Building your Cloud Operating Model

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building your Cloud Operating Model

Table of Contents

Building your Cloud Operating Model

Amazon Web Services (contributors)

August 2023 (document history)

The cloud is an enabler for transformations in business and information technology. However, as new cloud capabilities and services accelerate alongside existing on-premises environments, organizations need to balance current responsibilities against the transition to new ways of working. This transformation unlocks the benefits of the cloud, but it needs to be done with the least amount of disruption to existing operational practices.

After reviewing trends and approaches used by our most successful customers, we identified that having a well-defined Cloud Operating Model provides a way to balance where you are today with where you want to go tomorrow, which leads to faster adoption and higher transformational value.

This strategy document presents the AWS definition of a Cloud Operating Model and provides prescriptive guidance for organizations that are seeking to build their own Cloud Operating Model.

Contents

Introduction

This document provides a definition of the Cloud Operating Model and core capabilities that organizations should focus on when building their own model.

What is a Cloud Operating Model, and why do you need one?

We use the phrase *Cloud Operating Model* to refer to the operating model within an IT organization that is used to build, mature, and optimize one or more cloud environments. The ability to build maturity across a number of capabilities that move the IT organization in the same direction as the overall transformation strategy is becoming ever more important. We coach customers to use the opportunity of defining their Cloud Operating Model to explore cloud-first ways of working that will provide a solid foundation for the continuous evolution of their whole organization. Our experience shows that if you don't spend time on this aspect of your cloud journey, the initiative will stall and your organization will struggle to realize value from your transformation efforts.

This view is backed up by the report Predicts 2023: Collaborate, Automate and Orchestrate to Optimize Costs and Value During the Economic Crisis on the Gartner website, in which they summarize that infrastructure and operations leaders should use workload orchestration, automation, and collaborative practices to achieve the goal of delivering value while optimizing costs.

However, you cannot just implement these recommendations. They require an understanding of your current capabilities, how these capabilities are organized to meet operational requirements, and a plan to increase maturity across your teams. In effect, you need to understand your Cloud Operating Model so you can position your organization to execute on the cloud strategy. Your Cloud Operating Model must then evolve over time as capabilities continue to mature and your organization gains more value from the transformation.

Key Concepts

To start, let's define the key concepts used in this paper, because the terminology and approach can differ across cloud providers.

Capabilities

We use *capabilities* as a collective term that covers people, process, and technology. Because there is an inclination to only focus on the technology aspects of the cloud and deprioritize the

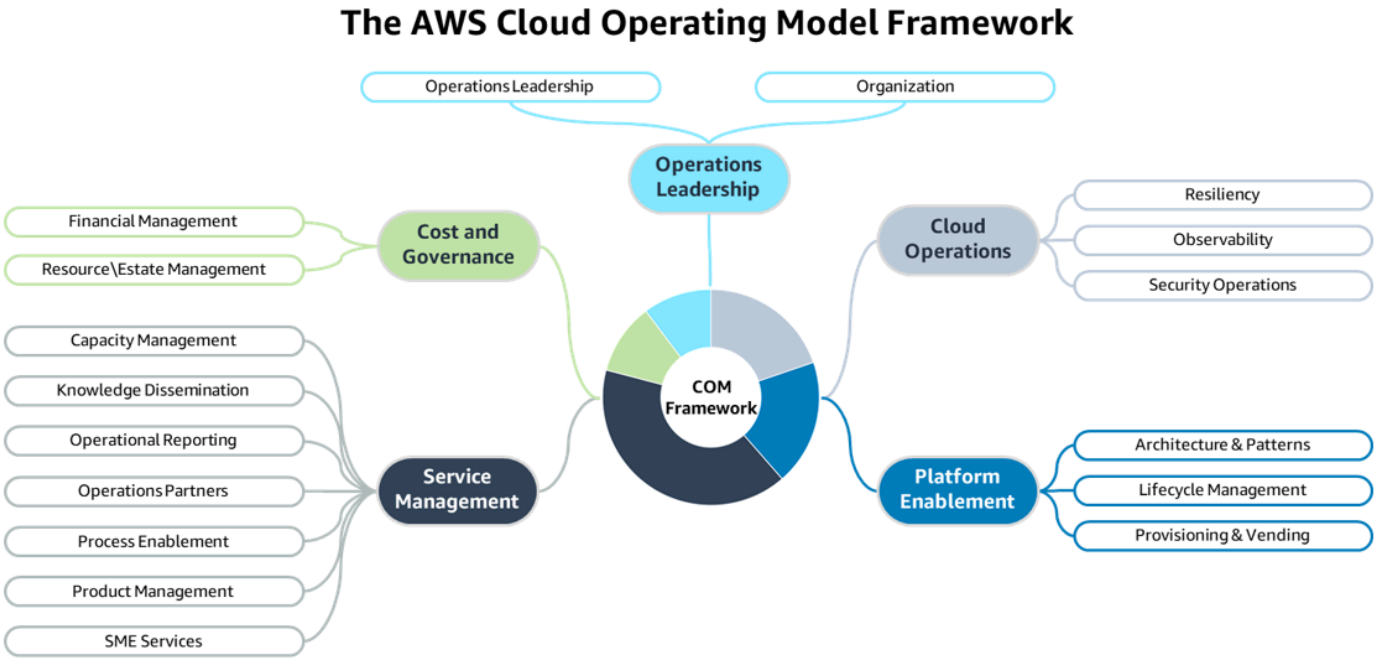
people and process angles, the term *capabilities* joins these three aspects to describe *the ability to do something*. This collective term also simplifies the identification of the people, process, and technology changes required at each point in your cloud journey.

It's a continuous journey

Defining a new operating model is not a one-time exercise. You need to build a model and supporting mechanisms that can serve the needs of the organization today, but, as cloud capability matures, can evolve and continuously improve over time to meet changing needs.

The AWS Cloud Operating Model Framework

The AWS Cloud Operating Model (COM) Framework consists of 73 capabilities, grouped into 17 domains and 5 perspectives, as illustrated in the following diagram.



Perspectives	Operations Leadership	Cloud Operations	Platform Enablement	Service Management	Cost and Governance
Domains	• Operations leadership	• Resiliency • Observability	• Architecture and patterns	• Capacity management	• Financial management

	• Organization	• Security operations	• Lifecycle management • Provisioning and vending	• Knowledge dissemination • Operational reporting • Operations partners • Process enablement • Product management • SME services	• Resource/estate management

Using a framework like ours supports the development of your Cloud Operating Model by providing consistency as you understand, organize, design, implement, and mature your organization in line with the goals of your transformation journey.

A Cloud Center of Excellence is not a Cloud Operating Model

A Cloud Center of Excellence (CCoE) has become a well-known concept when migrating to the cloud or running workloads in the cloud. However, the CCoE is not a Cloud Operating Model. It is a cross-organizational leadership function that supports successful cloud adoption across the enterprise through alignment, enablement, and automation; whereas the Cloud Operating Model is the operating model within an IT organization that is used to build, mature, and optimize one or more cloud environments.

The following table summarizes the differences between the two terms.

	Cloud Operating Model	**Cloud Center of Excellence**
Use case	When you have significant workloads in the cloud, but	When progress has stalled or your organization needs

	you aren't meeting the key performance indicators (KPIs), business outcomes, or values you were expecting to gain from the cloud over tradition al on-premises approaches	to enable the adoption of the cloud and new ways of thinking, deciding, behaving, and innovating by standardi zing best practices for autonomous work
Teams included	IT and business teams	Cross-functional, multi-ski lled resources aligned to the Cloud Leadership Team, Cloud Business Office, and Cloud Platform Engineering
Focus	Supporting, enabling, and optimizing cloud workloads by maturing your organizat ion's existing operating model and capabilities to adopt cloud-first ways of working	Establishing an entity to accelerate and build technical and cultural foundations to enable migration and innovation
Expected outcomes	Greater operational efficienc ies, reduced cost of IT delivery, reduced risk, greater agility, and more innovativ e technical capabilities and services	Accelerated and sustainab le cloud adoption; empower cloud-driven products teams with a self-service environme nt, minimized disruptions, greater adoption of standardi zed approaches and patterns, and increased productivity that accelerates delivery; optimized agility and value of cloud; scale through ongoing risk mitigation

There are similarities in the capabilities required by a Cloud Operating Model and a CCoE. However, because the CCoE focuses on the move to the cloud, it requires more capabilities, such as People Enablement and Organizational Acceleration. To be successful, a CCoE has to fit and work

within the existing operating model, but the two are distinct concepts and the two terms aren't interchangeable.

Managing your workforce

We often work with customers who are transitioning from on premises to cloud environments. This means that at the point of engagement with AWS, the majority of their infrastructure and workloads are still on-premises and still require management, often by the same teams that are part of the migration or transformation program. In the report 25 Amazing Cloud Adoption Statistics [2023]: Cloud Migration, Computing, and More (Zippia.com, June 22, 2023) the writer notes that 94 percent of enterprises surveyed use some form of cloud services. However, the same report says that by 2026 only 45 percent of enterprise IT budget will be on cloud expenses. This means that despite ubiquitous cloud services, large on-premises estates will continue to exist and will need to be managed. Therefore, many enterprises organize their workforce to deliver both cloud and non-cloud services. Building your Cloud Operating Model incrementally means that you can focus on what's needed now as well as what's coming next, and adapt as you go along to ensure that you are managing your workforce in a way that's sustainable to the teams involved.

Vision

As highlighted in the previous section, our definition of a Cloud Operating Model is one that builds, matures, and optimizes one or more cloud environments. It does this by maturing the existing (IT) operating model to adopt and be adept in utilizing cloud-first ways of working that support your targeted business outcomes.

We have observed two common challenges in helping our customers establish their Cloud Operating Models: knowing where to focus, and how to maintain momentum in the transformation. It's not uncommon for organizations to make several attempts before they establish a model that is rewarding to work in and that delivers results and value to the organization.

For this reason, the first stage of the AWS Cloud Adoption Framework (AWS CAF) is Envision:

> *[The] Envision phase focuses on demonstrating how the cloud will help accelerate your business outcomes. It does so by identifying and prioritizing transformation opportunities across each of the four transformation domains in line with your strategic business objectives. Associating your transformation initiatives with key stakeholders (senior individuals capable of influencing and driving change) and measurable business outcomes will help you demonstrate value as you progress through your transformation journey.*

Most enterprises have their own way of defining the vision. At AWS, many teams establish a vision by defining a mission statement, a set of tenets the teams that are building capabilities will use to make their prioritization decisions, and a press release document with associated frequently asked questions (PR-FAQ). We use this approach to help our customers establish their Cloud Operating Model, but we adapt the approach to develop a Vision Document or charter that helps align the team that implements the Cloud Operating Model and provides a reference for teams they interact with.

Developing a Vision Document

The Vision Document includes a Mission Statement, Tenets, Drivers, and Outcomes. Each section should be defined with the leadership team, linked to the overall business strategy, and then published on an internal site (such as a wiki) for everyone to read.

The *Mission Statement* for a Cloud Operating Model should be linked to the value that the cloud is expected to bring to the organization. It should reflect the business drivers, priorities, strategy, and mandate for cloud usage.

Tenets are principles or beliefs that help teams align and bring everyone into an agreement around critical decisions. Here are some example tenets from our engagements with customers:

- We prioritize the many over the few. We prioritize the delivery of services that are useful to the entire organization over those for a single department or business unit.
- We aim for customer delight. We will create and run simple to use, highly scalable services that accelerate application teams by abstracting complexity and reducing the operational effort by minimizing handoffs.
- We prioritize automation and self-service. We help application teams go faster by prioritizing self-service and automation over manual processes.
- Speed matters: start small and iterate. We prioritize incremental delivery over extensive analysis.

The implied level of priority is from the first tenet to the last one. This order can help the team focus on the most important deliverables in support of wider business outcomes.

We recommend that you review and iterate on your Mission Statement and Tenets regularly and update them to reflect the requirements of your organization, your Cloud Operating Model, and your current level of cloud maturity.

Drivers and Outcomes provide the connections to business strategy. *Drivers* refer to the need to develop the Cloud Operating Model—what is driving the change—and how the Cloud Operating Model is influenced by them.

Outcomes are what you can expect from the change, or the first step in the journey that the changes will enable. These are forward-looking statements that capture expectations as the changes are implemented. Outcomes are useful to document to ensure that benefits are connected to technical results as well as business values.

When you build your Cloud Operating Model, we recommend that you use this approach to help identify the key problems to solve, the benefits to be delivered, and what the user experience should look and feel like.

If you are interested in taking a similar customer-centric approach, we recommend watching Richard Halkett's Working backwards: Amazon's approach to innovation presentation (AWS

re:Invent 2020), which describes Amazon's method to driving innovation and designing new products and services.

Regardless of which method you use, creating and publishing an agreed vision for the Cloud Operating Model that aligns to your targeted business outcomes is very important. The next step is to align that model to your current state of cloud adoption.

The Cloud Operating Model journey

The Vision Document has clarified your target state, but you must understand where you are in your cloud adoption journey to connect the vision to your current capabilities, and then understand the next steps. We have found that many customers focus on where they want to go, but it can be hard to see what the first step should be on that journey.

After the **Envision** stage, the AWS CAF defines three more phases:

- **Align**: Focus is on identifying capability gaps across the six AWS CAF perspectives (business, people, governance, platform, security, and operations), identifying cross-organizational dependencies, and surfacing stakeholder concerns and challenges.
- **Launch**: Focus is on delivering pilot initiatives in production and on demonstrating incremental business value. Pilots should be highly impactful. If and when they are successful, they will help influence future direction.
- **Scale**: Focus is on expanding production pilots and business value to desired scale and ensuring that the business benefits associated with your cloud investments are realized and sustained.

Because the aim of AWS CAF is to improve your cloud readiness, we will add another phase after the **Scale** phase:

- **Optimize**: Focus is on continually revisiting and improving the end solution to deliver additional business benefits.

Using these stages together with the AWS COM Framework helps you identify the capabilities that are important to you at each point in time. For example, if you are in the **Launch** phase, you might be more interested in the *Architecture and Patterns* capability than the *Resource/Estate Management* capability, which is more relevant during the **Scale** phase.

You carry out specific activities at each stage. For example, in the **Align** phase, you identify the capabilities you currently have and maturity level, then determine which capabilities you need to focus on first. If you are in the **Launch** phase, identifying pilot teams to develop the next level of maturity will be important. This requires planning, so we recommend that you define a roadmap.

Define a roadmap

You might have seen the following quote from Werner Vogels, VP and CTO at Amazon: "*You build it, you run it.*"

This was from the 2006 interview A Conversation with Werner Vogels: Learning from the Amazon technology platform (*ACM Queue*, Vol. 4, Issue 4, June 30, 2006). Werner talked about how teams at Amazon functioned (the operating model) and described breaking down walls between development and operations. Establishing cross-functional teams that have all the capabilities required to build, deliver, and support their product has become a requirement for a true digital transformation.

However, that digital transformation, which is supported by your Cloud Operating Model, is often seen as too much change to manage at one time. Instead, we consider the analogy of a journey with a roadmap that takes you to *"You build it, you run it"* as the destination. Each increase in the maturity of your capabilities moves you closer to your destination. By the time you have reached your destination, your organization will have developed a way to continually update the Cloud Operating Model to match changing business outcomes, and the roadmap is updated with the next destination.

To support this incremental approach, we recommend that you develop a roadmap that directly relates to your organization's vision (mission and drivers) and defines the steps (increases in maturity, guided by tenets) that are necessary to reach the destination (outcomes).

Implement the roadmap

When you have established the roadmap, you need to implement it. We have found that this is where customers face the next challenge: they have spent time *thinking*, and now have to move to *doing*. To connect your strategy to implementation, we recommend the following steps:

- Decide where and how to start
- Organize for success
- Establish Mechanisms to drive change
- Develop maturity incrementally

Decide where and how to start

This sounds easy, but with so much to achieve, finding a starting point is often a difficult and debated question. Organizations that are moving to the cloud have a lot to focus on, and the initiative can become overwhelming if it isn't put into context. Over the years, customer trends have evolved, but a consistent starting point is transformational leadership. Driving directives and strategy from the top down and creating the mission statement, tenets, and PR-FAQ enable middle management and individuals to make decisions autonomously, drive clarity, and drive business value from the cloud transformation. If you haven't carried out this exercise or something similar, we recommend it as your first task.

During this exercise, you should recognize that unlike other technology transformations, cloud transformation brings technology closer to the business. Technology is a lever that businesses use to achieve broader goals by enabling agility, stability, cost optimization, and similar outcomes. You must plan this transformation with technology and the business, working back from your organization's 3-5 year strategy, identifying goals along the way, and not being afraid to pivot when needed.

Organize for success

How your organization is structured to achieve cloud migration, adoption, and transformation goals will change as your organization matures. Understanding this, preparing, and being intentional is key to ensuring success.

Generally, at the beginning of the journey the largest teams work on the on-premises environment. Then, as cloud adoption grows, these teams migrate to build, mature, operate, and optimize the cloud platform, and your organization has to adjust to the new ways of working at each of these stages. We have observed that a difficult but important change happens when an organization has moved 5 to 10 percent of their workloads to the cloud (transitioning from the Launch phase to the Scale phase). At this point, an organization uses on-premises teams to operate cloud resources because the migration is not large enough to merit full-time changes, so these teams have to strike a balance between existing and new responsibilities. At the same time, the on-premises teams that are now being asked to operate cloud services require new skills, which involves a steep learning curve.

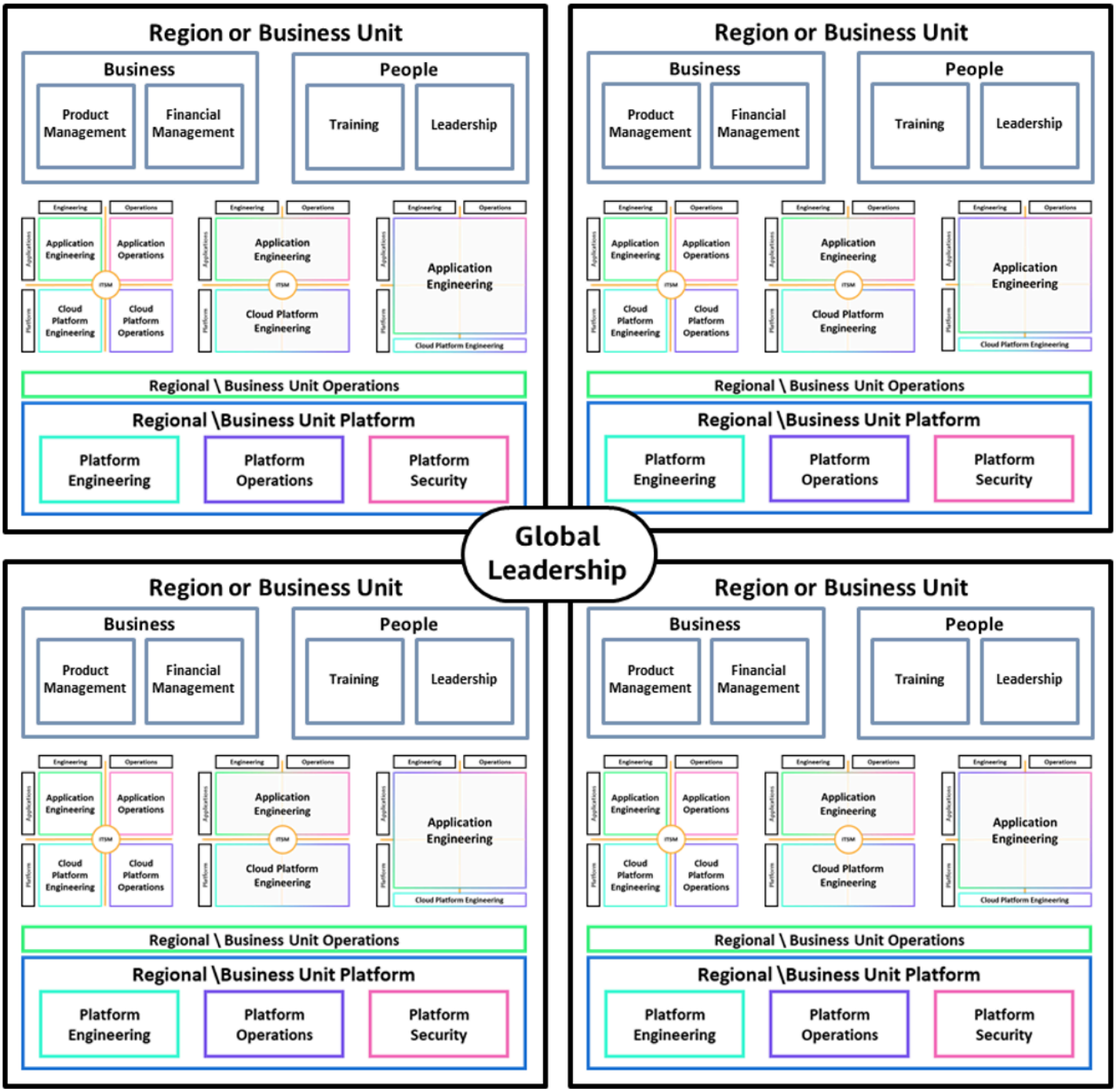
To understand your organization and develop a plan to enable these changes, we recommend that you look at the topology of teams across your IT organization. We use this method with customers to understand the arrangement and interlinking of functions within an IT organization, which is

often different from organizational structures, and then use the AWS COM Framework for guidance on how to organize to deliver against transformation stages and milestones. Any changes to the organizational structure that might be required are informed by this exercise.

The topologies we have used with customers include decentralized, centralized, and federated models. These expand on the operating model 2-by-2 representations covered in the AWS Well-Architected Framework, Operational Excellence Pillar.

Decentralized

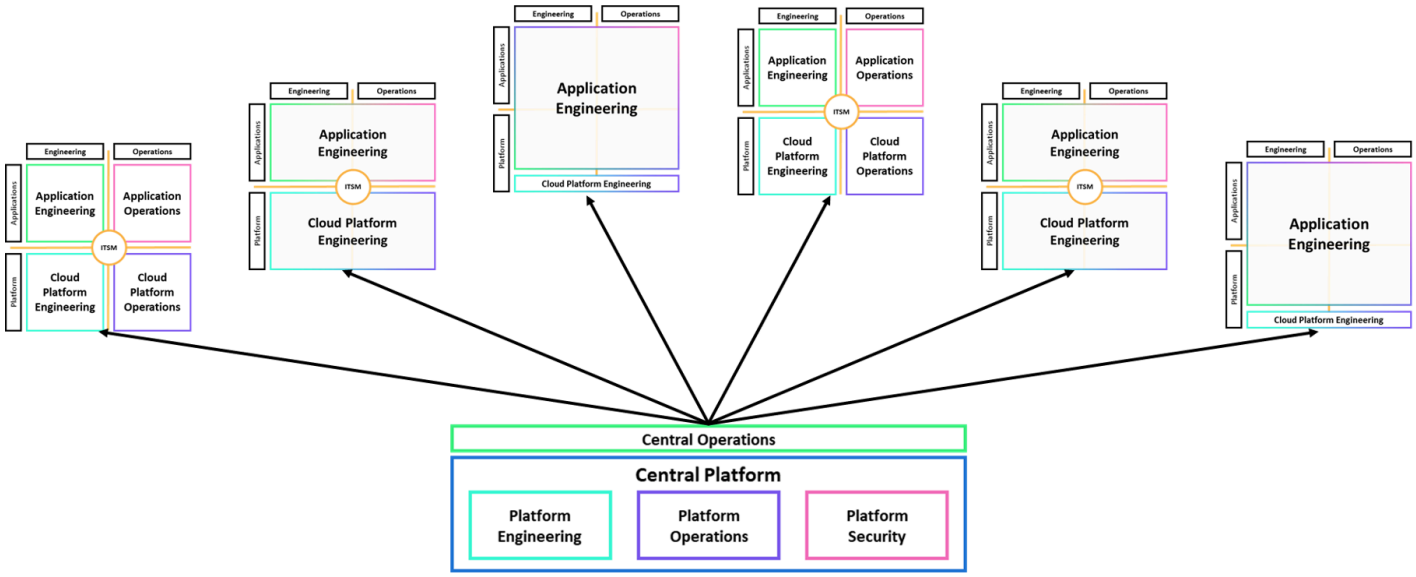
Large, global corporations that operate across different geographies or industry segments often use the decentralized model, which is illustrated in the following diagram. At these corporations, individual business units have their own IT provisions that can overlap with other regions or business units. However, this is often understood and accepted as a way to provide autonomy and specialization within the region.



Using the decentralized approach means that each region or business unit has its own Cloud Operating Model that is tailored to the needs of that region or business unit.

Centralized

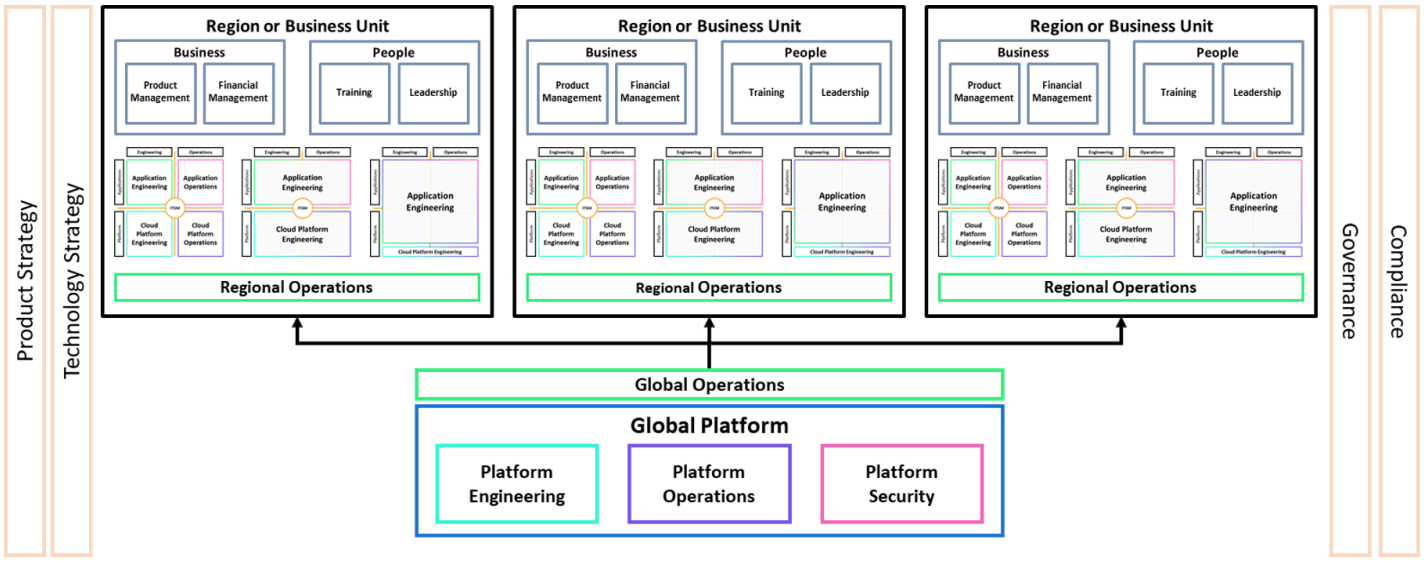
A centralized IT function is the model we see most frequently. When this model is in place, customers seek to maintain the same topology when establishing their Cloud Operating Model. This is illustrated in the following diagram.



In this model, the central team provides a curated platform that can be used by workload teams that have their own Cloud Operating Models. With this approach, workload teams can focus on the value they provide to their end-customers without having to worry about the services, operations, or security of the platform they are using. This model works well for smaller companies. However, in large, global organizations, the number of workload teams can be in the hundreds or thousands. To manage at this scale without losing the benefits of a central platform, organizations frequently transition to the federated model, which is outlined in the next section.

Federated

Many organizations adopt the federated IT model because it provides a central function that's responsible for the cloud platform but allows for a variety of operating models at the workload level. This means that the central team can focus on providing the best possible platform for the organization without the constraint of working to the lowest common denominator. The following diagram illustrates the federated model.



In large organizations, the federated model provides the autonomy required by engineering teams while ensuring that the central team provides the platform and undifferentiated heavy lifting that is common across all workloads. In this model, the central team has to work in the same product-centric way as the engineering teams, but their product is the platform.

Changing the topology to match the journey

The topology you choose depends on the size of your company, but it also adjusts to the stage of your cloud journey. The organization of departments or teams isn't static, but changes with each stage of cloud adoption. This means that you might design, discuss, and augment different topologies as the environment changes. Examples of influencing factors include:

- Moving from proof of concept (POC) to pilot workloads
- Geographic or business unit expansion
- Moving to product-centric teams
- Opportunities to benefit from economies of scale from shared components or patterns
- Realization of Conway's Law, which influences application and service design over architectural requirements
- Cloud-first mandates or other top-down initiatives
- KPI or business goal misses caused by incompatible team goals or organizations

Establish Mechanisms to drive change

Within Amazon, a *Mechanism* is defined as follows: *A complete process that converts Inputs to Outputs and is assembled from Organizational Levers. It uses data and feedback to support the process and ensure outcomes are met.* Because each organization is different, every Cloud Operating Model journey is different, but they all need a Mechanism to drive change.

We recommend that you spend time understanding and developing Mechanisms to match the changes required to implement your Cloud Operating Model. A popular approach is to adopt Agile principles. Agile mechanisms break down organizational and process-based barriers between siloed teams, and create feedback loops to ensure that your organization is spending time innovating on the most impactful activities that will drive the most business value.

Incrementally develop maturity

Maturity in the context of a Cloud Operating Model refers to how close your capabilities are to cloud-first ways of working. For example, how autonomous are your processes, and how much human involvement is needed to manage the business as usual (run the company) compared with innovation (change the company)? If your activities are more heavily weighted toward the former, your (cloud) maturity is low; if it's the latter, your maturity is higher. Being low on the maturity scale is not a negative—it's a reflection of where you are on your journey. The aim is to understand where you are and where you need to get to. When we work with AWS customers, we use a maturity scale within the AWS COM Framework to provide the steps along the journey.

We recommend using a Mechanism to incrementally increase maturity across the AWS COM Framework capabilities. An example of how we have worked with customers in this way is converting maturity reviews and prioritization (inputs) to an increase in maturity (output), and then carrying out experience-based events such as a Game Days (feedback loops) to verify results and adjust as required. By establishing these mechanisms alongside customers, we have found that when this organizational strength is developed, it not only enables the achievement of immediate milestones, but allows for incremental improvement that lasts beyond the initial phases of the journey.

Paying attention to maturing your organization's capabilities and incrementally building the changes required in specific capabilities, at specific times in your roadmap, ties strategy to implementation. It also helps you take advantage of the economies of scale that come with building on your previous achievements.

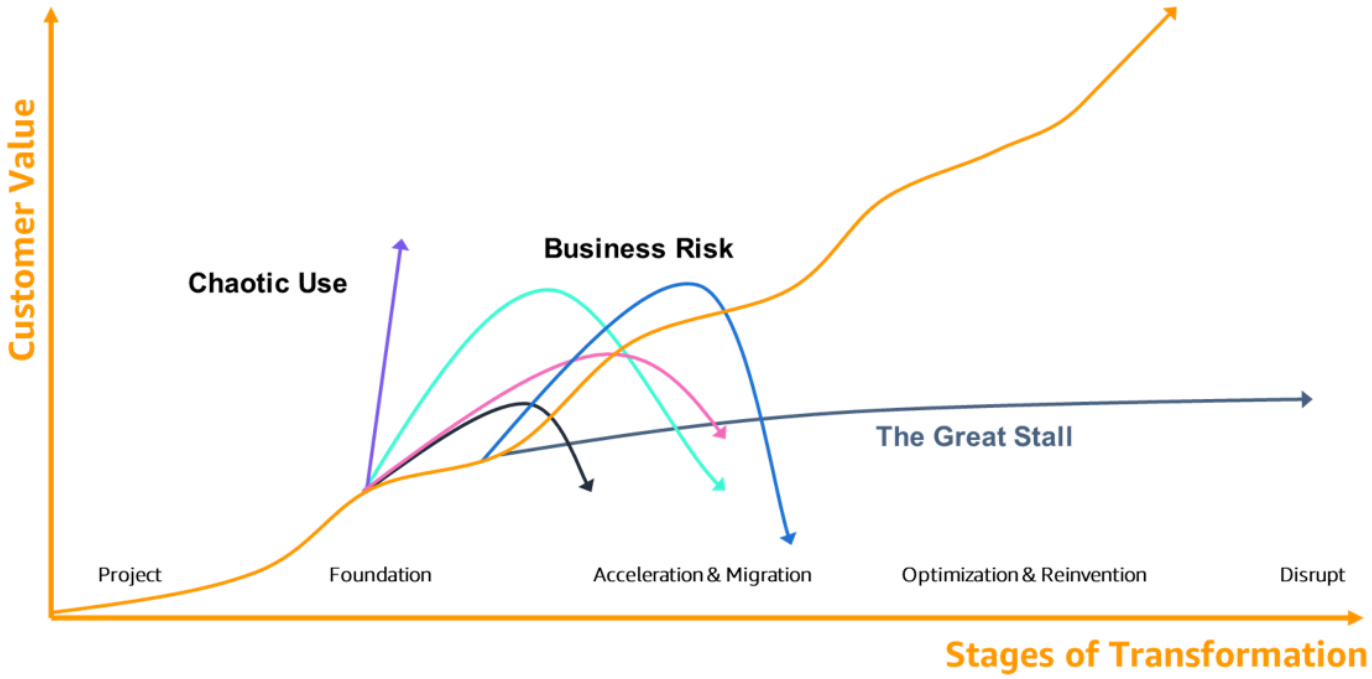
Measure progress

Earlier sections highlighted how cloud leaders can create a compelling vision for their Cloud Operating Model. We provided guidance on how to connect strategy to implementation to support building your Cloud Operating Model. We also explained the need for a framework, such as the AWS COM Framework, to understand and develop maturity levels, and build a roadmap of capabilities that meet the needs of your organization. There is one more piece that is required: ensuring that KPIs are established to measure progress and indicate where a change in direction is required to maintain momentum.

In the internal AWS transformation community, one of the most frequently asked questions is: *"How can our customers measure whether they are actually transforming their business?"*

To understand why this question is important and what can be done about it, see Eric Tachibana's 2015 re:Invent presentation 9 Best Practices to Avoid A Stalled Cloud Transformation Program. In this talk, Eric demonstrates how customers can slow down or even halt their cloud adoption journey (*The Great Stall*) and provides best practices gathered from AWS customers who have successfully accelerated through those delays.

The following graphic highlights what can happen at The Great Stall, and Eric discusses ways to get through that phase. We can take that discussion further to say that progression beyond The Great Stall and managing the journey require that you establish measures and have the ability to correct your course.



The adoption and consumption of cloud services enable this transformation journey, so the absence of a functional Cloud Operating Model and the lack of visibility into the journey can cause adoption to enter The Great Stall. Therefore, we recommend that cloud leaders look to establish observability in the form of a balanced scorecard. This scorecard consists of a set of metrics that are aligned to the digital or cloud transformation. It provides a way to understand your current position and foresee any trouble ahead.

Visualizing metrics

Building a balanced scorecard to visualize metrics helps to understand and put the current transformation efforts in context of the business value they intend to provide. One approach used by AWS teams with their customers is to create a Transformation Dashboard. This approach is based on analyst research of customers who have successfully completed their cloud transformation, and internal analysis of (anonymized) AWS service consumption data of over 5,000 customers from around the world, and across multiple industry segments.

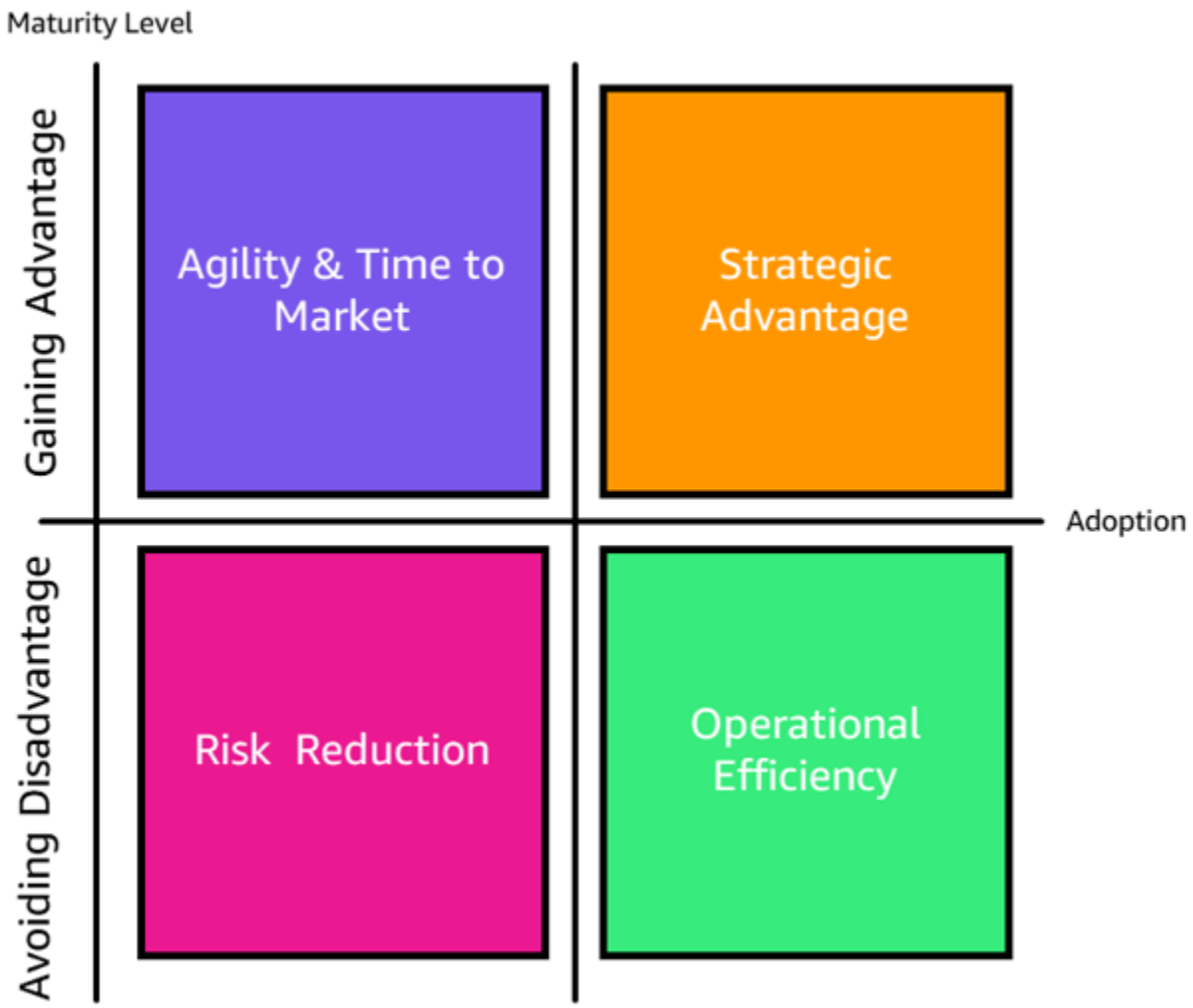
Although our discussion in this guide is based only on AWS Cloud services, you can extend this approach for a hybrid or multi-cloud environment. Using this method, we have identified a balanced scorecard for transformation and several patterns that can be associated with customers who are at different stages of their Cloud Operating Model journey. The objective of this approach is to help customers identify ways in which they can track their overall level of transformative

growth, avoid stall, and ensure that they continue to mature their Cloud Operating Model as an enabler of overall business transformation.

Our Transformation Dashboard balanced scorecard has four segments:

- Agility and time to market
- Strategic advantage (and service Innovation)
- Risk reduction
- Operational efficiency

In this scorecard, two segments highlight the values associated with time to market, agility, innovation, and gaining an advantage over competitors (in a commercial environment). The other two segments focus on measuring how the organization is becoming more efficient, effective, and resilient, and avoiding being at a disadvantage when compared with competitors. The scorecard is shown in the following diagram.



By plotting data points on this matrix you can represent the focus of your organization. This helps you understand whether your Cloud Operating Model is being developed to *avoid disadvantage* or to *gain advantage*. If it's the former, we recommend that you correct your course to ensure that you're developing capabilities to focus on the latter, because gaining advantage is where you can realize the greatest value.

Generally speaking, large-scale migration programs for rehosting workloads (*lift and shift*) focus on avoiding disadvantages. After migration has occurred, modernization activities such as adopting Platform as a Service (PaaS) or serverless technologies support gaining advantages. For example metrics, see the following two AWS-commissioned studies that review these approaches and provide KPIs based on market research:

- **Migration:** The Business Value of Migration to Amazon Web Services (The Hackett Group, February 2022). In this research, The Hackett Group measured the value of migrating to AWS across four categories: resiliency, agility, cost savings, and staff productivity.
- **Modernization:** Business Value of Cloud Modernization (Known, January 2022) captured the use of 22 unique KPIs to understand the value of modernization through cloud services. In this study, they surveyed over 500 enterprises that had already migrated workloads to the cloud to understand the value associated with four technical modernization strategies: containers, serverless, managed analytics, and managed data.

Throughout your Cloud Operating Model journey, it's important to choose measures that can cover both the Migration and Modernization aspects so that progress is tracked, data can be compared throughout the journey, and the results of course correction can be seen.

Conclusion

A Cloud Operating Model is a collection of capabilities that are required to build, mature, and optimize one or more cloud environments. Building capability in a considered and managed way is key to ensuring that your IT organization is aligned with your overall business objectives and is providing value to your organization.

In this strategy document, we provided guidance on how to build a Cloud Operating Model and provided recommendations for each stage of development. We've summarized these recommendations in the following list to help you take the actions necessary to develop and implement your own Cloud Operating Model.

1. Use a customer-centric approach to define or create a Vision Document.
2. Develop a roadmap that connects to the vision and outlines the steps required to reach the intended destination.
3. Review and document your organization's topology to understand the teams involved and what will need to change.
4. Develop mechanisms to drive the changes identified in the roadmap and topology exercises.
5. Use the mechanisms and incrementally increase maturity across the capabilities that you have identified as needing to change.
6. Establish metrics to measure and track progress, and course correct if necessary.

Contributors

Contributors to this document include:

- David Stanley, Principal Operations Transformation Consultant, AWS Professional Services
- Russell Easter, Principal Advisory Consultant, AWS Professional Services
- Brian Quinn, Senior Practice Manager, Operations Transformation, AWS Professional Services

Further reading

For additional information, see the following resources.

AWS resources:

- 9 Best Practices to Avoid A Stalled Cloud Transformation Program (by Eric Tachibana, AWS re:Invent 2015 presentation)
- AWS Cloud Adoption Framework (AWS CAF) 3.0
- AWS Cloud Adoption Framework: People Perspective — *Transformational leadership* section
- AWS Well-Architected Framework: Operational Excellence Pillar — *Operating model 2 by 2 representations* section
- Tenets: supercharging decision-making (by Phil Le-Brun on the AWS Cloud Enterprise Strategy Blog, June 1, 2023)
- Working backwards: Amazon's approach to innovation (by Richard Halkett and Rayford Davis, AWS re:Invent 2020 presentation)

Additional resources:

- 25 Amazing Cloud Adoption Statistics [2023]: Cloud Migration, Computing, and More (by Jack Flynn, Zippia.com, June 22, 2023)
- A Conversation with Werner Vogels: Learning from the Amazon technology platform (*ACM Queue*, Vol. 4, Issue 4, June 30, 2006)
- Business Value of Cloud Modernization (Known, January 2022)
- Conway's Law (by Martin Fowler, martinfowler.com, October 20, 2022)
- Gartner Glossary: Operating Model (Gartner Research)
- Predicts 2023: Collaborate, Automate and Orchestrate to Optimize Costs and Value During the Economic Crisis (Gartner Research, November 1, 2022)
- The Business Value of Migration to Amazon Web Services (by Richard Pastore, Michael Fuller, and Justin Gillespie, The Hackett Group, February 2022)
- What Is a Balanced Scorecard (BSC), How Is It Used in Business? (by Evan Tarver, *Investopedia*, March 10, 2023)

Document history

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

Change	Description	Date
Initial publication	—	August 11, 2023

AWS Prescriptive Guidance glossary

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

Numbers

7 Rs

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

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

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

active-passive migration

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

aggregate function

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

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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

anti-pattern

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

application control

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

application portfolio

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

artificial intelligence (AI)

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

artificial intelligence operations (AIOps)

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

asymmetric encryption

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

atomicity, consistency, isolation, durability (ACID)

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

attribute-based access control (ABAC)

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

authoritative data source

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

Availability Zone

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

AWS Cloud Adoption Framework (AWS CAF)

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

AWS Workload Qualification Framework (AWS WQF)

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

B

bad bot

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

BCP

See business continuity planning.

behavior graph

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

big-endian system

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

binary classification

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

bloom filter

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

blue/green deployment

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

bot

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

botnet

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

branch

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

break-glass access

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

brownfield strategy

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

buffer cache

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

business capability

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

business continuity planning (BCP)

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

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

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

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

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

chaos engineering

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

CI/CD

See continuous integration and continuous delivery.

classification

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

client-side encryption

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

Cloud Center of Excellence (CCoE)

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

cloud computing

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

cloud operating model

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

cloud stages of adoption

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

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

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

CMDB

See configuration management database.

code repository

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

cold cache

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

cold data

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

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

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

configuration management database (CMDB)

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

conformance pack

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

continuous integration and continuous delivery (CI/CD)

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

CV

See computer vision.

D

data at rest

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

data classification

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

data drift

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

data in transit

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

data mesh

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

data minimization

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

data perimeter

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

data preprocessing

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

data provenance

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

data subject

An individual whose data is being collected and processed.

data warehouse

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

database definition language (DDL)

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

database manipulation language (DML)

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

DDL

See database definition language.

deep ensemble

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

deep learning

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

defense-in-depth

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

delegated administrator

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

deployment

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

development environment

See environment.

detective control

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

development value stream mapping (DVSM)

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

digital twin

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

dimension table

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

disaster

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

disaster recovery (DR)

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

DML

See database manipulation language.

domain-driven design

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

DR

See disaster recovery.

drift detection

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

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

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

electronic data interchange (EDI)

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

encryption

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

encryption key

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

endianness

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

endpoint

See service endpoint.

endpoint service

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

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

enterprise resource planning (ERP)

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

envelope encryption

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

environment

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

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

epic

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

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

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

F

fact table

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

fail fast

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

fault isolation boundary

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

feature branch

See branch.

features

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

feature importance

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

feature transformation

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

few-shot prompting

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

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

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

flash-cut migration

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

FM

See foundation model.

foundation model (FM)

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

G

generative AI

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

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

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

Gitflow workflow

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

golden image

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

greenfield strategy

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

guardrail

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

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

H

HA

See high availability.

heterogeneous database migration

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

high availability (HA)

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

historian modernization

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

holdout data

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

homogeneous database migration

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

hot data

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

hotfix

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

hypercare period

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

I

IaC

See infrastructure as code.

identity-based policy

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

idle application

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

IIoT

See industrial Internet of Things.

immutable infrastructure

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

inbound (ingress) VPC

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

incremental migration

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

Industry 4.0

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

infrastructure

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

infrastructure as code (IaC)

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

industrial Internet of Things (IIoT)

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

inspection VPC

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

Internet of Things (IoT)

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

interpretability

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

IoT

See Internet of Things.

IT information library (ITIL)

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

IT service management (ITSM)

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

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

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

landing zone

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

large language model (LLM)

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

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

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

lift and shift

See 7 Rs.

little-endian system

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

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

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

main branch

See branch.

malware

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

managed services

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

manufacturing execution system (MES)

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

MAP

See Migration Acceleration Program.

mechanism

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

member account

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

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

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

microservice

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

microservices architecture

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

Migration Acceleration Program (MAP)

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

migration at scale

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

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners,

migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

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

Migration Readiness Assessment (MRA)

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

migration strategy

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

ML

See machine learning.

modernization

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

modernization readiness assessment

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

monolithic applications (monoliths)

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

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

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

mutable infrastructure

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

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

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

OI

See operations integration.

OLA

See operational-level agreement.

online migration

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

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

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

operational-level agreement (OLA)

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

operational readiness review (ORR)

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

operational technology (OT)

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

operations integration (OI)

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

organization trail

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

organizational change management (OCM)

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

origin access control (OAC)

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

origin access identity (OAI)

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

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

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

P

permissions boundary

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

personally identifiable information (PII)

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

PII

See personally identifiable information.

playbook

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

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

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

polyglot persistence

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

portfolio assessment

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

predicate

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

predicate pushdown

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

preventative control

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

principal

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

privacy by design

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

private hosted zones

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

proactive control

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

product lifecycle management (PLM)

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

production environment

See environment.

programmable logic controller (PLC)

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

prompt chaining

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

pseudonymization

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

publish/subscribe (pub/sub)

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

Q

query plan

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

query plan regression

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

R

RACI matrix

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

RAG

See Retrieval Augmented Generation.

ransomware

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

RASCI matrix

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

RCAC

See row and column access control.

read replica

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

re-architect

See 7 Rs.

recovery point objective (RPO)

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

recovery time objective (RTO)

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

refactor

See 7 Rs.

Region

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

regression

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

rehost

See 7 Rs.

release

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

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

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

resource-based policy

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

responsible, accountable, consulted, informed (RACI) matrix

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

responsive control

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

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotation

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

row and column access control (RCAC)

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

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

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

S

SAML 2.0

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

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

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

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

security by design

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

security control

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

security hardening

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

security information and event management (SIEM) system

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

security response automation

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

server-side encryption

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

service control policy (SCP)

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

service endpoint

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

service-level agreement (SLA)

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

service-level indicator (SLI)

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

service-level objective (SLO)

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

shared responsibility model

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

SIEM

See security information and event management system.

single point of failure (SPOF)

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

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your

organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

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

strangler fig pattern

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

subnet

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

supervisory control and data acquisition (SCADA)

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

symmetric encryption

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

synthetic testing

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

system prompt

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

T

tags

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

target variable

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

task list

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

test environment

See environment.

training

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

transit gateway

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

trunk-based workflow

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

trusted access

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

tuning

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

two-pizza team

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

U

uncertainty

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

undifferentiated tasks

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

upper environments

See environment.

V

vacuuming

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

version control

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

VPC peering

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

vulnerability

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

W

warm cache

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

warm data

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

window function

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

workload

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

workstream

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

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

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

Z

zero-day exploit

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

zero-day vulnerability

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

zero-shot prompting

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

zombie application

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