Transforming application development and maintenance operating models on AWS with generative AI

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Transforming application development and maintenance operating models on AWS with generative AI

Table of Contents

Transforming application development and maintenance operating models on AWS with generative AI

Dhana Vadivelan, Amazon Web Services (AWS)

April 2025 (document history)

Organizations face unprecedented challenges in application development and maintenance (ADM) practices today. Generative AI is fundamentally changing how applications are built, designed, tested, documented, and deployed—transforming the entire software development lifecycle (SDLC).

ADM encompasses the complete application lifecycle from business requirements analysis through development and maintenance, representing a comprehensive practice of managing applications. The SDLC defines the structured methodology and phases for building software within this broader ADM framework.

To assist your organization's transformation journey to AI-powered ADM practices, this strategy document offers:

- Comprehensive analysis of AI's impact on ADM, including operating model and role-specific changes
- Strategies for enhancing organizational capabilities and addressing key challenges
- A framework for building and implementing an AI-powered ADM operating model
- A phased implementation approach to an AI-powered ADM operating model, from quick wins to full AI integration

Intended audience

This strategic document is recommended for the following audiences:

- IT leaders, such as chief technology officers (CTOs), technical directors, technical leads, architects, and program managers
- Business leaders, such as chief information officers (CIOs), chief data officers (CDOs), vice presidents (VPs) of product engineering, and VPs of business operations

Objectives

This strategy document can help your organization achieve the following objectives:

- Examine your current ADM operating model for transition to the AI era.
- Address the unique challenges of generative AI integration.
- Implement a phased transformation strategy to integrate generative AI into your organization's ADM.

Benefits of integrating generative AI into ADM

For *IT leaders*, integrating generative AI into your organization's ADM can provide the following benefits to enhance your organization's capabilities:

- Accelerate innovation cycles through rapid prototyping and responsive software development.
- Automate routine tasks in architecture definition, code generation, and testing.
- Enhance software quality and reliability, minimizing defects and mitigating risks.
- Improve operational scalability by handling increased complexity and development volume.

For *business leaders*, integration of generative AI can deliver benefits that extend beyond technical improvements to create business value:

- Deliver customer-centric applications faster, adapting quickly to market demands.
- Gain competitive advantages by increasing operational efficiency with AI technologies.
- Position your organization as a leader in AI-driven development, attracting top talent.
- Achieve cost efficiency through improved productivity and optimized resource allocation.

Early adopters across industries are reaping the benefits from using AWS generative AI services in ADM:

- **Development speed** – BlackBerry improved SDLC agility and quality with Amazon Q Developer.
- **Code generation** – BT Group automated 12 percent of repetitive tasks using Amazon CodeWhisperer, which is becoming part of Amazon Q Developer.

- **Modernization** – Novacomp used Amazon Q Developer to reduce a Java application modernization time from 3 weeks to 50 minutes.
- **Documentation** – ADP used Amazon Q Developer to cut legacy system documentation time from weeks to less than a day.
- **Productivity** – National Australia Bank used Amazon Q Developer to achieve 50 percent acceptance of AI-generated code suggestions.
- **Application modernization** – Deloitte uses Amazon Q Developer to accelerate modernization phases, reducing project complexity and completion times. TCS used Amazon Q Developer to accelerate mainframe modernization, quickly analyzing and documenting legacy COBOL code.
- **Application migration** – Cognizant uses Amazon Q Developer to automate complex migration processes, increasing speed and simplicity in transformation projects. Also using Amazon Q Developer, HCLTech employs AI agents for accelerating VMware, .NET, and mainframe workloads.
- **Application efficiency** – IBM Consulting's AI-based SDLC solution on AWS Marketplace makes use of Amazon Bedrock to enhance efficiency and quality throughout the application lifecycle.

Understanding operating models in ADM

Before exploring the transformative impact of AI on ADM, it's important to understand the fundamentals of an operating model in the context of ADM. This section provides an overview of a typical IT operating model. Then, review the key components and layers of an ADM operating model, which sets the stage for AI-driven changes.

In this section:

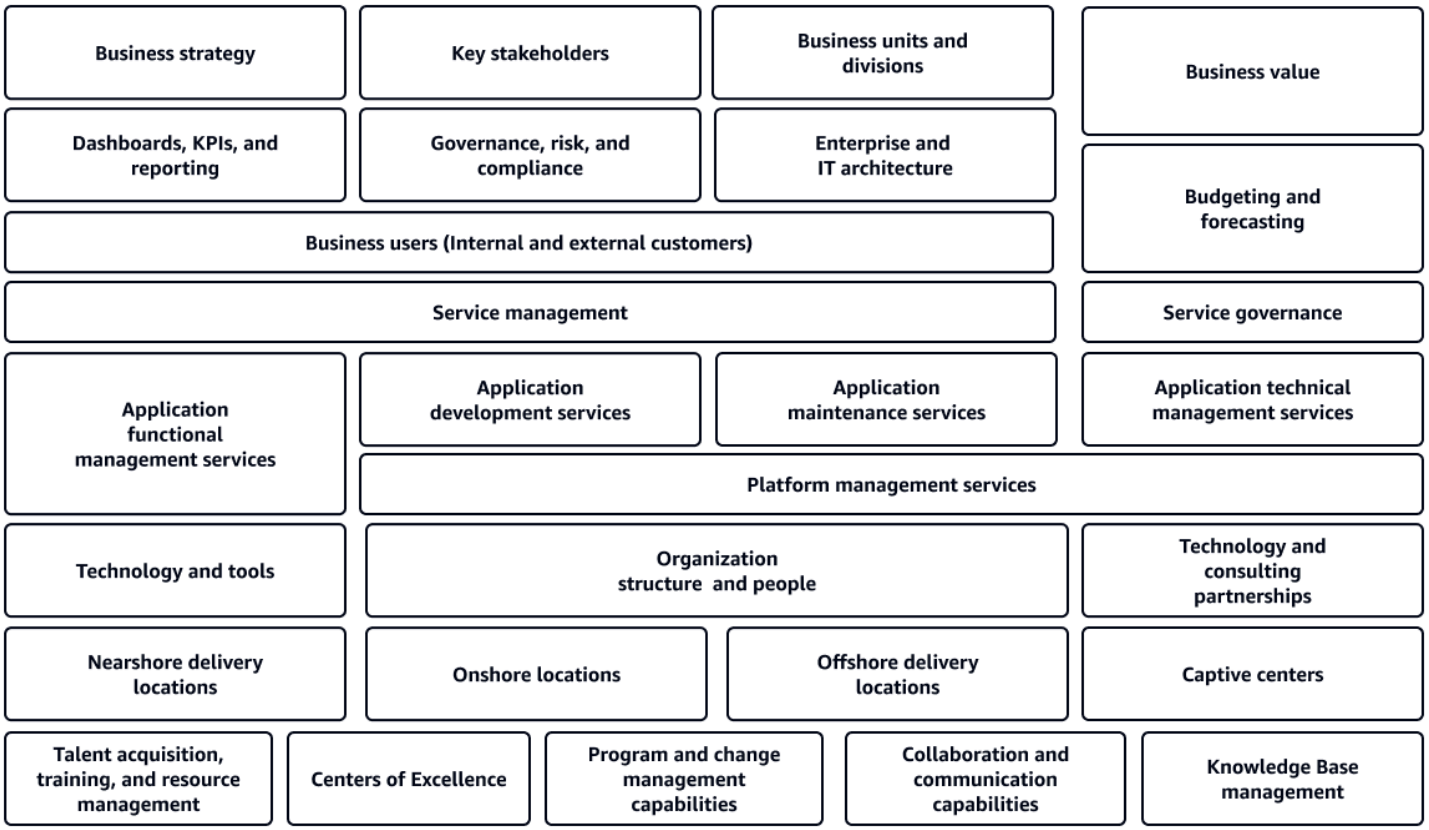
Overview of a typical IT operating model

An operating model serves as the cornerstone of successful IT service delivery in any organization. It's the blueprint that defines how an organization creates and delivers value through its operations. At its core, an operating model aligns people, processes, and various technologies with business strategy. (For more information about operating models, see Defining the IT Operating Model on The Open Group website.)

As shown in the following diagram, a typical IT operating model encompasses multiple key components:

- Organizational structure and roles
  - Key stakeholders
  - Business units and divisions
  - Business users (internal and external customers)
  - People roles
  - Technology and consulting partnerships
- Governance and decision-making frameworks
- Enterprise and IT architecture
- Core processes and workflows
  - Business strategy
  - Business value

    - Budgeting and forecasting
    - Application functional management services
    - Application development services
    - Application maintenance services
    - Application technology management services
    - Platform management services
- Technology and tools
- Performance metrics
    - Dashboards, key performance indicators (KPIs), and reporting
- Organization capabilities
    - Program and change management
    - Collaboration and communication
    - Knowledge base management
- Culture and ways of working
    - Talent acquisition, training, and resource management
    - Center of Excellence (COE)
    - Nearshore delivery locations
    - Offshore locations
    - Offshore delivery locations
    - Captive centers



A well-designed operating model does more than explain day-to-day operations. It's a strategic asset driving competitive advantage. The operating model enables organizations to respond quickly to market changes, innovate effectively, and deliver greater customer value. A key strength of a well-designed operating model is adaptability. Your organization's operating model must flex to support its chosen practices while maintaining consistency and efficiency. This ability to adapt applies whether you use traditional waterfall methodologies, agile frameworks, or a hybrid approach for your ADM.

Overview of an ADM operating model

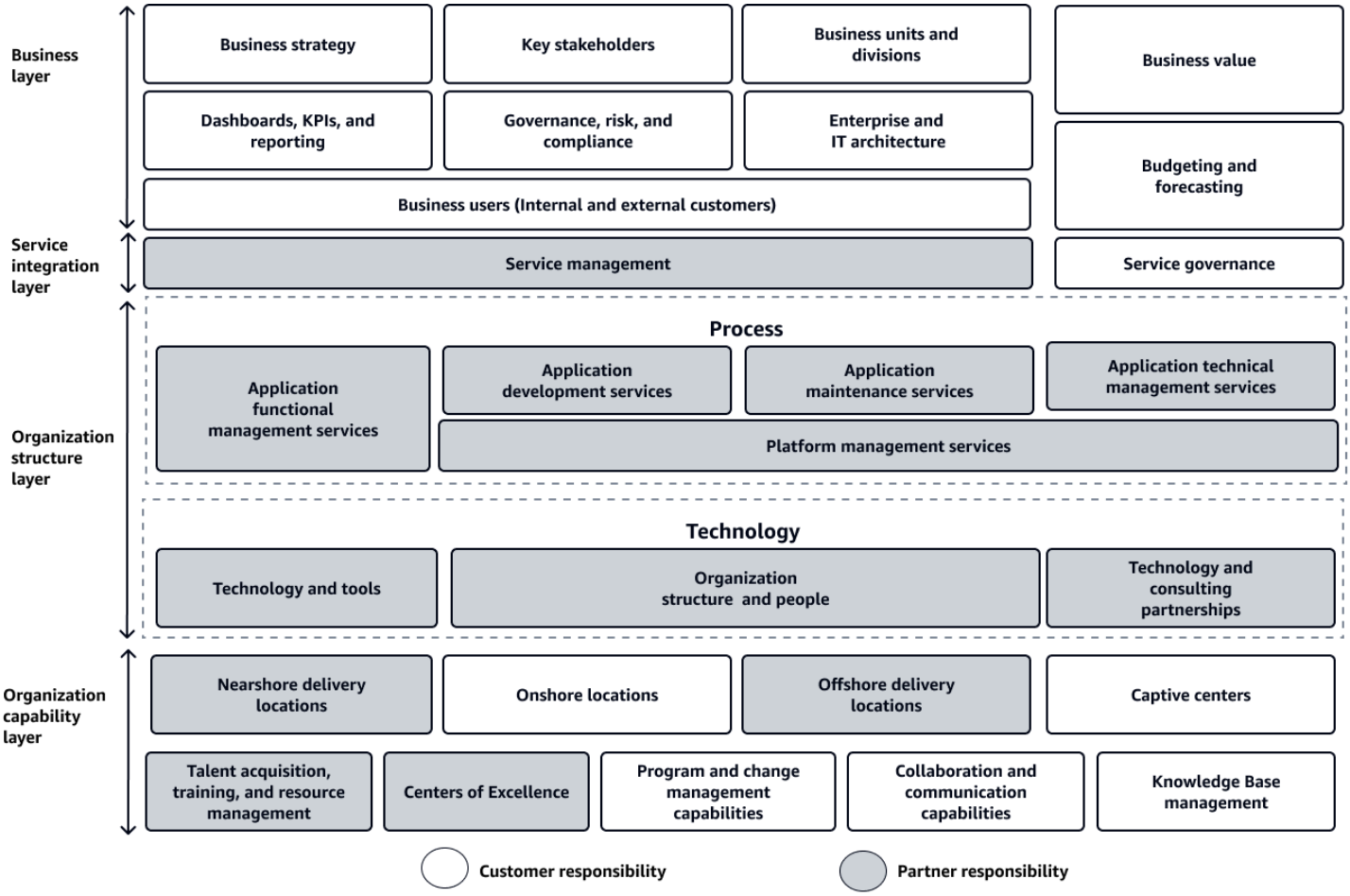
Transitioning from typical IT operating model concepts to the specific context of ADM requires understanding how these principles apply to software development and maintenance processes. The ADM operating model provides a comprehensive framework to manage the entire application lifecycle from planning to development to maintenance. It helps achieve successful alignment between business goals and IT execution.

Creating an ADM operating model is typically a shared responsibility between the *customer* (business and in-house IT) and/or *partners* (application managed services (AMS)) delivered by

consulting and technology partners). This collaborative approach makes use of diverse expertise and aligns with the organization's specific needs and technological landscape.

As shown in the following diagram, an ADM model consists of interconnected layers that play critical roles:

- Business layer – This top layer aligns ADM activities with the organization's strategic goals. Here, leaders define business strategy, shape enterprise architecture, and establish governance mechanisms. As generative AI integration becomes more prevalent, this layer becomes increasingly dynamic. It facilitates rapid and continuous alignment between business objectives and development activities.
- Service integration layer – This operational nexus bridges the gap between business needs and technical implementation. As organizations introduce generative AI, this layer orchestrates complex interactions between human teams and AI systems to deliver seamless services.
- Organization structure layer – This layer focuses on people, process, and technology, and it experiences significant changes during AI integration. Roles will evolve, teams will reimagine processes, and the technology stack will expand to include AI tools. This layer drives the practical implementation of an organization's generative AI transformation.
- Organization capability layer – This foundational layer involves the strategic distribution of resources globally and the cultivation of essential skills and expertise needed for AI-augmented ADM. As AI integration progresses, this layer plays a crucial role in developing new competencies, establishing Centers of Excellence (COE), and fostering a culture of continuous learning.



As organizations prepare to integrate generative AI into their ADM practices, they can reshape each layer of this model as necessary. Organizations can reimagine SDLC processes, redefine roles, and recalibrate technology stacks to benefit fully from generative AI.

The true power of an ADM operating model lies in its ability to transform and manage change. This transformation requires close collaboration among all stakeholders to help ensure a cohesive and effective implementation of AI-augmented ADM practices.

For more details about each layer, see the following sections:

- Business layer elements in an ADM operating model
- Service integration layer elements in an ADM operating model
- Organization structure layer elements in an ADM operating model
- Organization capability layer elements in an ADM operating model

Business layer elements in an ADM operating model

The customer is responsible for activities related to the following elements:

- Business strategy
  - Improve the customer experience and drive key business outcomes
  - Modernize core systems for high business impact
  - Enhance agility and innovation capabilities
- Business line and support functions (Geographic areas and country)
  - LOBs
  - Marketing
  - Human resources
  - Procurement
  - Legal
  - Information technology (IT)
- Dashboards, KPIs, and reporting
  - Service performance reporting
  - Service level agreement (SLA) and operating level agreement (OLA) monitoring and reporting
  - Business performance reporting
- Governance, risk, and compliance
  - Steering committee and quarterly review
  - Risk assessment and management
  - Audits, compliance, and regulatory reporting
- Enterprise and IT architecture
  - Business-aligned IT strategy
  - Architecture and design principles
  - Technology standards and policies
- Budgeting and forecasting
  - Budget planning and control
  - Financial performance management
  - Demand forecasting and planning

- Business value
  - Improve resiliency
  - Improve productivity
  - Improve business agility
  - New feature release

Service integration layer elements in an ADM operating model

This layer includes the following key areas of service management (responsibility of the consulting and technology partners) and service governance (responsibility of the customer):

- **Service management** encompasses delivery of IT services including service desk, incident and problem management, change management, and service level management. AI-powered automation and intelligent support capabilities enhance service quality and efficiency.
- **Service governance** focuses on oversight and control mechanisms including service validation, availability management, capacity planning, and configuration management. With effective service governance, services align with business objectives while maintaining compliance and performance standards.

Organization structure layer elements in an ADM operating model

The organization layer focuses on people, process, and technology.

Partners are responsible for activities related to *people* elements. In some cases, customers have a co-sourced engagement model resulting in shared responsibility for the following:

- Organization structure and people roles
  - Product management – Project owner and business analyst
  - Project management – Project manager, Scrum Master, and Agile Coach
  - Architecture and design – Solution architect, technical lead, and user experience (UX) designer
  - Development – Software developer and user interface (UI) designer
  - Quality assurance – Testing lead, quality assurance (QA) tester, and performance engineer
  - Operations – DevOps engineer and release manager
  - Support and maintenance – Support engineer and technical writer

    - Subject matter experts (SME) – Security subject matter experts (SMEs), integration SME, and domain-specific SMEs

Partners are responsible for activities related to the following *process* elements:

- Application functional management services
    - Business process management
    - Information and data management
    - Functional management
- Application development services
    - Project and requirement management
    - Architecture
    - Design and development
    - Testing and quality assurance (QA)
- Application maintenance services (Operations)
    - Services support management (ITSM)
    - Service request management
    - Updates and patch management
    - Service improvements
- Application technical management services
    - Application basics support (Level 1)
    - Middleware management
    - Database management
    - Service improvements
- Platform management services
    - Managed landing zone
    - Managed operating system (OS)
    - Database
    - Observability
    - Security
    - Network

- Backup
- Integration
- Cloud financial
- Other services

Partners are responsible for activities related to implementation and management of the following *technology* elements:

- Technology and tools
  - Includes cloud, virtualization, containers, operating systems, databases, and other management tools
  - Developer tools and integrated development environments (IDEs)
  - Continuous integration and continuous development (CI/CD) tools
  - Bug fix and IT service management (ITSM) tools
- Technology and consulting partnerships
  - Hyperscaler (AWS and others)
  - Technology ISVs
  - IT/Service desk supplier
  - Infrastructure outsourcing (Network, data center, security, and workplace services)

Organization capability layer elements in an ADM operating model

Generally, customers hold accountability for making the key decisions about the following activities:

- Program and change management capabilities
  - Portfolio management
  - Feature and backlog management
  - Organization change management
- Collaboration and communication capabilities
  - Productivity tools
  - Collaboration tools

    - Communication tools
- Knowledge base management
    - Market research
    - Customer feedback and issues resolutions
    - Business and domain knowledge
- Onshore locations, such as corporate office, regional office, or remote sites
- Captive centers

One or more consulting partners are responsible for implementing and managing activities related to the following elements:

- Nearshore delivery locations
- Offshore delivery locations
- Talent acquisition, training and certification, and resource management
- Centers of Excellence
    - Innovation
    - Technology evaluation and proof of concept (POC)
    - POVs, best practices, standards, and policies

Integrating generative AI into ADM practices

Generative AI reshapes ADM practices across all layers of the operating model. This transformative technology can shift an organization's focus from cost management to value creation and innovation, enabling more agile and customer-centric approaches.

This section provides an overview of how generative AI reshapes each of the following layers of the ADM operating model:

- Business layer
- Service integration layer
- Organization structure layer
- Organization capability layer

For each layer, an overview of its current state and an AI-powered future state provides insights into the transformative potential of generative AI integration. Additionally, the following sections can help you navigate the AI-driven evolution in ADM practices:

- Integration challenges and mitigation strategies
- Action areas and recommendations

Understanding these changes can help you make use of generative AI effectively in enhancing your organization's software development and maintenance capabilities.

Business layer of an ADM operating model

The business layer forms the strategic foundation of the ADM operating model. Generative AI is transforming business strategy, stakeholder roles, and key areas such as enterprise architecture, reporting, governance, and budgeting.

Strategy and key stakeholders

The ADM operating model includes both internal and external stakeholders focused on aligning business strategy and goals with organizational operations and outcomes. Traditionally, these stakeholders prioritized application reliability, release velocity, operational efficiency, cost reduction, and application rationalization.

In a shift from traditional methods to AI-enhanced processes, the following key changes occur in stakeholder roles and priorities:

- **Strategic focus** – Shift from cost management to value creation and innovation.
- **Collaborative decision-making** – AI-driven insights inform cross-functional strategies.
- **Agile responsiveness** – Faster adaptation to market changes and user needs.
- **Customer-centric approach** – Enhanced focus on user experience and satisfaction.
- **Continuous learning** – Emphasis on AI literacy and ongoing skill development.

These changes ripple through various aspects of the business and service integration layers, affecting the following key areas:

- Enterprise and IT architecture
- Dashboards and reporting
- Governance, risk, and compliance
- Budgeting and forecasting

Enterprise and IT architecture

The following table provides the current state and a corresponding future state with generative AI for key issues related to enterprise and IT architecture.

Current state	Future state with generative AI
Manual creation and updating of architecture documentation	Automated architecture documentation and reviews
Static impact analysis of architectural changes	Real-time impact analysis of architectural changes
Fixed roadmaps with infrequent updates	Adaptive roadmaps responding to market changes
Technical jargon-heavy communication of architectural concepts	AI-powered natural language interfaces for architectural concepts

Dashboards and reporting

The following table provides the current state and a corresponding future state with generative AI for key issues related to dashboards and reporting.

Current state	Future state with generative AI
Static dashboards with generic insights	Real-time adaptive dashboards with user-specific insights
Reactive issue management	Predictive analytics for addressing issues proactively
Technical query languages for data access	Natural language querying for non-technical stakeholders
Manual report generation and key performance indicator (KPI) tracking	Automated report generation and intelligent KPI suggestions

Governance, risk, and compliance

The following table provides the current state and a corresponding future state with generative AI for key issues related to governance, risk, and compliance.

Current state	Future state with generative AI
Manual policy checking and compliance audits	Automated policy checking and compliance monitoring
Periodic risk assessments based on historical data	Intelligent risk assessment with early warnings and mitigation strategies
Static compliance documentation	Dynamic compliance documentation generation and updates

Budgeting and forecasting

The following table provides the current state and a corresponding future state with generative AI for key issues related to budgeting and forecasting.

Current state	Future state with generative AI
Historical data-based manual cost modeling	Predictive cost modeling based on historical data
Periodic resource allocation adjustments	Dynamic resource allocation in real time
Limited scenario planning due to time constraints	Automated scenario planning for budget evaluations
Subjective project prioritization	Intelligent project prioritization aligned with business objectives

Service integration layer of an ADM operating model

The service integration layer acts as a critical bridge between business requirements and technical execution, orchestrating interactions across IT services. The integration of AI into this layer brings changes in *service management* and *service governance*.

Service management

The following table provides the current state and a corresponding future state with generative AI for key issues related to service management.

Current state	Future state with generative AI
Self-help using internal knowledge base search and manually created standard operating procedures (SOPs)	AI-powered self-service agents that generate dynamic SOPs using an enterprise repository
Self-service tools for standard service requests such as access to data and software installat ion	Automated service requests using AI-powered agent workflows

Human agents responding to user inquiries	AI-powered chatbots for instant, context-aware responses
Limited language and communication channel options	Multi-language, multi-channel support across chat, voice, SMS, and virtual assistants
Reactive issue management	AI-powered service desk that predicts common issues and proactively suggests solutions to users before they encounter problems

Service governance

The following table provides the current state and a corresponding future state with generative AI for key issues related to service governance.

Current state	Future state with generative AI
Reactive approach to service level agreement (SLA) management	Predictive service level management to forecast potential SLA breaches
Manual availability management	AI-enhanced availability management for continuous service delivery
Static capacity and performance management	Intelligent capacity and performance management for optimized resource allocation
Manual service validation and testing	Automated service validation and testing
Periodic configuration management database (CMDB) updates	AI-driven configuration management for real-time CMDB updates

The previous examples of future state with generative AI for the business layer and the service integration layer are just the beginning. As AI technologies evolve, expect more innovative solutions to emerge. These advancements can help to enhance proactive, efficient, and automated IT service management and governance.

Use these examples as a starting point for your organization's approach to generative AI transformation. Consider these examples along with your ADM operating model changes. Continuously evaluate new AI applications that align with your organization's needs and goals. This forward-thinking approach can help to keep you at the forefront of IT service management (ITSM) innovation.

Organization structure layer of an ADM operating model

The organization structure layer encompasses people, process, and technology. This layer is where the most visible and profound changes occur when organizations introduce generative AI in the ADM operating model. Roles evolve, organizations reimagine processes, and technology stacks expand to include generative AI tools.

This section provides insights into the practical implementation of generative AI in your organization's ADM transformation, covering changes in organizational structure, individual roles, and core processes. By embracing these strategic shifts, you can position your organization to integrate generative AI in the ADM operating model effectively. This transformation can improve development speed, software quality, and innovation capacity, potentially enhancing your competitive edge. The actual impact will vary based on your organization's specific context and implementation.

Platform management services, technology and tools, and partnerships

Platform management services provide a core set of shared capabilities and standardized services for application teams, including:

- Codified reference architectures and design patterns
- Self-service mechanisms for deploying approved architectures and configurations
- Standardized development, observability, and operational tools
- Support for setting up environments, continuous integration and continuous deployment (CI/CD) pipelines, and management processes
- Centralized governance and security standards

Typically, platform engineering and cloud operations teams manage these services, collaborating to support application teams and drive continuous improvement.

Generative AI is transforming platform management services in the following ways:

- **An AI assistant for architecture recommendations** suggests optimal reference architectures based on project requirements, recommended design patterns and organizational standards.
- **Intelligent self-service provisioning** uses AI to automate and optimize the deployment of resources and services addressing complex workflows.
- **AI-powered observability** provides deeper insights and automates anomaly detection across the platform.
- **AIOps agents** handle multiple automated remediation workflows using approved standard operating procedures (SOPs).
- **Automated compliance checking** continuously verifies and enforces governance and security standards using AI.

These AI-powered enhancements allow infrastructure teams to focus on resolving complex time-consuming issues and improving an application's reliability, enhancing the efficiency and effectiveness of platform management.

Integrate generative AI capabilities into your managed services partners' existing platform offerings. With this strategy, you can achieve the following benefits:

- Harness advanced AI technologies and make use of your partners' expertise and proven processes.
- Enhance your platform engineering and cloud operations with integrated AI capabilities.
- Maintain the benefits of your established managed services partner relationships while advancing your AI capabilities.

Organization structure and roles

Generative AI integration necessitates a reimagining of ADM organizational structure. Adapting the responsibilities of key roles within your organization structure is crucial. These AI-driven changes can help your teams to work more efficiently and deliver higher value.

The organization structure depends on several factors:

- **Engagement size** – Examples include the scope and complexity of applications such as trading systems, drug discovery, and enterprise resource planning (ERP).

- **Specific customer needs** – Examples include Payment Card Industry Data Security Standard (PCI DSS) compliance for payment systems and Good Practice (GxP) compliance for pharmaceutical industries.
- **Methodology used** – Examples include agile and waterfall methodologies.

Some roles combine or expand based on project requirements. Projects involving advanced technologies or strict compliance needs often include specialized roles such as data scientists, machine learning (ML) specialists, Advanced Business Application Programming (ABAP) developers, and compliance officers.

The following sections spotlight common roles in ADM that are evolving with generative AI integration. These roles are expanding and adapting to use AI capabilities, which can enhance their value and impact within the organization. This evolution represents opportunities for skill development and career growth across many roles. The following aspects provide insights into how each role evolves as it integrates with generative AI:

- **Current focus** – The primary tasks that the person in the role performs currently
- **AI-driven shift** – The ways in which generative AI can be incorporated into the role
- **Key benefits** – The benefits gained by incorporating generative AI into the role
- **Key considerations** – The considerations when considering an AI-driven shift for the role
- **Key steps** – The primary steps that the person in the role can take to help them adapt to AI

This comprehensive view can help you understand the current state, direction of change, and steps needed to navigate the AI-driven transformation for each role successfully. You can gain insights into how AI is enhancing existing roles, and how to prepare your organization structure for these advancements.

Product owner or business analyst

The following table provides an overview of how the product owner or business analyst roles can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Requirements gathering • Feature prioritization

	• Stakeholder communication
AI-driven shift	Make use of AI for: • Data-driven decision-making process and accelerated market insights • Business requirements document (BRD) creation, prioritizing the features based on customer feedback and demands
Key benefits	• Faster requirements gathering and analysis • Improved feature alignment with market needs • More comprehensive user stories and use cases
Key considerations	• Ensuring that AI comprehends complex business contexts • Maintaining meaningful stakeholder relationships
Key steps	• Implement AI-powered market analysis and requirements tools. • Develop prompt engineering skills for effective AI interaction. • Establish stakeholder processes to validate AI-generated insights

Project manager

The following table provides an overview of how the project manager role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Activity planning

	• Resource allocation • Risk management
AI-driven shift	• Adopt AI for enhanced predictive planning and real-time project intelligence.
Key benefits	• Improved resource allocation accuracy • Enhanced risk identification and mitigation • Real-time project health monitoring and predictive analytics
Key considerations	• Balancing AI recommendations with human judgment • Ensuring team adoption of AI-driven methodologies
Key steps	• Integrate AI-driven project planning and risk assessment tools. • Develop protocols for AI-human collabora tive decision making. • Upskill team in AI-augmented project management practices.

UI/UX designer

The following table provides an overview of how the user interface/user experience (UI/UX) designer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Creating user interface designs and prototypes • Conducting user research and usability testing

	• Ensuring optimal user experience across applications
AI-driven shift	• Use AI for rapid design iteration, data-driven user insights, and automated usability testing.
Key benefits	• Faster generation of UI design alternatives • Enhanced user research analysis and persona creation • Automated usability testing and feedback analysis
Key considerations	• Balancing AI-generated designs with brand guidelines and user needs • Maintaining creativity and innovation in an AI-assisted design process
Key steps	• Integrate AI-driven project planning and risk assessment tools. • Develop protocols and process for AI-human collaborative decision making. • Upskill team in AI-augmented project management practices.

Full-stack developer

The following table provides an overview of how the full-stack developer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Creating user interface designs and prototypes • Conducting user research and usability testing

	• Ensuring optimal user experience across applications
AI-driven shift	• Embrace AI for comprehensive full-stack development assistance and optimization.
Key benefits	• Accelerated full-stack code generation and optimization • AI-driven API design and integration • Automated performance tuning across the stack
Key considerations	• Maintaining proficiency across multiple technologies alongside AI tools • Ensuring consistency and integration between AI and manually developed components
Key steps	• Develop expertise in AI-assisted developme nt across the full stack. • Establish process and guidelines for integrating AI-generated and manual code. • Implement continuous learning programs for emerging AI tools in full-stack development.

Solutions architect

The following table provides an overview of how the solutions architect role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Designing comprehensive enterprise-wide solutions

	• Aligning technology solutions with business goals • Ensuring integration and interoperability across systems • Creating detailed design documents
AI-driven shift	• Use AI for rapid solution prototyping, data-driven architecture decisions, automated integration analysis, and design document generation.
Key benefits	• Faster generation and evaluation of solution alternatives • Enhanced alignment of technology solutions with business objectives • Improved assessment of system integration and interoperability • Accelerated creation of comprehensive design documentation
Key considerations	• Ensuring that AI-generated solutions address complex business requirements • Maintaining a holistic view of enterpris e architecture in AI-augmented design processes • Validating the accuracy and completeness of AI-generated design documents

Key steps	• Develop expertise in AI-powered solution design tools and methodologies. • Establish processes for validating AI-generated solution proposals against business needs. • Implement AI-driven tools for continuous solution optimization and integration assessment. • Adopt AI-assisted documentation tools for creating and maintaining design documents.

Software developer

The following table provides an overview of how the software developer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Code writing • Debugging • Maintenance
AI-driven shift	• Embrace AI as a coding companion for enhanced productivity and quality.
Key benefits	• Accelerated code generation and completion • Improved code quality and consistency • Faster bug detection and resolution
Key considerations	• Maintaining code readability and performance in AI-generated code • Balancing AI tool reliance with core programming skills

Key steps	• Improve the use of AI-assisted coding and pair programming techniques. • Establish guidelines for reviewing and optimizing AI-generated code. • Implement continuous learning programs for emerging AI dev tools.

Test engineer

The following table provides an overview of how the test engineer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Test case design • Defect identification • Quality assurance
AI-driven shift	• Implement AI for comprehensive, automated testing strategies.
Key benefits	• Increased automation in test case generatio n and execution • Improved test data quality and coverage • Earlier issue detection through predictive defect analysis
Key considerations	• Ensuring thorough coverage beyond AI-generated test cases • Balancing automated testing with explorato ry methods
Key steps	• Develop skills in AI test strategy design and data modeling.

	• Establish processes for continuous refinement of AI testing models. • Implement AI-augmented exploratory testing processes and techniques.

Release manager

The following table provides an overview of how the release manager role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Planning and coordinating software releases • Managing release schedules and dependenc ies • Ensuring smooth deployment and post-rele ase stability
AI-driven shift	• Adopt AI for intelligent release planning, automated deployment, and predictive stability management.
Key benefits	• AI-driven release planning and risk assessment • Automated deployment and rollback strategies • Predictive post-release monitoring and issue detection
Key considerations	• Balancing AI recommendations with business priorities and constraints • Maintaining control and oversight in automated deployment scenarios
Key steps	• Develop skills in AI-powered release management tools and predictive analytics.

	• Establish processes for human validation of AI-generated release plans. • Implement AI-driven post-release monitorin g and rapid response standard operating procedures (SOP).

Technical lead

The following table provides an overview of how the technical lead role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Overseeing application development and operational processes • Ensuring alignment between development teams and operational requirements • Managing the application lifecycle from development to production • Driving continuous improvement in development and operational efficiency
AI-driven shift	• Make use of AI for enhanced application lifecycle management, automated operation al analytics, and predictive resource optimization.
Key benefits	• Improved coordination between developme nt and operations teams • Enhanced application performance monitoring and predictive maintenance • Automated resource allocation and scaling based on operational analytics • Frequent number of changes

	• Accelerated issue resolution and reduced downtime
Key considerations	• Balancing AI-driven automation with human oversight in critical operations • Ensuring seamless integration of AI tools across the application lifecycle • Managing the cultural shift towards AI-augmented DevOps practices
Key steps	• Develop expertise in AI-powered application lifecycle management tools. • Establish processes for integrating AI insights into development and operational decision making. • Implement AI-driven monitoring and predictive maintenance systems. • Create training programs to upskill teams in AI-augmented DevOps practices.

DevOps engineer

The following table provides an overview of how the DevOps engineer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Implementing and maintaining continuous integration and continuous deployment (CI/CD) pipelines • Automating infrastructure provisioning and management • Ensuring seamless integration between development and operations

AI-driven shift	• Use AI for enhanced automation, predictiv e analytics, and intelligent infrastructure management.
Key benefits	• Accelerated deployment cycles • Improved system reliability and performan ce • Proactive issue detection and resolution
Key considerations	• Integrating AI tools with existing DevOps processes • Balancing automation with necessary human oversight
Key steps	• Implement AI-powered CI/CD pipeline optimization. • Adopt AI-assisted infrastructure as code (IaC) generation tools. • Develop skills in AIOps for predictive maintenance and automated remediation.

Support engineer

The following table provides an overview of how the support engineer role can adapt to use generative AI capabilities.

Aspect of the role	Description
Current focus	• Resolving user issues and incidents • Maintaining the system's reliability • Providing technical assistance to end -users
AI-driven shift	• Adopt AI for intelligent issue triage, automated problem resolution, and predictive support.

Key benefits	• Faster issue resolution times • Improved first-call resolution rates • Proactive identification of potential system issues
Key considerations	• Ensuring that AI systems accurately understand and categorize complex technical issues • Maintaining the human touch in customer interactions
Key steps	• Implement AI-powered knowledge bases for faster issue resolution. • Adopt AI-driven ticket classification and routing systems. • Develop process and skills in working alongside AI chatbots and virtual assistants for customer support.

Organization capability layer of an ADM operating model

Traditionally, organizational capabilities such as knowledge management, communication and collaboration, and program or change management tools lack an AI-specific focus. As you integrate generative AI into your ADM practices, your organizational capabilities must evolve. This section outlines key areas for transformation and strategies to make effective use of your AMS partners. This section also explores how AI drives global resource distribution, cultivates essential skills, develops new competencies, establishes AI CoEs, and fosters a continuous learning culture.

Strategic partners and talent development – To build strategic partnerships and develop talent for AI integration, focus on these key initiatives:

- Implement comprehensive AI training programs.
- Establish AI Centers of Excellence (COEs).
- Use AI for improved career planning, recruitment, training, and resource optimization.
- Implement location-specific AI adoption change management plans.

- Develop best practices, standards, and point of views (POVs) more efficiently by using AI.
- Conduct technology evaluation and proof of concepts (POCs) that are aligned with IT architecture roadmaps.

Operating model redesign – The integration of AI necessitates a redesign of the operating model, including the following changes:

- Redefine roles to incorporate AI-augmented development.
- Assign AI-driven strategic tasks to onshore teams to maintain close collaboration with key decision-makers.
- Develop new QA processes for AI-generated code.

Enhanced collaboration and knowledge management – Consider enhancing collaboration and knowledge management through these approaches:

- Implement AI-powered collaboration tools to reduce time zone dependencies.
- Make use of AI to catalog and index enterprise knowledge more effectively.
- Use AI-driven insights from customer feedback, issue resolution, and industry trends for accelerated market research and business requirements analysis.

Governance and compliance – To help ensure proper governance and compliance when integrating AI in an operating model, consider implementing the following measures:

- Establish a global AI governance framework with location-specific compliance requirements.
- Address IP ownership of AI-generated assets and mitigate infringement risks.

Infrastructure and tools standardization – Efforts to standardize infrastructure and tools across the organization for effective AI integration involves these steps:

- Invest in cloud-based AI-augmented platforms that are accessible from all locations.
- Standardize AI tools and environments globally.

Performance metrics and engagement model adaptation – Adapting performance metrics and engagement models for AI-driven processes includes these key actions:

- Develop new KPIs that account for AI contributions.
- Implement AI-assisted project estimation tools.
- Consider flexible engagement models, including outcome-based pricing.
- Define consumption-based pricing models for AI assets, covering licenses, infrastructure, and managed services efforts.

Program and change management augmentation – To strengthen program and change management, consider these strategies:

- Enhance the co-source model between in-house talent, consulting and AMS partners by using AI.
- Improve knowledge collection, methodology refinement, and experience reuse for new initiatives.

By focusing on these areas, you can integrate generative AI effectively across your global delivery locations and organizational capabilities. This approach helps to accelerate transforming your ADM operating model. It improves decision-making velocity and enhances the delivery of business outcomes while balancing the strengths of each location and addressing the challenges of AI integration.

Integration challenges and mitigation strategies

Although the benefits of integrating generative AI into ADM are substantial, challenges exist. Understanding these obstacles is crucial for developing effective mitigation strategies. The following table provides key challenges and corresponding mitigations for areas that are likely to be affected when integrating generative AI into ADM.

Area	Key challenges	Mitigation strategies
Data management	Data quality and integration challenges	Ensure consistent, high-quality data across diverse systems and processes.
Governance and ethics	AI governance and ethics	Establish clear guidelines for AI use and decision-making.

Workforce adaptation	Cultural adaptation	Prepare the workforce for AI-augmented roles.
Process integration	Integration with existing processes	Incorporate AI into established workflows seamlessly.
Trust, reliability, and human oversight	Validating AI-generated insights and recommendations for consistent accuracy	Maintain appropriate human control while taking advantage of AI automation.
Technical complexity	Lack of skills and experience	Manage the increased intricacy of AI-enhanced systems.
Security and compliance	Lack of data protection and IP ownership guidelines	Maintain data protection and regulatory adherence in AI-driven environments.
Organizational alignment	AI recommendation alignment	Ensure AI suggestions align with organizational policies and best practices.
Platform complexity	Lack of skills and readiness for change	Manage the intricacy of AI-enhanced platform and IT support services.
Outsourcing challenges	Capability gaps in outsourced operations	Address AI-readiness in managed service providers.

Action areas and recommendations

To integrate generative AI into your ADM operating model successfully, consider the recommendations in the following action areas. These recommendations can help you navigate your organization's transformation journey and overcome common challenges.

Governance and strategy – To establish effective AI governance and align it with overall business strategy, consider implementing these key actions:

1. Establish cross-functional AI steering committees with AI champions.
2. Develop clear AI governance policies, including ethical use guidelines.
3. Align KPIs and business objectives with AI capabilities continuously.
4. Collaborate with regulatory bodies on AI-driven compliance processes.

AI Center Of Excellence – To maximize the impact of an AI Center of Excellence (COE) in your ADM practices, focus on these initiatives:

1. Establish and launch a dedicated AI COE to drive adoption, ensure best practices, and provide guidance across ADM.
2. Develop comprehensive COE operating procedures and a service catalog outlining AI-related services and support.
3. Continuously expand COE capabilities through advanced AI research and strategic partnerships.

Education and culture– To support a culture of AI adoption and continuous learning across the organization, consider these actions:

1. Implement comprehensive AI literacy programs across the organization.
2. Foster a culture of experimentation, learning, and adaptation.
3. Create training programs to upskill platform teams in AI-augmented operations.

Technology and process– To integrate AI effectively into your technology stack and processes, prioritize these initiatives:

1. Implement AI-driven tools for architecture recommendation and resource provisioning.
2. Develop AI models for predictive capacity planning and performance optimization.
3. Integrate AI-powered observability and anomaly detection systems.
4. Establish AI-assisted compliance checking and security monitoring processes.
5. Implement standardized data collection frameworks across projects.
6. Develop AI models that accommodate both waterfall and agile methodologies.

Data and security – To support data quality and security efforts, focus on these actions:

1. Invest in data integration, quality assurance, and security processes.

2. Create feedback mechanisms for continuous improvement of AI systems.

Change management – To facilitate smooth adoption of AI technologies, use these change management approaches:

1. Redesign stakeholder communication channels for AI-enhanced collaboration.
2. Implement change management programs to build trust in AI-generated insights.

Skill development – To build the necessary AI capabilities, support this skill development initiative:

- Upskill teams in data science, AI interpretation, and AI-powered tools.

Partnerships – To harness external expertise, consider these ideas for partnerships:

1. Make use of application managed services (AMS) partners for AI implementation.
2. Consider infrastructure and/or CloudOps managed services partners for AI integration across platform engineering services.
3. Use IT services management partners for AI integration with service management and governance services.

Human oversight – To maintain appropriate human control and accountability, implement the following approach:

- Establish protocols for human oversight of AI-generated recommendations.

Embracing these AI-driven changes and addressing challenges systematically can help you create a more agile, efficient, and innovative ADM operating model. The key to success lies in balancing human expertise with AI capabilities, aligning IT services closely with organizational objectives. This approach can drive significant business value, enhance an organization's competitive advantage, and position the organization to lead in the next era of ADM.

Building an AI-powered ADM target operating model

As you consider your ADM practices with generative AI, it's important to design a comprehensive *target operating model* (TOM). A TOM describes the desired state of an organization's operating model. Your organization's ADM TOM should align its people, processes, technology, organization, and governance with its strategic vision.

The following table lists the eight components of a TOM.

TOM component	Component elements
Strategic alignment	• Value drivers • Business goals alignment • AI roadmap
Organizational structure	• AI Centers of Excellence • New AI roles • Cross-functional teams
Talent and skills	• Career paths • Continuous learning • AI literacy requirements • Skills gap analysis
Governance and ethics	• Regulatory compliance • Data privacy framework • AI ethics policies
Performance measurement	• Continuous monitoring • Business impact reporting • Feedback loops • AI-specific KPIs
Partner ecosystem	• Partner evaluation metrics • Data sharing protocols

	• AI capability requirements • Collaborative innovation
Technology and tools	• Data infrastructure • AI tools ecosystem • AI platforms selection • Legacy systems integration
Processes	• AI-enhanced SDLC • AI model management • Governance workflows

Building an ADM TOM is a transformative process that affects every aspect of an organization. Consider each ADM component and its interdependencies carefully to create a robust foundation for your AI-powered SDLC.

Implementing an ADM TOM should be tailored to an organization's specific needs and context. As you implement this model, continuously assess and adjust it based on your organization's unique challenges and opportunities.

The following sections provide more details about the components in the ADM operating model, including their interactions.

Strategic alignment component

The strategic alignment component defines strategic objectives for AI-powered ADM, aligning AI initiatives with business goals. This component articulates AI's value in ADM processes and sets success criteria for AI integration. This component interacts with other components as follows:

- *Value drivers* influence *AI-specific KPIs* in the *performance measurement* component.
- *Business goals alignment* informs the creation of *new AI roles* in the *organizational structure* component.
- The *AI roadmap* guides *AI platforms selection* in the *technology and tools* component.

Organizational structure component

The organizational structure component addresses the design of an ADM organization that supports AI-augmented development with new roles. This component establishes an AI Center of Excellence (COE) and evolves existing roles for AI integration.

- The *AI COE* supports *continuous learning* in the *talent and skills* component.
- *New AI roles* influences new *AI capability requirements* in the *partner ecosystem* component.
- *Cross-functional teams* enables agile integration with *AI-enhanced SDLC* in the *processes* component.

Talent and skills component

The talent and skills component identifies required AI skills and competencies across ADM roles and personnel. This component defines AI literacy requirements and creates AI-focused career paths.

- *Career paths* aligns with *new AI roles* in the *organizational structure* component.
- *AI literacy requirements* supports *AI ethics policies* in the *governance and ethics* component.
- *Skills gap analysis* informs the *AI tools ecosystem* in the *technology and tools* component.

Governance and ethics component

The governance and ethics component establishes an ethical framework for AI use in ADM, including policies and review boards. This component defines data privacy and security requirements for AI-powered ADM practices.

- *Regulatory compliance* affects *value drivers* in the *strategic alignment* component.
- The *data privacy framework* influences *data sharing protocols* in the *partner ecosystem* component.
- *AI ethics policies* guides *AI model management* in the *processes* component.

Performance measurement component

The performance measurement component designs a new framework with AI-specific KPIs for ADM performance measurement. This component outlines the methods to measure, report, and optimize AI impact in ADM.

- *Business impact reporting* influences *partner evaluation metrics* in the *partner ecosystem* component.
- *Feedback loops* supports *continuous learning* in the *talent and skills* component.
- *AI-specific KPIs* informs *business goals alignment* in the *strategic alignment* component.

Partner ecosystem component

The partner ecosystem component defines expectations for AI capabilities in AMS partners and collaborative processes. This component establishes data sharing and model ownership principles for partner interactions.

- *Partner evaluation metrics* informs *AI-specific KPIs* in the *performance measurement* component.
- *AI capability requirements* influences *skill gap analysis* in the *talent and skills* component.
- *Collaborative innovation* supports the *AI tools ecosystem* in the *technology and tools* component.

Technology and tools component

The technology and tools component specifies AI technologies and tools to support transformed ADM processes. This component identifies integration points and data requirements for AI-powered ADM.

- *Data infrastructure* supports *business impact reporting* in the *performance measurement* component.
- *Legacy system integration* affects *AI-enhanced SDLC* in the *processes* component.
- *AI platforms selection* influences *collaborative innovation* in the *partner ecosystem* component.

Processes component

The processes component redesigns the SDLC to incorporate AI, enhancing each stage with AI capabilities. This component develops new processes for AI model management and governance in development.

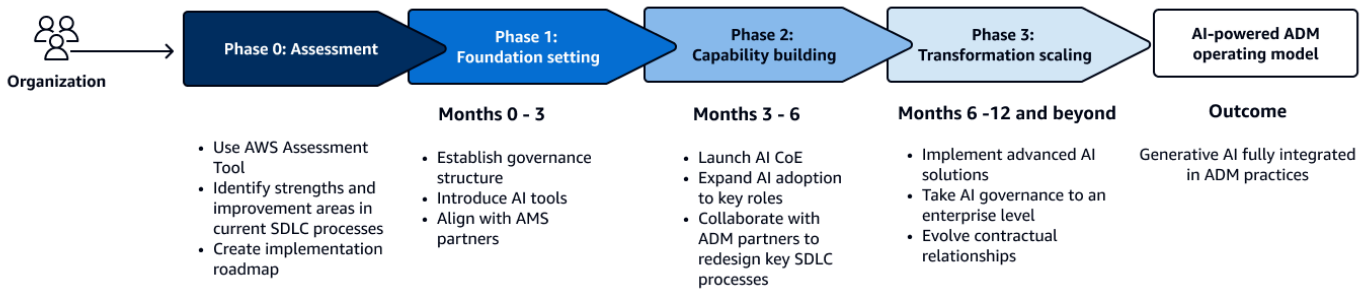
- *AI-enhanced SDLC* affects *continuous monitoring* in the *performance measurement* component.
- *AI model management* relates to *data infrastructure* in the *technology and tools* component.
- *Governance workflows* supports the *data privacy framework* in the *governance and ethics* component.

Implementing an AI-powered ADM target operating model

Use a structured, phased approach to implement a generative AI application development and maintenance (ADM) target operating model (TOM). The following approach balances quick wins with long-term transformative changes while minimizing disruption to current operations. Each phase addresses specific components of the TOM, highlighting their interdependencies and evolution throughout the implementation process.

As shown in the following diagram, the implementation strategy consists of phases that progress from basic to advanced complexity over a 12-month period:

- **Phase 1: Foundation setting** – This phase occurs in months 1–3. It establishes basic governance structures and introduces essential AI tools while achieving quick wins.
- **Phase 2: Capability building** – This phase occurs in months 3-6. It expands AI adoption and addresses processes of medium complexity. Launch your AI COE, expand AI adoption to project management and operations roles, and collaborate with your ADM partners to redesign key SDLC processes using generative AI.
- **Phase 3: Transformation scaling** – This phase occurs in months 6–12 (and beyond). It implements advanced solutions and tackles higher complexity challenges. For example, implement advanced AI solutions for architecture design, full-stack development, and security monitoring. Mature your AI governance to an enterprise level, and evolve your contractual relationships with ADM partners to reflect the new AI-powered reality.



> **Note**
>
> Before beginning implementation, conduct an AI-powered SDLC readiness assessment to establish a baseline of your organization's current SDLC capabilities and identify key areas for improvement. For more details, see Next steps.

Actual timelines can vary based on organizational context, implementation approach, and other factors such as the size and scale of implementation. Some organizations might achieve results in a shorter or longer time span, depending on their specific circumstances and maturity levels.

By progressing through these phases, you can transform your organization's ADM practices systematically, using AI to drive innovation, efficiency, and competitive advantage. For more information about using a phased approach in your organization, see Roadmap for implementing an AI-powered ADM TOM and Best practices for all implementation phases.

Organizations can enhance their in-house capabilities through this transformation journey. This journey also requires continuous adjustment and clear communication with all stakeholders. The result is an integrated, global ADM target operating model for AI-powered software development and maintenance with your consulting and technology service providers.

Roadmap for implementing an AI-powered ADM TOM

The following table provides a reference roadmap that uses a phased approach to implement an ADM TOM while minimizing disruption to current operations. For each ADM component, the roadmap describes the relevant activities that occur in each implementation phase.

ADM component	Foundation setting: Months 1-3	Capability building: Months 3–6	Transformation scaling: Months 6–12 and beyond
Strategic alignment	• Enable AI steering committee. • Set vision, mission, and goals with business alignment.	• Continuously align KPIs and business objectives with AI capabilities. • Maintain clear stakeholders	• Continuously align KPIs and business objectives with AI capabilities. • Maintain clear stakeholders

	• Develop AI technology and tools strategy and roadmap.	communication on AI initiatives with impact. • Review business outcomes and ROI.	communication on AI initiatives with impact. • Review business outcomes and ROI. • Integrate AI governance with EA. • Establish cross-functional AI governance with AMS partners. • Standardize AI tools globally across in-house and AMS partner teams.
Organizational structure	• Identify cross-functional AI champions. • Identify key roles for AI integration.	• Launch AI COE with dedicated team.	• Implement AI-driven organizat ion and continuous optimization.

Talent and skills	• Implement basic AI training program. • Adopt AI tools for high propensit y roles such as software developers and test engineers. • Implement advance AI training program. • Implement role-specific AI training program.	• Implement role-specific AI training program. • Develop AI-focuse d career paths and progression. • Implement shared training programs for onshore and offshore teams.	• Implement role-specific AI training program. • Extend AI adoption to product owners, BA, SA, and domain SMEs. • Establish AI innovation incentive program. • Establish mechanisms for ongoing AI knowledge sharing between your organization and AMS partners.

Governance and ethics	• Develop AI ethics guidelines. • Establish guidelines for AI-related IP and data usage. • Create risk assessment framework. • Collaborate with regulatory bodies for compliance.	• Implement AI governance policies and procedures. • Balance AI automation with human oversight to ensure quality and maintain control.	• Balance AI automation with human oversight to ensure quality and maintain control. • Develop AI-specific project and contract templates and SLAs for AMS partners. • Continuously review and address data privacy and security concerns in AI usage part of the ADM.
Performance measurement	• Establish AI goals and key success metrics for ADM. • Establish key success metrics for large language models (LLMs).	• Develop AI-specific KPIs for ADM processes. • Develop AI-specific KPIs for ADM partner performance. • Implement AI cost allocation and ROI tracking.	• Establish KPIs and implement an ADM and SDLC performance dashboard. • Implement AI-driven insights for continuous improvement of the ADM global delivery model. • Continuously monitor and adjust based on feedback and results.

Partner ecosystem	• Engage AMS partner for transformation planning. • Align AI integrati on roles with AMS partners. • Assess AI readiness with AMS and CloudOps partners. • Review existing AMS contracts for AI integration.	• Establish joint AI COE with AMS and CloudOps partners. • Work with ADM partners to integrate AI in the TOM. • Collaborate with AMS partners to implement advanced AI solutions for ADM.	• Collaborate with AMS partners to implement advanced AI solutions for ADM. • Standardize AI tools and environments with AMS partners. • Regularly assess the impact of AI on the AMS outsourci ng value propositi on. • Consider flexible engagemen t models and outcome-based pricing for AI-enhanced services.
Technology and tools	• Implement AI-powered knowledge bases for faster issue resolution. • Implement AI-powered collabora tion tools. • Adopt AI-assisted coding and testing tools.	• Integrate AI-driven project planning and risk assessmen t tools. • Implement AI-powered release management and predictive maintenance. • Implement AI-assisted project estimator tools.	• Implement AI-driven architecture decision support tools. • Adopt AI-powere d full-stack code generation and optimization tools. • Implement cloud-based AI-augmen ted platforms for all delivery locations.

Processes	• Establish guidelines for integrating AI-generated and manual code. • Establish process and SOPs for AI-powered tools. • Establish feedback loop for continuous improvement of LLMs.	• Redesign ADM processes to incorporate AI in the TOM. • Develop AI-driven SOPs between onshore, nearshore, and offshore locations.	• Establish processes for AI-driven architecture decision and full-stack code generation. • Establish AI-assisted compliance check and security monitoring processes. • Establish mechanism for process improvements on AI-powered ADM operating model.

For information about a framework of an AI vision for ADM that includes a mission statement, objectives, and strategic initiatives, see Appendix A: Sample framework of AI vision for ADM. For a detailed implementation checklist covering governance, organizational structure, roles, processes, and tools across all three phases, see Appendix B: Implementation checklist for an ADM TOM.

Best practices for all implementation phases

The following best practices are important to keep in mind through all implementation phases. For each best practice, its related operating model component is shown, indicating which aspect of the model is most affected:

- Monitor and adjust the approach continuously based on feedback and results. (Performance measurement)
- Communicate clearly with all stakeholders about various AI initiatives and their impact. (Strategic alignment)
- Balance AI automation with human oversight to help ensure quality and maintain control. (Governance and ethics)

- Assess regularly the return on investment (ROI) of AI initiatives and adjust strategy accordingly. (Performance measurement; strategic alignment)
- Address data privacy and security concerns that are specific to AI usage in a global delivery model. (Governance and ethics)
- Evaluate regularly the impact of AI on the outsourcing value proposition and adjust the engagement model as needed. (Partner ecosystem; strategic alignment)

Next steps

This strategic document explores how generative AI influences each layer of the application development and maintenance (ADM) operating model. It describes how potential benefits can be achieved such as increased development velocity, reduced production defects, and improved customer satisfaction scores. To begin your organization's AI-powered software development lifecycle (SDLC) journey and implement its target operating model for next generation ADM, use the following steps.

Successful AI integration requires balancing AI capabilities with human expertise. This balance drives innovation, efficiency, and competitive advantage throughout your organization's SDLC processes and ADM practices. By following these steps, you can position your organization at the forefront of AI-augmented software development. This approach drives significant business value and enhances your competitive edge in the industry.

Step 1: Conduct a readiness assessment

Use the **AI-Powered Software Development Assessment (AISDLC - V1.0)** in the AWS Assessment Tool to evaluate your current SDLC capabilities and the readiness of your current ADM operating model. This assessment can help you:

- Identify strengths and improvement areas in your existing SDLC processes and ADM practices.
- Pinpoint areas where AI can have the most significant impact to your business.
- Prioritize remediation activities and create an implementation roadmap.

Step 2: Build foundational capabilities

To understand and help build the foundational capabilities in your SDLC with generative AI, see Accelerating software development lifecycles on AWS with generative AI. This strategic document provides AWS architecture best practices and can help you with the following tasks to implement your roadmap:

- Establish a solid base for AI integration.
- Align your processes with industry best practices.
- Prepare your teams for AI-augmented development.

Step 3: Implement the phased approach

To implement an ADM target operating model, refer to the roadmap that encompasses all phases from initial quick wins to full AI integration. Make use of the sample framework and implementation checklist.

Success stories from early adopters demonstrate the transformative potential of AI in application development and maintenance.

Resources

AWS blog posts

- Significant new capabilities make it easier to use Amazon Bedrock to build and scale generative AI applications – and achieve impressive results
- Transforming the Software Development Lifecycle (SDLC) with Generative AI

AWS services resources

- Amazon Bedrock Agents
- Amazon Bedrock Agent Flow
- Amazon Bedrock Guardrails
- Amazon Bedrock Knowledge Bases
- Amazon Bedrock Security and Privacy
- Answering business questions with Amazon Quick Q
- What is Amazon Q Business?
- What is Amazon Q Developer?

AWS Solutions Library

- Generative AI Application Builder on AWS
- Guidance for Creating a Customized Coding Companion with Amazon Q Developer
- Guidance for Custom Search of an Enterprise Knowledge Base with Amazon OpenSearch Service

AWS other resources

- A practical approach to using generative AI in the SDLC
- Add intelligence to your developer operations
- Generative AI Competency Partners
- Generative AI Customer Stories
- Partner Success with AWS
- What is AIOps?

- What is SDLC (Software Development Lifecycle)?

Additional resources

- Defining the IT Operating Model, Document No. W17B (The Open Group, September 2017)

Appendix A: Sample framework of AI vision for ADM

Organizations can adapt this sample framework of an AI vision for application development and maintenance (ADM) to articulate their transformation goals. The sample includes an introduction, a clear mission statement, quantifiable objectives, and strategic initiatives that are aligned with measurable success metrics.

Introduction

In today's rapidly evolving digital landscape, organizations must continuously innovate to stay competitive. The Project <*Your Project Name*> initiative represents our bold vision to transform our application development and maintenance (ADM) practices through the strategic integration of generative AI technologies.

By harnessing the power of AI, <*Your Company Name*> aims to enhance our development speed dramatically, code quality, and operational efficiency. This approach fosters unprecedented levels of innovation. This transformation will streamline our processes, empowering our teams to deliver superior software solutions. These solutions will drive tangible business value and growth.

The following document outlines our mission, objectives, and key strategic initiatives to achieve an ADM target operating model (TOM) with generative AI.

Mission statement

To transform our ADM practices and software development lifecycle (SDLC) processes by leveraging generative AI technologies, enabling faster innovation, improved quality, and enhanced business value delivery.

Objectives

1. Accelerate application development and delivery timelines by <*Your Company Value*> percent through AI-assisted processes.
2. Improve code quality and reduce defects by <*Your Company Value*> percent using AI-powered analysis and optimization.
3. Enhance developer productivity by <*Your Company Value*> percent with AI-augmented tools and workflows.
4. Reduce operational costs by <*Your Company Value*> percent through intelligent automation and predictive maintenance.

5. Increase business agility by enabling <*Your Company Value*> x faster response to market changes and customer needs.

Strategic initiatives

To achieve our defined goals and measure success in delivering business value, we will focus on the strategic initiatives that align with our key performance metrics, as shown in the following table.

Strategic initiative	Key tasks	Performance metric
1. AI-powered development environment	1.1 Implement AI-assisted code generation and completion tools. 1.2. Integrate AI-driven code review and optimization processes. 1.3. Develop AI-enhanced testing and quality assurance workflows.	• Time-to-market for new features and applications
2. Intelligent operations and maintenance	2.1 Deploy AI-powered monitoring and predictive maintenance systems. 2.2. Implement AIOps for automated incident response and resolution. 2.3. Leverage AI for capacity planning and resource optimization.	• Defect rates and mean time to resolution • Customer satisfaction scores for delivered applications
3. AI-augmented requirements and design	3.1 Use AI for rapid prototyping and design iteration.	• Developer productivity (Example: lines of code per day, story points completed)

	3.2. Implement AI-assist ed market analysis and requirements gathering. 3.3. Develop AI-powered tools for translating business needs into technical specifications	
4. Talent and organization transformation	4.1 Establish an AI Center of Excellence (COE) for ADM. 4.2. Develop comprehensive AI training programs for all roles. 4.3. Redefine job roles and career paths to incorporate AI skills.	• Return on Investment (ROI) for AI implementation in ADM processes
5. Governance and ethics framework	5.1 Create policies for responsible AI use in ADM processes. 5.2. Establish an AI ethics review board for ongoing oversight. 5.3. Develop guidelines for data privacy and security in AI-powered ADM.	• Compliant on policies, standards, and regulatory needs

Focusing on these strategic initiatives and measuring progress against defined metrics will drive significant improvements in our ADM capabilities. This approach delivers greater value to our business and customers through AI-powered innovation and efficiency. We expect to achieve the following results:

- <*Your Company Value*> percent–<*Your Company Value*> percent increase in development velocity
- <*Your Company Value*> percent–<*Your Company Value*> percent reduction in production defects

- <*Your Company Value*> percent–<*Your Company Value*> percent improvement in customer satisfaction scores

Appendix B: Implementation checklist for an ADM TOM

This comprehensive checklist provides you with a structured approach to implementing an application development and maintenance (ADM) target operating model (TOM). The checklist considers governance, organizational structure, personnel roles, processes, and tools for each of the following phases of implementation:

- Phase 1: Foundation setting
- Phase 2: Capability building
- Phase 3: Transformation scaling

Each phase builds upon the previous phase, enabling organizations to scale their AI capabilities systematically while managing risks and ensuring sustainable enterprise-wide adoption.

Phase 1: Foundation setting

This phase occurs in months 1–3. It establishes basic governance structures and introduces essential AI tools while achieving quick wins.

Governance and organization

1.1. Establish an AI governance steering committee.

1.2. Develop initial AI ethics guidelines for ADM processes.

1.3. Create a baseline AI risk assessment framework.

1.4. Identify key roles for AI integration across ADM teams.

1.5. Define initial AI champion roles within existing teams.

1.6. Outline the vision and mission for an AI Center of Excellence (COE) in ADM.

1.7. Conduct an AI skills gap analysis across ADM teams.

1.8. Develop a basic AI literacy training program for all staff.

1.9. Review existing vendor contracts for AI integration potential.

1.10. Establish initial budgeting guidelines for AI initiatives in ADM.

Roles

1.11. Software developer

- Adopt AI-assisted coding, pair programming, and code completion tools.
- Establish guidelines for reviewing and optimizing AI-generated code.

1.12. Test engineer

- Adopt AI-powered test case generation, execution, and data quality improvement tools.
- Implement AI-augmented exploratory testing techniques.

1.13. UX designer

- Adopt AI-assisted design tools and data-driven design techniques.

1.14. DevOps engineer

- Implement AI-powered CI/CD pipeline optimization.
- Adopt AI-assisted infrastructure as code (IaC) generation tools.

1.15. Support engineer

- Use AI-powered knowledge bases for faster issue resolution.
- Implement AI-driven ticket classification and routing systems.

Processes

1.16. Create clear escalation protocols for complex issues.

1.17. Establish guidelines for integrating AI-generated and manual code.

1.18. Develop new QA processes for AI-generated code.

1.19. Establish processes for human oversight of AI-generated designs.

1.20. Establish processes for continuous refinement of AI testing models.

1.21. Improve knowledge collection, methodology refinement, and experience reuse for new initiatives.

Tools

1.22. Adopt AI-assisted coding, pair programming, and code completion tools.

1.23. Implement AI-driven code quality, consistency checks, and bug detection systems.

1.24. Adopt AI-assisted documentation tools for design documents.

1.25. Implement AI-powered collaboration tools to reduce time zone dependencies.

1.26. Adopt AI-powered test case generation, execution, and data quality improvement tools.

1.27. Implement AI-assisted project estimation tools.

1.28. Set up predictive defect analysis using AI.

1.29. Adopt AI-assisted design tools and data-driven design techniques.

Phase 2: Capability building

This phase occurs in months 3-6. It expands AI adoption and addresses processes of medium complexity.

Governance and organization

2.1. Implement AI governance policies and procedures.

2.2. Establish an AI ethics review process for ADM projects.

2.3. Develop AI-specific KPIs for ADM processes.

2.4. Create new AI-focused roles such as an AI integration specialist.

2.5. Realign team structures to support AI-augmented workflows.

2.6. Launch the AI COE with a dedicated team.

2.7. Establish COE operating procedures and service catalog.

2.8. Implement role-specific AI training programs.

2.9. Develop AI-focused career paths and progression models.

2.10. Develop AI-specific procurement guidelines.

2.11. Implement AI cost allocation and return on investment (ROI) tracking mechanisms.

Roles

2.12. Project manager

- Integrate AI-driven project planning, risk assessment, and resource allocation tools.
- Develop protocols for AI-human collaborative decision making.
- Set up real-time project health monitoring and predictive analytics using AI.

2.13. Release manager

- Adopt AI-powered release management, planning, and risk assessment tools.
- Implement automated deployment and rollback strategies using AI.
- Set up predictive post-release monitoring and issue detection systems.

2.14. Site reliability engineer

- Adopt AI-driven predictive maintenance tools.
- Implement AI-powered anomaly detection and automated remediation systems.

2.15. Technical writer

- Use AI-assisted documentation generation tools.
- Implement AI-powered content optimization and readability analysis.

Processes

2.16. Create feedback loops to improve AI models continuously based on project outcome.

2.17. Implement continuous learning mechanisms for AI support system.

2.18. Implement continuous learning mechanisms for AI prediction models.

2.19. Establish processes for validating AI-generated solution proposals.

2.20. Establish processes for human validation of AI-generated release plans.

Tools

2.21. Integrate AI-driven project planning, risk assessment, and resource allocation tools.

2.22. Set up real-time project health monitoring and predictive analytics using AI.

2.23. Implement AI-driven tools for continuous solution optimization.

2.24. Implement AI-driven user research analysis and persona creation systems.

2.25. Set up automated usability testing and feedback analysis using AI.

2.26. Adopt AI-powered release management, planning, and risk assessment tools.

2.27. Implement automated deployment and rollback strategies using AI.

2.28. Set up predictive post-release monitoring and issue detection systems.

2.29. Implement AI-driven monitoring, predictive maintenance, and resource allocation systems.

2.30. Set up accelerated issue resolution processes using AI.

Phase 3: Transformation scaling

This phase occurs in months 6–12 and beyond. It implements advanced solutions and tackles higher complexity challenges.

Governance and organization

3.1. Integrate AI governance into overall enterprise governance.

3.2. Implement continuous improvement processes for AI policies.

3.3. Establish cross-functional AI governance committees.

3.4. Fully integrate AI roles across all ADM teams.

3.5. Implement AI-driven organizational design optimization.

3.6. Expand COE capabilities to include advanced AI research.

3.7. Establish partnerships with external AI research institutions.

3.8. Implement AI-powered personalized learning paths.

3.9. Establish an AI innovation incentive program for employees.

3.10. Develop AI-specific contract templates and service level agreements (SLAs).

3.11. Implement AI-driven financial forecasting and optimization for ADM.

Roles

3.12. Product owner or business analyst

- Implement AI-powered market analysis and requirements gathering tools.
- Develop prompt engineering skills for effective AI interaction.

3.13. Solutions architect

- Adopt AI-powered solution design tools and methodologies.
- Implement AI-driven tools for continuous solution optimization.

3.14. Full-stack developer

- Adopt AI-powered full-stack code generation and optimization tools.
- Implement AI-driven API design and integration systems.

3.15. Technical lead

- Adopt AI-powered application lifecycle management tools.
- Create training programs to upskill teams in AI-augmented DevOps practices.

3.16. Security subject matter expert (SME)Implement AI-powered threat detection and response systems.

- Adopt AI-assisted security policy generation and compliance checking tools.

3.17. Domain-specific SME

- Use AI tools for domain-specific knowledge extraction and application.
- Implement AI-assisted domain modelling and simulation tools.

Processes

3.18. Redesign enterprise architecture (EA) processes to incorporate AI-driven insights and automation.

3.19. Implement continuous learning mechanisms for AI systems to stay current with evolving regulations.

3.20. Establish clear protocols for human oversight of AI-generated compliance recommendations.

3.21. Establish clear protocols for human oversight of AI-generated recommendations.

3.22. Implement a comprehensive change management strategy.

Tools

3.23. Implement AI-driven architecture decision support systems.

3.24. Set up AI-powered integration and interoperability assessment systems.

3.25. Invest in data integration and quality assurance processes for AI analytics.

3.26. Establish robust security and governance frameworks for AI-driven reporting.

3.27. Implement AI-driven tools for architecture recommendation and resource provisioning.

3.28. Integrate AI-powered observability and anomaly detection systems.

3.29. Establish AI-assisted compliance checking and security monitoring processes.

3.30. Implement AI-powered market analysis and requirements gathering tools.

3.31. Adopt AI-powered solution design tools and methodologies.

3.32. Adopt AI-powered full-stack code generation and optimization tools.

3.33. Implement AI-driven API design and integration systems.

3.34. Set up automated performance tuning across the stack using AI.

3.35. Adopt AI-powered application lifecycle management tools.

3.36. Invest in cloud-based AI-Augmented platforms accessible from all locations.

3.37. Standardize AI tools and environments globally.

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	—	April 18, 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

A2A (Agent-to-Agent)

A stateful protocol for agent-to-agent collaboration supporting task delegation and state transfer.

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.

Agent

An AI system that can autonomously reason, plan, and take actions using tools to achieve goals.

Agent Ops

Operational practices for building, testing, deploying, and running AI agents in production at scale.

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.

Citizen Developer

A business user who creates AI applications using no-code/low-code platforms without specialized technical skills.

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.

FM gateway

A centralized intermediary that controls and normalizes access to foundation models. Also known as an *LLM gateway*.

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.

guardrails (AI)

Safety mechanisms that filter, validate, and constrain agent inputs and outputs to help ensure responsible and safe AI behavior.

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.

human-in-the-loop (HitL)

A workflow pattern where agent execution pauses for human review and approval at critical decision points.

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.

MCP

See Model Context Protocol.

Model Context Protocol (MCP)

A stateless protocol for agent-to-tool communication.

MCP server

A service that exposes one or more tools through the Model Context Protocol.

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.

Shadow AI

Unauthorized AI applications built or used outside of governed channels within an organization.

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.

tool

A function or API that an agent can invoke to perform operations in external systems.

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.

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.