AWS Prescriptive Guidance

Automating large-scale server migrations with CloudEndure Migration Factory

AWS Prescriptive Guidance: Automating large-scale server migrations with CloudEndure Migration Factory

Table of Contents

Introduction 1
The CEMF workflow 2
The migration metadata pipeline tool 4
Automation scripts 5
How to get started 7
    Prerequisites 7
Migration process 8
    Step 1. Prepare the CSV file and import the data into CEMF 8
    Step 2. Build the servers 9
        Verify prerequisites for source servers 9
        Install the CloudEndure agent 9
        Push the post-launch script 9
    Step 3. Validate the migration 10
        Verify CloudEndure replication status 10
        (Optional) Create a local administrator account 10
    Step 4. Perform migration boot-up testing 10
        Launch servers for boot-up testing 10
        Verify instance boot-up status 11
        Terminate test instances 11
    Step 5. Cut over 11
        Verify CloudEndure replication status 11
        Shut down source servers in preparation for cutover 12
        Launch target EC2 instances for cutover 12
        Verify instance boot-up status 12
        (Optional) Get new IP addresses for target instances 12
        Test RDP/SSH access to target servers 12
        Reconfigure application and networking settings 13
        Test the application 13
        Complete the cutover 13
FAQ 14
    How do I know if this solution is suitable for my project? 14
    How much does it cost to run CEMF? 14
    How can I deploy or get support for CEMF? 14
    After I deploy CEMF, can AWS help me migrate my servers using this solution? 14
    What permissions do I need to deploy the migration factory solution? 14
    What permissions do I need to run the automation scripts? 15
    What firewall rules do I need to create for the migration execution server? 15
    Can I customize the CEMF schema to capture additional attributes? 15
    Why doesn't the CSV file include all CloudEndure blueprint attributes? 15
    Can I bulk-update the CloudEndure blueprint for all my servers? 15
    Does CEMF support SAML federation? 15
    Using Launch Server or Check Status in CEMF displays an "Unexpected timeout" error. 16
AWS Prescriptive Guidance glossary 17
Document history 24

Automating large-scale server migrations with CloudEndure Migration Factory

Wally Lu, Mass Migration Lead, AWS Professional Services

April 2020 (last update (p. 24): June 2020)

Many companies today want to migrate their servers to Amazon Web Services (AWS) as quickly as possible. This is not an easy job, especially if you want to migrate thousands of servers in a short period of time, such as 6–12 months.

Large-scale migrations present a few challenges:

- Integrating multiple tools is difficult. There are many tools that support migration, such as discovery tools, migration tools, and configuration management database (CMDB) tools. These tools must be connected so that data flows from one tool to another. However, different tools use different data formats, and that makes integration difficult. A migration is more likely to be successful if there is a way to integrate all the tools.
- Manual processes are slow and hard to scale. Migrations involve many small tasks, and each task takes a few minutes to complete. A migration is faster when these tasks are automated.

CloudEndure Migration Factory (CEMF) was designed to solve these problems for migrations that require rehosting (lift and shift). CloudEndure simplifies, expedites, and reduces the cost of cloud migration by offering a highly automated lift-and-shift solution. CEMF is an orchestration platform powered by CloudEndure for rehosting servers to AWS at scale. It helps customers with their medium-scale to large-scale migrations by automating manual processes, which are often slow or complex to scale. Thousands of servers have been migrated to AWS using CEMF. For example, AWS customers used CEMF to migrate 1,200 servers in 5 months, and were able to cut over more than 600 servers in a single cutover window.

This guide describes the CEMF process for rehosting servers at scale, for migration architects, program managers, and technical leads. For more information about migration factories, see Mobilize your organization to accelerate large-scale migrations on the AWS Prescriptive Guidance website.

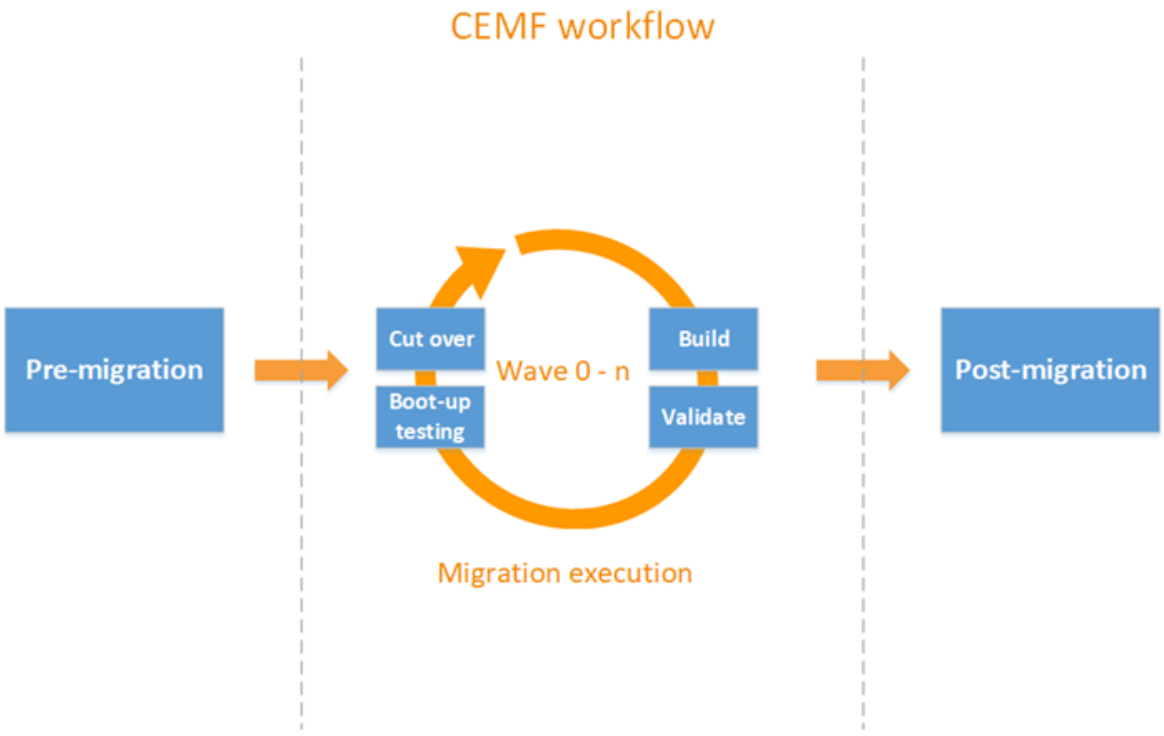
> **How to get access to CEMF**
> CEMF is available to all AWS customers and partners. To use CEMF, see the AWS CloudEndure Migration Factory Solution on the AWS Solutions website. The source code is available in a GitHub repository. If you have any questions, email AWS Professional Services at migration-factory-support@amazon.com.

If you want to get hands-on experience before using CEMF for your production migration, email us at migration-immersion-day@amazon.com to arrange a migration immersion day.

> **Note**
> AWS Application Migration Service (MGN) is the primary migration service recommended for lift-and-shift migrations to the AWS Cloud. Customers who currently use CloudEndure Migration or AWS Server Migration Service (AWS SMS) are encouraged to switch to MGN for future migrations.

The CEMF workflow

CEMF comes with a predefined process that includes three phases: pre-migration, migration implementation, and post-migration, as shown in the following diagram.

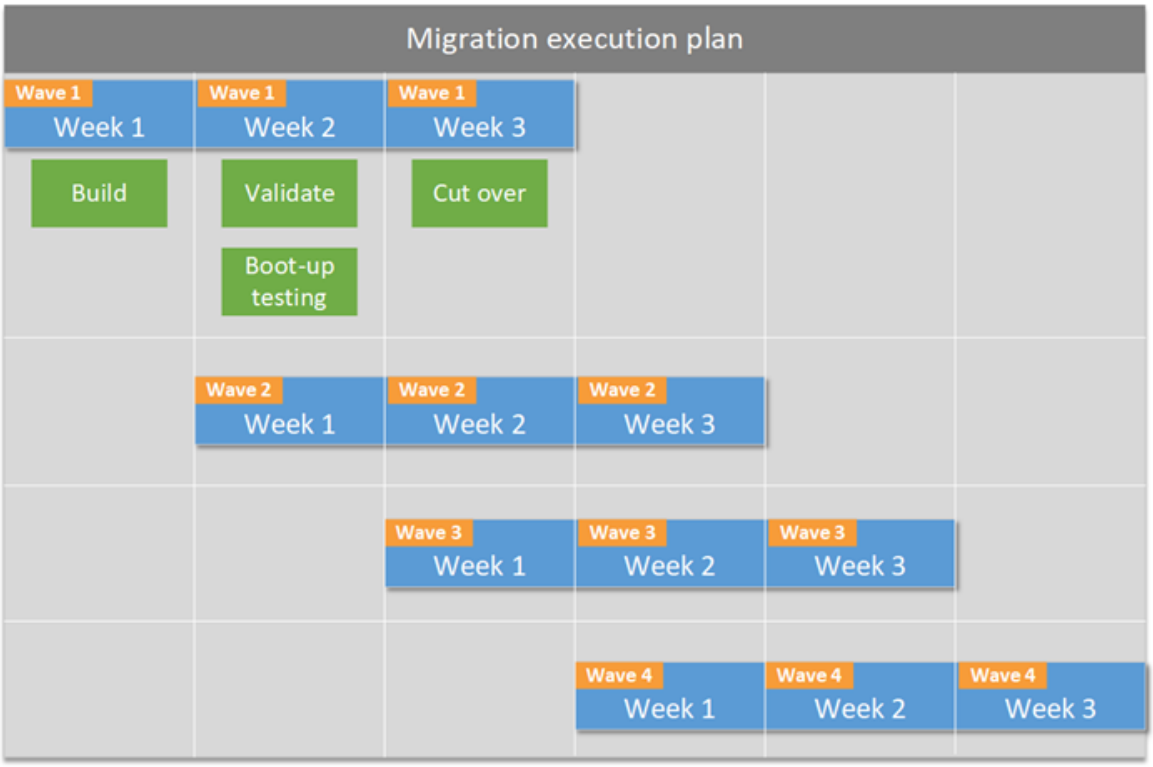


In the **pre-migration phase**, your migration team is responsible for preparing the implementation environment. This includes deploying CEMF, building a migration execution server, and setting up a CloudEndure Migration account.

During the **migration implementation phase**, the migration team is responsible for running predefined tasks that automate the migration process. These tasks can include:

- Verifying prerequisites
- Pushing the CloudEndure agent to the source machines for a given wave
- Verifying replication status
- Launching servers for boot-up testing
- Scheduling a window for application cutover

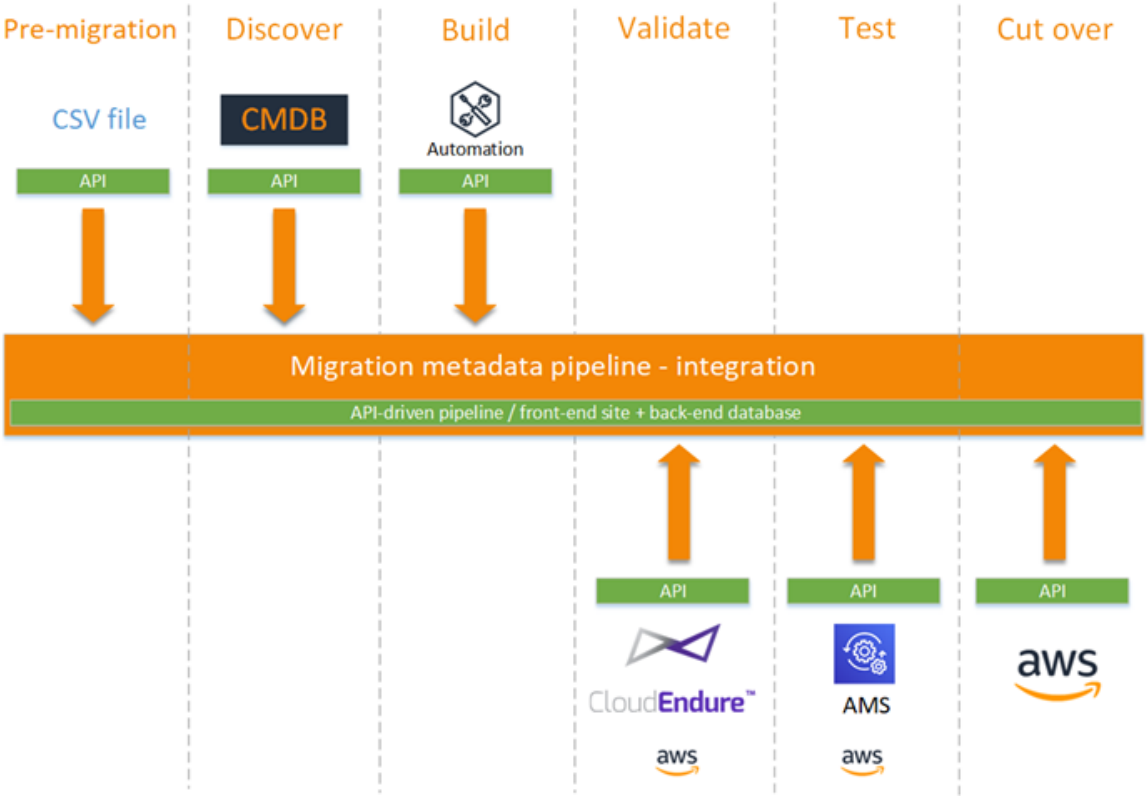
Migration tasks are scheduled in *waves*. Each wave consists of a group of applications and servers that have the same cutover date. As shown in the following diagram, each wave should be completed in a predefined period. For example, in the three-week period shown, week 1 is the build stage, week 2 is the validate and boot-up testing stage, and week 3 is the cutover stage. All the waves run in parallel.



Post-migration tasks depend on the specific migration scenario and your requirements. These tasks might include removing servers from the source CMDB, decommissioning source machines, and optimizing performance for the target Amazon Elastic Compute Cloud (Amazon EC2) instances.

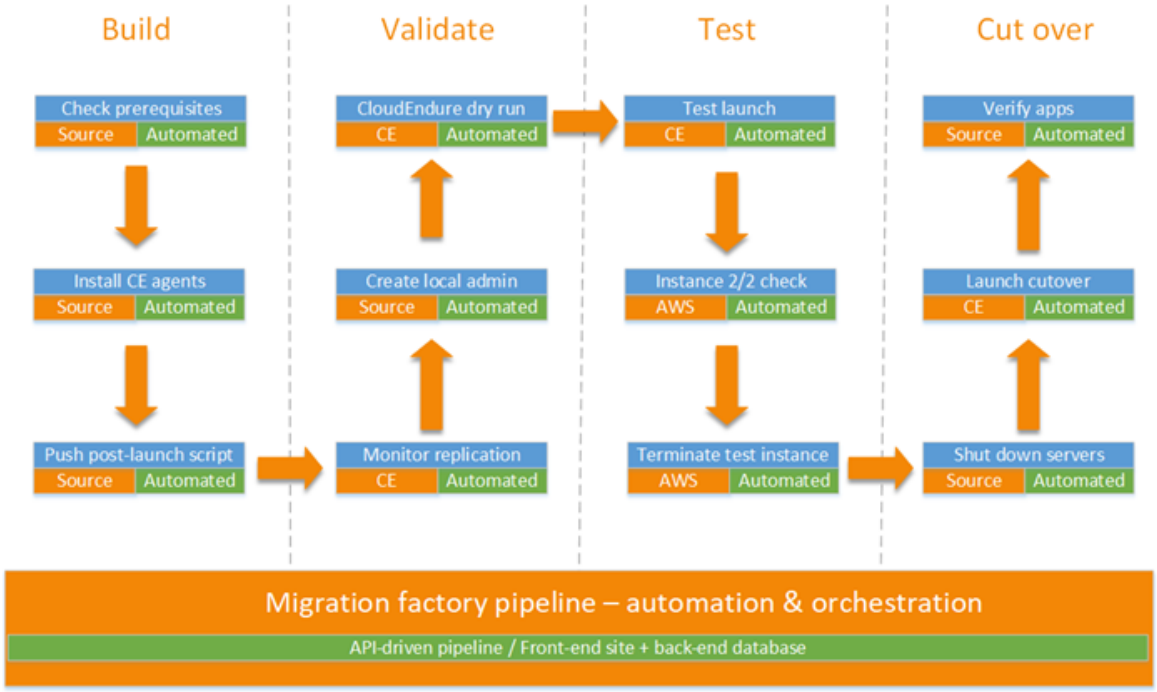
The CEMF migration metadata pipeline tool

CEMF includes a migration metadata pipeline tool and automation scripts. The metadata pipeline tool integrates with other migration tools and scripts through Representational State Transfer (REST) APIs, as shown in the following diagram. This enables migration metadata to flow from one tool to another to support end-to-end automation. Currently, CEMF is natively integrated with CloudEndure APIs and the AWS Managed Services (AMS) workload ingest process. By integrating these tools and processes, CEMF can automate migration tasks across multiple tools.



CEMF automation scripts

The following diagram shows the automation scripts included in CEMF. These scripts cover most of the automation tasks for rehost migration using CloudEndure. Automation scripts can be connected to the source machines, to the CloudEndure API, or to AWS APIs, as shown in the diagram.



CEMF includes scripts for the following tasks:

- Build phase:
  - Checking prerequisites for the migration
  - Installing CloudEndure agents for multiple servers
  - Pushing the post-launch script
- Validate phase:
  - Monitoring replication
  - Creating the local administrator account
  - Performing a CloudEndure dry run
- Test phase:
  - Testing the EC2 instance launch
  - Performing 2/2 (system status and instance status) health checks on instances
  - Terminating test instances
- Cutover phase:
  - Shutting down servers at the source location
  - Orchestrating the CloudEndure cutover process
  - Verifying that all application servers are up and running

These automation scripts help you save significant time and effort in your large-scale migration tasks. For example:

- Automating CloudEndure agent installation for 100+ servers. Installing the CloudEndure agent on one server takes about 5 minutes. However, if you have 100 servers running Microsoft Windows and Linux for 10 different CloudEndure projects, there can be 20 different ways to install the agent on the source machine, and this process could take over 500 minutes. The automation script reduces the agent installation time from 500 minutes to less than 5 minutes of operator time, and it works for both Windows and Linux operating systems and any target AWS account.
- Orchestrating the CloudEndure cutover process. This process involves replication status checking, server status checking, updating the blueprint, launching servers in cutover mode, verifying job status, server clean-up, and many other tasks. It is a long process even for one server, and it could be a nightmare if you have hundreds of servers in a single cutover. The CEMF solution automates and orchestrates the entire process for you.

These tasks and the CEMF scripts that automate them are described in more detail in the following sections.

How to get started with CEMF

The CloudEndure Migration factory (CEMF) solution is available to all AWS customers and partners, and it can be deployed to your AWS account in just a few minutes. To deploy CEMF, see the AWS CloudEndure Migration Factory Solution on the AWS Solutions website. The source code is available in a GitHub repository. If you have any questions, email AWS Professional Services at migration-factory-support@amazon.com.

Prerequisites

CEMF requires the following:

- Set up your AWS infrastructure.
- Complete the initial portfolio discovery and wave planning.
- Register a CloudEndure account and complete project setup so that the CloudEndure API token is ready to use.
- Follow the instructions in the AWS CloudEndure Migration Factory Solution implementation guide to deploy CEMF.

After you complete these prerequisites, we can help you complete the steps described in the following sections to perform the migration. If you have multiple waves, you must repeat the steps for each wave. The recommended wave size is 25–35 servers. If you are planning to cut over more (for example, 100 servers) in the same cutover window, we recommend that you split the 100 servers into multiple waves and run the automation multiple times, because smaller waves are easier to troubleshoot from our experience.

The CEMF migration process

The CEMF process consists of five main steps. These are discussed in the following sections:

- Step 1. Prepare the CSV file and import the data into CEMF (p. 8)
- Step 2. Build the servers (p. 9)
- Step 3. Validate the migration (p. 10)
- Step 4. Perform migration boot-up testing (p. 10)
- Step 5. Cut over to the new server instances on AWS (p. 11)

Important
The automation scripts for the migration steps described in this guide require CEMF. To get CEMF, see the AWS CloudEndure Migration Factory Solution on the AWS Solutions website. If you have any questions, email AWS Professional Services at migration-factory-support@amazon.com.

Step 1. Prepare the CSV file and import the data into CEMF

The first step in a large-scale migration is to prepare the application and server metadata. This metadata is usually collected from portfolio analysis and wave planning, and placed in a comma-separated values (CSV) file. You use the metadata to automate the migration process and to update the CloudEndure blueprint in order to launch target EC2 instances. The default CSV file that we'll provide includes the following attributes:

- `wave_id` – Unique wave ID, based on priority and dependencies.
- `app_name` – Application to be migrated.
- `cloudendure_projectname` – CloudEndure project created for the migration.
- `aws_accountid` – 12-digit account ID of the destination AWS account.
- `server_name` – On-premises server to be migrated.
- `server_os` – Operating system (Windows or Linux) running on the on-premises server.
- `server_os_version` – Version of the server operating system.
- `server_fqdn` – Fully qualified domain name (FQDN) of the server.
- `server_tier` – Type of server (web, application, or database).
- `server_environment` – Host environment for server (development, test, production, QA, pre-production).
- `subnet_IDs` – IDs of the AWS subnets to be used for the EC2 instances after cutover.
- `securitygroup_IDs` – IDs of the AWS security groups to be used after cutover.
- `subnet_IDs_test` – IDs of the AWS subnets to be used for testing.
- `securitygroup_IDs_test` – IDs of the AWS security groups to be used for testing.
- `instanceType` – EC2 instance type to be used for the servers.

- `tenancy` – Whether the instance runs on shared hardware, on single-tenant hardware (dedicated), or on an isolated server (Dedicated Host).

We recommend that you prepare a new CSV file for each migration wave.

CEMF includes an automation script that you run on the migration execution server to import the metadata from the CSV file to CEMF.

For detailed instructions, see Import the Application and Server Data in the CEMF implementation guide.

Step 2. Build the servers

After you import the application and server metadata, you verify the source machines and install the CloudEndure agent to start data replication.

Verify prerequisites for source servers

In this step, you make sure that your source servers have the required configuration to start the data replication. For example, if the source server is a Windows server, it must meet these requirements:

- TCP port 443 outbound must be open for the source machine to connect to the CloudEndure console.
- TCP port 1500 outbound must be open for the source machine to connect to the CloudEndure replication server in the target virtual private cloud (VPC) on AWS.
- The server must be running .NET Framework version 3.5 or later.
- The server must have at least 3 GB of free space on drive C.

CEMF includes an automation script that you run on the migration execution server to verify the prerequisites automatically for all source Windows and Linux servers.

For detailed instructions, see Check the Prerequisites in the CEMF implementation guide.

Install the CloudEndure agent

After checking the prerequisites, you install the CloudEndure agent on the source machines. This process usually takes 5-10 minutes per server, but CEMF includes an automation script to push the agents to all the source servers in the same wave. This script works for multiple target CloudEndure projects and AWS accounts.

The agent installation script uses the CloudEndure API token to pull the installation token for the target CloudEndure project. Follow the instructions in the CloudEndure documentation to get the API token if you do not have it already.

For detailed instructions, see Install the CloudEndure Agents in the CEMF implementation guide.

Push the post-launch script

One of the common tasks of rehost migration is uninstalling old software such as VMware tools and backup software from target EC2 instances, and installing new software such as the AWS Systems Manager agent. Completing these activities manually could take 15–30 minutes per server, but CEMF automates this process to accelerate the cutover time.

CloudEndure supports post-launch scripts to help you automatically run operating system configuration tasks such as installing or uninstalling software.

For detailed instructions, see Push the Post-Launch Scripts in the CEMF implementation guide.

Step 3. Validate the migration

After installing the CloudEndure agent on the source machines, you monitor the status of data replication and resolve issues such as permissions or network performance.

Verify CloudEndure replication status

If you have a small migration, you can verify the replication status manually from the CloudEndure console. However, if you have large-scale migrations, servers across multiple projects, and servers in multiple waves, this verification can be difficult. For example, if you have 100 servers in wave 1, you must repeat the following steps 100 times to verify their replication status:

- Find the target CloudEndure project for the server.
- Log in to the CloudEndure console, select the right project, and search for the server name.
- Check the progress bar and update the status of the server on your spreadsheet.

CEMF includes an automation script that you run once for all servers. The script retries every 5 minutes until the status of every server in wave 1 changes to *Continuous Data Replication*, and it updates the replication status in the CEMF database. You can troubleshoot replication issues by following the steps in the CloudEndure documentation or contacting AWS Support.

For detailed instructions, see Verify the CloudEndure Replication Status in the CEMF implementation guide.

(Optional) Create a local administrator account

Cutover is not always smooth for all servers, so you should have a plan B. One common issue during cutover is problems logging in to the target EC2 instances with Active Directory domain user or Lightweight Directory Access Protocol (LDAP) user accounts, because the target machine isn't able to communicate with the domain controller or LDAP servers. In this scenario, having a local administrator account is critical, because a local administrator can log in to the target EC2 instance with full permissions and doesn't have dependencies on external identity providers such as Active Directory domain controllers or LDAP servers.

For detailed instructions, see Create the Admin User in the CEMF implementation guide.

Step 4. Perform migration boot-up testing

After data replication is complete on all servers, you need to test the instance boot-up process and make sure that everything works as expected from the operating system perspective. That is, the EC2 instance must pass the 2/2 (system status and instance status) health checks.

Launch servers for boot-up testing

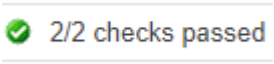
If you're migrating a small number of servers, you can select the server and launch it directly from the CloudEndure console. However, for large-scale migrations, it's more efficient to launch all the servers together from the CEMF web console. This console provides a single **Launch servers** button to automate the following processes:

- Verifying replication status and making sure that the lag time is less than 180 minutes.
- Updating the CloudEndure blueprint for all servers in the given wave with the metadata in the CEMF database.
- Sending all servers to a CloudEndure job and launching them in test mode.
- Avoiding mistakes by disabling server relaunch by default, but keeping the option to enforce a server relaunch.

For detailed instructions, see Launch Instances for Testing or Cutover in the CEMF implementation guide.

Verify instance boot-up status

It will take 15–30 minutes for the server instances to boot up. You can check the status manually by logging into the Amazon EC2 console, searching for the server name, and checking the status. You will see a "2/2 checks passed" message, which indicates that the instance is healthy from an infrastructure perspective.



However, for a large-scale migration, it's time-consuming to check the status of each instance, so CEMF provides a single automation script to verify the 2/2 status for all machines in a given wave.

If an instance fails the 2/2 status checks, contact AWS Support for assistance.

For detailed instructions, see Verify the Target Instance Status in the CEMF implementation guide.

Terminate test instances

After testing, you can manually terminate the instances from the Amazon EC2 console to avoid charges. However, it is more efficient to terminate all instances by using the single automation script that CEMF provides.

For detailed instructions, see Terminate the Test Instances in the CEMF implementation guide.

Step 5. Cut over

The last step in a typical rehost migration is to schedule a cutover window and prepare the resources to support the cutover.

Verify CloudEndure replication status

First, you must verify CloudEndure replication status and make sure that the status of all servers in the given wave is *Continuous Data Replication*.

As in step 3 (p. 10), you can run a CEMF script to automate this step. The script retries every 5 minutes until the status of every server in the given wave changes to *Continuous Data Replication*, and updates the replication status in the CEMF database.

For detailed instructions, see Verify the CloudEndure Replication Status in the CEMF implementation guide.

Shut down source servers in preparation for cutover

After you verify the source servers' replication status, you are ready to shut down the source servers to stop transactions from the client applications to the servers. Usually, you can shut down the source servers in the cutover window. Shutting down the source servers manually could take you 5 minutes per server, and, for large waves, it could take a few hours in total. Instead, you can run a CEMF automation script to shut down all your servers in the given wave.

For detailed instructions, see Shut Down the In-Scope Source Servers in the CEMF implementation guide.

Launch target EC2 instances for cutover

After shutting down the source servers, you can launch the target EC2 server instances. As in step 4 (p. 10), you can use a single **Launch servers** button to launch all the servers in the given wave in cutover mode. The only difference here is that you choose **Cutover** as the launch type. As in boot-up testing, the **Launch servers** button automates the following processes:

- Verifying replication status and making sure that the lag time is less than 180 minutes.
- Updating the CloudEndure blueprint for all servers in the given wave with the metadata in the CEMF database.
- Sending all servers to a CloudEndure job and launching them in cutover mode.
- Avoiding mistakes by disabling server relaunch by default, but keeping the option to enforce a server relaunch.

For detailed instructions, see Launch Instances for Testing or Cutover in the CEMF implementation guide.

Verify instance boot-up status

After launching the instances in the cutover mode, wait for at least 15 minutes before the next step, which is verifying instance boot-up status. When cutover launch is complete, you can run the CEMF automation script to verify the 2/2 status for all machines in the given wave.

If an instance fails the 2/2 status checks, contact AWS Support for assistance.

For detailed instructions, see Verify the Target Instance Status in the CEMF implementation guide.

(Optional) Get new IP addresses for target instances

If the target server instances use new IP addresses, the next step is to update the DNS server with the new IP addresses. In some scenarios, target instances support dynamic DNS registration and register the new IP address automatically with the DNS server. For example, if a Windows server uses a domain controller as the DNS server, DNS registration could be automatic. On the other hand, if the DNS update is a manual process, you need to get the new IP addresses for all the target instances. In this case, you can use the CEMF automation script to export the new IP addresses for all the instances in the given wave to a CSV file.

For detailed instructions, see Retrieve the Target Instance IPs in the CEMF implementation guide.

Test RDP/SSH access to target servers

After you update the DNS records, you can connect to the target instances with the host name. In this step, you check to see if you can log in to the operating system by using Remote Desktop Protocol (RDP) or through Secure Shell (SSH) access. You can manually log in to each server individually, but it is more efficient to test the server connection by using the CEMF automation script.

For detailed instructions, see Verify the Target Server Connections in the CEMF implementation guide.

Reconfigure application and networking settings

After the migration team completes operating system level testing, the application team makes changes at the application level. These changes might include the following:

- If the application requires a load balancer, change the application endpoint in the load balancer to point to the new instance IPs in AWS.
- Change the connection string for the application web tier to connect to the database.
- Change other application-specific settings.

Test the application

Application testing, which takes place after the updates described in the previous section, is generally handled by the application owner or support team. It involves logging in to the new servers and confirming that the application works as expected. If it doesn't, the application owner or support team works with the migration team to troubleshoot and fix issues.

Complete the cutover

This is the final step of the migration. The application owner decides whether the target application in AWS meets their expectations from both functionality and performance perspectives. If a rollback is required, it usually involves these activities:

- Terminating all AWS instances for the affected application.
- Turning on on-premises servers for the given application.
- Reverting DNS records to the old server IP addresses.

CloudEndure Migration Factory FAQ

This section provides answers to commonly raised questions about deploying and using CloudEndure Migration Factory.

How do I know if this solution is suitable for my project?

CEMF is suitable for customers who have more than 100 servers that they want to rehost on AWS. Those servers must also be within the list of operating systems supported by CloudEndure.

How much does it cost to run CEMF?

CEMF is a serverless solution that uses AWS services such as Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, Amazon DynamoDB, and AWS Lambda. Your cost depends on your usage.

How can I deploy or get support for CEMF?

To deploy CEMF, see the AWS CloudEndure Migration Factory Solution on the AWS Solutions website. The source code is available in a GitHub repository. If you have any questions, email AWS Professional Services at migration-factory-support@amazon.com. We will be happy to help you.

After I deploy CEMF, can AWS help me migrate my servers using this solution?

Yes, the AWS Professional Services team can help you. Please email us at migration-factory-support@amazon.com and tell us about your use cases. This process includes:

- An initial conversation to discuss your business case
- A one-hour CEMF introduction and demo session with your team
- Migration immersion day with CEMF, including a half-day lab session followed by Q&A
- Discussion of resources you might need from AWS Professional Services

What permissions do I need to deploy the migration factory solution?

Ideally, you will need to use AWS administrator credentials to deploy the stack. If not, we can provide AWS Identity and Access Management (IAM) policies for you to attach to the user who will deploy the factory.

What permissions do I need to run the automation scripts?

For Windows, you need a domain user with local administrator credentials to access all Windows servers. For Linux, you need a sudo user with `NOPASSWD` enabled.

What firewall rules do I need to create for the migration execution server?

The migration execution server needs to access source servers and external APIs (CloudEndure, CEMF, and AWS APIs). The server needs access to the internet to access these APIs. You can use a proxy server, but if you do, the proxy server can't use authentication. The server needs SSH access to the source Linux server on TCP port 22, or Windows Remote Management (WinRM) access to the source Windows server on TCP port 5985.

Can I customize the CEMF schema to capture additional attributes?

Yes, you can modify or extend the schema to capture additional server or application attributes.

Why doesn't the CSV file include all CloudEndure blueprint attributes?

CEMF hides uncommon attributes such as IAM role and public IP. However, you can add these attributes to the CSV file and import them to update the blueprint.

Can I bulk-update the CloudEndure blueprint for all my servers?

Yes, you can set hardcoded values for any CloudEndure blueprint attribute. For example, CEMF sets a private IP for all servers and sets the disk to use `gp2` volumes by default, but you can update these attributes.

Does CEMF support SAML federation?

Security Assertion Markup Language (SAML) federation isn't currently supported.

Using Launch Server or Check Status in CEMF displays an “Unexpected timeout” error.

If you receive a timeout error, your CEMF login token might have expired. Refresh the web page to get a new token.

AWS Prescriptive Guidance glossary

AI and ML terms (p. 17) | Migration terms (p. 18) | Modernization terms (p. 22)

AI and ML terms

The following are commonly used terms in artificial intelligence (AI) and machine learning (ML)-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

binary classification	A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"
classification	A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.
data preprocessing	To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.
deep ensemble	To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.
deep learning	An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.
exploratory data analysis (EDA)	The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.
features	The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.
feature transformation	To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

multiclass classification	A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"
regression	An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).
training	To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.
target variable	The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.
tuning	To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.
uncertainty	A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

Migration terms

The following are commonly used terms in migration-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

7 Rs	Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following: • Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition. • Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud. • Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com. • Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.

	• Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. This migration scenario is specific to VMware Cloud on AWS, which supports virtual machine (VM) compatibility and workload portability between your on-premises environment and AWS. You can use the VMware Cloud Foundation technologies from your on-premises data centers when you migrate your infrastructure to VMware Cloud on AWS. Example: Relocate the hypervisor hosting your Oracle database to VMware Cloud on AWS. • Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them. • Retire – Decommission or remove applications that are no longer needed in your source environment.
application portfolio	A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.
artificial intelligence operations (AIOps)	The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.
AWS Cloud Adoption Framework (AWS CAF)	A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.
AWS landing zone	A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.
AWS Workload Qualification Framework (AWS WQF)	A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.
business continuity planning (BCP)	A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.
Cloud Center of Excellence (CCoE)	A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-

	scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.
cloud stages of adoption	The four phases that organizations typically go through when they migrate to the AWS Cloud: • Project – Running a few cloud-related projects for proof of concept and learning purposes • Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model) • Migration – Migrating individual applications • Re-invention – Optimizing products and services, and innovating in the cloud These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.
configuration management database (CMDB)	A database that contains information about a company's hardware and software products, configurations, and inter-dependencies. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.
epic	In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.
heterogeneous database migration	Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.
homogeneous database migration	Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.
idle application	An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.
IT information library (ITIL)	A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.
IT service management (ITSM)	Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.
large migration	A migration of 300 or more servers.
Migration Acceleration Program (MAP)	An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration

	methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.
Migration Portfolio Assessment (MPA)	An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.
Migration Readiness Assessment (MRA)	The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.
migration at scale	The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.
migration factory	Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the CloudEndure Migration Factory guide in this content set.
migration metadata	The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.
migration pattern	A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.
migration strategy	The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs (p. 18) entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.
operational-level agreement (OLA)	An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).
operations integration (OI)	The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.
organizational change management (OCM)	A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

playbook	A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.
portfolio assessment	A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.
responsible, accountable, consulted, informed (RACI) matrix	A matrix that defines and assigns roles and responsibilities in a project. For example, you can create a RACI to define security control ownership or to identify roles and responsibilities for specific tasks in a migration project.
runbook	A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.
service-level agreement (SLA)	An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.
task list	A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.
workstream	Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.
zombie application	An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.

Modernization terms

The following are commonly used terms in modernization-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

business capability	What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.
microservice	A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.
microservices architecture	An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for

	specific functions of an application. For more information, see Implementing microservices on AWS.
modernization	Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.
modernization readiness assessment	An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.
monolithic applications (monoliths)	Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.
polyglot persistence	Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.
split-and-seed model	A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.
two-pizza team	A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development. For more information, see the Two-pizza team section of the Introduction to DevOps on AWS whitepaper.

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

update-history-change	update-history-description	update-history-date
Added links to the CEMF implementation guide for step-by-step deployment and migration instructions. (p. 24)	CEMF is now available as an AWS solution.	June 4, 2020
Initial publication (p. 24)	—	April 30, 2020