Economics for agentic AI on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Economics for agentic AI on AWS

Table of Contents

Economics for agentic AI on AWS

Hans Schabert and Prasanta Roy, Amazon Web Services

January 2026 (document history)

Organizations adopting AI-driven automation and agentic AI systems need to make informed economic decisions between human labor and intelligent agents. This becomes critical for sustainable cloud operations. This guide helps you evaluate, implement, and optimize the economic trade-offs between human workforce and agentic AI systems on AWS. You can maximize your return on investment (ROI) while maintaining operational excellence.

No system is 100% right. This fundamental principle drives the economic analysis of human and agentic AI systems. Organizations must move beyond simplistic cost comparisons to evaluate total economic impact, risk profiles, decision quality requirements, and long-term strategic value creation.

Customer behavior is shifting dramatically from traditional upfront technology investments to pay-per-outcome models that align costs with business results. This transformation requires new approaches for evaluation, implementation, and optimization of human-agent collaboration.

The path to success follows a clear pattern: start with appropriate jobs, measure everything, and scale what works. Organizations that adopt this approach achieve sustainable competitive advantage through intelligent resource allocation and outcome-focused automation.

Intended audience

This guide is intended for the following:

- Executives (CEOs, CTOs, CFOs) who are making strategic investment decisions
- Enterprise architects who are designing organizational automation strategies
- Financial operations practitioners who are optimizing cloud financial management
- Technology leaders who are evaluating AI implementation approaches
- Business unit leaders who want to understand the ROI of automation
- Procurement professionals who are navigating new AI pricing models

To understand the concepts in this guide, we recommend that you review Foundations of agentic AI on AWS.

Objectives

This guide helps you understand the following:

- How to evaluate jobs for agentic automation potential
- Economic models for comparing human labor costs against agentic AI system investments
- Pay-per-outcome pricing models and their impact on AI project economics
- Measurement techniques for demonstrating ROI and managing risk
- Scaling strategies that transform fixed costs into variable outcomes

About this content series

This guide is part of a series about agentic AI on AWS. For more information and to view the other guides in this series, see Agentic AI on the AWS Prescriptive Guidance website.

Understanding agentic AI economics on AWS

One of the key principles is to determine when to use AI agents and when to use traditional deterministic methods. Organizations must systematically evaluate which jobs warrant agentic automation and which should use traditional automation or continued human operation. This decision requires understanding the relationship between the task characteristics, risk tolerance, and operational approach.

Before deciding to implement agentic AI, you should use the decision framework to understand the economic impact. The decision framework includes the following three key questions:

1. Task assessment – Is this task right for an AI agent?
2. Risk impact assessment – What are the risks involved?
3. Return on investment – Will it be cost-effective?

Task assessment

Tasks with high-complexity, standardized decision rules can benefit from agentic AI approaches. Highly standardized, simple tasks are better served by traditional automation or robotic process automation. Agentic AI systems excel at reasoning, understanding context, or adaptively making decisions, They add value beyond rule-based processing. Successful agentic AI implementations require systems that are capable of learning and adapting.

Consider the following factors when evaluating a task:

- **Complexity** – Degree of reasoning and context understanding required. Tasks requiring contextual understanding, nuanced interpretation, or adaptive responses to changing conditions favor agentic approaches over traditional automation, while purely mechanical or calculational tasks may not require agentic intelligence.
- **Standardization** – Presence of clear patterns and rules. Agentic AI is recommended if the task requires contextual understanding. If no adaptation or learning is needed, consider traditional automation.
- **Volume** – Frequency of task performance. Agentic AI is recommended for autonomous activities. Traditional automation is recommended for high-volume, consistent tasks. However, volume alone doesn't determine approach. Low-volume, high-value decisions might justify agentic assistance for improved decision quality rather than cost reduction.

- **Value** – Business impact per task completion. Consider agentic AI for high-value outcomes that require human-like autonomous capability. Consider traditional automation for repeated, consistent tasks, which can be done in a deterministic manner.

Risk impact assessment

There are currently four agentic AI deployment approaches: fully autonomous, human in the loop, co-pilot, or human-led with agent support. Each has their own risk profile and error tolerance, and they all involve humans in some capacity. The following table describes the risk details of these approaches.

Autonomy level	Risk profile	Error tolerance	Example use cases	Human involvement
Fully autonomous	Low Risk	1-2% acceptable	• Basic data categorization • Document routing • Standard report generation	• Minimal oversight • Periodic audits
Human in the loop	Medium Risk	Below 0.5%	• Draft responses • Content moderation • Initial claims processing	• Regular review • Exception handling • Quality assurance
Co-pilot	High Risk	Near-zero	• Strategic planning input	• Human makes final decisions

			• Risk assessments • Investment decisions	• Agent provides recommendations
Human-led with agent support	Critical Risk	Zero tolerance	• Legal decisions • Medical diagnosis • Regulatory compliance	• Human drives process • Agent provides research or analysis and supporting information only

The following table describes key considerations when choosing between these approaches.

Consideration	**Fully autonomous**	**Human in the loop**	**Co-pilot**	**Human-led**
Cost efficiency	Highest	High	Medium	Low
Scalability	Unlimited	High	Medium	Limited
Processing speed	Fastest	Fast	Medium	Slow
Risk management	Basic	Enhanced	Strong	Strongest
Complexity handling	Simple tasks	Moderately complex tasks	Complex tasks	Critical tasks

This consideration framework helps organizations match autonomy levels to risk profiles, scale operations appropriately, balance efficiency with control, implement proper governance, and optimize resource allocation.

Return on investment

Calculating the return on investment for agentic AI systems begins with a comprehensive cost analysis. Organizations must first calculate their current human costs, including salary, benefits, and workspace expenses, along with process-specific expenses and hidden costs such as training, coverage, and downtime.

For break-even analysis, organizations should consider implementation costs, ongoing operational expenses, and the volume needed to justify investment. It's also important to account for seasonal variations and the learning curve benefits that emerge as systems mature and improve over time.

When evaluating AI agents, organizations should remember that these systems typically have higher upfront costs but lower per-transaction costs compared to human operations. Additionally, AI agents demonstrate improving performance over time and offer better scalability than human teams. This makes them increasingly cost-effective as deployment scales and operational experience accumulates.

Measuring the success and ROI of agentic AI systems

Measuring success in agentic AI system implementation requires a systematic approach. This section provides a clear methodology for evaluation and ongoing optimization that uses your existing analysis rather than starting from scratch.

Step 1: Use your existing foundation

Begin with a comprehensive cost assessment according to the recommendations in the Assessing your current process costs section. This provides an operational baseline for your ROI calculations. As described in the Risk impact assessment section, choose between the four autonomy levels (fully autonomous, human in loop, co-pilot approach, human-led with agent support) in order to determine appropriate measurement criteria and error tolerance thresholds for each process.

Step 2: Set clear success targets

Establish architecture and success targets that emphasize learning-capable systems, as described in the Successful patterns for implementing agentic AI systems section. Focus on continuous improvement rather than static performance. Set ROI timelines by using the break-even analysis methodology demonstrated in Case study: Comparing human and agentic AI costs for recruitment operations. Include clear decision points for terminating non-performing agents.

Step 3: Track key metrics

Monitor financial performance against your established baseline, and track cost savings and strategic value improvements. Measure operational metrics, including error rates within acceptable thresholds for your chosen autonomy level, processing speed improvements, and consistency gains. Focus on strategic indicators that demonstrate learning capability and adaptation over time.

Step 4: Use AgentOps

Apply the continuous learning framework from the Incorporating human feedback into agentic AI systems section to optimize decision-making through systematic human feedback integration. Create real-time learning systems that incorporate human insights for performance enhancement.

Monitor transformation toward outcome-based business models as described in Economic transformation to outcome-based pricing for agentic AI systems on AWS.

Assessing your current human-process costs

Understanding your true process costs is fundamental to making informed decisions about agentic AI system investments. First, you must establish an accurate baseline of what your current processes cost are, including all hidden expenses, failure rates, and opportunity costs. This helps you develop precise ROI calculations and make strategic decisions. This comprehensive cost assessment serves as the critical foundation for evaluating whether agentic AI systems can deliver genuine value as productive companions.

The baseline cost assessment is essential for the following key reasons:

- **ROI accuracy** – Accurate cost baselines support realistic ROI projections that account for the full spectrum of current operational expenses.
- **Agent implementation strategy** – Comprehensive cost understanding helps organizations to identify the most promising processes for initial agentic AI system deployment.
- **Performance measurement** – Established baselines provide the measurement framework for tracking actual and projected benefits from agentic AI implementations.

Organizations must systematically identify and evaluate all cost factors that influence process economics before comparing human and agentic alternatives. This assessment ensures accurate baseline calculations by accounting for both obvious and hidden cost drivers. It places particular emphasis on failure costs, historical failure rates, and missed business opportunities that represent the true total cost of current processes.

This section describes how to collect data in each cost category to establish accurate baseline measurements for your current processes. It discusses information sources and provides examples for the following cost categories:

- Labor costs
- Human performance and consistency costs
- Technology and infrastructure costs
- Lost business opportunity costs
- Risk and defect costs

Labor costs

Extract 24 months of payroll data that includes base salary, overtime, benefits, and training costs. Use your human resources information system (HRIS) to track recruitment expenses and turnover rates. Time-tracking systems reveal actual productivity versus scheduled hours. Performance management platforms show the correlation between skill levels and compensation costs. Calculate fully loaded hourly rates that allocation for management overhead.

The following is an example list of cost drivers for labor.

Cost driver	Business impact
Base compensation	$25–150 per hour fully loaded
Benefits and payroll taxes	25–40% of base salary
Training and development	5–15% of annual labor cost
Management overhead	15–25% of direct labor cost

Human performance and consistency costs

Combine data from project management systems that show task completion variations with attendance systems. This can reveal absenteeism patterns and seasonal changes. Customer service platforms demonstrate individual performance ranges through resolution metrics, and sales customer relationship management (CRM) data can show efficiency variations in deal closure. Quality management systems provide defect rates and process compliance data across teams and locations. Workflow systems capture completion times, approval delays, and exception-handling frequency. Communication analytics reveal coordination overhead through meeting frequency and collaboration patterns.

The following is an example list of cost drivers for human performance and consistency.

Cost driver	Business impact
Productivity fluctuations	20–50% performance range
Absenteeism and coverage	15–25% additional capacity needed

Fatigue and motivation cycles	10–30% productivity variance
Procedure inconsistencies	10–40% efficiency loss
Quality control variations	10–30% of total cost
Coordination overhead	15–25% of operational cost

Technology and infrastructure costs

License management platforms show software costs and utilization rates. Infrastructure monitoring provides uptime data, performance metrics, and maintenance costs. Help desk systems track support overhead and recurring technical issues. Vendor management systems capture total technology relationship costs, including integration expenses and service-level performance.

The following is an example list of cost drivers for technology and infrastructure.

Cost driver	Business impact
Technology systems	$50–500 per user per month
Workspace and equipment	$200–1,000 per employee per month

Lost business opportunity costs

CRM platforms contain lead response times, conversion rates, and lost opportunity documentation. Marketing automation shows follow-up delay impacts on lead conversions. Customer support systems reveal how operational issues affect satisfaction and retention. Competitive analysis provides market response requirements and win or loss data that connects operational performance to revenue outcomes.

The following is an example list of cost drivers for lost business opportunities.

Cost driver	Business impact
Market response delays	Revenue per day of delay

Capacity constraints	Lost business opportunities
Innovation resource allocation	Opportunity cost of routine work
Customer acquisition delays	50–90% lead loss from slow response

Risk and defect costs

Insurance policy documentation shows costs for general liability, professional liability, workers' compensation, and cyber liability coverage. Internal risk assessment reports identify operational vulnerabilities and associated mitigation costs. Defect-tracking systems document product or service failures, including detection costs, replacement expenses, and warranty claims. Asset replacement schedules show equipment failure rates and replacement costs. Safety incident reports track workplace accidents and associated workers' compensation claims. Business continuity plans detail backup system costs and disaster recovery investments.

The following is an example list of cost drivers for risks and defects.

Cost driver	**Business impact**
Insurance costs	1–5% of operational budget
Cost of errors	$50–5,000 per error incident
Human error impact	2–15% of total operational cost
Error rates and rework	1.5–4 times original cost for corrections

Successful patterns for implementing agentic AI systems on AWS

State of Enterprise AI Adoption (ISG 2025 report) reveals that the primary barrier to successful AI implementation is not technical capability but the *learning gap*. This term refers to systems that cannot adapt, remember context, or improve over time. Organizations that implement static AI tools see high failure rates. The following are common characteristics of agentic AI systems that achieve success:

- **Contextual memory** – Systems that retain conversation history and user preferences
- **Feedback integration** – Ability to learn from corrections and improve performance
- **Workflow adaptation** – Automatic adjustment to changing business requirements
- **Continuous improvement** – Measurable enhancement through operational experience

Organizations that achieve successful AI implementations often prioritize the following:

- Using comprehensive partner ecosystems rather than independently building and exploring AI capabilities
- Learning-capable systems over static tools
- Business-outcome focus over technical feature comparison
- Workflow integration rather than standalone tools
- Continuous adaptation rather than one-time implementation

These patterns align with many AWS service capabilities, particularly the foundation model access in Amazon Bedrock, the event-driven architecture in AWS Lambda, and comprehensive monitoring offered through Amazon CloudWatch. For more information about integrating human feedback and learning-capable systems, see the Incorporating human feedback into agentic AI systems section in this guide.

Incorporating human feedback into agentic AI systems

No system is 100% successful, and failure is bound to happen. With every failure, there is an associated cost of change. *Human in the loop* is an AI approach where AI performs a task, but human intervention or approval is required. This approach must be used when the cost of failure is higher than the cost of having a human-in-the-loop solution.

The success of agentic AI systems depends fundamentally on the agent's ability to learn and improve through human feedback. The cost of human effort must be taken into consideration, depending upon the level of effort of required. Unlike static automation tools that execute predetermined rules, human-in-the-loop solutions have learning-capable agentic systems that create a dynamic partnership between the autonomous agents and the human. Human expertise continuously enhances the agent's performance while agents handle routine processing at scale. This collaborative approach transforms AI implementation from a one-time deployment into an ongoing optimization process. The system adapts to organizational patterns, internalizes quality

standards, and refines its decision-making capabilities based on real-world operational experience. By systematically capturing human corrections, approvals, and insights, organizations can build AI agents that understand context, recognize patterns, and align increasingly with business objectives over time.

For solutions that do not require human intervention or support, there is no need to factor human-specific costs into the agent economics.

Behavioral learning from human operators

Human operators provide critical feedback that agentic AI systems can use to learn, adapt, and improve their responses over time. This feedback loop creates a collaborative environment where human expertise enhances agent capability while agents handle routine processing.

Through human behavior pattern recognition, agents learn from human interaction patterns to mirror successful communication approaches. This helps them adapt to organizational decision patterns and risk tolerance levels. Systems internalize quality expectations through human corrections and approvals. They can also learn appropriate responses for different customer segments and business contexts.

Effective feedback collection mechanisms systematically capture human edits and modifications to agent responses. They analyze what human reviewers approve, reject, or modify in agent recommendations. By understanding why certain cases require human intervention and incorporating human evaluation of agent performance across different scenarios and complexity levels, these systems continuously refine their capabilities to align more closely with organizational standards and expectations.

Continuous learning operations

Real-time learning integration enables agentic AI systems to incorporate human feedback and improve agent responses immediately through dynamic model updating. These systems use human insights to identify new patterns and edge cases. This enhances their pattern recognition capabilities while building organizational memory through human-guided learning experiences. Continuous refinement based on human-operator feedback and business outcomes drives ongoing performance optimization.

Human-guided training captures expert knowledge to enhance agent decision-making capabilities. It transfers critical expertise from experienced operators to the AI system. Through scenario-based

learning, systems use human-created examples to improve their handling of complex situations. They also align agent performance standards with human quality expectations through quality calibration. This approach incorporates human insights about organizational culture and customer expectations. This cultural adaptation helps agents respond appropriately across different contexts.

Operational excellence with human-AI collaboration

Automated risk-aware optimization enables continuous evaluation of operating conditions and error probability with human oversight for high-risk scenarios. This helps systems learn from human risk assessments and improve future decision-making. Amazon Bedrock provides access to multiple foundation models with different capabilities and cost profiles. This enables intelligent routing that considers both cost and risk profiles while incorporating human feedback to optimize model selection. Performance tuning balances efficiency with error-rate minimization by incorporating human feedback on quality standards and acceptable performance trade-offs. Automated decisions consider risk-adjusted total cost of ownership. Operators provide guidance about organizational risk tolerance and business priority weighting. This helps you optimize for costs while aligning with organizational objectives.

Human-enhanced learning systems prioritize human input by error impact and business consequences. This creates learning systems that understand both technical accuracy and business context through risk-weighted feedback. Regular performance analysis incorporates risk metrics and error cost analysis, with human insights providing context that automated systems cannot capture. Best practice development emphasizes risk management and error prevention by combining automated pattern recognition with human expertise and judgment. Organizational capability building through training programs develops both human skills for managing agentic AI systems and agent capabilities for supporting human decision-making. This ensures a comprehensive approach to human-AI collaboration that strengthens both components of the partnership.

Economic transformation to outcome-based pricing for agentic AI systems on AWS

The shift from traditional fixed-cost models to outcome-based pricing represents a fundamental transformation in how organizations structure their economic operations and manage risk. This transformation serves as a pathway for constant modernization of existing processes while financing the agentic AI transformation. It enables organizations to evolve from static, resource-intensive operations to dynamic, results-driven business models.

Traditional, upfront model

Departments often operate as cost centers with direct labor costs that are cost-allocation financed. Organizations typically want to reduce this cost allocation. If the process is not modernized, the department must deliver the same outcomes with a smaller workforce. This typically degrades quality. Traditional business models create significant challenges, including:

- **Linear cost scaling with volume increases** – This requires organizations to hire additional staff to handle increased volume.
- **Fixed cost commitments** – These persist regardless of business performance and process efficiency.
- **Advanced planning** – Limited flexibility during economic downturns and capacity constraints requires advance planning.
- **Quality degradation cycle** – Reduced budgets lead to diminished service quality when costs are cut without process improvements.

Outcome-based model

Modern outcome-based models tie payments directly to measurable business results, such as successful hires completed, quality metrics achieved, process efficiency improvements, or productivity gains realized. This fundamentally shifts financial risk from business units to service providers while creating natural incentive alignment. The following are the key benefits of an outcome-based model:

- Costs scale directly with business value generated
- Natural alignment between operational expenses and revenue
- Flexibility to adjust capacity based on market conditions
- Pay-per-success models reduce financial risk by shifting financial exposure from upfront investment to ongoing operational performance
- Focus on learning-capable systems that improve over time, rather than static alternatives

This transformation extends far beyond internal cost centers to fundamentally reshape how organizations engage with external partners and service providers. By applying outcome-based pricing to partner collaborations, organizations can drive long-term quality improvements and reduce costs while indirectly emphasizing agentic AI modernization.

Organizations can experiment rapidly, measure performance clearly, and scale based on actual business value generated rather than traditional fixed resource commitments. This approach enables the following:

- **Vendor relationship evolution** – Partners become invested in customer success rather than just service delivery.
- **Standardized outcome metrics** – Simplify procurement processes across multiple providers.
- **Market responsiveness** – Quickly adapt to changing market conditions and customer needs.
- **Competitive advantage** – Superior resource utilization and enhanced operational capabilities.
- **Quality-driven partnerships** – Long-term collaboration focuses on continuous improvement and measurable results.

Using AWS Marketplace as pay-per-outcome enabler

The key enabler for this transformation is AWS Marketplace, which serves as a transaction vehicle for agentic work and outcome-based pricing. It provides access to hundreds of pre-built AI agents and agentic solutions with transparent, usage-based pricing models. It can help eliminate upfront licensing costs, reduce implementation complexity, and enable organizations to focus on learning-capable systems that adapt and improve over time rather than static alternatives

Using AWS Marketplace can provide the following benefits:

- **Rapid experimentation** – Test multiple solutions without significant capital investment
- **Transparent pricing** – Usage-based costs with clear attribution to business outcomes
- **Proven solutions** – Access to battle-tested agents from experienced providers
- **Built-in integration** – Seamless connectivity with existing AWS services
- **Risk mitigation** – Ability to switch providers based on performance
- **Learning capability access** – Availability of adaptive systems without internal development costs

This approach enables organizations to compare multiple options based on outcome delivery and learning capabilities rather than feature lists. It can also help you establish clear success criteria and measurement methodologies and negotiate outcome-based pricing that is tied to business results and system improvement. By financing agentic AI transformation through outcome-based models, organizations can modernize their processes continuously while only paying for measurable improvements and successful outcomes.

Case study: Comparing human and agentic AI costs for recruitment operations

Recruitment operations provide a compelling case study for evaluating the economic trade-offs between human and agentic AI systems, but the ROI calculation depends critically on your current operational baseline. Organizations evaluating agentic AI investments often ask a fundamental question: "What if we simply optimize our existing human processes instead?" To address this directly, this analysis presents two distinct scenarios that bracket the range of human operational efficiency.

Scenario A models 45-minute curriculum vitae (CV) or resume screening times. Scenario B demonstrates optimized human operations at 15 minutes per application, which is a 66% efficiency improvement. For example, this improvement might be achieved through streamlined processes, experienced recruiters, or specialized tools.

By comparing identical agent system capabilities against these different human performance baselines, we reveal how existing process efficiency impacts ROI calculations, break-even timelines, and strategic implementation decisions. This dual-scenario approach serves multiple purposes. It prevents organizations from dismissing agentic AI by assuming process optimization alone is sufficient. It also helps organizations with already-efficient processes understand their specific economics. In addition, these scenarios highlight when non-financial advantages, such as 24/7 availability and scalability, become primary decision factors. Understanding these economic dynamics across different efficiency baselines enables organizations to make informed decisions about where and when to deploy agentic AI systems for maximum business impact.

Scenario A: 45-minute screening time

Scenario A represents recruitment operations where human recruiters spend 45 minutes screening each resume. This scenario models a mid-level recruiter with an annual fully-loaded cost of $112,250. This recruiter processes applications during standard business hours with typical human performance characteristics. In contrast, the agentic AI system requires an initial investment of $23,000 for development, customization, and ATS integration, and it has a minimal monthly operating costs of $500 for the cloud infrastructure. The agent processes applications in just 5 minutes with 24/7 availability, achieving a 2% error rate and monthly capacity exceeding 8,600 applications. This is a dramatic efficiency gap, where the agent operates 9 times faster per application and 39 times greater monthly capacity. This section examines cost structure analysis,

operational metrics, volume-based comparisons, and cumulative ROI calculations over the first six months of operation.

Base cost structure

The following table shows initial setup costs for scenario A.

Component	Human operations	Agentic AI system
Agent development and customization	N/A	$15,000
Applicant tracking system (ATS) integration	N/A	$5,000
Training and optimization	N/A	$3,000
Total initial setup cost	**$0**	**$23,000**

The following table shows annual fixed costs for scenario A.

Component	Human operations	Agentic AI system
Base salary	$65,000	N/A
Benefits (30%)	$19,500	N/A
Workspace and equipment	$12,000	N/A
Management oversight (15%)	$9,750	N/A
Training and development	$6,000	N/A
Total annual fixed cost	**$112,250**	**N/A**

The following table shows monthly operating costs for scenario A.

Component	Human operations	Agentic AI system

Cloud computing	N/A	$200
Storage	N/A	$100
Database operations	N/A	$100
Monitoring	N/A	$100
Total monthly fixed cost	**$9,354**	**$500**

Operational metrics

The following table shows operational metrics for scenario A.

Metric	Human operations	Agentic AI system
Processing time per application	45 minutes	5 minutes
Hourly capacity	1.33 applications	12 applications
Daily capacity (24 hours)	10-11 applications	288 applications
Monthly capacity	220 applications	8,640 applications
Cost per application	$45	$2.50
Cost per successful hire	$2,200	$125
Error rate	5%	2%
Error correction cost	$90 per error	$45 per escalation

Volume-based cost analysis

The following table shows a volume-based cost analysis for scenario A. In this example, the agentic AI system cost includes fixed costs and amortized setup costs of $1,917 per month over 12 months.

Monthly volume	Human cost	Agentic AI system cost	Monthly savings
100 applications	$4,500	$750	$3,750
500 applications	$22,500	$2,667	$19,833
1,000 applications	$45,000	$4,917	$40,083

ROI analysis

The following table shows an ROI analysis for scenario A that is based on processing 500 applications per month.

Metric	Value
Monthly human cost	$22,500
Monthly agent cost	$2,667
Monthly savings	$19,833
Annual savings	$237,996
Break-even period	1.16 months

Cumulative cost comparison

The following table shows a cumulative cost comparison for scenario A for the first six months, assuming 500 applications per month.

Month	Human cost	Agentic AI system cost	Cumulative savings
1	$22,500	$25,667	-$3,167
2	$45,000	$28,334	$16,666

3	$67,500	$31,001	$36,499
4	$90,000	$33,668	$56,332
5	$112,500	$36,335	$76,165
6	$135,000	$39,002	$95,998

Additional benefits of the agentic AI system

The following are additional benefits provided by the agentic AI system in Scenario A:

- **Scalability** – Can handle volume spikes without additional cost
- **Availability** – 24/7 operation with immediate response
- **Consistency** – Uniform screening criteria application
- **Time efficiency** – Significantly reduced time-to-hire
- **User experience** – Instant feedback to candidates

Scenario B: 15-minute screening time

Scenario B models optimized recruitment operations where human recruiters have streamlined their screening process to 15 minutes per application. This represents a 66% efficiency improvement over Scenario A. This scenario maintains the same fully loaded annual cost of $112,250 for a mid-level recruiter. However, it demonstrates significantly enhanced human productivity, with daily capacity increasing to 32 applications during an 8-hour shift and monthly throughput reaching 660 applications. The improved human efficiency reduces the cost per application from $45 to $15, narrowing the economic gap with the agentic AI system. However, the agent maintains its structural advantages: 5-minute processing time, 24/7 availability enabling 288 daily applications, a lower 2% error rate compared to the human 5%, and monthly capacity exceeding 8,600 applications. While this efficiency improvement extends the break-even period from 1.16 months to 4.76 months and reduces monthly savings from $19,833 to $4,833, the analysis reveals that agent systems remain economically viable even when competing against highly optimized human operations—a critical insight for organizations evaluating whether their current process efficiency levels justify agentic AI investment.

Base cost structure

The following table shows annual fixed costs for scenario B.

Component	Human operations	Agentic AI system
Base Salary	$65,000	N/A
Benefits (30%)	$19,500	N/A
Workspace & Equipment	$12,000	N/A
Management Oversight (15%)	$9,750	N/A
Training & Development	$6,000	N/A
Total Annual Fixed Cost	**$112,250**	**N/A**

The following table shows implementation costs for scenario B.

Component	Human operations	Agentic AI system
Initial setup	N/A	$23,000
Monthly fixed costs	$9,354	$500

Operational metrics

The following table shows operational metrics for scenario B.

Metric	Human operations	Agentic AI system
Processing time per application	15 minutes	5 minutes
Hourly capacity	4 applications	12 applications
Daily capacity (8-hour shift)	32 applications	288 applications

Monthly capacity	660 applications	8,640 applications
Cost per application	$15	$2.50
Cost per successful hire	$2,200	$125
Error rate	5%	2%
Error correction cost	$30 per error	$45 per escalation

Volume-based cost analysis

The following table shows a volume-based cost analysis for scenario B. In this example, the agentic AI system cost includes fixed costs and amortized setup costs of $1,917 per month over 12 months.

Monthly volume	Human cost	Agentic AI system cost	Monthly savings
100 applications	$1,500	$750	$750
500 applications	$7,500	$2,667	$4,833
1,000 applications	$15,000	$4,917	$10,083

ROI analysis

The following table shows an ROI analysis for scenario B that is based on processing 500 applications per month.

Metric	Value
Monthly human cost	$7,500
Monthly agentic AI system cost	$2,667
Monthly savings	$4,833
Annual savings	$57,996

Break-even period	4.76 months

Cumulative cost comparison

The following table shows a cumulative cost comparison for scenario B for the first six months, assuming 500 applications per month.

Month	Human cost	Agentic AI system cost	Cumulative savings
1	$7,500	$25,667	-$18,167
2	$15,000	$28,334	-$13,334
3	$22,500	$31,001	-$8,501
4	$30,000	$33,668	-$3,668
5	$37,500	$36,335	$1,165
6	$45,000	$39,002	$5,998

Comparing costs and benefits for each scenario

Metric	Scenario A	Scenario B	Impact
Screening time	45 minutes	15 minutes	66% improvement
Daily capacity	10–11 applications	32 applications	200% increase
Cost per application	$45	$15	66% reduction
Monthly savings (500 applications)	$19,833	$4,833	76% decrease
Break-even period	1.16 months	4.76 months	310% longer

Scenario B demonstrates significant efficiency gains in human operations, with processing time improvements that increase capacity without additional headcount and reduce cost per application substantially. However, the financial impact reveals a more nuanced picture: while the ROI remains positive, organizations face an extended break-even period and reduced monthly savings compared to Scenario A. These results highlight critical decision factors for implementation—the agent system remains financially viable even against optimized human operations, but organizations must adopt a longer-term investment perspective and carefully consider volume fluctuations and scalability needs when evaluating deployment timelines and expected returns.

However, the agentic AI system still maintains critical operational advantages that extend beyond pure cost savings. It provides 24/7 availability for immediate candidate engagement regardless of time zones or business hours. It delivers consistent screening quality by applying uniform criteria to every application, scales to handle volume spikes without incurring additional costs. It offers immediate candidate response that enhances the employer brand and candidate experience, and it operates with zero fatigue factor that ensures the same high-quality performance on the first application as the thousandth.

Human errors typically result from fatigue, distraction, or knowledge gaps and often involve miscommunication or incorrect information. Agentic AI system errors usually stem from edge cases, ambiguous inputs, or training data limitations. These errors tend to be more consistent in nature.

Quality and experience metrics reveal clear trade-offs between human and agent capabilities:

- **Customer satisfaction** – Humans excel in empathy and complex problem-solving, and agents provide consistent, accurate information for routine queries.
- **Response time** – Response time favors agents with immediate 24/7 availability. Humans provide business-hours support with potential queuing delays.
- **Consistency** – Agents deliver identical responses to similar queries. Humans can vary in approach and knowledge application.
- **Escalation handling** – Complex issues that require judgment, creativity, or emotional intelligence remain human strengths.

Conclusion and resources

The economics of human systems compared to agentic AI systems represents more than a technology decision. It reflects a fundamental transformation in how organizations create value, manage risk, and achieve competitive advantage. Success requires systematic evaluation of job characteristics, comprehensive measurement of outcomes (including risk factors), and strategic scaling based on proven results.

State of Enterprise AI Adoption (ISG 2025 report) reveals that most AI implementations fail due to learning gaps—systems that cannot adapt, remember context, or improve over time. Organizations that achieve success focus on learning-capable systems that integrate deeply into workflows and demonstrate continuous improvement through human feedback and operational experience.

Organizations that understand these principles—starting with appropriate jobs, decomposing jobs into tasks, measuring everything including risk impact, and scaling what works—will achieve sustainable competitive advantage through optimal resource utilization and outcome-focused automation that grows with business success.

The future belongs to organizations that can intelligently combine human expertise with agentic AI capabilities. This creates hybrid models that deliver superior outcomes while maintaining the flexibility, learning capacity, and collaborative benefits required for dynamic market conditions.

Resources

The following resources can help you plan, design, and implement agentic AI systems on AWS:

- Building serverless architectures for agentic AI on AWS (AWS Prescriptive Guidance)
- Operationalizing agentic AI on AWS (AWS Prescriptive Guidance)
- Agentic AI patterns and workflows on AWS (AWS Prescriptive Guidance)
- Agentic AI (AWS Prescriptive Guidance)
- AWS Cost Optimization Hub (AWS service)
- Amazon Bedrock documentation (AWS service)
- Cost optimization pillar (AWS Well-Architected Framework)
- AI agents and solutions (AWS Marketplace)

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	—	January 28, 2026

AWS Prescriptive Guidance glossary

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

Numbers

7 Rs

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

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

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

A

A2A (Agent-to-Agent)

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

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

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

active-passive migration

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

Agent

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

Agent Ops

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

aggregate function

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

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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

anti-pattern

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

application control

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

application portfolio

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

artificial intelligence (AI)

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

artificial intelligence operations (AIOps)

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

asymmetric encryption

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

atomicity, consistency, isolation, durability (ACID)

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

attribute-based access control (ABAC)

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

authoritative data source

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

Availability Zone

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

AWS Cloud Adoption Framework (AWS CAF)

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

AWS Workload Qualification Framework (AWS WQF)

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

B

bad bot

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

BCP

See business continuity planning.

behavior graph

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

big-endian system

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

binary classification

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

bloom filter

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

blue/green deployment

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

bot

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

botnet

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

branch

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

break-glass access

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

brownfield strategy

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

buffer cache

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

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities.

For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

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

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

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

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

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

chaos engineering

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

CI/CD

See continuous integration and continuous delivery.

classification

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

Citizen Developer

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

client-side encryption

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

Cloud Center of Excellence (CCoE)

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

cloud computing

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

cloud operating model

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

cloud stages of adoption

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

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

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

CMDB

See configuration management database.

code repository

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

cold cache

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

cold data

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

computer vision (CV)

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

configuration drift

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

configuration management database (CMDB)

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

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in

an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

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

CV

See computer vision.

D

data at rest

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

data classification

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

data drift

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

data in transit

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

data mesh

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

data minimization

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

data perimeter

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

data preprocessing

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

data provenance

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

data subject

An individual whose data is being collected and processed.

data warehouse

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

database definition language (DDL)

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

database manipulation language (DML)

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

DDL

See database definition language.

deep ensemble

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

deep learning

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

defense-in-depth

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

delegated administrator

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

deployment

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

development environment

See environment.

detective control

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

development value stream mapping (DVSM)

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

digital twin

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

dimension table

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

disaster

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

disaster recovery (DR)

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

DML

See database manipulation language.

domain-driven design

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

DR

See disaster recovery.

drift detection

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

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

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

electronic data interchange (EDI)

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

encryption

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

encryption key

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

endianness

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

endpoint

See service endpoint.

endpoint service

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

enterprise resource planning (ERP)

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

envelope encryption

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

environment

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

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

- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

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

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

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

F

fact table

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

fail fast

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

fault isolation boundary

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

feature branch

See branch.

features

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

feature importance

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

feature transformation

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

few-shot prompting

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

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

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

flash-cut migration

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

FM

See foundation model.

foundation model (FM)

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

FM gateway

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

G

generative AI

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

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

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

Gitflow workflow

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

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision

software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

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

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries. *Detective guardrails* detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub CSPM, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

guardrails (AI)

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

H

HA

See high availability.

heterogeneous database migration

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

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver

high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

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

holdout data

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

human-in-the-loop (HitL)

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

homogeneous database migration

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

hot data

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

hotfix

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

hypercare period

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

I

IaC

See infrastructure as code.

identity-based policy

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

idle application

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

IIoT

See industrial Internet of Things.

immutable infrastructure

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

inbound (ingress) VPC

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

incremental migration

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

Industry 4.0

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

infrastructure

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

infrastructure as code (IaC)

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

industrial Internet of Things (IIoT)

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

inspection VPC

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

Internet of Things (IoT)

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

interpretability

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

IoT

See Internet of Things.

IT information library (ITIL)

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

IT service management (ITSM)

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

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

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

landing zone

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

large language model (LLM)

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

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

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

lift and shift

See 7 Rs.

little-endian system

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

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

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

main branch

See branch.

malware

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

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage

Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

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

MAP

See Migration Acceleration Program.

MCP

See Model Context Protocol.

Model Context Protocol (MCP)

A stateless protocol for agent-to-tool communication.

MCP server

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

mechanism

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

member account

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

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

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

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include

microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

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

Migration Acceleration Program (MAP)

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

migration at scale

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

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

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

migration pattern

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

Migration Portfolio Assessment (MPA)

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

Migration Readiness Assessment (MRA)

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

migration strategy

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

ML

See machine learning.

modernization

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

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and

milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

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

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

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

mutable infrastructure

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

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

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

OI

See operations integration.

OLA

See operational-level agreement.

online migration

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

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

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

operational-level agreement (OLA)

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

operational readiness review (ORR)

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

operational technology (OT)

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

operations integration (OI)

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

organization trail

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

organizational change management (OCM)

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

origin access control (OAC)

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

origin access identity (OAI)

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

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends

setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

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

personally identifiable information (PII)

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

PII

See personally identifiable information.

playbook

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

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

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

polyglot persistence

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

portfolio assessment

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

predicate

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

predicate pushdown

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

preventative control

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

principal

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

privacy by design

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

private hosted zones

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

proactive control

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

product lifecycle management (PLM)

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

production environment

See environment.

programmable logic controller (PLC)

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

prompt chaining

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

pseudonymization

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

publish/subscribe (pub/sub)

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

Q

query plan

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

query plan regression

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

R

RACI matrix

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

RAG

See Retrieval Augmented Generation.

ransomware

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

RASCI matrix

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

RCAC

See row and column access control.

read replica

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

re-architect

See 7 Rs.

recovery point objective (RPO)

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

recovery time objective (RTO)

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

refactor

See 7 Rs.

Region

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

regression

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

rehost

See 7 Rs.

release

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

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

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

resource-based policy

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

responsible, accountable, consulted, informed (RACI) matrix

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

responsive control

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

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotation

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

row and column access control (RCAC)

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

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

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

S

SAML 2.0

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

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata. The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

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

security control

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

security hardening

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

security information and event management (SIEM) system

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

security response automation

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

server-side encryption

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

service control policy (SCP)

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

service endpoint

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

service-level agreement (SLA)

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

service-level indicator (SLI)

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

service-level objective (SLO)

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

shared responsibility model

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

Shadow AI

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

SIEM

See security information and event management system.

single point of failure (SPOF)

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

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

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

strangler fig pattern

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

subnet

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

supervisory control and data acquisition (SCADA)

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

symmetric encryption

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

synthetic testing

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

system prompt

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

T

tags

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

target variable

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

task list

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

test environment

See environment.

training

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

tool

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

transit gateway

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

trunk-based workflow

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

trusted access

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

tuning

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

two-pizza team

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

U

uncertainty

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

undifferentiated tasks

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

upper environments

See environment.

V

vacuuming

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

version control

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

VPC peering

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

vulnerability

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

W

warm cache

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

warm data

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

window function

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

workload

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

workstream

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

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

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

Z

zero-day exploit

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

zero-day vulnerability

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

zero-shot prompting

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

zombie application

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