AZ-305T00A Designing Microsoft Azure Infrastructure Solutions

Design a data storage solution for relational data

https://learn.microsoft.com/training/modules/design-data-storage-solution-for-relational-data/

Learning Objectives

- Design for data storage
- Design for Azure SQL databases
- Recommend a solution for database scalability
- Recommend a solution for database availability
- Design security for data at rest, data in transmission, and data in use
- Design for Azure SQL Edge
- Design for Azure Cosmos DB and tables
- Case study
- Learning recap

AZ-305: Design Data Storage Solutions (20-25%):

Design data storage solution for relational data

- Recommend a solution for storing relational data
- Recommend a database service tier and compute tier
- Recommend a solution for database scalability
- Recommend a solution for data protection

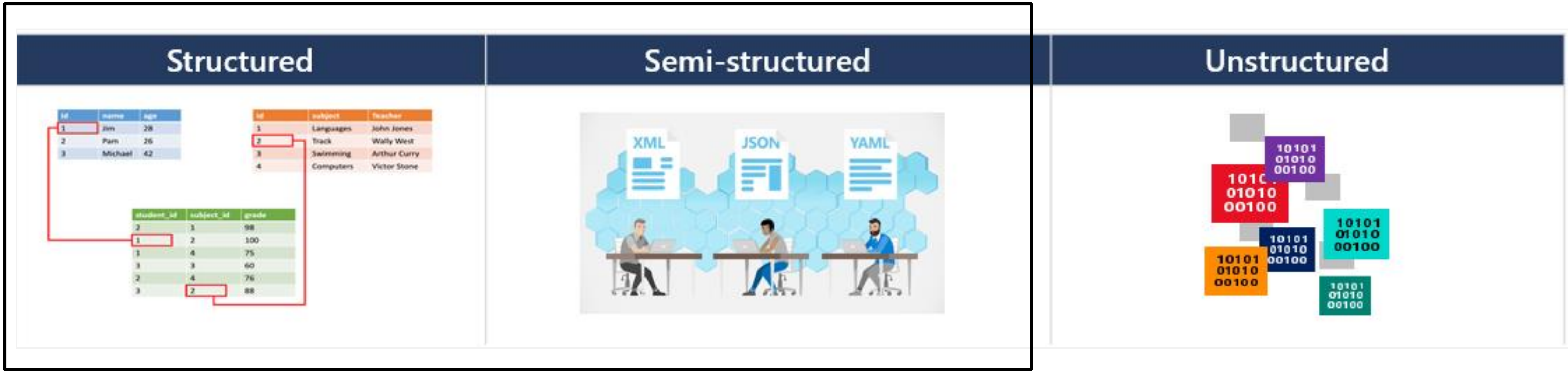
AZ-305: Design Business Continuity Solutions (15-20%):

Design for High Availability

- Recommend a high availability solution for relational data

Design for data storage

Design for structured and semi-structured data



id	name	age
1	Jim	28
2	Pam	26
3	Michael	42

id	subject	Teacher
1	Languages	John Jones
2	Track	Wally West
3	Swimming	Arthur Curry
4	Computers	Victor Stone

student_id	subject_id	grade
2	1	98
1	2	100
1	4	75
3	3	60
2	4	76
3	2	88

To design Azure storage, you first must determine what type of data you have.

- **Structured data** includes relational data and has a shared schema
- **Semi-structured** is less organized than structured data and isn't stored in a relational format
- **Unstructured data** is the least organized type of data

Design for Azure SQL databases

When to use Azure SQL databases

Relational data is a type of structured data that has a shared schema. It's often stored in database tables with rows, columns, and keys, and used for application storage like e-commerce websites.

SQL virtual machines

Best for migrations and applications requiring OS-level access

SQL virtual machine

- SQL Server and OS server access
- Expansive SQL and OS version support
- Automated manageability features

Managed instances

Best for most lift-and-shift migrations to the cloud

Single instance

- SQL Server surface area (vast majority)
- Native virtual network support
- Fully managed service

Instance pool

- Resource sharing between multiple instances to price optimize
- Simplified performance management for multiple databases
- Fully managed service

Databases

Best for modern cloud applications

Single database

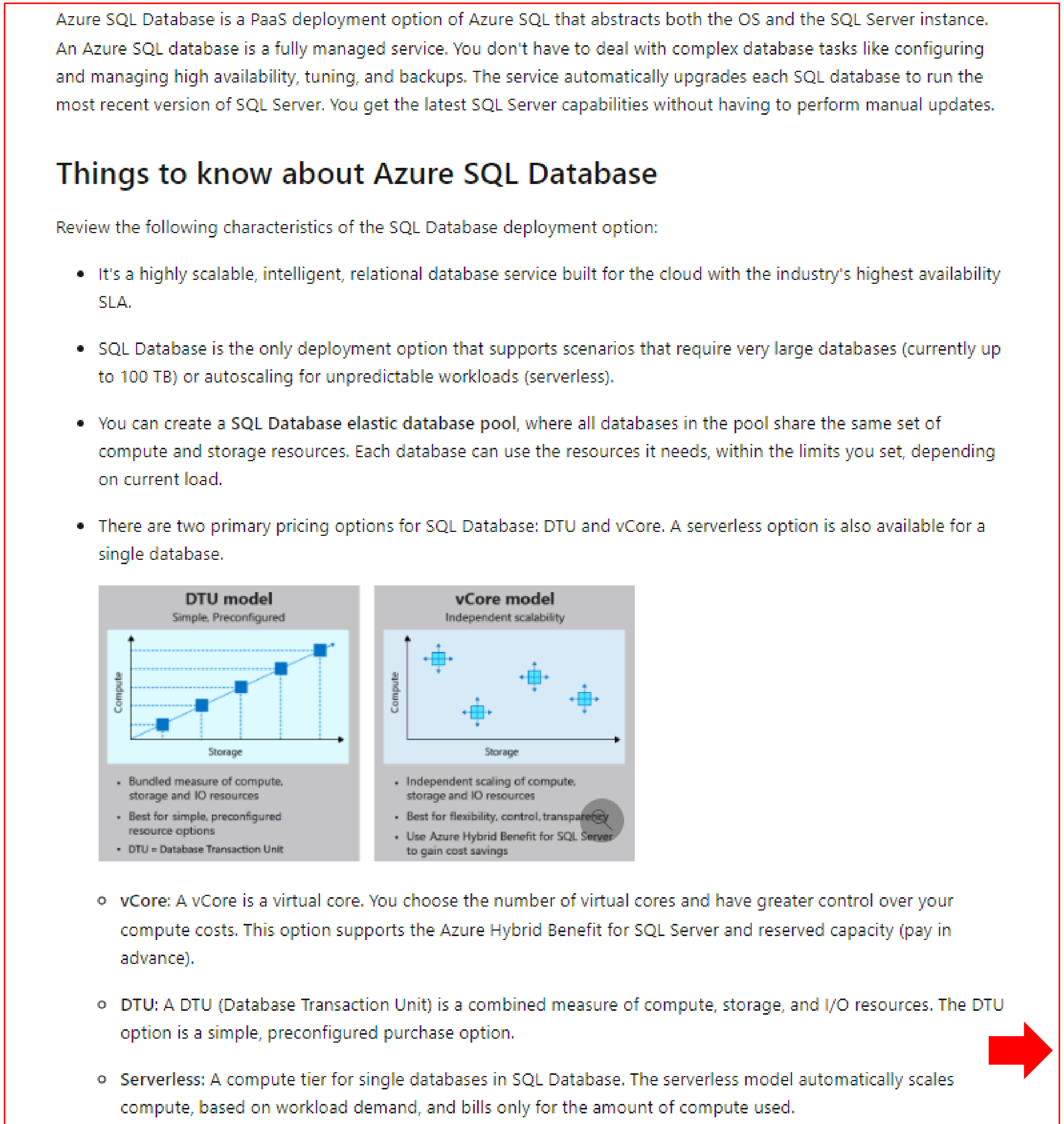
- Hyperscale storage (up to 100TB)
- Serverless compute
- Fully managed service

Elastic pool

- Resource sharing between multiple databases to price optimize
- Simplified performance management for multiple databases
- Fully managed service

Azure SQL Database

Azure SQL Database is a PaaS deployment option of Azure SQL that abstracts both the OS and the SQL Server instance. An Azure SQL database is a fully managed service. You don't have to deal with complex database tasks like configuring and managing high availability, tuning, and backups. The service automatically upgrades each SQL database to run the most recent version of SQL Server. You get the latest SQL Server capabilities without having to perform manual updates.



Azure SQL Database is a PaaS deployment option of Azure SQL that abstracts both the OS and the SQL Server instance. An Azure SQL database is a fully managed service. You don't have to deal with complex database tasks like configuring and managing high availability, tuning, and backups. The service automatically upgrades each SQL database to run the most recent version of SQL Server. You get the latest SQL Server capabilities without having to perform manual updates.

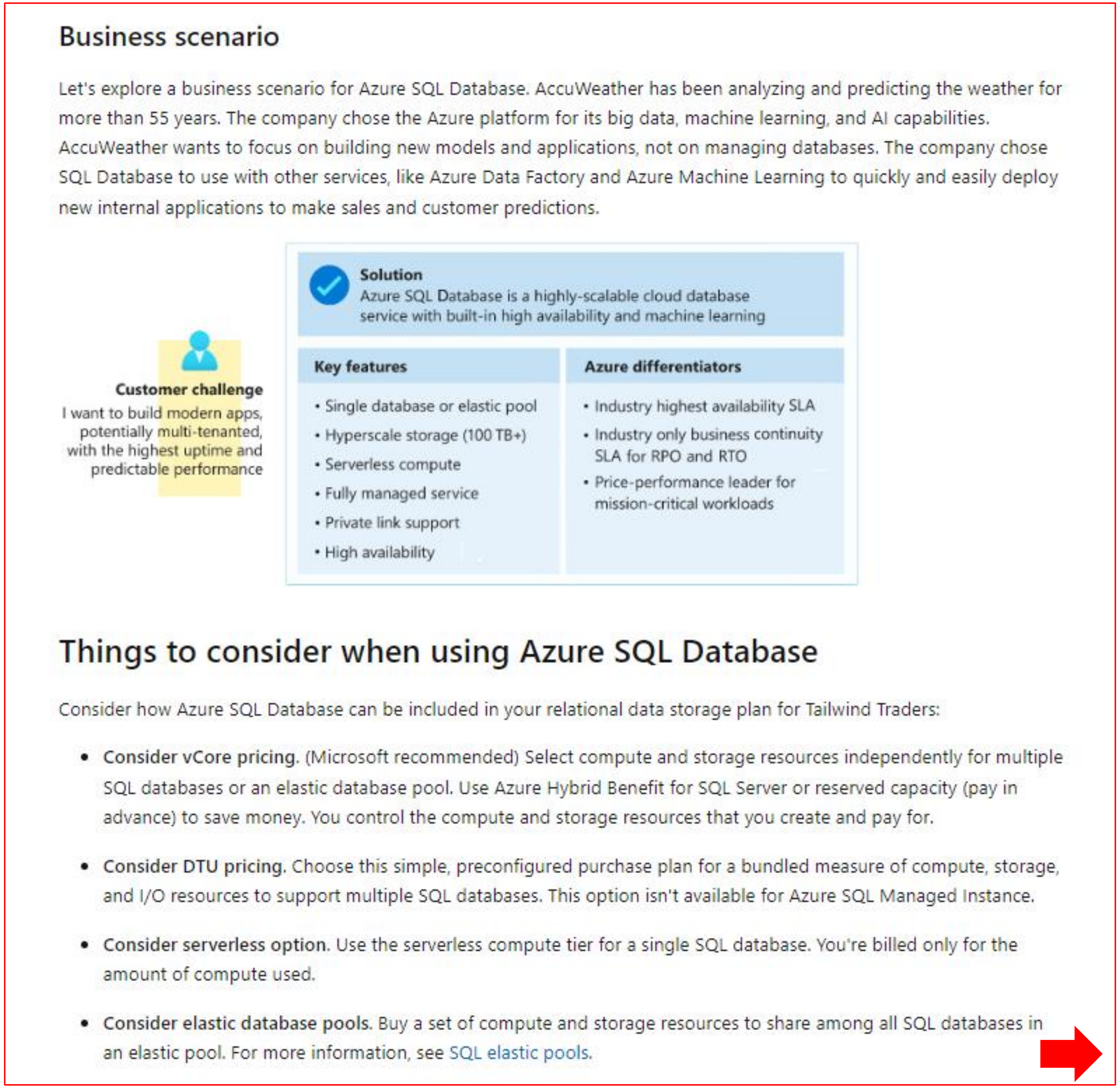
Things to know about Azure SQL Database

Review the following characteristics of the SQL Database deployment option:

- It's a highly scalable, intelligent, relational database service built for the cloud with the industry's highest availability SLA.
- SQL Database is the only deployment option that supports scenarios that require very large databases (currently up to 100 TB) or autoscaling for unpredictable workloads (serverless).
- You can create a **SQL Database elastic database pool**, where all databases in the pool share the same set of compute and storage resources. Each database can use the resources it needs, within the limits you set, depending on current load.
- There are two primary pricing options for SQL Database: DTU and vCore. A serverless option is also available for a single database.

DTU model
Simple, Preconfigured
Compute
Storage
Bundled measure of compute, storage and IO resources
Best for simple, preconfigured resource options
DTU = Database Transaction Unit
vCore model
Independent scalability
Compute
Storage
Independent scaling of compute, storage and IO resources
Best for flexibility, control, transparency
Use Azure Hybrid Benefit for SQL Server to gain cost savings

  - **vCore**: A vCore is a virtual core. You choose the number of virtual cores and have greater control over your compute costs. This option supports the Azure Hybrid Benefit for SQL Server and reserved capacity (pay in advance).
  - **DTU**: A DTU (Database Transaction Unit) is a combined measure of compute, storage, and I/O resources. The DTU option is a simple, preconfigured purchase option.
  - **Serverless**: A compute tier for single databases in SQL Database. The serverless model automatically scales compute, based on workload demand, and bills only for the amount of compute used.



Business scenario

Let's explore a business scenario for Azure SQL Database. AccuWeather has been analyzing and predicting the weather for more than 55 years. The company chose the Azure platform for its big data, machine learning, and AI capabilities. AccuWeather wants to focus on building new models and applications, not on managing databases. The company chose SQL Database to use with other services, like Azure Data Factory and Azure Machine Learning to quickly and easily deploy new internal applications to make sales and customer predictions.

Customer challenge
I want to build modern apps, potentially multi-tenanted, with the highest uptime and predictable performance

Solution
Azure SQL Database is a highly-scalable cloud database service with built-in high availability and machine learning

Key features

- Single database or elastic pool
- Hyperscale storage (100 TB+)
- Serverless compute
- Fully managed service
- Private link support
- High availability

Azure differentiators

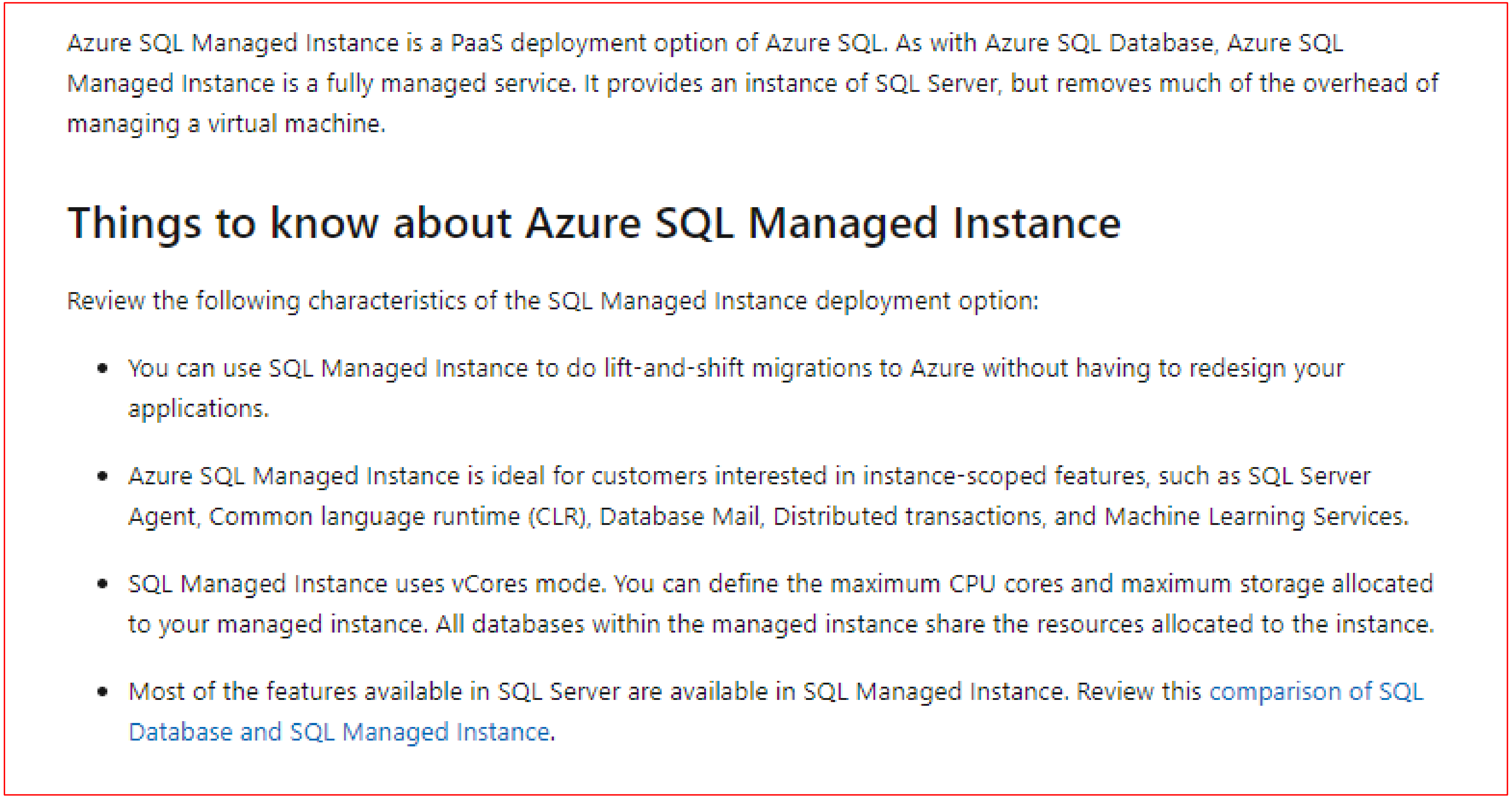
- Industry highest availability SLA
- Industry only business continuity SLA for RPO and RTO
- Price-performance leader for mission-critical workloads

Things to consider when using Azure SQL Database

Consider how Azure SQL Database can be included in your relational data storage plan for Tailwind Traders:

- **Consider vCore pricing.** (Microsoft recommended) Select compute and storage resources independently for multiple SQL databases or an elastic database pool. Use Azure Hybrid Benefit for SQL Server or reserved capacity (pay in advance) to save money. You control the compute and storage resources that you create and pay for.
- **Consider DTU pricing.** Choose this simple, preconfigured purchase plan for a bundled measure of compute, storage, and I/O resources to support multiple SQL databases. This option isn't available for Azure SQL Managed Instance.
- **Consider serverless option.** Use the serverless compute tier for a single SQL database. You're billed only for the amount of compute used.
- **Consider elastic database pools.** Buy a set of compute and storage resources to share among all SQL databases in an elastic pool. For more information, see SQL elastic pools.

Design for Azure SQL Managed Instance

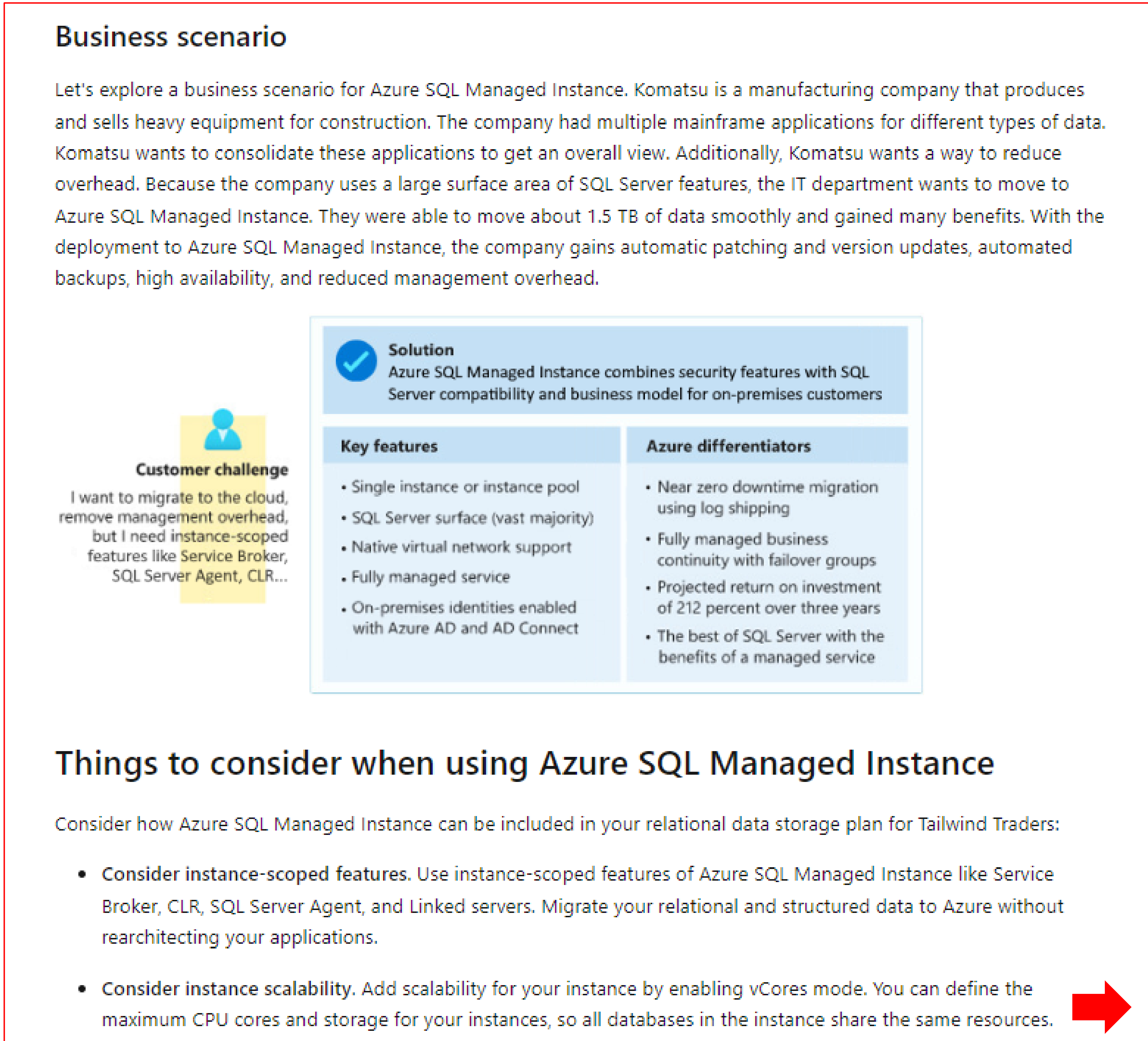


Azure SQL Managed Instance is a PaaS deployment option of Azure SQL. As with Azure SQL Database, Azure SQL Managed Instance is a fully managed service. It provides an instance of SQL Server, but removes much of the overhead of managing a virtual machine.

Things to know about Azure SQL Managed Instance

Review the following characteristics of the SQL Managed Instance deployment option:

- You can use SQL Managed Instance to do lift-and-shift migrations to Azure without having to redesign your applications.
- Azure SQL Managed Instance is ideal for customers interested in instance-scoped features, such as SQL Server Agent, Common language runtime (CLR), Database Mail, Distributed transactions, and Machine Learning Services.
- SQL Managed Instance uses vCores mode. You can define the maximum CPU cores and maximum storage allocated to your managed instance. All databases within the managed instance share the resources allocated to the instance.
- Most of the features available in SQL Server are available in SQL Managed Instance. Review this comparison of SQL Database and SQL Managed Instance.



Business scenario

Let's explore a business scenario for Azure SQL Managed Instance. Komatsu is a manufacturing company that produces and sells heavy equipment for construction. The company had multiple mainframe applications for different types of data. Komatsu wants to consolidate these applications to get an overall view. Additionally, Komatsu wants a way to reduce overhead. Because the company uses a large surface area of SQL Server features, the IT department wants to move to Azure SQL Managed Instance. They were able to move about 1.5 TB of data smoothly and gained many benefits. With the deployment to Azure SQL Managed Instance, the company gains automatic patching and version updates, automated backups, high availability, and reduced management overhead.

Customer challenge

I want to migrate to the cloud, remove management overhead, but I need instance-scoped features like Service Broker, SQL Server Agent, CLR...

Solution

Azure SQL Managed Instance combines security features with SQL Server compatibility and business model for on-premises customers

Key features

- Single instance or instance pool
- SQL Server surface (vast majority)
- Native virtual network support
- Fully managed service
- On-premises identities enabled with Azure AD and AD Connect

Azure differentiators

- Near zero downtime migration using log shipping
- Fully managed business continuity with failover groups
- Projected return on investment of 212 percent over three years
- The best of SQL Server with the benefits of a managed service

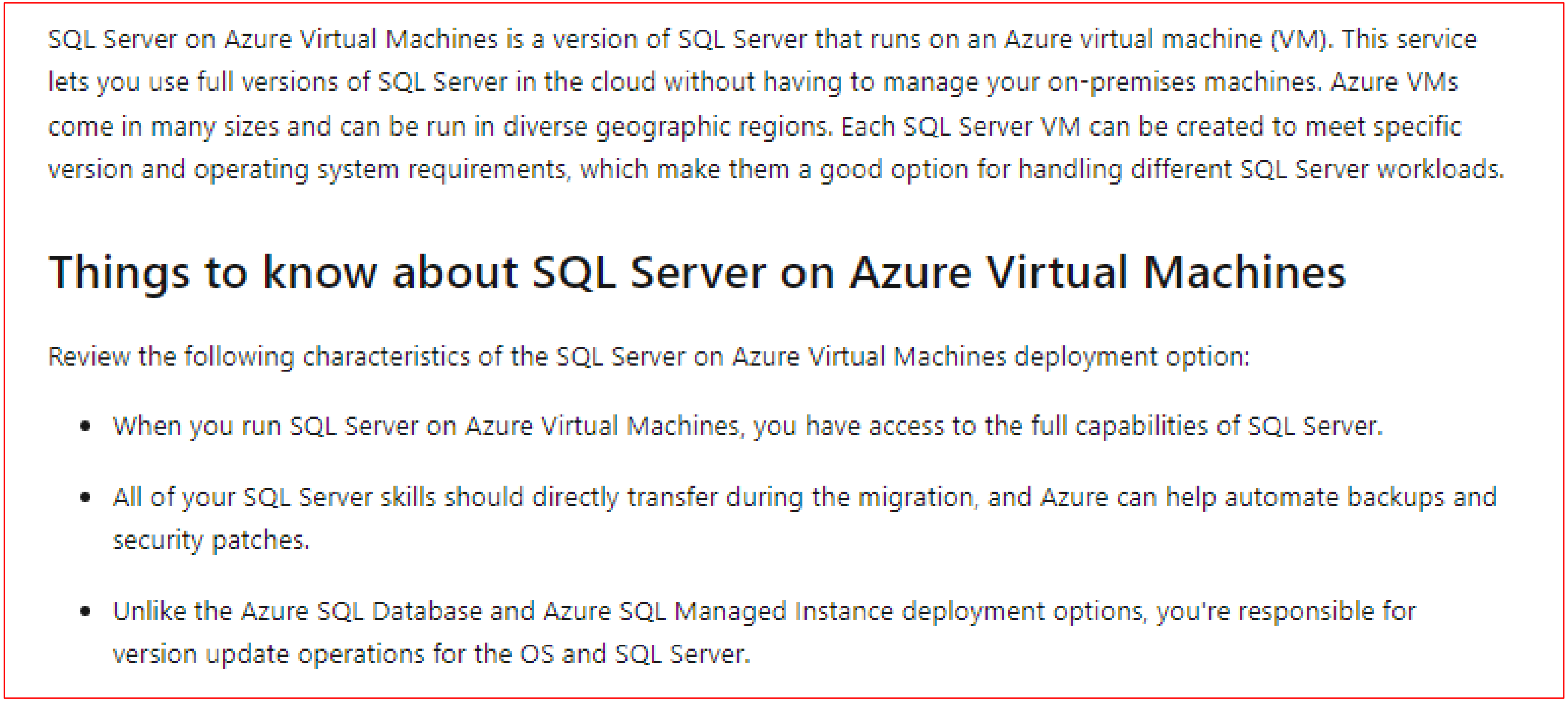
Things to consider when using Azure SQL Managed Instance

Consider how Azure SQL Managed Instance can be included in your relational data storage plan for Tailwind Traders:

- **Consider instance-scoped features.** Use instance-scoped features of Azure SQL Managed Instance like Service Broker, CLR, SQL Server Agent, and Linked servers. Migrate your relational and structured data to Azure without rearchitecting your applications.

- **Consider instance scalability.** Add scalability for your instance by enabling vCores mode. You can define the maximum CPU cores and storage for your instances, so all databases in the instance share the same resources.

Design for SQL Server on Azure Virtual Machines

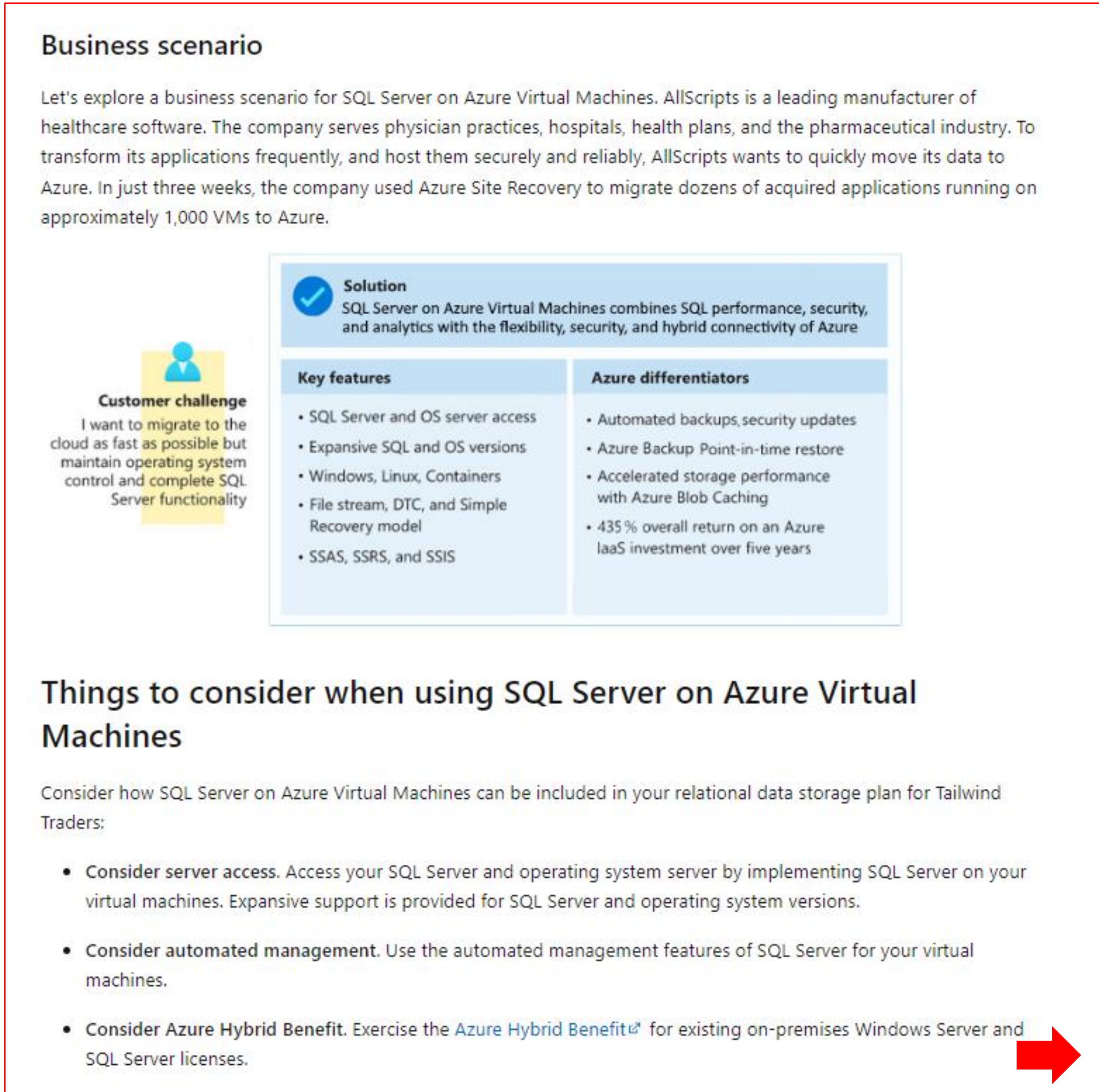


SQL Server on Azure Virtual Machines is a version of SQL Server that runs on an Azure virtual machine (VM). This service lets you use full versions of SQL Server in the cloud without having to manage your on-premises machines. Azure VMs come in many sizes and can be run in diverse geographic regions. Each SQL Server VM can be created to meet specific version and operating system requirements, which make them a good option for handling different SQL Server workloads.

Things to know about SQL Server on Azure Virtual Machines

Review the following characteristics of the SQL Server on Azure Virtual Machines deployment option:

- When you run SQL Server on Azure Virtual Machines, you have access to the full capabilities of SQL Server.
- All of your SQL Server skills should directly transfer during the migration, and Azure can help automate backups and security patches.
- Unlike the Azure SQL Database and Azure SQL Managed Instance deployment options, you're responsible for version update operations for the OS and SQL Server.



Business scenario

Let's explore a business scenario for SQL Server on Azure Virtual Machines. AllScripts is a leading manufacturer of healthcare software. The company serves physician practices, hospitals, health plans, and the pharmaceutical industry. To transform its applications frequently, and host them securely and reliably, AllScripts wants to quickly move its data to Azure. In just three weeks, the company used Azure Site Recovery to migrate dozens of acquired applications running on approximately 1,000 VMs to Azure.

Customer challenge

I want to migrate to the cloud as fast as possible but maintain operating system control and complete SQL Server functionality

Solution

SQL Server on Azure Virtual Machines combines SQL performance, security, and analytics with the flexibility, security, and hybrid connectivity of Azure

Key features

- SQL Server and OS server access
- Expansive SQL and OS versions
- Windows, Linux, Containers
- File stream, DTC, and Simple Recovery model
- SSAS, SSRS, and SSIS

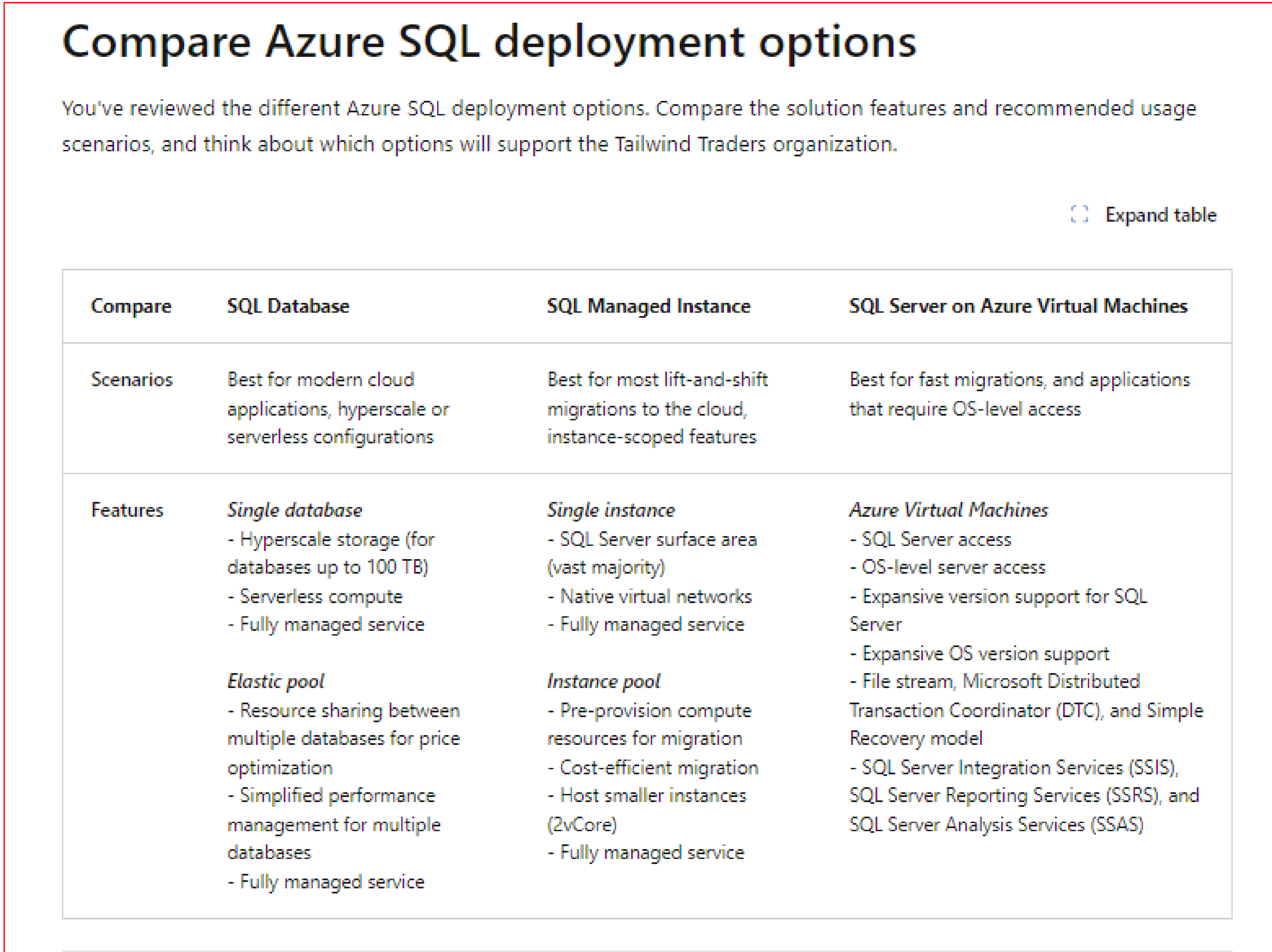
Azure differentiators

- Automated backups, security updates
- Azure Backup Point-in-time restore
- Accelerated storage performance with Azure Blob Caching
- 435 % overall return on an Azure IaaS investment over five years

Things to consider when using SQL Server on Azure Virtual Machines

Consider how SQL Server on Azure Virtual Machines can be included in your relational data storage plan for Tailwind Traders:

- **Consider server access.** Access your SQL Server and operating system server by implementing SQL Server on your virtual machines. Expansive support is provided for SQL Server and operating system versions.
- **Consider automated management.** Use the automated management features of SQL Server for your virtual machines.
- **Consider Azure Hybrid Benefit.** Exercise the Azure Hybrid Benefit for existing on-premises Windows Server and SQL Server licenses.



Compare Azure SQL deployment options

You've reviewed the different Azure SQL deployment options. Compare the solution features and recommended usage scenarios, and think about which options will support the Tailwind Traders organization.

Compare	SQL Database	SQL Managed Instance	SQL Server on Azure Virtual Machines
Scenarios	Best for modern cloud applications, hyperscale or serverless configurations	Best for most lift-and-shift migrations to the cloud, instance-scoped features	Best for fast migrations, and applications that require OS-level access
Features	*Single database* - Hyperscale storage (for databases up to 100 TB) - Serverless compute - Fully managed service *Elastic pool* - Resource sharing between multiple databases for price optimization - Simplified performance management for multiple databases - Fully managed service	*Single instance* - SQL Server surface area (vast majority) - Native virtual networks - Fully managed service *Instance pool* - Pre-provision compute resources for migration - Cost-efficient migration - Host smaller instances (2vCore) - Fully managed service	*Azure Virtual Machines* - SQL Server access - OS-level server access - Expansive version support for SQL Server - Expansive OS version support - File stream, Microsoft Distributed Transaction Coordinator (DTC), and Simple Recovery model - SQL Server Integration Services (SSIS), SQL Server Reporting Services (SSRS), and SQL Server Analysis Services (SSAS)

Recommend a solution for database scalability

Database scaling strategy

The following table identifies scenarios that require different scaling solutions

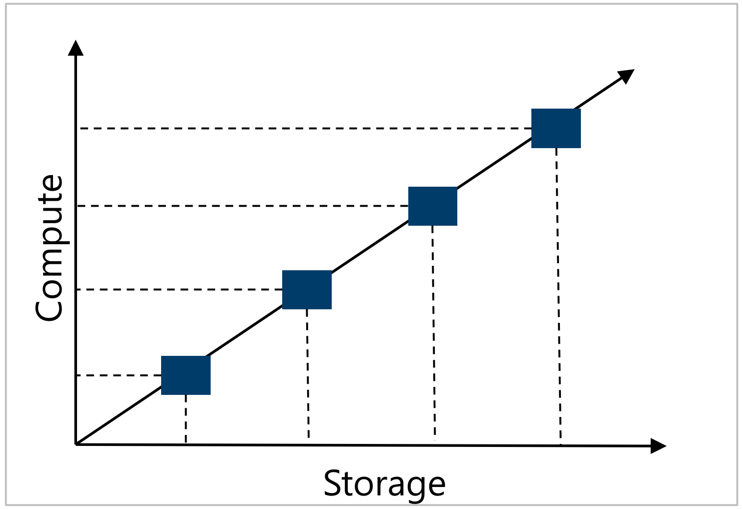
Requirement	Solution
Do you have to manage and scale multiple Azure SQL databases that have varying and predictable resource requirements?	SQL elastic pools.
Are you developing a new application with a single database that you want to test before launching it to thousands of users?	Azure SQL Database or SQL Managed Instance
Do you need to optimize the price performance for a group of databases within a prescribed budget while delivering performance elasticity for each database?	SQL elastic pools.

Consider cost together with your scaling strategy to find an optimal solution

Recommend a solution for database availability

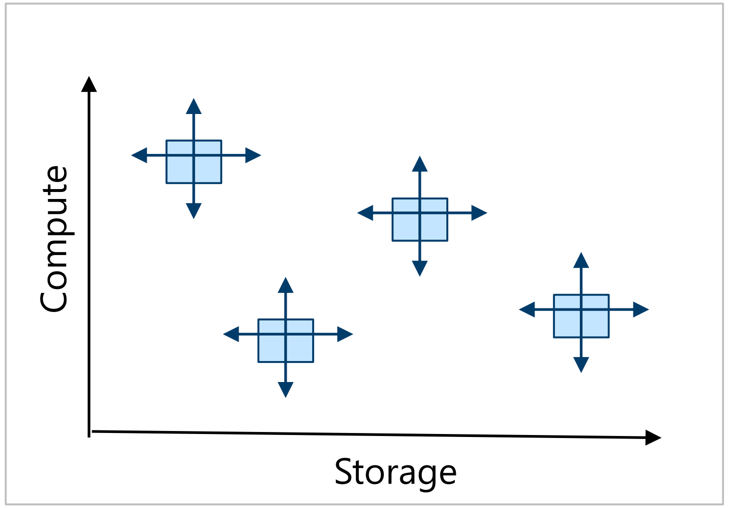
Select an Azure SQL Database pricing model

DTU



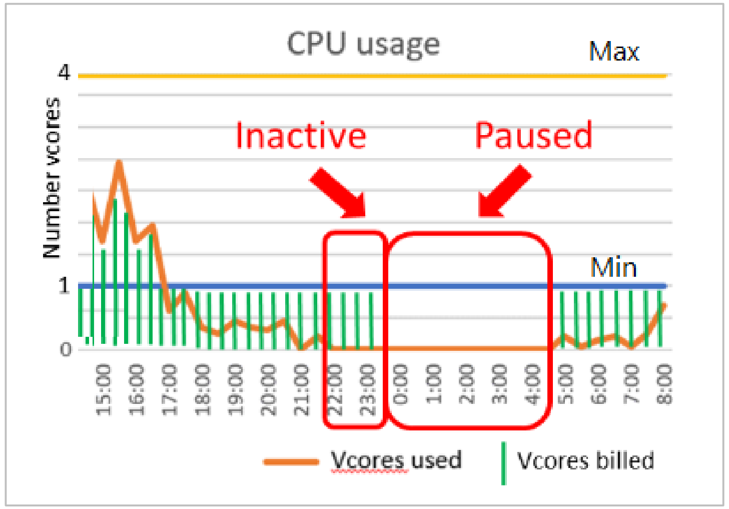
- A simple, preconfigured purchase option.
- A blended measure of CPU, memory, reads, and writes.

vCore

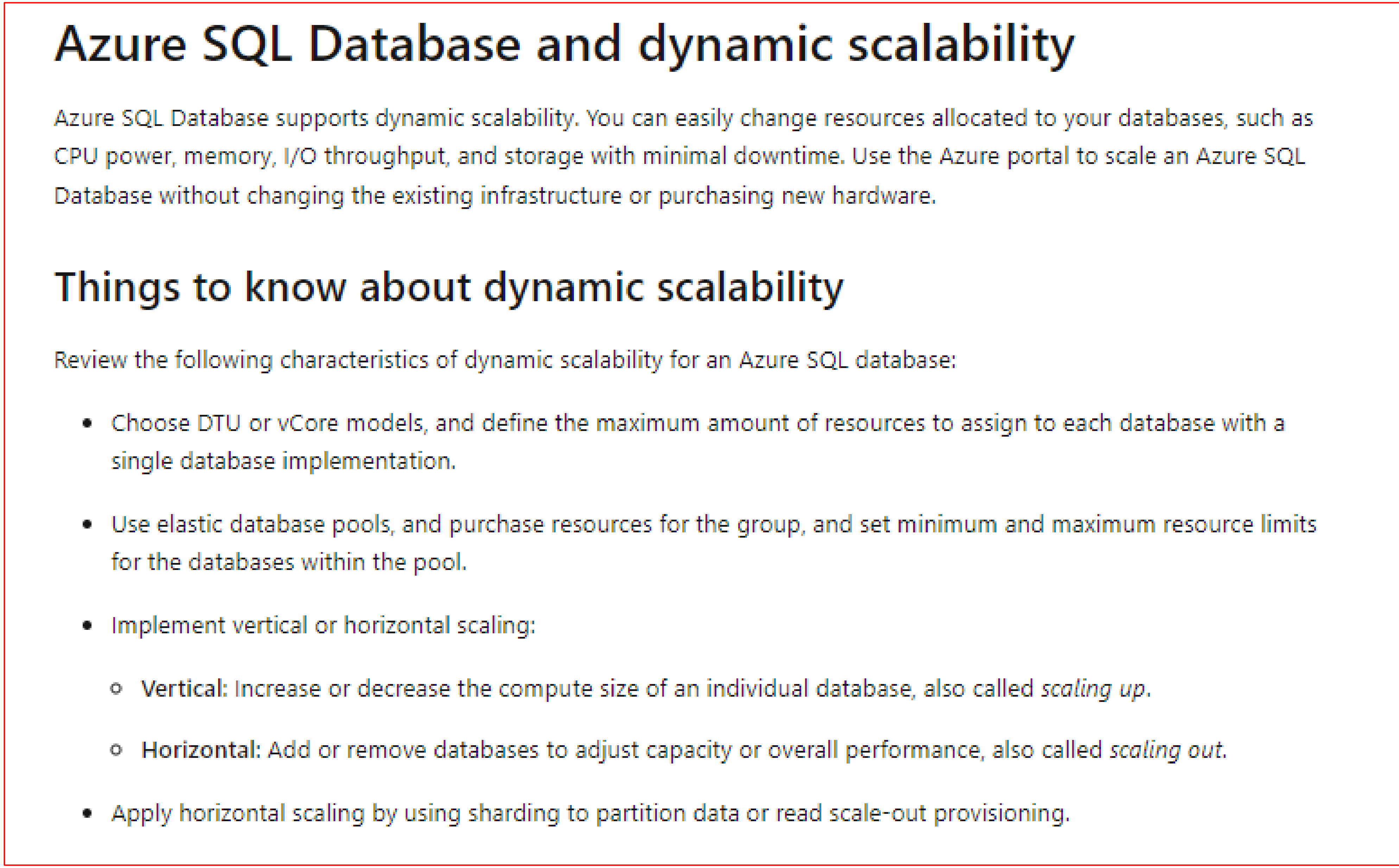


- Flexibility, control and transparency
- Independent scaling of compute, storage, and I/O resources

Serverless



- Intermittent, unpredictable usage
- Automatically scales compute, based on workload demand



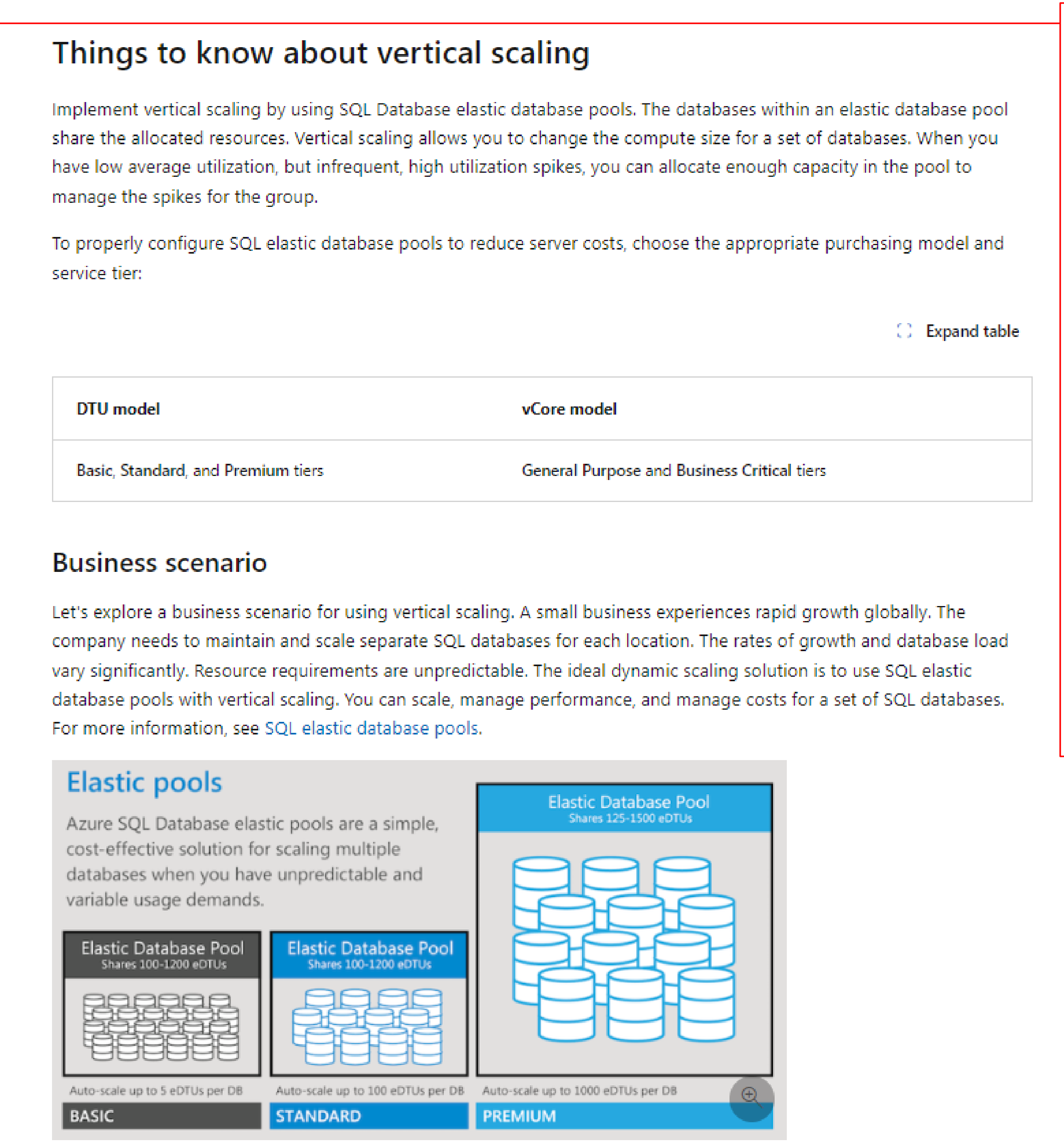
Azure SQL Database and dynamic scalability

Azure SQL Database supports dynamic scalability. You can easily change resources allocated to your databases, such as CPU power, memory, I/O throughput, and storage with minimal downtime. Use the Azure portal to scale an Azure SQL Database without changing the existing infrastructure or purchasing new hardware.

Things to know about dynamic scalability

Review the following characteristics of dynamic scalability for an Azure SQL database:

- Choose DTU or vCore models, and define the maximum amount of resources to assign to each database with a single database implementation.
- Use elastic database pools, and purchase resources for the group, and set minimum and maximum resource limits for the databases within the pool.
- Implement vertical or horizontal scaling:
  - **Vertical:** Increase or decrease the compute size of an individual database, also called *scaling up*.
  - **Horizontal:** Add or remove databases to adjust capacity or overall performance, also called *scaling out*.
- Apply horizontal scaling by using sharding to partition data or read scale-out provisioning.



Things to know about vertical scaling

Implement vertical scaling by using SQL Database elastic database pools. The databases within an elastic database pool share the allocated resources. Vertical scaling allows you to change the compute size for a set of databases. When you have low average utilization, but infrequent, high utilization spikes, you can allocate enough capacity in the pool to manage the spikes for the group.

To properly configure SQL elastic database pools to reduce server costs, choose the appropriate purchasing model and service tier:

⛶ Expand table

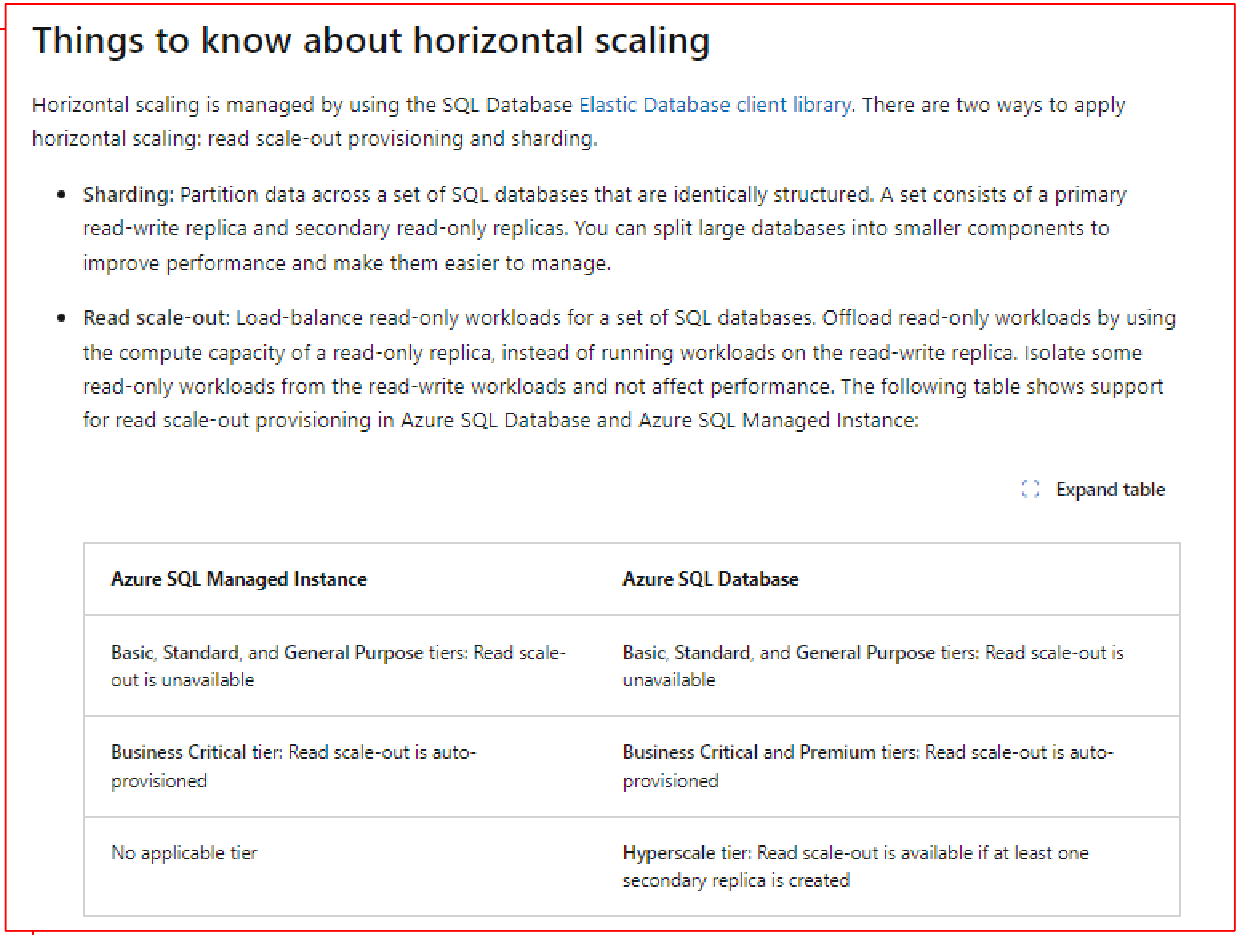
DTU model	vCore model
Basic, Standard, and Premium tiers	General Purpose and Business Critical tiers

Business scenario

Let's explore a business scenario for using vertical scaling. A small business experiences rapid growth globally. The company needs to maintain and scale separate SQL databases for each location. The rates of growth and database load vary significantly. Resource requirements are unpredictable. The ideal dynamic scaling solution is to use SQL elastic database pools with vertical scaling. You can scale, manage performance, and manage costs for a set of SQL databases. For more information, see SQL elastic database pools.

Elastic pools

Azure SQL Database elastic pools are a simple, cost-effective solution for scaling multiple databases when you have unpredictable and variable usage demands.



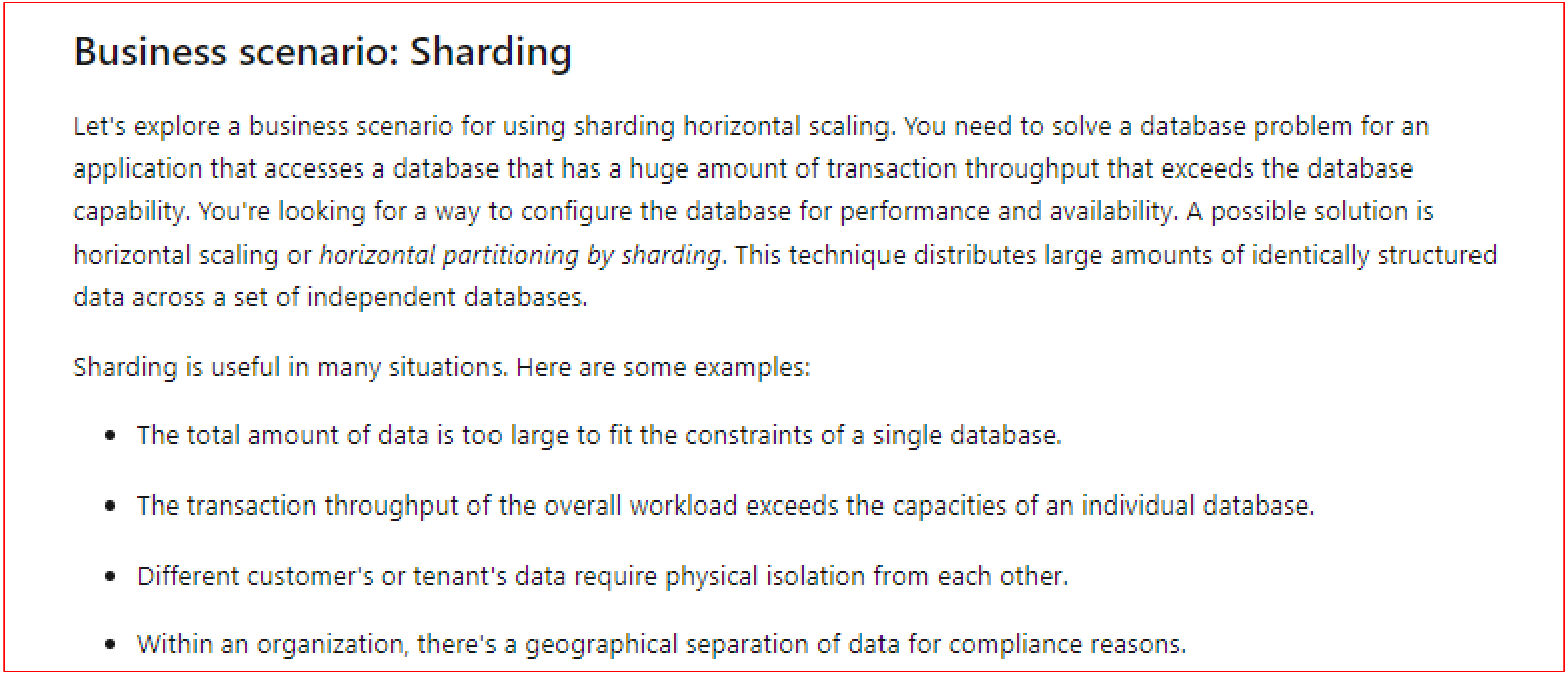
Things to know about horizontal scaling

Horizontal scaling is managed by using the SQL Database Elastic Database client library. There are two ways to apply horizontal scaling: read scale-out provisioning and sharding.

- **Sharding**: Partition data across a set of SQL databases that are identically structured. A set consists of a primary read-write replica and secondary read-only replicas. You can split large databases into smaller components to improve performance and make them easier to manage.
- **Read scale-out**: Load-balance read-only workloads for a set of SQL databases. Offload read-only workloads by using the compute capacity of a read-only replica, instead of running workloads on the read-write replica. Isolate some read-only workloads from the read-write workloads and not affect performance. The following table shows support for read scale-out provisioning in Azure SQL Database and Azure SQL Managed Instance:

⛶ Expand table

Azure SQL Managed Instance	Azure SQL Database
Basic, Standard, and General Purpose tiers: Read scale-out is unavailable	Basic, Standard, and General Purpose tiers: Read scale-out is unavailable
Business Critical tier: Read scale-out is auto-provisioned	Business Critical and Premium tiers: Read scale-out is auto-provisioned
No applicable tier	Hyperscale tier: Read scale-out is available if at least one secondary replica is created

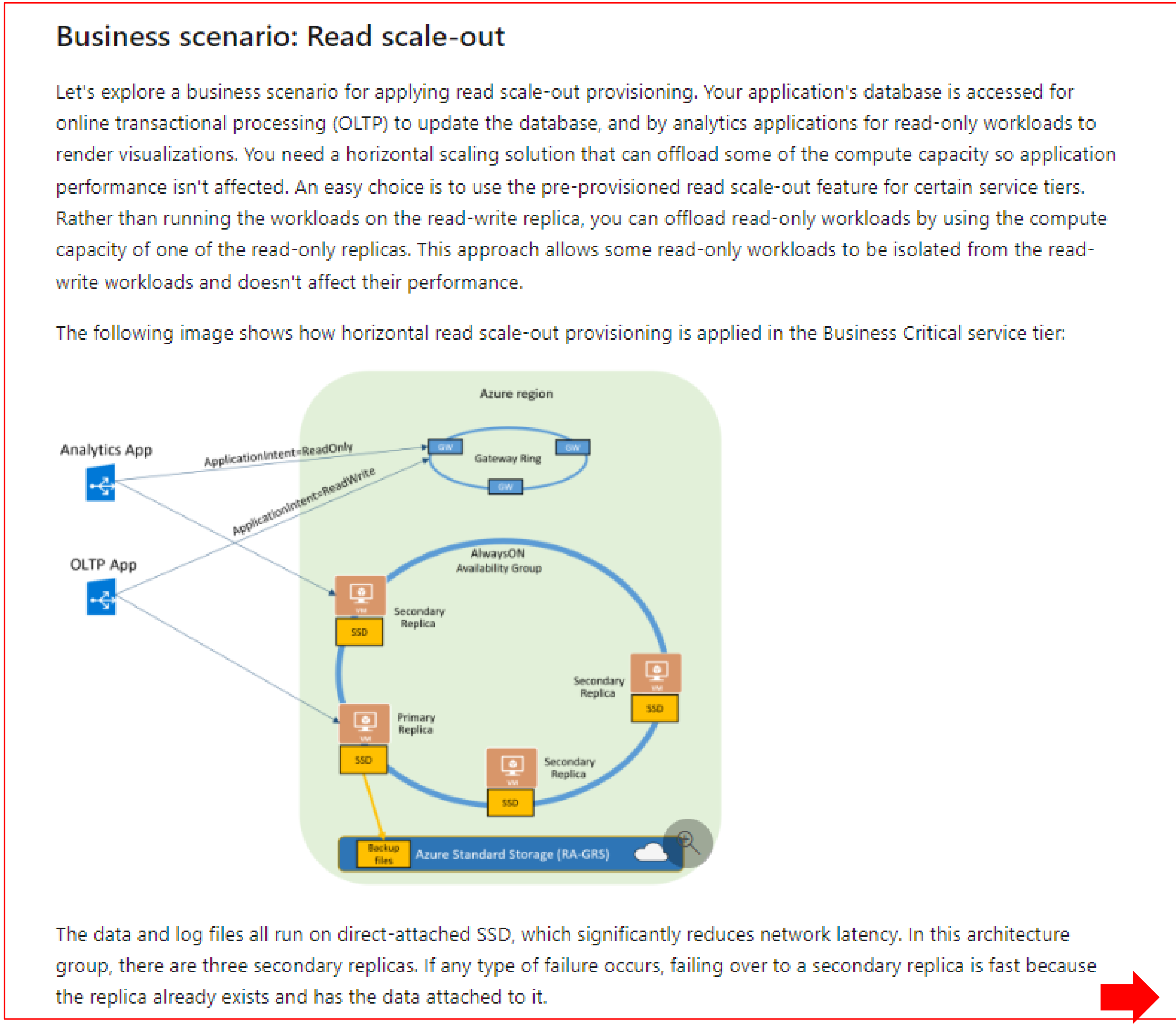


Business scenario: Sharding

Let's explore a business scenario for using sharding horizontal scaling. You need to solve a database problem for an application that accesses a database that has a huge amount of transaction throughput that exceeds the database capability. You're looking for a way to configure the database for performance and availability. A possible solution is horizontal scaling or *horizontal partitioning by sharding*. This technique distributes large amounts of identically structured data across a set of independent databases.

Sharding is useful in many situations. Here are some examples:

- The total amount of data is too large to fit the constraints of a single database.
- The transaction throughput of the overall workload exceeds the capacities of an individual database.
- Different customer's or tenant's data require physical isolation from each other.
- Within an organization, there's a geographical separation of data for compliance reasons.

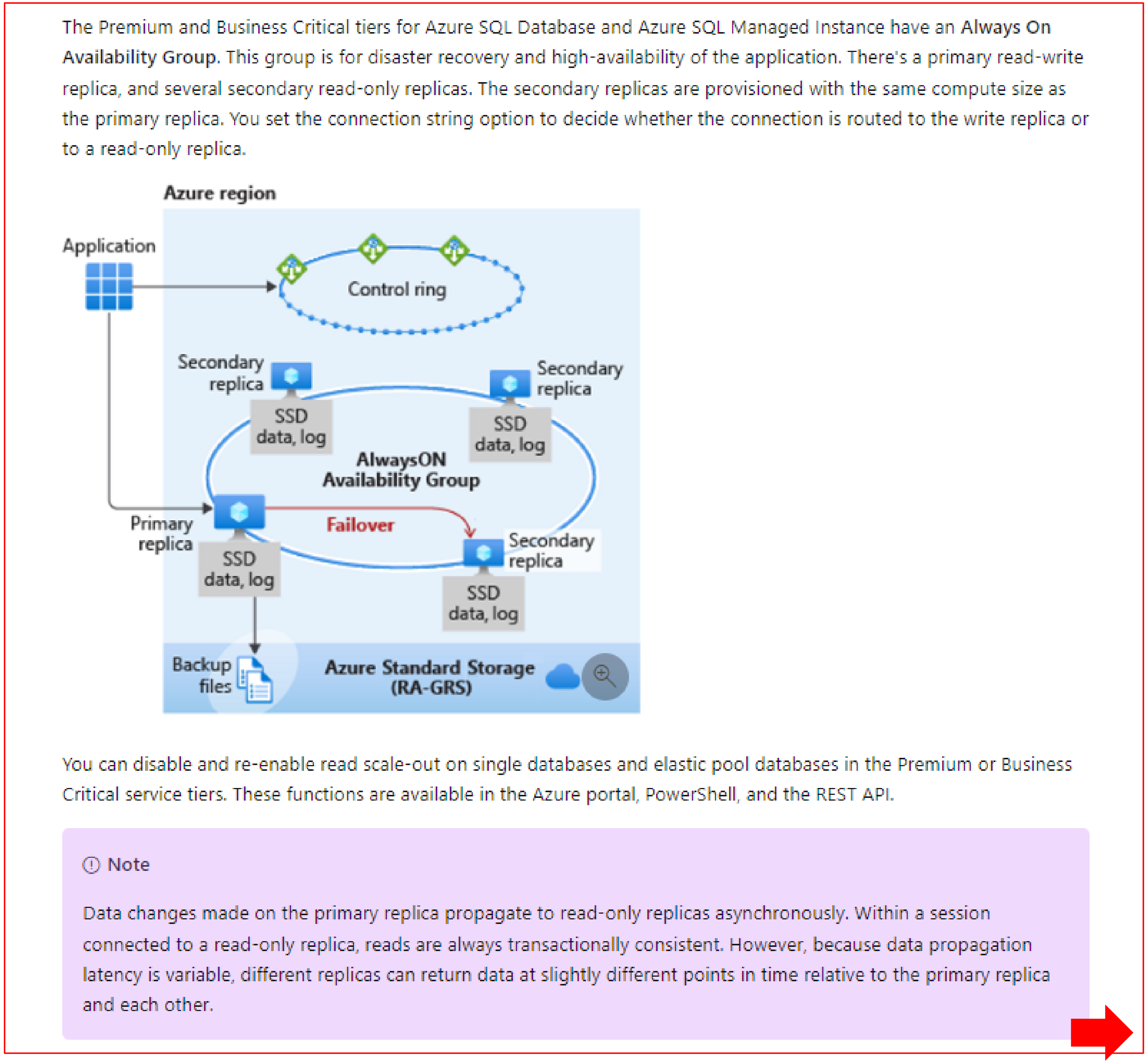


Business scenario: Read scale-out

Let's explore a business scenario for applying read scale-out provisioning. Your application's database is accessed for online transactional processing (OLTP) to update the database, and by analytics applications for read-only workloads to render visualizations. You need a horizontal scaling solution that can offload some of the compute capacity so application performance isn't affected. An easy choice is to use the pre-provisioned read scale-out feature for certain service tiers. Rather than running the workloads on the read-write replica, you can offload read-only workloads by using the compute capacity of one of the read-only replicas. This approach allows some read-only workloads to be isolated from the read-write workloads and doesn't affect their performance.

The following image shows how horizontal read scale-out provisioning is applied in the Business Critical service tier:

The data and log files all run on direct-attached SSD, which significantly reduces network latency. In this architecture group, there are three secondary replicas. If any type of failure occurs, failing over to a secondary replica is fast because the replica already exists and has the data attached to it.

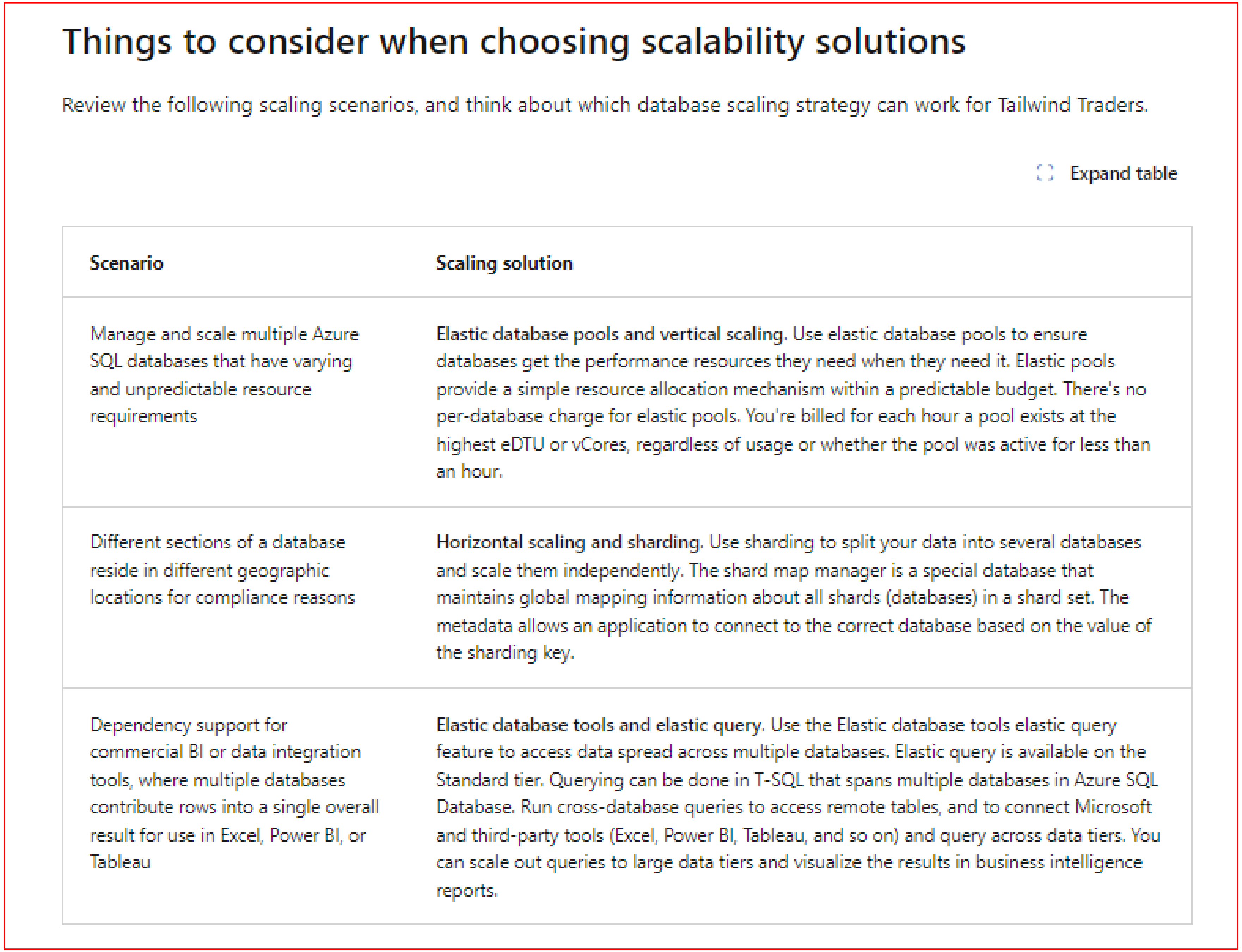


The Premium and Business Critical tiers for Azure SQL Database and Azure SQL Managed Instance have an **Always On Availability Group**. This group is for disaster recovery and high-availability of the application. There's a primary read-write replica, and several secondary read-only replicas. The secondary replicas are provisioned with the same compute size as the primary replica. You set the connection string option to decide whether the connection is routed to the write replica or to a read-only replica.

You can disable and re-enable read scale-out on single databases and elastic pool databases in the Premium or Business Critical service tiers. These functions are available in the Azure portal, PowerShell, and the REST API.

ⓘ Note

Data changes made on the primary replica propagate to read-only replicas asynchronously. Within a session connected to a read-only replica, reads are always transactionally consistent. However, because data propagation latency is variable, different replicas can return data at slightly different points in time relative to the primary replica and each other.



Things to consider when choosing scalability solutions

Review the following scaling scenarios, and think about which database scaling strategy can work for Tailwind Traders.

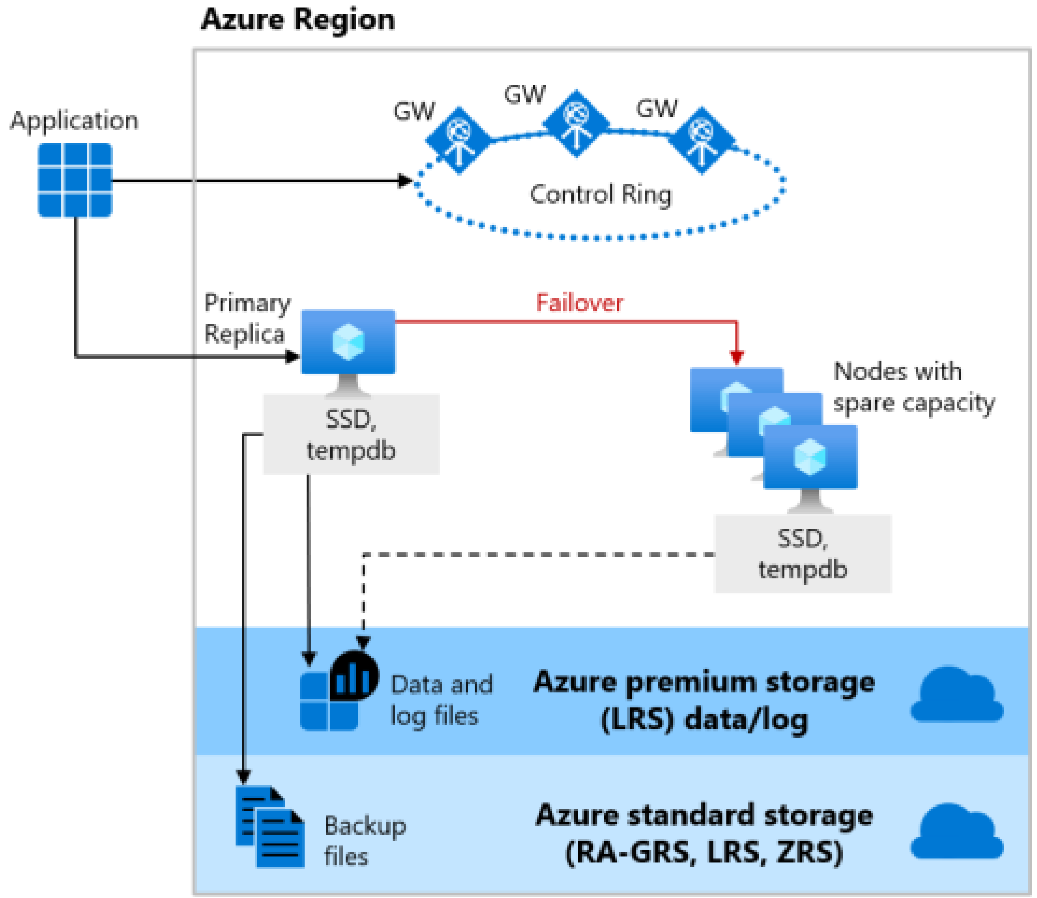
Scenario	Scaling solution
Manage and scale multiple Azure SQL databases that have varying and unpredictable resource requirements	**Elastic database pools and vertical scaling.** Use elastic database pools to ensure databases get the performance resources they need when they need it. Elastic pools provide a simple resource allocation mechanism within a predictable budget. There's no per-database charge for elastic pools. You're billed for each hour a pool exists at the highest eDTU or vCores, regardless of usage or whether the pool was active for less than an hour.
Different sections of a database reside in different geographic locations for compliance reasons	**Horizontal scaling and sharding.** Use sharding to split your data into several databases and scale them independently. The shard map manager is a special database that maintains global mapping information about all shards (databases) in a shard set. The metadata allows an application to connect to the correct database based on the value of the sharding key.
Dependency support for commercial BI or data integration tools, where multiple databases contribute rows into a single overall result for use in Excel, Power BI, or Tableau	**Elastic database tools and elastic query.** Use the Elastic database tools elastic query feature to access data spread across multiple databases. Elastic query is available on the Standard tier. Querying can be done in T-SQL that spans multiple databases in Azure SQL Database. Run cross-database queries to access remote tables, and to connect Microsoft and third-party tools (Excel, Power BI, Tableau, and so on) and query across data tiers. You can scale out queries to large data tiers and visualize the results in business intelligence reports.

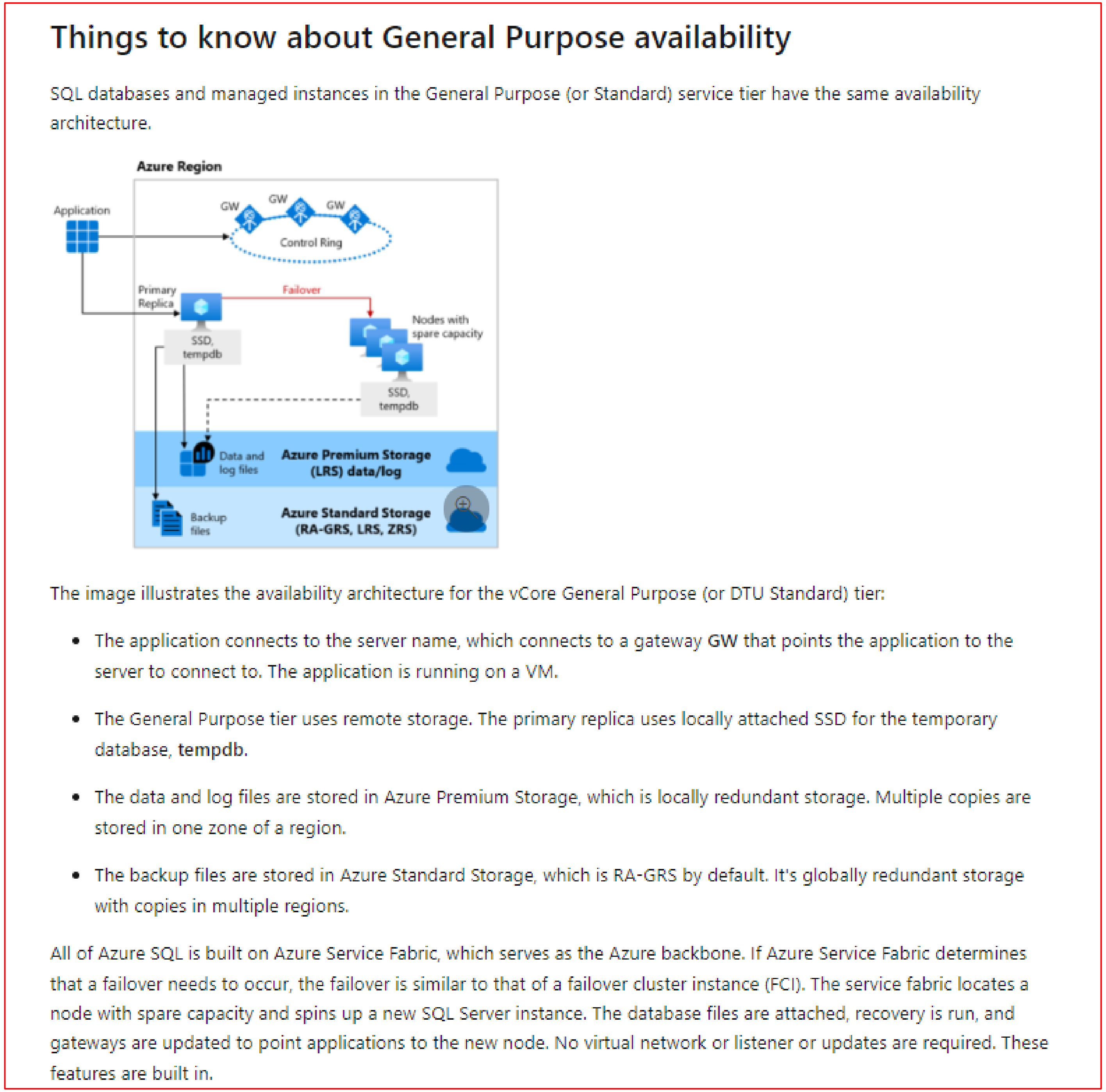
Recommend a solution for database availability

High availability with the General Purpose/Standard tier

Azure SQL Database offers three service tiers that are designed for different types of applications:

- Designed for common workloads
- Budget oriented balanced compute and storage
- Uses nodes with spare capacity to spin up a new SQL Server instances
- Uses LRS and RA-GRS (backup files)





Things to know about General Purpose availability

SQL databases and managed instances in the General Purpose (or Standard) service tier have the same availability architecture.

The image illustrates the availability architecture for the vCore General Purpose (or DTU Standard) tier:

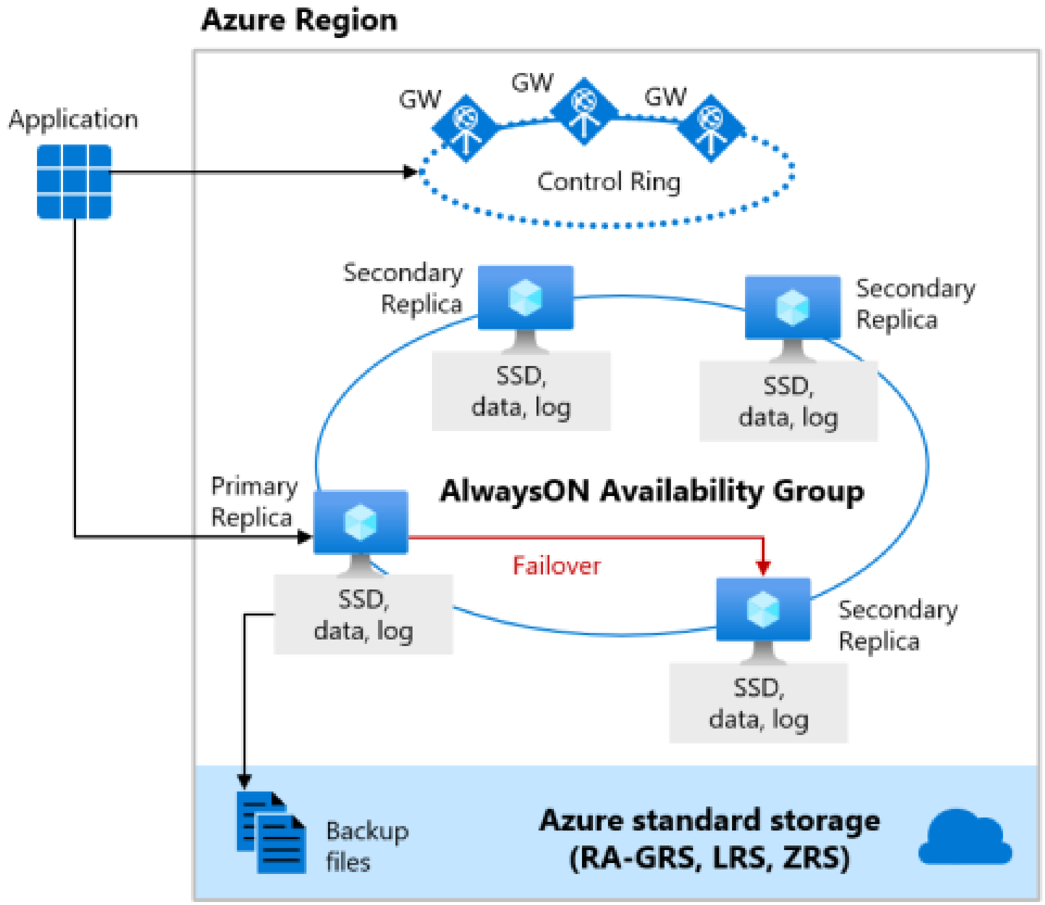
- The application connects to the server name, which connects to a gateway **GW** that points the application to the server to connect to. The application is running on a VM.
- The General Purpose tier uses remote storage. The primary replica uses locally attached SSD for the temporary database, **tempdb**.
- The data and log files are stored in Azure Premium Storage, which is locally redundant storage. Multiple copies are stored in one zone of a region.
- The backup files are stored in Azure Standard Storage, which is RA-GRS by default. 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 is similar to that of a failover cluster instance (FCI). The service fabric locates a node with spare capacity and spins up a new SQL Server instance. The database files are attached, recovery is run, and gateways are updated to point applications to the new node. No virtual network or listener or updates are required. These features are built in.

High availability with the Business Critical/Premium tier

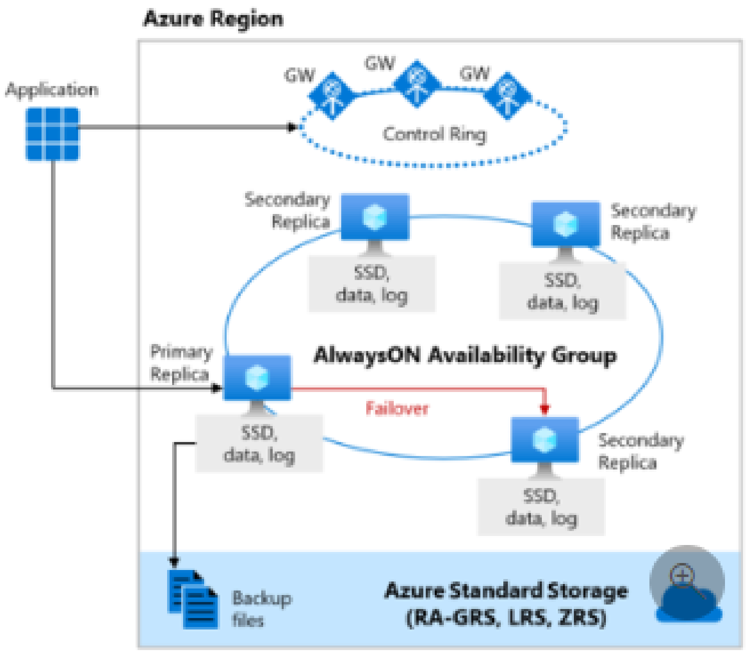
Azure SQL Database offers three service tiers designed for different types of applications:

- Designed for OLTP applications
- High transaction rate and low I/O latency
- Offers the highest resilience to failures by using several isolated replicas
- Deploys an Always On availability group using multiple synchronously updated replicas
- Uses local SSD storage and RA-GRS (backup files)



Things to know about Business Critical availability

In the Business Critical (or Premium) tier, you can generally achieve the highest performance and availability of all Azure SQL service tiers. This tier is meant for mission-critical applications that need low latency and minimal downtime.



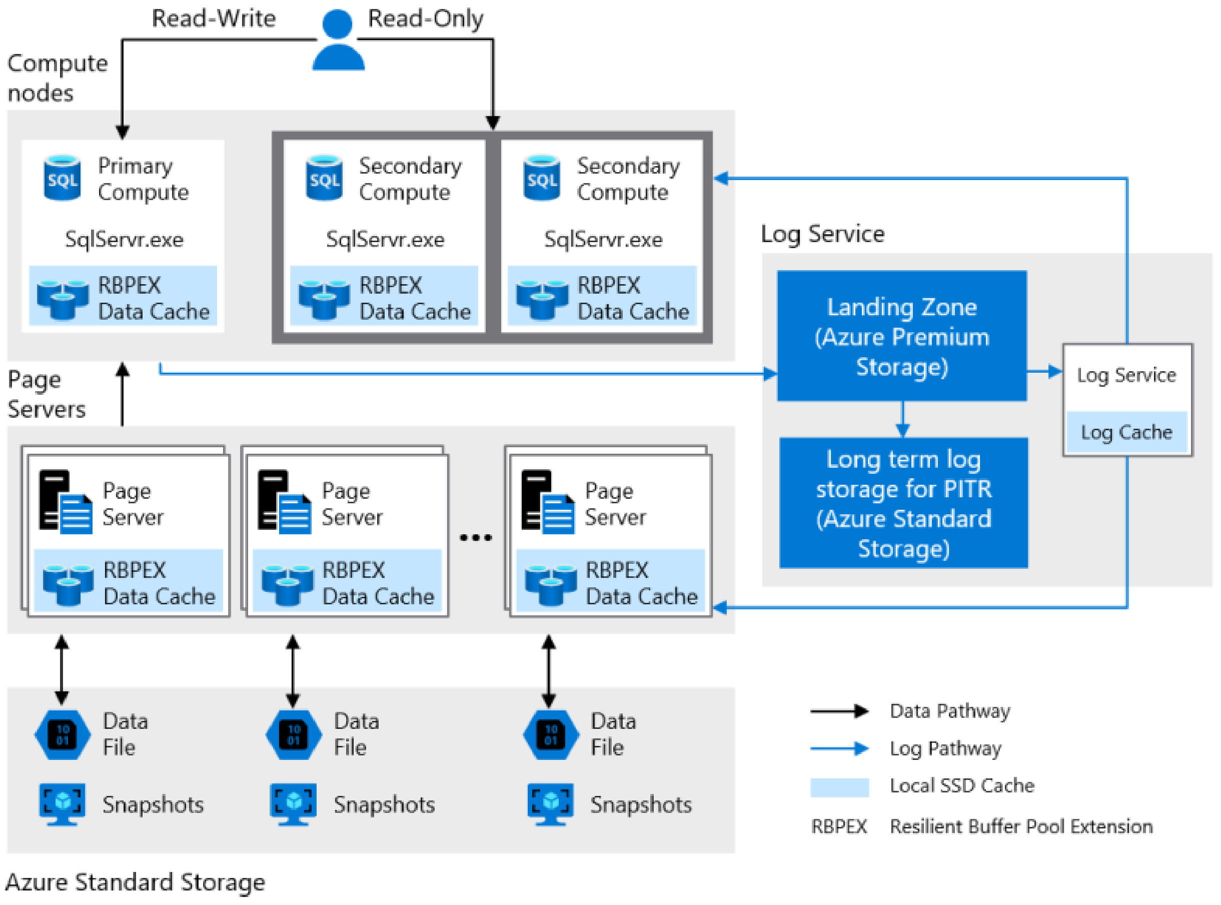
The image illustrates the availability architecture for the vCore Business Critical (or DTU Premium) tier:

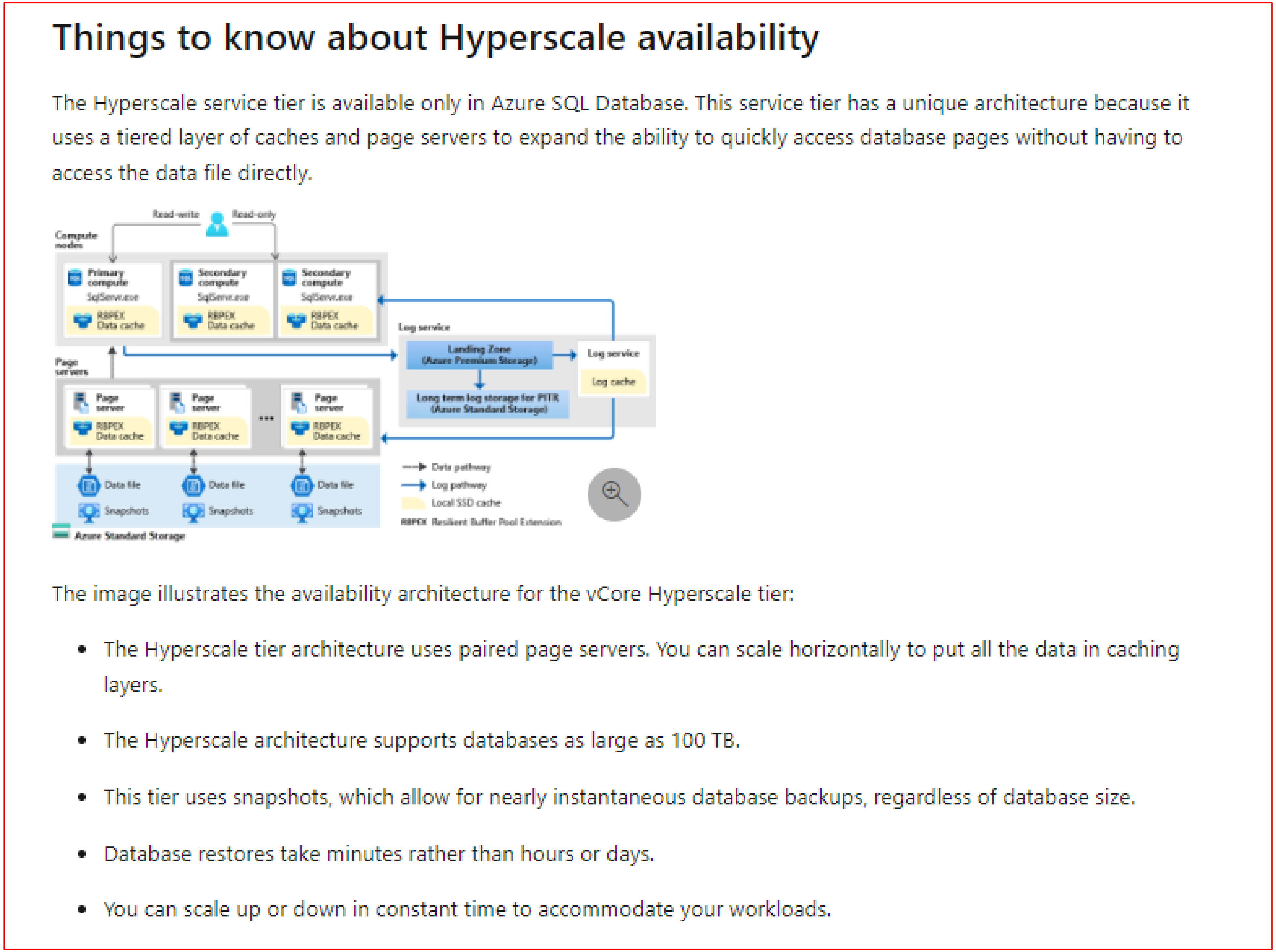
- Database availability in the Business Critical tier is like deploying an Always On availability group behind the scenes.
- Unlike the General Purpose tier, the data and log files all run on direct-attached SSD, which significantly reduces network latency.
- In this tier, there are three secondary replicas.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

High availability with the Hyperscale tier

Azure SQL Database offers three service tiers that are designed for different types of applications:

- Designed for very large OLTP databases – as large as 100 TB
- Able to autoscale storage and scale compute
- Captures instantaneous backups (using snapshots)
- Restores in minutes rather than hours and days
- Scale up or down in real time to accommodate workload changes





Things to know about Hyperscale availability

The Hyperscale service tier is available only in Azure SQL Database. 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.

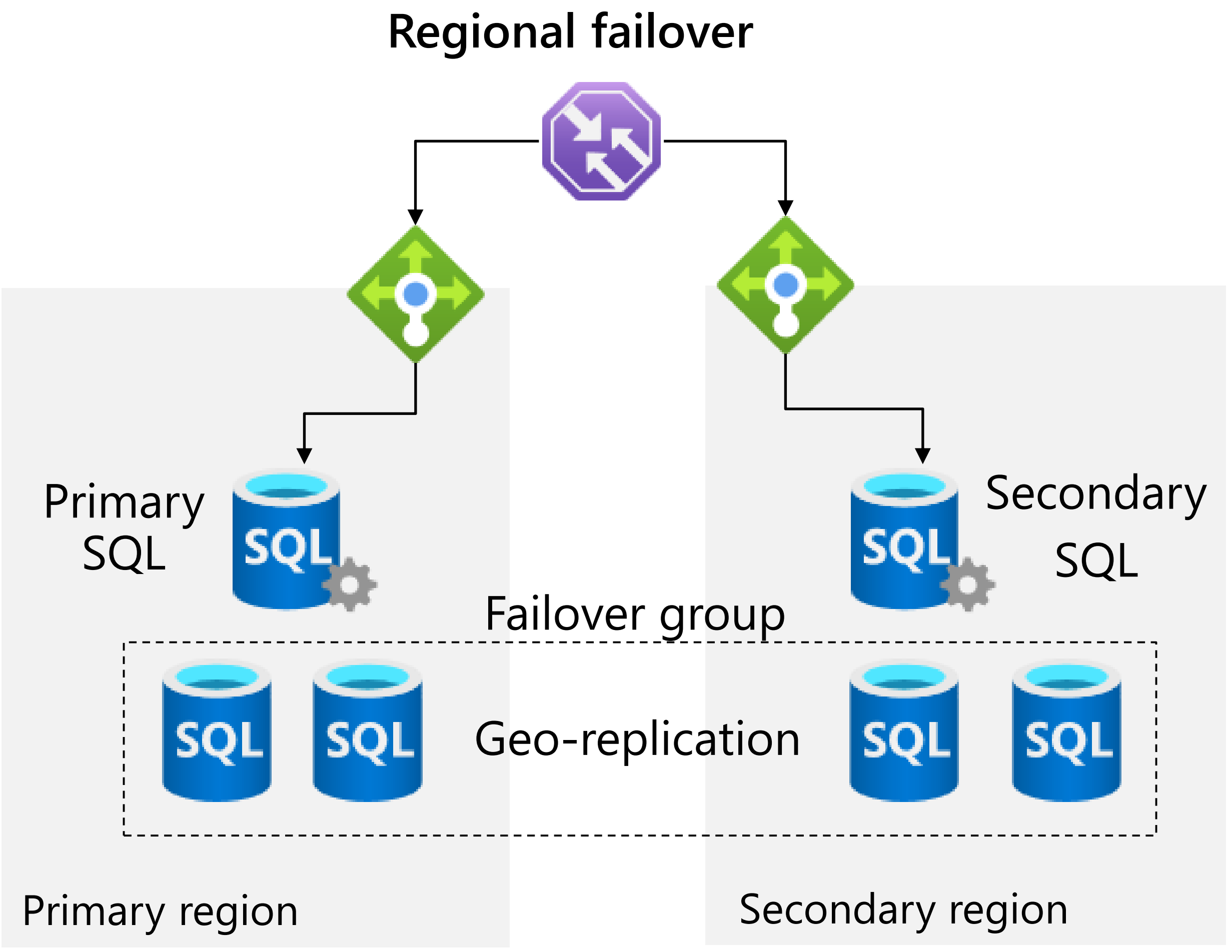
The image illustrates the availability architecture for the vCore Hyperscale tier:

- The Hyperscale tier architecture uses paired page servers. You can scale horizontally to put all the data in caching layers.
- The Hyperscale architecture supports databases as large as 100 TB.
- This tier uses snapshots, which allow for nearly instantaneous database backups, regardless of database size.
- Database restores take minutes rather than hours or days.
- You can scale up or down in constant time to accommodate your workloads.

Select a database failover strategy

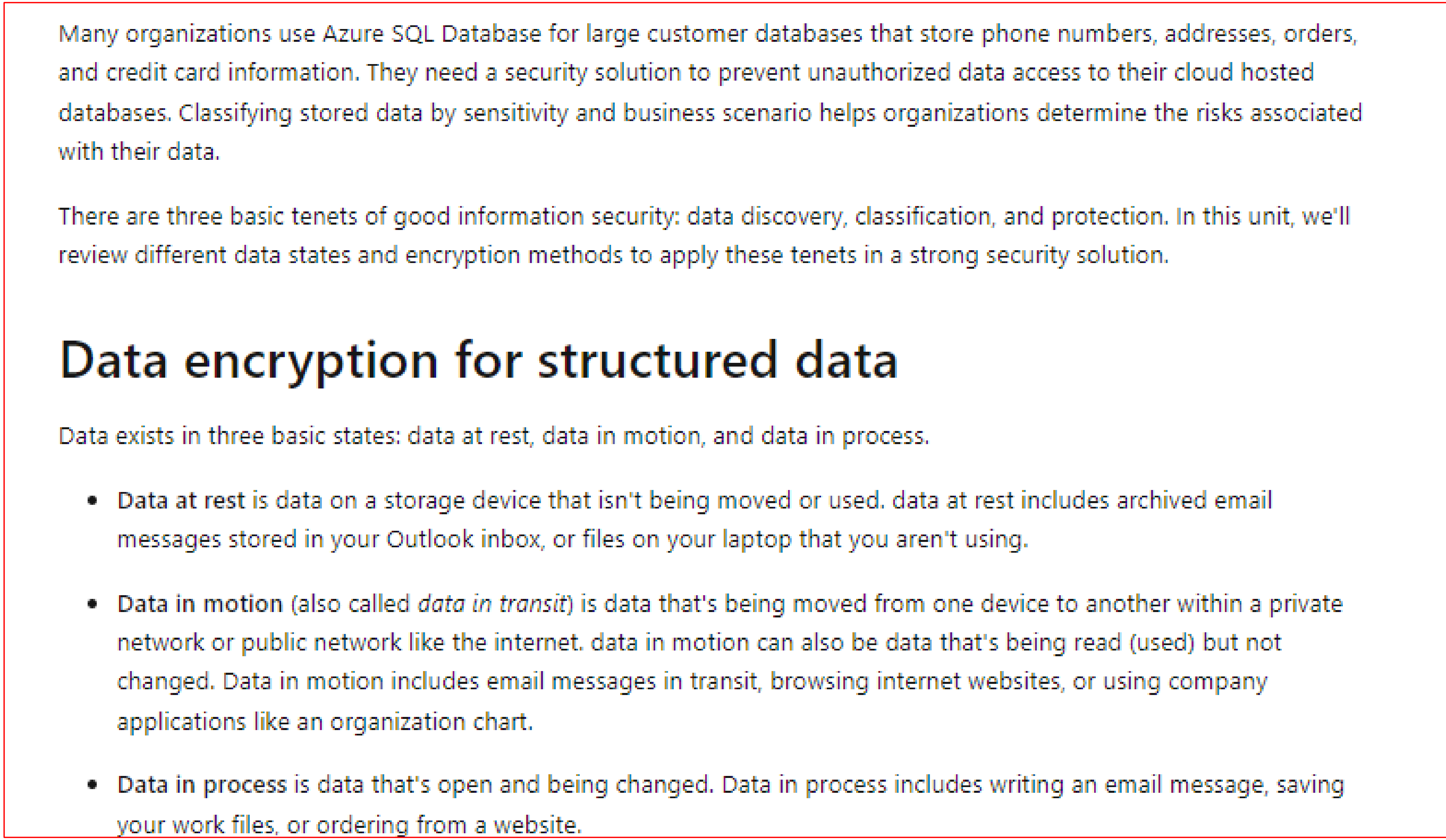
Consider datacenter and regional failover.

- In the same region -use AlwaysOn availability group with failover to secondary replicas
- Across regions – use geo-replication and failover groups



Design security for data at rest, data in transit, and data in use

Design security for data at rest, data in motion, and data in use



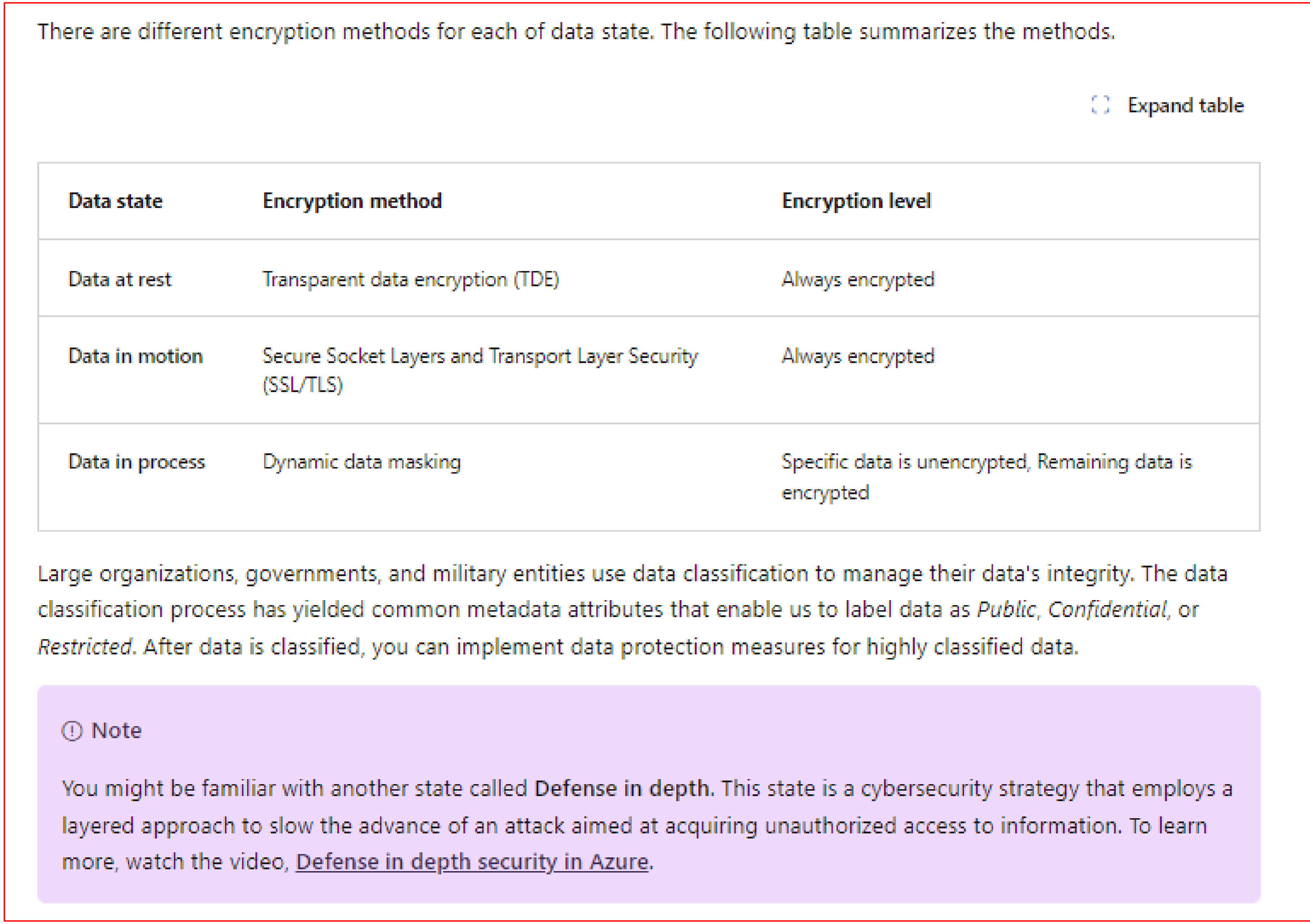
Many organizations use Azure SQL Database for large customer databases that store phone numbers, addresses, orders, and credit card information. They need a security solution to prevent unauthorized data access to their cloud hosted databases. Classifying stored data by sensitivity and business scenario helps organizations determine the risks associated with their data.

There are three basic tenets of good information security: data discovery, classification, and protection. In this unit, we'll review different data states and encryption methods to apply these tenets in a strong security solution.

Data encryption for structured data

Data exists in three basic states: data at rest, data in motion, and data in process.

- **Data at rest** is data on a storage device that isn't being moved or used. data at rest includes archived email messages stored in your Outlook inbox, or files on your laptop that you aren't using.
- **Data in motion** (also called *data in transit*) is data that's being moved from one device to another within a private network or public network like the internet. data in motion can also be data that's being read (used) but not changed. Data in motion includes email messages in transit, browsing internet websites, or using company applications like an organization chart.
- **Data in process** is data that's open and being changed. Data in process includes writing an email message, saving your work files, or ordering from a website.



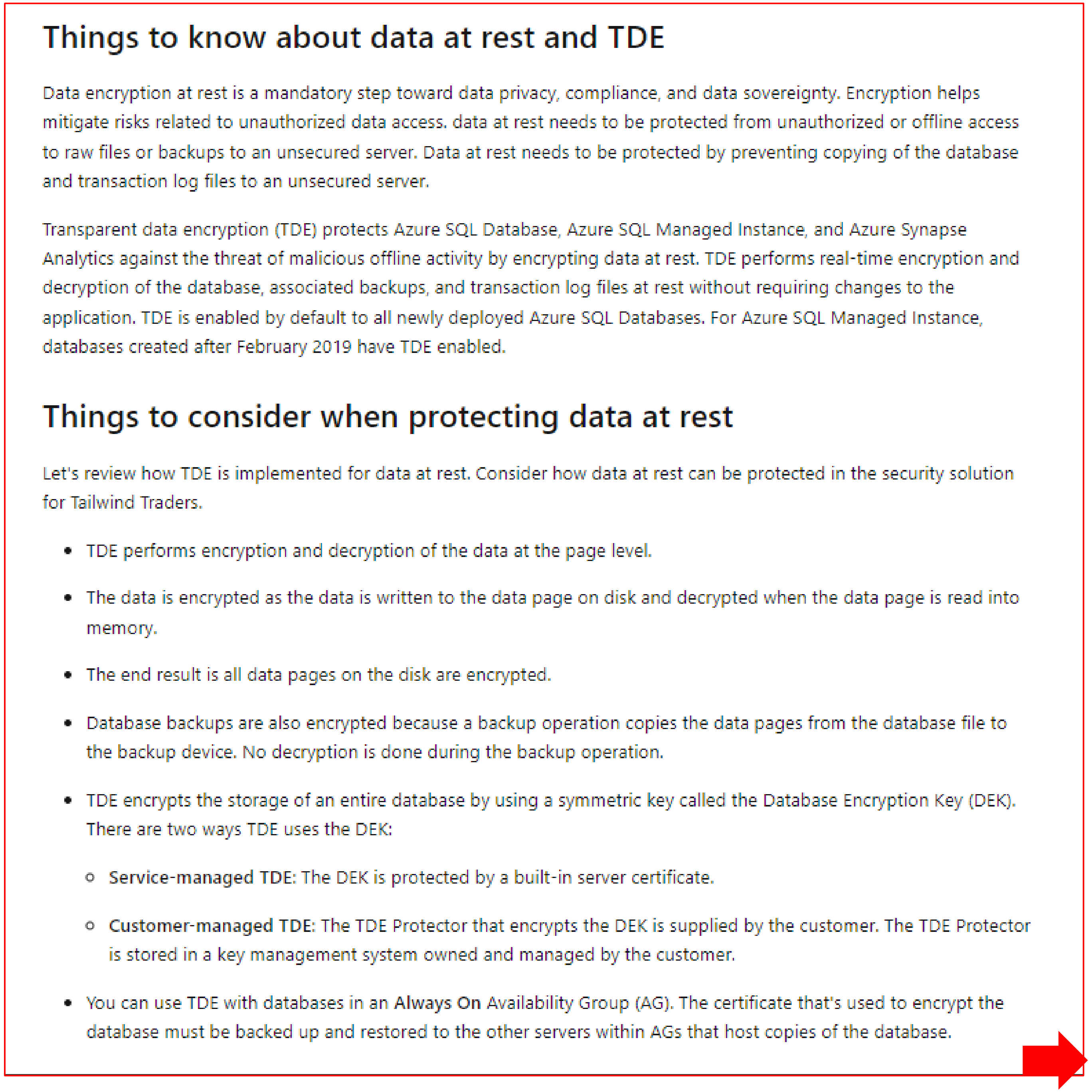
There are different encryption methods for each of data state. The following table summarizes the methods.

Data state	Encryption method	Encryption level
Data at rest	Transparent data encryption (TDE)	Always encrypted
Data in motion	Secure Socket Layers and Transport Layer Security (SSL/TLS)	Always encrypted
Data in process	Dynamic data masking	Specific data is unencrypted, Remaining data is encrypted

Large organizations, governments, and military entities use data classification to manage their data's integrity. The data classification process has yielded common metadata attributes that enable us to label data as *Public*, *Confidential*, or *Restricted*. After data is classified, you can implement data protection measures for highly classified data.

ⓘ Note

You might be familiar with another state called **Defense in depth**. This state is a cybersecurity strategy that employs a layered approach to slow the advance of an attack aimed at acquiring unauthorized access to information. To learn more, watch the video, Defense in depth security in Azure.



Things to know about data at rest and TDE

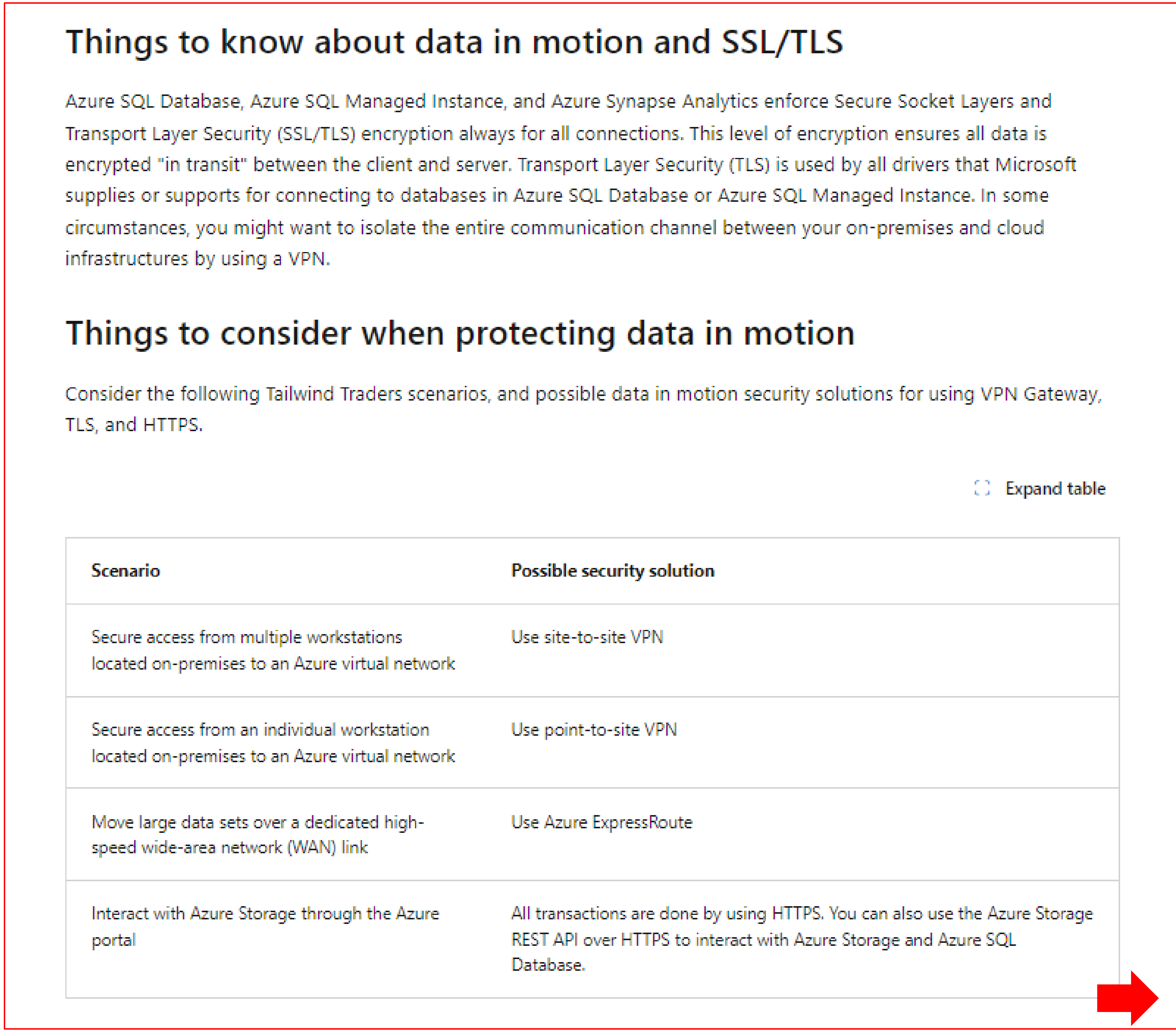
Data encryption at rest is a mandatory step toward data privacy, compliance, and data sovereignty. Encryption helps mitigate risks related to unauthorized data access. data at rest needs to be protected from unauthorized or offline access to raw files or backups to an unsecured server. Data at rest needs to be protected by preventing copying of the database and transaction log files to an unsecured server.

Transparent data encryption (TDE) protects Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics against the threat of malicious offline activity by encrypting data at rest. TDE performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application. TDE is enabled by default to all newly deployed Azure SQL Databases. For Azure SQL Managed Instance, databases created after February 2019 have TDE enabled.

Things to consider when protecting data at rest

Let's review how TDE is implemented for data at rest. Consider how data at rest can be protected in the security solution for Tailwind Traders.

- TDE performs encryption and decryption of the data at the page level.
- The data is encrypted as the data is written to the data page on disk and decrypted when the data page is read into memory.
- The end result is all data pages on the disk are encrypted.
- Database backups are also encrypted because a backup operation copies the data pages from the database file to the backup device. No decryption is done during the backup operation.
- TDE encrypts the storage of an entire database by using a symmetric key called the Database Encryption Key (DEK). There are two ways TDE uses the DEK:
  - **Service-managed TDE**: The DEK is protected by a built-in server certificate.
  - **Customer-managed TDE**: The TDE Protector that encrypts the DEK is supplied by the customer. The TDE Protector is stored in a key management system owned and managed by the customer.
- You can use TDE with databases in an **Always On** Availability Group (AG). The certificate that's used to encrypt the database must be backed up and restored to the other servers within AGs that host copies of the database.



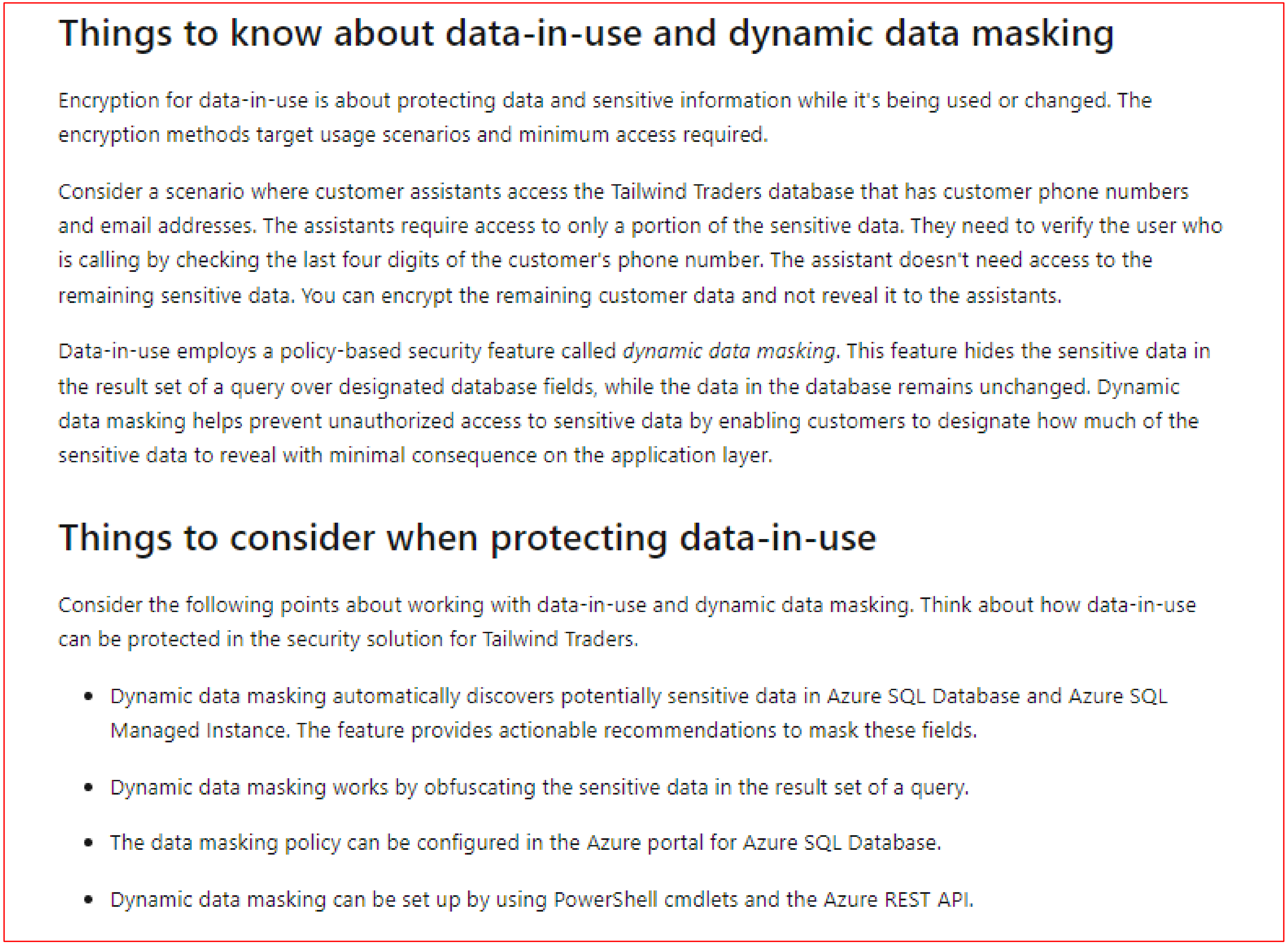
Things to know about data in motion and SSL/TLS

Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics enforce Secure Socket Layers and Transport Layer Security (SSL/TLS) encryption always for all connections. This level of encryption ensures all data is encrypted "in transit" between the client and server. Transport Layer Security (TLS) is used by all drivers that Microsoft supplies or supports for connecting to databases in Azure SQL Database or Azure SQL Managed Instance. In some circumstances, you might want to isolate the entire communication channel between your on-premises and cloud infrastructures by using a VPN.

Things to consider when protecting data in motion

Consider the following Tailwind Traders scenarios, and possible data in motion security solutions for using VPN Gateway, TLS, and HTTPS.

Scenario	Possible security solution
Secure access from multiple workstations located on-premises to an Azure virtual network	Use site-to-site VPN
Secure access from an individual workstation located on-premises to an Azure virtual network	Use point-to-site VPN
Move large data sets over a dedicated high-speed wide-area network (WAN) link	Use Azure ExpressRoute
Interact with Azure Storage through the Azure portal	All transactions are done by using HTTPS. You can also use the Azure Storage REST API over HTTPS to interact with Azure Storage and Azure SQL Database.



Things to know about data-in-use and dynamic data masking

Encryption for data-in-use is about protecting data and sensitive information while it's being used or changed. The encryption methods target usage scenarios and minimum access required.

Consider a scenario where customer assistants access the Tailwind Traders database that has customer phone numbers and email addresses. The assistants require access to only a portion of the sensitive data. They need to verify the user who is calling by checking the last four digits of the customer's phone number. The assistant doesn't need access to the remaining sensitive data. You can encrypt the remaining customer data and not reveal it to the assistants.

Data-in-use employs a policy-based security feature called *dynamic data masking*. This feature hides the sensitive data in the result set of a query over designated database fields, while the data in the database remains unchanged. Dynamic data masking helps prevent unauthorized access to sensitive data by enabling customers to designate how much of the sensitive data to reveal with minimal consequence on the application layer.

Things to consider when protecting data-in-use

Consider the following points about working with data-in-use and dynamic data masking. Think about how data-in-use can be protected in the security solution for Tailwind Traders.

- Dynamic data masking automatically discovers potentially sensitive data in Azure SQL Database and Azure SQL Managed Instance. The feature provides actionable recommendations to mask these fields.

- Dynamic data masking works by obfuscating the sensitive data in the result set of a query.

- The data masking policy can be configured in the Azure portal for Azure SQL Database.

- Dynamic data masking can be set up by using PowerShell cmdlets and the Azure REST API.

Protect your database

Use a layered (defense in depth) approach to data protection.

Network security

- VNet
- Firewall rules, NSG
- Private link

Identity and access

- Authentication options: Azure AD, SQL Auth, Windows Auth
- Azure RBAC
- Roles and permissions
- Row level security

Data protection

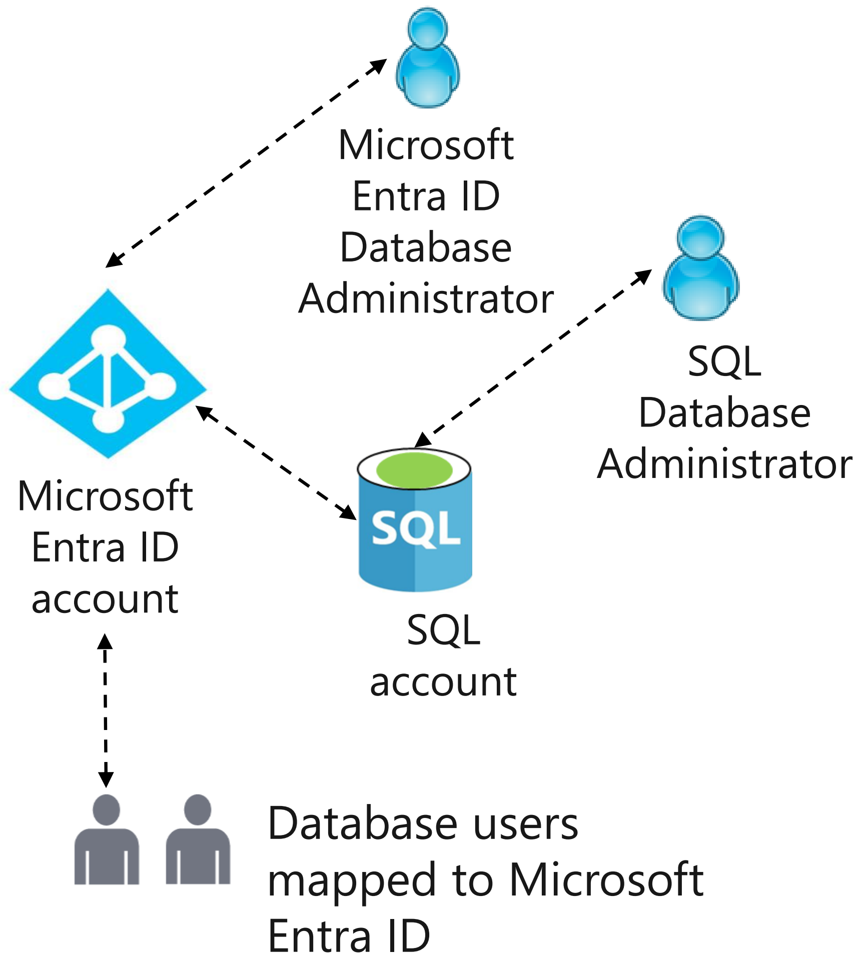
- Encryption-in-use (Always encrypted)
- Encryption-at-rest (TDE)
- Encryption-in-flight (TLS)
- Customer-managed keys
- Dynamic data masking

Security management

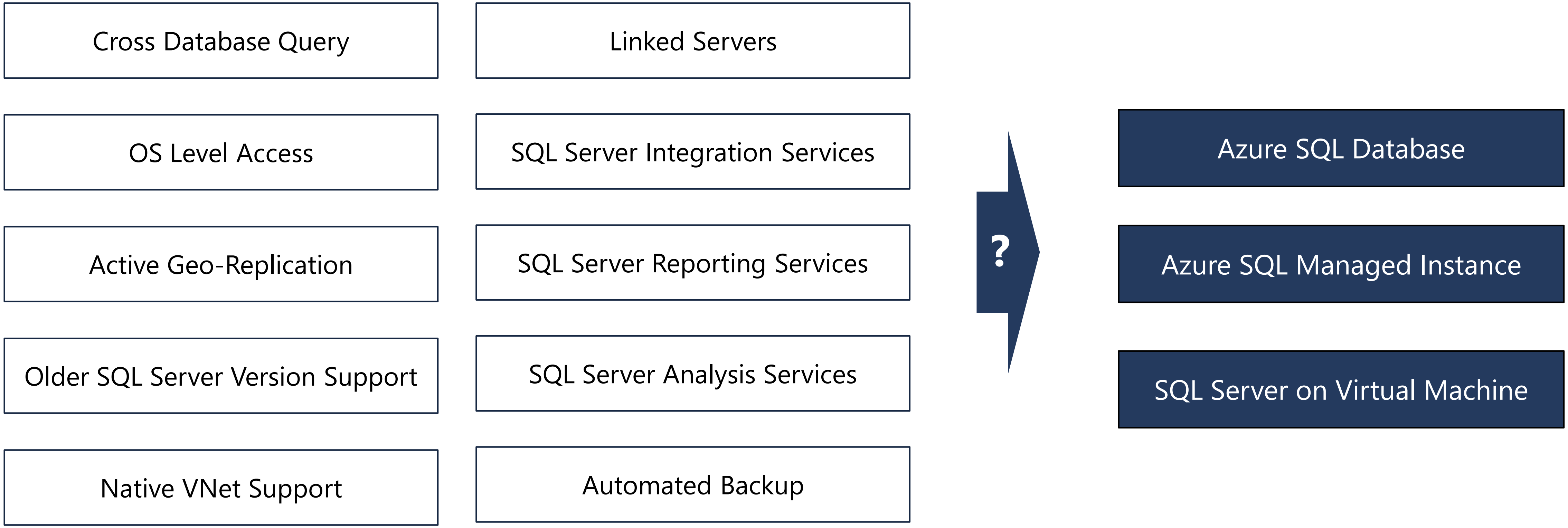
- Advanced threat protection
- SQL audit
- Audit integration with log analytics and event hubs
- Vulnerability assessment
- Data discovery and classification
- Microsoft Defender for Cloud

Authenticate to an Azure SQL database

- SQL database supports two types of authentication - SQL authentication and Microsoft Entra authentication
- SQL server authentication credentials are stored in the database
- Microsoft Entra authentication credentials are stored in Microsoft Entra ID


Microsoft Entra ID Database Administrator
SQL Database Administrator
Microsoft Entra ID account
SQL
SQL account
Database users mapped to Microsoft Entra ID

Select the appropriate features (activity)



Cross Database Query

OS Level Access

Active Geo-Replication

Older SQL Server Version Support

Native VNet Support

Linked Servers

SQL Server Integration Services

SQL Server Reporting Services

SQL Server Analysis Services

Automated Backup

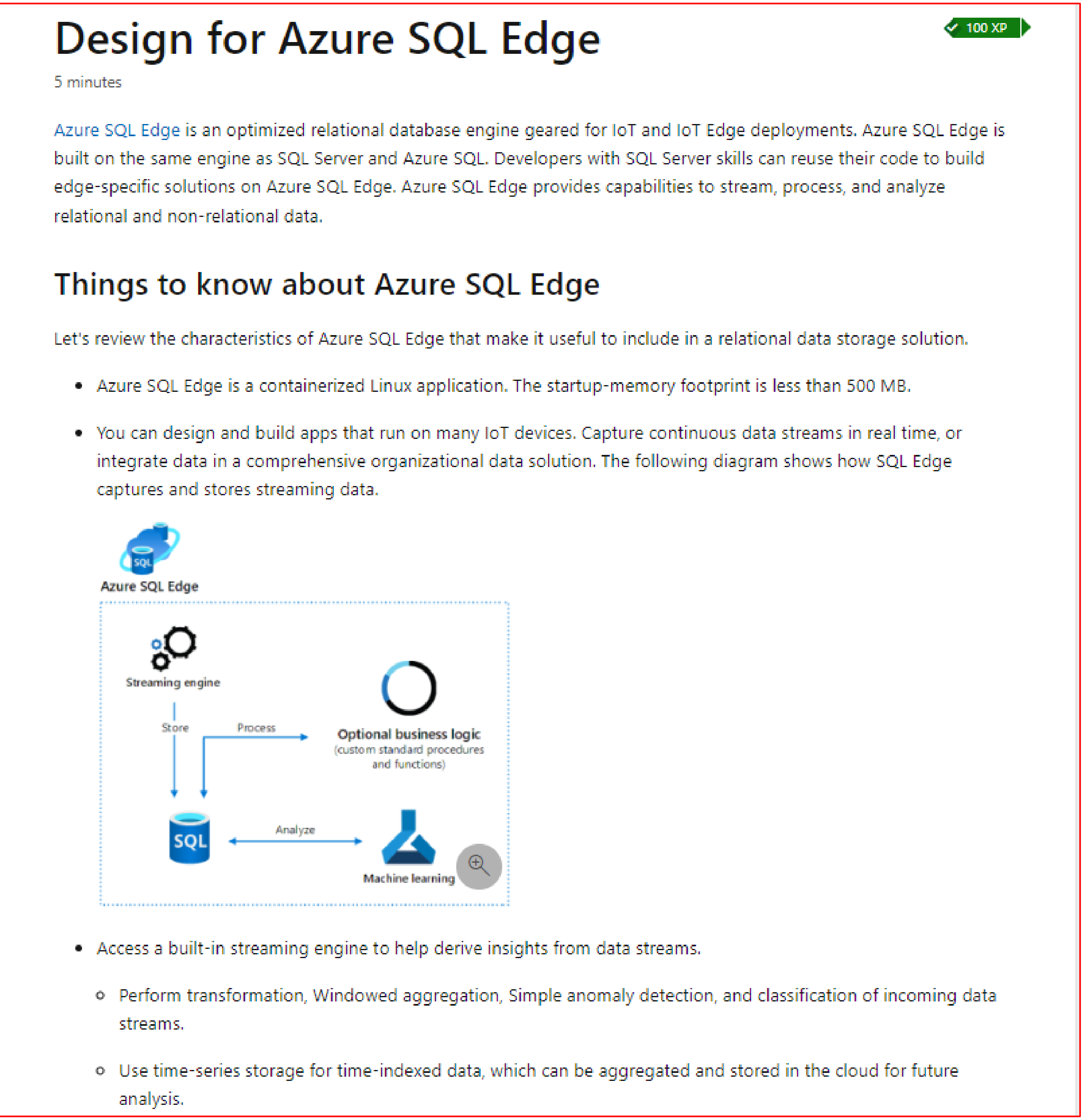
?

Azure SQL Database

Azure SQL Managed Instance

SQL Server on Virtual Machine

Design for Azure SQL Edge



Design for Azure SQL Edge

✓ 100 XP

5 minutes

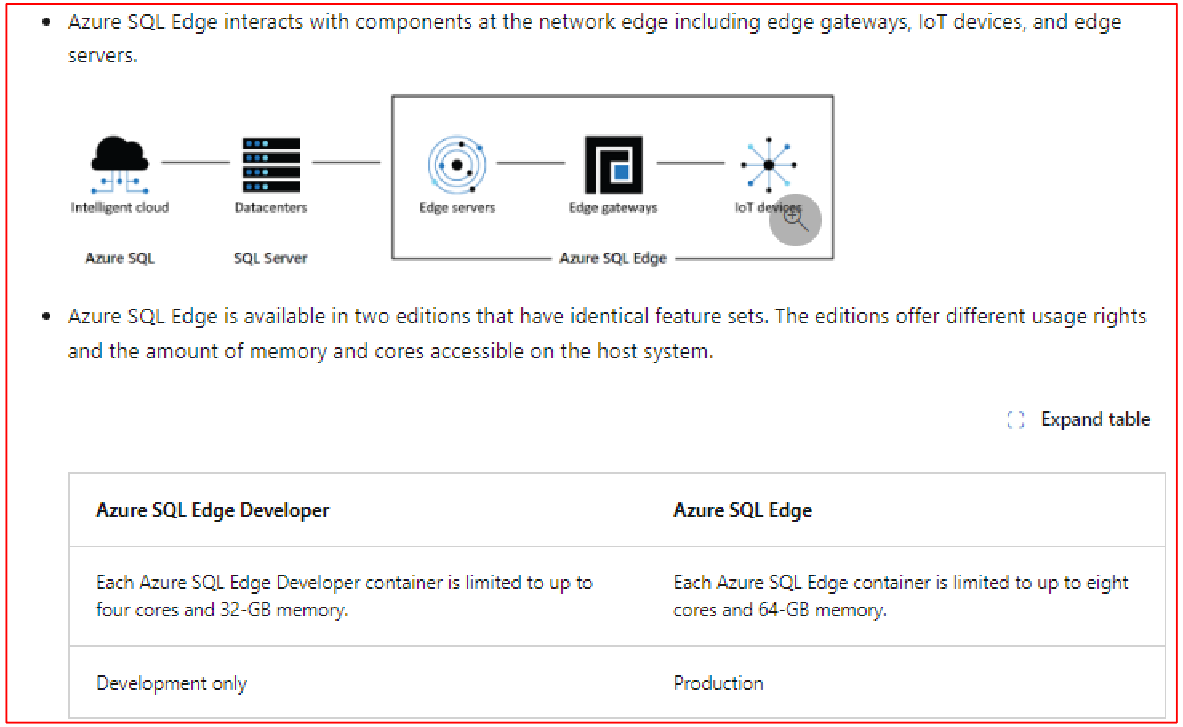
Azure SQL Edge is an optimized relational database engine geared for IoT and IoT Edge deployments. Azure SQL Edge is built on the same engine as SQL Server and Azure SQL. Developers with SQL Server skills can reuse their code to build edge-specific solutions on Azure SQL Edge. Azure SQL Edge provides capabilities to stream, process, and analyze relational and non-relational data.

Things to know about Azure SQL Edge

Let's review the characteristics of Azure SQL Edge that make it useful to include in a relational data storage solution.

- Azure SQL Edge is a containerized Linux application. The startup-memory footprint is less than 500 MB.
- You can design and build apps that run on many IoT devices. Capture continuous data streams in real time, or integrate data in a comprehensive organizational data solution. The following diagram shows how SQL Edge captures and stores streaming data.

- Access a built-in streaming engine to help derive insights from data streams.
  - Perform transformation, Windowed aggregation, Simple anomaly detection, and classification of incoming data streams.
  - Use time-series storage for time-indexed data, which can be aggregated and stored in the cloud for future analysis.

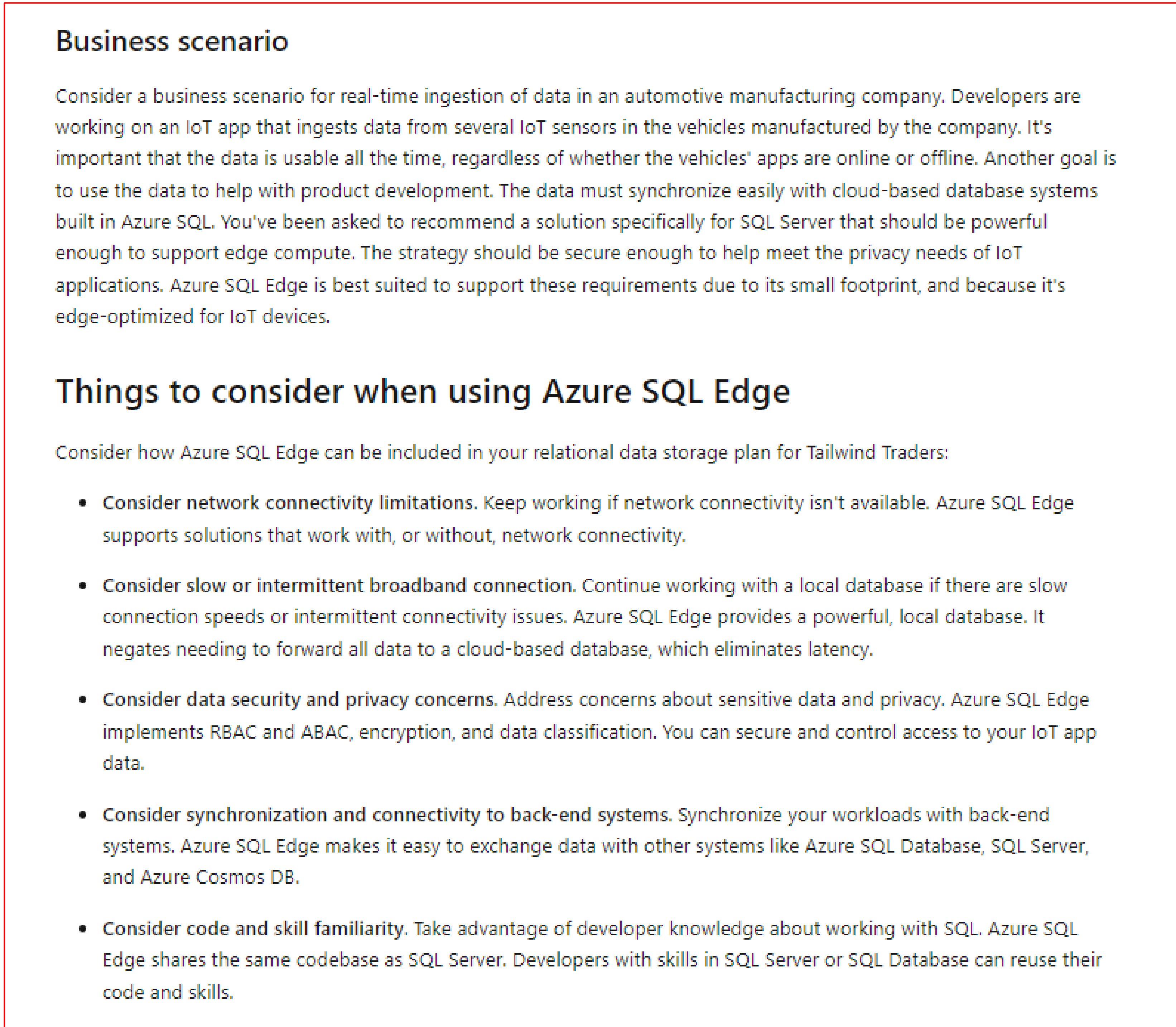


- Azure SQL Edge interacts with components at the network edge including edge gateways, IoT devices, and edge servers.

- Azure SQL Edge is available in two editions that have identical feature sets. The editions offer different usage rights and the amount of memory and cores accessible on the host system.

Expand table

Azure SQL Edge Developer	Azure SQL Edge
Each Azure SQL Edge Developer container is limited to up to four cores and 32-GB memory.	Each Azure SQL Edge container is limited to up to eight cores and 64-GB memory.
Development only	Production



Business scenario

Consider a business scenario for real-time ingestion of data in an automotive manufacturing company. Developers are working on an IoT app that ingests data from several IoT sensors in the vehicles manufactured by the company. It's important that the data is usable all the time, regardless of whether the vehicles' apps are online or offline. Another goal is to use the data to help with product development. The data must synchronize easily with cloud-based database systems built in Azure SQL. You've been asked to recommend a solution specifically for SQL Server that should be powerful enough to support edge compute. The strategy should be secure enough to help meet the privacy needs of IoT applications. Azure SQL Edge is best suited to support these requirements due to its small footprint, and because it's edge-optimized for IoT devices.

Things to consider when using Azure SQL Edge

Consider how Azure SQL Edge can be included in your relational data storage plan for Tailwind Traders:

- **Consider network connectivity limitations.** Keep working if network connectivity isn't available. Azure SQL Edge supports solutions that work with, or without, network connectivity.

- **Consider slow or intermittent broadband connection.** Continue working with a local database if there are slow connection speeds or intermittent connectivity issues. Azure SQL Edge provides a powerful, local database. It negates needing to forward all data to a cloud-based database, which eliminates latency.

- **Consider data security and privacy concerns.** Address concerns about sensitive data and privacy. Azure SQL Edge implements RBAC and ABAC, encryption, and data classification. You can secure and control access to your IoT app data.

- **Consider synchronization and connectivity to back-end systems.** Synchronize your workloads with back-end systems. Azure SQL Edge makes it easy to exchange data with other systems like Azure SQL Database, SQL Server, and Azure Cosmos DB.

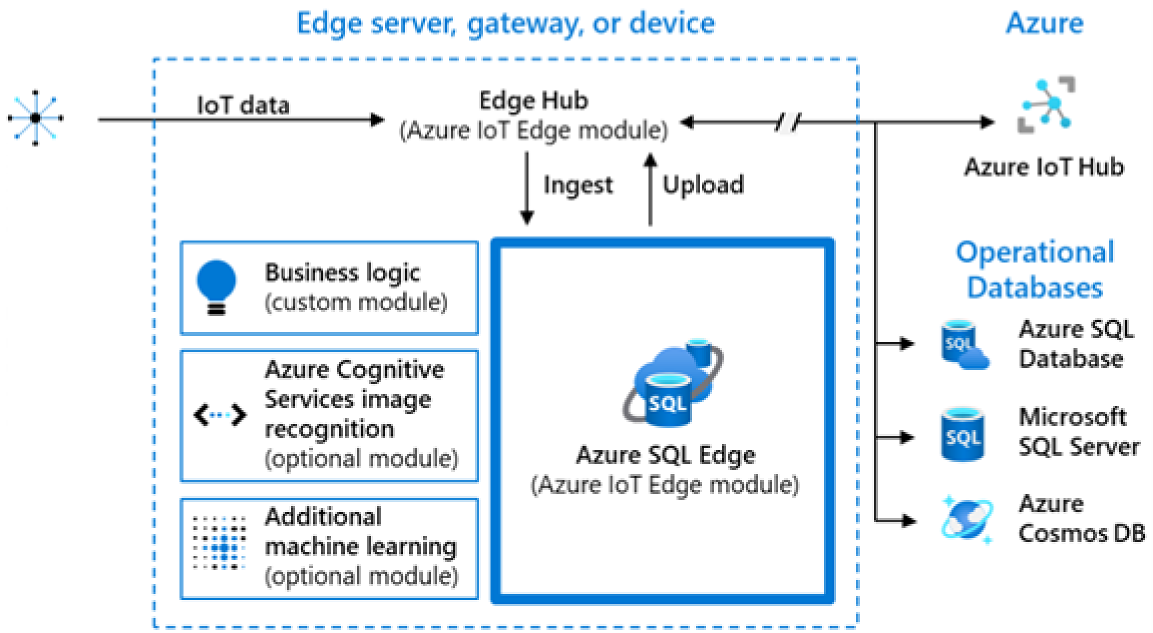
- **Consider code and skill familiarity.** Take advantage of developer knowledge about working with SQL. Azure SQL Edge shares the same codebase as SQL Server. Developers with skills in SQL Server or SQL Database can reuse their code and skills.

When to use Azure SQL Edge

An optimized relational database engine geared for IoT and IoT Edge deployments. It is a containerized Linux application that runs on a process that's based on ARM64 or x64.

Use SQL Edge when you need to:

- Capture continuous data streams in real time
- Integrate the data in a comprehensive organizational data solution
- Synchronization and connectivity to back-end systems
- Overcome slow or intermittent broadband connections

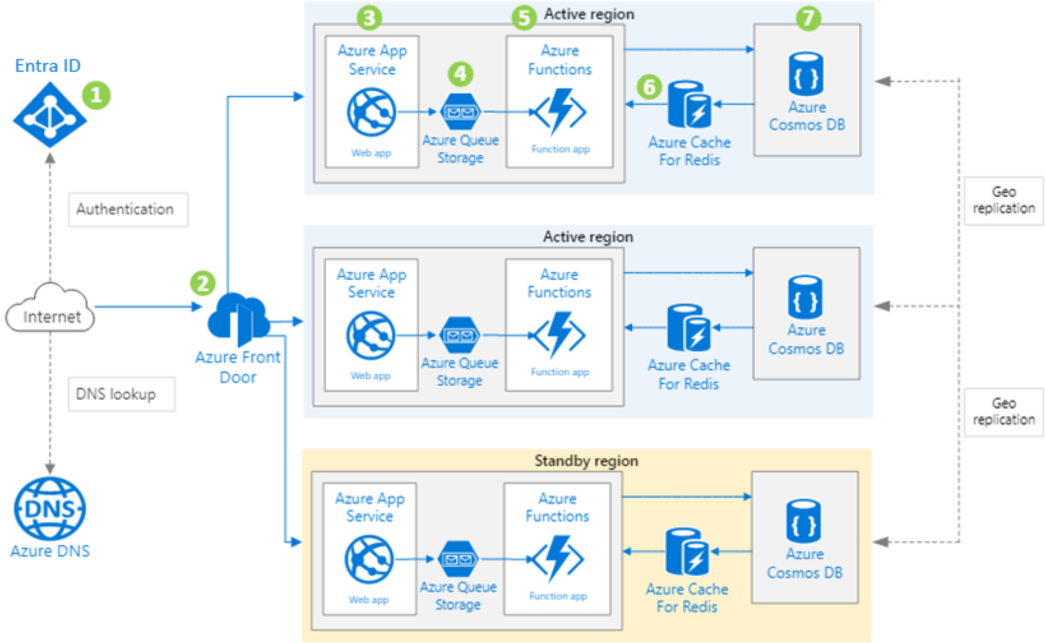


Design for Azure Cosmos DB

When to use Azure Cosmos DB

A fully managed NoSQL database service for modern app development. It has single-digit millisecond response times and guaranteed speed at any scale.

- Web and mobile applications that store and query user generated content like Tweets or blog posts
- Retail and marketing industry that store catalog data and event sourcing in order proccing pipelines
- Gaming that requires single-millisecond latencies for reads and writes and can handle massive spikes in request rates during new game launches or feature updates.
- IoT use cases can load data into Azure Cosmos DB for adhoc querying. New data and changes to existing data can be read on change feed. Then all data or just changes to data in Azure Cosmos DB can be used as reference data as part of real-time analytics.



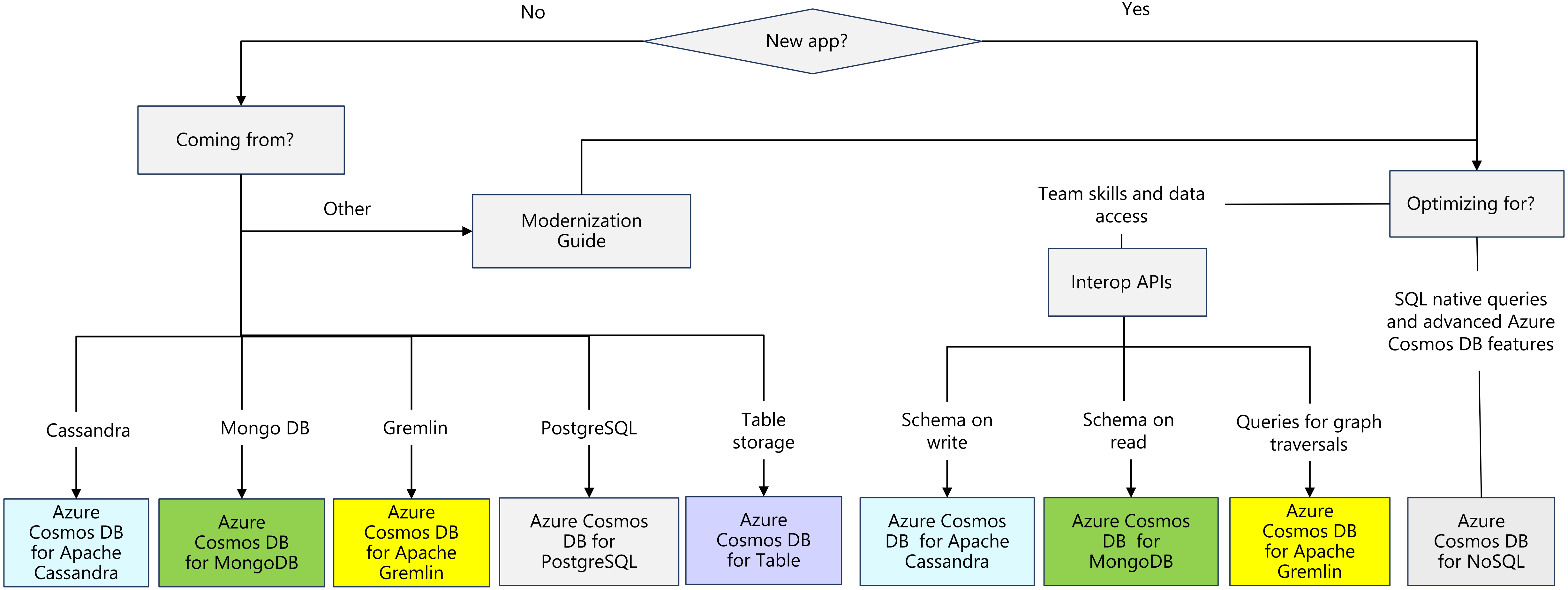
Azure Storage tables and Azure Cosmos DB tables

- **Azure Table storage** is a service that stores non-relational structured data (also known as structured NoSQL data) in the cloud, providing a key/attribute store with a schemeless design.
- **Azure Cosmos DB for Table** provides the Table API for applications that are written for Azure Table storage and that need premium capabilities like high availability, scalability, and dedicated throughput.

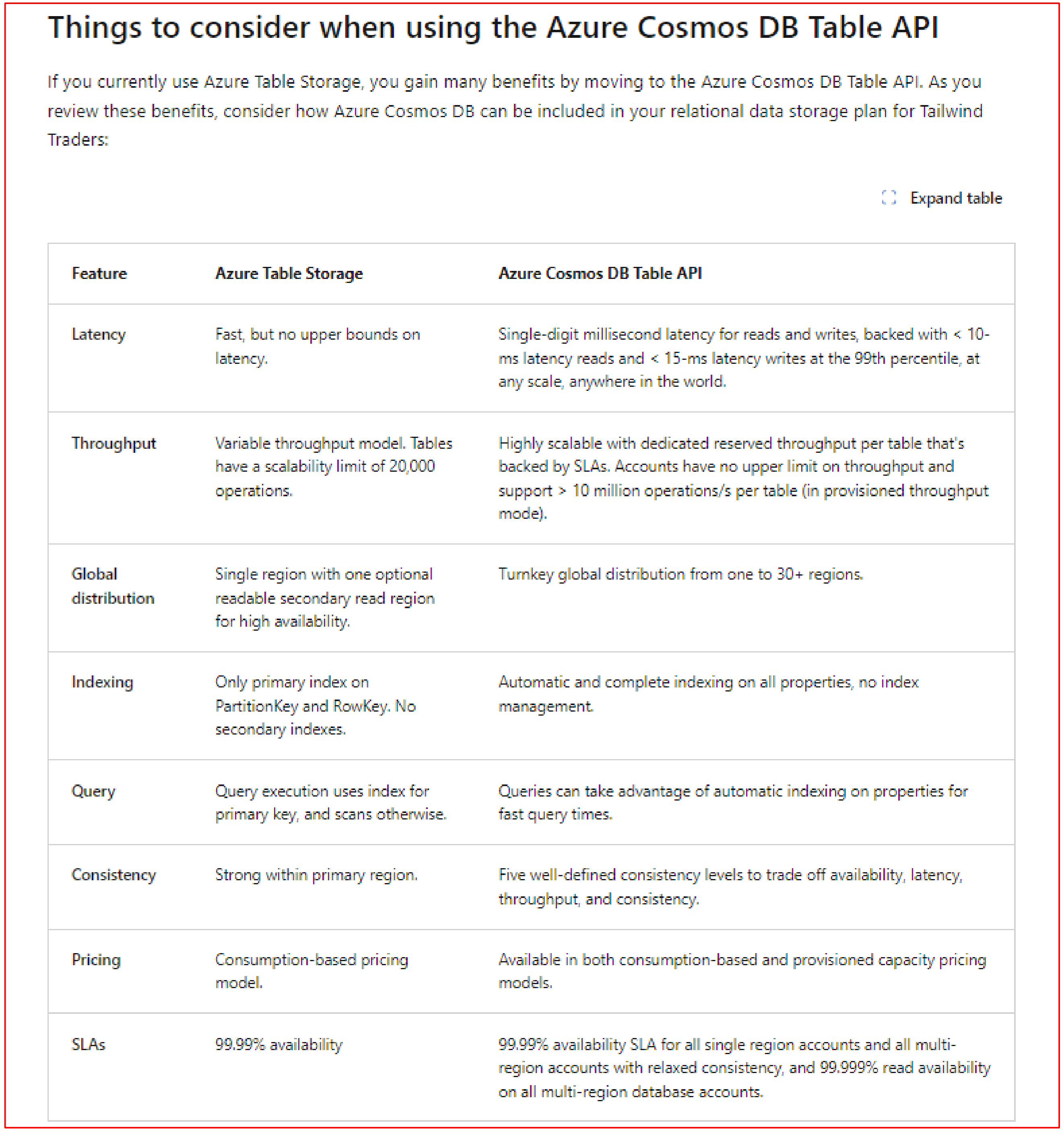
Differences in behavior

- You are charged for the capacity of an Azure Cosmos DB table as soon as it is created, even if that capacity isn't used.
- Query results from Azure Cosmos DB are not sorted in order of partition key and row key as they are from Storage tables.
- Row keys in Azure Cosmos DB are limited to 255 bytes.
- Table names are case-sensitive in Azure Cosmos DB. They are not case-sensitive in Storage tables.

What Azure Cosmos DB APIs are supported?



No
New app?
Yes
Coming from?
Other
Modernization Guide
Team skills and data access
Optimizing for?
Interop APIs
SQL native queries and advanced Azure Cosmos DB features
Cassandra
Mongo DB
Gremlin
PostgreSQL
Table storage
Schema on write
Schema on read
Queries for graph traversals
Azure Cosmos DB for Apache Cassandra
Azure Cosmos DB for MongoDB
Azure Cosmos DB for Apache Gremlin
Azure Cosmos DB for PostgreSQL
Azure Cosmos DB for Table
Azure Cosmos DB for Apache Cassandra
Azure Cosmos DB for MongoDB
Azure Cosmos DB for Apache Gremlin
Azure Cosmos DB for NoSQL



Things to consider when using the Azure Cosmos DB Table API

If you currently use Azure Table Storage, you gain many benefits by moving to the Azure Cosmos DB Table API. As you review these benefits, consider how Azure Cosmos DB can be included in your relational data storage plan for Tailwind Traders:

Feature	Azure Table Storage	Azure Cosmos DB Table API
Latency	Fast, but no upper bounds on latency.	Single-digit millisecond latency for reads and writes, backed with < 10-ms latency reads and < 15-ms latency writes at the 99th percentile, at any scale, anywhere in the world.
Throughput	Variable throughput model. Tables have a scalability limit of 20,000 operations.	Highly scalable with dedicated reserved throughput per table that's backed by SLAs. Accounts have no upper limit on throughput and support > 10 million operations/s per table (in provisioned throughput mode).
Global distribution	Single region with one optional readable secondary read region for high availability.	Turnkey global distribution from one to 30+ regions.
Indexing	Only primary index on PartitionKey and RowKey. No secondary indexes.	Automatic and complete indexing on all properties, no index management.
Query	Query execution uses index for primary key, and scans otherwise.	Queries can take advantage of automatic indexing on properties for fast query times.
Consistency	Strong within primary region.	Five well-defined consistency levels to trade off availability, latency, throughput, and consistency.
Pricing	Consumption-based pricing model.	Available in both consumption-based and provisioned capacity pricing models.
SLAs	99.99% availability	99.99% availability SLA for all single region accounts and all multi-region accounts with relaxed consistency, and 99.999% read availability on all multi-region database accounts.

Case study and review

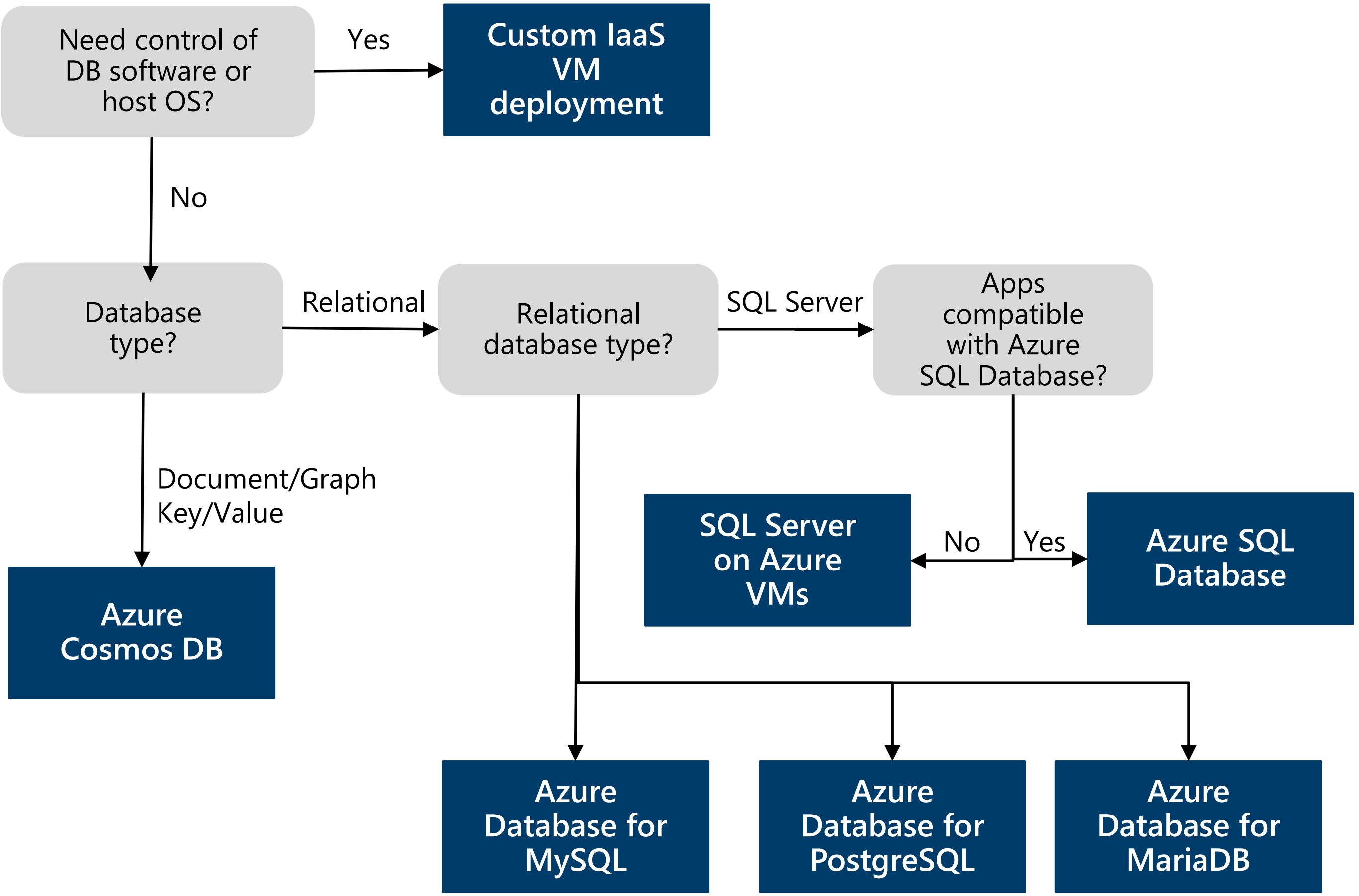
Select a structured data product (activity)

You need a globally distributed, multi-model database with support for NoSQL choices.

You need a fully managed, scalable MySQL relational database that has high availability and security built in at no extra cost.

You need a fully managed relational database that provisions quickly, scales on the fly, and includes built-in intelligence and security.

You need to host enterprise SQL Server applications in the cloud and have full control over the server OS.

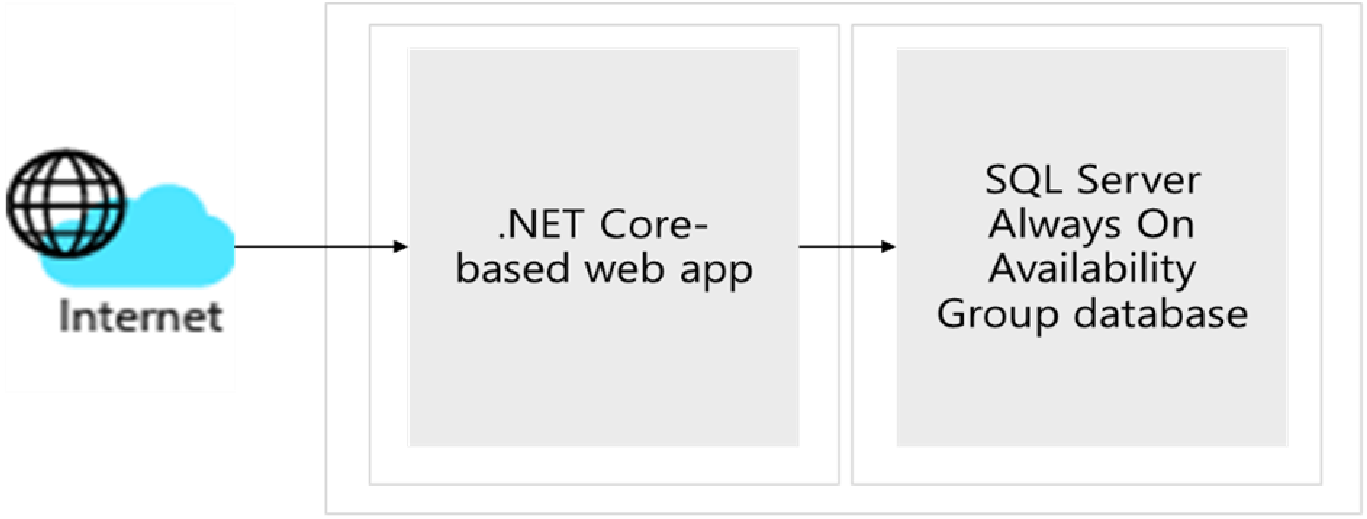

Need control of DB software or host OS?
Yes
Custom IaaS VM deployment
No
Database type?
Relational
Relational database type?
SQL Server
Apps compatible with Azure SQL Database?
Document/Graph Key/Value
Azure Cosmos DB
SQL Server on Azure VMs
No
Yes
Azure SQL Database
Azure Database for MySQL
Azure Database for PostgreSQL
Azure Database for MariaDB

Case Study – Relational data

- Design a database solution.

- Your design should include authorization, authentication, pricing, performance, and high availability.

- Diagram what you decide and explain your solution.

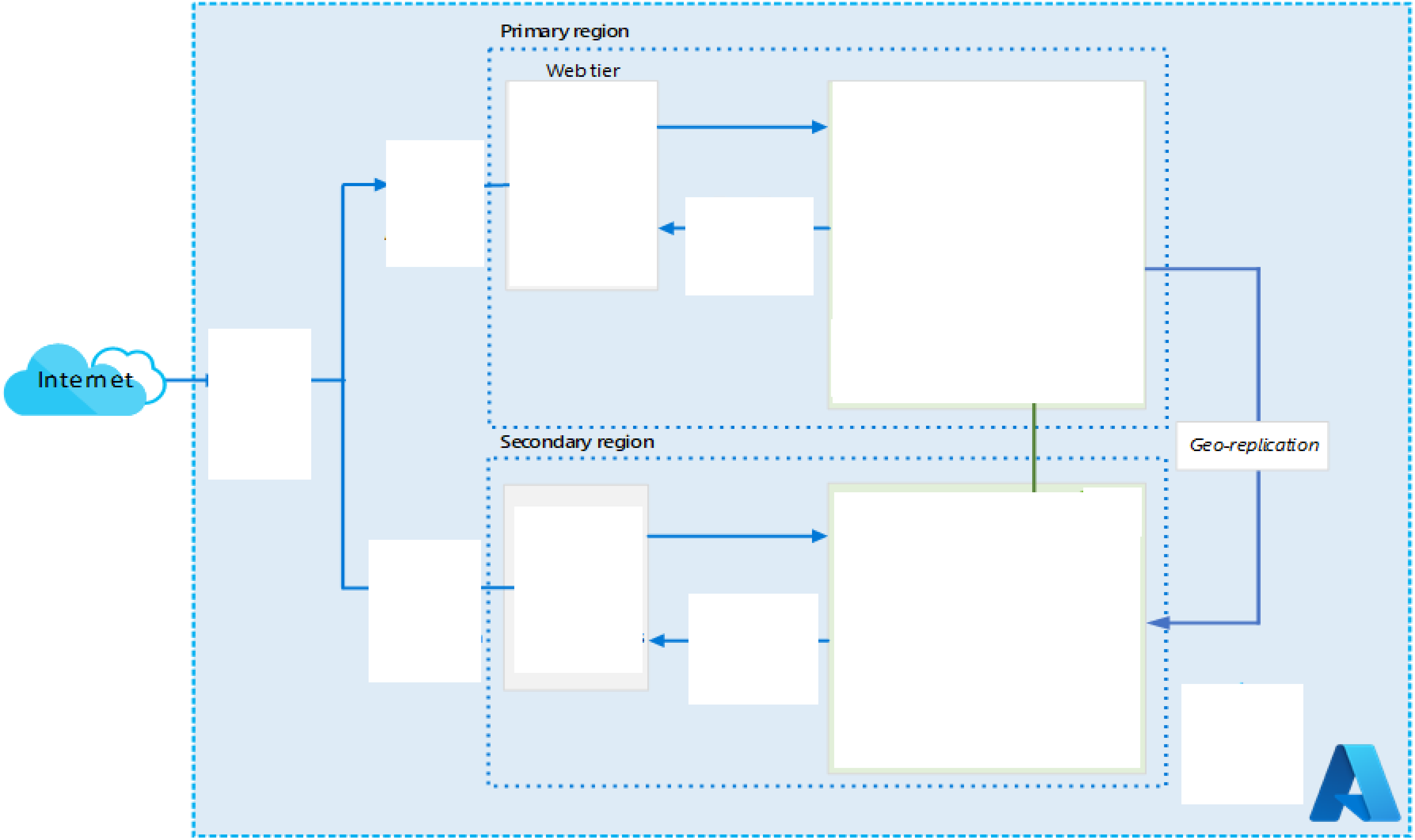


Internet

.NET Core-based web app

SQL Server Always On Availability Group database

- 2-tier Windows based .NET Core-based web app
- Provides access to the product catalog hosted in a SQL Server
- Categorized as mission-critical and requires high availability provisions



VNet

NSG

App Services

Cosmos DB

Microsoft Entra ID

Cache Redis

SQL DB

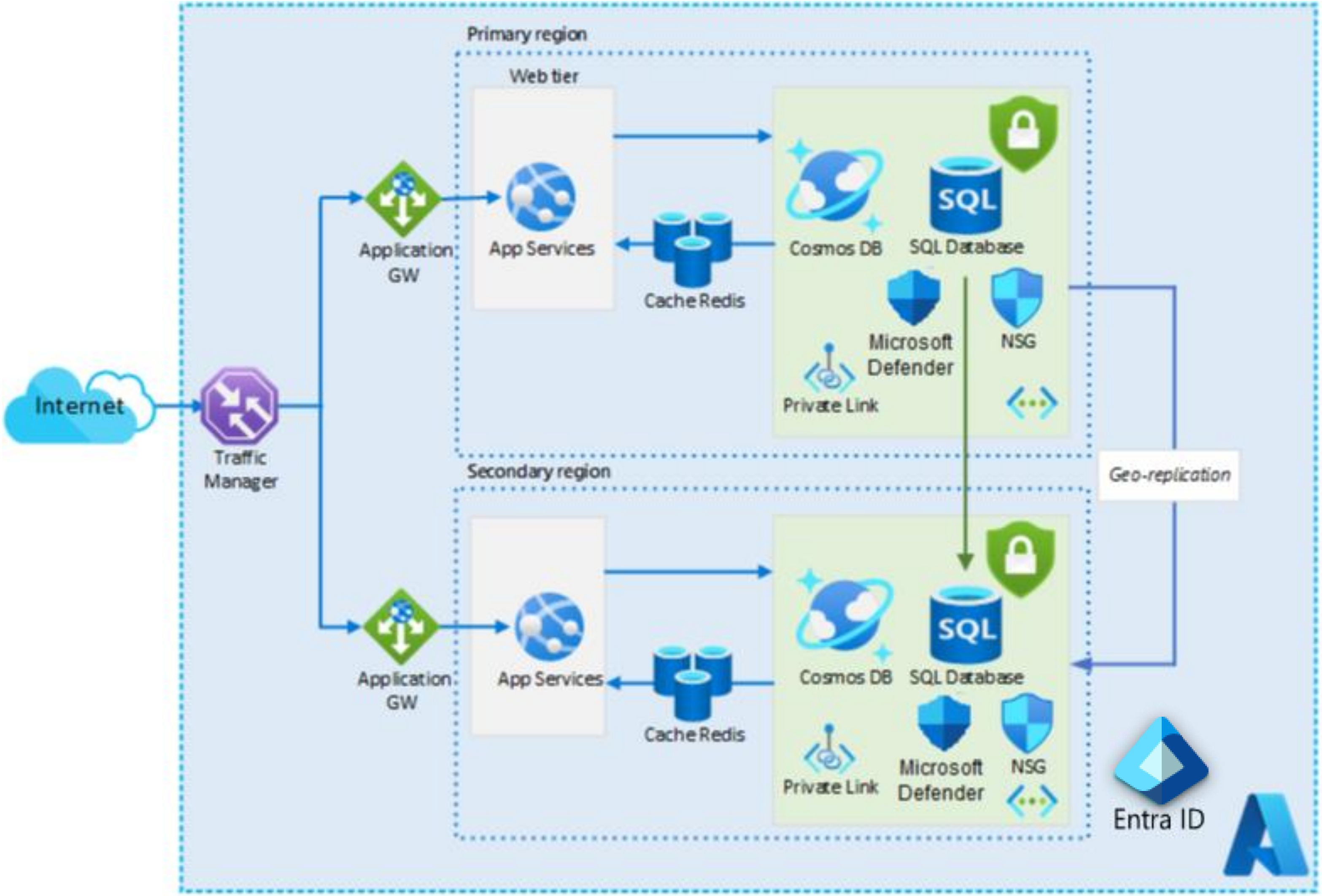
Private Link

Microsoft Defender

App GW

Traffic Manager

Solution Diagram (completed)


Primary region
Web tier
Internet
Traffic Manager
Application GW
App Services
Cache Redis
Cosmos DB
SQL
SQL Database
Microsoft Defender
NSG
Private Link
Geo-replication
Secondary region
Application GW
App Services
Cache Redis
Cosmos DB
SQL
SQL Database
Private Link
Microsoft Defender
NSG
Entra ID