AWS Prescriptive Guidance

Modernizing operations in the AWS Cloud

AWS Prescriptive Guidance: Modernizing operations in the AWS Cloud

Table of Contents

Modernizing operations in the AWS Cloud

Itai Njanji, Amazon Web Services (AWS)

August 2019 (document history (p. 18))

The process of modernizing operations in the cloud involves readiness, automation, and integration. In a migration project, these activities are commonly referred to as *operations integration (OI)*. This guide describes the OI workstream, and specifically how to incorporate tools, people, and process perspectives into various activities to build a cloud operating model.

Targeted business outcomes

Business outcomes can be categorized as follows:

- **Operations readiness:** Understand how you currently perform IT operations and how you will operate in AWS. One way to achieve this business outcome is to define a cloud operating model. For more information, see the cloud operating model diagram (p. 2) later in this guide.
- **Operations automation:** Invest in automation to deliver an AWS operating model. For more information, see the Using AWS services for automation (p. 3) section later in this guide.
- **Operations integration:** Continue using current IT tools and extend them through integration to AWS. For more information, see the Tools integration (p. 5) section later in this guide.

Summary process

When you start the process, your operations are in a hybrid state. Although the goal is a fully modernized operation, OI makes it possible for you to use tools and processes to meet the production needs of your applications when they are running in a hybrid cloud environment.

The objective of the OI workstream is to review your current operational model and develop an approach that supports the future-state operating models as you migrate to AWS. The operating model should take into account the relationships among people, processes, and tools. As a workstream owner, you identify and document high-level gaps around tools, processes, and people from the end-state operational model perspective. Then you can create an implementation roadmap or prioritization of the work.

Guidelines and steps

The three primary phases are discovery, design, and implementation. This section covers inputs required from other workstreams and outputs expected from each phase.

Discovery is the first phase. It forms the basis of operations readiness. During discovery, you should:

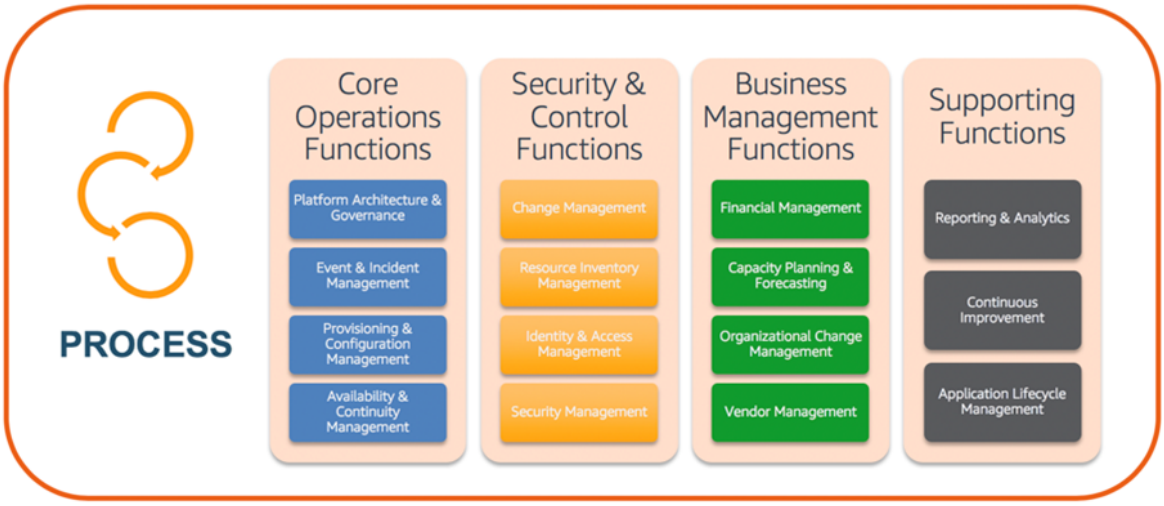
- Hold workshop or kickoff meetings to come up with a prioritized list of operational domains that need to be modernized.
- Gather as much information as possible to get the big picture of your processes and procedures.
- Review documentation, such as playbooks, runbooks, and organizational charts.
- Speak to application owners, support staff, and business owners to create a full picture of pain points.
- Review service-level agreements (SLAs), operational-level agreements (OLAs), and disaster recovery and backup strategies.

The design phase usually begins soon after discovery and before implementation, but you can perform design and implementation in the same sprint. During the design phase, you use the information you gathered during discovery to address gaps and change, improve, or even eliminate processes and procedures.

Because the design and implementation phases form the basis of operational automation and integration, everything must be in place and production-ready before the go-live date. For information about what to account for during the design and implementation phases, see the Best practices and recommendations (p. 3) section.

Perform these phases in a logical manner in one or more sprints. Deliver core operations functions or other prioritized domains. You can capture the work for each phase in a cloud playbook. A playbook can take the form of scripts, automated runbooks, or even a summary of processes or steps required to operate your modernized environment.

The following diagram shows the 15 OI domains organized in 4 functions: core operations, security and control, business management, and supporting functions.



Best practices and recommendations

This section provides best practices related to automation, readiness, tools integration, and testing:

Using AWS services for automation

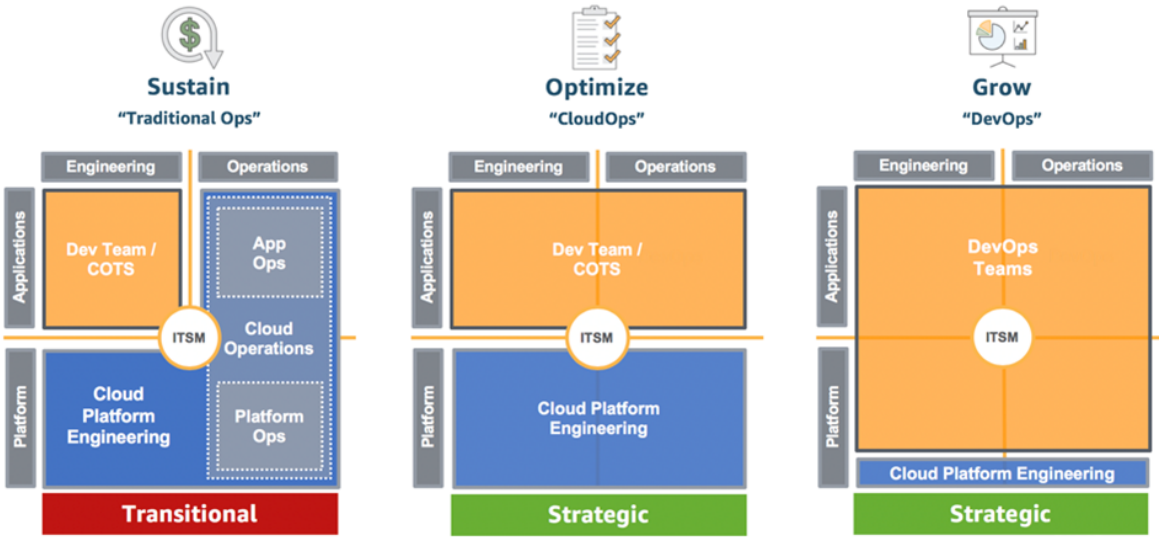
You can use a number of AWS services to automate your IT operations. The following table lists the 15 OI domains and provides information to help you select the right service for each operational need.

Area	Cycle 1 focus and tools
Platform architecture and governance	Governance, guardrails, enterprise architecture and platform design for AWS, tagging, AWS Systems Manager. Usually covered by deploying the AWS Landing Zone solution, AWS Control Tower, or AWS Managed Services.
Event and incident management	Logging and monitoring, service restoration, Amazon CloudWatch, Amazon Simple Notification Service (Amazon SNS).
Provisioning and configuration management	Template consumption, infrastructure as code, AWS Service Catalog, AWS CloudFormation.
Availability and continuity management	Reliability, serviceability, resiliency, Availability Zone failover, volume backup, SLAs for cloud.
IT change management	Compliance and controls, risk management, AWS Service Catalog, AWS Config, AWS CloudTrail.
Resource inventory management	Transparency, resource lifecycle, AWS Config.
Identity and access management	Least privilege, AWS Identity and Access Management (IAM), federation, usually implemented through the security workstream.
Security management	Security controls, security incident response, specified by the security workstream. For example, see the automated patch management guide on the AWS Prescriptive Guidance website.

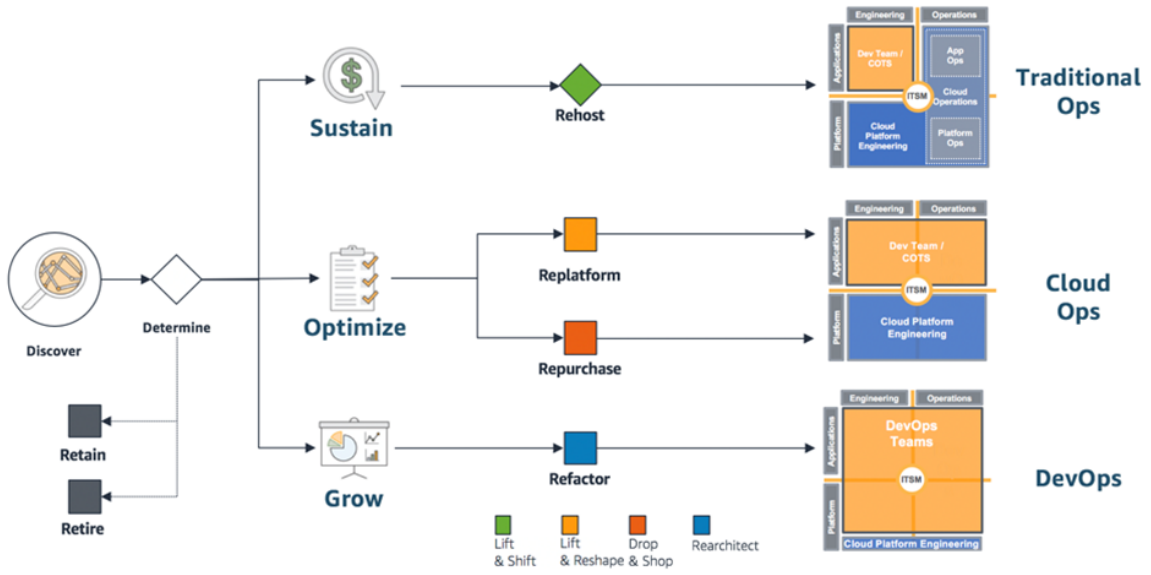
Area	Cycle 1 focus and tools
Financial management	Tagging, billing report, cost optimization (right-sizing, governance), AWS Trusted Advisor.
Capacity planning and forecasting	Fit for purpose designs, resource trends, AWS Config, AWS Trusted Advisor, budget.
Organizational change management	Training, communications, transformational buy-in.
Vendor management	Outsourced provider controls.
Reporting and analytics	Usage trends, service health, EKK (Amazon OpenSearch Service, Amazon Kinesis, and Kibana), Amazon QuickSight.
Continuous improvement	Process iterations, AWS Trusted Advisor, EKK (Amazon OpenSearch Service, Amazon Kinesis, and Kibana), Amazon QuickSight.
Application lifecycle management	Software development lifecycle, people/process/tools integrations, DevOps workstream.

Operations readiness

Workshops are an effective way to understand your current operating model and to define an AWS operating model. The following diagram shows suggested operating models.



The following diagram shows how your operating model might affect your migration path to AWS, based on your desired business outcome.



Operations automation

You are not expected to have all of your IT operations fully automated on day 1. Take a phased approach instead. Start by carefully prioritizing the core operations functions shown in the following diagram. Use the AWS services in the table provided earlier in this guide (p. 3) to streamline the modernization process.

Functions	Capabilities and Considerations
Platform Architecture & Governance	Account Strategy, VPC Strategy, Multi-Region/Multi-AZ Strategy, Tagging Strategy, IP Addressing and Connectivity, Security Strategy, Audit and Compliance
Event & Incident Management	Monitoring, Cloud Metrics, Logging, Analytics, Cloud Runbooks, Alerting/Alarming, Incident Management Process, AWS Support Engagement, Service Desk, Toolsets/integrations
Provisioning & Configuration Management	Provisioning, CICD Pipeline and Toolset, Release Management, Testing Framework and Toolset, Code Repository, Branching Strategy, Blue/Green Deployments, CMDB, Configuration Items
Availability & Continuity Management	High Availability Architecture, Auto-Scaling, Backup and Restore, Replication, RTO/RPO, Data Storage and Retention Policy, Disaster Recovery, Automation and Bots, Hybrid Options – Storage Gateways

Tools integration

Most enterprise customers use IT service management (ITSM) tools to automate their on-premises operations. They need the same tools when they move to the cloud. The alternative—relearning operations without their ITSM tools—is costly and would delay migration. The following table shows the common integration patterns between ITSM tools and AWS.

AWS Landing Zone integration with ITSM

Scenario	Description	AWS Tools
Self-service and self-provisioning	Enterprise users self provision resources in AWS using Service Catalog connector in ITSM without the to need sign into the AWS Management Console. AWS CloudFormation and Terraform are leveraged.	AWS Service Catalog, AWS CloudFormation
CMDB Integration	AWS Config is a cloud native discovery service that monitors CIs in AWS. Using AWS Config, CI states such as provision/termination and compliance states can be logged into CMDB.	AWS Config, Amazon Simple Notification Service
AMS-ITSM Integration	Standardized Delivery Kit to integrate AMS and ITSM using AMS connector App V2, and flexibility to customize integrations for various customer needs.	AWS Managed Services
Ticketing	Bi-directional integration between AWS Enterprise Support API and ITSM. Tickets seamlessly flow between AWS Enterprise Support and ITSM incidents.	AWS Trusted Advisor, AWS Support Ticketing
Monitoring	Amazon CloudWatch alarms are set to monitor cloud environment. When an alarm is triggered, an incident is created in ITSM for remediation or workflow initiation.	Amazon CloudWatch, Amazon Simple Notification Service
Migration Accelerator using ITSM (MAUS)	Customers can use ITSM Discovery modules/CMDB data to define their on-premises landscape to prepare for mass migration projects on AWS. Data of business mapping can be imported into AWS tools such MPA.	AWS Migration Portfolio Assessment, ITSM Business Mapping

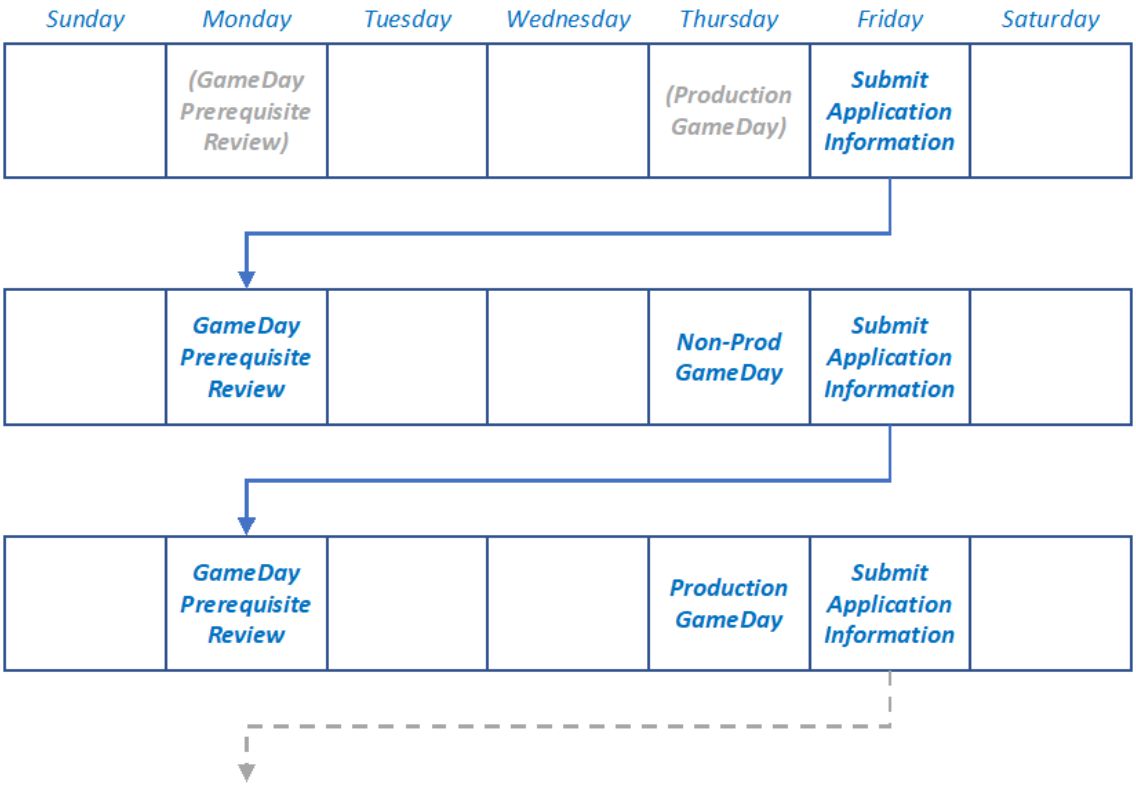
Operations testing

Like products, IT operations should be tested, end to end, on a regular cadence. Although enterprise customers have adopted operational testing for activities like disaster recovery, operational testing should be extended to other operations domains, such as incident and event management. Game-day scenarios, like fire drills, are activities that test how your processes, tools, and people react when an operations event occurs. Here are some prescriptive game-day scenarios used to test incident and event management:

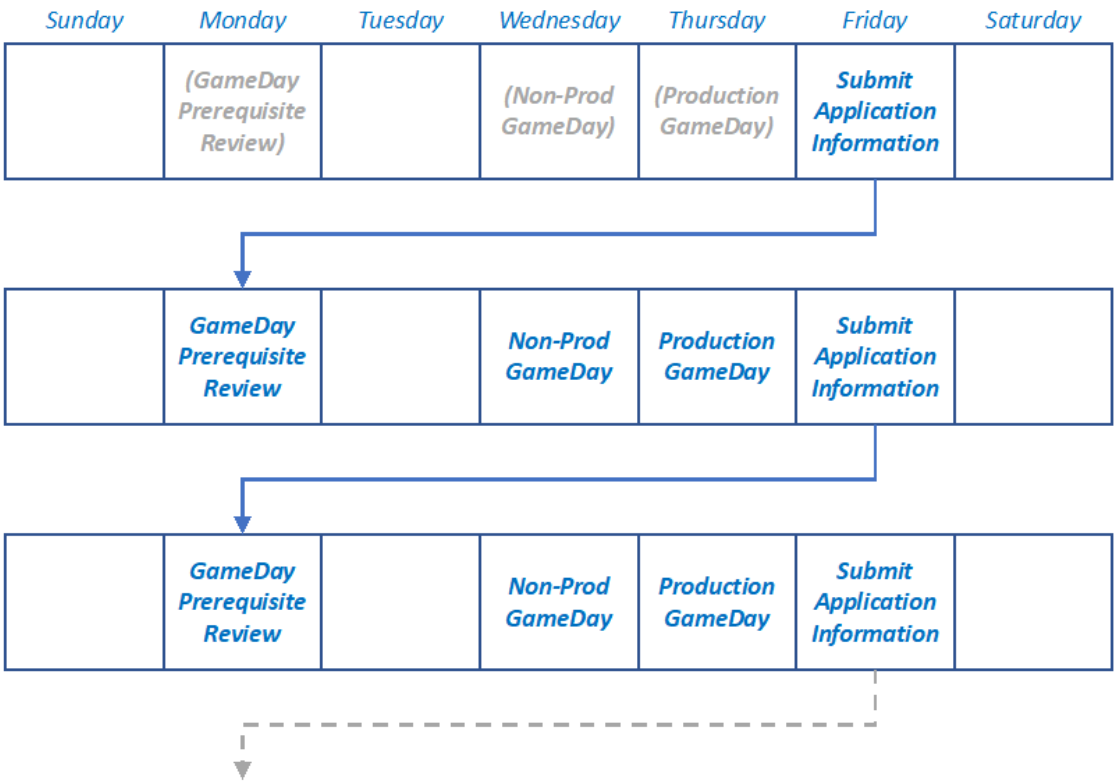
- Amazon Elastic Compute Cloud (Amazon EC2) CPU utilization stress test
- Amazon EC2 network stress test
- Amazon EC2 memory stress test
- Amazon Relational Database Service (Amazon RDS) memory stress test
- Amazon RDS storage stress

As a best practice, you should test your IT operations starting with incident and event management, and test them in other operational domains, too. As a best practice, you should also have a predetermined game-day schedule. Here are some examples.

Prod or non-prod schedule



Prod and non-prod schedule



AIOps

When you migrate your workloads to AWS, you can choose from many AWS services to monitor your infrastructure. Events and alarms in Amazon CloudWatch and rules in AWS Config, for example, can provide valuable insights. You can get even more insights by applying machine learning techniques to this data. Artificial intelligence operations (AIOps) is the process of using machine learning techniques to solve operational problems. The goal of AIOps is to reduce human intervention in the IT operations processes.

By using advanced machine learning techniques, you can reduce operational incidents and increase service quality. AIOps can help you:

- Increase service quality (for example, by grouping related incidents based on time and language or by predicting Knowledge Base articles to solve an incident).
- Predict incidents before they happen.
- Classify new incidents and insights.

Next steps

If you are in the process of modernizing your operations, use this guide and the associated patterns to start your journey. When considering a large-scale migration to the cloud, many organizations invest significant time and resources in planning and assessment, but it may be helpful to start with a targeted application to use as minimal viable product. You'll find it faster and more valuable to establish the foundation services, and then use the technical foundations, business learnings, and deployment processes developed in that project as the basis for larger-scale activities.

Modernizing operations is an iterative process. It evolves over time as you integrate other operations tooling with your cloud initiatives. As the following diagram illustrates, the process includes alignment, launch (cycle 1), scaling, and optimization.

- **Align.** In this phase, you conduct discovery activities to agree on the desired business outcome, educate the team, address blockers, and align technical and business stakeholders on the initiative. This is part of operational readiness and discovery described earlier in this guide.
- **Launch (cycle 1).** In this phase, you launch something small but fully functional into a production-like environment. The goal is to deliver minimum viable product (MVP) capabilities. For example, you can migrate an existing workload or create new functionality.
- **Scale.** In this phase, you expand the limited-scope work from the launch phase and expand it to full scale. For innovation, this means rolling out all functionality to all users. For migration, this means migrating all workloads to AWS.
- **Optimize.** In this phase, you revisit workloads in AWS to identify areas for improvement and implement them. It's during this phase that you might consider AIOps, for example.

The process of modernizing operations in the cloud involves readiness, automation, and integration. This guide described how to achieve readiness by defining a cloud operating model, how to automate your operations using AWS services, and how to integrate your tools using best practices. It also showed you how to use a migration project with a targeted application to start the process of modernizing operations.

AWS Prescriptive Guidance glossary

AI and ML terms (p. 10) | Migration terms (p. 11) | Modernization terms (p. 15)

AI and ML terms

The following are commonly used terms in artificial intelligence (AI) and machine learning (ML)-related strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

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

Migration terms

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

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. This migration scenario is specific to VMware Cloud on AWS, which supports virtual machine (VM) compatibility and workload portability between your on-premises environment and AWS. You can use the VMware Cloud Foundation technologies from your on-premises data centers when you migrate your infrastructure to VMware Cloud on AWS. Example: Relocate the hypervisor hosting your Oracle database to VMware Cloud on 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.
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 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.
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 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.
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.
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.
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 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.
configuration management database (CMDB)	A database that contains information about a company's hardware and software products, configurations, and inter-dependencies. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.
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.
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.
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.
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.
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.
large migration	A migration of 300 or more servers.
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 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 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 CloudEndure 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 strategy	The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs (p. 11) entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.
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).
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.

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.
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.
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.
responsible, accountable, consulted, informed (RACI) matrix	A matrix that defines and assigns roles and responsibilities in a project. For example, you can create a RACI to define security control ownership or to identify roles and responsibilities for specific tasks in a migration project.
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.
service-level agreement (SLA)	An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.
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.
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.
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.

Modernization terms

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

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

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.

update-history-change	update-history-description	update-history-date
Initial publication (p. 18)	—	August 5, 2019