Microsoft

Plan and Implement Data Platform Resources

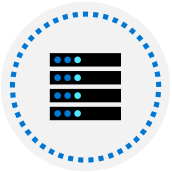
Introduction to deploying Azure resources, choosing an appropriate database offering, configuring Azure resources, and migration strategies for Azure

Objectives

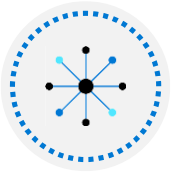
- Deploy resources using manual methods
- Recommend an appropriate database offering based on requirements
- Configure Azure SQL resources
- Evaluate and implement a strategy for moving a database to Azure SQL

Deploy IaaS Solutions with Azure SQL

Objectives



Explore the basics of SQL Server in an Infrastructure as a Service (IaaS) Offering



Learn the available options for provisioning and deployment

Explore performance and security options available

Understand the high availability and disaster recovery options

Azure SQL platform offerings

SQL Server on Azure Virtual Machines

Best for lift and shift and/or workloads requiring OS-level access

Azure SQL Managed Instance

Best for modernizing existing apps

Azure SQL Database

Best for supporting modern cloud apps

Best for extending apps to IoT edge

Infrastructure-as-a-Service | **Platform-as-a-Service** | **Edge Computing**

Azure SQL enabled by Azure Arc

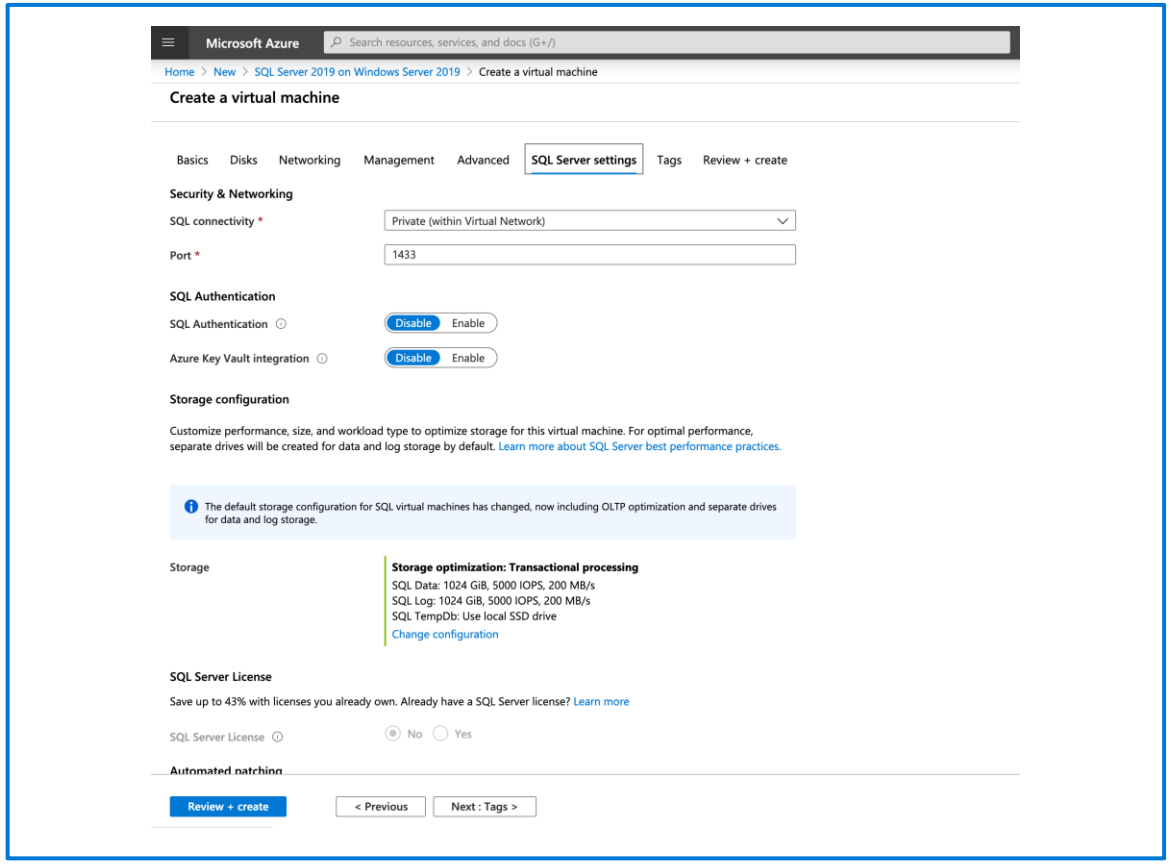
Run Azure SQL on premises and in multicloud environments

Elimination of Hardware and Software	Operational Efficiency & Savings	Improved Business Agility

Azure deployment options

You can deploy Azure resources using the following methods:

- Using the Azure Portal
- Running PowerShell or Azure CLI scripts
- Deploying Azure Resource Manager templates



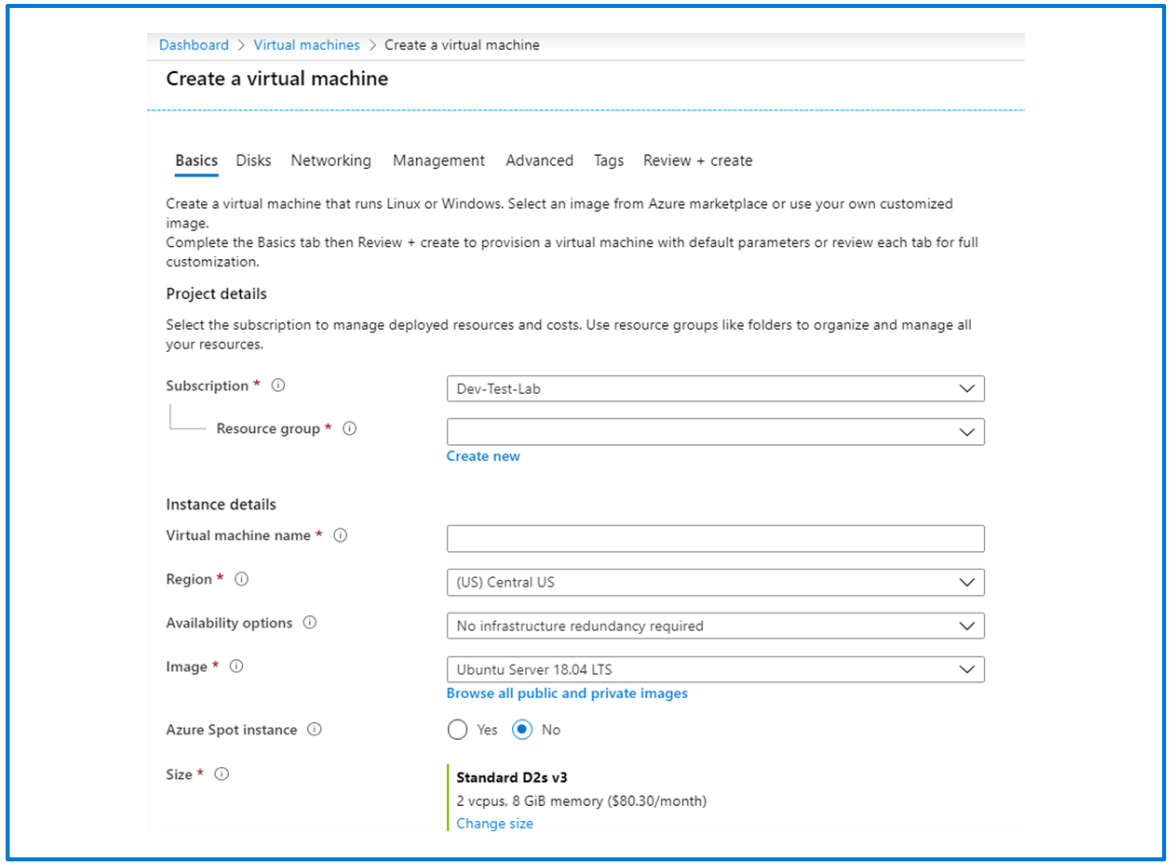
Deploying from the Azure portal

This method is useful for single VM deployments

The portal provides step by step instructions for deployment

This is a manual process

Not easily repeatable in a consistent matter

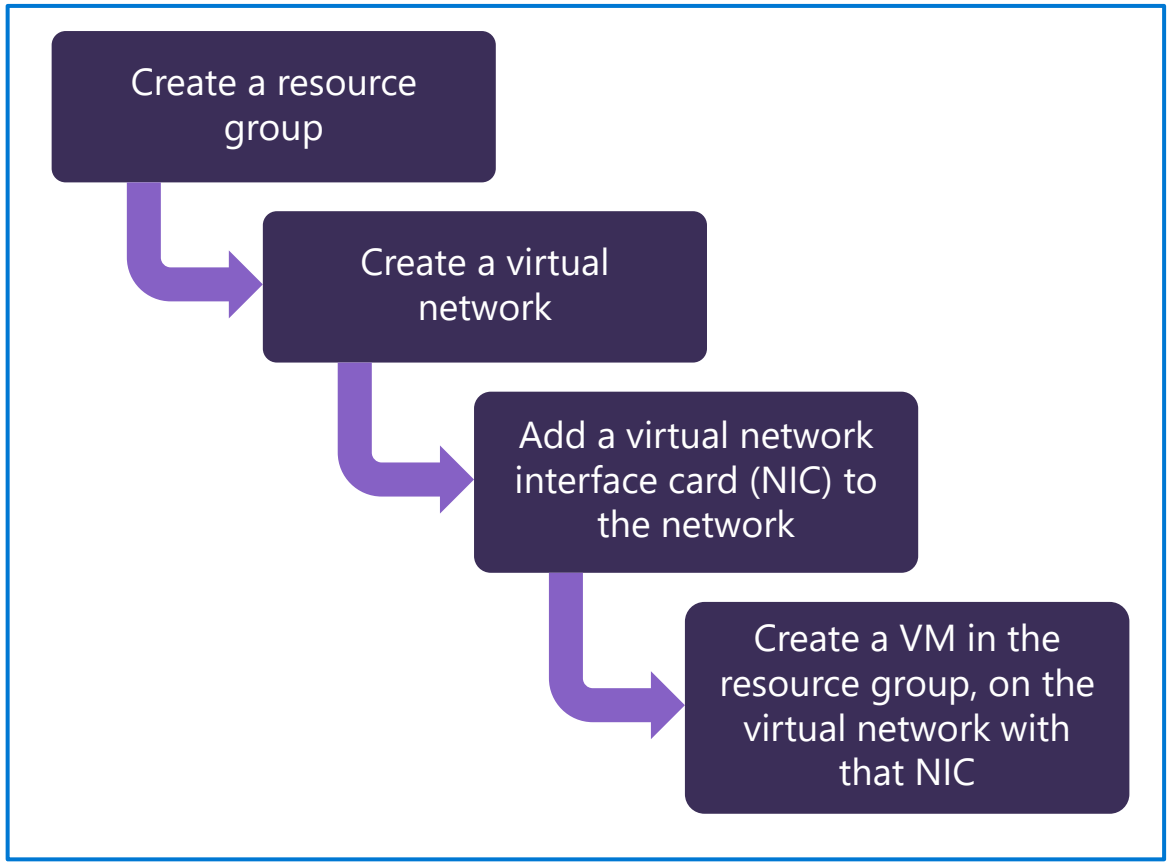


Deployment using PowerShell/Azure CLI

While not as flexible as using ARM templates, PowerShell and the Azure CLI provide a repeatable methodology for deploying resources

Using PowerShell or the Azure CLI are an imperative framework, which means they indicate a specific order of operations to execute

This approach can offer flexibility and ease of use for some situations compared to the relative complexity of ARM templates



Azure Resource Manager templates

ARM Templates provide a repeatable, declarative process for deploying Azure resources at scale

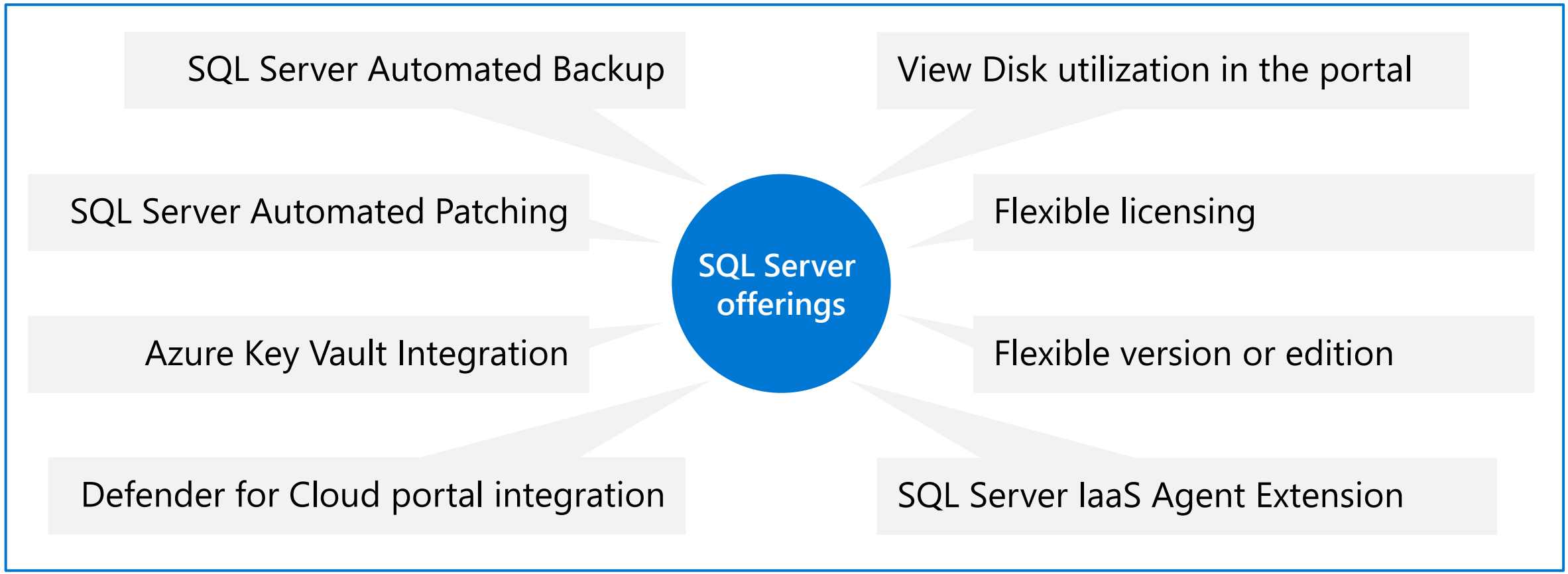
Allow for integration with popular continuous integration and deployment CI/CD tools

Allow you to create nearly any Azure resource or option

JSON

```
"resources": [
  {
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2016-01-01",
    "name": "mystorageaccount",
    "location": "westus",
    "sku": {
      "name": "Standard_LRS"
    },
    "kind": "Storage",
    "properties": {}
  }
]
```

SQL Server IaaS Agent Extension



SQL Server licensing models in Azure

Without Software Assurance

Use a SQL Server image from the marketplace and pay-per-minute. This is known as Pay as You Go licensing

With Azure Hybrid Licensing Benefit your options are

- Bring Your Own License (BYOL) – this is check box in the Azure Portal
- Manually installing SQL Server on a Windows or Linux VM
- Uploading a SQL Server VM image (VHD) to Azure and deploying that image as an Azure VM

Azure Virtual Machine families

- **General purpose:**
  Balanced CPU to memory ratio
- **Memory optimized:**
  High memory to CPU ratio
- **Storage optimized:**
  High disk throughput provided by local storage
- **Compute optimized:**
  High CPU to memory ratio
- **High performance compute:**
  Most powerful CPU

Type	Sizes
General purpose	B, Dsv3, Dv3 Dasv4, Dav4, DSv2, Dv2, Av2, DC, DCv2
Compute optimized	Fsv2
Memory optimized	Esv3, Ev3, Easv4, Eav4, Mv2, M, DSv2, Dv2
Storage optimized	Lsv2
GPU	NC, NCv2, NCv3, ND, NDv2(Preview), NV, NVv3, Nvv4
High performance compute	HB, HBv2, HC, H

Azure Virtual Machine sizing

A Series – Entry-level for dev/test

Bs Series – Economical bursting

D Series – General purpose compute

Dc Series – Protect data in use

E Series – In-memory hyper-threaded applications optimized

F Series – Compute optimized

G Series – Memory and storage optimized

H Series – High performance computing

Ls Series – Storage optimized

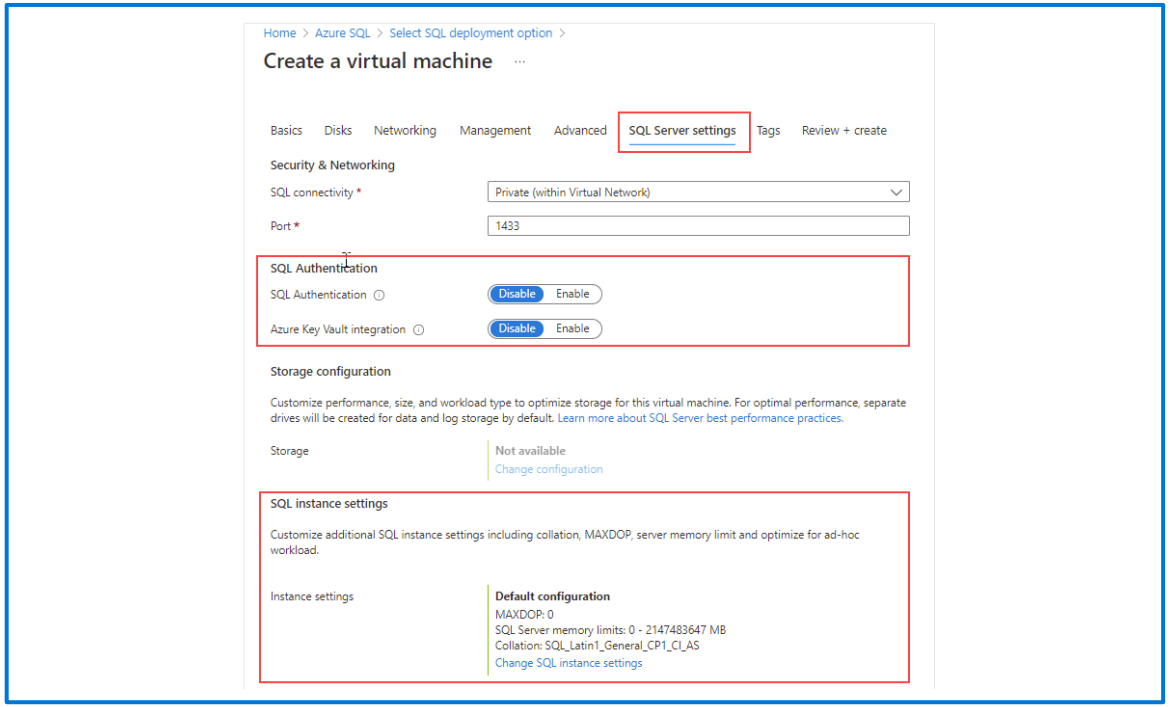
M Series – Memory optimized

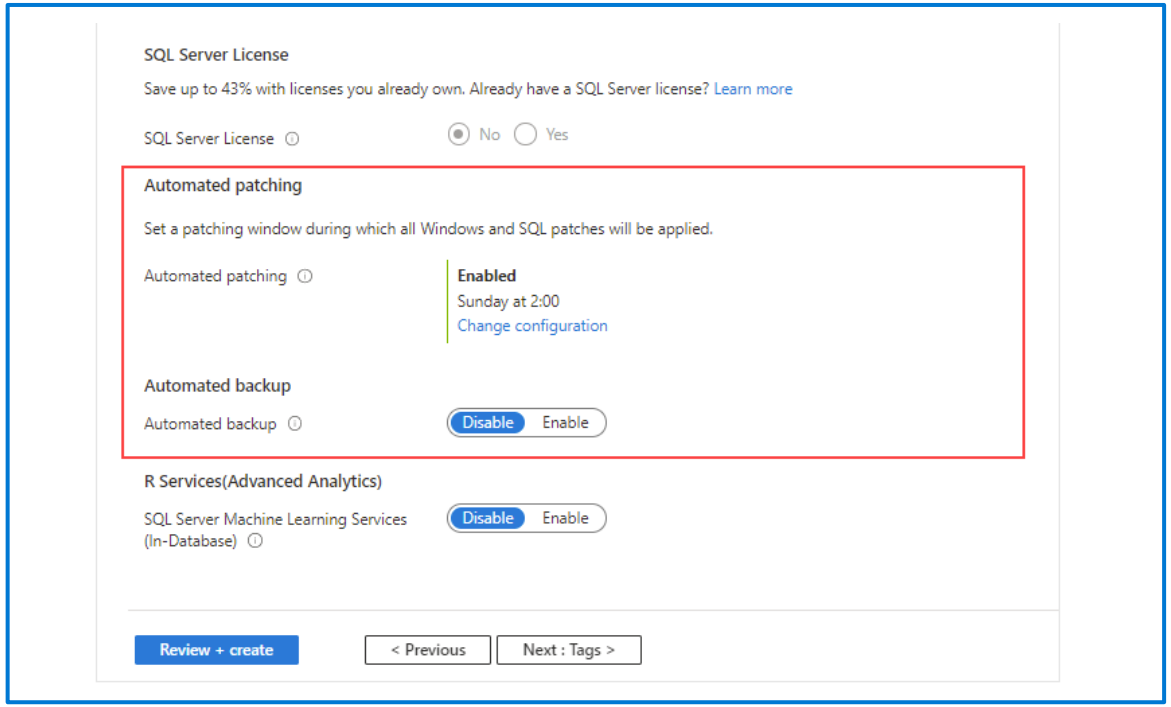
Mv2 Series – Largest memory optimized

N Series – GPU enabled

SQL Server configuration

Configure specific SQL Server settings like Security and Networking, SQL Authentication preferences, SQL instance settings, and other options.



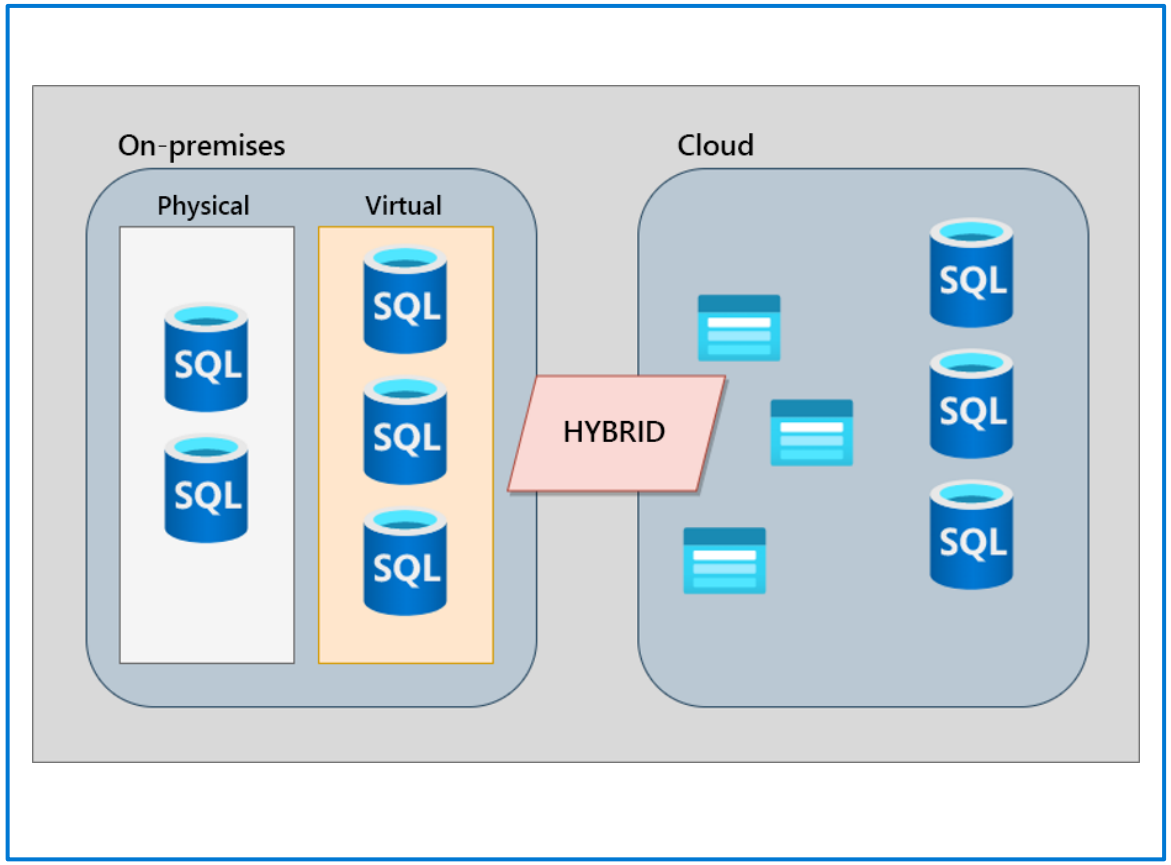


Understand hybrid scenarios

Organizations commonly have a mixture of physical and virtualized deployments of SQL Server

Offers the benefits of both on-premises and cloud services – extends on-prem solutions

Cloud component is usually used for storage or SQL Server VMs

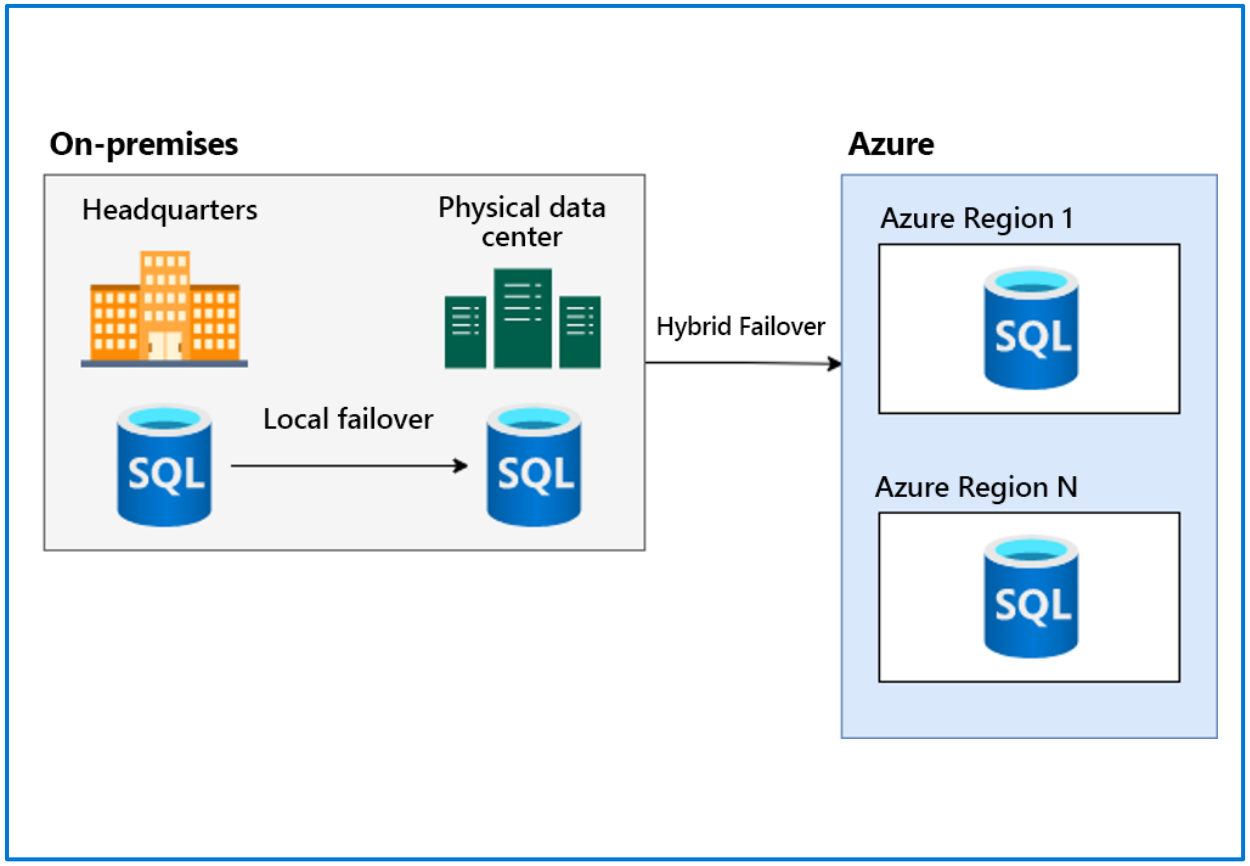


Hybrid scenarios for SQL Server – Disaster Recovery

Most common scenario for a hybrid deployment of SQL Server.

Organizations ensure business continuity during catastrophic events.

Azure is used for DR failover (to one or more regions) while the regular day-to-day processing continues to use on-premises servers for local high availability.



Hybrid scenarios for SQL Server – SQL Server backups

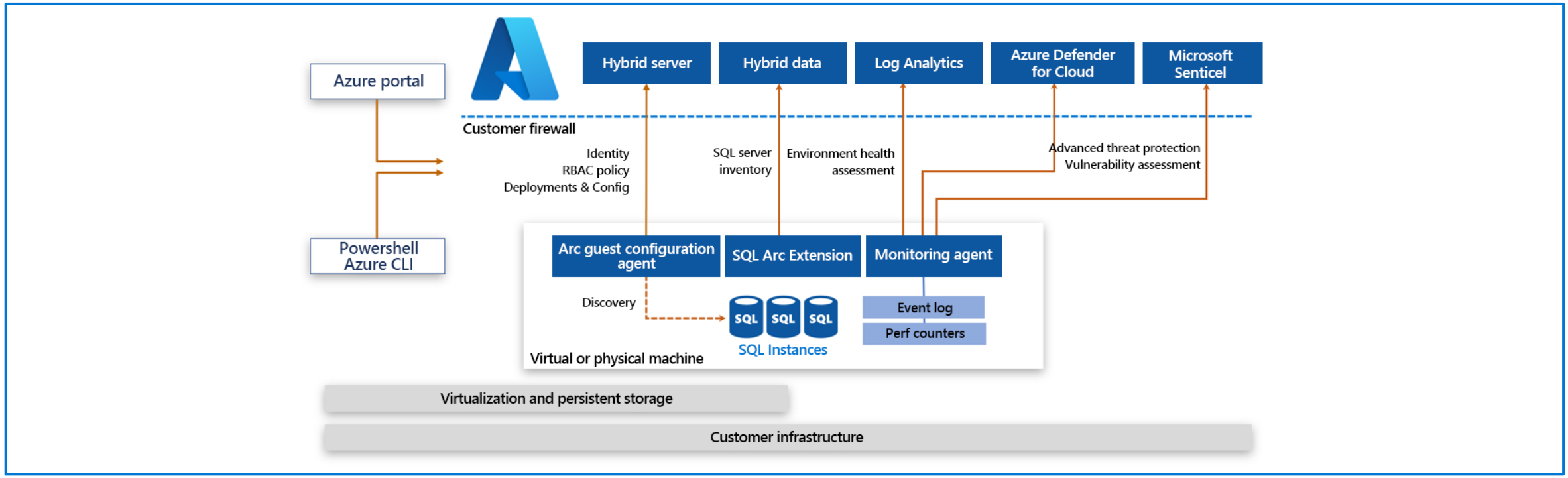
SQL Server backups

- Backups may go directly into Azure Storage via URL or Azure file share (SMB)
- Protects against data loss when on-site backup storage fails
- Can be restored to virtual machines in Azure and tested as part of Disaster Recovery procedures

Azure Storage

- Store on-premises SQL Server data files for user databases - user files and not system databases
- In the case of local storage failure, the user files are safely stored in the cloud, preventing data loss
- Built in reliability guarantees stored files in the cloud are more resilient

Hybrid scenarios for SQL Server – Azure Arc enabled SQL Servers



Extends and centralizes

Enables the inventory

Security threat introspection

Azure Virtual Machine storage

Each Azure VM has two or more disks:

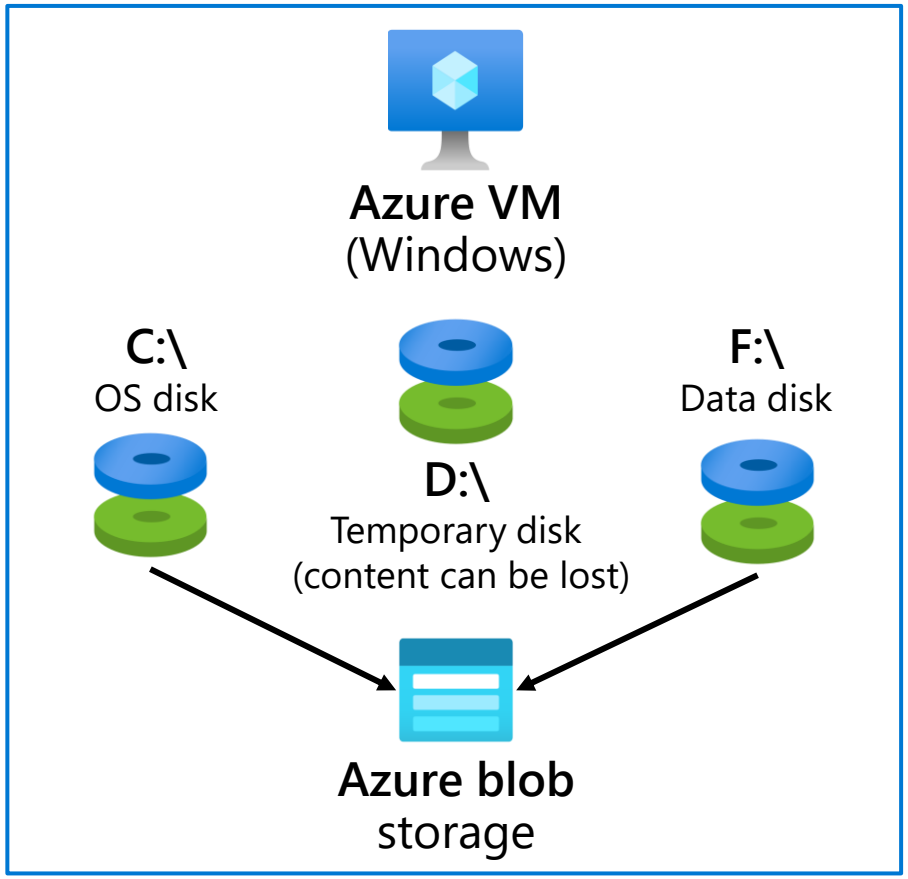
- OS disk
- Temporary disk
- Data disks (optional)

OS and data disks reside in Azure-based storage service:

- Standard (HDD, SSD)
- Premium (SSD)
- Ultra (SSD)

When creating an Azure VM, you can choose between:

- Managed disks (recommended)
- Unmanaged disks



Azure Virtual Machine storage cont'd

SQL Server workloads on Azure should use Premium SSD or Ultra SSD

You should separate transaction logs and data files onto their own volumes, with read-caching enabled on the data file volumes

You can stripe multiple disks using Storage Spaces or Logical Volume Management to get increased IOPs and storage volumes

Performance consideration – Table and index partitioning

- Can improve query performance of large tables, while improving performance and scalability
- Used when the table becomes large enough that starts compromising query performance
- Maintenance operations on a partitioned table will reduce maintenance duration

Four main components required: filegroups, partition function, partition schema and table

```
-- Partition function
CREATE PARTITION FUNCTION PartitionByMonth (datetime2)
    AS RANGE RIGHT
    -- The boundary values defined is the first day of each month
    FOR VALUES ('20210101', '20210201', '20210301',
      '20210401', '20210501', '20210601', '20210701',
      '20210801', '20210901', '20211001', '20211101',
      '20212101');

-- The partition scheme below will use the partition function cre
CREATE PARTITION SCHEME PartitionByMonthSch
    AS PARTITION PartitionByMonth
    TO (FILEGROUP1, FILEGROUP2, FILEGROUP3, FILEGROUP4,
        FILEGROUP5, FILEGROUP6, FILEGROUP7, FILEGROUP8,
        FILEGROUP9, FILEGROUP10, FILEGROUP11, FILEGROUP12);

-- Creates a partitioned table called Order that applies Partitio
CREATE TABLE Order ([Id] int PRIMARY KEY, OrderDate datetime2)
    ON PartitionByMonthSch (OrderDate) ;
GO
```

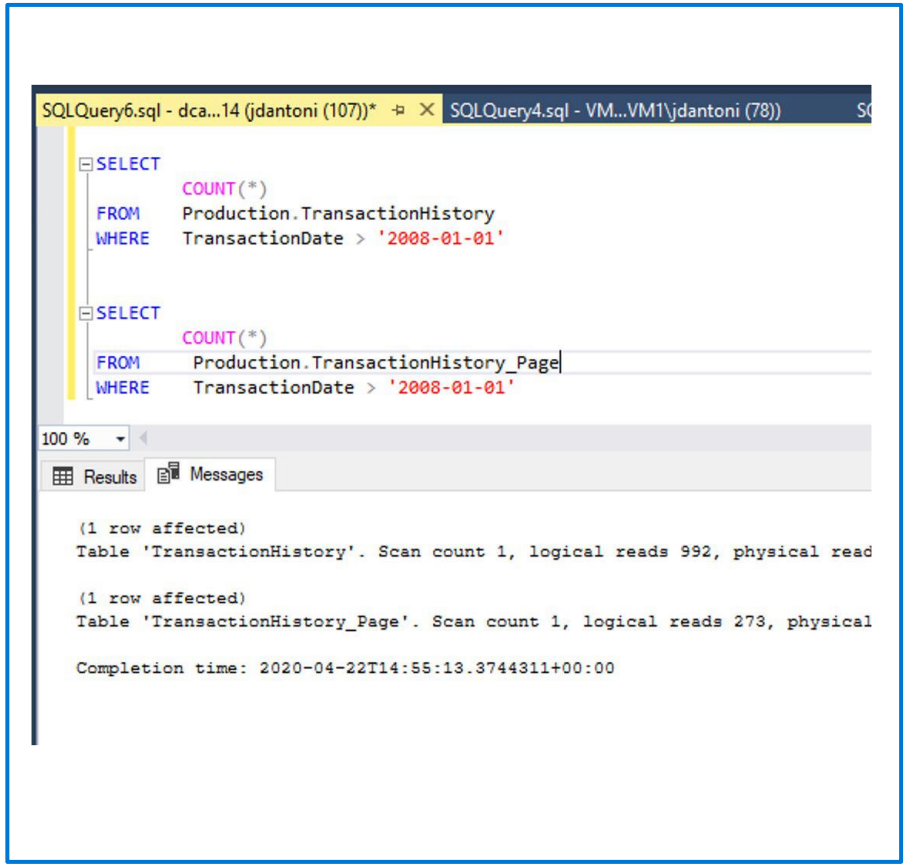
Performance consideration – Data compression

In this example, both tables have clustered and nonclustered indexes

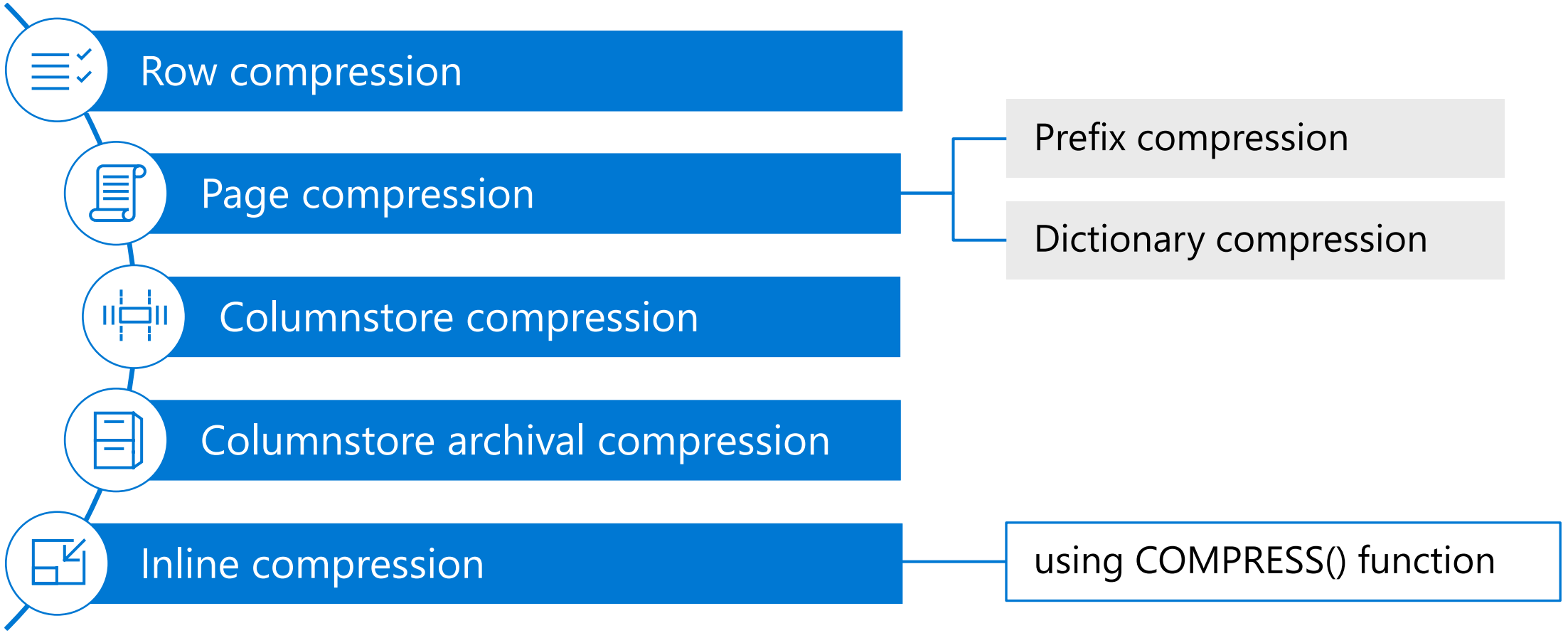
Production.TransactionHistory_Page table is **page compressed**

The query against the page compressed object performs **27% fewer logical reads** than the uncompressed table

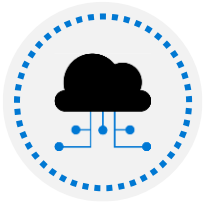
Compression is implemented at the index, table or partition level



Performance consideration – Data compression cont'd



Performance consideration – Additional options



For overall production workload:

- Enable backup compression
- Enable instant file initialization for data files
- Limit autogrowth of the database
- Disable autoshrink/autoclose for the databases
- Move all databases to data disks, including system databases
- Move SQL Server error log and trace file directories to data disks
- Set max SQL Server memory limit
- Enable lock pages in memory
- Enable optimize for adhoc workloads for OLTP heavy environments
- Enable Query Store
- Schedule SQL Server Agent maintenance jobs
- Monitor and manage the health and size of the transaction log files

Azure platform High Availability (HA) and Disaster Recovery (DR)

Availability sets

Protection from planned or unplanned Azure maintenance events and local hardware outage

Availability zones

Protection from datacenter failures

Paired Azure regions

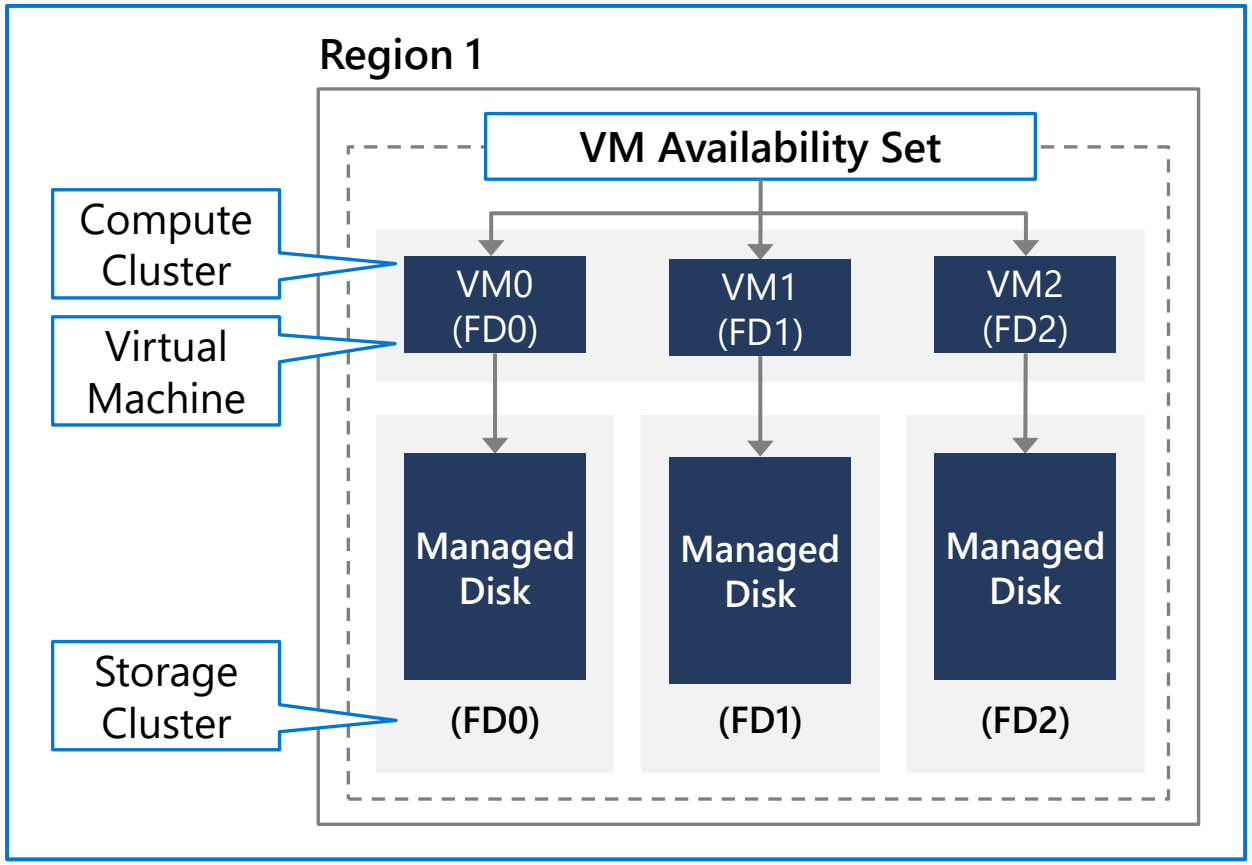
Protection from regional failures while preserving data residency and compliance boundaries

If a multi-region disaster happens, one region in each pair will be prioritized for recovery

Availability sets

Provide increased availability for multi-VM deployments by ensuring the VM are deployed to different physical hosts

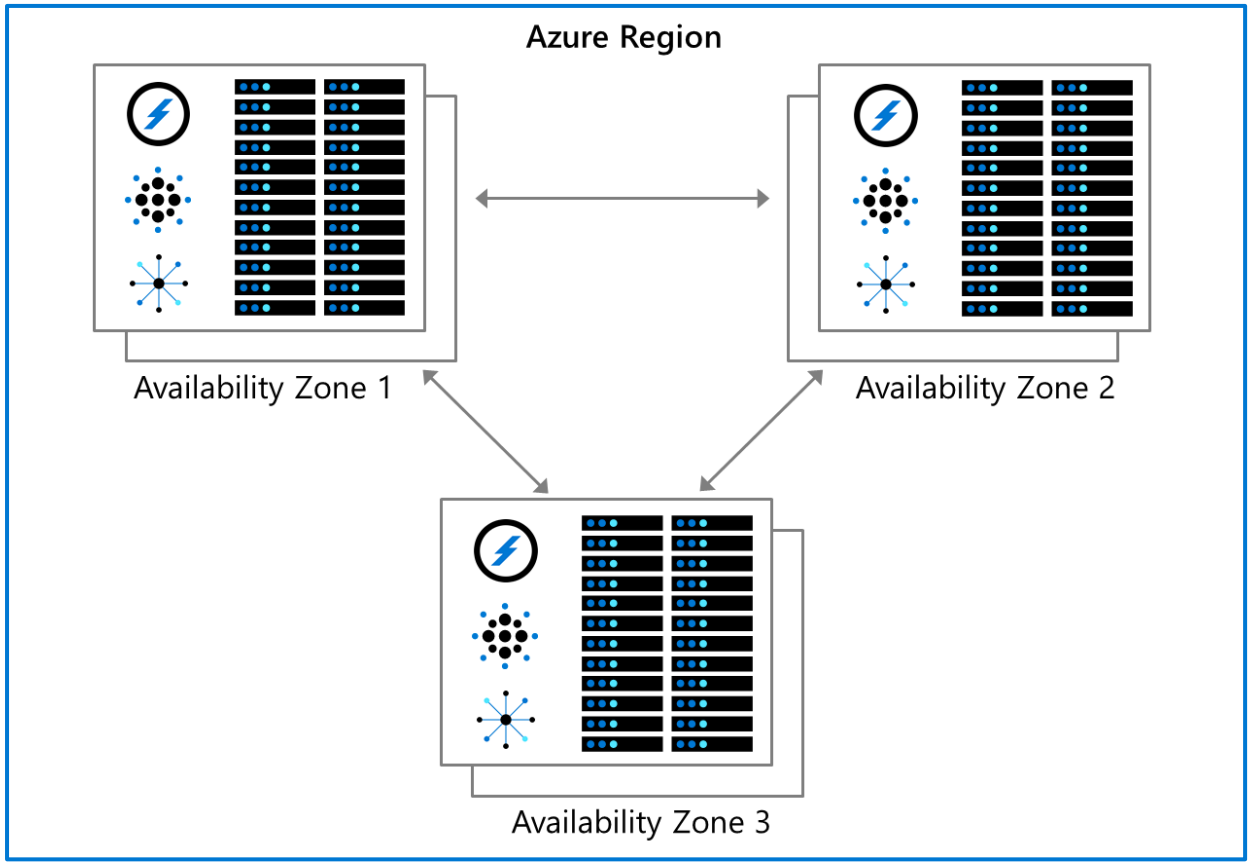
Use **fault domains** and **update domains** to protect workloads against host update failure and single points of hardware failure



Availability zones

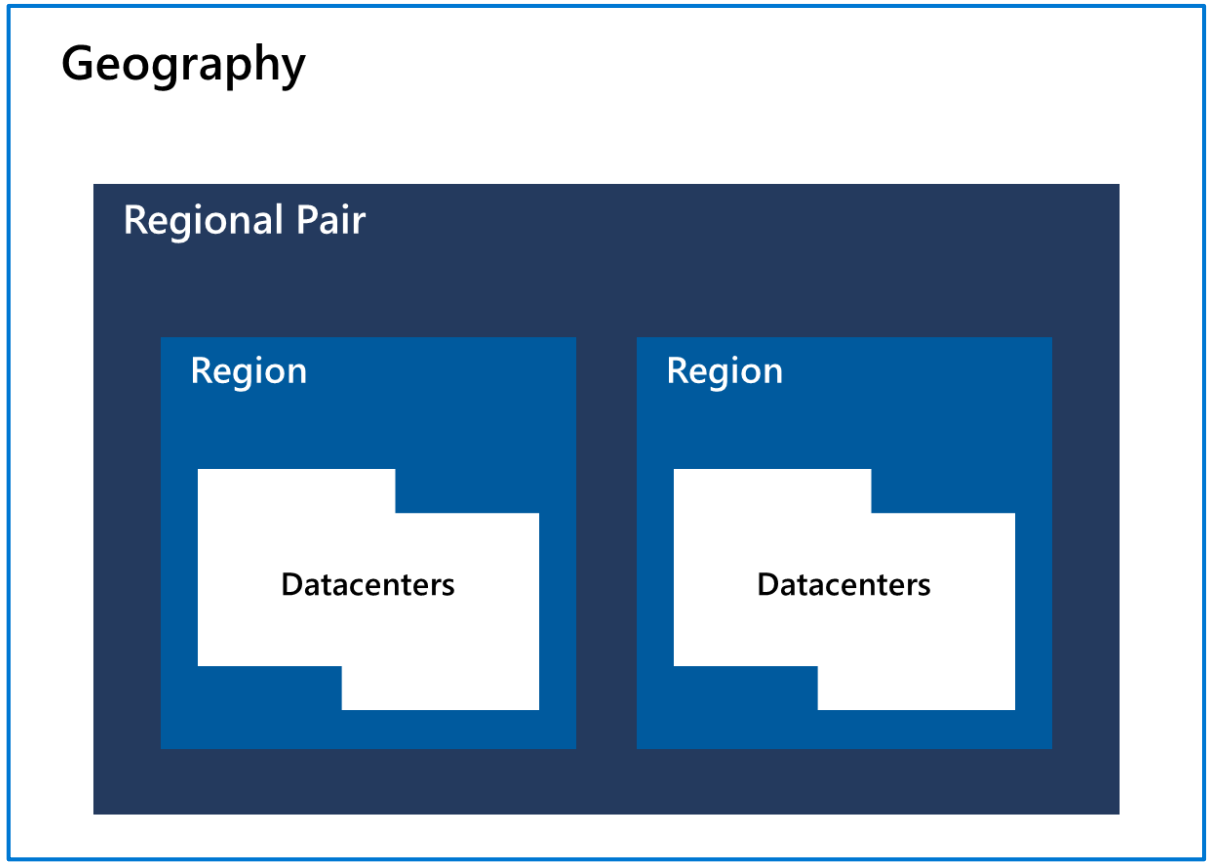
Allow workloads to be deployed to different data centers in the same Azure region

The latency between availability zones is low, and generally allows for **synchronous** data replication



Paired Azure regions

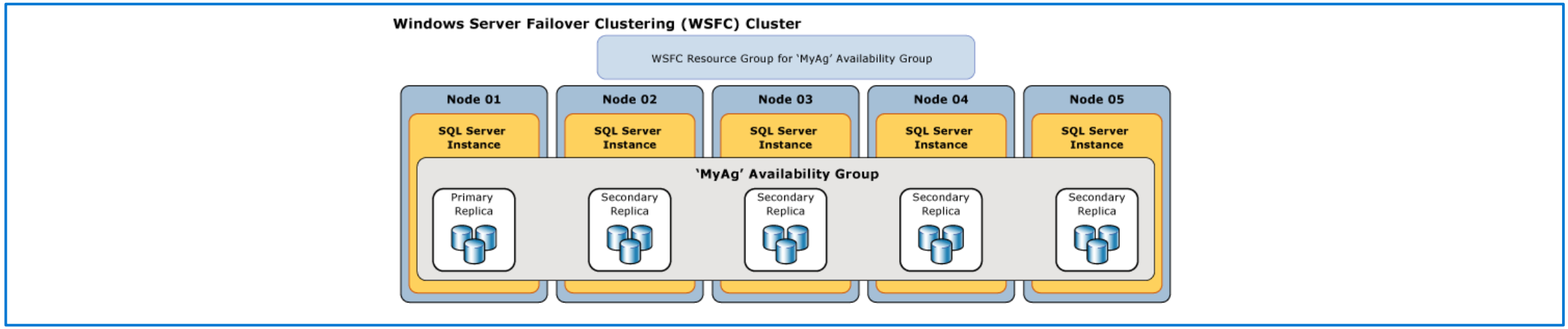
- All Azure regions have a “paired” region in which deployments and updates are applied
- If a multi-region disaster happens, one region in each pair will be prioritized for recovery
- Regional pairs are set and cannot be changed



Always On Availability Groups

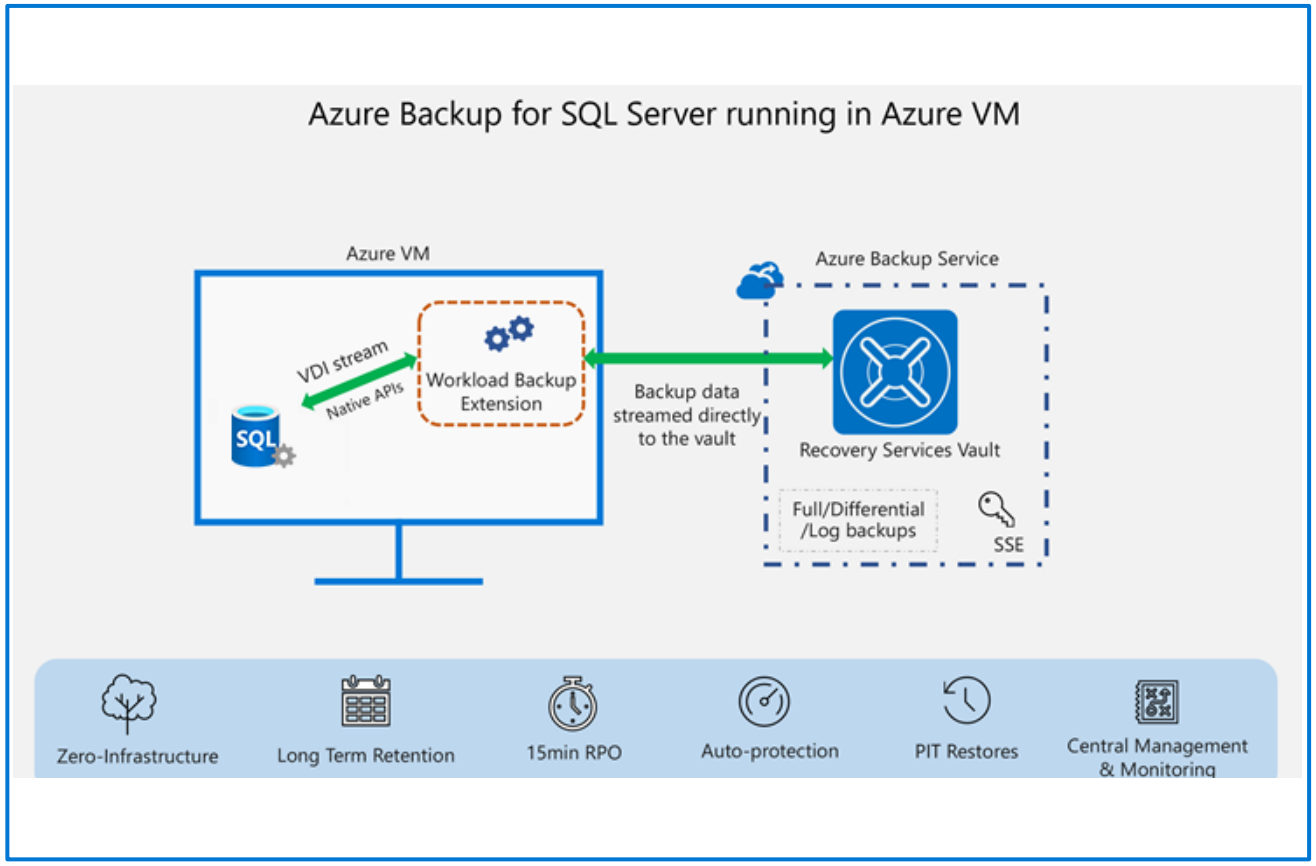
Database transactions are committed to the primary replica, and then the transactions are sent either synchronously or asynchronously to all secondary replicas

- **Asynchronous** mode - If the secondary replicas are geographically separate
- **Synchronous** mode - If the replicas are within the same Azure region



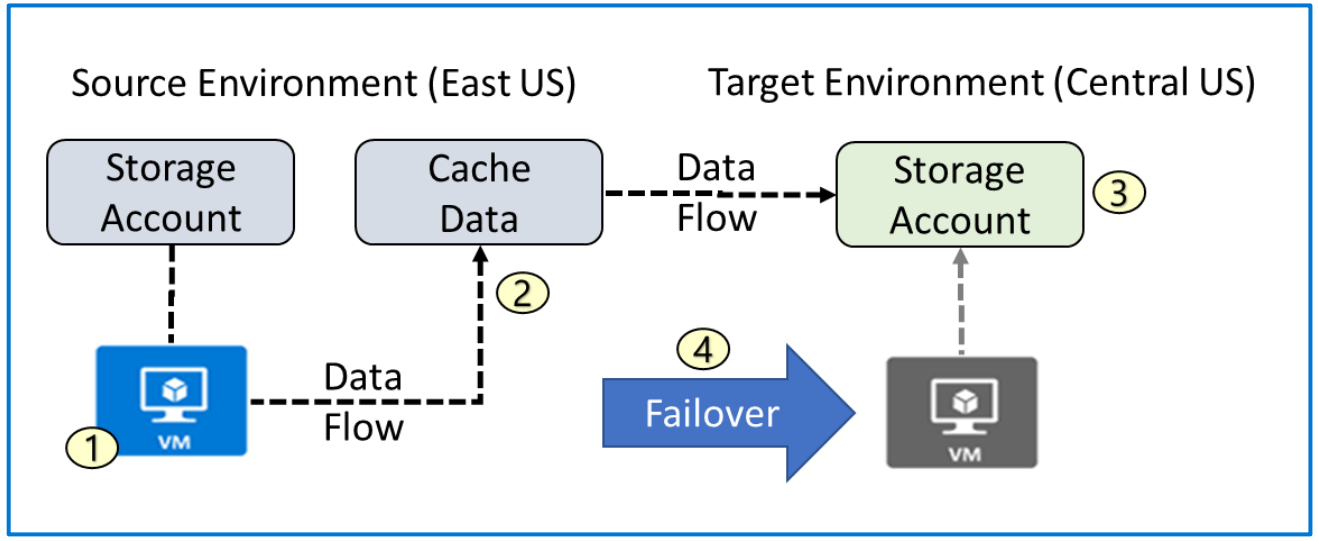
Azure Backup for SQL Server

- Provides long-term data retention, automated management and additional data protection
- Offers a more complete backup feature set
- Requires an agent to be installed on the virtual machine
- Central location to manage and monitor the backups



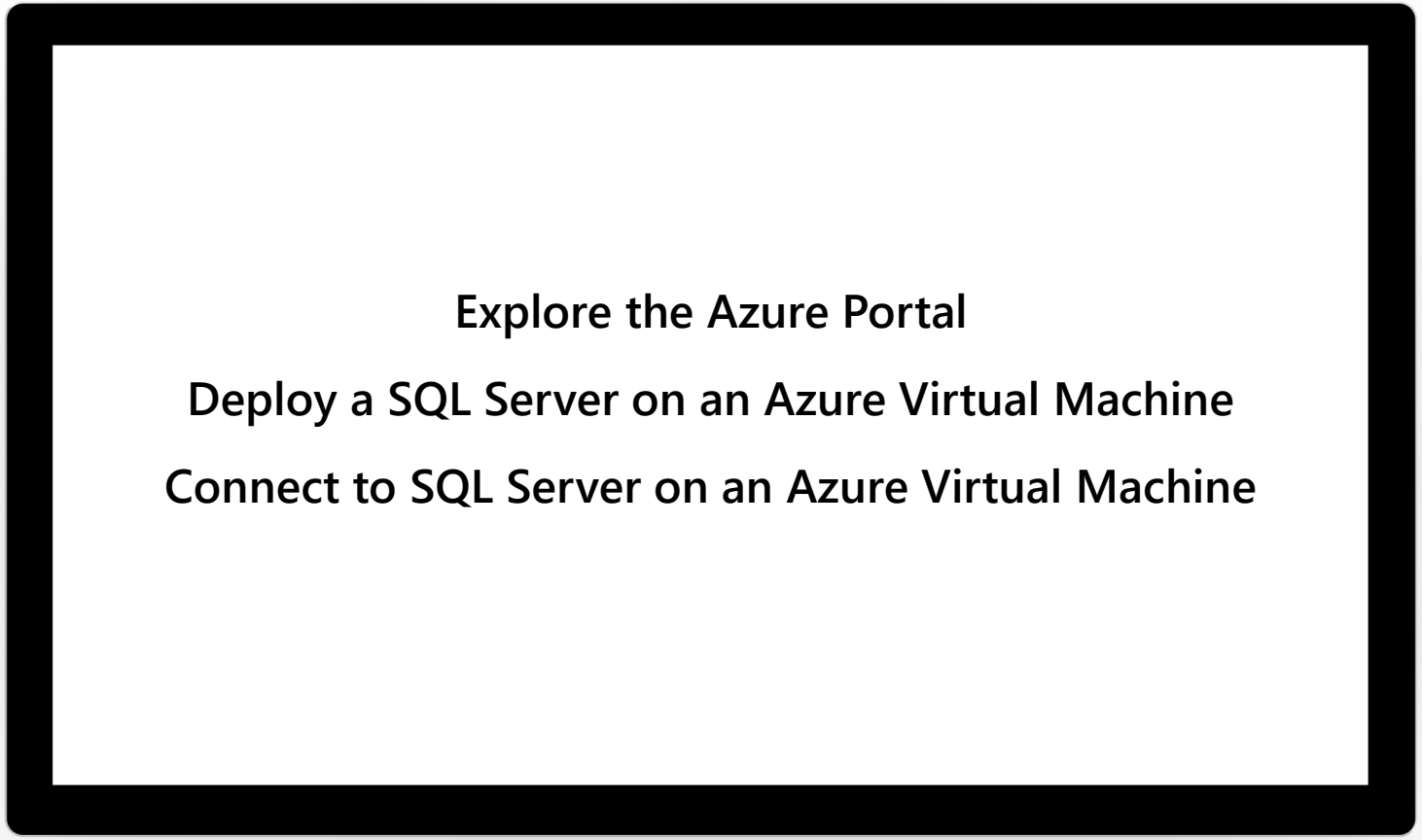
Azure Site Recovery (ASR)

- Low-cost solution that performs block level replication of your Azure virtual machine
- Better suited for migrations with some allowed downtime
- Use with Availability Groups to provide a lower RPO



1. VM is registered with Azure Site Recovery
2. Data is continuously replicated to cache
3. Cache is replicated to the target storage account
4. During failover the virtual machine is added to the target environment

Instructor led labs: Provision a SQL Server on an Azure Virtual Machine

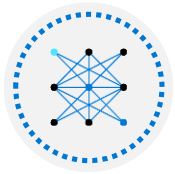


Deploy PaaS Solutions with Azure SQL

Objectives

Gain an understanding of SQL Server in a Platform as a Service (PaaS) offering

Understand PaaS provisioning and deployment options



Understand elastic pools

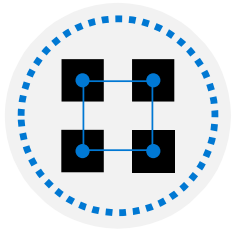
Examine Managed Instances

Configure a template for PaaS deployment

Azure SQL Database deployment models

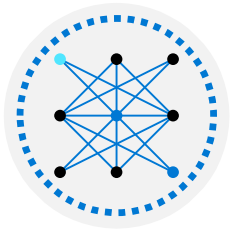
Single Database

A fully managed and isolated database that is managed and billed independently



Elastic Pool

A collection of databases that share a pool of resources



Hyperscale

A single database deployment that supports very large volumes of data

Serverless

Allows you to spend less for databases that do not need to be running 24x7

Azure SQL Database service tier options

Service tier options – DTU

The original model which uses a formula using memory, storage, and IO resources to assign a service tier

Three different service tiers:

- **Basic**
- **Standard**
- **Premium**

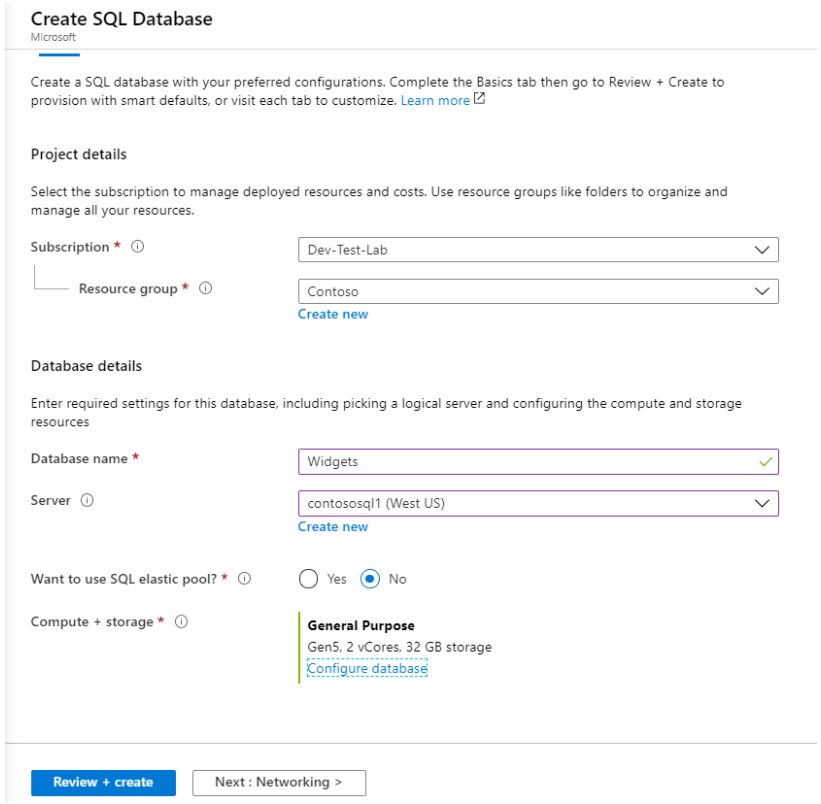
Platform as a service tiers – vCore

Allows you to choose a number of virtual CPUs, which have a fixed relationship to memory and storage provided by the database

Three different service tiers:

- **General Purpose**
- **Hyperscale**
- **Business Critical**

Deploy single database via Azure portal



Deploy single database via PowerShell/Azure CLI

PowerShell:

```powershell
# Create a server firewall rule that allows access from the specified IP range

$serverFirewallRule = New-AzSqlServerFirewallRule -ResourceGroupName $resourceGroupName `
 -ServerName $serverName `
 -FirewallRuleName "AllowedIPs" -StartIpAddress $startIp -EndIpAddress $endIp

# Create a blank database with an S0 performance level

$database = New-AzSqlDatabase -ResourceGroupName $resourceGroupName `
 -ServerName $serverName `
 -DatabaseName $databaseName `
 -RequestedServiceObjectiveName "S0" `
 -SampleName "AdventureWorksLT"
```

Azure CLI:

```azurecli
# Configure a firewall rule for the server

az sql server firewall-rule create \
 --resource-group $resourceGroupName \
 --server $servername \
 -n AllowYourIp \
 --start-ip-address $startip \
 --end-ip-address $endip

# Create a database in the server
az sql db create \
 --resource-group $resourceGroupName \
 --server $servername
 --name mySampleDatabase \
 --sample-name AdventureWorksLT \
 --edition GeneralPurpose \
 --family Gen4 \
 --capacity 1 \
```

Deploy single database via Azure Resource Manager templates

```
#Define Variables for parameters to pass to template
$projectName = Read-Host -Prompt "Enter a project name"
$location = Read-Host -Prompt "Enter an Azure location (i.e. centralus)"
$adminUser = Read-Host -Prompt "Enter the SQL server administrator username"
$adminPassword = Read-Host -Prompt "Enter the SQl server administrator password" -AsSecureString
$resourceGroupName = "${projectName}rg"

#Create Resource Group and Deploy Template to Resource Group
New-AzResourceGroup -Name $resourceGroupName -Location $location

New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName `
 -TemplateUri "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-sql-logical-serv
 -administratorLogin $adminUser -administratorLoginPassword $adminPassword

Read-Host -Prompt "Press [ENTER] to continue ..."
```

https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-sql-logical-server/azuredeploy.json

Azure SQL Database backups

Backup schedule:

- A weekly full backup
- A differential backup every 12 hours
- A transaction log backup every 5-10 minutes based on log utilization

Retention:

- **vCore and Basic Databases:** 7-day default retention, that can extend to 35 days
- **Standard and Premium Databases:** 35-day retention period
- **Long Term Retention (LTR):** Keep backups up to 10 years on Azure blob storage

Restoring an Azure SQL Database

- Cannot manually restore a database utilizing the T-SQL command RESTORE DATABASE.
- Not possible to restore over an existing database.

  Existing database must be dropped or renamed prior to initiating the restore.

Restore options:

Azure portal

PowerShell or Azure CLI

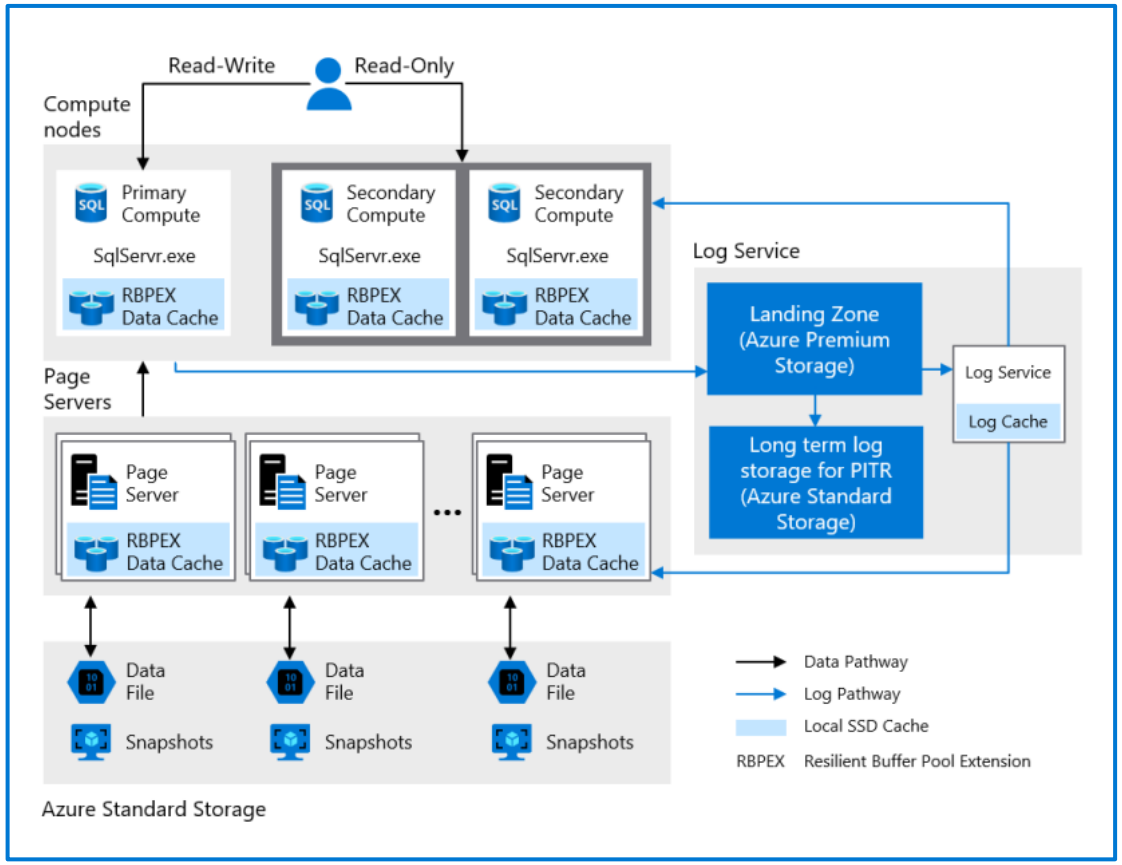
Hyperscale SQL Database

Designed for very large OLTP databases

Able to auto-scale independently storage and compute with no limits

Restores in minutes rather than hours and days

Scale up or down in real time to accommodate workload changes

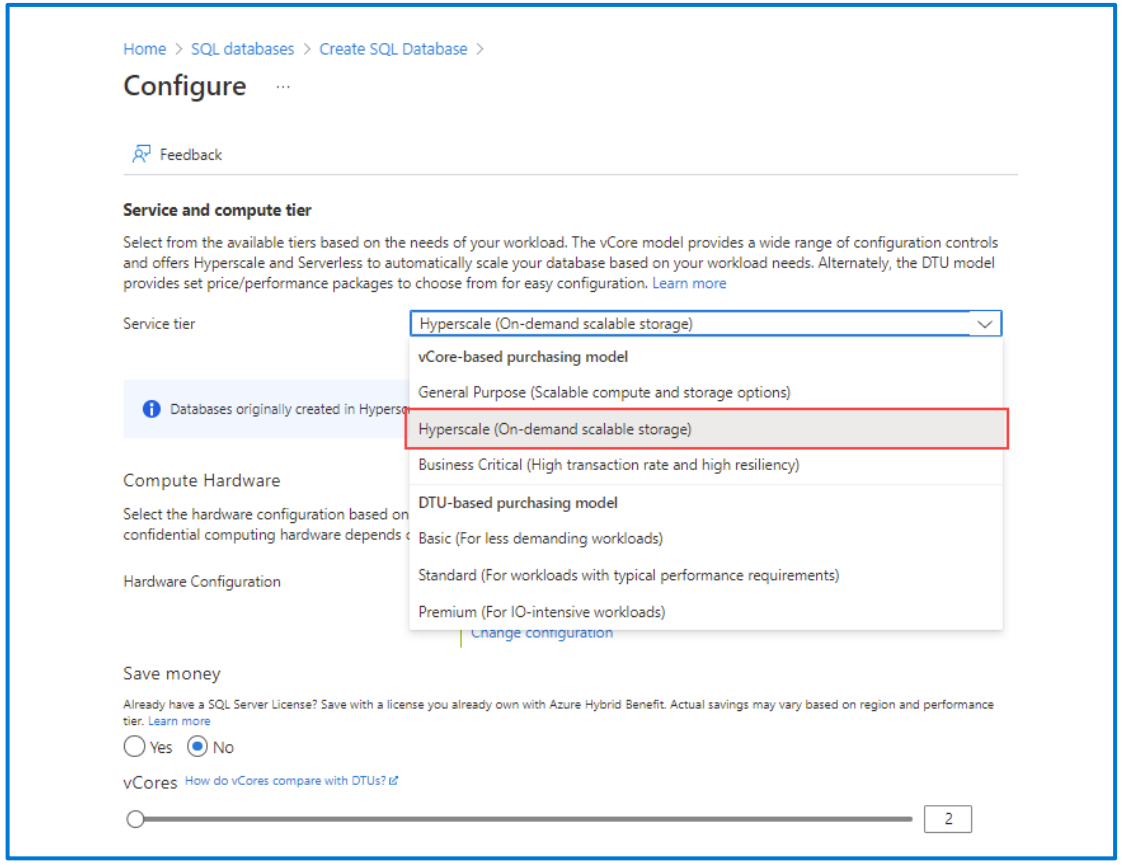


Hyperscale use cases

Large on-premises SQL Server databases that you want to modernize their apps by moving to the cloud

Using Azure SQL Database and want to significantly expand the potential for database growth

Workloads that need both high performance and high scalability

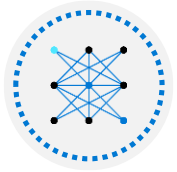


Hyperscale performance considerations

Nearly instantaneous database backups (based on file snapshots stored in Azure Blob storage) regardless of size with no IO effect on compute resources

Fast database restores – minutes rather than hours or days (not a size of data operation)

Rapid scale out – provision one or more read-only replicas for offloading read workload and for use as hot-standbys



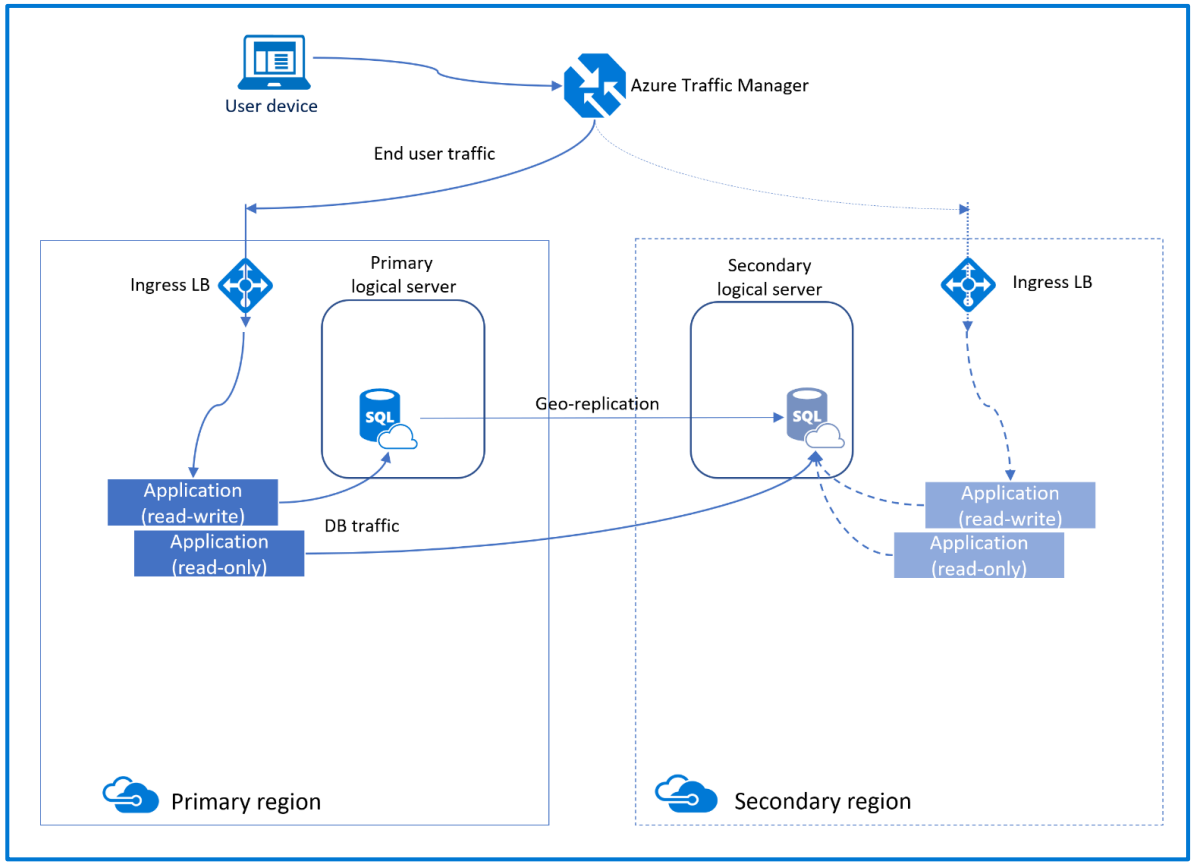
Rapid scale up – scale up your compute resources to accommodate heavy workloads when needed, and then scale the compute resources back down when not needed

Active geo-replication

Provides automatic **asynchronous** replication of committed transactions on primary database to secondary database in the Azure using **snapshot isolation**

Utilizes **Always on Availability Groups** and **Active Geo-replication**

Provides a **readable secondary copy** of your database, no matter your service tier



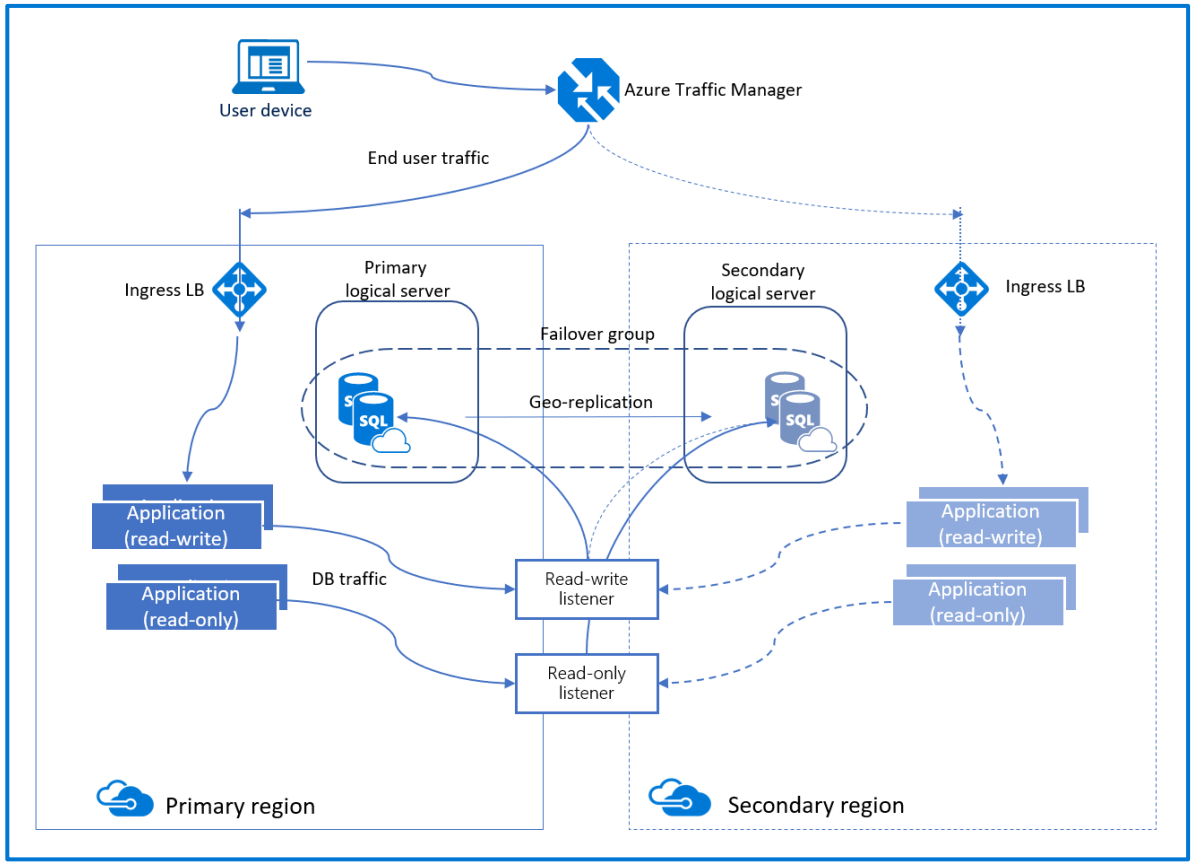
Failover groups

Builds on the geo-replication feature

Provides for automatic failover

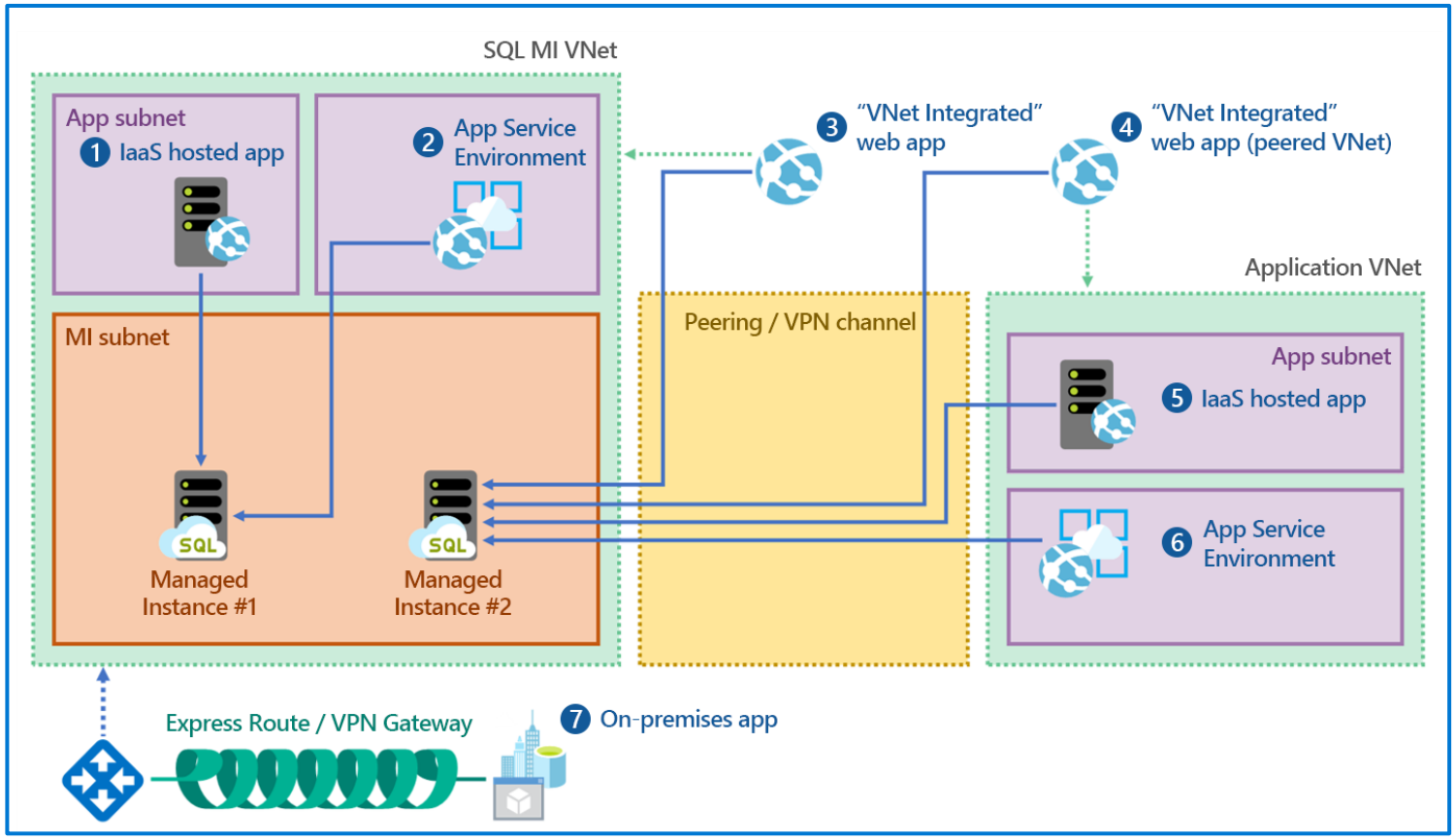
Allows a single connection string to connect to the group

Multiple databases can be in the same failover group



Azure managed networking

- SQL Managed Instance is deployed into a virtual cluster in a Virtual Network
- Connections must come from its virtual network or a peered network

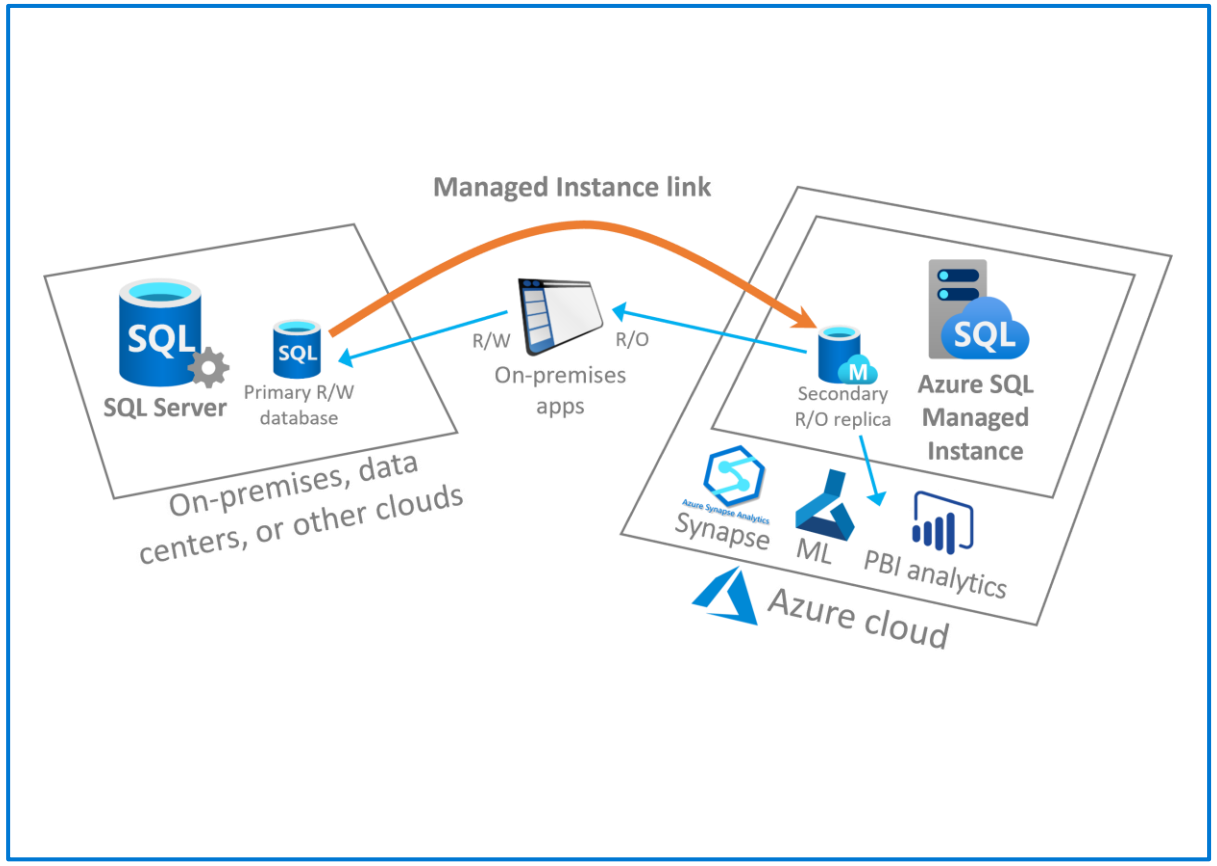


Azure SQL Managed Instance link

Connects your SQL Servers hosted anywhere to SQL Managed Instance, providing hybrid flexibility and database mobility

Supported scenarios:

- Use Azure services without migrating to the cloud
- Offload read-only workloads to Azure
- Migrate to Azure

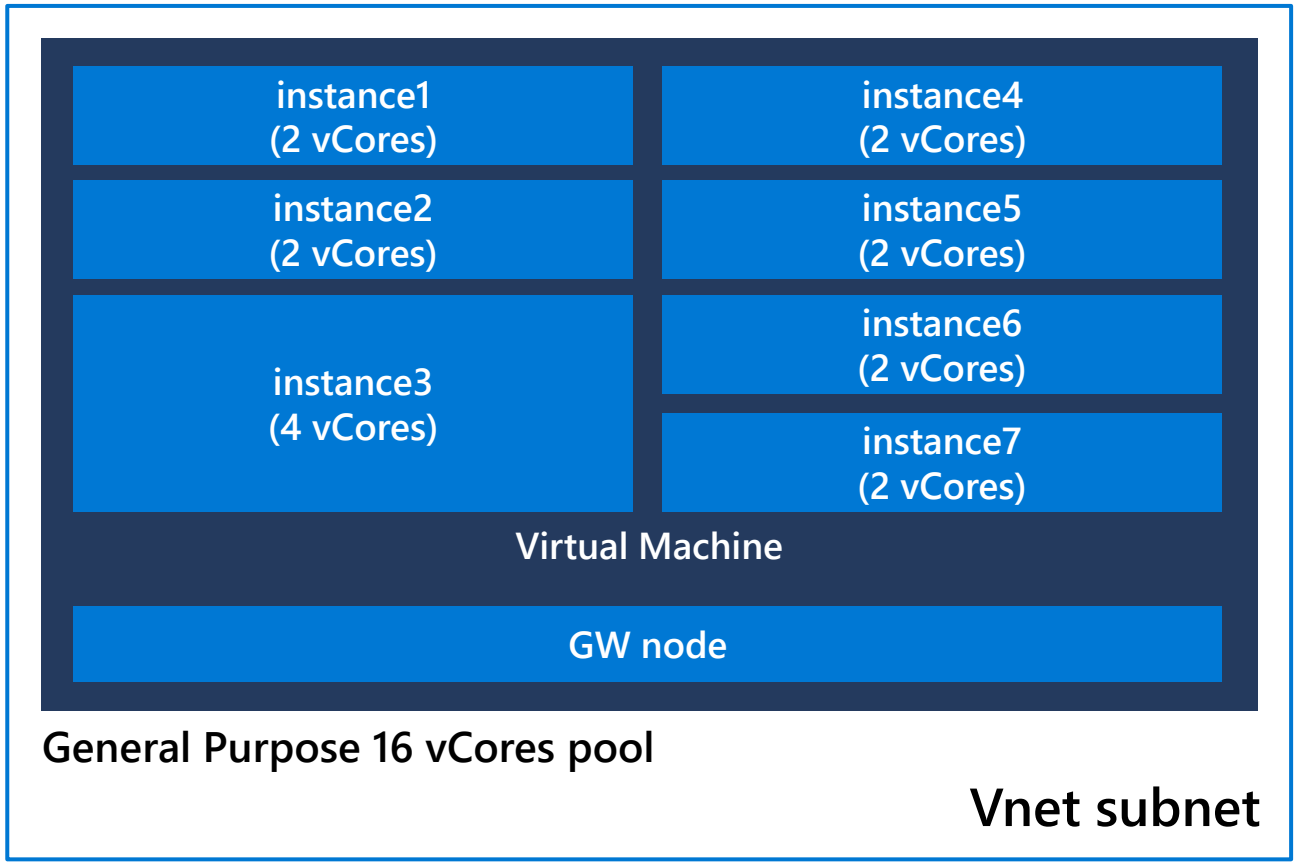


Azure SQL Managed Instance – Instance Pools

This feature supports multiple managed instances on the same virtual machine

Benefits

- Ability to host 2-vCore instances.
- Only for instances in the instance pools
- Predictable and fast instance deployment time.
- Minimal IP address allocation



Azure SQL Managed Instance backups

Nearly the same as Azure SQL Database.

Backup schedule:

- A weekly full backup
- A differential backup every 12 hours
- A transaction log backup every 5-10 minutes based on log utilization

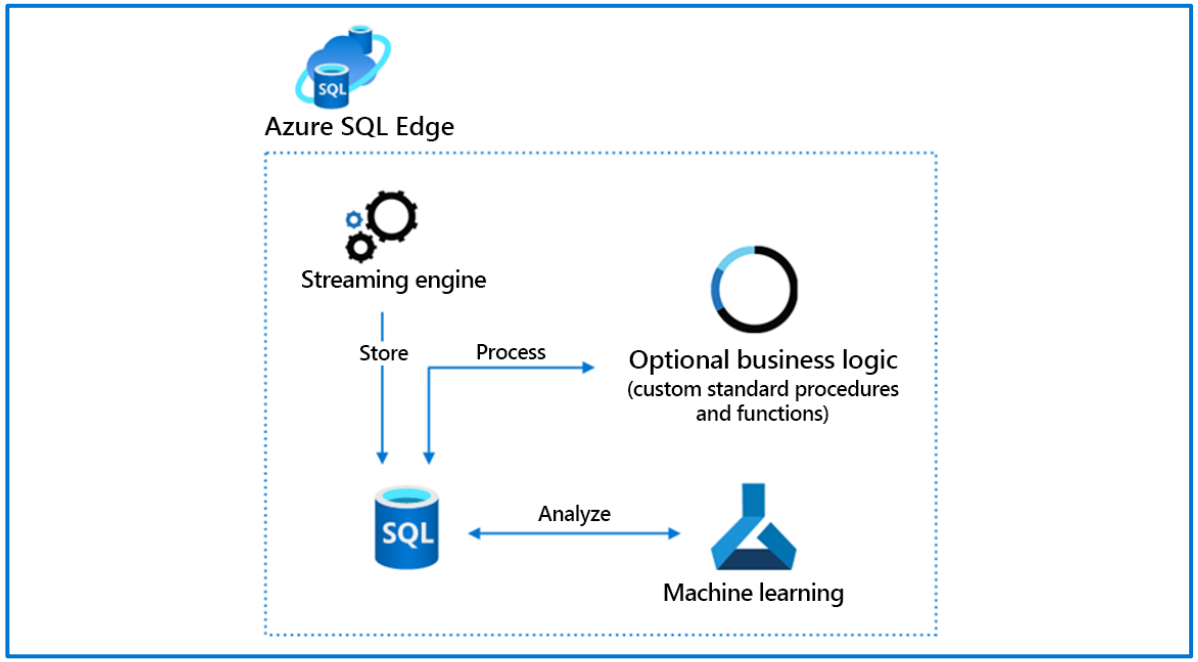
- **Retention:** 7-day default retention, that can extend to 35 days
- Backups are stored in geo-redundant storage accounts
- Ability to perform manual COPY_ONLY backups to Azure blob storage

Azure SQL Edge

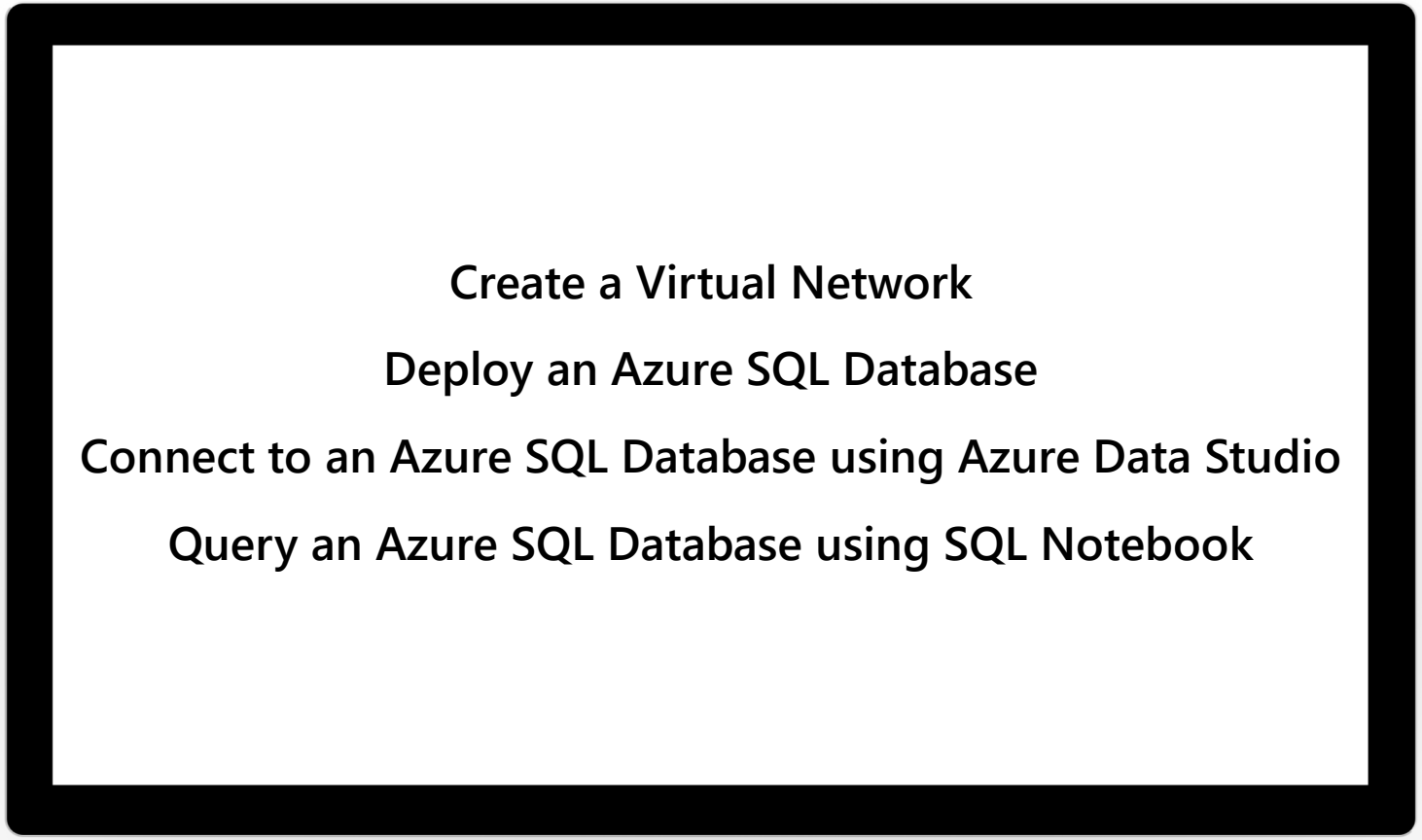
Optimized relational database engine geared for IoT and IoT Edge deployments. It is a containerized Linux application that runs on a process based on ARM64 or x64

Recommended scenarios:

- To capture continuous data streams in real time
- Integrate the data in a comprehensive organizational data solution
- Synchronize and connect to back-end systems
- Overcome connectivity limitations
- Overcome slow or intermittent broadband connection

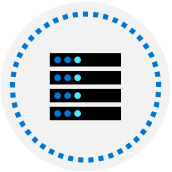


Instructor led labs: Provision an Azure SQL Database



Evaluate Strategies for Migrating to Azure SQL

Objectives



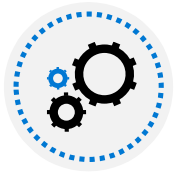
How SQL Server Compatibility Level affects database behavior

Microsoft's support policy for SQL Server

The differences between private and public preview

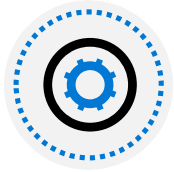
Describe the various options when you are performing a migration

Compatibility level



Database level setting

Currently (SQL Server 2019/Azure Services) supports compatibility levels 100-150 (2008-2019)



Allow query optimizer behavior and most T-SQL syntax to maintain behavior of older versions of database engine

Effects behavior of the given database, and not the entire server

SQL Server support model

SQL Server releases are in primary support for five years

This means performance, security, and functional updates in Cumulative Updates

SQL Sever provides extended support for the next five years

Security fixes will be addressed during this period

Currently supported releases of SQL Server

*Extended Support

Compatibility level-based certification for applications

Applications should stop certifying for specific version or platform (for example, SQL Server 2019 or Azure SQL Database)

Azure PaaS service versions are evergreen (always latest) so it makes the most sense to certify to a compatibility level

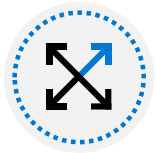
Any application certification process should be aimed at a certification level

Type of Azure preview

Private preview – Your subscription needs to be added to allowed list in order use the feature. May or may not have portal support

Public preview – Visible either in the portal, or at https://azure.microsoft.com/updates/

Preview feature caveats



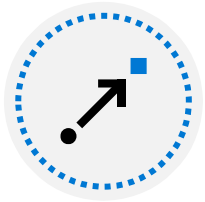
May be limited to specific regions

Preview features are often at discounted pricing

May not have full GUI support

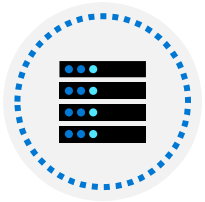
Different support policies than GA features

Migrating to Azure



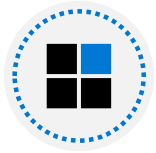
Most IaaS migrations are “Lift and Shift” - the on-premises architecture is recreated in Azure and workloads and data are migrated. Consider “Lift and Modernize”

The **Azure Migrate** tool automates migration allowing you to automatically migrate all of your workloads and data to VMs running in Azure



The **Azure Migrate** tool can discover your environment and optionally execute a migration of your databases

Database Migration Assistant (DMA)



Assesses your environment for moving from on-premises to either IaaS or PaaS

Detects issues that can affect SQL Server version upgrades or migration to Azure SQL Database or Managed Instance

Assesses the T-SQL code in your application to identity breaking changes

Recommends new features that would benefit your database

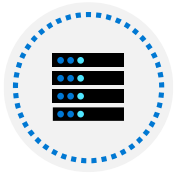
Can also migrate your database either to a VM, Azure SQL Database or Managed Instance

SQL Server migration tools

Azure SQL Database	Azure SQL Managed Instance	SQL Server on Azure Virtual Machines
Offline: Azure SQL Migration extension for Azure Data Studio, Azure Database Migration Service, Azure Migrate, BACPAC file (import), BCP, SQL Data Sync, Azure Data Factory, transactional replication	**Online:** Azure SQL Migration extension for Azure Data Studio, Azure Database Migration Service, Azure Migrate, Log Replay Service, Managed Instance link **Offline:** Native backup/restore, BACPAC file (import), BCP, transactional replication, Azure Data Factory	**Online:** Azure SQL Migration extension for Azure Data Studio, Azure Database Migration Service, Azure Migrate, distributed availability group **Offline:** Native backup/restore, BACPAC file (import), BCP, detach/attach, log shipping, Database Migration Assistant (DMA)

Migrate SQL Workloads to Azure SQL Database

Objectives

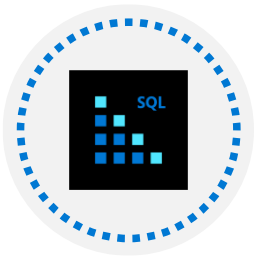


Explore options for migrating to Azure SQL Database

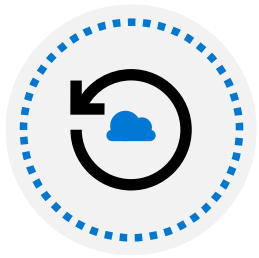
Learn how to migrate to an Azure SQL Database

Understand service continuity to Azure SQL Database

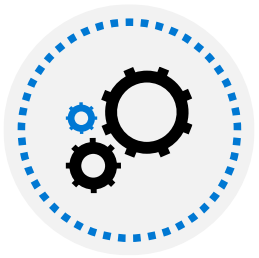
Migration benefits of SQL Server to Azure SQL Database



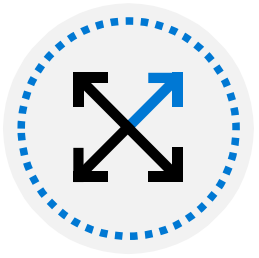
Backup and recovery



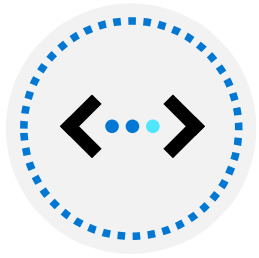
High availability



Disaster recovery



Service scalability



Security

Licensing

Tools to support your migration planning from SQL Server to Azure SQL Database

Azure Database Migration Service

Data Migration Assistant

Database Experimentation Assistant

What is Data Migration Assistant?

Helps you upgrade to a modern data platform by detecting compatibility issues that can impact database functionality in your new version of SQL Server or Azure SQL Database.

Supported Sources

- SQL Server 2005
- SQL Server 2008
- SQL Server 2008 R2
- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017 on Windows

Supported Targets

- SQL Server 2012
- SQL Server 2014
- SQL Server 2016
- SQL Server 2017 on Windows and Linux
- Azure SQL Database
- Azure SQL Managed Instance

Data Migration Assistant Configuration

Data Migration Assistant Wizard

Installation wizard is simple and just requires you to accept the license

Advanced configuration

You can fine-tune certain behavior of Data Migration Assistant by setting configuration values

Advanced Configuration can include:

Parallel assessment

```
<advisorGroup>

<workflowSettings>

<assessment parallelDatabases="8" />

</workflowSettings>

</advisorGroup>
```

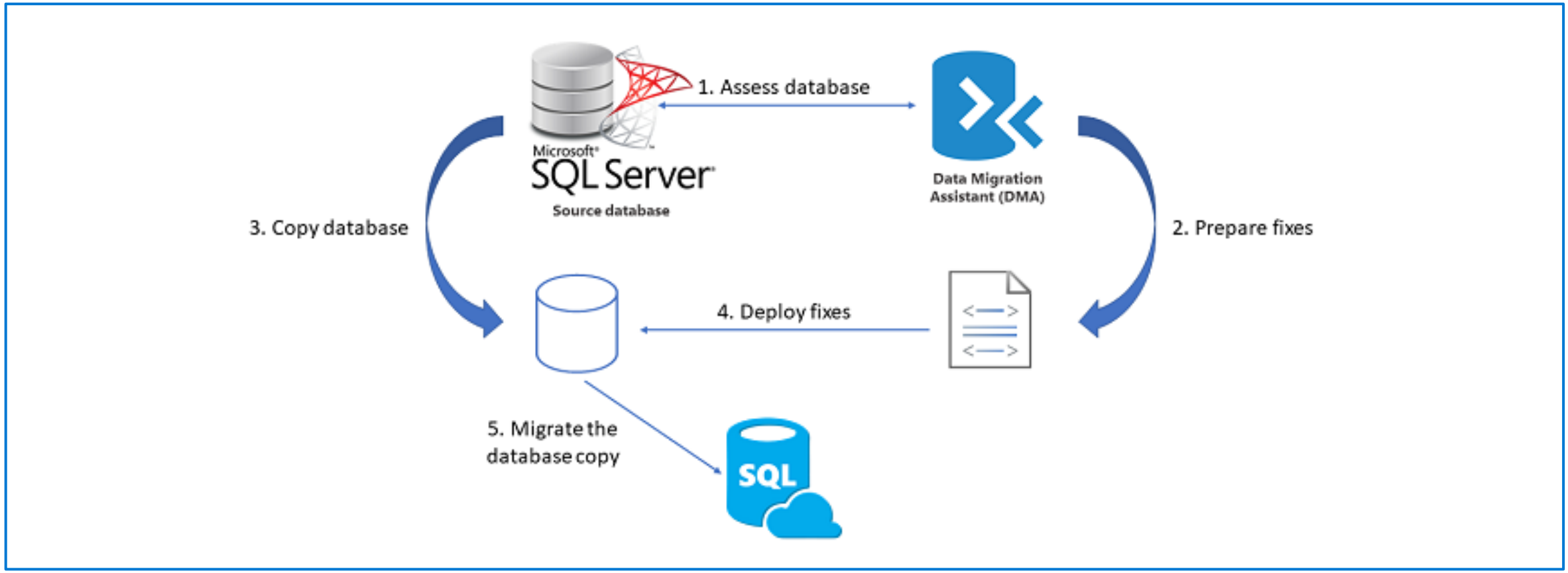
Parallel migration

```
<advisorGroup>

<workflowSettings>

<migration parallelDatabases="8" />

</workflowSettings>

</advisorGroup>
```

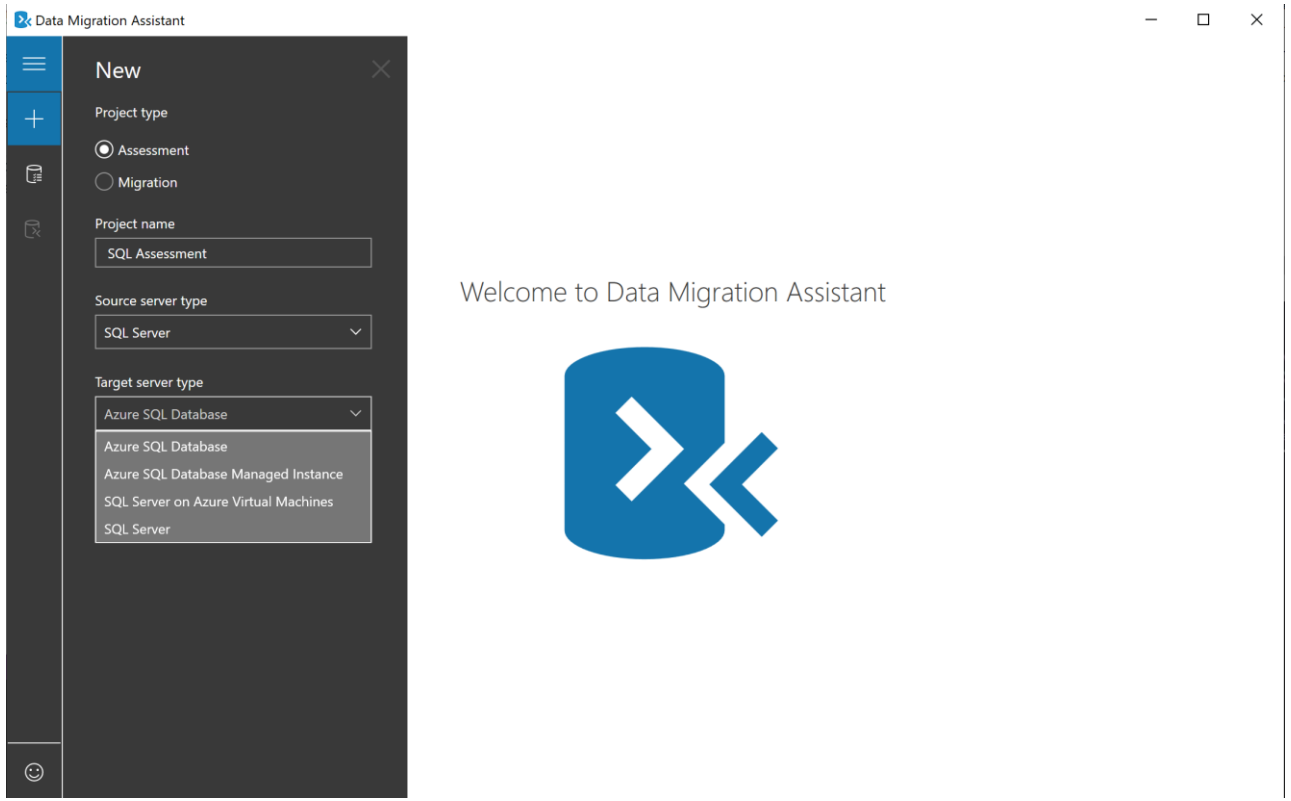
Connection time-out

```
<appSettings>

<add key="ConnectionTimeout" value="15" />

</appSettings>
```

Migrate using Data Migration Assistant



Using Data Migration Assistant



What is Database Experimentation Assistant?

Database Experimentation Assistant (DEA)

Enables A/B testing between the source system and the target system to evaluate a targeted version of SQL Server for a specific workload.

DEA outputs

Metrics include:

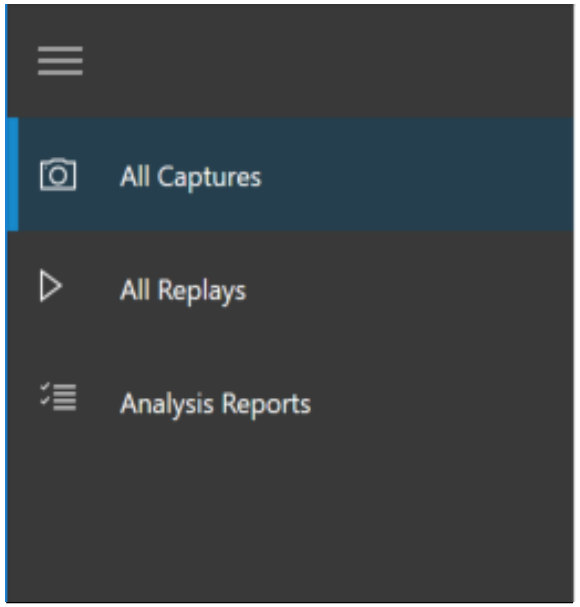
- Queries that have compatibility errors
- Degraded queries and query plans
- Other comparison data

The three tasks of the Database Experimentation Assistant are:

Capture workloads	Replay workloads	Analyze workloads
Capture trace file information from a production server to capture real world workloads.	Replay the captured trace files against a source and target server to generated information against each version.	View the results so that a workload performance comparison can be made between the source and target workloads.

Working with Database Experimentation Assistant

There are main 3 high level tasks that you perform

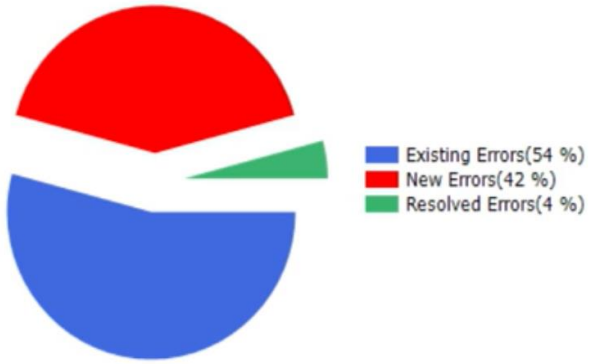
Tasks	Captures	Replays	Analyze
	You should define: • Trace name • Duration • SQL Server instance name • Database name • Path to store source trace file on SQL Server machine	You should define: • Replay name • Controller machine name • Path to source trace file on controller • SQL Server instance name • Path to store target trace file on SQL Server machine	You should define: • Report name • Trace for Target 1 SQL Server • Trace for Target 2 SQL Server

Analyzing Database Experimentation Assistant Output

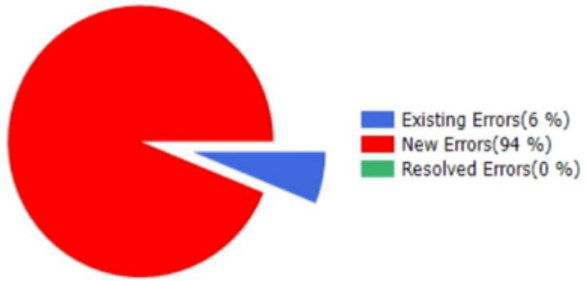
Error Queries

Source	SQL2008SOURCE
Target	SQL2016SOURCE

Errors by Distinct Queries



Errors by Execution Count



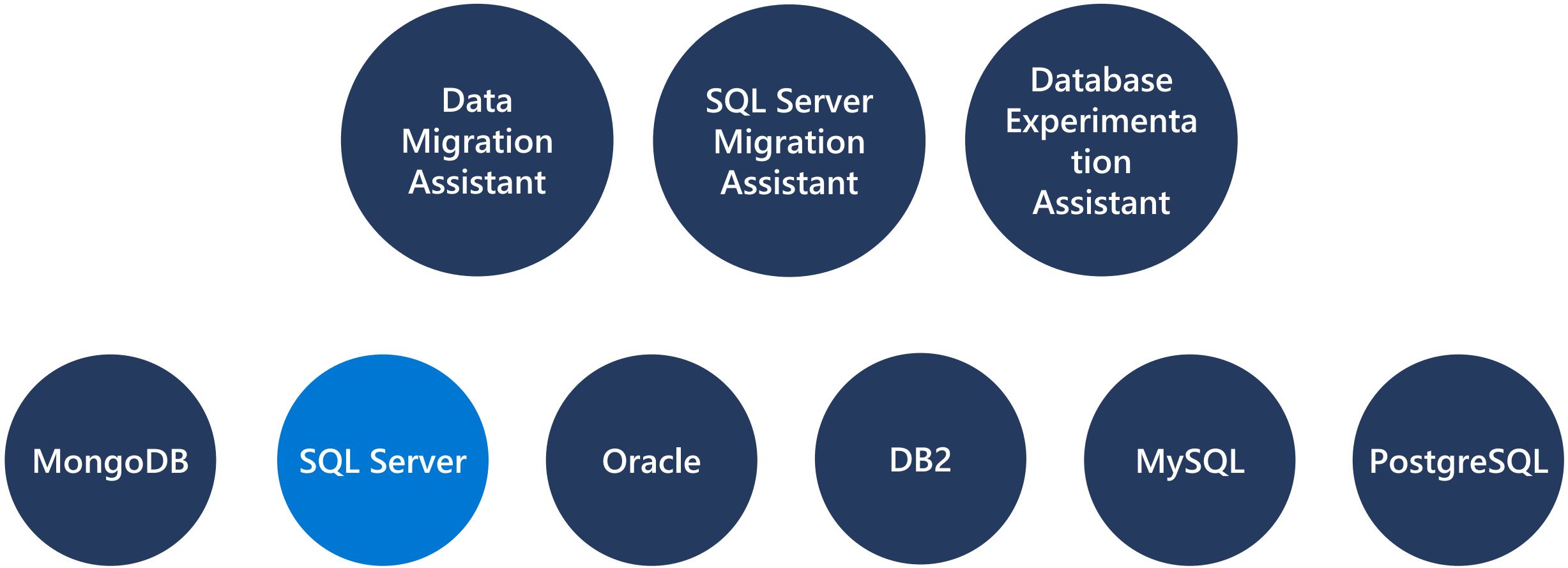
Errors Type	Query Count
Existing Errors	13
New Errors	10
Resolved Errors	1

Error Type	Execution Count
Existing Errors	751
New Errors	11453
Resolved Errors	1

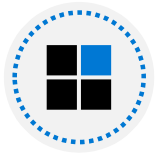
Existing Errors : Errors in source sever that continue to exist in target server
Resolved Errors : Errors that existed in source server but resolved in target server.
New Errors : New Errors found on target server

***One Error Text message shown for unique combination of Query and Error Code**

What is Azure Database Migration Service?



Prerequisites for using Azure Database Migration Service



Ensure the TCP/IP protocol is enabled in SQL Server

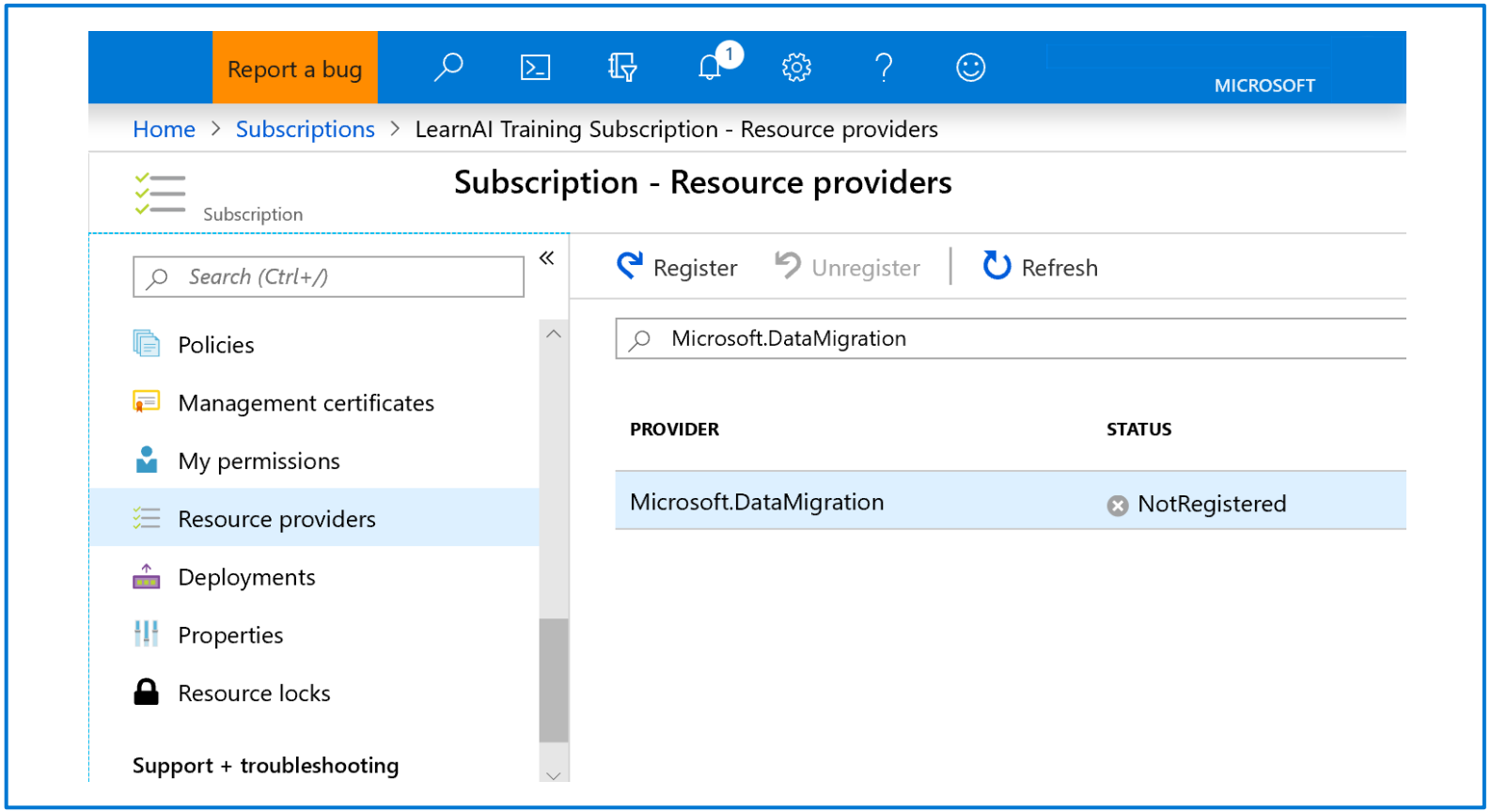
Configure your Windows Firewall for database engine access

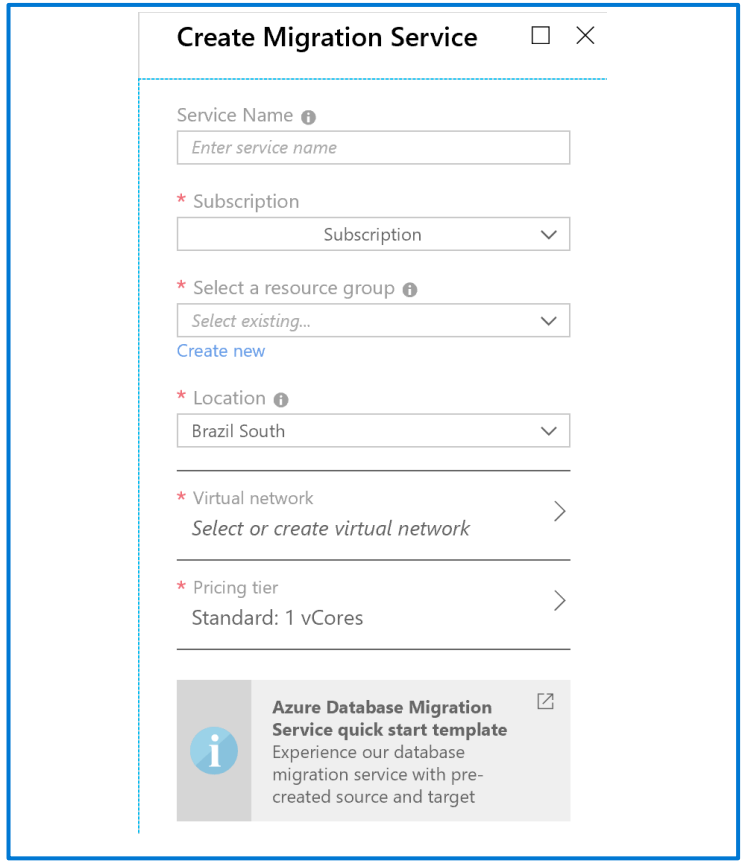
When using a firewall appliance in front of your source databases, add firewall rules

Create an Azure Virtual Network (VNet) for Azure Database Migration Service by using the Azure Resource Manager deployment model

Ensure that your VNet Network Security Group (NSG) rules don't block the following communication ports: 443, 53, 9354, 445, 12000

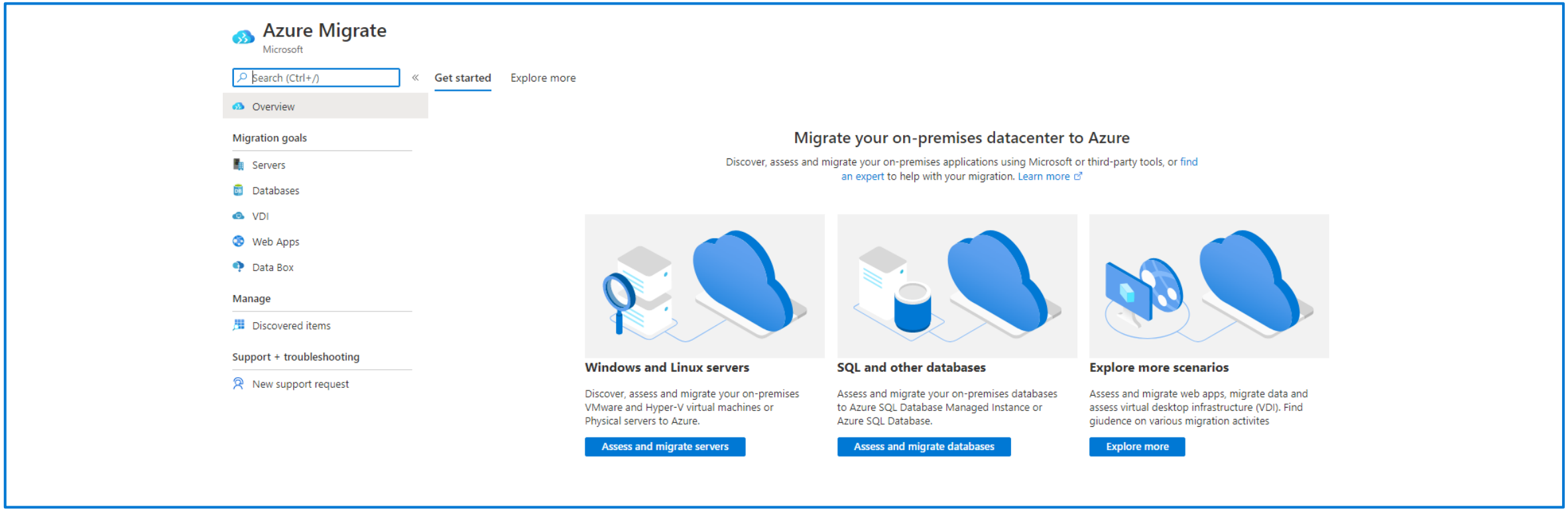
Create an instance of Azure Database Migration Service



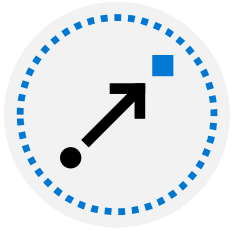


What is Azure Migrate?

Azure Migrate provides a dashboard for a suite of migration tools.

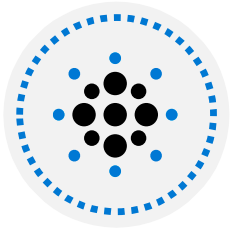


What is Azure Migrate?



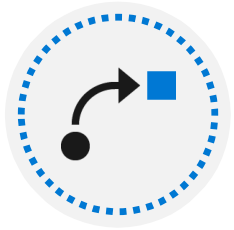
Azure Migrate

It can be used for migrations to Azure SQL Server Virtual Machines.



Assessment

You can use the Azure Migrate Service to perform assessment of source systems.

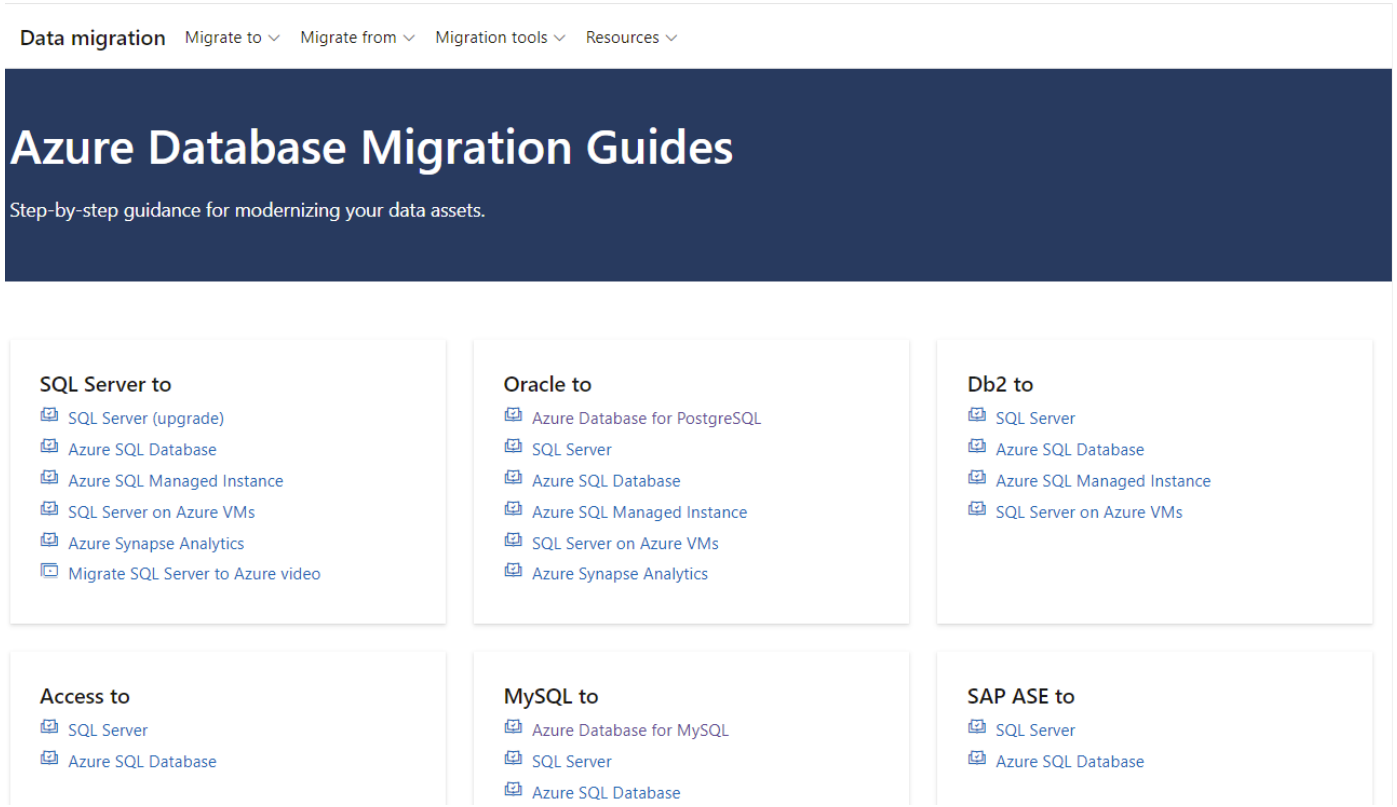


Migrations

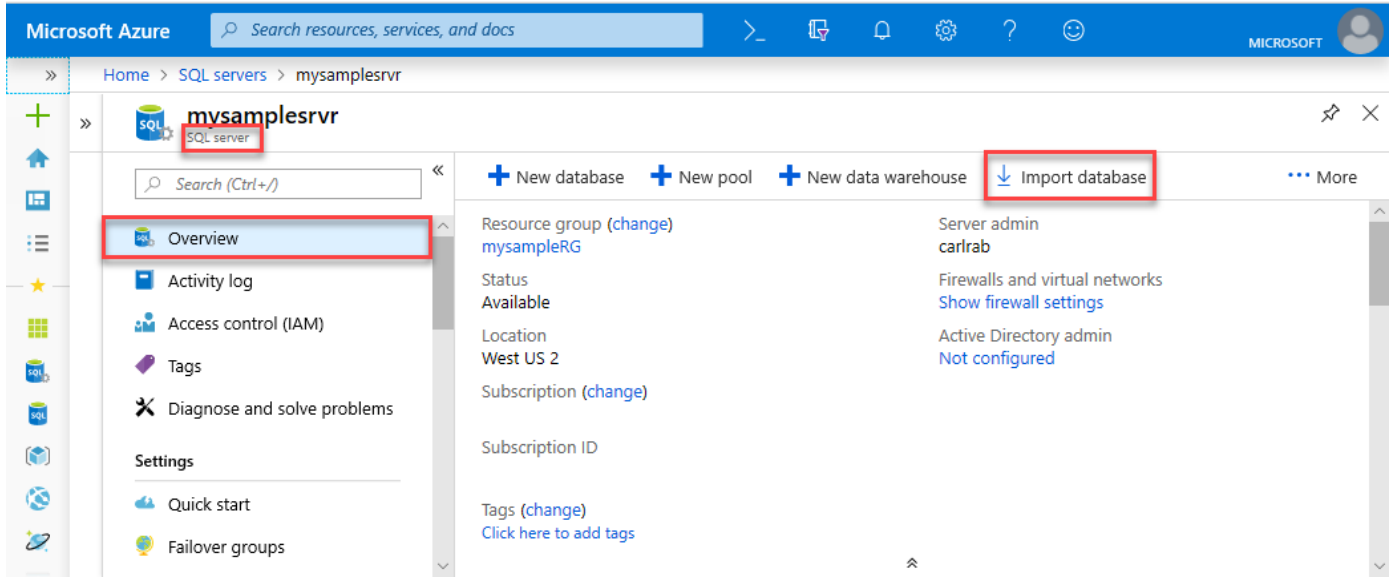
You can also use the Azure Migrate Service to perform the actual migration from VMware virtual machines.

Not to be confused with the Azure Database Migration Service.

Exploring the Database Migration Guide



Migrate using the BACPAC




```
SqlPackage.exe /a:import /tcs:"Data Source=mynewserver20170403.database.windows.net;Initial
Catalog=myMigratedDatabase;User
Id=<your_server_admin_account_user_id>;Password=<your_server_admin_account_password>"
/sf:AdventureWorks2008R2.bacpac /p:DatabaseEdition=Premium /p:DatabaseServiceObjective=P6
```

Migrate using transactional replication

Transactional replication

Transactional replication can be used to push out changes to data to a remote SQL Server.

When to use it

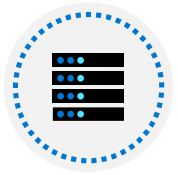
Use transactional replication when you need to minimize downtime and do not have an Always On on-premises deployment.

Three steps to migrate using transactional replication:

Distribution database	Create a publication	Create a subscription
Performed on the source database that will be migrated to a SQL Server hosted on an Azure Virtual Machine	You can selectively choose the tables within a database that you wish to publish. It can be the whole database or specific tables and views	On the SQL Server on the Azure Virtual Machine, you create a subscription that will host the migrated data

Migrate SQL Workloads to Azure SQL Managed Instance

Objectives

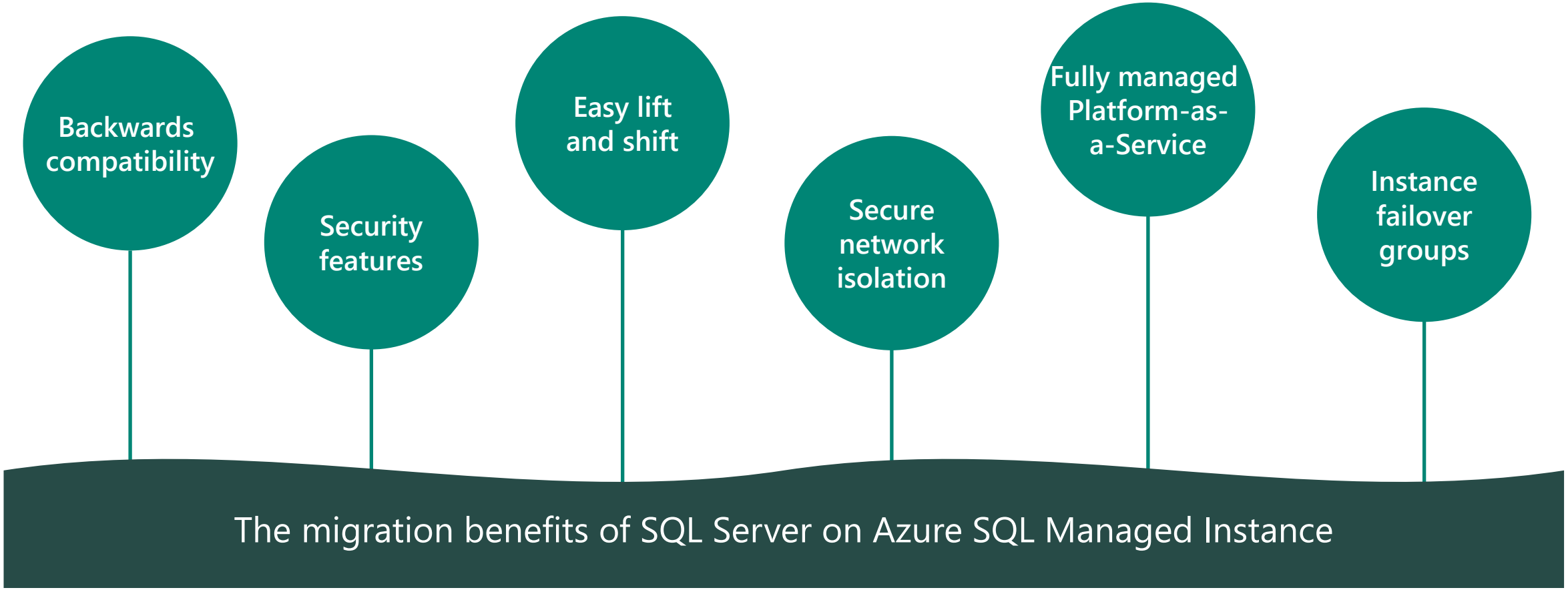


Explore options for migrating to Azure SQL Managed Instance

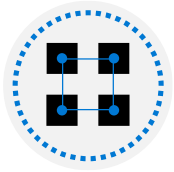
Synchronize data to Azure SQL Managed Instance

Considerations of migrating to Azure SQL Managed Instance

The migration benefits of SQL Server on Azure SQL Managed Instance



Tools to support migration planning from SQL Server to Azure SQL Managed Instance



Azure Database Migration Service

Data Migration Assistant

Database Experimentation Assistant

Evaluating SQL Managed Instance compatibility

- **Features supported Azure SQL Managed Instance include:**
  - Cross-database queries
  - Cross-database transactions
  - Linked servers
  - CLR
  - Global temp tables
  - Instance level views
  - Service Broker

- **Use the Data Migration Assessments to establish any compatibility issues**

Migrate using backup and restore from URL

BACKUP	CREDENTIAL	LIST	RESTORE	CHECK
Backup the database to an URL that points to an Azure storage account	Create a credential that accesses the storage account from SQL Managed Instance using Shared Access Signature (SAS)	Run the RESTORE FILELISTONLY FROM URL command to identify which backup to restore	Use the RESTORE DATABASE command to restore the database onto the managed instance	You can run SQL commands to check for the status of the restore operation to complete

Managing encrypted databases

Transparent data encryption

A SQL Server technology that ensures databases are encrypted at rest and can only be read when the certificate used to encrypt the data is leveraged to decrypt the database and database backups.

Transparent data encryption and Azure Key Vault

Allows you to encrypt the Database Encryption Key (DEK) with a customer-managed asymmetric key called the TDE Protector. This is also known as Bring Your Own Key (BYOK) support.

Steps to manually migrate the certificate:

Locate certificate	Backup the certificate	Copy and upload certificate
Use a Transact-SQL statement to get the database name and the certificate for each database.	Use the BACKUP CERTIFICATE statement to create a backup to a file location of your choice.	Copy the certificate to a personal information exchange and then upload to a managed instance.

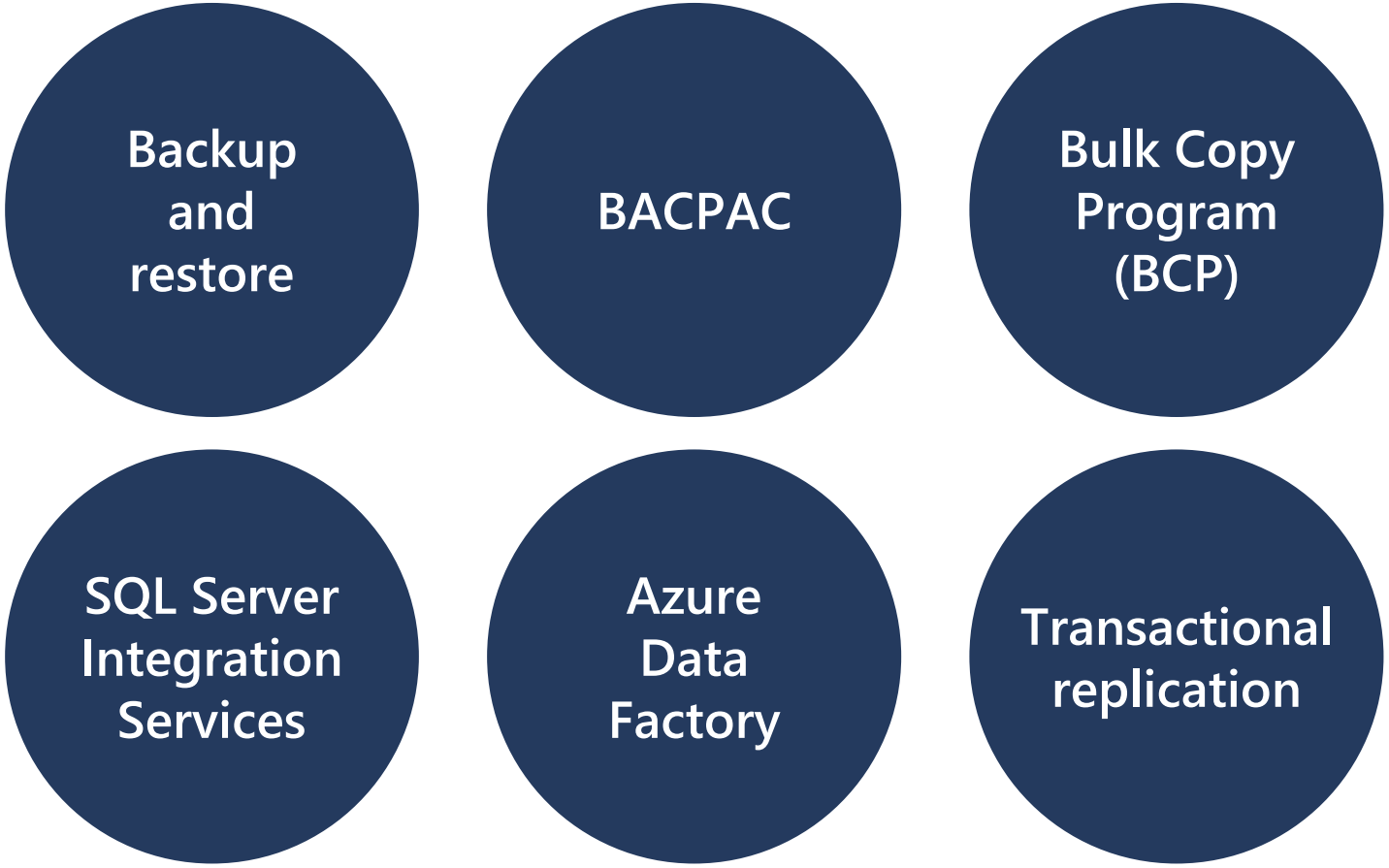
Connectivity options with on-premises servers

Create a secure connection to your virtual network from an individual client computer.

Site to site is used to connect an entire on-premises site to the Azure network.

Create private connections between Azure datacenters and on-premises infrastructure.

Options to synchronize data to and from SQL Managed Instance

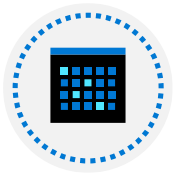


Synchronizing data using SSIS or Azure Data Factory

Continue leveraging your SSIS packages with Azure Data Factory or create a brand-new Azure Data Factory pipeline to execute your activities.

SQL Server Integration Services

ETL tool of choice for migrating data from point A to point B and taking advantage of the powerful control flow and data flow capabilities.



Azure Data Factory

Azure Data Factory is essentially a fully-managed data integration as a service in the cloud that can be leveraged for your ELT workloads.

Migrating SQL Server workloads using transactional replication

Transactional replication

Can be used to push out changes to data to a remote SQL Server.

When to use it

When you need to minimize downtime and do not have an Always On on-premises deployment.

Steps to manually migrate the certificate:

Distribution database	Create a publication	Create a subscription
Performed on the source database that will be migrated to a SQL Server hosted on an Azure Virtual Machine.	You can selectively choose the tables within a database that you wish to publish. It can be the whole database or specific tables and views.	On the SQL Server on the Azure Virtual Machine, you create a subscription that will host the migrated data.

Summary

Deploying IaaS solutions with Azure SQL:

- Learn the available options for provisioning and deployment
- Deploy SQL Server on an Azure Virtual Machine

Deploying PaaS solutions with Azure SQL:

- Understand the benefits of Platform as a Service offerings
- Understand the differences between Azure SQL offerings

Evaluating strategies for migrating to Azure SQL:

- Describe the various options when you are performing a migration
- How SQL Server Compatibility Level affects database behavior

Migrating SQL workloads to Azure SQL Database:

- Explore options for migrating to Azure SQL Database
- Understand service continuity to Azure SQL Database

Migrating SQL workloads to Azure SQL Managed Instance:

- Explore options for migrating to Azure SQL Managed Instance
- Considerations of migrating SQL Managed Instance