100 XP

Introduction

3 minutes

Meet Tailwind Traders

As the Tailwind Traders Enterprise IT team prepares to define the strategy to migrate some of company's workloads to Azure, it must identify the required SLAs and high availability requirements for various business critical workloads. Considering the US-wide scope of its operations, Tailwind Traders will be using multiple Azure regions to host its applications. Most of these applications have dependencies on infrastructure and data services, which will also reside in Azure. Tailwind Traders CTO has asked you to identify the availability requirements of their Azure workloads, as well as recommend a high availability solution for compute, relational data store, and non-relational data store for all new workloads that will be deployed in Azure.

Learning objectives

In this module, you'll be able to:

- Identify the availability requirements of Azure resources
- Design for Azure Front door
- Design for Azure Traffic Manager
- Recommend a high availability solution for Compute
- Recommend a high availability solution for relational data storage
- Recommend a high availability solution for non-relational data storage

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design Business Continuity Solutions

Design for High Availability

- Identify the availability requirements of Azure resources
- Recommend a high availability solution for Compute
- Recommend a high availability solution for non-relational data storage
- Recommend a high availability solution for relational data storage

Prerequisites

- Working experience with designing highly available architectures
- Conceptual knowledge of high availability for applications, databases, and compute resources

Next unit: Identify the availability requirements of Azure resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Identify the availability requirements of Azure resources

3 minutes

Organizations, such as Tailwind Traders, require a high degree of availability from their mission-critical applications. It's also vital that any mission-critical application, and its associated data, is designed in such a way that in the case of a region outage it can automatically failover and continue running in another region.

Highly available workloads are those which are:

- Resilient to component failure
- Highly available, and can run in a healthy state with no significant downtime

To achieve the desired resilience and high availability, you must first define your requirements.

Workload availability targets

Define your own target SLAs for each workload in your solution so you can determine whether the architecture meets the business requirements.

Consider cost and complexity

Everything else being equal, higher availability is better. But as you strive for more nines, the cost and complexity grow. An uptime of 99.99% translates to about five minutes of total downtime per month. Is it worth the additional complexity and cost to reach five nines? The answer depends on the business requirements.

Here are some other considerations when defining an SLA:

- To achieve four nines (99.99%), you can't rely on manual intervention to recover from failures. The application must be self-diagnosing and self-healing.
- Beyond four nines, it's challenging to detect outages quickly enough to meet the SLA.

- Think about the time window that your SLA is measured against. The smaller the window, the tighter the tolerances. It doesn't make sense to define your SLA in terms of hourly or daily uptime.
- Consider the mean time between failures (MTBF) and mean time to recover (MTTR) measurements. The higher your SLA, the less frequently the service can go down and the quicker the service must recover.
- Get agreement from your customers for the availability targets of each piece of your application, and document it. Otherwise, your design may not meet the customers' expectations.

Identify dependencies

Perform dependency-mapping exercises to identify internal and external dependencies. Examples include dependencies relating to security or identity, such as Active Directory, or third-party services such as a payment provider or e-mail messaging service.

Pay particular attention to external dependencies that might be a single point of failure or cause bottlenecks. If a workload requires 99.99% uptime but depends on a service with a 99.9% SLA, that service can't be a single point of failure in the system. One remedy is to have a fallback path in case the service fails. Alternatively, take other measures to recover from a failure in that service.

The following table shows the potential cumulative downtime for various SLA levels.

SLA	Downtime per week	Downtime per month	Downtime per year
99%	1.68 hours	7.2 hours	3.65 days
99.9%	10.1 minutes	43.2 minutes	8.76 hours
99.95%	5 minutes	21.6 minutes	4.38 hours
99.99%	1.01 minutes	4.32 minutes	52.56 minutes
99.999%	6 seconds	25.9 seconds	5.26 minutes

Every organization has unique requirements, and you should design your applications to best meet your complex business needs. Defining a target SLA will make it possible to evaluate whether the architecture meets your business requirements. Some things to consider include:

- What are the availability requirements?
- How much downtime is acceptable?
- How much will potential downtime cost your business?
- How much should you invest in making the application highly available?
- What are the data backup requirements?
- What are the data replication requirements?
- What are the monitoring requirements?
- Does your application have specific latency requirements?

For additional guidance, refer to Principles of the reliability pillar.

Identify critical system flows

Understanding critical system flows is vital to assessing overall operational effectiveness and should be used to inform a health model for the application. It can also tell if paths of the application are over or underutilized and should be adjusted to better meet business needs and cost goals.

Critical sub-systems or paths through the application may have higher expectations around availability, recovery, and performance due to the criticality of associated business scenarios and functionality. This also helps to understand if cost will be affected due to these higher needs.

Identify less critical components

Some less critical components or paths through the application may have lower expectations around availability, recovery, and performance. This can result in cost reduction by choosing lower SKUs with less performance and availability.

Availability metrics

Use these measures to plan for redundancy and determine customer SLAs.

- **Mean time to recover (MTTR)** is the average time it takes to restore a component after a failure.

- **Mean time between failures (MTBF)** is the how long a component can reasonably expect to last between outages.

Understand service-level agreements

In Azure, the Service Level Agreement describes Microsoft's commitments for uptime and connectivity. If the SLA for a particular service is 99.9%, you should expect the service to be available 99.9% of the time. Different services have different SLAs.

The Azure SLA also includes provisions for obtaining a service credit if the SLA is not met, along with specific definitions of availability for each service. That aspect of the SLA acts as an enforcement policy.

The Service Level Agreement Estimator sample shows how to calculate the SLA of your architecture.

Next unit: Design for Azure Front Door

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

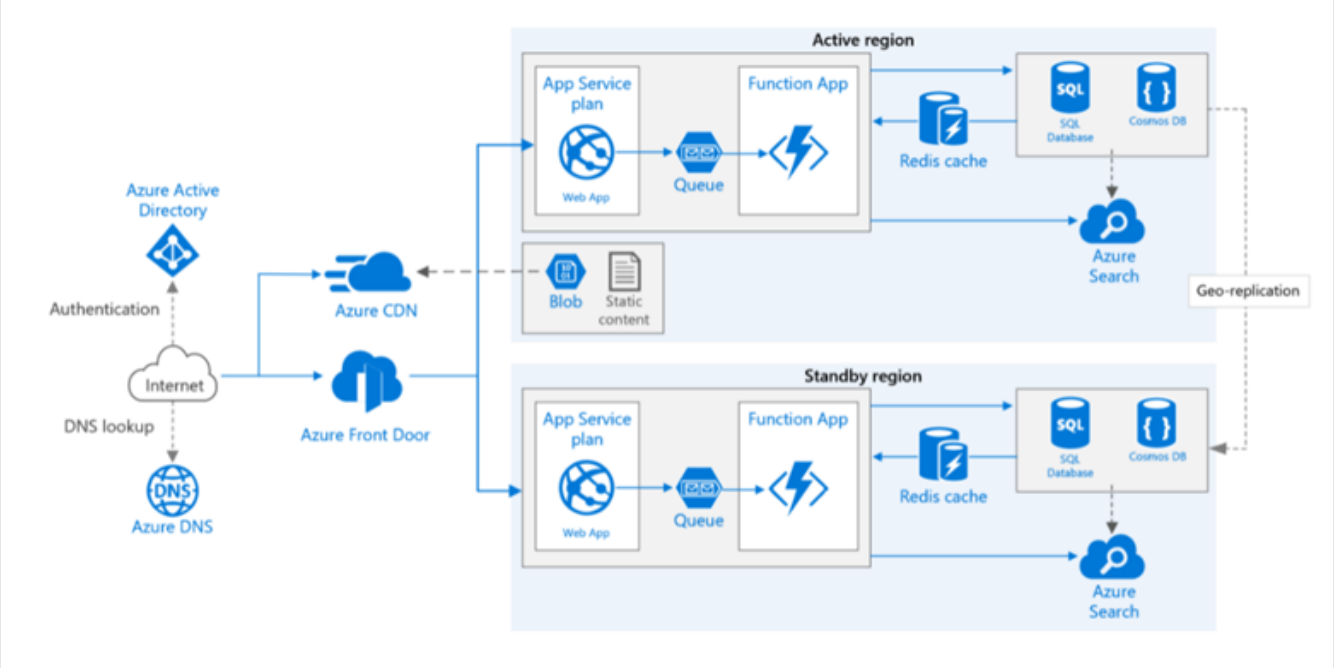
Design for Azure Front Door

3 minutes

Azure Front Door offers a fast, reliable, and secure modern cloud Content Delivery Network (CDN) by using the Microsoft global edge network to integrate with intelligent threat protection. Azure Front Door optimizes access times to content. Front Door can be used to provide another layer of reliability in front of your Azure resources. It is an application delivery network that provides global load balancing and site acceleration service for web applications. It offers Layer 7 capabilities for your application like SSL offload, path-based routing, fast failover, caching, etc. to improve performance and high-availability of your applications.

How Azure Front Door works in reliability scenarios

In the following graphic, users are connecting to an app hosted in the custom domain Contoso.com. Azure Front Door is implemented at the edge location. Initially, the app is hosted in the primary region (marked active in the graphic). Front Door routes incoming requests to that region. However, if the app running in that region becomes unavailable, Front Door fails over to the secondary region (shown as standby). Azure Front Door refers to this strategy as priority-based traffic-routing.



This architecture builds on the following:

- **Primary and secondary region.** This architecture uses two regions to achieve higher availability. The app is deployed to each region. During normal operations, network traffic is routed to the primary region. If the primary region becomes unavailable, traffic is routed to the secondary region.
- **Front Door.** Front Door routes incoming requests to the primary region. If the application running that region becomes unavailable, Front Door fails over to the secondary region.
- **Geo-replication.** Geo-replication of SQL Database and/or Cosmos DB.

A multi-region architecture can provide higher availability than deploying to a single region. If a regional outage affects the primary region, you can use Front Door to fail over to the secondary region.

> **Tip**
>
> This architecture can also help if an individual subsystem of the application fails.

High availability scenarios

There are several general approaches to achieving high availability across regions.

Approach	Description
Active/passive with hot standby	Traffic goes to one region, while the other waits on hot standby. Hot standby means the VMs in the secondary region are always running.
Active/passive with cold standby	Traffic goes to one region, while the other waits on cold standby. Cold standby means the VMs in the secondary region aren't allocated until needed for failover. This approach costs less to run but will generally take longer to come online during a failure.
Active/active	Both regions are active, and requests are load balanced between them. If one region becomes unavailable, it's taken out of rotation.

Next unit: Design for Azure Traffic Manager

Continue

100 XP

Design for Azure Traffic Manager

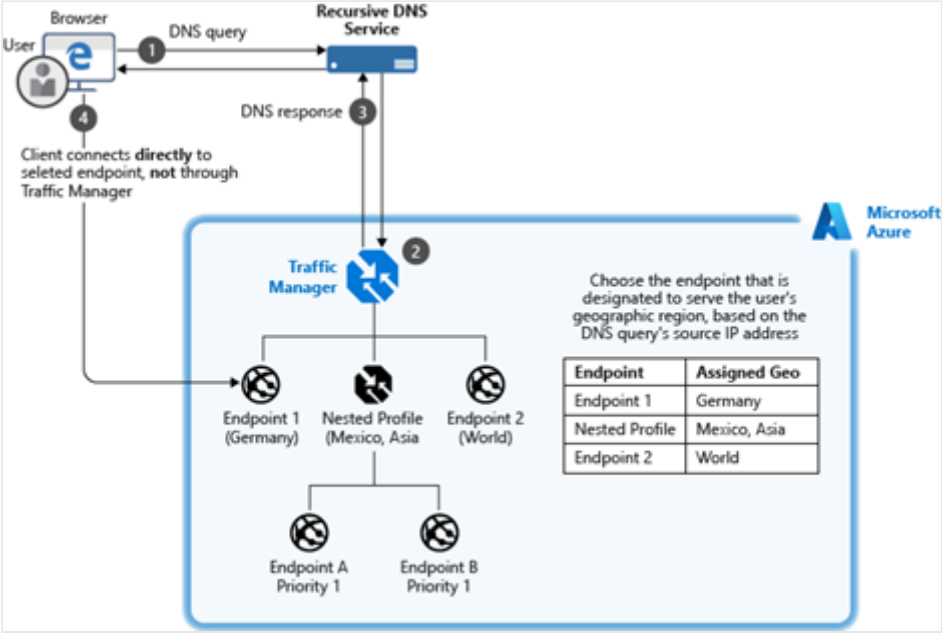
3 minutes

Azure Traffic Manager a DNS-based traffic load balancer. This service allows you to distribute traffic to your public facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness. It is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness. Because Traffic Manager is a DNS-based load-balancing service, it load balances only at the domain level. For that reason, it can't fail over as quickly as Front Door, because of common challenges around DNS caching and systems not honoring DNS TTLs.

How Azure Traffic Manager works in reliability scenarios

Traffic Manager uses DNS to direct the client requests to the appropriate service endpoint based on a traffic-routing method. You define these services by using Traffic Manager endpoints. Each endpoint is the service load balancer IP. You can use this configuration to direct network traffic from the Traffic Manager endpoint in one region to the endpoint in a different region. This is often referred to as geographic routing. For example, the following graphic depicts a typical scenario:

1. The user petitions a DNS server.
2. The DNS server queries Traffic Manager for the required record(s).
3. The result is returned from Traffic Manager.
4. The client connects directly to the defined endpoint.



High availability scenarios

Traffic Manager provides for high availability for your critical apps. It does so by monitoring your endpoints and providing automatic failover when an endpoint goes down. Traffic Manager, working with Azure DNS, enables a few failover approaches, described in the following table.

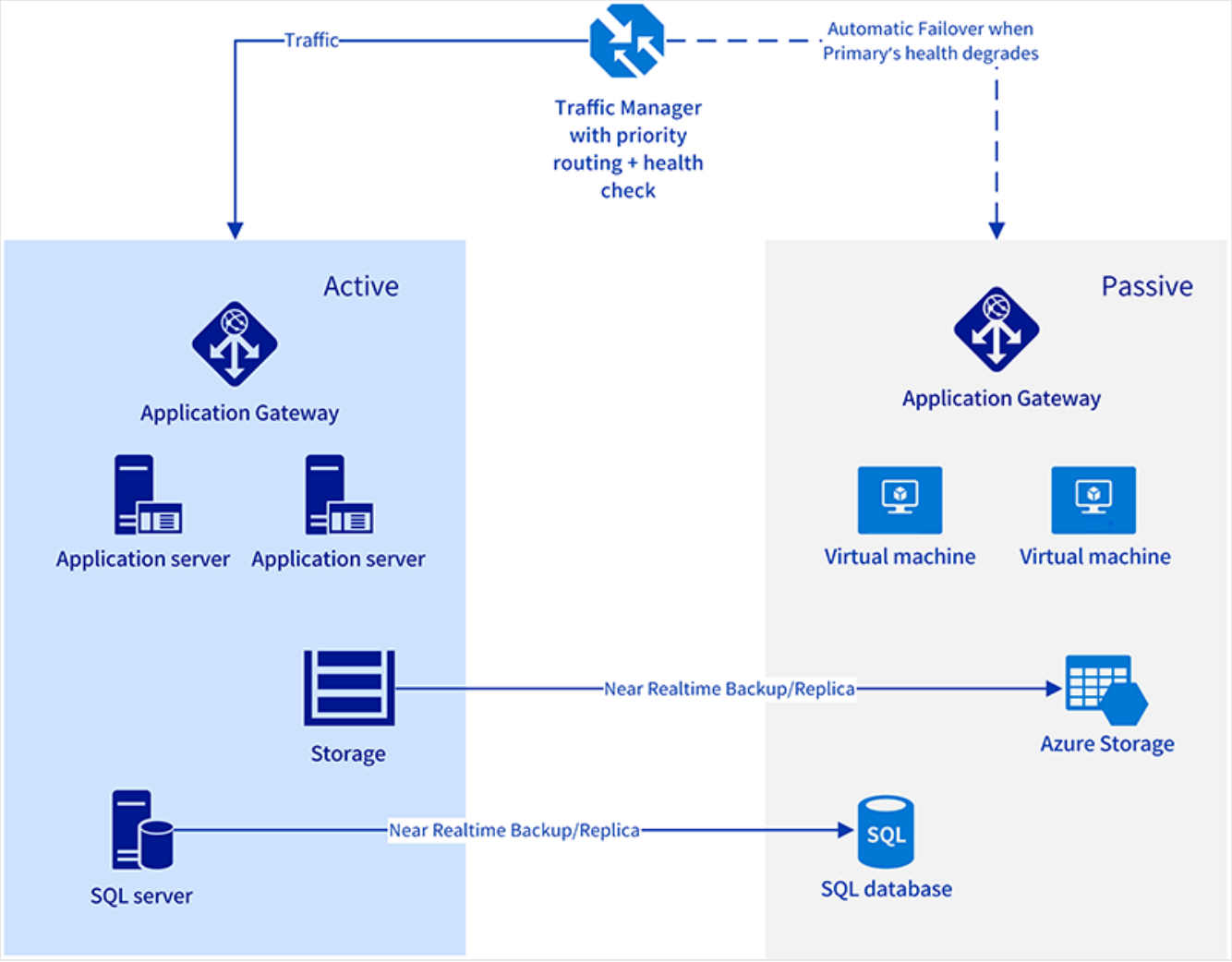
Approach	Description
Active/Passive with cold standby	Your VMs (and other appliances) that are running in the standby region aren't active until needed. However, your production environment is replicated to a different region. This approach is cost-effective but takes longer to undertake a complete failover.
Active/Passive with pilot light	You establish the standby environment with a minimal configuration; it has only the necessary services running to support a minimal and critical set of apps. In its default form, this approach can only execute minimal functionality. However, it can scale up and spawn more services, as needed, to take more of the production load during a failover.
Active/Passive with warm standby	Your standby region is pre-warmed and is ready to take the base load. Auto scaling is on, and all the instances are up and running. This approach isn't scaled to take the full production load but is functional, and all services are up and running.

Depending on what you want to achieve, and which approach suits your needs, you can implement failover:

- Manually, by using Azure DNS, this failover solution uses the standard DNS mechanism to fail over to your backup site. This option works best when used in conjunction with the cold standby or the pilot light approaches.

- Automatically, by using Traffic Manager, with more complex architectures and multiple sets of resources capable of performing the same function, you can configure Azure Traffic Manager (based on DNS). Traffic Manager checks the health of your resources and routes the traffic from the non-healthy resource to the healthy resource automatically.

In the following graphic, both the primary region (active) and the secondary region (passive) have a full deployment. This includes the cloud services and a synchronized database. Only the primary region is actively handling network requests from the users. The secondary region becomes active only when the primary region experiences a service disruption. In that case, all new network requests are routed by Traffic Manager to the secondary region.



100 XP

Recommend a high availability solution for compute

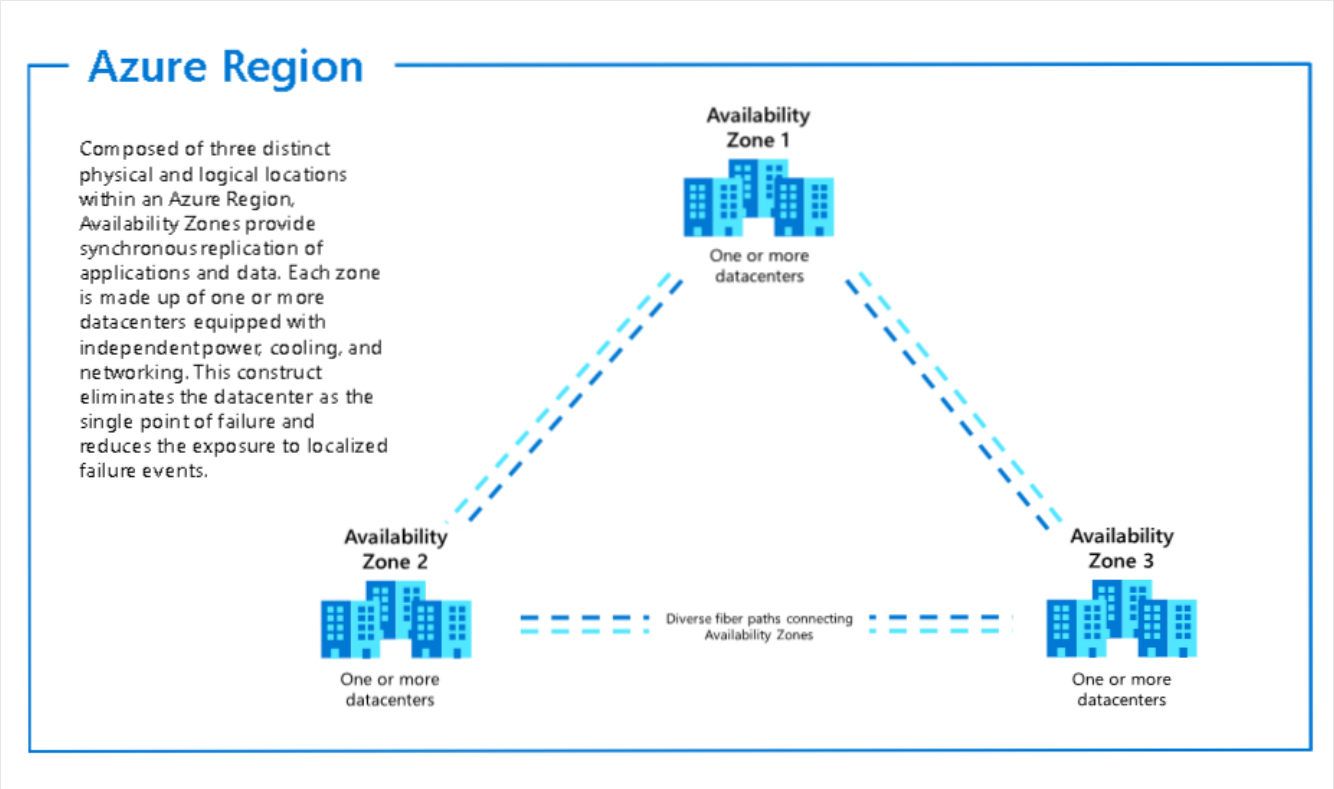
3 minutes

Microsoft Azure global infrastructure is designed and constructed at every layer to deliver the highest levels of redundancy and resiliency to its customers. Azure infrastructure is composed of geographies, regions, and Availability Zones, which limit the blast radius of a failure and therefore limit potential impact to customer applications and data. The Azure Availability Zones construct was developed to provide a software and networking solution to protect against datacenter failures and to provide increased high availability (HA) to our customers.

Design with Azure Availability Zones

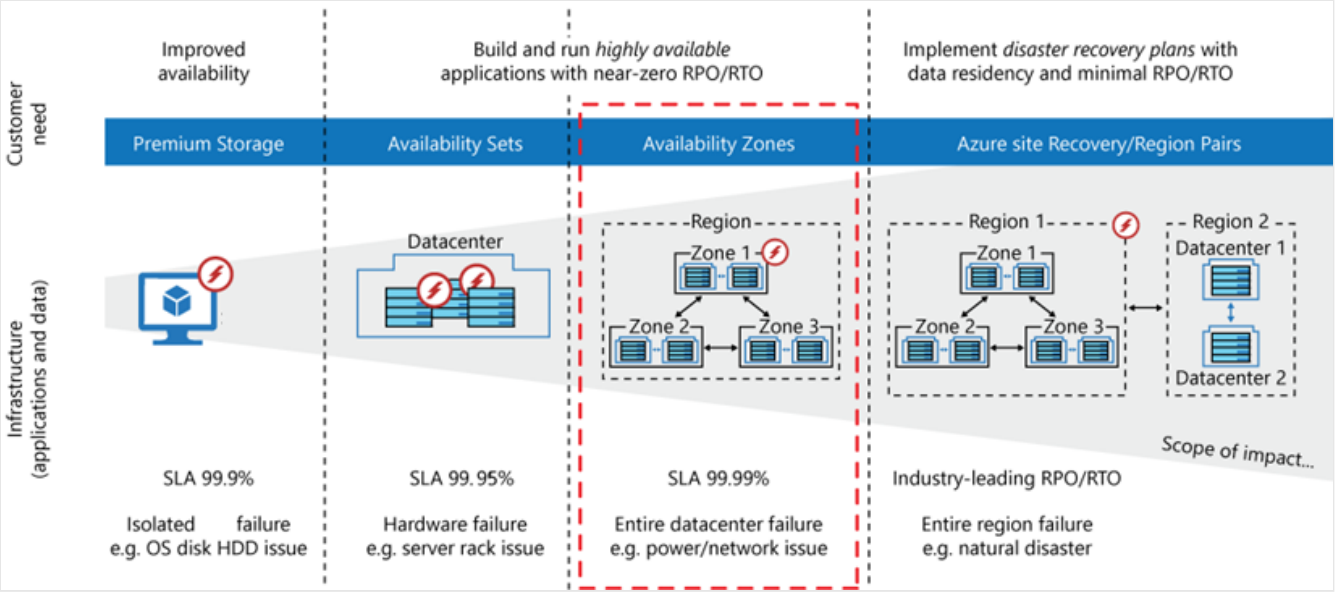
Availability Zones are unique physical locations within an Azure region. Each zone is made up of one or more datacenters with independent power, cooling, and networking. The physical separation of Availability Zones within a region limits the impact to applications and data from zone failures, such as large-scale flooding, major storms and superstorms, and other events that could disrupt site access, safe passage, extended utilities uptime, and the availability of resources. Availability Zones and their associated datacenters are designed such that if one zone is compromised, the services, capacity, and availability are supported by the other Availability Zones in the region.

Availability Zones can be used to spread a solution across multiple zones within a region, allowing for an application to continue functioning when one zone fails. With Availability Zones, Azure offers industry best 99.99% Virtual Machine (VM) uptime service-level agreement (SLA) . Zone-redundant services replicate your services and data across Availability Zones to protect from single points of failure.



With Availability Zones, Azure offers industry best 99.99% VM uptime SLA. The full Azure SLA explains the guaranteed availability of Azure as a whole.

The following diagram illustrates the different levels of HA offered by a single VM, Availability Sets, and Availability Zones.



Using a VM workload as an example, a single VM has an SLA of 99.9%. This means the VM will be available 99.9% of the time. Within a single datacenter, the use of Availability Sets can increase the level of SLA to 99.95% by protecting a set of VMs, ensuring they will not all be on the same hardware. Within a region, VM workloads can be distributed across Availability Zones to increase the SLA to 99.99%. For more information, refer to Availability options for VMs in Azure.

Azure services supporting Availability Zones fall into two categories: zonal services and zone redundant services. Customer workloads can be categorized to utilize either architecture scenario to meet application performance and durability requirements.

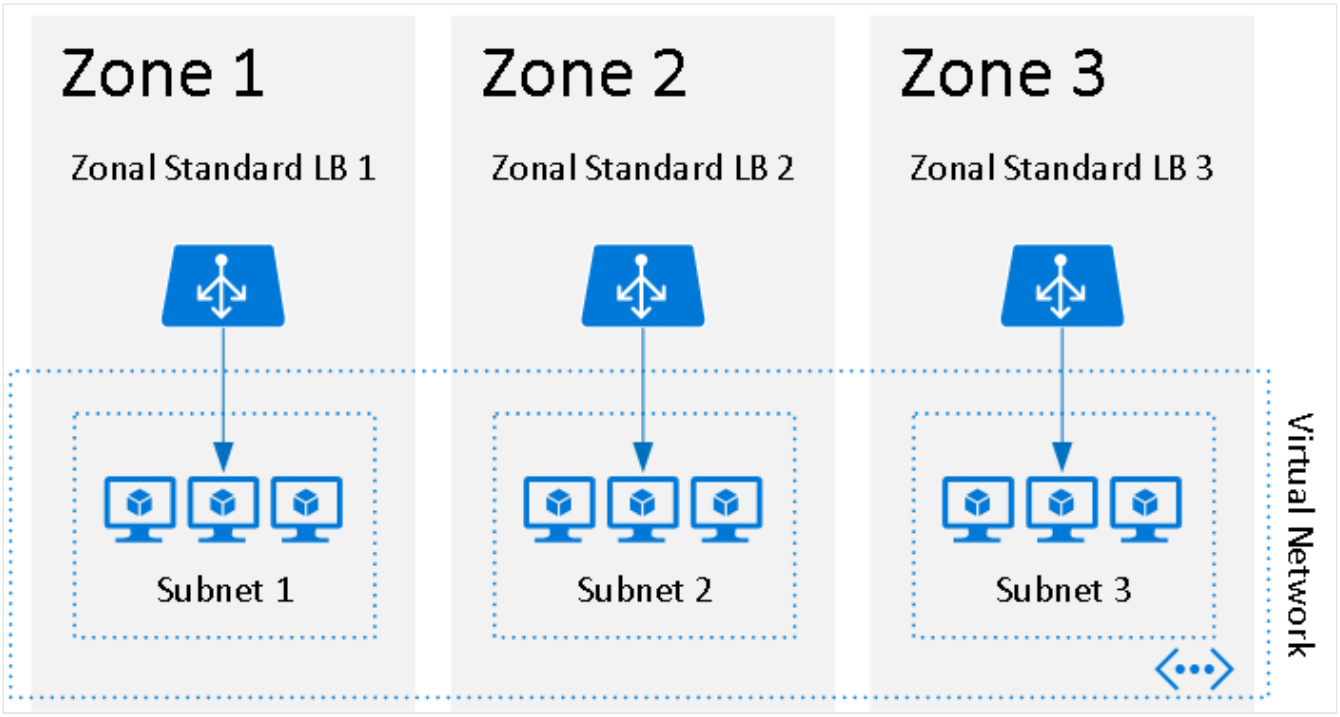
With zonal architecture, a resource can be deployed to a specific, self-selected Availability Zone to achieve more stringent latency or performance requirements. Resiliency is self-architected by replicating applications and data to one or more zones within the region. You can choose specific Availability Zones for synchronous replication, providing high availability, or asynchronous replication, providing backup or cost advantage. You can pin resources-for example, virtual machines, managed disks, or standard IP addresses-to a specific zone, allowing for increased resilience by having one or more instances of resources spread across zones.

With zone-redundant architecture, the Azure platform automatically replicates the resource and data across zones. Microsoft manages the delivery of high availability, since Azure automatically replicates and distributes instances within the region.

A failure to a zone affects zonal and zone-redundant services differently. In the case of a zone failure, the zonal services in the failed zone become unavailable until the zone has recovered. By architecting your solutions to use replicated VMs in zones, you can protect your applications and data from a zone becoming unavailable-for example, due to a power outage. If one zone is compromised, replicated apps and data are instantly available in another zone.

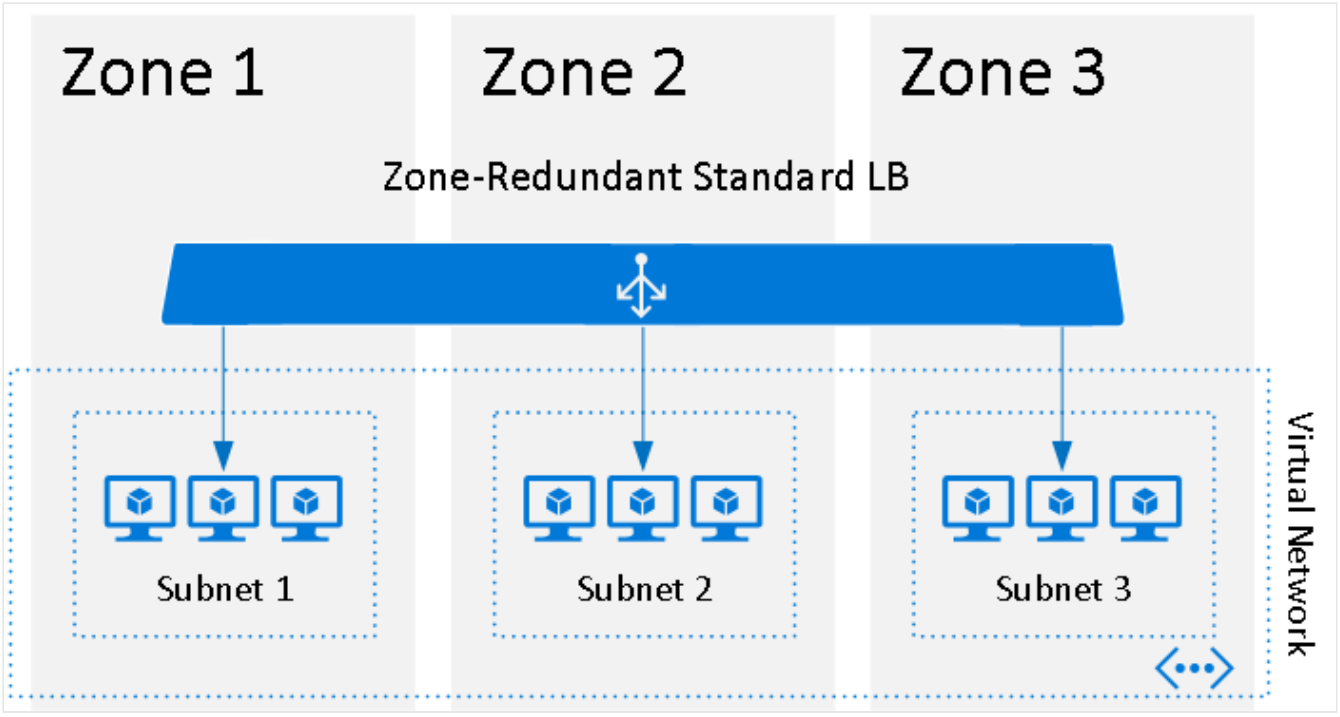
Zonal architecture applies to a specific resource, typically an infrastructure as a service (IaaS) resource, like a VM or managed disk, as illustrated. For example, zonal load balancer, VM, managed disks, virtual machine scale sets.

In the illustration, each VM and load balancer (LB) are deployed to a specific zone.



With zone-redundant services, the distribution of the workload is a feature of the service and is handled by Azure. Azure automatically replicates the resource across zones without requiring your intervention. ZRS, for example, replicates the data across three zones so a zone failure does not impact the HA of the data.

The following illustration is of a zone-redundant load balancer.



For a list of Azure services that support Availability Zones, per Azure region, refer to the Availability Zones documentation.

Would you recommend Tailwind use Zonal or Zone redundant architecture for their workloads? Why would you recommend one over the other?

When to select virtual machine scale sets

Scale sets are built from virtual machines. With scale sets, the management and automation layers are provided to run and scale your applications. You could instead manually create and manage individual VMs, or integrate existing tools to build a similar level of automation. The following table outlines the benefits of scale sets compared to manually managing multiple VM instances.

Scenario	Group of virtual machines	Virtual machine scale sets
You need to add VM instances for changing workload	Manual process to create, configure, and ensure compliance	Automatically create from central configuration
You need to balance and distribute workloads	Manual process to create and configure Azure load balancer or Application Gateway	Can automatically create and integrate with Azure load balancer or Application Gateway
You need high availability and redundancy	Manually create Availability Set or distribute and track VMs across Availability Zones	Automatic distribution of VM instances across Availability Zones or Availability Sets
You need to monitor and then scale virtual machines	Manual monitoring and Azure Automation	Autoscale based on host metrics, in-guest metrics, Application Insights, or schedule

Think back to Tailwind Traders application portfolio and where they could leverage virtual machine scale sets. Why would you recommend virtual machine scale sets over groups of VMs and for which specific applications?

Design a highly available container solution

Azure Kubernetes Service (AKS) is an Azure service that deploys a managed Kubernetes cluster. AKS is responsible for deploying the Kubernetes cluster and for managing the Kubernetes API server. It's important that any apps that you deploy that are based on AKS are designed to be highly available and reliable.

By default, AKS automatically provides a degree of high availability. This is achieved by using multiple nodes in a Virtual Machine Scale Set. However, these nodes can’t provide protection

against Azure region failure. Consequently, you should plan to implement your AKS clusters in multiple regions.

> **Tip**
>
> When you deploy multiple AKS clusters, choose regions where AKS is available. Always use paired regions.

Consider multiple region availability

When planning to implement AKS clusters across multiple region deployments, consider the following:

- **AKS region availability**. Choose regions that are close to your users. Keep in mind that AKS is continually expanding into new regions.
- **Azure paired regions.** For your geographic area, choose two regions paired together. Also consider that:
  - AKS platform updates (planned maintenance) are serialized with a delay of at least 24 hours between paired regions.
  - Recovery efforts for paired regions are prioritized where needed.
- **Service availability**. Decide whether your paired regions should be hot/hot, hot/warm, or hot/cold. In other words, do you want to run both regions at the same time, with one region ready to start serving traffic? Or do you want to give one region time to get ready to serve traffic?

Describe Azure Storage replication options

It's probable that your apps use Azure Storage for their data. Assuming they do, and that those apps are distributed across multiple AKS clusters in multiple regions, you'll need a way to synchronize storage. With Azure Storage, there are two possible options you can consider:

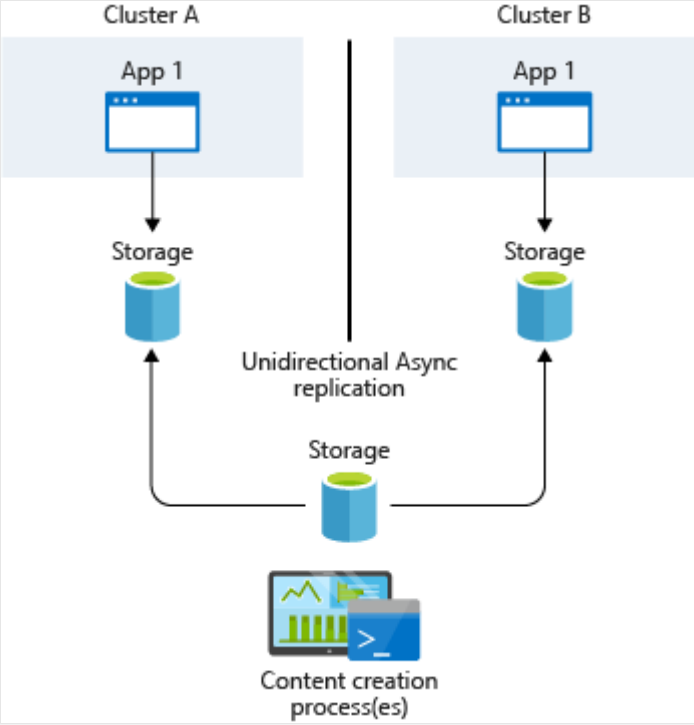
- Infrastructure-based asynchronous replication
- Application-based asynchronous replication

Infrastructure-based asynchronous replication

Your apps might require persistent storage. In Kubernetes, you can use persistent volumes to persist data storage. These persistent volumes are mounted to a node VM and then exposed to the pods.

> **Note**
>
> Persistent volumes follow pods even if the pods are moved to a different node inside the same cluster.

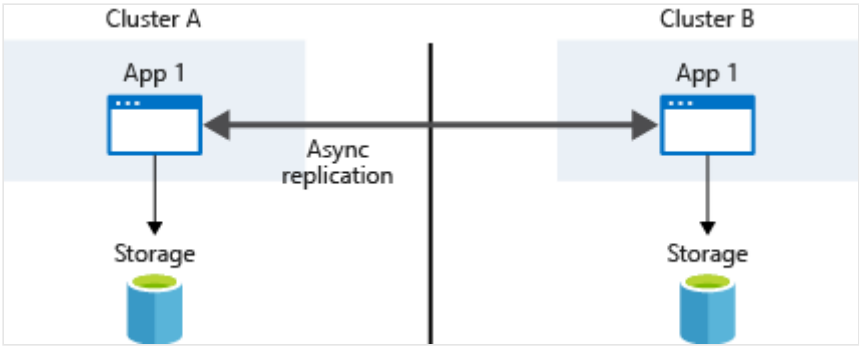
Typically, you provide a common storage point where apps write their data. This data is then replicated across regions and accessed locally, as displayed in the following graphic.



Application-based asynchronous replication

Kubernetes currently provides no native implementation for application-based asynchronous replication. However, because containers and Kubernetes are loosely coupled, you should be able to use any traditional app or language approach to replicate storage.

Typically, the apps themselves can replicate the storage requests These requests are then written to each cluster's underlying data storage. This process is displayed in the following graphic.



Consider Azure Backup or Velero

As with any app, it's important you back up the data related to your AKS clusters and their apps. When your apps consume and store data which is persisted on disks or in files, you should schedule frequent backups or take regular snapshots of that data. You can use several tools for these backup operations, including:

- Azure Disks: Azure Disks can use built-in snapshot technologies. However, your apps might need to flush writes-to-disk before the snapshot operation.
- Velero: Velero can back up persistent volumes along with additional cluster resources and configurations.

> **Note**
>
> Velero is an open-source tool: You can use Velero to back up and restore your data. You can also use it for disaster recovery and migration of Kubernetes cluster resources and persistent volumes.

Next unit: Recommend a high availability solution for relational data storage

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Recommend a high availability solution for relational data storage

3 minutes

Tailwind Traders has numerous apps that access and store data in databases. It's important that these apps, and their respective databases, are highly available and when necessary, recoverable.

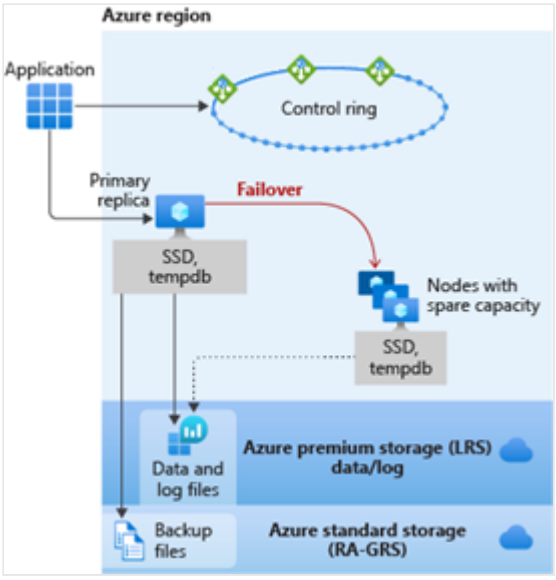
To understand the availability options and capabilities in Azure SQL, you need to understand service tiers. The service tier you select will determine the underlying architecture of the database or managed instance that you deploy.

There are two purchasing models to consider: DTU and vCore. Lets focus on the vCore service tiers and their architectures for high availability. You can equate the DTU model's Basic and Standard tiers to General Purpose, and its Premium tiers to Business Critical.

General purpose

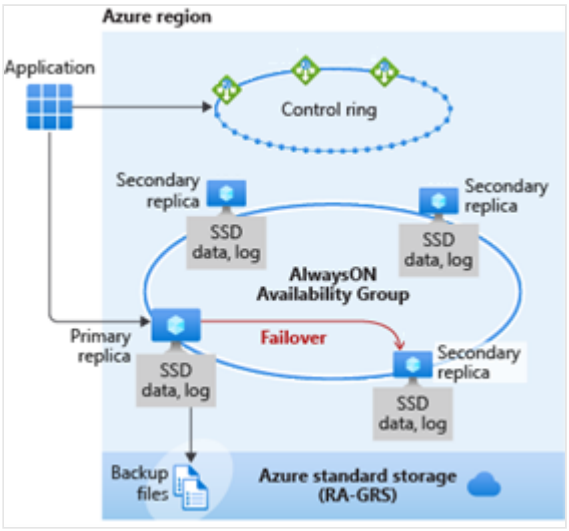
Databases and managed instances in the General Purpose service tier have the same availability architecture. Using the following figure as a guide, first consider the application and control ring. The application connects to the server name, which then connects to a gateway (GW) that points the application to the server to connect to, running on a VM. With General Purpose, the primary replica uses locally attached SSD for the tempdb. The data and log files are stored in Azure Premium Storage, which is locally redundant storage (multiple copies in one region). The backup files are then stored in Azure Standard Storage, which is RA-GRS by default. In other words, it's globally redundant storage (with copies in multiple regions).

All of Azure SQL is built on Azure Service Fabric, which serves as the Azure backbone. If Azure Service Fabric determines that a failover needs to occur, the failover will be similar to that of a failover cluster instance (FCI). The service fabric will identify a node with spare capacity and spin up a new SQL Server instance. The database files will then be attached, recovery will be run, and gateways will be updated to point applications to the new node. No virtual network or listener or updates are required. This capability is built in.



Business critical

The next service tier to consider is Business Critical, which can generally achieve the highest performance and availability of all Azure SQL service tiers (General Purpose, Hyperscale, Business Critical). Business Critical is meant for mission-critical applications that need low latency and minimal downtime.



Using Business Critical is like deploying an Always On availability group (AG) behind the scenes. Unlike in the General Purpose tier, in Business Critical, the data and log files are all running on direct-attached SSD, which significantly reduces network latency. (General Purpose uses remote storage.) In this AG, there are three secondary replicas. One of them can be used as a read-only endpoint (at no additional charge). A transaction can complete a commit when at least one of the secondary replicas has hardened the change for its transaction log.

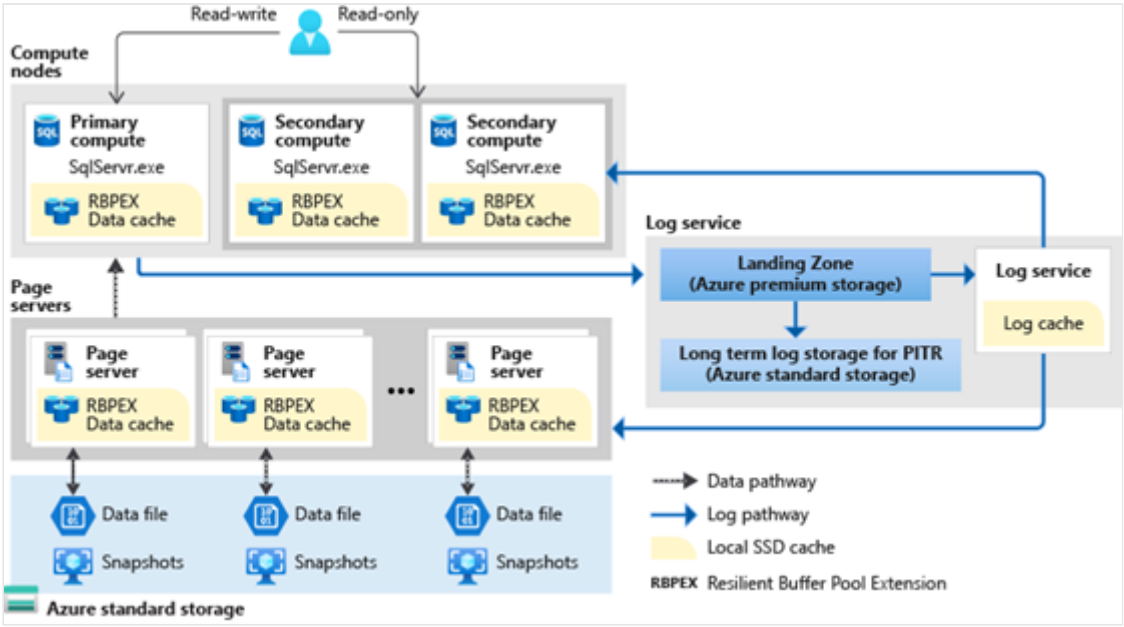
Read scale-out with one of the secondary replicas supports session-level consistency. So if the read-only session reconnects after a connection error caused by replica unavailability, it might be redirected to a replica that's not 100% up to date with the read-write replica. Likewise, if an

application writes data by using a read-write session and immediately reads it by using a read-only session, the latest updates might not immediately be visible on the replica. The latency is caused by an asynchronous transaction log redo operation.

If any type of failure occurs and the service fabric determines a failover needs to occur, failing over to a secondary replica is fast because the replica already exists and has the data attached to it. In a failover, you don't need a listener. The gateway will redirect your connection to the primary even after a failover. This switch happens quickly, and then the service fabric takes care of spinning up another secondary replica.

Hyperscale

The Hyperscale service tier is currently available for Azure SQL Database, and not Azure SQL Managed Instance. This service tier has a unique architecture because it uses a tiered layer of caches and page servers to expand the ability to quickly access database pages without having to access the data file directly.



Because this architecture uses paired page servers, you can scale horizontally to put all the data in caching layers. This new architecture also allows Hyperscale to support databases as large as 100 TB. Because it uses snapshots, nearly instantaneous database backups can occur regardless of size. Database restores take minutes rather than hours or days. You can also scale up or down in constant time to accommodate your workloads.

It's interesting to note how the log service was pulled out in this architecture. The log service is used to feed the replicas and the page servers. Transactions can commit when the log service hardens to the landing zone. So the consumption of the changes by a secondary compute replica isn't required for a commit. Unlike in other service tiers, you can determine

whether you want secondary replicas. You can configure zero to four secondary replicas, which can all be used for read-scale.

As in the other service tiers, an automatic failover will happen if service fabric determines it needs to. But the recovery time will depend on the existence of secondary replicas. For example, if you don't have replicas and a failover occurs, the scenario will be similar to that of the General Purpose service tier: the service fabric first needs to find spare capacity. If you have one or more replicas, recovery is faster and more closely aligns to that of the Business Critical service tier.

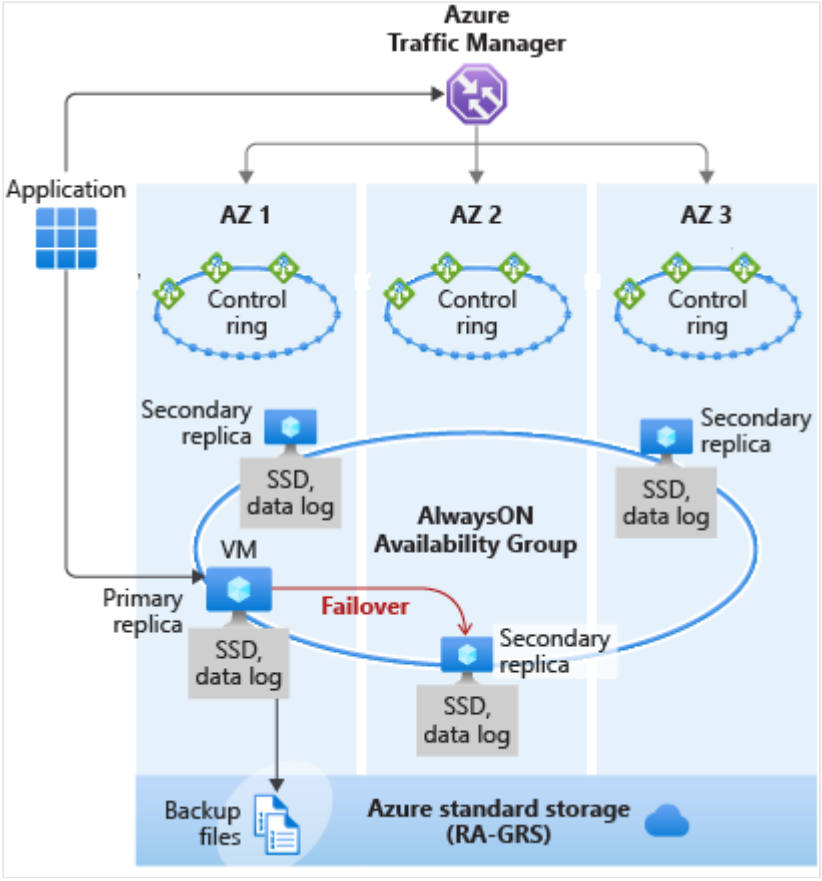
Business Critical maintains the highest performance and availability for workloads with small log writes that need low latency. But the Hyperscale service tier allows you to get a higher log throughput in terms of MB/second, provides for the largest database sizes, and provides up to four secondary replicas for higher levels of read scale. So, you'll need to consider your workload when you choose between the two.

Database service tiers for availability

Azure SQL Database and Azure SQL Managed Instance provide great availability options by default in the various service tiers. There are some additional things you can do to increase or modify the availability of your databases/instances. You'll be able to determine the impact on the service-level agreement (SLA). In this unit, you'll learn how you can go further with various options for availability in Azure SQL.

Availability Zones

In the Business Critical tier in Azure SQL Database, you can opt in (for no additional fee) for a zone-redundant configuration if your region supports that. At a high level, the Always On Availability Group (AG) that runs behind Business Critical databases and managed instances is deployed across three Availability Zones within a region. An Availability Zone is basically a separate datacenter within a given region. There's always a physical separation between Availability Zones. This capability protects against catastrophic failures that might occur to a datacenter in a region.



From performance standpoint, there might be a small increase in network latency because your AG is now spread across datacenters that have some distance between them. For this reason, Availability Zones isn't turned on by default. You can choose to use what's commonly called a "Multi-Az" or a "Single-Az" deployment. Configuring this option is as simple as adding a parameter to a PowerShell/Azure CLI command or selecting a check box in the portal.

Availability Zones are relatively new to Azure SQL, so they're currently available only in certain regions and service tiers. Over time, this capability is likely to be supported in more regions and potentially more service tiers.

Azure SQL SLA

Azure SQL maintains a service-level agreement (SLA) that provides financial backing to the commitment to achieve and maintain service levels. If your service level isn't achieved and maintained as described in the SLA, you might be eligible for a credit toward a portion of your monthly service fees.

Currently, you can achieve the highest availability (99.995%) from an Azure SQL Database Business Critical deployment that has Availability Zones configured. The Business Critical tier is the only option in the industry that provides RPO and RTO SLAs of 5 seconds and 30 seconds, respectively. RPO stands for recovery point object. It represents the amount of data you're potentially willing to lose in a worst case scenario. RTO stands for recovery time objective. It represents how long it takes to be back up and running again if a disaster occurs.

For General Purpose or single-zone Business Critical deployments of Azure SQL Database or Azure SQL Managed Instance, the SLA is 99.99%.

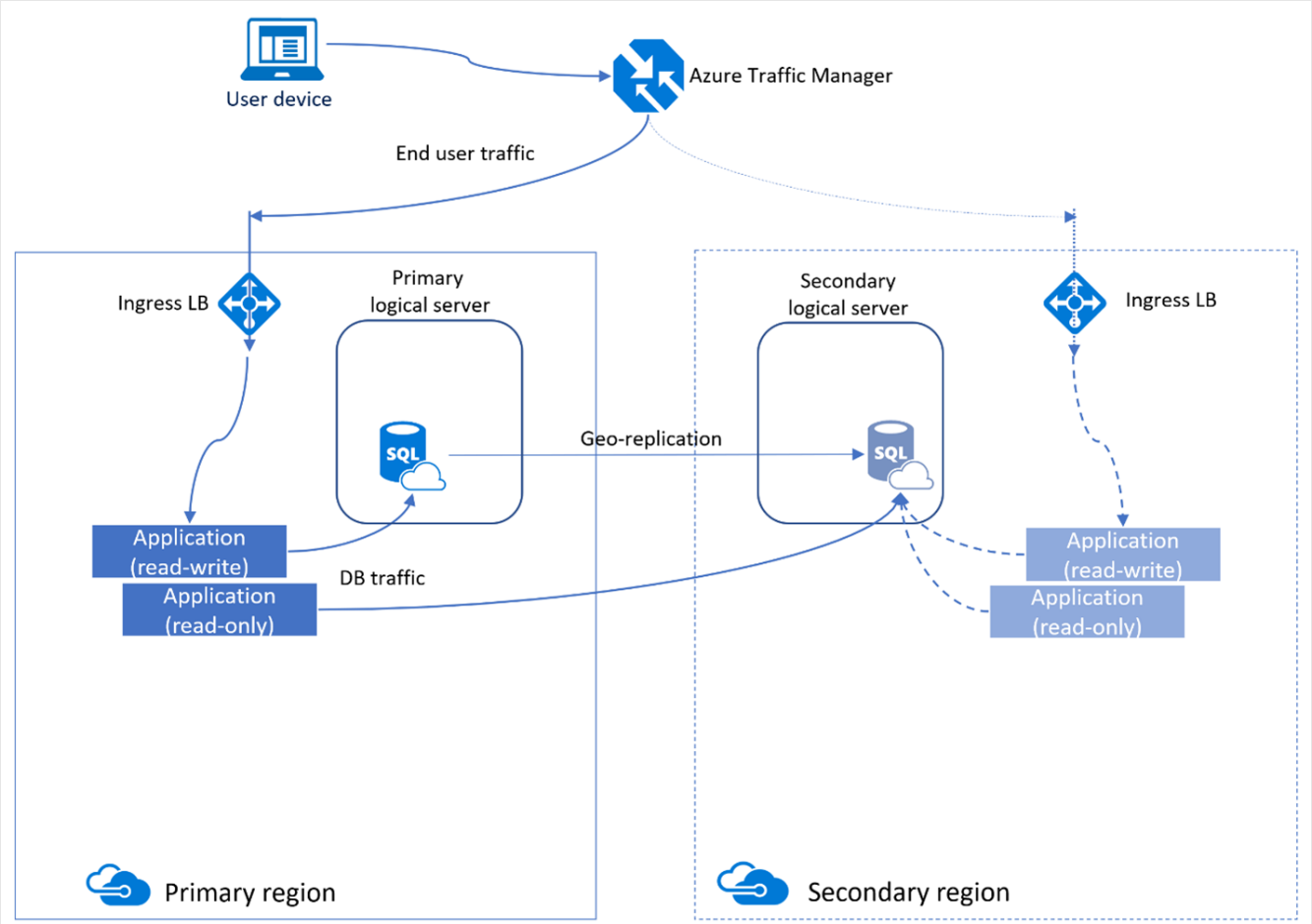
The Hyperscale tier's SLA depends on the number of replicas. Remember that you choose how many replicas you have in Hyperscale. If you don't have any, the failover behavior is more like that of General Purpose. If you have replicas, the failover behavior is more like that of Business Critical. Here are the SLAs, based on the number of replicas:

- 0 replicas: 99.5%
- 1 replica: 99.9%
- 2 or more replicas: 99.99%

What is active geo-replication?

Organizations, like Tailwind Traders, that want a local presence, or a hot backup, can run services in multiple Azure regions. Active geo-replication enables you to create readable secondary databases of individual databases. These databases can be on a server in the same or in a different region.

In the following graphic, an app is hosted in the custom domain Contoso.com. The app uses databases to store data. To manage failover to a secondary region, an administrator has configured a secondary region. Geo-replication has been enabled between the regions.



Active geo-replication is available for:

- Azure SQL Database: You can configure active geo-replication for any database in any elastic database pool. You can use active geo-replication to:
  - Create a readable secondary replica in a different region.
  - Fail over to a secondary database if your primary database fails or needs to be taken offline.

> **Tip**
>
> You can have up to four readable secondary replicas. Cosmos DB. Cosmos DB supports geo-replication across regions. However, you also can:
>
> - Designate one region as the writable region and all others as read-only replicas.
> - Fail over by selecting another region to be the write region during an outage.

> **Note**
>
> Support for Azure SQL Hyperscale is in preview.

The following table describes the capabilities of active geo-replication.

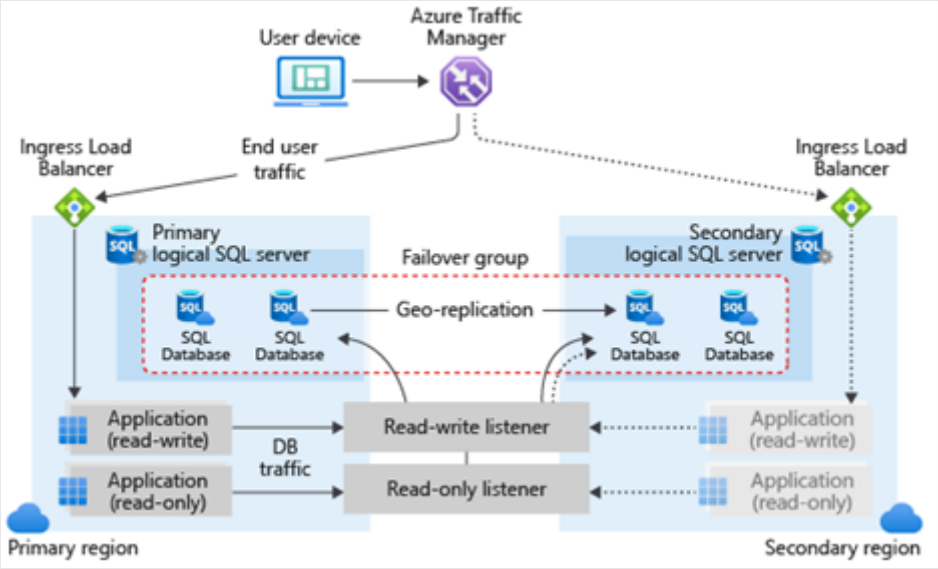
Capability	Description
Automatic Asynchronous Replication	You can only create a secondary database by adding to an existing database. The secondary database is seeded with data from the primary database. Thereafter, updates to the primary are asynchronously replicated to the secondary database automatically.
Readable secondary databases	An application can access a secondary database for read-only operations.
Planned failover	Use planned failover to switch the roles of primary and secondary databases. This is an online operation that doesn't result in data loss. Planned failover enables you to: conduct DR drills; relocate the database to a different region; return the database to the primary region.
Unplanned failover	Unplanned failover switches the secondary to the primary role immediately, and without any synchronization with the primary. Designed as a recovery method during outages when the primary isn't accessible, but database availability must be quickly restored. Note that transactions committed to the primary but not replicated to the secondary are lost.
Multiple readable secondaries	Configure up to four secondary databases for each primary.
Geo-replication of databases in an elastic pool	You can configure each secondary database in an elastic pool, or not, as required. Because each elastic pool is contained within a single region, multiple secondary databases in the same topology can never share an elastic pool.
User-controlled failover and failback	You can explicitly switch a secondary database to the primary role at any time.

What are auto-failover groups?

Because active geo-replication isn't supported by Azure SQL Managed Instance, for geographic failover of instances of SQL Managed Instance, you must use Auto-failover groups. Auto-failover groups enable you to manage the replication (and failover) of a group of databases (on a server or all databases in a managed instance) to another region. As with geo-replication, you can choose to initiate failover manually, or you can delegate it to the Azure service based on a policy that you define.

> **Tip**
>
> A failover group can include one or multiple databases, typically used by the same app.

You must configure the auto-failover group on the primary server. The configuration connects the primary to the secondary server in a different Azure region. The groups can include all or some databases in these servers. The following graphic depicts a typical configuration of a geo-redundant cloud application using multiple databases and auto-failover group.



The following table describes some of the capabilities of auto-failover groups.

Capability	Description
Automatic failover policy	A failover group is configured with an automatic failover policy by default. Azure triggers failover when a failure is detected, and the grace period has expired.
Planned failover	Performs full synchronization between primary and secondary databases. Then the secondary switches to the primary role.

Capability	Description
Unplanned failover	Immediately switches the secondary to the primary role without any synchronization with the primary.
Manual failover	Enables you to initiate failover manually, regardless of the automatic failover configuration. If you haven't configured an automatic failover policy, manual failover is required to recover databases in the failover group to the secondary.

Since auto-failover groups are a similar function to active geo-replication, it's worth considering when to use each.

Use active geo-replication when:

- You're implementing the Hyperscale service tier. Auto-failover groups aren't currently supported in the Hyperscale service tier.
- You want multiple Azure SQL Database secondaries in the same or different regions.

Use auto-failover groups when:

- You're implementing Azure SQL DB managed instance.

Geo-replication and auto-failover groups

After you choose a service tier (and consider Availability Zones as applicable), you can consider some other options for getting read-scale or the ability to fail over to another region: geo-replication and auto-failover groups. In SQL Server on-premises, configuring either of these options is something that would take a lot of planning, coordination, and time.

The cloud, and Azure SQL specifically, have made this process easier. For both geo-replication and auto-failover groups, you can get configured with a few clicks in the Azure portal or a few commands in the PowerShell/Azure CLI.

Here are some considerations to help you decide if geo-replication or auto-failover groups are best for your scenario:

	Geo-replication	Failover groups
Automatic failover	No	Yes

	Geo-replication	Failover groups
Fail over multiple databases simultaneously	No	Yes
User must update connection string after failover	Yes	No
SQL Managed Instance support	No	Yes
Can be in same region as primary	Yes	No
Multiple replicas	Yes	No
Supports read-scale	Yes	Yes

Global distribution enables you to replicate data from one region into multiple Azure regions. You can add or remove regions in which your database is replicated at any time, and Azure Cosmos DB ensures that when you add an additional region, your data is available for operations within 30 minutes, assuming your data is 100 TBs or less.

There are two common scenarios for replicating data in two or more regions:

1. Delivering low-latency data access to end users no matter where they are located around the globe
2. Adding regional resiliency for business continuity and disaster recovery (BCDR)

To deliver low-latency data access to customers, it is recommended that you replicate the data to regions closest to where your users are. For your online clothing company, you have customers in Los Angeles, New York, and Tokyo. Take a look at the Azure regions page, and determine the closest regions to those sets of customers, as those are the locations you'll replicate users to.

To provide a BCDR solution, it is recommended to add regions based on the region pairs described in the Business continuity and disaster recovery (BCDR): Azure Paired Regions article.

What high availability strategy would you recommend for Tailwind Traders relational databases? How would you design their production workloads compared to dev and test environments?

---

100 XP

Recommend a high availability solution for non-relational data storage

3 minutes

Azure storage redundancy

What is Azure storage redundancy?

Azure Storage provides several redundancy options that can help ensure your data is available. When choosing a redundancy option for storage, consider the following:

- How your data is replicated in the primary region that hosts your apps
- Whether your data is replicated to a second region that's geographically distant to help protect against regional disasters
- Whether your apps require read access to replicated data if the primary region is unavailable

Redundancy in the primary region can be provided as follows:

- Locally redundant storage (LRS). Helps protect your data against drive or server rack failures in a data center. But if a disaster occurs within the data center, all replicas of your storage account that uses LRS might be lost. This option:
  - Copies your data synchronously three times within a single physical location in the primary region.
  - Is the least expensive replication option.
  - Isn’t recommended for apps that require high availability or durability.
- Zone-redundant storage (ZRS). Helps ensure that your data is still accessible for both read and write operations even if a zone becomes unavailable. This option:
  - Copies your data synchronously across three Azure availability zones in the primary region.

- Recommended by Microsoft for apps requiring high availability in the primary region and replicating to a secondary region.

> **Note**
>
> Each availability zone is a separate physical location with independent power, cooling, and networking.

LRS is the lowest-cost redundancy option and offers the least durability compared to other options. LRS protects your data against server rack and drive failures. However, if a disaster such as fire or flooding occurs within the data center, all replicas of a storage account using LRS may be lost or unrecoverable. To mitigate this risk, Microsoft recommends using zone-redundant storage (ZRS), geo-redundant storage (GRS), or geo-zone-redundant storage (GZRS).

LRS is a good choice for the following scenarios:

1. If your application stores data that can be easily reconstructed if data loss occurs, you may opt for LRS.
2. If your application is restricted to replicating data only within a country or region due to data governance requirements, you may opt for LRS. In some cases, the paired regions across which the data is geo-replicated may be in another country or region. For more information on paired regions, visit Azure regions .

For your apps that require high durability, you can create copies of your data in a secondary region. Redundancy in the secondary region can be provided as follows:

- Geo-redundant storage (GRS):
  - Copies your data synchronously three times within a single physical location in the primary region using LRS.
  - Copies your data asynchronously to a single physical location in the secondary region.
  - Copies your data synchronously three times within the secondary region using LRS.
- Geo-zone-redundant storage (GZRS):
  - Copies your data synchronously across three Azure availability zones in the primary region using ZRS.
  - Copies your data asynchronously to a single physical location in the secondary region.
  - Copies your data synchronously three times using LRS within the secondary region.

> **Important**
>
> The primary difference between GRS and GZRS is how data is replicated in the primary region. Within the secondary region, data is always replicated synchronously three times using LRS. LRS in the secondary region protects your data against hardware failures.

With both GRS and GZRS, your data in the secondary region isn't available for read/write access unless there's a failover to the secondary region. To enable read access to the secondary region, configure your storage account to use one of the following:

- Read-access geo-redundant storage (RA-GRS).
- Read-access geo-zone-redundant storage (RA-GZRS)

Summary of storage redundancy options

The following table summarizes the redundancy options discussed and indicates whether your data is durable and available in each scenario.

Outage scenario	LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
A node within a data center becomes unavailable	Yes	Yes	Yes	Yes
An entire data center (zonal or non-zonal) becomes unavailable	No	Yes	Yes*	Yes
A region-wide outage occurs in the primary region	No	No	Yes*	Yes*
Read access to the secondary region is available if the primary region becomes unavailable	No	No	Yes (with RA-GRS)	Yes (with RA-GZRS)

> **Important**
>
> Account failover is required to restore write availability if the primary region becomes unavailable.

When deciding which redundancy option is best for your scenario, consider the tradeoffs between lower costs and higher availability. The factors that help determine which redundancy option you should choose include:

- How your data is replicated in the primary region
- Whether your data is replicated to a second region that is geographically distant to the primary region, to protect against regional disasters
- Whether your application requires read access to the replicated data in the secondary region if the primary region becomes unavailable for any reason

Data Lake Storage redundancy

Azure Storage is scalable by design whether you access via Data Lake Storage Gen2 or Blob storage interfaces. It is able to store and serve many exabytes of data. This amount of storage is available with throughput measured in gigabits per second (Gbps) at high levels of input/output operations per second (IOPS). Processing is executed at near-constant per-request latencies that are measured at the service, account, and file levels.

Because Data Lake Storage Gen2 is built on top of Azure Blob Storage, multiple concepts can describe the same, shared things.

The following are the equivalent entities, as described by different concepts. Unless specified otherwise these entities are directly synonymous:

Concept	Top Level Organization	Lower Level Organization	Data Container
Blobs – General purpose object storage	Container	Virtual directory (SDK only – does not provide atomic manipulation)	Blob
Azure Data Lake Storage Gen2 – Analytics Storage	Container	Directory	File

Next unit: Knowledge check

Continue >

✓ 200 XP

Knowledge check

3 minutes

Tailwind Traders have several workloads being migrated to Azure. It is important you design all production workloads to be highly available based on the following requirements:

- **HR application**. The HR system requires an SLA of 99.99% and needs to be migrated to Azure.
- **Database strategy**. Where possible the company wants to be able to have database replicas either in the same region or a secondary region and be able to have multiple replicas that are asynchronously kept up to date
- **High traffic web applications**. During peak customer traffic, all global web applications need to be able to automatically fail over to a secondary region if the primary region fails. These web applications require SSL offload.

Choose the best response for each of the questions below. Then select **Check your answers**.

1. Which of the following solutions should be used for the HR application SLA requirement?

- ○ Availability Zones

  ✓ That's correct. Availability Zones offer a VM SLA of 99.99%

- ◉ Availability Sets

  ✗ that's incorrect. Availability sets provide 99.95% SLA

- ○ Single Virtual Machine with no availability Zone or Set

2. Which of the following should be used for the database strategy requirement?

- ○ Active geo-replication

  ✓ That's correct. Active geo-replication creates a replica of the database in another region that is asynchronously kept up to date.

- ○ Failover Group
- ◉ Secondary replica that is readable

  ✗ That's incorrect. Active geo-replication creates a replica of the database in another region that is asynchronously kept up to date.

3. Which of the following should be used for their high traffic web applications?

- ○ Azure Front Door

  ✔ that's correct. Azure front door offers SSL offload capabilities and meets the requirements for the global web applications

- ○ Traffic Manager
- ◉ Azure Load balancer

  ✘ that's incorrect. Azure load balancer doesn't meet the SSL offload requirement

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

Introduction

3 minutes

Meet Tailwind Traders

You work for a Tailwind Traders, a home improvement retailer. Tailwind Traders currently manages on-premises data centers that host the company's retail website. These data centers also store all the data and streaming video for its applications.

The IT department is currently responsible for all the management tasks for its computing hardware and software. The IT team handles the procurement process to buy new hardware, installs and configures software, and deploys everything throughout the data center.

These management responsibilities create some obstacles for delivering applications to Tailwind's users and customers in a timely fashion. So, the company is shifting on-premises workloads to the cloud.

You've been tasked with selecting appropriate backup solutions for migrated workloads. You also need to select an appropriate disaster recovery option for these workloads.

Learning objectives

After completing this module, you'll be able to:

- Design for backup and recovery.
- Design for Azure Backup.
- Design for Azure blob backup and recovery.
- Design for Azure Files backup and recovery.
- Design for Azure virtual machine backup and recovery.
- Design for Azure SQL backup and recovery
- Design for Azure Site Recovery.

Skills measured

The content in the module will help you prepare for Exam AZ-305: Designing Microsoft Azure Infrastructure Solutions. The module concepts are covered in:

Design business continuity

- Design a Solution for Backup and Disaster Recovery

Prerequisites

- Conceptual knowledge of Business Continuity and Disaster Recovery solutions.
- Working experience with object replication, backup solution tools, and recovery options.

Next unit: Design for backup and recovery

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for backup and recovery

3 minutes

Organizations, such as Tailwind Traders, require a high degree of reliability from their mission-critical apps. To achieve the desired reliability for on-premises based apps, it's typical to purchase more computing resource, such as servers and storage. By purchasing more computing resources, organizations can build redundancy into their on-premises infrastructure.

It's also vital that any mission-critical app, and its associated data, is recoverable following a failure-ideally to the point of failure. This recoverability is often provided by backup, restore components, and procedures. For organizations with apps hosted in Azure, or organizations with hybrid app deployments, there are other considerations and options.

Reliable apps are those which are:

- Resilient to component failure.
- Highly available and can run in a healthy state with no significant downtime.

To achieve the desired resilience and high availability, you must first define your requirements.

> **Note**
>
> This module will use the term resiliency as the ability of a system to gracefully handle and recover from failures, both inadvertent and malicious.

Define your requirements

Defining your requirements involves:

- Identifying your business needs.
- Building your resiliency plan to address those needs.

Use the following table of considerations to provide guidance on this process.

Consideration	Description
What are your workloads and their usage?	A workload is a distinct capability or task that is logically separated from other tasks, in terms of business logic and data storage requirements. Each workload probably has different requirements for availability, scalability, data consistency, and disaster recovery.
What are the usage patterns for your workloads?	Usage patterns can determine your requirements. Identify differences in requirements during both critical and non-critical periods. To ensure uptime, plan redundancy across several regions in case one region fails. Conversely, to minimize costs during non-critical periods, you can run your application in a single region.
What are the availability metrics?	Mean time to recovery (MTTR) and mean time between failures (MTBF) are the typically used metrics. MTBF is how long a component can reasonably expect to last between outages. MTTR is the average time it takes to restore a component after a failure. Use these metrics to determine where you need to add redundancy, and to determine service-level agreements (SLAs) for customers.
What are the recovery metrics?	The recovery time objective (RTO) is the maximum acceptable time one of your apps can be unavailable following an incident. The recovery point objective (RPO) is the maximum duration of data loss that is acceptable during a disaster. Also consider the recovery level objective (RLO). This metric determines the granularity of recovery. In other words, whether you must be able to recover a server farm, a web app, a site, or just a specific item. To determine these values, conduct a risk assessment. Ensure that you understand the cost and risk of downtime or data loss in your organization.
What are the workload availability targets?	To help ensure that your app architecture meets your business requirements, define target SLAs for each workload. Account for the cost and complexity of meeting availability requirements, in addition to application dependencies.
What are your SLAs?	In Azure, the SLA describes the Microsoft commitments for uptime and connectivity. If the SLA for a particular service is 99.9 percent, you should expect the service to be available 99.9 percent of the time.

Tip

If the MTTR of any critical component in a highly available scenario exceeds the system RTO, then a failure in the system might cause an unacceptable business disruption. In other words, you can't restore the system within the defined RTO.

Define your own target SLAs for each workload in your solution by answering the preceding questions. This helps ensure that the architecture meets your business requirements. For example, if a workload requires 99.99 percent uptime, but depends on a service with a 99.9 percent SLA, that service can't be a single point of failure in the system.

After you've defined your recovery requirements, you're ready to select a suitable recovery technology.

Next unit: Design for Azure Backup

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

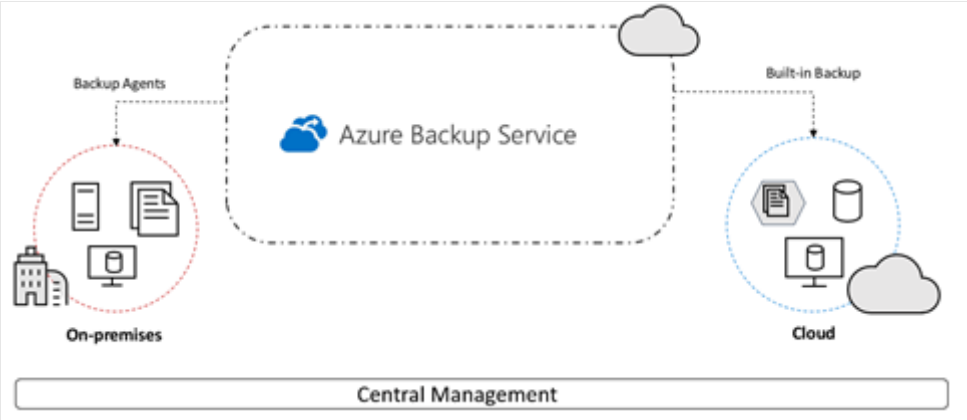
100 XP

Design for Azure Backup

3 minutes

The Azure Backup service uses Azure resources for short-term and long-term storage. Azure Backup minimizes or even eliminates the need for maintaining physical backup media. Examples of backup media are tapes, hard drives, and DVDs.

What can you do with Azure Backup?



You can use Azure Backup for these backup types:

- **On-premises.** Azure Backup can back up files, folders, and system state using the Microsoft Azure Recovery Services (MARS) agent. Alternatively, you can use Data Protection Manager (DPM) or the Microsoft Azure Backup Server (MABS) agent to protect on-premises VMs (both Hyper-V and VMware), and other on-premises workloads.
- **Azure VM.** Back up entire Windows or Linux VMs (using backup extensions), or back up files, folders, and system state using the MARS agent.
- **Azure Files shares**. Back up Azure File shares to a storage account.
- **Microsoft SQL Server in Azure VMs**. Back up SQL Server databases running on Azure VMs.
- **SAP HANA databases in Azure VMs**. Back up SAP HANA databases running on Azure VMs.

- **Microsoft cloud**. Azure Backup can replace your existing on-premises or off-site backup solution with a cloud-based solution that's reliable, secure, and cost-competitive.

Azure Backup offers multiple components that you can download and deploy on the appropriate computer, server, or in the cloud. The component (or agent) that you deploy depends on what you want to protect.

Where is the data backed up?

Azure Backup organizes your backup data in a storage entity called a vault. A storage vault stores backup copies, recovery points, and backup policies.. There are two types of vaults. The primary differences in the vaults are supported data sources and supported Azure products.

Capability	Supported data sources	Supported products
Backup vault	Azure database for PostgreSQL servers Azure blobs Azure disks	Azure Backup
Recovery services vault	Azure virtual machines (VMs) SQL in an Azure VM Azure Files SAP HANA in Azure VM Azure Backup Server Azure Backup Agent Data Protection Manager	Azure Backup, Azure Site Recovery

Considerations for storage vaults

- **Think about how to organize the vaults**. If your workloads are all managed by a single subscription and single resource, then you can use a single vault. If your workloads are spread across subscriptions, then you can create multiple vaults. Use separate vaults for Azure Backup and Azure Site Recovery.

- **Use Azure policy**. If you needed consistent policy across vaults, then you can use Azure Policy to propagate backup policy across multiple vaults. A backup policy is scoped to a vault.

- **Protect using Azure role-based access control (RBAC).** Protect and manage vault access by using Azure RBAC.

- **Design for Redundancy.** Specify how data in the vault is replicated for redundancy. Use Locally redundant storage (LRS) to protect against failure in a datacenter. LRS replicates data to a storage scale unit. Use Geo-redundant storage (GRS) to protect against region-wide outages. GRS replicates your data to a secondary region..

Next unit: Design for Azure blob backup and recovery

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

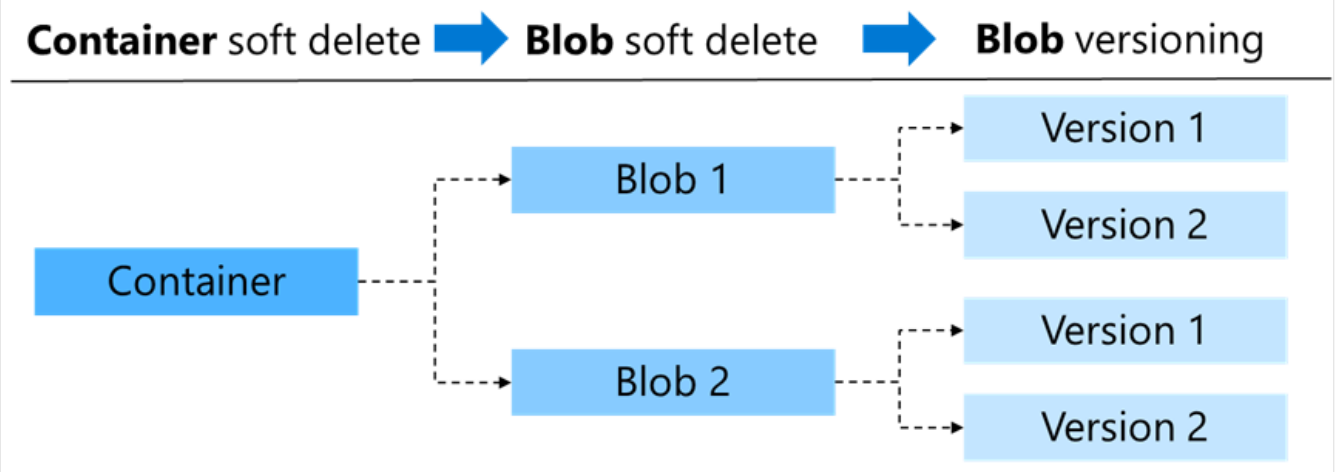
Design for Azure blob backup and recovery

3 minutes

Operational backup for blobs is a local backup solution. The backup data isn't transferred to the Backup vault but is stored in the source storage account itself. This is a **continuous backup** solution. You don't need to schedule any backups. All changes will be retained and restorable from a selected point in time.

Take advantage of blob soft delete and versioning

Soft delete protects an individual blob, snapshot, container, or version from accidental deletes or overwrites. Soft delete maintains the deleted data in the system for a specified retention period. During the retention period, you can restore a soft-deleted object to its state at the time it was deleted.



- **Container** soft delete can restore a container and its contents at the time of deletion. The retention period for deleted containers is between 1 and 365 days. The default retention period is seven days.
- **Blob** soft delete can restore a blob, snapshot, or version that has been deleted. Blob soft delete is useful for restoring specific files. The retention period for deleted blobs is also between 1 and 365 days.
- Blob versioning works to automatically maintain previous versions of a blob. When blob versioning is enabled, you can restore an earlier version of a blob. Versioning lets you recover your data if it's incorrectly modified or deleted. Blob versioning is useful if you

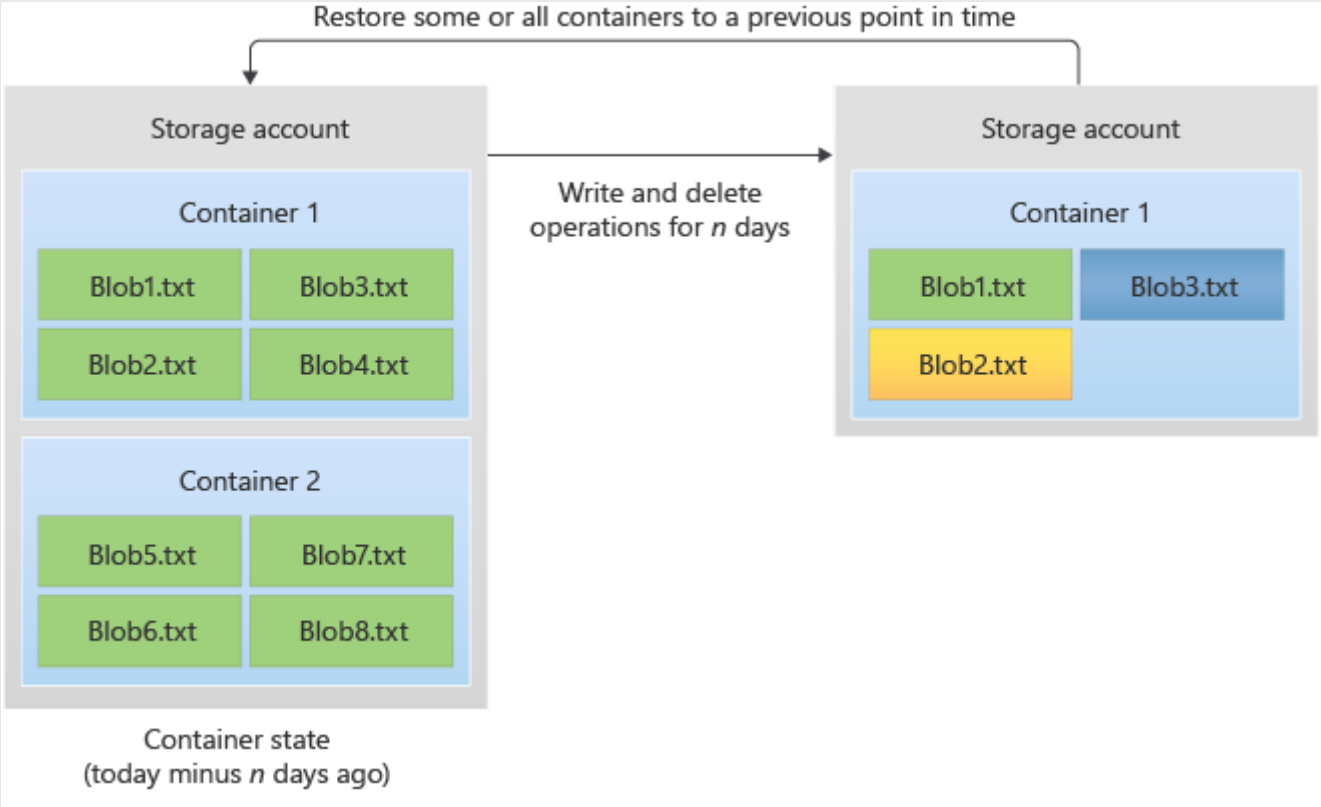
have multiple authors editing files and need to maintain or restore their individual changes.

> **Tip**
>
> Container soft delete doesn't protect against the deletion of a storage account, but only against the deletion of containers in that account.

Consider point-in-time restore for block blobs

Like soft delete, point-in-time restore for block blobs also protects against accidental deletion or corruption. For this feature, you create a management policy for the storage account and specify a retention period. During the retention period, you can restore block blobs from the present state to a state at a previous time.

The following diagram shows how point-in-time restore works. One or more containers or blob ranges is restored to its previous state. The effect is to revert write and delete operations that happened during the retention period.



> **Note**
>
> Point-in-time restore enables testing scenarios that require reverting a data set to a known state before running further tests.

Prevent accidental changes by using resource locks

A resource lock prevents resources from being accidentally deleted or changed. Consider using resources locks to protect your data. You can set the lock level to **CanNotDelete** or **ReadOnly**.

- **CanNotDelete** means authorized people can still read and modify a resource, but they can't delete the resource without first removing the lock.
- **ReadOnly** means authorized people can read a resource, but they can't delete or change the resource. Applying this lock is like restricting all authorized users to the permissions granted by the **Reader** role in Azure RBAC.

> **Tip**
>
> Consider the storage protection features you've learned about. Which ones do you think would be most useful? In what scenarios will you use these features?

Next unit: Design for Azure files backup and recovery

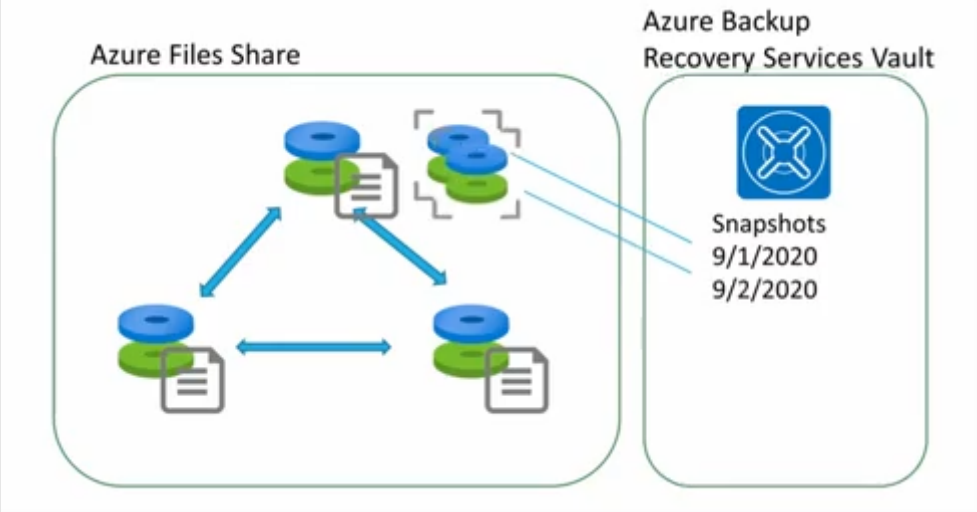
Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure files backup and recovery

3 minutes



Azure Files provides the capability to take share snapshots of file shares. Share snapshots give you an extra level of security and help reduce the risk of data corruption or accidental deletion. You can also use them as a general backup for disaster recovery.

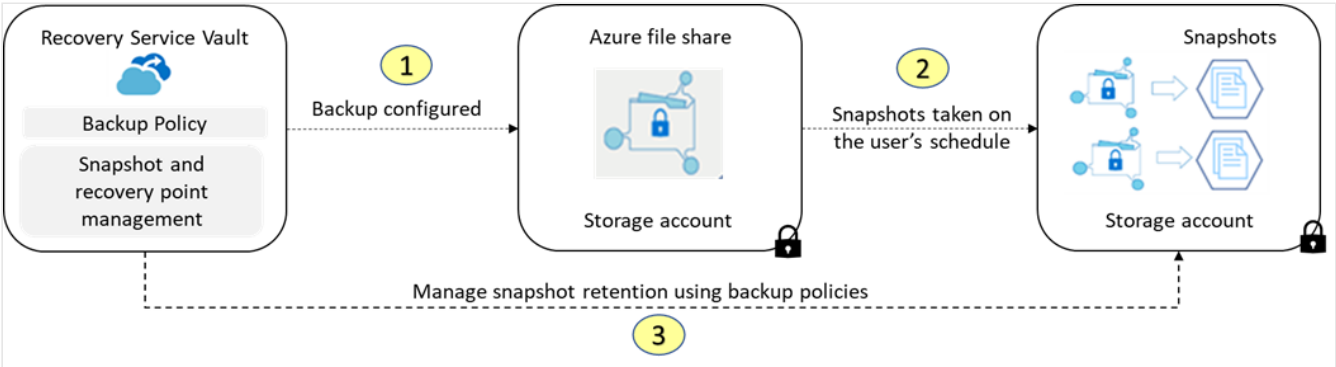
- Share snapshots capture the share state at that point in time.
- Snapshots can be created manually using the Azure portal, REST API, client libraries, the Azure CLI, and PowerShell.
- Snapshots can be automated using Azure Backup and backup policies.
- Snapshots are at the root level of a file share and apply to all the folders and files contained in it. Retrieval is provided at individual file level.
- Snapshots are incremental. Only the deltas between your snapshots will be stored.
- After a share snapshot is created, it can be read, copied, or deleted, but not modified.
- You cannot delete a share that has share snapshots. To delete the share you must delete all the share snapshots.

Important

Snapshots are not a replacement for cloud-side backups.

How can you automate file share backups?

It is recommended you use Azure Backup to automate and manage file share snapshots.



- Azure Backup keeps the metadata about the backup in the recovery services vault, but no data is transferred. This means a fast backup solution with built-in backup and reporting.
- When Azure Backup is enabled on the file share soft delete is also enabled.
- You can configure backups with daily/weekly/monthly/yearly retention according to your requirements.

Considerations for file share backups

- **Use instant restore.** Azure file share backup uses file share snapshots. You can select just the files you want to restore instantly.
- **Implement alerting and reporting**. You can configure alerts for backup and restore failures and use the reporting solution provided by Azure Backup. These reports provide insights on file share backups.
- **Consider self-service restore.** Backup uses server endpoint VSS snapshots. Consider giving advanced users the ability to restore files themselves.
- **Consider on-demand backups**. Azure Backup policies are limited to scheduling a backup once a day. If a user creates a file in the morning and works on it all day a nightly backup won't have the new file. For these reasons consider on-demand backups for the most critical file shares.
- **Organize file shares for backup**. If possible, consider organizing your file shares for backups. For example, public facing vs internal file shares.
- **Snapshot before code deployments**. If a bug or application error is introduced with the new deployment, you can go back to a previous version of your data on that file share.

To help protect against these scenarios, you can take a share snapshot before you deploy new application code.

Next unit: Design for Azure virtual machine backup and recovery

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

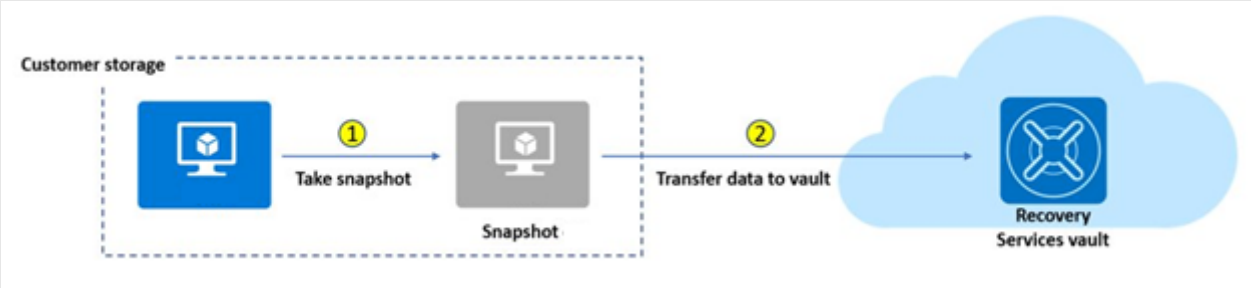
Design for Azure virtual machine backup and recovery

3 minutes

Azure Backup provides independent and isolated Azure virtual machines backups. Backups are stored in a Recovery Services vault with built-in management of recovery points. Configuration and scaling are simple, backups are optimized, and you can easily restore as needed. Back up is available for Windows and Linux VMs.

How do Azure virtual machines backups work?

An Azure backup job consists of two phases. First, a virtual machine snapshot is taken. Second, the virtual machine snapshot is transferred to the Azure Recovery Services vault.



> **Note**
>
> A recovery point is created only after both steps are completed. The recovery point is used to perform a restore.

Backup policies and retention

You can define the backup frequency and retention duration for your backups. Currently, the VM backup can be triggered daily or weekly, and can be stored for multiple years.

- **Snapshot tier**: In phase 1, snapshots are stored locally for a maximum period of five days. This is referred to as the snapshot tier. Snapshot tier restores are faster (than restore from vault) because they eliminate the wait time for snapshots to copy to the vault before triggering the restore. This capability is called **Instant Restore**.

- **Vault tier**: In phase 2, snapshots are transferred to the vault for additional security and longer retention. This is referred to as **vault tier**.

Considerations for Azure virtual machine backup and recovery

- **Plan backup schedule policies**. Consider grouping VMs that require the same schedule start time, frequency, and retention settings within a single policy. Ensure the backup scheduled start time is during non-peak production application time. To distribute backup traffic, consider backing up different VMs at different times of the day and make sure the times don't overlap. For example, have policies for critical and non-critical virtual machines.

- **Plan backup retention policies.** Implement both short-term (daily) and long-term (weekly) backups. If you need to take a backup not scheduled via backup policy, then you can use an on-demand backup. For example, backup on-demand multiple times per day when scheduled backup permits only one backup per day.

- **Optimize backup policies**. As your business requirements change, make sure to review and change your backup policies. Enable monitoring and alerting features and review the results.

- **Consider Cross Region Restore (CRR).** CRR allows you to restore Azure VMs in a secondary region, which is an Azure paired region. This option lets you to conduct drills to meet audit or compliance requirements You can also restore the VM or its disk if there's a disaster in the primary region. CRR is an opt-in feature for any recovery services vault. CRR also works for SQL databases and SAP HANA databases hosted on Azure VMs.

Next unit: Design for Azure SQL backup and recovery

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure SQL backup and recovery

3 minutes

It's essential that you can recover your SQL database data. You should consider automated backups of your Azure SQL Database and Azure SQL Managed Instances. Database backups enable database restoration to a specified point in time and within a configured retention period.

Describe automated backups

Both SQL Database and SQL Managed Instance use SQL Server technology to create full backups every week, differential backups every 12-24 hours, and transaction log backups every 5 to 10 minutes. The frequency of transaction log backups is based on the compute size and the amount of database activity. When you restore a database, the service determines which full, differential, and transaction log backups need to be restored.

- **Full backups**: In a full backup, everything in the database and the transaction logs is backed up. SQL Database makes a full back up once a week.
- **Differential backups**: In a differential backup, everything that changed since the last full backup is backed up. SQL Database makes a differential backup every 12 - 24 hours.
- **Transactional backups**: In a transactional backup, the contents of the transaction logs are backed up. If the latest transaction log has failed or is corrupted, the option is to fall back to the previous transaction log backup. Transactional backups enable administrators to restore up to a specific time, which includes the moment before data was mistakenly deleted. Transaction log backups every five to 10 minutes.

Describe backup usage cases

You can use the automated backups in several ways.

- Restore an existing database to a point in time in the past within the retention period. This operation creates a new database on the same server as the original database but uses a different name to avoid overwriting the original database. After the restore completes, you can delete the original database.

- Restore a deleted database to the time of deletion or to any point in time within the retention period. The deleted database can be restored only on the same server or managed instance where the original database was created.
- Restore a database to another geographic region. Geo-restore allows you to recover from a geographic disaster when you cannot access your database or backups in the primary region. It creates a new database on any existing server or managed instance, in any Azure region.
- Restore a database from a specific long-term backup of a single database or pooled database. If the database has been configured with a long-term retention policy you can restore an old version of the database.

Long-term backup retention policies

Azure SQL Database automatic backups remain available to restore for up to 35 days. This period is enough for the purposes of day-to-day administration. But sometimes you might need to retain data for longer periods. For example, data protection regulations in your local jurisdiction might require you to keep backups for several years.

For these requirements, use the long-term retention (LTR) feature. This way, you can store Azure SQL Database backups in read-access geo-redundant storage (RA-GRS) blobs for up to 10 years. If you need access to any backup in LTR, you can restore it as a new database by using either the Azure portal or PowerShell.

Next unit: Design for Azure Site Recovery

Continue >

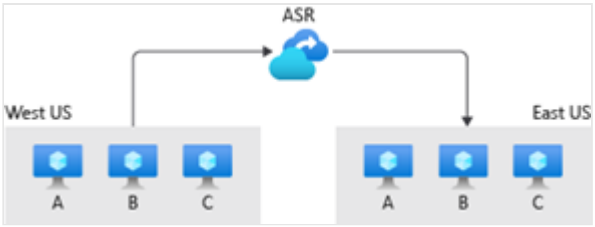
How are we doing? ☆ ☆ ☆ ☆ ☆

100 XP

Design for Azure Site Recovery

3 minutes

Azure Site Recovery is a service that provides BCDR features for your applications in Azure, on-premises, and in other cloud providers. Azure Site Recovery has plans that help automate your disaster recovery by enabling you to define how machines are failed over, and the order in which they're restarted after being successfully failed over. In this way, Azure Site Recovery helps to automate tasks and further reduce your recovery time objective. You also use Azure Site Recovery to periodically test failovers, and the overall effectiveness of the recovery process.



What can you do with Azure Site Recovery?

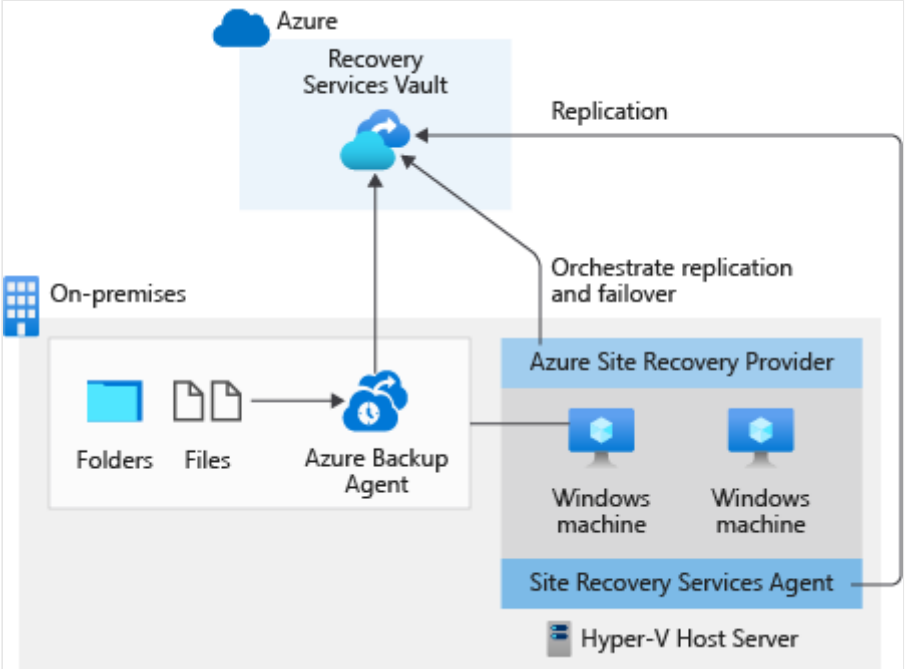
Site Recovery provides the capabilities described in the following table.

Feature	Details
Simple BCDR solution	Use Site Recovery in the Azure portal to setup and manage replication, failover, and failback from a single location.
Azure VM replication	Setup disaster recovery of your Azure VMs, and failover from a primary region to a secondary region.
On-premises VM replication	Replicate on-premises VMs and physical servers to Azure, or to a secondary on-premises datacenter.

Feature	Details
Workload replication	Replicate any workload running on supported Azure VMs, on-premises Hyper-V and VMware VMs, and Windows or Linux physical servers.
Data resilience	Orchestrate replication without intercepting app data by using Site Recovery. When failover occurs, Azure VMs are created, based on the replicated data. When you replicate to Azure, data is stored in Azure storage, with the resilience that provides.
RTO and RPO targets	Keep RTO and RPO within defined organizational limits. Site Recovery provides continuous replication for Azure VMs and VMware VMs, and replication frequency as low as 30 seconds for Hyper-V.
Keep apps consistent over failover	By using app-consistent snapshots, you can replicate using recovery points. These snapshots capture disk data, data in memory, and all in process transactions.
Testing without disruption	You can run disaster recovery tests, without affecting ongoing replication.
Flexible failovers	You can run planned failovers for expected outages with no data loss. Run unplanned failovers with minimal data loss. And fail back to your primary site when it's available again.
Customized recovery plans	Create recovery plans so that you can customize and sequence the failover and recovery of your multi-tier apps running on multiple VMs. You can group machines together in a recovery plan. You can then, optionally, add scripts and manual actions. You can integrate recovery plans with Azure automation runbooks.
BCDR integration	You can integrate Site Recovery with other BCDR technologies. For example, use Site Recovery to protect the SQL Server backend of your corporate workloads. Because of its native support for SQL Server AlwaysOn, you can manage the failover of availability groups.
Azure automation integration	Download, from the Azure Automation library, and integrate app-specific scripts with Site Recovery.

Consider using Azure Site Recover with Azure Backup

Here we have an on-premises environment that has a Hyper-V host server for hosting virtual machines. You want to back up all the files and folders in this virtual machine to Azure. You also want to protect any workloads running on the virtual machine and keep running them even if the virtual machine fails. Azure Backup and Site Recovery can be used together as part of a single solution.



In this scenario, Azure Backup periodically backs up the files and folders on the Windows machine to Azure. This process ensures they are secure and retrievable even if the whole on-premises environment stops functioning. Separately, Site Recovery will be used to protect running workloads and keep them running. Because Site Recovery can replicate frequently, the RTO for your workloads can be reduced.

Next unit: Knowledge check

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆

3 minutes

Choose the best response for each of the questions below. Then select **Check your answers**.

Tailwind Traders is assessing their backup continuity and disaster recovery plan. They have asked you assist with selecting an appropriate backup solution for migrated workloads. You must select an appropriate disaster recovery option for these workloads. Here are some specific requirements.

- **Regional outage option for Azure virtual machine backups.** The company has stringent recovery time objectives and recovery point objectives. The IT department is concerned about regional outages and would like a solution for that situation.
- **Backup option for on-premises virtual machines.** The IT department would like a way to back up on-premises virtual machines. They would like to define the frequency and only pay-as-they-go for storage.
- **Database backup option for Azure SQL database.** The company has several Azure SQL databases in an elastic pool. The IT department needs a solution for when the primary database fails or needs to be taken offline.
- **Accidental deletion and recovery of video files.** Each product in the catalog has an associated video. This video is critical to the marketing of the product on the website. The IT department wants to ensure they can recover snapshots of these files, if they are deleted.

1. What replication option would be best for the Azure virtual machine backups?

- ○ Azure Site Recovery

  ✔ That's correct. Azure Site Recovery is designed to provide continuous replication to a secondary region.

- ◉ Azure Backup

  ✘ That's incorrect. Azure Backup is designed to provide scheduled backups to a storage vault.

- ○ Active geo-replication

2. What backup solution is best for the on-premises virtual machines?

- ○ Azure Site Recovery
- ○ Azure Backup

  ✔ That's correct. Azure Backup can protect on-premises virtual machines.

- ◉ Active geo-replication

  ✗ That's incorrect. Active geo-replication refers to database solutions.

3. What solution would be best for the Azure SQL database requirement?

- ○ Azure Site Recovery
- ◉ Azure Backup

  ✗ That's incorrect. Azure Backup isn't used for databases unless the databases are running on virtual machines.
- ○ Active geo-replication.

  ✓ That's correct. Active geo-replication can fail over to a secondary database if your primary database fails or needs to be taken offline.

4. To address the company's concern with accidental data deletion, which of these solutions is best?

- ○ Disk caching
- ◉ Soft delete

  ✓ That's correct. With soft delete you can specify a retention period. The data is retained during the retention period and can be recovered.
- ○ Add a resource lock to the storage account

Next unit: Summary and resources

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆