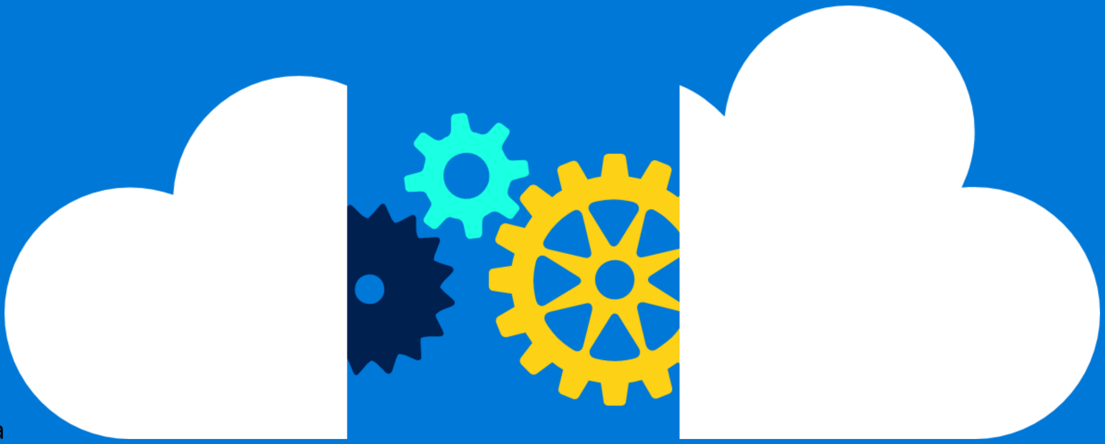
Microsoft



a

Migrate to Azure Stack Hub

Patterns and practices checklists

Azure Global CAT

February 2020

Contents

List of figures

Authored by Tiberiu Radu. Edited by Nanette Ray. Reviewed by Azure Global CAT.

The journey to Azure Stack Hub

Every organization has a unique journey to the cloud based on its history, business specifics, culture, and maybe most importantly their starting point. The journey to the cloud provides many options, features, and functionalities, including opportunities to improve existing governance and operations. It also offers the chance to implement new ones or even redesign older applications to take advantage of new cloud architectures.

This guide focuses on the specifics of migrating workloads to Azure Stack Hub and Azure, and the tools available today. At its core, Azure Stack Hub is an infrastructure as a service (IaaS) platform. It allows you to run Azure in your own datacenter. There are, however, some differences between Azure and Azure Stack Hub. Differences vary from the obvious ones such as the fact that Azure Stack Hub offers limited capacity when compared to Azure, to more nuanced differences such as the API versions and authentication mechanisms. Azure Stack Hub differs in who operates the cloud, which might sound obvious but has an impact on your workloads when you consider the end customer's experiences using IaaS versus platform as a service (PaaS) options and software as a service (SaaS) offerings. For example, you must consider which part of the Azure Stack Hub service that the Azure Stack Hub operator runs, because that determines whether the end customer calls a service PaaS or SaaS.

By following a consistent practice for cloud migration—both for Azure and Azure Stack Hub—teams can apply the same governance models to Azure Resource Manager resources. It's important to develop a good practice for governing a hybrid environment. Azure Stack Hub uses the same Azure Resource Manager model as Azure, although some services and features differ. Having realistic expectations from the beginning can ease the process of cloud migration and application modernization.

This guide focuses on the migration of existing applications that run either on physical servers or on existing virtualization platforms. By moving these workloads to the Azure Stack Hub IaaS environment, teams can benefit from smoother operations, self-service deployments, standardized hardware configurations, and Azure consistency, just to name a few. Using Azure Stack Hub to support application modernization becomes a key step in the journey to the cloud.

Audience

This guide is designed for Cloud Solution Providers (CSPs), Microsoft Service Providers (MSPs), system integrators, and any enterprise considering how to migrate workloads to Azure and Azure Stack Hub. The intent is to give the perspective of a consultant and architect. This guide addresses two types of audience:

- **Team doing the migration.** This term refers to a service provider or an enterprise IT acting as an internal service provider.
- **Customer.** This term refers to a customer using Azure Stack Hub services, which covers a mix of organizational roles. The customer can be an application owner, a business group owner, an organization to be onboarded, or a project team responsible for any of these services.

The framework presented in this guide applies to both teams and customers and is intended as a starting point from which you can build your own migration practice.

About this guide

This guide draws from a blog series written by the Azure Stack Hub product team. Together, these articles give you an overarching perspective on Azure Stack Hub and required migration steps:

- Azure Stack IaaS – part one: Azure Stack at its core is an IaaS platform
- Azure Stack IaaS – part two: Start with what you already have
- Azure Stack IaaS – part three: Foundation of Azure Stack IaaS
- Azure Stack IaaS – part four: Protect your stuff
- Azure Stack IaaS – part five: Do it yourself
- Azure Stack IaaS – part six: Pay for what you use
- Azure Stack IaaS – part seven: It takes a team
- Azure Stack IaaS – part eight: If you do it often, automate it
- Azure Stack IaaS – part nine: Build on the success of others
- Azure Stack IaaS – part ten: Journey to PaaS

In addition, many of the patterns described in the next chapters are based on the Cloud Migration and Modernization playbook. It focuses on migrating workloads to Azure or modernizing legacy applications in the cloud and is intended for business and technical leaders. The playbook provides an A-to-Z list of best practices that start with defining your strategy and training your people, and it also covers migration and optimization practices.

This guide also draws from a series of insights created by the Microsoft Partner Network team called the Cloud Practice Development Playbooks. These e-books are freely available to download and include technology and organizational best practices from more than a thousand Microsoft partners.

The migration processes

Most organization follow a similar three-phase migration process:

Assess	In the assessment phase, your team uses a mixture of software tools and consultancy best practices to discover: • The applications to migrate. • Current configurations of the applications. • People within your customer base to be impacted by the migration. • Dependencies of each application. The output of your assessment includes a comprehensive plan describing what to do with each application and the expectations about availability and functionality.
Migrate	The migration phase acts on the recommendations of your assessment plan. The steps usually include: • Setting up Azure and Azure Stack Hub subscriptions using best practices for security, connectivity, policies, and general governance before you migrate. This ensures that your customers are using Azure and Azure Stack Hub environments correctly from the start. • Performing the migration using the prescribed method identified in the assessment plan: rehost, refactor, rearchitect, rebuild, or retire. • Evaluating and testing to ensure the migrated applications meet the criteria outlined in your assessment plan.
Optimize	The optimization phase implements Azure security and management resources for governing, securing, and monitoring your cloud applications in Azure. This phase also assesses opportunities to optimize spending. Common tasks include: • Reviewing Azure cost management to track spending and identify areas for cost savings. • Evaluating migrated applications for opportunities to right-size overprovisioned virtual machines and services. • Implementing automation to resize or stop based on a utilization schedule. • Identifying applications that could benefit from optimization with PaaS options or containers.

You can learn more about each of these phases in the Cloud Migration and Modernization Playbook and the Operations and Management Playbook.

As you expand each of these phases, your organization begins to develop a cloud-first strategy comprised of a set of principles, models, disciplines, concepts, and guidelines for delivering services. These proven practices follow the disciplined approach of the Microsoft Solution Framework, shown in the following figure and described in the remaining sections in this guide, and include these processes:

- **Envision, discovery, and assessment.** Adopt a cloud-first strategy. Map and classify applications and data end-to-end. Collect and categorize applications based on the migration strategy.
- **Implementation.** Start the migration journey based on the path determined for each application and dataset. Identify the different tools needed for the various workloads.
- **Operationalization and modernization.** Continuously evaluate performance, availability, and utilization metrics to reduce cost and improve the user experience. Continuously reevaluate migrated applications to drive the next set of changes to towards 100-percent modernization.
- **Decommission.** Accelerate the retirement of retained applicatoins that are not in scope for modernization.

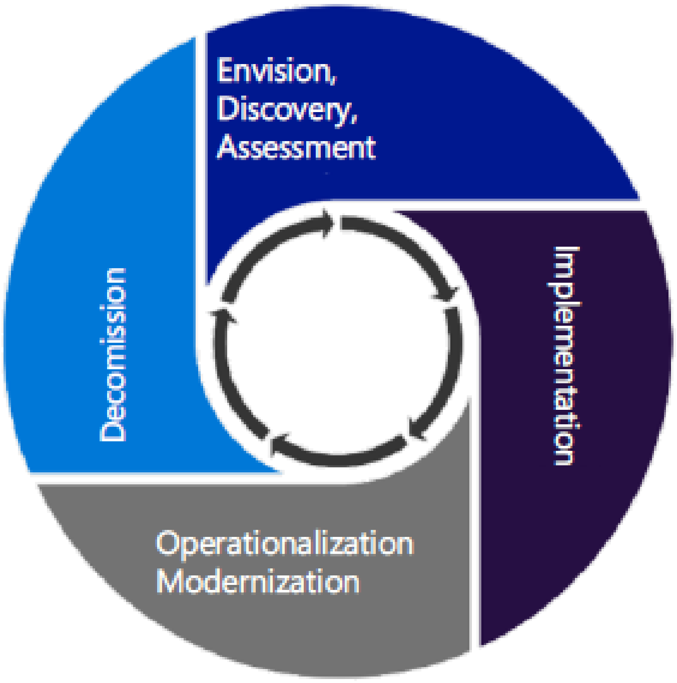


Figure 1. The Microsoft Solution Framework describes proven practices in the software development lifecycle (SDLC).

Envision

The envision phase is your opportunity to establish the level of an organization's maturity and familiarity with the cloud and cloud design principles. An important first step to identify where you are in the cloud journey. This understanding sets the tone for the subsequent planning steps required to migrate and modernize applications and data.

Microsoft takes a cloud-first approach. The priority is to move applications and data on one or more of the hyper-scale clouds, including the public Azure option or a sovereign, locale-specific cloud such as Azure China, Azure Germany, or Azure Government. Azure Stack Hub acts as another instance of a sovereign cloud, operated by customers in their datacenters or consumed through a cloud service provider. However, Azure Stack Hub is not a hyper-scale cloud, and Microsoft does not publish or support any service-level agreement (SLA) on Azure Stack Hub.

Microsoft has a breadth of experience with migration projects from direct work with customers in one-on-one discussions, executive briefings, and Envision sessions. Based on this experience, we recommend taking the time during the planning phase to set expectations appropriately about Azure and Azure Stack Hub, which helps avoid pitfalls or setbacks later in a project. The key to a successful implementation is a good understanding of when to use Azure and when to use Azure Stack Hub as well as the differences between them.

As Figure 2 shows, the triggers of a migration and modernization effort vary. Success depends on a thorough understanding of all these drivers and implementing best practices.

Envision stage: Desired outcome checklist

- ✓ Framework showing analysis of TCO and ROI, including capacity planning and pricing model.
- ✓ Scenario-based dependency map of desired IaaS, PaaS, and SaaS offerings.
- ✓ Data and application placement framework (cloud-first, hybrid, air-gapped).
- ✓ Leadership-level agreement on migration and modernization strategy and level of investment for each option (retain, retire, rehost, replatform, repurchase, refactor).

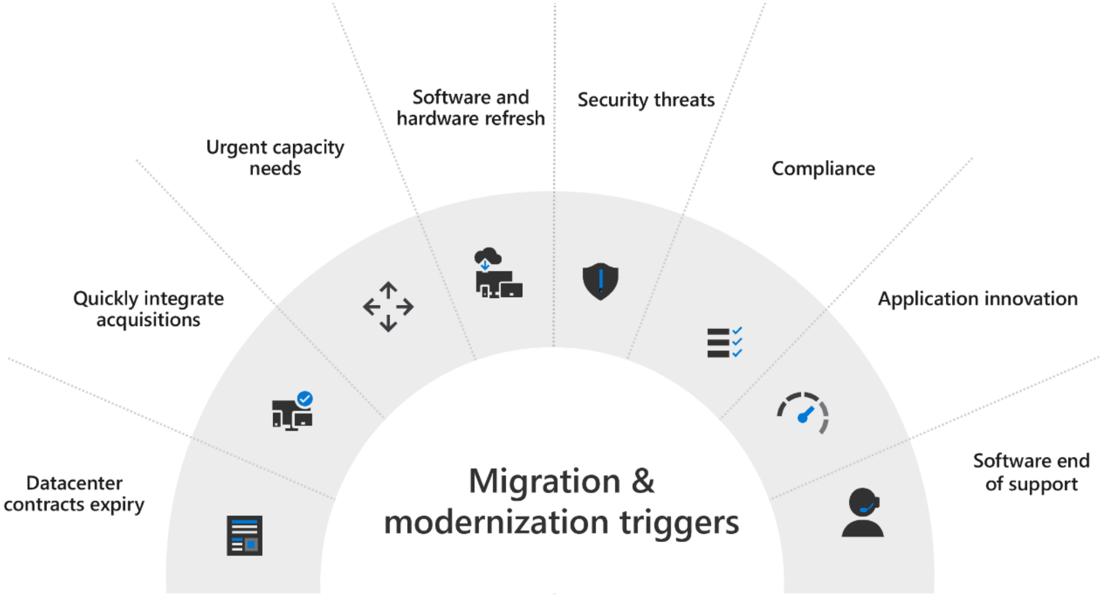

Software and hardware refresh
Security threats
Urgent capacity needs
Compliance
Quickly integrate acquisitions
Application innovation
Datacenter contracts expiry
Migration & modernization triggers
Software end of support

Figure 2. Drivers of modernization and migration efforts.

Resources that can help in the envision phase include:

- Azure Stack overview and roadmap
- Azure Stack Hub capacity planning
- Azure Stack Hub datacenter integration walkthrough
- Azure Stack Hub VM features

1. Identify stakeholders and influencers

- Identify the owners and decision makers that have direct influence over the application and data within the organization.
- Differentiate between stakeholders that embrace the cloud versus those who will limit or block adoption.
- Understand drivers and motivators for each.

2. Do a cloud reality check

- Understand the customer’s cloud journey and where they are in that journey.
- Determine the motivation to keep applications and data on-premises, such as regulatory requirements, data gravity, or compliance needs.
- Identify critical services that the organization expects to use over time and their desired fault tolerance and availability.

3. Establish business intent

- Is it tactical cost savings—reducing the datacenter count or licenses?
- Is it strategic transformation?
- Is it exploratory?

4. Review compliance, regulatory, and privacy restrictions

- Which regulations impact the customer’s cloud adoption?
- How will data privacy impact data migration?
- What is the impact of internal compliance groups on cloud users?

5. Define success

- Has the customer defined end results of migration and modernization?
- Which Rs are in scope as part of the migration and modernization taxonomy? (More on the Rs in the Assessment section.)
- Is migration defined as a modernization journey? If so, have you defined the next steps and the goals required after the migration?
- Review the SLA, Recovery Point Objective (RPO), Recovery Time Objective (RTO), and availability requirements.

Discovery

In the discovery phase, the focus shifts to developing a framework to identify the applications and datasets within the organization, categorize discovered assets, identify stakeholders, and creating a decision matrix to enter the assessment phase. The organization needs to determine the appropriate categories and prioritization of the short-term gains associated with a simple lift-and-shift move versus the long-term investments of modernization.

The discovery phase promotes the benefits of migration and modernization and streamlines the decision-making process during the assessment phase.

A critical step to ensuring the thoroughness of the discovery phase is to identify the stakeholders who have strategic control over the applications and datasets. These people are responsible for developing a migration and modernization plan for their applications and datasets. The opportunity is to educate stakeholders on the technical and business benefits of migration and modernization.

A variety of tools, information sources, maps, and stakeholders are involved during discovery. Each organization has its own approach. It's important to have standardized records of ownership for deployed assets and to keep controls in place to keep them updated. Otherwise, you won't be able to trust the quality and accuracy of the information collected during discovery.

Resources that can help in the discovery phase include:

- Azure Migrate includes useful assessments even though it does not fully support Azure Stack Hub as a target at this time. For example, with the Cloudbase Coriolis tool, the Azure Migrate assessment is used to create a migration environment for an entire application.
- Corent SurPaaS Platform is an Azure SaaS service that enables you to automate the process of assessing, planning, and cost modeling your customers' workloads, then migrating them to the cloud.
- Service Map identifies dependencies and creates a map for a solution before moving it. Once migrated, you can use the same solution on Azure Stack Hub to test network metrics and compare to the indicators before the move.

Discovery phase: Desired outcome checklist

- ✓ A plan to keep the map of stakeholders to applications and data sets updated and accurate. Will there be an auditing process? Self-identified or programmatic?
- ✓ A decision matrix that clearly identifies the organization's preferred migration and modernization path and that identifies the conditions for granting exceptions or exemptions.
- ✓ An assessment of the cost of migration, modernization, and maintaining exceptions and exemptions.
- ✓ A plan that tracks exceptions, exemptions, and forcing functions to ensure applications and datasets conform.
- ✓ An understanding of stakeholder assumptions about migration, modernization, and goals so you can minimize confusion.
- ✓ A framework that specifies where applications and datasets go—public cloud, sovereign cloud, service provider cloud, or on-premises cloud.

1. Identify sources of information

- Qualifying the accuracy of the sources.
- Plan to update information sources.

2. Automated discovery and crawling

- Determine tool selection criteria.
- Identify critical points and potential bottlenecks.
- Choose continuous or targeted discovery and define how long the monitoring will last.

3. Manual discovery

- Specify an information validation process.
- Correlate with automated discovery and identify gaps.

4. Isolate and categorize dependencies

- Identify the dependencies that make up the core applications and datasets.
- Develop ways to clearly isolate an application and data from its dependencies, especially for infrastructure shared services.
- Develop categories for applications, data sets, and infrastructure services.
- Use categories to map regulatory, compliance, and privacy requirements.

5. Assess impact on business continuity

- Determine the acceptable tolerances for downtime that can be associated with each category or application and dataset.

Assessment

In the assessment phase, customers make data-driven decisions about the applications and datasets to target for migration and modernization. These decisions must reflect an organization's transformation goals. Customers must coordinate with internal stakeholders to determine the optimal path for an application or dataset and be willing to question decisions and assumptions as additional data is presented.

Assessment is also the time to develop a readiness plan for operations and production. In this phase, you must also consider the tooling and automation needed to speed migration and modernization efforts and budget accordingly.

Sometimes applications appear to be obvious candidates for migration, but the support and licensing issues get in the way. Other times, a significant upfront investment in development and testing is needed. The assessment phase is when customers define downtime and data loss tolerances. As the acceptable amount of downtime and data loss approach zero, customers need to increase their investment to complete the transformation. They often need help navigating these decisions as they work on parallel efforts to meet short-term and long-term goals.

Resources that can help in the assessment phase include:

- Cloud rationalization
- Cloud Migration Essentials e-book
- Ready: Recommended naming and tagging conventions
- "Migrating to Virtual Machines" chapter of Cloud Migration & Modernization (PDF).
- Start with what you already have blog
- Azure enterprise scaffold: Prescriptive subscription governance—applies to both Azure and Azure Stack Hub resources

Assessment phase: Desired outcome checklist

- ✓ Deep technical understanding of the foundational elements of Azure and Azure Stack.
- ✓ Architecture and operational plan for shared/common services—networking, infrastructure capacity, identity, security, storage.
- ✓ Architectural guidance for users, including how to integrate applications with shared services.
- ✓ Clearly established SLAs among stakeholders.
- ✓ Identification of existing services to incorporate alongside Azure or Azure Stack services, such as Active Directory, Azure Active Directory, Exchange, SharePoint, and Office 365.
- ✓ Plan for new features and change management in both Azure and Azure Stack.

1. Map an application or data set to the five Rs

During assessment, a set of categories called the five Rs is used for cloud rationalization. These categories are used to evaluate assets and determine the best way to migrate or modernize each application and dataset. The goal in cloud rationalization is to set a baseline, not to rationalize every workload.

- **Rehost.** Also known as "lift and shift," this strategy entails migrating your physical servers and virtual machines (VMs) to the cloud just as they are.
- **Refactor.** This strategy involves using additional cloud provider services to optimize cost, reliability, and performance by refactoring your applications.
- **Rearchitect.** This strategy involves rearchitecting an application to modernize it—that is, to transform it with a modular architecture.
- **Rebuild.** This strategy involves revising the existing application by aggressively adopting PaaS or even software as a service (SaaS) services and architecture.
- **Replace.** This strategy involves moving or discarding an existing application and replacing it with commercial software delivered as a service.

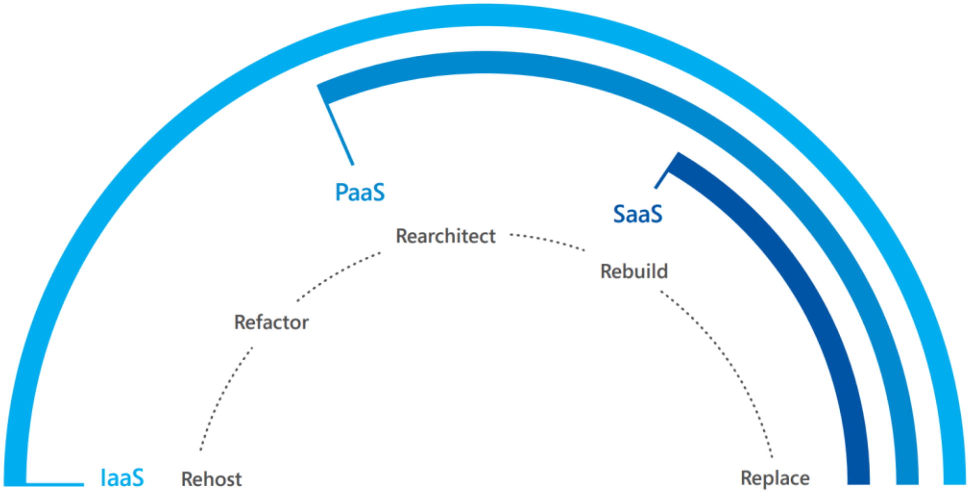


Figure 3. The five Rs of cloud rationalization help you assess applications.

2. Determine the placement of an application

- Public cloud
- Sovereign cloud
- Service provider cloud
- On-premises cloud

3. Quantify the value to the business and stakeholders of migrating or modernizing an application or dataset

- Less friction—short-term focus, limited long-term viability.
- More friction—long-term investment, easier to iterate and continue to modernize.

4. Assess impact of regulatory, compliance, and privacy requirements

- Which data can reside on Azure, and which data needs to stay on-premises?
- Who can manage the underlying platform?
- Is the data dependent on the location?
- Are there expiration dates for storing the data?

5. Assess infrastructure dependencies that must be addressed first

- Identity
- Connectivity
- Security
- Encryption

6. Determine success metrics and availability tolerances

- Performance
- Availability
- Resiliency
- Deployability

7. Assess impact of licensing and support

- Are there product licensing restrictions that will limit transformation?
- Is the application or dataset supportable in the new environment?
- Are there independent software vendors (ISVs) involved with the customer that need to be contacted to confirm support statements?

Implementation

In the implementation phase, customers have a plan and a schedule that drive the execution of the migration and modernization effort. This phase primarily focuses on the effectiveness and efficiency of the tools, techniques, and processes used to bring the project to completion.

During implementation, customers build expertise in cloud design principles, whether they're rehosting, rearchitecting, or rebuilding. This is the time to invest in the practices that will drive the customer's long-term cloud journey.

1. Create an implementation schedule

- How is the schedule developed? Who are the key stakeholders needed to approve the timeline and any changes to the plan of record?
- Set up sprints for the migrations. Have teams work with the customers to ensure that changes are communicated correctly to stakeholders, product owners, and any users who might be impacted by changes.

2. Use multilayered validation

- Validate each step in the workflow to ensure that an application is progressing towards the desired end state.
- Determine who will design and implement a comprehensive validation plan that identifies all the metrics of success.
- Are the validation goals and dependencies clear enough so all planners understand the risk to the schedule when workstreams start to fall behind?

3. Migrate according to the five Rs

Based on the map created during the assessment phase, begin rehosting, refactoring, rearchitecting, rebuilding, or replacing applications.

Rehost

In manual rehosting, the expectation is that the software can be installed and configured in the new environment—Azure or Azure Stack Hub. Options for rehosting include:

- **Using commercial off-the-shelf (COTS) products.** This is an opportunity to work with ISVs to provide Azure Resource Manger templates, VM extensions, and automation that will govern the life cycle of the application.

Implementation phase: Desired outcome checklist

- ✓ Continuous evaluation of application landscape to identify new opportunities for migration or modernization.
- ✓ Increase scrutiny of lift-and-shift application that will simply be retained and never modernized.
- ✓ Comprehensive validation plan that clearly defines success.
- ✓ Develop the skillset to accurately map out application dependencies across different deployment topologies, including hybrid services and hybrid networking.
- ✓ Clear articulation of the application's availability, resiliency, and performance requirements.
- ✓ Investments in attaching operational modernizations to each application - automation, monitoring, security, compliance, and data protection.

- **Working with an ISV.** There's no installer package to help customers navigate the deployment of an application in a public or hybrid cloud, so working with an ISV can help. ISVs can take advantage of the same infrastructure-as-code methodologies that customers implement for their own applications. Over time, customers invest in higher levels of automation, gradually automating the deployment, servicing, scaling, and remediation of an application in a cloud context.
- **Developing applications.** Customers with a development team can build business applications, automate scripts, and have more control over the life cycle of a cloud application. If they already invested in a code repository and in an integration pipeline, Azure or Azure Stack Hub simply become one more target to support.

Manual rehosting is an ideal time to make sure customers account for the full life cycle of an application. For example, in addition to deployment, customers can incorporate monitoring, remediation, servicing, backup, replication, and other workflows that ensure the application is ready for production.

By comparison, an automated approach to rehosting is needed if a customer wants to migrate software but no longer has the original deployment packages. The software can be migrated as-is, including the hosting operating system. Considerations include:

- **Long-term support.** If an ISV is no longer available or stopped development work on an application, supporting it will add to the customer's technical debt in the datacenter.
- **Cost reduction.** Rehosting is an opportunity to reduce the cost and overhead of operating an application in production. A lift-and-shift migration does not need to end after the application is restored to production. Customers should make the most of the new platform's underlying services and enable the infrastructure-as-code benefits of the cloud. This is especially true in a hybrid environment where the enhanced capabilities can be sourced directly from the cloud.

Refactor

This path introduces optimizations to the application without changing the core architecture. For example, updating the underlying operating system the application is installed on to avoid paying extended support costs or migrating a database from a stand-alone server to a fully managed PaaS offering.

These targeted changed must not compromise the customer's availability, resiliency, and performance requirements. Targeted platform changes must take into account the actual services available on Azure or Azure Stack Hub. It is necessary to understand the implications of different deployment topologies from fully and reliably connected to fully air-gapped.

Rearchitect

Some aging applications aren't compatible with cloud providers because of the architectural decisions that were made when the application was built. In these cases, the application might need to be rearchitected before transformation.

Customers can choose to keep an application in production as-is. However, having no plans to migrate or modernize an application adds long-term technical debt, and the cost should be passed on to the application owners. Assess whether the customer has a plan to incentivize the application owners to accelerate the retirement of these applications.

Rebuild

Rebuilding presents an opportunity to redesign an application and create a new code base that aligns with a cloud-native approach. Rebuilding may require a significant upfront investment to properly design, implement, test, validate, and operationalize an application. The amount of effort really depends on a customer's maturity in cloud technologies and an application's complexity and dependencies.

- Customers may use rebuilding as an opportunity to use advanced cloud services, and the new dependencies can result in a hybrid deployment. Does the customer have enough experience with these types of deployments to properly support the application? Has the customer developed a risk profile to assess the different failure domains and mitigation strategies for each?
- It's important to have a change control process in place to avoid scope creep and to avoid changes to the plan of record.

Replace

The decision to replace an application is a great opportunity to reevaluate alternatives, such as COTS products or equivalent SaaS offerings. The assessment must take into account the features and capabilities that customers depend on and evaluate functionality gaps. Replacement can mean retiring an application altogether.

- In the context of Azure and Azure Stack Hub, is there a need to integrate the new SaaS offering from the public cloud with an on-premises services like Active Directory?
- What are the networking implications of supporting a hybrid topology with secure connectivity over VPN or ExpressRoute?
- Is there an opportunity to attach operational life cycle services—for example, monitoring, security, and backup—to the SaaS or new COTS application?
- Retiring a standalone application is a simple decision. Other applications may have unexpected interdependencies that require you to retain the application longer, refactor it, or rebuild it to break the dependency. The goal should be to avoid unplanned downtime by completing the dependency map between applications and services.

Operationalization

In the operationalization phase, the goal is to ensure that applications are in a sustainable production environment. Declaring an application ready for production depends on the customer's requirements.

The operationalization phase presents an opportunity to help the customer evaluate how they manage their production applications and integrate new capabilities provided by Microsoft. It is important to identify which cloud services they are open to integrating with on-premises applications. Some customers are willing to back up and replicate their applications and data to Azure for the added layer of protection without requiring a secondary site to be built.

Optimizing the operational experience ensures that customers can take full advantage of the cloud. Even the simplest migration experience can be elevated to include some modern design principles.

Resources that help during the operationalization phase include:

- Understanding architectural patterns and practices for business continuity and disaster recovery on Microsoft Azure Stack
- Protect your stuff blog
- Azure Stack considerations for business continuity and disaster recovery white paper

**Operationalization phase:
Desired outcome checklist**

- ✓ Plan that incorporates the services identified by the customer as required for the application to go into production.
- ✓ Clear understanding of the dependencies created by these services, especially if they run in the cloud (hybrid).
- ✓ Metrics for cost, availability, and performance that map to business objectives.

1. Assess tolerance for hybrid cloud services

- Check the availability of reliable connectivity between on-premises and public Azure datacenters.
- Check the sufficient bandwidth and tolerable latency for critical data protection services.
- Do cost analysis of integration and scaling with cloud-based monitoring, log collection, security and compliance scanning, and performance analysis services.
- Perform gap analysis of existing offerings and potential replacements.

2. Set up hybrid, cloud-aware management

- Avoid duplication of effort and optimize SLA by examining the correlation between IT-managed cloud services and application-specific services.
- Consider the automation required to orchestrate provisioning of services during deployment and migration of applications.

3. Assess applicability of available services

- Monitor system health and operational status and performance using well-defined metrics that form the basis of the SLAs guaranteed to the end-user.
- Check security and compliance, asking how well the cloud environment meets the regulatory and compliance requirements imposed by the application.
- What are the processes for backup/restore and replication/failover?
- Find data protection services for IaaS, PaaS, and SaaS resources.
- Incorporate multiple vendors, technologies, and capabilities to achieve a comprehensive protection strategy.
- Use pricing and cost analysis.

4. Map cost estimates to actual consumption and projections

Optimization

In the optimization phase, the application and datasets running in their new environments need continuous monitoring to ensure all the critical indicators of success are being met. If the data used originally to guide sizing requirements was incomplete, or if the new environment introduces unexpected latencies, the optimization phase is the time for improvement. Continuous right-sizing across multiple dimensions reinforces the benefits of migration.

Resources that help during the optimization phase include:

- Azure Monitor on Azure Stack Hub
- How to consume monitoring data from Azure Stack Hub
- Pay for what you use blog

Optimization phase: Desired outcome checklist

- ✓ Customer has a plan to evaluate sizing in a continuously.
- ✓ Customer can support friction-free right-sizing with minimal downtime.

1. Right-size the environment

It is important to collect as many data points as possible about a migrated application, including how well it is performing, the utilization rates and trends, and breaches in SLAs resulting in unplanned downtime.

This data can be used to further optimize the application environment and refine the decisions made during the assessment phase. Maybe it is necessary to resize the VM to accommodate for more network throughput or adjust storage to maximize for peak IOPS. There can be datasets that rarely need to be accessed that can be migrated to a different tier of storage. Maybe there is a need to up level to a different offering to improve availability and resiliency.

2. Contain costs and reallocate

In a cloud environment, the customer may be shifting from traditional capital expense (CAPEX) funding models to an operational expense (OPEX) model. The cost of running and supporting an application in the cloud needs to be continuously monitored to minimize unnecessary or overprovisioned services. This is especially true if the OPEX costs get passed directly to the application owners. Application owners may not be used to getting a fluctuating bill from month to month. Application owners and IT need to take into consideration large migrations to the cloud which will spike charges.

3. Enable and disable services

A risk-averse stance can limit the original scope of a migration project. Over time, applications need to be amended with additional capabilities. In other cases, services that were deemed important at one point in the journey may no longer be required or get phased out due to newer offerings. In either case, customers should make changes to their production deployments in a structured way using automation and templates. Any changes to the services surrounding an application need to be captured and audited to ensure there is a reliable source of truth.

Modernization

As an extension of Azure, Azure Stack Hub differs from traditional virtualization platforms by offering PaaS options in addition to IaaS capabilities. With Azure-style infrastructure, you can use Azure Stack Hub to add value and support the following scenarios:

- Scalability and availability services such as pay per use, VM availability sets, Virtual Machine Scale Sets, network adapters, and the ability to add and resize VMs and disks.
- Self-service virtual networks that include firewall rules, load balancing, VPNs, routes, DNS, and a marketplace of appliances from other vendors.
- Organizational services such as resource groups, role-based access control (RBAC), change auditing, locks, and tags.
- Self-service support for your customers, including boot diagnostics, screenshots, serial logs, and metrics.
- Guest management, including VM extensions, the ability to run custom code, software inventory, and change tracking.
- Security services, including default firewalls, restrictions, VM updates and patch management, and malware status.
- DevOps options, including infrastructure as code (IaC), a portal with PowerShell and command-line interface (CLI), Azure Application Insights, and integration with Azure DevOps and Jenkins.
- Storage capacity, including the ability upload and download and also capture and deploy VM images.
- Business continuity through VM backups and disaster recovery options.

Modernization phase: Desired outcome checklist

- ✓ A plan to attach hybrid cloud services to lift-and-shift applications
- ✓ Monitoring
- ✓ Security
- ✓ Data protection
- ✓ Log analytics

In the modernization phase, customers have an opportunity to retrofit a migrated application in production with additional cloud services. Many customers who start out migrating applications and datasets using a lift-and-shift approach begin to modernize in this phase or optimize to reduce friction and accelerate migration. Customers who postponed the decision to add cloud services may reexamine that choice and looks for ways to modernize. If they opted to retain an application as-is after migration, they may now decide to retire it.

This phase presents an opportunity to introduce modernizations without changing the design of an application or requiring another migration to a new platform. Modernization can start with something small, such as creating a template for particular VMs in a solution as the first step in an IaC deployment and management approach. Eventually, entire environments can be managed through declarative and imperative scripts.

Resources that can help during modernization include:

- Azure Stack QuickStart Templates GitHub repository
- Azure QuickStart Templates GitHub repository
- What is Infrastructure as Code?

1. Follow a continuous migration and modernization cycle

- Look for optimization opportunities. Even if the original intention of migration was to accelerate the decommission of a legacy environment, or if a customer took short cuts to get the cloud, operational modernizations can be introduced to production applications after migration.
- Get comfortable with a continuous cycle of planning, discovery, and assessment to maximize the benefits of modernization efforts and minimize the population of retained applications.

2. Develop customer cloud maturity

Identify opportunities for ongoing education about cloud design principles.

3. Build skill sets to continuously retrofit applications with cloud services

Until the next migration and modernization cycle, existing applications can be operated efficiently using cloud services. This is an ideal opportunity to extend cloud services into an existing environment.

4. Optimize for recoverability

- Protect applications and data. If an application ever needs to be redeployed or recovered due to a disaster or an unplanned outage, it is not enough to protect the VMs or datasets. Complimentary services used during the operational life cycle of an application need to be correctly provisioned and attached before the application is ready to go back into production.
- Automate and templatize. Recovery efforts may require targeting a different environment. The automation and templates need to be flexible enough to account for these types of changes and remove any possibility of introducing human error into the deployment recovery workflow.

Decommission

In the decommission phase, the customer develops a retirement plan for migrated applications that they plan to retain but not modernize or migrate. The goal is to accelerate retirement and expose the ongoing operational cost of these applications.

Decommissioning is the time to quantify the ongoing supports costs for legacy applications. The process is similar to how many organizations require approval and justification to deploy a physical server instead of a virtual server. Application owners are held to a time-bound retirement plan.

Decommission phase: Desired outcome checklist

- ✓ After migration, the customer continuously reevaluates their applications to identify retirement candidates.
- ✓ The customer uses support costs or complexity to drive application owners to accelerate retirement of applications.

1. Quantify the ongoing cost of supporting legacy applications

Migration and modernization efforts set a new bar for the cost of supporting an application in production. This baseline can be used to extrapolate the cost of supporting legacy applications that do not take advantage of any new capabilities in the cloud. In some cases, the application itself may have additional support costs from the ISV that developed it.

2. Qualify the skill set required to maintain legacy applications

Retaining applications requires personnel trained in legacy requirements and protocols. Over time, the risk of knowledge gaps tends to grow, leading to unplanned downtime.

3. Accelerate the pace of retirement

- Give application owners incentive to implement a retirement plan and adhere to a reasonable timeline. The incentive can come in terms of educating the application owner on ways the cloud can help them achieve their business objectives.
- Have application owner pay up front for the fully loaded cost of supporting legacy application in production over a set period of time. Make the cost transparent—do not bundle the cost of supporting these legacy applications with the cost of operating modernized applications.

Resources for migration

Any number of tools and resources are needed during a migration. Many customers choose to work with a partner who specializes in the cloud. The Microsoft partner ecosystem includes ISVs offering a range of solutions—from simple migrations to full-service migrations. Microsoft also provides migration options that require manual steps to implement but often at a lower cost. For more information, see the Azure Stack IaaS – part two: Start with what you already have blog.

To prepare for database migration and learn about the tools and options for Microsoft SQL Server workloads, see Migrate SQL Server to SQL Server on Azure Virtual Machines. This guide applies to Azure Stack Hub migrations and can be used as reference towards ensuring a successful migration.

Several articles describe how Azure Stack Hub extends Azure services to help customers unlock new hybrid cloud innovations, such as intelligent edge and disconnected scenarios, regulatory and data sovereignty, or application modernization. The Azure Stack Hub User Documentation discusses many hybrid patterns and highlights the developer options for building cross-cloud scaling solutions and creating a geo-distributed application solution.

Other materials that can help your teams ramp up include the Azure Stack Development Kit and the Azure Stack developer forum, and Microsoft Learning Course 20537B: Configuring and Operating a Hybrid Cloud with Microsoft Azure Stack.