AWS ホワイトペーパー

AWS 障害分離境界

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 障害分離境界: AWS ホワイトペーパー

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

要約と序章 1
要約 1
Well-Architected の実現状況の確認 1
序章 1
AWS インフラストラクチャ 3
アベイラビリティーゾーン 3
リージョン 4
AWS Local Zones 5
AWS Outposts 5
Point of Presence 6
パーティション 7
コントロールプレーンとデータプレーン 7
静的安定性 8
まとめ 9
AWS のサービスカテゴリ 10
ゾーンサービス 10
リージョンサービス 13
グローバルサービス 14
パーティション固有のグローバルサービス 15
エッジネットワークにおけるグローバルサービス 16
グローバルな単一リージョンのオペレーション 17
デフォルトのグローバルエンドポイントを使用するサービス 21
グローバルサービスの概要 23
結論 27
付録 A - パーティションサービスに関するガイダンス 28
AWS IAM 28
AWS Organizations 28
AWS Account Management 29
Route 53 Application Recovery Controller 30
AWS Network Manager 30
Route 53 のプライベート DNS 31
付録 B - エッジネットワークのグローバルサービスに関するガイダンス 32
Route 53 32
Amazon CloudFront 33

Amazon Certificate Manager 33
AWS Web Application Firewall (WAF) および WAF クラシック 33
AWS Global Accelerator 34
Amazon Shield Advanced 34
付録 C - 単一リージョンのサービス 35
寄稿者 36
ドキュメントの改訂 37
AWS 用語集 38
注意 39

AWS 障害分離境界

発行日: 2022 年 11 月 16 日 (ドキュメントの改訂)

要約

Amazon Web Services (AWS) には、アベイラビリティーゾーン (AZ)、リージョン、コントロールプレーン、データプレーンなど、さまざまな分離境界があります。このホワイトペーパーでは、AWS が、これらの境界を使用してゾーンサービス、リージョンサービス、グローバルサービスをどのように構成するかを詳しく説明します。また、さまざまなサービスへの依存関係をどのように考慮し、これらのサービスを使用して構築したワークロードの回復力をどのように向上させるかについての規範的なガイダンスも提供します。

Well-Architected の実現状況の確認

AWS Well-Architected フレームワークは、クラウド内でのシステム構築に伴う意思決定の長所と短所を理解するのに役立ちます。このフレームワークの 6 つの柱により、信頼性、安全性、効率、費用対効果、持続可能性の高いシステムを設計および運用するための、アーキテクチャのベストプラクティスを確認できます。AWS Management Consoleで無料で提供されている AWS Well-Architected Toolを使用すると、柱ごとに一連の質問に答えることで、これらのベストプラクティスに照らしてワークロードを評価できます。

クラウドアーキテクチャに関する専門的なガイダンスやベストプラクティス (リファレンスアーキテクチャのデプロイ、図、ホワイトペーパー) については、AWS アーキテクチャセンターを参照してください。

序章

AWS は、グローバルインフラストラクチャの運用により、お客様が柔軟性、安全性、スケーラビリティ、可用性の高い方法でワークロードをデプロイするために役立つクラウドサービスを提供します。AWS のインフラストラクチャでは、複数の障害分離構成を使用して、お客様の回復力の目標の達成を支援します。これらの障害分離境界により、お客様は、障害の影響を指定した予測可能な範囲に限定するようにワークロードを設計できます。また、これらの境界を使用して AWS のサービスがどのように設計されているかを理解することも重要です。この理解により、ワークロードに合わせて依存関係を意図的に選択できるようになります。

このホワイトペーパーでは、まず、AWS のグローバルインフラストラクチャとその障害分離境界、および AWS がサービスの設計に使用するパターンのいくつかを紹介します。次に、この知識に基づき、AWS が提供するさまざまなサービスカテゴリ (ゾーン、リージョン、グローバル) について説明します。また、これらの分離境界とさまざまなサービスカテゴリを使用して、AWS で実行するワークロードの回復力を向上させるためのアーキテクチャの構築に関するベストプラクティスも示します。特に、グローバルサービスに依存しつつも単一障害点を最小限に抑える方法に関する規範的なガイダンスを提供します。このガイダンスを参考にして、AWS の依存関係や、高可用性 (HA) とディザスタリカバリ (DR) を実現するワークロードをどのように設計するかについて、十分な情報に基づいた選択を行うことができます。

AWS インフラストラクチャ

このセクションでは、AWS グローバルインフラストラクチャとその障害分離境界について概説します。さらに、AWS のサービスの設計方法における重要な違いであるコントロールプレーンとデータプレーンの概念についても概要を示します。この情報は、障害分離境界とサービスのコントロールプレーンおよびデータプレーンが、次のセクションで説明する AWS のサービスカテゴリにどのように適用されるかを理解するための基礎となります。

トピック

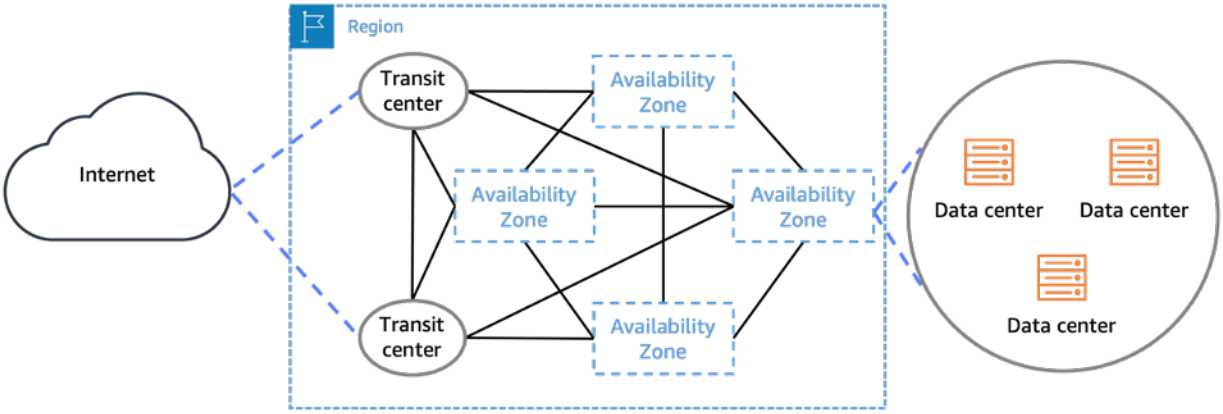
- アベイラビリティーゾーン
- リージョン
- AWS Local Zones
- AWS Outposts
- Point of Presence
- パーティション
- コントロールプレーンとデータプレーン
- 静的安定性
- まとめ

アベイラビリティーゾーン

AWS は世界中の複数のリージョンの 100 以上のアベイラビリティーゾーンを運用しています (最新の状況については、「AWS グローバルインフラストラクチャ」を参照してください)。アベイラビリティーゾーンは、AWS リージョン内で独立した冗長な電源設備、ネットワーク、接続を備えた 1 つ以上の個別のデータセンターです。リージョン内のアベイラビリティーゾーン間は、相関する障害を防ぐために意味のある距離として最大 60 マイル (約 100 km) 離れていますが、1 桁ミリ秒のレイテンシーで同期レプリケーションを利用するには十分近い距離にあります。停電、断水、ファイバー断線、地震、火災、竜巻、洪水などの広域災害のシナリオから同時に影響を受けないように設計されています。発電機や冷却装置などの一般的な障害点は、アベイラビリティーゾーン間で共有されず、独立した配電所から受電するように設計されています。AWS がサービスに更新プログラムをデプロイする場合、相関する障害を防ぐために、同じリージョン内の各アベイラビリティーゾーンへのデプロイは時間的に分離されます。

リージョン内のすべてのアベイラビリティーゾーン間は、完全冗長な専用メトロファイバーを介して、高帯域幅、低レイテンシーのネットワークで相互接続されています。リージョン内の各アベイラビリティーゾーンは、2 つのトランジットセンター (ここで AWS は複数の Tier-1 インターネットプロバイダーとピアリングします) を経由してインターネットに接続します。詳細については、「Amazon Web Services の概要」を参照してください。

これらの特性により、アベイラビリティーゾーンは互いから強く分離されます。これをアベイラビリティーゾーンの独立性 (AZI) と呼びます。アベイラビリティーゾーンの論理構造とインターネットへの接続を次の図に示します。



アベイラビリティーゾーンは 1 つ以上の物理データセンターで構成され、これらは相互に、またインターネットに冗長接続されている

リージョン

各 AWS リージョンは、1 つの地理的エリア内における、複数の独立した、物理的にも分離されたアベイラビリティーゾーンで構成されています。現在、すべてのリージョンに 3 つ以上のアベイラビリティーゾーンがあります。リージョン自体が他のリージョンから分離され、独立しています。ただし、このドキュメントで後述するいくつかの例外を除きます (「グローバルな単一リージョンのオペレーション」を参照してください)。このようなリージョン間の分離により、サービスの障害が発生しても影響は 1 つのリージョンに限定されます。この場合、他のリージョンの通常のオペレーションには影響しません。また、1 つのリージョンで作成したリソースやデータは、AWS のサービスが提供するレプリケーション機能やコピー機能を明示的に使用するか、手動でリソースをレプリケートした場合を除いて、他のどのリージョンにも複製されることはありません。

2022 年 12 月時点の現在の AWS リージョンと計画中の AWS リージョン

AWS Local Zones

AWS Local Zones は、コンピューティング、ストレージ、データベース、その他の一部の AWS のサービスを大都市や産業中心地の近くに配置するインフラストラクチャを活用するデプロイタイプです。Local Zone でコンピューティングやストレージなどの AWS のサービスを使用することで、低レイテンシーのアプリケーションをエッジで実行したり、ハイブリッドクラウド移行を簡素化したりできます。Local Zones は、レイテンシーを短縮するために自身のインターネット接続 (送新/受信) を備えていますが、Amazon の冗長で高帯域幅のプライベートネットワークを介して親リージョンにも接続されているため、AWS Local Zones で実行中のアプリケーションからあらゆるサービスに高速、安全、シームレスにアクセスできます。

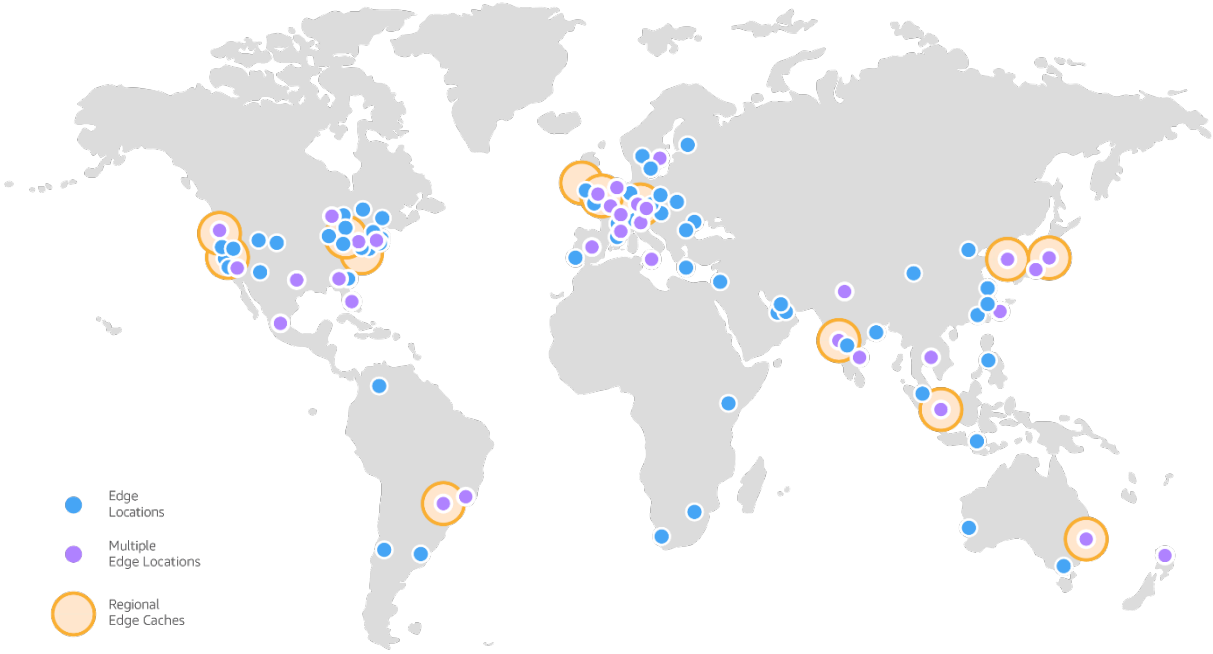
AWS Outposts

AWS Outposts は、AWS のインフラストラクチャとサービスを実質的にすべてのオンプレミスまたはエッジロケーションに提供し、真に一貫したハイブリッドエクスペリエンスを実現するフルマネージドソリューションのファミリーです。Outposts ソリューションを使用すると、AWS のネイティブサービスをオンプレミスで拡張および実行でき、1U および 2U Outposts サーバーから 42U Outposts ラックや複数のラックデプロイまで、さまざまなフォームファクタで利用できます。

AWS Outposts を使用すると、AWS の限定されたサービスをローカルで実行し、親の AWS リージョンで利用できる幅広いサービスに接続できます。AWS Outposts は、AWS が設計したハードウェアで構築された、フルマネージドで設定可能なコンピューティングおよびストレージラックであり、お客様はクラウド内で AWS の幅広いサービスにシームレスに接続しながら、オンプレミスでコンピューティングとストレージを実行できます。

Point of Presence

AWS リージョン およびアベイラビリティーゾーンに加えて、AWS は、グローバルに分散された Point of Presence (POP) ネットワークも運用しています。これらの PoP は、コンテンツ配信ネットワーク (CDN) である Amazon CloudFront、パブリックドメインネームシステム (DNS) 解決サービスである Amazon Route 53、およびエッジネットワーク最適化サービスである AWS Global Accelerator (AGA) をホストしています。グローバルエッジネットワークは現在、400 を超えるエッジロケーションを含む 410 を超える PoP と、48 か国の 90 を超える都市にある 13 のリージョンレベルの中間層キャッシュで構成されています (最新の状況については、「Amazon CloudFront の主な特徴」を参照してください)。



Amazon CloudFront のグローバルエッジネットワーク

各 PoP は、他の PoP から分離されているため、1 つの PoP や大都市圏に影響する障害が発生しても、残りのグローバルネットワークには影響しません。AWS ネットワークは、世界中の何千もの Tier 1/2/3 の通信事業者とピアリングしており、すべての主要なアクセスネットワークとの良好な接

続を通じて最適なパフォーマンスを実現し、数百テラビットの容量をデプロイしています。エッジロケーションは、AWS ネットワークバックボーンを介して AWS リージョン に接続されています。これは、地球を一周するほどの完全冗長の複数の 100 GbE 並列ファイバーであり、数万のネットワークとリンクして、オリジンの取得や動的コンテンツのアクセラレーションを向上させます。

パーティション

AWS は、リージョンをパーティションにグループ化します。各リージョンは 1 つのパーティションに厳密に属し、各パーティションには 1 つ以上のリージョンがあります。パーティションには AWS Identity and Access Management (IAM) の独立したインスタンスがあり、異なるパーティション内のリージョン間に厳密な境界を提供します。AWS 商用リージョンは aws パーティション内にあり、中国のリージョンは aws-cn パーティション内にあり、AWS GovCloud リージョンは aws-us-gov パーティション内にあります。AWS の一部のサービスは、Amazon S3 クロスリージョンレプリケーションや AWS Transit Gateway インターリージョンピアリングなどのクロスリージョン機能を提供するように設計されています。この種の機能は、同じパーティション内のリージョン間でのみサポートされます。あるパーティションの IAM 認証情報を使用して別のパーティションのリソースとやり取りすることはできません。

コントロールプレーンとデータプレーン

AWS では、概念として、ほとんどのサービスをコントロールプレーンとデータプレーンに分けています。これらの用語は、ネットワークの世界、特にルーターから来ています。ルーターの主な機能であるデータプレーンは、ルールに基づいてパケットを移動します。ただし、ルーティングポリシーを作成および配布する必要があり、その作成と配信を担うのがコントロールプレーンです。

コントロールプレーンは、リソースを作成、読み取り/表示、更新、削除、およびリスト (CRUDL) するための管理 API を提供します。例えば、新しい Amazon Elastic Compute Cloud (Amazon EC2) インスタンスの起動、Amazon Simple Storage Service (Amazon S3) バケットの作成、Amazon Simple Queue Service (Amazon SQS) キューの表示は、すべてコントロールプレーンのアクションです。EC2 インスタンスを起動する際、コントロールプレーンは、容量のある物理ホストの検索、ネットワークインターフェイスの割り当て、Amazon Elastic Block Store (Amazon EBS) ボリュームの準備、IAM 認証情報の生成、セキュリティグループルールの追加など、複数のタスクを実行する必要があります。コントロールプレーンは、複雑なオーケストレーションおよびアグリゲーションシステムになりがちです。

データプレーンは、サービスの主要な機能を提供するものです。例えば、実行中の EC2 インスタンス自体、EBS ボリュームへの読み取りと書き込み、S3 バケットのオブジェクトの取得と配

置、Route 53 による DNS クエリへの応答とヘルスチェックの実行は、どれもが該当する各サービスのデータプレーンが提供する機能です。

データプレーンは、ワークフロー、ビジネスロジック、データベースの複雑なシステムを通常実装するコントロールプレーンと比較して、可動部分が少なく、意図的に複雑さが軽減されています。これにより、コントロールプレーンと比べて、データプレーンで障害イベントが発生する可能性は統計的に低くなります。データプレーンとコントロールプレーンは、どちらもサービス全体の運用と成功に貢献しますが、AWS ではこれらを別個のコンポーネントとみなします。この分離により、パフォーマンスと可用性の両方の利点が得られます。

静的安定性

AWS のサービスの最も重要なレジリエンス特性の 1 つは、AWS が静的安定性と呼ぶものです。この用語が意味するのは、依存先に障害が発生したり利用できなくなったりしても、追加の変更を加える必要がなく、システムは通常どおり動作し続ける、つまり静的な状態で動作するということです。これを実現する 1 つの方法は、サービス間の循環依存関係をなくして、サービスの正常なリカバリを阻害しないようにすることです。これを行うもう 1 つの方法は、既存の状態を維持することです。コントロールプレーンは統計的にデータプレーンよりも障害が発生する可能性が高いという事実を考慮します。データプレーンは通常、コントロールプレーンから着信するデータに依存しますが、データプレーンは既存の状態を維持し、コントロールプレーンに障害があっても動作し続けます。データプレーンによるリソースへのアクセスは、いったんプロビジョニングされると、コントロールプレーンに依存しないため、コントロールプレーンの障害の影響を受けません。つまり、リソースを作成、変更、または削除する機能が損なわれた場合でも、既存のリソースは引き続き使用できます。これにより、AWS データプレーンはコントロールプレーンの障害に対して静的に安定します。さまざまなパターンを実装して、さまざまなタイプの依存障害に対して静的に安定させることができます。

静的安定性の例は Amazon EC2 に見ることができます。EC2 インスタンスを起動すると、データセンター内の物理サーバーと同じように利用できるようになります。コントロールプレーン API に依存することなく、実行を継続したり、再起動後に実行を再開したりできます。同じ特性が、VPC、Amazon S3 バケットおよびオブジェクト、Amazon EBS ボリュームなどの他の AWS リソースにも当てはまります。

静的安定性は、AWS のサービスの設計方法に深く根付いた概念ですが、お客様が使用できるパターンでもあります。実際、AWS のさまざまな種類のサービスを回復力のある方法で使用するためのベストプラクティスに関するガイダンスの大部分は、本番環境に静的安定性を実装することです。最も信頼性の高い回復および緩和メカニズムは、最も少ない変更で回復を達成するメカニズムです。障害が発生したアベイラビリティーゾーンから復旧する際、EC2 コントロールプレーンに依存して新し

い EC2 インスタンスを起動する代わりに、その余分な容量を事前にプロビジョニングしておくと、静的安定性を実現できます。したがって、リカバリパスでコントロールプレーン (リソースへの変更を実装する API) への依存関係をなくすことで、より回復力のあるワークロードを作成できます。静的安定性、コントロールプレーン、データプレーンの詳細については、Amazon Builders' Library の記事「アベイラビリティーゾーンを使用した静的安定性」を参照してください。

まとめ

AWS は、インフラストラクチャ内のさまざまな障害コンテナを利用して障害を分離します。インフラストラクチャのコアとなる障害コンテナは、パーティション、リージョン、アベイラビリティーゾーン、コントロールプレーン、およびデータプレーンです。次に、AWS のさまざまな種類のサービス、これらの障害コンテナをサービスの設計にどのように利用するか、これらのコンテナを使用して回復力を高めるワークロードをどのように設計するかについて検討します。

AWS のサービスカテゴリ

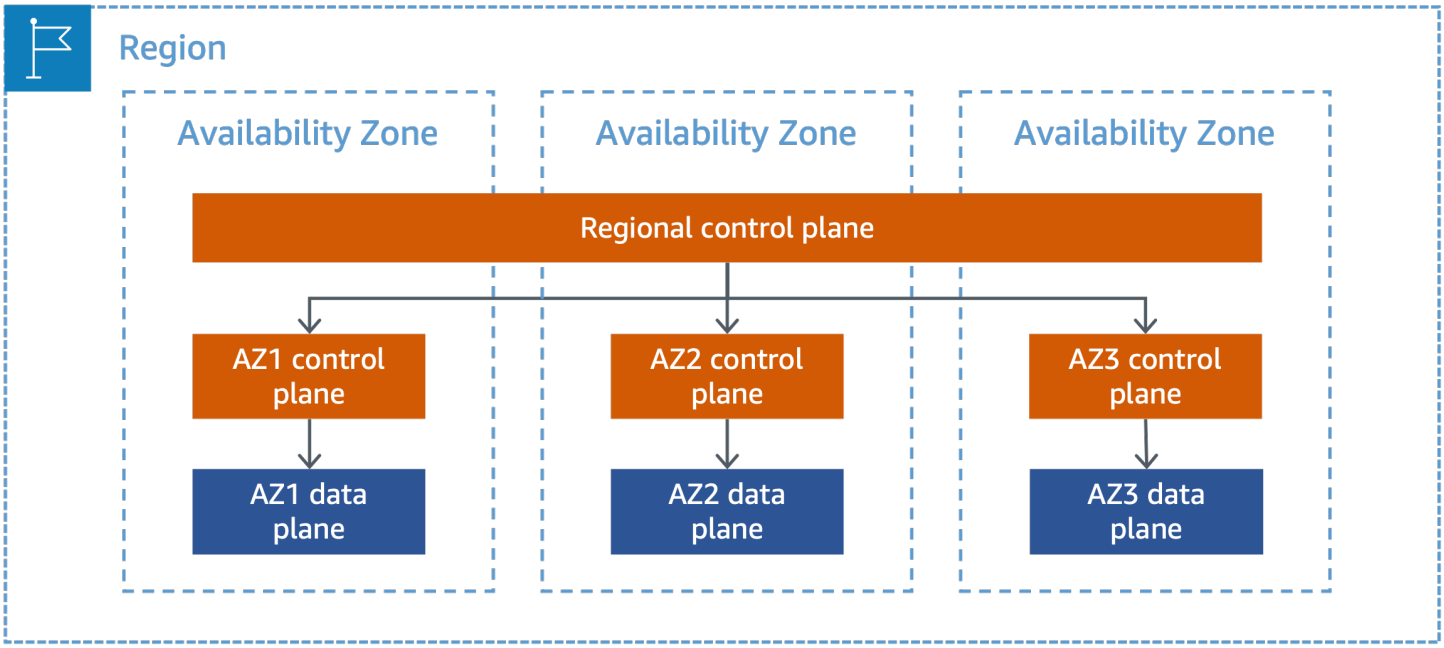
AWS は、障害分離境界に応じて、ゾーン、リージョン、グローバルという 3 つの異なるカテゴリのサービスを運用しています。このセクションでは、これらの異なるカテゴリのサービスがどのように設計されているかを詳しく説明し、特定のサービスカテゴリに属するサービスで障害が発生した場合に、AWS で実行しているワークロードにどのような影響があるかを判断できるようにします。また、ワークロードをどのように構築すれば、これらのサービスを回復力の高い方法で利用できるかについてもガイダンスの概要を示します。グローバルサービスについては、AWS のサービスのコントロールプレーンで発生した障害の影響がワークロードに及ばないようにするための規範的なガイダンスを、このドキュメントの「付録 A - パーティションサービスに関するガイダンス」と「付録 B - エッジネットワークのグローバルサービスに関するガイダンス」に記載しています。これは、単一障害点の発生を最小限に抑えながら、グローバルサービスに安全に依存するために役立ちます。

トピック

- ゾーンサービス
- リージョンサービス
- グローバルサービス

ゾーンサービス

アベイラビリティーゾーンの独立性 (AZI) があることで、AWS は Amazon EC2 や Amazon EBS などのゾーンサービスを提供できます。ゾーンサービスとは、リソースをデプロイする先のアベイラビリティーゾーンを指定できるサービスです。これらのサービスは、リージョン内のアベイラビリティーゾーンごとに独立して動作し、さらに重要な点として、アベイラビリティーゾーンごとに独立して故障します。つまり、あるアベイラビリティーゾーンのサービスのコンポーネントは、他のアベイラビリティーゾーンのコンポーネントに依存しません。これができるのは、ゾーンサービスにゾーンデータプレーンがあるためです。一部のサービス (EC2 など) には、ゾーン指定のオペレーション (EC2 インスタンスの起動など) 用にゾーンコントロールプレーンが含まれている場合があります。これらのサービスの場合、AWS は、サービスとのやり取りを容易にするためにリージョンコントロールプレーンのエンドポイントも提供します。リージョンコントロールプレーンは、リージョンにスコープされた機能を提供するとともに、ゾーンコントロールプレーン上の集約レイヤーおよびルーティングレイヤーとしても機能します。このアーキテクチャを次の図に示します。



ゾーンサービスのコントロールプレーンとデータプレーンはゾーンごとに分離される

複数のアベイラビリティーゾーンを使用すると、お客様は単一のデータセンターを使用した場合よりも、可用性、耐障害性、スケーラビリティが高い本番ワークロードを運用できます。ワークロードで複数のアベイラビリティーゾーンを使用する場合、1 つのアベイラビリティーゾーンの物理インフラストラクチャに影響する問題が発生した場合に、お客様はより適切に隔離して保護できます。これにより、お客様は、複数のアベイラビリティーゾーンにわたって冗長性のあるサービスを構築でき、正しく設計した場合は、1 つのアベイラビリティーゾーンで障害が発生しても運用を続行できます。お客様は AZI を活用して、可用性と耐障害性の高いワークロードを作成できます。アーキテクチャに AZI を実装すると、あるアベイラビリティーゾーンのリソースが他のアベイラビリティーゾーンのリソースとやり取りするのを最小限に抑えるか、完全になくすことができるため、分離されたアベイラビリティーゾーンの障害から迅速に回復できます。これにより、アベイラビリティーゾーン間の依存関係がなくなり、アベイラビリティーゾーンからの退避が簡単になります。アベイラビリティーゾーンからの退避メカニズムの構築の詳細については、「高度なマルチ AZ レジリエンスパターン」を参照してください。また、アベイラビリティーゾーンをさらに活用して、一度に 1 つのアベイラビリティーゾーンにのみ変更をデプロイしたり、アベイラビリティーゾーンに変更が適切に反映できなかった場合にサービスからアベイラビリティーゾーンを削除したりするなど、AWS が独自のサービスで使用しているのと同じベストプラクティスの一部に従うことができます。

静的安定性も、マルチアベイラビリティーゾーンアーキテクチャにとって重要な概念です。マルチアベイラビリティーゾーンアーキテクチャで計画すべき障害モードの 1 つは、アベイラビリティーゾーンの損失です。この損失に伴って、アベイラビリティーゾーン全体に相当する容量が失われる可能性があります。アベイラビリティーゾーンの損失を処理するのに十分な容量を事前にプロビジョ

ニングしていないと、残存容量が現在の負荷に対応できなくなる場合があります。さらに、失われた容量を補うには、使用しているゾーンサービスのコントロールプレーンに依存する必要があり、静的に安定した設計よりも信頼性が低くなる場合があります。この場合、十分な追加容量を事前にプロビジョニングしておくと、動的な変更を必要とせずに通常の運用を継続できるため、1 つのアベイラビリティーゾーンなどの障害ドメインが失われても静的安定性を維持できます。

複数のアベイラビリティーゾーンにデプロイした EC2 インスタンスの Auto Scaling グループを使用して、ワークロードのニーズに応じて動的にスケールイン/スケールアウトできます。自動スケーリングは、数分から数十分にわたって使用量が徐々に変化する場合に適しています。ただし、新しい EC2 インスタンスの起動には時間がかかります。特に、インスタンスがブートストラップを必要とする場合 (エージェント、アプリケーションバイナリ、設定ファイルをインストールするなど) はそうです。この間、残存容量が現在の負荷に対応できなくなる場合があります。さらに、自動スケーリングによる新しいインスタンスのデプロイは EC2 のコントロールプレーンに依存します。これにはトレードオフがあります。1 つのアベイラビリティーゾーンが失われても静的に安定するには、新しいインスタンスのプロビジョニングを自動スケーリングに頼るのではなく、他のアベイラビリティーゾーンに十分な EC2 インスタンスを事前にプロビジョニングして、障害が発生したアベイラビリティーゾーンから負荷を移動して処理する必要があります。追加容量を事前にプロビジョニングすると、追加コストが発生する可能性があります。

例えば、通常の運用は、顧客トラフィックを処理するために 3 つのアベイラビリティーゾーンにわたって 6 つのインスタンスがワークロードに必要であるとします。1 つのアベイラビリティーゾーンの障害に対して静的に安定するには、アベイラビリティーゾーンごとに 3 つずつ、合計 9 つのインスタンスをデプロイします。1 つのアベイラビリティーゾーンのインスタンスに障害が発生しても、残りのインスタンスが 6 つあるため、障害発生時に新しいインスタンスをプロビジョニングして設定しなくても、引き続き顧客トラフィックを処理できます。EC2 容量の静的安定性を実現するために、この例の場合は実行するインスタンス数が 50% 増えるため、追加コストがかかります。S3 バケットやユーザーの事前プロビジョニングなど、リソースの事前プロビジョニングが可能なすべてのサービスに追加コストが発生するわけではありません。ワークロードの望ましい復旧時間を超えてしまう危険性に対して静的安定性を実装することのトレードオフを検討する必要があります。

AWS Local Zones および Outposts は、限定された AWS のサービスのデータプレーンをエンドユーザーに近づけます。これらのサービスのコントロールプレーンは親リージョンにあります。Local Zones や Outposts のインスタンスには、Local Zones や Outposts サブネットを作成したアベイラビリティーゾーンの EC2 や EBS などのゾーンサービスのコントロールプレーンへの依存関係があります。また、Elastic Load Balancing (ELB)、セキュリティグループ、Amazon Elastic Kubernetes Service (Amazon EKS) マネージドの Kubernetes コントロールプレーン (EKS を使用している場合) など、リージョンサービスのリージョンコントロールプレーンへの依存関係もあります。Outposts 固有の追加情報については、関連ドキュメントとサポートおよびメンテナンスに関するよくある質

問を参照してください。Local Zones や Outposts を使用するときは、コントロールプレーンの障害や親リージョンへのネットワーク接続の中断に対する回復力を高めるために、静的安定性を実装してください。

リージョンサービス

リージョンサービスは、AWS が複数のアベイラビリティーゾーンの上に構築したサービスであるため、お客様はゾーンサービスを最大限に活用する方法を考え出す必要はありません。複数のアベイラビリティーゾーンにデプロイしたサービスを論理的にグループ化し、単一のリージョンエンドポイントをお客様に提供します。Amazon SQS や Amazon DynamoDB は、リージョンサービスの例です。リージョンサービスは、アベイラビリティーゾーンの独立性と冗長性を利用し、可用性と耐久性のリスクの 1 種としてのインフラストラクチャの障害を最小限に抑えます。例えば、Amazon S3 はリクエストとデータを複数のアベイラビリティーゾーンに分散し、1 つのアベイラビリティーゾーンの障害から自動的に回復するように設計されています。ただし、お客様はこのサービスのリージョンエンドポイントとのみやり取りします。

ほとんどのお客様は、リージョンサービス (またはゾーンサービスに依存するマルチ AZ アーキテクチャ) を使用することで、1 つのリージョンで回復力の目標を達成できると AWS は考えています。ただし、ワークロードによっては追加の冗長性を必要とする場合があるため、AWS リージョンの分離を使用して HA やビジネス継続性を目的としたマルチリージョンアーキテクチャを作成できます。AWS リージョン間を物理的および論理的に分離することで、リージョン間の相関する障害を回避できます。言い換えると、EC2 を複数のアベイラビリティーゾーンにデプロイすることでアベイラビリティーゾーンの分離の利点が得られるように、リージョンサービスを複数のリージョンにデプロイすることで同じ利点が得られます。そのためには、アプリケーションにマルチリージョンアーキテクチャを実装する必要があります。これにより、リージョナルサービスの障害に対する回復力を高めることができます。

ただし、マルチリージョンアーキテクチャの利点を実現するのは難しい場合があります。アプリケーションレベルで何も取り消すことなく、リージョンの分離を活用するには、慎重な作業が必要です。例えば、リージョン間でアプリケーションをフェイルオーバーする場合は、各リージョンでアプリケーションスタック間の分離を厳密に維持し、すべてのアプリケーションの依存関係に注意しながら、アプリケーションのすべての部分を一緒にフェイルオーバーする必要があります。これをアプリケーション間の依存関係が多い複雑なマイクロサービスベースのアーキテクチャで実現するには、多くのエンジニアリングチームおよびビジネスチーム間での計画と調整が必要です。個々のワークロードが独自のフェイルオーバー決定を行えるようにすると、調整の複雑さは軽減されますが、単一リージョンの場合と比べて、複数のリージョン間で発生するレイテンシーが大きく異なるため、モーダル動作が発生します。

現時点では、AWS はクロスリージョンの同期レプリケーション機能を提供していません。リージョン間で (AWS が提供する) データストアの非同期レプリケーションを使用すると、リージョン間でアプリケーションをフェイルオーバーしたときに、データの損失や不整合が発生する可能性があります。不整合が発生した場合に緩和するには、信頼できるデータ調整プロセスが必要であり、ワークロードポートフォリオ全体で複数のデータストアの運用が必要になる場合があります。そうでない場合は、データ損失を許容する覚悟が必要です。最後に、フェイルオーバーをテストして、必要なときにそれが機能することを確認する必要があります。フェイルオーバーを実践するためにアプリケーションをリージョン間で定期的にローテーションすることは、かなりの時間とリソースの投資が必要になります。リージョン間でのデータストアの同期レプリケーションを使用して複数のリージョンからのアプリケーションの同時実行をサポートする場合、数百マイルまたは数千マイルにわたるデータベースのパフォーマンス特性およびレイテンシーは、単一リージョンで動作するデータベースとは大きく異なります。このため、この動作を考慮してアプリケーションスタックをゼロから計画する必要があります。また、両方のリージョンの可用性がハード依存関係となるため、ワークロードの回復力が低下する可能性があります。

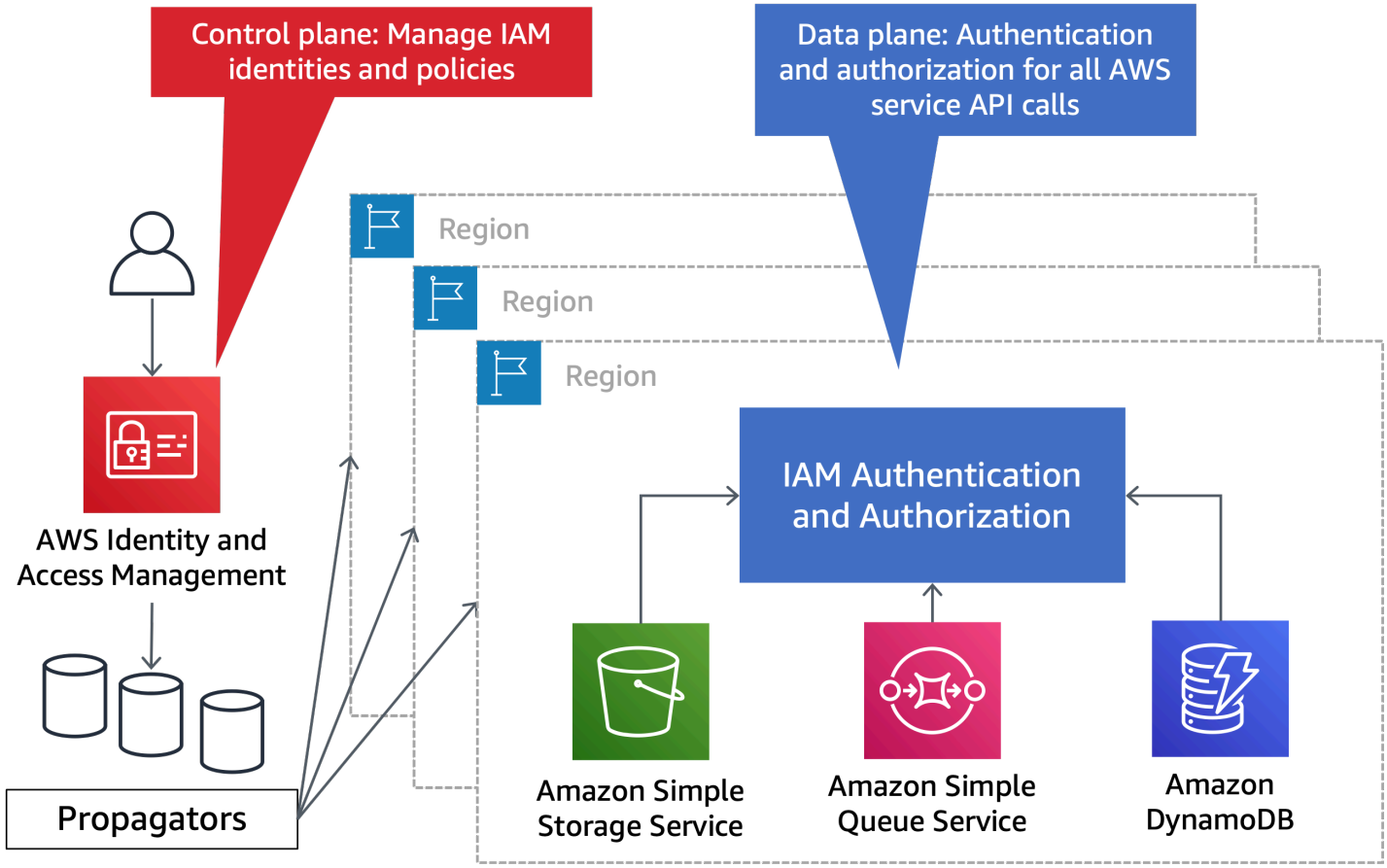
グローバルサービス

AWS のリージョンサービスとゾーンサービスに加えて、コントロールプレーンとデータプレーンがリージョン別に独立していない少数のAWS のサービスがあります。これらのリソースはリージョン固有ではないため、一般的にグローバルサービスと呼ばれます。AWS のグローバルサービスは、静的安定性を実現するために、コントロールプレーンとデータプレーンを分離するという AWS の従来の設計パターンに従っています。ほとんどのグローバルサービスの大きな違いは、コントロールプレーンが単一の AWS リージョンでホストされているのに対し、データプレーンはグローバルに分散されていることです。グローバルサービスには 3 つのタイプがあり、選択した設定によってはグローバルに見える一連のサービスもあります。

以下のセクションでは、各タイプのグローバルサービスと、これらのコントロールプレーンとデータプレーンがどのように分離されているかについて説明します。この情報を参考にして、グローバルサービスのコントロールプレーンに依存することなく、信頼性の高い高可用性 (HA) およびディザスタリカバリ (DR) メカニズムを構築できます。このアプローチは、グローバルサービスのコントロールプレーンがホストされているリージョンとは異なるリージョンで運用を行っている場合でも、アーキテクチャ内の単一障害点を排除し、クロスリージョンの潜在的な影響を回避するのに役立ちます。また、グローバルサービスのコントロールプレーンに依存しないフェイルオーバーメカニズムを安全に実装するのにも役立ちます。

パーティション固有のグローバルサービス

AWS の一部のグローバルサービスは、パーティション固有のサービスです (このホワイトペーパーではパーティションサービスと呼びます)。パーティションサービスは、単一の AWS リージョンでコントロールプレーンを提供します。一部のパーティションサービス (AWS Network Manager など) は、コントロールプレーンのみのサービスであり、他のサービスのデータプレーンをオーケストレーションします。他のパーティションサービス (IAM など) は、パーティション内のすべての AWS リージョンに分離および分散された独自のデータプレーンを持っています。パーティションサービスで障害が発生しても、他のパーティションには影響しません。aws パーティションの場合、IAM サービスのコントロールプレーンは us-east-1 リージョン内にあり、データプレーンはパーティション内の各リージョンに分離されています。パーティションサービスの独立したコントロールプレーンとデータプレーンは、aws-us-gov パーティションと aws-cn パーティションにもあります。IAM のコントロールプレーンとデータプレーンの分離を次の図に示します。



IAM には 1 つのコントロールプレーンとリージョンごとのデータプレーンがある

aws パーティション内のパーティションサービスとそのコントロールプレーンの場所は次のとおりです。

- AWS IAM (us-east-1)
- AWS Organizations (us-east-1)
- AWS アカウント管理 (us-east-1)
- Route 53 Application Recovery Controller (ARC) (us-west-2) - このサービスは aws パーティションにのみ存在する
- AWS Network Manager (us-west-2)
- Route 53 Private DNS (us-east-1)

これらのサービスのコントロールプレーンのいずれかで可用性に影響するイベントが発生した場合、これらのサービスが提供する CRUDL タイプのオペレーションは使用できなくなる可能性があります。したがって、リカバリ戦略がこれらのオペレーションに依存している場合、コントロールプレーンまたはコントロールプレーンをホストしているリージョンの可用性に影響が出ると、リカバリが成功する可能性が低くなります。リカバリ時にグローバルサービスのコントロールプレーンへの依存関係を削除する戦略については、「付録 A - パーティションサービスに関するガイダンス」を参照してください。

> **レコメンデーション**
>
> リカバリパスでは、パーティションサービスのコントロールプレーンに依存しないでください。代わりに、これらのサービスのデータプレーンオペレーションに依存します。パーティションサービスの設計方法の詳細については、「付録 A - パーティションサービスに関するガイダンス」を参照してください。

エッジネットワークにおけるグローバルサービス

以下の一連の AWS グローバルサービスは、コントロールプレーンを aws パーティション内で運用し、データプレーンをグローバル Point of Presence (POP) インフラストラクチャ (および、場合によっては AWS リージョン) でホストします。PoP でホストされているデータプレーンには、インターネットからだけでなく、任意のパーティションのリソースからもアクセスできます。例えば、Route 53 はコントロールプレーンを us-east-1 リージョン内で運用していますが、そのデータプレーンは世界中の何百という PoP と各 AWS リージョンに (リージョン内の Route 53 Public DNS と Private DNS をサポートするために) 分散されています。Route 53 のヘルスチェックもデータプレーンの一部であり、aws パーティション内の 8 つの AWS リージョンから実行されます。クライアントは、AWS 仮想プライベートクラウド (VPC) からでも、他のパーティション (GovCloud など) を含むインターネット上のどこからでも、Route 53 パブリックホストゾーンを使用して DNS

を解決できます。以下は、aws パーティション内のグローバルエッジネットワークサービスとそのコントロールプレーンの場所です。

- Route 53 Public DNS (us-east-1)
- Amazon CloudFront (us-east-1)
- AWS WAF Classic for CloudFront (us-east-1)
- AWS WAF for CloudFront (us-east-1)
- Amazon Certificate Manager (ACM) for CloudFront (us-east-1)
- AWS Global Accelerator (AGA) (us-west-2)
- AWS Shield Advanced (us-east-1)

EC2 インスタンスまたは Elastic IP アドレスで AGA ヘルスチェックを使用する場合、これらは Route 53 ヘルスチェックを使用します。AGA ヘルスチェックの作成または更新は、us-east-1 の Route 53 のコントロールプレーンに依存します。AGA ヘルスチェックの実行には、Route 53 ヘルスチェックデータプレーンを利用します。

これらのサービスのコントロールプレーンをホストするリージョンに影響する障害が発生するか、コントロールプレーン自体に影響する障害が発生した場合、これらのサービスが提供する CRUDL タイプのオペレーションは使用できない可能性があります。これらのオペレーションに依存するリカバリ戦略は、これらのサービスのデータプレーンにのみ依存する戦略と比べて、成功する可能性が低くなる可能性があります。

> **レコメンデーション**
>
> リカバリパスでは、エッジネットワークサービスのコントロールプレーンに依存しないでください。代わりに、これらのサービスのデータプレーンオペレーションに依存します。エッジネットワークにおけるグローバルサービスの設計方法の詳細については、「付録 B - エッジネットワークのグローバルサービスに関するガイダンス」を参照してください。

グローバルな単一リージョンのオペレーション

最後のカテゴリは、前のカテゴリのようにサービス全体ではなく、サービス内でグローバルな影響スコープを持つ特定のコントロールプレーンオペレーションで構成されます。ゾーンサービスやリージョンサービスとのやり取りは、指定したリージョンで行いますが、特定のオペレーションはリソースの配置先とは異なる単一リージョンに基盤となる依存関係があります。これらは単一リージョンで

のみ提供されるサービスとは異なります。後者のサービスのリストについては、「付録 C - 単一リージョンのサービス」を参照してください。

基盤となるグローバル依存関係に影響する障害が発生すると、依存オペレーションの CRUDL タイプのアクションは使用できなくなる場合があります。これらのオペレーションに依存するリカバリ戦略は、これらのサービスのデータプレーンにのみ依存する戦略と比べて、成功する可能性が低くなる場合があります。リカバリ戦略では、これらのオペレーションに依存しないようにする必要があります。

以下は、他のサービスが依存する可能性がある、グローバルなスコープを持つサービスのリストです。

- Route 53

  AWS のサービスのいくつかは、リソース固有の DNS 名を提供するリソースを作成します。例えば、Elastic Load Balancing (ELB) をプロビジョニングすると、サービスは ELB 用に Route 53 のパブリック DNS レコードとヘルスチェックを作成します。これは、`us-east-1` の Route 53 のコントロールプレーンに依存します。使用する他のサービスでも、コントロールプレーンのワークフローの一部として、ELB のプロビジョニング、パブリック Route 53 DNS レコードの作成、または Route 53 ヘルスチェックの作成が必要になる場合があります。例えば、Amazon API Gateway REST API リソース、Amazon Relational Database Service (Amazon RDS) データベース、Amazon Relational Database Service (Amazon RDS) データベース、または Amazon OpenSearch Service ドメインのプロビジョニングは、いずれも Route 53 での DNS レコードの作成を伴います。以下は、DNS レコードやホストゾーンを作成、更新、削除したり、Route 53 ヘルスチェックを作成したりするために、`us-east-1` で Route 53 のコントロールプレーンに依存するコントロールプレーンを持つサービスのリストです。このリストはすべてを網羅しているわけではなく、Route 53 のコントロールプレーンに依存してリソースの作成、更新、または削除のコントロールプレーンアクションを行う、最もよく使用されるサービスのいくつかを紹介しています。

  - Amazon API Gateway REST と HTTP API
  - Amazon RDS インスタンス
  - Amazon Aurora データベース
  - Amazon ELB ロードバランサー
  - AWS PrivateLink VPC エンドポイント
  - AWS Lambda URL
  - Amazon ElastiCache
  - Amazon OpenSearch Service

- Amazon CloudFront
- Amazon MemoryDB
- Amazon Neptune
- Amazon DynamoDB Accelerator (DAX)
- AGA
- DNS ベースのサービス検出を備えた Amazon Elastic Container Service (Amazon ECS) (AWS Cloud Map API を使用して Route 53 DNS を管理する)
- Amazon EKS Kubernetes コントロールプレーン

  EC2 インスタンスのホスト名の VPC DNS サービスはそれぞれ独立して AWS リージョンに存在し、Route 53の コントロールプレーンには依存しない点に注意することが重要です。AWS が VPC DNS サービス内で EC2 インスタンス用に作成するレコード (`ip-10-0-10.ec2.internal`、`ip-10-0-1-5.compute.us-west-2.compute.internal`、`i-0123456789abcdef.ec2.internal`、`i-0123456789abcdef.us-west-2.compute.internal` など) は、`us-east-1` の Route 53 のコントロールプレーンには依存しません。

  > **レコメンデーション**
  >
  > リカバリパスでは、Route 53 のリソースレコード、ホストゾーン、またはヘルスチェックの作成、更新、または削除を必要とするリソースの作成、更新、または削除に依存しないでください。リカバリパスで Route 53 のコントロールプレーンに依存しないように、これらのリソース (ELB など) は事前にプロビジョニングします。

- Amazon S3

  以下の Amazon S3 のコントロールプレーンオペレーションには、`aws` パーティションの `us-east-1` への基盤となる依存関係があります。`us-east-1` で Amazon S3 やその他のサービスに影響する障害が発生すると、他のリージョンでこれらのコントロールプレーンアクションが機能しなくなる可能性があります。

```
PutBucketCors
DeleteBucketCors
PutAccelerateConfiguration
PutBucketRequestPayment
```

```
PutBucketObjectLockConfiguration
PutBucketTagging
DeleteBucketTagging
PutBucketReplication
DeleteBucketReplication
PutBucketEncryption
DeleteBucketEncryption
PutBucketLifecycle
DeleteBucketLifecycle
PutBucketNotification
PutBucketLogging
DeleteBucketLogging
PutBucketVersioning
PutBucketPolicy
DeleteBucketPolicy
PutBucketOwnershipControls
DeleteBucketOwnershipControls
PutBucketAcl
PutBucketPublicAccessBlock
DeleteBucketPublicAccessBlock
```

Amazon S3 マルチリージョンアクセスポイント (MRAP) のコントロールプレーンは、us-west-2 でのみホストされているため、MRAP の作成、更新、または削除のリクエストは、このリージョンに直接送信されます。また、MRAP のコントロールプレーンには、us-west-2 の AGA、us-east-1 の Route 53、および各リージョンの ACM への基盤となる依存関係もあり、これらのコンテンツを処理するように MRAP が設定されています。リカバリパスや独自のシステムのデータプレーンでは、MRAP のコントロールプレーンの可用性に依存しないでください。これは、MRAP 内の各バケットをアクティブまたはパッシブのルーティングステータスに指定するために使用される MRAP フェイルオーバーコントロールとは異なります。これらの API は 5 つの AWS リージョンでホストされ、サービスのデータプレーンを使用してトラフィックを効果的にシフトするために使えます。

さらに、Amazon S3 のバケット名はグローバルに一意であり、名前の一意性を確保するために CreateBucket API と DeleteBucket API へのすべての呼び出しは aws パーティションの us-east-1 に依存します。これは、API の呼び出しが、バケットを作成する特定のリージョンに向けられる場合でも変わりません。最後に、重要なバケットの作成ワークフローがある場合は、バケット名の特定のスペル、特に識別可能なパターンに従うスペルの可用性に依存するべきではありません。

> **レコメンデーション**
>
> リカバリパスの一部として、S3 バケットの削除や新規作成、S3 バケット設定の更新に依存しないでください。変更を加えなくても障害から回復できるように、すべての必要な S3 バケットを必要な設定で事前にプロビジョニングします。このアプローチは MRAP にも当てはまります。

- CloudFront

  Amazon API Gateway は、エッジ最適化 API エンドポイントを提供します。これらのエンドポイントの作成は、ゲートウェイエンドポイントの前にディストリビューションを作成するために、`us-east-1` の CloudFront のコントロールプレーンに依存します。

  > **レコメンデーション**
  >
  > リカバリパスの一部として、新しいエッジ最適化 API ゲートウェイエンドポイントの作成に依存しないでください。すべての必要な API ゲートウェイエンドポイントは、事前にプロビジョニングします。

  このセクションで説明する依存関係はすべてコントロールプレーンのアクションであり、データプレーンのアクションではありません。静的に安定するようにワークロードを設定した場合、これらの依存関係はリカバリパスに影響を与えないはずです。なお、静的安定性の実装には追加の作業やサービスが必要であることに留意してください。

デフォルトのグローバルエンドポイントを使用するサービス

AWS のサービスが、AWS セキュリティトークンサービス (AWS STS) など、デフォルトのグローバルエンドポイントを提供する場合もあります。他のサービスは、このデフォルトのグローバルエンドポイントをデフォルト設定で使用することがあります。つまり、使用しているリージョンサービスが、単一の AWS リージョンにグローバルに依存している場合があるということです。以下では、サービスをリージョンレベルで利用するのに役立つデフォルトのグローバルエンドポイントへの意図しない依存関係を取り除く方法について詳しく説明します。

AWS STS: STS は、IAM ユーザーまたは認証するユーザー (フェデレーションユーザー) の一時的な、権限の制限された認証情報をリクエストできるウェブサービスです。AWS Software

Development Kit (SDK) とコマンドラインインターフェイス (CLI) からの STS の使用は、デフォルトの場所が us-east-1 になります。STS サービスはリージョンエンドポイントも提供します。これらのエンドポイントは、デフォルトで有効になっているリージョンにおいて、デフォルトで有効になります。これらのエンドポイントは、SDK または CLI を「AWS STS のリージョン別のエンドポイント」の手順に従って設定することで、いつでも利用できます。SigV4A を使用する場合も、リージョン STS エンドポイントに対して一時的な認証情報をリクエストすることが必要になります。このオペレーションにグローバル STS エンドポイントを使用することはできません。

> **レコメンデーション**
>
> リージョン STS エンドポイントを使用するように SDK と CLI の設定を更新してください。

Security Assertion Markup Language (SAML) サインイン: SAML サービスはすべての AWS リージョン で使用できます。このサービスを使用するには、https://us-west-2.signin.aws.amazon.com/saml などの 適切なリージョンの SAML エンドポイントを選択します。リージョンエンドポイントを使用するには、信頼ポリシーと ID プロバイダー (IdP) の設定を更新する必要があります。具体的な詳細については、AWS SAML のドキュメントを参照してください。

AWS で同じくホストされている IdP を使用している場合は、AWS での障害発生時に、これらも影響を受ける可能性があります。その結果、IdP 設定を更新できなくなったり、フェデレーションが完全にできなくなったりする場合があります。IdP に障害が発生したり、利用できなくなったりした場合に備えて、「ブレークグラス」ユーザーを事前にプロビジョニングする必要があります。静的に安定した方法でブレークグラスユーザーを作成する方法の詳細については、「付録 A - パーティションサービスに関するガイダンス」を参照してください。

> **レコメンデーション**
>
> 複数のリージョンからの SAML ログインを受け入れるように IAM ロールの信頼ポリシーを更新してください。障害が発生して、優先するエンドポイントが損なわれた場合は、別のリージョンの SAML エンドポイントを使用するように IdP 設定を更新します。IdP に障害が発生した場合や利用できない場合に備えて、ブレークグラスのユーザーを作成します。

AWS IAM Identity Center: Identity Center は、お客様の AWS アカウントおよびクラウドアプリケーションへのシングルサインオンを一元的に管理しやすくするクラウドベースのサービスです。Identity Center は、選択した単一リージョンにデプロイする必要があります。ただし、サービスのデフォルト動作では、us-east-1 でホストされているグローバル SAML エンドポ

イント (https://signin.aws.amazon.com/saml) を使用します。Identity Center を別の AWS リージョンにデプロイした場合は、Identity Center のデプロイと同じリージョンコンソールエンドポイントを対象とするように設定されたすべてのアクセス許可セットの relaystate URL を更新する必要があります。例えば、Identity Center を us-west-2 にデプロイした場合は、https://us-west-2.console.aws.amazon.com を使用するようにアクセス許可セットの relaystate を更新する必要があります。これにより、Identity Center のデプロイから us-east-1 への依存関係が削除されます。

さらに、IAM Identity Center をデプロイできるのは 1 つのリージョンに限られるため、デプロイに障害が発生した場合に備えて、「ブレークグラス」ユーザーを事前にプロビジョニングしておく必要があります。静的に安定した方法でブレークグラスのユーザーを作成する方法の詳細については、「付録 A - パーティションサービスに関するガイダンス」を参照してください。

> **レコメンデーション**
>
> IAM Identity Center のアクセス許可セットの relaystate URL を、サービスのデプロイ先のリージョンと一致するように設定します。IAM Identity Center のデプロイが利用できない場合に備えて、ブレークグラスのユーザーを作成します。

Amazon S3 ストレージレンズ: ストレージレンズには、default-account-dashboard というデフォルトのダッシュボードがあります。ダッシュボード設定および関連するメトリクスは、us-east-1 に保存されます。ダッシュボード設定およびメトリクスデータのホームリージョンを指定することで、他のリージョンに追加のダッシュボードを作成できます。

> **レコメンデーション**
>
> us-east-1 でサービスに影響する障害が発生したときに、デフォルトの S3 ストレージレンズダッシュボードからのデータが必要な場合は、代替のホームリージョンに追加のダッシュボードを作成してください。追加のリージョンで作成した他のカスタムダッシュボードを複製することもできます。

グローバルサービスの概要

グローバルサービスのデータプレーンには、AWS のリージョンサービスと同様の分離と独立性の原則が適用されます。特定のリージョンで IAM のデータプレーンに影響する障害は、別の AWS リージョンでの IAM のデータプレーンオペレーションには影響しません。同様に、特定の PoP で Route

53 のデータプレーンに影響する障害は、他の POP での Route 53 のデータプレーンオペレーションには影響しません。したがって、考慮しなければならないのは、コントロールプレーンが動作しているリージョンまたはコントロールプレーン自体に影響するサービスの可用性イベントです。各グローバルサービスのコントロールプレーンは 1 つのみであるため、そのコントロールプレーンに影響する障害は、リージョンをまたいで CRUDL タイプのオペレーション (サービスを直接使用する場合とは対照的に、サービスのセットアップや設定に通常使用する設定オペレーション) に影響する可能性があります。

グローバルサービスを回復力の高い方法で使用するようにワークロードを設計する最も効果的な方法は、静的安定性を使用することです。障害シナリオでは、コントロールプレーンを変更しなくても影響を軽減したり、別の場所にフェイルオーバーしたりできるようにワークロードを設計します。この種のグローバルサービスを活用してコントロールプレーンへの依存関係を削除し、単一障害点をなくす方法の規範的なガイダンスについては、「付録 A - パーティションサービスに関するガイダンス」と「付録 B - エッジネットワークのグローバルサービスに関するガイダンス」を参照してください。リカバリのためにコントロールプレーンオペレーションからのデータが必要な場合は、このデータを AWS Systems Manager パラメータストア (SSM パラメータストア) のパラメータ、DynamoDB テーブル、S3 バケットなど、データプレーンを介してアクセスできるデータストアにキャッシュします。冗長性を確保するために、データを追加のリージョンに保存することもできます。例えば、Route 53 Application Recovery Controller (ARC) のベストプラクティスに従い、5 つのリージョンクラスターエンドポイントをハードコーディングするか、ブックマークする必要があります。障害イベントが発生すると、信頼性が非常に高いデータプレーンクラスターでホストされていない Route 53 ARC API オペレーションなど、一部の API オペレーションにアクセスできなくなることがあります。`DescribeCluster` API オペレーションを使用して Route 53 ARC クラスターのエンドポイントを一覧表示できます。

以下は、グローバルサービスのコントロールプレーンへの依存関係を引き起こす、最も一般的な設定ミスまたはアンチパターンのいくつかをまとめたものです。

- フェイルオーバーを実行するために Route 53 レコードを変更する (A レコードの値を更新したり、加重レコードセットの重みを変更するなど)。
- フェイルオーバー中に IAM ロールやポリシーなどの IAM リソースを作成または更新する。これは通常、意図的なものではありませんが、テストされていないフェイルオーバープランの結果である可能性があります。
- 障害イベントの発生時にオペレーターが本番環境にアクセスするために IAM Identity Center に依存する。
- IAM Identity Center を別のリージョンにデプロイしている場合に、`us-east-1` でコンソールを使用するために Identity Center のデフォルト設定に依存する。

- リージョンフェイルオーバーを手動で実行するために AGA トラフィックダイヤルの重みを変更する。
- 障害が発生したオリジンからフェイルアウェイするように CloudFront ディストリビューションのオリジン設定を更新する。
- Route 53 の DNS レコードの作成に依存するディザスタリカバリ (DR) リソース (障害発生時の ELB や RDS インスタンスなど) をプロビジョニングする。

以下は、以上の一般的なアンチパターンを防ぐために役立つグローバルサービスを回復力の高い方法で使用するために、このセクションで紹介したレコメンデーションのまとめです。

レコメンデーションのまとめ

リカバリパスでは、パーティションサービスのコントロールプレーンに依存しません。代わりに、これらのサービスのデータプレーンオペレーションに依存します。パーティションサービスの設計方法の詳細については、「付録 A - パーティションサービスに関するガイダンス」を参照してください。

リカバリパスでは、エッジネットワークサービスのコントロールプレーンに依存しません。代わりに、これらのサービスのデータプレーンオペレーションに依存します。エッジネットワークにおけるグローバルサービスの設計方法の詳細については、「付録 B - エッジネットワークのグローバルサービスに関するガイダンス」を参照してください。

リカバリパスでは、Route 53 のリソースレコード、ホストゾーン、またはヘルスチェックの作成、更新、削除を必要とするリソースの作成、更新、または削除に依存しません。リカバリパスでは、Route 53 のコントロールプレーンへの依存関係を防ぐために、これらのリソース (ELB など) を事前にプロビジョニングします。

リカバリパスの一部として、S3 バケットの削除や新規作成、S3 バケット設定の更新に依存しません。変更を加えなくても障害から回復できるように、すべての必要な S3 バケットを必要な設定で事前にプロビジョニングします。このアプローチは MRAP にも当てはまります。

リカバリパスの一部として、新しいエッジ最適化 API ゲートウェイエンドポイントの作成に依存しません。すべての必要な API Gateway エンドポイントを事前にプロビジョニングします。

リージョン STS エンドポイントを使用するように SDK と CLI の設定を更新します。

複数のリージョンからの SAML ログインを受け入れるように IAM ロールの信頼ポリシーを更新します。障害が発生して、優先するエンドポイントが損なわれた場合は、別のリージョンの SAML エンドポイントを使用するように IdP 設定を更新します。IdP に障害が発生した場合や利用できない場合に備えて、break-glass ユーザーを作成します。

IAM Identity Center のアクセス許可セットの relaystate URL を、サービスのデプロイ先のリージョンと一致するように設定します。IAM Identity Center のデプロイが利用できない場合に備えて、ブレークグラスのユーザーを作成します。
us-east-1 でサービスに影響する障害が発生したときに、デフォルトの S3 ストレージレンズダッシュボードからのデータが必要な場合は、代替のホームリージョンに追加のダッシュボードを作成します。追加のリージョンで作成した他のカスタムダッシュボードを複製することもできます。

結論

AWS には、障害分離境界用の複数の異なる構成が用意されています。ゾーン、リージョン、およびグローバルサービスをどのように設計するかに加えて、コントロールプレーンでの障害発生時にワークロードが受ける潜在的な影響やワークロードの回復力について検討する必要があります。静的安定性は、AWS のサービスを使用するときに、コントロールプレーンへの依存を回避し、信頼性と耐障害性に優れた HA および DR メカニズムを作成できる主な方法の 1 つです。

付録 A - パーティションサービスに関するガイダンス

パーティションサービスの場合は、AWS のサービスのコントロールプレーンに障害が発生してもワークロードの回復力を維持するために、静的安定性を実装する必要があります。以下では、パーティションサービスへの依存関係をどのように考慮するかと、コントロールプレーンの障害発生時に何が機能し、何が機能しない可能性があるかについての規範的なガイダンスを示します。

AWS Identity and Access Management (IAM)

AWS Identity and Access Management (IAM) のコントロールプレーンは、IAM のすべてのパブリック API で構成されます (Access Advisor は含みますが、Access Analyzer や IAM Roles Anywhere は含みません)。これには、`CreateRole`、`AttachRolePolicy`、`ChangePassword`、`UpdateSAMLProvider`、`UpdateLoginP` どのアクションが含まれます。IAM のデータプレーンは、各 AWS リージョンの IAM プリンシパルに対して認証と承認を提供します。コントロールプレーンに障害が発生すると、IAM の CRUDL タイプのオペレーションは機能しない可能性がありますが、既存のプリンシパルの認証と承認は引き続き機能します。STS は、IAM とは別のデータプレーン専用のサービスであり、IAM のコントロールプレーンには依存しません。

つまり、IAM への依存関係を計画する場合、リカバリパスでは IAM のコントロールプレーンに依存すべきではありません。例えば、「ブレークグラス」の管理者ユーザー向けの静的に安定した設計では、適切なアクセス許可をアタッチしたユーザーの作成、パスワードの設定、アクセスキーとシークレットアクセスキーのプロビジョニングを行い、これらの認証情報を物理または仮想のボールトにロックします。緊急時に必要になった場合は、ボールトからユーザー認証情報を取り出し、ニーズに応じて使用します。静的に安定していない設計の場合、障害が発生した時点でユーザーをプロビジョニングするか、事前にユーザーをプロビジョニングしておきますが、管理者ポリシーは必要になったときにのみアタッチします。これらのアプローチは IAM のコントロールプレーンに依存します。

AWS Organizations

AWS Organizations のコントロールプレーンは、Organizations の `AcceptHandshake`、`AttachPolicy`、`CreateAccount`、`CreatePolicy`、`ListAccounts` などのすべてのパブリック API で構成されます。AWS Organizations には、データプレーンがありません。これは、他のサービス (IAM など) のデータプレーンをオーケストレーションします。コントロールプレーンに障害が発生すると、Organizations の CRUDL タイプのオペレーションは機能しない可能性がありますが、サービスコントロールポリシー (SCP) やタグポリシーなどのポリシーは引

き続き機能し、IAM 承認プロセスの一部として評価されます。Organizations がサポートする、他の AWS のサービスの委任された管理者機能やマルチアカウント機能も引き続き機能します。

つまり、AWS Organizations への依存関係を計画する場合、リカバリパスでは Organizations のコントロールプレーンに依存すべきではありません。代わりに、復旧計画に静的安定性を実装してください。例えば、静的に安定していないアプローチでは、`aws:RequestedRegion` 条件を介して許可した AWS リージョンに対する制限を削除したり、特定の IAM ロールで管理者アクセス許可を有効にしたりするように SCP を更新します。この更新は、Organizations のコントロールプレーンに依存します。より適切なアプローチは、セッションタグを使用して管理者アクセス許可の使用を許可することです。ID プロバイダー (IdP) には、`aws:PrincipalTag` 条件に照らして評価できるセッションタグを含めることができます。これにより、SCP を静的なままに維持して、特定のプリンシパルに対するアクセス許可を動的に設定できます。これに伴って、コントロールプレーンへの依存関係がなくなり、データプレーンのアクションのみが使用されます。

AWS Account Management

AWS Account Management のコントロールプレーンは us-east-1 でホストされ、AWS アカウントを管理するためのすべてのパブリック API (`GetContactInformation` や `PutContactInformation` など) で構成されます。これには、管理コンソールでの AWS アカウントの新規作成や閉鎖も含まれます。`CloseAccount`、`CreateAccount`、`CreateGovCloudAccount`、`DescribeAccount` の API は、AWS Organizations のコントロールプレーンの一部です。このコントロールプレーンも us-east-1 でホストされます。さらに、AWS Organizations 外からの GovCloud アカウントの作成は、us-east-1 での AWS アカウント管理のコントロールプレーンに依存します。また、GovCloud アカウントは `aws` パーティション内の AWS アカウントに 1 対 1 でリンクする必要があります。`aws-cn` パーティションでのアカウント作成は us-east-1 に依存しません。AWS アカウントのデータプレーンはアカウント自体です。コントロールプレーンに障害が発生すると、AWS アカウントの CRUDL タイプのオペレーション (新しいアカウントの作成や連絡先情報の取得と更新など) は機能しない場合があります。IAM ポリシーでのアカウントへの参照は引き続き機能します。

つまり、AWS Account Management への依存関係を計画する場合、リカバリパスでは Account Management のコントロールプレーンに依存すべきではありません。Account Management のコントロールプレーンには、リカバリ状況で通常使用するような直接的な機能はありませんが、必要になる場合もあります。例えば、静的に安定した設計では、フェイルオーバーに必要なすべての AWS アカウントを事前にプロビジョニングします。静的に安定していない設計では、障害の発生時に DR リソースをホストするために新しい AWS アカウントを作成します。

Route 53 Application Recovery Controller

Route 53 ARC のコントロールプレーンは、Amazon Route 53 Application Recovery Controller のエンドポイントとクォータで示しているように、リカバリコントロールとリカバリの準備のための API で構成されます。コントロールプレーンを使用して、準備状況チェック、ルーティングコントロール、およびクラスターオペレーションを管理します。ARC のデータプレーンはリカバリクラスターであり、Route 53 のヘルスチェックでクエリするルーティングコントロール値を管理し、安全ルールも実装します。Route 53 ARC のデータプレーン機能には、`https://aaaaaaaa.route53-recovery-cluster.eu-west-1.amazonaws.com` などのリカバリクラスター API を介してアクセスします。

つまり、リカバリパスでは Route 53 ARC のコントロールプレーンに依存すべきではありません。このガイダンスの実装に役立つベストプラクティスが 2 つあります。

- まず、リージョンクラスターの 5 つのエンドポイントをブックマークするか、ハードコーディングします。これにより、フェイルオーバーシナリオ中に DescribeCluster コントロールプレーンのオペレーションを使用してエンドポイント値を検出する必要がなくなります。
- 次に、AWS Management Console ではなく、CLI または SDK を通じて Route 53 ARC クラスター API を使用し、ルーティングコントロールを更新します。これにより、フェイルオーバープランで管理コンソールへの依存関係がなくなり、データプレーンのアクションのみに依存するようになります。

AWS Network Manager

AWS Network Manager サービスは、主に us-west-2 でホストされるコントロールプレーン専用のシステムです。その目的は、AWS アカウント、リージョン、オンプレミスの口ケーションにまたがる AWS クラウド ワイドエリアネットワーク (WAN) のコアネットワークと AWS Transit Gateway ネットワークの設定を一元管理することです。また、us-west-2 でクラウド WAN メトリクスを集約します。これには、CloudWatch データプレーンを介してアクセスすることもできます。Network Manager に障害が発生しても、それがオーケストレーションするサービスのデータプレーンには影響しません。クラウド WAN の CloudWatch メトリクスは、us-west-2 でも利用できます。us-west-2 に影響する障害の発生時に他のリージョンに移動するトラフィック量を把握するなどの運用上の目的のために、リージョンごとの入出力バイト数などのメトリクスの履歴データが必要な場合は、これらのメトリクスを CloudWatch コンソールから直接 CSV データとしてエクスポートするか、Amazon CloudWatch メトリクスを CSV ファイルに発行することができます。データは `AWS/Network Manager` 名前空間の下で見つかります。これを、選択したスケジュールで実行し、S3 や別の選択したデータストアにデータを保存できます。静的に安定したリカバリプランを実装するに

は、AWS Network Manager を使用してネットワークを更新したり、コントロールプレーンオペレーションからのデータにフェイルオーバー入力を依存したりしないでください。

Route 53 のプライベート DNS

Route 53 のプライベートホストゾーンは、パーティションごとにサポートされていますが、Route 53 のプライベートホストゾーンとパブリックホストゾーンの考慮事項は同じです。「付録 B - エッジネットワークのグローバルサービスに関するガイダンス」の「Amazon Route 53」を参照してください。

付録 B - エッジネットワークのグローバルサービスに関するガイダンス

エッジネットワークのグローバルサービスでは、AWS のサービスのコントロールプレーンに障害が発生したときにワークロードの回復力を維持するために、静的安定性を実装する必要があります。

Route 53

Route 53 のコントロールプレーンは、ホストゾーン、レコード、ヘルスチェック、DNS クエリログ、再利用可能な委託セット、トラフィックポリシー、コスト配分タグの各機能にわたる、Route 53 のすべてのパブリック API で構成されます。コントロールプレーンは、us-east-1 でホストされます。データプレーンは権威ある DNS サービスであり、200 を超える PoP ロケーションおよび各 AWS リージョンで実行され、ホストゾーンとヘルスチェックデータに基づいて DNS クエリに応答します。さらに、Route 53 にはヘルスチェック用のデータプレーンがあり、これも複数の AWS リージョンにまたがってグローバルに分散されたサービスです。このデータプレーンは、ヘルスチェックを実行し、結果を集約して、Route 53 のパブリックおよびプライベート DNS と AGA のデータプレーンに配信します。コントロールプレーンに障害が発生すると、Route 53 の CRUDL タイプのオペレーションは機能しない可能性がありますが、DNS 解決とヘルスチェック、およびヘルスチェックでの変更によるルーティングの更新は引き続き機能します。

つまり、Route 53 への依存関係を計画する場合、リカバリパスでは Route 53 のコントロールプレーンに依存すべきではありません。例えば、静的に安定した設計では、ヘルスチェックのステータスを利用してリージョン間でフェイルオーバーを実行したり、アベイラビリティーゾーンから退避したりします。Route 53 Application Recovery Controller (ARC) ルーティングコントロールを使用して、ヘルスチェックのステータスを手動で変更したり、DNS クエリへの応答を変更したりできます。ARC が提供しているものと同様のパターンがあり、要件に応じて実装できます。これらのパターンの一部については、「Route 53 を使用したディザスタリカバリメカニズムの作成」と「高度なマルチ AZ レジリエンスパターン」の「ヘルスチェックサーキットブレーカー」セクションで概説しています。マルチリージョン DR プランを使用する場合は、ELB や RDS インスタンスなど、DNS レコードの作成を必要とするリソースを事前にプロビジョニングします。静的に安定していない設計では、`ChangeResourceRecordSets` API 経由で Route 53 リソースレコードの値を更新したり、加重レコードの重みを変更したり、新しいレコードを作成してフェイルオーバーを実行したりします。これらのアプローチは、Route 53 のコントロールプレーンに依存します。

Amazon CloudFront

Amazon CloudFront のコントロールプレーンは、ディストリビューションを管理する CloudFront のすべてのパブリックAPI で構成され、us-east-1 でホストされます。データプレーンは、エッジネットワーク内の PoP から提供されるディストリビューション自体です。オリジンコンテンツのリクエスト処理、ルーティング、キャッシュを実行します。コントロールプレーンに障害が発生すると、CloudFront の CRUDL タイプのオペレーション (無効化リクエストを含む) は機能しない場合がありますが、コンテンツは引き続きキャッシュおよび配信され、オリジンフェイルオーバーは引き続き機能します。

つまり、CloudFront への依存関係を計画する場合、リカバリパスでは CloudFront のコントロールプレーンに依存すべきではありません。例えば、静的に安定した設計では、自動オリジンフェイルオーバーを使用して、いずれかのオリジンに対する障害の影響を軽減します。Lamda@Edge を使用してオリジン負荷分散またはフェイルオーバーを構築することもできます。このパターンの詳細については、「Amazon CloudFront を使用した高可用性アプリケーションのための 3 つの高度な設計パターン」および「Amazon CloudFront と Amazon S3 を使用してマルチリージョンの地理的に近接したアクティブ/アクティブなアプリケーションを構築する」を参照してください。静的に安定していない設計では、オリジンの障害に対応してディストリビューションの設定を手動で更新します。このアプローチは、CloudFront のコントロールプレーンに依存します。

Amazon Certificate Manager

CloudFront ディストリビューションでカスタム証明書を使用している場合は、ACM にも依存します。CloudFront ディストリビューションでのカスタム証明書の使用は、us-east-1 リージョンでの ACM のコントロールプレーンに依存します。コントロールプレーンに障害が発生しても、ディストリビューションで設定されている既存の証明書と、証明書の自動更新は引き続き機能します。リカバリパスの一部として、ディストリビューションの設定の変更や、新しい証明書の作成には依存しないでください。

AWS Web Application Firewall (WAF) および WAF クラシック

CloudFront ディストリビューションで AWS WAF を使用している場合は、同じく us-east-1 リージョンでホストされている WAF のコントロールプレーンに依存することになります。コントロールプレーンに障害が発生しても、設定済みのウェブアクセスコントロールリスト (ACL) および関連するルールは引き続き機能します。リカバリパスの一部として WAF のウェブ ACL の更新には依存しないでください。

AWS Global Accelerator

AGA のコントロールプレーンは、AGA のすべてのパブリック API で構成され、us-west-2 でホストされます。データプレーンは、登録済みのエンドポイントに対する AGA が提供するエニーキャスト IP アドレスのネットワークルーティングです。また、AGA は Route 53 ヘルスチェックを利用して、Route 53 のデータプレーンの一部である、AGA エンドポイントのヘルスを判断します。コントロールプレーンに障害が発生すると、AGA の CRUDL タイプのオペレーションは機能しない場合があります。既存のエンドポイントへのルーティングに加えて、他のエンドポイントやエンドポイントグループへのトラフィックのルーティングや移動に使用する既存のヘルスチェック、トラフィックダイヤル、エンドポイントの重み設定は、引き続き機能します。

つまり、AGA への依存関係を計画する場合、リカバリパスでは AGA のコントロールプレーンに依存すべきではありません。例えば、静的に安定した設計では、設定済みのヘルスチェックのステータスを利用して、異常のあるエンドポイントからフェイルアウェイします。この設定の例については、「AWS Global Accelerator を使用して AWS でマルチリージョンアプリケーションをデプロイする」を参照してください。静的に安定していない設計では、障害発生時に AGA トラフィックダイヤルのパーセンテージを変更したり、エンドポイントグループを編集したり、エンドポイントグループからエンドポイントを削除したりします。これらのアプローチは、AGA のコントロールプレーンに依存します。

Amazon Shield Advanced

Amazon Shield Advanced のコントロールプレーンは、Shield Advanced のすべてのパブリック API で構成され、us-east-1 でホストされます。これには、`CreateProtection`、`CreateProtectionGroup`、`AssociateHealthCheck`、`DesribeDRTAcces` どの機能が含まれます。データプレーンは、Shield Advanced が提供する DDoS 保護と、Shield Advanced メトリクスの作成です。Shield Advanced は、Route 53 のヘルスチェック (Route 53 のデータプレーンの一部) も利用します (設定している場合)。コントロールプレーンに障害が発生すると、Shield Advanced の CRUDL タイプのオペレーションは機能しない場合があります。ただし、リソースに設定されている DDoS 保護やヘルスチェックでの変更への対応は引き続き機能します。

つまり、リカバリパスでは Shield Advanced のコントロールプレーンに依存すべきではありません。Shield Advanced のコントロールプレーンには、リカバリ状況で通常使用するような直接的な機能はありませんが、必要になる場合があります。例えば、静的に安定した設計では、障害発生後に保護を設定するのではなく、DR リソースを保護グループの一部として事前に設定し、リソースにヘルスチェックを関連付けます。これにより、リカバリを Shield Advanced のコントロールプレーンに依存する必要がなくなります。

付録 C - 単一リージョンのサービス

以下は、単一リージョンでのみ利用できるサービス、またはそのサービスの特定の機能 (サービス名の後の括弧内に表示) のリストです。他のグローバルサービス用の静的安定性の実装に関するガイダンスは、これらのサービスのコントロールプレーンやデータプレーンへの依存関係を計画する必要がある場合にも当てはまります。

- Alexa for Business
- AWS Marketplace (AWS Marketplace Catalog API、AWS Marketplace Commerce Analytics、AWS Marketplace Entitlement Service)
- Billing and Cost Management (AWS Cost Explorer、AWS のコストと使用状況レポート、AWS Budgets、Savings Plans)
- AWS BugBust
- Amazon Mechanical Turk
- Amazon Chime
- Amazon Chime SDK (PSTN オーディオ、メッセージング、アイデンティティ)
- AWS Chatbot
- AWS DeepRacer
- AWS Device Farm
- Amazon GameSparks

寄稿者

このドキュメントの寄稿者は次のとおりです。

- Amazon Web Services、Principal Solutions Architect、Michael Haken

ドキュメントの改訂

このホワイトペーパーの更新に関する通知を受け取るには、RSS フィードにサブスクライブしてください。

変更	説明	日付
マイナーな改訂	IAM のベストプラクティスに合わせてガイドを更新しました。詳細については、「IAM でのセキュリティのベストプラクティス」を参照してください。	2023 年 2 月 9 日
初版発行	ホワイトペーパーを公開しました。	2022 年 11 月 16 日

AWS 用語集

AWS の最新の用語については、「AWS の用語集リファレンス」の「AWS 用語集」を参照してください。

注意

お客様は、本書に記載されている情報を独自に評価する責任を負うものとします。本書は、(a) 情報提供のみを目的とし、(b) AWS の現行製品と慣行について説明しており、これらは予告なしに変更されることがあり、(c) AWS およびその関連会社、サプライヤー、またはライセンサーからの契約上の義務や保証をもたらすものではありません。AWS の製品やサービスは、明示または黙示を問わず、一切の保証、表明、条件なしに「現状のまま」提供されます。お客様に対する AWS の責任は AWS 契約によって規定されます。本書は、AWS とお客様との間で締結されるいかなる契約の一部でもなく、その内容を修正するものでもありません。

© 2022 Amazon Web Services, Inc. or its affiliates. All rights reserved.