

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

**РЕАЛІЗАЦІЯ ЗАСОБУ ЗАБЕЗПЕЧЕННЯ БЕЗПЕРЕРВНОСТІ  
БІЗНЕС-ПРОЦЕСІВ ПРИ ВИКОРИСТАННІ СИСТЕМИ  
КЕРУВАННЯ БАЗАМИ ДАНИХ MS SQL SERVER**

**Текстова частина до курсової роботи  
за спеціальністю “Комп’ютерні науки”**

Керівник курсової роботи  
к.т.н. Ющенко Ю.О

---

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

Виконав студент  
Задорожний М. В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

Київ 2022

Міністерство освіти і науки України  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»  
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ  
Зав.кафедри інформатики,  
проф., д.ф.-м.н.  
\_\_\_\_\_ М. М. Глибовець  
(підпис)  
„\_\_\_\_\_” \_\_\_\_\_ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ  
на курсову роботу

студенту Задорожному Максиму факультету інформатики 4 курсу  
ТЕМА: Реалізація засобу забезпечення безперервності бізнес-процесів при  
використанні системи керування базами даних MS SQL Server  
Вихідні дані:

- 
- 

Зміст ТЧ до курсової роботи:

- Індивідуальне завдання
- Вступ
- 1 Порівняння з альтернативними методами
- 2 Використані технології
- 3 Проектування і розробка
- Висновки
- Список літератури
- Додатки

Дата видачі „\_\_\_\_\_” \_\_\_\_\_ 2022 р. Керівник \_\_\_\_\_

(підпис)

Завдання отримав \_\_\_\_\_

(підпис)

**Тема:** Реалізація засобу забезпечення безперервності бізнес-процесів при використанні системи керування базами даних MS SQL Server

**Календарний план виконання роботи:**

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Розгляд та аналіз проблеми	Вересень - грудень 2021р.	
2.	Розробка архітектури та програми аналіз технологій	Січень- лютий 2022р	
3.	Розробка back-end частини	Лютий- березень 2022р.	
4.	Розробка front-end частини	Березень-квітень 2022р.	
5	Написання текстової частини	Квітень 2022р.	
6.	Перегляд праці науковим керівником	Травень 2022р.	
7.	Підготовка до презентації роботи	Травень 2022р	
8.	Презентація курсової роботи	Травень 2022р.	

Студент Задорожний М.В. \_\_\_\_\_

Керівник Ющенко Ю.О. \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

<b>АНОТАЦІЯ</b>	<b>5</b>
<b>ВСТУП</b>	<b>6</b>
<b>РОЗДІЛ 1: Порівняння з альтернативними методами</b>	<b>8</b>
1.1. Час та точка відновлення	8
1.2. Стратегії аварійного відновлення для SQL Server	9
1.3. Резервне копіювання та відновлення	9
1.4. Log Shipping (Доставка журналів)	9
1.5. Mirroring	11
1.6. AlwaysOn availability groups	13
1.7. Висновки до Розділу 1	16
<b>РОЗДІЛ 2: Використані технології</b>	<b>17</b>
2.1. Hyper-V	17
2.2. Windows Server 2016	17
2.3. MS SQL Server Management Studio (SSMS)	18
2.4. Windows Server Failover Clustering (WSFC)	18
<b>РОЗДІЛ 3: Проектування і розробка</b>	<b>20</b>
3.1. Налаштування серверів у Hyper-V	20
3.2. Налаштування Windows Server 2016	20
3.3 Кворум	22
3.4. Налаштування AlwaysOn	23
3.5. Демонстрація роботи	30
3.6. Висновки до розділу 3	33
<b>Висновки</b>	<b>34</b>
<b>Список використаної літератури:</b>	<b>36</b>

## АНОТАЦІЯ

Одне із загальних завдань при розгортанні SQL Server полягає в тому, щоб забезпечити доступність всіх критично важливих екземплярів SQL Server і баз даних у них кінцевим користувачам відповідно до їх потреб — хоч винятково в робочі години, хоч цілодобово. Потрібно підтримувати працездатність бізнесу з мінімальними простоями або без них. Ця концепція також називається безперервністю бізнес-процесів.

Метою цієї роботи є реалізація засобу забезпечення безперервності бізнес-процесів при використанні системи керування базами даних MS SQL Server. А саме налаштування серверів баз даних таким чином, щоб забезпечувалась реалізація високої доступності та аварійного відновлення при збоях в роботі MS SQL Server.

## ВСТУП

Будь-які збої в роботі служб баз даних — чи то через стихійне лихо, поломку обладнання, кібератаки чи інші причини — можуть перешкодити організації виконувати рутинні операції та вести повсякденний бізнес. Це може призвести до незадоволених користувачів, втрати прибутку та зіпсованої репутації.

Організація повинна мати ефективну стратегію аварійного відновлення, яка допомагає мінімізувати збої в роботі служб SQL Server, особливо при підтримці критично важливих робочих навантажень.

Програми, які покладаються на SQL Server, повинні мати доступ до даних, щоб відповідати вимогам робочого навантаження, але неочікувані події можуть призвести до того, що бази даних стануть недоступними або призвести до втрати даних. Збої в роботі служб передачі даних можуть виникати з багатьох причин, у тому числі з таких:

- користувач ненавмисно видаляє важливі дані;
- атака зловмисного програмного забезпечення шифрує або знищує дані;
- працівник проливає каву на систему зберігання;
- збій живлення призводить до пошкодження даних;
- землетрус руйнує фізичну інфраструктуру зберігання;
- жорсткий диск (HDD) виходить з ладу, і всі дані на цьому диску втрачені;
- адміністратор ненавмисно форматує жорсткий диск, який містить активні дані;
- пошкодження програмного забезпечення призводить до втрати даних або їх відсутності;
- службовець-шахрай викрадає фізичний накопичувач;

Це далеко не єдині причини, через які дані можуть стати недоступними, але вони ілюструють широкий спектр криз, які можуть вразити будь-яку організацію в будь-який час.

Якою б не була причина, єдиний спосіб захиститися від втрати даних і збоїв у роботі служби — це запровадити стратегію аварійного відновлення, яка мінімізує час простою та гарантує безпечне відновлення будь-яких пошкоджених даних. На щастя, SQL Server має функції, які можуть допомогти захистити від катастрофи, багато з яких є частиною можливостей високої доступності платформи.

Висока доступність (HA) – це рішення \ процес \ технологія для забезпечення доступності програми \ бази даних в режимі 24×7 при планових або позапланових вимкненнях. В основному, в MS SQL Server є декілька варіантів досягнення \ налаштування рішення високої доступності для баз даних.

Забезпечення безперервності бізнес процесорів при використанні системи керування базами даних MS SQL Server є актуальною завжди. Особливо після початку повномасштабної війни Росії проти України ЦОД будь-якого підприємства по всій території України може бути уражений ракетами, а ЦОД-и, розміщені в окремих областях України, можуть бути захоплені агресором.

Тому саме в наш час особливої актуальності набувають стійкість підприємства до вищевказаних ризиків, спроможність здолати інциденти різної складності та швидко відновити критичні функції, здатність знов зробити доступними для клієнтів свої сервіси.

## РОЗДІЛ 1: Порівняння з альтернативними методами

### 1.1. Час та точка відновлення

Основною метою плану аварійного відновлення є забезпечення безперервності бізнесу в разі катастрофи. Однак не всі плани однакові і не повинні бути такими. План аварійного відновлення повинен бути розроблений відповідно до конкретних вимог організації щодо захисту даних.

Наприклад, дані, які керують критичною фінансовою програмою, можуть вимагати негайного відновлення, але дані, які використовуються для створення щомісячних звітів про продажі, можуть витримувати більш тривалі затримки. Плануючи стратегію аварійного відновлення, організація повинна визначити наступні три показники:

- Ціль часу відновлення (RTO). Це максимальна кількість часу, протягом якого програма може перебувати в автономному режимі через недоступність даних. Показник визначає, як швидко дані повинні бути в мережі після інциденту.
- Ціль точки відновлення (RPO). Це допустима кількість втрати даних, яку можна допустити, якщо станеться подія. Цей показник часто розглядають з точки зору часу. Наприклад, якщо резервне копіювання бази даних виконується кожні дві години і катастрофа сталася безпосередньо перед наступним запланованим резервним копіюванням, усі зміни даних, які відбулися з моменту останнього резервного копіювання, будуть втрачені.

Зрештою, організації потрібно збалансувати ризик втрати даних із витратами на реалізацію стратегії аварійного відновлення. Чим коротші RTO і RPO (з точки зору часу), тим більші витрати.

В цій роботі я не звертаю увагу на витрати, а зосереджуюся на тому, щоб час та точка відновлення були якомога коротшими.

## 1.2. Стратегії аварійного відновлення для SQL Server

Після того, як організації визначають потрібний рівень захисту, вони можуть скористатися перевагами наведених нижче функцій SQL Server, щоб пом'якшити вплив неочікуваних збоїв у роботі служби.

## 1.3. Резервне копіювання та відновлення

Добре перевірена стратегія резервного копіювання та відновлення може допомогти захистити бази даних від різноманітних проблем, зокрема від програм-вимагачів, помилок користувача, збоїв обладнання та стихійного лиха. Резервне копіювання даних в окремому місці дає змогу відновити базу даних до узгодженого стану та уникнути катастрофічної втрати даних; однак відновлення баз даних може бути трудомістким процесом, і зміни даних, внесені після останньої резервної копії, будуть втрачені. Резервне копіювання стало особливо важливим із зростанням атак програм-вимагачів.

Крім того, резервні копії бази даних корисні для рутинних адміністративних цілей, таких як копіювання бази даних з одного сервера на інший, налаштування груп доступності Always On або дзеркального відображення бази даних, а також архівування.

Резервне копіювання потрібно виконувати завжди, оскільки воно є найбільш надійним методом збереження даних. Та є найбільш ефективним у зв'язці з іншими стратегіями відновлення.

## 1.4. Log Shipping (Доставка журналів)

Доставка журналів SQL Server забезпечує високу доступність і рішення для аварійного відновлення, а також працює з наявними резервними копіями журналів транзакцій. Процес включає в себе первинний і один або кілька вторинних серверів баз даних. Нижче наведені етапи високого рівня доставки журналів:

Резервне копіювання. створення регулярних резервних копій журналу транзакцій (t-log) з основного сервера.

Копіювання: копіювання файлу резервної копії t-log на вторинні сервери.

Відновлення: відновлення резервних копій журналу транзакцій на вторинних серверах без відновлення бази даних.

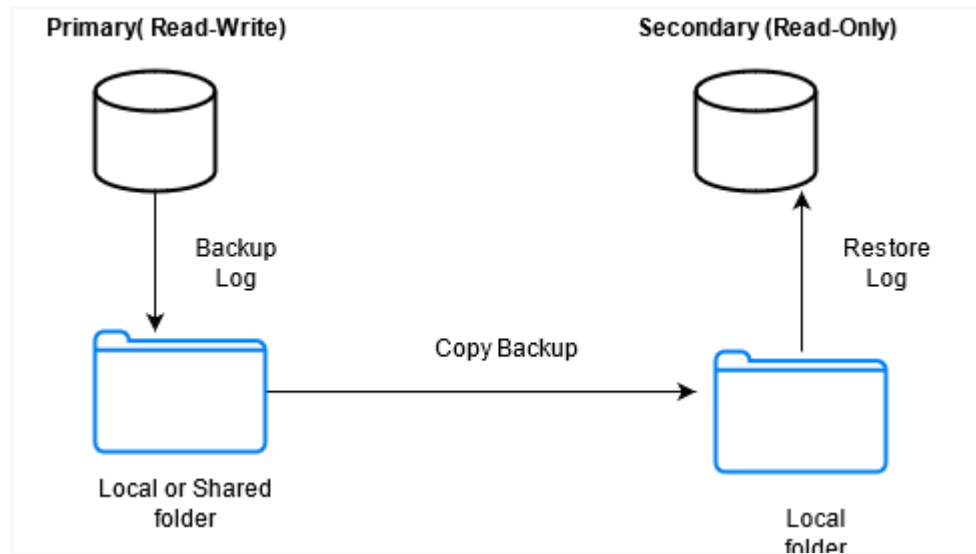


Рис 1.1 - Схема роботи Log Shipping

Доставка журналів реалізує процеси резервного копіювання, копіювання та відновлення за допомогою завдань агента SQL Server. Перше завдання (резервне копіювання) бере резервну копію t-log з первинної бази даних, друге завдання (копія) копіює цю резервну копію t-log на вторинні сервери, а третє завдання (відновлення) відновлює ці резервні копії t-log на вторинному. бази даних. Вторинна база даних залишається в режимі відновлення або очікування і чекає наступного резервного копіювання t-log з первинного сервера. Він не відновлює вторинні бази даних.

Додаткова база даних не синхронізується з основною базою даних у режимі реального часу. Натомість завжди існує затримка синхронізації, яка залежить від інтервалів резервного копіювання журналу транзакцій у первинній базі даних, тривалості копіювання та часу відновлення.

## Переваги

- Доставка журналів працює з наявними резервними копіями t-log;
- Доставка журналів підтримується як на SQL Server у Windows, так і в Linux;
- Ви можете мати кілька вторинних баз даних;
- Реалізація проста;
- Ви можете налаштувати доставку журналів для кількох баз даних із вихідного екземпляра;
- Вам не потрібно налаштовувати кластеризовані екземпляри;

## Недоліки

- Відмова(Failover) вручну;
- Додаткова база даних завжди відстає від первинної бази даних;
- Після відмовлення вручну програмі потрібно змінити рядок підключення, щоб він вказував на вторинний сервер;
- Якщо ви зчитуєте дані з вторинної бази даних, вони не будуть доступні, поки триває відновлення резервної копії t-log;

## 1.5. Mirroring

Дзеркальне відображення бази даних — це рішення, спрямоване на підвищення доступності бази даних SQL Server. Дзеркальне відображення кожної бази даних здійснюється окремо і працює лише з тими базами даних, які використовують модель повного відновлення.

Дзеркальне відображення бази даних підтримує дві копії однієї бази даних, які повинні знаходитися на різних екземплярах сервера Компонент SQL Server Database Engine. Зазвичай такі екземпляри знаходяться на комп'ютерах, розташованих у різних місцях. Запускаючи дзеркальне відображення бази даних, встановлює зв'язок, відомий як сеанс дзеркального відображення бази даних між цими примірниками сервера.

Один екземпляр сервера надає базу даних клієнтам (основний сервер). Інший екземпляр виступає як сервер гарячої заміни або гарячого резерву (дзеркальний сервер) в залежності від конфігурації та стану сеансу дзеркального відображення. Синхронізація сеансу дзеркального відображення бази даних дозволяє отримати сервер гарячого резервування, що підтримує швидке відпрацювання відмови без втрати даних зафіксованих транзакцій. Якщо сеанс не синхронізований, дзеркальний сервер зазвичай працює як сервер «теплого» резервування (з можливою втратою даних).

Основний та дзеркальний сервери з'єднані та працюють спільно, будучи учасниками сеансу дзеркального відображення бази даних. Два учасники виконують у сеансі взаємодоповнюючі ролі: основну роль та дзеркальну роль.

У будь-який час один із учасників грає роль основного сервера, а інший учасник — дзеркального. Учасник, який має дзеркальну роль, називається дзеркальним сервером, яке копія бази даних є поточної дзеркальної базою даних. При розгортанні дзеркального відображення бази даних у робочому середовищі основна база є робочою базою даних.

Дзеркальне відображення бази даних полягає в тому, що в дзеркальній базі даних повторюються всі операції вставки, оновлення та видалення, що виробляються в основній базі даних. Для цього потік записів активних транзакцій з максимальною швидкістю відправляється на дзеркальний сервер.

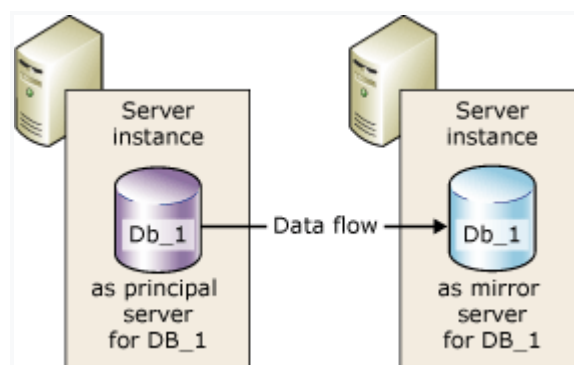


Рис 1.2 - Схема роботи Mirroring

- Можна використовувати його як для забезпечення високої доступності так і для аварійного відновлення.
- На його базі даних. Якщо у вас занадто багато баз даних, вам потрібно зробити це для всіх баз даних у екземплярі по черзі. Але він є більш гнучким, оскільки відмову можна здійснити на основі бази даних.
- Є дискове резервування. Кожен вузол використовує власні диски.
- Якщо ви налаштуєте Witness Server та налаштуєте його синхронно, відбувається автоматичне перемикання збоїв.
- Ви не можете читати або писати з вторинної бази даних. Він не буде доступний у пізніх версіях Microsoft SQL Server. Замість дзеркального відображення можна використовувати групу доступності Always On.
- Підтримує автоматичне відновлення сторінок. Приємна функція для DBA. Оскільки ця функція запобігає переходу бази даних у підозрілий режим.

## 1.6. AlwaysOn availability groups

SQL Server AlwaysOn — це рішення високого рівня доступності та аварійного відновлення за допомогою WSFC. Компоненти AlwaysOn є інтегрованими, гнучкими рішеннями, що підвищують доступність додатків, окупність вкладень в обладнання та спрощуюче розгортання систем високого рівня доступності та управління ними.

Примірники групи доступності AlwaysOn та екземпляри відмовостійкого кластера AlwaysOn використовують технологію платформи WSFC та реєструють компоненти як ресурси кластера WSFC. Пов'язані ресурси поєднуються в роль, яку можна зробити залежною від інших ресурсів кластеру WSFC. Потім кластер WSFC зможе виявляти необхідність перезапуску екземпляра SQL Server (і сигналізувати про цю необхідність), а

також автоматично виконувати відпрацювання відмови з переходом на інший серверний вузол у кластері WSFC.

Давайте розглянемо особливості групи доступності SQL Server Always On Availability Group (AG):

Захист від збою екземпляра SQL Server: Always On використовує декілька серверів, і кожен сервер має свій екземпляр та сховище SQL. Таким чином, програми можуть підключатися до вторинного екземпляра SQL після збою, якщо основний екземпляр SQL Server виходить з ладу.

Збої бази даних: група доступності — це набір баз даних користувачів, які повинні перемикатися разом.

Підключення клієнта: Always On використовує прослуховувач, який посилається на первинний екземпляр AG. Якщо відбувається відмовлення (ручне або автоматичне), прослуховувач автоматично вказує на нову основну репліку. Тому вам не потрібно змінювати рядок підключення, щоб зробити його придатним для запланованих і незапланованих відмов.

Гнучка політика відмов: SQL Server Always On використовує надійне виявлення збоїв за допомогою збереженої процедури `sp_server_diagnostics` для визначення серйозності збою, який може вплинути на екземпляри SQL Server.

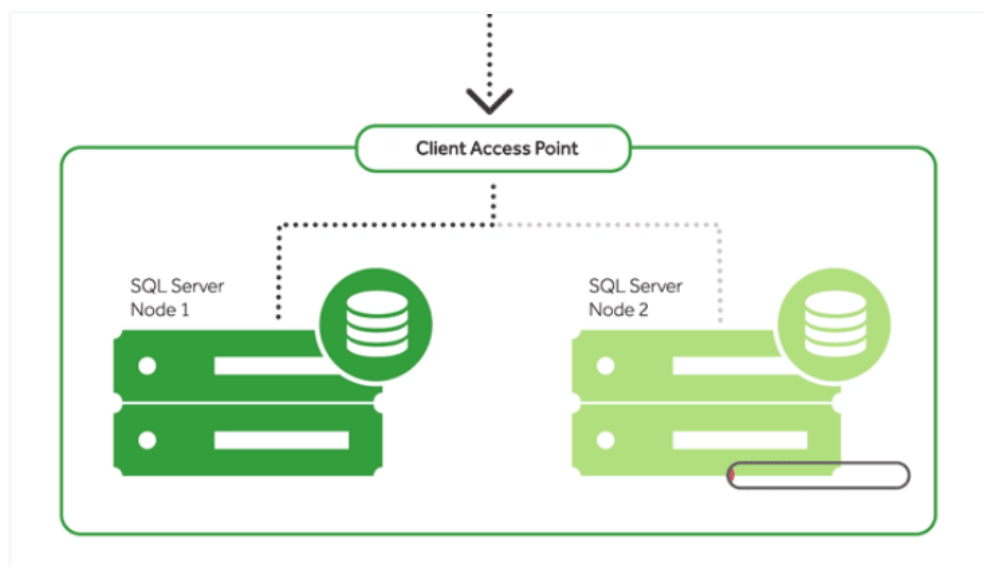


Рис 1.3 - Схема роботи AG AlwaysOn

### Переваги:

- SQL Server Always On можна використовувати для високої доступності та аварійного відновлення;
- Він підтримує повністю синхронізовану копію первинної бази даних без втрати даних;
- Ви можете розвантажити первинну базу даних, перемістивши запити лише для читання до вторинної репліки AG. Таким чином, ви можете використовувати резервну апаратну інфраструктуру для інших цілей, ніж аварійне відновлення. Однак зверніть увагу, що для цього потрібно мати ліцензію на SQL Server Enterprise Edition;
- Групи доступності підтримують обробники SQL Server, щоб уникнути модифікації рядка підключення після збою вторинної репліки. Таким чином, SQL Server надає додаткам єдину кінцеву точку бази даних;
- Групи доступності підтримують створення резервних копій журналу транзакцій і резервних копій лише для копіювання з вторинних баз даних;
- Він підтримує автоматичне відновлення сторінки з вторинної бази даних, що захищає від помилок підсистеми зберігання, які можуть пошкодити сторінку даних;
- Групи доступності підтримують автоматичне та ручне перемикавання збоїв;

### Недоліки

- Ви не можете захистити системні бази даних (головна, модель і MSDB) від збою на рівні екземпляра або бази даних. Завжди ввімкнено не підтримує додавання їх до груп доступності;
- Завжди ввімкнено не синхронізує логіни SQL Server, пов'язані сервери та завдання агента з вторинними базами даних;

- Ви не можете перевантажити повне резервне копіювання у вторинні бази даних, оскільки воно підтримує лише COPY\_ONLY резервне копіювання із вторинної бази даних;

### 1.7. Висновки до Розділу 1

Отже, проаналізувавши всі стратегії забезпечення високої доступності та аварійного відновлення, найбільше до моїх потреб реалізації неперервності бізнес процесів підходить стратегія AlwaysOn, оскільки:

- Керувати групою Always On Availability Group дуже легко.
- У групу доступності можна включити більше однієї бази даних.
- Є дискове резервування. Можна читати з вторинної бази даних, записуючи на основну.
- Ви можете переключити свою групу доступності на інший сервер без відчуття перерви перерви.
- Оскільки ви можете групувати бази даних, ви можете отримати максимальну користь від їхніх ресурсів на 2 серверах, запустивши деякі групи доступності з першого сервера, а деякі з другого.

## РОЗДІЛ 2: Використані технології

### 2.1. Hyper-V

Hyper-V найбільш відомий як технологія віртуалізації серверів; однак, починаючи з Windows 8, він також доступний у клієнтській операційній системі.

Microsoft Hyper-V, кодова назва Viridian, це нативний (тип 1) гіпервізор, який, на відміну від VMware Workstation, VirtualBox та інших гіпервізорів типу 2, працює безпосередньо на обладнанні. Вперше він був випущений у Windows Server 2008 і дає змогу запускати віртуальні машини в системах x86-64.

Hyper-V дозволяє розробникам швидко розганяти віртуальні машини для розробки на Windows 10 з чудовою продуктивністю.

В моїй курсовій використовую Hyper-V для запуску кількох віртуальних машин на Windows Server 2016 для симуляції роботи кластеру серверів, а також клієнтів, які підключені до баз даних розміщених на серверах.

### 2.2. Windows Server 2016

Windows Server 2016 - серверна операційна система, яка дозволяє розгорнути Microsoft SQL Server, а також дає доступ до спеціального Server Manager, який надає доступ до спеціальних інструментів по роботі з серверами, наприклад, Windows Server Failover Clustering (WSFC).

SQL Server побудовано на мові структурованих запитів (SQL), стандартизованій мові програмування, яка забезпечує основу для керування реляційними базами даних і запитів до них. SQL Server використовує модифіковану реалізацію SQL, яку називають Transact-SQL, яка додає набір власних розширень програмування до стандартної мови.

Microsoft SQL Server 2019 Enterprise Edition — це система керування реляційними базами даних, розроблена компанією Microsoft. Як сервер баз даних, це програмний продукт, основною функцією якого є зберігання та отримання даних за запитом інших програмних додатків, які можуть працювати або на тому ж комп'ютері, або на іншому комп'ютері в мережі (включаючи Інтернет).

SQL Server Enterprise Edition включає як основний механізм баз даних, так і додаткові служби, а також ряд інструментів для створення кластера SQL Server і керування ним. Він може керувати базами даних розміром до 524 петабайт і адресувати 12 терабайт пам'яті та підтримує 640 логічних процесорів (ядер ЦП).

### **2.3. MS SQL Server Management Studio (SSMS)**

MS SQL Server Management Studio(SSMS) — це інтегроване середовище для управління будь-якою інфраструктурою SQL. SSMS використовується для доступу, налаштування, керування, адміністрування та розробки всіх компонентів SQL Server, бази даних Azure SQL, керованого екземпляра Azure SQL, SQL Server на Azure VM та Azure Synapse Analytics. SSMS надає єдину комплексну утиліту, яка поєднує широку групу графічних інструментів з багатьма багатими редакторами сценаріїв, щоб надати доступ до SQL Server для розробників і адміністраторів баз даних усіх рівнів кваліфікації.

### **2.4. Windows Server Failover Clustering (WSFC)**

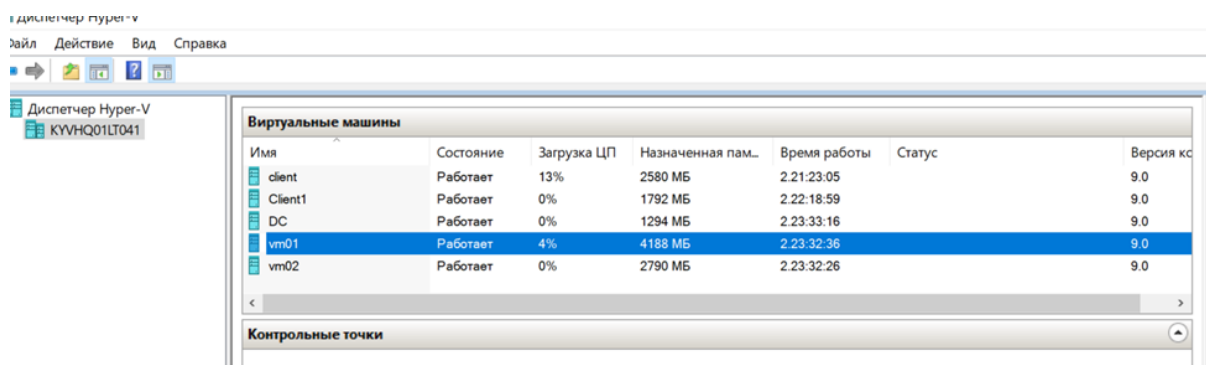
Windows Server Failover Clustering (WSFC) - є групою незалежних серверів, спільна робота яких дозволяє підвищити доступність додатків та служб. SQL Server підтримка екземплярів відмовостійкого кластера Групи доступності AlwaysOn та SQL Server здійснюється за допомогою служб та можливостей WSFC.

Відмовостійка кластеризація Windows Server передбачає інфраструктурні компоненти, що підтримують сценарії високого рівня доступності та аварійного відновлення для таких розміщених серверних програм, як Microsoft SQL Server та Microsoft Exchange. При відмові вузла кластера або служби всі служби, які розміщувалися на цьому вузлі, можуть автоматично або вручну переноситись на інший доступний вузол у рамках процесу під назвою відпрацювання відмови.

## РОЗДІЛ 3: Проектування і розробка

### 3.1. Налаштування серверів у Hyper-V

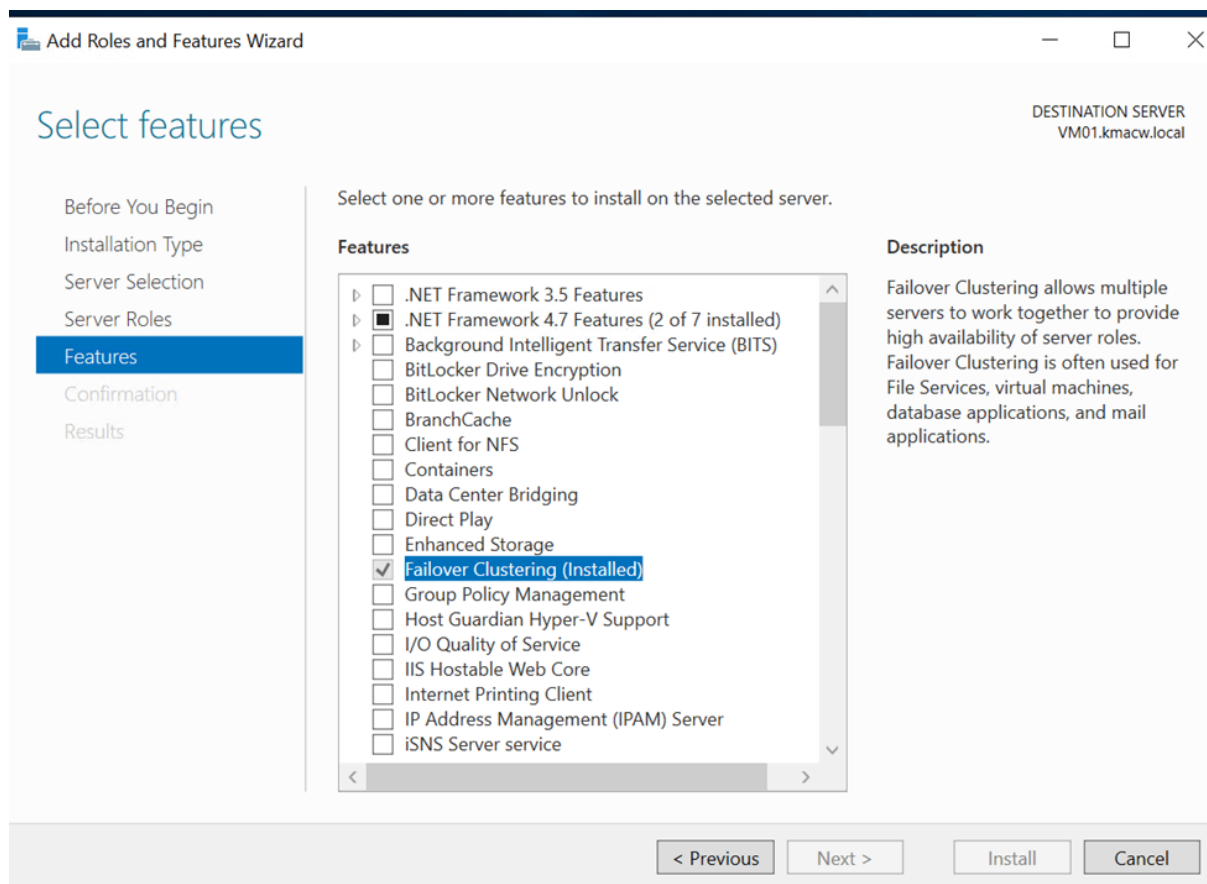
Симуляцію роботи було проведено в Hyper-V, який дозволяє симулювати віртуальні машини. Усього було створено 5 віртуальних машин. 2 машини (vm01 і vm02) симулювали 2 сервера, на які було встановлено Windows Server 2016. Було створено саме 2 сервери, оскільки це мінімальна кількість серверів, які потрібно для створення відмовостійкового кластеру AlwaysOn. Машина DC - Domain Controller, який відповідає за адміністрування серверів, а також для підтримки функції кворуму, яка використовується у відмовостійкому кластері. А також client, Client1, які відповідають за симуляцію клієнта, який підключається до кластеру. І який зчитує поведінку кластеру при збої серверу, який належить кластеру.



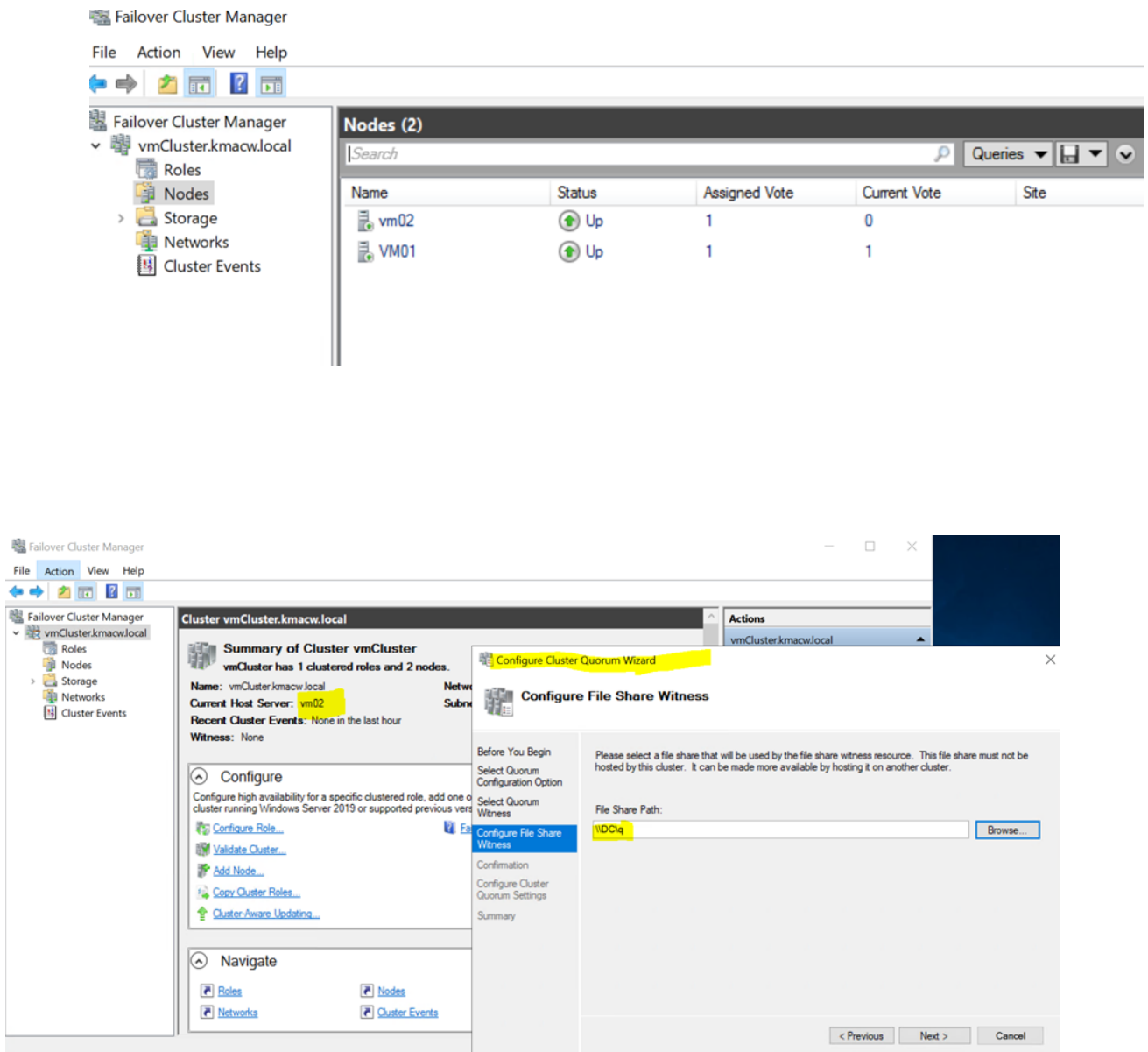
Имя	Состояние	Загрузка ЦП	Назначенная пам.	Время работы	Статус	Версия кс
client	Работает	13%	2580 МБ	2.21.23:05		9.0
Client1	Работает	0%	1792 МБ	2.22.18:59		9.0
DC	Работает	0%	1294 МБ	2.23.33:16		9.0
vm01	Работает	4%	4188 МБ	2.23.32:36		9.0
vm02	Работает	0%	2790 МБ	2.23.32:26		9.0

### 3.2. Налаштування Windows Server 2016

Для налаштування відмовостійкого кластеру AlwaysOn перш за все потрібно налаштувати Windows Server, а саме встановити Failover Clustering Feature.



Саме налаштування відмовостійкого кластера відбувається в програмі Failover Cluster Manager. Де налаштовуються вузли(Сервер, який є членом WSFC), ролі(Колекція ресурсів кластера, керована як єдиний об'єкт кластера і надає певні функціональні можливості. Для SQL Server роллю буде група доступності AlwaysOn або екземпляр відмовостійкого кластера AlwaysOn. Роль містить усі ресурси кластера, необхідні ролі групи доступності чи екземпляра отказоустойчивого кластера.) та режим кворуму.



### 3.3 Кворум

Кворум – це механізм, що дозволяє забезпечувати працездатність кластера WSFC шляхом перевірки достатньої кількості ресурсів у ньому. Якщо кластер WSFC має достатньо голосів, він працездатний і може забезпечувати стійкість до відмови на рівні вузлів.

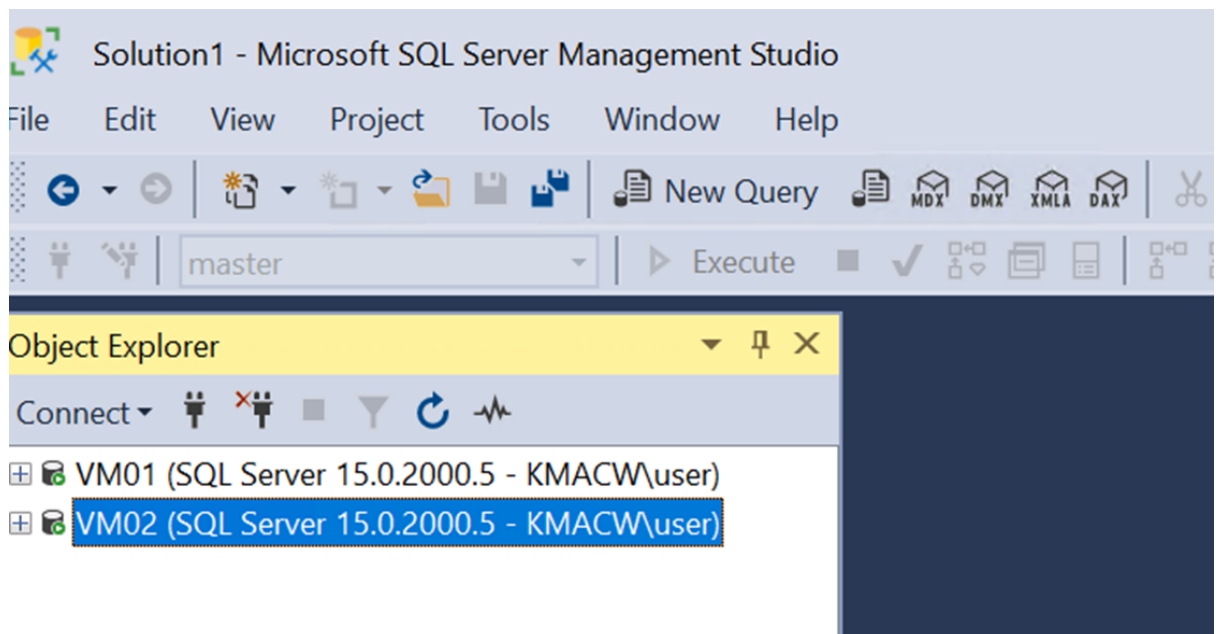
Режим кворуму налаштовується в кластері WSFC, який визначає методику голосування кворуму, а також момент виконання автоматичного переходу на інший ресурс або переведення кластера в режим "поза мережею".

Witness в даному випадку це файлова шара, яка знаходиться на іншому комп'ютері (DC)

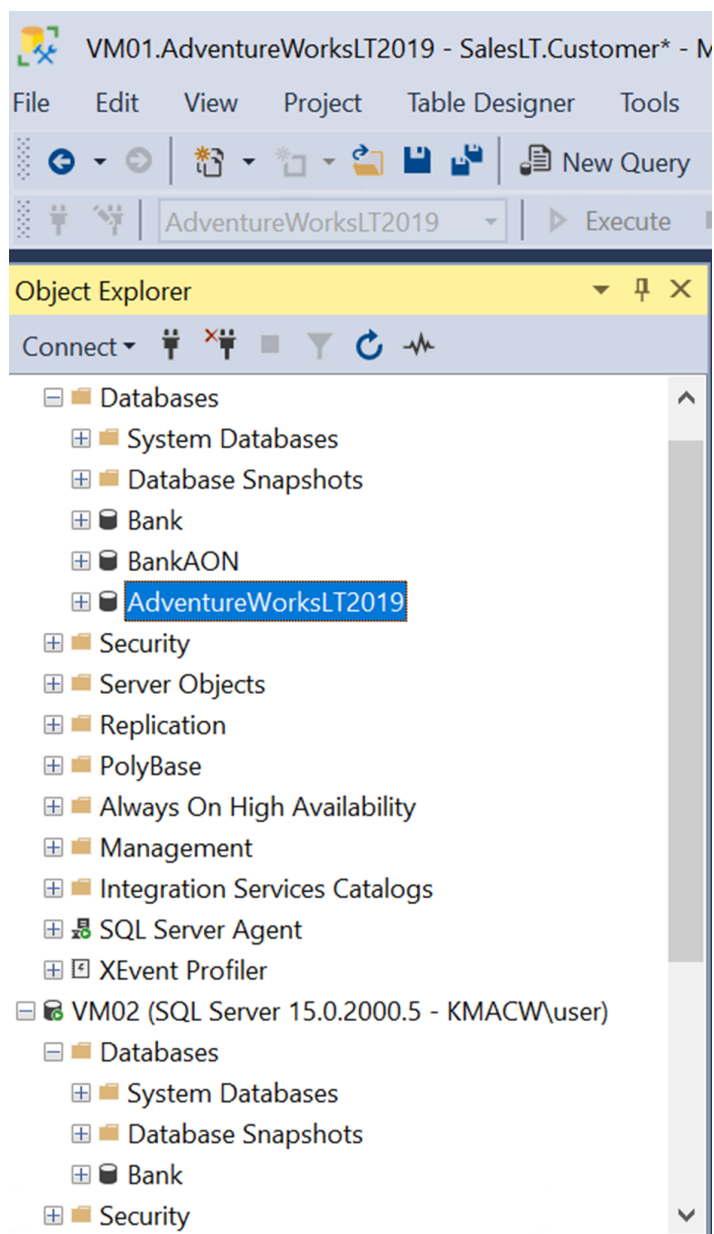
Якщо стається збій на першій ноді - VM01, друга нода не бачить VM02 і бачить файлову шару, приймає рішення про spaцювання fileover, після цього VM02 стає активним сервером, база стає доступна через listener.

### 3.4. Налаштування AlwaysOn

Налаштування відмовостійкого кластеру AlwaysOn відбувається в MS SQL Server Management Studio, в якій ми підключаємося до обох серверів vm01 і vm02.



Перебуваючи на vm01 для демонстрації відмовостійкості системи була створена база даних AdventureWorksLT2019.



Налаштування групи доступності AlwaysOn до якої входять 2 сервери vm01 і vm02, база даних Adventure WorksLT2019.

Група доступності AlwaysOn — це одна або кілька баз даних, для яких відпрацювання відмови виконується одночасно. Група доступності складається з первинної репліки доступності та від однієї до чотирьох вторинних реплік, які підтримуються за рахунок переміщення даних на підставі журналу SQL Server для забезпечення захисту даних, що не потребує загального сховища. Кожна репліка розміщується в екземплярі SQL Server в

окремому вузлі кластеру WSFC. Група доступності та відповідне ім'я віртуальної мережі реєструються як ресурси у кластері WSFC.

Прослуховувач(Listener) групи доступності на вузлі первинної репліки відповідає на вхідні клієнтські запити на підключення до імені віртуальної мережі та залежно від атрибутів у рядку підключення перенаправляє кожен запит у відповідний екземпляр SQL Server.

При відпрацюванні відмови замість перенесення володіння спільними фізичними ресурсами на інший вузол WSFC використовується для переналаштування вторинної репліки на іншому екземплярі SQL Server на первинну репліку групи доступності. Потім ресурс віртуального мережного імені групи доступності перекладається цей примірник.

Первинна репліка баз даних групи доступності одночасно може розміщуватися лише на одному екземплярі SQL Server , всі пов'язані вторинні репліки повинні знаходитись на окремому екземплярі, і кожен екземпляр повинен знаходитись на окремому фізичному вузлі.

New Availability Group



### Specify Availability Group Options

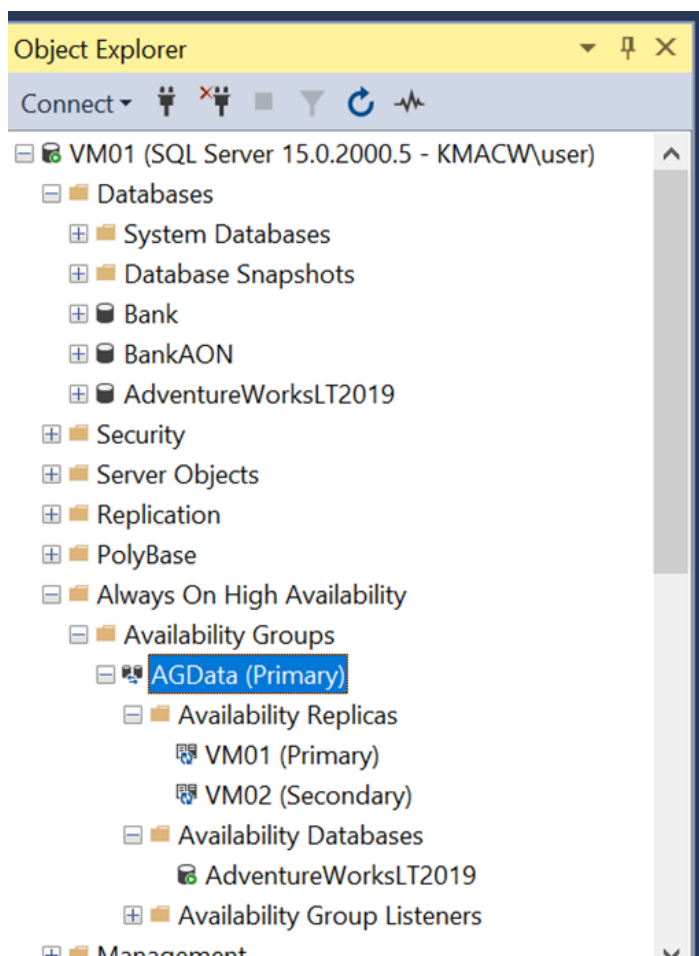
Introduction	<p><b>Specify availability group options</b></p> <p>Availability group name: <input type="text" value="AGData"/></p> <p>Cluster type: <input style="border: none; border-bottom: 1px solid #ccc;" type="text" value="Windows Server Failover Cluster"/> ▾</p> <p><input type="checkbox"/> Database Level Health Detection</p> <p><input type="checkbox"/> Per Database DTC Support</p>
<b>Specify Options</b>	
Select Databases	
Specify Replicas	
Select Data Synchronization	
Validation	
Summary	
Results	

Високий рівень доступності для вирішення AlwaysOn досягається за рахунок запобіжного моніторингу працездатності фізичних та логічних ресурсів кластеру WSFC, а також за рахунок автоматичного переходу на інший ресурс з переходом на дублююче обладнання та його переналаштування. Системний адміністратор також може запустити перехід на інший ресурс вручну групи доступності або екземпляра SQL Server для переходу з одного вузла на інший.

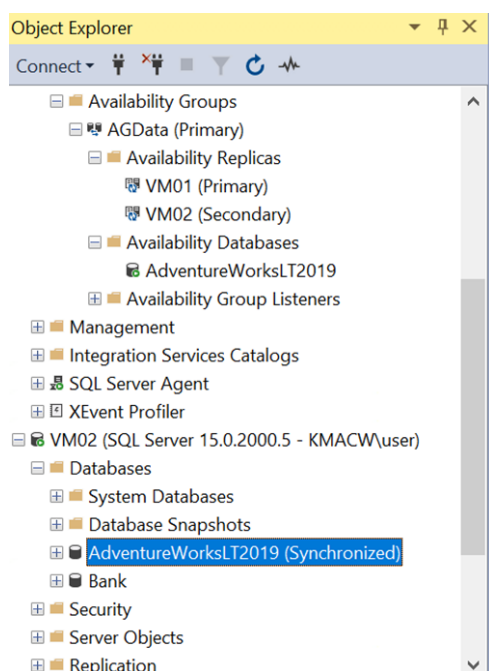
Політика відпрацювання відмови налаштовується на рівні вузла кластера WSFC, екземпляра відмовостійкого кластера SQL Server та групи доступності. Ця політика на основі серйозності, тривалості та частоти несправного стану ресурсу кластера та часу відгуку вузла може включати перезапуск служби або автоматичний перехід на інший ресурс з переходом з одного вузла на інший або включати переклад первинної репліки групи доступності з одного екземпляра SQL Server на інший.

В даному випадку в якості політики обробки відмови визначено автоматичний.

Автоматичний перехід на інший ресурс виникає у відповідь на збій, в результаті якого вторинна репліка, що синхронізується, приймає первинну роль (з гарантованим захистом даних). Коли колишня первинна репліка стає доступною, вона бере вторинну роль. Для автоматичного переходу на інший ресурс потрібно, щоб первинна репліка та цільова вторинна репліка працювали в режимі синхронної фіксації, а режим відпрацювання відмови від значення «Автоматичний». Крім цього, вторинна репліка вже має бути синхронізована, мати WSFC-кворум і відповідати умовам, зазначеним у гнучкій політиці переходу на інший ресурс для групи доступності.



Режим доступності – це властивість кожної репліки доступності. Режим доступності визначає, чи чекає первинна репліка перед фіксацією транзакцій для бази даних, щоб ця вторинна репліка записала записи журналу транзакцій на диск (записала журнал на диск). Групи доступності AlwaysOn підтримує два режими доступності: режим асинхронної фіксації та режим синхронної фіксації.



### Asynchronous-commit mode

Репліка доступності, яка використовує цей режим доступності, називається реплікою асинхронної фіксації. У режимі асинхронної фіксації первинна репліка фіксує транзакції, не чекаючи на підтвердження запису транзакції в журнал від вторинних реплік асинхронної фіксації. Режим асинхронної фіксації мінімізує затримку транзакцій у базах даних-одержувачах, але дозволяє їм не встигати за базами даних-джерелами, що створює ризик можливої втрати даних.

### Synchronous-commit mode

Репліка доступності, яка використовує цей режим доступності, називається реплікою синхронної фіксації. У режимі синхронної фіксації, перш ніж фіксувати транзакції, первинна репліка синхронної фіксації чекає, щоб вторинна репліка синхронної фіксації підтвердила, що запис журналу на диск завершено. У режимі синхронної фіксації після синхронізації бази даних-одержувача з базою даних-джерелом зафіксовані транзакції повністю захищені. Цей захист досягається за рахунок підвищення затримки транзакцій. У SQL Server 2017 введено додаткову функцію необхідної

кількості синхронних вторинних реплік для подальшого підвищення безпеки ціною затримки.

В даному випадку режим - синхронний, оскільки він забезпечує найбільше безпеку транзакцій.

294	False	Mr.	Alberto	F.	Baltazar	NULL	Flawless ...	adventure-works\li...	alberto0...	780-555-...	7Pu+s7F...	L1e3Y:
295	False	Mr.	Joseph	P.	Mitzner	Jr.	Engineer...	adventure-works\s...	joseph4...	129-555-...	70barvC...	k6zgRt
298	False	Mr.	Robert	NULL	Mitosinka	NULL	City Cycli...	adventure-works\jo...	robert10...	239-555-...	vCVsOjG...	mQH+
299	False	Mr.	Thierry	NULL	D'Hers	NULL	Citywide ...	adventure-works\jo...	thierry1...	680-555-...	i+ukExA...	FooIQI
300	False	Mr.	Daniel	NULL	Blanco	NULL	Nice Bikes	adventure-works\g...	daniel0@...	221-555-...	bnDkCX...	oOlbG
304	False	Ms.	Linda	NULL	Mitchell	NULL	Essential ...	adventure-works\ja...	linda9@a...	1 (11) 50...	LhS2DW...	hrRF3s
306	False	Mr.	Stephen	M.	Ayers	NULL	Work an...	adventure-works\m...	stephen1...	818-555-...	m79YsPy...	F0hRc
307	False	Ms.	Neva	M.	Mitchell	NULL	Riding A...	adventure-works\p...	neva0@a...	992-555-...	xUl8Ag9...	ru6E/q
308	False	Ms.	Megan	N.	Davis	NULL	Rugged ...	adventure-works\d...	megan1...	839-555-...	G6BhMd...	BNNuU
309	False	Mr.	John	NULL	Arthur	NULL	The Gear ...	adventure-works\jill...	john6@a...	149-555-...	AJwPH2L...	ZXH0E
310	False	Mr.	Frank	NULL	Miller	NULL	Orange B...	adventure-works\jill...	frank6@...	118-555-...	h7ho5pw...	n/q5in
311	False	Sr.	Alvaro	NULL	De Mato...	NULL	Principal ...	adventure-works\s...	alvaro0@...	551-555-...	BrhebOE...	v5pJp
312	False	Mr.	Thomas	B.	Armstrong	Jr.	Resale Se...	adventure-works\li...	thomas1...	226-555-...	8ml/xroz...	hXYQE
NULL	False	Mr.	MAX	V.	Zadoroz...	NULL	KursWork	Adventure-works\U...	test@test...	380-97-5...	abcd	hSwml

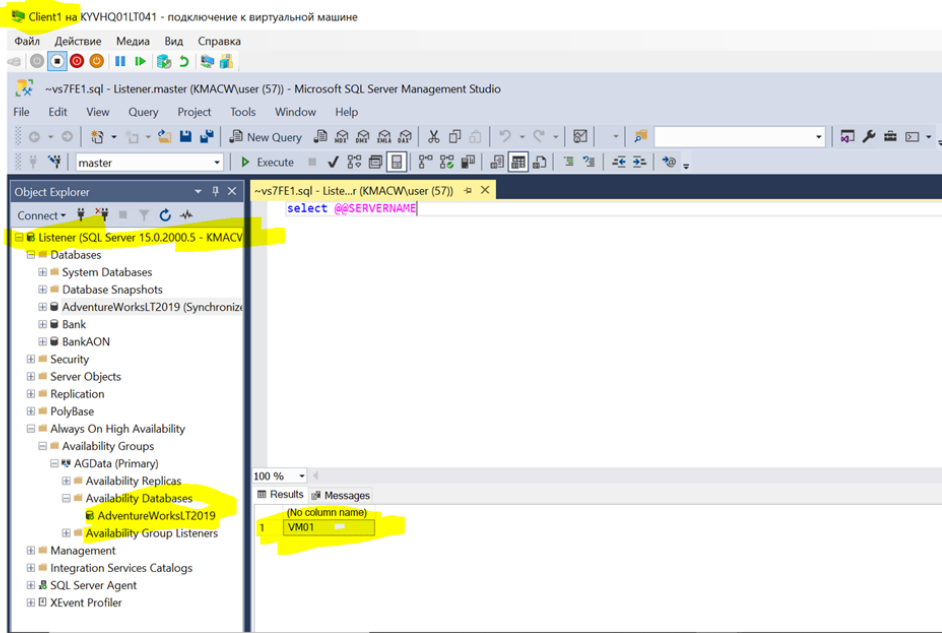
CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	CompanyName	SalesPerson	EmailAddress	Pht	
1	30133	0	Mr.	MAX	V.	Zadorozhny	NULL	KursWork	Adventure-works\Ukraine	test@test.test	38
2	544	0	Mr.	Jay	NULL	Adams	NULL	Valley Bicycle Specialists	adventure-works\jillian0	jay1@adventure-works.com	15
3	29492	0	Mr.	Jay	NULL	Adams	NULL	Valley Bicycle Specialists	adventure-works\jillian0	jay1@adventure-works.com	15
4	29490	0	Ms.	Margaret	J.	Smith	NULL	Bicycle Accessories and Kits	adventure-works\dawd8	margaret0@adventure-works.com	95
5	488	0	Ms.	Margaret	J.	Smith	NULL	Bicycle Accessories and Kits	adventure-works\dawd8	margaret0@adventure-works.com	95
6	491	0	Ms.	Frances	B.	Adams	NULL	Area Bike Accessories	adventure-works\shu0	frances0@adventure-works.com	99
7	29489	0	Ms.	Frances	B.	Adams	NULL	Area Bike Accessories	adventure-works\shu0	frances0@adventure-works.com	99
8	29485	0	Ms.	Catherine	R.	Abel	NULL	Professional Sales and Service	adventure-works\linda3	catherine0@adventure-works.com	74

Кожна репліка доступності намагається автоматично відновитися з урахуванням пошкоджених сторінок у локальній базі даних, дозволяючи певні типи помилок, які запобігають читанню сторінки даних. Якщо вторинна репліка не може прочитати сторінку, репліка вимагатиме свіжу копію сторінки з первинної репліки. Якщо первинна репліка не може прочитати сторінку, репліка надсилає запит отримання свіжої копії для всіх вторинних реплік і отримує сторінку від першої репліки, що відповіла. Якщо цей запит

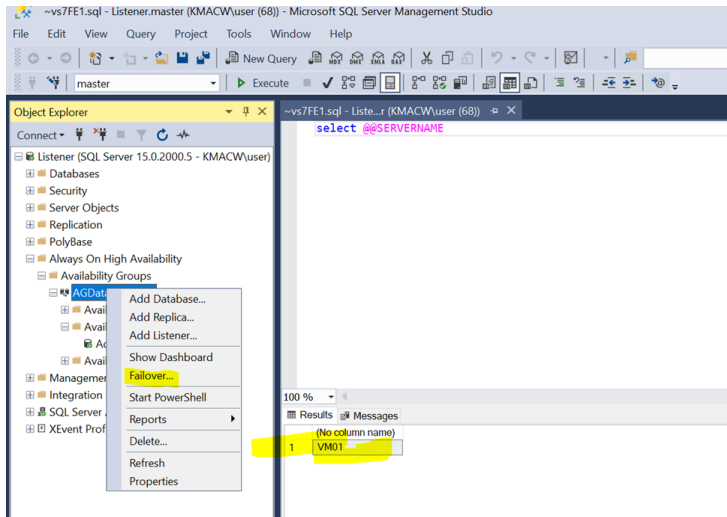
завершився успішно, то нечитана сторінка замінюється копією, внаслідок чого помилка зазвичай усувається.

### 3.5. Демонстрація роботи

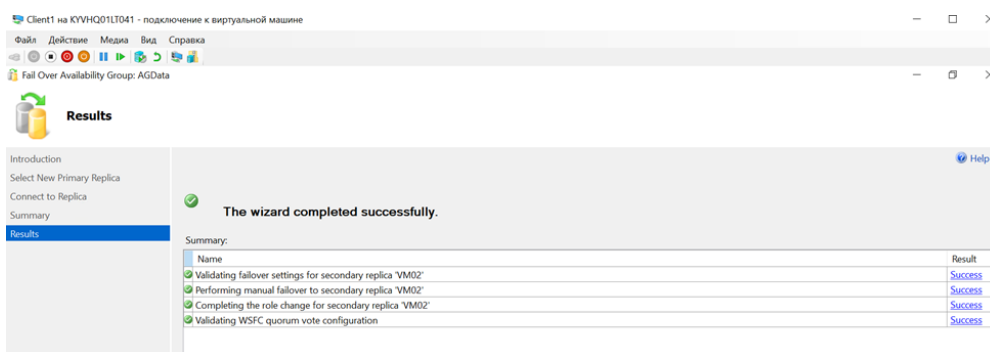
Для демонстрації роботи групи доступності always on я використовую віртуальну машину Client1, яка підключилась до Прослуховувача, в якому первинним сервером є vm01.

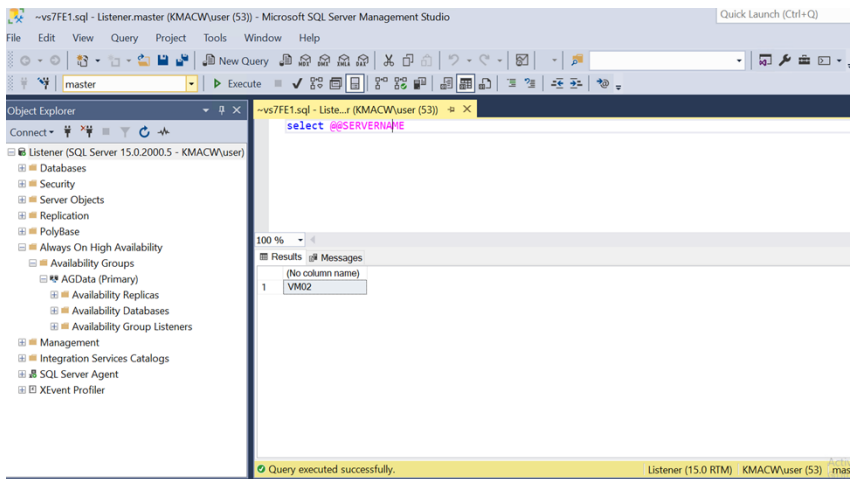


Далі я, як адміністратор бази даних, ініціюю заплановану відмову, і це змушує синхронізовану вторинну репліку взяти на себе роль первинної репліки. Для ручного переходу на відмову ваші бази даних AG повинні бути синхронізовані, щоб підтвердити, що ваша вторинна копія бази даних схожа на поточну первинну базу даних.



Припустимо, у вашому екземплярі SQL є кілька груп доступності. Через певну особливу вимогу ви хочете перейти до збою певної групи доступності баз даних. Якщо ви зупините Служби SQL на первинній репліці, вона вийде з ладу для всіх синхронізованих груп доступності. У цьому випадку спрацює запланований ручний перехід. Ви можете перемкнути певну групу доступності, однак зверніть увагу, що вона не працює над усіма базами даних у цій групі доступності. Ми не можемо перемкнути певну базу даних із групи доступності.





У режимі синхронізації даних можна використовувати як автоматичне, так і ручне перемикавання збоїв.

Ми не можемо мати автоматичну або заплановану ручну перехід на помилку в асинхронному режимі фіксації. У разі примусового переходу на відмову ми можемо перейти до вторинної репліки, особливо у сценарії аварійного відновлення.

В асинхронному коміті вторинна репліка може відставати від первинної бази даних. Таким чином, у разі відмови, це спричиняє втрату даних залежно від затримки синхронізації. Ви повинні виконувати примусове перемикавання збоїв AG, лише якщо вам потрібно відновити доступність бази даних і ви можете дозволити собі втратити деякі дані.

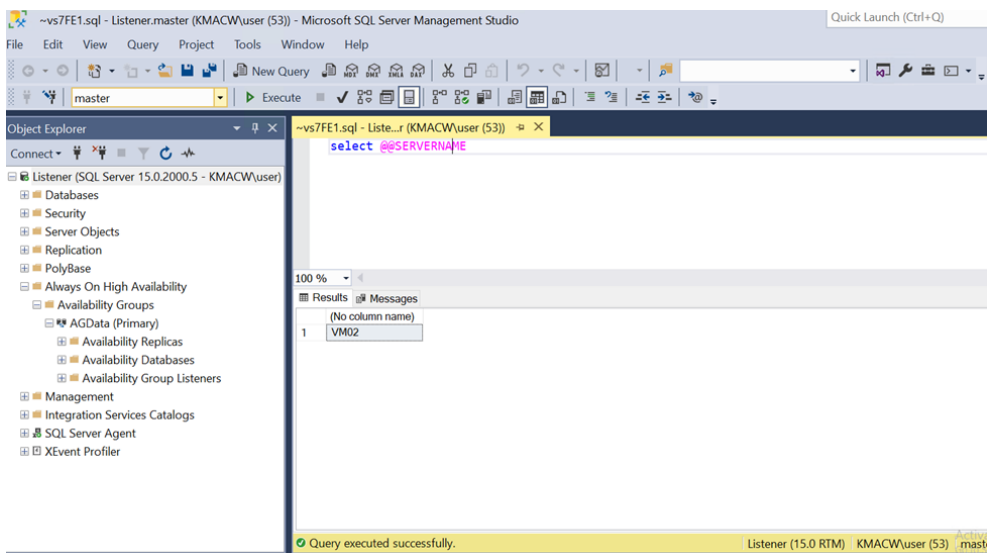
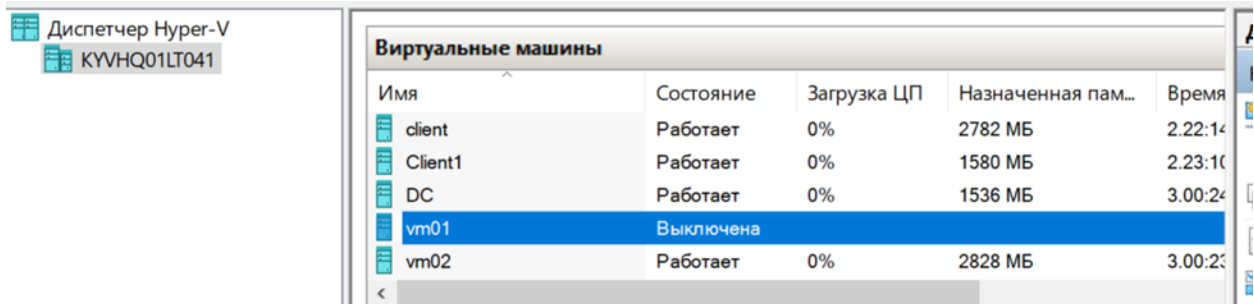
Для примусового перемикавання збоїв ваша база даних доступності має перебувати в режимі асинхронної фіксації. Він підтримує лише ручне перемикавання збоїв. Якщо ваша база даних AG перебуває в синхронному режимі, але ці бази даних не синхронізовані, вона може мати лише примусове перемикавання.

Автоматичне відмовлення відбувається автоматично, якщо первинна репліка виходить з ладу.

Для автоматичного перемикавання на помилку як первинну, так і додаткову репліку (мінімум одну додаткову) необхідно налаштувати за допомогою синхронного перемикавання та автоматичного перемикавання на помилку.

Обидві репліки повинні бути синхронізовані.

Я ініціюю автоматичне перемикавання вимикаючи vm01 первинний сервер.



Відбулося автоматичне перемикавання на vm02 сервер.

### 3.6. Висновки до розділу 3

Отже, мною були підготовлена платформа для розгортання СКБД MS SQL Server у складі гіпервізора Hyper-V та серверів під керуванням ОС Windows Server, на сервер встановлений інструмент роботи з базами даних MS SQL Management Studio, на сервери були встановлені СКБД MS SQL Server, СКБД MS SQL Server були зібрані в кластер за технологією AlwaysOn та була змодельована відмова основної ноди кластеру, в результаті чого було виконано автоматичне перемикавання кластеру на справну резервну ноду.

## Висновки

Отже, по завершенню роботи мною були підготовлена платформа для розгортання СКБД MS SQL Server у складі гіпервізора Hyper-V та серверів під керуванням ОС Windows Server, на сервер встановлений інструмент роботи з базами даних MS SQL Management Studio, на сервери були встановлені СКБД MS SQL Server, СКБД MS SQL Server були зібрані в кластер за технологією AlwaysOn та була змодельована відмова основної ноди кластеру, в результаті чого було виконано автоматичне перемикання кластеру на справну резервну ноду. Таким чином була досягнута основна мета - на платформі СКБД MS SQL Server побудований відмовостійкий кластер AlwaysOn, який забезпечив високу доступність бази даних для безперебійної роботи додатків.

В процесі вивчення документації та методичних матеріалів я також ознайомився з іншими стратегіями аварійного відновлення СКБД MS SQL Server, з іншими технологіями резервування баз даних (Log Shipping, Mirroring), зрозумів їх особливості, переваги та недоліки.

Можливі покращення:

- доповнити “гаряче резервування” СКБД MS SQL Server за допомогою технології AlwaysOn “холодним резервуванням” - налаштувавши за розкладом щотижневий full-бекап та щоденні differential-бекапи;
- доповнити зберігання даних на жорстких дисках систем зберігання даних - налаштувавши копіювання бекапів на роботизовану стрічкову бібліотеку.

В результаті таких покращень дані будуть додатково зарезервовані в “холодних бекапах” та будуть отримані додаткові “точки відкату” на тиждень та на дні тижня, а також дані будуть убезпечені від комп’ютерних вірусів-шифрувальників на іншому, ніж жорсткі диски, типі носія (магнітній

стрічці) - тобто буде забезпечений найвищий рівень надійності та відмовостійкості баз даних.

### Список використаної літератури:

1. How to manage disaster recovery in SQL Server [Електронний ресурс] // Robert Sheldon – Режим доступу до ресурсу:  
<https://www.techtarget.com/searchdisasterrecovery/feature/How-to-manage-disaster-recovery-in-SQL-Server>.
2. Comparing SQL Server HA Solutions: Always On Availability Groups vs. Log Shipping [Електронний ресурс] // Rajendra Gupta. – 2022. – Режим доступу до ресурсу:  
<https://severalnines.com/database-blog/comparing-high-availability-solutions-for-sql-server-always-on-availability-groups-vs-log-shipping>.
3. Difference Between Always On Failover Cluster, Database Mirroring, Always On Availability Group, Replication and Log Shipping [Електронний ресурс]. – 2019. – Режим доступу до ресурсу:  
<https://ramprasadtech.com/difference-between-always-on-failover-cluster-database-mirroring-always-on-availability-group-replication-and-log-shipping/>.
4. What is an Always On availability group? [Електронний ресурс] – Режим доступу до ресурсу:  
<https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/overview-of-always-on-availability-groups-sql-server?view=sql-server-ver15>.
5. Business continuity and database recovery - SQL Server [Електронний ресурс]. – 2022. – Режим доступу до ресурсу:  
<https://docs.microsoft.com/en-us/sql/database-engine/sql-server-business-continuity-dr?view=sql-server-ver15>.
6. Explore failover types in SQL Server Always On Availability Groups [Електронний ресурс] // Rajendra Gupta. – 2020. – Режим доступу до ресурсу:  
<https://www.sqlshack.com/explore-failover-types-in-sql-server-always-on-availability-groups/>.