aws

開発者ガイド

Amazon Route 53 Application Recovery Controller

Amazon Route 53 Application Recovery Controller: 開発者ガイド

Table of Contents

Amazon Route 53 Application Recovery Controller とは

Amazon Route 53 Application Recovery Controller (Route 53 ARC) は、 で実行されているアプリケーションのリカバリの準備と完了に役立ちます AWS。Route 53 ARC には、ゾーンシフトとゾーンオートシフトを含むマルチアベイラビリティーゾーン (AZ) リカバリ と、ルーティングコントロールと準備状況チェックを含むマルチリージョンリカバリ の 2 つの機能セットが用意されています。Route 53 ARC を使用すると、高可用性リカバリツールを活用して、マルチリージョンまたはマルチ AZ アプリケーションに影響を与える障害をすばやく軽減できます。また、準備状況チェックを使用して、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取得することもできます。

AWS グローバルクラウドインフラストラクチャは、耐障害性と耐障害性を提供し、それぞれ AWS リージョン が完全に分離された複数のアベイラビリティーゾーンで構成されています。Route 53 ARC はこの AWS 構造内で動作し、アプリケーションの耐障害性を高めます。

マルチ AZ リカバリ

のアベイラビリティーゾーンを利用するように構築されたアプリケーションがある場合は AWS、ゾーンシフトを使用して 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一時的に AZ から 内の正常な AZ に移動することで、アベイラビリティーゾーン (AZ) AZs障害から回復できます AWS リージョン。ゾーンシフトを開始すると、デベロッパーの不正なコードのデプロイや単一のアベイラビリティーゾーンの AWS 障害などから、アプリケーションを迅速に復旧できます。トラフィックを遠ざけることで、1 つの AZ で問題が発生した場合に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を軽減できます。

リージョン内のアカウントでサポートされているリソースのゾーンシフトを開始できます。 AWS サービスは、サポートされている AWS リソースを Route 53 ARC のゾーンシフトに自動的に登録するため、いつでもゾーンシフトを開始できます。

ゾーンオートシフトは、Route 53 ARC の機能であり、お客様に代わって、サポートされているリソースの AZ から正常な AZs にトラフィックをシフト AWS することを に許可できます AWS リージョン。内部テレメトリで、顧客に影響を与える可能性のあるリージョン内の 1 つの AZ に障害があることが示された場合、 はオートシフト 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。

ゾーンシフトとオートシフトは一時的なものです。手動ゾーンシフトを開始するときは、最初は最大 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。引き続き AZ からトラフィックを遠

ざける場合は、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。ゾーンオートシフトでは、は、問題や潜在的な問題がなくなったことがインジケータに表示されるとオートシフトを 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。

- Amazon Route 53 Application Recovery Controller のゾーンシフト
- Amazon Route 53 Application Recovery Controller のゾーンオートシフト

マルチリージョンリカバリ

オペレーションを続行するために別の から運用 AWS リージョン するように設計されたアプリケーションがある場合は、フェイルオーバーのルーティング制御を使用できます。ルーティングコントロールを使用すると、問題が発生したときにトラフィック AWS リージョン をフェイルオーバーできるため、アプリケーションが引き続き利用可能になります。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。これらのルールを使用すると、例えば、アクティブまたはスタンバイのアプリケーションレプリカの 1 つだけが有効で、一度に使用されていることを確認することができます。

マルチリージョンリカバリの場合、Route 53 ARC は 全体で DNS トラフィックをフェイルオーバーするのに役立ちます AWS リージョン。Route 53 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、障害のあるリージョンから正常なリージョンにトラフィックを再ルーティングすることで、アプリケーションを復旧できます。

準備状況チェックでは、Route 53 ARC 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、レプリカへのフェイルオーバーと復旧の機能に影響す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ことを確認するのに役立ちます。準備状況チェックは、Route 53 ARC を初めて設定するとき、および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使用することを目的としたものではありません。

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。

- Amazon Route 53 Application Recovery Controller でのルーティングコントロール
- Amazon Route 53 Application Recovery Controller の準備状況チェック

Amazon Route 53 Application Recovery Controller のマルチ AZ とマルチリージョンリカバリ機能を比較する

Amazon Route 53 Application Recovery Controller のゾーンシフト、ゾーン自動シフト、ルーティングコントロールはすべて、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。これらのオプションは可用性が高く、アプリケーションでレイテンシーが増加したり可用性が低下したりする場合のシナリオで復旧をサポートします。これらのオプションは、トラフィックを分離された障害から遠ざけることで、アプリケーションを迅速に回復するのに役立ちます。これにより、障害による影響と時間の損失を抑えることができます。

ルーティングコントロールは主に、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自動shift は、マルチ AZ アプリケーションを使用するロードバランサーのトラフィックの移行のみをサポートします。このセクションで説明するように、他にも相違点があります。

次の表の情報には、ゾーンシフト、ゾーン自動シフト、ルーティング制御の主な機能、およびオプション間の比較方法が含まれています。これらの説明は、特定のオプションが組織のディザスタリカバリニーズに最も適している可能性がある方法を理解するのに役立ちます。

ルーティングコントロール	ゾーンシフト	ゾーンオートシフト
リージョン別	ゾーン別	ゾーン別
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) アベイラビリティーゾーン間の再ルーティングも可能	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	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
セットアップが必要 構成とセットアップが必要	セットアップなしで使用可能 サポート対象のサービスによって自動的に使用可能となる	練習実行のセットアップが必要 サポート対象のサービスで利用可能

ルーティングコントロール	ゾーンシフト	ゾーンオートシフト
	(現在は、Network Load Balancer と Application Load Balancer)	(現在は、Network Load Balancer と Application Load Balancer)
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 から遠ざけます。
有料 ルーティングコントロールは別料金	サービスに込み AZ からトラフィックを遠ざけるためのゾーンシフトの作成は、サポート対象のロードバランサーに含まれる	サービスに込み ユーザーに代わって AZ からトラフィックを遠ざけるためのオートシフトの開始は、サポート対象のロードバランサーに含まれる
有効期限なし トラフィックはレプリカに無期限で再ルーティング可能	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 自動シフトを開始および終了する

これらの各機能の詳細については、次の章を参照してください。

- Amazon Route 53 Application Recovery Controller のゾーンシフト
- Amazon Route 53 Application Recovery Controller のゾーンオートシフト
- Amazon Route 53 Application Recovery Controller でのルーティングコントロール

ゾーンシフトとゾーンオートシフトを使用して Amazon Route 53 Application Recovery Controller でアプリケーションを復旧する

このセクションでは、Amazon Route 53 Application Recovery Controller の機能を使用して、アベイラビリティーゾーン (AZ) の問題から AWS アプリケーションを確実に復旧する方法について説明します。これらの機能、ゾーンシフト、ゾーン自動シフトは、トラフィックを Elastic Load Balancing リソースの AZ から一時的に移動して、アプリケーションの復旧までの時間を短縮します。

ゾーンシフトとゾーン自動シフトの主な違いは、1 つは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障害から自動的に移行することです。

- ゾーンシフトでは、 内のマネージド Elastic Load Balancing 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 リージョン 手動で移動します。
- ゾーン自動shift を使用すると、Elastic Load Balancing トラフィックは、ユーザーに代わってイベント中に、障害が発生した AZ からリージョン内の正常な AZs に自動的に移行されます。

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

トピック

Amazon Route 53 Application Recovery Controller のゾーンシフト

Amazon Route 53 Application Recovery Controller のゾーンシフトを使用すると、Elastic Load Balancing リソースのトラフィックを のアベイラビリティーゾーンから遠ざけて AWS リージョン、問題をすばやく軽減し、アプリケーションをすばやく復旧できます。この機能を使用するには、Elastic Load Balancing リソースでクロスゾーン負荷分散がオフになっている必要があります。

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で AWS アプリケーションをデプロイして実行すると、ゾーンシフトを開始することで、障害が発生した AZ でアプリケーションをすばやく復旧できます。アプリケーショントラフィックを正常な AZs に移行すると、停電、または AZ のハードウェアや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ます。

例えば、不適切なデプロイによってレイテンシーの問題が発生している場合や、アベイラビリティーゾーンに障害が発生している場合などに、トラフィックをシフトすることを選択できます。ゾーンシフトでは事前の設定手順は必要ありませんが、シフト元のアベイラビリティーゾーンを使用せずにクライアントロードを処理するには、 AWS 設定でサポートする必要があります。サポートされているロードバランサーリソースは Amazon Route 53 Application Recovery Controller に自動的に登録されるため、必要に応じてロードバランサーのゾーンシフトを簡単に開始できます。

ゾーンシフトを開始するには、セットアップや設定は必要ありません。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したら、遠ざけるアベイラビリティーゾーンとトラフィックを遠ざけるリソースを選択し、ゾーンシフトを開始します。シフトはいつでもキャンセルでき、トラフィックがアベイラビリティーゾーンに戻り始めることができます。

ゾーンシフトはすべて一時的な緩和策です。ゾーンシフトを開始するときに、1 時間から 3 日 (72 時間) まで初期有効期限を設定します。この有効期限は、トラフィックシフトを続行する必要がある場合に延長できます。

いくつかの特定のシナリオでは、ゾーンシフトは AZ からのトラフィックをシフトしないことに注意してください。ゾーンシフトのサポートに関する詳細は、「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照してください。

ゾーンシフトの仕組み

ロードバランサー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から移動されます。シフトを開始するために、Amazon Route 53 Application Recovery Controller は、アベイラビリティーゾーンのロードバランサーのヘルスチェックを異常に設定して、ヘルスチェックに失敗するように要求します。ヘルスチェックが正常でない場合、Amazon Route 53 はリソースの対応する IP アドレスを DNS から自動的に取り消し、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。

ゾーンシフトでは、ヘルスチェックがロードバランサーやアプリケーションの基盤となるヘルスをモニタリングする一般的な方法ではヘルスチェックを使用しないことに注意してください。代わりに、Route 53 ARC 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メカニズムとしてヘルスチェックを使用します。このメカニズムは、ヘルスチェックを明示的に異常に設定し、次に再び正常に設定して、トラフィックのフローを変更することを要求します。

トラフィックがシフトし始める - Route 53 ARC でゾーンシフトを開始すると、トラフィックフローに関連するステップが原因で、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。また、クライアントの挙動または接続の再利用によっては、アベイラビリティーゾー

ンで進行中の既存の接続が完了するまでに、若干時間がかかる場合もあります。DNS 設定やその他の要因によっては、既存の接続が数分で完了したり、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すぐ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、Route 53 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ます。これにより、トラフィックシフトを開始するプロセスが逆になり、Route 53 ヘルスチェックが再び正常になるように要求されます。正常なヘルスチェックでは、元のゾーン IP アドレスが復元されます。これで、復旧したアベイラビリティーゾーンがロードバランサーのルーティングに再び含まれ、トラフィックが AZ へのフローを再開し始めます。

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。

トラフィックが移動しない場合

いくつかの特定のシナリオでは、ゾーンシフトで、AZ からトラフィックがシフトされません。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の状態になります。このケースでは、ロードバランサーでゾーンシフトを開始しても、そのロードバランサーは既にフェイルオープンの状態になっているため、ロードバランサーが使用する AZ をゾーンシフトによって変えることはできません。これは想定される動作です。すべての AZ がフェイルオープンの状態になっている (異常がある) 場合、ゾーンシフトで 1 つの AZ を「異常」にしてトラフィックをそのリージョン内の他の AZ にシフトすることはできません。2 つ目のケースは、 AWS Global Accelerator にあるアクセラレーターのエンドポイントである Application Load Balancer で、ゾーンシフトを開始する場合です。Global Accelerator にあるアクセラレーターのエンドポイントである Application Load Balancer では、ゾーンシフトはサポートされていません。

ゾーンシフトのサポートに関する詳細は、「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照してください。

AWS リージョン ゾーンシフトの可用性

Amazon Route 53 Application Recovery Controller のリージョンサポートとリージョンサービスエンドポイントに関する詳細は、Amazon Web Services 全般のリファレンスにある「Amazon Route 53 Application Recovery Controller のエンドポイントとクォータ」を参照してください。

ゾーンシフトは現在、ここ AWS リージョン に記載されている で利用できます。ゾーンシフトは、中国 (北京) リージョンおよび中国 (寧夏) リージョンでも利用できます。

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	
米国東部 (バージニア北部)	us-east-1	arc-zonal-shift.us-east-1.amazonaws.com	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	
米国西部 (オレゴン)	us-west-2	arc-zonal-shift.us-west-2.amazonaws.com	HTTPS	
アフリカ (ケープタウン)	af-south-1	arc-zonal-shift.af-south-1.amazonaws.com	HTTPS	
アジアパシフィック (香港)	ap-east-1	arc-zonal-shift.ap-east-1.amazonaws.com	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	

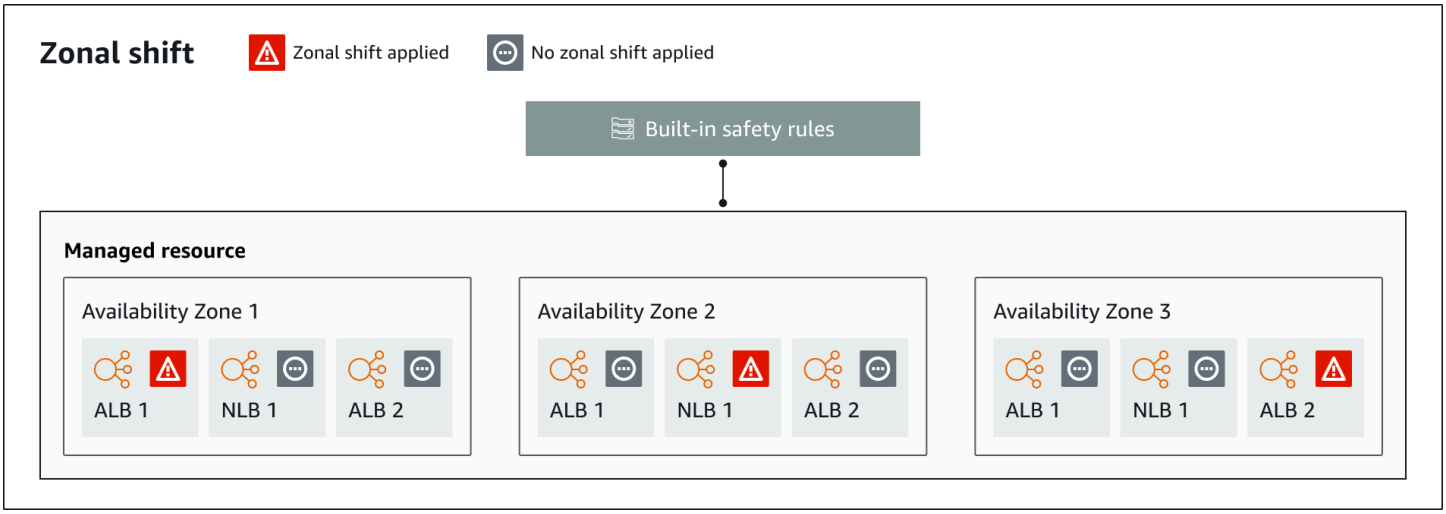
リージョン名	リージョン	エンドポイント	プロトコル	
アジアパシフィック (ジャカルタ)	ap-southeast-3	arc-zonal-shift.ap-southeast-3.amazonaws.com	HTTPS	
アジアパシフィック (メルボルン)	ap-southeast-4	arc-zonal-shift.ap-southeast-4.amazonaws.com	HTTPS	
アジアパシフィック (ムンバイ)	ap-south-1	arc-zonal-shift.ap-south-1.amazonaws.com	HTTPS	
アジアパシフィック (大阪)	ap-northeast-3	arc-zonal-shift.ap-northeast-3.amazonaws.com	HTTPS	
アジアパシフィック (ソウル)	ap-northeast-2	arc-zonal-shift.ap-northeast-2.amazonaws.com	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	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	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
アジアパシフィック (東京)	ap-northeast-1	arc-zonal-shift.ap-northeast-1.amazonaws.com	HTTPS	
カナダ (中部)	ca-central-1	arc-zonal-shift.ca-central-1.amazonaws.com	HTTPS	
カナダ西部 (カルガリー)	ca-west-1	arc-zonal-shift.ca-west-1.amazonaws.com	HTTPS	
欧州 (フランクフルト)	eu-central-1	arc-zonal-shift.eu-central-1.amazonaws.com	HTTPS	
欧州 (アイルランド)	eu-west-1	arc-zonal-shift.eu-west-1.amazonaws.com	HTTPS	
欧州 (ロンドン)	eu-west-2	arc-zonal-shift.eu-west-2.amazonaws.com	HTTPS	
ヨーロッパ (ミラノ)	eu-south-1	arc-zonal-shift.eu-south-1.amazonaws.com	HTTPS	
欧州 (パリ)	eu-west-3	arc-zonal-shift.eu-west-3.amazonaws.com	HTTPS	
欧州 (スペイン)	eu-south-2	arc-zonal-shift.eu-south-2.amazonaws.com	HTTPS	
欧州 (ストックホルム)	eu-north-1	arc-zonal-shift.eu-north-1.amazonaws.com	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル	
欧州 (チューリッヒ)	eu-central-2	arc-zonal-shift.eu-central-2.amazonaws.com	HTTPS	
イスラエル (テルアビブ)	il-central-1	arc-zonal-shift.il-central-1.amazonaws.com	HTTPS	
中東 (バーレーン)	me-south-1	arc-zonal-shift.me-south-1.amazonaws.com	HTTPS	
中東 (アラブ首長国連邦)	me-central-1	arc-zonal-shift.me-central-1.amazonaws.com	HTTPS	
南米 (サンパウロ)	sa-east-1	arc-zonal-shift.sa-east-1.amazonaws.com	HTTPS	
AWS GovCloud (米国東部)	us-gov-east-1	arc-zonal-shift.us-gov-east-1.amazonaws.com	HTTPS	
AWS GovCloud (米国西部)	us-gov-west-1	arc-zonal-shift.us-gov-west-1.amazonaws.com	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遠ざけ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により、リソースに既にアクティブなシフトがある場合に、別のゾーンシフトを開始できなくなります。



Route 53 ARC のゾーンシフト機能のコンポーネントを次に示します。

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。現在、ゾーンシフトを開始できるのは、クロスゾーン負荷分散が設定されていない Network Load Balancer と Application Load Balancer のみです。サポートされているロードバランサーは、Route 53 ARC に自動的に登録されます。

組み込み安全チェック

Route 53 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、そのリソースについて、顧客によって開始されたゾーンシフト、練習実行のゾーンシフト、またはオートシフトの 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、「Amazon Route 53 Application Recovery Controller 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、Route 53 ARC でサポートされている サービスのアカウント AWS 内のリソースのみを選択できます。これらの AWS サービスでサポートされているリソースは、 AWS サービスによって Route 53 ARC に自動的に登録されます。

> **Note**
>
> 現在、クロスゾーン負荷分散がオフになっている Network Load Balancer と Application Load Balancer のゾーンシフトのみを開始できます。

マネージドリソース

AWS のサービスは、ゾーンシフト用にリソースを Route 53 ARC に自動的に登録します。登録されたリソースは Route 53 ARC のマネージドリソースとなります。

リソース名

ゾーンシフトに指定できる Route 53 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遠ざけら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。

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。顧客が開始したゾーンシフトの場合、最初は 3 日間 (72 時間) 有効になるように設定できます。

ゾーンシフトを開始するときに、ゾーンシフトをアクティブにする時間を指定します。Route 53 ARC によってそれが有効期限 (満了期限) に変換されます。例えば、トラフィックをアベイラビリティーゾーンに戻す準備ができた場合は、顧客が開始した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を使用した練習実行で AWS 開始される、顧客主導のゾーンシフトとゾーンシフトの両方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。フェイルオーバー時に依存するメカニズムが可用性が高く、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には、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。これらはいずれも信頼性の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 が高可用性の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 Route 53 Application Recovery Controller 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。

Amazon Route 53 Application Recovery Controller で使用した分に対してのみお支払いいただきます。Route 53 ARC と料金例の詳細な料金情報については、「Amazon Route 53 の料金」を参照し、Amazon Route 53 Application Recovery Controller までスクロールします。

Route 53 ARC のゾーンシフトのベストプラクティス

Route 53 ARC で、マルチ AZ リカバリにゾーンシフトを使用する場合は、次のベストプラクティスに従うことが推奨されます。ゾーンシフトは、一般に、ライブアプリケーションのキャパシティを奪うため、本番で使用する場合は注意が必要です。

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

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。

例えば、1 つのロードバランサーリソースでゾーンシフトを開始すると、1 つのアベイラビリティーゾーンのキャパシティが、ロードバランサーの後方で一時的に削除されます。開始するゾーンシフトと、ロードバランサーの設定方法によっては、残りのアベイラビリティーゾーンで、増加する負荷に対応するための計画を慎重に立てておくことが必要になります。

クライアントがエンドポイントに接続したままになる時間を制限する

Amazon Route 53 Application Recovery Controller 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、Route 53 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 を更新すると、すべての新しい接続が障害が発生した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引き続きリクエストを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るだけでなく、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移動させるというトレードオフであることに注意してください。

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、トラフィックを受け取れるようにする

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。アプリケーションのロードバランサーのターゲットが概ね正常であり、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が受け入れられていることを確認するために欠かせません。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。

2 番目の隣接するリージョンからリソースの正常性をモニタリングすることを検討してください。この方法を使うことで、エンドユーザーの体験をより多く反映させることができ、さらに、アプリケーションとモニタリングの両方が同じ災害から同時に影響を受けるリスクを下げることができます (「運命を分け合う」)。

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。でゾーンシフトを開始して AWS Management Console、使いやすく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移動する

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

ゾーンシフト API オペレーション

次の表は、マルチ AZ アプリケーションについてアベイラビリティーゾーンからトラフィックを遠ざけるゾーンシフトで使用できる Route 53 ARC API オペレーションを、関連するドキュメントへのリンクと共に一覧にしたものです。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	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
ゾーンシフトを開始する	「ゾーンシフトの開始」を参照	StartZonalShift「��
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	UpdateZonalShift「��
ゾーンシフトを一覧表示する	「Amazon Route 53 Application Recovery Controller のゾーンシフト」を参照	ListZonalShifts「��
マネージドリソースを一覧表示する	「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照	ListManagedResources「��
マネージドリソースを取得する	「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照	GetManagedResource「��
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	CancelZonalShift「��

ゾーンシフト AWS CLI で を使用する例

このセクションでは、 を使用して、API オペレーションを使用する Amazon Route 53 Application Recovery Controller のゾーンシフト機能を操作する AWS Command Line Interface 、ゾーンシフト

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 を使用してゾーンシフトを操作する方法に関する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。

Route 53 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一時的にアベイラビリティーゾーンから遠ざけることができるため、アプリケーションは 内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。ゾーンシフトは、クロスゾーン負荷分散がオフになっている Application Load Balancer と Network Load Balancer でのみサポートされます。

AWS Command Line Interfaceを使用してゾーンシフトを開始する例を見てみましょう。また、AWS CLI を使用してゾーンシフトを更新し、例えば新しい有効期限を設定することもできます。すべてのゾーンシフトは一時的なもので、最初は 3 日以内に期限切れになるように設定する必要があります。ただし、後でゾーンシフトを更新して新しい有効期限を設定できます。

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

ゾーンシフトを開始する

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

```
aws arc-zonal-shift start-zonal-shift \
      --resource-identifier="arn:aws:testservice::111122223333:ExampleALB123456890" \
      --away-from="usw2-az1" \
      --expires-in="5m" \
      --comment="Shifting traffic away from USW2-AZ1"
```

```
{
   "zonalShiftId": "2222222-3333-444-1111",
   "resourceIdentifier": "arn:aws:testservice::111122223333:ExampleALB123456890",
   "awayFrom": "usw2-az1",
   "expiryTime": 2022-11-14T01:40:42+00:00,
   "startTime": 2022-11-14T01:35:42+00:00,
   "status": "ACTIVE",
   "comment": "Shifting traffic away from USW2-AZ1"
}
```

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:testservice::111122223333:ExampleALB123456890
```

```
{
    "arn": "arn:aws:testservice::111122223333:ExampleALB123456890",
    "name": "TestResource",
    "appliedWeights": {
        "usw2-az1": 1.0,
        "usw2-az2": 1.0,
        "usw2-az3": 1.0
    },
    "zonalShifts": []
}
```

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": [
        {
            "arn": "arn:aws:testservice::111122223333:ExampleALB123456890",
            "name": "TestResource",
            "availabilityZones": [
                "usw2-az1",
                "usw2-az2",
                "usw2-az3"
            ]
        }
    ]
}
```

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": [
        {
            "zonalShiftId": "2222222-3333-444-1111",
            "resourceIdentifier":
 "arn:aws:testservice::111122223333:ExampleALB123456890",
            "awayFrom": "usw2-az1",
            "expiryTime": 2022-11-15T09:10:42+00:00,
            "startTime": 2022-11-13T01:35:42+00:00,
            "status": "ACTIVE",
            "comment": "Shifting traffic away from USW2-AZ1"
        }
    ]
}
```

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="="arn:aws:testservice::111122223333:ExampleALB123456890" \
   --expires-in="1h" \
   --comment="Still shifting traffic away from USW2-AZ1"
```

```
{
    "zonalShiftId": "2222222-3333-444-1111",
    "resourceIdentifier": "arn:aws:testservice::111122223333:ExampleALB123456890",
    "awayFrom": "usw2-az1",
    "expiryTime": 2022-11-15T10:35:42+00:00,
    "startTime": 2022-11-15T09:35:42+00:00,
    "status": "ACTIVE",
    "comment": "Still shifting traffic away from USW2-AZ1"
}
```

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="="arn:aws:testservice::111122223333:ExampleALB123456890"
```

```
{
    "zonalShiftId": "2222222-3333-444-1111",
    "resourceIdentifier": "arn:aws:testservice::111122223333:ExampleALB123456890",
    "awayFrom": "usw2-az1",
    "expiryTime": 2022-11-15T10:35:42+00:00,
    "startTime": 2022-11-15T09:35:42+00:00,
    "status": "CANCELED",
    "comment": "Shifting traffic away from USW2-AZ1"
}
```

ゾーンシフトおよびゾーンオートシフトでサポートされているリソース

Amazon Route 53 Application Recovery Controller は現在、ゾーンシフトとゾーンオートシフトの次のリソースをサポートしています。

- Network Load Balancers
- アプリケーション ロード バランサー

サポートされている負荷分散リソースは Route 53 ARC に自動的に登録されるため、ゾーンシフト (およびゾーンオートシフト) で使用できます。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 Route 53 ARC で開始できます。

Route 53 ARC でゾーンシフトとリソースを使用する場合は、以下の条件を確認してください。

- ゾーンシフトは、クロスゾーン負荷分散ではサポートされていません。ロードバランサーを Route 53 ARC に登録するには、Elastic Load Balancing でロードバランサーのクロスゾーンロードバランシングをオフにしていることを確認してください。
- いくつかの特定のシナリオでは、ゾーンシフトは AZ からのトラフィックをシフトしません。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。

- パブリックと内部 (プライベート) 両方の、Network Load Balancer および Application Load Balancer がサポートされています。
- トラフィックをそのリソースにシフトするには、リソースがアクティブになっており、正常にプロビジョニングされている必要があります。リソースのゾーンシフトを開始するときは、事前に、そのリソースが Route 53 ARC のマネージドリソースであることを確認します。例えば、 でマネージドリソースのリストを表示したり 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たりできます。
- AWS Global Acceleratorにあるアクセラレーターのエンドポイントである Application Load Balancer では、ゾーンシフトはサポートされていません。
- Application Load Balancer が Network Load Balancer のターゲットである場合は、ゾーンシフトは Network Load Balancer から開始します。Application Load Balancer からゾーンシフトを開始すると、Network Load Balancer は Application Load Balancer とそのターゲットにトラフィックを送信し続けます。
- ゾーンシフトのリソースは、 AWS サービスによって Route 53 ARC に登録されたマネージドリソースである必要があります。Elastic Load Balancing は、クロスゾーン負荷分散がオフになっている Route 53 ARC Network Load Balancer と Application Load Balancer に自動的に登録されます。
- 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。ゾーンシフトが、シフトの対象となる AZ と同じリージョンで開始すること、および、トラフィックのシフト先となるリソースも、同じ AZ とリージョンにあることを、確認します。
- リソースでゾーンシフトを実行するときは、適切な IAM アクセス許可があることを確認します。詳細については、「ゾーンシフトの IAM とアクセス許可」を参照してください。

ゾーンシフトの開始、更新、またはキャンセル

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

ゾーンシフトの開始

このセクションでは、Amazon Route 53 Application Recovery Controller のコンソールで顧客開始のゾーンシフトを開始する手順について説明します。ゾーンシフトをプログラムで操作する方法については、「Zonal Shift API Reference Guide」を参照してください。

Route 53 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 Route 53 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. トラフィックを移動させたいアベイラビリティーゾーンを選択します。
5. [リソース] テーブルで、トラフィックを切り離すロードバランサーを選択します。
6. [ゾーンシフトの有効期限を設定] で、ゾーンシフトの有効期限を選択または入力します。ゾーンシフトは、最初は 1 分 ~ 3 日 (72 時間) まで設定できます。

   すべてのゾーンシフトは一時的なものです。有効期限は必ず設定しますが、アクティブなシフトは、後から新しい有効期限 (最大 3 日後) に更新できます。
7. コメントを入力します。必要に応じて、後でゾーンシフトを更新してコメントを編集できます。
8. このチェックボックスをオンにすると、ゾーンシフトを開始した際、トラフィックがアベイラビリティーゾーンからシフトし、アプリケーションの容量が減ることを了承します。
9. [開始] を選択します。

ゾーンシフトの更新またはキャンセル

このセクションでは、Amazon Route 53 Application Recovery Controller のコンソールで、開始するゾーンシフトを更新する方法、またはゾーンシフトをキャンセルする手順について説明します。ゾーンシフトをプログラムで操作する方法については、「Zonal Shift API Reference Guide」を参照してください。

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

開始したゾーンシフト、またはゾーンオートシフトの練習実行のリソースに対して AWS 開始し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 Route 53 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 Route 53 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 Route 53 Application Recovery Controller でのゾーンシフトのログ記録とモニタリング

AWS CloudTrail と Amazon を使用して、Amazon Route 53 Application Recovery Controller のゾーンシフトを EventBridge 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
- Amazon でのゾーンシフトの使用 EventBridge

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

Amazon Route 53 Application Recovery Controller のゾーンシフトは AWS CloudTrail、Route 53 ARC のユーザー、ロール、または 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。 は、ゾーンシフトのすべての API コールをイベントとして CloudTrail キャプチャします。キャプチャされる呼び出しには、Route 53 ARC コンソールからの呼び出しと、ゾーンシフトのための Route 53 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 CloudTrail イベントなど、Amazon S3 バケットへの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、コンソールのイベント履歴 で最新の CloudTrail イベントを表示できます。

で収集された情報を使用して CloudTrail、ゾーンシフトの Route 53 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。

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

のゾーンシフト情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 は で有効になります。ゾーンシフトのアクティビティが Route 53 ARC で発生すると、そのアクティビティは CloudTrail イベント履歴 の他の AWS サービスイベントとともに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 「イベント履歴の使用」を参照してください。

Route 53 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。証跡により、 はログファイル CloudTrail 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、 の他の AWS サービスを設定して、 CloudTrail ログで収集されたデータをより詳細に分析し、それに基づく対応を行うことができます。詳細については、次を参照してください:

- 「証跡作成の概要」
- CloudTrail でサポートされているサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

すべての Route 53 ARC アクションは によってログに記録 CloudTrail され、「Amazon Route 53 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、 および ListManagedResourcesアクションを呼び出すStartZonalShift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) ユーザーの認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって送信されたかどうか。

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

イベント履歴での Route 53 ARC イベントの表示

CloudTrail では、イベント履歴 で最近のイベントを表示できます。詳細については、「 ユーザーガイド」の CloudTrail 「イベント履歴の使用AWS CloudTrail 」を参照してください。

ゾーンシフトのログファイルエントリについて

証跡は、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できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。 CloudTrail ログファイルは、パブリック API コール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"
      }
    }
```

次の例は、ゾーンシフトの競合例外がある StartZonalShift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": "2022-11-14T16:10:38Z",
      "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"
      }
    }
```

Amazon でのゾーンシフトの使用 EventBridge

Amazon を使用すると EventBridge、ゾーンシフトリソースをモニタリングし、他の AWS の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、ゾーンシフトの開始時に Amazon SNS トピックにシグナルを送ることで、E メール通知を送信するルールを設定できます。

Amazon でルールを作成して EventBridge 、ゾーンシフトに対応できます。ゾーンシフトのイベントは、ゾーンシフトに関するステータス情報を指定します。例えば、ゾーンシフトを開始するとイベントが作成されます。

関心のある特定のゾーンシフトイベントをキャプチャするには、 EventBridge がイベントの検出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、Route 53 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と連携する方法については、「 のイベントとイベントパターン EventBridge」を参照してください。

でゾーンシフトリソースをモニタリングする EventBridge

では EventBridge、Route 53 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。例えば、ゾーンシフトの開始時に E メールメッセージを送信するルールを作成できます。

イベントパターンを入力またはコピーして EventBridge コンソールに貼り付けるには、コンソールで独自のオプションを入力する を使用するオプションを選択します。このトピックでは、役立つ可能性のあるイベントパターンを決定するのに役立つように、ゾーンシフトイベントマッチング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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
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の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] をクリックします。
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。

Route 53 ARC 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。

- Route 53 ARC ゾーンシフト からすべてのイベントを選択します。

```
{
    "source": [
        "aws.arc-zonal-shift"
    ]
}
```

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。で使用可能なターゲットのリストについては EventBridge、「 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、 によって EventBridge 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 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。 CloudWatch

コンソール外の CloudWatch オペレーションを使用して、 CloudWatch ロググループを作成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ルールに EventBridge 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、 API CloudWatch PutResourcePolicy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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。

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。

   Route 53 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CloudWatchとして 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。

Amazon Route 53 Application Recovery Controller でのゾーンシフトのための 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、誰を認証し (サインインさせ)、誰に Route 53 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 AWS のサービス 使用できる です。

内容

ゾーンシフトと IAM の連携方法

IAM を使用して Amazon Route 53 Application Recovery Controller 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 機能について学びます。

ゾーンシフトで使用できる IAM の機能

IAM 機能	ゾーンシフトのサポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes

IAM 機能	ゾーンシフトのサポート
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	No
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	Yes
プリンシパル権限	Yes
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。

Route 53 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ポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限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 ポリシーの作成」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。

Route 53 ARC のアイデンティティベースのポリシー例については、「Amazon Route 53 Application Recovery Controller 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。

Route 53 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	No

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に対するサポート	はい

管理者は AWS JSON 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んなリソースにどんな条件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、関連する 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権限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権限を付与するポリシーで使用されます。

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

ゾーンシフトの Route 53 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。例えば、次のようになります。

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

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

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

ゾーンシフトの Route 53 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 Route 53 Application Recovery Controller 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

ポリシーリソースに対するサポート	はい

管理者は AWS JSON 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どのような条件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、リソースレベルの権限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「サービス認証リファレンス」の次のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義されるアクション

条件キーで使用できるアクションとリソースを確認するには、「サービス認証リファレンス」の次のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

ゾーンシフトの Route 53 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 Route 53 Application Recovery Controller 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	はい

管理者は AWS JSON 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んなリソースにどんな条件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することで、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 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS 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演算を使用して条件 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、「サービス認証リファレンス」の次のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

条件キーで使用できるアクションとリソースについては、「サービス認可リファレンス」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
- Amazon Route 53 - ゾーンシフトで定義されるリソースタイプ

ゾーンシフトの Route 53 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 Route 53 Application Recovery Controller 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。

Route 53 ARC のアクセスコントロールリスト (ACL)

ACL のサポート	No

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権限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。

Route 53 ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート	部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限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操作を許可するように ABAC ポリシーを設計します。

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はいです。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 とは?」を参照してください。ABAC をセットアップするステップ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「属性に基づく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。

Route 53 ARC には、ABAC に対する以下の部分的なサポートが含まれています。

- ゾーンシフトは、ゾーンシフト用に Route 53 ARC に登録されている、マネージドリソースの ABAC をサポートしています。Network Load Balancer と Application Load Balancer マネージドリソースにおける ABAC の詳細については、「Elastic Load Balancing ユーザーガイド」の「Elastic Load Balancing での ABAC」を参照してください。

Route 53 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	はい

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。一時的な認証情報を AWS のサービス 使用できる 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は、一時的な認証情報を使用しています。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスは自動的に一時的な認証情報を作成します。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「ロールへの切り替え (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、長期的なアクセスキーを使用する代わりに、動的に一時的な認証情報を生成する AWS. AWS recommends にアクセスできます。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。

Route 53 ARC のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	はい

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場合 AWS、ユーザーはプリンシパルと見なされます。ポリシーは、プリンシパルに権限を付与します。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権限が必要です。

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、「サービス認証リファレンス」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

Route 53 ARC のサービスロール

サービスロールのサポート	いいえ

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 のサービスにアクセス許可を委任するロールの作成」を参照してください。

Route 53 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	はい

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権限を表示できますが、編集することはできません。

ゾーンシフトは、サービスにリンクされたロールを使用しません。

ゾーンシフトの IAM とアクセス許可

このセクションでは、Amazon Route 53 Application Recovery Controller のゾーンシフト機能のアクセス許可がどのように機能するかについて、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は、追加情報を提供します。Route 53 ARC の機能が IAM およびアクセス許可と連携する一般的な仕組みについては、概要トピック「」の情報を確認してくださ

いAmazon Route 53 Application Recovery Controller でのゾーンシフトのための Identity and Access Management。

IAM 概要トピックで概説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。

- Route 53 ARC でゾーンシフトを使用するのに必要なアクセス許可があることを確認します。詳細については、「ゾーンシフトコンソールアクセス」と「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。
- Route 53 ARC でアカウント内のマネージドロードバランサーリソースのゾーンシフトを実行するときは、IAM で Elastic Load Balancing のアクセス許可を追加する必要はありません。
- Elastic Load Balancing へのフルアクセスを提供する AWS マネージドポリシーには、ゾーンシフトを操作するためのアクセス許可が含まれています。Elastic Load Balancing アクセスに AWS マネージドポリシーを使用する場合、ロードバランサーのゾーンシフトを開始したり、Elastic Load Balancing コンソールで を操作するために、ゾーンシフトの IAM に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。

Amazon Route 53 Application Recovery Controller 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 Route 53 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受けることができます。

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 ポリシーの作成」を参照してください。

Route 53 ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認可リファレンス」の「Amazon Route 53 Recovery コントロールのアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーのベストプラクティス

- 例: ゾーンシフトコンソールへのアクセス
- 例: ゾーンシフト API アクション

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで Route 53 ARC リソースの作成、アクセス、削除を行える人を決めます。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を付与するには、多くの一般的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して、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「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。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、最小特権権限とも呼ばれています。IAM を使用して権限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 でのポリシーと権限」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、 などの特定の を通じてサービスアクションを使用する場合 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 policy elements: Condition] (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 ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、『IAM ユーザーガイド』の「MFA 保護 API アクセスの設定」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へのアクセス

Amazon Route 53 Application Recovery Controller コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。これらのアクセス許可により、 の Route 53 ARC リソースの詳細をリスト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、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してアプリケーションを復旧します。

ユーザーがゾーンシフト API アクションを使用できるようにするには、次のような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。

```
{
    "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": "*"
        }
    ]
}
```

Amazon Route 53 Application Recovery Controller のゾーンオートシフト

ゾーンオートシフトでは、復旧までの時間を短縮 AWS するために、 が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ユーザーに許可します。 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内部テレメトリによって示されたときにオートシフト 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シフトし始めます。

Route 53 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、影響のないリソースに対してトラフィックが遠ざけることがあります。

ゾーンオートシフトでは、定期的な練習実行のために、 AWS が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ことも許可します。ゾーンオートシフトには練習実行が必要です。Route 53 ARC が練習実行として開始するゾーンシフトは、オートシフト中にアベイラビリティーゾーンからトラフィックをシフトすることがアプリケーションにとって安全であることを保証するのに役立ちます。練習実行では、リソースのトラフィックをアベイラビリティーゾーンから遠ざけるゾーンシフトを開始することによって、1 つのアベイラビリティーゾーンがなくてもアプリケーションが正常に動作することを定期的にテストします。練習実行は毎週行われ、アプリケーションが期待どおりに動作するかどうか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リング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したり、練習実行を設定したりする予定の場合は、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。

ゾーンシフトによる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。Application Load Balancer と Network Load Balancer はどちらも、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。サービスおよび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。

ゾーンシフトを開始して評価することで、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し続けることができることを確認した後、Route 53 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。

Route 53 ARC コンソールで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だけでなく、Amazon EC2 コンソールで特定のロードバランサーのゾーンオートシフトを有効にすることもできます。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。 Elastic Load Balancing

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、影響を受けたアベイラビリティーゾーンが回復すると、 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のを 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。

オートシフトと練習実行について警告するように Amazon EventBridge 通知を設定できます。詳細については、「Amazon でのゾーン自動シフトの使用 EventBridge」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 Route 53 Application Recovery Controller 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、 内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、Route 53 ARC がリソースのトラフィックを定期的に 1 つのアベイラビリティーゾーンから遠ざけるようにする練習実行を設定する必要があります。Route 53 ARC は、練習実行設定が関連付けられているリソースにつき、約 1 週間ごとに練習実行をスケジュールします。各リソースの練習実行は個別にスケジュールされます。

各練習実行について、Route 53 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。

オートシフトと練習実行に関する情報を送信するように Amazon EventBridge 通知を設定できます。詳細については、「Amazon でのゾーン自動シフトの使用 EventBridge」を参照してください。

トピック

- がオートシフト AWS を開始および停止するとき
- Route 53 ARC が練習実行をスケジュール、開始、および終了するとき
- ゾーンシフト、練習実行、およびオートシフトの優先順位
- 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
- トラフィックの移動方法
- 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帯 (UTC)

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、復旧までの時間を短縮 AWS するために、 が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ユーザーに代わって許可します。

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すると、オートシフトを終了します。

Route 53 ARC が練習実行をスケジュール、開始、および終了するとき

Route 53 ARC は、1 つのリソースについて、毎週約 30 分間の練習実行をスケジュールします。Route 53 ARC は、各リソースの練習実行を個別にスケジュール、開始、および管理します。Route 53 ARC は、同じアカウントの複数のリソースの練習実行をまとめることはありません。

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。

Route 53 ARC が練習実行を中断して終了するシナリオがいくつかあります。例えば、練習実行中にオートシフトが開始した場合、Route 53 ARC は練習実行を中断して終了します。別の例として、練習実行に対してリソースが不利な反応を示し、練習実行を監視するために指定したアラームが ALARM 状態になったとします。このシナリオでも、Route 53 ARC は練習実行を中断して終了します。

さらに、Route 53 ARC がリソースについてスケジュールされた練習実行を開始しないシナリオもいくつかあります。

リソースの練習実行が中断またはブロックされた場合に対応して、Route 53 ARC は次のことを行います。

- リソースの練習実行が進行中に中断された場合、Route 53 ARC は毎週の練習実行が終了したと見なし、そのリソースの新しい練習実行を次の週にスケジュールします。このシナリオでは、毎週の練習の結果は FAILED ではなく INTERRUPTED です。練習実行の結果が FAILED に設定されるのは、練習実行を監視する結果アラームが練習実行中に ALARM 状態になった場合のみです。
- リソースの練習実行の開始が予定されているときにブロッキング制約がある場合、Route 53 ARC は練習実行を開始しません。Route 53 ARC は引き続き定期的な監視を行い、ブロッキング制約が 1 つ以上あるかどうかを判断します。ブロッキング制約がない場合、Route 53 ARC はリソースの練習実行を開始します。

以下は、Route 53 ARC がリソースの練習実行を開始または続行することを禁止するブロック制約の例です。

- 進行中の AWS Fault Injection Service 実験がある場合、Route 53 ARC は練習実行を開始または続行しません。Route 53 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になっている場合、Route 53 ARC は練習実行を開始しません。Route 53 ARC は、練習実行全体を通してイベントを含むブロック制約を監視します AWS FIS 。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、Route 53 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ている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、Route 53 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、Route 53 ARC は通常どおり 1 週間後に次の練習実行をスケジュールします。指定した AWS FIS 実験やブロックされた時間枠などのブロック制約が原因で練習実行が開始されない場合、Route 53 ARC は練習実行が開始されるまで練習実行を開始しようとし続けます。

ゾーンシフト、練習実行、およびオートシフトの優先順位

1 つのリソースに対して有効なトラフィックシフトは一度に 1 つだけです。つまり、そのリソースに対して 1 つの練習実行のゾーンシフト、顧客主導のゾーンシフト、またはオートシフトのみを設定できます。進行中のトラフィックシフトが複数ある場合、Route 53 ARC は優先順位に従って、どのトラフィックシフトがリソースについて有効かを決定します。

優先順位の全体的な原則は、顧客として開始するゾーンシフトが、練習実行よりも優先されるオートシフトよりも優先されることです。つまり、顧客が開始したゾーンシフト > オートシフト > 練習実行のゾーンシフトです。

これを説明するために、いくつかのシナリオ例における優先順位の仕組みを以下に示します。

- アクティブなオートシフトがあるときに、オートシフトが有効になっているリソースのゾーンシフトを開始した場合、開始するゾーンシフトは APPLIED になります。これで、リソースはゾーンシフトが適用されるアベイラビリティーゾーンから移動されます。 AWS がオートシフトを終了する前にゾーンシフトが終了した場合、オートシフトは APPLIED シフトになります。したがって、リソースは、 が AWS オートシフトを進行中のアベイラビリティーゾーンから遠ざけられます。
- オートシフトが有効になっているリソースに対してアクティブなゾーンシフトを開始し、オートシフト AWS を開始すると、そのリソースにオートシフトが存在します。ただし、ゾーンシフトは APPLIED に設定され、ゾーンシフトが終了するまでオートシフトは NOT APPLIED に設定されます。次に、オートシフトのステータスが に更新APPLIEDされ、オートシフト AWS が終了するまで、オートシフトはリソースのトラフィックを遠ざけます。
- あるリソースについてアクティブな練習実行があり、同じアベイラビリティーゾーンについてリソースのトラフィックを遠ざけるゾーンシフトを開始した場合、練習実行は中断されます。トラフィックを別のアベイラビリティーゾーンから遠ざけるゾーンシフトを開始した場合、練習実行は通常どおり続行されます。
- リソースのアクティブなゾーンシフトがあり、Route 53 ARC が練習実行を開始する予定である場合、練習実行は 1 時間延期されます。その後、Route 53 ARC は再び練習実行の開始を試みます。Route 53 ARC は、練習実行を開始できるまで、1 時間ごとにチェックを続けます。

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

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、そのリソースのゾーンオートシフトを無効にします。

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

練習実行設定を削除すると、 はリソースのトラフィックを毎週アベイラビリティーゾーンから遠ざける練習実行の実行を 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、Route 53 ARC コンソールを使用して練習実行設定を削除すると、このアクションによりリソースのゾーンオートシフトも無効になります。ただし、ゾーンオートシフト API を使用して練習実行を削除する場合は、まずリソースのゾーンオートシフトを無効にする必要があることに注意してください。

アクティブな練習実行を停止するには、練習実行のゾーンシフトをキャンセルします。詳細については、「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。

トラフィックを遠ざける方法

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

オートシフトでは、アベイラビリティーゾーンが回復してオートシフトを終了する AWS と、Route 53 ARC はヘルスチェックプロセスを逆にして、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、そのアベイラビリティーゾーンはロードバランサー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。Route 53 ARC は、ヘルスチェックを「異常」に設定し、オートシフトまたはゾーンシフトを終了したときにヘルスチェックを再び「正常」に戻すようにリクエストすることにより、ヘルスチェックを使用してトラフィックをアベイラビリティーゾーンから遠ざけます。

練習実行のアラーム

ゾーンオートシフトでの練習実行には 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。

練習実行を有効にするには、結果アラームとして、リソースまたはアプリケーションのメトリクスをモニタリングする CloudWatch アラームとして を指定します。このアラームは、アプリケーションが 1 つのアベイラビリティーゾーンの喪失によって悪影響を受けた場合に ALARM状態で応答します。詳細については、「ゾーンオートシフトを設定する際のベストプラクティス」の「練習実行について指定するアラーム」セクションを参照してください。

結果アラームには、Route 53 ARC が各練習実行について報告する練習実行結果の情報も表示されます。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)

特定の歴日、または特定の時間枠 (つまり UTC の日時) について練習実行をブロック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)。

ゾーンオートシフトについて

ゾーンオートシフトは、 がユーザーに代わって AWS 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機能です。 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内部テレメトリによって示されると、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。

クロスゾーン負荷分散がオフになっている Network Load Balancer と Application Load Balancer についてゾーンオートシフトを有効にできます。

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遠ざけることで、AZ 内のカスタマー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 にシフトすることで、 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要度を減らす AWS ことができます。

がロードバランシングリソースのオートシフト AWS を開始すると、Route 53 ARC はロードバランサーリソースの対応する IP アドレスの Amazon Route 53 ヘルスチェックを異常に設定し、リソースのトラフィックが AZ に送信されなくなります。 AWS が AZ がアプリケーショントラフィックを返す準備ができていると判断すると、Route 53 ARC は Route 53 ヘルスチェックを復元し、元のゾーン IP アドレスが復元されます。

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、「Amazon Route 53 Application Recovery Controller のゾーンオートシフト」を参照してください。

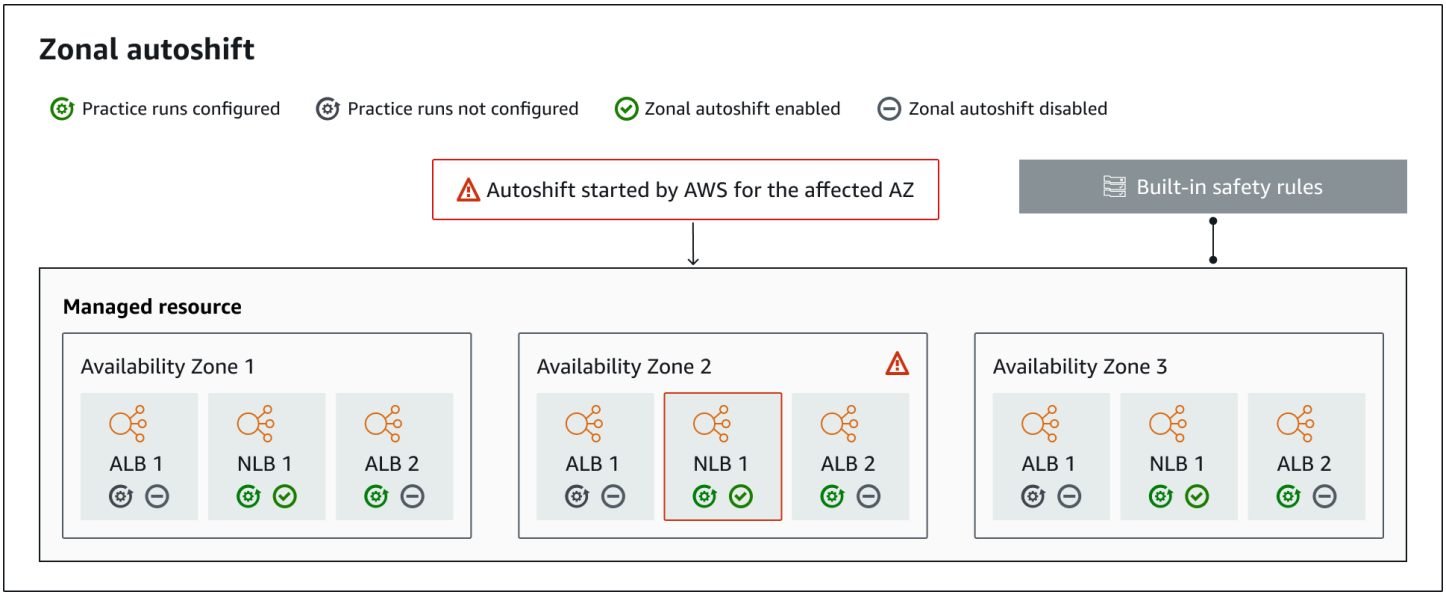
AWS リージョン ゾーンオートシフトの可用性

ゾーンオートシフトは現在、商用 で利用可能です AWS リージョン。

Amazon Route 53 Application Recovery Controller のリージョンサポートとリージョンサービスエンドポイントに関する詳細は、Amazon Web Services 全般のリファレンスにある「Amazon Route 53 Application Recovery Controller のエンドポイントとクォータ」を参照してください。

ゾーンオートシフトのコンポーネント

次の図は、トラフィックをアベイラビリティーゾーンから遠ざけるオートシフトの例を示しています。 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内部テレメトリによって示されたときにオートシフト AWS を開始します。



以下は、Route 53 ARC におけるゾーンオートシフト機能のコンポーネントです。

ゾーンオートシフト

ゾーンオートシフトは、何も操作しなくても、リソースのトラフィックを遠ざけます。ゾーンオートシフトは Route 53 ARC の機能で、 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内部テレメトリによって示されたときにオートシフト AWS を開始します。場合によっては、影響が及んでいないリソース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、約 30 分間、毎週練習実行のゾーンシフト AWS を実行します。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 シフトで 1 つのアベイラビリティーゾーンから遠ざけ、練習実行が終了したらトラフィックを元に戻します。

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 CloudWatchゾーンオートシフトのリソースの練習実行に指定したアラームを定義します。練習実行はいつでも編集でき、ブロックされる日付や時間枠を追加または変更したり、練習実行のアラームを更新したりできます。

ゾーンオートシフトを有効にするには、リソース用の練習実行設定が必要です。練習実行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、Route 53 ARC は練習実行のゾーンシフトを終了します。これにより、リソースのトラフィックはアベイラビリティーゾーンから遠ざけられなくなります。

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

Route 53 ARC は、各練習実行の結果を報告します。可能な練習実行の結果は以下のとおりです。

- PENDING: 練習実行のゾーンシフトはアクティブ (進行中) です。まだ結果は戻されていません。
- SUCCEEDED: 練習実行中、結果アラームは ALARM 状態にならず、練習実行は 30 分間のテスト期間をすべて完了しました。
- INTERRUPTED: 結果アラームが ALARM 状態になったのではない理由で、練習実行は終了しました。練習実行は、以下のようなさまざまな理由で中断されることがあります。例えば、練習実行について指定されたブロッキングアラームが ALARM 状態になったために終了した練習走行は、INTERRUPTED の結果になります。INTERRUPTED 結果の理由の詳細については、「練習実行の結果」を参照してください。

- FAILED: 練習実行中に結果アラームが ALARM 状態になりました。

組み込みの安全ルール

Route 53 ARC に組み込まれている安全ルールにより、1 つのリソースについて一度に複数のトラフィックシフトが行われることはありません。つまり、アベイラビリティーゾーンからトラフィックをアクティブにシフトできるのは、そのリソースについて、顧客によって開始されたゾーンシフト、練習実行のゾーンシフト、またはオートシフトの 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、「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を有効にできるのは、Route 53 ARC でサポートされている サービス内の AWS アカウント内のリソースのみです。これらの AWS サービスでサポートされているリソースは、 AWS サービスによって Route 53 ARC に自動的に登録されます。

> **Note**
> クロスゾーン負荷分散がオフになっている Network Load Balancer と Application Load Balancer に対してのみゾーンオートシフトを設定できます。

マネージドリソース

AWS サービスは、ゾーンオートシフト用にリソースを Route 53 ARC に自動的に登録します。登録されたリソースは Route 53 ARC のマネージドリソースとなります。

リソース名

Route 53 ARC のマネージドリソースの名前です。

適用ステータス

適用ステータスは、リソースに対してトラフィックシフトが適用されているかどうかを示します。ゾーンオートシフトを設定すると、1 つのリソースに複数のアクティブなトラフィックシフト、つまり、練習実行ゾーンシフト、顧客によって開始されたゾーンシフト、またはオートシフトが発生する可能性があります。ただし、リソースに適用される、つまり有効になるのは一度に 1 つだけです。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。

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、信頼性と耐障害性を最大限に高めるには、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 が高可用性の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 Route 53 Application Recovery Controller 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、カスタマー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。

Amazon Route 53 Application Recovery Controller で使用した分に対してのみお支払いいただきます。Route 53 ARC の料金情報と料金例の詳細については、「Amazon Route 53 の料金」を参照し、Amazon Route 53 Application Recovery Controller までスクロールします。

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 Route 53 Application Recovery Controller 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、Route 53 ARC がユーザーに代わってゾーンシフトを開始します。ゾーンシフトは、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遠ざけ、毎週の頻度で再びトラフィックをシフトします。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留意すべきベストプラクティスと考慮事項がいくつかあります。ゾーンオートシフトを有効にしたり、リソースの練習実行を設定したりする前に、以下のトピックを確認してください。

トピック

クライアントがエンドポイントに接続し続ける時間を制限する

Amazon Route 53 Application Recovery Controller 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、Route 53 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 を更新すると、すべての新しい接続が障害が発生した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引き続きリクエストを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を与える可能性があり、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移動することによるトレードオフであると見なしてください。

Application Load Balancer の keepalive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。Application Load Balancer

リソース容量をプリスケーリングし、トラフィックのシフトをテストする

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増加したリクエストレート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 AWS 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でも、2 つのアベイラビリティーゾーンにまたがる残りの 30 個のインスタンスをアプリケーションのクライアントに提供AWS できます。

Route 53 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。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、「Amazon Route 53 Application Recovery Controller 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。一般に、クロスゾーン負荷分散がオフになっている Network Load Balancer と Application Load Balancer はサポートされ

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。このシナリオで が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、オートシフトは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を変更しません。これは想定される動作です。オートシフトでは、すべてのアベイラビリティーゾーンがオープンに失敗 (異常) AWS リージョン した場合、1 つのアベイラビリティーゾーンが異常になり、 内の他のアベイラビリティーゾーンにトラフィックをシフトすることはできません。

2 つ目のシナリオは、 でアクセラレーターのエンドポイントである Application Load Balancer のオートシフトを が AWS 開始する場合です AWS Global Accelerator。ゾーンシフトと同様、オートシフトは Global Accelerator にあるアクセラレーターのエンドポイントである Application Load Balancer についてはサポートされていません。

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、「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照してください。

練習実行のアラームを指定する

ゾーンオートシフトによる練習実行では、少なくとも 1 つのアラーム、つまり結果アラームを設定します。オプションで、2 つ目のアラームである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。

- 必要な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、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とパフォーマンスに悪影響を与えることを示している場合に、 ALARM状態になるように CloudWatch アラームを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。
- AWS Well Architected Framework に従うことをお勧めします。このフレームワークでは、アラームとして CloudWatch主要業績評価指標 (KPIsを実装することをお勧めします。その場合、これらのアラームを使用して、安全トリガーとして使用する複合アラームを作成し、アプリケーションが KPI を見逃す可能性がある場合には練習実行が開始されないようにすることがで

きます。アラームが ALARM 状態でなくなると、Route 53 ARC はそのリソースに対して次回の練習実行がスケジュールされている時点で練習実行を開始します。

- 練習実行ブロッキングアラームを設定することにした場合は、練習実行を開始したく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ができます。
- 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。これは、まず Amazon で設定する必要があります CloudWatch。指定する CloudWatch アラームは複合アラームにすることができ、アラームが ALARM状態になるよう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 ユーザーガイド」の「アラームの組み合わせ CloudWatch 」を参照してください。
- 練習実行に指定する CloudWatch アラームが、練習実行を設定するリソースと同じリージョンにあることを確認します。

練習実行の結果を評価する

Route 53 ARC は、各練習実行の結果を報告します。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、容量をスケーリングしたり、アラームの設定を調整する必要がある場合があります。

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。
  - 練習実行設定に指定された CloudWatch アラームにアクセスできなくなるため、練習実行が終了しました。
  - 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。
- 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 Route 53 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	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
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	CreatePracticeRunConfiguration「��
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	DeletePracticeRunConfiguration「��
オートシフトを一覧表示する	「Amazon Route 53 Application Recovery Controller のゾーンオートシフト」を参照	ListAutoshifts「��
ゾーンオートシフト用のリソースを一覧表示する	「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照	ListManaged「 リソース」
ゾーンオートシフト用のリソースを取得する	「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照	GetManaged「 リソース」
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	UpdatePracticeRunConfiguration「��
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	UpdateZonalAutoshiftConfiguration「��

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 Amazon Route 53 Application Recovery Controller のゾーンオートシフト機能を API オペレーションを使用して操作 AWS Command Line Interface 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 Route 53 ARC の機能です。ゾーンオートシフトを使用すると、 AWS が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。ゾーンオートシフトには練習実行が含まれており、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、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継続的に検証できます。

ゾーンオートシフトは、現在、クロスゾーン負荷分散がオフになっている Network Load Balancer と Application Load Balancer をサポートしています。

詳細については、「ゾーンシフトおよびゾーンオートシフトでサポートされているリソース」を参照してください。

このセクションでは、ゾーンオートシフトの始め方と操作方法を説明するために、以下の例を紹介します。

- リソースの練習実行設定を作成します。
- リソースのオートシフトを有効または無効にします。
- 練習実行によって開始されたゾーンシフトをキャンセルして、進行中の練習実行を終了します。
- リソースのゾーンオートシフト機能を無効にすることによって、進行中のオートシフトを終了します。
- リソースの練習実行設定を編集して、指定したアラーム、ブロックされた日付または時間枠を変更します。
- リソースの練習実行設定を作成します。

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。

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。
- AWS リージョン リソースがデプロイされているのと同じ 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"
}
```

進行中のオートシフトをキャンセルする

リソースの進行中のオートシフトをキャンセルするには、ゾーンオートシフト機能を無効にします。これは、一般にゾーンオートシフトを無効にするために使用するコマンドと同じであるため、ゾーンオートシフトを無効にして進行中のオートシフトをキャンセルしても、リソースは将来のオートシフトの影響を受けません。ゾーンオートシフトを更新して、いつでも再び有効にすることができます。

リソースの練習実行設定を削除しなくても、リソースのゾーンオートシフトを無効にできることに注意してください。

CLI でオートシフトをキャンセルするには、update-zonal-autoshift-configuration 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。例えば、リソースのオートシフトを終了するには、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="DISABLED"
```

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

進行中の練習実行をキャンセルする

CLI で進行中の練習実行をキャンセルするには、その練習実行が開始したリソースのゾーンシフトをキャンセルします。練習実行をキャンセルするには、cancel-zonal-shift コマンドを使用します。

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

```
aws arc-zonal-shift cancel-zonal-shift \
   --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 Started"
}
```

練習実行設定を編集する

Route 53 ARC が練習実行を開始しない場合、CLI を使用してリソースの練習実行設定を編集し、さまざまな設定オプションを更新できます。例えば、練習実行のアラームを変更したり、ブロックされた日付やブロックされた時間枠を更新したりできます。練習実行設定を編集するには、update-practice-run-configuration コマンドを使用します。

リソースの練習実行設定を編集するときには、次の点に注意してください。

- 現時点でサポートされているアラームタイプは CLOUDWATCH のみです。
- AWS リージョン リソースがデプロイされているのと同じ 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、ゾーンオートシフトの有効化と無効化、練習実行の設定、進行中の練習実行のキャンセルなど、Amazon Route 53 Application Recovery Controller でゾーンオートシフトを操作する手順について説明します。

ゾーンオートシフトの有効化または無効化

このセクションでは、Amazon Route 53 Application Recovery Controller のコンソールからゾーンオートシフトを有効化または無効化する手順について説明します。ゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

ゾーンオートシフトが有効になっている場合、復旧までの時間を短縮 AWS するために、 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します。

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 Route 53 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、[ゾーンオートシフトを有効化] または [ゾーンオートシフトを無効化] 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。

練習実行設定の設定、編集、または削除

このセクションでは、Amazon Route 53 Application Recovery Controller のコンソールから練習実行設定を編集または削除する手順について説明します。ゾーンオートシフトをプログラムで操作する方法については、「ゾーンシフトおよびゾーンオートシフト API リファレンスガイド」を参照してください。

コンソールで練習実行設定を削除すると、ゾーンオートシフトは無効になります。API オペレーションで練習実行設定を削除するには、その前に、ゾーンオートシフトを無効にする必要があります。

ゾーンオートシフトを有効にしなくても練習実行を設定できます。ただし、ゾーンオートシフトがリソースについて有効であるためには、そのリソースに対して練習実行を設定してある必要があります。

練習実行を設定するには

1. で Route 53 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。Route 53 ARC がこのリソースの練習実行を開始しないようにする日付または時間枠 (曜日と時刻) 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 Route 53 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。
  - 練習実行をモニタリングする結果アラームには、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 Route 53 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。

練習実行のゾーンシフトのキャンセル

このセクションでは、Amazon Route 53 Application Recovery Controller のコンソールでゾーンシフトをキャンセルする手順について説明します。ゾーンシフト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 Route 53 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 Route 53 Application Recovery Controller 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 を使用して、Amazon Route 53 Application Recovery Controller でゾーンオートシフトを EventBridge 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

- を使用したゾーン自動shift API コールのログ記録 AWS CloudTrail
- Amazon でのゾーン自動シフトの使用 EventBridge

を使用したゾーン自動shift API コールのログ記録 AWS CloudTrail

Amazon Route 53 Application Recovery Controller のゾーン自動シフトは AWS CloudTrail、Route 53 ARC のユーザー、ロール、または 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。 は、ゾーンシフトのすべての API コールをイベントとして CloudTrail キャプチャします。キャプチャされる呼び出しには、Route 53 ARC コンソールからの呼び出しと、ゾーンシフトのための Route 53 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 CloudTrail イベントなど、Amazon S3 バケットへの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。

で収集された情報を使用して CloudTrail、ゾーンシフトの Route 53 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。

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

のゾーン自動シフト情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 は で有効になります。ゾーン自動shift のアクティビティが Route 53 ARC で発生すると、そのアクティビティは CloudTrail イベント履歴 の他の AWS サービスイベントとともに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 「イベント履歴の使用」を参照してください。

Route 53 ARC のゾーン自動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。証跡により、 はログファイル CloudTrail 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、 の他の AWS サービスを設定して、 CloudTrail ログで収集されたデータをより詳細に分析し、それに基づく対応を行うことができます。詳細については、次を参照してください:

- 「証跡作成の概要」
- CloudTrail でサポートされているサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

すべての Route 53 ARC アクションは によってログに記録 CloudTrail され、「Amazon Route 53 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、 および `ListManagedResources`アクションを呼び出す`StartZonalShift`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) ユーザーの認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって送信されたかどうか。

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

イベント履歴での Route 53 ARC イベントの表示

CloudTrail では、イベント履歴 で最近のイベントを表示できます。詳細については、「 ユーザーガイド」の CloudTrail 「イベント履歴の使用AWS CloudTrail 」を参照してください。

ゾーン自動shift ログファイルエントリについて

証跡は、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できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。 CloudTrail ログファイルは、パブリック API コール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、ゾーン自動shift 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 でのゾーン自動シフトの使用 EventBridge

Amazon を使用すると EventBridge、ゾーン自動shift リソースをモニタリングし、他の AWS の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、ゾーン自動shift のプラクティスの実行開始時に Amazon SNS トピックにシグナルを送ることで、E メール通知を送信するルールを設定できます。

Amazon でルールを作成して EventBridge 、ゾーンの自動シフトに対応できます。ゾーン自動shift イベントのイベントは、プラクティス実行が進行中の場合など、プラクティス実行自動shift に関するステータス情報を指定します。

関心のある特定のゾーン自動shift イベントをキャプチャするには、 EventBridge がイベントの検出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、Route 53 ARC から に EventBridge 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と連携する方法については、「 のイベントとイベントパターン EventBridge」を参照してください。

でゾーン自動shift リソースをモニタリングする EventBridge

では EventBridge、Route 53 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。例えば、ゾーン自動shift のプラクティス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。

イベントパターンを入力またはコピーして EventBridge コンソールに貼り付けるには、コンソールで独自のオプションを入力する を使用するオプションを選択します。このトピックでは、役立つ可

能性のあるイベントパターンを決定するのに役立つように、ゾーン自動shift イベントマッチングパターンとゾーン自動shift 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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
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の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] をクリックします。
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。

ゾーン自動shift 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。

このセクションから にイベントパターンをコピーして貼り付け EventBridge ると、ゾーンの自動 shift 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`

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。

- プラクティスの実行が開始されたすべてのイベントをゾーン自動shift から選択します。

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

- プラクティスの実行に失敗したすべてのイベントをゾーン自動shift から選択します。

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

ゾーン自動shift イベントの例

ゾーン自動shift アクションのイベント例を次に示します。

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

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。で使用可能なターゲットのリストについては EventBridge、「 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、 によって EventBridge 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 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。 CloudWatch

コンソール外の CloudWatch オペレーションを使用して、 CloudWatch ロググループを作成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ルールに EventBridge 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、 API CloudWatch PutResourcePolicy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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
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。

   Route 53 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CloudWatchとして 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。

ゾーン自動shift 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、誰を認証し (サインインさせ)、誰に

Route 53 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 AWS のサービス 使用できる です。

内容

Amazon Route 53 Application Recovery Controller のゾーン自動シフトと IAM の連携方法

IAM を使用して Amazon Route 53 Application Recovery Controller でゾーン自動shift へのアクセスを管理する前に、ゾーン自動shift で使用できる IAM 機能について学びます。

Amazon Route 53 Application Recovery Controller のゾーン自動シフトで使用できる IAM の機能

IAM 機能	ゾーン自動シフトのサポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	No
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	Yes
プリンシパル権限	Yes
サービスロール	いいえ

IAM 機能	ゾーン自動シフトのサポート
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。

Route 53 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ポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限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 ポリシーの作成」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。

Route 53 ARC のアイデンティティベースのポリシー例については、「Amazon Route 53 Application Recovery Controller 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。

Route 53 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	No

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。

Route 53 ARC のポリシーアクション

ポリシーアクションに対するサポート	はい

管理者は AWS JSON 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んなリソースにどんな条件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、関連する 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権限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権限を付与するポリシーで使用されます。

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

ゾーン自動shift の Route 53 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。例えば、次のようになります。

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

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

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

ゾーン自動シフトの Route 53 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ゾーン自動shift のアイデンティティベースのポリシーの例。

Route 53 ARC のゾーン自動シフトのポリシーリソース

ポリシーリソースに対するサポート	はい

管理者は AWS JSON 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どのような条件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、リソースレベルの権限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「サービス認証リファレンス」の次のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義されるアクション

条件キーで使用できるアクションとリソースを確認するには、「サービス認証リファレンス」の次のトピックを参照してください。

- Amazon Route 53 - ゾーンシフトで定義される条件キー

ゾーン自動シフトの Route 53 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ゾーン自動shift のアイデンティティベースのポリシーの例。

Route 53 ARC のゾーン自動shift 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	はい

管理者は AWS JSON 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んなリソースにどんな条件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することで、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 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS 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演算を使用して条件 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。

ゾーン自動shift の Route 53 ARC 条件キーのリストを確認するには、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 ゾーンシフトの条件キー

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

ゾーン自動シフトの Route 53 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ゾーン自動shift のアイデンティティベースのポリシーの例。

Route 53 ARC のアクセスコントロールリスト (ACL)

ACL のサポート	No

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権限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。

Route 53 ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート	部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限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操作を許可するように ABAC ポリシーを設計します。

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はいです。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 とは?」を参照してください。ABAC をセットアップするステップ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「属性に基づく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。

Route 53 ARC のゾーン自動シフトには、ABAC に対する以下の部分的なサポートが含まれています。

- ゾーン自動シフトは、ゾーンシフト用に Route 53 ARC に登録されているマネージドリソースの ABAC をサポートしています。Network Load Balancer と Application Load Balancer マネージドリソースにおける ABAC の詳細については、「Elastic Load Balancing ユーザーガイド」の「Elastic Load Balancing での ABAC」を参照してください。

Route 53 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	はい

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。一時的な認証情報を AWS のサービス 使用できる 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は、一時的な認証情報を使用しています。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスは自動的に一時的な認証情報を作成します。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「ロールへの切り替え (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、長期的なアクセスキーを使用する代わりに、動的に一時的な認証情報を生成する AWS. AWS recommends にアクセスできます。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。

Route 53 ARC のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	はい

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場合 AWS、ユーザーはプリンシパルと見なされます。ポリシーは、プリンシパルに権限を付与します。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権限が必要です。

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、「サービス認証リファレンス」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

Route 53 ARC のサービスロール

サービスロールのサポート	いいえ

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 のサービスにアクセス許可を委任するロールの作成」を参照してください。

Route 53 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	はい

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権限を表示できますが、編集することはできません。

Route 53 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、「Route 53 ARC でのゾーン自動シフトのサービスにリンクされたロールの使用」を参照してください。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の中から、[Service-linked role] (サービスにリンクされたロール) 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、[Yes] リンクを選択します。

ゾーン自動shift 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 Route 53 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受けることができます。

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 ポリシーの作成」を参照してください。

Route 53 ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認可リファレンス」の「Amazon Route 53 Recovery コントロールのアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーのベストプラクティス

- 例: ゾーン自動shift コンソールへのアクセス
- 例: Route 53 ARC API アクション

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで Route 53 ARC リソースの作成、アクセス、削除を行える人を決めます。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 管理ポリシーの使用を開始し、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を付与するには、多くの一般的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して、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「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。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、最小特権権限とも呼ばれています。IAM を使用して権限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 でのポリシーと権限」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、 などの特定の を通じてサービスアクションを使用する場合 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 policy elements: Condition] (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 ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、『IAM ユーザーガイド』の「MFA 保護 API アクセスの設定」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: ゾーン自動shift コンソールへのアクセス

Amazon Route 53 Application Recovery Controller コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。これらのアクセス許可により、 の Route 53 ARC リソースの詳細をリスト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、Route 53 ARC のゾーン自動シフト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、「Route 53 ARC でのゾーン自動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。

でゾーン自動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: Route 53 ARC API アクション

ポリシーを使用して、ユーザーがゾーン自動shift の Route 53 ARC API アクションを使用してゾーン自動shift を設定できるようにすることで、 がユーザーに代わってアプリケーション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 内の正常な AZs に AWS シフトし AWS リージョン、イベント中の復旧までの時間を短縮できます。これらのアクセス許可を付与するには、以下で説明するように、ユーザーが操作する必要がある API オペレーションに対応するポリシーをアタッチします。

一部のタスクを実行するには、ユーザーは Route 53 ARC に関連付けられたサービスにリンクされたロールのアクセス許可を持っている必要があります。サービスにリンクされたロールを作成するために必要なアクセス許可は、次のポリシーの例に含まれています。詳細については、「Route 53 ARC でのゾーン自動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。

ゾーン自動shift 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" : "*"
        }
    ]
}
```

Route 53 ARC でのゾーン自動シフトのサービスにリンクされたロールの使用

Amazon Route 53 Application Recovery Controller のゾーン自動シフトは、 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。サービスリンクロールは、サービス (この場合は Route 53 ARC) に直接リンクされる一意なタイプの IAM ロールです。サービスにリンクされたロールは、Route 53 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 の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ことで、必要なアクセス権限を手動で追加する必要がなくなるため、Route 53 ARC の設定が簡単になります。Route 53 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、Route 53 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することが防止され、Route 53 ARC ゾーン自動shift 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している他のサービスについては、「IAM と連携する AWS のサービス」で「サービスにリンクされたロール」列が「はい」になってい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。

のサービスにリンクされたロールのアクセス許可 AWSServiceRoleForZonalAutoshiftPracticeRun

Route 53 ARC は、 という名前のサービスにリンクされたロールAWSServiceRoleForZonalAutoshiftPracticeRunを使用して以下を実行します。

- 顧客が提供する 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、プラクティスを実行する
- 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。

のサービスにリンクされたロールのアクセス許可 AWSServiceRoleForZonalAutoshiftPracticeRun

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 サービスにリンクされたロールは、次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。

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) 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 ユーザーガイド」の「サービスにリンクされたロールのアクセス許可」を参照してください。

Route 53 AWSServiceRoleForZonalAutoshiftPracticeRun ARC のサービスにリンクされたロールの作成

AWSServiceRoleForZonalAutoshiftPracticeRun サービスにリンクされた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 で最初のプラクティス実行設定を作成すると、Route 53 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きには、Route 53 ARC がユーザーのためにサービスリンクロールを再び作成します。

Route 53 ARC AWSServiceRoleForZonalAutoshiftPracticeRunのサービスにリンクされたロールの編集

Route 53 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 を使用したロール記述の編集はできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Route 53 ARC AWSServiceRoleForZonalAutoshiftPracticeRunの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が必要な機能またはサービスが不要になった場合には、その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。

自動シフトを無効にしたら、AWSServiceRoleForZonalAutoshiftPracticeRunサービスにリンクされた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、「Amazon Route 53 Application Recovery Controller のゾーンシフト」を参照してください。

> **Note**
>
> リソースの削除を試みたときに、Route 53 ARC のサービスがロールを使用していた場合、サービスロールの削除が失敗することがあります。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。

サービスにリンクされたロールを IAM で手動削除するには

IAM コンソール、、または AWS API を使用して AWS CLI、サービスにリンクされたロールを削除します AWSServiceRoleForZonalAutoshiftPracticeRun。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。

ゾーン自動shift の Route 53 ARC サービスにリンクされたロールの更新

Route 53 ARC の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、Route 53 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。Route 53 ARC のドキュメント履歴のページで、自動 RSS アラートにサブスクライブすることもできます。

AWS Amazon Route 53 Application Recovery Controller のゾーン自動シフトの 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、多くの一般的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るため、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するわけではないことに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS マネージドポリシーで定義されているアクセス許可を変更することはできません。が AWS 管理ポリシーで定義されているアクセス許可 AWS を更新すると、その更新は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 AWS のサービス が起動されたとき、または既存のサービスで新しい API オペレーションが使用可能になったときに、 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： 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 Route 53 Application Recovery Controller がゾーン自動shift に対して次の操作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。

- 顧客が提供する 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、プラクティスの実行をモニタリングする
- 練習実行 (練習のゾーンシフト) を管理します。

詳細については、「Route 53 ARC でのゾーン自動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。

ゾーン自動shift の AWS マネージドポリシーの更新

Route 53 ARC でゾーン自動シフトの AWS マネージドポリシーの更新について、このサービスがこれらの変更の追跡を開始した以降のものについては、「」を参照してくださいAmazon Route 53 Application Recovery Controller の AWS マネージドポリシーの更新。このページの変更に関する自動通知を受け取るには、Route 53 ARC のドキュメントの履歴ページの RSS フィードにサブスクライブします。

ルーティングコントロールを使用して Amazon Route 53 Application Recovery Controller で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 Route 53 Application Recovery Controller 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 にデプロイした場合にユーザーの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、Route 53 ARC の機能である準備状況チェックについても説明しています。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機能と準備状況チェック機能、それらの設定方法、およびそれらの使用方法について説明します。

トピック

- Amazon Route 53 Application Recovery Controller でのルーティングコントロール
- Amazon Route 53 Application Recovery Controller の準備状況チェック

Amazon Route 53 Application Recovery Controller でのルーティングコントロール

複数の 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 Route 53 Application Recovery Controller 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。例えば、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。

Route 53 ARC のルーティングコントロールコンポーネントには、クラスター、コントロールパネル、ルーティングコントロール、ルーティングコントロールのヘルスチェックがあります。すべてのルーティングコントロールはコントロールパネルにグループ化されます。Route 53 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それらをグループ化したり、独自のカスタムコントロールパネルを作成したりもできます。コントロールパネルまたはルーティングコントロールを作成する前に、クラスターを作成する必要があります。Route 53 ARC の各クラスターは、5 つの AWS リージョンのエンドポイントからなるデータプレーンです。

ルーティングコントロールとルーティングコントロールのヘルスチェックを作成したら、ルーティングコントロール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に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) 、または 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、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ためにルーティングコントロールを作成して使用する方法について説明します。

> **Important**
>
> 災害シナリオにおけるアプリケーションのフェイルオーバープランの一環として、Route 53 ARC を使用したトラフィックの再ルーティングを準備する詳細については、「Route 53 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」を参照してください。

ルーティングコントロールについて

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

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

また、ルーティングコントロール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、Route 53 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。

Route 53 ARC のルーティングコントロールを Route 53 ヘルスチェックと DNS レコードセットと併用し、加重ルーティングポリシー付きの DNS レコードを使用することで、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、ブログ AWS 記事「Amazon Route 53 Application Recovery Controller を使用した耐障害性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少し時間がかかる場合があります。DNS 設定やその他の要因によっては、既存の接続が数分で完了したり、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すぐ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使用方法

Route 53 ARC のルーティングコントロールには、従来のヘルスチェックによるトラフィックの再ルーティングと比べるといくつか利点があります。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。

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んで、 のアプリケーションの耐障害性と可用性を向上させる例を次に示します 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。モニタリングシステムとヘルスチェックシステムからの情報に基づいて、レプリカとの間でフェイルアウェイを行うかどう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。この方法では、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、障害が発生した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 Route 53 Application Recovery Controller のリージョンサポートとリージョンサービスエンドポイントに関する詳細は、Amazon Web Services 全般のリファレンスにある「Amazon Route 53 Application Recovery Controller のエンドポイントとクォータ」を参照してください。

> **Note**
>
> Amazon Route 53 Application Recovery Controller 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 Route 53 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。

Route 53 ARC ルーティングコントロールは、Route 53 ARC ヘルスチェックの状態を変更するオン/オフスイッチです。このスイッチは、トラフィックをリダイレクトする DNS レコードに関連付けることができます。例えば、トラフィックをプライマリデプロイのレプリカからスタンバイデプロイのレプリカにリダイレクトします。

アプリケーション障害やレイテンシーの問題が発生した場合は、ルーティングコントロールの状態を更新して、例えばトラフィックをプライマリレプリカからスタンバイレプリカに移動できます。信頼性の高い Route 53 ARC データプレーン API オペレーションを使用して、ルーティングコントロールのクエリやルーティングコントロールの状態の更新を行うことで、ディザスタリカバリシナリオでのフェイルオーバーを Route 53 ARC に依存できます。詳細については、「Route 53 ARC API を使用して (推奨)、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」を参照してください。

Route 53 ARC は、5 つの冗長なリージョンエンドポイントのセットであるクラスター内で、ルーティングコントロールの状態を維持します。Route 53 ARC は、Amazon EC2 フリートにあるクラスター全体で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 リージョンで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 Route 53 ARC にルーティングコントロールの状態をクエリすると、コンセンサスビューが返されます。

5 つのクラスターエンドポイントのいずれかを操作して、ルーティングコントロールの状態を (例えば `Off` から `On` に) 更新できます。その後、Route 53 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。

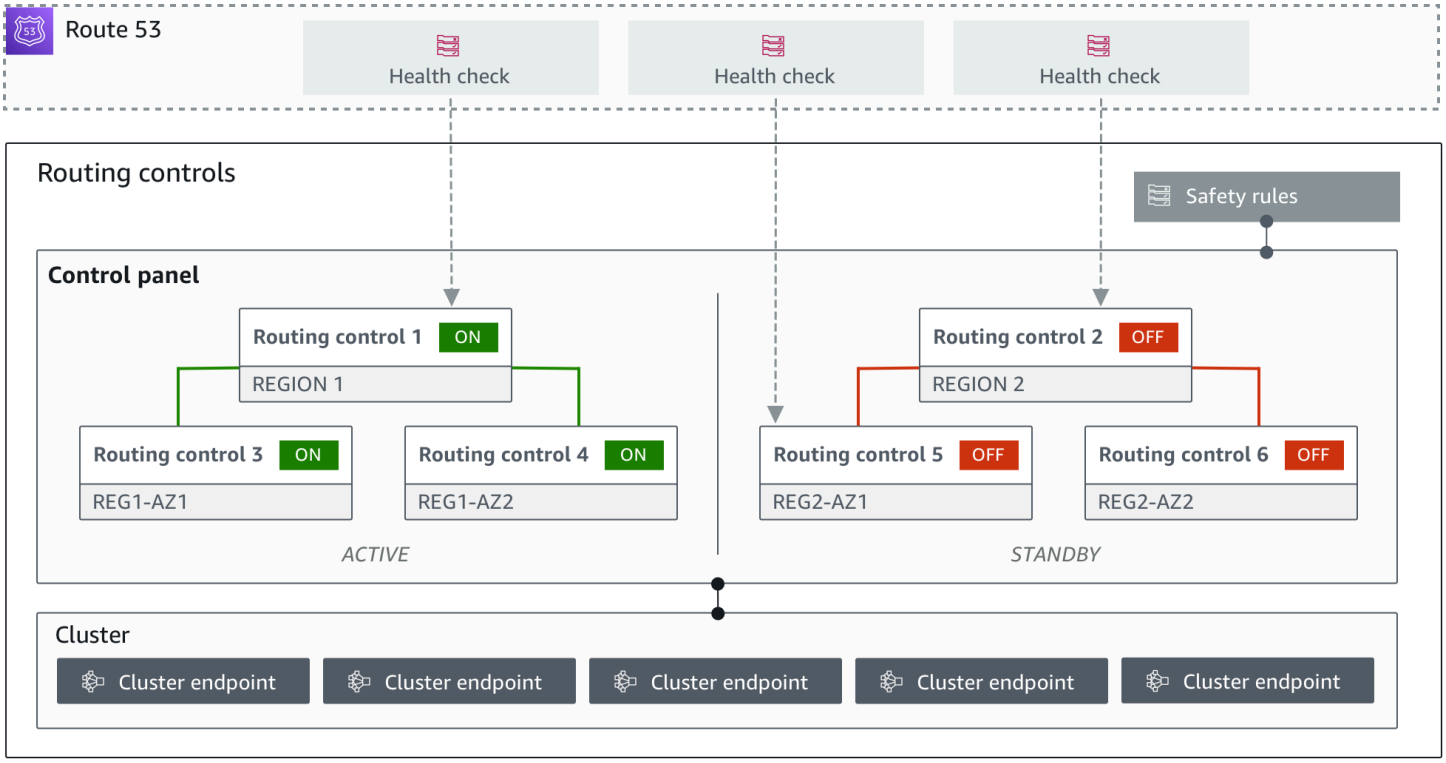
Route 53 ARC のデータプレーンは非常に信頼性が高く、セル間において手動でアプリケーションをフェイルオーバーできます。Route 53 ARC では、5 つのクラスターエンドポイントのうち少なくとも 3 つのエンドポイントに常にアクセスでき、ルーティングコントロールの状態を変更できるようにします。アクセスパターンを遅らせる可能性のある「うるさい隣人 (noisy neighbors)」の影響を受けないように、各 Route 53 ARC クラスターはシングルテナントであることに注意してください。

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

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

ルーティングコントロールのコンポーネント

次の図は、Route 53 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、Route 53 ARC は Amazon Route 53 のヘルスチェックを変更し、DNS トラフィックを別のセルにリダイレクトします。ルーティングコントロールに設定する安全ルールは、フェイルオープンシナリオやその他の意図しない結果を防ぐのに役立ちます。



以下は、Route 53 ARC におけるルーティングコントロール機能のコンポーネントです。

クラスター

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

ルーティングコントロール

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

ルーティングコントロールのヘルスチェック

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

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

コントロールパネル

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

デフォルトのコントロールパネル

クラスターを作成すると、Route 53 ARC でデフォルトのコントロールパネルが作成されます。デフォルトでは、クラスターで作成したすべてのルーティングコントロールがデフォルトのコントロールパネルに追加されます。もしくは、独自のコントロールパネルを作成して、関連するルーティングコントロールをグループ化することもできます。

安全ルール

安全ルールは、リカバリアクションによってアプリケーションの可用性が誤って損なわれないように、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)

Route 53 ARC の各クラスターには、ルーティングコントロールの状態の設定と取得に使用できる 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では、Route 53 ARC がメンテナンスのためにエンドポイントを定期的に起動または停止することを前提としているため、エンドポイントに接続するまで各エンドポイントを続けて試す必要があります。エンドポイントにアクセスして現在のルーティングコントロールの状態 (オンまたはオフ) を取得したり、ルーティングコントロールの状態を変更してアプリケーションのフェイルオーバーをトリガーしたりします。

ルーティングコントロールのデータプレーンとコントロールプレーン

フェイルオーバーとディザスタリカバリを計画する際は、フェイルオーバーメカニズムの耐障害性を考慮してください。フェイルオーバー中に依存するメカニズムは可用性が高く、災害シナリオで必要なときに使用できるようにすることをお勧めします。通常、信頼性と耐障害性を最大限に高めるには、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。どちらも信頼性を重視して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、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 AWS Management Console 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場合に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、 を使用して、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、ルーティングコントロールの状態と連携する API コール AWS Command Line Interface を作成す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リカバリ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Route 53 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 が高可用性の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 Route 53 Application Recovery Controller でのルーティングコントロールのタグ付け

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。

Route 53 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

Route 53 ARC でのタグ付けは、API を使用してのみ可能です。例えば、 AWS CLIを使用します。

以下は、 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 Route 53 Application Recovery Controller のTagResourceリカバリコントロール設定 API リファレンスガイド」の「」を参照してください。

Route 53 ARC でのルーティングコントロールの料金

Amazon Route 53 Application Recovery Controller は、サービスで使用するために設定した分のみ料金が請求されます。Route 53 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、「Route 53 ARC でクラスターのクロスアカウントをサポート」を参照してください。

Route 53 ARC の料金情報と料金例の詳細については、「Amazon Route 53 Application Recovery Controller の料金」を参照して、Amazon Route 53 Application Recovery Controller までスクロールします。

Amazon Route 53 Application Recovery Controller のマルチリージョンリカバリの開始方法

Amazon Route 53 Application Recovery Controller 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 にある 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で設定されていることを確認し、イベント中に 1 つの から別の 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た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。ゾーンシフトや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設定は必要ありません。詳細については、「ゾーンシフトとゾーンオートシフトを使用して Amazon Route 53 Application Recovery Controller でアプリケーションを復旧する」を参照してください。

Route 53 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するには、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、つまりセル 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リカバリ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 AWS CloudFormation と HashiCorp Terraform テンプレートを提供しています。詳細とテンプレートのダウンロードについては、「」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、アプリケーションの耐障害性が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、イベント発生時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。一方のレプリカの障害がもう一方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ステートフルデータをすべて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 Route 53 Application Recovery Controller 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、別の で実行されている冗長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。

の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 Route 53 ARC アクション AWS CLIを使用して変更することをお勧めします。API アクションはコンソールに依存しないため、耐障害性が向上します。

例えば、us-west-1 から us-east-1 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、一方からもう一方のレプリカ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 テンプレート AWS CloudFormation とサンプルアプリケーションを使用した HashiCorp Terraform テンプレートを使用して、Route 53 ARC のセットアップと使用を自分ですばやく検討できます。

サンプルアプリをデプロイしたら、テンプレートを使用して Route 53 ARC コンポーネントを作成し、アプリへのトラフィックフローを管理するルーティングコントロールを使用できます。独自のシナリオやアプリケーションに合わせて、テンプレートとプロセスを調整してください。

- AWS CloudFormation： 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、この Amazon S3 バケット の README 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。
- HashiCorp Terraform: サンプルアプリケーションと Terraform テンプレートの使用を開始するには、この Amazon S3 バケットの README 手順を参照してください。Terraform テンプレートの使用の詳細については、 HashiCorp ドキュメントを参照してください。

Route 53 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

Amazon Route 53 Application Recovery Controller でのルーティングコントロールのリカバリとフェイルオーバーの準備には、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係する DNS レコードの TTL 値を低くする

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

リカバリクラスターのデータプレーン API を使って Route 53 ARC でフェイルオーバータスクを実行するときは、Route 53 ARC の IAM ポリシーをユーザーにアタッチできます。詳細については、「Amazon Route 53 Application Recovery Controller 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係する DNS レコードの TTL 値を低くする

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 1 つの から別の AWS リージョン にシフトする場合、Amazon Route 53 Application Recovery Controller 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引き続きリクエストを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るだけでなく、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移動させるというトレードオフであることに注意してください。

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

Route 53 ARC のリージョンクラスターエンドポイントの、ローカルコピーをブックマークに保存するか、エンドポイントを再試行するために使用する自動化コードに保存することをお勧めします。障害イベントが発生すると、信頼性が非常に高いデータプレーンクラスターでホストされていない Route 53 ARC API オペレーションなど、一部の API オペレーションにアクセスできなくなることがあります。DescribeCluster API オペレーションを使用して、Route 53 ARC クラスターのエンドポイントを一覧表示できます。

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。

フェイルオーバーが必要な場合は、5 つのリージョンクラスターエンドポイントからランダムに選んだいずれかのエンドポイントを使って、ルーティングコントロールの状態を更新 (および取得) することが推奨されます。そのエンドポイントに障害が発生した場合は、他のリージョンエンドポイントを再試行します。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

Route 53 ARC データプレーン API を使用して、 ListRoutingコントロール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、ルーティングコントロールの状態を更新して、UpdateRoutingControlStateオペレーションでフェイルオーバーするトラフィックをリダイレクトします。 AWS CLI（これらの例のように）または AWS SDKsのいずれかを使用して記述したコードを使用できます。Route 53 ARC では、きわめて信頼性の高い方法として、データプレーンで API を使用してトラフィックをフェイルオーバーできます。 AWS 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。

データプレーン API を使用するには、Route 53 ARC のリージョンクラスターエンドポイントのいずれかに接続します。そのエンドポイントが使用できない場合は、別のクラスターエンドポイントに接続します。

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

Route 53 ARC でのフェイルオーバーのテスト

Route 53 ARC のルーティングコントロールを使って、プライマリのアプリケーションスタックからセカンダリのアプリケーションスタックにフェイルオーバーして、定期的にフェイルオーバーをテストします。追加した Route 53 ARC の構造がスタック内の正常なリソースと一致していること、および、すべてが予定どおりに機能していることを確認することが重要です。使用している環境に Route 53 ARC をセットアップした後でこのテストを行い、フェイルオーバー環境の準備が整うように定期的にテストを続ける必要があります。障害が発生した場合には、ユーザーのダウンタイムを回避できるよう、セカンダリシステムを起動してすばやく稼働させなければなりません。

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 Route 53 Application Recovery Controller 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表と、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 Route 53 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 Route 53 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	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
クラスターを作成する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	CreateCluster「��
クラスターを記述する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DescribeCluster「��
クラスターを削除	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DeleteCluster「��
アカウントのクラスターを一覧表示する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	ListClusters「��
ルーティングコントロールを作成する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	CreateRouting「コントロール」を参照してください
ルーティングコントロールについて説明する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DescribeRouting「コントロール」を参照してください
ルーティングコントロールを更新する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	UpdateRouting「コントロール」を参照してください

アクション	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
ルーティングコントロールを削除する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DeleteRouting「コントロール」を参照してください
ルーティングコントロールを一覧表示する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	ListRouting「 コントロール」
コントロールパネルを作成する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	CreateControl「パネル」を参照
コントロールパネルを説明する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DescribeControl「パネル」を参照
コントロールパネルを更新する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	UpdateControl「パネル」を参照
コントロールパネルを削除する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	DeleteControl「パネル」を参照
コントロールパネルを一覧表示する	「Route 53 ARC でルーティングコントロールのコンポーネントを作成する」を参照	ListControl「パネル」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	CreateSafety「 ルール」
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	DescribeSafety「 ルール」
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	UpdateSafety「 ルール」

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

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 Route 53 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	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
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	GetRoutingControlState「��
ルーティングコントロールを一覧表示する	該当なし	ListRouting「コントロール」
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	UpdateRoutingControlState「��

アクション	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
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	UpdateRoutingControlStates「��

AWS SDK でこのサービスを使用する

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
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	PowerShell コード例のツール
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

SDK ドキュメント	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 Route 53 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 Route 53 Application Recovery Controller 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操作する簡単なアプリケーション例について説明します。この例は、CLI を使用してルーティングコントロールを操作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。

Amazon Route 53 Application Recovery Controller 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 またはレプリカ間の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。Route 53 ARC クラスターは、グローバルにデプロイされるリージョンエンドポイントのセットです。クラスターエンドポイントは、ルーティングコントロールの状態の設定と取得に使用できる可用性の高い API を提供します。ルーティングコントロール機能のコンポーネントの詳細については、「ルーティングコントロールのコンポーネント 」を参照してください。

> Note
>
> Route 53 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 Route 53 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。例え

ば、リカバリグループ、コントロールパネル、クラスターを作成するときは、 `region`パラメータを使用します。
クラスターを作成すると、Route 53 ARC はリージョンのエンドポイントのセットを提供します。ルーティングコントロールの状態を取得または更新するには、CLI コマンドでリージョンエンドポイント (AWS リージョン およびエンドポイント URL) を指定する必要があります。

の使用の詳細については AWS CLI、 AWS CLI 「 コマンドリファレンス」を参照してください。ルーティングコントロール API アクションのリストについては、 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、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、最初にクラスターを作成します。

ルーティングコントロールコンポーネントを設定する

最初のステップでは、クラスターを作成します。Route 53 ARC クラスターは、5 つの異なる のそれぞれに 1 つずつ、5 つのエンドポイントのセットです AWS リージョン。Route 53 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。

```
aws route53-recovery-control-config --region us-west-2 create-cluster --cluster-name
 NewCluster
```

```
{
    "Cluster": {
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "Name": "NewCluster",
        "Status": "PENDING"
    }
}
```

Route 53 ARC リソースを初めて作成すると、クラスターの作成中はステータスが `PENDING` になります。その進行状況は、`describe-cluster` を呼び出して確認できます。

1b. クラスターを記述します。

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

```
{
    "Cluster":{
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "ClusterEndpoints":[
            {"Endpoint": "https://host-aaaaaa.us-east-1.example.com", "Region":"us-
east-1"},
            {"Endpoint": "https://host-bbbbbb.ap-southeast-2.example.com",
 "Region":"ap-southeast-2"},
            {"Endpoint": "https://host-cccccc.eu-west-1.example.com", "Region":"eu-
west-1"},
            {"Endpoint": "https://host-dddddd.us-west-2.example.com", "Region":"us-
west-2"},
            {"Endpoint": "https://host-eeeeee.ap-northeast-1.example.com",
 "Region":"ap-northeast-1"}
        ]
        "Name": "NewCluster",
        "Status": "DEPLOYED"
    }
}
```

ステータスが DEPLOYED の場合、ユーザーが操作できるエンドポイントのセットを含むクラスターを、Route 53 ARC が正常に作成したことを意味します。`list-clusters` を呼び出すと、すべてのクラスターを一覧表示できます。

1c. クラスターを一覧表示します。

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

```
{
    "Clusters": [
        {
            "ClusterArn": "arn:aws:route53-recovery-
control::111122223333:cluster/1234abcd-abcd-1234-abcd-1234abcdefgh",
            "ClusterEndpoints":[
```

```
                {"Endpoint": "https://host-aaaaaa.us-east-1.example.com", "Region":"us-
east-1"},
                {"Endpoint": "https://host-bbbbbb.ap-southeast-2.example.com",
 "Region":"ap-southeast-2"},
                {"Endpoint": "https://host-cccccc.eu-west-1.example.com", "Region":"eu-
west-1"},
                {"Endpoint": "https://host-dddddd.us-west-2.example.com", "Region":"us-
west-2"},
                {"Endpoint": "https://host-eeeeee.ap-northeast-1.example.com",
 "Region":"ap-northeast-1"}
            ],
            "Name": "AnotherCluster",
            "Status": "DEPLOYED"
        },
        {
            "ClusterArn": "arn:aws:route53-recovery-
control::111122223333:cluster/5678abcd-abcd-5678-abcd-5678abcdefgh",
            "ClusterEndpoints": [
                {"Endpoint": "https://host-ffffff.us-east-1.example.com", "Region":"us-
east-1"},
                {"Endpoint": "https://host-gggggg.ap-southeast-2.example.com",
 "Region":"ap-southeast-2"},
                {"Endpoint": "https://host-hhhhhh.eu-west-1.example.com", "Region":"eu-
west-1"},
                {"Endpoint": "https://host-iiiiii.us-west-2.example.com", "Region":"us-
west-2"},
                {"Endpoint": "https://host-jjjjjj.ap-northeast-1.example.com",
 "Region":"ap-northeast-1"}
            ],
            "Name": "NewCluster",
            "Status": "DEPLOYED"
        }
    ]
}
```

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、Route 53 ARC のルーティングコントロールを整理のために論理的にまとめたものです。クラスターを作成すると、Route 53 ARC は `DefaultControlPanel` という名前のコントロールパネルを自動的に提供します。このコントロールパネルはすぐに使用できます。

コントロールパネルは 1 つのクラスターにのみ存在できます。コントロールパネルを別のクラスターに移動する場合は、そのコントロールパネルを削除して 2 つ目のクラスターで作成する必要が

あります。アカウントのすべてのコントロールパネルは、list-control-panels を呼び出すことで確認できます。特定のクラスター内のコントロールパネルだけを表示するには、--cluster-arn フィールドを追加します。

2a. コントロールパネルを一覧表示します。

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

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

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

2b. コントロールパネルを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-control-panel \
        --control-panel-name NewControlPanel2 \
        --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": false,
```

```
        "Name": "NewControlPanel2",
        "RoutingControlCount": 0,
        "Status": "PENDING"
    }
}
```

Route 53 ARC リソースを初めて作成すると、作成中はステータスが PENDING になります。`describe-control-panel` を呼び出して、進行状況を確認できます。

2c. コントロールパネルを記述します。

```
aws route53-recovery-control-config --region us-west-2 describe-control-panel \
    --control-panel-arn arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456
```

```
{
    "ControlPanel": {
        "ControlPanelArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "ClusterArn": "arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh",
        "DefaultControlPanel": true,
        "Name": "DefaultControlPanel",
        "RoutingControlCount": 0,
        "Status": "DEPLOYED"
    }
}
```

3. ルーティングコントロールを作成する

これでクラスターをセットアップし、コントロールパネルを確認したので、ルーティングコントロールの作成を開始できます。ルーティングコントロールを作成するときには、少なくとも、ルーティングコントロールを組み込むクラスターの Amazon リソースネーム (ARN) を指定する必要があります。ルーティングコントロールのコントロールパネルの ARN を指定することもできます。また、コントロールパネルが配置されているクラスターも指定する必要があります。

コントロールパネルを指定しない場合、ルーティングコントロールは自動的に作成されたコントロールパネル (`DefaultControlPanel`) に追加されます。

`create-routing-control` を呼び出して、ルーティングコントロールを作成できます。

3a. ルーティングコントロールを作成します。

```
aws route53-recovery-control-config --region us-west-2 create-routing-control \
    --routing-control-name NewRc1 \
    --cluster-arn arn:aws:route53-recovery-control::111122223333:cluster/5678abcd-
abcd-5678-abcd-5678abcdefgh
```

```
{
    "RoutingControl": {
        "ControlPanelArn": " arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
        "Name": "NewRc1",
        "RoutingControlArn": "arn:aws:route53-recovery-
control::111122223333:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567",
        "Status": "PENDING"
    }
}
```

ルーティングコントロールは他の Route 53 ARC リソースと同じ作成パターンに従うため、describe オペレーションを呼び出すことで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 Route 53 Application Recovery Controller のAssertionRuleルーティングコントロール API リファレンスガイド」の「」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 Route 53 Application Recovery Controller のGatingRuleルーティングコントロール API リファレンスガイド」の「」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、安全ルールがデータプレーンに伝播された後に、安全ルールを記述、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
0123456bbbbbbb0123456bbbbbb0123456333333444444
```

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、Route 53 ARC ルーティングコントロールはヘルスチェックをフェイルに設定し、Route 53 がトラフィックを再ルーティングできるようにします。(ヘルスチェックはアプリケーションの正常性を無効にします。単にトラフィックを再ルーティングする方法として使用されます。)

例として、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 つはフェイルオーバー先となるセカンダリとして設定します。

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. Route 53 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 の Route 53 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 Route 53 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、Amazon Route 53 AWS CLI コマンドリファレンスの change-resource-record-sets コマンドの指示に従います。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 Route 53 Application Recovery Controller 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。

Route 53 ARC は、作成したクラスターごとに、5 つの AWS リージョンに 1 つずつ、クラスターエンドポイントのセットを提供します。ルーティングコントロールの状態を取得または設定するためにクラスターを呼び出すときは、これらのリージョンエンドポイント (AWS リージョン およびエンドポイント URL) 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の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。Route 53 ARC は、トラフィックの再ルーティングやセル間のフェイルオーバーを

可能にする Route 53 ARC ルーティングコントロールデータプレーンにより、きわめて高い信頼性を実現します。Route 53 ARC をフェイルオーバーに使用することに関するその他の推奨事項については、「Route 53 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」を参照してください。

ルーティングコントロールを作成すると、状態は Off に設定されます。つまり、そのルーティングコントロールのターゲットセルには、トラフィックはルーティングされません。ルーティングコントロールの状態を確認するには、get-routing-control-state コマンドを実行します。

指定するリージョンとエンドポイントを判断するには、describe-clusters コマンドを実行して ClusterEndpoints を表示します。各 ClusterEndpoint にはリージョンとそれに対応するエンドポイントが含まれ、これらを使用してルーティングコントロールの状態を取得または更新できます。DescribeCluster はリカバリコントロール設定 API オペレーションです。Route 53 ARC リージョンクラスターエンドポイントのローカルコピーをブックマークに保存するか、エンドポイントを再試行するために使用する自動化コードの中にハードコードしておくことを推奨します。

1. ルーティングコントロールを一覧表示する

信頼性の高い Route 53 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の更新

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
```

```
{}
```

Route 53 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

- Route 53 ARC でルーティングコントロールのコンポーネントを作成する
- Route 53 ARC でのルーティングコントロール状態の表示と更新
- ルーティングコントロールの安全ルールの作成
- Route 53 ARC でクラスターのクロスアカウントをサポート

Route 53 ARC でルーティングコントロールのコンポーネントを作成する

このセクションでは、Amazon Route 53 Application Recovery Controller 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することで AWS Resource Access Manager、1 つのクラスターがルーティングコントロールや、複数の が所有する他の Route 53 ARC リソースをホストできるようにします AWS アカウント。Route 53 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 Route 53 ARC でクラスターのクロスアカウントをサポート。料金情報についてはAmazon Route 53 Application Recovery Controller の料金表を参照し、Amazon Route 53 までスクロールしてください。

トラフィックをフェイルオーバーするルーティングコントロールを使用するには、アプリケーション内リソースの Amazon Route 53 DNS レコードに関連付けるルーティングコントロールのヘルスチェックを作成します。例として、アプリケーションのプライマリセルとして設定したセルと、フェイルオーバー先のセカンダリセルとして設定したセルの 2 つのセルがあるとします。

フェイルオーバーのヘルスチェックを設定するには、以下を実行してください。

1. 各セルにルーティングコントロールを作成します。
2. 各ルーティングコントロールにヘルスチェックを作成します。
3. 2 つの DNS レコード (例えば、2 つの DNS フェイルオーバーレコード) を作成し、それぞれにヘルスチェックを関連付けます。

ルーティングコントロールを作成する別のシナリオとしては、ゲートルールである安全ルールを作成する場合があります。この場合、ルーティングコントロールはゲートのルーティングコントロールとして使用するため、ルーティングコントロールをヘルスチェックと DNS レコードに関連付ける必要はありません。詳細については、「ルーティングコントロールの安全ルールの作成 」を参照してください。

Route 53 ARC コンソールでルーティングコントロールのコンポーネントを作成する手順については、以下のセクションに記載しています。Route 53 ARC でリカバリコントロール設定の API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

Route 53 ARC でクラスターを作成する

Route 53 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 Route 53 ARC コンソールを開きますhttps://console.aws.amazon.com/route53recovery/home#/dashboard。
2. [Clusters] 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。

Route 53 ARC でルーティングコントロールを作成する

トラフィックをルーティングする各セルに対してルーティングコントロールを作成します。例えば、リカバリ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 にセルがありAWS リージョン、各リージョン内の各アベイラビリティーゾーンに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 Route 53 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。

Route 53 ARC でルーティングコントロールのヘルスチェックを作成する

トラフィックの再ルーティングに使用する各ルーティングコントロールに、ルーティングコントロールのヘルスチェックを関連付けます。次に、各ヘルスチェックに Amazon Route 53 DNS レコード (フェイルオーバー DNS レコードなど) を設定します。そうすると、関連するルーティングコントロールの状態を更新して、On や Off に設定するだけで、Amazon Route 53 Application Recovery Controller のトラフィックを再ルーティングできます。

> **Note**
>
> 既存のルーティングコントロールのヘルスチェックを編集して、別のルーティングコントロールに関連付けることはできません。

ルーティングコントロールのヘルスチェックを作成するには

1. で Route 53 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。Route 53 ARC がルーティング制御を使用してトラフィックを正しくフェイルオーバーするには、まず、Route 53 に 2 つのフェイルオーバーレコード (プライマリとセカンダリ) を作成します。DNS フェイルオーバーレコードの設定に関する詳細については、「ヘルスチェックの概念」を参照してください。

プライマリフェイルオーバーレコードを作成すると、値は次のようになります。

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Primary
Failover: Primary
TTL: 0
Resource Records:
Value: cell1.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

セカンダリフェイルオーバーレコードの値は、次のようになります。

```
Name: myapp.yourdomain.com
Type: CNAME
Set Identifier: Secondary
Failover: Secondary
TTL: 0
Resource Records:
Value: cell2.yourdomain.com
Health Check ID: xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

ここで、障害が発生したためにトラフィックを再ルーティングしたいとしましょう。そのためには、関連するルーティングコントロールの状態を更新して、プライマリルーティングコントロールの状態を OFF に、セカンダリルーティングコントロールの状態を ON に変更します。これを行うと、関連するヘルスチェックによってプライマリレプリカへのトラフィックの送信が停止され、代わりにセカンダリレプリカへルーティングされます。ルーティング制御によるトラフィックのフェイルオーバーの詳細については、「Route 53 ARC API を使用して (推奨)、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」を参照してください。

Route 53 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 Route 53 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。

Route 53 ARC でコントロールパネルを作成する

Amazon Route 53 Application Recovery Controller のコントロールパネルで、関連するルーティングコントロールをグループ化できます。コントロールパネルでは、フェイルオーバーの範囲に応じて、アプリケーション内のマイクロサービス、アプリケーション全体、またはアプリケーションのグループに対応するルーティングコントロールを設定できます。ルーティングコントロールをコントロールパネルにグループ化することの利点は、コントロールパネルと安全ルールを併用することで、トラフィックのルーティング変更を防止できる点にあります。

クラスターを作成すると、Route 53 ARC でデフォルトのコントロールパネルが作成されます。デフォルトのコントロールパネルをルーティングコントロールに使用することも、複数のコントロールパネルを作成してルーティングコントロールをグループ化することもできます。コントロールパネル名は ASCII 文字のみサポートされることに注意してください。

Route 53 ARC コンソールでコントロールパネルを作成する手順については、以下のセクションに記載しています。Route 53 ARC でリカバリコントロール設定の API オペレーションを使用する方法については、「ルーティング制御 API オペレーション」を参照してください。

コントロールパネルを作成するには

1. で Route 53 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。

Route 53 ARC でのルーティングコントロール状態の表示と更新

このセクションでは、Amazon Route 53 Application Recovery Controller 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 または場合によってはアベイラビリティーゾーンです。ルーティングコントロールの状態が On の場合、トラフィックはそのルーティングコントロールによって制御されているセルに流れます。

論理的なフェイルオーバーグループであるコントロールパネルに、ルーティングコントロールをグループ化します。例えば、コンソールでコントロールパネルを開くと、グループ化されたルーティングコントロールを一度に表示して、トラフィックがどこに流れているかを確認できます。

ルーティングコントロールの状態は、Route 53 ARC コンソールまたは Route 53 ARC API を使用して更新できますが、API を使用してルーティングコントロールの状態を更新することをお勧めします。まず、Route 53 ARC はこれらのアクションを実行するデータプレーン内の API に関して、非常に高い信頼性があります。この点が重要となるのはルーティングコントロールの状態を変更する際です。ルーティングの状態変更は、アプリケーションのトラフィックを再ルーティングしてセル間でフェイルオーバーするためです。さらに、API を使用すれば、接続先のクラスターエンドポイントが使用できない場合、必要に応じて別のクラスターエンドポイントにローテーションで接続を試みることができます。

1 つのルーティングコントロールの状態を更新することも、複数のルーティングコントロールの状態を同時に更新することもできます。例えば、アプリケーションのレイテンシーが増大しているアベイラビリティーゾーンなど、あるルーティングコントロールの状態を Off に設定して、あるセルにトラフィックが流れないようにしたい場合が考えられます。同時に、別のルーティングコントロールの状態を On に設定して、別のセルまたは別のアベイラビリティーゾーンへのトラフィックフローを開始したい場合、このシナリオでは、両方のルーティングコントロールの状態を同時に更新して、トラフィックを続けて流すことができます。

トピック

- Route 53 ARC API を使用して (推奨)、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
- 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

Route 53 ARC API を使用して (推奨)、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

Amazon Route 53 Application Recovery Controller 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 Route 53 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。

ルーティング制御は高可用性クラスターに保存されるため、Route 53 ARC は API を使用してルーティング制御の状態を更新することで、セル (AWS リージョン) 間のフェイルオーバーの信頼性を大きく高めます。Route 53 ARC では、5 つのリージョンクラスターエンドポイントのうち少なくとも 3 つのエンドポイントに常にアクセスでき、ルーティング制御の状態を変更できます。API を使用してルーティング制御の状態を取得または変更するには、いずれかのリージョンクラスターエンドポイントに接続します。エンドポイントが使用できない場合は、別のクラスターエンドポイントに接続してみてください。

Route 53 コンソールで、または API アクション を使用して、クラスターのリージョンクラスターエンドポイントのリストを表示できますDescribeCluster。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。

Route 53 ARC API オペレーションを使用してルーティング制御の状態を取得および更新し、リージョンクラスターエンドポイントを操作する詳しい情報とコード例を提供しています。詳細については、次を参照してください。

- リージョンクラスターエンドポイント間をローテーションして、ルーティングコントロールの状態を取得および設定する方法を示すコード例については、「AWS SDKsアクション」を参照してください。
- を使用してルーティングコントロールの状態を取得および更新 AWS CLI する方法については、「」を参照してください を使用してルーティングコントロールと状態を一覧表示および更新する AWS CLI。

でのルーティングコントロールの状態の取得と更新 AWS 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 Route 53 Application Recovery Controller API を使用する場合とは異なり、コンソールではクラスターエンドポイントを選択してローテーションするプロセスはありません。さらに、Route 53 ARC のデータプレーンでは非常に高い信頼性を提供する一方で、コンソールの可用性は高くありません。これらの理由により、運用オペレーションでルーティングコントロールの状態を取得および更新するには、Route 53 ARC API を使用することをお勧めします。

Route 53 ARC をフェイルオーバーに使用することに関するその他の推奨事項については、「Route 53 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」を参照してください。

コンソールでルーティングコントロールを表示および更新するには、以下の手順に従ってください。

ルーティングコントロールの状態を取得するには

1. で Route 53 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手段を講じたい場合があります。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。Route 53 ARC でルーティングコントロールにこのような安全対策を確立するには、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

安全ルールには、アサーションルールとゲートルールの 2 種類があり、これらを使用してフェイルオーバーをさまざまな方法で保護できます。

アサーションルール

アサーションルールでは、1 つまたは一連のルーティングコントロールの状態を変更すると、Route 53 ARC はルールを作成したときに設定した基準を満たす必要があります。基準が満たされていない場合、ルーティングコントロールの状態は変更できません。

これが役立つ例としては、フェイルオープンシナリオを防ぐ場合です。例えば、あるセルへのトラフィックの流れを停止しても、別のセルへトラフィックの流れが開始しないというシナリオです。これを回避するために、アサーションルールでは、コントロールパネルにある一連のルーティングコントロールのうち、少なくとも 1 つのルーティングコントロールが常時 On に設定されていることを確認します。これにより、トラフィックはアプリケーションの少なくとも 1 つのリージョンまたはアベイラビリティーゾーンに流れるようになります。

この条件を適用する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 Route 53 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 Route 53 Application Recovery Controller のAssertionRuleルーティングコントロール API リファレンスガイド」の「」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。

この種の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 Route 53 ARC ルーティングコントロール API オペレーションを使用する例 AWS CLI。

ゲートルール API オペレーションプロパティの詳細については、Amazon Route 53 Application Recovery Controller の GatingRule Routing Control API リファレンスガイドの「」を参照してください。 Amazon Route 53

コンソールで安全ルールを作成する

このセクションの手順では、Route 53 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 Route 53 Application Recovery Controller でリカバリおよびルーティングコントロール API オペレーションを使用する方法については、「 ルーティング制御 API オペレーション」を参照してください。

安全ルールを作成するには

1. で Route 53 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 Route 53 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「オペレーション用に提供されている情報」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 Route 53 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

このセクションの手順では、Route 53 ARC コンソールで安全ルールを編集または削除する方法について説明します。名前の変更や待機期間の更新など、安全ルールでは限定的な編集のみ行えます。その他の変更を行うには、安全ルールを削除して再作成します。

Amazon Route 53 Application Recovery Controller で API オペレーションを使用する方法については、「 ルーティング制御 API オペレーション」を参照してください。

安全ルールを削除するには

1. で Route 53 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 および SDK での `SafetyRulesToOverride`プロパティの使用の詳細については、「」を参照してくださいUpdateRoutingControlState。 SDKs

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
```

Route 53 ARC でクラスターのクロスアカウントをサポート

Amazon Route 53 Application Recovery Controller 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。Route 53 ARC 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 内外の特定 AWS Organizations
- の組織内の組織単位 AWS Organizations
- の組織全体 AWS Organizations

の詳細については 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 Route 53 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 AWS アカウント 場合、指定した他の 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、異なるチーム間のルーティング機能をより詳細に制御し、柔軟性を高めることができます。

AWS RAM は、 AWS お客様が 間で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) 内で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的で制御された方法です。

クラスターを共有すると、組織が必要とするクラスターの総数を減らせます。共有クラスターを使用すると、クラスターを実行するための総コストを複数のチームに割り振ることができ、低コストで Route 53 ARC の利点を最大化できます (クラスターでホストされるリソースを作成しても、所有者や参加者に追加コストは発生しません)。アカウント間でクラスターを共有すると、複数のアプリケーションを Route 53 ARC にオンボーディングするプロセスも簡単になります。特に、多数のアプリケーションが複数のアカウントや運用チームに分散している場合に有効です。

Route 53 ARC でクロスアカウント共有を開始するには、 AWS RAMでリソース共有を作成します。リソース共有は、アカウントが所有するクラスターを共有する権限を持つ参加者を指定します。その後、参加者は を使用するか、 AWS Management Console または AWS Command Line Interface AWS SDKs を使用して Route 53 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

所有するクラスターを共有すると、クラスターの共有先に指定された参加者は、そのクラスター内で独自の Route 53 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、 または AWS Command Line Interface SDK で AWS RAM API オペレーションを使用できます。 AWS SDKs

ユーザーが 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許可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 SDK で AWS RAM API オペレーションを使用します。 SDKs

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。

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 SDK で AWS RAM API オペレーションを使用します。 SDKs

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。Route 53 ARC コンソールと AWS CLIを使用して、共有リソースに関する情報を取得することもできます。

一般に、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、Route 53 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 と共有すると AWS アカウント、クラスターの使用が許可されている参加者は、クラスター内にコントロールパネル、ルーティングコントロール、および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。例えば、 AWS Command Line Interface または AWS SDKs を使用して Route 53 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

Route 53 ARC のクラスターの所有者には、そのクラスターに関連する費用が請求されます。クラスターの所有者側でも参加者側でも、クラスターでホストされるリソースの作成に追加費用はかかりません。

料金の詳細情報と例については、Amazon Route 53 Application Recovery Controller の料金表を参照し、Amazon Route 53 Application Recovery Controller の項目までスクロールダウンしてください。

クォータ

共有クラスターで作成されたすべてのリソース (共有クラスターへのアクセス権を持つすべての参加者が作成したリソースを含む) は、そのクラスターや他のリソース (ルーティングコントロールなど) で有効なクォータにカウントされます。クラスターリソースを共有するアカウントのクォータがクラスター所有者のクォータよりも高い場合、クラスター所有者のクォータは、共有しているアカウントのクォータよりも優先されます。

この仕組みの詳細については、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

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

例えば、所有者がクラスターあたりのコントロールパネル数に対して 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、参加者のクォータが 150 に設定

されているとします。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 (つまり 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、それらはすべてクラスター全体の 50 個のコントロールパネルのクォータにカウントされます。

ルーティングコントロールのクォータ

ルーティングコントロールには複数のクォータがあります。コントロールパネルあたりのクォータ、クラスターあたりのクォータ、安全ルールあたりのクォータです。所有者のクォータは、これらすべてのクォータに優先されます。

例えば、所有者がクラスターあたりのルーティングコントロール数に対して 300 のクォータを持ち、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。

例えば、所有者がコントロールパネルあたりの安全ルール数に対して 20 のクォータを持ち、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 Route 53 Application Recovery Controller でのルーティングコントロールのログ記録とモニタリング

AWS CloudTrail を使用して、Amazon Route 53 Application Recovery Controller のルーティングコントロールをモニタリングし、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 Route 53 ARC API コールのログ記録 AWS CloudTrail

を使用した Route 53 ARC API コールのログ記録 AWS CloudTrail

Amazon Route 53 Application Recovery Controller は AWS CloudTrail、Route 53 ARC のユーザー、ロール、または AWS サービスによって実行されたアクションを記録するサービスである と統合されています。 は、Route 53 ARC のすべての API コールをイベントとして CloudTrail キャプチャします。キャプチャされたコールには、Route 53 ARC コンソールからのコールと、Route 53 ARC API オペレーションへのコードコールが含まれます。

証跡を作成する場合は、Route 53 ARC の CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。 Amazon S3 Route 53 証跡を設定しない場合でも、コンソールのイベント履歴 で最新の CloudTrail イベントを表示できます。

で収集された情報を使用して CloudTrail、Route 53 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。

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

の Route 53 ARC 情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 は で有効になります。Route 53 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 CloudTrail イベント履歴 の他の AWS サービスイベントとともに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 「イベント履歴の使用」を参照してください。

Route 53 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、 はログファイル CloudTrail 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口グに記録し、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 ログで収集されたデータをより詳細に分析し、それに基づく対応を行うように他の 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:

- 「証跡作成の概要」
- CloudTrail でサポートされているサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

すべての Route 53 ARC アクションは によってログに記録 CloudTrail され、Amazon Route 53 Application Recovery Controller のリカバリ準備 API リファレンスガイド、Amazon Route 53 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 Route 53 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、 および `CreateRecoveryGroup`アクションを呼び出すと`UpdateRoutingControlState`と`CreateCluster`、 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) ユーザーの認証情報のどちらを使用して送信されたか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが別の AWS サービスによって送信されたかどうか。

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

イベント履歴での Route 53 ARC イベントの表示

CloudTrail では、イベント履歴 で最近のイベントを表示できます。Route 53 ARC API リクエストのイベントを表示するには、コンソールの上部にあるリージョンセレクターで [米国西部 (オレゴン)] を指定する必要があります。詳細については、「 ユーザーガイド」の CloudTrail 「イベント履歴の使用AWS CloudTrail 」を参照してください。

Route 53 ARC のログファイルエントリを理解する

証跡は、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できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。 CloudTrail ログファイルは、パブリック API コール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:sts::111122223333:assumed-role/admin/smithj",
     "accountId": "111122223333",
     "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
     "sessionContext": {
          "sessionIssuer": {
              "type": "Role",
              "principalId": "A1B2C3D4E5F6G7EXAMPLE",
              "arn": "arn:aws:iam::111122223333:role/admin",
              "accountId": "111122223333",
              "userName": "admin"
          },
          "webIdFederationData": {},
          "attributes": {
              "mfaAuthenticated": "false",
              "creationDate": "2021-06-30T04:44:41Z"
          }
      }
  },
  "eventTime": "2021-06-30T04:45:46Z",
  "eventSource": "route53-recovery-control-config.amazonaws.com",
  "eventName": "UpdateRoutingControl",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.50",
  "userAgent": "aws-cli/2.0.0 Python/3.8.2 Darwin/19.6.0 botocore/2.0.0dev7",
  "requestParameters": {
      "RoutingControlName": "XYZRoutingControl3",
      "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
```

```
  },
  "responseElements": {
      "RoutingControl": {
          "ControlPanelArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456",
          "Name": "XYZRoutingControl3",
          "Status": "DEPLOYED",
          "RoutingControlArn": "arn:aws:route53-recovery-
control::012345678:controlpanel/0123456bbbbbbb0123456bbbbbb0123456/routingcontrol/
abcdefg1234567"
      }
  },
  "requestID": "6090509a-5a97-4be6-8e6a-7d73example",
  "eventID": "9cab44ef-0777-41e6-838f-f249example",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "111122223333"
}
```


ルーティングコントロールのための 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、誰を認証し (サインインさせ)、誰に Route 53 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 AWS のサービス 使用できる です。

内容

- Amazon Route 53 Application Recovery Controller でのルーティングコントロールと IAM の連携方法
- Amazon Route 53 Application Recovery Controller 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 Route 53 Application Recovery Controller 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 Route 53 Application Recovery Controller でのルーティングコントロールと IAM の連携方法

IAM を使用して Amazon Route 53 Application Recovery Controller 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 機能について学びます。

Amazon Route 53 Application Recovery Controller のルーティングコントロールで使用できる IAM の機能

IAM 機能	ルーティングコントロールのサポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	No
ABAC (ポリシー内のタグ)	部分的
一時的な認証情報	Yes
プリンシパル権限	Yes
サービスロール	いいえ
サービスリンクロール	いいえ

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

Route 53 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ポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限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 ポリシーの作成」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 Route 53 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 Route 53 Application Recovery Controller 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	No

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に対するサポート	はい

管理者は AWS JSON 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んなリソースにどんな条件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、関連する 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権限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権限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 Route 53 ARC アクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Route 53 Recovery Controls 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。

ルーティングコントロールの Route 53 ARC のポリシーアクションは、使用している 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 Route 53 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 Route 53 Application Recovery Controller 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。

Route 53 ARC のポリシーリソース

ポリシーリソースに対するサポート	はい

管理者は AWS JSON 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どのような条件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、リソースレベルの権限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": "*"
```

「サービス認証リファレンス」では、Route 53 ARC に関連する次の情報を確認できます。

リソースタイプとその ARN のリスト、および各リソースの ARN で指定できるアクションについては、「サービス認可リファレンス」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 クラスター で定義されるアクション。

ルーティングコントロールの Route 53 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 Route 53 Application Recovery Controller 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。

Route 53 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	はい

管理者は AWS JSON 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んなリソースにどんな条件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することで、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 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS 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演算を使用して条件 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の Route 53 ARC 条件キーのリストを確認するには、「サービス認証リファレンス」の以下のトピックを参照してください。

- Amazon Route 53 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 コントロールで定義されるリソース」および「Amazon Route 53 Recovery クラスターで定義されるリソース」を参照してください。

ルーティングコントロールの Route 53 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 Route 53 Application Recovery Controller 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

Route 53 ARC のアクセスコントロールリスト (ACL)

ACL のサポート	No

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権限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。

Route 53 ARC での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート	部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限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操作を許可するように ABAC ポリシーを設計します。

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はいです。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 とは?」を参照してください。ABAC をセットアップするステップ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「属性に基づく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。

Route 53 ARC ルーティングコントロールには、ABAC に対する以下のサポートが含まれています。

- Recovery Control Config は ABAC をサポートしています。
- リカバリクラスターは ABAC をサポートしていません。

Route 53 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	はい

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は、一時的な認証情報を使用しています。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスは自動的に一時的な認証情報を作成します。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「ロールへの切り替え (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、長期的なアクセスキーを使用する代わりに、動的に一時的な認証情報を生成する AWS. AWS recommends にアクセスできます。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。

Route 53 ARC のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	はい

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場合 AWS、ユーザーはプリンシパルと見なされます。ポリシーは、プリンシパルに権限を付与します。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権限が必要です。

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

Route 53 ARC のサービスロール

サービスロールのサポート	いいえ

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 のサービスにアクセス許可を委任するロールの作成」を参照してください。

Route 53 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	はい

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、サービスにリンクされたロールの権限を表示できますが、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。

Amazon Route 53 Application Recovery Controller 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 Route 53 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受けることができます。

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 ポリシーの作成」を参照してください。

Route 53 ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認可リファレンス」の「Amazon Route 53 Recovery コントロールのアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーの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 Route 53 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 Route 53 ARC API アクション

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで Route 53 ARC リソースの作成、アクセス、削除を行える人を決めます。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 管理ポリシーの使用を開始し、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を付与するには、多くの一般的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して、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「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。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、最小特権権限とも呼ばれています。IAM を使用して権限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 でのポリシーと権限」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、 などの特定の を通じてサービスアクションを使用する場合 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 policy elements: Condition] (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 ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、『IAM ユーザーガイド』の「MFA 保護 API アクセスの設定」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 Route 53 ARC コンソールアクセス

Amazon Route 53 Application Recovery Controller コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。これらのアクセス許可により、 の Route 53 ARC リソースの詳細をリ

スト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場合でも、ユーザーとロールが Route 53 ARC コンソールを使用できるようにするには、Route 53 ARC のReadOnly AWS マネージド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 Route 53 ARC マネージドポリシーのページまたは「ユーザーへのアクセス許可の追加 (コンソール)」を参照してください。

コンソールから Route 53 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、Route 53 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ための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 Route 53 ARC API アクション

ユーザーが Route 53 ARC API アクションを使用して Route 53 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": "*"
        }
    ]
}
```

災害発生時にフェイルオーバーするようにルーティングコントロールの状態を更新するなど、リカバリクラスターデータプレーン API を使用して Route 53 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 Route 53 ARC IAM ポリシーを IAM ユーザーにアタッチします。

`AllowSafetyRuleOverride` ブール値は、ルーティングコントロールのセーフガードとして設定した安全ルールを、上書きするアクセス許可を付与します。このアクセス許可は、「Break Glass」のシナリオで、災害などの緊急のフェイルオーバーシナリオで安全対策を回避するために必要になる場合があります。例えば、オペレーターがディザスタリカバリのためにすばやいフェイルオーバーを必要とする場合や、1 つ以上の安全規則により、トラフィックの経路変更に必要なルーティングコントロール状態の更新が、予期せず妨げられる場合などです。このアクセス許可により、オペレーターは、API コールを行ってルーティングコントロールの状態を更新するときに、オーバーライドする安全ルールを指定できるようになります。詳細については、「安全ルールを上書きしてトラフィックを再ルーティングする」を参照してください。

オペレーターにリカバリクラスターのデータプレーン API の使用を許可し、安全ルールの上書きを防ぐには、 `AllowSafetyRuleOverrides` にブール値を使用して次のようなポリ

シーをアタッチできますfalse。オペレーターが安全ルールを上書きできるようにするには、AllowSafetyRuleOverrides ブール値を 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 Route 53 Application Recovery Controller 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るため、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するわけではないことに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシー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ているアクセス許可 AWS を更新すると、その更新は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 AWS のサービス が起動されたとき、または既存のサービスで新しい API オペレーションが使用可能になったときに、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、Route 53 ARC の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AmazonRoute「53RecoveryControlConfigFullAccess」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、Route 53 ARC の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、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AmazonRoute「53RecoveryControlConfigReadOnlyAccess」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、Route 53 ARC の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AmazonRoute「53RecoveryClusterFullAccess」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、Route 53 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AmazonRoute「53RecoveryClusterReadOnlyAccess」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

Route 53 ARC で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について、このサービスがこれらの変更の追跡を開始した以降のものについては、「」を参照してくださいAmazon Route 53 Application Recovery Controller の AWS マネージドポリシーの更新。このページの変更に関する自動通知を受け取るには、Route 53 ARC のドキュメントの履歴ページの RSS フィードにサブスクライブします。

ルーティングコントロールのクォータ

Amazon Route 53 Application Recovery Controller 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

エンティティ	クォータ
コントロールパネルあたりの安全ルールの数	20
UpdateRoutingControlStates オペレーションコール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

Amazon Route 53 Application Recovery Controller の準備状況チェック

Amazon Route 53 Application Recovery Controller 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を把握できます。Route 53 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ことを確認するのに役立ちます。

この章では、Route 53 ARC でアプリケーションをモデル化して、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することで、準備状況チェックが機能する構造を設定する方法について説明します。次に、Route 53 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の範囲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、容量を追加したり、設定を変更して、アプリケーションレプリカを再度調整したりできます。

> **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 Route 53 Application Recovery Controller の準備状況チェックとは

Route 53 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の不一致、サービスクォータ、スロットル制限、およびチェックに含まれるリソースの設定とバージョンの不一致を継続的に (1 分間隔で) 監査します。準備状況チェックではこれらの差異がユーザーに通知されるため、各レプリカの設定のセットアップが同じであり、ランタイム時の状態が同じであることを確認できます。準備状況チェックでは、設定したキャパシティがレプリカ間で一定であることを確認できますが、ユーザーに代わってレプリカのキャパシティを決めてくれると考えるべきではありません。例えば、別のセルが使用できなくなった場合に備えて、各レプリカの、十分なバッファ容量を備えた Auto Scaling グループのサイズを決めるには、アプリケーション要件を理解する必要があります。

クォータについては、Route 53 ARC が準備状況チェックで不一致を検出すると、高いクォータに合わせて低いクォータを増やすことで、レプリカのクォータを調整する措置を講じることができます。クォータが一致すると、準備状況チェックのステータスが READY と表示されます (こちらは即時の更新プロセスではありません。また、合計時間は特定のリソースタイプやその他の要因に応じて変わります)。

最初のステップでは、アプリケーションを表すリカバリグループを作成するための、準備状況チェックをセットアップします。各リカバリグループには、個々の障害抑制ユニットまたはアプリケーションのレプリカのセルが含まれています。次に、アプリケーション内のリソースタイプごとにリソースセットを作成し、そのリソースセットに準備状況チェックを関連付けます。最後に、リソースを準備状況の範囲に関連付けます。そうすることで、リカバリグループ (アプリケーション) または個々のセル (レプリカ、つまりリージョンまたはアベイラビリティーゾーン (AZ)) 内のリソースに関する準備状況ステータスを取得できます。

準備状況 (つまり READY または NOT READY) は、準備状況チェックの範囲に含まれるリソースと、リソースタイプの一連のルールに基づいて決定されます。リソースタイプごとに一連の準備状況ルールがあり、Route 53 ARC のチェックではこれを使ってリソースの準備状況を監査します。リソースが READY であるか否かの判断は、各準備状況ルールの定義方法に基づきます。準備状況ルールで

は、通常リソースの評価が行われますが、リソースを相互に比較したり、リソースセット内の各リソースに関する特定の情報を調べたりする場合もあります。

準備状況チェックを追加することで、準備状況ステータスをモニタリングできます。 を使用するEventBridge方法、 を使用する方法 AWS Management Console、Route 53 ARC API アクションを使用する方法のいずれかです。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。Route 53 ARC のクロスアカウント認証機能を使用すると、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および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

Route 53 ARC は、準備状況チェックを使用してアプリケーションのレプリカを監査し、各レプリカが同じ構成設定で同じランタイムを持つことを確認します。準備状況チェックでは、アプリケーションの AWS リソース容量、設定、 AWS クォータ、およびルーティングポリシーを継続的に監査します。この情報は、レプリカがフェイルオーバーの準備が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

復旧の準備を整えるには、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、レプリカで常に十分な予備の容量を維持する必要があります。Route 53 ARC はアプリケーションを継続的に (1 分ごと) 検査して、プロビジョニングされた容量がすべてのアベイラビリティーゾーンまたはリージョンにわたって一致していることを確認します。

Route 53 ARC が検査する容量には、例えば、Amazon EC2 インスタンス数、Aurora の読み取りおよび書き込みキャパシティユニット、Amazon EBS ボリュームサイズなどが含まれます。リソース値に合わせてプライマリレプリカの容量をスケールアップしたものの、スタンバイレプリカの対応する値も増やすことを忘れた場合、Route 53 ARC が不一致を検出するため、スタンバイの値を増やすことができます。

> **Important**
>
> 準備状況チェックは、アプリケーションのレプリカの設定とランタイムの状態が一致していることを継続的に確認するときに、最も役立つサービスです。準備状況チェックは、本番の

レプリカが正常かどうかを示すために使用すべきではありません。また、準備状況チェックを、災害発生時のフェイルオーバーの主要なトリガーとして使用すべきでもありません。

アクティブスタンバイ構成において、セルからまたはセルにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。Route 53 ARC の準備状況チェックは可用性が高くないため、システム停止中に準備状況チェックにアクセスできるとは限りません。さらに、チェックされたリソースは、災害時には利用できなくなる可能性もあります。

特定のセル (AWS リージョンまたはアベイラビリティーゾーン) のアプリケーションリソースの準備状況ステータス、またはアプリケーション全体の準備状況をモニタリングできます。準備状況チェックのステータスが などに変わったら、 でルールを作成`Not ready`することで通知を受け取ることができます EventBridge。詳細については、「Amazon での Route 53 ARC の準備状況チェックの使用 EventBridge」を参照してください。準備状況ステータスは AWS Management Console、 で表示することも、 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

Route 53 ARC は、準備状況チェックを使用してアプリケーションのレプリカを監査し、各レプリカが同じ構成設定で同じランタイムを持つことを確認します。

例えば、リカバリに備えるには、別のアベイラビリティーゾーンまたはリージョンからのフェイルオーバートラフィックを吸収できる十分な予備の容量を常に保持している必要があります。Route 53 ARC はアプリケーションを継続的に (1 分ごと) 検査して、プロビジョニングされた容量がすべてのアベイラビリティーゾーンまたはリージョンにわたって一致していることを確認します。Route 53 ARC が検査する容量には、例えば、Amazon EC2 インスタンス数、Aurora の読み取りおよび書き込みキャパシティユニット、Amazon EBS ボリュームサイズなどが含まれます。リソース値に合わせてプライマリレプリカの容量をスケールアップしたものの、スタンバイレプリカの対応する値も増やすことを忘れた場合、Route 53 ARC が不一致を検出するため、スタンバイの値を増やすことができます。

Important

準備状況チェックは、アプリケーションのレプリカの設定とランタイムの状態が一致していることを継続的に確認するときに、最も役立つサービスです。準備状況チェックは、本番の

レプリカが正常かどうかを示すために使用すべきではありません。また、準備状況チェックを、災害発生時のフェイルオーバーの主要なトリガーとして使用すべきでもありません。

アクティブスタンバイ構成において、セルからまたはセルにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。Route 53 ARC の準備状況チェックは可用性が高くないため、システム停止中に準備状況チェックにアクセスできるとは限りません。さらに、チェックされたリソースは、災害時には利用できなくなる可能性もあります。

特定のセル (AWS リージョンまたはアベイラビリティーゾーン) のアプリケーションリソースの準備状況ステータス、またはアプリケーション全体の準備状況をモニタリングできます。準備状況チェックのステータスが などに変わったら、 でルールを作成`Not ready`することで通知を受け取ることができます EventBridge。詳細については、「Amazon での Route 53 ARC の準備状況チェックの使用 EventBridge」を参照してください。準備状況ステータスは AWS Management Console、 で表示することも、 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

Route 53 ARC の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。Route 53 ARC には、サポートされているリソースのタイプごとに、1 つのルールグループが含まれています。例えば、Route 53 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含まれています。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、アラームの状態をモニタリングするための準備状況チェックを作成できます。例えば、Amazon EKS コンテナサービスをモニタリングするカスタム CloudWatch アラームを作成し、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できます。

リソースセットを作成する AWS Management Console ときに、 で各リソースタイプのすべての準備状況ルールを表示できます。または、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できます。準備状況ルールは「Route 53 ARC での準備状況ルール」セクションでも確認できます。

準備状況チェックで一連のルールを使って一連のリソースを監査する場合、各ルールの定義方法によって、すべてのリソースで結果を READY または NOT READY にするのか、それともリソースごとに結果を変えるのかが決まります。さらに、準備状況ステータスは複数の方法で表示できます。例えば、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。

また、Route 53 ARC では、クォータに関して追加の対策を講じています。準備状況チェックで、サポートされているリソースのサービスクォータ (リソースの作成とオペレーションの最大値) のセル

間に不一致が見つかった場合、Route 53 ARC は、クォータが低い方のリソースで、自動的にクォータを引き上げます。これは、クォータ (制限) に対してのみ適用されます。キャパシティに関しては、アプリケーションのニーズに応じて、ユーザーが必要なキャパシティを追加する必要があります。

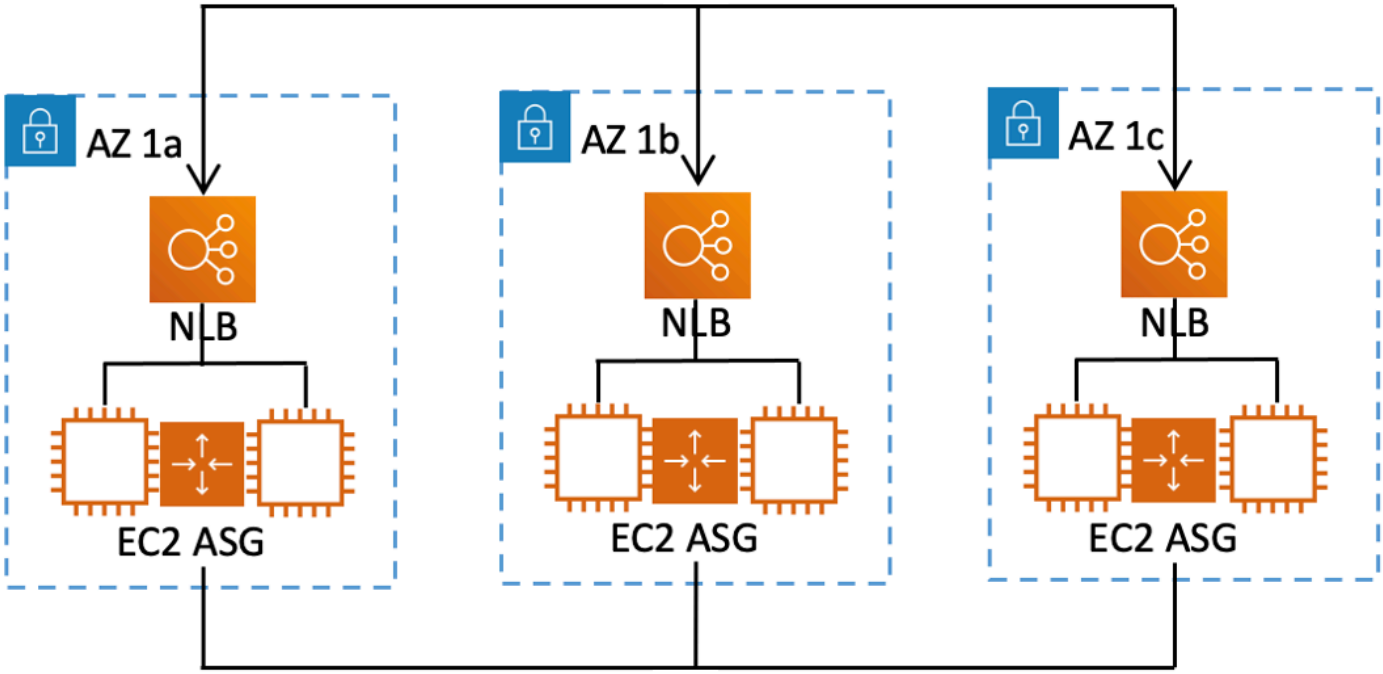
準備状況チェックのステータスが に変わった場合など、準備状況チェックの Amazon EventBridge 通知を設定することもできますNOT READY。次に、設定の不一致が検出されると、 から通知 EventBridge が送信され、修正アクションを実行して、アプリケーションレプリカが調整され、復旧の準備が整っていることを確認できます。詳細については、「Amazon での Route 53 ARC の準備状況チェックの使用 EventBridge」を参照してください。

準備状況チェック、リソースセット、準備状況の範囲が連携する方法

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。Route 53 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には、リソースセットを (別個に、または準備状況チェックの作成中に) 作成します。これにより、準備状況チェックを定義できます。リソースセットは、通常、同じ種類のリソース (Network Load Balancer など) から成るグループですが、アーキテクチャの準備状況をチェックする場合は DNS ターゲットリソースになる場合もあります。

通常、アプリケーション内のリソースの各タイプに、1 つのリソースセットと準備状況チェックを作成します。アーキテクチャの準備状況チェックでは、最上位の DNS ターゲットリソースとそれに対応するグローバルな (リカバリグループレベルの) リソースセットを作成し、続いて、別のリソースセット用にセルレベルの DNS ターゲットリソースを作成します。

次の図は、3 つのセル (アベイラビリティーゾーン) を持つリカバリグループの例です。各セルに Network Load Balancer (NLB) と Auto Scaling グループ (ASG) があります。



このシナリオでは、3 つの Network Load Balancer 用のリソースセットと準備状況チェック、3 つの Auto Scaling グループ用のリソースセットと準備状況チェックを作成します。これで、リカバリグループの各リソースセットで、リソースタイプごとに準備状況チェックを行えます。

リソースの準備状況の範囲を作成することで、セルまたはリカバリグループの準備状況チェックの概要を追加できます。リソースの準備状況の範囲を指定するには、セルまたはリカバリグループの ARN を、リソースセット内の各リソースに関連付けます。これは、リソースセットの準備状況チェックを作成する際に実行できます。

例えば、このリカバリグループにおける Network Load Balancer のリソースセットの準備状況チェックを追加すると、各 NLB に準備状況の範囲を同時に追加できます。この場合は、AZ 1a の ARN を AZ 1a の NLB に、AZ 1b の ARN を NLB AZ 1b に、AZ 1c の ARN を AZ 1c の NLB にそれぞれ関連付けます。Auto Scaling グループの準備状況チェックを作成するときも同じことを行い、Auto Scaling グループのリソースセットの準備状況チェックを作成するときに、準備状況の範囲をそれぞれに割り当てます。

準備状況チェックを作成するときに準備状況の範囲を関連付けるのは任意ですが、こちらを設定しておくことを強く推奨します。準備状況の範囲を設定しておくと、Route 53 ARC に、リカバリグループの準備状況チェックの概要と、セルレベルにおける準備状況チェックの概要の正確なステータスを READY または NOT READY で表示できます。準備状況の範囲を設定しないと、これらの概要を Route 53 ARC に表示できません。

アプリケーションレベルのリソースや、DNS ルーティングポリシーなどのグローバルなリソースを追加する場合、準備状況の範囲のリカバリグループやセルは選択しません。代わりに、グローバルリソース (セルなし) を選択します。

DNS ターゲットリソースの準備状況チェック: レジリエンシーの準備状況の監査

Route 53 ARC の DNS ターゲットリソースの準備状況チェックを使用すると、アプリケーションのアーキテクチャと障害耐性の準備状況を監査できます。このタイプの準備状況チェックでは、アプリケーションのアーキテクチャと Amazon Route 53 のルーティングポリシーを継続的にスキャンして、クロスゾーンおよびクロスリージョンの依存関係を監査します。

リカバリ指向のアプリケーションには、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た複数のレプリカがあるため、レプリカは互いに独立して失敗する可能性があります。正しくサイロ化するようにアプリケーションを調整する必要がある場合に、Route 53 ARC は、必要に応じてアーキテクチャを更新できる変更を提案します。これにより、レジリエンスとフェイルオーバーへの備えを確保できます。

Route 53 ARC は、アプリケーション内のセル (レプリカまたは障害抑制ユニットを表す) の数と範囲、およびセルがアベイラビリティーゾーンごとまたはリージョンごとにサイロ化されているかどうかを、自動的に検出します。次に、Route 53 ARC は、セル内のアプリケーションリソースを識別してユーザーに情報を提供し、それらがアベイラビリティーゾーンまたはリージョンに正しくサイロ化されているかどうかを判断します。例えば、特定のアベイラビリティーゾーンを対象とするセルがある場合、準備状況チェックでは、ロードバランサーとその背後にあるターゲットも、それらのゾーンにサイロ化されているかどうかをモニタリングできます。

この情報を使用することで、セル内のリソースを正しいゾーンまたはリージョンに一致させるために、変更すべきことがあるかどうかを判断できます。

開始するには、アプリケーション用の DNS ターゲットリソースと、それらのリソースセットおよび準備状況チェックを作成します。詳細については、「Route 53 ARC でアーキテクチャの推奨事項を取得する」を参照してください。

準備状況チェックとディザスタリカバリのシナリオ

Route 53 ARC の準備状況チェックでは、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を把握できます。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。

Route 53 ARC の準備状況チェックは、単一の AWS リージョン、つまり米国西部 (オレゴン) でホストされているため、停電時や災害時に、準備状況チェックの情報が古くなったりチェックを利用できなくなったりする場合がありますのでご注意ください。詳細については、「ルーティングコントロールのデータプレーンとコントロールプレーン」を参照してください。

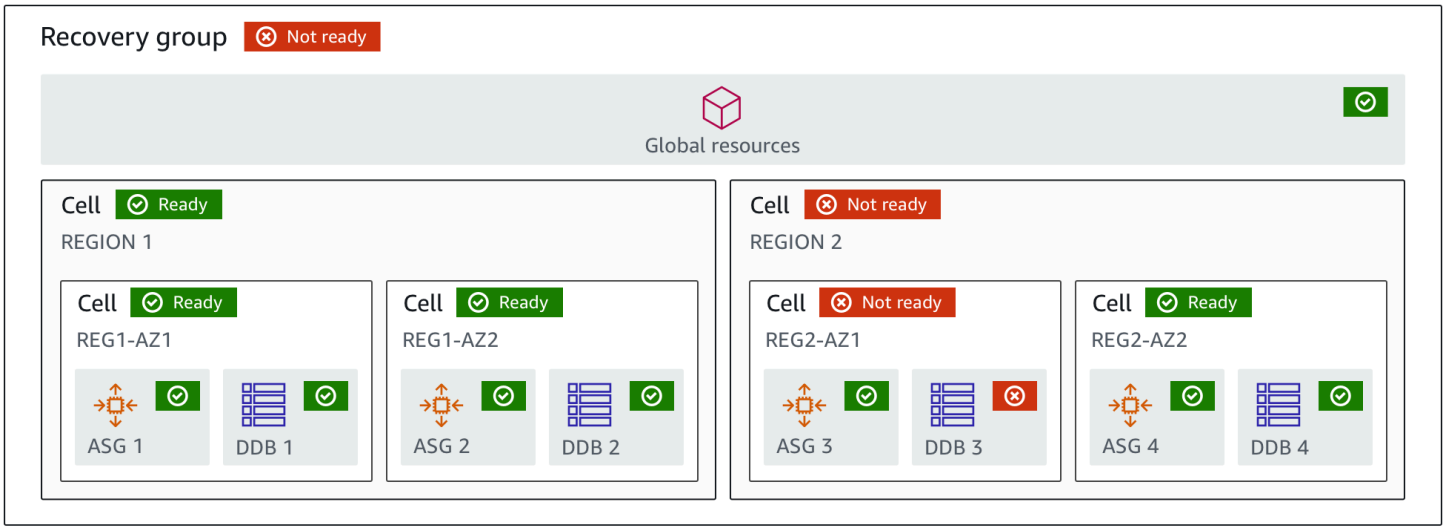
AWS 準備状況チェックのリージョンの可用性

Amazon Route 53 Application Recovery Controller のリージョンサポートとリージョンサービスエンドポイントに関する詳細は、Amazon Web Services 全般のリファレンスにある「Amazon Route 53 Application Recovery Controller のエンドポイントとクォータ」を参照してください。

> **Note**
>
> Amazon Route 53 Application Recovery Controller 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、リージョンの Route 53 ARC 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。 AWS CLI

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、Route 53 ARC における準備状況チェック機能のコンポーネントです。

セル

セルはアプリケーションのレプリカ、または独立したフェイルオーバーのユニットを定義します。アプリケーションがレプリカ内で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 間で複製されたウェブアプリケーションがあり、us-east-1b がフェイルオーバー環境である場合、Route 53 ARC ではこのアプリケーションを 2 つのセル (us-east-1a に 1 つ、us-east-1b に 1 つ) で構成されるリカバリグループとして表せます。リカバリグループには、Route 53 ヘルスチェックなどのグローバルリソースを含めることもできます。

リソースとリソース識別子

Route 53 ARC で準備状況チェックのコンポーネントを作成するときは、リソース識別子を使用して Amazon DynamoDB テーブル、Network Load Balancer、DNS ターゲットリソースなどのリソースを指定します。リソース識別子は、リソースの Amazon リソースネーム (ARN)、または

DNS ターゲットリソースの場合は Route 53 ARC がリソースを作成したときに生成した識別子のいずれかとなります。

DNS ターゲットリソース

DNS ターゲットリソースは、アプリケーションのドメイン名と、ドメインが指す AWS リソースなどの他の DNS 情報の組み合わせです。 AWS リソースを含めるのは任意ですが、含める場合は Route 53 リソースレコード、または Network Load Balancer でなければなりません。 AWS リソースを提供すると、アプリケーションの回復力を向上させるのに役立つ、より詳細なアーキテクチャの推奨事項を取得できます。Route 53 ARC で DNS ターゲットリソースのリソースセットを作成し、そのリソースセットの準備状況チェックを作成することで、アプリケーションに関するアーキテクチャの推奨事項を確認できます。準備状況チェックでは、DNS ターゲットリソースの準備状況ルールに基づいて、アプリケーションの DNS ルーティングポリシーも監視されます。

リソースセット

リソースセットは、複数のセルにまたがるリソース AWS または DNS ターゲットリソースを含む一連のリソースです。例えば、us-east-1a に 1 つのロードバランサーがあり、us-east-1b には別のロードバランサーがあります。ロードバランサーのリカバリの準備状況を監視するには、両方のロードバランサーを含むリソースセットを作成し、そのリソースセットの準備状況チェックを作成します。Route 53 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、Route 53 ARC がリソースセット内の一連のリソースに対して実施する監査です。Route 53 ARC には、準備状況チェックをサポートするリソースの種類ごとに準備状況ルールのセットがあります。各ルールには、Route 53 ARC がリソースを検査する目的を示す ID と説明が含まれています。

準備状況チェック

準備状況チェックは、Route 53 ARC がリカバリの準備状況を監査している Amazon Aurora インスタンスのセットなど、アプリケーション内のリソースセットをモニタリングします。準備状況チェックには、キャパシティ設定、 AWS クォータ、ルーティングポリシーなどの監査が含まれる場合があります。例えば、2 つのアベイラビリティーゾーンにまたがる Amazon EC2 Auto Scaling グループの準備状況を監査する場合、Auto Scaling グループごとに 1 つずつ、合計 2 つのリソース ARN を持つリソースセットの準備状況チェックを作成できます。そして、各グループが均等にスケールされるように、Route 53 ARC はその 2 グループのインスタンスタイプとインスタンス数を継続的に監視します。

準備状況の範囲

準備状況の範囲は、特定の準備状況チェックの対象となるリソースのグループを示します。準備状況チェックの範囲は、リカバリグループ (つまり、アプリケーション全体を対象とするグローバル) にすることも、セル (つまり、リージョンまたはアベイラビリティーゾーン) にすることもできます。Route 53 ARC のグローバルリソースの場合、準備状況の範囲をリカバリグループまたはグローバルリソースレベルに設定してください。例えば、Route 53 ヘルスチェックはリージョンやアベイラビリティーゾーンに固有のものではないため、Route 53 ARC のグローバルリソースとなります。

準備状況チェック用のデータ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、信頼性と耐障害性を最大限に高めるには、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対して、1 つの API である 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。準備状況チェックのコントロールプレーンとデータプレーンは信頼性がありますが、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 が高可用性の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 Route 53 Application Recovery Controller 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。

Route 53 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

Route 53 ARC でのタグ付けは、API を使用してのみ可能です。例えば、 AWS CLIを使用します。

以下は、 を使用した準備状況チェックでのタグ付けの例です 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 Route 53 Application Recovery Controller のTagResourceリカバリ準備 API リファレンスガイド」の「」を参照してください。

Route 53 ARC の準備状況チェックの料金

Amazon Route 53 Application Recovery Controller は、サービスで使用するために設定した分のみ料金が請求されます。準備状況チェックでは、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。

Route 53 ARC の料金情報と料金例の詳細については、「Amazon Route 53 Application Recovery Controller の料金」を参照し、Amazon Route 53 Application Recovery Controller までスクロールします。

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 AWS にあるアプリケーションで Amazon Route 53 Application Recovery Controller を使用するには、リカバリの準備状況を効果的にサポートできるように、アプリケーションの耐障害性を設定するためのガイドラインに従う必要があります。次に、アプリケーションの準備状況チェックを作成し、フェイルオーバー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、レジリエンシーを高めるアプリケーションのアーキテクチャについて、Route 53 ARC が提供する推奨事項を確認することもできます。

> **Note**
>
> アベイラビリティーゾーンによってサイロ化された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。ゾーンシフトや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、Route 53 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、「Amazon Route 53 Application Recovery Controller 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 Route 53 Application Recovery Controller のマルチリージョンリカバリの開始方法。

Route 53 ARC の準備状況チェックのベストプラクティス

Amazon Route 53 Application Recovery Controller の準備状況チェックには、次のベストプラクティスをお勧めします。

準備状況ステータスの変更の通知を追加する

準備状況チェックのステータスが から EventBridge などに変わるたびに通知を送信するREADYように Amazon のルールを設定しますNOT READY。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。

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する EventBridge ルールを設定できます。

詳細については、「Amazon での Route 53 ARC の準備状況チェックの使用 EventBridge」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

次の表は、リカバリ準備状況 (準備状況チェック) に使用できる Route 53 ARC オペレーションを、関連するドキュメントへのリンクと共に一覧にしたものです。

AWS Command Line Interfaceで一般的なリカバリ準備状況 API オペレーションを使用する方法の例については、「 で Route 53 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	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
セルを作成する	「 Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	CreateCell「��
セルを取得する	「 Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	GetCell「��
セルを削除する	「 Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	DeleteCell「��
セルを更新する	該当なし	UpdateCell「��
アカウントのセルを一覧表示する	「 Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	ListCells「��

アクション	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
リカバリグループを作成する	「Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	CreateRecovery「グループ」を参照
リカバリグループを取得する	「Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	GetRecovery「グループ」を参照
リカバリグループを更新する	「Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	UpdateRecovery「グループ」を参照
リカバリグループを削除する	「Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	DeleteRecovery「グループ」を参照
リカバリグループを一覧表示する	「Route 53 ARC でのリカバリグループの作成、更新、削除」を参照	ListRecovery「グループ」を参照
リソースセットを作成する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	CreateResource「Set」を参照
リソースセットを取得する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	GetResource「Set」を参照
リソースセットを更新する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	UpdateResource「Set」を参照
リソースセットを削除する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	DeleteResource「Set」を参照

アクション	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
リソースセットを一覧表示する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	ListResource「セット」を参照
準備状況チェックを作成する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	CreateReadiness「チェック」を参照してください
準備状況チェックを取得する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	GetReadiness「チェック」を参照
準備状況チェックを更新する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	UpdateReadiness「チェック」を参照してください
準備状況チェックを削除する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	DeleteReadiness「チェック」を参照
準備状況チェックを一覧表示する	「Route 53 ARC の準備状況チェックの作成と更新」を参照	ListReadiness「チェック」を参照
準備状況ルールを一覧表示する	「Route 53 ARC での準備状況ルールの説明」を参照	ListRules「��
準備状況チェック全体の状態をチェックする	「Route 53 ARC で準備状況ステータスをモニタリングする」を参照	GetReadinessCheckStatus「��
リソースの状態をチェックする	「Route 53 ARC で準備状況ステータスをモニタリングする」を参照	GetReadinessCheckResource「ステータス」を参照

アクション	Route 53 ARC コンソールを使用	Route 53 ARC API を使用
セルの状態をチェックする	「Route 53 ARC で準備状況ステータスをモニタリングする」を参照	GetCellReadinessSummary「��」
リカバリグループの状態をチェックする	「Route 53 ARC で準備状況ステータスをモニタリングする」を参照	GetRecoveryGroupReadiness「概要」を参照

で Route 53 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する Amazon Route 53 Application Recovery Controller の準備状況チェック機能 AWS Command Line Interface を操作する簡単なアプリケーション例について説明します。この例は、 CLI を使用して準備状況チェック機能を操作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。

Route 53 ARC の準備状況チェックでは、アプリケーションレプリカ内のリソースの不一致が監査されます。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 Route 53 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。

Route 53 ARC のセルは、障害の境界 (アベイラビリティーゾーンやリージョンなど) を表し、リカバリグループにまとめられます。リカバリグループとは、フェイルオーバーの準備状況を確認したい

アプリケーションのことです。準備状況チェックのコンポーネントの詳細については、「準備状況チェックのコンポーネント」を参照してください。

> **Note**
>
> Route 53 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 Route 53 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場合などです。

このアプリケーションの例では、まず、リソースがあるリージョンごとに 1 つのセルを作成します。次に、リカバリグループを作成し、準備状況チェックの設定を完了します。

1. セルを作成する

1a. us-east-1 セルを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name east-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
    "CellName": "east-cell",
    "Cells": [],
    "ParentReadinessScopes": [],
    "Tags": {}
}
```

1b. us-west-1 セルを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-cell \
    --cell-name west-cell
```

```
{
    "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell",
    "CellName": "west-cell",
    "Cells": [],
    "ParentReadinessScopes": [],
```

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

1c. これで 2 つのセルができました。list-cells API を呼び出して、それらが存在することを確認できます。

```
aws route53-recovery-readiness --region us-west-2 list-cells
```

```
{
    "Cells": [
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell",
            "CellName": "east-cell",
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        },
        {
            "CellArn": "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell",
            "CellName": "west-cell"
            "Cells": [],
            "ParentReadinessScopes": [],
            "Tags": {}
        }
    ]
}
```

2. リカバリグループを作成する

リカバリグループは、Route 53 ARC のリカバリの準備状況における最上位リソースです。リカバリグループはアプリケーション全体を表します。このステップでは、アプリケーション全体をモデル化するリカバリグループを作成し、作成した 2 つのセルを追加します。

2a. リカバリグループを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-recovery-group \
    --recovery-group-name simple-service-recovery-group \
       --cells "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"\
       "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
```

```
{
    "Cells": [],
    "RecoveryGroupArn": "arn:aws:route53-recovery-readiness::111122223333:recovery-
group/simple-service-recovery-group",
    "RecoveryGroupName": "simple-service-recovery-group",
    "Tags": {}
}
```

2b. (オプション) `list-recovery-groups` API を呼び出して、リカバリグループが正しく作成されたことを確認できます。

```
aws route53-recovery-readiness --region us-west-2 list-recovery-groups
```

```
{
    "RecoveryGroups": [
        {
            "Cells": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell",
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "RecoveryGroupArn": "arn:aws:route53-recovery-
readiness::111122223333:recovery-group/simple-service-recovery-group",
            "RecoveryGroupName": "simple-service-recovery-group",
            "Tags": {}
        }
    ]
}
```

アプリケーションのモデルができたので、モニタリングするリソースを追加しましょう。Route 53 ARC では、モニタリングしたいリソースのグループをリソースセットと呼びます。リソースセットには、すべて同じタイプのリソースが含まれています。リソースセット内のリソースを相互に比較して、セルのフェイルオーバー準備状況を判断します。

3. リソースセットを作成する

Simple-Service アプリケーションが本当にシンプルで、DynamoDB テーブルのみを使用していると仮定しましょう。us-east-1 に DynamoDB テーブルがあり、us-west-2 に別のテーブルがあります。リソースセットには、各リソースが含まれるセルを識別する準備状況の範囲も含まれています。

3a. Simple-Service アプリケーションのリソースを反映したリソースセットを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-resource-set \
    --resource-set-name ImportantInformationTables \
    --resource-set-type AWS::DynamoDB::Table \
    --resources
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
west-cell"
    ResourceArn="arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1",ReadinessScopes="arn:aws:route53-recovery-readiness::111122223333:cell/
east-cell"
```

```
{
    "ResourceSetArn": "arn:aws:route53-recovery-readiness::111122223333:resource-set/
sample-resource-set",
    "ResourceSetName": "ImportantInformationTables",
    "Resources": [
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/west-cell"
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "ReadinessScopes": [
                "arn:aws:route53-recovery-readiness::111122223333:cell/east-cell"
            ],
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
        }
    ],
    "Tags": {}
}
```

3b. (オプション) `list-resource-sets` API を呼び出すと、リソースセットに何が含まれているかを確認できます。これにより、 AWS アカウントのすべてのリソースセットが一覧表示されます。先ほど作成したリソースセットは 1 つだけであることがわかります。

```
aws route53-recovery-readiness --region us-west-2 list-resource-sets
```

```
{
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/east-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}{
    "ResourceSets": [
        {
            "ResourceSetArn": "arn:aws:route53-recovery-
readiness::111122223333:resource-set/ImportantInformationTables",
            "ResourceSetName": "ImportantInformationTables",
            "Resources": [
                {
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-readiness::111122223333:cell/west-
cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
                },
                {
```

```
                    "ReadinessScopes": [
                        "arn:aws:route53-recovery-
readiness::&ExampleAWSAccountNo1;:cell/east-cell"
                    ],
                    "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast1"
                }
            ],
            "Tags": {}
        }
    ]
}
```

これで、Route 53 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。Route 53 ARC の準備状況ルールすべてのリストを表示するには、`list-rules` を使用するか、または「Route 53 ARC での準備状況ルールの説明」を参照してください。Route 53 ARC には、リソースタイプごとに実行される特定のルールセットがあり、現時点ではカスタマイズできません。

4a. ImportantInformationTables というリソースセットの準備状況チェックを作成します。

```
aws route53-recovery-readiness --region us-west-2 create-readiness-check \
    --readiness-check-name ImportantInformationTableCheck --resource-set-name
 ImportantInformationTables
```

```
{
    "ReadinessCheckArn": "arn:aws:route53-recovery-readiness::111122223333:readiness-
check/ImportantInformationTableCheck",
    "ReadinessCheckName": "ImportantInformationTableCheck",
    "ResourceSet": "ImportantInformationTables",
    "Tags": {}
}
```

4b. (オプション) 準備状況チェックが正常に作成されたことを確認するには、list-readiness-checks API を実行します。この API は、アカウントのすべての準備状況チェックを表示します。

```
aws route53-recovery-readiness --region us-west-2 list-readiness-checks
```

```
{
    "ReadinessChecks": [
        {
            "ReadinessCheckArn": "arn:aws:route53-recovery-
readiness::111122223333:readiness-check/ImportantInformationTableCheck",
            "ReadinessCheckName": "ImportantInformationTableCheck",
            "ResourceSet": "ImportantInformationTables",
            "Tags": {}
        }
    ]
}
```

5. 準備状況チェックをモニタリングする

アプリケーションをモデル化し、準備状況チェックを追加したので、リソースをモニタリングする準備が整いました。アプリケーションの準備状況は次の 4 つのレベルでモデル化できます。準備状況チェックレベル (リソースのグループ)、個別のリソースレベル、セルレベル (アベイラビリティーゾーンまたはリージョン内のすべてのリソース)、リカバリグループレベル (アプリケーション全体) です。これらの各タイプの準備状況ステータスを取得するためのコマンドを次に示します。

5a. 準備状況チェックのステータスを確認します。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-status\
    --readiness-check-name ImportantInformationTableCheck
```

```
{
    "Readiness": "READY",
```

```
    "Resources": [
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:53:39Z",
            "Readiness": "READY",
            "ResourceArn": "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsEast2"
    ]
}
```

5b. チェックされた各ルールのステータスなど、準備状況チェックにおける単一のリソースの詳細な準備状況ステータスを確認します。

```
aws route53-recovery-readiness --region us-west-2 get-readiness-check-resource-status \
    --readiness-check-name ImportantInformationTableCheck \
    --resource-identifier "arn:aws:dynamodb:us-west-2:111122223333:table/
TableInUsWest2"
```

```
{"Readiness": "READY",
    "Rules": [
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoTableStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoPeakRcuWcu"
        },
```

```
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsPeakRcuWcu"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsConfig"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsStatus"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoGSIsCapacity"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoReplicationLatency"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoAutoScalingConfiguration"
        },
        {
            "LastCheckedTimestamp": "2021-01-07T00:55:41Z",
            "Messages": [],
            "Readiness": "READY",
            "RuleId": "DynamoLimits"
        }
    ]
```

```
}
```

5c. セルの全体的な準備状況を確認します。

```
aws route53-recovery-readiness --region us-west-2 get-cell-readiness-summary \
    --cell-name west-cell
```

```
{
    "Readiness": "READY",
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

5d. 最後に、リカバリグループレベルにおけるアプリケーションの最上位の準備状況を確認します。

```
aws route53-recovery-readiness --region us-west-2 get-recovery-group-readiness-summary
 \
    --recovery-group-name simple-service-recovery-group
```

```
{
    "Readiness": "READY",
    "ReadinessChecks": [
        {
            "Readiness": "READY",
            "ReadinessCheckName": "ImportantTableCheck"
        }
    ]
}
```

リカバリグループと準備状況チェックの使用

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、これらのリソースの作成、更新、削除を含めます。

Route 53 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 Route 53 Application Recovery Controller にあるアプリケーションを表します。通常は、リソースと機能の点から互いにレプリカとなる 2 つ以上のセルで構成されているため、一方のセルからもう一方のセルにフェイルオーバーできます。各セルには、1 つの AWS リージョンまたはアベイラビリティーゾーンのアクティブなリソースの Amazon 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。Route 53 ARC の準備状況チェックは、アプリケーションが、一方のレプリカから他方のレプリカにフェイルオーバーする準備ができているかどうかを判断するときに役立ちます。ただし、レプリカからまたはレプリカにフェイルオーバーするかどうかは、モニタリングのシステムやヘルスチェックのシステムに基づいてユーザーが判断する必要があります。準備状況チェックは、それらのシステムを補完するサービスとして捉えるのがよいでしょう。

準備状況チェックでは、リソースを監査して、そのタイプのリソースに対する事前定義された一連のルールに基づいて準備状況を判断します。レプリカを含むリカバリグループを作成したら、アプリケーション内のリソースの Route 53 ARC 準備状況チェックを追加します。これにより、Route 53 ARC は、レプリカの設定と構成が経時的に同じであることを確認できます。

トピック

- リカバリグループの作成
- リカバリグループとセルの更新および削除

リカバリグループの作成

このセクションでは、Route 53 ARC コンソールでリカバリグループを作成する手順について説明します。Amazon Route 53 Application Recovery Controller で リカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リカバリグループを作成するには

1. で Route 53 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

このセクションでは、Route 53 ARC コンソールでリカバリグループを更新および削除する手順、ならびにセルを削除する手順について説明します。Amazon Route 53 Application Recovery Controller で リカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リカバリグループを更新または削除し、セルを削除するには

1. で Route 53 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。

Route 53 ARC 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

このセクションでは、Route 53 ARC コンソールで準備状況チェックを作成する手順について説明します。Amazon Route 53 Application Recovery Controller で リカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

準備状況チェックは、準備状況チェックのリソースセットを編集してリソースを追加または削除するか、リソースの準備状況の範囲を変更することで、更新できます。

準備状況チェックを作成するには

1. で Route 53 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。以下のうちのひとつを選択します。

   - 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**
>
> Route 53 ARC が各リソースタイプに期待する ARN 形式の例および詳細については、「Route 53 ARC のリソースタイプと ARN フォーマット」を参照してください。

7. 必要に応じて、Route 53 ARC が、この準備状況チェックに追加したリソースのタイプをチェックする際に使用する、準備状況ルールを確認します。次いで、[次へ] を選択します。
8. (オプション) [リカバリグループ名] で、準備状況チェックを関連付けるリカバリグループを選択し、リソース ARN ごとに、そのリソースが含まれているドロップダウンメニューからセル (リージョンまたはアベイラビリティーゾーン) を選択します。リソースが、DNS ルーティングポリシーなどアプリケーションレベルのリソースである場合は、[グローバルリソース (セルなし)] を選択します。

   これにより、準備状況チェックにおけるリソースの準備状況の範囲が指定されます。

   > **Important**
   >
   > この手順はオプションですが、リカバリグループとセルの準備状況に関する情報の概要を手に入れるには、準備状況の範囲を追加する必要があります。この手順を飛ばし、ここで準備状況の範囲を選択して準備状況チェックをリカバリグループのリソースに関連付けなければ、Route 53 ARC は、リカバリグループまたはセルの準備状況に関する情報の概要を返せません。

9. [次へ] をクリックします。
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 Route 53 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。このセクションでは、Route 53 ARC コンソールでリソースセットを作成または編集する手順について説明します。Amazon Route 53 Application Recovery Controller で リカバリの準備状況の API オペレーションを使用する方法については、「準備状況チェック API オペレーション」を参照してください。

リソースセットを作成するには

1. https://console.aws.amazon.com/route53/home で Route 53 コンソールを開きます。
2. [アプリケーションリカバリコントローラー] で [リソースセット] を選択します。
3. [作成] を選択します。
4. リソースセットの名前を入力し、このセットに含めるリソースのタイプを選択します。
5. [追加] をクリックし、セットに追加するリソースの Amazon リソースネーム (ARN) を入力します。
6. リソースを追加したら、[リソースセットを作成] を選択します。

リソースセットを編集するには

1. で Route 53 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. [保存] を選択します。

Route 53 ARC で準備状況ステータスをモニタリングする

Amazon Route 53 Application Recovery Controller では、アプリケーションの準備状況を以下の各レベルで確認できます。

- リソースセット内のリソースの準備状況チェックレベル
- 個々のリソースレベル
- アベイラビリティーゾーンまたは AWS リージョン内のすべてのリソースのセル (アプリケーションレプリカ) レベル

- アプリケーション全体のリカバリグループレベル

準備状況ステータスの変化について通知を受けたり、Route 53 コンソールから、または Route 53 ARC CLI コマンドを使用して、準備状況ステータスの変化をモニタリングしたりできます。

準備状況ステータスの通知

Amazon を使用して EventBridge 、Route 53 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 での Route 53 ARC の準備状況チェックの使用 EventBridge」を参照してください。

Route 53 ARC コンソールで準備状況ステータスをモニタリングする

次の手順では、 でリカバリの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 Route 53 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。例:

```
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
```

Route 53 ARC でアーキテクチャの推奨事項を取得する

既存のアプリケーションがある場合、Amazon Route 53 Application Recovery Controller は、アプリケーションのアーキテクチャとルーティングポリシーを評価して、アプリケーションのレジリエンシーを高める設計に変更するための、推奨事項を提供します。アプリケーションを表すリカバリグループを Route 53 ARC に作成したら、このセクションの手順に従って、アプリケーションのアーキテクチャに関する推奨事項を取得します。

リカバリグループの DNS ターゲットリソースをまだ指定していない場合は、指定することが推奨されます。そうすれば、より詳細な推奨事項を取得できます。追加情報を提供すると、Route 53 ARC はユーザーにより適した推奨事項を提供できるようになります。例えば、Amazon Route 53 リソースレコードまたは Network Load Balancer をターゲットリソースとして入力すると、Route 53 ARC は、リカバリグループに作成されたセルの数が最適かどうかについての情報を提供できます。

DNS ターゲットリソースについては、次の点に注意してください。

- ターゲットリソースには、Route 53 リソースレコードまたは Network Load Balancer のみを指定します。
- 各リカバリグループには DNS ターゲットリソースを 1 つだけ作成します。
- 推奨: 各セルには DNS ターゲットリソースを 1 つ作成します。
- DNS ターゲットリソースを、準備状況チェックを行う 1 つのリソースセットにグループ化します。

次の手順では、DNS ターゲットリソースの作成方法と、アプリケーションのアーキテクチャの推奨事項を取得する方法について説明します。

アーキテクチャの更新に関する推奨事項を取得するには

1. で Route 53 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作成します。そうすれば、Route 53 ARC からアーキテクチャの推奨事項を取得できるようになります。[DNS ターゲットリソースの作成] を選択します。

   DNS ターゲットリソースの詳細については、「準備状況チェックのコンポーネント」を参照してください。
6. DNS ターゲットリソースのリソースセットを作成するには、準備状況チェックを作成します。準備状況チェックの名前を入力し、準備状況チェックのタイプとして [DNS ターゲットリソース] を選択します。
7. リソースセットの名前を入力します。
8. DNS 名、ホストゾーン ARN、レコードセット ID など、アプリケーションの属性を入力します。

   > **Tip**
   >
   > ホストゾーン ARN の形式を確認するには、「Route 53 ARC のリソースタイプと ARN フォーマット」でホストゾーンの ARN 形式を参照します。

   こちらはオプションですが、[オプションの属性を追加] を選択して、Network Load Balancer ARN またはドメインの Route 53 リソースレコードを指定することが強く推奨されます。
9. (オプション) [リカバリグループの設定] で、DNS ターゲットリソースのセルを選択し、準備状況の範囲を設定します。
10. [Create resource set] (リソースセットの作成) を選択します。

11. [リカバリグループ] の詳細ページで、[アーキテクチャの推奨事項の取得] を選択します。Route 53 ARC で、ページ上に一連の推奨事項が表示されます。

推奨事項のリストを確認します。その後、アプリケーションのレジリエンスを高めるための変更を加えるかどうか、また、どのように変更を加えるかを決定できます。

Route 53 ARC でのクロスアカウント認証の作成

リソースを複数のアカウントに分散させると AWS 、アプリケーションの状態を包括的に把握することが困難になる場合があります。また、迅速な意思決定に必要な情報の取得が難しくなることもあります。Amazon Route 53 Application Recovery Controller の準備状況チェックのためにこれを合理化するには、クロスアカウント認証 を使用できます。

Route 53 ARC のクロスアカウント認証は、準備状況チェック機能と連携します。クロスアカウント認証を使用すると、1 つの中央 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証の設定は、コンソールでは利用できません。代わりに、Route 53 ARC API オペレーションを使用して、クロスアカウント認証を設定し操作します。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。Route 53 ARC は、クロスアカウント認証を使用することで、1 つのアカウント us-west-2 から両方のリソースセットをモニタリングするためのアクセスを許可できます。

例えば、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証を有効にできます。認証を作成すると、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証を設定する方法を示したものです。Route 53 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証を有効にする必要があります。

> **Note**
>
> Route 53 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 Route 53 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。

次の AWS CLI コマンドは、この例のクロスアカウント認証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
```

この認証を無効にするには、次の手順を実行します。

```
aws route53-recovery-readiness --region us-west-2 --profile profile-in-us-east-1-
account \
    delete-cross-account-authorization --cross-account-authorization
 arn:aws:iam::999999999999:root
```

クロスアカウント認証を指定したすべてのアカウントで、特定のアカウントにチェックインするには、`list-cross-account-authorizations` コマンドを使用します。現時点では、反対方向にチェックインできません。つまり、リソースを追加およびモニタリングするためのクロスアカウント認証が付与されている、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

このセクションには、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形式に関するリファレンス情報が含まれています。

Route 53 ARC での準備状況ルールの説明

このセクションでは、Amazon Route 53 Application Recovery Controller でサポートされているすべての種類のリソースに関する、準備状況ルールの説明を一覧にしています。Route 53 ARC でサポートされているリソースタイプのリストについては、「Route 53 ARC のリソースタイプと ARN フォーマット」を参照してください。

準備状況ルールの説明は、Route 53 ARC コンソールで、または API オペレーションを使用して表示することもできます。方法は以下のとおりです。

- コンソールで準備状況ルールを表示するには、「コンソールに準備状況ルールを表示する」の手順に従います。
- API を使用して準備状況ルールを表示するには、 ListRulesオペレーションを参照してください。

トピック

- Route 53 ARC での準備状況ルール
- コンソールに準備状況ルールを表示する

Route 53 ARC での準備状況ルール

このセクションでは、Route 53 ARC でサポートされている各リソースタイプの、一連の準備状況ルールを一覧にしています。

ルールの説明を見ると、ほとんどのルールで「すべての ~ を検査」または「各 ~ を検査」という文言が使われていることがわかります。これらの文言が、準備状況チェックのコンテキストで、ルールの機能をどのように説明しているのかについて、また、Route 53 ARC が準備状況ステータスを設定する方法に関するその他の詳細については、「準備状況ルールが準備状況ステータスを判断する仕組み」を参照してください。

準備状況ルール

Route 53 ARC は、以下の準備状況ルールを使用してリソースをモニタリングします。

Amazon API Gateway バージョン 1 ステージ

- ApiGwV1ApiKeyCount: すべての API Gateway ステージを検査し、それらにリンクされている API キーの数が同数であることを確認します。
- ApiGwV1ApiKeySource: すべての API Gateway ステージを検査し、それらの `API Key Source` の値が同じであることを確認します。
- ApiGwV1BasePath: すべての API Gateway ステージを検査し、それらが同じベースパスにリンクされていることを確認します。
- ApiGwV1BinaryMediaTypes: すべての API Gateway ステージを検査し、それらが同じバイナリメディアタイプをサポートしていることを確認します。
- ApiGwV1CacheClusterEnabled: すべての API Gateway ステージを検査し、それらのすべてで `Cache Cluster` が有効になっているか、すべてで無効になっていることを確認します。
- ApiGwV1CacheClusterSize: すべての API Gateway ステージを検査し、それらの `Cache Cluster Size` が同じであることを確認します。いずれかの値がこれよりも大きいと、それ以外は NOT READY と表示されます。
- ApiGwV1CacheClusterStatus: すべての API Gateway ステージを検査し、`Cache Cluster` が AVAILABLE の状態になっていることを確認します。
- ApiGwV1DisableExecuteApiEndpoint: すべての API Gateway ステージを検査し、すべてで `Execute API Endpoint` が無効になっているか、いずれも無効になっていないことを確認します。
- ApiGwV1DomainName: すべての API Gateway ステージを検査し、同じドメイン名にリンクされていることを確認します。
- ApiGwV1EndpointConfiguration: すべての API Gateway ステージを検査し、同じエンドポイント設定でドメインにリンクされていることを確認します。
- ApiGwV1EndpointDomainNameStatus: すべての API Gateway ステージを検査し、それらにリンクしているドメイン名が AVAILABLE の状態であることを確認します。

- ApiGwV1MethodSettings: すべての API Gateway ステージを検査し、それらの `Method Settings` の値が同じであることを確認します。
- ApiGwV1MutualTlsAuthentication: すべての API Gateway ステージを検査し、それらの `Mutual TLS Authentication` の値が同じであることを確認します。
- ApiGwV1Policy: すべての API Gateway ステージを検査し、それらのすべてで API レベルのポリシーが使用されているか、またはすべてで使用されていないことを確認します。
- ApiGwV1RegionalDomainName: すべての API Gateway ステージを検査し、それらが同じリージョンのドメイン名にリンクされていることを確認します。注: このルールは準備状況ステータスには影響しません。
- ApiGwV1ResourceMethodConfigs: すべての API Gateway ステージを検査し、それらが、関連する設定を含め、同様のリソースの階層を持っていることを確認します。
- ApiGwV1SecurityPolicy: すべての API Gateway ステージを検査し、それらの `Security Policy` の値が同じであることを確認します。
- ApiGwV1Quotas: すべての API Gateway グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。
- ApiGwV1UsagePlans: すべての API Gateway ステージを検査し、それらが同じ設定で `Usage Plans` にリンクされていることを確認します。

Amazon API Gateway バージョン 2 ステージ

- ApiGwV2ApiKeySelectionExpression: すべての API Gateway ステージを検査し、それらの `API Key Selection Expression` の値が同じであることを確認します。
- ApiGwV2ApiMappingSelectionExpression: すべての API Gateway ステージを検査し、それらの `API Mapping Selection Expression` の値が同じであることを確認します。
- ApiGwV2CorsConfiguration: すべての API Gateway ステージを検査し、それらの CORS 関連の設定が同じであることを確認します。
- ApiGwV2DomainName: すべての API Gateway ステージを検査し、それらが同じドメイン名にリンクされていることを確認します。
- ApiGwV2DomainNameStatus: すべての API Gateway ステージを検査し、ドメイン名が AVAILABLE の状態になっていることを確認します。
- ApiGwV2EndpointType: すべての API Gateway ステージを検査し、それらの `Endpoint Type` の値が同じであることを確認します。
- ApiGwV2Quotas: すべての API Gateway グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

- ApiGwV2MutualTlsAuthentication: すべての API Gateway ステージを検査し、それらの `Mutual TLS Authentication` の値が同じであることを確認します。
- ApiGwV2ProtocolType: すべての API Gateway ステージを検査し、それらの `Protocol Type` の値が同じであることを確認します。
- ApiGwV2RouteConfigs: すべての API Gateway ステージを検査し、それらが同じ設定の、同じルートの階層を持つことを確認します。
- ApiGwV2RouteSelectionExpression: すべての API Gateway ステージを検査し、それらの `Route Selection Expression` の値が同じであることを確認します。
- ApiGwV2RouteSettings: すべての API Gateway ステージを検査し、それらの `Default Route Settings` の値が同じであることを確認します。
- ApiGwV2SecurityPolicy: すべての API Gateway ステージを検査し、それらの `Security Policy` の値が同じであることを確認します。
- ApiGwV2StageVariables: すべての API Gateway ステージを検査し、それらのすべてが他のステージと同じ `Stage Variables` を持っていることを確認します。
- ApiGwV2ThrottlingBurstLimit: すべての API Gateway ステージを検査し、それらの `Throttling Burst Limit` の値が同じであることを確認します。
- ApiGwV2ThrottlingRateLimit: すべての API Gateway ステージを検査し、それらの `Throttling Rate Limit` の値が同じであることを確認します。

Amazon Aurora クラスター

- RdsClusterStatus: 各 Aurora クラスターを検査し、ステータスが `AVAILABLE` または `BACKING-UP` であることを確認します。
- RdsEngineMode: すべての Aurora クラスターを検査し、それらの `Engine Mode` の値が同じであることを確認します。
- RdsEngineVersion: すべての Aurora クラスターを検査し、それらの `Major Version` の値が同じであることを確認します。
- RdsGlobalReplicaLag:各 Aurora クラスターを検査し、`Global Replica Lag` が 30 秒未満であることを確認します。
- RdsNormalizedCapacity: すべての Aurora クラスターを検査し、それらの正規化された容量が、リソースセットの最大容量の 15% 以内であることを確認します。
- RdsInstanceType: すべての Aurora クラスターを検査し、それらのインスタンスタイプが同じであることを確認します。
- RdsQuotas: すべての Aurora クラスターを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

「Auto Scaling グループ」

- AsgMinSizeAndMaxSize: すべての Auto Scaling グループを検査し、それらの最小グループのサイズと最大グループのサイズが同じであることを確認します。
- AsgAZCount: すべての Auto Scaling グループを検査し、それらのアベイラビリティーゾーンの数が同じであることを確認します。
- AsgInstanceTypes: すべての Auto Scaling グループを検査し、それらのインスタンスタイプが同じであることを確認します。注: このルールは準備状況ステータスには影響しません。
- AsgInstanceSizes: すべての Auto Scaling グループを検査し、それらのインスタンスのサイズが同じであることを確認します。
- AsgNormalizedCapacity: すべての Auto Scaling グループを検査し、それらの正規化された容量が、リソースセットの最大容量の 15% 以内であることを確認します。
- AsgQuotas: すべての Auto Scaling グループを検査し、それらが、Service Quotas が管理するクォータ (制限) に従っていることを確認します。

CloudWatch アラーム

- CloudWatchAlarmState: CloudWatch アラームを検査し、各アラームが `ALARM`または `INSUFFICIENT_DATA`状態になってい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証証明書 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 アラームの設定と設定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 Route 53 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。

Route 53 ARC のリソースタイプと ARN フォーマット

Amazon Route 53 Application Recovery Controller でリソースセットを作成するときは、セットに含めるリソースのタイプと、含める各リソースの Amazon リソースネーム (ARN) を指定します。Route 53 ARC では、リソースタイプごとに特定の ARN 形式が想定されています。このセクションでは、Route 53 ARC でサポートされているリソースタイプと、各リソースタイプに関連付けられた ARN 形式を一覧にします。

具体的な形式はリソースによって異なります。ARN を指定するには、*######*のテキストを、リソース固有の情報に置き換えます。

> **Note**
>
> Route 53 ARC がリソースで必要とする ARN 形式は、サービス自体がそのリソースで必要とする ARN 形式とは異なる場合がありますのでご注意ください。例えば、サービス認証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、Route 53 ARC が Route 53 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 で定義されるリソースタイプ CloudWatch」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。Route 53 ARC がリソースをポーリングできるようにするにはアカウント ID が必要です。この形式が、「サービス認可リファレンス」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 Route 53 Application Recovery Controller の準備状況チェックのログ記録とモニタリング

Amazon Route 53 Application Recovery Controller の準備状況チェック EventBridge のモニタリングに Amazon CloudWatch AWS CloudTrail、、および Amazon を使用して、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**
>
> Route 53 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

- Route 53 ARC CloudWatch の準備状況チェックでの Amazon 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 での Route 53 ARC の準備状況チェックの使用 EventBridge

Route 53 ARC CloudWatch の準備状況チェックでの Amazon の使用

Amazon Route 53 Application Recovery Controller は、準備状況チェック CloudWatch のために Amazon にデータポイントを発行します。 CloudWatch では、これらのデータポイントに関する統計を、メトリクス と呼ばれる時系列データの順序付けられた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。例えば、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った場合に、指定されたメトリクスをモニタリングし、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 CloudWatch アラームを作成できます。

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

- Route 53 ARC のメトリクス
- Route 53 ARC メトリクスの統計
- Route 53 ARC で CloudWatch メトリクスを表示する

Route 53 ARC のメトリクス

`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`	Route 53 ARC によって処理された準備状況のチェックの数を表します。このメトリクスは、以下に示すように状態別にディメンション化できます。 単位: `Count`

メトリクス	説明
	レポート条件: ゼロ以外の値がある。 統計: 使用できる統計は Sum のみです。 ディメンション • READY • NOT_READY • NOT_AUTHORIZED • UNKNOWN
Resources	Route 53 ARC によって処理されるリソースの数を表し、API で定義されているリソース識別子によってディメンション化できます。 単位: Count レポート条件: ゼロ以外の値がある。 統計: 使用できる統計は Sum のみです。 ディメンション • ResourceSetType ：これらはリソースタイプであり、Route 53 ARC によって評価された特定のタイプごとに、リソース数でフィルタリングされます。 例：AWS::CloudWatch::Alarm

Route 53 ARC メトリクスの統計

CloudWatch は、Route 53 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いて統計を提供します。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。

• Route 53 ARC によって準備状況が評価された準備状況チェックの数を表します。

- Route 53 ARC によって評価された、特定のリソースセットタイプの合計リソース数を表します。

Route 53 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、Route 53 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。

Route 53 ARC の CloudWatch メトリクスは、米国西部 (オレゴン) リージョンで、コンソールまたはを使用する場合に表示する必要があります AWS CLI。を使用する場合は AWS CLI、次のパラメータを指定して、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. https://console.aws.amazon.com/cloudwatch/ で CloudWatch コンソールを開きます。
2. ナビゲーションペインで 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

次のget-metric-statisticsコマンドを使用して、指定したメトリクスとディメンションの統計を取得します。は、ディメンションの一意の各組み合わせを個別のメトリクスとして CloudWatch 扱うことに注意してください。発行されていないディメンションの組み合わせを使用した統計を取得することはできません。メトリクス作成時に使用した同じディメンションを指定する必要があります。

次の例は、Route 53 ARC のアカウントについて、1 分ごとに評価された準備状況チェックの合計を一覧表示したものです。

```
aws cloudwatch get-metric-statistics --namespace AWS/Route53RecoveryReadiness \
--metric-name ReadinessChecks \
--region us-west-2 \
```

```
--statistics Sum --period 60 \
--dimensions Name=State,Value=READY \
--start-time 2021-07-03T01:00:00Z --end-time 2021-07-03T01:20:00Z
```

以下は、コマンドからの出力例です。

```
{
    "Label": "ReadinessChecks",
    "Datapoints": [
        {
            "Timestamp": "2021-07-08T18:00:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:04:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:01:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:02:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2021-07-08T18:03:00Z",
            "Sum": 1.0,
            "Unit": "Count"
        }
    ]
}
```

を使用した準備状況チェック API コールのログ記録 AWS CloudTrail

Amazon Route 53 Application Recovery Controller は AWS CloudTrail、Route 53 ARC のユーザー、ロール、または AWS サービスによって実行されたアクションを記録するサービスである と統合されています。 は、Route 53 ARC のすべての API コールをイベントとして CloudTrail キャプチャ

します。キャプチャされたコールには、Route 53 ARC コンソールからのコールと、Route 53 ARC API オペレーションへのコードコールが含まれます。

証跡を作成する場合は、Route 53 ARC の CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。 Amazon S3 Route 53 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。

で収集された情報を使用して CloudTrail、Route 53 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。

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

の Route 53 ARC 情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 は で有効になります。Route 53 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 CloudTrail イベント履歴 の他の AWS サービスイベントとともに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 「イベント履歴の使用」を参照してください。

Route 53 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、 はログファイル CloudTrail 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口グに記録し、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 ログで収集されたデータをより詳細に分析し、それに基づく対応を行うように他の 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:

- 「証跡作成の概要」
- CloudTrail でサポートされているサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail 口グファイルの受信と複数のアカウントからの CloudTrail 口グファイルの受信

すべての Route 53 ARC アクションは によって口グに記録 CloudTrail され、Amazon Route 53 Application Recovery Controller のリカバリ準備 API リファレンスガイド、Amazon Route 53 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 Route 53 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、、、および `CreateRecoveryGroup`アクションを呼び出

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

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

イベント履歴での Route 53 ARC イベントの表示

CloudTrail では、イベント履歴 で最近のイベントを表示できます。Route 53 ARC API リクエストのイベントを表示するには、コンソールの上部にあるリージョンセレクターで [米国西部 (オレゴン)] を指定する必要があります。詳細については、「 ユーザーガイド」の CloudTrail 「イベント履歴の使用AWS CloudTrail 」を参照してください。

Route 53 ARC のログファイルエントリを理解する

証跡は、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できるようにする設定です。 CloudTrail ログファイルには、1 つ以上のログエントリが含まれます。イベントは任意の送信元からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。 CloudTrail ログファイルは、パブリック API コール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": {
                "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 での Route 53 ARC の準備状況チェックの使用 EventBridge

Amazon を使用すると EventBridge、Amazon Route 53 Application Recovery Controller 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。例えば、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**
>
> Route 53 ARC は、米国西部 (オレゴン) (us-west-2) AWS リージョンでのみ準備状況チェックの EventBridge イベントを発行します。準備状況チェックの EventBridge イベントを受信するには、米国西部 (オレゴン) リージョンで EventBridge ルールを作成します。

Amazon でルールを作成して EventBridge 、次の Route 53 ARC 準備状況チェックイベントに対応できます。

- 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 Route 53 ARC イベントをキャプチャするには、イベントの検出 EventBridge 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、Route 53 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と連携する方法については、「」の「イベントとイベントパターン EventBridge」を参照してください。

で準備状況チェックリソースをモニタリングする EventBridge

では EventBridge、Route 53 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。

イベントパターンを入力またはコピーして EventBridge コンソールに貼り付けるには、 コンソールで、「自分のオプションを入力」オプションを選択します。このトピックでは、役立つ可能性のあるイベントパターンを特定できるように、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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
2. AWS リージョン でルールを作成するには、米国西部 (オレゴン) を選択します。これは準備状況イベントに必要なリージョンです。
3. [Create rule] を選択します。
4. ルールの [Name (名前)] を入力し、必要に応じて説明を入力します。
5. [イベントバス] については、デフォルト値の [デフォルト] のままにします。
6. [次へ] をクリックします。
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 に貼り付け EventBridge ると、Route 53 ARC の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、Route 53 ARC の準備状況チェック機能 EventBridge に で使用できる例を示しています。 Route 53

- Route 53 ARC の準備状況チェックからすべてのイベントを選択します。

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ]
}
```

- セルに関連するイベントのみを選択します。

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ]
}
```

- MyExampleCell という特定のセルに関連するイベントのみを選択します。

```
{
    "source": [
        "aws.route53-recovery-readiness"
    ],
    "detail-type": [
        "Route 53 Application Recovery Controller cell readiness status change"
    ],
    "resources": [
        "arn:aws:route53-recovery-readiness::111122223333:cell/MyExampleCell"
    ]
}
```

- リカバリグループ、セル、NOT READY のステータスとなった準備状況チェックのいずれかのイベントのみを選択します。

```
{
   "source":[
      "aws.route53-recovery-readiness"
   ],
   "detail-type":{
      "new-state":{
         "readiness-status":[
            "NOT_READY"
         ]
      }
   }
}
```

- リカバリグループ、セル、READY 以外のステータスになった準備状況チェックのいずれかのイベントのみを選択します。

```
{
   "source":[
      "aws.route53-recovery-readiness"
   ],
   "detail":{
      "new-state":{
         "readiness-status":[
            {
               "anything-but":"READY"
            }
         ]
      }
   }
}
```

以下は、リカバリグループの準備状況ステータス変更 の Route 53 ARC イベントの例です。

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller recovery group readiness
 status change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:recovery-group/BillingApp"
    ],
    "detail": {
        "recovery-group-name": "BillingApp",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

以下は、セルの準備状況ステータスを変更するための Route 53 ARC イベントの例です。

```
{
    "version": "0",
    "account":"111122223333",
    "detail-type":"Route 53 Application Recovery Controller cell readiness status
 change",
    "source":"route53-recovery-readiness.amazonaws.com",
    "time":"2020-11-03T00:31:54Z",
    "id": "1234a678-1b23-c123-12fd3f456e78",
    "region": "us-west-2",
    "resources":[
        "arn:aws:route53-recovery-readiness::111122223333:cell/PDXCell"
    ],
    "detail": {
        "cell-name": "PDXCell",
        "previous-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        },
        "new-state": {
            "readiness-status": "READY|NOT_READY|UNKNOWN|NOT_AUTHORIZED"
        }
    }
}
```

以下は、準備状況チェックステータスを変更するための Route 53 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を送信するターゲットを指定する必要があります。で使用可能なターゲットのリストについては EventBridge、「 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、 によって EventBridge 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 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。 CloudWatch

コンソール外の 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 ルールにロググループ EventBridge 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 PutResourceポリシー 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. https://console.aws.amazon.com/events/ で Amazon EventBridge コンソールを開きます。
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。

   準備状況の EventBridge ルールの作成の詳細については、「 で準備状況チェックリソースを監視する EventBridge」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CloudWatchとして 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。

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、誰を認証し (サインインさせ)、誰に

Route 53 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 AWS のサービス 使用できる です。

内容

SERVICElong の準備状況チェックと IAM の連携

IAM を使用して Route 53 ARC へのアクセスを管理する前に、Route 53 ARC で利用できる IAM の機能について確認します。

IAM を使用して Amazon Route 53 Application Recovery Controller 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 機能について学びます。

Amazon Route 53 Application Recovery Controller の準備状況チェックで使用できる IAM の機能

IAM 機能	準備状況チェックのサポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes
ポリシーリソース	Yes
ポリシー条件キー	Yes
ACL	No
ABAC (ポリシー内のタグ)	はい
一時的な認証情報	Yes
プリンシパル権限	Yes

IAM 機能	準備状況チェックのサポート
サービスロール	いいえ
サービスリンクロール	はい

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

準備状況チェックのためのアイデンティティベースのポリシー

アイデンティティベースポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限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 ポリシーの作成」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。

Route 53 ARC のアイデンティティベースのポリシー例については、「Amazon Route 53 Application Recovery Controller のアイデンティティベースのポリシーの例」を参照してください。

準備状況チェック内のリソースベースのポリシー

リソースベースのポリシーのサポート	No

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に対するサポート	はい

管理者は AWS JSON 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んなリソースにどんな条件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、関連する 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権限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権限を付与するポリシーで使用されます。

準備状況チェックのための Route 53 ARCアクションのリストを確認するには、「サービス認証リファレンス」の「Amazon Route 53 Recovery Readiness で定義されるアクション」を参照してください。

準備状況チェックのための Route 53 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。例えば、次のようになります。

```
"Action": [
      "route53-recovery-readiness:action1",
      "route53-recovery-readiness:action2"
         ]
```

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

```
"Action": "route53-recovery-readiness:Describe*"
```

準備状況チェックのための Route 53 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 Route 53 Application Recovery Controller の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

ポリシーリソースに対するサポート	はい

管理者は AWS JSON 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どのような条件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、リソースレベルの権限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 Route 53 ARC アクションのリストを確認するには、「Amazon Route 53 Recovery Readiness で定義されるアクション」を参照してください。

準備状況チェックのための Route 53 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 Route 53 Application Recovery Controller の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の例。

準備状況チェックのポリシー条件キー

サービス固有のポリシー条件キーのサポート	はい

管理者は AWS JSON 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んなリソースにどんな条件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することで、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 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS 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演算を使用して条件 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ます。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドのAWS 「 グローバル条件コンテキストキー」を参照してください。

準備状況チェックのための Route 53 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。

準備状況チェックのための Route 53 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 Route 53 Application Recovery Controller の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の例。

準備状況チェックでのアクセスコントロールリスト (ACLs

ACL のサポート	No

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権限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	部分的

属性ベースのアクセスコントロール (ABAC) は、属性に基づいて権限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操作を許可するように ABAC ポリシーを設計します。

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はいです。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 とは?」を参照してください。ABAC をセットアップするステップ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「属性に基づく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	はい

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。一時的な認証情報を AWS のサービス 使用できる 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 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ています。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスは自動的に一時的な認証情報を作成します。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「ロールへの切り替え (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、長期的なアクセスキーを使用する代わりに、動的に一時的な認証

情報を生成する AWS. AWS recommends にアクセスできます。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許可

フォワードアクセスセッション (FAS) をサポート	はい

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場合 AWS、ユーザーはプリンシパルと見なされます。ポリシーは、プリンシパルに権限を付与します。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権限が必要です。

準備状況チェックの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、「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	いいえ

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 のサービスに権限を委任するロールの作成」を参照してください。

準備状況チェックのサービスにリンクされたロール

サービスリンクロールのサポート	はい

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権限を表示できますが、編集することはできません。

Route 53 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、「Route 53 ARC で準備状況チェックにサービスにリンクされたロールを使用する」を参照してください。

サービスにリンクされたロールの作成または管理の詳細については、「IAM と提携するAWS のサービス」を参照してください。表の中から、[Service-linked role] (サービスにリンクされたロール) 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、[Yes] リンクを選択します。

Amazon Route 53 Application Recovery Controller の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の例

デフォルトでは、ユーザーとロールには Route 53 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受けることができます。

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 ポリシーの作成」を参照してください。

Route 53 ARC が定義するアクションとリソースタイプの詳細 (各リソースタイプの ARN の形式を含む) については、「サービス認可リファレンス」の「Amazon Route 53 Recovery コントロールのアクション、リソース、および条件キー」を参照してください。

トピック

- ポリシーの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

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで Route 53 ARC リソースの作成、アクセス、削除を行える人を決めます。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 管理ポリシーの使用を開始し、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を付与するには、多くの一般的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して、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「AWS ジョブ機能の管理ポリシー」を参照してください。

- 最小特権を適用する – IAM ポリシーで権限を設定するときは、タスクの実行に必要な権限のみを付与します。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、最小特権権限とも呼ばれています。IAM を使用して権限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 でのポリシーと権限」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、 などの特定の を通じてサービスアクションを使用する場合 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 policy elements: Condition] (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 ポリシーの検証」を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、セキュリティを強化するために MFA を有効にします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、『IAM ユーザーガイド』の「MFA 保護 API アクセスの設定」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 Route 53 Application Recovery Controller コンソールにアクセスするには、アクセス許可の最小限のセットが必要です。これらのアクセス許可により、 の Route 53 ARC リソースの詳細をリストおよび表示できます AWS アカウント。最小限必要な許可よりも制限が厳しいアイデンティティベースのポリシーを作成すると、そのポリシーを持つエンティティ (ユーザーまたはロール) に対してコンソールが意図したとおりに機能しません。

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

特定の API オペレーションのみへのアクセスを許可するときに、ユーザーとロールが引き続き準備状況チェックコンソールを使用できるようにするには、エンティティに準備状況チェック用のReadOnly AWS 管理ポリシーもアタッチします。詳細については、「IAM ユーザーガイド」の「準備状況チェックのマネージドポリシー」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、Route 53 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、「Route 53 ARC で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 Route 53 ARC API アクションを使用して Route 53 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場合などに、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、Route 53 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、「Route 53 ARC で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。

準備状況チェック用の API オペレーションを使用するには、次のようなポリシーをユーザーにアタッチします。

```
{
    "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": "*"
        }
    ]
}
```

Route 53 ARC で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 Route 53 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。サービスリンクロールは、サービス (この場合は Route 53 ARC) に直接リンクされる一意なタイプの IAM ロールです。サービスにリンクされたロールは、Route 53 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 の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。

サービスリンク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いため、Route 53 ARC の設定が容易になります。Route 53 ARC は、サービスリンク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、Route 53 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、リソースへの意図しないアクセスによる許可の削除が防止され、Route 53 ARC リソースは保護されます。

サービスにリンクされたロールをサポートしている他のサービスについては、「IAM と連携する AWS のサービス」で「サービスにリンクされたロール」列が「はい」になってい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。

Route 53 ARC には、この章で説明する以下のサービスリンクロールがあります。

- Route 53 ARC は、Route53RecoveryReadinessServiceRolePolicy 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、リソースと設定にアクセスして準備状況を確認します。
- Route 53 ARC は、自動shift プラクティス実行に 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、お客様が用意した 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、プラクティス実行を開始します。

Route53RecoveryReadinessServiceRolePolicy のサービスにリンクされたロールのアクセス許可

Route 53 ARC は、Route53RecoveryReadinessServiceRolePolicy 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、リソースと設定にアクセスして準備状況を確認します。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 のサービスにリンクされたロールのアクセス許可

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 サービスにリンクされたロールは、次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。

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) 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 ユーザーガイド」の「サービスにリンクされたロールのアクセス許可」を参照してください。

Route53RecoveryReadinessServiceRolePolicy ARC 用の Route 53サービスにリンクされたロールの作成

Route53RecoveryReadinessServiceRolePolicy サービスにリンクされたロールを手動で作成する必要はありません。、 AWS Management Console、または AWS API で最初の準備状況チェック AWS CLIまたはクロスアカウント認証を作成すると、Route 53 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。準備状況チェックまたはクロスアカウント認証を初めて作成するときは、Route 53 ARC が再度、ユーザーに代わって、サービスにリンクされたロールを作成します。

Route53RecoveryReadinessServiceRolePolicy ARC の Route 53サービスにリンクされたロールの編集

Route 53 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 を使用したロール記述の編集はできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Route53RecoveryReadinessServiceRolePolicy ARC の Route 53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が必要な機能またはサービスが不要になった場合には、その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認証を削除したら、Route53RecoveryReadinessServiceRolePolicy サービスにリンクされた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、「Amazon Route 53 Application Recovery Controller 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、「Route 53 ARC でのクロスアカウント認証の作成」を参照してください。

> **Note**
>
> リソースの削除を試みたときに、Route 53 ARC のサービスがロールを使用していた場合、サービスロールの削除が失敗することがあります。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。

サービスにリンクされたロールを 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 Route 53 ARC サービスにリンクされたロールの更新

Route 53 ARC の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、Route 53 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。Route 53 ARC のドキュメント履歴のページで、自動 RSS アラートにサブスクライブすることもできます。

AWS Amazon Route 53 Application Recovery Controller 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、多くの一般的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るため、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するわけではないことに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシー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ているアクセス許可 AWS を更新すると、その更新は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 AWS のサービス が起動されたとき、または既存のサービスで新しい API オペレーションが使用可能になったときに、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: 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 Route 53 Application Recovery Controller に、Route 53 ARC が使用または管理している AWS サービスやリソースへのアクセスを許可する、サービスにリンクされたロールにアタッチされます。詳細については、「Route 53 ARC で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、Route 53 ARC のリカバリの準備状況 (準備状況チェック) を操作するアクションへの、フルアクセスを許可します。これを、リカバリの準備状況へのフルアクセスを必要とする IAM ユーザーとその他のプリンシパルにアタッチします。

このポリシーの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AmazonRoute「53RecoveryReadinessFullAccess」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、Route 53 ARC のリカバリの準備状況を操作するアクションへの読み取り専用アクセスを許可します。これは、準備状況のステータスとリカバリグループの設定を確認する必要があるユーザーに役立つポリシーです。これらのユーザーは、リソースを作成、更新、削除できません。

このポリシーのアクセス許可を確認するには、「 AWS マネージドポリシーリファレンス」のAmazonRoute「53RecoveryReadinessReadOnlyAccess」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

Route 53 ARC がこれらの変更の追跡を開始してから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 Route 53 Application Recovery Controller の AWS マネージドポリシーの更新。このページの変更に関する自動通知を受け取るには、Route 53 ARC のドキュメントの履歴ページの RSS フィードにサブスクライブします。

準備状況チェックのクォータ

Amazon Route 53 Application Recovery Controller 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

エンティティ	クォータ
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

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、関連するシナリオやサービス間の例ではアクションのコンテキストが確認できます。

AWS SDK デベロッパー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 でこのサービスを使用する。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アクション
  - AWS SDK または CLI GetRoutingControlStateで を使用する
  - AWS SDK または CLI 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。これらは、Application Recovery Controller API を呼び出すものであり、コンテキスト内で実行する必要がある大規模なプログラムのコードの抜粋です。各例には GitHub、コードの設定と実行の手順を示す へのリンクが含まれています。

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 または CLI GetRoutingControlStateで を使用する
- AWS SDK または CLI UpdateRoutingControlStateで を使用する

AWS SDK または CLI **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**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、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、「 API リファレンスGetRoutingControlState」の「」を参照してください。 AWS SDK for Java 2.x

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、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、 GetRoutingControlState AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパー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 でこのサービスを使用する。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 または CLI **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**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、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、「 API リファレンスUpdateRoutingControlState」の「」を参照してください。 AWS SDK for Java 2.x

Python

SDK for Python (Boto3)

> **Note**
>
> については、「」を参照してください GitHub。AWS コード例リポジトリ で全く同じ例を見つけて、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、 UpdateRoutingControlState AWS SDK for Python (Boto3) API リファレンスの「」を参照してください。

AWS SDK デベロッパー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 でこのサービスを使用する。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 Route 53 Application Recovery Controller

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を担います AWS クラウド。また、 は、お客様が安全に使用できるサービス AWS も提供します。コンプライアンスAWS プログラムコンプライアンスプログラム の一環として、サードパーティーの監査者は定期的にセキュリティの有効性をテストおよび検証。Amazon Route 53 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、お客様は、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、Route 53 ARC を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために Route 53 ARC を設定する方法を示します。また、Route 53 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 Route 53 Application Recovery Controller でのデータ保護

AWS 責任共有モデル、Amazon Route 53 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。お客様は、このインフラストラクチャでホストされてい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の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 は必須であり TLS 1.3 がお勧めです。
- 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。
- 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 などの高度なマネージド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 140-2 検証済みの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) 140-2」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は、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配置し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 Route 53 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 名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

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。

Route 53 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、バックアップ用に Amazon EBS ボリュームに書き込まれます。Route 53 ARC は、データの保管時は、デフォルトの Amazon EBS 暗号化を使用します。

転送中の暗号化

Route 53 ARC の設定、準備状況のクエリ、セル状態の更新など、顧客のリクエストと応答は、TLS を使用してサービス全体で転送中に暗号化されます。

Amazon Route 53 Application Recovery Controller 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、誰を認証し (サインインさせ)、誰に Route 53 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 AWS のサービス 使用できる です。

対象者

AWS Identity and Access Management (IAM) の使用方法は、Route 53 ARC で行う作業によって異なります。

サービスユーザー – ジョブを実行するために Route 53 ARC サービスを使用する場合は、管理者から必要な認証情報とアクセス許可が与えられます。作業を実行するためにさらに多くの Route 53 ARC の機能を使用する際は、追加の許可が必要になる場合があります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。Route 53 ARC の機能にアクセスできない場合は、「Amazon Route 53 Application Recovery Controller のアイデンティティとアクセスのトラブルシューティング」を参照してください。

サービス管理者 - 社内の Route 53 ARC リソースを担当している管理者は、通常、Route 53 ARC にフルアクセスできます。サービスのユーザーがどの Route 53 ARC 機能や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。会社で Route 53 ARC の IAM をどう使用できるのかの詳細については、「Amazon Route 53 Application Recovery Controller の機能が IAM と連携する方法」を参照してください。

IAM 管理者 - IAM 管理者は、Route 53 ARC へのアクセスを管理するポリシーの詳しい作成方法を確認したい場合があります。IAM で使用できる Route 53 ARC のアイデンティティベースポリシーの例を確認するには、「Amazon Route 53 Application Recovery Controller 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。

アイデンティティを使用した認証

認証とは、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 の認証情報は、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。

ユーザーのタイプに応じて、 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、「 ユーザーガイド」の「 へのサインイン AWS アカウント方法AWS サインイン 」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 Software Development Kit (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に暗号で署名します。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM ユーザーガイドの API AWS リクエストの署名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。例えば、 AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを向上させる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「Multi-factor authentication」(多要素認証) および「IAM ユーザーガイド」の「AWSでの多要素認証 (MFA) の使用」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が、一時的な認証情報を使用して にアクセスするために ID プロバイダーとのフェデレーションを使用することを要求 AWS のサービス 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 ソースを通じ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 ID がにアクセスすると 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ことも、独自の ID ソース内のユーザーとグループのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。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 内のアイデンティティです。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、ロールでは一時的な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 (ロールではなく) の作成が適している場合」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 内のアイデンティティです。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。ロール を切り替える AWS Management Console ことで、 で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、 または AWS API AWS CLI オペレーションを呼び出すか、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「IAM ロールの使用」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詳細については、「IAM ユーザーガイド」の「Creating a role for a third-party Identity Provider」(サードパーティーアイデンティティプロバイダー向けロールの作成) 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 ユーザーガイド」の「アクセス許可セット」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、 (ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。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 でアクションを実行する場合 AWS、ユーザーは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 へのリクエストのリクエスト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 のサービスにアクセス許可を委任するロールの作成」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。
- 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。 AWS ロールを EC2 インスタンスに割り当て、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。

IAM ロールと IAM ユーザーのどちらを使用するかについては、IAM ユーザーガイドの(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?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、 AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられ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。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) AWS がリクエストを行うと、 はこれらのポリシー 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 AWS JSON 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んなリソースにどんな条件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継ぐことができます。

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 ポリシーの作成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 ユーザーガイドのマネージドポリシーとインラインポリシーの比較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、 AWS WAF、および Amazon VPC は、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、一般的では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のポリシーによって 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) に対する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、 AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。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「SCP の仕組み」を参照してください。

- 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。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 Route 53 Application Recovery Controller の機能が IAM と連携する方法

Amazon Route 53 Application Recovery Controller の各機能が 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

Amazon Route 53 Application Recovery Controller のアイデンティティベースのポリシーの例

Amazon Route 53 Application Recovery Controller 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。

- ゾーン自動shift のアイデンティティベースのポリシーの例

- Amazon Route 53 Application Recovery Controller 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
- Amazon Route 53 Application Recovery Controller 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
- Amazon Route 53 Application Recovery Controller の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の例

AWS Amazon Route 53 Application Recovery Controller 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 マネージドポリシーを含む、 マネージドポリシーを使用する Amazon Route 53 Application Recovery Controller 機能の マネージドポリシーについては、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 Route 53 Application Recovery Controller の AWS マネージドポリシーの更新

Route 53 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、このサービスがこれらの変更の追跡を開始した以降の分について表示します。このページの変更に関する自動通知を受け取るには、Route 53 ARC のドキュメントの履歴ページの RSS フィードにサブスクライブします。

変更	説明	日付
AWSServiceRoleForPercPracticePolicy – 新しいポリシー	Route 53 ARC に、オートシフトと練習実行用の新しいサービスリンクロールが追加されました。 Route 53 ARC は、サービスにリンクされたロールによって有効化されたアクセス許可を使用して、お客様が用意した Amazon CloudWatch ア	2023 年 11 月 30 日

変更	説明	日付
	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、「のサービスにリンクされたロールのアクセス許可 AWSServiceRoleForZonalAutoshiftPracticeRun」を参照してください。	
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ための `GetResourcePolicy` のアクセス許可を追加します。	2023 年 10 月 18 日

変更	説明	日付

Route53RecoveryReadinessServiceRolePolicy – 更新されたポリシー	Route 53 ARC に、Amazon EC2 インスタンスに関する情報をクエリするための新しいアクセス許可が追加されました。
	
	Route 53 ARC は、以下のアクセス許可を使用して Amazon EC2 インスタンスのポーリングをサポートし、準備状況チェックを実行して、インスタンスの準備状況のステータスを判定します。
	
	`ec2:DescribeVpnGateways`
	
	`ec2:DescribeCustomerGateways`
		2023 年 2 月 17 日

Route53RecoveryReadinessServiceRolePolicy – 更新されたポリシー	Route 53 ARC に、Lambda 関数の情報をクエリするための新しいアクセス許可が追加されました。 Route 53 ARC は、以下のアクセス許可を使用して Lambda 関数に関する情報をクエリし、準備状況チェックを実行して、関数の準備状況のステータスを判定します。 `lambda:ListProvisionedConcurrencyConfigs`	2022 年 8 月 31 日

変更	説明	日付
AmazonRoute53RecoveryControlConfigFullAccess – ポリシーの更新	ポリシーから Amazon Route 53 のアクセス許可が削除され、オプションのアクセス許可を記した注記が、新たに追加されました。	2022 年 5 月 26 日
AmazonRoute53RecoveryControlConfigFullAccess – ポリシーの更新	不足していた、ポリシーへの Amazon Route 53 のアクセス許可が追加されました。	2022 年 4 月 15 日
AmazonRoute53RecoveryClusterReadOnlyAccess – ポリシーの更新	Route 53 ARC に新しいアクセス許可 `route53-recovery-cluster:ListRoutingControls` が追加され、可用性の高いルーティングコントロール ARN をリスト化できるようになりました。	2022 年 3 月 15 日
AmazonRoute53RecoveryControlConfigReadOnlyAccess - ポリシーの更新	Route 53 ARC に新しいアクセス許可 `route53-recovery-control-config:ListTagsForResources` が追加され、リソースのタグをリスト化できるようになりました。	2021 年 12 月 20 日

変更	説明	日付
Route53RecoveryReadinessServiceRolePolicy – 更新されたポリシー	Route 53 ARC に、Amazon API Gateway に関する情報をクエリするための新たなアクセス許可が追加されました。 Route 53 ARC は、アクセス許可 `apigateway:GET` を使用して API ゲートウェイに関する情報をクエリし、準備状況チェックを実行して、準備状況のステータスを判定します。	2021 年 10 月 28 日
AmazonRoute53RecoveryReadinessReadOnlyAccess – 新しいアクセス許可を追加	Route 53 ARC は AmazonRoute53 に 2 RecoveryReadinessReadOnlyAccessつの新しいアクセス許可を追加しました。 Route 53 ARC は、`route53-recovery-readiness:GetArchitectureRecommendations` と `route53-recovery-readiness:GetCellReadinessSummary` を使って、リカバリの準備状況を操作するこれらのアクションへの、読み取り専用アクセスを許可します。	2021 年 10 月 15 日

変更	説明	日付

Route53RecoveryReadinessServiceRolePolicy – 更新されたポリシー	Route 53 ARC に、Lambda 関数の情報をクエリするための新しいアクセス許可が追加されました。
	
	Route 53 ARC は、以下のアクセス許可を使用して Lambda 関数に関する情報をクエリし、準備状況チェックを実行して、関数の準備状況のステータスを判定します。
	
	`lambda:GetFunctionConcurrency`
	
	`lambda:GetFunctionConfiguration`
	
	`lambda:GetProvisionedConcurrencyConfig`
	
	`lambda:ListAliases`
	
	`lambda:ListVersionsByFunction`
	
	`lambda:ListEventSourceMappings`
	
	`lambda:ListFunctions`
		2021 年 10 月 8 日

変更	説明	日付
Route53RecoveryReadinessServiceRolePolicy — 新しい管理ポリシーを追加	Route 53 ARC に以下の新しいマネージドポリシーが追加されました。 AmazonRoute53RecoveryReadinessFullAccess AmazonRoute53RecoveryReadinessReadOnlyAccess AmazonRoute53RecoveryClusterFullAccess AmazonRoute53RecoveryClusterReadOnlyAccess AmazonRoute53RecoveryControlConfigFullAccess AmazonRoute53RecoveryControlConfigReadOnlyAccess	2021 年 8 月 18 日
Route 53 ARC で変更追跡を開始	Route 53 ARC が AWS マネージドポリシーの変更の追跡を開始しました。	2021 年 7 月 27 日

Amazon Route 53 Application Recovery Controller 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 Route 53 Application Recovery Controller 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

- Route 53 ARC でアクションを実行する権限がありません。

- iam を実行する権限がありません。PassRole
- 自分の 以外のユーザーに Route 53 ARC リソース AWS アカウント へのアクセスを許可したい

Route 53 ARC でアクションを実行する権限がありません。

から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 を実行する権限がありません。PassRole

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Route 53 ARC にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを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。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Route 53 ARC でアクションを実行しようする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。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。サインイン認証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 Route 53 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:

- Route 53 ARC がこれらの機能をサポートしているかどうかを確認するには、「Amazon Route 53 Application Recovery Controller の機能が IAM と連携する方法」を参照してください。
- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、IAM ユーザーガイドの「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティー に提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

Amazon Route 53 Application Recovery Controller のログ記録とモニタリング

モニタリングは、Amazon Route 53 Application Recovery Controller 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、Route 53 ARC の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。例えば、AWS CloudTrail や Amazon です CloudWatch。

Route 53 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
- 準備状況チェックのログ記録とモニタリング

Amazon Route 53 Application Recovery Controller のコンプライアンスの検証

サードパーティーの監査者は、複数の コンプライアンスプログラムの一環として Amazon Route 53 Application Recovery Controller のセキュリティと AWS コンプライアンスを評価します。このプログラムには、SOC、PCI、HIPAA などを含みます。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」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 を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、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に重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。
- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャ — このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべての AWS のサービス が HIPAA の対象となるわけではありません。詳細については、HIPAA 対応サービスのリファレンスを参照してください。

- 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。このガイドでは、ガイダンスを保護し AWS のサービス 、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にマッピングするためのベストプラクティスをまとめています。
- 「 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、リソース設定が社内プラクティス、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確認できます 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リストについては、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 — これにより AWS アカウント、疑わしい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監視することで、、、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 は、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への対応に役立ちます。
- 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 Route 53 Application Recovery Controller のレジリエンス

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。 AWS リージョン は、低レイテンシー、高スループット、および冗長性の高い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。

Route 53 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に対応できるようにいくつかの機能が用意されています。

Amazon Route 53 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である Amazon Route 53 Application Recovery Controller 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、「セキュリティの柱 AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

AWS が公開した API コールを使用して、ネットワーク経由で Route 53 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 は必須で TLS 1.3 がお勧めです。
- 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 Route 53 Application Recovery Controller デベロッパーガイド、ドキュメント履歴

以下に、Amazon Route 53 Application Recovery Controller のドキュメントの重要な変更事項を記載します。

- バージョン: 最新
- ドキュメントの最終更新日: 2024 年 4 月 30 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す。つまり、Route 53 ARC の各機能の包括的な情報を含む個別のセクションができました。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、および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です。 詳細については、「Amazon Route 53 Application Recovery Controller 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	Route 53 ARC に新しい機能が追加されました。これにより、 が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	2023 年 11 月 30 日

変更	説明	日付
	、イベント中の復旧までの時間を短縮できるようになります。 詳細については、「Amazon Route 53 Application Recovery Controller のゾーンオートシフト」を参照してください。	
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練習実行AWSServiceRoleForZonalAutoshiftPracticeRun用の新しいサービスにリンクされたロール が追加されました。 詳細については、「 のサービスにリンクされたロールのアクセス許可 AWSServiceRoleForZonalAutoshiftPracticeRun」を参照してください。	2023 年 11 月 30 日

変更	説明	日付
クラスターのクロスアカウントのサポートを追加	で Route 53 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、「Route 53 ARC でクラスターのクロスアカウントをサポート」を参照してください。	2023 年 10 月 18 日
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

変更	説明	日付
サービスにリンクされたロールを更新	Amazon EC2 インスタンスのポーリングをサポートするため、Route 53 ARC のサービスリンクロールに新しいアクセス許可 `ec2:DescribeVpnGateways` と `ec2:DescribeCustomerGateways` を追加しました。 詳細については、「Route 53 ARC でサービスリンクロールを使用する」を参照してください。	2023 年 2 月 17 日
ゾーンシフトの一般提供版	Route 53 ARC のゾーンシフトの一般提供版がサポートされました。これには、ゾーンシフト用に Route 53 ARC に登録されているマネージドリソースの属性ベースのアクセス制御 (ABAC) が含まれています。 詳細については、「Route 53 ARC による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

変更	説明	日付
新しいマルチ AZ ゾーンシフトを追加	Route 53 ARC の、マルチ AZ アプリケーション用の新サービスであるゾーンシフトについて説明するコンテンツが追加されました。ゾーンシフトを開始すると、ロードバランサーのリソースのトラフィックを、アベイラビリティーゾーンから切り離せます。 詳細については、「Zonal shift in Route 53 ARC」を参照してください。	2022 年 11 月 28 日
サービスにリンクされたロールを更新	Route 53 ARC のサービスリンクロールに、Lambda 関数に関する情報をクエリするための新しいアクセス許可 `lambda:ListProvisionedConcurrencyConfigs` を追加しました。 詳細については、「Using service-linked roles for Route 53 ARC」を参照してください。	2022 年 8 月 31 日

変更	説明	日付
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 managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2022 年 5 月 26 日
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 managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2022 年 4 月 15 日

変更	説明	日付
新しいルーティングコントロールリスト API の CLI 例を追加	非常に信頼性の高い Route 53 ARC データプレーン API に含まれる、新しいルーティングコントロールリスト API オペレーションの、CLI コマンド例とベストプラクティスの推奨事項が追加されました。 詳細については、「List and update routing controls and states」を参照してください。	2022 年 3 月 31 日
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

変更	説明	日付
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、Route 53 ARC の追加リソースのタグ付けが、新たにサポートされました。 詳細については、「Tagging in Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 12 月 20 日
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 managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 12 月 20 日

変更	説明	日付
によるリアルタイムアラートのサポートを追加 EventBridge	のサポートが追加されました。つまり EventBridge、ステータスが READY から NOT READY に変わった場合などに、アラートを取得して Route 53 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 での Route 53 ARC EventBridgeの使用」を参照してください。	2021 年 12 月 20 日
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、「API examples for Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 11 月 16 日

変更	説明	日付
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 managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 11 月 9 日
Amazon API Gateway リソースタイプを新たにサポート	新しいリソースタイプとして Amazon API Gateway が追加され、Route 53 ARC の、サービスにリンクされたロールのアクセス許可が更新されました。今後は、Route 53 ARC で準備状況チェックを使って API Gateway を監査できます。 詳細については、「Readiness rules and supported resource types」および「Using service-linked roles for Route 53 ARC」を参照してください。	2021 年 10 月 28 日

変更	説明	日付
Lambda 関数のリソースタイプを新たにサポート	新しいリソースタイプとして Lambda 関数が追加され、Route 53 ARC の、サービスにリンクされたロールのアクセス許可が更新されました。今後は、Route 53 ARC で準備状況チェックを使って Lambda 関数 を監査できます。 詳細については、「Readiness rules and supported resource types」および「Using service-linked roles for Route 53 ARC」を参照してください。	2021 年 10 月 8 日
CloudFormation および Terraform テンプレートへのリンクを追加しました	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が追加され、Route 53 ARC の使用をすばやく開始できるようになりました。詳細については、「新しいアプリケーションによるリカバリの準備」を参照してください。	2021 年 9 月 13 日

変更	説明	日付
新しいマネージドポリシーを追加	Route 53 ARC に、、AmazonRoute53RecoveryReadinessFullAccess 、AmazonRoute53RecoveryReadinessReadOnlyAccess 、、AmazonRoute53RecoveryClusterReadOnlyAccess 、AmazonRoute53RecoveryClusterFullAccess の AWS マネージドポリシーを追加AmazonRoute53RecoveryControlConfigFullAccess しましたAmazonRoute53RecoveryControlConfigReadOnlyAccess 。 詳細については、「AWS managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 8 月 18 日
Amazon Route 53 Application Recovery Controller の AWS マネージド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 managed policies for Amazon Route 53 Application Recovery Controller」を参照してください。	2021 年 7 月 27 日

変更	説明	日付
Amazon Route 53 Application Recovery Controller の初回のリリース	Route 53 ARC は、 AWS リージョン内または複数のリージョンにまたがる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。Route 53 ARC に準備状況チェックが追加され、フェイルオーバートラフィックに対処できるようにアプリケーションをスケールし、障害を回避するように設定することが可能になりました。また、信頼性がきわめて高いルーティングコントロールが可能であるため、例えば、アベイラビリティーゾーンやリージョンを横断してトラフィックのルートを変更し、アプリケーションを復旧することが可能です。詳細については、「What is Route 53 ARC?」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。