Building an enterprise-ready generative AI platform on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Building an enterprise-ready generative AI platform on AWS

Table of Contents

Building an enterprise-ready generative AI platform on AWS

Ratan Kumar, Jeffrey Zeng, and Viral Shah, Amazon Web Services

June 2025 (document history)

It's common to hear that prototypes are easy, demos are cool, but production is hard. This is especially true with generative AI applications. This strategy document provides comprehensive guidance to help organizations implement secure, scalable, and effective generative AI platforms on AWS. It outlines a four-layer architectural approach that encompasses reliable infrastructure, foundation model selection, robust security and governance, and repeatable application patterns.

AI can help organization deliver differentiated products and services. Among the different domains of AI, generative AI has emerged as a powerful catalyst. It redefines how enterprises can simulate scenarios, optimize operations, and deliver compelling digital experiences to their customers.

While many organizations are creating numerous generative AI prototypes, they struggle to scale these into production-ready solutions. To facilitate widespread generative AI adoption, enterprises must address several critical challenges:

- Infrastructure readiness and scalability
- Security and compliance requirements
- Responsible AI implementation
- Integration with existing applications and processes
- Protection of sensitive data and intellectual property
- Cost optimization and return on investment (ROI) measurement

Additionally, you need to think about the ROI of solutions that are powered by generative AI. How can you integrate generative AI into existing applications? How can you make sure that AI-generated recommendations are safe and reliable? How can you help protect sensitive data and intellectual property when using existing foundation models or fine-tuned models?

By developing enterprise-ready generative AI capabilities, you can overcome these challenges. This strategy document offers a roadmap to help you understand and accelerate generative AI initiatives while maintaining compliance with security and ethical standards. It also describes how

you can use AWS services, such as Amazon Bedrock and Amazon Q, to achieve your target business outcomes. These services can power everything from customer service automations to developer productivity tools. They can also help you implement chat-based assistants and AI agents that transform your business operations.

> **Note**
>
> Before implementing this guidance, we recommend that you assess your organization's generative AI readiness by using the Generative AI workload assessment. This strategy document builds upon the established AWS best practices and patterns for enterprise generative AI adoption that are documented in this guide.

Intended audience

This strategy document is intended for chief executive officers (CEOs), chief technology officer (CTOs), chief information officers (CIOs), and other senior technology leaders who are responsible for driving AI innovation and digital transformation. To use this strategy document, you should be familiar with business and technical management concepts, such as enterprise architectures, cloud computing models, IT governance frameworks, strategic technology planning, and digital transformation initiatives.

Objectives

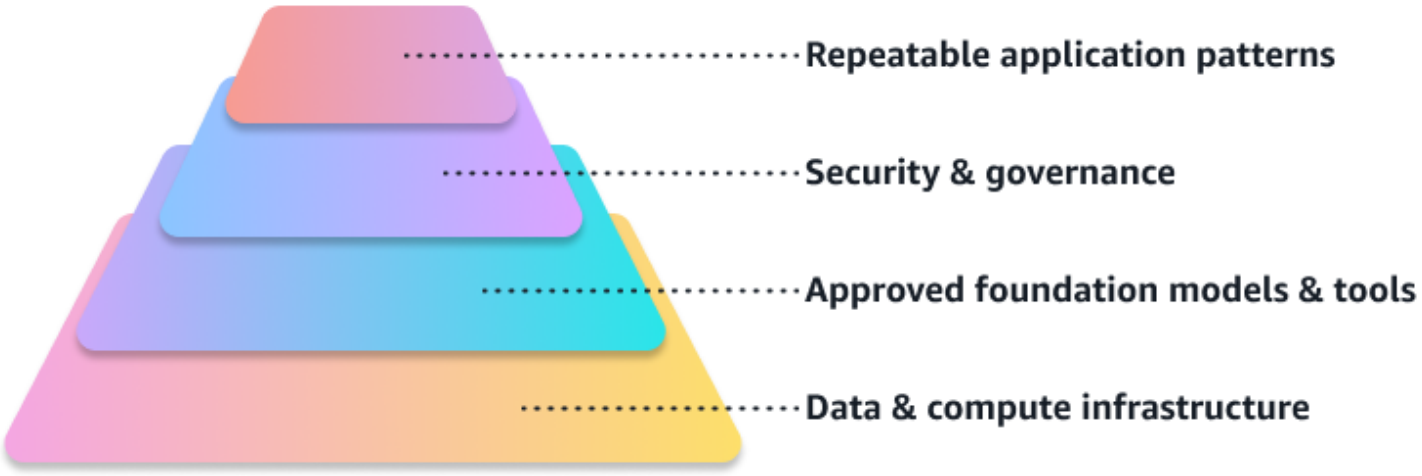
By implementing the recommendations in this strategy document, organizations can achieve the following targeted business outcomes:

- Transform prototypes into production-ready generative AI solutions that deliver measurable business value
- Help distributed teams experiment faster while maintaining security and governance
- Establish enterprise-wide access to approved foundation models with proper controls and governance
- Reduce development time and costs through reusable application patterns and standardized tooling
- Accelerate innovation through secure, self-service access to generative AI capabilities
- Implement responsible AI practices that promote ethical use and compliance

- Create a scalable framework for evaluating and adopting new foundation models
- Build trust in AI-generated content through proper validation and control mechanisms
- Optimize operational costs while maintaining high performance and security standards
- Seamlessly integrate generative AI capabilities into existing enterprise applications

Adopting a layered approach for generative AI development

To build a robust and scalable generative AI platform, organizations need a comprehensive environment that enables innovation while maintaining control and security. This strategy document recommends an enterprise-ready environment that consists of four essential layers, which are shown in the following image.



Each of the following layers plays a crucial role in successful generative AI adoption and scaling across the enterprise:

1. **Data and compute infrastructure** – Reliable infrastructure provides the foundation for developing, training, and deploying generative AI applications. This layer makes sure that organizations have the necessary computational resources and data management capabilities to support various use cases, from small experiments to large-scale production deployments. For Amazon Elastic Compute Cloud (Amazon EC2), consider using AWS Trainium, AWS Inferentia, UltraClusters, Elastic Fabric Adapter, Capacity Blocks for ML, AWS Nitro System, and AWS Neuron.
2. **Approved foundation models and tools** – This layer allows your organization to access approved, pretrained models while maintaining security and compliance. Teams can experiment with different models, evaluate their effectiveness, and select the most appropriate ones for specific use cases. You can use Amazon Bedrock to establish guardrails, build agents, customize models, or import your own custom model.
3. **Security and governance** – This layer helps you make sure that generative AI applications adhere to organizational policies, compliance requirements, legal and privacy requirements,

and ethical guidelines. In this layer, you implement consistent security controls and governance mechanisms across all generative AI initiatives.

4. **Repeatable application patterns** – This layer accelerates development and promotes consistency by providing standardized templates and best practices for common generative AI use cases. Providing repeatable patterns helps prevent duplication of effort and enables rapid scaling of successful solutions. Examples of repeatable patterns include intelligent document processing, Retrieval Augmented Generation (RAG), chat-based assistants, and agentic workflows.

This layered approach provides the following key benefits:

- It clearly separates the different aspects of generative AI implementation.
- It promotes consistent security and governance across all generative AI initiatives.
- It helps you build scalable infrastructure that grows with organizational needs.
- It standardizes approaches that accelerate development.
- It reduces risk through proven patterns and practices.

By implementing these four layers, organizations can create a solid foundation for their generative AI initiatives while maintaining the flexibility to adapt to new requirements and opportunities.

Layer 1: Building reliable data and compute infrastructure for generative AI

For developing generative AI applications, particularly if training or fine-tuning a foundation model is necessary, a robust data foundation and compute infrastructure is critical. As enterprises embark on their generative AI journey, they need infrastructure that can support the entire machine learning lifecycle. They also need to strike the right balance between performance, cost, and operational efficiency.

Reliable generative AI infrastructure should offer the following three key components: the foundational infrastructure, the vector storage and retrieval infrastructure, and the compute infrastructure. Together, these components provide the flexibility to serve the needs of any project, regardless of its scale, requirements, or environment.

This section contains the following topics:

Foundation infrastructure for API-based implementations

Most organizations begin their generative AI journey by using pretrained foundation models through APIs. Amazon Bedrock provides serverless access to leading foundation models through a unified API. This helps you experiment and deploy generative AI applications without managing complex infrastructure. For these implementations, you need the following:

- Amazon Elastic Compute Cloud (Amazon EC2) general-purpose instances for running your applications.
- Amazon Simple Storage Service (Amazon S3) for storing application data and outputs.
- Amazon CloudWatch for monitoring and logging your application's performance and usage.
- (Optional) AWS Lambda for serverless compute when building event-driven generative AI applications.

Vector storage and retrieval infrastructure

As your generative AI applications mature, you'll likely need to enhance them with domain-specific knowledge and context. With Retrieval Augmented Generation (RAG), the foundation model references an authoritative data source that is outside of its training data sources, such as your organization's data or documents, before generating a response. For more information, see Retrieval Augmented Generation options and architectures on AWS.

Amazon Bedrock Knowledge Bases provides a fully managed solution for building RAG applications. It helps you to securely connect your enterprise data with foundation models. For additional flexibility, you can use Amazon OpenSearch Service with vector search capabilities, or you can use Amazon Relational Database Service (Amazon RDS) for PostgreSQL with the pgvector extension to store vector embeddings.

For efficient vector operations, consider using Amazon EC2 compute-optimized instances for embedding generation. Also consider caching frequently accessed embeddings by using Amazon ElastiCache to optimize performance and reduce costs.

For more information about vector databases and caching, see What is a vector database? and Choosing an AWS vector database for RAG use cases.

High-performance compute infrastructure for model training and fine-tuning

For organizations that are ready to customize foundation models, AWS offers comprehensive infrastructure for model training and fine-tuning. You can use Amazon Simple Storage Service (Amazon S3) as low cost, scalable, and highly durable object storage for building data platforms and for storing training data and trained models. In addition, AWS Glue is a serverless data integration service that can help you prepare data for model training.

For the training infrastructure, Amazon SageMaker AI offers a fully managed machine-learning environment with the tools and workflows that you need to build, train, and deploy models. Using SageMaker AI can significantly reduce your operational overhead. Consider using accelerated computing instances, such as the G and P instance families. These instance families provide access to the latest GPUs from NVIDIA for machine learning training and inference. You can also use AWS Trainium, a purpose-built machine learning accelerator that speeds up training times by up to 50%, while lowering costs.

For truly large projects, you can use Amazon EC2 UltraClusters. UltraClusters consist of thousands of accelerated Amazon EC2 instances that are co-located in a given AWS Availability Zone and interconnected. UltraClusters can scale to thousands of GPUs or machine learning accelerators. They deliver exa-floating point operations per second (exaflops) of aggregate compute capacity, which reduces training times and can reduce time-to-solution from weeks to a few days.

Implementation recommendations

Consider the following recommendations for setting up a scalable and cost-effective infrastructure for your generative AI projects:

- For quick experimentation and initial deployments, start with Amazon Bedrock and use general-purpose compute instances for your applications.
- As your needs evolve, implement vector storage solutions by using Amazon Bedrock Knowledge Bases or Amazon OpenSearch Service. Scale your infrastructure accordingly.
- For advanced customization, standardize and automate the provisioning of secure and governed machine learning environments to support the requirements of distributed teams. For more information, see Setting up secure, well-governed machine learning environments on AWS (AWS blog post).
- Adopt machine learning operations (MLOps) to automate and standardize processes across the machine learning lifecycle. These processes include model development, testing, integration, release, and infrastructure management. For more information, see What is MLOps?
- For small-scale experiments or proof of concepts, start with Amazon SageMaker AI and general-purpose compute instances. As you scale to large production deployments, consider Amazon EC2 accelerated computing instances for maximum performance.
- Use managed spot training in SageMaker AI to optimize the cost of training models by up to 90% compared to on-demand instances. SageMaker AI manages the spot interruptions on your behalf.

Layer 2: Approved set of foundation models and tools

As organizations navigate the early stages of generative AI adoption, they quickly realize that no single model can address all use cases effectively. Different models excel in various domains and tasks, and enterprises need to balance capability, cost, and performance for each specific application. This reality drives the need for a flexible, yet controlled, approach to foundation model access.

This section contains the following topics:

Model experimentation and customization

Amazon Bedrock is designed to help you experiment with various foundation models, and it supports scalable production deployments. With Amazon Bedrock Knowledge Bases, you have a fully managed solution to build end-to-end Retrieval Augmented Generation (RAG) workflows. Amazon Bedrock also supports managed agents that can run complex tasks without code, from booking travel to managing inventory. Because it's serverless, Amazon Bedrock reduces infrastructure management concerns and integrates securely with other AWS services.

Additionally, you can use Amazon Bedrock to privately customize the foundation models with your own proprietary data and make them securely available to the users within your organization. For more information, see Customize your model to improve its performance for your use case in the Amazon Bedrock documentation. For access management, AWS Identity and Access Management (IAM) integrates with Amazon Bedrock so that you can configure fine-grained access control. The controls determine which users can enable and access specific models. For more information, see Layer 3: Security and governance for generative AI platforms on AWS in this guide.

Model evaluation

As your organization progresses from generative AI prototypes to production, it's essential to establish rigorous evaluation processes for foundation models. Although open benchmarks offer general insights into model performance, they often fall short when determining a model's

suitability for specific enterprise needs. Tailored evaluation strategies help users select the most appropriate model that aligns with the unique requirements of your organization.

Objectives of custom model evaluation

A custom evaluation approach helps organizations do the following:

- **Assess performance on business-relevant tasks** – Evaluate how well models handle tasks directly related to your enterprise use cases.
- **Identify potential biases and limitations** – Detect areas where models might exhibit biases or fail so that you can make sure the model is suitable for real-world deployment.
- **Compare models with relevant metrics** – Compare different models by using metrics that align you're your organizational priorities and objectives.
- **Make informed model selection** – Make data-driven decisions about model selection, fine-tuning, and deployment to production environments.

Choosing model evaluation metrics

Effective model evaluation draws on a mix of techniques that provide a holistic view of model performance. With these techniques, organizations can assess not only the technical accuracy of a model but also its alignment with enterprise-specific needs. To evaluate a model, you combine quantitative and qualitative metrics to determine performance, identify biases, and choose a model. Organizations should use both ground truth-based metrics (with reference data) and flexible metrics (without reference data) to gain a comprehensive view of model suitability. In AI, *ground truth* refers to data that is factual and is withheld during model training so that you can use it for model evaluation.

Use ground truth-based metrics if reference data exists. These metrics provide concrete, quantitative assessments. Meanwhile, techniques without ground truth can offer flexibility. These techniques help you evaluate models on dimensions such as readability and completeness, even when predefined correct answers aren't available.

Metrics based on ground truth data

If reference data is available, ground truth-based metrics can provide quantitative assessments. The following metrics provide quantitative performance assessments:

- **ROUGE-L score** – Recall-oriented understudy for gisting evaluation (ROUGE) for longest common subsequence (LCS), also known as *ROUGE-L*, measures the longest common subsequence between generated and reference texts. The metric assesses the coherence and content overlap.
- **Cosine similarity** – This metric evaluates semantic similarity between texts. It provides insights into the model's contextual understanding.
- **METEOR score** – Metric for evaluation of translation with explicit ordering (METEOR) scoring combines word alignment and semantic matching to provide a balanced assessment of content accuracy and meaning.
- **Binary similarity** – This metric checks for exact matches, making it particularly useful for tasks that require precise outputs, such as command generation.
- **LLM-as-a-judge score** – This metric uses another large language model (LLM) to rate similarity on a defined scale. It offers a nuanced evaluation of quality.

Amazon SageMaker Clarify and Amazon Bedrock include features to help you evaluate models. They can automate model evaluation jobs so that you can quantify model risks and response quality. For more information, see Evaluate, explain, and detect bias in models and Evaluate the performance of Amazon Bedrock resources.

Metrics without ground truth data

When reference data is unavailable, or as a complementary approach, you can use the *LLM-as-a-judge technique*. It uses a separate model to evaluate the outputs of a generative AI solution based on dimensions that are important to the business. This technique provides flexibility for assessing various model qualities. For more information, see Evaluate model performance using another LLM as a judge in the Amazon Bedrock documentation.

You can use computed metrics to evaluate how effectively a Retrieval Augmented Generation (RAG) system retrieves relevant information from your data sources, and how effective the generated responses are in answering questions. The results of a RAG evaluation allow you to compare different Amazon Bedrock Knowledge Bases and other RAG sources, and then to choose the best Knowledge Base or RAG system for your application. For more information, see Evaluate the performance of RAG sources in the Amazon Bedrock documentation.

Model evaluation techniques in practice

To effectively match enterprise needs with model capabilities, use a robust set of evaluation criteria to direct the model evaluation process. These criteria cover a spectrum of priorities, from correctness and completeness to factuality and sensitive data handling. Each criterion aligns with specific techniques to deliver actionable insights. For instance, when factual accuracy is critical, metrics like ROUGE-L score or cosine similarity provide concrete, quantifiable benchmarks. Conversely, use techniques such as LLM-as-a-judge to evaluate readability and freshness. These techniques offer flexible, qualitative insights tailored to organizational standards. Key dimensions for evaluating models include:

- **Correctness** – Validates the accuracy of information provided
- **Completeness** – Verifies the depth and coverage of responses
- **Readability** – Assesses clarity and ease of understanding
- **Freshness of information** – Checks that content is relevant and current
- **Sensitive data suppression** – Checks for proper handling of confidential information
- **Accuracy** – Measures alignment with factual information
- **Coherence** – Examines logical flow and consistency
- **Factuality** – Verifies the truthfulness of the content
- **Comprehensiveness** – Assesses the scope and thoroughness of coverage

By anchoring model evaluation techniques in these clear criteria, you can thoroughly vet models for their specific purposes. This structured approach aligns models with enterprise goals, compliance requirements, and user expectations. It also lays a strong foundation for deploying generative AI applications at scale in production environments.

Implementation recommendations

To establish an effective foundation model strategy, consider the following recommendations:

- Form a model governance committee that has clear roles, responsibilities, and decision-making processes.
- Develop evaluation criteria and scoring mechanisms for the assessment of foundation models before they are available for use across the organization.

- Remember that the largest foundation model is not necessarily always the best model for your use case. Begin proof-of-concept development with top-tier models to validate business value, and then systematically evaluate smaller models for cost optimization.
- Develop dashboards to track key metrics, such as inference latency, throughput, error rates, and cost per inference.
- Provide clear guidance to teams about how to select the right model for their use case, including experimentation processes and evaluation criteria.

Layer 3: Security and governance for generative AI platforms on AWS

A robust security and governance framework is essential for scaling generative AI adoption across the enterprise. We recommend a platform-centric approach. This approach makes sure that all generative AI powered applications, irrespective of which team builds them, benefit from a default set of security and responsible AI guardrails.

This section contains the following topics:

Core security disciplines

OWASP top 10 for LLM applications describes additional security considerations for generative AI workloads. However, many traditional security disciplines remain crucial. Security considerations for generative AI workloads include the following:

- **Governance and compliance** – Create the necessary policies, procedures, and reporting to empower the business while minimizing risk. For generative AI applications, this encompasses guidelines for model selection, data usage, and output validation.
- **Legal and privacy** – Meet the specific regulatory, legal, and privacy requirements for using or creating generative AI solutions. Organizations must carefully consider data protection laws, intellectual property rights, and industry-specific regulations when implementing generative AI solutions.
- **Risk management** – Identify potential threats to generative AI solutions and recommended mitigations. This includes addressing risks such as data poisoning, prompt injection attacks, or unintended biases in model outputs.
- **Controls** – Implement security controls that mitigate risk. For generative AI applications, this includes input sanitization, output filtering, and access controls for model usage. For more information, see Recommended security controls in this guide.

- **Resilience** – Architect generative AI solutions to maintain availability and meet business service-level agreements (SLAs). This includes considerations for model redundancy, fallback mechanisms, and scalability under varying load conditions.

Recommended security controls

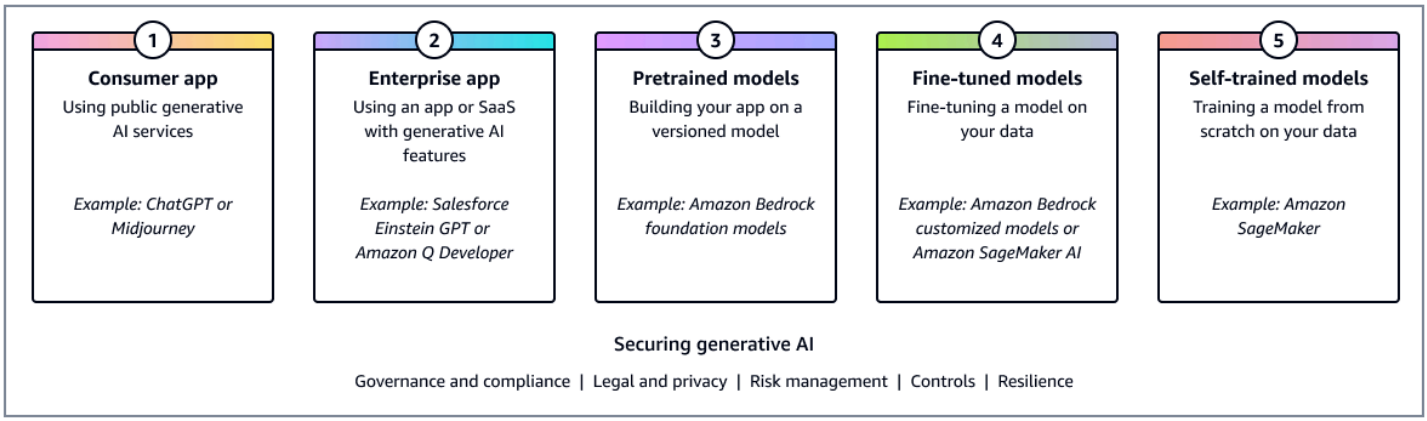
Defense-in-depth is a cybersecurity approach that uses layers of security controls to protect a system, network, or workload. Different layers help detect and stop attacks if one layer fails. At the minimum, we recommend the following controls to adopt a defense-in-depth approach for generative AI workloads and their environments:

- **Identity and access management** – AWS services, including Amazon Bedrock and Amazon SageMaker AI, natively integrate with AWS Identity and Access Management (IAM). IAM provides granular control over who can perform actions on your AWS accounts and resources, such as subscribing to foundation models or running inference on a model. We recommend that you create multiple IAM roles with granular permissions. For example, you might create the following roles:
  - *Evaluation role* – Users with this role can evaluate foundation models in a sandbox environment as new models are added to Amazon Bedrock. These users should coordinate with your legal and procurement teams before activating subscriptions that allow broader access.
  - *General access role* – Users with this role can access foundational models for standard usage.
  - *Fine-tuned model access role* – Users with this role can access models that are fine-tuned with your proprietary data.
  - *Specialized model access role* – Users with this role can access to cutting-edge, high-cost models for specific use cases.
- **Private network access through AWS PrivateLink** – Use AWS PrivateLink with Amazon Bedrock and Amazon SageMaker AI to invoke models from within your VPC and on-premises network. This helps keep sensitive data within your private network.
- **Guardrails for Amazon Bedrock** – Implement guardrails to manage and filter the requests and responses to and from foundation models. This adds an additional layer of control.
- **Securely store invocation logs** – To make sure that use meets your compliance needs, you can store all requests and responses to an Amazon S3 bucket or to Amazon CloudWatch Logs. For more information, see Monitor model invocation using CloudWatch Logs and Amazon S3 in the Amazon Bedrock documentation.

- **Audit and track access to foundation models** – Use Amazon CloudWatch, a service that records actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for Amazon Bedrock as events. For more information, see Monitor Amazon Bedrock API calls using CloudTrail and Logging Amazon SageMaker AI API calls using AWS CloudTrail.
- **Follow OWASP top 10 for LLM security** – Adhere to OWASP's top 10 security risks for LLM applications, and make sure that you have clearly identified strategies and security controls to architect layered defenses. For more information, see Architect defense-in-depth security for generative AI applications using the OWASP Top 10 for LLMs (AWS blog post).

Security scoping matrix

The specific security controls that you need depends on the nature of your generative AI application. The AWS Generative AI Security Scoping Matrix (AWS blog post) provides a framework for understanding the security requirements based on your deployment scope. The following image shows the generative AI security scoping matrix, which is a mental model to classify use cases.



The matrix helps organizations identify the appropriate security controls based on factors such as:

- Whether the application is consumer-facing or internal to the enterprise
- The use of pretrained models compared to fine-tuning
- The sensitivity of the data being processed
- The criticality of the application to business operations

Implementation recommendations

To implement a comprehensive security and governance framework for your generative AI workloads, consider the following recommendations:

- Establish clear policies and procedures for model access and usage.
- Implement security controls at multiple layers, such as at the network, application, and data layers.
- Perform regular security assessments and compliance audits.
- Maintain detailed documentation for security measures and controls.
- Provide security awareness training for teams that work with generative AI.
- Regularly review and update security controls as threats evolve.

Layer 4: Repeatable application patterns for common generative AI use cases

As enterprises adopt generative AI, clear patterns have emerged for how organizations are using this technology to drive business value. These common use cases demonstrate proven approaches to applying generative AI across different business functions. These use cases can help organizations identify and prioritize their own implementation opportunities.

By understanding these established use cases, organizations can better evaluate where generative AI can deliver the most immediate impact while building a foundation for broader adoption. This section describes key areas where enterprises are successfully deploying generative AI solutions.

This section contains the following topics:

Generative AI-powered assistants

Generative AI-powered assistants offer personalized, context-aware support across various business functions. These are tools are ideal for power users. They supercharge the output of those employees who learn to integrate them into their workflow. By using language models, these assistants enhance efficiency, productivity, and user satisfaction across organizations.

AWS offers Amazon Q Business, an enterprise-grade AI assistant that's ready to use and highly customizable. Implementation involves tailoring its knowledge base with organization-specific data to drive accurate, relevant responses. Integration with customer relationship management (CRM) and enterprise resource planning (ERP) systems enhances its utility and promotes smooth data flow and full functionality across the enterprise.

For software development teams, AWS provides Amazon Q Developer. This coding assistant is designed to boost productivity by assisting with code generation, bug detection, refactoring suggestions, and documentation creation. When integrated seamlessly into integrated development environments (IDEs), developers can access its features directly within their usual coding environment, which streamlines workflows and reduces context switching. For more information, see Code generation and analysis in this strategy document.

Common use cases for generative AI-powered assistants include the following:

- Executive assistants that support schedule management, report summarization, and brief preparation.
- HR assistants that support employee onboarding, answer policy-related questions, and help with performance management.
- Sales enablement tools that deliver real-time product information and competitive insights.

To maximize the effectiveness of generative AI-powered assistants, consider starting with a targeted deployment, such as within a specific department. This helps development teams to evaluate the impact and gather user feedback before rolling it out more broadly across the organization. This phased approach smooths the adoption process and optimizes the assistant's alignment with enterprise needs.

For more information about building enterprise generative AI-powered assistants, see Develop a fully automated chat-based assistant using Amazon Bedrock agents and knowledge bases.

RAG chat-based assistants

Out-of-the-box foundation models, while powerful, are constrained by their pretraining data. They lack knowledge of recent events beyond their training cutoff date and have no understanding of company-specific processes or custom data. This limitation makes them unsuitable for many business use cases that require knowledge of the organization.

Customizing models through fine-tuning is a complex and expensive process, and collecting high-quality training data for it can be a challenge. Retrieval Augmented Generation (RAG) addresses these limitations by combining large language models (LLMs) with an organization's proprietary data. RAG approaches create AI-powered interactions that are informed by specific domain knowledge. For more information about RAG and fine-tuning, see Generative AI options for querying custom documents.

Common use cases for RAG chat-based assistants include the following:

- Customer support systems that have access to product documentation and FAQs
- Internal knowledge management tools that quickly retrieve company-specific information
- Legal or compliance chat-based assistants that reference organizational policies and regulations

AWS simplifies the implementation of RAG chat-based assistants through Amazon Bedrock Knowledge Bases. This service helps you set up a vector database to query and retrieve data from custom documents. Key features include integration with Amazon Simple Storage Service (Amazon S3) for data storage, compatibility with third-party services (such as Microsoft SharePoint and Atlassian Confluence), efficient vector search capabilities for relevant document retrieval, and integration with foundation models (such as Anthropic Claude) for building conversational chat-based assistants. To begin, identify a well-defined, limited-scope knowledge base and use it to create a proof-of-concept RAG chat-based assistant. As implementations scale, pay attention to data freshness and the computational costs of large-scale retrieval operations.

For more information about building RAG chat-based assistants, see the following:

- Retrieval Augmented Generation options and architectures on AWS
- Deploy a RAG use case on AWS by using Terraform and Amazon Bedrock
- Develop advanced generative AI chat-based assistants by using RAG and ReAct prompting

Generative AI agents

Generative AI agents represent a significant advancement in automating complex business processes. These agents can handle sophisticated workflows that traditionally required human intervention by breaking down tasks and interacting with various systems.

Typical applications of generative AI agents include automated workflow orchestration for multi-step business processes, intelligent virtual assistants for cross-departmental task completion, and data analysis and report generation agents that can query multiple databases.

AWS provides tools, such as Amazon Bedrock Agents, to help you create and deploy AI agents. Implementation involves defining custom actions and API integrations that are specific to your business needs. You also need to set up robust error handling and logging mechanisms. You can use AWS Lambda for serverless function execution, Amazon API Gateway to manage APIs, and AWS Step Functions for workflow orchestration.

Organizations can start by identifying a repetitive, multi-step process and mapping out its decision points and system interactions. This serves as a blueprint for the initial generative AI agent. Key considerations include validating proper error handling and maintaining transparency in decision-making processes.

Intelligent document processing

Intelligent document processing (IDP) uses generative AI and natural language processing (NLP) to automatically extract, classify, and process information from various types of documents. IDP can transform unstructured document processing into automated, scalable workflows.

Modern IDP solutions can take two approaches: traditional, pipeline-based processing or direct, multimodal processing through foundation models. To build a traditional pipeline on AWS, you can combine Amazon Textract for optical character recognition (OCR) and document understanding with Amazon Bedrock foundation models for advanced analysis. Alternatively, for direct, multimodal processing through foundation models, you can use state-of-the-art foundation models (such as Anthropic Claude in Amazon Bedrock) to directly process images, PDFs, and scanned documents in a single step.

Organizations can choose their approach based on their specific needs. Traditional pipelines excel in high-volume processing and scenarios that require fine-grained control. Direct, multimodal processing through foundation models is ideal for complex documents that require improved accuracy through sophisticated contextual understanding.

Common use cases for IDP include the following:

- Claims processing that automates insurance claim document review, validation, and data extraction.
- Invoice processing that extracts line items, totals, and vendor information for accounts payable automation.
- Contract analysis that reviews legal documents in order to extract key terms, obligations, and expiration dates.
- Medical record processing that analyzes patient records to extract relevant medical information and coding.

Implementation typically uses event-driven architectures that use Amazon S3, AWS Lambda, and Amazon Simple Queue Service (Amazon SQS). When you upload a document to an Amazon

S3 bucket, it initiates processing workflows. Results are stored in databases, such as Amazon DynamoDB, or ingested into business systems through APIs. For optimal results, organizations should implement batching strategies, robust error handling, and audit trails. Start with a specific document type before expanding the scope.

This automation significantly reduces manual intervention, processing time, and human error. It also helps staff to focus on activities that require judgment and expertise.

For more information about document processing workflows, see Automatically extract content from PDF files using Amazon Textract.

Content generation and summarization

Generative AI models excel at generating and summarizing content. This makes them valuable tools for various business applications. You can use the natural language processing capabilities of large language models to create or condense text-based information.

Common use cases for content generation and summarization include the following:

- Automated report generation from structured data
- Content creation for marketing materials, product descriptions, and social media posts
- Document summarization for research papers, legal documents, or lengthy reports

Implementation on AWS typically involves using Amazon Bedrock for text generation and summarization tasks. Amazon Comprehend can be employed for entity recognition and sentiment analysis, enhancing the context and quality of generated content.

Best practices for content generation and summarization applications include implementing human-in-the-loop processes for content review and quality control, using fine-tuned models for domain-specific content generation, and implementing content filters to ensure brand consistency and adherence to guidelines.

For example, you can use generative AI to analyze patient clinical data and generate summaries and clinical notes. For more information, see Building a medical intelligence application with augmented patient data.

Code generation and analysis

Generative AI models have remarkable capabilities to understand and generate code across various programming languages. Your developers can use these capabilities to assist write, refactor, and analyze code.

Common use cases include the following:

- Automated code generation for boilerplate tasks
- Code refactoring and optimization suggestions
- Bug detection and security vulnerability analysis

AWS offers several services to help support this pattern. Amazon Q Developer provides AI-assisted code generation, helping developers write code faster and with fewer errors. Amazon CodeGuru offers intelligent code reviews and performance recommendations. You can integrate these services with AWS Lambda for serverless code execution and testing.

Best practices for code generation and analysis applications include integrating AI-assisted coding tools into existing development environments. Consider implementing strict code review processes to make sure that generated code meets quality standards. Use AI for initial code drafts but rely on human expertise for final implementation decisions.

For more information about building coding assistants, see Use Amazon Q Developer as a coding assistant to increase your productivity.

Multimodal AI applications

Multimodal AI applications combine different types of data inputs and outputs, such as text, images, audio, and video. This pattern uses the ability of advanced generative AI models to understand and generate content across multiple modalities.

Common use cases for multimodal AI applications include the following:

- Visual question-answering systems for customer support or internal documentation
- Intelligent image and video content moderation
- Multimodal data analysis for business intelligence

On AWS, you can use Amazon Bedrock for multimodal foundation models. Additional services include Amazon Rekognition for image and video analysis, Amazon Transcribe for speech-to-text conversion, and Amazon Polly for text-to-speech conversion. These services can enhance the multimodal capabilities of the application.

Best practices for multimodal AI applications include ensuring high-quality, diverse datasets for training and fine-tuning multimodal models. Also, implement robust error handling for different input modalities, and carefully consider the ethical implications and potential biases in multimodal AI systems.

For more information about building multimodal applications, see Document institutional knowledge from voice inputs by using Amazon Bedrock and Amazon Transcribe.

Implementation recommendations

To effectively implement repeatable application patterns for generative AI across your organization, consider the following recommendations:

- Begin with a pattern catalog that documents successful implementations and provides templates, reference architectures, and code samples that accelerate development.
- Establish a centralized pattern governance process to evaluate, approve, and maintain standard patterns in order to promote consistency and quality across the enterprise.
- Prioritize use cases based on business impact, technical feasibility, and alignment with strategic objectives.
- Create a cross-functional pattern development team that includes AI specialists, domain experts, and end users who can make sure that patterns address real business needs.
- Implement a "build once, reuse many times" approach by creating modular, composable components that can be assembled into different application patterns.
- Develop clear evaluation criteria for each pattern type in order to measure effectiveness. Include metrics for business value, user satisfaction, and technical performance.
- Establish a pattern maturity model that classifies patterns as experimental, proven, or enterprise-standard to guide teams in their selection process.
- Create comprehensive documentation that promotes successful adoption for each pattern, including implementation guides, limitations, and best practices.
- Conduct regular pattern reviews to identify opportunities for improvement, consolidation, or retirement as technology and business needs evolve.

Best practices for enterprise generative AI adoption and scaling

Successfully adopting and scaling generative AI across an enterprise requires a strategic balance of organizational structure, standardized processes, and technical capabilities. The following best practices draw from successful implementations across various organizations, providing a framework for effective enterprise-wide adoption.

This section contains the following topics:

Organizational structure and governance

Consider establishing an AI center of excellence and a model governance committee.

AI center of excellence

Establish an AI center of excellence (AI CoE) (AWS blog post) to guide generative AI initiatives across the organization. The AI CoE should offer guidance, best practices, and technical capabilities for building generative AI applications. Through regular engagement with business units, the AI CoE helps identify opportunities for generative AI adoption while maintaining governance and quality standards.

Model governance committee

It is essential to have a dedicated model governance committee that has clear roles and responsibilities. This committee develops evaluation criteria for foundation models, reviews usage requests, and validates compliance with ethical AI principles. Working closely with legal and compliance teams, the committee oversees model performance and risk assessments. This establishes a balanced approach to innovation and responsible AI use.

Standardization and technical excellence

Consider establishing a library of reusable generative AI patterns and tools, and also create an enterprise-wide access framework that democratizes access to AI resources.

Pattern library and tooling

Develop a suite of standardized tools and predefined patterns for common generative AI applications. Create a centralized repository that contains well-documented patterns, code templates, and architecture diagrams. This standardization lowers the barrier to entry, improves consistency across implementations, and accelerates development by providing clear starting points.

Enterprise-wide access framework

Facilitate organization-wide access to advanced tools, such as Amazon Q Business and Amazon Q Developer, through streamlined processes for requesting access, training, and onboarding. This democratization of AI resources empowers teams across departments while maintaining proper security controls and governance.

Process implementation

Implement processes that help you manage the following:

- Services
- Model selection and evaluation
- Performance monitoring and evaluation
- Security and compliance

Service management

Internal service management processes are crucial for organizing and controlling generative AI adoption. These processes should cover technical evaluation of new models, legal reviews, and access requests. By establishing clear workflows and responsibilities, organizations can maintain proper oversight while improving deployment and operations efficiency for generative AI solutions.

Model selection and evaluation

A systematic approach to model selection is critical for balancing capability, cost, and performance. Begin proof-of-concept development with top-tier models to quickly validate business value and gain stakeholder buy-in. After the use case is proven, systematically evaluate smaller models against established performance benchmarks to optimize costs for production. This approach accelerates initial development while promoting cost-effective scaling. This process requires clear communication about performance expectations and careful documentation of required capabilities.

Performance monitoring and optimization

Effective monitoring and optimization require a comprehensive approach to tracking both technical and business metrics. Develop dashboards to track key metrics, such as inference latency, throughput, error rates, and cost per inference. Set up alerts for anomalies or performance degradation. Conduct regular performance reviews to make sure that models continue to meet business needs.

Cost management should be proactive and strategic. Regular review of model usage patterns and compute resource optimization helps maintain efficient operations. Implement cost-allocation tags and budget monitoring to maintain visibility into expenses across different use cases and business units.

Security and compliance

Security and compliance considerations must be embedded throughout the generative AI implementation lifecycle. Develop a comprehensive risk management framework that addresses data privacy, model security, and ethical AI considerations. This framework should align with existing enterprise security policies and address the unique challenges of generative AI applications.

Implement security controls through a layered approach, starting with robust access management and authentication. Make sure that proper network security and data protection measures are in place, including comprehensive audit logging and monitoring capabilities. Establish clear incident response procedures that are specific to generative AI applications.

Implementation recommendations

For successful generative AI adoption across the enterprise, consider the following key recommendations:

- Start with well-defined, limited-scope projects that can demonstrate clear business value.
- Document success metrics and learnings to inform future projects.
- Scale successful implementations gradually, and make sure that proper controls and support mechanisms are in place.
- Maintain focus on continuous improvement through regular reviews of processes and procedures.
- Create comprehensive training programs that cover both technical and responsible AI practices.
- Establish clear mechanisms for knowledge sharing and cross-team collaboration.

Through careful attention to these best practices and recommendations, organizations can build a strong foundation for sustainable generative AI adoption while maintaining security, efficiency, and ethical considerations.

Next steps

Generative AI represents a transformative opportunity for enterprises to innovate, optimize operations, and create new value for customers. However, the journey from experimentation to enterprise-wide adoption requires a strategic approach that balances innovation with security, governance, and operational excellence.

The four-layer framework presented in this strategy—reliable infrastructure, approved foundation models, robust security and governance, and repeatable application patterns—provides organizations with a comprehensive approach to building and scaling their generative AI initiatives. This framework helps enterprises move beyond isolated prototypes to create sustainable, production-ready solutions that deliver measurable business value.

Success in enterprise generative AI adoption requires more than just technical implementation. Organizations must cultivate the right combination of people, processes, and technology. The establishment of an AI center of excellence, implementation of proper governance structures, and development of clear processes for model evaluation and deployment are as crucial as the technical infrastructure itself.

Security and governance cannot be afterthoughts in generative AI implementation. By embedding these considerations from the start, organizations can innovate while prioritizing the protection of sensitive data and compliance with regulatory requirements. The security controls and governance frameworks outlined in this strategy provide a foundation for responsible AI adoption that can evolve with changing requirements and emerging threats.

The application patterns presented in this strategy document demonstrate how organizations can accelerate their generative AI initiatives by using proven approaches. Whether implementing generative AI-powered assistants, RAG chat-based assistants, or intelligent document processing solutions, these patterns provide a starting point for rapid development while ensuring consistency and quality across implementations.

As organizations embark on their generative AI journey, consider taking the following next steps:

1. Assess your current AI and data infrastructure readiness against the framework presented in this strategy.
2. Establish cross-functional teams to oversee generative AI initiatives, including representatives from IT, data science, legal, and business units.

3. Start with small, well-defined proof-of-concept projects that can demonstrate clear business value.
4. Develop a comprehensive generative AI strategy that aligns with your organization's broader business objectives.
5. Invest in ongoing education and training to build generative AI capabilities across your organization.

Through the comprehensive suite of AWS services and with the solid foundation provided by Amazon Bedrock, organizations can confidently move from generative AI experimentation to scalable, enterprise-ready solutions that are designed to be secure, ethical, and effective. As you navigate this transformative journey, AWS remains committed to supporting your success with the tools, expertise, and infrastructure needed to unlock the full potential of generative AI in your enterprise.

Resources

The following resources provide additional context, technical details, and implementation guidance to support your organization's generative AI journey. As AWS continues to innovate with generative AI, you can regularly check these resources for updates and new capabilities.

AWS Prescriptive Guidance

Strategy resources

- Generative AI workload assessment
- Maturity model for adopting generative AI on AWS
- Retrieval Augmented Generation options and architectures on AWS
- Prompt engineering best practices to avoid prompt injection attacks on modern LLMs

RAG and chat-based assistants

- Choosing an AWS vector database for RAG use cases
- Deploy a RAG use case on AWS by using Terraform and Amazon Bedrock
- Develop advanced generative AI chat-based assistants by using RAG and ReAct prompting
- Develop a fully automated chat-based assistant by using Amazon Bedrock agents and knowledge bases

Coding assistants

- Use Amazon Q Developer as a coding assistant to increase your productivity

Document processing

- Document institutional knowledge from voice inputs by using Amazon Bedrock and Amazon Transcribe
- Automatically extract content from PDF files using Amazon Textract

AWS service documentation

- Amazon Bedrock documentation
- Amazon SageMaker AI documentation

Other AWS resources

- Transform your business with generative AI (AWS website)
- Generative AI innovation center (AWSwebsite)
- Securing generative AI: An introduction to the generative AI security scoping matrix (AWS blog post)
- Establishing an AI/ML center of excellence (AWS blog post)
- Architect defense-in-depth security for generative AI applications using the OWASP Top 10 for LLMs (AWS blog post)
- Setting up secure, well-governed machine learning environments on AWS (AWS blog post)
- Machine learning lens (AWS Well-Architected Framework)

Other resources

- OWASP top 10 for large language model applications (OWASP website)

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	—	June 26, 2025

AWS Prescriptive Guidance glossary

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

Numbers

7 Rs

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

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

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

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

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

active-passive migration

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

aggregate function

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

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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

anti-pattern

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

application control

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

application portfolio

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

artificial intelligence (AI)

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

artificial intelligence operations (AIOps)

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

asymmetric encryption

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

atomicity, consistency, isolation, durability (ACID)

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

attribute-based access control (ABAC)

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

authoritative data source

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

Availability Zone

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

AWS Cloud Adoption Framework (AWS CAF)

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

AWS Workload Qualification Framework (AWS WQF)

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

B

bad bot

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

BCP

See business continuity planning.

behavior graph

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

big-endian system

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

binary classification

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

bloom filter

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

blue/green deployment

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

bot

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

botnet

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

branch

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

break-glass access

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

brownfield strategy

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

buffer cache

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

business capability

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

business continuity planning (BCP)

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

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

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

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

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

chaos engineering

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

CI/CD

See continuous integration and continuous delivery.

classification

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

client-side encryption

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

Cloud Center of Excellence (CCoE)

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

cloud computing

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

cloud operating model

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

cloud stages of adoption

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

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

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

CMDB

See configuration management database.

code repository

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

cold cache

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

cold data

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

computer vision (CV)

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

configuration drift

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

configuration management database (CMDB)

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

conformance pack

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

continuous integration and continuous delivery (CI/CD)

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

CV

See computer vision.

D

data at rest

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

data classification

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

data drift

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

data in transit

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

data mesh

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

data minimization

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

data perimeter

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

data preprocessing

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

data provenance

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

data subject

An individual whose data is being collected and processed.

data warehouse

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

database definition language (DDL)

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

database manipulation language (DML)

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

DDL

See database definition language.

deep ensemble

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

deep learning

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

defense-in-depth

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

delegated administrator

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

deployment

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

development environment

See environment.

detective control

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

development value stream mapping (DVSM)

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

digital twin

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

dimension table

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

disaster

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

disaster recovery (DR)

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

DML

See database manipulation language.

domain-driven design

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

DR

See disaster recovery.

drift detection

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

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

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

electronic data interchange (EDI)

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

encryption

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

encryption key

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

endianness

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

endpoint

See service endpoint.

endpoint service

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

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

enterprise resource planning (ERP)

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

envelope encryption

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

environment

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

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

epic

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

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

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

F

fact table

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

fail fast

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

fault isolation boundary

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

feature branch

See branch.

features

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

feature importance

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

feature transformation

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

few-shot prompting

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

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

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

flash-cut migration

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

FM

See foundation model.

foundation model (FM)

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

G

generative AI

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

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

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

Gitflow workflow

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

golden image

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

greenfield strategy

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

guardrail

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

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

H

HA

See high availability.

heterogeneous database migration

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

high availability (HA)

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

historian modernization

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

holdout data

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

homogeneous database migration

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

hot data

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

hotfix

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

hypercare period

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

I

IaC

See infrastructure as code.

identity-based policy

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

idle application

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

IIoT

See industrial Internet of Things.

immutable infrastructure

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

inbound (ingress) VPC

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

incremental migration

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

Industry 4.0

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

infrastructure

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

infrastructure as code (IaC)

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

industrial Internet of Things (IIoT)

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

inspection VPC

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

Internet of Things (IoT)

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

interpretability

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

IoT

See Internet of Things.

IT information library (ITIL)

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

IT service management (ITSM)

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

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

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

landing zone

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

large language model (LLM)

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

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

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

lift and shift

See 7 Rs.

little-endian system

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

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

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

main branch

See branch.

malware

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

managed services

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

manufacturing execution system (MES)

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

MAP

See Migration Acceleration Program.

mechanism

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

member account

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

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

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

microservice

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

microservices architecture

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

Migration Acceleration Program (MAP)

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

migration at scale

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

migration factory

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

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

migration metadata

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

migration pattern

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

Migration Portfolio Assessment (MPA)

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

Migration Readiness Assessment (MRA)

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

migration strategy

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

ML

See machine learning.

modernization

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

modernization readiness assessment

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

monolithic applications (monoliths)

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

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

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

mutable infrastructure

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

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

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

OI

See operations integration.

OLA

See operational-level agreement.

online migration

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

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

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

operational-level agreement (OLA)

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

operational readiness review (ORR)

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

operational technology (OT)

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

operations integration (OI)

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

organization trail

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

organizational change management (OCM)

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

origin access control (OAC)

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

origin access identity (OAI)

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

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

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

P

permissions boundary

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

personally identifiable information (PII)

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

PII

See personally identifiable information.

playbook

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

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

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

polyglot persistence

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

portfolio assessment

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

predicate

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

predicate pushdown

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

preventative control

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

principal

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

privacy by design

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

private hosted zones

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

proactive control

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

product lifecycle management (PLM)

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

production environment

See environment.

programmable logic controller (PLC)

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

prompt chaining

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

pseudonymization

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

publish/subscribe (pub/sub)

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

Q

query plan

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

query plan regression

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

R

RACI matrix

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

RAG

See Retrieval Augmented Generation.

ransomware

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

RASCI matrix

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

RCAC

See row and column access control.

read replica

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

re-architect

See 7 Rs.

recovery point objective (RPO)

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

recovery time objective (RTO)

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

refactor

See 7 Rs.

Region

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

regression

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

rehost

See 7 Rs.

release

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

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

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

resource-based policy

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

responsible, accountable, consulted, informed (RACI) matrix

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

responsive control

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

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotation

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

row and column access control (RCAC)

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

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

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

S

SAML 2.0

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

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

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

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

security by design

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

security control

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

security hardening

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

security information and event management (SIEM) system

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

security response automation

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

server-side encryption

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

service control policy (SCP)

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

service endpoint

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

service-level agreement (SLA)

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

service-level indicator (SLI)

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

service-level objective (SLO)

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

shared responsibility model

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

SIEM

See security information and event management system.

single point of failure (SPOF)

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

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

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

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

SPOF

See single point of failure.

star schema

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

strangler fig pattern

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

subnet

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

supervisory control and data acquisition (SCADA)

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

symmetric encryption

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

synthetic testing

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

system prompt

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

T

tags

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

target variable

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

task list

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

test environment

See environment.

training

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

transit gateway

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

trunk-based workflow

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

trusted access

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

tuning

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

two-pizza team

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

U

uncertainty

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

undifferentiated tasks

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

upper environments

See environment.

V

vacuuming

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

version control

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

VPC peering

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

vulnerability

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

W

warm cache

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

warm data

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

window function

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

workload

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

workstream

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

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

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

Z

zero-day exploit

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

zero-day vulnerability

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

zero-shot prompting

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

zombie application

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