



ネットワークトラフィックを にカットオーバーするためのベストプラクティ
ス AWS

AWS 規範ガイドンス



AWS 規範ガイド: ネットワークトラフィックを にカットオーバーするためのベストプラクティス AWS

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

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

Table of Contents

序章	1
ターゲットを絞ったビジネス成果	1
カットオーバーフェーズの概要	2
コミュニケーションとガバナンス	3
コミュニケーションとガバナンスの戦略	3
コミュニケーションとガバナンスのチャネル	3
カットオーバー前段階	5
カットオーバープランの作成	5
カットオーバーランブックの作成	6
カットオーバーステージ	7
カットオーバーのアプローチ	7
一括アプローチ	7
段階的アプローチ (canary デプロイ)	8
適切なアプローチの選択	8
カットオーバーを実行するタイミング	10
カットオーバー前にテストすべきこと	10
カットオーバー前にテストできないこと	10
運用準備状況のレビュー	10
ロールバック	10
カットオーバー中、または新しいデータがない場合のロールバック	11
変更したデータを使ったロールバック	11
カットオーバー後の段階	12
モニタリングダッシュボードの設定	12
成功を理解し伝えること	12
保証期間の定義	12
結論	13
リソース	14
ドキュメント履歴	15
.....	xvi

移行 AWS 中にネットワークトラフィックを にカットオーバーするためのベストプラクティス

Damien Renner、Ernest Asare、Martin Fluck、Khurram Mahmood、Mark Willis、Amazon Web Services (AWS)

2022 年 10 月 ([ドキュメント履歴](#))

カットオーバーフェーズは、クラウド移行における最も重要な段階の 1 つです。カットオーバーでは、ネットワークトラフィックをソースシステムから Amazon Web Services (AWS) でホストされているターゲットシステムにリダイレクトします。カットオーバー中は、システムやユーザーにある程度の中断が発生する可能性が最も高くなります。中断を最小限に抑え、カットオーバーのリスクを軽減するために、このガイドで説明されているベストプラクティスに沿ったカットオーバープランを立てることを推奨します。このガイドは、移行コンサルタント、アプリケーションアーキテクト、プログラママネージャー、アプリケーション所有者、およびクラウド移行を実装または主導するその他の役割を対象としています AWS。

ターゲットを絞ったビジネス成果

このガイドの最終的な目標は、クラウド移行のカットオーバーフェーズに伴う中断を最小限に抑え、リスクを軽減できるようにすることです。このガイドを活用することで、次のようなターゲットを絞ったビジネス成果を実現し、全体としての目標を達成できます。

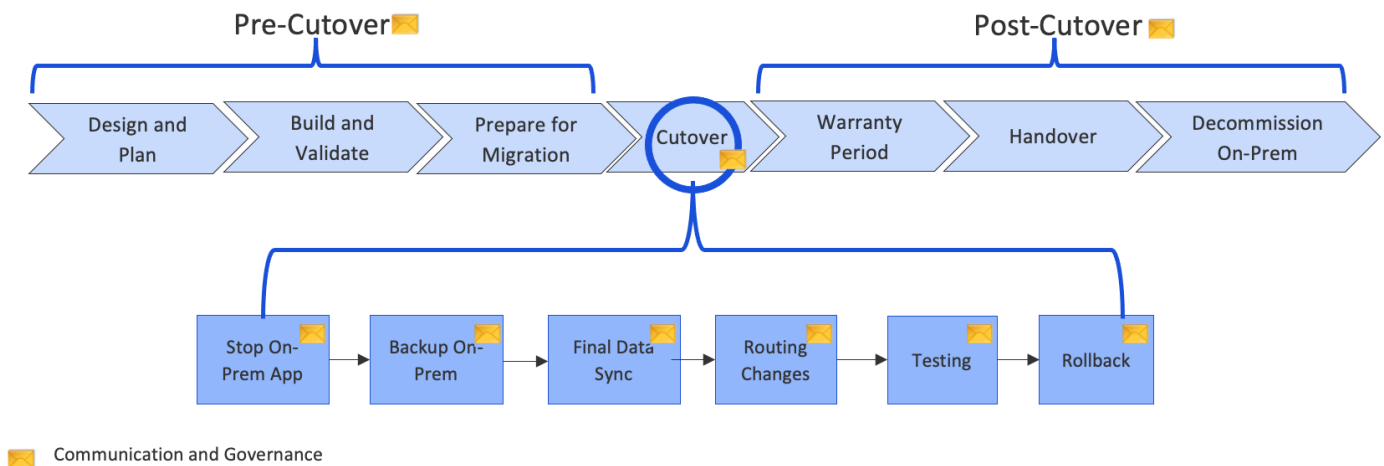
- 最適なカットオーバープロセスの開発
- カットオーバーのためのコミュニケーションとガバナンス戦略の策定
- さまざまなカットオーバーオプションを理解し、要件に基づいて最も効果的なものを選択

カットオーバーフェーズの概要

移行プロセスのカットオーバーフェーズでは、ネットワークトラフィックを既存のエンドポイントから AWS クラウドに新しくデプロイされたリソースに移動します。例えば、ドメインネームシステム (DNS) レコードを新しくプロビジョニングされた Elastic Load Balancer のエンドポイントに更新できます。ほとんどのアプリケーションでは、移行のこの段階でカットオーバーが必要な統合ポイントが複数存在することがよくあります (例えば、2 層アプリケーションを移行するためにアプリケーションとデータベース層をカットオーバーするなど)。

カットオーバーに伴う時間、複雑さ、リスクは、ビジネス要件、テクノロジースタック、リスク選好度、アプリケーションアーキテクチャ、その他の要因によって異なります。ただし一般的には、カットオーバー期間が短くなるにつれて、カットオーバーの複雑さは増します。カットオーバーに必要なダウンタイム期間の長さ、移行するアプリケーションの重要度との間には強い相関関係があります。通常、ビジネスクリティカルなアプリケーションではダウンタイム期間を短くする必要があるため、こうしたリスクは高まります。このようなリスクの増大に備え、複雑さを管理してカットオーバーのリスクを軽減するための、より広範囲にわたる計画が必要となります。

カットオーバーフェーズは [移行プロセス](#) の最終段階で行われます。次の図は、カットオーバーフェーズが移行プロセスのどの位置に当てはまるのかを示しています。図が示すように、カットオーバーフェーズではコミュニケーションとガバナンス (このガイドの次のセクションで説明します) が重要な役割を果たします。



コミュニケーションとガバナンスの計画

コミュニケーションとガバナンスの戦略

何をどのように伝えるかが、カットオーバーを成功させる鍵となります。成功させるためには、コミュニケーションツールに関する明確なガイダンスを提供し、オーナーシップの役割を定義し、各カットオーバータスクのコミュニケーションチャネルを指定するコミュニケーションとガバナンスの戦略を策定・実施することを推奨します。また、コミュニケーション戦略は移行チームがドキュメント化して管理し、プロジェクトガバナンスの一環として移行の 에스CALEーションパスと統合するといでしょう。

コミュニケーションとガバナンスのチャネル

コミュニケーションチャネルを作成するときは、ターゲットとなるオーディエンスを定義し、どのコミュニケーション方法が最も効果的かを判断することが重要です。たとえば、組織内のほとんどの人が積極的にメールチェックを行わない場合、メールよりもインスタントメッセージを使うほうが適しているかもしれません。関連するすべてのステークホルダーを対象としたチャネルの設置を推奨します。理想的には、カットオーバーの前、最中、後にチャネルを開放し、設定したコミュニケーションプロトコルについてステークホルダーに説明責任を負わせます。

場合によっては、コミュニケーションチャネルをグループ化する必要があるかもしれません (例えば、部署ごとに、または密接に関連するアプリケーションごとに Slack グループを作成するなど)。コミュニケーションとメッセージを管理する責任は、コミュニケーションリーダーまたは移行マネージャーに割り当てることを推奨します。

メモ

カットオーバーを行う前に、移行チーム、ウェブオーナー、アプリケーションオーナーから賛同を得ることがベストプラクティスです。アプリケーションのステークホルダー全員に、カットオーバーについて十分な通知を行います。

次の表は、さまざまなコミュニケーションチャネルに対する高レベルなコミュニケーションプロトコルの例を示しています。

Channel	[メソッド]	所有者	頻度
シニアステークホルダー (社内)	E メール	プログラマネージャーまたはカットオーバーマネージャー	主要イベント (開始日と終了日など) または確認済みの P1 問題
プロジェクト/テクニカルチーム	Slack または Microsoft Teams チャットルーム	アクティビティに関する最新情報の投稿を担当するすべてのステークホルダー プログラマネージャーまたはカットオーバーマネージャー	すべてのタスクの開始日と終了日と遅延
データセンターのモニタリングチーム/L1 OP	ブリーフィングセッションとEメール	プログラマネージャーまたはカットオーバーマネージャー	エンゲージメント開始 (4 週間後) と完了した各ステージの更新
顧客のシニアステークホルダー	ブリーフィングセッションとEメール	プログラマネージャーまたはカットオーバーマネージャー	エンゲージメント開始 (4 週間後) と完了した各ステージの更新
エンドカスタマー	E メール	プログラマネージャーまたはアプリケーションオーナー	カットオーバー開始日と終了日

カットオーバー前段階

カットオーバーオプションは、使用している移行パターンや、ソース環境およびターゲット環境のツール、テクノロジー、設定など、さまざまな要因によって異なります。例えば、VMware 環境をに移行する場合のカットオーバーオプションは、まったく新しいスタックを構築する AWS 場合とは異なります。

基本的に、すべてのカットオーバーは同じ結果、つまりワークロードを別の場所に移動することを目指しています。新しいエンドポイントに接続するためのアップストリームとダウンストリームの依存関係を設定できるように、サーバーやアプリケーションがどこに移動したかを把握することが重要です。

サービスの性質によっては、既存のデータベースと、新しい場所に移動する他のシステムへの接続が複雑になる場合があります。データベースを同期して両方の場所のデータの整合性を確保すると、複雑さが増すことがあります。さらに、複数のサービスでは、ネットワーク設定を調整するか、複数のサービスを同時にカットオーバーする必要があります。

カットオーバープランの作成

遅延、やり直し、計画外の停止、データ損失、パフォーマンスの問題、ユーザーエクスペリエンスの低下といったリスクを軽減するために、カットオーバープランの策定、カットオーバーワークブックの作成、およびカットオーバーのリハーサルを行うことができます。カットオーバープランには、カットオーバーが失敗した場合に備えて、緊急時対応計画と障害リスク軽減戦略を含めることを推奨します。カットオーバープランの一部として、ロールバックの手順を必ずドキュメント化しておきます。

カットオーバー前の計画プロセスの一環として、以下の重要項目を分析および評価することを推奨します。

- 必要となるダウンタイムには以下が含まれます。
 - 割り当てられたダウンタイム期間を超過した場合のビジネスへの影響 (収益や信頼など)
 - 不測の事態が発生した場合の「修正」活動に対する緊急時対応
 - 障害発生時のロールバック時間
- ユーザーの接続性
- アプリケーションとシステムの依存関係
- インフラストラクチャの変更

- 運用上の変更
- テストプランには以下が含まれます。
 - 機能テスト
 - ベースラインテストとカットオーバー後のテストによるパフォーマンステスト
 - 依存関係にあるアプリケーション、システム、サービスの統合テストおよび接続テスト
 - カットオーバー期間および保証期間中の戦略テスト
- ツールと戦略に関連するデプロイの変更

最後に、次のことをプランに組み込むことを検討してください。

- タスクの定義と期間
- タスクオーナー
- ステージとタスクの順序

カットオーバーランブックの作成

移行チームがそれぞれの予定開始時刻と終了時刻、順序、所有者などアクティビティの追跡に利用できるカットオーバーランブックの作成を推奨します。RACI マトリックスを使用すれば、適切なチームメンバーに責任を割り当てることができます。カットオーバーランブックには次のような利点があります。

- さまざまなステークホルダーを積極的に連携させることによるリスクの軽減
- 実証済みのメカニズムを再利用してアプリケーションを AWS クラウドに移行することで生産性を向上
- カットオーバーのプランニング用として構造化テンプレートを使用することによる、アプリケーション移行の速度向上
- 必要なプロセスに従うためのフレームワークの提供
- プロセスの変更をすべてのチームに伝達できる仕組みの提供

カットオーバーステージ

データを保存するコンポーネントを移行するときは、データ整合性が重要な要件であるかどうかを検討する必要があります。データ整合性が重要な要件である場合は、カットオーバープロセスを開始する前にソース環境 (データベースロックなど) をロックする必要がある場合があります。データベースをロックすることで、ソース環境に対して新規のトランザクションが行われないようにできます。ただし、ロックを行うと、より大きなダウンタイム時間が必要になることがあります。

カットオーバーには一般的に、次のようなフェーズが含まれます。

- **取り込みフリーズ** — オンプレミスのアプリケーションとデータのデータベースへの取り込みをフリーズします。これにより、カットオーバー中にオンプレミス版のアプリケーションが新規のトランザクションやデータを受け取ることがなくなります。
- **バックアップ** — オンプレミスシステムの最終バックアップを取ります。必要に応じて、緊急時のロールバックにこのバックアップを使用できます。
- **データの同期** — オンプレミス環境とターゲット (クラウド) 環境間の最終的なデータの同期を完了します。
- **ルーティング変更** — ユーザーをクラウド環境にルーティングします (DNS レコードの更新やロードバランサーのターゲット変更など)。
- **テスト** — 移行を完了としてマークする前に、すべてが機能していることをテストし検証します。

カットオーバーのアプローチ

考慮すべきカットオーバーのアプローチには、一括アプローチと段階的アプローチの 2 つがあります。最適なカットオーバーのアプローチを選択する鍵は、ビジネス要件と技術的制約を理解することにあります。このセクションでは、両方のアプローチの概要を示します。

一括アプローチ

一括アプローチを採用すると、スイッチを押すだけでソリューション全体がカットオーバーされます。例えば DNS を更新したり、ロードバランサーを変更することでこれを実行できます。そうすれば、すべてのユーザーとライブトラフィックが直ちに新しいシステムを使用できるようになります。この方法は、ホスト名の競合、ライセンスの問題、またはドメイン認証の制約により新しいシステムをオンラインにできない場合に便利です。時間が重要なため、いつ、誰がフェイルバックを要求するかに重点が置かれます。アプリケーションの機能的特徴と非機能的特徴の両方を検証できるように、

一括アプローチの計画には広範なパフォーマンステストと、該当する場合はリグレーションテストを含めることを推奨します。

段階的アプローチ (canary デプロイ)

段階的アプローチでは、一定期間にわたって段階的にカットオーバーします。このアプローチには、現在のシステムが負荷に耐えられるかどうか、各システムコンポーネントが期待どおりに機能しているかどうかを検証するための継続的なモニタリングとチェックが含まれます。段階的なアプローチでは、フィードバックに基づいてシステムパフォーマンスを調整できるため、カットオーバーに関する潜在的な問題のリスクを軽減できます。また、重大な問題を特定した場合に、変更をロールバックするのも簡単です。

適切なアプローチの選択

適切なアプローチを選択するために、以下を特定してください。

- 同じ移動グループに属する依存アプリケーションとサービス
- オンプレミスと移行したアプリケーションの間で利用できる共通のデータソース
- 部分的な負荷を異なるエンドポイントにリダイレクトできるアプリケーションとインフラストラクチャ

データソースとアプリケーションサーバー間のレイテンシー増加に耐えられないアプリケーションの場合、一括アプローチが必要であることを明確に示しています。このシナリオでは、すべてのアプリケーションリソース (サーバーとデータベース) をまとめてカットオーバーすることで、パフォーマンスへの影響を回避できます。

段階的なカットオーバーでは、アプリケーションを構成するサーバーとサービスの割合を全体から分割して にカットオーバーし AWS、残りのサーバーとサービスはオンプレミスのままです。次に、ロードバランシングまたは DNS ポリシーに基づいて、クライアントトラフィックを両方の環境にルーティングします。段階的なカットオーバーを行うと、ユーザーへの影響を最小限に抑えられるため、カットオーバーの影響を受けるユーザー数は少なくて済みます。影響を特定できれば、それに応じてカットオーバーするサーバーとサービスの割合を調整できます。ただし、段階的なカットオーバーのアプローチは、基盤となるアプリケーションのサポート力に左右されます。次の質問について検討してみることをお勧めします。

- アプリケーションには、復元力のあるペアまたは分割可能なサーバーグループで構成される複数の層 (フロントエンド、バックエンド、データベース) があるか？

- アプリケーションはロードバランサーを介してアクセスされ、ロードバランサーは AWS ネットワークとオンプレミスネットワークへのトラフィックのルーティングをサポートしていますか？
- データベースや他のバックエンド依存関係へのレイテンシーを AWS 許容するために移行されたアプリケーションサーバー。データベースが にカットオーバーされた場合 AWS、オンプレミスに残っているサーバーまたはサービスは引き続き機能し、必要に応じて動作しますか？

既存のオンプレミス容量 AWS を維持しながら、 で新しく移行されたサーバーにユーザーの一部を送信する機能は、ロールバック機能に関してはall-at-onceアプローチよりも大きな利点があります。移行したサーバーと既存のサーバーを混在させて、負荷を分散してアプリケーションを提供するため、問題が発生した場合は迅速かつ容易に元に戻すことができます。ほとんどの場合、必要なのは、ロードバランサー、DNS ルール、またはポリシーの変更だけです。段階的なカットオーバーアプローチでは、負荷を徐々に増やすこともできます。これにより AWS、アプリケーションチームはアプリケーションのパフォーマンスを評価し、フルロードが転送される前に必要な更新や変更を行うことができます。

アプリケーションまたは依存アプリケーションのスタックを一度にカットオーバーするのが最善か、それともサーバーとサービスを段階的にカットオーバーするアプローチを採用するのが最善か、という選択は、どちらかが万能と言うより、ケースバイケースであると言えるでしょう。次のようなアプローチを採用しているお客様は多いです。

- ある程度のダウンタイムを許容できる開発環境やテスト環境では、一括アプローチによるカットオーバーのほうが、簡単かつ労力が少なく済むというメリットがあります。
- これとは対照的に、段階的アプローチはより複雑で時間がかかりますが、通常はダウンタイム要件が低く、ロールバックオプションも速くなります。このため、ビジネスクリティカルな本番稼働ワークロードでは、段階的なアプローチの採用が最も一般的です。

カットオーバー変更期間前に、時間をかけてソースシステムを理解することを推奨します。計画の初期段階に多くの時間をかけることで、カットオーバーの準備や移行後の検証など、さまざまなプロセスをサポートできます。お客様は、移行時にサーバーの IP アドレスを変更することができます AWS。このシナリオで避けるべき重要な要因は、アプリケーション内で IP アドレスがハードコードされていることです。アップストリームとダウンストリームの両方の依存関係があるソース環境を包括的に見ることを推奨します。例えば、移行したサービスに接続している他のシステムに問題が生じる可能性が高くなります。カットオーバーを開始する前に、すべての接続を移動させて完全修飾ドメイン名 (FQDN) または DNS レコードを使用する価値があるかどうかを検討したほうが良いでしょう。

カットオーバーを実行するタイミング

一般的に、カットオーバーイベントの最適なタイミングは、ユーザー数が最も少ない、つまりビジネスへの影響が最も少ないと思われるときです。ただし、カットオーバー期間中のサポートチームの対応状況とバランスを取る必要があります。トラブルシューティングと潜在的な問題の解決を支援してくれるサポートチームが必要です。また、ステークホルダーの準備状況を踏まえてカットオーバーの日時を考慮することも重要です。予定された日時に準備が整っておらず、対応できないステークホルダーがいれば、カットオーバーが遅れるリスクがあります。

カットオーバー前にテストすべきこと

移行アプローチが許せば、カットオーバー期間前に機能テストと非機能テストを実施するのがベストプラクティスです。例えば、負荷テストツールを活用して、カットオーバー期間前に新しい環境が適切に設定されているかどうかを検証できます。一般に、AWS 環境がライブトラフィックを処理していないため、このフェーズでのテストは中断されません。

カットオーバー前にテストできないこと

他のアプリケーションとの依存関係により、本番稼働で発生するすべてのシナリオをテストできない場合があります。このような場合は、潜在的なリスク、リスクを特定する方法、テストが失敗したときにチームがとる対応策をドキュメント化しておきます。

運用準備状況のレビュー

アプリケーションを にカットオーバーする前に AWS、運用準備状況のレビューを実行することをお勧めします。ここでは、テストの完全性を評価し、チームがアラートをモニタリングして取得する能力を検証するほか、ステークホルダーがワークロードをサポートおよび維持する方法を理解していることを確認します。そのためには、ビジネスチームと技術チームの双方と協力する必要があります。運用準備の詳細については、AWS ドキュメントの [AWS 「Well-Architected」](#) の AWS Well-Architected Tool フレームワークのオペレーショナルエクセレンスの柱」を参照してください。

ロールバック

状況によっては、移行ロールバックが必要になることがあります。潜在的なロールバックに備えて、以下を含むロールバックプランを立てることを推奨します。

- 事前に定義された特定の基準を満たしている場合、カットオーバー中のロールバックを開始する定義済みのチェックポイント

- ロールバックを管理し、データを処理するためのロールバック戦略
- 移行の修正またはロールバックを決定する窓口

カットオーバー中、または新しいデータがない場合のロールバック

ステークホルダーと相談して、データを変更せずロールバックを実行すると決めた場合、ロールバックはオンプレミスインスタンスを再開し、ロードバランサーまたは DNS 設定を変更してトラフィックをリダイレクトするだけの簡単な方法でできます。

変更したデータを使ったロールバック

カットオーバーが正常に完了した後でロールバックが開始され、アプリケーションがすでに新しいトランザクションとデータを受け取っている場合は、データをクラウド環境からオンプレミス環境に復元しなければならない場合があります。このシナリオでは、次のアプローチを検討することを推奨します。

- フェイルフォワードアプローチ — 移行後のデータベースがメインデータベースになるため、オンプレミス AWS データベースはカットオーバー後に古くなる可能性があります。[AWS Database Migration Service \(AWS DMS\)](#) を使用してフェイルフォワードデータベースをセットアップし、データを新しいオンプレミスデータベースにレプリケートできます。問題が発生した場合、AWS DMS は古いオンプレミスデータベースではなく、指定されたフェイルフォワードデータベースにアプリケーションをロールバックします。
- 二重書き込み戦略 — この場合、アプリケーションロジックは古いデータベースと新しいデータベースの両方への書き込みを許可する必要があります。
- ネイティブバックアップおよび復元 — 復元に必要な時間を評価するには、カットオーバー前の段階で下位環境 (非本番稼働環境) を使用したバックアップと復元のテストを実行します。

カットオーバー後の段階

モニタリングダッシュボードの設定

アプリケーションをクラウドに移行した後は、モニタリングを行ってモニタリングダッシュボードに公開すべき情報またはパラメータを、ステークホルダーと相談して決めることをお勧めします。通常、ダッシュボードでは運用に欠かせない主要なサービスレベルアグリーメント (SLA) メトリクスとサービスレベル指標 (SLI) をモニタリングします。到達可能性、可用性、セキュリティ、トラフィックパターン、およびその他の関連データポイントをモニタリングできます。このデータをモニタリングすることで、アプリケーションが期待どおりに動作していることを確認できるだけでなく、アプリケーションが期待通りの品質を実現しているかどうかも明らかになります。

成功を理解し伝えること

ワークロードの移行が正常に完了したら、関連するステークホルダーに無事成功したことを伝えましょう。ステークホルダーには、シングルスレッドのリーダー、スポンサー、エンドユーザー、ワークロードアプリケーションの所有者、およびその他の依存関係にあるアプリケーションの所有者が含まれる場合があります。

また、今後のアプリケーション移行プロセスの改善に役立てられそうな学びについても、検討しておく価値があります。移行スコアカードを使用すれば、移行の成功度を測定し、複数のアプリケーションや事業部門に対して、移行の全体的な進捗状況を伝えることができます。この取り組みは、チームが互いに学び合い、移行リスクを最小限に抑え、ベストプラクティスを採用する機会を探すきっかけとなります。

保証期間の定義

移行プロジェクトでは、あらかじめ決められた期間 (通常は 1 日から 1 週間) 中に問題が発生した場合に備え、移行チームがサポートを提供する保証期間を設けるのが一般的です。移行するアプリケーションに基づいて、適切な期間での合意が重要となります。例えば、アプリケーションが四半期ごとのバッチスケジュールで実行される場合、1 週間の保証では不十分な場合があります。さらに、移行チームは保証期間を利用して、ビジネス継続性とディザスタリカバリの要素が想定どおりに設定され、機能しているかどうかを検証することを推奨します。

結論

カットオーバーは複雑なアクティビティです。トラフィックを新しいシステムに再ルーティングするだけではありません。この複雑さは、ビジネスの重要性、アップストリームシステムとダウンストリームシステムの数、ユーザー数など、多くの要因に左右されます。カットオーバーフェーズでは、人、プロセス、テクノロジーを取り巻くあらゆる複雑さをチームが理解することが重要です。これらの複雑さに基づいて、移行のカットオーバー前、カットオーバー、カットオーバー後のアクティビティに関するコミュニケーションとガバナンスの計画を特定、策定、採用することができます。最後に、オンプレミスシステムに関する既存の知識を用いて、カットオーバーにかかる時間の見積もりを出しておくことを推奨します。カットオーバーの望ましい結果から逆算して取り組むことが役に立つ場合もあります。

リソース

- [AWS WA ツール](#)
- [DMS AWS を使用した移行からのロールバック](#)
- [AWS Database Migration Service](#)

ドキュメント履歴

以下の表は、本ガイドの重要な変更点について説明したものです。今後の更新に関する通知を受け取る場合は、[RSS フィード](#) をサブスクライブできます。

変更	説明	日付
初版発行	—	2022 年 10 月 31 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。