Azure Stack Migration Practice and Patterns

azseospatterns@microsoft.com

Contents

[Introduction 1](#_Toc8670296)

[Training resources 2](#_Toc8670297)

[The migration processes 2](#_Toc8670298)

[Envision 4](#_Toc8670299)

[Discovery 6](#_Toc8670300)

[Assessment 8](#_Toc8670301)

[Implementation 10](#_Toc8670302)

[Operationalization 14](#_Toc8670303)

[Optimization 15](#_Toc8670304)

[Modernization 17](#_Toc8670305)

[Decommission 19](#_Toc8670306)

[The Migration Tools 20](#_Toc8670307)

# Introduction

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, functionalities, as well as opportunities to improve existing governance, operations, implement new ones, and even redesign the applications to take advantage of the cloud architectures. We do not plan to reiterate all these options, instead focus on the specifics of migrating workloads to Azure Stack and Azure, and the tools available today.

The Microsoft Partner Network team has crafted a series of insights and best practices from over 1000 Microsoft Partners. They are collated as “Cloud Practice Development Playbooks” and found here <https://aka.ms/practiceplaybooks>.

The Cloud Practice Development Playbooks also include the [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook. This playbook is intended for the business and technical leadership for new and existing Microsoft partners focused on migrating workloads to Microsoft Azure or modernizing legacy applications to the cloud. It covers best practices from defining your strategy, hire&train, skills&exams required, all the way to the migration and optimization best practices. Many of the patterns described in the next chapters are based on this playbook.

Several articles describe how Azure Stack extends Azure services to help [customers unlock new hybrid cloud innovation](https://azure.microsoft.com/en-us/blog/customers-are-using-azure-stack-to-unlock-new-hybrid-cloud-innovation/), whether in Edge & Disconnected, Regulatory & Data Sovereignty, or Application Modernization. There are also a number of [Hybrid Patterns](https://aka.ms/azsdevtutorials) we’ve built to highlight the options available to developers when creating applications on Azure Stack – whether it’s apps built on Azure Stack that offer [cross-cloud scaling solutions with Azure](https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-solution-cloud-burst), or [a geo-distributed app solution with Azure and Azure Stack](https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-solution-geo-distributed).

The “[Azure Stack at its core is an IaaS platform](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-one/)” blog series describes the ways in which you can modernize your operations after you move the workloads to Azure Stack.

In this article we will focus on the migration of existing applications that live on either physical servers, or existing virtualization platforms and how moving these workloads to Azure Stack IaaS will enable the benefits around [Operations](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-seven/), [Self-Service](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-five/), [Azure consistency](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-3/), as well as enable scenarios for modernization of these applications in the journey to the cloud.

This article is targeted at both Service Providers (CSP, MSP, SIs) as well as enterprises that look to migrate workloads on Azure and Azure Stack. In this article, there are mainly two sets of actors:

* The team doing the migration – this could be the Service Provider, or the IT team in the Enterprise that acts as an internal Service Provider
* The customer – the intent of this article is to have a Consultant and Architect mindset, which also means it addresses the end-user as a *customer*, but in reality the end-user could be the application owner, the business group owner, the organization that will be onboarded, or the actual project team responsible for those services. They are all labeled as “customers”, though in reality there is a strong mix of organizational personas.

The framework applies to both and is usable in various scenarios. It’s intended as a starting point from which you would build your own migration practice (check the [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook for even more context around the migration practice).

# Training resources

Azure Stack at its core is an IaaS platform. It allows you to run Azure in your own datacenter. There are some differences between Azure and Azure Stack, though. Differences vary from the obvious ones such as the fact that Azure Stack offers limited capacity when compared to Azure, to more nuanced differences such as the API versions and authentication mechanism. Azure Stack differs in who operates the cloud, which might seem like a truism, but has an impact on your workloads when talking about IaaS, PaaS, and SaaS for the end-customer. For example, you will need to answer which part of the Azure Stack service does the Azure Stack Operator runs in order for the end customer to call a service PaaS or SaaS.

These differences between Azure and Azure Stack may result in assumptions about the Azure Stack product that could easily lead to unreasonable expectations. When it comes to migration and modernization, you do not want bad assumptions to be the foundation of an engagement.

These are the online materials that can help your teams ramp up on the basics and the more advanced features of Azure Stack:

* [Azure Stack documentation](https://docs.microsoft.com/en-gb/azure/#pivot=products&panel=stack)
* Downloaded and try the Azure Stack Development Kit (<http://aka.ms/asdk>)
* Free online Azure Stack course INF240x (self-paced with 40+ hours content + ASDK-based labs) (<http://aka.ms/AzSMooC>)
* Azure Stack at Microsoft Ignite 2018 [sessions](https://azure.microsoft.com/en-us/blog/come-check-out-azure-stack-at-ignite-2018/)
* Fee-based in-classroom instructor-led Azure Stack official training course 20537B (<http://aka.ms/AzSMOC>)
* Azure Stack MSDN forum (<http://aka.ms/AzSForum>)

# The migration processes

Every organization is different. As a result each migration process is different and has its own specific challenges. At a high level, these can be broken down into three key phases:



**Assess**

The assessment phase is where your team will use a mixture of software tools and consultancy best practices to discover what applications can be migrated, what the applications current configurations are, the people within your customer that will be impacted by the migration, as well as identify the dependencies of each application. The output of your assessment will include a comprehensive plan for what to do with each application and the expectations on availability and functionality.

**Migrate**

The migration phase is when the recommendations in your assessment plan are put into place. The following steps are usually taken.

* Setup Azure and Azure Stack subscriptions using best practices for security, connectivity, policies and general governance before you migrate to ensure your customers are using Azure and Azure Stack environments correctly from the start.
* Perform the migration using the prescribed method identified in the assessment plan: rehost, retire, replace, rearchitect, or retain.
* Evaluate and test to ensure the migrated application meets the criteria outlined in your assessment plan.

**OPTIMIZE**

In the optimization phase, you will use Azure security and management resources to govern, secure, and monitor your cloud applications in Azure. This is also the time for you to look for opportunities to optimize spending. Common tasks at this stage are:

* Review Azure cost management to track spending and identify areas for cost savings.
* Evaluate migrated applications for opportunities to right size over provisioned virtual machines and services.
* Implement automation to resize or stop based on a utilization schedule.
* Identify applications that could benefit from optimization with platform-as-a-service (PaaS) services or containers.

You can learn more about each of these in the [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook and the [Operations and Management](http://assetsprod.microsoft.com/en-gb/mpn-playbook-cloud-operations.pdf) playbook.

When you expand these phases, you will have a structure similar to the Microsoft Solution Framework:



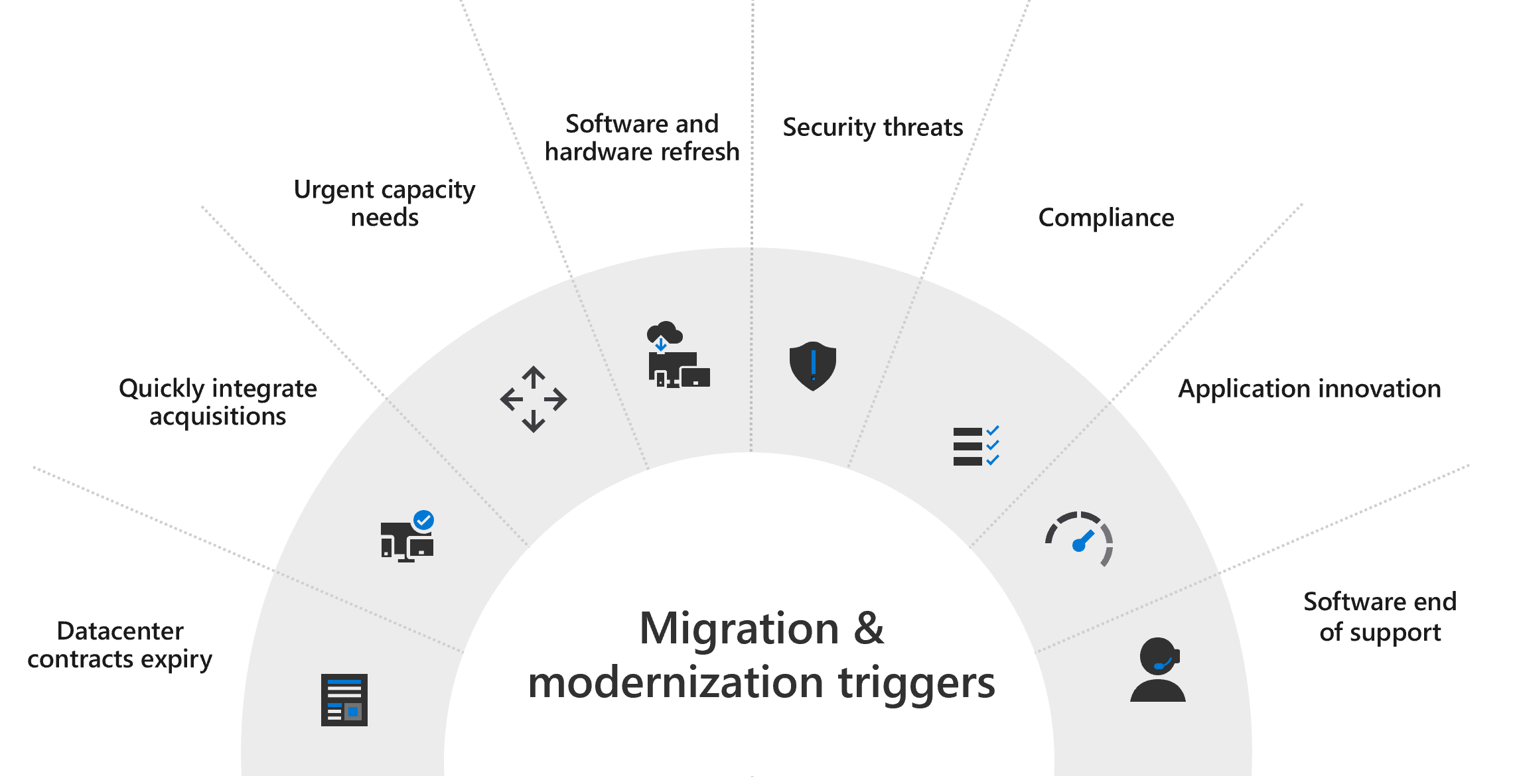
## Envision

In the envision phase, this is your opportunity to establish the organization’s maturity and familiarity with the cloud and cloud design principles. It is important to identify upfront where you are in the cloud journey. This foundational understanding will set the tone for the next plan around application and data migration and application modernization.

Microsoft’s cloud first approach prioritizes landing applications and data on one or more of the hyper-scale clouds – either public Azure or one of the sovereign Azure clouds (China, Germany, Gov/DOD). Azure Stack acts as another instance of a sovereign cloud, operated by the customer in their datacenter or consumed through a cloud service provider. However, Azure Stack is not a hyper-scale cloud and Microsoft does not publish or support any SLAs on Azure Stack.

Based on Microsoft’s experience working with customers directly by way of customer studies, 1:1 discussions, executive briefings, and envision sessions, the consensus is that you should take the time during the planning phase to level set expectations about Azure and Azure Stack and avoid pitfalls or setbacks later in the engagement. A good understanding of when to use Azure and when to use Azure Stack, as well as the differences between them, is the key to a successful implementation.

The drivers of the migration effort are varied and having a good understanding of them and the motive behind them, will guide the project towards a successful implementation:



|  |  |
| --- | --- |
| Area | Envisioning |
| Topics in scope | Identify stakeholders and influencers  Cloud reality check  Business intent and vertical alignment  Compliance, regulatory, and privacy restrictions  Definition of success |
| Discussion guidance | 1. Identify stakeholders and influencers    1. Identify the owners and decision makers that have direct influence over the application and data within the organization.    2. Differentiate between stakeholders that embrace the cloud vs those who will limit or block adoption.    3. Understand drivers and motivators for each 2. Cloud reality check    1. Understand customer’s cloud journey and where they are in that journey.    2. Motivation to keep apps/data on-premises (regulatory / data gravity / compliance).    3. Identify critical services that the organization expects to use overtime and their desired fault tolerance and availability. 3. Business intent    1. Tactical cost savings (reduce datacenter count or licenses).    2. Strategic transformation.    3. Exploratory. 4. Compliance, regulatory, and privacy restrictions    1. Which regulations impact the customer’s cloud adoption?    2. How will data privacy impact data migration?    3. What is the impact of internal compliance groups on cloud users? 5. Definition of success    1. Has the customer defined end results of migration and modernization?    2. Which R’s are in scope as part of the migration and modernization taxonomy (more on the R’s in the Assessment chapter)?    3. Have you consider the migration as a journey? If so, have you defined the next steps and the goals required after the migration?    4. Review the SLA, HA requirements and availability requirements. |
| Desired outcome checklist | * TCO and ROI analysis framework (capacity planning, pricing model) * Scenario based dependency map of desired IaaS/PaaS/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 (retain, retire, re-host, re-platform, re-purchase, re-factor) |
| Useful links | * [BRK2367 - Azure Stack Overview and Roadmap](https://myignite.techcommunity.microsoft.com/sessions/65842) * [Azure Stack capacity planning](https://docs.microsoft.com/en-us/azure/azure-stack/capacity-planning) * [BRK3317 - Best Practices for Planning Azure Stack deployment and post-deployment integrations with Azure](https://myignite.techcommunity.microsoft.com/sessions/65844) * [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook [“How is the Cloud Different?” chapter] * [Operations and Management](http://assetsprod.microsoft.com/en-gb/mpn-playbook-cloud-operations.pdf) playbook. * [Considerations for using virtual machines in Azure Stack](https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-vm-considerations) |

## 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 assessment phase. The organization will need to determine the appropriate categories and prioritization of short-term gains for simple lift and shift against long-term investments through modernization.

The categories defined need to help promote the benefits of migration and modernization and streamline the decision-making process during the assessment phase.

Discovering stakeholders with strategic control over the applications and datasets is critical to the completeness of the discovery phase. These are the people that are responsible for developing a migration and modernization plan for their applications and datasets. This is the opportunity to educate stakeholders on the technical and business benefits of migration and modernization.

The sources of information and tools used to discover and map applications and datasets to categories and stakeholders will vary from organization to organization. If the controls in place to maintain updated ownership records for deployed assets are not standardized, the quality and accuracy of the information collected will be impacted.

|  |  |
| --- | --- |
| Area | Discovery |
| Topics in scope | Identify sources of information  Automated discovery and crawling  Manual discovery  Dependency isolation and categorization  Business continuity impact |
| Discussion guidance | * Identify sources of information   + Qualifying the accuracy of the sources   + Plan to updates information sources * Automated discovery and crawling   + Tool selection criteria   + Identify critical points / potential bottlenecks   + Continuous vs targeted discovery / define how long the monitoring will last * Manual discovery   + Information validation process   + Correlating with automated discovery and Identify gaps * Dependency isolation and categorization   + Identify the dependencies that make up the core application and datasets   + Develop ways to clearly isolate the application and data from its dependencies, especially for infrastructure shared services   + Develop categories for applications, datasets and, infrastructure services   + Map regulatory, compliance, and privacy requirements * Business continuity impact   + Determine the acceptable tolerances for downtime that can be associated with each category or application/dataset |
| Desired outcome checklist | * Plan to ensure the association of stakeholders to applications and datasets will remain updated and accurate. Will there be an auditing process, self-identified, or programmatic? * Develop a decision matrix that clearly identifies the organizations preferred migration and modernization path and under what conditions an exception/exemption gets granted. * Define the cost of migration and modernization and the cost of maintaining exceptions/exemptions. * Propose a plan to actively track exceptions/exemptions and forcing functions to get these applications and datasets to conform. * Identify and minimize assumptions about the migration and modernization process and goals so all there is no confusion among stakeholders. * Develop a framework that will determine placement of the application and dataset – public cloud, sovereign cloud, service provider cloud, on-premise cloud. |
| Useful links | * [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook [“Discovery” chapter]. * [Azure Migrate](https://azure.microsoft.com/services/azure-migrate/) – while Azure Migrate doesn’t fully support Azure Stack as a target yet, it does include useful assessments which can be used. An example of this is offered through the [Cloudbase Coriolis tool](https://www.youtube.com/watch?v=ytRW034iueE), where the assessment done by Azure Migrate is used to create the migration environment for the entire application. * [Corent](https://www.corenttech.com/)’s SurPaaS® Platform is an Azure SaaS service that enables you to automate the scan, assessment, planning and cost modeling for your customers workloads, then automatically migrates them to the cloud, and then monitors, manages, optimizes and operates those workloads in the cloud. * [Service Map](https://docs.microsoft.com/en-us/azure/azure-monitor/insights/service-map) can also be used to identify dependencies and create a map for the solution before moving it. Once moved, you can use the same [solution on Azure Stack](https://aka.ms/azstackupdatemgmt) to test network metrics and compare to the indicators before the move. |

## Assessment

In the assessment phase, the customer is making data-driven decisions regarding which applications and datasets get targeted for migration and modernization. These decisions must reflect the organization’s transformation goals. Customer must coordinate with their 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. This is also the time to develop an operations and production readiness plan. Planning for and allocating budget and resources for investments in automation and tooling to maximize the migration and modernization velocity need to start here as well.

There are cases where certain “low hanging fruit” applications are identified for migration, but the support and licensing issues render the application unsupportable. While other cases will require a significant investment upfront in development and testing resources to accomplish the transformation. This is the phase when customers define downtime and data loss tolerances. As the acceptable amount of downtime and data loss approach zero, the customer will need to increase their investment to complete the transformation. Customers will need help navigating these decisions as they work on parallelizing efforts to meet short and long-term objectives.

|  |  |
| --- | --- |
| Area | Assessment |
| Topics in scope | Map an application or dataset to an R (more info below)  Determine the placement of an application  Quantify the value to the business and stakeholders of migrating or modernizing an application or dataset  Assess impact of regulatory, compliance, and privacy requirements  Assess infrastructure dependencies that must be addressed first  Determine success metrics and availability tolerances  Licensing and support impact |
| Discussion guidance | * Map an application or dataset to an R (more info below)   + Retain – no movement, exception/exemption process   + Retire – Develop a plan to decommission   + Re-host     - Manual redeploy     - Automated migration   + Re-platform   + Re-purchase   + Re-factor * Determine the placement of an application   + Public cloud   + Sovereign cloud   + Service provider cloud   + On-premise cloud * 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 * Assess impact of regulatory, compliance, and privacy requirements * Assess infrastructure dependencies that must be addressed first   + Identify   + Connectivity   + Security   + Encryption * Determine success metrics and availability tolerances   + Performance   + Availability   + Resiliency   + Deployability * Licensing and support impact   + Are there product licensing restrictions that will limit transformation?   + Is the application or dataset supportable in the new environment?   + Are there ISVs involved with the customer needs contact to confirm support statements? |
| 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 * Architecture guidance for users and application integrating with shared services * Clearly established SLAs between stakeholders * Identify existing services that need to be incorporated alongside Azure/Azure Stack services (AD and Azure AD, Exchange/SharePoint and O365) * Plan for new features and change management, both in Azure as well as Azure Stack |
| Useful links | * [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook [“Planning” chapter] * Learn more about the R’s in the [Cloud Migration Essentials e-book](https://azure.microsoft.com/mediahandler/files/resourcefiles/cloud-migration-essentials-e-book/Cloud_Migration_Essentials_E-Book.pdf) * Review the [Hybrid Patterns](https://aka.ms/azsdevtutorials) * [Naming conventions for Azure and resources](https://docs.microsoft.com/en-us/azure/architecture/best-practices/naming-conventions) – to a certain extent this applies to both Azure and Azure Stack resources * [Azure enterprise scaffold](https://docs.microsoft.com/en-us/azure/architecture/cloud-adoption/appendix/azure-scaffold) – to a certain extent this applies to both Azure and Azure Stack resources |

## Implementation

In the implementation phase, the customer now has a plan in hand with a project plan and schedule that drives the execution of the migration and modernization. This phase primarily focuses on the effectiveness and efficiency of the tools, techniques, and processes used to successfully drive the project to completion. Each “R” mapped to an application provides different opportunities to ensure the customer is building expertise in cloud design principles, even with traditional “lift and shift” style migrations. Use this as an opportunity to put into practice the investments that will drive the customer’s long-term cloud journey.

|  |  |
| --- | --- |
| Area | Implementation |
| Topics in scope | Implementation schedule  Multi-layered validation  Re-host – tradition lift and shift  Re-platform – targeted modifications without changing application design  Re-purchase – replace an application with a COTS or SaaS offering  Re-factor – redesign an application  Retain – keep the application as-is on the existing environment  Retire – eliminate the application from the environment, no need to replace |
| Discussion guidance | * Implementation schedule   + How is the schedule developed and who are the key stakeholders that sign off on the timeline and any changes to the plan of record?   + Migrations should be planned in sprints, and teams should work with the customers to ensure that changes are communicated correctly to stake holders, product owners, and any users that will be impacted by changes. * Multi-layered validation   + Validation at each step in the workflow is necessary to ensure the application in progressing towards the desired end state.   + Who owns designing and implementing 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? * Re-host   + Manual - Expectation is that the software can be installed and configured in the new environment like Azure or Azure Stack.     - COTS (Commercial off-the-shelf) – This is an opportunity to work with ISVs to provide ARM template, VM extensions, and automation that will govern the lifecycle of the application.     - ISVs are typically not motivated to develop Azure specific collateral. However, moving forward, there is no “MSI” that will help navigate a customer through setting up a public or hybrid cloud deployment of their application. Customers will need (and over time should demand) for more investment in higher level automation that covers the deployment, servicing, scaling, and remediation of the application in a cloud context. ISVs can harness the same Infrastructure-as-code methodologies customers implement for their own applications.     - Customer developed – Customers that have their own development teams building business applications and automation have more control over the lifecycle of the application. If they already invested in a code repository and in an integration pipeline, Azure or Azure Stack become one more target to support.     - This is the ideal opportunity to ensure they account for the full lifecycle of the 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.   + Automate – the software getting migrated lacks the original deployment packages and needs to be migrated “as-is,” including the hosting operating system.     - If the ISV is no longer available or stopped development work on the application, supporting the application long term will add to the customer’s technical debt in the datacenter.     - Are there opportunities to reduce the cost and overhead of operating the application in production? A “lift-and-shift” migration does not need to end after the application is put back into production. Leveraging the underlying services of the platform can enable the customer to do more with “infrastructure-as-code” by attaching capabilities provided by the cloud. Especially in a hybrid environment where the enhanced capabilities can be sourced directly from the cloud. * Re-platform   + 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; 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. It is necessary to understand the implications of different deployment topologies from fully and reliably connected to fully air-gapped. * Repurchase   + The decision to repurchase an application is a great opportunity to re-evaluate if there is an alternate COTS product or an equivalent SaaS offering that can replace the existing application. The assessment must take into account what features and capabilities the customer depends on and which ones are not available on the new environment.   + In the context of Azure and Azure Stack, 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 lifecycle services to the SaaS or new COTS application (e.g. monitoring, security, backup, etc.)? * Refactor   + Refactoring presents an opportunity to redesign an application. Depending the customer’s maturity in cloud technologies, refactoring may require significant upfront investment to properly design, implement, test, validate, and operationalize an application. This is further compounded by the applications complexity and dependencies.   + Customers may use this as an opportunity to take dependencies on advanced cloud services resulting in a hybrid deployment. Is the customer experience enough with these types of deployments to properly support this application? Has the customer developed a risk profile to assess the different failure domains and mitigate for each?   + It’s also important to have a change control process in place to avoid scope creep and avoid changes to the plan for record. * Retain   + Customers that keep an application in production with no plans to migrate or modernize it are adding long term technical debt that must be properly costed and passed on directly to the application owners.   + Does the customer have a plan to incentivize the application owners to accelerate the retirement of these applications? * Retire   + Retiring an application is an important decision and helps narrow the scope of migration and modernization. For standalone applications the decision to retire is simple.   + For more complicated applications, there may be unexpected interdependencies that may require retaining the application longer than originally expected or force re-platform or refactor to break the dependency. The goal should be to avoid unplanned downtime by completing the dependency map between applications and services. |
| 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, data protection. |
| Useful links: | * Learn more about the R’s in the [Cloud Migration Essentials e-book](https://azure.microsoft.com/mediahandler/files/resourcefiles/cloud-migration-essentials-e-book/Cloud_Migration_Essentials_E-Book.pdf) * [Cloud Migration and Modernization](https://assets.microsoft.com/en-us/mpn-playbook-cloud-migration.pdf) playbook [“Migrating to Virtual Machines” chapter] * “[Start with what you already have](https://azure.microsoft.com/en-us/blog/azure-stack-laas-part-two/)” blogpost |

## Operationalization

In the operationalization phase, the goal is to ensure the application is in a sustainable production environment. Declaring an application ready for production will depend on the customer’s requirements.

This is 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-premise applications. Some customers may be willing to backup and replicate their applications and data to Azure for the added layer of protection without requiring a build out of a second site. Optimizing the customers operational experience ensures the customer leverages the full benefits of the cloud and elevates even the simplest migration experience to include some modern design principles.

|  |  |
| --- | --- |
| Area | Operationalization |
| Topics in scope | Tolerance for hybrid cloud services  Hybrid cloud aware management  Applicability of available services |
| Discussion guidance | * Tolerance for hybrid cloud services   + Availability of reliable connectivity between on-premises and public Azure.   + Sufficient bandwidth and tolerable latency for critical data protection services.   + Cost analysis of integration and scaling with cloud-based monitoring, log collection, security and compliance scanning, and performance analysis services.   + Gap analysis of existing offerings and potential replacements. * Hybrid cloud aware management   + Correlation between IT managed cloud services and application specific services (avoid duplicate effort, optimize SLA).   + Automation required to orchestrate provisioning of services during deployment and migration of applications. * Applicability of available services   + Health/operational status and performance monitoring     - Well-defined metrics that form the basis of the SLAs guaranteed to the end-user.   + Security/compliance     - How well does the cloud environment meet the regulatory and compliance requirements imposed by the application?   + Backup/restore and replication/failover     - Data protection services for IaaS, PaaS, and SaaS resources.     - Incorporate multiple vendors, technologies, and capabilities to achieve a comprehensive protection strategy.   + Pricing/cost analysis     - Map of cost estimates to actual consumption and projections. |
| Desired outcome checklist | * Plan that incorporates services the customer has identified as required for the application to go into production. * Clear understanding of the dependencies created by these services, especially if they are hybrid services that run in the cloud. * Define metrics for cost, availability, and performance that map to business objectives. |
| Useful links | * [BRK3335 - Understanding architectural patterns and practices for business continuity and disaster recovery on Microsoft Azure Stack](https://myignite.techcommunity.microsoft.com/sessions/66253?source=sessions) * “[Protect your stuff](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-four/)” blogpost * [Azure Stack Considerations for Business Continuity and Disaster Recovery](https://aka.ms/azurestackbcdrconsiderationswp)  white paper |

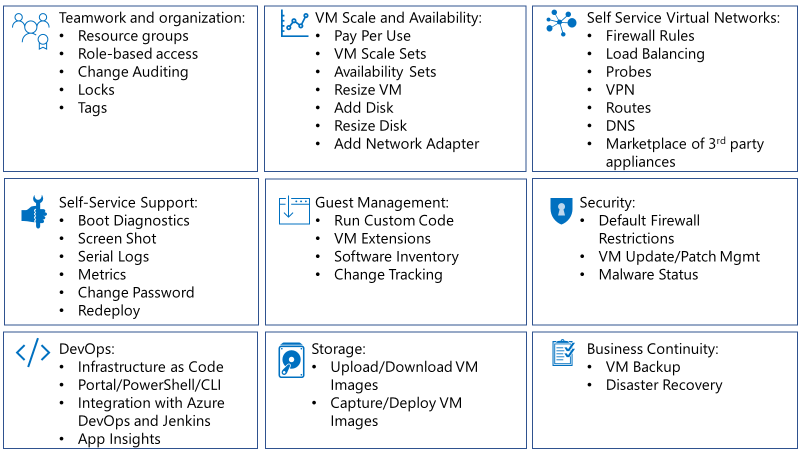
## Optimization

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. The original data used for sizing guidance could have been incomplete or the new environment introduces other latencies that were not expected. Continuous “right-sizing” across multiple dimensions re-enforces the benefits of migration.

|  |  |
| --- | --- |
| Area | Optimization |
| Topics in scope | Right-sizing the environment  Cost containment and re-allocation  Enabling and disabling of services |
| Discussion guidance | * Right-sizing the environment   + It is important to collect as many data points as possible about the application, how well it is performing, the utilization rates and trends, breaches in SLAs resulting in unplanned downtime.   + This data can be used to further optimize the environment the application lives in by refining 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. * Cost containment and re-allocation   + In a cloud environment, the customer may be shifting from traditional CAPEX funding models to OPEX. 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. * Enabling and disabling of services   + The original scope of the migration may have been very limited based on the customers comfort level with the cloud. Overtime, applications will 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, the customer should make changes to their production deployments in a structured way using automation and templatization. Any changes to the services surrounding the application need to be captured and audited to ensure there is a reliable “source of truth.” |
| Desired outcome checklist | * Customer has a plan to evaluate sizing continuously as opposed to ad-hoc * Customer can support friction-free right-sizing with minimal downtime |
| Useful links | * <https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-metrics-azure-data> * <https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-metrics-monitor> * “[Pay for what you use](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-six/)” blogpost |

## Modernization

Azure Stack is an extension of Microsoft Azure, which also means it is different than the traditional virtualization platforms (more info in [this](https://myignite.techcommunity.microsoft.com/sessions/65842?source=sessions) Ignite 2018 session). It offers PaaS services which are consistent with Azure, but also goes beyond virtualization platforms which offer VMs, by offering IaaS capabilities which are consistent with Azure. These capabilities enable a number of scenarios around the IaaS platform itself:

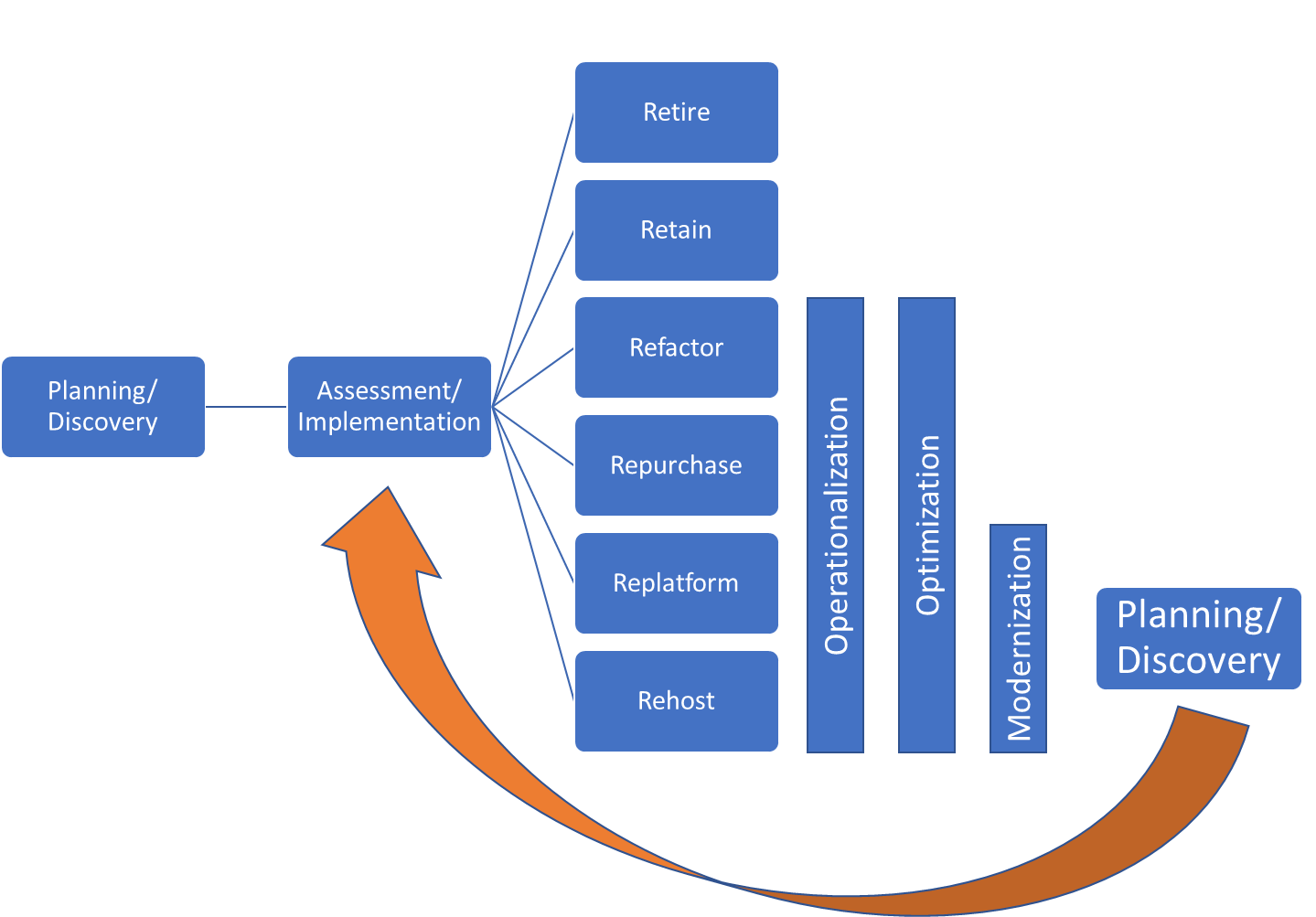


These topics are explored in the “Azure Stack at its core is an IaaS platform” blog series:

* [Azure Stack at its core is an Infrastructure-as-a-Service (IaaS) platform](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-one/)
* [Start with what you already have](https://azure.microsoft.com/en-us/blog/azure-stack-laas-part-two/)
* [Fundamentals of IaaS](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-3/)
* [Do it yourself](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-five/)
* [Pay for what you use](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-six/)
* [It takes a team](https://azure.microsoft.com/en-us/blog/azure-stack-iaas-part-seven/)
* If you do it often, automate it
* Build on the success of others
* Journey to PaaS

In the modernization phase, customers have an opportunity to retrofit a migrated application in production with these additional services. Many customers migrating applications and datasets using lift-and-shift are optimizing for minimal friction to accelerate migration of many applications. The customer may have decided to simply retain the application after migration but eventually retire it.

Other customers may have decided to postpone attaching additional services to or modernizing these applications. In either case there is an opportunity to introduce modernizations without changing the design of the application or requiring another migration to a new platform.



Modernization could start with something as trivial as creating a template for certain VMs in the solution and building on that to achieve a “Infrastructure as Code” deployment and management approach – where the entire environment is managed through declarative and imperative scripts.

|  |  |
| --- | --- |
| Area | Modernization |
| Topics in scope | Continuous migration and modernization cycle  Develop customer cloud maturity  Build skillset to continuously retrofit applications with cloud services  Optimizing for recoverability |
| Discussion guidance | * Continuous migration and modernization cycle   + The original intent of migration may have been to accelerate the decommission of a legacy environment. If the customer decided to take short cuts to get the cloud, there is still room to introduce operational modernizations to production application after migration.   + The customer must be comfortable with a continuous cycle of planning, discovery, assessment to increase modernization efforts and minimize the population of retained applications. * Develop customer cloud maturity   + Customers will need to identify opportunities for on-going education on cloud design principles. * Build skillset 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 the existing environment. * Optimizing for recoverability   + 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. Services complimentary services used during the operational lifecycle of the application need to be correctly provisioned and attached before the application is ready to go back into production.   + The recovery effort 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 to introduce human-error into the deployment recovery workflow. |
| Desired outcome checklist | * Customer has a plan to attached hybrid cloud services to lift-and-shift applications   + Monitoring   + Security   + Data protection   + Log analytics |
| Useful links | * <https://github.com/Azure/AzureStack-QuickStart-Templates> * <https://github.com/Azure/Azure-QuickStart-Templates> * <https://docs.microsoft.com/en-us/azure/devops/learn/what-is-infrastructure-as-code> |

## Decommission

In the decommission phase, the customer needs to develop retirement plan for migrated applications that they simply plan to retain and will never go through another migration and modernization cycle. The primary purpose of the plan is to accelerate retirement and expose the ongoing operational cost of retaining these applications indefinitely. Similar to how many organizations require approval and justification to deploy a physical server instead of a virtual server. This is an opportunity to quantify the ongoing support cost for legacy applications and push application owners to adhere to a time-bound retirement plan.

|  |  |
| --- | --- |
| Area | Decommission |
| Topics in scope | Quantifying on-going costs of support legacy applications  Qualifying the skillset required to maintain legacy application  Accelerating the pace of retirement |
| Discussion guidance | * Quantifying on-going costs of support legacy applications   + Migration and modernization sets 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. * Qualifying the skillset required to maintain legacy application   + Retaining applications requires keeping personnel trained on legacy application and protocols. Overtime there is higher risk of developing knowledge gaps that can lead to unplanned downtime. * Accelerating the pace of retirement   + Application owners will need to be incentivized 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.   + The application owner should also have to pay upfront the fully loaded cost of supporting legacy application in production over a set period of time. The cost of supporting these legacy applications should not get bundled/aggregated with the costs of operating modernized applications. |
| Desired outcome checklist | * After migration, customer is continuously re-evaluating their applications to identify retire candidates even after migration * Customer uses support costs or complexity to drive application owner to accelerate retirement of an application. |

# The Migration Tools

The migration itself can be done with a number of tools. Our partner ecosystem includes ISVs that have built solutions which range from simple migrations, to being able to deliver “as-a-service” solutions as well. There are also Microsoft Migration options, which require manual steps to implement, but offer a potential lower cost. These tools are described in the [part two](https://azure.microsoft.com/en-us/blog/azure-stack-laas-part-two/) of our Azure Stack IaaS blog series.

The “SQL 2008 and SQL 2008 R2 Migration to Azure Stack” document also talks about the tools and options to migrate SQL specific workloads. The “[Migrate SQL Server to SQL Server on Azure Virtual Machines](https://datamigration.microsoft.com/scenario/sql-to-sqlvm?step=1)” also applies to Azure Stack migrations and can be used as reference towards ensuring a successful migration.