AWS Prescriptive Guidance

AWS large-migration strategy and best practices

AWS Prescriptive Guidance: AWS large-migration strategy and best practices

Table of Contents

AWS large-migration strategy and best practices

Amazon Web Services (AWS)

May 2022 (document history (p. 22))

Many Amazon Web Services (AWS) customers want to migrate a large number of servers and applications to the AWS Cloud as fast as possible with the least impact to their business. Migrating 300 or more servers is considered a *large migration*. Your organization might be starting a large migration project because a data center lease is approaching renewal or termination or because your organization is taking the first steps in a technology transformation. However, large scale is not quantified only by the number of servers in scope. It also accounts for the level of organizational transformation that results from the migrations, considering complexities such as people, processes, technology, and priorities.

This guide focuses on your ability to move at scale to AWS. You can migrate existing applications with little to no change. You can use the cloud as a launch point to take those applications to cloud-native or serverless technologies, and you can modernize the applications to unlock additional business benefits.

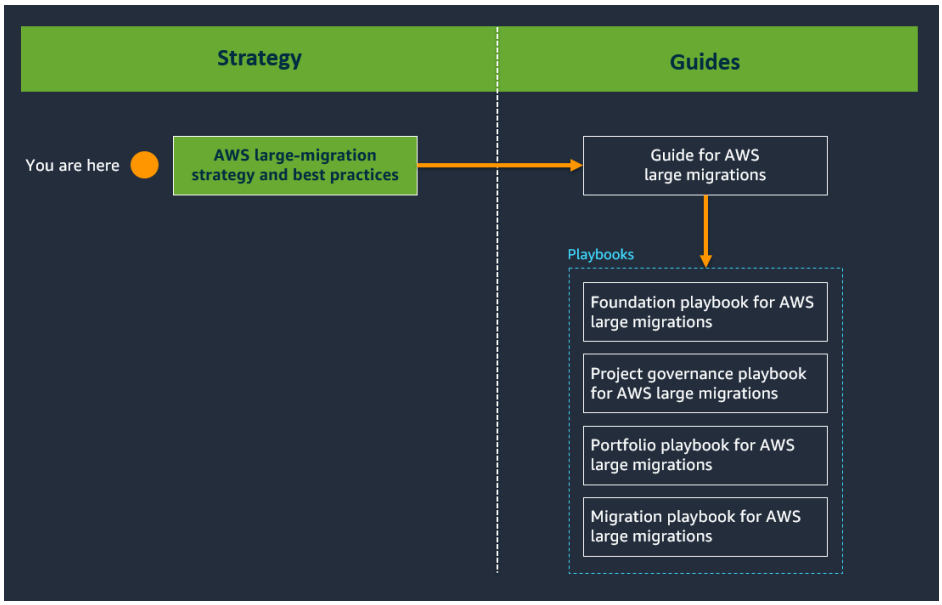
About this strategy

This guide is the first document in a series about large migrations to the AWS Cloud. It discusses best practices for large-scale migrations and provide use cases from customers across various segments, such as financial services, and healthcare. It also provides real-world examples of lessons learned during customer migrations to AWS. The aim of this guide is to assist customers who are at the initial stages of a large-scale migration. However, the best practices and strategies in this guide can be beneficial at any stage of the migration journey. It's assumed that you already have a 100-level knowledge of AWS services and that you're aware of the AWS recommended process for migrating.

When you are finished with this strategy, we recommend reading the Guide for AWS large migrations and the playbooks in the following order:

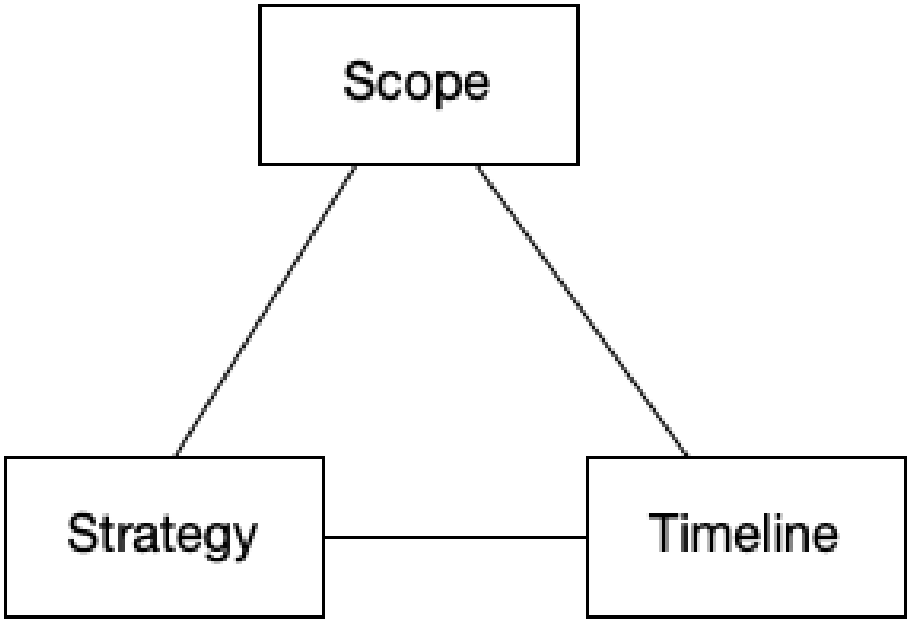
1. Foundation playbook for AWS large migrations
2. Project governance playbook for AWS large migrations
3. Portfolio playbook for AWS large migrations
4. Migration playbook for AWS large migrations

The following figure shows the structure of the AWS documentation series for large migrations. Review the strategy first, then this guide, and then proceed to the playbooks.


Strategy
Guides
You are here
AWS large-migration strategy and best practices
Guide for AWS large migrations
Playbooks
Foundation playbook for AWS large migrations
Project governance playbook for AWS large migrations
Portfolio playbook for AWS large migrations
Migration playbook for AWS large migrations

Scope, strategy, and timeline

Three key elements make up the building blocks of all programs and their relevance in large migrations: scope, strategy, and the timeline.



To set the stage for your migration journey, these elements must be aligned and understood from the start of a migration program. Any changes to one of these elements will affect the others. Realignment must be factored into every change, no matter how basic or sensible the change might seem.

Scope – What are you migrating?

It's common for the total scope of the program to be undefined, even when you're half way through the migration. This is because various factors might not be unpacked until the later stages. For example, halfway through your migration, you might uncover a pocket of shadow IT that was not recorded in your configuration management database (CMDB). Alternatively, the planning might have focused on a server view without considering the supporting network and security services that are required for those applications to run (such as VPN connections to AWS Partners, and certificate authorities to sign certificates). We recommend investing some time in defining the scope, working backwards from your target business outcome. You might end up using discovery tooling to uncover assets, a best practice that will be discussed later in this guide.

The scope will change, because large migrations come with unknowns. These unknowns could be in the form of systems that have become part of the archeology of the environment with little to no understanding of their relevance, or production incidents that cause delays and shifts to the plans you have made. The key is to be flexible and have contingency plans in place to keep the program moving forward.

Strategy – Why do you want to migrate?

You might be planning to migrate to AWS for one or more of the following reasons:

- Your application teams want to implement new CI/CD pipelines, deploy the latest application stacks, or modernize legacy platforms that are out of support.
- Your infrastructure team must get out of an aging data center quickly before the lease expires and the provider turns the power off.
- The board has decided that you need to move to the cloud as a strategic direction, allowing for a fast pace of change in the business's future.

Whatever the reason, all these reasons and more will be on the minds of your business and IT organizations. It's key to understand what your drivers are, to communicate them, and to prioritize them. Each additional driver potentially adds time, costs, scope, and risks to your already-large migration. Being fully aware of the impact that the strategy has on the timeline and scope is key.

After you define your migration strategy, one of the main keys to success is alignment of requirements across the various stakeholders and teams. Performing the migration requires different teams across the organization, including Infrastructure, Security, Application, and Operations. These teams will have individual priorities and other projects that might have already commenced. If these teams are working toward different timelines and priorities, it's more challenging to agree on and implement a migration plan. The migration team and key stakeholders must ensure that all involved teams work toward a single goal and align their priorities with a single timeline of migrations.

We recommend exploring how the desired business outcomes can be aligned across the various teams. For example, migrating to AWS and using AWS Key Management Service (AWS KMS) to encrypt storage at rest might satisfy both the migration and security goals.

Frequently, businesses want to modernize applications, which can result in infrastructure upgrades, while the infrastructure team wants to be frugal and minimize infrastructure changes. The mindset for large migrations should be as basic as possible. The teams involved must avoid trying to do everything at once.

To achieve this, set the right expectations early in the project. The key message should be "Migrate first, then modernize." This approach not only enables organizations to reduce technical debt and operate at scale eventually, it opens avenues for different modernization approaches by using the scalability and agility that the AWS Cloud can provide. Thinking long term will help infrastructure teams to streamline infrastructure deployment and management. As a result the business can have faster feature release cycles.

Timeline – When do you need to complete the migration?

Depending on your business case, you must ensure you are not taking on more than is possible to achieve in the time allocated. If your driver for migrating is based on a fixed date of completion, you must choose the strategy that meets that timeline requirement. Most large migrations are based on these time-based constraints, so the migration strategies must have defined, fixed timelines and outcomes, with little room for extensions or overrun.

In these time-sensitive types of migrations, we recommend the "Migrate first, then modernize" approach. This helps set expectations and encourages the teams to ensure that their individual project plans and budgets are aligned with the overall migration goal. It's important to find out any disagreements as early

as possible in the project, fail fast and address the disagreements at the Steering Committee level, and engage the right stakeholders to ensure that alignment is in place.

Conversely, if your main goal of migration is to gain the benefits of application modernization, this must be called out early in the program. Many programs start with an initial goal based on a fixed deadline, and they don't plan for the requirements from stakeholders who want to resolve outstanding issues and problems. In some cases, these issues have been present for years in the source systems, but now they become artificial blockers to migration.

Modernization activities during a migration can affect the functionality of business applications. Even what is perceived to be a small upgrade, such as an operating system version change, can have a major effect on the program timelines. These should not be considered trivial.

Best practices for large migrations

Large migrations can become challenging, depending on factors that govern how an organization functions. This section covers some of the key factors that can simplify large migrations if addressed during the initial phases of the effort and tracked throughout the project.

The following best practices for large migrations are based on data captured from other customers. The best practices are divided into three categories:

- People
- Technology
- Processes

People perspective

This section focuses on the following key areas of the people perspective:

- Executive support – Identifying a single-threaded leader who's empowered to make decisions
- Team collaboration and ownership – Collaborating among various teams
- Training – Proactively training teams on the various tooling

Executive support

In this section:

Identify a single-threaded leader

When starting a large migration, it's important to identify a single threaded technical leader who is 100 percent dedicated to the project and accountable. That leader is empowered to make decisions, help avoid silos, and streamline work-streams by maintaining consistent priorities.

A large migration global customer was able to scale from one server each week at the outset of the program to more than 80 servers each week at the start of the second month. The CIO's full support as a single threaded leader was critical to the rapid scale up of servers being migrated. The CIO attended weekly migration cutover calls with the migration team to ensure real-time escalation and resolution of issues, which accelerated the migration velocity.

Align the senior leadership team

It's important to create alignment between the various teams regarding the success criteria of the migration. While migration planning and implementation can be accomplished by a small, dedicated team, challenges arise when defining the strategy and performing peripheral activities. These potential

obstacles might require actions or escalations from different areas of the IT organization, including the following:

- Business
- Applications
- Networking
- Security
- Infrastructure
- Third-party vendors

Direct action from application owners, leadership, alignment, and a clear escalation to the single-threaded leader become important.

Team collaboration and ownership

In this section:

Create a cross-functional cloud-enablement team

A critical first step in a large migration project is to enable the organization to work in the cloud. To accomplish this, we recommend building a Cloud Enablement Engine (CEE). The CEE is an empowered and accountable team focused on organization's operational readiness for migrations to AWS. The CEE should be a cross-functional team that includes representation from infrastructure, applications, operations, and security. The team is charged with the following responsibilities:

- Developing policies
- Defining and implementing tools, processes and the architectures that will establish the organizations cloud operations model
- Continuing to facilitate stakeholder alignment across all the areas that they represent

One healthcare customer didn't start with a CEE. However, through initial pilot migrations, the gap was identified. Leading up to the final migration cutover date, with stringent deadlines in place, the team implemented a *migration war room*. In the migration war room, stakeholders from infrastructure, security, applications, and business could assist in resolving issues.

Define requirements for teams and individuals outside the core migration team in advance

Identify teams and individuals that are outside the core program, and define their involvement during the migration planning phases. To facilitate the momentum of the migration during the later stages, pay specific attention to the application teams' involvement. Their knowledge of the application, ability to diagnose issues, and requirement to sign off on the cutover will be required.

While the migration will be led by a core team, the application teams will likely be involved in validating the migration plan and testing during cutover. Customers often approach the cloud migration as an infrastructure project, instead of as an application migration. This can lead to issues during the migration.

We recommend considering the application team's required involvement when selecting a migration strategy. For example, a rehost strategy requires less application-team involvement compared with a replatform or refactor strategy in which more of the application landscape is being changed. If application owner availability is limited, consider using rehost or replatform as opposed to the refactor, relocate, or repurchase strategies.

Validate that there are no licensing issues when migrating workloads

Licensing might change when you migrate corporate off–the-shelf products to the cloud. Your license agreements might be focused on your on-premises estate. For example, a license might be by CPU or linked to a specific MAC address. Alternatively, license agreements might not include the right to host in a public cloud environment. However, renegotiating licensing with vendors can include long lead times and presents a hard blocker for the migration.

We recommend collaborating with your sourcing or vendor management teams as soon as the scope of the migration is defined. Licensing might also influence your target architecture and migration patterns.

Training

In this section:

Train teams on new tooling and processes

After the migration strategy is defined, invest time in understanding what training might be required for the migration and for your target operating model. During the migration, you will likely use tooling, such as AWS Database Migration Service, that is new to your organization. Proactively training teams reduces the delays experienced during the migration phases.

We recommend seeking active knowledge transfer methods that provide an opportunity to experiment with the tooling in a hands-on fashion. As an example, AWS Professional Services provided several Cloud Migration Factory training sessions for three systems integrator (SI) AWS Partners responsible for a large migration. This ensured that the team had basic familiarity as it moved into the migration phase. It also helped identify subject matter experts (SMEs) who could serve as first-line escalation within each SI AWS Partner team.

Technology perspective

Technology provides a great foundation for accelerating large migrations. For example, the Cloud Migration Factory solution is focused on how to provide end-to-end automation for migrations. This section explores some of the best practices for using technology to achieve the scale and velocity required, aligned with the scope, strategy, and timelines.

The overarching principle is to look at areas of automation wherever possible. If you have thousands of servers in scope, performing tasks manually can be a costly and time-consuming effort.

To perform a migration, several tools are typically used, such as the following:

- Discovery
- Migration implementation
- Configuration management database (CMDB)

- Inventory spreadsheet
- Project management

These tools are used at different stages of migrations, from assessment to mobilize through to implementation. Selection of these tools is driven by the business objectives and timelines.

After migration phases are planned, the next step is to ensure that the migration team has the skills to use the tools they will need. If a team lacks the skills or experience, plan targeted trainings to ramp up the skill set. If possible, create events where teams can get experience with the migration tooling in a safe environment. For example, are there sandpit or lab servers that teams can migrate to experience with the tooling? Alternatively, is it acceptable for initial development workloads to be used for learning purposes?

Automation, tracking, and tooling integration

In this section:

Automate migration discovery to reduce the time required

Most large migration programs commence by understanding the scope of the migration (what must be migrated) and developing a strategy (how it will be migrated). Discovery is an important aspect of this. The required metadata points are captured to form a migration strategy decision tree. To migrate workloads at pace, you must identify and import the required migration metadata into your implementation processes, such as a migration factory. A fully automated mechanism to extract, transform, load (ETL) the migration metadata greatly reduces the time and level of effort involved in the discovery process.

One customer developed a fully automated data intake process for their migration factory. The migration wave plan with all the migration metadata was hosted and maintained in a spreadsheet on Microsoft SharePoint. When changes were made to the source, an AWS Lambda function was initiated to load the data into the migration factory without manual intervention. This automated data intake process helped the customer reduce manual work, minimize human error, and accelerate their velocity. They were able to migrate more than 1,000 servers to AWS.

Automate repetitive tasks

In the migration implementation phase, many small processes must be repeated frequently. When using AWS Application Migration Service (MGN), for example, you must install the agent on each server that is in scope of the migration.

Building a migration factory that works for your specific business and technical requirements is the most effective way to achieve the efficiency and velocity required to deliver a successful large migration. A migration factory provides an integration and orchestration framework that uses a standardized dataset to accelerate the migration. After all the tasks are identified, spend time on automating all the manual tasks that can be automated alongside prescriptive runbooks.

The Cloud Migration Factory solution is an example of this. Cloud Migration Factory is designed to provide the migration automation foundations on which you can automate aspects that are specific to your organization. For example, you might want to update a flag in your CMDB to highlight that the

on-premises servers can now be decommissioned. In this scenario, you could create an automation that which performs this task at the end of the migration wave. Cloud Migration Factory has a centralized metadata store with all the wave, application, and server metadata. The automation script can connect to Cloud Migration Factory to get a list of servers in that wave and perform any actions accordingly. Cloud Migration Factory supports AWS Application Migration Service.

Automate tracking and reporting to speed decision making

We recommend building an automated migration reporting dashboard to track and report live data, including key performance indicators (KPIs) for the program. Migration projects involve stakeholders from across the organization, including the following:

- Application teams
- Testers
- Decommissioning teams
- Architects
- Infrastructure teams
- Leadership

To perform their roles, these stakeholders require live data. For example, network teams must know the upcoming migration waves to understand the impact on the shared connection between on-premises resources and AWS. Leadership teams want to understand how much of the migration is complete. Having a dependable, automated live feed of data prevents miscommunications and provides a basis on which decisions can be made.

A large healthcare customer was working toward a data center exit with an upcoming deadline. Given the scale and complexity, a significant amount of time was initially spent on tracking and communicating the migration status between stakeholders. The migration team later used Amazon QuickSight to build automated dashboards that visualized the data, significantly simplifying tracking and communications while increasing the migration velocity.

Explore tooling that can facilitate your migration

Choosing the right tools for your migration is not easy, especially if no one in your organization has managed a large migration before.

We recommend spending time to choose suitable tooling to support the migration. This exploration might involve a license cost, but it can provide a cost benefit when you consider the wider initiative. Alternatively, you might find that tooling embedded in your organization can provide a similar outcome. For example, you might already have application performance monitoring tooling deployed across your estate, which can provide rich discovery information.

A technology customer was initially reluctant to run automated discovery tooling during their migration due to a lack of familiarity. As a result, an SI AWS Partner had to run 5#10 hours of meetings per application to discovery the estate manually, including server names, operating system versions, and dependencies. It was estimated that if discovery tooling had been used, the discovery effort could have been reduced by more than 1,000 hours.

Prerequisites and post migration validation

In this section:

Build the landing zone during the pre-migration phase

We recommend building the AWS target environment, or landing zone, ahead of time, instead of building the target virtual private clouds (VPCs) and subnets during the migration wave. Building a well-architected landing zone is a prerequisite for the migration. The landing zone should include monitoring, governance, operational, and security controls.

Building and validating the landing zone ahead of the migration minimizes the uncertainty that comes with running your workloads in a new environment. With the landing zone in place, the stakeholders can focus on migrating the workloads without worrying about aspects managed at an account or VPC level.

Outline prerequisite activities

Alongside the landing zone, it's important to align other technical prerequisites before the migration, especially processes with lengthy lead time. For example, make the necessary firewall changes to allow the data to be replicated from on premises to AWS. Communicating technical prerequisites early helps to prepare and allocate the resources required. It's common for migrations to stall because prerequisites haven't been met. Not only does this impact the in-progress migration wave, it might push back the dates of all future migrations while the issue is being remediated.

A financial services company intended to perform a mass-migration to AWS, with the goal of vacating several data centers. However, their bandwidth available between on-premises and AWS was not sufficient for the velocity they intended. Unfortunately, increasing the bandwidth required a new connection and had a lead time of three months. This meant that the migration velocity was constrained for the first three months.

Implement post-migration checks for continuing improvement

Finally, remember to implement post-migration validations such as operations integration, cost optimization, and governance and compliance checks. Post-migration validation includes assessing previously migrated workloads to uncover technical lessons learned that should be applied to future waves.

Further, this is a great opportunity to implement cost-control operations. For example, during the migration you might decide to size match the AWS instances to your on-premises estate to reduce the need for performance testing. Now that testing is no longer on the data center closure critical path, you can use Amazon CloudWatch to assess the instance utilization and determine whether a smaller-sized instance would be suitable.

To illustrate the importance of this phase, a large technology customer was performing a large migration but initially did not include post-migration validations. After migrating more than 100 servers, they identified that the AWS Systems Manager Agent (SSM Agent) was not configured correctly. All previously migrated servers had to be remediated, and the migration stalled. The customer also identified that the instances were as large as five times the initial estimates, so they implemented a cost checkpoint at the end of each migration wave.

Process perspective

Processes bring consistency but they also evolve and are susceptible to change because each project is unique. As you run the process repeatedly, you will identify gaps and room for improvements that can add up to huge benefits as you fail, learn, adopt, and iterate. These changes can lead to new ideas or innovations that the project and the business can take advantage of in the future, which provides a catalyst for growth that brings quality and team confidence.

Processes in migrations can be complex as they cross technologies and boundaries that might not have been linked previously. This perspective provides processes and guidance on specific requirements for large migrations.

Preparing for your large migration

The following sections outline the core principles that are required to ensure that you start your migration journey with a clear direction and buy-in from the stakeholders that will be critical to its success.

In this section:

Define business drivers and communicate timeline, scope, and strategy

When approaching a large migration to AWS, you will quickly discover that there are numerous ways to migrate your servers. For example, you could do the following:

- Rehost workloads using AWS Application Migration Service.
- Containerize your application and host it on the Amazon Elastic Container Service (Amazon ECS) or the Amazon Elastic Kubernetes Service (Amazon EKS) managed container platform.
- Redesign your workload into a fully serverless application.

To determine the correct migration path, it's important to work backwards from your business drivers. If your ultimate goal is to increase business agility, you might favor the second two patterns, which involve more levels of transformation. If your goal is to vacate a data center by the end of the year, you might choose to rehost workloads because of the velocity that rehosting provides.

A large migration typically involves a wide range of stakeholders, including the following:

- Application owners
- Network teams
- Database administrators
- Executive sponsors

It is key to identify the business drivers of the migration and include that list in a document, such as a project charter that members of the migration program can access. Furthermore, create key performance indicators (KPIs) that closely align with your target business outcomes.

For example, one customer wanted to migrate 2,000 servers within 12 months to achieve their target business outcome of vacating their data center. However, their security teams were not aligned toward this goal. The result was several months of technical debates on whether to miss the data center closure date but modernize applications further or to rehost initially to enable the timely data center closure and then modernize applications on AWS.

Define a clear escalation path to help remove the blockers

Large cloud migration programs typically involve a wide range of stakeholders. After all, you're potentially changing applications that have been hosted on premises for several decades. It's common for each of the stakeholders to have conflicting priorities.

While all the priorities might drive value, the program will likely have a limited amount of budget and a defined target outcome. Managing the various stakeholders and focusing on the target business outcomes can be challenging. This challenge is compounded when you multiply it by the hundreds or thousands of applications that are in scope of the migration. Further, the stakeholders likely report into different leadership teams, which have other priorities. With this in mind, alongside clearly documenting the target business outcomes, it's important to define a clear escalation matrix to help remove blockers. This can save a significant amount of time and help align the various teams toward a common goal.

One example that demonstrates this is a financial services company whose goal was to vacate their primary data center within 12 months. There wasn't a clear mandate or escalation path, which resulted in the stakeholders crafting their desired migration paths, regardless of time and budget constraints. Following an escalation to the CIO, a clear mandate was set and a mechanism was provided for requesting required decisions.

Minimize unnecessary change

Change is good but more changes mean more risks. When the business case for the large migration is approved, there is most likely a target business outcome driving this initiative, such as vacating a data center by a specific date. While it's common for technologists to want to re-write everything to take full advantage of AWS services, this might not be your business goal.

One customer was focused on a two-year migration of the company's entire web-scale infrastructure to AWS. They created a two-week rule as a mechanism to prevent application teams from spending months rewriting their applications. By using the two-week rule, the customer was able to sustain a long-term migration with a consistent cadence when hundreds of applications had to be moved over a multiple-year period. For more information, see the blog post The Two-Week Rule: Refactor Your Applications for the Cloud in 10 Days.

We recommend minimizing any change that doesn't align with the business outcome. Instead, build mechanisms to manage these additional changes in future projects.

Document an end-to-end process early

Document the complete migration process and assignment of ownership in the early stages of a large migration program. This documentation is important in educating all stakeholders about how the migration will run and their roles and responsibilities. The documentation will also help you to understand where issues might occur and to provide updates and iterations of the process as you progress through the migrations.

During the development of the migration project, ensure that any existing processes are understood and that integration points and dependencies documented clearly. Include places where engagement with external process owners will be required, including change requests, service requests, vendor support, and network and firewall support. After the process is understood, we recommend documenting ownership in a responsible, accountable, consulted, informed (RACI) matrix to track the different activities. To finalize the process, establish a countdown plan by identifying the timelines involved in each step of the migration. The countdown plan generally works backwards from the workload migration cutover date and time.

This documentation approach worked well for a multinational home appliance corporation that migrated to AWS successfully in less than a year and exited four data centers. They had six different organizational teams and multiple third parties involved, which introduced management overhead resulting in back-

and-forth decisions and delays in implementation. The AWS Professional Services team, together with the customer and their third parties, identified key processes for the migration activities and documented them with respective owners. The resulting RACI matrix was shared and agreed upon by all involved teams. Using the RACI matrix and an escalation matrix, the customer alleviated the blockers and issues that were creating delays. They were then able to exit the data centers ahead of schedule.

In another example of using RACI and escalation matrices, an insurance firm was able to exit the data center in less than 4 months. The customer understood and implemented a shared responsibility model, and a detailed RACI matrix was followed to track the progress of each process and activity throughout the migration. As a result, the customer was able to migrate over 350 servers in the first 12 weeks of implementation.

Document standard migration patterns and artifacts

Think of this as creating cookie cutters for the implementation. Reusable references, documentation, runbooks, and patterns are the key to scale. These journal the experience, learning, pitfalls, issues, and solutions that future migration projects can reuse and avoid, significantly accelerating the migration. The patterns and artifacts are also an investment that will help improve the process and guide future projects.

For example, one customer was performing a year-long migration where applications were being migrated by three different SI AWS Partners. In the early stages, each AWS Partner was using their own standards, runbooks, and artifacts. This placed numerous stresses on the customer teams, because the same information could be presented to them in different ways. After these early pains, the customer established central ownership of all documentation and artifacts to be used in the migration, with a process for submitting recommended changes. These assets include the following:

- A standard migration process and checklists
- Network diagram style and format standards
- Application architectural and security standards based on business criticality

In addition, changes to any of these documents and standards were sent out to all teams on a weekly cadence, and each partner was required to confirm receipt and adherence to any changes. This greatly improved communication and consistency for the migration project, and when a separate large migration effort in another business unit started, that team was able to adopt the existing process and documents, greatly accelerating their success.

Establish a single source of truth for migration metadata and status

When it comes to planning a large migration, establishing a source of truth is important to keep the various teams aligned and enable data-driven decisions. When you start this journey, you might find numerous data sources that you can use, such as the configuration management database (CMDB), application performance monitoring tooling, inventory lists, and so on.

Alternatively, you might find that there are few data sources and you must create mechanisms to capture the data needed. For example, you might need to use discovery tooling to uncover technical information, and to survey IT leaders to obtain business information.

It's important to aggregate the various data sources into a single dataset that you can use for the migration. You can then use the single source of truth for tracking the migration during the implementation. For example, you can track which servers have been migrated.

A financial services customer that wanted to migrate all workloads to AWS focused on planning the migration with the dataset that had been provided. This dataset had key gaps, such as business criticality and dependency information, so the program started a discovery exercise.

In another example, a company in the same industry moved into migration wave implementation based on an out-of-date understanding of their server infrastructure inventory. They quickly started to see migration numbers decrease because the data was incorrect. In this case, application owners were not understood, which meant that they could not find testers in time. Additionally, the data were not aligned to the decommissioning that their applications teams had completed, so servers were running without being used for a business purpose.

Running your large migration

After you have established your business outcomes and communicated the strategy to the stakeholders, you can move to planning how you carve up the scope of the large migration into sustainable migration events or waves. The following examples provide key guidance for making the wave plan.

In this section:

Plan migration waves ahead of time to ensure a steady flow

Planning your migration is one of the most important phases of the program. It goes with the saying "if you fail to plan, you plan to fail." Planning migration waves ahead of time allows the project to flow swiftly as the team becomes more proactive to the migration situation. It helps the project scale more easily, and it improves decision making and forecasting as project demands increase and become complex. Planning ahead also improves the team's ability to adapt to changes.

For example, a large financial services customer was working on a data center exit program. Initially, the customer planned the migration waves in a sequential fashion, completing one wave before beginning to plan the next. This approach resulted in less time to prepare. When the stakeholders were informed that their applications were being migrated to AWS, they still had several steps to perform before starting the migration. This added significant delays to the program. After the customer realized this, they implemented a holistic and future-focused migration planning stream where migration waves were planned several months in advance. This provided enough notice for the application teams to perform their pre-migration activities such as notifying AWS Partners, licensing analysis, and so forth. They could then remove those tasks from the program's critical path.

Keep wave implementation and wave planning as separate processes and teams

When wave planning and wave implementation teams are separate, the two processes can work in parallel. With communication and coordination, this avoids slowing down the migration because not enough servers or applications are ready to achieve the expected velocity. For example, the migration team might need to migrate 30 servers each week, but only 10 servers are ready in the current wave. This challenge is often caused by the following:

- The migration implementation team was not involved in the wave planning, and the data collected in the wave planning phase are not complete. The migration implementation team must collect more server data before starting the wave.
- Migration implementation is scheduled to start right after wave planning, with no buffer between.

It is critical to plan waves ahead of time, and to create a buffer between preparation and the start of the wave implementation. It is also important to make sure that the wave planning team and the migration team work together to collect the right data and avoid rework.

Start small for great outcomes

Plan to start small and increase migration velocity with each subsequent wave. The initial wave should be a single, small application, fewer than 10 servers. Add additional applications and servers in subsequent waves, building up to your full migration velocity. Prioritizing less complex or risky applications, and ramping velocity on a schedule, gives the team time to adjust to working together and to learn the process. In addition, the team can identify and implement process improvements with each wave, which can substantially improve the velocity of later waves.

One customer was migrating more than 1,300 servers in a year. By starting with a pilot migration and a few smaller waves, the migration team was able to identify multiple ways to improve later migrations. For example, they identified new data center network segments earlier. They worked with their firewall team early in the process to put in place firewall rules that allowed communication with migration tooling. This helped prevent unnecessary delays in future waves. In addition, the team was able to develop scripts to help automate more of their discovery and cutover processes with each wave. Starting small helped the team focus on early process improvements, and greatly increased their confidence.

Minimize the number of cutover windows

Mass migrations require a disciplined approach to driving scale. Being too flexible in some areas is an anti-pattern for large migrations. By limiting the number of weekly cutover windows, time spent on cutover activities has higher value.

For example, if the cutover window is too flexible, you could end up with 20 cutovers with five servers each. Instead, you could have two cutovers with 50 servers each. Because the time and effort for each cutover are similar, having fewer, larger cutovers reduces the operational burden of scheduling, and limits unnecessary delays.

A large technology company was trying to migrate out of a few leased data centers before contract expiration. Missing the expiration would result in expensive, short-term renewal terms. Earlier in the migration, application teams were allowed to dictate migration schedule up to the last minute, including opting out of migration for any reason just days ahead of cutover. This led to numerous delays in the early stages of the project. Often, the customer had to negotiate with other application teams at the last minute to fill in. The customer eventually increased their planning discipline, but this early mistake led to constant stress for the migration team. Delays to the overall schedule resulted in some applications not making it out of the data centers in time.

Fail fast, apply experience, and iterate

Every migration has pitfalls initially. Failing early helps the team learn, understand the bottlenecks, and apply the lessons learned to larger waves. It is expected that the first couple of waves in a migration will be slow for the following reasons:

- Team members are adjusting to each other and the process.
- Large migrations usually involve many different tools and people.
- It takes time to integrate, test, fail, learn, and continuously improve the end-to-end process.

Issues are common and expected during the first couple of waves. It is important to understand and communicate this to the entire organization, because some teams may not like to try new things and fail. Failure can discourage the team and become a blocker for future migrations. Making sure everyone understands that initial issues are part of the job and encouraging everyone to try and fail is key to a successful migration.

One company planned to migrate more than 10,000 servers in 24–36 months. To achieve that goal, they needed to migrate nearly 300 servers a month. However, that does not mean they migrated 300 servers from day one. The first couple waves were learning waves so that the team could understand how things worked and who had permissions to do what. They also identified integrations that would improve the process, such as integrating with CMDB and CyberArk. They used the learning waves to fail, improve, and fail again, refining the process and automation. After 6 months, they were able to migrate more than 120 servers each week.

Don't forget the retrospective

This is an important part of an agile process. It's where the team communicates, adjusts, learns, agrees, and moves forward. A retrospective at the most basic level is looking back, discussing what happened, determining what went well and what needs to improve. Improvements can then be built based on those discussions. Retrospectives wrap some formality or process around the idea of lessons-learned. Retrospectives are important because to achieve the scale and velocity for large migrations to succeed, the processes, tools, and teams must constantly evolve and improve. Retrospectives can play a significant part in that.

Traditional lessons-learned sessions do not happen until the end of a program, so often these lessons do not get reviewed at the start of the next migration wave. With large migrations, lessons learned should be applied to the next wave and should be a key part of the wave planning process.

For one customer, weekly retrospectives were held to discuss and document lessons learned from the cutovers. In these sessions, they uncovered areas where there was scope for streamlining from a process standpoint or automation. This resulted in implementation of a countdown schedule with specific activities, owners, and automation scripts to minimize manual tasks, including validation of third-party tools and Amazon CloudWatch agent installation, during cutover.

At another large tech company, regular retrospectives were held with the team to identify problems with previous migration waves. This resulted in process, script, and automation improvements that drove the average migration time down by 40 percent over the course of the program.

Additional considerations

Many areas must be factored into a large migration program. The following sections provide thoughts on other items that must be considered.

In this section:

Clean up as you go

A migration isn't considered successful if it costs 10 times what you expected, and the project is not complete until the resources used for migration are shut down and cleaned up. This cleanup should be part of the post-migration activity. It ensures that you will not leave unused resources and services in your environment that will add to the costs. Post-migration cleanup is also a good security practice for preventing threats and vulnerabilities that expose your environment.

Two key outcomes of moving to the AWS Cloud are the cost savings and the security. Leaving unused resources can defeat the business purpose of moving to cloud. The most common resources that are not cleaned up include the following:

- Test data
- Test databases

- Test accounts, including firewall rules, security groups, and network access control list (network ACL) IP addresses
- Ports provisioned for testing
- Amazon Elastic Block Store (Amazon EBS) volumes
- Snapshots
- Replication (such as stopping the data replication from on-premises to AWS)
- Files that consume space (such as temporary database backups used to migrate)
- Instances that host the migration tools

In one example of bad cleanup practices, SI AWS Partners were not removing replication agents after a successful migration. An AWS audit discovered that replication servers and EBS volumes that had already been migrated were costing $20,000 (USD) each month. To mitigate the issue, AWS Professional Services created an automated audit process that notified SI AWS Partners when stale servers were still being replicated. The SI AWS Partners could then take action on unused and stale instances.

For future migrations, a process was adopted to define a post-migration hypercare period of 48 hours to ensure smooth platform adoption. The customer's infrastructure team then submitted a decommission request for on-premises servers. It was advised that upon approval of the decommission request, servers of the respective wave would be removed from the application migration service console.

Implement multiple phases for any additional transformation

When carrying out a large migration, it's important to remain focused on your core goal, such as data center closure or infrastructure transformation. In smaller migrations, scope creep might have a minimal impact. However, a few days of additional effort multiplied by potentially thousands of servers can add a significant amount of time to the program. Furthermore, the additional changes might also require updates to documentation, process, and training for support teams.

To overcome potential scope creep, you can implement a multiple-phase approach to your migration. For example, if your goal was to vacate a data center, phase 1 may include only rehosting the workload to AWS as fast as possible. After a workload is rehosted, phase 2 can implement transformational activities without risking the target business outcome.

For example, one customer planned to exit their data center in 12 months. However, their migration encompassed other transformation activities, such as rolling out new application performance monitoring tooling and upgrading operating systems. More than 1,000 servers were in scope of the migration, so these activities added a significant delay to the migration. Furthermore, this approach required training in the use of the new tooling. The customer later decided to implement a multiple-phase approach with an initial focus on rehost. This increased their migration velocity and reduced the risk of not meeting the data center closure date.

Conclusion

Large migrations present different challenges when compared to smaller migrations. This is mostly due to the complexities introduced by the scale. For example, installing an agent onto a single server is fairly straightforward and will take approximately 5 minutes. However, if you have 5,000 servers in scope for your migration, this will take approximately 416 hours and will present the following challenges:

- It's likely that there are multiple operating systems that require different processes.
- There might be separate Microsoft Active Directory domains to manage due to previous mergers and acquisitions.
- Effective processes and tools are required to orchestrate the agent installation for each wave and then track and report the progress.

This strategy outlines large migration best practices based on AWS Professional Services experiences helping a wide range of customers. This includes people, process, and technology perspectives. If you want to start or are in the process of migrating to AWS, consultants at AWS Professional Services would be happy to assist you. Contact your AWS representative to start the conversation.

Resources

AWS large migrations

Guide

- Guide for AWS large migrations

Playbooks

To start your large migration journey, we encourage you to read the detailed playbooks in the following order:

1. Foundation playbook for AWS large migrations
2. Project governance playbook for AWS large migrations
3. Portfolio playbook for AWS large migrations
4. Migration playbook for AWS large migrations

Other AWS Prescriptive Guidance resources

- Automating large-scale server migrations with Cloud Migration Factory
- Best practices for assessing applications to be retired during a migration to the AWS Cloud
- Setting up a secure and scalable multi-account AWS environment
- Evaluating migration readiness
- Mobilize your organization to accelerate large-scale migrations

Other references

- AWS Cloud Migration Factory solution
- Free cloud migration services on AWS
- AWS Database Migration Service
- Migrate with AWS

Videos

- Executing a large-scale migration to AWS (AWS re:Invent 2020)
- CloudEndure Migration Factory best practices (AWS re:Invent 2020)

Contributors

This strategy was authored by the global Large Migration tiger team within AWS Professional Services. The team has successfully migrated thousands of servers to AWS on behalf of AWS customers. Contributors to this document include:

- Chris Baker, Senior Migration Consultant
- Dwayne Bordelon, Senior Cloud Application Architect
- Rodolfo Jr. Cerrada, Senior Application Architect
- Pratik Chunawala, Senior Cloud Architect
- Bill David, Principal Customer Solutions Manager
- Dev Kar, Senior Consultant
- Wally Lu, Principal Consultant
- Jon Madison, Senior Cloud Infrastructure Architect
- Abhishek Naik, Senior Solution Architect
- Damien Renner, Senior Migration Specialist
- Amit Rudraraju, Senior Cloud Architect

Document history

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

Change	Description	Date
Removed CloudEndure Migration service (p. 22)	We removed references to the CloudEndure Migration service. AWS Application Migration Service is the primary migration service recommended for lift-and-shift migrations to the AWS Cloud.	May 11, 2022
Updated name of AWS solution (p. 22)	We updated the name of the referenced AWS solution from *CloudEndure Migration Factory* to *Cloud Migration Factory*.	May 2, 2022
Updated resources (p. 22)	We updated the Introduction and Resources sections with the latest documents in the large migration series.	March 8, 2022
Initial publication (p. 22)	—	September 16, 2021

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.

Migration terms

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 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.

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.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

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 migration

A migration of 300 or more servers.

Migration Acceleration Program (MAP)

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

Migration Portfolio Assessment (MPA)

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

Migration Readiness Assessment (MRA)

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

migration at scale

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

migration factory

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

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs (p. 23) entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

operational-level agreement (OLA)

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

operations integration (OI)

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

organizational change management (OCM)

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

playbook

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

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

responsible, accountable, consulted, informed (RACI) matrix

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

runbook

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

service-level agreement (SLA)

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

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.