



# Azure SQL Managed Instance Business Continuity

Module 3



# CONDITIONS AND TERMS OF USE:

© Microsoft Corporation. All rights reserved.

You may use these training materials solely for your personal internal reference and non-commercial purposes. You may not distribute, transmit, resell or otherwise make these training materials available to any other person or party without express permission from Microsoft Corporation. URL's or other internet website references in the training materials may change without notice. Unless otherwise noted, any companies, organizations, domain names, e-mail addresses, people, places and events depicted in the training materials are for illustration only and are fictitious. No real association is intended or inferred. THESE TRAINING MATERIALS ARE PROVIDED "AS IS"; MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED IN THESE TRAINING MATERIALS.

## Learning Units covered in this Module

- Lesson 1: Business Continuity
- Lesson 2: Auto-Failover Groups
- Lesson 3: Automated Backups and Retention
- Lesson 4: Long-Term Backup Retention
- Lesson 5: Database Restores
- Lesson 6: Accelerated Database Recovery

# Lesson 1: Business Continuity Features

# Objectives

After completing this learning, you will be able to:

- Understand the various business continuity options within Azure SQL Database.
- Understand how to copy and export Azure SQL Databases.
- Understand how to perform a point-in-time restore.
- Understand how to perform a restore of a deleted database.



# Business Continuity Problems

Enabling the application to continuously operate during unplanned and planned disruptive events.

Disruption scenarios in general:

- Local hardware or software failures
- Data corruption or deletion typically caused by an application bug or human error.
- Datacenter outage, possibly caused by a natural disaster.
- Upgrade or maintenance errors.

# Basic (DTU), Standard (DTU), General Purpose (vCore) High Availability

Behaves like Failover Cluster Instance

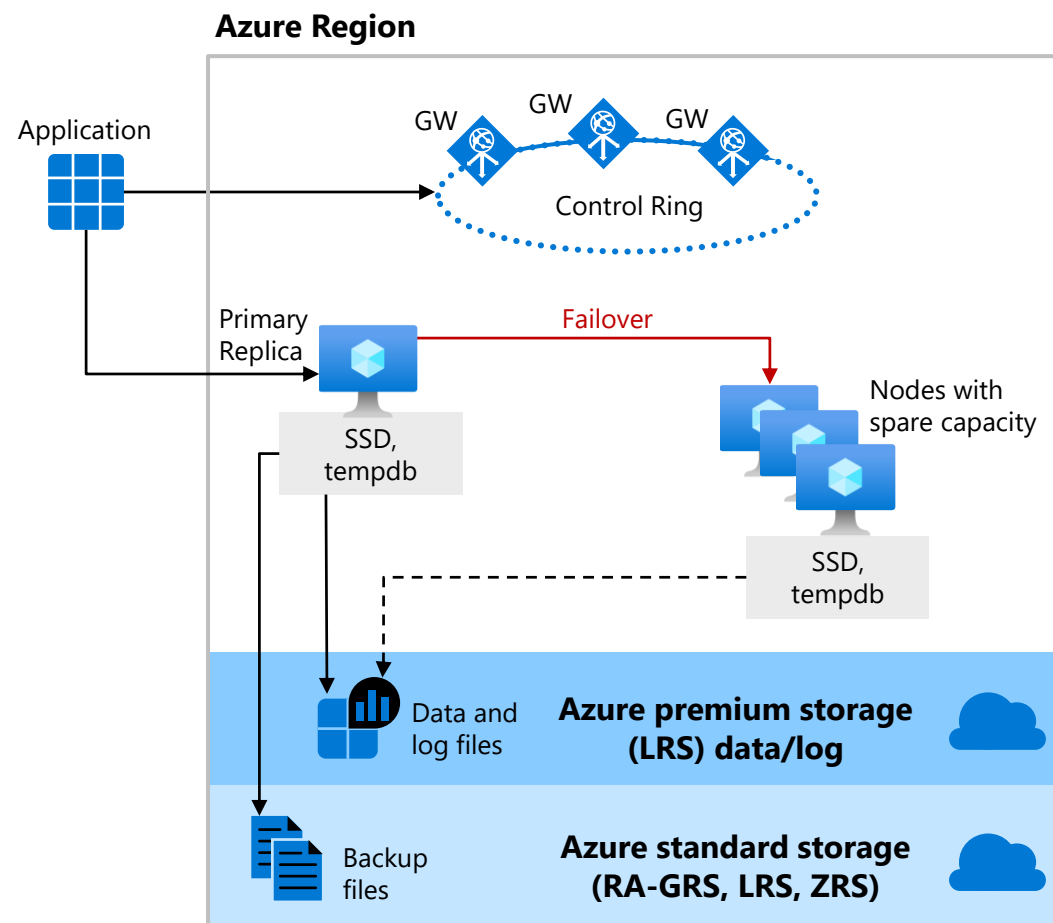
Remote storage provides data redundancy within a datacenter

Backup files are in a different location with geo-redundancy

Failover decisions based on SQL and Service Fabric

Recovery time depends on spare capacity

Connectivity redirection built-in



# Premium (DTU) and Business Critical (vCore) High Availability

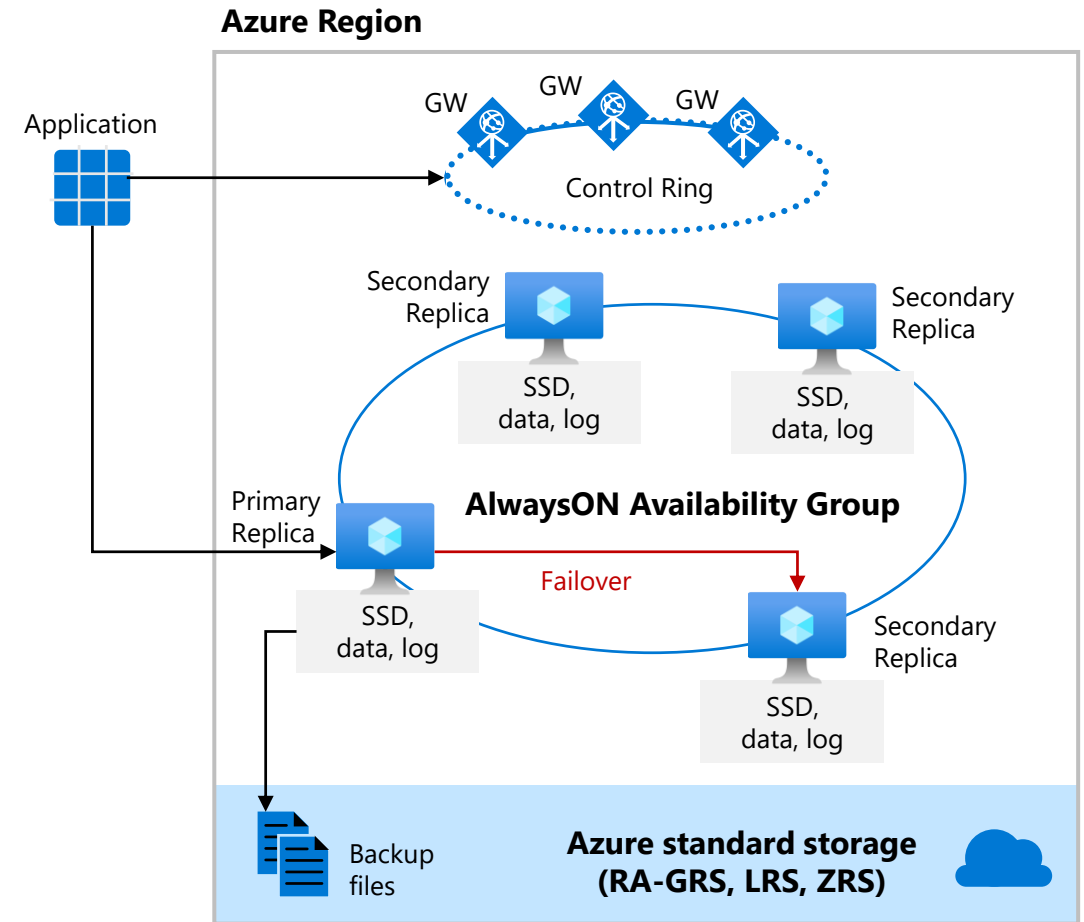
High availability is achieved by replicating both compute and storage to additional nodes.

High availability is implemented using a technology like SQL Server Always On Availability Groups.

The cluster includes a single primary replica for read-write workloads, and up to three secondary replicas (compute and storage) containing copies of data.

The failover is initiated by the Azure Service Fabric.

As an extra benefit, the premium availability model includes Read Scale-Out feature.





# Zone redundant configuration

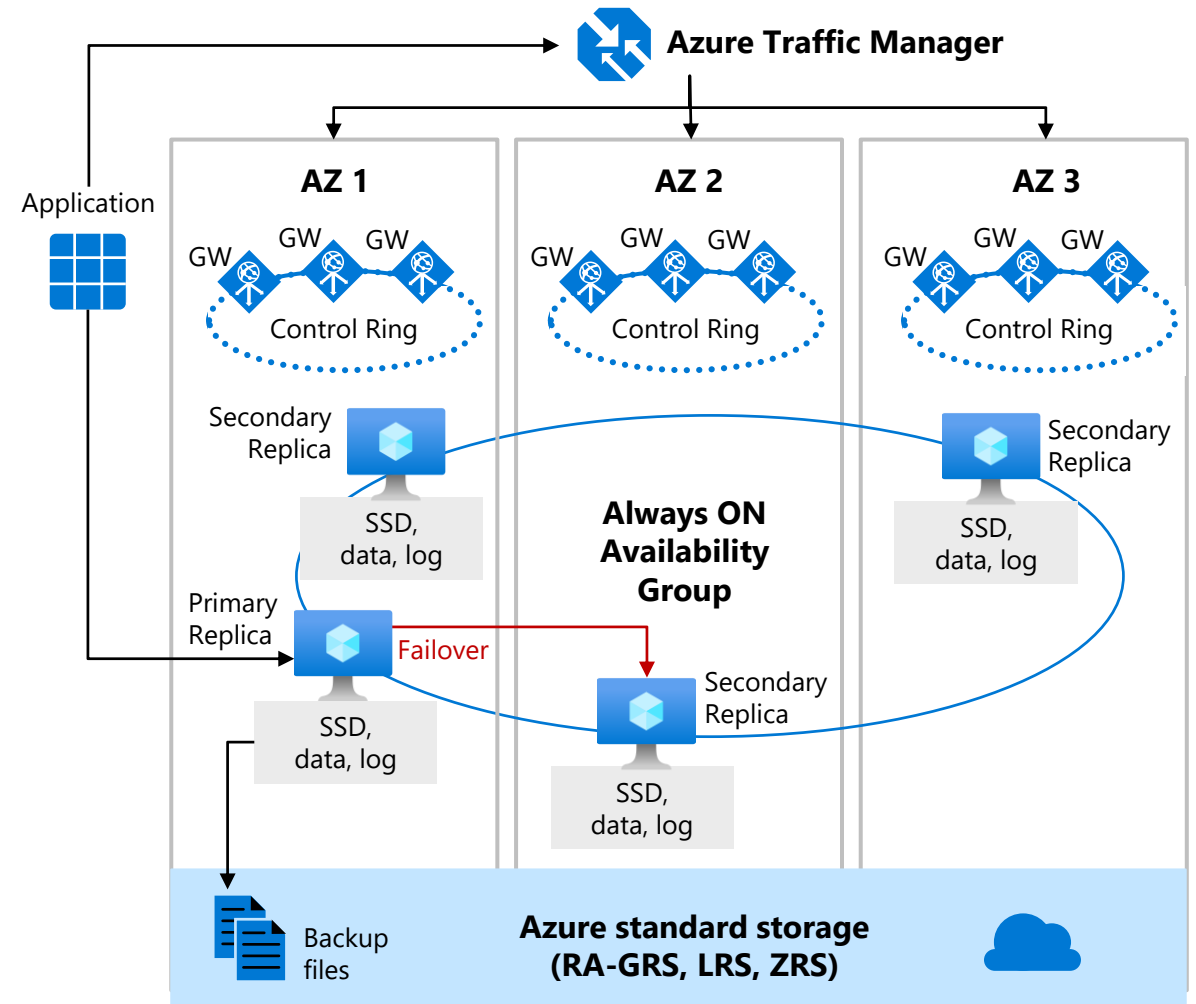
By default, the cluster of nodes for the premium availability model is created in the same datacenter.

SQL Database can place different replicas of the Business-Critical database to different availability zones in the same region.

The routing is controlled by Azure Traffic Manager (ATM).

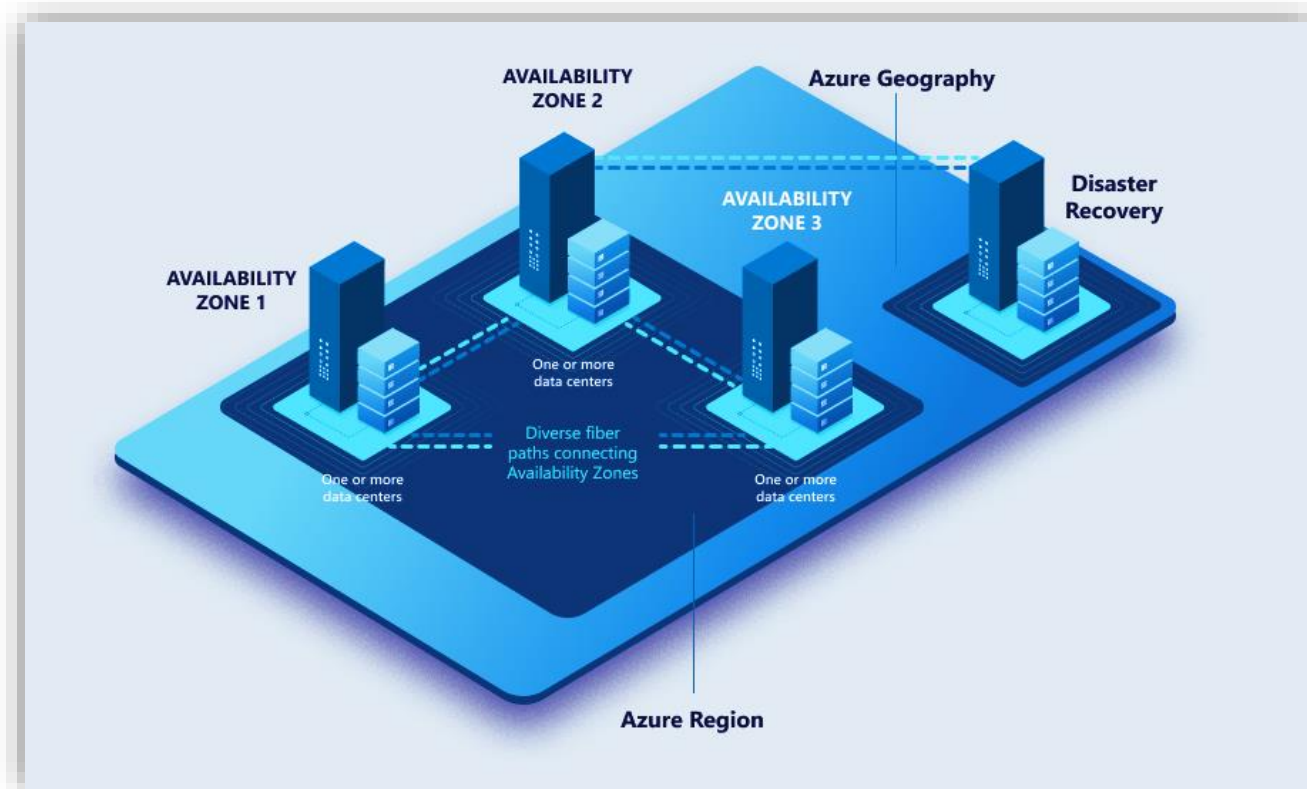
The zone redundant databases have replicas in different datacenters with some distance between them, the increased network latency may impact the performance.

Zone redundant configurations are currently only supported in the Premium or Business Critical tiers



# Backup storage redundancy

To enable high durability of backups several ways of replication are offered on instance creation.



The backups can be all located within

1. LRS: The same building (Local)
2. ZRS: Same region, different buildings (Zone)
3. GRS: Across paired regions (Geo)
4. GZRS: Different buildings AND paired regions (Geo-Zone)

# 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%            |

| Business continuity            | Service tier                  | SLA    |
|--------------------------------|-------------------------------|--------|
| Recovery point objective (RPO) | Business critical with Geo-DR | 5 sec  |
| Recovery Time Objective (RTO)  | Business critical with Geo-DR | 30 sec |

[SLA for Azure SQL Database](#)

[SLA for Azure SQL Managed Instance](#)

Questions?



## Lesson 2: Auto-Failover Groups

# Objectives

After completing this learning, you will be able to:


- Understand how Auto-Failover groups work and how to configure it.



# Auto-Failover Groups in Azure SQL MI



Allows the replication and failover of all databases in a managed instance to another region



Failovers can be initiated manually, or automatically based on a user-defined policy




Readable secondary databases can be used to offload read-only query workloads


# Auto-Failover Groups Limitations



Failover groups cannot be created between two servers or instances in the same Azure regions



Failover groups cannot be renamed, the group must be deleted and re-created with a different name



Database rename is not supported for instances in failover group, the group must be temporarily deleted to rename a database



# Auto-Failover Groups Terminology and Capabilities

Failover group  
(FOG)

Initial Seeding

DNS Zone

Failover Group  
Read-Write  
Listener

Failover Group  
Read-Only  
Listener

Automatic  
Failover Policy

Read-Only  
Failover Policy

Planned  
Failover

Unplanned  
Failover

Manual  
Failover

Grace Period  
with Data Loss

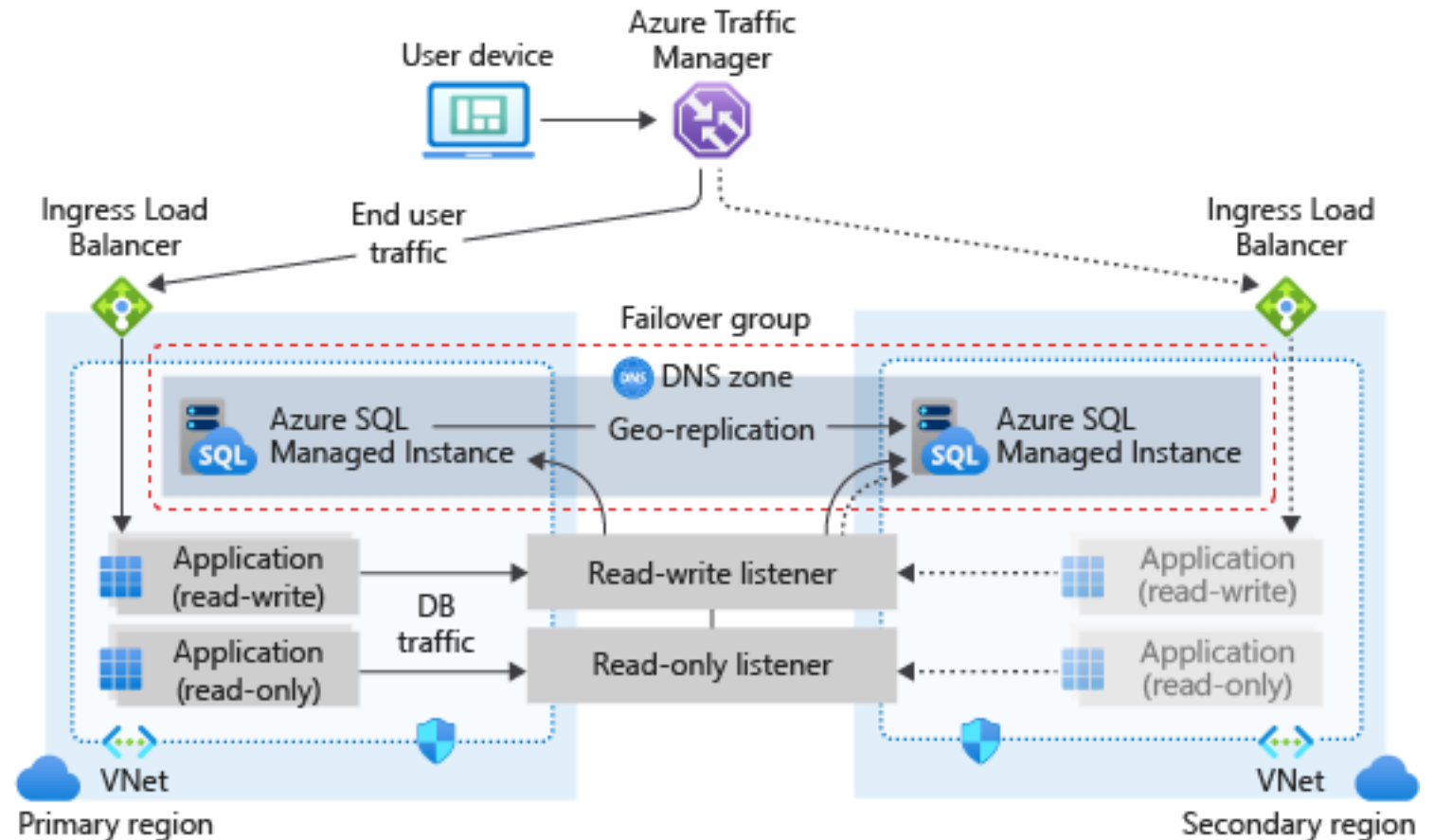
# Using Auto-Failover Groups in Azure SQL MI

## Capabilities

- Active / Standby
- All databases in the instance are automatically replicated
- Automatic or manual failover
- Read-write listener for read-write database connections
- Read-only listener for read-intended database connections

## Scenarios

- Transparent recovery from outage
- Load-balancing read-only workloads
- Failback after outage is mitigated



Questions?

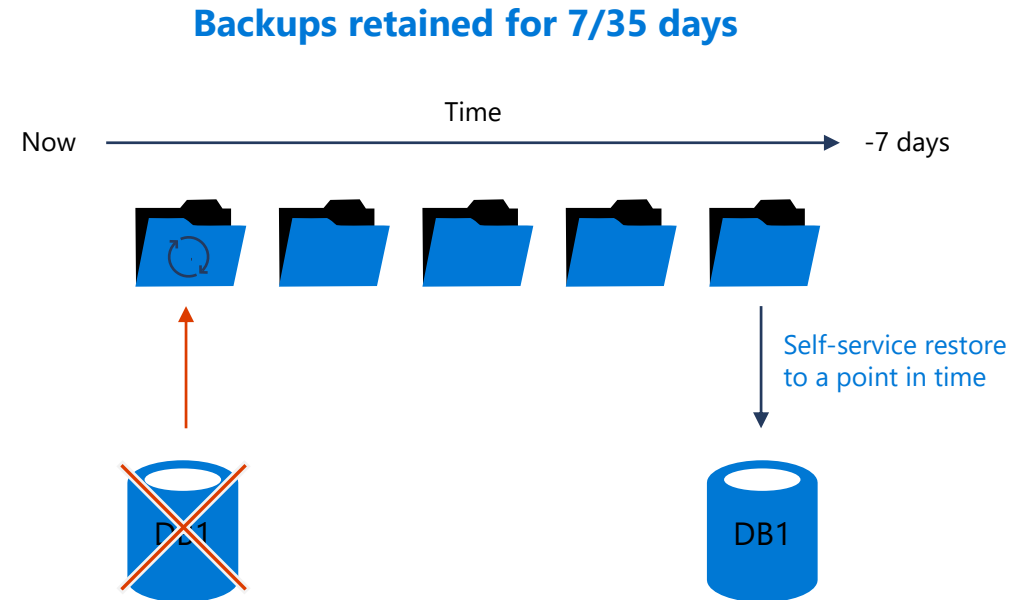


# Lesson 3: Automated Backups and Retention


# Backup and restore

## Auto backups and Point in Time Restore (PITR)

- Full Database backup once a week
- Log Backups every 5-10 minutes
- Differential Backups every 12-24 hours
- Backup files on Azure storage with RA-GRS replicated
  - Can optionally select LRS or ZRS
- Backup Integrity checks
- Restore to new database
- Long-term retention (up to 10 years) of backups
- Geo-restore of databases if primary region down
- Restore backups of deleted databases



# Automatic Backups

- 
- Uses SQL Server technology to create full, differential, and transaction log backups.
  - Transaction log backups, with full and differential backups, allow you to restore a database to a specific point-in-time to the same server that hosts the database.
  - When you restore a database, the service figures out which full, differential, and transaction log backups need to be restored.

# Automated Backups Usage



Point-in-time restore (PITR) of existing database



Point-in-time restore (PITR) of deleted database



Restore a database to another geographic region (Geo-Restore)



Restore from long-term backup

# Setting Backup Policies

Home > Resource groups > jdSQLRG > jdsqldb

**jdsqldb** | Backups ☆ ...

SQL server

Search

Data management

- Backups
- Deleted databases
- Failover groups
- Import/Export history

Security

- Networking
- Microsoft Defender for Cloud
- Transparent Data Encryption
- Identity
- Auditing

Intelligent Performance

Available backups

Retention policies

Configure and manage your automated backup retention policies. Long-term retention policies enable you to keep full backups for up to 10 years.

Search for a database

Databases in the Basic tier are limited to a 7 day retention policy.

| Database | PITR   | Differential backup frequency |
|----------|--------|-------------------------------|
| jdsqldb  | 7 Days | 24 Hours                      |

### Configure policies

SQL server

#### Point-in-time-restore

Specify how long you want to keep your point-in-time backups. [Learn more](#)

How many days would you like PITR backups to be kept? 7

#### Differential backup frequency

Specify how often you want differential backups to be taken. [Learn more](#)

Take a differential backup every:

24 Hours

#### Long-term retention

Specify how long you want to keep your long-term retention backups. You may choose to keep yearly backups for up to 10 years. [Learn more](#)

Weekly LTR Backups

Keep weekly backups for:

6 Week(s)



# Azure SQL Database Backup Retention Periods

All Azure SQL databases (single, pooled, and managed instance databases) have a default backup retention period of **seven** days.

You can change backup retention period up to 35 days.

If you delete a database, SQL Database will keep the backups in the same way it would for an online database.

If you need to keep the backups for longer than the maximum retention period, you can modify the backup properties to add one or more long-term retention periods to your database.

The point-in-time backups are geo-redundant and protected by Azure Storage cross-regional replication.  
[How long are backups kept.](#)

# How to change backup retention period

You can change the default PITR backup retention period using the Azure portal, PowerShell, or REST API.

The following examples illustrate how to change PITR retention to 28 days.

## PowerShell


```
Set-AzSqlDatabaseBackupShortTermRetentionPolicy -ResourceGroupName resourceGroup -ServerName testserver -  
DatabaseName testDatabase -RetentionDays 28
```

## REST


```
PUT https://management.azure.com/subscriptions/00000000-1111-2222-3333-  
444444444444/resourceGroups/resourceGroup/providers/Microsoft.Sql/servers/testserver/databases/testDatabase/back  
upShortTermRetentionPolicies/default?api-version=2017-10-01-preview
```

The supported values are: 7, 14, 21, 28 or 35 days.

# Backup Storage Consumption



PITR requires an uninterrupted backup chain: full backup, differential backup, and one or more transaction log backups



Additional full, differential, and transaction log backups must be kept for up to a week longer than the configured retention period



Backups that are no longer needed to provide PITR functionality are automatically deleted

Questions?



# Lesson 4: Long-Term Backup Retention


# Objectives

After completing this learning, you will be able to:

- Understand how long-term backup retention works for Azure SQL MI databases.
- Understand how to configure long-term backup retention for Azure SQL MI databases.




# How Long-Term Backups Work in Azure SQL MI



Long-term backup retention (LTR) leverages the full database backups that are created automatically to enable point-in-time restore (PITR)



If an LTR policy is configured, these backups are copied to different blobs for long-term storage



The copy is a background job that has no performance impact on the database workload



The LTR policy for each database specifies how frequently the backups are created

# Extending the Retention Period

You can configure a single or a pooled database with a long-term backup retention policy (LTR) to automatically retain the database backups in separate Azure Blob storage containers for up to 10 years.

You can then recover a database using these backups using the Azure portal or PowerShell.

Deleting LTR backup is non-reversible. To delete an LTR backup after the server has been deleted you must have Subscription scope permission.



# How SQL Database long-term retention works

Long-term backup retention leverages the automatic SQL Database backups created to enable point-time restore (PITR).

Specify for each SQL database how frequently you need to copy the backups to the long-term storage.

- Weekly backup retention (W)
- Monthly backup retention (M)
- Yearly backup retention (Y)
- Week of year (WeekOfYear)

$W=0, M=0, Y=5, \text{WeekOfYear}=3$

The 3rd full backup of each year will be kept for 5 years.

$W=0, M=3, Y=0$

The first full backup of each month will be kept for 3 months.

$W=12, M=0, Y=0$

Each weekly full backup will be kept for 12 weeks.

# Demonstration

Configure the long-term retention and view backups in long-term retention.



Questions?



# Lesson 5: Restoring Databases

# Objectives

After completing this learning, you will be able to:

- Understand the options available to restore automated backups.
- Understand how to use automated backups to restore SQL MI databases in several scenarios.



# Database recovery using automated backups




Create a new database on the same MI, recovered to a specified point in time within the retention period



Create a database on the MI, recovered to the deletion time for a deleted database



Create a new database on any MI in the same region, recovered to the point of the most recent backups



Create a new database on any MI, in any other region, recovered to the point of the most recent replicated backups

# Restore Considerations

Point-in-time restore can restore a database:

- From an existing database
- From a deleted database
- To the same or another managed instance

Restoring automated backups from within SSMS is not allowed

COPY\_ONLY, URL-based full backups can be restored using SSMS to SQL MI only

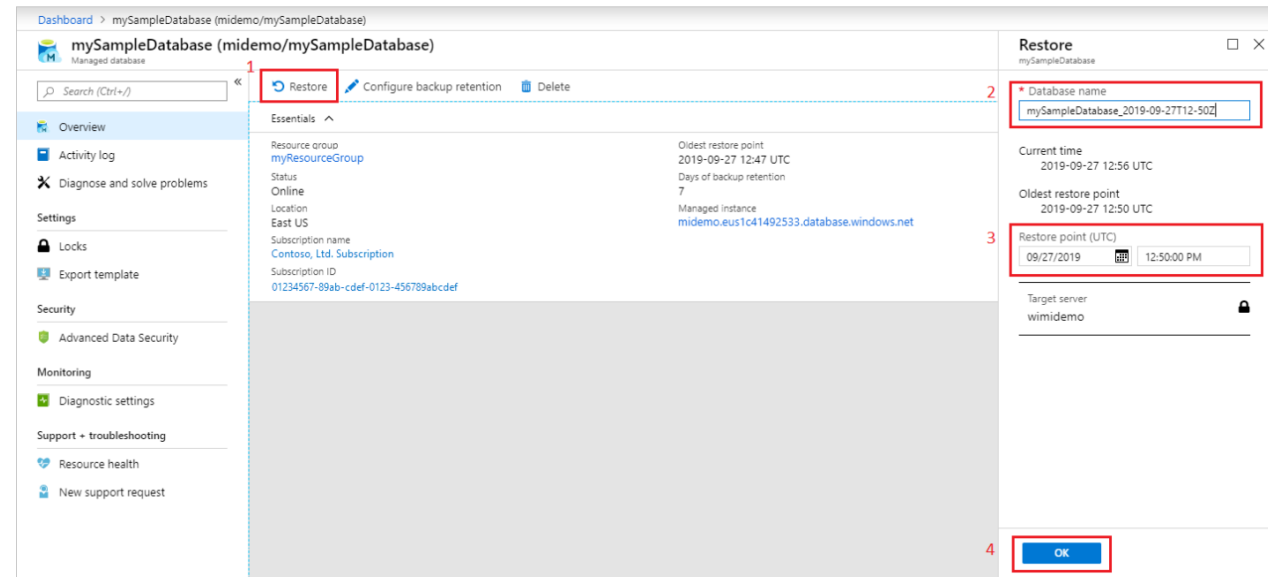


Limitations:

- Cross-region and cross-subscription restore aren't currently supported
- Point-in-time restore of a whole managed instance is not possible

# Point-in-time Restore by Using Azure Portal

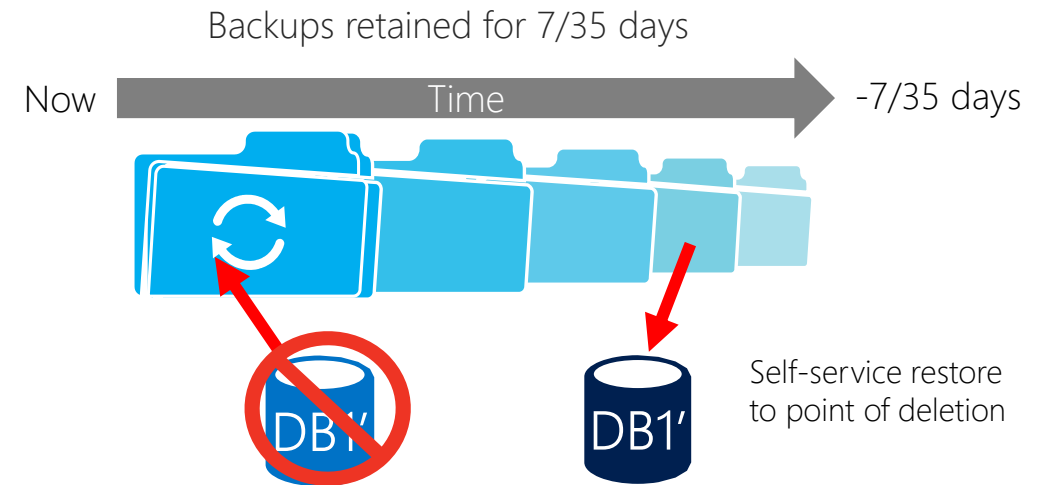
- Open the Managed Instance overview page
- Select **Restore** on the toolbar
- Choose the point-in-time backup point from which a new database will be created





# Deleted Database Restore by Using the Azure Portal

- Open the Managed Instance overview page
- Select **Deleted databases**
- Select a deleted database that you want to restore
- Type the name for the new database that will be created with data restored from the backup



Dashboard > midemo - Deleted databases

midemo - Deleted databases  
SQL managed instance

Deleted databases are available for restore within the retention period configured. To restore a deleted database select it from the list and follow the restore prompts.

| Database         | Creation time (UTC) | Earliest restore time (UTC) | Deletion time (UTC) |
|------------------|---------------------|-----------------------------|---------------------|
| mySampleDatabase | 2019-11-14 13:00    | 2019-11-14 13:05            | 2019-11-14 13:07    |

1 Deleted databases

2 Database

3 Database name \*  
mySampleDatabase\_2019-11-14T13-07Z

Restore point (UTC)  
11/14/2019 1:07:55 PM

4 OK

Restore  
mySampleDatabase

Deleted database is restored to a new database. To initiate the restore, type in a database name to which backed up database will be restored, and choose a restore point in time from the available backups.

Target server  
midemo

# Deleted Database Restore by Using PowerShell

## Same Managed Instance

```
$subscriptionId = "<Subscription ID>"
Get-AzSubscription -SubscriptionId $subscriptionId
Select-AzSubscription -SubscriptionId $subscriptionId

$resourceGroupName = "<Resource group name>"
$managedInstanceName = "<Managed instance name>"
$deletedDatabaseName = "<Source database name>"
$targetDatabaseName = "<target database name>"

$deletedDatabase = Get-AzSqlDeletedInstanceDatabaseBackup -
ResourceGroupName $resourceGroupName -InstanceName
$managedInstanceName -DatabaseName $deletedDatabaseName

Restore-AzSqlInstanceDatabase -Name $deletedDatabase.Name -
InstanceName $deletedDatabase.ManagedInstanceName -
ResourceGroupName $deletedDatabase.ResourceGroupName -DeletionDate
$deletedDatabase.DeletionDate -PointInTime UTCDateTime -
TargetInstanceDatabaseName $targetDatabaseName
```

## Another Managed Instance \*

```
$targetResourceGroupName = "<Resource group of target
managed instance>"
$targetInstanceName = "<Target managed instance name>"

Restore-AzSqlInstanceDatabase -Name
$deletedDatabase.Name -InstanceName
$deletedDatabase.ManagedInstanceName -
ResourceGroupName
$deletedDatabase.ResourceGroupName -DeletionDate
$deletedDatabase.DeletionDate -PointInTime
UTCDateTime -TargetInstanceDatabaseName
$targetDatabaseName -TargetResourceGroupName
$targetResourceGroupName -TargetInstanceName
$targetInstanceName
```

\* Within same subscription

# Geo-Restore

Restores last daily backup to any Azure region

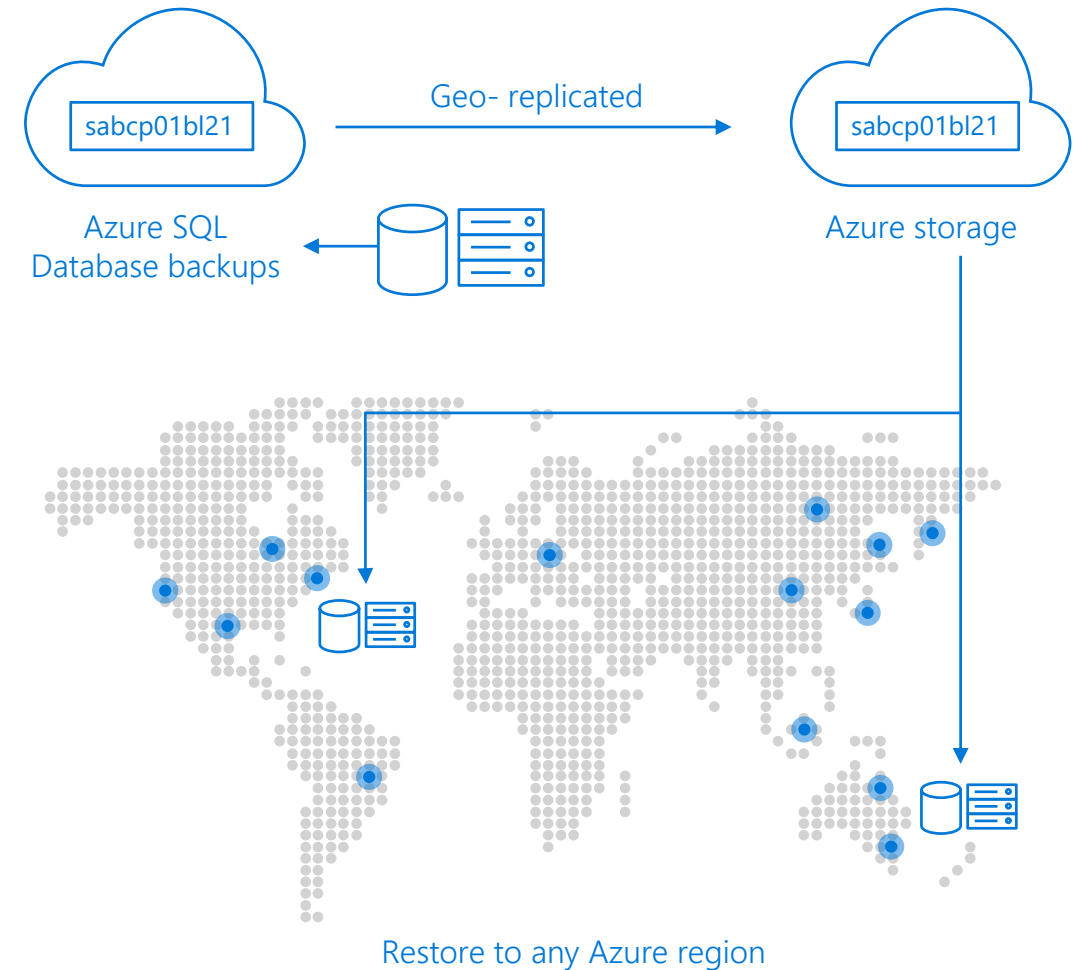
Built on geo-redundant Azure Storage

No extra cost, no capacity guarantee

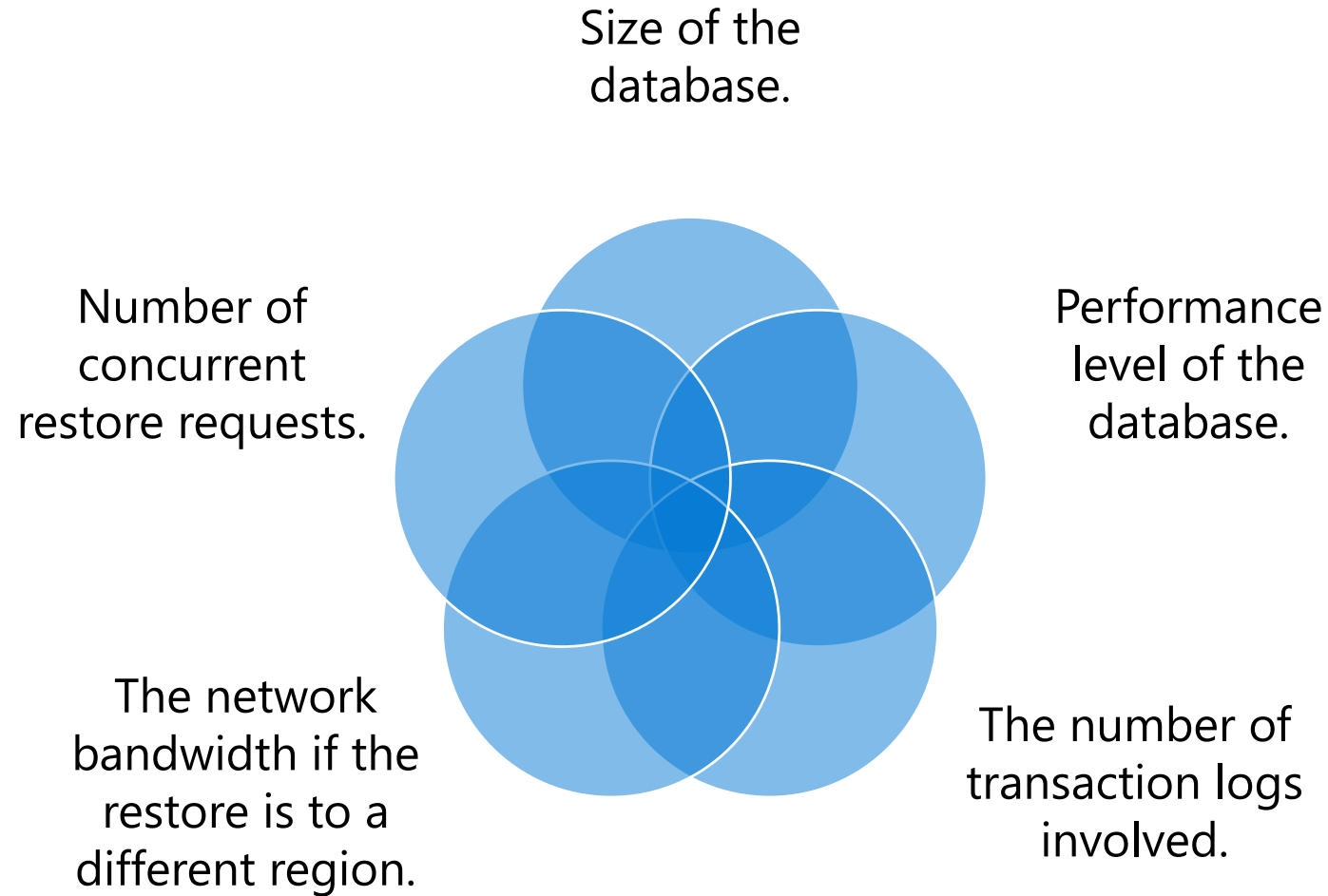
$RTO \leq 12h$ ,  $RPO \leq 1h$

Database URL will change after restore

Point-in-time restore on a geo-secondary is not currently supported



# Factors Affecting Recovery Time



Questions?



# Lesson 6: Accelerated Database Recovery

# Objectives

After completing this learning, you will be able to:

- Understand how Accelerated Database Recovery (ADR) works and its benefits.
- Understand the types of workloads that benefit the most from Accelerated Database Recovery (ADR).



# Accelerated Database Recovery (ADR)

A new feature that greatly improves database availability, especially in the presence of long running transactions, by redesigning the database engine recovery process

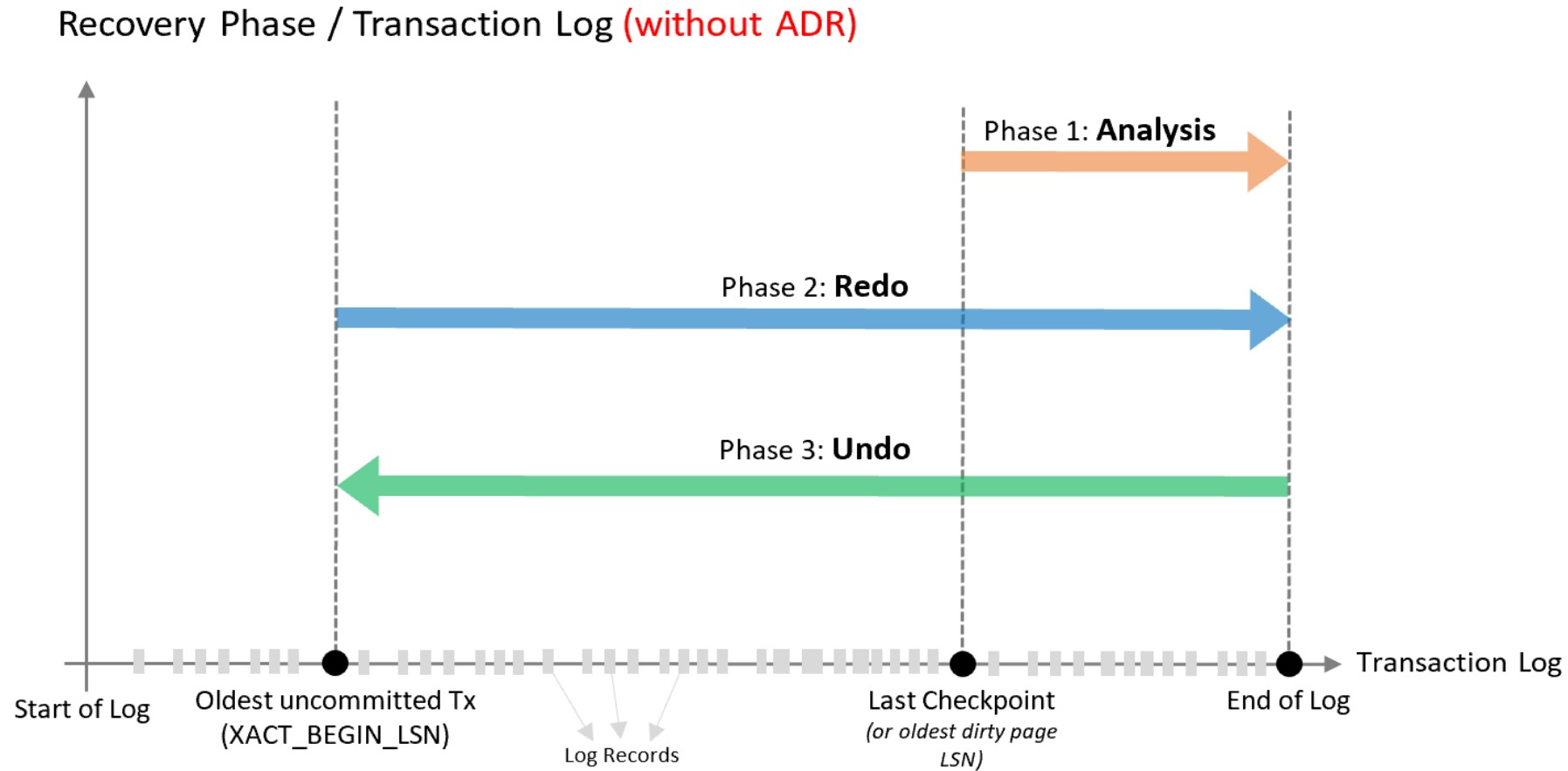
ADR is enabled by default in Azure SQL Database and Azure SQL Managed Instance and disabling ADR for either product is not supported

Primary benefits of ADR are:

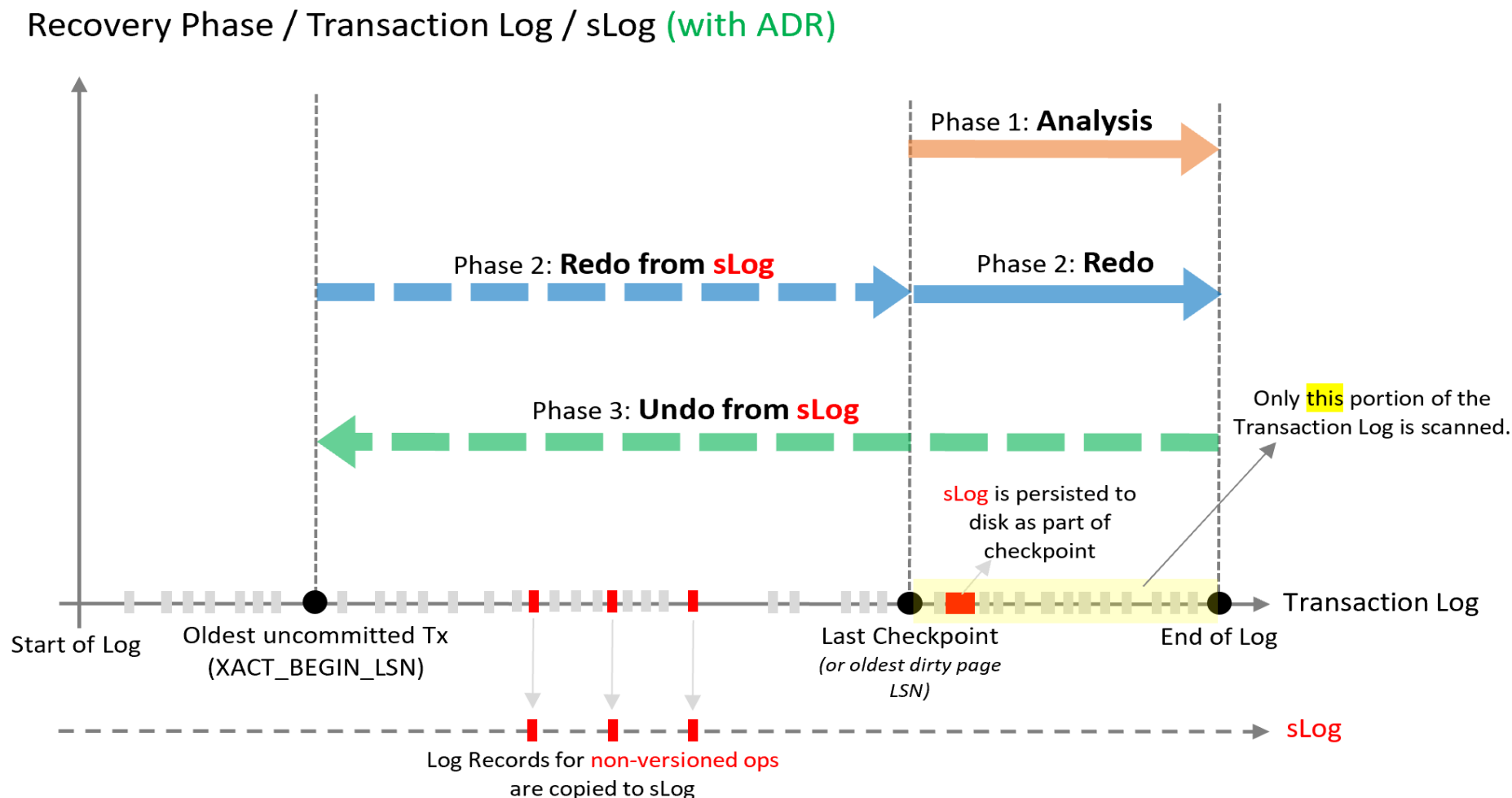
- Fast and consistent database recovery
- Instantaneous transaction rollback
- Aggressive log truncation



# Accelerated Database Recovery – without ADR



# Accelerated Database Recovery – with ADR



# Accelerated Database Recovery (ADR) Patterns

The following types of workloads benefit most from ADR:

- With long-running transactions
- That have seen cases where active transactions are causing the transaction log to grow significantly
- That have experienced long periods of database unavailability due to long running recovery, caused by unexpected service restart or manual transaction rollback

# Demonstration

## Accelerated Database Recovery

- Verify ADR behavior for databases on Azure SQL MI



Questions?



# Module Summary

---

Automated Backups and Retention

---

Long-Term Backup Retention

---

Database Restores

---

Accelerated Database Recovery

---

Auto-Failover Groups

