デベロッパーガイド

Amazon Application Recovery Controller (ARC)

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Application Recovery Controller (ARC): デベロッパーガイド

Copyright © 2026 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 Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

ARC とは 1
マルチアベイラビリティーゾーン復旧 1
マルチリージョンリカバリ 2
Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更 4
移行オプション 4
マルチ AZ とマルチリージョンの機能を比較する 5
マルチ AZ リカバリ 8
ゾーンシフト 8
ゾーンシフトの仕組み 9
AWS リージョン 10
ゾーンシフトのコンポーネント 15
データプレーンとコントロールプレーン 17
料金 18
ベストプラクティス 18
API オペレーション 20
CLI オペレーションの使用例 21
サポートされているリソース 25
ゾーンシフトの開始、更新またはキャンセル 37
ログ記録とモニタリング 39
ゾーンシフトの IAM 43
ゾーンオートシフト 54
ゾーンオートシフトの仕組み 56
AWS リージョン 65
ゾーンオートシフトのコンポーネント 65
データプレーンとコントロールプレーン 68
料金 69
ベストプラクティス 69
API オペレーション 73
CLI オペレーションの使用例 75
ゾーンオートシフトの有効化と操作 81
を使用したゾーンオートシフトのテスト AWS FIS 86
ログ記録とモニタリング 87
Identity and Access Management 98
クォータ 112

マルチリージョンリカバリ 114
ルーティングコントロール 114
ルーティングコントロールについて 115
AWS リージョン 118
コンポーネント 119
データプレーンとコントロールプレーン 122
Tagging 123
料金 124
マルチリージョンのリカバリの開始方法 124
ベストプラクティス 126
API オペレーション 129
CLI オペレーションの使用例 134
ルーティングコントロールコンポーネントを操作する 151
ログ記録とモニタリング 170
Identity and Access Management 175
クォータ 189
準備状況チェック 190
準備状況チェックとは 191
AWS リージョン 200
コンポーネント 201
データプレーンとコントロールプレーン 204
Tagging 205
料金 206
回復力のあるアプリケーションを設定する 206
ベストプラクティス 207
API オペレーション 208
CLI オペレーションの使用例 210
リカバリグループと準備状況チェックの使用 221
準備状況ステータスをモニタリングする 226
アーキテクチャの推奨事項を取得する 228
クロスアカウント認証の作成 230
準備状況ルール、リソースタイプ、ARNS 232
ログ記録とモニタリング 253
Identity and Access Management 269
クォータ 283
リージョン切り替え 284

リージョンの切り替えについて ... 285
ベストプラクティス ... 299
チュートリアル: アクティブ/パッシブプラン ... 301
チュートリアル: 自動生成をレポートする ... 308
チュートリアル: RDS 復旧後ワークフローを実行する ... 310
API オペレーション ... 312
リージョン切り替えの使用 ... 315
ダッシュボード ... 353
クロスアカウントのサポート ... 354
Identity and Access Management ... 360
ログ記録とモニタリング ... 382
クォータ ... 391
コードの例 ... 392
基本 ... 392
アクション ... 392
セキュリティ ... 403
データ保護 ... 404
保管中の暗号化 ... 405
転送中の暗号化 ... 405
Identity and Access Management ... 405
オーディエンス ... 405
アイデンティティを使用した認証 ... 405
ポリシーを使用したアクセスの管理 ... 407
Amazon Application Recovery Controller (ARC) 機能と IAM の連携方法 ... 409
アイデンティティベースのポリシーの例 ... 409
AWS マネージドポリシー ... 409
トラブルシューティング ... 416
AWS PrivateLink ... 418
ログ記録とモニタリング ... 420
コンプライアンス検証 ... 421
耐障害性 ... 421
インフラストラクチャセキュリティ ... 422
ドキュメント履歴 ... 423
... cdxli

ARC とは

Amazon Application Recovery Controller (ARC) は、 AWS グローバルクラウドインフラストラクチャで実行されているアプリケーションの迅速な復旧の準備と完了に役立ちます。

ARC には、次の機能が用意されています。

- ゾーンシフトやゾーンオートシフトなどのマルチアベイラビリティーゾーン (AZ) 復旧。これにより、障害のある AZ から正常な AZ にトラフィックを一時的に移行して、単一の AZ 障害から復旧できます。
- マルチリージョンリカバリ。これには、リージョンアプリケーションの復旧のためのルーティングコントロールとリージョン切り替えおよびアプリケーションモニタリングの準備状況チェックが含まれます。

マルチアベイラビリティーゾーン復旧

ゾーンシフト

ARC ゾーンシフトを使用すると、単一のアベイラビリティーゾーン (AZ) の障害をすばやく分離して復旧できます。ゾーンシフトは、サポートされているリソースのトラフィックを、障害のある AZ から同じ AWS リージョン内の正常な AZs に一時的に移行します。ゾーンシフトを開始すると、開発者の不正なコードデプロイや 1 つの AZ の AWS 障害などからアプリケーションをすばやく復旧できます。障害のある AZ からトラフィックを遠ざけると、障害のある AZ でアプリケーションを使用しているクライアントへの影響が軽減されます。

AWS リージョンのアカウントでサポートされているリソースに対してゾーンシフトを開始できます。ゾーンシフトは手動で一時的なものです。ゾーンシフトを開始するときは、最大 3 日間の (延長可能な) 有効期限を指定する必要があります。サポートされているリソースのゾーンシフトを有効にするには、「サポートされているリソース」を参照してください。

ゾーンオートシフト

ARC ゾーンオートシフトは、お客様に代わって、サポートされているリソースの障害のある AZ から同じ AWS リージョン内の正常な AZs にトラフィックを移行 AWS することを に許可します。は、内部テレメトリが、顧客に影響を与える可能性のある AWS リージョン内の 1 つの AZ に障害があることを示すと、ゾーンオートシフト AWS を開始します。内部テレメトリには、 AWS ネットワーク、Amazon EC2、Elastic Load Balancing サービスなど、複数のソースからのメトリクスが組み込まれています。

ゾーンオートシフトは一時的なものです。 は、内部テレメトリインジケータで問題や潜在的な問題がなくなったことが示されると、ゾーンオートシフトを AWS 終了します。

これらの機能の詳細については、次の章を参照してください。

- ARC のゾーンシフト
- ARC のゾーンオートシフト

マルチリージョンリカバリ

リージョン切り替え

ARC のリージョン切り替えは、マルチリージョンアプリケーション復旧のための一元化され、自動化された、確認可能なソリューションを提供します。リージョンの切り替えにより、アプリケーション全体の復旧を計画および調整し AWS リージョン、ビジネス継続性を確保して運用オーバーヘッドを削減できます。

リージョンスイッチを使用して、複数の AWS アカウント間でアプリケーションリソースの大規模で複雑な復旧タスクをオーケストレーションできます。に障害 AWS リージョン が発生した場合、リージョンスイッチを使用して作成したプランがフェイルオーバーしたり、リソースを別のリージョンに切り替えたりして、アプリケーションが正常に動作し続けることができます AWS リージョン。

ルーティングコントロール

ARC の非常に信頼性の高いルーティングコントロールにより、マルチリージョンリカバリが可能になり、アプリケーションは AWS リージョン間でドメインネームシステムの DNS トラフィックをフェイルオーバーできます。

アプリケーションが複数の AWS リージョンで動作するように設計されている場合は、ARC ルーティングコントロールを使用してリージョン間のフェイルオーバーを行うことができます。ルーティングコントロールを使用すると、障害のある AWS リージョンから正常な AWS リージョンにトラフィックをフェイルオーバーできるため、アプリケーションの可用性を維持できます。ルーティングコントロールには安全ルールが含まれ、自分で定義したガードレールを設けて、意図しない結果から自分を保護するのに役立ちます。例えば、アクティブまたはスタンバイのアプリケーションレプリカの 1 つだけが有効で使用中である安全ルールを強制できます。

準備状況チェック

ARC の準備状況チェックでは、 AWS リソースクォータ、容量、ネットワークルーティングポリシーを継続的にモニタリングし、レプリカアプリケーションへのフェイルオーバーやリージョンの障

害からの回復に影響する可能性のある変更について通知できます。継続的な準備状況チェックを行うことにより、マルチリージョンアプリケーションがフェイルオーバートラフィックを処理できるようにスケールおよび設定された状態にあることを確認できます。準備状況チェックは、ARC を初めて設定するときや、通常のアプリケーションオペレーション中に便利です。準備状況チェックは、イベント中のフェイルオーバーのクリティカルパスでの使用を意図していません。

これらの機能の詳細については、次の章を参照してください。

- ARC でのリージョン切り替え
- ARC でのルーティングコントロール
- ARC の準備状況チェック

Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。

慎重に検討した結果、2026 年 4 月 30 日以降、Amazon Application Recovery Controller (ARC) の準備状況チェック機能を新規顧客向けに終了することにしました。既存のお客様は、通常どおりサービスを引き続き使用できます。

ARC 準備状況チェックは、ディザスタリカバリのためのリソースの準備状況をモニタリングできる機能です。ARC は引き続き利用可能ですが、準備状況チェック機能は 2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。

> **Note**
>
> ARC および ARC リージョンスイッチは引き続き完全にサポートされています。この変更の影響を受けるのは、準備状況チェック機能のみです。リージョンスイッチ、ルーティングコントロール、ゾーンシフト、ゾーンオートシフトに変更はありません。

移行オプション

準備状況チェックと同様の機能については、マルチリージョンアプリケーションを ARC リージョンスイッチにオンボーディングすることをお勧めします。

ARC リージョンスイッチは、完全なマルチリージョンリカバリオーケストレーションを提供するフルマネージドサービスです。これには、計画評価と呼ばれる機能が含まれており、リージョン切り替え計画の状態を定期的にモニタリングして、実行の準備が整っていることを確認します。

ARC リージョンスイッチの使用を開始するには、「」を参照してくださいARC でのリージョン切り替え。

ARC でマルチ AZ とマルチリージョンのリカバリ機能を比較する

Amazon Application Recovery Controller (ARC) のゾーンシフト、ゾーンオートシフト、ルーティングコントロール、リージョン切り替えは、すべて AWS アプリケーションの迅速なリカバリを達成でき、回復力の維持に役立ちます。これらの機能は可用性が高く、アプリケーションでレイテンシーが増加したり可用性が低下したりするシナリオでの復旧のサポートに役立ちます。また、これらの機能は、トラフィックを独立した障害から遠ざけることにより、アプリケーションの迅速な復旧にも役立ちます。これにより、障害による影響と損失時間を制限します。

ルーティングコントロールとリージョン切り替えは、複数の AWS リージョン (マルチリージョン) にあるAWSアプリケーションに焦点を当てていますが、ゾーンシフトとゾーンオートシフトは、マルチ AZ アプリケーションでサポートされているリソースのトラフィックの移行のみをサポートします。

次の表の情報には、ARC のレジリエンス機能の主な機能の一部が含まれています。これらの説明は、特定のオプションがアプリケーションのニーズに最適な選択肢である可能性をよりよく理解するのに役立ちます。

ルーティングコントロール	リージョン切り替え	ゾーンシフト	ゾーンオートシフト
リージョン別 トラフィックをある AWS リージョンから別のリージョンに再ルーティングする (主として)	リージョン別 トラフィックをある AWS リージョンから別のリージョンに再ルーティングする (主として)	ゾーン別 トラフィックをアベイラビリティーゾーンから遠ざける トラフィックは特定のターゲットではなく、リージョン内の他のアベイラビリティーゾーンに移動する	ゾーン別 トラフィックをアベイラビリティーゾーンから遠ざける トラフィックは特定のターゲットではなく、リージョン内の他のアベイラビリティーゾーンに移動する

ルーティングコントロール	リージョン切り替え	ゾーンシフト	ゾーンオートシフト
セットアップが必要 構成とセットアップが必要	セットアップが必要 構成とセットアップが必要	セットアップが必要になる場合があります サポートされているリソースの一部にオプトインが必要です 詳細については、サポートされているリソース を参照してください。	セットアップが必要 サポートされているリソースに対して有効にする必要があります 詳細については、サポートされているリソース を参照してください。
顧客開始 トラフィックを再ルーティングするタイミングは顧客が決める	顧客開始 トラフィックを再ルーティングするタイミングは顧客が決める	顧客開始 ゾーンシフトを開始するタイミングは顧客が決める	AWS によって開始 AWS がユーザーに代わって、アプリケーショントラフィックを AZ から遠ざける
有料 ルーティングコントロールは別料金	有料 リージョン切り替えプランには個別の料金が必要です	サービスに付属 (追加料金なし) AZ からトラフィックを遠ざけるためのゾーンシフトの作成は、サポート対象のリソースに含まれる	サービスに付属 (追加料金なし) ユーザーに代わって AZ からトラフィックを遠ざけるためのオートシフトの開始は、サポート対象のリソースに含まれる
有効期限なし トラフィックはレプリカに無期限で再ルーティング可能	有効期限なし アプリケーションは無期限にレプリカに移行できます	一時的 すべてのゾーンシフトは有効期限を設定する必要がある	一時的 AWS がオートシフトを開始および終了する

これらの各機能の詳細については、次の章を参照してください。

- ARC のゾーンシフト
- ARC のゾーンオートシフト
- ARC でのルーティングコントロール
- ARC でのリージョン切り替え

ゾーンシフトとゾーンオートシフトを使用して ARC のアプリケーションを復旧する

このセクションでは、Amazon Application Recovery Controller (ARC) の機能を使用して、障害のあるアベイラビリティーゾーン (AZ) の問題から AWS リソースを確実に復旧する方法について説明します。ゾーンシフトとゾーンオートシフトは、サポートされているリソースのトラフィックを障害のある AZ から一時的に移行するため、アプリケーションの復旧までの時間が短縮されます。

ゾーンシフトとゾーンオートシフトの主な違いは、一方はユーザーが制御する手動トラフィックシフトであり、もう一方はユーザーに代わってトラフィックを自動的に障害から遠ざける点です。

- ゾーンシフトでは、AWS リージョン でサポートされているリソースのトラフィックをアベイラビリティーゾーンから手動でシフトします。
- ゾーンオートシフトを使用すると、サポートされているリソースのトラフィックは障害のある AZ から自動的に移行され、同じ AWS リージョン内の正常な AZ に再ルーティングされます。

以下のトピックでは、ゾーンシフト機能とゾーンオートシフト機能、それらを使用する方法について説明します。

トピック

- ARC のゾーンシフト
- ARC のゾーンオートシフト

ARC のゾーンシフト

Amazon Application Recovery Controller (ARC) ゾーンシフトを使用すると、サポートされているリソースのトラフィックを の障害のあるアベイラビリティーゾーン (AZ) から同じリージョンの正常な AZs AWS リージョン にシフトできます。障害が発生している AZ からリソースのトラフィックを切り替えることにより、停電や AZ 内のハードウェア/ソフトウェアの問題によって生じる影響の期間や深刻度を低減し、問題の緩和やアプリケーションの迅速な復旧に役立ちます。例えば、不適切なデプロイが原因でレイテンシーの問題が発生していたり、アベイラビリティーゾーンで障害が発生していたりする場合、トラフィックをシフトすることを選択できます。

ゾーンシフトを使用するには、リソースをオプトインする必要があります。詳細については、サポートされているリソース を参照してください。

ゾーンシフトを開始する前に、アプリケーションをプリスケールし、トラフィックをアベイラビリティーゾーンからシフトするのに十分な容量があることを確認する必要があります。プレスケーリング後、移行元のアベイラビリティーゾーンとトラフィックを移行するリソースを選択し、ゾーンシフトを開始できます。シフトはいつでもキャンセルでき、トラフィックを元のアベイラビリティーゾーンに戻すことができます。詳細については、ARC のゾーンシフトのベストプラクティスを参照してください。

すべてのゾーンシフトは一時的な緩和策です。ゾーンシフトを開始するときに、最初の有効期限を 1 分から 3 日 (72 時間) に設定します。トラフィックシフトを継続する必要がある場合は延長できます。

特定の状況では、ゾーンシフトによって AZ からトラフィックが切り替えられない場合があります。詳細については、「サポートされているリソース」を参照してください。

ゾーンシフトの仕組み

サポートされているリソースに対してゾーンシフトを開始すると、そのリソースのトラフィックは、指定したアベイラビリティーゾーン (AZ) から切り替えられます。ARC がサポートするリソースは、指定した AZ を異常としてマークするための連携機能を備えており、その結果として、障害のある AZ からトラフィックが切り替えられます。

トラフィックの切り替え開始 - ARC でゾーンシフトを開始しても、アベイラビリティーゾーンからすぐにトラフィックが移動しない場合があります。クライアントの挙動または接続の再利用によっては、アベイラビリティーゾーンで進行中の既存の接続が完了するまでに、若干時間がかかる場合もあります。DNS 設定や既存の接続を含むその他の要因は数分で完了しますが、時間がかかる場合があります。詳細については、「トラフィックのシフトを迅速に完了させる」を参照してください。

トラフィックシフトの終了 - ゾーンシフトの有効期限が切れるか、取り消した場合、ARC はトラフィックの切り替えを停止するための処理を行い、トラフィック切り替えを開始したときの手順を逆に実行します。これで、復旧した AZ がリソースで使用可能として認識され、AZ へのトラフィックが再開されます。

ゾーンシフトを開始するときは、すべてのゾーンシフトに有効期限を設定する必要があります。ゾーンシフトの有効期限は、初回は最大で 3 日 (72 時間) 後に設定できます。ただし、ゾーンシフトはいつでも新しい有効期限に更新できます。アベイラビリティーゾーンへのトラフィックを復旧する準備ができていたら、有効期限が切れる前にゾーンシフトをキャンセルすることも可能です。

トラフィックが切り替わらない場合 - 特定の状況では、ゾーンシフトを行ってもアベイラビリティーゾーンからトラフィックが切り替えられないことがあります。例えば、ロードバランサーのゾーンシ

フトを開始した時点で、対象の AZ にあるロードバランサーのターゲットグループにインスタンスが存在しない場合や、すべてのインスタンスが異常である場合などが該当します。この場合、ロードバランサーはフェイルオープン状態となり、ゾーンシフトを開始してもトラフィックは移動しません。

リソースのゾーンシフトを開始する前に、成功したゾーンシフトのすべての条件が満たされていることを確認してください。 AWS リソースは、ゾーンシフトを異なる方法で処理します。ゾーンシフトのサポートに関する詳細は、「サポートされているリソース」を参照してください。

AWS リージョン ゾーンシフトの可用性

Amazon Application Recovery Controller (ARC) のリージョンサポートとサービスエンドポイントに関する詳細は、「Amazon Web Services 全般のリファレンス」にある「Amazon Application Recovery Controller (ARC) エンドポイントとクォータ」を参照してください。

ゾーンシフトとゾーンオートシフトは現在、ここ AWS リージョン に記載されている で利用できます。ゾーンシフトおよびゾーンオートシフトは、中国 (北京) リージョンおよび中国 (寧夏) リージョンでも利用できます。Amazon Application Recovery Controller (ARC) を使用するリソースには、追加で考慮すべき点がある場合があります。詳細については、サポートされているリソース を参照してください。

リージョン名	リージョン	エンドポイント	プロトコル
米国東部 (オハイオ)	us-east-2	arc-zonal-shift.us-east-2.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-east-2.api.aws	HTTPS
		arc-zonal-shift.us-east-2.api.aws	HTTPS
米国東部 (バージニア北部)	us-east-1	arc-zonal-shift.us-east-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-east-1.api.aws	HTTPS
		arc-zonal-shift.us-east-1.api.aws	HTTPS
米国西部 (北カリフォルニア)	us-west-1	arc-zonal-shift.us-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-west-1.api.aws	HTTPS
		arc-zonal-shift.us-west-1.api.aws	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
米国西部 (オレゴン)	us-west-2	arc-zonal-shift.us-west-2.amazonaws.com arc-zonal-shift-fips.us-west-2.api.aws arc-zonal-shift.us-west-2.api.aws	HTTPS HTTPS HTTPS
アフリカ (ケープタウン)	af-south-1	arc-zonal-shift.af-south-1.amazonaws.com arc-zonal-shift.af-south-1.api.aws	HTTPS HTTPS
アジアパシフィック (香港)	ap-east-1	arc-zonal-shift.ap-east-1.amazonaws.com arc-zonal-shift.ap-east-1.api.aws	HTTPS HTTPS
アジアパシフィック (ハイデラバード)	ap-south-2	arc-zonal-shift.ap-south-2.amazonaws.com arc-zonal-shift.ap-south-2.api.aws	HTTPS HTTPS
アジアパシフィック (ジャカルタ)	ap-southeast-3	arc-zonal-shift.ap-southeast-3.amazonaws.com arc-zonal-shift.ap-southeast-3.api.aws	HTTPS HTTPS
アジアパシフィック (マレーシア)	ap-southeast-5	arc-zonal-shift.ap-southeast-5.amazonaws.com arc-zonal-shift.ap-southeast-5.api.aws	HTTPS HTTPS
アジアパシフィック (メルボルン)	ap-southeast-4	arc-zonal-shift.ap-southeast-4.amazonaws.com arc-zonal-shift.ap-southeast-4.api.aws	HTTPS HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
アジアパシフィック (ムンバイ)	ap-south-1	arc-zonal-shift.ap-south-1.amazonaws.com arc-zonal-shift.ap-south-1.api.aws	HTTPS HTTPS
アジアパシフィック (ニュージーランド)	ap-southeast-6	arc-zonal-shift.ap-southeast-6.amazonaws.com arc-zonal-shift.ap-southeast-6.api.aws	HTTPS HTTPS
アジアパシフィック (大阪)	ap-northeast-3	arc-zonal-shift.ap-northeast-3.amazonaws.com arc-zonal-shift.ap-northeast-3.api.aws	HTTPS HTTPS
アジアパシフィック (ソウル)	ap-northeast-2	arc-zonal-shift.ap-northeast-2.amazonaws.com arc-zonal-shift.ap-northeast-2.api.aws	HTTPS HTTPS
アジアパシフィック (シンガポール)	ap-southeast-1	arc-zonal-shift.ap-southeast-1.amazonaws.com arc-zonal-shift.ap-southeast-1.api.aws	HTTPS HTTPS
アジアパシフィック (シドニー)	ap-southeast-2	arc-zonal-shift.ap-southeast-2.amazonaws.com arc-zonal-shift.ap-southeast-2.api.aws	HTTPS HTTPS
アジアパシフィック (台北)	ap-east-2	arc-zonal-shift.ap-east-2.amazonaws.com arc-zonal-shift.ap-east-2.api.aws	HTTPS HTTPS

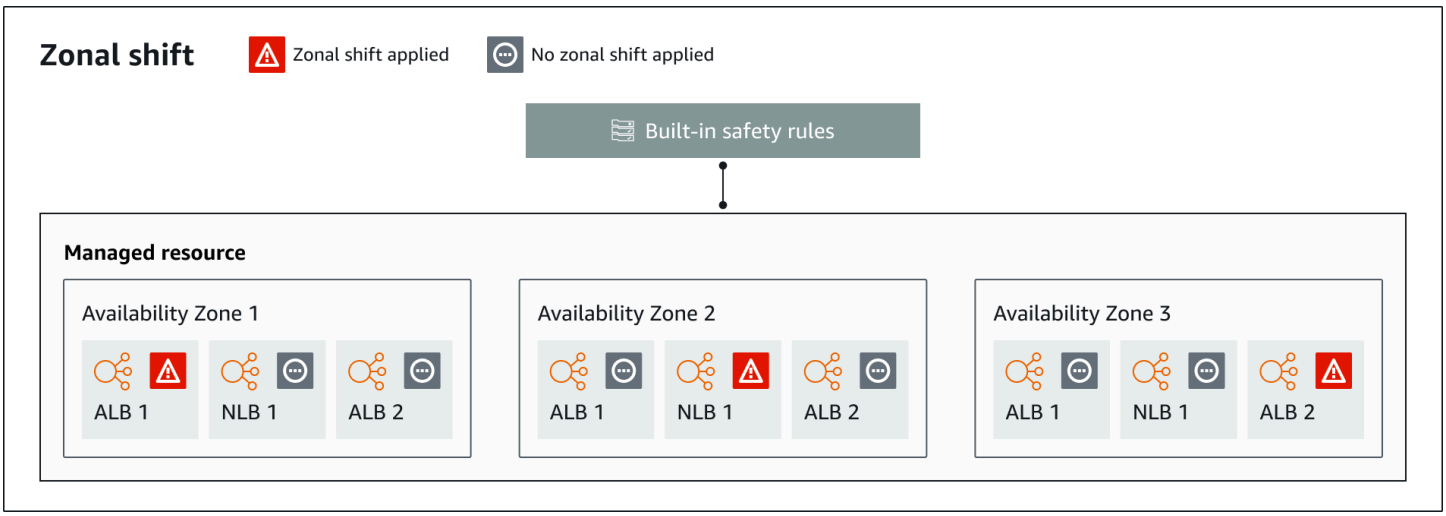
リージョン名	リージョン	エンドポイント	プロトコル
アジアパシフィック (タイ)	ap-southeast-7	arc-zonal-shift.ap-southeast-7.amazonaws.com	HTTPS
		arc-zonal-shift.ap-southeast-7.api.aws	HTTPS
アジアパシフィック (東京)	ap-northeast-1	arc-zonal-shift.ap-northeast-1.amazonaws.com	HTTPS
		arc-zonal-shift.ap-northeast-1.api.aws	HTTPS
カナダ (中部)	ca-central-1	arc-zonal-shift.ca-central-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-central-1.api.aws	HTTPS
		arc-zonal-shift.ca-central-1.api.aws	HTTPS
カナダ西部 (カルガリー)	ca-west-1	arc-zonal-shift.ca-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-west-1.api.aws	HTTPS
		arc-zonal-shift.ca-west-1.api.aws	HTTPS
欧州 (フランクフルト)	eu-central-1	arc-zonal-shift.eu-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-central-1.api.aws	HTTPS
欧州 (アイルランド)	eu-west-1	arc-zonal-shift.eu-west-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-1.api.aws	HTTPS
欧州 (ロンドン)	eu-west-2	arc-zonal-shift.eu-west-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-2.api.aws	HTTPS
ヨーロッパ (ミラノ)	eu-south-1	arc-zonal-shift.eu-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-south-1.api.aws	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
欧州 (パリ)	eu-west-3	arc-zonal-shift.eu-west-3.amazonaws.com	HTTPS
		arc-zonal-shift.eu-west-3.api.aws	HTTPS
欧州 (スペイン)	eu-south-2	arc-zonal-shift.eu-south-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-south-2.api.aws	HTTPS
欧州 (ストックホルム)	eu-north-1	arc-zonal-shift.eu-north-1.amazonaws.com	HTTPS
		arc-zonal-shift.eu-north-1.api.aws	HTTPS
欧州 (チューリッヒ)	eu-central-2	arc-zonal-shift.eu-central-2.amazonaws.com	HTTPS
		arc-zonal-shift.eu-central-2.api.aws	HTTPS
イスラエル (テルアビブ)	il-central-1	arc-zonal-shift.il-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.il-central-1.api.aws	HTTPS
メキシコ (中部)	mx-central-1	arc-zonal-shift.mx-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.mx-central-1.api.aws	HTTPS
中東 (バーレーン)	me-south-1	arc-zonal-shift.me-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.me-south-1.api.aws	HTTPS
中東 (アラブ首長国連邦)	me-central-1	arc-zonal-shift.me-central-1.amazonaws.com	HTTPS
		arc-zonal-shift.me-central-1.api.aws	HTTPS
南米 (サンパウロ)	sa-east-1	arc-zonal-shift.sa-east-1.amazonaws.com	HTTPS
		arc-zonal-shift.sa-east-1.api.aws	HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
AWS GovCloud (米国東部)	us-gov-east-1	arc-zonal-shift.us-gov-east-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-gov-east-1.api.aws	HTTPS
		arc-zonal-shift.us-gov-east-1.api.aws	HTTPS
AWS GovCloud (米国西部)	us-gov-west-1	arc-zonal-shift.us-gov-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.us-gov-west-1.api.aws	HTTPS
		arc-zonal-shift.us-gov-west-1.api.aws	HTTPS

ゾーンシフトのコンポーネント

次の図は、トラフィックを AWS リージョンのアベイラビリティーゾーンから遠ざけるゾーンシフトの例を示しています。ゾーンシフトに組み込まれているチェック機能により、リソースに既にアクティブなゾーンシフトがある場合は、別のゾーンシフトを開始できないようになっています。



以下は、ARC におけるゾーンシフト機能のコンポーネントです。

ゾーンシフト

AWS アカウントのマネージドリソースのゾーンシフトを開始して AWS リージョン、トラフィックを のアベイラビリティーゾーンからリージョン内の正常な AZs に一時的に移動し、1 つの AZ

の問題から迅速に復旧します。ゾーンシフトでサポートされているリソースの詳細については、「サポートされているリソース」を参照してください。

組み込みの安全チェック

ARC に組み込まれているチェックにより、1 つのリソースについて一度に複数のトラフィックシフトが行われることはありません。つまり、アベイラビリティーゾーンからトラフィックをアクティブにシフトできるのは、1 件のユーザー起動のゾーンシフト、練習用実行、またはオートシフトのいずれか 1 つだけです。例えば、あるリソースがオートシフトで遠ざけられているときにゾーンシフトを開始した場合は、ゾーンシフトが優先されます。詳細については、「ARC のゾーンオートシフト」と「練習実行の結果」を参照してください。

リソース識別子

ゾーンシフトに含めるリソースの識別子です。識別子は、リソースの Amazon リソースネーム (ARN) です。

ゾーンシフトでは、ARC でサポートされている AWS サービスのアカウント内のリソースのみを選択できます。ゾーンシフトでサポートされているリソースの詳細については、「サポートされているリソース」を参照してください。

マネージドリソース

一部の AWS リソースはゾーンシフトに手動でオプトインする必要があり、他のリソースは自動的に有効になります。ゾーンシフトでサポートされているリソースの詳細については、「サポートされているリソース」を参照してください。

リソース名

ゾーンシフトに指定できる ARC のリソースの名前。

ステータス (ゾーンシフトステータス)

ゾーンシフトのステータスです。ゾーンシフトの `Status` には、次のいずれかの値が設定されます。

- ACTIVE (アクティブ): ゾーンシフトが開始され、アクティブの状態です。
- EXPIRED (期限切れ): ゾーンシフトが期限切れの状態です (有効期限を超過)。
- CANCELED (キャンセル): ゾーンシフトがキャンセルされた状態です。

適用ステータス

適用ステータスは、リソースに対してシフトが適用されているかどうかを示します。ステータスが `APPLIED` のシフトによって、リソースについてアプリケーショントラフィックが遠ざけられたアベイラビリティーゾーンと、そのトラフィックシフトが終了するタイミングが決まります。

シフトタイプ

ゾーンシフトタイプを定義します。shiftType は以下のイベント値を持ちます。

- ZONAL_SHIFT
- ZONAL_AUTOSHIFT
- PRACTICE_RUN
- FIS_EXPERIMENT

有効期限 (満了期限)

ゾーンシフトの有効期限 (満了期限) です。ゾーンシフトは一時的なものです。ゾーンシフトは、最初は 3 日間 (72 時間) 有効になるように設定できます。

ゾーンシフトを開始するときに、ゾーンシフトをアクティブにする時間を指定します。ARC によってそれが有効期限 (満了期限) に変換されます。例えば、アベイラビリティーゾーンへのトラフィックをリカバリする準備ができている場合は、ゾーンシフトをキャンセルできます。または、顧客が開始したゾーンシフトを更新して別の有効期限を指定することによって、ゾーンシフトを延長することもできます。

ゾーンオートシフトの一部であるゾーンシフトの練習実行をキャンセルできます。

ゾーンシフトのデータプレーンとコントロールプレーン

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは、可用性が高く、災害シナリオで必要なときに使用できるようにすることをお勧めします。信頼性と耐障害性を最大限に高めるため、可能であれば通常、データプレーン機能をメカニズムに使用する必要があります。そのことを念頭に置いて、サービス機能がコントロールプレーンとデータプレーンにどのように分けられているのか、また、サービスのデータプレーンで非常に高い信頼性が期待できるのはどのような場合なのかを理解することが重要です。

ほとんどの AWS サービスと同様に、ゾーンシフト機能の機能はコントロールプレーンとデータプレーンでサポートされています。どちらのタイプも信頼できるように構築されていますが、コントロールプレーンはデータ整合性のために最適化され、データプレーンは可用性のために最適化されています。データプレーンは、コントロールプレーンが使用できなくなるような破壊的なイベントでも、可用性を維持できるように設計されています。

一般に、コントロールプレーンを使用すると、サービス内のリソースの作成、更新、削除などの基本的な管理機能を実行できます。データプレーンはサービスのコア機能を提供します。

データプレーン、コントロールプレーン、および が高可用性目標を達成するためのサービス AWS を構築する方法の詳細については、Amazon Builders' Library の「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。

ARC のゾーンシフトの料金

ゾーンシフトでは、サポートされているリソースに対してゾーンシフトを開始し、アベイラビリティーゾーンで発生した問題からアプリケーションを復旧させることができます。ゾーンシフトは追加料金なしで使用できます。

ARC の料金および料金例の詳細については、「ARC の料金」を参照してください。

ARC のゾーンシフトのベストプラクティス

ARC で、マルチ AZ リカバリにゾーンシフトを使用する場合は、次のベストプラクティスに従うことが推奨されます。

トピック

- キャパシティプランニングと事前スケーリング
- クライアントがエンドポイントに接続したままになる時間を制限する
- ゾーンシフトの開始を事前にテストする
- すべてのアベイラビリティーゾーンが正常で、トラフィックを取るようにする
- ディザスタリカバリにデータプレーン API オペレーションを使用する
- ゾーンシフトでトラフィックを一時的にのみ移動する

キャパシティプランニングと事前スケーリング

ゾーンシフトを開始するときは、事前にスケーリングするか、自動スケーリングができるようにキャパシティを計画することで、アベイラビリティーゾーンにかかる通常よりも大きな負荷に対応できるようにしておきます。リカバリに重点が置かれたアーキテクチャでは一般的に、(通常) 3 つのレプリカのうちいずれかがオフラインになったとき、ピーク時のトラフィックに対応できるだけの十分なヘッドルームを確保するように、コンピューティングキャパシティを事前にスケールすることが推奨されています。

サポートされているリソースに対してゾーンシフトを開始し、トラフィックが AZ から切り替えられると、アプリケーションがリクエストの処理に使用していたキャパシティは削除されます。AZ からのトラフィック切り替えを事前に計画し、残りの AZ でもリクエストの処理を継続できることを確認する必要があります。

クライアントがエンドポイントに接続したままになる時間を制限する

Amazon Application Recovery Controller (ARC) がゾーンシフトやゾーンオートシフトなどを使用してトラフィックを障害から遠ざける場合、ARC がアプリケーショントラフィックを移動するために使用するメカニズムは DNS 更新です。DNS 更新により、すべての新しい接続が障害のある場所から遠ざけられます。

ただし、既存のオープン接続を持つクライアントは、クライアントが再接続するまで、障害が発生したロケーションに対してリクエストを引き続き行う場合があります。迅速な復旧を確保するために、クライアントがエンドポイントに接続し続ける時間を制限することをお勧めします。

ゾーンシフトの開始を事前にテストする

ゾーンシフトを開始してトラフィックをアプリケーションのアベイラビリティーゾーンから移動するテストを、定期的に行います。ゾーンシフトを計画して、障害の発生時にアプリケーションをリカバリするフェイルオーバーの定期テストの一環として、できればテストと本番の両方の環境でその開始を実行します。定期テストは、運用上のイベントに備え、イベントの発生時には自信を持って緩和できるようにするために不可欠なものです。

すべてのアベイラビリティーゾーンが正常で、トラフィックを取るようにする

ゾーンシフトは、アベイラビリティーゾーン内でリソース、つまりアプリケーションレプリカを異常とマークすることにより機能します。つまり、アプリケーション内のリソースが概ね正常であり、リージョンの各アベイラビリティーゾーンでトラフィックを確実に処理していることを確認することが非常に重要です。それを追跡するには、ダッシュボードを使うのがお勧めです。ダッシュボードでは、異常のあるターゲットの Elastic Load Balancing メトリクスや、アベイラビリティーゾーン別の bytesProcessed などを確認できます。

リソースの状態のモニタリングは、2 つ目の隣接するリージョンから行うことを検討します。このアプローチにより、エンドユーザーの体験をより多く反映させることができ、さらに、アプリケーションとモニタリングの両方が同じ災害から同時に影響を受けるリスクを下げることができます。

ディザスタリカバリにデータプレーン API オペレーションを使用する

アプリケーションをほとんど依存関係なく迅速に復旧する必要がある場合にゾーンシフトを開始するには、可能であれば、 AWS Command Line Interface または API をゾーンシフトアクションとともに、認証情報を事前に保存しておくことをお勧めします。使いやすさを考慮して AWS マネジメントコンソール、 でゾーンシフトを開始することもできます。ただし、スピーディな、信頼性の高いリカバリがカギとなるケースでは、データプレーンオペレーションの方が適しています。詳細については、「Zonal Shift API Reference Guide」を参照してください。

ゾーンシフトでトラフィックを一時的にのみ移動する

ゾーンシフトは、障害を緩和するためにトラフィックをアベイラビリティーゾーンから一時的に移動する機能です。問題を解決するためのアクションを実行したら、すぐにアプリケーションのリソースをサービスに戻す必要があります。そうすることで、アプリケーション全体が、完全な冗長性とレジリエンスを備えた元の状態に戻ります。

ゾーンシフト API オペレーション

次の表は、マルチ AZ アプリケーションについてアベイラビリティーゾーンからトラフィックを遠ざけるゾーンシフトで使用できる ARC API オペレーションを一覧にしたものです。この表には、関連ドキュメントへのリンクも含まれています。

AWS Command Line Interfaceで一般的なゾーンシフト API オペレーションを使用する方法の例については、「ゾーンシフト AWS CLI で を使用する例」を参照してください。

アクション	ARC コンソールの使用	ARC API の使用
ゾーンシフトを開始する	「ゾーンシフトの開始」を参照してください。	「StartZonalShift」を参照
ゾーンシフトの更新	「ゾーンシフトの更新またはキャンセル」を参照してください。	「UpdateZonalShift」を参照
ゾーンシフトを一覧表示する	「ARC のゾーンシフト」を参照してください。	「ListZonalShifts」を参照
マネージドリソースを一覧表示する	「サポートされているリソース」を参照してください。	「ListManagedResources」を参照
マネージドリソースを取得する	「サポートされているリソース」を参照してください。	「GetManagedResource」を参照
ゾーンシフトのキャンセル	「ゾーンシフトの更新またはキャンセル」を参照してください。	「CancelZonalShift」を参照

ゾーンシフト AWS CLI で を使用する例

このセクションでは、 を使用して、API オペレーションを使用して Amazon Application Recovery Controller (ARC) のゾーンシフト機能 AWS Command Line Interface を操作する、ゾーンシフトを使用するアプリケーションの例を示します。これらの例は、CLI を使用したゾーンシフトの操作方法の基本的な理解を深めることを目的としています。

ARC のゾーンシフトを使用すると、サポートされているリソースのトラフィックを特定のアベイラビリティーゾーンから一時的に切り替え、アプリケーションが AWS リージョンの他のアベイラビリティーゾーンで正常に動作し続けることができます。

すべてのゾーンシフトは一時的なもので、最初は 3 日以内に期限切れになるように設定する必要があります。ただし、後でゾーンシフトを更新して新しい有効期限を設定できます。

の使用の詳細については AWS CLI、AWS CLI 「 コマンドリファレンス」を参照してください。ゾーンシフト API アクションのリストと詳細情報へのリンクについては、「ゾーンシフト API オペレーション」を参照してください。

ゾーンシフトを開始する

CLI で `start-zonal-shift` コマンドを使用して、ゾーンシフトを開始できます。

```
aws arc-zonal-shift start-zonal-shift \
       --resource-identifier arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05 \
       --away-from use1-az1 \
       --expires-in 10m \
       --comment "Shifting traffic away from use1-az1"
```

```
{
    "awayFrom": "use1-az1",
    "comment": "Shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T21:37:26-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "ACTIVE",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

マネージドリソースを取得する

マネージドリソースに関する情報は、CLI で get-managed-resource コマンドを使用して取得できます。

```
aws arc-zonal-shift get-managed-resource \
       --resource-identifier arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05
```

```
{
    "appliedWeights": {
        "use1-az1": 0.0,
        "use1-az2": 1.0,
        "use1-az6": 1.0
    },
    "arn": "arn:aws:elasticloadbalancing:us-east-1:111122223333:loadbalancer/app/
Testing/5a19403ecd42dc05",
    "autoshifts": [],
    "name": "Testing",
    "zonalAutoshiftStatus": "DISABLED",
    "zonalShifts": [
        {
            "appliedStatus": "APPLIED",
            "awayFrom": "use1-az1",
            "comment": "Shifting traffic away from use1-az1",
            "expiryTime": "2024-12-17T21:37:26-08:00",
            "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
            "startTime": "2024-12-17T21:27:26-08:00",
            "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
            "shiftType": "MANUAL"
        }
    ]
}
```

マネージドリソースを一覧表示する

CLI で list-managed-resources コマンドを使用して、アカウント内のマネージドリソースを一覧表示できます。

```
aws arc-zonal-shift list-managed-resources
```

```
{
    "items": [
        {
            "appliedWeights": {
                "use1-az1": 0.0,
                "use1-az2": 1.0,
                "use1-az6": 1.0
            },
            "arn": "arn:aws:elasticloadbalancing:us-east-1:111122223333:loadbalancer/
app/Testing/5a19403ecd42dc05",
            "autoshifts": [],
            "availabilityZones": [
                "use1-az1",
                "use1-az2",
                "use1-az6"
            ],
            "name": "Testing",
            "practiceRunStatus": "DISABLED",
            "zonalAutoshiftStatus": "DISABLED",
            "zonalShifts": [
                {
                    "appliedStatus": "APPLIED",
                    "awayFrom": "use1-az1",
                    "comment": "Shifting traffic away from use1-az1",
                    "expiryTime": "2024-12-17T21:37:26-08:00",
                    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
                    "startTime": "2024-12-17T21:27:26-08:00",
                    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
                }
            ]
        }
    ]
}
```

ゾーンシフトを一覧表示する

CLI で `list-zonal-shifts` コマンドを使用して、アカウント内のゾーンシフトを一覧表示できます。

```
aws arc-zonal-shift list-zonal-shifts
```

```
{
    "items": [
        {
            "awayFrom": "use1-az1",
            "comment": "Shifting traffic away from use1-az1",
            "expiryTime": "2024-12-17T21:37:26-08:00",
            "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
            "startTime": "2024-12-17T21:27:26-08:00",
            "status": "ACTIVE",
            "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
        }
    ]
}
```


ゾーンシフトを更新する

CLI で update-zonal-shift コマンドを使用して、ゾーンシフトを更新できます。

```
aws arc-zonal-shift update-zonal-shift \
       --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38 \
       --expires-in 1h \
       --comment "Still shifting traffic away from use1-az1"
```


```
{
    "awayFrom": "use1-az1",
    "comment": "Still shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "ACTIVE",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```


ゾーンシフトをキャンセルする

CLI で cancel-zonal-shift コマンドを使用して、ゾーンシフトをキャンセルできます。

```
aws arc-zonal-shift cancel-zonal-shift \
       --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38
```

```
{
    "awayFrom": "use1-az1",
    "comment": "Still shifting traffic away from use1-az1",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "CANCELED",
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```


サポートされているリソース

Amazon Application Recovery Controller (ARC) は現在、ゾーンシフトとゾーンオートシフトで次のリソースの有効化をサポートしています。

- Amazon EC2 Auto Scaling グループ
- アマゾン エラスティックKubernetesサービス
- クロスゾーンのロードバランサーの有効/無効を問わない アプリケーション ロード バランサー
- クロスゾーンのロードバランサーの有効/無効を問わない Network Load Balancer

Network Load Balancer と Application Load Balancer の具体的な要件については、このセクションの追加トピックを参照してください。

ARC でゾーンシフト、ゾーンオートシフト、リソースを使用する場合は、以下の条件を確認してください。

- トラフィックをそのリソースにシフトするには、リソースがアクティブになっており、正常にプロビジョニングされている必要があります。リソースのゾーンシフトを開始するときは、事前に、そのリソースが ARC のマネージドリソースであることを確認します。たとえば、 でマネージドリソースのリストを表示するか AWS マネジメントコンソール、リソースの識別子を指定して `get-managed-resource`オペレーションを使用します。
- リソースを使用してゾーンシフトを開始するときは、ゾーンシフトを開始するアベイラビリティーゾーンと AWS リージョン に、リソースをデプロイする必要があります。ゾーンシフトが、シフトの対象となる AZ と同じリージョンで開始すること、および、トラフィックのシフト先となるリソースも、同じ AZ とリージョンにあることを確認します。
- リソースでゾーンシフトを実行するときは、適切な IAM アクセス許可があることを確認します。詳細については、「ゾーンシフトの IAM とアクセス許可」を参照してください。

- Network Load Balancer または Application Load Balancer がフェイルオープン状態の場合、ゾーンシフトは効果がありません。これは想定された動作です。なぜなら、ロードバランサーがフェイルオープン状態のときに、ゾーンシフトが AZ を強制的に異常状態にして、リージョンの他の AZ にトラフィックを切り替えることはできないためです。詳細については、「Network Load Balancer ユーザーガイド」の「ロードバランサーの Route 53 DNS フェイルオーバーを使用する」および「Application Load Balancer ユーザーガイド」の「ロードバランサーの Route 53 DNS フェイルオーバーを使用する」を参照してください。
- 複数のロードバランサーが同じターゲットにトラフィックを転送する場合、クロスゾーン対応ロードバランサーのゾーンシフトは、トラフィックがゾーンシフトによってシフトされない場合でも、すべてのロードバランサーのターゲット容量を減らします。

Amazon EC2 Auto Scaling グループ

Amazon EC2 Auto Scaling グループには、オートスケーリングと管理のための、論理グループとして扱われる Amazon EC2 インスタンスの集合が含まれています。また、Auto Scaling グループによって、ヘルスチェックの置き換えやスケーリングポリシーなど、Amazon EC2 Auto Scaling の機能も使用できます。Auto Scaling グループでのインスタンス数の維持と自動スケーリングの両方が Amazon EC2 Auto Scaling サービスの主な機能です。

Auto Scaling グループのゾーンシフトの使用

ゾーンシフトを有効にするには、次のいずれかの方法を使用します。

Console

新しいグループでゾーンシフトを有効にするには (コンソール)

1. 「起動テンプレートを使用して Auto Scaling グループを作成する」の手順に従って、ステップ 10 までの手順の各ステップを完了します。
2. [他のサービスと統合する]ページの [ARC ゾーンシフト] で、チェックボックスを選択してゾーンシフトを有効にします。
3. [ヘルスチェックの動作] で、[異常を無視] または [異常なものとの置換を実行] を選択します。`replace-unhealthy` に設定すると、異常なインスタンスはアベイラビリティーゾーンのアクティブなゾーンシフトに置き換えられます。`ignore-unhealthy` に設定すると、異常なインスタンスはアクティブなゾーンシフトのあるアベイラビリティーゾーンで置き換えられません。
4. 「起動テンプレートを使用して Auto Scaling グループを作成する」の手順に進みます。

AWS CLI

新しいグループでゾーンシフトを有効にするには (AWS CLI)

create-auto-scaling-group コマンドに `--availability-zone-impairment-policy` パラメータを追加します。

`--availability-zone-impairment-policy` パラメータには次の 2 つのオプションがあります。

- ZonalShiftEnabled – `true` に設定すると、Auto Scaling は Auto Scaling グループを ARC ゾーンシフトに登録し、ARC コンソールでゾーンシフトを開始、更新、またはキャンセルできます。`false` に設定すると、Auto Scaling は Auto Scaling グループを ARC ゾーンシフトから登録解除します。`false` に設定するには、ゾーンシフトが既に有効になっている必要があります。
- ImpairedZoneHealthCheckBehavior – `replace-unhealthy` に設定すると、異常なインスタンスはアクティブなゾーンシフトのあるアベイラビリティーゾーンで置き換えられます。`ignore-unhealthy` に設定すると、異常なインスタンスはアクティブなゾーンシフトのあるアベイラビリティーゾーンで置き換えられません。

次の例では、*my-asg* という名前の新しい Auto Scaling グループでゾーンシフトを有効にします。

```
aws autoscaling create-auto-scaling-group \
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \
  --auto-scaling-group-name my-asg \
  --min-size 1 \
  --max-size 10 \
  --desired-capacity 5 \
  --availability-zones us-east-1a us-east-1b us-east-1c \
  --availability-zone-impairment-policy '{
      "ZonalShiftEnabled": true,
      "ImpairedZoneHealthCheckBehavior": IgnoreUnhealthy
    }'
```

Console

既存のグループでゾーンシフトを有効にするには (コンソール)

1. https://console.aws.amazon.com/ec2/ でAmazon EC2 コンソールを開き、ナビゲーションペインで [Auto Scaling グループ] を選択します。
2. 画面上部のナビゲーションバーで、Auto Scaling グループ AWS リージョン を作成した を選択します。
3. Auto Scaling グループの横にあるチェックボックスをオンにします。

   ページの下部にスプリットペインが開きます。
4. [統合] タブの [ARC ゾーンシフト] で、[編集] を選択します。
5. ゾーンシフトを有効にするには、チェックボックスをオンにします。
6. [ヘルスチェックの動作] で、[異常を無視] または [異常なものとの置換を実行] を選択します。

   - ヘルスチェックの動作が異常を無視するように設定されている場合、異常のあるインスタンスはアベイラビリティーゾーンでアクティブなゾーンシフトに置き換えられません。
   - ヘルスチェックの動作が異常を置き換えるように設定されている場合、異常のあるインスタンスはアベイラビリティーゾーンでアクティブなゾーンシフトに置き換えられます。
7. [更新] を選択します。

AWS CLI

既存のグループでゾーンシフトを有効にするには (AWS CLI)

update-auto-scaling-group コマンドに `--availability-zone-impairment-policy` パラメータを追加します。

`--availability-zone-impairment-policy` パラメータには次の 2 つのオプションがあります。

- ZonalShiftEnabled – TRUE に設定すると、Auto Scaling は Auto Scaling グループを ARC ゾーンシフトに登録し、ARC コンソールでゾーンシフトを開始、更新、またはキャンセルできます。FALSE に設定すると、Auto Scaling は Auto Scaling グループを ARC ゾーンシフトから登録解除します。FALSE に設定するには、ゾーンシフトが既に有効になっている必要があります。

- ImpairedZoneHealthCheckBehavior – `replace-unhealthy` に設定すると、異常なインスタンスはアクティブなゾーンシフトのあるアベイラビリティーゾーンで置き換えられます。`ignore-unhealthy` に設定すると、異常なインスタンスはアクティブなゾーンシフトのあるアベイラビリティーゾーンで置き換えられません。

次の例では、指定された Auto Scaling グループでゾーンシフトを有効にします。

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \
  --availability-zone-impairment-policy '{
      "ZonalShiftEnabled": true,
      "ImpairedZoneHealthCheckBehavior": IgnoreUnhealthy
    }'
```

ゾーンシフトを開始するには、「ゾーンシフトの開始、更新またはキャンセル」を参照してください。

Auto Scaling グループに対するゾーンシフトの仕組み

次のアベイラビリティーゾーンを含む Auto Scaling グループがあるとします。

- `us-east-1a`
- `us-east-1b`
- `us-east-1c`

`us-east-1a` の障害に気づき、ゾーンシフトを開始します。`us-east-1a` でゾーンシフトが開始されると、次の動作が発生します。

- スケールアウト – Auto Scaling は、正常なアベイラビリティーゾーン (`us-east-1b` および) ですべての新しいキャパシティーリクエストを起動しますus-east-1c。
- 動的スケーリング – Auto Scaling は、スケーリングポリシーが希望する容量を減らすのをブロックします。Auto Scaling は、スケーリングポリシーが希望する容量を増やすのをブロックしません。
- インスタンスの更新 – Auto Scaling は、アクティブなゾーンシフト中に遅延したインスタンスの更新プロセスのタイムアウトを延長します。

障害があるアベイラビリティーゾーンのヘルスチェック動作選択	ヘルスチェックの動作
Replace unhealthy	Instances that appear unhealthy will be replaced in all Availability Zones (`us-east-1a` , `us-east-1b` , and `us-east-1c`).
Ignore unhealthy	Instances that appear unhealthy will be replaced in `us-east-1b` and `us-east-1c` . Instances are not replaced in the Availability Zone with the active zonal shift (`us-east-1a`).

ゾーンシフトの使用に関するベストプラクティス

ゾーンシフトの使用時にアプリケーションの高可用性を維持するには、次のベストプラクティスをお勧めします。

- EventBridge 通知をモニタリングして、進行中のアベイラビリティーゾーンの障害イベントが発生するタイミングを特定します。詳細については、EventBridge による Amazon EC2 Auto Scaling の自動化」を参照してください。
- 適切なしきい値を持つスケーリングポリシーを使用して、アベイラビリティーゾーンの損失を許容できる十分な容量があることを確認します。
- 最小正常パーセンテージが 100 のインスタンスメンテナンスポリシーを設定します。この設定で、Auto Scaling は、異常なインスタンスを終了する前に、新しいインスタンスを使用する準備が整うのを待ちます。

事前スケーリングされたお客様には、以下もお勧めします。

- 障害イベント中に異常なインスタンスを置き換える必要がないため、障害のあるアベイラビリティーゾーンのヘルスチェック動作として [異常を無視] を選択します。
- Auto Scaling グループに対して ARC のゾーンオートシフトを使用します。のゾーンオートシフト機能 Amazon Application Recovery Controller (ARC) を使用すると AWS 、 がアベイラビリティーゾーンの障害 AWS を検出したときに、リソースのトラフィックをアベイラビリティーゾーンか

ら遠ざけることができます。詳細については、「ARC のゾーンオートシフト」を参照してください。

クロスゾーンが無効になっているロードバランサーを使用するお客様には、以下もお勧めします。

- アベイラビリティーゾーンの分散に対して [バランシング (限定)] を使用します。
- Auto Scaling グループとロードバランサーの両方でゾーンシフトを使用している場合は、まず Auto Scaling グループのゾーンシフトをキャンセルしてください。次に、すべてのアベイラビリティーゾーン間で容量のバランスを取れるまで待ってから、ロードバランサーでゾーンシフトをキャンセルします。
- ゾーンシフトを有効にし、クロスゾーン無効ロードバランサーを使用すると、容量が不均衡になる可能性があるため、Auto Scaling には追加の検証があります。ベストプラクティスに従っている場合は、 のチェックボックスを選択する AWS マネジメントコンソール かCreateAutoScalingGroup、、、または の skip-zonal-shift-validationフラグを使用してUpdateAutoScalingGroup、この可能性を確認できますAttachTrafficSources。

アマゾン エラスティックKubernetesサービス

Amazon EKS には、アベイラビリティーゾーンのヘルスの低下や障害などのイベントに対するアプリケーションの回復力を高める機能があります。Amazon EKS クラスターでワークロードを実行する場合、ゾーンシフトまたはゾーンオートシフトを使用して、アプリケーション環境の耐障害性とアプリケーション復旧をさらに改善できます。

Amazon Elastic Kubernetes Service でのゾーンシフトの使用

ゾーンシフトを有効にするには、次のいずれかの方法を使用します。詳細については、「Amazon Elastic Kubernetes Service ユーザーガイド」の「ARC ゾーンシフトについて」を参照してください。

Console

新しい Amazon EKS クラスターでゾーンシフトを有効にするには (コンソール)

1. ARC に登録する Amazon EKS クラスターの名前とリージョンを見つけます。
2. https://console.aws.amazon.com/eks/home#/clusters で Amazon EKS コンソールを開きます。
3. クラスターを選択します。

4. [クラスター情報] ページの [概要] タブを選択します。
5. [ゾーンシフト] で、[管理] を選択します。
6. [EKS ゾーンシフト] で [有効] または [無効] を選択します。

AWS CLI

新しい Amazon EKS クラスターでゾーンシフトを有効にするには (AWS CLI)

- 次のコマンドを入力します。

```
aws eks create-cluster --name my-eks-cluster --role-
arn my-role-arn-to-create-cluster --resources-vpc-config
 subnetIds=string,string,securityGroupIds=string,string,endpointPublicAccess=boolean,end
 --zonal-shift-config enabled=true
```

既存の Amazon EKS クラスターでゾーンシフトを有効にするには (AWS CLI)

- 次のコマンドを入力します。

```
aws eks update-cluster-config --name my-eks-cluster --zonal-shift-config
 enabled=true
```

Amazon EKS クラスターのゾーンシフトを開始することも、ゾーンオートシフトを有効にすることで AWS にゾーンシフトを許可することもできます。ARC で Amazon EKS クラスターのゾーンシフトを有効にしたら、ARC コンソール、 AWS CLI、またはゾーンシフトとゾーンオートシフト APIs を使用して、ゾーンシフトを開始したり、ゾーンオートシフトを有効にしたりできます。

ゾーンシフトの開始に関する詳細については、「ゾーンシフトの開始、更新またはキャンセル」を参照してください。

ゾーンシフトで Amazon EKS を有効にする方法の詳細については、「Amazon Elastic Kubernetes Service ユーザーガイド」の「Amazon EKS での ARC ゾーンシフトについて」を参照してください。

Amazon Elastic Kubernetes Service のゾーンシフトの仕組み

Amazon EKS のゾーンシフト中、次のことが自動的に行われます。

- 影響を受ける AZ 内のすべてのノードが遮断されます。これにより、Kubernetes スケジューラーが異常な AZ のノードに新しいポッドをスケジューリングできなくなります。
- Managed Node Groups を使用している場合、アベイラビリティーゾーンの再調整は中断され、新しい Amazon EKS データプレーンノードが正常な AZs でのみ起動されるように Auto Scaling グループが更新されます。
- 異常な AZ のノードは終了されず、ポッドはこれらのノードから削除されません。これは、ゾーンシフトの有効期限が切れたりキャンセルされたりしたときに、トラフィックがフル容量を保持している AZ に安全に戻ることができるようにするためです。
- EndpointSlice コントローラーは、障害のある AZ 内のすべてのポッドエンドポイントを検索し、関連する EndpointSlices から削除します。これにより、正常な AZ のポッドエンドポイントのみがネットワークトラフィックの受信対象になります。ゾーンシフトがキャンセルまたは期限切れになると、EndpointSlice コントローラーは EndpointSlices を更新して、復元された AZ にエンドポイントを含めます。

詳細については、「AWS Containers ブログ」を参照してください。

アプリケーション ロード バランサー

Application Load Balancer のゾーンシフト

ゾーンシフトで Application Load Balancer を使用するには、Application Load Balancer 属性で ARC ゾーンシフト統合を有効にする必要があります。Application Load Balancer は、クロスゾーンが有効な設定またはクロスゾーンが無効な設定のゾーンシフトをサポートします。

ARC 統合を有効にしてゾーンシフトの使用を開始する前に、次の情報を確認します。

- 1 つのアベイラビリティーゾーンに対してのみ、特定のロードバランサーのゾーンシフトを開始できます。複数のアベイラビリティーゾーンに対してゾーンシフトを開始することはできません。
- AWS は、複数のインフラストラクチャの問題がサービスに影響を与える場合、DNS からゾーンロードバランサーの IP アドレスをプロアクティブに削除します。ゾーンシフトを開始する前に、現在のアベイラビリティーゾーンの容量を必ず確認してください。
- ゾーンシフトは、シングル AZ ターゲットグループでは機能しません。
- Application Load Balancer が Network Load Balancer のターゲットである場合は、常に Network Load Balancer からゾーンシフトを開始します。Application Load Balancer からゾーンシフトを開始すると、Network Load Balancer はシフトを認識せず、引き続き Application Load Balancer にトラフィックを送信します。

ロードバランサーのゾーンシフトは、Elastic Load Balancing コンソール (ほとんどの AWS リージョン) または ARC コンソールで開始できます。

Console

ロードバランサーでゾーンシフトを有効にするには (コンソール)

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. [ナビゲーション] ペインの [ロードバランシング] で [ロードバランサー] を選択します。
3. Application Load Balancer の名前を選択します。
4. [属性] タブで、[編集] を選択します。
5. [アベイラビリティーゾーンルーティング設定] で、[ARC ゾーンシフト統合] を [有効化] に設定します。
6. [保存] を選択します。

AWS CLI

ロードバランサーでゾーンシフトを有効にするには (AWS CLI)

- 次のコマンドを入力します。

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn my-alb-arn --
attributes Key=zonal_shift.config.enabled,Value=true
```

ゾーンシフトの開始に関する詳細については、「ゾーンシフトの開始、更新またはキャンセル」を参照してください。

keepalive オプションを使用して、接続の継続時間を設定できます。詳細については、「Application Load Balancer ユーザーガイド」の「HTTP クライアントのキープアライブ期間」を参照してください。Application Load Balancer では、HTTP クライアントのキープアライブ期間の値はデフォルトで 3600 秒 (1 時間) に設定されます。アプリケーションの目標復旧時間に合わせて、例えば 300 秒のように値を小さくすることをお勧めします。HTTP クライアントのキープアライブ時間を設定する際は、一般的に再接続が増えてレイテンシーに影響が出ることと、障害のある AZ やリージョンからすべてのクライアントをより早く切り替えられることとのトレードオフである点を考慮してください。

Application Load Balancer のゾーンシフトの仕組み

クロスゾーン負荷分散が有効になっている Application Load Balancer でゾーンシフトが開始されると、影響を受けるアベイラビリティーゾーンでターゲットへのすべてのトラフィックがブロックされ、ゾーンシフトによってゾーン IP アドレスが DNS から削除されます。

詳細については、「Application Load Balancer ユーザーガイド」の「Application Load Balancer の統合」を参照してください。

Network Load Balancer

Network Load Balancer でのゾーンシフトの使用

ゾーンシフトで Network Load Balancer を使用するには、Network Load Balancer 属性で ARC ゾーンシフト統合を有効にする必要があります。Network Load Balancer は、クロスゾーンが有効な設定またはクロスゾーンが無効な設定のゾーンシフトをサポートします。

ゾーンシフトとゾーンオートシフトを使用するようにオプトインするリソース、および障害のあるアベイラビリティーゾーンからフェイルアウェイするタイミングを選択できます。インターネット向けと内部両方の Network Load Balancer がサポートされています。

クロスゾーン対応の Network Load Balancer のゾーンシフトを有効にするには、ロードバランサーにアタッチされているすべてのターゲットグループが次の条件を満たしている必要があります。

- クロスゾーン負荷分散が有効になっている (または `use_load_balancer_configuration` に設定されている)。
  - ターゲットグループのクロスゾーン負荷分散の詳細については、「ターゲットグループに対するクロスゾーン負荷分散」を参照してください。
- ターゲットグループプロトコルは TCP または TLS である必要があります。
  - Network Load Balancer ターゲットグループプロトコルの詳細については、「ルーティング設定」を参照してください。
- 異常のあるターゲットの接続終了は無効にする必要があります。
  - ターゲットグループ接続の終了の詳細については、「異常のあるターゲットの接続終了」を参照してください。
- ターゲットグループには、Application Load Balancer をターゲットとして含めないでください。
  - Application Load Balancer をターゲットとして使用する方法の詳細については、「Application Load Balancer を Network Load Balancer のターゲットとして使用する」を参照してください。

Network Load Balancer のゾーンシフトは、 AWS CLI、、 AWS マネジメントコンソールまたは Elastic Load Balancing ウィジェットを使用して開始できます。Application Load Balancer が Network Load Balancer のターゲットである場合は、ゾーンシフトは Network Load Balancer から開始する必要があります。Application Load Balancer からゾーンシフトを開始すると、Network Load Balancer は Application Load Balancer とそのターゲットにトラフィックを送信し続けます。

Console

ロードバランサーでゾーンシフトを有効にするには (コンソール)

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. [ナビゲーション] ペインの [ロードバランシング] で [ロードバランサー] を選択します。
3. Network Load Balancer の名前を選択します。
4. [属性] タブで、[編集] を選択します。
5. [アベイラビリティーゾーンルーティング設定] で、[ARC ゾーンシフト統合] を [有効化] に設定します。
6. [保存] を選択します。

AWS CLI

ロードバランサーでゾーンシフトを有効にするには (AWS CLI)

- 次のコマンドを入力します。

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn my-nlb-arn --
attributes Key=zonal_shift.config.enabled,Value=true
```

ゾーンシフトを開始する方法の詳細については、「ゾーンシフトの開始、更新またはキャンセル」を参照してください。

Network Load Balancer のゾーンシフトの仕組み

ARC は、登録された Network Load Balancer のヘルスチェックの失敗を作成し、ゾーンシフトの開始時に障害が発生した AZ の Network Load Balancer ノードが DNS から削除されるようにします。Network Load Balancer は、影響を受けたゾーンのターゲットを無効にしてトラフィックの受信を停止し、Elastic Load Balancing はこれらのターゲットをゾーンシフトの無効なターゲットとして扱います。無効状態のターゲットは、引き続きヘルスチェックを受信します。ターゲットが正常

で、ゾーンシフトの有効期限が切れる (またはキャンセルされる) と、以前に障害が発生したゾーンのターゲットへのルーティングが再開されます。

クロスゾーン負荷分散が有効になっている Network Load Balancer のゾーンシフト中に、ゾーンロードバランサーの IP アドレスは DNS から削除されます。障害のあるアベイラビリティーゾーン内のターゲットへの既存の接続は、それらが自然に閉じられるまで保持されますが、障害のあるアベイラビリティーゾーン内のターゲットへの新しい接続はルーティングされなくなります。

詳細については、「Network Load Balancer ユーザーガイド」の「Network Load Balancer のゾーンシフト」を参照してください。

ゾーンシフトの開始、更新またはキャンセル

このセクションでは、ゾーンシフトの開始やゾーンシフトのキャンセルなど、ゾーンシフトを操作する手順について説明します。

ゾーンシフトの開始

このセクションでは、Amazon Application Recovery Controller (ARC) のコンソールで顧客開始のゾーンシフトを開始する手順について説明します。ゾーンシフトをプログラムで操作する方法については、「Zonal Shift API Reference Guide」を参照してください。

ARC でゾーンシフトを開始するだけでなく、Elastic Load Balancing コンソール (サポートされているリージョン) でロードバランサーのゾーンシフトを開始することもできます。詳細については、「Elastic Load Balancing ユーザーガイド」の「ゾーンシフト」を参照してください。 Elastic Load Balancing

ゾーンシフトを開始するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [マルチ AZ] で [ゾーンシフト] を選択します。
3. [ゾーンシフト] ページで [ゾーンシフトを開始] を選択します。
4. トラフィックを移動させたいアベイラビリティーゾーンを選択します。
5. リソーステーブルからサポートされているリソースを選択して、トラフィックをシフトします。
6. [ゾーンシフトの有効期限を設定] で、ゾーンシフトの有効期限を選択または入力します。ゾーンシフトは、最初は 1 分 ~ 3 日 (72 時間) まで設定できます。

すべてのゾーンシフトは一時的なものです。有効期限は必ず設定しますが、アクティブなシフトは、後から新しい有効期限 (最大 3 日後) に更新できます。

7. コメントを入力します。必要に応じて、後でゾーンシフトを更新してコメントを編集できます。
8. このチェックボックスをオンにすると、ゾーンシフトを開始した際、トラフィックがアベイラビリティーゾーンからシフトし、アプリケーションの容量が減ることを了承します。
9. [開始] を選択します。

ゾーンシフトの更新またはキャンセル

このセクションでは、Amazon Application Recovery Controller (ARC) のコンソールで、開始するゾーンシフトを更新する方法、またはゾーンシフトをキャンセルする手順について説明します。ゾーンシフトをプログラムで操作する方法については、「Zonal Shift API Reference Guide」を参照してください。

ゾーンシフトは、更新して新しい有効期限を設定できます。また、コメントを編集したり置き換えたりもできます。ゾーンシフトは、有効期限が切れる前であればいつでもキャンセルできます。

開始したゾーンシフト、またはゾーンオートシフトの練習実行のリソースに対して AWS 開始されたゾーンシフトをキャンセルできます。ゾーンオートシフトの練習シフトの詳細については、「ゾーンオートシフトと練習実行の仕組み」を参照してください。

ゾーンシフトを更新するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [マルチ AZ] で [ゾーンシフト] を選択します。
3. 更新するゾーンシフトを選択し、[ゾーンシフトを更新] を選択します。
4. [Set zonal shift expiration time] (ゾーンシフトの有効期限の設定) で、オプションで有効期限を選択または入力します。
5. [Comment] (コメント) には、必要に応じて既存のコメントを編集するか、新しいコメントを入力します。
6. [更新] を選択します。

ゾーンシフトをキャンセルするには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [マルチ AZ] で [ゾーンシフト] を選択します。
3. 更新するゾーンシフトを選択し、[ゾーンシフトをキャンセル] を選択します。
4. ダイアログボックスで、[確認] を選択します。

Amazon Application Recovery Controller (ARC) のゾーンシフトのログ記録とモニタリング

AWS CloudTrail を使用して、Amazon Application Recovery Controller (ARC) のゾーンシフトをモニタリングし、パターンを分析し、問題のトラブルシューティングに役立てることができます。

トピック

- AWS CloudTrail を使用したゾーンシフト API コールのログ記録

AWS CloudTrail を使用したゾーンシフト API コールのログ記録

ARC のゾーンシフトは、AWS CloudTrail と統合されています。これは、ARC のユーザー、ロール、または AWS のサービスで実行されたアクションを記録するためのサービスです。CloudTrail は、ゾーンシフトに対するすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、ARC コンソールからのコールと、ゾーンシフトの ARC API オペレーションへのコードコールが含まれます。

証跡を作成する場合は、ゾーンシフトのイベントを含む、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。

CloudTrail で収集した情報を使用して、ゾーンシフトの ARC に対するリクエスト、リクエスト元の IP アドレス、リクエストした人、リクエストが行われた日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail のゾーンシフト情報

CloudTrail は、AWS アカウントを作成すると、その中で有効になります。ゾーンシフトの ARC でアクティビティが発生すると、そのアクティビティは他の AWS サービスイベントと共に、[イベン

ト履歴] 内で CloudTrail イベントに記録されます。最近のイベントは、AWS アカウント で表示、検索、ダウンロードできます。詳細については、「CloudTrail イベント履歴の操作」を参照してください。

ゾーンシフトの ARC のイベントなど、AWS アカウント で継続しているイベントの記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョン に適用されます。証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づいて対応するため、他の AWS サービスを構成できます。詳細については、次を参照してください:

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての ARC アクションは CloudTrail によってログに記録され、「Amazon Application Recovery Controller のルーティングコントロール API リファレンスガイド」に記載されています。例えば、`StartZonalShift` および `ListManagedResources` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

イベント履歴での ARC イベントの表示

CloudTrail では、[イベント履歴] に最近のイベントが表示されます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。

ゾーンシフトログファイルエントリの理解

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどの情報を含みます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、ゾーンシフトの `ListManagedResources` アクションを実行する CloudTrail ログエントリです。


```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:14:41Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
      "eventName": "ListManagedResources",
      "awsRegion": "us-west-2",
```

```
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "VGXG4ZUE7UZTVCMTJGIAF_EXAMPLE",
      "eventID": "4b5c42df-1174-46c8-be99-d67_EXAMPLE",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```

次の例は、ゾーンシフトの競合の例外を含む StartZonalShift アクションを実行する CloudTrail ログエントリです。

```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:10:38Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
```

```
      "eventName": "StartZonalShift",
     "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "errorCode": "ConflictException",
      "errorMessage": "There's already an active zonal shift for that resource
 identifier: 'arn:aws:testservice:us-west-2:077059137270:testResource/456apples'.
 Active zonal shift: 'bac23b74-176e-c073-de8f-484ca508910f'",
      "requestParameters": {
        "resourceIdentifier": "arn:aws:testservice:us-
west-2:077059137270:testResource/456apples",
        "awayFrom": "usw2-az1",
        "expiresIn": "2m",
        "comment": "HIDDEN_FOR_SECURITY_REASONS"
      },
      "responseElements": null,
      "requestID": "OP4OYXZ54HUPMIPGWH_EXAMPLE",
      "eventID": "0bca6660-e999-43a5-9008-EXAMPLE",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```


ARC でのゾーンシフトの Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

内容

- IAM を使用したゾーンシフトの仕組み
- ゾーンシフトの IAM とアクセス許可
- ARC でのゾーンシフトのアイデンティティベースのポリシーの例

IAM を使用したゾーンシフトの仕組み

IAM を使用して Amazon Application Recovery Controller (ARC) のゾーンシフトへのアクセスを管理する前に、ゾーンシフトで使用できる IAM 機能について説明します。

ゾーンシフトで使用できる IAM の機能

IAM 機能	ゾーンシフトのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	はい

AWS サービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

ARC のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティ

ベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

ARC のアイデンティティベースのポリシーの例を表示するには、「Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例」を参照してください。

ARC 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。

ゾーンシフトのポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

ゾーンシフトにおける ARC アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Route 53 ゾーンシフトで定義されるアクション」を参照してください。

ARC におけるゾーンシフト用ポリシーのアクションでは、アクションの前に次のプレフィックスを使用しています。

```
arc-zonal-shift
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。例えば、次のようになります。

```
"Action": [
      "arc-zonal-shift:action1",
      "arc-zonal-shift:action2"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Describe という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "arc-zonal-shift:Describe*"
```

ゾーンシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンシフトのアイデンティティベースのポリシーの例」を参照してください。

ゾーンシフトのポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

リソースタイプとその ARN のリスト、および各リソースの ARN で指定できるアクションについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義されるアクション

条件キーで使用できるアクションとリソースについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

ゾーンシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンシフトのアイデンティティベースのポリシーの例」を参照してください。

ゾーンシフトのポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

ゾーンシフト条件キーのリストについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

条件キーで使用できるアクションとリソースについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義されるアクション
- Amazon Route 53 - ゾーンシフトで定義されるリソースタイプ

ゾーンシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンシフトのアイデンティティベースのポリシーの例」を参照してください。

ARC のアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセスコントロール (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

ARC には、ABAC に対する以下の部分的なサポートが含まれています。

- ゾーンシフトは、ゾーンシフト用に ARC に登録されている、マネージドリソースの ABAC をサポートしています。Network Load Balancer と Application Load Balancer マネージドリソースにおける ABAC の詳細については、「Elastic Load Balancing ユーザーガイド」の「Elastic Load Balancing での ABAC」を参照してください。

ARC での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

ARC のクロスサービスプリンシパル権限

転送アクセスセッション (FAS) のサポート: あり

IAM エンティティ (ユーザーまたはロール) を使用して でアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使

用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。

アクションにポリシーで追加の依存アクションが必要かどうかを確認するには、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 ゾーンシフト

ARC のサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

ARC のサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

ゾーンシフトはサービスリンクロールを使用しません。

ゾーンシフトの IAM とアクセス許可

このセクションでは、特に Elastic Load Balancing などの別の AWS サービスの機能を使用する場合に、Amazon Application Recovery Controller (ARC) のゾーンシフト機能に対するアクセス許可の仕組みに関する追加情報を提供します。ARC 機能が IAM やアクセス許可とどのように連携するのかといった一般的な情報については、概要トピックの「ARC でのゾーンシフトの Identity and Access Management」を参照してください。

ゾーンシフトは、Application Load Balancer、Network Load Balancer、Amazon EC2 Auto Scaling グループ、Amazon EKS をサポートしています。IAM 条件キーを使用して、IAM アクセス許可ポリシーをこれらのリソースにスコープできます。以下は、異なるタイプの複数のリソースを持つ条件キーを使用するポリシーの例です。

```
{
    "Condition": {
        "StringLike": {
            "arc-zonal-shift:ResourceIdentifier": [
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/net/
*",
                "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/
*",
                "arn:aws:eks:us-east-1:123456789012:cluster/*"
            ]
        }
    },
    "Action": [
        "arc-zonal-shift:StartZonalShift"
    ],
    "Resource": "*",
    "Effect": "Allow"
}
```

詳細については、「サポートされているリソース」を参照してください。

IAM の概要トピックで説明されているアクセス許可に加えて、ゾーンシフトに関する IAM 権限には、以下の点が適用されます。

- ARC でゾーンシフトを使用するのに必要なアクセス許可があることを確認します。詳細については、「ゾーンシフトコンソールアクセス」および「ゾーンシフトオペレーションアクセス」を参照してください。
- ARC でアカウント内のマネージドロードバランサーリソースのゾーンシフトを実行するときは、IAM で Elastic Load Balancing のアクセス許可を追加する必要はありません。
- Elastic Load Balancing へのフルアクセスを提供する AWS マネージドポリシーには、ゾーンシフトを操作するためのアクセス許可が含まれています。Elastic Load Balancing アクセスに AWS マネージドポリシーを使用する場合、ロードバランサーのゾーンシフトを開始したり、Elastic Load Balancing コンソールで を操作するために、ゾーンシフトの IAM で追加のアクセス許可は必要ありません。詳細については、「Elastic Load Balancing のAWS マネージドポリシー」を参照してください。

ARC でのゾーンシフトのアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには、ARC リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認証リファレンス」の「Amazon Application Recovery Controller (ARC) のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- 例: ゾーンシフトコンソールアクセス
- 例: ゾーンシフト API アクション

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが ARC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。

- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の 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。

例: ゾーンシフトコンソールアクセス

Amazon Application Recovery Controller (ARC) コンソールにアクセスするには、最小限のアクセス許可セットが必要です。これらのアクセス許可により、 の ARC リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

でゾーンシフトを使用するためのフルアクセスをユーザーに付与するには AWS マネジメントコンソール、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
                   "arc-zonal-shift:UpdateZonalShift",
                   "arc-zonal-shift:CancelZonalShift"
             ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*"
        }
    ]
}
```

例: ゾーンシフト API アクション

ゾーンシフト API は一時的にトラフィックをアベイラビリティーゾーンからシフトしてアプリケーションを復旧します。

ユーザーがゾーンシフト API アクションを使用できるようにするには、以下のように、ユーザーが使用すべき API オペレーションに対応したポリシーをアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
```

```
                "arc-zonal-shift:UpdateZonalShift",
                "arc-zonal-shift:CancelZonalShift"
            ],
            "Resource": "*"
        }
    ]
}
```

ARC のゾーンオートシフト

ゾーンオートシフトでは、ユーザーに代わって、イベント中にアプリケーションのリソーストラフィック AWS をアベイラビリティーゾーン (AZ) から遠ざけることを に許可します。これにより、復旧までの時間を短縮できます。 は、内部テレメトリが、顧客に影響を与える可能性のあるアベイラビリティーゾーンの障害があることを示すと、オートシフト AWS を開始します。がオートシフト AWS を開始すると、ゾーンオートシフト用に設定したリソースへのアプリケーショントラフィックがアベイラビリティーゾーンから移行し始めます。

ARC は個々のリソースの状態を検査しないことに注意してください。 は、顧客に影響を与える可能性のあるアベイラビリティーゾーンの障害が AWS テレメトリによって検出されたときにオートシフト AWS を開始します。場合によっては、影響が及んでいないリソースのトラフィックがシフトされることもあります。

ゾーンオートシフトでは、通常の練習実行のために、ユーザーに代わってアプリケーションのリソーストラフィック AWS をアベイラビリティーゾーンから遠ざけることを に許可します。ゾーンオートシフトには練習実行が必要です。ARC が練習実行として開始するゾーンシフトは、オートシフト中にアベイラビリティーゾーンからトラフィックをシフトすることがアプリケーションにとって安全であることを保証するのに役立ちます。練習実行では、リソースのトラフィックをアベイラビリティーゾーンから遠ざけるゾーンシフトを開始することによって、1 つのアベイラビリティーゾーンがなくてもアプリケーションが正常に動作することを定期的にテストします。練習実行は毎週行われ、アプリケーションが期待どおりに動作するかどうかの判断に役立つ結果 (SUCCEEDED や FAILED など) が表示されます。

> **Important**
>
> 練習実行を設定したり、ゾーンオートシフトを有効にしたりする前に、アプリケーションリソースがデプロイされているリージョンのすべてのアベイラビリティーゾーンで、アプリケーションリソースの容量をプリスケールすることを強くお勧めします。オートシフトまた

は練習実行が開始されるとき、オンデマンドでのスケーリングに頼るべきではありません。練習実行を含むゾーンオートシフトは独立して動作し、自動スケーリングアクションの完了を待ちません。プレスケーリングの代わりに自動スケーリングを使用すると、アプリケーションの復旧に時間がかかる可能性があります。
自動スケーリングを使用して定期的なトラフィックサイクルを処理する場合は、アベイラビリティーゾーンが失われても正常に動作し続けるように、自動スケーリングの最小容量を設定することを強くお勧めします。

ゾーンオートシフトを有効にしたり、練習実行を設定したりする予定がある場合は、アプリケーションのリソース容量をプレスケーリングした後、1 つのアベイラビリティーゾーンがなくてもアプリケーションが正常に動作することをテストしてください。これをテストするには、ゾーンシフトを開始して、リソースのトラフィックをアベイラビリティーゾーンから遠ざけます。

ゾーンオートシフトを有効にした後は、オンデマンドの練習実行のゾーンシフトを開始し評価することにより、アベイラビリティーゾーンからシフトしたトラフィックでアプリケーションが正常な動作を継続できることを確認するようお勧めします。ARC が実行する通常の練習実行は、オートシフトに十分な容量があることを継続的に確認するために役立ちます。

ゾーンシフトによるテストが有効であることを確認するには、トラフィックが想定どおりにシフト元の AZ からドレインすることを検証することが重要です。例えば、Application Load Balancer と Network Load Balancer の両方とも、この監視に使用可能な Amazon CloudWatch による AZ ごとのメトリクスを提供します。サービスやクライアントが接続を再利用する時間によっては、シフトした AZ へのトラフィックが想定よりも長く続く場合があります。詳細については、「クライアントがエンドポイントに接続したままになる時間を制限する」を参照してください。

ARC コンソールでは、サポートされるリソースのゾーンオートシフトを有効にできます。または、Amazon EC2 コンソールには、特定のロードバランサーリソースのゾーンオートシフトを有効にするオプションがあります。Elastic Load Balancing でゾーンオートシフトを有効にする方法の詳細については、Elastic Load Balancing ユーザーガイド」の「ゾーンシフト」を参照してください。

オートシフトと練習実行のゾーンシフトは一時的なものです。オートシフトでは、影響を受けるアベイラビリティーゾーンが回復すると、 はアベイラビリティーゾーンからリソースのトラフィックの移行を AWS 停止します。顧客のアプリケーショントラフィックは、リージョン内のすべてのアベイラビリティーゾーンに戻ります。練習実行では、トラフィックは 1 つのリソースについて 1 つのアベイラビリティーゾーンから約 30 分間遠ざけられ、その後、リージョン内のすべてのアベイラビリティーゾーンに戻されます。

Amazon EventBridge 通知を設定して、オートシフトや練習実行についてアラートを受け取ることができます。詳細については、「Amazon EventBridge でのゾーンオートシフトの使用」を参照してください。

ゾーンオートシフトと練習実行の仕組み

Amazon Application Recovery Controller (ARC) のゾーンオートシフト機能を使用すると、 がアベイラビリティーゾーンの顧客に影響を与える可能性のある障害がある AWS と判断した場合、ユーザーに代わってリソースのトラフィックをアベイラビリティーゾーンから AWS 遠ざけることができます。ゾーンオートシフトは、 のすべてのアベイラビリティーゾーンで事前にスケーリングされたリソース用に設計されているため AWS リージョン、アプリケーションは 1 つのアベイラビリティーゾーンが失われても正常に動作します。

ゾーンオートシフトでは、ARC がリソースのトラフィックを定期的に 1 つのアベイラビリティーゾーンから遠ざけるようにする練習実行を設定する必要があります。ARC は、練習実行設定が関連付けられているリソースにつき、約 1 週間ごとに練習実行をスケジュールします。各リソースの練習実行は個別にスケジュールされます。

各練習実行について、ARC は結果を記録します。練習実行がブロック条件によって中断された場合、練習実行の結果は成功としてマークされません。練習実行の結果の詳細については、「練習実行の結果」を参照してください。

Amazon EventBridge 通知を設定して、オートシフトや練習実行についてアラートを受け取ることができます。詳細については、「Amazon EventBridge でのゾーンオートシフトの使用」を参照してください。

内容

- ゾーンオートシフトについて
- がオートシフト AWS を開始および停止する場合
- ARC が練習実行をスケジュール、開始、終了するタイミング
- 練習実行の容量チェック
- 練習実行とオートシフトの通知
- ゾーンシフトの優先順位
- リソースのアクティブなオートシフトまたは練習実行を停止する
- トラフィックを遠ざける方法
- 練習実行のアラーム
- ブロックされた時間枠と許可された時間枠 (UTC)

ゾーンオートシフトについて

ゾーンオートシフトは、ユーザーに代わってアプリケーションリソーストラフィックをアベイラビリティーゾーンから AWS 遠ざける機能です。 は、内部テレメトリが、顧客に影響を与える可能性のあるアベイラビリティーゾーンの障害があることを示すと、オートシフト AWS を開始します。内部テレメトリには、 AWS ネットワーク、Amazon EC2、Elastic Load Balancing サービスなど、複数のソースからのメトリクスが組み込まれています。

サポートされている AWS リソースに対してゾーンオートシフトを手動で有効にする必要があります。

リージョン内の複数の (通常は 3 つの) AZs のロードバランサーに AWS アプリケーションをデプロイして実行し、静的安定性をサポートするように事前スケーリングすると、 はトラフィックをオートシフトで移行することで、AZ 内の顧客アプリケーションをすばやく復旧 AWS できます。リソーストラフィックをリージョン内の他の AZs に移行することで、 は、停電、AZ のハードウェアまたはソフトウェアの問題、またはその他の障害による潜在的な影響の期間と重大度を減らす AWS ことができます。

ARC がサポートするリソースは、指定した AZ を異常としてマークするための連携機能を備えており、その結果として、障害のある AZ からトラフィックがシフトされます。

リソースのゾーンオートシフトを有効にする場合は、そのリソースの練習実行も設定する必要があります。リージョンのアベイラビリティーゾーンの 1 つがなくてもアプリケーションを実行するのに十分な容量があることを確認できるように、 AWS は約 1 週間ごとに 30 分間の練習実行を実施します。

ゾーンシフトと同様に、ゾーンオートシフトによってトラフィックが AZ から遠ざけられない特定のシナリオがいくつかあります。例えば、AZ 内のロードバランサーのターゲットグループにインスタンスが含まれていない場合や、すべてのインスタンスが「異常」である場合、ロードバランサーはフェイルオープン状態であり、AZ の 1 つをシフトできません。

ゾーンオートシフトの詳細については、「ARC のゾーンオートシフト」を参照してください。

がオートシフト AWS を開始および停止する場合

リソースのゾーンオートシフトを有効にすると、 AWS がユーザーに代わってイベント中にアプリケーションのリソーストラフィックをアベイラビリティーゾーンから遠ざけることを承認し、復旧までの時間を短縮できます。

これを実現するために、ゾーンオートシフトは AWS テレメトリを使用して、顧客に影響を与える可能性のあるアベイラビリティーゾーンの障害をできるだけ早く検出します。 AWS がオートシフトを

開始すると、設定済みリソースへのトラフィックは、顧客に影響を与える可能性のある障害のあるアベイラビリティーゾーンからただちに遠ざけられます。

ゾーンオートシフトは、 AWS リージョン内のすべてのアベイラビリティーゾーンのアプリケーションリソースをあらかじめスケールされたお客様向けに設計された機能です。オートシフトまたは練習実行が開始されるとき、オンデマンドでのスケーリングに頼るべきではありません。

AWS は、アベイラビリティーゾーンが回復したと判断されると、オートシフトを終了します。

ARC が練習実行をスケジュール、開始、終了するタイミング

ARC は、1 つのリソースについて、毎週約 30 分間の練習実行をスケジュールします。ARC は、各リソースの練習実行を個別にスケジュール、開始、および管理します。ARC は、同じアカウントの複数のリソースの練習実行をまとめることはありません。ゾーンオートシフトイベントのセットアップが安全であることを確認するために、オンデマンドの練習実行を自分で開始することもできます。

練習実行が予想された時間だけ中断されずに続行すると、SUCCESSFUL という結果でマークされます。他にも可能性のある結果として、FAILED、INTERRUPTED、CAPACITY_CHECK_FAILED、PENDING があります。結果の値と説明は、「練習実行の結果」セクションに記載されています。

ARC が練習実行を中断して終了するシナリオがいくつかあります。例えば、練習実行中にオートシフトが開始した場合、ARC は練習実行を中断して終了します。別の例として、練習実行に対してリソースが不利な反応を示し、練習実行を監視するために指定したアラームが ALARM 状態になったとします。このシナリオでも、ARC は練習実行を中断して終了します。

さらに、ARC がリソースについてスケジュールされた練習実行を開始しないシナリオもいくつかあります。

リソースの練習実行が中断またはブロックされた場合に対応して、ARC は次のことを行います。

- リソースの練習実行が進行中に中断された場合、ARC は毎週の練習実行が終了したとみなし、そのリソースの新しい練習実行を次の週にスケジュールします。このシナリオでは、毎週の練習の結果は FAILED ではなく INTERRUPTED です。練習実行の結果が FAILED に設定されるのは、練習実行を監視する結果アラームが練習実行中に ALARM 状態になった場合のみです。
- リソースの練習実行の開始が予定されているときにブロッキング制約がある場合、ARC は練習実行を開始しません。ARC は引き続き定期的な監視を行い、ブロッキング制約が 1 つ以上あるかどうかを判断します。ブロッキング制約がない場合、ARC はリソースの練習実行を開始します。

以下は、ARC がリソースの練習実行を開始または続行することを禁止するブロック制約の例です。

- ARC は、進行中の AWS Fault Injection Service 実験がある場合、練習実行を開始または続行しません。ARC が練習実行の開始をスケジュールしたときに AWS FIS イベントがアクティブな場合、ARC は練習実行を開始しません。ARC は、 AWS FIS イベントを含むブロック制約について練習実行全体を監視します。練習実行がアクティブな間に AWS FIS イベントが開始された場合、ARC は練習実行を終了し、リソースに対して次に定期的にスケジュールされた練習実行まで別のイベントを開始しようとしません。
- リージョンに現在の AWS イベントがある場合、ARC はリソースの練習実行を開始せず、リージョンでアクティブな練習実行を終了します。

練習実行が中断されずに終了すると、ARC は通常どおり 1 週間後に次の練習実行をスケジュールします。指定した AWS FIS 実験やブロックされた時間枠などのブロック制約が原因で練習実行が開始されない場合、ARC は練習実行が開始されるまで練習実行の開始を試行し続けます。

練習実行の容量チェック

練習実行が開始されると、一時的にトラフィックをアベイラビリティーゾーンから遠ざけるため、ARC は他のアベイラビリティーゾーンにトラフィックを安全に AZ から遠ざけるのに十分な容量があることをチェックします。十分な容量がない場合は、練習実行のトラフィックシフトは開始されず、練習実行が終了します。

さらに、ゾーンオートシフトが完了すると、オートシフトによって開始されたトラフィックシフトを ARC が終了する前に、ARC はロードバランサーリソースの容量チェックを実行します。オートシフトの終了時に容量チェックに失敗する場合、トラフィックは移動元のアベイラビリティーゾーンにシフトされません。

バランス容量のチェックは、ロードバランサーと Auto Scaling グループに対してのみ完了します。

ロードバランサーリソースの場合、容量チェックは、ロードバランサーに関連付けられた正常なホストがアベイラビリティーゾーンに分散されていることを検証します。特に、容量チェックでは、リソースが登録されているすべてのアベイラビリティーゾーンで、正常なホストの数のバランスが取れていることを確認します。容量チェックにおけるバランスとは、各アベイラビリティーゾーンの正常な容量が他のゾーンと同等であり、差異がわずかであることを意味します。

容量チェックは、Lambda タイプであるターゲットグループのロードバランサーにも、Application Load Balancer にも適用されないことに注意してください。これらのターゲットはゾーンに設定されていないためです。

Auto Scaling グループのキャパシティチェックも完了します。Auto Scaling グループの場合、キャパシティチェックは、Auto Scaling グループの正常なゾーン容量の合計、つまりすべてのアベイラビリ

ティーゾーンにわたる正常なホストの合計数が、その Auto Scaling グループに必要な容量セットを満たしていることを確認します。

容量チェックが失敗する場合

リソースで使用可能な容量のバランスが取れていないことが容量チェックでわかった場合、練習実行の結果は CAPACITY_CHECK_FAILED になります。容量チェックが失敗する理由の詳細については、ZonalShiftSummary のコメントフィールドを参照してください。練習実行ゾーンシフトのコメントフィールドを見つけるには、以下を実行してください。

1. を使用して AWS CLI、ListZonalShifts API オペレーションを使用して練習実行で指定したリソースのゾーンシフトを一覧表示します。

   例えば、ゾーンシフトを返すために、次のようなコマンドを実行できます。

   ```
   aws arc-zonal-shift start-practice-run
       --resource-
   identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890"
   ```

2. 返された ZonalShiftSummary オブジェクトの配列を確認して、容量チェックのために失敗した練習実行のゾーンシフトを見つけます。
3. 該当するゾーンシフトに関して、Comment フィールドの情報を確認してください。

練習実行とオートシフトの通知

Amazon EventBridge 通知を設定すると、リソースの練習実行とオートシフトに関する通知の受信を選択できます。オートシフトオブザーバー通知と呼ばれるリソースに対してゾーンオートシフトを有効にしていない場合でも、EventBridge 通知を設定できます。オートシフトオブザーバー通知では、アベイラビリティーゾーンに障害が発生する可能性があるときに ARC が開始するすべてのオートシフトについて通知されます。このオプションは、通知 AWS リージョン を受信する各 で設定する必要があることに注意してください。

オートシフトのオブザーバー通知を有効にする手順については、「オートシフトのオブザーバー通知の有効化または無効化」を参照してください。通知オプションの詳細と EventBridge での構成方法については、「Amazon EventBridge でのゾーンオートシフトの使用」を参照してください。

ゾーンシフトの優先順位

一度に適用できるゾーンシフトは 1 つのみです。つまり、リソースに対してゾーンシフト、お客様が開始したゾーンシフト、オートシフト、または AWS FIS 実験を実行する練習は 1 つだけです。2

番目のゾーンシフトが開始されると、ARC は優先順位に従いリソースに有効なゾーンシフトタイプを判断します。

優先順位の一般的な原則は、顧客として開始するゾーンシフトを他のシフトタイプよりも優先することです。ただし、現在実行中 AWSの練習実行では、オンデマンドの練習実行を開始できないことに注意してください。

ARC における優先順位について説明するために、シナリオ例における優先順位の仕組みを以下に示します。

適用されるゾーンシフトタイプ	開始されるゾーンシフトタイプ	結果
AWS FIS 実験	練習実行	AWS FIS 実験が優先されるため、練習実行は開始されません。
AWS FIS 実験	手動ゾーンシフト	AWS FIS 実験はキャンセルされ、手動ゾーンシフトが適用されます。
AWS FIS 実験	ゾーンオートシフト	AWS FIS 実験はキャンセルされ、ゾーンオートシフトが適用されます。
AWS FIS 実験	AWS FIS 実験	自動 AWS FIS シフトアクションをトリガーした既存の AWS FIS 実験が実行されているため、開始された実験は開始できません。
練習実行	手動ゾーンシフト	練習実行がキャンセルされて結果が `INTERRUPTED` に設定され、ゾーンシフトが適用されます。
練習実行	AWS FIS 実験	練習実行がキャンセルされて結果が `INTERRUPTED` に設

適用されるゾーンシフトタイプ	開始されるゾーンシフトタイプ	結果
		定され、 AWS FIS 実験が適用されます。
練習実行	ゾーンオートシフト	練習実行がキャンセルされて結果が INTERRUPTED に設定され、ゾーンオートシフトが適用されます。
手動ゾーンシフト	練習実行	練習実行の開始が失敗します。
手動ゾーンシフト	AWS FIS 実験	AWS FIS 実験は開始に失敗するか、すでに進行中の場合は失敗します。
手動ゾーンシフト	ゾーンオートシフト	ゾーンオートシフトは ACTIVE ですが、リソースに APPLIED されません。手動ゾーンシフトが優先されます。
ゾーンオートシフト	AWS FIS 実験	AWS FIS 実験は開始に失敗するか、進行中の場合は失敗します。
ゾーンオートシフト	手動ゾーンシフト	ゾーンオートシフトは ACTIVE ですが、リソースに APPLIED されません。手動ゾーンシフトが優先されます。
ゾーンオートシフト	練習実行	ゾーンオートシフトが優先されるため、練習実行の開始が失敗します。

リソースで現在実施されているトラフィックシフトは、適用されたゾーンシフトステータスが APPLIED に設定されています。一度に APPLIED に設定できるシフトは 1 つだけです。進行中の他のシフトは NOT_APPLIED に設定されますが、ACTIVE ステータスのままです。

リソースのアクティブなオートシフトまたは練習実行を停止する

リソースの進行中のオートシフトを停止するには、ゾーンシフトをキャンセルする必要があります。

そのリソースについては、これまでと同じスケジュールで定期的に練習実行が行われます。オートシフトを無効にするだけでなく、練習実行も停止したい場合は、リソースに関連付けられている練習実行設定を削除する必要があります。

練習実行設定を削除すると、 はリソースのトラフィックを毎週アベイラビリティーゾーンから遠ざける練習実行を AWS 停止します。さらに、ゾーンオートシフトには練習実行が必要なため、ARC コンソールを使用して練習実行設定を削除すると、このアクションによりリソースのゾーンオートシフトも無効になります。ただし、ゾーンオートシフト API を使用して練習実行を削除する場合は、まずリソースのゾーンオートシフトを無効にする必要があることに注意してください。

詳細については、「ゾーンオートシフトのキャンセル」および「ゾーンオートシフトの有効化と操作」を参照してください。

トラフィックを遠ざける方法

オートシフトと練習実行のゾーンシフトの場合、ARC が顧客によって開始されたゾーンシフトに使用するのと同じメカニズムを使用して、トラフィックはアベイラビリティーゾーンから遠ざけられます。ヘルスチェックに異常があると、Amazon Route 53 は、リソースの対応する IP アドレスを DNS から削除します。それにより、トラフィックはアベイラビリティーゾーンからリダイレクトされます。新しい接続は、 AWS リージョン 代わりに の他のアベイラビリティーゾーンにルーティングされるようになりました。

オートシフトでは、アベイラビリティーゾーンが回復してオートシフトを終了する AWS と、ARC はヘルスチェックプロセスを元に戻して、Route 53 ヘルスチェックの元に戻すようリクエストします。その後、元のゾーン IP アドレスが復元され、ヘルスチェックが引き続き正常であれば、そのアベイラビリティーゾーンはアプリケーションのルーティングに再び含まれます。

オートシフトは、ロードバランサーやアプリケーションの基本的な状態を監視するヘルスチェックに基づくものではないことに注意することが重要です。ARC は、ヘルスチェックを「異常」に設定し、オートシフトまたはゾーンシフトを終了したときにヘルスチェックを再び「正常」に戻すようにリクエストすることにより、ヘルスチェックを使用してトラフィックをアベイラビリティーゾーンから遠ざけます。

練習実行のアラーム

ゾーンオートシフトでは、練習実行に CloudWatch アラームを 2 つ指定できますが、それは結果アラームとブロッキングアラームです。

結果アラーム (必須)

最初のタイプのアラームである結果アラームでは、少なくとも 1 つのアラームを指定する必要があります。30 分間の練習実行中にトラフィックがアベイラビリティーゾーンから遠ざけられるときに、結果アラームを設定して、アプリケーションの状態を監視する必要があります。

練習実行を有効にするには、結果アラームとして、次の両方の基準を満たす CloudWatch アラームを少なくとも 1 つ指定します。

アラームは、リソースまたはアプリケーションのメトリクスをモニタリングします。

AND

1 つのアベイラビリティーゾーンを失ってアプリケーションに悪影響が及ぶと、アラームは ALARM 状態で応答します。

詳細については、「ゾーンオートシフトを設定する際のベストプラクティス」の「練習実行について指定するアラーム」セクションを参照してください。

結果アラームには、ARC が各練習実行について報告する練習実行結果の情報も表示されます。結果アラームが ALARM 状態になると、ARC は練習実行を終了し、FAILED の練習実行の結果を返します。練習実行が 30 分間のテスト期間を完了しても指定した結果アラームが ALARM 状態にならない場合、返される結果は SUCCEEDED です。すべての結果値のリストと説明は、「練習実行の結果」セクションに記載されています。

ブロッキングアラーム (オプション)

オプションで、2 種類目のアラーム、ブロッキングアラームを指定できます。1 つ以上のアラームが ALARM 状態のとき、ブロッキングアラームは練習実行の開始または続行をブロックします。ブロッキングアラームは、少なくとも 1 つのアラームが ALARM 状態になると、練習実行のトラフィックシフトの開始をブロックし、進行中の練習実行を停止します。

例えば、複数のマイクロサービスを使用する大規模なアーキテクチャでは、1 つのマイクロサービスに問題が発生すると、通常、アプリケーション環境内の他のすべての変更を停止する必要があり、これにはブロッキング練習実行も含まれます。ARC にブロッキングアラームを追加してこれを実行できます。

ブロックされた時間枠と許可された時間枠 (UTC)

特定の歴日、または特定の時間枠 (つまり UTC の日時) について練習実行をブロックまたは許可するオプションがあります。

例えば、2024 年 5 月 1 日にアプリケーションの更新を開始する予定があり、その時点で練習実行によってトラフィックが遠ざけられないようにしたい場合は、`2024-05-01` をブロック日に設定できます。

または、ビジネスレポートの概要を週に 3 日作成するとします。このシナリオでは、次のような定期的な曜日と時刻をブロックされる時間枠として設定できます (例: UTC: `MON-20:30-21:30 WED-20:30-21:30 FRI-20:30-21:30`)。

または、ARC が練習実行を開始してセットアップをテストするには、水曜日と金曜日の正午から 5:00 までが最適だと思うかもしれません。このシナリオでは、次のような定期的な曜日と時刻を許可される時間枠として設定できます (例: UTC: `WED-12:00-17:00 FRI-12:00-17:00`)。

AWS リージョン ゾーンオートシフトの可用性

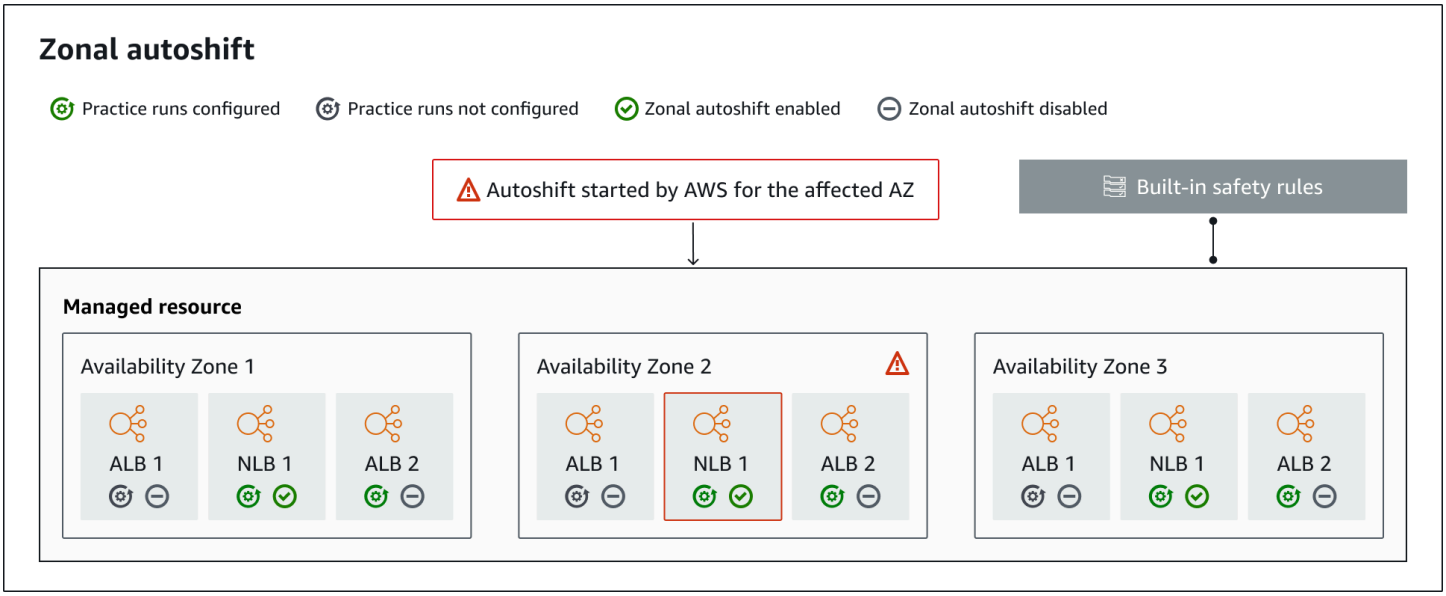
ゾーンシフトとゾーンオートシフトは現在 AWS リージョン、中国リージョン、つまり中国 (北京) リージョンと中国 (寧夏) リージョンで利用可能です。

Amazon Application Recovery Controller (ARC) を使用するリソースには、追加の考慮事項が含まれる場合があります。詳細については、「サポートされているリソース」を参照してください。

リージョンの一覧および ARC のリージョンサポートとリージョンサービスエンドポイントに関する詳細は、「Amazon Web Services 全般のリファレンス」にある「Amazon Application Recovery Controller (ARC) のエンドポイントとクォータ」を参照してください。

ゾーンオートシフトのコンポーネント

次の図は、トラフィックをアベイラビリティーゾーンから遠ざけるオートシフトの例を示しています。顧客に影響を与える可能性のあるアベイラビリティーゾーンの障害が内部テレメトリによって示された場合、 AWS はオートシフトを開始します。



以下は、ARC におけるゾーンオートシフト機能のコンポーネントです。

ゾーンオートシフト

ゾーンオートシフトは、何も操作しなくても、リソースのトラフィックを遠ざけます。ゾーンオートシフトは、カスタマーに影響を与える可能性のあるアベイラビリティーゾーンの障害が内部テレメトリによって示されると、 がオートシフト AWS を開始する ARC の機能です。場合によっては、影響が及んでいないリソースがシフトされることもあります。

練習実行

リソースのゾーンオートシフトを有効にする場合は、リソースのゾーンオートシフト練習実行も設定する必要があります。 は、練習実行のゾーンシフトを約毎週約 30 分間 AWS 実行します。練習実行は、オンデマンドでもスケジュールできます。

練習実行により、1 つのアベイラビリティーゾーンが失われても、アプリケーションが正常に動作することを確認できます。練習実行では、 はリソースのトラフィックをゾーン AWS シフトのある 1 つのアベイラビリティーゾーンからシフトし、練習実行が終了したらトラフィックをシフトバックします。

練習実行設定

練習実行設定を使用すると、ARC がゾーンオートシフトでリソースの練習実行を開始できる時間枠 (ブロックまたは許可される時間枠) を定義できます。また、 AWS 練習実行の CloudWatch アラームも定義します。練習実行設定はいつでも編集でき、ブロックまたは許可される時間枠を追加または変更したり、練習実行のアラームを更新したりできます。

ゾーンオートシフトを有効にするには、リソース用の練習実行設定が必要です。

練習実行は削除できますが、まず、ゾーンオートシフトを無効にする必要があります。

練習実行アラーム

練習実行を設定するときには、リソースとアプリケーションの要件に基づいて、(CloudWatch で最初に作成する) CloudWatch アラームを指定します。指定したアラームは、アプリケーションが練習実行によって悪影響を受けた場合に、練習実行の開始をブロックしたり、進行中の練習実行を停止したりできます。

指定したアラームが ALARM 状態になると、ARC は練習実行のゾーンシフトを終了します。これにより、リソースのトラフィックはアベイラビリティーゾーンから遠ざけられなくなります。

練習実行用に指定するアラームには 2 種類あります。1 つは練習実行中にリソースとアプリケーションの状態を監視する結果アラームです。もう 1 つはブロッキングアラームであり、練習実行の開始を妨げたり、進行中の練習実行を停止したりするように設定できます。少なくとも 1 つの結果アラームが必須です。ブロッキングアラームはオプションです。

練習実行の結果

ARC は各練習実行の結果を報告します。可能な練習実行の結果は以下のとおりです。

- PENDING: 練習実行のゾーンシフトはアクティブ (進行中) です。まだ結果は戻されていません。
- SUCCEEDED: 練習実行中、結果アラームは ALARM 状態にならず、練習実行は 30 分間のテスト期間をすべて完了しました。
- INTERRUPTED: 結果アラームが ALARM 状態になったのではない理由で、練習実行は終了しました。練習実行は、以下のようなさまざまな理由で中断されることがあります。例えば、練習実行について指定されたブロッキングアラームが ALARM 状態になったために終了した練習走行は、INTERRUPTED の結果になります。INTERRUPTED 結果の理由の詳細については、「練習実行の結果」を参照してください。
- FAILED: 練習実行中に結果アラームが ALARM 状態になりました。
- CAPACITY_CHECK_FAILED: アベイラビリティーゾーン間におけるロードバランシングと Auto Scaling グループリソースとのバランスの取れたキャパシティのチェックに失敗しました。

組み込みの安全ルール

ARC に組み込まれている安全ルールにより、1 つのリソースについて一度に複数のトラフィックシフトが行われることはありません。つまり、アベイラビリティーゾーンからトラフィックをア

クティブにシフトできるのは、そのリソースについて、顧客が開始したゾーンシフト、(AWS または顧客が開始した) 練習実行のゾーンシフト、またはオートシフトの 1 つのみです。例えば、あるリソースがオートシフトで遠ざけられているときにゾーンシフトを開始した場合は、ゾーンシフトが優先されます。詳細については、「ゾーンシフトの優先順位」を参照してください。

リソース識別子

ゾーンオートシフトを有効にするリソースの識別子で、リソースの Amazon リソースネーム (ARN) です。ゾーンシフトには、ARC がサポートする AWS サービスにあるアカウント内のリソース用ゾーンオートシフトのみを有効にできます。

マネージドリソース

Application Load Balancer は、ゾーンオートシフトの ARC でリソースを自動的に登録します。ゾーンオートシフトの他のリソースは、手動でオプトインする必要があります。

リソース名

ARC のマネージドリソースの名前です。

適用ステータス

適用ステータスは、リソースに対してトラフィックシフトが適用されているかどうかを示します。ゾーンオートシフトを設定すると、1 つのリソースに複数のアクティブなトラフィックシフト、つまり、練習実行ゾーンシフト、顧客によって開始されたゾーンシフト、またはオートシフトが発生する可能性があります。ただし、リソースに適用される、つまり有効になるのは一度に 1 つのみです。ステータス APPLIED のシフトによって、リソースについてアプリケーショントラフィックが遠ざけられたアベイラビリティーゾーンと、そのトラフィックシフトが終了するタイミングが決まります。

シフトタイプ

ゾーンシフトタイプを定義します。ゾーンシフトは、次のいずれかのタイプになります。

- ZONAL_SHIFT
- ZONAL_AUTOSHIFT
- PRACTICE_RUN
- FIS_EXPERIMENT

ゾーンオートシフトのデータプレーンとコントロールプレーン

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは、可用性が高く、災害シナリオで必

要なときに使用できるようにすることをお勧めします。信頼性と耐障害性を最大限に高めるため、可能であれば通常、データプレーン機能をメカニズムに使用する必要があります。そのことを念頭に置いて、サービス機能がコントロールプレーンとデータプレーンにどのように分けられているのか、また、サービスのデータプレーンで非常に高い信頼性が期待できるのはどのような場合なのかを理解することが重要です。

一般に、コントロールプレーンを使用すると、サービス内のリソースの作成、更新、削除などの基本的な管理機能を実行できます。データプレーンはサービスのコア機能を提供します。

データプレーン、コントロールプレーン、および が高可用性目標を達成するためのサービス AWS を構築する方法の詳細については、Amazon Builders' Library の「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。

ARC のゾーンオートシフトの料金

ゾーンオートシフトの場合、 は、顧客アプリケーションに悪影響を及ぼす可能性のある潜在的な問題があると AWS が判断した場合、サポートされているリソースのためにユーザーに代わってアベイラビリティーゾーンからトラフィックを AWS シフトします。ゾーンオートシフトは追加料金なしで使用できます。

ARC の料金および料金例の詳細については、「ARC の料金」を参照してください。

ゾーンオートシフトを設定する際のベストプラクティス

Amazon Application Recovery Controller (ARC) でゾーンオートシフトを有効にする際は、次のベストプラクティスと考慮事項に注意してください。

ゾーンオートシフトには、オートシフトと練習実行ゾーンシフトの 2 種類のトラフィックシフトがあります。

- オートシフト AWS を使用すると、ユーザーに代わってイベント中にアプリケーションリソーストラフィックをアベイラビリティーゾーンから遠ざけることで、復旧までの時間を短縮できます。
- 練習実行では、ARC がユーザーに代わってゾーンシフトを開始、またはユーザーがゾーンシフトの練習実行を開始します。 AWS 練習実行ゾーンシフトは、トラフィックをリソースのアベイラビリティーゾーンから遠ざけ、毎週の頻度で再度シフトします。練習実行は、リージョンのアベイラビリティーゾーンの容量を十分にスケールアップして、1 つのアベイラビリティーゾーンが失われてもアプリケーションの正常な動作を確保できます。

オートシフトと練習実行では、ベストプラクティスと考慮すべき点がいくつかあります。ゾーンオートシフトを有効にしたり、リソースの練習実行を設定したりする前に、以下のトピックを確認してください。

トピック

- クライアントがエンドポイントに接続したままになる時間を制限する
- リソース容量の事前スケーリングとトラフィックの移行のテスト
- リソースタイプと制約を理解する
- 練習実行のアラームを指定する
- 練習実行の結果を評価する

クライアントがエンドポイントに接続したままになる時間を制限する

Amazon Application Recovery Controller (ARC) がゾーンシフトやゾーンオートシフトなどを使用してトラフィックを障害から遠ざける場合、ARC がアプリケーショントラフィックを移動するために使用するメカニズムは DNS 更新です。DNS 更新により、すべての新しい接続が障害のある場所から遠ざけられます。ただし、既存のオープン接続を持つクライアントは、クライアントが再接続するまで、障害が発生したロケーションに対してリクエストを引き続き行う場合があります。迅速な復旧を確保するために、クライアントがエンドポイントに接続し続ける時間を制限することをお勧めします。

Application Load Balancer を使用する場合は、`keepalive` オプションを使用して接続の継続期間を設定できます。アプリケーションの目標復旧時間に合わせて、例えば 300 秒のように `keepalive` 値を小さくすることをお勧めします。`keepalive` 時間を設定する際は、一般的に再接続が増えてレイテンシーに影響が出ることと、障害のある AZ やリージョンからすべてのクライアントをより早く切り替えられることとのトレードオフである点を考慮してください。

Application Load Balancer の `keepalive` のオプション設定の詳細については、「Application Load Balancer ユーザーガイド」の「HTTP クライアントのキープアライブ期間」を参照してください。

リソース容量の事前スケーリングとトラフィックの移行のテスト

がゾーン AWS シフトまたはオートシフトのためにトラフィックを 1 つのアベイラビリティーゾーンから遠ざける場合、残りのアベイラビリティーゾーンがリソースのリクエストレートの増加に対応できることが重要です。このパターンは静的安定性と呼ばれます。詳細については、The Amazon Builder's Library のホワイトペーパー「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。

例えば、アプリケーションがクライアントにサービスを提供するために 30 個のインスタンスを必要とする場合、3 つのアベイラビリティーゾーンに 15 個のインスタンスをプロビジョニングして、合計 45 個のインスタンスをプロビジョニングする必要があります。これにより、 がオートシフトまたは練習実行中に 1 つのアベイラビリティーゾーンからトラフィックを遠ざ AWS ける場合でも、2 つのアベイラビリティーゾーンにまたがる残りの 30 個のインスタンスをアプリケーションのクライアントに提供AWS できます。

ARC のゾーンオートシフト機能は、1 つのアベイラビリティーゾーンが失われても正常に動作するように事前にスケーリングされたリソースを持つアプリケーションがある場合に、アベイラビリティーゾーンの AWS イベントから迅速に復旧するのに役立ちます。リソースのゾーンオートシフトを有効にする前に、 AWS リージョン内の設定済みのすべてのアベイラビリティーゾーンのリソース容量をスケーリングしてください。次に、リソースのゾーンシフトを開始して、トラフィックがアベイラビリティーゾーンから遠ざけられても、アプリケーションが正常に動作することをテストします。

ゾーンシフトでテストした後、ゾーンオートシフトを有効にして、アプリケーションリソースの練習実行を設定します。独自のオンデマンド練習実行を実行して、設定が適切にスケールされるようにします。ゾーンオートシフトを使った定期的な練習実行は、容量が引き続き適切にスケーリングされていることを継続的に確認するのに役立ちます。複数のアベイラビリティーゾーンにまたがって十分な容量があれば、アプリケーションはオートシフト中も中断することなくクライアントにサービスを提供し続けることができます。

リソースのゾーンシフトを開始する方法の詳細については、「ARC のゾーンシフト」を参照してください。

リソースタイプと制約を理解する

ゾーンオートシフトは、ゾーンシフトによってサポートされるすべてのリソースについて、アベイラビリティーゾーン外へのトラフィックのシフトをサポートします。一部の特定のリソースシナリオでは、ゾーンオートシフトではオートシフトのためにアベイラビリティーゾーンからトラフィックがシフトされません。

例えば、アベイラビリティーゾーン内のロードバランサーのターゲットグループにインスタンスが含まれていない場合や、すべてのインスタンスが「異常」である場合、ロードバランサーはフェイルオープン状態になります。がこのシナリオでロードバランサーのオートシフト AWS を開始した場合、ロードバランサーがすでにフェイルオープン状態になっているため、ロードバランサーが使用するアベイラビリティーゾーンはオートシフトによって変更されません。これは想定される動作です。Autoshift では、すべてのアベイラビリティーゾーンがオープンに失敗 (異常) AWS リージョン した場合、1 つのアベイラビリティーゾーンが異常になり、トラフィックを の他のアベイラビリティーゾーンにシフトすることはできません。

すべての要件や注意すべき例外など、サポートされているリソースの詳細を確認するには、「サポートされているリソース」を参照してください。

練習実行のアラームを指定する

ゾーンオートシフトによる練習実行には、少なくとも 1 つのタイプのアラーム (結果アラーム) を設定する必要があります。オプションで、2 番目のタイプのアラーム (ブロッキングアラーム) も設定できます。

リソースの練習実行に対して設定する CloudWatch アラームについて検討するときには、次の点に注意してください。

- 練習実行設定には、少なくとも 1 つの結果アラームを設定する必要があります。結果アラームについては、リソースまたはアプリケーションのメトリクスが、アベイラビリティーゾーンからトラフィックを遠ざけるとパフォーマンスに悪影響があることを示したときには、CloudWatch アラームを ALARM 状態になるように設定することをお勧めします。例えば、リソースのリクエストレートのしきい値を決定して、そのしきい値を超えたときには ALARM 状態になるようにアラームを設定できます。 AWS が練習実行を終了して FAILED 結果を返すように、適切なアラームを設定する必要があります。
- 重要業績評価指標 (KPI) を CloudWatch アラームとして実装することを推奨している「AWS Well Architected フレームワーク」に従うことをお勧めします。その場合、これらのアラームを使用して、安全トリガーとして使用する複合アラームを作成し、アプリケーションが KPI を見逃す可能性がある場合には練習実行が開始されないようにすることができます。アラームが ALARM 状態でなくなると、ARC はそのリソースに対して次回の練習実行がスケジュールされている時点で練習実行を開始します。
- 練習実行のブロックアラームでは、1 つ (または複数) を設定することを選択した場合、 AWS 練習実行を開始しないことを示すために使用する特定のメトリクスを追跡することを選択できます。たとえば、アラームが進行中のインシデントがあることを示す場合などです。
- 練習実行アラームでは、各アラームの Amazon リソースネーム (ARN) を指定します。そのため、まず Amazon CloudWatch でアラームを設定する必要があります。指定する CloudWatch アラームは複合アラームでもかまいません。これにより、アラームの ALARM 状態への移行をトリガーできるアプリケーションとリソースの複数のメトリクスとチェックを含めることができます。または、個別のアラームを設定してから、練習実行設定で各タイプの複数のアラームを指定できます。詳細については、「Amazon CloudWatch ユーザーガイド」の「アラームの結合」を参照してください。
- 練習実行について指定する CloudWatch アラームが、練習実行を設定しているリソースと同じリージョンにあることを確認してください。

練習実行の結果を評価する

ARC は各練習実行の結果を報告します。練習実行の後、結果を評価し、アクションを実行する必要があるかどうかを判断します。例えば、容量のスケーリングやアラーム設定の調整が必要になる場合があります。

可能な練習実行の結果は以下のとおりです。

- SUCCEEDED: 練習実行中に結果アラームが ALARM 状態にならず、練習実行は 30 分間のテスト期間をすべて完了しました。
- FAILED: 練習実行中に少なくとも 1 つの結果アラームが ALARM 状態になりました。
- INTERRUPTED: 結果アラームが ALARM 状態になったのではない理由で、練習実行は終了しました。練習実行は、以下のようなさまざまな理由で中断される可能性があります。
  - でオートシフト AWS が開始された AWS リージョン か、リージョンにアラーム条件が発生したため、練習実行が終了しました。
  - 練習実行は、リソースの練習実行設定が削除されたために、終了しました。
  - 練習実行は、練習実行ゾーンシフトでトラフィックが遠ざけられたアベイラビリティーゾーンのリソースについて、顧客開始のゾーンシフトが開始されたために終了しました。
  - 練習実行は、練習実行設定に指定された CloudWatch アラームにアクセスできなくなったために終了しました。
  - 練習実行に指定されたブロッキングアラームが ALARM 状態に入ったため、練習実行は終了しました。
  - 練習実行は未知の理由で終了しました。
  - 優先されるゾーンオートシフトが開始されたため、練習実行が終了しました。「ゾーンシフトの優先順位」を参照してください。
- CAPACITY_CHECK_FAILED: アベイラビリティーゾーン間におけるロードバランシングと Auto Scaling グループリソースとのバランスの取れたキャパシティのチェックに失敗しました。
- PENDING: 練習実行はアクティブ (進行中) です。まだ結果は戻されていません。

ゾーンオートシフト API オペレーション

次の表に、ゾーンオートシフトで使用できる ARC API オペレーションを示します。でゾーンオートシフト API オペレーションを使用する例については AWS CLI、「」を参照してください。

AWS Command Line Interfaceで一般的なゾーンオートシフト API オペレーションを使用する方法の例については、「ゾーンオートシフト AWS CLI で を使用する例」を参照してください。

Action	ARC コンソールの使用	ARC API の使用
練習実行設定を作成する	「ゾーンオートシフトの有効化または無効化」を参照してください。	「CreatePracticeRunConfiguration」を参照してください。
練習実行設定を削除する	「練習実行設定の設定、編集、または削除」を参照してください。	「DeletePracticeRunConfiguration」を参照してください。
オートシフトを一覧表示する	「ARC のゾーンオートシフト」を参照してください。	「ListAutoshifts」を参照してください。
ゾーンオートシフト用のリソースを一覧表示する	「サポートされているリソース」を参照してください。	「ListManagedResources」を参照
ゾーンオートシフト用のリソースを取得する	「サポートされているリソース」を参照してください。	「GetManagedResource」を参照
練習実行設定を編集する	「練習実行設定の設定、編集、または削除」を参照してください。	「UpdatePracticeRunConfiguration」を参照してください。
ゾーンオートシフトを有効化または無効化する	「ゾーンオートシフトの有効化または無効化」を参照してください。	「UpdateZonalAutoshiftConfiguration」を参照してください。
オートシフトのオブザーバー通知を有効または無効にする	「ゾーンオートシフトの有効化と操作」を参照してください。	「UpdateAutoshiftObserverNotificationStatus」を参照してください。
練習実行を開始する	「練習実行ゾーンシフトの開始」を参照してください。	「StartPracticeRun」を参照してください。
練習実行をキャンセルする	「練習実行のゾーンシフトのキャンセル」を参照	「CancelPracticeRun」を参照してください。

ゾーンオートシフト AWS CLI で を使用する例

このセクションでは、 を使用して、API オペレーションを使用して Amazon Application Recovery Controller (ARC) のゾーンオートシフト機能 AWS Command Line Interface を操作する、ゾーンオートシフトを使用する簡単なアプリケーション例について説明します。これらの例は、CLI を使用したゾーンオートシフトの操作方法の基本的な理解を深めることを目的としています。

ゾーンオートシフトは ARC の機能です。ゾーンオートシフトを使用すると、ユーザーに代わって、イベント中にサポートされているアプリケーションリソーストラフィック AWS をアベイラビリティーゾーンから遠ざけることを に許可し、復旧までの時間を短縮できます。ゾーンオートシフトで使用できるリソースの詳細については、「サポートされているリソース」を参照してください。

ゾーンオートシフトには、トラフィックをアベイラビリティーゾーンから遠ざける練習実行が含まれており、オートシフトがアプリケーションにとって安全であることを確認するのに役立ちます。

ゾーンオートシフト API アクションのリストと詳細情報へのリンクについては、「ゾーンオートシフト API オペレーション」を参照してください。の使用の詳細については AWS CLI、AWS CLI「 コマンドリファレンス」を参照してください。

内容

- 練習実行設定を作成する
- オートシフトを有効または無効にする
- オンデマンド練習実行を開始する
- 進行中の練習実行をキャンセルする
- 進行中のオートシフトをキャンセルする
- 練習実行設定を編集する
- 練習実行設定を削除する

練習実行設定を作成する

リソースのゾーンオートシフトを有効にする前に、リソースの練習実行設定を作成し、必要な練習実行のオプションを選択する必要があります。`create-practice-run-configuration` コマンドを使用して CLI でリソースの練習実行設定を作成します。

リソースの練習実行設定を作成するときは、次の点に注意してください。

- 現時点でサポートされているアラームタイプは `CLOUDWATCH` のみです。

- リソースがデプロイされている AWS リージョン のと同じ にあるアラームを使用する必要があります。
- 結果アラームを指定する必要があります。ブロッキングアラームの指定はオプションです。
- ブロックまたは許可する日付または時間枠の指定はオプションです。

create-practice-run-configuration コマンドを使用して CLI で練習実行設定を作成します。

例えば、リソースの練習実行設定を作成するには、次のようなコマンドを使用します。

```
aws arc-zonal-shift create-practice-run-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --outcome-alarms
 type=CLOUDWATCH,alarmIdentifier=arn:aws:cloudwatch:Region:111122223333:alarm:Region-
MyAppHealthAlarm \
      --blocking-alarms
 type=CLOUDWATCH,alarmIdentifier=arn:aws:cloudwatch:Region:111122223333:alarm:Region-
BlockWhenALARM \
      --blocked-dates 2023-12-01 --blocked-windows Mon:10:00-Mon:10:30
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "zonal-shift-elb"
   "zonalAutoshiftStatus": "DISABLED",
   "practiceRunConfiguration": {
       "blockingAlarms": [
                  {
              "type": "CLOUDWATCH",
              "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-BlockWhenALARM"
           }
       ]
       "outcomeAlarms": [
           {
               "type": "CLOUDWATCH",
               "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-MyAppHealthAlarm"
           }
       ],
       "blockedWindows": [
```

```
            "Mon:10:00-Mon:10:30"
        ],
        "blockedDates": [
            "2023-12-01"
        ]
}
```

オートシフトを有効または無効にする

リソースのオートシフトを有効または無効にするには、CLI でゾーンオートシフトのステータスを更新します。ゾーンオートシフトのステータスを変更するには、update-zonal-autoshift-configuration コマンドを使用します。

例えば、リソースのオートシフトを有効にするには、次のようなコマンドを使用します。

```
aws arc-zonal-shift update-zonal-autoshift-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --zonal-autoshift-status="ENABLED"
```

```
{
   "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
west-2:111122223333:ExampleALB123456890",
   "zonalAutoshiftStatus": "ENABLED"
}
```

オンデマンド練習実行を開始する

CLI で start-practice-run コマンドを使用して、オンデマンド練習実行のゾーンシフトを開始できます。

例えば、リソースの練習実行を開始するには、次のようなコマンドを使用します。

```
aws arc-zonal-shift start-practice-run
    --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
    "awayFrom": "usw2-az1",
```

```
{
    "awayFrom": "usw2-az1",
```

```
    "comment": "Practice run started. Shifting traffic away from Availability Zone
 usw2-az1.",
}
```

進行中の練習実行をキャンセルする

CLI で cancel-practice-run コマンドを使用すると、進行中の練習実行をキャンセルできます。

例えば、リソースの練習実行をキャンセルするには、次のようなコマンドを使用します。

```
aws arc-zonal-shift cancel-practice-run \
   --zonal-shift-id="="arn:aws:testservice::111122223333:ExampleALB123456890"
```

```
{
    "zonalShiftId": "2222222-3333-444-1111",
    "resourceIdentifier": "arn:aws:testservice::111122223333:ExampleALB123456890",
    "awayFrom": "usw2-az1",
    "expiryTime": 2024-11-15T10:35:42+00:00,
    "startTime": 2024-11-15T09:35:42+00:00,
    "status": "CANCELED",
    "comment": "Practice run canceled"
}
```

進行中のオートシフトをキャンセルする

リソースのゾーンオートシフトをキャンセルすると、CLI で進行中のオートシフトをキャンセルできます。ゾーンオートシフトをキャンセルするには、cancel-zonal-shift command を使用します。

```
aws arc-zonal-shift cancel-zonal-shift --zonal-shift-id
 9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38
```

```
{
    "awayFrom": "usw2-az1",
    "comment": "Zonal autoshift started. Shifting traffic away from Availability Zone
 usw2-az1.",
    "expiryTime": "2024-12-17T22:29:38-08:00",
    "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
east-1:111122223333:loadbalancer/app/Testing/5a19403ecd42dc05",
    "startTime": "2024-12-17T21:27:26-08:00",
    "status": "CANCELED",
```

```
    "zonalShiftId": "9ac9ec1e-1df1-0755-3dc5-8cf573cd9c38"
}
```

練習実行設定を編集する

ARC が練習実行を開始しない場合、CLI を使用してリソースの練習実行設定を編集し、さまざまな設定オプションを更新できます。例えば、練習実行のアラームを変更したり、ブロックされた日付やブロックされた時間枠を更新したりできます。練習実行設定を編集するには、update-practice-run-configuration コマンドを使用します。

リソースの練習実行設定を編集するときには、次の点に注意してください。

- 現時点でサポートされているアラームタイプは CLOUDWATCH のみです。
- リソースがデプロイされている AWS リージョン のと同じ にあるアラームを使用する必要があります。
- 結果アラームを指定する必要があります。ブロッキングアラームの指定はオプションです。
- ブロックする日付またはブロックする時間枠の指定はオプションです。
- ブロックする日付またはブロックする時間枠を指定すると、、既存の値は置き換えられます。

例えば、リソースの練習実行設定を編集して、新しいブロックする日付を指定するには、次のようなコマンドを使用します。

```
aws arc-zonal-shift update-practice-run-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --blocked-dates 2024-03-01
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "zonal-shift-elb"
   "zonalAutoshiftStatus": "DISABLED",
   "practiceRunConfiguration": {
       "blockingAlarms": [
                  {
              "type": "CLOUDWATCH",
              "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-BlockWhenALARM"
           }
       ]
```

```
        "outcomeAlarms": [
            {
                "type": "CLOUDWATCH",
                "alarmIdentifier": "arn:aws:cloudwatch:us-west-2:111122223333:alarm:us-
west-2-MyAppHealthAlarm"
            }
        ],
        "blockedWindows": [
            "Mon:10:00-Mon:10:30"
        ],
        "blockedDates": [
            "2024-03-01"
        ]
}
```

練習実行設定を削除する

リソースの練習実行設定を削除できますが、まず、リソースのゾーンオートシフトを無効にする必要があります。ゾーンオートシフトを有効にするには、リソースに練習実行設定が必要です。定期的な練習実行により、1 つのアベイラビリティーゾーンがなくてもアプリケーションが正常に動作することを確認できます。

CLI を使用して練習実行設定を削除するには、まず、必要に応じて update-zonal-autoshift コマンドを使用してゾーンオートシフトを無効にします。次に、練習実行設定を削除するには、delete-practice-run-configuration コマンドを使用します。

まず、次のようなコマンドを使用して、リソースのゾーンオートシフトを無効にします。

```
aws arc-zonal-shift update-zonal-autoshift-configuration \
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890" \
      --zonal-autoshift-status="DISABLED"
```

```
{
   "resourceIdentifier": "arn:aws:elasticloadbalancing:us-
west-2:111122223333:ExampleALB123456890",
   "zonalAutoshiftStatus": "DISABLED"
}
```

次に、次のようなコマンドを使用して、練習実行設定を削除します。

```
aws arc-zonal-shift delete-practice-run-configuration \
```

```
      --resource-
identifier="arn:aws:elasticloadbalancing:Region:111122223333:ExampleALB123456890"
```

```
{
   "arn": "arn:aws:elasticloadbalancing:us-west-2:111122223333:ExampleALB123456890",
   "name": "TestResource",
   "zonalAutoshiftStatus": "DISABLED"
}
```

ゾーンオートシフトの有効化と操作

このセクションでは、Amazon Application Recovery Controller (ARC) でゾーンオートシフトを操作する手順について説明します。ゾーンオートシフトを有効にすると、練習実行設定の変更、オンデマンド練習実行の開始、練習実行を含む進行中シフトのキャンセル、またはオートシフトのオブザーバー通知の有効化を実行できます。

ゾーンオートシフトの有効化または無効化

この手順では、Amazon Application Recovery Controller (ARC) のコンソールからゾーンオートシフトを有効化または無効化する手順について説明します。ゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

ゾーンオートシフトを有効にすると、 がユーザーに代わってイベント中にアプリケーションリソーストラフィックをアベイラビリティーゾーンから遠ざけることを承認 AWS し、復旧までの時間を短縮できます。

ゾーンオートシフトを有効化または無効化するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/autoshift で ARC コンソールを開きます。
2. [リソースのゾーンオートシフト設定] で、リソースを選択します。
3. [アクション] メニューで、[ゾーンオートシフトを有効化] を選択してから、手順に従って更新を完了します。

リソースに練習実行設定がない場合、[ゾーンオートシフトを有効化] は使用できません。練習実行設定を構成して、ゾーンオートシフトを有効にするには、[ゾーンオートシフトを設定] を選択します。

内容

- 練習実行設定の設定、編集、または削除
- ゾーンオートシフトのキャンセル
- 練習実行ゾーンシフトの開始
- 練習実行のゾーンシフトのキャンセル
- オートシフトのオブザーバー通知の有効化または無効化

練習実行設定の設定、編集、または削除

このセクションの手順では、Amazon Application Recovery Controller (ARC) のコンソールから練習実行設定を編集または削除する手順について説明します。ゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

コンソールで練習実行設定を削除すると、ゾーンオートシフトは無効になります。API オペレーションで練習実行設定を削除するには、その前に、ゾーンオートシフトを無効にする必要があります。ゾーンオートシフトを有効にしなくても練習実行を設定できます。ただし、ゾーンオートシフトがリソースについて有効であるためには、そのリソースに対して練習実行を設定してある必要があります。

練習実行を設定するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/autoshift で ARC コンソールを開きます。
2. [ゾーンオートシフトを設定] を選択します。
3. ゾーンオートシフトを設定するリソースを選択します。
4. AWS イベントが発生したときにリソースのオートシフト AWS を開始しない場合は、ゾーンオートシフトを無効にすることを選択します。必要に応じて、ウィザードを続行して、オートシフトを有効にせずに練習実行設定を構成できます。
5. リソースの練習実行のオプションを選択します。アラームの場合は、以下のことができます。

   - (必須) このリソースの練習実行を監視する結果アラームを少なくとも 1 つ指定します。
   - (オプション) このリソースの練習実行のブロッキングアラームを 1 つまたは複数指定します。

   詳細については、「ゾーンオートシフトを設定する際のベストプラクティス」の「練習実行について指定するアラーム」セクションを参照してください。

6. オプションとして、ブロックされた時間枠または許可された時間枠を指定して、ARC による練習実行の開始をブロック、または ARC によるこのリソースの練習実行の開始を許可します。すべての日付と時刻は UTC で表示されます。
7. チェックボックスを選択して、確認メモを読んだことを確認します。
8. [作成] を選択します。

練習実行設定を編集するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/autoshift で ARC コンソールを開きます。
2. [リソースのゾーンオートシフト設定] で、リソースを選択します。
3. [アクション] メニューで、[練習実行設定を編集] を選択します。
4. 練習実行設定に変更を加えて、次の 1 つ以上の操作を行います。

   - 例えば、以下のことができます。
     - ブロッキングアラームでは、1 つまたは複数のアラームを追加、またはアラームを削除できます。
     - 結果アラームでは、1 つまたは複数のアラームを追加、またはアラームを削除できます。少なくとも 1 つの結果アラームが必須であるため、設定内のすべての結果アラームを削除することはできません。
   - ブロックされる時間枠や許可される時間枠については、新しい日付や曜日と時刻を追加したり、既存の日付や曜日と時刻を削除または更新したりできます。すべての日付と時刻は UTC で表示されます。
5. [保存] を選択します。

練習実行設定を削除するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/autoshift で ARC コンソールを開きます。
2. [リソースのゾーンオートシフト設定] で、リソースを選択します。
3. [アクション] メニューで、[練習実行設定を削除] を選択します。
4. 確認ダイアログボックスで、`Delete` と入力し、[削除] を選択します。

   コンソールで練習実行設定を削除すると、リソースのゾーンオートシフトも無効になることに注意してください。ゾーンオートシフトでは、リソースの練習実行を設定する必要があります。

ゾーンオートシフトのキャンセル

リソースの進行中のゾーンオートシフトを停止するには、ゾーンオートシフトをキャンセルする必要があります。

進行中のゾーンオートシフトを停止するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/ で ARC コンソールを開きます。
2. キャンセルするゾーンオートシフトを選択してから、[ゾーンシフトをキャンセル] を選択します。
3. ダイアログボックスで、[確認] を選択します。

練習実行ゾーンシフトの開始

このセクションの手順では、ARC コンソールでオンデマンドの練習実行ゾーンシフトを開始する方法について説明します。ゾーンシフトとゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

ゾーンオートシフトを設定してから練習実行設定を作成すると、練習実行ゾーンシフトを開始できます。

練習実行ゾーンシフトを開始するには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/ で ARC コンソールを開きます。
2. [ゾーンオートシフトのリソース] で、ゾーンオートシフトが設定された個々のリソースを参照します。
3. [リソース概要] ページで、[練習実行を開始する] を選択します。
4. アベイラビリティーゾーンを選択してから、練習実行のコメントを入力します。練習実行は、選択したアベイラビリティーゾーンからトラフィックを遠ざけます。
5. [開始] を選択します。

練習実行のゾーンシフトのキャンセル

このセクションの手順では、ARC コンソールでゾーンシフトをキャンセルする方法について説明します。ゾーンシフトとゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

自分で開始したゾーンシフトまたは練習実行はキャンセルできます。ゾーンオートシフトの練習実行のリソースに対して AWS 開始されるゾーンシフトをキャンセルすることもできます。

練習実行のゾーンシフトをキャンセルするには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/ で ARC コンソールを開きます。
2. キャンセルする練習実行のゾーンシフトを選択してから、[ゾーンシフトをキャンセル] または [練習実行をキャンセル] を選択します。
3. ダイアログボックスで、[確認] を選択します。

オートシフトのオブザーバー通知の有効化または無効化

がオートシフト AWS を開始して、障害の可能性があるアベイラビリティーゾーンからトラフィックを遠ざけるたびに、Amazon EventBridge を介して通知するようにゾーンオートシフトを設定できます。このオプションは、通知 AWS リージョン を受信する各 で設定する必要があります。これらの個別の通知を有効にするために、ゾーンオートシフトで特定のリソースを設定する必要はありません。詳細については、「Amazon EventBridge でのゾーンオートシフトの使用」を参照してください。

このセクションの手順では、Amazon Application Recovery Controller (ARC) のコンソールを使用してオートシフトのオブザーバー通知を有効化する手順について説明します。ゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

オートシフトのオブザーバー通知を有効または無効にするには

1. https://console.aws.amazon.com/route53recovery/zonalshift/home#/ で ARC コンソールを開きます。
2. [利用開始] で、[オートシフトオブザーバー通知を有効にする] を選択します。
3. 確認ダイアログボックスで、[オブザーバー通知を有効にする] を選択します。

を使用したゾーンオートシフトのテスト AWS FIS

AWS Fault Injection Service を使用して、AZ アベイラビリティー: 電源中断シナリオなどの実際の条件をシミュレートする実験をセットアップして実行できます。このシナリオは、AZ の障害が広範囲にまたがる可能性がある場合に、オートシフトが有効なリソースでゾーンオートシフトを開始したときに AWS 何が起こるかを示します。

`aws:arc:start-zonal-autoshift` 復旧の開始アクションを使用すると、 AWS がゾーンオートシフトが有効なリソースのトラフィックを、障害の可能性がある AZ から自動的に移行し、AZs 可用性シナリオの実行 AWS リージョン 中に同じ 内の正常な AZ に再ルーティングする方法を示すことができます。

たとえば、 AWS FIS シナリオライブラリを使用して、停電によって発生した AZ の障害をシミュレートできます。この実験では、AZ の電源中断が始まってから 5 分後に、復旧アクション `aws:arc:start-zonal-autoshift` によってリソーストラフィックが指定の AZ から自動的にシフトします。AZ の障害が広範囲に及ぶ可能性がある場合にオートシフトがどのようにトリガーされるかを示すため、トラフィックは停電の残りの 25 分間シフトされます。実験が完了すると、トラフィックのシフトが終了し、トラフィックはすべての AZ に再び流れ始めます。このプロセスは、AZ に影響を与える電力イベントからの完全な復旧を示します。

実験とゾーンオートシフト練習実行の違い

AWS FIS 実験は、通常のプロセスの一環として ARC がリソースのトラフィックを 1 つの AZ から移行し、アプリケーションが AZ の損失を許容できるという点で、ゾーンオートシフトの練習実行とは異なります。ただし、実験中に AWS FIS 、 はユーザーに代わってオートシフトが有効なリソースに対して AZ の障害とオートシフトがどのようにトリガーされるか AWS FIS を示し、障害が解決されるとオートシフトをキャンセルします。

実行中に AWS FIS 開始ゾーンシフトを更新することはできません。さらに、 の外部でゾーンシフトをキャンセルすると AWS FIS、 AWS FIS 実験は終了します。

AWS FIS 有効期限ベースの安全メカニズム

AWS FIS は、StartZonalShift、UpdateZonalShift、および CancelZonalShift API オペレーションを使用してゾーンシフトを管理します。これらのリクエストの `expiresIn`フィールドは安全メカニズムとして 1 分に設定されています。これにより AWS FIS 、ネットワークの停止やシステムの問題などの予期しないイベントが発生した場合に、 はゾーンシフトを迅速にロールバックできます。ARC コンソールでは、有効期限フィールドが AWS FIS管理され、実際の予想される有効期限はゾーンシフ

トアクションで指定された期間によって決まります。練習実行の詳細については、「ゾーンオートシフトと練習実行の仕組み」を参照してください。

一度に適用できるゾーンシフトは 1 つのみです。つまり、1 つの練習のみが、リソースに対してゾーンシフト、お客様が開始したゾーンシフト、オートシフト、または AWS FIS 実験を実行します。2 番目のゾーンシフトが開始されると、ARC は優先順位に従いリソースに有効なゾーンシフトタイプを判断します。ゾーンシフトの優先順位の詳細については、「ゾーンシフトの優先順位」を参照してください。

AWS FIS 復旧アクションの詳細については、「 AWS Fault Injection Service ユーザーガイド」のAWS FIS 「復旧アクション」を参照してください。

Amazon Application Recovery Controller (ARC) のゾーンオートシフトのログ記録とモニタリング

AWS CloudTrail と Amazon EventBridge を使用して、Amazon Application Recovery Controller (ARC) でゾーンオートシフトをモニタリングし、パターンを分析し、問題のトラブルシューティングに役立てることができます。

トピック

- AWS CloudTrail を使用したゾーンオートシフト API コールのログ記録
- Amazon EventBridge でのゾーンオートシフトの使用

AWS CloudTrail を使用したゾーンオートシフト API コールのログ記録

ARC のゾーンオートシフトは、AWS CloudTrail と統合されています。これは、ARC のユーザー、ロール、または AWS のサービスで実行されたアクションを記録するためのサービスです。CloudTrail は、ゾーンシフトに対するすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、ARC コンソールからのコールと、ゾーンシフトの ARC API オペレーションへのコードコールが含まれます。

証跡を作成する場合は、ゾーンシフトのイベントを含む、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。

CloudTrail で収集した情報を使用して、ゾーンシフトの ARC に対するリクエスト、リクエスト元の IP アドレス、リクエストした人、リクエストが行われた日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail のゾーンオートシフト情報

CloudTrail は、AWS アカウントを作成すると、その中で有効になります。ゾーンオートシフトの ARC でアクティビティが発生すると、そのアクティビティは他の AWS サービスイベントと共に、[イベント履歴] 内で CloudTrail イベントに記録されます。最近のイベントは、AWS アカウントで表示、検索、ダウンロードできます。詳細については、「CloudTrail イベント履歴の操作」を参照してください。

ゾーンオートシフトの ARC のイベントなど、AWS アカウントで継続しているイベントの記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づいて対応するため、他の AWS サービスを構成できます。詳細については、次を参照してください:

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての ARC アクションは CloudTrail によってログに記録され、「Amazon Application Recovery Controller のルーティングコントロール API リファレンスガイド」に記載されています。例えば、`StartZonalShift` および `ListManagedResources` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、リクエストの生成者に関する情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

イベント履歴での ARC イベントの表示

CloudTrail では、[イベント履歴] に最近のイベントが表示されます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。

ゾーンオートシフトログファイルエントリについて

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどの情報を含みます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、ゾーンオートシフトの `ListManagedResources` アクションを実行する CloudTrail ログエントリです。


```
{
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AROA33L3W36EXAMPLE",
            "arn": "arn:aws:iam::111122223333:role/admin",
            "accountId": "111122223333",
            "userName": "EXAMPLENAME"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2022-11-14T16:01:51Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2022-11-14T16:14:41Z",
      "eventSource": "arc-zonal-shift.amazonaws.com",
```

```
      "eventName": "ListManagedResources",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.50",
      "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
      "requestParameters": null,
      "responseElements": null,
      "requestID": "VGXG4ZUE7UZTVCMTJGIAF_EXAMPLE",
      "eventID": "4b5c42df-1174-46c8-be99-d67_EXAMPLE",
      "readOnly": true,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "111122223333"
      "eventCategory": "Management"
      }
    }
```

Amazon EventBridge でのゾーンオートシフトの使用

Amazon EventBridge を使用すると、ゾーンオートシフトリソースをモニタリングし、他の AWS サービスを使用するターゲットアクションを開始するイベント駆動型ルールを設定できます。例えば、ゾーンオートシフトの練習実行が開始されたとき、Amazon SNS トピックに信号を伝達すると、E メール通知を送信するルールを設定できます。

Amazon EventBridge でゾーンオートシフトに関するルールを作成できます。ゾーンオートシフトのイベントは、練習実行が開始するときなどに、練習実行またはオートシフトに関するステータス情報を指定します。ゾーンオートシフトを設定して、サービスに対して有効にしたリソースのゾーンオートシフトイベントを通知できます。

他の通知に加えて、または他の通知の代わりに、オートシフトオブザーバー通知を有効にすることもできます。オートシフトオブザーバー通知は、 がアベイラビリティーゾーンのオートシフト AWS を開始するたびに通知イベントを提供します。オートシフトのオブザーバー通知は、ゾーンオートシフトで有効にしたリソースのトラフィックがアベイラビリティーゾーンから遠ざけられたときに受け取る通知とは異なります。オートシフトのオブザーバー通知を有効にするために、ゾーンオートシフトでリソースを設定する必要はありません。詳細については、「ゾーンオートシフトの有効化と操作」を参照してください。

関心のある特定のゾーンオートシフトイベントをキャプチャするには、そのイベント固有のパターンを定義し、イベントを検出する EventBridge がそれを使用できるようにします。イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

イベントは、ベストエフォートベースで出力されます。通常の運用状況下では、ARC から EventBridge にほぼリアルタイムで配信されます。ただし、イベントの配信を遅らせたり妨げたりする状況が発生する場合もあります。

EventBridge ルールがイベントパターンでどのように機能するかについては、「EventBridge のイベントとイベントパターン」を参照してください。

EventBridge でゾーンオートシフトリソースをモニタリングする

EventBridge でルールを作成すると、ARC がリソースに対してイベントを発生したときに実行されるアクションを定義できます。例えば、練習実行がゾーンオートシフトを開始したときに E メールメッセージを送信するルールを作成できます。

EventBridge コンソールにイベントパターンを入力またはコピーするには、コンソールの [独自のサンプルイベントを入力] オプションを選択します。有用なイベントパターンを決める際の参考になるように、このトピックでは、使用できるゾーンオートシフトイベントマッチングパターンとゾーンオートシフトイベントの両方の例を紹介します。

リソースイベントのルールを作成するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. ルール AWS リージョン を作成する 、つまりイベントを視聴するリージョンを選択します。
3. [Create rule] を選択します。
4. ルールの [Name (名前)] を入力し、必要に応じて説明を入力します。
5. [イベントバス] については、デフォルト値の [デフォルト] のままにします。
6. [次へ] を選択します。
7. [イベントパターンを構築] ステップでは、[イベントソース] はデフォルト値の [AWS イベント] のままにします。
8. [サンプルイベント] で [独自のサンプルイベントを入力] を選択します。
9. [サンプルイベント] には、イベントパターンを入力するか、コピーして貼り付けます。

ゾーンオートシフトイベントパターンの例

イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

このセクションのイベントパターンをコピーして EventBridge に貼り付け、ゾーンオートシフトのアクションとリソースの監視に使用できるルールを作成できます。

ゾーンオートシフトイベントのイベントパターンを作成するときには、detail-type に以下のいずれかを指定できます。

- Autoshift In Progress
- Autoshift Completed
- Practice Run Started
- Practice Run Succeeded
- Practice Run Interrupted
- Practice Run Failed
- FIS Experiment Autoshift In Progress
- FIS Experiment Autoshift Completed
- FIS Experiment Autoshift Canceled
- Manual Shift Started
- Manual Shift Updated
- Manual Shift Canceled

練習実行が中断されたとき、中断の原因について詳しくは、additionalFailureInfo フィールドを参照してください。

AWS Autoshift オブザーバー通知を有効にすることで、すべての Autoshift をモニタリングすることを選択できます。オートシフトのオブザーバー通知を有効にした後、通知を受信するには、ゾーンオートシフトの詳細タイプ Autoshift In Progress の通知を受け取ることを選択します。オートシフトのオブザーバー通知を有効にする手順については、「ゾーンオートシフトの有効化と操作」を参照してください。

例については、「ゾーンオートシフトイベントの例」セクションを参照してください。

- オートシフトが開始されたゾーンオートシフトからすべてのイベントを選択します。

  次の点に注意してください。

  - オートシフトのオブザーバー通知を有効にした場合、ARC はすべてのオートシフトイベントを返します。

- オートシフトのオブザーバー通知を有効にしていない場合、ゾーンオートシフト用に設定したリソースがオートシフトに含まれている場合にのみ、ARC はオートシフトイベントを返します。

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Autoshift In Progress"
    ]
}
```

- 練習実行が開始されたゾーンオートシフトからすべてのイベントを選択します。

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Practice Run Started"
    ]
}
```

- 練習実行が失敗したゾーンオートシフトからすべてのイベントを選択します。

```
{
    "source": [
        "aws.arc-zonal-shift"
    ],
    "detail-type": [
        "Practice Run Failed"
    ]
}
```

ゾーンオートシフトイベントの例

このセクションでは、ゾーンオートシフトアクションのイベント例について説明します。

以下は、`Autoshift In Progress` アクションのイベント例です。1) オートシフトのオブザーバー通知が有効である場合。2) オートシフトに含まれるゾーンオートシフトでリソースを設定していない場合。

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":"AWS has started an autoshift for an impaired Availability Zone.
 This notification
            is separate from autoshift notifications for resources, if any, that you
 have configured for
            zonal autoshift. For details, see the Developer Guide."
        }
    }
}
```

以下は、Autoshift In Progress アクションのイベント例です。1) オートシフトのオブザーバー通知が無効である場合。2) オートシフトに含まれるゾーンオートシフトでリソースを設定している場合。

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
```

```
            "awayFrom": "use1-az2",
            "notes":""
        }
    }
}
```

以下は、Practice Run Interrupted アクションのイベント例です。

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Practice Run Interrupted",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
        "data": {
            "additionalFailureInfo": "Practice run interrupted. The blocking alarm
 entered ALARM state."
        },
        "metadata": {
            "awayFrom": "use1-az2"
        }
    }
}
```

以下は、FIS Experiment Autoshift In Progress アクションのイベント例です。

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "FIS Experiment Autoshift In Progress",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
```

```
    ],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":""
        }
    }
}
```

以下は、 Manual Shift Startedアクションのイベント例です。これは、StartZonalShiftAPI がリソースで呼び出されたときに出力されます。

```
{
    "version": "0",
    "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4",
    "detail-type": "Manual Shift Started",
    "source": "aws.arc-zonal-shift",
    "account": "111122223333",
    "time": "2023-11-16T23:38:14Z",
    "region": "us-east-1",
    "resources": [
        "TEST-EXAMPLE-2023-11-16-23-28-11-5"
    ],
    "detail": {
        "version": "0.0.1",
        "data": "",
        "metadata": {
            "awayFrom": "use1-az2",
            "notes":""
        }
    }
}
```

ターゲットとして使用する CloudWatch ロググループを指定する

EventBridge ルールを作成するときは、ルールに一致するイベントの送信先のターゲットを指定する必要があります。EventBridge で利用できるターゲットの一覧については、「EventBridge コンソールで使用可能なターゲット」を参照してください。EventBridge ルールに追加できるターゲットの 1 つは、Amazon CloudWatch ロググループです。このセクションでは、CloudWatch ロググループを

ターゲットとして追加するための要件と、ルール作成時にロググループを追加する手順について説明します。

CloudWatch ロググループをターゲットとして追加するには、次のいずれかを実行します。

- 新しいロググループを作成する
- 既存のロググループを選択する

ルールの作成時にコンソールを使用して新しいロググループを指定する場合は、EventBridge によって自動的にロググループが作成されます。EventBridge ルールのターゲットとして使用するロググループが `/aws/events` で始まることを確認します。既存のロググループを選択する場合は、`/aws/events` で始まるロググループのみがドロップダウンメニューのオプションとして表示されることに注意してください。詳細については、「Amazon CloudWatch ユーザーガイド」の「新しいロググループを作成する」を参照してください。

コンソール外で CloudWatch オペレーションを使用して CloudWatch ロググループを作成またはターゲットとして使用する場合は、アクセス許可を正しく設定していることを確認してください。コンソールを使用して EventBridge ルールにロググループを追加すると、ロググループのリソースベースのポリシーが自動的に更新されます。ただし、 AWS Command Line Interface または AWS SDK を使用してロググループを指定する場合は、ロググループのリソースベースのポリシーを更新する必要があります。次のポリシー例は、ロググループのリソースベースのポリシーで定義する必要があるアクセス許可を示しています。

JSON

```
{
    "Version":"2012-10-17",

    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com",
                    "delivery.logs.amazonaws.com"
```

```
                ]
            },
            "Resource": "arn:aws:logs:us-east-1:222222222222:log-group:/aws/
events/*:*",
            "Sid": "TrustEventsToStoreLogEvent"
        }
    ]
}
```

コンソールを使用してロググループのリソースベースのポリシーを設定することはできません。必要なアクセス許可をリソースベースのポリシーに追加するには、CloudWatch PutResourcePolicy API オペレーションを使用します。次に、describe-resource-policies CLI コマンドを使用して、ポリシーが正しく適用されたことを確認できます。

リソースイベントのルールを作成して CloudWatch ロググループターゲットを指定するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. ルール AWS リージョン を作成する を選択します。
3. [ルールの作成] を選択して、イベントパターンまたはスケジュールの詳細など、そのルールに関する情報を入力します。

   ARC の EventBridge ルール作成の詳細については、このトピックの前半のセクションを参照してください。
4. [ターゲットを選択] ページで、ターゲットとして [CloudWatch] を選択します。
5. ドロップダウンメニューから CloudWatch ロググループを選択します。

ARC でのゾーンオートシフトの Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

内容

- ARC のゾーンオートシフトを IAM と連携させる方法
- ARC でのゾーンオートシフトのアイデンティティベースのポリシー例

- ARC でのゾーンオートシフトのサービスにリンクされたロールの使用
- AWS ARC でのゾーンオートシフトの マネージドポリシー

ARC のゾーンオートシフトを IAM と連携させる方法

IAM を使用して Amazon Application Recovery Controller (ARC) のゾーンオートシフトへのアクセスを管理する前に、ゾーンオートシフトで使用できる IAM 機能について説明します。

ARC のゾーンオートシフトで使用できる IAM の機能

IAM 機能	ゾーンオートシフトのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	はい

AWS サービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

ARC のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

ARC のアイデンティティベースのポリシーの例を表示するには、「Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例」を参照してください。

ARC 内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。

ARC のポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

ゾーンオートシフトにおける ARC アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Route 53 - ゾーンシフトで定義されるアクション」を参照してください。

ARC におけるゾーンオートシフト用ポリシーのアクションでは、アクションの前に次のプレフィックスを使用しています。

```
arc-zonal-shift
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。例えば、次のようになります。

```
"Action": [
      "arc-zonal-shift:action1",
      "arc-zonal-shift:action2"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Describe という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "arc-zonal-shift:Describe*"
```

ゾーンオートシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンオートシフトのアイデンティティベースのポリシー例」を参照してください。

ARC のゾーンオートシフトのポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

リソースタイプとその ARN のリスト、および各リソースの ARN で指定できるアクションについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義されるアクション

条件キーで使用できるアクションとリソースについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

ゾーンオートシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンオートシフトのアイデンティティベースのポリシー例」を参照してください。

ARC のゾーンオートシフトのポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

ARC の条件キーのリストについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 ゾーンシフトの条件キー

条件キーで使用できるアクションとリソースについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 ゾーンシフトで定義されるアクション

ゾーンオートシフトにおける ARC のアイデンティティベースのポリシーの例を表示するには、「ARC でのゾーンオートシフトのアイデンティティベースのポリシー例」を参照してください。

ARC のアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセスコントロール (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

ARC のゾーンオートシフトには、ABAC に対する以下の部分的なサポートが含まれています。

- ゾーンオートシフトは、ゾーンシフト用に ARC に登録されている、マネージドリソースの ABAC をサポートしています。Network Load Balancer と Application Load Balancer マネージドリソースにおける ABAC の詳細については、「Elastic Load Balancing ユーザーガイド」の「Elastic Load Balancing での ABAC」を参照してください。

ARC での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。長期的なアクセスキーを使用する代わりに、一時的

な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

ARC のクロスサービスプリンシパル権限

転送アクセスセッション (FAS) のサポート: あり

IAM エンティティ (ユーザーまたはロール) を使用して でアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。

アクションにポリシーで追加の依存アクションが必要かどうかを確認するには、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 ゾーンシフト

ARC のサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

ARC のサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

ARC サービスにリンクされたロールの作成または管理の詳細については、「ARC でのゾーンオートシフトのサービスにリンクされたロールの使用」を参照してください。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

ARC でのゾーンオートシフトのアイデンティティベースのポリシー例

デフォルトでは、ユーザーおよびロールには、ARC リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認証リファレンス」の「Amazon Application Recovery Controller (ARC) のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- 例: ゾーンオートシフトコンソールアクセス
- 例: ARC API アクション

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが ARC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。

- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の 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。

例: ゾーンオートシフトコンソールアクセス

Amazon Application Recovery Controller (ARC) コンソールにアクセスするには、最小限のアクセス許可セットが必要です。これらのアクセス許可により、 の ARC リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

一部のタスクを実行するには、ユーザーは ARC のゾーンオートシフトに関連付けられているサービスにリンクされたロールを作成するアクセス許可を持っている必要があります。詳細については[ARC でのゾーンオートシフトのサービスにリンクされたロールの使用]を参照してください。

でゾーンオートシフトを使用するためのフルアクセスをユーザーに付与するには AWS マネジメントコンソール、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
                   "arc-zonal-shift:UpdateZonalShift",
                   "arc-zonal-shift:CancelZonalShift",
                   "arc-zonal-shift:CreatePracticeRunConfiguration",
                   "arc-zonal-shift:DeletePracticeRunConfiguration",
                   "arc-zonal-shift:ListAutoshifts",
                   "arc-zonal-shift:UpdatePracticeRunConfiguration",
                   "arc-zonal-shift:UpdateZonalAutoshiftConfiguration"
             ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeAvailabilityZones",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "cloudwatch:DescribeAlarms",
            "Resource": "*"
        }
    ]
}
```

例: ARC API アクション

ポリシーを使用して、ユーザーがゾーンオートシフトの ARC API アクションを使用してゾーンオートシフトを設定し、 がユーザーに代わってアプリケーションリソーストラフィックをアベイラビリティーゾーンから の正常な AZs AWS にシフトして AWS リージョン、イベント中の復旧時間を短縮

できるようにします。これらのアクセス許可を付与するには、以下で説明するように、ユーザーが操作する必要がある API オペレーションに対応するポリシーをアタッチします。

一部のタスクを実行するには、ユーザーは ARC に関連付けられているサービスにリンクされたロールのアクセス許可を持っている必要があります。サービスにリンクされたロールの作成に必要なアクセス許可は、次のポリシー例に含まれています。詳細については[ARC でのゾーンオートシフトのサービスにリンクされたロールの使用]を参照してください。

ゾーンオートシフト用の API オペレーションを使用するには、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "arc-zonal-shift:ListManagedResources",
                   "arc-zonal-shift:GetManagedResource",
                   "arc-zonal-shift:ListZonalShifts",
                   "arc-zonal-shift:StartZonalShift",
                   "arc-zonal-shift:UpdateZonalShift",
                   "arc-zonal-shift:CancelZonalShift",
                   "arc-zonal-shift:CreatePracticeRunConfiguration",
                   "arc-zonal-shift:DeletePracticeRunConfiguration",
                   "arc-zonal-shift:ListAutoshifts",
                   "arc-zonal-shift:UpdatePracticeRunConfiguration",
                   "arc-zonal-shift:UpdateZonalAutoshiftConfiguration"
             ],
            "Resource": "*"
        },
        {
            "Effect" : "Allow",
            "Action" : [
                   "cloudwatch:DescribeAlarms",
                   "health:DescribeEvents"
            ],
            "Resource" : "*"
        },
        {
```

```
            "Effect" : "Allow",
            "Action" : [
                    "arc-zonal-shift:CancelZonalShift",
                    "arc-zonal-shift:GetManagedResource",
                    "arc-zonal-shift:StartZonalShift",
                    "arc-zonal-shift:UpdateZonalShift"
            ],
            "Resource" : "*"
        }
    ]
}
```

ARC でのゾーンオートシフトのサービスにリンクされたロールの使用

Amazon Application Recovery Controller のゾーンオートシフトは、 AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用します。サービスにリンクされたロールは、サービスに直接リンクされた一意のタイプの IAM ロールです。この場合は ARC です。サービスにリンクされたロールは ARC によって事前定義されており、特定の目的でサービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、ARC の設定が簡単になります。ARC はサービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、ARC のみがそのロールを引き受けることができます。定義されるアクセス許可には、信頼ポリシーと許可ポリシーが含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、まずその関連リソースを削除します。これにより、リソースへのアクセス許可を誤って削除できないため、ARC ゾーンオートシフトリソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスの詳細については、AWS 「IAM と連携するサービス」を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、はいリンクを選択します。

AWSServiceRoleForZonalAutoshiftPracticeRun のサービスリンクロールアクセス許可

ARC は、AWSServiceRoleForZonalAutoshiftPracticeRun という名前のサービスにリンクされたロールを使用して以下を実行します。

- お客様が用意した Amazon CloudWatch アラームと顧客 Health Dashboard イベントをモニタリングして、練習実行を行います。
- 練習実行 (練習のゾーンシフト) を管理します。

このセクションでは、サービスリンクロールのアクセス許可と、ロールの作成、編集、および削除に関して説明します。

AWSServiceRoleForZonalAutoshiftPracticeRun のサービスリンクロールアクセス許可

このサービスリンクロールは、マネージドポリシーである `AWSZonalAutoshiftPracticeRunSLRPolicy` を使用します。

AWSServiceRoleForZonalAutoshiftPracticeRun サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `practice-run.arc-zonal-shift.amazonaws.com`

このポリシーのアクセス許可を確認するには、「 AWS マネージドポリシーリファレンス」の[AWSZonalAutoshiftPracticeRunSLRPolicy」を参照してください。

サービスリンク役割の作成、編集、削除を IAM エンティティ (ユーザー、グループ、役割など) に許可するにはアクセス許可を設定する必要があります。詳細については、「IAM User Guide」(IAM ユーザーガイド) の[「Service-linked role permissions」](サービスにリンクされたロールのアクセス権限) を参照してください。

ARC 用の AWSServiceRoleForZonalAutoshiftPracticeRun サービスリンクロールの作成

AWSServiceRoleForZonalAutoshiftPracticeRun サービスリンクロールを手動で作成する必要はありません。 AWS マネジメントコンソール、、 AWS CLIまたは AWS SDK で最初の練習実行設定を作成すると、ARC によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は同じ方法でアカウントにロールを再作成できます。最初の練習実行設定を作成すると、ARC はサービスにリンクされたロールを再度作成します。

ARC の AWSServiceRoleForZonalAutoshiftPracticeRun サービスにリンクされたロールの編集

ARC では、AWSServiceRoleForZonalAutoshiftPracticeRun サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、他のエンティティがロールを参照する可能

性があるため、ロールの名前を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」を参照してください。

ARC の AWSServiceRoleForZonalAutoshiftPracticeRun サービスにリンクされたロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、モニタリングや保守が積極的に行われていない未使用のエンティティを排除できます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

オートシフトを無効にした後、AWSServiceRoleForZonalAutoshiftPracticeRun サービスリンクロールを削除できます。オートシフト機能の詳細については、「ARC のゾーンシフト」を参照してください。

> **Note**
>
> リソースを削除しようとしたときに ARC サービスがロールを使用している場合、サービスロールの削除が失敗する可能性があります。失敗した場合は、数分待ってからロールの削除をもう一度試してください。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、 AWS CLI、または AWS API を使用して、AWSServiceRoleForZonalAutoshiftPracticeRun サービスにリンクされたロールを削除します。詳細については、 IAM ユーザーガイド の「サービスにリンクされたロールの削除」を参照してください。

ゾーンオートシフトの ARC サービスにリンクされたロールの更新

ARC サービスにリンクされたロールの AWS マネージドポリシーの更新については、ARC の AWS マネージドポリシーの更新表を参照してください。ARC ドキュメント履歴ページで自動 RSS アラートをサブスクライブすることもできます。

AWS ARC でのゾーンオートシフトの マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: AWSZonalAutoshiftPracticeRunSLRPolicy

IAM エンティティに `AWSZonalAutoshiftPracticeRunSLRPolicy` をアタッチすることはできません。このポリシーは、Amazon Application Recovery Controller (ARC) がゾーンオートシフトに対して以下を実行できるようにする、サービスにリンクされたロールにアタッチされます。

- 顧客提供の Amazon CloudWatch アラームと顧客 Health Dashboard イベントをモニタリングして練習実行を行う
- 練習実行 (練習のゾーンシフト) を管理します。
- 練習実行とオートシフトのバランスの取れた容量チェックを管理する

詳細については、「ARC でのゾーンオートシフトのサービスにリンクされたロールの使用」を参照してください。

ゾーンオートシフトの AWS 管理ポリシーの更新

このサービスがこれらの変更の追跡を開始してからの ARC でのゾーンオートシフトの AWS マネージドポリシーの更新の詳細については、「」を参照してくださいAmazon Application Recovery Controller (ARC) の AWS マネージドポリシーの更新。このページの変更に関する自動通知については、ARC の「ドキュメント履歴ページ」で RSS フィードをサブスクライブしてください。

ゾーンオートシフトのクォータ

Amazon Application Recovery Controller (ARC) のゾーンオートシフトには、次のクォータが必要です。

エンティティ	クォータ
練習実行設定あたりの結果アラーム数	10 クォータの引き上げをリクエストできます。
練習実行設定あたりのブロッキングアラーム数	10 クォータの引き上げをリクエストできます。

ルーティングコントロールを使用して ARC のマルチリージョンアプリケーションを復旧する

このセクションでは、Amazon Application Recovery Controller (ARC) のルーティング制御機能を使用して中断を最小限に抑え、 AWS アプリケーションを複数の にデプロイするときにユーザーに継続性を提供する方法について説明します AWS リージョン。

また、ARC の機能である準備状況チェックについても説明します。これを使用して、アプリケーションとリソースの復旧準備が整っているかどうかを把握できます。

このセクションのトピックでは、ルーティングコントロールと準備状況チェックの機能、設定方法、および使用方法について説明します。

トピック

- ARC でのルーティングコントロール
- ARC の準備状況チェック
- ARC でのリージョン切り替え

ARC でのルーティングコントロール

複数の のアプリケーションレプリカへのトラフィックをフェイルオーバーするには AWS リージョン、Amazon Route 53 の特定の種類のヘルスチェックと統合された Amazon Application Recovery Controller (ARC) のルーティングコントロールを使用できます。ルーティングコントロールは、あるリージョンレプリカから別のレプリカにクライアントトラフィックを切り替えることができる、シンプルなオン/オフスイッチです。トラフィックの再ルーティングは、Amazon Route 53 DNS レコードを使用して設定されたルーティングコントロールのヘルスチェックによって行われます。例えば、各リージョンのアプリケーションレプリカのフロントとなるドメイン名に関連付けられた DNS フェイルオーバーレコードなどです。

このセクションでは、ルーティングコントロールの仕組み、ルーティングコントロールコンポーネントの設定方法、およびフェイルオーバー時にトラフィックを再ルーティングするルーティングコントロールの使用方法について説明します。

ARC のルーティングコントロールコンポーネントには、クラスター、コントロールパネル、ルーティングコントロール、ルーティングコントロールのヘルスチェックがあります。すべてのルーティ

ングコントロールはコントロールパネルにグループ化されます。ARC がクラスター用に作成するデフォルトのコントロールパネルでルーティングコントロールをグループ化したり、独自のカスタムコントロールパネルを作成したりもできます。コントロールパネルまたはルーティングコントロールを作成する前に、クラスターを作成する必要があります。ARC の各クラスターは、5 つの AWS リージョンのエンドポイントからなるデータプレーンです。

ルーティングコントロールとルーティングコントロールのヘルスチェックを作成したら、ルーティングコントロールに対して、リカバリ自動化の意図しない副作用を防止するのに役立つ安全ルールを作成できます。ルーティングコントロールの状態を更新して、トラフィックを個別またはバッチで再ルーティングするには、 AWS CLI または API アクション (推奨) を使用するか、 を使用します AWS マネジメントコンソール。

このセクションでは、ルーティングコントロールの仕組みと、アプリケーションのトラフィックを再ルーティングするルーティングコントロールの作成方法や使用方法について説明します。

> **Important**
>
> 災害シナリオにおけるアプリケーションのフェイルオーバープランの一環として、ARC を使用したトラフィックの再ルーティングを準備する詳細については、「ARC でのルーティングコントロールに関するベストプラクティス」を参照してください。

ルーティングコントロールについて

ルーティングコントロールは、Amazon Route 53 のヘルスチェックを使用してトラフィックをリダイレクトします。ヘルスチェックは、Elastic Load Balancing のロードバランサーなど、リカバリグループでセルの最上位リソースに関連付けられた DNS レコードで設定されます。例えば、ルーティングコントロールの状態を `Off` (あるセルへのトラフィックフローを停止) に更新し、別のルーティングコントロールの状態を On (別のセルへのトラフィックフローを開始) に更新することで、あるセルから別のセルにトラフィックをリダイレクトできます。トラフィックフローを変更するプロセスは、ルーティングコントロールに関連する Route 53 ヘルスチェックであり、ARC がヘルスを更新した後、対応するルーティングコントロールの状態に基づいて、正常または異常に設定します。

ルーティングコントロールは、DNS エンドポイントを持つすべての AWS サービスでのフェイルオーバーをサポートします。ディザスタリカバリ、またはアプリケーションのレイテンシー低下やその他の問題を検出したときに、ルーティングコントロールの状態を更新してトラフィックをフェイルオーバーできます。

また、ルーティングコントロールに安全ルールを設定することにより、ルーティングコントロールを使用してトラフィックを再ルーティングしても可用性が損なわれないようにすることもできます。詳細については、「ルーティングコントロールの安全ルールを作成する」を参照してください。

ルーティングコントロール自体は、エンドポイントの基盤状態を監視するヘルスチェックではないという点に注意してください。例えば、Route 53 ヘルスチェックとは異なり、ルーティングコントロールは応答時間や TCP 接続時間をモニタリングしません。ルーティングコントロールは、ヘルスチェックを制御するシンプルなオン/オフスイッチです。通常、状態を変更してトラフィックをリダイレクトすると、その変更によってトラフィックがアプリケーションスタック全体における特定のエンドポイントに移動したり、アプリケーションスタック全体へのルーティングができなくなったりします。例えば、ルーティングコントロールの状態を On から Off に変更する単純なシナリオでは、DNS フェイルオーバーレコードに関連付けた Route 53 ヘルスチェックが更新され、トラフィックがエンドポイントの外に移動します。

ルーティングコントロールの使用方法

ルーティングコントロールの状態を更新してトラフィックを再ルーティングするには、ARC のクラスターエンドポイントのいずれかにトラフィックを接続する必要があります。接続しようとしているエンドポイントが使用できない場合は、別のクラスターエンドポイントで状態を変更してみてください。クラスターのエンドポイントは、定期的なメンテナンスや更新により、使用可能状態と使用不可状態が切り替わるため、ルーティングコントロールの状態を変更するプロセスは各エンドポイントを交代で試すように準備しておく必要があります。

ルーティングコントロールを作成するときは、ルーティングコントロールのヘルスチェックを各アプリケーションレプリカのフロントにある Route 53 DNS 名に関連付けるように DNS レコードを設定します。例えば、2 つのリージョンにそれぞれ 1 つずつ、2 つのロードバランサー間のトラフィックフェイルオーバーを制御するには、ルーティングコントロールのヘルスチェックを 2 つ作成し、それらを 2 つの DNS レコード (フェイルオーバールーティングポリシー付きのエイリアスレコード、それぞれのロードバランサーのドメイン名が付いたエイリアスレコードなど) に関連付けます。

ARC のルーティングコントロールを Route 53 ヘルスチェックと DNS レコードセットと併用し、加重ルーティングポリシー付きの DNS レコードを使用することにより、より複雑なトラフィックフェイルオーバーシナリオを設定することもできます。詳細な例については、次の AWS ブログ記事の「ユーザートラフィックのフェイルオーバーに関するセクション」を参照してください。Amazon Application Recovery Controller (ARC) を使用した回復力の高いアプリケーションの構築、パート 2: マルチリージョンスタック

ルーティングコントロール AWS リージョン を使用して のフェイルオーバーを開始すると、トラフィックフローに関連する手順により、トラフィックがすぐにリージョン外に移動しないことがあ

ります。また、クライアントの挙動または接続の再利用によっては、リージョンで進行中の既存の接続が完了するまでに、若干時間がかかる場合もあります。お使いの DNS 設定やその他の要因によって、既存の接続が数分で完了したり、さらに時間がかかったりする場合があります。詳細については、「トラフィックのシフトを迅速に完了させる」を参照してください。

ルーティングコントロールのメリット

ARC のルーティングコントロールには、従来のヘルスチェックによるトラフィックの再ルーティングと比べるといくつかメリットがあります。例えば、次のようになります。

- ルーティングコントロールでは、アプリケーションスタック全体をフェイルオーバーできます。これは、Amazon EC2 インスタンスのように、リソースレベルのヘルスチェックに基づいてスタックの個々のコンポーネントをフェイルオーバーするのとは対照的です。
- ルーティングコントロールでは、安全で簡単に手動で上書きができ、内部モニタが問題を検出しなかった場合に、トラフィックをメンテナンスのためにシフトしたり、障害からリカバリするためにシフトしたりできます。
- ルーティングコントロールと安全ルールを組み合わせて使用することで、完全に自動化されたヘルスチェックベースの自動化で発生する可能性のある一般的な副作用 (フェイルオーバーの準備が整っていないスタンバイインフラストラクチャへのフェイルオーバーなど) を防げます。

アプリケーションの耐障害性と可用性を向上させるために、ルーティングコントロールをフェイルオーバー戦略に組み込む例を次に示します AWS。

リージョン間で複数の (通常は 3 つの) 冗長レプリカを実行する AWS ことで、 で高可用性 AWS アプリケーションをサポートできます。そして、Amazon Route 53 のルーティングコントロールを使用して、トラフィックを適切なレプリカにルーティングできます。

例えば、1 つのアプリケーションレプリカをアクティブに設定してアプリケーショントラフィックを処理し、もう 1 つのアプリケーションレプリカをスタンバイレプリカとして設定できます。アクティブなレプリカに障害が発生した場合、ユーザーのトラフィックをスタンバイレプリカに再ルーティングして、アプリケーションの可用性を復元できます。レプリカとの間でフェイルアウェイを行うかどうかは、モニタリングシステムとヘルスチェックシステムの情報に基づいて判断する必要があります。

より迅速なリカバリを実現したい場合、アーキテクチャに合わせて選択できる別のオプションとしては、アクティブ/アクティブ実装があります。このアプローチでは、レプリカが同時にアクティブになります。つまり、トラフィックを別のアクティブなレプリカに再ルーティングするだけで、障害が

発生したアプリケーションのレプリカからユーザーを遠ざけ、障害からリカバリできるということです。

AWS ルーティングコントロールのリージョンの可用性

Amazon Application Recovery Controller (ARC) のリージョンサポートとサービスエンドポイントに関する詳細は、「Amazon Web Services 全般のリファレンス」にある「Amazon Application Recovery Controller (ARC) エンドポイントとクォータ」を参照してください。

> **Note**
>
> Amazon Application Recovery Controller (ARC) のルーティングコントロールは、グローバル機能です。ただし、リージョン ARC AWS CLI コマンドで米国西部 (オレゴン) リージョンを指定する必要があります (パラメータ を指定`--region us-west-2`)。つまり、クラスター、コントロールパネル、ルーティングコントロールなどのリソースを作成する場合です。

ARC ルーティングコントロールは、ARC ヘルスチェックの状態を変更するオン/オフスイッチです。このスイッチは、トラフィックをリダイレクトする DNS レコードに関連付けることができます。例えば、トラフィックをプライマリデプロイのレプリカからスタンバイデプロイのレプリカにリダイレクトします。

アプリケーション障害やレイテンシーの問題が発生した場合は、ルーティングコントロールの状態を更新して、例えばトラフィックをプライマリレプリカからスタンバイレプリカに移動できます。信頼性の高い ARC データプレーン API オペレーションを使用して、ルーティングコントロールのクエリやルーティングコントロールの状態の更新を行うことにより、ディザスタリカバリシナリオでのフェイルオーバーを ARC に依存できます。詳細については、「ARC API を使用してルーティングコントロールの状態を取得および更新する (推奨)」を参照してください。

ARC は、5 つの冗長なリージョンエンドポイントのセットであるクラスター内で、ルーティングコントロールの状態を維持します。ARC は、Amazon EC2 フリートにあるクラスター全体にルーティングコントロールの状態の変更を伝達し、5 つの AWS リージョンにまたがるクォーラムを取得します。伝播後、API と信頼性の高いデータプレーンを使用して ARC にルーティングコントロールの状態をクエリすると、コンセンサスビューが返されます。

5 つのクラスターエンドポイントのいずれかを操作して、ルーティングコントロールの状態を (例えば `Off` から `On` に) 更新できます。その後、ARC はクラスターの 5 つのリージョンに更新を伝播します。

5 つのクラスターエンドポイントすべてにわたるデータ整合性は、平均 5 秒以内、最大 15 秒以内で達成されます。

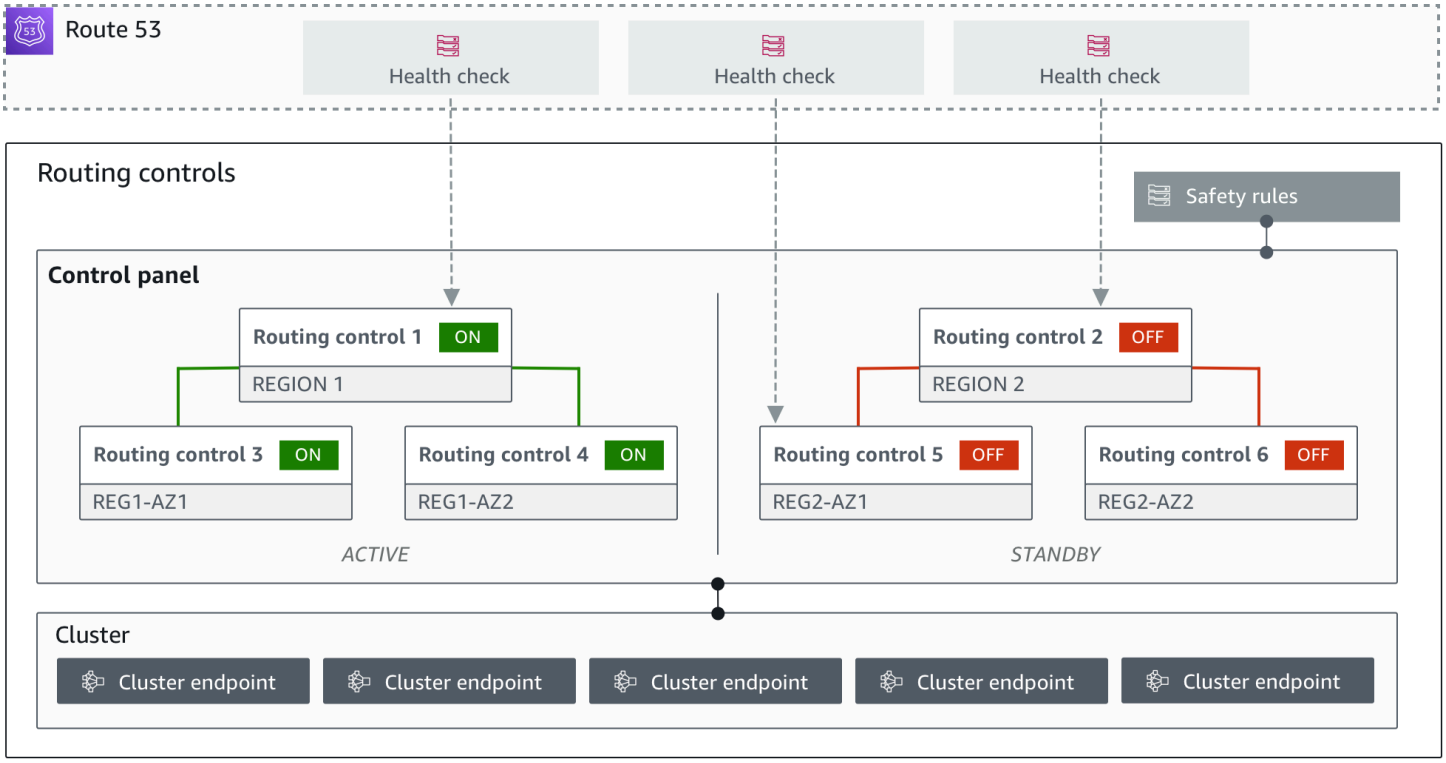
ARC のデータプレーンは非常に信頼性が高く、セル間において手動でアプリケーションをフェイルオーバーできます。ARC では、5 つのクラスターエンドポイントのうち少なくとも 3 つのエンドポイントに常にアクセスでき、ルーティングコントロールの状態を変更できます。アクセスパターンを遅らせる可能性のある「うるさい隣人 (ノイジーネイバー)」の影響を受けないように、各 ARC クラスターはシングルテナントであることに注意してください。

ルーティングコントロールの状態を変更するときは、次の 3 つの基準に基づいて行ってください。失敗する可能性が低くなります。

- 5 つのエンドポイントのうち少なくとも 3 つが利用可能で、クォーラムの一部を担っている。
- 有効な IAM 認証情報を所持しており、動作中のリージョンクラスターエンドポイントに照らして認証できる。
- Route 53 データプレーンが正常である (このデータプレーンは 100% の可用性 SLA を満たすように設計されている)。

ルーティングコントロールのコンポーネント

次の図は、ARC のルーティングコントロール機能をサポートするコンポーネントの例を示しています。ここに示されているルーティングコントロール (1 つのコントロールパネルにグループ化) では、2 つのリージョンそれぞれに配置する 2 つのアベイラビリティーゾーンへのトラフィックを管理できます。ルーティングコントロールの状態を更新すると、ARC は Amazon Route 53 のヘルスチェックを変更し、DNS トラフィックを別のセルにリダイレクトします。ルーティングコントロールに設定する安全ルールは、フェイルオープンシナリオやその他の意図しない結果を防ぐのに役立ちます。



以下は、ARC におけるルーティングコントロール機能のコンポーネントです。

クラスター

クラスターは、5 つの冗長なリージョンエンドポイントのセットであり、これに対して API コールを開始し、ルーティングコントロールの状態を更新したり取得したりします。クラスターにはデフォルトのコントロールパネルがあり、1 つのクラスターで複数のコントロールパネルと複数のルーティングコントロールをホストできます。

ルーティングコントロール

ルーティングコントロールは、クラスター上でホストされるシンプルなオン/オフスイッチであり、セルに出入りするクライアントトラフィックのルーティングを制御します。ルーティングコントロールを作成するときは、Route 53 に ARC ヘルスチェックを追加してください。これにより、ARC でルーティングコントロールの状態を更新したときに、(アプリケーションの DNS レコードで構成されたヘルスチェックを使用して) トラフィックを再ルーティングできます。

ルーティングコントロールのヘルスチェック

ルーティングコントロールは Route 53 のヘルスチェックと統合されています。ヘルスチェックは、フェイルオーバーレコードなど、各アプリケーションレプリカのフロントにある DNS レコードと関連付けられています。ルーティングコントロールの状態を変更すると、ARC は対応す

るヘルスチェックを更新し、トラフィックをリダイレクトします (スタンバイレプリカへのフェイルオーバーなど)。

コントロールパネル

コントロールパネルには、関連する一連のルーティングコントロールがグループ化されています。1 つのコントロールパネルに複数のルーティングコントロールを関連付けることができ、そのコントロールパネルの安全ルールを作成することで、実行したトラフィックリダイレクトの更新が安全に行われるようにします。例えば、各アベイラビリティーゾーンの各ロードバランサーにルーティングコントロールを設定して、それらを同じコントロールパネルにグループ化できます。次に、安全ルール (「アサーションルール」) を追加して、意図しない「フェイルオープン」シナリオを回避するために、常に 1 つ以上のゾーン (ルーティングコントロールで表される) がアクティブ状態であるようにします。

デフォルトのコントロールパネル

クラスターを作成すると、ARC でデフォルトのコントロールパネルが作成されます。デフォルトでは、クラスターで作成したすべてのルーティングコントロールがデフォルトのコントロールパネルに追加されます。もしくは、独自のコントロールパネルを作成して、関連するルーティングコントロールをグループ化することもできます。

安全ルール

安全ルールは、リカバリアクションによってアプリケーションの可用性が誤って損なわれることがないように、ルーティングコントロールに追加するルールです。例えば、全体的な「オン/オフ」スイッチとして機能するルーティングコントロールを生成する安全ルールを作成できます。これにより、他の一連のルーティングコントロールを有効または無効にできます。

エンドポイント (クラスターエンドポイント)

ARC の各クラスターには、ルーティングコントロールの状態の設定や取得に使用できる 5 つのリージョナルエンドポイントがあります。エンドポイントにアクセスするプロセスでは、ARC がメンテナンスのためにエンドポイントを定期的に起動または停止することを前提としているため、エンドポイントに接続するまで各エンドポイントを続けて試す必要があります。エンドポイントにアクセスして現在のルーティングコントロールの状態 (オンまたはオフ) を取得したり、ルーティングコントロールの状態を変更してアプリケーションのフェイルオーバーをトリガーしたりします。

ルーティングコントロールのデータプレーンとコントロールプレーン

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは、可用性が高く、災害シナリオで必要なときに使用できるようにすることをお勧めします。信頼性と耐障害性を最大限に高めるため、可能であれば通常、データプレーン機能をメカニズムに使用する必要があります。そのことを念頭に置いて、サービス機能がコントロールプレーンとデータプレーンにどのように分けられているのか、また、サービスのデータプレーンで非常に高い信頼性が期待できるのはどのような場合なのかを理解することが重要です。

ほとんどの AWS サービスと同様に、ルーティングコントロール機能の機能はコントロールプレーンとデータプレーンでサポートされています。どちらのタイプも信頼できるように構築されていますが、コントロールプレーンはデータ整合性のために最適化され、データプレーンは可用性のために最適化されています。データプレーンは、コントロールプレーンが使用できなくなるような破壊的なイベントでも、可用性を維持できるように設計されています。

一般に、コントロールプレーンを使用すると、サービス内のリソースの作成、更新、削除などの基本的な管理機能を実行できます。データプレーンはサービスのコア機能を提供します。このため、障害発生時にトラフィックをスタンバイレプリカに再ルーティングする必要がある場合など、可用性が重要な場合はデータプレーンオペレーションを使用することをお勧めします。

ルーティングコントロールでは、コントロールプレーンとデータプレーンは次のように分けられます。

- ルーティングコントロールのコントロールプレーン API は、米国西部 (オレゴン) リージョン (us-west-2) でサポートされている Recovery Control 設定 API です。これらの API オペレーションまたは を使用して AWS マネジメントコンソール 、クラスター、コントロールパネル、ルーティングコントロールを作成または削除し、アプリケーションのトラフィックを再ルーティングする必要があるディザスタリカバリイベントに備えることができます。ルーティングコントロール設定のコントロールプレーンは、可用性が高くありません。
- ルーティングコントロールのデータプレーンは、地理的に隔離された 5 つの AWS リージョンにまたがる専用クラスターです。ユーザーごとに、ルーティングコントロールのコントロールプレーンを使用して 1 つ以上のクラスターを作成します。クラスターはコントロールパネルとルーティングコントロールをホストします。そして、アプリケーションのトラフィックを再ルーティングしたい場合は、ルーティングコントロール (リカバリクラスター) API を使用してルーティングコントロールの状態を取得、リスト化、更新します。ルーティングコントロールのデータプレーンは、可用性が高い設計です。

ルーティングコントロールデータプレーンは可用性が高いため、イベントから回復するためにフェイルオーバーする場合は、 を使用して API コール AWS Command Line Interface をルーティングコントロールの状態と連携させることをお勧めします。ルーティングコントロールを使用して復旧オペレーションを準備して完了する際に考慮すべき重要事項の詳細は、「ARC でのルーティングコントロールに関するベストプラクティス」を参照してください。

データプレーン、コントロールプレーン、および が高可用性目標を達成するためのサービス AWS を構築する方法の詳細については、Amazon Builders' Library」の「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。

Amazon Application Recovery Controller (ARC) でのルーティングコントロールにタグ付けする

タグは、 AWS リソースを識別して整理するために使用する単語またはフレーズ (メタデータ) です。各リソースに複数のタグを追加できます。各タグには、定義したキーと値が含まれます。例えば、キーを環境、値を本番とできます。追加したタグに基づいて、リソースを検索したりフィルタ処理したりできます。

ARC のルーティングコントロールでは、以下のリソースにタグを付けることができます。

- クラスター
- コントロールパネル
- 安全ルール

ARC でのタグ付けは、API を使用してのみ可能です。例えば、 AWS CLIを使用します。

以下は、 AWS CLIを使用したルーティングコントロールでのタグ付けの例です。

```
aws route53-recovery-control-config --region us-west-2 create-cluster --
cluster-name example1-cluster --tags Region=PDX,Stage=Prod
```

```
aws route53-recovery-control-config --region us-west-2 create-control-panel
--control-panel-name example1-control-panel --cluster-arn arn:aws:route53-
recovery-control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh
--tags Region=PDX,Stage=Prod
```

詳細については、Amazon Application Recovery Controller (ARC) の「Recovery Control 設定 API リファレンスガイド」の「TagResource」を参照してください。

ARC でのルーティングコントロールの料金

ARC でのルーティングコントロールには、作成したクラスターごとに時間単位のコストが発生します。各クラスターは複数のルーティングコントロールをホストでき、それらを使用してアプリケーションのフェイルオーバーをトリガーします。

コストを管理し、効率を向上させるために、クラスターのクロスアカウント共有を設定し、1 つのクラスターを複数の AWS アカウントと共有できます。詳細については、「ARC でクラスターのクロスアカウントをサポート」を参照してください。

ARC の料金および料金例の詳細については、「ARC の料金」を参照してください。

Amazon Application Recovery Controller (ARC) でマルチリージョンのリカバリを開始する方法

Amazon Application Recovery Controller (ARC) でルーティングコントロールを使用してアプリケーションをフェイルオーバーするには、複数の AWS アプリケーションが必要です AWS リージョン。開始するには、まず、アプリケーションが各リージョンのサイロ化されたレプリカにセットアップされていることを確認し、イベント中に 1 つのレプリカから別のレプリカにフェイルオーバーできるようにします。次に、ルーティングコントロールを作成して、アプリケーショントラフィックをプライマリアプリケーションからセカンダリアプリケーションにフェイルオーバーするように再ルーティングし、ユーザーの継続性を維持します。

> **Note**
>
> アベイラビリティーゾーンによってサイロ化されているアプリケーションがある場合は、フェイルオーバーリカバリにゾーンシフトまたはゾーンオートシフトを使用することを検討してください。アベイラビリティーゾーンの障害からアプリケーションを確実に回復するために、ゾーンシフトまたはゾーンオートシフトを使用する設定は必要ありません。詳細については、「ゾーンシフトとゾーンオートシフトを使用して ARC のアプリケーションを復旧する」を参照してください。

ARC ルーティングコントロールを使用してイベント中にアプリケーションを復旧できるように、互いにレプリカとなるアプリケーションを少なくとも 2 つ設定することをお勧めします。各レプリカまたはセルは、 を表します AWS リージョン。リージョンに合わせてアプリケーションリソースを設定したら、次の手順を実行して、アプリケーションが復旧に成功するように設定されていることを確認します。

ヒント: セットアップを簡素化するために、冗長レプリカを持つアプリケーションを作成する HashiCorp Terraform テンプレート CloudFormation と HashiCorp Terraform テンプレートを提供しています。詳細とテンプレートのダウンロードは「サンプルアプリケーションのセットアップ」をご確認ください。

ルーティングコントロールを使用する準備をするには、以下を実行して、アプリケーションが回復力を持つように設定されていることを確認します。

1. 各リージョンで互いにレプリカとなるアプリケーションスタック (ネットワークレイヤーとコンピューティングレイヤー) の独立したコピーを構築して、イベントが発生したときにトラフィックをフェイルオーバーできるようにします。1 つのレプリカの障害がもう 1 つのレプリカに影響を与えるようなクロスリージョンの依存関係がアプリケーションコードにないことを確認してください。間で正常にフェイルオーバーするには AWS リージョン、スタックの境界がリージョン内にある必要があります。
2. アプリケーションに必要なすべてのステートフルデータをレプリカ全体に複製します。 AWS データベースサービスを使用して、データをレプリケートできます。

トラフィックフェイルオーバーのルーティングコントロールの開始

Amazon Application Recovery Controller (ARC) のルーティングコントロールを使用すると、個別の AWS リージョンで実行されている冗長アプリケーションコピーやレプリカ間のトラフィックのフェイルオーバーをトリガーできます。フェイルオーバーは、Amazon Route 53 データプレーンを使用して DNS で実行されます。

次のセクションで説明するとおり、各リージョンでレプリカを設定したら、それぞれをルーティングコントロールに関連付けられます。まず、ルーティングコントロールを各リージョンのレプリカの最上位ドメイン名に関連付けます。次に、ルーティングコントロールのヘルスチェックをルーティングコントロールに追加して、トラフィックフローをオンまたはオフにできるようにします。これにより、アプリケーションのレプリカ間のトラフィックルーティングを制御できます。

でルーティングコントロールの状態を更新 AWS マネジメントコンソール してトラフィックをフェイルオーバーできますが、代わりに API または を使用して ARC アクション AWS CLIを使用して変更することをお勧めします。API アクションはコンソールに依存しないため、耐障害性が向上します。

例えば、us-west-1 から us-east-1 へのアベイラビリティーゾーン間でフェイルオーバーを行う場合は、`update-routing-control-state` API アクションを使用して `us-west-1` のステータスを `Off` にし、`us-east-1` のステータスを `On` に設定できます。

ルーティングコントロールコンポーネントを作成してアプリケーションのフェイルオーバーを設定する前に、アプリケーションがリージョンレプリカにサイロ化されていることを確認し、一方から他方にフェイルオーバーできるようにします。詳細を確認し、新しいアプリケーションのサイロ化またはサンプルスタックの作成を開始するには、次のセクションを参照してください。

サンプルアプリケーションのセットアップ

ルーティングコントロールの仕組みを理解しやすくするために、TicTacToe というサンプルアプリケーションを提供しています。この例では、 CloudFormation テンプレートを使用してプロセスを簡素化し、ダウンロード可能な CloudFormation テンプレートを使用して、ARC のセットアップと使用をすばやく検討できます。

サンプルアプリをデプロイしたら、テンプレートを使用して ARC コンポーネントを作成し、アプリへのトラフィックフローを管理するルーティングコントロールを使用できます。テンプレートとプロセスは、独自のシナリオやアプリケーションに合わせて調整できます。

サンプルアプリケーションと CloudFormation テンプレートの使用を開始するには、ARC GitHub リポジトリの README の手順を参照してください。 CloudFormation テンプレートの使用の詳細については、「 AWS CloudFormation ユーザーガイド」のCloudFormation 概念を参照してください。

ARC でのルーティングコントロールに関するベストプラクティス

ARC でルーティングコントロールのリカバリとフェイルオーバーに備えるときは、以下のベストプラクティスに従うことをお勧めします。

トピック

- 専用で存続期間の長い AWS 認証情報を安全かつ常にアクセス可能に保つ
- フェイルオーバーに関連する DNS レコードの低い TTL 値を選択する
- クライアントがエンドポイントに接続したままになる時間を制限する
- 5 つのリージョンクラスターエンドポイントとルーティングコントロール ARNs
- いずれかのエンドポイントをランダムに選択して、ルーティングコントロールの状態を更新します。
- コンソールではなく、非常に信頼性の高いデータプレーン API を使用してルーティングコントロールの状態を一覧表示および更新する

専用で存続期間の長い AWS 認証情報を安全かつ常にアクセス可能に保つ

ディザスタリカバリ (DR) シナリオでは、復旧タスクにアクセスして AWS 実行するための簡単なアプローチを使用して、システムの依存関係を最小限に抑えます。DR タスク用に IAM の長期間有効な認証情報を作成し、オンプレミスの物理的な金庫または仮想ボールトにこれを保管して、必要に応じてアクセスできるようにします。IAM を使用すると、アクセスキーや AWS リソースへのアクセス許可などのセキュリティ認証情報を一元管理できます。DR 以外のタスクについては、AWS Single Sign-On など、 AWS サービスを使ったフェデレーションアクセスを引き続き使用することが推奨されます。

リカバリクラスターのデータプレーン API を使って ARC でフェイルオーバータスクを実行するときは、ARC の IAM ポリシーをユーザーにアタッチできます。詳細についてはAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例を参照してください。

フェイルオーバーに関連する DNS レコードの低い TTL 値を選択する

フェイルオーバーの一環として変更する必要がある DNS レコード、特にヘルスチェックの対象となるレコードは、TTL 値を低く設定しておくのが適切です。このシナリオでは、TTL を 60 秒または 120 秒に設定するのが一般的です。

DNS TTL (有効期間) の設定は、新しいレコードをリクエストするまでに、どの程度の期間、レコードをキャッシュすべきかを DNS リゾルバーに伝えます。TTL を選択する際は、レイテンシーと信頼性の間、また、変化への反応との間でいずれかを優先しなくてはなりません。レコードの TTL を短くすると、DNS リゾルバーはレコードの更新をより頻繁に通知します。TTL から、クエリを頻繁に実行するように指示されるためです。

詳細については、「Amazon Route 53 DNS のベストプラクティス」の「DNS レコードの TTL 値の選択」を参照してください。

クライアントがエンドポイントに接続したままになる時間を制限する

ルーティングコントロールを使用して 間でシフトする場合 AWS リージョン 、Amazon Application Recovery Controller (ARC) がアプリケーショントラフィックを移動するために使用するメカニズムは DNS 更新です。この更新により、すべての新しい接続が障害のある場所から遠ざけられます。

ただし、既存のオープン接続を持つクライアントは、クライアントが再接続するまで、障害が発生したロケーションに対してリクエストを引き続き行う場合があります。迅速な復旧を確保するために、クライアントがエンドポイントに接続し続ける時間を制限することをお勧めします。

Application Load Balancer を使用する場合は、keepalive オプションを使用して接続の継続期間を設定できます。詳細については、「Application Load Balancer ユーザーガイド」の「HTTP クライアントのキープアライブ期間」を参照してください。

Application Load Balancer では、HTTP クライアントのキープアライブ期間の値はデフォルトで 3600 秒 (1 時間) に設定されます。アプリケーションの目標復旧時間に合わせて、例えば 300 秒のように値を小さくすることをお勧めします。HTTP クライアントのキープアライブ時間を設定する際は、一般的に再接続が増えてレイテンシーに影響が出ることと、障害のある AZ やリージョンからすべてのクライアントをより早く切り替えられることとのトレードオフである点を考慮してください。

5 つのリージョンクラスターエンドポイントとルーティングコントロール ARNs

ARC のリージョンクラスターエンドポイントの、ローカルコピーをブックマークに保存するか、エンドポイントを再試行するために使用する自動化コードに保存することをお勧めします。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 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。ARC クラスターのエンドポイントは、DescribeCluster API オペレーションを使用すると一覧表示できます。

いずれかのエンドポイントをランダムに選択して、ルーティングコントロールの状態を更新します。

ルーティングコントロールでは、障害が発生した場合でも高可用性を確保できるよう 5 つのリージョンエンドポイントが提供されています。十分な耐障害性を実現するには、必要に応じて 5 つのエンドポイントをすべて使用できる再試行ロジックを用意することが重要です。クラスターエンドポイントを試す例など、 AWS SDK でのコード例の使用については、「」を参照してくださいAWS SDKsコード例。

コンソールではなく、非常に信頼性の高いデータプレーン API を使用してルーティングコントロールの状態を一覧表示および更新する

ルーティングコントロールと状態を確認するときは、ARC データプレーン API を使って ListRoutingControls オペレーションを実行し、フェイルオーバーのためルーティングコントロールの状態を更新してトラフィックをリダイレクトするときは UpdateRoutingControlState オペレーションを実行します。 AWS CLI (これらの例のように) またはいずれかの AWS SDKs を使用して記述したコードを使用できます。ARC では、きわめて信頼性の高い方法として、データプレーンで API を使用してトラフィックをフェイルオーバーできます。 AWS マネジメントコンソールでルーティングコントロールの状態を変更するのではなく、こちらの API を使用することをお勧めします。

データプレーン API を使用するには、ARC のリージョンクラスターエンドポイントのいずれかに接続します。そのエンドポイントが使用できない場合は、別のクラスターエンドポイントに接続します。

安全ルールが原因でルーティングコントロールの状態を更新できない場合は、そのルールを迂回して更新し、トラフィックをフェイルオーバーすることが可能です。詳細については、「安全ルールを上書きしてトラフィックを再ルーティングする」を参照してください。

ARC でのフェイルオーバーのテスト

ARC のルーティングコントロールを使って、プライマリのアプリケーションスタックからセカンダリのアプリケーションスタックにフェイルオーバーして、定期的にフェイルオーバーをテストします。追加した ARC の構造がスタック内の正常なリソースと一致していること、および、すべてが予定どおりに機能していることを確認することが重要です。使用している環境に ARC をセットアップした後でこのテストを行い、フェイルオーバー環境の準備が整うように定期的にテストを続ける必要があります。障害が発生した場合には、ユーザーのダウンタイムを回避できるよう、セカンダリシステムを起動してすばやく稼働させる必要があります。

ルーティング制御 API オペレーション

このセクションでは、Amazon Application Recovery Controller (ARC) でのルーティングコントロールのセットアップと使用に使える API オペレーションのリストを含むテーブルと、関連するドキュメントへのリンクを紹介します。

AWS Command Line Interfaceで一般的なルーティングコントロール設定 API オペレーションを使用する方法の例は、「で ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI」を参照してください。

次の表は、ルーティングコントロール設定に使用できる ARC API オペレーションを、関連するドキュメントへのリンクと共に一覧にしたものです。

Action	ARC コンソールの使用	ARC API の使用
クラスターを作成する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「CreateCluster」を参照

Action	ARC コンソールの使用	ARC API の使用
クラスターを記述する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DescribeCluster」を参照
クラスターを削除	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DeleteCluster」を参照
アカウントのクラスターを一覧表示する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「ListClusters」を参照
ルーティングコントロールを作成する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「CreateRoutingControl」を参照
ルーティングコントロールについて説明する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DescribeRoutingControl」を参照
ルーティングコントロールを更新する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「UpdateRoutingControl」を参照
ルーティングコントロールを削除する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DeleteRoutingControl」を参照
ルーティングコントロールを一覧表示する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「ListRoutingControls」を参照
コントロールパネルを作成する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「CreateControlPanel」を参照

Action	ARC コンソールの使用	ARC API の使用
コントロールパネルを説明する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DescribeControlPanel」を参照
コントロールパネルを更新する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「UpdateControlPanel」を参照
コントロールパネルを削除する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「DeleteControlPanel」を参照
コントロールパネルを一覧表示する	「ARC でルーティングコントロールのコンポーネントを作成する」を参照	「ListControlPanels」を参照
安全ルールを作成する	「ルーティングコントロールの安全ルールを作成する」を参照	「CreateSafetyRule」を参照
安全ルールを記述する	「ルーティングコントロールの安全ルールを作成する」を参照	「DescribeSafetyRule」を参照
安全ルールを更新する	「ルーティングコントロールの安全ルールを作成する」を参照	「UpdateSafetyRule」を参照
安全ルールを削除する	「ルーティングコントロールの安全ルールを作成する」を参照	「DeleteSafetyRule」を参照
安全ルールを一覧表示する	「ルーティングコントロールの安全ルールを作成する」を参照	「ListSafetyRules」を参照

Action	ARC コンソールの使用	ARC API の使用
関連付けられた Route 53 ヘルスチェックを一覧表示する	「ARC でルーティングコントロールのヘルスチェックを作成する」を参照	「ListAssociatedRoute53HealthChecks」を参照
クラスター共有の AWS RAM リソースポリシーを一覧表示する	「ARC でクラスターのクロスアカウントをサポート」を参照してください。	「GetResourcePolicy」を参照

次の表は、ルーティングコントロールのデータプレーンでトラフィックのフェイルオーバーを管理する際に使用できる一般的な ARC API オペレーションを、関連するドキュメントへのリンクと共に一覧にしたものです。

Action	ARC コンソールの使用	ARC API の使用
ルーティングコントロールの状態を取得する	「でのルーティングコントロールの状態の取得と更新 AWS マネジメントコンソール」を参照してください。	「GetRoutingControlState」を参照
ルーティングコントロールを一覧表示する	該当なし	「ListRoutingControls」を参照
ルーティングコントロールの状態を更新する	「でのルーティングコントロールの状態の取得と更新 AWS マネジメントコンソール」を参照してください。	「UpdateRoutingControlState」を参照
複数のルーティングコントロールの状態を更新する	「でのルーティングコントロールの状態の取得と更新 AWS マネジメントコンソール」を参照してください。	「UpdateRoutingControlStates」を参照

AWS SDK でのこのサービスの使用

AWS Software Development Kit (SDKsは、多くの一般的なプログラミング言語で使用できます。各 SDK には、デベロッパーが好みの言語でアプリケーションを簡単に構築できるようになる API、コード例、およびドキュメントが提供されています。

SDK ドキュメント	コード例
AWS SDK for C++	AWS SDK for C++ コード例
AWS CLI	AWS CLI コード例
AWS SDK for Go	AWS SDK for Go コード例
AWS SDK for Java	AWS SDK for Java コード例
AWS SDK for JavaScript	AWS SDK for JavaScript コード例
AWS SDK for Kotlin	AWS SDK for Kotlin コード例
AWS SDK for .NET	AWS SDK for .NET コード例
AWS SDK for PHP	AWS SDK for PHP コード例
AWS Tools for PowerShell	AWS Tools for PowerShell コード例
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) コード例
AWS SDK for Ruby	AWS SDK for Ruby コード例
AWS SDK for Rust	AWS SDK for Rust コード例
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP コード例
AWS SDK for Swift	AWS SDK for Swift コード例

このサービスに固有の例については、「AWS SDKsコード例」を参照してください。

可用性の例

必要なものが見つからなかった場合。このページの下側にある [Provide feedback] リンクから、コードの例をリクエストしてください。

で ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI

このセクションでは、 を使用して API オペレーションを使用して Amazon Application Recovery Controller (ARC) のルーティングコントロール機能 AWS Command Line Interface を操作する、ルーティングコントロールを使用する簡単なアプリケーション例について説明します。これらの例は、CLI を使用したルーティングコントロールの操作方法の基本的な理解を深めることを目的としています。

Amazon Application Recovery Controller (ARC) のルーティングコントロールを使用すると、個別のまたはアベイラビリティーゾーンで実行されている冗長なアプリケーションコピー AWS リージョンまたはレプリカ間でトラフィックフェイルオーバーをトリガーできます。

ルーティングコントロールは、クラスター上にプロビジョニングされるコントロールパネルと呼ばれるグループにまとめられます。ARC クラスターは、グローバルにデプロイされるリージョンエンドポイントのセットです。クラスターエンドポイントは、ルーティングコントロールの状態の設定と取得に使用できる可用性の高い API を提供します。ルーティングコントロール機能のコンポーネントの詳細については、「ルーティングコントロールのコンポーネント」を参照してください。

Note

ARC は、複数の AWS リージョンのエンドポイントをサポートするグローバルサービスです。ただし、ほとんどの ARC CLI コマンドでは米国西部 (オレゴン) リージョンを指定 (`--region us-west-2` パラメータを指定) します。例えば、リカバリグループ、コントロールパネル、クラスターを作成するときは、`region` パラメータを使用します。
クラスターを作成すると、ARC はリージョンのエンドポイントのセットを提供します。ルーティングコントロールの状態を取得または更新するには、CLI コマンドでリージョンエンドポイント (AWS リージョン および エンドポイント URL) を指定する必要があります。

の使用の詳細については AWS CLI、 AWS CLI 「 コマンドリファレンス」を参照してください。ルーティングコントロール API アクションのリストは、「ルーティング制御 API オペレーション」および「ルーティング制御 API オペレーション」を参照してください。

まず、クラスターの作成から始めて、ルーティングコントロールを使用してフェイルオーバーを管理するために必要なコンポーネントを作成します。

ルーティングコントロールコンポーネントの設定

最初のステップでは、クラスターを作成します。ARC クラスターは 5 つのエンドポイントのセットで、5 つの異なる AWS リージョンのそれぞれに 1 つずつあります。ARC のインフラストラクチャは、これらのエンドポイントが連携して機能することをサポートし、フェイルオーバーオペレーションの高い可用性と逐次一貫性を保証します。

1. クラスターを作成する

1a. クラスターを作成する。network-type はオプションで、IPV4 または DUALSTACK のいずれかになります。デフォルトは IPV4 です。

```
aws route53-recovery-control-config create-cluster --cluster-name test --network-type
 DUALSTACK
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "PENDING",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
}
```

ARC リソースを初めて作成すると、クラスターの作成中はステータスが PENDING になります。その進行状況は、describe-cluster を呼び出して確認できます。

1b. クラスターを記述します。

```
aws route53-recovery-control-config --region us-west-2 \
    describe-cluster --cluster-arn arn:aws:route53-recovery-
control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh
```

```
"Cluster": {
```

```
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "DEPLOYED",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
}
```

ステータスが DEPLOYED の場合、ユーザーが操作できるエンドポイントのセットを含むクラスターを、ARC が正常に作成したことを意味します。`list-clusters` を呼び出すと、すべてのクラスターを一覧表示できます。

1c. クラスターを一覧表示します。

```
aws route53-recovery-control-config --region us-west-2 list-clusters
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "DEPLOYED",
    "Owner": "123456789123",
    "NetworkType": "DUALSTACK"
}
```

1d. クラスターのネットワークタイプを更新します。オプションは `IPV4` または `DUALSTACK` です。

```
aws route53-recovery-control-config update-cluster \
--cluster-arn arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234 \
--network-type DUALSTACK
```

```
"Cluster": {
    "ClusterArn": "arn:aws:route53-recovery-
control::123456789123:cluster/12341234-1234-1234-1234-123412341234",
    "Name": "test",
    "Status": "PENDING",
    "Owner": "123456789123",
   "NetworkType": "DUALSTACK"
}
```

2. コントロールパネルを作成する

コントロールパネルは、ARC のルーティングコントロールを整理のために論理的にまとめたものです。クラスターを作成すると、ARC は DefaultControlPanel という名前のコントロールパネルを自動的に提供します。このコントロールパネルはすぐに使用できます。

コントロールパネルは 1 つのクラスターにのみ存在できます。コントロールパネルを別のクラスターに移動する場合は、そのコントロールパネルを削除して 2 つ目のクラスターで作成する必要があります。アカウントのすべてのコントロールパネルは、list-control-panels を呼び出すことで確認できます。特定のクラスター内のコントロールパネルだけを表示するには、--cluster-arn フィールドを追加します。

2a. コントロールパネルを一覧表示します。

```
aws route53-recovery-control-config --region us-west-2 \
    list-control-panels --cluster-arn arn:aws:route53-recovery-
control::111122223333:cluster/eba23304-1a51-4674-ae32-b4cf06070bdd
```

```
{
    "ControlPanels": [
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/1234567dddddd1234567dddddd1234567",
            "ClusterArn": "arn:aws:route53-recovery-
control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh",
            "DefaultControlPanel": true,
            "Name": "DefaultControlPanel",
            "RoutingControlCount": 0,
            "Status": "DEPLOYED"
        }
    ]
}
```

オプションで、create-control-panel を呼び出して独自のコントロールパネルを作成できます。

2b. コントロールパネルを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-control-panel \
        --control-panel-name NewControlPanel2 \
```

```
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": false,
        "Name": "NewControlPanel2",
        "RoutingControlCount": 0,
        "Status": "PENDING"
    }
}
```

ARC リソースを初めて作成すると、作成中はステータスが PENDING になります。describe-control-panel を呼び出して、進行状況を確認できます。

2c. コントロールパネルを記述します。

```
aws route53-recovery-control-config --region us-west-2 describe-control-panel \
    --control-panel-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456
```

```
{
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": true,
        "Name": "DefaultControlPanel",
        "RoutingControlCount": 0,
        "Status": "DEPLOYED"
    }
}
```

3. ルーティングコントロールを作成する

これでクラスターをセットアップし、コントロールパネルを確認したので、ルーティングコントロールの作成を開始できます。ルーティングコントロールを作成するときには、少なくとも、ルーティ

ングコントロールを組み込むクラスターの Amazon リソースネーム (ARN) を指定する必要があります。ルーティングコントロールのコントロールパネルの ARN を指定することもできます。また、コントロールパネルが配置されているクラスターも指定する必要があります。

コントロールパネルを指定しない場合、ルーティングコントロールは自動的に作成されたコントロールパネル (DefaultControlPanel) に追加されます。

create-routing-control を呼び出して、ルーティングコントロールを作成できます。

3a. ルーティングコントロールを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
    --routing-control-name NewRc1 \
    --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "RoutingControl": {
        "ControlPanelArn": " arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "Name": "NewRc1",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "Status": "PENDING"
    }
}
```

ルーティングコントロールは他の ARC リソースと同じ作成パターンに従うため、describe オペレーションを呼び出すことによって進行状況を追跡できます。

3b. ルーティングコントロールを記述します。

```
aws route53-recovery-control-config --region us-west-2 describe-routing-control \
       --routing-control-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
    "RoutingControl": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
```

```
        "Name": "NewRc1",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "Status": "DEPLOYED"
    }
}
```

list-routing-controls を呼び出すと、コントロールパネルにルーティングコントロールを一覧表示できます。コントロールパネルの ARN は必須です。

3c. ルーティングコントロールを一覧表示します。

```
aws route53-recovery-control-config --region us-west-2 list-routing-controls \
        --control-panel-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456
```

```
{
    "RoutingControls": [
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
            "Name": "Rc1",
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
            "Status": "DEPLOYED"
        },
        {
            "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
            "Name": "Rc2",
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
hijklmnop987654321",
            "Status": "DEPLOYED"
        }
    ]
}
```

ルーティングコントロールの状態を扱う次の例では、このセクションにリストされている 2 つのルーティングコントロール (Rc1 と Rc2) があることを前提としています。この例では、各ルーティ

ングコントロールは、アプリケーションがデプロイされているアベイラビリティーゾーンを表します。

4. 安全ルールを作成する

複数のルーティングコントロールを同時に使用する場合、両方のルーティングコントロールがオフになりすべてのトラフィックフローが停止するといった意図しない結果を避けるために、有効または無効にする際の安全対策を講じたいと思うかもしれません。これらの安全対策を作成するには、ルーティングコントロールの安全ルールを作成します。

安全ルールには、アサーションルールとゲートルールという 2 つのタイプがあります。安全ルールの詳細については、「ルーティングコントロールの安全ルールを作成する」を参照してください。

次の呼び出しは、2 つのルーティングコントロールのうち少なくとも 1 つが常に On に設定されているようにするアサーションルールの作成例です。ルールを作成するには、`assertion-rule` パラメータで `create-safety-rule` を実行します。

アサーションルール API オペレーションの詳細は、「Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイド」の「AssertionRule」を参照してください。

4a. アサーションルールを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-safety-rule \
        --assertion-rule '{"Name": "TestAssertionRule",
        "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
        "WaitPeriodMs": 5000,
        "AssertedControls":
        ["arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
        "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"],
        "RuleConfig": {"Threshold": 1, "Type": "ATLEAST", "Inverted": false}}'
```

```
{
    "Rule": {
        "ASSERTION": {
            "Arn": "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/safetyrule/333333444444",
            "AssertedControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
```

```
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"],
            "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
            "Name": "TestAssertionRule",
            "RuleConfig": {
                "Inverted": false,
                "Threshold": 1,
                "Type": "ATLEAST"
            },
            "Status": "PENDING",
            "WaitPeriodMs": 5000
        }
    }
}
```

次の呼び出しは、コントロールパネルにある一連のターゲットのルーティングコントロールに対する全体的なスイッチの「オン/オフ」または「ゲート」を提供するゲートルールの作成例です。これにより、例えば自動化による未承認の更新がされないように、ターゲットのルーティングコントロールの更新を禁止できます。この例では、ゲートスイッチは GatingControls パラメータで指定されるルーティングコントロールであり、制御または「ゲート」される 2 つのルーティングコントロールは TargetControls パラメータで指定されます。

> **Note**
>
> ゲートルールを作成する前に、DNS フェイルオーバーレコードを含まないゲートルーティングコントロールと、DNS フェイルオーバーレコードで構成するターゲットルーティングコントロールを作成する必要があります。

ルールを作成するには、gating-rule パラメータで create-safety-rule を実行します。

アサーションルール API オペレーションの詳細は、「Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイド」の「GatingRule」を参照してください。

4b. ゲートルールを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-safety-rule \
        --gating-rule '{"Name": "TestGatingRule",
        "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
```

```
        "WaitPeriodMs": 5000,
        "GatingControls": ["arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/
def123def123def"]
        "TargetControls": ["arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/
ghi456ghi456ghi",
        "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/lmn789lmn789lmn"],
        "RuleConfig": {"Threshold": 0, "Type": "OR", "Inverted": false}}'
```

```
{
    "Rule": {
        "GATING": {
            "Arn": "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/safetyrule/444444444444",
            "GatingControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/def123def123def"
            ],
            "TargetControls": [
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/ghi456ghi456ghi"
                "arn:aws:route53-recovery-control::888888888888:controlpanel/
zzz123yyy456xxx789zzz123yyy456xxx/routingcontrol/lmn789lmn789lmn"
            ],
            "ControlPanelArn": "arn:aws:route53-recovery-
control::888888888888:controlpanel/zzz123yyy456xxx789zzz123yyy456xxx",
            "Name": "TestGatingRule",
            "RuleConfig": {
                "Inverted": false,
                "Threshold": 0,
                "Type": "OR"
            },
            "Status": "PENDING",
            "WaitPeriodMs": 5000
        }
    }
}
```

他のルーティングコントロールのリソースと同様に、安全ルールはデータプレーンに伝播した後で記述、一覧表示、削除できます。

1 つ以上の安全ルールを設定した後は、引き続きクラスターを操作したり、ルーティングコントロールの状態を設定または取得したりできます。set-routing-control-state オペレーションによって作成したルールが破られると、次のような例外が発生します。

```
Cannot modify control state for [0123456bbbbbbb0123456bbbbbb01234560123
abcdefg1234567] due to failed rule evaluation
0123456bbbbbbb0123456bbbbbb01234563333333444444
```

最初の識別子は、ルーティングコントロールの ARN と連結されたコントロールパネルの ARN です。2 番目の識別子は、安全ルールの ARN と連結されたコントロールパネルの ARN です。

5. ヘルスチェックを作成する

ルーティングコントロールを使用してトラフィックをフェイルオーバーするには、Amazon Route 53 でヘルスチェックを作成し、そのヘルスチェックを DNS レコードに関連付けます。トラフィックをフェイルオーバーするために、ARC ルーティングコントロールはヘルスチェックをフェイルに設定して、Route 53 がトラフィックを再ルーティングするようにします。(ヘルスチェックはアプリケーションの正常性を無効にします。単にトラフィックを再ルーティングする方法として使用されます)。

例えば、2 つのセル (リージョンまたはアベイラビリティーゾーン) があるとします。1 つはアプリケーションのプライマリセルとして設定し、もう 1 つはセカンダリとしてフェイルオーバーするように設定します。

フェイルオーバー用にヘルスチェックを設定するには、例えば次の操作を行います。

1. ARC CLI を使用して、各セルのルーティングコントロールを作成します。
2. Route 53 CLI を使用して、ルーティングコントロールごとに Route 53 で ARC ヘルスチェックを作成します。
3. Route 53 CLI を使用して、Route 53 に 2 つのフェイルオーバー DNS レコードを作成し、それぞれにヘルスチェックを関連付けます。

5a. 各セルにルーティングコントロールを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
        --routing-control-name RoutingControlCell1 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
        --routing-control-name RoutingControlCell2 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

5b. 各ルーティングコントロールにヘルスチェックを作成します。

> **Note**
>
> Amazon Route 53 CLI を使用して ARC ヘルスチェックを作成します。

```
aws route53 create-health-check --caller-reference RoutingControlCell1 \
        --health-check-config \
        Type=RECOVERY_CONTROL,RoutingControlArn=arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
    "Location": "https://route53.amazonaws.com/2015-01-01/healthcheck/11111aaaa-bbbb-
cccc-dddd-ffffff22222",
    "HealthCheck": {
        "Id": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "CallerReference": "RoutingControlCell1",
        "HealthCheckConfig": {
            "Type": "RECOVERY_CONTROL",
            "Inverted": false,
            "Disabled": false,
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
        },
        "HealthCheckVersion": 1
    }
}
```

```
aws route53 create-health-check --caller-reference RoutingControlCell2 \
    --health-check-config \
```

```
    Type=RECOVERY_CONTROL,RoutingControlArn=arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567
```

```
{
    "Location": "https://route53.amazonaws.com/2015-01-01/healthcheck/11111aaaa-bbbb-
cccc-dddd-fffff22222",
    "HealthCheck": {
        "Id": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
        "CallerReference": "RoutingControlCell2",
        "HealthCheckConfig": {
            "Type": "RECOVERY_CONTROL",
            "Inverted": false,
            "Disabled": false,
            "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
        },
        "HealthCheckVersion": 1
    }
}
```

5c. 2 つのフェイルオーバー DNS レコードを作成し、それぞれにヘルスチェックを関連付けます。

Route 53 CLI を使用して、Route 53 でフェイルオーバー DNS レコードを作成します。レコードを作成するには、change-resource-record-sets コマンドの Amazon Route 53 AWS CLI Command Reference の指示に従います。レコードには、各セルの DNS 値と、Route 53 がヘルスチェックに作成した対応する `HealthCheckID` 値を指定します (6b を参照)。

プライマリセルの場合:

```
{
    "Name": "myapp.yourdomain.com",
    "Type": "CNAME",
    "SetIdentifier": "primary",
    "Failover": "PRIMARY",
    "TTL": 0,
    "ResourceRecords": [
        {
            "Value": "cell1.yourdomain.com"
        }
    ],
```

```
    "HealthCheckId": "xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

セカンダリセルの場合:

```
{
    "Name": "myapp.yourdomain.com",
    "Type": "CNAME",
    "SetIdentifier": "secondary",
    "Failover": "SECONDARY",
    "TTL": 0,
    "ResourceRecords": [
        {
            "Value": "cell2.yourdomain.com"
        }
    ],
    "HealthCheckId": "yyyyyy-yyyy-yyyy-yyyy-yyyyyyyyyyyy"
}
```

ここで、プライマリセルからセカンダリセルにフェイルオーバーするには、ステップ 4b の CLI の例に従って、`RoutingControlCell1` を `OFF` に、`RoutingControlCell2` を `ON` にします。

を使用してルーティングコントロールと状態を一覧表示および更新する AWS CLI

Amazon Application Recovery Controller (ARC) リソース (クラスター、ルーティングコントロール、コントロールパネルなど) を作成したら、クラスターを操作してフェイルオーバーしているルーティングコントロールの状態を一覧表示および更新できます。

ARC は、作成したクラスターごとに、5 つの AWS リージョンに 1 つずつ、クラスターエンドポイントのセットを提供します。ルーティングコントロールの状態を取得または設定するためにクラスターを呼び出すときは、これらのリージョンエンドポイント (AWS リージョン およびエンドポイント URL) On のいずれかを指定する必要がありますOff。を使用する場合 AWS CLI、リージョンエンドポイントに加えて、ルーティングコントロールの状態を取得または更新するには、このセクションの例に示すように、リージョンエンドポイント`--region`の も指定する必要があります。

どのリージョンクラスターエンドポイントも使用可能です。システムがリージョンエンドポイントをローテーションし、利用可能な各エンドポイントで再試行する準備をしておくことをお勧めします。クラスターエンドポイントを順番に試行するコードサンプルについては、「AWS SDKsアクション」を参照してください。

の使用の詳細については AWS CLI、 AWS CLI 「 コマンドリファレンス」を参照してください。ルーティング制御 API アクションのリストと詳細情報へのリンクについては、「ルーティング制御 API オペレーション」を参照してください。

> **Important**
>
> Amazon Route 53 コンソールでルーティングコントロールの状態を更新できますが、 AWS CLI または AWS SDK を使用してルーティングコントロールの状態を更新することをお勧めします。ARC は、トラフィックの再ルーティングやセル間のフェイルオーバーを可能にする ARC ルーティングコントロールデータプレーンにより、きわめて高い信頼性を実現します。ARC をフェイルオーバーに使用することに関するその他の推奨事項については、「ARC でのルーティングコントロールに関するベストプラクティス」を参照してください。

ルーティングコントロールを作成すると、状態は `Off` に設定されます。つまり、そのルーティングコントロールのターゲットセルには、トラフィックはルーティングされません。ルーティングコントロールの状態を確認するには、`get-routing-control-state` コマンドを実行します。

指定するリージョンとエンドポイントを判断するには、`describe-clusters` コマンドを実行して `ClusterEndpoints` を表示します。各 `ClusterEndpoint` にはリージョンとそれに対応するエンドポイントが含まれ、これらを使用してルーティングコントロールの状態を取得または更新できます。DescribeCluster はリカバリコントロール設定 API オペレーションです。ARC リージョンクラスターエンドポイントのローカルコピーをブックマークに保存するか、エンドポイントを再試行するために使用する自動化コードの中にハードコードしておくことを推奨します。

1. ルーティングコントロールを一覧表示する

信頼性の高い ARC データプレーンエンドポイントを使用して、ルーティングコントロールとルーティングコントロールの状態を表示できます。

1. 特定のコントロールパネルのルーティングコントロールを一覧表示します。コントロールパネルを指定しないと、`list-routing-controls` はクラスター内のすべてのルーティングコントロールを返します。

```
aws route53-recovery-cluster list-routing-controls --control-panel-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456 \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{
    "RoutingControls": [{
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ControlPanelName": "ExampleControlPanel",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "RoutingControlName": "RCOne",
        "RoutingControlState": "On"
    },
    {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::023759465626:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ControlPanelName": "ExampleControlPanel",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::023759465626:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
zzzzxxxxyyyy123456",
        "RoutingControlName": "RCTwo",
        "RoutingControlState": "Off"
    }
]
```

2. ルーティングコントロールを取得する

2. ルーティングコントロールの状態を取得します。

```
aws route53-recovery-cluster get-routing-control-state --routing-control-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567 \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
    "RoutingControlName": "RCOne",
    "RoutingControlState": "On"
}
```

2. ルーティングコントロールを更新する

ルーティングコントロールによって制御されているターゲットエンドポイントにトラフィックをルーティングするには、ルーティングコントロールの状態を On に更新します。update-routing-control-state コマンドを実行してルーティングコントロールの状態を更新します。(リクエストが成功すると、応答は空になります)。

2a. ルーティングコントロールの状態を更新します。

```
aws route53-recovery-cluster update-routing-control-state \
        --routing-control-arn \
        arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567 \
        --routing-control-state On \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{}
```

1 回の API コール (update-routing-control-states) で、複数のルーティングコントロールを同時に更新できます (リクエストが成功すると、応答は空になります)。

2b. 複数のルーティングコントロールの状態を一度に更新します (バッチ更新)。

```
aws route53-recovery-cluster update-routing-control-states \
        --update-routing-control-state-entries \
        '[{"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "RoutingControlState": "Off"}, \
        {"RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
hijklmnop987654321",
        "RoutingControlState": "On"}]' \
        --region us-west-2 \
        --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

```
{}
```

ARC でルーティングコントロールコンポーネントを操作する

トピック

- ARC でルーティングコントロールのコンポーネントを作成する
- ARC でルーティングコントロール状態を表示および更新する
- ルーティングコントロールの安全ルールを作成する
- ARC でクラスターのクロスアカウントをサポート

ARC でルーティングコントロールのコンポーネントを作成する

このセクションでは、Amazon Application Recovery Controller (ARC) でルーティングコントロールを操作するためのクラスター、ルーティングコントロール、ヘルスチェック、コントロールパネルを作成する方法について説明します。

まず、ルーティングコントロールとそれらをグループ化するのに使用するコントロールパネルをホストするクラスターを作成します。次に、ルーティングコントロールとヘルスチェックを作成して、トラフィックをあるセルから別のセルにフェイルオーバーするよう、再ルーティングできるようにします。例えば、トラフィックがバックアップのレプリカに送られるようにします。

作成するクラスターごとに時間単位で課金されることに注意してください。通常、アプリケーションのリカバリコントロール管理用のルーティングコントロールとコントロールパネルをホストするのに必要なクラスターは、1 つだけです。さらに、 を使用してリソース共有を設定して AWS Resource Access Manager、1 つのクラスターが複数の が所有するルーティングコントロールやその他の ARC リソースをホストできるようにします AWS アカウント。ARC でのリソース共有については、「ARC でクラスターのクロスアカウントをサポート」を参照してください。料金の詳細については、「Amazon Application Recovery Controller (ARC) の料金」を参照してください。

トラフィックをフェイルオーバーするルーティングコントロールを使用するには、アプリケーション内リソースの Amazon Route 53 DNS レコードに関連付けるルーティングコントロールのヘルスチェックを作成します。例として、アプリケーションのプライマリセルとして設定したセルと、フェイルオーバー先のセカンダリセルとして設定したセルの 2 つのセルがあるとします。

フェイルオーバーのヘルスチェックを設定するには、以下を実行してください。

1. 各セルにルーティングコントロールを作成します。
2. 各ルーティングコントロールにヘルスチェックを作成します。

3. 2 つの DNS レコード (例えば、2 つの DNS フェイルオーバーレコード) を作成し、それぞれにヘルスチェックを関連付けます。

ルーティングコントロールを作成する別のシナリオとしては、ゲートルールである安全ルールを作成する場合があります。この場合、ルーティングコントロールはゲートのルーティングコントロールとして使用するため、ルーティングコントロールをヘルスチェックと DNS レコードに関連付ける必要はありません。詳細については、「ルーティングコントロールの安全ルールを作成する」を参照してください。

ARC コンソールでルーティングコントロールのコンポーネントを作成する手順については、以下のセクションに記載しています。ARC でリカバリコントロール設定の API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

ARC でクラスターを作成する

ARC でルーティングコントロールとコントロールパネルをホストするには、クラスターを作成する必要があります。

クラスターは、冗長なリージョンエンドポイントのセットであり、これに対してAPI コールを実行して 1 つ以上のルーティングコントロールの状態を更新したり取得したりできます。1 つのクラスターで複数のルーティングコントロールをホストできます。

> **Important**
>
> 作成するクラスターごとに時間単位で課金されることに注意してください。1 つのクラスターで、アプリケーションのリカバリコントロール管理に通常十分な数のルーティングコントロールとコントロールパネルをホストできます。

クラスターを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [クラスター] を選択します。
3. [作成] を選択し、クラスターの名前を入力します。
4. [クラスターを作成] を選択します。

ARC でルーティングコントロールを作成する

トラフィックをルーティングする各セルに対してルーティングコントロールを作成します。たとえば、復旧可能性のためにサイロ化されたリソースを持つアプリケーションがある場合、各リージョン内の各アベイラビリティーゾーンにセルがあり AWS リージョン、ネストされたセルがある可能性があります。このシナリオでは、各セルと各ネストされたセルにルーティングコントロールを作成します。

ルーティングコントロールを作成する際、ルーティングコントロールの名前は各コントロールパネル内で一意の名前でなければなりません。

トラフィックの再ルーティングに使用するルーティングコントロールを作成したら、それぞれのルーティングコントロールをヘルスチェックに関連付けます。そうすることで、各ルーティングコントロールに関連付けた DNS レコードに基づいて、トラフィックをセルにルーティングできます。安全ルールとしてゲートルールを設定してゲートルーティングコントロールを作成する場合は、ルーティングコントロールにヘルスチェックを追加しないでください。

ルーティングコントロールを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。
3. [ルーティングコントロール] ページで、[作成] を選択し、[ルーティングコントロール] を選択します。
4. ルーティングコントロールの名前を入力して、コントロールを追加するクラスターを選択し、デフォルトのコントロールパネルを使用するなど、既存のコントロールパネルにクラスターを追加します。もしくは、新しいコントロールパネルを作成します。
5. 新しいコントロールパネルを作成する場合は、コントロールパネルを作成するクラスターを選択し、コントロールパネルの名前を入力します。
6. [ルーティングコントロールを作成] を選択します。
7. 手順に従って、ルーティングコントロールに名前を付けて作成します。

ARC でルーティングコントロールのヘルスチェックを作成する

トラフィックの再ルーティングに使用する各ルーティングコントロールに、ルーティングコントロールのヘルスチェックを関連付けます。次に、各ヘルスチェックに Amazon Route 53 DNS レコード (フェイルオーバー DNS レコードなど) を設定します。そうすると、関連するルーティングコント

ロールの状態を更新して、On や Off に設定するだけで、Amazon Application Recovery Controller (ARC) のトラフィックを再ルーティングできます。

> **Note**
>
> 既存のルーティングコントロールのヘルスチェックを編集して、別のルーティングコントロールに関連付けることはできません。

ルーティングコントロールのヘルスチェックを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。
3. [ルーティングコントロール] ページで、[ルーティングコントロール] を選択します。
4. [ルーティングコントロール] の詳細ページで、[ヘルスチェックの作成] を選択します。
5. ヘルスチェックの名前を入力し、[作成] を選択します。

次に、Route 53 DNS レコードを作成し、ルーティングコントロールのヘルスチェックをそれぞれのレコードに関連付けます。例えば、ルーティング制御のヘルスチェックを関連付けたい DNS フェイルオーバーレコードが 2 つあるとします。ARC がルーティングコントロールを使用してトラフィックを正しくフェイルオーバーするには、まず、Route 53 に 2 つのフェイルオーバーレコード (プライマリとセカンダリ) を作成します。DNS フェイルオーバーレコードの設定に関する詳細については、「ヘルスチェックの概念」を参照してください。

プライマリフェイルオーバーレコードを作成すると、値は次のようになります。

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Primary
Failover: Primary
TTL: 0
Resource Records:
Value: cell1.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

セカンダリフェイルオーバーレコードの値は、次のようになります。

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Secondary
Failover: Secondary
TTL: 0
Resource Records:
Value: cell2.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

ここで、障害が発生したためにトラフィックを再ルーティングしたいとしましょう。そのためには、関連するルーティングコントロールの状態を更新して、プライマリルーティングコントロールの状態を OFF に、セカンダリルーティングコントロールの状態を ON に変更します。これを行うと、関連するヘルスチェックによってプライマリレプリカへのトラフィックの送信が停止され、代わりにセカンダリレプリカへルーティングされます。ルーティング制御によるトラフィックのフェイルオーバーの詳細については、「ARC API を使用してルーティングコントロールの状態を取得および更新する (推奨)」を参照してください。

ARC API オペレーションを使用してルーティングコントロールおよび関連するヘルスチェックを作成するための AWS CLI コマンドの例については、「」を参照してくださいで ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI。

ARC でコントロールパネルを作成する

Amazon Application Recovery Controller (ARC) のコントロールパネルで、関連するルーティングコントロールをグループ化できます。コントロールパネルでは、フェイルオーバーの範囲に応じて、アプリケーション内のマイクロサービス、アプリケーション全体、またはアプリケーションのグループに対応するルーティングコントロールを設定できます。ルーティングコントロールをコントロールパネルにグループ化することの利点は、コントロールパネルと安全ルールを併用することで、トラフィックのルーティング変更を防止できる点にあります。

クラスターを作成すると、ARC でデフォルトのコントロールパネルが作成されます。デフォルトのコントロールパネルをルーティングコントロールに使用することも、複数のコントロールパネルを作成してルーティングコントロールをグループ化することもできます。コントロールパネル名は ASCII 文字のみサポートされることに注意してください。

ARC コンソールでコントロールパネルを作成する手順については、このセクションに記載しています。ARC でリカバリコントロール設定の API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

コントロールパネルを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。
3. [ルーティングコントロール] ページで、[作成] を選択し、[コントロールパネル] を選択します。
4. コントロールパネルを作成するクラスターを選択し、コントロールパネルの名前を入力します。
5. [コントロールパネルを作成] を選択します。

ARC でルーティングコントロール状態を表示および更新する

このセクションでは、Amazon Application Recovery Controller (ARC) でルーティングコントロールの状態を表示および更新する方法について説明します。ルーティングコントロールは、リカバリグループ内のセルへのトラフィックフローを管理するシンプルなオン/オフスイッチです。セルは通常 AWS リージョン、リソースを含むアベイラビリティーゾーンです。ルーティングコントロールの状態が On の場合、トラフィックはそのルーティングコントロールによって制御されているセルに流れます。

論理的なフェイルオーバーグループであるコントロールパネルに、ルーティングコントロールをグループ化します。例えば、コンソールでコントロールパネルを開くと、グループ化されたルーティングコントロールを一度に表示して、トラフィックがどこに流れているかを確認できます。

ルーティングコントロールの状態は、ARC コンソールまたは ARC API を使用して更新できますが、API を使用してルーティングコントロールの状態を更新することをお勧めします。まず、ARC はこれらのアクションを実行するデータプレーン内の API に関して、非常に高い信頼性があります。この点が重要となるのはルーティングコントロールの状態を変更する際です。ルーティングの状態変更は、アプリケーションのトラフィックを再ルーティングしてセル間でフェイルオーバーするためです。さらに、API を使用すれば、接続先のクラスターエンドポイントが使用できない場合、必要に応じて別のクラスターエンドポイントにローテーションで接続を試みることができます。

1 つのルーティングコントロールの状態を更新することも、複数のルーティングコントロールの状態を同時に更新することもできます。例えば、アプリケーションのレイテンシーが増大しているアベイラビリティーゾーンなど、あるルーティングコントロールの状態を `Off` に設定して、あるセルにトラフィックが流れないようにしたい場合が考えられます。同時に、別のルーティングコントロールの状態を On に設定して、別のセルまたは別のアベイラビリティーゾーンへのトラフィックフローを開始したい場合、このシナリオでは、両方のルーティングコントロールの状態を同時に更新して、トラフィックを続けて流すことができます。

トピック

- ARC API を使用してルーティングコントロールの状態を取得および更新する (推奨)
- でのルーティングコントロールの状態の取得と更新 AWS マネジメントコンソール

ARC API を使用してルーティングコントロールの状態を取得および更新する (推奨)

Amazon Application Recovery Controller (ARC) API オペレーションを使用して、 AWS CLI コマンドを使用するか、いずれかの AWS SDKs で ARC API オペレーションを使用するように開発したコードを使用して、ルーティングコントロールの状態を取得または更新することをお勧めします。ルーティング制御の状態を操作するには、 AWS マネジメントコンソールを使用するのではなく、CLI またはコードで API オペレーションを使用することをお勧めします。

ルーティングコントロールは高可用性クラスターに保存されるため、ARC は API を使用してルーティングコントロールの状態を更新することにより、セル (AWS リージョン) 間のフェイルオーバーの信頼性を大きく高めます。ARC では、5 つのリージョンクラスターエンドポイントのうち少なくとも 3 つのエンドポイントに常にアクセスでき、ルーティングコントロールの状態を変更できます。API を使用してルーティング制御の状態を取得または変更するには、いずれかのリージョンクラスターエンドポイントに接続します。エンドポイントが使用できない場合は、別のクラスターエンドポイントに接続してみてください。

クラスターにおけるリージョンクラスターエンドポイントのリストは、Route 53 コンソールまたは API アクション DescribeCluster を使用して確認できます。クラスターのエンドポイントは、定期的なメンテナンスや更新により、使用可能状態と使用不可状態が切り替わるため、ルーティングコントロールの状態を取得したり変更したりするプロセスは、必要に応じて各エンドポイントを交代で試す必要があります。

ARC API オペレーションを使用してルーティングコントロールの状態を取得および更新し、リージョンクラスターエンドポイントを操作する詳しい情報とコード例を提供しています。詳細については次を参照してください:

- リージョンクラスターエンドポイント間をローテーションして、ルーティングコントロールの状態を取得および設定する方法を示すコード例については、「AWS SDKsアクション」を参照してください。
- を使用してルーティングコントロールの状態を取得および更新 AWS CLI する方法については、「」を参照してくださいを使用してルーティングコントロールと状態を一覧表示および更新する AWS CLI。

でのルーティングコントロールの状態の取得と更新 AWS マネジメントコンソール

AWS マネジメントコンソールでルーティングコントロールの状態を取得および更新できます。ただし、コンソールでは異なるリージョンクラスターエンドポイントを選択できないことに注意してください。つまり、Amazon Application Recovery Controller (ARC) API を使用する場合とは異なり、コンソールではクラスターエンドポイントを選択してローテーションするプロセスはありません。さらに、ARC のデータプレーンでは非常に高い信頼性を提供する一方で、コンソールの可用性は高くありません。これらの理由により、運用オペレーションでルーティングコントロールの状態を取得および更新するには、ARC API を使用することをお勧めします。

ARC をフェイルオーバーに使用することに関するその他の推奨事項については、「ARC でのルーティングコントロールに関するベストプラクティス」を参照してください。

コンソールでルーティングコントロールを表示および更新するには、以下の手順に従ってください。

ルーティングコントロールの状態を取得するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。
3. リストからコントロールパネルを選択し、ルーティングコントロールを表示します。

1 つ以上のルーティングコントロールの状態を更新するには

1. https://console.aws.amazon.com/route53/home で Amazon Route 53 コンソールを開きます。
2. [アプリケーションリカバリコントローラー] で、[ルーティングコントロール] を選択します。
3. [アクション] を選択し、[トラフィックルーティングを変更] を選択します。
4. アプリケーションのトラフィックを流す場所、または流れを止める場所に応じて、1 つ以上のルーティングコントロールの状態を `Off` または `On` に更新します。
5. テキストボックスに「`confirm`」と入力します。
6. [トラフィックルーティングを更新] を選択します。

ルーティングコントロールの安全ルールを作成する

複数のルーティングコントロールを同時に使用する場合、意図しない結果が生じないよう安全対策を講じる必要があることがあります。例えば、アプリケーションのすべてのルーティング制御を誤って

オフにすると、フェイルオープンシナリオになってしまうのを防ぎたいケースが考えられます。あるいは、自動化によるトラフィックの再ルーティングを防ぐなど、一連のルーティングコントロールを無効にするマスターオン/オフスイッチを実装したい場合もあるでしょう。ARC でルーティングコントロールにこのような安全対策を講じるには、安全ルールを作成します。

ルーティングコントロール、ルール、その他の指定したオプションを組み合わせて、ルーティングコントロールの安全ルールを設定します。安全ルールは、それぞれ 1 つのコントロールパネルに関連付けられますが、1 つのコントロールパネルに複数の安全ルールを設定できます。安全ルールを作成する際、安全ルールの名前は各コントロールパネル内で一意でなければならないことに注意してください。

トピック

- 安全ルールのタイプ
- コンソールで安全ルールを作成する
- コンソールで安全ルールを編集または削除する
- 安全ルールを上書きしてトラフィックを再ルーティングする

安全ルールのタイプ

安全ルールには、アサーションルールとゲートルールの 2 種類があり、これらを使用してフェイルオーバーをさまざまな方法で保護できます。

アサーションルール

アサーションルールでは、1 つまたは一連のルーティングコントロールの状態を変更すると、ARC はルールを作成したときに設定した基準を満たす必要があります。基準が満たされていない場合、ルーティングコントロールの状態は変更できません。

これが役立つ例としては、フェイルオープンシナリオを防ぐ場合です。例えば、あるセルへのトラフィックの流れを停止しても、別のセルへトラフィックの流れが開始しないというシナリオです。これを回避するために、アサーションルールでは、コントロールパネルにある一連のルーティングコントロールのうち、少なくとも 1 つのルーティングコントロールが常時 On に設定されていることを確認します。これにより、トラフィックはアプリケーションの少なくとも 1 つのリージョンまたはアベイラビリティーゾーンに流れるようになります。

この条件を適用するためのアサーションルールを作成する AWS CLI コマンドの例を確認するには、「 で安全ルールを作成する」を参照してくださいで ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI。

アサーションルール API オペレーションプロパティの詳細は、Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイドの「AssertionRule」を参照してください。

ゲートルール

ゲートルールでは、一連のルーティングコントロールを全体的にオン/オフに切り替えることができるため、ルーティングコントロールの状態が変更できるかどうかは、ルールで指定する一連の基準に基づいて実行されます。最も単純な基準は、スイッチに指定する 1 つのルーティングコントロールが ON もしくは OFF に設定されているかどうかです。

これを実装するには、スイッチ全体として使用するゲートルーティングコントロールと、さまざまなリージョンやアベイラビリティーゾーンへのトラフィックフローを制御するターゲットルーティングコントロールを作成します。次に、ゲートルールに設定したターゲットルーティングコントロールの状態が手動または自動で更新されないように、ゲートルーティングコントロールの状態を Off に設定します。更新を許可する場合は On に設定します。

このような全体的なスイッチを実装するゲートルールを作成する AWS CLI コマンドの例については、「 で安全ルールを作成する」を参照してくださいで ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI。

ゲートルール API オペレーションプロパティの詳細は、「Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイド」の「GatingRule」を参照してください。

コンソールで安全ルールを作成する

このセクションの手順では、ARC コンソールで安全ルールを作成する方法について説明します。アサーションルールを作成する場合やゲートルールを作成する場合と手順は似ています。異なる点は手順をご確認ください。

Amazon Application Recovery Controller (ARC) でリカバリおよびルーティングコントロール API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

安全ルールを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。

3. [ルーティングコントロール] ページで、[コントロールパネル] を選択します。
4. [コントロールパネル] の詳細ページで、[アクション] を選択し、[安全ルールを追加] を選択します。
5. 追加するルールのタイプ ([アサーションルール] または [ゲートルール]) を選択します。
6. 名前を選択し、必要に応じて待機期間を変更します。
7. 安全ルールの設定オプションを指定します。

   - アサーションルールには、アサートされたルーティングコントロールを指定します。
   - ゲートルールには、ゲートルーティングコントロールとターゲットルーティングコントロールを指定します。

   どちらのルールでも、タイプとしきい値を選択し、ルールを逆にするかどうかを選択して、ルール設定を指定します。

   > **Note**
   >
   > アサーションルールの指定について、詳しくは Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイドに記載されている AssertionRule オペレーションに関する情報を参照してください。ゲートルールの指定について、詳しくは「Amazon Application Recovery Controller ルーティングコントロール API リファレンスガイド」に記載されている「GatingRule」オペレーションに関する情報を参照してください。

8. [作成] を選択します。

コンソールで安全ルールを編集または削除する

このセクションの手順では、ARC コンソールで安全ルールを編集または削除する方法について説明します。名前の変更や待機期間の更新など、安全ルールでは限定的な編集のみ行えます。その他の変更を行うには、安全ルールを削除して再作成してください。

Amazon Application Recovery Controller (ARC) で API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

安全ルールを削除するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [ルーティングコントロール] を選択します。
3. [ルーティングコントロール] ページで、[コントロールパネル] を選択します。
4. [コントロールパネル] の詳細ページで、[安全ルール] を選択し、[削除] または [編集] を選択します。

安全ルールを上書きしてトラフィックを再ルーティングする

設定した安全ルールによって実行される、ルーティングコントロールの安全対策をバイパスするシナリオについて説明します。例えば、ディザスタリカバリのためにフェイルオーバーを迅速に行いたい場合や、トラフィックの経路変更に必要なルーティングコントロール状態の更新が、1 つ以上の安全ルールによって予期せず妨げられる場合などです。このような「Break Glass」シナリオでは、1 つ以上の安全ルールを上書きしてルーティングコントロールの状態を変更し、アプリケーションをフェイルオーバーできます。

`safety-rules-to-override` パラメータで `update-routing-control-state`または `update-routing-control-states` AWS CLI コマンドを使用して、ルーティングコントロールの状態 (または複数のルーティングコントロールの状態) を更新するときに、安全ルールをバイパスできます。上書きしたい安全ルールの Amazon リソースネーム (ARN) を使用してパラメータを指定するか、2 つ以上の安全ルールを上書きする場合は ARN のカンマ区切りリストを指定します。

安全ルールがルーティングコントロール状態の更新をブロックする場合、エラーメッセージには更新をブロックしたルールの ARN が表示されます。そのため、ARN をメモしておき、安全ルールの上書きパラメータを使用してルーティングコントロール状態の CLI コマンドに指定できます。

> **Note**
>
> 更新するルーティングコントロールには複数の安全ルールが設定されている場合があるため、CLI コマンドを実行して 1 つの安全ルールの上書きでルーティングコントロールの状態を更新しても、別の安全ルールが更新をブロックしているというエラーが発生する可能性があります。更新コマンドが正常に完了するまで、更新コマンドで上書きするルールのリストに安全ルール ARN をカンマで区切って追加し続けます。

API と SDK で SafetyRulesToOverride プロパティを使用する方法について、詳しくは「UpdateRoutingControlState」を参照してください。

以下に、安全ルールを上書きしてルーティングコントロールの状態を更新する、2 つの CLI コマンドの例を示します。

1 つの安全ルールを上書きする

```
aws route53-recovery-cluster --region us-west-2 update-routing-control-state \
    --routing-control-arn \
    arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/
routingcontrol/abcdefg1234567 \
    --routing-control-state On \
    --safety-rules-to-override arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
yyyyyyy8888888 \
    --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

2 つの安全ルールを上書きする

```
aws route53-recovery-cluster --region us-west-2 update-routing-control-state \
    --routing-control-arn \
    arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/
routingcontrol/abcdefg1234567 \
    --routing-control-state On \
    --safety-rules-to-override "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
yyyyyyy8888888" \
    "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/safetyrule/
qqqqqqq7777777"
    --endpoint-url https://host-dddddd.us-west-2.example.com/v1
```

ARC でクラスターのクロスアカウントをサポート

Amazon Application Recovery Controller (ARC) は と統合 AWS Resource Access Manager してリソース共有を有効にします。 AWS RAM は、他の AWS アカウント または を通じてリソースを共有できるサービスです AWS Organizations。ARC ルーティングコントロールでは、クラスターリソースを共有できます。

では AWS RAM、リソース共有を作成して、所有しているリソースを共有します。リソース共有では、共有対象のリソースと、共有先である参加者を指定します。参加者には以下が含まれます。

- の所有者の組織 AWS アカウント 内外に固有 AWS Organizations
- の組織内の組織単位 AWS Organizations
- の組織全体 AWS Organizations

詳細については AWS RAM、AWS RAM 「 ユーザーガイド」を参照してください。

AWS Resource Access Manager を使用して ARC のアカウント間でクラスターリソースを共有することで、1 つのクラスターを使用して、複数の異なる が所有するコントロールパネルとルーティングコントロールをホストできます AWS アカウント。クラスターを共有する場合、指定した他の AWS アカウント はクラスターを使用して独自のコントロールパネルとルーティングコントロールをホストできるため、異なるチーム間でルーティング機能をより柔軟に制御できます。

AWS RAM は、 AWS お客様がリソースを安全に共有できるようにするサービスです AWS アカウント。を使用すると AWS RAM、IAM ロールとユーザーを使用して AWS Organizations、 の組織または組織単位 (OUs) 内のリソースを共有できます。 AWS RAM は、クラスターを共有するための一元化され制御された方法です。

クラスターを共有すると、組織が必要とするクラスターの総数を減らせます。共有クラスターを使用すると、クラスターを実行するための総コストを複数のチームに割り振ることが可能で、低コストで ARC の利点を最大化できます。(クラスターでホストされるリソースを作成しても、所有者や参加者に追加コストは発生しません)。アカウント間でクラスターを共有すると、複数のアプリケーションを ARC にオンボーディングするプロセスも簡単になります。特に、多数のアプリケーションが複数のアカウントや運用チームに分散している場合に有効です。

ARC でクロスアカウント共有を開始するには、 AWS RAMでリソース共有を作成します。リソース共有は、アカウントが所有するクラスターを共有する権限を持つ参加者を指定します。その後、参加者は、クラスター内で、 を使用するか、 AWS Command Line Interface AWS マネジメントコンソール または AWS SDKs を使用して ARC API オペレーションを実行することで、コントロールパネルやルーティングコントロールなどのリソースを作成できます。

このトピックでは、所有しているリソースの共有方法と、共有されているリソースの使用方法を説明します。

内容

- クラスター共有の前提条件

- クラスターの共有
- 共有クラスターの共有解除
- 共有クラスターの識別
- 共有クラスターの責任とアクセス許可
- 費用請求
- クォータ

クラスター共有の前提条件

- クラスターを共有するには、 でクラスターを所有している必要があります AWS アカウント。つまり、自分のアカウントにそのリソースが割り当てられているか、プロビジョニングされている必要があります。自分自身が共有を受けているクラスターは共有できません。
- 組織または AWS Organizations内の組織単位とクラスターを共有するには、 AWS Organizations との共有を有効にする必要があります。詳細については、AWS RAM ユーザーガイドの「AWS Organizationsで共有を有効化する」を参照してください。
- AWS RAM クラスターなどのグローバルリソースの リソース共有は、米国東部 (バージニア北部) リージョン (us-east-1) で作成する必要があります。

クラスターの共有

所有するクラスターを共有すると、クラスターの共有先に指定された参加者は、そのクラスター内で独自の ARC リソースを作成してホストできます。

クラスターを共有するには、リソース共有に追加する必要があります。リソース共有とは、 AWS アカウント間で自身のリソースを共有するための AWS RAM リソースです。リソース共有では、共有対象のリソースと、共有先の参加者を指定します。クラスターを共有するには、新しいリソース共有を作成するか、リソースを既存のリソース共有に追加します。新しいリソース共有を作成するには、 AWS RAM コンソールを使用するか、 AWS Command Line Interface または AWS SDKsで AWS RAM API オペレーションを使用します。

の組織に属 AWS Organizations していて、組織内での共有が有効になっている場合、組織内の参加者には共有クラスターへのアクセス権が自動的に付与されます。それ以外の場合、参加者はリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有クラスターに対するアクセス許可が付与されます。

所有しているクラスターを共有するには、 AWS RAM コンソールを使用するか、 AWS CLI または SDKs で AWS RAM API オペレーションを使用します。

AWS RAM コンソールを使用して所有しているクラスターを共有するには

「AWS RAM ユーザーガイド」の「リソース共有の作成」を参照してください。

を使用して所有しているクラスターを共有するには AWS CLI

create-resource-share コマンドを使用します。

クラスターを共有するアクセス許可の付与

アカウント間でクラスターを共有するには、クラスターを共有する IAM プリンシパルのアクセス許可が必要です AWS RAM。

AmazonRoute53RecoveryControlConfigFullAccess マネージド IAM ポリシーを使用して、IAM プリンシパルが共有クラスターを共有して使用するために必要なアクセス許可を持っていることを確認することをお勧めします。

カスタム IAM ポリシーを使用してクラスターを共有するには、そのクラスターの route53-recovery-control-config:PutResourcePolicy、route53-recovery-control-config:GetResourcePolicy、および route53-recovery-control-config:DeleteResourcePolicy のアクセス許可が必要です。PutResourcePolicy および DeleteResourcePolicy はアクセス許可のみの IAM アクションです。これらのアクセス許可 AWS RAM なしで を介してクラスターを共有しようとすると、エラーが発生します。

IAM AWS Resource Access Manager の使用方法の詳細については、AWS RAM 「 ユーザーガイド」の「IAM AWS Resource Access Manager の使用方法」を参照してください。

共有クラスターの共有解除

クラスターの共有を解除すると、次のことが参加者と所有者に適用されます。

- 現在の参加者のリソースは、共有解除されたクラスターに残ります。
- 参加者は引き続き、共有解除されたクラスターのルーティングコントロール状態を更新して、アプリケーションフェイルオーバーのルーティングを管理できます。
- 参加者は共有解除されたクラスターに新しいリソースを作成できません。
- 参加者のリソースがまだ共有解除されたクラスターにある場合、所有者はその共有クラスターを削除できません。

所有している共有クラスターの共有を解除するには、それをリソース共有から削除します。これを行うには、 AWS RAM コンソールを使用するか、 AWS CLI または SDKsで AWS RAM API オペレーションを使用します。

AWS RAM コンソールを使用して所有している共有クラスターの共有を解除するには

AWS RAM ユーザーガイド の「リソース共有の更新」を参照してください。

を使用して所有している共有クラスターの共有を解除するには AWS CLI

disassociate-resource-share コマンドを使用します。

共有クラスターの識別

所有者と参加者は、 AWS RAM内で情報を表示して、共有クラスターを識別できます。ARC コンソールと AWS CLIを使用して、共有リソースに関する情報を取得することもできます。

一般的に、共有したリソースまたは共有されたリソースの詳細については、 AWS Resource Access Manager 「 ユーザーガイド」の情報を参照してください。

- 所有者は、 AWS RAMを使用することで、他のユーザーと共有しているすべてのリソースを表示できます。詳細については、「 での共有リソースの表示 AWS RAM」を参照してください。
- 参加者として、 を使用して共有されているすべてのリソースを表示できます AWS RAM。詳細については、「 での共有リソースの表示 AWS RAM」を参照してください。

所有者は、 で情報を表示するか、ARC API オペレーション AWS Command Line Interface で AWS マネジメントコンソール を使用してクラスターを共有するかどうかを判断できます。

コンソールを使用して、所有しているクラスターが共有されているかどうかを確認するには

クラスター AWS マネジメントコンソールの詳細ページで、クラスターの共有ステータスを参照してください。

を使用して、所有しているクラスターが共有されているかどうかを確認するには AWS CLI

get-resource-policy コマンドを使用します。クラスターにリソースポリシーがある場合、コマンドはそのポリシーに関する情報を返します。

参加者がクラスターの共有を受ける際は、通常、共有を承諾する必要があります。また、クラスターの [所有者] フィールドにはクラスター所有者の説明が含まれます。

共有クラスターの責任とアクセス許可

所有者のアクセス許可

所有しているクラスターを他のユーザーと共有する場合 AWS アカウント、クラスターの使用が許可されている参加者は、クラスター内にコントロールパネル、ルーティングコントロール、その他のリソースを作成できます。

クラスター所有者は、クラスターの作成、管理、削除に責任を負います。ルーティングコントロールや安全ルールなど、参加者が作成したリソースを変更または削除できません。例えば、参加者が作成したルーティングコントロールを更新してルーティングコントロールの状態を変更できません。

ただし、自分が所有するクラスターの参加者が作成したルーティングコントロールの詳細は表示できます。たとえば、 AWS Command Line Interface または AWS SDKs を使用して ARC ルーティングコントロール API オペレーションを呼び出すことで、ルーティングコントロールの状態を表示できます。

参加者の作成したリソースを変更する必要がある場合、参加者にリソースへのアクセス許可を持つロールを IAM で設定してもらい、そのロールに自分のアカウントを追加してもらいます。

参加者のアクセス許可

一般に、参加者は、共有されたクラスター内でコントロールパネル、ルーティングコントロール、安全ルール、ヘルスチェックを作成し、使用できます。共有クラスター内のクラスターリソースの表示、変更、削除ができるのは、そのリソースを所有している場合に限られます。例えば、参加者は自分が作成したコントロールパネルの安全ルールを作成および削除できます。

以下の制限が適用されます。

- 参加者は、共有クラスターを使用して他のアカウントが作成したコントロールパネルを表示、変更、削除できません。
- 参加者は、他のアカウントが共有クラスターに作成したリソースについて、ルーティングコントロールの表示、作成、変更 (ルーティングコントロールの状態を含む) を行えません。
- 参加者は、共有クラスター内の他のアカウントが作成した安全ルールを作成、変更、表示できません。
- クラスター所有者のものであるため、参加者は共有クラスター内のデフォルトコントロールパネルにはリソースを追加できません。

前述のように、参加者は共有クラスターのデフォルトコントロールパネルにルーティングコントロールを作成できません。クラスター所有者がデフォルトコントロールパネルを所有しているためです。

ただし、クラスター所有者は、クラスターのデフォルトコントロールパネルへのアクセス許可を与えるクロスアカウント IAM ロールを作成できます。その後、所有者は参加者にロールを引き受ける許可を付与できます。これにより、参加者はデフォルトのコントロールパネルにアクセスし、所有者がロールのアクセス許可で指定した方法で使用できるようになります。

費用請求

ARC のクラスターの所有者には、そのクラスターに関連する費用が請求されます。クラスターの所有者側でも参加者側でも、クラスターでホストされるリソースの作成に追加費用はかかりません。

詳細な料金情報と例については、「Amazon Application Recovery Controller (ARC) の料金」を参照してください。

クォータ

共有クラスターで作成されたすべてのリソース (共有クラスターへのアクセス権を持つすべての参加者が作成したリソースを含む) は、そのクラスターや他のリソース (ルーティングコントロールなど) で有効なクォータにカウントされます。クラスターリソースを共有するアカウントのクォータがクラスター所有者のクォータよりも高い場合、クラスター所有者のクォータは、共有しているアカウントのクォータよりも優先されます。

この仕組みをより深く理解するために、以下の例を参照してください。リソース共有でのクォータの仕組みを説明するために、これらの例では、クラスター所有者が所有者で、クラスターが共有されているアカウントが参加者であるとします。

コントロールパネルのクォータ

クォータは、クラスターあたりの所有者の合計コントロールパネルに適用されます。

例えば、所有者のクラスターあたりのコントロールパネル数のクォータが 50 で、クラスター内に 13 のコントロールパネルがあるとします。次に、参加者がクォータを 150 に設定しているとします。このシナリオでは、参加者は共有クラスターに最大 37 個のコントロールパネル (50 ~ 13) しか作成できません。

さらに、クラスターを共有する他のアカウントもコントロールパネルを作成する場合、それらはすべて 50 のコントロールパネルのクラスター全体のクォータにもカウントされます。

ルーティングコントロールのクォータ

ルーティングコントロールには複数のクォータがあります。コントロールパネルあたりのクォータ、クラスターあたりのクォータ、安全ルールあたりのクォータです。所有者のクォータは、これらすべてのクォータに優先されます。

例えば、所有者のクラスターあたりのルーティングコントロール数のクォータが 300 で、クラスター内に既に 300 のルーティングコントロールがあるとします。次に、参加者がこのクォータを 500 に設定しているとします。このシナリオでは、参加者は共有クラスターに新しいルーティングコントロールを作成できません。

安全ルールのクォータ

クォータは、コントロールパネルのクォータごとに所有者の安全ルールに適用されます。

例えば、所有者がコントロールパネルあたりの安全ルールの数のクォータを 20 に設定し、参加者がこのクォータを 80 に設定しているとします。このシナリオでは、所有者の下限が優先されるため、参加者は共有クラスターのコントロールパネルに最大 20 個の安全ルールしか作成できません。

ルーティングコントロールクォータのリストについては、「ルーティングコントロールのクォータ」を参照してください。

Amazon Application Recovery Controller (ARC) でのルーティングコントロールのログ記録とモニタリング

AWS CloudTrail を使用して、Amazon Application Recovery Controller (ARC) のルーティング制御をモニタリングし、パターンを分析し、問題のトラブルシューティングに役立てることができます。

トピック

- AWS CloudTrail を使用した ARC API コールのログ記録

AWS CloudTrail を使用した ARC API コールのログ記録

Amazon Application Recovery Controller (ARC) は AWS CloudTrail と統合されています。これは、ユーザー、ロール、または ARC 内の AWS サービスで実行されるアクションの、レコードを提供するサービスです。CloudTrail は、ARC のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、ARC コンソールのコールと、ARC API オペレーションへのコードのコールが含まれます。

証跡を作成する場合は、ARC のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。

CloudTrail が収集した情報を使用して、ARC に対して行われたリクエスト、リクエスト元の IP アドレス、リクエストした人、リクエストが行われた日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail での ARC 情報

CloudTrail は、AWS アカウントを作成すると、その中で有効になります。ARC でアクティビティが発生すると、そのアクティビティは [イベント履歴] で AWS のその他のサービスのイベントと共に CloudTrail イベントに記録されます。最近のイベントは、AWS アカウント で表示、検索、ダウンロードできます。詳細については、「CloudTrail イベント履歴の操作」を参照してください。

ARC のイベントなど、AWS アカウントのイベントの継続的な記録に対して、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョン に適用されます。証跡は、AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づいて対応するため、他の AWS サービスを構成できます。詳細については、次を参照してください:

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- CloudTrail の Amazon SNS 通知の設定
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての ARC アクションは CloudTrail によって ログ記録されます。「Amazon Application Recovery Controller のリカバリの準備状況 API リファレンスガイド」、「Amazon Application Recovery Controller のリカバリコントロール設定 API リファレンスガイド」、「Amazon Application Recovery Controller のルーティングコントロール API リファレンスガイド」を参照してください。例えば、`CreateCluster`、`UpdateRoutingControlState`、`CreateRecoveryGroup` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます:

- リクエストが、ルート認証情報と AWS Identity and Access Management (IAM) ユーザー認証情報のどちらを使用して送信されたか。

- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

イベント履歴での ARC イベントの表示

CloudTrail では、[イベント履歴] に最近のイベントが表示されます。ARC API リクエストのイベントを表示するには、コンソールの上部にあるリージョンセレクターで [米国西部 (オレゴン)] を指定する必要があります。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。

ARC ログファイルエントリの概要

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどの情報を含みます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、ルーティングコントロール設定の `CreateCluster` アクションを行う CloudTrail ログエントリです。

```
{
  "eventVersion": "1.08",
   "userIdentity": {
     "type": "IAMUser",
     "principalId": "A1B2C3D4E5F6G7EXAMPLE",
     "arn": "arn:aws:iam::111122223333:user/smithj",
     "accountId": "111122223333",
     "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
     "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "A1B2C3D4E5F6G7EXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/smithj",
              "accountId": "111122223333",
              "userName": "smithj"
          },
          "webIdFederationData": {},
```

```
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-06-30T04:44:41Z"
            }
        }
    },
    "eventTime": "2021-06-30T04:45:46Z",
    "eventSource": "route53-recovery-control-config.amazonaws.com",
    "eventName": "CreateCluster",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "aws-cli/2.0.0 Python/3.8.2 Darwin/19.6.0 botocore/2.0.0dev7",
    "requestParameters": {
        "ClientToken": "12345abcdef-1234-5678-abcd-12345abcdef",
        "ClusterName": "XYZCluster"
    },
    "responseElements": {
        "Cluster": {
            "Arn": "arn:aws:route53-recovery-control::012345678901:cluster/abc123456-
aa11-bb22-cc33-abc123456",
            "ClusterArn": "arn:aws:route53-recovery-control::012345678901:cluster/
abc123456-aa11-bb22-cc33-abc123456",
            "Name": "XYZCluster",
            "Status": "PENDING"
        }
    },
    "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
    "eventID": "9cab44ef-0777-41e6-838f-f249example",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```

次の例は、ルーティングコントロールの UpdateRoutingControlState アクションを実行する CloudTrail ログエントリです。

```
{
    "eventVersion": "1.08",
     "userIdentity": {
       "type": "AssumedRole",
       "principalId": "A1B2C3D4E5F6G7EXAMPLE",
```

```
      "arn": "arn:aws:sts::111122223333:assumed-role/admin/smithj",
      "accountId": "111122223333",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "sessionContext": {
           "sessionIssuer": {
               "type": "Role",
               "principalId": "A1B2C3D4E5F6G7EXAMPLE",
               "arn": "arn:aws:iam::111122223333:role/admin",
               "accountId": "111122223333",
               "userName": "admin"
           },
           "webIdFederationData": {},
           "attributes": {
               "mfaAuthenticated": "false",
               "creationDate": "2021-06-30T04:44:41Z"
           }
       }
  },
  "eventTime": "2021-06-30T04:45:46Z",
  "eventSource": "route53-recovery-control-config.amazonaws.com",
  "eventName": "UpdateRoutingControl",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.50",
  "userAgent": "aws-cli/2.0.0 Python/3.8.2 Darwin/19.6.0 botocore/2.0.0dev7",
  "requestParameters": {
      "RoutingControlName": "XYZRoutingControl3",
      "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
  },
  "responseElements": {
      "RoutingControl": {
          "ControlPanelArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
          "Name": "XYZRoutingControl3",
          "Status": "DEPLOYED",
          "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
      }
  },
  "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
  "eventID": "9cab44ef-0777-41e6-838f-f249example",
  "readOnly": false,
```

```
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}
```

ルーティングコントロール用 Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

内容

- Amazon Application Recovery Controller (ARC) のルーティングコントロールが IAM と連携する仕組み
- ARC のルーティングコントロールに関するアイデンティティベースのポリシー例
- AWS Amazon Application Recovery Controller (ARC) でのルーティングコントロールの マネージドポリシー

Amazon Application Recovery Controller (ARC) のルーティングコントロールが IAM と連携する仕組み

IAM を使用して Amazon Application Recovery Controller (ARC) のルーティングコントロールへのアクセスを管理する前に、ルーティングコントロールで使用できる IAM 機能について説明します。

Amazon Application Recovery Controller (ARC) のルーティングコントロールで使用できる IAM の機能

IAM の機能	ルーティングコントロールのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし
ポリシーアクション	あり
ポリシーリソース	あり

IAM の機能	ルーティングコントロールのサポート
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	部分的
一時認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	いいえ

AWS サービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

ARC のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

ルーティングコントロール用の ARC アイデンティティベースのポリシー例は、「ARC のルーティングコントロールに関するアイデンティティベースのポリシー例」を参照してください。

ルーティングコントロール内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。

ルーティングコントロールのポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

ルーティングコントロールの ARC アクションのリストは、「サービス認証リファレンス」の「Amazon Route 53 リカバリコントロールで定義されるアクション」および「Amazon Route 53 リカバリクラスターで定義されるアクション」を参照してください。

ルーティングコントロール用の ARC ポリシーアクションは、使用中の API に応じて、アクションの前に以下のプレフィックスを使用します。

```
route53-recovery-control-config
route53-recovery-cluster
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。例えば、次の操作を実行できます。

```
"Action": [
      "route53-recovery-control-config:action1",
      "route53-recovery-control-config:action2"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、`Describe` という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "route53-recovery-control-config:Describe*"
```

ルーティングコントロール用の ARC アイデンティティベースのポリシー例は、「ARC のルーティングコントロールに関するアイデンティティベースのポリシー例」を参照してください。

ARC のポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Resource` JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

サービス認証リファレンスでは、ARC に関連する次の情報を確認できます。

リソースタイプとその ARN のリスト、および各リソースの ARN で指定できるアクションについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 Recovery コントロールで定義されるアクション
- Amazon Route 53 リカバリクラスターで定義されるアクション

ルーティングコントロール用の ARC アイデンティティベースのポリシー例は、「ARC のルーティングコントロールに関するアイデンティティベースのポリシー例」を参照してください。

ARC のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を

一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

ルーティングコントロール用の ARC の条件キーのリストについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 Recovery コントロールの条件キー
- Amazon Route 53 リカバリクラスターの条件キー

条件キーで使用できるアクションとリソースについては、「サービス認証リファレンス」の以下のトピックを参照してください。

- リソースタイプとその ARN のリストは、「Amazon Route 53 リカバリコントロールで定義されるアクション」および「Amazon Route 53 リカバリクラスターで定義されるアクション」を参照してください。
- 各リソースの ARN で指定できるアクションのリストは、「Amazon Route 53 リカバリコントロールで定義されるリソース」および「Amazon Route 53 リカバリクラスターで定義されるリソース」を参照してください。

ルーティングコントロール用の ARC アイデンティティベースのポリシー例は、「ARC のルーティングコントロールに関するアイデンティティベースのポリシー例」を参照してください。

ARC のアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセスコントロール (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/*key-name*、aws:RequestTag/*key-name*、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

ARC のルーティングコントロールには、ABAC の以下のサポートが含まれています。

- Recovery Control Config は ABAC をサポートしています。
- リカバリクラスターは ABAC をサポートしていません。

ARC での一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

ARC のクロスサービスプリンシパル権限

転送アクセスセッション (FAS) のサポート: あり

IAM エンティティ (ユーザーまたはロール) を使用して でアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。

アクションにポリシーで追加の依存アクションが必要かどうかを確認するには、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 リカバリクラスター

- Amazon Route 53 リカバリコントロール

ARC のサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

ARC のサービスにリンクされたロール

サービスリンクロールのサポート

サービスにリンクされたロールは、 サービスにリンクされた AWS サービスロールの一種です。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは AWS アカウントに表示され、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

ルーティングコントロールは、サービスにリンクされたロールを使用しません。

ARC のルーティングコントロールに関するアイデンティティベースのポリシー例

デフォルトでは、ユーザーおよびロールには、ARC リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認証リファレンス」の「Amazon Application Recovery Controller (ARC) のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- 例: ルーティングコントロール用の ARC コンソールアクセス

- 例: ルーティングコントロール設定の ARC API アクション

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが ARC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の 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。

例: ルーティングコントロール用の ARC コンソールアクセス

Amazon Application Recovery Controller (ARC) コンソールにアクセスするには、最小限のアクセス許可セットが必要です。これらのアクセス許可により、 の ARC リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

特定の API オペレーションのみへのアクセスを許可するときに、ユーザーとロールが引き続き ARC コンソールを使用できるようにするには、ARC の `ReadOnly` AWS 管理ポリシーもエンティティにアタッチします。詳細については、「IAM ユーザーガイド」の「ARC マネージドポリシー」ページまたは「ユーザーへのアクセス許可の追加 (コンソール)」を参照してください。

ユーザーに、コンソールから ARC ルーティングコントロール機能を使用するためのフルアクセスを許可するには、次のようなポリシーをユーザーにアタッチして、ARC ルーティングコントロールのリソースとオペレーションを設定するフルアクセス許可を付与します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-cluster:GetRoutingControlState",
                   "route53-recovery-cluster:UpdateRoutingControlState",
                   "route53-recovery-cluster:UpdateRoutingControlStates",
                   "route53-recovery-control-config:CreateCluster",
                   "route53-recovery-control-config:CreateControlPanel",
                   "route53-recovery-control-config:CreateRoutingControl",
                   "route53-recovery-control-config:CreateSafetyRule",
                   "route53-recovery-control-config:DeleteCluster",
```

```
                    "route53-recovery-control-config:DeleteControlPanel",
                    "route53-recovery-control-config:DeleteRoutingControl",
                    "route53-recovery-control-config:DeleteSafetyRule",
                    "route53-recovery-control-config:DescribeCluster",
                    "route53-recovery-control-config:DescribeControlPanel",
                    "route53-recovery-control-config:DescribeSafetyRule",
                    "route53-recovery-control-config:DescribeRoutingControl",
                    "route53-recovery-control-
config:ListAssociatedRoute53HealthChecks",
                    "route53-recovery-control-config:ListClusters",
                    "route53-recovery-control-config:ListControlPanels",
                    "route53-recovery-control-config:ListRoutingControls",
                    "route53-recovery-control-config:ListSafetyRules",
                    "route53-recovery-control-config:UpdateControlPanel",
                    "route53-recovery-control-config:UpdateRoutingControl",
                    "route53-recovery-control-config:UpdateSafetyRule"
                ],
                "Resource": "*"
            },
                    {
                "Effect": "Allow",
                "Action": [
                        "route53:GetHealthCheck",
                        "route53:CreateHealthCheck",
                        "route53:DeleteHealthCheck",
                        "route53:ChangeTagsForResource"
                ],
                "Resource": "*"
            }
        ]
}
```

例: ルーティングコントロール設定の ARC API アクション

ユーザーが ARC API アクションを使用して ARC ルーティングコントロール設定を操作できるようにするには、以下で説明するように、ユーザーが操作すべき API オペレーションに対応したポリシーをアタッチします。

リカバリコントロール設定に API オペレーションを使用するには、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-control-config:CreateCluster",
                   "route53-recovery-control-config:CreateControlPanel",
                   "route53-recovery-control-config:CreateRoutingControl",
                   "route53-recovery-control-config:CreateSafetyRule",
                   "route53-recovery-control-config:DeleteCluster",
                   "route53-recovery-control-config:DeleteControlPanel",
                   "route53-recovery-control-config:DeleteRoutingControl",
                   "route53-recovery-control-config:DeleteSafetyRule",
                   "route53-recovery-control-config:DescribeCluster",
                   "route53-recovery-control-config:DescribeControlPanel",
                   "route53-recovery-control-config:DescribeSafetyRule",
                   "route53-recovery-control-config:DescribeRoutingControl",
                   "route53-recovery-control-config:GetResourcePolicy",
                   "route53-recovery-control-
config:ListAssociatedRoute53HealthChecks",
                   "route53-recovery-control-config:ListClusters",
                   "route53-recovery-control-config:ListControlPanels",
                   "route53-recovery-control-config:ListRoutingControls",
                   "route53-recovery-control-config:ListSafetyRules",
                   "route53-recovery-control-config:ListTagsForResource",
                   "route53-recovery-control-config:UpdateControlPanel",
                   "route53-recovery-control-config:UpdateRoutingControl",
                   "route53-recovery-control-config:UpdateSafetyRule",
                   "route53-recovery-control-config:TagResource",
                   "route53-recovery-control-config:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

障害発生時にフェイルオーバーするようにルーティングコントロールの状態を更新するなど、リカバリクラスターデータプレーン API を使用して ARC ルーティングコントロールのタスクを実行するには、次のような ARC IAM ポリシーを IAM ユーザーにアタッチします。

`AllowSafetyRuleOverride` ブール値は、ルーティングコントロールのセーフガードとして設定した安全ルールを、上書きするアクセス許可を付与します。このアクセス許可は、「Break Glass」のシナリオで、災害などの緊急のフェイルオーバーシナリオで安全対策を回避するために必要になる場合があります。例えば、オペレーターがディザスタリカバリのためにすばやいフェイルオーバーを必要とする場合や、1 つ以上の安全規則により、トラフィックの経路変更に必要なルーティングコントロール状態の更新が、予期せず妨げられる場合などです。このアクセス許可により、オペレーターは、API コールを行ってルーティングコントロールの状態を更新するときに、オーバーライドする安全ルールを指定できるようになります。詳細については、「安全ルールを上書きしてトラフィックを再ルーティングする」を参照してください。

オペレーターにリカバリクラスターデータプレーン API の使用を許可し、かつ安全ルールのオーバーライドを防ぐときは、`AllowSafetyRuleOverrides` ブール値を `false` に設定して、次のようなポリシーをアタッチできます。オペレーターが安全ルールを上書きできるようにするには、`AllowSafetyRuleOverrides` ブール値を `true` に設定します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-cluster:GetRoutingControlState",
                "route53-recovery-cluster:ListRoutingControls"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-cluster:UpdateRoutingControlStates",
                "route53-recovery-cluster:UpdateRoutingControlState"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "Bool": {
                    "route53-recovery-cluster:AllowSafetyRulesOverrides": "false"
                }
            }
        }
    ]
}
```

AWS Amazon Application Recovery Controller (ARC) でのルーティングコントロールの マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryControlConfigFullAccess

IAM エンティティに `AmazonRoute53RecoveryControlConfigFullAccess` をアタッチできます。このポリシーは、ARC のリカバリコントロール設定を操作するアクションへの、フルアクセスを許可します。これを、リカバリコントロールの設定アクションへのフルアクセスを必要とする IAM ユーザーとその他のプリンシパルにアタッチします。

任意で、Amazon Route 53 アクションへのアクセスを追加して、ユーザーがルーティングコントロールのヘルスチェックを作成できるようにすることもできます。例えば、`route53:GetHealthCheck`、`route53:CreateHealthCheck`、`route53:DeleteHealthCheck`、のうち 1 つ以上のアクションにアクセス許可を付与できます。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryControlConfigFullAccess」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryControlConfigReadOnlyAccess

IAM エンティティに AmazonRoute53RecoveryControlConfigReadOnlyAccess をアタッチできます。これは、ルーティングコントロールとセーフティルールの設定を確認する必要があるユーザーに役立つポリシーです。このポリシーは、ARC のリカバリコントロール設定を操作するアクションへの、読み取り専用アクセスを許可します。これらのユーザーは、リカバリコントロールリソースを作成、更新、削除できません。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryControlConfigReadOnlyAccess」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryClusterFullAccess

IAM エンティティに AmazonRoute53RecoveryClusterFullAccess をアタッチできます。このポリシーは、ARC のクラスターデータプレーンを操作するアクションへの、フルアクセスを許可します。これは、ルーティングコントロールの状態を更新および取得するために、フルアクセスを必要とする IAM ユーザーとその他のプリンシパルにアタッチします。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryClusterFullAccess」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryClusterReadOnlyAccess

IAM エンティティに AmazonRoute53RecoveryClusterReadOnlyAccess をアタッチできます。このポリシーは、ARC のクラスターデータプレーンへの読み取り専用アクセスを許可します。これらのユーザーは、ルーティングコントロールの状態を取得することはできますが更新はできません。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryClusterReadOnlyAccess」を参照してください。

AWS マネージドポリシー: AmazonApplicationRecoveryControllerRegionSwitchPlanExecutionPolicy

IAM エンティティに AmazonApplicationRecoveryControllerRegionSwitchPlanExecutionPolicy をアタッチできます。このポリシーは、ARC リージョンの切り替えプランの実行と評価のアクセス許可を付与します。ポリシーをリージョン切り替えプランの実行に使用される IAM ロールにアタッチします。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- arc-region-switch:GetPlan – プリンシパルがリージョン切り替えプランの設定の詳細を取得できるようにします。
- arc-region-switch:GetPlanExecution – プリンシパルが特定のリージョン切り替えプランの実行に関する情報を取得することを許可します。
- arc-region-switch:ListPlanExecutions – プリンシパルがリージョン切り替えプランのすべての実行を一覧表示することを許可します。
- iam:SimulatePrincipalPolicy – プリンシパルが IAM ロールが実行できるアクションをシミュレートして評価することを許可します。このアクセス許可は IAM ロールのみを対象としており、リージョン切り替えプランを実行する前に必要なアクセス許可が設定されていることを確認するためにプラン評価中に使用されます。
- cloudwatch:DescribeAlarms – プリンシパルが Amazon CloudWatch アラームに関する情報を取得することを許可します。
- cloudwatch:DescribeAlarmHistory – プリンシパルが Amazon CloudWatch アラームの状態変更履歴を取得できるようにします。
- cloudwatch:GetMetricStatistics – プリンシパルが Amazon CloudWatch メトリクスの統計データを取得することを許可します。

JSON ポリシードキュメントの最新バージョンなど、ポリシーについてさらに詳しく確認するには、「AWS マネージドポリシーリファレンスガイド」の「AmazonApplicationRecoveryControllerRegionSwitchPlanExecutionPolicy」を参照してください。

ルーティングコントロールの AWS マネージドポリシーの更新

このサービスがこれらの変更の追跡を開始してからの ARC でのルーティングコントロールの AWS マネージドポリシーの更新の詳細については、「」を参照してくださいAmazon Application Recovery Controller (ARC) の AWS マネージドポリシーの更新。このページの変更に関する自動通知については、ARC の「ドキュメント履歴ページ」で RSS フィードをサブスクライブしてください。

ルーティングコントロールのクォータ

Amazon Application Recovery Controller (ARC) のルーティングコントロールには、次のクォータ (旧称「制限」) が適用されます。

エンティティ	クォータ

エンティティ	クォータ
アカウントあたりのクラスターの数	2
クラスターあたりのコントロールパネルの数	50
コントロールパネルあたりのルーティングコントロールの数	100
クラスターあたりの (すべてのコントロールパネル内の) ルーティングコントロールの総数	300
コントロールパネルあたりの安全ルールの数	20
UpdateRoutingControlStates オペレーション呼び出しあたりのルーティングコントロールの数	10
1 秒あたりのクラスターエンドポイントに対する API コールのミューテーションの数	3

ARC の準備状況チェック

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) の準備状況チェックを使用すると、アプリケーションとリソースが復旧の準備が整っているかどうかに関するインサイトを得ることができます。ARC で AWS アプリケーションをモデル化し、準備状況チェックを作成すると、チェックは AWS リソースクォータ、容量、ネットワークルーティングポリシーなど、アプリケーションに関する情報を継続

的にモニタリングします。次に、アプリケーションのレプリカにフェイルオーバーする機能に影響する変更について通知を受け取り、イベントからの復旧を選択できます。継続的に準備状況チェックを行うことにより、マルチリージョンアプリケーションがフェイルオーバートラフィックを処理できるようにスケールおよび設定されていることを確認できます。

この章では、ARC でアプリケーションをモデル化し、アプリケーションを説明するリカバリグループとセルを作成して、準備状況チェックを機能させる構造を設定する方法について説明します。次に、ステップに従って準備状況チェックと準備状況スコープを追加して、ARC がアプリケーションの準備状況を監査できるようにします。

準備状況チェックを作成すると、リソースの準備状況ステータスをモニタリングできるようになります。準備状況チェックを使用すると、スタンバイ状態のアプリケーションのレプリカとそのリソースを、本番アプリケーションの容量、ルーティングポリシー、その他の詳細な設定を反映しながら、本番のレプリカと継続的に一致させることができます。レプリカが一致しない場合には、容量を追加したり設定を変更したりすることにより、アプリケーションレプリカを再び一致させることができます。

> **Important**
>
> 準備状況チェックは、アプリケーションのレプリカの設定とランタイムの状態が一致していることを継続的に確認するときに、最も役立つサービスです。準備状況チェックは、本番のレプリカが正常かどうかを示すために使用すべきではありません。また、準備状況チェックを、災害発生時のフェイルオーバーの主要なトリガーとして使用すべきでもありません。

Amazon Application Recovery Controller (ARC) での準備状況チェックとは

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

ARC の準備状況チェックは、 AWS プロビジョニングされた容量の不一致、サービスクォータ、スロットル制限、およびチェックに含まれるリソースの設定とバージョンの不一致を継続的に (1 分間隔で) 監査します。準備状況チェックではこれらの差異がユーザーに通知されるため、各レプリカの

設定のセットアップが同じであり、ランタイム時の状態が同じであることを確認できます。準備状況チェックでは、設定したキャパシティがレプリカ間で一定であることを確認できますが、ユーザーに代わってレプリカのキャパシティを決めてくれると考えるべきではありません。例えば、別のセルが使用できなくなった場合に備えて、各レプリカの、十分なバッファ容量を備えた Auto Scaling グループのサイズを決めるには、アプリケーション要件を理解する必要があります。

クォータについては、ARC が準備状況チェックで不一致を検出すると、高いクォータに合わせて低いクォータを増やして、レプリカのクォータを調整する措置を講じることができます。クォータが一致すると、準備状況チェックのステータスが READY と表示されます (こちらは即時の更新プロセスではありません。また、合計時間は特定のリソースタイプやその他の要因に応じて変わります)。

最初のステップでは、アプリケーションを表すリカバリグループを作成するための、準備状況チェックをセットアップします。各リカバリグループには、個々の障害抑制ユニットまたはアプリケーションのレプリカのセルが含まれています。次に、アプリケーション内のリソースタイプごとにリソースセットを作成し、そのリソースセットに準備状況チェックを関連付けます。最後に、リソースを準備状況の範囲に関連付けます。そうすることで、リカバリグループ (アプリケーション) または個々のセル (レプリカ、つまりリージョンまたはアベイラビリティーゾーン (AZ)) 内のリソースに関する準備状況ステータスを取得できます。

準備状況 (つまり READY または NOT READY) は、準備状況チェックの範囲に含まれるリソースと、リソースタイプの一連のルールに基づいて決定されます。リソースタイプごとに一連の準備状況ルールがあり、ARC のチェックではこれを使ってリソースの準備状況を監査します。リソースが READY であるか否かの判断は、各準備状況ルールの定義方法に基づきます。準備状況ルールでは、通常リソースの評価が行われますが、リソースを相互に比較したり、リソースセット内の各リソースに関する特定の情報を調べたりする場合もあります。

準備状況チェックを追加することで、EventBridge、、ARC API アクションのいずれかの方法で AWS マネジメントコンソール準備状況ステータスをモニタリングできます。また、リソースの準備状況ステータスを、セルの準備状況やアプリケーションの準備状況など異なるコンテキストでモニタリングすることもできます。ARC のクロスアカウント認可機能を使用すると、1 つの AWS アカウントから分散リソースを簡単にセットアップしてモニタリングできます。

準備状況チェックを使ってアプリケーションレプリカをモニタリングする

ARC は、準備状況チェックを使用してアプリケーションのレプリカを監査し、各レプリカが同じ構成設定で同じランタイムを持つことを確認します。準備状況チェックでは、アプリケーションの AWS リソース容量、設定、 AWS クォータ、およびルーティングポリシーを継続的に監査します。この情報は、レプリカがフェイルオーバーの準備ができていることを確認するために使用できます。

準備状況チェックは、リカバリ環境がスケールされ、必要なときにフェイルオーバーできるように構成されていることを確認するのに役立ちます。

以下のセクションでは、準備状況チェックの仕組みについて詳しく説明します。

準備状況チェックとアプリケーションレプリカ

リカバリに備えるには、別のアベイラビリティーゾーンまたはリージョンからのフェイルオーバートラフィックを吸収できる十分な予備の容量をレプリカに常に保持している必要があります。ARC はアプリケーションを継続的に (1 分ごとに) 検査して、プロビジョニングされた容量がすべてのアベイラビリティーゾーンまたはリージョンにわたって一致していることを確認します。

ARC が検査する容量には、例えば、Amazon EC2 インスタンス数、Aurora の読み取りおよび書き込みキャパシティユニット、Amazon EBS ボリュームサイズなどが含まれます。リソース値に合わせてプライマリレプリカの容量をスケールアップしたものの、スタンバイレプリカの対応する値も増やすことを忘れた場合、ARC が不一致を検出するため、スタンバイの値を増やすことができます。

> **Important**
>
> 準備状況チェックは、アプリケーションのレプリカの設定とランタイムの状態が一致していることを継続的に確認するときに、最も役立つサービスです。準備状況チェックは、本番のレプリカが正常かどうかを示すために使用すべきではありません。また、準備状況チェックを、災害発生時のフェイルオーバーの主要なトリガーとして使用すべきでもありません。

アクティブスタンバイ構成において、セルからまたはセルにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。ARC の準備状況チェックは可用性が高くないため、システム停止中に準備状況チェックにアクセスできるとは限りません。さらに、チェックされたリソースは、災害時には利用できなくなる可能性もあります。

特定のセル (AWS リージョンまたはアベイラビリティーゾーン) またはアプリケーション全体のアプリケーションリソースの準備状況ステータスをモニタリングできます。EventBridge でルールを作成することで、準備状況チェックのステータスが変わったとき (`Not ready` に変わったときなど) に通知を受けることができます。詳細については、「Amazon EventBridge により ARC で準備状況チェックを使用する」を参照してください。準備状況ステータスは、 で表示することも AWS マネジメントコンソール、 などの API オペレーションを使用して表示することもできますget-

recovery-readiness。詳細については、「準備状況チェック API オペレーション」を参照してください。

準備状況チェックの仕組み

ARC は、準備状況チェックを使用してアプリケーションのレプリカを監査し、各レプリカが同じ構成設定で同じランタイムを持つことを確認します。

例えば、リカバリに備えるには、別のアベイラビリティーゾーンまたはリージョンからのフェイルオーバートラフィックを吸収できる十分な予備の容量を常に保持している必要があります。ARC はアプリケーションを継続的に (1 分ごとに) 検査して、プロビジョニングされた容量がすべてのアベイラビリティーゾーンまたはリージョンにわたって一致していることを確認します。ARC が検査する容量には、例えば、Amazon EC2 インスタンス数、Aurora の読み取りおよび書き込みキャパシティユニット、Amazon EBS ボリュームサイズなどが含まれます。リソース値に合わせてプライマリレプリカの容量をスケールアップしたものの、スタンバイレプリカの対応する値も増やすことを忘れた場合、ARC が不一致を検出するため、スタンバイの値を増やすことができます。

> **Important**
>
> 準備状況チェックは、アプリケーションのレプリカの設定とランタイムの状態が一致していることを継続的に確認するときに、最も役立つサービスです。準備状況チェックは、本番のレプリカが正常かどうかを示すために使用すべきではありません。また、準備状況チェックを、災害発生時のフェイルオーバーの主要なトリガーとして使用すべきでもありません。

アクティブスタンバイ構成において、セルからまたはセルにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。ARC の準備状況チェックは可用性が高くないため、システム停止中に準備状況チェックにアクセスできるとは限りません。さらに、チェックされたリソースは、災害時には利用できなくなる可能性もあります。

特定のセル (AWS リージョンまたはアベイラビリティーゾーン) またはアプリケーション全体のアプリケーションリソースの準備状況ステータスをモニタリングできます。EventBridge でルールを作成することで、準備状況チェックのステータスが変わったとき (Not ready に変わったときなど) に通知を受けることができます。詳細については、「Amazon EventBridge により ARC で準備状況チェックを使用する」を参照してください。準備状況ステータスは、 で表示することも AWS マネジメントコンソール、 などの API オペレーションを使用して表示することもできますget-

recovery-readiness。詳細については、「準備状況チェック API オペレーション」を参照してください。

準備状況ルールが準備状況ステータスを判断する仕組み

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

ARC の準備状況チェックでは、事前定義された各リソースタイプのルールとそれらのルールの定義方法に基づいて、準備状況ステータスが判断されます。ARC には、サポートされているリソースのタイプごとに、1 つのルールグループが含まれています。たとえば、ARC には Amazon Aurora クラスター、Auto Scaling グループなどの準備状況ルールのグループがあります。準備状況ルールには、セット内のリソースを相互に比較するものもあれば、リソースセット内の各リソースに関する特定の情報を調べるものもあります。

ユーザーは、準備状況ルールやルールのグループを、追加、編集、削除できません。ただし、Amazon CloudWatch アラームを作成したり、アラームの状態をモニタリングする準備状況チェックを作成したりはできます。例えば、Amazon EKS コンテナサービスをモニタリングするカスタムの CloudWatch アラームを作成したり、そのアラームの準備状況ステータスを監査する準備状況チェックを作成したりできます。

リソースセットを作成する AWS マネジメントコンソール ときに、各リソースタイプのすべての準備状況ルールを で表示することも、リソースセットの詳細ページに移動して、後で準備状況ルールを表示することもできます。準備状況ルールは「ARC での準備状況ルール」セクションでも確認できます。

準備状況チェックで一連のルールを使って一連のリソースを監査する場合、各ルールの定義方法によって、すべてのリソースで結果を READY または NOT READY にするのか、それともリソースごとに結果を変えるのかが決まります。さらに、準備状況ステータスは複数の方法で表示できます。たとえば、リソースセット内のリソースグループの準備状況ステータスを表示したり、リカバリグループまたはセル (つまり、リカバリグループの設定方法に応じて AWS リージョンまたはアベイラビリティーゾーン) の準備状況ステータスの概要を表示したりできます。

各ルールの説明の文言には、そのルールが適用されたときにどのようにリソースを評価し、準備状況ステータスを判断するのかが説明されています。ルールは、各リソースを検査するか、リソースセット内のすべてのリソースを検査して準備状況を判断するように定義されています。具体的には、ルールは以下のように機能します。

- ルールは、リソースセット内の各リソースを検査して条件を確認します。
  - すべてのリソースで条件が確認されると、すべてのリソースは READY に設定されます。
  - 1 つのリソースで条件の確認に失敗すると、そのリソースは NOT READY に設定され、それ以外のセルは READY のままとなります。

  例: MskClusterState: は各 Amazon MSK クラスターを検査し、ACTIVE の状態になっていることを確認します。
- このルールは、リソースセット内のすべてのリソースを検査して条件を確認します。
  - 条件が確認されると、すべてのリソースは READY に設定されます。
  - 条件を満たさないリソースがある場合、すべてのリソースは NOT READY に設定されます。

  例:VpcSubnetCount: はすべての VPC サブネットを検査し、それらのサブネット数が同じであることを確認します。
- 重要度の低いルール: このルールは、リソースセット内のすべてのリソースを検査して条件を確認します。
  - いずれかのリソースが条件を満たさなかったとしても、準備状況は変わりません。このような動作をするルールには、説明に注記が付きます。

  例: ElbV2CheckAzCount: は各 Network Load Balancer を検査し、アタッチされているアベイラビリティーゾーンが 1 つのみであることを確認します。注: このルールは準備状況ステータスには影響しません。

また、ARC では、クォータに関して追加の対策を講じています。準備状況チェックで、サポートされているリソースの Service Quotas (リソースの作成とオペレーションの最大値) のセル間に不一致が見つかった場合、ARC は、クォータが低い方のリソースで、自動的にクォータを引き上げます。これは、クォータ (制限) に対してのみ適用されます。キャパシティに関しては、アプリケーションのニーズに応じて、ユーザーが必要なキャパシティを追加する必要があります。

また、Amazon EventBridge で準備状況チェック用の通知を設定し、チェックのステータスが NOT READY に変わったときに通知を受け取ることもできます。設定の不一致が検出されると EventBridge から通知が送信されるので、修正措置を講じて、アプリケーションのレプリカを一致させ、リカバリ

に向けて準備を整えることができます。詳細については、「Amazon EventBridge により ARC で準備状況チェックを使用する」を参照してください。

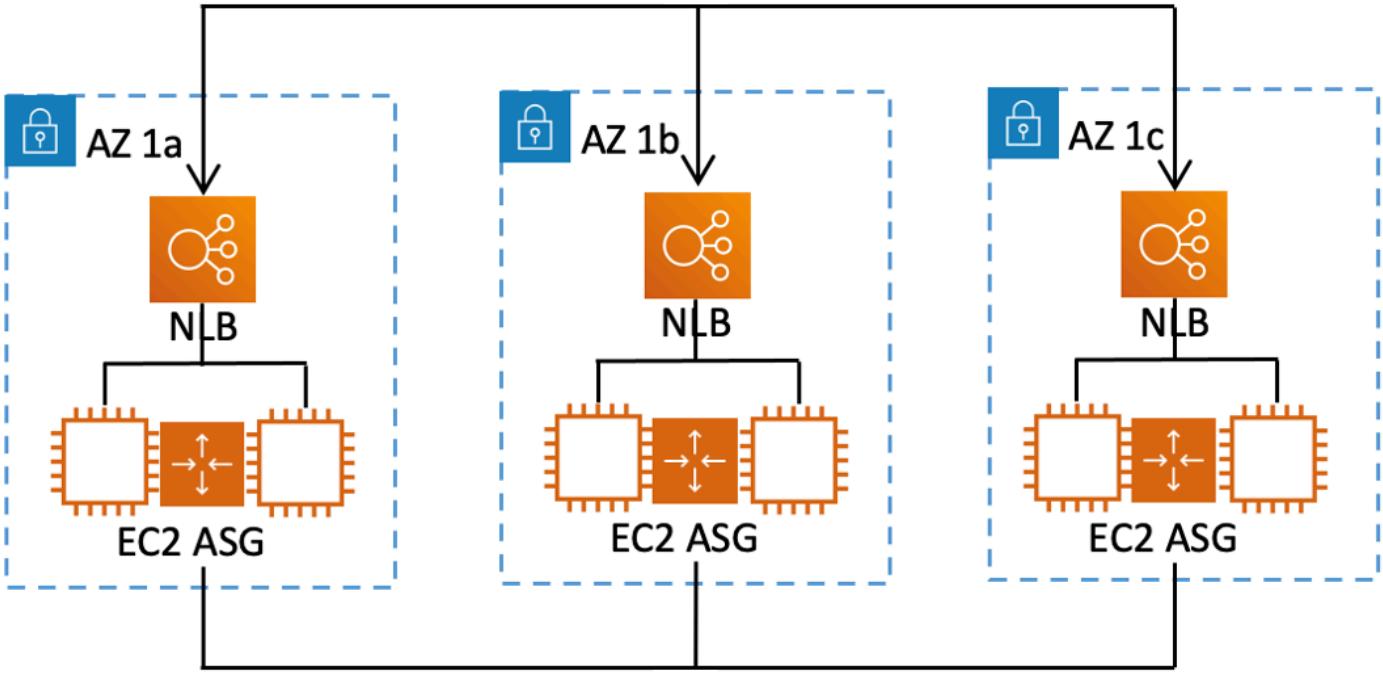
準備状況チェック、リソースセット、準備状況の範囲が連携する方法

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

準備状況チェックは、リソースセット内のリソースのグループを常時監査します。リソースセット (別個に、または準備状況チェックの作成中に) を作成して、ARC リカバリグループのセル (アベイラビリティーゾーンまたは AWS リージョン) にあるリソースをグループ化し、準備状況チェックを定義できるようにします。リソースセットは、通常、同じ種類のリソース (Network Load Balancer など) から成るグループですが、アーキテクチャの準備状況をチェックする場合は DNS ターゲットリソースになる場合もあります。

通常、アプリケーション内のリソースの各タイプに、1 つのリソースセットと準備状況チェックを作成します。アーキテクチャの準備状況チェックでは、最上位の DNS ターゲットリソースとそれに対応するグローバルな (リカバリグループレベルの) リソースセットを作成し、続いて、別のリソースセット用にセルレベルの DNS ターゲットリソースを作成します。

次の図は、3 つのセル (アベイラビリティーゾーン) を持つリカバリグループの例です。各セルに Network Load Balancer (NLB) と Auto Scaling グループ (ASG) があります。



このシナリオでは、3 つの Network Load Balancer 用のリソースセットと準備状況チェック、3 つの Auto Scaling グループ用のリソースセットと準備状況チェックを作成します。これで、リカバリグループの各リソースセットで、リソースタイプごとに準備状況チェックを行えます。

リソースの準備状況の範囲を作成することで、セルまたはリカバリグループの準備状況チェックの概要を追加できます。リソースの準備状況の範囲を指定するには、セルまたはリカバリグループの ARN を、リソースセット内の各リソースに関連付けます。これは、リソースセットの準備状況チェックを作成する際に実行できます。

例えば、このリカバリグループにおける Network Load Balancer のリソースセットの準備状況チェックを追加すると、各 NLB に準備状況の範囲を同時に追加できます。この場合は、AZ 1a の ARN を AZ 1a の NLB に、AZ 1b の ARN を NLB AZ 1b に、AZ 1c の ARN を AZ 1c の NLB にそれぞれ関連付けます。Auto Scaling グループの準備状況チェックを作成するときも同じことを行い、Auto Scaling グループのリソースセットの準備状況チェックを作成するときに、準備状況の範囲をそれぞれに割り当てます。

準備状況チェックを作成するときに準備状況の範囲を関連付けるのは任意ですが、こちらを設定しておくことを強く推奨します。準備状況の範囲を設定しておくと、ARC に、リカバリグループの準備状況チェックの概要と、セルレベルにおける準備状況チェックの概要の正確なステータスを READY または NOT READY で表示できます。準備状況の範囲を設定しないと、これらの概要を ARC に表示できません。

アプリケーションレベルのリソースや、DNS ルーティングポリシーなどのグローバルなリソースを追加する場合、準備状況の範囲のリカバリグループやセルは選択しません。代わりに、グローバルリソース (セルなし) を選択します。

DNS ターゲットリソースの準備状況チェック: レジリエンシーの準備状況の監査

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

ARC の、DNS ターゲットリソースの準備状況チェックを使うと、アプリケーションのアーキテクチャとレジリエンシーの準備状況を監査できます。このタイプの準備状況チェックでは、アプリケーションのアーキテクチャと Amazon Route 53 のルーティングポリシーを継続的にスキャンして、クロスゾーンおよびクロスリージョンの依存関係を監査します。

リカバリ指向アプリケーションには、複数のレプリカがあり、アベイラビリティーゾーンまたは AWS リージョンにサイロ化されているため、レプリカは互いに独立して失敗する可能性があります。正しくサイロ化するようにアプリケーションを調整する必要がある場合に、ARC は、必要に応じてアーキテクチャを更新できる変更を提案します。これにより、レジリエンスとフェイルオーバーへの備えを確保できます。

ARC は、アプリケーション内のセル (レプリカまたは障害抑制ユニットを表す) の数と範囲、およびセルがアベイラビリティーゾーンごとまたはリージョンごとにサイロ化されているかどうかを、自動的に検出します。次に、ARC は、セル内のアプリケーションリソースを識別してユーザーに情報を提供し、それらがアベイラビリティーゾーンまたはリージョンに正しくサイロ化されているかどうかを判断します。例えば、特定のアベイラビリティーゾーンを対象とするセルがある場合、準備状況チェックでは、ロードバランサーとその背後にあるターゲットも、それらのゾーンにサイロ化されているかどうかをモニタリングできます。

この情報を使用することで、セル内のリソースを正しいゾーンまたはリージョンに一致させるために、変更すべきことがあるかどうかを判断できます。

開始するには、アプリケーション用の DNS ターゲットリソースと、それらのリソースセットおよび準備状況チェックを作成します。詳細については、「ARC でアーキテクチャの推奨事項を取得する」を参照してください。

準備状況チェックとディザスタリカバリのシナリオ

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

ARC の準備状況チェックでは、フェイルオーバートラフィックに対処するためアプリケーションがスケールしていることを、ユーザーが確認できるようにして、アプリケーションとリソースのリカバリ準備が整っているかどうかに関するインサイトを提供しています。準備状況チェックのステータスは、本番のレプリカが正常であることを示す合図として使用すべきではありません。ただし、アプリケーションやインフラストラクチャのモニタリングや、レプリカから、またはレプリカにフェイルオーバーすべきか否かを判断するヘルスチェックシステムの補完に使用することは可能です。

緊急時や停電時には、ヘルスチェックとその他の情報を組み合わせて、スタンバイがスケールアップされ、正常で、本番トラフィックをフェイルオーバーする準備が整っているかどうかを判断します。例えば、スタンバイの準備状況チェックのステータスが READY であることを確認することに加え、スタンバイのセルに対して実行する canary が、成功基準を満たしているかどうかを確認します。

ARC の準備状況チェックは、単一の AWS リージョン、つまり米国西部 (オレゴン) でホストされているため、停電時や災害時に、準備状況チェックの情報が古くなったりチェックを利用できなくなったりする場合がありますのでご注意ください。詳細については、「ルーティングコントロールのデータプレーンとコントロールプレーン」を参照してください。

AWS 準備状況チェックのリージョンの可用性

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

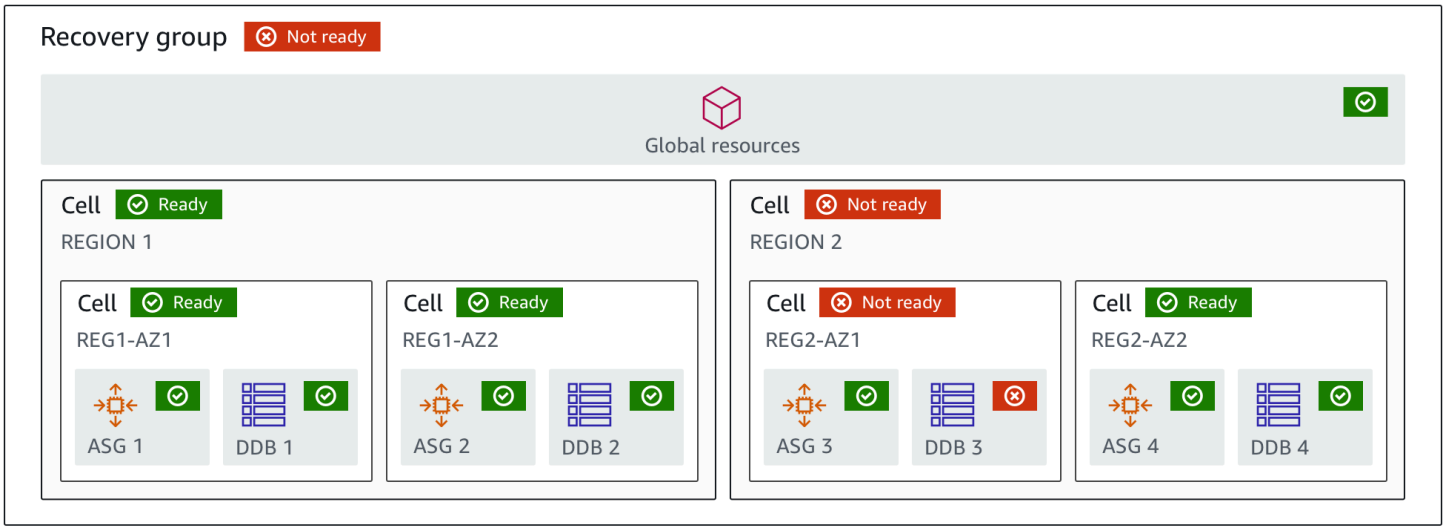
Amazon Application Recovery Controller (ARC) のリージョンサポートとサービスエンドポイントに関する詳細は、「Amazon Web Services 全般のリファレンス」にある「Amazon Application Recovery Controller (ARC) エンドポイントとクォータ」を参照してください。

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェックは、グローバル機能です。ただし、準備状況チェックリソースは米国西部 (オレゴン) リージョンにあるため、リソースセットや準備状況チェックなどのリソースを作成する場合など、リージョン ARC AWS CLI コマンドで米国西部 (オレゴン) リージョンを指定 (パラメータ を指定`--region us-west-2`) する必要があります。

準備状況チェックのコンポーネント

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

次の図は、準備状況チェック機能をサポートするように設定されたリカバリグループのサンプルを示しています。この例のリソースは、リカバリグループ内でセル (AWS リージョン別) とネストされたセル (アベイラビリティーゾーン別) にグループ化されています。リカバリグループ (アプリケーション) の全体的な準備状況ステータスに加え、セル (リージョン) とネストされたセル (アベイラビリティーゾーン) のそれぞれに個別の準備状況ステータスがあります。



以下は、ARC における準備状況チェック機能のコンポーネントです。

セル

セルはアプリケーションのレプリカ、または独立したフェイルオーバーのユニットを定義します。アプリケーションがレプリカ内で独立して実行するために必要なすべての AWS リソースをグループ化します。例えば、プライマリセルに 1 つのリソースセットがあり、スタンバイセルに別のリソースセットがあります。セルに含まれるものの境界はユーザーが決定しますが、セルは通常、アベイラビリティーゾーンやリージョンを表します。リージョン内の AZ のように、1 つのセル内に複数のセル (ネストされたセル) を持てます。ネストされた各セルは、独立したフェイルオーバーの単位を表します。

リカバリグループ

セルはリカバリグループに収集されます。リカバリグループは、フェイルオーバーの準備状況を確認したいアプリケーションまたはアプリケーションのグループを表します。機能的に互いに一致する 2 つ以上のセル、もしくはレプリカで構成されます。例えば、us-east-1a と us-east-1b 間で複製されたウェブアプリケーションがあり、us-east-1b がフェイルオーバー環境である場合、ARC ではこのアプリケーションを 2 つのセル (us-east-1a に 1 つ、us-east-1b に 1 つ) で構成されるリカバリグループとして表せます。リカバリグループには、Route 53 ヘルスチェックなどのグローバルリソースを含めることもできます。

リソースとリソース識別子

ARC で準備状況チェックのコンポーネントを作成するときは、リソース識別子を使用して Amazon DynamoDB テーブル、Network Load Balancer、DNS ターゲットリソースなどのリソースを指定します。リソース識別子は、リソースの Amazon リソースネーム (ARN)、または DNS

ターゲットリソースの場合は ARC がリソースを作成したときに生成した識別子のいずれかとなります。

DNS ターゲットリソース

DNS ターゲットリソースは、アプリケーションのドメイン名と、ドメインが指す AWS リソースなどの他の DNS 情報の組み合わせです。 AWS リソースを含めるのは任意ですが、含める場合は Route 53 リソースレコード、または Network Load Balancer でなければなりません。 AWS リソースを指定すると、アプリケーションの回復力を向上させるのに役立つ、より詳細なアーキテクチャの推奨事項を取得できます。ARC で DNS ターゲットリソースのリソースセットを作成し、そのリソースセットの準備状況チェックを作成して、アプリケーションに関するアーキテクチャの推奨事項を確認できます。準備状況チェックでは、DNS ターゲットリソースの準備状況ルールに基づいて、アプリケーションの DNS ルーティングポリシーも監視されます。

リソースセット

リソースセットは、複数のセルにまたがるリソース AWS または DNS ターゲットリソースを含む一連のリソースです。例えば、us-east-1a に 1 つのロードバランサーがあり、us-east-1b には別のロードバランサーがあります。ロードバランサーのリカバリの準備状況を監視するには、両方のロードバランサーを含むリソースセットを作成し、そのリソースセットの準備状況チェックを作成します。ARC は、セット内のリソースの準備状況を継続的にチェックします。また、準備状況の範囲を追加して、リソースセット内のリソースを、アプリケーション用に作成したリカバリグループに関連付けることもできます。

準備状況ルール

準備状況ルールは、ARC がリソースセット内の一連のリソースに対して実施する監査です。ARC には、準備状況チェックをサポートするリソースの種類ごとに準備状況ルールのセットがあります。各ルールには、ARC がリソースを検査する目的を示す ID と説明が含まれています。

準備状況チェック

準備状況チェックは、ARC がリカバリの準備状況を監査している Amazon Aurora インスタンスのセットなど、アプリケーション内のリソースセットをモニタリングします。準備状況チェックには、キャパシティ設定、 AWS クォータ、ルーティングポリシーなどの監査が含まれる場合があります。例えば、2 つのアベイラビリティーゾーンにまたがる Amazon EC2 Auto Scaling グループの準備状況を監査する場合、Auto Scaling グループごとに 1 つずつ、合計 2 つのリソース ARN を持つリソースセットの準備状況チェックを作成できます。次に、各グループが均等にスケールされるように、ARC はその 2 グループのインスタンスタイプとインスタンス数を継続的に監視します。

準備状況の範囲

準備状況の範囲は、特定の準備状況チェックの対象となるリソースのグループを示します。準備状況チェックの範囲は、リカバリグループ (つまり、アプリケーション全体を対象とするグローバル) にすることも、セル (つまり、リージョンまたはアベイラビリティーゾーン) にすることもできます。ARC のグローバルリソースの場合、準備状況の範囲をリカバリグループまたはグローバルリソースレベルに設定してください。例えば、Route 53 ヘルスチェックはリージョンやアベイラビリティーゾーンに固有のものではないため、ARC のグローバルリソースとなります。

準備状況チェック用のデータプレーンとコントロールプレーン

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは、可用性が高く、災害シナリオで必要なときに使用できるようにすることをお勧めします。信頼性と耐障害性を最大限に高めるため、可能であれば通常、データプレーン機能をメカニズムに使用する必要があります。そのことを念頭に置いて、サービス機能がコントロールプレーンとデータプレーンにどのように分けられているのか、また、サービスのデータプレーンで非常に高い信頼性が期待できるのはどのような場合なのかを理解することが重要です。

ほとんどの AWS サービスと同様に、準備状況チェック機能の機能は、コントロールプレーンとデータプレーンでサポートされています。どちらのタイプも信頼できるように構築されていますが、コントロールプレーンはデータ整合性のために最適化され、データプレーンは可用性のために最適化されています。データプレーンは、コントロールプレーンが使用できなくなるような破壊的なイベントでも、可用性を維持できるように設計されています。

一般に、コントロールプレーンを使用すると、サービス内のリソースの作成、更新、削除などの基本的な管理機能を実行できます。データプレーンはサービスのコア機能を提供します。

準備状況チェックには、コントロールプレーンとデータプレーンの両方に対応するリカバリの準備状況 API という単一の API があります。準備状況チェックと準備状況リソースは、米国西部 (オレゴ

ン) リージョン (us-west-2) にのみあります。準備状況チェックのコントロールプレーンとデータプレーンは、信頼性は高いですが、可用性は高くありません。

データプレーン、コントロールプレーン、および が高可用性目標を達成するためのサービス AWS を構築する方法の詳細については、Amazon Builders' Library の「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。

Amazon Application Recovery Controller (ARC) での準備状況チェックのタグ付け

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

タグは、 AWS リソースを識別して整理するために使用する単語またはフレーズ (メタデータ) です。各リソースに複数のタグを追加できます。各タグには、定義したキーと値が含まれます。例えば、キーを環境、値を本番とできます。追加したタグに基づいて、リソースを検索したりフィルタ処理したりできます。

ARC の準備状況チェックでは、次のリソースにタグを付けることができます。

- リソースセット
- 準備状況チェック

ARC でのタグ付けは、API を使用してのみ可能です。例えば、 AWS CLIを使用します。

以下は、 AWS CLIを使用した準備状況チェックでのタグ付けの例です。

```
aws route53-recovery-readiness --region us-west-2 create-resource-
set --resource-set-name dynamodb_resource_set --resource-set-type
AWS::DynamoDB::Table --resources ReadinessScopes=arn:aws:aws-recovery-
readiness::111122223333:cell/PDXCell,ResourceArn=arn:aws:dynamodb:us-
west-2:111122223333:table/PDX_Table ReadinessScopes=arn:aws:aws-recovery-
```

```
readiness::111122223333:cell/IADCell,ResourceArn=arn:aws:dynamodb:us-
east-1:111122223333:table/IAD_Table --tags Stage=Prod

aws route53-recovery-readiness --region us-west-2 create-readiness-
check --readiness-check-name dynamodb_readiness_check --resource-set-name
dynamodb_resource_set --tags Stage=Prod
```

詳細については、Amazon Application Recovery Controller (ARC) の「Recovery Readiness API リファレンスガイド」の「TagResource」を参照してください。

ARC の準備状況チェックの料金

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

設定した準備状況チェックごとに 1 時間あたりの料金を支払います。

ARC の料金および料金例の詳細については、「ARC の料金」を参照してください。

アプリケーションに回復力のある復旧プロセスを設定する

複数の AWS リージョンにある AWS アプリケーションで Amazon Application Recovery Controller (ARC) を使用するには、復旧準備を効果的にサポートできるように、アプリケーションの耐障害性を設定するためのガイドラインに従う必要があります。次に、アプリケーションの準備状況チェックを作成し、フェイルオーバーに備えてトラフィックを再ルーティングするルーティングコントロールを設定します。また、レジリエンシーを高めるアプリケーションのアーキテクチャについて、ARC が提供する推奨事項を確認することもできます。

> **Note**
>
> アベイラビリティーゾーンによってサイロ化されているアプリケーションがある場合は、フェイルオーバーリカバリにゾーンシフトまたはゾーンオートシフトを使用することを検討してください。アベイラビリティーゾーンの障害からアプリケーションを確実に回復するために、ゾーンシフトまたはゾーンオートシフトを使用する設定は必要ありません。

ロードバランサーリソースのアベイラビリティーゾーンからトラフィックを移動するには、ARC コンソールまたは Elastic Load Balancing コンソールでゾーンシフトを開始します。または、ゾーンシフト API アクションで AWS Command Line Interface または AWS SDK を使用できます。詳細については、「ARC のゾーンシフト」を参照してください。

回復力のあるフェイルオーバー設定の開始方法の詳細については、「Amazon Application Recovery Controller (ARC) でマルチリージョンのリカバリを開始する方法」を参照してください。

ARC の準備状況チェックのベストプラクティス

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) の準備状況チェックには、次のベストプラクティスをお勧めします。

準備状況ステータスの変更に関する通知を追加する

Amazon EventBridge で、準備状況チェックのステータスが変化したとき (READY から NOT READY へ、など) に通知を送信するためのルールを設定します。通知が届くと、問題を調査して対処し、アプリケーションとリソースが予定したとおりにフェイルオーバーできる状態になっていることを確認できます。

EventBridge ルールを設定して、リカバリグループ (アプリケーション用) 、セル (AWS リージョンなど) 、リソースセットの準備状況チェックなど、いくつかの準備状況チェックのステータス変更の通知を送信できます。

詳細については、「Amazon EventBridge により ARC で準備状況チェックを使用する」を参照してください。

準備状況チェック API オペレーション

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

次の表は、リカバリ準備状況 (準備状況チェック) に使用できる ARC オペレーションを、関連するドキュメントへのリンクと共に一覧にしたものです。

AWS Command Line Interfaceで一般的なリカバリ準備状況 API オペレーションを使用する方法の例については、「で ARC 準備状況チェック API オペレーションを使用する例 AWS CLI」を参照してください。

アクション	ARC コンソールの使用	ARC API の使用
セルを作成する	「ARC でのリカバリグループの作成、更新、削除」を参照	「CreateCell」を参照
セルを取得する	「ARC でのリカバリグループの作成、更新、削除」を参照	「GetCell」を参照
セルを削除する	「ARC でのリカバリグループの作成、更新、削除」を参照	「DeleteCell」を参照
セルを更新する	該当なし	「UpdateCell」を参照
アカウントのセルを一覧表示する	「ARC でのリカバリグループの作成、更新、削除」を参照	「ListCells」を参照
リカバリグループを作成する	「ARC でのリカバリグループの作成、更新、削除」を参照	CreateRecoveryGroup を参照
リカバリグループを取得する	「ARC でのリカバリグループの作成、更新、削除」を参照	「GetRecoveryGroup」を参照

アクション	ARC コンソールの使用	ARC API の使用
リカバリグループを更新する	「ARC でのリカバリグループの作成、更新、削除」を参照	「UpdateRecoveryGroup」を参照
リカバリグループを削除する	「ARC でのリカバリグループの作成、更新、削除」を参照	「DeleteRecoveryGroup」を参照
リカバリグループを一覧表示する	「ARC でのリカバリグループの作成、更新、削除」を参照	「ListRecoveryGroups」を参照
リソースセットを作成する	「ARC の準備状況チェックの作成と更新」を参照	「CreateResourceSet」を参照
リソースセットを取得する	「ARC の準備状況チェックの作成と更新」を参照	「GetResourceSet」を参照
リソースセットを更新する	「ARC の準備状況チェックの作成と更新」を参照	「UpdateResourceSet」を参照
リソースセットを削除する	「ARC の準備状況チェックの作成と更新」を参照	「DeleteResourceSet」を参照
リソースセットを一覧表示する	「ARC の準備状況チェックの作成と更新」を参照	「ListResourceSets」を参照
準備状況チェックを作成する	「ARC の準備状況チェックの作成と更新」を参照	「CreateReadinessCheck」を参照
準備状況チェックを取得する	「ARC の準備状況チェックの作成と更新」を参照	「GetReadinessCheck」を参照
準備状況チェックを更新する	「ARC の準備状況チェックの作成と更新」を参照	「UpdateReadinessCheck」を参照
準備状況チェックを削除する	「ARC の準備状況チェックの作成と更新」を参照	「DeleteReadinessCheck」を参照
準備状況チェックを一覧表示する	「ARC の準備状況チェックの作成と更新」を参照	「ListReadinessChecks」を参照

アクション	ARC コンソールの使用	ARC API の使用
準備状況ルールを一覧表示する	「ARC での準備状況ルールの説明」を参照	「ListRules」を参照
準備状況チェック全体の状態をチェックする	「ARC で準備状況ステータスをモニタリングする」を参照	「GetReadinessCheckStatus」を参照
リソースの状態をチェックする	「ARC で準備状況ステータスをモニタリングする」を参照	「GetReadinessCheckResourceStatus」を参照
セルの状態をチェックする	「ARC で準備状況ステータスをモニタリングする」を参照	「GetCellReadinessSummary」を参照
リカバリグループの状態をチェックする	「ARC で準備状況ステータスをモニタリングする」を参照	「GetRecoveryGroupReadinessSummary」を参照

で ARC 準備状況チェック API オペレーションを使用する例 AWS CLI

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

このセクションでは、 を使用して API オペレーションを使用して Amazon Application Recovery Controller (ARC) の準備状況チェック機能 AWS Command Line Interface を操作する簡単なアプリケーション例について説明します。これらの例は、CLI を使用した準備状況チェック機能の操作方法の基本的な理解を深めることを目的としています。

ARC 監査でアプリケーションレプリカ内のリソースの不一致を確認する準備状況チェック。アプリケーションの準備状況チェックを設定するには、アプリケーション用に作成したレプリカと一致するアプリケーションリソースを ARC セルにセットアップするか、モデル化する必要があります。次に、スタンバイアプリケーションレプリカとそのリソースが本番稼働用レプリカと継続的に一致するように、これらのレプリカを監査する準備状況チェックを設定します。

簡単な例として、米国東部 (バージニア北部) リージョン (us-east-1) で実行中の Simple-Service という名前のアプリケーションを見てみましょう。米国西部 (オレゴン) リージョン (us-west-2) にもアプリケーションのスタンバイコピーがあります。この例では、準備状況チェックを設定して、これら 2 つのバージョンのアプリケーションを比較します。これにより、フェイルオーバーのシナリオで必要な場合に、スタンバイの米国西部 (オレゴン) リージョンがトラフィックを受信できる状態になっていることを確認できます。

の使用の詳細については AWS CLI、AWS CLI 「 コマンドリファレンス」を参照してください。準備状況 API アクションのリストと詳細情報へのリンクについては、「準備状況チェック API オペレーション」を参照してください。

ARC のセルは障害の境界 (アベイラビリティーゾーンやリージョンなど) を表し、リカバリグループにまとめられます。リカバリグループとは、フェイルオーバーの準備状況を確認したいアプリケーションのことです。準備状況チェックのコンポーネントの詳細については、「準備状況チェックのコンポーネント」を参照してください。

> **Note**
>
> ARC は複数の のエンドポイントをサポートするグローバルサービス AWS リージョン ですが、ほとんどの ARC CLI コマンドで米国西部 (オレゴン) リージョンを指定 (つまり、パラメータ を指定`--region us-west-2`) する必要があります。例えば、リカバリグループまたは準備状況チェックなどのリソースを作成する場合です。

このアプリケーションの例では、まず、リソースがあるリージョンごとに 1 つのセルを作成します。次に、リカバリグループを作成し、準備状況チェックの設定を完了します。

1. セルを作成する

1a. us-east-1 セルを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name east-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
    "CellName": "east-cell",
    "Cells": [],
```

```
    "ParentReadinessScopes": [],
    "Tags": {}
}
```

1b. us-west-1 セルを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name west-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell",
    "CellName": "west-cell",
    "Cells": [],
    "ParentReadinessScopes": [],
    "Tags": {}
}
```

1c. これで 2 つのセルができました。list-cells API を呼び出して、それらが存在することを確認できます。

```
aws route53-recovery-readiness --region us-west-2 list-cells
```

```
{
    "Cells": [
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell",
            "CellName": "east-cell",
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        },
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell",
            "CellName": "west-cell"
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        }
    ]
```

```
}
```

2. リカバリグループを作成する

リカバリグループは、ARC のリカバリの準備状況における最上位リソースです。リカバリグループはアプリケーション全体を表します。このステップでは、アプリケーション全体をモデル化するリカバリグループを作成し、作成した 2 つのセルを追加します。

2a. リカバリグループを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-recovery-group \
    --recovery-group-name simple-service-recovery-group \
       --cells "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"\
       "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
```

```
{
    "Cells": [],
    "RecoveryGroupArn": "arn:aws:route53-recovery-readiness::111122223333:recovery-
group/simple-service-recovery-group",
    "RecoveryGroupName": "simple-service-recovery-group",
    "Tags": {}
}
```

2b. (オプション) `list-recovery-groups` API を呼び出して、リカバリグループが正しく作成されたことを確認できます。

```
aws route53-recovery-readiness --region us-west-2 list-recovery-groups
```

```
{
    "RecoveryGroups": [
        {
            "Cells": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "RecoveryGroupArn": "arn:aws:route53-recovery-
readiness::111122223333:recovery-group/simple-service-recovery-group",
            "RecoveryGroupName": "simple-service-recovery-group",
            "Tags": {}
        }
    ]
```

```
}
```

アプリケーションのモデルができたので、モニタリングするリソースを追加しましょう。ARC では、モニタリングするリソースのグループをリソースセットと呼びます。リソースセットには、すべて同じタイプのリソースが含まれています。リソースセット内のリソースを相互に比較して、セルのフェイルオーバー準備状況を判断します。

3. リソースセットを作成する

Simple-Service アプリケーションが本当にシンプルで、DynamoDB テーブルのみを使用していると仮定しましょう。us-east-1 に DynamoDB テーブルがあり、us-west-2 に別のテーブルがあります。リソースセットには、各リソースが含まれるセルを識別する準備状況の範囲も含まれています。

3a. Simple-Service アプリケーションのリソースを反映したリソースセットを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-resource-set \
    --resource-set-name ImportantInformationTables \
    --resource-set-type AWS::DynamoDB::Table \
    --resources
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
west-cell"
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
east-cell"
```

```
{
    "ResourceSetArn": "arn:aws:route53-recovery-readiness::111122223333:resource-set/
sample-resource-set",
    "ResourceSetName": "ImportantInformationTables",
    "Resources": [
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"
```

```
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
        }
    ],
    "Tags": {}
}
```

3b. (オプション) `list-resource-sets` API を呼び出すと、リソースセットに何が含まれているかを確認できます。これにより、 AWS アカウントのすべてのリソースセットが一覧表示されます。先ほど作成したリソースセットは 1 つだけであることがわかります。

```
aws route53-recovery-readiness --region us-west-2 list-resource-sets
```

```
{
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}{
```

```
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-
readiness::&ExampleAWSAccountNo1;:cell/east-cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}
```

これで、ARC で Simple-Service アプリケーションをモデル化するためのセル、リカバリグループ、リソースセットが作成されました。次に、準備状況チェックを設定して、リソースのフェイルオーバー準備状況をモニタリングします。

4. 準備状況チェックを作成する

準備状況チェックは、チェックにアタッチされているリソースセット内の各リソースに一連のルールを適用します。ルールはリソースタイプごとに異なります。つまり、`AWS::DynamoDB::Table` や `AWS::EC2::Instance` などには異なるルールがあるということです。ルールは、構成、容量 (利用可能かつ適用可能な場合)、制限 (利用可能で適用可能な場合)、ルーティング構成など、リソースのさまざまな側面をチェックします。

> **Note**
>
> 準備状況チェックでリソースに適用されるルールを確認するには、ステップ 5 に説明があるとおり get-readiness-check-resource-status API を使用できます。ARC の準備状況ルールすべてのリストを表示するには、list-rules を使用するか、または「ARC での準備状況ルールの説明」を参照してください。ARC には、リソースタイプごとに実行される特定のルールセットがあり、現時点ではカスタマイズできません。

4a. ImportantInformationTables というリソースセットの準備状況チェックを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-readiness-check \
    --readiness-check-name ImportantInformationTableCheck --resource-set-name
 ImportantInformationTables
```

```
{
    "ReadinessCheckArn": "arn:aws:route53-recovery-readiness::111122223333:readiness-
check/ImportantInformationTableCheck",
    "ReadinessCheckName": "ImportantInformationTableCheck",
    "ResourceSet": "ImportantInformationTables",
    "Tags": {}
}
```

4b. (オプション) 準備状況チェックが正常に作成されたことを確認するには、list-readiness-checks API を実行します。この API は、アカウントのすべての準備状況チェックを表示します。

```
aws route53-recovery-readiness --region us-west-2 list-readiness-checks
```

```
{
    "ReadinessChecks": [
        {
            "ReadinessCheckArn": "arn:aws:route53-recovery-
readiness::111122223333:readiness-check/ImportantInformationTableCheck",
            "ReadinessCheckName": "ImportantInformationTableCheck",
            "ResourceSet": "ImportantInformationTables",
            "Tags": {}
        }
    ]
}
```

5. 準備状況チェックをモニタリングする

アプリケーションをモデル化し、準備状況チェックを追加したので、リソースをモニタリングする準備が整いました。アプリケーションの準備状況は次の 4 つのレベルでモデル化できます。準備状況チェックレベル (リソースのグループ)、個別のリソースレベル、セルレベル (アベイラビリティーゾーンまたはリージョン内のすべてのリソース)、リカバリグループレベル (アプリケーション全体) です。これらの各タイプの準備状況ステータスを取得するためのコマンドを次に示します。

5a. 準備状況チェックのステータスを確認します。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-status\
    --readiness-check-name ImportantInformationTableCheck
```

```
{
    "Readiness": "READY",
    "Resources": [
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast2"
    ]
}
```

5b. チェックされた各ルールのステータスなど、準備状況チェックにおける単一のリソースの詳細な準備状況ステータスを確認します。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-resource-status \
    --readiness-check-name ImportantInformationTableCheck \
    --resource-identifier "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
```

```
{"Readiness": "READY",
    "Rules": [
        {
```

```
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoTableStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoPeakRcuWcu"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsPeakRcuWcu"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsConfig"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
```

```
            "Readiness": "READY",
            "RuleId": "DynamoReplicationLatency"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoAutoScalingConfiguration"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoLimits"
        }
    ]
}
```

5c. セルの全体的な準備状況を確認します。

```
aws route53-recovery-readiness --region us-west-2 get-cell-readiness-summary \
    --cell-name west-cell
```

```
{
    "Readiness": "READY",
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

5d. 最後に、リカバリグループレベルにおけるアプリケーションの最上位の準備状況を確認します。

```
aws route53-recovery-readiness --region us-west-2 get-recovery-group-readiness-summary
 \
    --recovery-group-name simple-service-recovery-group
```

```
{
    "Readiness": "READY",
```

```
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

リカバリグループと準備状況チェックの使用

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

このセクションでは、これらのリソースの作成、更新、削除を含め、リカバリグループと準備状況チェックの手順について説明します。

ARC でのリカバリグループの作成、更新、削除

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

リカバリグループは、Amazon Application Recovery Controller (ARC) にあるアプリケーションを表します。通常は、リソースと機能の点から互いにレプリカとなる 2 つ以上のセルで構成されているため、一方のセルからもう一方のセルにフェイルオーバーできます。各セルには、1 つの AWS リージョンまたはアベイラビリティーゾーンのアクティブなリソースの Amazon リソースネーム (ARNs) が含まれます。リソースは、Elastic Load Balancing ロードバランサー、Auto Scaling グループ、またはその他のリソースなどです。別のアベイラビリティーゾーンまたはリージョンを表す、対応するセルには、アクティブセルにある同じタイプのスタンバイリソース (ロードバランサー、Auto Scaling グループなど) が含まれています。

セルは、アプリケーションのレプリカを表します。ARC の準備状況チェックは、アプリケーションが、一方のレプリカから他方のレプリカにフェイルオーバーする準備ができているかどうかを判断するときに役立ちます。ただし、レプリカからまたはレプリカにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。

準備状況チェックでは、リソースを監査して、そのタイプのリソースに対する事前定義された一連のルールに基づいて準備状況を判断します。レプリカを含むリカバリグループを作成したら、アプリケーション内のリソースの ARC 準備状況チェックを追加します。これにより、ARC は、レプリカの設定と構成が経時的に同じであることを確認できます。

トピック

- リカバリグループの作成
- リカバリグループとセルの更新および削除

リカバリグループの作成

このセクションでは、ARC コンソールでリカバリグループを作成する手順について説明します。Amazon Application Recovery Controller (ARC) でリカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リカバリグループを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [リカバリの準備状況] ページで [作成] を選択し、続いて [リカバリグループ] を選択します。
4. リカバリグループの名前を入力し、[次へ] を選択します。
5. [セルを作成] を選択し、[セルを追加] を選択します。
6. セルの名前を入力します。アプリケーションレプリカが米国西部 (北カリフォルニア) にある場合、`MyApp-us-west-1` という名前のセルを追加できます。
7. [セルを追加] を選択し、2 番目のセルの名前を追加します。レプリカが米国東部 (オハイオ) にある場合は、 `MyApp-us-east-2` という名前のセルを追加できます。
8. ネストされたセル (レプリカが複数のリージョン内にある複数のアベイラビリティーゾーンにある) を追加する場合は、[アクション] を選択し、[ネストされたセルを追加] を選択してから、名前を入力します。

9. アプリケーションレプリカのすべてのセルおよびネストされたセルを追加したら、[次へ] をクリックします。
10. リカバリグループを確認し、[リカバリグループを作成] をクリックします。

リカバリグループとセルの更新および削除

このセクションでは、ARC コンソールでリカバリグループを更新および削除する手順、ならびにセルを削除する手順について説明します。Amazon Application Recovery Controller (ARC) でリカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リカバリグループを更新または削除し、セルを削除するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [リカバリの準備状況] ページでリカバリグループを選択します。
4. リカバリグループを操作するには、[アクション] を選択し、[リカバリグループを編集] または [リカバリグループを削除] を選択します。
5. リカバリグループを編集する際に、セルまたはネストされたセルを追加または削除できます。
   - セルを追加するには、[セルを追加] を選択します。
   - セルを削除するには、セルの横にある [アクション] ラベルで [セルを削除] を選択します。

ARC の準備状況チェックの作成と更新

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

このセクションでは、これらのリソースの作成、更新、削除を含め、準備状況チェックとリソースセットの手順について説明します。

準備状況チェックの作成と更新

このセクションでは、ARC コンソールで準備状況チェックを作成する手順について説明します。Amazon Application Recovery Controller (ARC) でリカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

準備状況チェックは、準備状況チェックのリソースセットを編集してリソースを追加または削除するか、リソースの準備状況の範囲を変更することで、更新できます。

準備状況チェックを作成するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [準備状況] ページで [作成] をクリックし、次に [準備状況チェック] を選択します。
4. 準備状況チェックの名前を入力し、チェックするリソースタイプを選択して [次へ] をクリックします。
5. 準備状況チェック用のリソースセットを追加します。リソースセットは、別のレプリカにある、同じタイプのリソースのグループです。次のいずれかを選択します。

   - 既に作成したリソースセット内のリソースを使用して準備状況チェックを作成します。
   - リソースセットを作成します。

   新しいリソースセットを作成することを選択した場合は、その名前を入力し、[追加] をクリックします。
6. そのリソースセットに含めるリソースごとに、Amazon リソースネーム (ARN) を 1 つずつコピーアンドペーストし、[次へ] をクリックします。

   > **Tip**
   >
   > ARC が各リソースタイプに期待する ARN 形式の例および詳細については、「ARC のリソースタイプと ARN フォーマット」を参照してください。
7. 必要に応じて、ARC が、この準備状況チェックに追加したリソースのタイプをチェックする際に使用する、準備状況ルールを確認します。次いで、[次へ] を選択します。
8. (オプション) [リカバリグループ名] で、準備状況チェックを関連付けるリカバリグループを選択し、リソース ARN ごとに、そのリソースが含まれているドロップダウンメニューからセル

(リージョンまたはアベイラビリティーゾーン) を選択します。リソースが、DNS ルーティングポリシーなどアプリケーションレベルのリソースである場合は、[グローバルリソース (セルなし)] を選択します。

これにより、準備状況チェックにおけるリソースの準備状況の範囲が指定されます。

> **Important**
>
> この手順はオプションですが、リカバリグループとセルの準備状況に関する情報の概要を手に入れるには、準備状況の範囲を追加する必要があります。この手順を飛ばし、ここで準備状況の範囲を選択して準備状況チェックをリカバリグループのリソースに関連付けなければ、ARC は、リカバリグループまたはセルの準備状況に関する情報の概要を返せません。

9. [次へ] を選択します。
10. 確認ページの情報を確認し、[準備状況チェックを作成] をクリックします。

準備状況チェックを削除するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. 準備状況チェックを選択し、[アクション] で [削除] をクリックします。

リソースセットの作成と編集

通常、リソースセットは準備状況チェックの作成の一環として作成しますが、個別に作成することも可能です。また、リソースセットを編集してリソースを追加または削除することもできます。このセクションでは、ARC コンソールでリソースセットを作成または編集する手順について説明します。Amazon Application Recovery Controller (ARC) でリカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リソースセットを作成するには

1. https://console.aws.amazon.com/route53/home で Route 53 コンソールを開きます。
2. [アプリケーションリカバリコントローラー] で [リソースセット] を選択します。
3. [作成] を選択します。

4. リソースセットの名前を入力し、このセットに含めるリソースのタイプを選択します。
5. [追加] をクリックし、セットに追加するリソースの Amazon リソースネーム (ARN) を入力します。
6. リソースを追加したら、[リソースセットを作成] を選択します。

リソースセットを編集するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [リソースセット] で [アクション] を選択して、[編集] を選択します。
4. 次のいずれかを行います。
   - リソースセットからリソースを削除するときは、[削除] をクリックします。
   - リソースセットにリソースを追加するには、[追加] をクリックし、リソースの Amazon リソースネーム (ARN) を入力します。
5. リソースの準備状況の範囲を編集してリソースを別のセルに関連付け、準備状況を確認することもできます。
6. [保存] を選択します。

ARC で準備状況ステータスをモニタリングする

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) では、アプリケーションの準備状況を以下の各レベルで確認できます。

- リソースセット内のリソースの準備状況チェックレベル
- 個々のリソースレベル

- アベイラビリティーゾーンまたは AWS リージョン内のすべてのリソースのセル (アプリケーションレプリカ) レベル
- アプリケーション全体のリカバリグループレベル

準備状況ステータスの変化について通知を受けたり、Route 53 コンソールから、または ARC CLI コマンドを使用して、準備状況ステータスの変化をモニタリングしたりできます。

準備状況ステータスの通知

Amazon EventBridge を使用して、ARC リソースをモニタリングして準備状況ステータスの変化を通知する、イベント駆動型のルールを設定できます。詳細については、「Amazon EventBridge により ARC で準備状況チェックを使用する」を参照してください。

ARC コンソールで準備状況ステータスをモニタリングする

次の手順では、 AWS マネジメントコンソールで復旧準備状況をモニタリングする方法について説明します。

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [準備状況] ページの [リカバリグループ] で、各リカバリグループ (アプリケーション) の [リカバリグループの準備状況のステータス] を表示します。

   特定のセルまたは個々のリソースの準備状況を表示することもできます。

CLI コマンドを使った準備状況ステータスのモニタリング

このセクションでは、アプリケーションとリソースの準備状況ステータスをさまざまなレベルで確認するために使用する AWS CLI コマンドの例を示します。

リソースセットの準備状況

リソースセット (リソースのグループ) 用に作成した準備状況チェックのステータスです。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-status --readiness-check-name ReadinessCheckName
```

1 つのリソースの準備状況

チェックされた各準備状況ルールのステータスを含め、準備状況チェックで 1 つのリソースのステータスを確認するには、準備状況チェック名とリソース ARN を指定します。例えば、次のようになります。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-
check-status --readiness-check-name ReadinessCheckName --resource-arn
"arn:aws:dynamodb:us-west-2:111122223333:table/TableName"
```

セルの準備状況

1 つのセル、つまりリージョンまたはアベイラビリティーゾーンのステータスです。

```
aws route53-recovery-readiness --region us-west-2 get-cell-readiness-
summary --cell-name CellName
```

アプリケーションの準備状況

リカバリグループレベルでのアプリケーション全体のステータスです。

```
aws route53-recovery-readiness --region us-west-2 get-recovery-group-
readiness-summary --recovery-group-name RecoveryGroupName
```

ARC でアーキテクチャの推奨事項を取得する

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

既存のアプリケーションがある場合、Amazon Application Recovery Controller (ARC) は、アプリケーションのアーキテクチャとルーティングポリシーを評価して、アプリケーションのレジリエンシーを高める設計に変更するための、推奨事項を提供します。アプリケーションを表すリカバリグループを ARC に作成したら、このセクションの手順に従って、アプリケーションのアーキテクチャに関する推奨事項を取得します。

リカバリグループの DNS ターゲットリソースをまだ指定していない場合は、指定することが推奨されます。そうすれば、より詳細な推奨事項を取得できます。追加情報を提供すると、ARC はユーザーにより適した推奨事項を提供できるようになります。例えば、Amazon Route 53 リソースレコードまたは Network Load Balancer をターゲットリソースとして入力すると、ARC は、リカバリグループに作成されたセルの数が最適かどうかについての情報を提供できます。

DNS ターゲットリソースについては、次の点に注意してください。

- ターゲットリソースには、Route 53 リソースレコードまたは Network Load Balancer のみを指定します。
- 各リカバリグループには DNS ターゲットリソースを 1 つだけ作成します。
- 推奨: 各セルには DNS ターゲットリソースを 1 つ作成します。
- DNS ターゲットリソースを、準備状況チェックを行う 1 つのリソースセットにグループ化します。

次の手順では、DNS ターゲットリソースの作成方法と、アプリケーションのアーキテクチャの推奨事項を取得する方法について説明します。

アーキテクチャの更新に関する推奨事項を取得するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [リカバリグループ名] で、アプリケーションを表すリカバリグループを選択します。
4. [リカバリグループの詳細] ページの [アクション] メニューで、[このリカバリグループのアーキテクチャに関する推奨事項を取得] を選択します。
5. DNS ターゲットリソースの準備状況チェックをまだ作成していない場合は作成します。そうすれば、ARC からアーキテクチャの推奨事項を取得できるようになります。[DNS ターゲットリソースの作成] を選択します。

   DNS ターゲットリソースの詳細については、「準備状況チェックのコンポーネント」を参照してください。
6. DNS ターゲットリソースのリソースセットを作成するには、準備状況チェックを作成します。準備状況チェックの名前を入力し、準備状況チェックのタイプとして [DNS ターゲットリソース] を選択します。
7. リソースセットの名前を入力します。

8. DNS 名、ホストゾーン ARN、レコードセット ID など、アプリケーションの属性を入力します。

   > **Tip**
   >
   > ホストゾーン ARN の形式を確認するには、「ARC のリソースタイプと ARN フォーマット」でホストゾーンの ARN 形式を参照します。

   こちらはオプションですが、[オプションの属性を追加] を選択して、Network Load Balancer ARN またはドメインの Route 53 リソースレコードを指定することが強く推奨されます。
9. (オプション) [リカバリグループの設定] で、DNS ターゲットリソースのセルを選択し、準備状況の範囲を設定します。
10. [Create resource set] (リソースセットの作成) を選択します。
11. [リカバリグループ] の詳細ページで、[アーキテクチャの推奨事項の取得] を選択します。ARC で、ページ上に一連の推奨事項が表示されます。

推奨事項のリストを確認します。その後、アプリケーションのレジリエンスを高めるための変更を加えるかどうか、また、どのように変更を加えるかを決定できます。

ARC でのクロスアカウント認証の作成

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

リソースを複数の AWS アカウントに分散させると、アプリケーションの状態を包括的に把握することが困難になる場合があります。また、迅速な意思決定に必要な情報を取得することも難しくなる可能性があります。Amazon Application Recovery Controller (ARC) での準備状況チェックでこれを効率化するために、クロスアカウント認証を使用できます。

ARC のクロスアカウント認証は、準備状況チェック機能と連携します。クロスアカウント認可では、中央にある 1 つの AWS アカウントを使用して、複数の AWS アカウントにあるリソースをモニ

タリングできます。モニタリングするリソースがある各アカウントで、中央のアカウントに、それらのリソースへのアクセスを許可します。それにより、中央のアカウントで、すべてのアカウントのリソースに対する準備状況チェックを作成し、中央のアカウントからフェイルオーバーの準備状況をモニタリングできるようになります。

> **Note**
>
> クロスアカウント認可の設定は、コンソールでは利用できません。代わりに、ARC API オペレーションを使用して、クロスアカウント認証を設定し操作します。開始しやすいように、このセクションでは AWS CLI コマンドの例を示します。

あるアプリケーションに、米国西部 (オレゴン) リージョンにリソースを有するアカウント (us-west-2) があり、さらに、モニタリングするリソースを米国東部 (バージニア北部) リージョンに有するアカウント (us-east-1) もあるとします。ARC は、クロスアカウント認証を使用することにより、1 つのアカウント us-west-2 から両方のリソースセットをモニタリングするためのアクセスを許可できます。

たとえば、次の AWS アカウントがあるとします。

- 米国西部のアカウント: 999999999999
- 米国東部のアカウント: 111111111111

us-east-1 アカウント (111111111111) では、us-west-2 IAM アカウントの (ルート) ユーザーの Amazon リソースネーム (ARN)`arn:aws:iam::999999999999:root`を指定することで、us-west-2 アカウント (999999999999) によるアクセスを許可するクロスアカウント認可を有効にできます。認可を作成すると、us-west-2 アカウントは、us-east-1 が所有するリソースをリソースセットに追加し、そのリソースセットで実行する準備状況チェックを作成できます。

次の例は、1 つのアカウントにクロスアカウント認可を設定する方法を示したものです。ARC で追加およびモニタリングする AWS リソースを持つ追加のアカウントごとに、クロスアカウント認可を有効にする必要があります。

> **Note**
>
> ARC は、複数の AWS リージョンのエンドポイントをサポートするグローバルサービスですが、ほとんどの ARC CLI コマンドで米国西部 (オレゴン) リージョンを指定 (つまり、パラメータ を指定`--region us-west-2`) する必要があります。

以下の AWS CLI コマンドは、こちらの例でクロスアカウント認可を設定する方法を示しています。

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    create-cross-account-authorization --cross-account-authorization
 arn:aws:iam::999999999999:root
```

この認可を無効にするには、次の手順を実行します。

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    delete-cross-account-authorization --cross-account-authorization
 arn:aws:iam::999999999999:root
```

クロスアカウント認証を指定したすべてのアカウントで、特定のアカウントにチェックインするには、list-cross-account-authorizations コマンドを使用します。現時点では、反対方向にチェックインできません。つまり、リソースを追加およびモニタリングするためのクロスアカウント認可が付与されている、アカウントすべてを一覧表示するためのアカウントプロファイルで使用できる API オペレーションはありません。

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    list-cross-account-authorizations
```

```
{
    "CrossAccountAuthorizations": [
        "arn:aws:iam::999999999999:root"
    ]
}
```

準備状況ルール、リソースタイプ、ARNS

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

このセクションでは、準備状況ルールの説明、サポートされているリソースタイプ、リソースセットに使用する Amazon リソースネーム (ARN) の形式に関するリファレンス情報について説明します。

ARC での準備状況ルールの説明

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

このセクションでは、Amazon Application Recovery Controller (ARC) でサポートされているすべての種類のリソースに関する、準備状況ルールの説明を一覧にしています。ARC でサポートされているリソースタイプのリストについては、「ARC のリソースタイプと ARN フォーマット」を参照してください。

準備状況ルールの説明は、ARC コンソールで、または API オペレーションを使用して表示することもできます。方法は以下のとおりです。

- コンソールで準備状況ルールを表示するには、「コンソールに準備状況ルールを表示する」の手順に従います。
- API を使用して準備状況ルールを表示するには、ListRules オペレーションを参照してください。

トピック

- ARC での準備状況ルール
- コンソールに準備状況ルールを表示する

ARC での準備状況ルール

このセクションでは、ARC でサポートされている各リソースタイプの、一連の準備状況ルールを一覧にしています。

ルールの説明を見ると、ほとんどのルールで「すべての ~ を検査」または「各 ~ を検査」という文言が使われていることがわかります。これらの文言が、準備状況チェックのコンテキストで、ルールの機能をどのように説明しているのかについて、また、ARC が準備状況ステータスを設定する方法に

関するその他の詳細については、「準備状況ルールが準備状況ステータスを判断する仕組み」を参照してください。

準備状況ルール

ARC は、以下の準備状況ルールを使用してリソースをモニタリングします。

Amazon API Gateway バージョン 1 ステージ

- ApiGwV1ApiKeyCount: すべての API Gateway ステージを検査し、それらにリンクされている API キーの数が同数であることを確認します。
- ApiGwV1ApiKeySource: すべての API Gateway ステージを検査し、それらの `API Key Source` の値が同じであることを確認します。
- ApiGwV1BasePath: すべての API Gateway ステージを検査し、それらが同じベースパスにリンクされていることを確認します。
- ApiGwV1BinaryMediaTypes: すべての API Gateway ステージを検査し、それらが同じバイナリメディアタイプをサポートしていることを確認します。
- ApiGwV1CacheClusterEnabled: すべての API Gateway ステージを検査し、それらのすべてで `Cache Cluster` が有効になっているか、すべてで無効になっていることを確認します。
- ApiGwV1CacheClusterSize: すべての API Gateway ステージを検査し、それらの `Cache Cluster Size` が同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は NOT READY と表示されます。
- ApiGwV1CacheClusterStatus: すべての API Gateway ステージを検査し、`Cache Cluster` が AVAILABLE の状態になっていることを確認します。
- ApiGwV1DisableExecuteApiEndpoint: すべての API Gateway ステージを検査し、すべてで `Execute API Endpoint` が無効になっているか、いずれも無効になっていないことを確認します。
- ApiGwV1DomainName: すべての API Gateway ステージを検査し、同じドメイン名にリンクされていることを確認します。
- ApiGwV1EndpointConfiguration: すべての API Gateway ステージを検査し、同じエンドポイント設定でドメインにリンクされていることを確認します。
- ApiGwV1EndpointDomainNameStatus: すべての API Gateway ステージを検査し、それらにリンクしているドメイン名が AVAILABLE の状態であることを確認します。
- ApiGwV1MethodSettings: すべての API Gateway ステージを検査し、それらの `Method Settings` の値が同じであることを確認します。

- ApiGwV1MutualTlsAuthentication: すべての API Gateway ステージを検査し、それらの `Mutual TLS Authentication` の値が同じであることを確認します。
- ApiGwV1Policy: すべての API Gateway ステージを検査し、それらのすべてで API レベルのポリシーが使用されているか、またはすべてで使用されていないことを確認します。
- ApiGwV1RegionalDomainName: すべての API Gateway ステージを検査し、それらが同じリージョンのドメイン名にリンクされていることを確認します。注: このルールは準備状況ステータスには影響しません。
- ApiGwV1ResourceMethodConfigs: すべての API Gateway ステージを検査し、それらが、関連する設定を含め、同様のリソースの階層を持っていることを確認します。
- ApiGwV1SecurityPolicy: すべての API Gateway ステージを検査し、それらの `Security Policy` の値が同じであることを確認します。
- ApiGwV1Quotas: すべての API Gateway グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。
- ApiGwV1UsagePlans: すべての API Gateway ステージを検査し、それらが同じ設定で `Usage Plans` にリンクされていることを確認します。

Amazon API Gateway バージョン 2 ステージ

- ApiGwV2ApiKeySelectionExpression: すべての API Gateway ステージを検査し、それらの `API Key Selection Expression` の値が同じであることを確認します。
- ApiGwV2ApiMappingSelectionExpression: すべての API Gateway ステージを検査し、それらの `API Mapping Selection Expression` の値が同じであることを確認します。
- ApiGwV2CorsConfiguration: すべての API Gateway ステージを検査し、それらの CORS 関連の設定が同じであることを確認します。
- ApiGwV2DomainName: すべての API Gateway ステージを検査し、それらが同じドメイン名にリンクされていることを確認します。
- ApiGwV2DomainNameStatus: すべての API Gateway ステージを検査し、ドメイン名が AVAILABLE の状態になっていることを確認します。
- ApiGwV2EndpointType: すべての API Gateway ステージを検査し、それらの `Endpoint Type` の値が同じであることを確認します。
- ApiGwV2Quotas: すべての API Gateway グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。
- ApiGwV2MutualTlsAuthentication: すべての API Gateway ステージを検査し、それらの `Mutual TLS Authentication` の値が同じであることを確認します。

- ApiGwV2ProtocolType: すべての API Gateway ステージを検査し、それらの `Protocol Type` の値が同じであることを確認します。
- ApiGwV2RouteConfigs: すべての API Gateway ステージを検査し、それらが同じ設定の、同じルートの階層を持つことを確認します。
- ApiGwV2RouteSelectionExpression: すべての API Gateway ステージを検査し、それらの `Route Selection Expression` の値が同じであることを確認します。
- ApiGwV2RouteSettings: すべての API Gateway ステージを検査し、それらの `Default Route Settings` の値が同じであることを確認します。
- ApiGwV2SecurityPolicy: すべての API Gateway ステージを検査し、それらの `Security Policy` の値が同じであることを確認します。
- ApiGwV2StageVariables: すべての API Gateway ステージを検査し、それらのすべてが他のステージと同じ `Stage Variables` を持っていることを確認します。
- ApiGwV2ThrottlingBurstLimit: すべての API Gateway ステージを検査し、それらの `Throttling Burst Limit` の値が同じであることを確認します。
- ApiGwV2ThrottlingRateLimit: すべての API Gateway ステージを検査し、それらの `Throttling Rate Limit` の値が同じであることを確認します。

Amazon Aurora クラスター

- RdsClusterStatus: 各 Aurora クラスターを検査し、ステータスが `AVAILABLE` または `BACKING-UP` であることを確認します。
- RdsEngineMode: すべての Aurora クラスターを検査し、それらの `Engine Mode` の値が同じであることを確認します。
- RdsEngineVersion: すべての Aurora クラスターを検査し、それらの `Major Version` の値が同じであることを確認します。
- RdsGlobalReplicaLag:各 Aurora クラスターを検査し、`Global Replica Lag` が 30 秒未満であることを確認します。
- RdsNormalizedCapacity: すべての Aurora クラスターを検査し、それらの正規化された容量が、リソースセットの最大容量の 15% 以内であることを確認します。
- RdsInstanceType: すべての Aurora クラスターを検査し、それらのインスタンスタイプが同じであることを確認します。
- RdsQuotas: すべての Aurora クラスターを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

「Auto Scaling グループ」

- AsgMinSizeAndMaxSize: すべての Auto Scaling グループを検査し、それらの最小グループのサイズと最大グループのサイズが同じであることを確認します。
- AsgAZCount: すべての Auto Scaling グループを検査し、それらのアベイラビリティーゾーンの数が同じであることを確認します。
- AsgInstanceTypes: すべての Auto Scaling グループを検査し、それらのインスタンスタイプが同じであることを確認します。注: このルールは準備状況ステータスには影響しません。
- AsgInstanceSizes: すべての Auto Scaling グループを検査し、それらのインスタンスのサイズが同じであることを確認します。
- AsgNormalizedCapacity: すべての Auto Scaling グループを検査し、それらの正規化された容量が、リソースセットの最大容量の 15% 以内であることを確認します。
- AsgQuotas: すべての Auto Scaling グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

CloudWatch アラーム

- CloudWatchAlarmState: CloudWatch アラームを検査し、いずれも `ALARM` または `INSUFFICIENT_DATA` の状態ではないことを確認します。

カスタマーゲートウェイ

- CustomerGatewayIpAddress: すべてのカスタマーゲートウェイを検査し、それらの IP アドレスが同じであることを確認します。
- CustomerGatewayState: カスタマーゲートウェイを検査し、いずれも `AVAILABLE` の状態になっていることを確認します。
- CustomerGatewayVPNType: すべてのカスタマーゲートウェイを検査し、それらの VPN タイプが同じであることを確認します。

DNS target resources

- DnsTargetResourceHostedZoneConfigurationRule: すべての DNS ターゲットリソースを検査し、それらの Amazon Route 53 のホストゾーン ID が同じであり、各ホストゾーンがプライベートではないことを確認します。注: このルールは準備状況ステータスには影響しません。
- DnsTargetResourceRecordSetConfigurationRule: すべての DNS ターゲットリソースを検査し、それらのリソースレコードのキャッシュ有効期限 (TTL) が同じで、TTL が 300 以下であることを確認します。
- DnsTargetResourceRoutingRule: エイリアスのリソースレコードセットに関連付けられている各 DNS ターゲットリソースを検査し、トラフィックが、ターゲットリソースで設定された

DNS 名にルーティングされていることを確認します。注: このルールは準備状況ステータスには影響しません。

- DnsTargetResourceHealthCheckRule: すべての DNS ターゲットリソースを検査し、ヘルスチェックがそれぞれのリソースレコードセットに適宜関連付けられ、それ以外の場合は関連付けられていないことを確認します。注: このルールは準備状況ステータスには影響しません。

Amazon DynamoDB テーブル

- DynamoConfiguration: すべての DynamoDB テーブルを検査し、それらのキー、属性、サーバー側の暗号化、ストリーム設定が同じであることを確認します。
- DynamoTableStatus: 各 DynamoDB テーブルを検査し、ステータスが ACTIVE になっていることを確認します。
- DynamoCapacity: すべての DynamoDB テーブルを検査し、それらのプロビジョニングされた読み込みキャパシティと書き込みキャパシティが、リソースセットの最大容量の 20% 以内であることを確認します。
- DynamoPeakRcuWcu: 各 DynamoDB テーブルを検査し、ピークトラフィックが他のテーブルと同程度に発生し、プロビジョニングされた容量が確保されていることを確認します。
- DynamoGsiPeakRcuWcu: 各 DynamoDB テーブルを検査し、読み取りと書き込みの最大キャパシティが他のテーブルと同程度であり、プロビジョニングされた容量が確保されていることを確認します。
- DynamoGsiConfig: グローバルセカンダリインデックスを持つすべての DynamoDB テーブルを検査し、テーブルが同じインデックス、キースキーマ、プロジェクションを使用していることを確認します。
- DynamoGsiStatus: グローバルセカンダリインデックスを持つすべての DynamoDB テーブルを検査し、グローバルセカンダリインデックスのステータスが ACTIVE の状態になっていることを確認します。
- DynamoGsiCapacity: グローバルセカンダリインデックスを持つすべての DynamoDB テーブルを検査し、テーブルの、プロビジョニングされた GSI 読み込みキャパシティと GSI 書き込みキャパシティが、リソースセットの最大容量の 20% 以内であることを確認します。
- DynamoReplicationLatency: グローバルテーブルであるすべての DynamoDB テーブルを検査し、レプリケーションレイテンシーがすべて同じであることを確認します。
- DynamoAutoScalingConfiguration: Auto Scaling が有効になっているすべての DynamoDB テーブルを検査し、それらの最小容量、最大容量、ターゲットの読み取り/書き込みキャパシティが同じであることを確認します。
- DynamoQuotas: すべての DynamoDB テーブルを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Elastic Load Balancing (Classic Load Balancer)

- ElbV1CheckAzCount: 各 Classic Load Balancer を検査し、アタッチされているアベイラビリティーゾーンが 1 つのみであることを確認します。注: このルールは準備状況ステータスには影響しません。
- ElbV1AnyInstances: すべての Classic Load Balancer を検査し、それらに EC2 インスタンスが 1 つ以上あることを確認します。
- ElbV1AnyInstancesHealthy: すべての Classic Load Balancer を検査し、それらに正常な EC2 インスタンスが 1 つ以上あることを確認します。
- ElbV1Scheme: すべての Classic Load Balancer を検査し、それらのロードバランサースキームが同じであることを確認します。
- ElbV1HealthCheckThreshold: すべての Classic Load Balancer を検査し、それらのヘルスチェックのしきい値が同じであることを確認します。
- ElbV1HealthCheckInterval: すべての Classic Load Balancer を検査し、それらのヘルスチェックの間隔値が同じであることを確認します。
- ElbV1CrossZoneRoutingEnabled: すべての Classic Load Balancer を検査し、それらのクロスゾーン負荷分散の値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV1AccessLogsEnabledAttribute: すべての Classic Load Balancer を検査し、それらのアクセスログの値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV1ConnectionDrainingEnabledAttribute: すべての Classic Load Balancer を検査し、それらの Connection Draining の値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV1ConnectionDrainingTimeoutAttribute: すべての Classic Load Balancer を検査し、それらの Connection Draining のタイムアウト値が同じであることを確認します。
- ElbV1IdleTimeoutAttribute: すべての Classic Load Balancer を検査し、それらのアイドルタイムアウトの値が同じであることを確認します。
- ElbV1ProvisionedCapacityLcuCount: プロビジョニングされた LCU が 10 を超えているすべての Classic Load Balancer を検査し、それらが、リソースセット内にあるプロビジョニング済み LCU の最大値の 20% 以内であることを確認します。
- ElbV1ProvisionedCapacityStatus: 各 Classic Load Balancer のプロビジョニング済み容量のステータスを検査し、値が DISABLED または PENDING になっていないことを確認します。

Amazon EBS ボリューム

- EbsVolumeEncryption: すべての EBS ボリュームを検査し、それらの暗号化の値が同じ (ENABLED または DISABLED) であることを確認します。

- EbsVolumeEncryptionDefault: すべての EBS ボリュームを検査し、それらのデフォルトの暗号化の値が同じ (ENABLED または DISABLED) であることを確認します。
- EbsVolumeIops: すべての EBS ボリュームを検査し、それらの 1 秒あたりの入出力オペレーション (IOPS) が同じであることを確認します。
- EbsVolumeKmsKeyId: すべてのEBSボリュームを検査し、デフォルトの AWS KMS キー ID が同じであることを確認します。
- EbsVolumeMultiAttach: すべての EBS ボリュームを検査し、それらのマルチアタッチの値が同じ (ENABLED または DISABLED) であることを確認します。
- EbsVolumeQuotas: すべての EBS ボリュームを検査し、それらが、Service Quotas が設定するクォータ (制限) に従っていることを確認します。
- EbsVolumeSize: すべての EBS ボリュームを検査し、それらの読み取り可能なサイズが同じであることを確認します。
- EbsVolumeState: すべての EBS ボリュームを検査し、それらのボリュームの状態が同じであることを確認します。
- EbsVolumeType: すべての EBS ボリュームを検査し、それらのボリュームタイプが同じであることを確認します。

AWS Lambda 関数

- LambdaMemorySize: すべての Lambda 関数を検査し、それらのメモリサイズが同じであることを確認します。メモリがこれよりも大きい関数が 1 つある場合、それ以外は `NOT READY` と表示されます。
- LambdaFunctionTimeout: すべての Lambda 関数を検査し、それらのタイムアウト値が同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は `NOT READY` と表示されます。
- LambdaFunctionRuntime: すべての Lambda 関数を検査し、それらのランタイムがすべて同じであることを確認します。
- LambdaFunctionReservedConcurrentExecutions: すべての Lambda 関数を検査し、それらの `Reserved Concurrent Executions` の値がすべて同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は `NOT READY` と表示されます。
- LambdaFunctionDeadLetterConfig: すべての Lambda 関数を検査し、すべてで `Dead Letter Config` が定義されているか、それともすべてで定義されていないか、いずれかであることを確認します。
- LambdaFunctionProvisionedConcurrencyConfig: すべての Lambda 関数を検査し、それらの `Provisioned Concurrency` の値が同じであることを確認します。

- LambdaFunctionSecurityGroupCount: すべての Lambda 関数を検査し、それらの `Security Groups` の値が同じであることを確認します。
- LambdaFunctionSubnetIdCount: すべての Lambda 関数を検査し、それらの `Subnet Ids` の値が同じであることを確認します。
- LambdaFunctionEventSourceMappingMatch: すべての Lambda 関数を検査し、選択した `Event Source Mapping` のプロパティがすべて、互いに一致していることを確認します。
- LambdaFunctionLimitsRule: すべての Lambda 関数を検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Network Load Balancer と Application Load Balancer

- ElbV2CheckAzCount: 各 Network Load Balancer を検査し、アタッチされているアベイラビリティーゾーンが 1 つのみであることを確認します。注: このルールは準備状況ステータスには影響しません。
- ElbV2TargetGroupsCanServeTraffic: 各 Network Load Balancer と Application Load Balancer を検査し、正常な Amazon EC2 インスタンスが 1 つ以上あることを確認します。
- ElbV2State: 各 Network Load Balancer と Application Load Balancer を検査し、ステータスが `ACTIVE` になっていることを確認します。
- ElbV2IpAddressType: すべての Network Load Balancer と Application Load Balancer を検査し、それらの IP アドレスのタイプが同じであることを確認します。
- ElbV2Scheme: すべての Network Load Balancer と Application Load Balancer を検査し、それらのスキームが同じであることを確認します。
- ElbV2Type: すべての Network Load Balancer と Application Load Balancer を検査し、それらのタイプが同じであることを確認します。
- ElbV2S3LogsEnabled: すべての Network Load Balancer と Application Load Balancer を検査し、それらの Amazon S3 サーバーアクセスログの値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV2DeletionProtection: すべての Network Load Balancer と Application Load Balancer を検査し、それらの削除保護の値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV2IdleTimeoutSeconds: すべての Network Load Balancer と Application Load Balancer を検査し、それらのアイドル時間の秒数が同じであることを確認します。
- ElbV2HttpDropInvalidHeaders: すべての Network Load Balancer と Application Load Balancer を検査し、それらの「無効なヘッダーを削除」の値が同じであることを確認します。
- ElbV2Http2Enabled: すべての Network Load Balancer と Application Load Balancer を検査し、それらの HTTP2 の値が同じ (ENABLED または DISABLED) であることを確認します。

- ElbV2CrossZoneEnabled: すべての Network Load Balancer と Application Load Balancer を検査し、それらのクロスゾーン負荷分散の値が同じ (ENABLED または DISABLED) であることを確認します。
- ElbV2ProvisionedCapacityLcuCount: プロビジョニングされた LCU が 10 を超えているすべての Network Load Balancer と Application Load Balancer を検査し、それらが、リソースセット内にあるプロビジョニング済み LCU の、最大値の 20% 以内であることを確認します。
- ElbV2ProvisionedCapacityEnabled: すべての Network Load Balancer と Application Load Balancer の、プロビジョニング済み容量のステータスを検査し、それらの値が DISABLED または PENDING になっていないことを確認します。

Amazon MSK クラスター

- MskClusterClientSubnet: 各 MSK クラスターを検査し、クライアントサブネットが 2 つまたは 3 つのみであることを確認します。
- MskClusterInstanceType: すべての MSK クラスターを検査し、それらの Amazon EC2 のインスタンスタイプが同じであることを確認します。
- MskClusterSecurityGroups: すべての MSK クラスターを検査し、それらのセキュリティグループが同じであることを確認します。
- MskClusterStorageInfo: すべての MSK クラスターを検査し、それらの EBS ストレージボリュームのサイズが同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は NOT READY と表示されます。
- MskClusterACMCertificate: すべての MSK クラスターを検査し、それらのクライアント認可証明書 ARN のリストが同じであることを確認します。
- MskClusterServerProperties: すべての MSK クラスターを検査し、それらの `Current Broker Software Info` の値が同じであることを確認します。
- MskClusterKafkaVersion: すべての MSK クラスターを検査し、それらの Kafka のバージョンが同じであることを確認します。
- MskClusterEncryptionInTransitInCluster: すべての MSK クラスターを検査し、それらの `Encryption In Transit In Cluster` の値が同じであることを確認します。
- MskClusterEncryptionInClientBroker: すべての MSK クラスターを検査し、それらの `Encryption In Transit Client Broker` の値が同じであることを確認します。
- MskClusterEnhancedMonitoring: すべての MSK クラスターを検査し、それらの `Enhanced Monitoring` の値が同じであることを確認します。
- MskClusterOpenMonitoringInJmx: すべての MSK クラスターを検査し、それらの `Open Monitoring JMX Exporter` の値が同じであることを確認します。

- MskClusterOpenMonitoringInNode: すべての MSK クラスターを検査し、それらの `Open Monitoring Not Exporter.` の値が同じであることを確認します。
- MskClusterLoggingInS3: すべての MSK クラスターを検査し、それらの `Is Logging in S3` の値が同じであることを確認します。
- MskClusterLoggingInFirehose: すべての MSK クラスターを検査し、それらの `Is Logging In Firehose` の値が同じであることを確認します。
- MskClusterLoggingInCloudWatch: すべての MSK クラスターを検査し、それらの `Is Logging Available In CloudWatch Logs` の値が同じであることを確認します。
- MskClusterNumberOfBrokerNodes: すべての MSK クラスターを検査し、それらの `Number of Broker Nodes` の値が同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は NOT READY と表示されます。
- MskClusterState: 各 MSK クラスターを検査し、それらのステータスが ACTIVE になっていることを確認します。
- MskClusterLimitsRule: すべての Lambda 関数を検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Amazon Route 53 ヘルスチェック

- R53HealthCheckType: 各 Route 53 ヘルスチェックを検査し、それらのタイプが CALCULATED ではなく、すべてのチェックが同じタイプであることを確認します。
- R53HealthCheckDisabled: 各 Route 53 ヘルスチェックを検査し、ステータスが DISABLED になっていないことを確認します。
- R53HealthCheckStatus: 各 Route 53 ヘルスチェックを検査し、ステータスが SUCCESS になっていることを確認します。
- R53HealthCheckRequestInterval: すべての Route 53 ヘルスチェックを検査し、`Request Interval` の値がすべて同じあることを確認します。
- R53HealthCheckFailureThreshold: すべての Route 53 ヘルスチェックを検査し、`Failure Threshold.` の値がすべて同じあることを確認します。
- R53HealthCheckEnableSNI: すべての Route 53 ヘルスチェックを検査し、`Enable SNI.` の値がすべて同じあることを確認します。
- R53HealthCheckSearchString: すべての Route 53 ヘルスチェックを検査し、`Search String.` の値がすべて同じあることを確認します。
- R53HealthCheckRegions: すべての Route 53 ヘルスチェックを検査し、 AWS リージョンのリストがすべて同じあることを確認します。

- R53HealthCheckMeasureLatency: すべての Route 53 ヘルスチェックを検査し、`Measure Latency` の値がすべて同じあることを確認します。
- R53HealthCheckInsufficientDataHealthStatus: すべての Route 53 ヘルスチェックを検査し、`Insufficient Data Health Status` の値がすべて同じあることを確認します。
- R53HealthCheckInverted: すべての Route 53 ヘルスチェックを検査し、すべて反転しているか、または、すべてが反転していないことを確認します。
- R53HealthCheckResourcePath: すべての Route 53 ヘルスチェックを検査し、`Resource Path` の値がすべて同じあることを確認します。
- R53HealthCheckCloudWatchAlarm: すべての Route 53 ヘルスチェックを検査し、それらに関連付けられている CloudWatch アラームの設定と構成が、同じであることを確認します。

Amazon SNS サブスクリプション

- SnsSubscriptionProtocol: すべての SNS サブスクリプションを検査し、プロトコルが同じであることを確認します。
- SnsSubscriptionSqsLambdaEndpoint: Lambda または SQS エンドポイントを持つすべての SNS サブスクリプションを検査し、エンドポイントがそれぞれ異なることを確認します。
- SnsSubscriptionNonAwsEndpoint: E メールなど、AWS サービス以外のエンドポイントタイプを持つすべての SNS サブスクリプションを検査し、サブスクリプションに同じエンドポイントがあることを確認します。
- SnsSubscriptionPendingConfirmation: すべての SNS サブスクリプションを検査し、それらの [保留中の確認] の値が同じであることを確認します。
- SnsSubscriptionDeliveryPolicy: HTTP/S を使用するすべての SNS サブスクリプションを検査し、[有効なデリバリー期間] の値が同じであることを確認します。
- SnsSubscriptionRawMessageDelivery: すべての SNS サブスクリプションを検査し、それらの [raw メッセージの配信] の値が同じであることを確認します。
- SnsSubscriptionFilter: すべての SNS サブスクリプションを検査し、それらの [フィルターポリシー] の値が同じであることを確認します。
- SnsSubscriptionRedrivePolicy: すべての SNS サブスクリプションを検査し、それらの [リドライブポリシー] の値が同じであることを確認します。
- SnsSubscriptionEndpointEnabled: すべての SNS サブスクリプションを検査し、それらの [エンドポイントの有効化] の値が同じであることを確認します。
- SnsSubscriptionLambdaEndpointValid: Lambda エンドポイントを持つすべての SNS サブスクリプションを検査し、有効な Lambda エンドポイントがあることを確認します。

- SnsSubscriptionSqsEndpointValidRule: SQS エンドポイントを使用するすべての SNS サブスクリプションを検査し、有効な SQS エンドポイントがあることを確認します。
- SnsSubscriptionQuotas: すべての SNS サブスクリプションを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Amazon SNS トピック

- SnsTopicDisplayName: すべての SNS トピックを検査し、それらの `Display Name` の値が同じであることを確認します。
- SnsTopicDeliveryPolicy: HTTPS サブスクライバーを持つすべての SNS トピックを検査し、`EffectiveDeliveryPolicy` が同じであることを確認します。
- SnsTopicSubscription: すべての SNS トピックを検査し、各プロトコルのサブスクライバー数が同じであることを確認します。
- SnsTopicAwsKmsKey: すべての SNS トピックを検査し、すべてのトピックに AWS KMS キーがあるか、いずれのトピックにもこのキーがないことを確認します。
- SnsTopicQuotas: すべての SNS トピックを検査し、それらが Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Amazon SQS キュー

- SqsQueueType: すべての SQS キューを検査し、`Type` の値がすべて同じであることを確認します。
- SqsQueueDelaySeconds: すべての SQS キューを検査し、`Delay Seconds` の値がすべて同じであることを確認します。
- SqsQueueMaximumMessageSize: すべての SQS キューを検査し、`Maximum Message Size` の値がすべて同じであることを確認します。
- SqsQueueMessageRetentionPeriod: すべての SQS キューを検査し、`Message Retention Period` の値がすべて同じであることを確認します。
- SqsQueueReceiveMessageWaitTimeSeconds: すべての SQS キューを検査し、`Receive Message Wait Time Seconds` の値がすべて同じであることを確認します。
- SqsQueueRedrivePolicyMaxReceiveCount: すべての SQS キューを検査し、`Redrive Policy Max Receive Count` の値がすべて同じであることを確認します。
- SqsQueueVisibilityTimeout: すべての SQS キューを検査し、`Visibility Timeout` の値がすべて同じであることを確認します。
- SqsQueueContentBasedDeduplication: すべての SQS キューを検査し、`Content-Based Deduplication` の値がすべて同じであることを確認します。

- SqsQueueQuotas: すべての SQS キューを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Amazon VPC

- VpcCidrBlock: すべての VPC を検査し、CIDR ブロックネットワークサイズの値がすべて同じであることを確認します。
- VpcCidrBlocksSameProtocolVersion: 同じ CIDR ブロックを持つすべての VPC を検査し、それらのインターネットストリームプロトコルのバージョン番号の値が同じであることを確認します。
- VpcCidrBlocksStateInAssociationSets: 全 VPC の CIDR ブロックアソシエーションセットをすべて検査し、すべてに `ASSOCIATED` 状態の CIDR ブロックがあることを確認します。
- VpcIpv6CidrBlocksStateInAssociationSets: 全 VPC の CIDR ブロックアソシエーションセットをすべて検査し、すべてに同じアドレス数の CIDR ブロックがあることを確認します。
- VpcCidrBlocksInAssociationSets: 全 VPC の CIDR ブロックアソシエーションセットをすべて検査し、すべてが同じサイズであることを確認します。
- VpcIpv6CidrBlocksInAssociationSets: 全 VPC の IPv6 CIDR ブロックアソシエーションセットをすべて検査し、すべてが同じサイズであることを確認します。
- VpcState: 各 VPC を検査し、`AVAILABLE` の状態であることを確認します。
- VpcInstanceTenancy: すべての VPC を検査し、`Instance Tenancy` の値がすべて同じであることを確認します。
- VpcIsDefault: すべての VPC を検査し、それらの `Is Default.` の値が同じであることを確認します。
- VpcSubnetState: 各 VPC サブネットを検査し、AVAILABLE の状態であることを確認します。
- VpcSubnetAvailableIpAddressCount: 各 VPC サブネットを検査し、使用可能な IP アドレスの数がゼロより多いことを確認します。
- VpcSubnetCount: すべての VPC サブネットを検査し、サブネットの数が同じであることを確認します。
- VpcQuotas: すべての VPC サブネットを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

Site-to-Site VPN 接続

- VpnConnectionsRouteCount: すべての VPN 接続を検査し、ルートが 1 つ以上あり、かつルートの数が同じであることを確認します。
- VpnConnectionsEnableAcceleration: すべての VPN 接続を検査し、それらの `Enable Accelerations` の値が同じであることを確認します。

- VpnConnectionsStaticRoutesOnly: すべての VPN 接続を検査し、それらの `Static Routes Only.` の値が同じであることを確認します。
- VpnConnectionsCategory: すべての VPN 接続を検査し、それらに `VPN` のカテゴリが 1 つあることを確認します。
- VpnConnectionsCustomerConfiguration: すべての VPN 接続を検査し、それらの `Customer Gateway Configuration` の値が同じであることを確認します。
- VpnConnectionsCustomerGatewayId: 各 VPN 接続を検査し、カスタマーゲートウェイが接続されていることを確認します。
- VpnConnectionsRoutesState: すべての VPN 接続を検査し、`AVAILABLE` の状態になっていることを確認します。
- VpnConnectionsVgwTelemetryStatus: 各 VPN 接続を検査し、VGW の状態が `UP` であることを確認します。
- VpnConnectionsVgwTelemetryIpAddress: 各 VPN 接続を検査し、外部 IP アドレスが VGW テレメトリごとに異なっていることを確認します。
- VpnConnectionsTunnelOptions: すべての VPN 接続を検査し、トンネルオプションが同じであることを確認します。
- VpnConnectionsRoutesCidr: すべての VPN 接続を検査し、宛先の CIDR ブロックが同じであることを確認します。
- VpnConnectionsInstanceType: すべての VPN 接続を検査し、`Instance Type` が同じであることを確認します。

Site-to-Site VPN ゲートウェイ

- VpnGatewayState: すべての VPN ゲートウェイを検査し、それらが AVAILABLE の状態になっていることを確認します。
- VpnGatewayAsn: すべての VPN ゲートウェイを検査し、ASN が同じであることを確認します。
- VpnGatewayType: すべての VPN ゲートウェイを検査し、タイプが同じであることを確認します。
- VpnGatewayAttachment: すべての VPN ゲートウェイを検査し、接続設定が同じであることを確認します。

コンソールに準備状況ルールを表示する

準備状況ルールは AWS マネジメントコンソール、各リソースタイプ別にリストされた で表示できます。

コンソールに準備状況ルールを表示するには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [準備状況チェック] を選択します。
3. [リソースタイプ] で、ルールを表示するリソースタイプを選択します。

ARC のリソースタイプと ARN フォーマット

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) でリソースセットを作成するときは、セットに含めるリソースのタイプと、含める各リソースの Amazon リソースネーム (ARN) を指定します。ARC では、リソースタイプごとに特定の ARN 形式が想定されています。このセクションでは、ARC でサポートされているリソースタイプと、各リソースタイプに関連付けられた ARN 形式を一覧にします。

具体的な形式はリソースによって異なります。ARN を指定するには、######のテキストを、リソース固有の情報に置き換えます。

> **Note**
>
> ARC がリソースで必要とする ARN 形式は、サービス自体がそのリソースで必要とする ARN 形式とは異なる場合がありますのでご注意ください。たとえば、サービス認可リファレンスの各サービスのリソースタイプセクションで説明されている ARN 形式には、ARC サービスで機能をサポートするために ARC が必要とする AWS アカウント ID やその他の情報が含まれていない場合があります。

AWS::ApiGateway::Stage

Amazon API Gateway バージョン 1 ステージ

- ARN 形式: `arn:partition:apigateway:region:account:/restapis/api-id/stages/stage-name`

  例: `arn:aws:apigateway:us-east-1:111122223333:/restapis/123456789/stages/ExampleStage`

  詳細については、「API Gateway Amazon リソースネーム (ARN) リファレンス」を参照してください。

AWS::ApiGatewayV2::Stage

Amazon API Gateway バージョン 2 ステージ

- ARN 形式: `arn:partition:apigateway:region:account:/apis/api-id/stages/stage-name`

  例: `arn:aws:apigateway:us-east-1:111122223333:/apis/123456789/stages/ExampleStage`

  詳細については、「API Gateway Amazon リソースネーム (ARN) リファレンス」を参照してください。

AWS::CloudWatch::Alarm

Amazon CloudWatch アラーム

- ARN 形式: `arn:partition:cloudwatch:region:account:alarm:alarm-name`

  例: `arn:aws:cloudwatch:us-west-2:111122223333:alarm:test-alarm-1`

  詳細については、「Amazon CloudWatch で定義されるリソースタイプ」を参照してください。

AWS::DynamoDB::Table

Amazon DynamoDB テーブル

- ARN 形式: `arn:partition:dynamodb:region:account:table/table-name`

  例: `arn:aws:dynamodb:us-west-2:111122223333:table/BigTable`

  詳細については、「DynamoDB resources and operations」を参照してください。

AWS::EC2::CustomerGateway

カスタマーゲートウェイデバイス

- ARN 形式: `arn:partition:ec2:region:account:customer-gateway/CustomerGatewayId`

  例: `arn:aws:ec2:us-west-2:111122223333:customer-gateway/vcg-123456789`

  詳細については、「Amazon EC2 で定義されるリソースタイプ」を参照してください。

AWS::EC2::Volume

Amazon EBS ボリューム

- ARN 形式: `arn:partition:ec2:region:account:volume/VolumeId`

  例: `arn:aws:ec2:us-west-2:111122223333:volume/volume-of-cylinder-is-pi`

  詳細については、「API Gateway Amazon リソースネーム (ARN) リファレンス」を参照してください。

AWS::ElasticLoadBalancing::LoadBalancer

Classic Load Balancer

- ARN 形式: `arn:partition:elasticloadbalancing:region:account:loadbalancer/LoadBalancerN`

  例: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/123456789abcbdeCLB`

  詳細については、「Elastic Load Balancing resources」を参照してください。

AWS::ElasticLoadBalancingV2::LoadBalancer

Application Load Balancer または Network Load Balancer

- Network Load Balancer の ARN 形式: `arn:partition:elasticloadbalancing:region:account:loadbalancer/net/LoadBalancerName`

  Network Load Balancer の例: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/net/sandbox-net/123456789acbdeNLB`

- Application Load Balancer の ARN 形式: `arn:partition:elasticloadbalancing:region:account:loadbalancer/app/LoadBalancerName`

Application Load Balancer の例: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/app/sandbox-alb/123456789acbdeALB`

詳細については、「Elastic Load Balancing resources」を参照してください。

AWS::Lambda::Function

AWS Lambda 関数。

- ARN 形式: `arn:partition:lambda:region:account:function:FunctionName`

  例: `arn:aws:lambda:us-west-2:111122223333:function:my-function`

  詳細については、「Lambda アクションのリソースと条件」を参照してください。

AWS::MSK::Cluster

Amazon MSK クラスター

- ARN 形式: `arn:partition:kafka:region:account:cluster/ClusterName/UUID`

  例: `arn:aws:kafka:us-east-1:111122223333:cluster/demo-cluster-1/123456-1111-2222-3333`

  詳細については、「Amazon Managed Streaming for Apache Kafka で定義されるリソースタイプ」を参照してください。

AWS::RDS::DBCluster

Aurora DB クラスター

- ARN 形式: `arn:partition:rds:region:account:cluster:DbClusterInstanceName`

  例: `arn:aws:rds:us-west-2:111122223333:cluster:database-1`

  詳細については、「Amazon RDS の Amazon リソースネーム (ARN) の使用」を参照してください。

AWS::Route53::HealthCheck

Amazon Route 53 ヘルスチェック

- ARN 形式: `arn:partition:route53:::healthcheck/Id`

  例: `arn:aws:route53:::healthcheck/123456-1111-2222-3333`

AWS::SQS::Queue

Amazon SQS キュー

- ARN 形式: `arn:partition:sqs:region:account:QueueName`

  例: `arn:aws:sqs:us-west-2:111122223333:StandardQueue`

  詳細については、「Amazon Simple Queue Service resource and operations」を参照してください。

AWS::SNS::Topic

Amazon SNS トピック

- ARN 形式: `arn:partition:sns:region:account:TopicName`

  例: `arn:aws:sns:us-west-2:111122223333:TopicName`

  詳細については、「Amazon SNS リソース ARN 形式」を参照してください。

AWS::SNS::Subscription

Amazon SNS サブスクリプション

- ARN 形式: `arn:partition:sns:region:account:TopicName:SubscriptionId`

  例: `arn:aws:sns:us-west-2:111122223333:TopicName:12345678901234567890`

AWS::EC2::VPC

Virtual Private Cloud (VPC)。

- ARN 形式: `arn:partition:ec2:region:account:vpc/VpcId`

  例: `arn:aws:ec2:us-west-2:111122223333:vpc/vpc-123456789`

  詳細については、「VPC Resources」を参照してください。

AWS::EC2::VPNConnection

仮想プライベートネットワーク (VPN) 接続

- ARN 形式: `arn:partition:ec2:region:account:vpn-connection/VpnConnectionId`

  例: `arn:aws:ec2:us-west-2:111122223333:vpn-connection/vpn-123456789`

  詳細については、「Amazon EC2 で定義されるリソースタイプ」を参照してください。

AWS::EC2::VPNGateway

仮想プライベートネットワーク (VPN) ゲートウェイ

- ARN 形式: `arn:partition:ec2:region:account:vpn-gateway/VpnGatewayId`

  例: `arn:aws:ec2:us-west-2:111122223333:vpn-gateway/vgw-123456789acbdefgh`

  詳細については、「Amazon EC2 で定義されるリソースタイプ」を参照してください。

AWS::Route53RecoveryReadiness::DNSTargetResource

準備状況チェックの DNS ターゲットリソースには、DNS レコードタイプ、ドメイン名、Route 53 ホストゾーン ARN、そして Network Load Balancer ARN か Route 53 レコードセット ID のいずれかが含まれています。

- ホストゾーンの ARN 形式: `arn:partition:route53::account:hostedzone/Id`

  ホストゾーンの例: `arn:aws:route53::111122223333:hostedzone/abcHostedZone`

  注: こちらに示すとおり、ホストゾーン ARN にはアカウント ID を含める必要があります。ARC がリソースをポーリングできるようにするにはアカウント ID が必要です。この形式が、「サービス認証リファレンス」の Route 53 サービスリソースタイプのセクションで説明されている Amazon Route 53 が必要とする ARN の形式と異なるのは、意図的なものです。

- Network Load Balancer の ARN 形式: `arn:partition:elasticloadbalancing:region:account:loadbalancer/net/LoadBalancerName`

  Network Load Balancer の例: `arn:aws:elasticloadbalancing:us-west-2:111122223333:loadbalancer/net/sandbox-net/123456789acbdefgh`

  詳細については、「Elastic Load Balancing resources」を参照してください。

Amazon Application Recovery Controller (ARC) の準備状況チェックのログ記録とモニタリング

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引

き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon CloudWatch、および Amazon EventBridge を使用して AWS CloudTrail、Amazon Application Recovery Controller (ARC) の準備状況チェックをモニタリングし、パターンを分析し、問題のトラブルシューティングに役立てることができます。

> **Note**
>
> 米国西部 (オレゴン) リージョンの ARC に関する CloudWatch メトリクスとログは、両方ともコンソール上および AWS CLIの使用時に表示する必要があります。を使用する場合は AWS CLI、 パラメータ を含めて、コマンドの米国西部 (オレゴン) リージョンを指定します`--region us-west-2`。

トピック

- ARC の準備状況チェックでの Amazon CloudWatch の使用
- を使用した準備状況チェック API コールのログ記録 AWS CloudTrail
- Amazon EventBridge により ARC で準備状況チェックを使用する

ARC の準備状況チェックでの Amazon CloudWatch の使用

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) は、準備状況をチェックするために、Amazon CloudWatch にデータポイントを発行します。CloudWatch では、それらのデータポイントについての統計を、(メトリクスと呼ばれる) 順序付けられた時系列データのセットとして取得できます。メトリクスは監視対象の変数、データポイントは時間の経過と共に変わる変数の値と考えることができます。たとえば、指定した期間に AWS リージョンを通過するトラフィックをモニタリングできます。各データポイントには、タイムスタンプと、オプションの測定単位が関連付けられています。

メトリクスを使用して、システムが正常に実行されていることを確認できます。例えば、メトリクスが許容範囲外になる場合、CloudWatch アラームを作成して、指定されたメトリクスを監視し、アクション (E メールアドレスに通知を送信するなど) を開始することができます。

詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

トピック

- ARC メトリクス
- ARC メトリクスの統計
- ARC の CloudWatch メトリクスを表示する

ARC メトリクス

AWS/Route53RecoveryReadiness 名前空間には、次のメトリクスが含まれます。

メトリクス	説明
ReadinessChecks	ARC によって処理された準備状況のチェックの数を表します。このメトリクスは、以下に示すように状態別にディメンション化できます。 単位: Count。 レポート条件: ゼロ以外の値がある。 統計: 使用できる統計は Sum のみです。 ディメンション • READY • NOT_READY • NOT_AUTHORIZED • UNKNOWN
Resources	ARC によって処理されるリソースの数を表し、API で定義されているリソース識別子によってディメンション化できます。 単位: Count。

メトリクス	説明
	レポート条件: ゼロ以外の値がある。 統計: 使用できる統計は Sum のみです。 ディメンション • ResourceSetType ：これらはリソースタイプであり、ARC によって評価された特定のタイプごとに、リソース数でフィルタリングされます。 例: AWS::CloudWatch::Alarm

ARC メトリクスの統計

CloudWatch では、ARC が発行したメトリクスのデータポイントに基づき、統計が表示されます。統計とは、指定された期間のメトリクスデータを集計したものです。統計を要求した場合、返されるデータストリームはメトリクス名とディメンションによって識別されます。ディメンションは、メトリクスを一意に識別する名前/値のペアです。

以下は、役に立つメトリクス/ディメンションの組み合わせの例です。

- ARC によって準備状況が評価された準備状況チェックの数を表します。
- ARC によって評価された、特定のリソースセットタイプの合計リソース数を表します。

ARC の CloudWatch メトリクスを表示する

ARC に関する CloudWatch メトリクスを表示するときは、CloudWatch コンソールまたは AWS CLI を使用します。コンソールでは、メトリクスはモニタリンググラフのように表示されます。

米国西部 (オレゴン) リージョンの ARC に関する CloudWatch メトリクスは、コンソールに、または AWS CLIを使用するときに表示します。を使用する場合は AWS CLI、 パラメータ を含めて、コマンドの米国西部 (オレゴン) リージョンを指定します`--region us-west-2`。

CloudWatch コンソールを使用してメトリクスを表示するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで [Metrics (メトリクス)] を選択してください。

3. Route53RecoveryReadiness 名前空間を選択します。
4. (オプション) すべてのディメンションでメトリクスを表示するには、検索フィールドに名称を入力します。

を使用してメトリクスを表示するには AWS CLI

使用可能なメトリクスを表示するには、次の list-metrics コマンドを使用します。

```
aws cloudwatch list-metrics --namespace AWS/Route53RecoveryReadiness --region us-west-2
```

を使用してメトリクスの統計を取得するには AWS CLI

以下の get-metric-statistics コマンドを使用して、指定されたメトリクスとディメンションの統計情報を取得します。CloudWatch は、ディメンションの一意の組み合わせをそれぞれ別のメトリクスとして扱うことに注意してください。発行されていないディメンションの組み合わせを使用した統計を取得することはできません。メトリクス作成時に使用した同じディメンションを指定する必要があります。

次の例は、ARC のアカウントについて、1 分ごとに評価された準備状況チェックの合計を一覧表示したものです。

```
aws cloudwatch get-metric-statistics --namespace AWS/Route53RecoveryReadiness \
--metric-name ReadinessChecks \
--region us-west-2 \
--statistics Sum --period 60 \
--dimensions Name=State,Value=READY \
--start-time 2021-07-03T01:00:00Z --end-time 2021-07-03T01:20:00Z
```

以下は、コマンドからの出力例です。

```
{
    "Label": "ReadinessChecks",
    "Datapoints": [
        {
            "Timestamp": "2021-07-08T18:00:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:04:00Z",
            "Sum": 1.0,
```

```
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:01:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:02:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:03:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        }
    ]
}
```

を使用した準備状況チェック API コールのログ記録 AWS CloudTrail

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

は、ARC のユーザー AWS CloudTrail、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、ARC のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、ARC コンソールのコールと、ARC API オペレーションへのコードのコールが含まれます。

証跡を作成する場合は、ARC のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。

CloudTrail が収集した情報を使用して、ARC に対して行われたリクエスト、リクエスト元の IP アドレス、リクエストした人、リクエストが行われた日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail での ARC 情報

CloudTrail は、アカウントの作成 AWS アカウント 時に で有効になります。ARC でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「CloudTrail イベント履歴の操作」を参照してください。

ARC のイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように他の AWS サービスを設定できます。詳細については、次を参照してください:

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- 「CloudTrail の Amazon SNS 通知の設定」
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての ARC アクションは CloudTrail によって ログ記録されます。「Amazon Application Recovery Controller のリカバリの準備状況 API リファレンスガイド」、「Amazon Application Recovery Controller のリカバリコントロール設定 API リファレンスガイド」、「Amazon Application Recovery Controller のルーティングコントロール API リファレンスガイド」を参照してください。例えば、`CreateCluster`、`UpdateRoutingControlState`、`CreateRecoveryGroup` の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。

- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

イベント履歴での ARC イベントの表示

CloudTrail では、[イベント履歴] に最近のイベントが表示されます。ARC API リクエストのイベントを表示するには、コンソールの上部にあるリージョンセレクターで [米国西部 (オレゴン)] を指定する必要があります。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。

ARC ログファイルエントリの概要

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、準備状況チェックの `CreateRecoveryGroup` アクションを実行する CloudTrail ログエントリです。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA33L3W36EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/admin",
                "accountId": "111122223333",
                "userName": "EXAMPLENAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
```

```
                "creationDate": "2021-07-06T17:38:05Z"
            }
        }
    },
    "eventTime": "2021-07-06T18:08:03Z",
    "eventSource": "route53-recovery-readiness.amazonaws.com",
    "eventName": "CreateRecoveryGroup",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
    "requestParameters": {
        "recoveryGroupName": "MyRecoveryGroup"
    },
    "responseElements": {
        "Access-Control-Expose-Headers": "x-amzn-errortype,x-amzn-requestid,x-amzn-
errormessage,x-amzn-trace-id,x-amzn-requestid,x-amz-apigw-id,date",
        "cells": [],
        "recoveryGroupName": "MyRecoveryGroup",
        "recoveryGroupArn": "arn:aws:route53-recovery-readiness::111122223333:recovery-
group/MyRecoveryGroup",
        "tags": "***"
    },
    "requestID": "fd42dcf7-6446-41e9-b408-d096example",
    "eventID": "4b5c42df-1174-46c8-be99-d67aexample",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


Amazon EventBridge により ARC で準備状況チェックを使用する

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon EventBridge を使用すると、Amazon Application Recovery Controller (ARC) の準備状況チェックリソースをモニタリングするイベント駆動型ルールを設定し、他の AWS サービスを使用するターゲットアクションを開始できます。例えば、準備状況チェックステータスが READY から NOT READY に変わるとき、Amazon SNS トピックに信号を伝達することにより、E メール通知を送信させるルールを設定できます。

> **Note**
>
> ARC は、米国西部 (オレゴン) (us-west-2) AWS リージョンでのみ、準備状況チェックのために EventBridge イベントを発行します。準備状況チェックの EventBridge イベントを受信するには、米国西部 (オレゴン) リージョンに EventBridge ルールを作成します。

Amazon EventBridge でルールを作成して、次の ARC 準備状況チェックイベントで反応させることが可能です。

- 準備状況チェックの準備。このイベントは、準備状況チェックのステータスが (例えば READY から NOT READY に) 変わった場合に指定します。

関心のある特定の ARC イベントをキャプチャするには、そのイベント固有のパターンを定義し、イベントを検出する EventBridge がそれを照合できるようにします。イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

イベントは、ベストエフォートベースで出力されます。通常の運用状況下では、ARC から EventBridge にほぼリアルタイムで配信されます。ただし、イベントの配信を遅らせたり妨げたりする状況が発生する場合もあります。

EventBridge ルールがイベントパターンでどのように機能するかについては、「EventBridge のイベントとイベントパターン」を参照してください。

EventBridge で準備状況チェックリソースをモニタリングする

EventBridge でルールを作成すると、ARC が準備状況チェックリソースに対してイベントを発生したときに実行されるアクションを定義できます。

EventBridge コンソールにイベントパターンを入力またはコピーアンドペーストするには、コンソールの [独自のものを入力] オプションを選択します。役に立つ可能性のあるイベントパターンを判断するために、このトピックには準備状況イベントパターンの例が含まれています。

リソースイベントのルールを作成するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. AWS リージョン でルールを作成するには、米国西部 (オレゴン) を選択します。これは準備状況イベントに必要なリージョンです。
3. [Create rule] を選択します。
4. ルールの [Name (名前)] を入力し、必要に応じて説明を入力します。
5. [イベントバス] については、デフォルト値の [デフォルト] のままにします。
6. [次へ] を選択します。
7. [イベントパターンを構築] ステップでは、[イベントソース] はデフォルト値の [AWS イベント] のままにします。
8. [サンプルイベント] で [独自のサンプルイベントを入力] を選択します。
9. [サンプルイベント] には、イベントパターンを入力するか、コピーして貼り付けます。例については、次のセクションを参照ください。

準備状況イベントパターンの例

イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

このセクションのイベントパターンをコピーして EventBridge に貼り付け、ARC のアクションとリソースの監視に使用できるルールを作成できます。

以下のイベントパターンは、EventBridge で ARC の準備状況チェック機能に使用できる例を示しています。

- ARC の準備状況チェックからすべてのイベントを選択します。

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ]
}
```

- セルに関連するイベントのみを選択します。

```
{
```

```
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ]
}
```

- MyExampleCell という特定のセルに関連するイベントのみを選択します。

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ],
    "resources": [
        "arn:aws:route53-recovery-readiness::111122223333:cell/MyExampleCell"
    ]
}
```

- リカバリグループ、セル、NOT READY のステータスとなった準備状況チェックのいずれかのイベントのみを選択します。

```
{
   "source":[
      "aws.route53-recovery-readiness"
   ],
   "detail-type":{
      "new-state":{
         "readiness-status":[
            "NOT_READY"
         ]
      }
   }
}
```

- リカバリグループ、セル、READY 以外のステータスになった準備状況チェックのいずれかのイベントのみを選択します。

```
{
   "source":[
```

```
      "aws.route53-recovery-readiness"
   ],
   "detail":{
      "new-state":{
         "readiness-status":[
            {
               "anything-but":"READY"
            }
         ]
      }
   }
}
```

以下は、リカバリグループの準備状況ステータスを変更するための ARC イベントの例です。

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller recovery group readiness
 status change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:recovery-group/BillingApp"
    ],
    "detail": {
        "recovery-group-name": "BillingApp",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

以下は、セルの準備状況ステータスを変更するための ARC イベントの例です。

```
{
    "version": "0",
```

```
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller cell readiness status
 change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:cell/PDXCell"
    ],
    "detail": {
        "cell-name": "PDXCell",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

以下は、準備状況チェックステータスを変更するための ARC イベントの例です。

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller readiness check status
 change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:readiness-check/
UserTableReadinessCheck"
    ],
    "detail": {
    "readiness-check-name": "UserTableReadinessCheck",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
```

```
        }
    }
}
```

ターゲットとして使用する CloudWatch ロググループを指定する

EventBridge ルールを作成するときは、ルールに一致するイベントの送信先のターゲットを指定する必要があります。EventBridge で利用できるターゲットの一覧については、「EventBridge コンソールで使用可能なターゲット」を参照してください。EventBridge ルールに追加できるターゲットの 1 つは、Amazon CloudWatch ロググループです。このセクションでは、CloudWatch ロググループをターゲットとして追加するための要件と、ルール作成時にロググループを追加する手順について説明します。

CloudWatch ロググループをターゲットとして追加するには、次のいずれかを実行します。

- 新しいロググループを作成する
- 既存のロググループを選択する

ルールの作成時にコンソールを使用して新しいロググループを指定する場合は、EventBridge によって自動的にロググループが作成されます。EventBridge ルールのターゲットとして使用するロググループが `/aws/events` で始まることを確認します。既存のロググループを選択する場合は、`/aws/events` で始まるロググループのみがドロップダウンメニューのオプションとして表示されることに注意してください。詳細については、「Amazon CloudWatch ユーザーガイド」の「新しいロググループを作成する」を参照してください。

コンソール外で CloudWatch オペレーションを使用して CloudWatch ロググループを作成またはターゲットとして使用する場合は、アクセス許可を正しく設定していることを確認してください。コンソールを使用して EventBridge ルールにロググループを追加すると、ロググループのリソースベースのポリシーが自動的に更新されます。ただし、 AWS Command Line Interface または AWS SDK を使用してロググループを指定する場合は、ロググループのリソースベースのポリシーを更新する必要があります。次のポリシー例は、ロググループのリソースベースのポリシーで定義する必要があるアクセス許可を示しています。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "events.amazonaws.com",
                    "delivery.logs.amazonaws.com"
                ]
            },
            "Resource": "arn:aws:logs:us-east-1:222222222222:log-group:/aws/
events/*:*",
            "Sid": "TrustEventsToStoreLogEvent"
        }
    ]
}
```

コンソールを使用してロググループのリソースベースのポリシーを設定することはできません。必要なアクセス許可をリソースベースのポリシーに追加するには、CloudWatch PutResourcePolicy API オペレーションを使用します。次に、describe-resource-policies CLI コマンドを使用して、ポリシーが正しく適用されたことを確認できます。

リソースイベントのルールを作成して CloudWatch ロググループターゲットを指定するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. ルール AWS リージョン を作成する を選択します。
3. [ルールの作成] を選択して、イベントパターンまたはスケジュールの詳細など、そのルールに関する情報を入力します。

   準備状況のための EventBridge ルールの作成の詳細については、「EventBridge で準備状況チェックリソースをモニタリングする」を参照してください。
4. [ターゲットを選択] ページで、ターゲットとして [CloudWatch] を選択します。
5. ドロップダウンメニューから CloudWatch ロググループを選択します。

ARC での準備状況チェックのための Identity and Access Management

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

内容

- Amazon Application Recovery Controller (ARC) で準備状況チェックが IAM と連携する仕組み
- ARC における準備状況チェックのアイデンティティベースのポリシーの例
- ARC の準備状況チェックにサービスにリンクされたロールを使用する
- AWS ARC での準備状況チェックのための マネージドポリシー

Amazon Application Recovery Controller (ARC) で準備状況チェックが IAM と連携する仕組み

IAM を使用して ARC へのアクセスを管理する前に、ARC で利用できる IAM の機能について学びます。

IAM を使用して Amazon Application Recovery Controller (ARC) の準備状況チェックへのアクセスを管理する前に、準備状況チェックで使用できる IAM 機能を確認してください。

Amazon Application Recovery Controller (ARC) で準備状況チェックにより使用できる IAM の機能

IAM 機能	準備状況チェックのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	なし

IAM 機能	準備状況チェックのサポート
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	あり
プリンシパルアクセス権限	あり
サービスロール	いいえ
サービスリンクロール	はい

AWS サービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

準備状況チェックのためのアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

ARC のアイデンティティベースのポリシーの例を表示するには、「Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例」を参照してください。

準備状況チェック内のリソースベースのポリシー

リソースベースのポリシーのサポート: なし

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。

準備状況チェックのポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

準備状況チェックの ARC アクションのリストを表示するには、「サービス認証リファレンス」の「Amazon Route 53 Recovery Readiness で定義されるアクション」を参照してください。

準備状況チェックの ARC におけるポリシーのアクションでは、アクションの前に次のプレフィックスを使用します。

```
route53-recovery-readiness
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。例えば、次のようになります。

```
"Action": [
      "route53-recovery-readiness:action1",
      "route53-recovery-readiness:action2"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、`Describe` という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "route53-recovery-readiness:Describe*"
```

準備状況チェックの ARC アイデンティティベースのポリシーの例を表示するには、「ARC における準備状況チェックのアイデンティティベースのポリシーの例」を参照してください。

準備状況チェックのためのポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Resource` JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

ゾーンシフトの ARC アクションのリストを確認するには、「Amazon Route 53 Recovery Readiness で定義されるアクション」を参照してください。

準備状況チェックの ARC アイデンティティベースのポリシーの例を表示するには、「ARC における準備状況チェックのアイデンティティベースのポリシーの例」を参照してください。

準備状況チェックのポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

準備状況チェックの ARC アクションのリストを確認するには、「Amazon Route 53 Recovery Readiness の条件キー」を参照してください

準備状況チェックにより条件キーで使用できるアクションとリソースを確認するには、「Amazon Route 53 Recovery Readiness で定義されるアクション」を参照してください。

準備状況チェックの ARC アイデンティティベースのポリシーの例を表示するには、「ARC における準備状況チェックのアイデンティティベースのポリシーの例」を参照してください。

準備状況チェックのアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

準備状況チェックによる属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: 一部

属性ベースのアクセスコントロール (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

リカバリの準備状況 (準備状況チェック) は ABAC をサポートしています。

準備状況チェックでの一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたは切り替えロールを使用する場合に自動的に作成されます。 AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

準備状況チェックのクロスサービスプリンシパル許可

転送アクセスセッション (FAS) のサポート: あり

IAM エンティティ (ユーザーまたはロール) を使用して でアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。

準備状況チェックのアクションでポリシー内の追加の依存アクションが必要かどうかを確認するには、「Amazon Route 53 Recovery Readiness」を参照してください。

準備状況チェックのサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

準備状況チェックのサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

ARC サービスにリンクされたロールの作成または管理の詳細については、「ARC の準備状況チェックにサービスにリンクされたロールを使用する」を参照してください。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つ

けます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

ARC における準備状況チェックのアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには、ARC リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認証リファレンス」の「Amazon Application Recovery Controller (ARC) のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- 例: 準備状況チェックコンソールへのアクセス
- 例: 準備状況チェックのための準備状況チェック API アクション

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが ARC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可

を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。

- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の 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。

例: 準備状況チェックコンソールへのアクセス

Amazon Application Recovery Controller (ARC) コンソールにアクセスするには、最小限のアクセス許可セットが必要です。これらのアクセス許可により、 の ARC リソースの詳細を一覧表示および表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

特定の API オペレーションのみへのアクセスを許可するときに、ユーザーとロールが引き続き準備状況チェックコンソールを使用できるようにするには、エンティティに準備状況チェック用のReadOnly AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の

「準備状況チェックのマネージドポリシーのページ」または「ユーザーへのアクセス許可の追加」を参照してください。

一部のタスクを実行するには、ユーザーは ARC の準備状況チェックに関連付けられているサービスにリンクされたロールを作成するアクセス許可を持っている必要があります。詳細についてはARC の準備状況チェックにサービスにリンクされたロールを使用するを参照してください。

コンソールから準備状況チェック機能を使用するためのフルアクセスをユーザーに付与するには、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                   "route53-recovery-readiness:CreateCell",
                   "route53-recovery-readiness:CreateCrossAccountAuthorization",
                   "route53-recovery-readiness:CreateReadinessCheck",
                   "route53-recovery-readiness:CreateRecoveryGroup",
                   "route53-recovery-readiness:CreateResourceSet",
                   "route53-recovery-readiness:DeleteCell",
                   "route53-recovery-readiness:DeleteCrossAccountAuthorization",
                   "route53-recovery-readiness:DeleteReadinessCheck",
                   "route53-recovery-readiness:DeleteRecoveryGroup",
                   "route53-recovery-readiness:DeleteResourceSet",
                   "route53-recovery-readiness:GetArchitectureRecommendations",
                   "route53-recovery-readiness:GetCell",
                   "route53-recovery-readiness:GetCellReadinessSummary",
                   "route53-recovery-readiness:GetReadinessCheck",
                   "route53-recovery-readiness:GetReadinessCheckResourceStatus",
                   "route53-recovery-readiness:GetReadinessCheckStatus",
                   "route53-recovery-readiness:GetRecoveryGroup",
                   "route53-recovery-readiness:GetRecoveryGroupReadinessSummary",
                   "route53-recovery-readiness:GetResourceSet",
                   "route53-recovery-readiness:ListCells",
                   "route53-recovery-readiness:ListCrossAccountAuthorizations",
                   "route53-recovery-readiness:ListReadinessChecks",
                   "route53-recovery-readiness:ListRecoveryGroups",
                   "route53-recovery-readiness:ListResourceSets",
```

```
                "route53-recovery-readiness:ListRules",
                "route53-recovery-readiness:UpdateCell",
                "route53-recovery-readiness:UpdateReadinessCheck",
                "route53-recovery-readiness:UpdateRecoveryGroup",
                "route53-recovery-readiness:UpdateResourceSet"
            ],
            "Resource": "*"
        }
    ]
}
```

例: 準備状況チェックのための準備状況チェック API アクション

ユーザーが ARC API アクションを使用して ARC 準備状況チェックコントロールプレーンと連携できるようにするには、例えば、リカバリグループ、リソースセット、準備状況チェックを作成するために、以下で説明するように、ユーザーが操作する必要がある API オペレーションに対応するポリシーをアタッチします。

一部のタスクを実行するには、ユーザーは ARC の準備状況チェックに関連付けられているサービスにリンクされたロールを作成するアクセス許可を持っている必要があります。詳細についてはARC の準備状況チェックにサービスにリンクされたロールを使用するを参照してください。

準備状況チェックに関する API オペレーションを使用するには、次のようなポリシーをユーザーにアタッチします。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "route53-recovery-readiness:CreateCell",
                "route53-recovery-readiness:CreateCrossAccountAuthorization",
                "route53-recovery-readiness:CreateReadinessCheck",
                "route53-recovery-readiness:CreateRecoveryGroup",
                "route53-recovery-readiness:CreateResourceSet",
                "route53-recovery-readiness:DeleteCell",
                "route53-recovery-readiness:DeleteCrossAccountAuthorization",
                "route53-recovery-readiness:DeleteReadinessCheck",
```

```
                "route53-recovery-readiness:DeleteRecoveryGroup",
                "route53-recovery-readiness:DeleteResourceSet",
                "route53-recovery-readiness:GetArchitectureRecommendations",
                "route53-recovery-readiness:GetCell",
                "route53-recovery-readiness:GetCellReadinessSummary",
                "route53-recovery-readiness:GetReadinessCheck",
                "route53-recovery-readiness:GetReadinessCheckResourceStatus",
                "route53-recovery-readiness:GetReadinessCheckStatus",
                "route53-recovery-readiness:GetRecoveryGroup",
                "route53-recovery-readiness:GetRecoveryGroupReadinessSummary",
                "route53-recovery-readiness:GetResourceSet",
                "route53-recovery-readiness:ListCells",
                "route53-recovery-readiness:ListCrossAccountAuthorizations",
                "route53-recovery-readiness:ListReadinessChecks",
                "route53-recovery-readiness:ListRecoveryGroups",
                "route53-recovery-readiness:ListResourceSets",
                "route53-recovery-readiness:ListRules",
                "route53-recovery-readiness:ListTagsForResources",
                "route53-recovery-readiness:UpdateCell",
                "route53-recovery-readiness:UpdateReadinessCheck",
                "route53-recovery-readiness:UpdateRecoveryGroup",
                "route53-recovery-readiness:UpdateResourceSet",
                "route53-recovery-readiness:TagResource",
                "route53-recovery-readiness:UntagResource"
            ],
            "Resource": "*"
        }
    ]
}
```

ARC の準備状況チェックにサービスにリンクされたロールを使用する

Amazon Application Recovery Controller は AWS Identity and Access Management 、(IAM) サービスにリンクされたロールを使用します。サービスにリンクされたロールは、サービスに直接リンクされた一意のタイプの IAM ロールです。この場合は ARC です。サービスにリンクされたロールは ARC によって事前定義されており、特定の目的でサービスがユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。

サービスにリンクされたロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、ARC の設定が簡単になります。ARC は、サービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、ARC のみがその ロールを引き受けることができます。

定義されるアクセス許可には、信頼ポリシーと許可ポリシーが含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、まずその関連リソースを削除します。これにより、リソースにアクセスするためのアクセス許可を誤って削除できないため、ARC リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスの詳細については、AWS「IAM と連携するサービス」を参照し、「サービスにリンクされたロール」列で「はい」があるサービスを探します。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、はいリンクを選択します。

ARC には、この章で説明されている以下のサービスにリンクされたロールがあります。

- ARC は RouteRoute53RecoveryReadinessServiceRolePolicyという名前のサービスにリンクされたロールを使用して、準備状況をチェックするためのリソースと設定にアクセスします。
- ARC は、オートシフト練習実行に という名前のサービスにリンクされたロールを使用して、お客様が提供する Amazon CloudWatch アラームとお客様 Health Dashboard イベントをモニタリングし、練習実行を開始します。

Route53RecoveryReadinessServiceRolePolicy のサービスリンクロールアクセス許可

ARC は RouteRoute53RecoveryReadinessServiceRolePolicyという名前のサービスリンクロールを使用して、準備状況をチェックするためのリソースと設定にアクセスします。このセクションでは、サービスリンクロールのアクセス許可と、ロールの作成、編集、および削除に関して説明します。

Route53RecoveryReadinessServiceRolePolicy のサービスリンクロールアクセス許可

このサービスリンクロールは、マネージドポリシーである `Route53RecoveryReadinessServiceRolePolicy` を使用します。

Route53RecoveryReadinessServiceRolePolicy サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `route53-recovery-readiness.amazonaws.com`

このポリシーのアクセス許可を確認するには、「 AWS マネージドポリシーリファレンス」のRoute53RecoveryReadinessServiceRolePolicy」を参照してください。

サービスリンク役割の作成、編集、削除を IAM エンティティ (ユーザー、グループ、役割など) に許可するにはアクセス許可を設定する必要があります。詳細については、「IAM User Guide」(IAM

ユーザーガイド) の「Service-linked role permissions」(サービスにリンクされたロールのアクセス権限) を参照してください。

ARC の Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールの作成

Route53RecoveryReadinessServiceRolePolicy サービスリンクロールを手動で作成する必要はありません。、、または AWS API で最初の準備状況チェック AWS マネジメントコンソール AWS CLIまたはクロスアカウント認可を作成すると、ARC によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は同じ方法でアカウントにロールを再作成できます。最初の準備状況チェックまたはクロスアカウント認可を作成すると、ARC はサービスにリンクされたロールを再度作成します。

ARC の Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールの編集

ARC では、Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、他のエンティティがロールを参照する可能性があるため、ロールの名前を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」を参照してください。

ARC の Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、積極的にモニタリングまたは保守されていない未使用のエンティティを排除できます。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

準備状況チェックとクロスアカウント承認を削除した
後、Route53RecoveryReadinessServiceRolePolicy サービスリンクロールを削除できます。準備状況チェックの詳細については、「ARC の準備状況チェック」を参照してください。クロスアカウント認証の詳細については、「ARC でのクロスアカウント認証の作成」を参照してください。

> **Note**
>
> リソースを削除しようとしたときに ARC サービスがロールを使用している場合、サービスロールの削除が失敗する可能性があります。失敗した場合は、数分待ってからロールの削除をもう一度試してください。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、 AWS CLI、または AWS API を使用して、Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールを削除します。詳細については、 IAM ユーザーガイド の「サービスにリンクされたロールの削除」を参照してください。

準備状況チェックのための ARC サービスにリンクされたロールの更新

ARC サービスにリンクされたロールの AWS マネージドポリシーの更新については、ARC の AWS マネージドポリシーの更新表を参照してください。ARC ドキュメント履歴ページで自動 RSS アラートをサブスクライブすることもできます。

AWS ARC での準備状況チェックのための マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: Route53RecoveryReadinessServiceRolePolicy

IAM エンティティに `Route53RecoveryReadinessServiceRolePolicy` をアタッチすることはできません。このポリシーは、Amazon Application Recovery Controller (ARC) に、ARC が使用または管理している AWS サービスやリソースへのアクセスを許可する、サービスにリンクされたロールにアタッチされます。詳細については、「ARC の準備状況チェックにサービスにリンクされたロールを使用する」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryReadinessFullAccess

IAM エンティティに AmazonRoute53RecoveryReadinessFullAccess をアタッチできます。このポリシーは、ARC のリカバリの準備状況 (準備状況チェック) を操作するアクションへの、フルアクセスを許可します。これを、リカバリの準備状況へのフルアクセスを必要とする IAM ユーザーとその他のプリンシパルにアタッチします。

このポリシーの許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryReadinessFullAccess」を参照してください。

AWS マネージドポリシー: AmazonRoute53RecoveryReadinessReadOnlyAccess

IAM エンティティに AmazonRoute53RecoveryReadinessReadOnlyAccess をアタッチできます。このポリシーは、ARC のリカバリの準備状況を操作するアクションへの読み取り専用アクセスを許可します。これは、準備状況のステータスとリカバリグループの設定を確認する必要があるユーザーに役立つポリシーです。これらのユーザーは、リソースを作成、更新、削除できません。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonRoute53RecoveryReadinessReadOnlyAccess」を参照してください。

準備のための AWS マネージドポリシーの更新

このサービスがこれらの変更の追跡を開始してからの ARC での準備状況チェックのための AWS マネージドポリシーの更新の詳細については、「」を参照してくださいAmazon Application Recovery Controller (ARC) の AWS マネージドポリシーの更新。このページの変更に関する自動通知については、ARC の「ドキュメント履歴ページ」で RSS フィードをサブスクライブしてください。

準備状況チェックのクォータ

> **Note**
>
> Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。

Amazon Application Recovery Controller (ARC) の準備状況チェックには、次のクォータ (旧称「制限」) が適用されます。

エンティティ	クォータ
アカウントあたりのリカバリグループの数	5
アカウントあたりのセルの数	15
セルあたりのネストされたセルの数	3
リカバリグループあたりのセルの数	3
セルあたりのリソースの数	10
リカバリグループあたりのリソースの数	10
リソースセットあたりのリソースの数	6
アカウントあたりのリソースセットの数	200
アカウントあたりの準備状況チェックの数	200
クロスアカウント認証の数	100

ARC でのリージョン切り替え

ARC のリージョンスイッチを使用して、 AWS アカウント間でアプリケーションリソースの大規模で複雑な復旧タスクを調整し、ビジネス継続性を確保し、運用オーバーヘッドを削減できます。リージョン切り替えは、手動で実行することも、Amazon CloudWatch アラームトリガーを使用して自動化することもできます。に障害 AWS リージョン が発生した場合は、リージョンスイッチを使用してリソースをフェイルオーバーするか、別のリージョンに切り替えることで、作成したプランを実行できます。これにより、アプリケーションは正常な AWS リージョンで引き続き稼働できます。

リージョン切り替えは、特定のリカバリのニーズに合わせて設計および構成するプランのコンセプトを中心に構成されています。各プランには、ステップで構成されるワークフローが含まれています。各ステップは 1 つ以上の実行ブロックを実行します。この実行ブロックは、リージョンスイッチが並列または順番に実行され、アプリケーション復旧を完了します。各実行ブロックは、アプリケーションのリソースの切り替えやトラフィックリダイレクトの管理など、さまざまなタスクを処理します。さらに柔軟性を高めるために、親プラン全体に子プランを追加することで、親プランを作成できます。

リージョン切り替えには以下が含まれます。

- アクティブ/パッシブ設定およびアクティブ/アクティブ設定のサポート。アクティブ/パッシブマルチリージョン設定がある場合はフェイルオーバーとフェイルバックを、複数のリージョンでアプリケーションがアクティブ/アクティブとして設定されている場合はシフトアウェイとリターンを行うことができます。
- アプリケーションリカバリに含めるアプリケーションリソースのクロスアカウントサポート。アカウント間でリージョン切り替えプランを共有することもできます。
- Amazon CloudWatch アラームに基づいてプラン実行をトリガーすることで、自動フェイルオーバーまたはスイッチオーバーを行います。または、リージョン切り替えプランを手動で実行できます。
- リカバリプロセスをリアルタイムで可視化するフル機能のダッシュボード。
- 各 のデータプレーン AWS リージョン。非アクティブ化するリージョンに依存することなく、リージョン切り替えプランを実行できます。

リージョン切り替えは AWSによって完全に管理されます。リージョン切り替えを使用すると、スクリプトを構築および保守したり、復旧に関するデータを手動で収集したりするのではなく、アプリケーション固有の要件に焦点を当てた復旧プラットフォームの耐障害性を活用できます。

リージョンの切り替えについて

リージョンスイッチを使用すると、マルチリージョンアプリケーションが実行され AWS リージョンている を切り替える特定のステップをオーケストレーションできます。

リージョン切り替えは、特定のリカバリのニーズに合わせて設計および構成するプランのコンセプトを中心に構成されています。各プランには、ステップで構成されるワークフローが含まれています。各ステップは、1 つ以上の実行ブロックを実行します。この実行ブロックは、リージョンスイッチを並列または順番に実行して、アプリケーションの復旧を完了します。各実行ブロックは、アプリケーションのリソースの切り替えやトラフィックリダイレクトの管理など、さまざまなタスクを処理します。子プランを追加することで、親プランをより柔軟に作成できます。

プランを作成または更新するたびに、リージョン切り替えはプラン評価を行い、IAM 権限、リソース構成、または実行キャパシティに問題がないことを確認します。リージョン切り替えは、これらの評価を定期的に実行し、問題が見つかると警告を生成します。

リージョン切り替えでは、プランが実行されるたびに実際のリカバリ時間の値も計算されるため、プランが目標を達成しているかどうかを評価するのに役立ちます。リカバリ時間やプランの実行に関す

るその他の詳細は、 AWS マネジメントコンソールのリージョン切り替えダッシュボードで確認できます。詳細については、「リージョン切り替えダッシュボード」を参照してください。

リージョン切り替えの各領域の詳細については、以下のセクションを参照してください。

リージョン切り替えプラン

リージョン切り替えプランは、リージョンの切り替えにおける最上位リソースです。プランの範囲は、特定のマルチリージョンアプリケーションに限定する必要があります。プランを使用すると、AWS リージョン 指定した で、アプリケーションとそのクロスアカウントリソースを含むリソースをアクティブ化または非アクティブ化する一連のリージョンスイッチ実行ブロックを実行して、アプリケーションを復旧するワークフローを構築できます。

プランは 1 つ以上のワークフローで構成され、特定のワークフローを有効または無効にできます AWS リージョン。ワークフローの実行ブロックを順番に実行するように設定することも、一部のブロックを並行して実行するように指定することもできます。

アクティブ/パッシブマルチリージョンアプローチ用に設定したプランでは、リージョンのアクティブ化に使用できるワークフローを 1 つ作成するか、リージョンごとに 1 つずつ、2 つの異なるアクティベーションワークフローを作成します。アクティブ/アクティブアプローチ用に設定したプランでは、1 つのワークフローを作成してリージョンをアクティブ化し、もう 1 つのワークフローを作成してリージョンを非アクティブ化します。

AWS リージョン は、 がデータセンターを AWS クラスター化する世界中の地理的な場所です。各リージョンは、他のリージョンと完全に分離されるように設計されており、耐障害性と安定性を実現します。リージョン切り替えを使用する場合は、アプリケーションがデプロイされているリージョンと、復旧に使用するリージョンを考慮する必要があります。

リージョンスイッチは、サービス AWS リージョン が利用可能な任意の 2 つの間の復旧をサポートします。リージョン切り替えプランを設定するときは、アプリケーションがデプロイされるリージョンと、アクティブ/パッシブまたはアクティブ/アクティブのリカバリアプローチを指定します。

例えば、us-east-1 をプライマリリージョンとし、us-west-2 をスタンバイリージョンとして、アクティブ/パッシブマルチリージョンアプローチを使用することができます。us-east-1 のアプリケーションに影響する運用上の問題からアプリケーションを復旧するには、リージョン切り替えプランを実行して us-west-2 をアクティブ化できます。これにより、アプリケーションは us-east-1 のリソースから us-west-2 のリソースに切り替わります。

リージョン切り替えプランは、プランの作成時に指定した IAM ロールに関連付けられたアクセス許可を使用して実行されます。

マルチリージョンアプリケーションごとに 1 つずつ複数のプランを作成し、親プランを作成して、これらのプラン間で必要な順序で復旧をオーケストレーションできます。親プランは、リージョン切り替えプランの実行ブロックをステップとして使用するプランです。プランの階層は 2 つのレベル (親と子) に制限されますが、同じ親プランの下に複数の子プランを含めることができます。

ワークフローと実行ブロック

リージョン切り替えプランを作成したら、1 つ以上のワークフローをプランに追加して、アプリケーション復旧のためにプランで実行するステップを定義する必要があります。ワークフローごとに、実行ブロックを含むステップを追加します。各実行ブロックは、リソースをスケールアップしたり、ルーティングコントロールを更新してトラフィックを再ルーティングしたりするなど、特定の復旧アクションを実行します。ステップは、これらの実行ブロックを整理し、並行して実行するか順番に実行するかを制御します。親プランを作成することで、複数のアプリケーションがアクティブ化するリージョンに復旧する順序を調整することもできます。

実行ブロックをワークフロー内のステップに整理します。各ステップには、並行して実行される 1 つ以上の実行ブロックを含めることができ、ワークフローで順番に実行するステップを整理します。また、リソースに応じて、実行ブロックを正常な (計画的な) 実行として、または不正な (計画外の) 実行として実行するオプションがあります。

- 正常な実行: 計画された実行ワークフロー。環境が正常であれば、正常なワークフローを使用して、秩序あるプラン実行のステップをすべて実行できます。
- 不正な実行: 予期しない実行。不正なワークフローモードは、必要なステップとアクションのみを使用します。このモードは、ワークフローの実行ブロックの動作を変更するか、特定の実行ブロックをスキップします。
- 復旧後実行: 将来のリージョンイベントに備えるために復旧が成功した後に実行されるワークフロー。復旧後の実行では、リードレプリカの作成、Lambda 関数を介したカスタムロジックの実行、手動承認ゲートの追加、複雑なオーケストレーションのための子プランの埋め込みを行うことができます。これらの実行では、両方のリージョンが正常であり、以前に障害が発生したリージョンで実行されている必要があります。

最後に、実行ブロックのクロスアカウントリソースを設定することもできます。まず、「リージョン切り替えでのクロスアカウントサポート」のガイダンスに従ってアクセス許可を設定する必要があります。必要な IAM ロールを設定したら、プランワークフローの実行ブロックにクロスアカウントリソースを追加できます。クロスアカウントリソースを追加するには、ステップを追加するときに、他のリソースへのアクセス許可を持つターゲット IAM ロールを指定します AWS アカウント。また、クロスアカウントロールの信頼ポリシーで指定した外部 ID も指定する必要があります。必要な IAM

ロールの作成の詳細については、「クロスアカウントリソースのアクセス許可」を参照してください。

ワークフローの詳細については、「リージョンの切り替えプランワークフローの作成」を参照してください。設定ステップ、仕組み、プラン評価の一部として評価される内容など、各タイプの実行ブロックの詳細については、「実行ブロックを追加する」を参照してください。

プラン評価

プラン評価は、プランの作成時または更新時にリージョン切り替えが実行され、その後、定常状態で 30 分ごとに実行される自動プロセスです。評価プロセスは、プラン設定とリソース設定のいくつかの重要な側面を検証します。評価には、IAM アクセス許可、リソース設定、および、実行するキャパシティの検証が含まれます。

リージョン切り替えは、プランの実行の成功を妨げる可能性のある問題を検出すると、プラン評価警告を生成します。警告は、コンソールのプラン詳細ページで強調表示されます。Amazon EventBridge でプラン評価警告を使用することも、リージョン切り替え API を使用して警告を表示することもできます。Plan Evaluation API の詳細については、「Amazon Application Recovery Controller (ARC) のリージョンスイッチ API リファレンスガイド」のGetPlanEvaluationStatus」を参照してください。

評価対象のプランに関する問題の詳細と推奨される修復方法は、プラン詳細ページの[プラン評価]タブで確認できます。また、リージョン切り替えプランを実行してアプリケーションの復旧をテストすることと、リージョン切り替えプランの評価だけに頼らず、復旧プランが期待どおりに機能するかどうかをテストすることをお勧めします。

自動計画実行レポート

リージョンスイッチは、計画実行の包括的な PDF レポートを自動的に生成し、規制コンプライアンス要件を満たすのに役立ちます。これらのレポートは、詳細な実行タイムライン、計画設定、リソースの状態など、ディザスタリカバリテストと実際の復旧イベントの証拠を提供します。

プランの自動レポート生成を設定すると、リージョンスイッチは、各プランの実行が完了した後に PDF レポートを作成し、指定した Amazon S3 バケットに配信します。レポートは通常、実行完了後 30 分以内に利用できます。S3 ストレージコストが適用されます。

各レポートには以下が含まれます。

- サービスの概要とレポート作成日を含むエグゼクティブサマリー
- 実行時に存在していた設定の詳細を計画する

- ステップ、影響を受けるリソース、ステータスを含む詳細な実行タイムライン
- 実行の開始時に存在していた警告を計画する
- 関連するアラームの Amazon CloudWatch アラーム状態とアラーム履歴
- 親プランの場合、子プランの設定と実行の詳細
- 用語と概念の用語集

自動レポート生成を有効にするには、プランを作成または更新するときにレポート出力先を設定します。また、プランの実行 IAM ロールにAmazon S3バケットにレポートを書き込み、レポートコンテンツの生成に必要なリソースにアクセスするために必要なアクセス許可があることを確認する必要があります。必要なアクセス許可の詳細については、「自動計画実行レポートのアクセス許可」を参照してください。

レポート生成のステータスを表示し、完了したレポートを コンソールの計画実行の詳細ページからダウンロードできます。レポート生成で、アクセス許可の不足や Amazon S3 バケットの設定ミスなどのエラーが発生した場合、リージョンスイッチは問題のトラブルシューティングに役立つエラーの詳細を提供します。

計画評価では、実行ロールに必要な IAM アクセス許可があることを確認するなど、レポート設定を継続的に検証します。リージョンスイッチは、レポート生成の成功を妨げる設定の問題を検出すると、計画の詳細ページで表示できる警告を生成します。

リージョンアラームと実際の復旧時間

リージョン切り替えは、プランの実行ごとに実際の復旧時間値を計算します。これはプラン実行後に表示できます。実際の復旧時間はプラン実行の詳細ページに表示されるため、実際の時間を、プランの作成時に指定した目標復旧時間と比較できます。

実際の復旧時間は、プランの実行が完了するまでにかかる時間と、設定した特定の Amazon CloudWatch アラームが緑色の状態に戻るまでに経過する追加時間の合計として計算されます。

計画実行の正確な実際の復旧時間の計算をサポートするには、各リージョンのアプリケーションの状態に関するシグナルを提供するリージョン切り替えプランのリージョン Amazon CloudWatch アラームを設定する必要があります。プランが実行されると、リージョン切り替えはこれらのアプリケーションヘルスアラームを使用して、アプリケーションがいつ正常に戻ったかを判断します。次に、リージョンスイッチは、設定したアプリケーションのヘルスアラームに基づいて、アプリケーションが正常に戻るのにかかる時間に追加した計画の実行にかかる時間に基づいて実際の復旧時間を計算します。

リージョン切り替えプランに CloudWatch アラームを追加する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「アプリケーションヘルス許可の CloudWatch アラーム」を参照してください。

AWS リージョン

リージョン切り替えは、 AWS GovCloud (米国) リージョンだけでなく AWS リージョン、すべての商用リージョンで利用できます。

Amazon Application Recovery Controller (ARC) のリージョンサポートとサービスエンドポイントに関する詳細は、「Amazon Web Services 全般のリファレンス」にある「Amazon Application Recovery Controller (ARC) エンドポイントとクォータ」を参照してください。

リージョン名	リージョン	エンドポイント	プロトコル
米国東部 (オハイオ)	us-east-2	arc-region-switch.us-east-2.api.aws arc-region-switch-fips.us-east-2.api.aws	HTTPS HTTPS
米国東部 (バージニア北部)	us-east-1	arc-region-switch.us-east-1.api.aws arc-region-switch-control-plane-fips.us-east-1.api.aws arc-region-switch-fips.us-east-1.api.aws arc-region-switch-control-plane.us-east-1.api.aws	HTTPS HTTPS HTTPS HTTPS
米国西部 (北カリフォルニア)	us-west-1	arc-region-switch.us-west-1.api.aws arc-region-switch-fips.us-west-1.api.aws	HTTPS HTTPS
米国西部 (オレゴン)	us-west-2	arc-region-switch.us-west-2.api.aws arc-region-switch-fips.us-west-2.api.aws	HTTPS HTTPS

リージョン名	リージョン	エンドポイント	プロトコル	
アフリカ (ケープタウン)	af-south-1	arc-region-switch.af-south-1.api.aws	HTTPS	
アジアパシフィック (香港)	ap-east-1	arc-region-switch.ap-east-1.api.aws	HTTPS	
アジアパシフィック (ハイデラバード)	ap-south-2	arc-region-switch.ap-south-2.api.aws	HTTPS	
アジアパシフィック (ジャカルタ)	ap-southeast-3	arc-region-switch.ap-southeast-3.api.aws	HTTPS	
アジアパシフィック (マレーシア)	ap-southeast-5	arc-region-switch.ap-southeast-5.api.aws	HTTPS	
アジアパシフィック (メルボルン)	ap-southeast-4	arc-region-switch.ap-southeast-4.api.aws	HTTPS	
アジアパシフィック (ムンバイ)	ap-south-1	arc-region-switch.ap-south-1.api.aws	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
アジアパシフィック (ニュージーランド)	ap-southeast-6	arc-region-switch.ap-southeast-6.api.aws	HTTPS	
アジアパシフィック (大阪)	ap-northeast-3	arc-region-switch.ap-northeast-3.api.aws	HTTPS	
アジアパシフィック (ソウル)	ap-northeast-2	arc-region-switch.ap-northeast-2.api.aws	HTTPS	
アジアパシフィック (シンガポール)	ap-southeast-1	arc-region-switch.ap-southeast-1.api.aws	HTTPS	
アジアパシフィック (シドニー)	ap-southeast-2	arc-region-switch.ap-southeast-2.api.aws	HTTPS	
アジアパシフィック (台北)	ap-east-2	arc-region-switch.ap-east-2.api.aws	HTTPS	
アジアパシフィック (タイ)	ap-southeast-7	arc-region-switch.ap-southeast-7.api.aws	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
アジアパシフィック (東京)	ap-northeast-1	arc-region-switch.ap-northeast-1.api.aws	HTTPS	
カナダ (中部)	ca-central-1	arc-region-switch.ca-central-1.api.aws	HTTPS	
カナダ西部 (カルガリー)	ca-west-1	arc-region-switch.ca-west-1.api.aws	HTTPS	
欧州 (フランクフルト)	eu-central-1	arc-region-switch.eu-central-1.api.aws	HTTPS	
欧州 (アイルランド)	eu-west-1	arc-region-switch.eu-west-1.api.aws	HTTPS	
欧州 (ロンドン)	eu-west-2	arc-region-switch.eu-west-2.api.aws	HTTPS	
ヨーロッパ (ミラノ)	eu-south-1	arc-region-switch.eu-south-1.api.aws	HTTPS	
欧州 (パリ)	eu-west-3	arc-region-switch.eu-west-3.api.aws	HTTPS	
欧州 (スペイン)	eu-south-2	arc-region-switch.eu-south-2.api.aws	HTTPS	
欧州 (ストックホルム)	eu-north-1	arc-region-switch.eu-north-1.api.aws	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル
欧州 (チューリッヒ)	eu-central-2	arc-region-switch.eu-central-2.api.aws	HTTPS
イスラエル (テルアビブ)	il-central-1	arc-region-switch.il-central-1.api.aws	HTTPS
メキシコ (中部)	mx-central-1	arc-region-switch.mx-central-1.api.aws	HTTPS
中東 (バーレーン)	me-south-1	arc-region-switch.me-south-1.api.aws	HTTPS
中東 (アラブ首長国連邦)	me-central-1	arc-region-switch.me-central-1.api.aws	HTTPS
南米 (サンパウロ)	sa-east-1	arc-region-switch.sa-east-1.api.aws	HTTPS
AWS GovCloud (米国東部)	us-gov-east-1	arc-region-switch.us-gov-east-1.api.aws arc-region-switch-fips.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (米国西部)	us-gov-west-1	arc-region-switch.us-gov-west-1.api.aws arc-region-switch-control-plane-fips.us-gov-west-1.api.aws arc-region-switch-fips.us-gov-west-1.api.aws arc-region-switch-control-plane.us-gov-west-1.api.aws	HTTPS HTTPS HTTPS HTTPS

リージョン切り替えコンポーネント

以下は、Amazon Application Recovery Controller (ARC) のリージョン切り替え機能のコンポーネントと概念です。

プラン

プランは、アプリケーションの基本的な復旧プロセスです。プランを作成するには、実行ブロックを使用して 1 つ以上のワークフローを構築し、順番に、または並行して実行します。次に、リージョンの障害が発生した場合、アプリケーションを正常なリージョンで実行するようシフトすることにより、アプリケーションの復旧を完了するためのプランを実行します。

子プラン

子プランは、より複雑なアプリケーション復旧シナリオを調整するために親プラン内から実行できる自己完結型プランです。リージョン切り替えプランは 1 つのレベルでネストできます。

ワークフロー

リージョン切り替えプランには、1 つ以上のワークフローが含まれます。ワークフローは、実行ブロックを含むステップで構成されます。実行ブロックは、リカバリプランの一部としてリージョンのアクティブ化または非アクティブ化を完了するために並列または順番に実行するように指定します。アクティブ/パッシブアプローチを使用するよう設定したプランでは、リージョンのいずれかをアクティブ化するために使用できるワークフローを 1 つ作成するか、リージョンごとに 1 つずつ個別のアクティベーションワークフローを作成します。アクティブ/アクティブアプローチ用に設定したプランでは、1 つのワークフローを作成してリージョンをアクティブ化し、もう 1 つのワークフローを作成してリージョンを非アクティブ化します。

実行ブロック

実行ブロックを含むリージョン切り替え計画ワークフローにステップを追加します。実行ブロックを使用すると、アクティブ化するリージョンへの複数のアプリケーションまたはリソースの復旧を指定できます。ワークフローにステップを追加するときは、他のステップと並行して追加することも、1 つ以上の他のステップと並行して追加することもできます。

正常な設定および不正な設定

正常な (計画的) または不正な (計画的でない) 実行で、特定の実行ブロックを実行するよう選択できます。環境が正常であれば、正常なワークフローを使用して、秩序あるプラン実行のステップをすべて実行できます。不正なワークフローモードは、必要なステップとアクションのみを使用します。不正モードでプランを実行すると、プランはワークフロー内の実行ブロックの動作を変更するか、実行ブロックのタイプに応じて特定の実行ブロックをスキップします。

特定のタイプの実行ブロックは、正しく実行されない場合には異なる動作で実行されます。これらの違いの詳細については、各タイプの実行ブロックの詳細が記載されているセクションで説明されています。詳細については、「実行ブロックを追加する」を参照してください。

アクティブ/アクティブ設定およびアクティブ/パッシブ設定

複数のリージョンにまたがるアプリケーションの回復力のある設定を作成するには、主にアクティブ/パッシブとアクティブ/アクティブの 2 つのアプローチがあります。リージョン切り替えは、これらの両方のアプローチでアプリケーション復旧をサポートします。

アクティブ/パッシブ設定では、アプリケーションの 2 つのレプリカを 2 つの異なるリージョンにデプロイします。カスタマートラフィックは 1 つのリージョンにのみ送信されます。

アクティブ/アクティブ設定では、2 つの異なるリージョンに 2 つのレプリカをデプロイしますが、両方のレプリカが作業を処理しているか、トラフィックを受信しています。

プラン実行

リージョン切り替えプランを実行すると、アプリケーションと受信するトラフィックに対して正常なリージョンをアクティブ化することにより、リージョンに障害が発生したときのアプリケーション復旧を実装します。アクティブ/アクティブ設定では、プラン実行を実行して、障害のあるリージョンを非アクティブ化することもできます。

アプリケーションヘルスアラーム

アプリケーションヘルスアラームは、各リージョンのアプリケーションの状態を示すためにプランに指定する CloudWatch アラームです。リージョンを切り替えると、アプリケーションヘルスアラームを使用して、リージョンを切り替えてリカバリを実装した後の実際のリカバリ時間を判断できます。

トリガ

リージョン切り替えでトリガーを使用して、アプリケーションの復旧を自動化できます。トリガーを作成するときは、1 つ以上の Amazon CloudWatch アラームを指定し、どのアラーム条件 (「赤」や「緑」など) が計画実行を開始するかを定義します。指定された条件が満たされると、リージョンスイッチは自動的にプランを実行します。トリガーはアプリケーションのヘルスアラームとは異なります。トリガーは計画の実行を開始しますが、アプリケーションのヘルスアラームは、計画の完了後にリージョンが実際の復旧時間を計算するのに役立ちます。

復旧後のワークフロー

復旧後ワークフローは、将来のリージョンイベントに備えて復旧が成功した後に実行されるオプションのワークフローです。これらのワークフローでは、両方のリージョンが正常であり、以前

に障害が発生したリージョンで実行されている必要があります。復旧後の実行は、最新の復旧実行の復旧実行 ID を参照します。

復旧後のワークフローは、次の実行ブロックをサポートします。

- RDS クロスリージョンレプリカの作成
- カスタムアクション Lambda
- 手動承認
- リージョン切り替えプラン

ダッシュボード

リージョン切り替えには、プランの実行に関する詳細をリアルタイムで追跡できるダッシュボードが組み込まれています。

リージョン切り替えのデータプレーンとコントロールプレーン

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは、可用性が高く、災害シナリオで必要なときに使用できるようにすることをお勧めします。信頼性と耐障害性を最大限に高めるため、可能であれば通常、データプレーン機能をメカニズムに使用する必要があります。そのことを念頭に置いて、サービス機能がコントロールプレーンとデータプレーンにどのように分けられているのか、また、サービスのデータプレーンで非常に高い信頼性が期待できるのはどのような場合なのかを理解することが重要です。

多くの AWS サービスと同様に、リージョンスイッチ機能の機能は、コントロールプレーンとデータプレーンでサポートされています。どちらのタイプも信頼できるように構築されていますが、コントロールプレーンはデータ整合性のために最適化され、データプレーンは可用性のために最適化されています。データプレーンは、コントロールプレーンが使用できなくなるような破壊的なイベントでも、可用性を維持できるように設計されています。

一般に、コントロールプレーンを使用すると、サービス内のリソースの作成、更新、削除などの基本的な管理機能を実行できます。データプレーンはサービスのコア機能を提供します。このため、停止中にリージョン切り替えプランに関する情報を取得する必要がある場合など、可用性が重要な場合はデータプレーンオペレーションを使用することをお勧めします。

リージョン切り替えでは、コントロールプレーンとデータプレーンは次のように分けられます。

- リージョンスイッチのコントロールプレーンは、米国東部 (バージニア北部) リージョン (us-east-1)、 AWS GovCloud (米国西部) リージョン (us-gov-west-1) にあり、サービス管理、つま

り復旧ではなく計画の作成と更新、つまり計画の実行にのみ使用されることを目的としています。リージョン切り替え設定のコントロールプレーン API オペレーションは、可用性が高くありません。

- リージョン切り替えには、各 AWS リージョンに独立したデータプレーンがあります。データプレーンは、復旧アクション、つまりリージョン切り替えプランの実行に使用します。データプレーンオペレーションのリストについては、「」を参照してくださいリージョン切り替え API オペレーション。これらのリージョン切り替えデータプレーンオペレーションには、高い可用性があります。

リージョンスイッチは AWS リージョン、復旧タスクのデータプレーン API オペレーションを呼び出す独立したコンソールを各 で提供するため、アクティブ化するリージョンでコンソールを使用してアプリケーション復旧の計画を実行できます。リージョン切り替えで復旧オペレーションを準備して完了する際に考慮すべき重要事項の詳細は、「ARC でのリージョン切り替えのベストプラクティス」を参照してください。

データプレーン、コントロールプレーン、および が高可用性目標を達成するためのサービス AWS を構築する方法の詳細については、Amazon Builders' Library の「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。

ARC リージョン切り替えのタグ付け

タグは、 AWS リソースを識別して整理するために使用する単語またはフレーズ (メタデータ) です。各リソースに複数のタグを追加できます。各タグには、定義したキーと値が含まれます。例えば、キーを環境、値を本番とできます。追加したタグに基づいて、リソースを検索したりフィルタ処理したりできます。

ARC のリージョン切り替えでは以下のリソースにタグを付けることができます。

- プラン

ARC でのタグ付けは、API を使用してのみ可能です。例えば、 AWS CLIを使用します。

以下は、 AWS CLIを使用したリージョン切り替えでのタグ付けの例です。

```
aws arc-region-switch --region us-east-1 create-plan --plan-name example-plan --tags Region=IAD,Stage=Prod
```

詳細については、Amazon Application Recovery Controller (ARC) の「リージョン切り替え API リファレンスガイド」の「TagResource」を参照してください。

ARC でのリージョン切り替えの料金

設定したリージョン切り替えプランごとに固定月額コストを支払います。

ARC の料金および料金例の詳細については、「ARC の料金」を参照してください。

ARC でのリージョン切り替えのベストプラクティス

Amazon Application Recovery Controller (ARC) でのリージョン切り替えによる復旧とフェイルオーバーの準備については、以下のベストプラクティスに従うことをお勧めします。

トピック

- 専用で存続期間の長い AWS 認証情報を安全かつ常にアクセス可能に保つ
- フェイルオーバーに関連する DNS レコードの低い TTL 値を選択する
- 重要なアプリケーションに必要なキャパシティを予約する
- 非常に信頼性の高いデータプレーン API オペレーションを使用して、リージョン切り替えプランに関する情報を一覧表示および取得する
- ARC でのフェイルオーバーのテスト

専用で存続期間の長い AWS 認証情報を安全かつ常にアクセス可能に保つ

ディザスタリカバリ (DR) シナリオでは、復旧タスクにアクセスして AWS 実行するための簡単なアプローチを使用して、システムの依存関係を最小限に抑えます。DR タスク用に IAM の長期間有効な認証情報を作成し、オンプレミスの物理的な金庫または仮想ボールトにこれを保管して、必要に応じてアクセスできるようにします。IAM を使用すると、アクセスキーなどのセキュリティ認証情報や、 AWS リソースへのアクセス許可を一元管理できます。DR 以外のタスクについては、AWS Single Sign-On など、 AWS サービスを使ったフェデレーションアクセスを引き続き使用することが推奨されます。

フェイルオーバーに関連する DNS レコードの低い TTL 値を選択する

フェイルオーバーの一環として変更する必要がある DNS レコード、特にヘルスチェックの対象となるレコードは、TTL 値を低く設定しておくのが適切です。このシナリオでは、TTL を 60 秒または 120 秒に設定するのが一般的です。

DNS TTL (有効期間) の設定は、新しいレコードをリクエストするまでに、どの程度の期間、レコードをキャッシュすべきかを DNS リゾルバーに伝えます。TTL を選択する際は、レイテンシーと信頼性の間、また、変化への反応との間でいずれかを優先しなくてはなりません。レコー

ドの TTL を短くすると、DNS リゾルバーはレコードの更新をより頻繁に通知します。TTL から、クエリを頻繁に実行するように指示されるためです。

詳細については、「Amazon Route 53 DNS のベストプラクティス」の「DNS レコードの TTL 値の選択」を参照してください。

重要なアプリケーションに必要なキャパシティを予約する

リージョン切り替えには、復旧の一環としてコンピューティングリソースをスケーリングするのに役立つ実行ブロックタイプが含まれています。プランでこれらの実行ブロックを使用する場合、リージョン切り替えによって必要なコンピューティング能力が達成されることは保証されていません。重要なアプリケーションがあり、キャパシティへのアクセスを保証する必要がある場合は、キャパシティを予約することをお勧めします。

セカンダリリージョンでコンピューティングキャパシティを予約しながら、コストを制限するための戦略があります。詳細については、「リザーブドキャパシティのパイロットライト: オンデマンドキャパシティ予約を使用して DR コストを最適化する方法」を参照してください。

非常に信頼性の高いデータプレーン API オペレーションを使用して、リージョン切り替えプランに関する情報を一覧表示および取得する

データプレーン API オペレーションを使用して、イベント中にリージョン切り替えプランを操作して実行します。リージョン切り替えデータプレーンオペレーションのリストについては、「リージョン切り替え API オペレーション」を参照してください。

各リージョンのリージョン切り替えコンソールでは、リージョン切り替えプランを実行するためにデータプレーンオペレーションが使用されます。データプレーン API オペレーションを呼び出すには、 を使用する AWS CLI か、いずれかの AWS SDKs を使用して記述したコードを実行します。ARC では、データプレーンの API によって、きわめて高い信頼性を実現します。

ARC でアプリケーションの復旧をテストする

ARC リージョンスイッチを使用してアプリケーション復旧を定期的にテストし、別の でセカンダリアプリケーションスタックをアクティブ化するか AWS リージョン、リージョンスイッチプランを実行していずれかのリージョンを非アクティブ化することで、アクティブ/アクティブ設定を切り替えます。

作成したリージョン切り替えがスタック内の正常なリソースと一致していること、および、すべてが予定どおりに機能していることを確認することが重要です。環境のリージョン切り替えを設定した後、これをテストし、定期的にテストして、復旧プロセスが正しく機能することを検証する必要があります。ユーザーのダウンタイムを回避するために、障害が発生する前にこのテストを定期的に実行してください。

ARC リージョンスイッチの DNS フェイルオーバーと Route 53 Accelerated Recovery

> 高速復旧では、この機能が有効になっているパブリックホストゾーンレコードを更新するために使用される APIs のターゲット RTO が 60 分になります。RTO の制御を維持し、必要な APIs の復旧 AWS が完了するまで待機しない場合は、ARC ルーティングコントロールまたは ARC リージョンスイッチ Route 53 ヘルスチェック実行ブロックを使用する必要があります。

チュートリアル: アクティブ/パッシブリージョン切り替えプランを作成する

このチュートリアルでは、us-east-1 で実行されているアプリケーションのアクティブ/パッシブリージョン切り替えプランを作成し、us-west-2 に復旧する方法について説明します。この例には、コンピューティング用の Amazon EC2 インスタンス、ストレージ用の Amazon Aurora Global Database、DNS 用の Amazon Route 53 が含まれます。

このチュートリアルでは、次の手順を実行します。

- リージョン切り替えプランの作成
- プランのワークフローと実行ブロックを構築する
- EC2 Auto Scaling グループ実行ブロックを構築する
- 2 つの手動承認実行ブロックを構築する
- 2 つのカスタムアクション Lambda 実行ブロックを構築する
- Amazon Aurora Global Database 実行ブロックを構築する
- ARC ルーティングコントロールブロックを構築する
- リージョン切り替えプランを実行する

前提条件

このチュートリアルを開始する前に、両方のリージョンで以下の前提条件が満たされていることを確認してください。

- 適切なアクセス許可を持つ IAM ロール
- EC2 Auto Scaling グループ
- メンテナンスページとフェンシング用の Lambda 関数

- Aurora Global Database
- ARC ルーティングコントロール

ステップ 1: リージョン切り替えプランを作成する

1. リージョン切り替えコンソールから [リージョン切り替えプランを作成] を選択します。
2. 次の詳細情報を入力します。
   - プライマリリージョン: us-east-1 を選択
   - スタンバイリージョン: us-west-2 を選択
   - 目標復旧時間 (RTO) (オプション)
   - IAM ロール: プラン実行の IAM ロールを入力します。この IAM ロールにより、リージョンの切り替えは実行中に AWS サービスを呼び出すことができます。
3. [作成] を選択します。

(オプション) 異なる AWS アカウントのリソースをリージョン切り替えプランに追加します。

1. クロスアカウントロールの作成:
   - リソースをホストするアカウントで、IAM ロールを作成します。
   - プランがアクセスする特定のリソースに対するアクセス許可を追加します。
   - 実行ロールが新しいロールを引き受けることを許可する信頼ポリシーを追加します。
   - 共有シークレットとして使用する外部 ID を入力して書き留めておきます。
2. プランでのリソースの設定:
   - リソースをプランに追加するときは、以下の 2 つの追加フィールドを指定します。
     - crossAccountRole: ステップ 1 で作成したロールの ARN
     - externalId: ステップ 1 で入力した外部 ID

アカウント 987654321 のリソースにアクセスする EC2 Auto Scaling 実行ブロックの設定例:

```
{
  "executionBlock": "EC2AutoScaling",
  "name": "ASG",
  "crossAccountRole": "arn:aws:iam::987654321:role/RegionSwitchCrossAccountRole",
  "externalId": "unique-external-id-123",
```

```
  "autoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:987654321:autoScalingGroup:*:autoScalingGroupName/CrossAccountASG"
}
```

必要なアクセス許可:

- 実行ロールには、クロスアカウントロールの sts:AssumeRole アクセス許可が必要です。
- クロスアカウントロールには、アクセスする特定のリソースに対するアクセス許可のみが必要です。
- クロスアカウントロールの信頼ポリシーには、以下を含める必要があります。
  - 信頼されたエンティティとしての実行ロールのアカウント。
  - 外部 ID 条件。
- クロスアカウントロールの設定の詳細については、「」を参照してくださいクロスアカウントリソースのアクセス許可。

プランを実行する前に、リージョン切り替えは以下を検証します。

- 実行ロールは、クロスアカウントロールを引き継ぐことができます。
- クロスアカウントロールには、必要なアクセス許可があります。
- 外部 ID は信頼ポリシーと一致します。

ステップ 2: プランのワークフローと実行ブロックを構築する

1. リージョン切り替えプランの詳細ページから、[ワークフローを構築] を選択します。
2. [すべてのリージョンで同じアクティベーションワークフローを構築する] を選択します。
3. リージョンアクティベーションワークフローの説明を入力します (オプション)。これは、プランの実行時にワークフローを簡単に識別するために使用されます。
4. [保存して続行] を選択します。

EC2 Auto Scaling 実行ブロックを追加する

この実行ブロックの詳細については、「」を参照してくださいAmazon EC2 Auto Scaling グループ実行ブロック。

1. [ステップを追加] を選択し、[シーケンス実行] を選択します。

2. EC2 Auto Scaling 実行ブロックを選択し、追加と編集を選択します。このブロックにより、パッシブリージョンでキャパシティの増加を開始できます。
3. 右側のパネルで、以下のブロックを設定します。

   - ステップ名:「スケール」と入力
   - ステップの説明 (オプション)
   - us-east-1 の Auto Scaling グループ ARN: us-east-1 の ASG の ARN
   - us-west-2 の Auto Scaling グループ ARN: us-west-2 の ASG の ARN
   - ソースリージョンのキャパシティと一致する割合: 100 を入力
   - キャパシティモニタリングアプローチ:「最新」のまま
   - タイムアウト (オプション)

   この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してくださいEC2 Auto Scaling 実行ブロックのサンプルポリシー。
4. [保存ステップ] を選択します。

手動承認実行ブロックを追加する

この実行ブロックの詳細については、「」を参照してください手動承認実行ブロック。

1. [ステップを追加] を選択します。
2. [手動承認実行ブロック] を選択し、設計ウィンドウに追加します。このブロックにより、先に進む前に人間による検証が可能になります。
3. 右側のパネルで、以下のブロックを設定します。

   - ステップ名:「セットアップ前の手動承認」と入力
   - ステップの説明 (オプション)
   - IAM 承認ロール: 実行を承認するためにユーザーが引き継ぐ必要があるロール
   - タイムアウト (オプション) タイムアウト後、実行は一時停止し、再試行、スキップ、またはキャンセルを選択できます。

   この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してください手動承認実行ブロックのサンプルポリシー。
4. [保存ステップ] を選択します。

メンテナンスページのカスタムアクション Lambda 実行ブロックを追加する

この実行ブロックの詳細については、「」を参照してくださいカスタムアクション Lambda 実行ブロック。

1. [ステップを追加] を選択します。
2. [カスタムアクション Lambda の実行ブロック] を選択し、[追加および編集] を選択します。このブロックは、アクティブ化しているリージョンにメンテナンスページを発行します。
3. 右側のパネルで、以下のブロックを設定します。

   - ステップ名:「メンテナンスページの表示」と入力
   - ステップの説明 (オプション)
   - us-east-1 をアクティブ化するための Lambda ARN: us-east-1 にデプロイされたメンテナンスページの Lambda 関数の ARN
   - us-west-2 をアクティブ化するための Lambda ARN: us-west-2 にデプロイされたメンテナンスページの Lambda 関数の ARN
   - Lambda 関数を実行するリージョン: [アクティブ化するリージョンで実行] を選択
   - タイムアウト (オプション)
   - 再試行間隔 (オプション)

   この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してくださいカスタムアクション Lambda 実行ブロックのサンプルポリシー。
4. [保存ステップ] を選択します。

Aurora グローバルデータベース実行ブロックを追加する

この実行ブロックの詳細については、「」を参照してくださいAmazon Aurora Global Database 実行ブロック。

1. [ステップを追加] を選択します。
2. [Aurora Global Database 実行ブロック] を選択し、[追加および編集] を選択します。このブロックは、Aurora Global Database のスイッチオーバー (データ損失なし) をトリガーします。詳細については、「Aurora ユーザーガイド」の「Aurora Global Database でのスイッチオーバーまたはフェイルオーバーの使用」を参照してください。
3. 右側のパネルで、以下のブロックを設定します。

- ステップ名: [Aurora スイッチオーバー] を入力
- ステップの説明 (オプション)
- Aurora Global Database 識別子: Aurora クラスターの名前
- us-east-1 のアクティブ化に使用されるクラスター ARN: us-east-1 の Aurora クラスター ARN
- us-west-2 のアクティブ化に使用されるクラスター ARN: us-west-2 の Aurora クラスター ARN
- Aurora データベースのオプションの選択: [スイッチオーバー] を選択
- タイムアウト (オプション)

この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してくださいAurora Global Database 実行ブロックのサンプルポリシー。

4. [保存ステップ] を選択します。

ARC ルーティングコントロール実行ブロックを追加する

この実行ブロックの詳細については、「」を参照してくださいARC ルーティングコントロール実行ブロック。

1. [ステップを追加] を選択します。
2. [ARC ルーティングコントロール実行ブロック] を選択し、[追加および編集] を選択します。このブロックは DNS フェイルオーバーを実行して、トラフィックをパッシブリージョンにシフトします。
3. 右側のパネルで、以下のブロックを設定します。

- ステップ名: [DNS の切り替え] を入力
- ステップの説明 (オプション)
- us-east-1 のアクティブ化に使用されるルーティングコントロール: [ルーティングコントロールの追加] を選択
- タイムアウト: タイムアウト値を入力します。

4. [ルーティングコントロールを追加]を選択します。

- ルーティングコントロール ARN: us-east-1 を制御するルーティングコントロールの ARN
- ルーティングコントロールの状態: [オン] を選択

5. [ルーティングコントロールを追加] をもう一度選択します。

- ルーティングコントロール ARN: us-west-2 を制御するルーティングコントロールの ARN
- ルーティングコントロールの状態: [オフ] を選択

6. [保存] を選択します。
7. us-west-2 のアクティブ化に使用されるルーティングコントロール: [ルーティングコントロールを追加] を選択
8. [ルーティングコントロールを追加]を選択します。
   - ルーティングコントロール ARN: us-west-2 を制御するルーティングコントロールの ARN
   - ルーティングコントロールの状態: [オン] を選択
9. [ルーティングコントロールを追加] をもう一度選択します。
   - ルーティングコントロール ARN: us-east-1 を制御するルーティングコントロールの ARN
   - ルーティングコントロールの状態: [オフ] を選択
10. [保存] を選択します。
11. [保存ステップ] を選択します。

    この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してくださいARC ルーティングコントロール実行ブロックサンプルポリシー。
12. [保存] を選択します。

ステップ 3: プランを実行する

1. リージョン切り替えプランの詳細ページで、右上の[実行] を選択します。
2. 実行の詳細を入力します。
   - アクティブ化するリージョンを選択します。
   - プラン実行モードを選択します。
   - (オプション) 実行ステップを表示します。
   - プランの実行を確認します。
3. [開始] を選択します。
4. 実行の詳細ページで、プランの実行時に詳細な手順を確認できます。開始時刻、終了時刻、リソース ARN、ログメッセージなど、プラン実行の各ステップを確認できます。

障害が発生したリージョンが復旧したら、プランを再度実行 (指定したパラメータを変更) して元のリージョンをアクティブ化し、アプリケーションオペレーションを元のプライマリリージョンに戻すことができます。

チュートリアル: プラン実行レポートの自動生成を設定する

このチュートリアルでは、リージョン切り替えプランのプラン実行レポートの自動生成を設定する方法について説明します。レポートは、コンプライアンス上の目的で計画の実行に関する包括的な PDF ドキュメントを提供します。

このチュートリアルでは、次の手順を実行します。

- レポートストレージ用の Amazon S3 バケットを作成する
- リージョン切り替えプランでレポートの自動生成を有効にする
- 計画を実行し、レポートをダウンロードする

前提条件

このチュートリアルを開始する前に、以下があることを確認してください。

- ワークフローが設定された既存のリージョン切り替えプラン
- Amazon S3 バケットを作成するアクセス許可
- 必要なアクセス許可で設定されたプランの実行 IAM ロール。詳細については、「自動計画実行レポートのアクセス許可」を参照してください。

ステップ 1: レポートの Amazon S3 バケットを作成する

1. https://console.aws.amazon.com/s3/ で Amazon S3 コンソールを開きます。
2. [バケットを作成] を選択します。
3. 次の詳細情報を入力します。

    - バケット名: などの一意の名前を入力します。 `my-region-switch-reports`
    - パブリックアクセスのブロック設定: すべてのパブリックアクセスをブロックしておく (推奨)
    - バケットバージョニング: バージョニングを有効にする (オプションですが推奨)
    - デフォルトの暗号化: 暗号化を選択します。SSM-KMS を使用する場合、planExecutionRole には s3 バケットのデフォルト CMK に対する kms:Encrypt および kms:GenerateDataKey アクセス許可が必要です

4. [バケットを作成] を選択します。
5. 次のステップで使用するバケット名を書き留めます。

ステップ 2: プランでレポートの自動生成を有効にする

1. でリージョンスイッチコンソールを開きますhttps://console.aws.amazon.com/route53recovery/regionswitch/home。
2. レポートを設定するプランを選択します。
3. ナビゲーションバーで「アクション」に移動し、「計画の詳細の編集」を選択します。
4. レポート設定セクションで、以下を指定します。
   - レポートの自動生成を有効にする を選択します。
   - Amazon S3 URI: ステップ 1 で作成したバケット S3 URI を選択または入力します。
   - バケットを所有するアカウント ID: バケット所有者のアカウント ID を入力します
5. [保存] を選択します。
6. 計画の評価が完了するまで待ちます。設定に問題がある場合は、プランの詳細ページに警告が表示されます。

ステップ 3: 計画を実行し、レポートをダウンロードする

1. プランの詳細ページで、実行を選択します。
2. 通常どおりに計画の実行を完了し、アクティブ化するリージョンと実行モードを選択します。
3. 計画の実行が完了したら、実行の詳細ページに移動します。
4. 計画実行レポートセクションで、レポート生成ステータスをモニタリングします。レポートの生成は通常、実行が完了してから 30 分以内に完了します。
5. レポートのステータスが「完了」と表示されたら、「計画実行レポートのダウンロード」を選択して PDF をダウンロードします。
6. または、Amazon S3 バケットに移動して、レポートに直接アクセスします。レポートは、次の命名パターンで保存されます。 `ExecutionReport-${planVersion.ownerAccountId}-${planName}-${execution.regionTo}-${event.executionId}-${dateStr}.pdf`

生成されたレポートには以下が含まれます。

- サービスの概要とレポート作成日を含むエグゼクティブサマリー

- 実行時に存在していた設定の詳細を計画する
- ステップ、影響を受けるリソース、ステータスを含む詳細な実行タイムライン
- 実行の開始時に存在していた警告を計画する
- 関連するアラームの Amazon CloudWatch アラーム状態とアラーム履歴
- 親プランの場合、子プランの設定と実行の詳細
- 用語と概念の用語集

トラブルシューティング

レポートの生成に失敗した場合は、以下を確認してください。

- アクセス許可エラー: 実行ロールに正しい IAM アクセス許可があることを確認します。詳細については、「自動計画実行レポートのアクセス許可」を参照してください。特定のアクセス許可の問題については、プラン評価の警告を確認してください。
- Amazon S3 バケットアクセス: Amazon S3 バケットが存在し、プランが設定されたリージョンからアクセス可能であることを確認します。バケットポリシーが実行ロールからのアクセスをブロックしていないことを確認します。
- バケット暗号化: バケット暗号化にカスタマー管理の KMS キーを使用する場合は、実行ロールに KMS キーを使用するアクセス許可があることを確認してください。

その他のヘルプについては、実行の詳細ページで詳細なエラーメッセージを表示するか、 AWS サポートにお問い合わせください。

チュートリアル: RDS 復旧後ワークフローを実行する

このチュートリアルでは、RDS フェイルオーバーが成功した後に復旧後のワークフローを実行する方法について説明します。この復旧後の実行は、RDS データベースのクロスリージョンレプリケーションを再確立することで冗長性を復元し、RDS データベースが将来のリージョンイベントに備えられるようにします。

このチュートリアルでは、次の手順を実行します。

- 復旧後の実行の前提条件を検証する
- RDS クロスリージョンレプリカの作成実行ブロックを使用して復旧後のワークフローを作成する
- 復旧後のワークフローを実行する

前提条件

このチュートリアルを開始する前に、以下があることを確認してください。

- RDS Promote リードレプリカ実行ブロックを含むアクティブ化ワークフローを持つリージョンスイッチのアクティブ/パッシブプラン
- 他のリージョンでリードレプリカを昇格させた正常なアクティブ化の実行
- 両方のリージョンは正常でアクセス可能
- 最新の復旧実行の実行 ID

ステップ 1: 復旧後のワークフローを作成する

1. リージョンスイッチコンソールからプランを選択し、ワークフローの編集を選択し、Config を選択し、プランに復旧後ワークフローを含めるをチェックして保存します。
2. ワークフローの編集ページで、ワークフローの選択を選択してステップをドロップダウンに追加し、復旧後を選択します。
3. [ステップを追加] を選択します。
4. Amazon RDS クロスリージョンレプリカ作成実行ブロックを選択します。
5. 右側のパネルで、以下のブロックを設定します。
   - ステップ名:「クロスリージョンリードレプリカの作成」と入力します。
   - ステップの説明 (オプション)
   - プライマリリージョンの RDS DB インスタンス ARN: プライマリリージョンのデータベースの ARN は、リードレプリカの昇格ステップと同じである必要があります
   - セカンダリリージョンの RDS DB インスタンス ARN: セカンダリの昇格されたデータベースの ARN は、リードレプリカの昇格ステップと同じである必要があります
   - タイムアウト (オプション): 90 分などのタイムアウト値を入力します

   この実行ブロックに必要な IAM アクセス許可の詳細については、「」を参照してくださいAmazon RDS 実行ブロックのサンプルポリシー。
6. [保存ステップ] を選択します。
7. ワークフローの保存を選択します。

ステップ 2: 復旧後のワークフローを実行する

1. リージョンスイッチプランの詳細ページの右上で、復旧後の実行を選択します。
2. 実行の詳細を入力します。
   - 復旧実行 ID: 最新の復旧実行の実行 ID を入力します。このフィールドは、現在アクティブなリージョンを識別するために使用されます。
   - 実行するリージョン: アプリケーショントラフィックを受信していない非アクティブなリージョンを選択します。これは、リードレプリカが作成されるリージョンです。
3. 実行ステップを確認し、実行を承認します。
4. [実行のスタート] を選択します。
5. 実行の詳細ページで実行の進行状況をモニタリングします。RDS クロスリージョンレプリカの作成実行ブロックは、古いプライマリインスタンスの名前を変更し、以前に障害が発生したリージョンに新しいリードレプリカを作成します。

復旧後の実行が正常に完了すると、アプリケーションはクロスリージョンレプリケーションを再確立し、今後のリージョンイベントに備えることができます。新しいリードレプリカが作成されたかどうかを確認するには、ターゲットリージョンの RDS コンソールを確認します。古いプライマリの名前が変更され、renamedByRegionSwitch というタグが付けられます。

> **Important**
>
> リージョンスイッチは、復旧実行 ID がプランの最後の既知の実行と一致することを検証します。実行 ID が無効であるか、既知の最後の復旧実行の ID でない場合、復旧後の実行は実行されません。

リージョン切り替え API オペレーション

次の表は、リージョン切り替えに使用できる ARC 操作と、関連するドキュメントへのリンクの一覧です。

アクション	ARC コンソールの使用	ARC API の使用	データプレーン API
プラン実行ステップを承認または拒否する	「手動承認実行ブロック」を参照してください。	「ApprovePlanExecutionStep」を参照してください。	はい
プラン実行をキャンセルする	「リージョン切り替えプランの作成」を参照してください。	「CancelPlanExecution」を参照してください。	はい
プランを作成する	「リージョン切り替えプランの作成」を参照してください。	「CreatePlan」を参照してください。	いいえ
プランを削除する	「リージョン切り替えの使用」を参照してください。	「DeletePlan」を参照してください。	いいえ
プランを取得する	「リージョン切り替えの使用」を参照してください。	「GetPlan」を参照してください。	いいえ
プラン評価ステータスを取得する	「プラン評価」を参照してください。	「GetPlanEvaluationStatus」を参照してください。	はい
プラン実行を取得する	「リージョン切り替えダッシュボード」を参照してください。	「GetPlanExecution」を参照してください。	はい
リージョンでプランを取得する	「リージョン切り替えの使用」を参照してください。	「GetPlanInRegion」を参照してください。	はい
プラン実行イベントを一覧表示する	「リージョン切り替えプランを実行してアプリケーションを	「ListPlanExecutionEvents」を参照してください。	はい

アクション	ARC コンソールの使用	ARC API の使用	データプレーン API
	復旧する」を参照してください。		
プラン実行を一覧表示する	「リージョン切り替えプランを実行してアプリケーションを復旧する」を参照してください。	「ListPlanExecutions」を参照してください。	はい
プランを一覧表示する	「リージョン切り替えの使用」を参照してください。	「ListPlans」を参照してください。	いいえ
リージョンのプランを一覧表示する	「リージョン切り替えの使用」を参照してください。	「ListPlans InRegion」を参照してください。	はい
プランの Route 53 ヘルスチェックを一覧表示する	「Amazon Route 53 ヘルスチェック実行ブロック」を参照してください。	「ListRoute53HealthChecksForPlan」を参照してください。	いいえ
リージョンのプランの Route 53 ヘルスチェックを一覧表示する	「Amazon Route 53 ヘルスチェック実行ブロック」を参照してください。	「ListRoute53HealthChecksForPlanInRegion」を参照してください。	はい
リソースのタグの一覧表示	「ARC リージョン切り替えのタグ付け」を参照してください。	「ListTagsForResource」を参照してください	いいえ

アクション	ARC コンソールの使用	ARC API の使用	データプレーン API
プラン実行を開始する	「リージョン切り替えプランを実行してアプリケーションを復旧する」を参照してください。	「StartPlanExecution」を参照してください。	はい
リソースをタグ付けする	「リージョン切り替えプランの作成」を参照してください。	「TagResource」を参照してください	いいえ
リソースからタグを削除する	「ARC リージョン切り替えのタグ付け」を参照してください。	「UntagResource」を参照してください	いいえ
プランを更新する	「リージョン切り替えプランの作成」を参照してください。	「UpdatePlan」を参照してください。	いいえ
プラン実行を更新する	「リージョン切り替えプランの作成」を参照してください。	「UpdatePlanExecution」を参照してください。	はい
プラン実行ステップを更新する	「リージョン切り替えプランの作成」を参照してください。	「UpdatePlanExecutionStep」を参照してください。	はい

リージョン切り替えの使用

このセクションでは、マルチリージョンアプリケーションの復旧に使用できるリージョン切り替えプランを使用するステップバイステップ手順について説明します。リージョン切り替えを使用すると、アクティブ/パッシブリカバリアプローチとアクティブ/アクティブリカバリアプローチの両方のプランを作成できます。

アプリケーションの復旧プランを作成するには、次の手順を実行します。

1. リージョン切り替えプランの作成 プランは、 AWS リージョン アプリケーションが実行される特定の など、特定の属性を持つ構造です。各プランには 1 つ以上のワークフローが含まれます。

   必要に応じて、複数のプランを作成し、それらの子プランを全体的な復旧プラン内にネストできます。
2. プランのワークフローを作成します。最初にワークフローを作成しないと、プランを実行できません。
3. ワークフローで、それぞれが実行ブロックである 1 つ以上のステップを追加します。

   たとえば、送信先リージョンで EC2 Auto Scaling グループをスケールアップするステップを追加できます。
4. ワークフローにステップを追加した後、Amazon Route 53 でヘルスチェックを設定するなど、追加のステップが必要になる場合があります。各実行ブロックセクションには、必要な設定情報が含まれています。詳細については、「実行ブロックを追加する」を参照してください。
5. 障害が発生した で実行されているアプリケーションを復旧するには AWS リージョン、 プランを実行します。

   グローバルダッシュボードまたはリージョンダッシュボードで情報を表示することにより、プラン実行の進行状況を追跡できます。

以下のセクションでは、プランとワークフローを作成し、ワークフローに実行ブロックステップを追加するための詳細な情報と手順について説明します。

内容

- リージョン切り替えプランの作成
- リージョンの切り替えプランワークフローの作成
- 実行ブロックを追加する
- 子プランを作成する
- リージョン切り替えプランのトリガーを作成する
- リージョン切り替えプランを実行してアプリケーションを復旧する

このセクションの手順では、 AWS マネジメントコンソールを使用してプラン、ワークフロー、実行ブロック、トリガーを操作する方法について説明します。代わりにリージョン切り替え API オペレーションを使用するには、「リージョン切り替え API オペレーション」を参照してください。

リージョン切り替えプランの作成

リージョンの切り替えでは、アクティブ/アクティブプランとアクティブ/パッシブプランの 2 種類のプランを作成できます。プランを作成するときは、フェイルオーバーの管理方法に適用されるタイプを指定します。

- アクティブ/パッシブアプローチでは、2 つのアプリケーションレプリカを 2 つのリージョンにデプロイし、トラフィックはアクティブなリージョンのみにルーティングされます。リージョン切り替えプランを実行することで、パッシブリージョンのレプリカをアクティブ化できます。
- アクティブ/アクティブアプローチでは、2 つのアプリケーションレプリカを 2 つのリージョンにデプロイし、両方のレプリカが作業を処理したり、トラフィックを受信したりします。

リージョン切り替えプランを作成するには

1. リージョン切り替えコンソールから、アクティブ/パッシブアプローチで [リージョン切り替えプランを作成] を選択します。
2. 次の詳細情報を入力します。
   - プラン名 - プランのわかりやすい名前を入力します。
   - マルチリージョンアプローチ - [アクティブ/パッシブ] または [アクティブ/アクティブ] を選択します。このアプローチでは、2 つのアプリケーションレプリカが 2 つのリージョンにデプロイされ、トラフィックはアクティブなリージョンにのみルーティングされます。リージョン切り替えプランを実行することで、パッシブリージョンのレプリカをアクティブ化できます。
     - アクティブ/パッシブアプローチでは、2 つのアプリケーションレプリカを 2 つのリージョンにデプロイし、トラフィックはアクティブなリージョンのみにルーティングされます。リージョン切り替えプランを実行することで、パッシブリージョンのレプリカをアクティブ化できます。
     - アクティブ/アクティブアプローチは、2 つのアプリケーションレプリカを 2 つのリージョンにデプロイしていて、両方のレプリカが作業を処理しているかトラフィックを受信している場合に選択します。
   - [プライマリリージョンとスタンバイリージョン] または [リージョン] - アプリケーションのプライマリリージョンとスタンバイリージョンを選択します。アクティブ/アクティブデプロイの場合は、レプリカがデプロイされるリージョンを選択します。
   - 目標復旧時間 (RTO) - 目的の RTO を入力します。リージョン切り替えはこれを使用して、リージョン切り替えプランの実行が目的の RTO と比較して完了するまでにかかる時間に関するインサイトを提供します。

- IAM ロール - プランの実行に使用するリージョン切り替え用の IAM ロールを指定します。権限の詳細については、「ARC でのリージョン切り替えの Identity and Access Management」を参照してください。
- Amazon CloudWatch アラーム - Amazon CloudWatch で作成したアプリケーションヘルスアラームを指定して、各リージョンのアプリケーションの状態を示します。リージョンを切り替えると、これらのアプリケーションヘルスアラームを使用して、リージョンを切り替えてリカバリを実装した後の実際のリカバリ時間を判断できます。

  リージョン切り替えプランに CloudWatch アラームを追加する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「アプリケーションヘルス許可の CloudWatch アラーム」を参照してください。
- 自動レポート生成 - オプションで、プラン実行の自動レポート生成を有効にします。有効にすると、リージョンスイッチは各計画の実行が完了した後に包括的な PDF レポートを生成し、指定した Amazon S3 バケットに配信します。Amazon S3 URI とバケットを所有するアカウント ID を指定します。

  プランの自動レポート生成を有効にする前に、正しい IAM ポリシーが設定されていることを確認してください。レポートの生成と必要なアクセス許可の詳細については、「」を参照してください自動計画実行レポート。
- タグ - オプションで、プランに 1 つ以上のタグを追加します。

リージョンの切り替えプランワークフローの作成

リージョン切り替えプランを作成したら、アプリケーションの復旧プロセスを指定するワークフローを定義して作成する必要があります。プランごとに、アプリケーションの復旧を完了する 1 つ以上のワークフローを定義します。各ワークフローには、アプリケーションのリカバリのためにリージョン切り替えに実行させたい各アクションを定義する実行ブロックを含むステップを追加します。

作成するワークフローの数は、アプリケーションのデプロイシナリオとリカバリ管理の好みによって異なります。例えば、次のようになります。

- リージョン切り替えプランがアクティブ/アクティブアプリケーションのデプロイ用である場合は、非アクティブ化ワークフローも作成する必要があります。つまり、アクティブ/アクティブ展開の場合、アクティベーションワークフローと非アクティベーションワークフローの少なくとも 2 つのワークフローが必要になります。

- リージョン切り替えプランがアクティブ/パッシブアプリケーションデプロイの場合、プライマリリージョンとセカンダリリージョンがあります。リージョンごとに個別のアクティベーションワークフローを設定する場合は、リージョンごとに 1 つずつ、2 つのワークフローを作成します。

リージョンの切り替えプランワークフローを作成するには

1. 作成したリージョン切り替えプランで、[ワークフローを構築] を選択します。
2. 以下のワークフローオプションのいずれかを選択します。
   - すべてのリージョンで同じアクティベーションワークフローを構築する - リージョン間で同じアクティベーションワークフローを使用できます。
   - リージョンごとに個別にワークフローを構築する - リージョンごとに個別のアクティベーションワークフローを構築します。
3. 必要に応じて、各ワークフローの説明を入力します。
4. アプリケーションの復旧に必要なワークフローを定義します。ワークフローで、実行ブロックを追加して、復旧のためにリージョン切り替えで実行する手順を定義します。各実行ブロックは、アクティブ化するリージョンでのアプリケーショントラフィックの再ルーティングやデータベース復旧などのアクションを定義し、別の AWS アカウントのリソースをサポートします。実行ブロックを並行して実行するか、順番に実行するかを選択できます。ワークフローに追加できる特定の実行ブロックの詳細については、「実行ブロックを追加する」を参照してください。
5. 選択したワークフローオプションに応じて、以下を実行します。
   - [すべてのリージョンで同じアクティベーションワークフローを構築する] を選択した場合、1 つのアクティベーションワークフローが必要です。
   - [リージョンごとに個別にワークフローを構築する] を選択した場合、2 つのアクティベーションワークフローが必要です。

   アクティブ/アクティブプランの場合、アクティベーションワークフローと非アクティブ化ワークフローの両方を定義する必要があります。

実行ブロックを追加する

リージョンスイッチプランのワークフローにステップを追加して、アプリケーションのフェイルオーバーまたはスイッチオーバーを完了する個々のステップを実行します。各タイプの実行ブロックの機能と動作の詳細については、以下の説明を参照してください。

リージョン切り替えは、プランを作成または更新した直後にプラン評価を実行し、その後は定常状態で 30 分ごとにプラン評価を実行します。リージョン切り替えは、プランが設定されたすべてのリージョンにプラン評価に関する情報を保存します。各実行ブロックセクションには、リージョン切り替えがプラン評価を実行するときに評価される内容に関する情報が含まれています。

リージョン切り替えには、復旧の一環としてコンピューティングリソースをスケーリングするのに役立つ実行ブロックタイプが含まれています。プランでこれらの実行ブロックを使用する場合、リージョン切り替えによって必要なコンピューティング能力が達成される保証がないため、注意が必要です。重要なアプリケーションがあり、キャパシティへのアクセスを保証する必要がある場合は、キャパシティを予約することをお勧めします。セカンダリリージョンでコンピューティングキャパシティを予約しながら、コストを制限するための戦略があります。詳細については、「リザーブドキャパシティのパイロットライト: オンデマンドキャパシティ予約を使用して DR コストを最適化する方法」を参照してください。

リージョン切り替えは、以下の実行ブロックをサポートします。

実行ブロック	関数	不正な設定
ARC リージョン切り替えプラン実行ブロック	実行する子プランを指定して、1 回の実行で複数のアプリケーションの復旧をオーケストレーションします。	不正な設定で子プランを開始します。
Amazon EC2 Auto Scaling グループ実行ブロック	プランの実行の一環として、Auto Scaling グループにある EC2 コンピューティングリソースをスケールします。	アクティブ化するリージョンで一致させるコンピューティングキャパシティの最小パーセンテージを指定します。
Amazon EKS リソーススケーリング実行ブロック	プラン実行の一環として Amazon EKS クラスターポッドをスケールします。	該当なし
Amazon ECS サービススケーリング実行ブロック	プラン実行の一環として Amazon ECS サービスタスクをスケールします。	該当なし

実行ブロック	関数	不正な設定
ARC ルーティングコントロール実行ブロック	ステップを追加して、1 つ以上の ARC ルーティングコントロールの状態を変更し、アプリケーショントラフィックをターゲット AWS リージョンにリダイレクトします。	該当なし
Amazon Aurora Global Database 実行ブロック	Aurora Global Database の復旧ワークフローを実行します。	Aurora Global Database のフェイルオーバーを実行します (データ損失が発生する可能性があります)。
Amazon DocumentDB グローバルクラスター実行ブロック	Amazon DocumentDB グローバルクラスターの復旧ワークフローを実行します。	Amazon DocumentDB グローバルクラスターのフェイルオーバーを実行します (データ損失が発生する可能性があります) 。
Amazon RDS Promote リードレプリカ実行ブロック	Amazon RDS リードレプリカをスタンドアロンデータベースインスタンスに昇格させます。	該当なし
Amazon RDS クロスリージョンレプリカの作成実行ブロック	復旧後の一環として、Amazon RDS データベースインスタンスのクロスリージョンリードレプリカを作成します。	該当なし
手動承認実行ブロック	承認ステップを挿入して、続行する前に実行の承認またはキャンセルを要求します。	該当なし
カスタムアクション Lambda 実行ブロック	Lambda 関数を実行するためのカスタムステップを追加して、カスタムアクションを有効にします。	ステップをスキップします。

実行ブロック	関数	不正な設定
Amazon Route 53 ヘルスチェック実行ブロック	フェイルオーバー中にアプリケーショントラフィックがリダイレクトされる先のリージョンを指定します。	該当なし

ARC リージョン切り替えプラン実行ブロック

リージョン切り替えプラン実行ブロックを使用すると、他の子リージョン切り替えプランを参照することにより、複数のアプリケーションがアクティブ化するリージョンに切り替える順序をオーケストレーションできます。この親子関係を使用すると、インフラストラクチャ全体で複数のリソースと依存関係を管理する、複雑で調整された復旧プロセスを作成できます。

設定

リージョン切り替えプラン実行ブロックを使用する場合は、作成するプランのワークフローで実行する特定のリージョン切り替えプランを選択します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「リージョン切り替えプラン実行ブロックのサンプルポリシー」を参照してください。

リージョン切り替えプラン実行ブロックを設定するには、以下の値を入力します。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョン切り替えプラン: 現在のプランのワークフローで実行するプランを選択します。

次に、[ステップを保存] を選択します。

仕組み

リージョン切り替えプラン実行ブロックを使用して、親子関係を持つ親ワークフローを作成します。この実行ブロックは、追加のレベルの子プランをサポートしておらず、親子プランの数を制限するこ

とに注意してください。子プランは、親プランがサポートするリージョンと同じリージョンをサポートし、親プランと同じ復旧アプローチ (アクティブ/アクティブまたはアクティブ/パッシブ) を持つ必要があります。

このブロックは、正常な実行モードと不正な実行モードの両方をサポートします。不正な設定は、不正な設定で子プランを開始します。リージョン切り替えブロックが正常に実行され、正常ではない実行モードに切り替えた場合、子プランも正常ではない実行モードに切り替わります。

プラン評価の一環として評価されるもの

アカウント間でプランを共有し、そのプランが親プランのアカウントと共有されなくなった場合、リージョン切り替え評価は、そのプランが無効であるという警告を返します。

Amazon EC2 Auto Scaling グループ実行ブロック

EC2 Auto Scaling グループ実行ブロックを使用すると、マルチリージョンリカバリプロセスの一部として EC2 インスタンスをスケーリングできます。退出するリージョン (送信元と送信先) に対するキャパシティの割合を定義できます。

設定

EC2 Auto Scaling グループ実行ブロックを設定するときは、プランに関連付けられている特定のリージョンの EC2 Auto Scaling ARNs を入力します。プランの実行中にスケールアップする各リージョンに EC2 Auto Scaling ARNs を入力する必要があります。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「EC2 Auto Scaling 実行ブロックのサンプルポリシー」を参照してください。

EC2 Auto Scaling グループ実行ブロックを設定するには、次の値を入力します。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョンの EC2 Auto Scaling グループ ARN: プランの各リージョンの EC2 Auto Scaling グループの ARN を入力します。
4. アクティブ化されたリージョンの容量と一致する割合: アクティブ化されたリージョンに一致する Auto Scaling グループ内の実行中のインスタンス数の必要な割合を入力します。

5. キャパシティモニタリングアプローチ: EC2 Auto Scaling グループのキャパシティをモニタリングするには、次のいずれかのアプローチを選択します。
   - 24 時間にわたってサンプリングされた最大実行容量: EC2 Auto Scaling グループ設定で指定された希望する容量値を使用するには、このオプションを選択します。このオプションは追加コストを発生させませんが、他のオプションである CloudWatch メトリクスを使用するよりも精度が低い可能性があります。

     リージョン切り替え API では、このオプションは `sampledMaxInLast24Hours` の指定に対応します。

     詳細については、Amazon EC2 Auto Scaling ユーザーガイド」の「Auto Scaling グループのスケーリング制限を設定する」を参照してください。 Auto Scaling
   - CloudWatch で 24 時間にわたってサンプリングされた最大実行容量: Amazon CloudWatch for EC2 Auto Scaling で指定されたメトリクスを使用するには、このオプションを選択します。オプションを使用すると、より正確になる場合がありますが、CloudWatch メトリクスを使用するための追加コストが発生します。

     リージョン切り替え API では、このオプションは `autoscalingMaxInLast24Hours` の指定に対応します。

     このオプションを使用するには、まず Auto Scaling グループのグループメトリクスを有効にする必要があります。詳細については、Amazon EC2 Auto Scaling ユーザーガイド」の「Auto Scaling グループのメトリクスを有効にする」を参照してください。 Auto Scaling
6. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

EC2 Auto Scaling 実行ブロックを設定すると、リージョンスイッチは、送信元 Auto Scaling グループと送信先 Auto Scaling グループが 1 つだけであることを確認します。複数の Auto Scaling グループがある場合、プラン評価中に実行ブロックは失敗します。ターゲットキャパシティは、状態が `InService` に設定されているインスタンスの数として定義されます。詳細については、EC2 Auto Scaling インスタンスのライフサイクル」を参照してください。

一致する割合に対して指定した値 (Auto Scaling 実行ブロックを設定する場合) に基づいて、リージョンスイッチは送信先の Auto Scaling グループの新しい希望する容量を計算します。新しい希望する容量は、移行先の Auto Scaling グループの希望する容量と比較されます。希望するキャパ

シティの計算でリージョン切り替えが使用する式は `ceil(percentToMatch * Source Auto Scaling group capacity)` です。ceil() は、小数の結果を四捨五入する関数です。送信先の Auto Scaling グループの現在の希望するキャパシティが、リージョンスイッチが計算する新しい Auto Scaling グループの希望するキャパシティ以上である場合、実行ブロックが続行されます。リージョンスイッチは Auto Scaling グループの容量をスケールダウンしないことに注意してください。

リージョンスイッチが Auto Scaling ブロックを実行すると、リージョンスイッチはターゲットリージョン Auto Scaling グループの容量を目的の容量に合わせてスケールアップしようとします。次に、リージョンスイッチは、リージョンスイッチがプランの次のステップに進む前に、ターゲットリージョンの Auto Scaling グループでリクエストされた Auto Scaling グループの容量が満たされるまで待機します。

> **Note**
>
> このブロックを実行すると、Auto Scaling グループの最小容量と希望する容量の設定が変更されるため、infrastructure-as-code ツールやその他のオートメーションを使用してこれらの値を管理すると、設定ドリフトが発生する可能性があります。意図しないロールバックを防ぐために、設定管理プロセスでこれらの変更が考慮されていることを確認してください。

アクティブ/アクティブアプローチを使用している場合、リージョン切り替えは他の設定済みリージョンをソースとして使用します。つまり、リージョンが非アクティブ化されている場合、リージョン切り替えは、スケールするパーセントに一致するソースとして他のアクティブなリージョンを使用します。

このブロックは、正常な実行モードと不正な実行モードの両方をサポートします。リージョンの切り替えがプランの次のステップに進む前に、ターゲットリージョンで一致するコンピューティングキャパシティの最小パーセンテージを指定することにより、不正な実行を設定できます。

プラン評価の一環として評価されるもの

リージョンスイッチがプランを評価すると、リージョンスイッチは EC2 Auto Scaling グループ実行ブロック設定とアクセス許可に対していくつかの重要なチェックを実行します。リージョンスイッチの評価では、Auto Scaling グループが両方のリージョンに存在することを確認し、それらが正しく設定され、アクセス可能であることを確認し、各リージョンで実行中のインスタンスの数を記録します。また、ターゲットリージョンの Auto Scaling グループの最大容量が、必要な容量のスケールの指定された一致率を処理するのに十分であることを確認します。

リージョンスイッチは、プランの IAM ロールに Auto Scaling に対する正しいアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。いずれかのチェックが失敗した場合、リージョン切り替えは警告メッセージを返します。警告メッセージはコンソールで表示できます。または、EventBridge または API オペレーションを使用して検証警告を受け取ることができます。

Amazon EKS リソーススケーリング実行ブロック

EKS リソーススケーリング実行ブロックを使用すると、マルチリージョンリカバリプロセスの一部として EKS リソースをスケーリングできます。実行ブロックを設定するときに、非アクティブ化されるリージョンのキャパシティを基準として、スケールするキャパシティのパーセンテージを定義します。

EKS アクセスエントリのアクセス許可を設定する

EKS リソーススケーリングのステップを追加する前に、EKS クラスター内の Kubernetes リソースでアクションを実行するために必要なアクセス許可をリージョンスイッチに提供する必要があります。リージョン切り替えへのアクセスを提供するには、リージョン切り替えアクセスポリシー `arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy` を使用して、リージョン切り替えがプランの実行に使用する IAM ロールの EKS アクセスエントリを作成する必要があります。

リージョン切り替え EKS アクセスポリシー

以下の情報は、EKS アクセスポリシーに関する詳細を示します。

名前: `AmazonARCRegionSwitchScalingPolicy`

ポリシー ARN: `arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy`

Kubernetes API グループ	Kubernetes resources	Kubernetes 動詞 (許可)
*	*/scale	get、update
*	*/status	get
autoscaling	horizontalpodautoscalers	get、patch

リージョン切り替えの EKS アクセスエントリを作成する

次の例では、リージョン切り替えが Kubernetes リソースに対して特定のアクションを実行できるように、必要なアクセスエントリとアクセスポリシーの関連付けを作成する方法について説明します。この例では、IAM ロール `arn:aws:iam::555555555555:role/`*`my-role`* の EKS クラスター *`my-cluster`* 内の名前空間 *`my-namespace1`* にアクセス許可が適用されます。

これらのアクセス許可を設定するときは、実行ブロック内の両方の EKS クラスターに対してこれらのステップを実行してください。

前提条件

開始する前に、クラスターの認証モードを API_AND_CONFIG_MAP または API に変更します。認可モードを変更すると、アクセスエントリの API が追加されます。詳細については、「Amazon EKS ユーザーガイド」の「アクセスエントリを使用するように認証モードを変更する」を参照してください。

アクセスエントリを作成する

最初のステップでは、次のような AWS CLI コマンドを使用してアクセスエントリを作成します。

```
aws eks create-access-entry --cluster-name my-cluster --principal-arn
        arn:aws:iam::555555555555:user/my-user --type STANDARD
```

詳細については、「Amazon EKS ユーザーガイド」の「アクセスエントリを作成する」を参照してください。

アクセスエントリの関連付けを作成する

次に、次のような AWS CLI コマンドを使用して、リージョンスイッチアクセスポリシーへの関連付けを作成します。

```
aws eks associate-access-policy --cluster-name my-cluster --principal-arn
 arn:aws:iam::555555555555:role/my-role \
         --access-scope type=namespace,namespaces=my-namespace1 --policy-arn
         arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy
```

詳細については、「Amazon EKS ユーザーガイド」の「アクセスポリシーをアクセスエントリに関連付ける」を参照してください。

実行ブロックの 2 番目の EKS クラスターで、他のリージョンでこれらのステップを繰り返し、両方のクラスターにリージョン切り替えでアクセスできるようにします。

設定

> **Important**
>
> EKS リソーススケーリングステップを追加する前に、まず が正しいアクセス許可が設定されていることを確認してください。詳細については、「EKS アクセスエントリのアクセス許可を設定する」を参照してください。また、正しい IAM ポリシーが設定されていることを確認します。詳細については、「Amazon EKS リソーススケーリング実行ブロックサンプルポリシー」を参照してください。

リージョン切り替えは現在、apps/v1、Deployment、apps/v1 の ReplicaSet リソースをサポートしています。

実行ブロック設定には、次の値を入力します。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. アプリケーション名: myApplication など、EKS アプリケーションの名前を入力します。
4. Kubernetes リソースの種類: Deploymentなど、アプリケーションのリソースの種類を入力します。
5. リージョンのリソース: リージョンごとに、EKS クラスター ARN、リソース名前空間など、EKS クラスターの情報を入力します。
6. アクティブ化されたリージョンのキャパシティと一致する割合: アクティブ化されたリージョンで一致する、ソースリージョンで実行中のポッドの希望する割合を入力します。
7. キャパシティモニタリングアプローチ: キャパシティモニタリングの唯一のオプション [24 時間にわたってサンプリングされた最大実行キャパシティ] が既に選択されています。

   このキャパシティモニタリングアプローチでは、EKS サービスリクエストに `ReplicaCount` 値を使用します。詳細については、「Amazon Elastic Kubernetes Service ユーザーガイド」の Amazon EKS での ARC ゾーンシフトの説明を参照してください。
8. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

プランの実行中に、リージョン切り替えは、アクティブ化するリージョン内のターゲットリソースについて、過去 24 時間のサンプルされたレプリカの最大数を取得します。次に、式 `ceil(percentToMatch * Source replica count)` を使用して、送信先リソースに必要なレプリカ数を計算します。

レプリケート先の準備完了レプリカ数が希望する値よりも少ない場合、リージョン切り替えはレプリケート先のリソースレプリカ値を希望するキャパシティにスケールします。レプリカの準備が整うまで待機し、必要に応じてノードの自動スケーラーを活用してノードのキャパシティを増やします。

オプションの `hpaName` フィールドが空でない場合、リージョンは HorizontalPodAutoscaler にパッチを適用し、パッチ `{"spec":{"behavior":{"scaleDown":{"selectPolicy":"Disabled"}}}}` を使用して実行中または実行後に自動スケールダウンを防止します。

パッチ内のリソースのレプリカフィールドと HorizontalPodAutoscaler フィールドを無視するよう、GitOps ツールなどのドリフト修正ツールを必ず設定してください。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えは設定された EKS 実行ブロックとアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、プランの IAM ロールに EKS クラスターを記述し、関連するアクセスエントリポリシーを一覧表示するための正しいアクセス許可があることを確認します。リージョン切り替えは、IAM ロールが正しいアクセスエントリポリシーに関連付けられていることも検証するため、リージョン切り替えには Kubernetes リソースを操作するために必要なアクセス許可が付与されます。最後に、リージョン切り替えは、設定された EKS クラスターと Kubernetes リソースが存在することを確認します。

さらに、リージョン切り替えは、必要なモニタリングデータ (Kubernetes レプリカ数) が正常に収集および保存されていることをチェックし、リージョン切り替えプランの実行に必要な実行中のポッドの数をキャプチャします。

Amazon ECS サービススケーリング実行ブロック

ECS サービススケーリング実行ブロックを使用すると、マルチリージョンリカバリプロセスの一環として、送信先リージョンで ECS サービスをスケーリングできます。キャパシティの割合は、リージョン切り替えがフェイルオーバーまたは非アクティブ化するリージョンを基準として定義できます。

設定

ECS サービススケーリング実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Amazon ECS サービススケーリング実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョンのリソース: リージョンごとに、ECS クラスター ARN と ECS サービス ARN を入力します。
4. ソースリージョンのタスク数と一致する割合: アクティブ化されたリージョンで一致するソースリージョンで実行中のタスクの望ましい割合を入力します。
5. キャパシティモニタリングアプローチ: Amazon ECS のキャパシティをモニタリングするための次のいずれかのアプローチを選択します。
   - 24 時間にわたってサンプリングされた最大実行キャパシティ: Amazon ECS サービスで実行中のタスク数の値を使用するには、このオプションを選択します。このオプションは追加コストを発生させませんが、他のオプションである CloudWatch メトリクスを使用するよりも精度が低い可能性があります。

     リージョン切り替え API では、このオプションは `sampledMaxInLast24Hours` の指定に対応します。

     詳細については、「Amazon Elastic Container Service 開発者ガイド」の「Amazon ECS サービスを自動的にスケールする」を参照してください。
   - コンテナインサイトを介して 24 時間にわたってサンプリングされた最大実行キャパシティ: Amazon ECS Container Insights メトリクスを使用するには、このオプションを選択します。オプションを使用すると、より正確になりますが、コンテナインサイトを使用するための追加コストが発生します。

     リージョン切り替え API では、このオプションは `autoscalingMaxInLast24Hours` の指定に対応します。

このオプションを使用するには、まずコンテナインサイトを有効にする必要があります。詳細については、「Amazon CloudWatch ユーザーガイド」の「コンテナインサイトの設定」を参照してください。

6. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

プランで実行ブロックを設定すると、リージョン切り替えは送信元 ECS サービスが 1 つと送信先サービスが 1 つしかないことを確認します。複数のサービスがある場合、リージョン切り替えは実行ブロックの警告を返します。リージョン切り替えは、プランが設定されているすべてのリージョンにこのデータを保存します。ターゲットキャパシティは、ECS サービスで設定された必要数として定義されます。

アクティブ/パッシブアプローチの場合、リージョン切り替えは送信先 (アクティブ化) リージョンの ECS サービスの、希望する新しいキャパシティを計算します。希望する新しいキャパシティは、送信先 ECS サービスの希望するキャパシティと比較されます。希望するキャパシティの計算でリージョン切り替えが使用する式は `ceil(percentToMatch * Source Auto Scaling group capacity)` です。ceil() は、小数の結果を四捨五入する関数です。送信先 ECS サービスの現在の必要数が、ECS サービスの計算された新しい必要キャパシティよりも多い場合、プランの実行が続行されます。リージョン切り替えは ECS サービスキャパシティをスケールダウンしないことに注意してください。

ECS サービスでアプリケーション自動スケーリングが有効になっている場合、リージョン切り替えはアプリケーション自動スケーリングの最小キャパシティを更新し、ECS サービスの必要数も更新します。

リージョン切り替え ECS サービスブロックを実行すると、リージョン切り替えは、ターゲットリージョン ECS キャパシティを、希望するキャパシティに合わせてスケールアップしようとします。次に、リージョン切り替えは、リージョン切り替えがプランの次のステップに進む前に、ターゲットリージョンの ECS サービスでリクエストされた ECS サービスキャパシティが満たされるまで待機します。必要に応じて、リージョン切り替えがキャパシティフルフィルメントを待機するタイムアウト制限を設定して、フルフィルメントの完了前にステップが完了するよう設定できます。

アクティブ/アクティブアプローチを使用している場合、リージョン切り替えは他の設定済みリージョンをソースとして使用します。つまり、リージョンが非アクティブ化されている場合、リージョ

ン切り替えは、スケールするパーセントに一致するソースとして他のアクティブなリージョンを使用します。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えは ECS サービス実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、ソースリージョンとターゲットリージョンの両方に ECS サービスが存在することを確認し、ターゲットリージョンの ECS サービスに設定された最大キャパシティが、ターゲットリージョンのキャパシティの指定された割合の一致を処理するのに十分であることを確認します。リージョン切り替えは、プランの IAM ロールに ECS サービスに対する正しいアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

さらに、リージョン切り替えは、 `ResourceMonitor` が ECS サービスに必要なモニタリングデータを正常に収集して保存したことを確認し、実行中のタスクの数をキャプチャします。

いずれかのチェックが失敗した場合、リージョン切り替えは警告メッセージを返します。警告メッセージはコンソールで表示できます。または、EventBridge または API オペレーションを使用して検証警告を受け取ることができます。

ARC ルーティングコントロール実行ブロック

アプリケーションの Amazon Application Recovery Controller (ARC) ルーティングコントロールを設定している場合は、ARC ルーティングコントロールステップを追加してアプリケーショントラフィックをリダイレクトできます。このステップでは、1 つ以上の ARC ルーティングコントロールの状態を変更して、アプリケーショントラフィックを送信先にリダイレクトできます AWS リージョン。ARC ルーティングコントロールは、ルーティングコントロールに関連付けられた DNS レコードで設定された Amazon Route 53 のヘルスチェックを使用してトラフィックをリダイレクトします。

> **Important**
>
> Amazon Application Recovery Controller (ARC) ルーティングコントロールは、 AWS 商用パーティションでのみ使用できます。

設定

ルーティングコントロール実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「ARC ルーティングコントロール実行ブロックサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. 必要なルーティングコントロール: アクティブ化または非アクティブ化するリージョンごとに、ルーティングコントロールの ARN と、ルーティングコントロールの初期状態であるオンまたはオフを入力します。
4. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

この実行ブロックの想定されるパターンは、特定の でのアプリケーションのセットアップ方法と一致するルーティングコントロールと初期状態を指定することです AWS リージョン。例えば、アプリケーションのリージョン A とリージョン B をアクティブ化できるプランがある場合、状態をオンに設定するリージョン A のルーティングコントロールと、状態をオンに設定するリージョン B のルーティングコントロールがあります。

次に、プランを実行し、リージョン A をアクティブ化するように指定すると、この実行ブロックを含むワークフローは指定されたルーティングコントロールをオンに更新し、トラフィックをリージョン A に転送します。

仕組み

ARC ルーティングコントロール実行ブロックを設定することで、アプリケーショントラフィックを宛先に再ルーティングしたり AWS リージョン、アクティブ/アクティブアプローチでは、非アクティブ化するリージョンへのトラフィックのルーティングを停止したりできます。プランに複数のワークフローが含まれている場合は、使用するすべてのルーティングコントロール実行ブロックの DNS レコードに同じ入力を指定してください。

このブロックは、不正な実行モードをサポートしていません。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えはルーティングコントロールの実行ブロック設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、指定されたルーティングコントロールが正しく設定され、アクセス可能であることを確認します。

リージョン切り替えは、プランの IAM ロールにルーティングコントロールの状態にアクセスして更新するために必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

ルーティングコントロール実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップが実行されている間に、ARC ルーティングコントロールを管理および操作するために必要なアクセス権がプランに付与されます。

ARC ルーティングコントロールと Route 53 ヘルスチェック実行ブロックの比較

リージョンスイッチの Amazon Route 53 ヘルスチェック実行ブロックは、DNS ベースのトラフィック管理の低コストな代替手段を提供します。ただし、この実行ブロックはアクティブ化する によって AWS リージョン 異なるため、リージョンが使用可能である必要があります。これは、正常なリージョンをアクティブ化しているため、ほとんどのお客様のニーズを満たすことができます。

ARC ルーティングコントロールは、信頼性の高い DNS ベースのトラフィック管理と 100% の可用性 SLA を提供します。ルーティングコントロールを使用すると、運用チームは安全ガードレールを使用してリージョン間でトラフィックをシフトできます。ルーティングコントロールは、100% の SLA を備えたシングルテナントソリューションを提供します。ルーティングコントロールクラスターは 5 つのリージョンに分散され、2 つのリージョンがオフラインになることを許容できます。非常に重要なアプリケーションがある場合は、ルーティングコントロールの使用を検討してください。

リージョンスイッチを使用するには、ルーティングコントロールは必要ありません。リージョンスイッチを使用して、ルーティングコントロールなしで Route 53 ヘルスチェック実行ブロックを使用してトラフィックリダイレクトを管理できます。

ルーティングコントロールは、次の状況でリージョンスイッチで値を追加します。

- トラフィック制御メカニズム自体には 100% の可用性 SLA が必要です。
- 組織では、重要なアプリケーションの安全ルールを使用した手動運用管理が必要です。

- 運用チームが必要に応じて自動トラフィックルーティングを手動で上書きできるように、defense-in-depthが必要です。

Route 53 ヘルスチェック実行ブロックはコントロールプレーンに依存しません。ヘルスチェックレコードの変更はデータプレーンを使用するため、設定の更新を処理するために リージョンをアクティブ化する必要はありません。以下の状況では、Route 53 ヘルスチェック実行ブロックで十分です。

- アプリケーションは、アクティブ化 AWS リージョン する によって異なります。
- 復旧ワークフローの一部としての自動トラフィックリダイレクトは、要件を満たします。
- コストの最適化が優先されます。Route 53 ヘルスチェック実行ブロックは、ルーティングコントロールよりもコストが低くなります。

ほとんどのお客様は、Route 53 ヘルスチェック実行ブロックをデフォルトのトラフィックルーティングメカニズムとして開始し、トラフィック管理メカニズムの信頼性が最も高い最も重要なアプリケーションにのみルーティングコントロールを追加します。

Amazon Aurora Global Database 実行ブロック

Amazon Aurora Global Database 実行ブロックを使用すると、グローバルデータベースのフェイルオーバーまたはスイッチオーバーリカバリワークフローを実行できます。

- フェイルオーバー — このアプローチを使用して、計画外のシステム停止から回復します。この方法では、Aurora Global Database 内のセカンダリ DB クラスターの 1 つにクロスリージョンフェイルオーバーを実行します。このアプローチの目標復旧時点 (RPO)は、通常、秒単位で測定される 0 以外の値です。データ損失の量は、障害発生 AWS リージョン 時の 全体の Aurora グローバルデータベースのレプリケーションラグによって異なります。詳細については、「Amazon Aurora ユーザーガイド」の「計画外の停止から Amazon Aurora Global Database を復旧する」を参照してください。
- スイッチオーバー — この操作は、以前は「マネージドプランニングフェイルオーバー」と呼ばれていました。このアプローチは、運用上のメンテナンスやその他の計画された運用上の手順など、すべての Aurora クラスターおよびこれらとやり取りする他のサービスが正常な状態にあることを確認する制御されたシナリオで使用します。この機能は、他の変更を行う前にセカンダリ DB クラスターとプライマリクラスターを同期するため、RPO は 0 (データの損失なし) になります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora Global Database のスイッチオーバーの実行」を参照してください。

設定

Aurora Global Database 実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Aurora Global Database 実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. Aurora Global Database クラスター名: グローバルデータベースの識別子を入力します。
4. リージョンのクラスター ARN: プランの各リージョンで使用するクラスター ARN を入力します。
5. Aurora データベースのオプションを指定する: 必要に応じて[スイッチオーバー] または [フェイルオーバー (データ損失)] を選択します。
6. Aurora Global Database クラスター名:
7. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

Aurora Global Databases 実行ブロックを設定することで、アプリケーション復旧の一環としてグローバルデータベースをフェイルオーバーまたはスイッチオーバーできます。アクティブ/アクティブアプローチを使用している場合、リージョン切り替えは他の設定済みリージョンをソースとして使用します。つまり、リージョンが非アクティブ化されている場合、リージョン切り替えは、スケールするパーセントに一致するソースとして他のアクティブなリージョンを使用します。

このブロックは、正常な実行モードと不正な実行モードの両方をサポートします。不適切な設定により Aurora Global Database のフェイルオーバーが実行され、データが失われる可能性があります。

フェイルオーバーやスイッチオーバーを含む Aurora Global Database のディザスタリカバリの詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora Global Database でのスイッチオーバーまたはフェイルオーバーの使用」を参照してください。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えは Aurora 実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、以下が正しいことを確認します。

- 設定で指定された Aurora グローバルクラスターが存在する。
- 送信元リージョンと送信先リージョンの両方に Aurora DB クラスターがある。
- ソース DB クラスターと宛先 DB クラスターは、グローバルデータベースのスイッチオーバーを許可する状態にある。
- 送信元クラスターと送信先クラスターの両方に DB インスタンスがある。
- スイッチオーバーアクションのグローバルクラスターエンジンのバージョンに互換性がある。これには、クラスターが同じメジャーバージョン、マイナーバージョン、パッチバージョンにあることの検証が含まれますが、Aurora ドキュメントに記載されているいくつかの例外があります。

リージョン切り替えは、プランの IAM ロールに Aurora フェイルオーバーとスイッチオーバーに必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

Aurora 実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップが実行されている間に、Aurora を管理および操作するために必要なアクセス権がプランに付与されます。

Amazon DocumentDB グローバルクラスター実行ブロック

Amazon DocumentDB グローバルクラスター実行ブロックを使用すると、グローバルクラスターのフェイルオーバーまたはスイッチオーバーリカバリワークフローを実行できます。

- フェイルオーバー — このアプローチを使用して、計画外のシステム停止から回復します。このアプローチでは、Amazon DocumentDB グローバルクラスター内のセカンダリクラスターの 1 つへのクロスリージョンフェイルオーバーを実行します。このアプローチの目標復旧時点 (RPO)は、通常、秒単位で測定される 0 以外の値です。データ損失の量は、障害発生 AWS リージョン 時の全体の Amazon DocumentDB グローバルクラスターレプリケーションの遅延によって異なります。

- スイッチオーバー – このアプローチは、すべての Amazon DocumentDB クラスターが正常な状態にある運用メンテナンスやその他の計画された運用手順など、制御されたシナリオに使用します。この機能は、他の変更を行う前にセカンダリクラスターをプライマリと同期するため、RPO は 0 (データ損失なし) です。

設定

Amazon DocumentDB グローバルクラスター実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Amazon DocumentDB グローバルクラスター実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. Amazon DocumentDB グローバルクラスター識別子: グローバルクラスターの識別子を入力します。
4. リージョンのクラスター ARN: プランの各リージョンで使用するクラスター ARN を入力します。
5. Amazon DocumentDB クラスターの オプションを指定します。スイッチオーバーまたはフェイルオーバー (データ損失) を選択します。
6. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

Amazon DocumentDB グローバルクラスター実行ブロックを設定することで、アプリケーション復旧の一環としてグローバルクラスターをフェイルオーバーまたはスイッチオーバーできます。アクティブ/アクティブアプローチを使用している場合、リージョンスイッチは他の設定済みリージョンをソースとして使用します。つまり、リージョンが非アクティブ化されている場合、リージョン切り替えは、スケールするパーセントに一致するソースとして他のアクティブなリージョンを使用します。

このブロックは、正常な実行モードと不正な実行モードの両方をサポートします。不正な設定により Amazon DocumentDB グローバルクラスターのフェイルオーバーが実行され、データが失われる可能性があります。

スイッチオーバーまたはフェイルオーバーオペレーション中に、顧客が書き込みに使用する DNS エンドポイントが変更されます。お客様は、オペレーションの完了後に正しいエンドポイントを使用していることを確認する責任があります。

プラン評価の一環として評価されるもの

リージョンスイッチがプランを評価すると、リージョンスイッチは Amazon DocumentDB 実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、以下が正しいことを確認します。

- 設定で指定された Amazon DocumentDB グローバルクラスターが存在します。
- 送信元リージョンと送信先リージョンの両方に Amazon DocumentDB クラスターがあります。
- 送信元クラスターと送信先クラスターは使用可能な状態です。
- 送信元クラスターと送信先クラスターの両方にインスタンスがあります。
- グローバルクラスターエンジンのバージョンには互換性があります。

リージョンスイッチは、プランの IAM ロールに Amazon DocumentDB フェイルオーバーとスイッチオーバーに必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

Amazon DocumentDB 実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップが実行されている間に、プランが Amazon DocumentDB を管理および操作するために必要なアクセス権を持つようになります。

Amazon RDS Promote リードレプリカ実行ブロック

Amazon RDS Promote リードレプリカ実行ブロックを使用すると、マルチリージョンリカバリプロセスの一環として、Amazon RDS リードレプリカをスタンドアロンデータベースインスタンスに昇格させることができます。これにより、そのリージョンのリードレプリカを新しいプライマリデータベースに昇格させることで、正常なリージョンにフェイルオーバーできます。

設定

Amazon RDS Promote リードレプリカ実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Amazon RDS 実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョンの RDS DB インスタンス ARN: プランの各リージョンのリードレプリカのデータベースインスタンス ARN を入力します。
4. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

Amazon RDS Promote リードレプリカ実行ブロックを設定することで、アプリケーション復旧の一環として、リードレプリカをスタンドアロンデータベースインスタンスに昇格させることができます。プランを実行すると、リージョンスイッチは、アクティブ化するリージョンのリードレプリカを昇格させて、独立したデータベースインスタンスにします。

> **Note**
>
> このブロックはアクティブ/パッシブプランのみをサポートします

昇格中、データベースへの接続に使用する DNS エンドポイントは変わりません。ただし、昇格されたインスタンスは元のプライマリデータベースからレプリケートされなくなります。オペレーションが完了したら、アプリケーションが正しいエンドポイントを使用するように設定されていることを確認する責任があります。

昇格後、昇格されたインスタンスは元のプライマリインスタンスから次のバックアップ設定を継承します。

- バックアップの保存期間
- 優先バックアップウィンドウ

プラン評価の一環として評価されるもの

リージョンスイッチがプランを評価すると、リージョンスイッチは Amazon RDS 実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、以下が正しいことを確認します。

- 設定で指定された Amazon RDS データベースインスタンスが存在します。
- プライマリリージョン以外のデータベースインスタンスはリードレプリカです。
- リードレプリカは使用可能な状態です。
- データベースインスタンスは、クロスリージョンレプリケーション用に適切に設定されています。

リージョンスイッチは、プランの IAM ロールに Amazon RDS リードレプリカの昇格に必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

Amazon RDS 実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップを実行するときに、プランに Amazon RDS を管理および操作するために必要なアクセス権が付与されます。

Amazon RDS クロスリージョンレプリカの作成実行ブロック

Amazon RDS クロスリージョンレプリカの作成実行ブロックを使用すると、リカバリ後プロセスの一環として Amazon RDS データベースインスタンスのクロスリージョンリードレプリカを作成できます。この実行ブロックは通常、リードレプリカを昇格させてクロスリージョンレプリケーションを再確立した後に使用され、アプリケーションが将来のリージョンイベントに備えられるようにします。

設定

Amazon RDS クロスリージョンレプリカの作成実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Amazon RDS 実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョンのソース DB インスタンス ARN: プランの各リージョンのソースデータベースのデータベースインスタンス ARN を入力します。実行ブロックは、アクティブ化されているリージョンの識別子をソースデータベースとして使用して、クロスリージョンリードレプリカを作成します。
4. レプリカ DB インスタンス ARN: 新しいリードレプリカに使用するインスタンス ARN を入力します。
5. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

Amazon RDS クロスリージョンレプリカの作成実行ブロックを設定することで、復旧後プロセスの一環として、他のリージョンにリードレプリカを作成できます。この実行ブロックは、フェイルオーバーが成功した後に実行され、クロスリージョンレプリケーションを再確立するように設計されています。

このブロックは、アクティブ/パッシブプランにのみ追加できます。

実行中、古いプライマリインスタンスの名前が変更され、renamedByRegionSwitch でタグ付けされます。次に、古いプライマリから次の設定をコピーした新しいリードレプリカインスタンスが作成されます。

- インスタンス識別子
- DB パラメータグループ
- DB サブネットグループ
- KMS キー

- VPC セキュリティグループ
- オプショングループ
- マルチ AZ 設定
- ドメイン認証シークレット ARN

> **Important**
>
> 名前が変更されたプライマリインスタンスは実行中のままで、引き続き料金が発生します。リージョンスイッチは識別のために renamedByRegionSwitch でタグ付けしますが、それ以外の場合は変更または削除しません。変更したインスタンスは、運用要件とコスト要件に基づいて、実行を維持するか、停止するか、削除するかを決定するなど、管理する必要があります。

> **Note**
>
> この実行ブロックは復旧後のワークフロー用に設計されており、ソースリージョンが正常でアクセス可能である必要があります。これは、フェイルオーバーが成功した後にクロスリージョンレプリケーションを再確立するために使用します。

プラン評価の一環として評価されるもの

リージョンスイッチがプランを評価すると、リージョンスイッチは Amazon RDS 実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、以下が正しいことを確認します。

- 設定のデータベースインスタンス ARNs は有効で、正しくフォーマットされています。
- ソースデータベースインスタンスは、それぞれのリージョンに存在します。
- ソースデータベースインスタンスは使用可能な状態です。

リージョンスイッチは、プランの IAM ロールに Amazon RDS リードレプリカの作成に必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

Amazon RDS 実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップを実行するときに、プランに Amazon RDS を管理および操作するために必要なアクセス権が付与されます。

手動承認実行ブロック

手動承認実行ブロックを使用すると、IAM ロールに関連付ける承認ステップを挿入できます。ロールにアクセスできるユーザーは、ステップの実行を承認または拒否したり、承認が付与されるまでステップを一時停止したり、プランの進行を妨げたりすることができます。

プランの実行中に手動承認が必要であることを確認するには、ワークフロー内の特定の場所で手動承認ステップを入力し、ステップを承認できるユーザーを指定するよう IAM ロールを設定します。

設定

手動承認実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「手動承認実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. IAM 承認ロール: リージョン切り替えプランの実行継続を手動で承認するアクセス許可を持つ IAM ロールの ARN を入力します。IAM ロールは、プランの所有者であるアカウント内にある必要があります。
4. タイムアウト: タイムアウト値を入力します。

次に、[ステップを保存] を選択します。

仕組み

手動承認実行ブロックを設定することで、アプリケーションの復旧の一環として承認を要求できます。手動実行ブロックの場合、リージョン切り替えは以下を実行します。

- リージョン切り替えが手動実行ブロックを実行すると、実行を一時停止し、プランの実行ステータスを承認待ちに設定します。
- 実行ブロックで定義されたロールにアクセスできるユーザーは、ステップの実行を承認または拒否できます。
- ステップの実行を承認すると、リージョン切り替えはプランの実行に進みます。拒否した場合、リージョン切り替えはプランの実行をキャンセルします。

このブロックは、不正な実行モードをサポートしていません。

プラン評価の一環として評価されるもの

リージョン切り替えは、手動承認実行ブロックの評価を完了しません。

カスタムアクション Lambda 実行ブロック

カスタムアクションの Lambda 実行ブロックを使用すると、Lambda 関数を使用してカスタマイズされたステップをプランに追加できます。

設定

Lambda 実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「カスタムアクション Lambda 実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. リージョンをアクティブ化または非アクティブ化するときに呼び出される Lambda 関数 ARN: このステップで実行する Lambda 関数の ARN を指定します。
4. Lambda 関数を実行するリージョン: ドロップダウンメニューで、Lambda 関数を実行するリージョンを選択します。
5. タイムアウト: タイムアウト値を入力します。

6. 再試行間隔: 再試行間隔を入力して、この間隔内に成功しなかった場合に Lambda 関数を再実行します。

次に、[ステップを保存] を選択します。

仕組み

- カスタムアクション Lambda 実行ブロックを作成する場合、実行するステップに 2 つの Lambda 関数を指定する必要があります。1 つはプランのリージョンごとに指定します。
- Lambda を実行するリージョンを設定できます。例えば、アクティブ化するリージョンや非アクティブ化するリージョンなどです。ただし、非アクティブ化されたリージョンで を実行する場合は、そのリージョンに依存します。非アクティブ化するリージョンに依存することはお勧めしません。

このブロックは、正常な実行モードと不正な実行モードの両方をサポートします。不正な実行モードでは、リージョン切り替えは Lambda 実行ブロックステップをスキップします。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えは Lambda 実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、以下が正しいことを確認します。

- 設定で指定された Lambda 関数が存在します。
- Lambda 関数の同時実行設定は、以下を検証することを含め、スロットリングされません。
  - 同時実行が 0 に設定されない。
  - 少なくとも 1 つの同時実行が使用可能であるか、予約されていない同時実行が存在する。

リージョン切り替えは、Lambda 関数のドライランを実行して、実際の関数ロジックを実行せずに、指定されたパラメータとアクセス許可を検証します。ドライランを実行すると、標準の Lambda コストが発生します。

リージョン切り替えは、プランの IAM ロールに Lambda 実行に必要なアクセス許可があることも検証します。リージョン切り替え実行ブロックに必要なアクセス許可の詳細については、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

Lambda 実行ブロックを適切に機能させるには、正しい IAM アクセス許可が不可欠です。これらの検証のいずれかが失敗した場合、リージョン切り替えは問題があることを示す警告を返し、アクセス

許可または設定の問題を解決するのに役立つ特定のエラーメッセージを提供します。これにより、プランの実行中にこのステップが実行されている間に、Lambda を管理および操作するために必要なアクセス権がプランに付与されます。

Amazon Route 53 ヘルスチェック実行ブロック

Amazon Route 53 ヘルスチェック実行ブロックを使用すると、フェイルオーバー中にアプリケーションのトラフィックがリダイレクトされる先のリージョンを指定できます。実行ブロックは Amazon Route 53 ヘルスチェックを作成し、アカウントの Route 53 DNS レコードにアタッチします。リージョン切り替えプランを実行すると、Route 53 ヘルスチェックの状態が更新され、DNS 設定に基づいてトラフィックがリダイレクトされます。

> **Important**
>
> Route 53 ホストゾーンは、リージョン切り替えプランと同じパーティションにある必要があります。

設定

Route 53 ヘルスチェック実行ブロックを設定するには、次の値を入力します。

> **Important**
>
> 実行ブロックを設定する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「Route 53 ヘルスチェック実行ブロックのサンプルポリシー」を参照してください。

1. ステップ名: 名前を入力します。
2. ステップの説明 (オプション): ステップの説明を入力します。
3. ホストゾーン ID: Route 53 のドメインと DNS レコードのホストゾーン ID。
4. レコード名: 使用するレコードのレコード名 (ドメイン名) と、関連するヘルスチェックを入力して、アプリケーションのトラフィックをリダイレクトします。リージョン切り替えは、レコード名の Route 53 レコードセットを見つけ、レコードセットの [値] または [セット識別子] 内のリージョン名に基づいて、各レコードセットをリージョンにマッピングしようとします。
5. レコードセット識別子 (オプション): プランの作成後にステップ 4 で指定したレコード名からリージョン切り替えがレコードセットを自動的にリージョンにマッピングできない場合、レコード

セット識別子を手動で指定できます。プラン評価で詳細情報が必要であることを示す警告が返された場合は、リージョンごとに以下を含めて、レコードセット識別子でプランを更新します。

- レコードセット識別子: レコードセットの [セット識別子] または [値/トラフィックのルーティング先] を入力します。
- リージョン: レコードセット識別子情報を持つレコードセットに関連付けられたリージョンを入力します。

6. [保存ステップ] を選択します。
7. Route 53 でヘルスチェックを設定します。

   リージョン切り替えは、実行ブロックで定義されたホストゾーン内の各レコード名について、リージョンごとにヘルスチェック ID を提供します。Route 53 のアカウント内の対応するレコードセットのヘルスチェックを設定し、プランの実行中にリージョン切り替えがアプリケーションのトラフィックを正しくリダイレクトできるようにします。プランの詳細ページの [ヘルスチェック] タブで、すべての実行ブロックとリージョンのヘルスチェックを表示できます。

仕組み

リージョン切り替えワークフローにヘルスチェックステップを追加して、アクティブ/パッシブ設定の場合はセカンダリリージョンに、アクティブ/アクティブ設定の場合は非アクティブリージョンからトラフィックをリダイレクトできるようにします。プランに複数のワークフローを追加する場合は、同じ DNS レコードを使用するすべてのヘルスチェック実行ブロックに同じ設定値を指定します。

リージョン切り替えは、実行ブロックの設定時に指定した情報に基づいて、プラン内の各リージョンの正しいレコードセットを決定しようとします。通常、ホストゾーン ID とレコード名は、レコードセットと関連するリージョンを決定するのに十分な情報です。そうでない場合、プランの作成後にリージョン切り替えが自動プラン評価を実行すると、詳細情報が必要であることを知らせる警告が返されます。

リージョン切り替えは、Route 53 ヘルスチェック実行ブロックごとにヘルスチェックを提供します。アクティブ/パッシブリカバリアプローチを使用するプランの場合、プライマリリージョンのヘルスチェックは正常として開始され、スタンバイリージョンのヘルスチェックは最初は異常に設定されます。アクティブ/アクティブリカバリアプローチを使用するプランの場合、すべてのリージョンのヘルスチェックは正常な状態で開始されます。

リージョン切り替えを有効にしてプランに対してこの実行ブロックを正常に実行するには、ヘルスチェックを DNS レコードに追加する必要があります。

アクティブ/アクティブプランの場合、実行ステップは次の方法で機能します。

- リージョンに対して非アクティブ化ワークフローが実行されると、ヘルスチェックは異常に設定され、トラフィックはリージョンに転送されなくなります。
- リージョンに対してアクティブ化ワークフローを実行すると、ヘルスチェックは正常に設定され、トラフィックはリージョンにルーティングされます。

アクティブ/パッシブプランの場合、実行ステップは次の方法で機能します。

- リージョンに対してアクティブ化ワークフローを実行すると、そのリージョンのヘルスチェックは正常に設定され、トラフィックはリージョンにルーティングされます。同時に、プラン内の他のリージョンのヘルスチェックは非正常に設定され、トラフィックはそのリージョンに転送されなくなります。

プラン評価の一環として評価されるもの

リージョン切り替えがプランを評価すると、リージョン切り替えは Route 53 ヘルスチェック実行ブロックの設定とアクセス許可に対していくつかのチェックを実行します。リージョン切り替えは、ヘルスチェックが実行ブロック設定で指定された DNS レコードにアタッチされていることを確認します。つまり、リージョン切り替えは、特定の AWS リージョン の DNS レコードが、そのリージョンのヘルスチェックを使用するように設定されていることを確認します。

ARC ルーティングコントロールと Route 53 ヘルスチェック実行ブロックの比較

リージョンスイッチの Amazon Route 53 ヘルスチェック実行ブロックは、DNS ベースのトラフィック管理の低コストな代替手段を提供します。ただし、この実行ブロックはアクティブ化する に依存する AWS リージョン ため、リージョンが使用可能である必要があります。これは、正常なリージョンをアクティブ化しているため、ほとんどのお客様のニーズを満たすことができます。

ARC ルーティングコントロールは、信頼性の高い DNS ベースのトラフィック管理と 100% の可用性 SLA を提供します。ルーティングコントロールを使用すると、運用チームは安全ガードレールを使用してリージョン間でトラフィックをシフトできます。ルーティングコントロールは、100% の SLA を持つシングルテナントソリューションを提供します。ルーティングコントロールクラスターは 5 つのリージョンに分散され、2 つのリージョンがオフラインになることを許容できます。非常に重要なアプリケーションがある場合は、ルーティングコントロールの使用を検討してください。

リージョンスイッチを使用するには、ルーティングコントロールは必要ありません。リージョンスイッチを使用して、ルーティングコントロールなしで Route 53 ヘルスチェック実行ブロックを使用してトラフィックリダイレクトを管理できます。

ルーティングコントロールは、以下の状況でリージョンスイッチで値を追加します。

- トラフィック制御メカニズム自体には 100% の可用性 SLA が必要です。
- 組織には、重要なアプリケーションの安全ルールを使用した手動操作コントロールが必要です。
- 運用チームが必要に応じて自動トラフィックルーティングを手動で上書きできるように、defense-in-depthが必要です。

Route 53 ヘルスチェック実行ブロックはコントロールプレーンに依存しません。ヘルスチェックレコードの変更はデータプレーンを使用するため、設定の更新を処理するために リージョンをアクティブ化する必要はありません。以下の状況では、Route 53 ヘルスチェック実行ブロックで十分です。

- アプリケーションは、アクティブ化 AWS リージョン する によって異なります。
- 復旧ワークフローの一部としての自動トラフィックリダイレクトは、要件を満たします。
- コストの最適化が優先されます。Route 53 ヘルスチェック実行ブロックは、ルーティングコントロールよりもコストが低くなります。

ほとんどのお客様は、Route 53 ヘルスチェック実行ブロックをデフォルトのトラフィックルーティングメカニズムとして開始し、トラフィック管理メカニズムに最も高い信頼性を必要とする最も重要なアプリケーションにのみルーティングコントロールを追加します。

子プランを作成する

より複雑な復旧シナリオをサポートするために、リージョン切り替えプラン実行ブロックで子プランを追加して子プランを作成できます。階層は 2 つのレベルに制限されていますが、1 つの親プランに複数の子プランを含めることができます。

> **Important**
>
> 子プランを作成する前に、正しい IAM ポリシーが設定されていることを確認してください。詳細については、「リージョン切り替えプラン実行ブロックのサンプルポリシー」を参照してください。

互換性のために、子プランは親プランがサポートするすべてのリージョンをサポートしている必要があります。さらに、アクティブ/アクティブまたはアクティブ/パッシブの復旧アプローチは、親プランと子プランで同じである必要があります。

子プランが、親プランおよび親プランシナリオに加えた変更に応答する次の方法に注意してください。

- 親実行ブロックは、その中のすべての子プランと他の実行ブロックが完了すると、完了としてマークされます。
- 子プランでいずれかのステップが失敗した場合、リージョン切り替えプランの実行ブロックは親プランで失敗します。
- 一時停止、正常な切り替えや不正な切り替え、キャンセルなど、リージョン切り替えステップ中に親プランで開始されたコントロールアクションは、子プランの現在のステップに関係なく、子プランで自動的に試行されます。
- スキップオペレーションには特別な動作があります。親プランはスキップされますが、子プランは引き続き実行されます。
- 子プランがリージョン切り替えブロックで既に実行されている場合、子プランが引き続き実行されているかどうかを判断するために、リージョン切り替えは子プランと親プランの互換性を評価します。子プランの設定が親プランの要件と一致する場合、リージョン切り替えは子プランを親プランによって開始されたものとして扱います。
- 子プランが次のような互換性のない設定パラメータで実行されている場合、親プランのステップは失敗します。
  - 子プランが別のリージョンで運用されている
  - リージョン切り替えがアクティブ化オペレーションの実行を予期している場合、子プランは非アクティブ化オペレーションを実行している
- 親プランが一時停止されている間に子プランが正常に完了すると、親プランが再開すると親プランは成功します。

リージョン切り替えプランのトリガーを作成する

リージョン切り替えでアプリケーションのリカバリを自動化したい場合は、リージョン切り替えプラン用に 1 つ以上のトリガーを作成できます。トリガーは、選択した CloudWatch アラーム条件に基づいて、リージョン切り替えプランの実行を自動的に開始します。

リージョン切り替えプランのトリガーを作成するには

1. プランを作成したら、[プランの詳細] ページで [トリガー] タブを選択します。
2. [トリガーの管理] を選択します。
3. 実行を自動化するワークフローを選択し、[トリガーの追加] を選択します。

4. トリガーの説明を入力します。
5. CloudWatch アラームを選択し、最大 10 個の CloudWatch アラームを選択してトリガーの条件を作成します。

   複数の条件を選択した場合、プランの自動実行を開始する前に、すべての条件が満たされている必要があります。

トリガーは、CloudWatch アラームがトリガーの条件を満たすように移行すると、計画の実行を開始します。トリガーがプランに追加されると、条件が既に満たされている場合、プランは実行されず、意図しないフェイルオーバーイベントが防止されます。

リージョン切り替えプランを実行してアプリケーションを復旧する

AWS リージョン に障害が発生したときにアプリケーションを復旧するには、Amazon Application Recovery Controller (ARC) でリージョンスイッチプランを実行します。

- アプリケーションがアクティブ/アクティブアプローチでデプロイされている場合、プランのワークフローは障害が発生したリージョンを非アクティブ化し、他のアクティブなリージョンが適切にスケーリングされ、すべてのアプリケーショントラフィックの受信を開始します。
- アプリケーションがアクティブ/パッシブアプローチでデプロイされている場合、プランのワークフローは、障害のあるリージョンを非アクティブ化し、スタンバイリージョンをアクティブ化します。必要に応じてそこでリソースをスケールアップし、アプリケーショントラフィックをスタンバイリージョンにリダイレクトします。

アプリケーション復旧を手動で実行するには、以下を実行してリージョン切り替えプランを実行します。

また、プランの実行を開始するように指定した特定の Amazon CloudWatch アラームを使用して自動的に実行をトリガーする方法もあります。プランを作成または更新するときに、プラン実行のトリガーを指定できます。詳細については、「リージョン切り替えプランのトリガーを作成する」を参照してください。

リージョン切り替えプランを実行するには

1. で AWS マネジメントコンソール、アプリケーションに対してアクティブ化 AWS リージョン する に移動します。
2. Amazon Application Recovery Controller (ARC) コンソールで、[リージョン切り替え] を選択し、実行するプランを選択します。

3. [プランの実行] を選択します。
4. プランに手動承認ステップが含まれている場合には、プロンプトが表示されたら各ステップを承認します。

プランの実行中に、実行の詳細ページで進捗状況を追跡できます。これは、プランの実行を選択すると開きます。

また、進行中のアプリケーションリカバリに関する情報は、リージョン切り替えダッシュボードで確認することもできます。リージョン切り替えコンソールの左側のナビゲーションの [リージョン切り替え] で、次のいずれかを選択します。

- グローバルダッシュボード
- リージョン名における実行

リージョンに障害がある場合は、グローバルダッシュボードにすべてのプランデータが表示されない可能性があることに注意してください。このため、運用イベント中はリージョン実行ダッシュボードのみを使用することをお勧めします。リージョン実行ダッシュボードは、ローカルのリージョン切り替えデータプレーンを使用するため、より回復力があります。

プランの実行が完了すると、[プラン実行の履歴] タブの [プランの詳細] ページで、プラン実行に関する情報と、リージョン切り替えによって実行された他のプランの情報を確認できます。

リージョン切り替えダッシュボード

リージョン切り替えには、組織やリージョン全体におけるリージョン切り替えプランの状態を確認できるグローバルダッシュボードが含まれています。リージョン切り替えには、現在 AWS マネジメントコンソールにログインしているリージョンのプラン実行のみを表示するリージョン実行ダッシュボードもあります。

リージョンに障害がある場合は、グローバルダッシュボードにすべてのプランデータが表示されない可能性があることに注意してください。このため、運用イベント中はリージョン実行ダッシュボードのみを使用することをお勧めします。リージョン実行ダッシュボードは、ローカルのリージョン切り替えデータプレーンを使用するため、より回復力があります。

リージョン切り替えグローバルダッシュボードを開くには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。

2. [リージョン切り替え] で、[グローバルダッシュボード] を選択します。

リージョン切り替えリージョンダッシュボードを開くには

1. https://console.aws.amazon.com/route53recovery/home#/dashboard で ARC コンソールを開きます。
2. [リージョン切り替え] で、[リージョンダッシュボード] を選択します。

リージョン切り替えでのクロスアカウントサポート

リージョン切り替えでは、他のアカウントのリソースをプランに追加できます。リージョン切り替えプランを他のアカウントと共有することもできます。詳細については、次のセクションを参照してください。

クロスアカウントのリソース

リージョン切り替えを使用すると、リージョン切り替えプランを含むアカウントとは別のアカウントでリソースをホストできます。リージョン切り替えは、プランを実行するときに executionRole を引き継ぎます。プランが、プランをホストするアカウントとは異なるアカウントのリソースを使用する場合、リージョン切り替えは executionRole を使用して crossAccountRole を引き継ぎ、それらのリソースにアクセスします。

リージョン切り替えプランの各リソースには、crossAccountRole と externalId の 2 つのオプションフィールドがあります。

- crossAccountRole: このロールは、リージョン切り替えプランをホストするアカウントとは異なるアカウントのリソースへのアクセスを許可します。ロールには、アカウント内のリソースを操作するためのアクセス許可のみが必要です。リージョン切り替えプランをホストするアカウントのリソースを操作するためのアクセス許可は必要ありません。
- ExternalId: これは、アクションを必要とするリソースを含むアカウントの信頼ポリシーからの STS 外部 ID です。これは、2 つのアカウント間で共有されるシークレットである英数字の文字列です。

リージョン切り替えプランの共有

リージョンスイッチは AWS Resource Access Manager (AWS RAM) と統合され、プランを共有できます AWS アカウント。プランを共有すると、指定したアカウントはプランの詳細を表示したり、プ

ランを実行したり、プランの実行を確認したりできます。これにより、さまざまなチーム間で復旧機能をより柔軟に制御できます。

リージョン切り替えでクロスアカウント共有を開始するには、 AWS RAMでリソース共有を作成します。リソース共有は、アカウントが所有するプランを共有する権限を持つ参加者を指定します。参加者は、コンソール、CLI、または AWS SDKs を使用して共有プランを表示および実行できます。

重要: は、共有するプランを所有している AWS アカウント 必要があります。自身が共有を受けているプランを他者に共有することはできません。組織または AWS Organizations内の組織部門とプランを共有するには、組織との共有を有効にする必要があります。

詳細については AWS RAM、「」を参照してくださいARC リージョン切り替えのアカウント間でのプランの共有をサポート。

ARC リージョン切り替えのアカウント間でのプランの共有をサポート

Amazon Application Recovery Controller (ARC) は と統合 AWS Resource Access Manager してリソース共有を有効にします。 AWS RAM は、他の AWS アカウント または を通じてリソースを共有できるサービスです AWS Organizations。ARC リージョン切り替えでは、リージョン切り替えプランを共有できます (プラン内の別のアカウントのリソースを使用するには、crossAccount ロールを使用します。詳細については、「クロスアカウントのリソース」を参照してください)。

では AWS RAM、リソース共有を作成して、所有しているリソースを共有します。リソース共有では、共有対象のリソースと、共有先である参加者を指定します。参加者には以下が含まれます。

- での所有者の組織 AWS アカウント 内外の特定 AWS Organizations
- の組織内の組織単位 AWS Organizations
- の組織全体 AWS Organizations

詳細については AWS RAM、AWS RAM 「 ユーザーガイド」を参照してください。

AWS Resource Access Manager を使用して ARC のアカウント間でプランを共有することで、1 つのプランを複数の異なるプランで使用できます AWS アカウント。プランの共有を選択すると、指定した他の AWS アカウント がプランを実行してアプリケーション復旧を実行できます。

AWS RAM は、 AWS お客様がリソースを安全に共有できるようにするサービスです AWS アカウント。を使用すると AWS RAM、IAM ロールとユーザーを使用して AWS Organizations、 の組織または組織単位 (OUs) 内のリソースを共有できます。 AWS RAM は、計画を共有するための一元的で制御された方法です。

プランを共有すると、組織が必要とするプランの総数を減らすことができます。共有プランを使用すると、プランを実行するための総コストを複数のチームに割り振ることが可能で、低コストで ARC の利点を最大化できます。アカウント間でプランを共有すると、複数のアプリケーションを ARC にオンボーディングするプロセスも簡単になります。特に、多数のアプリケーションが複数のアカウントや運用チームに分散している場合に有効です。

ARC でアカウント間共有を開始するには、 AWS RAM内にリソース共有を作成します。リソース共有は、アカウントが所有するプランを共有する権限を持つ参加者を指定します。

このトピックでは、所有しているリソースの共有方法と、共有されているリソースの使用方法を説明します。

内容

- プランを共有するための前提条件
- プランの共有
- 共有プランの共有解除
- 共有プランの特定
- 共有プランの責任とアクセス許可
- 費用請求
- クォータ

プランを共有するための前提条件

- プランを共有するには、 でそのプランを所有している必要があります AWS アカウント。つまり、自分のアカウントにそのリソースが割り当てられているか、プロビジョニングされている必要があります。自身が共有を受けているプランを他者に共有することはできません。
- 組織、または AWS Organizations内の組織単位とプランを共有するには、 AWS Organizations との共有を有効にする必要があります。詳細については、AWS RAM ユーザーガイドの「AWS Organizationsで共有を有効化する」を参照してください。

プランの共有

プランを共有すると、プランの共有を指定した参加者はそのプランを表示でき、追加のアクセス許可を付与した場合はそのプランを実行できます。

プランを共有するには、リソース共有にそのプランを追加する必要があります。リソース共有とは、AWS アカウント間で自身のリソースを共有するための AWS RAM リソースです。リソース共有で

は、共有対象のリソースと、共有先の参加者を指定します。プランを共有するには、新しいリソース共有を作成するか、リソースを既存のリソース共有に追加します。新しいリソース共有を作成するには、 AWS RAM コンソールを使用するか、 AWS Command Line Interface または AWS SDKsで AWS RAM API オペレーションを使用します。

の組織に属 AWS Organizations していて、組織内での共有が有効になっている場合、組織内の参加者には共有プランへのアクセス権が自動的に付与されます。それ以外の場合、参加者はリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有プランに対するアクセス許可が付与されます。

所有するプランを共有するには、 AWS RAM コンソールを使用するか、 AWS CLI または SDKs で AWS RAM API オペレーションを使用します。

AWS RAM コンソールを使用して所有するプランを共有するには

「AWS RAM ユーザーガイド」の「リソース共有の作成」を参照してください。

を使用して所有しているプランを共有するには AWS CLI

create-resource-share コマンドを使用します。

プランを共有するアクセス許可の付与

アカウント間でプランを共有するには、 AWS RAMを使用してプランを共有する IAM プリンシパルに次の追加のアクセス許可が必要です。

```
# read and execute plan permissions
"arc-region-switch:GetPlan",
"arc-region-switch:GetPlanInRegion",
"arc-region-switch:GetPlanExecution",
"arc-region-switch:ListPlanExecutionEvents",
"arc-region-switch:ListPlanExecutions",
"arc-region-switch:ListRoute53HealthChecks",
"arc-region-switch:GetPlanEvaluationStatus",
"arc-region-switch:StartPlanExecution",
"arc-region-switch:CancelPlanExecution",
"arc-region-switch:UpdatePlanExecution",
"arc-region-switch:UpdatePlanExecutionStep"
```

プランを共有する所有者には、次のアクセス許可が必要です。これらのアクセス許可 AWS RAM なしで を通じてプランを共有しようとすると、エラーが返されます。

```
"arc-region-switch:PutResourcePolicy" # Permission only apis
"arc-region-switch:DeleteResourcePolicy" # Permission only apis
"arc-region-switch:GetResourcePolicy" # Permission only apis
```

IAM AWS Resource Access Manager の使用方法の詳細については、「 AWS RAM ユーザーガイド」の「IAM AWS Resource Access Manager の使用方法」を参照してください。

共有プランの共有解除

プランの共有を解除すると、次のことが参加者と所有者に適用されます。

- 参加者は、共有されていないプランを表示または実行できなくなります。

所有している共有プランの共有を解除するには、それをリソース共有から削除します。これを行うには、 AWS RAM コンソールを使用するか、 AWS CLI または SDKs で AWS RAM API オペレーションを使用します。

AWS RAM コンソールを使用して所有している共有プランの共有を解除するには

AWS RAM ユーザーガイド の「リソース共有の更新」を参照してください。

を使用して所有している共有プランの共有を解除するには AWS CLI

disassociate-resource-share コマンドを使用します。

共有プランの特定

所有者と参加者は、 AWS RAM内で情報を表示して、共有プランを識別できます。ARC コンソールと AWS CLIを使用して、共有リソースに関する情報を取得することもできます。

一般的に、共有したリソースまたは共有されたリソースの詳細については、 AWS Resource Access Manager 「 ユーザーガイド」の情報を参照してください。

- 所有者は、 AWS RAMを使用することで、他のユーザーと共有しているすべてのリソースを表示できます。詳細については、「 での共有リソースの表示 AWS RAM」を参照してください。
- 参加者として、 を使用して共有されているすべてのリソースを表示できます AWS RAM。詳細については、「 での共有リソースの表示 AWS RAM」を参照してください。

所有者は、 で情報を表示する AWS マネジメントコンソール か、ARC API オペレーション AWS Command Line Interface で を使用してプランを共有するかどうかを判断できます。

コンソールを使用して、所有しているプランが共有されているかどうかを確認するには

で AWS マネジメントコンソール、プランの詳細ページで、プランの共有ステータスを確認します。

参加者がプランの共有を受ける際は、通常、プランにアクセスできるよう共有を承認する必要があります。。

共有プランの責任とアクセス許可

所有者のアクセス許可

参加者はプランを表示または実行できます (適切なアクセス許可がある場合) 。

参加者のアクセス許可

所有しているプランを他のユーザーと共有すると AWS アカウント、参加者はプランを表示または実行できます (適切なアクセス許可がある場合) 。

を使用してプランを共有すると AWS RAM、参加者はデフォルトで読み取り専用のアクセス許可を持ちます。リージョン切り替えの読み取り専用アクセス許可のリストを確認するには、「読み取り専用のアクセス許可」を参照してください。参加者は、リージョン切り替えプランを実行するための追加のアクセス許可が必要です。プランを実行する必要がある参加者には、追加のアクセス許可が必要です。以下のオペレーションでは、 AWS RAM 参加者にアクセス許可を付与できないことに注意してください。

- `ApprovePlanExecutionStep`
- `UpdatePlan`

費用請求

ARC のプランの所有者には、そのプランに関連する費用が請求されます。プランの所有者側でも参加者側でも、プランでホストされるリソースの作成に追加費用はかかりません。

詳細な料金情報と例については、「Amazon Application Recovery Controller (ARC) の料金」を参照してください。

クォータ

共有プランで作成されたすべてのリソースは、プラン所有者のクォータにカウントされます。

リージョン切り替えプランのクォータのリストについては、「リージョン切り替えのクォータ」を参照してください。

ARC でのリージョン切り替えの Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

内容

- ARC のリージョン切り替えと IAM の連携方法
- ARC でのリージョン切り替えのアイデンティティベースのポリシーの例

ARC のリージョン切り替えと IAM の連携方法

IAM を使用して ARC へのアクセスを管理する前に、ARC で利用できる IAM の機能について学びます。

IAM を使用して Amazon Application Recovery Controller (ARC) のリージョン切り替えへのアクセスを管理する前に、リージョン切り替えで使用できる IAM 機能について説明します。

Amazon Application Recovery Controller (ARC) でリージョン切り替えにより使用できる IAM の機能

IAM 機能	リージョン切り替えのサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	はい
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	はい
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	あり
プリンシパルアクセス権限	あり

IAM 機能	リージョン切り替えのサポート
サービスロール	いいえ
サービスリンクロール	いいえ

AWS サービスがほとんどの IAM 機能と連携する方法の概要を把握するには、「IAM ユーザーガイド」のAWS 「IAM と連携する のサービス」を参照してください。

リージョン切り替えのアイデンティティベースポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

ARC のアイデンティティベースのポリシーの例を表示するには、「Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例」を参照してください。

リージョン切り替えのリソースベースポリシー

リソースベースのポリシーのサポート: あり

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシー や Amazon S3 バケットポリシー があげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。

リージョン切り替えのポリシーアクション

ポリシーアクションのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

リージョン切り替えの ARC のポリシーアクションでは、アクションの前に次のプレフィックスが使用されます。

```
arc-region-switch
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。例えば、次のようになります。

```
"Action": [
      "arc-region-switch:action1",
      "arc-region-switch:action2"
         ]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Describe という単語で始まるすべてのアクションを指定するには次のアクションを含めます。

```
"Action": "arc-region-switch:Describe*"
```

リージョン切り替えの ARC アイデンティティベースポリシーの例を表示するには、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

リージョン切り替えのポリシーリソース

ポリシーリソースのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベ

ルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

リージョン切り替えの ARC アイデンティティベースポリシーの例を表示するには、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

リージョン切り替えのポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

リージョン切り替えの ARC アイデンティティベースポリシーの例を表示するには、「ARC でのリージョン切り替えのアイデンティティベースのポリシーの例」を参照してください。

リージョン切り替えのアクセスコントロールリスト (ACL)

ACL のサポート: あり

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

リージョン切り替えでの属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: あり

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、aws:ResourceTag/*key-name*、aws:RequestTag/*key-name*、または aws:TagKeys の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

リージョン切り替えでの一時的な認証情報の使用

一時的な認証情報のサポート: あり

一時的な認証情報は、 AWS リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。 AWS では、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

リージョン切り替えのクロスサービスプリンシパル許可

転送アクセスセッション (FAS) のサポート: あり

IAM エンティティ (ユーザーまたはロール) を使用して でアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。

リージョン切り替えのサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

リージョン切り替えのサービスにリンクされたロール

サービスにリンクされたロールのサポート: なし

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の「サービスリンクロール」列に Yes と記載されたサービスを見つけます。サービスにリンクされたロールに関するドキュメントをサービスで表示するには、[はい] リンクを選択します。

ARC でのリージョン切り替えのアイデンティティベースのポリシーの例

デフォルトでは、ユーザーおよびロールには、ARC リソースを作成または変更する権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認証リファレンス」の「Amazon Application Recovery Controller (ARC) のアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーに関するベストプラクティス
- プラン実行ロール信頼ポリシー
- フルアクセス許可
- 読み取り専用のアクセス許可
- 実行ブロックのアクセス許可
- アプリケーションヘルス許可の CloudWatch アラーム
- 自動計画実行レポートのアクセス許可
- クロスアカウントリソースのアクセス許可

- プラン実行ロールのアクセス許可を完了する

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが ARC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の 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。

プラン実行ロール信頼ポリシー

これは、ARC がリージョン切り替えプランを実行できるようにするために、プランの実行ロールに必要な信頼ポリシーです。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "arc-region-switch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

フルアクセス許可

次の IAM ポリシーは、すべてのリージョン切り替え API にフルアクセスを許可します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "arc-region-switch.amazonaws.com"
        }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:CreatePlan",
        "arc-region-switch:UpdatePlan",
        "arc-region-switch:GetPlan",
        "arc-region-switch:ListPlans",
        "arc-region-switch:DeletePlan",
        "arc-region-switch:GetPlanInRegion",
        "arc-region-switch:ListPlansInRegion",
        "arc-region-switch:ApprovePlanExecutionStep",
        "arc-region-switch:GetPlanEvaluationStatus",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:StartPlanExecution",
        "arc-region-switch:CancelPlanExecution",
        "arc-region-switch:ListRoute53HealthChecks",
        "arc-region-switch:ListRoute53HealthChecksInRegion",
        "arc-region-switch:ListPlanExecutions",
        "arc-region-switch:ListPlanExecutionEvents",
        "arc-region-switch:ListTagsForResource",
        "arc-region-switch:TagResource",
        "arc-region-switch:UntagResource",
        "arc-region-switch:UpdatePlanExecution",
        "arc-region-switch:UpdatePlanExecutionStep"
      ],
      "Resource": "*"
    }
  ]
}
```

読み取り専用のアクセス許可

次の IAM ポリシーは、リージョン切り替えの読み取り専用アクセス許可を付与します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:GetPlan",
        "arc-region-switch:ListPlans",
        "arc-region-switch:GetPlanInRegion",
        "arc-region-switch:ListPlansInRegion",
        "arc-region-switch:GetPlanEvaluationStatus",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:ListRoute53HealthChecks",
        "arc-region-switch:ListRoute53HealthChecksInRegion",
        "arc-region-switch:ListPlanExecutions",
        "arc-region-switch:ListPlanExecutionEvents",
        "arc-region-switch:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

実行ブロックのアクセス許可

以下のセクションでは、リージョン切り替えプランに追加する特定の実行ブロックに必要なアクセス許可を提供する IAM ポリシーの例を示します。

内容

- EC2 Auto Scaling 実行ブロックのサンプルポリシー
- Amazon EKS リソーススケーリング実行ブロックサンプルポリシー
- Amazon ECS サービススケーリング実行ブロックのサンプルポリシー
- ARC ルーティングコントロール実行ブロックサンプルポリシー
- Aurora Global Database 実行ブロックのサンプルポリシー
- Amazon DocumentDB グローバルクラスター実行ブロックのサンプルポリシー
- Amazon RDS 実行ブロックのサンプルポリシー
- 手動承認実行ブロックのサンプルポリシー
- カスタムアクション Lambda 実行ブロックのサンプルポリシー
- Route 53 ヘルスチェック実行ブロックのサンプルポリシー
- リージョン切り替えプラン実行ブロックのサンプルポリシー

EC2 Auto Scaling 実行ブロックのサンプルポリシー

EC2 Auto Scaling グループのリージョンスイッチプランに実行ブロックを追加する場合にアタッチするポリシーの例を次に示します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:UpdateAutoScalingGroup"
      ],
      "Resource": [
        "arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:123d456e-123e-1111-abcd-
EXAMPLE22222:autoScalingGroupName/app-asg-primary",
        "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:1234a321-123e-1234-aabb-
EXAMPLE33333:autoScalingGroupName/app-asg-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon EKS リソーススケーリング実行ブロックサンプルポリシー

以下は、Amazon EKS リソーススケーリングのリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster"
      ],
      "Resource": [
        "arn:aws:eks:us-east-1:123456789012:cluster/app-eks-primary",
        "arn:aws:eks:us-west-2:123456789012:cluster/app-eks-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:ListAssociatedAccessPolicies"
      ],
      "Resource": [
        "arn:aws:eks:us-east-1:123456789012:access-entry/app-eks-primary/*",
        "arn:aws:eks:us-west-2:123456789012:access-entry/app-eks-secondary/*"
      ]
    }
  ]
}
```

注: この IAM ポリシーに加えて、プラン実行ロール
は、AmazonArcRegionSwitchScalingPolicyアクセスポリシーを使用して Amazon EKS クラスターのアクセスエントリに追加する必要があります。詳細については、「EKS アクセスエントリのアクセス許可を設定する」を参照してください。

Amazon ECS サービススケーリング実行ブロックのサンプルポリシー

以下は、Amazon ECS サービススケーリングのリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeServices",
        "ecs:UpdateService"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:123456789012:service/app-cluster-primary/app-
service",
        "arn:aws:ecs:us-west-2:123456789012:service/app-cluster-secondary/app-
service"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:123456789012:cluster/app-cluster-primary",
        "arn:aws:ecs:us-west-2:123456789012:cluster/app-cluster-secondary"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:ListServices"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "application-autoscaling:DescribeScalableTargets",
        "application-autoscaling:RegisterScalableTarget"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricStatistics"
      ],
      "Resource": "*"
    }
  ]
}
```

ARC ルーティングコントロール実行ブロックサンプルポリシー

注: Amazon ARC ルーティングコントロールの実行ブロックでは、プランの実行ロールに適用されるサービスコントロールポリシー (SCP) で、これらのサービスの次のリージョンへのアクセスを許可する必要があります。

- `route53-recovery-control-config: us-west-2`
- `route53-recovery-cluster: us-west-2, us-east-1, eu-west-1, ap-southeast-2, ap-northeast-1`

以下は、ARC ルーティングコントロールのリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53-recovery-control-config:DescribeControlPanel",
        "route53-recovery-control-config:DescribeCluster"
      ],
      "Resource": [
```

```
        "arn:aws:route53-recovery-control::123456789012:controlpanel/
abcd1234abcd1234abcd1234abcd1234",
        "arn:aws:route53-recovery-
control::123456789012:cluster/4b325d3b-0e28-4dcf-ba4a-EXAMPLE11111"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53-recovery-cluster:GetRoutingControlState",
        "route53-recovery-cluster:UpdateRoutingControlStates"
      ],
      "Resource": [
        "arn:aws:route53-recovery-
control::123456789012:controlpanel/1234567890abcdef1234567890abcdef/
routingcontrol/abcdef1234567890",
        "arn:aws:route53-recovery-
control::123456789012:controlpanel/1234567890abcdef1234567890abcdef/
routingcontrol/1234567890abcdef"
      ]
    }
  ]
}
```

CLI を使用して、ルーティングコントロールパネル ID とクラスター ID を取得できます。詳細については、「ルーティングコントロールコンポーネントの設定」を参照してください。

Aurora Global Database 実行ブロックのサンプルポリシー

以下は、Aurora データベースのリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeGlobalClusters",
        "rds:DescribeDBClusters"
```

```
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:FailoverGlobalCluster",
        "rds:SwitchoverGlobalCluster"
      ],
      "Resource": [
        "arn:aws:rds::123456789012:global-cluster:app-global-db",
       "arn:aws:rds:us-east-1:123456789012:cluster:app-db-primary",
        "arn:aws:rds:us-west-2:123456789012:cluster:app-db-secondary"
      ]
    }
  ]
}
```

Amazon DocumentDB グローバルクラスター実行ブロックのサンプルポリシー

以下は、Amazon DocumentDB グローバルクラスターのリージョンスイッチプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeGlobalClusters",
        "rds:DescribeDBClusters",
        "rds:FailoverGlobalCluster",
        "rds:SwitchoverGlobalCluster"
      ],
      "Resource": "*"
    }
  ]
}
```

Amazon RDS 実行ブロックのサンプルポリシー

以下は、Amazon RDS リードレプリカ昇格またはクロスリージョンレプリカ作成のリージョンスイッチプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:PromoteReadReplica",
        "rds:CreateDBInstanceReadReplica",
        "rds:ModifyDBInstance"
      ],
      "Resource": "*"
    }
  ]
}
```

手動承認実行ブロックのサンプルポリシー

以下は、手動承認のためにリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:ApprovePlanExecutionStep"
      ],
      "Resource": "arn:aws:arc-region-switch::123456789012:plan/sample-
plan:0123abc"
    }
  ]
}
```

カスタムアクション Lambda 実行ブロックのサンプルポリシー

以下は、Lambda 関数のリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lambda:GetFunction",
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:us-east-1:123456789012:function:app-recovery-primary",
        "arn:aws:lambda:us-west-2:123456789012:function:app-recovery-secondary"
      ]
    }
  ]
}
```

Route 53 ヘルスチェック実行ブロックのサンプルポリシー

以下は、Route 53 ヘルスチェックのリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "route53:ListResourceRecordSets"
      ],
      "Resource": [
```

```
        "arn:aws:route53:::hostedzone/Z1234567890ABCDEFGHIJ"
      ]
    }
  ]
}
```

リージョン切り替えプラン実行ブロックのサンプルポリシー

以下は、子プランを実行するリージョン切り替えプランに実行ブロックを追加する場合にアタッチするポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:StartPlanExecution",
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:CancelPlanExecution",
        "arc-region-switch:UpdatePlanExecution",
        "arc-region-switch:ListPlanExecutions"
      ],
      "Resource": [
        "arn:aws:arc-region-switch::123456789012:plan/child-plan-1/abcde1",
        "arn:aws:arc-region-switch::123456789012:plan/child-plan-2/fghij2"
      ]
    }
  ]
}
```

アプリケーションヘルス許可の CloudWatch アラーム

以下は、アプリケーションのヘルスに関する CloudWatch アラームにアクセスするためにアタッチするポリシーの例です。このアラームは、実際の復旧時間を決定するために使用されます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:app-health-primary",
        "arn:aws:cloudwatch:us-west-2:123456789012:alarm:app-health-secondary"
      ]
    }
  ]
}
```

自動計画実行レポートのアクセス許可

以下は、リージョン切り替えプランの自動レポート生成を設定する場合にアタッチするポリシーの例です。このポリシーには、Amazon S3 へのレポートの書き込み、CloudWatch アラームデータへのアクセス、親プランの子プラン情報の取得のためのアクセス許可が含まれています。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::your-bucket-name/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarms",
        "cloudwatch:DescribeAlarmHistory"
      ],
      "Resource": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:app-health-primary"
```

```
        "arn:aws:cloudwatch:us-west-2:123456789012:alarm:app-health-secondary"
      ],
    },
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:GetPlanExecution",
        "arc-region-switch:ListPlanExecutionEvents"
      ],
      "Resource": [
        "arn:aws:arc-region-switch:us-east-1:123456789012:plan/child-plan-1/abcde1",
        "arn:aws:arc-region-switch:us-west-2:123456789012:plan/child-plan-2/fghij2"
      ],
    }
  ]
}
```

注: Amazon S3 バケット暗号化用にカスタマーマネージド AWS KMS キーを設定する場合は、キーの kms:GenerateDataKeyおよび アクセスkms:Encrypt許可も追加する必要があります。

クロスアカウントリソースのアクセス許可

異なるアカウントにリソースがある場合は、クロスアカウントロールが必要です。クロスアカウントロールの信頼ポリシーの例を次に示します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/RegionSwitchExecutionRole"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
```

```
  ]
}
```

このクロスアカウントロールを引き受けるプラン実行ロールのアクセス許可は次のとおりです。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::987654321098:role/RegionSwitchCrossAccountRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

プラン実行ロールのアクセス許可を完了する

すべての実行ブロックのアクセス許可を含む包括的なポリシーを作成するには、かなり大きなポリシーが必要です。実際には、特定のプランで使用する実行ブロックのアクセス許可のみを含める必要があります。

以下は、プラン実行ロールポリシーの開始点として使用できるポリシーの例です。プランに含める特定の実行ブロックに必要なポリシーを追加してください。最小特権の原則に従って、プランで使用する特定の実行ブロックに必要なアクセス許可のみを含めます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "arn:aws:iam::123456789012:role/
RegionSwitchExecutionRole"
        },
        {
            "Effect": "Allow",
            "Action": [
                "arc-region-switch:GetPlan",
                "arc-region-switch:GetPlanExecution",
                "arc-region-switch:ListPlanExecutions"
            ],
            "Resource": "*"
        }
    ]
}
```

ARC でのリージョン切り替えのログ記録とモニタリング

Amazon CloudWatch、および Amazon EventBridge を使用して AWS CloudTrail、Amazon Application Recovery Controller (ARC) のリージョンスイッチをモニタリングし、アラートの取得、パターンの分析、問題のトラブルシューティングに役立てることができます。

トピック

- を使用したリージョンスイッチ API コールのログ記録 AWS CloudTrail
- Amazon EventBridge での ARC でのリージョン切り替えの使用

を使用したリージョンスイッチ API コールのログ記録 AWS CloudTrail

Amazon Application Recovery Controller (ARC) リージョンスイッチは AWS CloudTrail、ARC のユーザー、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されています。CloudTrail は、ARC のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、ARC コンソールのコールと、ARC API オペレーションへのコードのコールが含まれます。

証跡を作成する場合は、ARC のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、CloudTrail コンソールの [イベント履歴] で最新のイベントを表示できます。

CloudTrail が収集した情報を使用して、ARC に対して行われたリクエスト、リクエスト元の IP アドレス、リクエストした人、リクエストが行われた日時などの詳細を確認できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail での ARC 情報

CloudTrail は、アカウントの作成 AWS アカウント 時に で有効になります。ARC でアクティビティが発生すると、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「CloudTrail イベント履歴の操作」を参照してください。

ARC のイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成するときに、証跡がすべての AWS リージョンに適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集されたイベントデータをさらに分析して処理するように、他の AWS サービスを設定できます。詳細については、次を参照してください:

- 追跡を作成するための概要
- CloudTrail がサポートされているサービスと統合
- 「CloudTrail の Amazon SNS 通知の設定」
- CloudTrail ログファイルを複数のリージョンから受け取る、複数のアカウントから CloudTrail ログファイルを受け取る

すべての ARC アクションは CloudTrail によって記録され、「TBD API REFERENCE LINK」に文書化されています。例えば、TBD、TBD、TBD の各アクションを呼び出すと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity エレメント」を参照してください。

イベント履歴でのリージョン切り替えイベントの表示

CloudTrail では、[イベント履歴] に最近のイベントが表示されます。リージョン切り替え API リクエストのほとんどのイベントは、リージョン切り替えプランを操作するリージョンにあります。例えば、プランを作成したり、プランを実行したりします。ただし、ARC コンソールで実行する一部のリージョン切り替えアクションは、データプレーンオペレーションではなく、コントロールプラン API オペレーションを使用して行われます。コントロールプレーンオペレーションの場合、米国東部 (バージニア北部) のイベントを表示します。コントロールプレーンオペレーションである API コールについては、「リージョン切り替え API オペレーション」を参照してください。

ARC ログファイルエントリの概要

「トレイル」は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API 呼び出しの順序付けられたスタックトレースではないため、特定の順序では表示されません。

次の例は、リージョン切り替えの `StartPlanExecution` アクションを実行する CloudTrail ログエントリです。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "A1B2C3D4E5F6G7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/admin",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROA33L3W36EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/admin",
                "accountId": "111122223333",
                "userName": "EXAMPLENAME"
            },
            "attributes": {
                "mfaAuthenticated": "false",
```

```
                "creationDate": "2025-07-06T17:38:05Z"
            }
        }
    },
    "eventTime": "2025-07-06T18:08:03Z",
    "eventSource": "arc-region-switch.amazonaws.com",
    "eventName": "StartPlanExecution",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.50",
    "userAgent": "Boto3/1.17.101 Python/3.8.10 Linux/4.14.231-180.360.amzn2.x86_64
 exec-env/AWS_Lambda_python3.8 Botocore/1.20.102",
    "requestParameters": {
        "planArn": "arn:aws:arc-region-switch::555555555555:plan/
CloudTrailIntegTestPlan:bbbbb",
        "targetRegion": "us-east-1",
        "action": "activate"    }
    "responseElements": {
        "executionId": "us-east-1/ddddddddEXAMPLE",
        "plan": "arn:aws:arc-region-switch::555555555555:plan/
CloudTrailIntegTestPlan:bbbbb",
        "planVersion": "1",
        "activateRegion": "us-east-1"    },
    "requestID": "fd42dcf7-6446-41e9-b408-d096example",
    "eventID": "4b5c42df-1174-46c8-be99-d67aexample",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
      "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "us-east-1.arc.amazon.aws"
}
```


Amazon EventBridge での ARC でのリージョン切り替えの使用

Amazon EventBridge を使用すると、Amazon Application Recovery Controller (ARC) でリージョンスイッチリソースをモニタリングするイベント駆動型ルールを設定し、他の AWS サービスを使用するターゲットアクションを開始できます。例えば、リージョン切り替えプランの実行が完了するたびに Amazon SNS トピックをシグナリングすることで、E メール通知を送信するルールを設定できます。

Amazon EventBridge でルールを作成して、次の ARC リージョン切り替えイベントで反応させることが可能です。

- リージョン切り替えプランの実行。イベントは、リージョン切り替えプランが実施された (実行された) ことを示します。
- リージョン切り替えプランの評価。イベントは、リージョン切り替えプランの評価が完了したことを指定します。

関心のある特定の ARC イベントをキャプチャするには、そのイベント固有のパターンを定義し、イベントを検出する EventBridge がそれを照合できるようにします。イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

イベントは、ベストエフォートベースで出力されます。通常の運用状況下では、ARC から EventBridge にほぼリアルタイムで配信されます。ただし、イベントの配信を遅らせたり妨げたりする状況が発生する場合もあります。

EventBridge ルールがイベントパターンでどのように機能するかについては、「EventBridge のイベントとイベントパターン」を参照してください。

EventBridge を使用してリージョン切り替えリソースをモニタリングする

EventBridge でルールを作成すると、ARC がリージョン切り替えリソースに対してイベントを発生したときに実行されるアクションを定義できます。

EventBridge コンソールにイベントパターンを入力またはコピーアンドペーストするには、コンソールの [独自のものを入力] オプションを選択します。役に立つ可能性のあるイベントパターンを判断するために、このトピックにはリージョン切り替えパターンの例が含まれています。

リソースイベントのルールを作成するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. でルール AWS リージョン を作成するには、イベントをモニタリングするプランを作成したリージョンを選択します。
3. [Create rule] を選択します。
4. ルールの [Name (名前)] を入力し、必要に応じて説明を入力します。
5. [イベントバス] については、デフォルト値の [デフォルト] のままにします。

6. [次へ] を選択します。
7. [イベントパターンを構築] ステップでは、[イベントソース] はデフォルト値の [AWS イベント] のままにします。
8. [サンプルイベント] で [独自のサンプルイベントを入力] を選択します。
9. [サンプルイベント] には、イベントパターンを入力するか、コピーして貼り付けます。例については、次のセクションを参照ください。

リージョン切り替えパターンの例

イベントパターンは、一致するイベントと同じ構造をしています。イベントのパターンでは、照合する対象のフィールドを引用符で囲み、検出したい値を指定します。

このセクションのイベントパターンをコピーして EventBridge に貼り付け、ARC のアクションとリソースの監視に使用できるルールを作成できます。

以下のイベントパターンは、EventBridge で ARC のリージョン切り替え機能に使用できる例を示しています。

- PlanExecution のリージョン切り替えからすべてのイベントを選択します。

```
{
    "source": [ "aws.arc-region-switch" ],
    "detail-type": [ "ARC Region switch Plan Execution" ]
}
```

- PlanEvaluation のリージョン切り替えからすべてのイベントを選択します。

```
     {
     "source": [ "aws.arc-region-switch" ],
     "detail-type": [ "ARC Region Switch Plan Evaluation" ]
     }
```

以下は、リージョン切り替えプラン実行の ARC イベントの例です。

```
{
   "version": "0",
   "id": "1111111-bbbb-aaaa-cccc-dddddEXAMPLE", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
```

```
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/aaaaaExample"], #
 planArn
   "detail": {
    "version": "0.0.1",
    "eventType": "ExecutionStarted",
    "executionId": "bbbbbbEXAMPLE",
    "executionAction": "activating/deactivating {region}",
    "idempotencyKey": "1111111-2222-3333-4444-5555555555", # As there is a possibility
 of dual logging
   }
}
```

以下は、リージョン切り替えプランのステップレベル実行の ARC イベントの例です。

```
{
   "version": "0",
   "id": "1111111-bbbb-aaaa-cccc-dddddEXAMPLE", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/aaaaaExample"], #
 planArn
   "detail": {
    "version": "0.0.1",
    "eventType": "StepStarted",
    "executionId": "bbbbbbEXAMPLE",
    "executionAction": "activating/deactivating {region}",
    "idempotencyKey": "1111111-2222-3333-4444-5555555555", # As there is a possibility
 of dual logging
    "stepDetails" : {
     "stepName": "Routing control step",
     "resource": ["arn:aws:route53-recovery-control::111122223333:controlpanel/
abcdefghiEXAMPLE/routingcontrol/jklmnopqrsEXAMPLE"]
    }
   }
}
```

以下は、リージョン切り替えプラン評価警告の ARC イベントの例です。

リージョン切り替えプランの評価では、警告が返されるとイベントが出力されます。警告がクリアされない場合、警告に対してイベントが出力されるのは 24 時間に 1 回のみです。イベントがクリアされると、その警告に対してそれ以上イベントは出力されません。

```
{
   "version": "0",
   "id": "05d4d2d5-9c76-bfea-72d2-d4614802adb4", # Random uuid
   "detail-type": "ARC Region Switch Plan Execution",
   "source": "aws.arc-region-switch",
   "account": "111122223333",
   "time": "2023-11-16T23:38:14Z",
   "region": "us-east-1",
   "resources": ["arn:aws:arc-region-switch::111122223333:plan/a2b89be4821bfd1d"],
   "detail": {
     "version": "0.0.1",
     "idempotencyKey": "1111111-2222-3333-4444-5555555555",
     "metadata": {
        "evaluationTime" : "timestamp",
        "warning" : "There is a plan evaluation warning for arn:aws:arc-region-
switch::111122223333:plan/a2b89be4821bfd1d. Navigate to the Region switch console to
 resolve."
     }
   }
}
```

ターゲットとして使用する CloudWatch ロググループを指定する

EventBridge ルールを作成するときは、ルールに一致するイベントの送信先のターゲットを指定する必要があります。EventBridge で利用できるターゲットの一覧については、「EventBridge コンソールで使用可能なターゲット」を参照してください。EventBridge ルールに追加できるターゲットの 1 つは、Amazon CloudWatch ロググループです。このセクションでは、CloudWatch ロググループをターゲットとして追加するための要件と、ルール作成時にロググループを追加する手順について説明します。

CloudWatch ロググループをターゲットとして追加するには、次のいずれかを実行します。

- 新しいロググループを作成する
- 既存のロググループを選択する

ルールの作成時にコンソールを使用して新しいロググループを指定する場合は、EventBridge によって自動的にロググループが作成されます。EventBridge ルールのターゲットとして使用するログ

ループが /aws/events で始まることを確認します。既存のロググループを選択する場合は、/aws/events で始まるロググループのみがドロップダウンメニューのオプションとして表示されることに注意してください。詳細については、「Amazon CloudWatch ユーザーガイド」の「新しいロググループを作成する」を参照してください。

コンソール外で CloudWatch オペレーションを使用して CloudWatch ロググループを作成またはターゲットとして使用する場合は、アクセス許可を正しく設定していることを確認してください。コンソールを使用して EventBridge ルールにロググループを追加すると、ロググループのリソースベースのポリシーが自動的に更新されます。ただし、 AWS Command Line Interface または AWS SDK を使用してロググループを指定する場合は、ロググループのリソースベースのポリシーを更新する必要があります。次のポリシー例は、ロググループのリソースベースのポリシーで定義する必要があるアクセス許可を示しています。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:us-east-1:222222222222:log-group:/aws/events/
*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ]
}
```

コンソールを使用してロググループのリソースベースのポリシーを設定することはできません。必要なアクセス許可をリソースベースのポリシーに追加するには、CloudWatch PutResourcePolicy API

オペレーションを使用します。次に、describe-resource-policies CLI コマンドを使用して、ポリシーが正しく適用されたことを確認できます。

リソースイベントのルールを作成して CloudWatch ロググループターゲットを指定するには

1. Amazon EventBridge コンソールの https://console.aws.amazon.com/events/ を開いてください。
2. ルール AWS リージョン を作成する を選択します。
3. [ルールの作成] を選択して、イベントパターンまたはスケジュールの詳細など、そのルールに関する情報を入力します。

   準備状況のための EventBridge ルールの作成の詳細については、「EventBridge で準備状況チェックリソースをモニタリングする」を参照してください。
4. [ターゲットを選択] ページで、ターゲットとして [CloudWatch] を選択します。
5. ドロップダウンメニューから CloudWatch ロググループを選択します。

リージョン切り替えのクォータ

Amazon Application Recovery Controller (ARC) のリージョン切り替えには、次のクォータが適用されます。

エンティティ	クォータ
アカウントあたりのプラン数	10 クォータの引き上げをリクエストできます。
プランあたりの実行ブロック数	100
プランあたりのリージョン切り替えプラン実行ブロック数	25
ステップあたりの並列実行ブロック数	20
トリガー条件あたりの CloudWatch アラーム数	10

AWS SDKsコード例

次のコード例は、 AWS Software Development Kit (SDK) で Application Recovery Controller を使用する方法を示しています。

アクションはより大きなプログラムからのコードの抜粋であり、コンテキスト内で実行する必要があります。アクションは個々のサービス機能を呼び出す方法を示していますが、コンテキスト内のアクションは、関連するシナリオで確認できます。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

コードの例

- AWS SDKs基本的な例
  - AWS SDKsアクション
    - AWS SDK GetRoutingControlStateで を使用する
    - AWS SDK UpdateRoutingControlStateで を使用する

AWS SDKs基本的な例

次のコード例は、 AWS SDK で Amazon Route 53 Application Recovery Controller を使用する基本的な方法を示しています。

例

- AWS SDKsアクション
  - AWS SDK GetRoutingControlStateで を使用する
  - AWS SDK UpdateRoutingControlStateで を使用する

AWS SDKsアクション

次のコード例は、 AWS SDKs を使用して個々の Application Recovery Controller アクションを実行する方法を示しています。それぞれの例には、GitHub へのリンクがあり、そこにはコードの設定と実行に関する説明が記載されています。

以下の例には、最も一般的に使用されるアクションのみ含まれています。完全版は、「Amazon Route 53 Application Recovery Controller API Reference」を参照してください。

例

- AWS SDK GetRoutingControlStateで を使用する
- AWS SDK UpdateRoutingControlStateで を使用する

AWS SDK **GetRoutingControlState**で を使用する

次のサンプルコードは、GetRoutingControlState を使用する方法を説明しています。

Java

SDK for Java 2.x

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
    public static GetRoutingControlStateResponse
 getRoutingControlState(List<ClusterEndpoint> clusterEndpoints,
            String routingControlArn) {
        // As a best practice, we recommend choosing a random cluster endpoint to
 get or
        // set routing control states.
        // For more information, see
        // https://docs.aws.amazon.com/r53recovery/latest/dg/route53-arc-best-
practices.html#route53-arc-best-practices.regional
        Collections.shuffle(clusterEndpoints);
        for (ClusterEndpoint clusterEndpoint : clusterEndpoints) {
            try {
                System.out.println(clusterEndpoint);
                Route53RecoveryClusterClient client =
 Route53RecoveryClusterClient.builder()
                        .endpointOverride(URI.create(clusterEndpoint.endpoint()))
                        .region(Region.of(clusterEndpoint.region())).build();
                return client.getRoutingControlState(
                        GetRoutingControlStateRequest.builder()
```

```
                                .routingControlArn(routingControlArn).build());
            } catch (Exception exception) {
                System.out.println(exception);
            }
        }
        return null;
    }
```

- API の詳細については、「AWS SDK for Java 2.x API Reference」の「GetRoutingControlState」を参照してください。

Python

SDK for Python (Boto3)

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
import boto3


def create_recovery_client(cluster_endpoint):
    """
    Creates a Boto3 Route 53 Application Recovery Controller client for the
 specified
    cluster endpoint URL and AWS Region.

    :param cluster_endpoint: The cluster endpoint URL and Region.
    :return: The Boto3 client.
    """
    return boto3.client(
        "route53-recovery-cluster",
        endpoint_url=cluster_endpoint["Endpoint"],
        region_name=cluster_endpoint["Region"],
    )
```

```
def get_routing_control_state(routing_control_arn, cluster_endpoints):
    """
    Gets the state of a routing control. Cluster endpoints are tried in
    sequence until the first successful response is received.

    :param routing_control_arn: The ARN of the routing control to look up.
    :param cluster_endpoints: The list of cluster endpoints to query.
    :return: The routing control state response.
    """

    # As a best practice, we recommend choosing a random cluster endpoint to get
 or set routing control states.
    # For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    random.shuffle(cluster_endpoints)
    for cluster_endpoint in cluster_endpoints:
        try:
            recovery_client = create_recovery_client(cluster_endpoint)
            response = recovery_client.get_routing_control_state(
                RoutingControlArn=routing_control_arn
            )
            return response
        except Exception as error:
            print(error)
            raise error
```

- API の詳細については、「AWS SDK for Python (Boto3) API Reference」の「GetRoutingControlState」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA lo_exception TYPE REF TO /aws1/cx_rt_generic.
    DATA lo_session TYPE REF TO /aws1/cl_rt_session_base.
    DATA lo_client TYPE REF TO /aws1/if_r5v.
    DATA lt_endpoints TYPE TABLE OF string.
    DATA lv_endpoint TYPE string.
    DATA lv_region TYPE /aws1/rt_region_id.

    " Parse the comma-separated cluster endpoints
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

    " As a best practice, shuffle cluster endpoints to distribute load
    " For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    " For simplicity, we'll try them in order (shuffling can be added if needed)

    " Try each endpoint in order
    LOOP AT lt_endpoints INTO lv_endpoint.
      TRY.
          " Parse endpoint and region from the format "url|region"
          DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
          IF lv_pos > 0.
            DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
            lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
          ELSE.
            " If no region specified, use default
            lv_url = lv_endpoint.
            lv_region = 'us-east-1'.
          ENDIF.

          " Create session for this region
          lo_session = /aws1/cl_rt_session_aws=>create( cv_pfl ).

          " Create client with the specific endpoint
          lo_client = create_recovery_client(
            iv_endpoint = lv_url
            iv_region   = lv_region
            io_session  = lo_session ).

          " Try to get the routing control state
          oo_result = lo_client->getroutingcontrolstate(
```

```
            iv_routingcontrolarn = iv_routing_control_arn ).

          " If successful, return the result
          RETURN.

        CATCH /aws1/cx_r5vendpttmpyunavailex INTO DATA(lo_endpoint_ex).
          " This endpoint is temporarily unavailable, try the next one
          lo_exception = lo_endpoint_ex.
          CONTINUE.

        CATCH /aws1/cx_r5vaccessdeniedex
              /aws1/cx_r5vinternalserverex
              /aws1/cx_r5vresourcenotfoundex
              /aws1/cx_r5vthrottlingex
              /aws1/cx_r5vvalidationex
              /aws1/cx_rt_generic INTO lo_exception.
          " For other errors, re-raise immediately
          RAISE EXCEPTION lo_exception.
      ENDTRY.
    ENDLOOP.

    " If we get here, all endpoints failed - re-raise the last exception
    IF lo_exception IS BOUND.
      RAISE EXCEPTION lo_exception.
    ENDIF.
```

- API の詳細については、 AWS SDK for SAP ABAP API リファレンスのGetRoutingControlState」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

AWS SDK **UpdateRoutingControlState**で を使用する

次のサンプルコードは、UpdateRoutingControlState を使用する方法を説明しています。

Java

SDK for Java 2.x

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
    public static UpdateRoutingControlStateResponse
 updateRoutingControlState(List<ClusterEndpoint> clusterEndpoints,
            String routingControlArn,
            String routingControlState) {
        // As a best practice, we recommend choosing a random cluster endpoint to
 get or
        // set routing control states.
        // For more information, see
        // https://docs.aws.amazon.com/r53recovery/latest/dg/route53-arc-best-
practices.html#route53-arc-best-practices.regional
        Collections.shuffle(clusterEndpoints);
        for (ClusterEndpoint clusterEndpoint : clusterEndpoints) {
            try {
                System.out.println(clusterEndpoint);
                Route53RecoveryClusterClient client =
 Route53RecoveryClusterClient.builder()
                        .endpointOverride(URI.create(clusterEndpoint.endpoint()))
                        .region(Region.of(clusterEndpoint.region()))
                        .build();
                return client.updateRoutingControlState(
                        UpdateRoutingControlStateRequest.builder()

 .routingControlArn(routingControlArn).routingControlState(routingControlState).build());
            } catch (Exception exception) {
                System.out.println(exception);
            }
        }
        return null;
    }
```

- API の詳細については、「AWS SDK for Java 2.x API Reference」の「UpdateRoutingControlState」を参照してください。

Python

SDK for Python (Boto3)

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
import boto3


def create_recovery_client(cluster_endpoint):
    """
    Creates a Boto3 Route 53 Application Recovery Controller client for the
 specified
    cluster endpoint URL and AWS Region.

    :param cluster_endpoint: The cluster endpoint URL and Region.
    :return: The Boto3 client.
    """
    return boto3.client(
        "route53-recovery-cluster",
        endpoint_url=cluster_endpoint["Endpoint"],
        region_name=cluster_endpoint["Region"],
    )



def update_routing_control_state(
    routing_control_arn, cluster_endpoints, routing_control_state
):
    """
    Updates the state of a routing control. Cluster endpoints are tried in
    sequence until the first successful response is received.
```

```
    :param routing_control_arn: The ARN of the routing control to update the
 state for.
    :param cluster_endpoints: The list of cluster endpoints to try.
    :param routing_control_state: The new routing control state.
    :return: The routing control update response.
    """

    # As a best practice, we recommend choosing a random cluster endpoint to get
 or set routing control states.
    # For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    random.shuffle(cluster_endpoints)
    for cluster_endpoint in cluster_endpoints:
        try:
            recovery_client = create_recovery_client(cluster_endpoint)
            response = recovery_client.update_routing_control_state(
                RoutingControlArn=routing_control_arn,
                RoutingControlState=routing_control_state,
            )
            return response
        except Exception as error:
            print(error)
```

- API の詳細については、「AWS SDK for Python (Boto3) API Reference」の「UpdateRoutingControlState」を参照してください。

SAP ABAP

SDK for SAP ABAP

> **Note**
>
> GitHub には、その他のリソースもあります。用例一覧を検索し、AWS コード例リポジトリでの設定と実行の方法を確認してください。

```
    CONSTANTS cv_pfl TYPE /aws1/rt_profile_id VALUE 'ZCODE_DEMO'.
    DATA lo_exception TYPE REF TO /aws1/cx_rt_generic.
```

```
    DATA lo_session TYPE REF TO /aws1/cl_rt_session_base.
    DATA lo_client TYPE REF TO /aws1/if_r5v.
    DATA lt_endpoints TYPE TABLE OF string.
    DATA lv_endpoint TYPE string.
    DATA lv_region TYPE /aws1/rt_region_id.

    " Parse the comma-separated cluster endpoints
    " Expected format: "https://endpoint1.com|us-west-2,https://endpoint2.com|us-
east-1"
    SPLIT iv_cluster_endpoints AT ',' INTO TABLE lt_endpoints.

    " As a best practice, shuffle cluster endpoints to distribute load
    " For more information, see https://docs.aws.amazon.com/r53recovery/latest/
dg/route53-arc-best-practices.html#route53-arc-best-practices.regional
    " For simplicity, we'll try them in order (shuffling can be added if needed)

    " Try each endpoint in order
    LOOP AT lt_endpoints INTO lv_endpoint.
      TRY.
          " Parse endpoint and region from the format "url|region"
          DATA(lv_pos) = find( val = lv_endpoint sub = '|' ).
          IF lv_pos > 0.
            DATA(lv_url) = substring( val = lv_endpoint len = lv_pos ).
            lv_region = substring( val = lv_endpoint off = lv_pos + 1 ).
          ELSE.
            " If no region specified, use default
            lv_url = lv_endpoint.
            lv_region = 'us-east-1'.
          ENDIF.

          " Create session for this region
          lo_session = /aws1/cl_rt_session_aws=>create( cv_pfl ).

          " Create client with the specific endpoint
          lo_client = create_recovery_client(
            iv_endpoint = lv_url
            iv_region   = lv_region
            io_session  = lo_session ).

          " Try to update the routing control state
          oo_result = lo_client->updateroutingcontrolstate(
            iv_routingcontrolarn     = iv_routing_control_arn
            iv_routingcontrolstate   = iv_routing_control_state
            it_safetyrulestooverride = it_safety_rules_override ).
```

```
        " If successful, return the result
        RETURN.

      CATCH /aws1/cx_r5vendpttmpyunavailex INTO DATA(lo_endpoint_ex).
        " This endpoint is temporarily unavailable, try the next one
        lo_exception = lo_endpoint_ex.
        CONTINUE.

      CATCH /aws1/cx_r5vaccessdeniedex
            /aws1/cx_r5vconflictexception
            /aws1/cx_r5vinternalserverex
            /aws1/cx_r5vresourcenotfoundex
            /aws1/cx_r5vthrottlingex
            /aws1/cx_r5vvalidationex
            /aws1/cx_rt_generic INTO lo_exception.
        " For other errors, re-raise immediately
        RAISE EXCEPTION lo_exception.
    ENDTRY.
  ENDLOOP.

  " If we get here, all endpoints failed - re-raise the last exception
  IF lo_exception IS BOUND.
    RAISE EXCEPTION lo_exception.
  ENDIF.
```

- API の詳細については、 AWS SDK for SAP ABAP API リファレンスのUpdateRoutingControlState」を参照してください。

AWS SDK 開発者ガイドとコード例の完全なリストについては、「」を参照してくださいAWS SDK でのこのサービスの使用。このトピックには、使用開始方法に関する情報と、以前の SDK バージョンの詳細も含まれています。

Amazon Application Recovery Controller のセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

セキュリティは、 AWS お客様とお客様の間の責任共有です。責任共有モデルではこれをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任があります AWS クラウド。 AWS また、 では、安全に使用できるサービスも提供しています。サードパーティーの監査者は、AWS コンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。Amazon Application Recovery Controller に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウド内のセキュリティ – お客様の責任は、使用する AWS サービスによって決まります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは、ARC を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために ARC を設定する方法を示します。また、ARC リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

トピック

- Amazon Application Recovery Controller でのデータ保護
- Amazon Application Recovery Controller (ARC) の Identity and Access Management
- ARC でのログ記録とモニタリング
- Amazon Application Recovery Controller のコンプライアンスの検証
- Amazon Application Recovery Controller のレジリエンス
- Amazon Application Recovery Controller のインフラストラクチャセキュリティ

Amazon Application Recovery Controller でのデータ保護

責任 AWS 共有モデル、Amazon Application Recovery Controller でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。また、使用する「 AWS のサービス 」のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、データプライバシーに関するよくある質問を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

データ保護の目的で、認証情報を保護し AWS アカウント 、 AWS IAM アイデンティティセンターまたは AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 は必須ですが、TLS 1.3 を推奨します。
- で API とユーザーアクティビティのログ記録を設定します AWS CloudTrail。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「 AWS CloudTrail ユーザーガイド」のCloudTrail 証跡の使用」を参照してください。
- AWS 暗号化ソリューションと、 内のすべてのデフォルトのセキュリティコントロールを使用します AWS のサービス。
- Amazon Macie などの高度な管理されたセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介して にアクセスするときに FIPS 140-3 検証済み暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、タグ、または [名前] フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これは、コンソール、API、または SDK を使用して ARC AWS CLIまたは他の AWS のサービス を使用する場合も同様です。 AWS SDKs タグ、または名前に使用される自由記述のテキストフィールドに入力したデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、認証情報を URL に含めないことを強くお勧めします。

保管中の暗号化

Amazon Application Recovery Controller に保存されている顧客の設定情報は、保管時に暗号化されます。

転送中の暗号化

Amazon Application Recovery Controller に対する顧客のリクエストと応答は、TLS を使用してサービス全体の転送中に暗号化されます。

Amazon Application Recovery Controller (ARC) の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、誰が認証 (サインイン) でき、誰が ARC リソースの使用を許可される (権限を持つ) かを制御します。IAM は、追加料金なしで使用できる AWS のサービス です。

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、ロールによって異なります。

- サービスユーザー - 機能にアクセスできない場合は、管理者にアクセス許可をリクエストします (「Amazon Application Recovery Controller (ARC) のアイデンティティとアクセスのトラブルシューティング」を参照)。
- サービス管理者 - ユーザーアクセスを決定し、アクセス許可リクエストを送信します (「Amazon Application Recovery Controller (ARC) 機能と IAM の連携方法」を参照)
- IAM 管理者 - アクセスを管理するためのポリシーを作成します (「Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例」を参照)

アイデンティティを使用した認証

認証は、ID 認証情報 AWS を使用して にサインインする方法です。、IAM ユーザー AWS アカウントのルートユーザー、または IAM ロールを引き受けることで認証される必要があります。

AWS IAM アイデンティティセンター (IAM Identity Center)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーティッド ID としてサイ

ンインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「AWS アカウントにサインインする方法」を参照してください。

プログラムによるアクセスの場合、 は SDK と CLI AWS を提供してリクエストを暗号化して署名します。詳細については、「IAM ユーザーガイド」の「API リクエストに対するAWS 署名バージョン 4」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、まず、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーテッドアイデンティティ

ベストプラクティスとして、人間のユーザーが一時的な認証情報 AWS のサービス を使用して にアクセスするには、ID プロバイダーとのフェデレーションを使用する必要があります。

フェデレーティッド ID は、エンタープライズディレクトリ、ウェブ ID プロバイダー、または ID ソースからの認証情報 AWS のサービス を使用して Directory Service にアクセスするユーザーです。フェデレーテッドアイデンティティは、一時的な認証情報を提供するロールを引き受けます。

アクセスを一元管理する場合は、 AWS IAM アイデンティティセンターをお勧めします。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「IAM アイデンティティセンターとは」を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧めします。詳細については、IAM ユーザーガイドの「ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可を持つアイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替えるか、 または API オペレーションを呼び出すことで、ロールを引き受けることができます。 AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイド の IAM でのクロスアカウントリソースアクセス を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられたときにアクセス許可を定義します。は、プリンシパルがリクエストを行うときにこれらのポリシー AWS を評価します。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「管理ポリシーとインラインポリシーのいずれかを選択する」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の上限を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可境界」を参照してください。
- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「ポリシー評価ロジック」を参照してください。

Amazon Application Recovery Controller (ARC) 機能と IAM の連携方法

各 Amazon Application Recovery Controller (ARC) 機能が IAM とどのように連携するかについては、以下のトピックを参照してください。

- ゾーンシフトの IAM
- ゾーンオートシフトの IAM
- ルーティングコントロールの IAM
- 準備状況チェックの IAM
- リージョン切り替えの IAM

Amazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例

Amazon Application Recovery Controller (ARC) の各機能のアイデンティティベースのポリシーの例を確認するには、各機能の AWS Identity and Access Management 章の以下のトピックを参照してください。

- ARC でのゾーンオートシフトのアイデンティティベースのポリシー例
- ARC でのゾーンシフトのアイデンティティベースのポリシーの例
- ARC のルーティングコントロールに関するアイデンティティベースのポリシー例
- ARC における準備状況チェックのアイデンティティベースのポリシーの例

AWS Amazon Application Recovery Controller (ARC) の マネージドポリシー

サービスにリンクされたロールの AWS 管理ポリシーなど、 管理ポリシーを持つ ARC 機能の 管理ポリシーの詳細については、以下のトピックを参照してください。

- ゾーンオートシフトのマネージドポリシー
- ルーティングコントロールのマネージドポリシー
- 準備状況チェックのためのマネージドポリシー

Amazon Application Recovery Controller (ARC) の AWS マネージドポリシーの更新

このサービスがこれらの変更の追跡を開始してからの ARC の機能の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動通知については、ARC の「ドキュメント履歴ページ」で RSS フィードをサブスクライブしてください。

変更	説明	日付
AmazonApplicationRecoveryControllerRegionSwitchPlanExecutionPolicy – 新しいポリシー	Amazon Application Recovery Controller (ARC) は、リージョン切り替えプランの実行と評価のためのアクセス許可を付与する新しいマネージドポリシーをリリースしました。 このポリシーは、リージョン切り替えプラン情報、実行ステータス、Amazon CloudWatch モニタリングデータへの読み取り専用アクセスを提供します。また、プラン評価のために IAM プリンシパルポリシーをシミュレートするアクセス許可も含まれています。	2025 年 11 月 3 日
AWSZonalAutoshiftPracticeRunSLRPolicy マネージドポリシー – 更新されたポリシー	ポリシーステートメント `AutoshiftPracticeCheckPermissions` を追加し、`autoscaling:DescribeAutoScalingGroups` 、`ec2:DescribeInstances` 、`elasticloadbalancing:DescribeTargetHealth` 、および `elasticloadbalancing:DescribeTargetH`	2025 年 6 月 30 日

変更	説明	日付
	ealth の権限を付与し、バランスの取れた容量チェックをサポートします。 詳細については[ゾーンオートシフトと練習実行の仕組み]を参照してください。	
[AWSServiceRoleForPercPracticePolicy] - 新しいポリシー	ARC に、オートシフトと練習実行用の新しいサービスリンクロールが追加されました。 ARC は、サービスにリンクされたロールによって有効化されたアクセス許可を使用して、お客様が用意した Amazon CloudWatch アラームとお客様 Health Dashboard イベントをモニタリングし、練習実行を開始します。 新しいサービスリンクロールの詳細については、「[AWSServiceRoleForZonalAutoshiftPracticeRun のサービスリンクロールアクセス許可]」を参照してください。	2023 年 11 月 30 日
[AmazonRoute53RecoveryControlConfigReadOnlyAccess] — ポリシーの更新	共有リソースの AWS Resource Access Manager リソースポリシーの詳細を返すことをサポートするためにGetResourcePolicy 、のアクセス許可を追加します。	2023 年 10 月 18 日

変更	説明	日付

Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC に、Amazon EC2 インスタンスに関する情報をクエリするための新しいアクセス許可が追加されました。
	
	ARC は、以下のアクセス許可を使用して Amazon EC2 インスタンスのポーリングをサポートし、準備状況チェックを実行して、インスタンスの準備状況のステータスを判定します。
	
	`ec2:DescribeVpnGateways`
	
	`ec2:DescribeCustomerGateways`
		2023 年 2 月 17 日

Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC に、Lambda 関数の情報をクエリするための新しいアクセス許可が追加されました。 ARC は、以下のアクセス許可を使用して Lambda 関数に関する情報をクエリし、準備状況チェックを実行して、関数の準備状況のステータスを判定します。 `lambda:ListProvisionedConcurrencyConfigs`	2022 年 8 月 31 日

変更	説明	日付
AmazonRoute53RecoveryControlConfigFullAccess – ポリシーの更新	ポリシーから Amazon Route 53 のアクセス許可が削除され、オプションのアクセス許可を記した注記が、新たに追加されました。	2022 年 5 月 26 日
AmazonRoute53RecoveryControlConfigFullAccess – ポリシーの更新	不足していた、ポリシーへの Amazon Route 53 のアクセス許可が追加されました。	2022 年 4 月 15 日
AmazonRoute53RecoveryClusterReadOnlyAccess – ポリシーの更新	ARC に新しいアクセス許可 `route53-recovery-cluster:ListRoutingControls` が追加され、可用性の高いルーティングコントロール ARN をリスト化できるようになりました。	2022 年 3 月 15 日
AmazonRoute53RecoveryControlConfigReadOnlyAccess — ポリシーの更新	ARC に新しいアクセス許可 `route53-recovery-control-config:ListTagsForResources` が追加され、リソースのタグをリスト化できるようになりました。	2021 年 12 月 20 日

変更	説明	日付
Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC に、Amazon API Gateway に関する情報をクエリするための新たなアクセス許可が追加されました。 ARC は、アクセス許可 `apigateway:GET` を使用して API ゲートウェイに関する情報をクエリし、準備状況チェックを実行して、準備状況のステータスを判定します。	2021 年 10 月 28 日
AmazonRoute53RecoveryReadinessReadOnlyAccess - 新しいアクセス許可を追加	ARC に、AmazonRoute53RecoveryReadinessReadOnlyAccess への 2 つのアクセス許可が新たに追加されました。 ARC は、`route53-recovery-readiness:GetArchitectureRecommendations` と `route53-recovery-readiness:GetCellReadinessSummary` を使って、リカバリの準備状況を操作するこれらのアクションへの、読み取り専用アクセスを許可します。	2021 年 10 月 15 日

変更	説明	日付

Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC に、Lambda 関数の情報をクエリするための新しいアクセス許可が追加されました。
	
	ARC は、以下のアクセス許可を使用して Lambda 関数に関する情報をクエリし、準備状況チェックを実行して、関数の準備状況のステータスを判定します。
	
	`lambda:GetFunctionConcurrency`
	
	`lambda:GetFunctionConfiguration`
	
	`lambda:GetProvisionedConcurrencyConfig`
	
	`lambda:ListAliases`
	
	`lambda:ListVersionsByFunction`
	
	`lambda:ListEventSourceMappings`
	
	`lambda:ListFunctions`
		2021 年 10 月 8 日

変更	説明	日付
Route53RecoveryReadinessServiceRolePolicy — 新しいマネージドポリシーを追加	ARC に以下の新しいマネージドポリシーが追加されました。 AmazonRoute53RecoveryReadinessFullAccess AmazonRoute53RecoveryReadinessReadOnlyAccess AmazonRoute53RecoveryClusterFullAccess AmazonRoute53RecoveryClusterReadOnlyAccess AmazonRoute53RecoveryControlConfigFullAccess AmazonRoute53RecoveryControlConfigReadOnlyAccess	2021 年 8 月 18 日
ARC は変更の追跡を開始しました	ARC は、 AWS 管理ポリシーの変更の追跡を開始しました。	2021 年 7 月 27 日

Amazon Application Recovery Controller (ARC) のアイデンティティとアクセスのトラブルシューティング

次の情報は、Amazon Application Recovery Controller (ARC) および IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

トピック

- ARC でアクションを実行する権限がない

- iam:PassRole を実行する権限がない
- 自分の 以外のユーザーに ARC リソース AWS アカウント へのアクセスを許可したい

ARC でアクションを実行する権限がない

にアクションを実行する権限がないと AWS マネジメントコンソール 通知された場合は、管理者に連絡してサポートを依頼する必要があります。管理者は、認証情報を自分に提供した人物です。

以下のエラー例は、mateojackson IAM ユーザーがコンソールを使用して架空の *my-example-widget* リソースに関する詳細情報を表示しようとしているが、架空の route53-recovery-readiness:*GetWidget* 権限がないという場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 route53-recovery-readiness:GetWidget on resource: my-example-widget
```

この場合、Mateo は、route53-recovery-readiness:*GetWidget* アクションを使用して *my-example-widget* リソースにアクセスできるように、ポリシーの更新を管理者に依頼します。

iam:PassRole を実行する権限がない

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して ARC にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、既存のロールをそのサービスに渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して ARC でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与されたアクセス許可が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

自分の 以外のユーザーに ARC リソース AWS アカウント へのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください:

- ARC がこれらの機能をサポートしているかどうかを確認するには、「Amazon Application Recovery Controller (ARC) 機能と IAM の連携方法」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「IAM ユーザーガイド」の「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイド の 外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可 を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、IAM ユーザーガイド の IAM でのクロスアカウントのリソースへのアクセス を参照してください。

インターフェイスエンドポイント (AWS PrivateLink) を使用して Amazon Application Recovery Controller (ARC) ゾーンシフトにアクセスする

を使用して AWS PrivateLink 、VPC と Amazon Application Recovery Controller (ARC) ゾーンシフトの間にプライベート接続を作成できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、または Direct Connect 接続を使用せずに、VPC 内にあるかのように ARC ゾーンシフトにアクセスできます。VPC のインスタンスは、パブリック IP アドレスがなくても ARC ゾーンシフトにアクセスできます。

このプライベート接続を確立するには、 AWS PrivateLinkを利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、ARC ゾーンシフト宛てのトラフィックのエントリポイントとして機能するリクエスタ管理型ネットワークインターフェイスです。

詳細については、「 AWS PrivateLink ガイド」の「Access AWS のサービス through AWS PrivateLink」を参照してください。

ARC ゾーンシフトに関する考慮事項

ARC ゾーンシフトのインターフェイスエンドポイントを設定する前に、「AWS PrivateLink ガイド」の「考慮事項」を確認してください。

ARC ゾーンシフトは、インターフェイスエンドポイントを介してすべての API アクションの呼び出しをサポートしています。

ARC ゾーンシフトのインターフェイスエンドポイントを作成する

Amazon VPC コンソールまたは AWS Command Line Interface () を使用して、ARC ゾーンシフトのインターフェイスエンドポイントを作成できますAWS CLI。詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントを作成」を参照してください。

以下のサービス名を使用して、ARC ゾーンシフト用のインターフェイスエンドポイントを作成します。

```
com.amazonaws.region.arc-zonal-shift
```

インターフェイスエンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名を使用して、ARC ゾーンシフトへの API リクエストを実行できます。例えば、`arc-zonal-shift.us-east-1.amazonaws.com`。

インターフェイスエンドポイントのエンドポイントポリシーを作成する

エンドポイントポリシーは、インターフェイスエンドポイントにアタッチできる IAM リソースです。デフォルトのエンドポイントポリシーでは、インターフェイスエンドポイントを介した ARC ゾーンシフトへのフルアクセスが許可されています。VPC から ARC ゾーンシフトに許可されるアクセスを制御するには、カスタムエンドポイントポリシーをインターフェイスのエンドポイントにアタッチします。

エンドポイントポリシーは以下の情報を指定します。

- アクションを実行できるプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール)。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、AWS PrivateLink ガイドのControl access to services using endpoint policies (エンドポイントポリシーを使用してサービスへのアクセスをコントロールする)を参照してください。

例: ARC ゾーンシフトアクション用の VPC エンドポイントポリシー

以下は、カスタムエンドポイントポリシーの例です。インターフェイスエンドポイントにアタッチされると、このポリシーは、すべてのリソースですべてのプリンシパルに、リストされている ARC ゾーンシフトアクションへのアクセス権を付与します。

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "arc-zonal-shift:ListManagedResources",
            "arc-zonal-shift:StartZonalShift",
            "arc-zonal-shift:CancelZonalShift"
         ],
         "Resource":"*"
      }
   ]
}
```

Resource は arn:aws:elasticloadbalancing:us-east-*1:111122223333*:loadbalancer/app/Testing/*1111111ecd42dc05* としてリストすることもできます。

ARC でのログ記録とモニタリング

モニタリングは、ARC と AWS ソリューションの可用性とパフォーマンスを維持する上で重要な部分です。マルチポイント障害が発生した場合は、その障害をより簡単にデバッグできるように、 AWS ソリューションのすべての部分からモニタリングデータを収集する必要があります。 には、ARC リソースとアクティビティをモニタリングし、潜在的なインシデントに対応するための複数のツール AWS が用意されています。たとえば、 AWS CloudTrail や Amazon CloudWatch などです。

ARC の各機能のモニタリングについては、以下のトピックを参照してください。

- ゾーンシフトのログ記録とモニタリング

- ゾーンオートシフトのログ記録とモニタリング
- ルーティングコントロールのログ記録とモニタリング
- リージョン切り替えのログ記録とモニタリング
- 準備状況チェックのログ記録とモニタリング

Amazon Application Recovery Controller のコンプライアンスの検証

サードパーティーの監査者は、複数のコンプライアンスプログラムの一環として Amazon Application Recovery Controller のセキュリティと AWS コンプライアンスを評価します。このプログラムには、SOC、PCI、HIPAA などを含みます。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、「コンプライアンスAWS のサービス プログラムによるスコープ」の「コンプライアンス」を参照して、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。を使用する際のコンプライアンス責任の詳細については AWS のサービス、AWS 「 セキュリティドキュメント」を参照してください。

Amazon Application Recovery Controller のレジリエンス

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。 AWS リージョン は、低レイテンシー、高スループット、高度に冗長なネットワークで接続された、物理的に分離および分離された複数のアベイラビリティーゾーンを提供します。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェールオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、フォールトトレランス、および拡張性が優れています。

AWS リージョン およびアベイラビリティーゾーンの詳細については、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。

グローバル AWS インフラストラクチャに加えて、ARC には、データの耐障害性とバックアップのニーズをサポートするのに役立つ機能がいくつか用意されています。

Amazon Application Recovery Controller のインフラストラクチャセキュリティ

マネージドサービスとして、 は AWS グローバルネットワークセキュリティで保護されています。AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して環境を AWS 設計するには、「Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

AWS 公開された API コールを使用して、ネットワーク経由で ARC にアクセスします。クライアントは次をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

Amazon Application Recovery Controller (ARC) デベロッパーガイド、ドキュメント履歴

以下に、Amazon Application Recovery Controller (ARC) ドキュメントの、重要な変更事項を記載します。

- バージョン: 最新
- ドキュメントの最終更新日: 2026 年 3 月 31 日

変更	説明	日付
準備状況チェックの可用性の変更	Amazon Application Recovery Controller (ARC) の準備状況チェック機能は、2026 年 4 月 30 日以降、新規のお客様に公開されなくなります。既存のお客様は、通常どおりサービスを引き続き使用できます。 詳細については、「Amazon Application Recovery Controller (ARC) の準備状況チェックの可用性の変更」を参照してください。	2026 年 3 月 31 日
リージョン切り替えプラン実行の新しいマネージドポリシー	Amazon Application Recovery Controller (ARC) は、リージョン切り替えプランの実行と評価のためのアクセス許可を付与する新しい管理ポリシー、`AmazonApplicationRecoveryControllerRegionSwitchPlanExecutionPolicy` をリリースしました。	2025 年 11 月 3 日

変更	説明	日付
	詳細については、「Amazon Application Recovery Controller (ARC) AWS 管理ポリシーの更新」を参照してください。	
VPC と Amazon Application Recovery Controller (ARC) ゾーンシフト AWS PrivateLink の間で を使用できるようになりました。	を使用して AWS PrivateLink 、VPC と Amazon Application Recovery Controller (ARC) ゾーンシフトの間にプライベート接続を作成できます。 詳細については、「インターフェイスエンドポイント (AWS PrivateLink) を使用して Amazon Application Recovery Controller (ARC) ゾーンシフトにアクセスする」を参照してください。	2025 年 8 月 11 日
新しいリージョン切り替えサービス	リージョン切り替えを使用すると、マルチリージョンアプリケーションを別の AWS リージョンから運用するために必要な、クロスアカウントをサポートする特定のステップをオーケストレーションできます。 詳細については、「ARC でのリージョン切り替え」を参照してください。	2025 年 8 月 1 日

変更	説明	日付
練習実行の機能強化	ARC でオンデマンド練習実行を開始できるようになりました。さらに、練習実行には、リージョン内の他の AZ で十分な容量のチェックが含まれるようになりました。 詳細については、「仕組み」を参照してください。	2025 年 6 月 30 日
マネージドポリシーを更新	`autoscaling:DescribeAutoScalingGroups` 、`ec2:DescribeInstances` 、`elasticloadbalancing:DescribeTargetHealth` 、`elasticloadbalancing:DescribeTargetHealth` のアクセス許可を持つポリシーステートメント `AutoshiftPracticeCheckPermissions` を追加して `AWSZonalAutoshiftPracticeRunSLRPolicy` マネージドポリシーを更新し、バランスの取れたキャパシティチェックをサポートします。 詳細については、「AWSZonalAutoshiftPracticeRunSLRPolicy マネージドポリシー」を参照してください。	2025 年 6 月 30 日

変更	説明	日付
ゾーンオートシフトの例外タイプへの更新	ゾーンオートシフトをリソースごとに操作できるようになりました。 詳細については、「仕組み」を参照してください。	2025 年 4 月 21 日
で ARC ゾーンオートシフトをテストする AWS FIS	を使用して AWS FIS 、AZ 電源の中断中に ARC ゾーンオートシフトがアプリケーションを自動的に復旧する方法をテストできます。 詳細については、「 を使用したゾーンオートシフトのテスト AWS FIS」を参照してください。	2025 年 3 月 26 日
ARC は、ルーティングコントロールとゾーンシフトの IPv6 エンドポイントをサポートするようになりました。	ARC は、ルーティングコントロールとゾーンシフトの IPv6 エンドポイントをサポートするようになりました。 詳細については、「ルーティングコントロールコンポーネントの設定」を参照してください。	2024 年 11 月 21 日

変更	説明	日付
Amazon EC2 Auto Scaling グループのゾーンシフト機能	ARC が Amazon EC2 Auto Scaling グループのゾーンシフトをサポートするようになりました。 詳細については、「Amazon EC2 Auto Scaling グループのサポート」を参照してください。	2024 年 11 月 18 日
Amazon EKS のゾーンシフト機能	Amazon EKS クラスターのゾーンシフトを開始することも、ゾーンオートシフトを有効にして AWS に許可することもできます。このシフトは、正常な AZ のワーカーノードで実行されているポッドのネットワークエンドポイントのみを使用するようにクラスター内の東西ネットワークトラフィックのフローを更新します。 詳細については、「Amazon Elastic Kubernetes Service のサポート」を参照してください。	2024 年 10 月 22 日

変更	説明	日付
Network Load Balancer のゾーンシフト機能	ARC は、クロスゾーンが有効な設定またはクロスゾーンが無効な設定の Network Load Balancer のゾーンシフトをサポートするようになりました。 詳細については、「Network Load Balancer のサポート」を参照してください。	2024 年 10 月 11 日
オートシフトオブザーバー通知	オートシフトオブザーバー通知を使用すると、 AWS がオートシフトを開始してトラフィックを障害の可能性があるアベイラビリティーゾーンから遠ざけるたびに、Amazon EventBridge を介して通知するようにゾーンオートシフトを設定できます。これらの個別の通知を有効にするために、ゾーンオートシフトで特定のリソースを設定する必要はありません。 詳細については、「Amazon EventBridge でのゾーンオートシフトの使用」を参照してください。	2024 年 7 月 12 日

変更	説明	日付
各機能によるドキュメントの再編成	デベロッパーガイドのコンテンツがサブデベロッパーガイドにサイロ化するように再編成しました。つまり、マルチ AZ 復旧のためのゾーンシフトとゾーンオートシフト、マルチリージョン復旧のためのルーティング制御と準備状況チェックなど、ARC の各機能に関する包括的な情報を含む個別のセクションが追加されました。 詳細については、「Amazon Application Recovery Controller (ARC) とは」を参照してください。	2024 年 4 月 30 日
ゾーンオートシフト機能を追加	ユーザーに代わってアプリケーションのリソーストラフィックをアベイラビリティーゾーンから遠ざける AWS ことを に許可する ARC の新しい機能を追加し、イベント中の復旧時間を短縮します。 詳細については、「Amazon Application Recovery Controller (ARC) のゾーンオートシフト」を参照してください。	2023 年 11 月 30 日

変更	説明	日付
新しいサービスリンクロールを追加する	ゾーンオートシフトの練習実行のために、新しいサービスリンクロール AWSServiceRoleForZonalAutoshiftPracticeRun を追加します。 詳細については、「AWSServiceRoleForZonalAutoshiftPracticeRun のサービスリンクロールのアクセス許可」を参照してください。	2023 年 11 月 30 日
クラスターのクロスアカウントのサポートを追加	を使用して ARC のクラスターのクロスアカウントサポートを追加し AWS Resource Access Manager、1 つのクラスターを簡単かつ安全に使用して、複数の異なる AWS アカウントが所有するコントロールパネルとルーティングコントロールをホストできるようにします。 詳細については、「ARC でクラスターのクロスアカウントをサポート」を参照してください。	2023 年 10 月 18 日

変更	説明	日付
マネージドポリシーを更新	AmazonRoute53RecoveryControlConfigReadOnly マネージドポリシーを更新して、 のアクセス許可を追加しGetResourcePolicy 、共有 AWS Resource Access Manager リソースのリソースポリシーに関する詳細を返すことをサポートします。 詳細については、「AWS マネージドポリシー」を参照してください。	2023 年 9 月 19 日
サービスにリンクされたロールを更新	Amazon EC2 インスタンスのポーリングをサポートするため、ARC のサービスリンクロールに新しいアクセス許可 ec2:DescribeVpnGateways と ec2:DescribeCustomerGateways を追加しました。 詳細については、「ARC のサービスにリンクされたロールの使用」を参照してください。	2023 年 2 月 17 日

変更	説明	日付
ゾーンシフトの一般提供版	ARC のゾーンシフトの一般提供版がサポートされました。これには、ゾーンシフト用に ARC に登録されているマネージドリソースの属性ベースのアクセス制御 (ABAC) が含まれています。 詳細については、「ARC による属性ベースのアクセス制御 (ABAC)」を参照してください。	2023 年 1 月 10 日
新しいマルチ AZ ゾーンシフトを追加	マルチ AZ アプリケーション用の ARC の新サービスであるゾーンシフトについて説明するコンテンツが追加されました。ゾーンシフトを開始すると、ロードバランサーのリソースのトラフィックを、アベイラビリティーゾーンから切り離せます。 詳細については、「ARC のゾーンシフト」を参照してください。	2022 年 11 月 28 日

変更	説明	日付
サービスにリンクされたロールを更新	ARC のサービスリンクロールに、Lambda 関数に関する情報をクエリするための新しいアクセス許可 `lambda:ListProvisionedConcurrencyConfigs` を追加しました。 詳細については、「ARC のサービスにリンクされたロールの使用」を参照してください。	2022 年 8 月 31 日
マネージドポリシーの更新	Amazon Route 53 のアクセス許可を削除してそれらをオプションとしてリスト化するように、`AmazonRoute53RecoveryControlConfigFullAccess` マネージドポリシーが更新されました。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2022 年 5 月 26 日

変更	説明	日付
マネージドポリシーの更新	必要な Amazon Route 53 のアクセス許可を追加するように `AmazonRoute53RecoveryControlConfigFullAccess` マネージドポリシーが更新されました。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2022 年 4 月 15 日
新しいルーティングコントロールリスト API の CLI 例を追加	非常に信頼性の高い ARC データプレーン API に含まれる、新しいルーティングコントロールリスト API オペレーションの、CLI コマンド例とベストプラクティスの推奨事項が追加されました。 詳細については、「List and update routing controls and states」を参照してください。	2022 年 3 月 31 日

変更	説明	日付
安全ルールのオーバーライドを新たにサポート	安全ルールのオーバーライドが新たにサポートされました。これにより、設定済みの安全ルールにより適用される、ルーティングコントロールのセーフガードを安全に迂回できるようになります。安全ルールのオーバーライドが必要になるのは、例えば、ディザスタリカバリのフェイルオーバーにおける「Break Glass」のシナリオにおいてです。 詳細については、「Override safety rules to reroute traffic」を参照してください。	2022 年 3 月 2 日
タグ付けのサポートが新たに追加	クラスター、コントロールパネル、ルーティングコントロール、安全ルールなど、ARC の追加リソースのタグ付けが、新たにサポートされました。 詳細については、「Amazon Application Recovery Controller (ARC) でのタグ付け」を参照してください。	2021 年 12 月 20 日

変更	説明	日付
マネージドポリシーの更新	リソースのタグをリスト化するアクセス許可を追加するように、AmazonRoute53RecoveryControlConfigReadOnly マネージドポリシーが更新されました。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2021 年 12 月 20 日
EventBridge でのリアルタイムアラートが新たにサポート	EventBridge が新たにサポートされました。これにより、例えば READY から NOT READY に変わったときなど、ARC の準備状況チェックのステータスの変更に基づいて、アラートを受信したり何らかの対応をしたりするためのルールを追加できるようになりました。 詳細については、「Amazon EventBridge で ARC を使用する」を参照してください。	2021 年 12 月 20 日

変更	説明	日付
ルーティングコントロールの状態のコードサンプルを追加	API オペレーションを使用してルーティングコントロールの状態を取得または更新するときに、クラスターエンドポイントを順番に試行する例を示す、コードサンプルが追加されました。 詳細については、「Amazon Application Recovery Controller (ARC) の API の例」を参照してください。	2021 年 11 月 16 日
読み取り専用ポリシーの新たなアクセス許可を追加	ポリシー AmazonRoute53RecoveryReadinessReadOnlyAccess に、route53-recovery-readiness:GetArchitectureRecommendations と route53-recovery-readiness:GetCellReadinessSummary の 2 つの新しいアクセス許可が追加されました。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2021 年 11 月 9 日

変更	説明	日付
Amazon API Gateway リソースタイプを新たにサポート	新しいリソースタイプとして Amazon API Gateway が追加され、ARC のサービスにリンクされたロールのアクセス許可が更新されました。今後は、ARC で準備状況チェックを使って API Gateway を監査できます。 詳細については、「準備状況ルールとサポートされているリソースタイプ」および「ARC のサービスにリンクされたロールの使用」を参照してください。	2021 年 10 月 28 日
Lambda 関数のリソースタイプを新たにサポート	新しいリソースタイプとして Lambda 関数が追加され、ARC の、サービスにリンクされたロールのアクセス許可が更新されました。今後は、ARC で準備状況チェックを使って Lambda 関数を監査できます。 詳細については、「準備状況ルールとサポートされているリソースタイプ」および「ARC のサービスにリンクされたロールの使用」を参照してください。	2021 年 10 月 8 日

変更	説明	日付
CloudFormation と Terraform テンプレートへのリンクを追加	ARC の使用をすばやく開始できるように、ダウンロード可能なテンプレート CloudFormation と Hashicorp Terraform テンプレートへのリンクを追加しました。詳細については、「新しいアプリケーションでのリカバリの準備」を参照してください。	2021 年 9 月 13 日
新しいマネージドポリシーを追加	ARC の次の AWS 管理ポリシーを追加しました: `AmazonRoute53RecoveryReadinessFullAccess` 、`AmazonRoute53RecoveryReadinessReadOnlyAccess` 、`AmazonRoute53RecoveryClusterFullAccess` 、`AmazonRoute53RecoveryClusterReadOnlyAccess` 、`AmazonRoute53RecoveryControlConfigFullAccess` 、および `AmazonRoute53RecoveryControlConfigReadOnlyAccess` 。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2021 年 8 月 18 日

変更	説明	日付
Amazon Application Recovery Controller (ARC) の管理 AWS ポリシーの追跡を開始しました	マネージドポリシーの更新は初回のリリース日から追跡されます。 詳細については、「Amazon Application Recovery Controller (ARC) のAWS マネージドポリシー」を参照してください。	2021 年 7 月 27 日
Amazon Application Recovery Controller (ARC) の初回のリリース	ARC は、リージョン内または複数の AWS リージョン間でフェイルオーバーを一元的に調整することで、アプリケーションの可用性を向上させます。ARC に準備状況チェックが追加され、フェイルオーバートラフィックに対処できるようにアプリケーションをスケールし、障害を回避するように設定することが可能になりました。また、信頼性がきわめて高いルーティングコントロールが可能であるため、例えば、アベイラビリティーゾーンやリージョンを横断してトラフィックのルートを変更し、アプリケーションを復旧することが可能です。詳細については、「ARC とは」を参照してください。	2021 年 7 月 27 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。