Microsoft

SQL Server High Availability

Module 6

## Learning Units covered in this Module

- Lesson 1: Introduction to Failover Clustering

- Lesson 2: Introduction to Availability Groups

- Lesson 3: What's new with Availability Groups

- Lesson 4: Intro to Managed Instance Link

- Lesson 5: Availability Group Best Practices

Lesson 1: Introduction to Failover Clustering

Objectives

After completing this learning, you will be able to:

- Understand the offerings for standalone and clustered instances of Microsoft SQL Server.

- Showcase and engage in architectural discussions regarding the appropriate usage of standalone and clustered instances.
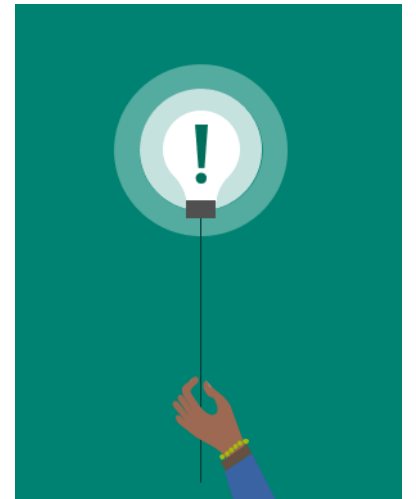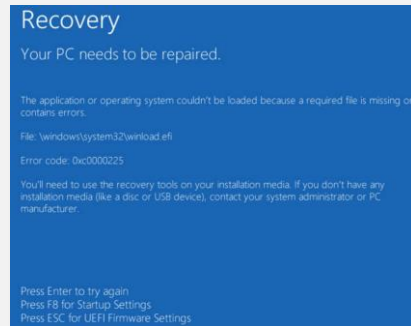
Business Continuity Problem

Mechanisms, Policies and Procedures that enable a business to continue operating in face of disruption.

Customer Error

Customer deleted their data by mistake.

Equipment Failures

Hardware or other equipment failures

Natural Disasters

Flood, Hurricanes, Tornadoes, Earthquakes

Manmade Disasters

Wars or major ecological accidents

Different configuration environments

| Standalone Physical Server	Standalone Virtual Server	Failover Cluster Instances

• Simple to configure
• Straightforward monitoring and management
• Can be taken offline by:
  • Hardware failures
  • Driver issues
  • Software updates
  • Human error (shutdown instead of log off)
  • Disaster
	• Can provide some HA elements
	• Multiple levels to consider for monitoring and management
	• Can be taken offline by:
	  • Driver issues
	  • Patching
	  • Human error (shutdown instead of log off)
	  • Disaster
		• Highly Available
		• Generally monitored like a standalone server
		• Multiple levels to consider for management
		• Require shared storage
		• Can be taken offline briefly by:
		  • Driver issues (blue screens)
		  • Software updates
		  • Human error (shutdown instead of log off)
		• Still vulnerable to disasters without data replication solution

Failover Clustering Overview
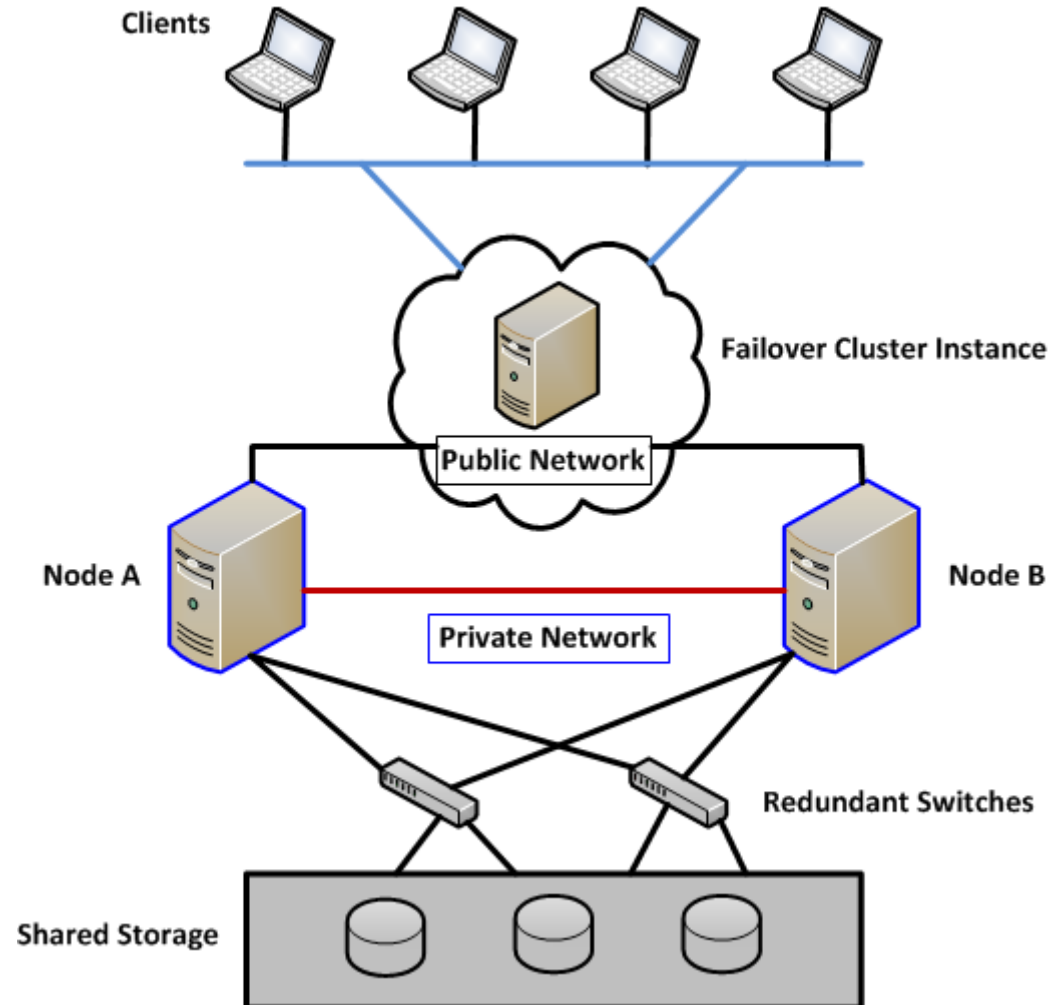
FCI Scenarios

Single Instance

Simple architecture
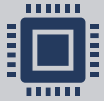
Easiest to schedule maintenance

Low utilization of resources

FCI Scenarios
Multiple Instances (N+x)

Most complex except for geographically dispersed clusters.

Higher hardware costs for greater flexibility and more redundancy.

Idle node(s) can be leveraged to reduce complexity and coordination needed for maintenance.

Upgrading or patching an instance can be planned to minimize the impact on other instances.

FCI Scenarios

Geographically Dispersed

High complexity

Requires third-party data (SAN) replication technology

Can be a single instance, N+0, or N+x

Most expensive but also provides disaster recovery capability

FCI Considerations

Quorum Components

Quorum

- Majority of voting nodes and witnesses in a Windows Server Failover Cluster.

FCI witness

- Can be a disk or file share that acts as an arbitrator if you have an even number of nodes.

Votes

- Each component (nodes and witness) can have a vote.

Verify

- Ensure you have an appropriate Quorum model for your needs
- Update as architecture and SLAs change.

FCI Considerations
Quorum Models

Quorum Model	Configuration Detail
Node Majority	Eligible for automatic selection when creating a cluster with an odd number of nodes, no witness disk configured.
Node and Disk Majority	Eligible for automatic selection when creating a cluster with an even number of nodes. A witness disk is configured if storage is available.
Node and File Share Majority	Manual configuration required and a connection to a Server Message Block (SMB) network share required. Copy of full cluster hive is not placed on the file share witness. The file share witness may be on premises or in Azure.
No Majority – Disk Only	Manual configuration required single point of failure and equates to the old concept of quorum.

FCI Considerations

Simple Rule

Multisite Failover Clusters

- Identify nodes that will be used to establish high availability and treat as a cluster with that many nodes, setting the NodeWeight to 0 for all remaining nodes.

Dynamic Witness

FCI Considerations

MSDTC

Local to SQL

- MSDTC in a local SQL Server resource group

Mapped

- MSDTC mapped to an instance
  - msdtc -tmMappingSet

Cluster Default

- Cluster's default MSDTC
  - First MSDTC resource installed in a Failover Cluster

Local

- MSDTC local to the server

Knowledge Check

Describe a scenario that would be a good fit for:

A standalone physical server

A standalone virtual server

An AlwaysOn failover cluster instance

One or more AlwaysOn availability groups

Lesson 2: Introduction to Availability Groups

Objectives

After completing this learning, you will be able to:

· Describe the components of an availability group.

· Describe how to use an availability group.

Availability Groups
Overview

Always On Availability Groups feature is a high-availability and disaster-recovery solution.

An availability group supports a failover environment for a discrete set of user databases.

An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only.

From SQL 2016, database issues such as a database becoming suspect due to a loss of a data file, deletion of a database, or corruption of a transaction log can trigger failovers, if Database Health detection is configured for the Availability Group.

Availability Groups

Overview

High Availability for groups of databases

- Databases fail over together from one SQL Server to another.
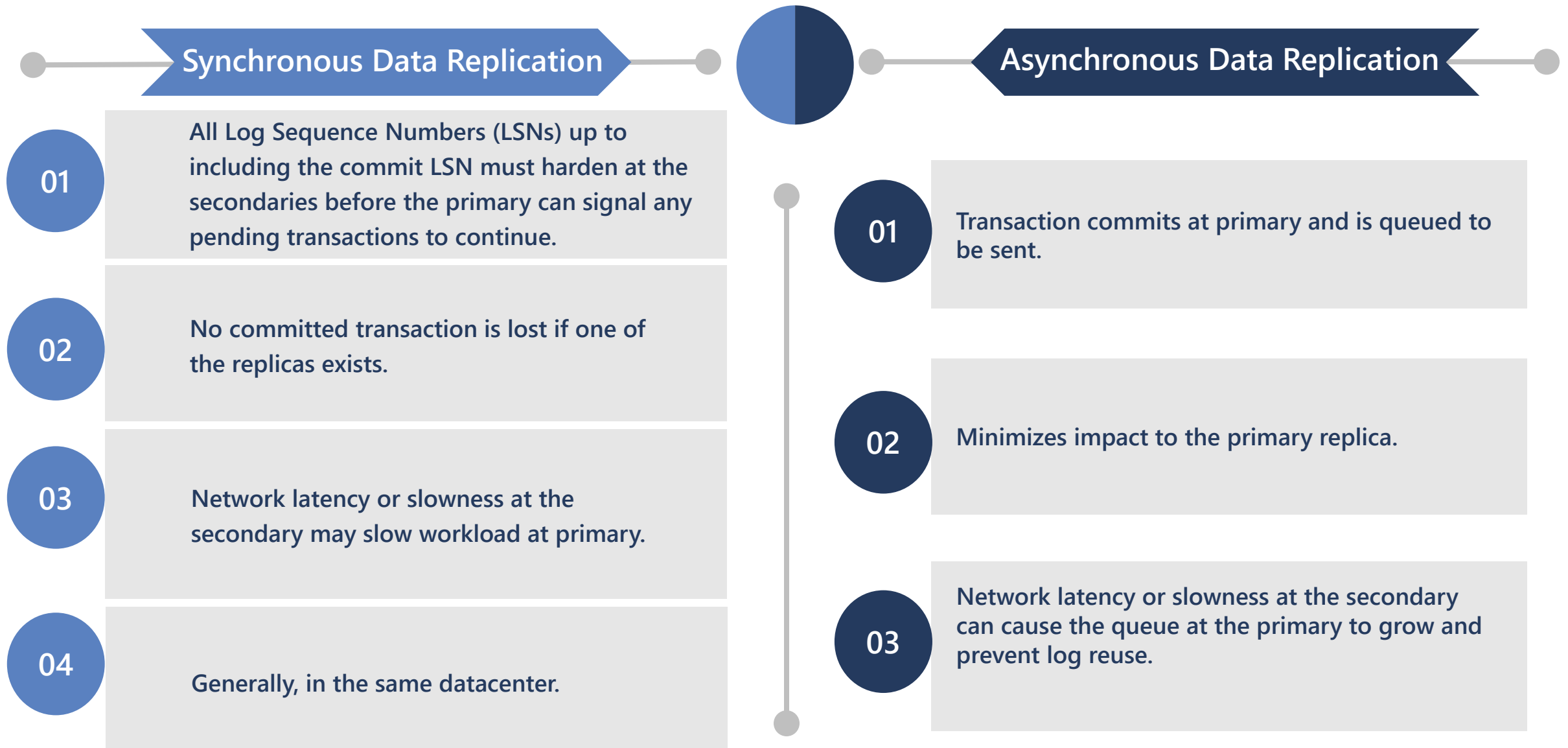
Failovers can be automatic or manual

- Automatic: SQL Server failures (software/hardware/network)
- Manual: For service packs or upgrade

SQL Server replicates transactions from primary to secondaries

- Physical replication (log blocks are delivered, stored, and applied).
- Does not need shared storage.

Synchronous versus Asynchronous

Synchronous Data Replication

01 All Log Sequence Numbers (LSNs) up to including the commit LSN must harden at the secondaries before the primary can signal any pending transactions to continue.

02 No committed transaction is lost if one of the replicas exists.

03 Network latency or slowness at the secondary may slow workload at primary.

04 Generally, in the same datacenter.

Asynchronous Data Replication

01 Transaction commits at primary and is queued to be sent.

02 Minimizes impact to the primary replica.

03 Network latency or slowness at the secondary can cause the queue at the primary to grow and prevent log reuse.
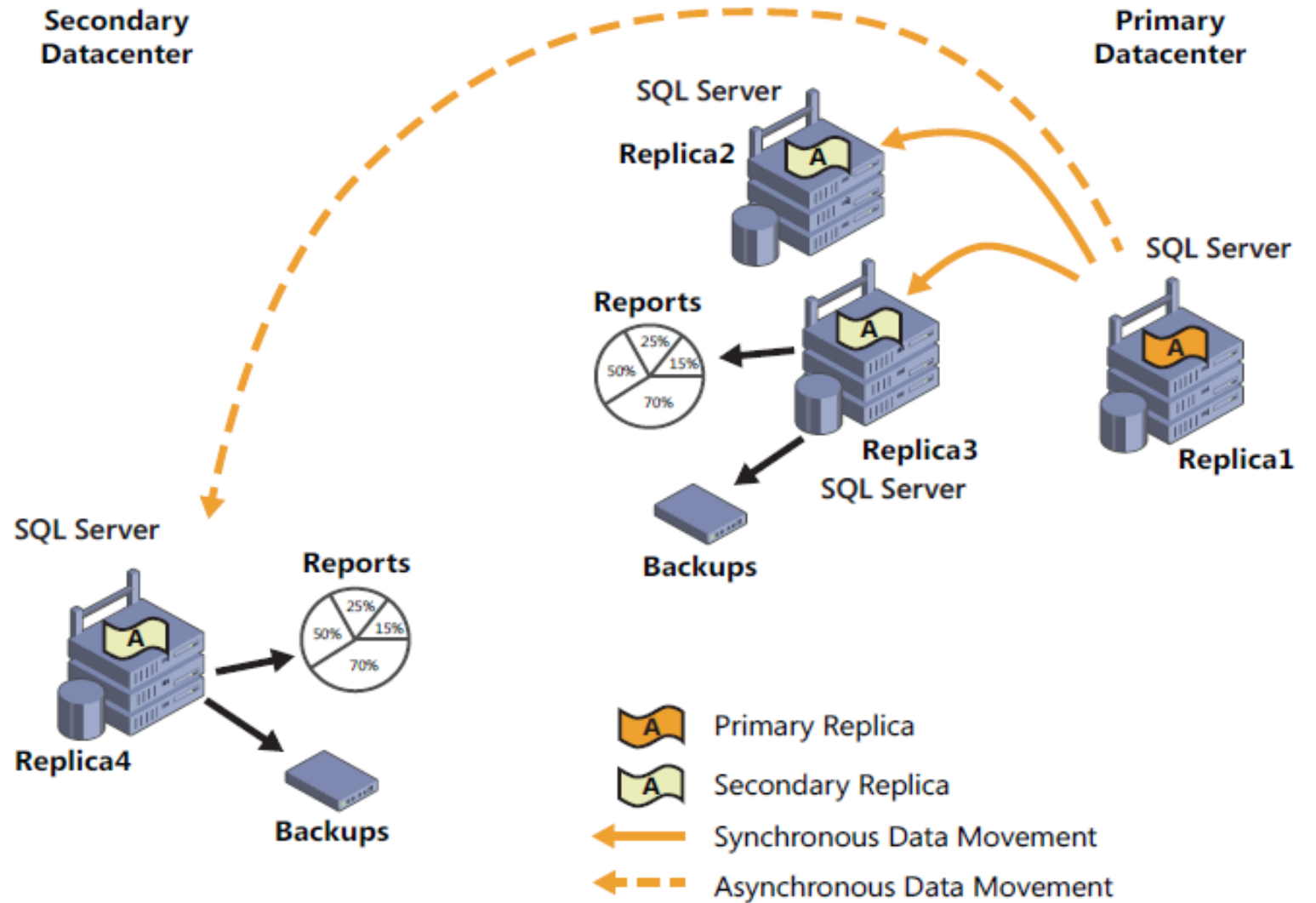
AlwaysOn Availability Group

Sample Architecture

Synchronous data movement is used to provide high availability within the primary datacenter.

Asynchronous data movement is used to provide disaster recovery.

Secondary replica 3 and replica 4 are employed to offload reports and backups from the primary replica.

Secondary Datacenter

Primary Datacenter

SQL Server

Replica2

SQL Server

Reports

25%
50% 15%
70%

Replica3
SQL Server

Replica1

Backups

SQL Server

Reports

25%
50% 15%
70%

Replica4

Backups

Primary Replica

Secondary Replica

Synchronous Data Movement

Asynchronous Data Movement

Read-only Scale Out

Up to 8 readable replicas

Read-scale out (7x faster than Replication)

Supported for traditional tables, durable In-Memory tables, and ColumnStore Indexes

Users may connect by using the listener by specifying an ApplicationIntent of ReadOnly and by specifying a database name in the connection string

ReadOnly routing specifies a failover order sending all connections to the first available instance in the read-only routing list

Load balancing can be achieved through round robin in the listener, or through an appliance

Questions?

Knowledge Check

Is automatic failover available to asynchronous replica?

Can different replicas in the same Availability Group be synchronous and asynchronous?

What risk is involved with a forced failover to an asynchronous replica?

What are some reasons to chose asynchronous over synchronous?

Lesson 3: What's new with Availablity Groups

Objectives

After completing this learning, you will be able to:

· Learn what recent improvements have been made in availability groups.

Improvements in Availability Groups

High Availability	Scalability	Flexibility	Consistency	Ease of use
• Five automatic failover replicas • Database-level failure detection • Support for group managed service accounts (gMSAs)	• Much higher synchronization throughput	• Replicas in different domains or no domain • Distributed availability groups • Basic availability groups in SQL Server Standard Edition	• Support for distributed transactions	• Load balancing of read workloads • Read/Write Connection Redirection • Contained availability groups

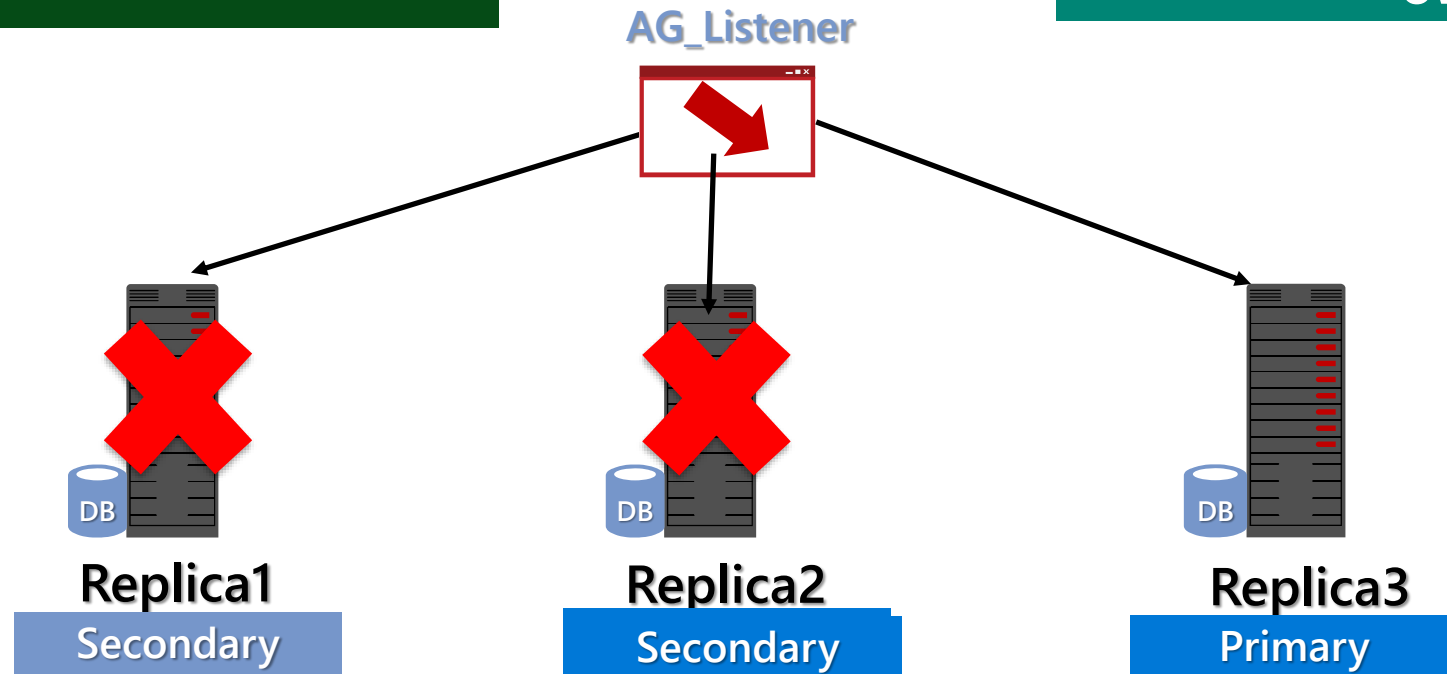# Five Automatic Failover Replicas

Increased from two or three replicas in prior versions

Maintain availability despite four failed replicas

Listener connects to primary as before

Failover priority defined in the windows cluster's preferred owners

AG_Listener

Replica1
Secondary

Replica2
Secondary

Replica3
Primary

Database Failure Detection

Added in SQL Server 2016

Automatic failover if any user database in the Availability Group goes offline unexpectedly (For example, due to inaccessible database files)

This is an option on top of the Availability Group Health Level (evaluates server health using sp_server_diagnostics)

```
CREATE AVAILABILITY GROUP ag_name WITH
(
    FAILURE_CONDITION_LEVEL = { 1 | 2 | 3 | 4 | 5 },
 DB_FAILOVER = { ON | OFF }
)
```

MDF

LDF

DB
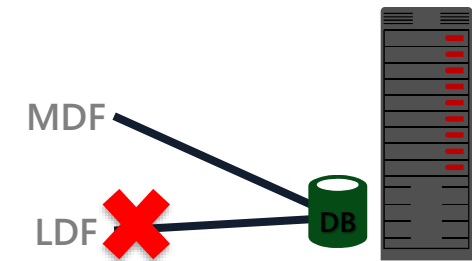
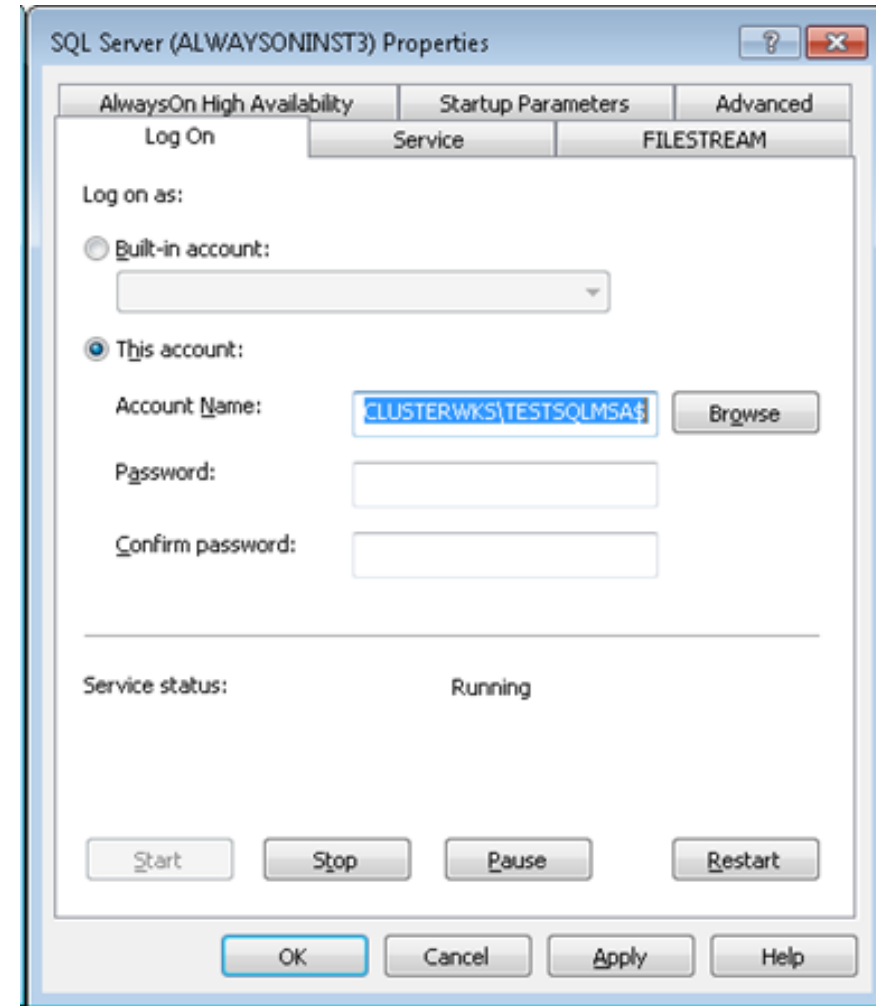# Group Managed Service Accounts

Support for gMSAs

Active Directory Services handles password changes for service accounts.

- No need to restart SQL Server services
- 120-character passwords

More secure than regular domain accounts.

Enables cross-system security context.

**Service Principal Name (SPN) registration is done automatically.**

DTC Support

Support DTC (Distributed Transaction Coordinator)

- Guarantees transactional (ACID) properties of distributed transactions for databases that are a part of an availability group.
- AG registers databases in DTC with a globally unique identifier (GUID).
- Requires Windows Server 2012 R2 or later.

```
CREATE AVAILABILITY GROUP ag_name WITH
(
  DTC_SUPPORT = { PER_DB | NONE }
)
```


Replica1
(Primary)

DTC

Replica2
(Secondary)

Client
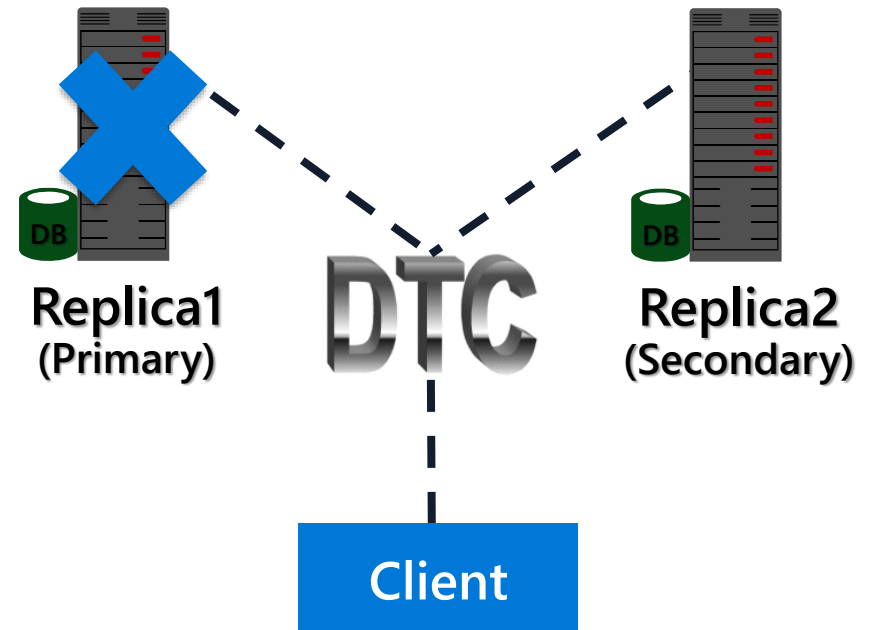
Synchronization Throughput

	Throughput MB/s	Average CPU utilization (secondary)	MB sent on wire/sec
<=2014	82	17	35
2016	540	36	230
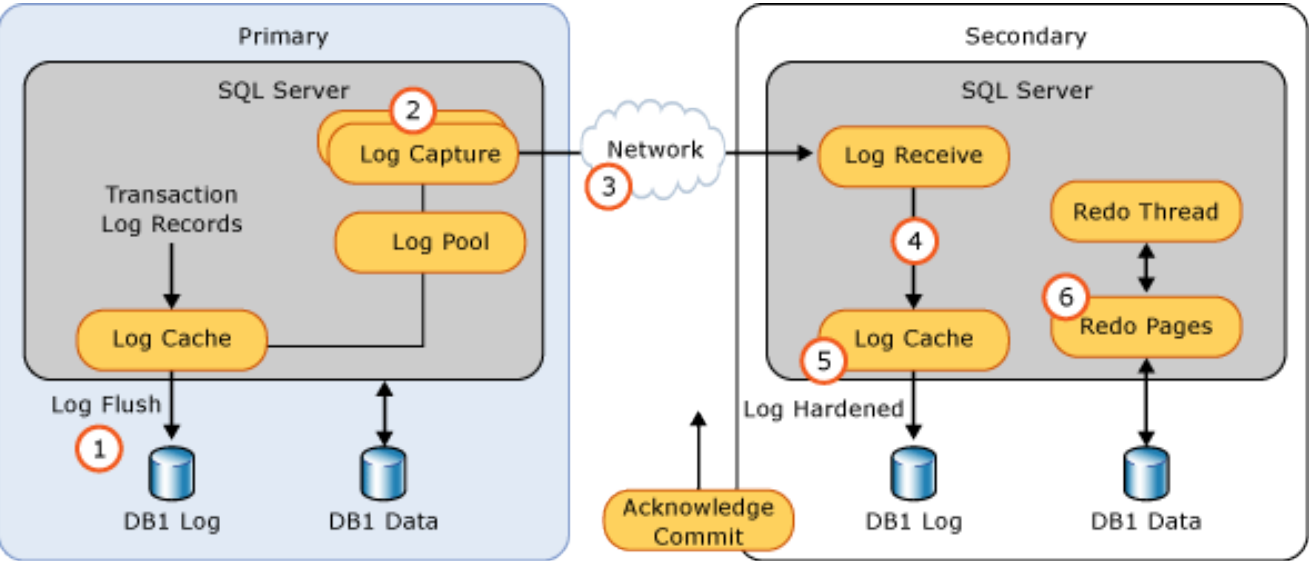
Reduced context switches (5x)

Multi-threaded log redo in log replay on secondaries

New compression function: LZ4

Multi-threaded log compression/decompression

Hardware compression

Replicas in Different Domains or Domain-less

SQL Server management does not change	Windows Cluster nodes are configured with certificate-based authentication such as database mirroring	Requires Windows Server 2016

AG_Listener

Domain A

Replica1
(Primary)

Replica2
(Secondary)

Domain B

Replica3
(Secondary)
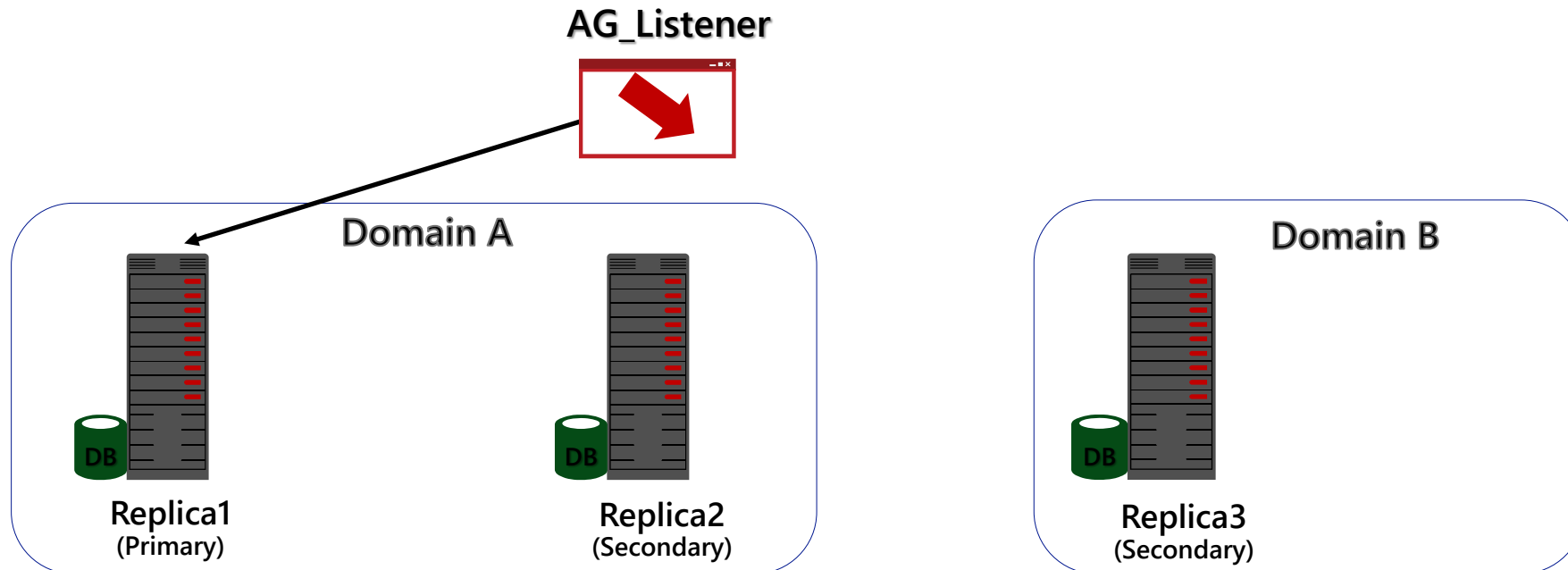
Distributed Availability Groups

One Availability Group can synchronize to one or more Availability Groups	Availability Groups are in different Windows Clusters	Unidirectional synchronization (only one master replica), uses different listeners
• Reduces primary replica's network and CPU usage	• Flexible management of Windows Clusters (such as an independent quorum)	

```
CREATE AVAILABILITY GROUP dist_ag
WITH (DISTRIBUTED) AVAILABILITY GROUP ON
'ag1' WITH (
    LISTENER_URL = 'tcp://ag1-listener:5022',
    AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
    FAILOVER_MODE = AUTOMATIC),
 'ag2' WITH (
    LISTENER_URL = 'tcp://ag2-listener:5022',
    AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT,
    FAILOVER_MODE = AUTOMATIC)
```
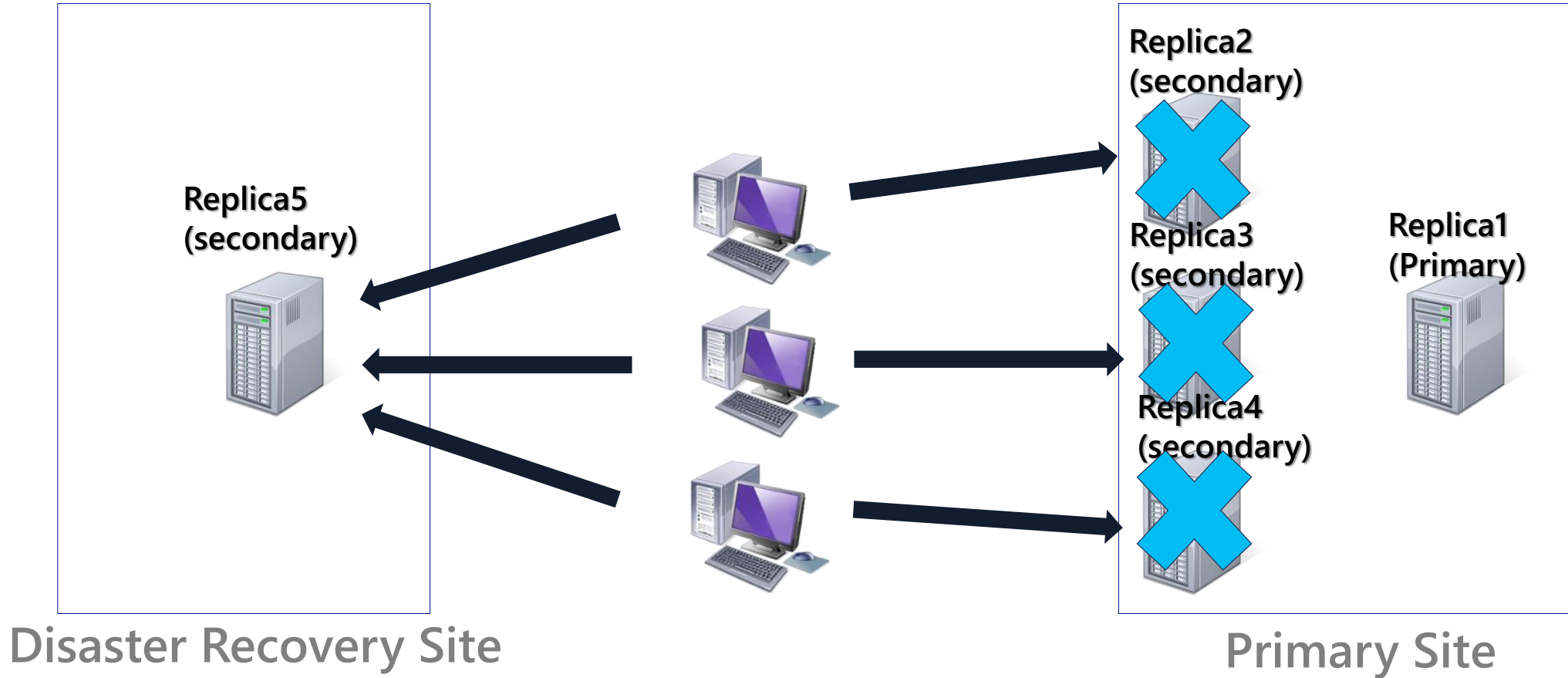
Load Balancing of Read Workloads

Read connections (specifying application_intent='read_only') are load balanced between readable secondaries

Integrated load balancing (instead of Domain Name System (DNS)-based or appliance-based)

Possible to specify prioritized load balancing groups (such as local first)

Simple load balancing: round robin
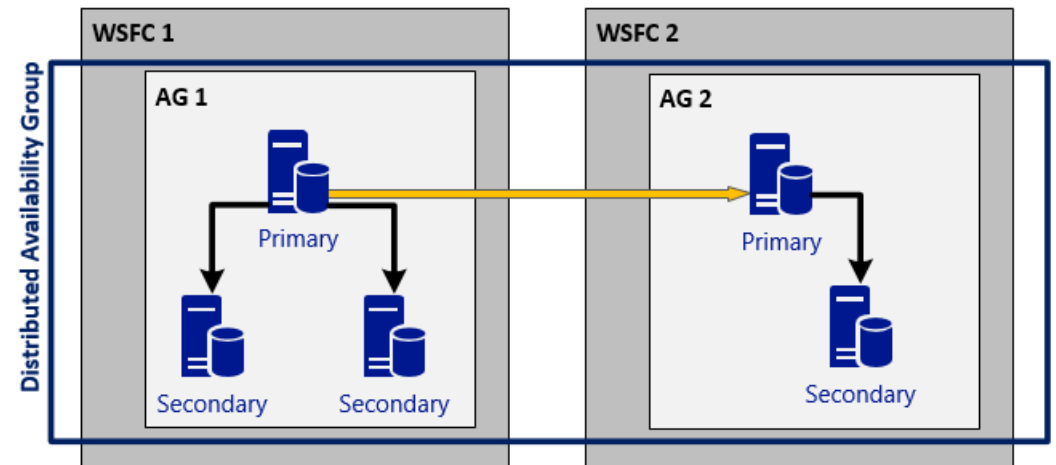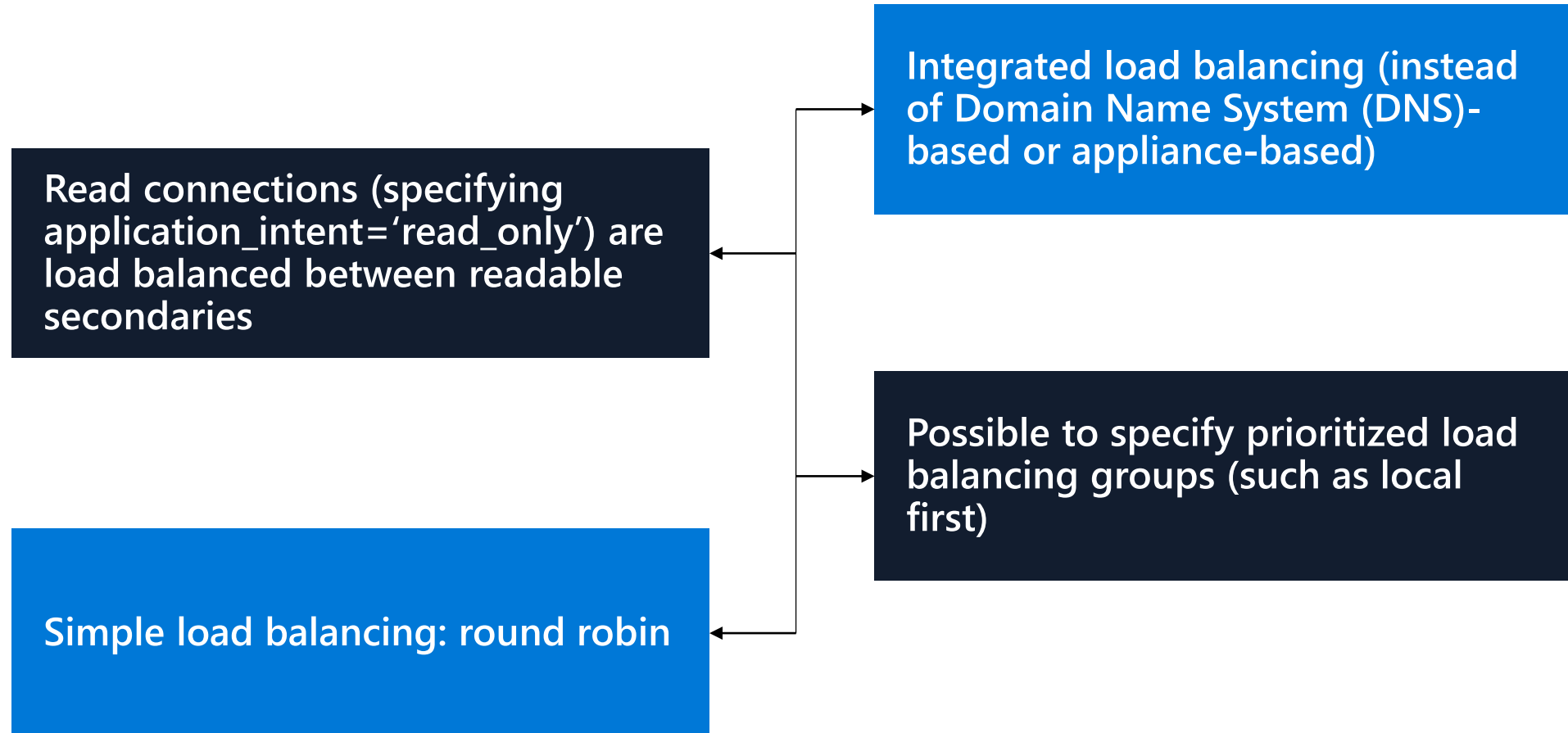
Load Balancing Read Routing List

READ_ONLY_ROUTING_LIST= (('Replica2','Replica3','Replica4'), 'Replica5')

Replica2
(secondary)

Replica3
(secondary)

Replica1
(Primary)

Replica4
(secondary)

Replica5
(secondary)

Disaster Recovery Site

Primary Site

Basic Availability Groups

Basic high availability solution in SQL Server 2016 Standard Edition (replaces Database Mirroring)
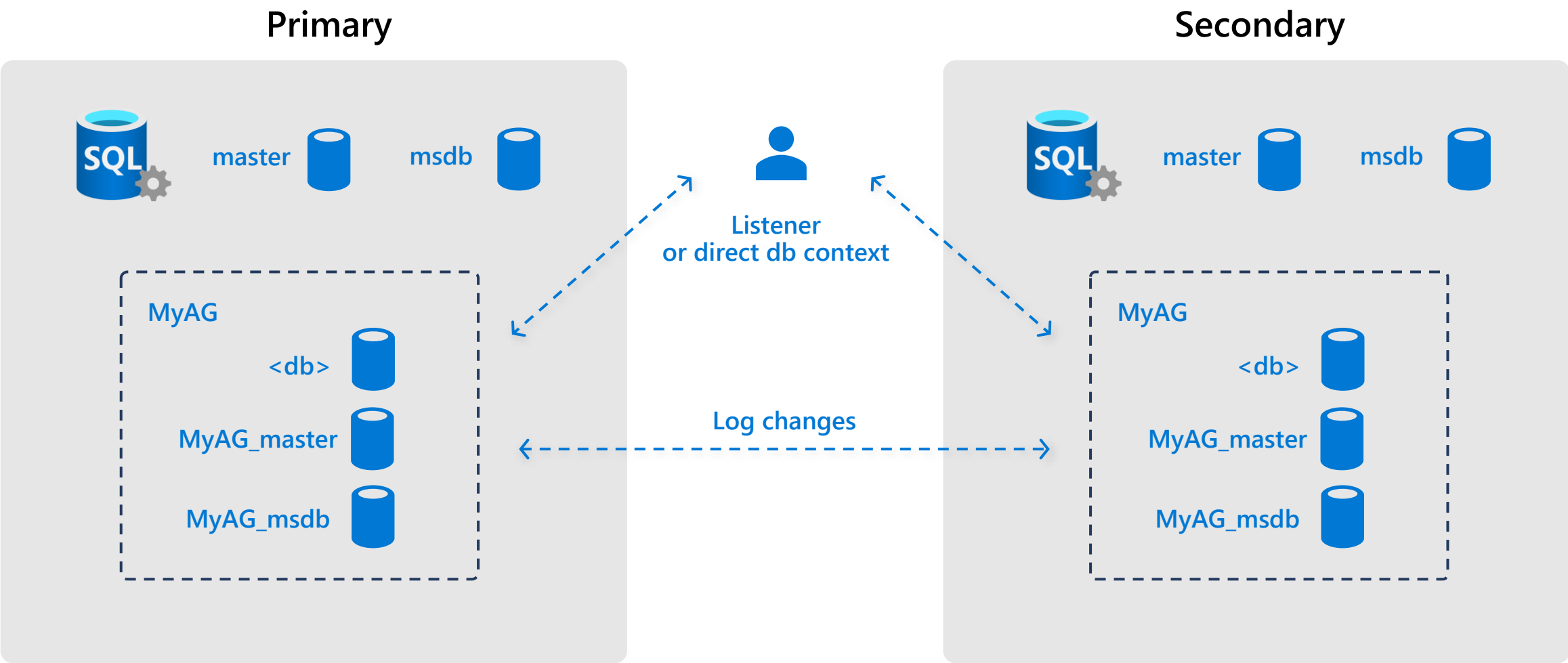
Limited to two replicas

- Sync or Async
- Can form a hybrid availability group (one on-premises and one in Azure Infrastructure as a service (IaaS))
- Not readable on the secondary
- Backup preference of the primary only

Automatic or manual failover

One database per availability group

Contained Availability Groups

Contained Availability Groups

Changed characteristics.

Has its own Master and msdb system databases

Separated from the actual SQL Server instance

Logins and agent jobs automatically replicated

Instance abstracted from the contained group

Must create a listener

Create a Contained Availability Group

SQL Server Management Studio

- New Availability Group Wizard
- New Availability Group

T-SQL


```
CREATE AVAILABILITY GROUP [AG01]
WITH (AUTOMATED_BACKUP_PREFERENCE = SECONDARY,
DB_FAILOVER = OFF,
DTC_SUPPORT = NONE,
REQUIRED_SYNCHRONIZED_SECONDARIES_TO_COMMIT = 0,
CONTAINED)
FOR DATABASE [AdventureWorks2022]
```


New Availability Group

Specify Availability Group Options

Introduction
Specify Options
Select Databases
Specify Replicas
Select Data Synchronization
Validation
Summary
Results

Specify availability group options

Availability group name: AG1

Cluster type: Windows Server Failover Cluster

☐ Database Level Health Detection
☐ Per Database DTC Support
☑ Contained
☐ Reuse System Databases

< Previous   Next >   Cancel

Demonstration

Contained Availability Groups

Questions?

Lesson 4: Managed Instance Link

Objectives

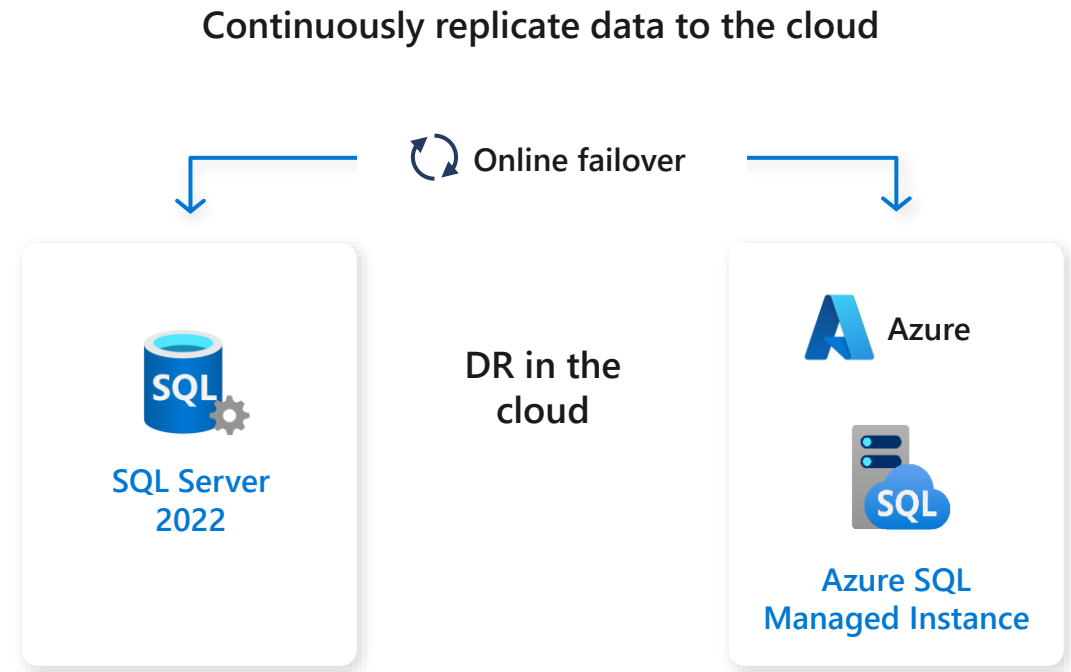After completing this learning, you will know:

· Understand how Managed Instance work.

· Failing over to a Managed Instance.

· For what scenarios to use Managed Instance Link.

Managed Instance Link overview.

Disaster recovery in the cloud.

- Replica in Azure

- Uses distributed AOAG

- Support stand-alone or existing Availability Groups.

- Replicates one database per Managed Instance Link.

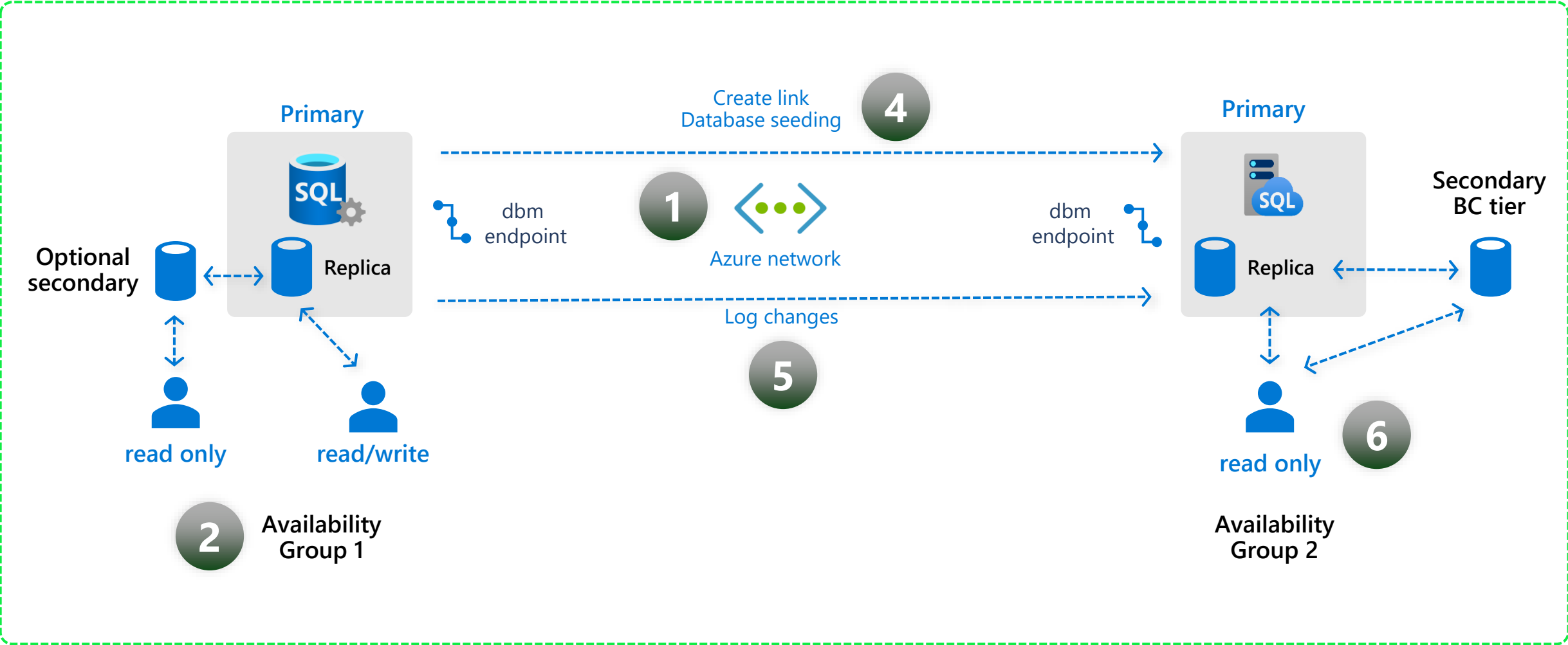- SQL 2016 and 2019 limited functionality.

Continuously replicate data to the cloud

Online failover

DR in the cloud

SQL Server 2022

Azure

Azure SQL Managed Instance

Link feature for Managed Instance

When to use Managed Instance Link

Usage scenarios.

Disaster Recovery site in the cloud.

Offload workloads to Azure.

Migrate to Azure.

Copy data on-premises.

Lesson 5: Availability Group Best Practices

Objectives

After completing this learning, you will know:

- How to manage critical elements of Availability Groups (AG).
- What situations to monitor closely.
- How to handle common issues.
- How to manage security and maintenance in an AG solution.

Key Elements to AG Management

1
Do not use Windows Failover Cluster Manager to perform Availability Group Failover and Administration.

2
Be aware of Windows Hotfixes that may affect your AlwaysOn Availability Group configuration.

3
Be aware of issues that affect transaction log performance:
- Log Generation Rate
- High number of virtual log files
- Replication and Database Snapshots

4
Limits to the number of availability groups and availability databases per computer.

Common Issues with Availability Groups

- Disk drive paths are different on each replica.

- Failover of unsynchronized replicas are causing splits in recovery.

- After failover, the old primary will not resume data sync automatically due to potential data loss.
  - Read the database and copy the data to the new primary if you do not want to lose it.
  - Resume sync with the new primary and transactions not mirrored from the old primary to the new one will be rolled back. It then catches up to the new primary to stay in sync (Potential Data Loss).
  - If you create two or more forks in the transaction log, the recovery process can only recover by using one fork.

- Slow connections in a multi-subnet cluster.

Managing Objects Outside of the AG Database

· Always On Availability Groups help protect at the database / instance level.

· The following objects exist outside of the database, which may require additional consideration in the event of a failover (either planned or unplanned):

| SQL Server Logins | SQL Server Agent Jobs | SQL Server Integration Services Packages | Applications installed on the Availability Group Replicas |

SQL Server Logins

Utilize sp_help_revlogin (same approach as Database Mirroring).

Synchronizing the logins to the servers as the database replicas already have the SID from the primary.

Create a SQL Server Integration Services (SSIS) package to traverse the output of sp_help_revlogin and synchronize the SIDs on a schedule.

Questions?

Knowledge Check

What problems could occur if database file locations differ between replicas?

What are some common items not replicated by Availability Groups?

What are some ideas that could be used to handle the items not replicated by Availability Groups?