AWS Well-Architected フレームワークを Amazon WorkSpaces アプリケーションに適用する

AWS 規範ガイダンス

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

AWS 規範ガイダンス: AWS Well-Architected フレームワークを Amazon WorkSpaces アプリケーションに適用する

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
対象者 1
目的 2
運用上の優秀性の柱 3
ビジネス成果に関するチームを編成する 3
実用的なインサイトのためのオブザーバビリティを実装する 4
可能な限り安全に自動化する 5
頻繁で小さく、可逆的な変更を行う 7
オペレーション手順を頻繁に絞り込む 8
失敗を予測する 8
すべての運用イベントとメトリクスから学ぶ 10
マネージドサービスを使用する 10
セキュリティの柱 11
強力な ID 基盤を実装する 11
トレーサビリティを維持する 12
すべてのレイヤーにセキュリティを適用する 13
セキュリティのベストプラクティスを自動化する 14
データから遠ざける 14
セキュリティイベントに備える 15
信頼性の柱 17
障害から自動的に復旧する 17
復旧手順をテストする 18
ワークロードの全体的な可用性を高めるために水平方向にスケールする 18
容量の推測を停止する 19
自動化による変更の管理 19
パフォーマンス効率の柱 20
高度なテクノロジーの民主化 20
数分でグローバル化 21
サーバーレスアーキテクチャを使用する 21
より頻繁に実験する 22
機械的な交感神経を検討する 22
コスト最適化の柱 24
クラウド財務管理はどのように実装しますか? 24
消費モデルを追加する 24

全体的な効率を測定する 25
差別化されていない重い作業にお金を費やすのをやめる 25
支出の分析と属性付け 26
持続可能性の柱 27
影響を把握する 27
持続可能性の目標を設定する 27
使用率を最大化する 28
より効率的な新しいハードウェアおよびソフトウェアサービスを予測して採用する 28
使用済みマネージドサービス 28
クラウドワークロードのダウンストリームへの影響を軽減する 29
リソース 30
AWS ドキュメント 30
AWS ブログ投稿 30
ドキュメント履歴 33
..... xxxiv

AWS Well-Architected フレームワークを Amazon WorkSpaces アプリケーションに適用する

モハメド・ウォリ、アマゾン ウェブ サービス

2025 年 7 月 (ドキュメント履歴)

このガイドでは、Amazon WorkSpaces アプリケーションを使用する際に AWS Well-Architected フレームワークを適用するためのベストプラクティスについて説明します。WorkSpaces Applications は、デスクトップアプリケーションを書き換えることなくユーザーにストリーミングできるフルマネージド型のアプリケーションストリーミングサービスです。

AWS Well-Architected フレームワークは、クラウドアーキテクトがさまざまなアプリケーションやワークロードのための安全で高性能、回復力があり、効率的なインフラストラクチャを構築するのに役立ちます。また、ユーザーと AWS パートナーがアーキテクチャを評価し、スケーラブルな設計を実装するための一貫したアプローチも提供します。

AWS Well-Architected フレームワークは、次の 6 つの柱を中心に構築されています。

- オペレーショナルエクセレンス
- セキュリティ
- 信頼性
- パフォーマンス効率
- コスト最適化
- 持続可能性

このガイドでは、これらの柱とベストプラクティスが WorkSpaces アプリケーションの使用にどのように適用されるかについて説明します。

対象者

このガイドは以下を対象としています。

- WorkSpaces アプリケーションソリューションを設計および実装し、アーキテクチャが AWS Well-Architected Framework のベストプラクティスに従っていることを確認する必要があるクラウドアーキテクトとエンジニア。

- WorkSpaces アプリケーション環境を管理および保守し、フリートの管理、スケーリング、モニタリングを処理し、コストとパフォーマンスを最適化する必要がある IT 運用チーム。
- WorkSpaces アプリケーションを検討中または既に使用しており、デスクトップアプリケーションをユーザーにストリーミングしたいと考えている組織や企業は、安全で高性能、回復力があり、効率的なインフラストラクチャを構築する必要があります。

目的

このガイドのベストプラクティスに従うと、次のことに役立ちます。

- でデスクトップアプリケーションをストリーミングするための、安全で高性能、回復力があり効率的なインフラストラクチャを構築します AWS クラウド。
- WorkSpaces アプリケーションアーキテクチャを評価し、スケーラブルな設計を実装するときは、一貫したアプローチを適用します。

運用上の優秀性の柱

運用上の優秀性 (OE) は、ユーザーの期待に一貫して応え、それを超える高品質のソフトウェアソリューションを作成することに専念しています。 AWS Well-Architected フレームワークの運用上の優秀性の柱には、効果的なチーム組織、堅牢なワークロード設計、効率的な大規模な運用、時間の経過とともに変化する要件へのシームレスな適応のための実証済みの戦略が含まれています。これらの原則に従うことで、組織はシステムの耐障害性、パフォーマンス、進化するビジネスニーズとの整合性を維持できます。

この柱を WorkSpaces アプリケーションストリーミング環境に適用するための主な重点領域:

- モニタリングとオブザーバビリティ
- 自動化と DevOps
- 運用手順とドキュメント
- サポートとインシデント管理

ビジネス成果に関するチームを編成する

ビジネス目標と主要業績評価指標 (KPIs) が最適化された人材、プロセス、テクノロジーを通じて組織の変革を促進する、強力なリーダーシップコミットメントを持つクラウドに沿った運用モデルを作成します。

- チーム構造。アプリケーションストリーミングの結果に沿った専用チームを確立します。例:
  - イメージ管理チームは、アプリケーションのパッケージングとイメージの最適化を担当します。
  - フリート運用チームは、容量、パフォーマンス、スケーリングを管理します。
  - ユーザーエクスペリエンスチームは、エンドユーザーのサポートと満足度を処理します。
- KPIsとメトリクス。次のようなビジネスに沿ったメトリクスを定義して追跡します。
  - アプリケーションの可用性レート
  - 新しいアプリケーションをデプロイする時間
  - アプリケーションストリーミング時間あたりのコスト
- 運用モデル。以下の明確なプロセスを作成します。
  - アプリケーションのオンボーディングと更新
  - フリート容量管理
  - ユーザーアクセスのプロビジョニング

- インシデント対応と解決

実用的なインサイトのためのオブザーバビリティを実装する

包括的なモニタリングとオブザーバビリティを実装して、KPIs。この原則により、データ駆動型の意思決定が可能になり、パフォーマンス、信頼性、コストにわたってプロアクティブな改善が可能になります。

- パフォーマンスモニタリングを実装します。Amazon CloudWatch を次のように設定します。
  - 需要を満たすのに十分な容量を確保します。たとえば、次のメトリクスを使用できます。
    - `AvailableCapacity` 利用可能なストリーミングインスタンスをモニタリングするには
    - `InUseCapacity` 現在使用されているインスタンスを追跡するには
    - `CapacityUtilization` フリート使用率をモニタリングする
  - ユーザーエクスペリエンスとパフォーマンスをモニタリングします。
  - サービスの問題を迅速に特定して対処します。
- WorkSpaces アプリケーション使用状況レポートを追跡および分析します。
- アプリケーションログをキャプチャして分析します。詳細については、 AWS ブログ記事「Using Kinesis Agent for Linux to stream application logs in WorkSpaces Applications」および「Using Kinesis Agent for Microsoft Windows to store WorkSpaces Applications Windows event logs」を参照してください。
- チャット通知を使用して WorkSpaces アプリケーションのメトリクスとイベントをモニタリングします。詳細については、 AWS ブログ記事「Chatbot による AWSAWS エンドユーザーコンピューティング (EUC) のモニタリングと自動化」を参照してください。
- ビジュアルキューを使用してプロアクティブセッション管理を有効にします。詳細については、AWS ブログ記事Amazon WorkSpaces アプリケーションでセッションの有効期限とカウントダウンタイマーを表示する」を参照してください。
- 使用パターンと傾向の視覚化を作成します。詳細については、 AWS ブログ記事「Amazon OpenSearch Service での Amazon WorkSpaces アプリケーション使用状況レポートの取り込みと視覚化」を参照してください。
- EUC ツールキットを使用して、アクティブなセッションのモニタリング、フリートインベントリの追跡、セッションレポートの生成 (CSV エクスポート) を行います。詳細については、AWS ブログ記事「EUC Toolkit を使用して Amazon WorkSpaces アプリケーションと Amazon WorkSpaces を管理する」を参照してください。

可能な限り安全に自動化する

Infrastructure as Code (IaC) の原則を適用して、ワークロードオペレーションのあらゆる側面を自動化します。ガードレールを使用すると、手動による介入を減らしながら、安全で一貫した実行を確保できます。

- Image Assistant CLI を使用して、WorkSpaces アプリケーションイメージの作成と設定を自動化します。詳細については、WorkSpaces Applications ドキュメントの「Image Assistant CLI オペレーションを使用してプログラムで Amazon WorkSpaces アプリケーションイメージを作成する」を参照してください。 WorkSpaces
  - アプリケーションのインストール: Image Assistant CLI を使用して、イメージの作成中にアプリケーションのインストールを自動化します。
  - イメージの作成: Image Assistant CLI コマンドを使用して、プログラムで WorkSpaces アプリケーションイメージを作成します。
  - 設定管理: デフォルトのアプリケーション設定と起動パラメータの設定を自動化します。
- WorkSpaces Applications イメージのカスタマイズを自動化します。詳細については、 AWS ブログ記事「カスタマイズされた WorkSpaces Applications Windows イメージを自動的に作成する」を参照してください。
- IaC を適用して、WorkSpaces アプリケーションのインフラストラクチャとアプリケーションコンポーネントをデプロイします。詳細については、 AWS ブログ記事「Automation of infrastructure and application deployment for Amazon WorkSpaces Applications with Terraform」を参照してください。
- 以下を含むフリート管理の自動プロセスを実装します。
  - 需要に基づくフリートスケーリング。自動スケーリングポリシーを設定して、使用率メトリクスに基づいてフリート容量を自動的に調整します。詳細については、 AWS ブログ記事「Use AWS Lambda to adjust scaling steps and thresholds for Amazon WorkSpaces Applications」を参照してください。
  - ベースイメージの更新。が提供する WorkSpaces Applications ベースイメージの自動更新を利用できます AWS。
  - 容量の最適化。自動スケーリングしきい値を設定して、需要パターンに基づいてリソース使用量を最適化します。
- 安全制御を自動化するようにガードレールを設定します。
  - 最大フリートサイズ制限。過剰プロビジョニングを防ぐために、フリート容量の上限を設定します。

  - スケーリングポリシーの設定。適切なしきい値を使用して、ステップスケーリングまたはターゲット追跡スケーリングポリシーを実装します。
  - サービスクォータ。 AWS 過剰なリソース割り当てを防ぐために、サービスクォータを組み込みの制限として使用します。
  - スケールイン保護。スケールイン保護を設定して、スケーリングイベント中にアクティブなインスタンスが削除されないようにします。
- Image Builder、フリート、統合テストなど、テストと検証を実行します。
  - Image Builder のテスト:
    - Image Builder インターフェイスで直接アプリケーションをテストします。
    - アプリケーションの起動と機能を確認します。
    - ユーザー設定と設定をテストします。
    - アプリケーションの互換性を検証します。
  - フリートテスト:
    - さまざまなクライアントデバイスからストリーミングセッションをテストします。
    - ユーザーの使用権限とアクセスを確認します。
    - アプリケーションのパフォーマンスを検証します。
    - クリップボード、ファイル転送、印刷などの要素とオペレーションのユーザーエクスペリエンスをテストします。
  - 統合テスト:
    - Active Directory または SAML 2.0 ベースの認証をテストします。
    - ホームフォルダと永続的ストレージをテストします。
    - アプリケーションのエンタイトルメントをテストします。
    - USB デバイスリダイレクトをテストします (設定されている場合) 。
- WorkSpaces アプリケーションマネージャーを使用して、アプリケーションのパッケージ化とデプロイを自動化します。詳細については、 AWS ブログ記事Amazon WorkSpaces Applications のアプリケーションマネージャーによるアプリケーションのオンボーディングを効率化する」を参照してください。
- 継続的インテグレーションと継続的デリバリー (CI/CD) パイプラインを使用して、新しいアプリケーションバージョンのデプロイを自動化します。詳細については、 AWS ブログ記事「Eagle: Optimize CI/CD and end user experience in Amazon WorkSpaces Applications」を参照してください。

頻繁で小さく、可逆的な変更を行う

最小限のリスクと簡単なロールバック機能で、頻繁で小規模な自動デプロイを可能にする、疎結合でスケーラブルなワークロードを構築します。

- イメージの更新には、バージョニングされたイメージの作成と増分更新を使用します。
  - バージョニングされたイメージの作成:
    - Image Builder を使用して、変更のセットごとに新しいイメージを作成します。
    - ロールバックシナリオをサポートするために、複数のイメージバージョンを維持します。
    - AWS タグ付け戦略を使用して、イメージのバージョンと属性を追跡します。
  - 増分更新:
    - アプリケーションまたは設定に小さな増分変更を加えます。
    - 新しいイメージを作成する前に、Image Builder で更新を徹底的にテストします。
    - 各新しいイメージバージョンで行ったすべての変更を文書化します。
- コントロールフリートの更新の場合:
  - テスト用に更新されたイメージを使用して新しいフリートを作成します。
  - アクティブなセッションを中断することなく、既存のフリート属性を変更します。
- ドキュメント、テストプロトコル、承認ワークフロー、モニタリングプロセスに関する変更管理手順を確立します。
  - ドキュメント:
    - すべてのイメージとフリートの更新の詳細な変更ログを維持します。
    - 各変更のテスト手順と結果を文書化します。
    - AWS CloudTrail を使用して、設定の変更を追跡および監査します。
  - プロトコルのテスト:
    - すべての変更に対して包括的なテストプロセスを確立します。
    - アプリケーションの機能、パフォーマンス、ユーザーエクスペリエンステストを含めます。
    - 新しいイメージを作成する前に、Image Builder でテストを実行します。
    - フルデプロイの前に、非本番稼働フリートで追加のテストを実行します。
  - 承認ワークフロー:
    - 本番環境への変更の承認プロセスを実装します。
    - 承認と標準の更新を必要とする変更の基準を定義します。
    - 変更承認の役割と責任を確立します。

- モニタリングと検証:
  - Amazon CloudWatch を使用して、変更後のフリートとアプリケーションのパフォーマンスをモニタリングします。
  - 主要メトリクスのアラートを設定して、更新後に問題をすばやく特定します。
  - 実装後のレビューを実施して、変更の成功を検証し、学習情報を収集します。

オペレーション手順を頻繁に絞り込む

定期的なレビュー、更新、チームエンゲージメントを通じて運用手順を継続的に改善し、すべてのステークホルダーに最新情報を提供し、ベストプラクティスに合致させます。

- ドキュメント管理。WorkSpaces Applications の手順に関するバージョン管理された最新のドキュメントを一元管理し、運用上の一貫性とチーム間の知識共有を確保します。
  - 必要なドキュメント: イメージの作成と管理、フリートオペレーション、トラブルシューティングのための重要な WorkSpaces アプリケーションオペレーションに関するup-to-dateドキュメントを維持します。
  - 運用レビュー: パフォーマンスメトリクスやインシデント管理など、主要な運用面を監視およびレビューします。
- 継続的な改善。 AWS のサービス 更新、運用メトリクス、学習したベストプラクティスを標準手順に組み込むことで、WorkSpaces アプリケーションの運用を体系的に強化します。
  - サービスの更新: WorkSpaces アプリケーションのリリースノートをモニタリングして、新機能、サービスの改善、セキュリティの更新、リージョンの可用性を確認します。
  - ベストプラクティス: AWS Well-Architected Framework の更新、WorkSpaces アプリケーションのベストプラクティス、 AWS リファレンスアーキテクチャ、および AWS セキュリティの推奨事項を確認して組み込みます。
  - ナレッジ管理: 標準運用手順、ランブック、トラブルシューティングガイド、およびユーザーサポートドキュメントを維持および更新します。

失敗を予測する

障害シナリオテストを定期的に実施して、リスクを理解し、対応手順を検証し、実際のインシデントを処理するためのチームの準備状況を向上させます。

- 失敗テスト。フリート容量の枯渇、アプリケーションの起動障害、ネットワーク接続の問題などの障害を定期的にシミュレートしてテストします。
  - フリート容量の枯渇:
    - 容量制限に近づいたときのフリートスケーリング動作をモニタリングしてテストします。
    - `CapacityUtilization` および `AvailableCapacity`メトリクスの CloudWatch アラームを設定します。
    - ピーク時の使用時に容量の制約を処理する手順を実装します。
  - アプリケーション起動の失敗:
    - ストリーミングインスタンスでアプリケーションの起動動作をテストします。
    - さまざまなフリート設定でアプリケーションアクセスとパフォーマンスを検証します。
  - ネットワーク接続の問題:
    - さまざまなネットワーク条件でストリーミングセッションのパフォーマンスをテストします。
    - 接続品質の問題`StreamingSessionLatency`を監視します。
    - VPC 設定とセキュリティグループが適切に設定されていることを確認します。
- 復旧手順。以下の手順を開発してテストします。
  - 間のフリートフェイルオーバー AWS アベイラビリティーゾーン。さらに、フリート容量のスケーリング、フリートの更新の管理、インスタンスのヘルス問題への対応の手順を文書化します。
  - ユーザーデータ管理:
    - Windows フリートの場合は Amazon Simple Storage Service (Amazon S3) のホームフォルダのアプリケーション設定の永続性とストレージソリューションを設定し、Linux フリートの場合は Amazon Elastic File System (Amazon EFS) の共有ファイルシステムを設定します。
    - セッション間のデータ同期を検証します。
  - サービスの継続性。新しいフリートインスタンスの作成、イメージの更新の管理、セッションの切断の処理の手順を維持します。
- リスク管理。以下を特定して軽減します。
  - 適切なフリート最小容量の設定、需要パターンに基づく自動スケーリングポリシーの設定、`CapacityUtilization`、、 などの CloudWatch メトリクスを使用したフリート使用率の傾向のモニタリングによる容量の制約`InUseCapacity``AvailableCapacity`。
  - などの主要なメトリクスを追跡`StreamingSessionLatency`し、適切な CloudWatch アラームを設定することで、パフォーマンスのボトルネックが発生します。

すべての運用イベントとメトリクスから学ぶ

運用上のイベントや失敗から学んだ教訓を組織全体で共有することで、継続的な改善の文化を育みます。ビジネス成果への影響を強調します。

- イベント分析。サービスの中断、パフォーマンスの低下、ユーザーの苦情、容量の問題を文書化して分析します。
- メトリクスのレビュー。使用状況パターン、パフォーマンス傾向、コストメトリクス、ユーザー満足度データを定期的に分析します。
- ナレッジ共有。チーム学習セッション、ベストプラクティスドキュメント、チーム間の知識移転、インシデントの遡及のためのプロセスを確立します。

マネージドサービスを使用する

AWS マネージドサービスを使用し、その周囲に標準化された手順を構築することで、運用上のオーバーヘッドを最小限に抑えます。を次の AWS マネージドサービスと統合します。

- AWS Systems Manager オートメーション用
- モニタリング用の Amazon CloudWatch
- アクセスコントロールの AWS Identity and Access Management (IAM)
- Windows フリートのユーザーストレージ用 Amazon S3
- Linux フリートのユーザーストレージ用の Amazon EFS
- AWS Directory Service ユーザー認証用

セキュリティの柱

AWS Well-Architected フレームワークのセキュリティの柱は、クラウド機能を活用して、情報、インフラストラクチャ、リソースの堅牢な保護メカニズムを確立することに焦点を当てています。これらの原則は、イノベーションを可能にしながら、全体的なセキュリティ体制を強化するのに役立ちます。

この柱を WorkSpaces Applications ストリーミング環境に適用するための主な重点領域:

- データの整合性と機密性
- ユーザーアクセス許可の管理
- セキュリティイベントを検出するためのコントロールの確立

強力な ID 基盤を実装する

ID 管理を一元化し、長期的な認証情報を回避しながら AWS 、リソースにアクセスするために必要な最小限のアクセス許可を使用します。

- WorkSpaces アプリケーションリソースに最小特権のアクセス許可を付与します。
  - 最小限のアクセス許可で WorkSpaces アプリケーションフリートの特定の IAM ロールを作成します。
  - Image Builder の制限された IAM アクセス許可を設定します。
  - WorkSpaces アプリケーション管理機能への管理アクセスを制限します。
  - スタックとフリート管理の詳細なアクセス許可を定義します。
- 適切なユーザー認証メカニズムを実装します。
  - エンタープライズ ID プロバイダーの統合用に SAML 2.0 フェデレーションを設定します。
  - ユーザー管理AWS IAM アイデンティティセンター用に をセットアップします。
  - 特定の認証シナリオで必要な場合のみ、カスタム ID ブローカーを使用します。
  - サポートされている場合は、多要素認証 (MFA) を実装します。
- アプリケーションへのユーザーアクセスを制御します。
  - アプリケーションの使用権限を設定して、特定のアプリケーションへのアクセスを制限します。
  - ユーザーロールに基づいてアプリケーション割り当てグループを作成します。
  - スタックのアクセス許可を使用してアプリケーションアクセスを管理します。

    - アプリケーションの動作を制御するセッションポリシーを実装します。
- 適切なコントロールを使用してユーザーセッションを保護します。
    - セッションタイムアウトポリシーを設定します。
    - 切断タイムアウトアクションを設定します。
    - セッション永続性要件を実装します。
    - ファイルシステムのリダイレクト許可を制御します。
- WorkSpaces アプリケーションの証明書ベースの認証を設定します。詳細については、 AWS ブログ記事「Simplify certificate-based authentication for WorkSpaces Applications and WorkSpaces with AWS Private CA Connector for Active Directory」を参照してください。
- セッションタグを使用して、きめ細かなアクセスコントロールを実装します。詳細については、AWS ブログ記事「セッションタグを使用して WorkSpaces アプリケーションのアクセス許可を簡素化する」を参照してください。

トレーサビリティを維持する

すべての環境の変更とアクティビティに対して、リアルタイムモニタリングと自動応答システムを実装します。

- アプリケーションログの CloudWatch ログ記録を設定して、アプリケーションの起動、クラッシュ、エラーなど、アプリケーション固有のイベントをモニタリングします。セッションの開始、停止、ユーザー接続イベントなど、ストリーミングセッション情報を追跡するようにセッションログを設定します。
- CloudTrail をアクティブ化してすべての WorkSpaces Applications API コールをログに記録し、フリートの作成と変更、Image Builder オペレーション、スタック設定、ユーザー管理アクティビティなどの管理イベントを追跡します。
- WorkSpaces Applications インスタンスのアクティビティをモニタリングします。
    - システムレベルのイベントをキャプチャするようにインスタンスのログ記録を設定します。
    - アプリケーションの起動と失敗を追跡します。
    - システムリソースの使用状況とパフォーマンスをモニタリングします。
- ユーザーアクティビティを追跡します。
    - ユーザー認証の試行と失敗をモニタリングします。CloudWatch メトリクスと CloudWatch Logs を使用して、ユーザーのログイン試行、セッションの開始時刻と終了時刻、セッション切断イベントを追跡します。

  - アプリケーションの使用パターンを追跡します。WorkSpaces アプリケーション使用状況レポートを有効にして、セッション期間、開始時刻と終了時刻、使用されるインスタンスタイプ、アクセスされたアプリケーションなどの情報を取得します。
  - 有効なホームフォルダを使用してファイルシステムのアクティビティを記録します。
  - データ損失防止の目標を達成するために、クリップボード設定と印刷オペレーションを設定します。
- ユーザー認証の失敗、異常なセッションパターン、リソースアクセス違反などのセキュリティ関連のメトリクスに対して CloudWatch アラームを設定します。
- EUC ツールキットを使用して、アクティブなセッションと状態を追跡し、使用中のアクティブなセッションの IP アドレスをモニタリングし、監査のためにセッションデータをエクスポートします。詳細については、 AWS ブログ記事「Using the EUC toolkit to manage Amazon WorkSpaces Applications and Amazon WorkSpaces」を参照してください。

すべてのレイヤーにセキュリティを適用する

ネットワークエッジからアプリケーションコードまで、インフラストラクチャのすべてのコンポーネントに複数のセキュリティコントロールレイヤーを実装します。

- ネットワークレイヤーのセキュリティを設定します。
  - 厳格なセキュリティグループルールを実装します。
  - WorkSpaces Applications フリートインスタンスを、直接インターネットにアクセスできないプライベートサブネットに配置します。NAT デバイスによるインターネットアクセスを制御します。
  - 仮想プライベートクラウド (VPC) エンドポイントを使用して、サポートされている AWS のサービス (Amazon S3 など) にアクセスします。
  - 追加のネットワークセキュリティレイヤーとしてネットワークアクセスコントロールリスト (ACLs) を実装します。
  - ストリーミングポート (HTTPS および WebSocket Secure の場合は TCP 8443) アクセスを特定の IP 範囲に制限します。
- アクセスレイヤーのセキュリティを設定します。
  - セッションタイムアウトポリシーを実装して、非アクティブなユーザーを自動的に切断します。
  - セッションタグを使用して属性ベースのアクセスコントロールを実装します。詳細については、AWS ブログ記事「セッションタグを使用して WorkSpaces アプリケーションのアクセス許可を簡素化する」を参照してください。

- アプリケーションレイヤーのセキュリティを設定します。
  - アプリケーションの使用権限を設定して、特定のアプリケーションにアクセスできるユーザーを制御します。
  - ファイルシステムのリダイレクトコントロールを有効にして、ローカルドライブへのアクセスを制限します。
  - セキュリティ要件に基づいて、クリップボード、ファイル転送、印刷のアクセス許可を設定します。
  - セキュリティポリシーに従って USB デバイスアクセスコントロールを設定します。
- イメージレイヤーのセキュリティを設定します。
  - セキュリティ要件を満たす強化ベースイメージを作成して維持します。
  - ベースイメージを最新のセキュリティパッチで更新します。
  - ベースイメージで Windows セキュリティ設定を構成します。
  - ベースイメージの不要な Windows サービスと機能を無効にします。

セキュリティのベストプラクティスを自動化する

バージョン管理されたテンプレートでコード定義の自動セキュリティコントロールを使用して、安全でスケーラブルなインフラストラクチャのデプロイを可能にします。

- などのサービスを使用してInfrastructure as Code (IaC) を使用し AWS CloudFormation 、すべてのフリートデプロイに一貫したセキュリティ設定を実装します。詳細については、 AWS ブログ記事「Amazon Amazon WorkSpaces アプリケーションと Amazon WorkSpaces に追加のセキュリティグループを自動的にアタッチする」を参照してください。
- Image Assistant CLI を使用して、イメージ作成のセキュリティプロセスを自動化します。
- Amazon CloudWatch アラーム、Amazon EventBridge ルール、自動応答用の AWS Lambda 関数を使用して、容量使用率のしきい値を超えた自動応答、不正アクセスの試行、セキュリティグループの変更を設定します。

データから遠ざける

データ処理プロセスを自動化して、直接の人間によるアクセスを最小限に抑え、エラーや誤処理のリスクを軽減します。

- アプリケーションの使用権限を設定して、特定のアプリケーションにアクセスできるユーザーを制御します。
- 動的アプリケーションフレームワークを使用して動的アプリケーションプロバイダーを構築し、ユーザー属性に基づいてアプリケーションを動的に利用できるようにします。
- ファイルシステムのリダイレクトを設定して、ユーザーがアクセスできるローカルドライブを制御し、特定のフォルダへのアクセスを制限し、ローカルセッションとストリーミングセッション間のファイル転送アクセス許可を管理します。
- クリップボードの制限を実装して、ローカルセッションとストリーミングセッション間のクリップボード共有を無効にし、必要に応じて一方向クリップボードフローを有効にして、不正なデータコピーを防止します。
- アプリケーション設定の永続性を設定して、アプリケーション設定を自動的に保存および復元し、手動設定の必要性を排除し、一貫したユーザーエクスペリエンスを維持します。

セキュリティイベントに備える

自動ツールを使用してインシデント対応計画を策定して実践し、セキュリティイベントを迅速に検出、調査、復旧できるようにします。

- 認証試行の失敗、フリートセキュリティグループの変更、イメージ設定の変更、異常なストリーミングセッションパターンに対して CloudWatch アラームを設定します。
- 次のような一般的な WorkSpaces アプリケーションセキュリティシナリオの応答手順を文書化します。
  - 許可されていないアクセスの試行
    - 検出: 認証の失敗をモニタリングします。
    - レスポンス: ユーザー権限の取り消し、セッションログの確認、アクセスポリシーの更新を行います。
  - 侵害されたストリーミングインスタンス
    - 検出: インスタンスの動作をモニタリングします。
    - レスポンス: 影響を受けるセッションを終了し、フリートインスタンスを置き換え、セキュリティグループ設定を確認します。
  - データ流出の試行
    - 検出: ファイル転送アクティビティをモニタリングします。
    - レスポンス: クリップボードとファイル転送ログの確認、ファイル転送アクセス許可の調整、データ保護ポリシーの更新を行います。

- フリートインスタンスの交換、セキュリティグループの復元、ユーザーアクセスの再設定、アプリケーション設定の復旧のための自動復旧プロセスを実装します。
- セキュリティの検出 AWS のサービス 結果には 、脅威の検出 AWS Security Hub CSPM には Amazon GuardDuty などのセキュリティ管理に使用します。

信頼性の柱

AWS Well-Architected フレームワークの信頼性の柱は、システムが意図した機能とパフォーマンスレベルを、その存続期間を通じて期待される運用期間中にどの程度維持しているかに対処します。ワークロードライフサイクルのすべての段階でテストと検証の戦略など AWS、 で信頼性の高いシステムを構築および維持するための包括的なガイドラインを提供します。

この柱を WorkSpaces アプリケーションストリーミング環境に適用するための主な重点領域:

- フリートの管理とスケーリング
- セッションの信頼性
- アプリケーションの可用性
- 復旧手順

障害から自動的に復旧する

ビジネス価値KPIs をモニタリングして、障害がオペレーションに影響を与える前に障害を予測、防止、または復旧できる自動応答をトリガーします。

- IP サブネットを割り当てる際には、拡張性と可用性を考慮するようにします。
- 重要な CloudWatch メトリクスをモニタリングして、 や などのフリート容量メトリクス`AvailableCapacity`、 などのストリーミング品質メトリクスなど`InUseCapacity`、サービスの可用性とパフォーマンスを確保します`StreamingSessionLatency`。
- キャパシティのしきい値、セッションヘルスメトリクス、パフォーマンスの低下、フリートのヘルスステータスの変更に関するアラートを設定します。
- 組み込み WorkSpaces アプリケーションの自動スケーリング機能を使用して、次のことを行います。
  - 最小および最大フリート容量を設定します。
  - 容量使用率に基づいてスケーリングポリシーを設定します。
  - 技術的なメトリクスだけでなく、ユーザーエクスペリエンスメトリクスとビジネス要件に基づいてスケールアウトとスケールインのしきい値を定義します。
- WorkSpaces アプリケーション環境のディザスタリカバリ環境を構築します。詳細については、AWS ブログ記事Amazon WorkSpaces アプリケーションのディザスタリカバリに関する考慮事項」を参照してください。

復旧手順をテストする

クラウド環境により、障害シナリオと復旧手順の自動テストが可能になります。これらの機能は、実際の障害が発生する前に脆弱性を特定して修正するのに役立ちます。

- フリート復旧テスト。複数のシナリオにわたって包括的なフリート復旧テストを実装します。
  - インスタンスの終了をシミュレートして、自動スケーリングレスポンスを検証します。
  - フリートの最小容量のメンテナンスを検証します。
  - インスタンスの置き換えタイミングとユーザーリダイレクトをテストします。
  - スケーリングポリシーの有効性を検証します。
  - フリートの容量制限とオーバーフロー処理をテストします。
- セッション復旧テスト。セッション復旧の検証手順を実装します。
  - 切断と再接続のシナリオをテストします。
  - アプリケーションの状態の保存を確認します。
  - さまざまなネットワーク中断シナリオをテストします。
  - セッションタイムアウトの動作を検証します。
  - ユーザー認証の永続性を検証します。
  - 一時的なストレージ処理を確認します。

ワークロードの全体的な可用性を高めるために水平方向にスケールする

ワークロードを複数の小規模なリソースに分散して、個々の障害の影響を最小限に抑え、単一障害点を排除します。

- 複数のアベイラビリティーゾーンにフリートインスタンスをデプロイします。
- 適切な最小フリート容量を設定します。
- フリートの自動スケーリングを設定し、適切なスケーリングしきい値を設定します。
- フリート全体のキャパシティ使用率をモニタリングします。
- WorkSpaces アプリケーションスタックを複数のリージョンにデプロイします。詳細については、AWS ブログ記事「Amazon WorkSpaces アプリケーションのレイテンシーベースのルーティングによるユーザーエクスペリエンスの最適化」を参照してください。

容量の推測を停止する

クラウドの自動スケーリング機能を使用して、需要に基づいてリソースを動的に調整します。これにより、最適な効率を維持しながら、リソースの飽和を防ぐことができます。

- CapacityUtilization、、 などの主要なメトリクスをモニタリングInUseCapacityしてAvailableCapacity、容量のニーズを把握します。
- さまざまな期間におけるフリート使用率の傾向を追跡します。毎日のパターン、毎週のバリエーション、毎月の傾向、季節的なピークをモニタリングします。
- スケーリングポリシーを設定し、スケーリングしきい値を設定します。
- フェイルオーバーを行える十分な最大使用量に配慮して、現在のクォータが設定されていることを確認します。
- アーキテクチャ全体で、固定のサービスクォータおよび制約に対応します。

自動化による変更の管理

自動化コード自体のバージョン管理された変更を含む、自動化によるインフラストラクチャの変更を実装します。

- フリート設定には IaC を使用します。
- 一貫したスケーリングポリシーを実装します。
- Image Assistant CLI を使用して、一貫したイメージを作成します。

パフォーマンス効率の柱

AWS Well-Architected フレームワークのパフォーマンス効率の柱は、需要の変動や新たなテクノロジーへの適応性を確保しながら、パフォーマンス目標を達成または上回るようにクラウドリソースの使用を最適化することに重点を置いています。動的なクラウド環境でピーク効率を維持するために、システムを継続的に微調整することの重要性を強調しています。

この柱を WorkSpaces アプリケーションストリーミング環境に適用するための主な重点領域:

- インスタンスタイプの選択と最適化
- ストリーミングパフォーマンスの最適化
- フリート容量管理

高度なテクノロジーの民主化

複雑なテクノロジーのクラウドベンダーマネージドサービスを活用して、チームがインフラストラクチャ管理ではなく製品開発に集中できるようにします。

- アプリケーションの要件に基づいて適切なインスタンスタイプを設定します。
  - グラフィックを多用するアプリケーションには GPU 対応インスタンスを選択します。
  - アプリケーションのニーズに基づいて、適切な GPU ファミリー (Graphics G4dn や Graphics G5 など) を選択します。
- 次のいずれかの認証方法を選択して設定します。
  - SAML 2.0 ベースの ID プロバイダーとの統合を設定します。
  - ユーザープールの設定を行います。
  - を と統合しますAWS Directory Service。
- ユーザーのニーズに基づいてストレージオプションを有効にして設定します。
  - Amazon S3 for Windows ベースのフリートにホームフォルダを設定します。
  - Linux ベースのフリート用に Amazon EFS で共有ファイルシステムを設定します。
  - 永続的ストレージのアクセス許可を設定します。
  - アプリケーション設定の永続化を有効にします。

数分でグローバル化

マルチリージョンデプロイを使用すると、レイテンシーが短縮され、グローバルユーザーエクスペリエンスが向上します。

- ユーザーに最も近いリージョンにフリートをデプロイし、リージョンごとに個別のスタックを作成 AWS リージョン することで、複数の にフリートを設定します。
- クロスリージョンリダイレクトを実装して、WorkSpaces アプリケーションユーザーを現在の場所に最も近い AppStream スタックに自動的にリダイレクトします。
- アプリケーション設定の永続性、ホームフォルダ、エラスティックフリートなど、WorkSpaces アプリケーションのオプション機能を使用している場合は、Windows ベースのフリートのユーザーデータの Amazon S3 クロスリージョンレプリケーションと Linux ベースのフリートのクロスリージョンレプリケーションを設定する必要があります。
- リージョン間でイメージをレプリケートします。詳細については、 AWS ドキュメントのAWS リージョン Amazon WorkSpaces アプリケーションの別の に所有しているイメージをコピーする」を参照してください。
- ドメインに参加しているフリートの場合、Active Directory フェデレーションサービス (AD FS) を含む Active Directory インフラストラクチャ (代替として SAML 2.0 と Amazon Cognito を使用している場合を除く) が他のリージョンで適切に設定されていること、およびマルチリージョンレプリケーション機能AWS Directory Service for Microsoft Active Directoryに使用することを確認してください。
- レイテンシーが最も低い WorkSpaces アプリケーションエンドポイントにユーザーを誘導します。詳細については、 AWS ブログ記事Amazon WorkSpaces アプリケーションのレイテンシーベースのルーティングによるユーザーエクスペリエンスの最適化」を参照してください。

サーバーレスアーキテクチャを使用する

サーバーレスアーキテクチャは、コンピューティング機能にクラウドマネージドサービスを使用することで、サーバー管理のオーバーヘッドを排除し、コストを削減します。

次のような AWS サーバーレスサービスを使用します。

- AWS Lambda イベント駆動型関数を使用してタスクを自動化し、カスタムロジックを統合するには
- WorkSpaces アプリケーションのユーザーデータ、アプリケーションファイル、セッションアーティファクトにスケーラブルなストレージを提供する Amazon S3

- WorkSpaces アプリケーションのパフォーマンスと使用状況メトリクスのモニタリング、ログ記録、アラートを提供する Amazon CloudWatch
- WorkSpaces アプリケーションアプリケーションのユーザー認証とアクセスコントロールを容易にする Amazon Cognito
- WorkSpaces アプリケーションと他の サービスまたはカスタムアプリケーション間のインターフェイスとなる RESTful API を作成する Amazon API Gateway APIs WorkSpaces

より頻繁に実験する

クラウドインフラストラクチャを使用すると、さまざまなリソース設定を迅速にテストして、パフォーマンスとコストを最適化できます。

- パフォーマンスとコストを最適化するために、さまざまなインスタンスタイプをテストします。
  - 異なるインスタンスファミリー間でストリームパフォーマンスを比較します。
  - グラフィックアプリケーションの GPU インスタンスと非 GPU インスタンスを評価します。
  - メモリを大量に消費するアプリケーションのメモリ最適化インスタンスをテストします。
- Image Builder を使用してアプリケーション設定をテストします。
  - さまざまなアプリケーション設定でテストイメージを作成します。
  - デプロイ前にアプリケーションのパフォーマンスを検証します。
  - さまざまなインスタンスタイプとのアプリケーションの互換性をテストします。
- 最小容量と最大容量、スケーリングポリシー、最大セッション期間などのセッション設定、切断タイムアウト設定などのフリート容量設定を使用して、フリート設定をテストします。

機械的な交感神経を検討する

ワークロード固有の要件と使用パターンに基づいてクラウドサービスを選択し、最適なパフォーマンスと効率を確保します。

- グラフィックを多用するアプリケーション、DirectX、OpenGL、OpenCL、または 3D 視覚化ソフトウェアを必要とするアプリケーションには、グラフィックス G5 インスタンスを選択します。
- ビジネスアプリケーション、ウェブブラウザ、ライトグラフィックスアプリケーションの`stream.standard`インスタンスを選択する
- などの CloudWatch メトリクスに基づいてストリーミングプロトコルをモニタリングおよび調整します`StreamingSessionLatency`。

- ユーザーに最も近い VPCs で WorkSpaces アプリケーションを設定し、アプリケーションの要件に基づいて適切なネットワーク帯域幅を使用します。
- アプリケーションの動作に基づいて、適切なフリートタイプを選択します。たとえば、専用リソースを必要とするアプリケーションにはシングルセッションフリートを選択し、リソースを効率的に共有できるアプリケーションにはマルチセッションフリートを選択します。
- マルチセッション環境とのアプリケーションの互換性を検討してください。
- ファイルシステムのリダイレクト機能を使用して、リモートアプリケーションとローカルアプリケーション間のインタラクションを処理します。詳細については、 AWS ブログ記事「Launching local applications from an Amazon WorkSpaces Applications streaming session」を参照してください。

コスト最適化の柱

AWS Well-Architected フレームワークのコスト最適化の柱は、支出を最小限に抑えながらビジネス価値を最大化することに重点を置いています。クラウドリソースに費やすすべてのドルが、組織の目標の達成に効果的に貢献するのに役立ちます。

この柱を WorkSpaces Applications ストリーミング環境に適用するための主な重点領域:

- フリート容量管理とインスタンスタイプの選択
- スケーリングとスケジューリングの最適化
- 使用パターンのモニタリングと分析
- コスト配分と追跡

クラウド財務管理はどのように実装しますか?

構造化されたプログラムとプロセスを通じて、クラウド財務管理とコスト最適化に専念する組織能力を構築し、クラウドの価値と効率を最大化します。

- AWS Cost Explorer および 使用状況レポートを使用して WorkSpaces アプリケーションのコストをモニタリングし、ストリーミング時間の使用状況を追跡し、フリートインスタンスのコストを分析し、リージョンのコスト配分をモニタリングします。
- AWS Budgets を使用して、WorkSpaces Applications のサービスコスト全体のアラートを設定し、サービスの予算しきい値を作成し、予算額に対する実際の支出を監視することで、コストコントロールを計画および設定します。詳細については、 AWS ブログ記事「オートメーションを使用して Amazon WorkSpaces アプリケーションのコストを最適化および制御する方法」を参照してください。

消費モデルを追加する

実際の使用パターンに基づいてコンピューティングリソースとコストをスケールします。たとえば、営業時間外に非本番環境をシャットダウンして、支出を最適化できます。

- 適切な料金モデルを選択します。例えば、常時オンフリートを一貫した使用に使用し、オンデマンドフリートを可変ワークロードに使用します。

- 最適なインスタンスタイプを選択します。たとえば、一般的なアプリケーションには `stream.standard` インスタンスを使用し、必要な場合にのみグラフィックスインスタンス (G4dn) を使用します。

全体的な効率を測定する

ビジネス出力のcost-per-unitを計算して追跡し、効率の向上を定量化し、最適化の取り組みをガイドします。

- セッション効率を追跡します。
- 次の CloudWatch メトリクスを使用してフリートの使用率をモニタリングします。
  - `AvailableCapacity` 未使用の容量を追跡する
  - `InUseCapacity` 実際の使用量を測定するには
- ストリーミング時間あたりのコスト、ユーザーあたりのコスト、アプリケーションあたりのコストなど、セッションあたりのコストを計算して追跡します。
- WorkSpaces アプリケーションのコストオプティマイザーを実装して、ビルダーをモニタリングします。
- フリートタイプ間でコストを比較します。たとえば、以下を比較します。
  - 単一セッションとマルチセッションのライセンスコスト
  - リソース使用率
  - インスタンスあたりのユーザー密度
- プロセス追跡データを使用して、使用率の低いアプリケーションや不要なアプリケーションを特定します。詳細については、 AWS ブログ記事 Track user processes in Amazon WorkSpaces Applications sessions を参照してください。

差別化されていない重い作業にお金を費やすのをやめる

AWS はインフラストラクチャの運用を管理し、マネージドサービスを提供するため、組織は IT メンテナンスではなくビジネス目標に集中できます。

- Image Builder を使用してアプリケーションイメージを作成および維持し、アプリケーションのパッケージ化、アプリケーション設定の構成、アプリケーションの互換性のテストを行います。
- 適切なインスタンスタイプを選択し、スケーリングしきい値を定義し、希望する容量制限を設定して、フリートの仕様を設定します。

- Windows ベースのフリートの場合は Amazon S3 のホームフォルダを設定し、Linux ベースのフリートの場合は Amazon EFS の共有ファイルシステムを設定して、永続的ストレージオプションを設定します。ストレージのアクセス許可を設定し、保持ポリシーを定義します。

支出の分析と属性付け

クラウドにより、ワークロードあたりのリソース使用量とコストを正確に追跡できるため、投資収益率 (ROI) の正確な測定とターゲット最適化の機会が可能になります。

- コスト配分用のフリート、アセット追跡用のイメージ、環境指定用の Image Builder、組織グループ化用のスタックの包括的なタグ付け戦略を実装します。
- AWS コストと使用状況レポート (AWS CUR) を使用して、タグ付けされたリソース別に WorkSpaces アプリケーションのコストを分類し、フリート、スタック、イメージあたりのコストを分析します。
- AWS Cost Explorer を使用して WorkSpaces アプリケーションの支出傾向を視覚化し、リージョンやインスタンスタイプなどのさまざまなディメンションのコストを比較します。
- フリート使用率、インスタンスタイプの効率、ストリーミング時間をアプリケーション別にモニタリングおよび分析します。
- 未使用のリザーブドキャパシティ、使用率の低いフリートまたはスタック、フリート使用状況のアイドル期間を追跡します。
- 各アプリケーションのユーザーあたりのコスト、アプリケーションあたりのストリーミング時間、ストリーミングアプリケーションのユーザー導入率を計算して追跡します。
- WorkSpaces アプリケーションの使用状況レポートを設定し、Amazon Athena を使用して使用状況データをクエリし、Amazon Quick でビジュアライゼーションを作成してコストと使用状況に関するインサイトを取得することで、詳細な使用状況分析を設定します。
- Windows Server ライセンス、アプリケーションライセンスモデル、ユーザーごとのライセンスなどの総コストの考慮事項を、デバイスごとのライセンスと比較して評価します。
- Amazon Athena を使用して、ホームフォルダのストレージコストとユーザー別の使用パターンをクエリおよび分析します。詳細については、 AWS ブログ記事「How to report Amazon WorkSpaces Applications home folder use with Amazon Athena」を参照してください。

持続可能性の柱

AWS Well-Architected フレームワークの持続可能性の柱は、環境フットプリントを最小限に抑え、エネルギー使用量と効率を最適化することを強調しています。アーキテクトがシステム設計とリソース配分戦略で環境意識に基づいた意思決定を行うようにガイドします。

この柱を WorkSpaces アプリケーションストリーミング環境に適用するための主な重点領域:

- 実際の需要に合わせてリソース割り当てを理解して最適化し、ストリーミング環境の無駄を最小限に抑える
- アプリケーション配信とストリーミングセッションの効率を向上させるためのユーザー消費パターンの分析と適応
- 適切なハードウェア設定を選択して使用し、パフォーマンス要件を満たしながらエネルギー効率を最大化する
- AWS マネージドサービス機能を使用して、これらのサービスが提供するスケールメリットと組み込み効率機能を活用する

影響を把握する

出力単位あたりのリソース効率と排出量を測定することで、ワークロードの環境への影響をモニタリングおよび最適化します。このデータを使用して KPIs。

- フリート使用率パターンをモニタリングします。
- ユーザーあたりのストリーミング時間を追跡します。
- フリート容量の使用状況の傾向を分析します。

持続可能性の目標を設定する

組織の目標に沿った各ワークロードについて、測定可能な持続可能性目標を設定します。スケーリングするトランザクションあたりのリソース強度を減らすことに焦点を当てます。

- フリート使用率、インスタンスタイプの効率、ストリーミング時間の最適化のターゲットを設定します。
- 実際の使用パターンに基づいて容量を計画します。

使用率を最大化する

リソースのサイズを適正化し、使用率を最大化することで、ワークロードの効率を最適化します。アイドル容量を減らしてエネルギー消費を最小限に抑え、持続可能性を向上させます。

- 実際の需要に合わせて自動スケーリングを設定します。
- 使用パターンに基づいて適切なサイズのフリート容量。
- 適切な最小容量制限と最大容量制限を実装します。
- ワークロードに適したインスタンスタイプを選択します。
- ストリーミングセッションの密度をモニタリングして最適化します。
- オフピーク時のアイドル容量を減らします。

より効率的な新しいハードウェアおよびソフトウェアサービスを予測して採用する

パートナーやサプライヤーからの新しい効率的なテクノロジーを常に把握し、迅速に導入して、ワークロードの環境への影響を継続的に改善します。

- 現行世代のインスタンスタイプを使用します。
- 利用可能な場合は、新しいインスタンスタイプにアップグレードします。
- アプリケーションストリーミング設定を最適化します。
- 適切なストリーミングプロトコルを設定します。
- 最新の WorkSpaces アプリケーション機能を更新します。

使用済みマネージドサービス

共有クラウドサービスとマネージドソリューションを活用して、リソース使用率を最大化し、自動スケーリングとライフサイクル管理による環境への影響を最小限に抑えます。

- Windows ベースのフリートのユーザーストレージには Amazon S3 を使用し、Linux ベースのフリートの共有ファイルシステムには Amazon EFS を使用します。
- モニタリング用の CloudWatch を実装します。
- アクセス管理用に IAM を設定します。

クラウドワークロードのダウンストリームへの影響を軽減する

クライアント側のリソース要件を最小限に抑え、エネルギー消費を削減し、ユーザーのデバイス寿命を延長するサービスを設計します。

- 不要なリソースの消費を防ぐため、最大セッション時間を調整します。
- 適切なセッションタイムアウトを設定します。
- 切断タイムアウトポリシーを設定します。
- 必要に応じてセッション永続性ポリシーを実装します。

リソース

AWS ドキュメント

- AWS Well-Architected フレームワーク
- Amazon WorkSpaces アプリケーション管理ガイド
- Amazon CloudWatch ユーザーガイド
- Amazon EFSS ユーザーガイド
- Amazon S3 ユーザーガイド
- IAM ユーザーガイド

AWS ブログ投稿

- Active Directory グループのメンバーシップベースの WorkSpaces アプリケーションのターゲット設定
- Azure AD ですべての Amazon WorkSpaces アプリケーションスタックの単一の ID プロバイダーを作成する
- Amazon WorkSpaces および Amazon WorkSpaces アプリケーションの Windows リモートアシスタンスの設定
- AS2TrustedDomains DNS TXT レコードを作成して Amazon WorkSpaces アプリケーションのネイティブクライアントをサードパーティーの ID プロバイダーにリダイレクトする
- Amazon WorkSpaces アプリケーションでのカスタムログ記録と Amazon CloudWatch アラートの作成
- Geo Targetly および Amazon WorkSpaces アプリケーションによるクロスリージョンリダイレクト
- クロスアカウントリソースと Amazon WorkSpaces アプリケーション
- バイオキー PortalGuard および Amazon WorkSpaces アプリケーションでフェデレーションを有効にする
- SimpleSAMLphp および Amazon WorkSpaces アプリケーションでフェデレーションを有効にする
- Duo Single Sign-On および Amazon WorkSpaces アプリケーションで ID フェデレーションを有効にする
- Shibboleth および Amazon WorkSpaces アプリケーションで ID フェデレーションを有効にする

- Amazon エンドユーザーコンピューティングを使用したオンプレミス VDI のフェイルオーバー戦略
- Amazon が Amazon WorkSpaces アプリケーションを使用してデータサイエンティストとアナリストに機密データへのアクセスを提供する方法
- Amazon WorkSpaces アプリケーションの証明書ベースの認証を設定する方法
- Amazon WorkSpaces アプリケーションのアプリケーションエンタイトルメントで Okta クレームを使用する方法
- オープンソースの仮想アプリケーション管理を使用した Amazon WorkSpaces アプリケーションのコンピュータラボの管理
- WorkSpaces アプリケーションのコストをビジネスユニットに割り当てる方法
- Amazon OpenSearch Service と Amazon Kinesis Data Firehose を使用した Amazon WorkSpaces アプリケーションのモニタリング
- Amazon WorkSpaces、Amazon WorkSpaces アプリケーション、Amazon WorkSpaces Amazon Macie を使用したネットワーク分離とデータサニタイズ
- Amazon WorkSpaces アプリケーションを使用した OneLogin SSO
- Amazon WorkSpaces アプリケーションで Connect Customer Call オーディオパスを最適化する
- Amazon Elastic File System 上の Amazon WorkSpaces アプリケーション Linux フリートの永続的ストレージ
- Okta SAML アプリを Amazon WorkSpaces Applications ネイティブクライアントにリダイレクトする
- Application Masking を使用して Amazon WorkSpaces アプリケーションのイメージ管理を簡素化する
- Amazon WorkSpaces Applications Elastic フリートと Linux との互換性により、アプリケーションを低コストでストリーミングする
- WorkSpaces アプリケーションを使用した規制対象環境のインターフェイス VPC エンドポイントからのストリーミング
- Azure AD で Amazon WorkSpaces Applications アプリケーションの使用権限を使用する
- Amazon WorkSpaces アプリケーションのユーザー問題レポーター
- Google Workspace での Amazon WorkSpaces アプリケーションアプリケーションの使用権限の使用
- Amazon WorkSpaces アプリケーションでの Microsoft Active Directory での Auth0 の使用
- Microsoft AppLocker を使用した Amazon WorkSpaces アプリケーションのアプリケーションエクスペリエンスの管理

- Python を使用して WorkSpaces アプリケーション Linux イメージングアシスタント GUI を強化する
- WorkSpaces アプリケーションクライアントのウェブアプリケーションリダイレクトオプション

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	—	2025 年 7 月 23 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。