動化の要件が出たときは、ポータル手順ではなく Graph API が正解になります。

1-1-2 Conditional Access ポリシーの実装 [1.1]

Conditional Access ポリシーは、**Assignments**(誰に・何に・どこで)と**Access controls**(どうするか)から成る「もし〜ならば〜する」文です。複数のシグナルを組み合わせて判断し、組織のポリシーを強制します。

評価のされ方(ここが土台)

1つのユーザーに対して**複数の Conditional Access ポリシーが同時に適用され得ます。**その場合、**適用されるすべてのポリシーを満たす必要があります。**たとえば、あるポリシーが multifactor authentication を要求し、別のポリシーが準拠デバイスを要求するなら、MFA を完了し、かつ準拠デバイスを使う必要があります。

Assignments はすべて論理積(AND)で結合されます。複数の割り当てを構成した場合、ポリシーが発動するにはすべての割り当て条件が満たされる必

要があります。

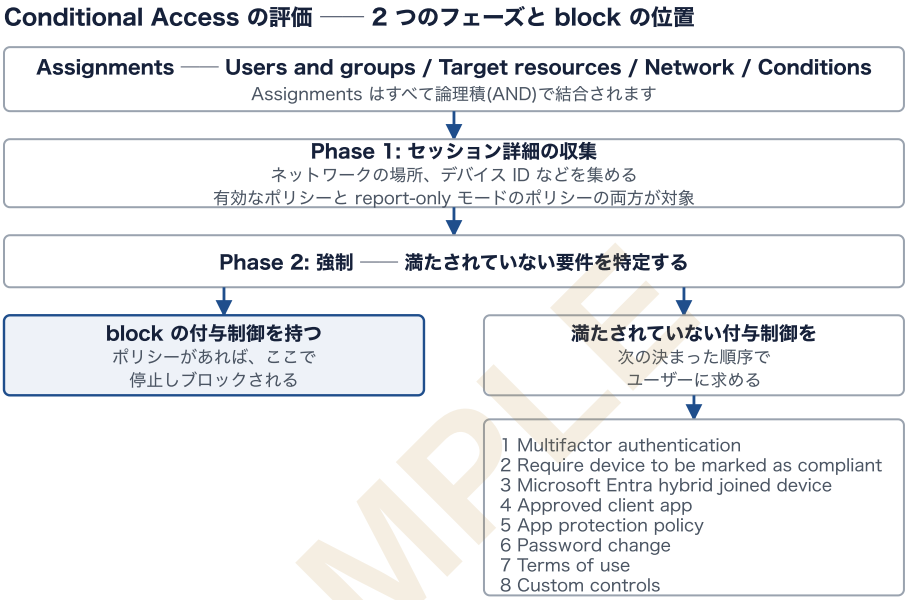
強制は**2つのフェーズ**で行われます。

フェーズ	何をするか	どのポリシーが対象か
Phase 1: セッション詳細の収集	ネットワークの場所、デバイス ID など、評価に必要な情報を集めます	**有効なポリシーと report-only モードのポリシーの両方**
Phase 2: 強制	フェーズ1の情報で、満たされていない要件を特定します。**block の付与制御を持つポリシーがあれば、ここで停止しユーザーはブロックされます**	有効なすべてのポリシー

フェーズ2で満たされていない付与制御は、**次の決まった順序**でユーザーに求められます。この順序も出題対象です。

1. Multifactor authentication
2. Require device to be marked as compliant(デバイスが準拠としてマークされていること)
3. Microsoft Entra hybrid joined device
4. Approved client app
5. App protection policy
6. Password change
7. Terms of use
8. Custom controls

すべての付与制御が満たされた後に、セッション制御(App Enforced、Microsoft Defender for Cloud Apps、トークンの有効期間)が適用されます。



すべての付与制御が満たされた後に、セッション制御(App Enforced、Microsoft Defender for Cloud Apps、トークンの有効期間)が適用されます。block は phase 2 の最初に効いて評価が止まるため、許可のポリシーが同時に当たっていても block が勝ちます。

図 1-2　Conditional Access の評価 —— 2 つのフェーズと block の位置

> **よく出る取り違え**:「ブロックのポリシーと許可のポリシーが両方当たったら、許可が優先される」は誤りです。**block は phase 2 の最初に効き、そこで評価が止まります。**Conditional Access は明示的な拒否が勝つ仕組みです。

Assignments の中身

Users and groupsでは、すべてのユーザー、特定のグループ、ディレクトリ ロール、外部ゲスト ユーザーを含めたり除外したりします。Microsoft Entra

Workload ID ライセンスがある組織は、**ワークロード ID(サービス プリンシパル)** も対象にできます。

ここに重要な制約があります。**ロールやグループを対象にしたポリシーは、トークンが発行されるときにだけ評価されます。**したがって、

- ロールやグループに新しく追加されたユーザーは、**新しいトークンを取得するまでポリシーの対象になりません。**
- 追加される前に有効なトークンを持っていたユーザーには、ポリシーが遡って適用されません。

この穴を塞ぐ推奨策が、**PIM のロール アクティブ化やグループ メンバーシップのアクティブ化のタイミングで Conditional Access の評価を発動させる**ことです。ここで 1-1-1 の「On activation, require multifactor authentication」とつながります。

Target resources には、クラウド アプリケーション、ユーザー アクション、認証コンテキストを含めたり除外したりします。

Network には IP アドレス、地理的な場所、そして Global Secure Access の準拠ネットワークが含まれます。管理者は場所を定義し、組織の主要な拠点などを**信頼済み(trusted)** としてマークできます。

Conditions は複数を組み合わせられます。

条件	内容	押さえる注意点
Sign-in risk	Microsoft Entra ID Protection が生成したリスク検出をポリシーに反映します	ID Protection が前提です

条件	内容	押さえる注意点
Device platforms	OS プラットフォームごとに別のポリシーを強制します	**判定情報はユーザー エージェント文字列など検証されない情報源に由来し、変更され得ます。**これを単独の防御線にしてはいけません
Client apps	ブラウザー、モバイル アプリとデスクトップ クライアントなど、利用しているソフトウェア	**新規に作成したポリシーは、client apps 条件を構成しなくても既定ですべてのクライアント アプリ種別に適用されます**
Filter for devices	デバイスの属性に基づいて特定のデバイスを対象にします	属性ベースの絞り込みに使います

Access controls の中身

Grant は「ブロックするか、条件付きで許可するか」を決めます。

- **Block access**……指定した割り当ての条件下でアクセスを遮断します。強力なので、扱いには相応の知識が要ります。
- **Grant access**……次の制御を1つ以上強制します。

付与制御	内容
Require multifactor authentication	MFA を要求します
Require authentication strength	認証強度(1-1-3 参照)を要求します
Require device to be marked as compliant	Intune で準拠と判定されたデバイスを要求します

付与制御	内容
Require Microsoft Entra hybrid joined device	ハイブリッド参加済みデバイスを要求します
Require approved client app	承認済みクライアント アプリを要求します
Require app protection policy	アプリ保護ポリシーの適用を要求します
Require password change	パスワード変更を要求します
Require terms of use	利用規約への同意を要求します

複数の制御を選んだ場合、**Require all the selected controls(すべて満たす)** と **Require one of the selected controls(いずれか1つ)** を選べます。**既定は「すべて」**です。「いずれか1つ」を選んだ場合、プロンプトは定義された順序で表示され、要件が満たされた時点でアクセスが許可されます。

Session はユーザー体験を制限します。

セッション制御	内容	制約
Use app enforced restrictions	デバイス情報をアプリに渡し、フル アクセスか制限付きアクセスかを制御します	**Exchange Online と SharePoint Online でのみ動作します**
Use Conditional Access App Control	Microsoft Defender for Cloud Apps のシグナルで、機密文書のダウンロード・切り取り・コピー・印刷のブロック、危険なセッション動作の監視、機密ファイルへのラベル付けの要求などを行います	Defender for Cloud Apps が前提です

セッション制御	内容	制約
Sign-in frequency	最新の認証に対する既定のサインイン頻度を変更します	—
Persistent browser session	ブラウザーを閉じて開き直してもサインイン状態を維持できるようにします	—
Customize continuous access evaluation	継続的アクセス評価の動作をカスタマイズします	—
Disable resilience defaults	障害時の回復性の既定動作を無効にします	—

> **よく出る取り違え**:「SharePoint Online で機密文書のダウンロードだけをブロックしたい」に対して、`Use app enforced restrictions` を選ぶのは**不十分**です。app enforced restrictions が渡すのは「フルか制限付きか」の粗い情報です。ダウンロード・コピー・印刷といった操作単位のブロックは **Conditional Access App Control(Defender for Cloud Apps)** の仕事です。

ポリシーとして成立する最小構成

Conditional Access ポリシーが強制されるには、少なくとも次が必要です。

- ポリシーの **Name**
- **Assignments** のうち、適用する **Users and/or groups** と **Target resources**
- **Access controls** の **Grant** または **Block**

実装手順(全ユーザーに MFA を要求する基本ポリシー)

Microsoft が推奨するベースラインは「**すべてのユーザー、すべてのリソース(アプリの除外なし)、multifactor authentication を要求**」です。手順は次のとおりで、**並べ替え問題の典型**です。

順	操作
1	Microsoft Entra 管理センターに、少なくとも **Conditional Access Administrator** としてサインインする
2	Entra ID > Conditional Access > Policies > New policy を選び、命名規則に沿った名前を付ける
3	Assignments > Users or workload identities > Include で **All users** を選ぶ
4	Exclude で**緊急アクセス(break-glass)アカウント**を除外する。Microsoft Entra Connect や Cloud Sync を使うなら Directory roles から **Directory Synchronization Accounts** も除外する
5	Target resources > Resources > Include で **All resources** を選ぶ
6	Access controls > Grant > Grant access で **Require authentication strength** を選び、組み込みの **Multifactor authentication strength** を選ぶ
7	Enable policy を **Report-only** にして作成する
8	report-only の結果を確認してから、トグルを **On** に切り替える

除外すべきアカウントは2種類です。1つは**緊急アクセス(break-glass)アカウント**で、ポリシーの構成ミスによる締め出しを防ぎます。もう1つは**サービス アカウントとサービス プリンシパル**(Microsoft Entra Connect の同期アカウントなど)です。サービス プリンシパルが行う呼び出しは、ユーザーを対象にした Conditional Access ポリシーではブロックされないため、**Conditional Access for workload identities** で別途ポリシーを定義します。スクリプトやコードでサービス アカウントを使っている場合は、**managed identities への置き換え**が推奨されます(1-1-6 につながります)。

社内ネットワークからのアクセスだけ MFA を免除したい場合は、Assignments の Network を Yes にし、Include で **Any network or location**、Exclude で **All trusted networks and locations** を指定します。

> **誤答肢の切り方**:「外部認証方法(external authentication methods)を使っている環境で、全ユーザーに MFA を要求したい」に対して **Require authentication strength** を選ぶのは誤りです。外部認証方法は現時点で authentication strength と互換性がありません。この場合は **Require multifactor authentication** の付与制御を使います。

1-1-3 認証方法の実装と構成 — multifactor authentication (MFA) と passwordless [1.1]

認証は、アプリ、サービス、デバイス、ネットワークへのアクセスを許可する前に本人性を検証する処理です。Microsoft Entra ID の各 **authentication methods** は、「主要認証(第1要素)に使えるか」「**multifactor authentication (MFA)** の第2要素に使えるか」「セルフサービス パスワード

リセット(SSPR)またはアカウント回復に使えるか」で性格が分かれます。**この表の縦横を取り違えるのが典型的な失点です。**

方法	主要認証	第2要素	SSPR / アカウント回復
Authenticator Lite	不可	MFA	不可
Certificate-based authentication	可	MFA	不可
Email OTP	不可	SSPR とサインイン (ゲスト向け)	SSPR
External MFA	不可	MFA	不可
Hardware OATH tokens(プレビュー)	不可	MFA	SSPR
Microsoft Authenticator passwordless	可	不可	不可
Microsoft Authenticator push notifications	可	MFA	SSPR
Passkey (FIDO2)	可	MFA	不可
Passkey in Microsoft Authenticator	可	MFA	不可
Password	可	不可	不可
Platform Credential for macOS	可	MFA	不可

方法	主要認証	第2要素	SSPR / アカウント回復
QR code	可	不可	不可
SMS sign-in	可	MFA	SSPR
Software OATH tokens	不可	MFA	SSPR
Synced passkey	可	MFA	不可
Temporary Access Pass (TAP)	可	MFA	不可
Verified ID	不可	不可	アカウント回復のみ
Voice call	不可	MFA	SSPR
Windows Hello for Business	可	MFA(条件付き)	不可

Windows Hello for Business が第2要素(ステップアップ)として使えるのは、**ユーザーが passkey (FIDO2) に対して有効化されていて、passkey を登録している場合**です。

Verified ID は従来型の認証方法ではありません。これは ID 検証の機能で、サインイン、MFA、SSPR の要件を満たすためには使えません。使えるのは、すべての認証方法を失ったときの**アカウント回復**で、暗号学的に検証された ID の証明を提供する場面だけです。

passwordless とフィッシング耐性

SMS、Email OTP、認証アプリによる従来型の MFA は、パスワードのみに比べて大幅にセキュリティを向上させますが、**リモート フィッシング攻撃に弱い**という弱点があります。攻撃者はソーシャル エンジニアリングと AI を使った道

具で、デバイスへの物理アクセスなしにパスワードやワンタイム コードを盗みます。

Microsoft Entra ID で利用できる**フィッシング耐性のある認証方法**は次のとおりです。**この一覧が passwordless 系の設問の答えの母集団になります。**

- Windows Hello for Business
- Platform Credential for macOS
- Synced passkeys (FIDO2)
- FIDO2 security keys
- Passkeys in Microsoft Authenticator
- Certificate-based authentication (CBA)

Microsoft は、最も安全なサインイン体験として、Windows Hello for Business、passkeys (FIDO2) と FIDO2 security keys、または certificate-based authentication を推奨しています。

高保証のアカウント回復

すべての資格情報を失ったユーザーの回復は、従来はヘルプデスクへの電話と口頭での本人確認でした。Microsoft Entra ID は現在、**政府発行 ID の検証と生体照合による高保証のアカウント回復**をサポートし、ヘルプデスクの介在を不要にしてソーシャル エンジニアリングのリスクを排除します。

Microsoft Entra Verified ID の **Face Check** は Azure AI サービスを基盤とし、ユーザーのリアルタイムの自撮りと ID 文書の写真を照合して本人の存在を検証します。**共有されるのは照合結果だけ**で、機微な ID データそのものは共有されません。ID 検証プロファイルは、どのユーザーが Verified ID フロー

に参加できるか、どのプロバイダーが検証を行うか、ID クレームをどう検証するかを制御します。

authentication strengths(認証強度)

authentication strength は Conditional Access の制御で、あるリソースにアクセスするために使える認証方法の**組み合わせ**を指定します。ユーザーは、許可された組み合わせのいずれかで認証すれば要件を満たせます。利用には **Microsoft Entra ID P1** ライセンスが必要です。

組み込みの認証強度は**3つ**です。組み込みは常に利用でき、**変更はできません**。新しい方法が使えるようになると Microsoft が更新します。

認証方法の組み合わせ	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
FIDO2 security key	該当	該当	該当
Windows Hello for Business またはプラットフォーム資格情報	該当	該当	該当
Certificate-based authentication (multifactor)	該当	該当	該当
Microsoft Authenticator (phone sign-in)	該当	該当	—
Temporary Access Pass(1回限り/複数回)	該当	—	—

認証方法の組み合わせ	MFA strength	Passwordless MFA strength	Phishing-resistant MFA strength
パスワード + ユーザーが持っているもの	該当	—	—
フェデレーション単一要素 + ユーザーが持っているもの	該当	—	—
フェデレーション多要素	該当	—	—
Certificate-based authentication (single-factor)	—	—	—
SMS sign-in	—	—	—
Password	—	—	—
QR code	—	—	—

「ユーザーが持っているもの」とは、テキスト メッセージ、音声通話、プッシュ通知、software OATH token、hardware OATH token のいずれかを指します。

内側の強度を満たす方法は、外側の強度も満たします。組み込みの 3 つは常に利用でき、変更はできません。
Certificate-based authentication (single-factor)、SMS sign-in、Password、QR code はどの強度にも含まれません。利用には Microsoft Entra ID P1 ライセンスが必要です。

図 1-3　組み込みの authentication strength は 3 つ —— 入れ子の関係

カスタム認証強度は、アクセス要件に正確に合わせて管理者が作成できます。

authentication strength の**制限**は、そのまま誤答肢の材料になります。

- **初回認証は制限できません。**Conditional Access ポリシーは初回認証の後に評価されるため、Phishing-resistant MFA strength を使っていてもユーザーはパスワードを入力できます。ただし、続行する前に FIDO2 security key などフィッシング耐性のある方法でサインインする必要があります。
- **Require multifactor authentication と Require authentication strength は、同じ Conditional Access ポリシーの中で併用できません。**組み込みの Multifactor authentication strength が Require multifactor authentication と等価だからです。

- **Email one-time pass (Guest)** は現時点で利用できる組み合わせに含まれていません。
- Windows Hello for Business を主要認証として使った場合は、それを含む認証強度の要件を満たせます。しかしパスワードなど別の方法で主要認証を行い、認証強度が Windows Hello for Business を要求している場合、**ユーザーは自動的には Windows Hello for Business を求められません**。セッションを開始し直し、Sign-in options から要求された方法を選ぶ必要があります。

authentication strength と Authentication methods ポリシーの使い分けも問われます。Authentication methods ポリシーは「テナント全体でどの方法をどのユーザーに使わせるか」を決める道具です。authentication strength は「機密リソースへのアクセス、ユーザー リスク、場所といった特定のシナリオで、その中のどれに絞るか」を決める道具です。**前者が母集団、後者が場面ごとの絞り込み**、と覚えます。

> **誤答肢の切り方**:「機密アプリだけ passwordless を必須にし、他のアプリは Authenticator のプッシュ通知でよい」に対して、Authentication methods ポリシーで Authenticator の Authentication mode を Passwordless に限定するのは**やり過ぎ**です。それだとテナント全体でプッシュ通知が使えなくなります。正解は、Authentication methods ポリシーで Authentication mode を **Any** にしたうえで、機密アプリを対象にした Conditional Access ポリシーに **Passwordless MFA strength** を適用する構成です。

1-1-4 アプリケーションの ID の実装と構成 — enterprise applications と app registrations [1.1]

アプリケーションが Microsoft Entra ID に ID とアクセス管理を委任するには、**Microsoft Entra テナントに登録(app registrations)** する必要があります。ここで作られるオブジェクトが2種類あり、**その区別が 1.1 で最も混乱しやすい箇所**です。

application object と service principal

アプリケーションの登録を行うと、ホーム テナントに application object と service principal object が自動的に作成されます。ただし Microsoft Graph API でアプリケーションを登録・作成した場合は、**service principal object の作成は別の手順**になります。

観点	application object	service principal object
どこにあるか	アプリケーションを登録した**ホーム テナント**に1つだけ	アプリケーションが**使われるすべてのテナント**にそれぞれ1つ
役割	service principal を作るための**テンプレート(設計図)**	テナント内での**実体(インスタンス)**
管理画面	**App registrations** ページ	**Enterprise applications** ページ
関係	ソフトウェアと 1 対 1	application object と 1 対多
何を記述するか	トークンの発行方法、アプリがアクセスする必要があるリソース、アプリが取れるアクション	そのテナントでアプリが実際に何をできるか、誰がアプリにアクセスできるか、どのリソースにアクセスできるか
Graph のエンティティ	Application	ServicePrincipal

登録が完了すると、**グローバルに一意なアプリのインスタンス(application object)** と、**グローバルに一意な ID(app ID / client ID)** が手に入ります。ここにシークレットや証明書、スコープを追加し、サインイン画面のブランドをカスタマイズできます。登録時には**単一テナント(single tenant)** か**マルチテナント(multitenant)** かを選び、必要に応じて**リダイレクト URI** を設定します。

service principal の3つの種類

種類	内容	特徴
Application	グローバルな application object の、あるテナントでのローカルな表現	登録時または同意時に作成されます。Azure PowerShell、Azure CLI、Microsoft Graph などでも作成できます
Managed identity	managed identity を表す種類	**関連する app object を持ちません。**アクセスや権限を付与できますが、**直接更新・変更することはできません**
Legacy	app registrations が導入される前、または旧来の方法で作られたアプリ	資格情報、サービス プリンシパル名、応答 URL などを持てますが、**関連する app registration を持たず、作成されたテナントでのみ使えます**

> **よく出る取り違え**:「Enterprise applications 画面でアプリの資格情報を編集する」は、Application 種別では正しくありません。編集の起点は **App registrations(application object)** 側です。逆に、「そのテナントでどのユーザーがアプリを使えるか」「どの権限に同意されたか」「誰が同

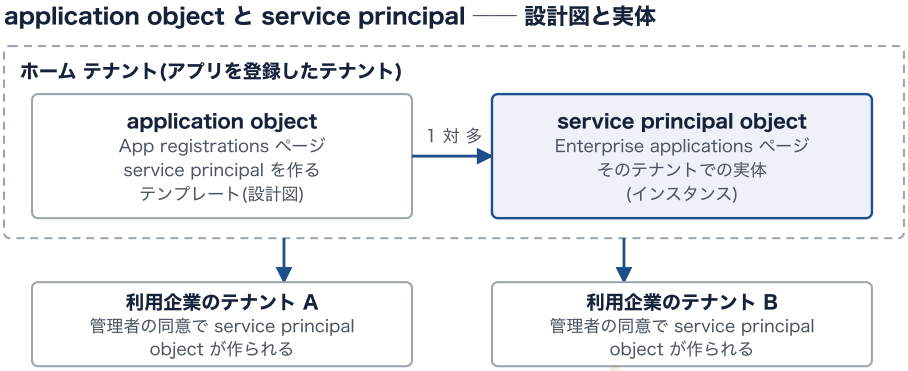
> 意したか」「サインインの情報」を見るのは **Enterprise applications(service principal)** 側です。

マルチテナントでの動き(シナリオ問題の型)

HR アプリを開発した Adatum 社と、それを使う Contoso 社・Fabrikam 社という3テナントの例で整理します。

段階	何が起きるか
1	アプリのホーム テナント(Adatum)で application object と service principal object が作られる
2	Contoso と Fabrikam の管理者が同意を完了すると、**各社のテナントに service principal object が作られ**、管理者が付与した権限が割り当てられる
3	各利用企業は自社の service principal を持ち、それぞれの管理者が同意した権限の範囲で実行時のアプリ利用を表す

単一テナント アプリは**ホーム テナントに service principal が1つだけ**です。マルチテナント アプリは、そのテナントのユーザーが利用に同意したすべてのテナントにも service principal が作られます。



application object はホーム テナントに 1 つだけ、service principal はアプリが使われる
すべてのテナントにそれぞれ 1 つ。application object を削除するとホーム テナントの
service principal も削除され、UI から復元しても service principal は復元されません。
調査のために構成を残したまま止めるなら、削除ではなく非アクティブ化(deactivate)です。

図 1-4　application object と service principal —— 設計図と実体

変更と削除の影響(ここも頻出)

- application object に加えた変更は、**ホーム テナントの service principal object にのみ**反映されます。
- したがって **application object を削除すると、ホーム テナントの service principal object も削除されます。**
- ただし、**App registrations の UI から application object を復元しても、対応する service principal は復元されません。**
- 恒久的な削除ではなく一時的な停止が必要な場合は、アプリケーションを**非アクティブ化(deactivate)** できます。非アクティブ化は、調査や将来の再有効化のために application object と service principal を保持したまま、**新しいトークンの発行を止めます。**

> **誤答肢の切り方**:「侵害の疑いがあるアプリを止めたい。ただし調査のため構成は残したい」に対して、application object の削除は誤りです。削

> 除すると service principal も消え、復元しても戻りません。正解は**非アクティブ化**です。

特定のアプリの service principal を探す

- ポータル……App registrations の概要で **Managed application in local directory** を選びます。
- Microsoft Graph PowerShell…… `Get-MgServicePrincipal -Filter "appId eq '{AppId}'"`
- Azure CLI…… `az ad sp list --filter "appId eq '{AppId}'"`

ワークロード ID の作り方の選択肢

アプリケーションの ID を用意する方法は1つではありません。**シナリオで「シークレットを保存したくない」と言われたら、選択肢が切り替わります。**

方法	何に使うか	資格情報
App registration	従来型のアプリケーションやプラットフォーム統合	クライアント シークレット、証明書、またはフェデレーション資格情報
Managed identities	Azure 上で動くワークロード	**Azure が管理し、コードや構成にシークレットを置きません**
Workload identity federation	Azure の外のワークロード、または app registration として認証する Azure 内のワークロード	外部 ID プロバイダーとの信頼関係。シークレットを保存しません

方法	何に使うか	資格情報
Flexible federated identity credentials(プレビュー)	GitHub、GitLab、Terraform Cloud が発行するトークンに対して、ワイルドカードやクレーム ベースの照合が必要な場面	同上

これらのワークロード ID にも、ユーザーと同じ多層の統制を当てます。**Conditional Access for workload identities** はサービス プリンシパルがいつ・どこから認証できるかを制限しますが、**Workload Identities Premium ライセンスが必要**で、**自テナントに登録された単一テナントのサービス プリンシパルにのみ適用されます**。managed identities、マルチテナント アプリ、サード パーティ SaaS アプリは対象外です。

app instance property lock は、マルチテナント アプリが別のテナントにプロビジョニングされた後、service principal の機微なプロパティが不正に変更されるのを防ぎます。

managed identity を **Entra ID アプリのフェデレーション ID 資格情報 (FIC)** として使うこともできます。Entra ID アプリが必要な場面では、これが資格情報を持たない推奨の方法です。**managed identity を FIC として使う場合の上限は 20 件**です。

1-1-5 OAuth permission grants と consent settings の管理 [1.1]

アプリケーションが組織のデータにアクセスする前に、ユーザーが権限を付与(同意)する必要があります。**既定では、すべてのユーザーが管理者の同意を必要としない権限についてアプリに同意できます。**たとえば既定のままなら、ユ

ーザーは自分のメールボックスへのアクセスをアプリに許可できますが、組織のすべてのファイルの読み書きを無制限に許可することはできません。

悪意あるアプリがユーザーを騙して組織のデータへのアクセスを得るリスクを下げるため、Microsoft は「**verified publisher(検証済み発行元)が公開したアプリにのみユーザー同意を許可する**」ことを推奨しています。

3つの consent settings

User consent settings は Microsoft Entra 管理センターの Identity > Applications > Enterprise apps > **Consent and permissions** > **User consent settings** で構成します。設定は組織のすべてのユーザーに適用されます。

設定	組み込みポリシー ID	内容
Do not allow user consent	(割り当てなし)	ユーザーの同意を禁止します。管理者だけが同意できます
Allow user consent for apps from verified publishers, for selected permissions	`microsoft-user-default-low`	verified publisher のアプリと自テナントに登録されたアプリに限り、しかも**low impact に分類した権限**についてのみユーザー同意を許可します。**権限の分類(permission classifications)を先に行う必要があります**
Allow user consent for apps	`microsoft-user-default-legacy`	管理者の同意を必要としないすべての権限について、すべてのアプリにユーザーが同意できます

構成に必要なロールは **Privileged Role Administrator** です。Microsoft Entra 管理センターを使う場合に限り **Global Administrator** が必要になります。

> **重要な例外**: アプリへのユーザー割り当てを必須にしているアプリケーションは、たとえディレクトリの同意ポリシーが本人による同意を許していても、**権限は管理者が同意する必要があります**。

プログラムからの構成(テナント全体の authorizationPolicy)

テナント全体の設定は `authorizationPolicy` が保持し、個々の条件は `permissionGrantPolicies` が保持します。この2つの区別が問われます。

オブジェクト	何を決めるか
authorizationPolicy	ユーザーがアプリに同意できるかどうか、既定のユーザー ロールにどの permission grant policy が割り当てられているか、といったグローバル設定
permissionGrantPolicies	どの権限を、どんな状況下でアプリに付与できるか。各ポリシーは特定の条件を「含み(includes)」、他を「除外(excludes)」します

Microsoft Graph PowerShell では `Update-MgPolicyAuthorizationPolicy` で authorizationPolicy を更新し、`Get-MgPolicyPermissionGrantPolicy` で現在の permission grant policies を一覧します。Microsoft Graph API では `GET https://graph.microsoft.com/v1.0/policies/authorizationPolicy/authorizationPolicy` と `GET https://graph.microsoft.com/v1.0/policies/permissionGrantPolicies` に相当します。

更新前には必ず現在の authorizationPolicy を取得し、すでに割り当てられている permission grant policy を確認します。これを怠ると、開発者が自分の所有するアプリの同意を管理できるようにする `ManagePermissionGrantsForOwnedResource.DeveloperConsent` のような設定を、意図せず消してしまいます。

ユーザー同意を無効にしつつ開発者の同意管理だけ残す構成は、`permissionGrantPoliciesAssigned` に `ManagePermissionGrantsForOwnedResource.DeveloperConsent` だけを割り当てる形で実現します。

読み取りだけなら `Policy.Read.All`、読み書きするなら `Policy.ReadWrite.Authorization` の権限で `Connect-MgGraph` します。

> **よく出る取り違え**:「同意設定を厳しくしたら、すでに与えられた権限も取り消される」は誤りです。**同意設定の更新は将来の同意操作にのみ影響し、既存の同意付与は変更されません。**ユーザーは以前に付与された権限に基づいてアクセスを継続します。既存の付与を取り消すには、次項の手順が別途必要です。

admin consent workflow

ユーザーが同意できないアプリについて、**管理者のレビューと承認を要求できるようにする**のが **admin consent workflow** です。ユーザー同意を無効にしている場合や、アプリがユーザーには付与できない権限を要求している場合に有効化します。

付与済み権限のレビューと取り消し

悪意あるアプリを検出したとき、または必要以上の権限を持つアプリを見つけたときは、付与済みの権限をレビューして取り消します。この作業に必要なロールは **Cloud Application Administrator** または **Application Administrator** です。管理者でないサービス プリンシパルの所有者でも、**更新トークンの無効化はできます**。

権限には2種類あり、**取り消しの方法が異なります**。

種類	実体	ポータルでの場所	ポータルで取り消せるか
委任された権限 (delegated permissions)	`oauth2PermissionGrants`	Enterprise apps > 該当アプリ > **Permissions** > **Admin consent** タブ / **User consent** タブ	Admin consent タブは可能。**User consent タブはポータルでは取り消せず、Microsoft Graph API か PowerShell が必要**です
アプリケーション権限(application permissions)	`appRoleAssignments`	同上	Graph / PowerShell で `appRoleAssignments` を削除します

Microsoft Graph での操作は次のとおりです。

- サービス プリンシパルの取得…… `GET https://graph.microsoft.com/v1.0/servicePrincipals/{id}`
- 委任された権限の一覧…… `GET https://graph.microsoft.com/v1.0/servicePrincipals/{id}/oauth2PermissionGrants`

- 委任された権限の削除…… `DELETE https://graph.microsoft.com/v1.0/oAuth2PermissionGrants/{id}`
- アプリケーション権限の一覧…… `GET https://graph.microsoft.com/v1.0/servicePrincipals/{servicePrincipal-id}/appRoleAssignments`
- アプリケーション権限の削除…… `DELETE https://graph.microsoft.com/v1.0/servicePrincipals/{resource-servicePrincipal-id}/appRoleAssignedTo/{appRoleAssignment-id}`
- ユーザーとグループの割り当ての一覧…… `GET https://graph.microsoft.com/v1.0/servicePrincipals/{servicePrincipal-id}/appRoleAssignedTo`

Graph Explorer から実行する場合に必要な同意は `Application.ReadWrite.All`、`Directory.ReadWrite.All`、`DelegatedPermissionGrant.ReadWrite.All`、`AppRoleAssignmentReadWrite.All` に相当する一式です。PowerShell では `Connect-MgGraph -Scopes "Application.ReadWrite.All", "Directory.ReadWrite.All", "DelegatedPermissionGrant.ReadWrite.All", "AppRoleAssignment.ReadWrite.All"` としてから、`Get-MgOauth2PermissionGrant` と `Remove-MgOauth2PermissionGrant`、`Get-MgServicePrincipalAppRoleAssignment` と `Remove-MgServicePrincipalAppRoleAssignedTo` を使います。

> **誤答肢の切り方**:「侵害されたアプリの権限をすべて取り消せば、それで完了」は誤りです。**権限を取り消しても、ユーザーがアプリの要求する権限に再度同意することは止められません。**再同意を止めるには、アプリ側の動的同意による権限要求を止めるか、`configure-user-consent` の手順でユーザー同意そのものをブロックする必要があります。あわせて**ユーザー**

> **とグループの割り当てを削除して更新トークンを無効化する**手順も踏みます。

委任された権限とアプリケーション権限だけが認可の経路ではない点も注意します。Microsoft Entra の組み込みロール、Exchange RBAC、Teams のリソース固有の同意(RSC)といった別の認可システムからも、機微な情報へのアクセスが与えられ得ます。

1-1-6 Azure リソースの managed identities の実装と構成 [1.1]

シークレット、資格情報、証明書、キーの管理は、既知の障害とセキュリティ問題の発生源です。**managed identities** は、開発者がこれらを管理する必要をなくします。アプリケーションは資格情報を一切管理せずに Microsoft Entra のトークンを取得できます。

managed identity は Azure のコンピューティング リソース(Azure Virtual Machine、Virtual Machine Scale Set、Service Fabric クラスター、Azure Kubernetes クラスター)や、Azure がサポートする任意のアプリ ホスティング プラットフォームに割り当てられます。割り当てられた後は、ストレージ アカウント、SQL データベース、Cosmos DB といった下流の依存リソースへのアクセスを直接または間接に認可できます。

managed identities の利点は3つです。

- 資格情報を管理する必要がありません。**資格情報は管理者からもアクセスできません。**
- Microsoft Entra 認証をサポートするあらゆるリソース(自作のアプリケーションを含む)に対して認証できます。
- **追加費用なしで利用できます。**

system-assigned と user-assigned の分かれ目

この表が対応付け問題の答えそのものです。

プロパティ	system-assigned managed identity	user-assigned managed identity
作成	Azure リソースの一部として作成されます(Azure Virtual Machines、Azure App Service など)	**スタンドアロンの Azure リソース**として作成されます
ライフ サイクル	作成元の Azure リソースと**ライフ サイクルを共有**します。親リソースを削除すると managed identity も削除されます	**独立したライフ サイクル**。明示的に削除する必要があります
Azure リソース間での共有	**共有できません。**単一の Azure リソースにのみ関連付けられます	**共有できます。**同じ user-assigned managed identity を複数の Azure リソースに関連付けられます
典型的な用途	単一の Azure リソースの中で完結するワークロード。独立した ID が必要なワークロード。たとえば単一の仮想マシンで動くアプリケーション	複数のリソースで動き1つの ID を共有できるワークロード。プロビジョニングの流れの中で、あらかじめ保護されたリソースへの認可が要るワークロード。**リソースが頻繁に作り直されるが権限は一定に保ちたい**ワークロード。たとえば複数の仮想マシンから同じリソースにアクセスする場合

user-assigned managed identity は、コンピューティングとは独立してプロビジョニングされ複数のコンピューティング リソースに割り当てられるた

め、Microsoft サービスにとって推奨される種別です。

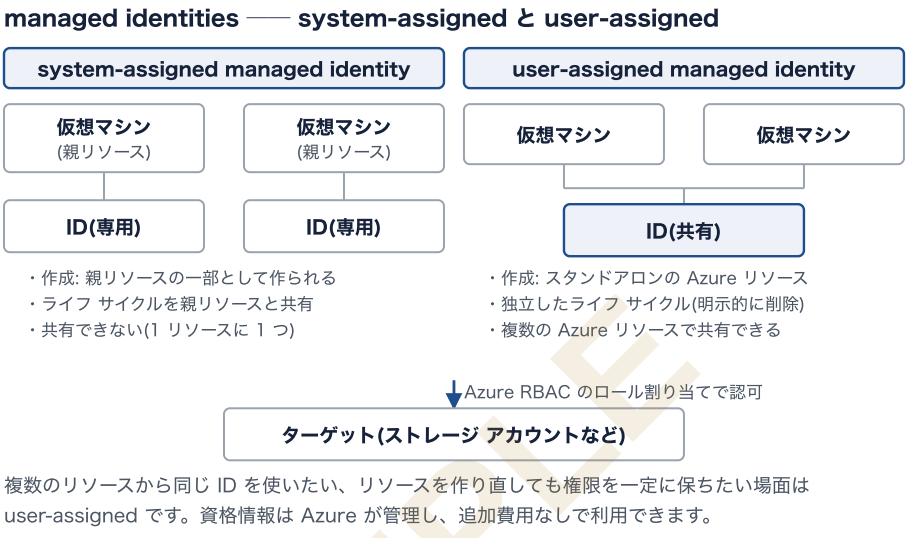


図 1-5　managed identities —— system-assigned と user-assigned

system-assigned を有効にすると、**特別な種類のサービス プリンシパル**が Microsoft Entra ID に作られます。このサービス プリンシパルは**その Azure リソースのライフ サイクルに紐づき**、リソースを削除すると Azure が自動的にサービス プリンシパルを削除します。**設計上、そのリソースだけがこの ID でトークンを要求できます。**system-assigned のサービス プリンシパル名は、常に作成元の Azure リソース名と同じです。デプロイ スロットの場合は `<app-name>/slots/<slot-name>` になります。

user-assigned を有効にすると、同じく特別な種類のサービス プリンシパルが作られますが、こちらは**使用するリソースとは別に管理**されます。

どちらの種別でも、**Azure RBAC のロール割り当てで権限を付与**し、**CRUD 操作は Azure アクティビティ ログ**で、**サインイン アクティビティは**

Microsoft Entra ID のサインイン ログで確認できます。操作は ARM テンプレート、Azure ポータル、Azure CLI、PowerShell、REST API から行えます。

利用の3ステップ(並べ替え問題の根拠)

順	操作
1	Azure で managed identity を作成する。system-assigned か user-assigned かを選ぶ。**user-assigned の場合は、仮想マシン、Azure Logic App、Azure Web App などの「ソース」リソースに割り当てる**
2	managed identity に「ターゲット」サービスへのアクセスを**認可**する(Azure RBAC のロール割り当てなど)
3	managed identity を使ってリソースにアクセスする。Azure SDK の Azure.Identity ライブラリまたは MSAL を使う。ソース リソースがコネクタを提供している場合は、その機能として ID を使う

Azure のコンピューティング リソース上で動くサービス コードは、**MSAL または Azure.Identity SDK** を使って managed identity のトークンを取得します。**このトークン取得にシークレットは一切不要**で、コードが動いている環境に基づいて自動的に認証されます。

> **誤答肢の切り方**:「複数の仮想マシンから同じストレージ アカウントにアクセスさせたい。管理負荷を最小にしたい」に対して、system-assigned managed identity を各 VM で有効化する構成は**不適切**です。system-assigned は共有できないため、VM の台数だけロール割り当てが増え、VM を作り直すたびに権限を付け直すことになります。正解は **user-**

> **assigned managed identity を1つ作り、全 VM に割り当て、ストレージ側で1回だけロール割り当てを行う**構成です。

> **もう1つの取り違え**: managed identity を表すサービス プリンシパルは、**アクセスや権限は付与できますが、直接更新・変更することはできません。**「managed identity のサービス プリンシパルにアプリ登録の資格情報を追加する」といった選択肢は誤りです。

1-1-7 到達目標 1.1 の分かれ目の早見表 [1.1]

シナリオで問われる典型的な要件と、正解になる構成の対応です。**対応付け問題ではこの表がそのまま答えになります。**

要件の言葉	正解になる構成	よく置かれる誤答肢
管理者権限を常時持たせたくない	**PIM** の eligible 割り当て + JIT アクティブ化	Conditional Access で MFA を要求する(権限の常時保有は解消しません)
昇格時に承認を挟みたい	PIM のロール設定で **Require approval to activate**、承認者を1名以上(2名以上推奨)指定	Access reviews(事後の確認であって昇格の関所ではありません)
昇格時に必ず再認証させたい	PIM で **Conditional Access authentication context** を要求し、そのポリシーの sign-in frequency を **Every time** にする	On activation, require multifactor authentication だけ(セッション内では再要求されない場合があります)
権限が今も必要か定期確認したい	**access reviews**	audit history のダウンロード(記録であって確認プロセスではありません)

要件の言葉	正解になる構成	よく置かれる誤答肢
特定のアプリだけフィッシング耐性を必須にしたい	Conditional Access で **Require authentication strength** に **Phishing-resistant MFA strength**	Authentication methods ポリシーで方法を絞る(テナント全体に効いてしまいます)
ユーザーが怪しいアプリに同意するのを止めたい	**User consent settings** を Do not allow user consent、または verified publishers + low impact に限定し、**admin consent workflow** を有効化	付与済みの権限を取り消すだけ(再同意を止められません)
すでに与えた権限を剥がしたい	Enterprise apps > Permissions で **Admin consent** を取り消し、User consent は Graph / PowerShell で `oauth2PermissionGrants` を削除、あわせて `appRoleAssignments` を削除して更新トークンを無効化	アプリを削除する(復元しても service principal は戻りません)
侵害の疑いがあるアプリを止めたいが構成は残したい	アプリケーションの**非アクティブ化(deactivate)**	application object の削除
コードや構成にシークレットを置きたくない	**managed identities**(Azure 内)または **workload identity federation**(Azure 外)	app registration にクライアント シークレットを作り Key Vault に置く(シークレット自体は残ります)
複数リソースで同じ権限を共有したい	**user-assigned managed identity**	system-assigned managed identity(共有できません)

要件の言葉	正解になる構成	よく置かれる誤答肢
リソース削除時に ID も自動で消えてほしい	**system-assigned managed identity**	user-assigned managed identity(明示的な削除が必要です)
サービス プリンシパルの認証を場所や条件で縛りたい	**Conditional Access for workload identities**(Workload Identities Premium が必要、単一テナントのサービス プリンシパルのみ)	通常の Conditional Access ポリシー(ユーザー対象では止まりません)

1-2 Azure Key Vault を使用してシークレットとキーをセキュリティで保護する [1.2]

この到達目標は「**秘密をどこに置き、誰に、どの経路から、どこまで触らせるか**」の範囲です。Key Vault の配置、設定、アクセス制御、ファイアウォール、キーとシークレットと証明書の管理に加えて、**Defender Cloud Security Posture Management (Defender CSPM)** による漏れたシークレットの発見と、**Defender for Key Vault** による異常検知までが含まれます。

手順の順序を先に置きます。これがそのまま並べ替え問題になります。

順	工程	何を決めるか
1	**Deploy Key Vault**	リージョン、SKU(Standard / Premium)、soft-delete の保持期間、purge protection。**保持期間は作成時にしか決められません**

順	工程	何を決めるか
2	**Configure Key Vault settings**	権限モデル(Azure RBAC か access policy か)、purge protection、診断ログ
3	**Configure access to Key Vault**	制御プレーンと データ プレーンのロール割り当て、または access policy
4	**Configure firewall settings on Key Vault**	既定の拒否、IP 規則、仮想ネットワーク規則、信頼されたサービスのバイパス、プライベート エンドポイント
5	**Manage keys, secrets, and certificates**	キーの種類、バージョン、ローテーション、証明書の発行
6	**Defender for Key Vault の有効化**	異常なアクセスの検知

1-2-1 Key Vault の配置(Deploy Key Vault) [1.2]

Azure Key Vault は Azure における複数の鍵管理ソリューションのひとつで、次の3つの問題を解きます。

用途	何を守るか
Secrets Management	トークン、パスワード、証明書、API キーなどのシークレットを安全に保管し、アクセスを厳密に制御します
Key Management	データの暗号化に使う暗号化キーを作成・制御します

用途	何を守るか
Certificate Management	Azure および社内の接続されたリソースで使う公開/プライベートの TLS/SSL 証明書をプロビジョニング、管理、デプロイします

Standard と Premium の分かれ目

SKU	暗号モジュール	使えるキーの保護
Standard tier	**FIPS 140 Level 1** 検証済みのソフトウェア ライブラリでデータを暗号化します	ソフトウェア保護キー
Premium tier	**FIPS 140-3 Level 3** 検証済みの Marvell LiquidSecurity HSM で生成・保護される **HSM 保護キー**	RSA-HSM、EC-HSM、oct-HSM(対称鍵、プレビュー)

すべての Key Vault は、**HSM に保管されたキーで保存時に暗号化**されています。Premium の HSM 保護キー(RSA-HSM、EC-HSM、OCT-HSM)は **HSM の境界から出ません。Azure Key Vault は、Microsoft がデータを見たり取り出したりできない設計**になっています。

vaults と Managed HSMs

Key Vault のリソース プロバイダーは **vaults** と **managed HSMs** の2種類のリソース タイプをサポートします。**アクセス制御の仕組みが異なる**ため、混同は失点に直結します。

リソース種別	キーの保護方法	サポートするキーの種類	データ プレーンのエンドポイント
Vaults	ソフトウェア保護と HSM 保護(HSM キー種別は Premium SKU)	RSA、EC、(Premium のみ)oct-HSM	`https://<vault-name>.vault.azure.net`
Managed HSMs	HSM 保護のみ	RSA-HSM、EC-HSM、oct-HSM	`https://<hsm-name>.managedhsm.azure.net`

Vaults は低コストで導入が容易な**マルチテナント**のゾーン冗長な鍵管理で、一般的なクラウド アプリケーションのシナリオに適します。**Managed HSMs** は**シングル テナント**の高可用な HSM で、価値の高いキーを扱う用途、最も厳格なセキュリティ・コンプライアンス・規制要件を満たす必要がある用途に適します。

Key Vault が解決する運用上の問題

- **アプリケーション シークレットの一元化**……アプリのコードに接続文字列を持たせる代わりに Key Vault に保管し、アプリは **URI** で特定のバージョンのシークレットを取得します。
- **監視**……ログを有効化し、ストレージ アカウントへのアーカイブ、イベント ハブへのストリーム、Azure Monitor ログへの送信が選べます。
- **管理の簡素化**……HSM の社内知識が不要になり、使用量の急増に短時間でスケールでき、**リージョン内および 2 次リージョンへ内容が複製**されます。複製は高可用性を確保し、**フェールオーバーのために管理者が何かをする必要はありません**。パブリック CA から購入した証明書の登録と更新は自動化されます。

- **分離**……アプリケーションごとに Key Vault を作り、シークレットを特定のアプリと開発チームに限定できます。

推奨される粒度は「アプリケーションごと、環境ごと(Development、Pre-Production、Production)に1つの vault を作り、ロールは vault スコープで割り当てる」です。

1-2-2 Key Vault の設定(Configure Key Vault settings) [1.2]

soft-delete(既定で有効・無効化できない)

soft-delete は、削除された vault と vault 内のオブジェクト(キー、シークレット、証明書)を復旧できるようにする機能です。**数字がそのまま問われます。**

項目	値
新しい vault を作成したときの既定	**soft-delete は既定で有効**
一度有効にした後	**無効にできません**
保持期間の設定範囲	**7〜90 日**
保持期間の既定値	**90 日**
保持期間を変更できるタイミング	**vault の作成時のみ。作成後は変更できません**
保持期間の適用範囲	soft-delete と purge protection の**両方の保持ポリシーに同じ間隔が適用されます**

シークレット、キー、証明書、または vault を恒久的に削除するには**2つの操作**が必要です。まずオブジェクトを削除して soft-deleted の状態にし、次にその状態のオブジェクトを **purge(消去)** します。soft-deleted の状態にあるシークレット、キー、証明書を purge するには、セキュリティ プリンシパルに **purge**

操作の権限(たとえば組み込みロールの **Key Vault Purge Operator**)が必要です。

soft-deleted になった Key Vault の名前は、保持期間が満了するまで再利用できません。保持期間中、同じ場所に同じ名前の Key Vault を作成することはできず、削除された状態の同名オブジェクトが vault にある場合はその vault 内に同名オブジェクトを作れません。

保持期間中にできることも整理しておきます。

- サブスクリプション内の soft-delete 状態の vault とオブジェクトを一覧し、削除と復旧の情報にアクセスできます。**ただし削除された vault を一覧できるのは特別な権限を持つユーザーだけ**で、この扱いのためのカスタム ロールの作成が推奨されています。
- 復旧できるのは特別な権限を持つユーザーだけです。リソース グループ配下に Key Vault を作成する権限を持つ、そのカスタム ロールのメンバーが vault を復元できます。
- 強制削除できるのも特別な権限を持つユーザーだけです。
- **復旧されない限り、保持期間の満了時にサービスが purge を実行します。リソースの削除は再スケジュールできません。**

削除された vault については、サブスクリプション配下に**プロキシ リソース**が作られ、復旧に十分なメタデータが保持されます。プロキシ リソースは削除された Key Vault と同じ場所に置かれます。**vault に対応する DNS レコードも保持期間中は保持されます。**

> **重要な落とし穴: Key Vault が soft-delete されると、その Key Vault と統合されていたサービスは削除されます。**たとえば Azure RBAC のロール割り当てや Event Grid のサブスクリプションです。**soft-deleted の Key**

> **Vault を復旧しても、これらのサービスは復元されません。作り直す必要があります。**

課金については、削除状態のオブジェクトに対して可能な操作は `purge` と `recover` だけで、他はすべて失敗するため課金は発生しません。ただし `purge` と `recover` 自体は通常の Key Vault 操作として課金されます。HSM キーの場合は、直近 30 日以内にキー バージョンが使われていれば「HSM Protected key」のキー バージョンごと月額料金がかかります。

purge protection(既定では無効)

purge protection は任意の動作で、**既定では有効になっていません。**有効化できるのは **soft-delete が有効になった後**だけです。

purge protection が有効な場合、**削除状態の vault やオブジェクトは保持期間が経過するまで purge できません。**soft-deleted の vault とオブジェクトは引き続き復旧でき、保持ポリシーが確実に守られます。

暗号化にキーを使う場合はデータ損失を防ぐために purge protection が推奨され、**Storage をはじめとする Azure Key Vault と統合する多くの Azure サービスは、データ損失を防ぐために purge protection を要求します。**有効化は CLI、PowerShell、ポータルから行えます。

purge の実行権限は、一般に**サブスクリプションの所有者**か **Key Vault Purge Operator** ロールを持つユーザーだけです。プロキシ リソースへの POST 操作で、その vault の即時かつ回復不能な削除が発生します。例外は2つあります。1つは Azure サブスクリプションが削除不可としてマークされている場合で、このときはサービスだけがスケジュールされた処理として実際の削除を行います。もう1つは vault 側で `--enable-purge-protection` 引数が有効

な場合で、このとき Key Vault は元のシークレット オブジェクトが削除対象としてマークされてから **7〜90 日**待ってから恒久的に削除します。

> **よく出る取り違え**: 「soft-delete を有効にしておけば、悪意ある管理者に消されても必ず戻せる」は誤りです。soft-delete だけなら、権限を持つ者が **purge して恒久削除できます**。「悪意ある削除でもデータを失いたくない」という要件なら、**purge protection が必須**です。soft-delete は事故のための機能、purge protection は悪意のための機能、と切り分けます。

Azure Policy を使って、**soft-delete の適用を vault に強制する**ことも推奨されています。soft-delete 保護が有効でない Key Vault では、キーを削除すると恒久的に削除されます。

1-2-3 Key Vault へのアクセスの構成(Configure access to Key Vault) [1.2]

Key Vault へのアクセスは**制御プレーン(control plane)** と**データ プレーン(data plane)** という2つのインターフェイスで制御されます。**この2面の区別が、この節のすべてです。**

面	エンドポイント	操作	アクセス制御の仕組み
制御プレーン	`management.azure.com:443`	Key Vault の作成・読み取り・更新・削除、Key Vault のアクセス ポリシーの設定、Key Vault のタグの設定	**Azure RBAC のみ**

面	エンドポイント	操作	アクセス制御の仕組み
データ プレーン	`<vault-name>.vault.azure.net:443`	**キー**: encrypt、decrypt、wrapKey、unwrapKey、sign、verify、get、list、create、update、import、delete、recover、backup、restore、purge、rotate、getrotationpolicy、setrotationpolicy、release **証明書**: managecontacts、getissuers、listissuers、setissuers、deleteissuers、manageissuers、get、list、create、import、update、delete、recover、backup、restore、purge **シークレット**: get、list、set、delete、recover、backup、restore、purge	**Key Vault access policy(レガシ)または Azure RBAC**

両方の面が **Microsoft Entra ID で認証**します。認可の仕組みだけが異なります。**2つの面のアクセス制御は独立して動作します。**アプリケーションに vault のキーを使わせたいなら Azure RBAC か access policy で**データ プレ**

ーンのアクセスを付与し、ユーザーに Key Vault のプロパティとタグの読み取りだけを許してデータには触らせたくないなら Azure RBAC で**制御プレーン**のアクセスを付与します。

権限モデルの既定が変わった点

API バージョン 2026-02-01 以降、新しく作成される Key Vault の既定のアクセス制御モデルは Azure RBAC です。Azure RBAC は Azure Resource Manager 上に構築された認可システムで、Azure リソースへのアクセス管理を一元化します。

Azure RBAC モデルでは、**管理グループ、サブスクリプション、リソース グループ、個々のリソース**という異なるスコープ レベルで権限を設定できます。**Key Vault の Azure RBAC は、個々のキー、シークレット、証明書に対して個別の権限を持たせることもできます。**

データ プレーンの Key Vault 組み込みロール

この表は対応付け問題で頻出です。すべて「Azure role-based access control 権限モデルを使う Key Vault でのみ動作する」点に注意します。

組み込みロール	できること
Key Vault Administrator	vault とその中のすべてのオブジェクト(証明書、キー、シークレット)に対して**すべてのデータ プレーン操作**を実行できます。**Key Vault リソースの管理やロール割り当ての管理はできません**
Key Vault Reader	vault とその証明書、キー、シークレットの**メタデータを読み取ります。シークレットの内容やキー マテリアルといった機微な値は読めません**

組み込みロール	できること
Key Vault Purge Operator	soft-deleted の vault の**恒久的な削除**を許可します
Key Vault Certificates Officer	vault の証明書に対する任意の操作。**権限の管理は除きます**
Key Vault Certificate User	秘密鍵部分を含む**証明書の内容全体**を読み取ります
Key Vault Crypto Officer	vault のキーに対する任意の操作。**権限の管理は除きます**
Key Vault Crypto Service Encryption User	キーのメタデータを読み取り、**wrap / unwrap 操作**を行います
Key Vault Crypto User	**キーを使った暗号操作**を行います
Key Vault Crypto Service Release User	Azure Confidential Computing と同等の環境向けにキーをリリースします
Key Vault Secrets Officer	vault のシークレットに対する任意の操作。**権限の管理は除きます**
Key Vault Secrets User	**シークレットの内容**(秘密鍵を持つ証明書のシークレット部分を含む)を読み取ります

Key Vault Contributor ロールは制御プレーンの操作専用で、vault の管理はできますが、**キー、シークレット、証明書へのアクセスは与えません。**ここは頻出の誤答肢です。

ロール割り当て自体の管理には **Key Vault Data Access Administrator** ロールがあります。これは Key Vault Administrator、Key Vault Certificates Officer、Key Vault Crypto Officer、Key Vault Crypto Service Encryption User、Key Vault Crypto User、Key Vault Reader、Key Vault Secrets

Officer、Key Vault Secrets User のロール割り当ての追加と削除だけを許し、**ABAC 条件でロール割り当てを制約**します。

Azure **属性ベースのアクセス制御(Azure ABAC)** 条件でこれらのロールをさらに制約できます。たとえば、名前が特定のパターンに一致するシークレットだけに絞る、vault 名で特定の vault だけに絞る、といった制約です。**プレビュー段階では、Key Vault の ABAC はシークレットのデータ アクションにのみ適用されます。**

> **重大な注意: Key Vault の制御プレーンに Contributor 権限を持つユーザーは、Key Vault access policy を設定することで自分自身にデータ プレーンへのアクセスを付与できます。**したがって、Key Vault に対する Contributor ロールを誰が持つかは厳密に管理する必要があります。これは「Contributor は制御プレーンだけだから安全」という誤答肢を切る根拠になります。

権限モデルを切り替えるときの権限と副作用

- 権限モデルの変更には、**制限のない** `Microsoft.Authorization/roleAssignments/write` **権限**が必要です。これは **Owner** と **User Access Administrator** ロールに含まれます。
- User Access Administrator を使う場合は、加えて `Microsoft.KeyVault/vaults/write` (**Key Vault Contributor** ロールに含まれます)が必要です。
- **Service Administrator や Co-Administrator といったクラシックなサブスクリプション管理者ロール、および制限付きの Key Vault Data Access Administrator では権限モデルを変更できません。**

- **Azure RBAC 権限モデルを設定すると、access policy の権限はすべて無効化されます。**同等の Azure ロールが割り当てられていないと**停止(outage)を招きます**。

ロール割り当ての管理には `Microsoft.Authorization/roleAssignments/write` と `Microsoft.Authorization/roleAssignments/delete` が必要で、これを持つのは Key Vault Data Access Administrator(特定の Key Vault ロールの割り当て/解除のみに制限)、User Access Administrator、Owner です。

割り当ての操作例は次のとおりです。

- Azure CLI…… `az role assignment create --role "Key Vault Secrets Officer" --assignee <user-principal-name> --scope /subscriptions/<subscription-id>/resourcegroups/<resource-group>/providers/Microsoft.KeyVault/vaults/<vault-name>`
- Azure PowerShell…… `New-AzRoleAssignment -RoleDefinitionName "Key Vault Secrets Officer" -SignInName <user-principal-name> -Scope "<スコープ>"`

スクリプトではロール名ではなく一意のロール ID を使うことが推奨されます。ロール名が変わってもスクリプトが動き続けるからです。**ロールがリネームされてもロール ID は変わりません。**

ロール割り当ての反映には数分かかります。ブラウザーはキャッシュを使うため、ロール割り当てを削除した後はページの更新が必要です。

個々のオブジェクトへのロール割り当ての是非

推奨は「アプリケーションごと・環境ごとに vault を分け、ロールは vault スコープで割り当てる」です。個々のキー、シークレット、証明書へのロール割り当ては**推奨されません。**例外は次の2つだけです。

- 個々のシークレットに個別のユーザー アクセスが必要な場合。たとえば **Azure Bastion** を使って仮想マシンに認証するために、ユーザーが自分の SSH 秘密鍵を読む必要がある場合です。
- 個々のシークレットを複数のアプリケーション間で共有する必要がある場合。たとえば、あるアプリケーションが別のアプリケーションのデータにアクセスする必要がある場合です。

> **よく出る設問**:「Azure RBAC のオブジェクト スコープの割り当てを使って、アプリケーション チームごとに Key Vault 内を分離できますか」に対する答えは **いいえ** です。個々のオブジェクトへのアクセスは割り当てられますが、**ネットワーク アクセス制御、監視、オブジェクト管理といった管理操作には vault レベルの権限が必要**で、その権限を与えるとアプリケーション チームをまたいで機密情報が露出してしまいます。

カスタム ロールでの絞り込み

組み込みロールで足りない場合はカスタム ロールを作ります。たとえばキーのバックアップと復元だけを許すロールは、`DataActions` に `Microsoft.KeyVault/vaults/keys/read`、`Microsoft.KeyVault/vaults/keys/backup/action`、`Microsoft.KeyVault/vaults/keys/restore/action` を指定して `az role definition create` で作成します。

1-2-4 Key Vault のファイアウォール設定(Configure firewall settings on Key Vault) [1.2]

既定の状態

新しい Key Vault を作成すると、Azure Key Vault ファイアウォールは既定で無効です。すべてのアプリケーションと Azure サービスが Key Vault にアクセスし、要求を送れます。ただしこれは「誰でも操作できる」という意味ではありません。Key Vault は依然として **Microsoft Entra 認証とアクセス許可**を要求して、シークレット、キー、証明書へのアクセスを制限します。

4つの構成パターン

構成	ポータルでの選択	何が通るか
ファイアウォール無効(既定)	Allow public access from all networks	すべて通ります(認証と認可は別途必要)
信頼された Microsoft サービスのバイパス	Exception の **Allow trusted Microsoft services to bypass this firewall**	信頼済みサービス一覧にあるサービスだけがファイアウォールを迂回します
IPv4 アドレスと範囲(静的 IP)	Firewall に IPv4 アドレスまたは CIDR 範囲を入力	指定した公開 IP からのみ
仮想ネットワーク(動的 IP)	Allow public access from specific virtual networks and IP addresses	サービス エンドポイントを有効にしたサブネットからのみ
パブリック アクセス無効	**Disable public access**	プライベート エンドポイント経由のみ(信頼済みサービスのバイパスは残ります)

信頼済みサービスのバイパスについては、次の点が問われます。

- **信頼済みサービスの一覧はすべての Azure サービスを網羅していません。**たとえば **Azure DevOps は信頼済みサービスの一覧に含まれません。**
- これは「一覧にないサービスが信頼できない、安全でない」という意味ではありません。**一覧に含まれるのは、そのサービス上で実行されるコードをすべて Microsoft が制御しているサービス**です。Azure DevOps のようにユーザーがカスタム コードを書けるサービスには、包括的な承認を与える選択肢が用意されていません。
- **一覧にないサービスは、バイパス オプションを有効にしていてもファイアウォールでブロックされます。**
- `publicNetworkAccess` **を Disabled にしても、信頼済みサービスへのバイパスは適用され続けます。**これらのサービスは vault に到達するのにプライベート エンドポイントを必要としません。
- ただし、**Network Security Perimeter に関連付けて publicNetworkAccess を「Secure by perimeter」にすると、バイパスは上書きされ、明示的なペリメーター アクセス規則がない限り信頼済みサービスもブロックされます。**

設定コマンドは次のとおりです。

- Azure CLI……`az keyvault update --resource-group "myresourcegroup" --name "mykeyvault" --bypass AzureServices`
- Azure PowerShell……`Update-AzKeyVaultNetworkRuleSet -VaultName "mykeyvault" -Bypass AzureServices`

仮想ネットワーク規則の設定手順は3段階で、並べ替え問題になります。

順	操作	コマンド例
1	サブネットで Key Vault のサービス エンドポイントを有効化する	`az network vnet subnet update --resource-group "myresourcegroup" --vnet-name "myvnet" --name "mysubnet" --service-endpoints "Microsoft.KeyVault"`
2	vault にその仮想ネットワークとサブネットのネットワーク規則を追加する	`az keyvault network-rule add --resource-group "myresourcegroup" --name "mykeyvault" --subnet $subnetid`
3	既定のアクションを拒否にする	`az keyvault update --resource-group "myresourcegroup" --name "mykeyvault" --default-action Deny`

ポータルの場合、サービス エンドポイントが有効でなければ有効化を促され、**反映には最大 15 分**かかることがあります。

IP 規則の追加は `az keyvault network-rule add --resource-group "myresourcegroup" --name "mykeyvault" --ip-address "191.10.18.0/24"` または `Add-AzKeyVaultNetworkRule -VaultName "mykeyvault" -IpAddressRange "16.17.18.0/24"` です。

構成の上限と制約(数字がそのまま出ます)

- **仮想ネットワーク規則は最大 200、IPv4 規則は最大 1000。**
- **IP ネットワーク規則はパブリック IP アドレスのみ許可されます。**RFC 1918 で定義されたプライベート ネットワーク用に予約された範囲(**10. で始まるもの、172.16-31、192.168.**)は IP 規則に使えません。
- **現時点で IPv4 アドレスのみサポート**されます。

ファイアウォールが効いた後の影響

ファイアウォール規則が有効になると、ユーザーは許可された仮想ネットワークまたは IPv4 アドレス範囲から要求が発信された場合にのみ、Key Vault のデータ プレーン操作を実行できます。これは Azure ポータルからのアクセスにも適用されます。ユーザーはポータルから Key Vault を開けても、クライアント マシンが許可一覧にない場合はキー、シークレット、証明書を一覧できないことがあります。他の Azure サービスが使う Key Vault ピッカーにも同じ影響が出ます。

Azure Key Vault のファイアウォール規則はデータ プレーンの操作にのみ適用されます。制御プレーンの操作は、ファイアウォール規則で指定した制限を受けません。これには **ARM テンプレートによるシークレットやキーのデプロイ**が含まれます。ARM テンプレートは Key Vault のデータ プレーン エンドポイントではなく、Azure Resource Manager の制御プレーン エンドポイント(`management.azure.com`)を使うからです。

ファイアウォールが効くのはデータ プレーンの操作だけで、制御プレーンの操作は制限を受けません。
ARM テンプレートは制御プレーン エンドポイントを使うため、ファイアウォールでは止まりません。
両方の面が Microsoft Entra ID で認証し、認可の仕組みだけが異なります。

図 1-6　Key Vault の 2 つの面と、ファイアウォールが効く位置

> **誤答肢の切り方**:「Key Vault のファイアウォールを有効にすれば、ARM テンプレートからのシークレット参照も社内ネットワークからしか行えなくなる」は誤りです。ファイアウォールはデータ プレーン専用です。

パブリック アクセスの無効化とプライベート エンドポイント

ネットワーク セキュリティを高めるため、vault のパブリック アクセスを無効化できます。これによりすべてのパブリック構成が拒否され、**プライベート エンドポイント経由の接続のみ**が許可されます。パブリック アクセスを無効化した後も信頼済みサービスのバイパスは残ります。**信頼済みサービス一覧にない Microsoft サービス(たとえば Azure DevOps)はブロックされ、プライベート エンドポイント経由での接続が必要**になります。

パブリック アクセスを無効化しても、公開 DNS からは Key Vault の FQDN(`<vault-name>.vault.azure.net` と `privatelink.vaultcore.azure.net` へ

の CNAME)が解決され続けます。これは設計どおりで、すべての Azure PaaS サービスが使う Private Link の DNS オーバーレイ モデルに必要な動作です。公開インターネットから `dig` や `nslookup` を行うと Key Vault のパブリック リージョン受信 IP が返りますが、**パブリック アクセスが無効ならその受信点はすべての要求を拒否します。**公開 DNS の解決から、プライベート エンドポイントの有無、そのプライベート IP、vault のネットワーク規則、データ プレーンの内容が漏れることはありません。

設定は `az keyvault update --resource-group "myresourcegroup" --name "mykeyvault" --public-network-access Disabled` または `Update-AzKeyVault -ResourceGroupName "myresourcegroup" -VaultName "mykeyvault" -PublicNetworkAccess "Disabled"` です。**パブリック アクセスを無効化する前に、プライベート エンドポイントの構成を完了させておく必要があります。**

Network Security Perimeter

Network Security Perimeter は、組織の仮想ネットワークの外にデプロイされた PaaS リソース(Azure Key Vault、Azure Storage、SQL Database など)に対して、論理的なネットワーク分離境界を定義します。ペリメーターの外からのパブリック ネットワーク アクセスを制限し、明示的なアクセス規則で受信・送信の例外を作れます。

- ペリメーター内のすべてのリソースは、ペリメーター内の他のリソースと通信できます。
- パブリックの受信アクセスは、送信元 IP アドレスやサブスクリプションといったクライアントのネットワーク属性・ID 属性で承認できます。
- パブリックの送信は、外部宛先の FQDN で承認できます。
- ペリメーター内の PaaS リソースでは、監査とコンプライアンスのための診断ログが有効になります。

プライベート エンドポイントのトラフィックは非常に安全とみなされるため、Network Security Perimeter の規則の対象外です。信頼済みサービスを含むそれ以外のすべてのトラフィックは、vault がペリメーターに関連付けられていればペリメーターの規則に従います。

アクセス モードは2つです。

モード	動作
Transition mode(旧 Learning mode)	**既定のアクセス モード。**強制モードだったら拒否されていたはずのトラフィックをすべてログに記録します。強制の前に既存のアクセス パターンを把握するために使います
Enforced mode	アクセス規則で明示的に許可されていないトラフィックをログに記録し、**拒否します**

`publicNetworkAccess` の設定は、Transition mode ではリソースへのパブリック アクセスを制御しますが、**Enforced mode では Network Security Perimeter の規則で上書きされます。**

Azure Key Vault には送信規則の概念がありません。送信規則を持つペリメーターに Key Vault を関連付けることはできますが、Key Vault はそれを使いません。

1-2-5 キー、シークレット、証明書の管理(Manage keys, secrets, and certificates) [1.2]

Key Vault の暗号化キーは **JSON Web Key (JWK)** オブジェクトです。JWK、JWE、JWA、JWS の各仕様を使い、Key Vault 独自のキー種別のために基本仕様を拡張しています。

vault は **FIPS 140 検証済みの HSM** でキーを保護します。**HSM プラットフォームは2つ**あり、**HSM Platform 1 は FIPS 140-2 Level 2**、**HSM Platform 2 は FIPS 140-3 Level 3** でキー バージョンを保護します。**新しいキーとキー バージョンはすべて HSM Platform 2 が保護します。**どちらのプラットフォームが保護しているかは、キー バージョンの `hsmPlatform` 属性で確認できます。

キーの種類とサイズ

保護方法	キー種別	サイズ / 曲線
HSM 保護(Key Vault Premium)	**EC-HSM**(楕円曲線)	P-256、P-384、P-521、secp256k1/P-256K
	RSA-HSM	2048 ビット、3072 ビット、4096 ビット
	oct-HSM(対称鍵 AES、プレビュー)	128 ビット、192 ビット、256 ビット
ソフトウェア保護 (Standard と Premium)	**RSA**	2048 ビット、3072 ビット、4096 ビット
	EC	P-256、P-384、P-521、secp256k1/P-256K

コンプライアンス レベルの対応は次のとおりです。

キー種別と配置先	コンプライアンス
vault のソフトウェア保護キー(HSM Platform 0)	**FIPS 140-2 Level 1**
vault の HSM Platform 1 保護キー (Premium SKU)	**FIPS 140-2 Level 2**

キー種別と配置先	コンプライアンス
vault の HSM Platform 2 保護キー (Premium SKU)	**FIPS 140-3 Level 3**

対称鍵(oct-HSM) は Key Vault Premium で**パブリック プレビュー**です。vault の対称鍵は **HSM バックアップのみ**で、**Premium vault が必須**です。サポートされる操作は `encrypt`、`decrypt`、`wrapKey`、`unwrapKey`、アルゴリズムは暗号化とラップに AES-KW、AES-GCM、AES-CBC、HMAC の署名/検証に HS256、HS384、HS512 です。作成例は `az keyvault key create --vault-name <vault-name> --name sample-aes-key --kty oct-HSM --size 256 --ops encrypt decrypt wrapKey unwrapKey` です。

AES-GCM の操作では、暗号化応答の `iv`、`result`、`tag` の3つの値を保存する必要があります。復号にはこの3つすべてが必要です。

耐量子性については、Azure Key Vault Premium(プレビュー)と Managed HSM が提供する AES アルゴリズムで使う **256 ビットの対称鍵(oct-HSM / AES)が耐量子(quantum-resistant)** とされています。

キーの主な使いどころ

いつ使うか	例
統合されたリソース プロバイダーによる**サーバー側のデータ暗号化**をカスタマー マネージド キーで行うとき	Azure Key Vault のカスタマー マネージド キーによるサーバー側暗号化
クライアント側のデータ暗号化	Azure Key Vault によるクライアント側暗号化
キーレス TLS	Key Vault のクライアント ライブラリを使用

Key Vault は、Azure Disk Encryption、SQL Server と Azure SQL Database の Always Encrypted と Transparent Data Encryption 、Azure App Service との統合シナリオを簡素化します。Key Vault 自体もストレージ アカウント、イベント ハブ、Log Analytics と統合できます。

シークレットと証明書

vault は暗号化キーに加えて、**シークレット、証明書、ストレージ アカウント キー**といった複数種類のオブジェクトを保管・管理できます。アプリケーションは **URI** を使って必要な情報に安全にアクセスし、**特定のバージョンのシークレット**を取得できます。シークレット情報を保護するためのカスタム コードを書く必要はありません。

証明書については、**パブリック CA から購入した証明書の登録と更新といった一部の作業が自動化**されます。

1-2-6 Defender Cloud Security Posture Management (Defender CSPM) によるシークレットのスキャン [1.2]

Defender Cloud Security Posture Management (Defender CSPM) は、攻撃者がセキュリティ シークレットを悪用するリスクを最小化する機能を提供します。

攻撃者は初期アクセスを得た後、ネットワーク内を**横方向に移動(lateral movement)** し、リソースにアクセスして脆弱性を突き、重要な情報システムに損害を与えようとします。横方向の移動では、**露出した資格情報やシークレット(パスワード、キー、トークン、接続文字列)** を悪用する資格情報の脅威が典型です。

シークレットはマルチクラウド環境のファイル、VM のディスク、コンテナーの中に散在します。**露出が起きる理由**は、リスクへの認識不足、明確な社内ポリシ

ーの欠如、検出ツールの不在、環境の複雑さと開発速度、そしてセキュリティと使いやすさのトレードオフです。

スキャンの種類とプラン(対応付け問題の中心)

スキャンの種類	内容	どのプランに含まれるか
マシンのスキャン	マルチクラウドの VM に対する**エージェントレスのシークレット スキャン**	**Defender CSPM プラン**、または **Defender for Servers Plan 2**
クラウド デプロイ リソースのスキャン	マルチクラウドの **infrastructure-as-code デプロイ リソース**に対するエージェントレスのシークレット スキャン	**Defender CSPM プランのみ**
コード リポジトリのスキャン	Azure DevOps で露出したシークレットを発見するスキャン	**Defender CSPM プランのみ**

> **よく出る取り違え**:「Defender for Servers Plan 2 があれば IaC デプロイ リソースのシークレットもスキャンされる」は誤りです。Plan 2 が含むのは**マシンのスキャンだけ**で、クラウド デプロイ リソースとコード リポジトリのスキャンは **Defender CSPM が必要**です。

シークレット スキャンに必要な権限は、Security Reader、Security Admin、Reader、Contributor、Owner のいずれかです。

検出結果の確認方法(4つ)

- **資産インベントリでの確認**……Defender for Cloud に接続されたリソースのセキュリティ状態を示すインベントリから、特定のマシンで見つかった

シークレットを確認します。

- **推奨事項での確認**……資産上でシークレットが見つかると、Defender for Cloud の推奨事項ページの **Remediate vulnerabilities** セキュリティ制御の下に推奨事項が発生します。
- **cloud security explorer での確認**……クラウド セキュリティ グラフに対してクエリを実行し、シークレットの洞察を得ます。自分でクエリを作ることも、組み込みテンプレートを使って環境全体の VM シークレットを問い合わせることもできます。
- **attack paths(攻撃パス)での確認**……攻撃パス分析がクラウド セキュリティ グラフを走査し、攻撃者が環境を侵害して高影響資産に到達し得る**悪用可能な経路**を明らかにします。VM のシークレット スキャンは複数の攻撃パス シナリオに対応します。

すべての方法がすべてのシークレット種別で使えるわけではありません。この点も出題されます。

検出されるシークレットの種類

インベントリ、cloud security explorer、推奨事項、攻撃パスのすべてで扱える主なシークレットは次のとおりです。

- 安全でない SSH 秘密鍵(PuTTy ファイルの RSA アルゴリズム、PKCS#8 と PKCS#1 標準、OpenSSH 標準に対応)
- 平文の Azure SQL 接続文字列
- 平文の Azure Database for PostgreSQL / MySQL
- 平文の Azure Cosmos DB(PostgreSQL、MySQL を含む)
- 平文の AWS RDS 接続文字列(Amazon Aurora の Postgres / MySQL、Oracle と SQL Server のカスタム RDS)

- 平文の Azure ストレージ アカウント接続文字列
- 平文の Azure ストレージ アカウント **SAS トークン**
- 平文の AWS アクセス キー
- 平文の AWS S3 の事前署名済み URL

インベントリと cloud security explorer でのみ扱える主なシークレットには、Google ストレージの署名付き URL、**Microsoft Entra ID Client Secret**、Azure DevOps の個人用アクセス トークン、GitHub の個人用アクセス トークン、Azure App Configuration のアクセス キー、Azure Cognitive Service のキー、Microsoft Entra のユーザー資格情報、**Azure Container Registry のアクセス キー**、Azure App Service のデプロイ パスワード、Azure Databricks の個人用アクセス トークン、Azure SignalR のアクセス キー、Azure API Management のサブスクリプション キー、Azure Machine Learning の Web サービス API キー、Azure Communication Services のアクセス キー、Azure Event Grid のアクセス キー、Azure Maps のサブスクリプション キー、Azure Web PubSub のアクセス キー、**OpenAI API キー**、Azure Batch の共有アクセス キー、NPM の作成者トークン、Azure サブスクリプション管理証明書などがあります。

VM のスキャンでは検出されず、クラウド デプロイ リソースのスキャンでのみ検出される種類もあります。GCP API キー、AWS Redshift の資格情報、平文の秘密鍵、ODBC 接続文字列、一般的なパスワード、ユーザー ログイン資格情報、Travis の個人用トークン、Slack のアクセス トークン、ASP.NET のマシン キー、HTTP Authorization ヘッダー、Azure Redis Cache のパスワード、Azure IoT の共有アクセス キー、Azure DevOps のアプリ シークレット、Azure Function の API キー、Azure の共有アクセス キー、Azure Logic App の共有アクセス署名、Azure Active Directory のアクセス トークン、Azure Service Bus の共有アクセス署名などです。

1-2-7 Defender for Key Vault の実装 [1.2]

Microsoft Defender for Key Vault を有効にすると、Azure Key Vault に対する **Azure ネイティブの高度な脅威保護**が追加されます。リリース状態は **一般提供(GA)** で、課金は Defender for Cloud の価格ページに従います。Defender for Cloud のコスト計算ツールで見積もることもできます。

Defender for Key Vault は、**Key Vault アカウントへのアクセスや悪用を試みる異常で潜在的に有害な操作を検出**します。この保護層により、セキュリティの専門家でなくても、サードパーティのセキュリティ監視システムを管理せずに脅威に対処できます。異常なアクティビティが発生すると**アラートを表示**し、必要に応じて組織の関係者に**メールで送信**します。アラートには不審なアクティビティの詳細と、調査・修復の推奨が含まれます。

アラートは3か所に現れます。**Key Vault の Security ページ**、**Workload protections**、**Defender for Cloud のセキュリティ アラート ページ**です。

アラートに含まれる要素は次の2つです。

- **Object ID**
- 不審なリソースの **User Principal Name または IP アドレス**

アクセスの種類によっては、一部のフィールドが得られません。アプリケーションが Key Vault にアクセスした場合は User Principal Name が表示されず、トラフィックが Azure の外から来た場合は Object ID が表示されません。

Azure の仮想マシンには Microsoft の IP が割り当てられます。そのため、Microsoft の外で行われた操作に関するアラートであっても Microsoft の IP が含まれることがあります。**アラートに Microsoft の IP が含まれていても調査は必要**です。

対応の4ステップ(並べ替え問題の根拠)

順	段階	何をするか
1	**送信元の特定**	トラフィックが自分の Azure テナント内から来たかを確認します。Key Vault ファイアウォールが有効なら、アラートを引き起こしたユーザーやアプリケーションにアクセスを与えている可能性が高いです。送信元を特定できたら、そのユーザーまたはアプリケーションの所有者に連絡します
2	**適切な対応**	不明な IP アドレスからなら**Key Vault ファイアウォールを有効化**し、信頼できるリソースと仮想ネットワークで構成します。不正なアプリケーションや不審なユーザーが原因なら、Key Vault の**アクセス ポリシー設定を開き、該当のセキュリティ プリンシパルを削除するか実行できる操作を制限**します。送信元がテナントの Microsoft Entra ロールを持つなら管理者に連絡し、Microsoft Entra の権限を削減または取り消す必要があるか判断します

順	段階	何をするか
3	**影響範囲の測定**	Key Vault の **Security ページ**で発生したアラートを開き、アクセスされたシークレットの一覧とタイムスタンプを確認します。Key Vault の診断ログを有効にしていれば、該当する呼び出し元 IP、ユーザー プリンシパル、オブジェクト ID の過去の操作も確認します
4	**対処**	不審なユーザーやアプリケーションがアクセスしたシークレット、キー、証明書の一覧をまとめたら、**直ちにローテーション**します。影響を受けたシークレットは Key Vault で無効化または削除します。資格情報が特定のアプリケーションで使われていた場合は、そのアプリケーションの管理者に連絡し、侵害後に資格情報が使われた形跡がないか環境を監査してもらいます

> **重要な注意: Defender for Key Vault は、盗まれた資格情報による不審なアクティビティを特定するための機能です。ユーザーやアプリケーションに見覚えがあるという理由だけでアラートを却下してはいけません。**アプリケーションの所有者やユーザーに連絡し、その操作が正当だったかを確認します。ノイズを減らす必要がある場合は**抑制規則(suppression rules)** を作ります。

1-2-8 到達目標 1.2 の分かれ目の早見表 [1.2]

要件の言葉	正解になる構成	よく置かれる誤答肢
誤って削除したシークレットを戻したい	**soft-delete**(既定で有効、7〜90 日、既定 90 日)	バックアップからの復元(復元の手間が大きく、最小の管理負荷にはなりません)
悪意ある管理者による恒久削除を防ぎたい	**purge protection**(既定では無効。soft-delete が前提で、有効化後は保持期間の満了まで purge 不可)	soft-delete のみ(権限があれば purge できます)
キー、シークレット、証明書ごとに別々の権限を与えたい	**Azure RBAC 権限モデル**(オブジェクト スコープのロール割り当て)	access policy(オブジェクト単位の粒度を持ちません)
チームごとに Key Vault 内を分離したい	**アプリケーションごと・環境ごとに vault を分ける**	オブジェクト スコープのロール割り当てで分離する(管理操作には vault レベル権限が要り、分離になりません)
シークレットの中身は見せず一覧だけ見せたい	**Key Vault Reader**	Key Vault Secrets User(内容を読めてしまいます)
キーで暗号化・復号だけさせたい	**Key Vault Crypto User**	Key Vault Crypto Officer(キーの作成・削除までできてしまいます)
特定の仮想マシンからだけアクセスさせたい(動的 IP)	サブネットで **Microsoft.KeyVault サービス エンドポイント**を有効化し、仮想ネットワーク規則を追加して default-action を Deny	IP 規則(プライベート IP は登録できません)

要件の言葉	正解になる構成	よく置かれる誤答肢
インターネットからのデータプレーン アクセスを完全に断ちたい	**Disable public access + プライベート エンドポイント**	信頼済みサービスのバイパスのみ(バイパスは残ります)
信頼済みサービスも含めて完全に遮断したい	**Network Security Perimeter を Enforced mode** で関連付ける	publicNetworkAccess を Disabled にする(信頼済みサービスは通ります)
VM のディスクに残った接続文字列を見つけたい	**Defender CSPM または Defender for Servers Plan 2** のマシンのエージェントレス シークレット スキャン	Defender for Key Vault(vault へのアクセスを見る機能で、ディスク上の平文は見ません)
IaC のデプロイ リソースに埋め込まれたシークレットを見つけたい	**Defender CSPM**(クラウド デプロイ リソースのスキャン)	Defender for Servers Plan 2(マシンのスキャンのみ)
vault への異常なアクセスを検知したい	**Defender for Key Vault**	診断ログの有効化のみ(記録は残りますが検知とアラートはありません)

1-3 セキュリティと規制のコンプライアンスを適用するためのガバナンスを実装する [1.3]

この到達目標は「**組織のルールを、人の善意ではなく仕組みで守らせる**」範囲です。ルールを機械に強制させる Azure Policy、ルールとの差分を測る Defender for Cloud の regulatory compliance と security standards、削除を物理的に止める resource locks、権限を配る Azure built-in role assignments と custom roles、配りすぎた権限を回収する overprivileged

access の是正、最後の砦であるバックアップの保護、そしてすべてをコードで再現する infrastructure as code の9本です。

この節の全体像を先に置きます。

何をしたいか	使う道具	効き方
誰が操作しても構成が基準から外れないようにしたい	**Azure Policy**	リソースの**状態**を評価し、作成・更新を拒否したり自動で直したりします
誰が何をできるかを決めたい	**Azure RBAC**	**ユーザーの操作**を許可・拒否します
すべてのユーザーとロールに対して削除・変更を一律で止めたい	**resource locks**	RBAC を**上書き**して全員に制限をかけます
基準からどれだけ離れているかを測りたい	**Defender for Cloud** の security standards と recommendations	継続的に評価し、推奨事項とセキュア スコアで示します
規制フレームワークへの適合を示したい	**regulatory compliance** ダッシュボード	Azure Policy のイニシアティブとして評価します

1-3-1 Azure Policy による security controls の実装と構成 — built-in and custom policy definitions [1.3]

Azure Policy は、組織の標準を大規模に適用し、コンプライアンスを評価します。コンプライアンス ダッシュボードで環境全体の状態を集約して見せ、**リソースごと・ポリシーごとの粒度までドリルダウン**できます。既存リソースの一括修復と、新規リソースの自動修復も提供します。

Azure Policy でできるガバナンスの例は、許可されたリージョンにしかリソースをデプロイさせない、タグの適用を一貫させる、リソースに診断ログを Log

Analytics ワークスペースへ送らせる、要求を出した ID に基づいてリソース操作をブロックする(**multifactor authentication (MFA) を要求する、重要なリソースの削除をユーザーに行わせない**)などです。**Azure Arc により、このポリシー ベースのガバナンスを他のクラウド プロバイダーやオンプレミスのデータセンターにまで広げられます。**

Azure Policy のすべてのデータとオブジェクトは保存時に暗号化されます。

3つのオブジェクト

オブジェクト	内容
policy definition(ポリシー定義)	ビジネス ルールを **JSON 形式**で記述したもの。メタデータとポリシー ルールを含み、ルールは関数、パラメーター、論理演算子、条件、プロパティの**エイリアス**を使ってシナリオを正確に表現します
initiative definition(イニシアティブ定義、policySet)	1つの大きな目的に向けた**ポリシー定義の集合**。SDK(Azure CLI、Azure PowerShell)では **PolicySet** という名前のプロパティやパラメーターで参照されます
assignment(割り当て)	定義またはイニシアティブを特定の**スコープ**に割り当てたもの

組み込み(built-in)のポリシー定義は既定で用意されています。代表例は次のとおりです。

組み込み定義	効果	内容
Allowed Storage Account SKUs	Deny	デプロイされるストレージ アカウントが許可された SKU の集合に入っているか判定します

組み込み定義	効果	内容
Allowed Resource Type	Deny	デプロイできるリソースの種類を定義し、それ以外を拒否します
Allowed Locations	Deny	新しいリソースの場所を制限し、地理的なコンプライアンス要件を強制します
Allowed Virtual Machine SKUs	Deny	デプロイできる仮想マシンの SKU を指定します
Add a tag to resources	Modify	デプロイ要求で指定されていない場合に、必要なタグと既定値を適用します
Not allowed resource types	Deny	指定した種類のリソースのデプロイを防ぎます

カスタム(custom)のポリシー定義は、組み込みで足りない場合に JSON で作成します。**パラメーター**を定義することで、1つの定義を異なる値で使い回せます。たとえば `location` というパラメーターを定義し、割り当て時に `EastUS` や `WestUS` を渡します。**パラメーターを使うと、作成すべきポリシー定義の数を減らせます。**

効果(effects)の一覧と評価順序

各ポリシー定義は `policyRule` の中に **1つの** `effect` を持ちます。サポートされる効果は次のとおりです。

- `addToNetworkGroup`
- `append`
- `audit`

- `auditIfNotExists`
- `deny`
- `denyAction`
- `deployIfNotExists`
- `disabled`
- `manual`
- `modify`
- `mutate`

評価の順序は並べ替え問題としてそのまま出ます。Azure Policy はまずリソースの作成・更新要求を評価し、そのリソースに適用されるすべての割り当てを列挙してから、各定義に照らして評価します。**Resource Manager モードでは、要求を適切なリソース プロバイダーに渡す前に、いくつかの効果を先に処理**します。これにより、Azure Policy のガバナンス制御を満たさないリソースをリソース プロバイダーが無駄に処理せずに済みます。**Resource Provider モードでは、リソース プロバイダーが評価と結果を管理し、結果を Azure Policy に報告します。**

順	効果	理由
1	`disabled`	ポリシー ルールを評価すべきかどうかを最初に判定します
2	`append` と `modify`	どちらも要求を変更し得るため、変更によって audit や deny が発動しなくなる可能性があります。**この2つは Resource Manager モードでのみ利用できます**

順	効果	理由
3	`deny`	audit の前に評価することで、望ましくないリソースの**二重のログ記録を防ぎます**
4	`audit`	—
5	`manual`	—
6	`auditIfNotExists`	—
7	`denyAction`	**最後に評価されます**

リソース プロバイダーが Resource Manager モードの要求に成功コードを返した後、`auditIfNotExists` と `deployIfNotExists` が評価され、追加のコンプライアンス ログや処理が必要かどうかを判断します。

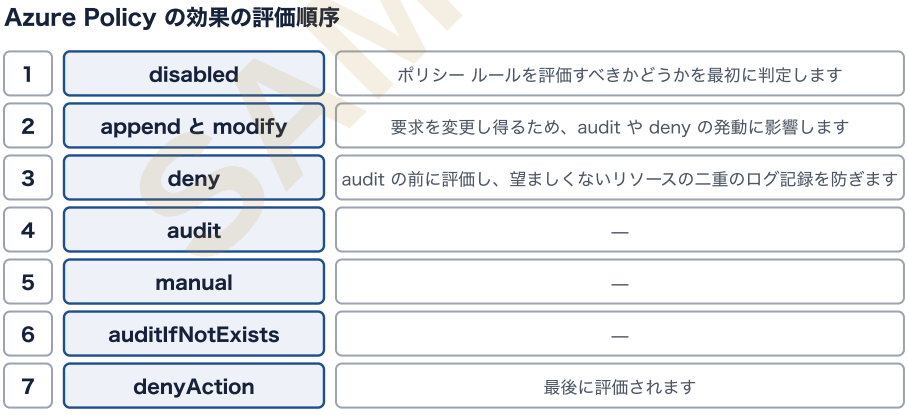


図 1-7　Azure Policy の効果の評価順序

`tags` 関連のフィールドのみを変更する `PATCH` 要求では、ポリシー評価は `tags` 関連フィールドを検査する条件を含むポリシーに限定されます。

効果の互換性

複数の効果が同じポリシー定義に対して有効なことがあります。**allowedValues** を使って割り当て時に効果を選べるようにするのが一般的です。ただし、**すべての効果が交換可能なわけではありません。**

- `audit`、`deny`、そして `modify` または `append` は、しばしば交換可能です。
- `auditIfNotExists` と `deployIfNotExists` は、しばしば交換可能です。
- **`manual` は交換できません。**
- **`disabled` はどの効果とも交換可能です。**

`audit` はリソース自身のプロパティに基づいてコンプライアンスを評価しますが、**`auditIfNotExists` は子リソースまたは拡張リソースのプロパティに基づいて評価**します。したがって `auditIfNotExists` のポリシー ルールには、`audit` では不要な追加の詳細が必要です。

評価が起きるタイミング

リソースは、リソースのライフサイクル、ポリシー割り当てのライフサイクル、そして定期的な継続評価の中の特定のタイミングで評価されます。

- ポリシー割り当てのあるスコープで**リソースが作成または更新された**とき
- スコープに**ポリシーまたはイニシアティブの新しい割り当てが行われた**とき
- スコープに**すでに割り当てられているポリシーまたはイニシアティブが更新された**とき
- **24 時間ごとに 1 回**行われる標準のコンプライアンス評価サイクル

割り当てのスコープと除外

割り当てのスコープは、**管理グループから個々のリソースまで**の範囲です。**すべての子リソースが割り当てを継承します。**リソース グループに適用された定義は、そのリソース グループ内のリソースにも適用されます。ただし、**サブスコープを除外(exclusion / notScopes)することはできます。**

重要な性質として、**Azure Policy は明示的な拒否のシステム**です。管理グループ レベルで許可リストの定義を割り当て、子の管理グループやサブスクリプションでより緩やかなポリシーを割り当てても、**緩い方が勝つことはありません。**この場合の正しい構成は、**管理グループ レベルの割り当てから子の管理グループやサブスクリプションを除外し、そこにより緩やかな定義を割り当てる**ことです。

割り当ては常に、割り当てられた定義またはイニシアティブの最新の状態を使って評価します。割り当て済みのポリシー定義が変更されると、その定義の既存の割り当てはすべて更新後のロジックで評価されます。

効果の重ね合わせ(layering)

複数の割り当てが同じリソースに影響することがあります。**各ポリシーの条件と効果は独立して評価されます。**重ね合わせの結果は「**累積的に最も制限が厳しい(cumulative most restrictive)**」ものになります。両方が `deny` なら、重複し矛盾するポリシー定義によってリソースはブロックされます。そのスコープでリソースを作りたい場合は、**各割り当ての除外(exclusions)を見直す**必要があります。

具体例で確認します。ポリシー1が場所を `westus` に制限し、サブスクリプション A に `deny` 効果で割り当てられ、ポリシー2が場所を `eastus` に制限し、サブスクリプション A のリソース グループ B に `audit` 効果で割り当てられている場合、次のようになります。

- リソース グループ B の `eastus` にある既存リソースは、ポリシー2には準拠、ポリシー1には非準拠です。
- リソース グループ B にあり `eastus` にない既存リソースは、ポリシー2に非準拠で、`westus` にもなければポリシー1にも非準拠です。
- ポリシー1は、サブスクリプション A の `westus` 以外の新しいリソースを拒否します。
- サブスクリプション A のリソース グループ B の `westus` に作られる新しいリソースは作成され、ポリシー2には非準拠になります。

評価対象になるリソース

ポリシーは管理グループ レベルで割り当てられますが、評価されるのはサブスクリプションまたはリソース グループ レベルのリソースだけです。Machine configuration、Azure Kubernetes Service、Azure Key Vault といった一部のリソース プロバイダーには、設定やオブジェクトを管理するためのより深い統合(**Resource Provider modes**)があります。

ID ベースのアクションの管理

Azure Policy はリソースのプロパティだけでなく、**リソースに対して行われたアクション**も評価します。**誰が要求したか**に基づいて強制を行えます。鍵になるのはポリシー ルールの `requestContext().identity` プロパティで、要求者の ID 情報にアクセスできます。

- 任意のクライアントからユーザーが要求を開始したかどうかを確認する条件を追加して、**重要なリソースの手動削除をユーザーにさせない**ようにできます。
- **MFA が有効でないユーザーが開始したリソースの作成・更新・削除要求をブロック**して、全ユーザーに MFA を要求できます。

修復(remediation)と managed identity

効果は主にリソースの作成・更新時に働きますが、Azure Policy は**既存の非準拠リソースを変更せずに扱う**ことも、**修復して準拠させる**こともできます。

`deployIfNotExists` または `modify` のポリシー割り当ての managed identity には、**対象リソースを作成または更新するのに十分な権限が必要です。**この managed identity に必要な権限を付与するには **User Access Administrator** が必要です。

Azure Policy と Azure RBAC の違い(頻出)

観点	Azure Policy	Azure RBAC
何を評価するか	Resource Manager に表現されたリソースの**プロパティ**と、一部のリソース プロバイダーのプロパティ	各スコープでの**ユーザーのアクション**
誰が変更したかを気にするか	**気にしません。**誰が変更したか、誰に権限があるかに関わらず、リソースの状態がビジネス ルールに準拠していることを保証します	**気にします。**ユーザー情報に基づいてアクションを制御する必要があるならこちらです
可視性	ポリシー定義、イニシアティブ定義、割り当てといった一部のリソースは**すべてのユーザーから参照できます。**環境にどんなポリシー ルールが設定されているかの透明性を全ユーザーとサービスに与える設計です	ロール割り当ての参照には権限が要ります

観点	Azure Policy	Azure RBAC
組み合わせたとき	**個人がアクションを実行する権限を持っていても、結果が非準拠なリソースになるなら Azure Policy が作成・更新をブロックします**	—

Azure Policy の操作権限は **Microsoft.Authorization** と **Microsoft.PolicyInsights** の2つのリソース プロバイダーにあります。

ロール	できること
Resource Policy Contributor	ほとんどの Azure Policy 操作を含みます
Owner	すべての権限を持ちます
Contributor	すべての読み取り操作に加え、**リソースの修復をトリガーできますが、定義や割り当ての作成・更新はできません**
Reader	すべての読み取り操作
User Access Administrator	`deployIfNotExists` や `modify` の割り当ての managed identity に必要な権限を付与するために必要です

すべてのポリシー オブジェクト(定義、イニシアティブ、割り当て)は、そのスコープ以下のすべてのロール保有者から読み取れます。

運用の推奨(シナリオ問題の評価軸になります)

- **強制(** `deny` **、** `modify` **、** `deployIfNotExists` **)ではなく** `audit` **または** `auditIfNotExists` **から始める。**既存の自動スケール スクリプトなどを妨げる恐れがあるためです。

- **定義は管理グループやサブスクリプションといった上位で作り、割り当ては1つ下の子レベルで行う。**
- **単一のポリシー定義から始める場合でも、イニシアティブ定義を作って割り当てる。**後からポリシー定義を追加しても、管理する割り当ての数が増えません。**イニシアティブの割り当てを作成した後にイニシアティブに追加したポリシー定義は、その割り当ての一部になります。**個別に評価したいポリシーは、イニシアティブに含めない方が適切です。
- **ポリシー定義、イニシアティブ、割り当ての変更に人のレビューを挟み、Azure Policy のリソースをコードとして管理する。**

オブジェクトの上限(数字が出ます)

どこ	何	上限
スコープ	ポリシー定義	**500**
スコープ	イニシアティブ定義	**200**
テナント	イニシアティブ定義	**2,500**
スコープ	ポリシーまたはイニシアティブの割り当て	**200**
スコープ	除外(Exemptions)	**1000**
ポリシー定義	パラメーター	**20**
イニシアティブ定義	ポリシー	**1000**
イニシアティブ定義	パラメーター	**400**
ポリシー/イニシアティブの割り当て	除外スコープ(notScopes)	**400**
ポリシー ルール	入れ子の条件	**512**

どこ	何	上限
修復タスク	リソース	**50,000**
定義・イニシアティブ・割り当ての要求本文	バイト	**1,048,576**

Azure Virtual Network Manager (AVNM) の動的グループは Azure Policy 定義でメンバーシップを評価します。AVNM の動的グループ ポリシーを作成、編集、削除するには、対象ポリシーへの読み書きの Azure RBAC 権限と、ネットワーク グループへの参加権限 (`Microsoft.Network/networkManagers/networkGroups/join/action`) が必要です。**クラシック管理者の認可はサポートされません。**Co-Administrator のサブスクリプション ロールだけを割り当てられたアカウントには、AVNM の動的グループに対する権限がありません。

1-3-2 Microsoft Defender for Cloud による規制コンプライアンスの評価(Evaluate regulatory compliance) [1.3]

Defender for Cloud において、**規制コンプライアンス標準は Azure Policy のイニシアティブとして実装されています。**Defender for Cloud はこれらの標準を **Regulatory compliance ダッシュボード**で評価します。

規制コンプライアンス標準は、**Azure サブスクリプション、Amazon Web Services (AWS) アカウント、Google Cloud Platform (GCP) プロジェクト**といったスコープに割り当てられます。Defender for Cloud は選択したスコープを各標準に照らして継続的に評価し、リソースが準拠か非準拠かを示して修復の推奨を提供します。

前提条件

- **Defender for Cloud のいずれかのプランをオンボードする必要があります。ただし Defender for Servers Plan 1 と Defender for API Plan 1 は除きます。**
- 標準を追加するには **Owner または Policy Contributor** 権限が必要です。

> **よく出る取り違え**:「無償の基礎的な CSPM だけで PCI DSS の規制コンプライアンス標準を割り当てられる」は誤りです。**いずれかの Defender for Cloud プランのオンボードが必要**で、しかも Defender for Servers Plan 1 と Defender for API Plan 1 では要件を満たしません。

割り当ての手順(並べ替え問題の根拠)

順	操作
1	Azure ポータルにサインインする
2	**Microsoft Defender for Cloud > Regulatory compliance** に移動する。標準ごとに適用されているサブスクリプションを確認できます
3	**Manage compliance policies** を選ぶ
4	規制コンプライアンス標準を割り当てるアカウントまたは管理アカウント(Azure サブスクリプションまたは管理グループ、AWS アカウントまたは管理アカウント、GCP プロジェクトまたは組織)を選ぶ
5	**Security policies** を選ぶ

順	操作
6	有効化したい標準を探し、状態を **On** に切り替える。標準の有効化に情報が必要な場合は **Set parameters** ページが現れます

推奨は、標準に該当する最上位のスコープを選ぶことです。そうすればコンプライアンス データが集約され、入れ子になったすべてのリソースについて追跡できます。

> **重要な挙動: 規制標準を割り当てても、評価対象のリソースが1つもない場合、その標準は Regulatory compliance ダッシュボードに表示されません。**「標準を有効にしたのにダッシュボードに出ない」というトラブルシューティングの設問で、この一文が答えになります。

1-3-3 Defender for Cloud における security controls の実装と構成 — security standards and recommendations [1.3]

Defender for Cloud の**セキュリティ ポリシー**は、Azure、AWS、GCP を横断してクラウド リソースをどう評価するかを定義します。ポリシーは、リソース構成を評価しセキュリティ リスクを特定するために Defender for Cloud が使う**標準、制御、条件**を指定します。

各ポリシーは **security standards** を含み、標準が環境に適用される制御と評価ロジックを定義します。Defender for Cloud はリソースを継続的に評価し、**リソースが定義された制御を満たさない場合に security recommendation を生成**します。推奨事項は問題と、修復に必要なアクションを記述します。

3種類の security standards

種類	内容
Security benchmarks(セキュリティ ベンチマーク)	**Microsoft Cloud Security Benchmark (MCSB)** をはじめとする組み込みのベースラインと、クラウド プロバイダーのベンチマーク。基礎的なベスト プラクティスを定義します
Regulatory compliance standards(規制コンプライアンス標準)	業界およびコンプライアンス プログラムのフレームワーク。**Defender for Cloud のプランを有効にすると利用できます**
Custom standards(カスタム標準)	組織が定義する標準。組み込みまたはカスタムの推奨事項を含み、Defender for Cloud の評価を社内のセキュリティ ポリシーに合わせます

security recommendations の中身

各推奨事項には次が含まれます。

- 問題の短い説明
- 修復の手順
- 影響を受けるリソース
- 重大度とリスク要因
- 攻撃パスのコンテキスト(利用可能な場合)

Defender for Cloud のリスク モデルは、**露出、データの機微度、横方向の移動の可能性、悪用可能性**に基づいて推奨事項に優先順位を付けます。

> **重要な注意: リスクによる優先順位付け(risk prioritization)はセキュア スコアに影響しません。**「リスクの高い推奨事項を先に直せばセキュア

スコアが大きく上がる」という選択肢は誤りです。

custom recommendations

Kusto Query Language (KQL) を使って独自の評価ロジックを定義する**カスタム推奨事項**を作成できます。この機能は **Defender CSPM プランが有効な場合に、すべてのクラウドで利用できます**。

カスタム推奨事項は**カスタム標準の中で作成**され、必要に応じて他の標準にもリンクできます。各推奨事項にはクエリ ロジック、修復手順、重大度、適用対象のリソース種別が含まれます。作成後は**組み込みの推奨事項と並んで Regulatory compliance ダッシュボードに現れ、全体的なセキュリティ態勢の評価に寄与します**。

動作の具体例

MCSB には、期待されるセキュリティ構成を定義する複数の制御が含まれます。そのひとつが「Storage accounts should restrict network access using virtual network rules(ストレージ アカウントは仮想ネットワーク規則でネットワーク アクセスを制限すべき)」です。Defender for Cloud はリソースを継続的に評価し、この制御を満たさないものを見つけると**非準拠としてマークし、推奨事項をトリガー**します。この場合のガイダンスは「仮想ネットワーク規則で保護されていない Azure ストレージ アカウントを強化する」になります。

> **よく出る取り違え**: 「Azure Policy と Defender for Cloud の推奨事項は別物」という理解は不正確です。Defender for Cloud の規制コンプライアンス標準は**Azure Policy のイニシアティブとして実装**されており、推奨事項はその評価結果です。「規制標準を割り当てるのに Policy Contributor 権限が要る」のはこのためです。

1-3-4 resource locks の実装 [1.3]

管理者は、Azure サブスクリプション、リソース グループ、またはリソースをロックして、**誤った削除や変更から保護**できます。**ロックはあらゆるユーザーの権限を上書きします。**

ロックの種類(コマンドライン)	ポータルでの名前	何ができて何ができないか
CanNotDelete	**Delete**	認可されたユーザーはリソースを**読み取りと変更はできますが、削除できません**
ReadOnly	**Read-only**	認可されたユーザーはリソースを**読み取れますが、削除も更新もできません。すべての認可済みユーザーを Reader ロールの権限に制限するのと同様**です

Azure RBAC と異なり、管理ロックはすべてのユーザーとロールにまたがって制限を適用します。この一文が RBAC との対比問題の答えです。

継承

親スコープにロックを適用すると、そのスコープ内のすべてのリソースが同じロックを継承します。後から追加したリソースも同じ親のロックを継承します。**継承の連鎖の中で最も制限の厳しいロックが優先されます。**

拡張リソース(extension resources)は、適用先のリソースからロックを継承します。たとえば `Microsoft.Insights/diagnosticSettings` は拡張リソース種別です。ストレージ BLOB に診断設定を適用してストレージ アカウントをロックすると、その診断設定は削除できません。

Delete ロックを持つリソースがある場合、そのリソース グループを削除しようとすると、削除操作全体がブロックされます。リソース グループや他のリソースがロックされていなくても削除は行われません。**部分的な削除はできません。**

Azure サブスクリプションのキャンセルについては、リソース ロックはキャンセルをブロックしません。Azure はリソースをすぐには削除せず**無効化して保存**し、待機期間の後にはじめて恒久的に削除します。

ロックのスコープ(最重要の制約)

ロックは Azure の制御プレーンの操作にのみ適用され、データ プレーンの操作には適用されません。

Azure の制御プレーン操作は `https://management.azure.com` に向かい、データ プレーン操作はサービス インスタンス(たとえば `https://myaccount.blob.core.windows.net/`)に向かいます。この区別が意味するのは、**ロックはリソースを変更から保護しますが、リソースが機能を果たす方法は制限しない**ということです。SQL Database の論理サーバーに ReadOnly ロックをかけると、サーバーの削除と変更は防げますが、**サーバー内のデータベースのデータの作成・更新・削除は許されます**。

管理グループにはロックを追加できません。

図 1-8　resource locks は 2 種類 —— RBAC を上書きして全員に効く

適用前に知っておくべき副作用(誤答肢の宝庫)

ロックは、リソースを変更しないように見える操作もブロックすることがあります。**ロックは POST メソッドが Azure Resource Manager API にデータを送るのを防ぎます。**代表例を挙げます。

- **ストレージ アカウントへの read-only ロックは、アカウント キーの一覧表示を防ぎます。**List Keys 操作は POST 要求で処理されるためです。アカウント キーはストレージ アカウント内のデータへの完全なアクセスを提供します。**read-only ロックが構成されている場合、アカウント キーを持たないユーザーは BLOB やキュー のデータにアクセスするために Microsoft Entra の資格情報を使う必要があります。**
- **ストレージ アカウントへの read-only ロックは、ストレージ アカウントまたはデータ コンテナー(BLOB コンテナーやキュー)にスコープされた Azure RBAC ロールの割り当ても防ぎます。**

- **ストレージ アカウントへの read-only ロックは、BLOB コンテナーの作成を防ぎます。**ただし、ストレージ アカウントへの作成操作は制御プレーンとデータ プレーンの両方から行えます。read-only ロックが止めるのは制御プレーンの作成要求だけで、**データ プレーン経由の作成操作は依然として実行できます。**
- **ストレージ アカウントへの read-only ロックも cannot-delete ロックも、そのデータが削除・変更されることは防ぎません。**BLOB、キュー、テーブル、ファイル内のデータも保護しません。たとえば File Shares - Delete(制御プレーン操作)なら削除は失敗しますが、Delete Share(データ プレーン操作)なら削除は成功します。**制御プレーンの操作を使うことが推奨されます。**
- **ネットワーク セキュリティ グループ (NSG) への read-only ロックは、対応する NSG フロー ログの作成を防ぎます。**NSG への cannot-delete ロックは、NSG フロー ログの作成や変更を防ぎません。
- App Service リソースへの read-only ロックは、Visual Studio Server Explorer がファイルを表示するのを防ぎます(書き込みアクセスが必要なため)。
- **App Service プランを含むリソース グループへの read-only ロックは、プランのスケール アップとスケール アウトを防ぎます。**
- **仮想マシンを含むリソース グループへの read-only ロックは、すべてのユーザーが仮想マシンを起動・再起動するのを防ぎます。**これらの操作は POST 要求だからです。
- **リソース グループへの read-only ロックは、既存のリソースをそのリソース グループに移動したり外に出したりするのを防ぎます。**ただし read-only ロックを持つリソースを別のリソース グループに移動することはできます。

- Automation アカウントを含むリソース グループへの read-only ロックは、すべての Runbook の開始を防ぎます。
- **リソースまたはリソース グループへの cannot-delete ロックは、Azure RBAC 割り当ての削除を防ぎます。**
- **リソース グループへの cannot-delete ロックは、Resource Manager が履歴内のデプロイを自動削除するのを防ぎます。**履歴が **800 デプロイ**に達するとデプロイが失敗します。
- **Azure Backup サービスが作成したリソース グループに cannot-delete ロックがあると、バックアップが失敗します。**サービスは最大 **18 個の復旧ポイント**をサポートしますが、ロックされていると復旧ポイントをクリーンアップできません。
- Azure Machine Learning ワークスペースを含むリソース グループへの cannot-delete ロックは、Azure Machine Learning コンピューティング クラスターの自動スケールを正しく動作させません。未使用ノードを削除できず、必要以上のリソースを消費します。
- **Log Analytics ワークスペースへの read-only ロックは、User and Entity Behavior Analytics (UEBA) の有効化を防ぎます。**
- **Log Analytics ワークスペースへの cannot-delete ロックは、データのパージ操作を防ぎません。**代わりに **Data Purger** ロールをユーザーから外します。
- **サブスクリプションへの read-only ロックは、Azure Advisor が正しく動作するのを防ぎます。**Advisor がクエリの結果を保存できないためです。
- Application Gateway への read-only ロックは、バックエンドの正常性の取得を防ぎます(POST メソッドのため)。
- **Azure Kubernetes Service (AKS) クラスターへの read-only ロックは、ポータルからクラスター リソースへアクセスする方法を制限します。**

AKS クラスターの Kubernetes リソース セクションからクラスター リソースを選ぶことができなくなります(認証に POST 要求が必要なため)。

- **Site Recovery で保護されている仮想マシンへの cannot-delete ロックは、保護の解除やレプリケーションの無効化時に Site Recovery 関連のリソース リンクの削除を妨げます。**後で再度保護する予定があるなら、保護を無効化する前にロックを外します。
- **PostgreSQL では、仮想ネットワークまたはサブネット レベルにリソース ロックを設定してはいけません。**ロックがネットワークと DNS の操作を妨げる可能性があります。仮想ネットワーク内にサーバーを作成する前に、仮想ネットワークとすべてのサブネットから delete または read-only ロックを外し、サーバー作成後に付け直します。

ロックを作成・削除できるのは誰か

管理ロックの作成と削除には、`Microsoft.Authorization/*` または `Microsoft.Authorization/locks/*` **アクションへのアクセス**が必要です。**Owner** と **User Access Administrator** ロールが割り当てられたユーザーは必要なアクセスを持ちます。一部の特殊な組み込みロールもこのアクセスを与えます。必要な権限を持つカスタム ロールを作成することもできます。

マネージド アプリケーションとロック

Azure Databricks のように**マネージド アプリケーション**でサービスを実装する Azure サービスがあります。この場合、サービスは**2つのリソース グループ**を作ります。1つはサービスの概要を含むロックされていないリソース グループ、もう1つは**サービスのインフラストラクチャを含むロックされたリソース グループ**です。

インフラストラクチャのリソース グループを削除しようとするとロックのエラーが出ます。そのロックを削除しようとすると、**システム アプリケーションが所有しているためロックを削除できないというエラー**になります。**正しい対処は、サービス自体を削除することです。**そうするとインフラストラクチャのリソース グループも削除されます。

ロックの構成方法

ポータルの左メニューでは、サブスクリプションのロック機能は **Resource locks**、リソース グループのロック機能は **Locks** という名前です。ロックには名前とロック レベルを付け、必要に応じてメモを追加します。

ARM テンプレートまたは Bicep でロックをデプロイするときは、**デプロイのスコープとロックのスコープの関係**を理解する必要があります。

- **リソース グループやサブスクリプションをロックするなど、デプロイ スコープにロックを適用する場合は、`scope` プロパティを設定しません。**
- **デプロイ スコープ内のリソースをロックする場合は、ロックに `scope` プロパティを設定**し、ロック対象のリソース名を指定します。

リソース種別は `Microsoft.Authorization/locks`、API バージョンは **2016-09-01** を使います。

コマンド例は次のとおりです。

- Azure PowerShell…… `New-AzResourceLock -LockLevel CanNotDelete -LockName LockSite -ResourceName examplesite -ResourceType Microsoft.Web/sites -ResourceGroupName exampleresourcegroup`。取得は `Get-AzResourceLock`、削除は `Remove-AzResourceLock -LockId $lockId`。
- Azure CLI…… `az lock create --name LockSite --lock-type CanNotDelete --resource-group exampleresourcegroup --resource-name examplesite --`

`resource-type Microsoft.Web/sites`。一覧は `az lock list`、削除は `az lock delete --ids $lockid`。

- REST API…… `PUT https://management.azure.com/{scope}/providers/Microsoft.Authorization/locks/{lock-name}?api-version=2016-09-01` に `{"properties": {"level": "CanNotDelete", "notes": "..."}}` を送ります。

1-3-5 Azure built-in role assignments の管理 [1.3]

Azure role-based access control (Azure RBAC) は Azure Resource Manager 上に構築された認可システムで、Azure リソースに対するきめ細かなアクセス管理を提供します。**利用は無料**で、Azure サブスクリプションに含まれます。

ロール割り当ての3要素

role assignment は、**security principal**、**role definition**、**scope** の3つで構成されます。この3語をそろえて答えさせる設問が頻出です。

要素	内容
security principal	Azure リソースへのアクセスを要求する**ユーザー、グループ、サービス プリンシパル、managed identity** を表すオブジェクト。**このすべてにロールを割り当てられます**
role definition	権限の集合。実行できるアクション(read、write、delete など)を列挙します。Owner のような高レベルのものから Virtual Machine Reader のような具体的なものまであります
scope	アクセスが適用されるリソースの集合

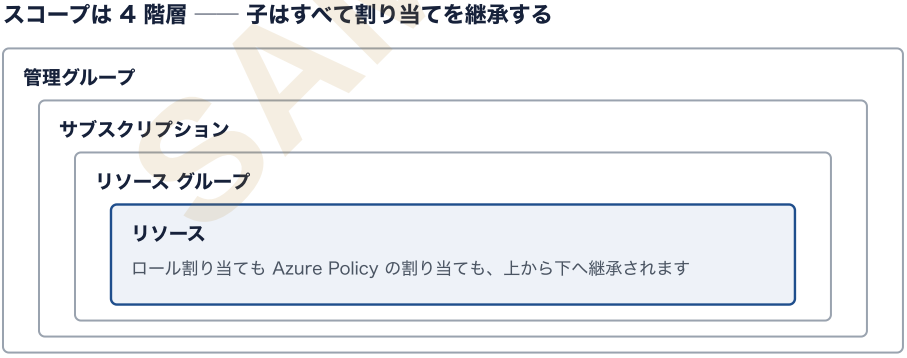
4つのスコープ レベル

Azure では **管理グループ、サブスクリプション、リソース グループ、リソース**の4レベルでスコープを指定できます。スコープは**親子関係**で構成され、どのレベルでもロールを割り当てられます。

ロール割り当てはグループに対して推移的(transitive) です。ユーザーがあるグループのメンバーで、そのグループがロール割り当てを持つ別のグループのメンバーであれば、そのユーザーはロール割り当ての権限を持ちます。

複数のロール割り当ての重なり方

Azure RBAC は加算モデル(additive model) です。実効権限は**ロール割り当ての合計**になります。サブスクリプション スコープで Contributor、リソース グループで Reader を付与された場合、合計は「サブスクリプションに対する Contributor」であり、**Reader の割り当ては実質的に何の影響も与えません。**



Azure RBAC は加算モデルで、実効権限はロール割り当ての合計です。上位で Contributor、
下位で Reader を割り当てても、Reader の割り当ては実質的に何の影響も与えません
(Azure RBAC には減算の概念がありません)。
Azure Policy も明示的な拒否のシステムで、下位に緩い定義を割り当てても緩い方は勝ちません。
下位だけ緩めたいときは、上位の割り当てからそのスコープを除外(notScopes)します。

図 1-9　スコープは 4 階層 —— 子はすべて割り当てを継承する

> **よく出る取り違え**:「上位で Contributor、下位で Reader を割り当てれば、下位では読み取りだけに制限できる」は誤りです。**Azure RBAC には減算の概念がありません。**制限したいなら上位の割り当てを外すか、`deny assignment` や resource locks のような別の仕組みを使います。

アクセス判定のロジック(順序が出ます)

順	何が起きるか
1	ユーザー(またはサービス プリンシパル)が Azure Resource Manager のトークンを取得します。**トークンにはユーザーのグループ メンバーシップ(推移的なものを含む)が含まれます**
2	トークンを付けて Azure Resource Manager に REST API 呼び出しを行います
3	Azure Resource Manager が、そのリソースに適用されるすべての**ロール割り当てと deny 割り当て**を取得します
4	**deny 割り当てが適用される場合、アクセスはブロックされます。**そうでなければ評価が続きます
5	このユーザーまたはグループに適用されるロール割り当てを絞り込み、このリソースに対して持つロールを決定します

順	何が起きるか
6	API 呼び出しのアクションが、そのロールに含まれるかを判定します。ロールの `Actions` にワイルドカード(`*`)が含まれる場合、**実効権限は `Actions` から `NotActions` を引いて計算**します。データアクションも同様です
7	要求されたスコープでそのアクションを持つロールがなければ、アクセスは許可されません。あれば条件の評価に進みます
8	ロール割り当てに条件が含まれる場合は評価されます。含まれなければアクセスは許可されます
9	条件が満たされればアクセスは許可され、満たされなければ許可されません

計算式で覚えます。

- `Actions - NotActions = 実効的な管理権限`
- `DataActions - NotDataActions = 実効的なデータ権限`

Azure には**データ アクション(data actions)** があり、オブジェクト内のデータへのアクセスを付与できます。ストレージ アカウントへの読み取りデータ アクセスを持つユーザーは、その中の BLOB やメッセージを読めます。

割り当ての手段とデータの保管

ロールの割り当ては Azure ポータル、Azure CLI、Azure PowerShell、Azure SDK、REST API から行えます。

ロール定義、ロール割り当て、deny 割り当てはグローバルに保存されます。これはリソースを作成したリージョンに関わらずアクセスできるようにするためです。Azure RBAC は Azure Resource Manager によって強制され、Resource Manager はグローバル エンドポイントを持ち、要求は速度と回復性のために最寄りのリージョンにルーティングされます。したがって Azure RBAC はすべてのリージョンで強制される必要があり、データはすべてのリージョンに複製されます。**ロール割り当てなどの Azure RBAC データが削除されると、データはグローバルに削除されます。**

1-3-6 custom roles の管理 — Azure roles と Microsoft Entra roles [1.3]

この節で最も重要なのは、Azure roles と Microsoft Entra roles が別の体系であることです。

観点	Azure roles(Azure RBAC)	Microsoft Entra roles
何へのアクセスを制御するか	仮想マシンやストレージといった **Azure リソース**。Azure Resource Management 経由	ユーザー、グループ、アプリケーションといった **Microsoft Entra のリソース**。Microsoft Graph API 経由
権限の互換性	—	**Microsoft Entra ロールの権限を Azure カスタム ロールに使うことはできず、その逆もできません**
カスタム ロールのライセンス	追加費用なし	**カスタム ロールの割り当てを持つユーザーごとに Microsoft Entra ID P1 ライセンスが必要**(組み込みロールの利用は無料)

観点	Azure roles(Azure RBAC)	Microsoft Entra roles
割り当ての手段	Azure ポータル、Azure PowerShell、Azure CLI、SDK、REST API	Microsoft Entra 管理センター、Microsoft Graph PowerShell、Microsoft Graph API。**Azure CLI はサポートされません**

Azure のカスタム ロール

組み込みロールで組織の要件を満たせない場合、独自のカスタム ロールを作成します。組み込みロールと同じく、**管理グループ、サブスクリプション、リソース グループ**のスコープで、ユーザー、グループ、サービス プリンシパルに割り当てられます。カスタム ロールは、**同じ Microsoft Entra テナントを信頼するサブスクリプション間で共有**できます。

作成手順は次の4段階です。

順	操作
1	必要な権限を決める。**通常は既存の組み込みロールから始めて、必要に合わせて修正します**
2	作成方法を決める(Azure portal、Azure PowerShell、Azure CLI、REST API)
3	カスタム ロールを作成する
4	**期待どおりに動作するかテストする**

必要な権限を特定する方法は、既存の組み込みロールを見る、アクセスを付与したい Azure サービスを列挙する、**そのサービスにマッピングされるリソース プロバイダーを特定する**、利用可能な権限を検索する、の4つです。Azure サービスは**リソース プロバイダー**を通じて機能と権限を公開します。たとえ

ば Microsoft.Compute は仮想マシンのリソースを、Microsoft.Billing はサブスクリプションと課金のリソースを提供します。ポータルでカスタム ロールを作るときは、キーワードで権限を検索したり、すべての権限を CSV でダウンロードして検索したりできます。

ロール定義のプロパティは次のとおりです。

プロパティ	必須	内容
`Name` / `roleName`	必須	カスタム ロールの表示名。**Microsoft Entra テナントのスコープで一意である必要があります。**推奨最大 256 文字、最大 512 文字
`Id` / `name`	必須	一意の ID。Azure PowerShell と Azure CLI では作成時に自動生成されます
`IsCustom` / `roleType`	必須	カスタム ロールなら `true` または `CustomRole`、組み込みなら `false` または `BuiltInRole`
`Description` / `description`	必須	説明。推奨最大 512 文字、最大 2,048 文字
`Actions` / `actions`	必須	ロールが許可する**制御プレーン**のアクションの配列
`NotActions` / `notActions`	任意	許可された `Actions` から除外する制御プレーンのアクション

プロパティ	必須	内容
`DataActions` / `dataActions`	任意	オブジェクト内のデータに対して許可する**データ プレーン**のアクション。**`DataActions` を持つカスタム ロールは管理グループ スコープに割り当てられません**
`NotDataActions` / `notDataActions`	任意	許可された `DataActions` から除外するデータ プレーンのアクション
`AssignableScopes` / `assignableScopes`	必須	このカスタム ロールを割り当て可能なスコープの配列。**最大 2,000**

ワイルドカードは `Actions`、`NotActions`、`DataActions`、`NotDataActions` で使えます。たとえば `Microsoft.CostManagement/exports/*` は、その配下の action / read / write / delete / run/action をまとめて含み、**将来追加される権限も含みます。そのため、ワイルドカードではなく `Actions` と `DataActions` を明示的に指定することが推奨されます。**将来のアクションによって意図しないアクセスが与えられる可能性があるからです。

AssignableScopes は「誰がこのカスタム ロールを作成・削除・更新・参照できるか」も制御します。

作業	必要なアクション	誰ができるか
カスタム ロールの作成・削除	`Microsoft.Authorization/roleDefinitions/write`	**カスタム ロールのすべての `AssignableScopes` でこのアクションを許可されたユーザー**。たとえば管理グループ、サブスクリプション、リソース グループの **Owner** と **User Access Administrator**
カスタム ロールの更新	`Microsoft.Authorization/roleDefinitions/write`	同上
カスタム ロールの参照	`Microsoft.Authorization/roleDefinitions/read`	そのスコープでこのアクションを許可されたユーザー。**すべての組み込みロールがカスタム ロールを割り当て可能として参照できます**

カスタム ロールの上限と制約は数字で問われます。

- **テナントごとに最大 5,000 のカスタム ロール**(Microsoft Azure operated by 21Vianet は **2,000**)
- **`AssignableScopes` にルート スコープ(`"/"`)を設定できません**
- **`AssignableScopes` にワイルドカード(`*`)を使えません。**これはロール定義の更新によってユーザーがスコープへのアクセスを得るのを防ぐためです
- **1つのアクション文字列に置けるワイルドカードは1つだけ**
- **カスタム ロールの `AssignableScopes` に定義できる管理グループは1つだけ**

- Azure Resource Manager は、ロール定義の `AssignableScopes` に書かれた管理グループの実在を検証しません
- **`DataActions` を持つカスタム ロールは管理グループ スコープに割り当てられません。**ただし `DataActions` と1つの管理グループを `AssignableScopes` に持つカスタム ロールを作ることはでき、管理グループ スコープ自体には割り当てられないものの、**その管理グループ配下のサブスクリプションのスコープには割り当てられます。**複数のサブスクリプションで割り当てが必要な `DataActions` 付きのカスタム ロールを、サブスクリプションごとに作らずに済みます

カスタム ロールを削除する前に、そのカスタム ロールを使うロール割り当てをすべて削除する必要があります。割り当てが残ったまま削除しようとすると `There are existing role assignments referencing role (code: RoleDefinitionHasAssignments)` というメッセージが出ます。手順は、カスタム ロール定義を一覧する、`AssignableScopes` から管理グループ・サブスクリプション・リソース グループを取得する、それぞれでロール割り当てを一覧する、該当するロール割り当てを削除する、**Microsoft Entra Privileged Identity Management を使っているなら eligible なカスタム ロールの割り当ても削除する**、最後にカスタム ロールを削除する、という流れです。

ロールがリネームされてもロール ID は変わりません。スクリプトや自動化でロール割り当てを作る場合は、ロール名ではなく一意のロール ID を使うのがベスト プラクティスです。

入出力形式は道具によって少し異なります。Azure PowerShell と Azure CLI での作成入力は `Name`、`Description`、`Actions`、`NotActions`、`DataActions`、`NotDataActions`、`AssignableScopes` を持つ JSON です。**PowerShell で更新する場合は `Id` プロパティが追加されます。**REST API では `properties` の下

に `roleName`、`description`、`assignableScopes`、`permissions` を置く形式で、これはポータルで作成したときに生成される形式と同じです。

権限文字列は大文字小文字を区別しません。ただし慣例として、Azure リソース プロバイダーの操作で表示される表記に合わせます。

Microsoft Entra のカスタム ロール

Microsoft Entra ID は**組み込みロール**と**カスタム ロール**の2種類のロール定義をサポートします。組み込みロールは固定された権限セットを持ち、**ロール定義を変更できません。**

カスタム Microsoft Entra ロールで権限を付与するのは**2段階**です。まず**カスタム ロール定義**を作成し、次に**ロール割り当て**で割り当てます。カスタム ロール定義は、**あらかじめ用意された一覧から追加する権限の集合**です。**これらの権限は組み込みロールで使われているものと同じ権限**です。

ロール割り当ては、ロール定義をセキュリティ プリンシパルに特定のスコープで結び付ける Microsoft Entra のリソースです。構成する3要素は Azure RBAC と同じく security principal、role definition、scope です。security principal はユーザー、グループ、サービス プリンシパルで、**グループの場合はロール割り当て可能なグループ(role-assignable group)として設定されている必要があります。**

スコープの種類は次の3つです。

スコープの種類	例
Tenant(テナント、組織全体)	組織のすべてのリソースに対して権限を持ちます
Administrative unit(管理単位)	その管理単位に含まれるオブジェクトに対して権限を持ちます

スコープの種類	例
Microsoft Entra resource	**Microsoft Entra グループ**、**enterprise applications**、**application registrations** のいずれか

重要な違いがあります。**テナントや管理単位のようなコンテナー スコープでロールを割り当てると、その中に含まれるオブジェクトに対する権限が付与されますが、コンテナー自体には付与されません。**逆に、**リソース スコープでロールを割り当てると、そのリソース自体への権限が付与されますが、その先 (たとえば Microsoft Entra グループのメンバー)には及びません。**

同じロールを、あるユーザーには組織のすべてのアプリケーションに対して、別のユーザーには Contoso Expense Reports アプリだけのスコープで割り当てる、という使い方ができます。

アクセス判定のロジックは次のとおりです。ユーザー(またはサービス プリンシパル)が Microsoft Graph エンドポイントへのトークンを取得し、そのトークンで Microsoft Graph 経由の API 呼び出しを行います。Microsoft Entra ID は、アクセス トークンの `wids` **クレーム**に基づいてユーザーのロール メンバーシップを評価するか、直接またはグループ メンバーシップ経由で適用されるすべてのロール割り当てを取得します。API 呼び出しのアクションがそのリソースに対して持つロールに含まれるかを判定し、含まれなければアクセスは許可されません。

ロール割り当ての3つの方法も出題されます。

方法	ライセンス	内容
ユーザーに直接割り当てる	既定の方法	組み込みロールもカスタム ロールも割り当てられます

方法	ライセンス	内容
role-assignable group に割り当てる	**Microsoft Entra ID P1**	ロール割り当て可能なグループを作り、そこにロールを割り当てます。**個人ではなくグループに割り当てることで、ユーザーの追加・削除が容易になり、メンバー全員に一貫した権限を与えられます**
Microsoft Entra PIM で JIT アクセスを与える	**Microsoft Entra ID P2**	ロールへの恒久的なアクセスではなく、**時間制限付きのアクセス**を必要とするユーザーに与えます。詳細なレポートと監査の機能も提供します

1-3-7 Azure role-based access control (RBAC) による overprivileged access assignments の評価と是正(Evaluate and remediate) [1.3]

overprivileged access assignments(過剰な権限の割り当て) の評価と是正は、「誰が何にアクセスできるか」を可視化し、不要な常時権限を減らす作業です。**単一の道具ではなく、複数の道具を組み合わせる**のがこの節の要点です。

「誰が何にアクセスできるか」を見るための道具

道具	何が分かるか	ライセンス上の注意
Role assignments の一覧	テナント、アプリケーション、管理単位のスコープで誰が Microsoft Entra ロールを持っているか。**CSV としてダウンロード**して分析でき、**List unifiedRoleAssignments** の Microsoft Graph API で機械的にも取得できます	—
App role assignments と consent grants	どのユーザーとグループが特定の enterprise application にアクセスできるか。ユーザーまたは管理者が同意した委任された権限とアプリケーション権限	—
Custom security attributes	ユーザーとサービス プリンシパルに、テナント独自の業務属性でタグ付けし、属性でディレクトリを絞り込み・照会できます	—
Access reviews	ユーザーが現在のグループ メンバーシップと enterprise application への割り当てを今も必要としているかを定期的に検証します	**Microsoft Entra ID Governance または Microsoft Entra Suite が必要**(一部の機能は Microsoft Entra ID P2 で動作)

道具	何が分かるか	ライセンス上の注意
PIM access reviews	Microsoft Entra ロールまたは Azure リソース ロールに割り当てられたユーザーとサービス プリンシパルをレビューし、レビュー担当者が各ユーザーの継続的なアクセスを承認または拒否します	**サービス プリンシパルのレビューには Microsoft Entra Workload ID Premium が追加で必要**
Entitlement management	アクセス パッケージを通じてどのユーザーにアクセスが付与されたか。アクセス パッケージは**カタログ**にまとめられ、アクセスの委任とガバナンスに使えます	**Microsoft Entra ID Governance または Microsoft Entra Suite が必要**(一部は P2 で動作)
Sign-in logs	誰が実際にリソースにアクセスしているか、どんな条件下か	—
Audit logs	ロール割り当て、グループ メンバーシップ、その他のディレクトリ オブジェクトへの変更を時系列で追跡します	—

監査ログは構成の変更を、サインイン ログはサインインのイベントを記録します。この2つを合わせることで「付与されたアクセス」と「実際に行使されたアクセス」を区別できます。Microsoft Graph API 呼び出しをより詳細に追跡するには Microsoft Graph アクティビティ ログを使います。**大規模なテナントでは、ログを Log Analytics ワークスペースにストリーミング**して、数千のユーザーとロールを横断してアクセス パターンをクエリ・分析することが推奨されます。

PIM の Discovery and insights による是正

Discovery and insights(旧 Security Wizard)は、**組織で誰が特権ロールに割り当てられているかを示し、恒久的なロール割り当てを just-in-time の割り当てに素早く変換する**ためのページです。**分析ツールであり、同時にアクション ツールでもあります。**

PIM を使い始める前は、**すべてのロール割り当てが恒久的**です。ユーザーは特権を必要としないときでも常にロールの中にいます。Discovery and insights は特権ロールの一覧と、そのロールに現在何人のユーザーがいるかを示し、ロールごとに割り当てを展開して、見覚えのないユーザーの詳細を確認できます。

操作手順は次のとおりです。

順	操作
1	Microsoft Entra 管理センターに、少なくとも **Privileged Role Administrator** としてサインインする
2	**ID Governance > Privileged Identity Management > Microsoft Entra roles > Discovery and insights** に移動する。**ページを開くと該当するロール割り当てを探す検出処理が始まります**
3	**Reduce Global Administrators** を選ぶ
4	Global Administrator のロール割り当ての一覧をレビューする
5	**Next** を選び、eligible にしたいユーザーまたはグループを選択して、**Make eligible** または **Remove assignment** を選ぶ

順	操作
6	必要に応じて、すべての Global Administrator に自身のアクセスをレビューさせる(access reviews)
7	**Eliminate standing access** または **Review service principals** を選び、他の特権ロールとサービス プリンシパルのロール割り当てに対して同じ手順を繰り返す。**サービス プリンシパルのロール割り当てについては、削除しかできません**

> **重要な例外(必ず恒久のまま残すもの)**: Microsoft は、**2つのクラウド専用の緊急アクセス アカウントに Global Administrator ロールを恒久的に割り当てておく**ことを推奨しています。これらは特権が非常に高く、特定の個人に割り当てられません。通常のアカウントが使えない、あるいはすべての管理者が誤ってロックアウトされたといった緊急事態(break-glass)に限定して使います。
>
> もう1つの例外は、**ユーザーが Microsoft アカウント(Skype や Outlook.com へのサインインに使うアカウント)を持っている場合**です。Microsoft アカウントを持つユーザーにロール割り当てのアクティブ化時の multifactor authentication を要求すると、**そのユーザーはロックアウトされます**。この場合もロール割り当ては恒久のままにします。

ワークロード ID の過剰権限も見る

過剰権限の是正は、ユーザーだけでなく**アプリケーション、サービス プリンシパル、managed identities といったワークロード ID** も対象にする必要があります。1-1-4 で挙げた統制がそのまま当てはまります。

- **Conditional Access for workload identities** で、サービス プリンシパルがいつ・どこから認証できるかを制限します。Workload Identities Premium ライセンスが必要で、**自テナントに登録された単一テナントのサービス プリンシパルのみが対象**です。
- **グループとアプリケーションの access reviews** で、割り当てられたユーザーが今もアクセスを必要としているかを確認します。
- **PIM access reviews** で、Microsoft Entra ロールと Azure リソース ロールに割り当てられたサービス プリンシパルをレビューします。
- サービス プリンシパルに **custom security attributes** でタグ付けし、絞り込み可能なインベントリを作って **Azure ABAC** の判断を業務属性から駆動します。
- マルチテナント アプリでは **app instance property lock** を有効にし、別テナントにプロビジョニングされた後にサービス プリンシパルの機微なプロパティが不正に変更されるのを防ぎます。

Azure RBAC 側での是正の考え方

Azure RBAC は加算モデルなので、**過剰な権限を「下位で弱いロールを重ねて打ち消す」ことはできません**。是正の手段は次の3つに限られます。

手段	効き方	使いどころ
ロール割り当ての削除・スコープの縮小	アクセスは割り当ての削除で取り消されます	恒久的に不要な権限
PIM で eligible 化	権限は残すが常時保有をやめます	業務上たまに必要な権限
resource locks	すべてのユーザーとロールに一律で削除・変更を禁止します	権限の設計とは別に、事故と悪用を物理的に止めたい重要リソース

> **誤答肢の切り方**:「サブスクリプションの Owner が多すぎる。最小権限にしたい」に対して、「Owner を残したままリソース グループに Reader を割り当てる」は誤りです。加算モデルなので Owner のままです。正解は **PIM の Discovery and insights で恒久の割り当てを eligible に変換する**、あるいは**割り当てそのものを削除して必要最小のロールに置き換える**構成です。

1-3-8 Azure Backup security features による backup protection の構成 [1.3]

バックアップは**最後の砦**です。したがって「バックアップそのものが攻撃対象になる」前提で構成します。Azure Backup は、データの転送中と保存時の両方でバックアップ環境にセキュリティを提供します。

ID とユーザー アクセスの管理

Recovery Services コンテナーが使うストレージ アカウントは**分離**されており、ユーザーが悪意ある目的でアクセスすることはできません。**アクセスは復元などの Azure Backup の管理操作を通じてのみ許可されます。**

バックアップ管理操作を制御する **Azure RBAC の組み込みロールは3つ**です。**対応付け問題で頻出です。**

ロール	できること
Backup Contributor	バックアップの作成と管理。**ただし Recovery Services コンテナーの削除と、他者へのアクセス付与はできません**
Backup Operator	Contributor ができることのうち、**バックアップの削除とバックアップ ポリシーの管理を除いたすべて**

ロール	できること
Backup Reader	すべてのバックアップ管理操作を**参照**する権限

データの分離

Azure Backup では、コンテナーに格納されたバックアップ データは**Microsoft が管理する Azure サブスクリプションとテナント**に保存されます。**外部のユーザーやゲストは、このバックアップ ストレージやその内容に直接アクセスできません。**これにより、データ ソースが存在する運用環境からバックアップ データが分離されます。

Azure 内では、すべての通信と転送中のデータが **HTTPS と TLS 1.2 以上**で安全に転送され、**Azure のバックボーン ネットワーク**内にとどまります。保存時のバックアップ データは既定で **Microsoft マネージド キー**で暗号化され、より強い制御が必要なら**独自のキーを持ち込め**ます。

この仕組みにより、環境が侵害された場合でも、既存のバックアップが権限のないユーザーによって改ざんされたり削除されたりすることはありません。

Azure VM のバックアップにインターネット接続は不要

Azure VM のバックアップでは、仮想マシンのディスクから Recovery Services コンテナーへデータを移動する必要があります。しかし、**必要な通信とデータ転送はすべて Azure のバックボーン ネットワーク上でのみ行われ、仮想ネットワークへのアクセスを必要としません。**したがって、**保護されたネットワーク内に置かれた Azure VM のバックアップのために、IP や FQDN へのアクセスを許可する必要はありません。**

プライベート エンドポイント

Private Endpoints を使うと、仮想ネットワーク内のサーバーから Recovery Services コンテナーへ安全にバックアップできます。プライベート エンドポイントはコンテナー用に **VNET のアドレス空間から IP を使う**ため、仮想ネットワークをパブリック IP に公開する必要がありません。Azure VM 内で動く SQL と SAP HANA のデータベースのバックアップと復元、および **MARS エージェント**を使うオンプレミス サーバーにも使えます。

暗号化の段階

段階	内容
転送中	Azure ストレージとコンテナー間のデータは **HTTPS で保護**され、Azure のバックボーン ネットワークにとどまります
保存時(既定)	**プラットフォーム マネージド キー**で自動的に暗号化されます。**有効化のために明示的な操作は不要**です
保存時(顧客管理)	**Azure Key Vault に保存したカスタマー マネージド キー**で暗号化できます。Recovery Services コンテナーにバックアップされるすべてのワークロードに適用されます
VM のディスク	**Azure Disk Encryption (ADE)** で暗号化された OS / データ ディスクを持つ Azure VM、および **CMK で暗号化されたディスク**を持つ VM のバックアップと復元をサポートします
MARS(オンプレミス)	**パスフレーズ**でアップロード前に暗号化され、ダウンロード後にのみ復号されます

soft delete と enhanced soft delete

soft delete により、VM のバックアップを削除しても**バックアップ データは追加で 14 日間保持**され、データを失わずにそのバックアップ項目を復旧できます。**この「soft delete 状態」での追加 14 日間の保持には費用がかかりません。**

enhanced soft delete は、誤って、あるいは悪意によって削除されたデータを、削除後でも復旧できるようにする強化版です。**指定した期間だけデータの恒久的な削除を遅らせる**ことで、取り戻す機会を提供します。**soft delete を always-on にして無効化できないようにする**こともできます。

immutable vaults(不変コンテナー)

Immutable vault は、**復旧ポイントの喪失につながり得る操作をブロック**することでバックアップ データを保護します。さらに、**immutable vault の設定をロックして不可逆にする**ことで、悪意ある行為者が不変性を無効化してバックアップを削除するのを防げます。

multi-user authorization (MUA)

Multi-user authorization (MUA) は、Recovery Services コンテナーと Backup コンテナーに対する**重要な操作に追加の保護層**を加えます。MUA のために、Azure Backup は **Azure Resource Guard** という別の Azure リソースを使い、重要な操作が該当する認可のもとでのみ実行されるようにします。

監視と不審なアクティビティのアラート

Azure Backup は、Azure Backup に関連するイベントを表示しアクションを構成するための**組み込みの監視とアラート**機能を提供します。**Backup Reports** は、使用状況の追跡、バックアップと復元の監査、さまざまな粒度での主要な傾向の特定を1か所で行える窓口です。これらを使うことで、**認可さ**

れていない、不審な、または悪意あるアクティビティが発生した時点でアラートを受け取れます。

ハイブリッド バックアップ(MARS)の保護

Azure Backup サービスは **Microsoft Azure Recovery Services (MARS) エージェント**を使い、オンプレミスのコンピューターからファイル、フォルダー、ボリューム、システム状態を Azure にバックアップ・復元します。MARS が提供するセキュリティ機能は3つです。

- **パスフレーズの変更のような重要な操作を行うたびに、追加の認証層が加わります。**これは、有効な Azure 資格情報を持つユーザーだけがそうした操作を行えるようにするための検証です。
- **削除されたバックアップ データは削除日から追加で 14 日間保持されます。**攻撃が起きても一定期間内はデータを復旧でき、データ損失が発生しません。また、破損したデータに備えて**より多くの最小復旧ポイント**が維持されます。
- **パスフレーズ**によりアップロード前に暗号化され、ダウンロード後にのみ復号されます。**パスフレーズの詳細は、作成したユーザーと構成されたエージェントだけが知り、サービスには送信も共有もされません。**中間者攻撃などで不注意に露出したデータも、パスフレーズがなければ使えません。

セキュリティ態勢とセキュリティ レベル(対応付け問題の中心)

Azure Backup は、コンテナーに保存されたバックアップ データを保護するために、**コンテナー レベルのセキュリティ機能**を提供します。セキュリティ レベルは4段階に分類されます。

レベル	名称	満たすべき条件
最高	**Excellent (Maximum)**	**immutability または soft-delete のコンテナー設定が有効で、かつ不可逆 (ロック済み / always-on) であること、加えて multi-user authorization (MUA) がコンテナーで有効であること。**すべてのバックアップ データが誤削除から保護され、ランサムウェア攻撃を防ぎます
良	**Good (Adequate)**	**ロック付きの immutability または soft-delete のいずれかを有効化**していること。既存のバックアップを意図しない削除から守り、データ復旧の可能性を高めます
可	**Fair (Minimum/Average)**	**コンテナーで multi-user authorization (MUA) を有効化**していること。基本的な操作に追加の保護層が加わります
不可	**Poor (Bad/None)**	高度な保護機能も、可逆的な機能のみの状態も整っていない状態。None レベルのセキュリティは主に誤削除からの保護しか提供しません

セキュリティ レベルは **Resiliency** を通じて、すべてのデータ ソースについてそれぞれのコンテナーで確認・管理できます。

> **誤答肢の切り方**:「MUA を有効にすれば Excellent になる」は誤りです。**Excellent には、不可逆な immutability または soft-delete と、MUA の両方**が必要です。MUA だけなら **Fair** です。逆に、ロック付き immutability だけなら **Good** です。

1-3-9 infrastructure as code による security controls の実装と構成 [1.3]

infrastructure as code (IaC) によるセキュリティ制御の実装は、「本番に到達する前に、テンプレートの段階で構成ミスを止める」考え方です。Defender for Cloud の **DevOps security** がこの役割を担います。

DevOps security の全体像

Defender for Cloud の DevOps security は、**Azure DevOps**、**GitHub**、**GitLab** を含む複数パイプライン環境にまたがって、コードからクラウドまでアプリケーションとリソースを保護するための中央コンソールを提供します。DevOps セキュリティの推奨事項は、他のコンテキストのあるクラウド セキュリティの洞察と相関付けて、コード内の修復に優先順位を付けられます。

主要な機能は3つです。

機能	内容
DevOps セキュリティ態勢の統合的な可視化	複数パイプライン・マルチクラウド環境における DevOps インベントリと、本番前のアプリケーション コードのセキュリティ態勢を完全に把握します。**コード、シークレット、オープンソース依存関係の脆弱性スキャンの結果**を確認でき、DevOps 環境のセキュリティ構成も評価できます

機能	内容
開発ライフサイクル全体でクラウド リソース構成を強化	**IaC テンプレートとコンテナー イメージを保護**して、本番環境に到達するクラウドの構成ミスを最小化します
コード内の重要な問題の修復を優先	Defender for Cloud 内のコードからクラウドまでのコンテキストの洞察を適用します。**pull request annotations** で開発者に重要な修正の優先順位を伝え、開発者が使うツールに直接つながるカスタム ワークフローをトリガーして開発者のオーナーシップを割り当てます

DevOps security コンソールで見えるもの

ページの部分	内容
スキャン結果のメトリック	**コード、シークレット、依存関係、infrastructure-as-code** の DevOps セキュリティ スキャン結果の総数を、重大度と結果の種類でグループ化して表示します
態勢管理の推奨事項	DevOps 環境の態勢管理の推奨事項の数、重大度の高い結果、影響を受けるリソースの数を可視化します
Advanced security のカバレッジ	オンボードされたリソースの総数に対して、高度なセキュリティ機能を持つ DevOps リソースの数を、環境ごとに可視化します

DevOps インベントリ テーブルの列は次のとおりです。

- **Name**……Azure DevOps、GitHub、GitLab からオンボードされた DevOps リソース。

- **DevOps environment**……そのリソースの DevOps 環境。複数の環境をオンボードしている場合に環境で並べ替えられます。
- **Advanced security status**…… `On` (有効)、`Off` (無効)、`Partially enabled` (コード スキャンがオフになっているなど一部の機能が無効)、`N/A` (Defender for Cloud が有効化の情報を持たない)。**この情報は現時点で Azure DevOps と GitHub のリポジトリでのみ利用できます。**
- **Pull request annotation status**…… `On` 、`Off` 、`N/A` 。**現時点で Azure DevOps のリポジトリでのみ利用できます。**
- **Findings**……その DevOps リソースで特定されたコード、シークレット、依存関係、infrastructure-as-code の結果の総数。

テーブルは DevOps リソース レベル(Azure DevOps と GitHub はリポジトリ、GitLab はプロジェクト)のフラット ビューでも、組織・プロジェクト・グループの階層を示すグループ ビューでも表示できます。サブスクリプション、リソース種別、結果の種類、重大度で絞り込めます。

IaC テンプレートのスキャン(Microsoft Security DevOps)

Microsoft Security DevOps (MSDO) を設定して、接続した GitHub リポジトリまたは Azure DevOps プロジェクトをスキャンします。**GitHub Action または Azure DevOps 拡張機能**を使い、**MSDO を IaC ソース コードだけに対して実行**することでパイプラインの実行時間を短縮できます。

前提条件は、GitHub なら Microsoft Security DevOps GitHub Action、Azure DevOps なら Microsoft Security DevOps Azure DevOps 拡張機能の設定と、リポジトリに IaC テンプレートがあることです。

GitHub での設定手順は次のとおりです。

順	操作
1	GitHub にサインインし、リポジトリのメインページに移動する
2	ファイル ディレクトリで **.github > workflows > msdevopssec.yml** を選ぶ
3	編集アイコンを選ぶ
4	YAML の **Run analyzers** セクションに `with:` の下で `categories: 'IaC'` を追加する。**値は大文字小文字を区別します**
5	変更をコミットする
6	必要なら IaC テンプレートをリポジトリに追加する
7	リポジトリの **Actions** でワークフローを選び、スキャンの完了を確認する

Azure DevOps での設定手順は、パイプラインを編集し、**MicrosoftSecurityDevOps@1** タスクの `displayName` 行の下に `inputs:` の下で `categories: 'IaC'` を追加して保存し、Pipelines から該当パイプラインの実行を選んで結果を確認する、という流れです。

結果の確認先は2つです。

- **Defender for Cloud > DevOps security……GitHub Advanced Security (GHAS) ライセンスを必要としません。**
- GHAS ライセンスがある場合は、**GitHub の Security > Code scanning alerts** でネイティブに確認できます。

MSDO に含まれる IaC スキャン ツールと対応テンプレート

この対応表が対応付け問題になります。

ツール	対応するテンプレート
Template Analyzer(**PSRule** を内包)	**ARM テンプレートと Bicep テンプレート**
Terrascan	ARM テンプレート、**CloudFormation、Docker、Helm、Kubernetes、Kustomize、Terraform**
Checkov	ARM テンプレート、**CloudFormation、Docker、Helm、Kubernetes、Kustomize、Terraform**

> **よく出る取り違え**:「Bicep テンプレートをスキャンしたいので Checkov を使う」は誤りです。**Bicep に対応するのは Template Analyzer** です。逆に「Terraform をスキャンしたい」なら Terrascan または Checkov です。

IaC とガバナンスのつなぎ方

IaC でセキュリティ制御を実装するというのは、テンプレートをスキャンして終わりではありません。**制御そのものをテンプレートとして書く**ことも含みます。

- **resource locks を ARM テンプレート / Bicep でデプロイする**(1-3-4 の手順)。デプロイ スコープにロックを適用する場合は `scope` プロパティを設定せず、デプロイ スコープ内のリソースをロックする場合は `scope` プロパティを設定します。
- **Azure Policy のリソース(定義、イニシアティブ、割り当て)をコードとして管理し、変更に人のレビューを挟む。**

- **soft-delete の適用を Azure Policy で強制する**ことで、テンプレートから作られる Key Vault が保護のない状態にならないようにします。

1-3-10 到達目標 1.3 の分かれ目の早見表 [1.3]

要件の言葉	正解になる構成	よく置かれる誤答肢
許可されていないリージョンへのデプロイを止めたい	**Azure Policy** の `deny` 効果(Allowed Locations)	Azure RBAC(誰が操作するかの話であって、構成の話ではありません)
すでにある非準拠リソースを直したい	`deployIfNotExists` または `modify` + **修復タスク**。managed identity に権限を与えるには **User Access Administrator**	`deny`(既存リソースは変更されません)
まず影響を測ってから強制したい	`audit` / `auditIfNotExists` **から始める**	いきなり `deny`(既存の自動化を壊す恐れがあります)
上位で禁止し、下位だけ緩めたい	**上位の割り当てから下位を除外(notScopes)し、下位に緩い定義を割り当てる**	下位により緩いポリシーを割り当てる(明示的な拒否のシステムなので効きません)
リソースの削除を全員に対して止めたい	**CanNotDelete の resource lock**	Azure RBAC で削除権限を外す(別のロールを持つユーザーには効きません)
ストレージ アカウントのデータの削除を止めたい	**ロックでは止まりません。**データ プレーンの保護(不変ストレージ、バックアップなど)が必要です	ReadOnly ロック(制御プレーンのみに効きます)

要件の言葉	正解になる構成	よく置かれる誤答肢
PCI DSS への適合を測りたい	Defender for Cloud の **Regulatory compliance** で標準を割り当てる(**Owner または Policy Contributor**、Defender for Servers Plan 1 と Defender for API Plan 1 以外のプランが必要)	セキュア スコアを見る(規制フレームワークの評価ではありません)
社内独自の基準で評価したい	**Defender CSPM** を有効化し、**KQL によるカスタム推奨事項をカスタム標準の中で作る**	組み込みの規制標準を流用する(社内基準に一致しません)
特定リソース種別だけを操作できるロールが欲しい	**Azure のカスタム ロール**(`Actions` を明示、`AssignableScopes` を限定)	組み込みの Contributor + resource lock(権限の設計にはなりません)
特定のアプリ登録だけを管理させたい	**Microsoft Entra のカスタム ロール**を、そのアプリ登録のリソース スコープで割り当てる(**P1 が必要**)	Azure のカスタム ロール(Entra リソースには効きません)
Global Administrator が多すぎる	**PIM の Discovery and insights** で `Reduce Global Administrators` から eligible へ変換、または割り当てを削除	下位スコープで弱いロールを重ねる(加算モデルなので効きません)
誰が実際に権限を使っているかを知りたい	**サインイン ログ**(行使されたアクセス)と**監査ログ**(付与の変更)を組み合わせ、Log Analytics へストリーミングして分析	ロール割り当ての一覧のみ(付与は分かっても行使は分かりません)

要件の言葉	正解になる構成	よく置かれる誤答肢
バックアップをランサムウェアから守り最高レベルにしたい	**不可逆(ロック済み / always-on)の immutability または soft-delete** に加えて **MUA(Resource Guard)**	MUA だけ(Fair レベルにとどまります)
バックアップを社内ネットワークからのみ取りたい	**Private Endpoints**(SQL / SAP HANA on Azure VM、MARS)	NSG でコンテナーの IP を許可する(Azure VM のバックアップはバックボーンを通り、そもそも不要です)
Bicep テンプレートの構成ミスを本番前に止めたい	**MSDO の IaC スキャン**(`categories: 'IaC'`)。Bicep は **Template Analyzer** が担当	Checkov / Terrascan(Bicep には対応しません)
GHAS ライセンスなしでスキャン結果を見たい	**Defender for Cloud > DevOps security**	GitHub の Security > Code scanning alerts(**GHAS ライセンスが必要**です)

1-4 この章の総まとめ — 3つの到達目標をつなぐ線 [1.1/1.2/1.3]

最後に、ドメイン1を貫く3本の線を確認します。**シナリオ問題は、この線のどこかを切り取って出題されます。**

1本目は「常時の権限をなくす線」です。恒久的なロール割り当て(1-3-5)を PIM の eligible 割り当て(1-1-1)に変え、Discovery and insights で残った過剰権限を洗い出し(1-3-7)、アクティブ化のたびに Conditional Access で本人

確認の強度を要求します(1-1-2、1-1-3)。**評価軸が「最小権限」なら、この線をたどります。**

2本目は「資格情報をなくす線」です。アプリの資格情報をコードから Key Vault へ移し(1-2)、さらに Key Vault にすら置かなくて済むよう managed identities に置き換え(1-1-6)、それでも漏れたシークレットを Defender CSPM のスキャンで発見し(1-2-6)、アプリに与えた過剰な同意を OAuth permission grants の見直しで回収します(1-1-5)。**評価軸が「シークレットを保存しない」なら、この線です。**

3本目は「人の善意に頼らない線」です。構成のずれを Azure Policy が拒否・修復し(1-3-1)、基準との差を Defender for Cloud が測り(1-3-2、1-3-3)、削除は resource locks が全員に対して止め(1-3-4)、それでも失われたときは Azure Backup の不変性と MUA が守り(1-3-8)、すべてをテンプレートとして書いて本番前にスキャンします(1-3-9)。**評価軸が「誰が操作しても」「悪意ある管理者でも」なら、この線です。**

この3本の線は第2章以降でも繰り返し現れます。ストレージとデータベースとネットワーク(第2章)、コンピューティングと AI(第3章)、そして脅威の検知と対応(第4章)は、いずれもこの章で作った ID とガバナンスの土台の上に載ります。

サンプルはここまでです

続き(第2章〜巻末)は、SC-500 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。

SC-500 学習用テキスト(無料サンプル)

版

2026年7月版(SC-500 出題範囲(Study guide・Last updated 2026-05-13)準拠)

発行

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

お問い合わせ

info@shikaku-doujou.com