AWS Prescriptive Guidance

Portfolio discovery and analysis for migration

AWS Prescriptive Guidance: Portfolio discovery and analysis for migration

Table of Contents

Portfolio discovery and analysis for migration

Migration specialists, AWS Professional Services

August 2019 (last update (p. 18): December 2019)

The portfolio discovery and analysis workstream defines the approach for defining, collecting, and analyzing the data required to create a migration plan. This workstream helps you estimate the level of effort and costs associated with migrating your portfolio of applications, as well as the estimated annual run rates for compute and storage resources on Amazon Web Services (AWS).

This workstream also defines the complete dataset required to properly analyze, group and prioritize application migrations, suggests suitable data collection tools if your data is incomplete or inaccurate, determines the criteria to select pilot migration workloads, and provides a general prioritization of migrations. This guide walks you through the high-level steps to accomplish these objectives.

Targeted business outcomes

This guide discusses the AWS prescriptive model for portfolio discovery and analysis. Although other methods are available, they may compromise the completeness or depth of the results.

There are three outcomes expected from portfolio discovery and analysis:

- A detailed knowledge of the migrating environment
- An understanding of the interdependencies of the portfolio
- A migration wave plan that will contain details about which workloads will be migrating and when

To make decisions about the workloads you are migrating, you must fully understand the migrating environment. You must be familiar with each individual workload, as well as how each workload depends on other workloads to support your business.

Developing a well thought out plan that requires minimal revisions helps support your business objectives by mitigating risks like unplanned downtime and reducing the cost of migrating. Although you will find out more about the workloads during portfolio discovery, and need to update the plan accordingly, this guide will help you minimize the frequency and extent of those revisions.

Guidelines

You might find yourself simultaneously running different parts of this workstream with different workloads. For example, you do not need to finish collecting all data for all workloads before considering how to prioritize the workloads. In fact, you will likely do both of them at the same time. The key is to constantly ask “can I move this forward?” and do so if your answer is “yes!”

The three main parts of this workstream are data collection, prioritization, and planning.

- Data collection is focused on understanding what data needs to be gathered and how it will be gathered.
- Prioritization is focused on using the data you have collected about the migrating workloads in order to align the migration plan to your business and technical drivers for the migration.
- Planning is focused on taking the prioritization of the workloads and combining that with the dependency analysis.

Data collection

Begin the project by discovering your environment. The level of detail in the data you collect depends on your business needs. If you need to support a business case or provide financial estimates for other purposes, start by collecting the necessary data for creating annual run rate and migration cost estimates. If you don’t require financial estimates, focus on the required application and infrastructure data. This data supplements the core data needed to create cost estimates and will be used in the analysis phase.

Understanding the dependencies between applications and infrastructure (that is, application to application, application to infrastructure) is critical to determining the impact of moving workloads. The amount of data required can also vary based on factors such as business impact or scope of impact if an application is unavailable, because migration typically requires a planned outage window. It’s rare to get all data collected, so use good judgment to decide when you have enough data to proceed to the next stage.

Decide what data needs to be gathered

You might require no data or complete datasets, depending on the use case for the data. For example, if you are exploring migration costs, you need nothing more than a high-level understanding of your on-premises environment (for example, the number of servers with a breakdown by operating system). The resulting estimates will closely correlate with the accuracy of the data inputs, so keep your output requirements in mind.

If you need to decide how and when applications are going to migrate, you will need a complete and accurate dataset that includes detailed documentation of the applications and infrastructure to be migrated. We strongly recommend that you use an automated application/infrastructure discovery tool to ensure completeness and accuracy. See Resources (p. 8) for a list of these.

Consider what the objective is and what your drivers are, and then determine the data that you will need to collect. A key consideration for deciding whether to collect specific data is how much time it will take to collect.

Decide how to gather the data

When you have determined what data must be collected, determine how to get it. Remember to consider how long and how much effort it will take to get the data.

Your primary decision at this point is whether to install a data collection tool to help you gather data rapidly. Unless you have a compelling reason not to use one, we recommend that you use a discovery tool because it can significantly accelerate discovery. Here are a few questions to ask that will help justify your decision:

- Do my subject matter experts know the answers to the questions?
- Do I have legacy workloads where people who know about those workloads are no longer with the organization?
- What will the discovery tool collect? Does this align with the data I have decided to collect?
- What data will I need to gather manually?
- How long should I expect data collection to take? How long do I have?
- What is the security review process to install a tool? Can we install agents to discover the workloads?
- How long will procurement take? Can I shortcut this by using free tools or AWS Marketplace offerings?
- How accurate does my dataset have to be? Can I take someone's word for it or should I gather more accurate and precise empirical data?

The last question is a key decision that must be made by the leadership team: What is the risk tolerance for making a wrong decision? Wrong decisions happen when you have incomplete or inaccurate data.

When you have decided whether to use a discovery tool, you must define the processes for collating your data sources. Discovery tools are beneficial but they can't give you everything you need. Understand what will and will not be provided by the tool. It generally takes two to four weeks to get good data. While you wait for the discovery tool to collect data, gather the supplemental information you will use in future phases of migration. Here are some examples of data to gather outside a discovery tool:

- Who owns or supports the application?
- What business units does this application support?
- What is the relative importance (criticality or tier) of the application to the business?

> **Warning**
> If you aren't careful, discovery can be an infinite resource drain. At some point, you must decide to move forward with incomplete data. It is nearly impossible to get 100 percent accurate and complete data during discovery. The goal for this phase is not complete accuracy but, rather, good enough data that can get you to the next phase, minimizing the churn you will experience during upcoming phases. After the minimum one to two months of discovery investment, the amount of new data you discover rapidly decreases.

Prioritization

A key to understanding and analyzing a portfolio is aligning your dataset to your organization's business and technical drivers. Align these drivers to the data elements you collect and use them to rank each application that you're planning to migrate. Common drivers include increased agility, cost reduction, and operational resilience. If agility is a primary driver, you might look at how many deployments an application has had or will have in a year. You can then use that data point during prioritization. For cost reduction, you might look at the expected annual savings of right-sizing the environment. For resiliency, you might want a data point that represents the expected revenue lost per hour of downtime.

Decide what criteria to use

In this phase, you work with stakeholders to define the business and technical drivers to prioritize the migrating workloads. Set the migration workload priorities according to the impact the workloads have on your business and technical drivers. Select pilot migration candidates and first movers. These might not be the top applications or migration groups based on portfolio prioritization, because there are special criteria to ensure that the migration and operations teams benefit from pilot migration waves. You **will** learn lessons during every cutover, so it is typical to start with lower risk workloads.

Select 2–10 data points to prioritize your workloads. These data points should represent significant differences in how soon you will gain value from having the workload migrated so you gain more value sooner. A good example would be selecting the business criticality (for example, mission critical or important) if you want to start migrating to gain experience. A poor example would be selecting the business unit (for example, marketing, finance, or facilities) when there isn't a clear driver to migrate one unit before another.

Decide how to use the criteria

Once you choose meaningful data points, determine a scoring scheme for each value of each data point. Assign higher scores for critically lower applications and prioritize them to migrate first.

After scoring values for each data point, you can compare each data point with the other. This optional step is so that you don't need to worry about ensuring you have every value of every data point aligned exactly with your prioritization as it can become overwhelming to keep aligned as you iterate on assigning scores to the values.

To compare data points, use a multiplier for each data point. For example, you can differentiate a Business Critical data point from a Business Unit data point by doing the following:

0.2x	0.4x	0.6x	0.8x	1x
		Business Unit		Business Critical

In this example, the Business Critical scores would stay the same (multiplied by 1) and the Business Unit scores would be 60% of their assigned score (multiplied by 0.6). This indicates that workload Business Critical scores are more important than Business Unit scores.

Once you have assigned a score to each workload, look at the distribution of scores across the entire portfolio. The scores themselves don't matter. It is the difference between scores that matters. For example, you might find that the top score is 8,000 and the bottom score is 800.

We recommend that you plot the scores out as a histogram, so you can verify that you have a good distribution. The ideal distribution will look like a standard bell curve, with a few very high priority workloads and a few very low priority workloads. The majority of workloads will be somewhere in the middle.

> **Important**
> The point of this exercise is to understand which workloads are most valuable to migrate first. So, make sure that you don't have the same or similar scores for most workloads because that would mean that everything has the same priority. We recommend that you look at what is at the top and bottom of the list and see if you agree. If you don't generally agree, you might want to revisit the criteria you used to score the workloads.

Planning

You will assign each application, server, and database that you want to migrate to a wave. Each wave will have a start date and an end (cutover) date. The time between start and end dates can be as short as 2 weeks or as long as 20 weeks, depending on the activities required while preparing to cut over. Creating a high-level roadmap enables you to focus on the technical details of migrating specific workloads and how to migrate the set of workloads at the same time. The wave plan **must** account for dependencies between applications and infrastructure. Building this migration plan will help you understand what the migration timeline will look like and help you set expectations within your organization. The intent of the roadmap is to answer the common questions, “Which workload is moving and when?” and “How long will this migration project take?” The roadmap also provides clarity and focus of what needs to be worked on by the core migration team as well as ancillary teams.

When data collection is complete or sufficient (that is, it’s complete for one workload but not all workloads), you move on to the planning phase. In this phase, you do the following:

- Assign applications and infrastructure elements to migration groups
- Identify your initial migration strategy
- Determine target Amazon Elastic Compute Cloud (Amazon EC2) instance types and Amazon Elastic Block Store (Amazon EBS) volumes

At this point, you can go back to the estimation stage to validate or refine estimates, although this is not required.

Creating the migration groups is a critical step to ensuring that you will be able to meet your migration end date. When you create migration groups, consider each dependency in your environment so that your business doesn’t experience unanticipated results when you migrate to the cloud. Groups are considered atomic migration units, so they should be consistent with workloads that **must** migrate at the same time.

Keep migration groups as small as possible, while also maintaining the integrity of the system’s performance during and after migration. Small migration groups are easier to switch around than larger groups. Small groups allow you handle the inevitable conflicts as they arise. For example, if there’s a last-minute priority 1 outage and you must move a group at a different time, you can adjust with a small group. Large groups don’t allow for this flexibility.

As a general rule, migration groups should not exceed 20 applications, 150 servers, and 30 databases. Migrating more than this in a single cutover can prove very challenging logistically. It also significantly raises your chances of failure, and might require rollback or an extended outage window. We recommend that you justify and critically examine any exceptions to this rule.

Focus most of your time on splitting up the largest groups. Find out if there are easy wins to split into smaller groups. For example, understanding a single dependency allows you to split the group in two. Remember that smaller groups enable more agility in the migration phase. However, this is a complex exercise to conduct on an entire portfolio, and so the analysis should focus on where you will get valuable agility. You don’t need to closely analyze every group. But you should review every group, even if only for a few moments.

Create migration groups

Creating migration groups is both an art and a science. You must make both data-driven and intuitive decisions. The fundamental idea behind migration groups is that you are establishing clear boundaries around what does and doesn’t need to migrate at the same time. To do this, you must look critically at the dependencies of your workloads.

The following are several key data points you must evaluate as part of assessing migration groups:

- Shared databases
- Shared servers
- Application communication

There are other dependency types that could become a factor for migration groups. But this guide focuses on shared databases, shared servers, and application communication because they are present in nearly every environment and have the most significant impact on how your groups will be created.

Provided they do not have dependencies on each other, try to keep different application environments in different groups. For example, put the CRM production environment in a different group from the CRM development and test environments. This allows you to migrate the non-production environments first, so you can test your migration processes and configuration on AWS.

Evaluating shared databases

When you evaluate shared databases, focus on the following:

- First, decide whether you plan to migrate the database or the server with all of its databases. Generally, we recommend moving the database by itself because it helps to create smaller groups, especially if you have a very large shared database server with many unrelated databases.
- Second, decide whether the application must move with the database. An app server must be near a database server for performance reasons. Generally, the answer is going to be "yes, it must move together," but researching this question can gain you significant flexibility in migration.

For example, Applications A, B, and C all connect to db01. A and B are both tightly coupled to the database, and they conduct many write and read operations every second. Application C, however, runs a batch job every night at off-peak hours. You determine that Applications A and B must move with db01, but Application C can move separately. Therefore, Application A, Application B, and db01 are placed in Group 1 and Application C is placed in Group 2.

Evaluating shared servers

Evaluating shared servers is often much easier than evaluating shared databases. Most organizations keep application servers dedicated to an application environment with a few common exceptions. A web server is an example of a common shared server. When deciding how to handle shared servers, try to figure out if the server function is easy to duplicate. In the case of web servers, this is often true because it is trivial to create a new server, install the web server software, and update the configuration.

For example, Applications A, B, and C all use an Apache web server for their web traffic. Applications A and B also share another application server, whereas Application C has its own dedicated application servers. You determine that Applications A and B will migrate with the Apache web server in Group 1. Application C will use a net new web server and migrate in Group 2.

Evaluating application communications

Understanding which applications must move together is where the art of this process starts to come into play. If you attempt to group everything that has a dependency you will likely find that a huge portion of your portfolio, sometimes as high as 80 percent, is connected through a chain of dependencies (for example, Application A requires Application B, which in turn requires Application C, and so on). Because it's usually not feasible to migrate the majority of your portfolio at once, you will likely be forced to separate applications that you would ideally migrate at the same time.

This can be a challenging conversation to have with application subject matter experts because the default response is often, "No, they need to migrate at the same time; I need that application!" The way you phrase your question has a significant impact on getting the data-driven response. The question of whether applications must migrate at the same time is best stated as, "Are these applications tightly coupled?" If the answer is yes, dive deeper and ask these follow-up questions: "What would happen if these applications were offline at separate times?" and "What if they were separated geographically and had (for example) 30 ms of latency in their communication?" This will help you to understand where the limitations to separating applications into different groups truly are.

For example, Application A requires Application B. Application B requires Application C. You determine that if Applications A and B are migrated separately, there will be a significant performance impact to Application A. You also determine that Application B will lose a small portion of functionality while Application C is migrating and offline. You decide to put Application A and Application B in Group 1 and Application C in Group 2.

Tools to help

This process can be challenging, especially if you have no prior experience and a spreadsheet is the only tool you have available. We recommend that you find an AWS Partner Network (APN) Partner who has experience with this exercise and can help lead you through it. AWS Partners have access to tools and resources to assist with this analysis.

If you can't use a partner, we recommend that you visualize the dependencies in your environment. Graph databases are an inherently good tools that can help you analyze your applications, databases, and servers as nodes or vertices, and then also ingest the dependencies as edges.

Other tools to support this process can be found in the migration tools catalog.

Turn the groups into a plan

When you have completed the migration grouping exercise, start scheduling the groups into waves. Each migration group should be assigned an anticipated start and end date. Groups with the same end date are considered part of the same wave.

Consider workloads that ideally migrate at the same time when assigning dates to each group. Successful migrations require significant contributions from many different parts of the business, so publish the migration plan across your entire business. Teams will want to review and schedule their contributions (for example, determine when they should communicate cutover times to users) and prepare themselves for the transformation by training, creating operational plans, and so on.

You might need some additional discovery as you work through the migration plan. These activities include a deep dive into the application architecture where you capture additional data required to successfully design, build, and migrate the application to AWS.

Resources

Tools

- AWS Application Discovery Service
- Tools for discovery and planning from AWS Migration Competency Partners
- Tools for application profiling from AWS Migration Competency Partners
- Tools for data migration from AWS Migration Competency Partners

Partners

- Migration consulting partners
- Migration delivery partners

Guides and patterns

- AWS Prescriptive Guidance guides
- AWS Prescriptive Guidance patterns

FAQ

This section provides answers to commonly raised questions about portfolio discovery and analysis during migrations.

How do I build the right business case?

Your business case should be driven by your organizational KPIs, and the common drivers such as operational costs, workforce productivity, cost avoidance, operational resilience, and business agility.

How do I accurately assess my environment?

Consider the volume of resources used by each application, and automate the assessment process to confirm it is done rapidly and accurately. Assessing your environment manually is a time-consuming process. It exposes your organization to human error. Automating the process will help you gain insight into what you don’t know, and help you more clearly understand and define these uncertainties so they can be factored into your migration strategy.

How do I identify and evaluate the right partners to help me?

Details on the offerings of partners can be found at:

- AWS Migration Partner Solutions
- Migration Solutions in AWS Marketplace

How do I estimate the cost of a large transition like this?

Use the AWS Total Cost of Ownership Calculator to compare how much you will spend running your applications in an on-premises or colocation environment, with what it will cost on AWS.

How long will the migration process take to complete?

Enterprise migrations that are completed within 18 months generate the greatest ROI. The duration of a migration depends on scope and resources.

How do I handle my legacy applications?

Consider taking an incremental approach to your migration by determining which of your legacy applications can be moved most easily. Move these to the cloud first For legacy applications that will require a more complicated approach you can develop an effective plan for migration.

How do I accelerate the migration effort to realize the business and technology benefits more quickly?

Automating the migration process as much as possible, with migration tools from AWS and APN Partners, is the best way to accelerate the migration effort.

AWS Prescriptive Guidance glossary

AI and ML terms (p. 11) | Migration terms (p. 12) | Modernization terms (p. 15)

AI and ML terms

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

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don’t know the target.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

Migration terms

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

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

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

- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. This migration scenario is specific to VMware Cloud on AWS, which supports virtual machine (VM) compatibility and workload portability between your on-premises environment and AWS. You can use the VMware Cloud Foundation technologies from your on-premises data centers when you migrate your infrastructure to VMware Cloud on AWS. Example: Relocate the hypervisor hosting your Oracle database to VMware Cloud on AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there’s no business justification for migrating them.
- Retire – Decommission or remove applications that are no longer needed in your source environment.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

configuration management database (CMDB)

A database that contains information about a company's hardware and software products, configurations, and inter-dependencies. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

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.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the CloudEndure Migration Factory guide in this content set.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines and assigns roles and responsibilities in a project. For example, you can create a RACI to define security control ownership or to identify roles and responsibilities for specific tasks in a migration project.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

service-level agreement (SLA)

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

Modernization terms

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

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization’s applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application’s features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

polyglot persistence

Independently choosing a microservice’s data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization’s capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

For more information, see the Two-pizza team section of the Introduction to DevOps on AWS whitepaper.

Document history

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

update-history-change	update-history-description	update-history-date
Updated guidelines	We added more details to the guidelines for data collection, prioritization, and planning.	December 13, 2019
Initial publication (p. 18)	—	August 5, 2019