aws

デベロッパーガイド

Amazon Application Recovery Controller (ARC)

Copyright © 2025 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 © 2025 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リカバリ 1
マルチリージョンリカバリ 2
マルチ AZ とマルチリージョンの機能を比較する 4
マルチ AZ リカバリ 7
ゾーンシフト 7
ゾーンシフトの仕組み 8
AWS リージョン 9
ゾーンシフトのコンポーネント 14
データプレーンとコントロールプレーン 16
料金 17
ベストプラクティス 17
API オペレーション 19
CLI オペレーションの使用例 19
サポート リソース 24
ゾーンシフトの開始、更新、またはキャンセル 36
ログ記録とモニタリング 38
ゾーンシフトの IAM 42
ゾーンオートシフト 53
ゾーンオートシフトの仕組み 55
AWS リージョン 64
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 68
ベストプラクティス 68
API オペレーション 73
CLI オペレーションの使用例 74
ゾーンオートシフトの有効化と操作 80
を使用したゾーンオートシフトのテスト AWS FIS 85
ログ記録とモニタリング 86
Identity and Access Management 97
マルチリージョンリカバリ 113
ルーティングコントロール 113
ルーティングコントロールについて 114

AWS リージョン 117
コンポーネント 118
データプレーンとコントロールプレーン 121
Tagging 122
料金 123
マルチリージョンリカバリの開始方法 123
ベストプラクティス 125
API オペレーション 128
CLI オペレーションの使用例 132
ルーティングコントロールコンポーネントの使用 149
ログ記録とモニタリング 169
Identity and Access Management 173
クォータ 188
準備状況チェック 188
準備状況チェックとは 189
AWS リージョン 197
コンポーネント 197
データプレーンとコントロールプレーン 200
Tagging 201
料金 201
回復力のあるアプリケーションを設定する 202
ベストプラクティス 202
API オペレーション 203
CLI オペレーションの使用例 205
リカバリグループと準備状況チェックの使用 215
準備状況ステータスをモニタリングする 220
アーキテクチャの推奨事項を取得する 222
クロスアカウント認可の作成 224
準備状況ルール、リソースタイプ、ARNS 226
ログ記録とモニタリング 246
Identity and Access Management 261
クォータ 276
リージョンの切り替え 276
リージョンスイッチについて 277
ベストプラクティス 284
チュートリアル: アクティブ/パッシブプラン 286

API オペレーション ... 293
リージョンスイッチの使用 ... 296
ダッシュボード ... 321
クロスアカウントのサポート ... 321
Identity and Access Management ... 327
ログ記録とモニタリング ... 346
クォータ ... 355
コードの例 ... 356
基本 ... 356
アクション ... 356
セキュリティ ... 363
データ保護 ... 364
保管中の暗号化 ... 365
転送中の暗号化 ... 365
Identity and Access Management ... 365
対象者 ... 365
アイデンティティを使用した認証 ... 366
ポリシーを使用したアクセスの管理 ... 369
Amazon Application Recovery Controller (ARC) 機能が IAM と連携する方法 ... 372
アイデンティティベースのポリシーの例 ... 372
AWS マネージドポリシー ... 373
トラブルシューティング ... 380
ログ記録とモニタリング ... 382
コンプライアンス検証 ... 383
耐障害性 ... 384
インフラストラクチャセキュリティ ... 384
ドキュメント履歴 ... 386
... cdiii

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 には以下の機能があります。

- 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。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することで、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復旧のためのルーティングコントロールとリージョンスイッチ、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リカバリ

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と一時的なものです。ゾーンシフトを開始するときは、最大 3 日間の (拡張可能な) 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、「」を参照してくださいサポート リソース。

ゾーンオートシフト

ARC ゾーンオートシフトは、ユーザーに代わって、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、以下の章を参照してください。

- ARC でのゾーンシフト
- ARC でのゾーンオートシフト

マルチリージョンリカバリ

リージョンスイッチ

ARC のリージョンスイッチは、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、観測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でフェイルオーバーできます。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ており、定義したガードレールを課すことで、意図しない結果からユーザーを保護するのに役立ちます。たとえば、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により、フェイルオーバートラフィックを処理するようにスケーリングおよび設定された状態でマルチリージョンアプリケーションを維持できます。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、以下の章を参照してください。

- ARC でのリージョンスイッチ
- ARC でのルーティングコントロール
- ARC での準備状況チェック

ARC でマルチ AZ とマルチリージョンの復旧機能を比較する

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、リージョンスイッチはすべて、迅速な復旧を実現し、 AWS アプリケーションの耐障害性を確保するのに役立ちます。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の増加や可用性の低下が発生している場合の復旧をサポートします。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で、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が制限されます。

ルーティングコントロールとリージョンスイッチは、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シフトのみをサポートします。

次の表の情報には、ARC 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	リージョンスイッチ	ゾーンシフト	ゾーンオートシフト
リージョン別 あるリージョンから別の AWS リージョンにトラフィックをルーティングする (主に)	リージョン別 あるリージョンから別の AWS リージョンにトラフィックをルーティングする (主に)	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	リージョンスイッチ	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 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 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 は、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) サポートされているリソースには、AZs からトラフィックを移動するためのゾーンシフトの作成が含まれています	サービスに付属 (追加料金なし) オートシフトを開始してユーザーに代わってトラフィックを AZs から遠ざけることは、サポートされているリソースに含まれます。
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 オートシフトの開始と終了

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 でのゾーンシフト
- ARC で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 でのリージョンスイッチ

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) の問題からリソースを確実に復旧 AWS 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を短縮できます。

ゾーンシフトとゾーンオートシフトの主な違いは、1 つ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 1 つ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ことです。

- ゾーンシフトでは、 でサポートされているリソースのトラフィックを手動でアベイラビリティーゾーンから遠 AWS リージョン ざけます。
- 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 AZs に再ルーティングされます。

以下のトピックでは、ゾーンシフトとゾーンオートシフトの機能、およびそれらの使用方法について説明します。

トピック

- ARC でのゾーンシフト
- ARC でのゾーンオートシフト

ARC でのゾーンシフト

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。リソースのトラフィックを障害のある AZ から遠ざけると、停電や AZ のハードウェアまたはソフトウェアの問題による影響の期間と重要度が軽減され、問題を軽減し、アプリケーションをすばやく復旧できます。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 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遠ざけ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。事前スケーリング後、移行する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。いつでもシフトをキャンセルして、トラフィックが元のアベイラビリティーゾーンに戻るようにすることができます。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、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。

特定のシナリオでは、ゾーンシフトはトラフィックを AZ から遠ざけません。詳細については、「サポート 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のゾーンシフトを開始すると、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) から遠ざけられます。ARC でサポートされているリソースは、指定された AZ を異常としてマークする統合を提供するため、障害のある AZ からトラフィックが移行します。

トラフィックがシフトし始める - ARC でゾーンシフトを開始すると、トラフィックがすぐにアベイラビリティーゾーンから移動しないことがあります。クライアントの動作と接続の再利用によっては、アベイラビリティーゾーン内の進行中の既存の接続が完了するまでに短い時間がかかる場合があります。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、「トラフィックシフトが迅速に終了するようにする」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キャンセルすると、ARC はトラフィックのシフトを停止するステップを実行し、トラフィックシフトを開始するプロセスを逆にします。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 に流れるトラフィックが再開されます。

シフトを開始するときに、すべてのゾーンシフトの有効期限が切れるように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。

トラフィックが離れない場合 - 特定のシナリオでは、ゾーンシフトはアベイラビリティーゾーンからのトラフィックをシフトしません。たとえば、AZ のロードバランサーターゲットグループにイン

スタンス AZs がない場合、またはすべてのインスタンスが異常である場合に、ロードバランサーのゾーンシフトを開始するとします。このシナリオでは、ロードバランサーはフェイルオープン状態で、ゾーンシフトを開始してもトラフィックは移行しません。

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 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の詳細については、Amazon Web Services 全般のリファレンスの「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。ゾーンシフトとゾーンオートシフトは、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、追加の考慮事項がある場合があります。詳細については、「サポート 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	HTTPS
		arc-zonal-shift-fips.us-west-2.api.aws	HTTPS
		arc-zonal-shift.us-west-2.api.aws	HTTPS
アフリカ (ケープタウン)	af-south-1	arc-zonal-shift.af-south-1.amazonaws.com	HTTPS
		arc-zonal-shift.af-south-1.api.aws	HTTPS
アジアパシフィック (香港)	ap-east-1	arc-zonal-shift.ap-east-1.amazonaws.com	HTTPS
		arc-zonal-shift.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-zonal-shift.ap-south-2.amazonaws.com	HTTPS
		arc-zonal-shift.ap-south-2.api.aws	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	HTTPS
		arc-zonal-shift.ap-southeast-3.api.aws	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	HTTPS
		arc-zonal-shift.ap-southeast-5.api.aws	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	HTTPS
		arc-zonal-shift.ap-southeast-4.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 (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 (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
アジアパシフィック (タイ)	ap-southeast-7	arc-zonal-shift.ap-southeast-7.amazonaws.com arc-zonal-shift.ap-southeast-7.api.aws	HTTPS HTTPS
アジアパシフィック (東京)	ap-northeast-1	arc-zonal-shift.ap-northeast-1.amazonaws.com arc-zonal-shift.ap-northeast-1.api.aws	HTTPS HTTPS

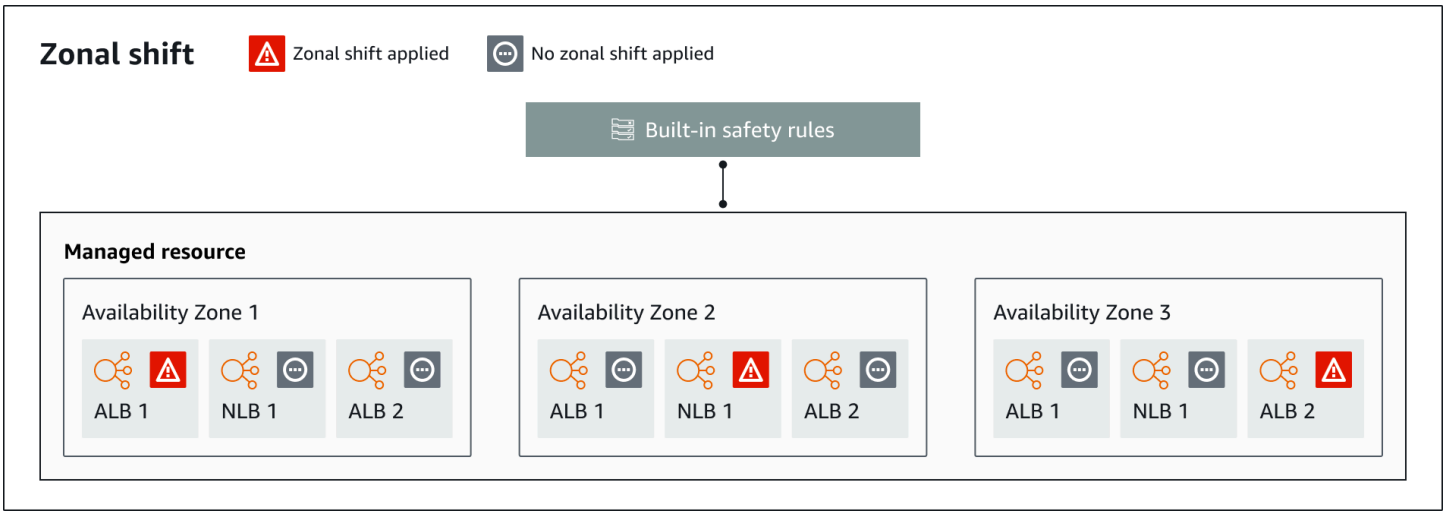
リージョン名	リージョン	エンドポイント	プロトコル
カナダ (中部)	ca-central-1	arc-zonal-shift.ca-central-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-central-1.api.aws	HTTPS
		arc-zonal-shift.ca-central-1.api.aws	HTTPS
カナダ西部 (カルガリー)	ca-west-1	arc-zonal-shift.ca-west-1.amazonaws.com	HTTPS
		arc-zonal-shift-fips.ca-west-1.api.aws	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
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

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 (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
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

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、 のアベイラビリティーゾーンからトラフィックを移行するゾーンシフトの例を示しています AWS リージョン。ゾーンシフトに組み込まれているチェックは、アクティブなシフトがすでにある場合に、リソースの別のゾーンシフトを開始できないようにします。



以下は、ARC の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、「」を参照してくださいサポート リソース。

組み込み安全チェック

ARC に組み込まれているチェックにより、リソースの複数のトラフィックシフトが一度に有効になるのを防ぐことができます。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、リソースに対してお客様が開始したゾーンシフト、練習実行、またはオートシフトを 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 で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、「」を参照してくださいサポート リソース。

マネージドリソース

一部の AWS リソースは手動でゾーンシフトに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、「」を参照してくださいサポート リソース。

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

適用されたステータスは、シフトがリソースに対して有効かどうかを示します。ステータスのシフトによって、リソースのアプリケーショントラフィックが移行されたアベイラビリティーゾーンと、そのシフトが終了するタイミングが`APPLIED`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。には次の値`shiftType`を指定できます。

- ゾーンシフト

- 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。ゾーンシフトの場合、最初にゾーンシフトを最大 3 日間 (72 時間) アクティブに設定できます。

ゾーンシフトを開始するときは、アクティブにする期間を指定します。ARC はこれを有効期限 (有効期限) に変換します。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。通常、最大限の信頼性と耐障害性を実現するために、可能な限りメカニズムにデータプレーン関数を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。どちらも信頼性が高いように構築されていますが、データ整合性のためにコントロールプレーンが最適化され、可用性のためにデータプレーンが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 で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のゾーンシフトを開始して、アベイラビリティーゾーンの問題からアプリケーションを復旧できます。ゾーンシフトは追加料金なしで使用できます。

ARC の料金と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 でのマルチ AZ リカバリにゾーンシフトを使用するための以下のベストプラクティスをお勧めします。

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
- ゾーンシフトのみを使用してトラフィックを一時的に移動する

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のゾーンシフトを開始し、トラフィックが AZ から遠ざけると、アプリケーションがサービスリクエストに使用していた容量が削除されます。AZ からのトラフィックの移行を計画し、残りの AZ で引き続きリクエストを処理できるようにする必要があります AZs 。

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一般的に正常であり、リージョンのアベイラビリティーゾーンでトラフィックをアクティブにとっていることを確認することが重要です。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。

隣接する 2 番目のリージョンからリソースの状態をモニタリングすることを検討してください。このアプローチの利点は、エンドユーザーのエクスペリエンスをより代表し、アプリケーションとモニタリングの両方が同時に同じ災害の影響を受けるリスクを軽減できることです。

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

依存関係がほとんどなく、アプリケーションを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、事前に保存された認証情報とともに使用することをお勧めします。使いやすさを考慮して AWS Management Console、 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

次の表に、マルチ AZ アプリケーション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ゾーンシフトを使用して使用できる ARC API オペレーションを示します。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 でのゾーンシフト」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 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 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、ゾーンシフトを使用するアプリケーション例を示します。 AWS Command Line Interface 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を操作します。この例は、 CLI を使用したゾーンシフトの操作方法の基本的な理解に役立つことを目的としています。

ARC のゾーンシフトにより、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一時的に移動できるため、アプリケーションは 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 AWS リージョン。

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 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
- Amazon Elastic Kubernetes Service
- アプリケーション ロード バランサー クロスゾーン負荷分散を有効または無効にする
- Network Load Balancers クロスゾーン負荷分散を有効または無効にする

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際には、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前に、それが 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 Management Console、リソースの識別子で `get-managed-resource`オペレーションを使用します。
- リソースでゾーンシフトを開始するには、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 にデプロイする必要があります。移行する AZ があるリージョンと同じリージョンでゾーンシフトを開始し、トラフィックを移行するリソースも同じ AZ とリージョンにあることを確認してください。
- リソースでゾーンシフトを使用するための正しい 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、「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にある場合、ゾーンシフトは効果がありません。ゾーンシフトは AZ を強制的に異常にし、ロードバランサーが開いているときにリージョン内の他の AZs にトラフィックをシフトできないため、これは予想される動作です。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 フェイルオーバーの使用」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 フェイルオーバーの使用」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、ゾーンシフトされていない場合でも、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. 他のサービスとの統合ページで、ARC ゾーンシフトの場合は、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、「異常を無視する」または「異常を置き換える」を選択します。に設定すると`replace-unhealthy`、異常な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。に設定すると`ignore-unhealthy`、アベイラビリティーゾーンの異常なインスタンスはアクティブなゾーンシフトに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 2 つのオプションがあります。

- ZonalShiftEnabled – に設定するとtrue、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。に設定するとfalse、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。を 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false。
- ImpairedZoneHealthCheckBehavior – に設定するとreplace-unhealthy、異常な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。に設定するとignore-unhealthy、アベイラビリティーゾーンの異常なインスタンスはアクティブなゾーンシフトに置き換えられません。

次の例では、 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my-asg。

```
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. 画面の上部のナビゲーションバーで、Auto Scaling グループを作した AWS リージョン 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選択します。

   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、「異常を無視する」または「異常を置き換える」を選択します。に設定するとreplace-unhealthy、異常な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。に設定するとignore-unhealthy、アベイラビリティーゾーンの異常なインスタンスはアクティブなゾーンシフトに置き換えられません。
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)

updateupdate-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 2 つのオプションがあります。

- ZonalShiftEnabled – に設定するとtrue、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。に設定するとfalse、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。を 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false。
- ImpairedZoneHealthCheckBehavior – に設定するとreplace-unhealthy、異常な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。に設定するとignore-unhealthy、アベイラビリティーゾーンの異常なインスタンスはアクティブなゾーンシフトに置き換えられません。

次の 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
  }'
```

ゾーンシフトをトリガーするには、「」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。

Auto Scaling グループの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持つ Auto Scaling グループがあるとします。

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

で障害に気づus-east-1aき、ゾーンシフトをトリガーします。でゾーンシフトがトリガーされると、次の動作が発生しますus-east-1a。

- 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。

アベイラビリティーゾーンのヘルスチェック動作選択の障害	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 will not be replaced in the Avail

アベイラビリティーゾーンのヘルスチェック動作選択の障害	ヘルスチェックの動作 ability Zone with the active zonal shift (`us-east-1a`).

ゾーンシフトを使用するためのベストプラクティス

ゾーンシフトを使用するとき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、継続的なアベイラビリティーゾーンの障害イベントがあるかどうかを判断します。詳細については、「Automating Amazon EC2 Auto Scaling with Event Bridge」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するのに十分な容量があることを確認します。
- 最小正常率が 100 のインスタンスメンテナンスポリシーを設定します。この設定では、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。

プリ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 グループの ARC で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、「Amazon Application Recovery Controller (ARC) デベロッパーガイド」の「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ご利用のお客様には、以下もお勧めします。

- アベイラビリティーゾーンディストリビューションにのみバランス型を使用します。
- 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、キャパシティがすべての

アベイラビリティーゾーン間でバランスが取れるまで待ちます。 は、ロードバランサーのゾーンシフトをキャンセルする前に待ちます。

- 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 Management Console 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。

Amazon Elastic Kubernetes Service

Amazon EKS には、アベイラビリティーゾーン (AZ) 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耐障害性を高める機能が用意されています。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 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 EKS ゾーンシフトを有効にしてアベイラビリティーゾーンの障害を回避する」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. ゾーンシフトの見出しで、[管理] ボタンを選択します。
6. EKS ゾーンシフトの有効化または無効化を選択します。

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トリガーすることも、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 クラスターゾーンシフトを有効にすると、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、ゾーンシフトをトリガーしたり、ゾーンオートシフトを有効にしたりできます。

ゾーンシフトのトリガーの詳細については、「」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。

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」トピック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 ゾーンシフト中、以下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。
- マネージドノードグループ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 Data Plane ノードが正常な AZs でのみ起動されるように Auto Scaling Group (ASG) 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、トラフィックが引き続きフルキャパシティーを持つ 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 のゾーンシフトの使用

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利用を開始する前に、以下を確認してください。

- 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。
- 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. ナビゲーションページ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
```

ゾーンシフトのトリガーの詳細については、「」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。

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。デフォルトでは、Application Load Balancer は HTTP クライアントのキープアライブ期間値を 3600 秒、つまり 1 時間に設定します。300 秒など、アプリケーションの目標復旧時間に合わせて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期間を選択する場合、この値は、一般的に再接続の頻度が高くなり、レイテンシーに影響する可能性があり、すべてのクライアントをより迅速に障害のある AZ またはリージョンから遠ざけることのトレードオフであると見なしてください。

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、ターゲットへのすべてのトラフィックが影響を受けるアベイラビリティーゾーンでブロックされ、DNS からゾーン IP アドレスが削除されます。

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 Balancers

Network Load Balancer 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 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たすべてのターゲットグループが次の要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を有効にするか、 に設定する必要がありますuse_load_balancer_configuration。
  - 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の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。
- 異常なターゲットの接続終了を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、「異常なターゲットの接続の終了」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 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、Network Load Balancer からゾーンシフトを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した場合、Network Load Balancer は Application Load Balancer とそのターゲットへのトラフィックの送信を停止しません。

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. ナビゲーションページ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 名を選択します。
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
```

ゾーンシフトのトリガーの詳細については、「」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。

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誘発し、ゾーンシフトをトリガーすると、障害のある AZ の Network Load Balancer ノードが DNS から削除されます。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によって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」トピックを参照してください。 Load Balancer

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

このセクションのステップでは、Amazon Application Recovery Controller (ARC) コンソールでお客様が開始した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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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. トラフィックを遠ざける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、トラフィックを遠ざけます。
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

このセクションのステップでは、Amazon Application Recovery Controller (ARC) コンソールで開始またはキャンセルしたゾーンシフトを更新する方法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、「」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。

ゾーンシフトを更新するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。

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) で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

- 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 AWS CloudTrail

ARC のゾーンシフトは 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、ゾーンシフトのすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、ARC コンソールからの呼び出しと、ゾーンシフトの ARC API オペレーションへのコード呼び出しが含まれます。

証跡を作成する場合は、ゾーンシフトのイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信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 で収集された情報を使用して、ARC に対してゾーンシフト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、リクエスト者、リクエスト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、アカウントの作成 AWS アカウント 時に で有効になります。ゾーンシフトのアクティビティが ARC で発生すると、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 のゾーンシフトのイベントなど 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。追跡により、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 root 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。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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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

- ゾーンシフトと IAM の連携方法
- 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

ゾーンシフトと IAM の連携方法

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 機能

IAM の機能	ゾーンシフトのサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	Yes
ACL	いいえ
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	はい
プリンシパル権限	はい
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 機能とどのように連携するか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 ユーザーグループ、ユーザーのグループ、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。ID ベースの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。プリンシパルは、それが添付されている

ユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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 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、「」を参照してくださいAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例。

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。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

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。

ゾーンシフトの ARC アクションのリストを確認するには、「サービス認可リファレンス」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。

ゾーンシフトの 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を使用します。

```
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。たとえば、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*"
```

ゾーンシフトの ARC アイデンティティベースのポリシーの例については、「」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。

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。ステートメントにはResource または NotResource 要素を含める必要があります。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。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限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 ARNs「サービス認可リファレンス」の次のトピックを参照してください。

- Amazon Route 53 で定義されるアクション - ゾーンシフト

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、「サービス認可リファレンス」の次のトピックを参照してください。

- Amazon Route 53 で定義される条件キー - ゾーンシフト

ゾーンシフトの ARC アイデンティティベースのポリシーの例については、「」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。

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 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。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。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。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を確認するには、「サービス認可リファレンス」の次のトピックを参照してください。

- Amazon Route 53 で定義される条件キー - ゾーンシフト

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、「サービス認可リファレンス」の以下のトピックを参照してください。

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

ゾーンシフトの ARC アイデンティティベースのポリシーの例については、「」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。

ARC のアクセスコントロールリスト (ACLs)

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 を使用した属性ベースのアクセスコントロール (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制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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、ゾーンシフトのために 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

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドAWS のサービス の「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の方法 AWS Management Console を使用して にサインインする場合、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。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 の一時的セキュリティ認証情報」を参照してください。

ARC のクロスサービスプリンシパルアクセス許可

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。

アクションがポリシーで追加の依存アクションを必要とするかどうかを確認するには、「サービス認可リファレンス」の次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。

ゾーンシフトでは、サービスにリンクされたロールは使用されません。

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 機能と IAM および アクセス許可の一般的な仕組みについては、概要トピック「」の情報を参照してくださいARC でのゾーンシフトの Identity and Access Management。

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 リソース」を参照してください。

IAM 概要トピックで説明されているアクセス許可に加えて、IAM および アクセス許可のゾーンシフトには以下が適用されます。

- ARC でゾーンシフトを操作するために必要なアクセス許可があることを確認してください。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 のアカウントで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操作するために、IAM で Elastic Load Balancing 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、ユーザーとロールには ARC リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの 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。

各リソースタイプの ARNs「サービス認可リファレンス」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、アカウント内で誰かが ARC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。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。例えば、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 AWS 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 Access Analyzer 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 Access Analyzer 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のセットが必要です。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 Management Console、次のようなポリシーをユーザーにアタッチします。

```
{
    "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遠ざけ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、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。次に例を示します。

```
{
    "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 で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、影響が発生していないリソースに対してトラフィックが移行される可能性があります。

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 が練習実行のために開始するゾーンシフトは、オートシフト中のアベイラビリティーゾーンからのトラフィックの移行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。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を理解するのに役立つ `SUCCEEDED`や `FAILED`などの結果を提供します。

> **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内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容量を事前に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。自動スケーリングに依存すると、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するか、練習実行を設定する場合は、アプリケーションリソース容量を事前にスケーリングした後、1 つのアベイラビリティーゾーンなしで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。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ら、オンデマンドの練習実行ゾーンシフトを開始して評価することで、アベイラビリティーゾーンから離れたトラフィックでアプリケーションが正常に動作し続けることができることを確認することをお勧めします。次に、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のに役立ちます。

ゾーンシフトを使用したテストが有効であることを確認するには、移行元の AZ から想定どおりにトラフィックが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 の両方が、これをモニタリングするために使用できる Amazon CloudWatch の 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、トラフィックが想定よりも長く移行した AZ に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、サポートされてい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 コンソールで、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離れたリソースのトラフィックのシフトを 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 1 つのアベイラビリティーゾーンから定期的に移行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ごとに、練習実行を約毎週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。

練習実行ごとに、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とき
- ARC が練習実行をスケジュール、開始、終了するとき
- 練習実行のキャパシティチェック
- 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
- ブロックされた日付とブロックされた時間枠 (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 が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 AWS 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、 はオートシフトでトラフィックを移行することで、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 でサポートされているリソースは、指定された AZ を異常としてマークする統合を提供するため、障害のある AZ からトラフィックが移行されます。

リソースのゾーンオートシフトを有効にする場合は、リソースの練習実行も設定する必要があります。 AWS は、リージョン内のアベイラビリティーゾーンの 1 つなしでアプリケーションを実行するのに十分な容量を確保するために、約 30 分間毎週練習実行を実行します。

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 で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とき

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、 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事前にスケーリングしたお客様向けに設計された機能です AWS リージョン。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、終了するとき

ARC は、リソースの練習実行を毎週約 30 分間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、管理します。ARC は、同じアカウントのリソースの練習実行をバッチ処理しません。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、および 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。たとえば、練習実行中にオートシフトが開始された場合、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。このシナリオでは、ARC は練習実行を中断して終了します。

さらに、ARC がリソースのスケジュール練習実行を開始しないシナリオがいくつかあります。

リソースの中断およびブロックされた練習実行に応答して、ARC は以下を実行します。

- リソースの練習実行が進行中の間に中断された場合、ARC は毎週の練習実行が終了したと見なし、リソースの新しい練習実行を来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スケジュールされている場合、ARC は練習実行を開始しません。ARC は定期的なモニタリングを継続し、1 つ以上のブロッキング制約がまだあるかどうかを判断します。ブロック制約がない場合、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継続するのを停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 は通常のように次の練習実行を 1 週間で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のために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。

練習実行のキャパシティチェック

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に、ARC はチェックを実行して、AZ からトラフィックを安全に遠ざけるのに十分な容量が他のアベイラビリティーゾーンにあることを確認します。十分な容量がない場合、練習実行のトラフィックシフトは開始されず、練習実行は終了します。

さらに、ARC は、ゾーンオートシフトが完了すると、ロードバランサーリソースのキャパシティチェックを実行します。その後、ARC は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終了します。オートシフトの終了時に容量チェックが失敗した場合、トラフィックは移動元のアベイラビリティーゾーンに戻されません。

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、キャパシティチェックは、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確認します。具体的には、キャパシティチェックでは、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。キャパシティチェックの場合、バランスとは、各アベイラビリティーゾーンの正常なキャパシティが他のゾーンと同等であり、わずかな差異があることを意味します。

キャパシティチェックは、Lambda タイプのターゲットグループを持つ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的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。

キャパシティチェックが失敗した場合

キャパシティチェックでリソースに対して使用可能なキャパシティのバランスが取れていないことがわかった場合、練習実行の結果は になりますCAPACITY_CHECK_FAILED。キャパシティチェックが失敗した理由の詳細については、 のコメントフィールドを参照してくださいZonalShiftSummary。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、以下を実行します。

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。

FOr の例では、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、キャパシティ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ついては、 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 通知を設定することで、リソースの練習実行とオートシフトに関する通知を受け取るように選択できます。Autoshift 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。 Autoshift 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 Autoshift 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。

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

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、1 つの練習のみが、リソースのゾーンシフト、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 は優先順位に従って、リソースに対して有効なゾーンシフトタイプを決定します。

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が他のシフトタイプよりも優先されることです。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 の優先順位を説明するために、シナリオ例で優先順位がどのように機能するかを次に示します。

ゾーンシフトタイプが適用されました	ゾーンシフトタイプが開始されました	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	練習実行はキャンセルされ、結果は に設定されINTERRUPTED 、ゾーンシフトが適用されます。
練習実行	AWS FIS 実験	練習実行はキャンセルされ、結果は に設定されINTERRUPTED AWS FIS 、実験が適用されます。
練習実行	ゾーンオートシフト	練習実行はキャンセルされ、結果は に設定されINTERRUPTED 、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	練習実行は開始できません。

ゾーンシフトタイプが適用されました	ゾーンシフトタイプが開始されました	結果
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 リソースAPPLIEDにありますACTIVEが、 にはありません。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 リソースAPPLIEDにありますACTIVEが、 にはありません。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、練習実行は開始されません。

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 に設定されますがNOT_APPLIED、 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、 は毎週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練習実行の実行を 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って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

自動シフトおよび練習実行ゾーンシフトの場合、トラフィックは、ARC が顧客主導の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、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 から取り消し、トラフィックがアベイラビリティーゾーンからリダイレクトされるようにします。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逆転させ、Route 53 ヘルスチェックの元に戻すことをリクエストします。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、アベイラビリティーゾーンが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するようリクエストすることで、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、オートシフトまたはゾーンシフトが終了したときにヘルスチェックを再び正常に戻します。

練習実行のアラーム

ゾーンオートシフトでの練習実行には、2 つの CloudWatch アラームを指定できます。最初のアラーム、つまり結果アラームは必須です。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。

練習実行を効果的に実行するには、リソースまたはアプリケーションのメトリクスを監視する CloudWatch アラームを結果アラームとして指定します。このアラームは、1 つのアベイラビリティーゾーンが失われたことでアプリケーションが悪影響を受けたときに ALARM 状態を返します。

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。

結果アラームは、ARC が練習実行ごとに報告する練習実行結果に関する情報も提供します。アラームが ALARM 状態になると、練習実行は終了し、練習実行の結果は FAILED として返されます。練習実行が予定されている 30 分間のテスト期間を完了しても結果アラームが ALARM 状態にならない場合、結果は SUCCEEDED として返されます。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。

オプションで、2 つ目のアラーム、ブロッキングアラームを指定できます。ブロッキングアラームは、練習実行が ALARM 状態のときに練習実行の開始または続行をブロックします。このアラームは、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。

ブロックされた日付とブロックされた時間枠 (UTC)

特定の暦日、または特定の時間枠、つまり UTC での日時の練習実行をブロック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)。

AWS リージョン ゾーンオートシフトの可用性

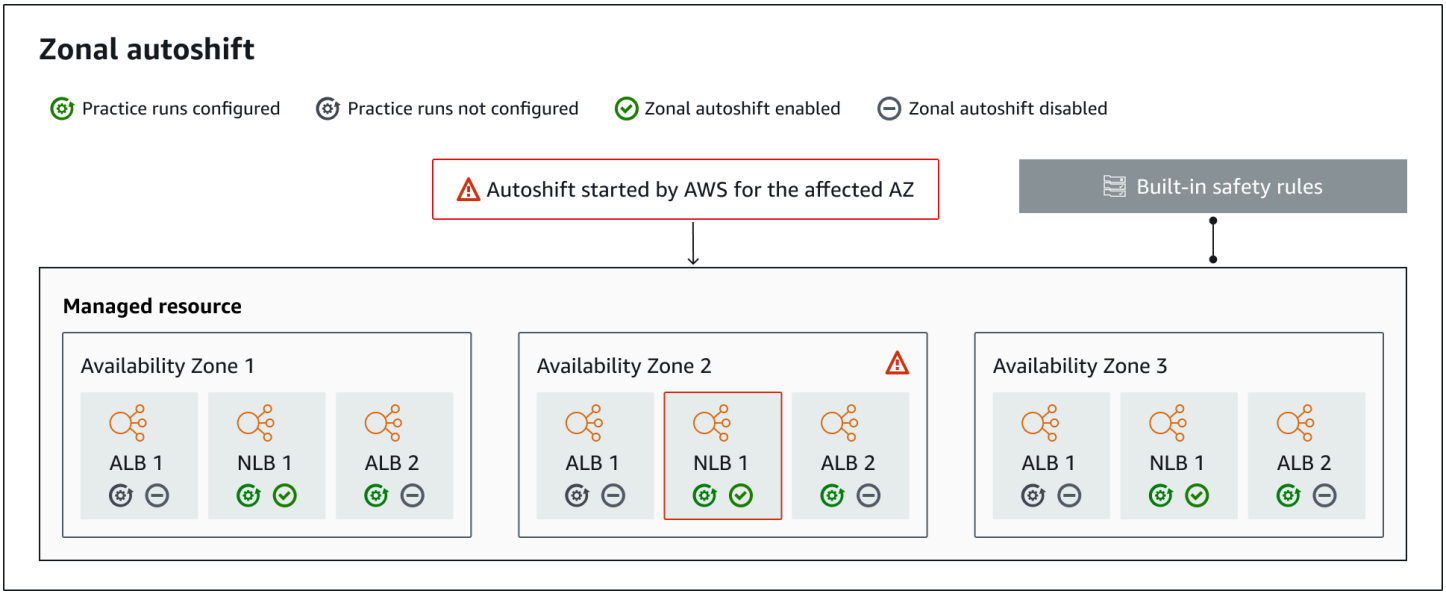
ゾーンシフトとゾーンオートシフトは現在 AWS リージョン、商用および中国リージョン、つまり中国 (北京) リージョンと中国 (寧夏) リージョンで利用できます。

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 リソース」を参照してください。

ARC のリージョンサポートとサービスエンドポイントのリストと詳細については、「Amazon Web Services 全般のリファレンス」の「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。 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。



以下は、ARC の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。練習実行をオンデマンドでスケジュールすることもできます。

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

練習実行設定では、ブロックされた日付とウィンドウ、およびゾーンオートシフトのリソースの AWS 練習実行に指定した CloudWatch アラームを定義します。練習実行設定はいつでも編集でき、ブロックされた日付やウィンドウ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、リソースの練習実行設定が必要です。練習実行を削除することもできます。リソースの練習実行設定を削除するには、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、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移動しなくなります。

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。結果アラームは必須であり、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、練習実行ごとに結果を報告します。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: ロードバランシングと Auto Scaling グループリソースのアベイラビリティーゾーン間でバランスの取れた容量のチェックに失敗しました。

組み込みの安全ルール

ARC に組み込まれた安全ルールにより、リソースの複数のトラフィックシフトが一度に有効になるのを防ぐことができます。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移行できるのは、お客様が開始したゾーンシフト 1 つ、練習実行ゾーンシフト (お客様または AWS お客様が開始) 、またはリソースの自動シフトのみです。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。リソースの Amazon リソースネーム (ARN) です。ゾーンオートシフトは、ARC でサポートされている AWS サービスにあるアカウント内のリソースに対してのみ有効にできます。

マネージドリソース

Application Load Balancer は、ゾーンオートシフト用にリソースを ARC に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を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。

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、一度にリソースに適用されるのは 1 つだけです。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。ゾーンシフトには、次のいずれかのタイプがあります。

- ゾーンシフト
- 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。通常、最大限の信頼性と耐障害性を実現するために、可能な限りメカニズムにデータプレーン関数を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 で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 の料金と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ときは、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含まれます。

- 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するか、ゾーンシフト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には、いくつかのベストプラクティスと考慮事項があります。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限に注意する
- 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時間を設定できます。300 秒など、アプリケーションの目標復旧時間に合わせて`keepalive`値を小さくすることをお勧めします。`keepalive` 時間を選択するときは、この値が一般的に再接続頻度が高いことによるトレードオフであり、レイテンシーに影響を与える可能性があり、障害のある AZ またはリージョンからすべてのクライアントをより迅速に遠ざけることができることを考慮してください。

Application Load Balancer `keepalive`のオプションの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合計 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リング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 でのゾーンシフト」を参照してください。

リソースタイプと制限に注意する

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既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開かれていない (異常) AWS リージョン 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 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 つのアラーム (結果アラーム) を設定します。必要に応じて、2 番目のアラーム (ブロッキングアラーム) を設定することもできます。

リソースの練習実行用に設定した CloudWatch アラームを検討するときは、次の点に注意してください。

- 必須の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。練習実行を終了して AWS 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状態ではなくなった場合、ARC は次にリソースに対して練習実行がスケジュールされたときに練習実行を開始します。
- 練習実行ブロッキングアラームでは、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。
- 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。これらは、まず Amazon CloudWatch で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。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、練習実行ごとに結果を報告します。練習実行後、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の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、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。
- FAILED: 練習実行中に結果アラームが 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。
  - 優先順位が の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: ロードバランシングと Auto Scaling グループリソースのアベイラビリティーゾーン間でバランスの取れた容量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。

アクション	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 で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 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 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。
Autoshift 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。

アクション	ARC コンソールの使用	ARC API の使用
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」を参照してください。	「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。この例は、 CLI を使用してゾーンオートシフトを操作する方法の基本的な理解に役立つ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、「」を参照してくださいサポート リソース。

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。

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

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

たとえば、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

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

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ことで、 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、 を使用しますcancel-zonal-shift command。

```
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

CLI を使用してリソースの練習実行設定を編集して、練習実行のアラームの変更、ARC が練習実行を開始しない場合のブロックされた日付やブロックされたウィンドウの更新など、さまざまな設定オプションを更新できます。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使用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したら、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のシフトのキャンセル、オートシフトオブザーバー通知の有効化を行うことができます。

ゾーンオートシフトの有効化または無効化

ここでは、Amazon Application Recovery Controller (ARC) コンソールでゾーンオートシフトを有効または無効にする方法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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンオートシフト] を選択します。

3. [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。
4. Actions メニューで、Enable zonal autoshift を選択し、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
- Autoshift 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

このセクションのステップでは、Amazon Application Recovery Controller (ARC) コンソールで練習実行設定を編集または削除する方法を説明します。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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンオートシフト] を選択します。
3. [ゾーンオートシフトを設定] を選択します。
4. ゾーンオートシフトを設定するリソースを選択します。
5. 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。

6. リソースの練習実行のオプションを選択します。例えば、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。
7. オプションで、ブロックする日付とブロックする時間枠を指定します。ARC がこのリソースの練習実行を開始するのをブロックする日付またはウィンドウ (日時) を選択します。すべての日付と時刻は UTC で表示されます。
8. 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。
9. [作成] を選択します。

練習実行設定を編集するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンオートシフト] を選択します。
3. [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。
4. [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。
5. 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については、アラームを追加したり、アラームを削除したり、別のブロッキングアラームを指定したりできます。
     - 練習実行を監視する結果アラームについては、使用する別の CloudWatch アラームを指定できます。結果アラームは必須なので、結果アラームを削除することはできません。
   - ブロックされる日付やブロック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。
6. [保存] を選択します。

練習実行設定を削除するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンオートシフト] を選択します。
3. [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。
4. [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。
5. 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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンシフト] を選択します。
3. キャンセルするゾーンオートシフトを選択し、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。

練習実行ゾーンシフトの開始

このセクションのステップでは、ARC コンソールでオンデマンド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練習実行設定を作成した後、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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。

2. [マルチ AZ] で [ゾーンオートシフト] を選択します。
3. ゾーンオートシフトリソースで、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。
4. リソースの概要ページで、練習実行の開始を選択します。
5. アベイラビリティーゾーンを選択し、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。
6. [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

このセクションのステップでは、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を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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [マルチ AZ] で [ゾーンシフト] を選択します。
3. キャンセルする練習実行ゾーンシフトを選択し、ゾーンシフトのキャンセルまたは練習実行の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。

Autoshift 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。

このセクションのステップでは、Amazon Application Recovery Controller (ARC) コンソールを使用してオートシフトオブザーバー通知を有効にする方法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。

Autoshift オブザーバー通知を有効または無効にするには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. 開始方法 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っている可能性があるとき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に対して、障害の可能性がある AZ から自動的にトラフィックを移行し、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 電源の中断が開始されてから 5 分後に、復旧アクションによってリソーストラフィックが指定された AZ から`aws:arc:start-zonal-autoshift`自動的に移行されます。トラフィックは、停電の残りの 25 分間シフトされ、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します。実験が完了すると、トラフィックシフトは終了し、トラフィックはすべての AZs 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ています。

実験とゾーンオートシフトの練習実行の違い

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、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 は優先順位に従って、リソースに対して有効なゾーンシフトタイプを決定します。ゾーンシフトの優先順位の詳細については、「」を参照してくださいゾーンシフトの優先順位。

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) で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

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

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

ARC のゾーンオートシフトは、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、

ゾーンシフトのすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、ARC コンソールからの呼び出しと、ゾーンシフトの ARC API オペレーションへのコード呼び出しが含まれます。

証跡を作成する場合は、ゾーンシフトのイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的な配信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 で収集された情報を使用して、ARC に対してゾーンシフト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、リクエスト者、リクエスト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、アカウントの作成 AWS アカウント 時に で有効になります。ゾーンオートシフトのアクティビティが ARC で発生すると、そのアクティビティは CloudTrail イベントとイベント履歴の他の AWS サービス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 のゾーンオートシフトのイベントなど 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。追跡により、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 root 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。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`アクションを示す CloudTrail ログエントリを示しています。

```
{
      "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、ゾーンオートシフトに対応できます。ゾーンオートシフトのイベントは、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。たとえば、練習実行が開始されたときなどです。ゾーンオートシフトを設定して、サービスで有効にするリソースのゾーンオートシフトイベントを通知することができます。

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、 が障害の可能性がある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。Autoshift オブザーバー通知は、ゾーンオートシフトを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別のものです。自動シフト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、EventBridge がイベントを検出するために使用できるイベント固有のパターンを定義します。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発生します。これらは、通常の運用状況下でほぼリアルタイムで ARC から EventBridge に配信されます。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 を使用してゾーンオートシフトリソースをモニタリングする

EventBridge を使用すると、ARC がリソースのイベントを発行するときに実行するアクションを定義するルールを作成できます。たとえば、ゾーンオートシフトの練習実行が開始されたときに 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。このトピックでは、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、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示します。

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 、つまりイベントの監視対象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 に貼り付けると、ゾーンオートシフトアクションとリソースのモニタリング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`

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。Autoshift 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 の通知を受け取ることを選択します`Autoshift In Progress`。自動シフトオブザーバー通知を有効にする手順については、「」を参照してください [ゾーンオートシフトの有効化と操作]。

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:

  - 自動シフトオブザーバー通知が有効になっている場合、ARC はすべての自動シフトイベントを返します。
  - 自動シフトオブザーバー通知が有効になっていない場合、ARC は、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を示します。

以下は、 Autoshift In Progress アクションのイベント例です。1) Autoshift オブザーバー通知が有効で、2) Autoshift 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です。

```
{
    "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) 自動シフトオブザーバー通知が無効になっており、2) 自動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です。

```
{
    "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":""
        }
    }
}
```

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が送信されるターゲットを指定する必要があります。EventBridge で使用可能なターゲットのリストについては、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、ルールの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と、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 で始まっていることを確認します`/aws/events`。既存のロググループを選択する場合は、 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`/aws/events`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。

コンソールの外部で 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使用して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。

```
{
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

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、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、イベントパターンや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 ルールの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. ターゲットの選択ページで、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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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が IAM と連携する方法
- 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の例
- 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が IAM と連携する方法

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 機能

IAM の機能	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ポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	Yes

IAM の機能	ゾーンオートシフトのサポート
ACL	いいえ
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	はい
プリンシパル権限	はい
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 ユーザーグループ、ユーザーのグループ、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。ID ベースの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。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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 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、「」を参照してくださいAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例。

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。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

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の ARC アクションのリストを確認するには、「サービス認可リファレンス」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。

ゾーンオートシフトの ARC のポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
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。たとえば、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、「」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の例。

ARC で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。ステートメントにはResource または NotResource 要素を含める必要があります。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。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限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 ARNs「サービス認可リファレンス」の次のトピックを参照してください。

- Amazon Route 53 で定義されるアクション - ゾーンシフト

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、「サービス認可リファレンス」の次のトピックを参照してください。

- Amazon Route 53 で定義される条件キー - ゾーンシフト

ゾーンオートシフトの 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、「」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の例。

ARC で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 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。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。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。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。

ゾーンオートシフトの ARC 条件キーのリストを確認するには、「サービス認可リファレンス」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、「サービス認可リファレンス」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、「」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の例。

ARC のアクセスコントロールリスト (ACLs)

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 を使用した属性ベースのアクセスコントロール (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制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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、ゾーンシフトのために 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

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドの「IAM AWS のサービス と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。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 の一時的セキュリティ認証情報」を参照してください。

ARC のクロスサービスプリンシパルアクセス許可

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。

アクションがポリシーで追加の依存アクションを必要とするかどうかを確認するには、「サービス認可リファレンス」の次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、「」を参照してください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。

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の例

デフォルトでは、ユーザーとロールには ARC リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの 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。

各リソースタイプの ARNs「サービス認可リファレンス」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、アカウント内で誰かが ARC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。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。例えば、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 AWS 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 Access Analyzer 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 Access Analyzer 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のセットが必要です。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、ARC で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がユーザーに必要です。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 Management Console、次のようなポリシーをユーザーにアタッチします。

```
{
    "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、ARC に関連付けられたサービスにリンクされたロールに対するアクセス許可がユーザーに必要です。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。

```
{
    "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 AWS 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 Management Console、、 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 でのゾーンシフト」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 AWS 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キャパシティ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アラートについては、ARC ドキュメント履歴ページの RSS フィードに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学習できます。これを使用して、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。

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 での準備状況チェック
- ARC でのリージョンスイッチ

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、クライアントトラフィックをリージョンレプリカ間で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。たとえば、各リージョンのアプリケーションレプリカの前にあ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、およびそれらを使用してフェイルオーバーのためにトラフィックを再ルーティングする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です。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グループ化することも、独自のカスタムコントロールパネルを作成することもできます。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 つのエンドポイントのデータプレーンです AWS リージョン。

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の安全ルールを作成して、意図しない復旧自動化の副作用を防ぐことができます。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 Management Console。

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、それらを作成して使用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する方法について説明します。

> **Important**
>
> ARC を使用して、災害シナリオでアプリケーションのフェイルオーバープランの一部としてトラフィックを再ルーティングする準備については、「」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ベストプラクティス。

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、ARC がそれを更新して、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た後、ルーティングコントロールに関連付けられた Route 53 ヘルスチェックです。

ルーティングコントロールは、DNS エンドポイント AWS を持つサービス間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、ルーティングコントロールの安全ルールを設定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の作成」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できるようにするには、ARC のクラスターエンドポイントのいずれかに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。

また、ARC ルーティングコントロールと Route 53 ヘルスチェックおよび DNS レコードセットを併用し、加重ルーティングポリシーを持つ DNS レコードを使用することで、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、 AWS ブログ記事「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」の「フェイルオーバーに関するセクション、パート 2: マルチリージョンスタック」を参照してください。

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、クライアントの動作と接続の再利用によっては、リージョン内の進行中の既存の接続が完了するまでに短い時間がかかる場合があります。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、「トラフィックシフトが迅速に終了するようにする」を参照してください。

ルーティングコントロールの利点

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でトラフィックを再ルーティングする利点がいくつかあります。例:

- 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。モニタリングおよびヘルスチェックシステムからの情報に基づいて、レプリカとフェイルアウェイするかレプリカとフェイルアウェイするかを決定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、レプリカは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、障害のあるアプリケーションレプリカからユーザーを遠ざけることで、障害から回復できます。

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の詳細については、Amazon Web Services 全般のリファレンスの「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/オフスイッチであり、プライマリデプロイレプリカからスタンバイデプロイレプリカに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。

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制御クエリを作成し、ルーティング制御状態を更新することで、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 に任せることができます。詳細については、「 ARC API を使用したルーティングコントロールの状態の取得と更新 (推奨)」を参照してください。

ARC は、5 つの冗長リージョンエンドポイントのセットであるクラスター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。次に、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。

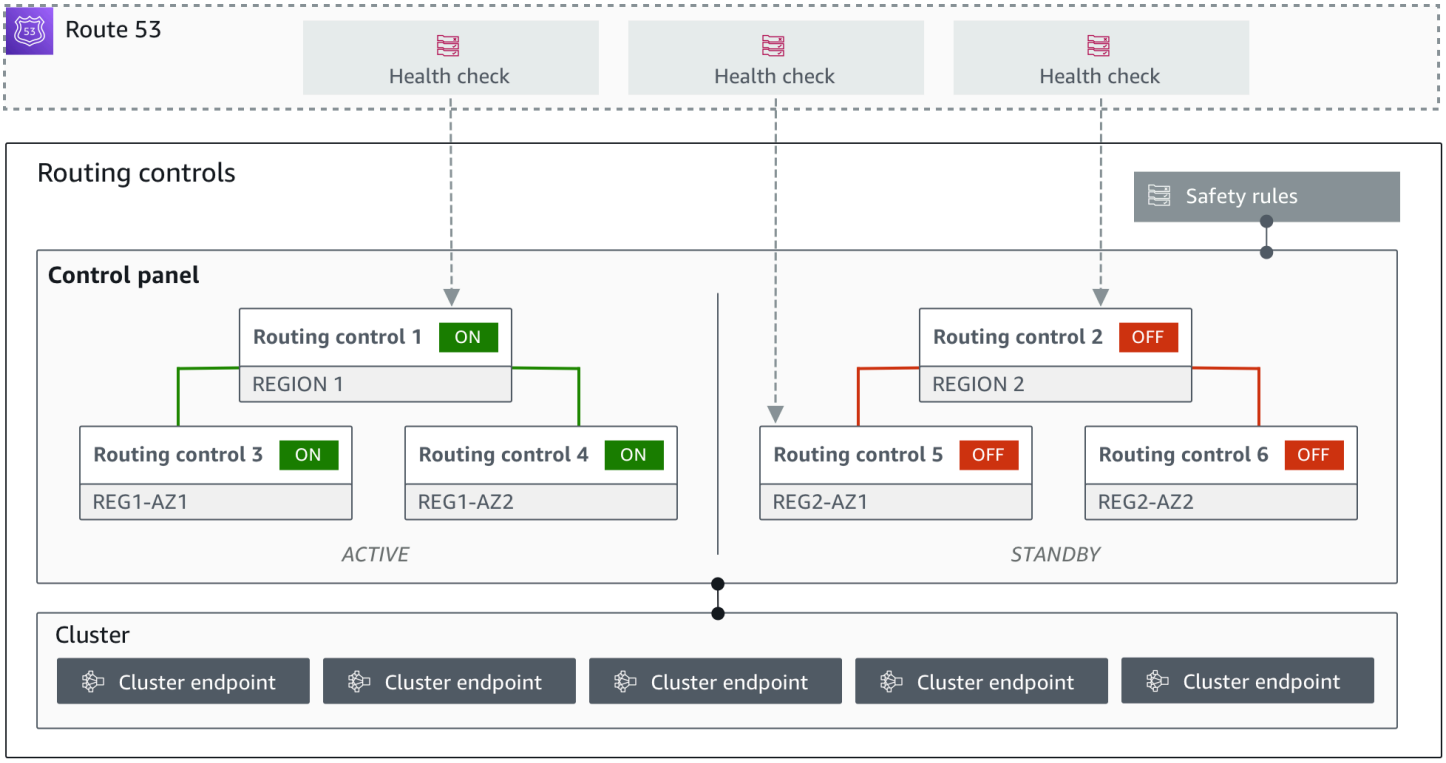
ARC は、セル間でアプリケーションを手動でフェイルオーバーできるように、データプレーンで非常に高い信頼性を提供します。ARC では、5 つのクラスターエンドポイントのうち少なくとも 3 つに常にアクセスして、ルーティングコントロールの状態の変更を実行できます。各 ARC クラスターはシングルテナントであり、アクセスパターンを遅くする可能性のある「ノイズの多い近隣」の影響を受けないようにします。

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 トラフィックを異なる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 の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ます。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するときに、トラフィックを再ルーティングできます (アプリケーションの DNS レコードで設定されたヘルスチェックを使用) 。

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、トラフィックをスタンバイレプリカへのフェイルオーバーなどにリダイレクトします。

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 はデフォルトのコントロールパネルを作成します。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を設定および取得するために使用できる 5 つのリージョン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は、ARC が定期的にエンドポイントをメンテナンスのために上下させると仮定する必要があります。そのため、エンドポイントに接続するまで、各エンドポイントを連続し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。通常、最大限の信頼性と耐障害性を実現するために、可能な限りメカニズムにデータプレーン関数を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、ルーティング制御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。どちらも信頼性が高いように構築されていますが、データ整合性のためにコントロールプレーンが最適化され、可用性のためにデータプレーンが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の場合、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割されます。

- ルーティングコントロール用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 Configuration API です。これらの API オペレーションまたは を使用して AWS Management Console 、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データプレーンは、地理的に分離された AWS 5 つの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の重要な考慮事項の詳細については、「」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ベストプラクティス。

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) でのルーティング制御のタグ付け

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。各リソースには複数のタグを追加でき、各タグにはユーザーが定義したキーと値が含まれています。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、次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、 を使用するなど、 API を介してのみ使用できます AWS CLI。

以下は、 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 CLI。

```
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) のリカバリコントロール設定 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の場合、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 でのクラスターのクロスアカウントのサポート」を参照してください。

ARC の料金と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) でのマルチリージョンリカバリの開始方法

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。ゾーンシフトまたはゾーンオートシフトを使用して、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復旧するためのセットアップは必要ありません。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、相互にレプリカであ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を成功させ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 AWS CloudFormation と を提供しています。詳細とテンプレートのダウンロードについては、「」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. 各リージョンで相互にレプリカであ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使用を開始する

Amazon Application Recovery Controller (ARC) のルーティングコントロールを使用すると、トラフィックのフェイルオーバーをトリガーして、個別に実行されている冗長なアプリケーションコピーまたはレプリカ間でフェイルオーバーできます AWS リージョン。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ように、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ることができます。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 Management Console 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。

たとえば、us-west-1 から us-east-1 までのリージョン間でフェイルオーバーするには、 `update-routing-control-state` API アクションを使用して の状態を `us-west-1` に設定`Off`し、 の状態を `us-east-1` に設定します`On`。

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するために、 というサンプルアプリケーションを提供していますTicTacToe。この例では、 AWS 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 AWS 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。

サンプルアプリケーションをデプロイしたら、 テンプレートを使用して ARC コンポーネントを作成し、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へのトラフィックフローを管理できます。テンプレートとプロセスを独自のシナリオと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 AWS 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。 AWS 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AWS 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のベストプラクティス

ARC でのルーティング制御の復旧とフェイルオーバーの準備には、以下のベストプラクティスをお勧めします。

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データプレーン API を使用して ARC でフェイルオーバータスクを実行するには、ARC IAM ポリシーをユーザーにアタッチします。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たままにな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。

デフォルトでは、Application Load Balancer は HTTP クライアントのキープアライブ期間値を 3600 秒、つまり 1 時間に設定します。300 秒など、アプリケーションの目標復旧時間に合わせて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期間を選択する場合、この値は一般的に再接続の頻度が高いことによるトレードオフであり、レイテンシーに影響する可能性があります。また、すべてのクライアントを障害のある AZ またはリージョンからより迅速に遠ざけることができます。

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 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、エンドポイントの再試行に使用するオートメーションコードに保存することをお勧めします。失敗イベント中に、非常に信頼性の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い場合があります。DescribeCluster API オペレーションを使用して、ARC クラスターのエンドポイントを一覧表示できます。

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は、障害が発生した場合でも高可用性を確保するために 5 つのリージョンエンドポイントを提供します。完全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すべてを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

ARC データプレーン API を使用して、ListRoutingControls オペレーションでルーティングコントロールと状態を表示し、UpdateRoutingControlState オペレーションでフェイルオーバーのためにトラフィックをリダイレクトするようにルーティングコントロール状態を更新します。 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 は、トラフィックをフェイルオーバーするために、データプレーンの API で非常に高い信頼性を提供します。 AWS Management Console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。

ARC がデータプレーン API を使用するには、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。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 によるフェイルオーバーのテスト

ARC 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定期的にテストし、プライマリアプリケーションスタックからセカンダリアプリケーションスタックにフェイルオーバーします。追加した ARC 構造がスタック内の正しいリソースと一致し、すべてが期待どおりに機能することを確認することが重要です。これは、環境に ARC を設定した後でテストし、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する必要があります。その後、ユーザーのダウンタイムを回避するために、セカンダリシステムをすばやく稼働させる必要がある障害状況が発生します。

ルーティング制御 API オペレーション

このセクションには、Amazon Application Recovery Controller (ARC) でのルーティングコントロールの設定と使用に使用できる 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が含まれています。

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については AWS Command Line Interface、「」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。

次の表に、ルーティングコントロール設定に使用できる ARC API オペレーションと、関連するドキュメントへのリンクを示します。

アクション	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 でのルーティングコントロールコンポーネントの作成 」を参照	「CreateCluster」を参照
クラスターを記述する	「ARC でのルーティングコントロールコンポーネントの作成 」を参照	「DescribeCluster」を参照

アクション	ARC コンソールの使用	ARC API の使用
クラスターを削除	「ARC でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」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 でのルーティングコントロールコンポーネントの作成 」を参照	「ListRoutingControls」を参照
コントロールパネルを作成する	「ARC でのルーティングコントロールコンポーネントの作成 」を参照	「CreateControlPanel」を参照
コントロールパネルを説明する	「ARC でのルーティングコントロールコンポーネントの作成 」を参照	「DescribeControlPanel」を参照

アクション	ARC コンソールの使用	ARC API の使用
コントロールパネルを更新する	「ARC でのルーティングコントロールコンポーネントの作成 」を参照	「UpdateControlPanel」を参照
コントロールパネルを削除する	「ARC でのルーティングコントロールコンポーネントの作成 」を参照	「DeleteControlPanel」を参照
コントロールパネルを一覧表示する	「ARC でのルーティングコントロールコンポーネントの作成 」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の作成 」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の作成 」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の作成 」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の作成 」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の作成 」を参照	「ListSafetyRules」を参照
関連付けられた Route 53 ヘルスチェックを一覧表示する	「ARC でのルーティングコントロールのヘルスチェックの作成 」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 でのクラスターのクロスアカウントのサポート」を参照してください。	「GetResourcePolicy」を参照

次の表に、ルーティングコントロールデータプレーンによるトラフィックフェイルオーバーの管理に使用できる一般的な ARC API オペレーションと、関連するドキュメントへのリンクを示します。

アクション	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 Management Console」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 Management Console」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 Management Console」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する 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

SDK ドキュメント	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 (フィードバックを送信)] 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に役立つ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、クラスターにプロビジョニングされたコントロールパネルと呼ばれるグループに整理します。ARC クラスターは、グローバルにデプロイされるエンドポイントのリージョン別セットです。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 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 AWS リージョン。ただし、ほとんどの ARC CLI コマンド`--region us-west-2`では、米国西部 (オレゴン) リージョン、つまり パラメータを指定する必要があります。たとえば、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 は一連のリージョンエンドポイントを提供します。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については、 ルーティング制御 API オペレーション「」および「」を参照してください ルーティング制御 API オペレーション。

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をセットアップする

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あり、5 つの異なるエンドポイントのそれぞれに 1 つずつあります AWS リージョン。ARC 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動作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可用性とシーケンシャル整合性を保証します。

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 のいずれかになりますDUALSTACK。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PENDING中にステータスが 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、ARC は操作する一連のエンドポイントを使用してクラスターを正常に作成しました。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 ルーティングコントロールを整理するための論理グループです。クラスターを作成すると、ARC は というコントロールパネルを自動的に提供しますDefaultControlPanel。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場合、作成PENDING中は のステータスになります。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し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。これらの保護を作成するには、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の作成 」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については、「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」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については、「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」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リソースと同様に、安全ルールはデータプレーンに伝達された後に記述、一覧表示、または削除できます。

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
0123456bbbbbbb0123456bbbbbb012345633333444444
```

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するため、Route 53 はトラフィックを再ルーティングします。(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)。

たとえば、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-ffffff22222",
    "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

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の Amazon Application Recovery Controller (ARC) リソースを作成したら、クラスターを操作して、フェイルオーバーのために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。

作成するクラスターごとに、ARC はクラスターエンドポイントのセットを 5 つに 1 つずつ提供します AWS リージョン。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) 0n 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。システムはリージョンエンドポイントをローテーションし、使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、ARC ルーティングコントロールデータプレーンを使用してトラフィックを再ルーティングし、セル間でフェイルオーバーするための非常に高い信頼性を提供します。フェイルオーバーに ARC を使用するその他の推奨事項については、「」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ベストプラクティス。

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持するか、エンドポイントの再試行に使用する自動化コードでハードコードすることをお勧めします。

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 でのルーティングコントロールコンポーネントの操作

トピック

- ARC でのルーティングコントロールコンポーネントの作成
- ARC でのルーティングコントロールの状態の表示と更新
- ルーティングコントロールの安全ルールの作成
- ARC でのクラスターのクロスアカウントのサポート

ARC でのルーティングコントロールコンポーネントの作成

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 でのリソース共有の詳細については、「」を参照してください ARC でのクラスターのクロスアカウントのサポート。料金の詳細については、「Amazon Application Recovery Controller (ARC) の料金」を参照して、Amazon Route 53 までスクロールダウンします。

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の作成 」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用のコンポーネントを作成する手順は、これらのセクションに含まれています。ARC でのリカバリコントロール設定 API オペレーションの使用については、「」を参照してください ルーティング制御 API オペレーション。

ARC でのクラスターの作成

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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 でのルーティングコントロールの作成

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、回復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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 でのルーティングコントロールのヘルスチェックの作成

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。その後、関連付けられたルーティングコントロールの状態を更新して、Amazon Application Recovery Controller (ARC) でトラフィックを再ルーティングし、 `On`または `Off` に設定できます。

> **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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 でプライマリとセカンダリの 2 つのフェイルオーバーレコードを作成します。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 を使用したルーティングコントロールの状態の取得と更新 (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 でのコントロールパネルの作成

Amazon Application Recovery Controller (ARC) のコントロールパネルを使用すると、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 はデフォルトのコントロールパネルを作成します。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は、このセクションに含まれています。ARC でリカバリコントロール設定 API オペレーションを使用する方法については、「」を参照してください ルーティング制御 API オペレーション。

コントロールパネルを作成するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 でのルーティングコントロールの状態の表示と更新

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 0n 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。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ために、データプレーンの API で非常に高い信頼性を提供します。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 を使用したルーティングコントロールの状態の取得と更新 (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 Management Console

ARC API を使用したルーティングコントロールの状態の取得と更新 (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 Management Console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。

ARC は、ルーティングコントロールが高可用性クラスターに保存されるため、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で、セル (AWS リージョン) 間でフェイルオーバーするための非常に高い信頼性を提供します。ARC は、ルーティングコントロールの状態を変更するために、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 つに常にアクセスできるようにします。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ための詳細な情報とコード例を提供します。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 Management Console

AWS Management Console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 を使用して実行できるように、コンソールで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 データプレーンは非常に高い信頼性を提供しますが、コンソールは可用性が高くありません。このため、ARC API を使用して、本稼働オペレーションの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ことをお勧めします。

フェイルオーバーに ARC を使用するその他の推奨事項については、「」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ベストプラクティス。

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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。

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の作成

複数のルーティングコントロールを同時に操作する場合、意図しない結果を避けるために保護策を講じる必要がある場合があります。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 でのルーティング制御に対してこのような保護を確立するには、安全ルールを作成します。

指定したルーティングコントロール、ルール、およびその他の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状態を変更すると、ARC はルールの設定時に設定した基準が満たされるか、それ以外の場合はルーティングコントロール状態が変更されないことを強制します。

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については、「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」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 オペレーションのプロパティの詳細については、「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」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

このセクションのステップでは、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) でのリカバリおよびルーティングコントロール API オペレーションの使用については、「」を参照してください ルーティング制御 API オペレーション。

安全ルールを作成するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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の詳細については、「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」のAssertionRule オペレーションで提供される情報」を参照してください。ゲートルールの指定の詳細については、「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」のGatingRule オペレーションで提供される情報」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

このセクションのステップでは、ARC コンソールで安全ルールを編集または削除する方法を説明します。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ます。

Amazon Application Recovery Controller (ARC) での API オペレーションの使用については、「」を参照してください ルーティング制御 API オペレーション。

安全ルールを削除するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 でのクラスターのクロスアカウントのサポート

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 ルーティング制御では、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、 で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合計コストをさまざまなチームに割り当てることができ、ARC の利点を低コストで最大化できます。(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、特に複数のアカウントや運用チームに多数のアプリケーションが分散されている場合に、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容易になります。

ARC でクロスアカウント共有を開始するには、 でリソース共有を作成します AWS RAM。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、 を使用するか、 AWS Command Line Interface AWS Management Console 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、クラスター内に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。

クラスターの共有

所有しているクラスターを共有すると、クラスターを共有するために指定した参加者は、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および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route53-recovery-control-config:PutResourcePolicy、そのクラスターの 、route53-recovery-control-config:GetResourcePolicy、および アクセスroute53-recovery-control-config:DeleteResourcePolicy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。また、ARC コンソールと 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 AWS CLI。

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 Management Console 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 Management Console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、クラスター内の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、クラスターに関連するコストが請求されます。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) 料金表」を参照し、「Amazon Application Recovery Controller (ARC)」までスクロールダウンします。

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。

これがどのように機能するかをよりよく理解するには、次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

クラスターあたりの所有者の合計コントロールパネルにはクォータが適用されます。

たとえば、所有者がクラスターあたりのコントロールパネル数に 50 のクォータを持ち、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。

たとえば、所有者がクラスターあたりのルーティングコントロールの数のクォータが 300 で、クラスターにすでに 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。

たとえば、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、「」を参照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) でのルーティング制御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

- 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 AWS CloudTrail

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。キャプチャされた呼び出しには、ARC コンソールからの呼び出しと ARC API オペレーションへのコード呼び出しが含まれます。

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 イベントの継続的な配信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 によって収集された情報を使用して、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、リクエスト者、リクエスト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 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。追跡により、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 によってログに記録され、Amazon Application Recovery Controller のリカバリ準備 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、および 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、`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 root 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) を選択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、ルーティングコントロールを設定するための CreateClusterアクションを示す CloudTrail ログエントリを示しています。

```
{
  "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"
}
```

でのルーティング制御のための 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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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) でのルーティングコントロールと IAM の連携
- ARC でのルーティングコントロール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) 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) でのルーティングコントロールと IAM の連携

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) のルーティング制御で使用できる IAM 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ポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	Yes
ACL	いいえ
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	はい
プリンシパル権限	はい
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 機能とどのように連携するか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 ユーザーグループ、ユーザーのグループ、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。ID ベースの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。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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の ARC アイデンティティベースのポリシーの例については、「」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アイデンティティベースのポリシーの例。

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。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

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を確認するには、「サービス認可リファレンス」の「Amazon Route 53 Recovery Controls で定義されるアクション」および「Amazon Route 53 Recovery Cluster で定義されるアクション」を参照してください。

ARC のルーティングコントロールのポリシーアクションは、操作する API に応じて、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-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の ARC アイデンティティベースのポリシーの例については、「」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アイデンティティベースのポリシーの例。

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。ステートメントにはResource または NotResource 要素を含める必要があります。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。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限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": "*"
```

サービス認可リファレンスでは、ARC に関連する以下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、「サービス認可リファレンス」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 Recovery Cluster で定義されるアクション。

ルーティングコントロールの ARC アイデンティティベースのポリシーの例については、「」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アイデンティティベースのポリシーの例。

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 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。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。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。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 条件キーのリストを確認するには、「サービス認可リファレンス」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 Recovery 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、「サービス認可リファレンス」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 ARNs「Amazon Route 53 Recovery コントロールで定義されるアクション」および「Amazon Route 53 Recovery 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を確認するには、「Amazon Route 53 Recovery Controls で定義されるリソース」および「Amazon Route 53 Recovery Cluster 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、「」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アイデンティティベースのポリシーの例

ARC のアクセスコントロールリスト (ACLs)

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 を使用した属性ベースのアクセスコントロール (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制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初

の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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 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。
- Recovery Cluster 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

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報 AWS のサービス を使用する場合などの詳細については、IAM ユーザーガイドAWS のサービス の「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合は、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用してにアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。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 の一時的セキュリティ認証情報」を参照してください。

ARC のクロスサービスプリンシパルアクセス許可

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。

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、「サービス認可リファレンス」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

サービスにリンクされたロールをサポートします。

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 でのルーティングコントロール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、ユーザーとロールには ARC リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、また

は AWS API を使用してタスクを実行することはできません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの 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。

各リソースタイプの ARNs「サービス認可リファレンス」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のための 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、アカウント内で誰かが ARC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。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。例えば、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 AWS 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 Access Analyzer 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 Access Analyzer 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のための 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のセットが必要です。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 管理ポリシーを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 管理ポリシー」ページまたは「ユーザーへのアクセス許可の追加」を参照してください。

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ユーザーに付与します。

```
{
    "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。

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には、次のようなポリシーをユーザーにアタッチします。

```
{
    "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": "*"
```

```
        }
    ]
}
```

障害発生時にルーティングコントロールの状態をフェイルオーバーするように更新するなど、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で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できます。

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、安全ルールの上書きを防ぐ場合は、 ブール値を AllowSafetyRuleOverrides に設定した次のようなポリシーをアタッチできますfalse。オペレータ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 に設定しますtrue。

```
{
    "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、多くの一般的なユースケースに対するアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールへのアクセス許可の割り当てを開始できます。

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 管理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) が更新されます。 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 管理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: 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 でリカバリコントロール設定を操作するための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 で復旧コントロール設定を操作するための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 でクラスターデータプレーンを操作するための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。

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アラートについては、ARC ドキュメント履歴ページの RSS フィードに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 (以前は制限と呼ばれていました) 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 での準備状況チェック

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、イベントから復旧することを選択できます。準備状況チェックは、

マルチリージョンアプリケーションをフェイルオーバートラフィックを処理するようにスケーリングおよび設定された状態で継続的に維持できるようにするのに役立ちます。

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、ARC がアプリケーションの準備状況を監査できるように、準備状況チェックと準備状況スコープを追加するステップに従います。

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。準備状況チェックは、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、その他の設定の詳細を反映して、本番稼働用レプリカと継続的に一致することを確認するのに役立ちます。レプリカが一致しない場合は、アプリケーションレプリカが再度整列するように容量を追加したり、設定を変更したりできます。

> **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) の準備状況チェックとは

ARC の準備状況チェックでは、 AWS プロビジョニングされた容量、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。

クォータの場合、ARC が準備状況チェックとの不一致を検出すると、低いクォータを高いクォータに合わせて引き上げることで、レプリカのクォータを調整する手順を実行できます。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準備状況ルールのセットがあり、ARC チェックはリソースの準備状況を監査するために使用します。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 を使用する方法、 を使用する方法、ARC API アクションを使用する方法のいずれかで AWS Management Console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。

準備状況チェックによるアプリケーションレプリカのモニタリング

ARC は、準備状況チェックを使用してアプリケーションレプリカを監査し、各レプリカの設定とランタイム状態が同じであることを確認します。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整っていることを確認するのに役立ちます。準備状況チェックは、復旧環境がスケーリングされ、必要に応じて にフェイルオーバーするように設定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

復旧に備えるには、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するために、レプリカに常に十分な予備容量を維持する必要があります。ARC はアプリケーションを継続的に (1 分に 1 回) 検査し、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で一致することを確認します。

ARC が検査する容量には、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あります。プライマリレプリカの容量をリソース値に合わせてスケールアップしても、スタンバイレプリカの対応する値も増やすのを忘れた場合、ARC は不一致を検出して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 準備状況チェックは可用性が高くないため、停止中にアクセス可能なチェックに依存しないでください。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 で ARC の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 Management Console、 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レプリカを監査し、各レプリカの設定設定とランタイム状態が同じであることを確認します。

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 分に 1 回) 検査し、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で一致することを確認します。ARC が検査する容量には、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あります。プライマリレプリカの容量をリソース値に合

わせてスケールアップしても、スタンバイレプリカの対応する値も増やすのを忘れた場合、ARC は不一致を検出して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 準備状況チェックは可用性が高くないため、停止中にアクセス可能なチェックに依存しないでください。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 で ARC の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 Management Console、 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

ARC 準備状況チェックは、各リソースタイプの事前定義されたルールと、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を決定します。ARC には、サポートす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力

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 Management Console 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 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 (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。

さらに、ARC はクォータに追加のステップを実行します。準備状況チェックで、サポートされているリソースのサービスクォータ (リソースの作成とオペレーションの最大値) のセル間で不一致が検出された場合、ARC はクォータを低くしてリソースのクォータを自動的に引き上げます。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。

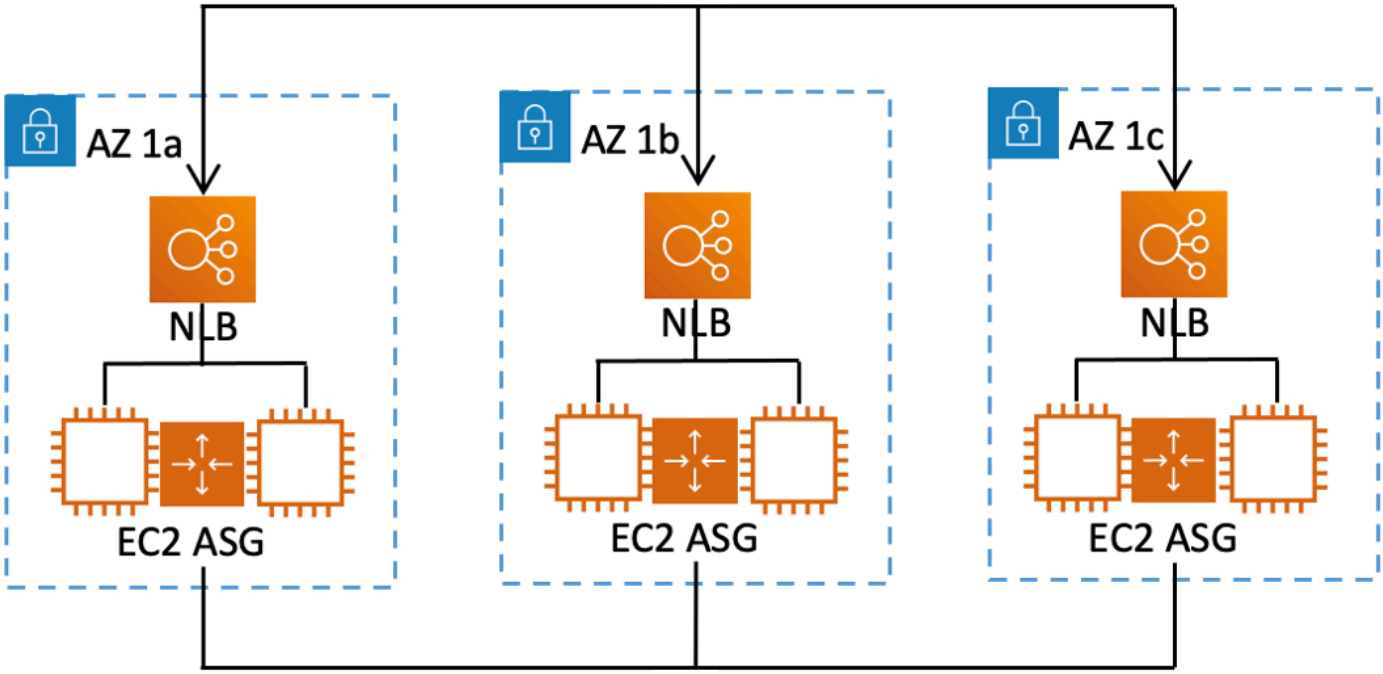
また、Amazon EventBridge で準備状況チェック用の通知を設定し、チェックのステータスが NOT READY に変わったときに通知を受け取ることもできます。設定の不一致が検出されると EventBridge から通知が送信されるので、修正措置を講じて、アプリケーションのレプリカを一致させ、リカバリに向けて準備を整えることができます。詳細については、「Amazon EventBridge で ARC の準備状況チェックを使用する」を参照してください。

準備状況チェック、リソースセット、準備状況スコープがどのように連携するか

準備状況チェックは、常にリソースセット内のリソースのグループを監査します。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するリソースセットを (個別に、または準備状況チェックの作成中に) 作成して、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。準備状況スコープを使用すると、ARC はリカバリグループの概要NOT READY準備状況チェックとセルレベルの概要準備状況チェックの正しいREADYステータスまたは準備状況ステータスを表示できます。準備状況の範囲を設定しない限り、ARC はこれらの概要を提供できません。

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

ARC での DNS ターゲットリソース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回復力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。

復旧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障害が発生する可能性があります。アプリケーションが正しくサイロ化されるように調整する必要がある場合、ARC は、回復力がありフェイルオーバーの準備が整っていることを確認するために、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。

ARC は、アプリケーション内のセルの数と範囲 (レプリカまたは障害封じ込めユニットを表す) 、およびセルがアベイラビリティーゾーンまたはリージョンによって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情報を提供し、それらが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確認します。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 でのアーキテクチャの推奨事項の取得」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

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を処理するようにアプリケーションがスケーリングされていることを確認することで、アプリケーションとリソースを復旧する準備ができているかどうかに関するインサイトが得られます。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、米国西部 (オレゴン) の 1 つの AWS リージョンでホストされており、停止または災害時に準備状況チェック情報が古くなったり、チェックが利用できなくなったりする可能性があることに注意してください。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。

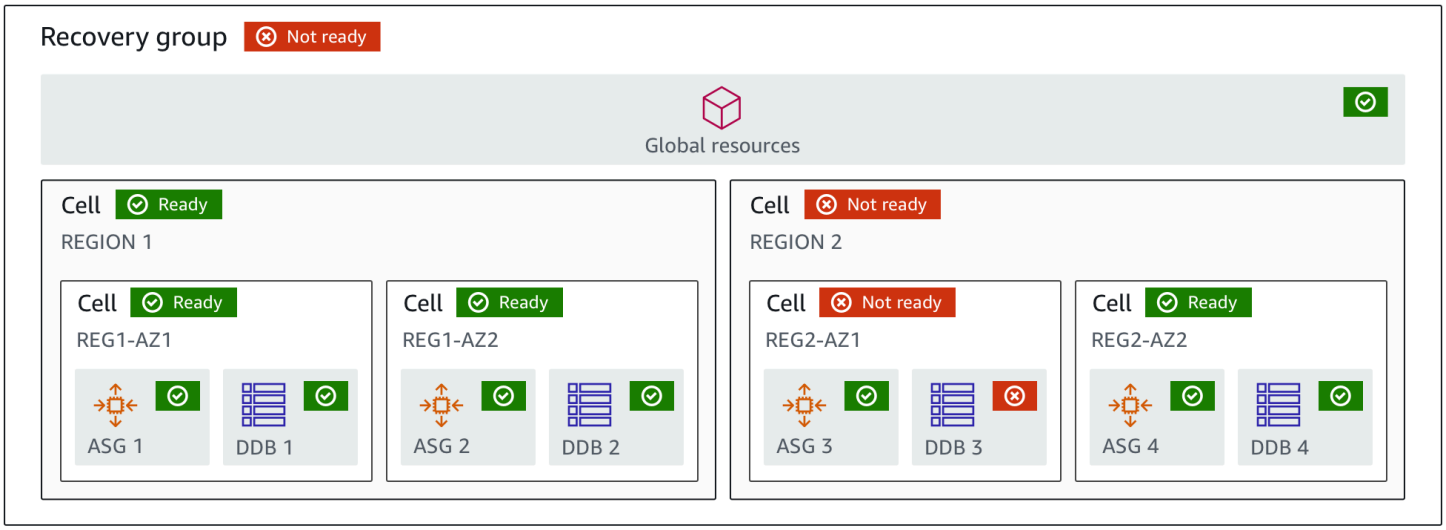
AWS 準備状況チェックのリージョンの可用性

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の詳細については、Amazon Web Services 全般のリファレンスの「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

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のセル (別 AWS 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別) にグループ化されます。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 の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個別に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 間でレプリケートされる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、このアプリケーションを ARC で、us-east-1a と us-east-1b の 2 つのセルを持つリカバリグループとして表すことができます。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用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、リソースの作成時に ARC が生成する識別子です。

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 リソースを提供すると、アプリケーションの回復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に関する推奨事項を取得できます。DNS ターゲットリソースのリソースセットを ARC で作成し、リソースセットの準備状況チェックを作成して、アプリケーションのアーキテクチャに関する推奨事項を取得できます。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、リソースセット内の一連のリソースに対して ARC が実行する監査です。ARC には、準備状況チェックをサポートするリソースのタイプごとに一連の準備状況ルールがあります。各ルールには、ARC がリソースを検査する対象を説明する 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 が復旧準備状況を監査している一連の Amazon Aurora インスタンスなど、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を含めることができます。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リングされていることを確認するために、ARC は 2 つのグループのインスタンスタイプとカウントを継続的にモニタリングします。

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であるリソースの場合、準備範囲を に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ます。例えば、Route 53 ヘルスチェックはリージョンまたはアベイラビリティーゾーンに固有ではないため、ARC のグローバルリソースです。

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

フェイルオーバーとディザスタリカバリを計画する際には、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。通常、信頼性と耐障害性を最大限に高めるために、可能な限りメカニズムにデータプレーン関数を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。これらはいずれも信頼性が高いように構築されていますが、データ整合性のためにコントロールプレーンが最適化され、可用性のためにデータプレーンが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 1 つの API、Recovery Readiness 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。準備状況チェックコントロールプレーンとデータプレーンは信頼性はありますが、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に 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) の準備状況チェックのタグ付け

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。各リソースには複数のタグを追加でき、各タグにはユーザーが定義したキーと値が含まれています。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、 を使用するなど、 API を介してのみ使用できます AWS CLI。

以下は、 を使用して準備状況チェックでタグ付けする例です AWS CLI。

```
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) のリカバリ準備 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

設定した準備状況チェックごとに時間単位のコストを支払います。

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

アプリケーションの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、回復力を高めるためにアプリケーションをセットアップするためのガイドラインがあります。次に、アプリケーションの準備状況チェックを作成し、フェイルオーバーのためにトラフィックを再ルーティングするためのルーティングコントロールを設定できます。また、ARC がアプリケーションのアーキテクチャに関して提供する推奨事項を確認して、回復性を向上させることもできます。

> **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。ゾーンシフトまたはゾーンオートシフトを使用して、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復旧するためのセットアップは必要ありません。
>
> 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 でのゾーンシフト」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、「」を参照してくださいAmazon Application Recovery Controller (ARC) でのマルチリージョンリカバリの開始方法。

ARC の準備状況チェックのベストプラクティス

Amazon Application Recovery Controller (ARC) の準備状況チェックでは、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の通知を追加する

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 で ARC の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

次の表に、復旧準備 (準備状況チェック) に使用できる ARC オペレーションと、関連するドキュメントへのリンクを示します。

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
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

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 で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 で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 で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 で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 で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 で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 で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 で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 で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 での準備状況チェックの作成と更新」を参照	「ListReadinessChecks」を参照
準備状況ルールを一覧表示する	「ARC の準備状況ルールの説明」を参照	「ListRules」を参照

アクション	ARC コンソールの使用	ARC API の使用
準備状況チェック全体の状態をチェックする	「ARC の準備状況ステータスのモニタリング」を参照	「GetReadinessCheckStatus」を参照
リソースの状態をチェックする	「ARC の準備状況ステータスのモニタリング」を参照	「GetReadinessCheckResourceStatus」を参照
セルの状態をチェックする	「ARC の準備状況ステータスのモニタリング」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 の準備状況ステータスのモニタリング」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

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。例は、 CLI を使用して準備状況チェック機能を使用する方法の基本的な理解を深めやすくす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 セルで設定またはモデル化する必要があります。次に、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て、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、復旧グループに収集されます。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、リカバリグループや準備状況チェックなどのリソースを作成するには。

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 でのリカバリ準備状況に関する最上位のリソースです。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はリソースセットと呼ばれます。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 のすべての準備状況ルールのリストを表示するには、 を使用するlist-rulesか、「」を参照してくださいARC の準備状況ルールの説明。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ます。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

このセクションでは、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。これには、これらのリソースの作成、更新、削除が含まれます。

ARC でのリカバリグループの作成、更新、削除

リカバリグループは、Amazon Application Recovery Controller (ARC) 内の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レプリカ間で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のに役立ちます。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。レプリカを使用して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、レプリカの設定と設定が時間の経過とともに同じであることを確認するのに役立ちます。

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

このセクションのステップでは、ARC コンソールでリカバリグループを作成する方法について説明します。Amazon Application Recovery Controller (ARC) でのリカバリ準備 API オペレーションの使用については、「」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。

リカバリグループを作成するには

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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

このセクションのステップでは、ARC コンソールでリカバリグループを更新および削除し、セルを削除する方法について説明します。Amazon Application Recovery Controller (ARC) でのリカバリ準備 API オペレーションの使用については、「」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。

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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 での準備状況チェックの作成と更新

このセクションでは、これらのリソースの作成、更新、削除など、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

このセクションのステップでは、ARC コンソールで準備状況チェックを作成する方法について説明します。Amazon Application Recovery Controller (ARC) でのリカバリ準備 API オペレーションの使用については、「」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。

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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 形式の例と詳細については、「」を参照してくださいARC のリソースタイプと ARN 形式。

7. 必要に応じて、ARC がこの準備状況チェックに含めたリソースのタイプをチェックするときに使用する準備状況ルールを表示します。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。このステップをスキップし、ここで準備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い場合、ARC はリカバリグループまたはセルの概要準備情報を返すことができません。

9. [Next (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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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。このセクションのステップでは、ARC コンソールでリソースセットを作成または編集する方法について説明します。Amazon Application Recovery Controller (ARC) でのリカバリ準備 API オペレーションの使用については、「」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。

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. [Create] (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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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、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. [Save] を選択します。

ARC の準備状況ステータスのモニタリング

Amazon Application Recovery Controller (ARC) でアプリケーションの準備状況は、次の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更について通知を受け取ることも、Route 53 コンソールまたは ARC CLI コマンドを使用して準備状況ステータスの変更をモニタリングすることもできます。

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、準備状況ステータスの変更を通知するイベント駆動型ルールを設定できます。詳細については、「Amazon EventBridge で ARC の準備状況チェックを使用する」を参照してください。

ARC コンソールでの準備状況ステータスのモニタリング

次の手順では、 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 AWS Management Console。

1. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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。以下に例を示します。

```
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 でのアーキテクチャの推奨事項の取得

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、アプリケーションの回復回復性を向上させるために設計を変更するための推奨事項を提供できます。アプリケーションを表すリカバリグループを ARC で作成したら、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。追加情報を入力すると、ARC はより良いレコメンデーションを提供できます。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した場合、ARC はリカバリグループに最適なセル数を作成したかどうかに関する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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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、ARC がアーキテクチャの推奨事項を提供できるように作成します。[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 形式」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 は、ページに一連の推奨事項を表示します。

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可の作成

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ヘルス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の取得が困難になる可能性もあります。Amazon Application Recovery Controller (ARC) の準備状況チェックを効率化するために、クロスアカウント認可を使用できます。

ARC でのクロスアカウント認可は、準備状況チェック機能で機能します。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可を設定して操作します。このセクションでは、使用開始に役立つ 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 では、クロスアカウント認可を使用して、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可を付与したすべてのアカウントについて特定のアカウントを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

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、およびリソースセットに使用する Amazon リソースネーム (ARNs) 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 の準備状況ルールの説明

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タイプのリソースの準備状況ルールについて説明します。ARC でサポートされているリソースタイプのリストを確認するには、「」を参照してくださいARC のリソースタイプと ARN 形式。

準備状況ルールの説明は、ARC コンソールで表示することも、API オペレーションを使用して表示することもできます。

- 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 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 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準備状況ルールのセットを一覧表示します。

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。これらの用語が準備状況チェックのコンテキストでルールがどのように機能するか、および 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決定する方法」を参照してください。

準備状況ルール

ARC は、次の準備状況ルールを使用してリソースを監査します。

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。

AWS 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。

AWS 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 Management Console、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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
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 形式

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 (ARNs) を指定します。ARC では、リソースタイプごとに特定の ARN 形式を想定しています。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、それぞれの関連する ARN 形式を一覧表示します。

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 がリソースに必要とする ARN 形式は、サービス自体がリソースに必要とする ARN 形式とは異なる場合があることに注意してください。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 が ARC サービスの機能をサポートするために必要とする 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、「サービス認可リファレンス」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

Amazon CloudWatch と 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**
>
> ARC の CloudWatch メトリクスとログは、 コンソールと の使用時の両方で、米国西部 (オレゴン) リージョンで表示する必要があります AWS CLI。を使用する場合は AWS CLI、次のパラメータ 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 で ARC の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

Amazon Application Recovery Controller (ARC) は、準備状況チェックのために 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 で 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ます。これは、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。 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、ARC によって評価される特定のタイプあたりの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 は、ARC によって発行されたメトリクスデータポイントに基づく統計を提供します。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 によって準備状況について評価された準備状況チェックの数を表示します。
- ARC によって評価される特定のリソースセットタイプのリソースの合計数を表示します。

ARC で CloudWatch メトリクスを表示する

CloudWatch コンソールまたは を使用して、ARC の CloudWatch メトリクスを表示できます AWS CLI。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。

ARC の CloudWatch メトリクスは、米国西部 (オレゴン) リージョンで、コンソールまたは の使用時に表示する必要があります AWS CLI。を使用する場合は AWS CLI、次のパラメータ 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)] を選択します。
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 の例では、ARC のアカウントについて 1 分あたりに評価される準備状況チェックの合計を一覧表示します。

```
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

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。キャプチャされた呼び出しには、ARC コンソールからの呼び出しと ARC API オペレーションへのコード呼び出しが含まれます。

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 イベントの継続的な配信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 によって収集された情報を使用して、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、リクエスト者、リクエスト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 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。追跡により、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 によってログに記録され、Amazon Application Recovery Controller のリカバリ準備 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、および 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、`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 root 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) を選択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、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 で ARC の準備状況チェックを使用する

Amazon EventBridge を使用すると、Amazon Application Recovery Controller (ARC) で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。たとえば、準備状況チェックのステータスが READY から NOT READY に変わったときに Amazon SNS トピックにシグナルを送信することで、E メール通知を送信するルールを設定できます。

> **Note**
>
> ARC は、米国西部 (オレゴン) (us-west-2) AWS リージョンで準備状況チェックのために EventBridge イベントのみを発行します。準備状況チェックのために EventBridge イベントを受信するには、米国西部 (オレゴン) リージョンで 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を実行できます。

- 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、EventBridge がイベントを検出するために使用できるイベント固有のパターンを定義します。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発生します。これらは、通常の運用状況下でほぼリアルタイムで ARC から EventBridge に配信されます。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 を使用すると、ARC が準備状況チェックリソースのイベントを発行するときに実行するアクションを定義するルールを作成できます。

EventBridge コンソールでイベントパターンを入力またはコピーして貼り付けるには、 オプション自分のオプションを入力 を選択します。このトピックでは、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、準備状況イベントパターンの例を示します。

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. 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. [Next (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してください。

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 に貼り付けると、ARC アクションとリソースのモニタリングに使用できるルールを作成できます。

次のイベントパターンは、ARC の準備状況チェック機能に EventBridge で使用できる例を示しています。

- ARC 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変更の 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変更の 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、準備状況チェックのステータス変更の 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が送信されるターゲットを指定する必要があります。EventBridge で使用可能なターゲットのリストについては、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、ルールの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と、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 で始まることを確認します/aws/events。既存のロググループを選択する場合は、 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/aws/events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の作成」を参照してください。

コンソールの外部で 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使用して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。

```
{
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

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、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、イベントパターンや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。

   EventBridge の準備状況ルールの作成の詳細については、EventBridge での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. ターゲットの選択ページで、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

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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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) の準備状況チェックが IAM と連携する方法
- ARC での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) の準備状況チェックが IAM と連携する方法

IAM を使用して ARC へのアクセスを管理する前に、ARC で使用できる IAM 機能を確認してください。

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) の準備状況チェックで使用できる IAM 機能

IAM の機能	準備状況チェックのサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	いいえ
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	Yes
ACL	いいえ
ABAC (ポリシー内のタグ)	あり

IAM の機能	準備状況チェックのサポート
一時的な認証情報	はい
プリンシパル権限	はい
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 機能とどのように連携するか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 ユーザーグループ、ユーザーのグループ、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。ID ベースの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。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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 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、「」を参照してくださいAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例。

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。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

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。

準備状況チェック用の ARC アクションのリストを確認するには、「サービス認可リファレンス」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。

準備状況チェック用の ARC のポリシーアクションは、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。たとえば、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*"
```

準備状況チェック用の ARC アイデンティティベースのポリシーの例については、「」を参照してくださいARC での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。

準備状況チェック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。ステートメントにはResource または NotResource 要素を含める必要があります。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。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限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。

準備状況チェック用の ARC アイデンティティベースのポリシーの例については、「」を参照してくださいARC での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。

準備状況チェック用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 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。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。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。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。

準備状況チェック用の 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。

準備状況チェックで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。

準備状況チェック用の ARC アイデンティティベースのポリシーの例については、「」を参照してくださいARC での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。

準備状況チェックのアクセスコントロールリスト (ACLs)

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コントロール (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制御 (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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。

Recovery Readiness (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

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報と AWS のサービス 連携する などの詳細については、AWS のサービス IAM ユーザーガイドの「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の方法 AWS Management Console を使用して にサインインする場合、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。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 の一時的セキュリティ認証情報」を参照してください。

準備状況チェックのためのクロスサービスプリンシパルのアクセス許可

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ため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、「」を参照してください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。

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、ユーザーとロールには ARC リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの 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。

各リソースタイプの ARNs「サービス認可リファレンス」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、アカウント内で誰かが ARC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。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 – 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。例えば、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 AWS 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 Access Analyzer 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 Access Analyzer 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のセットが必要です。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 準備状況チェック 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」を参照してください。

一部のタスクを実行するには、ARC の準備状況チェックに関連付けられた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がユーザーに必要です。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。

```
{
    "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、たとえば、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、ARC の準備状況チェックに関連付けられた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がユーザーに必要です。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。

準備状況チェックのために 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。

```
{
    "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 AWS 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 Management Console 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 で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可の作成」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提供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) が ARC によって使用または管理されるサービスとリソースにアクセス AWS 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。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 で復旧準備状況 (準備状況チェック) を操作するための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 でリカバリの準備状況を操作するための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アラートについては、ARC ドキュメント履歴ページの RSS フィードに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

Amazon Application Recovery Controller (ARC) の準備状況チェックには、次のクォータ (以前は制限と呼ばれていました) 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 でのリージョンスイッチ

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。リージョンスイッチは、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引き続き動作し、正常な で実行されます AWS リージョン。

リージョンスイッチは、特定の復旧ニーズに合わせて設計および設定するプランの概念を中心に構築されています。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。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、リソースの切り替えやアプリケーションのトラフィックリダイレクトの管理など、異なる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、ネストされたプランを作成できます。

リージョンスイッチには以下が含まれます。

- アクティブ/パッシブおよびアクティブ/アクティブ設定のサポート。アクティブ/パッシブの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して返すことができます。
- 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計画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することもできます。
- 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。

リージョンスイッチは によって完全に管理されます AWS。リージョンスイッチを使用すると、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。

リージョンスイッチについて

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。

リージョンスイッチは、特定の復旧ニーズに合わせて設計および設定するプランの概念を中心に構築されています。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。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、リソースの切り替えやアプリケーションのトラフィックリダイレクトの管理など、異なるタスクを処理します。子プランを追加することで、ネストされたプランをより柔軟に作成できます。

計画を作成または更新するたびに、リージョンスイッチは計画評価を実行し、IAM アクセス許可、リソース設定、または実行中の容量に問題がないことを確認します。リージョンスイッチは、これらの評価を定期的に実行し、検出された問題について警告を生成します。

リージョンスイッチは、プランの実行ごとに実際の復旧時間値を計算し、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。のリージョン切り替えダッシュボードで、復旧時間やその他の計画実行に関する詳細を表示できます AWS Management Console。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。

リージョンスイッチ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

リージョンスイッチプランは、リージョンスイッチの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 をアクティブ化または非アクティブ化できます 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計画では、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計画では、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、他のリージョンから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提供します。リージョンスイッチを使用する場合は、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。リージョンスイッチ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。

たとえば、us-east-1 をプライマリリージョンとして、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場合があります。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、リージョンスイッチプランを実行して 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) に制限されていますが、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

リージョンスイッチ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追加して、リソースのスケールアップやルーティングコントロールの更新によるトラフィックの再ルーティングなど、特定のタスクを完了します。実行ブロックを使用すると、これらのタスクとその完了順序を指定できます。ネストされた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。

ワークフローに実行ブロックを順次追加することも、1 つ以上の実行ブロックを並行して追加することもできます。また、リソースに応じて、正常な (計画的な) 実行または不正な (計画外の) 実行で実行ブロックを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、適切なワークフローを使用して、秩序ある計画を実行するためのすべてのステップを実行できます。
- 不正な実行: 予期しない実行。不正なワークフローモードでは、必要なステップとアクションのみが使用されます。このモードは、ワークフロー内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。

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、「」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リージョン切り替えでのクロスアカウントサ

ポート。必要な IAM ロールを設定したら、計画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、実行ブロックを追加するときに、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、「」を参照してくださいクロスアカウントリソースアクセス。

ワークフローの詳細については、「」を参照してくださいリージョン切り替え計画ワークフローを作成する。設定ステップ、仕組み、計画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、「」を参照してください実行ブロックを追加する。

評価を計画する

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スイッチが実行され、その後は定常状態で 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、計画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、および実行中の容量の検証が含まれます。

リージョンスイッチは、計画の実行の成功を妨げる可能性のある問題を検出した場合、コンソールの計画の詳細ページで強調表示されている計画評価警告を生成します。Amazon EventBridge でプラン評価警告を使用するか、リージョンスイッチ API を使用して警告を表示することもできます。

評価対象の計画に関する問題の詳細と推奨される修復については、計画の詳細ページの計画評価タブで確認できます。また、リージョンスイッチ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もお勧めします。また、復旧プランが期待どおりに機能することをテストするために、リージョンスイッチプランの評価だけに依存しないことをお勧めします。

リージョンアラームと実際の復旧時間

リージョンスイッチは、計画実行ごとに実際の復旧時間値を計算し、計画実行後に表示できます。実際の復旧時間は計画実行の詳細ページに表示されるため、実際の時間を計画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合計時間と、設定した特定の Amazon CloudWatch アラームがグリーン状態に戻るまでに が経過する追加時間に基づいて計算されます。

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、リージョンの Amazon CloudWatch アラームをリージョンスイッチプランに追加し、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します。プランが実行されると、リージョンスイッチ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再び正常かを判断します。次に、リージョンスイッチ

は、指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。

AWS リージョン

リージョンスイッチは、すべての商用 で使用できます AWS リージョン。

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の詳細については、Amazon Web Services 全般のリファレンスの「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。

リージョンスイッチコンポーネント

以下は、Amazon Application Recovery Controller (ARC) のリージョンスイッチ機能のコンポーネントと概念です。

計画

計画は、アプリケーションの基本的な復旧プロセスです。計画を作成するには、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にシフトして、アプリケーションの復旧を完了するための計画を実行します。

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、並列または順番に実行するように指定する実行ブロックで構成され、復旧計画の一環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します。アクティブ/パッシブアプローチを使用するように設定する計画では、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計画では、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

リージョンスイッチプランワークフローにリージョンスイッチ実行ブロックを追加します。実行ブロックを使用すると、アクティブ化してい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。ワークフローに実行ブロックを追加する場合、他のブロックと順番に追加するか、1 つ以上の他のブロックと並行して追加できます。

グレースフルおよび非グレースフルな設定

特定の実行ブロックを、正常な (計画的) 実行または不正な (計画的でない) 実行で実行することを選択できます。環境が正常であれば、適切なワークフローを使用して、秩序ある計画を実行するためのすべてのステップを実行できます。不正なワークフローモードでは、必要なステップとアクションのみが使用されます。不正なモードで計画を実行すると、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動作が異なります。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を含む 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。リージョンスイッチは、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。

計画実行

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で、リージョンに障害が発生したときにアプリケーションの復旧を実装します。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。

アプリケーションのヘルスアラーム

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プランに指定する CloudWatch アラームです。リージョン切り替えでは、アプリケーションのヘルスアラームを使用して、リージョンを切り替えて復旧を実装した後の実際の復旧時間を決定します。

トリガー

リージョンスイッチ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、アプリケーションの正常性を示す 1 つ以上の Amazon CloudWatch ア

ラームを指定します。アラームがアラーム状態になると、リージョンスイッチは対応する復旧計画を自動的に実行します。

ダッシュボード

リージョンスイッチには、計画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含まれています。

リージョンスイッチ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。通常、最大限の信頼性と耐障害性を実現するために、可能な限りメカニズムにデータプレーン関数を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性が高い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。

リージョンスイッチの場合、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割されます。

- 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) 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。リージョンスイッチ設定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。
- リージョンスイッチには、それぞれ独立したデータプレーンがあります AWS リージョン。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。データプラ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スイッチデータプレーンオペレーションは高可用性です。

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。リージョンスイッチを使用して復旧オペレーションを準備して完了する際の重要な考慮事項の詳細については、「」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。

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 リージョンスイッチのタグ付け

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。各リソースには複数のタグを追加でき、各タグにはユーザーが定義したキーと値が含まれています。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 のリージョンスイッチで次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、 を使用するなど、 API を介してのみ使用できます AWS CLI。

以下は、 を使用したリージョンスイッチでのタグ付けの例です AWS CLI。

```
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) のリージョンスイッチ API リファレンスガイド」のTagResource」を参照してください。

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 の料金と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) のリージョンスイッチによる復旧とフェイルオーバーの準備については、次のベストプラクティスをお勧めします。

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容量を予約する
- 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 による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。

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容量を予約する

リージョンスイッチには、リカバリ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。これらの実行ブロックをプランで使用すると、リージョンスイッチは で必要なコンピューティング容量が達成されることを保証しません。

重要なアプリケーションがあり、容量へのアクセスを保証する必要がある場合は、容量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ーを予約しながら、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、「」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。

各リージョンのリージョンスイッチコンソールは、リージョンスイッチ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を使用します。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 は、データプレーンの API で非常に高い信頼性を提供します。

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プランがスタック内の正しいリソースと一致していること、およびすべてが期待どおりに機能することを確認することが重要です。環境のリージョンスイッチをセットアップした後、これをテストし、復旧プロセスが正しく動作することを検証するために、定期的にテストを続ける必要があります。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。

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スイッチプランを作成し、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を作成する
- 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、両方のリージョンに次の前提条件があ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. リージョンスイッチコンソールから、リージョンスイッチプランの作成を選択します。
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 を選択する
   - スタンバイリージョン: us-west-2 を選択する
   - 目標復旧時間 (RTO) (オプション)
   - IAM ロール: 計画実行 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を作成します。
   - 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ます。
2. プランで リソースを設定します。
   - リソースをプランに追加するときは、次の 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。

計画を実行する前に、リージョンスイッチ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受けることができます。

- 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. リージョンスイッチプランの詳細ページで、ワークフローの構築を選択します。
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、計画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. [Save and continue] を選択します。
5. ステップを追加を選択し、順番に実行を選択します。
6. 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容量の増加を開始できます。
7. 右側のパネルで、 ブロックを設定します。
   - ステップ名 :「スケール」と入力します。
   - 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容量と一致する割合: 100 を入力します
   - キャパシティモニタリングアプローチ:「最新」のままにする
   - タイムアウト (オプション)
8. 保存ステップを選択します。
9. ステップの追加 を選択します。
10. 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。
11. 右側のパネルで、 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」と入力します。
    - 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受ける必要があるロール
    - 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。

12. 保存ステップを選択します。
13. ステップの追加 を選択します。
14. カスタムアクション Lambda 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。
15. 右側のパネルで、 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」と入力します。
    - 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: リージョンのアクティブ化で実行を選択します
    - タイムアウト (オプション)
    - 再試行間隔 (オプション)
16. 保存ステップを選択します。
17. ステップの追加 を選択します。
18. 2 番目のカスタムアクション Lambda 実行ブロックを選択し、追加と編集を選択します。このブロックは、アクティブなリージョンでフェンシングメカニズムをトリガーし、非アクティブ化されたリージョンがトラフィックを受け入れないようにします。
19. 右側のパネルで、 ブロックを設定します。

    - ステップ名 :「フェンシング」と入力します。
    - 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 にデプロイされたフェンシング 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 にデプロイされたフェンシング Lambda 関数の ARN
    - Lambda 関数を実行するリージョン: リージョンの非アクティブ化で実行を選択します
    - タイムアウト (オプション)
    - 再試行間隔 (オプション)
20. 保存ステップを選択します。
21. ステップの追加を選択します。

22. 手動承認実行ブロックを選択し、追加と編集を選択します。このブロックは、チームメンバーに承認をリクエストします。
23. 右側のパネルで、 ブロックを設定します。
    - ステップ名: データベースと DNS の変更前に手動承認を入力する
    - ステップの説明 (オプション)
    - IAM 承認ロール: ユーザーが実行を承認できるように引き受ける必要があるロール
    - タイムアウト (オプション)
24. 保存ステップを選択します。
25. ステップの追加 を選択します。
26. Aurora グローバルデータベース実行ブロックを選択し、追加と編集を選択します。このブロックは、Aurora グローバルデータベースのスイッチオーバーをトリガーします (データ損失なし)。詳細については、Aurora ユーザーガイドの「Aurora Global Database 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。
27. 右側のパネルで、 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する
    - ステップの説明 (オプション)
    - Aurora グローバルデータベース識別子: 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を選択します。スイッチオーバーを選択します。
    - タイムアウト (オプション)
28. 保存ステップを選択します。
29. ステップの追加 を選択します。
30. ARC ルーティングコントロール実行ブロックを選択し、追加と編集を選択します。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。
31. 右側のパネルで、 ブロックを設定します。
    - ステップ名: DNS の切り替えと入力
    - 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する
- タイムアウト: タイムアウト値を入力します。

32. ルーティングコントロールの追加を選択します。
    - 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
33. ルーティングコントロールを再度追加するを選択します。
    - 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: オフを選択
34. [保存] を選択します。
35. 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の追加を選択する
36. ルーティングコントロールの追加を選択します。
    - 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
37. ルーティングコントロールを再度追加するを選択します。
    - 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: オフを選択
38. [保存] を選択します。
39. 保存ステップを選択します。
40. [保存] を選択します。

ステップ 3: 計画を実行する

1. リージョンスイッチプランの詳細ページの右上で、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。
    - 計画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。

- 計画の実行を承認します。

3. [開始] を選択します。
4. プランの実行時に詳細なステップを実行の詳細ページで表示できます。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。

障害のあるリージョンが回復したら、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。

リージョン切り替え API オペレーション

次の表に、リージョンの切り替えに使用できる ARC オペレーションと、関連するドキュメントへのリンクを示します。

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
計画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を作成する」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を作成する」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	「リージョンスイッチの使用」を参照してください。	「DeletePlan」を参照してください。	いいえ

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
プランを取得する	「リージョンスイッチの使用」を参照してください。	「GetPlan」を参照してください。	いいえ
計画評価ステータスの取得	「評価を計画する」を参照してください。	「GetPlanEvaluationStatus」を参照してください。	はい
プランの実行を取得する	「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	「リージョンスイッチの使用」を参照してください。	「GetPlanInRegion」を参照してください。	はい
プランの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。	「ListHealthChecksForPlan」を参照してください。	いいえ
計画実行イベントを一覧表示する	「リージョンスイッチ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。	「ListPlanExecutionEvents」を参照してください。	はい
計画実行を一覧表示する	「リージョンスイッチ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

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
プランを一覧表示する	「リージョンスイッチの使用」を参照してください。	「ListPlans」を参照してください。	いいえ
リージョンの計画を一覧表示する	「リージョンスイッチの使用」を参照してください。	「ListPlans InRegion」を参照してください。	はい
リソースのタグの一覧表示	「 ARC リージョンスイッチのタグ付け」を参照してください。	「ListTagsF orResource」を参照してください	いいえ
計画の実行を開始する	「リージョンスイッチ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
リソースにタグを付ける	「リージョン切り替えプランを作成する」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 リージョンスイッチのタグ付け」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を作成する」を参照してください。	「UpdatePlan」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
計画実行を更新する	「リージョン切り替えプランを作成する」を参照してください。	「UpdatePlanExecution」を参照してください。	はい
計画実行ステップを更新する	「リージョン切り替えプランを作成する」を参照してください。	「UpdatePlanExecutionStep」を参照してください。	はい

リージョンスイッチの使用

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step-by-stepの手順について説明します。リージョンスイッチを使用すると、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計画を作成できます。

アプリケーションの復旧計画を作成するには、次の手順を実行します。

1. リージョンスイッチプランを作成します。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。ワークフローを最初に作成しないと、計画を実行できません。
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、実行ブロックを追加して、送信先リージョンの EC2 Auto Scaling グループをスケールアップできます。
4. ワークフローに実行ブロックを追加した後、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で、計画実行の進行状況を追跡できます。

以下のセクションでは、計画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を作成する
- リージョン切り替え計画ワークフローを作成する
- 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
- リージョンスイッチ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は、 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を示しています AWS Management Console。代わりにリージョンスイッチ API オペレーションを使用するには、「」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。

リージョン切り替えプランを作成する

リージョンスイッチでは、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で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ているか、トラフィックを受信しています。

リージョン切り替えプランを作成するには

1. リージョンスイッチコンソールから、アクティブ/パッシブアプローチでリージョンスイッチプランを作成するを選択します。
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でレプリカをアクティブ化できます。
     - 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、トラフィックがアクティブなリージョンにのみルーティングされている場合は、アクティブ/パッシブを選択します。次に、アクティブ/パッシブを指定するリージョン切り替えプランを実行して、パッシブリージョンでレプリカをアクティブ化できます。
     - 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ているか、トラフィックを受信している場合は、アクティブ/アクティブを選択します。
   - プライマリリージョンとスタンバイリージョン - 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。リージョンスイッチはこれを使用して、リージョンスイッチ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 でのリージョンスイッチの 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。リージョンスイッチは、これらのアプリケーションヘルスアラームを使用して、リージョンを切り替えて復旧を実装した後の実際の復旧時間を決定します。
   - 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。

リージョン切り替え計画ワークフローを作成する

リージョンスイッチ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。各ワークフローで、アプリケーション復旧のためにリージョンスイッチで実行する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復旧を管理するための設定によって異なります。例:

- リージョンスイッチ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、 またはアクティブ/アクティブデプロイの場合、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 2 つ以上のワークフローがあります。
- リージョンスイッチ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。

リージョン切り替え計画ワークフローを作成するには

1. 作成したリージョンスイッチプランで、ワークフローの構築を選択します。
2. 次のいずれかのワークフローオプションを選択します。
   - 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 - 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スイッチで実行するステップを定義します。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、別のリージョンのリソースをサポートします AWS アカウント。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、「」を参照してください実行ブロックを追加する。
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は、1 つのアクティベーションワークフローが必要です。
- リージョンごとにビルドワークフローを個別に選択した場合は、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実行ブロックを追加して、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。

リージョンスイッチは、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、定常状態では 30 分ごとにプラン評価を実行します。リージョンスイッチは、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、リージョンスイッチが計画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。

リージョンスイッチには、リカバリ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。これらの実行ブロックをプランで使用する場合は、リージョンの切り替えによって、 で必要なコンピューティング容量が達成されることが保証されないことに注意してください。重要なアプリケーションがあり、容量へのアクセスを保証する必要がある場合は、容量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ーを予約しながら、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。

リージョンスイッチは、次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 リージョンスイッチプラン実行ブロック	実行する子プランを指定して、複数のアプリケーションの復旧を 1 回の実行でオーケストレーションします。	不適切な設定で子プランを開始します。

実行ブロック	関数	不正な設定
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容量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	プランの実行の一環として 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	プランの実行の一環として Amazon ECS サービスタスクをスケールします。	該当なし
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にリダイレクトします AWS リージョン。	該当なし
Amazon Aurora グローバルデータベース実行ブロック	Aurora グローバルデータベースの復旧ワークフローを実行します。	Aurora グローバルデータベース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)。
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リージョンを指定します。	該当なし

ARC リージョンスイッチ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計画実行ブロックを使用する場合は、作成する計画のワークフローで実行する特定のリージョン切り替え計画を選択します。

リージョンスイッチプラン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. リージョン切り替え計画: 現在の計画のワークフローで実行する計画を選択します。

次に、保存ステップを選択します。

仕組み

リージョン切り替え計画実行ブロックを使用して、親子関係を持つネストされた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、ネストされた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。リージョンスイッチブロックが正常に実行され、グレースフルでない実行モードに切り替えた場合、子プランもグレースフルでない実行モードに切り替わります。

計画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) に対する容量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。

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 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 グループで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: ドロップダウンメニューで、EC2 Auto Scaling グループのモニタリングアプローチを選択します。
6. タイムアウト: タイムアウト値を入力します。

次に、保存ステップを選択します。

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、1 つのソース Auto Scaling グループと 1 つの宛先 Auto Scaling グループのみが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。ターゲット容量は、 状態が 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`InService`。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。リージョンスイッチが希望する容量の計算に使用する式は です。ここで`ceil(percentToMatch * Source Auto Scaling group capacity)`、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。

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、リージョンスイッチは、スケーリングする割合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容量の最小パーセンテージを指定することで、不正な実行を設定できます。

計画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。リージョンスイッチ実行ブロックに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。いずれかのチェックが失敗した場合、リージョンスイッチ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は、非アクティブ化されているリージョンの容量に対するスケーリングする容量の割合を定義します。

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実行ブロックを追加する前に、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、次のリージョン切り替えアクセスポリシーを使用して、リージョン切り替えが計画の実行に使用する 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。 `arn:aws:eks::aws:cluster-access-policy/AmazonARCRegionSwitchScalingPolicy`

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提供します。

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)
*	*/スケール	の取得、更新
*	*/ステータス	get
オートスケーリング	水平ポッドオートスケーラー	取得、パッチ適用

リージョンスイッチの EKS アクセスエントリを作成する

次の例では、リージョンスイッチが 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、アクセス許可は、IAM ロール の EKS クラスター *my-cluster ##### my-namespace1* に適用されます`arn:aws:iam::555555555555:role/`*my-role*。

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 に変更しますAPI。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「アクセスエントリの作成」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「アクセスポリシーとアクセスエントリの関連付け」を参照してください。

実行ブロックの 2 番目の EKS クラスターを他のリージョンで繰り返して、両方のクラスターにリージョンスイッチでアクセスできるようにします。

設定

EKS リソーススケーリング実行ブロックを設定するには、まず正しいアクセス許可が設定されていることを確認します。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。

リージョンスイッチは現在、apps/v1、Deployment、apps/v1 の ReplicaSet リソースをサポートしています。

次に、実行ブロック設定に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: デプロイなど、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容量と一致する割合: 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: ドロップダウンメニューで、EKS リソースのモニタリングアプローチを選択します。
8. タイムアウト: タイムアウト値を入力します。

次に、保存ステップを選択します。

仕組み

プランの実行中、リージョンスイッチは、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、次の式を使用して、送信先リソースに必要なレプリカ数を計算します。 `ceil(percentToMatch * Source replica count)`

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、リージョンスイッチはレプリケート先のリソースレプリカ値を希望する容量にスケーリングします。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容量を増やします。

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、次のパッチ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。 `{"spec":{"behavior":{"scaleDown":{"selectPolicy":"Disabled"}}}}`

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に、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。

計画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。リージョンスイッチは、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。リージョンスイッチは、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、リージョンスイッチには 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、リージョンスイッチは、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、リージョンスイッチは、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、リージョンスイッチ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。リージョンの切り替えがフェイルオーバーまたは非アクティブ化されるリージョンに関連する容量の割合を定義できます。

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。

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: ドロップダウンメニューで、ECS リソースのモニタリングアプローチを選択します。
6. タイムアウト: タイムアウト値を入力します。

次に、保存ステップを選択します。

仕組み

プランで実行ブロックを設定すると、リージョンスイッチは送信元 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、リージョンスイッチは実行ブロックの警告を返します。リージョンスイッチは、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。ターゲット容量は、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、リージョンスイッチは送信先 (アクティブ化) リージョンの ECS サービス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、送信先 ECS サービスの希望する容量と比較されます。リージョンスイッチが希望する容量の計算に使用する式は です。`ceil(percentToMatch * Source Auto Scaling group capacity)`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容量より大きい場合、プランの実行が続行されます。リージョンスイッチは ECS サービス容量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 Application Autoscaling が有効になっている場合、リージョンスイッチは Application Autoscaling の最小容量を更新し、ECS サービスの必要数も更新します。

リージョンスイッチが ECS サービス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 ECS 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 ECS サービスでリクエストされた ECS サービス容量が満たされるまで待機します。必要に応じて、リージョンスイッチ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することで、フルフィルメントが完了する前に完了するようにステップを設定できます。

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、リージョンスイッチは、スケーリングする割合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。

計画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 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。リージョンスイッチは、ECS サービスがソースリージョンとターゲットリージョンの両方に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容量が、ターゲットリージョンの容量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。リージョンスイッチは、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。リージョンスイッチ実行ブロックに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。

さらに、リージョンスイッチは、 `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。

チェックのいずれかが失敗した場合、リージョンスイッチ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実行ブロック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。この実行ブロックを使用すると、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。

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。

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、保存ステップを選択します。

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計画がある場合、状態を On 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 On 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 On 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。

計画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。リージョンスイッチは、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。

リージョンスイッチは、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。リージョンスイッチ実行ブロックに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。

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、リージョンスイッチ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。

Amazon Aurora グローバルデータベース実行ブロック

Amazon Aurora グローバルデータベース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。このアプローチでは、Aurora グローバルデータベース内のセカンダリ DB クラスターのいずれか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ゼロ以外の値です。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「Amazon Aurora グローバルデータベースを予期しない停止から復旧する」を参照してください。
- スイッチオーバー — このオペレーションは、以前はマネージド計画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 グローバルデータベース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。

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 グローバルデータベースクラスター名: 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: 目的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、保存ステップを選択します。

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切り替えることができます。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、リージョンスイッチは、スケーリングする割合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 グローバルデータベース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 グローバルデータベース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 グローバルデータベース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。

計画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 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。リージョンスイッチは、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します。
- 送信元リージョンと送信先リージョンの両方に Aurora DB クラスターがあります。
- 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です。
- 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は互換性があります。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。

リージョンスイッチは、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。リージョンスイッチ実行ブロックに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。

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、リージョンスイッチ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、計画の実行中にこのステップが実行されている間に、計画が Aurora 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。

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、計画の進行を妨げたりすることができます。

計画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に 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。

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、保存ステップを選択します。

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、リージョンスイッチは以下を実行します。

- リージョンスイッチが手動実行ブロックを実行すると、実行を一時停止し、計画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、リージョンスイッチは計画の実行に進みます。拒否した場合、リージョンスイッチは計画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。

計画評価の一環として評価されるもの

リージョンスイッチは、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。

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、保存ステップを選択します。

仕組み

- カスタムアクション Lambda 実行ブロックを作成するときは、ステップを実行するために 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、リージョンスイッチは Lambda 実行ブロックステップをスキップします。

計画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 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。リージョンスイッチは、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、以下の検証を含め、スロットリングされません。
  - 同時実行は 0 に設定されません。
  - 少なくとも 1 つの同時実行が利用可能であるか、予約されていない同時実行が存在する。

リージョンスイッチは 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。

リージョンスイッチは、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。リージョンスイッチ実行ブロックに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。

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、リージョンスイッチ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、プランが Lambda 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 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。リージョンスイッチ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。

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。

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: アプリケーションのトラフィックをリダイレクトするために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。
5. レコードセット識別子 (オプション): 必要に応じて、リージョンの切り替えで、トラフィックをそのリージョンにリダイレクトするために各リージョンに関連付けるレコードセットを識別するのに役立つ追加情報を提供します。必要に応じて入力する情報の識別は次のとおりです。
   - 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/ルートトラフィックを に入力できます。
   - リージョン: 入力した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。

リージョンスイッチは、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 のアカウントで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スイッチ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実行ブロックを追加すると、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。

リージョンスイッチは、実行ブロックの設定時に指定した情報に基づいて、プラン内のリージョンごとに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、リージョンスイッチが計画の作成後に自動計画評価を実行すると、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。

リージョンスイッチは、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異常に設定され、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。

計画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 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。リージョンスイッチは、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、リージョンスイッチは、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。

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、リージョンスイッチ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。

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スイッチ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、リージョンスイッチ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、リージョンスイッチ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
  - リージョンスイッチ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ます
- 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した場合、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

リージョンスイッチでアプリケーションの復旧を自動化する場合は、リージョンスイッチプランの 1 つ以上のトリガーを作成できます。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. 計画を作成したら、計画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。

   複数の条件を選択すると、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。

リージョンスイッチプランを実行してアプリケーションを復旧する

に障害 AWS リージョン 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スイッチプランを実行します。

もう 1 つのオプションは、計画実行を開始するために指定した特定の Amazon CloudWatch アラームを使用して実行を自動的にトリガーすることです。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 Management Console、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、リージョンスイッチを選択し、実行するプランを選択します。
3. 「計画の実行」を選択します。
4. 計画に手動承認ステップが含まれている場合は、プロンプトが表示されたら各ステップを承認します。

計画の実行中に、実行の詳細ページで進捗状況を追跡できます。これは、計画の実行を選択すると開きます。

リージョンスイッチダッシュボードで進行中のアプリケーション復旧に関する情報を表示することもできます。リージョンスイッチコンソールの左側のナビゲーションのリージョンスイッチで、次のいずれかを選択します。

- グローバルダッシュボード
- リージョン名での実行

リージョンに障害がある場合、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、ローカルリージョンスイッチデータプレーンを使用するため、回復力が向上します。

計画の実行が完了すると、計画の実行履歴タブの計画の詳細ページに、計画の実行に関する情報と、リージョン切り替えが実行されたその他の計画に関する情報が表示されます。

リージョン切り替えダッシュボード

リージョンスイッチには、組織およびリージョン全体のリージョンスイッチプランの状態を監視するために使用できるグローバルダッシュボードが含まれています。リージョンスイッチには、現在 にログインしているリージョンの計画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 AWS Management Console。

リージョンに障害がある場合、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、ローカルリージョンスイッチデータプレーンを使用するため、回復力が向上します。

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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. リージョンスイッチで、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. で ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. リージョンスイッチで、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

リージョンスイッチでは、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。

クロスアカウントリソース

リージョンスイッチを使用すると、リージョンスイッチ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。リージョンスイッチは、プランを実行するときに executionRole を引き受けます。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、リージョンスイッチは executionRole を使用して crossAccountRole を引き受け、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計画の共有

リージョンスイッチは AWS Resource Access Manager （ AWS RAM) と統合され、計画を共有できます 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、プランを実行し、プランの実行を表示できます。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。

リージョンスイッチでクロスアカウント共有を開始するには、 でリソース共有を作成します AWS RAM。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、共有する計画を所有している AWS アカウント 必要があります。共有されたプランを共有することはできません。組織または の組織単位と計画を共有するには AWS Organizations、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、「」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計画の共有をサポート。

ARC リージョン切り替えのアカウント間での計画の共有をサポート

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 を介してリソースを共有できるサービスです AWS Organizations。ARC リージョンスイッチでは、リージョンスイッチプラ

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、「」を参照してくださいクロスアカウントリソース。)

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。プランを共有する場合、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。

計画を共有すると、組織が必要とする計画の合計数を減らすことができます。共有プランを使用すると、プランを実行する合計コストをさまざまなチームに割り当てることができ、ARC の利点を低コストで最大化できます。アカウント間で計画を共有すると、特に複数のアカウントと運用チームに多数のアプリケーションが分散されている場合に、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容易になります。

ARC でクロスアカウント共有を開始するには、リソース共有を作成します AWS RAM。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。共有されたプランを共有することはできません。
- の組織または組織単位と計画を共有するには AWS Organizations、 との共有を有効にする必要があります AWS Organizations。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、プランを共有するために指定した参加者は を表示でき、追加のアクセス許可を付与した場合は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、既存のリソース共有にリソースを追加します。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、招待を承諾すると共有プランへのアクセス権が付与されます。

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、 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 AWS RAM。

```
# 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
```

が IAM AWS Resource Access Manager を使用する方法の詳細については、AWS RAM 「 ユーザーガイド」の「 が IAM AWS Resource Access Manager を使用する方法」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、参加者と所有者に以下が適用されます。

- 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、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、情報を表示することで共有プランを特定できます AWS RAM。また、ARC コンソールと 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 AWS CLI。

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、「 で共有リソースを表示する 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、「 で共有リソースを表示する 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 Management Console 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 Management Console、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。

参加者として、プランを共有する場合、通常、プランにアクセスできるように共有を受け入れる必要があります。

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、「」を参照してください読み取り専用アクセス許可。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。次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、プランに関連するコストが請求されます。プランでホストされるリソースの作成には、プラン所有者または参加者に対して追加料金はかかりません。

詳細な料金情報と例については、「Amazon Application Recovery Controller (ARC) 料金表」を参照し、「Amazon Application Recovery Controller (ARC)」までスクロールダウンします。

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、「」を参照してくださいリージョン切り替えのクォータ。

ARC でのリージョンスイッチの 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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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 のリージョンスイッチと 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 のリージョンスイッチと 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 で使用できる IAM 機能を確認してください。

IAM を使用して Amazon Application Recovery Controller (ARC) のリージョンスイッチへのアクセスを管理する前に、リージョンスイッチで使用できる IAM 機能について説明します。

Amazon Application Recovery Controller (ARC) のリージョンスイッチで使用できる IAM 機能

IAM の機能	リージョンスイッチのサポート
アイデンティティベースポリシー	はい
リソースベースのポリシー	はい
ポリシーアクション	はい
ポリシーリソース	あり
ポリシー条件キー	Yes
ACL	はい
ABAC (ポリシー内のタグ)	あり
一時的な認証情報	はい
プリンシパル権限	はい
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 機能とどのように連携するか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 ユーザーグループ、ユーザーのグループ、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。ID ベースの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。プリンシパルは、それが添付されている

ユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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 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、「」を参照してくださいAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例。

リージョンスイッチ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: 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。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは依存アクションと呼ばれます。

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。

ARC for Region スイッチのポリシーアクションは、アクションの前に次のプレフィックスを使用します。

```
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。たとえば、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のポリシーの例を表示するには、「」を参照してくださいARC でのリージョン切り替えのアイデンティティベースのポリシーの例。

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。ステートメントにはResource または NotResource 要素を含める必要があります。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。これは、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限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のポリシーの例を表示するには、「」を参照してくださいARC でのリージョン切り替えのアイデンティティベースのポリシーの例。

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 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。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。

1 つのステートメントに複数の Condition 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、 AWS では AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、「IAM ユーザーガイド」の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。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のポリシーの例を表示するには、「」を参照してくださいARC でのリージョン切り替えのアイデンティティベースのポリシーの例。

リージョンスイッチのアクセスコントロールリスト (ACLs)

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。

リージョンスイッチを使用した属性ベースのアクセスコントロール (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、属性に基づいてアクセス許可を定義する認可戦略です。では AWS、これらの属性はタグと呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合にオペレーションを許可するように ABAC ポリシーをします。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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。

TODO リカバリリージョンスイッチ (リージョンスイッチ) は ABAC をサポートしています。

リージョンスイッチ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

一部の AWS のサービス は、一時的な認証情報を使用してサインインすると機能しません。一時的な認証情報と AWS のサービス 連携する などの詳細については、AWS のサービス IAM ユーザーガイドの「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。たとえば、会社のシングルサインオン (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ユーザーから IAM ロールに切り替える (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用してアクセスすることができます AWS。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 の一時的セキュリティ認証情報」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アクセス許可

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、ユーザーとロールには ARC リソースを作成または変更するアクセス許可はありません。また、、 AWS Command Line Interface （ AWS CLI ） AWS Management Console、または AWS API を使用してタスクを実行することはできません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

これらサンプルの 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。

各リソースタイプの ARNs「サービス認可リファレンス」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
- 計画実行ロールの信頼ポリシー
- フルアクセス許可
- 読み取り専用アクセス許可
- 実行ブロックのアクセス許可
- クロスアカウントリソースアクセス
- 計画実行ロールポリシーを完了する

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、アカウント内で誰かが ARC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。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。例えば、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 AWS 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 Access Analyzer 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 Access Analyzer 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。

計画実行ロールの信頼ポリシー

これは、プランの実行ロールに必要な信頼ポリシーです。

```
{
  "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スイッチ APIs にフルアクセスを許可します。

```
{
  "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:CancelPlanExecution",
        "arc-region-switch:ListRoute53HealthChecks",
        "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。

```
{
  "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: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の IAM ポリシーについて説明します。

EC2 Amazon EC2 Auto Scaling 実行ブロック

EC2 Amazon EC2 Auto Scaling グループを管理するためのプラン実行ロールのポリシー：

```
{
  "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": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/AutoScaling"
        }
      }
    }
  ]
}
```

Amazon EKS リソーススケーリング実行ブロック

Amazon EKS クラスターを管理するためのプラン実行ロールのポリシー：

```
{
  "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

Amazon ECS サービスを管理するためのプラン実行ロールのポリシー：

```
{
  "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": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "ECS/ContainerInsights"
        }
      }
    }
  ]
}
```

ARC ルーティングコントロールの実行ブロック

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 (SCPs) 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`

ARC ルーティングコントロールを管理するためのプラン実行ロールのポリシー：

```
{
  "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をセットアップする」を参照してください。

Aurora グローバルデータベース実行ブロック

Aurora グローバルデータベースを管理するための計画実行ロールのポリシー：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "rds:DescribeGlobalClusters",
        "rds:DescribeDBClusters"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:FailoverGlobalCluster",
        "rds:SwitchoverGlobalCluster"
      ],
      "Resource": [
        "arn:aws:rds:us-east-1:123456789012: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"
      ]
    }
  ]
}
```

手動承認実行ブロック

手動ステップを承認できるロールのポリシー：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "arc-region-switch:ApprovePlanExecutionStep"
      ],
      "Resource": "arn:aws:arc-region-switch::123456789012:plan/sample-plan:0fba5e"
    }
  ]
}
```

カスタムアクション Lambda 実行ブロック

Lambda 関数を呼び出すためのプラン実行ロールのポリシー：

```
{
  "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

Route 53 ヘルスチェックを使用するプラン実行ロールのポリシー：

```
{
  "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計画実行ブロック

子プランを実行するプラン実行ロールのポリシー：

```
{
  "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-1:50c1a1",
        "arn:aws:arc-region-switch::123456789012:plan/child-plan-2:d1e5e1"
      ]
    }
  ]
}
```

アプリケーションのヘルスに関する CloudWatch アラーム

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の計画実行ロールのポリシー。実際の復旧時間を決定するために使用されます。

```
{
  "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"
      ]
    }
  ]
}
```

クロスアカウントリソースアクセス

リソースが異なるアカウントにある場合は、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。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/RegionSwitchExecutionRole"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

また、プラン実行ロールがこのクロスアカウントロールを引き受けるためのアクセス許可：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::987654321098:role/RegionSwitchCrossAccountRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "UniqueExternalId123"
        }
      }
    }
  ]
}
```

計画実行ロールポリシーを完了する

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は、かなり大きくなります。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。 ポリシーの例を次に示します。

```
{
```

```
  "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": "*"
    },
    // Include additional statements for specific execution blocks here
  ]
}
```

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るようにしてください。

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 と 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 でのリージョンスイッチの使用

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。キャプチャされた呼び出しには、ARC コンソールからの呼び出しと ARC API オペレーションへのコード呼び出しが含まれます。

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 イベントの継続的な配信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 で収集された情報を使用して、ARC に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト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 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 によってログに記録され、TBD API REFERENCE LINK に記載されています。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。たとえば、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スイッチアクションは、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、「」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。

ARC 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、リージョン切り替えの `StartPlanExecution`アクションを示す CloudTrail ログエントリを示しています。

```
{
    "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 でのリージョンスイッチの使用

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、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 トピックにシグナルを送信することで、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 リージョンの切り替えイベントを実行できます。

- リージョンスイッチプランの実行。イベントは、リージョン切り替えプランが実行された (実行された) ことを指定します。
- リージョン切り替え計画の評価。イベントは、リージョン切り替え計画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、EventBridge がイベントを検出するために使用できるイベント固有のパターンを定義します。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発生します。通常の運用状況では、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スイッチリソースをモニタリングする

EventBridge を使用すると、ARC がリージョンスイッチリソースのイベントを発行するときに実行するアクションを定義するルールを作成できます。

EventBridge コンソールで、イベントパターンを入力またはコピーして貼り付けるには、 オプション 自分のオプションを入力 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. AWS リージョン でルールを作成するには、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してください。

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 に貼り付けると、ARC アクションとリソースのモニタリングに使用できるルールを作成できます。

次のイベントパターンは、ARC のリージョン切り替え機能に EventBridge で使用できる例を示しています。

- PlanExecution のリージョンスイッチ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 のリージョンスイッチ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、リージョンスイッチプラン実行の 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が送信されるターゲットを指定する必要があります。EventBridge で使用可能なターゲットのリストについては、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、ルールの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と、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 で始まることを確認します/aws/events。既存のロググループを選択する場合は、 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/aws/events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。

コンソールの外部で 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使用して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。

```
{
  "Statement": [
    {
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "events.amazonaws.com",
          "delivery.logs.amazonaws.com"
        ]
      },
      "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
      "Sid": "TrustEventsToStoreLogEvent"
    }
  ],
  "Version": "2012-10-17"
}
```

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、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、イベントパターンや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. ターゲットの選択ページで、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) のリージョンスイッチには、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
プランあたりの実行ブロックの数	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の数	25
トリガー条件あたりの CloudWatch アラームの数	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、 SDKs で AWS Amazon Route 53 Application Recovery Controller の基本を使用する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

以下のコード例は、GetRoutingControlState の使用方法を示しています。

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。

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

以下のコード例は、UpdateRoutingControlState の使用方法を示しています。

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。

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からメリットを得られます。

セキュリティは、 AWS とユーザーの間で共有される責任です。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 を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティとコンプライアンスの目的を達成するように ARC を設定する方法について説明します。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検証
- Amazon Application Recovery Controller の耐障害性
- 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 Identity Center 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 が必須で、TLS 1.3 をお勧めします。
- 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

顧客の設定情報は、サービスが所有する Amazon DynamoDB グローバルテーブルに保存され、保管時には暗号化されます。

ARC クラスター内のセルのステータスを含むデータセットは、バックアップのために Amazon EBS ボリュームに書き込まれます。ARC は、データの保管中にデフォルトの Amazon EBS 暗号化を使用します。

転送中の暗号化

ARC 設定、準備状況ステータスクエリ、セル状態の更新などのお客様のリクエストとレスポンスは、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、誰を認証 (サインイン) し、誰に ARC リソースの使用を許可する (アクセス許可を付与する) 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。

対象者

AWS Identity and Access Management (IAM) の使用方法は、ARC で行う作業によって異なります。

サービスユーザー – ARC サービスを使用してジョブを実行する場合、管理者は必要な認証情報とアクセス許可を提供します。さらに多くの ARC 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解すると、管理者に適切なアクセス許可をリクエストするのに役に立ちます。ARC の機能にアクセスできない場合は、「」を参照してください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。

サービス管理者 – 社内の ARC リソースを担当している場合は、通常、ARC へのフルアクセスがあります。サービスユーザーがどの ARC 機能とリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社が ARC で IAM を使用する方法の詳細については、「」を参照してくださいAmazon Application Recovery Controller (ARC) 機能が IAM と連携する方法。

IAM 管理者 – IAM 管理者は、ARC へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。IAM で使用できる 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、「」を参照してくださいAmazon Application Recovery Controller (ARC) のアイデンティティベースのポリシーの例。

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。として、IAM ユーザーとして 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によって、認証 (にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center （ IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook 認証情報は、フェデレーション ID の例です。フェデレーティッド ID としてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーの種類に応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 AWS サインイン ユーザーガイド」の「 へのサインイン AWS アカウント方法」を参照してください。

AWS プログラムで にアクセスする場合、 はソフトウェア開発キット (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストを暗号化して署名します。AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。リクエストに自分で署名する推奨方法の使用については、「IAM ユーザーガイド」の「API リクエストに対するAWS Signature Version 4」を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。たとえば、 では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお勧めします。詳細については、「AWS IAM Identity Center ユーザーガイド」の「多要素認証」および「IAM ユーザーガイド」の「IAM のAWS 多要素認証」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、アカウント内のすべての およびリソースへの AWS のサービス完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。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。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実

行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、「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。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、ID プロバイダーとのフェデレーションを使用して一時的な認証情報 AWS のサービス を使用して にアクセスすることを要求します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、アイデンティティセンターディレクトリ、または ID ソースを介して提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッドアイデンティティがアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成するか、独自の ID ソースのユーザーとグループのセットに接続して同期し、すべての AWS アカウント とアプリケーションで使用できます。IAM Identity Center の詳細については、「AWS IAM Identity Center ユーザーガイド」の「What is IAM Identity Center?」(IAM Identity Center 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、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内の ID です。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーション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「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユー

ザーには永続的な長期の認証情報がありますが、ロールでは一時認証情報が提供されます。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 AWS アカウント を持つ 内の ID です。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。で IAM ロールを一時的に引き受けるには AWS Management Console、ユーザーから IAM ロール (コンソール) に切り替えることができます。ロールを引き受けるには、 または AWS API オペレーションを AWS CLI 呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「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 ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーティッド ID が認証されると、その ID はロールに関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールについては、「IAM ユーザーガイド」の「サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する」を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center User Guide」の「Permission sets」を参照してください。
- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、(プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。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。
- クロスサービスアクセス — 一部の は他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。

- 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して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。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、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストをリクエストする と組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。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。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
- サービスロール - 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、削除することができます。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。
- 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。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。EC2 インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」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 AWS 、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) 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。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。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構造と内容の詳細については、「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 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。

デフォルトでは、ユーザーやロールに権限はありません。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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

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ポリシーは、IAM ユーザーグループ、ユーザーのグループ、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。

アイデンティティベースのポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたは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。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制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。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。プ

リンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

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。

アクセスコントロールリスト (ACL)

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。

Amazon S3、および Amazon VPC は AWS WAF、ACLs。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的でない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の AWS アカウント をグループ化して一元管理するためのサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー (SCP)」を参照してください。
- リソースコントロールポリシー (RCP) – RCP は、所有する各リソースにアタッチされた IAM ポリシーを更新することなく、アカウント内のリソースに利用可能な最大数のアクセス許可を設定する

ために使用できる JSON ポリシーです。RCP は、メンバーアカウントのリソースのアクセス許可を制限し、組織に属しているかどうかにかかわらず AWS アカウントのルートユーザー、 を含む ID の有効なアクセス許可に影響を与える可能性があります。RCP をサポートする のリストを含む Organizations と RCP の詳細については、AWS Organizations RCPs「リソースコントロールポリシー (RCPs」を参照してください。 AWS のサービス

- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。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権限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) 機能が IAM と連携する方法

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用の IAM
- 準備状況チェック用の IAM
- リージョンスイッチの 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の例
- 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 でのルーティングコントロールのアイデンティティベースのポリシーの例
- ARC での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管理ポリシー
- 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管理ポリシー

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アラートについては、ARC ドキュメント履歴ページの RSS フィードにサブスクライブしてください。

変更	説明	日付
AWSZonalAutoshiftPracticeRunSLRPolicy 管理ポリシー – 更新されたポリシー	、autoscaling:DescribeAutoScalingGroups 、ec2:DescribeInstances 、および のアクセス許可AutoshiftPracticeCheckPermissions を持つポリシーステートメントを追加してelasticloadbalancing:DescribeTargetHealth 、バランスの取れたキャパシティチェックelasticlo	2025 年 6 月 30 日

変更	説明	日付
	`adbalancing:DescribeTargetHealth` 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 は、オートシフトと練習実行用の新しいサービスにリンクされたロールを追加しました。 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、お客様が提供する Amazon CloudWatch アラームとお客様 AWS 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

変更	説明	日付
AmazonRoute53RecoveryControlConfigReadOnlyAccess — ポリシーの更新	共有リソースの AWS Resource Access Manager リソースポリシーに関する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

Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC は、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を追加しました。
	
	ARC は、次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、インスタンスの準備状況ステータスを判断します。
	
	`ec2:DescribeVpnGateways`
	
	`ec2:DescribeCustomerGateways`
		2023 年 2 月 17 日

変更	説明	日付
Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC は、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を追加しました。 ARC は、次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、関数の準備状況ステータスを判断します。 `lambda:ListProvisionedConcurrencyConfigs`	2022 年 8 月 31 日
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 が追加され`route53-recovery-cluster:ListRoutingControls` 、高可用性のルーティングコントロール ARNs一覧表示できるようになりました。	2022 年 3 月 15 日

変更	説明	日付
AmazonRoute53RecoveryControlConfigReadOnlyAccess — ポリシーの更新	ARC は、リソース`route53-recovery-control-config:ListTagsForResources` のタグの一覧表示を許可する新しいアクセス許可 を追加しました。	2021 年 12 月 20 日
Route53RecoveryReadinessServiceRolePolicy – ポリシーの更新	ARC は、Amazon API Gateway に関する情報をクエリするための新しいアクセス許可を追加しました。 ARC は、 アクセス許可 を使用して API Gateway に関する情報をクエリし`apigateway:GET` 、準備状況チェックを実行し、準備状況ステータスを判断します。	2021 年 10 月 28 日

変更	説明	日付
AmazonRoute53RecoveryReadinessReadOnlyAccess - 新しいアクセス許可を追加	ARC は AmazonRoute53RecoveryReadinessReadOnlyAccess に 2 つの新しいアクセス許可を追加しました。 ARC は `route53-recovery-readiness:GetArchitectureRecommendations` と `route53-recovery-readiness:GetCellReadinessSummary` を使用して、復旧の準備を整えるために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 は、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を追加しました。
	
	ARC は、次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、それらの関数の準備状況ステータスを判断します。
	
	`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 は、次の新しい管理ポリシーを追加しました。 AmazonRoute53RecoveryReadinessFullAccess AmazonRoute53RecoveryReadinessReadOnlyAccess AmazonRoute53RecoveryClusterFullAccess AmazonRoute53RecoveryClusterReadOnlyAccess AmazonRoute53RecoveryControlConfigFullAccess AmazonRoute53RecoveryControlConfigReadOnlyAccess	2021 年 8 月 18 日
ARC が変更の追跡を開始	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

以下の情報は、Amazon Application Recovery Controller (ARC) と IAM の使用時に発生する可能性がある一般的な問題の診断と修正に役立ちます。

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 Management Console 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、ARC にロールを渡すことができるようにポリシーを更新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権限が必要です。

次の例のエラーは、 という IAM ユーザーがコンソールを使用して marymajor ARC でアクションを実行しようとすると発生します。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権限が必要です。メアリーには、ロールをサービスに渡す許可がありません。

```
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、「」を参照してくださいいAmazon Application Recovery Controller (ARC) 機能が IAM と連携する方法。
- 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「サードパーティーが所有する へのアクセスを提供する AWS アカウント」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。

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ツールがいくつか AWS 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、AWS のサービス「コンプライアンスプログラムによる範囲内」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 「 Compliance ProgramsAssurance」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。 は、コンプライアンスに役立つ以下のリソース AWS を提供します。

- セキュリティのコンプライアンスとガバナンス – これらのソリューション実装ガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスの機能をデプロイする手順を示します。
- HIPAA 対応サービスのリファレンス – HIPAA 対応サービスの一覧が提供されています。すべてが HIPAA AWS のサービス の対象となるわけではありません。
- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド – コンプライアンスの観点から責任共有モデルを理解します。このガイドは、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールを保護し、そのガイダンスに AWS のサービス マッピングするためのベストプラクティスをまとめたものです。
- 「 デベロッパーガイド」の「ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が内部プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config

- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に把握できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールの一覧については、Security Hub のコントロールリファレンスを参照してください。
- Amazon GuardDuty – 不審なアクティビティや悪意のあるアクティビティがないか環境をモニタリングすることで AWS アカウント、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。GuardDuty を使用すると、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件に対応できます。
- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Amazon Application Recovery Controller の耐障害性

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イ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が高く、フォールトトレラントで、スケーラブルです。

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。

ARC には、 AWS グローバルインフラストラクチャに加えて、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 AWS 環境を設計するには、「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。クライアントは以下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、ほとんどの最新システムでサポートされています。

また、リクエストにはアクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token ServiceAWS STSを使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon Application Recovery Controller (ARC) デベロッパーガイドのドキュメント履歴

以下のエントリでは、Amazon Application Recovery Controller (ARC) ドキュメントに加えられた重要な変更について説明します。

- バージョン: 最新
- ドキュメントの最終更新日: 2025 年 8 月 1 日

変更	説明	日付
新しいリージョン切り替えサービス	リージョンスイッチを使用すると、お客様は、マルチリージョンアプリケーションを別の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 AWS リージョン。 詳細については、「ARC のリージョンスイッチ」を参照してください。	2025 年 8 月 1 日
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 AZs 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:DescribeAutoScalingGroup	2025 年 6 月 30 日

変更	説明	日付
	s 、ec2:DescribeInstances 、および のアクセス許可Autoshift PracticeCheckPermissions を持つポリシーステートメントを追加してAWSZonalAutoshiftPracticeRunSLRPolicy 管理ポリシーを更新 elasticloadbalancing:DescribeTargetHealth しelasticloadbalancing:DescribeTargetHealth 、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管理ポリシー」を参照してください。	
ゾーンオートシフトの例外タイプの更新	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

変更	説明	日付
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 が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 が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セットアップ」を参照してください。	2024 年 11 月 21 日
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

変更	説明	日付
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か、ゾーンオートシフトを有効にして AWS にゾーンシフトを開始させることができます。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
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

変更	説明	日付
Autoshift 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、 がオートシフト AWS 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 を通じ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を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、イベント中の復旧までの時間を短縮します。 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 でのクラスターのクロスアカウントのサポート」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ec2:DescribeCustomerGateways のポーリングをサポートするために、ARC のサービスにリンクされたロールに新しいアクセス許可 ec2:DescribeVpnGateways と 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	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コントロール (ABAC) を含む、ARC のゾーンシフトの GA リリースをサポートします。 詳細については、「ARC を使用した属性ベースのアクセスコントロール (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、ゾーンシフトの新しいサービスを説明するコンテンツを追加しました。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	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 ARC のサービスにリンクされたロールに追加しました。 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、AWS「Amazon Application Recovery Controller (ARC) の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、AWS「Amazon Application Recovery Controller (ARC) の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	信頼性の高い ARC データプレーン API に含まれる新しいリストルーティングコントロール API オペレーションの CLI コマンドの例とベストプラクティスの推奨事項を追加しました。 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のサポートが追加されました。 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、AWS「Amazon Application Recovery Controller (ARC) の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 のサポートが追加されました。つまり、アラートを取得し、ステータスが 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変更に対応する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 での ARC の使用」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、AWS「Amazon Application Recovery Controller (ARC) の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である Amazon API Gateway を追加し、ARC が準備状況チェックで API Gateway を監査できるように ARC サービスにリンクされたロールのアクセス許可を更新しました。 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である Lambda 関数を追加し、ARC が準備状況チェックで Lambda 関数を監査できるように ARC サービスにリンクされたロールのアクセス許可を更新しました。 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 AWS 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、AWS「Amazon Application Recovery Controller (ARC) の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 AWS 管理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、AWS「Amazon Application Recovery Controller (ARC) の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 は、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するようにスケーリングされ、障害を回避するように設定されていることを確認するための準備状況チェックを提供します。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。