VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

TECHNOLOGIE VYSOKÉ DOSTUPNOSTI MS SQL SERVERU

HIGH AVAILABILITY MICROSOFT SQL SERVER

DIPLOMOVÁ PRÁCE
MASTER´S THESIS

AUTOR PRÁCE — Bc. PAVEL PYSZKO
AUTHOR

VEDOUCÍ PRÁCE — Doc. Ing. VÁCLAV ZEMAN, Ph.D.
SUPERVISOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor

Telekomunikační a informační technika

Student: Bc. Pavel Pyszko ***ID:*** 119586

Ročník: 2 ***Akademický rok:*** 2014/2015

NÁZEV TÉMATU:

Technologie vysoké dostupnosti MS SQL Serveru

POKYNY PRO VYPRACOVÁNÍ:

Věnujte se problematice vysoké dostupnosti v rámci MSSQL 2005, 2008, 2008 R2, 2012, 2014. Porovnejte jednotlivé technologie a určete jejich výhody a nevýhody. Technologie aplikujte ve Vámi vybraném prostředí. Rozeberte jejich vlastnosti, funkčnost, bezpečnost. Navrhněte optimální varianty využití jednotlivých technologií.

DOPORUČENÁ LITERATURA:

[1] WHALEN, Edward. Microsoft SQL Server 2005: velký průvodce administrátora. Vyd. 1. Brno: Computer Press, 2008, 1080 s. Administrace (Computer Press). ISBN 978-80-251-1949-5.

[2] WALTERS, E. Robert. Mistrovství v Microsoft SQL server 2008. 1. vyd. Brno: ComputerPress,2009. 864 s. ISBN 978-80-251-2329-4.

Termín zadání: 9.2.2015 ***Termín odevzdání:*** 26.5.2015

Vedoucí práce: doc. Ing. Václav Zeman, Ph.D.

Konzultanti diplomové práce: Ing. Boris Bělousov

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Purkynova 118, CZ-61200 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

ANOTACE:

Diplomová práce obsahuje ucelený teoretický přehled o technologiích vysoké dostupnosti v prostředí Microsoft SQL Serveru. U každé technologie je uveden návod pro nasazení technologie. Technologie jsou rozebrány z pohledu bezpečnosti, jsou určeny výhody a nevýhody použití technologií v praxi a je určena optimální varianta využití dané technologie. Technologie vysoké dostupnosti jsou mezi sebou porovnány a je uvedena dostupnost jednotlivých technologií ve verzích MS SQL Serveru. Diplomová práce obsahuje tři scénáře, s praktickou ukázkou využití technologií vysoké dostupnosti v praxi. Je uveden rozbor funkcí vysoké dostupnosti v prostředí Oracle a následně jsou porovnány funkce vysoké dostupnosti v prostředí Oracle, s technologiemi vysoké dostupnosti v prostředí MS SQL Serveru.

Klíčová slova:
vysoká dostupnost, Microsoft SQL Server, failover, záloha, obnova

ABSTRACT:

The thesis contains a complete theoretical overview of high availability technologies in Microsoft SQL Server. For each technology, guidance is provided for the deployment of technologies. Technologies are analyzed from a security perspective, they are determined advantages and disadvantages of using technology in practice and is determined the optimal variant of use of the technology. High availability technology are compared with each other and is given the availability of individual technologies in versions of MS SQL Server. The thesis contains three scenarios with practical examples of using technology for high availability in practice. It provides an analysis of high-availability features in Oracle and are subsequently compared high availability features in Oracle environments with high availability technology in MS SQL Server.

Key words:
high availability, Microsoft SQL Server, failover, backup, restore

Bibliografická citace mé práce:

PYSZKO, P. *Technologie vysoké dostupnosti MS SQL Serveru.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 80 s. Vedoucí diplomové práce doc. Ing. Václav Zeman, Ph.D..

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Technologie vysoké dostupnosti MS SQL Serveru jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne ....

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Doc. Ing. Václavu Zemanovi, Ph.D. za metodickou a velmi užitečnou odbornou pomoc a cenné rady při zpracování mé diplomové práce. Dále děkuji Ing. Borisi Bělousovi za poskytnutí testovacího prostředí a taktéž za odbornou pomoc při zpracování mé diplomové práce.

V Brně dne ....

podpis autora

Obsah

Seznam obrázků viii

Seznam tabulek ix

Seznam výpisů kódu ix

Úvod 1

1. Technologie vysoké dostupnosti 2

1.1. Záloha a obnova databází 2

1.1.1. Zálohovací média, typy zálohování, schéma obnovy a rotace úložišť 2

1.1.2. Možnosti a postup zálohování a obnovy databáze 4

1.1.3. Bezpečnost, výhody a nevýhody při zálohování a obnově databází 7

1.1.4. Optimální varianta pro využití zálohy a obnovy databází v praxi 8

1.2. Záloha a obnova serveru 10

1.2.1. Typy zálohování a obnovy serveru 10

1.2.2. Hyper-V funkce určené pro zálohu a obnovu serverů 10

1.2.3. Bezpečnost, výhody a nevýhody použití zálohování a obnovy serverů v prostředí Hyper-V 12

1.2.4. Optimální varianta využití technologií Hyper-V 12

1.3. Zrcadlení databáze 12

1.3.1. Princip, typy, stavy a provozní režimy zrcadlení databáze 12

1.3.2. Nasazení zrcadlení databáze 16

1.3.3. Bezpečnost, výhody a nevýhody použití zrcadlení databáze 18

1.3.4. Optimální varianta využití technologie zrcadlení databáze 18

1.4. Replikace 19

1.4.1. Replikace dat v prostředí server-klient, server-server a typy replikací 19

1.4.2. Nasazení replikací 21

1.4.3. Bezpečnost, výhody a nevýhody použití replikací 23

1.4.4. Optimální varianta využití replikací 24

1.5. Odesílání souboru protokolu 25

1.5.1. Termíny, definice a princip odesílání souboru protokolu 25

1.5.2. Postup k nasazení odesílání souboru protokolu 27

1.5.3. Bezpečnost, výhody a nevýhody odesílání souboru protokolu 30

1.5.4. Optimální varianta využití odesílání souboru protokolu 30

1.6. AlwaysOn Failover Cluster Instance 31

1.6.1. Přehled a princip Failover Cluster instance 31

1.6.2. Nasazení Failover Cluster Instance 32
1.6.3. Bezpečnost, výhody a nevýhody Failover Cluster Instance 34
1.6.4. Optimální varianta využití technologie Failover Cluster Instance 35
1.7. AlwaysOn Availability Group 37
1.7.1. Termíny a definice, princip, režimy a failover AlwaysOn Availability Group 37
1.7.2. Nasazení technologie AlwaysOn Availability Group 39
1.7.3. Bezpečnost, výhody a nevýhody AlwaysOn Availability Group 44
1.7.4. Optimální varianta využití AlwaysOn Availability Group 44
1.8. Dostupnost a porovnání technologií vysoké dostupnosti 46
2. Scénáře pro využití technologií vysoké dostupnosti v praxi 48
2.1. Scénář první, varianta pro malé a střední firmy 48
2.1.1. Popis současného stavu 48
2.1.2. Požadavky na výsledné řešení 49
2.1.3. Návrh řešení 49
2.2. Scénář druhý, shromažďovaní dat na centrálním serveru 56
2.2.1. Popis současného stavu s požadavky na výsledné řešení 56
2.2.2. Návrh řešení 56
2.3. Scénář třetí, datové centrum 57
2.3.1. Popis současného stavu, požadavky na řešení 57
2.3.2. Návrh řešení 58
3. Porovnání MS SQL s produkty technologií vysoké dostupnosti od jiných firem 60
3.1. Funkce vysoké dostupnosti Oracle databáze 60
3.2. Porovnání technologií vysoké dostupnosti v prostředí Oracle a MS SQL 62
4. Testovací prostředí 63
4.1. Popis testovacího prostředí 63
4.2. Ne-clusterované prostředí 63
4.3. Windows Server Cluster prostředí 64
4.3.1. Instalace a nastavení failover clusteru 65
Závěr 67
Seznam literatury 68
Seznam symbolů a zkratek 69

Seznam obrázků

Obr. 1.1: Milníkové schéma obnovy. 3
Obr. 1.2: Rozdílové schéma obnovy. 3
Obr. 1.3: Inkrementální schéma obnovy. 4
Obr. 1.4: Obnova databáze pomocí SQL Server Management Studia. 5
Obr. 1.5: Průvodce nastavení plánu údržby a plánovač zálohování. 6
Obr. 1.6: Další funkce plánu údržby 6
Obr. 1.7: GFS rotační schéma 9
Obr. 1.8: Vývojový diagram použité rotace uložišť. 9
Obr. 1.9: Příklad replikace virtuálních počítačů. 10
Obr. 1.10: Ukázka exportu virtuálního počítače ve správci technologie Hyper-V. 11
Obr. 1.11: Ukázka importu virtuálního počítače ve správci technologie Hyper-V. 11
Obr. 1.12: Architektura zrcadlení databáze. 13
Obr. 1.13: Konfigurace relace v režimu vysoký výkon. 14
Obr. 1.14: Konfigurace v režimu vysoké ochrany bez automatického předání služby při selhání. 14
Obr. 1.15: Konfigurace v režimu vysoké bezpečnosti s automatickým předáním služby při selhání. 15
Obr. 1.16: Nastavení full recovery módu. 16
Obr. 1.17: Obnovení databáze v NORECOVERY režimu. 16
Obr. 1.18: Postup k nastavení zrcadlení databáze. 17
Obr. 1.19: Nastavení zrcadlení databáze. 17
Obr. 1.20: Monitor zrcadlení databáze. 18
Obr. 1.21: Vytvoření a shrnutí konfigurace distributora. 21
Obr. 1.22: Vytvoření a shrnutí konfigurace vydavatele. 22
Obr. 1.23: Vytvoření a shrnutí konfigurace odběratele. 23
Obr. 1.24: Monitor replikací. 23
Obr. 1.25: Typická konfigurace odesílání souboru protokolu. 26
Obr. 1.26: Povolení odesílání souboru protokolu na primární databázi. 27
Obr. 1.27: Nastavení zálohy transakčního protokolu. 28
Obr. 1.28: Nastavení sekundární databáze. 28
Obr. 1.29: Nastavení monitorovací instance. 29
Obr. 1.30: Souhrn Instalace SQL Server Failover Cluster instance. 33
Obr. 1.31: Souhrn instalace u přidání Failover Cluster uzlu. 34
Obr. 1.32: Kombinace technologií vysoké dostupnosti FailoverClustering a zrcadlení. 36
Obr. 1.33: Schéma AlwaysOn Availability Group s jednou primární replikou a čtyřmi sekundárními. 38
Obr. 1.34: Zapnutí funkce AlwaysOn Availability Group na instanci SQL Serveru. 40
Obr. 1.35: Vytvoření nové Availability Group 40
Obr. 1.36: Konfigurace replik v Availability Group. 41
Obr. 1.37: Zálohovaní databází Availability Group. 42
Obr. 1.38: Konfigurace Availability Group Listener. 42
Obr. 1.39: Nastavení prvotní synchronizace dat v Availability Group. 43
Obr. 1.40: Nástěnka Availability Group. 43
Obr. 1.41: Kombinace Failover Cluster Instance a Availability Group. 45
Obr. 2.1: Předchozí schéma sítě firmy X 48
Obr. 2.2: Návrh výsledného schéma sítě. 52
Obr. 2.3: Scénář 2, současné schéma serverů. 56

Obr. 2.4: Scénář 3, schéma řešení vysoké dostupnosti v prostředí datových center. 58
Obr. 4.1: Průvodce vytvořením Clusteru. 65
Obr. 4.2: Konfigurace fondu úložišť a disku. 66
Obr. 4.3: Schéma sítě clustrovaného testovacího prostředí. 66

Seznam tabulek

Tab. 1.1: Výhody a nevýhody použití technologie zálohování a obnovy databáze 7
Tab. 1.2: Shrnutí výhod a nevýhod použití zálohy a obnovy serveru v prostředí Hyper-V. 12
Tab. 1.3: Stavy zrcadlení databáze. 15
Tab. 1.4: Výhody a nevýhody použití zrcadlení databáze. 18
Tab. 1.5: Výhody a nevýhody použití replikace. 24
Tab. 1.6: Výhody a nevýhody použití odesílání souboru protokolu. 30
Tab. 1.7: Výhody a nevýhody použití FCI. 35
Tab. 1.8: Výhody a nevýhody použití AlwaysOn Availability Group. 44
Tab. 1.9: Porovnání technologií vysoké dostupnosti. 46
Tab. 1.10: Dostupnost technologií vysoké dostupnosti v jednotlivých verzích SQL Serveru. 47
Tab. 2.1: Návrh serverů. 50
Tab. 2.2: Kupované licence pro firmu X. 51
Tab. 3.1: Porovnání technologií vysoké dostupnosti v prostředí Oracle a MS SQL. 62

Seznam výpisů kódu

Výpis kódu 1.1: Záloha databáze pomocí skriptu. 5
Výpis kódu 1.2: Komprimace databáze a kopírování souboru. 8
Výpis kódu 2.1: Powershell skript pro přenos virtuálních počítačů. 52
Výpis kódu 2.2: Kód dávkového souboru pro komprimaci záloh VM a následný přenos na vzdálené úložiště. 55
Výpis kódu 2.3: Kód dávkového souboru pro komprimaci záloh VM a následný přenos na vzdálené diskové úložiště. 55

Úvod

V současné době, kdy technologie se vyvíjí velmi rychlým tempem, jsou informace velmi důležitým a chráněným zdrojem dat. Ztráta důvěrných či tajných informací vede k nepředstavitelným potížím, jak pro firmy, tak pro jednotlivce. Částečnou ochranu před ztrátou informací uložených v relačních databází představují technologie vysoké dostupnosti.

Co pojem vysoká dostupnost znamená? Vysoká dostupnost znamená, že v případě plánovaného či neplánovaného výpadku serveru, organizace může dál pokračovat na přijatelné úrovni a nedojde ke ztrátě dat. Přijatelná úroveň se liší dle organizace. Jedna organizace považuje za přijatelný výpadek 5 hodin, druhá 5 minut. Jedná se tedy o překlenovací řešení v případě plánovaného (údržba systému) či neplánovaného výpadku (havárie) serveru, kde délku výpadku a následné obnovení systému ovlivňují použité technologie vysoké dostupnosti. Dostupné technologie vysoké dostupnosti v prostředí Microsoft SQL Serveru jsou: záloha a obnova databází, zrcadlení databáze, replikace, odesílání souboru protokolu a technologie AlwaysOn, která se rozděluje na Failover Cluster Instance a Availability Group.

Jedním z cílů diplomové práce je poskytnout ucelený teoretický přehled technologií vysoké dostupnosti převážně v prostředí Microsoft SQL Serveru, provést uživatele kroky instalace jednotlivých technologií, vytyčit výhody a nevýhody použití technologie v praxi, rozebrat technologie z pohledu bezpečnosti, porovnat technologie mezi sebou a navrhnout optimální variantu využití dané technologie. Dalším cílem je částečně rozebrat dostupné technologie vysoké dostupnosti od jiných dodavatelů software a následně provést srovnání s Microsoft SQL Server. Posledním cílem je pak pomocí fiktivních scénářů, které simulují výběrové řízení v praxi, vypracovat návod, jakou technologií vysoké dostupnosti v daném scénáři použít a jakým způsobem postupovat. Uživatel tak získá znalosti, potřebné pro použití technologie vysoké dostupnosti ve vlastním prostředí.

1. Technologie vysoké dostupnosti

Diplomová práce se zabývá technologiemi vysoké dostupnosti v rámci databáze nebo instance SQL Serveru či technologiemi, které s tímto tématem úzce souvisí. Hlavním produktem, zabývajícím se v diplomové práci, je produkt od firmy Microsoft, MS SQL Server. Existují však i produkty jiných firem, které obsahují řešení vysoké dostupnosti v databázovém prostředí.

Hlavním konkurentem je firma Oracle, která poskytla ucelené řešení vysoké dostupnosti v databázovém prostředí již v roce 2005. Microsoft sice řešení vysoké dostupnosti ve starších verzích MS SQL Serveru také podporuje, ale řešení nezahrnovalo všechny potřebné funkce pro úplné pokrytí vysoké dostupnosti. Až od verze MS SQL 2012, kdy Microsoft uvedl na trh novou technologii vysoké dostupnosti AlwaysOn se produkty vyrovnaly.

Na trhu jsou nejvíce využívaný produkty v databázovém prostředí od firem Oracle a Microsoft. Další firmy, které disponují produkty v databázovém prostředí podporující vysokou dostupnost, jsou PostgreSQL, Sybase, SAP a další.

1.1. Záloha a obnova databází

Zálohování a obnova databází představují operace, kde administrátor pomocí různých nástrojů např. SQL Server Management Studia (SSMS), ručně nebo automaticky provádí zálohu a obnovu vybraných databází. Jedná se o minimální ochranu před havárií, kterou by každá firma nebo osoba, obsahující důležitá data v prostředí Microsoft SQL Server, měla podstoupit.

1.1.1. Zálohovací média, typy zálohování, schéma obnovy a rotace úložišť

Pro zálohu a obnovu databází v prostředí Microsoft SQL Serveru se nejvíce využívá nástroj SQL Server Management Studio, který má následující vlastnosti:

- Je součástí instalačního balíčku MS SQL Serveru.
- Jedná se o integrované grafické prostředí pro správu serverové infrastruktury SQL.
- Pomocí nástroje je možné vytvářet zálohovací či obnovovací skripty, plány údržby nebo ručně zálohovat nebo obnovovat vybrané databáze.

Mezi další nástroje umožňující zálohování a obnovu databází, produkt od firmy Arcserver, Arcserver Backup.

Mezi nejpoužitelnější zálohovací média lze zařadit:

- pevné disky a disková pole, které lze dále rozdělit na:
  - Lokální disky v počítači.
  - Přenosné flash disky.
  - Síťové disky, pole (např. Storage Area Network- SAN[1]).
- CD a DVD média.
- Magnetopáskové paměti.

Microsoft SQL Server umožňuje zálohovat následující databáze:

- Systémové databáze, databáze potřebné pro běh SQL Serveru (master, model, temp, msdb).

[1] Storage Area Network (SAN) je dedikovaná vysokorychlostní síť, která propojuje a prezentuje sdílené disky z úložných zařízení na více serverů [6].

- Uživatelské databáze, aplikační databáze jednotlivých systémů.

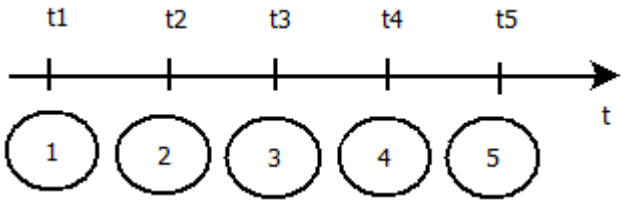
Jestli už víme, na jaké médium zálohy budeme ukládat a jaké databáze zálohovat, je třeba ještě určit typ zálohování:

- Úplná záloha (Full backup) -kopie veškerých dat, která se v okamžiku *t* nacházela v hlavním úložišti.
- Dílčí záloha, intervalová záloha.
- Záloha transakčního logu.

Typy zálohování je možné různě kombinovat, v důsledku toho se používají následující schéma obnovy [4]:

- Milníkové:
    - Výhradně s úplných záloh.
    - Nejrychlejší obnova, velké objemy záloh.

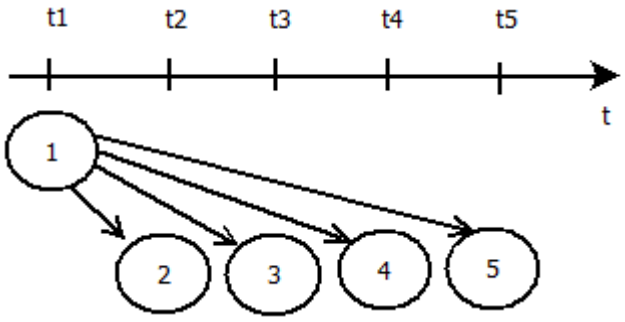
Obr. 1.1 zobrazuje milníkové schéma obnovy, kde t1 až t5 značí čas, kdy se provedla úplná záloha databáze (obvykle pracovní dny v týdnu) a číslo v kruhu určuje pořadí zálohy.



Obr. 1.1: Milníkové schéma obnovy.

- Rozdílové:
    - První záloha úplná a všechny ostatní intervalové, přičemž je pro ně referenční zálohou první záloha.
    - Střední objemy záloh, rychlá obnova.

Obr. 1.2 zobrazuje rozdílové schéma obnovy, kde t1 až t5 značí čas, kdy se provedla úplná záloha databáze (obvykle pracovní dny v týdnu) a číslo v kruhu určuje pořadí zálohy.

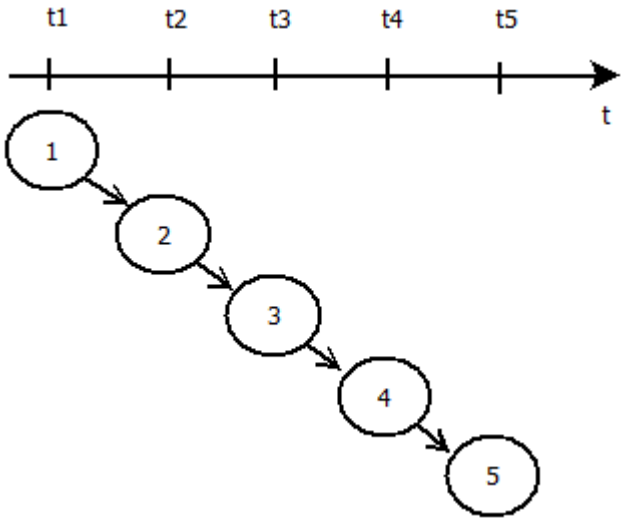


Obr. 1.2: Rozdílové schéma obnovy.

- Inkrementální:
    - První záloha úplná a všechny ostatní intervalové, přičemž pro všechny je referenční zálohou nejbližší předchozí záloha.

- Nejmenší objemy záloh, nejpomalejší obnova.

Obr. 1.3 zobrazuje milníkové schéma obnovy, kde t1 až t5 značí čas, kdy se provedla úplná záloha databáze (obvykle pracovní dny v týdnu) a číslo v kruhu určuje pořadí zálohy.



Obr. 1.3: Inkrementální schéma obnovy.

- Kombinované – kombinace výše uvedených typů schémat.

Vzhledem k omezené kapacitě úložiště je zapotřebí stanovit pravidla přepisování starých záloh novými zálohami. Tato pravidla se nazývají rotace úložišť. Nejznámější rotace úložišť [4]:

- Ploché rotace (jen úplné).
- Hierarchické rotace (referenční):
  - 6 úložišť.
  - GFS (Děd - Otec - Syn).
  - Hanojské věže.

Pro různé podmínky se používají různé rotace. Volba správné rotace je závislá na tom, jestli je potřeba se zálohami pracovat velmi často nebo je naopak požadovaná maximální délka archivace zálohovaných dat. V případě velmi časté manipulace s databázemi je vhodné provádět zálohování častěji, pomocí úplných záloh, milníkové schéma obnovy. Naopak pokud požadujeme dlouhou archivaci dat, je vhodné zálohovat po delších časových intervalech s některou s dílčích metod.

1.1.2. Možnosti a postup zálohování a obnovy databáze

Nástroj SQL Server Management Studio umožňuje několik typů zálohování. Mezi nejpoužívanější patří:

- Zálohování pomocí skriptu v jazyce SQL.
- Ruční zálohování v grafickém prostředí studia.
- Zálohování pomocí plánů údržby

Zálohování a obnova pomocí skriptu

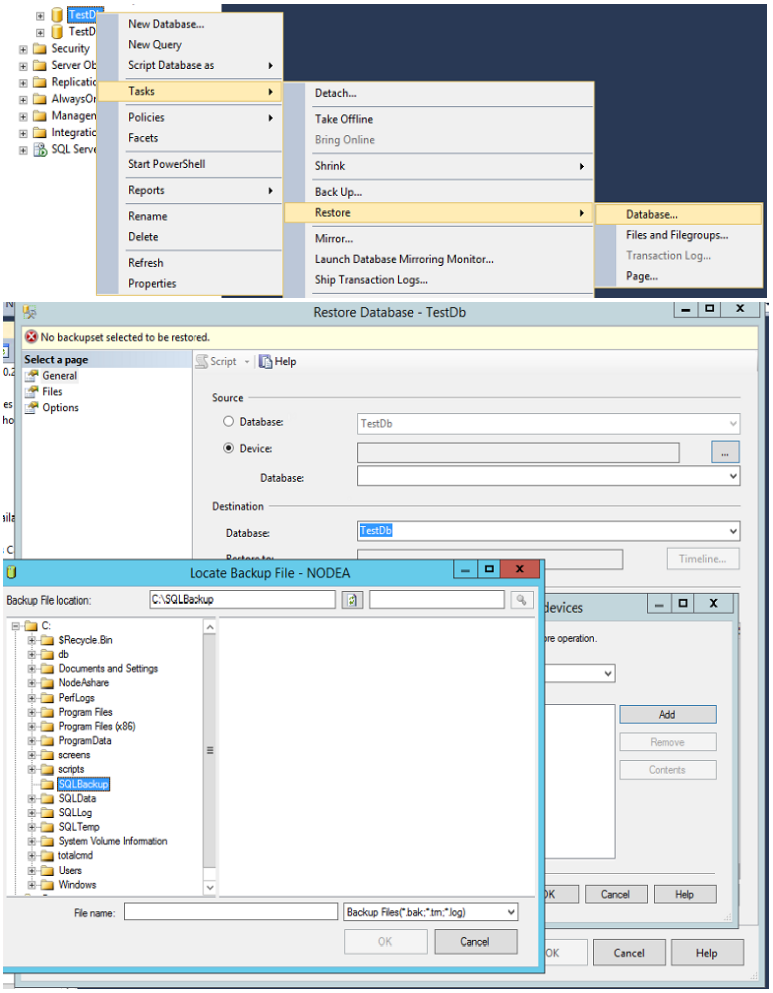
Výpis kódu 1.1: Záloha databáze pomocí skriptu.

```
BACKUP DATABASE [TestDb] TO DISK =
N'C:\nodeaShare\TestDb_20150216190003.trn' WITH NOFORMAT,
NOINIT,  NAME = N'TestDb-Full Database Backup', SKIP,
NOREWIND, NOUNLOAD,  STATS = 10
```

Výpis kódu 1.1 zálohuje databázi TestDb do určené cesty s volitelnými parametry. Pro opakované použití, je skript možné spouštět předem nadefinovanou naplánovanou úlohou nebo SQL jobem[2], můžeme tak dodržet zvolenou metodu zálohování. Zálohování pomocí skriptu však vyžaduje znalosti v oblasti jazyka SQL, naplánovaných úloh či SQL jobů, čímž je pro administrátora výhodnější použít zálohování pomocí grafickém prostředí SSMS.

Ruční zálohování a obnova v grafickém prostředí SQL Server Management Studia

Jedná se o jednoduchý způsob využívaný většinou v případě, kdy administrátor chce provést jednorázovou zálohu, obnovu databáze. K pravidelnému zálohování je nejvhodnější použít poslední způsob, a to zálohování pomocí plánů údržby. Obnova databáze nelze provést pomocí plánů údržby, takže postup obnovy databáze je pro oba způsoby stejný viz obr. 1.4.

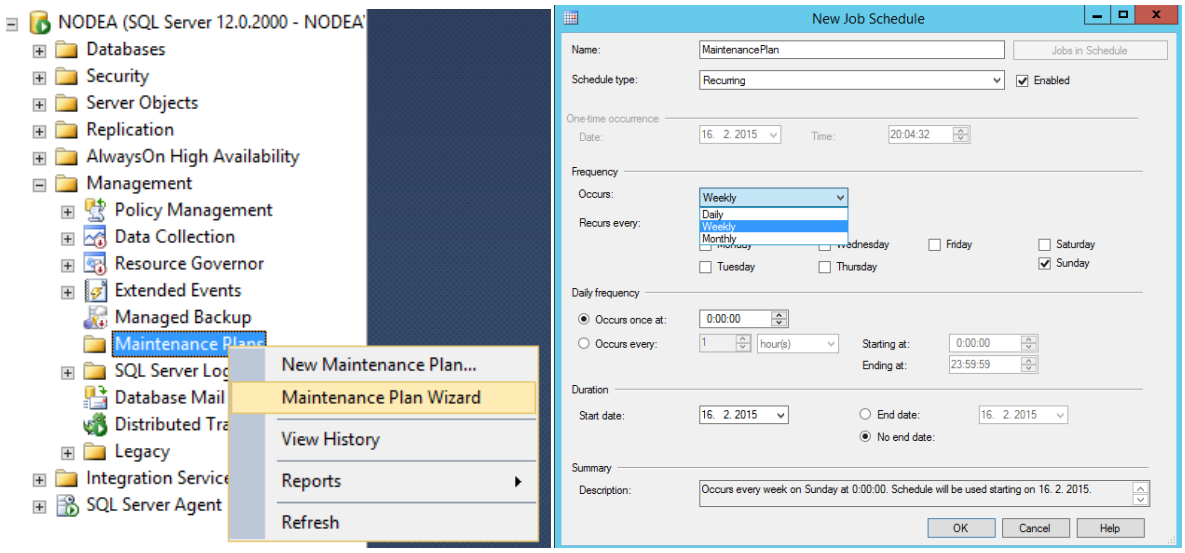


Obr. 1.4: Obnova databáze pomocí SQL Server Management Studia.

[2] SQL job – naplánovaná úloha v prostředí SQL Server Agent [5].

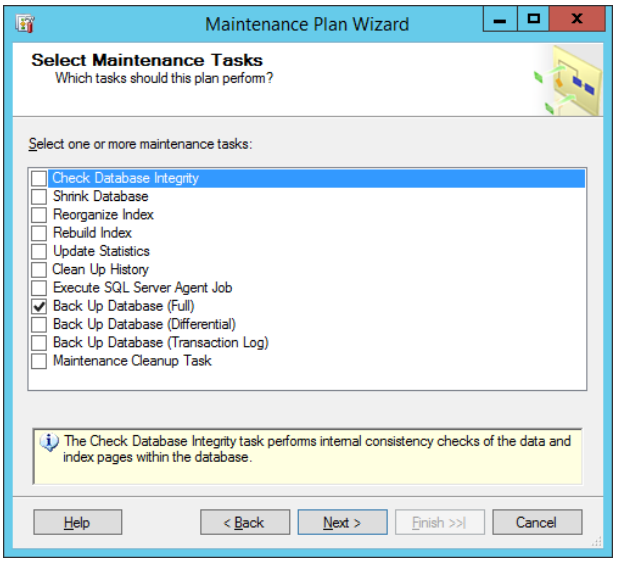
Zálohování pomocí plánů údržby

Vytvořit plán údržby lze buď za pomocí průvodce, nebo přímo v grafickém prostředí. Na obr. 1.5 je uveden postup k vytvoření plánu údržby pomocí průvodce a plánovač zálohování, což je jeden krok průvodce. Pomocí plánovače lze naplánovat spouštění záloh dle vlastní potřeby.



Obr. 1.5: Průvodce nastavení plánu údržby a plánovač zálohování.

Plán údržby ke klasickému zálohování umožňuje přidat další funkce, jako např. kontrolu integrity databáze, vyčištění historie a další. Seznam všech dostupných úloh je uveden na obr. 1.6.



Obr. 1.6: Další funkce plánu údržby

V dalších krocích průvodce se nastavují parametry zvolných funkcí plánu údržby a před samotnou instalací se ke kontrole zobrazí přehled nastavené konfigurace. Po vytvoření plánu údržby, lze plán upravovat a měnit parametry dle potřeby.

1.1.3. Bezpečnost, výhody a nevýhody při zálohování a obnově databází

Strategie obrany do hloubky, s překrývajícími se vrstvami zabezpečení je nejlepší způsob, jak čelit bezpečnostním hrozbám.

První stupněm ochrany je nastavení oprávnění k přístupu do operačního systému, kde je nainstalován SQL Server. Přístup by měli mít pouze administrátoři spravující server, aby nedošlo k nechtěnému vypnutí serveru či instalaci aplikací, které znemožní správný chod SQL Serveru. Pro běžné uživatele lze vytvořit přístup na SQL Server pomocí SSMS nainstalovaného na lokální stanici či na terminálovém serveru.

Další stupeň, je ochrana na úrovni SQL Serveru, který poskytuje následující možnosti zabezpečení:

- Autentizace SQL Server – jedná se o první stupeň ochrany SQL Serveru, kde uživatel musí znát údaje pro přihlášení na SQL Server. Autentizace je dvojího typu, windows autentizace a smíšená autentizace.
- Databázové a serverové role SQL Serveru – pomocí rolí lze nastavit uživatelům oprávnění k přístupu na server či na danou databázi a přiřadit oprávnění k jednotlivým operacím.
- Autorizace a práva v SQL Server – v případě vytvoření databázového objektu musíte přiřadit oprávnění na daný objekt, aby byl přístupný pro zvolené uživatele.
- Šifrování dat v SQL Serveru – SQL Server poskytuje funkce pro šifrování a dešifrování dat pomocí certifikátu a asymetrických nebo symetrických klíčů.

Tyto body zabezpečení jsou společné pro všechny technologie vysoké dostupnosti.

V případě zálohy a obnovy databází jsou doporučeny následující druhy bezpečnosti:

- Vytvářet více kopií záloh - jednu zálohu uložit na SQL Server a druhou na SAN.
- Fyzická ochrana – např. SAN, kam se ukládají zálohy databází přesunout na jiné místo než je SQL Server.
- Ochrana záloh a médií heslem.
- Obnovení záloh z důvěryhodných zdrojů – na SQL Serveru lze zakázat obnovení databází z neznámých a nedůvěřivých zdrojů.

V tab. 1.1 jsou uvedeny klady a zápory použití této technologie vysoké dostupnosti.

Tab. 1.1: Výhody a nevýhody použití technologie zálohování a obnovy databáze

Výhody	**Nevýhody**
+ jednoduchost	- dlouhá doba obnovy systému po havárii
+ jiné funkce než "jenom" zálohování např. ověření integrity databáze	- možnost určité ztráty dat v případě výpadku
	- částečná automatizace, v případě výpadku nutný zásah administrátora

1.1.4. Optimální varianta pro využití zálohy a obnovy databází v praxi

Záloha a obnova databází se většinou používá v prostředí malých, středních firem a u jednotlivců, tedy v prostředí, kde finanční stránka má větší prioritu než bezpečnost prostředí. Zálohování a obnova databází je obsaženo ve všech edicích SQL Serveru a oproti jiným technologií vysoké dostupnosti, je možné ji použít i v režimu obnovení SIMPLE. Což je režim, kdy se nezapisují všechny operace nad databází do transakčního protokolu. Tato technologie je vhodná i pro použití archivace, kdy v případě chyby v databázi, lze obnovit poslední zálohovanou databázi a vrátit se tak do funkčního stavu.

Pro automatizaci a jednoduchost, je vhodné použít zálohování databází pomocí plánu údržby, který je naplánován tak, aby se spouštěl každý den ve večerních hodinách, kdy je nejmenší provoz. Před samotným zálohováním je vhodné přidat následující funkce plánu údržby:

- Kontrola databázové integrity - zkontroluje integritu zvolených databází, a pokud zjistí, že některá z databází je poškozená, neprovede její zálohu. Nedoporučuje se používat kontrolu integrity dat u velkých kapacit databází.
- Smazat historii záloh – v případě omezené diskové kapacity na databázovém serveru, tato funkce umožňuje automaticky mazat zálohy starší než zvolený čas.

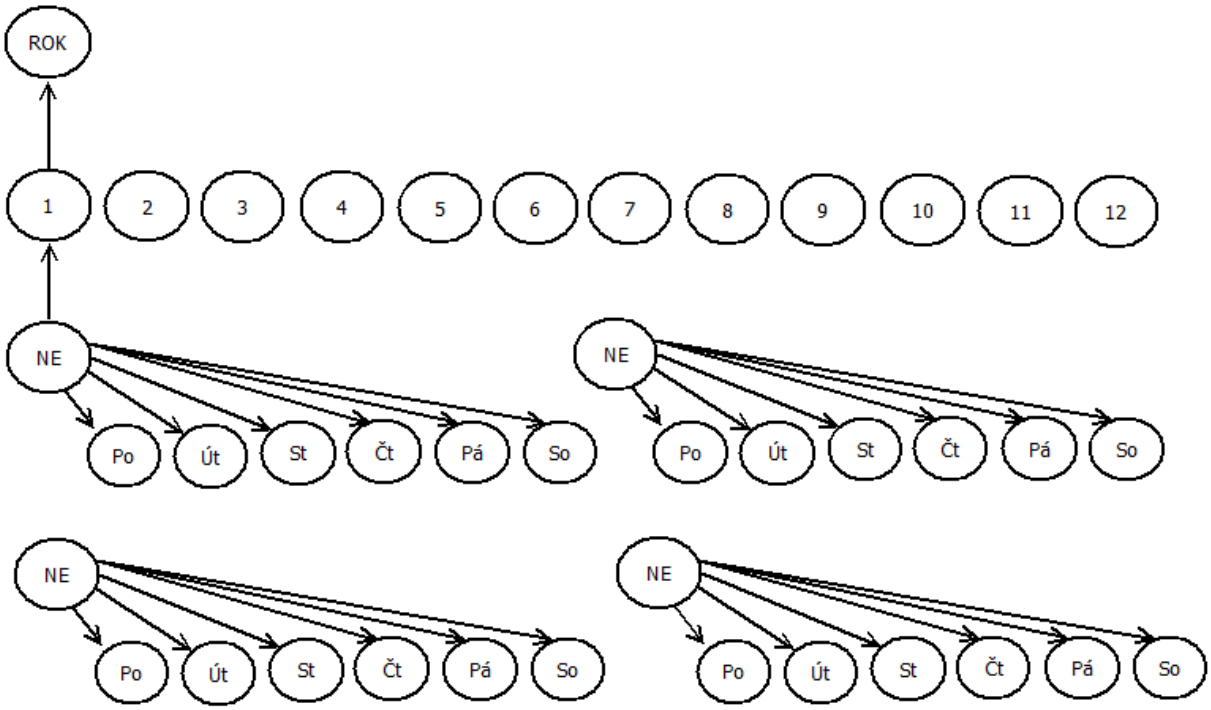
Po vytvoření plánu údržby, je možné nastavit notifikaci emailem v případě neprovedení zálohy. Notifikace se nastavuje nad SQL jobem, který plán údržby po dokončení průvodce automaticky vytvoří.

Pro zajištění lepší bezpečnosti, je doporučené zálohy kopírovat na jiný server či NAS. Před kopírováním je z důvodu ušetření kapacit a zrychlení přenosu po síti, lepší zálohy zabalit pomocí komprimačních programů. Všechny tyto operace lze zautomatizovat pomocí dávkového souboru, znázorněném ve výpis kódu 1.1, který se spouští pomocí naplánované úlohy vytvořené ve Windows prostředí.

Výpis kódu 1.2: Komprimace databáze a kopírování souboru.

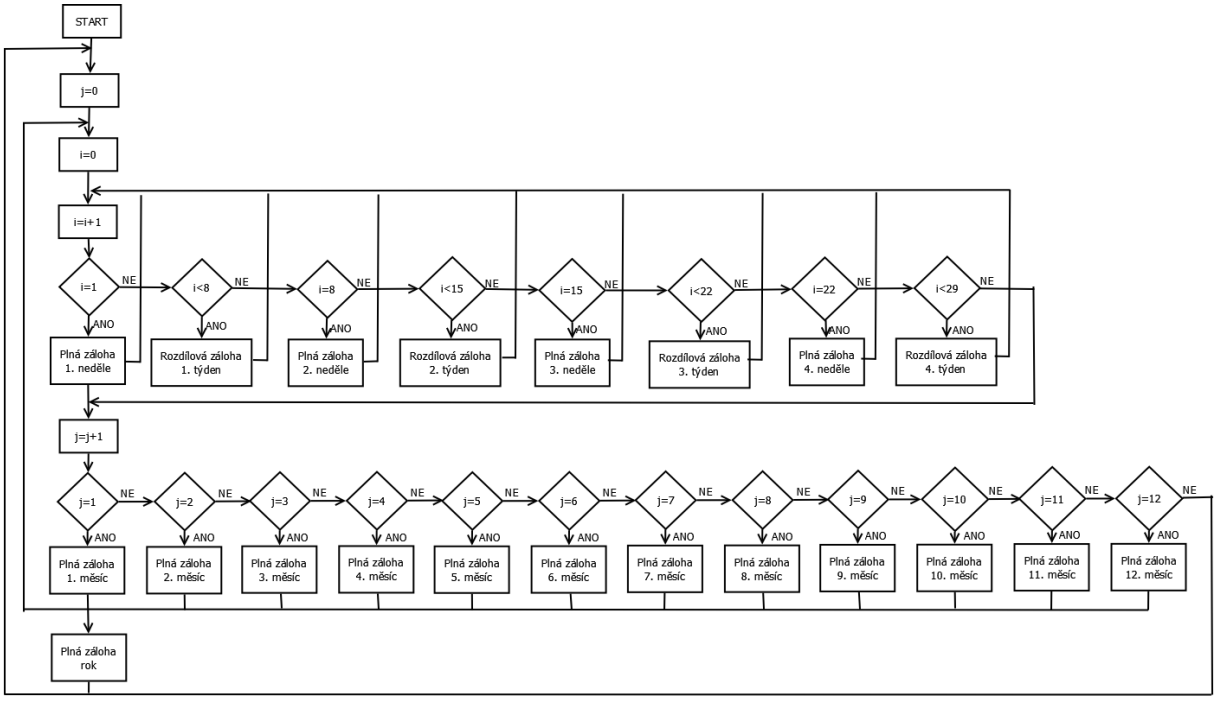
```
"c:\Program Files (x86)\7-Zip\7z.exe" a -tzip "d:\Backup\backupdb.zip" "d:\SQLBackup"
ECHO zabalení souborů
XCOPY d:\Backup\backupdb.zip \\diskstation\SHARED\Daily\db\
ECHO kopírování souborů na SAN
```

Administrátor si pak dle diskové kapacity NAS či délky archivace volí rotaci úložiště. Na obr. 1.7 je graficky znázorněn příklad rotace uložišť.



Obr. 1.7: GFS rotační schéma

Obr. 1.7 znázorňuje roční rotaci uložišť. Zvolená rotace uložišť má tři úrovně: rok, měsíc (1 až 12) a týden. Pro roční úroveň se na vzdáleném uložišti vytvoří jedna složka, pro měsíční 12 složek a pro týdenní 4 složky. Na začátku zálohování se přenese úplná záloha do 1. týdenní, 1. měsíční a roční složky. S postupujícími dny se vyplňují složky od nejnižší úrovně do nejvyšší, tedy od týdnů až po rok. V případě nového cyklu se zálohy nahrazují novějšími, čímž docílíme ušetření kapacity diskového uložiště. Pro lepší pochopení systému zálohování je na obr. 1.8 uveden vývojový diagram.



Obr. 1.8: Vývojový diagram použité rotace uložišť.

1.2. Záloha a obnova serveru

Nejedená se o technologii vysoké dostupnosti v prostředí Microsoft SQL, ale pomocí této technologie je možné zajistit vysokou dostupnost databázového serveru. Kombinací se zálohováním databází, tak vzniká ucelená část vysoké dostupnosti, která se v praxi hojně využívá, kde se zálohují jak databáze, tak server samotný.

1.2.1. Typy zálohování a obnovy serveru

Trendem dnešní doby je virtualizovat, což znamená, že pod jedním fyzickým serverem může současně existovat více virtuálních serverů, počítačů (VM). Vytvářet a spravovat virtuální servery je možné pomocí tzv. hypervizorů. Nejznámější hypervizory na trhu jsou, Vmware od stejnojmenné firmy Vmware a Hyper-V od firmy Microsoft. Jelikož hypervizor Hyper-V je vestavěnou funkcí Windows, zaměříme se tedy na něj.

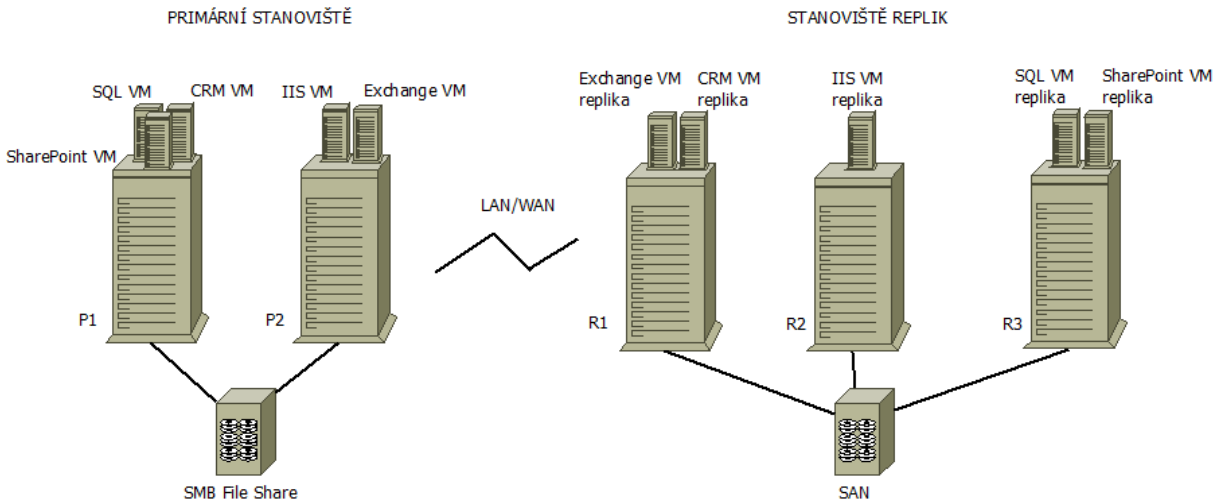
Hyper-V umožňuje exportovat nebo importovat virtuální počítače, čímž lze docílit požadované vysoké dostupnosti. Stejně jako u záloh databází, lze virtuální počítače umístit na stejná zálohovací média a použít podobné rotace úložišť.

V případě, že nevyužíváte virtuální prostředí, je možné provést zálohu operačního systému serveru i pomocí jiných nástrojů, jako jsou stínové kopie, Windows zálohovací nástroje, Wbadmin a další.

1.2.2. Hyper-V funkce určené pro zálohu a obnovu serverů

Hyper-V replika

Hyper-V replika je vestavěnou funkcí hypervizoru Hyper-V od verze 2012. Umožňuje asynchronně replikovat vybrané VM z Hyper-V primárního serveru, přes LAN/WAN prostředí na Hyper-V replikované servery. Na obr. 1.9 je znázorněn příklad použití Hyper-V repliky.



Obr. 1.9: Příklad replikace virtuálních počítačů.

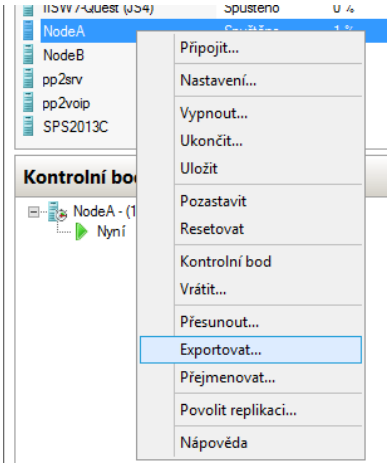
Hyper-V replika sleduje operace zápisu na primárním virtuálním počítači a pak replikuje změny přes prostředí LAN/WAN na replikované prostředí. Síťové připojení mezi server je realizováno pomocí http

protokolu, podporuje ověřování Kerberos a ověřování na základě certifikátu, s volitelnou podporou šifrování.

Hyper-V replika je úzce propojena s failover clustering ve Windows Server prostředí a poskytuje téměř bezproblémovou replikaci napříč různými scénáři pro migraci na primární a replikované servery. To umožňuje uložení virtuálních disků v různém umístění, což je výhodné v případě havárie.

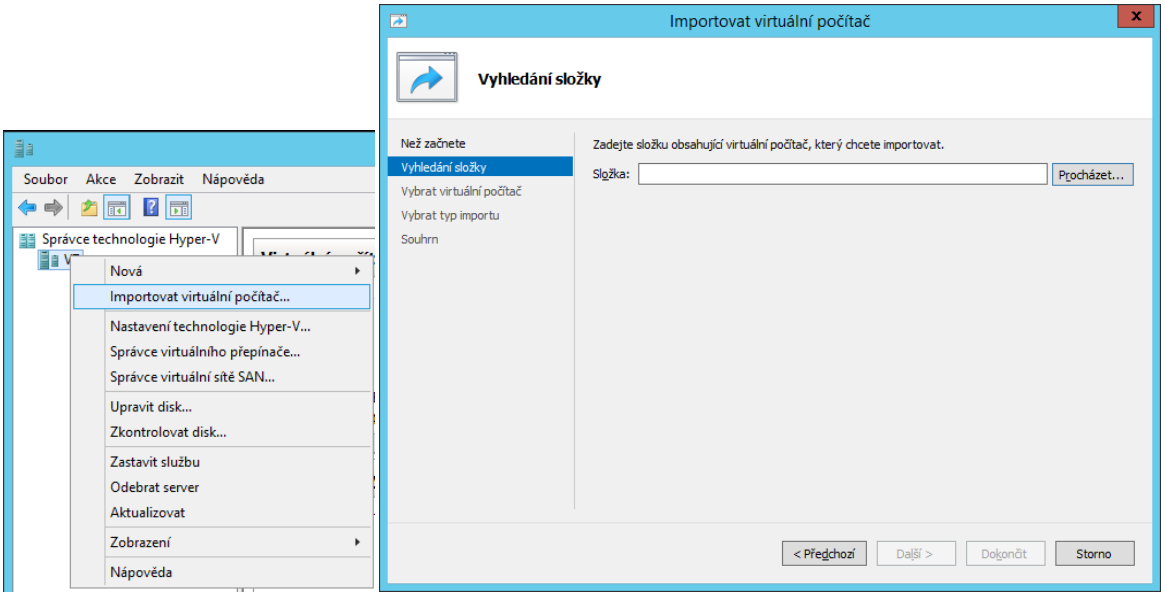
Export, import virtuálních počítačů

Pomocí správce technologie Hyper-V lze jednoduše vyexportovat virtuální počítač na určené místo. Exportovat lze ručně viz obr. 1.10 nebo pomocí PowerShellu[3].



Obr. 1.10: Ukázka exportu virtuálního počítače ve správci technologie Hyper-V.

Stejně jako export, lze i import provést buď pomocí správce technologie Hyper-V viz obr. 1.11 nebo pomocí PowerShellu.



Obr. 1.11: Ukázka importu virtuálního počítače ve správci technologie Hyper-V.

[3]Powershell – skriptovací jazyk v prostředí Windows určený zejména pro správu systému [7].

1.2.3. Bezpečnost, výhody a nevýhody použití zálohování a obnovy serverů v prostředí Hyper-V

V rámci bezpečnosti, zde platí stejná pravidla jako v kapitole 1.1.3, kde navíc, v případě migrace nebo replikace VM je možné nastavit ověřování pomocí Kerberos nebo certifikátu s podporou šifrování.

Shrnutí výhod použití zálohy a obnovy serverů v prostředí Hyper-V je uvedeno v tab. 1.2.

Tab. 1.2: Shrnutí výhod a nevýhod použití zálohy a obnovy serveru v prostředí Hyper-V.

Výhody	Nevýhody
+ jednoduchost, jednoduchá správa	- vhodné použití až od verze Hyper-V 2012
+ vestavěná funkce Windows Severu	- možnost určité ztráty dat v případě výpadku
+ možnost přiřadit systémové prostředky dle stavu (RAM, kapacita HDD)	- částečná automatizace, v případě výpadku nutný zásah administrátora

1.2.4. Optimální varianta využití technologií Hyper-V

Optimální varianta využít technologie Hyper-V, je sloučení funkcí Hyper-V replika a export, import virtuálních počítačů, kde dojde jak k rychlému přepnutí v případě havárie (Hyper-V replika), tak k vrácení se k funkčnímu stavu v případě ztráty a poškození dat (import, export VM).

1.3. Zrcadlení databáze

Zrcadlení databáze je především softwarové řešení pro zvýšení dostupnosti databáze. Zrcadlení je možné implementovat pouze pro uživatelské databáze (nikoliv systémové), které používají úplný model obnovení (Full recovery mode) tzv. všechny operace se zaznamenávají do transakčního protokolu. Podpora technologie zrcadlení databáze je pouze v edicích SQL Serveru Standard a Enterprise [2].

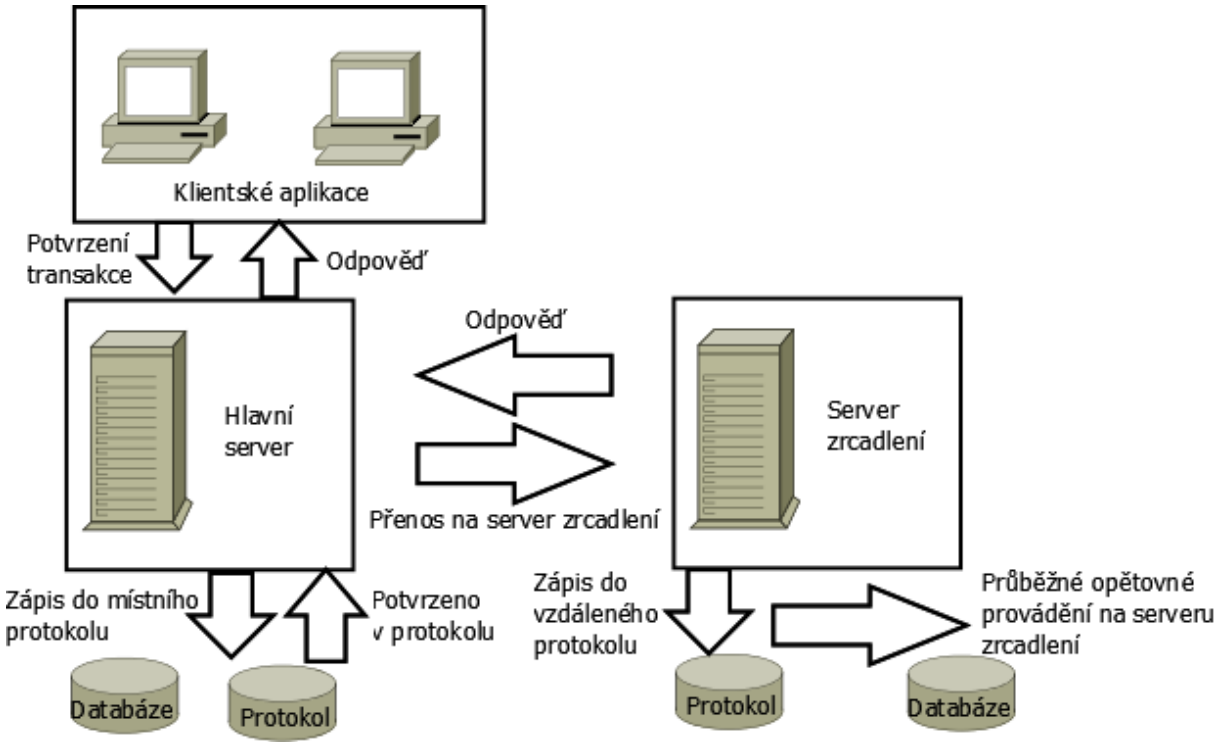
1.3.1. Princip, typy, stavy a provozní režimy zrcadlení databáze

Hlavní myšlenkou zrcadlení databáze je udržovat synchronizované databáze na hlavním a zrcadleném serveru. V případě selhání hlavní databáze nebo serveru, na kterém hlavní databáze běží, se klientské aplikace automaticky přepnou na zrcadlenou databázi a operace budou nadále pokračovat [1].

Tedy, všechny klientské aplikace jsou připojeny k hlavnímu serveru, databázi a odesílají transakce. Hlavní server zapíše příchozí změny do transakčního protokolu a automaticky přepošle změny na zrcadlo, které také zapíše stejné změny do svého transakčního protokolu. Zrcadlo pak odešle hlavnímu serveru potvrzení o přijetí dat [1].

Datový proud mezi hlavním a zrcadleným serverem je komprimován v poměru 12,5%. Systém SQL Server asynchronně přidá záznamy do protokolu zrcadla a současně zpracovává záznamy z protokolu na disku [1].

Na obr. 1.12 je znázorněna architektura zrcadlení databáze s popisem funkčnosti této technologie [1].



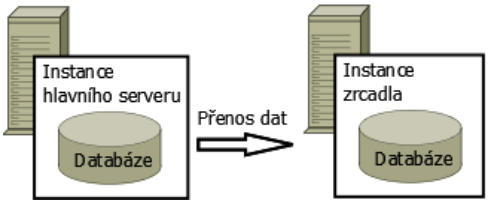
Obr. 1.12: Architektura zrcadlení databáze.

Zrcadlení databáze umožňuje následující typy a provozní režimy zrcadlení databáze [2]:

- Asynchronní zrcadlení databáze
  - Režim vysoký výkon
- Synchronní zrcadlení databáze
  - Režim vysoké ochrany bez automatického předání služeb při selhání
  - Režim vysoké ochrany s automatickým předáním služeb při selhání

Asynchronní zrcadlení databáze

Pokud transakce bezpečnosti je nastavena na „vypnuto“ (safety OFF), zrcadlení databáze pracuje asynchronně. Asynchronní zrcadlení databáze podporuje pouze jeden provozní režim – vysoký výkon (high-performance mode). Hlavní databáze před provedením transakce nečeká na potvrzení o přijetí od zrcadla, čímž se maximalizuje výkon, ale v případě ztráty synchronizace nemáte záruku, že transakce na zrcadle byla skutečně provedena. V případě selhání hlavního serveru nebo databáze, administrátor musí ručně přepnout službu na zrcadlo. Asynchronní zrcadlení databáze se většinou používá v síti s nízkou propustností, mezi zrcadlem a hlavní databází. Obr. 1.13 znázorňuje konfiguraci relace v režimu vysoký výkon [2].



Obr. 1.13: Konfigurace relace v režimu vysoký výkon.

V režimu vysoký výkon je možné použít i třetí instanci serveru, známou jako svědek, která slouží jako monitorovací server, ale pro asynchronní zrcadlení databáze se silně nedoporučuje.

Synchronní zrcadlení databáze

Pokud je transakce bezpečnosti nastavena na „FULL“, zrcadlení databáze je spuštěno v režimu vysoké ochrany, a po počáteční synchronizaci pracuje synchronně. V režimu vysoké ochrany se při spuštění relace synchronizuje zrcadlo s hlavní databází tak rychle, jak jen je to možné. Jakmile jsou databáze synchronizovány, obě databáze se zavazují, že budou čekat na potvrzení o přijetí dat, za cenu zvýšeného transakčního zpoždění [2].

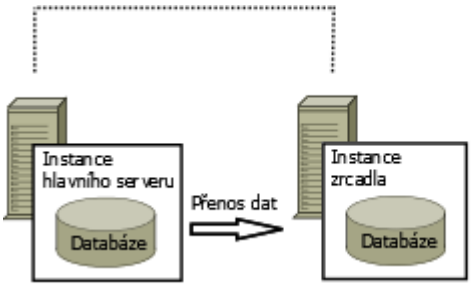
Synchronní operace jsou udržovány následovně [2]:

1. Hlavní server obdrží transakci od klienta, kterou zapíše do svého transakčního protokolu.
2. Při zápisu do transakčního protokolu, hlavní server současně odesílá transakce přijaté od klienta na zrcadlo a čeká na potvrzení.
3. Zrcadlo zapíše transakce do svého transakčního protokolu a vrací potvrzení na hlavní server.
4. Po obdržení potvrzení ze zrcadla, hlavní server odešle klientovy zprávu s potvrzením.

Při provozu synchronní zrcadlení databáze se používá myšlenka kvóra. Kvórum je nejnižší počet účastníků potřebných k rozhodnutí, co provést se zrcadlenou sadou. Účastníci mohou být hlavní databáze a zrcadlo nebo hlavní databáze, svědek (monitor) a zrcadlo. Pokud se v jakémkoliv bodě ztratí účastník, např. hlavní server, další dva účastníci stanoví kvórum a rozhodnou, co se stane. Každý účastník v kvóru dostane hlas a na konci svědek rozhodne jakékoli spory mezi hlavní databází a zrcadlem [1].

Režim vysoké ochrany bez automatického předání služeb při selhání

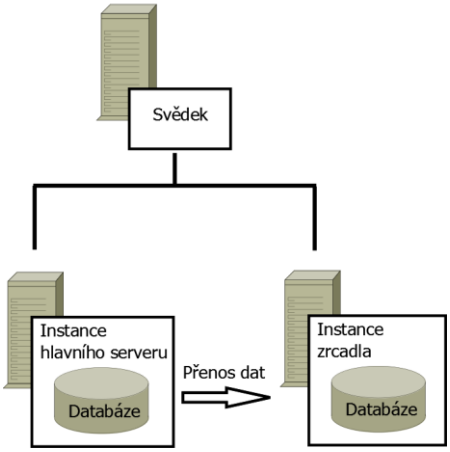
Stejně jako v režimu vysoký výkon, v případě selhání hlavního serveru nebo databáze, administrátor musí ručně přepnout službu na zrcadlo. Naopak, pokud je zrcadlo nedostupné, hlavní server se odpojí a dále funguje bez zrcadlení. Obr. 1.14 znázorňuje režim vysoké ochrany bez automatického předání služeb při selhání [2].



Obr. 1.14: Konfigurace v režimu vysoké ochrany bez automatického předání služby při selhání.

Režim vysoké ochrany s automatickým předání služeb při selhání

V režimu vysoké ochrany s automatickým předání služeb při selhání se vyžaduje třetí instance serveru, známá jako svědek nebo monitor (witness). Na rozdíl od hlavní databáze a zrcadla, svědek neprovádí žádné operace nad databází. Jeho hlavní funkcí je umožnit automatické přepnutí služeb v případě selhání. V podstatě sleduje operace hlavní databáze a zrcadla, a pokud hlavní databáze neodpoví během stanoveného časového limitu, provede přepnutí na zrcadlo. V případě, že selže zrcadlo a hlavní databáze je stále v kontaktu se svědkem, hlavní databáze může nadále fungovat. Pokud svědek detekuje, že je zrcadlo opět funkční, dá příkaz zrcadlu, aby se opět synchronizovalo s hlavní databází. Obr. 1.15 znázorňuje režim vysoké ochrany s automatickým předáním služeb při selhání [2].



Obr. 1.15: Konfigurace v režimu vysoké bezpečnosti s automatickým předáním služby při selhání.

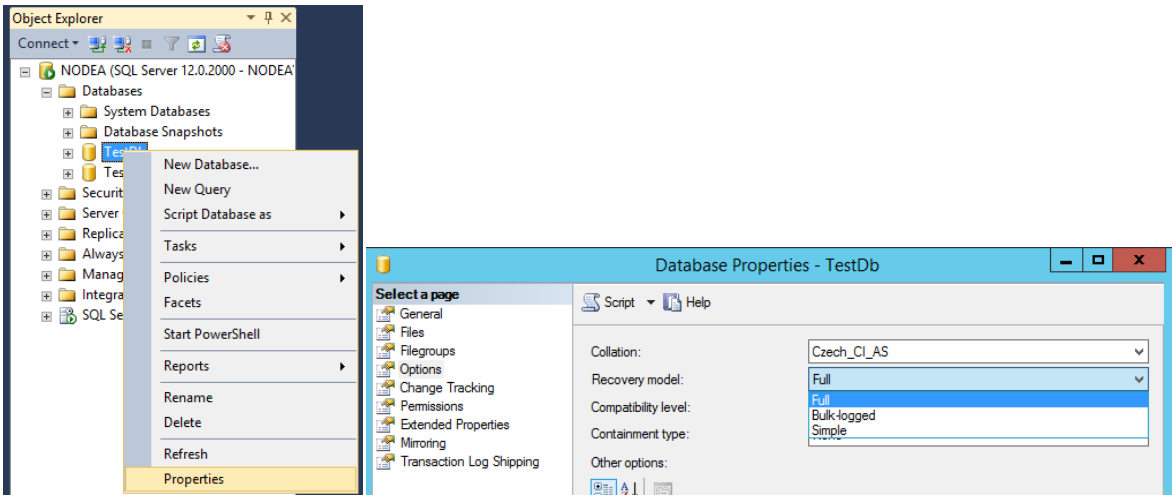
Při zrcadlení databáze systém prochází sadou stavů databáze. Tyto stavy indikují stav vašeho zrcadla. Tab. 1.3 obsahuje seznam stavů databází, které jsou součástí zrcadlení databáze [1].

Tab. 1.3: Stavy zrcadlení databáze.

Stav	Popis stavu
SYNCHRONIZED	Hlavní databáze i zrcadlo jsou synchronizovány. Hlavní databáze posílá veškeré změny k aplikaci zrcadlu a zrcadlo nezaostává. V tomto režimu je možné automatické i ruční přepnutí v případě poruchy.
SUSPENDED	Zrcadlení nadále neprobíhá, buď kvůli zásahu správce, nebo kvůli chybám při aplikaci změn na zrcadle. Hlavní databáze je považována za vystavenou riziku, což znamená, že hlavní databáze se provozuje bez účastníka na druhé straně zrcadla.
PENDING_FAILOVER	Správce provedl ruční přepnutí v případě poruchy, ale zrcadlo toto přepnutí ještě nepřijalo. Tento stav se vyskytuje jen na hlavní databázi.
DISCONNECTED	Účastník ztratil spojení s druhým účastníkem a svědkem, pokud byl svědek přítomen.

1.3.2. Nasazení zrcadlení databáze

Jelikož je technologie zrcadlení databáze postavena na principu posílání transakcí mezi hlavním serverem a zrcadlem, je nutné zapnout pro danou databázi Full recovery mód viz obr. 1.16.



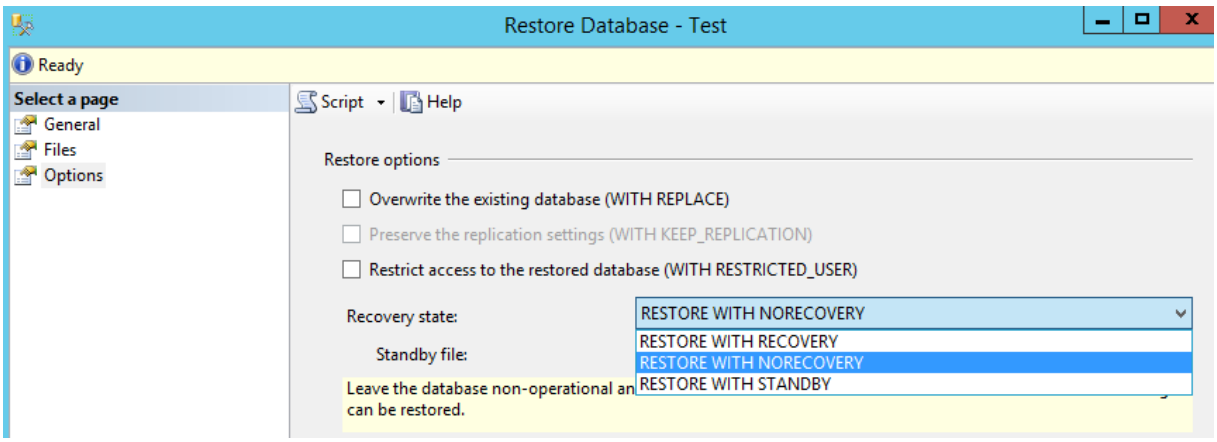
Obr. 1.16: Nastavení full recovery módu.

Full recovery mód zapisuje všechny operace nad databází do transakčního protokolu. Čímž výrazně narůstá kapacita transakčního protokolu, a proto není vhodné zapínat tento mód, pokud není dostatečná kapacita diskového prostoru.

Nasazení technologie zrcadlení databáze je možné buď pomocí skriptu nebo pomocí grafického prostředí SQL Server Management Studia (SSMS). Pro lepší orientaci je vhodnější použít grafické prostředí.

Postup nasazení zrcadlení databáze v prostředí SSMS:

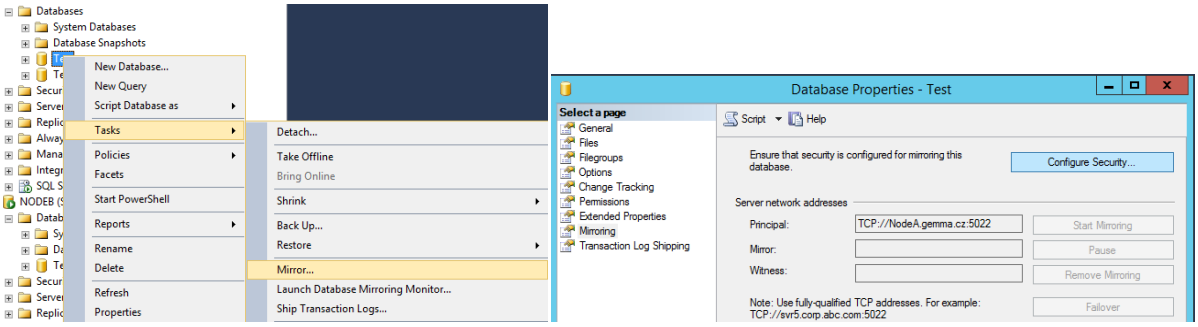
1. Vytvoření plné zálohy databáze, nad kterou chceme použít technologii zrcadlení databáze.
2. Vytvoření zálohy transakčního protokolu databáze.
3. Přenesení záloh databází na zrcadlený server.
4. Obnova plné zálohy databáze na zrcadleném serveru v NORECOVERY režimu viz obr. 1.17.



Obr. 1.17: Obnovení databáze v NORECOVERY režimu.

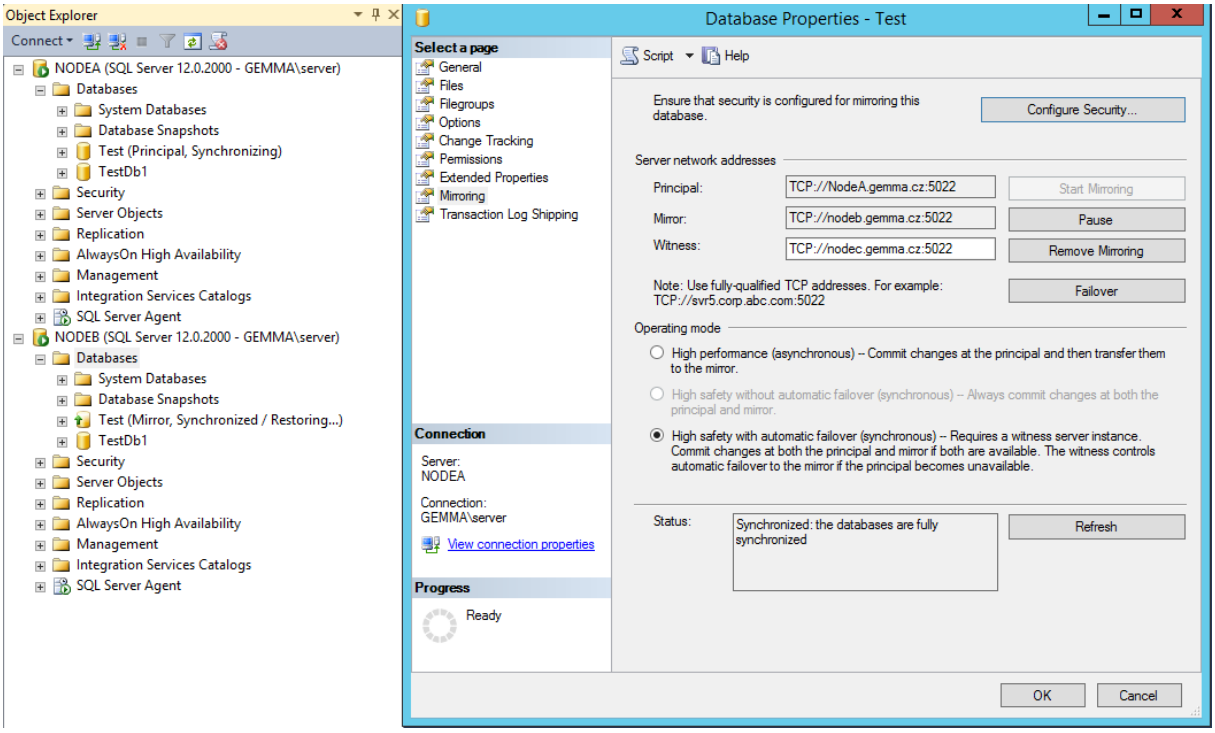
5. Obnova zálohy transakčního protokolu na zrcadleném serveru, také v NORECOVERY režimu.

6. Nastavení zrcadlení databáze na hlavním serveru viz obr. 1.18.



Obr. 1.18: Postup k nastavení zrcadlení databáze.

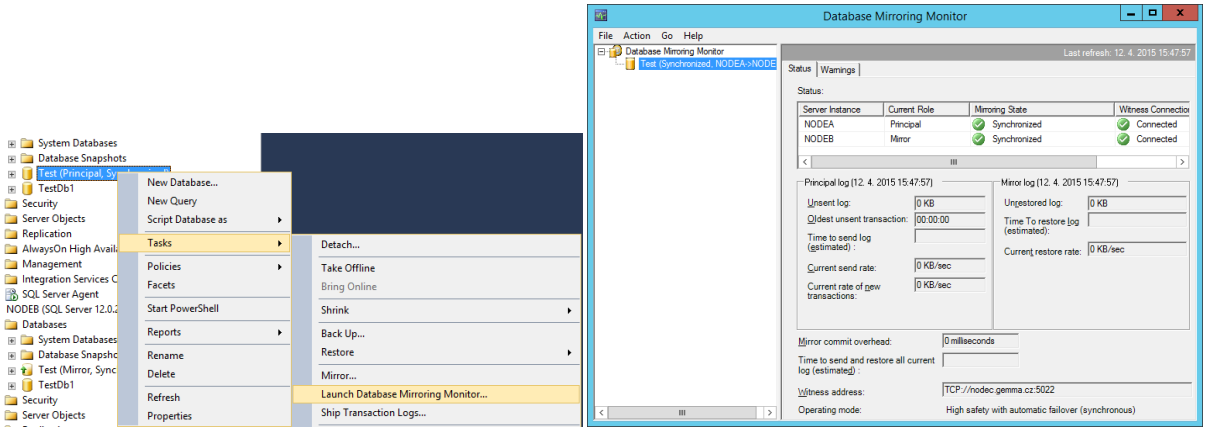
V sekci zrcadlení (Mirroring), zvolte možnost konfigurace (ConfigureSecurity…), kde pomocí průvodce nastavíte zrcadlení databáze. V prvním kroku vyberete, zda použijete technologie se svědkem nebo bez svědka. Dále konfigurujete hlavní instanci serveru a následně zrcadlenou instanci či instanci pro svědka. Pro všechny instance musí existovat účet společný pro všechny instance, který se vyplňuje v dalším kroku. Pokud jste všechno správně nastavili, můžete spustit zrcadlení databáze.



Obr. 1.19: Nastavení zrcadlení databáze.

Obr. 1.19 zobrazuje nastavení zrcadlení databáze se svědkovou instancí. Na obrázku je také patrné, jakým způsobem lze přepínat mezi jednotlivými provozními režimy. Pomocí tlačítka Failover lze jednoduše přepnout hlavní databázi na zrcadlený server a naopak.

Pro ověření synchronizace mezi hlavním a zrcadlovém serveru je možné využít nástroj Database Mirroring Monitor, kde lze přehledně viz obr. 1.20 zjistit, zda jsou databáze synchronizované.



Obr. 1.20: Monitor zrcadlení databáze.

1.3.3. Bezpečnost, výhody a nevýhody použití zrcadlení databáze

Mimo nastavení bezpečnosti popsané v kapitole 1.1.3, lze při konfiguraci zrcadlení databází zvolit možnost šifrování dat mezi instancemi. Šifrování dat mezi instancemi je založeno na bázi certifikátu.

Shrnutí kladů a záporů použití zrcadlení databáze ve Vašem prostředí je shrnuto v tab. 1.4.

Tab. 1.4: Výhody a nevýhody použití zrcadlení databáze.

Výhody	Nevýhody
+ automatické přepnutí v případě havárie při použití svědka	- relativně krátká doba obnovy systému po havárii
+ jednoduchost	- možnost určité ztráty dat v případě výpadku
	- částečná automatizace, v případě výpadku nutný zásah administrátora

1.3.4. Optimální varianta využití technologie zrcadlení databáze

Optimální varianta využití technologie zrcadlení databáze je při použití provozního režimu vysoké ochrany s automatickým předáním služeb při selhání, kde svědek monitoruje situaci a v případě selhání systému automaticky přepne databázi na zrcadlený server. Není tedy nutná přítomnost administrátora. Toto řešení však vyžaduje více instancí SQL Serveru, čímž se stává finančně náročnější. Tato technologie se většinou využívá tam, kde je potřeba rychlá obnova dat v případě havárie.

Poslední verze MS SQL Serveru, která technologii zrcadlení databáze používá, je verze SQL Serveru 2014. V následujících verzích Microsoft SQL Serveru již technologie zrcadlení databáze nebude součástí řešení vysoké dostupnosti.

1.4. Replikace

SQL Server replikace je sada technologií pro kopírování, distribucí dat a databázových objektů z jedné databáze do jiné a následné synchronizaci mezi databázemi k udržení konzistence dat. Je vhodné rozdělit replikace do dvou obecných kategorií:

- Replikace dat v prostředí server-server.
- Replikace dat v prostředí server-klient.

Replikace dat mezi servery obvykle podporuje zlepšení škálovatelnosti a dostupnosti, datové sklady a vytváření sestav, a integraci dat z více míst.

Replikace dat v prostředí server-klient obvykle podporuje výměnu dat s mobilními uživateli, spotřebitelské POS [4]aplikace a integraci dat z více míst [3].

Existují tři typy replikací:

- Transakční replikace.
- Slučovací replikace.
- Snímková replikace.

Každý typ replikace je vhodný pro odlišné scénáře. Mezi hlavní kritéria pro volbu jednotlivých možností patří frekvence změn, množství změn, velikost publikace, zdroj změny dat, ale také zpoždění mezi jednotlivými servery, dostupná konektivita atd. [3].

1.4.1. Replikace dat v prostředí server-klient, server-server a typy replikací

Replikace dat v prostředí server-klient

Jak už je z názvu patrné, replikace probíhá mezi servery a klienty, včetně pracovních stanic, notebooků, tabletu a dalších koncových zařízení. Data jsou obvykle replikována mezi servery a klienty pro podporu těchto aplikací [3]:

- Výměna dat s mobilními uživateli.
  Mnoho aplikací vyžaduje, aby údaje byly k dispozici pro vzdálené uživatele jako např. prodejcům, řidičů dodávek atd. Tyto aplikace zahrnují řízení vztahu se zákazníky (CRM), automatizace prodejní síly (SFA) a pole automatizací (FFA) v reálném čase.
- Spotřebitelské POS aplikace.
  POS aplikace, jako jsou například pokladní terminály a bankomaty vyžadují údaje, které jsou replikovány ze vzdálených míst do centrálního místa.
- Integrace dat z více serverů.
  Například aplikace pro regionální kanceláře mohou vyžadovat údaje v jednom nebo obou směrech mezi krajskými úřady a ústředními kancelářemi.

Replikace dat mezi servery

Data jsou obvykle replikovány mezi servery pro podporu následujících aplikací a požadavků [3]:

- Zlepšení škálovatelnosti a dostupnosti.

[4] Point of sale (POS) představují reklamní materiály a produkty využívané v místě prodeje pro propagaci konkrétního výrobku či určitého druhu sortimentu [8].

Udržování kontinuálně aktualizované kopie dat, umožňuje čtení aktivit mezi více servery. Redundance vyplývající ze zachování více kopií stejných dat je zásadní během plánované a neplánované údržby systému.

- Uskladnění dat a vytváření sestav.
  Datový sklad a reportování serveru často používá data z online zpracování transakcí (OLTP) serveru. Replikace umožňuje přenášet data mezi OLTP servery, reportování a podporuje systém pro rozhodování.
- Integrace dat z více serverů.
  Data ze vzdálených kanceláří jsou často shromažďovány a konsolidovány do centrálních kanceláří. Podobně data mohou být replikovány mezi vzdálenými pobočkami.
- Integrace heterogenních dat.
  Některé aplikace jsou závislé na údajích zasílaných do nebo z jiných ne-SQL Server databází. Pomocí replikace je možné integrovat data z ne-SQL Server databází.
- Snižování zátěže dávkového zpracování.
  Dávkové operace jsou někdy příliš náročné pro běh na OLTP serverech. Replikace pomocí snižování zátěže zpracování provádí operace na jiném vyhrazeném serveru.

Replikace snímků

Replikace snímků je známá také pod názvem Snapshot replikace. Snímková replikace distribuuje data přesně tak, jak se zobrazují v konkrétním čase, a nesleduje aktualizaci dat. Dojde-li k synchronizaci, je celý snímek vygenerován a odeslán do odběratele (Subscriber). Funguje samostatně, ale nejčastěji se používá jako základ pro další dva druhy replikací transakční a slučovací. Každý snímek vzniká vždy znovu, je tedy náročný na systémové prostředky [3].

Snímková replikace je vhodná v následujících případech [3]:

- Zřídkakdy se mění data.
- Malé objemy dat replikace.
- Dochází k velkému množství změn v krátkém čase.

Slučovací replikace

Obvykle začíná snímkem databáze. Následné změny dat a úpravy schémat provedeny u vydavatele (Publisher) a odběratele (Subscriber), jsou sledovány pomocí aktivačních procedur (triggerů). Po připojení do sítě, se odběratel synchronizuje s vydavatelem a vymění všechny řádky změněné od poslední synchronizace. Slučovací replikace se obvykle používají v prostředí server-klient [3].

Použití slučovací replikace je vhodné v následujících situací [3]:

- Více odběratelů může aktualizovat stejná data v různých časech a šířit tyto změny na vydavatele nebo jiné odběratele.
- V práci offline, kdy po připojení do sítě dojde k synchronizaci (např. čtečky kódů).
- Každý odběratel vyžaduje jiný oddíl dat.
- Umožňuje pokročilou detekci konfliktů a jejich zpracování.

Slučovací replikace umožňuje pracovat samostatně a později sloučit aktualizace do jednoho jednotného výsledku. Vzhledem k tomu, že aktualizace se provádí na více než jeden uzel, stejné data

mohou být aktualizovány vydavatelem a více než jedním odběratelem. Při sloučení aktualizací proto může docházet ke konfliktům [3].

Transakční replikace

Stejně jako u slučovací replikace, obvykle začíná snímkem databáze. Jakmile je počáteční snímek přijat, následné změny dat a úpravy schématu provedené na vydavateli, jsou téměř v reálném čase dodávány na odběratele. Změny jsou u odběratele aplikovány ve stejném pořadí jako u vydavatele. Proto v rámci publikace je zaručena konzistence dat [3].

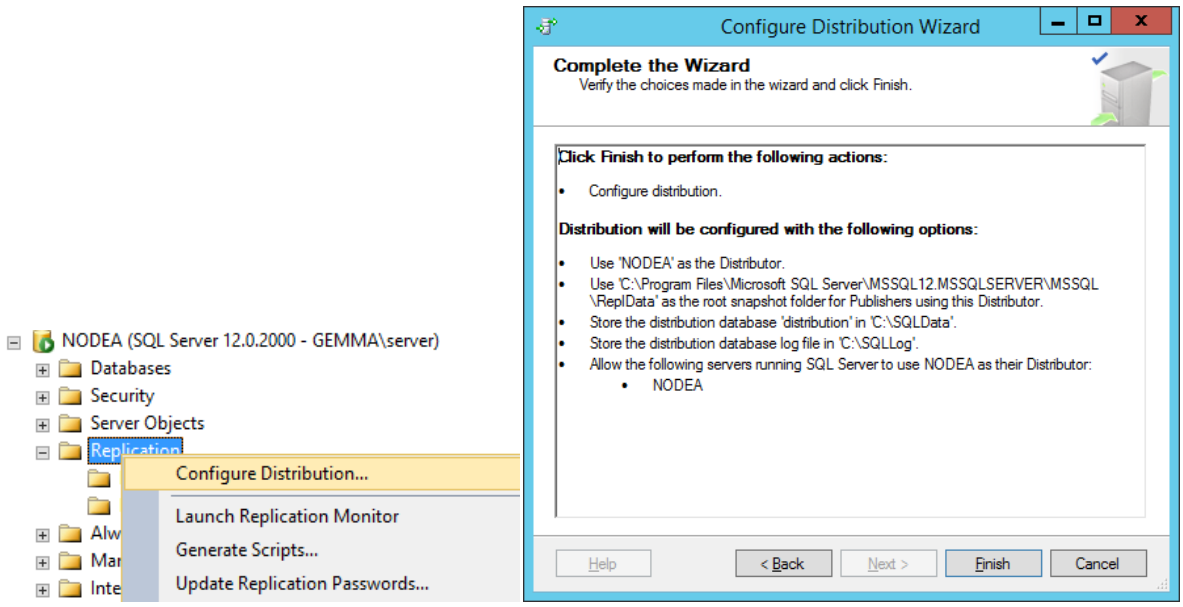
Transakční replikace je obvykle využívána v prostředí server-server a je doporučována v následujících případech [3]:

- Pokud chcete odesílat změny, jakmile nastanou.
- Aplikace vyžaduje nízkou časovou prodlevu mezi vydavatelem a odběratelem.
- Vydavatel používá velmi často příkazů vložit, aktualizovat či smazat (INSERT, UPDATE, DELETE).
- Vydavatel nebo odběratel používá ne-SQL Server databázi např. Oracle.
- Při replikaci velkého množství dat.

1.4.2. Nasazení replikací

Chcete-li nasadit do Vašeho prostředí jakoukoliv replikaci dat, prvně je nutné nastavit distributora. Distributor je server osahující distribuční databáze, na které se ukládají metadata a historie dat pro všechny typy replikací. Nastavit distributora lze pomocí SQL skriptu nebo v grafickém prostředí SQL Server Management Studia (SSMS).

Nastavení distributora pomocí SSMS je znázorněno na obr. 1.21, probíhá postupnými kroky za pomocí průvodce. Na obr. 1.21 vpravo je souhrn instalace distributora.

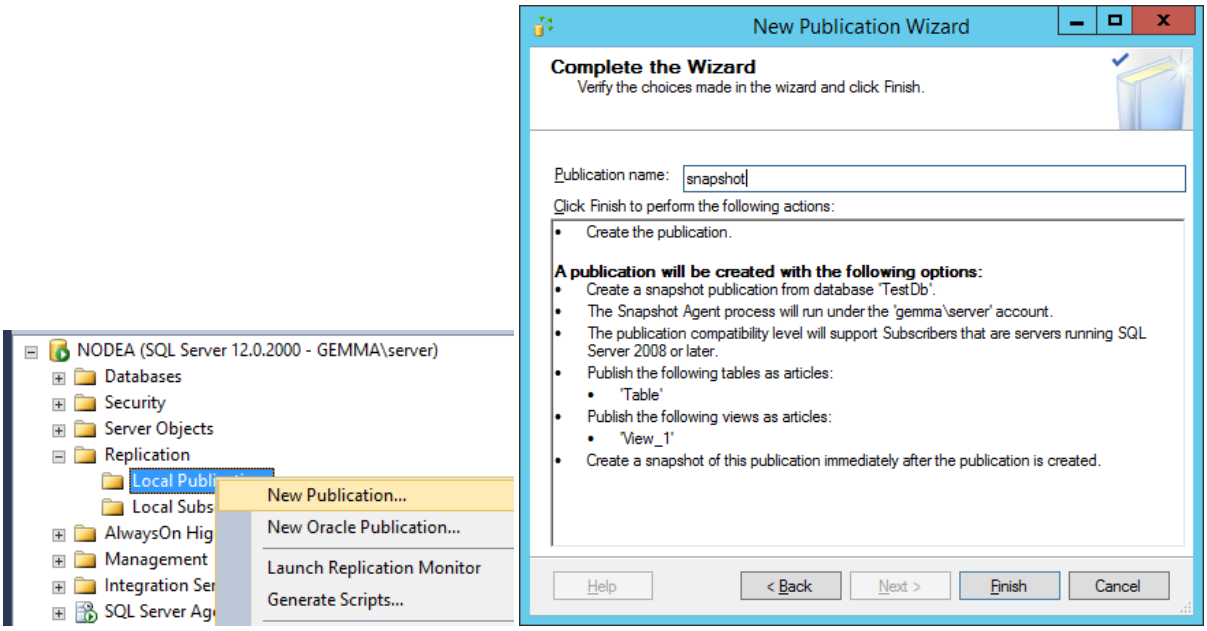


Obr. 1.21: Vytvoření a shrnutí konfigurace distributora.

V případě úspěšné instalace distributora, dalším krokem je vytvoření vydavatele. Každý vydavatel může být přiřazen pouze k jedné instanci distributora. Stejně jako u distributora, vydavatele lze vytvořit buď

pomocí SQL skriptu nebo pomocí průvodce v prostředí SSMS viz obr. 1.22. Průvodce je pro všechny typy replikací stejný a umožňuje volbu:

- Databáze, která se použije pro replikaci.
- Typ replikace.
- Úroveň kompatibility.
- Tabulky, pohledy (view) a funkce.
- Čas replikace: plánovaně nebo ihned.

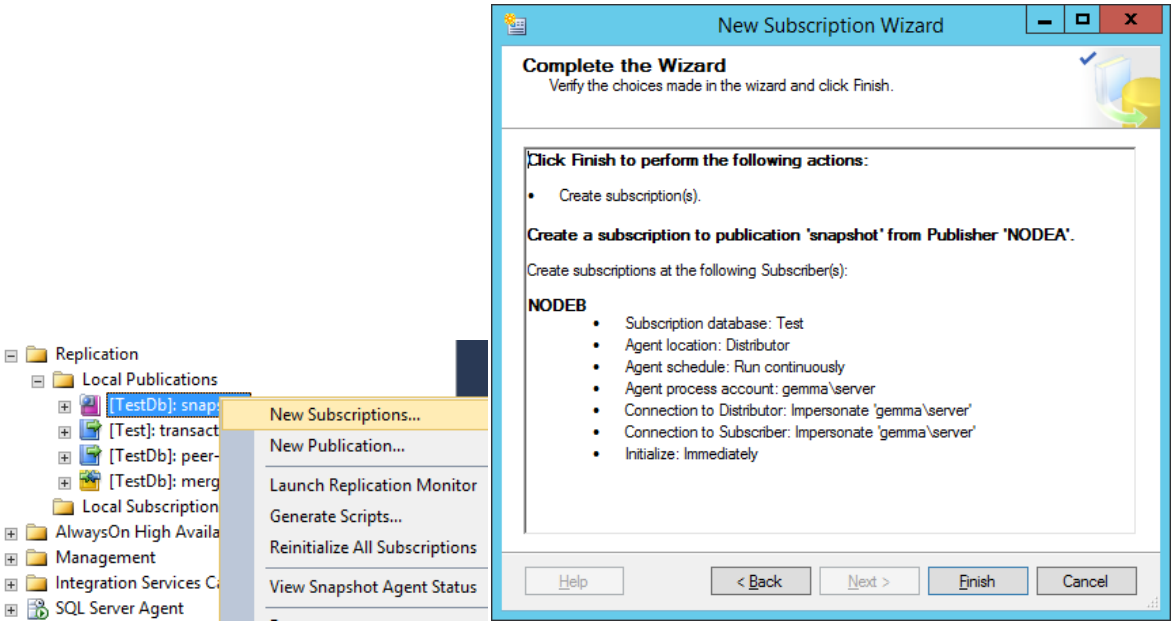


Obr. 1.22: Vytvoření a shrnutí konfigurace vydavatele.

Posledním krokem k nasazení replikací je vytvoření odběratele. Pro jednoho vydavatele může existovat více odběratelů. Vytvoříme jej buď pomocí SQL skriptu, nebo průvodce s následující možnostmi:

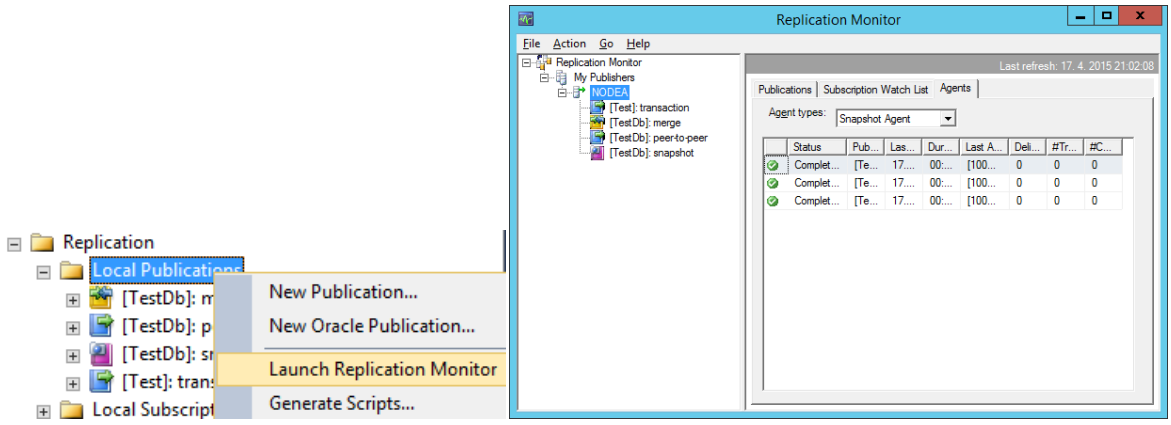
- Výběr instance odběratele a databáze
- Výběr účtu, pod kterým služba běží
- Čas provedení
- U slučovací replikace se navíc volí, zda odběratel je server nebo klient

Na obr. 1.23 je uveden postup k vytvoření odběratele a jeho následné shrnutí v průvodci instalace.



Obr. 1.23: Vytvoření a shrnutí konfigurace odběratele.

Pro monitorování stavu replikací a pro případné řešení problému nefunkčnosti replikací slouží nástroj monitor replikací. Postup ke spuštění monitoru replikací a ukázka prostředí je uvedena na obr. 1.24.



Obr. 1.24: Monitor replikací.

1.4.3. Bezpečnost, výhody a nevýhody použití replikací

Replikace přesouvá data v distribuovaných prostředí od intranetu na jedné doméně, přes aplikace přistupující k datům přes nedůvěryhodné domény nebo přes internet. V případě nedůvěryhodné domény a přístupu přes internet, je vhodné použít následující prostředky zabezpečení:

- Virtuální privátní sítě (VPN).
- Secure Socket Layer (SSL).
- IP Security (IPSec).

Další bezpečnostní rizika jsou na úrovni replikačních agentů, kde je doporučeno:

- Každý agent by měl běžet pod jiným účtem.

- Povolit jenom nezbytné oprávnění pro každého agenta.
- Všechny slučovací a distribuční účty agentů by měly být v PAL[5].
- Zajistit, aby všechny agenti v PAL, měly minimální potřebné oprávnění k plnění úkolů replikace.
- Zajištění minimálního přístupu ke sdílené složce, kde se ukládají snímky databází.

V tab. 1.5 jsou uvedeny výhody a nevýhody použití replikací dat v praxi.

Tab. 1.5: Výhody a nevýhody použití replikace.

Výhody	**Nevýhody**
+ škálovatelnost a dostupnost	- v případě havárie nutný zásah administrátora
+ rozložení zátěže	- relativní složitost nasazení technologie
+ integrace dat z více serverů	- bezpečnostní hrozba v případě použití nedůvěryhodného prostředí
+ možnost použít ne-SQL databázi	

1.4.4. Optimální varianta využití replikací

V případě replikací je obtížné vybrat, který typ je optimální, každá typ má jiné vlastnosti a lze jej využít ve mnoha scénářích.

Použití snímkové replikace je vhodné například v prostředí, kde spouštíme generování rozsáhlých sestav a nechceme zatížit produkční servery. Pomocí snímkové replikace, tak přeneseme po určitém časovém úseku produkční databáze na připravené servery, na kterých následně spustíme generaci sestav.

Naopak transakční replikace je vhodná v reálném prostředí, nejlépe v prostředí server-server, kde v případě výpadku jednoho ze serverů, aplikace mohou dále komunikovat s dalšími servery. Nebo v prostředí, kde potřebuje na všech serverech v jednom okamžiku stejná data, reálné prostředí.

Slučovací replikace je oproti transakční replikace vhodná v prostředí server-klient, kde jak klient, tak i server, mohou průběžně aktualizovat data v databázi. Optimální například pro systém bankomatů.

[5] Publication access lists (PAL) je hlavní mechanismus k zabezpečení publikace na vydavateli [9].

1.5. Odesílání souboru protokolu

Technologie odesílání souboru protokolu (Log Shipping), umožňuje automatické odesílání záloh transakčního protokolu z primární databáze na primárním serveru, na jednu nebo více sekundárních databází samostatné instance sekundárních serverů. Záloha transakčního protokolu je aplikována na každou sekundární databázi jednotlivě. Volitelná třetí instance serveru, známá jako monitorovací server, zaznamenává historii a stav operací zálohování a obnovení, a v případě selhání operace, upozorňuje na selhání. Jakmile je monitorovací server nakonfigurovaný, konfiguraci nelze změnit, pokud se první neodstraní odesílání souboru protokolu [3].

1.5.1. Termíny, definice a princip odesílání souboru protokolu

Primární server – instance SQL Serveru, na které běží váš produkční server.

Primární databáze – databáze na primárním serveru, kterou chcete zálohovat na jiné servery. Veškerá konfigurace protokolu se provádí z primární databáze.

Sekundární server – instance SQL Serveru, kde chcete udržovat kopie primární databáze.

Sekundární databáze – kopie primární databáze. Sekundární databáze může být buď v módu RECOVERING nebo STANDBY, které umožňují omezený přístup „pouze pro čtení“ [3].

Monitorovací server – volitelná instance serveru, která sleduje všechny detaily odesílání souboru protokolu, včetně [3]:

- Kdy naposledy byla provedena záloha primární databáze.
- Kdy naposledy proběhlo kopírování a obnova zálohy na sekundární servery.
- Informací o případných selháních.

Úloha zálohování (backup job) – SQL Server Agent úloha, která provádí operaci zálohování, zaznamenává historii na lokální a monitorovací server, odstraňuje staré zálohy a starou historii. Je-li zapnuto odesílání souboru protokolu, úloha zálohování se vytvoří na primárním serveru [3].

Úloha kopírování (copy job) – SQL Server Agent úloha, která zkopíruje záložní soubory z primárního serveru na sekundární servery a zapíše historii na sekundární a monitorovací server. Pokud je odesílání souboru protokolu povoleno na databázi, úloha se vytvoří na každém ze sekundárních serverů [3].

Úloha obnovy (restore job) – SQL Server Agent úloha, která obnoví zálohované soubory zkopírované na sekundárních databázích. Zaznamenává historii na lokálním a monitorovacím serveru, odstraní staré soubory a starou historii. Pokud je odesílání souboru protokolu povoleno na databázi, úloha se vytvoří na každém ze sekundárních serverů [3].

Úloha upozornění (job alert) – SQL Server Agent úloha, která upozorní, pokud při operaci zálohy či obnovy na primární nebo sekundární databázi dojde k chybě. Je-li zapnuto odesílání souboru protokolu, úloha upozornění se vytvoří na monitorovacím serveru [3].

Odesílání souboru protokolu se skládá ze tří operací [3]:

1. Záloha protokolu transakcí na primární instanci serveru.
2. Kopírování protokolu transakcí na sekundární instanci serveru.
3. Obnovení zálohy protokolu na sekundární instanci serveru.

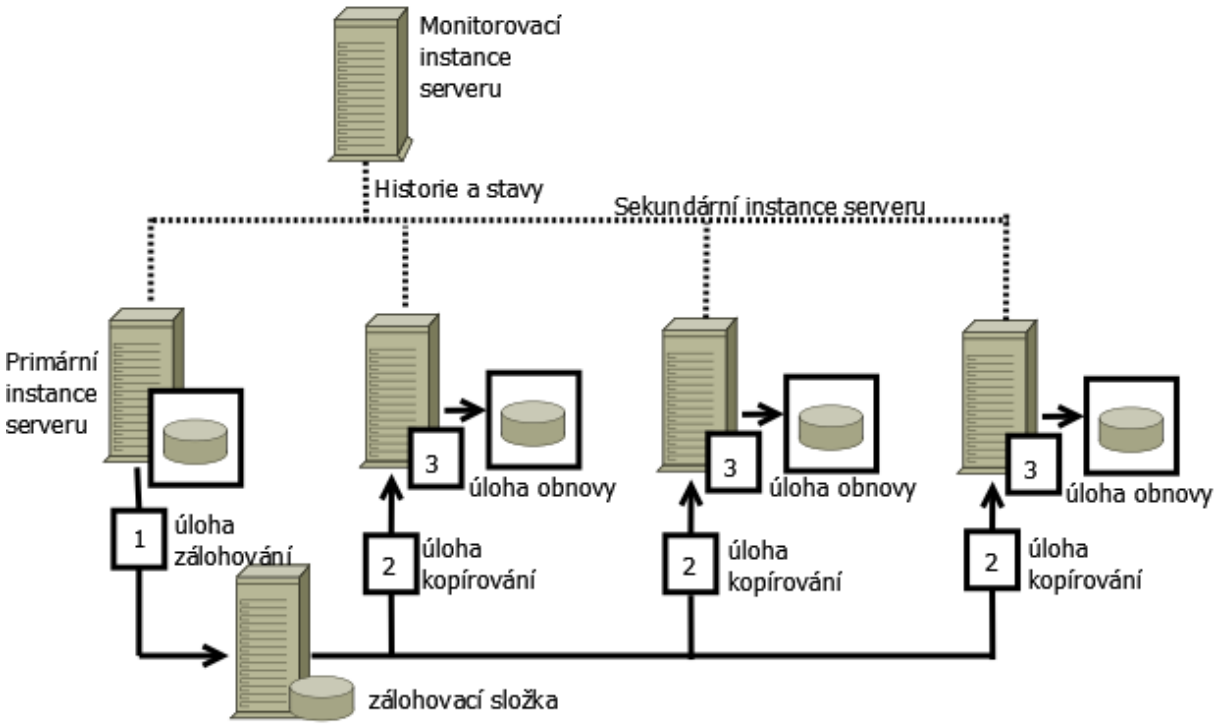
Transakční protokol může být směřován na více sekundárních instancí. V takovém případě, se krok 2 a 3 opakuje pro každou instanci zvlášť. Konfigurace odesílání souboru protokolu neobsahuje automatické přepnutí v případě selhání. Pokud se stane primární databáze nedostupnou, správce provede přepnutí na sekundární databázi ručně. Sekundární databáze může sloužit pro účely vykazování [3].

Typická konfigurace odesílání souboru protokolu

Obr. 1.25 ukazuje konfiguraci odesílání souboru protokolu s jednou primární instancí, třemi sekundárními instancemi a s monitorovací instancí serveru. Obr. 1.25 ilustruje následující kroky [3]:

1. Primární instance serveru spustí úlohu zálohování k provedení zálohy transakčního protokolu na primární databázi. Instance serveru umístí zálohy protokolu do primárního „log-backup“ souboru, který zasílá do zálohovací složky. V obrázku je zálohovací složka sdíleným adresářem.
2. Každá ze tří sekundárních instancí serveru spustí svoji kopírovací úlohu, která zkopíruje soubor „log-backup“ do své lokální složky.
3. Každá ze tří sekundárních instancí serveru spustí svojí úlohu obnovy, která obnoví „log-backup“ soubor z lokální složky do své databáze.

Primární i sekundární instance posílají svoji historii a stavy na monitorovací instanci.

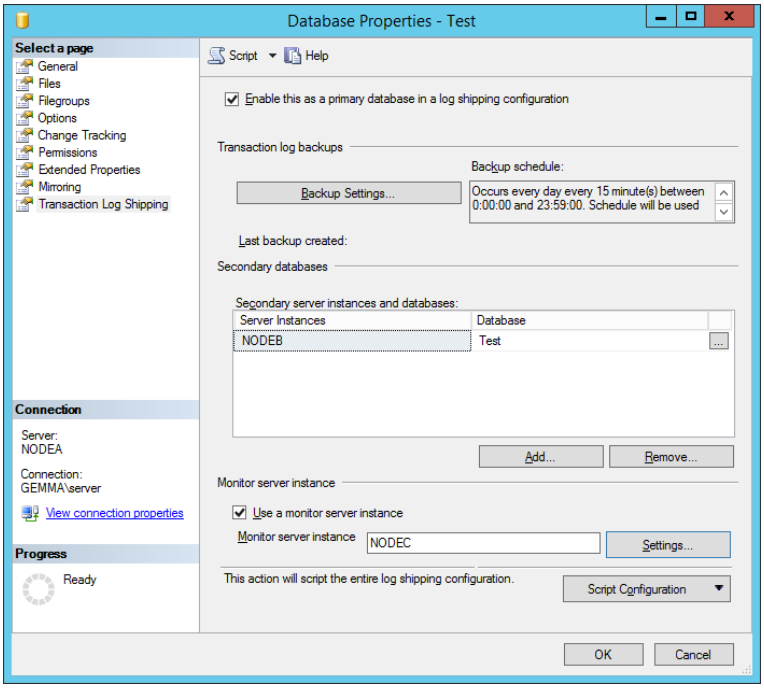


Obr. 1.25: Typická konfigurace odesílání souboru protokolu.

1.5.2. Postup k nasazení odesílání souboru protokolu

Odesílání souboru protokolu lze nasadit pouze v případě, kdy primární databáze je v režimu obnovení FULL nebo BULK-LOGED.

Na primární instanci v SQL Server Management Studiu (SSMS) pravým tlačítkem myši nad zvolenou primární databází vyberte možnost vlastnosti, z nabídky vyberte sekci odesílání souboru protokolu (Transaction Log Shipping) a zaškrtněte políčko „povolit primární databázi v konfiguraci odesílání souboru protokolu“ (Enable tis as a primary database in a log shipping configuration). Pro lepší orientaci jsou operace zobrazeny i graficky na obr. 1.26.

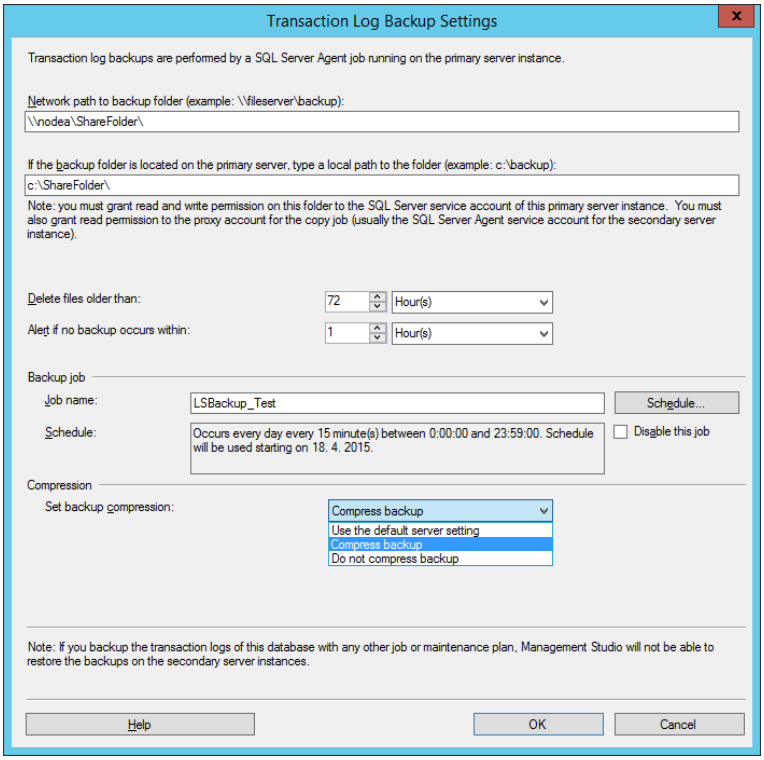


Obr. 1.26: Povolení odesílání souboru protokolu na primární databázi.

Dalším krokem je konfigurace a plánování zálohy transakčního protokolu. Zvolte možnost nastavení zálohování (Backup Settings) zobrazeném na obr. 1.26. V nastavení zálohy transakčního protokolu lze dle obr. 1.27 nakonfigurovat následující:

- Sdílenou složku, kam se transakční protokol bude ukládat.
- Mazání souborů po určité době.
- Upozornění jestli v určené době se neprovede záloha transakčního protokolu.
- Naplánovaní zálohovací úlohy (Backup job).
- Kompresi dat.

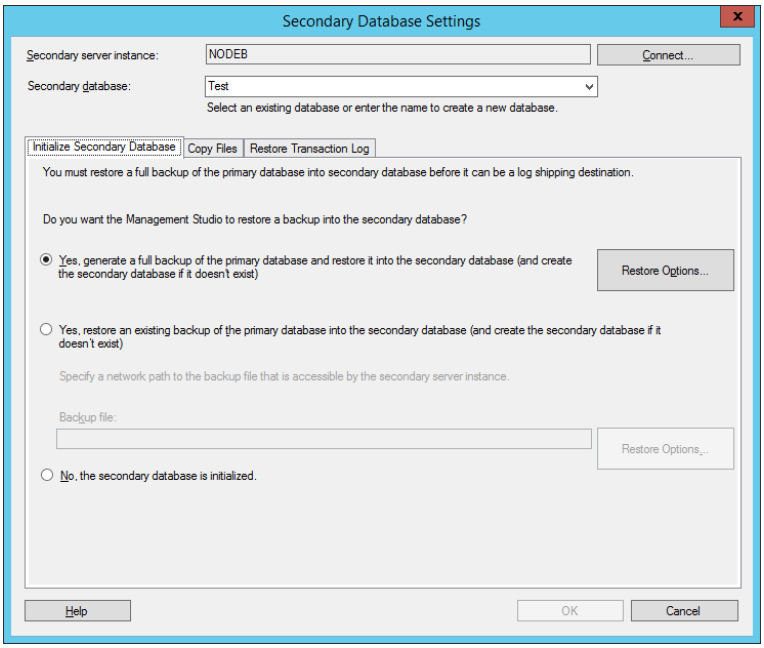
Po vyplnění parametrů pro zálohování transakčního protokolu, se vytvoří na primárním serveru zálohovací úloha (Backup job).



Obr. 1.27: Nastavení zálohy transakčního protokolu.

Tlačítkem přidat (Add) v nastavení odesílání souboru protokolu obr. 1.26, přidáme dostupné sekundární instance SQL Serveru.

K přidání sekundární instance, nás systém vyzve k připojení na sekundární instanci. Jakmile úspěšně navážeme spojení se sekundární instancí, otevře se okno k nastavení sekundární databáze znázorněné na obr. 1.28.



Obr. 1.28: Nastavení sekundární databáze.

V první záložce se nastavuje inicializace sekundární databáze, kde má následující možnosti:

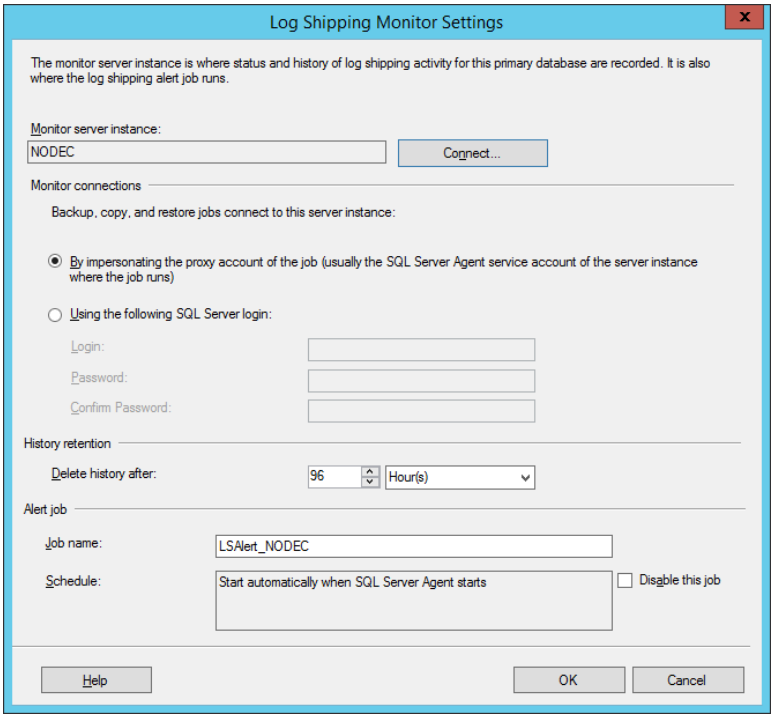
- Vytvoří se čerstvá záloha primární databáze, která se obnoví na sekundárním serveru.
- Použije se existující záloha primární databáze, která se obnoví na sekundárním serveru.
- Sekundární databáze je již inicializovaná.

V další záložce se nastavuje cílové umístění pro kopírování a konfiguraci kopírovací úlohy (copy job). A v poslední záložce nastavujeme:

- stav obnovení sekundární databáze:
    - No recovery mode – sekundární databáze je ve stavu obnovení, dokud databáze není online, nelze nad ní provádět žádné operace.
    - Standby mode – sekundární databáze je v režimu čtení, uživatel může číst data.
        - Při zvolení této možnosti se doporučuje zaškrtnout políčko odpojit uživatele připojené k sekundární databázi v případě obnovení databáze.
- Zpoždění obnovy sekundární databáze.
- Upozornění jestli se v určené době neprovede obnova transakčního protokolu.
- Naplánovaní úlohy obnovení (restore job).

Po dokončení nastavení sekundární databáze, se na sekundárním serveru vytvoří úloha kopírování a obnovení.

Posledním krokem, který není nutný, je konfigurace monitorovacího serveru. Ve vlastnostech databáze v podsložce odesílání souborů protokolu zaškrtněte políčko „použít instanci monitorovacího serveru“ viz obr. 1.26. Nastavení monitorovací instance je znázorněno na obr. 1.29.



Obr. 1.29: Nastavení monitorovací instance.

1.5.3. Bezpečnost, výhody a nevýhody odesílání souboru protokolu

Bezpečnost technologie odesílání souboru protokolu spočívá v dodržení bezpečnosti zmíněné v kapitole 1.1.3, v nastavení oprávnění pro sdílené složky na zálohování a kopírování, a v nastavení oprávnění na úrovni uživatelů a SQL Agent účtu.

Jednotlivé výhody a nevýhody nasazení technologie odesílání souboru protokolu jsou uvedeny v tab. 1.6.

Tab. 1.6: Výhody a nevýhody použití odesílání souboru protokolu.

Výhody	Nevýhody
+ poskytuje řešení zotavení po havárii pro jednu primární databázi a jednu nebo více sekundárních databází, každé na samostatné instanci SQL Serveru	- v případě havárie nutný zásah administrátora
+ podporuje omezený přístup „pouze pro čtení“ k sekundárním databázím (v průběhu restore job)	- bezpečnost
+ umožňuje uživatelsky stanovit zpoždění mezi dobou, kdy primární server zálohuje protokol primární databáze a kdy sekundární servery obnoví zálohy protokolu	- nutnost spravovat všechny databáze samostatně
+ jednoduchost	- nemožnost číst data v průběhu obnovení

1.5.4. Optimální varianta využití odesílání souboru protokolu

Optimální varianta odesílání souboru protokolu je při použití monitorovací instance, která nás upozorní v případě problému. Další velkou výhodou technologie odesílání souboru protokolu např. oproti zrcadlení databáze, je nastavení operačního módu sekundární databáze do Standby režimu, který umožnuje použít sekundární databázi ke čtení. Oproti dalším technologiím však v průběhu obnovení databáze nelze pracovat se sekundární databází.

1.6. AlwaysOn Failover Cluster Instance

SQL Server AlwaysOn je nové řešení vysoké dostupnosti a obnovení po havárii, které využívá funkce Windows Server Failover Clustering (WSFC). AlwaysOn poskytuje integrované, flexibilní řešení, které zvyšuje dostupnost aplikací, poskytuje lepší návratnost investic do hardwaru a zjednodušuje nastavení a správu vysoké dostupnosti [3].

Failover Cluster Instance (FCI) byla přidána do skupiny technologií AlwaysOn od verze SQL Serveru 2012. Ve verzi SQL Server 2008 R2 a níže, byla tato technologie známa pod názvem SQL Server Failover Clustering. Největším rozdílem oproti ostatním technologiím je, že FCI poskytuje podporu vysoké dostupnosti pro celou instanci SQL Serveru.

AlwaysOn Failover Cluster Instance je SQL Server instance, která je nainstalován přes uzly ve WSFC clusteru. Tento typ instance vyžaduje pro svou správnou funkčnost, sdílené diskové úložiště (iSCSI, SAN, atd.).

1.6.1. Přehled a princip Failover Cluster instance

FCI je skupina prostředků běžících ve WSFC s jedním nebo více uzly. Při zapnutí FCI, jeden z uzlů WSFC clusteru převezme vlastnictví skupiny prostředků a provede přepnutí SQL Server instance do online režimu. Skupina prostředků ve vlastnictví tohoto uzlu obsahuje [3]:

- Název sítě.
- IP adresy.
- Sdílené disky.
- Služby SQL Server Database Engine.
- Služby SQL Server Agent.
- Pokud jsou instalované, tak služby SQL Server Analysis.
- Pokud je nainstalována funkce FILESTREAM, tak jeden soubor sdílených prostředků.

Posloupnost událostí při plánovaném nebo neplánovaném převzetí služeb při selhání [3]:

1. Pokud dojde k selhání hardware nebo systému, všechny „špinavé“ stránky ve vyrovnávací paměti jsou zapsány na disk.
2. Na aktivním uzlu jsou zastaveny všechny příslušné služby SQL Serveru.
3. Vlastnictví skupiny prostředků se přenese do jiného uzlu v FCI.
4. Nový majitel vlastnictví skupiny prostředků spustí příslušné služby SQL Serveru.
5. Požadavky k připojení od klientských aplikací jsou automaticky přesměrovány na nový aktivní uzel ve stejné virtuální síti.

FCI zůstává online, pokud je WSFC cluster kvórum„zdravé“ tzv. majorita kvóra WSFC uzlů je k dispozici k automatickému převzetí služeb v případě selhání. Pokud WSFC cluster ztratí kvórum, usnášeníschopnost, ať už v důsledku selhání hardware, software, poruchy sítě nebo nesprávné konfigurace kvóra, celý WSFC cluster, spolu s FCI, je přiveden do režimu offline. V tomto případě je nutný zásah administrátora, aby ve zbývajících dostupných uzlech WSFC clusteru znovu zřídil kvórum, s cílem uvést WSFC Cluster a FCI do režimu online [3].

Předvídatelnost failover

V závislosti, kdy naposledy instance SQL Serveru vytvořila kontrolní bod operací (check point), může být značné množství „špinavých“ stránek uloženo ve vyrovnávací paměti. V důsledku toho, převzetí služeb při selhání trvá tak dlouho, dokud nejsou všechny „špinavé“ stránky zapsány na disk, což může vést k dlouhé a nepředvídatelné době převzetí služeb při selhání.

Od verze MS SQL Server 2012, lze ve FCI použít nepřímé kontrolní body pro omezení „špinavých“ stránek uložených ve vyrovnávací paměti. Při použití nepřímých kontrolních bodů vzrůstá pravidelné vytížení, ale na druhou stranu, je failover více předvídatelný a konfigurovatelný. Použití je vhodné např. při určení časové úseku, do kdy musí být zajištěna vysoká dostupnost [3].

Monitoring „zdraví“ a flexibilní politika failover

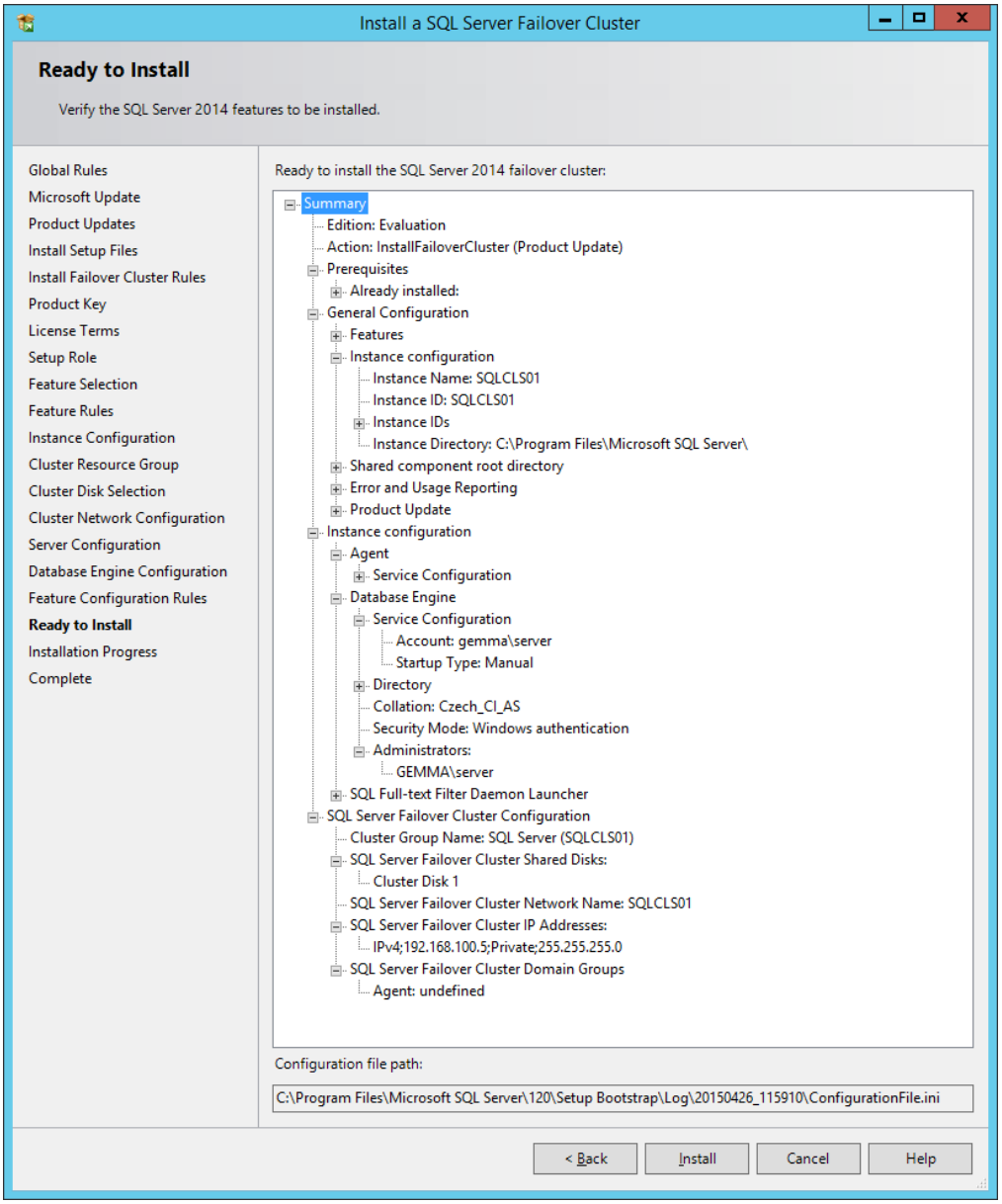
Po úspěšném spuštění FCI, služba WSFC monitoruje jak „zdraví“ podkladového WSFC clusteru, tak i „zdraví“ instance SQL Serveru. Od verze MS SQL Server 2012, služba WSFC používá vyhrazené připojení k dotazování aktivní instance SQL Serveru, pro detailní diagnostiku komponent prostřednictví uložených systémových procedur. V důsledku toho je dosaženo [3]:

- Vyhraněným připojením k instanci SQL Serveru je umožněno, spolehlivé dotazování pro diagnostiku komponent po celou dobu, i když FCI je vysoce zatížen. Díky tomu, je možné rozlišit mezi systémem, který je zatížen a systémem, který ve skutečnosti má poruchový stav. Předejde se tak falešnému převzetí služeb při selhání.
- Podrobná diagnostika komponent umožňuje nastavit pružnější politiku převzetí služeb při selhání, přičemž si můžete vybrat podmínky, za jakých dojde k převzetí služeb při selhání.
- Podrobná diagnostika komponent umožňuje také lepší řešení problému při automatickém přepnutí služeb při selhání zpětně. Podrobné diagnostické informace jsou ukládány do logů, které jsou připojeny k chybovým protokolům SQL Serveru. Pomocí některých programů je možné tyto soubory otevřít a určit příčinu selhání.

1.6.2. Nasazení Failover Cluster Instance

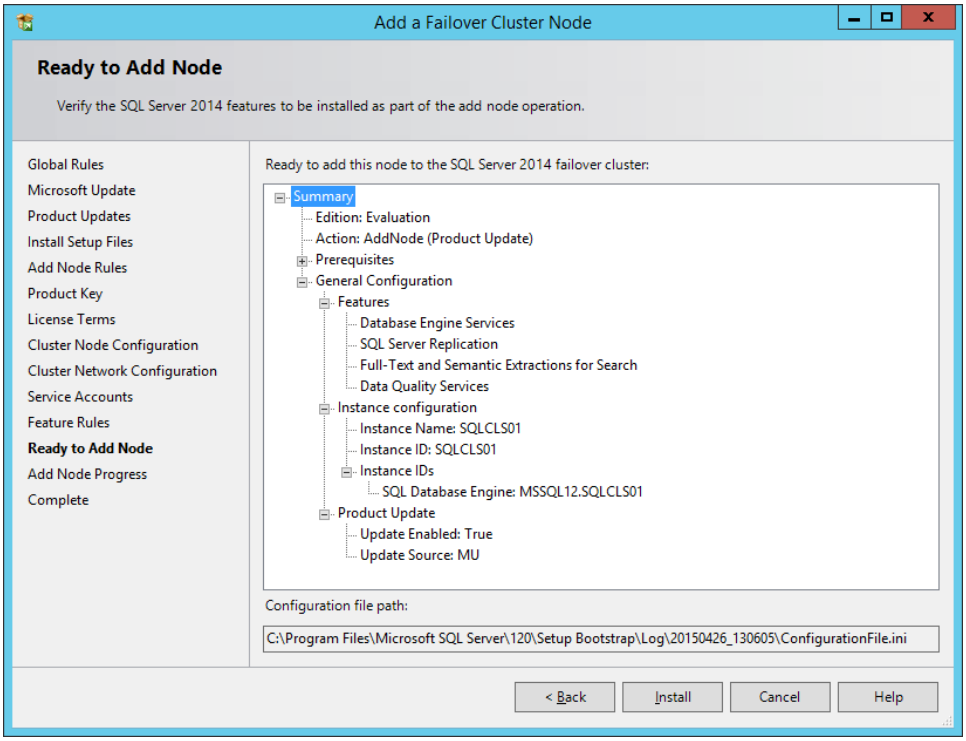
Pro nasazení technologie FCI využijeme testovací prostředí popsané v kapitole 4.3.

Začneme spuštěním instalace SQL Server Failover instance z instalačního média na NodeA a pokračujeme pomocí průvodce instalace. Na obr. 1.30 je znázorněn souhrn instalace SQL Server Failover instance.



Obr. 1.30: Souhrn Instalace SQL Server Failover Cluster instance.

Po dokončení instalace SQL Server Failover, spustíme instalačním médium SQL Serveru na NodeB a zvolíme možnost instalace, přidání Failover Cluster uzlu. Stejně jako u instalace SQL Server Failover instance na NodeA, postupujeme pomocí průvodce instalace, kde souhrn instalace je zobrazen na obr. 1.31.



Obr. 1.31: Souhrn instalace u přidání Failover Cluster uzlu.

Po dokončení instalace přidání uzlu máme vytvořenou Failover Cluster Instanci.

1.6.3. Bezpečnost, výhody a nevýhody Failover Cluster Instance

Již několik verzí Windows Server podporuje službu BitLocker[6] pro fyzické šifrování disků. Nově, od verze Windows Server 2012, se BitLocker rozšiřuje i tam, kde se dosud nenacházel a můžete jej mít nasazen v rámci clusteru snad kdekoli. Na úložišti, ze kterého se spouští virtuální servery, nám nic nebrání v tom, abychom šifrovali všechny možné a myslitelné disky serverů a to zcela bez ohledu na to, zdali se jedná o servery fyzické, či virtuální. Další stupně zabezpečení jsou stejné, jako v kapitole 1.1.3.

[6]BitLocker je nástroj na ochranu dat operačního systému Windows.

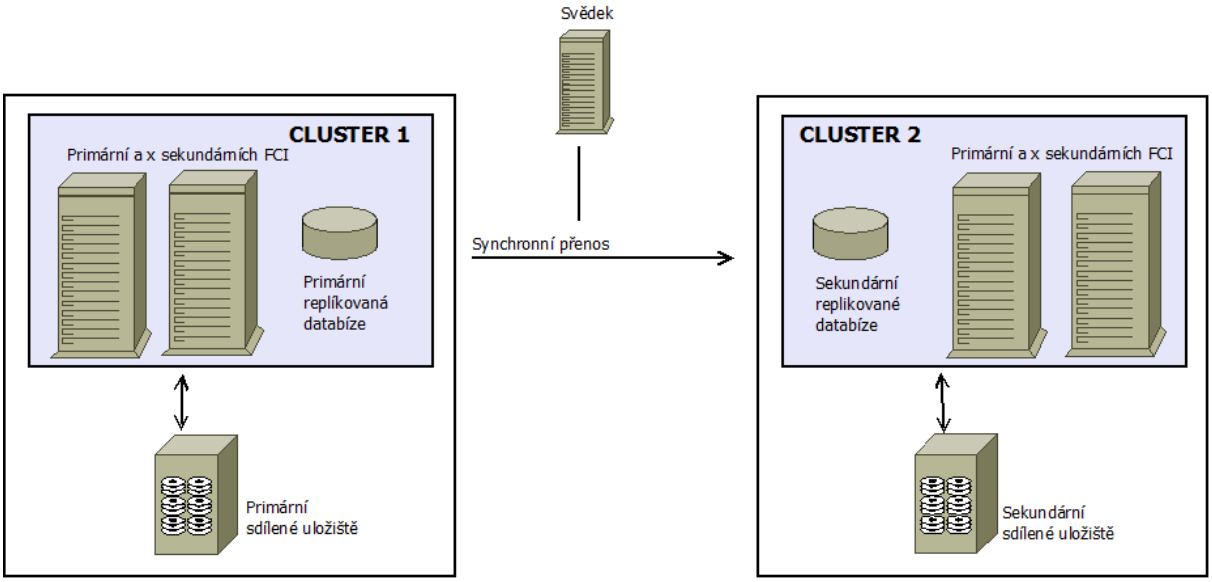
Výhody a nevýhody použití Failover Cluster Instance jsou vedeny v tab. 1.7.

Tab. 1.7: Výhody a nevýhody použití FCI.

Výhody	Nevýhody
+ Ochrana na úrovni instance prostřednictvím redundance.	- Složité nasazení sytému.
+ Automatické převzetí služeb při selhání.	- nutnost WSFC cluster, tudíž větší finanční náročnost.
+ Široká škála úložišť, včetně WSFC cluster disků (iSCSI, optické vlákno, atd.) a sdílení souborů.	- Závislost na sdíleném úložišti, v případě havárie sdíleného uložiště je vše ztraceno.
+ Řešení obnovy dat po havárii prostřednictvím multi-podsítě FCI nebo spuštěním FCI hostované databáze uvnitř AlwaysOnAvailabilityGroup.	
+ S podporou multi-podsítě v SQL Server, není vyžadováno virtuální sítě LAN, čímž se zvyšuje ovladatelnost a bezpečnost multi-podsítí FCI.	
+ Žádná rekonfigurace aplikací a klientů při převzetí služeb při selhání.	
+ Flexibilní politika failover pro granulované spouštění událostí v případě automatického převzetí služeb při selhání.	
+ Spolehlivé převzetí služeb při selhání prostřednictvím pravidelné detekce „zdraví“ pomocí vyhrazeného a stálého připojení.	
+ Konfigurovatelnost a předvídatelnost v čase při převzetí služeb při selhání prostřednictvím nepřímých kontrolních stanovišť na pozadí.	

1.6.4. Optimální varianta využití technologie Failover Cluster Instance

Optimální varianta využití technologie FCI je při použití kombinace s dalšími technologiemi vysoké dostupnosti. Před verzí MS SQL Serveru 2012 se v praxi používala kombinace s technologií zrcadlení, kde mezi clustrovanými databázemi probíhá operace zrcadlení, nejlépe synchronní a nejlépe s umístěním sekundární databáze na geograficky vzdáleném pracovišti. V případě, že dojde k výpadku na jednom clusteru, například při poškození primárního sdíleného úložiště, svědek rozpozná nefunkčnost primárního cluster a přepne služby na druhý cluster. Schéma je vyobrazeno na obr. 1.32.



Obr. 1.32: Kombinace technologií vysoké dostupnosti FailoverClustering a zrcadlení.

Od verze Windows server 2012 je optimální varianta využití technologie FCI spolu s technologií Availability Group. Tahle optimální varianta je popsána v kapitole 1.7.4.

1.7. AlwaysOn Availability Group

AlwaysOn Availability Group (AG) je technologií vysoké dostupnosti a obnovení dat po havárii, která poskytuje alternativu k technologii zrcadlení databáze. AG byla uvedena poprvé ve verzi MS SQL Serveru 2012. Účelem této technologie je maximalizovat dostupnost skupiny uživatelských databází, známé jako Availability databáze. AG podporuje sadu pro čtení a zápis na primární databáze a jednu až osm odpovídajících sekundárních databází. Sekundární databáze mohou být k dispozici pro čtení nebo pro některé operace zálohování [3].

1.7.1. Termíny a definice, princip, režimy a failover AlwaysOn Availability Group

AvailabilityGroup
Množina uživatelských databází, které selžou společně ve stejném čase nebo jejichž společnou dostupnost AlwaysOn technologie zajišťuje [3].

Availability databáze
Databáze přiřazené do Availability Group. Pro každou Availability databázi Availability Group udržuje jednu kopii pro čtení a zápis (primární databáze) a jednu až osm kopií pro čtení (sekundární databáze) [3].

Primární databáze
Kopie Availability databáze pro čtení a zápis.

Sekundární databáze
Kopie Availability databáze pouze pro čtení.

Availability replika
SQL Server instance, která udržuje kopii Availability databáze, patřící do Availability Group. Dva typy Availability replik: primární replika a sekundární replika [3].

Primární replika
Availability replika, která umožňuje čtení a zápis na primární databázi pro připojené klienty, a také posílá záznamy transakčního logu na každou primární databázi sekundárních replik [3].

Sekundární replika
Availability replika, která udržuje sekundární kopii každé Availability databáze, aby v případě převzetí služeb při selhání zastoupila primární repliku. Sekundární replika dále slouží pro přístup ke čtení na sekundární databáze a k záloze sekundární databáze [3].

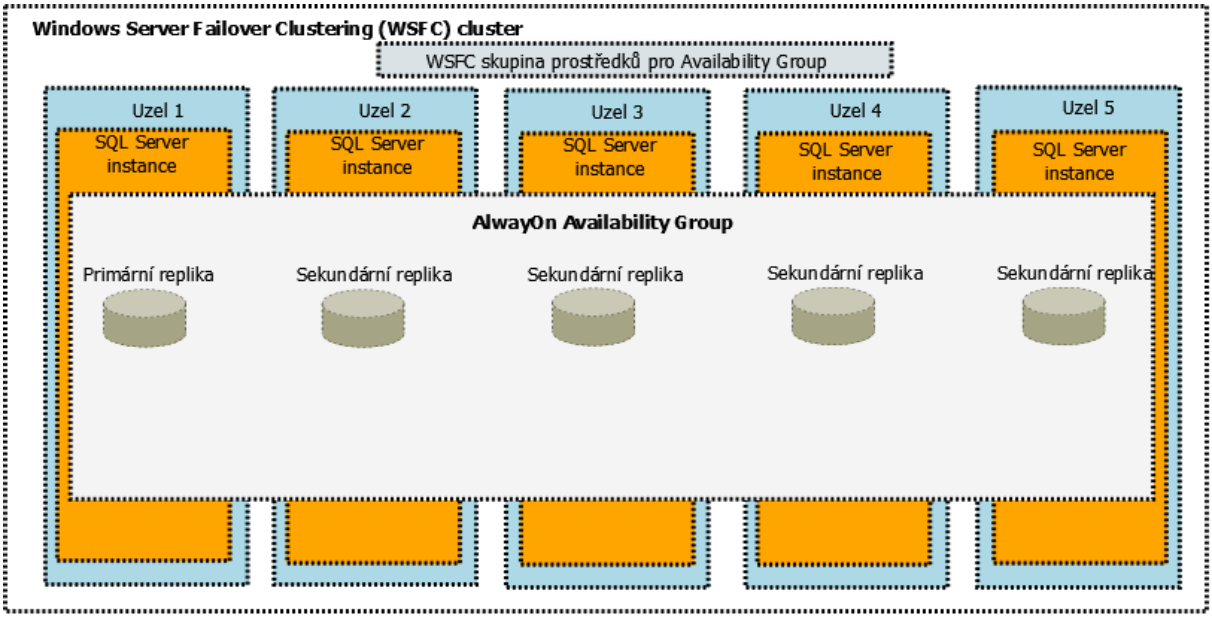
Availability Group Listener
Název serveru, ke kterému se mohou připojit klienti, aby získali přístup na primární nebo sekundární repliku AG. Availability Group Listener směřuje příchozí požadavky od klientů, aplikací na primární nebo sekundární repliky [3].

Princip AG tkví v tom, že primární replika posílá záznamy transakčního protokolu na každou primární databázi a na každou sekundární databázi. Každá sekundární replika ukládá záznamy transakčního protokolu a následně je aplikuje na odpovídající sekundární databáze. Dochází tak, k synchronizaci dat mezi primární databází a každou připojenou sekundární databází, nezávisle na ostatních databázích. Z toho důvodu, může být sekundární databáze pozastavena nebo selhat, aniž by to ovlivnilo ostatní sekundární databáze a stejně tak v případě primární databáze [3].

Fakta a požadavky AG [3]:

- Nasazení AG vyžaduje Windows Server Failover Clustering (WSFC) cluster. Každá Availability replika, každé AG musí být umístěna na jiném uzlu stejného WSFC clusteru.
- WSFC skupina prostředků je vytvořena pro každou AG, kterou vytvoříte.
- WSFC cluster sleduje prostředky skupiny k hodnocení „zdravotního“ stavu primární repliky.
- Kvórum AG je založeno na všech uzlech ve WSFC clusteru bez ohledu na to, zda daný uzel clusteru hostí nějaké Availability repliky.
- Na rozdíl od zrcadlení databáze se nepoužívá žádný svědek.

Obr. 1.33 znázorňuje AG, která obsahuje jednu primární repliku a čtyři sekundární repliky [3].



Obr. 1.33: Schéma AlwaysOn Availability Group s jednou primární replikou a čtyřmi sekundárními.

AG podporuje dva typy režimů [3]:

- Asynchronní režim.

V asynchronním režimu primární replika potvrzuje transakce bez čekání na potvrzení o zápisu záznamů transakčního protokolu na disk od sekundární repliky. Minimalizuje tak zpoždění transakcí na sekundárních databázích, ale umožňuje zaostávat za primární databází, čímž může dojít ke ztrátě dat.

- Synchronní režim.

Primární replika čeká na potvrzení o dokončení zápisu záznamů transakčního logu na disk od sekundární repliky. Synchronní režim zajišťuje synchronizaci primární databáze se sekundárními, na úkor zvýšení zpoždění transakcí.

V rámci relace mezi primární a sekundární replikou, jsou primární a sekundární role potencionálně zaměnitelné v procesu známém jako failover. Při převzetí služeb při selhání, se na cílovou sekundární repliku přepnou primární role, čímž se stane novou primární replikou. Nová primární replika přepne své databáze do režimu online, čímž se stanou primárními databázemi pro připojení klientských aplikací. Pokud bývalá primární replika se stane dostupnou, přejde do sekundární role a stane se z ní sekundární replika. Bývalé primární databáze se stanou sekundárními a dále se pokračuje v synchronizaci [3].

Existují tři formy failover: automatický, manuální a vynucený. Formy závisí na zvoleném Availability režimu [3]:

- Synchronní režim podporuje dvě formy failover:
    - Plánovaný ruční failover (bez ztráty dat).

    Ruční převzetí služeb při selhání nastane poté, co správce databáze vydá příkaz k převzetí služeb při selhání, čímž sekundární replika převezme primární roli a primární replika sekundární roli. Ruční převzetí služeb při selhání vyžaduje, aby primární a cílová sekundární replika, byly spuštěny v synchronním režimu a sekundární replika již byla synchronizovaná.
    - Automatický failover (bez ztráty dat).

    Automatické přepnutí při selhání vyžaduje, aby primární a cílová sekundární replika byly spuštěny v synchronním režimu, s nastaveným módem failover na „Automatic“. Kromě toho sekundární replika musí byt již synchronizována, musí být ustanoveno WSFC kvórum a splněny podmínky stanovené politikou failover AG.
- V asynchronním režimu je pouze jedna forma přepnutí, vynucené přepnutí (s možnou ztrátou dat). Nucené převzetí služeb při selhání je považováno za formu ručního převzetí služeb při selhání služeb, protože může být zahájeno pouze ručně. Nucené převzetí služeb při selhání je forma obnovení dat po havárii. Je to jediná forma failover, která nevyžaduje synchronizaci mezi primární a sekundární replikou.

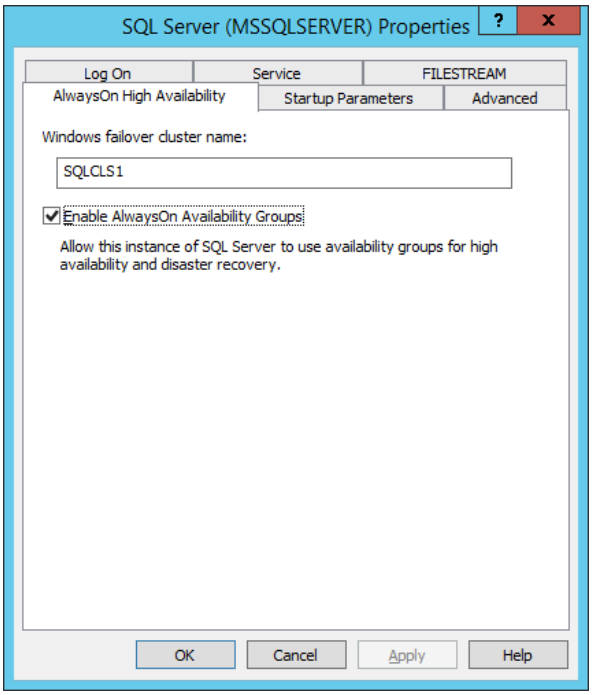
1.7.2. Nasazení technologie AlwaysOn Availability Group

Pro správný chod technologie AlwaysOn Availability Group (AG) je nutné splnit následující požadavky:

- Mít WSFC cluster s několika uzly, na kterých je nainstalována SQL Server Instance.
- Musí existovat minimálně dva uzly v clusteru, jeden pro primární repliku a druhý pro sekundární repliku.
- Pro řešení AG lze použít i samostatnou instalaci SQL Server instance, může být i clusterovaná.
- Availability Group Listener zarezervovaný v DNS.

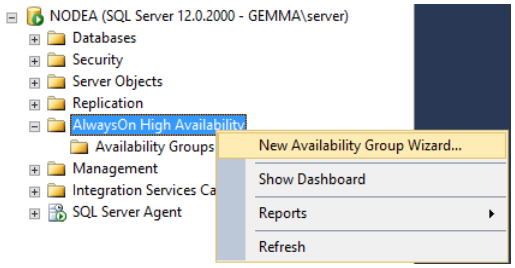
Využijeme tedy obou připravených testovacích prostředí popsaných v kapitole 4.2 a 4.3. Kde z neclustrovaného prostředí využijeme nainstalované samostatné instance SQL Serveru NodeA a NodeB, a z clusterovaného prostředí využijeme nakonfigurovaný cluster SQLCLS1.

Před nasazením technologie je nutné na instancích SQL Server NodeA a NodeB povolit funkci AlwaysOn Availability Group a přiřadit jej k existujícímu clusteru. Povolení funkce AG je znázorněno na obr. 1.34.



Obr. 1.34: Zapnutí funkce AlwaysOn Availability Group na instanci SQL Serveru.

Po zapnutí funkce AG přejdeme k instalaci. V SSMS vytvoříme novou Availability Group viz obr. 1.35.



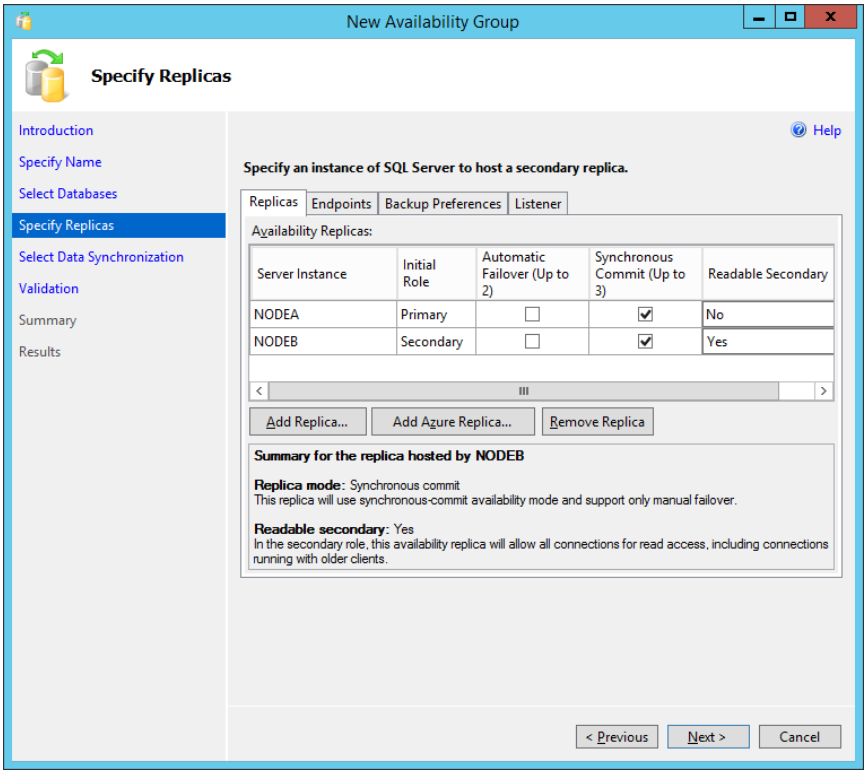
Obr. 1.35: Vytvoření nové Availability Group.

V dalších krocích instalace postupujeme dle průvodce, kde určujeme:

- Název AG.
- Databáze v AG.
- Specifikaci replik.
- Synchronizaci dat.

V sekci specifikace replik, v záložce repliky přidáváme sekundární repliky a následně konfigurujeme všechny dostupné repliky. Nastavujeme režimy, failover a možnost využití repliky pro čtení.

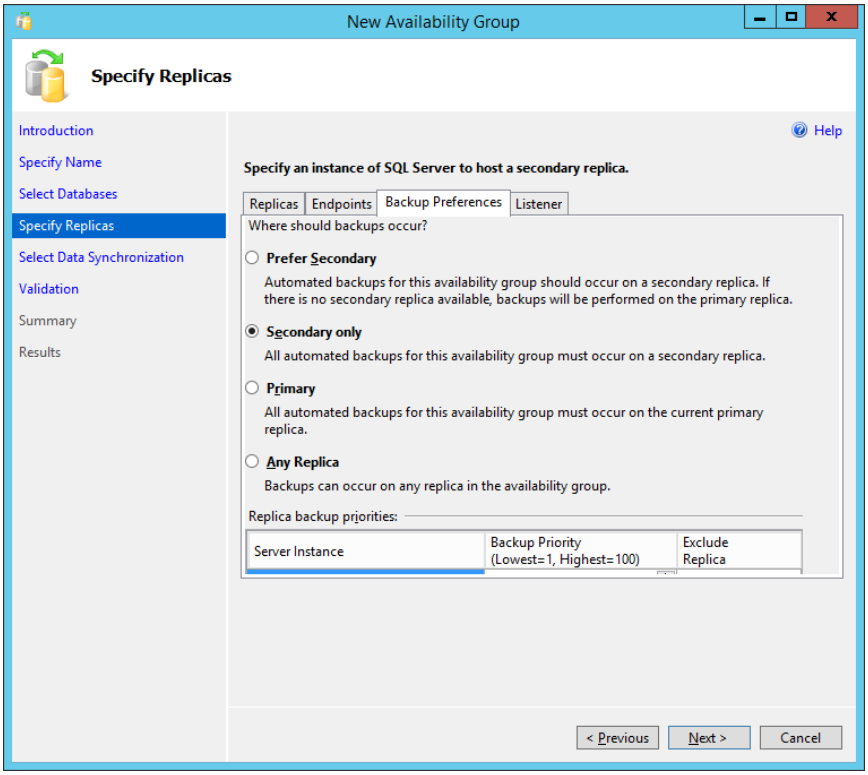
Pro testovací účely jsem zvolil manuální failover se synchronním režimem a s možnosti čtení nad sekundární replikou viz obr. 1.36.



Obr. 1.36: Konfigurace replik v Availability Group.

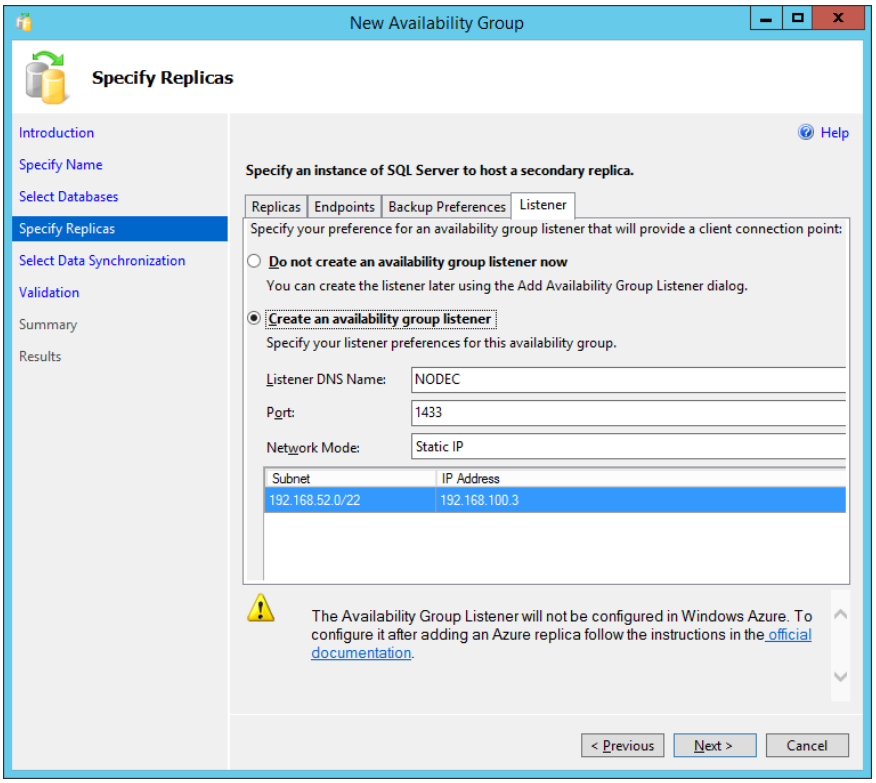
V záložce Endpoints můžete určit URL koncových bodů, porty a šifrování komunikace mezi primární a sekundární replikou. Ponecháme defaultní nastavení.

V další záložce, upřednostnění zálohování, se vybírá, kde přednostně se bude provádět záloha. Je vhodné provádět zálohování na sekundárních replikách, čímž se ušetří systémové prostředky na primární replice. Nastavení zálohování je znázorněno na obr. 1.37.



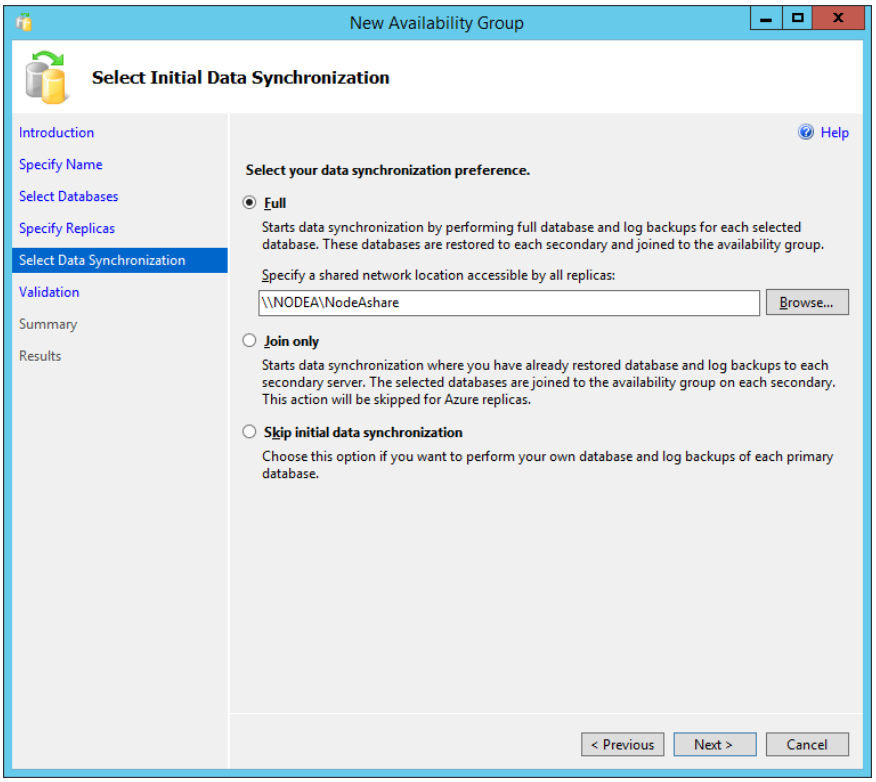
Obr. 1.37: Zálohovaní databází Availability Group.

V poslední záložce sekce specifikace replik, se konfiguruje Availability Group Listener, konfigurace je znázorněna na obr. 1.38.



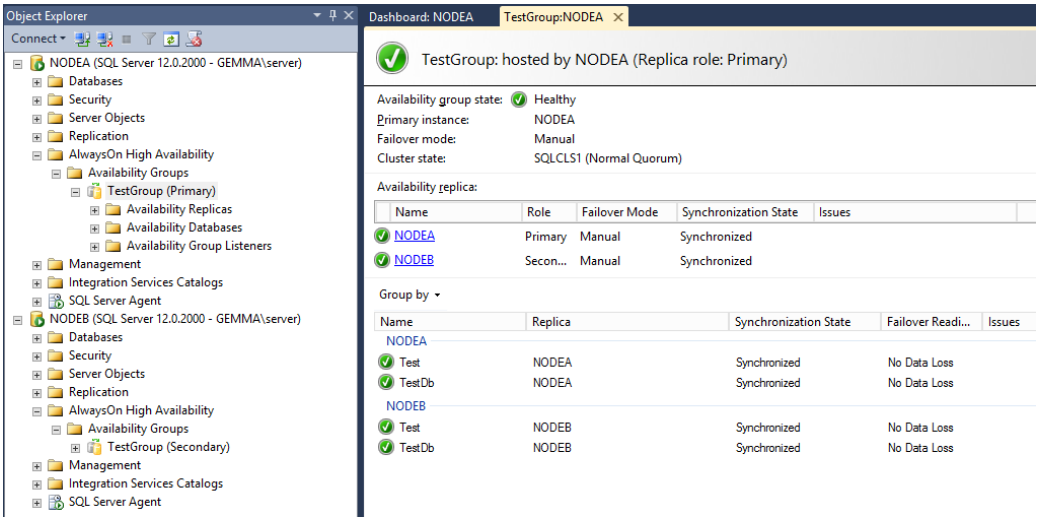
Obr. 1.38: Konfigurace Availability Group Listener.

V poslední sekci před validací, souhrnem a samotnou instalací AG, je nastavení synchronizace dat. Jedná se o prvotní synchronizaci dat. V našem případě nemáme na NodeB žádné databáze, takže zvolíme Full synchronizaci. Pro tento způsob synchronizace je třeba nadefinovat sdílenou složku, kde mají oprávnění přístupu všechny repliky. Nastavení synchronizace je znázorněno na obr. 1.39.



Obr. 1.39: Nastavení prvotní synchronizace dat v Availability Group.

Po dokončení instalace AG, je možné zkontrolovat nastavení AG pomocí tzv. nástěnky. Nástěnka je znázorněna na obr. 1.40.



Obr. 1.40: Nástěnka Availability Group.

1.7.3. Bezpečnost, výhody a nevýhody AlwaysOn Availability Group

Bezpečnost AG je stejná jako v kapitole 1.1.3, navíc je přidáno šifrování komunikace mezi primární a sekundárními databázemi.

AG poskytuje bohatou sadu možností, které zlepšují dostupnost databáze a umožňují lepší využití zdrojů. Klíčové prvky jsou shrnuty v tab. 1.8.

Tab. 1.8: Výhody a nevýhody použití AlwaysOn Availability Group.

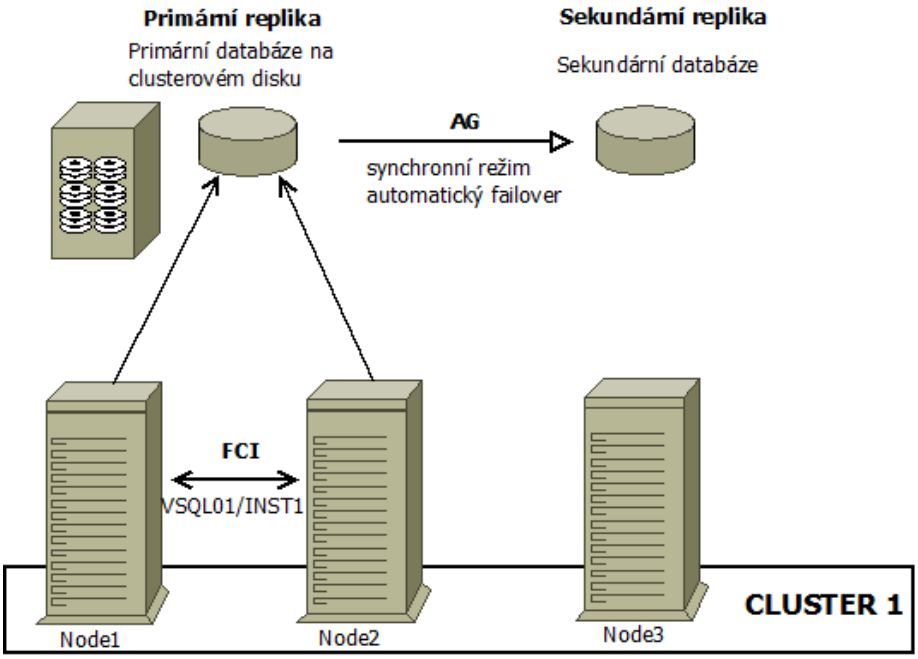
Výhody	Nevýhody
+ Umožňuje až devět Availability replik, jednu primární a jednu až osm sekundárních.	- složité nasazení sytému
+Poskytuje asynchronní a synchronní režimy.	- nutnost WSFC cluster, tudíž větší finanční náročnost
+Umožňuje několik forem failover: automatický, plánovaný manuální a nucený.	
+ Podporuje provádění operací nad sekundárními databázemi jako čtení dat a zálohování.	
+ Je dostupný Availability Group Listener pro každou Availability Group.	
+ Obsahuje flexibilní politiku failover pro větší kontrolu při selhání.	
+Podporuje automatickou opravu stránek.	
+ Poskytuje šifrování a kompresi, což umožňuje bezpečnou a vysoce výkonnou přepravu dat.	
+ Poskytuje ucelenou sadu nástrojů pro zjednodušení nasazení a správu.	

1.7.4. Optimální varianta využití AlwaysOn Availability Group

Při použití samostatné AG, je optimální varianta použití této technologie vysoké dostupnosti v synchronním režimu a s automatickým failover. Další důležitou součástí je použití Availability Group Listener, který zpracovává a směruje požadavky od klienta/aplikace na příslušnou repliku. V případě selhání, tak Listener směruje požadavky od klienta/aplikace na fungující repliky.

Při využití kombinace více technologií vysoké dostupnosti, je optimální využít kombinaci AG a FCI. Možné řešení je znázorněno na obr. 1.41.

K dispozici jsou tři servery Node1, Node2 a Node3 v clusteru a sdílené úložiště. Node1 a Node2 použijeme pro FCI, která bude navrhnuta jako primární replika. Node3 určíme jako sekundární repliku. Mezi FCI a Node3 použijeme synchronní režim s automatickým failover. V případě selhání Node1 se služby přepnout na Node2 a v případě selhání FCI, dále máme sekundární repliku na Node3. Node3 také určíme jako primární zálohovací uzel, pro zálohování databází s možnosti čtení pro vyobrazení velkých sestav dat.



Obr. 1.41: Kombinace Failover Cluster Instance a Availability Group.

1.8. Dostupnost a porovnání technologií vysoké dostupnosti

Tab. 1.9: Porovnání technologií vysoké dostupnosti.

Technologie vysoké dostupnosti / obnovení po havárii v systému SQL Server	Ztráta dat	Doba mimo provoz	Automatický failover	Složitost
AlwaysOnAvailability Group - asynchronní režim	Možnost určité ztráty dat	Minuty	NE	velká
AlwaysOnAvailability Group - synchronní režim	Žádná ztráta dat	Sekundy	ANO	velká
AlwaysOnFailover Cluster Instance	Žádná ztráta dat	Sekundy až minuty	ANO	velká
Odesílání souboru protokolu	Možnost určité ztráty dat	Sekundy plus doba obnovení databáze	NE	malá
Replikace transakcí	Možnost určité ztráty dat	Sekundy	NE	malá
Zálohování, kopírování, obnova	Možnost určité ztráty dat	Hodiny až dny, v závislosti na kapacitě databáze	NE	nejmenší
Zrcadlení - režim vysoké ochrany s automatickým předáním služeb při selhání	Žádná ztráta dat	Sekundy	ANO	malá
Zrcadlení - režim vysoký výkon (asynchronní)	Možnost určité ztráty dat	Minuty	NE	malá

Význam jednotlivých sloupců uvedených v tab. 1.9:

- **Ztráta dat**
  Určuje, zda v případě plánovaného či neplánovaného výpadku dojde ke ztrátě dat v závislosti na použité technologii.
- **Doba mimo provoz**
  Určuje dobu, po kterou je systém mimo provoz v případě plánovaného či neplánovaného výpadku.
- **Automatický failover**
  Určuje, zda v případě plánovaného či neplánovaného výpadku dojde k automatickému přepnutí na záložní systém bez pomocí administrátora.
- **Složitost**
  Určuje složitost nasazení jednotlivých technologií.

Tab. 1.10: Dostupnost technologií vysoké dostupnosti v jednotlivých verzích SQL Serveru.

Technologie vysoké dostupnosti SQL Serveru	SQL Server 2005	SQL Server 2008	SQL Server 2008 R2	SQL Server 2012	SQL Server 2014
AlwaysOn Availability Group - asynchronní režim	NE	NE	NE	ANO	ANO
AlwaysOn Availability Group - synchronní režim	NE	NE	NE	ANO	ANO
AlwaysOn Failover Cluster Instance	dříve Failover Clustering	dříve Failover Clustering	dříve Failover Clustering	ANO	ANO
Odesílání souboru protokolu	ANO	ANO	ANO	ANO	ANO
Replikace transakcí	ANO	ANO	ANO	ANO	ANO
Zálohování, kopírování, obnova	ANO	ANO	ANO	ANO	ANO
Zrcadlení - režim vysoké ochrany s automatickým předáním služeb při selhání	ANO	ANO	ANO	ANO	ANO
Zrcadlení - režim vysoký výkon (asynchronní)	ANO	ANO	ANO	ANO	ANO

Pole ANO nebo NE v tab. 1.10 určuje, zda technologie vysoké dostupnosti je dostupná ve vybrané verzi SQL Serveru.

2. Scénáře pro využití technologií vysoké dostupnosti v praxi

2.1. Scénář první, varianta pro malé a střední firmy

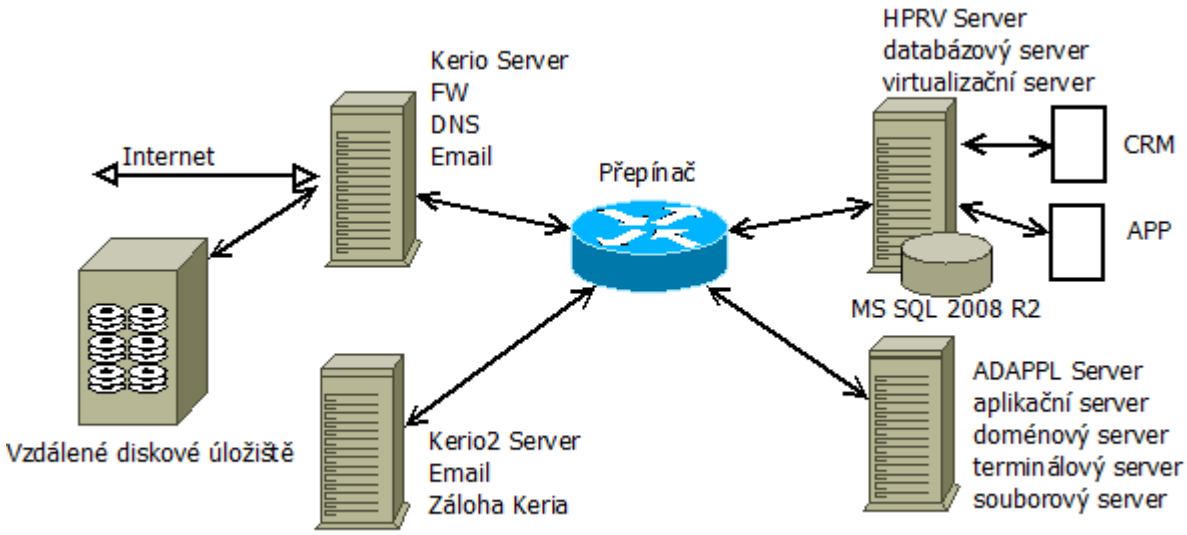
2.1.1. Popis současného stavu

Firma X o velikosti 15 uživatelů provozuje následující servery:

- Kerio – poštovní server, který slouží zároveň jako firewall a DHCP server. Operační systém Windows 7 Professional.
- Kerio2 – záložní poštovní server, který slouží k zálohování emailových schránek ze serveru Kerio. Operační systém Windows XP.
- ADAPLL – aplikační, souborový, terminálový a doménový server, který obsahuje:
    - Doménový kontrolér, DNS server.
    - Aplikaci na účetnictví.
    - Aplikaci pro sledování firemních vozidel.
    - Aplikaci pro výpočet mezd.
    - Souborový server.

  Operační systém Windows Server 2003 R2.
- HPRV – databázový a Hyper-V server, který obsahuje:
    - MS SQL 2008 R2, na kterém běží produkční databáze CRM a dalších aplikací.
    - Platformu Hyper-V, na které existují následující virtuální servery:
        - CRM – Aplikační server pro provoz CRM. OS Windows Server 2008 R2.
        - APP – Aplikační server pro provoz vlastní aplikace. OS Windows XP.
- Vzdálené diskové uložiště

Pro lepší orientaci je znázorněno schéma sítě na obr. 2.1.



Obr. 2.1: Předchozí schéma sítě firmy X.

Vysoká dostupnost v současném prostředí je řešena zálohováním databází na HPRV serveru a souborů na souborovém serveru ADAPPL na vzdálené úložiště, a v záloze emailových schránek z Keria na Kerio2.

Na serveru ADAPPL došlo k havárii s nenávratným poškozením serveru. Jelikož se jednalo o server, na kterém je nainstalovaný doménový kontrolér a většina důležitých aplikací, nefungoval provoz firmy X několik dní. Jako provizorní řešení si firma X vypůjčila náhradní server, na kterém byla obnovena doména a aplikace z tří měsíční zálohy ADAPLL serveru. Došlo tedy k výrazné finanční újmě. Aby se situace neopakovala, firma X vypsala výběrové řízení na řešení vysoké dostupnosti.

2.1.2. Požadavky na výsledné řešení

Vytvořte co nejlevnější řešení vysoké dostupnosti pro systém serverů, souborů a databází ve firmě X, kde obnovitelnost dat v případě havárie nepřesáhne 5 hodin, s přijatelnou úrovní ztráty dat jeden den. K dispozici máte:

- HPRV server na němž máte k dispozici následující licence:
  - 1x SQL Server Standard 2008 R2.
  - 15x SQL 2008 R2 Caly.
  - 3x Windows Server 2008 R2.
- Kerio server, který dále bude fungovat jako FW a DHCP server s tím, že poštovní server se přesune na jiný server a tento server bude sloužit pouze pro zálohu emailových schránek.
- Vzdálené diskové úložiště

Aplikace nainstalované na zápůjčním serveru ADAPPL, CRM, aplikace na APP a databáze mají přibližně velikost 700 GB.

2.1.3. Návrh řešení

Hlavním kritériem, z kterého vycházíme je doba obnovy systému do 5 hodin a maximální možná ztráta dat 1 den.

Pro správnou funkčnost celého systému je vhodné oddělit separátně aplikace a funkčnosti systému na jednotlivé servery, aby se v případě výpadku jednoho serveru neovlivnil chod celého systému. Avšak zakoupit pro každou aplikaci a funkčnost nový server, je velice finančně náročné jak z hlediska HW, tak SW. Z toho důvodu využijeme výhod virtualizace, což nám umožní oddělit aplikace a funkce na jednotlivé virtuální počítače. Funkce virtualizace popsané v kapitole 1.2, nám dále umožní sestavit plán obnovení v případě havárie, tedy zajistit vysokou dostupnost. Ke splnění výše zmíněných kritérií využijeme funkce export-import virtuálních počítačů.

Jelikož server ADAPPL je pouze zápůjční, je nutné navrhnout nový fyzický server, který převezme všechny funkce současného serveru. Pokud však dojde k poškození nového fyzického serveru, nelze splnit požadované kritéria. Výsledkem toho, je nutné sestavit minimálně dva fyzické servery. Sestavení serverů je uvedeno v tab. 2.1.

Každý večer pracovního dne, tak pomocí funkce Hyper-V export VM, přeneseme virtuální počítače z jednoho serveru na druhý a opačně. V případě selhání jednoho z fyzických serverů, obnovíme večer přenesené virtuální počítače, čímž splníme požadované kritéria. Pokud však dojde k výpadku obou serverů, což je velice nepravděpodobné, ztratíme všechny data. Z toho důvodu vytvoříme skript, který přenese z pátku vytvořené virtuální počítače na vzdálené diskové úložiště.

Servery jsou identické a jsou navrženy tak, aby v případě selhání jednoho ze serverů mohly fungovat všechny virtuální počítače na jediném serveru. Jeden server obsahuje následující komponenty:

- 4 jádrový procesor.
- 4x 16GB operační paměti.
- 4x 1TB SATA disk, v RAID10, takže výsledná kapacita bude 2 TB.
- Raidový kontrolér.
- 2x napájecí zdroje.
- 3 roky podpory v případě havárie, s garantovanou 4 hodinovou odezvou.

Tab. 2.1: Návrh serverů.

Popis	Počet	Cena	Cena celkem
Express x3650 M4, Xeon 4C E5-2603v2 80W 1.8GHz/1333MHz/10MB, 1x4GB, O/Bay HS 2.5in SAS/SATA, SR M5110e, Multi-Burner, 550W p/s, Rack	2	33 627 Kč	67 254 Kč
Express 16GB (1x16GB, 2Rx4, 1.5V) PC3-14900 CL13 ECC DDR3 1866MHz LP RDIMM	8	7 088 Kč	56 704 Kč
Express IBM 1TB 2.5in SFF HS 7.2K 6Gbps NL SATA HDD	8	10 355 Kč	82 840 Kč
Express ServeRAID M5100 Series 512MB Cache/RAID 5 Upgrade for IBM System x	2	3 479 Kč	6 958 Kč
Express ServeRAID M5100 SeriesBatteryKitfor IBM System x	2	2 661 Kč	5 322 Kč
Express IBM System x 550W HighEfficiencyPlatinum AC Power Supply	2	4 716 Kč	9 432 Kč
3 YearOnsiteRepair 9x5 4 Hour Response	2	13 437 Kč	26 874 Kč
		Celkem	**255 384 Kč**

Optimální varianta je pro každou aplikaci a funkci vytvořit vlastní VM. K vytvoření VM, je však zapotřebí licence, z toho důvodu je vhodné aplikace a funkce uspořádat tak, abychom využili co nejméně licencí. Uspořádání VM:

- Doménový kontrolér.
- Databázový server a druhý doménový kontrolér.
- CRM.
- Terminálový server, na kterém nainstalujeme aplikace účetnictví, sledování firemních vozidel a výpočet mezd.
- Poštovní server.
- Aplikační server pro provoz vlastní aplikace.

Souborový server vytvoříme na „starém“ serveru HPRV.

Z důvodu optimalizace výkonu, pak rozložíme zátěž jednotlivých VM na nové fyzické servery NodeA a NodeB. Rozložení zátěže:

- NodeA:
    - Databázový server a druhý doménový kontrolér (DCDB).
    - Terminálový server (TERM).
    - Aplikační server s vlastní aplikací (APP).
- NodeB:
    - CRM server (CRM).

    - Doménový kontrolér (DC).
    - Poštovní server (Mail).
- HPRV:
    - Souborový server.

Je vhodné umístit každý doménový kontrolér na jiný fyzický server, aby v případě výpadku jednoho ze serverů byl jeden doménový kontrolér vždy v provozu. Předejdeme tak hodně problémům.

Když už máme uspořádané virtuální počítače a rozloženou zátěž, potřebujeme opatřit VM příslušnými licencemi. V tab. 2.2 jsou uvedeny potřebné licence pro správný běh systému.

Tab. 2.2: Kupované licence pro firmu X.

Produkt	ks	Cena	Celkem
OEM WinSvrStd 2012 R2 x64 CZ 1pk DVD 2CPU / 2VM	6	16 770,00 Kč	100 620,00 Kč
WinSvr CAL 2012 OLP NL User CAL	15	1 032,00 Kč	15 480,00 Kč
SQL SvrStd 2014 OLP NL	0	27 555,00 Kč	0,00 Kč
SQL CAL 2014 OLP NL User CAL	0	6 413,00 Kč	0,00 Kč
WinRmtDsktpSrvcsCAL 2012 OLP NL UsrCAL	15	3 577,00 Kč	53 655,00 Kč
		CELKEM	**169 755,00 Kč**

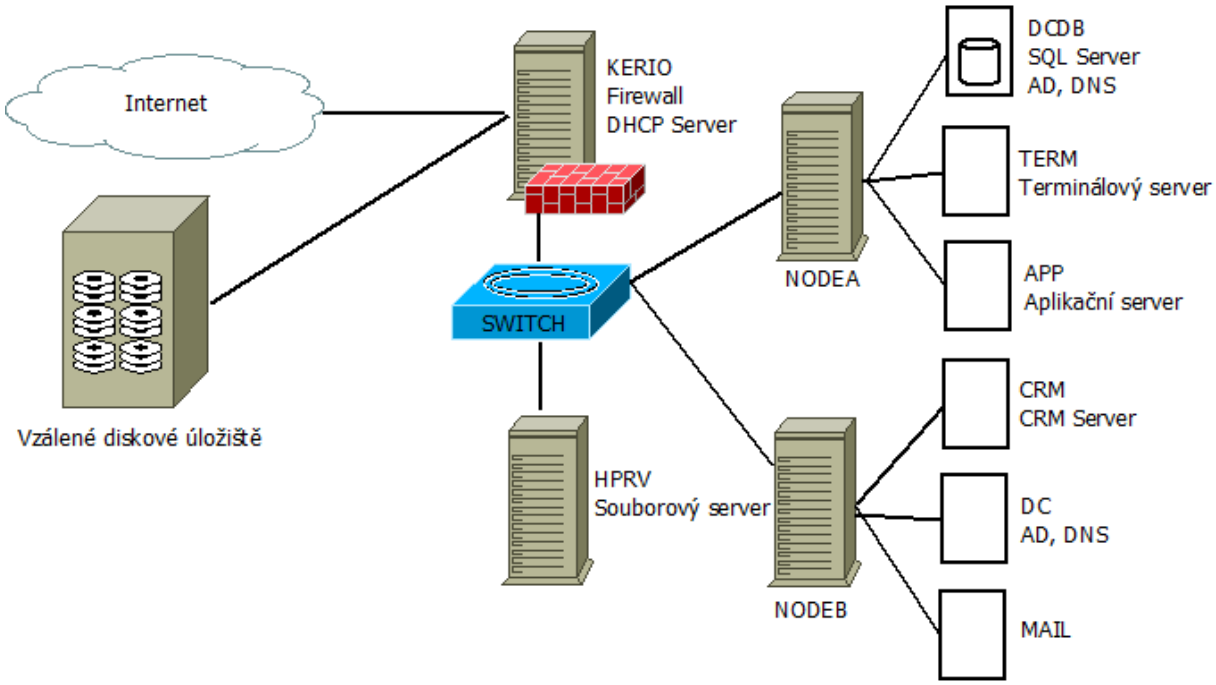
Význam licencí uvedených v tab. 2.2:

- *OEM WinSvrStd 2012 R2 x64 CZ 1pk DVD 2CPU / 2VM*
  Licence Windows Server Standard 2012 R2, která je vázaná na fyzický server (OEM), s kterou nelze přesouvat mezi servery. Licence umožňuje vytvořit 2 VM, tím pádem máme k dispozici pro každý fyzický server šest licencí. V případě havárie jednoho ze serverů a obnovy VM na druhém serveru máme dostatek licencí pro správný běh systému.
- *WinSvr CAL 2012 OLP NL User CAL*
  Otevřená licence (neváže se na fyzický server) pro každého uživatele v síti.
- *WinRmtDsktpSrvcsCAL 2012 OLP NL UsrCAL*
  Otevřená licence k přístupu na terminálový server pro každého uživatele v síti.

Licenci SQL Server Standard převezmeme ze serveru HPRV z důvodu finanční úspory.

Celková cena za hardware a software tedy činí **425 139 Kč**.

Pro lepší orientaci v navrhovaném řešení slouží schéma sítě ve firmě X znázorněné na obr. 2.2.



Obr. 2.2: Návrh výsledného schéma sítě.

Detailní popis zajištění vysoké dostupnosti ve firmě X

K exportu VM na druhý fyzický server slouží powershell skript ve výpis kódu 2.1, který se spouští pomocí naplánované úlohy každý pracovní den večer, tedy v době kdy je nejmenší vytížení serverů a sítě.

Výpis kódu 2.1: Powershell skript pro přenos virtuálních počítačů.

```
[cmdletbinding(SupportsShouldProcess=$True)]


Param(
[Parameter(Position=0,#Mandatory=$True,
HelpMessage="Enter thevirtualmachinenameornames",
ValueFromPipeline=$True,ValueFromPipelineByPropertyName=$True)]
[ValidateNotNullorEmpty()]
[Alias("name")]
[string[]]$VM,


[Parameter(Position=1)]
[ValidateNotNullorEmpty()]
[string]$Path,


[Parameter(Position=2)]
[switch]$Daily,


[Parameter(Position=3)]
[switch]$AsJob
)
Begin {
```

```
################################KONFIGURACE###################################
#CO chci exportovat
$VM= ("APP","TERM","DCDB")
#KAM to chci vyexportovat
$Path="\\NodeB\VM_backup\"

#JAK ČASTO jenom název - žádna funkce
$type="Daily"
#KOLIK složek chci zanechat
$retain=1
##############################################################################

Write-Verbose"Processing$typebackups. Retaining last $retain."

#getbackupdirectory list
Try {
Write-Verbose"Checking$pathforsubfolders"

#getonlydirectoriesunderthepaththat start withWeeklyorMonthly
$subFolders=dir-Path$path\$type*-Directory-ErrorActionStop
}
Catch {
Write-Warning"Failed to enumeratefoldersfrom$path"
#bailoutofthescript
return
}

#checkifanybackupfolders
if ($subFolders) {
#iffound, getcount
Write-Verbose"Found$($subfolders.count)folder(s)"

#if more thanthevalueof $retain, deleteoldestone
if ($subFolders.count-ge$retain ) {
#getoldestfolderbased on itsCreationTimeproperty
$oldest=$subFolders|sortCreationTime|Select-first1
Write-Verbose"Deletingoldestfolder$($oldest.fullname)"
#deleteit
$oldest|Remove-Item-Recurse-Force
    }

 } #if $subfolders
else {
#ifnonefound, createfirstone
Write-Verbose"No matchingfoldersfound. Creatingthefirstfolder"
}
```

```
#createthefolder
#getthecurrentdate
$now=Get-Date

#nameformatisType_Year_Month_Day_HourMinute
$childPath="{0}_{1}_{2:D2}_{3:D2}_{4:D2}{5:D2}"-f$type,$now.year,$now.month,$now.day,$now.hour,$now.minute

#create a variablethatrepresentsthenewfolderpath
$new=Join-Path-Path$path-ChildPath$childPath

Try {
Write-Verbose"Creating$new"
#Createthenewbackupfolder
$BackupFolder=New-Item-Path$new-ItemTypedirectory-ErrorActionStop
}
Catch {
Write-Verbose"Exportingvirtualmachines"
#failed to createfolder so bailoutofthescript
Return
}
} #end begin

Process {

#onlyprocessif a backupfolderwascreated
if ($BackupFolder) {
#export VMs
#define a hashtableofparameters to splat to Export-VM
$exportParam= @{
Path=$new
Name=$Null
ErrorAction="Stop"
    }
if ($asjob) {
Write-Verbose"Exporting as background job"
$exportParam.Add("AsJob",$True)
    }

Write-Verbose"Exportingvirtualmachines"
<#
     Go througheachvirtualmachinename, and export itusing Export-VM
    #>
foreach ($namein$VM) {
$exportParam.Name=$name
#ifthe user did not include -WhatIfthenthemachinewillbeexported
```

```
#otherwisetheywillget a WhatIfmessage
if ($PSCmdlet.shouldProcess($name)) {
Try {
Export-VM@exportParam
            }
Catch {
Write-Warning"Failed to export virtualmachine(s). $($_.Exception.Message) "
            }
        } #whatif
    } #closeforeach
} #ifbackupfolderexists
} #Process
End {
Write-Host"Export scriptfinished."-ForegroundColorGreen
}
```

Výpis kódu 2.1 vyexportuje ze serveru NodeA určené VM na server NodeB a naopak. Z důvodu omezené diskové kapacity na fyzických serverech, je skript upraven tak, aby každý den přemazával vytvořené VM z minulého večera.

Pro případ, že havárie bude na obou serverech, se každou sobotu spouští pomocí naplánované úlohy dávkový soubor ve výpis kódu 2.2, který zkomprimuje vyexportované VM z pátku a přenese jej na vzdálené síťové úložiště.

Výpis kódu 2.2: Kód dávkového souboru pro komprimaci záloh VM a následný přenos na vzdálené úložiště.

```
"c:\Program Files (x86)\7-Zip\7z.exe" a -tzip "\\diskstation\SHARED\Virtual_Images\backupB.zip" "c:\VM_backup\"
rename "\\diskstation\SHARED\Virtual_Images\backupB.zip" "Backup_HypervB (%date%).zip"
```

Vysoká dostupnost databázového serveru DCDB je zajištěna pomocí zálohování databází. Použití jiné technologie není vhodná z důvodu finanční náročnosti, zejména v koupi nových licencí SQL Serveru.

Na DCDB je vytvořen plán údržby, který vytváří denní zálohu do cílového umístění. K zajištění obnovy databází v případě havárie, se vytvořené zálohy přenáší pomocí dávkové úlohy ve výpis kódu 2.3 na vzdálené diskové úložiště, který se denně spouští naplánovanou úlohou.

Výpis kódu 2.3: Kód dávkového souboru pro komprimaci záloh VM a následný přenos na vzdálené diskové úložiště.

```
"c:\Program Files (x86)\7-Zip\7z.exe" a -tzip "d:\Backup\backupdb.zip" "d:\SQLBackup"
ECHO zabalení souborů
XCOPY d:\Backup\backupdb.zip \\diskstation\SHARED\Daily\db\
ECHO kopírování souborů na SAN
```

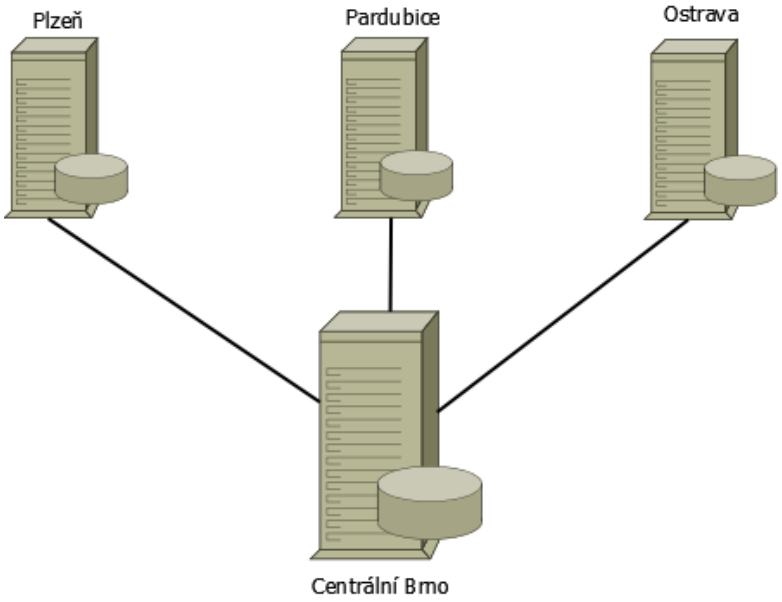
Stejným způsobem, pomocí dávkové a naplánované úlohy se přenáší soubory ze souborového server HPRV.

2.2. Scénář druhý, shromažďovaní dat na centrálním serveru

2.2.1. Popis současného stavu s požadavky na výsledné řešení

Firma Y prodávající produkty se zaměřením na meteorologii, sbírá ze svých přístrojů naměřená data do centrální databáze, kde následně vypracovává statistiky pro předpověď počasí v ČR.

Jelikož počet přístrojů narůstá, vzniká tím větší časová prodleva pro zpracování dat. Firma Y se rozhodla rozšířit současné řešení o další servery. Dokoupila potřebný hardware a software, a servery geograficky rozmístili do Ostravy, Plně a Pardubic s tím, že centrální server zůstává v Brně. Pro lepší představu je rozvržení serverů zobrazeno na obr. 2.3. Jelikož si všechny potřebné licence a hardware zakoupili svépomoci, cena řešení v tomto scénáři nelze určit.



Obr. 2.3: Scénář 2, současné schéma serverů.

Přístroje jsou přenastaveny na příslušné servery dle polohy, čímž dojde k celkovému zrychlení systému.

Firma Y vypsala výběrové řízení, na zpracování studie pro optimální variantu zasílání dat z databází v Plzni, Pardubicích a Ostravě na centrální server do Brna s tím, že finanční stránka nehraje roli.

2.2.2. Návrh řešení

Jaké databázové řešení vysoké dostupnosti použít pro tento scénář? Projdeme si všechny a určíme nejvýhodnější technologii vysoké dostupnosti:

1. Obnova databází – nepoužitelné. Rychlost obnovy dat je v našem případě nejdůležitější, tato technologie je nejpomalejší, kde obnova dat může trvat i několik hodin. Navíc sběr dat probíhá ze tří zdrojů do jednoho, což tato technologie neumožňuje.
2. Obnova serverů – nepoužitelné, jelikož se nejedná o databázové řešení.
3. Zrcadlení databáze – nepoužitelné. Zrcadlení databáze pracuje na principu identické kopie databází mezi dvěma instancemi SQL Serveru. Nepodporuje tak zasílání dat z více zdrojů do jednoho. Sekundární instance nepodporuje možnost čtení dat nad databázemi.

4. Replikace transakcí – optimální řešení. Změny se odesílají hned, jak nastanou. Je umožněno zasílat data z více zdrojů do jednoho. Centrální server má oprávnění provádět veškeré operace na databázích (číst, zapisovat, měnit). Možnost použití ne-SQL databází.
5. Odesílání souboru protokolu – použitelné, ale ne optimální. Umožňuje automatické odesílání záloh transakčního protokolu z primární databáze na primárním serveru, na jednu nebo více sekundárních databází samostatné instanci sekundárního serveru. Firma Y potřebuje přesně opačný způsob, kdy ze sekundárních instancí je zasílán transakční protokol na primární instanci. Technologii by však šlo za předpokladu, že servery v Plzni, Pardubicích a Ostravě budou primárními a v Brně bude sekundární instance. Nejedná se však o optimální řešení.
6. Failover Cluster Instance – nepoužitelné. Jedná se spíše o řešení v případě havárie. Řešení vysoké dostupnosti instance SQL Serveru.
7. Availability Group – nepoužitelné. Na primární a sekundárních replikách se udržují v synchronizaci identické databáze. Firma Y obsahuje na sekundárních serverech databáze s rozdílnými daty.

Optimální variantou je tedy využití technologie vysoké dostupnosti replikace transakcí.

Za předpokladu, že na centrálním serveru v Brně již existují data a na sekundárních serverech nikoliv. Nasazení technologie replikace transakcí bude probíhat v následujících krocích:

1. Dle postupu popsaného v kapitole 1.4.2 na sekundární servery, nad uživatelskými databázemi vytvoříme distributora.
2. Vytvoření vydavatele na každém sekundárním serveru bez použití snímku databáze na začátku replikace, jelikož už databáze s daty na primárním serveru existuje.
3. Vytvoření odběratele – primární server.
4. Kontrola funkčnosti pomocí monitoru replikací.

V případě, že na sekundárních serverech v databázi nastane jakákoliv změna, pomocí nakonfigurované replikace transakcí, se změna okamžitě promítne na centrální server. Firma Y, tak může pracovat s reálnými daty v reálném čase.

2.3. Scénář třetí, datové centrum

Datové centrum zkráceně datacentrum je místo, ve kterém je možné v „ideálních“ podmínkách, s vysokou mírou fyzické bezpečnosti, vysokou mírou vysoké dostupnosti a velmi dobrou konektivitou do veřejné sítě provozovat servery, datové úložiště, cloudové služby a další.

2.3.1. Popis současného stavu, požadavky na řešení

Firma Z, nadnárodní společnost hostující databáze na vlastním datovém centru, vypsala výběrové řízení na řešení vysoké dostupnosti s největší možnou mírou bezpečnosti pro jednu instanci s několika uživatelskými databázemi. Finanční stránka nehraje žádnou roli.

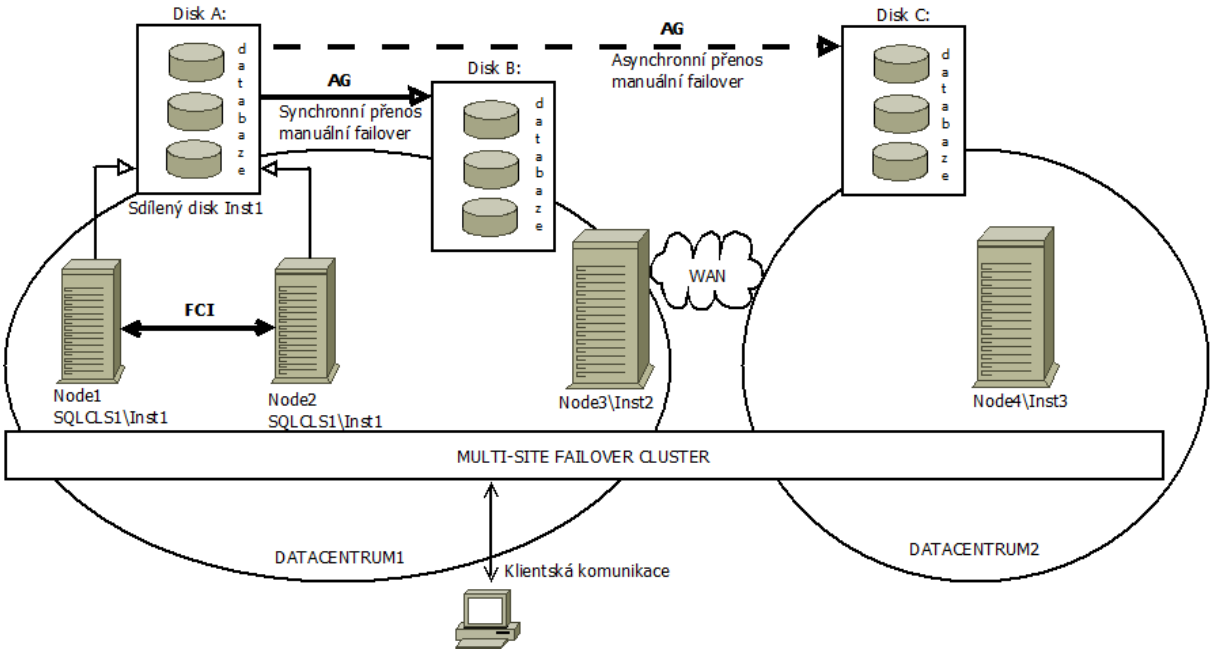
K dispozici máte:

- Datacentrum1, umístěné kousek od Prahy se zajištěnou vysokou bezpečností. Prakticky s neomezenými možnostmi v rámci systémových prostředků.
- Datacentrum2, umístěné v Indii.

- Neomezené množství použití software licencí.

2.3.2. Návrh řešení

Pro navrhované řešení máme neomezené možnosti, tudíž využijeme výhod nejnovějších technologií. Na obr. 2.4 je zobrazeno navrhované řešení použití technologií vysoké dostupnosti v datacentrech.



Obr. 2.4: Scénář 3, schéma řešení vysoké dostupnosti v prostředí datových center.

Celé řešení je postaveno na multi-site failover cluster[7], který je vytvořen mezi dostupnými datacentry. Klienti se připojují přímo na role tohoto clusteru, který pak směruje komunikaci na určitou instanci SQL Serveru dle dostupnosti.

Vysoká dostupnost je řešena následovně:

- V prvním datacentru, uzel Node1 a Node2 tvoří FCI s automatickým přepnutí služeb v případě selhání.
- V prvním datacentru je vytvořena AG, kde primární replika je vytvořené FCI a sekundární replika je uzel Node3. Mezi FCI a Node3 je nastaven synchronní přenos data a manuální falover.
- AG vytvořené v prvním datacentru zasahuje i do druhého datacentra v Indii, kde primární databáze je FCI a sekundární uzel Node4. Mezi FCI a Node je nastaven asynchronní přenos data a manuální falover.

Možnosti obnovy dat v případě havárie:

1. Pokud selže jeden z uzlů Node1 nebo Node2 ve FCI, úlohu primárního serveru převezme vždy funkční uzel z této dvojice.
2. Pokud však selže Node1 a Node2 zároveň nebo sdílený disk, na kterém jsou uloženy databáze, musí zasáhnout administrátor a přepnout služby na sekundární repliku Node3\Inst2.

[7] Multi-site failover cluster je skupina clusterovaných uzlů napříč více sítěmi, propojených pomocí LAN/WAN [11].

3. V případě, že selže datacentrum kousek od Prahy, musí zasáhnout administrátor, který přepne služby do Indie, tedy na druhé datacentrum.

Dlouhodobý výpadek v tomto řešení vysoké dostupnosti je krajně nepravděpodobný.

3. Porovnání MS SQL s produkty technologií vysoké dostupnosti od jiných firem

Mezi největší dodavatele řešení vysoké dostupnosti v databázovém řešení, patří firma Oracle a Microsoft, z toho důvodu se primárně budeme zabývat těmito technologiemi. Jelikož jsme řešení vysoké dostupnosti v prostředí Microsoft SQL Server již probrali, tato kapitola se bude zabývat popisem základních funkcí vysoké dostupnosti Oracle databáze a v následném porovnání Oracle funkcí a Microsoft technologií.

3.1. Funkce vysoké dostupnosti Oracle databáze

Oracle poskytuje následující funkce pro vysokou dostupnost [12]:

Fast-Start Fault zotavení
Rychlé předvídatelné zotavení ze systémových poruch a poruch databáze. Automaticky ohraničuje čas zotavení databáze při spuštění pomocí vlastních laděných kontrolních bodů.

Oracle Real application cluster a Oracle Clusterware
Oracle Real Application cluster (RAC Oracle) a Oracle Clusterware umožňuje spustit libovolnou aplikaci přes sadu cluster serverů. Tato funkce zajišťuje nejvyšší úroveň dostupnosti a nejvíce flexibilní škálovatelnost. Pokud clusterovaný server selže, oracle databáze pokračuje v běhu na přežívajících serverech. Pokud je potřeba více výpočetního výkonu, můžete přidat další server bez přerušení přístupu k datům.

Oracle RAC
Umožňuje několik instancí, které jsou propojeni pomocí vnitřního připojení ke sdílenému přístupu k databázi Oracle. Oracle databáze běží na dvou nebo více systémech v clusteru při současném přístupu k jediné sdílené databázi. Výsledkem je jediný databázový systéme, který se klene nad více hardwarovými systémy, což umožní Oracle RAC zajistit vysokou dostupnost a redundanci při selhání v clusteru. Oracle RAC pojme všechny typy systému, od datových skladů pouze pro čtení až k náročně aktualizovaným online zpracování transakcí (OLTP) systémům.

Oracle Clusterware
Oracle Clusterware je přenosný cluster software, který umožňuje shlukování nezávislých serverů tak, aby spolupracovali jako jediný systém.

Oracle Data Guard
Řešení pro ochranu dat, dostupnost dat a zotavení po havárii. Umožňuje správu, monitorování a automatizaci software k vytváření a údržbě jedné nebo více pohotovostních databází pro ochranu dat v případě selhání a zároveň poskytuje vysokou dostupnost pro kritické aplikace.

Oracle Streams
Oracle Streams je velmi flexibilní a výkonná databázová funkce k implementaci replikací, transformaci dat a řízení front zpráv.

Oracle Flashback Technology
Oracle Flashback technologie poskytuje sadu funkcí k přepínání pohledů dat v různých časových bodech. Pomocí těchto funkcí se můžete dotázat na historická data a schémata objektů. Umožňuje také provádět analýzu změn a samoobslužnou opravu logických chyb zatímco je databáze online

Automatic Storage Management (ASM)
ASM poskytuje vertikálně integrovaný systém souborů a správy svazků přímo v jádře Oracle, což má za následek:

- Podstatně méně práce s ustanovením úložiště databáze.
- Vyšší úroveň dostupnosti.
- Eliminace nákladů pro instalaci a údržbu specializovaných produktů pro ukládání dat.
- Jedinečné funkce pro databázové aplikace.

Recovery Manager (RMAN)
Recovery Manager je Oracle nástroj pro správu zálohování a obnovy databáze. RMAN eliminuje provozní složitost a zároveň poskytuje vynikající výkon a dostupnost databáze.

Oracle Secure Backup
Oracle Secure Backup je centralizované řešení pro správu zálohování na pásky. Poskytuje kompletní páskové zálohovací řešení pro vaše IT prostředí.

Data recovery Advisor
Data recovery Advisor je funkce, která automaticky diagnostikuje selhání dat (na disku), představuje vhodné možnosti opravy a na vaši žádost spustí opravu dat.

Flash Recovery Area
Flash Recovery Area je jednotné místo skladování pro všechny soubory a činnosti obnovy související s Oracle databází.

Oracle Securitiy Features
Široká škála bezpečnostních nástrojů k řízení přístupu aplikačním datům.

LogMiner
LogMiner je plně relační nástroj ke zpracování logových souborů.

Hardware Assisted Resilent Data (HARD) Initiative
HADR initiative je iniciativou mezi společností Oracle a dodavateli hardware k zabránění poškození dat před zapsání na disk.

Data Block Corruption Prevention and Detection Parameters
Databázové součásti a pomůcky k detekci a předcházení některých korupcí a ztrátě zápisu.

3.2. Porovnání technologií vysoké dostupnosti v prostředí Oracle a MS SQL

V tab. 3.1 je shrnutí nejdůležitější funkcí vysoké dostupnosti a podpora jednotlivých prostředí.

Tab. 3.1: Porovnání technologií vysoké dostupnosti v prostředí Oracle a MS SQL.

	Oracle	MS SQL
Podporované platformy	Všechny Unix/Linux certifikované platformy Windows Server	Windows Server
Není potřeba sdílené uložiště	Ano	Ano
Replikace dat a služeb	Ano	Ano
Které komponenty jsou replikovány	Instance a databáze	Pouze databáze
Automatický failover	Ano (Observer & Fast-Start Failover)	Ano (Availability Group)
Automatická obnova ze sekundární instance v případě selhání	Ano (reinstate)	Ano (Availability Group)
Ochrana při dvou primárních systémech	Ano (Observer & Fast-Start Failover)	Ano (Kvórum z cluster funkcí)
Složitost implementace	střední	jednoduchá
SYNC/ASYNC přenos dat	Ano	Ano
Sekundární systém v režimu čtení pro vytváření sestavy	Ano (ale s licencí Active Guard)	Ano
Sekundární systém v režimu čtení/zápis pro testování aplikace	Ano (Snapshot Standby)	Ne

Rozdíly mezi Oracle a MS SQL jsou nepatrné, jak můžete vidět v tab. 3.1. Je třeba si však uvědomit, že Oracle představil tohle řešení již v roce 2005, zatímco uspokojivé řešení MS SQL bylo oficiálně uvedeno až ve verzi 2012.

4. Testovací prostředí

Pro testovací prostředí je vhodné využít výhod virtualizace, která umožňuje nad jedním fyzickým serverem vytvořit více virtuálních serverů. Operační systému Windows Server 2012 R2 v edici Standard nebo Datacenter (v dřívějších verzích Windows Server Enterprise) nainstalovaný na fyzickém serveru má vestavěnou funkci Hyper-V, kde je možné virtuální stroje vytvářet a jednoduše konfigurovat. Tímto vcelku jednoduchým způsobem vytvoříme testovací prostředí.

4.1. Popis testovacího prostředí

K dispozici jsou tři virtuální servery (VM) NodeA, NodeB, NodeC s následnou konfigurací:

- Operační systém: Windows Server 2012 r2 Evaluation Edice[8].
- RAM: NodeA 8196 GB, NodeB a NodeC 4096 GB.
- HDD: NodeA a NodeB 127 GB, NodeC 256 GB.
- Veřejný a privátní síťový adaptér:
    - NodeA: 192.168.54.160 veřejná IP adresa, 192.168.100.1 privátní IP adresa.
    - NodeB: 192.168.54.154 veřejný IP adresa, 192.168.100.2 privátní IP adresa.
    - NodeC: 192.168.54.171 veřejný IP adresa, 192.168.100.3 privátní IP adresa.

4.2. Ne-clusterované prostředí

Pro instalaci SQL Serveru na virtuálních serverech NodeA, NodeB a NodeC použiji instalační médium Microsoft SQL Server 2014 evaluation edici, která je volně dostupná na stránkách Microsoftu. Po spuštění instalačního média zvolím instalaci SQL Serveruv ne-clusterovém prostředí. Pomocí průvodce nainstaluji SQL Server na všech virtuálních serverech. Doporučené volby při instalaci SQL Serveru:

- Při výběru funkcí vybírat z důvodu optimalizace výkonu jen funkce, které budete potřebovat. Určitě by neměli chybět: všechny položky Database Engine a management nástroje.
- Pokud na instalovaném serveru budete instalovat jednu instanci SQL Serveru, je vhodné nastavit pojmenování instance defaultně. Instance si tak převezme jméno serveru nastavené v operačním systému.
- Při nastavení servisních účtů pro jednotlivé funkce je doporučeno použít, pokud instalujete v doméně, doménový účet určený pouze pro SQL služby, který je stále platný a má roli doménového administrátora.

Po dokončení instalace je dále nutné zkontrolovat a případně nastavit firewall na všech serverech. SQL služby komunikují přes port 1433.

Pro testovací účely byla dodána konzultantem diplomové práce testovací databáze přibližně o velikosti 5 GB.

[8]Evaluation edice je zkušební verze standardní edice, která je časově omezená.

4.3. Windows Server Cluster prostředí

Windows server cluster je skupina nezávislých počítačových systémů, známých jako uzly (node), které společně fungují jako jeden systém, s cílem zajistit klientům dostupnost a škálovatelnost zvláště důležitých aplikací a prostředků např. MSSQL. Uzly clusteru spolu neustále komunikují pomocí periodických zpráv nazývaných prezenční signály. Přestane-li být některý z uzlů clusterů k dispozici z důsledku selhání či údržby, začne službu okamžitě poskytovat jiný uzel (proces je známý jako převzetí služeb při selhání – failover). Existují dva typy clusteru:

Cluster vyrovnání zatížení sítě (Network Load Balancing cluster):
Cluster vyrovnání zatížení sítě (NLB) rozdělí zátěž na různé uzly, které jsou součástí clusteru na základě předem stanovených pravidel. Klientská aplikace musí komunikovat na jedinou clusterovanou IP adresu, čímž klient nezjistí, který uzel v clusteru zpracovává žádost. NLB pomáhá zvýšit dostupnost a škálovatelnost aplikace.

Failover Cluster:
Windows Server Failover Cluster (WSFC) poskytuje funkce podporující vysokou dostupnost a scénáře obnovy po havárii na hostovaných serverových aplikacích, jako je například Microsoft SQL Server nebo Microsoft Exchange. Pokud uzel nebo služba clusteru selžou, může služby, které byly hostitelem tohoto uzlu, automaticky nebo manuálně převést na jiné dostupné uzly, v procesu známém jako failover, tedy převzetí služeb při selhání [3].

Doporučení pro technologie WFSC:

- Mít k dispozici dva a více uzlů s operačním systémem, který podporuje failover clustering. Aktuální verze Windows Serveru 2012 R2 podporuje failover clustering v edicích Standard a Datacenter. Zdarma, po dobu 180 dní, lze využít pro testovací účely edici Windows Server 2012 R2 Evaluation.
- Minimálně jeden lokální disk pro OS a SQL binární soubory, a další disk jako aplikační. Disky by měly být minimálně v RAIDU 1.
- Každý uzel musí být připojen k externímu sdílenému úložišti. Na clusterované instanci SQL Serveru se databáze ukládají na sdílené úložiště, což znamená, že všechny uzly v clusteru jsou fyzicky připojeny k diskovému poli. Tato konfigurace umožňuje v případě selhání serveru převedení služeb na jiný uzel v clusteru.
- Minimálně dvě síťové karty. Obvykle se jedna síťová karta používá pro veřejnou síť, která slouží k přístupu do intranetu nebo internetu a druhá síťová karta zase pro komunikaci v clusteru, tedy v privátní síti.

Uzly ve WSFC poskytují následující [3]:

- **Distribuovat metadata a oznámení.**
  WSFC služby a hostované aplikace metadat jsou udržovány v každém uzlu clusteru. Tato metadata zahrnují konfiguraci WSFC a stavy nastavení hostovaných aplikací. Změny metadat nebo stavů uzlu jsou automaticky přeneseny na ostatní uzly v clusteru.
- **Řídit prostředky clusteru.**
  Jednotlivé uzly v clusteru mohou poskytnout fyzické prostředky, jako je například přímé připojení úložiště, síťové rozhraní a přístup ke sdílenému diskovému prostoru. Hostované

aplikace se registrují jako prostředek clusteru, umožňují nastavit spuštění a závislost na jiné prostředky.

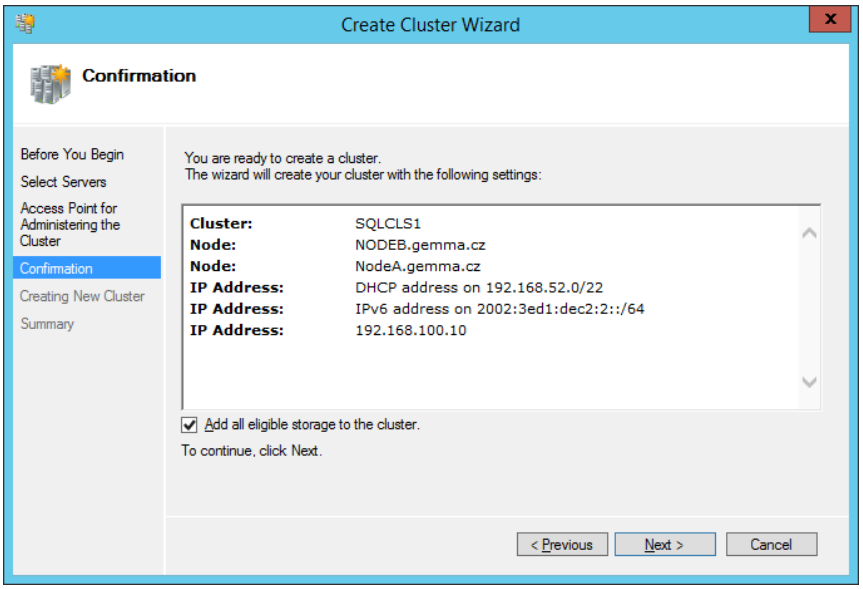
- **Sledovat stav clusteru.**
  „Zdravé“ detekce vnitřního a primárního uzlu je dosaženo kombinací srdečního stylu komunikace v síti a monitorováním prostředků. Celkový „zdravotní“ stav clusteru je určen hlasy kvóra uzlů v clusteru.
- **Failover koordinaci.**
  Každý prostředek je nakonfigurován tak, aby mohl být umístěn na primárním uzlu, a každý z nich může být automaticky nebo manuálně převeden na jeden nebo více sekundárních uzlů. „Zdravotní“ politika failover řídí automatický převod zdrojů mezi uzly. Uzly a hostované aplikace jsou upozorněny, pokud dojde k převzetí služeb při selhání, aby mohly vhodně reagovat.

4.3.1. Instalace a nastavení failover clusteru

Pro instalaci failover cluster je nutné mít připravené sdílené úložiště. Jelikož máme omezené možnosti a nemáme žádné vzdálené úložiště, vytvoříme si jej z NodeC.

Následující postup vás provede vytvořením sdíleného úložiště, vytvořením a konfiguraci failover clusteru:

1) Na NodeA a NodeB nainstalujeme roli failover clustering.
2) Spustíme nástroj Failover Cluster Manager.
3) Pomocí průvodce vytvoříme failover cluster. Nastavení clusteru je shrnuto v obr. 4.1.

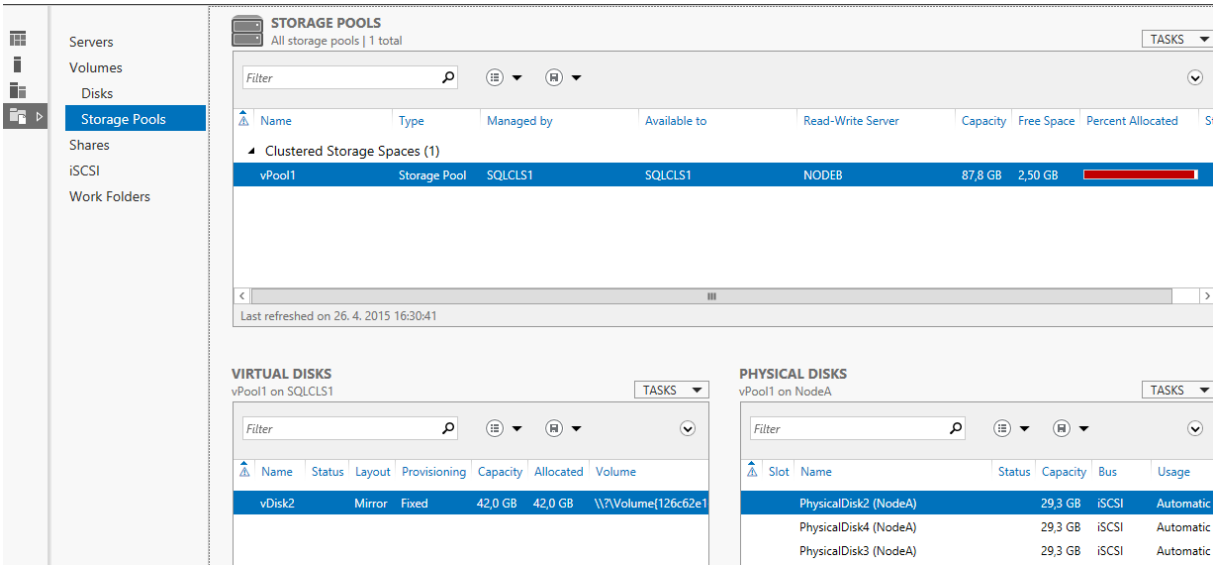


Obr. 4.1: Průvodce vytvořením Clusteru.

4) Jelikož není k dispozici sdílené diskové úložiště, využijeme pro testovací účely pevný disk počítače NodeC, na kterém vytvoříme čtyři iSCSI virtuální disky:
   - vDisc1 s kapacitou 100 GB.
   - vDisc2, vDisc3 a vDisc4 s kapacitou 30 GB.
5) Přiřadíme oprávnění k přístupu na nově vytvořené virtuální iSCSI disky pro počítače NodeA a NodeB.

6) Na NodeA a NodeB pomocí iSCSI iniciátoru inicializujeme vytvořené iSCSI virtuální disky z NodeC.
7) Z iSCSI virtuálních disků 2,3 a 4 vytvoříme fond úložišť (storage pool) vPool1.
8) Z fondu úložišť vPool1 vytvoříme pomocí průvodce virtuální disk vDisc5, který nastavíme do mirroru s maximální možnou kapacitou. Z něj pak vytvoříme pomocí průvodce svazek disku vDisk2.

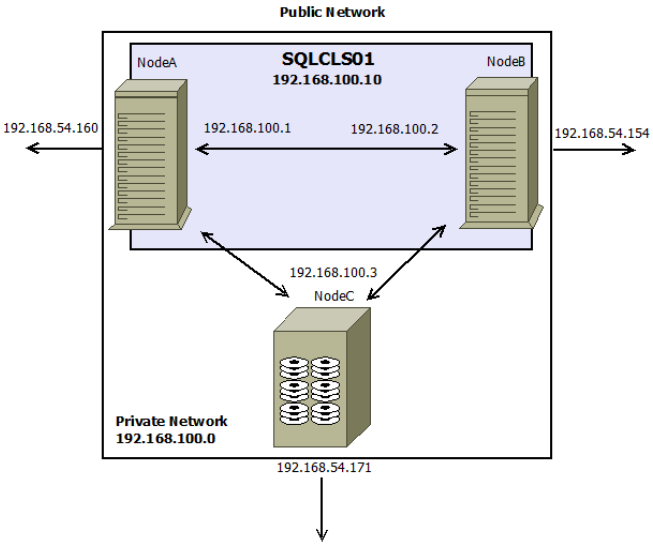
Konfigurace fondu úložišť a disků je znázorněna na obr. 4.2.



Obr. 4.2: Konfigurace fondu úložišť a disku.

9) Vytvořený svazek disku vDisk2 spolu s iSCSI virtuálním diskem vDisc1 přidáme pomocí nástroje Failover Cluster Manager do clusteru.
10) Po přidání disku do clusteru nakonfigurujeme vDisk2 jako svědka kvóra.

Testovací clusterované prostředí je připravené pro instalaci SQL Server Failover Cluster Instanci. Na obr. 4.3 je vyobrazeno schéma sítě testovacího clusterovaného prostředí.



Obr. 4.3: Schéma sítě clustrovaného testovacího prostředí.

Závěr

Hlavním cílem diplomové práce *Vysoká dostupnost v prostředí MS SQL Serveru* bylo seznámit čtenáře s technologiemi vysoké dostupnosti v prostředí Microsoft SQL Serveru. Čtenář tak získá znalosti o dostupných technologiích vysoké dostupnosti v prostředí MS SQL Serveru a na základě získaných znalostí, by měl být schopný určit optimální variantu řešení vysoké dostupnosti ve svém prostředí.

V první kapitole diplomové práce jsou detailně vysvětleny všechny dostupné technologie vysoké dostupnosti v prostředí MS SQL Server. U každé technologie je uveden návod pro nasazení technologie. Jednotlivé technologie vysoké dostupnosti jsou rozebrány z pohledu bezpečnosti. Jsou určeny výhody a nevýhody proč jednotlivou technologii použít v praxi. Je určena optimální varianta využití dané technologie. Technologie vysoké dostupnosti jsou mezi sebou porovnány v přehledné tabulce. A v poslední řadě je uvedena dostupnost jednotlivých technologií ve verzích MS SQL Serveru.

V další kapitole jsou vytvořeny tři scénáře pro využití technologií vysoké dostupnosti v praxi. Scénář je tvořen formou fiktivního výběrového řízení, kde firma, stejně jako v praxi, určí požadavky na výsledné řešení vysoké dostupnosti. Na základě požadavků zákazníka, je v každém scénáři uveden myšlenkový postup, jakým způsobem vypracovat nabídku a navrhnout tak nejlepší možné řešení vysoké dostupnosti.

Jelikož na trhu existuji i jiné řešení vysoké dostupnosti, od jiných firem. V další kapitole je uveden rozbor funkcí vysoké dostupnosti v prostředí Oracle a následně jsou porovnány funkce vysoké dostupnosti v prostředí Oracle, s technologiemi vysoké dostupnosti v prostředí MS SQL Serveru.

V poslední kapitole diplomové práce je rozebráno testovací prostředí poskytnuté vedoucím diplomové práce. V testovacím prostředí jsou uvedeny použité licence, hardware a způsob realizace. Prostředí je rozděleno do dvou sekcí. První sekce je prostředí Windows Server bez použití clusteru, kde byly aplikovány následující technologie vysoké dostupnosti: zrcadlení databáze, replikace a odesílání souboru protokolu. Druhá sekce je prostředí Windows Server s použitím clustru, které primárně sloužilo pro testování technologie Failover Clustering Instance. Pro testování technologie Availability Group byly použity obě sekce.

Doposud technologie vysoké dostupnosti v prostředí MS SQL Serveru nebyly systematicky zpracovány, a jelikož si to praxe vyžadovala, vznikla tato diplomová práce, která je průvodcem technologiemi vysoké dostupnosti v prostředí MS SQL Serveru a bude nadále využívaná ve společnosti, ve které pracuji, k určení optimální varianty řešení vysoké dostupnosti v prostředí MS SQL Server pro zákazníky.

Seznam literatury

[1] WALTERS, E. Robert. Mistrovství v Microsoft SQL server 2008. 1. vyd. Brno: ComputerPress,2009. 864 s. ISBN 978-80-251-2329-4.

[2] Internetové stránky technických materiálů firmy Microsoft. High availability (Database Engine) [online],
[cit. 25. 10. 2014]
URL: <http://technet.microsoft.com/cs-cz/library/bb522583(v=sql.100).aspx>

[3] Internetové stránky technické knihovny firmy Microsoft. High availability solution (SQL Server) [online],
[cit. 30. 11. 2014]
URL: <http://msdn.microsoft.com/en-us/library/ms190202.aspx>

[4] K. BURDA, Dokument předmětu bezpečnost informačních systémů, Zálohování dat.

[5] Internetové stránky technické knihovny firmy Microsoft. SQL Server Agent [online],
[cit. 5. 12. 2014]
URL: < https://msdn.microsoft.com/en-us/ms189237.aspx>

[6] Internetové stránky zabývající se úložišti, SearchStorage. Storage area network [online],
[cit. 5. 12. 2014]
URL: <http://searchstorage.techtarget.com/definition/storage-area-network-SAN>

[7] Internetové stránky technických materiálů firmy Microsoft. Windows PowerShell [online],
[cit. 8. 12. 2014]
URL: <https://technet.microsoft.com/en-us/library/cc732114(v=ws.10).aspx>

[8] Internetové stránky zabývajícím se médii, MediaGuru. Slovník POS/POP [online],
[cit. 12. 12. 2014]
URL: <http://www.mediaguru.cz/medialni-slovnik/pos-pop/>

[9] Internetové stránky technických materiálů firmy Microsoft. Publication Access Lists [online],
[cit. 12. 12. 2014]
URL: <https://technet.microsoft.com/en-us/library/aa256150(v=sql.80).aspx>

[10] Internetové stránky technických materiálů firmy Microsoft. BitLocker [online],
[cit. 14. 3. 2015]
URL: <https://technet.microsoft.com/cs-cz/library/hh831713.aspx>

[11] Internetové stránky technického blogu firmy Microsoft. Microsoft Windows Multi-Site Failover Cluster [online],
[cit. 9. 5. 2015]
URL: <http://blogs.technet.com/b/meamcs/archive/2013/11/09/microsoft-windows-multi-site-failover-cluster-best-practices.aspx>

[12] Internetové stránky technických materiálů firmy Oracle. Database High Availability [online],
[cit. 16. 5. 2015]
URL: <http://docs.oracle.com/cd/B28359_01/server.111/b28281/toc.htm>

Seznam symbolů a zkratek

AG	Availability Group
ASM	Automatic Storage Management
CD	Compact Disk
CRM	Customer relationship management
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DVD	Digital Versatile Disc
FCI	Failover Cluster Instance
FFA	Field Force Automation
GFS	Grandfather Father Son
HARD	Hardware Assisted Resilent Data
IP	Internet Protocol
IPSec	Internet Protokol Security
iSCSI	internet Small Computer System Interface
LAN	Local Area Network
MS	Microsoft
NAS	Network Attached Storage
OLTP	Online Transaction Processing
OS	Operation System
POS	Point Of Sale
RAC	Real Application Cluster
RMAN	Recovery Manager
SAN	Storage Area Network
SFA	Sale Force Automation
SQL	Structured Querry Language
SSL	Secure Socket Layer
SSMS	SQL Server Management Studio
VM	Virtual Machine
VPN	Virtual Private Network
WSFC	Windows Server Failover Clustering