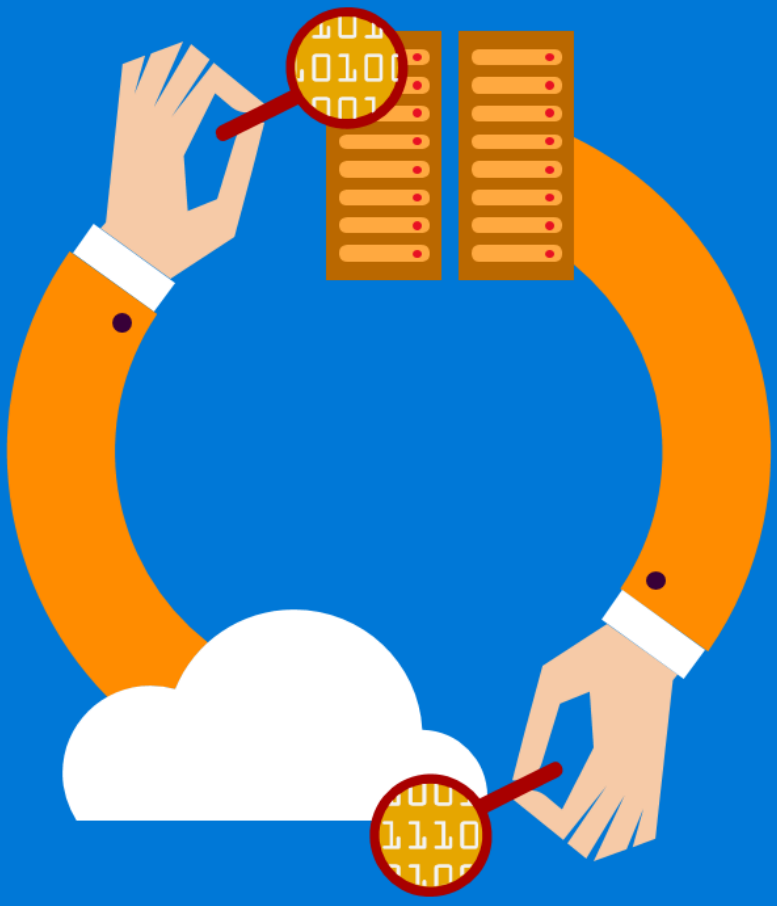
Microsoft



Azure Stack

Considerations for business continuity and disaster recovery

By Hector Linares and Joel Yoker
AzureCAT

March 2019

Contents

List of figures

Authored by Hector Linares and Joel Yoker. Edited by Nanette Ray. Reviewed by AzureCAT.

Executive summary

Microsoft Azure Stack is an extension of Azure that lets you deliver Azure services from your organization's datacenter or consume them directly from a service provider. But Azure Stack is more than just an on-premises cloud infrastructure. Consuming infrastructure-as-a-service (IaaS) and platform-as-a-service (PaaS) services from Azure Stack requires a modern approach to planning business continuity and disaster recovery (BC/DR) compared to traditional on-premises solutions.

Operating a cloud environment requires you to be prepared to handle a spectrum of issues ranging from simple component failures to degraded systems to full outages. Starting with standard break-fix mitigations, Azure Stack is designed to heal itself using auto-remediation at different levels of the system. More complicated situations that result in outages or a site-level disaster require planning to protect against extended downtime for your applications.

In our conversations with customers, we learned that even experienced datacenter and cloud administrators mistakenly think that Azure Stack is simply an evolution of virtualization, or a management layer on top of storage and virtualization. In fact, they expect to treat Azure Stack like any other virtualization platform. This "business-as-usual" mindset generates the wrong expectation. The tools and processes used for Hyper-V or VMware environments don't work with Azure Stack in the same way.

It's better to think of Azure Stack as a cloud environment like Azure. You must design a business continuity and disaster recovery strategy for the applications and user data separate from the Azure Stack infrastructure. Many of the practices your organization uses—backup, recovery, replication, high availability—apply to Azure Stack. However, you must think through how these practices apply to the applications and user data apart from the underlying Azure Stack infrastructure. Critical to this exercise is the need to update your organization's recovery time objectives (RTOs), recovery point objectives (RPOs), and service-level agreements (SLAs) for availability and recovery, so you provide the best service to your end-users.

This guide looks at what recovery means for Azure Stack and is intended for operators and architects. Familiarity with traditional protection strategies—such as backup/restore, replication/failover, and high availability—still apply, but the focus is at the user level, not the infrastructure. Applications owners can also benefit from this guide as it helps illustrate how to think about application and service availability with Azure Stack. This material is based on our work with customers who have deployed Azure Stack and isn't intended to be prescriptive guidance. Our goal is to help you understand how Azure Stack differs from environments you already know so you can plan appropriately.

For an introduction to Azure Stack planning and implementation, see Azure Stack: Building an end-to-end validation environment.

Understanding Azure Stack recovery

Both public Azure and Azure Stack provide common elements that support workload availability. Workloads on Azure Stack can achieve high availability, site failover, and application resiliency with a combination of native capabilities in Azure Stack, best practices and patterns for application design, and partner products. In traditional virtualization environments, some customers expect that all methods of protection are handled under the covers by the infrastructure. This approach isn't compatible with Azure and Azure Stack.

Azure Stack is a co-engineered integrated solution comprising Microsoft software, an integrated system provided by a certified hardware partner, and infrastructure (IaaS) and platform (PaaS) services. One or more Azure Stack systems can be deployed in your datacenter or in multiple datacenters in different geographic locations. Each system is independent and must be fully deployed and configured so it's ready to provide capacity to your users.

When you deploy an application to Azure Stack, you need to think about the various failures that can affect applications, starting with hardware issues, system-level issues, and even major events, such as the datacenter or a site going offline. Depending on the risk tolerance of the application owner, you need to determine the impact of the various levels of risk:

- You may have applications that have no requirements for quick recovery and can sustain complete data loss. Others application can sustain extended downtime if their data is protected in an external environment.
- More critical application may require higher levels of availability to withstand failures of system hardware or even the entire system. For these application, you may have to provide multiple systems in one datacenter for quick recovery and availability or across multiple datacenters for disaster recovery.

In Azure, you can protect at a rack or zone level within a region or across regions. At the region level, there is no expectation that Microsoft will fail over the Azure infrastructure service from one region to another. All regions run all services, so it's up to the application to fail over and attach to the resources it needs in that region.

In a similar way, systems that run the Azure Stack infrastructure services don't fail over to another system. Rather, it's your responsibility to plan availability and recovery for each application based on the services they depend on. Figure 1 provides an example of a typical Azure Stack infrastructure.

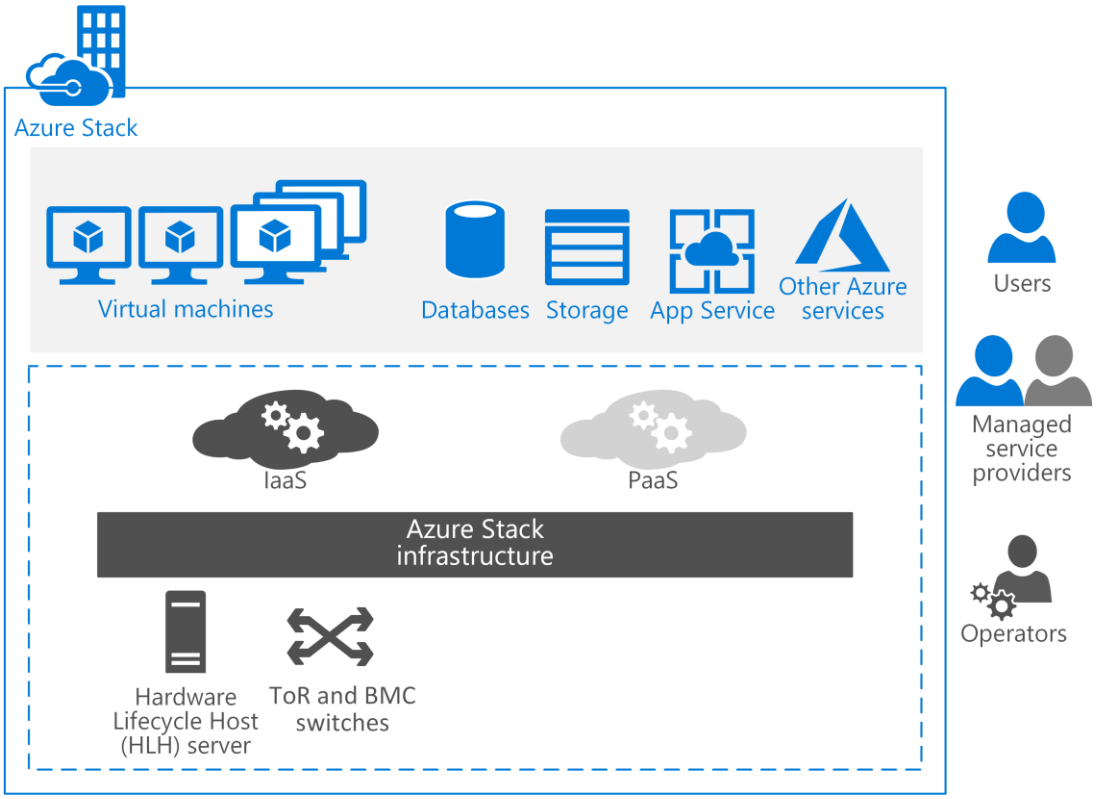


Figure 1. Azure Stack is a physical infrastructure and a platform for services, user applications, and data

In the virtualization world, it's common to think about recovery from the bottom up—starting at the hypervisor and storage level. The infrastructure comes first, and then applications are restored using a point-in-time backup of virtual machines (VMs). As a provider of Azure Stack services, you need to design a recovery strategy from the top down, starting with your end user's application and data. Separately you work through the recovery strategy of the infrastructure.

When planning for protection and recovery of Azure Stack, you must consider the following:

- **Applications and data.** What are the requirements for keeping user workloads highly available and resilient? What are the service dependencies? What aspects of availability and recovery are handled at the application level and what is required from the platform? Finally, how do you recover from failure?
- **Services.** What needs to happen to prepare the cloud so that user applications and data are restored? How long does restoration take? What is your protection strategy for data stored in your services?
- **Infrastructure.** In the event of a disaster, are redundant Azure Stack systems available? What agreements and infrastructure are required to restore or redeploy Azure Stack? How are your SLAs affected in the event of a catastrophic failure of an Azure Stack region?

It's important to understand that the strategies, roles, and responsibilities for Azure Stack recovery are closer to Azure than traditional physical and virtualization infrastructures. In Azure, the underlying infrastructure is operated by Microsoft. Microsoft is responsible for restoring

services in the event of an outage or a disaster. The users of the multitenant cloud are responsible for their applications and data that reside on the cloud. Developers must take into account the resiliency, availability, and recovery mechanisms as part of their overall cloud workload strategy.

Like Azure, the approach to business continuity and disaster recovery for Azure Stack separates the protection and recovery of the infrastructure from the user applications and data. The main difference is that your organization, as the service provider and operator, is responsible for restoring services in the event of an outage or disaster. By implementing Azure Stack, your organization takes on the responsibility for delivering Azure services and maintaining an on-premises Azure infrastructure. This carries several dependencies, and in some cases, you must involve your certified hardware partner as part of the recovery effort.

In a multitenant Azure Stack cloud, your users are responsible for developing applications with the same cloud design principles they would use in Azure. Your organization can use multiple Azure Stack clouds within your enterprise network or use the public cloud to provide recovery capacity for business and mission-critical applications.

What Azure Stack is—and isn't

Azure Stack gives organizations the ability to develop solutions using cloud-native Azure services within your datacenter. Each Azure Stack scale unit is built with a high level of redundancy—down to the power, network, nodes (servers), and disks required to host resilient Azure services. However, this redundancy is constrained to a single *scale unit* or Azure Stack region. A scale unit in Azure Stack is a collection of nodes with homogenous hardware, meaning each node has the same hardware specifications for CPU, memory, and disks.

While Azure consists of multiple regions across the globe, organizations can choose to deploy a single Azure Stack environment to host their workloads. The redundancy built into the platform addresses their overall resiliency needs for a single system in a rack within a datacenter. However, for applications that need fast recovery in the event of an outage or disaster, you need to deploy multiple Azure Stack systems in multiple locations to employ a proper resiliency strategy for your organization's workloads.

Figure 2 shows what Azure Stack is through the lens of the logical and physical topology, starting with the site where a scale unit is deployed. At the site, the scale unit may be co-located with other servers within a datacenter—a room, a row, or some physical place where the rack goes. The rack is where the scale unit is deployed. The cloud is a logical construct within a rack and provides access to Azure Resource Manager, the control plane for cloud resources. The cloud hosts a logical Azure Stack region. A region contains one scale unit. Within a scale unit are the physical nodes hosting the user applications, data, and all the Azure Stack services.

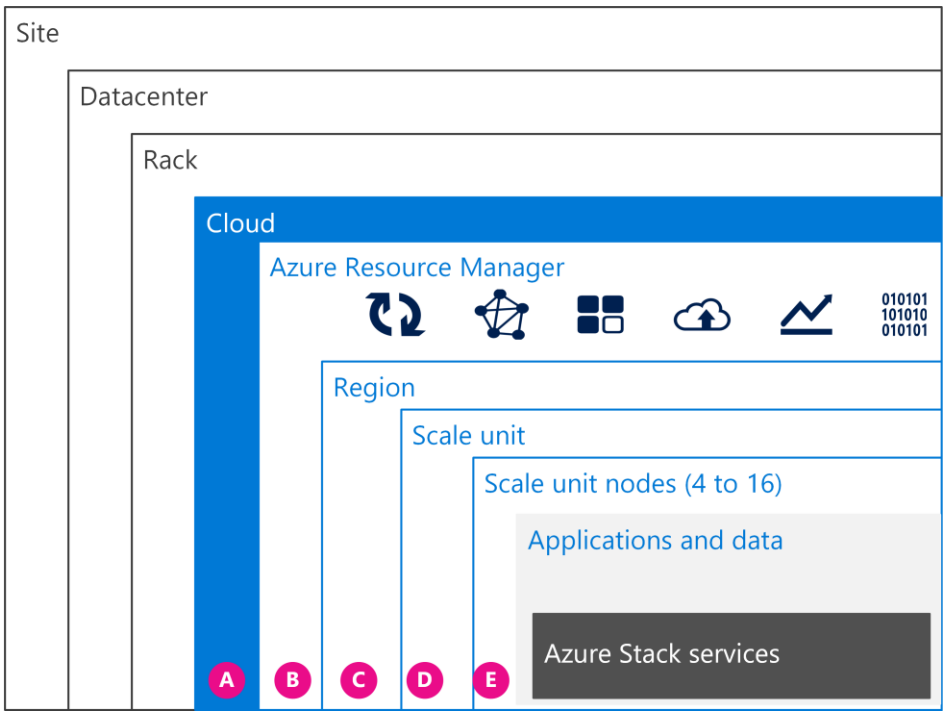


Figure 2. Azure Stack is an integrated system representing a single region with a single scale-unit deployment consisting of:

- A One set of management and portal endpoints
- B One region
- C One scale unit per region
- D 4 to 16 scale-unit nodes
- E Azure Stack services, user apps, and user data

When you lose a site, you can lose the scale unit at that site. The Azure Stack region, scale unit, and the Azure Resource Manager portal go away. That is why business continuity and disaster recovery planning for Azure Stack starts with planning for site-level disasters if required for your organization.

IT organizations may assume that their existing virtualization environments can serve as the model for developing a business continuity plan or disaster recovery strategy for Azure Stack. This misunderstanding creates the following common misconceptions.

Misconception 1: Azure Stack is a virtualization platform

This idea misses a fundamental benefit of Azure Stack. It's an extension of Azure in your datacenter. That means you need to think in terms of protecting applications and data within the context of cloud services.

You are responsible for supporting high availability and redundancy and for performing familiar operations—such as backup-recovery, archiving, and monitoring—separate from how the infrastructure is operated. Don't expect these operations to be offloaded to the underlying infrastructure.

A simple example is that Azure Stack is delivered as an integrated system on prescriptive hardware, so there is no opportunity to attach external storage devices into the system. Even for services such as Azure Backup and Azure Site Recovery, the experience is at the user level in the context of Azure Resource Manager resources. This differs from virtualization environments, where these services operate at the hypervisor level. Azure Stack is a hybrid cloud platform, as shown in Figure 3.

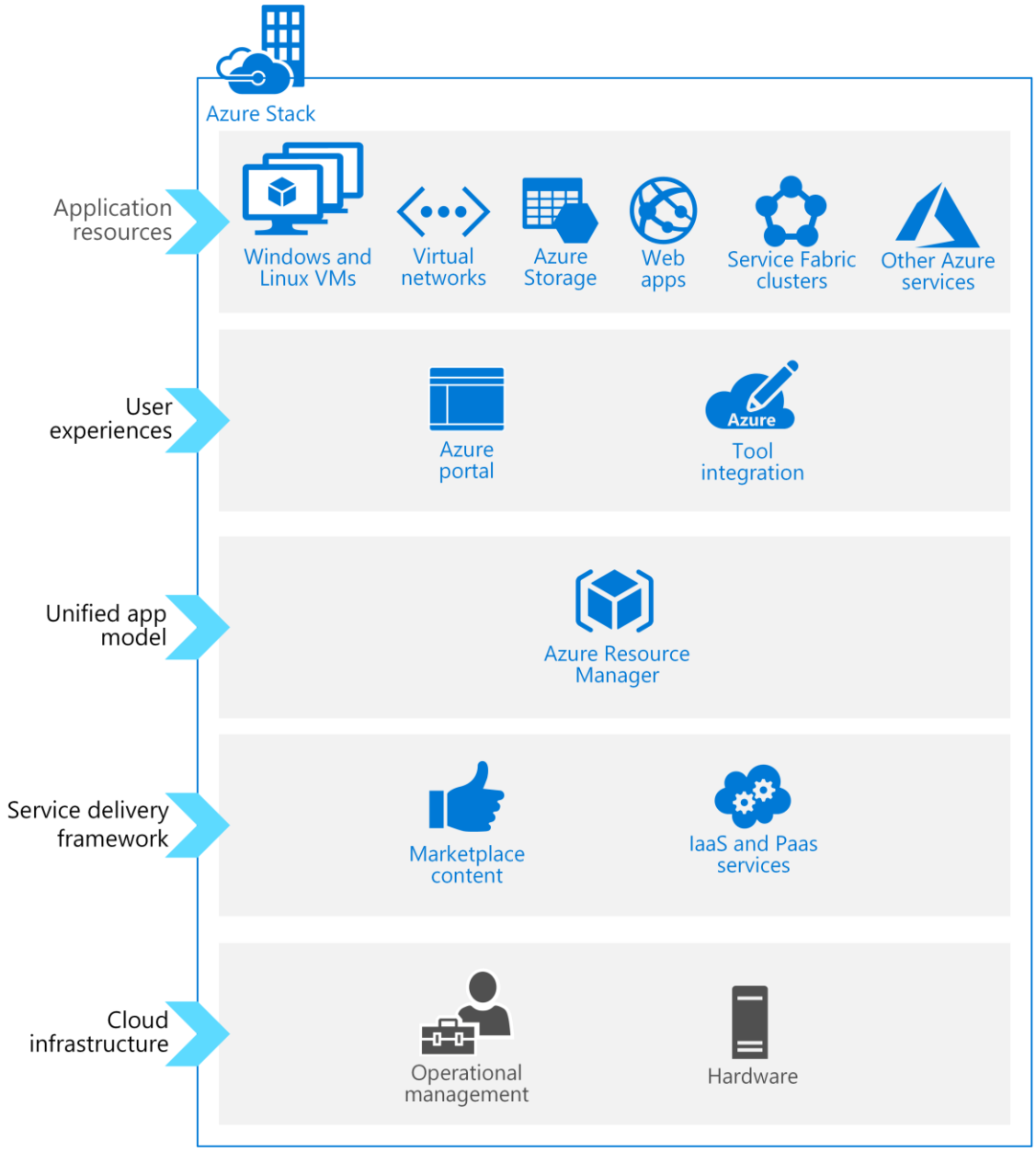


Figure 3. Azure Stack is a hybrid cloud platform

Misconception 2: I can protect Azure Stack like other virtualization platforms

Protecting, healing, and recovering Azure Stack and the Azure workloads hosted within it are

fundamentally different tasks compared to those you may do today in other virtualization environments. Managing the Azure Stack system takes place through an administration portal—there is no access to traditional platform tools.

In addition, Azure Stack assumes a sealed host model that is secure-by-default with standard, uniform security settings. It doesn't permit outside software to be deployed on the servers. This model means that host-level software agents used for monitoring, security, or backup solutions can't be deployed to the Azure Stack infrastructure. In fact, the default configuration of Azure Stack is uniform across all deployments, including the network access control lists (ACLs) applied to the top-of-rack switches that limit traffic flow to prescribed paths. All Azure Stack administrative access occurs through the Azure Stack operator portal or through a privileged endpoint for diagnostic operations.

From a deployment perspective, an Azure Stack scale-unit is designed to be deployed within a single site. Within the scale-unit, high availability and resiliency for compute and storage resources are protected against local failures with N-1 fault tolerance. As the following figure shows, deploying a scale-unit across two separate sites isn't supported and doesn't provide the level of protection typical of a multiple-site failover cluster environment. This design is intentional, and no hardware partner will deploy Azure Stack in this topology.

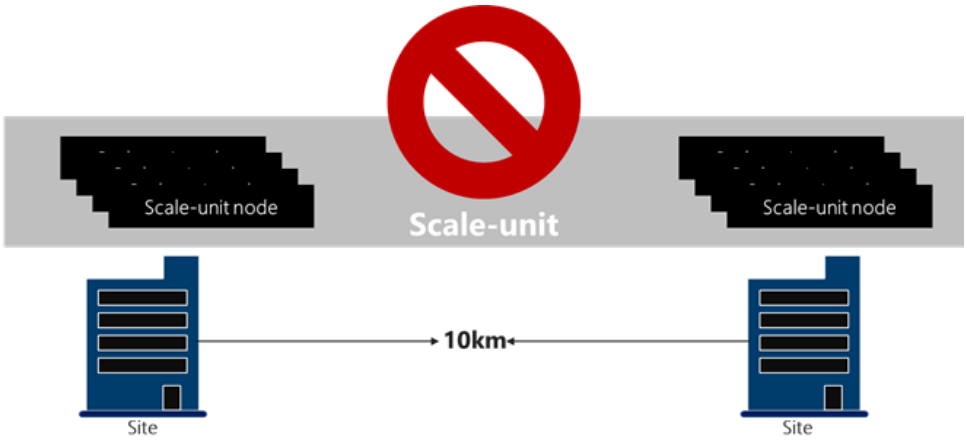


Figure 4. Scale-units don't stretch across datacenters or sites

Most important, as a co-engineered integrated solution deployed by a hardware partner, in the recovery case, if the system needs to be redeployed, only the hardware partner can provide those deployment services.

Misconception 3: But I'm not in the Azure cloud—it doesn't apply to me

Unlike traditional physical and virtualization infrastructures, Azure Stack follows the cloud design patterns used by Azure. In this way, a business continuity and disaster recovery strategy for Azure Stack needs to consider the protection and recovery strategies of a multitenant cloud, such as:

- Can an application be deployed across multiple fault domains and into multiple locations?
- Does an application support active/active? If so, how do you achieve data consistency across sites?
- Is application federation across multiple clouds supported?

- What kind of automation is needed to rapidly redeploy?

The implementation of a business continuity and disaster recovery strategy determines the service providers, software vendors, and technologies that can help you achieve your recovery objectives.

Recovery objectives and SLAs

Recovery point objectives (RPOs) and recovery time objectives (RTOs) are important considerations when developing your Azure Stack disaster recovery plan. These metrics are usually documented by the business continuity team as part of the continuity requirements for specific business functions.

As a review of these concepts:

- **RPO** covers the maximum amount (in time) of data that can be lost in case of a disruption. It answers the question, "To what point in time can I recover?"
- **RTO** covers the maximum amount of time it will take from a disruption to bring back the business functions, including data. It answers the question, "At what point in time can I expect business operations to continue?"

A realistic RTO for Azure Stack includes the timeline for recovering both the Azure Stack infrastructure and user applications and data. An acceptable RTO considers the severity of the event, especially if Azure Stack must be redeployed to restore IaaS and PaaS services. From that, you can determine realistic SLAs.

Azure Stack recovery involves multiple stakeholders and occurs in phases as Figure 5 shows.

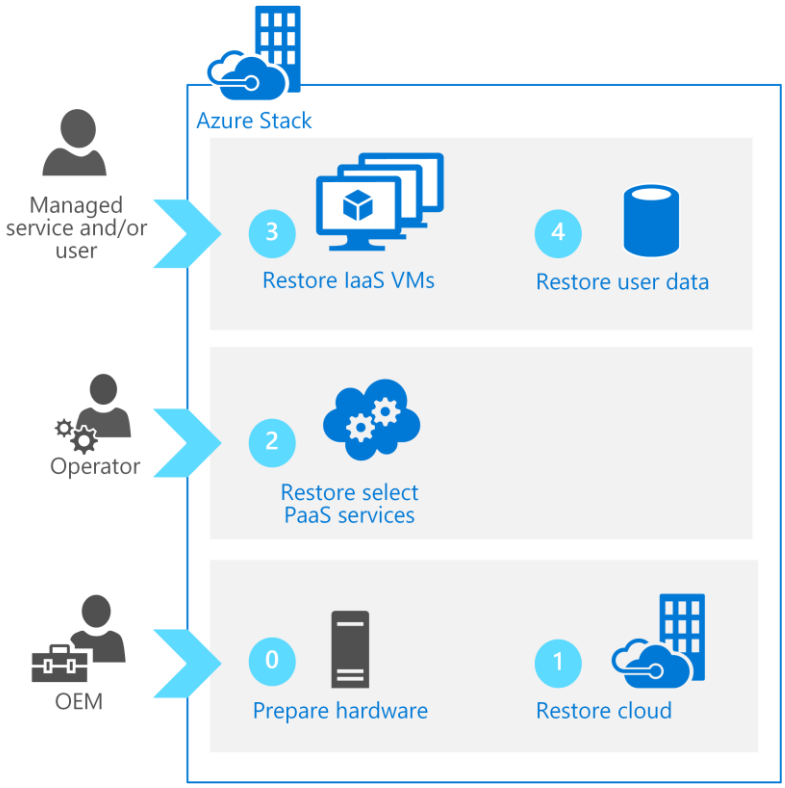


Figure 5. Azure Stack recovery is a shared responsibility and occurs in phases

The following table describes the phases and how long each phase takes:

Phase	Task	How long
0	**Ensure the Azure Stack integrated system is ready.** This phase is the responsibility of the selected Azure Stack hardware partner. Recovery of Azure may take place using an existing Azure Stack integrated system or a replacement unit. This phase ensures that all the components of the Azure Stack integrated system are operational and ready to support the recovery process. If the disruption is a result of a component failure (nodes or disks), ensure that the replacement process has been performed and the physical components of the Azure Stack integrated system is operational.	Weeks if new hardware is needed. **NOTE:** If new hardware is needed, time-to-delivery is a factor you should consider as part of the RTO of the solution in the event of a catastrophic disaster.
1	**Perform the Azure Stack cloud recovery process.** This phase is the responsibility of the selected Azure Stack hardware partner in collaboration with the customer and Microsoft. There are two deployment options at this point: Standard deployment (the same as when a new system is deployed for the first time) or cloud recovery. The Azure Stack cloud recovery process recovers the core Azure Stack metadata and configuration for the Azure Stack scale unit. Azure Stack configuration information—such as the internal identity system, internal CA root certificate, subscriptions, plans, offers, and quotas—is restored to accelerate onboarding of users who had access to the system before the redeployment.	A week once the new hardware is on site, racked, and connected. Required professional services must be on site or have remote access to drive deployment.
2	**Restore PaaS resources.** Once the Azure Stack system is redeployed, the next step is to install PaaS resources and data for Azure Key Vault, role-based access control (RBAC), and storage accounts. This phase is the responsibility of the Azure Stack cloud operator.	Days.
3	**Restore IaaS resources.** The next step is to restore tenant-level IaaS VMs from available backups. This phase is the responsibility of the Azure Stack users or the managed service provider.	Hours, days, or weeks.
4	**Recover user data.** The next step is to restore tenant-level PaaS data from available backups. This can include blobs and databases, for example. This phase is the responsibility of the Azure Stack users or managed service provider, who use backups or other tools to bring data back to the system.	Hours, days, or weeks.

As the table shows, recovery can take weeks if replacement hardware is involved. Not until phases 0, 1, and 2 are complete can the managed service provider and users begin to restore the workloads on Azure Stack along with the associated user data. In the event of a catastrophic failure or a disaster, users who can't wait the time required to complete a full redeployment must plan to have a secondary site for recovery of applications. This can be Azure or another Azure Stack system. If there are multiple Azure Stack systems, all systems must be fully deployed, registered, and configured. There is no case where Azure Stack can be deployed on demand directly by a customer. The hardware partner must always be involved. Later in this document, we describe the various protection schemes you can use when you have two Azure Stack systems.

Recovery timeline and phases

At a high level, it's necessary to establish the timeline that an organization can expect to recover Azure Stack in the event all data is lost and the system needs to be redeployed. From an architecture perspective, application owners need to consider the impact of downtime and design their application to meet the organization's availability and data durability requirements. The RTO timeline in the previous table calls out the key milestones that determine the RTO range for recovering Azure Stack. The recovery phases cover the work involved in each major phase. These phases don't need to map to RTO milestones directly. Recovery phases represents a generic business continuity plan (BCP) template for Azure Stack.

Figure 6. Major phases in recovery

The following table captures activities that an organization needs to account for in their business continuity plan (BCP) for Azure Stack infrastructure.

Phase	Tasks

Initial triage	Assess the health of the cloud (degraded, offline). Determine if the issue affecting Azure Stack is internal or external.
	
	Procedure for an internal issue that may require support intervention:
	
	• Evaluate the impact of the event to the cloud's health.
	• Determine if the issues are transient or permanent.
	• Determine if there are any available endpoints.
	  - Are the user workloads available?
	  - Is the user portal available?
	  - Is the operator portal available?
	  - Is the privileged endpoint available?
	  - Is the hardware lifecycle host (HLH) available?
	• Confirm support status.
	  - See if the affected environment is running the latest builds or is within

the N-2 release deferment window.

- If required, invoke the appropriate recovery procedure for critical workloads based on the SLA established with end users and application owners.

Internal & external engagement

Engage internal and external entities:

- Notify internal and external users that the recovery procedure is in effect.
- Collaborate with key stakeholders who need to act in the recovery plan for infrastructure and user services. Engage external entities that support the Azure Stack solution.
  - Determine if the issue is related to hardware, software, or both.
  - Have the support team diagnose and troubleshoot the problems with the intent to heal the system and bring it back to an operational state.
    - Assess if the solution is degraded or in critical condition and product support can't make the solution healthy.
    - Assess if the solution is completely or partially offline and product support can't bring the solution online.
- Spend time getting the system back to a healthy state. Expect mitigation to take a while. Only in the worst cases will a system remain offline permanently.
- In the worst-case scenario, work with product support and product engineering from Microsoft (along with the appropriate internal and external stakeholders) to determine if the only path forward is to redeploy the solution.
- Collaborate with your hardware partner to determine what is required to get the system back online. If no hardware was damaged, the only requirement is to redeploy Azure Stack. However, if hardware must be replaced, schedule the delivery of components. In the worst case, order a new system, which must involve your account team with the hardware vendor.
- Determine if the redeployment will follow standard deployment procedures or if the solution should use cloud recovery.
- Work with the hardware partner to update the deployment worksheet in preparation for deployment.

Phase	Tasks
Hardware restoration	This step covers the procedures for an internal issue that requires some level of support intervention without the need to repair system hardware. • Coordinate with hardware partner and professional services who need access to the system. • Partner will use their standard deployment guides, generate new configurations, update the top-of-rack (ToR) switches and baseboard management controllers (BMCs), update HLH, and run through all the hardware checks. • Firmware leveling may be required if the system isn't on the correct version based on the baseline version that will be used for deployment. If hardware must be replaced: • Coordinate arrival of replacement hardware. • Replace failed components in preparation for deployment. If a new system must be deployed: • Coordinate arrival of a new system. • Coordinate provisioning of network and power for the system. • Coordinate rack installation and connection to power and networking.
Software restoration	Restore software: • For cloud recovery deployment, confirm access to the storage location that contains infrastructure backups (using deployment ID to identify the content). • Coordinate the arrival of personnel who will drive deployment. • Complete deployment worksheet and generate new deployment files. The goal here is to recover specific Azure Stack configurations and user data from the original deployment. For cloud recovery, the goal is to redeploy the system as close to the original as possible. Some overrides to deployment are supported—primarily changes to network configuration. Cloud recovery is hardware-agnostic, so if there is new hardware in place, no changes are required to backup. • Execute deployment in cloud recovery mode. - Make sure file share with backups is available and accessible for the duration of deployment. Customers are responsible for providing the share. - Note that the location of the backup files used by restore can be different from the backup location used by the new deployment after deployment for new backups.

Phase	Tasks
Validation & handoff	• Hardware partner will complete the steps in the deployment guide related to registration, testing Azure Stack, and required hand-off procedures.
Operator validation	• Validate subscriptions, plans, offers, quotas, secrets, storage accounts, RBAC roles, and policies. • Inform customers that they can start to log back into the portal so they can start the recovery process.
Application & data recovery	Users must work though the restoration of their IaaS VMs and PaaS data.

Roles and responsibilities during recovery

With traditional on-premises virtualization solutions in your datacenter, you manage everything and retain control over any virtualization infrastructure. With Azure Stack, the lines of responsibility are shared by several stakeholders as the following table shows:

Stakeholder	Role	Responsibilities
Microsoft	Customer support Product engineering	Provides the initial point of contact when you call Microsoft. Triages the issue, troubleshoots, and attempts to bring the system back online. Escalates as needed. Works with you to create an action plan, diagnose the root cause, and discuss redeployment if needed.
Original equipment manufacturer (OEM)	Customer support Product engineering	Provides the initial point of contact when you call. Triages the issue and troubleshoots. Escalates the issue as needed, working with Microsoft to resolve hardware issues. Works with you to diagnose the root cause. If hardware-related, works with Microsoft to complete the recovery.

Stakeholder	Role	Responsibilities
Customer	Application and infrastructure architects	Develops a business continuity plan and SLAs for their Azure Stack offering and works with users of the cloud to determine the correct business continuity and disaster recovery strategy for applications.
Customer	Azure Stack operator	Configures and executes infrastructure backups. Monitors and manages the external file share that contains infrastructure backups.
Customer	Azure Stack users	For managed offerings, user works with the team that owns the recovery workflow for the application. For self-service environments, the user is responsible for invoking recovery procedures themselves.

The key point is that the operator's role in recovery of Azure Stack differs from their typical role—especially with the active involvement of Microsoft and the hardware partners. It's very important to establish the key stakeholders and their roles for your organization as part of the recovery plan.

Cloud resiliency

A comprehensive business continuity and disaster recovery strategy must identify the requirements for resiliency and availability and determine the level of financial investments and effort required to meet those requirements.

Hardware fault tolerance

To achieve continuous operation of your Azure Stack system despite temporary failures in services, network connectivity, or hardware, Azure Stack provides fault tolerance at these levels.

- **Node level.** Each Azure Stack node has two power supplies, two network adapters, and two boot drives (depending on OEM configuration). Each is configured for fault tolerance.

  NOTE Each node has one BMC network port.

- **System level.** An Azure Stack system is deployed with 4 to 16 nodes in N-1 redundancy configuration.
- **Rack level.** An Azure Stack system ships with two ToR switches for fault tolerance.

  NOTE The system ships with one BMC switch. If the BMC switch fails, the applications and foundational services will not be affected. Loss of the BMC switch will affect some node-level operations, such as power on.

High availability

To achieve availability of the infrastructure and applications deployed by users, Azure Stack provides high availability at three levels:

- **Scale-unit level.** An Azure Stack scale-unit is configured to support failure of one node (N-1). Infrastructure compute, storage, and networking services are designed to continue operating without interruption.
- **Foundational services.** The software services that manage underlying capacity allocation and present consumable resources to users are designed to continue servicing requests when a node is down.
- **User applications.** Users can protect against node-level failures by placing virtual machines in an *availability set* that spreads them across multiple *fault domains*.

Disaster recovery

Loss of the Azure Stack system or the site that contains the system—and all the applications and data contained within—is a catastrophic event. You must consider the following scenarios:

- **Single site.** Organizations with a single Azure Stack system don't have the additional capacity to run applications if the system goes offline. If the system is permanently offline and unrecoverable, you need to wait for that system to be deployed before you can restore user applications and data. This might be acceptable in some cases, depending on the SLA established between you and your end users.
- **Site to cloud.** The public cloud can also be used as a secondary site to recover applications and data in the event an Azure Stack system goes offline. The benefit of this approach is that Azure is available globally and on demand. There is no infrastructure to purchase.
- **Multiple systems.** In this case, you have more than one Azure Stack system deployed in one or more sites. These sites can be separate datacenters in the same or different geographic locations. Applications deployed on these systems need to be configured for high availability or protected with a backup product to enable on-demand restoration in a secondary location. When you invest in multiple Azure Stack systems, you need to provide this level of recovery if your applications and data can't reside in the public cloud.

These concepts aren't new to anyone familiar with managing traditional and virtualization environments. Enterprises rely on multiple strategies for preventing failures and minimizing time to recovery. The difference in Azure and Azure Stack environments as the following figure shows is that shift in focus to recover applications and data separate from how the underlying infrastructure is recovered. The goal is to minimize the effect of a failure.

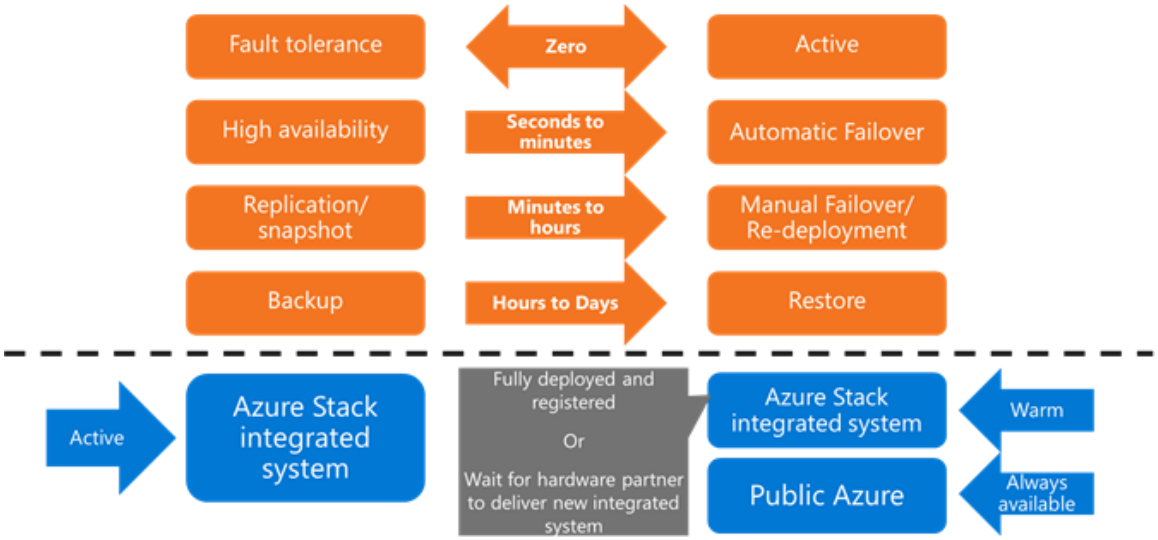


Figure 7. Cloud resiliency includes planning for recovery time. Recovery above the line (IaaS and PaaS systems, applications and data) depends on recovery below the line (infrastructure).

Continuity for workloads hosted on Azure Stack

Workload continuity means becoming aware of Azure Stack features that can be used to enhance availability, the architecture of the application, and site availability to protect against disasters.

Start by understanding the data persistence model of your applications:

- *Stateful* workloads write data to one or more repositories. It is necessary to understand which parts of the architecture need point-in-time data protection and high availability to recover from a catastrophic event. Databases, file servers, and object stores are popular repositories that need a protection strategy. The ability to keep the data synchronized for high availability is limited to short distances to ensure low latency and maximum bandwidth. If a secondary site is geographically distant, you must rely on asynchronous replication or backup/restore techniques.
- *Stateless* workloads, on the other hand, don't contain data that needs to be protected. These workloads typically support on-demand scale-up and scale-down and can be deployed in multiple locations in a scale-out topology behind a load balancer. In Azure, these types of applications are deployed in multiple regions and configured behind Azure Traffic Manager. With Azure Stack, you must deploy the same application to multiple Azure Stack systems and have an external set of load balancers configured for high availability across multiple datacenters.

To support high availability for workloads within an Azure Stack system, multiple virtual machines are grouped into an *availability set*. Applications deployed in an availability set sit behind a load balancer that distributes incoming traffic randomly among multiple virtual machines. When you place multiple virtual machines under the same load balancer and within an availability set, traffic can be continuously served by at least one instance. This approach provides general availability of a workload during local network failures, local disk-hardware failures, and any planned downtime that the platform might require. Using an availability set ensures that Azure Stack locates the virtual machines in different *fault domains*. A fault domain in Azure Stack maps to a scale-unit node.

Across Azure Stack systems, a similar approach is possible with the following differences:

- The load balancer must be external to both systems or in Azure (for example, Azure Traffic Manager).
- Availability sets cannot span independent Azure Stack systems.

Data protection

The data protection strategy you implement for your application depends on how the application is designed. In Azure Stack, you can have IaaS-based or PaaS-based applications.

IaaS-based applications

These applications are deployed in one or more virtual machines and can reply on backups at the guest OS level or at the application level.

Data source is first consideration:

- **Disk.** Requires block-level backup of one, some, all disks exposed to the guest OS. Protects the entire disk and captures any changes at block level.
- **File or folder.** Requires file system-level backup of specific files and folders on one, some, or all volumes attached to the guest OS.
- **OS state.** Requires backup targeted at the OS state.
- **Application.** Requires a backup coordinated with the application installed in the guest OS. Application-aware backups typically includes quiescing I/O in the guest for application consistency (for example, Volume Shadow Copy Service (VSS) in the Windows OS).

Data churn is another consideration. This means the amount of data that needs to be protected after the first full backup and depends on the churn at the level of the disk, OS, and application. The churn rate over time is used for sizing purposes to understand the impact to the network (transfer) and backup storage (retention).

Backup of one of the following types is the next consideration:

- **Full.** A full backup is required when data protection is enabled for the first time. Full backups can place a large strain on the network and storage given the size of the payload.
- **Incremental or differential.** Backups that take place after the full backup can be incremental or differential. Which approach depends on what the data source supports natively. This backup type places less strain on the network and storage depending on the churn rate for the application.
    - Disk/volume: Tracking changes in disks or volumes is typically done at the block level using a filter-level driver installed in the guest OS.
    - Database: Databases typically support differential backups (that is, changes from the last full backup) or transaction log backups.
- **Synthetic full.** Some backup products support the ability to generate a full restore from the latest backup (even if it is incremental or differential). This type is ideal for backup since the payload on the network should always be smaller than the full size of the source. This backup type is a feature of the backup product used.

Topology is another consideration:

- An application contained within a single virtual machine can be safely backed up without coordinating across other virtual machines or applications.
- Applications or services that span multiple virtual machines must be analyzed to determine if the backup needs to be coordinated at the application level. Backing up each virtual machine individually—without any coordination at the application level—may result in inconsistencies that can cause the restore effort to fail. For example, clustered services such as Kubernetes do not benefit from a backup of the individual virtual machines in the cluster. Application-aware backup is ideal for applications that span multiple virtual machines.

PaaS-based applications

These types of application consume resources presented in an Azure Stack RP. For this type, there is no virtual machine to manage. Instead, services such as Azure App Services, Azure Functions, and Azure Database for MySQL are operated as part of Azure Stack. The applications that user deploy to these PaaS options are abstracted from the underlying infrastructure, like Azure. Therefore, data protection needs to be evaluated for each service the application consumes.

For example, for an application based on:

- **App Services.** If the application includes an App Services component, is it necessary to back up the instance of the deployed web app, or you can rely on an external code repository to maintain the production versions of the application.
- **SQL or MySQL services.** Backup of the database relies on services natively provided by SQL and MySQL to export and import a copy of the database.

Network considerations

When you're enabling protection for your users, network topologies matter. All traffic will flow north-bound through the top of rack switches and out the border devices. There are several important considerations to keep in mind.

Line of sight

Protecting at the level of the guest OS requires direct line of sight to the backup endpoint in one of the following ways:

- Guest OS connects to an external backup endpoint or appliance via NAT (the virtual machine does not have a public VIP).
- Guest OS connects over the public VIP.
- Guest OS connects over a virtual private network (VPN) configured for the virtual network.
- Guest OS connected over a GRE tunnel established through a set of RRAS VMs deployed in the same vNET.

Bandwidth

The bandwidth available to the virtual machine depends on the virtual machine size and the uplink network capacity:

- **Size.** Larger virtual machines sizes get a larger maximum network bandwidth. The bandwidth is not a guaranteed or reserved amount. This is strictly a maximum the VM can consume if the resources are available.

- **Uplinks.** An Azure Stack system can ship with 10 GB, 25 GB or faster switches depending on the type supported by the hardware vendor. Each node has two uplinks to the top of rack switches for fault tolerance. The system allocates half of the uplink capacity for critical infrastructure. The remainder is shared capacity for Azure Stack services and all user traffic. Systems deployed with faster speeds have more bandwidth available for backup traffic.

VPN or GRE tunnel

If a VPN or GRE tunnel is required to establish a connection the backup endpoint, the following considerations apply:

- Azure Stack ships with a multitenant VPN gateway presented as a resource in the portal. Connections through this gateway can have 100 Mbps or 200 Mbps of bandwidth depending on the SKU. For more information and details about limitations, see About VPN gateway for Azure Stack in the documentation.
- If users on Azure Stack require more bandwidth, they can use a network virtualization appliance for VPN or GRE tunnel services.
- You must validate that the border switches that will terminate the VPN or tunnel traffic can handle the number of connections and traffic.
- From a security perspective, establishing a VPN or tunnels to a common network means all IaaS VMs have access to the same environment. You must protect each virtual network with a network security group (NSG) on the Azure Stack side to ensure that users of one IaaS VM remain isolated from any other IaaS VM.

Multiple VM network adapters

Network traffic can be segregated using a second network adapter. Keep in mind the following considerations:

- All IaaS VM northbound traffic shares the same uplink. A second virtual network adapter will not segregate traffic at the physical transport level.
- Azure Stack aggregates all public IPs allocated to it into a single shared public VIP pool. There is no way to allocate a specific public IP or range of IPs to a specific virtual network adapter. For this reason, there's no way to guarantee a second virtual network adapter will get an IP used only for backup traffic.
- Azure Stack supports user-defined routes at a virtual network adapter level to segregate backup traffic over a specific adapter.

Storage considerations

Any protection strategy must consider the user and application data stored on Azure Stack. Azure Stack provides locally redundant storage (LRS) within the scale-unit that protects against component and node-level failures. The services that run on Azure Stack benefit from the underlying data resiliency. As an application architect and IT organization, you must design protections for the data contained in key services that store application data:

- IaaS virtual machines, including the related compute, storage, network objects, metadata, and Blob storage (including page blobs with virtual machine-attached disks).
- Blob storage, such as block blobs, tables, and queues.
- Azure Key Vault secrets and keys.

- App Services web applications.
- Azure Functions code.
- SQL and MySQL databases.
- Marketplace offerings, such as custom templates and images.

For these services, the scope of data protection is the resources themselves, not the service. In the event of a disaster, your application must run on a separate Azure Stack system, so the application data must be available using a manual copy or a product that replicates data.

NOTE Azure Stack doesn't offer replication of blob data across independent Azure Stack systems. Replication of object stores across two independent Azure Stack systems isn't equivalent to Azure geo-redundant storage that replications data between two paired Azure regions.

When your recovery target is a secondary Azure Stack unit or the Azure cloud, you can manually export data using blob snapshots, blob copy, AzCopy Export, App Service Backup, and SQL export tools. Automated migration of data from in-guest solutions to Azure Storage or other appliances can also be considered as part of your strategy.

For more information, see Azure Stack storage: Differences and considerations.

Disaster recovery and application availability

Disaster recovery implies availability of a secondary location that can host your application and data. The secondary location can be another datacenter or the public Azure cloud. To prepare for a disaster, here are a few key considerations:

- **Services are local.** Azure Stack services are local to the system deployed. These services include the portal, Azure Resource Manager, and resource providers. This system isn't aware of another Azure Stack system deployed in an adjacent rack or in a different datacenter. Each system is independent and uses its own instances of the portal, Resource Manager, and resource providers.
- **Secondary sites.** To prepare for a disaster, you must have two or more Azure Stack systems deployed (unless you plan to use Azure). These systems must be fully deployed and registered. Only hardware partners can deploy Azure Stack. Don't expect to deploy Azure Stack on your own in response to a disaster or catastrophic event.
- **Traditional apps.** Traditional IaaS-based applications entirely contained within one virtual machine, or a few, can be either redeployed to a secondary system, restored from a backup, or replicated in real-time using in-guest or application-level technologies.
- **Scale-out apps.** Modern IaaS-based applications that support scale-out architectures or primarily host stateless applications can be deployed in the secondary location in addition to the primary location. You can place the instances of the application behind a load balancer and decide if the application is active on both sites (active-active) or just one (active-passive).
- **Data storage.** Persistent data stores, such as Microsoft SQL Server, use native replication or transaction log shipping to send data to a secondary site. SQL Server Always On capability, distribution groups, and availability groups can be used to sync data across sites and switch the primary and secondary status after an event. Replicated data should always be stored at an external site with a network line of sight to the target.

For example, Figure 8 shows the primary site (A) with a deployment of SQL Server in an availability set, which provides high availability within the scale-unit using synchronous replication. In the event of a disaster, a standalone replica of SQL Server is deployed within an availability group in a secondary site (B). If you create a single availability group across the sites, the cluster is effectively stretching across the sites. Alternatively, you can use log shipping, which requires multiple manual steps. If necessary, a site-to-site VPN is configured between the sites. For more information, see Business continuity and database recovery - SQL Server.

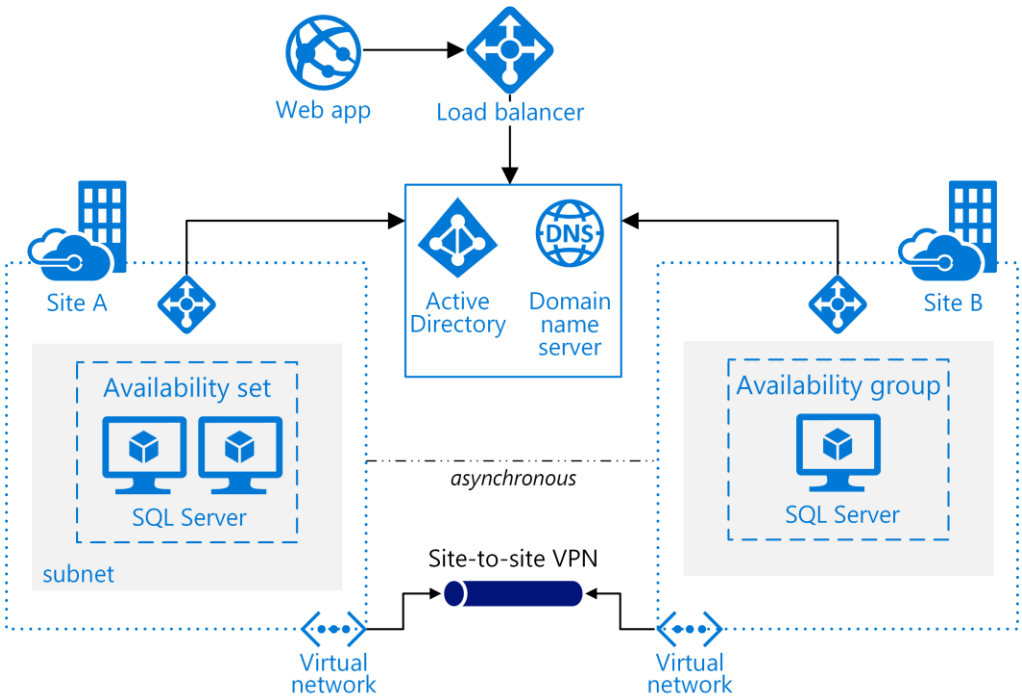


Figure 8. This recovery scenario provides high availability for SQL Server within a scale-unit and replication to a secondary site in the event of a disaster in the primary site

You can use a distributed availability group to replicate data across two independent SQL clusters (instead of using a single, multi-site SQL cluster). In this scenario, the primary and secondary sites are both configured with pairs of VMs in availability sets running the SQL Server in a local availability group for local high availability. A distributed group is layered on top of the two availability groups in the two sites. If the primary site goes down, a manual failover workflow must be initiated so the secondary site is ready. For more information, see the Quick start templates on GitHub for deploying SQL Server with high availability in an Always On availability group on Azure Stack.

If you use Azure as a secondary site for Azure Stack, you can use the Azure Site Recovery service to replicate guest OS and data disks to Azure. You can then support planned and unplanned failover and validate to test that it works. You can also enable protection again after failover. For details, see Replicate Azure Stack VMs to Azure.

PaaS recovery scenarios

PaaS-based applications can be deployed to be highly available within a site. For disaster recovery, they need to be recovered from a second location. Recovery of the application is separate from recovery of the services used, such as App Services, SQL Server, and MySQL. You must invest each layer with solutions for high availability and disaster recovery.

Service layer

The resource providers and hosting servers deployed locally in a scale-unit can be deployed in a highly available configuration. To learn how, refer to the resource provider documentation. The configuration can include database servers and file servers used by the resource provider.

Application layer

For a PaaS-based application, you need to consider how each tier of the application can be made highly available within a scale-unit and across two or more Azure Stack systems. The tiers include:

- **App Services.** For a web application, you can use an external load balancer, such as Azure Traffic Manager or an enterprise load balancer, deployed in a high-availability pair in your datacenter.
- **Database services.** For SQL Server or MySQL resource providers, you must use backup and restore of databases in a hosting server. The instance must be manually created in the secondary location. Otherwise, if you have decided to manage your own instances of SQL Server on virtual machines, you can use availability groups and distributed availability groups for high availability and disaster recovery with manual failover.

Figure 9 shows the PaaS layer including separate instances of the App Services resource provider, which are deployed in a high-availability configuration. Each site and database layer is configured for high availability and disaster recovery across sites.

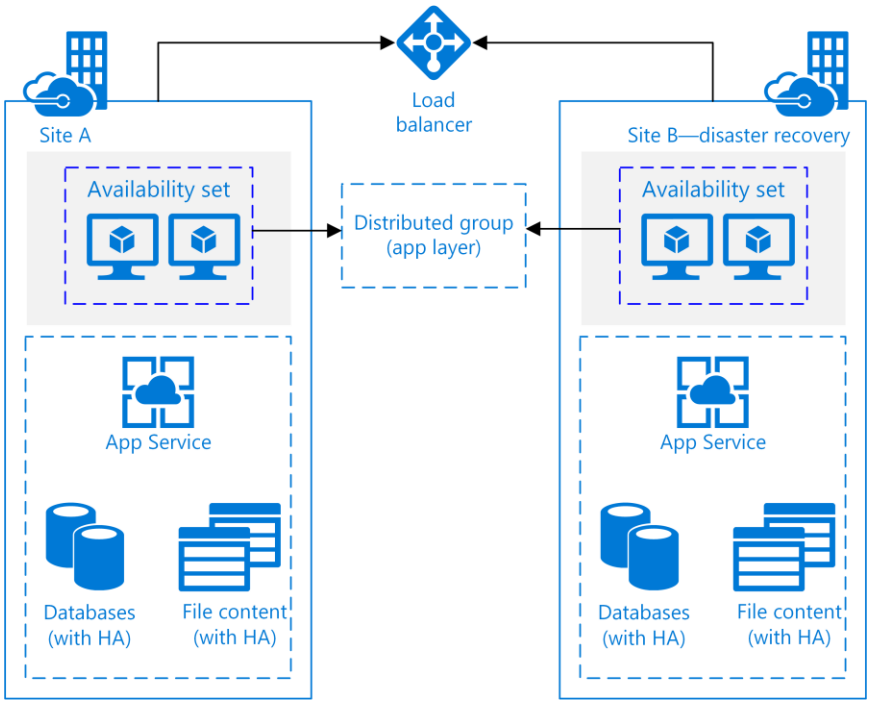


Figure 9. The App Service resource provider is deployed on each site

Modern operations, applications, and hybrid patterns

Some businesses focus on deploying or migrating simple applications to VMs on Azure Stack. But increasingly, more organizations want to take advantage of services in Azure Stack and Azure (for connected environments) so they can modernize existing applications and embrace the capabilities of IaaS VMs. Monitoring, security, data protection, and other services add value to your application and help you step away from traditional monolithic design patterns. For more information on modernizing your IaaS VMs, see the Azure Stack IaaS VM blog series.

With Azure and Azure Stack, applications can be designed, deployed, and managed as cloud natives and consume PaaS resources in addition to IaaS resources. For example, applications that are designed for containers and microservices are optimized for rapid deployment and simple scale-in and scale-out. Azure Stack supports the virtuous life cycle of application development and operations called *DevOps*. In Azure Stack, developers can quickly create development environments and integrate them with the wide range of tools in use by the organization. They can create applications efficiently, deploy them to the cloud easily, and update them on a continuous basis.

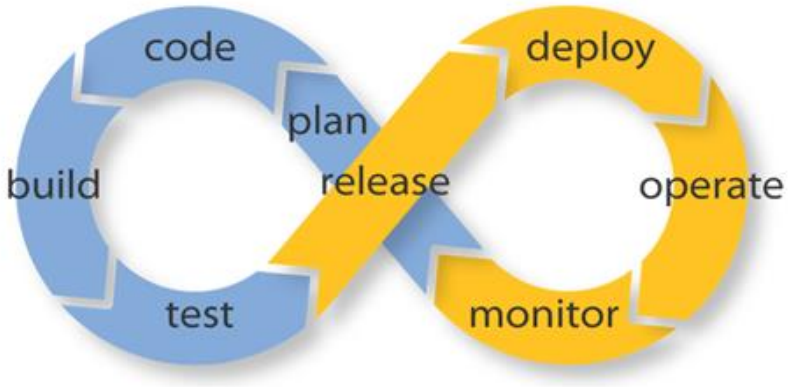


Figure 10. DevOps processes automate software delivery and speed the technology deployment life cycle

DevOps works through a process of continuous integration (CI) and continuous delivery (CD). Whenever a developer checks in code to their source repository, a build can be triggered automatically—the CI step. After a build and automated unit tests are successful, the application can be deployed automatically to an environment where the developer can do more in-depth testing—the CD step. A CI/CD pipeline on Azure Stack automates the build, test, and deployment phases of application development.

In addition, the use of external repositories and CI/CD pipelines for development and deployment of your workloads ensures that applications can be deployed quickly in the event of a disaster or interruption. CI/CD processes decouple the running solution from the code-based assets that enable it.

For more information, see the following tutorials for hybrid cloud patterns:

- Deploy apps to Azure and Azure Stack
- Configure hybrid cloud identity for Azure and Azure Stack applications
- Configure hybrid cloud connectivity with Azure and Azure Stack
- Create a staged data analytics solution with Azure and Azure Stack
- Create cross-cloud scaling solutions with Azure
- Create a geo-distributed app solution with Azure and Azure Stack
- Deploy a hybrid cloud solution with Azure and Azure Stack

Get started with the Azure Stack Development Kit

Application development in the cloud enables you to minimize the cost of maintaining a test environment because you only pay for the environment resources if you're using them. To help you get started, the Azure Stack Development Kit (ASDK) provides an evaluation environment. ASDK is a single-node deployment of Azure Stack that you can download and use for free. You can develop modern applications using APIs and tooling consistent with Azure in a non-production environment. For details, see What is the ASDK?

Template-based deployment using Azure Resource Manager

At the core of Azure Stack, Azure Resource Manager provides an API that allows a wide variety of user interfaces to communicate with the platform. This API gives you powerful infrastructure-as-code capabilities to deploy and configure any type of resource that is available on Azure Stack. In addition, with a single template, you can deploy and configure a complete application to an operational end state.

A Resource Manager template is a JavaScript Object Notation (JSON) file that defines one or more resources to deploy to an Azure resource group. It also defines the dependencies between the resources.

For an introduction to Azure Resource Manger templates, see Template deployment in the Azure Resource Manager overview.

Cloud recovery

Cloud resiliency is vital for the high availability and disaster recovery of your application and data on Azure Stack. However, at some point you must also consider the impact of an Azure Stack system going offline.

Azure Stack supports a deployment mode called *cloud recovery* designed to seed an Azure Stack deployment with critical metadata. Cloud recovery uses a backup of the original Azure Stack system. The cloud recovery deployment mode is a top-down recovery of the foundational Azure Stack services and helps accelerate onboarding of users back to a recovered Azure Stack system. Cloud recovery includes restoring the subscriptions, plans, offers, quotas, Key Vault secrets, RBAC, and resource groups. User data, including virtual machine disks and databases, isn't restored as part of the backup. This backup is focused on internal infrastructure services.

Two important considerations apply to cloud recovery:

- Infrastructure backup must be configured on the original Azure Stack system to generate a backup. This backup data is strongly coupled to the system, so it can be used only to seed the recovery of a replacement system. It can't be used to clone an Azure Stack instance.
- Cloud recovery is a deployment mode that can be initiated by hardware partners. Using cloud recovery for deployment and restoration is a major event. The expectation is that the original system is no longer available.

> **NOTE** Given the extended time required to receive replacement systems (if required) and to schedule professional services from the hardware vendor, who must be on site for deployment, you must have a strategy for high availability and disaster recovery for your applications and data **that is separate** from how the underlying system is redeployed.

Infrastructure backup in cloud recovery

To support cloud recovery, you must enable infrastructure backup. This internal service ships with Azure Stack and orchestrates backup across a subset of services to an external file share you provide. As Figure 11 shows, the external share can be a remote target, Azure, or your own datacenter.

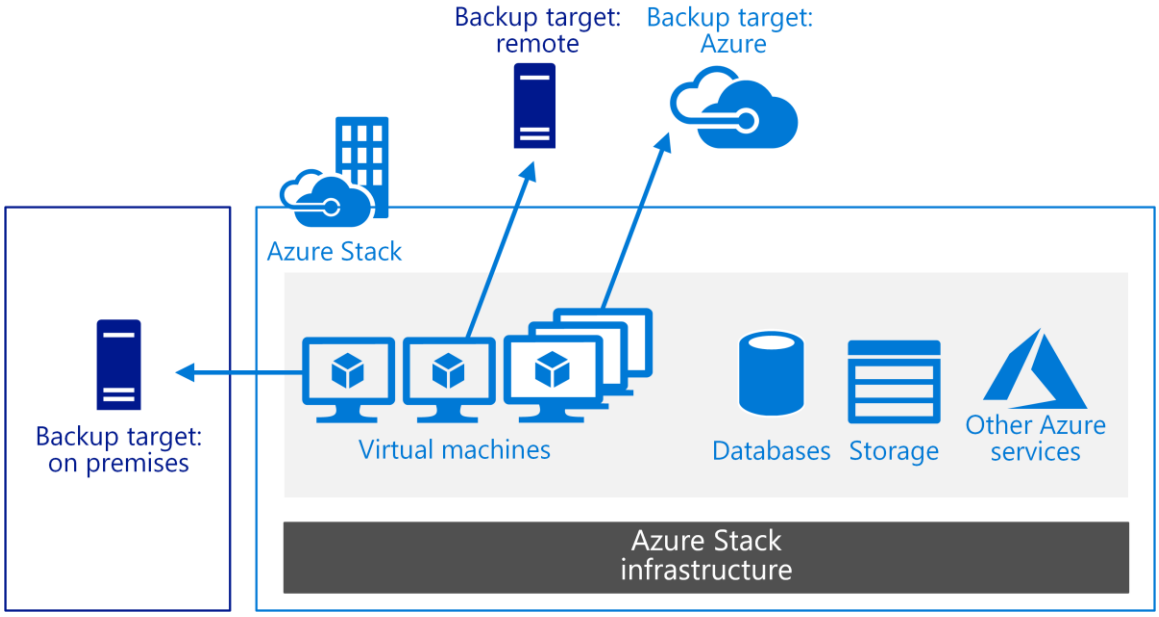


Figure 11. Store backups externally, using a remote target, Azure, or your own datacenter

You need at least 1 TB of available capacity per cloud for the file share to store the last seven days of infrastructure backups. More than seven days isn't necessary because you need just the last good backup for cloud recovery. For the list of data in the backup, See Recover from catastrophic data loss.

This backup does not automatically include network switch data or the HLH. The OEM is responsible for providing guidance on how to back up and when to restore these components. In a cloud-recovery scenario, these components get deployed as a new solution. You can manually

back up data from the HLH or the switches to the same external file share used for infrastructure backup.

The operator of the Azure Stack is responsible for provisioning and managing the file share that will contain the Azure Stack infrastructure backups. The file server used for your Azure Stack backup and disaster recovery plan needs to meet the minimum requirements in the Azure Stack documentation.

Summary

Azure Stack is a unique offering unlike other virtualization platforms you may have used, and its business continuity and disaster recovery strategy is unique as well. With Azure Stack, your entire cloud can start with a small configuration—as small as four servers in a single rack to multiple systems in multiple locations. To minimize the impact of a local failure, system outage, catastrophic event, or a large-scale disaster, you need to think like a cloud provider, beyond the virtualization skillset. That means:

- Determine the level of availability you want to design for in your organization—one system, multiple systems in a datacenter, or multiple systems in multiple datacenters.
- Collaborate with the application owners and identify the type of applications they want to deploy or migrate to Azure Stack.
- Determine the recovery objective for each application type.
- Identify the protection products that will help you accomplish those recovery objectives.
- Articulate the protection and recovery workflows that guarantee you can achieve the recovery objective defined in collaboration with the application owners.

Learn more

Azure Stack IaaS – part one (Azure blog)

Protecting applications and data on Azure Stack (Azure blog)

Backup and data recovery for Azure Stack with the Infrastructure Backup Service

Replicate Azure Stack VMs to Azure

Understanding architectural patterns and practices for business continuity and disaster recovery on Microsoft Azure Stack (Microsoft Ignite 2018 video)

Manage the availability of Windows virtual machines in Azure

Key features and concepts in Azure Stack

Tutorial: Create and deploy highly available virtual machines with Azure PowerShell

List of all the BC/DR partners with validated offers for Azure Stack