Options, tools, and best practices for migrating Microsoft workloads to AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Options, tools, and best practices for migrating Microsoft workloads to AWS

Table of Contents

Options, tools, and best practices for migrating Microsoft workloads to AWS

Jerroll Harewood, Christine Megit, Dror Helper, Daniel Maldonado, Phil Ekins, Mani Pachnanda, Siddharth Mehta, Rich Benoit, Rob Higareda, Saleha Haider, Siavash Irani, and Yogi Barot, Amazon Web Services (AWS)

June 2023 (document history)

Overview

Organizations have been migrating and running their Microsoft workloads on AWS for over a decade—longer than any other cloud provider. Based on the knowledge and expertise that AWS has gained from migration and modernization efforts over the years, this guide is designed to streamline the migration of your Microsoft workloads to the AWS Cloud. You can use this guide to plan and implement all phases of your Windows migration. This guide is applicable to a variety of migration use cases, including the following:

- You're starting a Windows migration as part of a digital transformation and modernization journey in your organization.
- The lease on the data center where you run your Microsoft workloads is nearing expiration.
- You have a variety of Windows applications with varying availability requirements, but you don't have the resources to deploy your workloads across geographically distributed locations.

In this guide, you learn about a variety of AWS tools that can help streamline your migration journey, such as AWS Migration Hub, AWS Application Migration Service, and more. To align with AWS best practices, this guide follows the three-phase AWS migration process: assess, mobilize, and migrate and modernize. This process is based on a time-tested migration framework that can help you structure and streamline your Windows migration. In the assess phase, you evaluate your readiness for operating in the cloud. In the mobilize phase, you draft migration plans and close readiness gaps identified in the assess phase. Then, you start to migrate your workloads in the migrate and modernize phase by using a combination of automation tools and templates to systematically migrate your workloads and meet your business requirements.

Intended audience

This guide is intended for IT architects, migration leads, technical leads, AWS Partner teams, and other roles responsible for the following:

- Migrating Microsoft workloads from a data center to the AWS Cloud
- Managing a Windows environment in the AWS Cloud

Targeted business outcomes

This guide can help you and your organization achieve the following objectives:

1. Learn about the strategies, programs, and services available for migrating Microsoft workloads to AWS.
2. Understand the AWS migration paths for specific Microsoft workloads, such as Active Directory, Windows File Server, SQL Server, and .NET workloads.
3. Run your Microsoft workloads on AWS while meeting your security, availability, and reliability requirements.
4. Familiarize yourself with licensing best practices for running Microsoft workloads on AWS.

Why choose AWS for Microsoft workloads?

AWS has been helping customers migrate and modernize their Microsoft workloads for over 14 years and has the broadest portfolio of services, programs, and expertise to accelerate the transformation of key applications that power businesses. If you use AWS to migrate and modernize, you can look forward to the following benefits:

- **Unlock innovation –** Moving from a traditional monolithic architecture to a cloud-based microservices architecture can give you the freedom to adapt and experiment quickly so that your organization can unlock innovation faster. AWS has the broadest set of container technologies, including Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), and AWS Fargate. Additionally, AWS has the most mature serverless offering (AWS Lambda), deeply integrated .NET support, DevOps utilities to automate development cycles, several open-source integrations, and purpose-built databases such as Amazon Aurora to power modern architectures.
- **Reduce costs –** You can avoid paying for expensive Windows or SQL Server licensing by moving to open-source database solutions. For example, Aurora provides the same functionality as commercial databases at one-tenth the cost. If you move to DevOps and use containers and serverless solutions, you can reduce your total cost of ownership (TCO) and maximize compute consumption.
- **Improve security –** AWS offers 230 security, compliance, and governance services and key features—five times more services than the next largest cloud provider. You can use AWS Directory Service, also known as AWS Managed Microsoft AD, to improve your cloud security and eliminate the need to synchronize or replicate data from your existing Active Directory during migrations. You can also use AWS Identity Services to manage identities and permissions at scale, while providing flexible options for where and how you manage your employee, partner, and customer information.
- **Develop skills with trusted experts –** AWS has unmatched experience helping millions of organizations reach their migration goals faster through unique tools and services. The AWS Migration Acceleration Program (MAP) for Windows provides best practices, tools, and incentives to reduce the complexity and cost of migrating to the cloud with support from AWS Partners and AWS Professional Services. The End-of-Support Migration Program for Windows Server can help you migrate legacy Windows Server applications to the latest supported versions of Windows Server on AWS. 90 percent of Fortune 100 companies and the majority of Fortune 500 companies use AWS Partner solutions and services.

- **Improve the price and performance of your processing power –** AWS is a leader in processing innovation, offering Graviton2-based instances that are 20 percent less expensive per hour than Intel x86-based instances, with up to 40 percent better performance. Aurora also brings five times the throughput of standard MySQL and three times the throughput of standard PostgreSQL. This performance is on par with commercial databases, at one-tenth the cost.
- **Take advantage of flexible licensing options –** AWS offers the most options in the cloud for using new and existing Microsoft software licenses on AWS. If you purchase license-included Amazon Elastic Compute Cloud (Amazon EC2) or Amazon Relational Database Service (Amazon RDS) instances, you get new, fully compliant SQL Server licenses from AWS. You can bring your existing licenses to AWS with Amazon EC2 Dedicated Hosts, Amazon EC2 Dedicated Instances, or EC2 instances with default tenancy by using Microsoft License Mobility through Software Assurance. AWS License Manager makes it easier to track the usage of software licenses and reduce the risk of non-compliance.

For more information, see Windows on AWS in the AWS documentation.

Foundational best practices

Establishing a scalable and secure foundation for your AWS migration can enable you to easily manage and efficiently run your Windows environment on AWS. Before you migrate your Microsoft workloads to AWS, we recommend that you consider the following foundational best practices:

- **Optimize your spending on Microsoft licensing –** Licensing is a critical factor in your cloud migration because it impacts all other decisions moving forward. We recommend that you understand licensing options as early as possible. For more information about licensing, see the Microsoft licensing on AWS section of this guide.
- **Streamline your cloud architecture –** The AWS Well-Architected Framework helps you run your workloads reliably in the cloud. You receive guidance and strategies to help you follow the framework, avoid serious issues, and scale to meet your organization's needs. This guidance also covers billing, access control, and security controls.
- **Build an integrated, easy-to-manage cloud network –** AWS Transit Gateway can help you more easily manage networks and prevent overlapping networks—for example, Classless Inter-Domain Routing (CIDR) range planning—from being created with your on-premises or other cloud environments. That way, you can route traffic to each network as needed. You must determine how accounts route to each other and to on-premises environments and the internet. This enables you to set up proper controls to protect your network traffic. For example, you must decide to make the AWS accounts an extension of existing on-premises data centers and use their perimeter defenses, such as firewalls, intrusion detection systems (IDS) and intrusion prevention systems (IPS), or set up an AWS network account encompassing these perimeter defenses to protect your AWS resources.
- **Prioritize cloud security –** We recommend moving from a single-account to a multi-account environment while adhering to the security best practice of applying least-privilege permissions. We also recommend that you have a thorough understanding of the AWS shared responsibility model and plan how you can secure your environment while maintaining your organization's agility. To improve and maintain security, you can use Amazon API Gateway, AWS WAF, Application Load Balancers, Amazon CloudWatch, AWS CloudTrail, Amazon GuardDuty, and other services. To learn more about multi-account strategy, see Transitioning to multiple AWS accounts in the AWS Prescriptive Guidance documentation.
- **Manage shared IT services in the cloud –** To efficiently manage workloads in the cloud, it's critical to identify all shared services used by your workloads and plan how they will be provided in the cloud. For example, these include Active Directory, file servers, SQL databases, Domain

Name System (DNS), virtual private network (VPN), Simple Mail Transfer Protocol (SMTP), backup, and monitoring services. After you take an inventory, you can decide between extending existing services to the cloud, setting up a completely new instance of the service, or using an alternative managed cloud service. Subsequent sections of this guide will cover these considerations in more detail.

Paths to the cloud

This section describes a high-level approach for implementing best practices to migrate your Windows applications to AWS. Details of these migration strategies and steps are described in the subsequent sections of this guide.

Migration strategies

A migration strategy is the approach used to migrate a workload to the AWS Cloud. There are seven migration strategies for moving applications to the cloud. These strategies are known as the 7 Rs and build upon the 7 Rs that Gartner identified in 2019.

- **Rehost (lift and shift) –** Move an application to the cloud without making any changes to take advantage of cloud capabilities.
- **Relocate (hypervisor-level lift and shift) –** Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations.
- **Replatform (lift and reshape) –** Move an application to the cloud and introduce some level of optimization to take advantage of cloud capabilities.
- **Repurchase (drop and shop) –** Switch to a different product, typically by moving from a traditional license to a software as a service (SaaS) model.
- **Refactor/re-architect –** Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability.
- **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.

Main transformations

The following main transformations take place when you modernize legacy Windows applications and databases:

- **Rehost –** The first step is moving your on-premises infrastructure to cloud infrastructure. This strategy is often referred to as "lift and shift" or rehosting. Rehosting means migrating existing applications and databases to a cloud server instance. There is no need for code changes and you're responsible for managing the instance configuration, software image, and other resources.

- **Replatform –** After you migrate to a cloud environment, the next transformation is around replatforming the applications and databases into a more automated and managed environment. From an application perspective, that means moving from virtual machines (VMs) to containers. Containerizing applications can help you develop, maintain, and deploy applications faster and improve portability. AWS has tools, like AWS App2Container, to help automate the process of containerizing legacy applications. On the database side, moving from a self-service model to a managed database service, like Amazon RDS for SQL Server, eliminates the need for provisioning, patching, and backups. This ultimately frees up resources for activities that can add more value to your organization.
- **Refactor/re-architect –** The third area of transformation is to move from commercial software licensing to open-source options. Many traditional commercial software vendors have built their businesses around software license agreements that are aimed at locking in customers and using punitive licensing terms to force upgrades and migrations. Often, commercial software license fees typically add 20-50 percent of cost on top of equivalent open-source options. We recommend refactoring your applications and databases to take advantage of open-source options so that you can reduce costs, improve performance, and gain access to the latest innovations.

You can complete these main areas of transformation progressively in stages or all at once depending on your application and overall readiness to modernize.

Choosing a migration strategy

The migration strategy to choose depends on the business and IT goals of your organization. Some of the most common business drivers are reducing cost, reducing risk, improving efficiency, addressing skill gaps, and speeding up innovation. We recommend that you evaluate which drivers are important for you, and then choose a migration strategy based on your drivers by using the following guidance. Also, remember that all three approaches are possible roads on your cloud modernization journey, depending on your priorities during each phase of the journey.

When to rehost

Rehosting (or lift and shift) is typically faster and easier because you don't need to make code or architecture changes in the application. Rehosting also minimizes risks and disruption to the business. The operations team can continue to run the business as usual because the application isn't changed. This is especially true for migrations at scale where even a small change becomes significant because of the large number of workloads involved. However, it's important to consider that rehosting doesn't take full advantage of cloud benefits. For example, if you migrate an application with an existing platform issue, that issue will remain after the migration. Finally, it's worth considering that the total cost of ownership (TCO) and return on investment (ROI) for rehosting is lower compared with the other migration approaches.

When to replatform/re-architect

Replatforming is generally more cost-effective than rehosting. You can use replatforming to enhance automation and enable your applications to better use cloud capabilities such as auto-scaling, monitoring, and performing backups. Replatforming reduces operational overhead for the cloud operations team and minimizes risks from pre-existing platform issues. However, replatforming takes longer than a rehosting migration. Also, replatforming requires additional skills to configure the automation that performs code changes on the application and to operationalize the new platform.

When to refactor

A refactor is generally the most cost-effective migration approach. Refactoring is a cloud-native approach that enables applications to rapidly adapt to new requirements by decoupling application components to improve on application resiliency. However, refactoring requires more advanced coding and automation skills. Refactoring also takes longer to implement because it involves rebuilding applications.

Windows migration process

Migrating an existing Windows environment to AWS requires careful planning and implementation. The process involves identifying your current usage of resources, assessing the cost savings potential of migrating to AWS, determining your security needs, and building a well-defined cloud architecture that meets all your organization's requirements. You can use AWS to migrate your current Windows server infrastructure quickly and easily, reducing operational costs while maximizing system efficiency. AWS also offers a range of powerful tools and services to help you maintain control over the entire process and to make sure that your Windows environment in the cloud is optimally configured for maximum performance.

This section provides an overview of the three-phase migration process that AWS developed to assist organizations in the successful migration of several applications to the AWS Cloud: assess, mobilize, and migrate and modernize.

Assess

The assess phase helps you understand the state of your organization’s readiness to move to the cloud. You can use AWS tools to assist you in the assess phase by assessing your on-premises computing resources and building a cost projection for running applications on AWS. We recommend that you consider the following tools:

- Use the migration readiness assessment to understand where you are in the cloud journey.
- Use the AWS Optimization and Licensing Assessment (AWS OLA) to assess and optimize current on-premises and cloud environments, based on actual resource utilization, third-party licensing, and application dependencies.
- Use the Migration Evaluator to help you build a data-driven business case for migration to AWS.
- Use the Cloud Economics Center to build a business case for your migration by defining your objectives, such as improved reliability, cost optimization, and scalability.
- Use AWS Migration Hub to collect server and application inventory data for the assessment, planning, and tracking of your migration.
- Use the Migration Validator Toolkit PowerShell module to discover your Microsoft workloads and migrate them to AWS.

Mobilize

During the mobilize phase, you develop a migration plan and iterate on your business plan and address any gaps in your readiness that were revealed in the assess phase. It's critical to focus on building your baseline environment, driving operational readiness, and developing cloud skills. Migrating a large application portfolio can be a complex task. To ease this process, AWS provides a range of tools and services to help you migrate a set of pilot workloads to the cloud quickly, securely, and cost effectively. Gathering data on your application portfolio and rationalizing applications using one or more of the seven common migration strategies—rehost, relocate, replatform, repurchase, refactor/re-architect, retain, and retire—can provide an improved basis for decision-making. AWS offers a suite of services that you can use to migrate Windows-based applications and workloads to the cloud, including the following:

- AWS Application Discovery Service
- AWS Application Migration Service
- AWS Database Migration Service
- AWS Migration Competency Partners
- Management and Governance on AWS
- AWS Control Tower

Migrate and modernize

In the migrate and modernize phase, you must carefully design, migrate, and validate each application that's in scope for migration. Application Migration Service makes it easy to migrate large numbers of servers from physical, virtual, or cloud infrastructure to AWS. With Application Migration Service, you can use the same automated process for a wide range of applications and quickly lift and shift them from an existing environment to the cloud.

The Cloud Migration Factory on AWS solution is designed to coordinate and automate manual processes for large-scale migrations involving a substantial number of servers. This solution helps you improve performance and prevents long cutover windows by providing an orchestration platform for migrating workloads to AWS at scale. AWS Professional Services, AWS Partners, and other enterprises have already used this solution to help customers migrate thousands of servers to the AWS Cloud.

After the migrations are complete, you can use AWS Migration Hub Refactor Spaces to reduce undifferentiated work when refactoring your application for AWS. Refactor Spaces provides an easy-to-use workspace that enables developers to incrementally refactor existing applications into a modern architecture with minimal overhead or disruption. You can use Refactor Spaces to quickly take advantage of the full range of AWS services optimized for your application.

Your teams are experts in building and running Microsoft workloads on premises. That experience can be enhanced in the cloud. Migrating to AWS can provide an even more efficient and reliable experience for the Windows world you've come to rely on. With AWS, you'll get access to a broad range of cloud services that are designed to make it easier and faster to migrate your existing Microsoft workloads. You can benefit from more scalable capacity, improved storage options, and enhanced security.

Windows environment discovery

With today’s available technologies, such as Application Migration Service, moving Windows Server, Linux, and other x86-based operating systems and their workloads to AWS is fairly straightforward. Getting those workloads to work properly and doing it at scale, however, presents a different set of challenges. This section is intended to identify migration considerations that can enable you to quickly, securely, and smoothly migrate your Microsoft workloads.

Assess

Although you can "brute force" smaller migrations (such as those involving 100 servers) with minimal planning and automation, you can't move 500 or more servers by using this methodology. The following considerations are major contributors to a successful large-scale migration, and you can use the Migration Readiness Assessment (MRA) to identify areas of consideration that you want to focus on.

Enterprise architecture

The more technology debt there is in the environment the more difficult it is to migrate. Organizations that have healthy enterprise architecture programs strive to limit their environment to current and recent versions of software and systems (often called N and N -1 versions of major releases). This not only reduces the number of scenarios that you must account for, but it also takes advantage of the advances of newer releases. For example, Windows Server 2012, Windows Server 2008, and older versions of Windows Server are progressively much more difficult to automate in the Windows Server environment than more current versions. Licensing is also more difficult for older and unsupported versions.

Standardization and configuration management

Standardization of the environment is another factor to consider. Organizations that have environments that are built by hand and maintained are considered to be more like pets. Each system is unique and there are far more possible configuration combinations than if they were built using standardized images, infrastructure as code (IaC), or continuous integration and continuous delivery (CI/CD) pipelines.

For example, it's a best practice to rebuild a typical web server using IaC or CI/CD when migrating, as opposed to manually migrating the individual server. It's also a best practice to store all persistent data in a datastore such as a database, file share, or repository. If systems aren't rebuilt using IaC or CI/CD, they should at least use configuration management tools (such as Puppet, Chef, or Ansible) to standardize the servers they have.

Good data

Good data is also a key factor for successful migrations. Accurate data regarding current servers and their metadata is essential for automation and planning. Lack of good data increases the difficulty when planning a migration. Examples of good data include an accurate inventory of servers, applications on the servers, software on the servers with versions, the number of CPUs, amount of memory, and number of disks. We recommend that you capture any data that wave planners need for planning or any data that you plan to use as part of automating the migration process.

Automation

Automation is essential for migrations at scale. Examples of automation include installing the agent, updating software versions of utilities needed for automation such as .NET or PowerShell, loading or updating software for AWS such as the AWS Systems Manager Agent (SSM Agent), Amazon CloudWatch agent, or other backup or management software needed to run in AWS.

Detailed planning

Developing and managing a detailed plan is also essential for migrations at scale. You must have a well-defined plan in place to migrate 50 servers a week for many weeks. An effective plan includes the following:

- Use **wave planning** to organize servers into waves according to your dependencies and priorities.
- Use **weekly planning** (leading up to cutover) to communicate with application teams and identify network, DNS, firewall, and other details that must be addressed during cutover.
- Use detailed, **hour-to-hour planning** (around actual cutover) to describe the cutover maintenance window.
- Use **go/no-go criteria** to describe under what circumstances an application will either be considered cut over to AWS or must be failed back to the source location.
- Use **cleanup activities** as follow-up activities that must be completed. These activities can happen outside the cutover maintenance window or after the completion of hypercare. Clean-

up activities include verifying backups and various agents, removing the Application Migration Service agent from a server, or removing the source server and associated resources.

Mobilize

During the mobilize phase, it's important to discover as many of your organization's complexities and variations as possible so that they can be accounted for during migration planning. Ideally, you can avoid dealing with such complexities and variations during the cutover maintenance window and prevent any failbacks.

Challenges of migrations at scale

Migration failures occur when an application or applications are cut over to their new environments and performance or functional requirements can't be met within the migration maintenance window. This forces the application or applications to fail back to their original location. In addition, all other applications that are dependent on that application or applications also need to fail back. Failed migrations tend to impact not only the current wave but future waves as applications must be rescheduled.

Latency-sensitive dependencies

A major reason for failed migrations is latency-sensitive dependencies. Failing to identify dependencies that are latency sensitive can introduce performance issues that result in unacceptable response times or transaction times. For example, typically an application moves its database and application servers to the cloud at the same time because they communicate with each other frequently and need the sub-millisecond response time they have when both are in the same data center. Moving only the database to the cloud is likely to introduce many seconds of latency into those transactions, resulting in significant performance impact to the application. This also applies to applications that are heavily dependent on one another and must be in the same data center to perform adequately.

Understanding and addressing application dependencies is therefore of primary importance when planning migrations. Applications and services that are dependent on one another must be identified so that they can be migrated together.

IT shared services

After a workload is in the cloud, it needs a variety of services to function and be maintained properly and securely. This includes a landing zone, network and security perimeter, authentication, patching, security scanners, IT service management tools, backups, bastion hosts, and other resources. Without these services, workloads might not operate properly and will be forced to fail back to their original location.

Configuration updates

In most cases, you must make several configuration changes for a workload to function properly after that workload is moved to the cloud. These configuration changes are often associated with the following dependencies of the workload:

- Firewall rules
- Allow lists
- DNS records
- Connection strings

If you don't make the proper configuration updates, then the workload, its users, and its dependent systems may fail to communicate with each other. Resolving these issues within the outage window could be possible, but changes at this time can be time consuming or require change records that can't be satisfied in time.

Application functional testing

Another challenge for migrations at scale is the need for application functional testing. This is of particular importance since many organizations rely on application teams to identify latency-sensitive dependencies, IT shared services, or needed configuration updates. Ideally, an application team provides a written or automated test plan that they can run during the cutover maintenance window to validate that their application is fully functional with acceptable performance. To keep the cutover maintenance window to a minimum, the test should be able to be completed within 30 minutes.

Tools for application dependency discovery

Determining dependencies between applications is critical for successful migrations—both for detecting latency-sensitive dependencies and connectivity configuration items. There are several tools available in the marketplace for discovering dependencies, such as Application Discovery Service (agent and agentless tool) and Cloudamize (agent-based tool).

When you choose a tool for application dependency discovery, consider the following:

- **Duration** – We recommended that you run discovery tools long enough to capture application-specific events such as known peaks, month end, and other events. The recommended minimum is 30 days.
- **Active (agent based)** – Active dependency discovery tools are often embedded in the kernel of the operating system and capture all transactions. However, this is typically the most expensive and time-consuming method.
- **Passive (agentless)** – Passive dependency discovery tools are much cheaper and faster to implement but risk missing some lesser used connections.
- **Institutional knowledge** – Although application discovery tools provide more detailed and accurate information, most organizations rely on their application teams and their institutional knowledge to discover application dependencies. Application teams are often knowledgeable about latency-sensitive dependencies, but it's not uncommon for them to miss some details such as connectivity configuration settings, firewall rules, or allow list requirements from a partner. You can use institutional knowledge to enhance your application dependency discovery, but we recommend that you also consider and mitigate the risks involved. For example, there is a risk of missing connectivity configuration items or latency-sensitive dependencies if you only depend on the knowledge of your application teams. This could result in outages or failed migrations. To mitigate this risk, we recommend that you conduct detailed application functional testing.

Migrating Microsoft workloads

This section covers prescriptive guidance for specific Microsoft workloads. All the following workload-specific approaches adhere to the assess, mobilize, and migrate and modernize framework.

Migrating Active Directory

Active Directory is a typical identity and access management solution for many corporate environments. The coupling of DNS, user, and machine management makes Active Directory an ideal choice for both Microsoft and Linux workloads for centralized user authentication. When you're planning your journey to the cloud or to AWS, you're faced with the choice of extending Active Directory into AWS or using a managed service to offload the management of the directory service infrastructure. We recommend that you understand the risks and benefits of each option when deciding the right approach for your organization.

The right strategy for an Active Directory migration is one that fits your organization's needs and enables you to take advantage of the AWS Cloud. This involves taking into consideration not only the directory services themselves but how they interact with other AWS services. Additionally, you must consider the long-term goals for the teams that manage Active Directory.

In addition to the Active Directory migration, you must decide the account structure for where Active Directory will be located, the network topology of your AWS accounts, and what DNS integrations and other potential AWS services you plan to use that require Active Directory. For information about designing your account topology and other migration strategy considerations, see the Foundational best practices section of this guide.

Assess

To implement a successful migration, it’s important to asses your existing infrastructure and understand the key features required for your environment. We recommend that you review the following areas before choosing how to migrate:

- **Review existing AWS infrastructure design** – Follow the guidance in the Windows environment discovery section of this guide and use the assessment methods to help review the existing Active Directory infrastructure if you’re not already aware of its footprint and infrastructure requirements. We recommend that you use the prescribed sizing from Microsoft for Active Directory infrastructure in AWS. If you're extending your Active Directory infrastructure to AWS, you may require only a partial amount of your Active Directory authentication footprint in AWS. For this reason, avoid oversizing your environment unless you're completely moving your Active Directory footprint to AWS. For more information, see Capacity planning for Active Directory Domain Services in the Microsoft documentation.

- **Review existing on-premises Active Directory design** – Review the current utilization of your on-premises (self-managed) Active Directory. If you're extending your Active Directory environment to AWS, then we recommend running Active Directory on multiple domain controllers in AWS even as an extension to your on-premises environment. This adheres to the AWS Well-Architected Framework of designing for potential failures by deploying instances in multiple Availability Zones.
- **Identify dependencies in applications and networking** – Before choosing what migration strategy is best, you must fully understand all the features of Active Directory that your organization requires for functionality. This means that when choosing between a managed service or self-hosting it's important to understand the options for each. Consider the following items when deciding which migration is right for you:
  - **Requirements for access** – The requirements for access to control Active Directory will stipulate the right migration path for you. If you require full access to the Active Directory domain controllers to install any type of agents for compliance regulations, then AWS Managed Microsoft AD might not be the right solution for you. Instead, investigate an extension of Active Directory from your domain controllers to Amazon EC2 within your AWS accounts.
  - **Migration timelines** – If you have an extended timeline for migration that doesn't have clear dates for completion, verify that you have contingencies in place for administration of instances in the cloud and in on-premises environments. Authentication is a key component to be in place for Microsoft workloads to avoid administration issues. We recommend that you plan move Active Directory early in your migration.
- **Backup strategies –** If you use an existing Windows backup for capturing the systems state of Active Directory domain controllers, then you can continue to use your existing backup strategies in AWS. Additionally, AWS offers technology options to help you back up your instances. For example, AWS Data Lifecycle Manager, AWS Backup, and AWS Elastic Disaster Recovery are supported technologies for backing up Active Directory domain controllers. To avoid issues, it's best to not rely on restoration of Active Directory. The recommended best practice is to build a resilient architecture, but it's critical to have a backup method in place if recovery is required.

- **Disaster recovery (DR) needs –** If you're migrating Active Directory to AWS you must design for resiliency in the event of a disaster. If you're moving your existing active directory to AWS, you can use a secondary AWS Region and connect the two Regions by using Transit Gateway to allow replication to occur. This is typically the preferred method. There are some organizations that have various requirements for testing failover in an isolated environment, where you sever connectivity between the primary and secondary site for days to test reliability. If this is a requirement in your organization, it could take time to clean up split-brain issues from Active Directory. You might be able to use AWS Elastic Disaster Recovery as an active/passive implementation where you leave your DR site as a failover environment and must routinely test your DR strategy in isolation. Planning for your organization's recovery time objective (RTO) and recovery point objective (RPO) requirements is an important factor while assessing your migration to AWS. Be sure you have your requirements defined along with a testing and failover plan to validate the implementation.

Mobilize

The proper strategy to meet your organizational and operational needs is an important element in migrating or extending Active Directory to AWS. Choosing how you'll integrate with AWS services is critical for adopting AWS. Be sure to choose the method extension of Active Directory or AWS Managed Microsoft AD that meets your business requirements. There are some features in services like Amazon RDS that are dependent on using AWS Managed Microsoft AD. Be sure you evaluate AWS service limitations to determine if there are compatibility constraints for Active Directory on Amazon EC2 and AWS Managed Microsoft AD. We recommend that you consider the following integration points as part of your planning process.

Consider the following reasons for using Active Directory in AWS:

- Enable AWS applications to work with Active Directory
- Use Active Directory to log in to the AWS Management Console

Enable AWS applications to work with Active Directory

You can enable multiple AWS applications and services such as AWS Client VPN, AWS Management Console, AWS IAM Identity Center (successor to AWS Single Sign-On), Amazon Chime, Amazon Connect, Amazon FSx for Windows File Server, Amazon QuickSight, Amazon RDS for SQL Server (only applicable for Directory Service), Amazon WorkDocs, Amazon WorkMail, and Amazon

WorkSpaces to use your AWS Managed Microsoft AD directory. When you enable an AWS application or service in your directory, your users can access the application or service with their Active Directory credentials. You can use familiar Active Directory administration tools to apply Active Directory group policy objects (GPOs) to centrally manage your Amazon EC2 for Windows or Linux instances by joining your instances to your AWS Managed Microsoft AD directory.

Your users can sign in to your instances with their Active Directory credentials. This eliminates the need to use individual instance credentials or distribute private key (PEM) files. This makes it easier for you to instantly grant or revoke access to users by using Active Directory user administration tools that you already use.

Use Active Directory to log in to the AWS Management Console

AWS Managed Microsoft AD enables you to grant members of your directory access to the AWS Management Console. By default, your directory members don't have access to any AWS resources. You assign AWS Identity and Access Management (IAM) roles to your directory members to give them access to the various AWS services and resources. The IAM role defines the services, resources, and level of access that your directory members have.

For example, you can enable your users to sign in to the AWS Management Console with their Active Directory credentials. To do this, you enable the AWS Management Console as an application in your directory, and then assign your Active Directory users and groups to IAM roles. When your users sign in to the AWS Management Console, they assume an IAM role to manage AWS resources. This makes it easy for you to grant your users access to the AWS Management Console without needing to configure and manage a separate SAML infrastructure. For more information, see How AWS IAM Identity Center Active Directory sync enhances AWS application experiences in the AWS Security Blog. You can grant access to user accounts in your directory or in your on-premises Active Directory. This enables users to sign in to the AWS Management Console or through the AWS Command Line Interface (AWS CLI) by using their existing credentials and permissions to manage AWS resources by assigning IAM roles directly to the existing user accounts.

Before you can grant console access to your directory members, your directory must have an access URL. For more information about how to view directory details and get your access URL, see View directory information in the AWS Directory Service Administration Guide. For more information about how to create an access URL, see Creating an access URL in the AWS Directory Service Administration Guide. For more information about how to create and assign IAM roles to your directory members, see Grant users and groups access to AWS resources in the AWS Directory Service Administration Guide.

Consider the following migration options for Active Directory:

- Extend Active Directory
- Migrate to AWS Managed Microsoft AD
- Use a trust to connect Active Directory with AWS Managed Microsoft AD
- Integrate Active Directory DNS with Amazon Route 53

Extend Active Directory

If you already have an Active Directory infrastructure and want to use it when migrating Active Directory-aware workloads to the AWS Cloud, AWS Managed Microsoft AD can help. You can use trusts to connect AWS Managed Microsoft AD to your existing Active Directory. This means your users can access Active Directory-aware and AWS applications with their on-premises Active Directory credentials, without needing you to synchronize users, groups, or passwords. For example, your users can sign in to the AWS Management Console and WorkSpaces by using their existing Active Directory user names and passwords. Also, when you use Active Directory-aware applications such as SharePoint with AWS Managed Microsoft AD, your logged-in Windows users can access these applications without needing to enter credentials again.

In addition to using a trust, you can extend Active Directory by deploying Active Directory to run on EC2 instances in AWS. You can do so on your own or work with AWS to help you with the process. We recommend that you deploy at least two domain controllers in different Availability Zones when extending your Active Directory to AWS. You might need to deploy more than two domain controllers based on the number of users and computers you have in AWS, but the minimum number that we recommend is two for resiliency reasons. You can also migrate your on-premises Active Directory domain to AWS to be free of the operational burden of your Active Directory infrastructure by using the Active Directory Migration Toolkit (ADMT) and the Password Export Server (PES) to perform the migration. You can also use the Active Directory Launch Wizard to deploy Active Directory on AWS.

Migrate to AWS Managed Microsoft AD

You can apply two mechanisms for using Active Directory in AWS. One method is to adopt AWS Managed Microsoft AD to migrate your Active Directory objects to AWS. This includes users, computers, group policies, and more. The second mechanism is a manual approach where you export all users and objects, and then manually import users and objects by using the Active Directory Migration Tool.

There are additional reasons to move to AWS Managed Microsoft Active Directory:

- AWS Managed Microsoft AD is an actual Microsoft Active Directory domain that enables you to run traditional Active Directory-aware workloads such as Microsoft Remote Desktop Licensing Manager, Microsoft SharePoint, and Microsoft SQL Server Always On in the AWS Cloud.
- AWS Managed Microsoft AD helps you to simplify and improve the security of Active Directory-integrated .NET applications by using group Managed Service Accounts (gMSAs) and Kerberos constrained delegation (KCD). For more information, see Simplify Migration and Improve Security of Active Directory–Integrated .NET Applications by Using AWS Microsoft AD in the AWS documentation.

You can share AWS Managed Microsoft AD across multiple AWS accounts. This enables you to manage AWS services, such as Amazon EC2, without the need to operate a directory for each account and each Amazon Virtual Private Cloud (Amazon VPC). You can use your directory from any AWS account and from any Amazon VPC within an AWS Region. This capability makes it easier and more cost effective to manage directory-aware workloads with a single directory across accounts and VPCs. For example, you can now easily manage your Microsoft workloads deployed in EC2 instances across multiple accounts and Amazon VPCs by using a single AWS Managed Microsoft AD directory. When you share your AWS Managed Microsoft AD directory with another AWS account, you can use the Amazon EC2 console or AWS Systems Manager to seamlessly join your instances from any Amazon VPC within the account and AWS Region.

You can quickly deploy your directory-aware workloads on EC2 instances by eliminating the need to manually join your instances to a domain or to deploy directories in each account and Amazon VPC. For more information, see Share your directory in the AWS Directory Service Administration Guide. Keep in mind that there is a cost to share an AWS Managed Microsoft AD environment. You can communicate with the AWS Managed Microsoft AD environment from other networks or accounts by using an Amazon VPC peer or Transit Gateway peer, so sharing might not be needed. If you intend to use the directory with the following services, then you must share the domain: Amazon Aurora MySQL, Amazon Aurora PostgreSQL, Amazon FSx, Amazon RDS for MariaDB, Amazon RDS for MySQL, Amazon RDS for Oracle, Amazon RDS for PostgreSQL, and Amazon RDS for SQL Server.

Use a trust with AWS Managed Microsoft AD

To grant users from an existing directory access to AWS resources, you can use a trust with your AWS Managed Microsoft AD implementation. It’s also possible to create trusts between AWS

Managed Microsoft AD environments. For more information, see the Everything you wanted to know about trusts with AWS Managed Microsoft AD post in the AWS Security Blog.

Integrate Active Directory DNS with Amazon Route 53

When you migrate to AWS, you can integrate DNS into your environment by using Route 53 resolvers to allow access to your servers (by using their DNS names). We recommend that you use Route 53 resolver endpoints to accomplish this rather than modifying DHCP option sets. This is a more centralized approach for managing your DNS configuration than modifying DHCP options sets. Additionally, you can take advantage of a variety of resolver rules. For more information, see the Integrating your Directory Service’s DNS resolution with Amazon Route 53 Resolvers post in the Networking & Content Delivery Blog and Set up DNS resolution for hybrid networks in a multi-account AWS environment in the AWS Prescriptive Guidance documentation.

Migrate

As you begin your migration to AWS, we recommend that you consider configuration and tooling options to help you migrate. It’s also important to consider long-term security and operational aspects of your environment.

Consider the following options:

- Cloud-native security
- Tools to migrate Active Directory to AWS

Cloud-native security

- **Security group configurations for Active Directory controllers –** If you’re using AWS Managed Microsoft AD, the domain controllers come with a VPC security configuration for limited access to the domain controllers. It might be necessary for you to modify the security group rules to allow access for some potential use cases. For more information on security group configuration, see Enhance your AWS Managed Microsoft AD network security configuration in the AWS Directory Service Administration Guide. We recommend that you don't allow users to modify these groups or use them for any other AWS services. Allowing other users to use these could cause service interruptions to your Active Directory environment if the users modify them to block necessary communications.
- **Integrate with Amazon CloudWatch Logs for Active Directory event logs –** If you’re running AWS Managed Microsoft AD or using a self-managed Active Directory, then you can take

advantage of Amazon CloudWatch Logs to centralize your Active Directory logging. You can use CloudWatch logs to copy authentication, security, and other logs to CloudWatch. This gives you an easy way to search logs in one place, and it can help to satisfy some compliance requirements. We recommend integration with CloudWatch Logs because it can help you better respond to future incidents in your environment. For more information, see Enabling Amazon CloudWatch Logs for AWS Managed Active Directory in the AWS Directory Service Administration Guide and Amazon CloudWatch Logs for Windows Event Logs in the AWS Knowledge Center.

Tools to migrate Active Directory to AWS

We recommend that you use the Active Directory Migration Tool (ADMT) and Password Export Server (PES) to perform your migration. This enables you to easily move users and computers from one domain to another. Keep in mind the following considerations if you use PES or migrate from one managed Active Directory domain to another:

- **Active Directory Migration Tool (ADMT) for users, groups, and computers –** You can use ADMT to migrate users from self-managed Active Directory to AWS Managed Microsoft AD. An important consideration is the migration timeline and the importance of Security Identifier (SID) History. SID History is not transferred over during the migration. If supporting SID History is a critical need, then consider using self-managed Active Directory on Amazon EC2 instead of ADMT so that you can maintain SID History.
- **Password Export Server (PES) –** PES can be used to migrate passwords into but not out of AWS Managed Microsoft AD. For information on how to migrate users and passwords from your directory, see How to migrate your on-premises domain to AWS Managed Microsoft AD using ADMT in the AWS Security Blog and Password Export Server version 3.1 (x64) from the Microsoft documentation.
- **LDIF –** LDAP Data Interchange Format (LDIF) is a file format used to extend the schema of an AWS Managed Microsoft AD directory. LDIF files contain the necessary information to add new objects and attributes to the directory. Files must meet the LDAP standards for syntax and must contain valid object definitions for each object the files add. After you create the LDIF file, you must upload the file to the directory to extend its schema. For more information about using LDIF files to extend the schema of an AWS Managed Microsoft AD directory, see Extending the schema of AWS Managed AD in the AWS Directory Service Administration Guide.
- **CSVDE –** In some cases, you might need to export and import users to a directory without creating a trust and using ADMT. Although not ideal, you can use Csvde (a command-line tool) to migrate Active Directory users from one domain to another. To use Csvde, you must create a CSV

file that contains the user information, such as user names, passwords, and group membership. Then, you can use the csvde command to import the users into the new domain. You can also use this command to export existing users from the source domain. This may be helpful if you're migrating from another directory source, such as SAMBA Domain Services to Microsoft Active Directory. For more information, see How to Migrate Your Microsoft Active Directory Users to Simple AD or AWS Managed Microsoft AD in the AWS Security Blog.

Additional resources

- Everything you wanted to know about trusts with AWS Managed Microsoft AD (AWS Security Blog)
- How to migrate your on-premises domain to AWS Managed Microsoft AD using ADMT (AWS Security Blog)
- Active Directory on AWS Immersion Day (AWS Workshop Studio)

Migrating Windows Server

This section focuses on the different options available for migrating Windows Server to AWS.

Assess

First, identify the applications and workloads that need to be migrated to AWS. You can use AWS Application Discovery Service to create a map of your on-premises infrastructure and dependencies between applications. This helps you identify the servers, applications, and services that you need to migrate to AWS.

You can use AWS Migration Hub to create an inventory of your applications and evaluate their compatibility with AWS. Migration Hub provides a centralized view of your application portfolio and helps you plan, track, and manage your migration projects. You can also use third-party assessment tools that support AWS, such as Cloudamize or Evolve.

Mobilize

It can be a significant challenge to find the right path for rehosting (lift and shift) large scale infrastructure. While there are numerous best practices that are helpful, the choice of tool depends on multiple factors, such as workload type, affordable downtime, and operating system requirements. We recommend that you use AWS Application Migration Service to rehost.

AWS Application Migration Service

You can use Application Migration Service to quickly lift and shift physical, virtual, or cloud servers without compatibility issues, performance impact, or long cutover windows. Application Migration Service continuously replicates your source servers to your AWS account. Then, when you're ready to migrate, Application Migration Service automatically converts and launches your servers on AWS with minimal downtime. For more information, see the What Is AWS Application Migration Service? Application Migration Service User Guide.

AWS Migration Hub Orchestrator

AWS Migration Hub Orchestrator simplifies and automates the migration of servers and enterprise applications to AWS by using Application Migration Service. It provides a single location to run and track your migrations. You can use Migration Hub Orchestrator to migrate SAP NetWeaver-based applications—such as S/4HANA, BW/4HANA, SAP ECC on HANA, and others—to AWS and rehost supported custom applications to Amazon EC2. Migration Hub Orchestrator offers templates to create a migration workflow that can be customized to fit your unique migration requirements. Also, Migration Hub Orchestrator automates the steps in your chosen workflow and displays the status of migration.

VM Import/Export

AWS VM Import/Export enables you to import VM images from your existing virtualization environment to Amazon EC2, and then export them back. This enables you to migrate applications and workloads to Amazon EC2, copy your VM image catalog to Amazon EC2, or create a repository of VM images for backup and disaster recovery. For more information, see What is VM Import/Export? in the Amazon EC2 User Guide.

After assessing the workloads for migration, create a migration plan that outlines the migration strategy, timeline, and costs involved in the migration process. You can use AWS Pricing/TCO Tools to estimate the cost savings of running your applications on AWS. You can also use Application Discovery Service to identify the right AWS services to host your migrated workloads.

Migrate

Migrating a Windows workload to AWS involves several phases, including the migration planning, readiness assessment, and migration implementation phases. The migrate phase is the last phase, which involves migrating the Windows workload to AWS. Here are some steps to consider during the migrate phase:

- **Prepare the AWS environment –** Before you begin the migration process, you must prepare the AWS environment by creating an Amazon Machine Image (AMI) and setting up a VPC where you're migrating the workload.
- **Select the migration tool –** There are various migration methods to choose from, including Migration Hub, Application Migration Service, and VM Import/Export. Choose the method that best suits your needs.
- **Configure the migration –** Configure the migration by selecting the source server and specifying the target instance type, storage, and network settings.
- **Perform the migration –** After the configuration is complete, perform the migration. The process involves replicating the data, testing the migrated workload, and performing final cutovers to switch over to the migrated workload. The migration tool you selected above will guide you through these steps.
- **Validate the migration –** After the migration is complete, validate that the migrated workload is functioning as expected. Perform tests and ensure that the security and compliance requirements are met.
- **Optimize the migrated workload –** Optimize the migrated workload by resizing the instance, configuring auto-scaling, and implementing cost-saving strategies such as Reserved Instances or Spot Instances.
- **Monitor and manage the migrated workload –** Continuously monitor and manage the migrated workload to ensure optimal performance and security. You can use Amazon CloudWatch for monitoring.

Migrating file servers

Storage is an essential component to any workload you run. AWS has a number of options to store files in the cloud, including block, file, and object storage. For Microsoft workloads the most common options are block and file storage options. This section provides strategies to help you migrate your storage for Microsoft workloads to the AWS Cloud and guides you through the migration of your file servers.

Assess

There are three major storage types: object, block, and file storage. AWS offers a wide portfolio of storage services that can be categorized under each of these. A successful migration depends on understanding your current needs and then comparing them with various AWS storage services to gauge what works best for you. Choosing the right technology for your workload is key to long-term success. We recommend that you avoid trying to match exactly what you use currently for storage. Instead, we recommend that you look into what all the available options are and select the option that makes the most sense to optimize the cost and performance of your Microsoft workloads. For example, consider a large on-premises file server that requires local block storage. On AWS, the optimal choice could be to move it to Amazon FSx to get the same performance you had for your file server, while removing the undifferentiated heavy lifting of administering the file server and backend storage.

TCO is a key item to evaluate as you assess which storage option works best for you. Keep in mind that using an AWS managed service to help reduce operation costs can help you choose the right overall storage solution on AWS. To request a storage assessment, contact us at migration-evaluator@amazon.com. A storage specialist will help you assess your workloads, map your workloads to the most appropriate AWS storage service, and provide you with directional cost estimates. The storage assessment has three phases:

- You start the discovery process by installing an agentless collector or receiving output from an existing tool set in a flat file.
- You let the discovery process run for 7–60 days.
- The storage collector analyzes the data from the discovery tool, and then proposes a target storage solution and provides directional cost estimates for the solution.

If the cost is slightly higher for a storage option, consider if that storage option reduces the overall cost in the long term and find out what your teams must do to maintain the security and reliability of your storage. It could be the right long-term solution for your workload.

When you're assessing the right solution it's important to look at performance and costs. You can use tools like Windows Performance Monitor to identify the IOPS, throughput, and other performance needs of your workload and then implement the same testing on the AWS solution that you choose for your workload. Additionally, you can use the CloudWatch agent to view metrics for Performance Monitor on a Windows server and analyze the metrics of your workloads before putting those workloads into production.

Identify the AWS storage service that best meets your needs

The choice of storage service typically depends on your use case, application needs, familiarity, performance profiles, and data management capabilities. Consider the following:

- **Amazon Simple Storage Service (Amazon S3)** – Amazon S3 is object storage built to store and retrieve any amount of data from anywhere. Amazon S3 offers a range of storage classes that you can choose from based on the data access, resiliency, and cost requirements of your workloads. You can implement file-based access to Amazon S3 by using AWS Storage Gateway. This enables you to take advantage of the low cost storage of Amazon S3, while not having to completely rewrite an application that uses a Server Message Block (SMB).
- **Amazon Elastic Block Store (Amazon EBS)** – Amazon EBS provides block-level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volumes that are attached to an instance are exposed as storage volumes that persist independently from the life of the instance.
- **Amazon FSx** – Amazon FSx offers four different file systems: NetApp ONTAP, OpenZFS, Windows File Server, and Lustre. For guidance about choosing the right system, see Choosing an Amazon FSx File System in the Amazon FSx documentation. Amazon FSx offers a managed file storage solution in various file system types to enable you to migrate your Microsoft workloads to AWS and remove some of the operational overhead from your IT staff. This enables IT to focus on other critical business drivers.
- **AWS Snow Family** – If you have petabyte scales of data to move into AWS, consider using a storage solution from the AWS Snow Family. While your storage won't rely on the AWS Snow Family device for the long-term life of your data, it can help you seed large data sets to AWS offline by using an AWS Snowcone, AWS Snowball, or AWS Snowmobile device. For more information, see the Seamlessly migrate large SQL databases using AWS Snowball and AWS DataSync post on the AWS Storage Blog.

We recommend that you conduct tests by using stress/load testing tools before moving production data, after you identify the storage service for your workloads. For example, if you're moving your SQL databases on Amazon FSx for Windows File Server, you can use Microsoft SQL Server Distributed Replay. Similarly, you can use DISKSPD for general IOPS and throughput.

Mobilize

After you identify a storage service, the next step is to select a tool for data transfer. Several tools are available, including older solutions like Robocopy and more modern tools like AWS DataSync. DataSync includes a number of controls that aren't available in tools like Robocopy, such as scheduled transfer and easier control of network throttling to help migrate your data without impacting your overall network traffic. For more information about successful migrations completed with Data Sync, see the customer testimonials in AWS DataSync customers.

If you're more comfortable with Robocopy, you can use it to migrate your data to AWS. We recommend that you review this guide on how to optimize file transfer performance. The guide can help you avoid running into issues during your migration. If you use Robocopy with a file system that has deduplication enabled, see Data deduplication in the Amazon FSx Windows User Guide and Troubleshooting Data Deduplication Corruptions in the Microsoft documentation to avoid issues with data corruption.

AWS Storage Gateway can migrate data to AWS in three ways: files, volumes, and virtual tapes. You can install Storage Gateway on a VMware or Hyper-V hypervisor running on-premises, an Amazon EC2 instance in your Amazon VPC, or a dedicated hardware appliance.

Storage Gateway can help you bridge the gap from on-premises to AWS and help you reduce your costs. You can use Storage Gateway to implement your migration in phases and use it to replace an on-premises backup device and tapes with a virtual tape library (VTL). You could also use Storage Gateway as an archival storage solution to start migrating only your local unused files to AWS as the first phase of your migration. There are a number of options for using Storage Gateway to host your Microsoft workload on AWS.

Migrate

DataSync and Robocopy are both equipped to preserve network access control lists (ACLs, also known as Windows ACLs). Before you begin migration, we recommend that you take a backup copy of ACLs by using icacls and review the following resources:

- Migrating on-premises file shares to Amazon FSx for NetApp ONTAP (AWS Storage Blog)
- Migrating existing file storage to Amazon FSx (Amazon FSx Windows User Guide)
- Transferring files from on premises to AWS and back without leaving your VPC using AWS DataSync (AWS Storage Blog)
- Migrate small sets of data from on premises to Amazon S3 using AWS SFTP (AWS Prescriptive Guidance)

Migrating SQL Server

In your journey to the cloud, you have multiple options for migrating your SQL Server environments to AWS. A successful migration is based on generating a detailed inventory of your SQL Server workloads and their dependencies, identifying your authentication scheme, capturing your high availability and disaster recovery (HADR) requirements, assessing your performance targets, and evaluating your licensing options. This inventory helps you determine the target database platform and define your migration options.

You have many options to consider when migrating your SQL Server workloads to AWS, each resulting in optimized price/performance, a more intuitive user experience, and a lower TCO. You can choose to deploy SQL Server on the following: Amazon EC2, Amazon RDS for SQL Server, or Amazon RDS Custom for SQL Server.

Assess

To implement a successful migration, it's important to evaluate your existing infrastructure and understand the key features required for your environment. We recommend that you review the following key areas before choosing a migration plan:

- **Review existing infrastructure** – Review your existing SQL Server infrastructure by using data collected in the discovery phase of your migration (see Windows environment discovery). We recommend that you use the Microsoft prescribed sizing for SQL Server infrastructure on AWS. Understanding current utilization of your on-premises SQL Server instance—including memory, CPU, IOPS, and throughput—is very important to right size your SQL Server instance on AWS.
- **Review existing licensing** – You can take advantage of the complementary AWS Optimizing and Licensing Assessment (AWS OLA) to build a migration and licensing strategy on AWS. AWS OLA provides you with a report that models your deployment options using existing licensing entitlements. These results can help you explore available cost savings across flexible AWS licensing options.
- **Review existing SQL Server architecture** – If you're using a SQL Server failover cluster with shared storage or SQL Server Always On Availability group architecture, then understanding your current high availability architecture requirements will help you define the SQL Server deployment options on AWS.
- **Develop backup strategies** – You can use native backup in SQL Server to back up your databases to the cloud. There are various options to back up databases to Amazon EBS, Amazon FSx for Windows File Server, Amazon FSx NETAPP ONTAP, and Amazon S3 using Storage Gateway. Additionally, you can back up your SQL Server instance by using a snapshot approach. For more information about SQL Server backups, see Backup and restore options for SQL Server on Amazon EC2 in the AWS Prescriptive Guidance documentation.
- **Understand disaster recovery (DR) needs** – If you're moving your existing SQL Server workloads to AWS, then you can use a secondary Region and connect the two Regions by using Transit Gateway (which allows replication to occur). You can use SQL Server distributed availability

group architecture within SQL Enterprise edition to set up DR, or you can use log shipping based on your RTO and RPO requirements. Additionally, you can use AWS Elastic Disaster Recovery (AWS DRS) as an active/passive implementation where you leave your DR as a failover environment. For more information, see the Architect a disaster recovery for SQL Server on AWS: Part 1 post on the AWS Database Blog.

Mobilize

There are three main migration options that we recommend you consider for your SQL Server workloads:

- **Rehosting (lift and shift)** – This involves migrating your on-premises SQL Server databases to SQL Server on an EC2 instance in the AWS Cloud. This approach is useful if a faster migration to AWS is your priority.
- **Replatforming (lift and reshape)** – This involves migrating your on-premises SQL Server databases to Amazon RDS for SQL Server in the AWS Cloud. Replatforming is best suited for when you want to continue using SQL Server but want to offload the undifferentiated heavy lifting tasks, such as installation, configuration, patching, upgrades, and setting up high availability. For a feature comparison of SQL Server on Amazon EC2, Amazon RDS, and Amazon RDS Custom, see Choosing between Amazon EC2 and Amazon RDS in the AWS Prescriptive Guidance documentation.
- **Refactoring (re-architect)** – This typically involves application changes and modernizing by using open-source databases or databases built for the cloud. In this scenario, you modernize your on-premises SQL Server databases to use either Amazon RDS for MySQL, Amazon RDS for PostgreSQL, or Amazon Aurora. By moving to an open-source database you can reduce licensing costs and prevent unnecessary vendor lock-in periods and licensing audits.

Migrate

As you migrate your SQL Server workloads to AWS, take into consideration the following items on configuration and tooling.

Rehosting

Rehosting is homogeneous. Choose this approach when you want to migrate your SQL Server database as-is without changing the database software or configuration. For example, in large-scale legacy migrations, you might want to move quickly to meet your business objectives and choose to rehost most of your applications.

Migrating SQL Server using Amazon EC2

If you migrate to Amazon EC2, you can bring your existing SQL Server licenses. This is known as the Bring Your Own License (BYOL) model. Alternatively, you can purchase license included instances from AWS. For more information, see the Cost optimization with SQL BYOL using license included Windows instance on Amazon EC2 Dedicated Hosts post on the AWS Cloud Operations & Migrations Blog. The BYOL option enables you to reduce costs by using your existing SQL Server licenses. AWS License Manager assists in controlling the allocation of your available licenses when instantiating VMs with SQL Server in Amazon EC2. License Manager helps ensure compliance with licensing rules that you specify.

You can rehost SQL Server to shared-tenancy (default) EC2 instances by using BYOL only if you have Microsoft Software Assurance (SA). If you don’t have SA on your SQL licenses, you can rehost to Amazon EC2 Dedicated Hosts, as long as the licenses were purchased prior to October 1, 2019, or added as a true-up under an active Enterprise Enrollment that was effective prior to October 1, 2019.

There are ways to migrate a SQL Server database to an Amazon EC2 instance by using SQL Server features like backup and restore, log shipping, and Always On availability groups. These options are appropriate if you're migrating a single database or set of databases to a new SQL Server instance running on Amazon EC2. These options are database-native and dependent on specific SQL Server versions and editions. In addition to the database migration, you could also be required to perform steps to migrate objects such as logins, jobs, database mail, and linked servers.

The following approaches are available for rehosting your SQL Server databases on AWS:

- Server rehosting by using Application Migration Service or AWS Database Migration Service (AWS DMS)
- SQL Server backup and restore
- SQL Server transactional replication
- Extending your availability group to the cloud
- AWS DMS
- Log shipping

You could also use AWS Launch Wizard for SQL Server to guide you through the sizing, configuration, and deployment of Microsoft SQL Server on Amazon EC2. It supports both SQL Server single instance and HA deployments on Amazon EC2. To learn more, see AWS Launch Wizard for SQL Server.

Migrating SQL Server using Application Migration Service

Application Migration Service is a good option if you want to lift and shift one or more large-scale machines from an on-premises environment to AWS without changing the SQL Server version, operating system, or code in the databases with near-zero or minimal downtime. You can use AWS Application Migration Service to quickly lift and shift physical, virtual, or cloud servers without compatibility issues, performance impact, or long cutover windows. For guidance on migrating a SQL Server database from an on-premises environment to an Amazon EC2 instance by using Application Migration Service, see Migrating Microsoft SQL Server databases to the AWS Cloud in the AWS Prescriptive Guidance documentation. You can also refer to best practices when you use Application Migration Service to migrate Microsoft SQL Server database workloads to AWS.

SQL Server on Linux

The SQL Server database engine basically runs in a similar way on both Windows Server and Linux. However, there are some changes to certain tasks when using Linux. Launch Wizard can help you adjust to these changes and configure highly available solutions. If you have in-house Linux administration expertise, rehosting to Amazon EC2 Linux is a good choice to save on Windows Server licensing costs. Consider using the Windows to Linux replatforming assistant for Microsoft SQL Server Databases tool to automate this process. For more information, see Migrate an on-premises Microsoft SQL Server database to Microsoft SQL Server on Amazon EC2 running Linux in the AWS Prescriptive Guidance documentation.

Replatforming

Replatforming is a homogeneous approach that's best suited for reducing the time you spend managing database instances by using a fully-managed database offering. A fully-managed database in Amazon RDS for SQL Server limits you from accessing the underlying operating system, system volume, or installation of custom drivers. For more information, see Amazon RDS for Microsoft SQL Server in the Amazon RDS User Guide. If fully-managed database capabilities are necessary for your use case or if you want to use existing SQL server licenses, consider replatforming to Amazon RDS Custom for SQL Server.

The Bring Your Own Media (BYOM) option is available for Amazon RDS Custom for SQL Server. BYOM enables you to use your own installation media and licenses, but the licenses must comply with Microsoft's License Mobility terms. You can either replatform SQL Server to Amazon RDS for SQL Server or to Amazon RDS Custom for SQL Server. The choice depends on whether you require access to the underlying operating system, require database customizations, or want to leverage your existing SQL Server licenses by using BYOM.

The following methods are available for migrating SQL Server to Amazon RDS for SQL Server:

- Log shipping using PowerShell or Log shipping using TSQL
- SQL Server backup and restore
- Transactional replication
- AWS DMS

To replatform your SQL Server databases to run on Amazon RDS for SQL Server, consider using the approaches provided in Amazon RDS for SQL Server resources. For information about how to migrate end of support workloads, see the Migrate end of support Microsoft SQL Server databases to Amazon RDS for SQL Server confidently post on the AWS Database Blog. For information about on-premises databases, see Migrating an on-premises database to Amazon RDS Custom for SQL Server in the Amazon RDS User Guide.

Refactoring

Refactoring is heterogeneous. Choose this approach when you're ready to restructure, rewrite, and rearchitect your database and application to take advantage of open-source and built-for-the-cloud database offerings. If you're open to refactoring your database and respective applications, you can modernize your SQL Server workloads to either Amazon RDS for MySQL, Amazon RDS for PostgreSQL, Amazon Aurora MySQL-Compatible Edition, or Amazon Aurora PostgreSQL-Compatible Edition. You can refactor depending on many modernization timelines and performance requirements.

Amazon RDS for MySQL and Amazon RDS for PostgreSQL are fully-managed database offerings for their respective open-source databases. Amazon Aurora is a relational database management system (RDBMS) built for the cloud with full MySQL and PostgreSQL compatibility. Aurora features a fault-tolerant storage system and gives you the performance and availability of commercial-grade databases at one-tenth the cost.

You can also use Amazon Aurora Serverless to run your database on AWS without managing database capacity. Amazon Aurora Serverless v2 scales instantly to hundreds of thousands of transactions in a fraction of a second. You pay only for the capacity your application consumes, and you can save up to 90 percent on database costs compared to the cost of provisioning capacity for peak load.

To refactor your SQL Server databases to one of these offerings, consider using AWS Schema Conversion Tool (AWS SCT) with AWS DMS. For more information, see AWS SCT in the *Migrating Microsoft SQL Server databases to the AWS Cloud* guide.

If your goal is to accelerate your application and database migrations to AWS, consider using Babelfish for Aurora PostgreSQL. Babelfish enables applications that were originally written for SQL Server to work with Aurora with minimal code changes. As a result, the effort required to modify and move to Babelfish for Aurora PostgreSQL applications developed for SQL Server 2019 or older is reduced, leading to faster, lower-risk, and more cost-effective refactoring.

Consider the following resources for migrating with Babelfish:

- Migrate from SQL Server to Amazon Aurora using Babelfish (AWS Database Blog)
- Prepare for Babelfish migration with the AWS SCT assessment report (AWS Database Blog)
- Migrate from SQL Server to Aurora PostgreSQL using SSIS and Babelfish (AWS Database Blog)
- Using Babelfish as a target for AWS Database Migration Service (AWS Database Migration Service User Guide)

Additional resources

- Migrating SQL Server to AWS prescriptive guidance (AWS Prescriptive Guidance)
- Migration and Modernization Strategies for your SQL Server on AWS (AWS Blog)

Migrating .NET applications

Migrating your .NET applications to AWS enables you to create highly available workloads with elastic scaling capabilities, reduce operation overhead, and increase your business agility by focusing on your differentiating value. This section focuses on the different options for hosting your .NET applications on AWS. You can choose between using a VM, a managed solution such as

AWS Elastic Beanstalk, containerizing your code, or refactoring your code to a microservices- or serverless-based architecture.

Assess

Choosing a migration path for your .NET workload relies on the following key factors:

- **Find the .NET version used** – There are two different .NET implementations supported by Microsoft: .NET Framework (1.0–4.8) and .NET (.NET Core 1.0–3.1 and .NET 5 and later). Both share many of the same components and can run application code written using the different .NET programming languages (such as C#, F#, and VB.NET). Choosing a migration strategy and hosting service depends on the runtime used since .NET Framework runs on Windows while the newer .NET is multi-platform. For the .NET Framework, you can either host on a Windows OS or refactor your code to use the newer .NET. The newer .NET can also be hosted on Linux OS-based services. When modernizing .NET Framework-based workloads, you can use Porting Assistant for .NET or the AWS Toolkit for .NET Refactoring to scan your code and generate a compatibility assessment report. By finding if there are incompatible .NET Framework APIs referenced by your project, you can plan for the complexity of a migration project and decide if and when to refactor your code to use a newer runtime.
- **Review your current deployment** – Check if the currently migrated workload has existing CI/CD pipelines that can be updated to deploy the same workloads to the cloud. Using an existing build and deploy pipeline can reduce the time it takes to deploy your application to the cloud by automating the steps necessary for building, configuring, and deploying your workloads.
- **Review your roadmap** – Depending on the current state of the project, you might already be planning to rearchitect or redesign your applications. Any modernization performed should take the product roadmap into consideration. For example, deciding to containerize existing code or refactoring a monolithic architecture into microservices is ideally part of the product roadmap and aligned with other development efforts.

Mobilize

There are three different migration paths to consider when migrating your .NET workloads to AWS. You can choose between the different options depending on the complexity of your existing codebase, time allocated for migration, and the size of the team allocated to support the migration effort. When considering modernization as part of your migration, it's a best practice to be aligned with the product’s roadmap.

- **Rehost (lift & shift)** – You can choose this approach if your priority is faster migration to AWS with little to no changes. You can rehost ASP.NET-based websites to Internet Information Services (IIS) running on Amazon EC2 instances. You can rehost your desktop-based applications (such as Windows Presentation Foundation, Web Forms, and .NET MAUI) to one of the end user computing platforms such as Amazon AppStream 2.0 or Amazon WorkSpaces.
- **Replatform** – Replatforming is best suited for when you want to host your application using a managed service without making code changes but want to reduce your operational overhead by offloading undifferentiated heavy lifting such as installation, patching, upgrades, and instance management. This strategy is also suited for teams who want to move to container-based workloads. You can replatform your existing applications to Elastic Beanstalk, or use Docker containers hosted on Amazon ECS, Amazon EKS, or AWS App Runner.
- **Refactor** – Choose this approach if you can invest time and effort into making code and architecture changes that reduce operational overhead and achieve better scaling, high availability, and disaster recovery by using AWS cloud-native services. Refactoring involves modernizing your codebase by porting existing .NET framework applications to .NET (previously .NET Core) or modernizing an existing codebase to run better in the cloud. You can use the AWS SDK for .NET to call many AWS cloud services from within your .NET code. Tools such as Porting Assistant for .NET and AWS Microservice Extractor for .NET can be used to port your codebase from .NET Framework to .NET and break your monolithic application into microservices. By refactoring your existing .NET workloads to run on Lambda, you can use serverless computing to avoid provisioning and managing infrastructure.

Migrate

The steps of your .NET workloads migration depend on the migration path that you chose during the assess stage and your type of application.

Rehost .NET applications

Choose this migration path if you want to migrate your application without making any code changes but want to benefit from automatic scaling, load balancing, and elasticity in the cloud. For Windows-based websites, rehosting usually means running them on Internet Information Services (IIS) on AWS. For desktop-based applications, you must install the application and enable users to connect to the application from outside.

Internet Information Services on AWS

Internet Information Services (IIS) is a Microsoft web server that runs on a Windows operating system and is used to host websites and web services. IIS can be installed on any EC2 instance running Windows Server. After IIS is enabled and configured, you can deploy your ASP.NET websites and services by using the same deployment mechanism that you use for on-premises environments.

If you host IIS on EC2 Windows instances, it's important to follow the AWS Well-Architected Framework by using load balancing, Auto Scaling groups, and multi-AZ deployment depending on your workload and HADR needs. We recommend using the AWS Launch Wizard because it guides you through the sizing, configuration, and deployment of a Windows Server workload running IIS resources on AWS. Launch Wizard deploys a highly available architecture that spans two Availability Zones with the required compute, networking, and storage components for a newly created or existing VPC.

Hosting desktop applications on AWS

Many clients have the need to access Windows based thick client applications. You have the choice between three different platforms:

- Amazon EC2 – Choose this option if you want your users to connect to a Windows Server-based environment by using Microsoft Remote Desktop. With this option you're responsible for patching and maintaining your operating system. You must also purchase additional Remote Desktop Services client access licenses (RDS CALs) for your users and active Software Assurance (SA). For more information, see Microsoft Licensing on AWS in the AWS documentation.
- Amazon WorkSpaces – Choose this option if you need a fully managed virtual desktop infrastructure (VDI) for your users. You can use WorkSpaces to provide a persistent Windows Desktop experience to your users. You can also customize your WorkSpaces environment and install .NET applications by using a custom image, or use AWS System Manager to deliver your .NET applications to your WorkSpaces environments. Users can connect either by using their browser or the Amazon WorkSpaces client.
- Amazon AppStream 2.0 – Choose this option to provide secure, reliable, and scalable access to applications and non-persistent desktops from any location. You can use AppStream 2.0 to enable your users to access your .NET applications from the web. If you already have existing RDS CALs and active SA, then you can use those licenses with AppStream 2.0 by using License Mobility.

Replatform

Replatforming involves changing your hosting environment with little to no code changes. Choose this strategy to reduce your operational overhead and take advantage of cloud capabilities and services.

AWS Elastic Beanstalk

You can use Elastic Beanstalk to replatform your .NET Framework workloads. If you package your ASP.NET-based or ASP.NET Core-based applications, then you can quickly deploy and manage applications in AWS without having to learn about the infrastructure that runs those applications. This reduces complexity without restricting choice or control. You simply upload your application and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

To learn more, see the following resources:

- Creating and deploying .NET applications on Elastic Beanstalk (AWS Elastic Beanstalk Developer Guide)
- Working with .NET Core on Linux (AWS Elastic Beanstalk Developer Guide)
- Multi-App Support with Custom Domains for .NET and AWS Elastic Beanstalk (AWS Developer Tools Blog)

Containerize existing applications

You can use Amazon ECS or Amazon EKS to host your Docker-based containerized applications. AWS manages both services. The choice between the two depends on existing knowledge and preference. Both options can run either Linux-based containers or Windows-based containers.

To learn more, see the following resources:

- Amazon EC2 Windows containers (Amazon ECS Developer Guide)
- Enabling Windows support for your Amazon EKS cluster (Amazon EKS User Guide)
- Running Windows Containers with Amazon ECS on AWS Fargate (AWS Blog)
- Speeding up Windows container launch times with EC2 Image builder and image cache strategy (AWS Blog)
- Quick start: CI/CD for .NET Applications on AWS Fargate (AWS documentation)

Containerizing .NET based applications depends on the .NET runtime used. Consider the following:

- **.NET Framework-based applications run on Windows containers –** Adding Docker support to existing applications is done by creating a Docker file that outlines how the application needs to be containerized. You can use AWS App2Container to easily containerize and migrate existing .NET Framework-based applications to AWS. App2Container scans your IIS server to determine the required files and extracts the target application to create a Docker image. You can also use App2Container to create the deployment artifacts needed to host your application in the AWS Cloud.
- **.NET or .NET Core –** In addition to running newer .NET-based web applications on Amazon ECS or Amazon EKS, you can also use AWS App Runner. App Runner is a serverless, fully managed solution that runs your code or container image and manages load balancing, auto scaling, logging, certificates, and networking.

Refactor/re-architect existing code

Choose this option if you have a strong business need to add features, scale, or performance that's otherwise difficult to achieve in the application's current environment. Depending on your application roadmap you can choose to change your code to use the latest framework, cloud-native services, or re-architect it to better run in the cloud.

The first refactoring option available is to migrate your existing .NET Framework application to .NET. The move to .NET gives you the benefit of running on Linux instead of Windows. This reduces your total licensing cost, gives you the latest frameworks, and offers the newest versions of the .NET programming languages.

AWS SDK for .NET

AWS SDK for .NET simplifies the use of AWS services by providing a set of libraries that are consistent and familiar for .NET developers. The AWS SDK offers cross-platform support and is distributed using NuGet. Developers can use the AWS SDK to easily call cloud services from their .NET code, meeting their application's storage, queuing, authentication, and configuration requirements.

Modernize .NET Framework applications

You can migrate from the .NET Framework by using Porting Assistant for .NET, which scans your code files and creates a report that helps plan your application portfolio migration roadmap.

Porting Assistant can also reduce your porting overhead by identifying incompatible .NET Core APIs and packages and finding known replacements. The AWS Toolkit for .NET refactoring is a Visual Studio extension that reduces the time and effort required for developers to refactor legacy .NET applications to cloud-based alternatives on AWS. It assesses the application source code to recommend possible modernization pathways such as porting to .NET Core, identifies Windows-specific IIS and Active Directory dependency configurations, performs code modifications where possible to enable Linux compatibility, and helps validate the refactored application on AWS services. Migrating .NET Framework applications to .NET enables running them on ARM64-based Graviton processors for a better price to performance ratio. For more information, see .NET on Graviton on GitHub and Graviton2 and containers from *Optimizing cost with AWS Graviton based services* in the AWS Workshop Studio documentation.

Monolith to microservices

Many development teams want to re-architect their existing monolithic applications into microservices. By moving to microservice-based architectures, your development teams can increase development agility, decrease compute costs, scale services individually, and decrease their deployment times. AWS Microservice Extractor for .NET simplifies the process of refactoring older monolithic applications into a microservice-based architecture. By identifying components and grouping functionality, development teams can incrementally extract functionality from .NET Framework monolithic applications into .NET services.

Refactor to serverless applications

AWS Lambda is a serverless, event-driven compute service that enables you run code for virtually any type of application or backend service without provisioning or managing servers. You can extract logic from your existing application to create event-based serverless workflows that scale automatically when needed by using .NET and Lambda. Common use cases for Lambda include event driven workloads that run for a few seconds or minutes with varying scaling needs, such as file processing, analytics, websites, and mobile applications. For more information, see Building Lambda functions with C# in the Lambda Developer Guide.

Additional resources

- Amazon CodeCatalyst (Amazon CodeCatalyst documentation)
- AWS Toolkit for Azure DevOps (AWS documentation)
- Setting up a CI/CD pipeline by integrating Jenkins with AWS CodeBuild and AWS CodeDeploy (AWS DevOps Blog)

- About the AWS Deploy Tool for .NET (AWS GitHub)
- .NET on AWS (AWS documentation)
- aws/dotnet (GitHub)

Migrating Windows failover clusters

A Microsoft failover cluster is a group of servers with mostly shared storage between them. You can use failover clusters to facilitate high availability for your applications and services. You can also migrate your failover clusters to the AWS Cloud to benefit from its reliability, performance, and lower TCO.

Windows failover clusters work differently in the cloud than in on-premises environments. It's important to note that only multi-subnet clusters can be deployed in the cloud. Unlike in on-premises environments, the IP address in a Windows failover cluster is assigned to an Elastic Network Adapter (ENA) rather than at the operating system level. In an on-premises environment, the operating system handles IP address assignment, but a cloud provider (AWS) handles the IP address assignment in the cloud. Because failover clustering is an operating system level feature it can't take control of the IP failover. Therefore, the same IP can't fail over between nodes. To work around that, you can use multi-subnet clusters where clusters fail over to a secondary IP. The secondary IP is assigned to ENA in another subnet and can come online. For more information, see Failover Clustering Networking Basics and Fundamentals in the Microsoft documentation.

Migrating a Windows failover cluster to AWS can be a complex process, but with careful planning and implementation it can be done with minimal disruption to your business operations. For example, every application is configured differently on a failover cluster, so it's imperative to understand its needs and then find out how they can be met in the cloud beforehand. The process involves the following steps:

- Ensuring that all cluster nodes are running the same version of Windows and all necessary updates
- Configuring the cluster quorum
- Ensuring that all applications and data are backed up and can be restored during the migration

Assess

The assess phase is a critical step in the process of migrating a failover cluster to AWS. During this phase, you gather information about your current environment, determine the feasibility of migrating to AWS, and identify any potential challenges or risks. We recommend that you follow these steps during the assess phase:

- **Assess the readiness of your applications –** Determine whether your applications can be migrated to AWS without modifications or if they need to be updated or rewritten to take advantage of cloud-native services.
- **Evaluate your networking and security requirements –** Determine your network and security requirements, including the configuration of firewalls, load balancers, and VPNs.
- **Assess your data migration requirements –** Determine how your data gets migrated to AWS, including the size and location of your data, the time required for the migration, and any data transfer costs. In an on-premises environment, you might be using diverse storage technologies like JBOD, NAS, and SAN. Each one can present data to your application through different access methods, such as SAN Fiber Channel, iSCSI, SAS, or SMB/NFS shares.
- **Identify potential risks and challenges –** Identify any potential risks or challenges that could impact the migration process, such as downtime, compatibility issues, or data loss.
- **Estimate costs –** Estimate the cost of migrating to AWS, including the cost of EC2 instances, storage, data transfer, and any other AWS services required.
- **Create a migration plan –** Based on the information gathered during the assess phase, create a detailed migration plan that includes timelines, required resources, and the steps involved in migrating to AWS.

Evaluate your current environment

Assess your current environment, including the hardware and software configurations, to determine what needs to be migrated to AWS. Identify any dependencies between applications, servers, and databases.

Determine your migration strategy

Consider your options for migrating to AWS, including a lift-and-shift approach or re-architecting your environment to take advantage of cloud-native services.

- **Traditional failover cluster migration –** If you're configuring a cluster from scratch in the cloud, you can follow the steps from Tutorial: Set up a Windows HPC cluster on Amazon EC2 in the Amazon EC2 User Guide for Windows Instances, while skipping the HPC-specific steps. Alternatively, you can create a SQL Server Always On availability group cluster without going through the SQL-specific steps. Shared storage is one of the most important considerations for a failover cluster migration. Amazon EBS multi-attach doesn’t support SCSI-3 Persistent Reservation, but Amazon FSx for Windows File Server and FSx for NetApp ONTAP both work well as shared storage options. One of the most common use cases is using an Always On Failover Cluster Instance for a SQL Server cluster with Amazon FSx for Windows File Server. For more information, see the Simplify your Microsoft SQL Server high availability deployments using Amazon FSx for Windows File Server post in the AWS Storage Blog. The next step is bringing the nodes to the cloud. This can be achieved by using Application Migration Service. For more information, see the Migrating your Microsoft Windows clusters to AWS using CloudEndure Migration post in the AWS Storage Blog. Then, you can configure a clustered role for your application to provide high availability.
- **Migrating with virtually no downtime using a stretch cluster –** A stretch cluster could be a good fit if you have a business-critical application to migrate to the cloud and can't afford downtime. With a Microsoft stretch cluster, Site A and Site B must communicate with each other over a network but they can have their own individual shared storage. You can use this to your advantage in a migration scenario. For example, your source (whether it's on-premises or in another provider's cloud) can be Site A, which has network connectivity with an Amazon VPC where you deploy site B. After Site B is up and running, you can cut over to site B. The data replication mechanism is critical in this approach because your source storage technology might have limiting factors in terms of what replication method could work.
- **Migrating a failover cluster deployed on VMware on-premises to VMware in the cloud on AWS –** VMware Cloud on AWS (VMC on AWS) has native support for SCSI-3 Persistent Reservation. This makes it possible to host a failover cluster on a virtual machine disk (VMDK) on VMC on AWS. For more information, see Migrating SQL Server FCI cluster with shared disks to VMware Cloud on AWS in the VMware documentation.

The assess phase is critical for ensuring a successful migration of your failover cluster to AWS. If you take the time to gather information and identify potential challenges, you can develop a comprehensive migration plan that minimizes downtime, reduces risk, and ensures a smooth transition to AWS.

Mobilize

During the migration of a failover cluster to AWS, the mobilize phase involves preparing the cluster for migration to AWS and testing it to ensure its functioning properly. The mobilize phase includes the following steps:

1. **Prepare the target environment** – In this step, you create the AWS resources needed to host the failover cluster. This involves setting up an Amazon VPC, subnets, security groups, and other necessary resources.
2. **Prepare the source environment –** In this step, you prepare the existing failover cluster for migration. This can involve making changes to the network configuration, configuring replication, or installing necessary software.
3. **Validate the cluster** – After both the source and target environments are prepared, you can perform a validation test to ensure that the cluster is functioning properly. This involves running a series of tests to ensure that the cluster can fail over to the target environment successfully.
4. **Create a replication link –** After the validation test, you can create a replication link between the source and target environments. This ensures that any changes made to the source environment are replicated to the target environment.
5. **Monitor replication –** After the replication link is established, monitor the replication process to ensure that all changes are being replicated properly.
6. **Fail over the cluster** – After verifying that replication is working correctly, perform the final failover to the target environment. This involves stopping the cluster services on the source environment and starting them on the target environment.
7. **Test the failover –** After the failover is complete, perform a test to ensure that the applications and services running on the cluster are functioning properly in the new environment

Migrate

Migrating a Microsoft failover cluster can be a complex process that requires careful planning and implementation to ensure a successful outcome. It's essential to thoroughly assess the existing environment, identify potential issues, and develop a comprehensive migration plan that includes testing and validation before making any changes to the production environment. During the migration phase, it's important to closely monitor the process and address any issues or unexpected behavior promptly. Communication and collaboration between all stakeholders—including IT teams, business users, and vendors—are crucial for a smooth migration process.

Additionally, it's important to consider the impact of the migration on any third-party applications or services that are running on the failover cluster. Identify any dependencies and test those applications thoroughly to ensure that they continue to function as expected after the migration. Another key aspect of the migration phase is to establish a rollback plan in case of any unforeseen issues or failures during the migration process. This plan ideally includes steps to revert the migration and restore the original environment, while minimizing any impact on the production environment.

Finally, after the migration is complete and the failover cluster is successfully running on the new environment, it's important to perform post-migration validation and testing to confirm that everything is working as intended. This includes monitoring performance, validating failover capabilities, and ensuring that all applications and services are functioning properly.

Monitoring Microsoft workloads

Microsoft workloads typically use SQL Server in the backend to retrieve and persist data. Often in the journey to the cloud, a rehost decision is made for such a solution by using a simple lift-and-shift approach. When such applications are hosted on a Windows on Amazon EC2 platform, you can use native Windows-based tools to monitor the health of these applications at the server level. However, getting a holistic view across the different components and servers deployed as part of the solution is a challenge, but this pain point can be addressed by Amazon CloudWatch Application Insights.

CloudWatch Application Insights is a cloud-native monitoring service that can help you set up and monitor application resources for your AWS workloads. Enterprise customers deal with a variety of workloads and require a monitoring service that can correlate telemetric data from different sources. If you're an enterprise customer, then CloudWatch Application Insights can help you avoid the complexity in setting up monitoring by automating resource discovery and helping create the application from a variety of resources.

Assess

Tracking an application's performance and backend health is essential for most organizations. You need to know when and where along the journey an abnormality was found and why it happened. You also need to monitor your systems and reduce maintenance costs.

CloudWatch can help you with your monitoring needs, and CloudWatch Application Insights uses CloudWatch metrics, alarms, and events. You can use CloudWatch to set up monitoring and management of metrics, telemetry, and logs for many AWS resources. Amazon CloudWatch ServiceLens provides a combination of services to give you everything you need for monitoring the health of your applications.

Mobilize

CloudWatch Application Insights provides a low-click user interface that you can use to quickly and easily set up the optimal telemetry metrics and logs for your applications. CloudWatch Application Insights tailors its monitors to your specific workload so you can continuously analyze signs of problems for your specific applications. It also delivers auto-configuration and analysis of recommended workload telemetry. Some examples include .NET CLR, requests per second for application/web server technologies, identifying common issues related to .NET garbage collection, and SQL Server failed backups.

When you're looking to onboard a monitoring solution, you typically must understand and configure CPU, memory, and other threshold requirements. However, CloudWatch Application Insights automatically detects these resources and relevant metrics. When you add your applications to CloudWatch Application Insights, it scans the resources, and recommends and configures metrics and logs on CloudWatch for application components. Example application components include SQL Server backend databases and Microsoft IIS/web tiers.

Based on the resource group selected, CloudWatch Application Insights automatically sets up monitoring for each component. In the case of account-based application monitoring, all of the resources discovered in your account are added automatically. You can also benefit from the resource detection capabilities of CloudWatch Application Insights.

CloudWatch Application Insights analyzes metric patterns using historical data to detect anomalies, and continuously detects errors and exceptions from the application, operating system, and infrastructure logs. It correlates these observations using a combination of classification algorithms and built-in rules. Then, it automatically creates dashboards that show the relevant observations and problem severity information to help you prioritize your actions. For common problems in .NET and SQL application stacks, such as application latency, SQL Server failed backups, memory leaks, large and invalid HTTP requests, and canceled I/O operations, CloudWatch Application Insights provides additional insights that point to a possible root cause and steps for resolution.

Built-in integration with AWS Systems Manager OpsCenter allows you to resolve issues by running the relevant AWS Systems Manager Automation document. CloudWatch Application Insights passes the severity level for each problem to AWS Systems Manager OpsCenter, which further helps the you prioritize and assign tasks within your support teams.

Migrate

CloudWatch Application Insights is part of the Windows on Amazon EC2 ecosystem. Using CloudWatch Application Insights for monitoring is an essential part of this offering. After you start the migration of workloads into AWS, you can depend on CloudWatch Application Insights to monitor your Microsoft workloads. Additionally, CloudWatch Application Insights provides support beyond Microsoft workloads, including support for SAP, Java, Oracle, MySQL, PostgreSQL, and other AWS resources (including support for serverless applications). To get started with CloudWatch Application Insights, see Getting set up in the Amazon CloudWatch User Guide.

Migration tools, programs, and training

This section outlines AWS and partner tools available to assist with your cloud migration, the training opportunities available to provide your team with the skills they need for migrating to and operating in the cloud, and key migration programs available to accelerate your migration journey and reduce migration costs.

Tools

Assessment tools

AWS Optimization and Licensing Assessment

We recommend that use the AWS Optimization and Licensing Assessment (AWS OLA) to build your migration and licensing strategy on AWS. You can use the AWS OLA to evaluate your Windows environment. The evaluation helps you to identify potential savings on your licensing costs and discover ways to run your resources more efficiently.

AWS OLA is an obligation free program for new and existing customers. You can use AWS OLA to assess and optimize your current on-premises and cloud environments, based on actual resource utilization, third-party licensing, and application dependencies. A third-party study in 2022 by the Enterprise Strategy Group and Evolve Cloud Services calculated that AWS OLA saves customers an average of 45 percent on Microsoft SQL Server licensing costs and 77 percent on Windows Server. Licensing costs equal three times the cost of actually running these workloads in the AWS Cloud so potential savings can have a significant impact on your TCO.

AWS OLA provides you with a report that models your deployment options. These results can help you explore available cost savings across the flexible licensing options offered by AWS. You can also use AWS OLA in combination with AWS Migration Acceleration Program for Windows to get support and resources during your cloud migration.

You can use AWS OLA before, during, or even after your migration. This tool-based approach can help you determine your actual utilization requirements. The AWS OLA makes recommendations for the lowest cost EC2 instance size and type for each workload. It can also help you find the right blend of On-Demand Instances, Spot Instances, Amazon EC2 Dedicated Hosts, savings plans, and other options specific to your environment. Additionally, the AWS OLA provides you with a migration plan, directional business case, and roadmap.

Licensing savings are a significant part of your TCO, and AWS OLA can help you reduce licensing costs by providing Bring Your Own License (BYOL) or license included recommendations based on your existing licensing entitlements and workloads. AWS OLA optimizes your licenses by configuring instances to require fewer licenses while retaining high performance for your applications. AWS OLA also helps you to understand the differences between on-premises licensing compared to licensing in the cloud. You can use this knowledge to adapt your licensing strategy to further reduce costs in the future.

The scope of AWS OLA includes the following use cases:

- Directional business case, recommendation outlining EC2 instance costs, and configurations based on actual on-premises utilization and data
- Dedicated Host modeling for Host-level licensing
- Virtual CPU (vCPU) reduction for SQL instance optimization and consolidation
- On-premises TCO estimations based on industry averages
- Modeling VMware Cloud on AWS
- Recommendations based on your Microsoft license position (regarding license mobility and potential reduction)
- License impact modelling for T3 Dedicated Hosts
- SQL and Oracle modelling on Amazon RDS, edition optimization, and analysis of Oracle Real Application Clusters (RAC) and Oracle Exadata
- Active and passive modeling for SQL high availability license impact
- Modernization assessment

AWS uses the internal Migration Evaluator or trusted tools from third-party vendors (or qualified AWS OLA migration partners) to conduct broad-based discovery or securely upload exports if you have an existing inventory. The tool that's used depends on your specific needs and requirements. AWS uses discovery tool outputs and combines them with expert recommendations from third-party licensing consultants to give you an optimized TCO that you can trust.

For more information, see the following resources:

- AWS Optimization and Licensing Assessment (AWS documentation)
- Optimize Your Windows Workloads for AWS - AWS Online Tech Talks (YouTube)
- Run Optimization and Licensing Assessment (AWS documentation)

Migration Hub Strategy Recommendations

Migration Hub Strategy Recommendations helps you plan migration and modernization initiatives by offering migration and modernization strategy recommendations for viable transformation paths for your applications. Strategy Recommendations performs an analysis of your server inventory and runtime environment. It can also perform source code and database analysis. Strategy Recommendations combines this analysis with your business goals, and the transformation preferences of the applications and databases provided to recommend the following:

- The most effective migration strategy for each of your applications
- Migration and modernization tools or programs that you can use
- Application incompatibilities and anti-patterns to resolve for a specific option

Strategy Recommendations recommends migration and modernization strategies for rehosting, replatforming, and refactoring with associated deployment destinations, tools, and programs. For example, Strategy Recommendations might recommend straightforward options, such as rehosting on Amazon EC2 by using Application Migration Service. More optimized recommendations might include replatforming to containers by using AWS App2Container or refactoring to open-source technologies such as .NET Core and PostgreSQL.

To use Strategy Recommendations, follow the instructions from Getting started with Strategy Recommendations in the Migration Hub Strategy Recommendations User Guide.

Migration Validator Toolkit PowerShell module

We recommend that you use the Migration Validator Toolkit PowerShell module to discover your Microsoft workloads and migrate them to AWS. The module works by performing multiple checks and validations for common tasks associated with any Microsoft workload. The Migration Validator Toolkit PowerShell module can help your organization reduce the time and effort involved in discovering what applications and services are running on your Microsoft workloads. The module can also help you identify the configurations of your workloads so that you can find out if your configurations are supported on AWS. The module also provides recommendations for next steps and mitigation actions, so that you can avoid any misconfigurations before, during, or after your migration.

AWS Cloud Readiness Assessment

We recommend that you use the AWS Cloud Readiness Assessment to transform your idea of moving to the cloud into a detailed plan that follows AWS Professional Services best practices. You can use the AWS Cloud Readiness Assessment to develop efficient and effective plans for cloud adoption and enterprise cloud migrations, regardless of the size of your organization. This 16-question online survey and assessment report details your cloud migration readiness across six perspectives, including business, people, process, platform, operations, and security.

After you complete an assessment, you can provide your contact details to download a customized cloud migration assessment that charts your readiness and what you can do to improve it. Your summary report includes a heatmap and radar chart with detailed scoring information and resources to help you improve your readiness score. This take-away report cab help you plan and communicate with your stakeholders. For a sample assessment report, see AWS Cloud Adoption Readiness Assessment Report. To take the assessment, go to the AWS Cloud Adoption Readiness Assessment.

Migration tools

AWS Migration Hub

AWS Migration Hub provides a central location to collect server and application inventory data for the assessment, planning, and tracking of migrations to AWS. Migration Hub can also help you accelerate application modernization following migration. Migration Hub network visualization enables you to accelerate migration planning by quickly identifying servers and their dependencies, identifying the role of a server, and grouping servers into applications. To use network visualization, install AWS Application Discovery Agent (Discovery Agent), and then start data collection.

AWS Migration Hub Orchestrator

AWS Migration Hub Orchestrator helps accelerate your application migration to reduce the time and effort of the migration. You can use predefined workflow templates to easily create a migration workflow, customize your workflow per your specific needs, automate the migration steps, and track the migration progress from start to finish in one place. Orchestrator supports the following:

- Migration of applications based on SAP NetWeaver with SAP HANA databases
- Rehosting of any applications to Amazon EC2
- Rehosting of SQL Server databases to Amazon EC2

- Replatforming of SQL Server databases to Amazon RDS
- Importing VM images of an Open Virtual Appliance (OVA) or VMware Virtual Machine Disk (VMDK) to an AMI for Amazon EC2

AWS Migration Hub dashboard

The Migration Hub dashboard shows the latest status and metrics for your rehost and replatform migrations. You can use the dashboard to quickly understand the progress of your migrations and identify and troubleshoot any issues. Migration Hub lets you track the status of your migrations into any AWS Region supported by your migration tools. Regardless of which Regions you migrate into, the migration status appears in Migration Hub when using an integrated tool.

AWS Application Migration Service

AWS Application Migration Service minimizes time-intensive, error-prone manual processes by automating the conversion of your source servers to run natively on AWS. It also simplifies application modernization with built-in and custom optimization options. The use cases for Application Migration Service include the following:

- On-premises workloads such as SAP, Oracle, and SQL Server running on physical servers or on VMware vSphere, Microsoft Hyper-V, and other on-premises infrastructure
- Cloud-based workloads running from other public clouds to AWS

You can use Application Migration Service to access over 200 services that reduce costs, increase availability, and facilitate innovation. Additionally, you can use it to move your EC2 workloads between AWS Regions, Availability Zones, or accounts more easily to meet your business, resilience, and compliance needs.

Alternatively, as a modernization strategy you can optimize your applications by applying custom modernization actions or selecting built-in actions such as cross-Region disaster recovery, CentOS conversion, and SUSE Linux subscription conversion.

AWS Database Migration Service

AWS Database Migration Service (AWS DMS) is a managed migration and replication service that helps move your database and analytics workloads to AWS quickly, securely, and with minimal downtime and zero data loss. AWS DMS supports migration between 20-plus database and analytics engines, including SQL Server.

AWS DMS enables you to use a managed databases model to migrate from legacy or on-premises databases to managed cloud services through a simplified migration process, which gives developers time to innovate. You can also use AWS DMS to break free from licensing costs, accelerate business growth, and use purpose-built databases to innovate and build faster for any use case at scale for one-tenth the cost.

You can also use AWS DMS to do the following:

- Replicate backup files
- Create redundancies of business-critical databases and data stores to minimize downtime and data loss
- Build data lakes to perform real-time processing on change data from your data stores
- Integrate data marts by building data lakes
- Perform real-time processing on change data from your data stores

Migration Partner tools

CloudBasix

CloudBasix makes cloud-native workload optimization and data integration products. You can use its flagship product, CLOUDBASIX for RDS SQL Server Read Replicas and Disaster Recovery (DR), to enable the following:

- In-Region read replicas
- Cross-Region DR
- Inter-cloud Azure to AWS disaster recovery
- AI-driven data lakes and data houses
- Integration for Amazon Redshift and Snowflake

Management tools

Amazon CloudWatch Application Insights

Amazon CloudWatch Application Insights facilitates observability for your applications and underlying AWS resources. It helps you set up the best monitors for your application resources to continuously analyze data for signs of problems with your applications. CloudWatch Application

Insights, which is powered by Amazon SageMaker and other AWS technologies, provides automated dashboards that show potential problems with monitored applications. This can help you quickly isolate ongoing issues with your applications and infrastructure.

When you add your applications to CloudWatch Application Insights, it scans the resources in the applications and recommends and configures metrics and logs on CloudWatch for application components. Example application components include SQL Server backend databases and Microsoft IIS or web tiers. CloudWatch Application Insights analyzes metric patterns using historical data to detect anomalies and continuously detects errors and exceptions from your application, operating system, and infrastructure logs. It correlates these observations using a combination of classification algorithms and built-in rules. Then, CloudWatch Application Insights automatically creates dashboards that show the relevant observations and problem severity information to help you prioritize your actions. For common problems in .NET and SQL application stacks—such as application latency, SQL Server failed backups, memory leaks, large HTTP requests, and canceled I/O operations—it provides additional insights that point to a possible root cause and steps for resolution. Built-in integration with AWS Systems Manager OpsCenter enables you to resolve issues by running the relevant Systems Manager Automation document.

AWS License Manager

AWS License Manager makes it easier for you to manage your software licenses from vendors, such as Microsoft, SAP, Oracle, and IBM, across AWS and your on-premises environments. You can use License Manager to streamline license management by switching between license types and automating the discovery, tracking, and reporting of existing licenses. You can also simplify the windows BYOL experience through the managing of a collection of Dedicated Hosts as a single entity with automated allocation, release, and recovery. Additionally, you can handle marketplace licenses across accounts by automating the distribution and activation of software entitlements and workloads across AWS accounts for end users.

AWS Backup

AWS Backup is a cost-effective, fully managed, policy-based service that simplifies data protection at scale. You can use AWS Backup to make cloud-native backups for key data stores, such as your buckets, volumes, databases, and file systems across AWS services. AWS Backup centralizes your data's protection by providing data protection management for your applications running in hybrid environments, such as VMware workloads and AWS Storage Gateway volumes. You can also centrally manage polices for configuring, managing, and governing your backup activity across your organization's AWS accounts, resources, and Regions.

AWS Systems Manager Fleet Manager

Fleet Manager, a capability of AWS Systems Manager, is a unified user interface (UI) experience that helps you remotely manage your nodes running on AWS or on premises. With Fleet Manager, you can view the health and performance status of your entire server fleet from one console. You can also gather data from individual nodes to perform common troubleshooting and management tasks from the console. This includes connecting to Windows instances by using the Remote Desktop Protocol (RDP), viewing folder and file contents, Windows registry management, operating system user management, and more. You can use Fleet Manager if you want to centralize the management of your node fleet or your Amazon ECS clusters.

Programs

AWS Migration Acceleration Program

The AWS Migration Acceleration Program (MAP) is a comprehensive and proven cloud migration program based upon AWS's experience migrating thousands of enterprise customers to the cloud. Enterprise migrations can be complex and time-consuming, but MAP can help you accelerate your cloud migration and modernization journey with an outcome-driven methodology.

MAP provides tools that reduce costs and automate and accelerate implementation, tailored training approaches and content, expertise from Partners in the AWS Partner Network, a global partner community, and AWS investment. MAP also uses a proven three-phased framework to help you achieve your migration goals. Through MAP, you can build strong AWS cloud foundations while reducing risk, boosting productivity, improving operational resilience, and offsetting the initial cost of migrations. You can also take advantage of the performance, security, and reliability of the cloud.

AWS Windows Migration Accelerator

AWS Windows Migration Accelerator helps reduce the cost of your migration by using AWS Promotional Credit when you accelerate the migration of Windows servers using Application Migration Service. AWS Windows Migration Accelerator incentives can be applied on top of other agreed upon sales incentives and promotional programs. If you use Application Migration Service to migrate at least 40 servers to AWS in one month, including a minimum of 15 Windows servers, you may be eligible to receive a $200 AWS Promotional Credit per Windows server, until December 31, 2023. If you migrate more than 80 servers, including at least 25 Windows servers, in a calendar month, the discount increases to $250 AWS Promotional Credit for each Windows server you migrate to AWS using Application Migration Service. Migrated servers must be migrated from locations outside of AWS and continuously run on AWS for at least four weeks after migration.

AWS Migration Acceleration Program for Windows

The AWS Migration Acceleration Program (MAP) for Windows, an extension of the existing AWS MAP program, is designed to help organizations reach their migration goals even faster with AWS services, best practices, tools, and incentives. AWS uses a three-step approach to help you reduce the uncertainty, complexity, and cost of migrating to the cloud. In addition, MAP can help you modernize current and legacy versions of Windows Server and SQL Server workloads to reduce costs by using cloud solutions such as SQL Server running on Linux, Aurora, container-based services, and Lambda. Cloud-native or open-source solutions can help you break free from the high costs of commercial licensing.

AWS Infrastructure Event Management

AWS Infrastructure Event Management (IEM) offers architecture and scaling guidance and operational support during the preparation and implementation of planned events, such as shopping holidays, product launches, and migrations. For these events, IEM helps you assess operational readiness, identify and mitigate risks, and implement your event confidently with AWS experts by your side. The program is included in the Enterprise Support plan and is available to Business Support customers for an additional fee.

AWS experts lead a highly focused engagement to provide you with architectural and operational guidance for your planned event using a prescriptive, phased approach that helps you do the following:

- Understand your success criteria and desired business outcome
- Assess the readiness of your AWS environment, help identify and mitigate risks, and document your plan
- Confidently host your event with AWS experts by your side
- Analyze results post-event and scale services to normal operating levels, so you can focus on planning your next event

Training

Self-paced, interactive, and classroom training

AWS offers both digital and classroom training to support you in your migration journey. You can start learning with hundreds of self-paced digital training courses built by the experts at AWS. Then, you can gain hands-on skills by completing interactive training with the AWS Skill Builder. With classroom training you can ask questions, work through solutions in person, and get feedback from AWS-accredited instructors with deep technical knowledge. For more information, explore AWS Training and Certification offerings.

AWS Partner training

AWS Partners also offer digital training as self-paced courses covering a range of topics from AWS Cloud fundamentals to machine learning at top online learning platforms such as EdX and Coursera. For more information, explore AWS Partner Training and Certification offerings. You can be certified by role and solution. For example, roles include Cloud Practitioner, Solutions Architect, Developer, and SysOps Administrator. Solutions include Advanced Networking, Data Analytics, Databases, Machine Learning, Security, Storage, and more.

Microsoft licensing on AWS

This section describes how Microsoft licensing works on AWS, provides licensing best practices and strategies for deployment of Microsoft workloads on AWS, and helps you remain compliant with Microsoft’s licensing terms while optimizing costs. Due to the impact of licensing on the cost of a migration, Microsoft licensing and Bring Your Own License (BYOL) options often influence the deployment options available to AWS customers. That's why it's important to understand how licensing works before you start the migration process.

Assess

When assessing your Microsoft workloads for migration to AWS, it's important to consider licensing requirements. For Microsoft workloads, we recommend that you take advantage of an AWS Optimization and Licensing Assessment (AWS OLA) to assess on-premises or cloud workloads and build a right-sized and optimized roadmap for running workloads in AWS. An AWS OLA will not only make optimized suggestions for the right EC2 instances for your workloads, but it will also look at your Microsoft licensing position. The result will be recommendations for the best path forward to save on compute and licensing costs. An AWS OLA is available for new and existing customers, and is fully funded and obligation-free. For more information, contact the AWS OLA team.

If an AWS OLA is not an option for you at this time, it's still important to understand how Microsoft licensing works in AWS. If you're looking to BYOL, we recommend that you request an updated copy of your Microsoft License Statement (MLS) from your Microsoft licensing purchasing contact. Use this to review what licensing you have and any purchase dates and SA quantities where applicable. For assistance with your MLS, reach out to your AWS representative. Your representative can connect you with a Microsoft specialist.

Different Microsoft products have different licensing requirements, so it's important to have a clear picture of what Microsoft products you have deployed. AWS has different options available to meet the needs of different Microsoft products, including shared/default tenancy for Amazon EC2 for products with License Mobility and dedicated options for products without License Mobility. AWS also has license included options, where the cost of the licensing is included in the Amazon EC2 compute costs. You could benefit from a mixed licensing model when migrating to AWS. A mixed licensing model is where shared-tenancy EC2 instances are used with all or some license included options. The mixed licensing model is best for variable workloads and when dedicated EC2 options are used for stable, predictable workloads—especially when Windows Server Datacenter or SQL Server Enterprise BYOL is an option.

For more information about current Microsoft licensing terms for products purchased through Microsoft’s Volume Licensing programs, see the Microsoft Product Terms site.

License included options

License included refers to Amazon EC2 instances that include the cost of the license in the compute costs. For Microsoft server workloads, AWS currently offers Windows Server (Amazon EC2, Amazon EC2 Dedicated Hosts, Amazon EC2 Dedicated Instances, AWS Outposts) and SQL Server Enterprise, Standard, and Web editions (Amazon EC2). These server licenses are offered per vCPU per second with the pay-as-you-go model as a benefit of license included EC2 instances. If the EC2 instance is scheduled to stop, or scales up or down based on demand, you only pay for the licensing for the time the instance is running. With on-demand pricing there are no long-term commitments, which is ideal for future modernization plans.

License included is available for current and legacy versions with Amazon Machine Images (AMIs) available for all supported versions. End-of-support versions, such as Windows Server 2008 or SQL Server 2012, can still be licensed with license included, but you must bring your own media.

There are no software upgrade fees with the license included option. As soon as a new version of the product is released by Microsoft, the new version is made available in the Amazon EC2 console right away for no additional cost above the current license included costs. Most importantly, AWS is responsible for the licensing compliance for license included EC2 instances. This can save a lot of time and effort for you because licensing compliance can be complex and difficult.

The SQL Server license included options offer core-based licenses with no client access licenses (CALs) required. An unlimited number of users can access a license included Windows Server EC2 instance without counting or licensing CALs. Windows Server license included EC2 instances also include two Microsoft Remote Desktop connections for administrative purposes only. If you need additional Microsoft Remote Desktop connections, you can buy Remote Desktop Services User CALs with Software Assurance (SA) from Microsoft and bring them to AWS through License Mobility benefits.

AWS also offers some user-based license included options. Visual Studio 2022 Enterprise and Professional editions (Amazon EC2 and Lambda) and Office LTSC Professional Plus 2021 (Amazon EC2) are charged per user, per month. These include Microsoft Remote Desktop connections for each user. Amazon WorkSpaces also offers Office Professional Plus 2016 or 2019 as an add-on, charged per user, per month.

AWS offers the following license included options for Microsoft workloads:

Product	Availability	Versions available
Windows Server	EC2, EC2 Dedicated Instances , EC2 Dedicated Hosts, Outposts	All*
SQL Server Enterprise	EC2	All*
SQL Server Standard	EC2	All*
SQL Server Web**	EC2	All*
Visual Studio Enterprise	EC2, Lambda	2022
Visual Studio Professional	EC2, Lambda	2022
Office Professional Plus	WorkSpaces	2019, 2016
Office Professional Plus LTSC	EC2	2021

*Out-of-support and supported versions require your own media.

**SQL Server Web edition has a restricted use case based on Microsoft’s licensing terms. SQL Server Web edition may be used only to support public and internet accessible webpages, websites, web applications, and web services. It may not be used to support line-of-business applications (for example, customer relationship management, enterprise resource management, and other similar applications).

License included options are best for variable workloads. For example, this is when workloads don't need to run most of the time or when workloads frequently need to scale up and down.

BYOL options

Using the Bring Your Own License (BYOL) model is a great way to capitalize on your existing investments in on-premises software while benefiting from the efficiencies of the AWS Cloud. BYOL allows you to extend the lifecycle of prior software versions and purchases, and deploy products not offered by AWS as license included. Whenever you bring your own licenses, you must also bring your own media. This means that you must create your own AMI with your own media, rather than using Amazon-provided AMIs. The VM Import/Export tool is free to use and enables you to create your own AMIs. Alternatively, you can use Application Migration Service to create your own media and AMIs.

Microsoft products with License Mobility through Software Assurance

Because AWS is an Authorized Mobility Partner any Microsoft products with License Mobility that are covered by active SA can be brought to AWS on shared or dedicated tenant environments. Products eligible for License Mobility through SA include SQL Server, SharePoint Server, Exchange Server, Project Server, Skype for Business Server, BizTalk Server, Remote Desktop Services User CALs, and System Center Server. Microsoft products that have License Mobility Rights are not affected by the October 1, 2019 licensing changes made by Microsoft. As a result, products with License Mobility don't have any purchase date or version restrictions. They are eligible for BYOL to AWS as long as the licenses have active SA. For example, SQL Server 2022 licenses with active SA can be brought to shared-tenancy (default) EC2 instances (doesn't required Dedicated Instances) as long as SA is maintained.

Products with License Mobility through SA are licensed on AWS the same way they would be within a virtualized on-premises environment, with the exception of System Center Server. System Center Server licenses have specialized license counting applied when being brought to the AWS Cloud. For every 16 cores of System Center Server Datacenter edition, you can manage up to 10 EC2 instances (of any size). For every 16 cores of System Center Server Standard edition, you can manage up to two EC2 instances (of any size).

SQL Server is the most commonly brought product with License Mobility to AWS. SQL Server core licenses with active SA or subscription licenses (except those purchased through the Cloud Solution Provider, or CSP, program) are licensed per vCPU on shared-tenancy (default) EC2 instances, with a minimum Microsoft licensing requirement of four vCPUs per EC2 instance. SQL Server/CAL licenses with active SA are licensed with one server license per EC2 instance. Plus, all users or devices with access must have the corresponding CALs assigned to them. SQL Server also has a passive failover benefit with active SA and subscriptions. For every active, licensed SQL Server on EC2, you're eligible for a secondary, passive SQL Server instance on EC2 without having to license the SQL Server portion on the passive instance. For more information, see the Microsoft SQL Server 2022 Licensing guide (downloadable PDF) on the Microsoft website.

AWS is an Authorized Mobility Partner (downloadable PDF). If you bring Microsoft products with License Mobility to AWS, you must fill out and submit a License Mobility Verification Form to Microsoft. This form is a brief Microsoft Word document that asks for the following:

- Your name and contact information
- Microsoft agreement number
- Your cloud partner
- Products being brought through License Mobility
- Number of licenses that you're bringing

You must submit the form to Microsoft directly or through your Microsoft reseller within 10 days of bringing the products to AWS. To learn more about the verification process, see License Mobility through Software Assurance in the Microsoft documentation. The License Mobility Verification Form has a section to provide information about the Authorized Mobility Partner. You can use *microsoft@amazon.com* as the email address, *Amazon Web Services* as the Partner name, and *aws.amazon.com* as the Partner website. For more guidance, see Microsoft's Verification Guide for Customers (downloadable PDF) in the Microsoft documentation. To download a copy of the License Mobility Verification Form, see Licensing Resources and Documents in the Microsoft documentation.

Note

The Flexible Virtualization Program offered by Microsoft isn't available on AWS because AWS has been named a Listed Provider* cloud by Microsoft. Microsoft named Alibaba, Amazon, and Google Cloud as Listed Providers as part of the October 1, 2019 licensing changes. Beginning October 1, 2019, on-premises licenses purchased without SA and License Mobility rights can't be deployed hosted cloud services offered by Listed Providers.

Microsoft products without License Mobility

Windows Server, Visual Studio, Microsoft Developer Network (MSDN), Windows desktop operating systems, Microsoft Office, and Microsoft 365 apps (formerly Office 365) don't have License Mobility rights granted to them in the Microsoft Product Terms, even if the licenses have SA or are active subscription licenses. As a result, bringing licenses for these products requires dedicated infrastructure: EC2 Dedicated Hosts, EC2 Dedicated Instances, VMware Cloud on AWS, and Dedicated Hosts on Outposts. You must also follow other specific requirements to be eligible for BYOL to AWS. These requirements are a result of changes Microsoft made to the license terms for products without License Mobility when deployed on Listed Provider clouds, effective October 1, 2019. For more information, see Updated Microsoft licensing terms for dedicated hosted cloud services in the Microsoft documentation.

To be eligible for BYOL to AWS, licenses for products without License Mobility must meet the following requirements from Microsoft:

- Licenses must be purchased as perpetual use rights (not subscription).
- The purchase date of the licenses must be before October 1, 2019, or the licenses must be purchased within a Microsoft Enterprise Agreement term that started before October 1, 2019.
- The version deployed must have been publicly available prior to October 1, 2019.
- The product must be deployed on dedicated infrastructure.

Subscription licenses for products without License Mobility will lose BYOL once purchased or renewed on or after October 1, 2019.

Note

Products without License Mobility don't require active SA for BYOL on AWS, as long as the licenses meet the requirements above.

As licensing can be complex, see the Amazon Web Services and Microsoft FAQ site to determine if your licenses are eligible for the BYOL to AWS option. If you don't find the information you need in the FAQ or are unsure where to start with migrating your Microsoft workloads to AWS, contact Microsoft@Amazon.com. AWS has Microsoft workload and licensing specialists available to help make sure you have all the information that you need.

Note

Windows Server BYOL requires EC2 Dedicated Hosts, Dedicated Hosts on Outpost, or VMware Cloud on AWS because Windows Server BYOL must be licensed by a physical core.

BYOL for the Services Provider License Agreement (SPLA)

The Services Provider License Agreement (SPLA) program was not affected by the October 1, 2019 licensing changes made by Microsoft. As a result, net new Windows Server licenses can be brought through SPLA for customers with their own SPLA licensing, without any purchase date or version restrictions. Any core or processor-based products licensed through SPLA require EC2 Dedicated Hosts, where user-based Subscriber Access Licenses (SALs) can be brought to shared-tenancy (default) EC2 instances. This is because the user-based SALs in SPLA are eligible for data center providers (DCPs) in the Services Provider Use Rights (SPUR).

> **Note**
>
> Microsoft has announced that they will no longer allow SPLA BYOL on AWS or the other Listed Provider clouds after September 30, 2025.

Amazon EC2 Dedicated Hosts

Some key capabilities of Amazon EC2 Dedicated Hosts include the following:

- Pre-configured EC2 Nitro and Xen hypervisors with visibility into physical sockets and cores
- Multiple instance sizes within the same family supported on the same Dedicated Host (For the latest set of supported instance types, see Dedicated Hosts on the Amazon EC2 User Guide for Windows Instances.)
- Automated management, auto-scaling, and instance placement control
- Ability to share a host across multiple AWS accounts
- Integrated with AWS License Manager for tracking license usage and management
- Ability to maintain instance affinity to a host
- Automated host recovery
- Continuous monitoring with AWS Config

Because Windows Server BYOL requires dedicated infrastructure and physical core counts, EC2 Dedicated Hosts is a great option that can help you:

- Achieve significant savings

- Enable you to bring any Microsoft application to AWS, regardless of SA or License Mobility (subject to the October 1, 2019 purchase and version requirements)
- Maximize the physical core licensing benefits of Windows Server Datacenter and SQL Server Enterprise editions
- Pay only per host, not per EC2 instance (This means that when you use dedicated hosts you can use the maximum number of instances available on the host without incurring any extra compute charges.)

If you bring BYOL-eligible Windows Server licenses to EC2 Dedicated Hosts, you can license all physical cores (not vCPUs) of the host. For example, an R5 EC2 Dedicated Host has 48 physical cores. Bringing 48 cores of Windows Server Datacenter edition to an R5 EC2 Dedicated Host allows for as many EC2 instances to be deployed on the host as technically possible. Bringing 48 cores of Windows Server Standard edition allows up to two EC2 instances of any size on the host.

You can stack Windows Server Standard edition licenses to allow for additional EC2 instances on the same host, where all of the physical cores of the host licensed a second time allows for two additional EC2 instances (and so on). Licensing SQL Server Enterprise by physical core also requires that all physical cores of the host be licensed. This enables you to deploy the number of EC2 instances for SQL Server on the host equal to the number of physical cores licensed. For example, an R5 EC2 Dedicated Host licensed with 48 cores of SQL Server Enterprise allows you to deploy up to 48 EC2 instances running SQL Server on that host.

If you bring BYOL-eligible Windows Server Datacenter and SQL Server Enterprise licenses and license the total physical cores of the host, you can see significant cost savings over license included for the same number and size of EC2 instances. This assumes the workloads can mostly fill the host and are running most of the time. For example, you could deploy 12 R5.2xlarge EC2 instances on shared-tenancy instances with license included Windows Server and SQL Server Enterprise BYOL with a total of 96 cores of SQL Server Enterprise required for licensing. However, if you deploy an R5 EC2 Dedicated Host (which can fit the same 12 R5.2xlarge EC2 instances), you can bring 48 cores of Windows Server Datacenter and 48 cores of SQL Server Enterprise BYOL-eligible licenses. You would not only save the Windows Server license included costs, but you would also only need to bring half the number of SQL Server Enterprise core licenses.

BYOL on EC2 Dedicated Hosts is best for stable, predictable workloads where you can fill the host by at least 70 percent and where the workloads are running most of the time. To learn more about Microsoft Licensing on AWS, see Microsoft Licensing on AWS on YouTube and Amazon Web Services and Microsoft Frequently Asked Questions in the Microsoft documentation.

VMware Cloud on AWS

To learn more about migrating to VMware Cloud on AWS, see VMware Cloud on AWS overview and operating model in the AWS Prescriptive Guidance documentation.

Mobilize

AWS License Manager

As part of the mobilize phase for Microsoft licensing considerations, we recommend that you input the licenses you're planning to allocate to your workloads in AWS in AWS License Manager. License Manager is a free tool that makes it easier for you to manage your software licenses from vendors such as Microsoft, Oracle, IBM, and SAP across not only AWS but workloads on-premises or in other clouds, too.

Inputting the Microsoft licensing you're bringing to AWS into License Manager will help you:

- Gain greater visibility and control over how software licenses are used and prevent misuse before they happen.
- Save money with the maximum use of licenses, including how you track and manage licenses.
- Reduce the risk of noncompliance by enforcing license usage limits, blocking new launches, and using other controls.
- Increase your productivity by automating the placement, release, and recovery of hosts using host resource groups.

To learn more about License Manager, see Working with AWS License Manager in the AWS License Manager User Guide.

Licensing considerations

Consider planning your migration around the licenses currently assigned to the workloads prior to migration. For example, if you're bringing several on-premises hosts to AWS, consider migrating by host rather than by grouping workloads that fall across several different hosts. This is because as you decommission an on-premises host, you free up the licenses associated to that host for use in AWS. Alternatively, you can use license included instances for Windows Server or SQL Server during your migration and switch over to the BYOL option after the migration is complete. However, this option requires using your own media and AMI from the beginning (even for license included options). The license conversion feature available with AWS License Manager only allows you to switch to BYOL from license included if the EC2 instances were originally created from your own media and AMIs.

Migrate

Within 10 days of deploying your Microsoft workloads on AWS, be sure to submit the License Mobility Verification Form to Microsoft for any licenses with License Mobility that you're bringing to AWS. You can submit this form multiple times, based on the different stages of your migration. The form asks for the following:

- Your name and contact information
- Microsoft agreement number
- Your cloud partner
- Products being brought through License Mobility
- Number of licenses that you're bringing

To learn more about the verification process, see License Mobility through Software Assurance in the Microsoft documentation. For more guidance, see Microsoft's Verification Guide for Customers (downloadable PDF) in the Microsoft documentation. To download a copy of the License Mobility Verification Form, see Licensing Resources and Documents in the Microsoft documentation.

AWS Partners

Benefits of engaging an AWS Competency Partner

Migrating your Microsoft workloads efficiently to the cloud requires careful planning and streamlined implementation. Key steps include scoping, creating a cloud migration business case, gaining executive sponsor alignment, setting cloud financial management KPIs, building a cloud center of excellence, validating migration services, deploying automation tools for large-scale migrations, and extending security strategy to the cloud.

We recommend you engage a validated AWS Competency Partner to lead your organization through your migration journey. AWS Partners are strategic experts and experienced builders that help address the aforementioned key steps and your business objectives by leading you through all phases of your migration journey. The AWS Partner community features over 100,000 partners from over 150 countries who can support you in your cloud journey and help you to focus on innovating, increasing agility, and reducing costs.

Build a plan

AWS Partners can perform readiness assessments, create migration plans, and bring migration tools to accelerate your journey to the cloud. Additionally, they can help you close skill gaps, recommend cost optimization strategies, and help you qualify for exclusive migration incentives to subsidize the cost of migrating to AWS.

Optimize costs

In today's rapidly evolving technological landscape, many organizations face significant cost challenges when it comes to their digital transformation journey. One common concern is the perception that the cloud is too expensive, making it difficult to see the significant business benefits it offers. Additionally, the cost of modernizing your technology stack can pose financial challenges.

Working with an AWS Microsoft Workloads Competency Partner ensures access to the most qualified AWS Partners for deploying Microsoft workloads on AWS. These Partners have validated technical capabilities and demonstrated success in helping customers migrate, manage, or deploy

Microsoft workloads to AWS. Workloads supported by these partners include Windows Server, Microsoft SQL Server, Windows File Server, SharePoint, and .NET applications.

AWS Partners use AWS best practices to build secure, available, reliable, performant, and cost-optimized architectures. Partners also help to fully leverage funding made available by AWS to cost optimize and ensure quicker time to value with their expertise. Finally, AWS Partners can leverage the AWS Migration Acceleration Program for Windows to offset your migration cost to AWS.

Save time

Many enterprises are heavily invested in on-premises infrastructure. It's possible that your organization has made large investments in VMware software to manage your on-premises infrastructure, and would like to use the same on-premises tools to manage your infrastructure on AWS. You may even have specialized workloads and infrastructure that are challenging to migrate to the cloud but have dependencies on migrated workloads. Also, you might have a hybrid infrastructure pattern, where some of your infrastructure is in a traditional on-premises data center with other parts deployed in the cloud.

When time is of the essence, we recommend you engage an AWS Migration Competency Partner with a proven track record of delivering a broad range of large-scale migrations due to their skilled talent, refined processes, and technological capabilities. Supported workload categories include Windows, SAP, Oracle, VMware on AWS, Database, Analytics, Storage, IoT, Machine learning, and Software as a Service.

AWS Partners understand that moving to AWS doesn't mean an all-or-nothing move and getting rid of your present investments. They are adept at optimizing and streamlining infrastructure, optimizing for what parts are best kept on-premises and what parts are best fit for the cloud. AWS has a broad offering of hybrid cloud solutions, which include Amazon VPC, Direct Connect, and Storage Gateway.

AWS Partners can qualify eligible customers for the AWS Migration Acceleration Program (MAP), a comprehensive and proven cloud migration program based on AWS's experience migrating thousands of enterprise customers to the cloud. MAP supports specialized workloads through comprehensive tooling, services, guidance, training, and additional incentives. Specialized workload support is available for mainframe, Windows, storage, VMware Cloud on AWS, SAP, databases, and Amazon Connect.

Enhance security

You may be concerned about the privacy and security of your data. Additionally, you may need assurance that data handling practices comply with the Clarifying Lawful Overseas Use of Data (CLOUD) Act and the General Data Protection Regulation (GDPR). We recommend you engage an AWS Security Competency Partner who can provide you with a team of security experts for delivering security-focused solutions for your specific workloads and use cases. AWS Partner solutions enable automation and agility and scaling with your workloads.

At the time of publishing, AWS supports a broad range of security standards and compliance certifications, such as PCI-DSS, HIPAA/HITECH, FedRAMP, GDPR, FIPS 140-2, and NIST 800-171. We help to satisfy compliance requirements for most regulatory agencies around the globe.

Private and public-sector organizations, in some of the most security-sensitive verticals such as healthcare, banking, legal, and pharmaceutical, have trusted AWS to improve their security posture. Whether you're a small, medium, or large enterprise, or a public sector organization, there’s an AWS Partner with the right skills and experience available to help you move your business forward. AWS Partner specialists can help you find and connect with the right cloud partners aligned to your business needs. For more information, contact an AWS Partner specialist. To learn how customers around the world accelerate their cloud adoption and fuel innovation with the AWS Partner Network (APN), see Customer Success with AWS Partners.

Next steps

We recommend that you take the following next steps:

1. Learn more about specific migration and modernization scenarios. For more information, see Migrating Microsoft SQL Server databases to the AWS Cloud, Modernizing your application by migrating from an RDBMS to Amazon DynamoDB, and Choosing an approach for modernizing .NET applications.
2. Learn more about the organizational impact of large migrations. Large migrations are not only technology transformations but they also accompany changes to the roles, processes, and priorities of your organization. For more information, see Strategy and Best Practices for AWS Large Migrations.
3. Review the AWS for Microsoft Workloads Self-Study Guide.
4. Complete the Migrating Microsoft Workloads to AWS Hands-on Workshop.

Resources

Microsoft to AWS migration guidelines

- Migrating Microsoft Workloads to AWS: Self-Study Guide
- Migrating Microsoft Workloads to AWS: Hands-on Lab
- Migrating Microsoft SQL Server databases to the AWS Cloud
- Modernizing your application by migrating from an RDBMS to Amazon DynamoDB
- Choosing an approach for modernizing .NET applications
- Strategy and Best Practices for AWS Large Migrations.

General guidelines

- Windows on AWS
- Strategy and best practices for AWS large migrations
- Welcome to AWS Documentation

Videos

- AWS re:Invent 2020: Migrating Microsoft workloads to AWS
- Rehost Windows Workloads with AWS Application Migration Service - AWS Virtual Workshop

AWS Blog posts

- How to migrate on-premises workloads with AWS Application Migration Service
- Why you should migrate your Windows workloads with AWS (and how we can help)

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
Update	Added link to Migration Validator Toolkit PowerShell module. Clarified instructions for using the *Tutorial: Set up a Windows HPC cluster on Amazon EC2* on the *Migrating Windows failover clusters* page.	December 14, 2023
Update	Updated the *Migrating Windows failover clusters* page	December 8, 2023
Update	Updated list of supported instance types for Dedicated Hosts in the *Amazon EC2 Dedicated Hosts* section of the *Microsoft licensing on AWS* page	November 16, 2023
Update	Added complete list of supported instance families to the *Amazon EC2 Dedicated Hosts* section of the *Microsoft licensing on AWS* page	July 31, 2023
Update	Added BYOM guidance to the *Replatforming* section of the *Migrating SQL server* page	June 23, 2023
Initial publication	—	June 9, 2023

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. 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.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

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

active-passive migration

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

aggregate function

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

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

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

anti-pattern

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

application control

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

application portfolio

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

artificial intelligence (AI)

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

artificial intelligence operations (AIOps)

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

asymmetric encryption

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

atomicity, consistency, isolation, durability (ACID)

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

attribute-based access control (ABAC)

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

authoritative data source

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

Availability Zone

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

AWS Cloud Adoption Framework (AWS CAF)

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

AWS Workload Qualification Framework (AWS WQF)

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

B

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.

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.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

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

chaos engineering

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

CI/CD

See continuous integration and continuous delivery.

classification

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

client-side encryption

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

Cloud Center of Excellence (CCoE)

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

cloud computing

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

cloud operating model

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

cloud stages of adoption

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

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

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

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or AWS CodeCommit. 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

A field of AI used by machines to identify people, places, and things in images with accuracy at or above human levels. Often built with deep learning models, it automates extraction, analysis, classification, and understanding of useful information from a single image or a sequence of images.

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.

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 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.

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.

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.

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.

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.

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.

G

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.

greenfield strategy

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

guardrail

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

H

HA

See high availability.

heterogeneous database migration

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

high availability (HA)

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

historian modernization

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

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.

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 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.

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.

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*.

MAP

See Migration Acceleration Program.

mechanism

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

member account

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

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.

MPA

See Migration Portfolio Assessment.

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.

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.

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.

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.

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.

policy

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

polyglot persistence

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

portfolio assessment

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

predicate

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

predicate pushdown

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

preventative control

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

principal

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

Privacy by Design

An approach in system engineering that takes privacy into account throughout the whole engineering 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*.

production environment

See environment.

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.

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).

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 Managing AWS Regions in *AWS General Reference*.

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.

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.

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.

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 Secret in the Secrets Manager documentation.

security control

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

security hardening

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

security information and event management (SIEM) system

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

security response automation

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

server-side encryption

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

service control policy (SCP)

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

service endpoint

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

service-level agreement (SLA)

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

service-level indicator (SLI)

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

service-level objective (SLO)

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

shared responsibility model

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

SIEM

See security information and event management system.

single point of failure (SPOF)

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

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

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

SPOF

See single point of failure.

star schema

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

strangler fig pattern

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

subnet

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

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.

T

tags

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

target variable

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

task list

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

test environment

See environment.

training

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

transit gateway

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

trunk-based workflow

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

trusted access

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

tuning

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

two-pizza team

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

U

uncertainty

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

undifferentiated tasks

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

upper environments

See environment.

V

vacuuming

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

version control

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

VPC peering

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

vulnerability

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

W

warm cache

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

warm data

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

window function

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

workload

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

workstream

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

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

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

Z

zero-day exploit

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

zero-day vulnerability

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

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.