



Migrating OpenText TeamSite and Media Management workloads to the AWS Cloud

AWS Prescriptive Guidance



AWS Prescriptive Guidance: Migrating OpenText TeamSite and Media Management workloads to the AWS Cloud

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Introduction 1

Overview 1

High-level migration overview 3

Cost and licenses 6

Assumptions and prerequisites 7

Source and target architecture 8

Source architecture 8

Target architecture 12

High-level migration process flow 15

Automation and tooling 18

Resources 19

Document history 20

Migrating OpenText TeamSite and Media Management workloads to the AWS Cloud

Battulga Purevragchaa, Amazon Web Services

Overview

[OpenText TeamSite](#) helps deliver personalized, omnichannel digital experiences through an enterprise web content management system (CMS). OpenText TeamSite also integrates with OpenText digital asset management solutions, such as [OpenText Media Management](#) and [OpenText MediaBin](#). These solutions offer digital asset management and distribution capabilities, and you can use them to complement OpenText TeamSite.

Many [OpenText Customer Experience](#) workloads are hosted on premises or on traditional hosting solutions with fixed capacity or legacy hosting cost models. Migrating your OpenText Customer Experience platform to the Amazon Web Services (AWS) Cloud provides additional capabilities and value by increasing your business agility and integration capabilities, and reducing the total cost of ownership (TCO). By migrating OpenText TeamSite and Media Management workloads to the AWS Cloud, you can achieve the following three business outcomes:

- **Improved business agility** – Reduce the cost and time to market for your new campaigns, products, solutions, and functionalities by using improved development and deployment tools, and integrating with the AWS Cloud.
- **Significant cost reduction** – Benefit from economies of scale and use an infrastructure in the AWS Cloud that quickly scales to your requirements. Typically, hosting OpenText workloads on the AWS Cloud [can reduce your costs by up to 70 percent](#).
- **Enable innovation and digital transformation** – Use AWS cloud-native services or integrate with external cloud applications to enable innovative use cases for improving user experience, gaining business insights, or optimizing costs and resources (for example, improved data and customer analytics, custom-made content, or artificial intelligence (AI)-driven customer interactions).

You can achieve these business outcomes by migrating your OpenText TeamSite platform to the AWS Cloud and modernizing it with AWS products and services.

Modernizing an OpenText TeamSite workload depends on the characteristics of your migration's implementation and your specific business requirements. We recommend that you use the [API in the cloud](#) approach when modernizing an OpenText Customer Experience platform. This approach enables [Amazon API Gateway](#) to orchestrate different applications deployed in containers on [Amazon Elastic Container Service \(Amazon ECS\)](#) or [Amazon Elastic Kubernetes Service \(Amazon EKS\)](#), serverless applications (for example, [AWS Lambda](#)), and commercial-off-the-shelf (COTS) applications. The approach also exposes a consolidated catalog of services and applications to the public internet.

This guide focuses on migrating OpenText TeamSite workloads to the AWS Cloud to reduce costs and enable modernization. For a full overview of the migration steps, see the pattern [Migrate OpenText TeamSite workloads to the AWS Cloud](#) on the [AWS Prescriptive Guidance](#) website. This guide was created by AWS and [TBSCG](#), an AWS Partner, and is intended for technical managers, technical executives, and technical engineering and architectural teams that are migrating OpenText TeamSite workloads to the AWS Cloud.

High-level migration overview

The following table shows a high-level overview of an OpenText TeamSite and Media Management migration to the AWS Cloud.

Workload	Source workload	OpenText Customer Experience solutions: • OpenText TeamSite • OpenText LiveSite • OpenText Media Management • OpenText MediaBin
	Source environment	On premises or on another cloud provider.
Migration	Migration strategy (7 Rs)	Depending on your source environments, you can use the following migration strategies: • Refactor • Rehost • Replatform
	Is this an upgrade in workload version?	No
	Is the source workload different from an independent software vendor (ISV) workload?	No
	Estimated migration duration	Depending on your source environment, a migration can

take between two and three weeks.

Estimated cost

Estimated cost of running ISV workloads on the AWS Cloud

A sample cost structure for three environments (development, QA-UAT, and production) using

[Elastic Load Balancing](#)

and a DevOps toolset for development and deployment is available in [this estimate](#) from AWS Pricing Calculator.

Assumptions and prerequisites

Minimum and maximum system requirements

Check the compatibility matrix for your OpenText product version. This matrix is available in the release notes on the [OpenText support portal](#).

Service-level agreements (SLAs)

You can meet SLAs with a basic setup that uses Elastic Load Balancing, AWS Lambda functions, and two Availability Zones.

Licensing and operating model in the AWS account

For more information about your OpenText license agreement, contact your

OpenText account manager.

AWS services used in the AWS account

- [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 \(Amazon S3\)](#)

Cost and licenses

When assessing the costs and licenses required to migrate OpenText TeamSite, LiveSite, and Media Management workloads to the AWS Cloud, we assume that you have development, QA, and production environments to migrate. Your DevOps toolset should also include the following AWS services:

- [Amazon CloudWatch](#) and [Amazon CloudWatch Logs](#) for monitoring and logging
- [Amazon Simple Storage Service Glacier](#) for archiving
- [AWS CodeDeploy](#) for software deployment automation
- [Amazon Simple Notification Service \(Amazon SNS\)](#) for application-to-application and application-to-person communication
- [AWS Config](#) for recording and evaluating the configurations of your AWS resources

Your remaining environments should use the following AWS products and services:

- Amazon Elastic Compute Cloud (Amazon EC2) instances for OpenText Media Management, TeamSite, and Livesite
- Amazon Relational Database Service (Amazon RDS) instances for OpenText TeamSite and Media Management databases
- [Application Load Balancers](#) to distribute requests across multiple OpenText LiveSite servers

For more information about costs, see [this estimate](#) from AWS Pricing Calculator.

Assumptions and prerequisites

This guide's source architecture is not hosted on virtual machines (VMs). The guide also assumes that you won't update OpenText versions during your migration. If you want to update an OpenText TeamSite version or upgrade from Open Text MediaBin to Media Management, you should carry out your migration and then implement any required upgrades or updates.

Finally, the guide assumes that there is no significant refactoring of your OpenText architecture.

Refactoring should be limited to supporting applications that might be containerized or migrated to AWS Lambda functions. We recommend that you migrate the existing application landscape and then makes changes to your workloads on the AWS Cloud.

The following table provides information about the resources that must be available for your migration project.

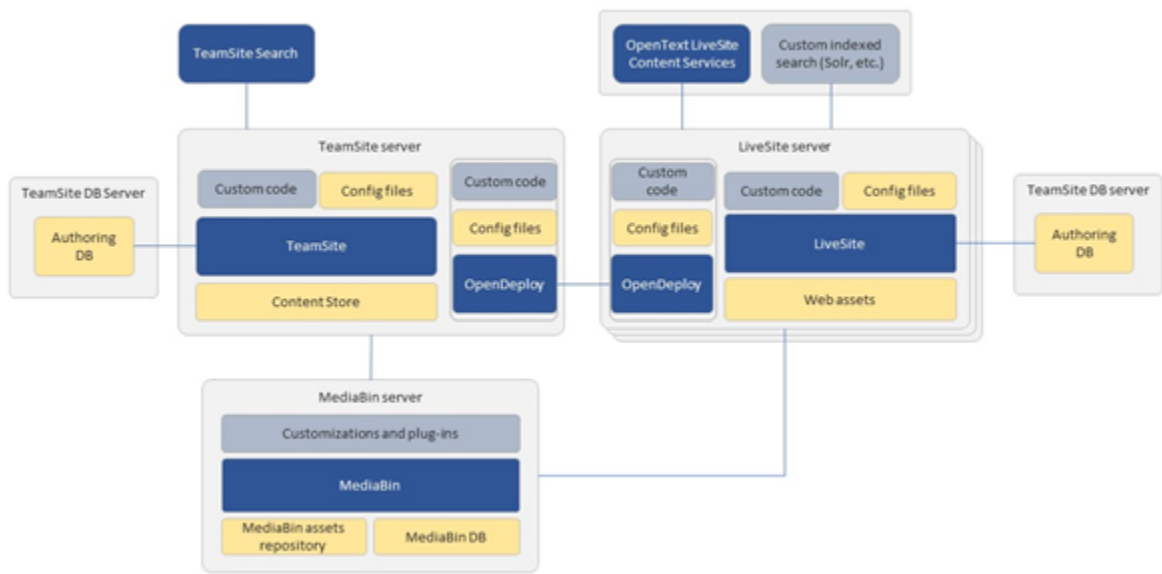
Resources	Usage
Access to all required source infrastructure resources	Accelerate the delivery and avoid waiting for access requests to be granted
Infrastructure experts (for example, technical architects and system administrators)	Discovery of current infrastructure
Solution experts for your current OpenText or related applications	Discovery of application landscape and identify dependencies
A group of final users, including content contributors and final users of supported web applications	User acceptance testing (UAT) and validation of the new platform

Source and target architecture

The following sections describe the source and target architecture of an OpenText migration to the AWS Cloud. This includes the components of a typical OpenText Customer Experience platform implementation, the target AWS architecture, a high-level migration process, and the most common migration strategy (7 Rs) for each component type.

Source architecture

The following diagram shows a typical OpenText Customer Experience application architecture that uses OpenText core components, custom functionalities connected to the OpenText core components, and databases, files, and repositories. Although an OpenText architecture varies for each customer implementation, the diagram shows the typical components and these are covered by this guide.



The main architectural elements that are targeted for migration by this guide are described in the following table.

Solution	Main elements to consider for migration
OpenText TeamSite	TeamSite instance
	Content store

Configuration files (for example, tsgroups.xml or roles.xml)

Custom code – Code customizations such as integrations with external data sources or custommade functionalities

Authoring database – This database is typically deployed in a dedicated database server

TeamSite search – Deployed in its own server (optional)

OpenDeploy:

- OpenDeploy instance
- Configuration, for example OpenDeploy users
- Custom code – Code customizations for custommade functionalities (for example, multienvironment deployments)

OpenText LiveSite

LiveSite instance

Web assets repository

Configuration files

Custom code

A runtime database deployed on its own server

OpenDeploy:

- OpenDeploy instance
- Configuration

- Custom code

Indexed search

This can be an OpenText LiveSite content server or a similar index search implementation, such as [Apache Solr](#)

OpenText Media Management or MediaBin

MediaBin instance

Custom code for customizations or existing plugins

MediaBin asset repository

MediaBin database

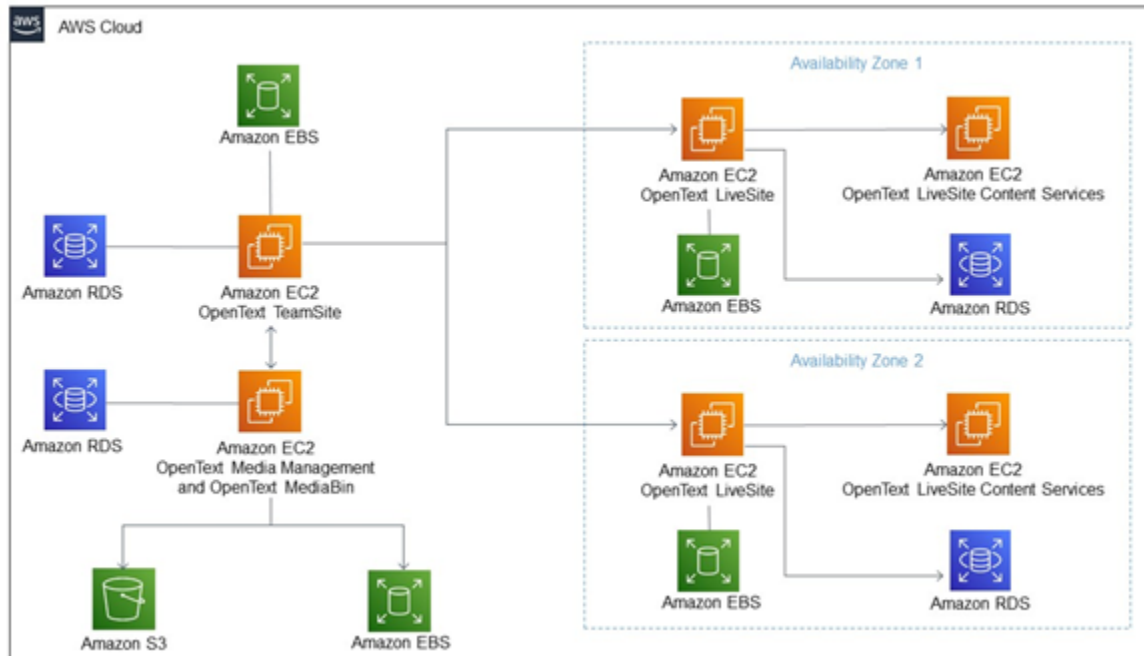
The migration strategy and AWS products and services that you can choose depend on the characteristics of your source system and your individual requirements. The following table describes the most common strategies for the migrations.

Element type	Target AWS services	Migration strategies
OpenText core components	• Amazon Elastic Compute Cloud (Amazon EC2) instances • Containers such as Amazon Elastic Container Service (Amazon ECS) and Amazon Elastic Kubernetes Service (Amazon EKS)	• Rehost • Replatform
	Elastic Kubernetes Service (Amazon EKS)	Typically, you install new instances of the products. Installation of each instance type is fully automated.
Custom functionalities and integrations	• Amazon EC2, integrated with OpenText core components	• Rehost • Replatform • Refactor • Retain

	• Containers (for example, Amazon ECS and Amazon EKS) • Serverless microservices (for example, AWS Lambda) • Amazon API Gateway	Provision and configure the deployment pipelines that are used for the maintenance and evolution of the OpenText platform. These pipelines are used to deploy code. Some dependent functionalities that are built as OpenText TeamSite customizations or as external applications can be containerized or refactored as Lambda functions. If this is the case, you can orchestrate the serverless functions through API Gateway.
Databases	• Amazon Relational Database Service (Amazon RDS)	• Replatform Typically, you can migrate databases to Amazon RDS DB instances by using AWS Database Migration Service (AWS DMS).
Storage	• Amazon Elastic Block Store (Amazon EBS) • Amazon Simple Storage Service (Amazon S3)	• Rehost Data repositories are copied to the Amazon EBS volumes associated with the OpenText core component instances. S3 buckets can be used for larger data repositories, such as OpenText MediaBin or the Media Management assets repository.

Target architecture

The target architecture, migration strategy, and technical steps depend on the source architecture, connectivity constraints, security requirements, and other policies. They also depend on your organization's individual policies or requirements. The following diagram shows a sample architecture on the AWS Cloud that could host your OpenText platform.



The following table provides an overview of the high-level steps required for a migration from a source environment to the target AWS architecture. Detailed steps for this migration are available in the pattern [Migrate OpenText TeamSite workloads to the AWS Cloud](#) on the AWS Prescriptive Guidance website.

High-level steps	Description
1	Provision the AWS infrastructure Use AWS CloudFormation to automatically provision your new AWS infrastructure.
2	Set up DevOps repositories, tools, and procedures Use AWS products and services (for example, AWS CodeCommit or AWS CodePipeline) to build a continuous integration and

continuous deployment (CI/CD) pipeline and automate your release process.

3	Install new OpenText instances	We recommend that you install new OpenText software to run a step-by-step migration and help troubleshoot issues.
4	Migrate databases and content	This is an iterative process and an initial copy is made from the production databases and content repositories. You then run and test the migration process in a non-production environment. Incremental or live updates must be run until the go-live date to ensure that no data is lost during the migration.
5	Migrate OpenText LiveSite	Migrate the LiveSite web server content.
6	Configure OpenText TeamSite, LiveSite, Media Management, and MediaBin	Copy configuration files and adapt the configuration to the new environments.
7	Deploy code	Deploy code to the new environments, for both customizations to OpenText solutions and additional functionalities.

8	Smoke testing	Preliminary testing to validate that the new OpenText platform can go live.
9	Go live	At this stage, the project's goals are accomplished and the final users are using the new OpenText platform. The platform is ready for further realization and maintenance.

Important

These steps don't represent the complete migration process. They only represent the technical steps for the migration implementation.

Before you begin implementing these high-level migration steps, you must define the migration inventory, analyze existing architecture and migration requirements, define the planned architecture, create the automation scripts, and refactor code where required.

The implementation of the high-level steps will not happen in a clear sequence. There are tests, changes, re-implementations, and processes that run at the same time. These depend on the specific characteristics of each migration and the following diagrams shows a sample migration sequence.



High-level migration process flow

The migration process occurs in five phases: discovery, design, development, testing and refinement, and implementation. The following table provides an overview of these five phases and the migration process flow.

	Discovery	Design	Development	Testing and refinement	Implementation
Outcomes	- Existing system architecture - Migration inventory - Functional and non-functional requirements - Future system architecture (High-level)	- Migration strategy for each component (7 Rs) - Target AWS architecture - TCO analysis and assessment - High-level migration plan	- Ready-to-deploy applications - Automation scripts in CloudFormation - A step-by-step migration plan, including automated and non-automated tasks - Detailed test plans	- Testing reports - Refined automation scripts, migration plans, and testing plans	- An OpenText Customer Experience platform running on the AWS Cloud - Testing reports
Tasks	- Discovery workshops - Documentation analysis	Multi-disciplinary subject matter expert	- Multi-disciplinary SME meetings - Application refactori	- Automated testing - Adapt automation and	- Migration plan implementation - Automated testing

		(SME) meetings	ng and adaptation	migration plans	
		• Cost analysis • Validation workshops	• Adapt and create automation scripts • Create or provision new services or functionalities		
Customer resources	• Technical and business documentation • Business owners and technical architects – Participation in workshops and interviews	• Current TCO data • Business owners and technical architects – Participation in validation workshops and interviews. • Access to technical assets (environments, source code, or deployment pipelines)	• Access to technical assets (environments, source code, or deployment pipelines) • Technical teams – Support about current applications.	• User testing • Infrastructure teams – Support for migration activities	• User testing • Infrastructure teams – Support for migration activities

Accelerators	• OpenText architecture analysis framework • Diagrams • Focus points	• Existing knowledge about the most common migration strategies and target AWS products and services for each solution component	Existing automation assets for: • Provisioning of AWS resources • Data migration • Application installation and configuration	Automated testing assets creating during previous migrations, such as: • System health checks • Performance and Load testing scripts	Automated testing assets creating during previous migrations, such as: • System health checks • Performance and Load testing scripts
---------------------	--	--	--	---	---

Automation and tooling

We recommend automating your migration by building automation scripts, running them in nonproduction environments, testing and adjusting, and then running them until the environment set-up, content migration, and code deployments happen smoothly and create a fully functional environment.

Automation scripts and tools are custom-made and based on existing assets from previous migrations. Some of the main assets that are developed for a typical migration include the following:

- AWS CloudFormation scripts to fully provision the AWS services
- AWS Database Migration Service (AWS DMS) configuration, including database migration scripts if required
- Configuration of CI/CD tools according to your requirements (for example, [AWS CodeDeploy](#), Jenkins, or Bamboo).
- Custom scripts (for example, .sh) for additional tasks

When you run and refine the whole migration process, you should also create a detailed step-by-step guide with scripts, manual actions, tests, and fallback options.

Resources

- [Migrate OpenText TeamSite workloads to the AWS Cloud](#)
- [OpenText AWS migration playbook](#)
- [Cloud Based Experience Platform](#)

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Initial publication	—	June 24, 2021