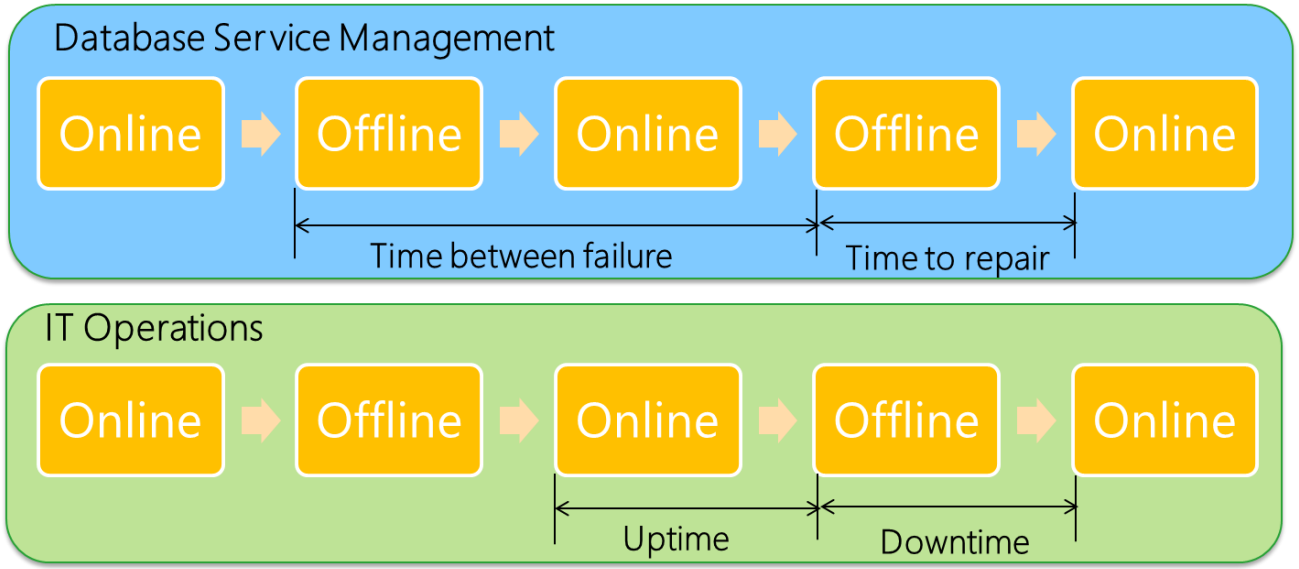
SQL Server Backup

Marek Chmel
Lead Database Administrator, SQL Team at&t Czech Republic
Microsoft MVP Data Platform
Certified Ethical Hacker
Microsoft Certified Trainer: Regional Lead

Session Agenda

- Introduction
- Our first backup
- Making things faster
- Making things more secure

High Availability



$$Availability = \frac{MTBF}{MTBF + MTTR}$$

$$downtime\ per\ year\ (in\ days) = (1 - uptime\ ratio) * 365$$

$$uptime\ ratio = \frac{Availability}{100}$$

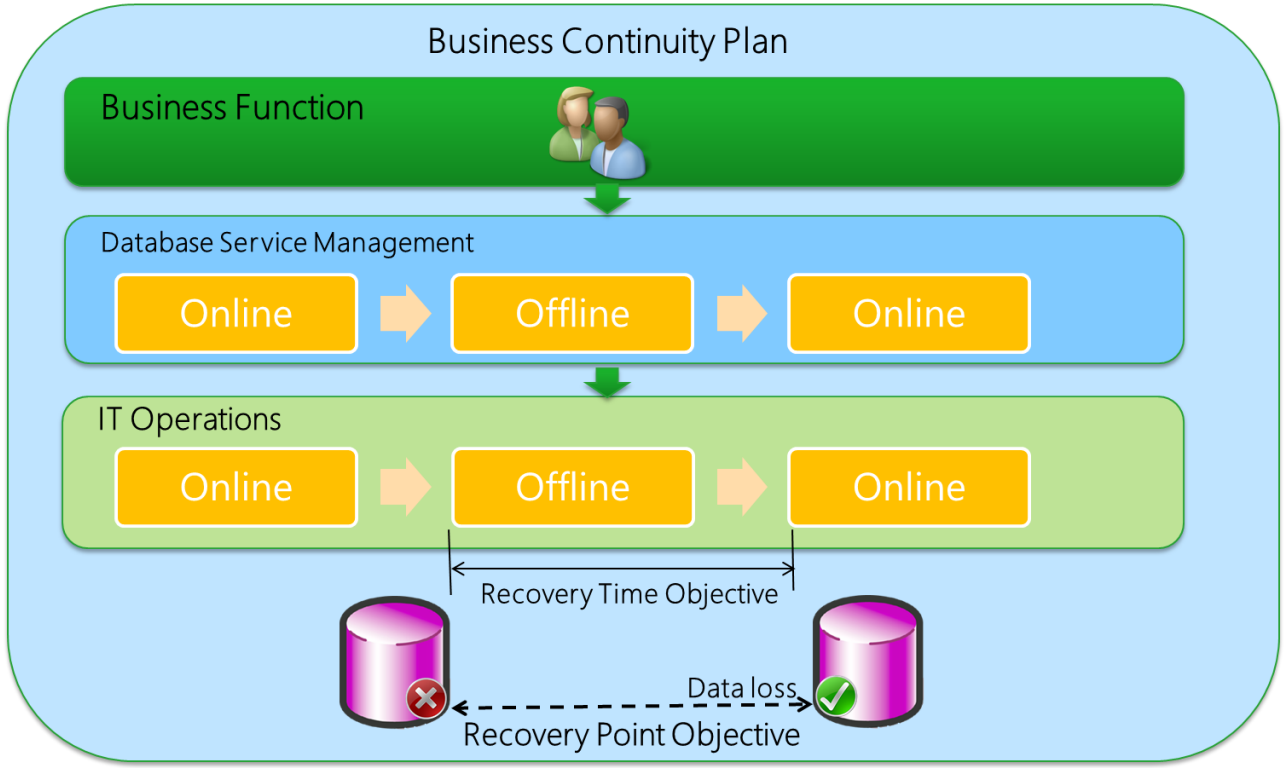
Example

- Operations Log (12 hours)
  - Recovered from previous failure at 00:00:00 Hours
  - Malfunctioned again at 10:00:00 Hours
  - Repaired and operational at 10:06:00 Hours
- Availability (Service)
  - Mean Time Between Failures (MTBF) = 10 Hours
  - Mean Time To Repair (MTTF) = 0.1 Hour
  - Availability = 10/(10+0.1) = 99%
- Downtime (Systems)
  - Uptime ratio = 99/100 = 0.99
  - Downtime per year (in days) = (1-0.99)*365 = 3.65 Days

Availability	Downtime per year
99%	3.65 Days
99.9%	8.76 Hours
99.99%	52.56 Minutes
99.999%	5.26 Minutes
99.9999%	31.5 Seconds
99.99999%	3.15 Seconds

RTO & RPO

- RPO and RTO will define our backup strategy



Recovery models

Restarting a backup chain does require a full backup

- Full Recovery
  - All operations are fully logged
  - Frequently required by SQL features – mirroring, log shipping, AG etc.
- BULK_Logged
  - For minimally logged operations only the start and space allocation are logged in the transaction log
  - Might enhance performance for minimally logged operations
- Simple
  - Same as bulk_logged in it's internal nature (not like full, common misconception), log is truncated whenever a checkpoint occurs
  - You can't make log backups
  - Transactional replication still works!

Backup Types

Differential backups are incremental

- Full backup
  - Copies all the pages from database onto a backup device
- Differential backup
  - Copies only the extents that
- Log backup
  - copies all log records that have been written to the transaction log since the last log backup was made
- Partial backup

- To maintain consistency for either full, differential, or fle backups, SQL Server records the current log sequence number (LSN) at the time the backup starts and again at the time the backup ends. This allows the backup to capture the relevant parts of the log as well.

Backup considerations

Backups prevent access, full backups can't run at the same time as log backups

- Backups are performed online
  - Do not prevent user access
  - Might slow down other operations due to I/O load
- Database must be online for normal backup operations
  - Transaction log backups are still possible on a damaged database
  - Log file must be intact
- Integration with operating system options
  - SQL Writer service provides backup functionality through the Volume Shadow Copy Service (VSS) framework

Running a simple backup

```
BACKUP DATABASE AdventureWorks TO DISK = 'h:\backup\AW.bak'
WITH OPTIONS
```

- Options include many useful parameters
  - Compression – since 2008R2 as a part of STD edition
  - Naming
  - Formating
  - Copy_only
  - Mirroring
  - Checksum
- Destination could be disk, network storage, tape
  - `TO DISK = '\\192.168.1.1\backup\AW.bak'`

Choosing a "*backup*" strategy

- Different backup types can be combined
  - Data based Backups (Full, Differential, Copy, Filegroup, File)
  - Log based Backups (Log , Tail Log)
- Safety levels should be determined
  - How long can recovering take? (RTO)
  - How much data is it acceptable to lose? (RPO)
  - Is it possible to recover the data from other sources?
- Backup strategy should be mapped to requirements
  - Types and frequency of backups
  - Backup media to use
  - Retention period for backups and for media
  - Backup testing policy

Common options

- Full Database Backups:
  - Backup all data and part of the log records
  - Can be used to restore the whole database
  - Permit recovery to backup times only
- A Database and Transaction Log Backup Strategy:
  - Involves at least full and transaction log backups
  - Enables point in time recovery
  - Allows the database to be fully restored in the case of data file loss
- A Differential Backup Strategy:
  - Involves performing full and differential database backups
  - Includes differential backups with only changed data
  - Is useful if only a subset of a database is modified more frequently than the rest of the database

Tail Log backup

- Used to capture the tail of the log before starting a restore sequence
  - Performs a regular log backup
- Options:
  - NORECOVERY when restore operations will follow (database set to RECOVERING state)
  - CONTINUE_AFTER_ERROR when data files are missing or damaged but log files are intact

Running a faster backup

- Multiple backup file
- Compression
  - 2016 can combine compression and encryption with modifying below options
- Special options
  - Maxtransfersize
    - multiples of 65536 bytes (64 KB) ranging up to 4194304 bytes (4 MB)
  - Buffercount
    - Specifies the total number of I/O buffers to be used for the backup operation
    - (NumberofBackupDevices * [mystery_multiplier]) + NumberofBackupDevices + (2 * NumberofVolumesInvolved)
  - Blocksize
    - 512, 1024, 2048, 4096, 8192, 16384, 32768, and 65536 (64 KB)

Securing a backup

- SQL Server does not check the source of the backup file
- Limitations may apply for restoring a backup to new SQL Server
  - Version check
  - Edition check
- Securing a backup with permissions
- Securing a backup with encryption
  - Transparent Data Encryption
  - Backup Encryption

Few words about restore

You can restore the backup to lower version of SQL server.

- Perform a tail-log backup if needed
  - Only applies to full and bulk-logged recovery model
- Identify the backups to restore
  - Last Full, File or Filegroup backup as a base
  - Last differential backup, if applicable
  - Log backups if using full and bulk-logged recovery model
- The restore process of a SQL Server® 2012 database happens in three phases

Phase	Description
Data Copy	Creates files and copies data to the files
Redo	Applies committed transactions from restored log entries
Undo	Rolls back transactions that were uncommitted at the recovery point

WITH RECOVERY Option

- A database must be recovered before it can be brought online
- WITH RECOVERY (default) restore option performs a recovery after restore and brings the database online
- WITH NORECOVERY restore option leaves the database in a recovering state
  - Allows additional restore operations on the database
- Restore process always involves
  - WITH NORECOVERY for all backups restored except the last
  - WITH RECOVERY for the last backup restored

System DB Recovery

System Database	Description
master	Backup Required: Yes Recovery Model: Simple Restore using Single User Mode
model	Backup Required: Yes Recovery Model: User configurable Restore using –T3608 trace flag
msdb	Backup Required: Yes Recovery Model: Simple (default) Restore like any user database
tempdb /resource	No backups can be performed tempdb is created during instance startup Restore resource using file restore or setup

Session End

Ing. Marek Chmel, MSc

Lead Database Administrator, Cloud, Platform, Application & Data Layer Team

marek.chmel@att.com