Proven practices for developing a multicloud strategy

AWS Prescriptive Guidance

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Prescriptive Guidance: Proven practices for developing a multicloud strategy

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Proven practices for developing a multicloud strategy

Tom Godden and Ellie Tamari, Amazon Web Services

September 2025 (document history)

Organizations today face conflicting messages about multicloud adoption. Some advise against it entirely, while others claim that everyone is switching to a multicloud environment. The reality lies between these extremes: Legitimate reasons exist both for and against multicloud strategies, and success depends on balancing potential business value against inherent complexity and risk.

At AWS, our commitment to interoperability is a key reason many customers choose our platform. We believe in giving you the freedom to innovate wherever your workloads are and empowering you to choose the technology that best suits your needs. At AWS, we have been at the forefront of developing solutions that enable you to build and deploy applications in any environment. This customer-centric approach is fundamental to the AWS Cloud, which is trusted by millions of customers worldwide.

We understand that customers need cloud platforms that work seamlessly with both existing tools and future technology choices. You shouldn't have to rebuild everything when you add capabilities from another provider. Your cloud should help you connect, secure, and manage workloads across environments without forcing you to become an expert in every platform. AWS builds connection points directly into its services to help you operate effectively, whether your strategy is to use AWS exclusively or to follow a selective multicloud approach.

We recognize that every organization has unique business requirements that drive their cloud strategy decisions. Whether you're running workloads primarily on AWS, running them across multiple clouds, or using AWS as part of a broader multicloud architecture, we're committed to helping you succeed. AWS delivers the depth and breadth of tooling and capabilities to help you build, migrate, and operate with greater ease and speed, wherever your workloads reside. AWS tools simplify management across providers while maximizing the performance and value of your cloud investments.

This paper focuses on proven tenets for succeeding with a multicloud strategy, including when and where a multicloud approach makes sense and how AWS helps enterprises succeed with their multicloud strategies. It provides prescriptive guidance to help executives make informed strategy and decision-making choices related to multicloud adoption. This paper doesn't offer a technical, in-depth discussion of multicloud implementations. For technical implementation support and

assistance with your specific challenges, we recommend that you work with your AWS solutions architect.

This paper presents nine proven tenets for multicloud success based on our experiences with AWS enterprise customers. Each tenet addresses a critical aspect of multicloud strategy, from aligning business goals to security implementation. By applying these principles, organizations can navigate multicloud complexity with confidence.

Tenet 1. Align multicloud goals with your strategy

Research by Gartner and industry trends show that organizations are increasingly adopting multicloud approaches to address specific business needs. The following scenarios demonstrate when a multicloud infrastructure can be strategically advantageous.

Mergers and acquisitions

Mergers and acquisitions (M&A) create immediate decisions about cloud strategy. Although operating multiple clouds might increase costs and complexity, rapid consolidation can delay integration value and disrupt business operations. Your cloud decisions become central to realizing M&A benefits.

Integration planning should account for the complete technology landscape. Each workload requires evaluation within the context of your integration timeline and business priorities.

Our guidance:

- Develop a business-driven consolidation strategy that balances immediate integration needs with long-term operational efficiency. Maintain multiple clouds initially in circumstances where hasty consolidation could disrupt critical business operations or delay M&A value realization.
- Create clear workload placement criteria that align with your integration timeline. Prioritize revenue-generating applications and core business processes while accounting for technical dependencies and operational requirements.

Desire to leverage long-term differentiated capabilities of another CSP

The fear of missing out drives some companies to want a bit of every cloud. Workload placement decisions affect the entire organization—from engineering teams to finance to security operations.

Organizations therefore need to examine their reasoning for pursuing multiple clouds. Some argue that each workload should live on the cloud services provider (CSP) that best meets its needs. However, individual workload optimization must be balanced against the broader organizational impact. Each additional cloud provider risks increasing operational complexity, creating new

talent requirements, and introducing security considerations that affect the entire technology organization.

Our guidance:

- Follow an 80/20 approach: Select a primary provider for most workloads and consider additional providers only for specific, high-value use cases. This strategy maximizes efficiency and talent retention while reducing complexity.
- Consider the total cost of operating across clouds. Include security tooling, governance products, financial management systems, and operational overhead in your analysis.
- Evaluate each workload's dependencies and interactions. Workloads rarely operate in isolation; they share data, security controls, and operational processes.
- Conduct thorough price-performance analysis across providers. Compare not just the direct costs but also the overhead of managing multiple environments.

Multicloud at the holding company and primary cloud at the operating company or line of business

Private equity firms and holding companies face unique cloud strategy considerations. Their portfolio companies often maintain independent cloud strategies, frequently resulting from past M&A activity. This structure reduces the complexity that's typically associated with multicloud operations, because each business unit operates independently. However, this independence can limit opportunities to take advantage of enterprise-wide volume discounts and purchasing incentives.

The effectiveness of cloud strategy at the holding company level depends on the autonomy of portfolio companies and their individual technology needs. Although consolidation might create purchasing leverage, it might conflict with the independent operation model that's typical of holding companies and private equity portfolios.

Our guidance:

- Understand CSP volume discount structures. Each provider offers mechanisms for adding or removing subsidiaries from enterprise agreements and spinning off business units into separate entities. These represent two-way door decisions.
- Plan cloud purchasing commitments carefully. Engage your CSP's account team early, or contact an AWS Partner with the AWS Cloud Operations competency for assistance.

- Balance independence with efficiency. Consider shared services or purchasing agreements that benefit portfolio companies without constraining their operations.
- Focus on business objectives first. Develop technology strategies that support your operating model rather than pursuing a multicloud strategy for its own sake.
- Evaluate cloud strategies through the lens of portfolio management. Consider how cloud choices affect potential divestitures or future acquisitions.

Tenet 2. Be mindful of multicloud misconceptions

When you're developing your multicloud strategy, avoid the common misconceptions that are discussed in the following sections.

Everyone is adopting multicloud strategies

Advisory firms and media companies paint a complex picture of multicloud adoption. Research shows broad interest in multicloud approaches, but spending patterns often tell a different story. In practice, many enterprises maintain either single cloud environments or clear primary/secondary CSP relationships. This disconnect highlights the importance of looking beyond headlines and focusing instead on the specific needs of your organization.

Our guidance:

- Make cloud decisions based on your specific business requirements instead of following industry trends. Focus on measurable costs and risks for your organization.
- Examine multicloud use cases within your industry context. Cloud strategies that work for consumer technology companies might not translate to financial services, manufacturing, or gaming environments.
- Consider data gravity as a primary factor in workload placement decisions. The location and movement of data often determine the most effective cloud architecture.
- Look beyond adoption statistics to understand spending patterns. High reported multicloud adoption rates often mask actual spending patterns.
- Evaluate technical constraints before committing to a multicloud environment. Some workloads perform best when their components remain within a single cloud environment.

Multicloud reduces the risk of vendor lock-in

Vendor flexibility is a legitimate consideration in cloud strategy development. Organizations value the ability to adapt their technology choices as business needs evolve. This concern reflects prior experiences with traditional IT investments that created binding, long-term commitments. Cloud services offer different dynamics around provider flexibility. AWS provides open source compatible services and data portability options that reduce technical barriers to migration. However, the trade-off between flexibility and operational efficiency remains important. Organizations must

weigh the business value of maintaining provider options against the technical advantages of deeply integrating with specialized services from a primary provider.

Some customers attempt to avoid lock-in by engineering cloud-agnostic solutions that use containers. This approach often restricts them to basic compute and storage services, and bypasses the advantages of advanced cloud capabilities. Our experience shows that this strategy adds considerable complexity due to the increased development time and resources required, compared with using native services.

Our guidance:

- Consider the full cost of cloud-agnostic architectures. The additional engineering overhead might not justify portability benefits.
- Use cloud-native capabilities for maximum value. Basic compute and storage services alone often sacrifice significant advantages in security, scalability, and innovation.
- Plan cloud strategies based on business requirements. When a multicloud implementation adds clear value, such as the ability to serve users on multiple platforms, the additional engineering investment becomes worthwhile.
- Evaluate realistic exit scenarios and costs. Compare the likelihood and expense of changing providers against the benefits of using the complete set of AWS services.
- Build on the open source foundations of AWS. AWS managed services such as Amazon Relational Database Service (Amazon RDS) give you both flexibility and operational excellence, and support the database engines you are using today.
- Leverage the comprehensive migration tools provided by AWS. We help you move workloads in any direction and provide free data egress if you leave AWS to use other providers. For more information, see the AWS blog post Free data transfer out to internet when moving out of AWS.

Multicloud improves availability and resilience

The belief in seamless workload switching between cloud providers during outages drives some organizations toward multicloud strategies. This mindset creates an oversimplified view of cloud infrastructure resilience that ignores fundamental technical realities.

Based on years of experience working with multicloud customers on AWS, we've seen that maintaining full workload portability between providers often creates substantial complexity without delivering all the expected benefits. Data-intensive applications face insurmountable

challenges due to data gravity constraints. In fact, in our view, it is nearly impossible for organizations to successfully implement a truly seamless multicloud failover for data-heavy workloads.

Lydia Leong, Distinguished VP Analyst at Gartner, reinforces this perspective in a social media post: "Multicloud failover is complex and costly to the point of nearly almost always being impractical, and it's not an especially effective way to address cloud resilience risks." The inherent differentiation between providers in networking, storage, databases, machine learning, and security makes true portability nearly impossible. Spreading workloads across providers might increase risk, because a failure in either environment could trigger an outage across all enviornments.

Our guidance:

- Focus on mastering AWS capabilities for individual workloads instead of pursuing complex multicloud architectures.
- Build resilience through AWS Regions and Availability Zones instead of attempting cross-provider failover. For a technical deep dive into how AWS can automatically fail over workloads between physical data centers, see the AWS blog post, Zonal autoshift – Automatically shift your traffic away from Availability Zones when we detect potential issues.
- Migrate workloads strategically to AWS, and focus on one application at a time to maximize success.

Multicloud provides better pricing

Price competitiveness might be the weakest argument of all for multicloud environments. Organizations' experiences with complicated, expensive software or data center contracts that lock them into multi-year agreements have made them wary when procuring IT services. Traditional procurement approaches have not adapted to pay-as-you-go purchasing, volume discounts, or the reality of price competition in the cloud. (As of January 2025, AWS has reduced prices 151 times since its inception.)

The biggest single driver of cost reduction is a well-managed and optimized cloud environment. A company sees better cost optimization by working primarily with a provider whose services offer price-performance advantages (such as compute instances that are based on custom-designed chips such as AWS Graviton) and has superior cloud financial management solutions. According to a 2022 Hackett Group study of more than 1,000 organizations, infrastructure spending as a

percentage of total IT spending was 20% lower for AWS customers compared with multicloud organizations.

Our experience has shown that companies do not anticipate the added cost and complexity of operating in multiple clouds, nor do they appropriately weigh this cost against the perceived gain in a head-to-head sourcing engagement.

Our guidance:

- Build your cost optimization strategy on the AWS Well-Architected Framework Cost Optimization Pillar. There are five design principles:
  - **Implement cloud financial management:** To achieve financial success and accelerate business value realization in the cloud, you must invest in cloud financial management. Your organization must dedicate the necessary time and resources for building capability in this new domain of technology and usage management. As with your security or operations capability, you need to grow capabilities through knowledge building, programs, resources, and processes to help become a cost-efficient organization.
  - **Adopt a consumption model:** Pay only for the computing resources you consume, and increase or decrease usage depending on business requirements. For example, development and test environments are typically used only for eight hours a day during the work week. You can stop these resources when they're not in use for a potential cost savings of 75% (40 hours versus 168 hours).
  - **Measure overall efficiency:** Measure the business output of your workload and the costs that are associated with delivery. Use this data to understand the gains you make from increasing output, increasing functionality, and reducing cost.
  - **Stop spending money on undifferentiated heavy lifting:** CSPs do the heavy lifting of data center operations such as racking, stacking, and powering servers. They also remove the operational burden of managing operating systems and applications by using managed services. This allows you to focus on your customers and business projects instead of IT infrastructure.
  - **Analyze and attribute expenditure:** The cloud makes it easier to accurately identify the cost and usage of workloads, which then allows transparent attribution of IT costs to revenue streams and individual workload owners. This helps measure return on investment (ROI) and gives workload owners an opportunity to optimize their resources and reduce costs.
- Given the financial overhead of operating across different providers, we guide customers to invest heavily in automation and cost optimization tooling. Each CSP offers extensive native tools in this area, such as the AWS Cost Optimization Hub. Most native tools provide excellent

capabilities for customers in their cloud environment. However, to understand spend across multiple CSPs, you can choose from a rich set of ISV and software as a service (SaaS) products that extend these capabilities to provide a single experience for cost optimization.

- Diluting purchasing power through a *spend equity* strategy doesn't generate business value. It can undermine potential volume discounts and potentially undermines technical design. The most efficient way to consume cloud services is by using a primary provider for the bulk of your operations and using other CSPs only where it adds business value.

Tenet 3. Have a clear strategy and governance to support it

Deciding to pursue a multicloud strategy is insufficient; you must establish a strategy for delivering on your objectives, including clear governance for which workloads will go where and why. Evaluation criteria should be used to optimize workloads and their dependencies. If the evaluation is left up to individuals, an uncoordinated sprawl across CSPs will likely erode the value of the multicloud strategy. We recommend that you evaluate CSP workload performance regularly and use your assessment as a key input to CSP selection, criteria, and future usage.

An effective governance strategy requires visibility into the total number of services, applications, and components used across the enterprise. Integral to this is a robust tagging strategy that spans CSPs and establishes clear ownership, usage, and environment (such as development, QA, staging, and production) for all deployed resources. Everything should be tagged to an owner; if it is not tagged or an owner cannot be identified, it should be removed. We work closely with a major financial services organization that automatically finds and removes any untagged resources, and considers this a best practice, regardless of the inconvenience it presents to development teams. This tagging approach codifies governance rules and automates enforcement instead of creating blocks to progress (that is, it implements guardrails, not gates). Cost, operations, and security must be tracked, monitored, and acted upon in the same way, with the same depth of data and transparency across CSPs.

When you implement a multicloud strategy, establishing a clear and consistent account structure across cloud providers is crucial for maintaining operational control and security. We recommend adopting a hub-and-spoke model, where you create separate AWS accounts for different business units. These are anchored by two critical central accounts: a security/audit account for consolidated compliance and security monitoring, and a central networking account for managing interconnectivity. (This approach is codified in the design of AWS Control Tower. However, the principles of least privilege and separation of duties are equally applicable to other clouds. The AWS Well-Architected Framework discusses these concepts at length, and is highly recommended for technical audiences.) This foundational approach should be mirrored across cloud providers to maintain consistency in governance and operations. Workload accounts should be organized by environment (development, staging, production) or function, with clear processes established for account creation and deletion.

Our guidance:

- Implement a comprehensive tagging strategy to maintain clear ownership and usage patterns across all cloud resources. Track environments, cost centers, applications, and business units through consistent tagging policies. Remove resources that lack proper tags to enforce governance standards and maintain environment clarity.
- Establish a unified compliance framework that maps regulatory requirements across your multicloud environment. Maintain clear documentation of how each cloud provider's controls and certifications support your compliance obligations.
- Automate governance enforcement through automation instead of using manual approval processes. Code your governance rules into automated systems that prevent policy violations before they occur. This removes human error while maintaining development velocity.
- Structure accounts in a hub-and-spoke model with centralized security and networking control. Create dedicated accounts for security auditing and network management to centralize critical functions. This foundation enables consistent security policies and network connectivity across the organization.
- To maintain operational boundaries, create separate accounts, subscriptions, or projects (depending on your CSP's nomenclature) for different environments and functions. Divide workloads by development, staging, and production environments. This separation prevents security incidents from spreading and maintains clear operational domains.
- Monitor costs, operations, and security through consistent metrics across the environment. Implement unified monitoring for resource utilization, security events, and spending patterns. Use this data to optimize workload placement and resource allocation decisions.
- Prevent unauthorized cloud usage through organizational policies and automated controls. Define clear processes for account creation and resource provisioning. Implement service control policies (SCPs) to enforce compliance with organizational standards across all accounts.
- Establish detective and preventive controls to prevent shadow IT from emerging through unauthorized provider accounts. Monitor for unauthorized cloud usage through expense reports and network traffic. Block unauthorized provider access while maintaining approved paths for innovation.

Tenet 4. Do not spread contiguous workloads across clouds

Spreading contiguous workloads across multiple cloud providers creates unnecessary complexity, risk, and cost. When workloads that process and analyze data together span multiple providers, organizations face challenges in data movement, synchronization, and consistency. Teams must navigate different APIs, management interfaces, security models, and operational processes for each provider, which increases the likelihood of errors and adds operational overhead. This complexity increases the chances of errors and operational overhead, and can hinder agility and scalability.

However, in some practical scenarios, organizations might need to distribute contiguous workloads across clouds because of specific business or technical requirements. In these cases, we recommend that you establish clear criteria and guiding principles to evaluate the trade-offs, and ensure that the approach aligns with your organization's overall multicloud strategy.

When organizations choose to distribute workloads across multiple clouds, adopting an architecture that's centered on messaging and loose coupling can alleviate many of the associated challenges. This is the best way to separate concerns between clouds and to reduce the scope of impact if a provider is impaired. Operations that are the most time-bound, such as financial transactions, should ideally be kept within a single environment. An outage in one environment should never be allowed to endanger workloads in another environment.

Our guidance:

- Design cloud workloads for operational independence to minimize real-time dependencies between providers. When workload distribution is necessary, implement efficient bulk data transfer mechanisms instead of maintaining constant cross-cloud connections.
- Evaluate each proposed distributed workload against clear business criteria. Consider both the strategic benefits and the operational complexity introduced by the distribution.

Tenet 5. Have a longer-term integration strategy

Be careful when you move large volumes of data between applications in different clouds, especially if your compute resources and applications are deployed in one CSP, and your data storage resources are deployed in another. Such a situation can add complexity and latency that might offset perceived benefits. We speak with many customers who have a data lake on one cloud but want to perform machine learning (ML) or analytics with tools from another CSP. Deciding where to place workloads in a multicloud environment is one of the most crucial—and often most challenging—decisions organizations face. We recommend that you evaluate each workload placement decision through three critical dimensions: technical requirements, business needs, and provider strengths.

Start technical evaluations by mapping each workload's essential characteristics: computing power, data operations, response time needs, and growth requirements. Applications naturally perform best when they're located near their data. Moving applications away from their data sources creates unnecessary technical hurdles and slows performance.

Business decisions must account for provider pricing, data residency requirements, and vendor contracts. Each workload placement affects the entire organization's operations, security, and productivity. Looking at workloads in isolation leads to suboptimal decisions.

Our guidance:

- Implement bulk data transfer between clouds instead of real-time access. Schedule periodic data refresh by using efficient bulk operations instead of using constant API calls between clouds. This approach reduces costs, improves reliability, and maintains consistent performance. For example, export summarized daily sales data instead of querying individual transactions across clouds.
- Consider data gravity when designing workload placement. Keep applications close to their primary data sources to maintain performance and to reduce costs. ML models, analytics engines, and transaction processing systems all benefit from direct access to their data. Moving these workloads away from their data creates unnecessary network latency and complexity.
- Evaluate workload decisions within the context of your complete cloud strategy instead of reviewing them in isolation. Consider how each placement choice affects operational processes, security controls, and team capabilities across your organization. A decision that seems optimal for a single workload might complicate monitoring or increase security risks when viewed holistically.

- Define clear data ownership and governance policies that specify where different types of data can reside. Create a data classification framework that drives consistent decisions about data placement across cloud providers.

Tenet 6. Use containers strategically

Containers can play a valuable role in supporting a multicloud strategy, but it's important to recognize their limitations as well. Using containers is generally a good idea for any modern, cloud-native application, because they provide benefits to portability and consistency across different environments. Containers are platform-agnostic, which means that they can run on any cloud platform or infrastructure that supports containerization technology, such as Kubernetes. Organizations that use containers can develop and package their applications once and then deploy them consistently across multiple cloud providers or on-premises environments, without the need for significant modifications. By encapsulating application code, dependencies, and runtime environment within a container, you can achieve a high degree of portability, which enables you to move workloads seamlessly between cloud providers or between the cloud and on-premises data centers.

However, containers might not solve every use case or eliminate all the challenges an organization might face in adopting a multicloud strategy. Containers work best with modern, microservices-based architectures, but they might not be as well-suited for large, monolithic applications. Additionally, although containers can address certain aspects of portability, such as the application runtime, they do not automatically resolve issues around data management, security policies, and other cross-cloud dependencies. Organizations still need to carefully plan and architect their multicloud solutions to ensure consistent data management, unified security controls, and seamless integration between cloud-hosted and on-premises components.

Our guidance:

- Use each cloud provider's native container management capabilities to maximize business value and accelerate delivery. This approach ensures optimal performance while avoiding the complexity of creating cloud-agnostic solutions that rarely deliver meaningful returns.
- Develop container strategies that address the complete operational picture, including data management, security, and cross-cloud dependencies. Focus on business outcomes when you make container architecture decisions.

Tenet 7. Have a single CCoE, but specialize within it

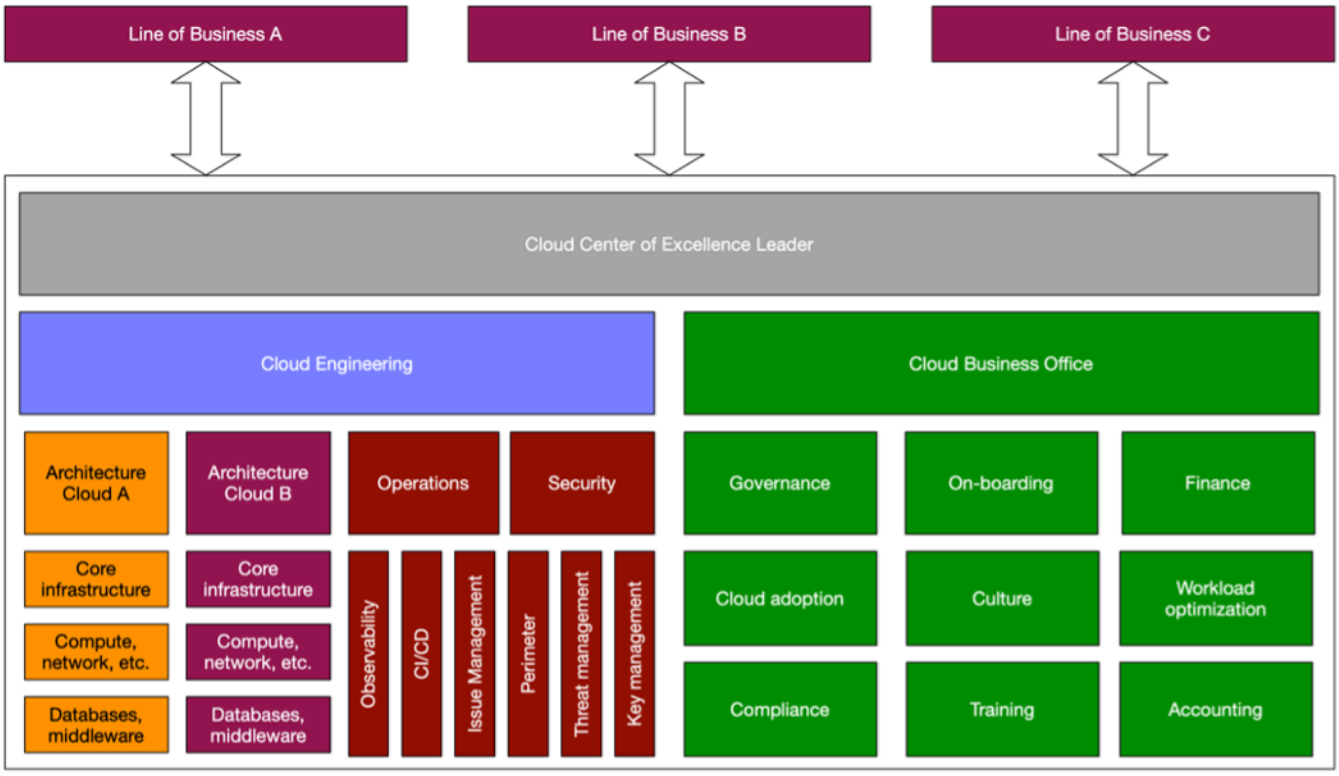
As we advise many AWS customers, you should build a Cloud Center of Excellence (CCoE) within your organization to provide leadership, standardization, and acceleration of your cloud journey. When it comes to multicloud environments, we find that the most successful companies take a balanced approach with their CCoE.

Instead of establishing separate CCoEs for each CSP, we recommend that you have a single, unified CCoE that oversees the organization's multicloud strategy. This helps ensure a coordinated, consistent approach instead of siloed efforts that can lead to divergence, reengineering, and waste. Make sure that the teams within your single CCoE have the necessary specialized skills, tools, and mechanisms for each CSP that your organization uses. This specialized knowledge allows the CCoE to govern, support, and accelerate the use of the different cloud platforms effectively.

For example, the CCoE should have AWS-specific experts who understand the AWS Cloud, services, and best practices in depth, as well as experts for other CSPs who can guide the organization's use of those cloud technologies. This specialized expertise within the single CCoE can help your organization benefit from the coordination and standardization of a centralized approach while ensuring that each cloud platform is being used optimally.

The single CCoE should serve as the central governing body that establishes standards, policies, and best practices for the organization's multicloud strategy. The actual implementation of cloud workloads and projects can be distributed to specialized teams or business units while the CCOE provides oversight, support, and coordination. This balanced approach helps ensure a cohesive multicloud strategy while providing the necessary degree of flexibility and autonomy within the organization.

The following diagram illustrates how a CCoE can provide a centralized approach and governance across multiple lines of business (LOBs), cloud engineering teams, and Cloud Business Office (CBO) teams.



Our guidance:

- Structure your CCoE to maintain strategic oversight while embedding specialized expertise for each cloud provider. Focus on recruiting deep expertise in individual cloud platforms instead of seeking rare multicloud specialists, and foster internal knowledge sharing to build organizational capabilities.
- Empower your CCoE to establish enterprise-wide standards for cross-cutting concerns such as security and observability, while giving individual teams the autonomy to execute within these guidelines by using cloud-native tools and services.
- Develop a comprehensive talent strategy that balances deep expertise in primary cloud platforms with broader architectural knowledge. Focus on building teams that combine strong, cloud-specific skills with enterprise architecture experience.

Tenet 8. Make sure that security is always a top priority

A multicloud approach makes it harder to ensure security by increasing the risk of unauthorized access, because your security posture must account for more attack surfaces. A multicloud strategy often forces companies to deal with multiple security models across CSPs in areas such as identity management, network security, asset management, and audit logging. This complexity risks making transparency harder, increases the burden on security teams, and elevates risk.

Security automation is essential in multicloud environments. Identity management must work seamlessly across environments; it must connect existing identity providers while maintaining consistent access policies. Security requires integrated protection across data, network, and endpoint layers. Data classification, encryption, and lifecycle management form the foundation. Network security builds on standardized designs and connection patterns. Endpoint protection completes the framework through consistent patch management and host-based controls.

These foundational elements are critical to successful and safe adoption of multiple cloud providers and must be considered early in any multicloud strategy planning.

Our guidance:

- Implement an integrated security framework across your multicloud environment that focuses on three core elements: data protection through standardized classification and encryption, network security through consistent design patterns, and endpoint protection through systematic controls and patch management.
- Establish a unified security operations model that takes advantage of each cloud provider's native security capabilities while maintaining centralized visibility and control through standardized tools and processes.
- Centralize security data collection and analysis by using Amazon Security Lake. This platform aggregates security information from AWS, other cloud providers, SaaS applications, and on-premises systems into a single view. It supports the Open Cybersecurity Schema Framework (OCSF) and enables standardized analysis across your hybrid and multicloud environment. This centralized approach improves threat detection and response while simplifying security operations.
- Deploy each provider's native security tools to enhance your protection capabilities. These purpose-built services address provider-specific features while feeding data back to your centralized security platform. A combination of native tools and centralized visibility helps provide comprehensive security coverage across your entire infrastructure.

- Implement a unified observability strategy that provides comprehensive visibility across your entire cloud landscape, including operational and security data, from the ground up. Standardize on industry-leading monitoring approaches that enable consistent tracking of business services regardless of where they operate.
- Establish enterprise-wide standards for operational data collection and visualization that enable rapid issue identification and resolution across your multicloud environment. Focus on creating a single source of truth for operational insights that serves both technical and business stakeholders.

Tenet 9. Embrace an 80/20 approach over equal distribution

How you distribute workloads across providers fundamentally determines your multicloud success. Many organizations mistakenly pursue equality in their cloud distribution, and attempt to spread workloads evenly across providers. This approach increases complexity without delivering proportional benefits. Equal distribution fragments your technical capabilities, dilutes your purchasing power, and creates unnecessary operational overhead. Teams struggle to develop deep expertise when they're forced to maintain competency across multiple platforms simultaneously.

The 80/20 approach delivers demonstrably better results than equal distribution across clouds. Concentrating 80% of your investment with one primary provider while selectively using others for specific capabilities creates a balanced strategy that reduces both cost and complexity. This concentrated approach accelerates innovation because your teams can develop deep expertise with your primary platform's advanced services. Your technical staff can become specialists in one architecture instead of maintaining surface-level knowledge across multiple environments. When engineers master one platform, they build more efficiently, troubleshoot faster, and implement more sophisticated solutions.

Companies that follow the 80/20 approach typically report better talent retention because their teams develop valuable, marketable expertise instead of being stretched thin across multiple technologies. This concentrated strategy also helps simplify security management by limiting the complexity of different security models across providers. The primary cloud receives most of your investment in security tools, monitoring solutions, and operational processes. This creates a stronger security foundation than what's possible with equally divided resources.

Our guidance:

- Select a primary cloud provider that aligns with most of your business and technical requirements. This provider should support at least 80% of your workloads and become the foundation of your cloud strategy. Focus your training investments, architectural standards, and operational processes on maximizing value from this primary platform.
- Develop clear criteria for workloads that warrant placement on secondary clouds. These criteria should focus on specific business value that cannot be achieved on your primary provider. Resist placing workloads on secondary clouds simply to maintain spending equity or artificial balance between providers.

- Structure your enterprise agreements to reflect your 80/20 approach. Negotiate volume discounts with your primary provider based on concentrated spending, and maintain flexibility with secondary providers for specific use cases. This approach maximizes your purchasing leverage and typically results in better overall pricing than dividing your spend equally.
- Align your talent strategy with your 80/20 approach. Invest in developing deep expertise with your primary provider's services while maintaining sufficient knowledge of secondary platforms to support specific workloads. This focused talent strategy improves productivity, accelerates delivery, and reduces the risk of critical skill gaps.
- Measure the business outcomes of your multicloud strategy regularly. Track metrics that demonstrate the value gained from each provider and adjust your distribution if necessary. The goal isn't to avoid multicloud entirely but to implement it strategically where specific workloads truly benefit from capabilities that are unique to other providers.

Conclusion

This paper has outlined nine key tenets for developing an effective multicloud strategy. Organizations achieve greatest success through a primary cloud approach with strategic use of additional providers where specific business needs demand it. The 80/20 approach we've described balances focus with flexibility and enables organizations to develop deeper expertise, maintain stronger provider relationships, and build more valuable talent while still addressing legitimate multicloud requirements.

Successful multicloud implementation requires a clear assessment of business needs instead of following industry trends. Companies must establish robust governance, maintain security as a top priority, avoid spreading connected workloads across providers, keep applications with their transactional data, recognize container limitations, and maintain a unified but specialized Cloud Center of Excellence.

The AWS approach to the cloud is fundamentally built on customer choice and interoperability. We've designed our tools and services to work seamlessly across environments because we understand that your business needs often extend beyond a single provider. From hybrid connectivity solutions to container orchestration that spans environments, AWS delivers capabilities that help you operate effectively across your technology landscape.

Instead of forcing you to become experts in multiple platforms, AWS simplifies multicloud management through intuitive tools and consistent interfaces. We focus on removing complexity so you can focus on innovation. These capabilities help you implement your multicloud strategy on your own terms—whether that means using AWS exclusively or using specific AWS services alongside other environments.

The cloud should empower your business strategy, not constrain it. By applying the principles outlined in this paper and leveraging AWS interoperability capabilities, you can build a cloud approach that maximizes value, minimizes unnecessary complexity, and positions your organization for long-term success in today's dynamic business environment.

To learn more about AWS solutions that can help simplify management across hybrid and multicloud environments, see AWS solutions for multicloud.

Resources

References

- Using a Cloud Center of Excellence (CCOE) to Transform the Entire Enterprise (AWS blog post)
- AWS Well-Architected Framework
- Identifying opportunities with Cost Optimization Hub (AWS Cost Management documentation)
- The Business Value of Migration to Amazon Web Services (The Hackett Group, February 2022)
- Free data transfer out to internet when moving out of AWS (AWS blog post)

Tools

- Zonal autoshift – Automatically shift your traffic away from Availability Zones when we detect potential issues (AWS blog post)
- AWS solutions for multicloud

AWS Partners

- AWS Cloud Operations competency

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	—	September 3, 2025

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.