AWS Well-Architected Framework

Migration Lens

Migration Lens: AWS Well-Architected Framework

Table of Contents

Migration Lens - AWS Well-Architected Framework

Publication date: **January 24, 2024** (*Document revisions*)

This whitepaper describes the Migration Lens for the AWS Well-Architected Framework. It provides AWS customers with a set of Well-Architected best practices and guidance on the migration of their on-premises or hybrid workloads into a fully cloud-based implementation.

Introduction

The three phases of an AWS Migration consist of *assess*, *mobilize*, and *migrate and modernize*. The Well-Architected Framework superimposes the six pillars (operational excellence, security, reliability, performance efficiency, cost optimization, and sustainability) to help you reduce your cloud migration and implementation risks.

In doing so, the Migration Lens combines the three phases of migration (assess, mobilize, and migrate and modernize) and the six pillars of the AWS Well-Architected Framework, and serves as a foundational guidance for migration best practices that customers can reference to evaluate the decisions they make on their migration and measure them against AWS best practices.

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. Using the Framework, you can learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems in the cloud. It provides a way for you to measure your architectures against best practices and identify areas for improvement. We believe that having well-architected systems greatly increases the likelihood of business success.

One of the first decisions to make when you start thinking about migrating workloads to the cloud is to decide your migration strategy. A migration strategy is the approach used to move applications to the cloud, also known as the *7 Rs*: *retire, retain, rehost, relocate, repurchase, replatform,* and *refactor*. For more details on each R, see Definitions.

The Migration Lens focuses specifically on rehost, relocate, replatform, and retire migration strategies. The refactor strategy involves modernizing the application during the migration. These topics are addressed in other AWS publications. For brevity, we have only covered details from the Well-Architected Framework that are specific to migration. Consider best practices and questions that have not been included in this document when designing your architecture. We recommend

that you complete a full Well-Architected Framework Review (WAFR) prior to performing this AWS Migration Lens review.

This lens whitepaper is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, and operations team members. After reading this paper, you should understand AWS best practices and strategies to use when migrating workloads to the AWS Cloud.

Lens availability

The Migration Lens is available as an AWS-official lens in the Lens Catalog of the AWS Well-Architected Tool.

To get started, follow the steps in Adding a lens to a workload and select the **Migration Lens**.

Definitions

Well-Architected terminology

- **Workload**: A workload is used to identify a set of components that together deliver business value. A workload is usually the level of detail that business and technology leaders communicate about.
- **Technology portfolio**: Within an organization, the technology portfolio is the collection of workloads that are required for the business to operate.
- **Architecture**: An architecture is a set of IT services and components that work together in a workload.
- **Component**: A component is the code, configuration, and AWS resources that together deliver against a requirement. A component is often the unit of technical ownership, and is decoupled from other components.
- **AWS Well-Architected Framework**: A framework that provides a consistent set of best practices for customers and partners to evaluate architectures, and provides a set of questions you can use to evaluate how well an architecture is aligned to AWS best practices based on six pillars.
- **Trade-offs**: Trade-offs are decisions you make while architecting a workload, based on business context, that drives your engineering priorities.

Migration terminology

- **AWS Migration:** AWS Migration is the process of moving applications and data from one location, usually an organization's private on-site (on-premises), or other cloud providers, to the AWS Cloud.
- **Migration process:** Migration process is the three-phase approach methodology designed to help your organization migrate tens, hundreds, or thousands of applications. While each phase is a common component of a successful migration, they are not discrete phases, but an iterative process.
- **Migration drivers:** Migration drivers are the reasons an organization uses to make a business decision to move to the cloud. Reasons could include reducing capital expenditure, decreasing ongoing cost, improving scalability and elasticity, improving time-to-market, and attaining improvements in security and compliance

- **Migration key performance indicators (KPIs)**: Metrics you identify at the start of your migration project, after you establish migration goals, to measure the success of these goals.
- **Migration phase:** A migration phase refers to one of the following three phases: *assess, mobilize,* or *migrate and modernize*.
- **Assess:** *Assess* is the first migration phase. At the start of your journey, you assess your organization's current readiness for operating in the cloud. Most importantly, you want to identify the desired business outcomes and develop the business case for migration.
- **Mobilize:** *Mobilize* is the process of creating a migration plan and refining your business case. You address gaps in your organization's readiness that were uncovered in the assess phase, with a focus on building your baseline environment (the *landing zone*), driving operational readiness, and developing cloud skills. Consider this phase as a pilot migration project.
- **Migrate and modernize:** During the *migrate and modernize* phase, each application is designed, migrated, and validated. Leverage the services below through our migration specialists, with one of our migration competency partners, or on your own to start the process of moving applications and data to AWS.
- **Migration services:** Migration services are a comprehensive portfolio of AWS migration services, migration competency partners, and mature third-party migration tooling ecosystem. They provide automation and intelligent recommendations based on AWS machine learning to simplify and accelerate each step of the three-phase migration process.
- **Migration strategy**: The approach used to migrate a workload into the AWS Cloud. There are seven migration strategies for moving applications to the cloud, known as the seven Rs.
- **Retire:** Retiring the application means that you can shut down the servers within that application stack.
- **Retain:** This is the migration strategy for applications that you want to keep in your source environment or applications that you are not ready to migrate. You might choose to migrate these applications in the future.
- **Rehost (lift and shift):** Rehost is the process of moving applications from your source environment to the AWS Cloud without making any changes to the application.
- **Relocate:** Relocate is transferring a large number of servers, comprising one or more applications, at a given time from on-premises platform to a cloud version of the platform. For example, you can use this strategy to transfer servers in bulk from VMware software-defined data center (SDDC) to VMware Cloud on AWS.
- **Repurchase (drop and shop)**: Repurchase means replacing your application with a different version or product.

- **Replatform (lift, tinker, and shift)**: Replatform is moving an application to the cloud and introducing some level of optimization in order to operate the application efficiently, reduce costs, or take advantage of cloud capabilities.
- **Refactor**: Refactor is moving an application to the cloud, and modifying its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability.
- **AWS Migration Acceleration Program (MAP):** MAP is a comprehensive and proven cloud migration program based on our experience migrating thousands of enterprise customers to the cloud. MAP provides tools that reduce costs, automate execution, and accelerate results. It also offers tailored training approaches and content, expertise from AWS Professional Services, a global partner ecosystem, and AWS investment.
- **MAP specialized workloads**: MAP specialized workloads is a scaling mechanism designed to accelerate migration and modernization of on-premises workloads to AWS. MAP specialized workloads are available for Mainframe, Windows, Storage, VMware Cloud on AWS, SAP, Databases, and Connect.
- **Cloud Center of Excellence (CCoE)**: CCoE is the process of building a team of subject matters experts across business segment with cross functional skills and experiences to lead the migration project across the organization.

Design principles

Well-Architected migration design principles are a set of considerations used as the basis for a well-architected migrated workload. The design principles are high-level guidance that we recommend you follow for a successful migration.

1. **Create a clear vision for the journey**: Define the business goal of the migration (*why* migrate?) and think more about *how* you transform both your business and technology while migrating to AWS, not just about moving your workloads. The why and transformation goals are the key principles that help you define everything else during the migration journey.
2. **Get leadership support early in the project**: Executive sponsorship is critical as strong organizational alignment is required to make timely decisions and resolve potential challenges and tradeoffs. Consider forming a dedicated team, usually known as CCoE (for more detail, see What is a cloud center of excellence and why should organization create one?)
3. **Understand where you are moving to**: Learn about AWS and its differences from traditional on-premises data centers (or other clouds). This is critical to the migration's success. Define your AWS accounts strategy and leverage AWS Control Tower and AWS Organizations to build landing zones to provide ongoing account management, governance, and implementation of AWS best practices.
4. **Define the migration scope:** Determine what to migrate and, based on that, define the migration strategy (how to migrate).
5. **Know your applications:** Define what good looks like when moving to AWS. Based on that, choose the right migration strategy for each based on the 7 Rs approach (for more detail, see Determining the R type for migration).
6. **Get the application owners and teams buy-in early**: Understand application dependencies, both technical and business, internal, and external.
7. **Understand your application requirements**: For example, performance, utilization, resiliency, security, compliance, and operations. Optimize (right-size) to create more efficient and elastic workloads after the migration.
8. **Align with applicable governance, regulatory and compliance frameworks**: Incorporate security best practices as part of your workload migration strategy.
9. **Maintain your operations during the migration to the Cloud**: Consider your current operations while you have a hybrid environment and once you are completely on cloud. Plan your monitoring, backup, and lifecycle management at level required by your organization for production workloads.

10. **Create the migration plans**: Split migration into smaller units (migration waves), where each unit could be an application or a group of applications. Move each unit, test, validate and repeat. Create or adapt operational runbooks, seeking to automate the process of early migration waves (mobilize) and apply these runbooks to scale the migration of each subsequent wave.

Migration lifecycle

The iterative approach to cloud adoption discussed in this guide can be broken out into the three high-level phases of *assess, mobilize,* and *migrate and modernize*. These phases are briefly described below, and each phase could be considered as a separate process.

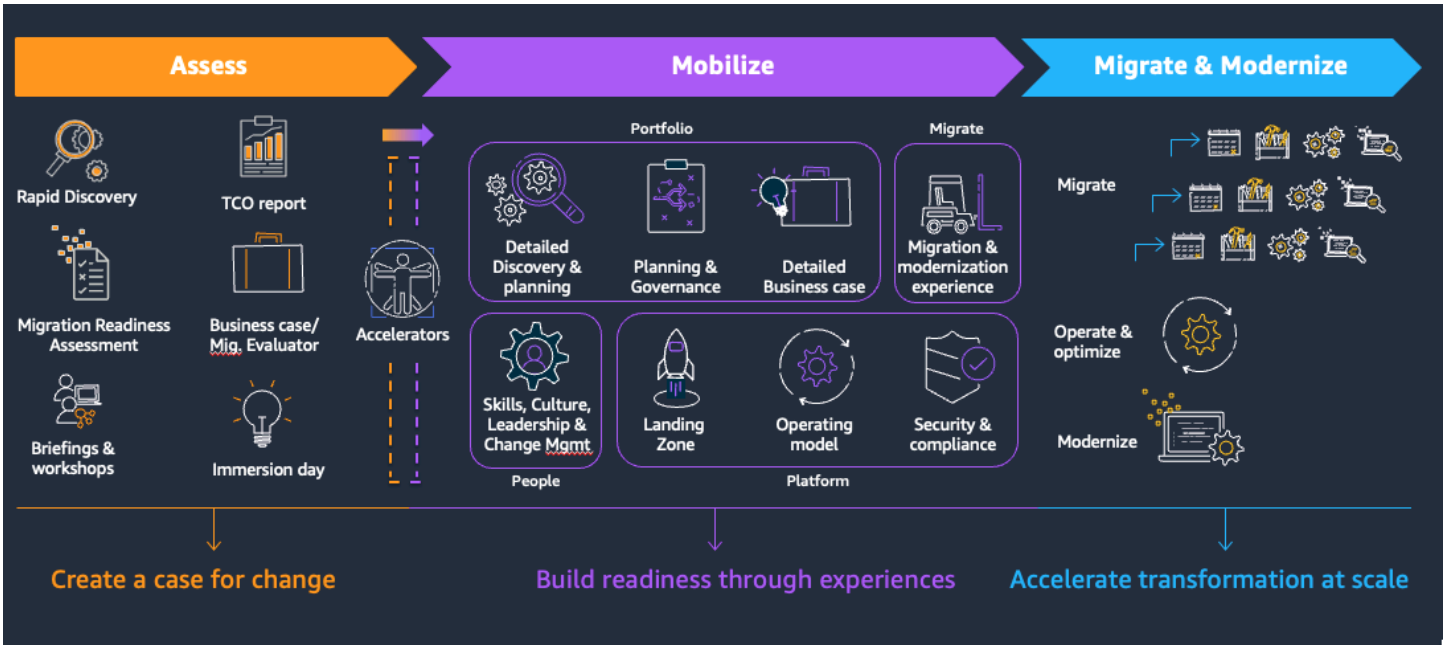


Figure 1- Migration Phases

Assess

Planning data center migrations can involve thousands of workloads that are often deeply interdependent. Server utilization data and dependency mapping are important early first steps in the migration process. The first phase of a cloud migration begins with collecting configurations, usage, and behavior data from your servers to help you better understand your workloads. For more information, see Assess phase.

Mobilize

The goal of the mobilize phase is to build foundational capability both in the organization and the AWS environment, with hands-on migration experience focused on security and operations automation. This process brings together your portfolio of tools and practices in a scalable and secure AWS landing zone. In this phase, you migrate a small set of business applications to

the cloud, while enforcing an agile and scalable delivery culture, team structure, and change management process. Some of the activities of Mobilize phase include defining applications for migration and selecting the migration strategy for each, defining and automating security, and building a team of skilled staff to manage the migration. For more information, see Mobilize phase.

Migrate and modernize

The migrate phase uses the patterns, processes, tools, resources, and methodology defined and tested during the mobilize phase to migrate applications at scale. After using the best practices and lessons learned from the earlier phases, you can implement a migration factory solution through automation and agile delivery. For more information, see Migrate phase.

Well-Architected migration

While migration is usually a linear process, the cloud adoption journey consists of ongoing constant and recursive improvement cycles.

Regardless of where you are in your migration journey, you can apply the Well-Architected Framework and Migration Lens perspectives. Each pillar of the Migration Lens has specific questions and best practices aligned per migration phase, so you can navigate to the most relevant recommendations related to your current migration phase, or review all recommendations per pillar across all the phases. The Well-Architected migration lifecycle, shown in Figure 2, takes the migration phases described and applies the Well-Architected Framework pillars to each phase.

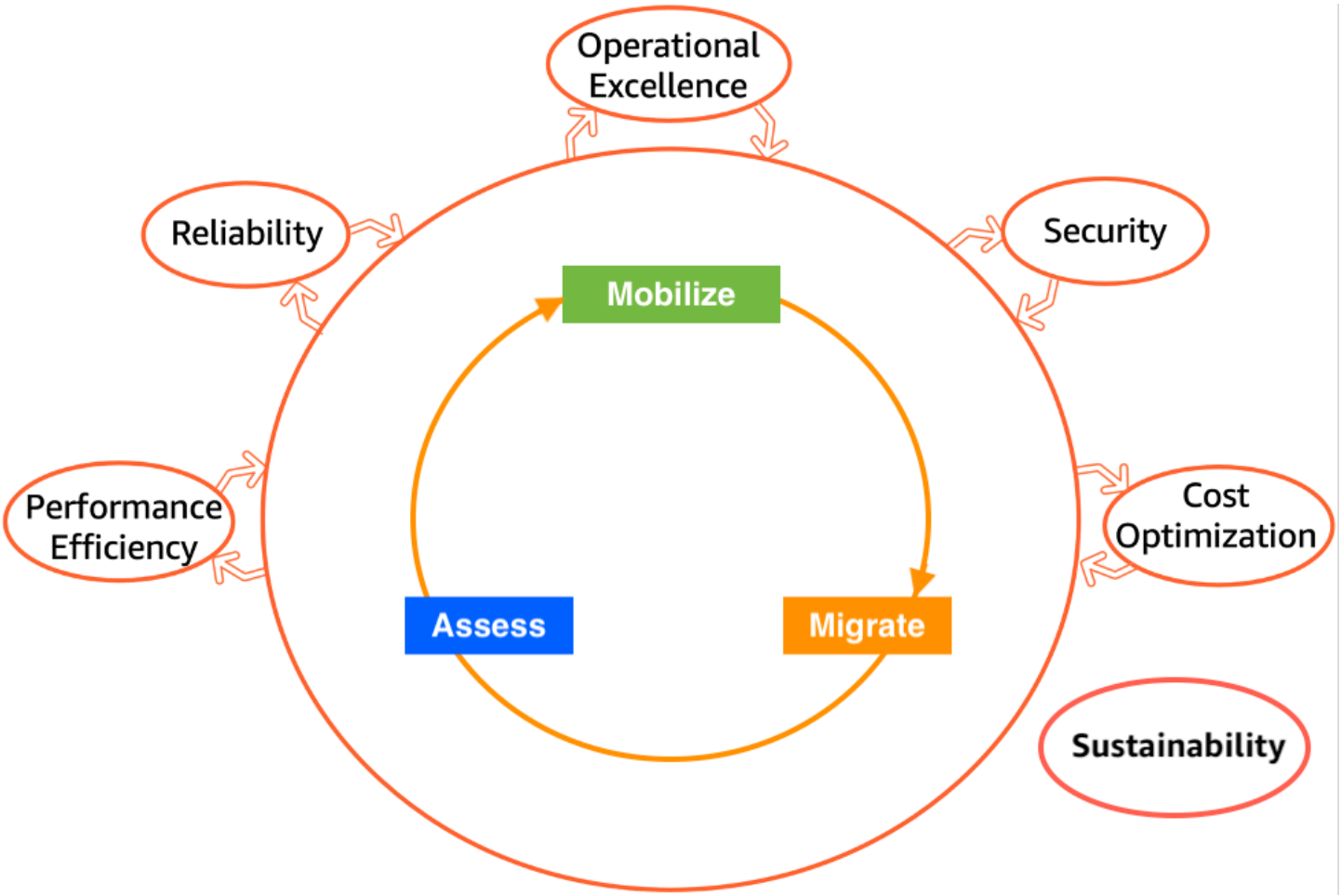


Figure 2- Well-Architected Migration

The pillars of the Well-Architected Framework

The AWS Well-Architected Framework provides architectural best practices for designing and operating workloads in the cloud. The Framework consists of six pillars:

- **Operational excellence**: Includes the ability to run, monitor, and gain insights into workloads. It enables delivering business value and improves supporting processes and procedures. Best practice focus areas include organization, prepare, operate, and evolve.
- **Security**: Includes the ability to protect information, systems, and assets. It enables delivering business value through risk assessments and mitigation strategies. Best practice focus areas include security foundations, identity and access management, detection, infrastructure protection, data protection, incident response, and application security.
- **Reliability**: Includes the ability of a workload to recover from infrastructure or service disruptions. Ensures a workload performs its intended function correctly and consistently when it's expected to. It enables dynamically acquiring computing resources to meet demand, and mitigating disruptions such as misconfigurations and transient network issues. Best practice focus areas include foundations, workload architecture, change management, and failure management.
- **Performance efficiency**: Focuses on the efficient use of computing resources to meet requirements. It enables maintaining efficiency as demand changes and technologies evolve. Best practice focus areas include architecture selection, compute and hardware, data management, networking and content delivery, and process and culture.
- **Cost optimization**: Includes the continuous process of refinement and improvement of a system over its entire lifecycle. It enables building and operating cost-aware systems that minimize costs, maximize return on investment, and achieve business outcomes. Best practice focus areas include Cloud Financial Management, expenditure and usage awareness, resource cost-effectiveness, resource demand and supply management, and optimization.
- **Sustainability**: Focuses on environmental impacts, especially energy consumption and efficiency, since they are important levers for architects to inform direct action to reduce resource usage. Best practice focus areas include Region selection, alignment to demand, software and architecture, data, hardware and services, and process and culture.

For more information on the Framework and its pillars, see the AWS Well-Architected Framework whitepaper.

Pillars

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization
- Sustainability

Operational excellence

Achieving operational excellence in cloud migrations involves addressing various crucial aspects through the three migration phases: assess, mobilize, and migrate. This includes realizing benefits, organizing necessary skills within the organization, ensuring adequate bandwidth for migration, defining operational requirements for the target estate, and closely monitoring migration velocity. Proactive attention to these areas can significantly mitigate the risk of migration delays or operational issues arising after the workloads have been migrated. For instance, it's vital to confirm the presence of AWS expertise within your organization and confirm the availability of resources to manage workloads before initiating the migration process.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

The assess phase of AWS migration is a crucial step that lays the foundation for a successful migration journey. In this phase, you delve into various aspects of your migration plan, aligning it with your organization's goals. To achieve this, you must consider the scope, technology, and processes involved. Your migration plan should be based on up-to-date data obtained through a comprehensive discovery exercise, particularly vital for long-running migrations. This data informs your migration patterns and helps in refining your plans regularly.

MIG-OPS-01: Does your migration planning consider scope, technology, and process?

Your migration planning is the key to a successful migration to AWS, and needs to cover many aspects. These include ensuring you have the right skills at the points when they are needed and the capacity required to meet your timeline, scope, and budget. The requirements on your staff are driven by what you are migrating, so your plan has to be based on up-to-date data from your environment, obtained through a discovery exercise early on in the migration process. For long running migrations spanning over a year, this data needs to be refreshed and used to refine the plans on a regular basis. The gathered discovery data may inform which migration patterns can be adopted.

MIG-OPS-BP-1.1 Your migration plan must be informed by and reflect technology, processes and business

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 1.1.1: Define the scope. What are you migrating?

In large migrations, the scope of the program can often remain undefined, even when you're halfway through the migration process. This uncertainty arises because certain factors may only surface in later stages. For instance, you might discover pockets of shadow IT or overlooked network and security requirements essential for your applications to function correctly. To address this, it's advisable to invest time in clearly defining the scope, starting from your desired business outcomes and potentially using discovery tools to uncover assets, as discussed later in this guide.

Furthermore, it's important to acknowledge that the scope is likely to evolve as large migrations frequently encompass unexpected elements. These surprises may include unidentified systems or unforeseen production incidents that can disrupt your plans. Therefore, it's crucial to remain adaptable and have contingency plans in place to ensure the smooth progress of your migration program. For more detail, see Strategy and best practices for AWS large migrations.

Suggestion 1.1.2: Understand your current on-premise inventory. Identify the missing details you need to collect and select the right discovery tool to do so.

Begin by identifying the information you have available regarding the environment you intend to migrate and the format in which it exists. Determine what additional information is missing,

which is crucial for the workloads you plan to migrate. For instance, you may need insights into your on-premises database systems, networking metrics, application dependencies, visualization requirements, or other resource utilization profiles. Based on these requirements, you should select a discovery tool that can provide this information while adhering to your organization's operational and security standards (for example, whether the tool should be agent-based or agent-less).

Once you've pinpointed the discovery requirements, use a comparison matrix to filter out some of the available options. This results in a list of discovery tools that meet your specific requirements. Following this initial filtration, it is advisable to apply this three-step technique to further narrow down and prioritize the list of discovery tools based on the essential features required for your business.

Suggestion 1.1.3: Perform a comprehensive portfolio discovery exercise to understand dependencies and complexity.

Completing a portfolio and discovery exercise is a requirement for a successful migration. In rehost migrations, this discovery exercise needs to be focused on capturing inventory, server to application mapping, and dependencies between systems in order to understand the affinities to drive migration waves and planning. It also provides validation of the scope and approach. For further guidance, see Portfolio playbook for AWS large migrations.

Suggestion 1.1.4: Familiarize yourself with recommended strategies and best practices for large migrations.

Migrating to AWS can be driven by various reasons, such as moving away from aging data centers, improving operational resilience and security posture, or reducing costs. Regardless of the motive, it's crucial to identify and prioritize these drivers, as they can impact migration in terms of time, cost, scope, and risk. Alignment of requirements across different teams, including Infrastructure, security, application, and operations, is key to a successful migration. This alignment aligns everyone towards a common goal and timeline. It's essential to explore how desired business outcomes can harmonize with various team objectives. In large migrations, it's advisable to adopt a *migrate first, then modernize* mindset to manage technical debt efficiently and leverage AWS scalability for long-term benefits in infrastructure deployment and feature release cycles. For detail on setting up and running a migration project at scale, see Strategy and best practices for AWS large migrations.

Suggestion 1.1.5: Familiarize yourself with the technology available to you to expedite the migration.

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. For more detail, review the Technology Perspective in Strategy and best practices for AWS large migrations. Check this guidance to explore some of the best practices for using technology to achieve the scale and velocity required, aligned with the scope, strategy, and timelines.

Suggestion 1.1.6: Define your process.

Having a well-defined process is a key for a successful migration. Things like clear escalation path to remove blocker, communication plan, and change request process are examples of processes that need to be defined and refined as the migration occurs. For a more comprehensive list of example processes to define, see Process perspective.

Mobilize

The mobilize phase of migration is a critical step in a smooth transition to the cloud. It involves comprehensive planning and preparation, guided by best practices to maximize success. During this phase, it's essential to establish mechanisms for tracking planned versus actual business benefits and assess your team's skills, implementing training plans where required. Acquire the necessary bandwidth to operate workloads while migrating to the cloud in parallel. Establish a Cloud Center of Excellence (CCoE) responsible for cloud operations, and define a comprehensive Cloud Operations Strategy, covering resource allocation, security, cost management, and governance to streamline the migration process effectively.

MIG-OPS-02: How do you report on planned versus actual business benefits throughout your migration?

We see customers with many drivers for migrating their workloads to AWS, including consolidating or vacating data centers, achieving cost savings, improving security and operational resilience, increasing business agility, and improving staff productivity. Throughout the migration, a wide range of decisions need to be made, such as the target Amazon Elastic Compute Cloud (Amazon EC2) instance type or what type of Amazon Elastic Block Store (Amazon EBS) to use. These micro decisions may have a large impact on achieving the expected business benefits at an organizational level. Defining and measuring KPIs helps make sure that the program remains on track to achieve the target business outcomes, and influences the behavior of stakeholders who are making the

decisions throughout a migration. Furthermore, it helps identify when the program is trending negatively against a KPI so that remediating actions can be applied following a deep-dive review.

MIG-OPS-BP-2.1 Define and measure key performance indicators (KPIs) which can be shared with all teams involved in the migration

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 2.1.1: Establish KPIs that support your target business outcomes.

When determining the KPIs, it's crucial to work backwards from your target business outcome. There might be more than one target business outcome for your migration, so you usually need multiple KPIs. It's recommended to socialize the KPIs with your organization's leaders to create alignment towards the goals. For example, if you're migrating to AWS to increase your operational stability, start by determining specific KPIs that measure operational capability. This could include comparing service availability with the agreed service level agreements (SLAs), the number of unplanned outages per quarter, and mean time to resolve issues. Once this is understood, you can define how it is measured in AWS so that comparisons can be made.

For more detail, see The Importance of Key Performance Indicators (KPIs) for Large-Scale Cloud Migrations.

MIG-OPS-03: Have you assessed your team's skills, identified any gaps, and implemented a training plan while tracking progress?

There's no denying that migration entails additional work for your teams. For those responsible for operating and managing the current environment, it necessitates acquiring new skills for migrating, operating, and managing applications effectively in the AWS Cloud. This can result in hesitation or resistance. However, training your teams not only fosters familiarity with the AWS Cloud, but also offers clarity on their roles in this new environment, ultimately boosting their self-confidence in their capabilities.

MIG-OPS-BP-3.1 Invest time and effort to ensure the required migration and operations skills are captured, skills gaps identified, and training plans are implemented and managed

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 3.1.1: Assess your team's current skill level and training needs.

Before embarking on your migration journey, we strongly recommend conducting an AWS Learning Needs Analysis to evaluate the current roles and develop tailored training plans for each individual, aligning their skills with the requirements post-migration to AWS. Once the plan is established, you can identify knowledge gaps and begin preparing your teams. Depending on your team's existing knowledge level of migration, it's advisable to structure your training plan into three tiers: prerequisites, fundamentals, and advanced.

For a large migration project, it is essential that every team member completes the prerequisite-level training, which covers fundamental information about the AWS Cloud and migration concepts. As for the fundamentals and advanced levels, you can use a training plan to assign each workstream a suitable training tier. We recommend structuring training by workstreams rather than job roles and titles, as these can vary significantly across organizations. For more detailed information on training plans, see the following:

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

Suggestion 3.1.2: Include references to requirements for on-premises or legacy systems.

Having the requisite skills from day one is paramount to the successful migration of workloads, especially when dealing with the transition from on-premises or legacy estate. This is essential for maintaining and improving the level of service provided post-migration.

Suggestion 3.1.3: Consider engaging with AWS ProServe or a certified AWS Migration Competency Partner to accelerate your migration readiness.

Education is not something that happens overnight in many cases, and if there is a requirement to move faster, engaging with an AWS Migration Competency Partner or AWS Professional Services

could be a worthwhile option. For more detail, contact your AWS account manager. Another great way to find individuals with the specific AWS skills you need is through AWS IQ for Experts.

Suggestion 3.1.4: Run an AWS Experience Based Accelerator (EBA).

Running an AWS Experience Based Accelerator (EBA), more specifically a Migration EBA, is another great way to improve your team's experience through migrating non-production or pilot applications, with the comfort of having AWS migration experts working alongside you. For more information, contact your AWS account team.

Suggestion 3.1.5: Leverage other available training resources available.

There are other training resources that you can leverage to improve your team's migration skills. For example, the AWS Migration Immersion Day is a one day workshop that emulates an on-premise migration and allows customers to run a migration to AWS. The migration flow is aligned with Migration Acceleration Program (MAP) best practices and includes steps from the assess, mobilize, and migrate phases.

MIG-OPS-04: Do you have the bandwidth required to operate your workloads while deliverin g the cloud migration in parallel?

Migration initiatives require involvement and input from various teams in order to be successful. For example, input is required from application owners to determine the move groups (like which servers and applications must be migrated at the same time), as well as shaping the target architecture. This extends to operational teams who are required to support pre-migration, migration, and cutover activities. At the same time, they need to perform the roles they carried out before the migration initiative (like maintaining workloads) and training on the AWS Cloud, so that they are prepared to support workloads once they are migrated. This makes it important to understand if your teams have the bandwidth required to operate your workloads while delivering the cloud migration in parallel.

MIG-OPS-BP-4.1 Build a comprehensive resource model for your migration that reflects the demands of the migration as well as the regular activities

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 4.1.1: Identify a cloud sponsor.

This sponsor serves as a driving force behind the migration, linking key performance indicators (KPIs) and objectives to the organization's overarching business goals. In essence, a migration sponsor helps navigate the complexities of migration, making critical decisions, and ultimately propelling the organization towards realizing the full benefits of the AWS Cloud.

Suggestion 4.1.2: Consider the workstreams, roles and team composition required for your migration.

Review People foundation for guidance on workstreams, roles, and team composition before you start constructing your migration workstreams and teams. When assigning resources to roles from your existing teams, be sure to assess their current utilization and where that demand comes from (like normal business operation or other business initiatives and projects).

Suggestion 4.1.3: Consider augmenting your existing teams with skilled resources from other parts of your organization or from a trusted partner.

You cannot expect to run a large-scale time-bound migration without increasing your teams' workload. If you are not time-bound, and the migration can be spread over a longer period of time, then it may be possible to use the teams you have. However, the higher the migration velocity, the sooner you realize the full benefits of being in the cloud, so it could make sense to engage additional migration resources, such as AWS Professional Services or AWS Migration Competency Partners to assist without impacting your business operations.

Alternatively, you may decide to leverage AWS Managed Services to extend your team with operational capabilities, including monitoring, incident management, AWS Incident Detection and Response, security, patch, backup, and cost optimization for migrated workloads.

MIG-OPS-05: Have you established a Cloud Enablement Engine (CEE) responsible for operating your cloud environment?

A key focus of the Cloud Enablement Engine (CEE) is transforming the information technology (IT) organization from an on-premises operating model to a Cloud Operating Model (COM). These components include the operations, security and control, platform architecture and governance, and infrastructure provisioning and configuration management functions. The target state

architecture, as defined by your migration strategies and patterns, dictates the services and platforms that need to be catered for. The Cloud Enablement Engine (CEE), sometimes referred to as Cloud Center of Excellence (CCoE), is defined as a multi-disciplinary team that is assembled to implement the governance, best practices, training, and architecture needed for cloud adoption in a manner that provides repeatable patterns for the larger enterprise to follow.

MIG-OPS-BP-5.1 Build a Cloud Center of Excellence (CCoE) team within your organization as part of your migration planning

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 5.1.1: Review the Foundation playbook for AWS large migrations.

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

Suggestion 5.1.2: Establish a cross-functional CCOE in your organization.

One of the foundational steps that enterprises take as part of their journey to the cloud is establishing a Cloud Center of Excellence (CCoE). The CCoE is a multi-disciplinary team that is assembled to implement the governance, best practices, training, and architecture needed for cloud adoption in a manner that provides repeatable patterns for the larger enterprise to follow. Many companies have found that CCOE can accelerate their migrations to the cloud and broader digital transformations. More details on best practices to creating CCOE be found in Designing a Cloud Center of Excellence (CCOE) blog post.

Suggestion 5.1.3: Define the operational support model during migration.

In a migration to cloud scenario, it is likely you need to maintain a capability to provide operational support to your on-premises environment, as well as your new cloud environment, at least while you exit your on-premises estate. Operational support for the cloud environment may come from the team that currently provides on-premises support, but frequently a new team is created with skills and experience in the cloud services to be consumed to provide operational support in the cloud. For more detail, see Building your Cloud Operating Model.

Suggestion 5.1.4: Define a RACI matrix (responsible, accountable, consulted, and informed).

A clear shared understanding of where the operational support delineation lines are to ensure a consistent and reliable operational support service. A RACI matrix can be used to capture each domain and activity and identify who is responsible, accountable, consulted, and informed in each case. For guidance on creating a cloud operations RACI matrix, see Create a RACI or RASCI matrix for a cloud operating model.

MIG-OPS-06: What is your cloud operations strategy?

When migrating to AWS, you most likely have people, tools, and processes already in place to manage your current on-premises architecture. However, what you have now may not be aligned or best suited to the environment you are migrating to. Before migrating workloads to cloud, you should establish an operational capability in terms of people, tools, and process that provides all the required operational capabilities expected by the business. Initially this may be a minimum viable product (MVP) capability aligned with supporting non-production or pilot migrations, but prior to migrating production or business-critical applications, it is strongly recommended to have a full operational capability in place, serving all necessary operational requirements.

MIG-OPS-BP-6.1 Define Cloud Operations Strategy: understand your current operating model, processes and tools, and explore how to implement them efficiently, securely and reliably in the cloud to create your cloud operations strategy

This BP applies to the following best practice areas: Organization

Implementation guidance

Suggestion 6.1.1: Prepare the people and process involved in your project for the activities in each stage of the large migration.

You need to define the workstreams in your project, organize individuals into functional teams, confirm that the roles and responsibilities are well understood, and complete the necessary training. For more detail, see People foundation.

Suggestion 6.1.2: Check the operations perspective in Cloud Adoption Framework.

The operations perspective focuses on delivering cloud services at an agreed-upon level with your business stakeholders. Automating and optimizing operations allows you to effectively scale while

improving the reliability of your workload. For more detail, see Operations perspective: health and availability.

Suggestion 6.1.3: Create your cloud operating strategy and model.

The process of modernizing operations in the cloud involves readiness, automation, and integration. To be operationally ready for your migrated workloads, incorporate tools, people, and process to deliver the various activities that together create a cloud operating model. For guidance on creating a cloud operations strategy and model, see Modernizing operations in the AWS Cloud.

Suggestion 6.1.4: Train operational teams on operational AWS services.

Cloud native tools and services for operational capabilities are built for the cloud, and exhibit increased scalability, reliability, and availability. In many cases, cloud-based tools can be used to manage on-premises environments, so a transition to cloud-hosted operational tools may be a sensible approach to avoid duplication of tooling. However, your operations teams need to be trained on these new tools prior to migration. Complete an AWS Learning Needs Analysis for each member of your team to provide them an education plan to meet their role specific requirements.

Suggestion 6.1.5: Consider using managed service providers or AWS Managed Services (AMS).

If your organization doesn't have enough operational capability to fully cover your cloud operational strategy, consider using managed service providers (MSPs) or AWS Managed Services (AMS) offerings as an initial step. AMS helps you accelerate your adoption of AWS at scale and operate more efficiently and securely. AMS leverages standard AWS services and offers guidance and execution of operational best practices using specialized automation, skills, and experience that are contextual to your environment and application. For a selection of AWS-certified cloud operations managed service providers, see Find an AWS Partner. For more detail on AMS offerings, see AWS Managed Services.

MIG-OPS-BP-6.2 Align AWS operational requirements with your existing tools and identify any gaps

This BP applies to the following best practice areas: Prepare

Implementation guidance

Suggestion 6.2.1: Define your AWS operational requirements and identify operational tools. Mapping your AWS operational capability requirements to your existing operational tools and

processes helps identify where there are gaps, allowing you to build an action plan to fill them. For example, you might decide to use AWS Backup to centrally manage and automate your backups on AWS. We recommend reviewing the AWS Well Architected Framework Operational Excellence Pillar for guidance on developing your operational requirements specification and the aligned AWS tools.

MIG-OPS-BP-6.3 Regularly test your operations in the cloud

This BP applies to the following best practice areas: Operate

Implementation guidance

Suggestion 6.3.1: Simulate operational events.

One way to test operational capability is to simulate life-like system failures. An effective way to do this is by running events in your organization, also known as game days. Game days test systems, processes, and team responses, and help evaluate your readiness to react and recover from operational issues. The purpose is to actually perform the actions the team would perform as if an exceptional event had happened. The Build Your Own Game Day to Support Operational Resilience blog from AWS guides you through this process and provides links to further information.

MIG-OPS-07: How many servers do you plan to replicate and migrate in each wave, and what factors have you considered when arriving at this number?

The number of servers that can be included in a migration wave, and the duration required to perform the migration, is generally dictated by the people you have to perform the migration, the applications you are migrating, the migration approach used, and the network bandwidth available between the source location and AWS.

For example, let's assume a customer has 1,000 servers to migrate in order to vacate their source data center. They're planning to rehost all of their servers using the AWS Application Migration Service (MGN) and have calculated it'll take approximately five weeks to complete a migration wave from an end-to-end perspective (including change control, governance, migrating the data, and acceptance testing). On average, their migration waves include 50 servers, so with one migration team it could take approximately two years to complete (100 weeks). However, they have sufficient network bandwidth and people to increase this to four migration teams working in parallel, so their migration could take approximately 25 weeks to complete. During the 25-week window, there's a two week change freeze where all migrations are impacted, which means their

total estimated migration duration is 27 weeks, with an average velocity of 200 servers every five weeks.

MIG-OPS-BP-7.1 Calculate your potential migration velocity using both technical and non-technical perspectives (like network bandwidth, team availability, volume of changes, and change freezes)

This BP applies to the following best practice areas: Prepare

Implementation guidance

Suggestion 7.1.1: Determine how many migration waves you need.

When calculating how long your entire migration may take, we recommend first determining how many migration waves you need to perform based on the size of the in-scope estate and how they are migrated.

Suggestion 7.1.2: Assess available non-technical resources.

Assess your human resources to determine how many waves you can run in parallel to achieve the target outcomes, and validate it aligns with the business goals.

Suggestion 7.1.3: Determine technical limitations like bandwidth.

Assess your network bandwidth to estimate how many waves you can run in parallel to achieve target outcomes, and validate it aligns with the business goals. For more detail, see AWS Application Migration Service best practices.

Suggestion 7.1.4: Include process estimations such as change management, testing strategy, and outage and maintenance windows.

Don't forget to include aspects like change freezes, which impact the migration.

Suggestion 7.1.5: Understand the volume of change necessary for each application in-scope for migration.

Your migration velocity is heavily influenced by how well you know your applications. When using rehost to lift and shift your applications to the cloud, there may still be configuration changes required for the application to work as expected after migration. You need to know what configuration changes are required, who can perform them, how long they will take to perform,

and if the changes can be automated. This information should be gathered in a discovery exercise during the migration planning phase and should influence the applications that are assigned to each migration wave. You should have the people required to make these changes (ideally with automation) during the migration event so that the cut over can be performed within the expected time frame. For more detail, see Application portfolio assessment guide for AWS Cloud migration.

Migrate

The migrate phase is a pivotal stage in any migration process, and effective strategies are key to success. It's essential to have a well-defined testing phase strategy that covers thorough testing of workloads in the AWS cloud environment. Rigorous and appropriate testing helps identify and rectify issues before they impact operations. Additionally, it's crucial to review your application lifecycle management, such as the CI/CD pipeline, and make necessary adjustments once your workloads are in the AWS Cloud. Aligning your application deployment and management processes with the cloud environment can enhance efficiency and maximize the benefits of AWS migration.

MIG-OPS-08: What is your testing phase strategy?

Every application migration has a testing phase, and you have to plan how the tests are done, what to test, and what are the test criteria. Functional testing ensures the seamless integration of your application with the new environment, requiring the development of comprehensive unit tests to validate application workflows. Meanwhile, performance testing evaluates system response times, identifies bottlenecks, and facilitates optimization efforts, with cycles of testing and optimization as needed.

MIG-OPS-BP-8.1 Ensure you have a testing strategy in place

This BP applies to the following best practice areas: Prepare

With a rehost migration strategy, there is generally no need to perform a full regression functional test, like what you might perform with a major update to an application's code base. Logically, the application architecture and code base are unchanged in AWS, but there have been changes in the infrastructure and it's important to focus the testing on those areas.

When using a rehost migration pattern, your source workloads are cloned into AWS, and when they launch on the Amazon EC2 platform, they may attempt to speak to the services and applications

which are still hosted on-premises. This could cause outages to your live systems and corrupt application data, so any testing with cloned workloads must be performed within an isolated network environment, or while the source systems are powered off. Even with source systems powered off, there can still be complications with shared user authentication systems (like Microsoft Active Directory) being updated by test systems.

This need for isolation makes meaningful application testing challenging. Technically, you could provision an isolated network in AWS for testing, but this would also need to permit safe and secure access to a number of shared infrastructure services, perhaps interface with other business critical applications still on-premises, and end users would need to be able to connect to the test application to run the tests. This generally requires significant effort, without really providing the assurance sought, as so much needs to be implemented to protect or replicate the source environment that it becomes questionable if you are actually performing representative testing.

Implementation guidance

Suggestion 8.1.1: Use a risk-based, *points of change* testing strategy with your applications.

The primary change point is in the network, as the application's servers would be hosted in the AWS Cloud, which may have changed the prevailing network latency and could negatively impact end-user performance or interfaces with other applications. Additionally, there may be new controls implemented in the network, with additional firewall rules or network security groups which could impact connectivity to internal or external users, other applications, or external systems. You should create a series of test cases that perform transactions most likely to be impacted by increased network latency, or new network controls, and this may need to be performed from a variety of different user types (for example, internal browser user, fat-client desktop internal user, external browser user, or cloud-hosted virtual desktop user). Have a minimal set of functional test cases that validate the basic application capabilities (for example, user login, navigation around the application, and access to interfacing systems), and create a baseline set of test results from the applications before migration. Finally create a set of test cases that validate operational integration with the cloud operating strategy and model to verify the operational readiness to run the applications in AWS Cloud.

Suggestion 8.1.2: Test potential network latencies before moving workloads to the AWS Cloud.

Understand the network latency variance between the source and target environment. Measure the network latency between your users and your on-premises application servers, then deploy test servers on your target AWS networks and measure network latency between your users and the test workloads. If you have multiple source and target geographies, sites, or campuses

and user locations (such as virtual desktop on-premises, fat-client on corporate network, or remote laptop on internet and VPN), document and baseline each scenario, then provide the network latency baselines to the migration planning team. It's a common scenario to migrate some of your workloads while keeping identity and access management systems on-premises (for example, Microsoft Active Directory). In this case, you need to consider the extra latency required to authenticate user and application activity. One best practice to minimize potential impact is to extend and configure your on-premises Active Directory domain into AWS, so that workloads running in AWS can communicate with Microsoft AD services hosted there. To explore the options for Microsoft Active Directory in AWS, see Active Directory Domain Services on AWS.

Suggestion 8.1.3: Test application performance before and after migration.

If application transaction performance is important, procure up-to-date performance tests results for the applications before migration, and repeat the same performance tests using the same test suite in the AWS Cloud. Attempting to compare test results from different test tools won't give you the assurance you want. Restrict performance testing to only what is needed to validate acceptable performance, with tests that focus on the points of change in the migrated application instance. If performance testing is not within an acceptable range, a rollback decision should be taken immediately to avoid impacting your business. Consider using automated test tools to minimize the time and effort required.

Suggestion 8.1.4: Perform a test cutover in AWS Application Migration Service.

The server test cutover is essential for confirming Application Migration Service is able to successfully create a clone of the source server which can boot up on the Amazon EC2 platform. The test cutover should be performed within an isolated subnet in AWS, especially for Active Directory connected Windows workloads, to protect live systems and data hosted on-premises. Application Migration Service provides the ability to specify which AWS subnet to use for test cutovers.

MIG-OPS-09: Have you reviewed your application lifecycle management (like your CI/CD pipeline) and verified if it needs any adjustment once your workloads are in the AWS Cloud?

CI/CD pipelines or software development lifecycles (SDLC) can vary in complexity between applications and businesses. Many contain deployment steps that interact with the underlying infrastructure in order to provision and de-provision resources, while others may just need

connectivity to code repositories and tools. When migrating these applications to AWS, take these processes into consideration, as it requires multiple stages to migrate both the running application and the associated deployment pipelines if present.

MIG-OPS-BP-9.1 Determine if your current CI/CD pipeline works on AWS

This BP applies to the following best practice areas: Prepare

Implementation guidance

Suggestion 9.1.1: Assess your existing pipeline tools.

An assessment of each tool in the pipeline is required to verify that it has capabilities to work with the AWS services required. This assessment drives the requirement to either build the migration plan and updates required to support the new target AWS landing zone, or the selection of new tools to achieve the desired outcome.

MIG-OPS-BP-9.2 Provision resources through infrastructure as code (IaC) templates

This BP applies to the following best practice areas: Prepare

Implementation guidance

Suggestion 9.2.1: Consider using AWS CloudFormation.

It is recommended that all resources required for each application are provisioned through IaC templates. These templates could be written in AWS CloudFormation or another IaC tool. This approach keeps configuration with the application code so it can be managed centrally. With the rehost migration pattern, AWS MGN takes care of the source workload migrations, including all application data and configuration present on the source systems. However, the infrastructure components and configuration around the migrated application server on the Amazon EC2 platform still need to be deployed (for example, VPCs, subnets, network security groups, network access control lists, and load balancers).

Suggestion 9.2.2: Consider using a configuration management tool.

Maintain an accurate and complete record of all your cloud workloads, their relationships, and configuration changes over time. For more detail, see Configuration management.

Security

The security pillar encompasses the ability to protect information, systems, and assets to take advantage of cloud technologies designed to improve your security. It provides an overview of design principles, best practices, and questions to ask. This pillar helps you understand how to apply the shared responsibility model during the three migration phases. You can find additional prescriptive guidance on implementation details in the security pillar whitepaper.

At AWS, security is our top priority. As an AWS customer, you benefit from a data center and network architecture that are built to meet the requirements of the most security-sensitive organizations. Security is a shared responsibility between AWS and you. The shared responsibility model describes this by using the terms security *of* the cloud and security *in* the cloud:

- **Security of the cloud**: AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services. The effectiveness of our security is regularly tested and verified by third-party auditors as part of the AWS compliance programs
- **Security in the cloud**: Your responsibility is determined by the AWS services you use. You are also responsible for other factors including data management, your organization's security and compliance requirements, and applicable laws and regulations.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

It's important to conduct a security review of the discovery tool used to assess on-premises inventory and understand how they work to ensure they don't introduce any vulnerabilities. During this phase, it's also key to review your current security tools and map them to their equivalents on AWS and to identify your compliance framework.

MIG-SEC-01: Have you performed a security review of the discovery tool you plan to use to assess your on-premises inventory?

Organizations have to meet different security and compliance standards. Ensure you fully understand the impact of any discovery tool against your security posture by assessing the risk profile of the discovery tool, how the data about on-premises environment is collected, where the data is stored, and how the stored data is secured.

MIG-SEC-BP-1.1 Understand the security credentials needed by the discovery tool

This BP applies to the following best practice area: Identity and access management

Implementation guidance

Suggestion 1.1.1: Determine the required discovery tools and techniques.

Discovery tools, whether AWS native or partner solutions, typically leverage two types of discovery methodologies: agent-based discovery or agentless discovery. Agent-based tools require access to the workloads to install the discovery agent for data collection. Agentless discovery requires permission to scan the network and identify workloads. Identify the least privilege model and apply accordingly.

For more detail, see the following:

- AWS Application Discovery Service FAQs
- Discovery, Planning, and Recommendation migration tool details

Suggestion 1.1.2: Identify and safeguard credentials needed by discovery tools.

Follow the principle of least privilege, granting only the necessary permissions to discovery tools and their associated AWS Identity and Access Management (IAM) roles or users. Use a credential management system, such as AWS Secrets Manager, to limit sharing and proliferations of credentials. Additionally, limit the use of long-term credentials when possible.

For more detail, see the following:

- Least privilege
- AWS Secrets Manager

MIG-SEC-BP-1.2 Understand how the discovery tool works

This BP applies to the following best practice areas: Infrastructure protection

Implementation guidance

Suggestion 1.2.1: Understand the network requirements for discovery tools.

Discovery tools, whether AWS native or partner solutions, typically leverage 2 types of discovery methodologies: Agent based discovery or agentless discovery. These 2 methods use different ports and protocols to collect information. Once you choose a discovery tool, study the documentation to understand the networking and potential security and availability considerations. For example, if the tool uses a non-standard port, is that port reachable on the assets you want to assess.

For more detail, refer to the following information:

- Agent and Agentless discovery tools
- Discovery, Planning, and Recommendation migration tool details

MIG-SEC-BP-1.3 Understand the discovery tool's data security and apply appropriate controls

This BP applies to the following best practice areas: Data protection

Implementation guidance

Suggestion 1.3.1: Understand what data the discovery tool collects.

Discovery tools collect various pieces of data, such as server names, IP addresses, allocated and utilized resources, network ports, and applications installed on the machine. Organizations should try to limit the collection of data to only the minimum data types necessary and relevant to support migration planning.

Suggestion 1.3.2: Understand where the discovery data is stored.

Collected data from discovery tools is typically stored locally within a customer's data center or sent over the network directly to the tool vendor as a SaaS model. Understand the tool's data storage capabilities and vendor's security controls to verify that they align to your data management, handling, and storage policies. You can also collect the data and store it locally within your data center and only share redacted data with AWS or partners for analysis.

Suggestion 1.3.3: Understand how the data is encrypted in transit and at rest.

Different discovery tools come with different security controls when it comes to how data is encrypted in transit and at rest. Ensure this meets your organization's security policy requirements.

Suggestion 1.3.4: Get the necessary approval from your security team in your organization to install and use the discovery tool.

After deciding the right discovery tool to use, work with the security team in your organization to get the necessary approval to start using the tool. This process make take time, so plan accordingly.

For more detail, refer to the following information:

- Selecting the discovery tool for your cloud migration

MIG-SEC-02: Have you reviewed and mapped your existing security tools and controls to equivalent AWS services?

Customers moving into AWS can leverage a comprehensive selection of AWS cloud-native security services. Before migrating to AWS, it is important to map your on-premise security tools and controls to those available in your AWS environment. This includes controls like identity and access management, perimeter security, encryption tools, network security, data security, vulnerability management tools, code scanning tools, and threat detection.

MIG-SEC-BP-2.1 Perform a tools mapping exercise

This BP applies to the following best practice areas: Infrastructure security

Implementation guidance

Suggestion 2.1.1: Understand the AWS Shared Responsibility Model.

Identify the security controls of resources hosted in AWS, keeping in mind the AWS Shared Responsibility Model and how this model shifts based on the AWS service being used. While AWS is responsible for the security *of* the cloud, and the customer is responsible for security *in* the cloud, the customer needs to understand how both sides of the shared responsibility model align to any compliance and regulatory requirements. Review the security functionality and configuration options of individual AWS services within the security chapters of AWS service documentation. Customers can view a variety of security and compliance reports created by third-party auditors by using AWS Artifact.

Suggestion 2.1.2: Map network security tools.

Map network security tools such as firewalls, IDS/IPS, deep packet inspection, and web application firewalls, and understand AWS native capabilities. Understand the difference between networking in AWS and on-premise data centers, as it is important to apply this to your design and tools selection. AWS native services can be complemented with AWS Partner services of choice through the AWS Marketplace to reach a desired security posture.

For more detail, see the following:

- Network and Application Protection
- Detection

Suggestion 2.1.3: Map operating system (OS) level security tools, including third-party tools.

Customers should understand if they can continue to use the existing OS level security tools in their self-managed EC2 instances or containers. Understand the technical and licensing limitations of porting those tools to AWS. For more detail, see AWS Systems Manager FAQs.

Suggestion 2.1.4: Map on-premises vulnerability management controls.

Map your on-premises vulnerability management security control policies and requirements to AWS workload architectures, service capabilities, and controls.

For more detail, see the following:

- Amazon Inspector
- SEC06-BP01 Perform vulnerability management
- Application Security partner tools

Suggestion 2.1.5: Map your on-premises data security control policies and requirements to AWS data and storage architectures, service capabilities, and controls.

Map data security tools and services, such as certificate management tools, key management, TLS certificates, encryption tools, and AWS Secret Manager to AWS service capabilities and controls. For more detail, see Data Protection services.

MIG-SEC-03: Do you have an established compliance framework?

Customers have distinct risk and compliance requirements, based on factors such as industry, geographical location, customer base, and governmental and regulatory authorities. The AWS Compliance Program helps customers understand the robust controls in place at AWS to maintain security and compliance of the cloud. By tying together governance-focused, audit-friendly service features with applicable compliance or audit standards, AWS Compliance Enablers build on traditional programs, helping customers to establish and operate in an AWS security control environment.

MIG-SEC-BP-3.1 Understand, establish, and implement compliance framework

This BP applies to the following best practice areas: Security foundations

Implementation guidance

Suggestion 3.1.1: Identify your compliance requirements.

IT standards that AWS complies with are broken out by Certifications and Attestations, Laws, Regulations and Privacy, and Alignments and Frameworks. Compliance certifications and attestations are assessed by third-party, independent auditors and result in a certification, audit report, or attestation of compliance. AWS customers remain responsible for complying with applicable compliance laws, regulations, and privacy programs. Compliance alignments and frameworks include published security or compliance requirements for a specific purpose, such as a specific industry or function.

For more detail, see the following:

- AWS Compliance Programs
- AWS Customer Compliance Guides

Suggestion 3.1.2: Determine if you operate and store data in multiple countries and identify any geography-based compliance requirements.

An organization's compliance needs can vary depending on the city, state, country, or even Region. Work closely with your organization's compliance and legal teams to understand any regulatory or other compliance requirements, including data residency and digital sovereignty requirements.

For more detail, see the following:

- Compliance FAQs

- Compliance Resources
- Digital Sovereignty at AWS

Suggestion 3.1.3: Familiarize yourself with the compliance postures of the AWS services that make up your solution's architecture.

Security and compliance are a shared responsibility between AWS and the customer. Depending on the services deployed, this shared model can help relieve the customer's operational burden. AWS is responsible for compliance of underlying service capabilities, while the customer is responsible for compliance of the specific implementations. Use AWS Artifact to look at compliance reports for various AWS services and assess what you as a customer are responsible for meeting in terms of compliance and what AWS as a service provider is responsible for.

For more detail, see the following:

- AWS Services in Scope by Compliance Program
- AWS Shared Responsibility Model
- AWS Artifact

Mobilize

During the mobilize phase of the migration, you plan for your authentication and authorization systems to ensure secure access to your migrated workloads. This phase also involves building your AWS environment in alignment with AWS security foundations. Establishing a secure connection between on-premises and AWS is essential for safely migrating workloads to AWS. This includes establishing policies and tools for data encryption at rest and in transit. Furthermore, it's important to consider any third-party integrations and align them with the overall security strategy. These steps collectively enhance the security resilience of the migration process and prepare the infrastructure for a successful transition to AWS.

MIG-SEC-04: Do you have an established standard for authentication and authorization?

AWS Identity and Access Management (IAM) provides fine-grained access control across the entire AWS platform. You can use IAM to specify who or what can access which services and

resources, and under which conditions. IAM policies let you manage permissions to your workforce and systems to ensure least privilege permissions. Least privilege is an AWS Well-Architected Framework best practice for building securely in the cloud.

MIG-SEC-BP-4.1 Implement strong identity and least privilege principles

This BP applies to the following best practice areas: Identity and access management

Implementation guidance

Suggestion 4.1.1: Protect and limit the use of the AWS account root user.

It's vital to ensure strong security measures for your AWS account's root user, treating its credentials with the utmost confidentiality and limitation. You should regard your root user credentials with the same seriousness as vital personal information, deploying them only when required.

For a comprehensive guide on the best practices surrounding the AWS root account, see Root user best practices for your AWS account.

Suggestion 4.1.2: Assess how user identities are managed and authenticated in AWS.

In the migration process, the selection of a suitable identity provider (IDP) is essential. This choice determines how smoothly and securely you can connect to the cloud. When migrating to AWS, it's crucial to evaluate and optimize how user identities are managed and authenticated to pick the most appropriate option based on your long-term authentication and authorization requirements:

- **AWS Identity and Access Management (IAM):** Define distinct user roles and permissions tailored to AWS resources. Consider the enhanced security of AWS multi-factor authentication for high-priority accounts. IAM's federated capabilities integrate effortlessly with established identity systems, like Microsoft Active Directory. Federation should be leveraged in place of IAM users whenever feasible. This allows users to authenticate using their existing credentials, streamlining the authentication process and simplifying the account management provisioning and de-provisioning processes.
- **Directory Service:** Facilitate your migration by integrating with corporate directories, enhancing user experience and reducing operational burdens.
- **AWS IAM Identity Center:** Centrally coordinate workforce access, a pivotal asset during the migration phase. AWS IAM Identity Center is the preferred method for organizations to federate existing workforce identity stores.

- **Amazon Cognito:** Provides customer identity and access management to applications and workloads.
- **External identity providers**: While adopting AWS, integrate with existing IDPs to establish connections. External identity providers can easily integrate directly with AWS IAM, AWS IAM Identity Center, and Amazon Cognito. Manual configuration may be required to provide optimal connectivity. Regularly synchronize identities to maintain accurate access controls.

For more detail, see the following:

- Identity and Access Control
- Require human users to use federation with an identity provider to access AWS using temporary credentials
- Security best practices in IAM

Suggestion 4.1.3: Implement a strong privileged access management program and controls.

A key security consideration for the enterprise is monitoring and administrating elevated access, often known as privileged access, for business-critical applications that are running in the AWS Cloud. You need to have a process to request, fulfill, certify, and govern privileged assets in the cloud to maintain privileged access management (PAM). Based on your compliance requirements, you may need to limit the privileged access to a certain group of resources or for a specific period of time.

For more detail, see the following:

- AWS Marketplace for PAM solutions
- Temporary elevated access management with IAM Identity Center

MIG-SEC-05: Have you built your AWS environment following the AWS recommended security foundations?

As you move into the mobilize phase of the migration journey, you build the foundational components, such as AWS accounts and networking and security, before the workloads move

to AWS. We refer to this as building a landing zone (not to be confused with AWS Landing Zone Service, which is part of AWS Control Tower).

MIG-SEC-BP-5.1 Implement AWS multi-account structure

This BP applies to the following best practice areas: Security foundations

Implementation guidance

Suggestion 5.1.1: Understand and design AWS multi-account structure for isolation boundaries at the AWS account, VPC, business unit, and environment levels.

As you adopt AWS, we recommend that you determine how your business, governance, security, and operational requirements can be met in AWS. Use of multiple AWS accounts plays an important role in how you meet those requirements. The use of multiple accounts allows for benefits like group workloads based on business purpose and ownership.

Apply distinct security controls by environment, constrain access to sensitive data, and limit scope of impact from adverse events.

For more detail, see the following:

- Building a landing zone
- The AWS Security Reference Architecture
- Best practices for AWS Control Tower administrators
- Security in AWS Control Tower

Suggestion 5.1.2: Take note of AWS service quotas per AWS account.

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased. As you scale, AWS multi-account strategy quotas play an important role in designing multi-account strategy and workload grouping strategy.

MIG-SEC-06: Have you established secure connectivity in preparation for migrating workloads to AWS?

There are many different mechanisms available for connectivity between a customer's data center and AWS. Which solution you choose is dependent on your use case and requirements. For all solutions, secure connectivity between your on-premises infrastructure and AWS is paramount during the migration process. This involves the use of robust strategies for maintaining data confidentiality and integrity in transit.

MIG-SEC-BP-6.1 Establish secure connectivity to AWS

This BP applies to the following best practice areas: Data protection

Implementation guidance

Suggestion 6.1.1: Establish secure data transmission capabilities between on-premise networks and AWS

Create secure data transmission utilizing virtual private networks (VPNs) or dedicated private connections to establish secure network connections for your migration. These connections keep the data confidential and maintain its integrity as it moves between your on-premises environment and AWS. If your organization has compliance requirements for encryption in transit, implement VPN or encryption for connectivity between your data center and AWS. This provides secure transmission of data during the migration process. You might consider using AWS Transit Gateway in conjunction with a VPN to securely connect your on-premise datacenters to your VPCs.

For more detail, see the following:

- Site-to-Site VPN
- AWS Transit Gateway

Suggestion 6.1.2: Use AWS Direct Connect for large bandwidth and dedicated connectivity

Use AWS Direct Connect for stable connectivity for large data movement with stable bandwidth and low latency network connectivity. It provides a dedicated, private network connection from your premises to AWS, which is crucial for large workload migrations.

Suggestion 6.1.3: Use AWS PrivateLink to limit exposure between VPCs and AWS services. Establish connectivity between VPCs and AWS services without exposing data to the internet using AWS PrivateLink. AWS Application Migration Service interacts with interface VPC endpoints to establish a private connection between your VPC and AWS Application Migration Service.

MIG-SEC-BP-6.2: Establish network security controls

This BP applies to the following best practice areas: Infrastructure protection and Data protection

During the migration process to AWS, it's important to ensure robust network protection, including the implementation of intrusion detection and prevention systems (IDS/IPS), as well as OSI layer 4 to layer 7 security. AWS and the Amazon Partner Network offer a variety of services that can support these requirements.

Implementation guidance

Suggestion 6.2.1: Enable layer 7 Security with AWS Web Application Firewall (WAF) to protect your web applications from common web exploits.

AWS WAF allows you to control how traffic reaches your applications by creating security rules that block common attack patterns, such as SQL injection or cross-site scripting (XSS). Use AWS Shield for managed Distributed Denial of Service (DDoS) protection. AWS Shield Advanced provides additional DDoS protections and capabilities.

Suggestion 6.2.2: Use VPCs and network segmentation.

Use the appropriate network controls to isolate your applications appropriately. Virtual Private Clouds (VPCs) allow you to create logically isolated virtual networks. Within a VPC, you can use security groups (SGs) and network access control lists (NACLS) that implement inbound and outbound traffic rules and ensure appropriate segmentation. For more detail, see Zero Trust.

Suggestion 6.2.3: Explore IDS/IPS solutions in the AWS Marketplace.

Explore third-party IDS/IPS solutions offered in the AWS Marketplace. Many of these solutions offer additional security features and capabilities that can complement those provided by AWS services. For more detail, see AWS Network Firewall.

Suggestion 6.2.4: Identify anomalous network behavior from migrated workloads using Amazon GuardDuty.

Amazon GuardDuty monitors your accounts and various workloads to identify malicious and anomalous behaviors, including monitoring network and DNS traffic. When migrating workloads such as virtual machines and containers, Amazon GuardDuty can detect and alert you if those workloads are attempting to use your network for potentially malicious or unauthorized activities.

MIG-SEC-07: Do you have policies and tools defined for data encryption at rest during and after migration?

Data at rest represents any data that you persist in non-volatile storage for any duration in your workload. This includes block storage, object storage, databases, archives, IoT devices, and any other storage medium on which data is persisted. Protecting your data at rest reduces the risk of unauthorized access, when encryption and appropriate access controls are implemented. AWS provides robust and scalable encryption solutions for both data at rest and in transit to help you meet your data security requirements and compliance needs.

MIG-SEC-BP-7.1 Establish security controls for protecting data at rest

This BP applies to the following best practice areas: Data protection

Implementation guidance

Suggestion 7.1.1: Classify your data based on its sensitivity

Understand what data is sensitive, confidential, or public. This helps in applying appropriate security controls. To effectively manage risk, organizations should consider classifying data by working backward from the contextual use of the data, and creating a categorization scheme that takes into account whether a given use case results in significant impact to an organization's operations (for example, if data is confidential, it needs to have integrity, and it needs to be available). Customers also need to take into account their regulatory and compliance requirements for protection of data like GDPR.

Suggestion 7.1.2: Use AWS Key Management Service (KMS) for protecting data at rest.

Protect data at rest by using AWS Key Management Service (KMS) to create and control the cryptographic keys used to encrypt your data. Additionally, use the built-in encryption capabilities of services like Amazon S3, Amazon EBS, Amazon RDS, and AWS Lambda for protecting data at rest.

Suggestion 7.1.3: Use AWS CloudHSM when compliance dictates.

If compliance requirements dictate the need for hardware-based cryptographic key storage, commonly referred to as hardware security models (HSMs), consider AWS CloudHSM. HSMs provided by CloudHSM are FIPS 140-2 level 3 certified.

Suggestion 7.1.4: Use strong IAM policies for key management.

Establish granular IAM policies that explicitly delineate permissions for activities related to data encryption at rest. Verify that only trusted roles or users can decrypt the data or manage encryption keys, further bolstering the security of your data during and after migration.

For more detail, see the following:

- AWS IAM Policy Best Practices
- AWS Key Management Service (KMS) Best Practices

MIG-SEC-08: Have you identified and applied application security controls?

Protecting applications, hosting environments, and detecting irregular behavior is critical to a secure cloud environment. Customers transitioning to AWS have the advantage of tapping into a comprehensive array of AWS cloud-native application security services and work on existing applications to match the overall security posture.

MIG-SEC-BP-8.1: Establish application layer security controls

This BP applies to the following best practice areas: Application security

Implementation guidance

Suggestion 8.1.1: Implement application layer vulnerability scanning.

AWS emphasizes the importance of application security through comprehensive practices such as regular updates, vulnerability scanning, penetration testing, and secure coding principles. Conduct regular scanning and testing to identify weaknesses within AWS applications and infrastructure. Use AWS tools like Amazon Inspector for streamlined security assessments.

Suggestion 8.1.2: Implement full-lifecycle secure coding practices and supporting tools.

Implement secure coding practices for applications within AWS, leveraging code review and proper methodologies. Use AWS services such as AWS CodeGuru for enhanced code quality insights and security. Use Amazon CodeWhisperer to provide additional security context and recommendations within your IDE as you write your application code. For more detail, see Building a secure CICD pipeline.

Suggestion 8.1.3: Perform threat modeling.

Identify and prioritize risks using a threat model. Use a threat model to identify and maintain an up-to-date register of potential threats. Prioritize your threats and adapt your security controls to prevent, detect, and respond. Revisit on a recurring basis and maintain this in the context of the evolving security landscape.

Suggestion 8.1.4: Implement customer identity and access management for your applications that target non-workforce users.

Implement a customer identity and access management (CIAM) solution that allows your customers and end-users (like non-employee accounts) to access your application securely. Use Amazon Cognito, which is designed to handle the scale and full lifecyle of CIAM account management, or consider various partner CIAM solutions in the AWS Marketplace. Additionally, use Amazon Verified Permissions (AVP) for scalable, fine-grained permissions management and authorization service for custom applications built by you.

MIG-SEC-BP-8.2: Optimize application security with AWS Application Migration Service

This BP applies to the following best practice areas: Application security

Implementation guidance

Suggestion 8.2.1: Automate the migration and conversion processes using AWS-provided services.

Use the AWS Application Migration Service (MGN) to convert source servers to run natively on AWS, streamlining the conversion and migration processes and minimizing manual errors. This provides a seamless transition through a tested non-interactive and secure conversion, introduces automation for post-migration configurations, and optimizes applications to benefit from robust AWS infrastructure.

Suggestion 8.2.2: Modernize and enhance your application.

During migration, take advantage of the service's in-built options such as disaster recovery, OS or license conversion, and cloud-native capabilities. This ensures applications are not just migrated but also modernized to meet contemporary security and operational standards.

MIG-SEC-9: Do you have a data backup and disaster recovery strategy during migration?

Data backups are an essential element of data security. In the context of migration to AWS, planning for data backup and disaster recovery is critical to assure business continuity and protect against data loss. These concepts are covered in more details in the Reliability pillar of this document. AWS provides several services that can help with data backup and restoration, as well as managing and testing disaster recovery plans.

MIG-SEC-BP-9.1: Establish a data backup and restore strategy

This BP applies to the following best practice areas: Data protection

Implementation guidance

Suggestion 9.1.1: Implement and test backup and recovery capabilities.

Use AWS Backup to create backup plans, which define when and how often backups are created and how long they're stored. Regularly test backup restoration to test that your backup strategy is effective and backups are usable in case of data loss or system failure.

Suggestion 9.1.2: Audit and validate your backup requirements.

Use AWS Backup Audit Manager to audit the compliance of your AWS Backup policies against controls you define. Audit and identify issues regarding backup schedules, which resources are being backed up, and any non-compliance against the controls you set up can be reported and leveraged for remediation.

MIG-SEC-BP-9.2: Establish a Disaster recovery plan

This BP applies to the following best practice areas: Data protection and Infrastructure protection

Implementation guidance

Suggestion 9.2.1: Develop and test a disaster recovery plan and capabilities.

Leverage AWS Elastic Disaster Recovery to minimize downtime and data loss with fast, reliable recovery of physical, virtual, and cloud-based servers into AWS. Use the AWS Well-Architected Framework Reliability Pillar to design, deploy, and manage workloads and align them with disaster recovery strategies and requirements.

MIG-SEC-10: Have you established monitoring controls with the right set of tools?

Establishing robust monitoring controls for security is essential to detect and respond to potential security threats in your AWS environment. By implementing comprehensive monitoring controls, you can gain visibility into activities, monitor for unusual behavior, and proactively identify security incidents.

MIG-SEC-BP-10.1: Validate and use AWS native monitoring tools.

This BP applies to the following best practice areas: Incident response

Implementation guidance

Suggestion 10.1.1: Develop and deploy a comprehensive logging strategy

An effective logging strategy is a cornerstone of any successful migration to AWS. By leveraging the right combination of AWS and third-party tools, you can maintain full visibility into your infrastructure and ensure your operations are running smoothly.

For more detail, see the following:

- Getting started with AWS CloudTrail tutorials
- Setting Up AWS Config with the Console
- Getting set up (Amazon CloudWatch)
- Analyze Network Traffic of Amazon Virtual Private Cloud (VPC) by CIDR blocks
- Considerations for the security operations center in the cloud: deployment using AWS security services
- Logging strategies for security incident response

MIG-SEC-BP-10.2: Explore cloud native AWS partner monitoring tools

This BP applies to the following best practice areas: Incident response

Implementation guidance

Suggestion 10.2.1: Deploy application monitoring capabilities.

Alongside AWS tools such as AWS X-Ray, consider third-party partner tools which provide application-level insights and monitoring on AWS. They can supplement AWS services and help create a more holistic monitoring strategy tailored to your business needs.

MIG-SEC-11: Do you have any third-party integrations?

When integrating third-party services into your AWS migration, it's crucial to review the security features, permissions, and data handling practices of these services to maintain a secure and compliant migration process. Review their security practices and verify that they align with your organization's security requirements.

MIG-SEC-BP-11.1: Perform third-party integration due diligence

This BP applies to the following best practice areas: Security foundations

Implementation guidance

Suggestion 11.1.1: Review third-party integration patterns and security practices.

When reviewing third-party integration patterns, conduct thorough due diligence and consider engaging with the vendor directly to discuss their security practices and address any specific security concerns you may have. Additionally, consult the AWS Shared Responsibility Model to understand the division of security responsibilities between AWS and third-party service providers.

Review the following checklist in regard to third-party integrations:

1. **Authentication and authorization:** The third-party should supports secure mechanisms like multi-factor authentication (MFA) and role-based access control (RBAC).
2. **Data encryption**: Confirm encryption both in transit (using TLS) and at rest with robust algorithms.
3. **Compliance and certifications**: Assess adherence to standards like SOC 2, ISO 27001, and other relevant industry certifications.
4. **Data privacy and residency**: Verify that data handling aligns with organizational privacy policies and legal regulations.
5. **Logging and monitoring:** Review capabilities for security analysis and incident response visibility.
6. **Security incident response:** Understand incident management, customer communication, and resolution speed.
7. **Third-party audits and assessments:** Request information on security tests and independent reviews undergone.

8. **Data backup and recovery**: Check mechanisms against data loss.
9. **Service-level agreements (SLAs):** Check that they fulfill organizational needs in terms of availability, performance, and security.
10. **Integration with AWS services:** Verify that AWS integration adheres to security best practices.
11. **Vendor reputation and support:** Research vendor credibility, reviews, and their support effectiveness.
12. **Continual security updates:** Confirm timely vulnerability addressing and update provision.

Migrate

As the migration progresses, security remains top priority. It's essential to understand the data security and compliance, establish a secure credential mechanism, and have robust mechanisms in place for monitoring, identifying, and responding to any security incidents or anomalies. This involves not only employing the right tools, but also training teams and preparing them to respond effectively. In this phase, you also implement mechanisms to protect the migrated resources of compute, network, databases and applications.

MIG-SEC-12: Have you performed a security review of your migration tools?

When using AWS migration tools to move data and applications from on-premises or other cloud environments to AWS, it's essential to consider security throughout the migration process. This section contains key security considerations when using AWS migration tools.

MIG-SEC-BP-12.1: Understand the data security and compliance

This BP applies to the following best practice areas: Data protection

Implementation guidance

Suggestion 12.1.1: Verify data encryption and integrity.

Data transferred during the migration process should be encrypted in transit. AWS migration tools, such as AWS DataSync, AWS Database Migration Service, and AWS Application Migration Service support TLS encryption to secure data as it moves between your on-premise and AWS environments. Verify the integrity of data during migration by using cryptographic hashes or checksums. This helps detect any unauthorized changes or tampering during transit.

Suggestion 12.1.2: Implement secure credential management.

Safeguard access credentials used by migration tools. Follow the principle of least privilege, granting only the necessary permissions to migration tools and their associated IAM roles or users. Use a credential management system, such as AWS Secrets Manager, to limit sharing and proliferations of credentials. Also, limit the use of long-term credentials when possible, and use IAM Roles Anywhere to prevent the need for storage secrets.

MIG-SEC-13: How do you detect and investigate security events?

Migrating your workloads from on-premises to AWS also requires you to update your security detection and investigation processes. Detective controls, crucial for governance and compliance, help identify potential threats and support threat identification and response. These controls include asset inventory for informed decision-making (you must know what resources you have so you know how to protect them) and internal auditing of information systems to align practices with policies and correctly set automated alerting notifications. Such controls are pivotal in pinpointing and understanding anomalous activity.

MIG-SEC-BP-13.1: Understand AWS service capabilities for event detection and investigation

This BP applies to the following best practice areas: Incident response

Implementation guidance

Suggestion 13.1.1: Configure service and application logging.

Retain security event logs from services and applications, a fundamental principle for audit, investigations, and operational use cases. This retention must be in line with governance, risk, and compliance (GRC) requirements and industry-specific standards. Ahead of security investigations, capture logs to reconstruct AWS account activity. Select log sources pertinent to your workloads based on your business use cases and any regulatory or compliance requirements you may have. Establish a logging trail for each AWS account, and in each region using AWS CloudTrail or an AWS Organizations trail, store logs in a dedicated and centralized Amazon S3 bucket.

Suggestion 13.1.2: Analyze logs, findings, and metrics centrally.

As you migrate to AWS, security operations teams should use advanced data search and analytics tools, especially given the volume of data from complex architectures and applications. Reliance on manual data analysis is not suited for the majority of most customer security requirements and could impact your ability to investigate potential security issues in a timely manner. Use services such as AWS Security Hub to aggregate security events and alerts from other security services, evaluate your security posture, and automate remediations. Amazon Detective can help you investigate security events by contextualizing events in relationship across several service and log sources.

For more detail, see the following:

- Collect, analyze, and display Amazon CloudWatch Logs in a single dashboard with the Centralized Logging on AWS solution
- Amazon Security Lake
- AWS Security Competency Partners
- Searching and analyzing logs in CloudWatch

Suggestion 13.1.3: Automate and implement the response to security events.

Using automation to investigate and remediate events reduces human effort and error, quickens responses, and allows you to scale investigation capabilities. Regular reviews help you tune automation tools and continually iterate. AWS provides guidance on how this can be achieved using a combination of AWS security services and patterns. Consider the impact to the availability of your workloads carefully when implementing security automations, as you may not be able to fully automate all remediation activities.

For more detail, see Threat management in the cloud: Amazon GuardDuty and AWS Security Hub.

MIG-SEC-14: Do you have security incident response capabilities in place?

To migrate your security incident response capabilities from on-premises to AWS, careful planning and adopting cloud-native practices are essential. It's crucial to understand AWS security incident response concepts, prepare and educate your teams, and identify automation-based remediation methods for faster and more consistent responses. Additionally, it is key to understand your compliance and regulatory requirements for your security incident response program, and how they relate to building a security incident response program to fulfill those requirements.

MIG-SEC-BP-14.1: Understand AWS best practices for incident response

This BP applies to the following best practice areas: Incident response

Implementation guidance

Suggestion 14.1: Develop and test an incident response plan.

The first document to develop for incident response is the incident response plan, which is designed to serve as the foundation for your incident response program. It typically includes:

1. **An incident response team overview**: Outlines the goals and functions of the incident response team.
2. **Roles and responsibilities**: Lists stakeholders and their incident roles.
3. **A communication plan**: Details contact information and how to communicate during incidents. Emphasizes the best practice of having out-of-band communication as a backup. AWS Wickr can be used as a secure out-of-band communications channel.
4. **Phases of incident response and actions to take**: Enumerates response phases (like detect, analyze, eradicate, contain, and recover) and associated high-level actions.
5. **Incident severity and prioritization definitions**: Explains incident severity classification, prioritization, and their impact on escalation procedures.

While these sections are common, each organization's plan is unique and should be tailored accordingly.

MIG-SEC-15: How do you protect your compute and network resources in AWS?

Compute resources that support your workloads require multiple layers of defense to help protect from external and internal threats. Compute resources include EC2 instances, containers, AWS Lambda functions, database services, and IoT devices.

MIG-SEC-BP-15.1: Protect your network resources

This BP applies to the following best practice areas: Infrastructure protection

Implementation guidance

Suggestion 15.1.1: Create a layered networking architecture with isolation boundaries.

Components such as Amazon Elastic Compute Cloud (Amazon EC2) instances, Amazon Relational Database Service (Amazon RDS) database clusters, and AWS Lambda functions that share reachability requirements can be segmented into layers formed by subnets. Consider deploying serverless workloads, such as Lambda functions, within a VPC or behind an Amazon API Gateway. AWS Fargate tasks (and all other compute workloads) that have no need for internet access should be placed in subnets with no route to or from the internet. This layered approach mitigates the impact of a single layer misconfiguration, which could allow unintended access. For AWS Lambda, you can run your functions in your VPC to take advantage of VPC-based controls. Regardless of your workload type, it is imperative to understand the data flow between layers of your application, and implement the controls necessary to restrict ingress and egress traffic to only authorized users and services.

For more detail, see the following:

- Well-Architected Lab - Automated Deployment of VPC
- Amazon VPC | AWS Security Blog

Suggestion 15.1.2: Create centralized policies for network security.

Use AWS Firewall Manager to centrally configure and manage your network security policies across all accounts and applications in your organization, simplifying the administration of AWS WAF, AWS Shield Advanced, AWS Network Firewall, and Amazon VPC security groups.

With AWS Network Firewall, you can define firewall rules that provide fine-grained control over network traffic. Network Firewall works together with AWS Firewall Manager, so you can build policies based on Network Firewall rules and then centrally apply those policies across your virtual private clouds (VPCs) and accounts.

MIG-SEC-16: What are your authentication and authorization processes for applications and databases?

When migrating databases and applications to AWS, it's essential to have robust authentication and authorization controls to secure access to sensitive data. AWS offers several services and best practices to achieve this.

MIG-SEC-BP-16.1: Manage authentication for applications and databases

This BP applies to the following best practice areas: Identity and access management

Implementation guidance

Suggestion 16.1.1: Consider more secure database authentication and authorization methods.

When moving databases to AWS managed services, such as RDS (Relational Database Service) and Aurora databases, you can enable IAM database authentication. This allows you to use IAM roles instead of static or hard coded credentials to authenticate and access the databases, improving security by removing the need to manage database passwords. Define database-level roles and permissions to control access to specific tables, views, and stored procedures within your databases.

Suggestion 16.1.2: Consider stronger application authentication and authorization mechanisms for applications.

Implement strong authentication mechanisms for your applications. Use protocols like OAuth 2.0 or OpenID Connect for web applications, and consider token-based authentication for APIs. Implement role-based access control (RBAC) within your applications. Map AWS IAM roles to application roles, and control access to application features and data based upon business need. Verified Permissions can also be leveraged to help manage permissions and fine-grained authorizations in applications.

Suggestion 16.1.3: Use AWS Secrets Manager for storing credentials.

Use AWS Secrets Manager to manage, retrieve, and rotate database credentials, application credentials, OAuth tokens, API keys, and other secrets throughout their lifecycles. Secrets Manager helps you improve your security posture, because you no longer need hard-coded credentials in application source code. Storing the credentials in Secrets Manager helps avoid possible compromise by anyone who can inspect your application or the components.

Reliability

The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle. The reliability pillar provides guidance to help you apply best practices, current recommendations in the design, delivery, and maintenance for implementing

reliable workloads on AWS. This paper provides an overview of best practices and questions. You can find prescriptive guidance on implementation in the reliability pillar whitepaper.

A reliable cloud migration strategy involves planning, risk assessment and contingency measures to address potential disruptions. Factors like data integrity, network stability, and application availability play crucial roles in determining the reliability of the migration process. Plan to migrate your workload to AWS based on existing reliability requirements, during the migration activities, and after the migration cut-over.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

The Assess phase involves evaluating the current state of the workloads that are targeted for the migration. To achieve this, we will focus on assessing the existing workloads for any potential points of failure during the migration process.

MIG-REL-01: Do you have any existing compliance requirements around service availability or service-level agreements (SLA) that apply to applications within the migration scope?

Existing applications have current service levels which must be maintained during migration. During migration assessment, it is important to understand the existing availability requirements, and then define the migration strategy and target architecture.

MIG-REL-BP-1.1: Define SLAs across all applications or environments (like production, development, or test) and confirm them with your business team

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 1.1.1: Evaluate the unique aspects of your applications to understand if you have different availability requirements for each application.

Define goals for each application based on availability.

MIG-REL-BP-1.2: Define and automate runbooks and communicate them to your teams

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 1.2.1: Prepare, document and validate procedures for your workload to minimize the disruption of your workload during events.

It is recommended to automate runbook procedures so runbook activities are performed consistently. For more detail, see OPS07-BP03 Use runbooks to perform procedures.

MIG-REL-BP-1.3: Map AWS Global Infrastructure to your business SLAs before migrations starts

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 1.3.1: Understand AWS Global Infrastructure terminology and definitions.

If you operate multiple datacenters on-premises today, how does this map to AWS infrastructure and your existing availability requirements? **Suggestion 1.3.2** Identify the services you plan to use when you migrate and compare the AWS Service Level Agreements to your existing business SLAs.

Your existing SLAs may need to be updated based on AWS Service Level Agreements.

MIG-REL-BP-1.4: Select tools to monitor SLAs and notify you in case thresholds are exceeded

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 1.4.1: Reduce communication to on-premises monitoring tools.

If you choose to use existing monitoring tools on your migrated workloads, you should optimize data egress to systems which are remaining on-premises. This is achieved differently by different tools, but a common method is to deploy a collector within AWS that optimizes communication.

When migrating to AWS, there may be agents which are no longer needed (for example, VMware Tools and tools for physical hardware monitoring). Use custom post-launch actions to remove these agents during migration with AWS Application Migration Service.

Suggestion 1.4.2: Use managed services to reduce operational overhead and save licensing costs.

Before migrating, assess if changing monitoring tools for AWS Managed Services like Amazon Cloudwatch and AWS Systems Manager could reduce the overhead of running these tools and the licensing costs from those tools.

Suggestion 1.4.3: Monitor networking links during migration.

Measure the additional migration related network traffic and prevent this traffic from affecting business applications. For example, if you are using AWS Direct Connect between your on-premises solution and AWS, you can monitor the throughput of the migrated workload using AWS Direct Connect resources and set up Amazon CloudWatch alarm throughput notification.

Suggestion 1.4.4: Use metrics and logs from AWS Migration Services to monitor inflight migrations.

For more detail, see Monitoring Application Migration Service and Monitoring AWS DMS tasks.

MIG-REL-02: What is your business continuity plan for the migrated workload?

Each organization has different set of requirements to build a business continuity plan (BCP) or disaster recovery (DR) plan. The BCP needs to be updated during migration, as the locations of workloads are changing. The risks associated with cloud services need to be added to the BCP. For more detail, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud.

MIG-REL-BP-2.1: Keep your business impact analysis up-to-date

This BP applies to the following best practice areas: Failure management

Implementation guidance

Suggestion 2.1.1: Check that your portfolio and CMDB data is correct.

Keeping application metadata up-to-date is challenging. Commonly, application metadata (for example, number of users) is only updated periodically. Additionally, applications which were once

important might not be so critical as alternatives become available. The application metadata should be verified so the correct BCP is put in place as part of the migration project.

MIG-REL-BP-2.2: Update the risk assessment for the type of disaster events covered by your BCP

This BP applies to the following best practice areas: Failure management

Implementation guidance

Suggestion 2.2.1: Add new events for your cloud environment.

Various events in the cloud environment for example complete loss of AWS Region, complete loss of an AWS Availability zone or service degradation of a single AWS service need a risk assessment. A risk assessment measures how likely an event will occur vs the impact of that event to business applications and helps determine the recovery targets for certain events.

MIG-REL-BP-2.3: Define the recovery point objective (RPO) and recovery time objective (RTO) targets

This BP applies to the following best practice areas: Failure management

Implementation guidance

Suggestion 2.3.1: Create a small number of different RPO and RTO classes.

Migrations can have hundreds of applications in scope, and creating many different RPO or RTO targets which map to different disaster recovery strategies can increase the complexity of migration.

MIG-REL-BP-2.4: Select a disaster recovery strategy based on cloud best practices

This BP applies to the following best practice areas: Failure management

Implementation guidance

Suggestion 2.4.1: Familiarize yourself with disaster recovery options in the cloud.

You must a select disaster recovery option which meets your RPO and RTO targets and addresses risks defined in your BCP. For example, AWS Elastic Disaster Recovery replicates Amazon EC2 instances to another AWS Availability Zone (or another Region) to address the risk of disasters within AWS.

Suggestion 2.4.2: Automate disaster recovery options to be implemented as migrations occur.

For example, in migrations using AWS Application Migration Service, there is a post-launch action to configure AWS Elastic Disaster Recovery (AWS DRS).

MIG-REL-03: What is the maintenance window for the migration cutover?

During migration activity, business process may not be resilient to extend downtime windows. Align the migration event based on your business need.

MIG-REL-BP-3.1: Estimate the required maintenance window

This BP applies to the following best practice areas: Change management

Implementation guidance

Suggestion 3.1.1: Migration to AWS could involve a brief or extended outage of service during the cutover from the current environment.

A typical application cutover involves shutting down the source application, then running a final synchronization of data. The amount of data in the final synchronization, combined with the speed of network links, determines the outage period required for the migration. For example, database migrations can be performed using a backup and restore method or AWS Database Migration Service. These methods offer different cutover windows. Users of the applications being migrated need to be informed of the accurate outage period, with appropriate lead time to assess the impact and plan contingencies.

MIG-REL-BP-3.2: Test the migration window and impact

This BP applies to the following best practice areas: Change management

Implementation guidance

Suggestion 3.2.1: Dry-run the migration activities to validate that they can be completed in the defined maintenance window.

Perform dry-run tests in environments with similar data volume or anticipate the additional volume of data that the production environment has compared to non-production testing (usually significant). Monitoring tools can help provide accurate data change rates. For more detail, see Running a proof of concept.

Suggestion 3.2.2: In case the migration data synchronization activities or testing take longer than the defined maintenance window, define a process to measure the impact on your business and set a contingency plan.

For some applications, it may be fine to extend the maintenance period, but for others, immediate rollback of the migration is required. For more detail, see Developing a cutover plan.

MIG-REL-BP-3.3: Plan for failure

This BP applies to the following best practice areas: Change management

Implementation guidance

Suggestion 3.3.1: Calculate and document the time required to rollback.

A migration checkpoint should be put in place to enable rollback to be performed within the defined maintenance window. For more detail, see Rollback.

Suggestion 3.3.2: Define a communication channel for the migration event and communication intervals agreed with stakeholders.

Communication channels should be used to make decisions during unexpected events. For example, if the maintenance window needs to be extended, a message can be sent to application owners to approve extension or initiate rollback. For more detail, see Communication and governance planning.

Suggestion 3.3.3: Determine how data can be copied back to the source environment.

After deciding to rollback a migration, you may need to copy data back to the source environment. For EC2 instances, AWS Elastic Disaster Recovery can be used to perform a failback from AWS to on-premises environments. For databases, depending on the amount of data to be synchronized, native replication tools can be used, or a database backup and restore can be performed.

Mobilize

The mobilize phase involves setting up the resources and tools needed for the migration. During this phase, the team impacted by the migration are trained to handle the migration. Have a backup plan ready in case anything unexpected happens.

MIG-REL-04: Have you reviewed service quotas and constraints for new migrated resources?

Service quotas exist to prevent accidentally provisioning more resources than you need, and to limit request rates on API operations so as to protect services from abuse. For more detail, see Manage service quotas and constraints. Migrations can add new resources to existing accounts and therefore affect service quotas. Migration services have quotas which can affect the speed migrations can be performed.

MIG-REL-BP-4.1: Be aware of service quotas and constraints for migration services

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 4.1.1: Review service quotes for AWS Application Migration Service and AWS DMS, as these can affect the sizing of your migration waves and number of operations which can be performed simultaneously.

For example, you can only migrate 200 servers within one job. Hitting limits for these services can disrupt migration plans.

MIG-REL-BP-4.2: Estimate the impact of new workloads on existing service quotas across accounts and Regions

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 4.2.1: Review service quota values that would breach if new migration workloads are added to an account.

For example, adding many new EC2 instances into a VPC can cause the soft limit network interfaces per Region (default 5000) to be reached. Request limit increases for such quotas before your migration events.

MIG-REL-BP-4.3: Be aware of unchangeable service quotas and how you determine which accounts or VPCs workloads use

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 4.3.1: Depending on the migration scope, you may have to split workloads into multiple AWS accounts or VPCs to avoid hitting unchangeable service quotas.

MIG-REL-05: How do you plan your network topology for migration activity?

Workloads often exist in multiple environments. It could be between multiple cloud environments and your existing data center. During the migration planning phase, include network considerations, such as intra-system and intersystem connectivity, public IP address management, private IP address management, and domain name resolution.

MIG-REL-BP-5.1: Provide sufficient bandwidth for normal and traffic from data replication

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.1.1: Replication traffic may need to be throttled so it does not overwhelm network links.

For more detail, see Data routing and throttling and Improving the performance of an AWS DMS migration.

MIG-REL-BP-5.2: Assure that links and equipment to on-premises are highly available

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.2.1: If network connectivity is disrupted, migration replication may need to be restarted. Use multiple AWS Direct Connect connections or VPN tunnels between separately deployed private networks.

Use multiple Direct Connect locations for high availability. If using multiple AWS Regions, ensure redundancy in at least two of them. You might want to evaluate AWS Marketplace appliances

that terminate VPNs. If you use AWS Marketplace appliances, deploy redundant instances for high availability in different Availability Zones.

MIG-REL-BP-5.3: Verify that your network design enables communication between on-premises and cloud networks

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.3.1: Design networks to prevent overlapping IP addresses with your on-premises network.

Select new IP ranges to be assigned to AWS VPCs which do not clash with any existing networks. Even though some networks may eventually by freed by the migration, both networks are in use during the migration and difficult to free up until the end of the migration.

Suggestion 5.3.2: Not all networks need to be routable to on-premises.

To preserve routable IP space, non-routable CIDR ranges can be used. For more detail ,see Preserve routable IP space in multi-account VPC designs for non-workload subnets.

MIG-REL-BP-5.4: Use an IP scheme that allows for sufficient growth within cloud workloads and burst auto-scaling

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.4.1: Amazon VPC IP address ranges must be large enough to accommodate workload requirements, including factoring in future expansion and allocation of IP addresses to subnets across Availability Zones.

This includes load balancers, EC2 instances, and container-based applications.

MIG-REL-BP-5.5: Complete a reliable DNS design that enables resolutions to existing domains, plus new domains in AWS

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.5.1: DNS resolution between multiple AWS Accounts and on-premises is fundamental for application communication.

For more detail, see designs for single-account and multi-account hybrid DNS resolutions. During any multi-phase migration, on-premises applications need to talk to migrated applications through DNS, and migrated applications need to talk to on-premises applications. For more detail, see Automating DNS infrastructure using Route 53 Resolver endpoints.

Suggestion 5.5.2: Windows workloads require DNS for active directory (AD).

For rehost (lift and shift) migrations, the same AD domain needs to be resolvable in both on-premises and cloud environments. To avoid affecting the production environment during testing windows server migrations, machine password rotation and dynamic DNS updates should be disabled.

Suggestion 5.5.3: Plan for how to update DNS records during migration testing and cutovers.

Some systems (like Windows) can update dynamically, while others require manual updates. Provide the migration team access to update DNS records or develop automated processes.

MIG-REL-BP-5.6: Test network performance prior to migration

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.6.1: Network latency has varying effects on applications.

Testing how migrated applications respond to the new networking environment can be performed using distributed load testing and monitoring performance metrics.

MIG-REL-BP-5.7: Test network component failure

This BP applies to the following best practice areas: Foundations

Implementation guidance

Suggestion 5.7.1: Network outages have varying effects on applications.

Testing how applications respond to connectivity events can be performed using AWS Fault Injection Service. For more detail, see Tutorial: Simulate a connectivity event.

MIG-REL-06: How do you back up data on migrated workloads?

Back up data, applications, and configuration to meet requirements for recovery time objectives (RTO) and recovery point objectives (RPO). Backup capabilities of cloud environment will differ from on-premises.

MIG-REL-BP-6.1: Identify and back up all data that needs to be backed up, or reproduce the data from sources

This BP applies to the following best practice areas: Workload architecture

Implementation guidance

Suggestion 6.1.1: Validate that your backup process implementation meets your recovery time objectives (RTO) and recovery point objectives (RPO) by performing a recovery test both before and after the migration.

For more detail, see the following:

- Identify and back up all data that needs to be backed up, or reproduce the data from sources
- Secure and encrypt backups
- Perform data backup automatically
- Perform periodic recovery of the data to verify backup integrity and processes

MIG-REL-07: Are your migrated workloads utilizing fault isolation?

Your workload must operate reliably despite data loss or latency in these networks. Components of a migrated workload can use AWS capabilities to design the workloads in more fault-tolerate ways. Follow these best practices to ensure your migrated workloads are fault-tolerant.

MIG-REL-BP-7.1: Deploy the workload to multiple locations

This BP applies to the following best practice areas: Workload architecture

Implementation guidance

Suggestion 7.1.1: Follow the best practices (BPs) in the Reliability pillar to distribute workload data and resources across multiple Availability Zones or, where necessary, across AWS Regions.

These locations can vary as required by your business requirements. Always (when possible) deploy your workload components to multiple AZs for high availability. For components that can only run in a single AZ, implement the capability to do a complete rebuild of the workload within your defined recovery objectives.

For more detail, see the following:

- Deploy the workload to multiple locations
- Automate recovery for components constrained to a single location

Migrate

The migrate phase is where the actual migration of the workload takes place. In this phase, we perform migration as planned, monitor the migration process, and keep a plan in place to rollback in case issues encountered during the migration.

MIG-REL-08: Have you tested high availability (HA), fault tolerance (FT), and disaster recovery (DR)?

Test to validate that your workload meets functional and non-functional requirements before and after migration cutover. It is important to validate and update existing reliability components, which may be different in the new cloud environment.

MIG-REL-BP-8.1 Before the cut-over, test HA and FT for the migrated workloads, and perform a DR dry-run after the migration

This BP applies to the following best practice areas: Failure management

Implementation guidance

Suggestion 8.1.1: Follow the best practices (BPs) in the reliability pillar to complete the failure management testing for the migrated workloads.

This includes using playbooks to investigate failures, performing post-incident analysis, testing functional requirements for the migrated applications, testing scaling and performance, and testing resilience using chaos engineering.

For more detail, see How do you test reliability.

Performance efficiency

The performance efficiency pillar focuses on the efficient use of computing resources to meet requirements and the maintenance of that efficiency as demand changes and technologies evolve. As migration progresses through the assess, mobilize, and migrate and modernize phases, it is important to avoid impacting performance. This can be done by selecting the right architecture or right infrastructure and resources for the migrated workload and ensuring you have mechanism to automatically detect smigub-optimal performance. This section includes best practice considerations to maintain performance efficiency while migrating workloads into the AWS Cloud.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

In the assess phase of migration to AWS, it's important to evaluate performance requirements for your workloads and ensure your existing OS platforms align with those needs. Efficient data transfer methods and the selection of the most suitable storage options are key considerations. Additionally, identifying network requirements and implementing a strategy for managing IP address conflicts and DNS requirements are vital steps for a smooth and successful migration process.

MIG-PERF-01: Have you evaluated performance requirements for the workloads that you are migrating?

AWS has various options for instance class, sizes, purchase options, scaling options, and managed services. Consider how using these capabilities during and after migration can lead to improved performance in your cloud infrastructure.

MIG-PERF-BP-1.1: Understand the performance characteristics of your current infrastructure to select the best performant optimized cloud infrastructure

This BP applies to the following best practice areas: Architecture selection

Implementation guidance

Suggestion 1.1.1: Use discovery tools for a comprehensive view of IT inventory.

Discovery tools offer a comprehensive view of an organizations IT environment, including physical servers, virtual machines, applications and their inter-dependencies. This enhanced visibility enables better planning and decision-making during the migration process. Organizations can identify potential bottlenecks, performance issues, and optimization opportunities using discovery tools.

Discovery tools collect various information, such server names, CPU, disk, and memory utilized. They also collect both server and database configuration information. Server information includes hostnames, IP addresses, and MAC addresses, as well as the resource allocation and utilization details of key resources such as CPU, network, memory, and disk. Collected database information includes the database engine identity, version, and edition. Once collected, this information can be used to size AWS resources as part of migration planning. Addressing these issues before migration can lead to improved performance in the cloud environment.

Accurate information provided by the discovery tools helps determine the appropriate resource allocation and instance sizing on AWS. By understanding the resource utilization patterns and peak loads of application, organizations can provision the right type and size of AWS instance to support the migrated workloads efficiently.

MIG-PERF-02: Have you identified your existing OS platforms to meet your performance requirements?

Gain insights into the assessment and selection process for your current OS platforms and the migration tools considered to meet your performance requirements.

MIG-PERF-BP-2.1: Evaluate operating systems and versions that are running in your environment

This BP applies to the following best practice areas: Compute and hardware

Implementation guidance

Suggestion 2.1.1: Consider an assessment for a legacy workload migration.

Many companies are still running legacy and non-x86 systems in their datacenters, such as mainframe, midrange, or UNIX proprietary systems. Additionally, some applications run on legacy operating systems like Windows Server 2003, 2008, and 2012. Migrating these workloads across hardware architectures to AWS can be a complex process, but there are several best practices to improve the likelihood of a successful transition.

To embark on a successful migration process, it is essential to conduct a comprehensive evaluation of your current systems, delving into their interdependencies, configurations, and resource demands. Pay close attention to any legacy operating system versions and non-x86 hardware that might still be in operation. Equipped with this understanding, create a migration plan that outlines clear objectives and a well-defined timeline. This plan should serve as the roadmap for a smooth transition, making the migration process efficient and minimizing potential disruptions to your operations.

Suggestion 2.1.2: When dealing with non-x86 architectures, there are primarily two approaches: emulation and virtualization. Evaluate both.

- Emulation involves the system simulating the behavior of a different architecture. Essentially, it acts as if it were the target architecture, translating instructions as needed. While emulation is crucial when running software designed for a completely distinct architecture, it can be relatively slower and less efficient than native running or virtualization. It might also consume more system resources, potentially impacting performance compared to the native architecture.
- Virtualization, on the other hand, involves creating a virtual machine (VM) that can run an operating system designed for a specific architecture. Virtualization is generally more efficient and provides better performance compared to emulation because it leverages the underlying hardware and allows multiple VMs to run on the same physical server. While this approach often requires more initial setup, it's a popular choice for running non-x86 architectures in data centers.

The choice between emulation and virtualization depends on your specific use case, performance requirements, and the compatibility of the software you want to run with the chosen method. For more detail, see the following:

- Rehosting Legacy systems to AWS with Stromasys
- Refactoring applications with AWS Blue Age
- Legacy Migration Options to AWS cloud

MIG-PERF-03: How do you find the best transfer methods for efficiently transferring storage data into and out of AWS?

Planning is crucial when migrating data to the cloud. Data is the foundation for successful application deployments, analytics, and machine learning. Customers frequently perform bulk migrations of their application data when moving to the cloud. There are different online and offline methods for moving your data to the cloud. When proceeding with a data migration, data owners must consider the amount of data, transfer time, frequency, bandwidth, network costs, and security concerns. No matter how data makes its way to the cloud, customers often ask us how they can transfer their data to the cloud as quickly and as efficiently as possible.

MIG-PERF-BP-3.1: Evaluate the different methods to migrate data and select the one best for you use case: online mode, offline mode, or hybrid approach

This BP applies to the following best practice areas: Data management

Implementation guidance

Suggestion 3.1.1: Review AWS Cloud Data Migration services for online, offline, and hybrid data transfer options.

Data is a cornerstone of successful application deployments, analytics workflows, and machine learning innovations. When moving data to the cloud, you need to understand where you are moving it for different use cases, the types of data you are moving, and the network resources available, among other considerations. AWS offers a wide variety of services and partner tools to help you migrate your data sets, whether they are files, databases, machine images, block volumes, or even tape backups. AWS provides a portfolio of data transfer services to provide the right solution for any data migration project. The connectivity is a major factor in data migration, and

AWS has offerings that can address your hybrid cloud storage, online data transfer, and offline data transfer needs.

For more detail, see Best practices for accelerating data migrations using AWS Snowball Edge Edge.

MIG-PERF-04: How do you select the best-performing storage option for your workload?

AWS offers a broad portfolio of reliable, scalable, and secure storage services for storing, accessing, protecting, and analyzing your data. This makes it easier to match your storage methods with your needs, and provides storage options that are not easily achievable with on-premises infrastructure. When selecting a storage service, aligning it with your access patterns is critical to achieve the performance you want. You can select from block, file, and object storage services, as well as cloud data migration options for your workload.

MIG-PERF-BP-4.1: Select the storage solution based on the characteristics of your workloads

Identify and document the workload storage needs and define the storage characteristics of each location. Examples of storage characteristics include: shareable access, file size, growth rate, throughput, IOPS, latency, access patterns, and persistence of data. Use these characteristics to evaluate if block, file, object, or instance storage services are the most efficient solution for your storage needs.

This BP applies to the following best practice areas: Data management

Implementation guidance

Suggestion 4.1.1: Understand storage characteristics and requirements.

Identify your workload's most important storage performance metrics and implement improvements as part of a data-driven approach, using benchmarking or load testing. Use this data to identify where your storage solution is constrained, and examine configuration options to improve the solution. Determine the expected growth rate for your workload and choose a storage solution that meets those rates. Research AWS storage offerings to determine the correct storage solution for your various workload needs. Provisioning storage solutions in AWS provide the opportunity for you to test storage offerings and determine if they are appropriate for your workload needs.

Suggestion 4.1.2: Make decisions based on access patterns and metrics.

Choose storage systems based on your workload's access patterns and by determining how the workload accesses data. Configure the storage options you choose to match your data access patterns.

How you access data impacts how the storage solution performs. Select the storage solution that aligns best to your access patterns, or consider changing your access patterns to align with the storage solution to maximize performance.

For example, creating a RAID 0 array allows you to achieve a higher level of performance for a file system than what you can provision on a single volume. Consider using RAID 0 when I/O performance is more important than fault tolerance. For example, you could use it with a heavily used database where data replication is already set up separately.

For storage systems that are a fixed size, such as Amazon EBS or Amazon FSx, monitor the amount of storage used versus the overall storage size and create automation if possible to increase the storage size when reaching a threshold.

For more detail, see the following:

- Amazon EBS Volume Types
- Amazon EC2 Storage
- Amazon EFS: Amazon EFS Performance
- Amazon FSx for Lustre Performance
- Amazon FSx for Windows File Server Performance
- Amazon Glacier: Amazon Glacier Documentation
- Amazon S3: Request Rate and Performance Considerations
- Cloud Storage with AWS
- EBS I/O Characteristics
- Monitoring and understanding Amazon EBS performance using Amazon CloudWatch

Related videos:

- Deep dive on Amazon EBS (STG303-R1)
- Optimize your storage performance with Amazon S3

Related examples:

- Amazon EFS CSI Driver
- Amazon EBS CSI Driver
- Amazon EFS Utilities
- Amazon EBS Auto scale
- Amazon S3 Examples

MIG-PERF-BP-4.2: Choose the optimal storage solutions for specialized workloads, such as SAP and VMware cloud on AWS

This BP applies to the following best practice areas: Data management

Implementation guidance

Suggestion 4.2.1: Implement one of four categories of storage capabilities for VMware Cloud on AWS.

VMware Cloud on AWS is a jointly engineered solution by VMware and AWS that brings VMware's Software-Defined Data Center (SDDC) technologies to the global AWS infrastructure.

If you have workloads with varying storage requirements, it's important to understand the storage options available and how they could work best for different scenarios.

VMware Cloud on AWS offers VMware vSphere workloads with choice and flexibility to integrate with multiple storage services. However, each service is optimized for a specific scenario and no single approach is ideal for all workloads. To choose the right service, you must first understand the storage requirements and performance profiles of your VMware vSphere workloads. With that in mind, you can plan and implement your storage with cost, availability, and performance requirements optimized for your workloads.

Suggestion 4.2.2: Select the optimal storage solutions for your SAP workloads.

AWS offers a wide range of services, including block, file, and object storage, to meet the storage needs of your SAP databases, applications, and backups. We recommend following the guidelines that have been benchmarked and certified by SAP. For SAP HANA, there are very specific guidelines. Other databases require more analysis to match your workload.

MIG-PERF-BP-4.3: Evaluate the different storage tiers at prices to meet your migrated workload's performance

This BP applies to the following best practice areas: Data management

By identifying the most appropriate destination for specific types of data, you can reduce Amazon Elastic Block Store (Amazon EBS) and Amazon Simple Storage Service (Amazon S3) cost while maintaining the required performance and availability. For example, where performance requirements are lower, using Amazon EBS Throughput Optimized HDD (st1) storage typically costs half as much as the default General Purpose SSD (gp2) storage option.

Implementation guidance

Suggestion 4.3.1: Understand EBS storage tiers to balance performance and cost in AWS.

- Use provisioned IOPS SSD (io1) volumes for high performance databases and transactional workloads. io1 provides low latency and the ability to provision high IOPS. However, it is more expensive than other EBS types.
- Use general purpose SSD (gp2) volumes for most workloads. gp2 provides a good blend of price and performance. You can provision up to 16,000 IOPS per volume.
- Use throughput optimized HDD (st1) for large, sequential workloads like log processing. st1 provides low cost per GB of storage.
- Use cold HDD (sc1) for infrequently accessed storage. sc1 is the lowest cost EBS storage.
- Use EBS snapshots to take backups of EBS volumes. Snapshots only copy changed blocks, minimizing storage costs.
- Resize EBS volumes up or down as needed to right-size storage to your current workload. This avoids over-provisioning expensive storage.
- Use Elastic File System (EFS) for shared storage across multiple EC2 instances. EFS storage auto-scales on demand without needing to provision capacity ahead of time.
- Use Lifecycle Manager to automatically move old EBS snapshots to cheaper S3 storage. This reduces your EBS storage costs.
- Monitor your storage metrics in CloudWatch and adjust.

For more detail, see Cost-optimizing Amazon EBS volumes using AWS Compute Optimizer.

Suggestion 4.3.2: Lower your storage costs without sacrificing performance with Amazon S3.

If you have an increasing number of Amazon S3 buckets, spread across tens or even hundreds of accounts, you might be in search of a tool that makes it easier to manage your growing storage footprint and improve cost efficiencies. Amazon S3 Storage Lens is an analytics feature built in to the Amazon S3 console to help you gain organization-wide visibility into your object storage usage and activity trends, and to identify cost savings opportunities. Amazon S3 Storage Lens is available for all Amazon S3 accounts. You can also upgrade to advanced metrics to receive additional metrics, insights, and an extended data retention period.

For more detail, see Amazon S3 Storage Classes.

MIG-PERF-05: Have you identified the network requirements for your migration?

Establishing secure and reliable network connectivity is paramount to facilitating workload migrations in the AWS Cloud. In order to accomplish this, it is necessary to examine network requirements in detail, including on-premises firewall rules, traffic prioritization rules, and source change rates. This practice creates seamless communication during and after migration, minimizes disruptions, ensures optimal performance, and maintains uninterrupted connectivity. AWS offers a wide variety of connectivity options and features tailored to suit the migration requirements and existing network infrastructure of organizations.

MIG-PERF-BP-5.1: Establish a reliable network connectivity from on-premises to AWS to ensure performance

Implementation guidance

Suggestion 5.1.1: Use dedicated network connectivity options for reliably connecting on-premises to AWS.

There are public and private connectivity options, but data transfer over the internet may not be a reliable means of data communication. VPNs provide private connectivity, but they too use internet in the background, therefore relying heavily on external factors of the network. Such customers use a dedicated network channel or option such as AWS Direct Connect to ensure performance over network. AWS Direct Connect creates highly resilient network connections between Amazon Virtual Private Cloud and your on-premises infrastructure. As a result, it is a viable solution for workloads requiring low latency and high bandwidth, such as real-time applications and large data transfers.

Suggestion 5.1.2: Identify the network bandwidth required and supported for ensuring performance.

First of all, network bandwidth *required* and *supported* are two different identification points. Let's first look at how to identify network bandwidths *required* to ensure performance. The requirement depends on workloads or applications that you are looking to migrate. Sensitive applications that are heavily write intensive require continuous data protection mechanisms in order to migrate them to the cloud. The change rate (in Mbps or Gbps) on these source applications determine how much bandwidth you want to provision. Accordingly, you can provision the network bandwidth higher than the source change rate. AWS Direct Connect provides multiple options for connection speeds (1 Gbps, 10 Gbps, 100 Gbps) that you can leverage to provision higher network bandwidths than the source change rate.

Once the network is provisioned for migrating data, you need to identify how much bandwidth does it actually support. You can check that by running any third-party network speed tests (like iperf).

MIG-PERF-BP-5.2: Assure that network performance is not impacted by external factors

Implementation guidance

Suggestion 5.2.1: Identify network bottlenecks on-premises.

Identify network bottlenecks in your on-premises firewalls, perimeter networks, proxies, routers, or any other traffic de-prioritizations. This could impact the network throughputs required for migrating data to cloud.

Suggestion 5.2.1: Provision the right AWS instance types and EBS volumes that support the required network bandwidth.

Make sure that the AWS instance types you provision for your target workloads support the network bandwidths required for the data migration. Each AWS instance type support a specific baseline and burst bandwidth, so make sure that you correctly right-size the instance type for your workload on AWS. Similarly, provision the right EBS volume to support the required IO performance.

For more detail, see the following:

- Amazon EC2 instance network bandwidth

- RDS Instance types and bandwidths supported
- EBS volume types and the maximum throughput it supports

MIG-PERF-06: Do you have a strategy to manage IP address conflicts and DNS requirements as part of the migration process?

In cloud migrations, the key elements of DNS, DHCP, and IP address considerations are essential for the seamless operation of applications and services in the cloud environment.

MIG-PERF-BP-6.1: Identify a migration strategy for your network components (DNS, IP addressing, and DHCP) migration

This BP applies to the following best practice areas: Networking and content delivery

Implementation guidance

Suggestion 6.1.1: Define a DNS management system for your migrated workloads on AWS.

The DNS management planning and setup is a pre-migration task. There are two options for setting up DNS for migrated workloads:

1. Customers choose to use the same DNS management system on-premises while their workloads are migrated to AWS. In this scenario, customers can use AWS Route 53 Resolver endpoints to create a hybrid DNS solution between AWS and an on-premises network.
2. Customers can set up DNS on Amazon Route 53 and migrate existing records, or create new records from the on-premises DNS environment to the public or private hosted zone on Amazon Route 53.

Suggestion 6.1.2: Design a migration strategy for IPs.

Request elastic IP addresses for resources requiring static public IP addresses, allocate appropriate CIDR blocks to VPCs and subnets to accommodate all migrated resources, and conduct meticulous IP range planning to prevent IP conflicts between on-premises and AWS environments post-migration. It is essential to determine if IP addresses need to be reassigned after migration to meet specific requirements during the migration process. A reassignment of IP addresses is likely

necessary for compatibility with third-party systems that rely on fixed IP addresses to establish connections or communicate with the migrated resources.

It is also possible that certain regulatory requirements require the use of static private IP addresses for specific applications or services, necessitating the use of same private IP on AWS to comply with those requirements. For rehost migrations using AWS Application Migration service (AWS MGN), customer often use the *copy private IP* feature to use the same private IP from the source server on the target environment on AWS.

If you are looking to migrate from IPv4 to IPv6 within AWS, you can use the weighted routing feature with Amazon VPC Lattice to slowly shift the traffic.

Suggestion 6.1.3: Use Amazon-provided DHCP servers and option sets.

DHCP servers should be configured in the new infrastructure to provide IP addresses within the appropriate range if IP addresses are assigned using DHCP.

For more detail, see Hybrid Cloud DNS Options for Amazon VPC.

Suggestion 6.1.4: Consider the following network migration checklist.

Proper DNS configuration, IP planning, and DHCP are key factors to consider when migrating workloads to AWS. Familiarize yourself with the following items to plan for a successful network's components migration.

1. Identify the most efficient method of collecting the existing and new IP schemes for to-be migrated systems. This fosters a seamless transition while ensuring accurate addressing for optimal performance.
2. Implement a well-defined process to acquire the new and current DNS names for the systems undergoing migration. This helps with accurate name resolution while preserving seamless communication.
3. Modify load balancers, proxies, or any other network devices in order to redirect to the new IP addresses or domains post-migration. This avoids interrupting resources.
4. Update DNS settings after the migration to point towards the newly migrated cloud resources so that cloud-based services are properly routed and accessible.
5. The DHCP configuration may need to be adjusted in order to accommodate the integration of new systems. This verifies that IP allocation and network settings accurately reflect the newly-migrated components.

Mobilize

During the mobilize phase of your migration, you need to evaluate the different components that make up the building blocks of your migrated workloads and make trade-offs to select the performance that fits your business requirements. To do this, you need to set up the right metrics for performance monitoring, evaluate the different options to build your architecture, and benchmark the migrated workload against the on-premises workload to measure the different components' performance and adjust if needed.

MIG-PERF-07: How do you verify that the shared services used for migration during the mobilize phase are performing efficiently?

The mobilize phase lays the foundation for tools, process, and culture that accelerate your migration at scale. The account planning, Architecture selection, monitoring, and observability setup in the mobilize phase are building blocks for any future migrations done. Some of the shared services we set up and validate in the mobilize phase are landing zones, AWS Transit Gateway, and Amazon VPC Lattice.

MIG-PERF-BP-7.1: Identify the right CloudWatch metrics to capture or detect anomaly and identify performance blockers for shared services

This BP applies to the following best practice areas: Architecture selection

Implementation guidance

Suggestion 7.1.1: Identify the right metrics and detect anomalies.

During cloud migration, it's essential to monitor AWS resource performance effectively. CloudWatch Metrics Insights helps by providing a SQL query engine to analyze performance metrics in real-time, aiding in the detection of trends and patterns during the migration process. For more focused monitoring, CloudWatch anomaly detection can be enabled for critical metrics, using machine learning to forecast normal behavior and alert on anomalies. Additionally, metrics explorer is a tag-based tool that allows for the organization and visualization of metrics by tags and resource properties, which is particularly useful for maintaining oversight during and after resource migration.

MIG-PERF-BP-7.2: Select the best performing cloud infrastructure that can scale for additional workloads in future without any performance impact

This BP applies to the following best practice areas: Architecture selection

Implementation guidance

Suggestion 7.2.1: Select the best performing architecture from storage, database, compute, and network perspective.

During the mobilize phase, you are essentially laying the groundwork for your first wave of migration and any future migrations. In this phase, you define and implement an AWS landing zone, and other AWS security and network services that can scale as you migrate additional applications. There are multiple approaches and considerations when selecting the best performing architecture, like factoring cost requirements into decisions, or selecting the best compute, or storage, or database, or network architecture. The best performing architecture in your case would be what best fit your requirements.

For more detail, see the following:

- Architecture selection - Performance Efficiency Pillar
- Compute and hardware - Performance Efficiency Pillar
- Data management - Performance Efficiency Pillar
- Networking and content delivery - Performance Efficiency Pillar

Suggestion 7.2.2: Use existing reference patterns for your architecture to achieve a cost-effective solution.

AWS Solutions Architects, AWS Reference Architectures, and AWS Partner Network (APN) partners can help you select an architecture based on industry knowledge. You can maximize performance and efficiency by evaluating existing reference architectures and using your analysis to select services and configurations for your workload.

MIG-PERF-BP-7.3: Reduce the blast radius for performance impact into a single account

This BP applies to the following best practice areas: Architecture selection

Implementation guidance

Suggestion 7.3.1: Organize your AWS accounts to isolate performance impact.

As you are laying the foundation during the mobilize phase of the migration, account structuring is essential to safeguard performance. Account structuring or organizing can isolate performance impact to a single account, reducing the blast radius. Customers can use AWS Organizations, an AWS service to centrally manage and govern multiple accounts. We looked at account structuring during the security pillar, but this specific best practice applies to multiple pillars of the Well-Architected Framework. There are several strategies for multi-account landing zone accounts.

MIG-PERF-BP-7.4: Benchmark existing workloads for performance

Implementation guidance

Suggestion 7.4.1: Benchmark the performance of an existing workload to understand how it performs on the cloud.

Use the data collected from benchmarks to drive architectural decisions. Benchmarking is generally quicker to set up than load testing and is used to evaluate the technology for a particular component. Benchmarking is often used at the start of a new project, when you lack a full solution to load test.

You can either build your own custom benchmark tests, or you can use an industry standard test, such as TPC-DS, to benchmark your data warehousing workloads. Industry benchmarks are helpful when comparing environments. Custom benchmarks are useful for targeting specific types of operations that you expect to make in your architecture.

When benchmarking, it is important to pre-warm your test environment to ensure valid results. Run the same benchmark multiple times to capture any variance over time.

Because benchmarks are generally faster to run than load tests, they can be used earlier in the deployment pipeline and provide faster feedback on performance deviations. When you evaluate a significant change in a component or service, a benchmark can be a quick way to see if you can justify the effort to make the change. Using benchmarking in conjunction with load testing is important because load testing informs you about how your workload will perform in production.

Migrate

As you migrate your workload, you need to consistently compare the migrated workload against the performance requirements that you identified through KPIs or benchmarks. To do this, you

need to perform the necessary testing on your migrated applications, capture any issues or lessons learned, and iterate for the next migration wave.

MIG-PERF-08: How do you ensure improved and consistent performance of your applications during migration?

Migration is an iterative process as most enterprise customers are migrating thousands of servers and applications. Migration is often planned in waves. Migration waves typically span four to eight weeks, and they can contain one or more migration events. Applications and their dependencies are combined into waves so customers can meet the challenges of dependency mapping. But how do you ensure good and consistent performance during this whole process?

MIG-PERF-BP-8.1: Perform stress and user acceptance tests on migrated workloads before the actual cutover.

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 8.1.1: Perform stress and user acceptance tests for quality check before the actual cutover.

Migration tools such as AWS Application Migration Service (AWS MGN) provide features to test and cutover workloads. It is highly recommended to run tests in your test or staging environments so that performance is maintained after cutover. The test could be a stress test to test the systems with three to four times the load in production environments, or a user acceptance test to ensure the application can function properly in the organization. It is also recommended to perform tests at least two weeks prior to the cutover, so there is sufficient time to fix any issues before the cutover.

For more detail, see the following:

- Testing and cutover
- Application migration process
- Testing Phase

MIG-PERF-BP-8.2: Review and implement the lessons learned from previous migration waves

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 8.2.1: Take note of lessons learned from previous migration waves.

As the migration program moves forward and more waves are migrated, it is key to evolve the migration wave plan based on lessons learned and changing business priorities. In particular, for long-running migration programs, it is important to reassess business drivers and organizational change, and to verify that the migration wave plan is still valid. Similarly, lessons learned from the migration influence the wave plan composition and the scope of each wave. To avoid losing visibility into what is happening, keep the wave plan up to date. The plan should reflect and track what is being delivered, and it should manage and assess change to the migration scope.

MIG-PERF-BP-8.3: Perform a Well-Architected Framework Review on each iteration of the migrated workload.

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 8.3.1: Review Well-Architected best practices after each iteration of your migration.

The review of architecture needs to be done in an iterative manner after each migration wave. This involves reviewing the current and target state architectures for future waves to see what can be improved performance wise. Each team member should take responsibility for the quality of their architecture. We recommend that the team members who build an architecture use the Well-Architected Framework to continually review their architecture after each migration wave, rather than conducting a formal performance review meeting.

For more detail, see the following:

- How to perform a Well-Architected Framework Review- Part 1
- How to perform a Well-Architected Framework Review- Part 2
- How to perform a Well-Architected Framework Review- Part 3

Suggestion 8.4.1: Review the 7 Rs migration strategy.

After each migration wave, we recommend to review 7 R decision tree for your applications, considering the learnings throughout migration of the initial pilot applications or subsequent migration waves that had been completed. You need to ensure that the migration strategy continues to provide the best performance for the workload, and verify that it aligns with your initial assessment. The migration strategy is not only derived for the application component but also for the associated infrastructure. The final migration strategy should always provide and optimize the performance for the application and infrastructure. AWS Migration Hub strategy recommendations help automate the analysis of your application portfolios and review of the 7 R strategy. Strategy recommendations analyzes your running applications to determine runtime environments and process dependencies, optionally analyzes source code and databases, and more.

For more detail, refer to the following:

- Iterating the 7 Rs migration strategy selection
- AWS Migration Hub Strategy Recommendations

MIG-PERF-09: How do you monitor the performance through all the phases of your migration journey?

Monitoring performance during the mobilize and migrate phase is essential for a successful migration. Monitoring can help remediate issues before they impact your customers. Monitoring metrics should be used to raise alarms when thresholds are breached. During the mobilize and migrate phases, look at the following best practices for setting up monitoring.

MIG-PERF-BP-9.1: Generate alarm-based notifications for metric's threshold breach

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 9.1.1: Generate alarm-based notifications using Amazon CloudWatch and Amazon SNS.

As you are migrating workloads, system performance can degrade over time. It is recommended to monitor the workload's performance to identify degradations and bottlenecks, and remediate

them automatically. Amazon CloudWatch generates system-defined metrics, and customers can also create custom user-defined metrics. You can use these metrics to generate CloudWatch alarms and add an Amazon SNS topic to send an email notification when the alarm changes state. These SNS notifications can also be integrated with AWS Lambda to take actions for remediating the issue.

AWS X-Ray helps developers analyze and debug production in distributed applications. With AWS X-Ray, you can glean insights into how your application is performing, discover root causes, and identify performance bottlenecks. You can use these insights to react quickly and keep your workload running smoothly.

MIG-PERF-BP-9.2: Determine the need for a real-time or a near real-time monitoring solution

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 9.2.1: Use a real-time or a near-real-time monitoring solution.

Most applications can tolerate some performance degradation and can be monitored in near-real-time. But some applications process data instantly, and therefore need to be monitored in real-time. It is essential to identify the performance needs for your migrated applications and implement a monitoring solution accordingly. Amazon CloudWatch delivers metrics and logs in near-real-time. Metric streams send CloudWatch metrics to destinations like Amazon S3, with near-real-time delivery and low latency. You could also monitor microservices and cloud-native applications in real-time with IBM Instana SaaS on AWS.

MIG-PERF-BP-9.3: Implement CloudWatch or a Quicksight dashboard as a single pane view for visualizing all metrics

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 9.3.1: Implement CloudWatch or a Quicksight dashboard for monitoring metrics.

CloudWatch or a Quicksight dashboard can provide a single pane view of all the monitoring metrics. A single view for selected metrics and alarms help you assess the health of your resources and applications across regions. You can create CloudWatch cross-account observability dashboard

or cross-account, cross-Region dashboards to summarize your CloudWatch data from multiple AWS accounts and multiple Regions into one dashboard.

MIG-PERF-BP-9.4: Set up automated testing for your application metrics

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 9.4.1: Use CloudWatch synthetic monitoring for setting up automated testing for your applications.

Using CloudWatch synthetic monitoring, you can create canaries to monitor your endpoints and APIs. Canaries perform automated testing (perform same actions as a customer) to continually verify your customer experience, even when there is no load. This helps discover issues even before your customer do.

MIG-PERF-BP-9.5: Re-evaluate your compute usage with AWS Trusted Advisor, AWS Compute Optimizer, or partner tools

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 9.5.1: Use Compute Optimizer for reviewing your performance metrics.

AWS Compute Optimizer collects resource utilization data and helps avoid over-provisioning and under-provisioning resources such as Amazon Elastic Compute Cloud (EC2), Amazon Elastic Block Store (EBS) volumes, Amazon Elastic Container Service (ECS) services on AWS Fargate, and AWS Lambda functions.

Cost optimization

The cost optimization pillar includes the continual process of refinement and improvement of a system over its entire lifecycle to optimize cost. As part of their migration journey into the AWS Cloud, successful companies undergoing large scale migration and modernization initiatives incorporate Cloud Financial Management (CFM) principles to optimize cost. The AWS Migration Acceleration Program (MAP) is a comprehensive and proven cloud migration program that can help you accelerate your cloud migration and modernization journey with an outcome-driven methodology.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

In the assess phase, prioritizing cost-effectiveness is essential. This phase involves a comprehensive evaluation of existing infrastructure usage and a thorough analysis of application dependencies. By assessing these aspects, you can pinpoint opportunities for optimizing costs throughout the migration journey. To expedite this cost-effective assessment, consider leveraging AWS programs and workshops designed to remove common blockers and accelerate migrations. By incorporating these best practices, you not only ensure a well-informed migration strategy, but also lay the groundwork for maximizing cost efficiency in your cloud migration.

MIG-COST-01: Are you collecting the right information about your source resources to create cost-optimized destination architectures?

Successful migrations require high-quality data about the source environment and thorough analysis of technology, people, and processes to move quickly and safely.

MIG-COST-BP-1.1: Thoroughly assess existing infrastructure usage and application dependencies prior to migration

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 1.1.1: Use discovery tools or existing data to gather enough data about your source infrastructure to make informed decisions about your target architecture.

Collect a complete inventory of assets to be migrated and analyze dependencies between servers, databases, and applications to create migration wave plans that minimize network chatter and latency between source and target infrastructure. Collect fine-grained infrastructure usage data, including CPU, memory, and disk reads and writes. It's important to understand actual usage from

your source servers, not just how many resources are allocated, in order to right-size the target infrastructure in AWS.

These data should be gathered with frequent samples in order to understand the minimum, average, and maximum usage over time, typically at least two weeks. AWS and our partners offer several tools that can help collect the required information, such as Application Discovery Service and Migration Evaluator. Some customers already have this information in their change management databases (CMDB) or observability tools.

For more detail, see the following:

- AWS Prescriptive Guidance regarding migration tool selection
- AWS Prescriptive Guidance regarding Application portfolio assessment
- AWS Prescriptive Guidance for Wave Planning

MIG-COST-BP-1.2: Leverage AWS programs and workshops designed to remove common blockers and accelerate migrations

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 1.2.1: Leverage AWS and partner programs and experience to improve assessments and identify and remove costly blockers early.

The Migration Acceleration Program (MAP) provides tools that reduce costs and automate and accelerate migration assessment and implementation. In some cases AWS invests in customer migrations in the form of service credits or partner investments. MAP also leverages proven workshops such as Migration Readiness Assessments, Experience-Based Accelerators (EBA), and AWS Learning and Needs Analysis (LNA) to assess and address technology, people, and processes that may create costly blockers or reduce migration velocity.

Suggestion 1.2.2: Use the AWS Optimization and Licensing Assessment (OLA) program to conduct thorough discovery of existing Windows license footprints and cost optimization exercises.

The AWS OLA delivers a comprehensive report that models your deployment options based on actual resource use and your existing licensing entitlements, helping you uncover potential cost savings through our flexible licensing options, including Bring-Your-Own-License (BYOL) and license-included options.

Mobilize

As you start planning for your migration in the mobilize phase, you need to consider planning for optimizing resource utilization and cost management. To achieve this, use existing automation tools to streamline migration processes effectively. Additionally, minimize data transfer to conserve bandwidth and mitigate data egress costs, ensuring a cost-effective transition. Right-sizing replication servers is essential to prevent bottlenecks without unnecessary over-provisioning. Furthermore, establish robust cost and usage governance through IAM policies and define a customized cost allocation strategy tailored to your organization's financial management needs. These practices collectively contribute to a smooth and cost-efficient mobilization of your migration efforts.

MIG-COST-02: Are you using automation efficiently for your migration?

AWS and our partners offer a wide variety of tools and services to help perform your migration. Use these tools efficiently to reduce infrastructure and operational costs during the migration.

MIG-COST-BP-2.1: Leverage existing tools to automate your migration

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 2.1.1: Understand the capabilities of each tool available, and select the one best suited to your situation.

AWS and our partners offer a range of tools to help migration. For instance, AWS Application Migration Service can help with ongoing replication, planning, testing, and cutover for lift and shift server migrations. AWS Migration Hub or Cloud Migration Factory can provide additional planning and reporting functionality on top of Application Migration Service. Some tools are purpose-built for specific workloads, such as Database Migration Service (DMS) and Kubernetes Migration Factory. There are also many other tools offered by AWS partners.

MIG-COST-BP-2.2: Minimize the number of applications and the amount of data that is migrated

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 2.2.1: Only migrate what needs to be migrated and minimize ongoing replication.

In the analyze and mobilize phases, you may have discovered some applications that are still running but are no longer needed. Those are easy targets to retire to limit how much you're migrating. Consider discarding archival data that is beyond its useful retention period. Non-production servers for applications that are not in active development may also be retired.

Additionally, ongoing replication, such as change data capture (CDC) that Application Migration Service or AWS DMS uses, can consume a lot of bandwidth when the rate of change in the source server is high. Too much simultaneous replication may require additional bandwidth to avoid network issues. If migrating from another cloud service provider (CSP), you may incur unnecessary data egress costs when you have unnecessary replication. You can reduce bandwidth requirements by limiting the time your servers are actively replicating, as well as how many you are replicating simultaneously, especially the source servers with a high rate of change.

MIG-COST-BP-2.3: Right-size your replication servers to prevent bottlenecks without over-provisioning

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 2.3.1: Monitor your replication server performance and adjust their size as needed.

You can monitor Application Migration Service and DMS replication server performance in CloudWatch. A replication server with too little performance causes a bottleneck that can increase costs elsewhere, such as operations. A replication server with too much performance can itself cost more than it needs.

MIG-COST-03: Have you established standards to measure, monitor and create accountability to manage the cost of operating in the cloud?

AWS provides tools and services for measuring, monitoring and creating accountability for your cloud spend. Your organization should establish a financial attribution model for the migrated resources. Creating a financial accountability model allows departments to cross-charge departments for shared resources.

MIG-COST-BP-3.1: Plan and set up cost and usage governance of AWS resources with help of IAM policies

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 3.1.1: To effectively manage the costs of your migration, it's essential to have robust control over your AWS resource usage.

Before embarking on mass migrations, establish access control standards in AWS by creating and enforcing policies that are closely tied to migration objectives. These policies can be attached to AWS Identity and Access Management (IAM) principals, including roles or policies, as well as AWS resources. AWS offers various policy types to provide the flexibility needed for cost management within the migration process.

Identity-based policies should be employed to define permissions for IAM roles. For instance, you can attach a policy to an IAM role, specifying that the role is permitted to launch specific instance types or access particular services. These identity-based policies play a crucial role in setting permissions boundaries, which facilitate governance aimed at cost control.

Additionally, resource-based policies should be applied to relevant AWS resources involved in your migration. For example, these policies can be attached to S3 buckets, Amazon SQS queues, VPC endpoints, and AWS Key Management Service encryption keys, aligning security and access controls with migration goals. This keeps cost management tightly integrated with your migration strategy and implementation.

For more detail, see the following:

- How to manage cost overruns in your AWS multi-account environment
- Control developer account costs with AWS CloudFormation and AWS Budgets

MIG-COST-BP-3.2: Define a cost allocation strategy that meets your organizations specific financial management process

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 3.2.1: Migration cost can be optimized by creating a cost awareness culture in your organization.

A good way to start this shift is by information teams on how their decisions impact cost. Cost allocation is foundational to making informed decisions to best support business outcomes. To do this, you need to define a cost allocation strategy that speaks to your specific financial management process, and ties cost and resources usage data to the business needs and outcomes.

Set up resource tagging for cost allocation. Create your resource tags, and then activate your cost allocation tags in the Billing and Cost Management console. There are user-defined and AWS-generated cost allocation tags. Based on the types of services you need to tag and the level of customization you require, you can use one of these two cost allocation tags or a hybrid of both. AWS Cost Categories allows you to logically group accounts and resources with attributes, such as tags, to better map your cost and usage information to your organizational structure.

Use four step process to design chargeback for shared services (for example, central compute savings plans, or enterprise support cost at billing account).

1. Decide on the cost units to chargeback to.
2. Calculate the total cost of the shared services.
3. Choose a distribution logic (equitable or proportional).
4. Gather the data to chargeback accurately.

For more detail, see Chargeback | AWS Cloud Financial Management.

MIG-COST-BP-3.3: Design a strategy to monitor, track and analyze your AWS cost and usage as you move resources to AWS

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 3.3.1: Implement appropriate management, tracking and measurement for your migration cost.

You can use Amazon CloudWatch to collect and track metrics, monitor log files, set alarms, and automatically react to changes in your AWS resources. You can also use Amazon CloudWatch to

gain system-wide visibility into resource utilization, application performance, and operational health.

With Trusted Advisor, you can provision your resources following best practices to improve system performance and reliability, increase security, and look for opportunities to save money. You can also turn off non-production instances, and use Amazon CloudWatch and autoscaling to match increases or reductions in demand.

AWS Cost Explorer has an easy-to-use interface that lets you visualize, understand, and manage your AWS costs and usage over time. You can get started quickly by creating custom reports that analyze cost and usage data. Analyze your data at a high level (for example, total costs and usage across all accounts), or dive deeper into your cost and usage data to identify trends, pinpoint cost drivers, and detect anomalies.

Migrate

Cost optimization doesn't stop after you complete the migration to Cloud. It's essential to seamlessly manage resources and optimize costs on regular basis. Start by closely monitoring your resources throughout both the migration and post-migration stages to ensure smooth transitions. Develop a thoughtful metrics strategy to demystify cloud economics, enabling informed decision-making. Establish budgeting mechanisms to continually monitor cost and usage, and construct user-friendly dashboards with pre-built visualizations for enhanced visibility. Choose the right purchase options and scalable architectures tailored to your specific workloads, and consider a long-term modernization strategy that embraces cost-effective cloud-native services.

MIG-COST-04: What custom monitoring strategies you have put in place to monitor and manage ongoing cost and usage data as you migrate resources to AWS?

Monitoring is an important part of maintaining and managing the cost and usage of AWS resources during and after migration. Customers use different AWS services and tools to successfully manage and optimize their AWS bill.

MIG-COST-BP-4.1: Create a deliberate metrics strategy to help demystify cloud economics

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 4.1.1: Regularly evaluate cloud metrics to ensure they meet current business needs.

Start by creating an effective metrics and reporting strategy, then define a set of metrics to measure the implementation of your strategy. See Crafting a robust metrics strategy to quantify your benefits from the cloud for some proven metrics that follow our *see, save, plan,* and *run* framework, and can serve as a starting point for any company

MIG-COST-BP-4.2: Monitor spend and limit unintended or unnecessary costs with budgeting and forecasting tools

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 4.2.1: Migrating to AWS can likely reduce your total cost of ownership (TCO), but you still need an effective cost control mechanism to make sure you only pay for what you need.

Set up your cost control system, by focusing on the following core principles:

- Budget your spend with custom thresholds.
- Monitor and analyze how your costs progress toward limits.
- Take action to reduce unintended costs.

AWS Budgets gives you the ability to set custom budgets that alert you when your costs or usage (actual or forecasted) exceed your budgeted amount. With the recent launch of AWS Budget actions, you can now preconfigure actions that can trigger the implementation of AWS Identity and Access Management (IAM) policies or service control policies (SCPs). In addition, you can stop target Amazon Elastic Compute Cloud (Amazon EC2) or Amazon RDS instances in your account.

In a multi-account AWS environment, there are two patterns for managing the budget that you can choose from based on your organization's governance structure:

- Centralized budget management, where the budget is set by the management account for all its member accounts
- Decentralized budget management, where the budget is set for individual member accounts by its owners

For more detail, see the following:

- How to manage cost overruns in your AWS multi-account environment – Part 1
- Control developer account costs with AWS CloudFormation and AWS Budgets
- Smart Budgeting Using Lambda and Service Catalog

MIG-COST-BP-4.3: Use AWS Cost Anomaly Detection in Cost Explorer to quickly improve cost controls

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 4.3.1: Cost Anomaly Detection is an AWS Cost Management service that uses advanced machine learning to detect anomalous spend and provide contextualized alert notifications through email and Amazon SNS.

Each anomaly includes root cause analysis and direct links to further investigate in Cost Explorer to understand the unexpected usage and its drivers.

The default configuration includes the creation of an AWS services monitor, which tracks charges of most AWS services (see Quotas and restrictions) deployed now, or in the future, by your management and member accounts. It also includes a daily email subscription, which sends an email if any anomaly is detected or ongoing during that specific day. By default, the primary email address associated with the account will receive a daily summary email for any service anomaly detected that is above $100 and exceeds 40% of the expected spend.

MIG-COST-BP-4.4: Use dashboards that provide pre-built visualizations to help you get a detailed view of your AWS usage and costs as you move resources to AWS

This BP applies to the following best practice areas: Expenditure and usage awareness

Implementation guidance

Suggestion 4.4.1: AWS Cost Explorer provides a high-level view of costs and usage, using the same dataset that is used to generate the AWS Cost and Usage Reports.

To extract resource-level granularity, you can use Amazon Athena queries, which requires familiarity and previous experience to build complex SQL queries.

Having dashboards that provide pre-built visualizations can help you get a detailed view of your AWS usage and costs.

The Cloud Intelligence Dashboards are a collection of Quick Suite dashboards. They offer powerful visuals, in-depth insights, and intuitive querying without having to build complex solutions or share your cost data with third-party companies.

The Cost Intelligence Dashboard, CUDOS Dashboard, Trusted Advisor Organization (TAO) Dashboard, and Trends Dashboard are built on native AWS services. They are inherently secure because the data resides in the organization. They inherit all the features of Quick Suite, including integration with AWS Identity and Access Management, which makes them highly secure, and Quick Suite being a serverless service allows you to pay as you go and scale on demand.

You do not need coding or SQL skills to customize these dashboards. The visualizations include Machine Learning (ML) driven insights, live trends, actionable recommendations, links to relevant blog posts and AWS service documentation that help you make informed business decisions.

MIG-COST-05: How are you ensuring your target infrastructure is optimized for your workloads?

AWS has a lot of options for instance shapes and sizes, purchase options, scaling options, and managed services. Consider how you can leverage these capabilities during and after migration to maximize cost benefits from your cloud infrastructure.

MIG-COST-BP-5.1: Leverage the right purchase options and scalable architecture for your workloads

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 5.1.1: Leverage the right purchase model for your workload.

Selecting the correct purchasing models can greatly reduce the total cost of running a workload. Workloads with fairly consistent resource requirements should consider reserved instances or savings plans to save up to 72 percent on cost compared to the same resources purchased on-demand. On the other hand, some workloads, such as question and answer (QA) environments,

may be temporary or intermittent. This type of workload may benefit from Spot Instances, which can be much more cost effective for well-fit workloads compared to comparable on-demand resources.

In some cases, licensing may have an impact on purchasing models. For instance, some licenses are bound to physical cores and not transferable to an EC2 instance. In this case, it may be more cost effective to purchase a dedicated host to use for the duration of those licenses. For more detail, see EC2 Reservation Models.

Suggestion 5.1.2: Use fine-grained data collected in the assess phase to right-size your infrastructure as you're migrating.

It is unlikely that the resources allocated to your source infrastructure are exactly what your workload is using. The cloud offers more flexibility when provisioning resources and makes it easier to change the size and shape of the provisioned resources. By sizing your target infrastructure to the actual usage, you can see immediate cost savings by migrating to the cloud.

When selecting your target resources, consider burstable performance instances, which allow you to run a workload at a fraction of the cost of comparable-sized non-burstable instances. These instances are especially cost-effective for non-production or other workloads that are often near-idle.

Suggestion 5.1.3: Enable autoscaling immediately after migrating when your workload allows it.

Autoscaling can have a significant impact on infrastructure costs, especially for applications that that have large fluctuations in compute capacity needs over time. If you are able to autoscale, consider that when sizing your target infrastructure. Choose instances that are between your minimum and average compute, memory, and disk usage, and make sure your autoscaling rules allow you to scale up to at least the maximum expected usage.

MIG-COST-BP-5.2: Identify resources during migration that are likely candidates for cost optimizations later

This BP applies to the following best practice areas: Cost-effective resources

Implementation guidance

Suggestion 5.3.1: Tag your resources as you're migrating to make additional cost optimizations later.

It's often not feasible to make every cost optimization during the initial migration. Tag resources as they're being created to distinguish important categories that can help with cost optimization later. For instance, tag production and pre-production instances so you can use the tags later to automate stopping pre-production instances at night.

MIG-COST-06: What cost optimization tools are you leveraging to reduce your cloud spend?

Once you have successfully migrated workloads to AWS, it is critical to choose the right set of AWS cost optimization tools that provide ability to manage, monitor, and track your spend in the cloud. Understanding your cost structure in comprehensive manner and applying it across spectrum of AWS services allows you to implement the right cost optimization solutions in order to optimize your operational cost after migration or modernization.

MIG-COST-BP-6.1: Use automation to re-evaluate your compute usage periodically

This question applies to best practice area: Optimize over time

Implementation guidance

Suggestion 6.1.1: Employ cost-optimization tools built specifically for AWS infrastructure.

AWS provides comprehensive cost management tools for cost optimization. Most workloads continue to evolve over time. Use combination of cost optimization tools to continue cost-optimization post-migration such as Compute Optimizer, Trusted Advisor, rightsizing recommendations, Savings Plans (SP) and Reserve Instances (RI) reports, Amazon CloudWatch alarms, and Amazon S3 Lens based on various services that are part of your AWS environments. These tools analyze your AWS usage and make simple, actionable suggestions to reduce costs of running your workloads on AWS, while incorporating the latest features and pricing.

- AWS Trusted Advisor: Provides recommendations that help you follow AWS best practices. Trusted Advisor evaluates your account by using checks. These checks identify ways to optimize your AWS infrastructure, improve security and performance, reduce costs, and monitor service quotas. You can then follow the recommendations to optimize your services and resources.
- AWS Compute Optimizer: Identifies whether your AWS resources are optimal, and offers recommendations to improve cost and performance. It helps us with rightsizing recommendations by avoiding over provisioning and under provisioning.

- Rightsizing recommendations: This feature in AWS Cost Explorer helps you identify cost-saving opportunities by downsizing or terminating instances in Amazon Elastic Compute Cloud (Amazon EC2). Rightsizing recommendations analyze your Amazon EC2 resources and usage to show opportunities for how you can lower your spending.
- **SP and RI reports**: If you own Savings Plans or Reserved Instances, this helps to understand how much SP or RI to purchase. Once you purchase, it also helps you understand the coverage, which is a measure of how many instances are covered out of all your EC2 and RDS instances.
- Amazon CloudWatch: Provides detailed monitoring of infrastructure components at a line item about different services being consumed and allows to set near real-time alarms for Cloud Watch.
- Amazon S3 Lens: Allows customers to get visibility into their Amazon S3 usage. Amazon S3 Storage Lens is a single place to understand Amazon S3 consumption and provides recommendations for objects that haven't been accessed for long time, different storage tiers, and other storage related metrics.

Follow rightsizing recommendations provided by combination of different AWS native Cloud Financial Management tools.

MIG-COST-07: How are you prioritizing your migrated AWS workloads and further driving cost optimization through modernization?

Migrations don't end when a workload is in AWS. It's important to continue to optimize costs post-migration to get the most value from the cloud. There are various modernization pathways that allow you to further optimize your cost profile on AWS while delivering innovative solutions, reducing time-to-value and improving customer experiences.

MIG-COST-BP-7.1: Create a plan early to optimize after the initial migration

This question applies to best practice area: Optimize over time

Implementation guidance

Suggestion 7.1.1: Depending on your specific AWS architecture and services being deployed, there are a number of techniques to further optimize cost both at the infrastructure layer, including refactoring your application to take advantage of modern, cloud-native services.

- **Use newer, cost-efficient compute options available for your workloads:** There are three important ways to optimize compute costs, and AWS has tools to help you with all of them. It starts with choosing the right Amazon EC2 purchase model for your workloads, then selecting the right instance to fine tune price and performance, and finally mapping usage to actual demand.
- **Use an optimal combination of AWS Managed Services:** Depending on your existing AWS architecture, you can re-platform applications to further save on operating costs in the cloud. If you are running a SQL database on Amazon EC2, modernizing to an Amazon RDS managed service can remove lot of undifferentiated heavy-lifting and lower overall total cost of ownership (TCO).
- **Modernization pathways**: The four most popular modernization pathways are as follows:
  - **Serverless:** Helps organizations to build and run applications without provisioning or managing infrastructure. These services, such as AWS Lambda and AWS Fargate, allow organizations to worry less about operational overhead and have a faster time to market. Features like automatic scaling and pay-for-use billing drive business agility and cost-efficiency.
  - **Containers**: Containers provide a standard way to package an application's code, configurations, and dependencies into a single object. Examples of container services include Amazon Elastic Container Service (ECS) and AWS Elastic Kubernetes Service (EKS).
  - **Managed data:** A fully managed, purpose-built database service, supporting diverse data models and applications.
  - **Managed analytics:** A range of services supporting analytics use cases like data lake initiatives, big data processing, real-time analytics, and operational analytics.

Sustainability

Sustainability is increasingly becoming a motivator for customers to migrate to the cloud. The sustainability pillar includes the ability to continually improve sustainability impacts by reducing energy consumption and increasing efficiency across all components of a workload by maximizing the benefits from the provisioned resources and minimizing the total resources required. It's important to consider sustainability while migrating workloads to AWS. This includes enhancing energy efficiency, transitioning to renewable energy, reducing embodied carbon, using water responsibly, driving a circular economy, and enabling sustainability for customers. Sustainability *of* and *through* the cloud is covered in the Sustainability Pillar of the AWS Well-Architected Framework.

Migration phases

- Assess
- Mobilize
- Migrate

Assess

By migrating to AWS, customers can benefit from our investments in renewable energy and economy-of-scale to run their workloads with lower carbon emissions. As part of the assess phase, you can calculate the expected carbon emission reductions when migrating to AWS by comparing estimated annual carbon emissions for a customer's current on-premises infrastructure with corresponding carbon emissions of right-sized workloads in AWS. This phase also enables you to identify sustainability related key performance indicators (KPIs) to align key stakeholders.

MIG-SUS-01: Is sustainability a consideration for creating your migration business case?

Sustainability is increasingly becoming a motivator to migrate to the cloud. By migrating to AWS, customers can benefit from our investments in renewable energy and economy-of-scale to run their workloads with lower carbon emissions. The migration business case should demonstrate the carbon emission reductions customers can expect when migrating to AWS.

MIG-SUS-BP-1.1: Include sustainability considerations as part of your migration business case and preliminary assessments

This BP applies to the following best practice areas: Process and culture

A complete migration business case includes the following business impact areas: cost savings, staff productivity, operational resilience, business agility, and sustainability considerations and goals. Sustainability should be included in the business case and aligned with organizational goals.

Implementation guidance

Suggestion 1.1.1: Identify a migration stakeholder to own sustainability goals.

A single-threaded owner is required to identify and align sustainability goals to overall migration goals. The owner also owns the sustainability portion of the business case for migration. The owner is responsible for capturing sustainability-relevant data before, during, and after the migration.

Suggestion 1.1.2: Include sustainability impact in the business case along with TCO and return on investment (ROI) calculations.

Sustainability impact should be included in the final business case for the migration. Alignment with organizational sustainability goals can be showcase here. You can use tools such as AWS Migration Evaluator to highlight estimated carbon emission reductions when migrating to AWS with right-sized workloads.

MIG-SUS-02: Does your migration strategy include assessing an AWS Region to meet business and sustainability goals?

An important decision that needs to be made prior to migrating to AWS is the Region you select to deploy and migrate your workloads. This choice significantly affects KPIs, including latency, cost, and carbon footprint. To effectively improve these KPIs, you should choose Regions for your workloads based on both business requirements and sustainability goals.

MIG-SUS-BP-2.1: Choose a Region for the workloads you plan to migrate based on your business requirements and your sustainability goals

This BP applies to the following best practice areas: Region selection

It can be challenging to select the optimal Region for a workload to migrate. This decision must be made carefully, as it has an impact on compliance, cost, performance, services available for your workloads, and sustainability goals.

Implementation guidance

Suggestion 2.1.1: Shortlist potential Regions for your workloads based on your business requirements.

If your workload contains data that is bound by local regulations, shortlist Regions that comply with those regulations. This applies to workloads that are bound by data residency laws, where choosing an AWS Region located in that country is mandatory.

There are four key business factors to consider when evaluating and shortlisting each AWS Region for a workload: compliance, latency, cost, and services and features. Evaluating all these factors can make coming to a decision complicated. Try to shortlist potential Regions based on these KPIs.

Suggestion 2.1.2: Select Regions to support your sustainability goals as part of your migration strategy.

After shortlisting the potential Regions, the next step is to choose Regions near Amazon renewable energy projects, or Regions where the grid has a lower published carbon intensity.

For more detail, see the following:

- How to select a Region for your workload based on sustainability goals.
- Renewable Energy Methodology
- Understanding your carbon emission estimations
- Video - Architecting sustainably and reducing your AWS carbon footprint

Mobilize

The next step in preparing your workforce and resources to migrate your enterprise at scale is to break down the **mobilize activities** into different workstreams. Although the goal of the mobilize phase is the migration of business applications, most prescriptive guidance and answers on achieving your sustainability goals are found here.

MIG-SUS-03: How do you define and optimize cloud resources during migration so that you become more energy efficient by minimizing idle resources?

As a part of your migration planning, one of the important tasks is to define the key workload performance matrix based on your assessment. This plays the significant part in deciding how you would like to migrate the target workload, instead of replicating as-is on-premises configuration. Optimizing cloud resources by removing idle resources helps lower carbon emissions without compromising business requirements.

MIG-SUS-BP-3.1: Focus on efficiency across all aspects of infrastructure

For example, during migration, verify that you use only the required resources, instead of trying to match with source on-premises capacity. This BP applies to the following best practice areas: Alignment to demand

Implementation guidance

Suggestion 3.1.1: Review the on-premises capacity to plan the workload requirements for your target environment.

During migration assessment, define the workload performance metrics for client requests. To optimize cloud resources, take advantage of elasticity in the cloud so you can meet the increasing demand of the migrating workload. As part of your assessment, review and analyze the following:

- How to respond to the overall demand, rate of change, and required response time, which could potentially help to minimize the environmental impact. Implement dynamic scaling and automation practice aligning to your SLAs to remove excess capacity and assign only needed capacity.
- Identify redundancy, underutilization, and potential decommission targets, and plan how you can consolidate the redundant content, scale down underutilized resources, and decommission unused assets.

Suggestion 3.1.2: Use proven workflow templates to migrate enterprise applications.

The way you plan your migration can help you identify the automation opportunity to improve the efficiency during migration process. For example:

- You can use Migration Hub Orchestrator templates to create a migration workflow that can be customized to fit your unique migration requirements, instead of manually performing all the tasks.
- You can leverage services like AWS Control Tower to get started. Control Tower helps you set up a multi-account environment and automate the creation of AWS accounts with built-in governance.

Suggestion 3.1.3: Evaluate your migrated workload to consider and configure auto scaling mechanism.

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance. Define the metrics that have the most relevance to your application's performance to meet changes in demand, and ensure that workloads can scale down quickly and easily during periods of low user load.

Suggestion 3.1.4: Define and update service-level agreements (SLAs).

- Review and optimize your workload service-level agreements (SLA) based on your sustainability goals to minimize the resources required to support your workload, while continuing to meet business needs. Consider your SLA requirements as part of your design and architecture as well.
- Define and update SLAs of the migrating workload, such as:
  - Availability or data retention periods, to minimize the number of resources required to support your workload. For more detail, see SUS02-BP02 Align SLAs with sustainability goals
  - Power off the workload during non-functional period. You can use Instance Scheduler on AWS to do this, which automates the starting and stopping of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Relational Database Service (Amazon RDS) instances.
  - Identify if use of a messaging component can lead to relaxed service-level requirements that can be met with fewer resources. Employ batching requests, where possible, for optimal use of resources. For more detail, see SUS03-BP01 Optimize software and architecture for asynchronous and scheduled jobs.

For more detail, see the following:

- Throttle API requests for better throughput
- Working with Service auto scaling
- AWS Summit SF 2022 - Optimizing your AWS infrastructure for sustainability
- Capacity management made easy with Amazon EC2 Auto Scaling
- Architecting sustainably and reducing your AWS carbon footprint

MIG-SUS-04: Do you consider sustainability when selecting and prioritizing applications for migration and modernization?

Have sustainability in mind when deciding which applications to migrate and modernize. To do this, first identify metrics that can act as a stand-in for your application's sustainability. Then, use these metrics to initiate migration and modernization projects that improve the application's sustainability.

MIG-SUS-BP-4.1: Adopt metrics that can signal the sustainability of your application

This BP applies to the following best practice areas: Process and culture

Adopt metrics to understand what you have provisioned and how those resources are consumed. Evaluate potential improvements, and estimate their potential impact, the cost to implement, and the associated risks. Measure improvements over time to study trends and the impacts of any migration and modernization initiatives.

Implementation guidance

Suggestion 4.1.1: Adopt sustainability metrics relevant to the application.

Understand the resources provisioned by your application to complete a unit of work. Leverage monitoring tools to define proxy metrics, business metrics, and sustainability key performance indicators (KPI) for your workloads. Documenting sustainability impact over time, including after migration and modernization, enables iterative improvement of your application over time.

You can also add other sustainability attributes that signal the efficiency of your application. An example of this is the version of your operating systems, runtimes, middleware, libraries, and applications. Keeping your workloads up-to-date can improve workload efficiency, and capturing this information keeps stakeholder informed. Another example of this is an indicator to note if you are using Graviton-based instances improve the performance efficiency of your application. For more detail, see AWS Well-Architected Framework - Sustainability Pillar.

For more detail, see Measure and track cloud efficiency with sustainability proxy metrics, Part II: Establish a metrics pipeline.

Suggestion 4.1.2: Introduce organizational dashboards to share sustainability metrics with stakeholders.

Leverage automation to report, visualize, and enforce sustainability metrics for your application. Introduce organizational dashboards for sustainability that can be shared with application owners and other stakeholders, including key organizational functions such as a Cloud Center of Excellence (CCoE) and Application Review Board (ARB). Make sustainability metrics a part of your architectural decisions.

MIG-SUS-BP-4.2: Include sustainability metrics in the application portfolio analysis to drive migration and modernization initiatives

This BP applies to the following best practice areas: Process and culture

Include sustainability metrics when scoping and prioritizing applications for migrations. Sustainability may be excluded if alignment from migration goals and organizational goals is missing.

Implementation guidance

Suggestion 4.2.1: Include sustainability metrics in the application portfolio analysis.

Establish a sustainability improvement process and adopt methods that can rapidly introduce sustainability improvements and keep your workloads up-to-date.

Including sustainability metrics in the application portfolio analysis assures that relevant data is captured early and is included in architecture and implementation considerations. These metrics capture the business and technical value of retiring applications, and prioritize rightsizing and application scaling to meet infrequent demand.

MIG-SUS-05: How do you implement efficient workload design to support your sustainability goals?

Compute and storage services make up the foundation of many customers, which brings great potential for workload design consideration that can improve the energy efficiency of the migrating workload.

MIG-SUS-BP-5.1: Implement efficient workload design by leveraging the underlying infrastructure.

For example, right-size the workload for the target state before migrating to minimize idle resources, and to avoid over provisioned capacity. This BP applies to the following best practice areas: Hardware and services

Implementation guidance

Suggestion 5.1.1: Select the most efficient hardware and services for your workload migration.

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. For example, Graviton3 provides up to 60% less energy for the same performance as non-Graviton EC2 instances.

Suggestion 5.1.2: Gain insight into workload performance.

Gain insight into workload performance metrics like CPU utilization, memory utilization, network utilization, and disk and usage patterns to perform the right alignment of the cloud resource.

- Key matrices to consider are the following:
  - If the workload is idle for a long time, it's a good sign to investigate if this workload can retire instead of migrating it.
  - Understand your average CPU and memory utilization, and identify resources that are underutilized. Rightsize those instances to reduce your carbon footprint.
  - Identify the network and storage performance, such as I/OPS, and size for the target workload. Analyze the overall demand, rate of change, and required response time to rightsize the throttle or buffer required.
  - For fault-tolerant, flexible, and stateless workloads that can trade-off with minimal interruption, adopt Spot Instances in your design.
  - Evaluate your migrated workload continually using AWS Compute Optimizer and AWS Trusted Advisor, and proactively make rightsizing adjustments on your workload.

Suggestion 5.1.3: Consider managed services in your workload design.

Remove the need for you to run and maintain physical servers, as AWS operates at scale and is responsible for their efficient operation. For example, instead of migrating your virtual machine or container to Amazon ECS running on Amazon EC2, consider using a service like Amazon Fargate, where you can run containers without having to manage the servers, or package and deploy Lambda functions as container images.

Containerize and migrate existing applications using , for migrating and modernizing Java and .NET web applications into container format. Use managed services to operate more efficiently in the cloud.

Suggestion 5.1.4: Configure your instance tenancy, which defines how EC2 instances are distributed across physical hardware.

Tenancy provides the opportunity to further optimize the workload. Migrating to shared instances instead of migrating to dedicated underutilized instances or hosts can be more beneficial when working towards the sustainability goal.

For more detail, see the following:

- Right Size Before Migrating
- Getting started with Graviton
- Streamline selection and right size EC2 with AWS Compute Optimizer
- Effortless migration - Is your app already Graviton-ready?
- Design patterns for success in serverless microservices
- Lift and shift a web application to serverless

MIG-SUS-06: How do you take advantage of software and architecture patterns for workloads you are going to migrate to support your sustainability goals?

Software and architecture patterns can be used to influence utilization of resources while migrating workloads. The aim is to use patterns that maximize utilization so that resources consumed for the workloads are minimized. Idling of resources due to user behavior or nature of workload can also be minimized by applying appropriate software and architecture patterns. on-premises workloads are not designed to take advantage of AWS cloud features that can optimize utilization of resources, like autoscaling. These workloads can have multiple instances of software or services running at multiple locations or are overprovisioned all the time to cater to peak demand. Some workloads are running all the time, even when not in use. Using the 7 Rs can optimize resource usage. Adopt patterns and architecture to consolidate underutilized components to increase overall utilization. Retire components that are no longer required.

MIG-SUS-BP-6.1: Identify environments and workloads that can be consolidated or retired

This BP applies to the following best practice areas: Software and architecture

Implementation guidance

Suggestion 6.1.1: Consolidate environments and workloads in the AWS Cloud.

When moving workloads from multiple on-premises environments or in merger and acquisition cases, consolidating environments on AWS leads to optimal usage of resources and eliminates duplicate functionality. Identify workloads during the assess stage that can be eliminated or consolidated. Identify multiple instances of services or applications running at multiple locations on-premises.

Retire workloads that are not used. Use automated tools such as Migration Evaluator to identify workloads that are not being used.

MIG-SUS-BP-6.2: Identify workloads that can use efficient software and architecture patterns to maintain consistent high utilization of deployed resources

This BP applies to the following best practice areas: Software and architecture

Implementation guidance

Suggestion 6.2.1: Optimize software and architecture for asynchronous and scheduled jobs.

Identify applications and workloads that can benefit from software and architecture patterns to maintain consistently-high utilization of deployed resources while migrating applications.

While migrating applications, evaluate use of integration patterns to scale the processing independently of the receiving of messages, which reduces resource utilization. Identify if use of a messaging component can lead to relaxed service level requirements that can be met with fewer resources. Employ batching requests where possible for optimal use of resources, as batching provides consistent usage. Scheduling the batch jobs reduces idle time for resources. For detail, see SUS03-BP01 Optimize software and architecture for asynchronous and scheduled jobs.

MIG-SUS-BP-6.3: Analyze your data access patterns and data lifecycle processes, and evaluate how you can become more efficient and sustainable in your data management

This BP applies to the following best practice areas: Software and architecture

Implementation guidance

Storing and accessing data efficiently, in addition to reducing idle storage resources, results in a more efficient and sustainable architecture. When migrating data, understand how data is used within your workload, consumed by your users, transferred, and stored. Use software patterns and

architectures that best support data access and storage to minimize the compute, networking, and storage resources required to support the workload.

Suggestion 6.3.1: Define and implement a data lifecycle process for data in your object store.

Design a data lifecycle management process based on your data access patterns, observed in your on-premises facility. That process either removes data that is no longer required or archives data into less resource-intensive storage. While migrating data, implement a data lifecycle policy. For more detail, see Best practice 15.4 – Implement data retention processes to remove redundant data from your analytics environment.

Suggestion 6.3.2: Evaluate use of columnar data formats and compression.

While migrating data, evaluate if columnar data formats like Parquet and ORC can be used. These formats require less storage capacity compared to row-based formats like CSV and JSON.

- Parquet consumes up to six times less storage in Amazon S3 compared to text formats. This is because of features such as column-wise compression, different encodings, or compression based on data type.
- You can improve performance and reduce query costs of Amazon Athena by 30–90 percent by compressing, partitioning, and converting your data into columnar formats. Using columnar data formats and compressions reduces the amount of data scanned.

MIG-SUS-BP-6.4: Understand and influence business requirements, and optimize areas of code to reach your sustainability goals

This BP applies to the following best practice areas: Software and architecture

Understanding your sustainability goals is the first step to focusing on the factors needed to meet those goals. Defining such criteria involves adopting metrics that can be used to measure and evaluate your current sustainability posture, report progress against goals, and accelerate improvements. By analyzing the current environmental impact of the underlying cloud-based infrastructure, you can quantify the tradeoffs and changes required to meet your sustainability objectives.

Implementation guidance

Suggestion 6.4.1: Define criteria to measure and understand your sustainability impact after your migration.

Post-migration, you can use sustainability proxy metrics for your monitoring scenarios. Proxy metrics allow architecture teams to evaluate correlated improvements made to a workload instead of real-time carbon metrics. Defining proxy metrics across compute, storage, and network infrastructure can help you understand how infrastructure changes can impact sustainability results.

Example proxy metrics include vCPU minutes for compute, GBs provisioned for storage, and GBs transferred for network traffic. Proxy metrics combined with business metrics can define sustainability KPIs, which can be used to drive sustainability optimizations while keeping business needs in focus. One example would be to measure vCPU minutes per transaction and define an improvement goal to minimize this metric. Business stakeholders would have to weigh the cost, as reducing vCPUs could ultimately become detrimental to delivering on business needs. When running workloads in AWS, the change in these measured resources correlates with a similar change in cost (except as noted in the following), making overall infrastructure spend a useful proxy metric.

By agreeing on a set of sustainability metrics, the architect team can evaluate different technical approaches to reduce environmental impact.

For more detail, see the following:

- Cloud sustainability
- Evaluate specific improvements
- Measure and track cloud efficiency with sustainability proxy metrics, Part II: Establish a metrics pipeline
- Best Practices from IBM and AWS for Optimizing SaaS Solutions for Sustainability
- re:Invent 2022: Delivering sustainable, high-performing architectures

Suggestion 6.4.2: Consider using Amazon CodeWhisperer to reduce your cloud costs, improve your application performance, and reduce your carbon emissions attributable to your workload.

Migrate

In the migrate phase, we ensure the migration proceeds as planned, monitor the migration process, and have a plan in place to rollback in case any issue encountered during the migration. During migration, you can scale your resources corresponding to the volume of data to be migrated. Furthermore, you can adopt best practices that can reduce interim resource consumption during your migration.

MIG-SUS-07: Does your on-premises to AWS data migration strategy consider sustainability?

Data makes up the large portion of the scope of many workload migrations. Identifying and optimizing the data storage with latest technologies helps improve the power efficiency and reduce carbon footprint.

MIG-SUS-BP-7.1: Implement data management practices

This BP applies to the following best practice areas: Data

Data management is a continuous process and should be implemented during and after the migration. With the latest storage technologies, it provides the opportunity to configure and provision sufficient storage without compromising the business needs.

Implementation guidance

Suggestion 7.1.1: Avoid over-provisioning for storage system to influence your environmental impact.

- Perform application discovery to identify data characteristics and access patterns that can be supported by storage technology.
- You can use shared file systems or storage that allows for sharing data to one or more consumers without having to copy the data. For example, you can have a shared drive to store common files instead of copying those common files to each VM.
- After migrating the workload, from time to time, analyze data access and date movement to identify opportunities to become more efficient. When opportunities are found, change the lifecycle by moving to other storage classes or deleting unneeded data.
- Use technologies that support data access and storage patterns. For example, migrating data to other object storage types eliminates provisioning the excess capacity from fixed volume sizes on block storage. For more detail, see SUS04-BP02 Use technologies that support data access and storage patterns.

Suggestion 7.1.2: As part of your per-migration planning evaluate your current recovery time objective (RTO) and recovery point objective (RPO).

- Design your backup strategy based on your actual business requirements. Avoid backing up non-critical data that has no business value, and detach volumes from clients that are not used before considering to migrate those workloads. For more detail, see SUS04-BP08 Back up data only when difficult to recreate.
- Use an automated solution or managed service to back up business-critical data. AWS Backup is a fully-managed service that makes it easy to centralize and automate data protection across AWS services, in the cloud, and on-premises. Next to other capabilities, AWS Backup helps you become more sustainable. For example, you can use Backup to set an expiration on your manual snapshots.
- Set automated lifecycle policies to enforce lifecycle rules for the migrated data. For more detail, see SUS04-BP03 Use policies to manage the lifecycle of your datasets.
- If you are setting up disaster recovery for your migrating workload, evaluate your RTO and RPO, and see if you could meet the requirement using the backup data instead of replicating the entire data to the recovery site. For more detail, see AWS Elastic Disaster Recovery.

Suggestion 7.1.3: Choose the right migration tool, and scale your resources corresponding to the volume of data to be migrated.

- AWS provides migration services like AWS Database Migration Service and AWS Application Migration Service. You may be able to scale down the replication instance type selected if the amount and velocity of the ongoing data is much smaller than the amount of historical data.
- Another alternative is to use a serverless migration tool like AWS DMS Serverless.
- Here are some other options to choose from to migrate your storage with their key characteristics.

Migrate your storage	Key Characteristic
AWS DataSync	Simplify, automate, and accelerate data movement to and from AWS Storage, as well as between AWS Storage. Easily manage data movement workloads with bandwidth throttling, migration scheduling, task filtering , and task reporting with a fully managed

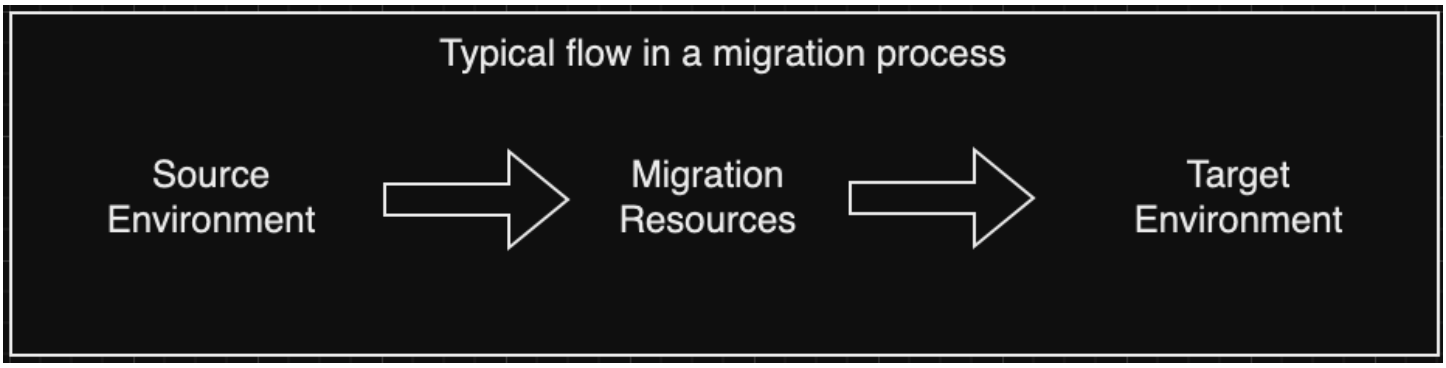
Migrate your storage	Key Characteristic
	service that seamlessly scales as data loads increase.
AWS Transfer Family	Simply and seamlessly move your files to Amazon S3 and Amazon Elastic File System (Amazon EFS) using SFTP, FTPS and FTP protocol. Store information in Amazon S3 or Amazon EFS, manage workflows, and initiate automated, event-driven tasks with a fully-managed, low-code service. Quickly scale your business-to-business (B2B) file transfers for each line-of-business user.
AWS Snow Family	Collect and process data at the edge, and migrate data into and out of AWS through physical devices and capacity points. Device options range to optimize for space- or weight-constrained environments, portability, and flexible networking options.

For more detail, see the following:

- Data lifecycle management
- Amazon S3 Intelligent-Tiering
- I/O characteristics and monitoring
- Optimizing your AWS Infrastructure for Sustainability, Part III: Networking
- Top 10 Data Migration Best Practices
- AWS Summit SF 2022 - Optimizing your AWS infrastructure for sustainability
- Amazon EBS and Snapshot Optimization Strategies for Better Performance and Cost Savings

MIG-SUS-08: Are you adopting practices that can reduce interim resource consumption during the migration?

During a migration, your consumption of resources may increase due to the provisioning of resources in both the source and target environments. The increase in consumption is often referred to as a *double bubble*. In addition, your consumption may also increase due to provisioning of migration resources, such as the networking between your source and target environments, SFTP servers, AWS Application Migration Service (MGN), or AWS Database Migration Service (DMS).



Typical migration process flow

You can reduce the resource consumption during the migration either by reducing the resources deployed or by reducing the duration of their deployment.

$$\text{Additional resource consumption during a migration} = \sum_{i=0}^{n1} ResourceinTargetEnvironment \times DurationofDeployment + \sum_{i=0}^{n2} ResourceinMigrationProcess \times DurationofDeployment$$

Additional resource consumption equation

MIG-SUS-BP-8.1: Adopt methods that can reduce interim resource consumption during the migration

This BP applies to the following best practice areas: Process and culture

Implementation guidance

Suggestion 8.1.1: Reduce interim resources created in the target environment.

- Reconsider the migration of development and other non-production environments, as these can be rebuilt when required in AWS. If you decide on migrating your non-production environments,

revisit the portions of the environment that need to be migrated. For example, you may choose to migrate only some of the databases in a database server.

- If you decide to migrate your build environment, increase the utilization of these environments.
- Use managed device farms to test new features on a representative set of hardware.
- During the migration, consider the impact on sustainability of day-to-day operations. For example, consider avoiding frequent backups of the target environment and setting up HA or DR during the migration. You can also reduce the retention period of logs or backup snapshots taken during the migration.

Suggestion 8.1.2: Reduce interim resources used in the migration process.

- During a migration, you typically have to migrate historical and on-going data. Historical data refers to the data that was created prior to the start of the migration. On-going data refers to the new data that is generated in the source environment at the time of the migration until the cutover. The resource needs for the migration of historical data may differ from that of the on-going data. Choose the right migration process and tool, and also scale your resources corresponding to the data to be migrated. For example, in the case of AWS DMS and AWS MGN, you may be able to scale down the replication instance type selected if the volume and velocity of the ongoing data is significantly less to volume of historical data. Another alternative is to use a serverless migration tool like AWS DMS Serverless that can automatically scale based on the volume of data being migrated.
- Share your migration resources if possible. Some migration tools let you share migration resources. An example of this is AWS MGN, which automatically shares the replication instance with multiple source servers being migrated.
- For migration resources that cannot be scaled easily, such as the networking resources between your data center and AWS Cloud, consider flattening the demand curve using buffering and throttling to reduce the required provisioned capacity for the workload. For example, you can throttle your network in AWS MGN when migrating your servers to AWS.
- Review the need to include migration resources in your day-to-day operations. For example, avoid including AWS MGN replication servers in your backup strategy. If you are capturing logs for migration resources, you can consider reducing the retention period for these logs.

Suggestion 8.1.3: Reduce duration of deployment for the interim resources created during the migration.

- Consider selecting a partner who has the technical expertise and experience migrating to AWS
- Create a cross-functional cloud-enablement team to implement the governance, best practices, training, and architecture needed for cloud adoption. The team will define tools, processes, and architectures that establish the organizations cloud operating model. In addition, it will coordinate with stakeholders across different units such as infrastructure, security, applications, and business to alleviate obstacles in a migration.
- Explore tooling that can facilitate your migration and can automate and expedite aspects of the migration such as discovery, project management, and testing.
- Train staff on tools and processes early in the migration to give them the required skillset.
- Build a robust migration factory consisting of people, tools, and processes that help streamline your migration. Operate in an agile fashion increases the velocity of the applications being moved to AWS.
- Assess the application to be migrated and satisfy all prerequisites a few weeks prior to the migration.
- Start small to build experience, find patterns, and create blueprints. Prioritize workloads and run the migration in waves with short migration cycles. Create reusable blueprints for common workload patterns that increase the velocity of the migration. Empower your team to automate the migration steps.

For more detail, see the following:

- Strategy and best practices for AWS large migrations
- A beginners' guide for Finance and Operations teams in their cloud migration journey

Best practice arranged by migration phase

Assess Phase

Operational excellence pillar

- ???

Security pillar

- ???
- ???
- ???
- ???
- ???

Reliability pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Performance efficiency pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Cost optimization pillar

- ???
- ???

Sustainability pillar

- ???
- ???

Mobilize Phase

Operational excellence pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???

- ???

Security pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Reliability pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Performance efficiency pillar

- ???
- ???
- ???
- ???
- ???

Cost optimization pillar

- ???
- ???
- ???
- ???
- ???
- ???

Sustainability pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

Operational excellence pillar

- ???
- ???
- ???

Security pillar

- ???
- ???
- ???
- ???
- ???

Reliability pillar

- ???

Performance efficiency pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Cost optimization pillar

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Sustainability pillar

- ???
- ???

Best practices arranged by pillars

Operational excellence pillar best practices

Assess Phase

- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???
- ???
- ???

Security pillar best practices

Assess Phase

- ???
- ???
- ???

- ???
- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???
- ???
- ???
- ???
- ???

Reliability pillar best practices

Assess Phase

- ???
- ???

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???

Performance efficiency pillar best practices

Assess Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Cost optimization pillar best practices

Assess Phase

- ???
- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Sustainability pillar best practices

Assess Phase

- ???

- ???

Mobilize Phase

- ???
- ???
- ???
- ???
- ???
- ???
- ???
- ???

Migrate Phase

- ???
- ???

Conclusion

This lens helps you migrate your workloads to the AWS Cloud. The whitepaper discussed best practices for each of the three-migration phase, and how they can be implemented across the six pillar of the AWS Well-Architected Framework. The whitepaper also discussed the AWS Well-Architected Framework pillars through a migration lens, providing you with a set of questions to consider when migrating a new workload to AWS.

As this migration landscape continues to evolve with new services, maturation of tools, process, and adoption, Well-Architected lenses are updated to help provide you with the resources and knowledge needed to build and operate workloads on AWS.

Contributors

Contributors to this document include:

- Ebrahim (EB) Khiyami, Cloud Optimization Success Solutions Architect, Amazon Web Services
- Mike Kuznetsov, Principal Migrations Solutions Architect, Amazon Web Services
- Geoffrey Burdett, Senior Solutions Architect Migrations Modernization
- Hemant Ahire, Principal Solutions Architect, Amazon Web Services
- Tejpreet Reen, Senior Migrations Solutions Architect, Amazon Web Services
- Chris Baker, Senior Product Engineer, Amazon Web Services
- Damien Renner, Senior Consultant, Amazon Web Services
- Simon Champion, Senior Cloud Migration Specialist, Amazon Web Services
- Kiran Randhi, Principal Partner Solutions Architect, Amazon Web Services
- Pavan Yanamadala, Solutions Architect, Migrations, Amazon Web Services
- Prashanth Nalubandhu, Migration Principal Solutions Architect, EntTrans, Amazon Web Services
- Sanket Nasre, Senior Migration Solutions Architect, Amazon Web Services
- Nabil Mohamed, Senior Migrations Solutions Architect, Amazon Web Services
- Peter Giuliano, Senior Migration Solutions Architect, Amazon Web Services
- Harpreet Virk, Senior Migrations Solutions Architect, Amazon Web Services
- Kiran Kuppa, Principal Migration Solutions Architect, Amazon Web Services
- Matt Saner, Senior Manager, Security Solutions Architect, Amazon Web Services
- Zulia Shavaeva, Security Consultant, Amazon Web Services
- Anil Sharma, Senior Partner Solutions Architect, DACK, Migration, Amazon Web Services
- Bill Evans, Senior Customer Solutions Manager, Amazon Web Services
- Franz Stefan, Solutions Architect, Migration and Modernization, Amazon Web Services
- Phurba Sherpa, Senior Partner Solutions Architect, Migrations, Amazon Web Services
- Vineedh George, Senior Migrations Solutions Architect, Amazon Web Services

Document revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Migration Lens first published.	January 24, 2024

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.