Azure Database Administration Documentation Exam (DP-300) V1
Created by Mustafa Elmasry
https://mostafaelmasry.com/
https://www.linkedin.com/in/mostafaelmasry/
Eng.Mostafa_Elmasry@WindowsLive.Com
+966 543990968

Azure Database Administration Documentation Exam (DP-300) V1

DP-300 Document Contents V1

Author Information 3
Azure SQL Resources 4
SQL Server on Azure VM 5
MariaDB, MySQL, and PostgreSQL on Azure 11
Azure SQL Database Architecture 13
Azure SQL Database Connectivity 14
Introduction to Azure SQL 15
Azure SQL Database Types 24
Scaling out Azure SQL Database 26
Azure SQL Database 26
- Service Tier and Purchase Model 28
Azure SQL Database Elastic Pool 31
- Azure SQL Database Elastic Pool Compute + Storage 32
Azure SQL Database MI Managed Instance 33
- Differences between SQL Server On-Premises and Azure SQL Managed instance 37
- Azure Managed instance Service Tier 37
- Azure SQL Managed instance Security 38
- Azure SQL MI References 39
Differences between Azure SQL Database and SQL Server and Non-Supported Features 40

Azure Database Migration 42
Azure SQL backup 45
Azure SQL Restore 48
Azure SQL Security 49
SQL Azure HA High Availability 54
Azure SQL Performance 62
- Query performance insight 68
- Azure SQL Analytics 69
- Azure SQL Performance Recommendation Services 71
- Azure SQL Automatic Tuning Services 73

Azure SQL Elastic Database Jobs 74
Azure SQL Database Data Sync 85
Azure DB-300 Exam QA 90

Author Information

Mustafa Elmasry is a Microsoft database consultant working as a Database administrator for more than +10 Years I have very good knowledge about Database Migration, Consolidation, Performance Tuning, Automation Using T-SQL and so many other tasks I do it with multiple customers here in KSA, I am Microsoft SQL Server Certified Development and administration 2008 and 2016 (**2x MCTS, 2x MCTIP, MCSA, MCSE**), Microsoft Certified Trainer (**MCT**), Azure Certified **5X Azure Fundamental**, **Azure Data Fundamental**, **Azure Administrator, Azure Database administrator, Azure Data Engineer** also I was awarded by **Microsoft Azure Heroes** 3 times as (**Azure Content hero, Azure Community hero and Azure Mentor**) for more information about my experiences and my activity for spreading the knowledge please check my webpage **https://mostafaelmasry.com/about-me/** You will find on it my certification, Articles, Technical documents in Azure I participated on writing it also you can find below all of my Social account links

- **LinkedIn Account:** https://www.linkedin.com/in/mostafaelmasry/
- **Twitter Account:** https://twitter.com/Elmasrydba
- **My Azure Articles:** https://lnkd.in/edn6nyY/#AllAzurePosts
- **Microsoft Azure Hero 3X:** https://www.azureheroes.community/user/5363
- **Founder of Database Cloud Tech blog:** https://mostafaelmasry.com/
- **Founder of SQLIdol website:** http://www.sqlidol.com/team/mustafa-el-masry/
- **My Articles on SQLShack:** https://www.sqlshack.com/author/mustafaelmasry/
- **Founder of LinkedIn Group:** https://www.linkedin.com/groups/8406281/
- **My Certifications:** https://www.youracclaim.com/users/mustafa-elmasry/badges
- **Other Certifications:** https://skillsoft.digitalbadges.skillsoft.com/profile/mustafaelmasry/wallet
- **One of the top bloggers on the Middle East in SQL Server**: Around 300 SQL Server Article, also I have **Arabic** posts on Microsoft MSDN
  - ✓ https://msdn.microsoft.com/ar-sa/library/dn974980.aspx
  - ✓ https://msdn.microsoft.com/ar-sa/library/mt131032.aspx
  - ✓ https://msdn.microsoft.com/ar-sa/library/mt147014.aspx
  - ✓ https://msdn.microsoft.com/ar-sa/library/mt131033.aspx
  - ✓ https://msdn.microsoft.com/ar-sa/library/mt131034.aspx

Azure SQL Resources

- **Azure SQL for beginners**: https://aka.ms/azuresql4beginners or from here https://aka.ms/azuresql4beginnersch9
- **Azure SQL Bootcamp**: https://aka.ms/azuresqlbootcamp
- **Azure SQL Workshop**: https://aka.ms/sqlworkshops
- **Azure SQL Workshop Slides**: https://aka.ms/azuresqlworkshopslides
- **Azure SQL fundamentals:** https://aka.ms/azuresqlfundamentalsm
- **Data Exposed**: https://channel9.msdn.com/Shows/Data-Exposed/
- **My Azure SQL Articles**: https://lnkd.in/edn6nyY/#AzureSQL
- **DP-300 Test Practices**: https://www.examtopics.com/exams/microsoft/dp-300/view/
- **Microsoft Learning Path**: https://docs.microsoft.com/en-us/learn/certifications/exams/dp-300
- **Pluralsight Guide**: https://www.pluralsight.com/guides/cloud-certifications:-azure-database-administrator-associate
- **DP-300 Exam Preparation**: https://www.sqlshack.com/how-to-prepare-for-the-exam-dp-300-administering-relational-databases-on-microsoft-azure/
- **DP-300 Study Guide**: https://ravikirans.com/dp-300-azure-exam-study-guide/
- **Udemy Course:** https://www.udemy.com/course/professional-azure-sql-database-administration/

Microsoft CERTIFIED AZURE DATA FUNDAMENTALS
Microsoft CERTIFIED AZURE FUNDAMENTALS
MICROSOFT CERTIFIED PROFESSIONAL Microsoft MCSA SQL 2016 Database Administration Certified 2020
MICROSOFT CERTIFIED PROFESSIONAL Microsoft MCSE Data Management and Analytics Certified 2020
Microsoft CERTIFIED AZURE ADMINISTRATOR ASSOCIATE
Microsoft CERTIFIED AZURE DATA ENGINEER ASSOCIATE
Microsoft CERTIFIED AZURE DATABASE ADMINISTRATOR ASSOCIATE
MICROSOFT CERTIFIED TRAINER Microsoft MCT Trainer 2020 - 2021
Microsoft Azure MENTOR
Microsoft Azure COMMUNITY HERO
Microsoft Azure CONTENT HERO

SQL Server on Azure VM

1- When you are installing SQL Server IaaS on Azure VM part of the process installs the IaaS Agent Extension and this IaaS Agent Extension is are code that is executed on your VM post-deployment to perform some configuration, examples are installing anti-virus features, or installing a Windows feature, also IaaS Agent Extension provides three key features that can reduce your administrative overhead and In addition to these features, the extension allows you to view information about your SQL Server's configuration and storage utilization

- ➢ SQL Server automated backup
- ➢ SQL Server automated patching
- ➢ Azure Key Vault integration

2- SQL Server Licenses on Azure VM

- ➢ **If you aren't participating** in the Microsoft Software Assurance (SA) program: You can deploy SQL on Azure VM from Azure Marketplace images and you will pay per minute for the use of SQL Server (PAY-AS-You-GO)
- ➢ **If you are participating** in the Microsoft Software Assurance (SA) program at this time you have to options either to pay per minute or you can use your License (BYOL), also, you can use Windows Server licensing.

3- **VM Families/Size Options**: Each Family or series is a combination of memory and CPU, add in your note that Microsoft Azure Supporting Resizing the VM and the process will require VM restart.

- ➢ **General-purpose** – this VM provides a balance between CPU to memory and it can be helpful for testing and development environment.
- ➢ **Compute-optimized**: VMs have a high CPU-to-memory ratio and are good for web servers with a medium amount of traffic
- ➢ **Memory-optimized**: High Memory to CPU ratio up to 4 TB of RAM Helpful for most database workloads.
- ➢ **Storage optimized**: fast, local, NVMe storage that is ephemeral and this environment are suited for scale-out data workloads such as Cassandra and you can use it with SQL Server but you need to consider data protection using a feature like Always On Availability Groups or Log Shipping because the storage is ephemeral
- ➢ **GPU** - Azure VMs with GPUs are targeted at two main types of workloads—naturally graphics processing operations like video rendering and processing, but also massively parallel machine learning workloads that can take advantage of GPUs.
- ➢ **High performance computes** - High Performance Compute workloads support applications that can scale horizontally to thousands of CPU cores. This support is provided by high-performance CPU and remote direct memory access (RDMA) networking that provides low latency communications between VMs.

4- **Availability Zones**: Unique physical locations within a region. Each zone is made up of one or more data centers equipped with independent power, cooling, and networking, When you Choose the Availability zone during the VM deployment you can choose between 3 Zones per region this in case your rejoin support the Availability zone.

5- **Availability Sets**: it is the same concept of Availability Zones but instead of spreading the workload across data centers in the region with Availability Sets you will spread the workload across Servers and racks in the same data center. And this option should be used in two cases

- ➢ Availability Zones are unavailable in a region.
- ➢ an application cannot tolerate intra-zone latency.

6- **Azure Site Recovery:** is a service provided by Microsoft can be used to replicate a VM from region to another region in case of outage or disaster So you can replicate the workload from Primary to secondary for physical and VM machines, for more information about the other benefits from this services check this link: https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-overview

7- **SQL backup on Azure VM:** it is the same concept of the SQL Server backup but the benefits in Azure, Microsoft provided (geo-redundant storage (GRS) or read-access geo-redundant storage (RA-GRS)) to replicate your backup across Azure regions, in case of disaster.

8- **Azure Backup Services for SQL Server:** another big benefit from Hosting your SQL Server on Azure VM is Azure backup services and this agent you need to install it on your VM to allow the agent to communicate with Azure service that manages automatic backups of your SQL Server databases, also in when you use the services you will have a centralized location that you can use it for monitoring the backups for more information about How to configure Azure backup services for SQL Server check this link: https://docs.microsoft.com/en-us/azure/backup/tutorial-sql-backup

- ➢ Backup is protected and encrypted.
- ➢ LTR Long term retention.
- ➢ The point in time restore.
- ➢ 15 Min RPO Recovery point objective.
- ➢ Auto-Protect for new databases.
- ➢ Central management and monitoring.
- ➢ Cost-effective.

9- **Azure VM Storage for SQL Server**: As we know SQL Server required very good disks to deliver high performance results in most of us consider SSD disks in an on-premises environment, Azure VM provides 4 Types of storage (Blob, File, queue, table) in the most cases SQL Server will use managed disks and by default when you install VM you will have two managed disks first one used by an operating system and the second one will be hosted on VM with a letter (D:\) this is temporary disk so don't use for critical data otherwise you lose it, After VM deployment you can easily add new managed disks to the VM to use it for your SQL Server data files.

- **Azure Managed disks** provide two types of encryption: Azure Server-side encryption acts as encryption-at-rest and Azure Disk Encryption uses BitLocker on Windows and DM-Crypt on Linux
- Both technologies of encryption are compatible with Azure Key Vault.
- Azure managed disks are block-level storage volumes
- Premiums SSD Support Read-Caching and Ultra SSD not supporting the technique of Stripping your data in multiple disks

10- **Azure Storage Type of disks**

- Standard HDD: suitable for backups and file storage that is infrequently accessed
- Standard SSD: lightly used dev/test workloads or web servers that do a small amount of IO
- Premium SSD: high-throughput and low latency and The best option for SQL Server on Azure VM and it is supporting Read-Caching.
- Ultra-SSD: support high-IO workloads for mission-critical databases with low latency and not supporting the technique of Stripping your data in multiple disks

11- **New Azure Disk Storage Enhancement Azure shared disks Finally is GA** (general availability), Microsoft announced the general availability of shared disks on Azure Disk Storage, that will give the users more Flexibility to Migrate existing on-premises Windows and Linux-based clustered environments to Azure, Shared Disk is a new Feature in Azure Managed disks allow you to attach managed to multipole VM to build cluster APP on Azure and it is supported Windows and Linux

- shared disks available on both Ultra Disks and Premium SSDs
- Shared disks support SQL Server Failover Cluster Instances (FCI
- https://azure.microsoft.com/en-us/blog/announcing-the-general-availability-of-azure-shared-disks-and-new-azure-disk-storage-enhancements/

12- **SQL Server Data and Log files on Azure VM**:

- Enable read-caching on Azure VM disks will be used by SQL Server Data Files.
- Disable read-caching on Azure VM disks will be used by SQL Server Log Files.

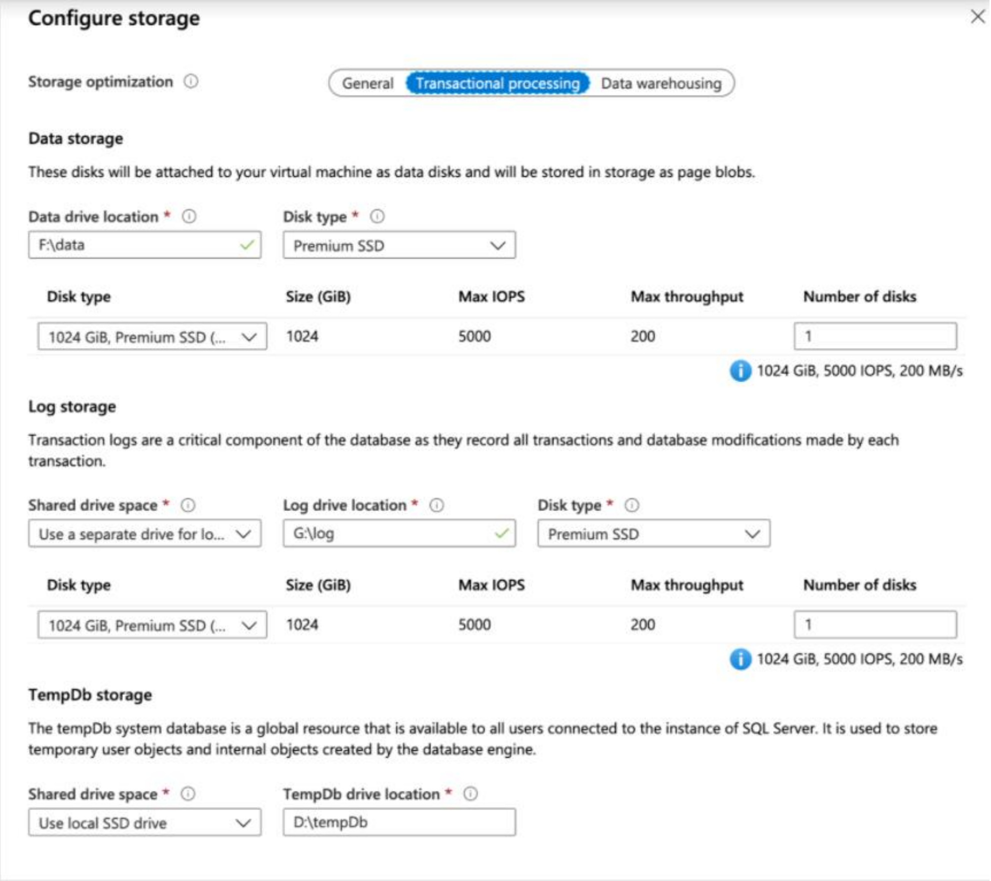
13- **Blob Storage** - Blob storage is what is known as object-based storage and includes cold, hot, and archive storage tiers. In a SQL Server environment, blob storage will typically be used for database backups, using SQL Server's back up to URL functionality.

14- **File Storage** - File storage is effectively a file share that can be mounted inside a virtual machine, without the need to set up any hardware. SQL Server can use File storage as a storage target for a failover cluster instance.

15- **Disk Storage** - Azure managed disks offer block storage that is presented to a virtual machine. These disks are managed just like a physical disk in an on-premises server, except that they are virtualized. There are several performance tiers within managed disks depending on your workload. This type of storage is the most commonly used type for SQL Server data and transaction log files.

16- SQL Server Configuration on Azure VM:

- Create a separate volume for data and transaction log files
- Enable read caching on the data file volume
- Do not enable any caching on the log file volume
- Plan for an additional 20% of IOPs and throughput when building your storage for your VM to handle workload peaks
- Use the D: drive (the locally attached SSD) for TempDB files because TempDB is recreated upon server restart, so there is no risk of data loss
- Enable instant file initialization to reduce the impact of file-growth activities
- Move trace file and error log directories to data disks
- For workloads requiring storage latency under one millisecond, consider using Ultra disk over Premium SSD.



Configure storage

Storage optimization: General | Transactional processing | Data warehousing

Data storage

These disks will be attached to your virtual machine as data disks and will be stored in storage as page blobs.

Data drive location *: F:\data
Disk type *: Premium SSD

Disk type	Size (GiB)	Max IOPS	Max throughput	Number of disks
1024 GiB, Premium SSD (...	1024	5000	200	1

1024 GiB, 5000 IOPS, 200 MB/s

Log storage

Transaction logs are a critical component of the database as they record all transactions and database modifications made by each transaction.

Shared drive space *: Use a separate drive for lo...
Log drive location *: G:\log
Disk type *: Premium SSD

Disk type	Size (GiB)	Max IOPS	Max throughput	Number of disks
1024 GiB, Premium SSD (...	1024	5000	200	1

1024 GiB, 5000 IOPS, 200 MB/s

TempDb storage

The tempDb system database is a global resource that is available to all users connected to the instance of SQL Server. It is used to store temporary user objects and internal objects created by the database engine.

Shared drive space *: Use local SSD drive
TempDb drive location *: D:\tempDb

17- **Azure Migrate Service**: Are Microsoft services used to assess and migrate workload from on-premises VM or physical servers to Azure VM and it can be used to move SQL Database to Azure SQL Database https://docs.microsoft.com/en-us/azure/migrate/migrate-services-overview

18- **To enable automated patching, automated backup in SQL Server on Azure VM:** You need to do unlock for these features by doing Register a SQL Server VM in Azure with the SQL VM resource provider (RP), Automated Patching depends on the SQL Server infrastructure as a service (IaaS) Agent Extension. The SQL Server IaaS Agent Extension (**SqlIaasExtension**) runs on Azure virtual machines to automate administration tasks. The SQL Server IaaS extension is installed when you register your SQL Server VM with the SQL
Server VM resource provider. https://docs.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/sql-server-iaas-agent-extension-automate-management, https://docs.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/sql-vm-resource-provider-register?tabs=azure-cli%2Cbash

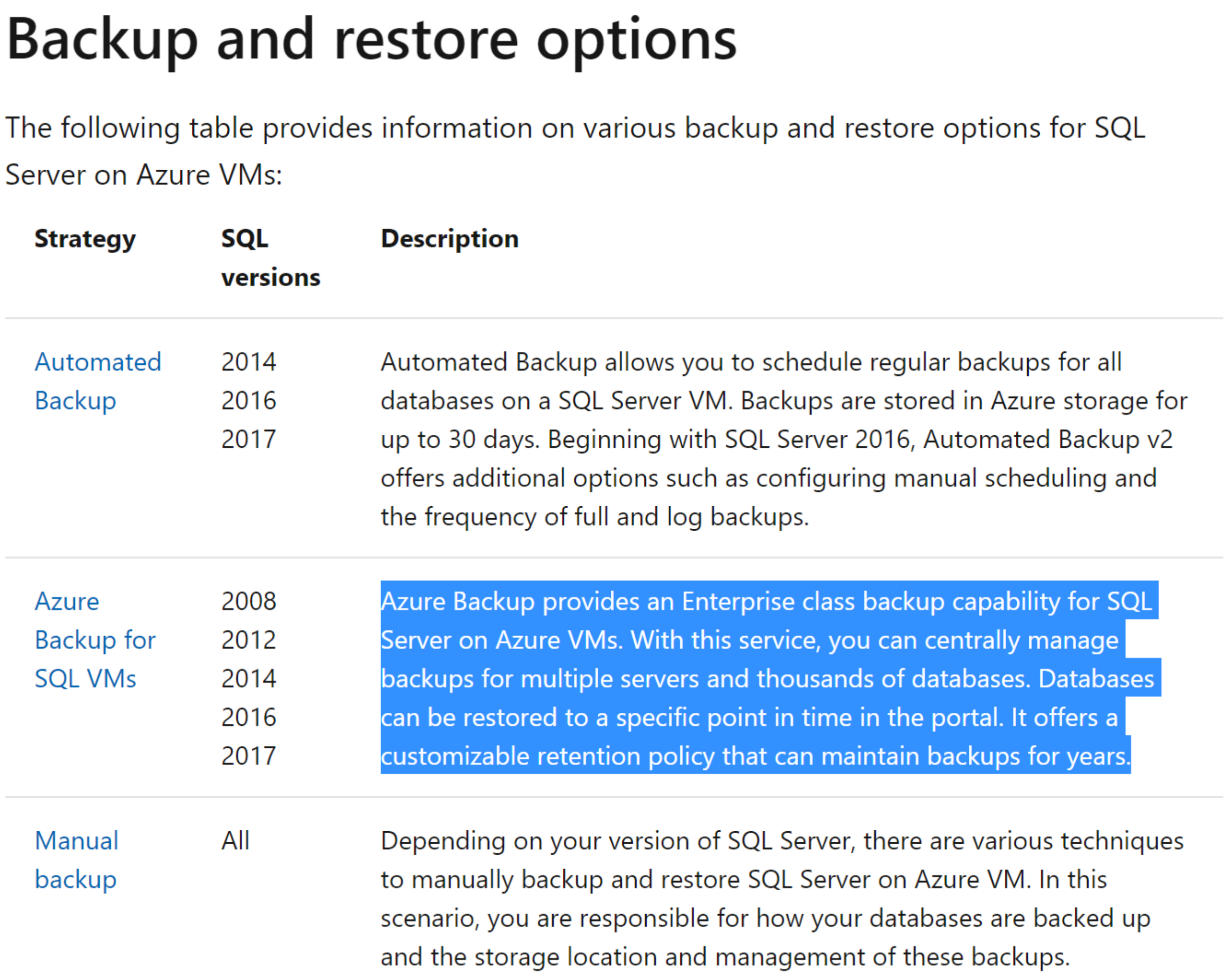
19- **SQL Server Backup on Azure VM:** https://docs.microsoft.com/en-us/azure/backup/tutorial-sql-backup

20- **Automated Backup** allows you to schedule regular backups for all databases on a SQL Server VM. Backups are stored in Azure storage for up to 30 days. Beginning with SQL Server 2016, Automated Backup v2 offers additional options such as configuring manual scheduling and the frequency of full and log backups. And this option support SQL versions (2014, 2016, 2017)

21- **But Azure Backup** provides an Enterprise-class backup capability for SQL Server on Azure VMs. With this service, you can centrally manage backups for multiple servers and thousands of databases. Databases can be restored to a specific point in time in the portal. It offers a customizable retention policy that can maintain backups for years. And this option support 2008, 2012, 2014, 2016, 2017

22- https://docs.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/backup-restore#backup-and-restore-options

23-



Backup and restore options

The following table provides information on various backup and restore options for SQL Server on Azure VMs:

Strategy	SQL versions	Description
Automated Backup	2014 2016 2017	Automated Backup allows you to schedule regular backups for all databases on a SQL Server VM. Backups are stored in Azure storage for up to 30 days. Beginning with SQL Server 2016, Automated Backup v2 offers additional options such as configuring manual scheduling and the frequency of full and log backups.
Azure Backup for SQL VMs	2008 2012 2014 2016 2017	Azure Backup provides an Enterprise class backup capability for SQL Server on Azure VMs. With this service, you can centrally manage backups for multiple servers and thousands of databases. Databases can be restored to a specific point in time in the portal. It offers a customizable retention policy that can maintain backups for years.
Manual backup	All	Depending on your version of SQL Server, there are various techniques to manually backup and restore SQL Server on Azure VM. In this scenario, you are responsible for how your databases are backed up and the storage location and management of these backups.

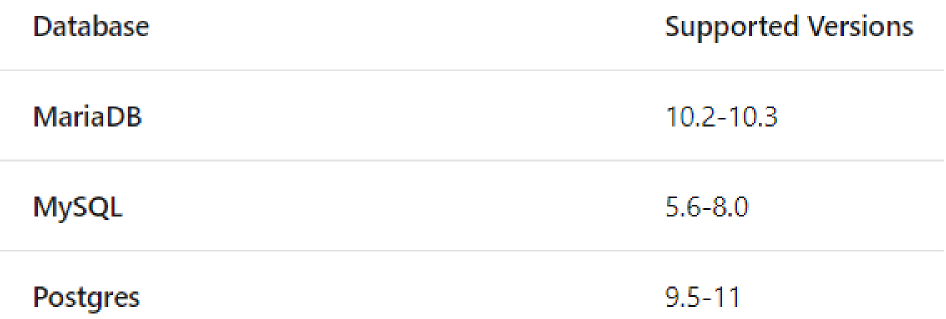
MariaDB, MySQL, and PostgreSQL on Azure

1- **Microsoft Azure Offers 3 different open source** Database Platform on Azure (MySQL, MariaDB, PostgreSQL) and this 3 Database platform service comes with native high availability, automatic patching, automatic backups, and the highest level of security protection

2- **Service Tiers** Supported by open source Database Platform on Azure

- Basic: Best of a light workload
- GP (General Purpose): Best for High Workload required high IO
- Memory-Optimized: Best for High Workload required high performance and in-memory speed.

3- **Supported version** for open Source database platform on Azure



Database	Supported Versions
MariaDB	10.2-10.3
MySQL	5.6-8.0
Postgres	9.5-11

4-

5- **MySQL and MariaDB on Azure Transactions** on either platform are written synchronously to storage. If a node interruption occurs, the database server will automatically create a new node and subsequently attach the storage to the new node. Any transactions in flight are not committed and active connections to the database are dropped. As mentioned with Azure SQL Database, it is important to ensure that applications that connect to the database service include retry logic, also known as connection resiliency, in their database connections.

6- **Database migration** can be done for 3 open source database platform using Microsoft **DMS** Azure Database Migration Service

7- **Azure SQL Database for MySQL and PostgreSQL does not have a TDE:** But Microsoft provided a disk encryption method.

8- **Azure PostgreSQL Deployment Model** (Single Server, Flexible Server {Preview} or Hyperscale {Citus})

9- **Azure PostgreSQL Hyperscale** used for large-scale databases that scale-out across multiple nodes

10- **Azure PostgreSQL Hyperscale** Server Called Nodes and it is working together in a shared-nothing

11- And the nodes are added to the Server Group

12- Each Server group have something called a coordinator node and multiple workers nodes

13- When the APP Sends the Transaction to Azure PostgreSQL Hyperscale, it sends it to the coordinator node and the coordinator node will find the worker nodes to collect the data to the APP.

14- Azure PostgreSQL Hyperscale is sharded, this means the data in a table can be split into multiple nodes using a type of table called a distributed table.

15- During the Deployment of Azure PostgreSQL Hyperscale Microsoft allow you to create additional worker nodes along with a coordinator node

16- You can deploy Up to 20 worker nodes and in case if you need more you should communicate with the Microsoft support team

17- You can connect to Azure PostgreSQL Using **SQL** or **pgAdmin** Clint

18- More information Check Documentation: https://docs.microsoft.com/en-us/azure/postgresql/

19- Query Store Supported in Azure MySQL https://docs.microsoft.com/en-us/azure/mysql/concepts-query-store

Azure SQL Database Architecture

1- **Azure SQL Database Architecture depends on 4 layers (**Client Layer, Services Layer, Platform Layer, Infrastructure Layer

- https://subscription.packtpub.com/book/cloud_and_networking/9781789802542/1/ch01lvl1sec02/the-azure-sql-database-architecture

2- Client Layer: Is the interface for applications to access a SQL database The **Tabular Data Stream** (**TDS**) is used to transfer data between a SQL database and applications. SQL Server also uses TDS to communicate with applications. This allows applications such as .NET, ODBC, ADO.NET, and Java to easily connect to Azure SQL Database without any additional requirements

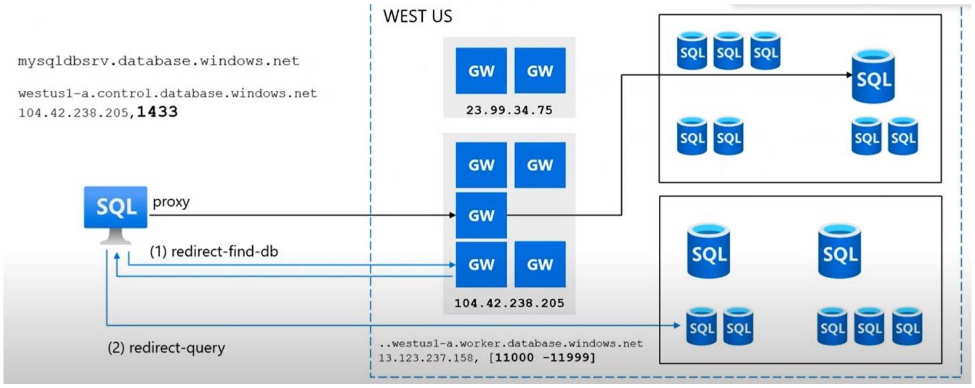
3- Service Layer: acts as a gateway between the client and platform layers

4- Platform Layer: (SQL Server, Azure Fabric, Management Services)

- **Azure Services Fabric:** Responsible for Load Balancing, Automatic Failover and Automatic replication of the SQL Databases between Physical Servers https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview
- **Management Services** takes care of an individual server's health monitoring and patch updates.

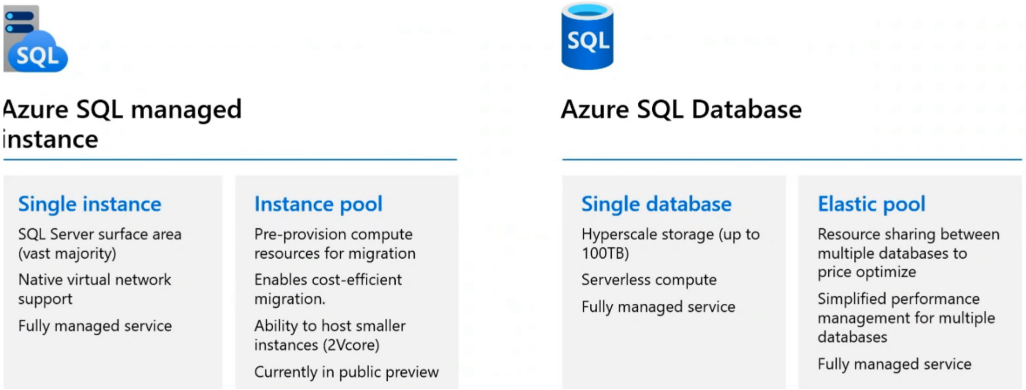
5- Infrastructure Layer: Layer responsible for administrating the OS and physical hardware

Azure SQL Database Connectivity

- **Azure SQL Connectivity How it is work:** we have two option here Proxy and redirect
    - **Proxy:** All connection to Azure SQL used a proxy and this increased the latency and this configuration is the default configuration when you connected to Azure SQL from the outsize azure environment for example from SQL Server SSMS. And Microsoft recommended changing this configuration to redirect to reduce the latency. So, in the Proxy each time your APP will connect to Azure SQL will connect through the getaway.
    - **Redirect**: App connected direct to the node that Azure SQL Database hosted on it and this option reduced the latency and this is the default option when you try to connect to Azure SQL from inside the azure environment, for example, Azure VM, So in Redirect mode, the APP will connect the first time only to Azure SQL Database through the getaway to know the Database IP then in the next time the App will connect directly to the Database node without connecting to the getaway that's why it is best option to reduce the latency.
    - **Default Proxy** outside of Azure and **Redirect** of the inside of Azure
    - **For More information:** Check this article https://docs.microsoft.com/en-us/azure/azure-sql/database/connectivity-architecture.
    - **Update the Azure Connection policy from Proxy to Redirect:** https://channel9.msdn.com/Shows/Data-Exposed/Updating-connection-policies-for-Azure-SQL.
    - 

Introduction to Azure SQL

- Azure SQL Paas Deployment Options.
- Azure SQL Database VS Azure SQL Managed instance MI: https://docs.microsoft.com/en-us/azure/azure-sql/database/features-comparison.



Azure SQL managed instance

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

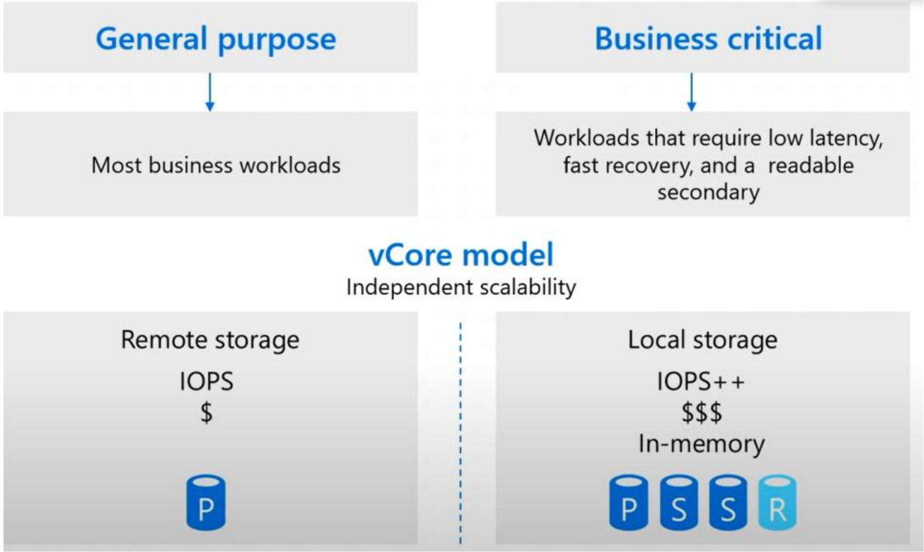
Instance pool
Pre-provision compute resources for migration
Enables cost-efficient migration.
Ability to host smaller instances (2Vcore)
Currently in public preview

Azure SQL Database

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 MI Services Tier Support** (GP General Purpose, BC Business Critical)
  - GP General Purpose Service Tier in MI Support Remote Storage IOPS and it **support only Provision Compute**
  - BC Business Critical Service Tier in MI Support Remote Storage IOPS + In-Memory.
  - GP General Purpose Service Tier in MI Support One Primary replica
  - BC Business Critical Service Tier in MI Support one Primary replica and 3 Secondary replica one of them is Read-only



General purpose
Most business workloads

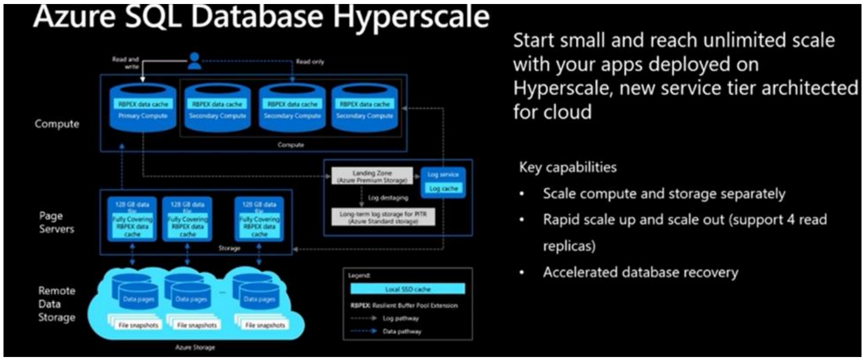
Business critical
Workloads that require low latency, fast recovery, and a readable secondary

vCore model
Independent scalability

Remote storage
IOPS
$

Local storage
IOPS++
$$$
In-memory

- **Azure SQL Single Database Services Tier Support** (GP General Purpose, BC Business Critical, Hyperscale)

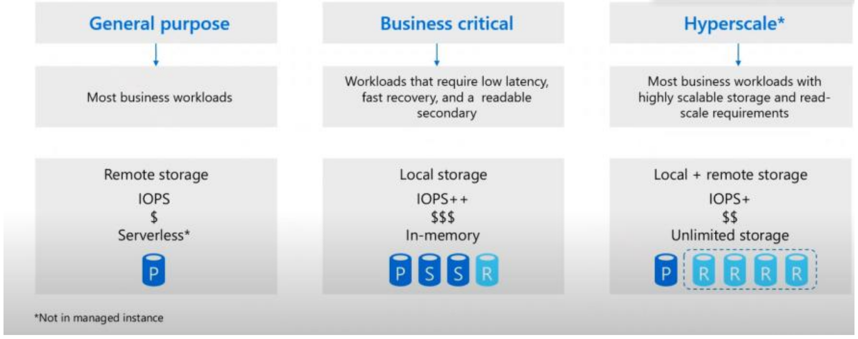
- **GP** General Purpose Service Tier in SQL Single Database Support Remote Storage IOPS and it **support Provision Compute and Serverless Compute**
- **GP** General Purpose Service Tier in SQL Single Database Support One Primary replica
- **BC** Business Critical Service Tier in SQL Single Database Support Remote Storage IOPS + In-Memory.
- **BC** Business Critical Service Tier in SQL Single Database Support One Primary replica and 3 Secondary replicas one of them is Read-only
- **HS Hyperscale** Tier in SQL Single Database Support Local Storage + Remote Storage IOPS + Unlimited Storage + 100 TB Database Size
- **HS Hyperscale** Tier in SQL Single Database Support One Primary replica + 4 Secondary replica all of them are read-only and you can use Round Robin to distribute your read-only transaction on 4 replicas Read More about Round Robin from here: https://www.sqlshack.com/how-to-configure-read-only-routing-for-an-availability-group-in-sql-server-2016/.
- 

Azure SQL Database Hyperscale

Start small and reach unlimited scale with your apps deployed on Hyperscale, new service tier architected for cloud

Key capabilities

- Scale compute and storage separately
- Rapid scale up and scale out (support 4 read replicas)
- Accelerated database recovery

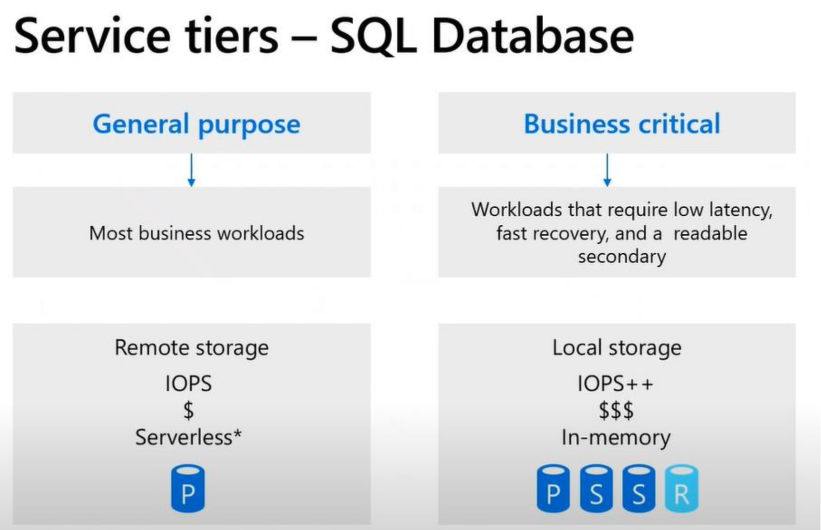


General purpose
Most business workloads
Remote storage
IOPS
$
Serverless*

Business critical
Workloads that require low latency, fast recovery, and a readable secondary
Local storage
IOPS++
$$$
In-memory

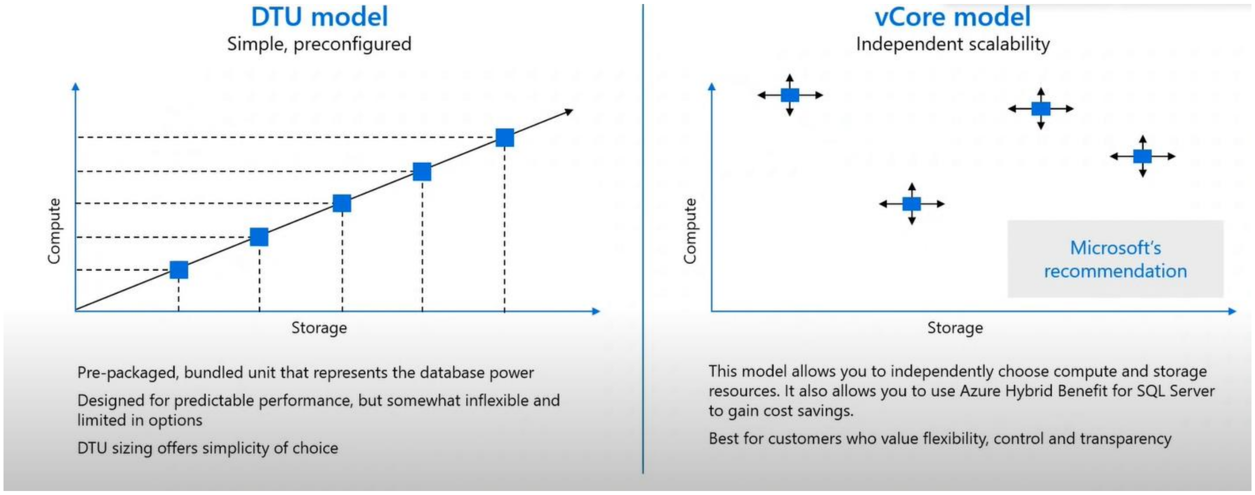
Hyperscale*
Most business workloads with highly scalable storage and read-scale requirements
Local + remote storage
IOPS+
$$
Unlimited storage

*Not in managed instance

- 

  ## Service tiers – SQL Database

General purpose	Business critical
Most business workloads	Workloads that require low latency, fast recovery, and a readable secondary
Remote storage IOPS $ Serverless*	Local storage IOPS++ $$$ In-memory

- Azure SQL Database Provisioned Compute VS Serverless Compute: https://docs.microsoft.com/en-us/azure/azure-sql/database/serverless-tier-overview#comparison-with-provisioned-compute-tier.
- **Azure SQL Purchase Model options** (DTU VS Vcore) Model in Azure Single Database: https://docs.microsoft.com/en-us/azure/azure-sql/database/purchasing-models.
- 

  DTU model

  Simple, preconfigured

  Pre-packaged, bundled unit that represents the database power

  Designed for predictable performance, but somewhat inflexible and limited in options

  DTU sizing offers simplicity of choice

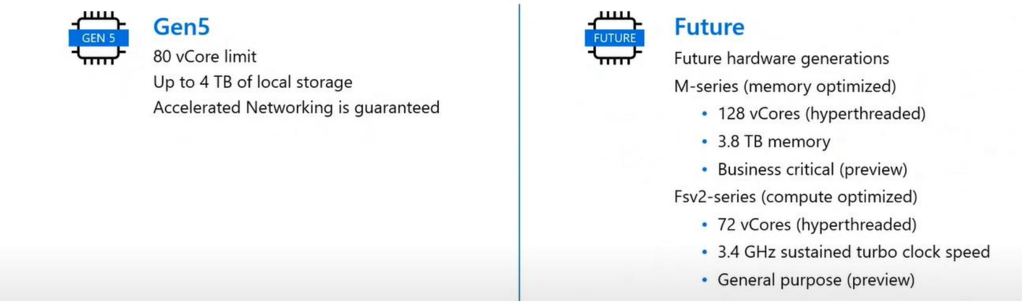
  vCore model

  Independent scalability

  This model allows you to independently choose compute and storage resources. It also allows you to use Azure Hybrid Benefit for SQL Server to gain cost savings.

  Best for customers who value flexibility, control and transparency
- Azure SQL MI Support Backup Copy-only
- In Azure SQL Database Single instance Exactly in GP (General Purpose) Service Tier, we have two options for Computing (Provisioning, Serverless) in Serverless option if the connections on the Database reduced Azure will pause the Database to stop Compute Costs and the only charge you for storage.
- **MS (Microsoft) Recommended** Vcore purchase Model VS DTU model because of two important points can reduce your cost first point is it you can use your on-premises licenses, the second

point in Vcore you can reserve your Capacity as you need from Storage and CPU, add in your note Azure SQL MI Support only Vcore Purchase Model and Azure SQL Single Database support Both Purchase model (DTU and Vcore). For Example, if you have an application need High Compute (CPU and memory) and it needs Low Storage at this time your choice should be Vcore, not DTU

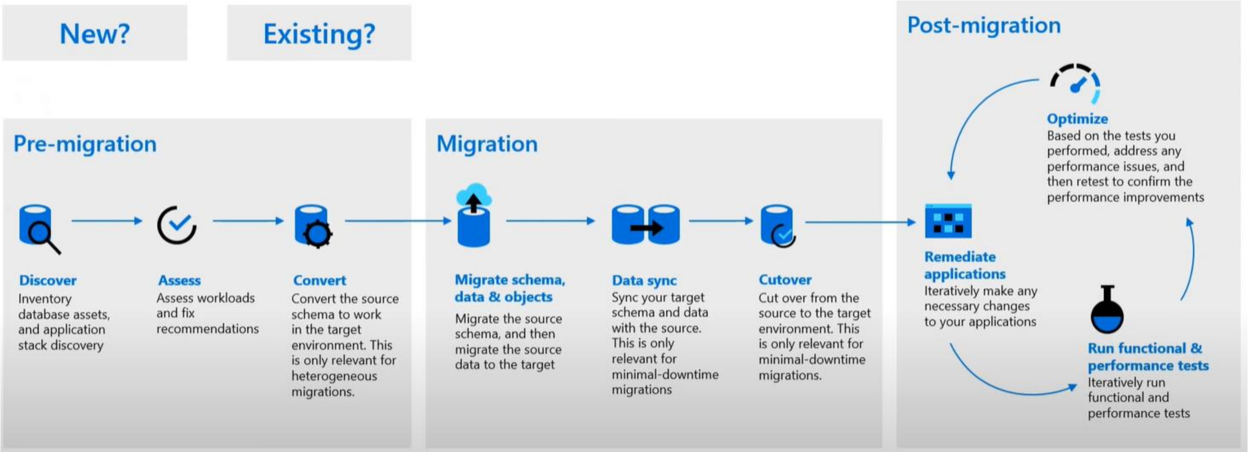
- **Azure SQL Hardware:**
  - MS Recommended **GEN5 (**80 Vcore Limit, up to 4 TB local Storage, accelerated networking are guaranteed**)**
  - 

Gen5
80 vCore limit
Up to 4 TB of local storage
Accelerated Networking is guaranteed

Future
Future hardware generations
M-series (memory optimized)
- 128 vCores (hyperthreaded)
- 3.8 TB memory
- Business critical (preview)

Fsv2-series (compute optimized)
- 72 vCores (hyperthreaded)
- 3.4 GHz sustained turbo clock speed
- General purpose (preview)

- **Azure Database Restore:**
  - In Azure VM you have full control to do normal Backup and restore
  - In Azure SQL Database and Azure SQL Database MI, you have the option to restore the database on a new database because it is a new deployment so there is no option to restore overwrite existing database in Azure SQL.
  - In Azure SQL MI you can use the Restore option from the URL So Restore natively supported only on Azure MI.
  - You can’t restore a backup from Azure SQL or Azure SQL MI to Azure VM because of version less only you can restore it on the same version for example backup take from Azure SQL MI you can restore it only on Azure SQL MI.
- **Azure Migration Process**



New?
Existing?

Pre-migration

Discover
Inventory database assets, and application stack discovery

Assess
Assess workloads and fix recommendations

Convert
Convert the source schema to work in the target environment. This is only relevant for heterogeneous migrations.

Migration

Migrate schema, data & objects
Migrate the source schema, and then migrate the source data to the target

Data sync
Sync your target schema and data with the source. This is only relevant for minimal-downtime migrations

Cutover
Cut over from the source to the target environment. This is only relevant for minimal-downtime migrations.

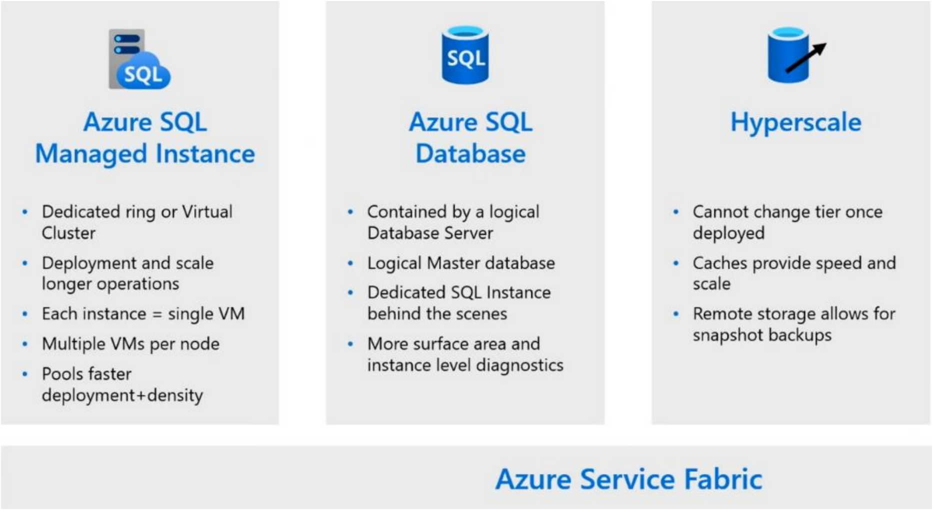
Post-migration

Remediate applications
Iteratively make any necessary changes to your applications

Run functional & performance tests
Iteratively run functional and performance tests

Optimize
Based on the tests you performed, address any performance issues, and then retest to confirm the performance improvements

- The most three things can be affected by your choice for Azure SQL deployment options is (Tempdb Size, Max Log Size, Azure SQL Backup Retention)
- **Azure SQL key deployment details:**



Azure SQL Managed Instance

- Dedicated ring or Virtual Cluster
- Deployment and scale longer operations
- Each instance = single VM
- Multiple VMs per node
- Pools faster deployment+density

Azure SQL Database

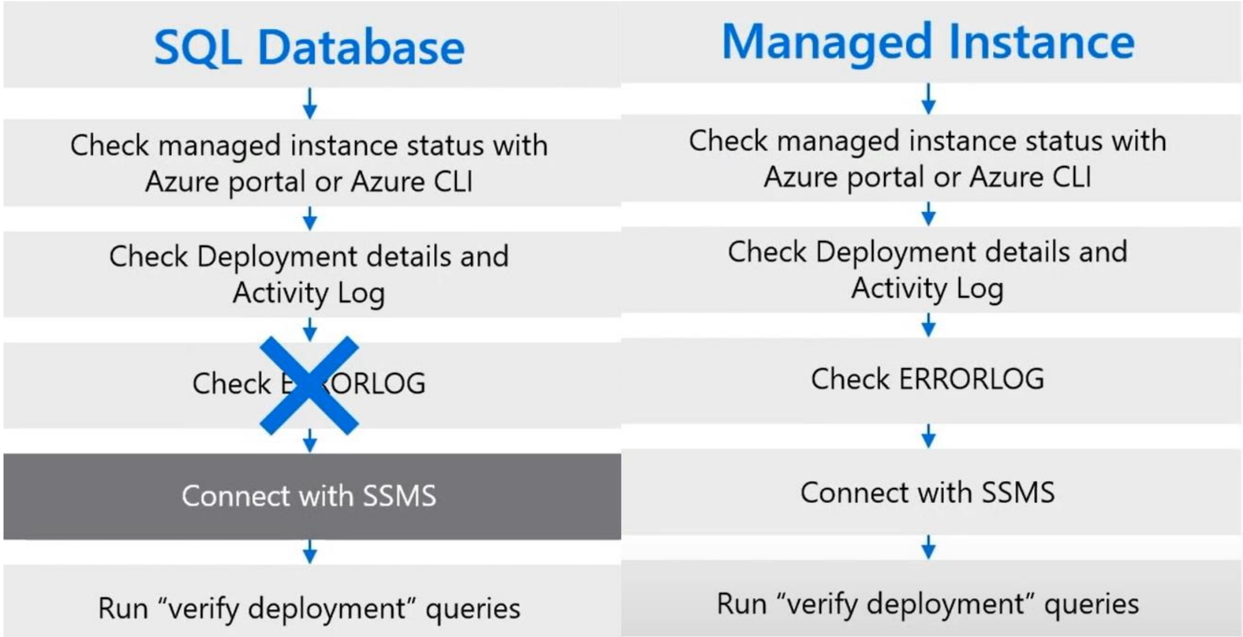
- Contained by a logical Database Server
- Logical Master database
- Dedicated SQL Instance behind the scenes
- More surface area and instance level diagnostics

Hyperscale

- Cannot change tier once deployed
- Caches provide speed and scale
- Remote storage allows for snapshot backups

Azure Service Fabric

- **How to do verification on your Azure SQL after deployment:**



SQL Database

Check managed instance status with Azure portal or Azure CLI

Check Deployment details and Activity Log

~~Check ERRORLOG~~

Connect with SSMS

Run "verify deployment" queries

Managed Instance

Check managed instance status with Azure portal or Azure CLI

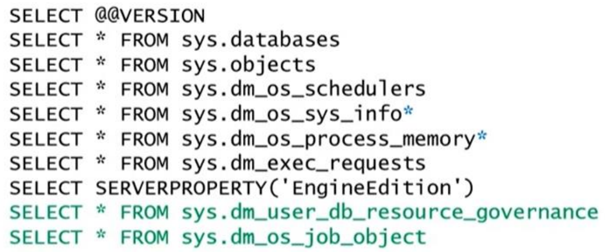
Check Deployment details and Activity Log

Check ERRORLOG

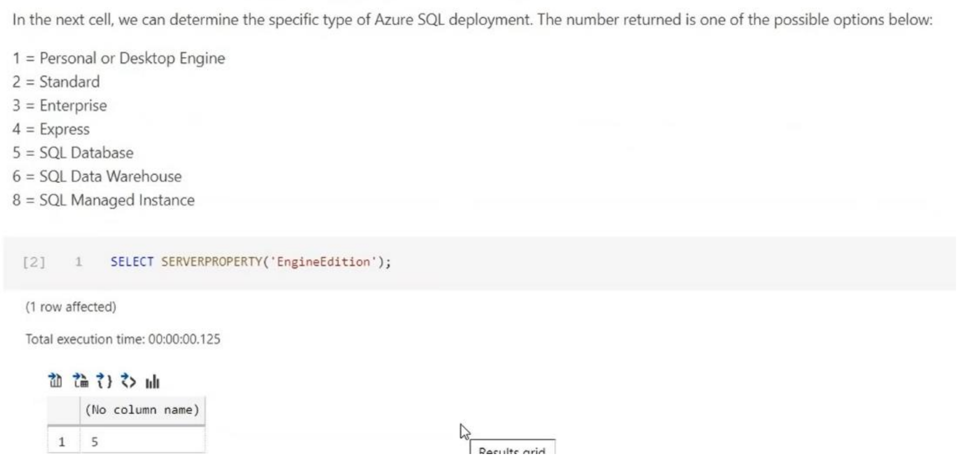
Connect with SSMS

Run "verify deployment" queries

- IF you need to know more about the quirks that you can use it to verify Azure SQL deployment check the below image and this demo https://channel9.msdn.com/Series/Azure-SQL-for-Beginners/Demo-Verify-Azure-SQL-15-of-61.



```
SELECT @@VERSION
SELECT * FROM sys.databases
SELECT * FROM sys.objects
SELECT * FROM sys.dm_os_schedulers
SELECT * FROM sys.dm_os_sys_info*
SELECT * FROM sys.dm_os_process_memory*
SELECT * FROM sys.dm_exec_requests
SELECT SERVERPROPERTY('EngineEdition')
SELECT * FROM sys.dm_user_db_resource_governance
SELECT * FROM sys.dm_os_job_object
```

- 

In the next cell, we can determine the specific type of Azure SQL deployment. The number returned is one of the possible options below:

1 = Personal or Desktop Engine
2 = Standard
3 = Enterprise
4 = Express
5 = SQL Database
6 = SQL Data Warehouse
8 = SQL Managed Instance

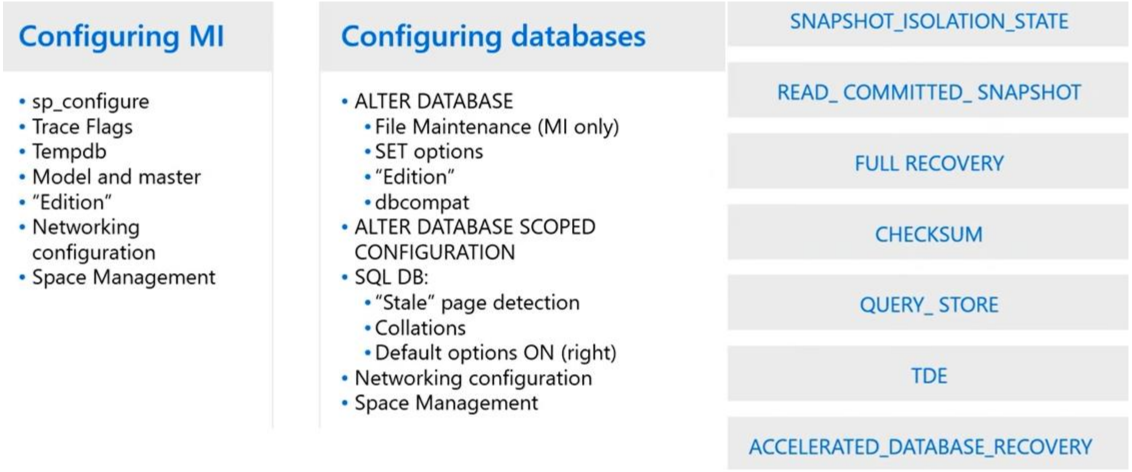
```
[2]   SELECT SERVERPROPERTY('EngineEdition');
```

```
(1 row affected)

Total execution time: 00:00:00.125
```

	(No column name)
1	5

-
- **System Database in Azure SQL**: you will not able to see the (Tempdb, Model, MSDB) like normal SQL Server only you will See the master database.
- **Azure Database Configuration:** There is some configuration you can do it on Azure SQL and Azure SQL MI and some other configuration supported only by Azure SQL MI



Configuring MI

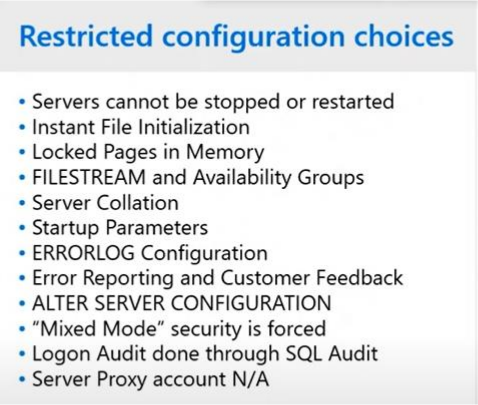
- sp_configure
- Trace Flags
- Tempdb
- Model and master
- "Edition"
- Networking configuration
- Space Management

Configuring databases

- ALTER DATABASE
  - File Maintenance (MI only)
  - SET options
  - "Edition"
  - dbcompat
- ALTER DATABASE SCOPED CONFIGURATION
- SQL DB:
  - "Stale" page detection
  - Collations
  - Default options ON (right)
- Networking configuration
- Space Management

SNAPSHOT_ISOLATION_STATE
READ_ COMMITTED_ SNAPSHOT
FULL RECOVERY
CHECKSUM
QUERY_ STORE
TDE
ACCELERATED_DATABASE_RECOVERY

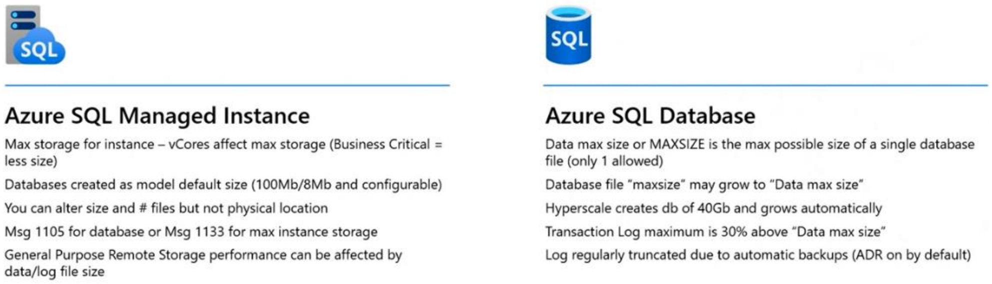
-
- **Azure SQL Restricted Configuration:** There is some configuration you cannot do it on Azure SQL or Azure SQL MI



Restricted configuration choices

- Servers cannot be stopped or restarted
- Instant File Initialization
- Locked Pages in Memory
- FILESTREAM and Availability Groups
- Server Collation
- Startup Parameters
- ERRORLOG Configuration
- Error Reporting and Customer Feedback
- ALTER SERVER CONFIGURATION
- "Mixed Mode" security is forced
- Logon Audit done through SQL Audit
- Server Proxy account N/A

- **Learn More about Azure SQL Database Space**



Azure SQL Managed Instance

Max storage for instance – vCores affect max storage (Business Critical = less size)

Databases created as model default size (100Mb/8Mb and configurable)

You can alter size and # files but not physical location

Msg 1105 for database or Msg 1133 for max instance storage

General Purpose Remote Storage performance can be affected by data/log file size

Azure SQL Database

Data max size or MAXSIZE is the max possible size of a single database file (only 1 allowed)

Database file "maxsize" may grow to "Data max size"

Hyperscale creates db of 40Gb and grows automatically

Transaction Log maximum is 30% above "Data max size"

Log regularly truncated due to automatic backups (ADR on by default)

- **Example:** IF your DB is hosted on Azure SQL Database and your DB size is 10 GB at this time the DB log Size will be 30% from the total Database size (3 GB) and the Log size will be truncated by the Automated backup taken by azure.
- **Change Azure SQL Database Service Level Objectives from SSMS:** https://sqlespresso.com/2020/08/28/how-to-quickly-change-azure-sql-database-service-level-objectives/.

1- Azure SQL Database allows you to choose from two different **purchasing models**:
   - ➢ DTU Database Transaction Unit Model: combining compute, storage, and I/O resources
   - ➢ Vcore Model: Allow you to purchase Core based on your workload

2- **Azure SQL Database include**: Automatic Backup, patching, Built-in high availability, and SQL Server Feature Enhancement

3- **Azure SQL Service Tier** Options are based on the Purchase model
   - ➢ DTU Purchase Model
     - ✓ Basic
     - ✓ Stander
     - ✓ Premium
   - ➢ Vcore Purchase Model
     - ✓ General Purpose: Azure premium storage

- ✓ Business Critical: high performing workloads offering the lowest latency, Local SSD, built-in read-only database replica
- ✓ Hyperscale: supports databases of up to 100 TB

4- **Azure SQL Database backup**: by default, backup is taken automatically and the backup file saved into Azure blob storage geo-redundant and the backup retained for between 7 and 35 days, based on the service tier of the database, Basic and vCore databases default to seven days of retention, and on the vCore databases this value can be adjusted by the administrator, and the retention period can be extended up to 10 years using long-term retention (LTR) and you can use read-accessible geo-redundant blob storage

- ➢ FULL Backup Weekly
- ➢ DIFF Backup every 12 Hours
- ➢ Log Backup: Daily Every 5-10 minutes depending on transaction log activity

5- **Azure SQL Database Restore**: You can do restore the database using a portal, PowerShell, and T-SQL and there is no Restore overwrite existing current database, restore meaning new Database deployment it seems like creating a new database, and you cannot do restore using T-SQL this feature supported only in Azure SQL MI.

6- **Geo-replication**: Is Microsoft Azure Services used for replicating database up to 4 secondaries and you can do manual failover or automatic failover but when the database failover the APP should change the connection string because there is no Failover Group here.

7- **Failover Group**: Built-in services on the top of the Geo-Replication to give the user tow endpoint connection one direct the APP for the Primary replica and second one direct the APP for the Read-only replica and here when the Database Failover happened the APP no need to change the connection string in case the APP using the Failover Group endpoint

8- **Provisioned**: is compute tier used in Azure SQL Database general-purpose services tier, Compute resources are pre-allocated and billed per hour based on the number of configured vCores

9- **Serverless**: is the compute tier used in Azure SQL Database general-purpose services tier, and it used to Scale up or down the resources for the Database based on the Workload, So in case the Workload on Azure Database doesn’t need Resource the Serverless will pause the Database and you will not be charged during this Paused only you will be charged for storage and when the connection return again to the database the serverless will resume the database

- ➢ Serverless is billed per second based on the number of vCores utilized
- ➢ With Serverless you can configure the Minimum and Maximum resources
- ➢ Auto pause delay option between 60 MIN to 7 days
- ➢ Any applications using serverless should be configured to handle connection errors and include retry logic, as connecting to a paused database will generate a connection error
- ➢ Some Features will not be paused such as (Geo-replication, LTR, A job database in elastic jobs, SQL Data Sync)

10- **Hyperscale**: Azure SQL Database Limited the database size to 4 TB but in Hyperscale you have a database size up to 100 TB, note that once an Azure SQL Database is converted to Hyperscale, you cannot convert it back to a "regular" Azure SQL Database

11- I**n Azure SQL Database there are two sets of firewall rules**, server-level firewall rules, and database-level firewall rules.

12- **Both server and database level firewalls** use IP Address rules instead of SQL Server Logins

13- **Server level firewalls** are configured to allow users to connect to the master database and all databases on the instance

14- **Database level firewalls** are used to grant or block specific IP Addresses from accessing specific databases.

15- **The server Firewall** Can be configured using Azure Portal or with sp_set_database_firewall_rule Stored Procedure.

16- **Database level Firewall** rules are configured through T-SQL only using the sp_delete_database_firewall_rule stored procedure

17- **The Private Link feature** allows you connect to Azure SQL Database using a private endpoint

18- **Encryption at rest does not inherently encrypt data within the database**. It protects someone restoring a backup to an unsecured server or making a copy of a database and transaction log file and attaching it to another unsecured server

19- **TDE Enabled on Azure SQL Database:** The data is the database is encrypted as the data is written to the data page and decrypted when the data page in memory is accessed. The result is that all data pages on disk are encrypted.

20- **Azure SQL Database Created** after May 2017 have TDE enabled automatically

21- **Databases that were created** before May 2017 will have TDE disabled by default

22- **Azure SQL Managed Instance,** databases that were created after February 2019 have TDE enabled.

23- **Azure SQL Managed Instance Databases** created before February 2019 will have TDE disabled.

24- **Azure Key Vault** is a tool used for storing and accessing secrets

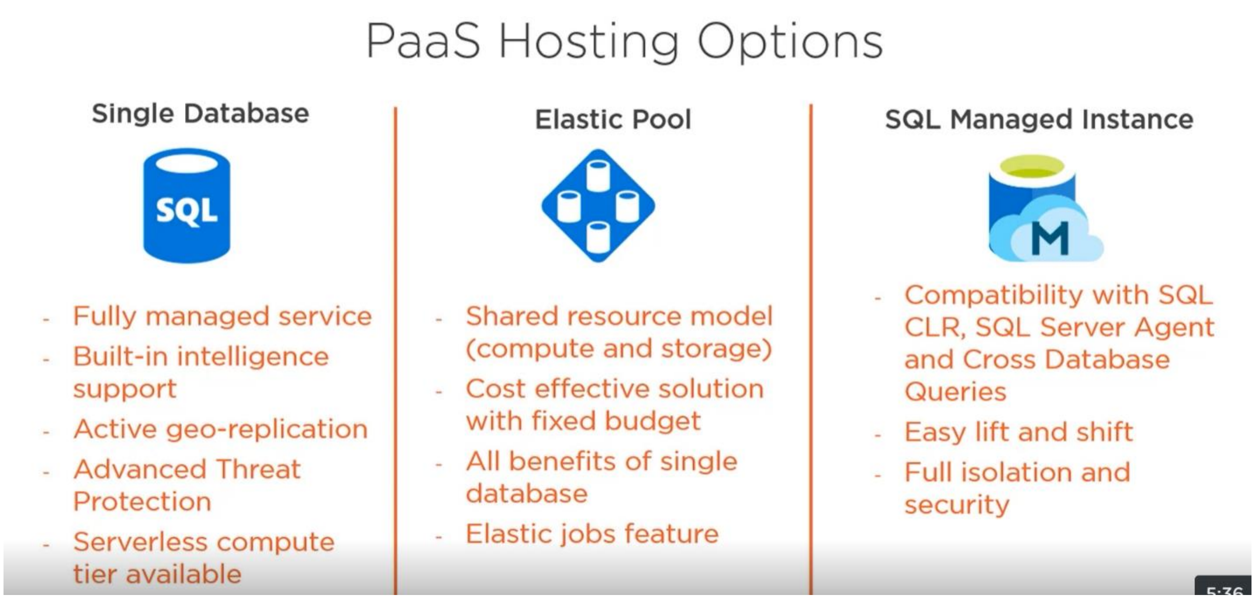
25- Always Encrypted and Secure enclaves: https://docs.microsoft.com/en-us/learn/modules/protect-data-transit-rest/5-explain-object-encryption-secure-enclaves

26- dynamic data masking: https://docs.microsoft.com/en-us/learn/modules/protect-data-transit-rest/6-explain-dynamic-data-masking

27- Dynamic Data Masking can be implemented in the Azure portal or using T-SQL

28- **Azure Private Link** enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure-hosted customer-owned/partner services over a private endpoint in your virtual network.
Traffic between your virtual network and the service travels the Microsoft backbone network.
Exposing your service to the public internet is no longer necessary Reference:
https://docs.microsoft.com/en-us/azure/private-link/private-link-overview

Azure SQL Database Types



PaaS Hosting Options

Single Database

- Fully managed service
- Built-in intelligence support
- Active geo-replication
- Advanced Threat Protection
- Serverless compute tier available

Elastic Pool

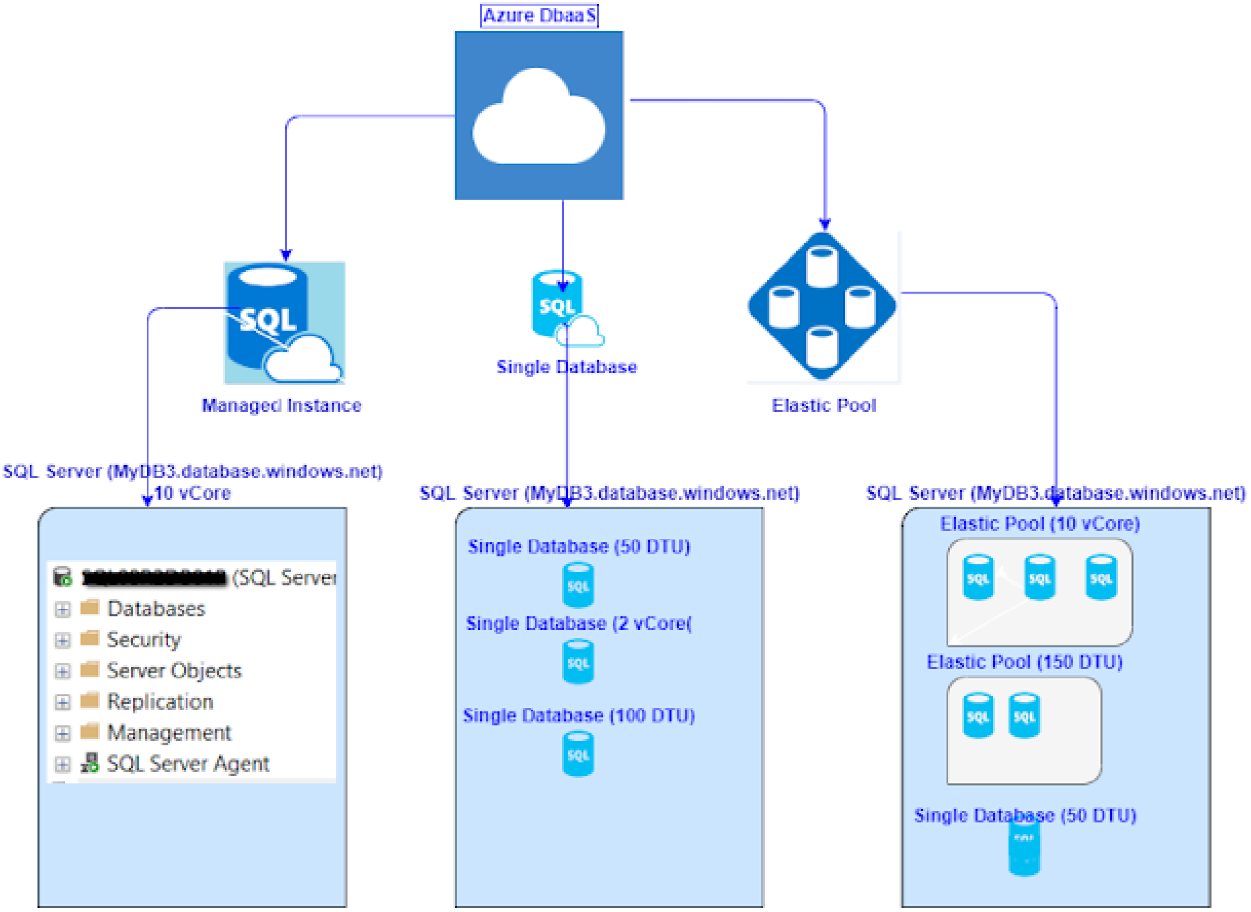
- Shared resource model (compute and storage)
- Cost effective solution with fixed budget
- All benefits of single database
- Elastic jobs feature

SQL Managed Instance

- Compatibility with SQL CLR, SQL Server Agent and Cross Database Queries
- Easy lift and shift
- Full isolation and security

1. **Single Database:** This is a fully managed service with built-in artificial intelligence to support administration tasks for your database. Also, you have several useful features available that allow you to easily, if highly available, secure, and perform in a database with minimal effort required from your site. In terms of cost, you have an amazing option for a pricing tier called serverless. This means you use a pay-as-you-go pricing model and you end up only paying for what you use
2. **Elastic Database**: Elastic Pool provides compute and storage resources that are shared between all the databases hosted in the pool. Elastic pools provide a simple and cost-effective solution for managing the performance of multiple databases within a fixed budget with all the benefits of Azure SQL Database. This option is especially useful in scenarios where you have predefined usage patterns of your database, and typically they are not all using resources at the same time, so you distribute those resources to the different databases being used from time to time. Elastic Pool also provides useful features to make it easier to perform your administration tasks, such as elastic jobs where you can execute jobs to perform tasks against your database in an automated way
3. **Managed Database**: SQL managed instance. In case you are managing complex databases using features such as SQL CLR, SQL Server Agent, or even cross-database queries, you would face some limitations in case you plan to migrate your database to Azure. A managed instance is the solution to overcome those limitations since it is one of the most intelligent database

services that requires 0 code changes and provides near 100% compatibility for you to migrate your SQL database to Azure. This option is a bit concurrent of VMs hosting your database since you can easily and quickly migrate your SQL database with near no code changes and assuring almost 100% of compatibility with your database considering that you do not need full control over your environment since with this option you have a full abstraction of the environment hosting your database. Also, it provides you mechanisms to fully isolate the environment hosting your database and enhance the security of the environment hosting your database.



Scaling out Azure SQL Database

- Azure SQL Database Scaling has two types:
  - ✓ **Vertical**: Switching the DB to Higher or lower Services tier, scaling up allows you to improve the performance, scaling down allow you to reduce the cost
  - ✓ **Horizontal**: dividing data from a single table into different individual databases
- Changing the services tier for azure SQL database when you are doing scale up or scale down it is not required any downtime.
- Automatically scale up or scale down the services tier on Azure DB can be done based on DTU (**Database Throughput unit**) usage.
- **Sharding Azure SQL Database:** is a technique to distribute large amounts of identically structured data across several independent databases https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-scale-introduction

Azure SQL Database

A single database is a fully-managed database as a service (DbaaS) under the Azure PaaS offering. This deployment option is most suited for modern application development like microservices. The single database deployment option creates a database in Azure with its own set of resources and is managed via a SQL database server. With a single database, each database resources are isolated from each other. A SQL database server is completely different from the SQL server that we used to have in the on-premise installation. In this context of azure DbaaS, an SQL database server is a logical construct that acts as a central administrative point for one or more single databases.

1- Default Collection in Azure SQL Database SQL_Latin1_General_CP1_CI_AS
2- DBCC SHRINKDATABASE Working With Azure SQL but DBCC SHRINKFILE not working with Azure SQL.
3- The Single Database deployment option creates a single isolated database in Azure SQL Database
4- Azure Single Database Support DTU-based and Vcore-based Purchased Model
5- The internal SQL Database server is hidden from the end-user and all are managed by Microsoft Azure
6- Azure Single Database has it is own Resources (DTU or V-Core) and this resource will not be shared with other databases in the same single instance. Each Database has dedicated resources.
7- Single Database Isolated meaning if you created two single databases on the same single database server each DB Each Database has dedicated resources and the two databases can be communicated together and I think this one is a security feature
8- This means Azure Single Database is the best option for small workloads with applications connects to one DB.

9- To Provision Azure SQL Single Database, you need (resource Group, Database name, Server Name, User Name, Password, Location, deployment option either single or elastic pool, Compute Storage and purchase model either DTU-based or vCore-based)

10- In Azure SQL Single database provisioning you can create new or select existing one from (Resource group and Server name)

11- In Azure SQL Single database provisioning the (resource Group, Database name, Server Name) Should be a unique name.

12- The single database does not provide access to the Operating system.

13- It does not allow you to specify the version of the SQL Server. The single database always runs the latest stable SQL engine version, which is equal to or higher than the latest available RTM version of SQL Server.

14- In short, a Single database supports only database-level features and does not support server level features

15- Azure SQL Database Support PaaS option Like No separate purchase of infrastructure and managing the underlying hardware of the database server

16- Azure SQL Database Support PaaS option Like Automated patching and version upgrade. There is no option to control the maintenance window of patching or upgrade. To handle the connection error during the maintenance window, you need to implement the retry mechanism in your code

17- Azure SQL Database Support PaaS option Like No headache of scheduling and monitoring of backups. Automated backups are available

18- Azure SQL Database doesn’t support:

- SQL Agent service
- DB mail service
- Service broker
- Replication (can be push subscriber)
- Change Data Capture (CDC)
- CLR creation
- linked server
- To choose the time zone
- File Stream and File table

19- Azure Database provides two types of purchase Model:

- DTU Base
- vCore Base

20- Users can select the option based on their workload. In the vCore model, azure allows you to bring your license.

Service Tier and Purchase Model

1- Purchase Model (DTC Base, vCore Base)

2- Service Tier in **DTU**

- **Basic**: lowest tier available and applies to small, infrequently used applications (2 GB, 5 DTU, 30 Concurrent User, 300 concurrent sessions, 5$ Cost)
- **Stander**: it is the best choice for a web application or workgroup with low or medium IO performance requirement and it supported 3000 S12 DTU and 1 TB Data Max Size
- **Premium**: For Critical application with High transaction volume, it supports large numbers of logins, High IO performance, it contains 6 performance level (P1, P2, P4, P6, P11, P15), 4 TB, The Premium service tier supports read scale-out and zone redundancy (**Azure Availability zone)** when opting for the Premium service tier, you can choose the Azure SQL Database to be zone-redundant.
  - ✓ **In Premium Services Tier you have the option for Read-Scale out** and this capability redirects the read-only client connections to one of the automatically provisioned HA replicas and effectively doubles the compute capacity of the database or elastic pool at no additional charge. This is ideal for load balancing of complex analytical workloads without affecting the primary OLTP workload https://azure.microsoft.com/en-us/updates/general-availability-read-scale-out-support-for-azure-sql-database/
  - ✓ **In Premium Services Tier you have an option for database zone redundant:** Azure SQL Database Premium tier supports multiple redundant replicas for each database that are automatically provisioned in the same datacenter within a region. This design leverages the SQL Server Always ON technology and provides resilience to server failures with 99.99% availability SLA and RPO=0 https://azure.microsoft.com/en-us/blog/azure-sql-database-now-offers-zone-redundant-premium-databases-and-elastic-pools/#:~:text=Azure%20SQL%20Database%20Premium%20tier,availability%20SLA%20and%20RPO%3D0.

3- Service Tier in vCore Base:

- **GP General Purpose:** Scalable Computer and Storage option and it is containing two compute tiers:
  - ✓ **Provisioned**: Compute resource are pre-allocated billed per hours based on vCores configured (up to 80 vCores, up to 408 GB memory, 4TB Data MAX Size)
  - ✓ **Serverless**: Compute resource are out-scaled billed per second based on vCores used, you can configure the MAX vCores and Min vCores, Auto pause delay (up to 40 vCores, up to 120 GB memory, 4 TB) https://docs.microsoft.com/en-us/azure/azure-sql/database/serverless-tier-overview
- **Hyperscale**: In the Hyperscale tier, storage costs are calculated based on actual allocation. Allocated space increases automatically as needed, up to 100 TB (up to 80 vCores, up to 408 GB memory), Supported Secondary Replicas and Configurable backup

storage redundancy is currently not supported for Hyperscale. By default, data is stored in geo-redundant (RA-GRS) storage blobs that are replicated to a paired region. https://docs.microsoft.com/en-us/azure/azure-sql/database/service-tier-hyperscale

- **Business Critical:** High transaction rate and high resiliency (up to 80 vCores, up to 408 GB memory, 4TB), Supported **Read-Scale out and database zone redundant**
  - ✓ **In Business-Critical Services Tier you have the option for Read-Scale out** and this capability redirects the read-only client connections to one of the automatically provisioned HA replicas and effectively doubles the compute capacity of the database or elastic pool at no additional charge. This is ideal for load balancing of complex analytical workloads without affecting the primary OLTP workload https://azure.microsoft.com/en-us/updates/general-availability-read-scale-out-support-for-azure-sql-database/
  - ✓ **In Business-Critical Services Tier you have an option for database zone redundant:** Azure SQL Database Premium tier supports multiple redundant replicas for each database that are automatically provisioned in the same datacenter within a region. This design leverages the SQL Server Always ON technology and provides resilience to server failures with 99.99% availability SLA and RPO=0 https://azure.microsoft.com/en-us/blog/azure-sql-database-now-offers-zone-redundant-premium-databases-and-elastic-pools/#:~:text=Azure%20SQL%20Database%20Premium%20tier,availability%20SLA%20and%20RPO%3D0.

4- By default, data is stored in geo-redundant (RA-GRS) storage blobs that are replicated to a paired region in all services tiers except in **Hyperscale not supported.**

5- Azure Database Purchase Model (DTU, V-Core) https://docs.microsoft.com/en-us/azure/sql-database/sql-database-purchase-models

6- Azure Database Service Tier(General Purpose, BC Business-critical, Hyperscale) https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-general-purpose-business-critical

7- DTU- Based Service tier: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-service-tiers-dtu

8- DTU-Based Calculator: https://dtucalculator.azurewebsites.net/

9- Azure SQL Database managed instance not supported DTU-Based Servies Teir: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance

10- Premium and Business Critical service tiers leverage the Premium availability model, which integrates compute resources (sqlservr.exe process) and storage
(locally attached SSD) on a single node. High availability is achieved by replicating both compute and storage to additional nodes creating a three to a four-node cluster.
By default, the cluster of nodes for the premium availability model is created in the same datacenter. With the introduction of Azure Availability Zones, SQL
A database can place different replicas of the Business-Critical database to different availability zones in the same region. To eliminate a single point of failure, the control ring is also duplicated across multiple zones as three gateway rings (GW). https://docs.microsoft.com/en-us/azure/azure-sql/database/high-availability-sla.

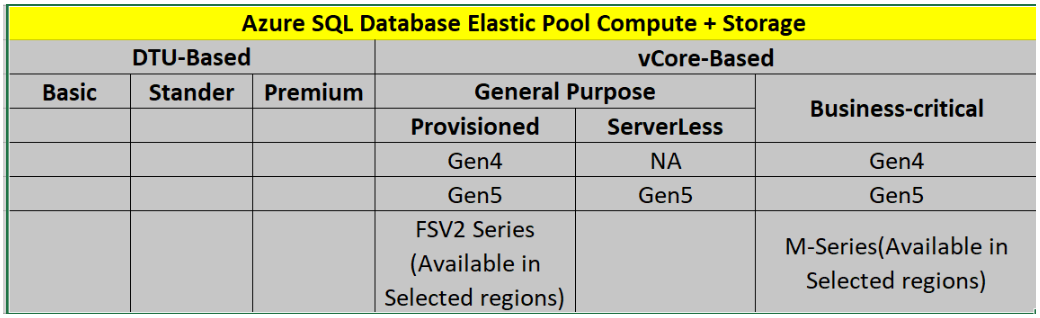
Azure SQL Database Elastic Pool

1- Hosting Multiple databases sharing the resources of the DTU and you can specify exactly DTU for each database even you can scale-out and scale-in the DTU based on peak time
2- Azure SQL Elastic Pool is Collection for Multiple Azure Single Databases
3- Azure SQL Database elastic pool is the most cost-effective solution for multiple databases
4- The Databases in the Elastic pool are on a single Azure SQL database but all of the databases sharing the same resources of the DTU so if we assume the Elastic pool have 100 DTU and we have 5 databases on the elastic pool all of the 5 databases are sharing the 100 DTU
5- IF you have for example around 3 Azure Single Database and you need to move them to one Azure SQL Database elastic pool is it applicable the answer yes you can do it
6- We can host up to 5000 databases on Azure SQL Database elastic pool
7- Azure SQL Database elastic pool can have up to 45000 DTU
8- All Databases hosted on the Azure SQL Database elastic pool should be in the same region and the same resource group.
9- IF you are looking for an economic feature so Azure SQL Database elastic pool is your choice
10- SQL Database resource limits and resource governance → https://docs.microsoft.com/en-us/azure/sql-database/sql-database-resource-limits-database-server
11- More information About Azure SQL Database elastic pool → https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-pool

Azure SQL Database Elastic Pool Compute + Storage

While deploying the Azure SQL Database Elastic Pool you will find one option called (**COMPUTE + STORAGE**) in this section you will be able to select the Purchase model DTU or Vcore and each one support services tier and each services tier support some kind of storage and some other benefits and limitation in the resource, AS you can see below Azure SQL Database Elastic Pool database supported the DTU-based with 3 services tier (basic, Stander, premium) and supported also the vCore-based with 2 services tier (General Purpose, Business-critical)

1- Azure SQL Database Elastic Pool Not Supported Hyperscale Services Tier Like Azure Single Database
2- Azure SQL Database Elastic Pool Supported General Purpose with Compute Hardware Type Provisioning and not support Compute Hardware Type Serverless like Azure Single Database



Azure SQL Database Elastic Pool Compute + Storage					
DTU-Based			**vCore-Based**		
Basic	**Stander**	**Premium**	**General Purpose**		**Business-critical**
			Provisioned	**ServerLess**	
			Gen4	NA	Gen4
			Gen5	Gen5	Gen5
			FSV2 Series (Available in Selected regions)		M-Series(Available in Selected regions)

An elastic Pool Is a set of single databases with a shared resource pool. In a single database, each database will have a dedicated resource. In the elastic pool, resources are configured at the pool level and every single database connected with that pool shares the resource of the elastic pool. SQL Database elastic pools are a simple solution for managing and scaling multiple databases that have varying and unpredictable usage demands. For example, let us assume you are providing some SaaS solutions to multiple clients in different time zone. Each customer data are stored in independent databases. In this scenario, the resource can be utilized in a much better way by adding those databases to the elastic pool instead of allocating dedicated resources to each customer database. Apart from this, the rest of the characteristics are the same as the Single database mentioned in the above section.

Elastic pools are a deployment option in which you purchase Azure compute resources (CPU, memory, and storage) that is then shared among multiple databases:

- ➢ From Azure Portal > Setting > Configuration: You can do many configurations on Elastic pool like changing service tier, increase database size, Change Pool Size DTU or Vcore
- ➢ If you change the size of the pool the active connection will be drooped.
- ➢ https://docs.microsoft.com/en-us/learn/modules/deploy-azure-sql-database/4-deploy-elastic-pool

Azure SQL Database MI Managed Instance

There are two deployment options here Azure SQL MI Managed instance and Azure SQL instance pool, MI is a fully functional SQL Server instance that is almost 100% compatible with your on-premises ecosystem including features like SQL Agent, access to Tempdb, cross-database query, and common language runtime (CLR). The service uses the same infrastructure as the Azure SQL Database and includes all the benefits of the PaaS service such as automatic backups, automatic patching, and built-in high availability

Why Azure SQL MI take a longer time than Azure SQL Database in Deployment operation?

Behind the scenes, for Azure SQL Managed Instance, Azure deploys a dedicated ring (sometimes called a virtual cluster) for your service. This architecture helps in providing security and native virtual network support. Because of this architecture, deployment and scaling operations can take longer. For example, when you scale up or down, Azure deploys a new virtual cluster for you and then seeds it with your data. You can think of every instance as running on a single virtual machine.

Azure SQL Instance pools were introduced to help with the long deployment time. You can pre-deploy a "pool" of dedicated resources. Deploying into a pool and scaling within a pool is faster than traditional deployments. (And you get a higher packing density because you can deploy multiple instances within a single VM.)

1- A managed instance is one of the most suitable deployment options for those who are moving to the cloud from an on-premise server. This deployment model supports most of our on-premise database features.
2- The managed instance does not provide access to the Operating system.
3- It does not allow you to specify the version of the SQL Server. Managed instance always runs the latest stable SQL engine version, which is equal to or higher than the latest available RTM version of SQL Server.
4- The managed instance also supports to bring your license with software assurance (SA). Considering all these, managed instances are the best option for lift and shift of your existing workload to the cloud. Note that, if you need direct access to OS/filesystem or dependent on a specific version of SQL server or required specific features that are not supported in the Azure SQL server then Azure Paas database service is not suitable unless you resolve those dependencies
5- Azure SQL Managed instance Supporting two service Tiers (General Purpose and Business Critical)
6- Azure SQL Managed instance Management operation (Instance Create can take from 4 to 6 hours, instance update can take around 2.5 Hours, instance Delete can take around 1.5 hours) https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#managed-instance-management-operations

7- Azure SQL Managed instance not appeared under SQL Database Category it appears under a different category Called **SQL Managed instance**

8- allowing restores from on-premises backups

9- provide an entire SQL Server instance, allowing up to 100 databases

10- Service tier is (Business Critical and General Purpose)

11- General Purpose, which uses storage replication for availability, and Business critical using multiple replicas

12- A standalone Managed Instance offers a 99.99% Service Level Agreement (SLA) which guarantees at most 52.60 minutes of downtime per year

13- In MI you can manually make a copy-only backup of a database. You must back up to a URL, as access to the local storage is not permissible, and you should disable TDE.

14- To restore from one instance to another, both instances must reside within the same Azure subscription as well as the same Azure region

15- You cannot do restore for all instance only individual database

16- Restore on Existing Database not supported

17- You can do restore using T-SQL Command

18- You must restore from a URL endpoint. You do not have access to local drives

19- Backup files containing multiple log files cannot be restored

20- Backup files containing multiple backup sets cannot be restored

21- Backups containing In-Memory/FILESTREAM cannot be restored

22- By default, the databases in a managed instance are encrypted using Transparent Data Encryption (TDE)

23- SQL managed instance is a fully managed SQL Server it provides 100% surface area compatibility with on-premises SQL Server instances

24- It is the easiest option for DB migration from on-premises

25- When you migrate to a managed instance on Azure, you don't only migrate databases, you migrate licenses too.

26- Important Features supported by SQL Server Azure managed instances that are not supported by Azure SQL DB (Native backup and restore, Global temporary tables, Cross-database queries and transactions, Linked servers, CLR modules, SQL agent, Database mail, Transactional replication)

27- DTC (**Distributed Transaction Coordinator**) Services not supported by SQL Server Azure managed instances.

28- **The price tier is V-Core based purchasing model** that gives you the flexibility to custom the SQL Server compute, memory, and storage based on different workload requirements

29- **When your Provision managed instance**, it is created on Azure VM, A virtual cluster can have one or more managed instances.

30- SQL managed instance supports Azure Virtual Network (**VNET**)

31- **Azure Subscription types that are supported by managed instance**: subscription types: Enterprise Agreement (EA), Pay-As-You-Go, Cloud Service Provider (CSP), Enterprise Dev/ Test, and Pay-As-You-Go Dev/test.

32- Resource Governor is a feature in SQL Server and Azure SQL managed instance that allows you to granularly control how much CPU, physical IO, and memory resources can be used by an incoming request from an application https://docs.microsoft.com/en-us/learn/modules/configure-sql-server-resources-optimal-performance/5-control

33- SQL Agent service

34- DB mail with an external SMTP server

35- Service broker

36- SQL Agent service

37- Transactional replication

38- Change Data Capture (CDC)

39- CLR creation from binary (Not using the assembly file)

40- linked server

41- Managed instance supports SQL Server Integration Services (SSIS) and can host SSIS catalog (SSISDB) that stores SSIS packages, but they are executed on a managed Azure-SSIS Integration Runtime (IR) in Azure Data Factory (ADF)

42- Not yet supporting the file stream or file table

43- Migration to the managed instance is much easier as this supports restoring from the native backup created from the on-premises server. To restore the backup in the managed instance, the backup should be available in the Azure storage account and should use RESTORE DATABASE FROM URL. Managed instances also allow the customer to take COPY_ONLY backups which do not break the azure automated backup chain.

44- A managed instance is placed inside the Azure virtual network and in a dedicated subnet. This provides:

- Secure private IP address.
- The ability to connect an on-premises network to a managed instance.
- The ability to connect a managed instance or another on-premises database server through a linked server.
- Managed instances also provide public endpoints. Public endpoint provides the ability to connect to the Managed Instance from the Internet without using a VPN. Access is disabled by default unless explicitly allowed. We need to explicitly whitelist the IP address to access through the public IP address.

45- Managed instance provides two flavors of computing and storage

- **General Purpose**: This supports most of the production workload. Support up to 80 vCore and 8TB fast storage
- **Business Critical**: For IO intensive and compute-intensive workload. Support up to 80 vCore and 4 TB super-fast storage

46- SQL managed instances support two connection types, **Redirect** and **Proxy:**

- **Redirect is the recommended connection** type because the client directly connects to the node hosting the database, and therefore it offers low latency and high throughput

- **In the Proxy connection type**, requests to the database are proxied through the Azure SQL Database gateways

47- We can take a backup from SQL Server Database on-premises and save it on Azure Blob Storage then on Azure Managed instance you can do restore from Azure Blob Storage using T-SQL Query

48- When you are Migrating SQL DB to Azure Managed instance using SQL Server backup Make sure to take every backup on a separate backup media (backup files). Azure Database Migration Service doesn't support backups that are appended to a single backup file and make sure this backup was taken with Checksum option because (Azure Database Migration Service only supports backups created using checksum)

Differences between SQL Server On-Premises and Azure SQL Managed instance

1- High Availability on the Managed instance is built-in but on SQL Server on-premises it is required Pre-Configuration
2- Full Physical paths not supported on Azure Managed instance
3- Azure Active Directory Authentication is the replacement of Windows Authentication in the Azure Managed instance
4- Azure Managed instance automatically managed Filegroup, in Memory OLTP objects
5- SSIS not Supported in Azure managed instance it is replaced by **ADF** Azure data factory

Azure Managed instance Service Tier

1- General Purpose Service Tier: Used High performance Blob Storage Up to (8 TB)
2- Business Critical Service Tier: Used Super-Fast Local SSD Up to 1 TB on Gen4 and Up to 4 TB in Gen5
3- General Purpose Service Tier: Supporting Built-in High Availability
4- Business Critical Service Tier: Supporting Built-in High Availability on Always on Availability Group and it gives additional Read-only DB option to overload the read-Only Workload.

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#vcore-based-purchasing-model

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#managed-instance-service-tiers

Azure SQL Managed instance Security

Azure Managed instance supporting all of the security features supported In Azure single database and Azur SQL Database elastic pool

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#azure-sql-database-security-features

(TDE, Threat Protection, RLS Row-Level Security, Dynamic data masking, Managed instance auditing, Azure AD integration) Plus other features supported only for Azure managed instance (**Managed instance Security advanced options**):

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#managed-instance-security-isolation

1- A managed instance using a native Virtual network that is allowed the connection from on-premises to use this network to connect using Azure express route or VPN Gateway
2- In Azure Managed instance by default, it is allowed the SQL Endpoint in only exposed through Private IP and this allowing safe connectivity
3- Azure SQL Managed instance deployed on Single-tenet this meaning it has dedicated infrastructure

Azure SQL MI References

1- Understanding Azure SQL Database managed instance
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas#a-closer-look-at-azure-sql-database-and-sql-server-on-azure-vms
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-resource-limits#service-tier-characteristics

2- Securing a managed instance (VNET implementation, private IP & single-tenant infrastructure)
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#advanced-security-and-compliance
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-connectivity-architecture

3- Provision an Azure SQL database managed instance
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance-get-started
   - ➢ https://docs.microsoft.com/en-us/sql/sql-server/stretch-database/stretch-database?view=sql-server-2017

4- Key differences between SQL Server on-premises and a managed instance
   - ➢ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-managed-instance#key-differences-between-sql-server-on-premises-and-in-a-managed-instance

Differences between Azure SQL Database and SQL Server and Non-Supported Features

- **Backup and Restore on Azure DB:**
  - ✓ Normal database backup and restore statements aren't supported.
  - ✓ Backups are automatically scheduled and start within a few minutes of the database provisioning.
  - ✓ Backups are consistent, transaction-wise, which means that you can do a point-in-time restore.
  - ✓ There is no additional cost for backup storage until it goes beyond **200%** of the provisioned database storage
  - ✓ You can reduce the backup retention period to manage backup storage costs.
  - ✓ You can also use the long-term retention period feature to store backups in the Azure vault for a much lower cost for a longer duration
  - ✓ Apart from automatic backups, you can also export the Azure SQL Database **BACPAC** or **DACPAC** file to Azure storage
  - ✓ Database file shrink NO
  - ✓ Backup Retention Period: in basic tier 7 days, in stander and Premium tier 35 days, however, we have "**Long Term backup retention**" for 10 years but it is still under preview
  - ✓ Restore process not overwriting the Existing DB, this meaning you will pay the extra cost because you restore additional Azure DB
- **Recovery Model**
  - ✓ The recovery model cannot be modified because the master DB in Azure is read-only
  - ✓ Recovery mode for any DB on Azure is FULL Model
- **SQL Server Agent**
  - ✓ Azure SQL Server doesn't have SQL Server Agent
  - ✓ As a workaround, we can use SQL agent on an on-premise or on an Azure SQL VM to connect and run on the Azure SQL database
  - ✓ **Azure Automation** allows users to schedule jobs in Microsoft Azure to automate manual tasks.
  - ✓ **Elastic Database Jobs** is an Azure Cloud service that allows the scheduled execution of ad hoc tasks
- **Database Mail:** it is not supported on Azure but we can use Alert Rule that can be used to add certain matrices and notification email
- **Event and Notification:** Not supported on Azure because it is depending on the Service broker and this also not supported but we can use Alert Rule that can be used to add certain matrices and notification email

- **Change Data Capture**
  - ✓ CDC is as a concept is not available on Azure DB because it is depending on SQL jobs and SQL agent is not available in Azure SQL DB, but we can do the same job using an alternative solution like (temporal tables, SSIS, or Azure Data Factory)
- **Auditing**
  - ✓ C2 Auditing, Extended Event, SQL Trace, and anything else writing alerts and events in the log is not available in Azure SQL DB This is because it's a PaaS
  - ✓ But in SQL Azure there is an auditing and threat-detection feature available
  - ✓ In single DB and elastic pool auditing working on DB level, in Managed instance auditing working on the server level and log files stored on Blob storage, in SQL Server on-premises or virtual machines Auditing working on the server level and events are stored on the file system or windows event.
- **Mirroring**
  - ✓ You cannot build mirroring between two azure SQL DB, but we can use Azure SQL DB is a mirror server
  - ✓ You can also set up a readable secondary for an Azure SQL database, which is better than mirroring.
- **Table Partitioning**: partition scheme and partition function is allowed in Azure
- **Replication**
  - ✓ We can’t create replication (Snapshot, transactional, and merge replication) between two SQL Azure DB.
  - ✓ But we can use Azure SQL DB as a subscriber to an on-premise starting from SQL 2012 or Azure VM SQL Server but It will support one-way direction not (Peer to Peer and bi-direction replication) and it will support only push subscription.
- **Multi-Part Names**
  - ✓ You can't access the tables in different SQL databases in Azure on the same Azure this means the **Linked server** not supported
  - ✓ But you can use an elastic query to access tables from different databases from an Azure SQL server.
- **Unsupported Features** (SQL Browser Service, Filestream, filetable, Common Language Runtime (SQL CLR), Resource Governor, Global Temporary Tables, Log Shipping, SQL Trace and Profiler, Trace Flags, System Stored Procedures, USE Statement, SP_Configure, SSRS, SSIS, SSAS (You can use Azure DWH), SQL Profiler)

Azure Database Migration

To migrate your existing workload to a single database, there are three primary methods.

- **Export to a BACPAC** file and import the BACPAC file into an azure single database. (A BACPAC file is a ZIP file with an extension of BACPAC containing the metadata and data from a SQL Server database. A BACPAC file can be stored in Azure Blob storage or local storage in an on-premises location and later imported back into Azure SQL Database or a SQL Server on-premises installation). This is a time-consuming process and requires downtime. The time required to complete the migration depends on the size of the database.
- **Using transactional replication**: Azure single database can be configured as a subscriber of your on-premise database publication. There is no UI available. Need to configure through T-SQL. Once the complete data synched with the source server, on migration day cut down the traffic to your on-premise database, and after synchronizing the data, point your application to the new Azure database
- Use the **Azure DMS** (Data Migration service).

Azure database server does not allow you to host inside the Vnet. By default, the access is through the public endpoint. All the traffic through this endpoint is blocked and you need to explicitly whitelist the required IP address to connect to the database server through this endpoint. There is no option to disable the private endpoint but as mentioned earlier, no one can connect through the public endpoint unless you provide access by whitelisting the IP addresses. You can enable private endpoints by integrating virtual network private endpoints. Private endpoint helps to connect from the same VNet or peered VNet in the same/cross-region or from on-premises using a VPN. Note that virtual network private endpoint will be billed separately

Below a list of all tools, you can use it on this project some of them can be acting as the assessment tool, and some of them acting the migration tool offline or online migration, and some of them doing both jobs. The first 4 tools are the most commonly used.

1- **Azure Data Migration Services (DMS):** it is an Online migration using Azure DMS services it can be used to migrate multiple databases from multiple sources to multiple targets, DMS services used the DMA (Data Migration Assistant) services to generate assessment tool, and DMS services it can be used for automated migration for more information about how to use this services check this Microsoft video (https://azure.microsoft.com/en-us/resources/videos/online-migrations-using-azure-dms/)

2- **Data Migration Assistant DMA**: it is a Microsoft tool used for assessment to check the compatibility, function, and features that are not compatible with Azure even and the same tool used to migrate the DB from on-premises to Azure SQL Database, you can download it easily from this link https://www.microsoft.com/en-us/download/details.aspx?id=53595 and install it on your local PC or any on-premises server. For more information about the tool check, this post https://docs.microsoft.com/en-us/sql/dma/dma-overview?view=sql-server-ver15 and for how to use it check this video https://www.youtube.com/watch?time_continue=385&v=qu-euCEnaFI&feature=emb_title

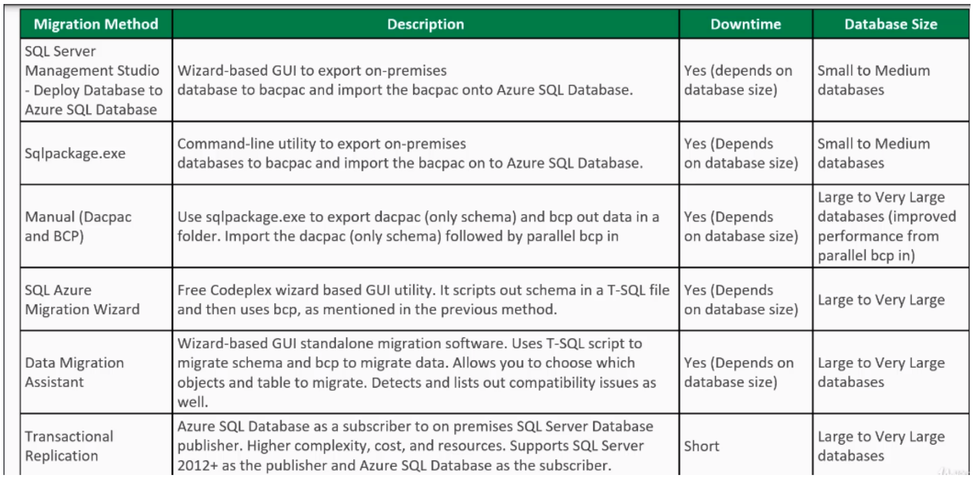
- **Transactional Replication:** SQL Server replication features can be used as services to migrate DB from on-premises to Azure SQL DB but SQL on Premises should be not less than SQL version 2012 SP2 CU8 and SQL on Premises will be acting as the publisher and the Azure SQL DB will be acting as a subscriber it will support only push subscription it is the best option for short downtime and large database for more information check Microsoft Post (https://docs.microsoft.com/en-us/azure/sql-database/sql-database-single-database-migrate#method-2-use-transactional-replication)
- **Export and Import BACPAC File using SSMS (Export data-tier application)**: For more information check this post-http://itproguru.com/expert/2015/03/how-to-move-or-migrate-sql-server-workload-to-azure-sql-database-cloud-services-or-azure-vm-all-version-of-sql-server-step-by-step/
- **SSDT SQL Server Data Tool**
- **SQLPackage.exe**: it is SQL CMD Package from Microsoft used for testing the compatibility issue and to migrate the DB from on-premises to Azure SQL Database with importing, exporting the BACPAC or DACPACk files
- **SQL Azure Migration Wizard SAMW:** It is community support Code Plex with GUI interface used for testing the compatibility issue and to migrate the DB from on-premises to Azure SQL Database but this tool replaced now by DMA (Data Migration Assistant)

Migrating other databases to Azure Database

If you target to migrate any other DB like Oracle, MongoDB, MySQL, DB2, PostgreSQL, Cassandra, MariaDB, access, SAP you should look into this site https://datamigration.microsoft.com/

References and interesting articles

- Database Migration from on-prem to Azure SQL
- Moving Your SQL Workload to the Cloud
- Azure SQL Database documentation
- Azure SQL Database pricing
- Azure Database Migration Service
  Azure Database Migration Service documentation
- Azure Database Migration Guide Step-by-step guidance for modernizing your data assets
- Migrating Databases to Azure SQL Database
- What's new in SQL Database V12

- 

Migration Method	Description	Downtime	Database Size
SQL Server Management Studio - Deploy Database to Azure SQL Database	Wizard-based GUI to export on-premises database to bacpac and import the bacpac onto Azure SQL Database.	Yes (depends on database size)	Small to Medium databases
Sqlpackage.exe	Command-line utility to export on-premises databases to bacpac and import the bacpac on to Azure SQL Database.	Yes (Depends on database size)	Small to Medium databases
Manual (Dacpac and BCP)	Use sqlpackage.exe to export dacpac (only schema) and bcp out data in a folder. Import the dacpac (only schema) followed by parallel bcp in	Yes (Depends on database size)	Large to Very Large databases (improved performance from parallel bcp in)
SQL Azure Migration Wizard	Free Codeplex wizard based GUI utility. It scripts out schema in a T-SQL file and then uses bcp, as mentioned in the previous method.	Yes (Depends on database size)	Large to Very Large
Data Migration Assistant	Wizard-based GUI standalone migration software. Uses T-SQL script to migrate schema and bcp to migrate data. Allows you to choose which objects and table to migrate. Detects and lists out compatibility issues as well.	Yes (Depends on database size)	Large to Very Large databases
Transactional Replication	Azure SQL Database as a subscriber to on premises SQL Server Database publisher. Higher complexity, cost, and resources. Supports SQL Server 2012+ as the publisher and Azure SQL Database as the subscriber.	Short	Large to Very Large databases

Azure SQL backup

- **Backup Concept:** Backup on SQL Azure DB is taken Automatically and the first backup is taken after the DB Provisioning, however, you can't download this backup or doing a manual restore from it.
- **Backup Schedule**: Full Backup weekly, DIFF backup every one hour, Log backup every 5 RTO min
- **Backup Storage:** Microsoft support us with free storage based on the DB size on Services Tier selected, and it gives us 2 MAX size of the DB size, Azure SQL Database backups are stored in geo-replicated blob storage (RA-GRS storage type)
- **Reaching Backup Storage Limit**: at this time, we have two solutions (**1**) Reduce the retention period by Contacting the Support team and this time you will not pay an extra charge (**2**) Pay for an extra backup build at the stander Read Access Geographically redundant storage rate
- **Archiving data**: In case if we have data archiving, we have two solutions to archive it on azure (**1**) Export the DB as a **BACPAC** file and save it on Azure blob storage (**2**) Use **Long-Term-backup-retention (LTR)**to the Azure **backup vault** and you can keep it to 10 years
- **Backup retention period:** It is depending on the services tier model (basic = 7 days, stander and premium = 35 days) however we have "**Long Term backup retention**" for 10 years to an azure **backup vault**

Update: Azure changed the backup retention policy by default **7** days for all pricing models but we have to change to **35** days for standard and premium tiers and managed instances manually.

- **Backup Azure Geo-Replicated:** this means the Azure replicated the backup cross regions, and in case of the region down you can restore the backup in another region, but this feature replicated the FULL and DIFF backup only. This meaning the RPO will be one hour because the DIFF taking every one Hour
- **Azure backup RTO and RPO:** Microsoft Support Point in Time Recovery with 12 Hours as RTO and 5 minutes as RPO for the in-Region backups
- **Disaster:** **If you delete the DB** you can restore it on the retention period time based on your Services tier For Example if your DB in the basic service tier and you delete the DB you can restore the DB in 7 days after this you will not find your backup, **If you delete the server** all backup will be deleted and you will not have any option to restore your DB
- **Manual backup:** There is no Manual backup on Azure SQL Database but we can take a backup from the data and Structure database using another way called **BACPAC** and we can do Export for the **BACPAC** file on the azure storage account using Azure portal, PowerShell and we can do the same to an on-Premises system using sqlpackage.exe, also we can export **BACPAC** file using SSMS SQL Server Management studio
- **When using the Azure Backup service to store backups of virtual machines, which container is required?** Recovery Services Vault

- **What is Azure Backup Explorer** Today, we are pleased to share the preview of Backup Explorer? Backup Explorer is a built-in Azure Monitor Workbook enabling you to have a single pane of glass for performing real-time monitoring across your entire backup estate on Azure. It comes completely out-of-the-box, with no additional costs, via native integration with T Azure Resource Graph and Azure Workbooks. https://azure.microsoft.com/en-us/blog/backup-explorer-now-available-in-preview/?WT.mc_id=linkedin-social-thmaure
- **What are the benefits of Azure backup Explorer?**
  - **At-scale views:** With Backup Explorer, monitoring is no longer limited to a Recovery Services vault. You can get an aggregated view of your entire estate from a backup perspective. This includes not only information on your backup items but also resources that are not configured for backup
  - **Deep drill-downs** – You can quickly switch between aggregated views and highly granular data for any of your backup-related artifacts, be it backup items, jobs, alerts, or policies
  - **Quick troubleshooting and actionability** – The at-scale views and deep drill-downs are designed to aid you in getting to the root cause of a backup-related issue. Once you identify an issue, you can act on it by seamlessly navigating to the backup item or the Azure resource, right from Backup Explorer.
- Azure SQL Taken the backup automatically and it is kept from 7 to 35 days
- Backup Types is Full, Differential, Log transaction
- Full Backup is taken Weekly
- Differential Backup is taken Daily every 24 hours
- Log Transaction Backup is taken every 5 10 Minutes
- Azure Backup Stored on Azure Blob Storage with replication types RA-GRS this means it is stored in two regions and the second region is read-only
- Azure Backup Encrypted by default by TDE
- Backup Retention period point in time restore from 7 to 35 days
- By default, the stander backup for Single Database, Elastic Pool, and the Managed instance is kept for 7 days and you can change the backup retention period to 35 days.
- Stander backup cannot be disabled
- Backup LTR Long-Term retention period the backup saved for up to 10 years, add in your note the Full backup only saved in this option of LTR
- LTR Backup option supported only Azure Single database and Azure Database SQL Elastic Pool
- For backup LTR you can add one or more Long-Term Retention period policy
- You can change the backup policy using the Azure portal or PowerShell Command.

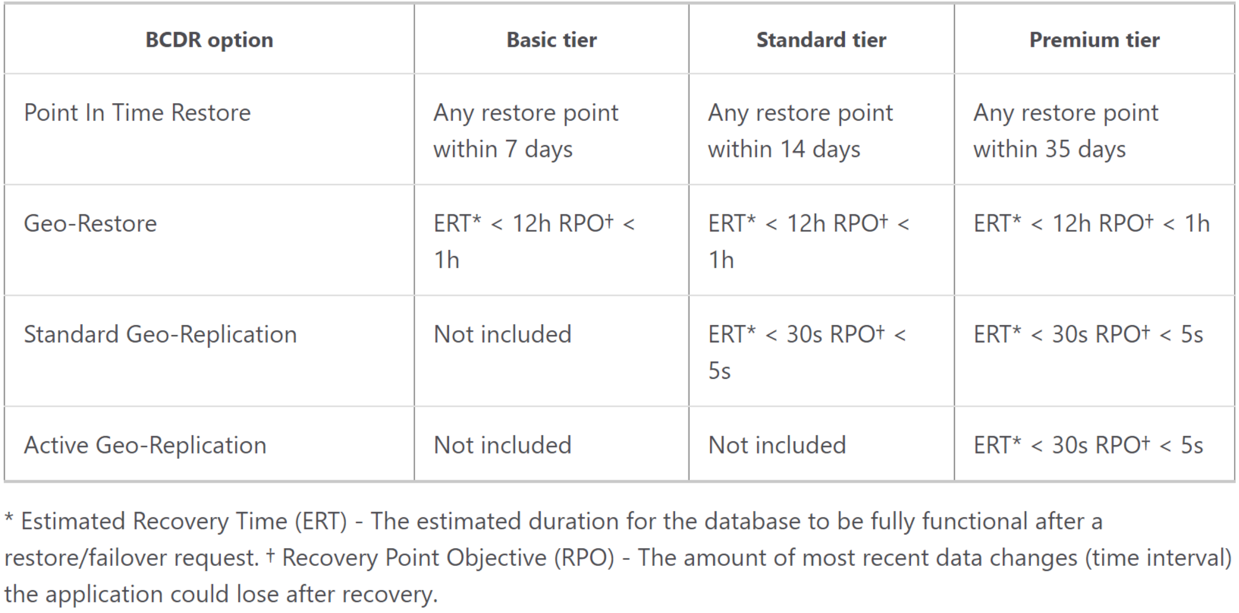
- IF you delete the Database you can restore it from Azure Backup but if you deleted the Azure logical Server it will delete the Database with it and you cannot restore the database at this time.
- You cannot restore the backup database with overwrite option this means the restore will create a new database.

References

- Manage Azure SQL Database long-term backup retention https://docs.microsoft.com/en-us/azure/sql-database/sql-database-long-term-retention & https://docs.microsoft.com/en-us/azure/sql-database/sql-database-long-term-backup-retention-configure
- Point-in-time restore by using Azure portal https://docs.microsoft.com/en-us/azure/sql-database/sql-database-recovery-using-backups#point-in-time-restore-by-using-azure-portal
- Understanding Azure SQL Database automated backups https://docs.microsoft.com/en-us/azure/sql-database/sql-database-automated-backups & https://docs.microsoft.com/en-us/azure/sql-database/sql-database-automated-backups#what-is-a-sql-database-backup
- Deleted database restore https://docs.microsoft.com/en-us/azure/sql-database/sql-database-recovery-using-backups#deleted-database-restore
- Geo-restore https://docs.microsoft.com/en-us/azure/sql-database/sql-database-recovery-using-backups#geo-restore
- Azure SQL Database: Built-in Backups vs Import/Export https://azure.microsoft.com/en-us/blog/azure-sql-database-built-in-backups-vs-importexport-2/

Azure SQL Restore

- **Restore Backup:** you cannot restore DB on Azure with an overwrite option. And in the restore process, you can change the Service tier of this new DB.
- **Restore Time:** In Azure, the restore time depends on UTC, not on your local time.
- **Azure Restore options:** Point-in-Time, Deleted Database Restore, Geo-Restore, Manual restore
- **Restore Point-In-Time-Restore PITR Tools:** You can restore a backup using Azure portal, Azure PowerShell command using command "**Restore-AzureRmSQLDatabase**" and you can restore the backup using DACPAC or BACPAC files, Azure CLI, Azure SDK
- **What is Geo-restore**: Geo-restore provides the ability to restore a database from a geo-redundant backup to create a new database. The database can be created on any server in any Azure region. Because it uses a geo-redundant backup as its source it can be used to recover a database even if the database is inaccessible due to an outage. Geo-restore is automatically enabled for all service tiers at no extra cost
- To recover DB by using automated backups, you must be a member of the SQL Server contributor role in the subscription
- To restore a database from the LTR storage (**Long-Term-backup-Retention**), you can select a specific backup based on its timestamp. The database can be restored to any existing server under the same subscription as the original database



BCDR option	Basic tier	Standard tier	Premium tier
Point In Time Restore	Any restore point within 7 days	Any restore point within 14 days	Any restore point within 35 days
Geo-Restore	ERT* < 12h RPO† < 1h	ERT* < 12h RPO† < 1h	ERT* < 12h RPO† < 1h
Standard Geo-Replication	Not included	ERT* < 30s RPO† < 5s	ERT* < 30s RPO† < 5s
Active Geo-Replication	Not included	Not included	ERT* < 30s RPO† < 5s

* Estimated Recovery Time (ERT) - The estimated duration for the database to be fully functional after a restore/failover request. † Recovery Point Objective (RPO) - The amount of most recent data changes (time interval) the application could lose after recovery.

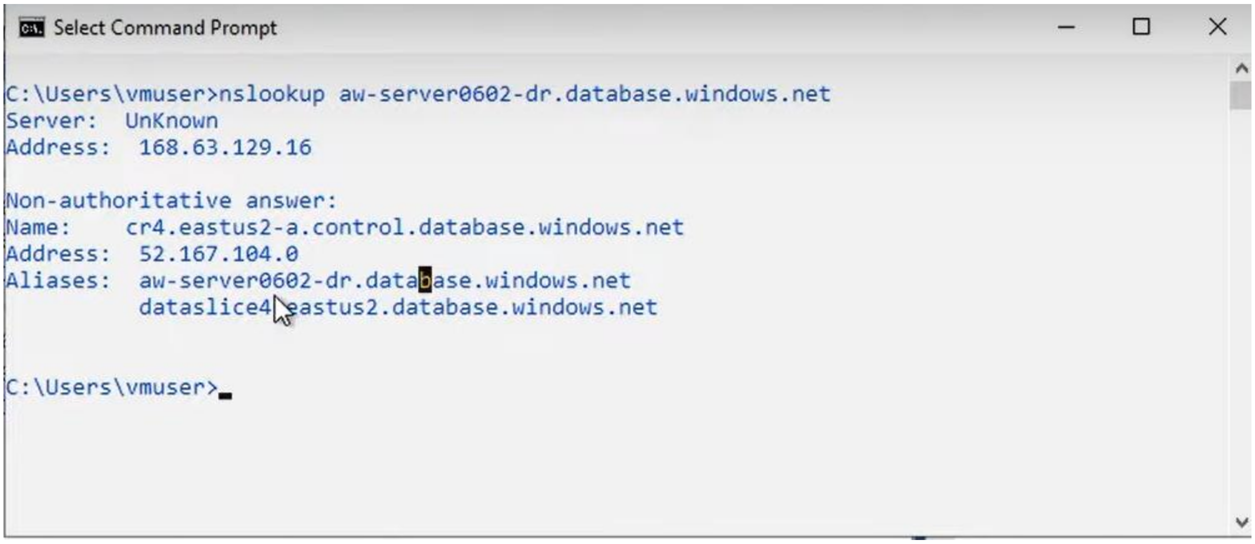
Azure SQL Security

- Advanced-Data Security (ADS) is a unified package for advanced SQL security capabilities. ADS is available for Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. It includes functionality for discovering and classifying sensitive data

1- Azure SQL Private link Benefits
  - Private endpoint for Azure SQL (server level)
  - On-premises connectivity via Express Route/VPN
  - Data exfiltration protection
  - Can be part of a network monitoring strategy using a Network Virtual Appliance (NVA)

2- Currently in public preview for:
  - Azure SQL Database (singleton databases)
  - Azure SQL Data Warehouse
  - Azure Storage

3- Azure SQL Network Options: https://docs.microsoft.com/en-us/learn/modules/azure-sql-secure-data/2-security-capabilities
  - **Allow access to Azure services:** Any Services in Azure can access the Azure SQL
  - **Firewall rules**: Server Firewall and Database Firewall Rule, setting this up can be complicated. It means that you'll have to specify a range of IP addresses for all your connections, which can sometimes have dynamic IP addresses, and while using the Firewall rule and do DNS Lookup on the Azure SQL instance name you will be able to see the Database Configuration (Region, Services Tier). But with Private links this information is hidden.



```
C:\Users\vmuser>nslookup aw-server0602-dr.database.windows.net
Server:  UnKnown
Address:  168.63.129.16

Non-authoritative answer:
Name:    cr4.eastus2-a.control.database.windows.net
Address:  52.167.104.0
Aliases:  aw-server0602-dr.database.windows.net
          dataslice4.eastus2.database.windows.net


C:\Users\vmuser>_
```

  -
  - **Virtual network rules:**

- **Private Link for an Azure SQL Database instance:** when you add a Private link you don't need to change the connection string still you have the option to connect with FQDN. So this change with ZERO development cost and without any downtime.
- SELECT client_net_address FROM sys.dm_exec_connections WHERE session_id=@@SPID;
-

4- Azure Role-Based Access Control (RBAC)

5- Authentication to Azure SQL:

  - "Mixed Mode" authentication forced and cannot be changed
  - SQL Auth login required during deployment called **server admin**.
  - Server-level principal for a logical server. Becomes default dbo for Azure SQL Database
  - Member of sysadmin server role for Managed Instance
  - CREATE LOGIN supported for both MI and Database
  - loginmanager and dbmanager roles for Azure Database admins
  - SA disabled for Managed Instance and guest disabled for database
  - logins for Azure SQL Database are created in the context of the logical server.
  - The server admin specified during database server deployment is a server-level principal and effectively has "sysadmin" rights for the server and all databases.
  - You can create other more limited admins using the following database-level roles in the master of the logical server.
  - loginmanager role is a database level role in the master of the logical server. Members are allowed to create logins for the database server
  - dbmanager is a database level role in the master of the logical server. Members can create and delete databases based on the database server.
  - CREATE LOGIN has some syntax differences for MI and DB (Ex. supporting Azure AD auth). See you at https://docs.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=azuresqldb-current and https://docs.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=azuresqldb-mi-current.

6- Dynamic Data Masking

7- Ensure applications force connection encryption

8- Evaluate the use of TDE

9- Take advantage of Dynamic Data Masking

10- Setup Always Encrypted for advanced protection

11- **Advanced Threat Protection**: SQL Threat Detection allows you to respond to unusual and harmful attempts to breach your database

- Detects potential SQL injection attacks
- Detects unusual access & data exfiltration activities
- Actionable alerts to investigate & remediate
- View alerts for your entire Azure tenant using Azure Security Center

  - **It is super simple to enable** and **requires no modifications to your application code.**
  - **It provides you with a set of world-class algorithms** that learn, profile, and detect **potential SQL injections** and **unusual behavior patterns**
  - It triggers **security alerts upon detection**, which include clear description and actionable investigation and remediation steps.

12- **Security Center**: is the main place that you need to look into it when you need to check the security of Azure SQL

- Azure SQL DB limited the access control on DB level or server level using the Azure server firewall and Azure Database Firewall and this can be implemented from the Azure portal or using T-SQL, And it is the first layer for accessing the DB on Azure, and the Firewall allows an IP address or range of an IP address and this is the first configuration you should do it after provisioning new DB on Azure because it is by default not enabled. When the APP connected to Azure DB first validation is Firewall configuration that's why it is the first layer for accessing the DB on Azure
- Server Firewall allow Clients to access all databases on Azure SQL DB, Server Firewall role stored in master DB, and it can be configured using Azure Portal, PowerShell, and T-SQL
- **Database Firewall** allows the clients to access particular databases on Azure SQL DB, And the rules stored on the DB level can be configured using T-SQL after configuring the first server firewall. To manage database firewall rules, you can use the **sp_set_database_firewall_rule** and **sp_delete_database_firewall_rule** system stored procedures in the database to which these firewall rules apply. You can use Azure REST API or Windows PowerShell to implement the same functionality. To view database firewall rules in a specific database, you can query its **sys.database_firewall_rules** system view
- There is can be only one server admin account on azure SQL DB.
- You can integrate your on-premises Windows AD to Azure AD using Azure AD Connect.
- There are 3 different ways to authenticate Active directory login
  - ✓ **Azure active directory password:** User name and password created and managed by azure active directory
  - ✓ **Azure active directory integrated**: User name managed by on-premises active directory and integrated with azure active directory
  - ✓ **Azure active directory universal with MFA**: User name managed by Azure active directory and used Multi-factor authentication and this will code received by SMS or by calling
- **Azure Server level administration roles**: Database Creator, Login Managers

We can secure our database using any one or all of 4 options (Network Security, Access Management, Threat Protection, information protection, and encryption) Tutorial: Secure a single or pooled database

→ https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-tutorial

- Network Security Server Firewall and Database Firewall and add Endpoint
- Access Management (SQL Account or Azure AD account)
- Authorization (RBAC Role-based Access Control) and RLS Row-level security
- Threat Protection (Data Discovery and Classification, Vulnerability assessment, advanced threat protection)
- Information protection (TLS Transparent Layer security, TDE Transparent Data Encryption, Always Encrypted, DDM Dynamic Data Masking
- **Network Security** Allow you to Create a Firewall rule to grant access to the DB for certain IPs or subnet and we have Server Firewall and Database Firewall and to enable Firewall settings on the database level you need to do it using T-SQL → https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-set-firewall-rule-azure-sql-database?view=azuresqldb-current
- Create a server-level firewall rule → https://docs.microsoft.com/en-us/azure/sql-database/sql-database-server-level-firewall-rule
- Configuring the Azure SQL Database Firewall → https://www.sqlshack.com/configuring-the-azure-sql-database-firewall/
- Azure SQL Database Firewall → https://rajbos.github.io/blog/2020/02/12/Azure-SQL-Database-Firewall
- **Access management** Azure SQL Database supported SQL Authentication User name and password, Supported Azure AD Authentication using identities in Azure Active directory, Supported Row-level Security to able to control the access per row in tables.
- Azure Identity Management and access control security best practices → https://docs.microsoft.com/en-us/azure/security/fundamentals/identity-management-best-practices
- Row Level Security in Azure SQL Database → https://azure.microsoft.com/en-us/resources/videos/row-level-security-in-azure-sql-database/
- Threat Protection Can be done using (SQL Auditing in Azure monitoring logs and Event Hub or advanced threat Protection)
- SQL Auditing in Azure monitoring logs and Event Hub using this feature will be able to tracks the activities of the database to help you in maintenance compliance → https://techcommunity.microsoft.com/t5/azure-sql-database/sql-audit-logs-in-azure-log-analytics-and-azure-event-hubs/ba-p/386242
- Advanced threat Protection using this feature you can analyze the Azure SQL Server database logs to detect unusual behavior → https://docs.microsoft.com/en-us/azure/sql-database/sql-database-threat-detection-overview

- information protection, and encryption: Connection in Azure is secured by TLS Transparent Layer security, TDE is Available on Azure SQL Database, Dynamic Data Masking Available on Azure SQL Database, Always Encrypted Available on Azure SQL Database and All of the Encryption Keys and stored in Azure Key Vault.

SQL Azure HA High Availability

1- **Auto and user-controlled BACKUP/RESTORE**

- Full Database backup once a week
- Log Backups every 5-10 minutes
- Differential Backups every 12 hours
- Backup files on Azure storage with RA-GRS replicated
- By default, SQL Database and SQL Managed Instance store data in geo-redundant (RA-GRS) storage blobs that are replicated to a paired region. This helps to protect against outages impacting backup storage in the primary region and allow you to restore your server to a different region in the event of a disaster
- When you are deploying Azure SQL or Azure SQL MI managed instance you can change the default geo-redundant backup storage redundancy and configure either Locally-redundant (LRS) or zone-redundant (ZRS) storage blobs for backups https://docs.microsoft.com/en-us/azure/azure-sql/database/automated-backups-overview?tabs=single-database and This feature is not yet available for Hyperscale tier
- Backup Integrity checks
- Restore to a new database
- Long-term retention (up to 10 years) of backups
- Current limits for LTR of backups for Managed Instance, Using long-term backup retention with an Azure SQL Database managed instances has the following limitations
  - ✓ **Limited public preview** - This preview is only available to EA and CSP subscriptions and is subject to limited availability
  - ✓ **PowerShell only** - There is currently no Azure portal support. LTR must be enabled using PowerShell https://docs.microsoft.com/en-us/azure/azure-sql/managed-instance/long-term-backup-retention-configure
- Geo-restore of databases if primary region down
- Restore backups of deleted databases
- You can not Restore backup on Top of the existing database, Restore meaning new Azure SQL Deployment.
- Manual COPY_ONLY Backup/Restore with SQL MI Managed Instance
-

2- **Built-in HADR and read replicas In General Purpose**

- Primary replica Tempdb data and log files located in SSD Disks (LRS) Local Redundant storage
- Failover decisions based on SQL and Service Fabric
- When Failover happens here in General Purpose, Azure will deploy a new instance Azure SQL or Azure SQL MI and it will point into your DB hosted on Azure Storage, then it will

Attach the SQL Database files, then it will run recovery and the getaway network connection will be updated automatically by the new node this failover doesn't need any change on APP level No virtual network or listener required

3- **Built-in HADR and read replicas In Business Critical**

- Based on Always-on Availability Groups
- 3 secondary replicas automatically created
- Four replicas kept available
- Backup files in a different location with geo-redundancy
- At least one secondary must sync for commits
- One secondary you can use it as Read-only for free
- Automatic failover based on SQL and Service Fabric
- Recovery time extremely fast
- Connectivity redirection built-in
- Read Scale-Out from one of the replicas
- Tempdb kept on local storage SSD not Like in General Purpose located in Azure Storage LRS. And the other databases Data and log files located In local SSD storage
- in Business-Critical Azure Build multiple secondaries with the same concept data and log located in local SSD Storage
- When Failover happens here Azure SQL will move the database to the sync Database secondary node
- Faster Failover with Business Critical

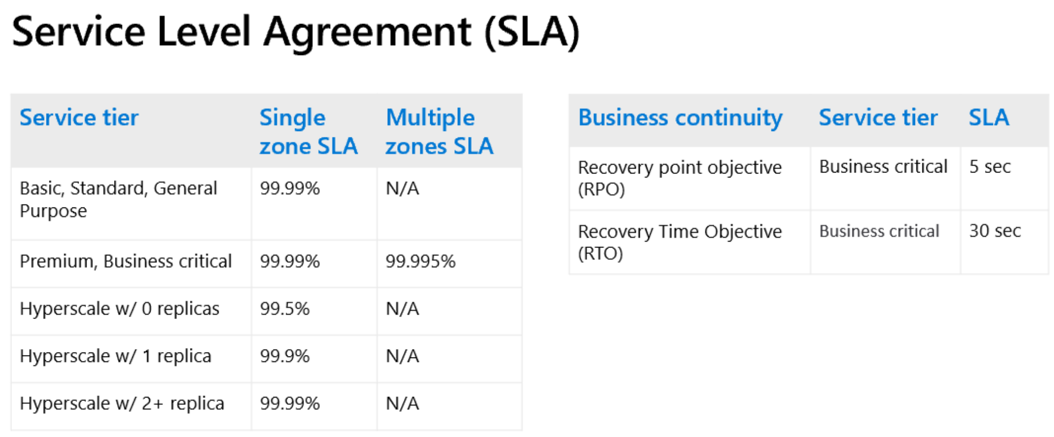
4- **Built-in HADR and read replicas in Hyperscale**

- Hyperscale only available in Azure SQL
- If you deploy your Azure SQL on Hyperscale you can not move it back to another service tier you need to migrate.
- HS Hyperscale Tier in SQL Single Database Support Local Storage + Remote Storage IOPS + Unlimited Storage + 100 TB Database Size
- HS Hyperscale Tier in SQL Single Database Support One Primary replica + 4 Secondary replica all of them are read-only and you can use Round Robin to distribute your read-only transaction on 4 replicas Read More about Round Robin from here: https://www.sqlshack.com/how-to-configure-read-only-routing-for-an-availability-group-in-sql-server-2016/.
- Hyperscale – 100TB Max Database Size
- Azure SQL Database Limited the database size to 4 TB but in Hyperscale you have a database size up to 100 TB, note that once an Azure SQL Database is converted to Hyperscale, you cannot convert it back to a “regular” Azure SQL Database
- Paired page servers
- Redundant log and data through Azure Storage

- Backup/Restore snapshots
- Log Service feeds replicas
- 0 to 4 secondary replicas for reading scale and failover
- Automatic failover based on SQL and Service Fabric
- Recovery time depends on the existence of replicas
- Azure SQL Hyperscale not used Always On Availability Groups it is used something called Log Service feeds replicas
- Log Service feeds replicas. Transactions can commit when the log service hardens changes to the Landing Zone. The consumption of changes by a Secondary Compute replica is not required to commit
- You are not required to have replicas for Hyperscale. If a replica does not exist, HA is still possible but it behaves more like General Purpose since a new primary replica must be deployed.
- Backup/Restore snapshots: Hyperscale data files are on Azure Storage and since we have levels of “caching” we can use snapshots to quickly backup and restore a database even if the size is large.

5- Availability Zones

- Some regions are made up of multiple data centers (think campuses)
- Availability Zone is or more physical data centers separated from others but close enough for connectivity
- Protect your data from “region” issues by using Zones
- Only available today for Premium and Business Critical
- Not supported for Managed Instance today
- Could slow down latency-sensitive OLTP applications
- Available in almost all regions



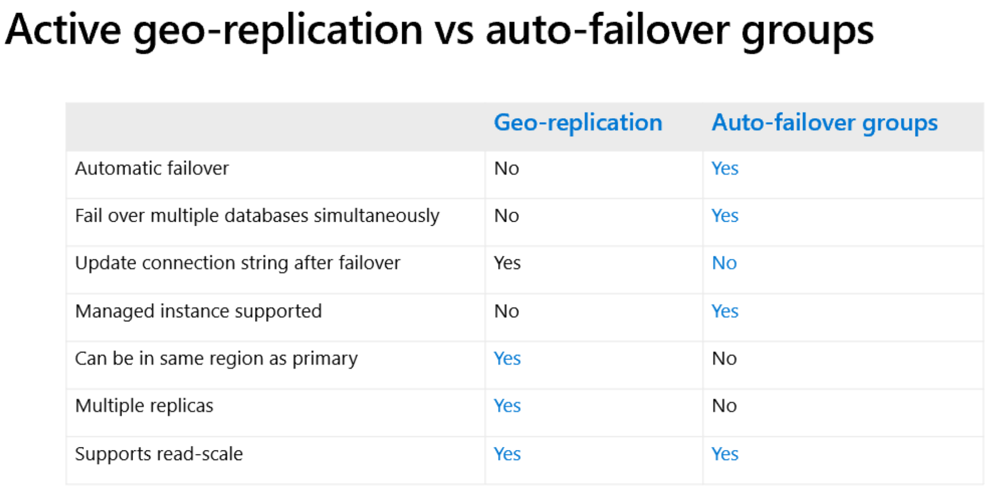
Service Level Agreement (SLA)

Service tier	Single zone SLA	Multiple zones SLA
Basic, Standard, General Purpose	99.99%	N/A
Premium, Business critical	99.99%	99.995%
Hyperscale w/ 0 replicas	99.5%	N/A
Hyperscale w/ 1 replica	99.9%	N/A
Hyperscale w/ 2+ replica	99.99%	N/A

Business continuity	Service tier	SLA
Recovery point objective (RPO)	Business critical	5 sec
Recovery Time Objective (RTO)	Business critical	30 sec

- https://azure.microsoft.com/support/legal/sla/sql-database/
- Only business continuity SLA in the industry, *RTO & RPO, 100% service credit*
- RPO stands for recovery **POINT** objective, i.e., how much data is one potentially prepared and willing to lose, worst case
- RTO stands for recovery **TIME** objective, i.e., if/when the ‘bad thing’ happens, how much time does it take to be back up and running again
- Azure is the only cloud provider, give you business continuity SLA in the industry, *RTO & RPO, 100% service credit*
-

6- Geo-replication and Failover Groups



Active geo-replication vs auto-failover groups

	Geo-replication	Auto-failover groups
Automatic failover	No	Yes
Fail over multiple databases simultaneously	No	Yes
Update connection string after failover	Yes	No
Managed instance supported	No	Yes
Can be in same region as primary	Yes	No
Multiple replicas	Yes	No
Supports read-scale	Yes	Yes

-
- In Geo-Replication the relation between primary and secondary is ASYNC relation so before you are doing manual failover you should change this relation to SYNC to avoid data loss
- **Azure DB Geo-Replication**: Azure Services used for replicating the data in multiple regions and it is available on all Services Tiers (basic, Stander, Premium) and it provides readable replica secondary Database in a different region, it is allowed to replica in DB Size, and it is allowed to scale up the secondary Database. Finally, you can Expect the ERT (**Estimated Recovery Time**) < 30 Second and RPO (**Recovery point objective**) < 1 Hour and it can be used for Read-only Scale or in Failover in case of a disaster
- **Geo-Replication Mode: Stander Option** is a legacy option provide non-Readable DB Replicas so you will have the option to failover but you can read from the other nodes and this feature ended in April 2017, **ACTIVE Option** Recommended option it is providing 4 readable databases replicas and it provides 30 Second as RTO and 5 seconds as RPO

- **Geo-Replication Synchronization mode:** this service does not provide synchronizes replication this meaning you have the option of data loss but to avoid this risk you can configure session wait until the log has been replicated to all of the replicas node using system Stored procedure "**SP_Wait_For_Database_Copy_Sync**".
- **Geo-Replication Permission**: the user should have at least permission as DBManger on both of primary server and Secondary Server
- **Geo-Replication Monitor**: Azure provide us 3 DMVs we can use it for this purpose (Sys.dm_Operation_Status / Sys.geo_replication_links / sys.dm_geo_replication_links_status)
- **Geo-replication Price:** You will pay full price on each Geo-Replication Secondary
- **Geo-Replication Setup and Configuration**: Can be done through Azure Portal or by T-SQL Command
- Active Geo-Replication is available for Basic, Standard, Premium, and Premium RS databases. It's designed for write-intensive applications with the most aggressive recovery requirements. Using Active Geo-Replication, you can create up to four readable secondaries on servers in different regions. You can initiate a failover to any of the secondaries. Also, Active Geo-Replication can be used to support the application upgrade or relocation scenarios, as well as load balancing for read-only workloads. Refer to Design for business continuity for details on how to configure Geo-Replication and to Recover from an outage for details of how to failover to the secondary database. Refer to Application upgrade without downtime for details on how to implement the application upgrade without downtime.
- The Active Geo-Replication feature implements a mechanism to provide database redundancy within the same Microsoft Azure region or in different regions (geo-redundancy). Active Geo-Replication asynchronously replicates committed transactions from a database to up to four copies of the database on different servers. The original database becomes the primary database of the continuous copy. Each continuous copy is referred to as an online secondary database. The primary database asynchronously replicates committed transactions to each of the online secondary databases. While at any given point, the online secondary data might be slightly behind the primary database, the online secondary data is guaranteed to always be transactionally consistent with changes committed to the primary database. Active Geo-Replication supports up to four online secondaries, or up to three online secondaries and one offline secondary.
- One of the primary benefits of Active Geo-Replication is that it provides a database-level disaster recovery solution. Using Active Geo-Replication, you can configure a user database in the Premium service tier to replicate transactions to databases on different Microsoft Azure SQL Database servers within the same or different regions. Cross-region

redundancy enables applications to recover from a permanent loss of a datacenter caused by natural disasters, catastrophic human errors, or malicious acts.

- Another key benefit is that online secondary databases are readable. Therefore, an online secondary can act as a load balancer for reading workloads such as reporting. While you can create an online secondary in a different region for disaster recovery, you could also have an online secondary in the same region on a different server. Both online secondary databases can be used to balance read-only workloads serving clients distributed across several regions.
- Other scenarios where Active Geo-Replication can be used include Database migration: You can use Active Geo-Replication to migrate a database from one server to another online with minimum downtime, or Application upgrades: You can use the online secondary as a fallback option
- To achieve real business continuity, adding redundancy between datacenters to relational storage is only part of the solution. Recovering an application (service) end-to-end after a disastrous failure requires recovery of all components that constitute the service and any dependent services. Examples of these components include the client software (for example, a browser with a custom JavaScript), web front ends, storage, and DNS. All components must be resilient to the same failures and become available within the recovery time objective (RTO) of your application. Therefore, you need to identify all dependent services and understand the guarantees and capabilities they provide. Then, you must take adequate steps to ensure that your service functions during the failover of the services on which it depends. For more information about designing solutions for disaster recovery, see Designing Cloud Solutions for Disaster Recovery Using Active Geo-Replication.
-

7- Database Availability

- You cannot set OFFLINE and EMERGENCY
- RESTRICTED_USER access allowed: RESTRICTED_USER allows for only members of the db_owner fixed database role and dbcreator and sysadmin fixed server roles to connect to the database but does not limit their number
- RESTRICTED_USER allowed for Azure SQL Databases but not MI
- SINGLE_USER is also not allowed for DB and MI
- Dedicated Admin Connection (DAC) allowed
- Accelerated Database Recovery on by default ADR
- Users cannot disable ADR Accelerated Database Recovery or we could not meet SLAs.
- Accelerated Database Recovery (ADR): it is a new Feature in SQL Server 2019 and it is on by default in **#Azure** SQL and the Users cannot disable ADR Accelerated Database Recovery or we could not meet SLAs, it is Uses a Persisted Version Store (PVS), with ADR

Rollback faster than you can react and Undo recovery faster than you can look it up
More information about
•Does it requires more space?
•Will it affects performance?
•Will I still see versions in tempdb?
•How does it work with HA?
Check this white paper {Constant Time Recovery in Azure SQL Database} that is created by the Microsoft engine team https://www.microsoft.com/en-us/research/uploads/prod/2019/06/p700-antonopoulos.pdf

8- **Database Consistency**

- Multiple copies of data and backups
- Users can execute DBCC CHECKDB (no repair)
- Database CHECKSUM on by default
- Auto Page Repair when possible
- Data integrity error alert monitoring
- Backup and restore integrity checks
- “lost write” and “stale read” detection
- Repair without notification if no impact
- Proactive notification to customers

9- **Which Option is the best for AZURE HA?**

- Decide if you need long-term backups
- Decide on your RTO and RPO needs
- Review the Azure SQL SLA
- Do you need to read replicas?
- Do you need Availability Zones?
- Do you need geo HADR or Failover Groups?

10- **Replication**

11- https://mostafaelmasry.com/2020/04/06/azure-active-geo-replication-services/

12- Azure Keep 3 local high Availability copies in the same server region without any extra cost and when the Primary Failed for any reason the DB will failover to the secondary server and when it is up the Azure will create another secondary server and add it to quorum-set all of this configuration done by azure automatically don’t worry about it

13- You can restore the DB in the same region or another region in case of a disaster

14- **Azure DB Geo-Replication**: Azure Services used for replicating the data in multiple regions and it is available on all Services Tiers (basic, Stander, Premium) and it provides readable replica secondary Database in a different region, it is allowed to replica in DB Size, and it is allowed to scale up the secondary Database. Finally, you can Expect the ERT (**Estimated Recovery Time**) <

30 Second and RPO (**Recovery point objective**) < 1 Hour and it can be used for Read-only Scale or in Failover in case of a disaster

15- **Geo-Replication Mode: Stander Option** is a legacy option provide non-Readable DB Replicas so you will have the option to failover but you can read from the other nodes and this feature ended in April 2017, **ACTIVE Option** Recommended option it is providing 4 readable databases replicas and it provides 30 Second as RTO and 5 seconds as RPO

16- **Geo-Replication Synchronization mode:** this service does not provide synchronizes replication this meaning you have the option of data loss but to avoid this risk you can configure session wait until the log has been replicated to all of the replicas node using system Stored procedure "**SP_Wait_For_Database_Copy_Sync**".

17- **Geo-Replication Permission**: the user should have at least permission as DBManger on both of primary server and Secondary Server

18- **Geo-Replication Monitor**: Azure provide us 3 DMVs we can use it for this purpose (Sys.dm_Operation_Status / Sys.geo_replication_links / sys.dm_geo_replication_links_status)

19- **Geo-replication Price:** You will pay full price on each Geo-Replication Secondary

20- **Geo-Replication Setup and Configuration**: Can be done through Azure Portal or by T-SQL Command

21- **Failover the DB:** we can do it using the T-SQL command (**Alter Database XX Failover)** but this command should be executed on the Secondary replica that you need to failover for it not on the primary server

Azure SQL Performance

1- Configuring and Maintaining for Performance
2- Monitoring and Troubleshooting Performance
3- Performance Scenarios
4- Accelerating and Tuning Performance
5- Max Capacities
    - Azure SQL Database - Up to 128 vCores and 4TB Memory and 4TB Database (data)
    - Hyperscale – 100TB Max Database Size
    - Managed Instance – Up to 80 vCores, 400GB Memory, and 8TB Database (data)
    - Use **sys.dm_os_job_object** for match *true* capacities.
6- Indexes and Statistics
    - All index types are supported across Azure SQL
    - Online and resumable indexes fully supported
    - Columnstore Indexes are available in almost all tiers.
    - REBUILD and REORG fully supported for DB and MI
    - Automatic Stats supported for DB and MI just like SQL Server
    - Maintenance Plans in SSMS are not available in DB or MI
7- In-Memory OLTP
    - Available in Business-Critical Tiers
    - Memory-Optimized FILEGROUP created with database creation
    - Max memory a ration of overall memory limit
8- Partitions
    - Supported for both Azure SQL Database and Managed Instance
    - Placement on filegroups only supported for Managed Instance
9- Many of the performance enhancements in SQL Server 2019 are available in Azure SQL. Check the documentation for each option. https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-ver15?view=sql-server-ver15
10- Configuring Tempdb
    - Always kept on local SSD drives so I/O perf shouldn't be an issue For Azure DB (vCores), we scale # files with vCores (2vcores=4 files,...) max of 16.
    - You get 12 files with MI independent of vCores We are looking to allow user config for this in the future.
    - MIXED_PAGE_ALLOCATION IS OFF and AUTOGROW_ALL_FILES is ON for tempdb
    - Tempdb Metadata Optimization not supported
11- Database Configuration:
    - An only full recovery supported so minimal logging for bulk operations not possible
12- Configuring Files and Filegroups

- MI supports adding files and sizes but not physical placement # files and file size can be used to tune I/O performance. Read more here. https://techcommunity.microsoft.com/t5/datacat/storage-performance-best-practices-and-considerations-for-azure/ba-p/305525
- User-defined FILEGROUP not supported but you can't physically place files anyway

13- Configuring MAXDOP

- ALTER DATABASE SCOPED CONFIGURATION for DB and MI
- sp_configure supported for MI
- Query hints allowed
- MI supports RG (Resource Governor)

14- Server Configuration: Optimized for ad hoc workloads supported in MI but not in DB

15- Monitoring and Troubleshooting Performance

- Azure Monitor Metrics and Logs
- Dynamic Management Views **DMV**
- Query Store on by default
- Extended Events
  - ✓ Azure SQL DB supports file (Azure Blob Storage), ring_buffer, and counter targets. File targets stored in Azure Blob Storage
  - ✓ Azure MI supports all targets of SQL Server
  - ✓ Azure SQL DB supports a subset of SQL Server events plus Azure specific events
  - ✓ Azure SQL MI supports all SQL Server events plus other Azure specific events
- As of 2/23/2020 Azure SQL MI supports 2400+ events. SQL Server 2019 has 1800+ events, Azure SQL DB supports 392 events
- Azure SQL DB has 202 DMVs
- Azure SQL MI has 323 DMVs
- SQL Server 2019 has 273 DMVs

16- Most important Dynamic Management Views DMV

- Azure SQL Managed Instance
  - ✓ All SQL Server DMVs available
  - ✓ sys.server_resource_stats
  - ✓ sys.dm_io_virtual_file_stats
  - ✓ sys.dm_os_performance_counters
  - ✓ sys.dm_instance_resource_governance
  - ✓ sys.dm_user_db_resource_governance
- Azure SQL Database
  - ✓ Common SQL Server DMVs available
  - ✓ sys.dm_db_resource_stats
  - ✓ sys.elastic_pool_resource_stats

- ✓ sys.dm_user_db_resource_governance_internal
- ✓ sys.dm_resource_governor_resource_pools_history_ex
- ✓ sys.dm_resource_governor_workload_groups_history_ex

17- Query performance insight

18- **Automatic Tuning** (Force Plan, Create Index, Drop Index) and it is a combination between Query Store and DMV so take benefits from this feature you should keep the Query store enable, and it is enabled by default

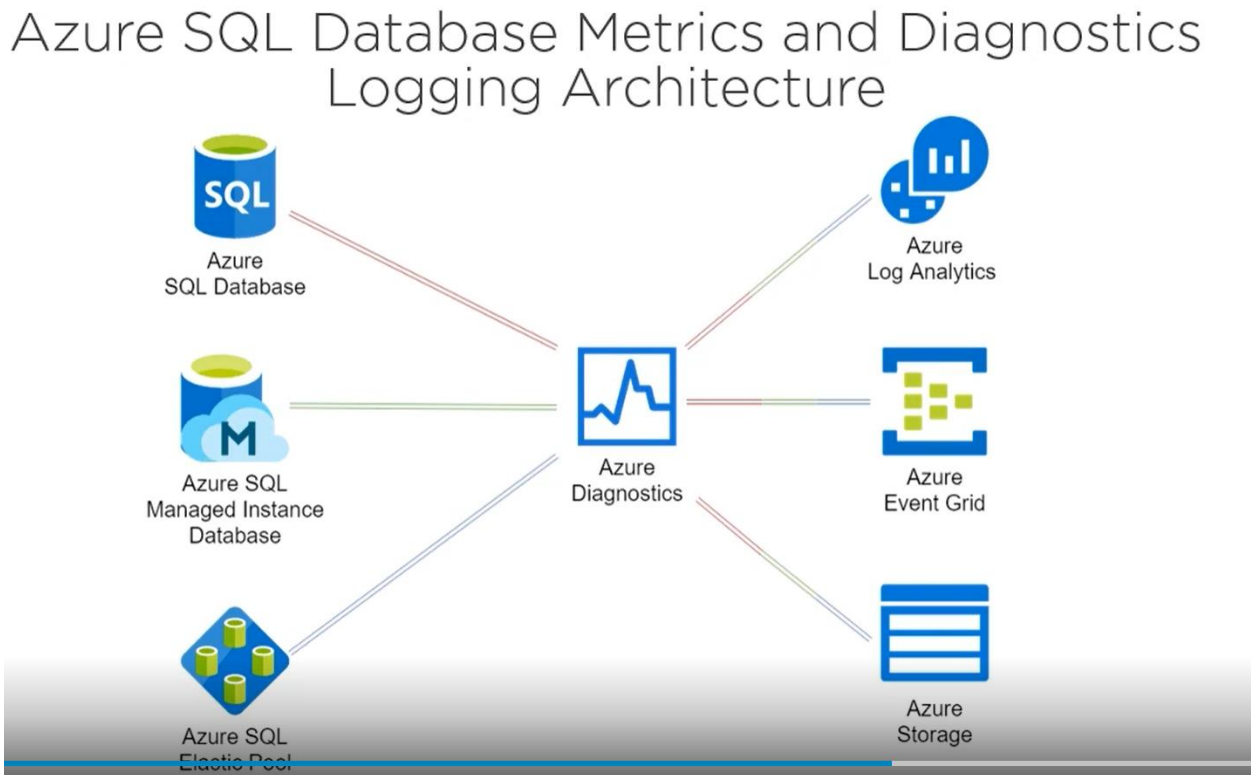
19- **Automatic Plan Correction:** it can help the query plan regressions that may be caused by parameter sensitive plans (PSP)

20- **IQP intelligent Query Processing:** it is a built-in Capability in the query processor enabled through database compatibility levels

- You can configure and manage metrics and diagnostics logging by using one of the following methods, Azure Portal, PowerShell, Azure CLI, Azure Monitor REST API, and Azure Resource Manager template.
- The metrics are collected every minute and are retained by the platform for 93 days.
- If you need to maintain access to them for an extended period, you have a few options such as sending them to a Log Analytics workspace or streaming them to an event hub, or archiving them to a storage account. The same options are available for the storage of diagnostics logs, which provide visibility to a variety of aspects of database operations, such as blocks, deadlocks, timeouts, and errors.
- **Connectivity Monitoring from Azure** (Connection Blocked by a firewall, Failed Connection, Successful Connection)
- **Performance Monitoring from Azure (**DTU Percentage, Limit and Usage, CPU Percentage, Log, and Data IO Percentage**)**
- **Database Monitoring from Azure (**Deadlocks, Session Percentage, DB Size, Worker Percentage, In-Memory OLTP Storage Percentage**)**
- **We can Define Alerts on Azure SQL DB** using Portal or PowerShell in Portal there is an option called **Alert Rule** under the DB and this option Is a replacement of DB Mail.
- **Perfmon tools** are not available on Azure SQL DB because we don't have access to the OS operating system, but we can use Performance system views (Resource Usage Views, Waite Stats view)
- **Resource Usage Views**: Will track the (AVG CPU%, AVG Data IO, and Log Writer %, AVG Memory %)
- **Waite Stats view**: Used to track the wait types and the Statistics for each DB (**Sys.db_os_Wait_stats**)
- **Azure performance tool (performance Recommendation tool):** using this tool it will give you a recommendation for your DB based on analyzing the Db workload.

- **Azure performance tool (Query Performance insight):** Used to browse the performance status of the query running on Azure DB like Long-running Query, Query with high CPU, and High IO it like activity monitor on SQL Server management studio but with advanced option
- **Azure performance tool (Automatic Tuning):** Azure tool used to create sufficient index (Missing Index) that is highly needed by Azure SQL based on the DB Workload and Execution plan, also this tool we can use it to drop the index that is not used by SQL Engine.
- **Azure performance tool (Query Store):** It is a feature on Azure SQL DB and Azure on Premise starting from SQL Server 2016 and it will give you the availability to see the Execution plan history for the Query and you can compare them also you can force the most suitable execution plan for your query

Resource Usage Views	**Scope**	**Measures**	**History**	**When it is work**
Sys.resource_Stats	Server level	5 Minute	Keep data for 14 days	Capture working when resource usage changes
Sys.dm_db_resource_stats	DB Level	15 second	Keep data for 1 hour	Capture even with no activity



Azure SQL Database Metrics and Diagnostics Logging Architecture

- **Query performance insight**: feature available in Azure SQL Database that helps you understand the impact of queries running against your database. Query Performance Insight depends on Query Store to be able to manage the data performance, and by default Query store feature is enabled on Azure SQL Database. By using Query performance insight you will have visibility on some charts and reports like Top CPU query, TOP query by duration, top query by execution count. And you can deep dive into a single query
- **Azure SQL Performance Recommendation Services**: Azure Feature enable by default on Azure SQL DB and it is a feature based on SQL Database Adviser and it is available on Azure SQL single DB or elastic pooled database Using this feature, we can take recommendation by Creating missing index automatically by azure and removing unused index automatically by azure, also recommendation advisor it will give you parametrize queries recommendations. Dropping unused indexes is not available for Premium and Business Critical service tiers.
- **Azure SQL Analytics**: monitoring Solution that allows you to collect important data related to the performance of your Azure SQL Database in a single place. And It is based on Log Analytics services. Azure SQL analytics allows you to create custom rules and alerts with specific metrics. And as proactive action, you can identify issues in your database. Azure SQL Analytics depends on Diagnostics settings so you should enable the Diagnostics settings first on your Azure SQL database. Storing the data in Log Analytics while creating managing actions in Diagnostics

settings. Azure SQL Analytics will provide you an amazing graphical user interface that can be used while troubleshooting any performance issue.

- **Azure SQL Automatic Tuning Services**: This fully monitoring services from azure it is doing monitoring for SQL Server continuously to be analyzed by Built-in intelligence and generate performance recommendations. One of the most amazing features in these services that after applying the recommendation for the tuning services it will be monitored by the services and if the services find this recommendation not adding value for Azure DB performance it will be rollback automatically. all of these actions will be recorded in the logs to be able to track it. This feature is provided with full support in Azure SQL Single Database and Azure SQL Pooled Database. This feature also available on SQL Server 2017 version and above but there is a difference here in SQL Server 2017 or above Automatic tuning used to force query plan and it is depending also on the query store. This means you should enable query store first on your database and keep it running for some hours to take the benefits from SQL Automatic tuning features
- **Diagnostics settings:** Enabling this feature on SQL Server for collecting Some logs based on some matrices and sending it to (Log Analytics, Saving it on Azure Storage account, Stream it on Event Hub) then we can build our alerts on it with creating managing actions that we can use to send an alert by email, SMS, alert to the logic app like doing tweet on Twitter, Executing Azure Runbook) many of options we can use it on managing actions.

21-

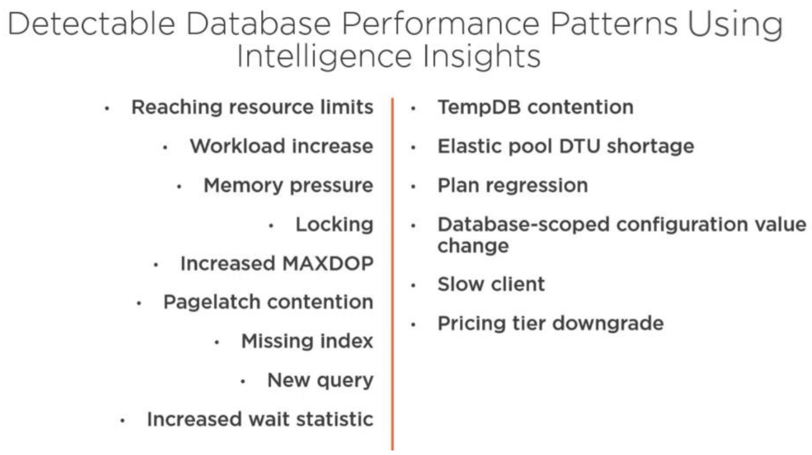
Query performance insight

Query Performance Insight is a feature available in Azure SQL Database that helps you understand the impact of queries running against your database and how they are related to the consumption of resources allocated to your database or database servers. This feature is recommended in scenarios of basic performance tuning and troubleshooting. You can access this feature directly from the Azure portal and have access to the detailed data exposing the performance of individual queries, and in case it is available, the corresponding performance recommendations to fix a given performance issue provided by SQL Database Advisor. Query Performance Insight relies on Query Store to manage performance data. By default, Query Store is enabled for Azure SQL Database, but you can enable it anytime if it is not running in your database. For you to view the information in Query Performance Insight, Query Store needs to capture a few hours of data. If your database has no activity or if the Query Store is not active during a certain period, the charts will not show any information during that time range. Query Performance Insight provides you an extensive set of features to support your analysis, such as review top CPU-consuming queries, review top queries per duration, review top queries per execution count, view individual query details, and understand the performance tuning annotations available.

Azure SQL Analytics

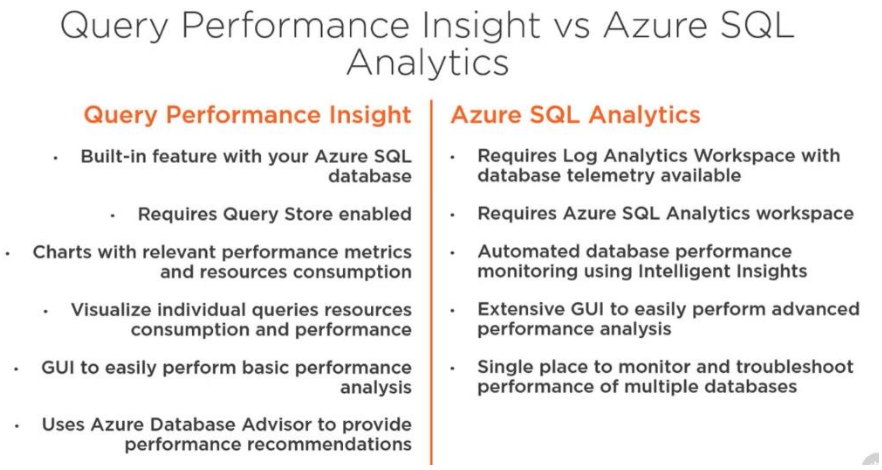
https://www.youtube.com/watch?v=4BDnc-ePtMs&feature=youtu.be

Azure SQL Analytics solution is the solution for advanced monitoring scenarios that allows you to collect and visualize important data related to the performance of your Azure SQL Database, elastic pools, and managed instances in a single place. It is based on Log Analytics, so it relies on the Log Analytics workspace available to manage all metrics and logs. It also allows you to leverage its functionality and create custom rules and alerts associated with the metrics collected, so you can proactively identify issues in your database and applications. To configure the type of data to show in Azure SQL Analytics, you need to configure the diagnostic settings of your Azure SQL Database and store the data in your Log Analytics workspace. The solution relies on the built-in intelligence to present relevant data and charts related to your Azure SQL Database performance, including database errors, timeouts, query durations, and query waits. Intelligent Insight is an artificial intelligence-based solution that provides continuous database monitoring and relies on collecting SQLInsights logs. You can enable these logs by configuring their diagnostic settings and designating one or more destinations as the persistent store for log data, which can then be analyzed and reviewed. Intelligent Insights is capable of detecting temporary deviations from a long-term workload baseline, evaluating their impact, performing root cause analysis, and providing remediation options. Intelligent Insights can identify many database performance issues based on the following patterns. When a database is reaching resource limits. When there is an unusual workload increase. When there is unusual memory pressure. When locks are happening in your database. When there is an increased MAXDOP. When there is a contention of page latch. When there is an index missing in your tables. When new queries are executed against your database. When there is a significant increase in wait statistics. Also, when there is the contention of TempDB. When you have a shortage of elastic pool DTUs. When you have plan regressions. When a database-scoped configuration value changes. When you have a slow client querying your database and taking too long to retrieve the data. And also, when there is a pricing tier downgrade that degrades the database performance. In this diagram, we have the architecture of database telemetry in Azure. As the source, we have Azure SQL Database, SQL managed instance, or Elastic Pool. Then we have Azure Diagnostics that enables us to configure the metrics and logs to collect and store them in the following target options, Log Analytics, Event Grid, or Storage. It is important to note that each link from each source to Azure Diagnostics has a different color, and the links going from Azure Diagnostics to Log Analytics has all the colors combined. By this, it means Azure Diagnostics can group the metrics and logs from multiple sources and send them to a specific target. Now that we have explored Query Performance Insight and Azure SQL Analytics as options to monitor and troubleshoot the performance of your database, it is important to understand some of the differences between them and what is the best option for the different scenarios. Starting with the Query Performance Insight, it is a built-in feature that comes with your Azure SQL Database and is based on the Query Store, so it requires the Query Store enabled. Also, it provides you many charts with relevant performance metrics and resource consumption. You are also able to visualize an individual query's resource consumption and performance. It provides you an amazing graphical user interface that allows you to perform your basic performance analysis and use Azure Database Advisor to provide you performance recommendations to be applied to your database. Moving next to the Azure SQL Analytics, it is a solution based on Log Analytics, so it requires Log Analytics workspace with database telemetry available. It also requires an Azure SQL Analytics workspace so then you can use the solution. You can leverage the automated database performance monitoring using Intelligent Insights, and it also provides an amazing graphical user interface that allows you to easily perform advanced database performance analysis, and it provides you a single place to monitor and troubleshoot the performance of multiple databases.



Detectable Database Performance Patterns Using Intelligence Insights

- Reaching resource limits
- Workload increase
- Memory pressure
- Locking
- Increased MAXDOP
- Pagelatch contention
- Missing index
- New query
- Increased wait statistic
- TempDB contention
- Elastic pool DTU shortage
- Plan regression
- Database-scoped configuration value change
- Slow client
- Pricing tier downgrade

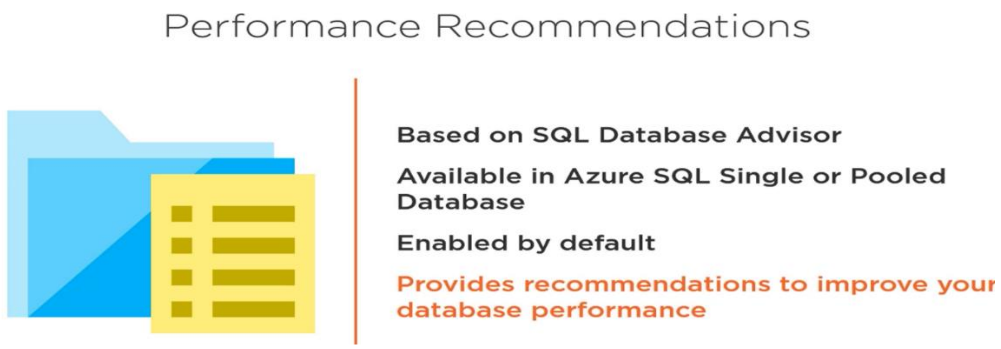


Query Performance Insight vs Azure SQL Analytics

Query Performance Insight	Azure SQL Analytics
Built-in feature with your Azure SQL database	Requires Log Analytics Workspace with database telemetry available
Requires Query Store enabled	Requires Azure SQL Analytics workspace
Charts with relevant performance metrics and resources consumption	Automated database performance monitoring using Intelligent Insights
Visualize individual queries resources consumption and performance	Extensive GUI to easily perform advanced performance analysis
GUI to easily perform basic performance analysis	Single place to monitor and troubleshoot performance of multiple databases
Uses Azure Database Advisor to provide performance recommendations	

Azure SQL Performance Recommendation Services

- ✓ Performance Recommendations is a feature based on SQL Database Adviser and available in your Azure SQL single or pooled database.
- ✓ This feature is enabled by default and continuously assesses and analyzes the usage of your database and provides recommendations when it detects ways to improve the database performance.
- ✓ Performance Recommendations has an extensive set of capabilities to identify performance issues in your database and provide recommendations.
- ✓ The databases are monitored and assessed to identify issues by predefined patterns. We have the following group of recommendations provided.
- ✓ Create index recommendations. This group identifies indexes missing in the tables used in the queries that run against your database and then recommends the creation of the missing index.
- ✓ Also, we have the drop index recommendations. This group identifies indexes that are compromising somehow the performance of your database by identifying the tables being used by the queries running against your database and then identifying the index available on each of the tables. Then, it assesses if the index is not used for a long time, more than 90 days, or if it's duplicated, and if that's the case, then it recommends you to drop the index.
- ✓ This recommendation is not compatible with scenarios of using partitioned switching or index hints. Also, dropping unused indexes is not available for Premium and Business Critical service tiers.
- ✓ Then, we have the parametrize queries recommendations. And these recommendations are applied in scenarios that you have queries running against your database, constantly being recompiled, and using the same query execution plan. In these scenarios, you will receive recommendations to parametrize your queries so they are not recompiled constantly.
- ✓ Also, we have fixed schema issues recommendations. Yes, I said we had because this option is being deprecated by the Azure team since Intelligent Insights already provides you this option and more.
- ✓ Last, you can also use Performance Recommendations in your applications by using its API.
- ✓ There are PowerShell cmdlets available that use the Azure SQL Database Advisor API to allow them to manage performance recommendations.



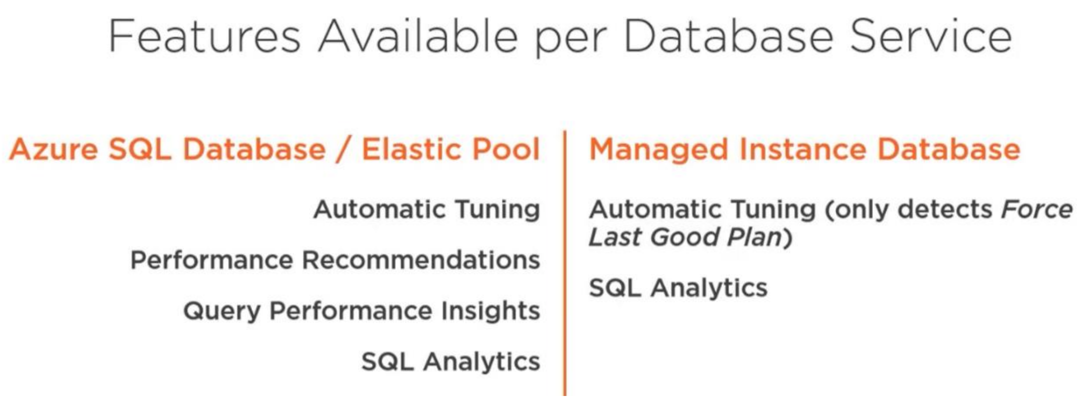
Performance Recommendations

Based on SQL Database Advisor

Available in Azure SQL Single or Pooled Database

Enabled by default

Provides recommendations to improve your database performance



Features Available per Database Service

Azure SQL Database / Elastic Pool	Managed Instance Database
Automatic Tuning	Automatic Tuning (only detects *Force Last Good Plan*)
Performance Recommendations	SQL Analytics
Query Performance Insights	
SQL Analytics	

Azure SQL Automatic Tuning Services

Automatic tuning is a fully managed service available in your Azure SQL Database. The feature continuously monitors your Azure SQL Database and collects observations to be analyzed by the built-in intelligence and generate performance recommendations. Also, it leverages artificial intelligence to learn from all SQL databases on Azure and provides accurate recommendations. This feature also provides you continuous performance tuning while averaging the built-in artificial intelligence and machine learning to ensure the best performance and stable workloads in your database through continuous performance tuning. Automatic tuning provides you an extensive set of capabilities to ensure your database achieves the best performance. One of the capabilities is the automated performance tuning that continuously monitors and assesses the performance of your database including actively verifies the performance gains from the recommendations applied, and in case any of these recommendations are not resulting in any gains, the feature can roll back the recommendation applied and self-correct it. All these options performed by this feature are recorded, and it allows you access to the tuning history to view the history of actions. To ensure that your database achieves the best performance, this feature continuously monitors the performance of the workloads running in your database, and sometimes depending on the change of workloads the recommendations applied previously may no longer make sense since they are no longer reflecting any gains to your database, and in this case, the feature can adapt to those changes and apply new or refer to existing recommendations previously applied to fix the issue. It is important to understand the different Azure services hosting options that provide this feature. This feature is provided with full support in Azure SQL Single Database and Azure SQL Pooled Database. It is important also to mention that you can always create your virtual machine with SQL Server 2017 or above installed, also the database there, and enable automatic tuning since this feature is also available in SQL Server 2017 or newer versions. But there is a difference when we plan to use Azure Managed Instance database since this service only supports partially this feature since it only corrects the query plan and is forced to use the last good plan, so any actions related to the creation of a missing index or dropping duplicated or unused indexes are not supported in this service.

Azure SQL Elastic Database Jobs

- It is Azure Service used to execute a custom job or many jobs in Azure SQL Databases
- Differences between Azure SQL Database Jobs and SQL Agent
  - ✓ Azure SQL Agent used to execute custom jobs on one database or multiple databases on the same SQL Server instance and we can configure it using T-SQL or SQL Server management studio SSMS
  - ✓ Azure SQL Elastic Database Jobs used to execute custom jobs in different database Servers or Subscription or Rejoins and it can target a Single database, Elastic Pool Database, Data Warehouse and to configure it we can use Azure Portal, PowerShell, T-SQL, or Azure Resource Manager
- Azure SQL Elastic Database Jobs supported Single database or Elastic Pool Database but SQL Agent Supported Azure SQL Managed database.

Azure SQL Elastic Database jobs Component

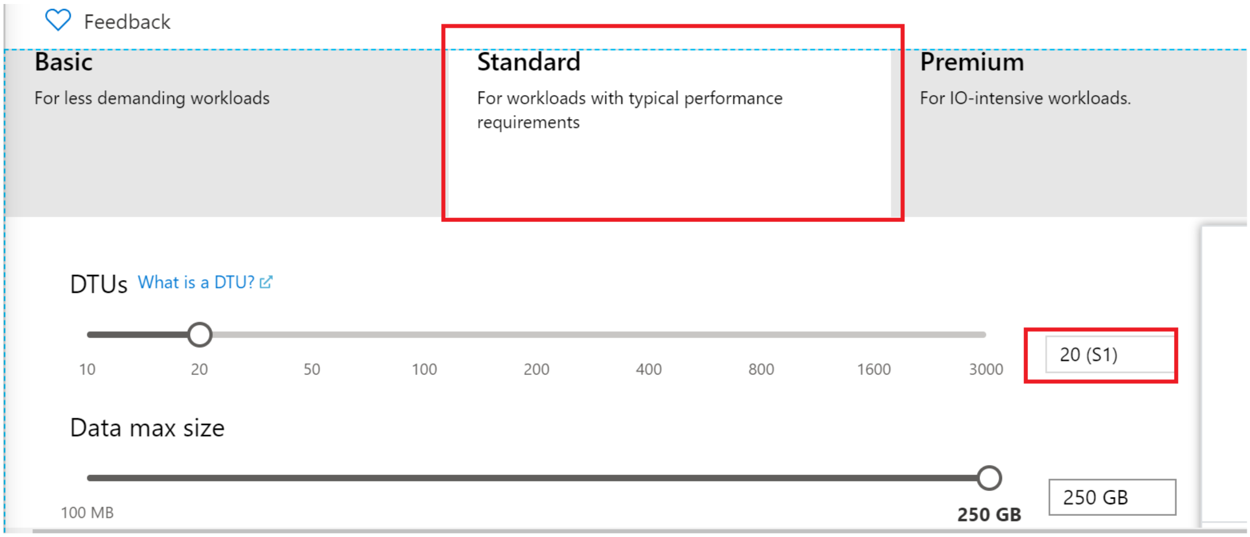
- **Job Agent:** Is Azure resource used for creating and managing the jobs and it is required an existing free SQL database Called **Job Database** and this Job Database used to store the jobs definition and the history of the jobs like our normal jobs in on-premises SQL Server the history and the configuration of the SQL jobs saved in system tables in MSDB database. Also, inside the Job database, we will have a default Stored procedure can be used for collecting information about the job definition and history, the recommended Service Tier for Azure Job Database is S1 or Higher.
- **Target Group**: and in this Target group we will define the target servers or the target databases that we need to execute these jobs on it and we can add in the target group (Azure Single database logical server or database, Azure SQL Database Elastic pool logical server or Elastic Pool Database, SQL Server database and Shardmap database)
- **Job Output**: and this is used to store the Outcome of the job execution and this information saved in the table on the existing database or a new database and this time this Database is called the **Output database.**

Azure SQL Job Agent Database and target Database Configuration

- First, we need to create a new empty Single database "**jobagentdatabase**" used as a Job database for the Job agent component on New SQL Azure Single Logical Server "**jobagentlogicalserver**" For more information about How to Provision Azure Single database check this **article** https://mostafaelmasry.com/2020/05/20/azure-single-database-fundamental/
- Second, we need to Create Target Database "**targetazuresqldb**" on New Azure Single Database Logical Server "**targetlogicalserver**" for our demo to execute our Jobs on it, If you already have a

database created you can use it as a target but as I am doing the demo I will create new target database.

- During the Configuration for Agent database and Target database we will select the services tier as Stander DTU S1 as we mentioned before it is recommended add in your note that you can configure this Agent database on a vCore-based Purchase model



Feedback

Basic	Standard	Premium
For less demanding workloads	For workloads with typical performance requirements	For IO-intensive workloads.

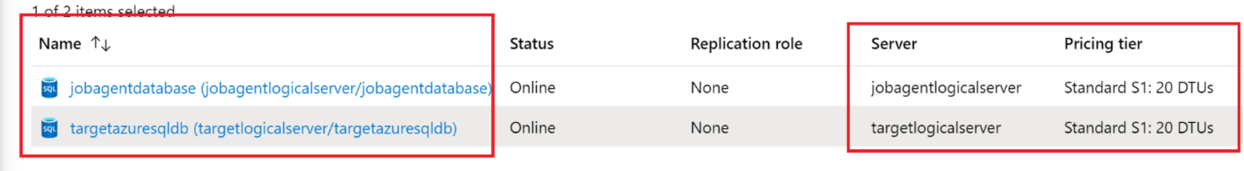
DTUs What is a DTU?

DTUs: 20 (S1)

Data max size

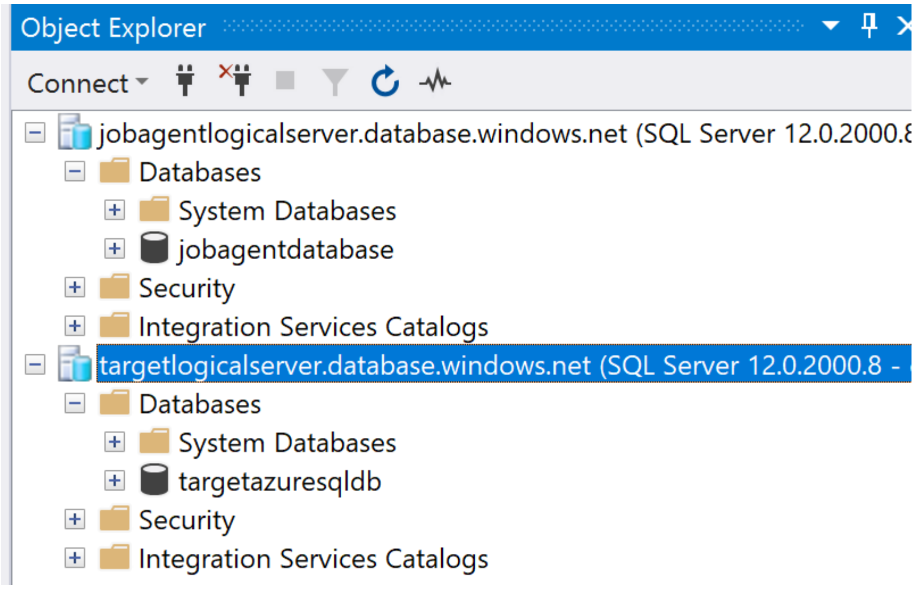
Data max size: 250 GB

- So now we have two Azure Single database one as an agent server and the second one as a target server each server contains a new empty database.
- Enable Server Firewall by adding your local IP address to be able to connect to the new servers through the SQL Server management studio SSMS on your local machine Check this **article** https://docs.microsoft.com/en-us/azure/sql-database/sql-database-server-level-firewall-rule#create-a-server-level-ip-firewall-rule for How to configure azure server firewall



1 of 2 items selected

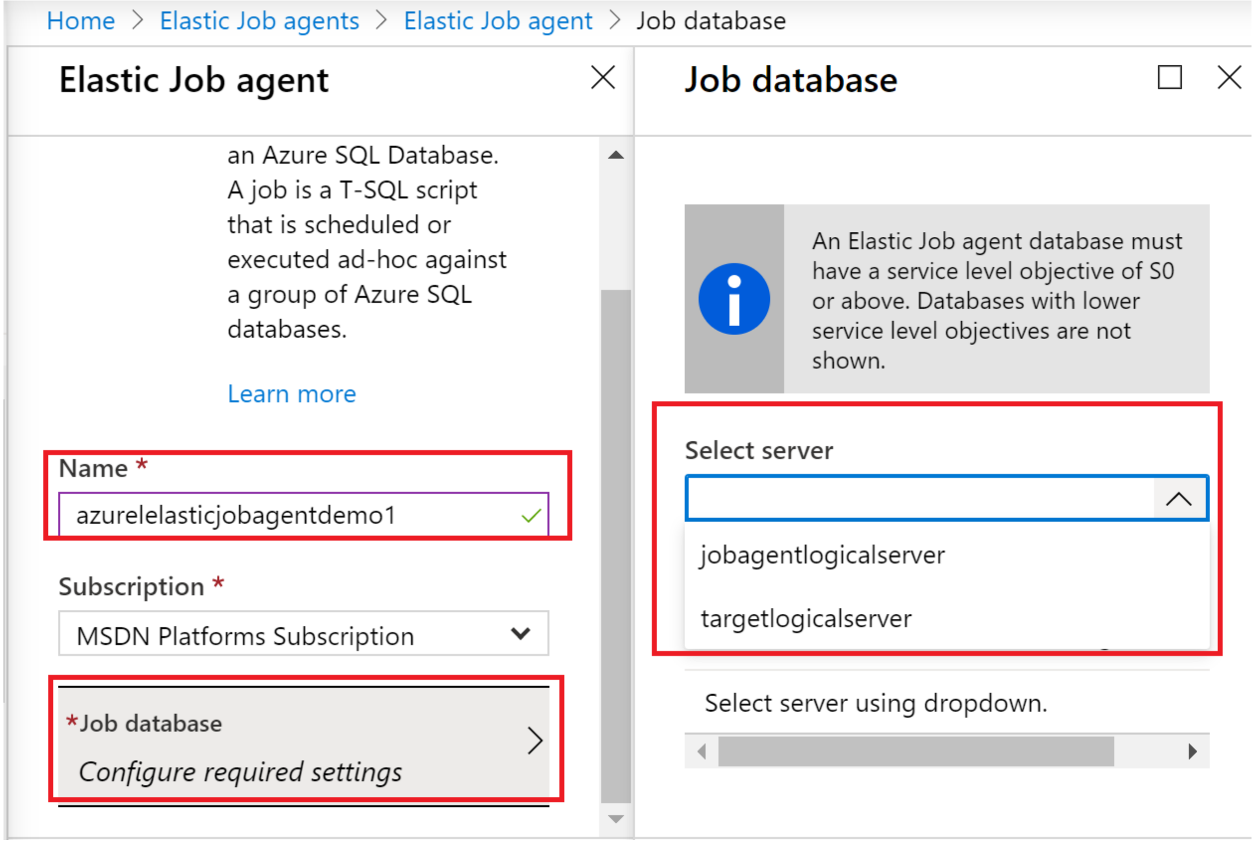
Name ↑↓	Status	Replication role	Server	Pricing tier
jobagentdatabase (jobagentlogicalserver/jobagentdatabase)	Online	None	jobagentlogicalserver	Standard S1: 20 DTUs
targetazuresqldb (targetlogicalserver/targetazuresqldb)	Online	None	targetlogicalserver	Standard S1: 20 DTUs

- 

jobagentlogicalserver.database.windows.net (SQL Server 12.0.2000.8
Databases
System Databases
jobagentdatabase
Security
Integration Services Catalogs
targetlogicalserver.database.windows.net (SQL Server 12.0.2000.8 -
Databases
System Databases
targetazuresqldb
Security
Integration Services Catalogs

Configure the Azure SQL Elastic job Agent

- Open Azure Portal and write on the search “ **Elastic Job agents**” or you can click on this **link** https://portal.azure.com/#blade/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Sql%2Fservers%2FjobAgents to target you to the **Elastic Job agents**
- **Click on Add to add new Elastic job Agent for example “azurelelasticjobagentdemo1”**
- **Name: You need to Provide the Name of the Elastic Job agents**
- **Job Database: in the Part, you will select the Job database logical server we created it “jobagentlogicalserver”**



Home > Elastic Job agents > Elastic Job agent > Job database

Elastic Job agent

an Azure SQL Database. A job is a T-SQL script that is scheduled or executed ad-hoc against a group of Azure SQL databases.

Learn more

Name *
azurelelasticjobagentdemo1 ✓

Subscription *
MSDN Platforms Subscription

*Job database
Configure required settings

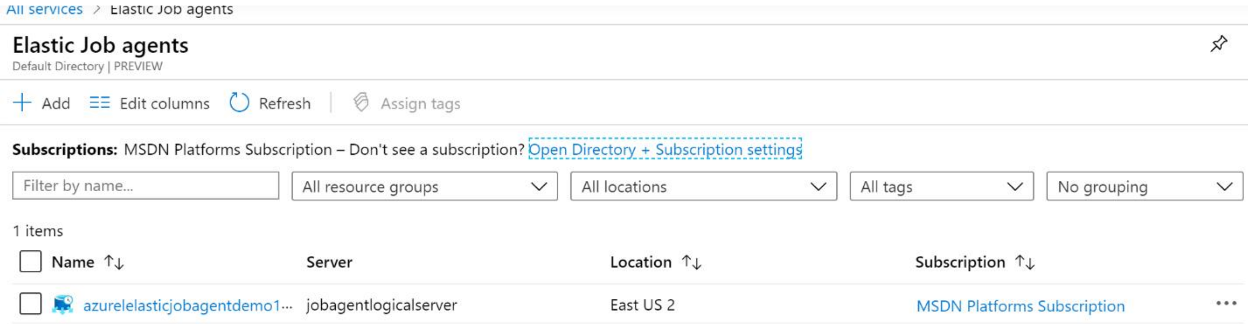
Job database

An Elastic Job agent database must have a service level objective of S0 or above. Databases with lower service level objectives are not shown.

Select server
- jobagentlogicalserver
- targetlogicalserver

Select server using dropdown.

- After this, the database Agent "**jobagentdatabase**" will appear to select it
- Now if you do refresh on **Elastic Job Agent** Services you will find the new Agent, we created it



All services > Elastic Job agents

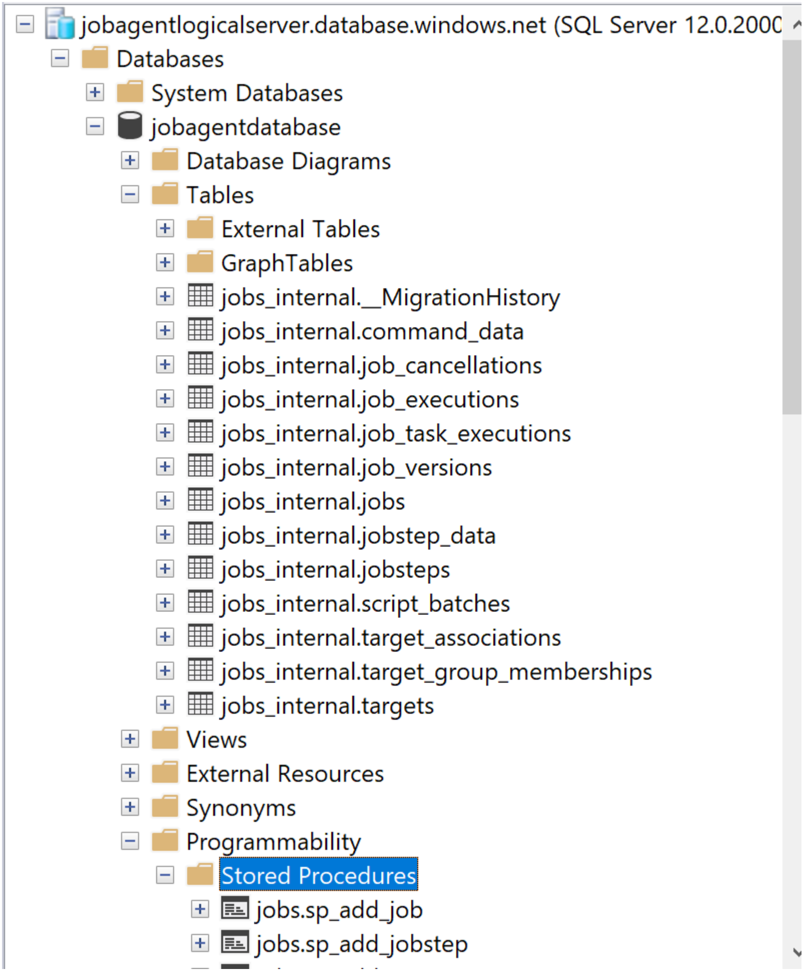
Elastic Job agents
Default Directory | PREVIEW

Subscriptions: MSDN Platforms Subscription – Don't see a subscription? Open Directory + Subscription settings

1 items

Name	Server	Location	Subscription
azurelelasticjobagentdemo1...	jobagentlogicalserver	East US 2	MSDN Platforms Subscription

- After the **Elastic Job agents If you do refresh now for the Agent database "jobagentdatabase" you will find some new tables and Stored Procedures created**

- 

  - jobagentlogicalserver.database.windows.net (SQL Server 12.0.2000
    - Databases
      - System Databases
      - jobagentdatabase
        - Database Diagrams
        - Tables
          - External Tables
          - GraphTables
          - jobs_internal.__MigrationHistory
          - jobs_internal.command_data
          - jobs_internal.job_cancellations
          - jobs_internal.job_executions
          - jobs_internal.job_task_executions
          - jobs_internal.job_versions
          - jobs_internal.jobs
          - jobs_internal.jobstep_data
          - jobs_internal.jobsteps
          - jobs_internal.script_batches
          - jobs_internal.target_associations
          - jobs_internal.target_group_memberships
          - jobs_internal.targets
        - Views
        - External Resources
        - Synonyms
        - Programmability
          - Stored Procedures
            - jobs.sp_add_job
            - jobs.sp_add_jobstep

Create and Configure Job

In the demo we will configure a new Job to insert some data in table “testdemo” we will create this table on target database “**targetazuresqldb**” and this Job Each execute will insert a new record on this table, this configuration we can do it using T-SQL or PowerShell in the references you will find articles for how to do this, and this demo I will use T-SQL, and to configure this we will need to do below steps

- Create credentials on Agent database
- Create target Group members
- Create a job for inserting record on the new table
- Execute the Job and test the results
- Create Logins on the target Database

Create credentials on Agent database

- Connect to Agent Server “**jobagentlogicalserver.database.windows.net**” Using SSMS and open new query on agent database “**jobagentdatabase**”
- Create master Key Encryption
- Crate CREDENTIAL on Agent database “**jobagentdatabase**” and this will be used to execute the job on the target database
- Create second CREDENTIAL Agent database “**jobagentdatabase**” and this will be used to connect to the Master database on target Server “**targetlogicalserver**”

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD='Elmasrydemo!@#';
GO
CREATE DATABASE SCOPED CREDENTIAL ElmasryjobScopcred
WITH IDENTITY = 'Elmasryjobcred',
SECRET = '<Elmasrydemo!@#';
GO
CREATE DATABASE SCOPED CREDENTIAL ElmasrymasterScopcred
WITH IDENTITY = 'Elmasrymastercred',
SECRET = 'Elmasrydemo!@#';
GO
```

Create target Group members

In this part we will create a target group that will contain all of the jobs will be executed on the target database and this target group will be created on the Agent database “**jobagentdatabase**”

```
EXEC jobs.sp_add_target_group 'ElmasryServerTargetGroup1'
```

Now we need to add the target Server “**targetlogicalserver.database.windows.net**” to the target group “'**ElmasryServerTargetGroup1**'” this target server is the server that we need to execute the job on it, this configuration will be executed on job Agent database “**jobagentdatabase**”

In this script, we will need to Provide the Target group Name “'**ElmasryServerTargetGroup1**'” the Second Credential we created it to be used to connect to the Master database on the Target server “**ElmasrymasterScopcred**” and finally we need to provide the Target Server name “**targetlogicalserver.database.windows.net**”

Add in your note if the target server contains multiple databases all of this database are stored in the Target group and you can run your job on all of them if you need, if you have multiple target servers you should do these steps on the agent database by the information of each target server so this mean the job agent database will be used as a repository for all of the jobs information and hasty the executed on one target server or multiples target servers

```
EXEC jobs.sp_add_target_group_member
'ElmasryServerTargetGroup1',
@target_type =  N'SqlDatabase',
--@refresh_credential_name='ElmasrymasterScopcred',
@server_name='targetlogicalserver.database.windows.net',
@database_name =N'targetazuresqldb'
GO
```

Create a job for inserting record on the new table

In this part we will configure and create our job on job agent database "**jobagentdatabase**" that will be used to insert new record on our target database before doing this step I will connect to the target database "**targetazuresqldb**" that exists on the target server "**targetlogicalserver.database.windows.net**" and I will create new Empty table "**TargetDemotable**"

```
Create table TargetDemotable
(
ID int Primary key identity(1,1) Not NULL,
Empname Nvarchar(100) NOT NULL
)
```

Now we will return back to agent job database "**jobagentdatabase**" to create the Job on it

```
EXEC jobs.sp_add_job @job_name='insert_data_into_TargetDemotable',
@description='
insert test data in table TargetDemotable
inside target database targetazuresqldb
Using Azure SQL Elastic Agent Job'
```

Now we will add step to the job '**insert_data_into_TargetDemotable**' and this T-SQL required the job name, Command you need to Execute it though this step, Target Group Name and the First CREDENTIAL **ElmasryjobScopcred** we created it to be used to execute the job on Target database

```
EXEC jobs.sp_add_jobstep @job_name='insert_data_into_TargetDemotable',
@command=N'insert into TargetDemotable
values
(''EMP_Elmasry'');',
@credential_name='ElmasryjobScopcred',
@target_group_name='ElmasryServerTargetGroup1'
```

Execute the Job and test the results

Now after creating the job on the Job agent database we need to execute the job to test the results of it

- Execute the Job command

```
EXEC jobs.sp_start_job 'insert_data_into_TargetDemotable'
```

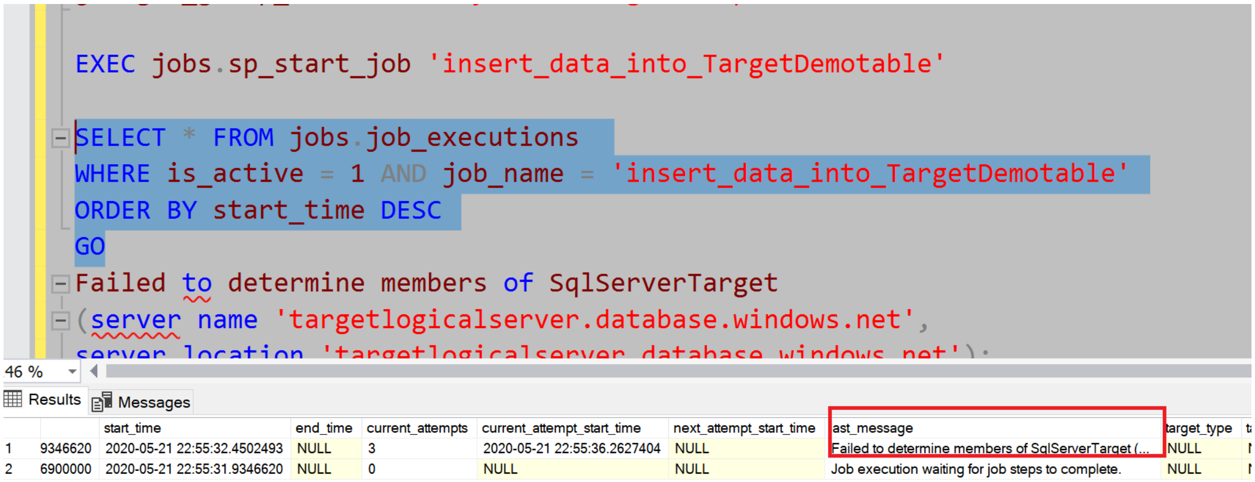
- Check the job history is it a success for fail

```
SELECT * FROM jobs.job_executions
WHERE is_active = 1 AND job_name = 'insert_data_into_TargetDemotable'
ORDER BY start_time DESC
GO
```

The Job will be failed because we are not allowed the Agent database server to connect to the target database server we should add the agent database server IP on the Firewall configuration of the target Server

Error Description

```
Failed to determine members of SqlServerTarget
(server name 'targetlogicalserver.database.windows.net',
server location 'targetlogicalserver.database.windows.net'):
Cannot open server 'targetlogicalserver' requested by the login.
The client with IP address '40.70.145.9' is not allowed to access the server.
To enable access, use the Windows Azure Management Portal or run sp_set_firewall_rule
on the master, database to create a firewall rule for this IP address or address range.
It may take up to five minutes for this change to take effect.
(Msg 40615, Level 14, State 1, Line 65536)
```

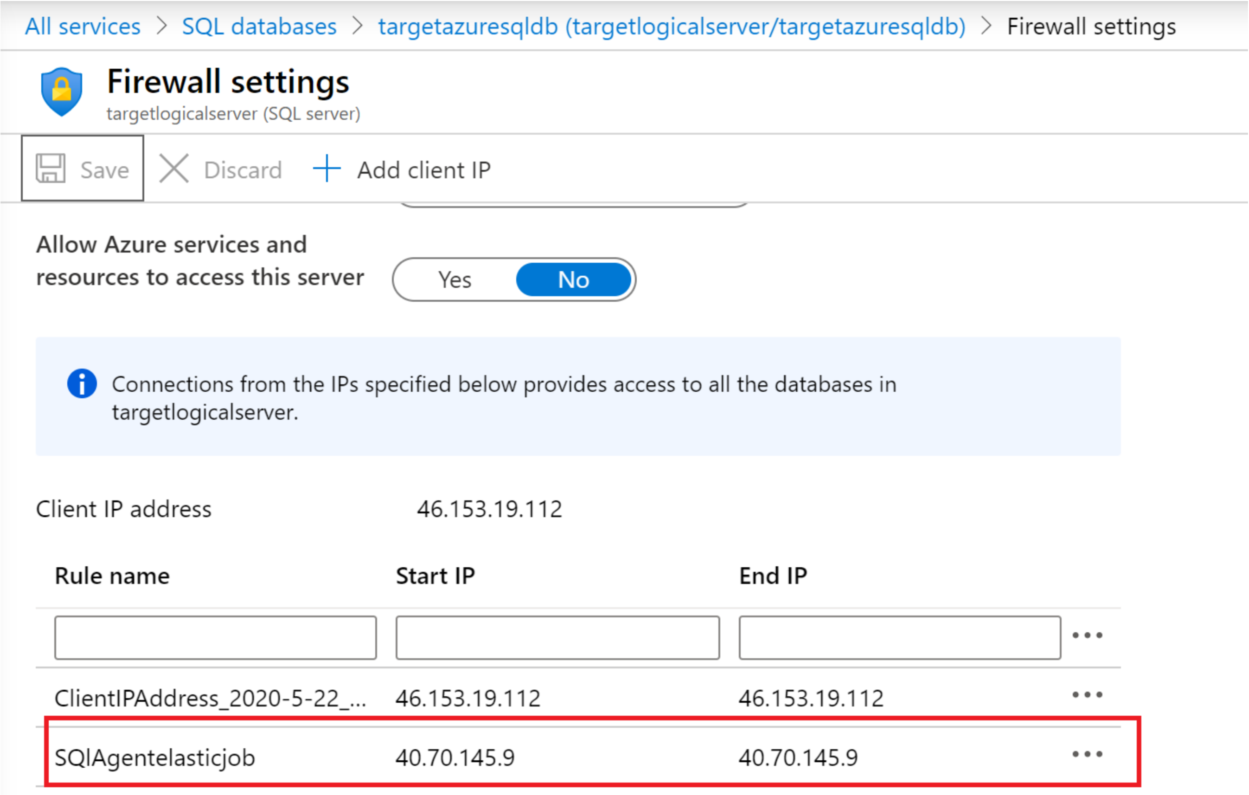


```
EXEC jobs.sp_start_job 'insert_data_into_TargetDemotable'

SELECT * FROM jobs.job_executions
WHERE is_active = 1 AND job_name = 'insert_data_into_TargetDemotable'
ORDER BY start_time DESC
GO
Failed to determine members of SqlServerTarget
(server name 'targetlogicalserver.database.windows.net',
server location 'targetlogicalserver.database.windows.net');
```

		start_time	end_time	current_attempts	current_attempt_start_time	next_attempt_start_time	ast_message	target_type	ta
1	9346620	2020-05-21 22:55:32.4502493	NULL	3	2020-05-21 22:55:36.2627404	NULL	Failed to determine members of SqlServerTarget (...	NULL	N
2	6900000	2020-05-21 22:55:31.9346620	NULL	0	NULL	NULL	Job execution waiting for job steps to complete.	NULL	N

To add this IP in the target Server Firewall open Azure Portal → Open the target database → in the top Select "**Set Server Firewall**" then add the Job Agent Server IP appeared in the Error massage



All services > SQL databases > targetazuresqldb (targetlogicalserver/targetazuresqldb) > Firewall settings
Firewall settings
targetlogicalserver (SQL server)
Save
Discard
Add client IP

Allow Azure services and resources to access this server: Yes [✓ No]

Connections from the IPs specified below provides access to all the databases in targetlogicalserver.

Client IP address 46.153.19.112

Rule name	Start IP	End IP
ClientIPAddress_2020-5-22_...	46.153.19.112	46.153.19.112
SQlAgentelasticjob	40.70.145.9	40.70.145.9

Now let us Execute again the command of the Execute Job and Select the results again to check is it a success or failed

```
EXEC jobs.sp_start_job 'insert_data_into_TargetDemotable'
```

```
SELECT * FROM jobs.job_executions
WHERE is_active = 1 AND job_name = 'insert_data_into_TargetDemotable'
ORDER BY start_time DESC
GO
```

You will find the job failed again because we are not Created the two logins (**Elmasrymastercred**, **Elmasryjobcred**) we created it on the Agent database

```
Failed to determine members of SqlServerTarget
(server name 'targetlogicalserver.database.windows.net',
server location 'targetlogicalserver.database.windows.net'):
Login failed for user 'Elmasrymastercred'.
(Msg 18456, Level 14, State 1, Line 65536)
```

Create Login on the target server

To fix the last issue of “Failed to determine members of SqlServerTarget” we need to do the below steps

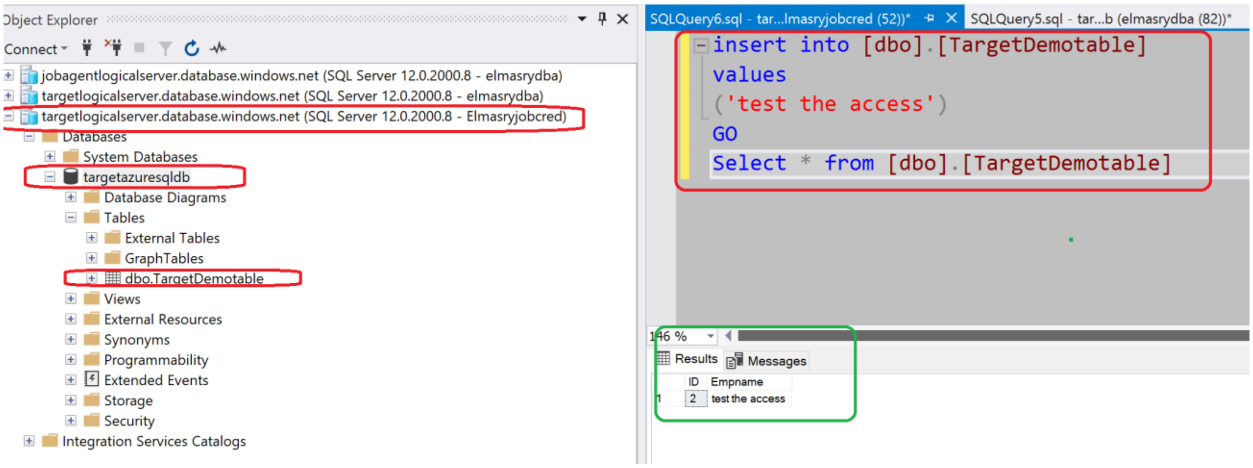
- Connect to the target server “**targetlogicalserver.database.windows.net**” using the SSMS
- Open New Query on Master database to create login and user for “**Elmasrymastercred**” that we created it to be used to connect to the master database and we will create login only for the login “**Elmasryjobcred**” that we created it to be used to Executing the job on all of the Target databases, so this means the User for this login “**Elmasryjobcred**” will be created on all of the target databases

```
CREATE LOGIN Elmasrymastercred WITH PASSWORD='Elmasrydemo!@#'
CREATE USER Elmasrymastercred FROM LOGIN Elmasrymastercred
CREATE LOGIN Elmasryjobcred WITH PASSWORD='Elmasrydemo!@#'
```

- Open new Query on the target database “**targetazuresqldb**”
- Execute the below query to create a user for login “**Elmasryjobcred**” and grant access to this user on the database to be able to insert the new record

CREATE USER Elmasryjobcred FROM LOGIN Elmasryjobcred
GRANT ALTER ON SCHEMA::dbo TO Elmasryjobcred
EXEC sp_addrolemember 'db_datareader', 'Elmasryjobcred';
EXEC sp_addrolemember 'db_datawriter', 'Elmasryjobcred';

- Now if you need to test the new login "**Elmasryjobcred**" you can connect to it using SSMS to the target server


Object Explorer
Connect
jobagentlogicalserver.database.windows.net (SQL Server 12.0.2000.8 - elmasrydba)
targetlogicalserver.database.windows.net (SQL Server 12.0.2000.8 - elmasrydba)
targetlogicalserver.database.windows.net (SQL Server 12.0.2000.8 - Elmasryjobcred)
Databases
System Databases
targetazuresqldb
Database Diagrams
Tables
External Tables
GraphTables
dbo.TargetDemotable
Views
External Resources
Synonyms
Programmability
Extended Events
Storage
Security
Integration Services Catalogs
SQLQuery6.sql - tar...lmasryjobcred (52))*
SQLQuery5.sql - tar...b (elmasrydba (82))*
146 %
Results
Messages

```
insert into [dbo].[TargetDemotable]
values
('test the access')
GO
Select * from [dbo].[TargetDemotable]
```

```
   ID  Empname
1  2   test the access
```

It is time now to Execute the Job again after we fixed two logins issue I insist to write this issue to be clear for all of us in the real scenario after creating the job and the step go direct to create the login on the Master target server and target database and add the Agent server IP on the target Server Firewall to avoid these two issues

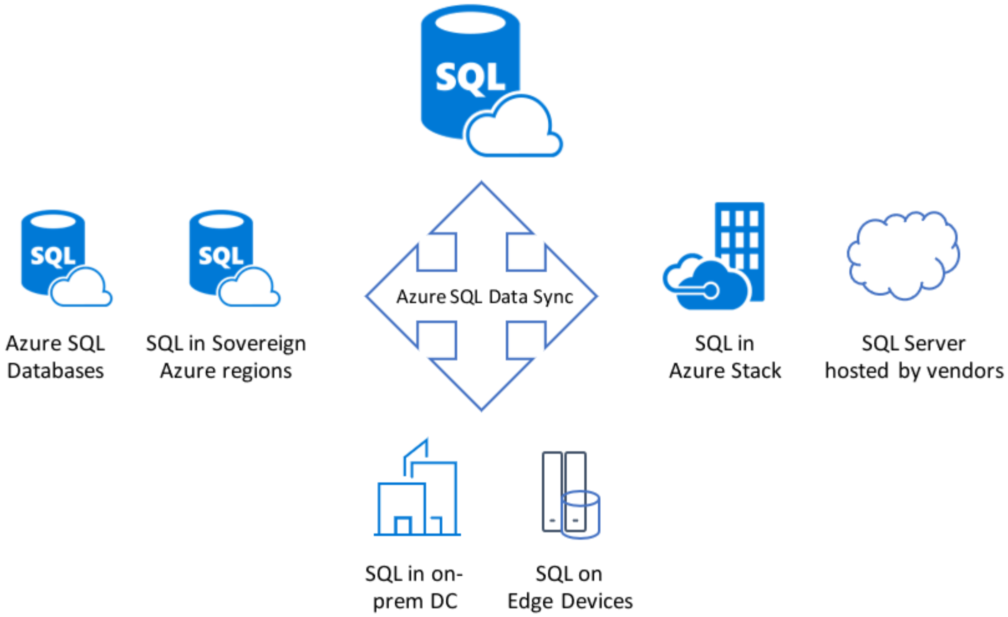
References

- Elastic pools help you manage and scale multiple Azure SQL databases https://docs.microsoft.com/en-us/azure/sql-database/sql-database-elastic-pool
- Create, configure, and manage elastic jobs https://docs.microsoft.com/en-us/azure/sql-database/elastic-jobs-overview
- Create an Elastic Job agent using PowerShell https://docs.microsoft.com/en-us/azure/sql-database/elastic-jobs-powershell
- Use Transact-SQL (T-SQL) to create and manage Elastic Database Jobs https://docs.microsoft.com/en-us/azure/sql-database/elastic-jobs-tsql

Azure SQL Database Data Sync

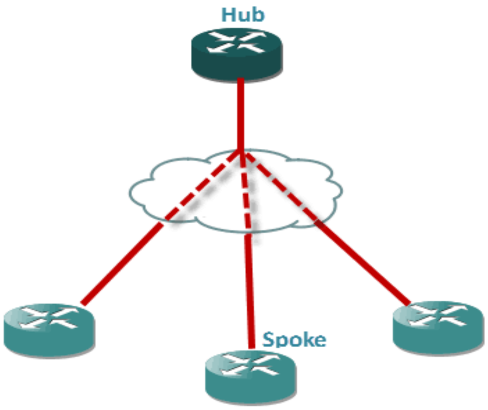
Introduction

Azure Data Sync is a Microsoft service **announced** on June 18, 2018, and it is used to sync the data from Azure SQL to another Azure SQL Database or from SQL Server on-premises to Azure SQL Bi-Directional these services used for data sync only and you should not use it as disaster recovery or as migration tool or as read the only replica IF you need to know when you should use azure data sync check this **Microsoft article**, and if you need to know What is the best option for Azure SQL Database Migration, replication and read-only check this link **https://lnkd.in/edn6nyY/#AzureSQL** you will find very useful articles



Azure Data Sync Concept

- **Sync Group:** is one of the databases that you want to synchronize
- **Hub Database**: is one of the synchronize Databases
- **Member database:** another database in the data sync synchronization is member databases
- **Synchronization**: Happened between Hub Database and member databases this means when you do change on one of the member databases the changes reflected on the other member databases through the Hub database



Sync Group Properties

- **Sync Schema:** Contain the description of the data you need to sync it and you can select multiple tables or you can specify some columns in certain tables
- **Sync Direction:** we have two option Bi-Directional or can be only one direction IF we select the Bi-Directional this means when changes happened on the Hub database it will be reflected on all member database and the reveres is correct
- **Sync interval:** With Sync interval you can decide How and when synchronization occurs.
- **Conflict Resolution Policy:** it is a group policy used to fix the issues can be happened in case a conflict happened between the Hub database and Member database and we have two options here Hub Wins or Member Wins
- **Hub Wins:** IF you selected this option as Conflict resolution this means the change from the Hub database is persisted
- **Member Wins:** IF you selected this option as Conflict resolution this means he changes from the member database is persisted when a conflict arises.

Azure Data Sync Tips and Considerations

- Hub Database and Sync Database must be an Azure SQL Database
- Hub Database and Sync Database must be hosted on the same Azure region
- Sync Database created Automatic when you are configuring Azure data sync
- Member database can be SQL Server on-premises, Azure SQL Database, or SQL Server database on Azure VM
- IF you will use SQL Server on-premises as a Member database in Azure data sync you must install Azure Data Sync agent on this on-premises server. Download it from **here**
- Azure data sync not supported till this moment Azure SQL Managed instance IF you need to know more about Managed instance check this **article**

- Azure Data Sync used SQL triggers (insert, Update, Delete) to track the changes and to move it to the member databases and it is created side tables in member databases for change tracking
- Any table you need to add it in Azure data sync must have a primary key
- Snapshots isolation must be enabled
- The table you will add it in Data Sync if it is containing identity column should be primary key but identity column and non-primary key this column cannot be included in Azure data Sync
- Columns with user-defined data types are not supported
- Automatic Sync can be configured between 5 minutes till 30 days

For more information about Azure Data Sync limitation that you should consider it before implementation Check this **Microsoft documentation**

How Azure Data Sync Work

As I mentioned in Azure tips that Azure data Sync depend on Triggers to tracking the data and the Hub will sync the data to the members For more information check this **Link**

Demo introduction

Let us assume that we have Azure SQL Single Database named “**Member01**” and this DB hosted on logical Server “**memberlogicaldbserver.database.windows.net**” and inside this DB You have some tables and you need to build Sync between this DB and another Azure Single DB named “Member02” hosted on logical Server “**member02logicaldbserver.database.windows.net**”, So whatever any update, insert, delete happened on certain tables or certain columns in **Member01** should be reflected on **Member02** database and vice versa

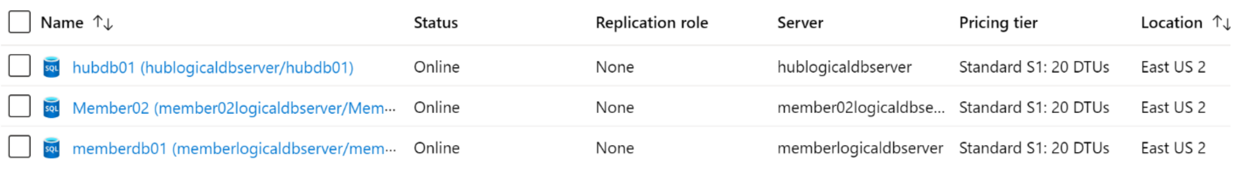
What we Should do achieve these requirements

- IF you are in the demo we should build two SQL Server Azure Single Databases Member01 and Member02 and the schema of the table should be created on both databases
- We need to create a Hub Database that will be used to move the changes between the two members databases
- We need to create Sync Database it will contain the Sync Configuration, history, actions this database created automatically when we are creating the Database Sync Group Services, add in your note sync database should be hosted in the same region of the Hub database
- Based on this Configuration we will have 3 Azure Single logical Server and 4 Databases

Server Name	Database name

memberlogicaldbserver.database.windows.net	Memberdb01
member02logicaldbserver.database.windows.net	Member02
hublogicaldbserver.database.windows.net	Hubdb01
	Syncdb01

- Memberdb01, Member02, and Hubdb01 all of them are normal Azure Single Database deployment for more information about how to provision Azure Single Database check this article
- Syncdb01 will be created through the upcoming configuration we will do it to create Database Sync Group
- Memberdb01 and member02 databases you can host them in one logical server no problem for example “memberlogicaldbserver.database.windows.net”
- Hubdb01 and Syncdb01 can be hosted on separate Azure Logical Server only the requirements are both databases should be hosted in the same region.
- You Should Enable the Server Firewall by adding local machine IP on all Azure SQL Database to be able to connect to them through the SQL Server Management Studio
- Allow Azure services and resources to access this server from the Server Firewall Configuration to allow the Hub and member databases to communicate to sync the data.



Name ↑↓	Status	Replication role	Server	Pricing tier	Location ↑↓
hubdb01 (hublogicaldbserver/hubdb01)	Online	None	hublogicaldbserver	Standard S1: 20 DTUs	East US 2
Member02 (member02logicaldbserver/Mem...	Online	None	member02logicaldbse...	Standard S1: 20 DTUs	East US 2
memberdb01 (memberlogicaldbserver/mem...	Online	None	memberlogicaldbserver	Standard S1: 20 DTUs	East US 2

As we can see we have 3 Logical Server each server contains Azure Single Database. On the next step, we will create Database Sync Group and during the configuration process we will create the Sync database on HublogicalServer

Create Database Sync Group between two Azure SQL Single Database.

Check the below GIF File to know when we can create Database Sync Group and add the member databases on the syn group then sync the changes between the members and Hub.

Create Database Sync Group between Azure SQL Database and On-premises SQL Server

Azure Data Sync References

- https://www.sqlshack.com/how-to-sync-azure-sql-databases-and-on-premises-databases-with-sql-data-sync/

- https://www.sqlshack.com/azure-sql-data-sync-replicate-data-and-schema-changes-between-azure-sql-databases/
- https://docs.microsoft.com/en-us/azure/sql-database/sql-database-get-started-sql-data-sync#step-1---create-sync-group
- https://docs.microsoft.com/en-us/azure/sql-database/sql-database-sync-data
- https://azure.microsoft.com/en-us/blog/announcing-the-general-availability-of-azure-sql-data-sync/

Azure DB-300 Exam QA

- In case if you have 20 Azure SQL DB and you need to move them to Azure SQL Elastic pool in this case you need to consider 3 things (total size of all the databases, number of concurrently peaking databases * peak CPU utilization per database, the total number of databases * average CPU utilization per database) Explanation:
  - Estimate the vCores needed for the pool (For vCore-based purchasing model: MAX (<Total number of DBs X average vCore utilization per DB>, <Number of concurrently peaking DBs X Peak vCore utilization per DB))
  - Estimate the storage space needed for the pool by adding the number of bytes needed for all the databases in the pool
- IF You need to apply 20 built-in Azure Policy definitions to all new and existing Azure SQL Database deployments in an Azure subscription to minimize administrative effort
  - **Create an Azure Policy Initiative**: The first step in enforcing compliance with Azure Policy is to assign a policy definition. A policy definition defines under what condition a policy is enforced and what effect to take, with an initiative definition, you can group several policy definitions to achieve one overarching goal. An initiative evaluates resources within the scope of the assignment for compliance with the included policies
  - **Create an Azure Policy Initiative assignment**: Assign the initiative definition you created in the previous step.
  - **Run Azure Policy remediation tasks**: To apply the Policy Initiative to the existing SQL databases.

  - https://docs.microsoft.com/en-us/azure/governance/policy/tutorials/create-and-manage
- To ensure that an automatic email notification is sent once the job completes:
  - In Object Explorer, expand a SQL Server instance.
  - Right-click SQL Server Agent, and then click Properties.
  - Click Alert System.
  - Select Enable Mail Profile.
  - In the Mail system list, select Database Mail.
  - In the Mail profile list, select a mail profile for Database Mail.
  - Restart SQL Server Agent.
  - Prerequisites include:
    - ✓ Enable Database Mail.
    - ✓ Create a Database Mail account for the SQL Server Agent service account to use.
    - ✓ Create a Database Mail profile for the SQL Server Agent service account to use and add the user to the DatabaseMailUserRole in the msdb database.
    - ✓ Set the profile as the default profile for the msdb database.

- https://docs.microsoft.com/en-us/sql/relational-databases/database-mail/configure-sql-server-agent-mail-to-use-database-mail

1. Which authentication method would you recommend for the application on an Azure VM?
   - Azure AD authentication: managed identity

2- Which authentication method would you recommend for the application on a non-Azure machine that is domain-joined?
   - Azure AD authentication: integrated authentication

3- Which authentication method would you recommend for SQL admin tools (SSMS, PowerShell) on a non-Azure machine that is not domain-joined?
   - Azure AD authentication: interactive with multi-factor authentication (MFA)

4- Which authentication method would you recommend for the application on the legacy application where you can't change the driver/connection string on a non-Azure machine?
   - SQL authentication: username and password

5- What is the best option for them to enable geo-redundancy and maintain high availability?
   - Auto-failover groups

6- How can you ensure that DBAs cannot see sensitive data stored in specific columns?
   - Always Encrypted with role separation

7- How can you track access to tables containing sensitive data?
   - SQL Audit and Data Classification

Copy protected with Online-PDF-No-Copy.com