



OpenText TeamSite およびメディア管理ワークロードを に移行する AWS クラウド

AWS 規範ガイドンス



AWS 規範ガイド: OpenText TeamSite およびメディア管理ワークロードを に移行する AWS クラウド

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

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

Table of Contents

序章	1
高レベル移行の概要	3
コストとライセンス	6
前提条件	7
ソースアーキテクチャとターゲットアーキテクチャ	8
ソースアーキテクチャ	8
ターゲットアーキテクチャ	12
高レベル移行プロセスフロー	16
オートメーションとツール	19
リソース	20
.....	20
ドキュメント履歴	21
.....	xxii

OpenText TeamSite およびメディア管理ワークロードを に移行する AWS クラウド

Michael Stewart、Carlos Marruenda Molina、Battulga Purevragchaa、Amazon Web Services (AWS)

2021 年 6 月 ([ドキュメント履歴](#))

[OpenText TeamSite](#) エンタープライズ・ウェブ・コンテンツ管理システム (CMS) を通じて、パーソナライズされたオムニチャネルデジタルエクスペリエンスを提供するのに役立ちます。OpenText TeamSite は、[OpenText メディア管理](#) および [OpenText MediaBin](#) など OpenText デジタルアセット管理ソリューションとも統合しています。これらのソリューションは、デジタルアセット管理と配信機能を提供し、OpenText TeamSite を補完するために使用できます。

多数 [OpenText カスタマーエクスペリエンス](#) ワークロードは、固定容量またはレガシーホスティングコストモデルを備えたオンプレミスまたは従来のホスティングソリューションでホストされます。OpenText カスタマーエクスペリエンスプラットフォームをAmazon Web Services に移行する (AWS) クラウドは、ビジネスの俊敏性と統合機能を向上させ、総保有コスト (TCO) を削減することで、さらなる機能と価値を提供します。OpenText TeamSite およびメディア管理のワークロードを AWS クラウドに移行することで、次の 3 つのビジネス成果を達成できます。

- ビジネスの俊敏性の向上 - 改善された開発および展開ツールを使用し、AWS クラウドと統合することにより、新しいキャンペーン、製品、ソリューション、および機能の市場投入までのコストと時間を削減します。
- 大幅なコスト削減 – 規模の経済からメリットを得、要件に合わせて迅速にスケール AWS クラウドするインフラストラクチャを で使用します。通常、 で OpenText ワークロードをホスト AWS クラウド [すると、コストを最大 70% 削減できます](#)。
- イノベーションとデジタルトランスフォーメーションの有効化 — AWS クラウドネイティブサービスを使用するか、外部クラウドアプリケーションと統合して、ユーザーエクスペリエンスの向上、ビジネスインサイトの取得、またはコストとリソースの最適化 (たとえば、データと顧客分析の改善、カスタムメイドのコンテンツ、人工知能 (AI) 主導の顧客インタラクション) により、革新的なユースケースを実現します。

OpenText TeamSite プラットフォームを に移行 AWS クラウド し、AWS 製品やサービスでモダナイズすることで、これらのビジネス成果を達成できます。

OpenText TeamSite ワークロードのモダナイズーションは、移行の実装の特性と特定のビジネス要件によって異なります。OpenText カスタマーエクスペリエンスプラットフォームを最新化

するときは、[クラウドのAPI](#) アプローチを使用することをお勧めします。このアプローチにより、[Amazon API Gateway](#) は、[Amazon Elastic Container Service \(Amazon ECS\)](#) または [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#) 上のコンテナで展開されたさまざまなアプリケーション、サーバーレスアプリケーション (例えば、[AWS Lambda](#))、および市販の商用 (COTS) アプリケーションをオーケストレーションできます。このアプローチは、サービスおよびアプリケーションの統合カタログを公開インターネットにも公開します。

このガイドでは、OpenText TeamSite ワークロード AWS クラウド を に移行してコストを削減し、モダナイゼーションを実現することに重点を置いています。移行ステップの完全な概要については、「規範ガイダンス」ウェブサイトの[OpenText TeamSite ワークロードを AWS クラウドに移行する](#)」パターンを参照してください。[AWS](#) このガイドは、AWS および AWS パートナーである [TBSCG](#) によって作成され、OpenText TeamSite ワークロードを に移行する技術マネージャー、技術エグゼクティブ、技術エンジニアリングチーム、アーキテクチャチームを対象としています AWS クラウド。

高レベル移行の概要

次の表は、OpenText TeamSite と Media Management の AWS クラウドへの移行の概要を示しています。

ワークロード	ソースワークロード	OpenText カスタマーエクスペリエンスソリューション:
		• OpenText TeamSite • OpenText LiveSite • OpenText メディア管理 • OpenText MediaBin
	ソース環境	オンプレミスまたは別のクラウドプロバイダー。
	移行戦略 (7 Rs)	ソース環境に応じて、以下のような移行戦略があります。 • リファクタリング • リホスト • リプラットフォーム
	これはワークロードバージョンのアップグレードですか。	いいえ
移行	ソースワークロードは、独立系ソフトウェアベンダー (ISV) ワークロードとは異なっていますか。	いいえ
	推定移行時間	ソース環境によっては、移行に 2 ~ 3 週間かかることがあります。
	推定コスト	で ISV ワークロードを実行する推定コスト AWS クラウド Elastic Load Balancing を使用した 3 つの環境 (開発、QA-

		UAT、本番稼働) のコスト構造と、開発およびデプロイ用の DevOps ツールセットのサンプルは、 この見積り で入手できます AWS 料金見積りツール。
前提条件と必要事項	システムの最小および最大要件	お使いの OpenText 製品バージョンの互換性マトリックスを確認してください。このマトリックスは、「 OpenText サポートポータル 」のリリースノートに掲載されています。
	サービスレベルアグリーメント (SLA)	Elastic Load Balancing、AWS Lambda 関数、および 2 つのアベイラビリティゾーンを使用する基本的なセットアップで SLAs を満たすことができます。
	AWS アカウントでのモデルのライセンスと運用	OpenText ライセンス契約の詳細については、OpenText アカウントマネージャーにお問い合わせください。

AWS で使用される のサービス AWS アカウント

- [Amazon API Gateway](#)
- [Amazon Elastic Block Store \(Amazon EBS\)](#)
- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#)
- [Amazon Elastic Container Service \(Amazon ECS\)](#)
- [Amazon OpenSearch Service](#)
- [Elastic Load Balancing](#)
- [Lambda](#)
- [Amazon Relational Database Service](#) (Amazon RDS)
- [Amazon Simple Storage Service \(Amazon S3\)](#)

コストとライセンス

OpenText TeamSite、LiveSite、および Media Management ワークロードを に移行するために必要なコストとライセンスを評価するときは AWS クラウド、移行する開発環境、QA 環境、および本番環境があることを前提としています。DevOps ツールセットには、次の AWS サービスも含まれている必要があります。

- [Amazon CloudWatch](#) そして [Amazon CloudWatch Logs](#) のモニタリングとログの記録
- アーカイブ用の [Amazon Glacier](#)
- [AWS CodeDeploy](#) ソフトウェア展開の自動化のために
- [Amazon Simple Notification Service \(Amazon SNS\)](#) アプリケーション間およびアプリケーション間およびアプリケーション対人コミュニケーション用
- [AWS Config](#) の構成を記録して評価するための AWS リソース

残りの環境では、次の AWS 製品とサービスを使用する必要があります。

- OpenText Media Management、TeamSite、Livesite 用の Amazon Elastic Compute Cloud (Amazon EC2) インスタンス
- OpenText TeamSite およびメディア管理データベース用の Amazon Relational Database Service (Amazon RDS) インスタンス
- [Application Load Balancer](#) 複数の OpenText LiveSite サーバー間でリクエストを配布するには

コストの詳細については、[この見積り](#)を参照してください AWS 料金見積りツール。

前提条件

このガイドのソースアーキテクチャは、仮想マシン (VMs) でホストされていません。また、このガイドでは、移行中に OpenText のバージョンを更新しないことを想定しています。OpenText TeamSite のバージョンを更新する場合、または Open Text MediaBin からメディア管理にアップグレードする場合は、移行を実行し、必要なアップグレードまたは更新を実装する必要があります。

最後に、このガイドでは OpenText アーキテクチャに重要なリファクタリングがないことを前提としています。リファクタリングは、コンテナ化または AWS Lambda 関数に移行される可能性のあるサポートアプリケーションに制限する必要があります。既存のアプリケーションランドスケープを移行してから、AWS クラウド上のワークロードを変更することをお勧めします。

次の表に、移行プロジェクトで利用できる必要があるリソースについての情報を示します。

リソース	使用方法
必要なすべての出典インフラストラクチャリソースへのアクセス	配信を高速化し、アクセス要求の許可待ちを回避することが可能
インフラストラクチャの専門家 (テクニカルアーキテクトやシステム管理者など)	現在のインフラストラクチャの発見
現在の OpenText または関連アプリケーションのソリューションエキスパート	アプリケーションランドスケープの検出と依存関係の識別
コンテンツ投稿者やサポートされている Web アプリケーションの最終ユーザーを含む、最終ユーザーのグループ	ユーザー受け入れテスト (UAT) と新しいプラットフォームの検証

ソースアーキテクチャとターゲットアーキテクチャ

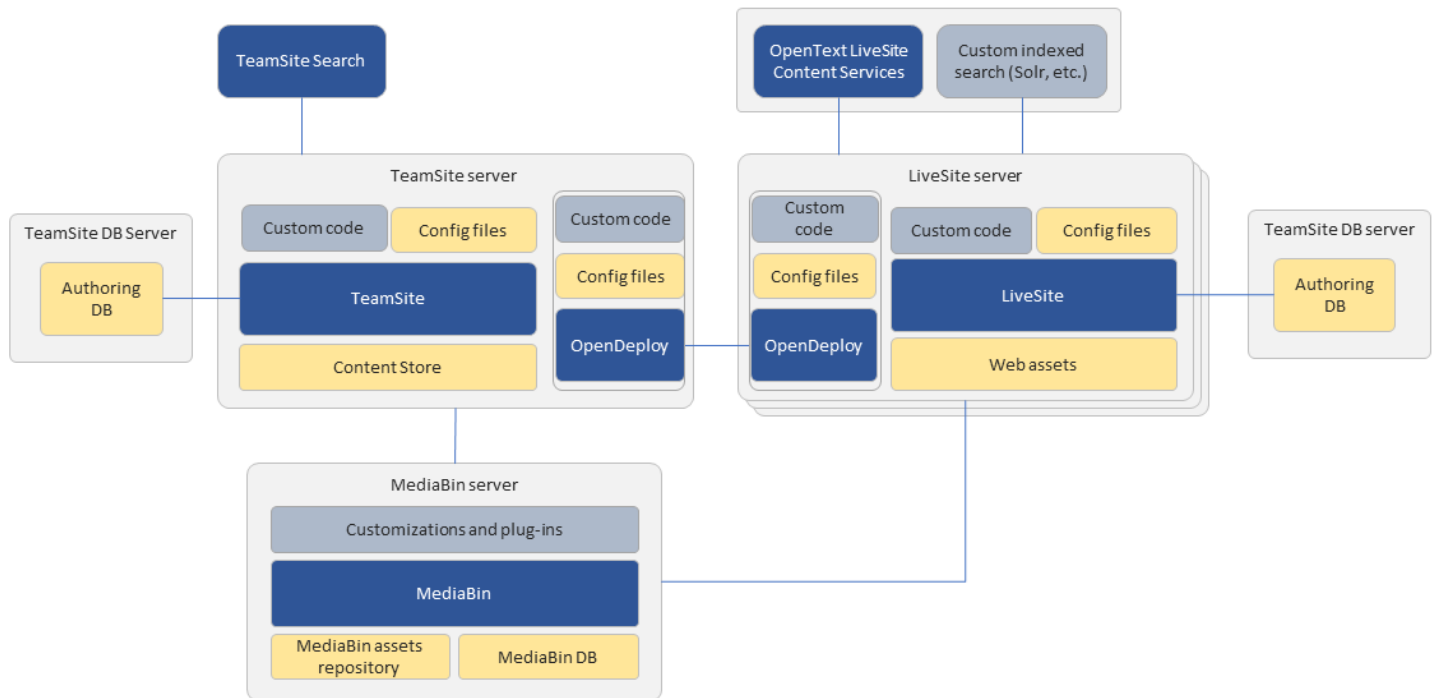
以下のセクションでは、OpenText の移行を AWS クラウドにするためのソースアーキテクチャとターゲットアーキテクチャを説明しています。これには、一般的な OpenText カスタマーエクスペリエンスプラットフォーム実装のコンポーネント、ターゲット AWS アーキテクチャ、高レベルの移行プロセス、各コンポーネントタイプの最も一般的な移行戦略 (7 R) が含まれます。

トピック

- [ソースアーキテクチャ](#)
- [ターゲットアーキテクチャ](#)

ソースアーキテクチャ

次の図は、OpenText カスタマーエクスペリエンスアプリケーションの典型的なアーキテクチャを示しています。このアーキテクチャでは、OpenText コアコンポーネント、OpenText コアコンポーネントに接続されたカスタム機能、およびデータベース、ファイル、リポジトリが使用されています。OpenText アーキテクチャはお客様の実装によって異なりますが、この図では典型的なコンポーネントを示しており、これらのコンポーネントについては、このガイドで説明します。



次の表に、このガイドで移行の対象となる主なアーキテクチャ要素について説明します。

解決策	移行で考慮すべき主要要素
OpenText TeamSite	TeamSite インスタンス コンテンツストア 設定ファイル (例 : tsgroups.xml または roles.xml) カスタムコード — 外部データソースまたはカスタムメイドの機能との統合などのコードのカスタマイズ オーサリングデータベース — このデータベースは通常、専用のデータベースサーバーにデプロイされます。 TeamSite 検索 — 独自のサーバーにデプロイ (オプション) OpenDeploy: • OpenDeploy インスタンス • 設定 (OpenDeploy ユーザーなど) • カスタムコード — カスタムメイドの機能 (マルチ環境デプロイなど) のコードのカスタマイズ
OpenText LiveSite	LiveSite インスタンス Web アセットリポジトリ 設定ファイル カスタムコード 独自のサーバーにデプロイされたランタイムデータベース OpenDeploy:

インデックス検索	• OpenDeploy インスタンス • 設定 • カスタムコード これは OpenText LiveSite コンテンツサーバー、または Apache Solr のような類似のインデックス検索の実装です。
OpenText メディア管理または MediaBin	MediaBin インスタンス カスタマイズまたは既存のプラグインのカスタムコード MediaBin アセットリポジトリ MediaBin データベース

選択できる移行戦略と AWS 製品とサービスは、ソースシステムの特性と個々の要件によって異なります。次の表に、移行に関する最も一般的な戦略を示します。

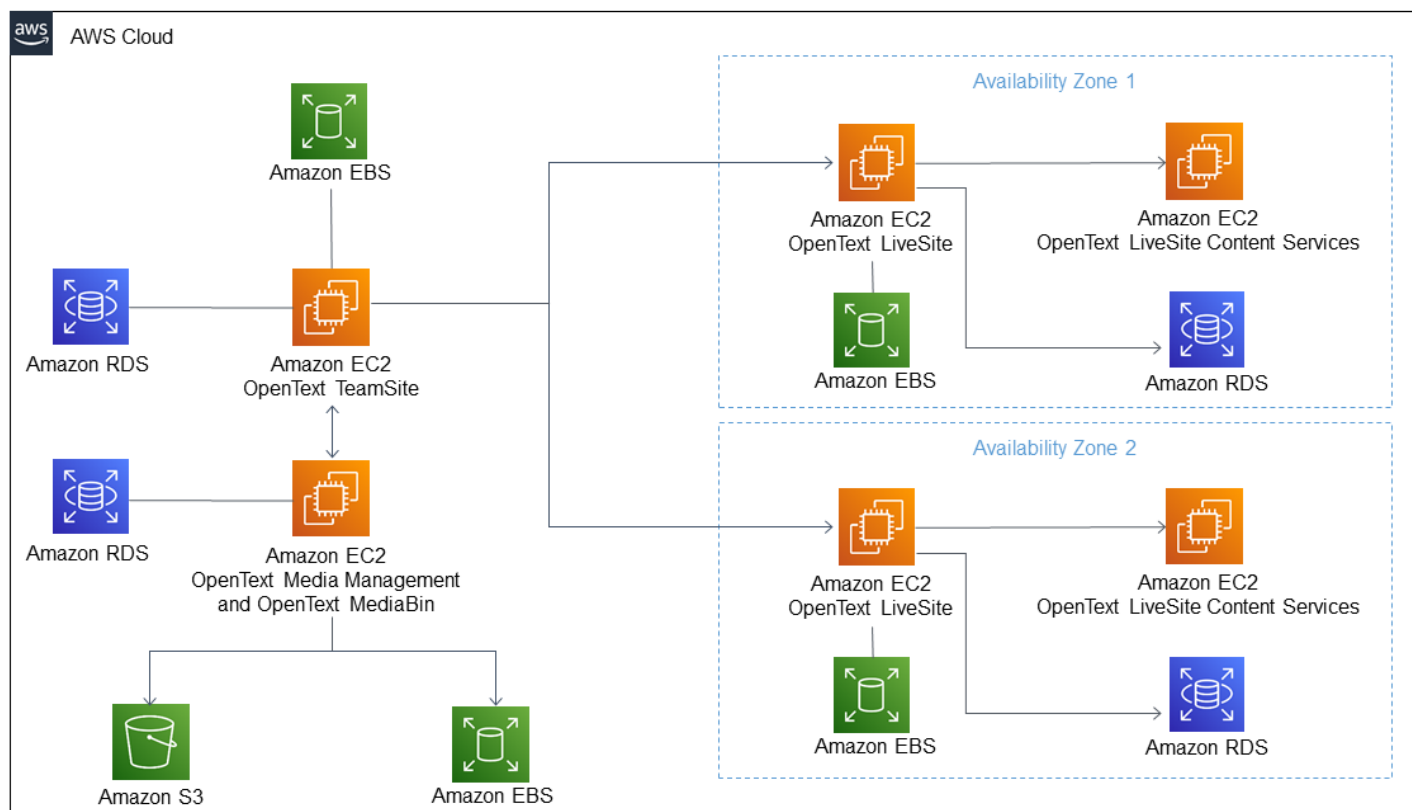
要素タイプ	ターゲット AWS サービス	移行戦略
OpenText コアコンポーネント	• Amazon Elastic Compute Cloud (Amazon EC2) インスタンス • Amazon Elastic Container Service (Amazon ECS) および Amazon Elastic Kubernetes Service (Amazon EKS) を持つコンテナ	• リホスト • リプラットフォーム 通常、製品の新しいインスタンスをインストールします。各インスタンスタイプのインストールは完全に自動化されています。
カスタム機能と統合	• Amazon EC2、OpenText コアコンポーネントと統合 • コンテナ (Amazon ECS や Amazon EKS など)	• リホスト • リプラットフォーム • リファクタリング • 保持する

	• サーバーレスマイクロサービス (例: AWS Lambda) • Amazon API Gateway	OpenText プラットフォームのメンテナンスと進化に使用されるデプロイパイプラインをプロビジョニングして設定します。これらのパイプラインは、コードをデプロイするために使用されます。 OpenText TeamSite のカスタマイズとして、または外部アプリケーションとして構築されたいくつかの依存機能は、Lambda 関数としてコンテナ化またはリファクタリングできます。この場合、API Gateway を使用してサーバーレス機能をオーケストレーションできます。
データベース	• Amazon Relational Database Service (Amazon RDS)	• リプラットフォーム 通常は、AWS Database Migration Service (AWS DMS) を使用して Amazon RDS DB インスタンスにデータベースを移行できます。

Storage	• Amazon Elastic Block Store (Amazon EBS) • Amazon Simple Storage Service (Amazon S3)	• リホスト データリポジトリは、OpenText コアコンポーネントのインスタンスに関連付けられた Amazon EBS ボリュームにコピーされます。 S3 バケットは、OpenText MediaBin やメディア管理アセットリポジトリなどの大規模なデータリポジトリに使用できます。
---------	---	---

ターゲットアーキテクチャ

ターゲットアーキテクチャ、移行戦略、および技術的な手順は、ソースアーキテクチャ、接続性の制約、セキュリティ要件、およびその他のポリシーによって異なります。また、組織の個々のポリシーや要件にも依存します。次の図は、OpenText プラットフォームをホスト AWS クラウド できる の サンプルアーキテクチャを示しています。



次の表は、ソース環境からターゲット AWS アーキテクチャへの移行に必要な大まかな手順の概要を示しています。この移行の詳細な手順は、AWS「規範ガイド」ウェブサイトの[OpenText TeamSite ワークロードを AWS クラウドに移行する](#)パターンで確認できます。

高レベルのステップ	説明
1 AWS インフラストラクチャのプロビジョニング	を使用して CloudFormation、新しい AWS インフラストラクチャを自動的にプロビジョニングします。
2 DevOps リポジトリ、ツール、およびプロシージャをセットアップする	AWS 製品やサービス (例: AWS CodePipeline) を使用して、継続的インテグレーションと継続的デプロイ (CI/CD) パイプラインを構築し、リリースプロセスを自動化します。
3 新しい OpenText インスタンスをインストールする	新しい OpenText ソフトウェアをインストールして、段階的な移行を実行し、トラブルシューティング問題に役立てることをお勧めします。

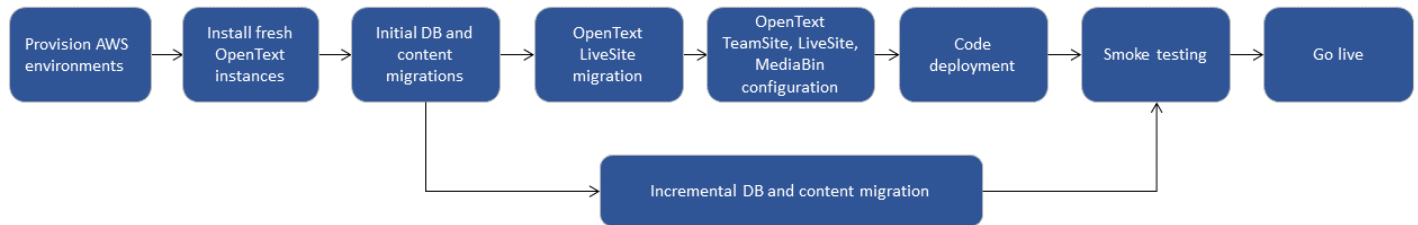
4 データベースとコンテンツの移行	これは繰り返し行われるプロセスであり、本番データベースとコンテンツリポジトリから初期コピーが作成されます。次に、本番稼働用環境以外で移行プロセスを実行してテストします。移行中にデータが失われないように、増分アップデートまたはライブアップデートを稼働開始日まで実行する必要があります。
5 OpenText LiveSite を移行する	LiveSite ウェブサーバーのコンテンツを移行します。
6 OpenText TeamSite、LiveSite、メディア管理、MediaBinを設定する	設定ファイルをコピーし、新しい環境に設定を適応させます。
7 コードのデプロイ	OpenText ソリューションのカスタマイズと追加機能の両方について、新しい環境にコードをデプロイします。
8 スモークテスト	新しい OpenText プラットフォームが稼働可能であることを検証するための予備テスト。
9 稼働	この段階で、プロジェクトの目標が達成され、最終ユーザーは新しい OpenText プラットフォームを使用しています。プラットフォームは、さらなる実現とメンテナンスの準備が整いました。

Important

これらのステップは、完全な移行プロセスを表しているわけではありません。これらは、移行実装の技術的ステップのみを表しています。

これらの高度な移行手順の実装を開始する前に、移行インベントリの定義、既存のアーキテクチャおよび移行要件の分析、計画アーキテクチャの定義、自動化スクリプトの作成、および必要に応じてコードのリファクタリングを行う必要があります。

高度なステップの実装は、明確な順序では実行されません。同時に実行されるテスト、変更、再実装、およびプロセスがあります。これらは、各移行の特性に左右され、次の図は移行順序の例を示しています。



高レベル移行プロセスフロー

移行プロセスは、検出、設計、開発、テストと改良、実装の 5 つのフェーズで行われます。次の表は、これらの 5 つのフェーズの概要と移行プロセスのフローを示しています。

	Discovery	設計	開発	テストと改良	実装
結果	- 既存のシステムアーキテクチャ - 移行インベントリ - 機能的要件と非機能的要件 - 将来のシステムアーキテクチャ (高レベル)	- 各コンポーネントの移行戦略 (7 Rs) - ターゲット AWS アーキテクチャ - TCO の分析と評価 - 高レベル移行ステップ	- すぐにデプロイできるアプリケーション - の自動化スクリプト CloudFormation - 自動化タスクと非自動化タスクを含む、段階的な移行計画 - 詳細なテストプラン	- テストレポート - 自動化スクリプト、移行計画、テストプランの洗練化	- で実行されている OpenText カスタマーエクスペリエンスプラットフォーム AWS クラウド - テストレポート
タスク	- 検出ワークショップ - ドキュメント分析	- 複数分野の SME (Subject Matter Experts) ミーティング - コスト分析 - 検証ワークショップ	- 複数分野にわたる SME ミーティング - アプリケーションのリファクタリングと適応 - 自動化スクリプトの適応と作成	- 自動化テスト - 自動化と移行プランの適応	- 移行プランの実装 - 自動化テスト

			新しいサービスまたは機能を作成またはプロビジョニングする		
カスタマーリソース	- 技術文書およびビジネス文書 - ビジネスオーナーとテクニカルアーキテクト — ワークショップやインタビューへの参加	- 現在の TCO データ - ビジネスオーナーとテクニカルアーキテクト — ワークショップやインタビューへの参加 - 技術資産 (環境、ソースコード、またはデプロイパイプライン) へのアクセス	- 技術資産 (環境、ソースコード、またはデプロイパイプライン) へのアクセス - 技術チーム — 現在のアプリケーションに関するサポート。	- ユーザーテスト - インフラストラクチャチーム — 移行アクティビティのサポート	- ユーザーテスト - インフラストラクチャチーム — 移行アクティビティのサポート

アクセラレー ター	• OpenText アーキテク チャ解析フ レームワー ク • 図表 • フォーカス ポイント	• 最も一般的 な移行戦 略や、各ソ リユーシ ョンコン ポーネン トのター ゲットとな る AWS 製 品、サービ スに関する 既存の知識	既存のオート メーションア セット: • AWS リ ソースのプ ロビジョニ ング • データ移行 • アプリケー ションのイ ンストール と設定の変 更	過去の移行で 作成した自動 テスト資産の 例: • システムヘル スチェッ ク • パフォーマ ンスおよび 負荷テスト スクリプト	過去の移行で 作成した自動 テスト資産の 例: • システムヘル スチェッ ク • パフォーマ ンスおよび 負荷テスト スクリプト
--------------	--	--	---	--	--

オートメーションとツール

オートメーションスクリプトを構築し、非本番環境で実行し、テストと調整を行い、環境のセットアップ、コンテンツの移行、およびコードのデプロイがスムーズに実行され、完全に機能する環境を作成するまで移行を自動化することをお勧めします。

自動化スクリプトとツールはカスタムメイドで、以前の移行の既存のアセットに基づいています。一般的な移行用に開発された主な資産には、次のようなものがあります。

- CloudFormation AWS サービスを完全にプロビジョニングするための スクリプト
- AWS Database Migration Service 必要に応じてデータベース移行スクリプトを含む (AWS DMS) 設定
- お客様の要件に応じた CI/CD ツールの構成 (例: [AWS CodeDeploy](#)、Jenkins、または Bamboo)。
- 追加タスクのためのカスタムスクリプト(例: .sh)

移行プロセス全体を実行して洗練させる場合は、スクリプト、手動アクション、テスト、およびフォールバックオプションを含む詳細なステップバイステップガイドも作成する必要があります。

リソース

- [OpenText TeamSite ワークロードを AWS クラウドに移行する](#)
- [OpenText AWS 移行プレイブック](#)
- [クラウドベースのエクスペリエンスプラットフォーム](#)

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
二	初版発行	2021 年 6 月 24 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。