AWS Prescriptive Guidance
Setting up an agile program
to accelerate cloud migrations
# Table of Contents

Introduction ...................................................................................................................................... 1  
Targeted business outcomes ........................................................................................................ 1  
Reviewing current cloud adoption and migration frameworks .................................................. 3  
Adopting an agile approach with epics ......................................................................................... 4  
Approach .................................................................................................................................. 4  
Preparation ............................................................................................................................... 5  
Implementation ......................................................................................................................... 5  
Value-add outcomes ................................................................................................................... 6  
Guidelines for a successful migration ........................................................................................... 7  
Planning ................................................................................................................................... 7  
Activities ................................................................................................................................... 7  
Next steps ................................................................................................................................... 8  
AWS Prescriptive Guidance glossary .......................................................................................... 9  
Document history ............................................................................................................................. 17
Setting up an agile program to accelerate cloud migrations

Vijay Thumma, Global Practice Manager, AWS Professional Services

August 2019 (last update (p. 17): August 2021)

Cloud adoption is more than just a technology transformation. To move to the cloud effectively, organizations can use the AWS Cloud Adoption Framework (AWS CAF) and its six perspectives (business, people, governance, platform, security, and operations) as a framework to ensure a comprehensive view of the transformation initiative.

This framework can help you understand which areas to improve before moving forward with a migration effort for enterprises. You can combine this framework with agile practices and streamlined cloud migrations to move quickly and innovate faster. We recommend embracing an agile approach for large migrations to the cloud—not just as a discrete activity.

Targeted business outcomes

This guide discusses how you can take an agile approach to accelerating cloud migrations at scale. There are two primary outcomes expected from this guide:

- **Steps to set up an agile program with epics for defined workstreams in the readiness and planning phase** (see the following diagram). This program builds the core components of the cloud operational foundation to align, enable, and mobilize the workforce and resources for a successful migration.
- **Guidelines for migration planning and implementation using sprints.** These guidelines help you to discover, transform, and deploy complex sets of workload components.

The prescriptive model that we describe in this guide is built on direct customer engagement experiences in the field. Your adoption strategy should be closely aligned with your business goals and objectives. Your decision process for migrating resources to AWS depends on the migration method (for example, rehost, refactor, replatform). This guide leads you through the implementation process, and helps you align with other processes and launch functions in the readiness and planning phase.

The following diagram describes the various phases of the overall migration process, including assessment, readiness and planning, and migrations. It also includes details about the workstreams that are part of the readiness and planning phase.
AWS Prescriptive Guidance Setting up an agile program to accelerate cloud migrations
Targeted business outcomes
Reviewing current cloud adoption and migration frameworks

When organizations contemplate migrating to AWS at scale, they are somewhere along the path of what is called the stages of adoption, as shown in the following figure. The four stages are project, foundation, migration, and reinvention.

- **Project** – The organization evaluates the AWS Cloud as a viable option and determines whether it addresses their specific needs. AWS skills aren’t required at this stage.
- **Foundation** – The organization starts extending their data centers to AWS. Specifically, the organization:
  - Establishes scalable security and compliance models on AWS.
  - With AWS migration specialists, deploys the initial framework of the landing zone in a non-production environment.
  - Sets up cloud-focused and AWS skills training for employees.
  - Moves 3-5 enterprise-grade production applications to AWS.
- **Migration** – At this stage, the organization:
  - Prepares for long-term cloud operations.
  - Defines IT roles for cloud operations.
  - Establishes a Cloud Center of Excellence (CCoE), which is a multi-disciplinary team that drives cloud adoption efforts across an organization.
  - Migrates groups of production applications and data centers.
- **Reinvention** – At this stage, the cloud becomes the default choice for all projects. The organization explores new, cloud-based business models, and becomes more comfortable with cloud operations than it is with on-premises operations.

The purpose of a readiness assessment is to understand where along the stages of adoption the organization is, and which parts of the environment are sufficiently mature to move to the next stage. You complete the readiness assessment before you start the readiness and planning phase of migration. In the agile program described in this guide, you review the results of the readiness assessment to understand your current state.
Adopting an agile approach with epics

This section describes the agile approach for cloud migrations, including preparation, implementation, and value-add outcomes.

- Approach (p. 4)
- Preparation (p. 5)
- Implementation (p. 5)
- Value-add outcomes (p. 6)

Approach

Using an agile approach with epics (large stories), you start small, iterate, measure, manage, and scale. You can use an agile methodology for the preparation and implementation stages of large migrations to the cloud. With this approach, you do the following:

- Structure your work in the form of epics and stories to be able to respond to change
- Produce a well-prioritized backlog
- Report your progress
- Build a migration roadmap to efficiently and effectively tie business stakeholder needs to technology initiatives

The following diagram illustrates the agile process that we describe in this guide.
Preparation

The preparation stage of the migration consists of these steps:

1. Set up two 2-pizza teams as scrum teams. These teams are made up of internal resources from workstreams that are defined in the readiness and planning phase. Group the teams based on the underlying functional/technical roles. Together these teams are responsible for driving adoption, enabling initial migrations, and preparing the organization for running enterprise-scale migrations.
   - **2-Pizza Team 1** structure and resources – advisory, business case, program governance, people skills, and center of excellence (CoE).
   - **2-Pizza Team 2** structure and resources – app discovery/migrations, landing zone, security, and operations integration.

2. Kick off a planning meeting with both scrum teams to review the results from the migration readiness assessment.

3. Identify 10 to 30 applications to migrate from on premises to AWS in Wave 1.

4. Set up a backlog. Prioritize the use of “pre-baked” epics for all workstreams from existing migration patterns. Here are a few examples:

5. Assign a scrum leader and a product owner, who are responsible for managing the backlog.

6. Set up eight two-week sprints for migrating applications.

7. Build a migration plan with resources, a backlog (epics, user stories), a risk/mitigation log, and a roles and responsibilities matrix (for example, a RACI matrix). You can use this plan to manage the risks that occur during the project, and to identify ownership for each resource involved.

Implementation

The implementation stage of the migration consists of these steps:

1. Set up sprint schedules and objectives for each of the eight sprints with both teams.

2. Review epics, break them into stories with acceptance criteria, and organize them into Sprints 1 through 8.

3. Set up a daily cadence meeting between both 2-pizza teams. This meeting is important for uninterrupted workflow and faster feedback. The scrum leader organizes daily status calls through sprints to discuss progress, resolve blockers, and report on deliverables.

4. Build a communications and reporting plan for implementation oversight meetings. Use agile metrics to prove that teams are healthy and delivering value.
5. Manage sprints with a primary focus on the following:
   • Learn by doing—align, enable, and mobilize a workforce and resources.
   • Start small, create early successes, and build to full migration capacity.
6. Capture outputs, best practices, and lessons learned to use as building blocks for migrations at scale.
7. Create blueprints of agile delivery for scaling migrations based on Wave 1. Present the approach as a foundation for the readiness and planning migration phase.

Working through these initial Wave 1 migrations helps to build confidence and experience. This process includes the migration plan with the patterns and tool choices that fit your organization’s needs. It provides validation and testing of the operational and security processes. It also identifies patterns (for example, common architectures and technology stacks) in the portfolio to create a list of application groupings based on common patterns. This creates a common process for group migrations.

Value-add outcomes

Using this agile model, you can accelerate the path to achieving business outcomes and value drivers for operating at scale. Value-add business outcomes that are outlined in this agile approach include the following (see the agile migration process (p. 4) illustrated earlier in this guide).

- Application selection – Analyze application based on the 7 Rs of migration.
- Migration wave plan and migration tools – Develop a prioritized list of move groups with available tools.
- Landing zone – Validate various components of your landing zone via deployment and testing.
- DevOps – Deliver features, fixes, and updates frequently, using the DevOps model.
- Automation – Use vendor tools for automated migrations.
- Migration patterns – Use the available patterns in the AWS Prescriptive Guidance catalog.
- Design architecture – Design target state architectures for server, data, and application migrations.
- Training tools – Train resources on AWS services, and provide hands-on migration experience.
- Operational runbook – Validate the operating model by deploying, monitoring, and reporting on applications that are running on AWS (see Modernizing operations in the AWS Cloud).
- Governance – Manage migration scope, schedule, resource plan, issues and risks, and communications.
Guidelines for a successful migration

This section includes guidelines for planning your migration and for migration sprint activities.

Planning

Before migrating resources to AWS, you need to choose the migration method, tools (if applicable), and operating model that you’re using to manage, govern, and secure your environment. As part of planning for the migration, do the following:

• Create a comprehensive wave plan with key assumptions. Identify dependencies, tools, the number of sprints, and ownership of the sprints.
• Build the account structure, access, security, network, connectivity, automation, and pipelines.
• Have the landing zone ready.
• Select and obtain the migration tools.
• Define the change control and impact assessment processes.
• Define the communication plan.
• Define the production operations and operating model.
• Select and commit the resources, including partners.

Activities

Plan activities for all sprint weeks. Everyone on the team must participate!

• Create key metrics to guide the team and monitor progress. Identify key results for each objective.
• Conduct a pre-mortem to understand and plan for risks.
• Establish a schedule and select tools to keep everyone informed.
• Hold an inception retrospective to identify and improve any sprint activities.
• Reprioritize the product backlog.
• Create a day roadmap for practices to continue.
• Hold a retrospective to improve.
• Don’t deviate from the schedule.

Teams should also establish a shorter cadence for sprints, planning, and retrospective meetings. We suggest one-week or two-week sprints to provide quick feedback, to allow for pivoting, and to create a sense of urgency.

In the Wave 1 migrations, you developed a blueprint and building blocks for the migration process, and tested specific migration patterns. Now you’re ready to scale teams to support your initial wave of migrations to build the migration factory process, and expand on the agile team model for your entire application portfolio.

You can use the migration factory process illustrated earlier in this guide (p. 2) to pave the way for the larger move to AWS. This is a process that can efficiently handle the migration workload by combining the technical components of a cloud migration with the business and people components.
Next steps

For large migrations to the cloud, it's critical to introduce agile program steps for both the preparation and implementation stages. Analyzing your current state, building a plan, and iterating the work breaks a large migration into manageable activities for efficient implementation. Use the AWS Cloud Adoption Framework (AWS CAF) to analyze your environment through the different perspectives: business, people, governance, platform, security, and operations. This framework gives you a complete view of which areas to improve before moving forward with a large migration effort.

Use the agile program to kick-start your initial wave of migrations with 10 to 30 applications. Then, build upon it, use the migration factory process, and iterate the migration patterns to create an optimal move to the AWS Cloud.
## AWS Prescriptive Guidance glossary

### 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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>binary classification</strong></td>
<td>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?”</td>
</tr>
<tr>
<td><strong>classification</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>data preprocessing</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>deep ensemble</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>deep learning</strong></td>
<td>An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.</td>
</tr>
<tr>
<td><strong>exploratory data analysis (EDA)</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>features</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>feature importance</strong></td>
<td>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 <a href="https://aws.amazon.com/blogs/machine-learning/machine-learning-model-interpretability-amazon-gluon-ts/">Machine learning model interpretability with AWS</a>.</td>
</tr>
</tbody>
</table>
### 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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Migration terms</strong></td>
<td></td>
</tr>
<tr>
<td><strong>7 Rs</strong></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
</tr>
</tbody>
</table>

---
• 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. 10) 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)
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](https://aws.amazon.com/about-aws/sustainability/guidelines/).

<table>
<thead>
<tr>
<th>update-history-change</th>
<th>update-history-description</th>
<th>update-history-date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added details (p. 17)</td>
<td>Provided more information about the stages of adoption.</td>
<td>August 30, 2019</td>
</tr>
<tr>
<td>Initial publication (p. 17)</td>
<td>—</td>
<td>August 5, 2019</td>
</tr>
</tbody>
</table>