

SQLskills Immersion Event IE0: Accidental/Junior DBA

Module 5: Disaster Recovery and High Availability

Jonathan Kehayias
Jonathan@SQLskills.com



Overview

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- Failover Clustering
- Database Mirroring
- Availability Groups
- Log Shipping



2

© SQLskills, All rights reserved.
<http://www.SQLskills.com>



#1 Requirement: SLAs

- **Do you know what SLAs are?**
 - Service Level Agreements
- **Do you know why they're important?**
 - They're a contract to be met and also they're basic design input
- **Do you know what your SLAs are?**
 - YOU are responsible for meeting them
- **Do you think you can meet them?**
 - Have you taken them into account when designing?
- **Once the system is operational, do you know you can meet them?**
 - Have you tested the SLAs on the live system?
- **The answer to all of these should be YES!**

Downtime SLA

- **Maximum allowable downtime or RTO**
 - Recovery Time Objective
- **Commonly discussed in terms of 'number of nines'**
 - 5-nines = 99.999% uptime
 - Slightly over 5 minutes downtime per year
 - 4-nines = 99.99% uptime
 - Almost 52.5 minutes downtime per year
 - 3-nines = 99.9% uptime
 - Almost 8.75 hours downtime per year
- **Must consider how downtime is defined for you**
 - Is it 24x7 or, maybe just 9am-5pm weekdays
- **Achieving 5-nines of 24x7 operation is very hard**
- **Management may ask for zero downtime!**
 - Not technically possible
- **You have to know how technologies introduce downtime**
 - E.g. failover time with database mirroring

Data-Loss SLA

- **Maximum allowable data loss or RPO**
 - Recovery Point Objective
- **Must consider how data loss is defined for you**
 - Number of transactions
 - Work done in a period of time
- **May be different for different tables or databases**
- **Zero data loss is much more easily achievable than 5-nines uptime**
- **Press the business owners for proper answer**
- **Management may ask for zero data loss too!**
 - This IS achievable, but is it absolutely necessary?

Definition of 'High Availability'

- **Minimize or avoid service downtime whether planned or unplanned**
 - Automatic failover ensures interruption is brief or non-existent should a component fail in the architecture
- **Eliminate single points of failure to the extent possible (cost factor)**
 - Redundant components and/or fault-tolerant servers
- **The goal of 'High Availability' is for users/applications to always be able to do what they need to be able to do**
- **Depending on requirements 'High Availability' may protect:**
 - A single table
 - Group of databases
 - Entire server instance
 - An entire data center

Redundant Components

- Implementing redundant components forms the basis for any high-availability strategy by removing as many single points of failure as possible from a given configuration
- Redundant hardware components
 - ECC RAM
 - Multiple I/O paths, switches, and controllers
 - RAID configurations for disks
 - Multiple network cards and switches
- Redundant server nodes with copies of one or more databases
 - May require implementing multiple technologies to achieve business SLAs

What High Availability is NOT...

- HA is NOT a single, pre-packaged solution from any vendor
 - Hardware, software or application
- HA is NOT a technology choice isolated from:
 - Business knowledge and risk assessment
 - The cost of downtime
 - The cost of data loss
- HA is NOT a business decision isolated from:
 - The cost/complexity of design and implementation
 - The cost/complexity of management and monitoring

HA is Also NOT DR

- HA is how you protect something against becoming unavailable
- DR is how you make something available again after it has become unavailable
- Example:
 - Implementing database mirroring is adding an HA technology to protect a database, but performing a manual failover is performing DR using the HA technology after a failure
- Many people consider HA and DR together because they're so interlinked

Disaster Recovery Planning

- Designing a disaster-recovery strategy is integral to designing a highly-available system
- Even with the most sophisticated redundancy, recovery from total loss of all data centers can only be done using backups
 - Even if redundancy is available when disaster strikes, failover may not be automatic or the first resort
- The requirements gathering process feeds into the backup/restore strategy too
 - And the fact that storage space and management are required for backups then feeds back into the requirements
- What restores you need to be able to do depends on:
 - What needs to be brought online first
 - Data loss SLA
 - Downtime SLA

How To Plan an HA/DR Strategy

- First answer from many people is “get a failover cluster” or the latest new whiz-bang feature: wrong!
- It takes a very layered approach
 - Requirements gathering
 - Requirements ordering
 - Limitations gathering
 - Trade-off/compromise
 - Technology evaluation and choice
- It requires the right solution for the problem
- You can’t pick a technology without evaluating requirements
 - Don’t just choose incumbent technologies
- It needs to be constantly re-evaluated, re-examined, updated
- It should just work, even through a fault
 - If your system isn’t designed to protect against the fault that occurs then what’s the point?

Consider Limitations

- There’s no point going through the entire design process only to find that its flawed because of a pre-existing limitation
 - E.g. the design calls for a redundant data centre but the budget is only US\$10k
- Limitations need to be known up front and are just as important as requirements
- There are technical and non-technical limitations

Non-Technical Limitations (1)

- **Power**
 - Consider power requirements for all parts of the system
 - Servers, drives, networking, HVAC
 - Do you need UPSs? Backup generators?
- **Space**
 - Only so many servers you can fit in a closet
 - Consider servers, I/O subsystem, networking, people, HVAC, PSUs
- **Air Conditioning**
 - Too many servers in the closet and they'll overheat
 - More HVAC means more power and space required too
- **Manpower**
 - Who's going to implement, monitor, and maintain the system?
 - Consider costs of failover site – vendor or company operated?
 - One person can't recover multiple systems simultaneously without the possibility of mistakes

Non-Technical Limitations (2)

- **Time to implement**
 - A good design and solid implementation won't happen overnight
 - After a disaster, HA usually becomes the hot topic
- **Single solution provider for hardware or software**
 - E.g. does your hardware vendor provide a SAN replication solution?
 - E.g. does your software vendor provide backup compression?
- **Politics**
 - E.g. who needs to authorize a failover?
 - E.g. which teams need to be involved?
- **Most of the limitations boil down to BUDGET**
 - You can't implement a new data-centre with US\$10k

Technical Limitations (1)

- **What is the transaction log generation rate of your workload?**
 - Impacts database mirroring, log shipping, replication, log file size management, Availability Groups
- **How large are the average transactions?**
 - Impacts all failover technologies, log file size management
 - Smaller transactions affected more on sync mirroring and AGs
- **What is the maximum recovery time?**
- **What recovery model are you running?**
 - Database mirroring requires FULL, impacts index maintenance
- **What's the network bandwidth and latency to the second site?**
 - Impacts ability to implement any cross-site technology
- **Can you alter the application at all?**
 - Impacts ability to failover gracefully
- **What features do you need?**
 - FILESTREAM can heavily impact backup size
 - FILESTREAM not available with database mirroring

Technical Limitations (2)

- **Are you limited in terms of:**
 - CPU architecture
 - 32-bit is limited by worker threads which impacts database mirroring
 - Disk space
 - Impacts ability to implement partitioning, backup strategy, RAID 10
 - I/O subsystem
 - Low throughput can hamper all multi-instance HA technologies
 - Memory
 - Impacts ability to have multi-instance clustering
 - SQL Server Edition
 - Standard Edition can't use asynchronous database mirroring
- **How long does it take the physical server to boot?**
- **Does the hardware in the other data-centers match that in the primary data-center?**
 - If not, workload performance after failover may be lower

Compromise

- **Limitations may stop all requirements being met**
- **If they do, several solutions:**
 - Push back on limitations and/or requirements
 - Accept limitations, prioritize requirements, and design to meet them in order
- **Crucial that management are made aware of which requirements cannot be met and why**
 - Unacceptable for you to only complain for the first time when a disaster strikes

Testing

- **Test the solution before going into production with various disasters**
 - Pull out a drive
 - Drop a table
 - Unplug a network cable
 - And everything else you've specified as a requirement
- **Try doing a bare metal install or a full restore from backups**
- **What if you can't meet your requirements?**
 - Push back or tweak the strategy as appropriate
 - Make sure management knows what's possible and what's not BEFORE going into production
- **Perform regular real-life disaster testing IN production**
 - No other way to test it for real... but easier said than done

When to Pick Technologies?

- **Note I said 'technologies', not 'a technology'**
 - Most solutions involve multiple technologies
- **Once the final, post-compromise requirements are known, THEN and only then is the time to start evaluating technologies**
 - Choosing technologies then evaluating requirements and having to then change technology choice wastes time
- **Make sure to understand technologies before picking them**
- **Success will not come from picking a technology and then trying to make it do something it's not designed to**
 - E.g. picking log shipping for a system that requires zero data-loss
 - E.g. picking database mirroring for a system that requires multiple databases to failover

Evaluating Technologies

- **For each technology, consider:**
 - Cost
 - Complexity
 - Implementation, configuration, manageability
 - Impact on performance
 - Data loss exposure
 - Downtime potential
 - Feature compatibility
 - Does it allow you to meet only some or all of your requirements?
- **If, after evaluating technologies you can't meet all requirements, you'll need to reduce them**

Overview

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- **Failover Clustering**
 - Database Mirroring
 - Availability Groups
 - Log Shipping

Failover Clustering Overview

- High-availability solution built on top of Windows Clustering that provides automatic detection and failover at the instance level
- Only one node can own the instance or service and its dependent resources such as shared disks, IP address, and MSDTC (if configured as a cluster resource)
- Standard Edition supports 2 nodes, Enterprise and Datacenter support the OS-Edition maximum for nodes
 - Up to 16 nodes can participate in a failover cluster
- Supports rolling updates for minimal downtimes during patching
- Only maintains a single copy of the databases on the shared storage
 - Single point of failure

Failover Cluster Requirements

- Active Directory is required for failover clustering
- DNS name resolution must be working properly using fully-qualified domain names (FQDN)
- Shared storage – either SAN or SMB in SQL Server 2012
- Configuration must pass the cluster validation tests in Windows Server 2008 onwards for supportability
- Separate public and private networks are not required but still recommended for redundancy
- MSDTC is not required as a clustered resource

Clustering Concepts

Understanding Quorum

- Quorum determines the number of failures that a cluster can sustain and still remain online
- Quorum exists to handle scenarios where communication between cluster nodes has failed to prevent multiple nodes from trying to host the same resource group(s) simultaneously resulting in a 'split-brain'
- Voting towards quorum
 - The cluster has quorum if more than half of the voters are online and communicating with each other
 - Each cluster node has 1 vote
 - A disk or file share witness can be configured for 1 vote
 - Always configure quorum so that an odd number of votes exists in the cluster
 - 5-node cluster needs 3 nodes online to run, so does a 4-node cluster without a disk or file share witness

Clustering Concepts

Voting Towards Quorum Windows 2008R2 and earlier

- The cluster has quorum if more than half of the voters are online and communicating with each other
- Each cluster node has 1 vote
 - Based on its NodeWeight configuration in Windows Server 2008 onwards
 - <http://support.microsoft.com/kb/2494036>
- A disk or file share witness can be configured for 1 vote
- Always configure quorum so that an odd number of votes exists in the cluster
 - 5-node cluster needs 3 nodes online to run, so does a 4-node cluster without a disk or file share witness
 - Plan for servers participating in the cluster that are in a remote data center when working with geo-clusters

Clustering Concepts

Voting Towards Quorum Windows 2012

- **Dynamic Quorum**
 - Provides the ability of the cluster to recalculate quorum on the fly to maintain a working cluster
 - Provides the ability to continue to run a cluster even if less than 50% of the nodes remain available
 - Allows the cluster to be reduced down to the last node (known as last man standing) and still maintain quorum
- **Dynamic Quorum requires:**
 - The cluster has already achieved quorum
 - Sequential failures of nodes occurs
- **If multiple nodes in a cluster go down simultaneously, dynamic quorum will not recalculate the number of votes to maintain quorum**
 - When this occurs, a regroup must occur to determine quorum can be maintained, and then dynamic quorum will resume when a subsequent node failure occurs

Clustering Concepts

Quorum Types

- **Node Majority**
 - Default for odd number of nodes
- **Node and Disk Majority**
 - Recommended for even number of nodes
 - Requires an additional small clustered disk that can failover between the nodes as a witness and additional voter for quorum
- **Node and File Share Majority**
 - Required for multi-site or geo-clustering
 - Requires a file share in the same AD forest as the cluster nodes

How Failover Works

Failure Detection

- **Node failure is detected by MSCS when a node misses 6 consecutive heartbeats**
- **Prior to SQL Server 2012, resource failure is detected through IsAlive and LooksAlive checks provided by the Resource DLL**
 - SQL Server performs the LooksAlive check every 5 seconds and validates that the SQL Server service status matches the status reported by the Service Control Manager (SCM)
 - SQL Server performs the IsAlive check every 60 seconds by executing the command `SELECT @@SERVERNAME` against the instance
- **SQL Server 2012 leverages `sp_server_diagnostics` for health detection**
- **Intervals for the IsAlive and LooksAlive checks and the number of missed heartbeats to trigger failure can be configured using the Failover Cluster Manager**

How Failover Works

Instance Failover

- When instance failover occurs the resource group for the instance is taken offline on the current owning node
- The Failover Manager performs arbitration to locate a new owner for the group and brings the resources online on the new node in dependency order
- When the SQL Server service starts up instance recovery begins
 - The SQL Resource DLL is up once the master database completes recovery
 - User databases become available when REDO completes for Enterprise Edition or after UNDO completes for Standard Edition
 - Called 'Fast Recovery'

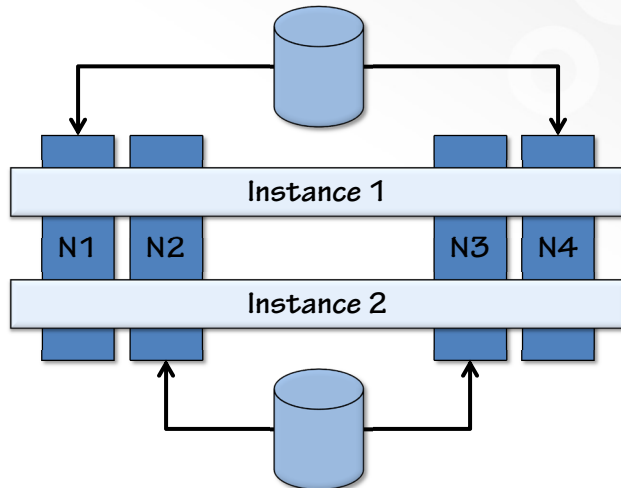
How Failover Works

Failback

- Cluster resource groups can be assigned a 'preferred owner'
- When the preferred owner node comes back online, the cluster resource group can automatically failback to the node based on its configuration
- Failback is not required and should only be configured in multi-node/multi-instance clusters where specific instances should ideally be running on specific nodes
- Cluster resource groups should be configured to perform automatic failbacks outside of normal operating hours only
- Failback is the same as a failover and all client application connections are broken impacting end users

Clustered Instance Rolling Updates

1. Remove N3 & N4 from possible owners
2. Patch N3 & N4
3. Add N3 & N4 to possible owners, remove N1 and N2 from possible owners
4. Failover to N3 & N4
5. Patch N1 & N2
6. Add N1 & N2 back to possible owners



Overview

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- Failover Clustering
- **Database Mirroring**
- Availability Groups
- Log Shipping

Database Mirroring Overview

- Solution for increasing the availability of a single database by maintaining a second copy of the database on another instance
- Provides zero to minimal data loss through a configured database mirroring partnership, which includes a copy of the database, providing redundancy at the database level
- Removes single point of failure when compared to failover clustering
- No hardware dependencies
- Transparent client redirect for client connection management
- Synchronous configurations may impact performance but ensure no data loss is possible, asynchronous configurations allow data loss but do not impact performance
- Deprecated feature in SQL Server 2012

Mirroring Components

- **Principal database/server**
 - The principal server is the server which clients connect to and perform their updates and reporting
- **Mirror database/server**
 - The mirror server is performing the same changes on the mirrored database
- **Witness server (optional, lightweight, even SQL Express)**
 - The witness server monitors status of the principal and mirror servers
 - The witness does NOT trigger the failover, just helps provide 'quorum'
- **Quorum**
 - In the event of the principal becoming unavailable, the mirror can only failover if it can still see the witness, and the witness agrees it cannot see the principal
 - If the mirror fails, principal can only continue if it still sees the witness

Database Mirroring Configurations

- **High Availability: synchronous mirroring with a witness**
 - Automatic detection/failover
 - No data loss
 - Also called SAFETY FULL
- **High Protection: synchronous mirroring without a witness**
 - Manual failover
 - No data loss
 - Also called SAFETY FULL
- **High Performance: asynchronous mirroring**
 - Manual failover
 - Some data loss possible
 - Also called SAFETY OFF
 - Enterprise Edition only

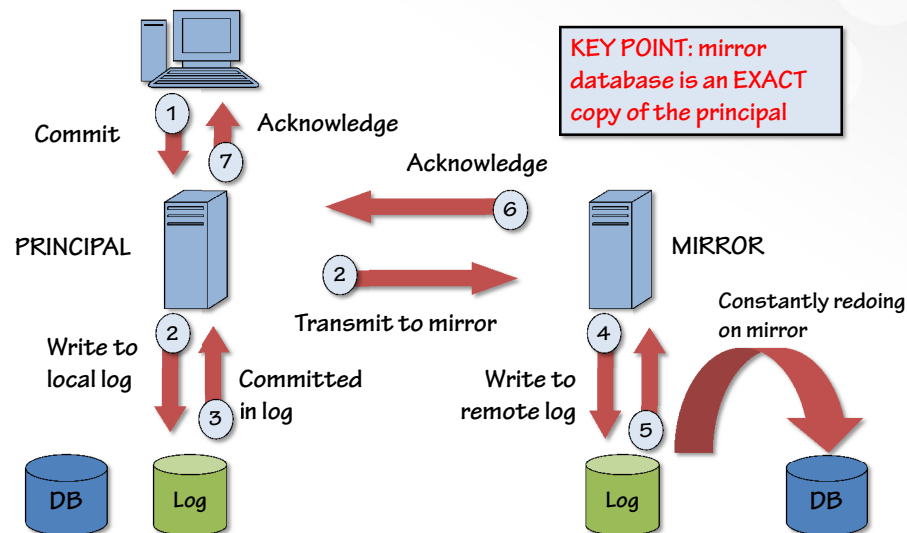
Synchronous Mode

- **Provides the most data protection**
- **Transactions cannot commit on the principal until the log has been written to the mirror's log**
 - Note: But the log doesn't have to be recovered, just written
- **Failover can be either automatic or manual**
 - Based on achieving quorum and the existence of a witness server
 - With a witness: high-availability mode
 - Without a witness: high-protection mode
- **Not achieved until the mirror database has caught up (in SYNCHRONIZED state) with the principal database**

Asynchronous Mode

- Allows for faster processing on the principal server, but at the cost of not having an up-to-date copy of the database on the mirror server
- The principal server commits each new log record locally just before sending the record to the mirror server
- The mirror server continuously applies all the outstanding log records to the mirror database in an effort to catch up to the principal database
- Only available on Enterprise Edition

Principles of Database Mirroring Synchronous, High-Protection Configuration



Requirements for Setup

- Both the principal and mirror servers must have SQL Server 2005+ installed and have enough space for the database
 - For automatic failover, the witness server also must have SQL Server 2005+ installed
- The principal database must use the FULL recovery model
 - Implications for index maintenance
- The mirror database must be “prepared”
- Pick TCP ports for servers involved and open them in firewalls
- Make sure logins have connection permission to endpoints
- Choose encryption requirements
 - BOL: Database Mirroring Transport Security
 - <http://msdn.microsoft.com/en-us/library/ms186360.aspx>

Preparing the Mirror Database

- Back up the principal database
- Restore full backup onto the mirror server using
 - RESTORE DATABASE [dbname] WITH NORECOVERY
- Restore the latest logs to bring the mirror to current
 - RESTORE LOG [dbname] WITH NORECOVERY
- At least one log backup must be restored
- All versions require that no further log backups can be taken on the principal until mirroring is enabled
- All file paths must exist on the mirror, otherwise must use WITH MOVE on the initial restore
- Setup for mirroring

Non-contained Objects and Features

- **Consider non-contained object synchronization requirements to ensure application functionality continues after failover:**
 - Logins (transfer via sp_help_revlogin)
 - SQL Server Agent jobs
 - Custom execution logic to execute as appropriate
 - Linked servers
 - SSIS Packages
 - Replication and Change Data Capture

Overview

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- Failover Clustering
- Database Mirroring
- **Availability Groups**
- Log Shipping

Availability Group Overview

- Availability group contains one or more user databases that fail over together to another instance of SQL Server
- Availability replica is a SQL Server instance (standalone or SQL FCI) that hosts the availability databases
 - Primary replica hosts the read-write databases (only 1 allowed)
 - Secondary replica(s) host the secondary copies, non-writeable copies (up to 4 allowed)
- Availability group listeners support automatic client redirection to the primary replica or redirection to available readable secondaries
- Secondary replicas can also be configured to permit read-only workloads, with low latency updates from the primary

Availability Group Architecture

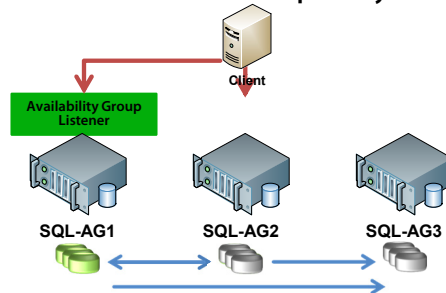
- Leverages WSFC for health detection, failover coordination, and to provide the Availability Group Listener functionality for connectivity across any node
- Up to 5 nodes per Availability Group, 1 Primary and 4 Secondary, allowing two Secondary nodes to be synchronous with automated failover in the environment
- Asynchronous replicas can exist in remote data centers for disaster recovery without requiring secondary technologies like Log Shipping

Synchronous vs. Asynchronous

- **No data loss?**
 - Up to three synchronous replicas
 - Primary replica
 - Two secondaries
 - Beyond that, asynchronous
- **Some data loss allowed? No tolerance for synchronous overhead?**
 - Up to four asynchronous replicas
- **Failover mode**
 - Automatic or manual
 - If automatic, up to two replicas (including current primary)

Application Connection Failover

- Clients connect using the Availability Group Listener Virtual Name
- During a failover the Listener is taken offline and moved to a new node in the Availability Group
- WSFC tells the AG Resource DLL to bring the AG online on a new node
- Clients reconnect as soon as the Listener is back online
- WSFC provides notification of the new primary server to all secondary replicas



Non-contained Objects

- **Similar to database mirroring, you'll need to consider non-contained object synchronization:**
 - Logins (transfer via sp_help_revlogin)
 - SQL Server Agent jobs
 - Custom execution logic to execute as appropriate
 - Linked servers
 - SSIS Packages
 - Replication and Change Data Capture

Multi-Subnet Connection Support

- **SQL Server 2012 supported on Windows Server Failover Clusters with nodes that cross-subnets**
- **Microsoft recommends using the client MultiSubnetFailover attribute for both single and multi-subnet topologies that reference the availability group listener name**
- **Opens up TCP sockets for availability group listener IP addresses in parallel**
- **Older client libraries?**
 - Recommendation is to increase the client login timeout
 - Consider adjusting the HostRecordTTL value to a lower value

Quorum Model

- **Quorum model is managed via the WSFC**
 - Number of 'elements' needed to keep the WSFC running, and also avoid 'split brain' by ensuring majority of elements are visible
- **Must consider non-shared vs. shared-storage models**
 - Node and Disk Majority (shared disk)
 - No Majority: Disk Only (shared disk)
 - Node Majority
 - Node and File Share Majority
- **Odd vs. even number of nodes drives quorum decisions**
 - Node Majority
 - Node and File Share Majority

Using Failover Cluster Manager

- **Failover Cluster Manager (FCM) will let you do bad things to your availability group resources**
- **Don't use FCM to change the following availability group and availability group listener resource settings:**
 - Preferred owners
 - Possible owners
- **Don't use FCM to fail over the availability group or availability group listener**
- **Don't use FCM to add or remove resources contained within the availability group and availability group listener resource groups**

Overview

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- Failover Clustering
- Database Mirroring
- Availability Groups
- **Log Shipping**

Log Shipping Overview

- Log shipping creates a warm standby copy of a database on another server by copying transaction log backups to the standby server and automatically restoring them based on the configuration
- Requires manual application failover to redirect connections to the log shipped secondary
 - May include data loss in the event that the source server becomes unavailable to create a tail log backup or copy log backups that have not been copied to the standby server
- May be used for read-only access to the log shipped secondary copy
 - Requires licensing the standby server appropriately
 - Users must be disconnected for additional log backups to be applied
- Supported on standard hardware
- Less common for HA and more common for DR and maintaining multiple DR sites with high latency connections

Log Shipping Jobs

- **Back up job on the primary server**
 - The primary server instance runs the backup job to back up the transaction log on the primary database and then sends the file to the backup folder
- **Copy backup file job on the secondary servers**
 - Each of the configured secondary server instances runs its own copy job to copy the primary log-backup file to its own local destination folder
- **Restore job on the secondary servers**
 - Each secondary server instance runs its own restore job to restore the log backup from the local destination folder onto the local secondary database

Failover Steps with Log Shipping

- **Copy any uncopied backup files from the backup share to the copy destination folder of each secondary server**
- **Apply each unapplied transaction log backups in sequence to the secondary database**
- **If the original primary server instance is not damaged, back up the tail of the transaction log of the primary database using WITH NORECOVERY**
 - This leaves the database in the restoring state and therefore unavailable to users, allowing you to roll this database forward by applying transaction log backups from the new primary database
- **Recovering the secondary database and redirecting clients to the new server instance**
- **Reconfigure Log Shipping to act as a primary database for other secondary databases or reverse the log shipping configuration**

Review

- Understanding RPO and RTO
- Planning a recovery strategy to meet RPO and RTO
- Failover Clustering
- Database Mirroring
- Availability Groups
- Log Shipping

Questions?