Foundation playbook for AWS large migrations

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Foundation playbook for AWS large migrations

Table of Contents

Foundation playbook for AWS large migrations

Amazon Web Services (contributors)

February 2021 (document history)

A large migration project is built upon its people foundation and platform foundation. Properly preparing these foundations is critical to the success of the project. *Platform* refers to the technology decisions you make, such as infrastructure, operations, and security. *People* refers to the teams and individuals who contribute to the project, from beginning to end.

In this playbook, you build the foundation workstream. Because this workstream is intended to prepare the platform and people before you begin migrating applications, you start and complete this workstream within the first stage of a large migration, initialization. For more information about core and supporting workstreams, see Workstreams in a large migration in the *Foundation playbook for AWS large migrations*.

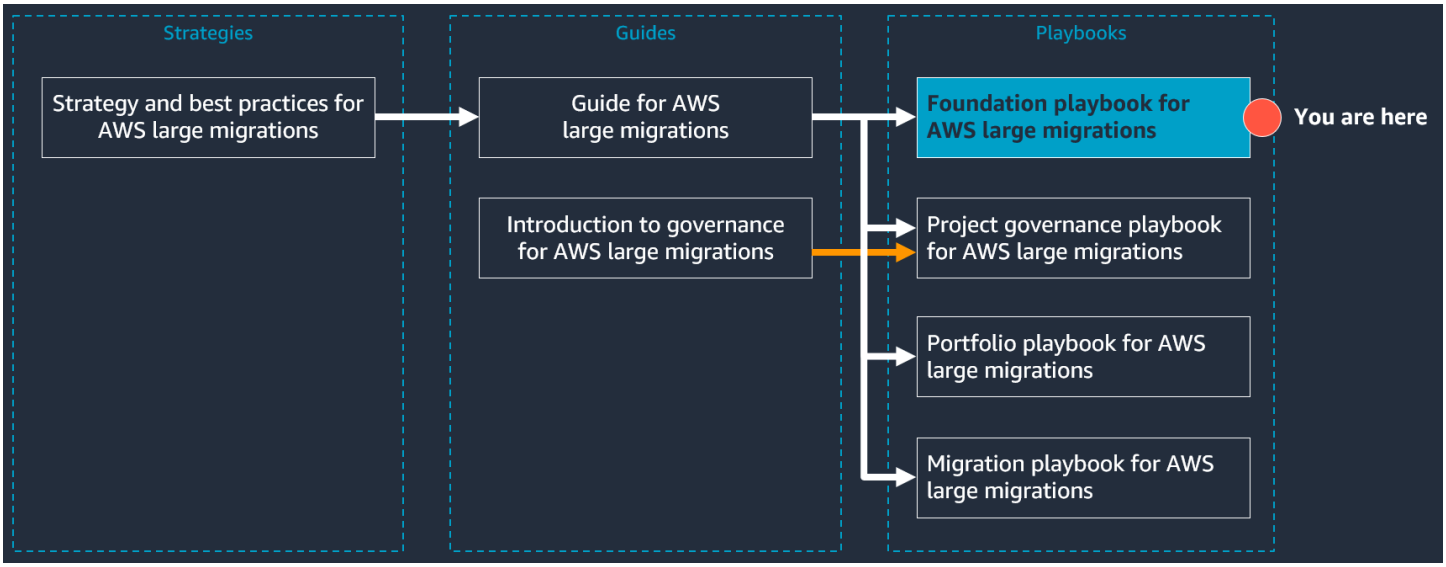
The purpose of this playbook is to prepare the platform foundation and people foundation to support a large-scale migration effort. Both of these foundations are critical to the success of large migrations. This guide consists of the following sections:

- **People foundation** – In this section, you define the workstreams in your large migration project and build a responsible, accountable, consulted, informed (RACI) matrix for each high-level task. It also includes recommendations for establishing a Cloud Enablement Engine (CEE). This section also contains training resources and helps you build a training dashboard for your large migration.
- **Platform foundation** – In this section, you review technology considerations for the on-premises and AWS Cloud environments, such as infrastructure, operations, security. You make key decisions in these categories, which you record as migration principles.

Guidance for large migrations

Migrating 300 or more servers is considered a large migration. The people, process, and technology challenges of a large migration project are typically new to most enterprises. This document is part of an AWS Prescriptive Guidance series about large migrations to the AWS Cloud. This series is designed to help you apply the correct strategy and best practices from the outset, to streamline your journey to the cloud.

The following figure shows the other documents in this series. Review the strategy first, then the guides, and then proceed to the playbooks. To access the complete series, see Large migrations to the AWS Cloud.



About the tools and templates

In this playbook, you create the following tools, which you use to prepare the platform and people:

- Migration principles
- RACI matrices
- Dashboard for training

We recommend using the foundation playbook templates included in this playbook and then customizing them for your portfolio, processes, and environment. The instructions in this playbook tell you when and how to customize each of these templates. This playbook includes the following templates:

- **Dashboard template for training** – This dashboard template helps you build a training plan for each workstream and track each individual's progress toward completing the required training.
- **Data replication calculator** – This workbook helps you estimate the amount of time needed to complete data replication.
- **Migration principles template** – This template helps you record the key infrastructure, operations, and security decisions that you need to make when preparing your platform.

- **RACI template** – This template helps you build a high-level and detailed RACI matrices that outline the roles and responsibilities of your large migration project.

People foundation

This section focuses on preparing the people and processes involved in your project for the activities in each stage of the large migration. To build the people foundation, you need to define the workstreams in your project, organize individuals into functional teams, confirm that that roles and responsibilities are well understood, and complete training.

This section consists of the following topics:

Workstreams in a large migration

Large migration projects typically consist of multiple workstreams, and each workstream has a clear scope of tasks. Each workstream is independent but also supports the other workstreams to accomplish the same goal – migrate servers at scale. This section discusses the standard core workstreams for large migrations as well as common supporting workstreams.

Core workstreams

Core workstreams are needed for every large migration, regardless of company size or segment. The following is an overview of the primary roles of each core workstream:

- **Foundation workstream** – This workstream is focused on preparing the people and platform for the large migration.
- **Project governance workstream** – This workstream manages the overall migration project, facilitates communication, and focuses on completing the project within budget and on time.
- **Portfolio workstream** – The teams in this workstream collect metadata to support the migration, prioritize applications, and perform wave planning.
- **Migration workstream** – Using the wave plan and collected metadata from the portfolio workstream, the teams in this workstream migrate and cutover the applications and servers.

Information and activities flow from upstream to downstream in a large migration, as shown in the following table. Information comes from the upstream foundation and project governance workstreams, through the portfolio workstream, and into the migration workstream. For example, the portfolio workstream is upstream of the migration workstream because the portfolio workstream prepares the metadata and wave plan that the migration workstream uses to migrate and cutover the applications and servers. Adding additional, supporting workstreams in your large migration project might change the flow of information and activities through the core workstreams.

> **Important**
>
> You need to assign a project-level technical leader for your large migration project. This role is not part of any individual workstream but has the total responsibility of all workstreams. This individual oversees all workstreams to make sure they work together and stay focused on the project-level goals.

Core workstream name	Upstream workstreams	Downstream workstreams
Foundation	—	Migration Portfolio
Project governance	—	Migration Portfolio
Portfolio	Foundation Project governance	Migration
Migration	Foundation Project governance Portfolio	—

The following are the primary functions of each core workstream in the phases of a large migration. The playbooks in this document series are structured to help you navigate the tasks for each workstream in the appropriate phase and stage.

		Foundation	Project governance	Portfolio	Migration
Phase 1: Assess		—	—	—	—
Phase 2: Mobilize		You might have designed the AWS landing zone or workstreams in this phase.	You might have designed a project management process in this phase.	You might have completed an initial portfolio assessment and discovery in this phase.	You might have completed a pilot migration in this phase.
Phase 3: Migrate	**Stage 1: Initialize**	Establish workstreams and review landing zone design. Prepare for change. Formalize migration principles, teams, and RACI matrix. Complete training.	Develop project management processes and communication and meeting plans.	Develop metadata, wave planning, and application prioritization runbooks.	Develop migration runbooks.
	Stage 2: Impement	—	Facilitate and communicate the status of waves and	Collect metadata for the migration, prioritiz	Migrate and cutover waves, and iterate the

		Foundation	Project governance	Portfolio	Migration
			the overall migration project.	e applicati ons, and plan waves.	runbooks to increase velocity.

The following sections describe each of the core workstreams in more detail, including common tasks for each workstream, the expected outcome of each workstream, and the skills required in each workstream. It is not required that each individual in the workstream have every skill. A workstream consists of one more cross-functional teams, so each person contributes different skills. But as a team, they should have all the skills listed.

Foundation workstream

The foundation workstream consists of two categories: platform foundation and people foundation. Building the platform foundation helps confirm that both the AWS and the on-premises infrastructures are ready to support the large migration. Building a people foundation prepares and trains the project teams for the migration and sets up all workstreams.

Common tasks	• Build and validate the AWS landing zone • Prepare the on-premises infrastructure to support the migration, such as making networking or firewall changes, permissions changes, or Active Directory changes • Set up the project core workstreams and supporting workstreams • Set up the training plan for the team • Build the RACI matrices with project managers
Expected outcome	• Source and target platforms are prepared for the large migration. • People are ready to support the large migration

	• All workstreams are set up.
Required skills	• Deep knowledge of on-premises data centers, including servers, storage, and networking • Experience with the AWS Cloud and knowledge of AWS compute services, including landing zones and AWS Control Tower • Experience with large data center or cloud migrations • Experience building a training plan • Experience building a cross-functional team

Project governance workstream

The project governance workstream manages the overall migration project and is responsible for delivering the project on budget and on time.

Common tasks	• Kick off the project • Set up the governance model • Set up the Cloud Enablement Engine (CEE) • Set up the communication plan • Set up the escalation plan • Build RACI matrices • Set up the project management framework • Set up status reporting and project tracking • Set up risk and issue tracking • Continuously manage the project by using the predefined processes and tools
Expected outcome	• Ensure that every workstream is able to complete their tasks on time • Ensure collaboration across workstreams

	• Ensure that the project achieves the defined business outcomes • Deliver the project on budget and on time
Required skills	• Experience with common project management methodologies, such as waterfall, agile, Kanban, and scrum • Experience with common project management tools, such as Jira, Microsoft Project, and Confluence • Experience with large migration project management

Portfolio workstream

The portfolio workstream manages all of the migration discovery activities, collects metadata, prioritizes applications, and creates a wave plan to support the migration workstream.

Common tasks	• Validate the migration strategies and patterns • Complete portfolio discovery by using discovery tools and configuration management database (CMDB) • Define the required metadata, collection processes, and storage location • Prioritize applications • Perform application deep dives, including dependency analysis and target state design • Perform wave planning • Collect migration metadata
Expected outcome	• Continuously create wave plans and collect migration metadata, and then hand off to the migration workstream

Required skills	• Deep knowledge of on-premises CMDB, data repositories, and content management tools • Experience with common portfolio discovery tools, such as AWS Application Discovery Service, Flexera One, and modelizeIT • Experience with portfolio assessment and application prioritization • Experience with application deep dives and application owner interviews • Experience with application designs for the AWS Cloud • Experience with wave planning for large migrations • Experience with automation, including shell scripting, Python, and Microsoft PowerShell

Migration workstream

The migration workstream manages the migration implementation-related activities, including data replication and cutover. Because the migration team performs the migration and cutover, a common misconception is that the migration workstream does everything in a large migration project. However, the migration workstream is dependent on other workstreams to build the foundation and provide portfolio data to support the migration.

Tip

The migration workstream is generally the largest workstream in a large migration project. Depending on the size and strategy of your project, consider dividing this workstream into multiple sub-workstreams. For example:

- Rehost migration workstream
- Replatform migration workstream
- Refactor migration workstream
- Relocate migration workstream

- Migration workstream for a specialized workload, such as SAP or databases

Common tasks	• Validate the migration wave plans • Build the migration runbooks • Use AWS migration services to transfer data, such as AWS Application Migration Service (AWS MGN), AWS Database Migration Service (AWS DMS), and AWS DataSync • Install and uninstall software on source and target servers as needed support the migration • Write automation scripts to automate migration activities • Launch target AWS environments, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, for testing or cutover • Work with change management team for changes and cutovers • Perform migration cutover • Support application owners during applicati on testing • If cutover fails, help roll back the server
Expected outcome	• Complete migration cutover and application go-live in target AWS accounts
Required skills	• Deep knowledge of on-premises data centers, including servers, storage, and networking • Experience with the AWS Cloud and knowledge of AWS compute services, including landing zone and AWS Control Tower

	• Experience with AWS migration services, including Application Migration Service, AWS DMS, DataSync, and AWS Snow Family • Experience with large data center or cloud migrations and cutovers • Experience with automation, including shell scripting, Python, and Microsoft PowerShell

Supporting workstreams

Supporting workstreams support the core workstreams. These workstreams are optional, and you might decide to use them based on your use case and the current stage of your migration. The following are some common supporting workstreams that you might want to include in your large migration project:

- **Security and compliance workstream** – This workstream defines and builds the security standards for the target AWS infrastructure and supports migrations.
- **Cloud operations (Cloud Ops) workstream** – This workstream manages applications after cutover, when the hypercare period is complete.
- **Application testing workstream** – This workstream performs application testing before and during the cutover.
- **Specialized workload migration workstream** – This workstream supports migrations for specific, specialized workloads, such as SAP or databases.

You might not need a dedicated workstream for these activities. It is common to have an individual or set of individuals be responsible for these activities and then embed those individuals in one of the core workstreams. For example, every large migration requires a security and compliance person because you need to make sure your target infrastructure is secure and compliant. However, security and compliance assessments and decisions are typically performed early in the migration, most commonly in the mobilize phase. If you have already completed this, you do not need a dedicated workstream to repeat the same tasks. However, it is recommended that you embed a security and compliance person in the migration workstream in order to support the migration activities.

When you add supporting workstreams, it modifies the flow of information and activities through the core workstreams. The following table is an example of how adding workstreams changes this flow. Your supporting workstreams might differ from the examples in this table.

Workstream name	Type	Upstream workstreams	Downstream workstreams
Migration	Core	Foundation Project governance Portfolio Security and compliance	Application testing Cloud operations
Portfolio	Core	Foundation Project governance Security and compliance	Migration
Project governance	Core	—	Migration Portfolio
Foundation	Core	—	Migration Portfolio Cloud operations
Security and compliance	Supporting	—	Migration Portfolio
Cloud operation	Supporting	Migration Application testing	—

Workstream name	Type	Upstream workstreams	Downstream workstreams
		Foundation	
Application testing	Supporting	Migration	Cloud operations
Specialized workload migration	Supporting	Foundation Project governance Portfolio Security and compliance	Application testing Cloud operations

Security and compliance workstream

The security and compliance workstream defines and builds the security standards for AWS infrastructure and supports migrations. Using the standards established by this workstream, application owners typically define the security and compliance requirements for each application. You might decide to have the security and compliance workstream review and approve the requirements for some or all applications.

Common tasks	• Define the security requirements for the AWS landing zone, such as centralized logging, encryption, AWS Identity and Access Management (IAM) policies, and Active Directory integration • Define the compliance requirements, such as HIPAA, personally identifiable informati on (PII), Service Organization Control (SOC), and Federal Risk and Authorization Management Program (FedRAMP) • Define the security requirements for the migration, such as firewall, security group, and IAM role requirements

	• Manage changes for security-related tasks, such as changes to firewalls, security groups, and permissions
Expected outcome	• Complete migration cutover and application go-live in target AWS accounts
Required skills	• Deep knowledge of on-premises data centers, including servers, storage, and networking • Deep knowledge of the specialized workload in scope • Experience with the AWS Cloud and knowledge of AWS compute services, including landing zones and AWS Control Tower • Experience with AWS migration tools, including Application Migration Service, AWS DMS, DataSync, and AWS Snow Family • Experience with large data center or cloud migrations and cutovers

Cloud operations workstream

The cloud operations workstream supports the applications after migration cutover. Sometimes cloud operations is in a separate workstream with dedicated resources, but most commonly, these resources come from existing IT operations teams. In that case, no dedicated workstream is required.

Common tasks	• Monitor and back up the migrated servers and applications • Manage the business-as-usual service requests from the application teams, such as increasing the disk size or changing instance types

	• Resolve any application issues and outages as needed • Manage the patching policies and schedules • Manage the maintenance tasks and requests
Expected outcome	• Migrated servers and applications are running smoothly on AWS • Respond to service requests from users and resolve any issues
Required skills	• Deep understanding of how the on-premis es data center currently operates • Experience with common AWS operations services, such as Amazon CloudWatch, AWS Config, AWS CloudTrail, AWS Backup, AWS Support • Experience with troubleshooting, and understands the SLA • Experience with supporting large migrations

Application testing workstream

The application testing workstream supports application testing before and during the cutover. This workstream is more common in projects where system integrators manage the data centers because the application owners don't have sufficient knowledge to perform the application tests. In most cases, the application owner performs these activities, and a dedicated application testing workstream is not required.

Common tasks	• Perform application testing before the cutover • Perform application testing during the cutover • Make application changes as needed to work in the new environment

	• Make a go or no-go decision for applications based on testing results during cutover
Expected outcome	• Complete application testing on time during cutover • Make application changes as needed to support the target environment
Required skills	• Deep knowledge of the applications and how they operate on premises • Experience with the AWS Cloud, especially the target AWS services • Experience with large migrations

Migration workstream for a specialized workload

You can create a migration workstream that is dedicated to specialized workloads. Generally, you can build standard migration patterns and runbooks to migrate servers and applications at scale, and these are managed by the migration workstream. However, in some cases, certain applications require special migration processes. For example, you might need a special process in order to migrate Hadoop workloads, SAP HANA databases, or mission-critical applications that cannot tolerate the standard amount of down time. For more information about specialized workloads, see *MAP specialized workloads* at AWS Migration Acceleration Program.

Common tasks	• Validate the migration wave plans • Build migration runbooks • Use migration tools or native application tools to transfer data • Launch target AWS environments, such as EC2 instances, for testing or cutover • Work with the change management team for changes and cutovers • Perform migration cutover • Support application owners during applicati on testing

	• If cutover fails, roll back the application or server
Expected outcome	• Complete migration cutover and application go-live in target AWS accounts
Required skills	• Deep knowledge of on-premises data centers, including servers, storage, and networking • Deep knowledge of the specialized workload in scope • Experience with the AWS Cloud and knowledge of AWS compute services, including landing zones and AWS Control Tower • Experience with AWS migration tools, including Application Migration Service, AWS DMS, DataSync, and AWS Snow Family • Experience with large data center or cloud migrations and cutovers • Experience with migrating the specialized workload

Roles

The following are the common roles in a large migration project. Because these roles might go by another title in your organization, a brief description of each role is provided. If a role is not available in your organization, you might investigate whether other resources in your organization can perform this role or seek outside support in the form of a consultant.

General role	Alternate titles	Workstreams	Characteristics
Application owner	Application architect , application project	All	Should have in-depth knowledge of their applications

General role	Alternate titles	Workstreams	Characteristics
	coordinator, applicati on project manager		
Automation engineer	DevOps engineer	Migration, portfolio	Should have experience and in-depth knowledge of how to build automation scripts
Cloud architect	Cloud engineer, migration consultan t, architecture lead, cloud infrastructure architect	Migration, foundatio n, portfolio	Should have experience and in-depth knowledge of how to design the AWS Cloud infrastructure, how to perform portfolio assessment and wave planning, and how to use migration tools to migrate workloads to the AWS Cloud
Cloud operations lead	Migration technical support, cloud operations workstrea m lead	Cloud operations	Should have experience and in-depth knowledge of how to operate workloads in the AWS Cloud
Communication lead	Business unit liaison	Project governance	Should have relations hip to the business unit and manage all communications

General role	Alternate titles	Workstreams	Characteristics
Executive leadership	Project sponsor	All	Should have clear vision of the migration project
Migration lead	Migration support lead, migration technical product owner, migration workstream lead	Migration	Should have experience and in-depth knowledge of all migration patterns and how to use migration tools to migrate workloads to the AWS Cloud
Portfolio lead	Discovery lead, wave planning lead, portfolio workstream lead	Portfolio	Should have experience and in-depth knowledge of how to perform discovery, portfolio assessment, and wave planning
Project manager	Program manager, project coordinator, Scrum master, project delivery lead, program delivery lead, large migration manager	Project governance	Should have experience and in-depth knowledge of how to manage a large migration project and how to use agile methodologies

General role	Alternate titles	Workstreams	Characteristics
Project technical lead	Engineering lead, technical lead, chief architect	All	Should have experience and in-depth knowledge of all workstreams and how to deliver a migration project from start to finish. Responsible for the entire project outcome across all workstreams
System integrator	Global system integrator	All	Varies, depending on the workstrea m. Should have in-depth knowledge of workstream-level activities, such as portfolio assessment or server migration
Testing lead	Testing specialist, application testing workstream lead	Application testing	Should have experience and in-depth knowledge of how to perform application testing in the AWS Cloud

Team organization and composition

This section includes the following topics:

- Best practices for team organization and composition
- Creating RACI matrices
- Cloud Enablement Engine (CEE)

Best practices for team organization and composition

Team composition in a large migration varies by organization and changes over the course of the project. The following are best practices that are common for all large migration projects:

- **Identify a single-threaded technical leader at the project level and avoid silos** – Large migration projects often have multiple workstreams and teams, each team has different tasks and expected outcomes. A single-threaded leader at the project level is important because this leader makes sure all workstreams work together and stay connected. This helps prevent silos and boundaries. For example, the portfolio workstream needs to continuously send the migration metadata to the migration workstream to support the migration activities. Without a complete understanding of the required migration metadata, the output of the portfolio workstream might not work as an input for the migration workstream. A single-threaded leader helps coordinate the inputs and outputs of each workstream to help the migration run efficiently.
- **Align all workstream-level outcomes with the project-level business outcomes** – Project-level business outcomes should be communicated to all workstream leaders before the migration starts. Each workstream leader must understand the role of their workstream and design their processes to support the project-level business outcomes. For example, if a project-level business outcome is exiting a data center in the next 12 months and speed is the most important factor, the workstream leaders should do the following:
  - All workstreams should prioritize rehost migrations, reduce the number of manual tasks, and add automation to improve the velocity.
  - The portfolio workstream should define standardized patterns and limit customizable patterns to reduce the amount of time required to design the target environment.
- **Design workstreams based on project scope and stage** – Every migration project is different, and one size does not fit all. We recommend having four core workstreams for all large migration projects: migration workstream, portfolio workstream, project governance workstream, and foundation workstream. You might decide to create additional, supporting workstreams depending on your use case. For more information about workstreams, see Workstreams in a large migration. For example, if you have not yet designed the security guardrails in the mobilize phase, you need to create a security and compliance workstream that can define the security and compliance requirements before you start migrating. For more information about building the security guardrails in the mobilize phase, see Security, risk and compliance in *Mobilize your organization to accelerate large-scale migrations*.

- **Get the application team involved before the migration** – A large migration is never just an IT infrastructure project – it changes the operating model for your business. Involving the application team early and embedding the application owners into your large migration workstreams is critical to the success of large migration project. For example, during portfolio assessment, schedule your meetings early with application owners so that they can participate in the deep dive and help design their application's target state on AWS.
- **Determine the team size based on workstreams and business outcomes** – Your expected business outcomes and migration strategies drive the size of each team, which is composed of smaller units called pods. In each workstream, you define teams for each migration strategy and then separate those teams into pods. For example, if rehost is your primary migration strategy, then you should have a rehost migration team that is composed of pods that contain 3–5 people. When operating at peak velocity, a pod of 4–5 people on a migration team can typically rehost up to 50 servers per week. This is approximately 200 servers per month or 2,500 servers per year. If your target is to rehost 100 servers per week, you should create two pods of 4–5 people within the rehost migration team. If you are targeting less than 50 per week, you can reduce the size of the migration pod to 3 people. Replatform migrations usually cost more than rehost, and the same size pod can migrate up to 20 servers per week. The portfolio workstream is usually half the size of the migration workstream. You create additional teams and pods in each workstream to support each migration strategy. These recommendations assume that your migration resources are skilled and do not require significant training. The following table is an example of how you would divide the migration and portfolio workstreams into teams and pods for the rehost and replatform migration strategies. The following example assumes that you need to migrate 120 servers per week (100 rehost + 20 replatform) or 6,000 servers per year. This example is the maximum velocity. We recommend that you plan for additional resources in order to help prevent delays.

Workstream	Team	Pod	Resources
Migration workstrea m	Rehost migration team	Rehost migration pod 1	4–5 people
		Rehost migration pod 2	4–5 people
	Replatform migration team	Replatform migration pod	4–5 people

Workstream	Team	Pod	Resources
Portfolio workstream	Portfolio team	Portfolio pod 1	3–4 people
		Portfolio pod 1	3–4 people

- **Build a governance model in the early stage** – A large migration typically involves many people, including people from your own company, third-party software vendors, system integrators, or external consultants. Your project might include representatives from AWS, such as your account team, support engineers, or experts from AWS Professional Services. Your delivery model varies depending on your project scope and who you work with to deliver the project. For example, your project might include AWS or a system integrator, or you might include both. It is important to build a governance model early and create a RACI matrix that clearly defines the roles and responsibilities. As a recommendation, we also recommend creating a Cloud Enablement Engine (CEE), also known as *Cloud Center of Excellence*, in your organization and including representation from all parties. The key purpose of the CEE is to transform the organization from an on-premises operating model to a cloud-operating model. This centralized team is critical to the success of a large migration because it manages relationships, makes key decisions, and handles escalations throughout the project. The CEE is discussed in more detail later in this guide.

Creating RACI matrices

A large migration project typically involves a lot of people, so building a governance model is important to manage the project. One of the key components of a governance model is a *RACI matrix*, which is used to define the roles and responsibilities for all parties involved in the large migration. The name RACI matrix is derived from the four responsibility types defined in the matrix:

- **Responsible (R)** – This role is responsible for performing the work to complete the task.
- **Accountable (A)** – This role is held accountable for making sure the task is completed. This role is also responsible for ensuring the prerequisites are met and delegating the task to those who are responsible.
- **Consulted (C)** – This role should be consulted for opinions or expertise on the task. Depending on the task, this responsibility type might not be required.
- **Informed (I)** – This role should be kept up to date on the progress of the task and notified when the task is completed.

Because of the complexity of a large migration, we do not recommend using a single RACI matrix to document every task in the large migration. A multi-layer RACI matrix is a much more accessible approach. You start by building a high-level RACI matrix, and then you add more details to each section to build a detailed matrix. Building a detailed RACI matrix is not a one-off approach. You need to build new matrices or add more details to the existing ones as you progress through the portfolio and discover more migration strategies and patterns.

In the foundation playbook templates, you can use the RACI template (Microsoft Excel format) as a starting point for building your own high-level and detailed RACI matrices. This template includes two examples of detailed RACI matrices, one for a rehost migration and another for a replatform migration. The tasks in these examples are included for sample purposes only, and you should customize these examples based on your use case.

Build a high-level RACI matrix

Before you start building a high-level RACI matrix, you need to have the following information ready:

- **Who are the high-level parties involved in this migration?**Identify any partners or consultants that will be involved in this project, such as AWS Professional services or system integrators. Consider whether any part of your current IT infrastructure is managed by an external partner. The following are examples of high-level parties:
  - Your organization
  - AWS Professional Services
  - System integrators
- **What are the workstreams in your migration?** For more information, see Workstreams in a large migration. At a minimum, you should have the four core workstreams, and you can add support workstreams as needed for your project.
- **What are the high-level tasks in your migration?** Create a list of the high-level tasks in your migration. The following are examples of high-level tasks:
  - Build an AWS landing zone
  - Perform portfolio assessment and collect migration metadata
  - Perform a rehost, replatform, or relocate migration
  - Perform application testing and cutover
  - Perform project management and governance tasks

Do the following to build your high-level RACI matrix:

1. In the foundation playbook templates, open the *RACI template* (Microsoft Excel format).
2. On the **High-level RACI** tab, in the first row, enter your organization name and any partners that you identified.
3. In the first column, enter the high-level tasks and workstreams that you identified.
4. In the matrix, determine which parties are responsible for each task as follows:
    - If a party is **responsible** for completing the task, enter an **R**.
    - If a party is **accountable** for the task, enter an **A**.
    - If a party should be **consulted** about the task, enter a **C**.
    - If a party should be **informed** about the task, enter an **I**.

The following table is an example of a high-level RACI matrix.

Task	Your organization	Partner A	Partner B	Partner C
Build an AWS landing zone	R/C	A	I	I
Perform portfolio assessment and wave planning	R/C	A	I	I
Perform rehost migration activities	C	C	R/A	I
Perform replatform migration activities	C	C	I	R/A

Task	Your organization	Partner A	Partner B	Partner C
Project management and governance	R/C	A	I	I
Application changes and testing	C	R/A	C	C
Cloud operations	I	C	R/A	I

Build the detailed RACI matrices

After creating the high-level RACI matrix, the next step is to create a detailed RACI for each high-level task and further refine the tasks, parties, and ownership. Before you start building detailed matrices, you need to have the following information ready:

- **What are the detailed tasks in your migration?** After you have prepared the runbooks and task lists for your large migration project, the processes and details in these runbooks form the detailed layer of your RACI matrix. For example, for a rehost migration, detailed tasks might include installing a replication agent, verifying replication, and launching test instances for boot-up testing. If you haven't done so already, follow the instructions in the following playbooks to create these documents:
  - Portfolio playbook for AWS large migrations
  - Migration playbook for AWS large migrations
- **What smaller teams make up each workstream and each high-level party?** For example, teams in your organization might include an application team, infrastructure team, operations team, networking team, or a project management office.

Do the following to build a detailed RACI matrix:

1. Open your high-level RACI matrix.
2. Create a copy of the **Detailed RACI (template)** spreadsheet.

3. Name the copied spreadsheet for a high-level task that you identified in Build a high-level RACI matrix.
4. In the first row, enter the names of the teams involved in this high-level task.
5. In the first column, enter the detailed tasks that you identified for this high-level task. You can group the detailed tasks into logical sequential groups, which helps readers navigate the matrix.
6. In the matrix, determine which teams are responsible for each task as follows:
   - If a team is **responsible** for completing the task, enter an *R*.
   - If a team is **accountable** for completing the task, enter an *A*.
   - If a team should be **consulted** about the task, enter a *C*.
   - If a team should be **informed** about the task, enter an *I*.
7. For each detailed task, confirm that only one team is responsible and only one team is accountable. If multiple teams are responsible or accountable, this can indicate that the task is not clearly defined or doesn't have clear ownership.
8. Share the detailed RACI matrix with the identified teams and confirm that all teams are familiar with their roles and responsibilities.
9. Repeat this process for each high-level task that you identified in Build a high-level RACI matrix.

For examples of detailed RACI matrices, see the **Rehost RACI** and **Replatform RACI** spreadsheets in the *RACI template*, available in the foundation playbook attachments.

Cloud Enablement Engine (CEE)

Best practices for using a CEE

The purpose of a CEE is transforming an IT organization from an on-premises operating model to a cloud-operating model, and it is responsible for guiding the organization through the organizational and cultural changes. As a best practice, it is recommended that you establish a CEE for your large migration. The well-defined foundational processes and guard rails of a CEE can help you achieve the scale and velocity required for large migrations. For information about setting up a CEE, see Cloud Enablement Engine: A Practical Guide. The following are additional recommendations and best practices for establishing a CEE for a large migration project:

- The CEE team should be comprised of cross-functional leaders with the following qualities:

  - Have deep institutional knowledge
  - Have strong, long-standing internal relationships
  - Have a vested interest in the progress and success in the large migration
  - Are curious and want to learn
  - Are primarily or solely focused on the migration
- The CEE team should be a mix of people who have worked together previously and newcomers who can provide fresh insights.
- The CEE team should have strong executive support and alignment on the migration objectives.
- Make sure the goals of the CEE team are specific to the large migration.
- Conduct regular, open meetings that provide opportunities for questions and answers, demonstrate cloud services and architectures, and share updates on successful migrations and other wins.
- The CEE team should be empowered to make critical decisions about the large migration project.

Typical CEE roles and responsibilities for large migrations

The following table provides roles in a large-migration CEE team, and it describes the typical tasks and responsibilities for each role. The actual composition of your team and their responsibilities can vary based your use case, scope, and business objective.

Roles	Tasks and responsibilities
Executive sponsor	• Managing escalations • Aligning the organization tightly around the objectives and criticality of the migration. • Serving as the voice of authority
Enterprise architect or project-level technical lead	• Identifying and documenting the reference architecture for known workload types • Designing and building migration processes for the entire project, across all workstreams • Serving as the single-threaded technical leader who makes sure all workstreams are

Roles	Tasks and responsibilities
	collaborating and working to deliver the same business-level objectives • Strong institutional knowledge of major applications and common architectures
Project management office lead	• Managing timelines, onboarding, training, documentation, reporting, communication, and resource governance • Managing resourcing and training • Managing migration-related town halls
Migration lead	• Designing migration processes and tools • Designing migration strategies and automation • Overseeing migration cutovers and achieving the target velocity
Portfolio lead	• Designing portfolio assessment and wave planning processes and tools • Designing portfolio discovery and data collection processes • Overseeing the continuous supply of migration metadata and wave plans
Cloud operations lead	• Designing the operating model for running workloads on AWS • Designing strategies for monitoring, incident response, tagging, business continuity, and disaster recovery strategies

Roles	Tasks and responsibilities
Application team leader	• Managing the relationship with individual application owners • Managing migration planning and cutovers for their applications • Managing application changes, testing, and approvals
Network and infrastructure lead	• Designing the AWS landing zone for target accounts • Designing network connectivity and infrastructure • Designing and deploying security groups • Managing infrastructure and networking changes to support the large migration
Licensing lead	• Identifying all commercial off-the-shelf (COTS) and enterprise applications and working with the migration team and application team to plan migration strategie s around licensing
Security and compliance lead	• Designing authentication and authorization for the large migration, including Active Directory, single sign-on, and IAM policies • Designing network security, including on-premises firewalls, and managing vulnerabi lities • Designing compliance requirements for in-scope workloads

Training and skills required for large migrations

The people involved in the large migration are a critical resource, and it is equally as important to prepare them for the migration as it is to prepare the landing zone or workstreams. This section is dedicated to training the people in your project, ensuring that your teams have the skills necessary to perform a large migration. While some skills are common and required for many roles, other skills are more specialized and require thoughtful recruitment or training. By ensuring individuals are properly trained for their roles before the migration starts, the workstreams can operate efficiently, and you can quickly ramp up the migration to the target velocity.

Training is divided into levels: prerequisites, fundamentals, and advanced. Every person in your large migration project should complete the prerequisite-level training, which reviews basic information about the AWS Cloud and migration concepts. For fundamentals and advanced levels, you use a training plan to assign a training level to each workstream. You then use a training tracking tool to record each individual's progress toward completing the required trainings in their workstream. It is important to note that we recommend training based on workstreams rather than roles and job titles because roles can vary significantly between organizations.

Each of the following sections lists and describes the training resources recommended for the level:

- Large migration training – Prerequisites
- Large migration training – Fundamentals
- Large migration training – Advanced

Prerequisites

At a minimum, the resources in every workstream should have foundational understanding of infrastructure, networking, and core AWS services, AWS Cloud Adoption Framework (AWS CAF) and the AWS Well-Architected Framework. The following are recommended for this training level:

- **AWS Technical Essentials** – This foundational training module provides an overview of AWS services and cloud technology, such as virtual private clouds (VPCs), Amazon Elastic Compute Cloud (Amazon EC2), Availability Zones, and AWS Regions.
- **Foundational training for infrastructure, networking, and data centers** – Provide foundational training about infrastructure and networking, such as Transmission Control Protocol (TCP), Internet Protocol (IP), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and load balancers. Provide training about data center technologies, such as the

software development lifecycle (SDLC) and IT service management (ITSM). Training requirements in this category vary based on your environment and use case, and many training resources are available. We recommend working with your IT department to identify technology-level training that is appropriate for all personnel in your large migration project

- **Organizational processes** – Provide training for any processes that are specific to your organization, such as change management processes. You must understand the deadlines, approvals, and formal documents required to make changes in your organization, such as firewall and domain changes. Determine whether external partners or consultants need this training in order to support your project.
- **Shared Responsibility Model** – If you are working with AWS Professional Services, this webpage describes how you will share roles and responsibilities with AWS.
- **An Overview of the AWS Cloud Adoption Framework (AWS CAF)** – This whitepaper helps you understand the goals of AWS CAF, the AWS CAF perspective, and the stakeholders involved.

Fundamentals

This section provides an overview of the processes, tools, and guidelines required to successfully complete a large migration. The following are recommended for this training level:

- How to migrate This webpage helps you understand the three-phase migration process.
- About the migration strategies – This section of the *Guide for AWS large migrations* describes each of the migration strategies and common use cases for each in a large migration project.
- Migrating to AWS: A high level introduction – This course provides an overview of the key topics and target audience of the *Migrating to AWS* classroom course.
- Migrating to AWS – This course explains how to plan and migrate existing workloads to the AWS Cloud.
- Strategy and best practices for AWS large migrations – This strategy discusses best practices for large migrations and provides use cases from customers across various industries.
- Introduction to Database Migration – In this course, you learn how to migrate a production database by using the AWS Database Migration Service (AWS DMS) and AWS Schema Conversion Tool (AWS SCT).
- AWS DataSync Primer – The course helps you get started with DataSync, showing you how to move large amounts of data between on-premises storage and the AWS Cloud.
- Lift-and-Shift Application Workloads – This webpage helps you understand the basics the rehost, or lift-and-shift, migration strategy.

- AWS Application Migration Service (AWS MGN) – A Technical Introduction – This course introduces the Application Migration Service.
- Portfolio discovery and analysis for migration – This guide defines the approach for defining, collecting, and analyzing the data required to create a migration plan.
- Application portfolio assessment strategy for AWS Cloud migration – This AWS Prescriptive Guidance strategy helps you understand the key stages to successfully assess your application portfolio.
- AWS Cloud Migration Factory Solution – This webpage helps you understand what AWS Cloud Migration Factory Solution is.
- CloudEndure Migration Factory best practices (YouTube video) – This video and provides an overview of the AWS Cloud Migration Factory Solution and shares best practices for large-scale migrations. It includes information about how to coordinate and automate many manual migration processes.

Advanced training

Advanced training for large migrations dives deeper into the migration methodologies, tools, and best practices by providing workshops and training resources for the workstreams. The following are recommended for this training level:

- Cloud migration factory workshop – This technical workshop provides information about how to accelerate a large migration by using automation and the migration factory model.
- Guide for AWS large migrations – This guide contains high-level information about performing a large migration and introduces the large migration playbooks.
- Foundation playbook for AWS large migrations (this guide) – Use this playbook to train workstreams about preparing the platform foundation and people foundation for a large migration.
- Project governance playbook for AWS large migrations – This playbook provides step-by-step instructions for setting up the project governance framework and providing continuous governance throughout the migration.
- Portfolio playbook for AWS large migrations – This playbook provides step-by-step instructions to help you build your application prioritization runbook, metadata management runbook, and wave planning runbook.
- Migration playbook for AWS large migrations – This playbook provides step-by-step instructions for preparing migration runbooks for each migration pattern and preparing migration task lists.

Create your training dashboard

In the foundation playbook templates, you can use the *Dashboard template for training* (Microsoft Excel format) as a starting point for building your own training plan and tracking tool. You use a training plan to assign a training level to each workstream. You then use a training tracking tool to record each individual's progress toward completing the required trainings in their workstream.

1. On the **Prerequisites** spreadsheet, **Fundamentals** spreadsheet, and **Advanced** spreadsheet, add or remove workstreams as appropriate for your large migration project.
2. On the **Prerequisites** spreadsheet, update the training materials as needed for your use case. Define the appropriate training for infrastructure, networking, and data centers. We recommend working with your IT department to identify technology-level training that is appropriate for all personnel in your large migration project. This spreadsheet should contain the training materials that you want all members of every workstream to complete.
3. On the **Fundamentals** spreadsheet, update the training materials as needed for your use case, and identify which workstreams should train on each item listed.
4. On the **Advanced** spreadsheet, update the training materials as needed for your use case, and identify which workstreams should train on each item listed.
5. On the **Training** tracker spreadsheet, enter the name of each individual in your large migration project and their workstream.
6. As each individual completes the required training for their workstream, mark the training as **complete**.

Platform foundation

This section focuses on assessing the readiness of the on-premises infrastructure, preparing the AWS landing zone or reviewing the existing landing zone design, and identifying the migration tools needed. You review the common infrastructure, operations, and security questions that you should consider for building a platform. You document your answers and decisions as migration principles. As a result, you have a solid platform to achieve the scale and velocity required for large migrations.

This section includes the following topics:

- Landing zone considerations for a large migration
- On-premises considerations for a large migration
- Document your migration principles

Landing zone considerations for a large migration

A *landing zone* is a well-architected AWS environment that is scalable and secure. By establishing standards for the landing zone, such as defining the number of accounts and designing the subnets and security groups, you build a solid foundation. This foundation gives you the ability to enable, provision, and operate your environment for both business agility and governance at scale while accelerating your cloud adoption journey. For more information about landing zones and strategies for building them, see Setting up a secure and scalable multi-account AWS environment.

In addition to the standard business, operational, security and compliance considerations for your landing zone strategy, you must consider how to facilitate a large migration. You must design the landing zone to support existing, on-premises workloads during the migration and after, in cases where some workloads remain on premises. This guide provides additional landing zone considerations that affect the migration velocity and overall migration timeline.

Typically, landing zones are designed and deployed to support new workloads in the AWS Cloud. This is because organizations are adopting AWS before making the decision to migrate a large number of existing applications. The benefit of this approach is that the organization gains valuable knowledge and skills in AWS before the large migration, but it can also lead to conflicts between the various stakeholders. Some stakeholders might want to modernize the application during the migration because they want to take advantage of cloud-native features. However, the common goal of a large migration is to achieve maximum migration velocity and ease the

transition by migrating as many applications as possible without modifying the workload. You then modernize these applications after the migration is complete.

Some key factors of the landing zone that can affect your large migration program project are:

- Network bandwidth availability and management
- Account strategy for workload isolation and resource management
- Security and administrative controls for migrated workloads

This section reviews the infrastructure, operations, and security questions that you should consider when building your AWS landing zone. It also contains recommendations for how to design and deploy your landing zone to support a large migration project. As you answer the questions in this section, these decisions become migration principles, which you document according to the instructions in Document your decisions as large migration principles.

Infrastructure considerations

Have you considered?	Description	Actions
How much data will you migrate per day and per week?	The desired migration velocity dictates the type of network connection and network throughput requirements. It also can affect the wave planning selection criteria.	After you have completed the portfolio assessment, determine the total amount of storage needed for all migrated resources in the cloud. Use this value to calculate the amount of time required to migrate the data using the current network bandwidth. You might need to increase the bandwidth to meet the migration timeframes, or you might need to use alternati ves, such as AWS Snow Family solutions. In the foundatio n playbook templates, you can use the *Data replication*

Have you considered?	Description	Actions
		calculator (Microsoft Excel format) to calculate the required bandwidth for each migration wave.
What is the average write speed of the source servers in each wave?	The bandwidth required to transfer the replicated data is based on the write speed of the participating source servers. The amount of bandwidth required for server replication is the average write speed of your source servers multiplied by the number of servers in the largest wave.	During portfolio assessment, you need to determine the average number of data writes performed per by each server. In the foundation playbook templates, you can use the *Data replication calculator* (Microsoft Excel format) to understand the bandwidth required for migration traffic. The bandwidth required for migration traffic is in addition to the bandwidth used for normal business activity. After the migration is complete, you no longer need the additional bandwidth to support the migration activities.

Have you considered?	Description	Actions
Could additional network activities or existing infrastru cture limit or reduce the replication speed?	If the network bandwidth also supports other business functions, these activitie s can reduce the amount of bandwidth available for replicating servers during the migration.	Early in the project lifecycle , carefully assesses and calculate the network bandwidth required to support all business activitie s. Consider the bandwidth needed for normal business activities, server replication, and new migration-related activities, such as syncing on-premises file shares with data on AWS. Providers might have long lead times to increase the network capacity, and you might need to upgrade the existing on-premises infrastru cture. Consider whether any additional upgrades would be required as a consequen ce of upgrading the network infrastructure. Assessing bandwidth requirements early in the project provides time to make any necessary changes.

Have you considered?	Description	Actions
Does your current AWS subnet strategy meet the IP addressing requirements for migrating the on-premises workloads?	The number of servers and workload isolation requirements dictates the subnet strategy for your landing zone. Large migrations might require larger subnets than you expect. In a large migration, you group workloads in subnets similar to their setup in the on-premises infrastructure. To simplify the migration, larger, flatter subnet designs are preferred initially, and then, during modernization, you redesign the subnets as needed.	When the portfolio assessment has enough information about the infrastructure inventory, assess the on-premises network structure and incorporate it into the landing zone design as early as possible.
How many servers do you plan to replicate and migrate in parallel?	The size of the largest migration wave affects the subnet requirements and AWS service quotas.	Review the high-level migration plan, and use that to design your subnet. For example, if you have a plan to migrate 200 servers into one subnet, the Classless Inter-Domain Routing (CIDR) range for that subnet should have enough IP addresses to support the target number of servers. Also, increase the AWS service quota for each target account as needed.

Have you considered?	Description	Actions
Have you identified the security group strategies for your migration resources?	Security groups are used to manage the inbound and outbound traffic for AWS resources. It is important to design security groups early in order to avoid delaying the migration.	In your runbook for applicati on prioritization, review the migration strategies, and then design the security groups based on the migration strategies. For example, if the migration strategy is to rehost most of the workloads, consider a temporary, generic security group that supports migration cutover instead of refactoring the network and applying application-specific security groups.
Are there load balancers in use?	Typically, when migrating servers in an environment with load balancers, you need to assess the configuration of the load balancer and then migrate the load balancer. Migration options for the load balancer include using Elastic Load Balancing (ELB) or a partner appliance-based solution.	Assessment of load balancers needs to start early in the discovery phase in order to account for any custom configurations. In most environments, load balancer configurations are fairly standard, but some might have complex logic that determines whether you can migrate to ELB or a partner appliance-based solution.

Have you considered?	Description	Actions
Do any servers need to retain their source IP address?	The safest and easiest way to migrate servers to the cloud is to allocate new IP addresses to the migrated instances. In some situation s, you might need to keep the same IP address as the source server. For example, a legacy application might have a hardcoded IP address that no one knows how to change.	Keeping source IP addresses affects how you form move groups when wave planning. The most common approach is to migrate a whole subnet to AWS in a single move group because this makes routing and switching straight-forward at the network level. The following are key actions for keeping IP addresses: • Carefully assess cross subnet communications between servers. • Decide how you will switch routing of IP addresses for migrated servers. Common options include switching a whole subnet or deploying a network technology that manages static IP routing on a server-by-server basis.

Have you considered?	Description	Actions
How much latency is acceptable between the source and AWS?	It is common to start the migration with VPN links because they can be set up quickly and then transitio n to a direct connectio n established using AWS Direct Connect. VPN links generally have higher and more variable latency, which affects data throughput and, more importantly, application response times.	If you are using a high or variable latency connection type, review each applicati on's requirements and plan the migration waves accordingly. Plan to put applications that require low latency connections in later waves, when alternative connection types are available .

Operations considerations

Have you considered?	Description	Actions
Have you identified an AWS account strategy for your landing zone?	AWS best practices for a well-architected environment recommend that you should separate your resources and workloads into multiple AWS accounts. You can think of AWS accounts as isolated resource containers: they offer workload categorization and can reduce the scope of impact in the event of a disaster.	In your runbook for applicati on prioritization, review your selected migration strategies and use them to determine your account strategy. For example, if you want to migrate as quickly as possible and rehost is the most common migration strategy, fewer accounts is easier to manage. However, if your migration strategies require modernizing applicati ons and you need to separate business units for complianc e reasons, you should include one or more accounts for

Have you considered?	Description	Actions
		each business unit in your account strategy.
Do you need to switch monitoring tools during the migration? If so, is this part of the migration process, or does it occur before or after the migration?	Monitoring tools are critical for cloud operations. Your existing tools might not work in the cloud because of compatibility or licensing reasons. As part of the design, you need to decide which monitoring tools to use for the workload in the AWS Cloud.	Select a monitoring tool before starting the migration . Make sure the migration team incorporates instructi ons for setting up monitorin g in the migration patterns. We recommend building an automation script that replaces or reuses the monitoring tools, as needed.
Have you identified applicati on owners, and are they aware of any changes that must be made to the applicati on so that it functions properly in the cloud?	Large migration is a transformation rather than just an infrastructure project. Include applicati on owners early to support the migration. For example, application owners validate the wave plan, create test plans, and participate in the cutover.	Work with a project management office and Cloud Enablement Engine team to align with application team leaders and make sure that communication is clear across all application teams. For more information about communication and project transparency, see the Project governance playbook for AWS large migrations.

Have you considered?	Description	Actions
Have you selected a backup and recovery solution, and does it work with migrated workloads?	Backup and recovery tools are critical for cloud operations. Your existing tools might not work in the cloud because of compatibility or licensing reasons. As part of the design, you need to decide which backup and recovery tools to use for the workload in the AWS Cloud.	Select backup and recovery tools before starting the migration. Make sure the migration team incorporates instructions for setting up backup and recovery in the migration patterns. We recommend building an automation script that replaces or reuses the backup and recovery tools, as needed.
Have you identified all shared services and deployed them in the landing zone?	*Shared services* are services that support multiple applications, such as email, Active Directory, or shared database environments. You typically need to deploy shared services in the cloud before the migration so that migrated applications perform as expected.	Schedule a deep dive with the infrastructure team and application team leaders before completing the landing zone design. Review and confirm the list of shared services that you must deploy in the cloud before starting the migration. The most common shared services are Active Directory, network devices, Domain Name System (DNS), and infrastructure software.
Have you reviewed AWS service quotas for your target AWS Region and account?	Every AWS service has a service quota. Some of these quotas can be increased. It is important to review quotas before cutover. If insufficient resources are available, the cutover might fail.	Review the migration plan. For any target account that requires an increased service quota, request an increase. For more information and instructions, see AWS service quotas.

Have you considered?	Description	Actions
Do you need to upgrade your AWS Support plan?	AWS Enterprise support plan offers 24/7 phone support and faster response times than other plans. Because the cutover window is usually very short, having access to an experienced engineer to help resolve cutover issues can be critical to the success of a large migration.	Contact your AWS account team to discuss different support options and select the appropriate support plan for your large migration project.
Have you notified your AWS technical account manager (TAM) about your large migration plan?	The AWS Enterprise On-Ramp support team assigns a pool of Technical Account Managers (TAMs) who coordinate access to proactive programs, preventative programs, and AWS subject matter experts. Your TAMs can schedule availability of support resources as needed.	Notify your AWS technical account manager of your upcoming large migration project and share your migration plan. Your TAMs will make sure AWS support resources are available when needed. For example, your TAMs can schedule a support engineer during cutover, and the engineer can help mitigate technical issues and streamline the cutover.

Security considerations

Have you considered?	Description	Actions
Have you identified AWS Identity and Access Management (IAM) roles and policies for access management?	Manage identity and access for all members of your large migration project. By attaching IAM roles to the migrated resources and	Work with the migration team to identify the roles and responsibilities. Determine which roles can access which AWS account, and identify the

Have you considered?	Description	Actions
	defining access policies, you control who can access the migrated resources in the cloud.	level of access that each role has. Work with the security teams to validate that the IAM roles are correct for each target AWS resource.
Are there any compliance requirements for your workloads?	Workloads might have different compliance requirements, such as the Health Insurance Portability and Accountability Act (HIPAA) or payment card industry Data Security Standard (PCI DSS). You must identify these requirements before the migration and plan for how to meet them.	Work with compliance team and portfolio team to identify the compliance requirements for each application, and design your target AWS account accordingly. For example, you might need to migrate some workloads to AWS GovCloud (US) or to a specific AWS Region. We recommend that you document the compliance requirements for each application so that you can use this information later in the application prioritization and wave planning process.

Have you considered?	Description	Actions
Does your security team need to review and approve any tools or services that you plan to use during the migration?	A large migration project to the AWS Cloud uses many services, such as AWS Application Migration Service, AWS Database Migration Service (AWS DMS), AWS DataSync, and portfolio discovery tools (such as Flexera One). Some organizations require that all new tools and services are approved before use.	Work with the migration team to identify all of the tools, services, and applications that you expect to use in the migration. Work with the security team to review the company policies and approve these tools accordingly before the migration starts.

On-premises considerations for a large migration

On-premises infrastructure that supports your business operations must also be prepared for the large migration. By preparing the current infrastructure, you can help reduce the impact of the large migration to the business operations and application users.

This section reviews the infrastructure, operations, and security questions that you should consider when preparing your on-premises infrastructure for the large migration. As you answer the questions in this section, these decisions become *migration principles*, which you document according to the instructions in Document your decisions as large migration principles.

Infrastructure considerations

Have you considered?	Description	Actions
Have you designed the on-premises DNS and routers to support traffic to and from target AWS accounts?	Because of the large number of servers and target AWS accounts, it is important to confirm that different networking components are configured correctly	Review the design of routing tables, and make sure there are correct routes between the AWS accounts and on-premises data centers. Also, make sure the DNS server is

Have you considered?	Description	Actions
	to support the migration strategies and scale.	able to support DNS queries from both on-premises servers and AWS resources.
How will the migration team access both the on-premises and AWS environments?	The migration team needs to access the source and target servers to perform migration activities, such as install a replication agent on a source server or uninstall old software on a target server.	Review the existing authentication and authorization mechanisms and build a strategy to grant access. You can use an Active Directory group, IAM role, and Security Assertion Markup Language 2.0 (SAML 2.0) federation to allow single sign-on to the AWS account. We recommend creating a local admin user in case there are any authentication issues with Active Directory.
Are there any known congestion points in the current network configuration that would slow data throughput during the migration?	A large migration requires lots of bandwidth to replicate the data from on-premises data center to the cloud. Understanding any existing congestion points or limitations helps you better plan the migration.	Review the network configuration with the networking team to better understand the network path from the source machines to the target AWS accounts. Identify potential congestion points, such as a connection that is shared between the migration and production workloads.

Operations considerations

Have you considered?	Description	Actions
Do you have any scheduled blocked days, also known as *change freezes*, that could impact the migration?	A change freeze during migration can take critical resources and time away from an ongoing migration project.	Review the change management process with the operations team, and take blocked days into considera tion when you plan cutover windows.
Have you reserved change days for the migration?	Change management processes can be complex, and some organizations allow changes only in certain maintenance windows.	According to your change management process, schedule changes at least five waves in advance. This helps prevent delays
Have all of the servers in scope for the migration been recently rebooted?	System changes or uninstall ed patches might cause issues during the migration, which would necessitate long cutover windows or rolling back the server. The best practice is to confirm that the server has been recently rebooted on the target side before migrating.	Review the dates of the last server reboots. If a server has not been restarted within the last 90 days, schedule a restart before migrating the server.
How does the disaster recovery and business continuity plan work today, and has this been factored into the landing zone design?	Disaster recovery and business continuity plans are critical components of meeting the recovery time objective (RTO) and recovery point objective (RPO) of the application. You need to make sure these plans work for both your on-premises and	Review the existing disaster recovery and business continuity plans and make sure the plans work for your target AWS account. If not, design new plans before moving workload to the AWS Cloud.

Have you considered?	Description	Actions
	AWS workloads during the transition period.	

Security considerations

Have you considered?	Description	Actions
Have you created firewall rules to support the large migration?	Depending on the processes in your organization, it can take a long time to complete a change request for firewall configurations.	Review the existing firewall change process with security team, and design a strategy for large migration firewall changes accordingly. You might need to design a custom process for the large migration project, or you might need to submit changes early in the project. It is recommended that you consider using an AWS virtual private cloud (VPC) as an extension to your data center and avoid building firewall rules that are too complex, which could significantly delay the large migration.
Have you set up Active Directory in the AWS environment?	Active Directory is used for authentication and authorization. You need to make sure the target account workloads are able to connect to the domain controller for authentication and authorization. You can either add a	Review the Active Directory design with your security and infrastructure teams. Make sure the target AWS account has connectivity to the correct domain controller. Make sure that the target AWS subnet CIDR blocks are

Have you considered?	Description	Actions
	new domain controller in the target VPC, or you can allow the AWS workload to connect to the on-premises domain controllers.	in the correct Active Directory sites so that the workloads in AWS are able to connect to the nearest domain controlle rs.
Have you identified third-par ty connections and applicati on interdependencies?	Third-party connections and application interdependencies require that you modify the firewall rule, network access control list, and security group.	During the deep dive session with the application owners, review the external dependencies for each application. Submit a request to modify firewall rules and the network access control list and change security groups accordingly, based on the third-party dependency requirements.
Does your on-premises environment have any additional security tools that control access and processes running on the systems, such as CyberArk?	You might need to assess and update these security tools in order to allow the migration tools to function in the AWS landing zone.	Review the access policy in your source environment. If a security tool is being used in the access policy, confirm that the tool functions in the AWS Cloud, and then make sure that the migration team has access to both the source and target environments. If any changes are required, add these steps into your migration runbooks.

Document your migration principles

After reviewing the landing zone and on-premises considerations, you should document your answers and decisions. These become the migration principles that guide the rest of the project.

Do the following:

1. In the foundation playbook templates, open the *Migration principles template* (Microsoft Word format).
2. Review the infrastructure, operations, and security considerations in the Landing zone considerations for a large migration and On-premises considerations for a large migration sections of this guide, and discuss the questions with the recommended teams.
3. Document the infrastructure, operations, and security decisions in your migration principles document. For examples of how to record these decisions, see the following table.
4. As needed for your use case, add new categories, items, and principles. For example, you might want to record migration principles for portfolio assessment or project management decisions.

The following is an example of how you might record your decisions to some of the questions in this guide.

Category	Item	Principle
Infrastructure	DNS server	Use Amazon-provided DNS as the primary DNS server for all Amazon Elastic Compute Cloud (Amazon EC2) instances. Set up a condition al forwarder that forwards queries to an on-premises DNS server.
	Security groups	Use a temporary security group to permit all standard infrastructure traffic between the source and target environments.
	EC2 instance types	If utilization data is available from a discovery tool, such as Flexera One or modelizeIT, use this information to help

Category	Item	Principle
		determine the target instance type. If utilization data is not available, size the target instance based on the provisioned central processing unit (CPU) and memory of the on-premises infrastructure.
Operations	Clean up	Servers remain in the staging area until the migration phase is complete, at the end of the hypercare period.
	AWS Backup	By default, the tag applied to each instance is `backup = true`. If backups are not required, the migration teams should change the tag to `false`.
	Monitoring	Use Amazon CloudWatch for monitoring of EC2 instances. After cutover, remove the existing monitoring agent from the target EC2 instances.

Category	Item	Principle
Security	Active Directory	Build a domain controlle r in each VPC, and link the subnet of that VPC to your Active Directory site. For more information, see Designing the Site Topology. This configures all clients to use the correct domain controller.
	Server access	Users must retrieve a password from CyberArk to connect to the source machines.
	AWS Management Console access	Users must use federated login to access the AWS Management Console.

Resources

AWS large migrations

To access the complete AWS Prescriptive Guidance series for large migrations, see Large migrations to the AWS Cloud.

Training resources

For training resources, see the following sections of this document:

- Prerequisites
- Fundamentals
- Advanced

Additional references

- AWS service quotas
- Cloud Enablement Engine: A Practical Guide
- Overview of Data Transfer Costs for Common Architectures (AWS blog post)
- Setting up a secure and scalable multi-account AWS environment

Contributors

The following individuals contributed to this document:

- Chris Baker, Senior Migration Consultant
- Dwayne Bordelon, Senior Cloud Application Architect
- Dev Kar, Senior Consultant
- Wally Lu, Principal Consultant

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.

Change	Description	Date
Updated name of AWS solution	We updated the name of the referenced AWS solution from *CloudEndure Migration Factory* to *Cloud Migration Factory*.	May 2, 2022
Initial publication	—	February 28, 2022

AWS Prescriptive Guidance glossary

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

Numbers

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. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to 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.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

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 (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

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.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

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 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.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

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?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

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.

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.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

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.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

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 computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

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.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, AWS Panorama offers devices that add CV to on-premises camera networks, and Amazon SageMaker provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

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.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

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.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more

information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

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.

ERP

See enterprise resource planning.

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.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

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 importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with :AWS.

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.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

Detective guardrails detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

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.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

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.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

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.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

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.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

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.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

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.

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 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 Cloud 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 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 strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

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.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

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?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

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).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

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.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

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.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

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.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

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.

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.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

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).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

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.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

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.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

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.

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.

test environment

See environment.

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.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

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.

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.

U

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.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

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.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

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.