central data
TECHNOLOGY
a member of CTI Group

AWS Transform: Agentic AI for Enterprise IT Transformation

Topics

1. What Is AWS Transform?
2. AWS Transform Architecture
3. Migration Assessment and Discovery
4. Infrastructure Migration
5. Application Modernization and Containerization
6. Mainframe and Custom Code Modernization
7. Continuous Modernization
8. Security, Governance, and Cost Considerations
9. Conclusion

What Is AWS Transform?

AWS Transform is an agentic AI-powered service designed to accelerate and simplify the transformation of infrastructure, applications, and code. It provides specialized AI agents that help organizations manage complex migration and modernization activities across different workloads.

Unlike conventional tools that focus on individual migration or development tasks, AWS Transform coordinates multiple activities within a unified workspace. These activities include infrastructure discovery, migration planning, dependency analysis, code transformation, testing, and deployment preparation.

AWS Transform supports several transformation requirements, including migrating virtual and bare-metal server environments to Amazon EC2, modernizing IBM z/OS and Fujitsu GS21 mainframe applications, upgrading .NET applications, transforming SQL Server databases to Amazon Aurora PostgreSQL, and performing source-code containerization.

a. Agentic AI-Powered Transformation

AWS Transform uses specialized AI agents to execute tasks based on the transformation objective defined by users. Instead of manually coordinating every activity, users can describe their desired outcome through natural language.

AWS Transform translates these objectives into structured plans consisting of multiple tasks. The agents then execute the required activities and request additional information or human approval when necessary.

b. Infrastructure Migration

AWS Transform supports the migration of virtual and physical server environments from platforms such as VMware, Hyper-V, KVM, and other supported infrastructure sources to Amazon EC2. Its capabilities cover infrastructure discovery, migration planning, landing-zone creation, network migration, server replication, testing, and cutover.

c. Application and Code Modernization

AWS Transform helps organizations update legacy applications and source code to more modern technologies. This includes .NET modernization, SQL Server modernization, mainframe transformation, and custom code transformations.

Organizations can also use AWS Transform to containerize applications from source code and prepare them for deployment on Amazon ECS or Amazon EKS.

d. Continuous Modernization

Beyond individual migration projects, AWS Transform supports ongoing code analysis and remediation through Continuous Modernization. This capability helps organizations identify

outdated dependencies, security vulnerabilities, code-quality issues, and modernization opportunities across multiple source-code repositories.

AWS Transform Architecture

AWS Transform brings together several interconnected components to coordinate transformation activities. At a high level, users define objectives through a unified interface, AI agents process the requested tasks, and connectors provide access to the infrastructure, applications, and repositories involved in the transformation.

a. User Interaction Layer

The user interaction layer provides the interface through which users communicate with AWS Transform. Users can access the service through its web application and supported developer tools, including Kiro Power, agent plugins, IDE plugins, and the Model Context Protocol (MCP) server. Natural-language interaction allows users to define transformation objectives, ask questions about ongoing jobs, provide additional instructions, and review generated outputs.

b. AI Agent and Orchestration Layer

The AI agent layer manages the execution of transformation activities. AWS Transform uses task-specific agents that handle different workload requirements. For example, an agent responsible for VMware migration performs different activities from an agent designed for .NET modernization. The orchestration process converts a user-defined objective into a sequence of plans and tasks. These tasks can include infrastructure analysis, code transformation, migration execution, and validation.

c. Connector and Asset Layer

Connectors provide AWS Transform with access to customer-owned resources in external systems. These resources, known as assets, can include server environments, databases, source-code repositories, and network configurations. Depending on the transformation workflow, AWS Transform can integrate with infrastructure environments, AWS accounts, and supported source-control platforms such as GitHub, GitLab, and Bitbucket.

d. Transformation and Execution Layer

This layer contains the capabilities used to execute different transformation activities.

These include:

- Migration assessment and infrastructure discovery.
- Landing-zone and network migration.
- Server rehosting and replication.
- Source-code containerization.
- .NET and SQL Server modernization.
- Mainframe modernization.
- Custom code transformation.
- Continuous repository analysis and remediation.

Each capability provides a workflow designed for its specific transformation requirements.

e. Collaboration and Governance Layer

AWS Transform provides a workspace-based permission model that allows teams to collaborate while maintaining control over transformation activities. Users can be assigned roles such as Administrator, Approver, Contributor, and Reader. These roles determine which actions

they can perform within a workspace. Human review and approval checkpoints are also incorporated into important activities, including code changes, application deployment, and migration cutover.

Migration Assessment and Discovery

Before moving workloads to AWS, organizations need to understand their existing infrastructure, workload requirements, and migration costs. AWS Transform provides migration assessment and discovery capabilities to help teams evaluate their current environment and develop a data-driven migration plan.

a. Infrastructure Discovery

The AWS Transform Discovery Tool helps collect infrastructure inventory from VMware vCenter, Microsoft Hyper-V, and imported server records.

Depending on the configured sources and permissions, the tool can collect information such as:

- Server specifications, including CPU, memory, and disk capacity.
- Operating system information.
- CPU and memory utilization.
- Storage performance metrics, including IOPS and throughput.
- Database information, including SQL Server and Oracle discovery.
- Network connections between servers.

The collected information helps organizations understand their existing infrastructure before defining migration requirements.

b. Inventory Data Sources

AWS Transform supports multiple inventory formats, allowing organizations to work with existing discovery data instead of starting from scratch.

Supported data sources include RVTools exports, AWS Migration Evaluator files, Configuration Management Database (CMDB) records, Migration Portfolio Assessment (MPA) files, and AWS Transform Discovery Tool exports.

Organizations can also use the AWS Transform Assessment Data Template to prepare inventory information manually.

The quality and completeness of the inventory affect the accuracy of the resulting assessment. More detailed utilization data, for example, can help improve instance-sizing recommendations.

c. Migration Assessment

Migration assessment helps organizations understand the cost, feasibility, and business value of moving workloads to AWS.

AWS Transform analyzes the uploaded inventory and generates recommendations for suitable AWS services, infrastructure configurations, pricing options, and licensing considerations.

The assessment can also help identify over-provisioned servers, opportunities for right-sizing, potential AWS Graviton adoption, and alternative pricing models.

d. Assessment Scenarios and TCO Analysis

Organizations can configure different assessment assumptions, including target AWS Region, pricing models, Reserved Instance terms, processor architecture, tenancy, and licensing requirements.

AWS Transform supports up to five scenarios within an assessment, allowing teams to compare alternative migration approaches.

For example, organizations can evaluate the cost differences between running SQL Server on Amazon EC2 and migrating it to Amazon RDS for SQL Server.

The assessment generates several deliverables:

- PPTX presentations containing executive summaries and recommendations.
- XLSX workbooks containing server-level recommendations and cost breakdowns.
- PDF reports covering compute, storage, licensing, and cost comparisons.

These outputs provide technical and business teams with information to support migration decisions.

Infrastructure Migration

AWS Transform helps coordinate infrastructure migration from existing environments to AWS. Its migration workflows cover discovery, planning, account preparation, network migration, and server rehosting.

a. Migration Planning

Migration planning helps organizations determine how workloads should be grouped and moved to AWS. AWS Transform analyzes the collected inventory, identifies application relationships and dependencies, and supports the creation of prioritized migration waves.

Organizations can use different migration job types depending on their current infrastructure readiness. For example, an organization that has already established its AWS landing zone and network may proceed directly with migration planning and server rehosting.

b. Landing Zone Creation

A landing zone provides the foundational AWS environment required to host migrated workloads. AWS Transform helps design and deploy a multi-account AWS foundation, including organizational structures, governance controls, and account configurations. This provides a structured environment for managing workloads across multiple AWS accounts.

c. Network Migration

AWS Transform analyzes existing network configurations and translates them into target Amazon VPC architectures. The process includes mapping source network configurations, designing target VPC structures, and generating infrastructure deployment artifacts. Depending on the workflow, AWS Transform can generate Infrastructure as Code (IaC) using supported formats such as AWS CDK, CloudFormation, and Terraform.

d. Server Rehosting

AWS Transform supports server rehosting to Amazon EC2 through guided migration workflows. The process is organized into migration waves, where groups of servers are moved through inventory validation, replication-agent deployment, continuous data replication, testing, and final cutover.

AWS Transform uses AWS Transform MGN for replication activities. Continuous block-level replication keeps source and target environments synchronized before cutover. During migration, teams can monitor server lifecycle states, review replication progress, and identify workloads that are ready for testing or cutover. Human approval checkpoints help ensure that production changes follow the organization's deployment procedures.

central data
TECHNOLOGY
a member of CTI Group

Application Modernization and Containerization

Application modernization involves updating existing applications to use more current technologies, frameworks, and deployment architectures. AWS Transform supports different modernization approaches depending on the application stack and desired target environment.

a. Source-Code Containerization

AWS Transform can automate the containerization of applications from source code. Its AI agent analyzes the application and generates Docker-related artifacts, including the files required to build container images.

The containerization workflow consists of:

1. Reviewing and accepting the security disclaimer.
2. Connecting or uploading the application source code.
3. Analyzing the codebase and generating Docker artifacts.
4. Reviewing generated artifacts and code changes.
5. Publishing container images to Amazon ECR.
6. Generating deployment templates for Amazon ECS or Amazon EKS.
7. Deploying and validating test infrastructure.
8. Cleaning up test resources.
9. Deploying production cutover infrastructure.

This approach allows organizations to move applications toward container-based architectures while maintaining review points throughout the workflow.

b. .NET Modernization

AWS Transform provides an AI-powered .NET agent that helps modernize legacy .NET applications to cross-platform .NET and upgrade existing applications to later versions.

Its capabilities include:

- Assessing .NET repositories and preparing modernization plans.
- Transforming legacy .NET Framework applications.
- Upgrading existing .NET applications.
- Updating supported Azure cloud-native application code to AWS equivalents.
- Performing local build verification and unit-test execution.
- Generating transformation reports and next-step recommendations.

The .NET agent supports various project types, including class libraries, console applications, ASP.NET MVC, Web API, Web Forms, WCF services, and desktop applications.

Developers can access the agent through the AWS Transform web application, Visual Studio IDE, or compatible with AI code companions.

c. SQL Server Modernization

AWS Transform supports the modernization of Microsoft SQL Server databases and associated .NET applications to Amazon Aurora PostgreSQL. The database transformation process includes schema conversion, stored-procedure transformation, and database-object conversion.

Supported activities include converting SQL Server schemas, translating T-SQL stored procedures into PL/pgSQL, updating triggers and functions, and generating synthetic test data for validation.

d. Application Code Transformation and Validation

In addition to database conversion, AWS Transform helps update application code to work with PostgreSQL. This includes updating Entity Framework configurations, converting ADO.NET data-access code, replacing SQL Server providers with Npgsql, and modifying database connection strings.

Automated validation helps check schema compatibility, data integrity, and application functionality. Human review checkpoints allow developers to inspect converted objects and application changes before deployment.

Important: The SQL Server modernization workflow converts and deploys database schemas but does not migrate real production data. Organizations that need to transfer actual database data should use AWS Database Migration Service (AWS DMS).

Mainframe and Custom Code Modernization

Organizations running legacy mainframe applications or maintaining multiple codebases may face complex modernization requirements. AWS Transform provides specialized capabilities for mainframe workloads and custom code transformations to address these requirements.

a. Mainframe Modernization

AWS Transform supports the modernization and migration of IBM z/OS and Fujitsu GS21 workloads to AWS. For IBM z/OS, supported source artifacts include COBOL, Copybooks, PL/I, JCL, CICS System Definitions, BMS, Db2, VSAM, and IMS TM. For Fujitsu GS21, supported file types include PSAM, ADL, and NDB. The service helps analyze application structures, extract business logic, document existing systems, decompose applications, and develop modernization plans.

b. Mainframe Modernization Journey

The mainframe modernization journey is organized into four phases:

Phase 1: Assess

Evaluate modernization readiness, analyze source artifacts, identify candidate applications, and develop an initial modernization plan.

Phase 2: Mobilize

Establish the necessary environments, tools, security controls, operating model, and pilot modernization activities.

Phase 3: Migrate and Modernize

Transform application code and data, establish CI/CD pipelines, perform functional and performance testing, and prepare production cutover.

Phase 4: Operate, Optimize, and Innovate

Monitor application performance, security, compliance, and costs while continuing to optimize and maintain modernized applications.

c. AWS Transform Custom

AWS Transform Custom allows organizations to define repeatable transformations based on their own development standards and technical requirements.

Users can provide natural-language instructions, technical documentation, API specifications, migration guides, and code samples to create transformation definitions.

A transformation definition can contain:

- SKILL.md for core transformation instructions.
- A references folder for supporting documentation.
- A scripts folder for optional scripts used during execution.

Organizations can create custom transformations for code migrations, framework upgrades, API updates, and other modernization requirements.

d. Transformation Execution and Continual Learning

AWS Transform Custom follows a workflow that starts with defining a transformation and testing it on sample codebases. After validating the results, teams can execute transformations across multiple repositories using the AWS Transform CLI.

The continual-learning mechanism collects developer feedback, code fixes, and issues encountered during execution to improve future transformation runs. Organizations can also

review and manage accumulated lessons to ensure that only useful transformation of knowledge is applied in subsequent executions. AWS-managed transformation definitions are also available for supported modernization scenarios, allowing teams to use predefined transformations without creating every definition from scratch.

Continuous Modernization

Modernization is an ongoing process. Applications continue to accumulate technical debt as dependencies, frameworks, and security requirements evolve. AWS Transform Continuous Modernization helps organizations analyze and remediate code issues across connected repositories.

a. Repository Integration

AWS Transform supports connecting source-code repositories from GitHub organizations, GitLab groups, Bitbucket workspaces, and local directories. After connecting the sources, organizations can discover available repositories and organize them using labels based on teams, priorities, or migration waves.

b. Code and Technical Debt Analysis

AWS Transform can run different analysis types to identify issues across source-code repositories.

These analyses include:

- Rapid technical-debt analysis.
- Comprehensive technical-debt analysis.
- Security analysis.
- Agentic-readiness analysis.
- Modernization-readiness analysis.
- Graviton-readiness analysis.

- Custom analysis.

The generated findings contain information such as severity, repository, status, and associated transformation definitions when an automated fix is available.

c. Automated Remediation

AWS Transform can apply transformation definitions to eligible findings and generate code changes. The remediation process can create branches and open Pull Requests or Merge Requests in supported repository platforms. Developers can review generated changes before merging them into their main branches.

d. Continuous Monitoring and Reporting

Organizations can schedule recurring analyses to identify new issues and monitor code health over time. Supported analysis schedules include daily, weekly, and monthly execution. AWS Transform also provides HTML reporting to help teams review findings, severity distribution, and remediation progress.

Continuous modernization can run using local resources, customer-managed infrastructure through Amazon EC2 or AWS Batch, or AWS Transform-managed infrastructure for supported workflows.

Security, Governance, and Cost Considerations

Security and governance are essential when transformation services interact with infrastructure, application code, and organizational resources. AWS Transform provides identity management, encryption, connectors, and approval mechanisms to help organizations manage these activities.

a. Identity and Access Management

AWS Transform supports authentication through AWS IAM Identity Center, third-party identity providers such as Microsoft Entra ID and Okta Workforce Identity, and IAM-only access. Workspace roles provide additional control over user activities, including workspace administration, job execution, artifact access, and critical approvals. Organizations should apply least-privilege permissions when configuring user access, connectors, and target AWS accounts.

b. Data Protection and Encryption

AWS Transform protects customer data through encryption in transit and at rest. Communication between customers and AWS Transform is secured using TLS 1.2 or higher. Data at rest is encrypted by default using AWS-owned keys. Organizations can also use symmetric customer-managed AWS KMS keys for an additional layer of encryption control.

c. Human Review and Governance

AWS Transform includes human approval checkpoints for important transformation activities. These checkpoints allow authorized users to review generated artifacts, validate changes, and approve critical actions before they proceed. AWS CloudTrail and AWS Transform worklogs also provide activity records that help organizations maintain operational traceability.

d. Data Residency

AWS Transform stores data in the AWS Region where its profile was created. However, cross-Region inference may process requests in other Regions within the relevant geography. Organizations with strict data-residency requirements should review the applicable AWS documentation and validate the service configuration against their internal policies.

e. Cost Considerations

AWS documentation states that there is no additional charge to use AWS Transform itself. However, organizations should consider the costs of AWS resources and supporting services used throughout transformation activities. These may include Amazon EC2, AWS Batch, Amazon S3, AWS KMS, Amazon ECR, Amazon ECS, Amazon EKS, and Amazon Aurora PostgreSQL, depending on the selected workflow and deployment architecture. Organizations should therefore evaluate the infrastructure and service costs associated with their transformation projects.

Conclusion

AWS Transform provides an integrated approach to enterprise IT transformation through specialized agentic AI capabilities. Its capabilities cover infrastructure discovery, migration assessment, server rehosting, application modernization, source-code containerization, mainframe transformation, custom code transformation, and continuous modernization.

By coordinating complex activities through specialized agents and structured workflows, AWS Transform helps reduce repetitive manual tasks while allowing technical teams to maintain control over review, validation, and production approvals. The adoption of AWS Transform should begin with an understanding of existing infrastructure, application dependencies, and modernization objectives. Organizations can then select the appropriate transformation workflow, conduct a pilot, validate the results, and gradually expand implementation across their environment.

Ultimately, AWS Transform provides a way for organizations to approach migration and modernization as a coordinated process rather than a collection of disconnected technical activities.

central data TECHNOLOGY
a member of CTI Group

Start Your IT Transformation Journey with Central Data Technology

From infrastructure assessment to application modernization, Central Data Technology (CDT) can help your organization identify migration opportunities and develop transformation approaches based on your technical and business requirements.

CDT can support your organization across:

- Cloud Migration Assessment and Planning
- AWS Infrastructure and Landing Zone Design
- Application and Database Modernization
- Server Migration and Containerization
- AI-Powered Development and Code Transformation
- AWS Migration and Modernization Services

Explore how AWS Transform can support your organization's migration and modernization initiatives with Central Data Technology. **Contact CDT here to discuss your IT transformation requirements.**

Credit to: AWS Documentation – AWS Transform User Guide