Практична робота №7

РОБОТА З AZURE STORAGE: STORAGE ACCOUNT, BLOB STORAGE, AZURE FILES ТА AZURE FILE SYNC

Мета: ознайомитися зі службою Azure Storage та її ключовими компонентами – сховищем об’єктів Azure Blob Storage, файловим сховищем Azure Files та службою синхронізації Azure File Sync. Навчитися створювати й конфігурувати Storage Account, обирати тип надлишковості, керувати рівнями доступу та життєвим циклом даних, захищати дані за допомогою soft delete, версіонування, знімків, політик незмінності та object replication, керувати шифруванням (encryption scopes, customer-managed keys). Набути навичок авторизації доступу за допомогою Azure RBAC, ключів доступу та Shared Access Signatures, обмеження мережевого доступу за допомогою firewall, service endpoints і private endpoints, а також роботи з даними за допомогою Storage browser, Azure Storage Explorer та AzCopy.

Теоретичні відомості

Storage Account та служби Azure Storage

Storage Account – ресурс Azure, що надає унікальний простір імен для даних Azure Storage, доступний через HTTP або HTTPS. Обліковий запис типу Standard general-purpose v2 (рекомендований для більшості сценаріїв) містить чотири служби: Blob Storage (неструктуровані об’єкти), Azure Files (керовані файлові ресурси SMB та NFS), Queue Storage (черги повідомлень) і Table Storage (NoSQL-сховище пар ключ–значення). Для сценаріїв із низькою затримкою використовуються облікові записи категорії Premium: Premium block blobs, Premium file shares та Premium page blobs.

Ім’я Storage Account має бути глобально унікальним, містити від 3 до 24 символів і складатися лише з малих латинських літер і цифр, оскільки воно входить до імен кінцевих точок служб:

`https://<ім’я>.blob.core.windows.net`,
`https://<ім’я>.file.core.windows.net`,
`https://<ім’я>.queue.core.windows.net`,
`https://<ім’я>.table.core.windows.net`.

Параметр **Preferred storage type**, що обирається під час створення, лише налаштовує підказки в майстрі створення та не обмежує використання інших служб облікового запису.

Надлишковість даних

Azure Storage завжди зберігає кілька копій даних. Спосіб їх розміщення визначається параметром надлишковості (redundancy), порівняння варіантів наведено в табл. 1. Для геонадлишкових варіантів вторинний регіон визначається парою регіонів (region pair) і не обирається користувачем; у нових регіонах без пари (nonpaired regions) GRS, GZRS та їх RA-варіанти недоступні. Реплікація у вторинний регіон асинхронна, тому час останньої синхронізації (**Last sync time**) показує, до якого моменту дані гарантовано скопійовано. Варіанти з префіксом RA (read-access) дозволяють читати дані з вторинного регіону через окрему кінцеву точку з суфіксом `-secondary`, наприклад `https://<ім’я>-secondary.blob.core.windows.net`, ще до виконання failover.

Таблиця 1

Варіанти надлишковості Azure Storage

Варіант	Розміщення копій	Захист від відмови	Читання з вторинного регіону
LRS	3 копії в одному центрі обробки даних основного регіону	Стійки, сервера, диска	Ні
ZRS	3 копії в трьох зонах доступності основного регіону	Центру обробки даних	Ні
GRS	LRS в основному регіоні + LRS у парному регіоні	Регіону	Лише після failover
RA-GRS	Як GRS	Регіону	Так
GZRS	ZRS в основному регіоні + LRS у парному регіоні	Зони та регіону	Лише після failover
RA-GZRS	Як GZRS	Зони та регіону	Так

Azure Blob Storage, рівні доступу та життєвий цикл даних

Blob Storage зберігає об’єкти (blobs) у контейнерах. Ієрархія «папок» у плоскому просторі імен віртуальна: вона формується символом / в імені blob. Ввімкнення hierarchical namespace перетворює обліковий запис на Azure Data Lake Storage з реальними каталогами. Тип blob (block, append або page) задається під час створення і надалі не змінюється; перетворити append blob чи page blob на block blob можна лише копіюванням із зазначенням типу blob призначення. Порівняння типів blobs наведено в табл. 2.

Block blob складається з блоків, кожен з яких має ідентифікатор. Клієнт завантажує блоки (зокрема паралельно), а потім фіксує їх список; так великі файли передаються частинами, а невдало передані блоки надсилаються повторно. Block blobs підходять для документів, зображень, відео, резервних копій та є єдиним типом, що підтримує рівні доступу. **Append blob** також складається з блоків, але нові блоки можна лише дописувати в кінець, а змінити чи видалити наявні неможливо, що робить його зручним для журналів подій і даних аудиту: кілька клієнтів можуть

дописувати записи без координації між собою. **Page blob** – набір сторінок по 512 байтів, оптимізований для довільного читання та запису за зміщенням; розмір page blob задається під час створення і має бути кратним 512 байтам. У форматі VHD як page blobs зберігаються керовані диски віртуальних машин (managed disks) – в облікових записах, якими керує Azure. Під'єднання page blobs з облікового запису користувача безпосередньо до віртуальних машин (unmanaged disks) виведено з експлуатації 31 березня 2026 року, проте page blobs і надалі використовуються для імпорту та експорту VHD, баз даних з довільним доступом (наприклад, Azure SQL Database), Azure Site Recovery та Azure Backup.

Таблиця 2

Типи blobs в Azure Blob Storage

Тип	Структура та операції	Макс. розмір	Рівні доступу	Застосування
Block blob	Блоки до 4000 MiB (до 50 000 блоків); завантаження блоків і фіксація їх списку; перезапис цілком або заміна окремих блоків	≈ 190,7 ТіБ	Hot, Cool, Cold, Archive	Документи, медіа, резервні копії, дані аналітики
Append blob	Блоки лише дописуються в кінець (Append Block); змінити чи видалити наявні блоки неможливо; до 50 000 блоків	≈ 195 ГіБ (блоки по 4 MiB)	Немає	Журнали, аудит, телеметрія
Page blob	Сторінки по 512 байтів; читання та запис довільних діапазонів за зміщенням; розмір кратний 512 байтам	8 ТіБ	Немає (крім рівнів продуктивності Premium)	VHD, бази даних з довільним доступом

Для block blobs доступні рівні доступу (access tiers), що відрізняються вартістю зберігання та доступу (табл. 3). Рівень за замовчуванням задається на рівні облікового запису (Hot або Cool), а для окремого blob може бути змінений вручну чи правилами **Lifecycle management**. Правило життєвого циклу містить фільтр (тип blob, префікс) та дії, що виконуються за умовою, наприклад «перемістити в Cool, якщо blob не змінювався 30 днів» або

«видалити через 365 днів». Видалення чи переміщення blob до завершення мінімального терміну зберігання рівня тарифікується як дострокове видалення.

Таблиця 3

Рівні доступу Azure Blob Storage

Рівень	Мінімальний термін зберігання	Доступ до даних	Типове застосування
Hot	–	Онлайн, найнижча вартість доступу	Дані, що використовуються часто
Cool	30 днів	Онлайн	Рідко використовувані дані, короткострокові резервні копії
Cold	90 днів	Онлайн	Дані, до яких звертаються дуже рідко, але потрібен швидкий доступ
Archive	180 днів	Офлайн; потребує відновлення (rehydration) до онлайн-рівня	Довгострокове архівне зберігання

Blob у рівні Archive неможливо прочитати: спочатку необхідно змінити його рівень на онлайн (rehydrate). Відновлення зі стандартним пріоритетом (**Standard**) триває до 15 годин, з високим (**High**) – як правило, до однієї години для об’єктів розміром до 10 ГБ, але коштує дорожче. Рівень Archive доступний лише для облікових записів LRS, GRS та RA-GRS.

Захист даних Blob Storage та Azure Files

Soft delete зберігає видалені blobs, контейнери або файлові ресурси протягом заданого періоду (від 1 до 365 днів), протягом якого їх можна відновити. **Versioning** автоматично зберігає попередню версію blob під час кожного перезапису або видалення; будь-яку попередню версію можна зробити поточною. **Change feed** – упорядкований журнал змін blobs, необхідний, зокрема, для object replication. Знімок файлового ресурсу (**share snapshot**) – інкрементна копія Azure file share лише для читання на певний момент часу, з якої можна відновити окремі файли.

Immutable storage зберігає дані в стані WORM (Write Once, Read Many). Політика time-based retention забороняє змінювати та видаляти blobs протягом заданого інтервалу, legal hold – до явного зняття. Політику time-based retention спочатку створено в стані unlocked (її можна змінити або видалити, що корисно для тестування); після блокування (locked) її можна лише подовжити, і ні адміністратор підписки, ні служба підтримки Microsoft не можуть видалити захищені дані до завершення інтервалу. Політики можна задавати на рівні контейнера (container-level WORM) або версії (version-level WORM).

Object replication асинхронно копіює block blobs із контейнера облікового запису-джерела в контейнер облікового запису-призначення (в тому самому або іншому регіоні). Для неї необхідно ввімкнути versioning в обох облікових записах і change feed у джерелі. Політика реплікації містить правила (пари контейнерів) з необов'язковими фільтрами за префіксом; за замовчуванням копіюються лише нові об'єкти.

Авторизація доступу до даних

Запит до даних Azure Storage авторизується одним із способів. **Microsoft Entra ID та Azure RBAC** – рекомендований спосіб: права надаються ролями рівня даних, наприклад **Storage Blob Data Reader**, **Storage Blob Data Contributor**, **Storage File Data Privileged Contributor**; ролі рівня керування (Owner, Contributor) самі по собі доступу до даних через Microsoft Entra ID не надають. **Shared Key** – два ключі доступу облікового запису (key1, key2), кожен з яких надає повний доступ до всіх даних; їх періодично ротують, використовуючи другий ключ як проміжний. Авторизацію Shared Key можна повністю заборонити параметром **Allow storage account key access**. **Анонімний доступ** – читання контейнера або blobs без авторизації; дозволяється спочатку на рівні облікового запису, а потім для окремого контейнера.

Shared Access Signature (SAS) – маркер у рядку запиту URL, що надає обмежений доступ до ресурсу: визначені дозволи, час початку та завершення дії, дозволені IP-адреси та протокол. Типи SAS наведено в табл. 4. Ad hoc SAS неможливо відкликати інакше, ніж ротацією ключа, яким його підписано. **Stored access policy** – політика доступу, що зберігається на контейнері (до п’яти на ресурс); service SAS, прив’язаний до неї, успадковує її дозволи та терміни, а зміна чи видалення політики миттєво впливає на всі такі SAS.

Таблиця 4

Типи Shared Access Signature

Тип SAS	Чим підписується	Особливості
User delegation SAS	Ключем делегування користувача Microsoft Entra ID	Рекомендований; лише Blob Storage; права обмежені правами RBAC користувача; не залежить від ключів облікового запису
Service SAS	Ключем облікового запису	Доступ до однієї служби; підтримує stored access policy
Account SAS	Ключем облікового запису	Доступ до однієї чи кількох служб, зокрема до операцій рівня служби

Шифрування даних

Усі дані Azure Storage автоматично шифруються на стороні сервера (Storage Service Encryption) алгоритмом AES-256; вимкнути шифрування неможливо. За замовчуванням використовуються ключі, керовані Microsoft (**Microsoft-managed keys**). **Customer-managed keys (CMK)** зберігаються в Azure Key Vault або Managed HSM; сховище звертається до ключа від імені managed identity, якій надано роль **Key Vault Crypto Service Encryption User**. Для Key Vault обов’язково вмикається purge protection, щоб ключ не можна було остаточно видалити. Якщо в URI ключа не вказано версію, Azure Storage раз на добу автоматично переходить на найновішу версію ключа, тому ротація виконується створенням нової версії в Key Vault. **Encryption scopes** дозволяють шифрувати окремі контейнери чи blobs окремим ключем (Microsoft-managed або customer-managed), наприклад для

розмежування даних різних клієнтів; encryption scope неможливо видалити, лише вимкнути, а тарифікація кожного scope становить щонайменше один місяць. **Infrastructure encryption** додає другий рівень шифрування на рівні інфраструктури.

Мережевий доступ до Storage Account

Параметр **Public network access** визначає доступ до публічних кінцевих точок облікового запису: дозволено з усіх мереж, лише з вибраних віртуальних мереж та IP-адрес, або вимкнено. Правила firewall можуть дозволяти доступ з публічних IP-адрес і діапазонів, з підмереж віртуальних мереж, з окремих екземплярів ресурсів Azure, а також винятки для довірених служб Azure. Правила IP-адрес не діють для запитів, що надходять із того самого регіону Azure, що й обліковий запис; для таких клієнтів використовуються правила віртуальних мереж.

Service endpoint (`Microsoft.Storage` для облікових записів у регіоні мережі або `Microsoft.Storage.Global` для будь-якого регіону) змінює маршрут трафіку з підмережі до Azure Storage на магістральну мережу Microsoft і передає в запиті ідентичність підмережі, завдяки чому правило firewall може дозволити саме цю підмережу. Кінцева точка облікового запису при цьому залишається публічною. **Private endpoint** – мережевий інтерфейс із приватною IP-адресою з підмережі віртуальної мережі, пов’язаний із конкретною службою облікового запису (blob, file тощо). Ім’я `<ім’я>.blob.core.windows.net` через CNAME `<ім’я>.privatelink.blob.core.windows.net` і приватну DNS-зону розв’язується всередині мережі в приватну IP-адресу, тому приватна кінцева точка працює навіть за вимкненого публічного доступу.

Azure Files та Azure File Sync

Azure Files надає повністю керовані файлові ресурси, доступні за протоколами SMB (Windows, Linux, macOS; порт TCP 445) та NFS (Linux).

Файлові ресурси в Storage Account (`Microsoft.Storage`) у порталі називаються **Classic file shares** і підтримують SMB та NFS. Ресурсний провайдер `Microsoft.FileShares` дозволяє створювати файлові ресурси як окремі ресурси Azure без Storage Account, проте наразі підтримує лише NFS. Для стандартних облікових записів доступні рівні Transaction optimized, Hot та Cool; ресурс підключається (mount) за допомогою скрипту, згенерованого в порталі (**Connect**).

Azure File Sync перетворює Windows Server на кеш файлового ресурсу Azure. Складові: **Storage Sync Service** (ресурс Azure, з яким встановлюється довіра серверів), **registered server** (Windows Server із встановленим агентом Azure File Sync), **sync group** (топологія синхронізації), **cloud endpoint** (Azure file share; лише один у групі) та **server endpoint** (шлях на зареєстрованому сервері). Функція **cloud tiering** залишає на сервері лише часто використовувані файли, а решту – в Azure Files. Зміни, внесені на сервері, синхронізуються майже одразу, а зміни, внесені безпосередньо у файловий ресурс Azure, виявляються завданням change detection, що запускається раз на 24 години.

Інструменти роботи з даними

Storage browser – вбудований у портал інструмент перегляду та керування даними облікового запису. **Azure Storage Explorer** – безкоштовний настільний застосунок для Windows, macOS і Linux з тими самими можливостями та підтримкою кількох облікових записів і підписок. **AzCopy** – утиліта командного рядка для високопродуктивного копіювання та синхронізації даних між локальною файловою системою, облікові записи Azure Storage та іншими хмарами; авторизується через Microsoft Entra ID або SAS і попередньо встановлена в Azure Cloud Shell.

Хід роботи

У завданнях використовуються такі позначення: **xxx** – скорочена назва обраного регіону серед доступних у політиці Allowed resource deployment regions відповідно до цього файлу; **yyy** – ініціали студента латинськими літерами; **zzz** – номер варіанту з трьох цифр (наприклад, для 15 варіанту – 015). Скорочення типів ресурсів в іменах відповідають рекомендаціям Cloud Adoption Framework. Імена Storage Account не можуть містити дефіси, тому мають вигляд `stxxxyyyzzz`; якщо ім'я зайняте, додається суфікс `rrr` з трьох випадкових символів a-z0-9. Усі ресурси та Resource groups створюються в регіоні, обраному в кроці 1.2. Команди виконуються в Azure Cloud Shell (Bash), якщо не зазначено інше. Звіт має містити скріншоти та виведення команд, що підтверджують виконання кожного кроку.

Частина 1. Storage Account

1. Обрати регіон і розмір віртуальних машин та підготувати Subscription.
   1.1. Відкрити Azure Portal та Azure Cloud Shell (Bash). Виконати скрипт, який для кожного регіону з політики Allowed resource deployment regions виводить розміри віртуальних машин з не більше ніж 2 vCPU, доступні для підписки та з ненульовою

квотою сімейства (обсяг пам’яті та, у дужках, сімейство, використана кількість vCPU і ліміт):

```
REGIONS=$(az policy assignment show \
  --name sys.regionrestriction \
  --query "parameters.listOfAllowedLocations.value" -o tsv)
Q_USAGE="[?to_number(limit)>\`0\`] \
  .[name.value, currentValue, limit]"
Q_SKU="[?length(restrictions[?type=='Location'])==\`0\` \
  && capabilities[?name=='vCPUs' \
  && to_number(value)<=\`2\`]] \
  .[name, family, \
  capabilities[?name=='MemoryGB'].value|[0]]"

for L in $REGIONS; do
  echo "=== $L"
  az vm list-usage --location $L --query "$Q_USAGE" \
    -o tsv > ~/quota-$L.txt
  az vm list-skus --location $L \
    --resource-type virtualMachines \
    --query "$Q_SKU" -o tsv |
  while read SKU FAMILY MEM; do
    Q=$(grep -i "^$FAMILY[[:space:]]" ~/quota-$L.txt) &&
      echo "  $SKU  RAM=${MEM}GB  ($Q)"
  done
done
```

1.2. Обрати регіон, який має парний регіон (геонадлишкові варіанти GRS та RA-GRS недоступні в регіонах без пари, наприклад Spain Central, Poland Central, Italy North, Austria East), у якому доступні розміри з квотою сімейства щонайменше 4 vCPU, та розмір з 2 vCPU і не менше 4 ГіБ пам’яті (наприклад, `Standard_B2ls_v2` або `Standard_B2als_v2`): у частинах 3 і 4 одночасно працюватимуть дві віртуальні машини Linux та Windows Server. Не обирати розміри серій DC, EC та L. Обрані значення зафіксувати у звіті.

1.3. Задати змінні та зареєструвати resource providers, необхідні для практичної роботи (команди, як і змінні, знадобляться в

наступних завданнях; після перезапуску Cloud Shell змінні необхідно задати знову):

```
LOC=<обраний регіон, наприклад francecentral>
SIZE=<обраний розмір, наприклад Standard_B2ls_v2>
SUFFIX=xxx-yyy-zzz
RG=rg-st-$SUFFIX
SA=stxxxyyyzzz
for NS in Microsoft.Storage Microsoft.KeyVault \
          Microsoft.StorageSync Microsoft.Network \
          Microsoft.Compute; do
  az provider register --namespace $NS
done
```

2. Створити Storage Account.

2.1. У головному пошуку Azure Portal знайти **Storage accounts** та натиснути **Create**. На вкладці **Basics** задати: **Resource group** – **Create new**, `rg-st-xxx-yyy-zzz`; **Storage account name** – `stxxxyyyzzz`; **Region** – регіон з кроку 1.2; **Preferred storage type** – **Azure Blob Storage or Azure Data Lake Storage**; **Performance** – **Standard**; **Redundancy** – **Geo-redundant storage (GRS)**; встановити прапорець **Make read access to data available in the event of regional unavailability**.

2.2. На вкладках **Advanced** та **Security** переглянути доступні параметри та залишити значення за замовчуванням. Зафіксувати у звіті значення параметрів **Allow enabling anonymous access on individual containers**, **Enable storage account key access** та **Minimum TLS version**.

2.3. На вкладці **Networking** для **Public network access** обрати **Disable**.

2.4. На вкладці **Data protection** переконатися, що ввімкнено **Enable soft delete for blobs**, **Enable soft delete for containers** та **Enable soft delete for file shares** з періодом 7 днів, та додатково встановити прапорці **Enable versioning for blobs** і **Enable blob change feed**. На вкладці **Encryption** залишити значення за

замовчуванням. Натиснути **Review + create** та **Create**, після розгортання – **Go to resource**.

3. Дослідити конфігурацію Storage Account та надати права доступу до даних.
   3.1. На сторінці **Overview** переглянути служби облікового запису (Blob service, File service, Queue service, Table service) та значення **Replication** – **Read-access geo-redundant storage (RA-GRS)**.
   3.2. Перейти в **Security + networking** > **Networking** та переконатися, що публічний доступ вимкнено. Перейти в **Storage browser** та зафіксувати помилку доступу до даних.
   3.3. Перейти в **Data management** > **Redundancy** та переглянути основне (**Primary**) й вторинне (**Secondary**) розташування даних. Зафіксувати у звіті назву парного регіону.
   3.4. Перейти в **Security + networking** > **Networking**, у розділі **Public network access** натиснути **Manage**, обрати **Enable** та **Public network access scope** – **Enable from all networks**, зберегти зміни. Публічний доступ з усіх мереж тимчасово потрібен для частин 1–2 (зокрема для Azure Cloud Shell, правила IP-адрес для якого можуть не діяти) і буде обмежений у частині 3.
   3.5. Перейти в **Access Control (IAM)**, натиснути **Add** > **Add role assignment** та призначити власному обліковому запису роль **Storage Blob Data Contributor** (вкладка **Members** – **User, group, or service principal** > **Select members**). Аналогічно призначити роль **Storage File Data Privileged Contributor**. Пояснити у звіті, чому ролі Owner чи Contributor на рівні підписки недостатньо для доступу до даних через Microsoft Entra ID. Призначення ролей може поширюватися до 5 хвилин.

Частина 2. Azure Blob Storage

4. Створити контейнер, завантажити blob та надати доступ за допомогою user delegation SAS.
   4.1. Перейти в **Data storage** > **Containers**, натиснути **Add container** та створити контейнер `data` з рівнем анонімного доступу **Private (no anonymous access)**.
   4.2. Відкрити контейнер `data`, натиснути **Upload**, обрати невеликий файл з комп'ютера (зображення або текстовий документ) та розгорнути **Advanced**: **Blob type** – **Block blob**; **Block size** – **4 MiB**; **Access tier** – **Hot**; **Upload to folder** – `securitytest`; **Encryption scope** – значення за замовчуванням. Натиснути **Upload** та переконатися, що з'явилися папка `securitytest` і файл.
   4.3. Відкрити завантажений файл, на вкладці **Overview** скопіювати **URL** та відкрити його в приватному вікні браузера (InPrivate або Incognito). Зафіксувати помилку `ResourceNotFound` або `PublicAccessNotPermitted` та пояснити її причину.
   4.4. Для файлу натиснути **…** > **Generate SAS** та задати: **Signing method** – **User delegation key**; **Permissions** – **Read**; **Start** – вчорашня дата та поточний час; **Expiry** – завтрашня дата та поточний час; **Allowed IP addresses** – порожнє; **Allowed protocols** – **HTTPS only**. Натиснути **Generate SAS token and URL**, скопіювати **Blob SAS URL**, зберегти його та відкрити в приватному вікні браузера. Переконатися, що файл відображається. Пояснити у звіті параметри `sv`, `st`, `se`, `sr`, `sp`, `skoid` та `sig` у рядку запиту.

5. Дослідити типи blobs.

5.1. В Azure Cloud Shell задати змінні з кроку 1.3 (якщо Cloud Shell перезапускався), створити контейнер `blobtypes` та завантажити до нього текстовий документ як block blob:

```
A="--account-name $SA --auth-mode login"
az storage container create $A --name blobtypes
C="--container-name blobtypes"
echo "Звіт з практичної роботи, версія 1" > report.txt
az storage blob upload $A $C --file report.txt \
  --name report.txt --type block
```

5.2. Змінити вміст документа та завантажити його повторно з параметром `--overwrite`. Переконатися, що вміст blob замінено повністю, а на вкладці **Versions** blob `report.txt` у порталі з'явилася попередня версія:

```
echo "Звіт з практичної роботи, версія 2" > report.txt
az storage blob upload $A $C --file report.txt \
  --name report.txt --type block --overwrite
az storage blob download $A $C --name report.txt \
  --file report-dl.txt -o none && cat report-dl.txt
```

5.3. Змоделювати журнал застосунку в append blob: дописати до blob `app.log` п'ять записів (кожне завантаження з типом `append` без `--overwrite` додає блок у кінець наявного append blob), завантажити журнал і переглянути кількість зафіксованих блоків:

```
for i in 1 2 3 4 5; do
  echo "$(date -u +%FT%TZ) INFO request $i" > line.txt
  az storage blob upload $A $C --file line.txt \
    --name app.log --type append -o none
  sleep 2
done
az storage blob download $A $C --name app.log \
  --file app.log -o none && cat app.log
az storage blob show $A $C --name app.log \
  --query properties.appendBlobCommittedBlockCount
```

5.4. Спробувати дописати запис до block blob `report.txt` як до append blob. Зафіксувати помилку та пояснити, чому дописування в кінець можливе лише для append blobs:

```
az storage blob upload $A $C --file line.txt \
  --name report.txt --type append
```

5.5. Створити файл розміром 1 МіБ з випадковими даними та завантажити його як page blob, а потім спробувати завантажити як page blob файл розміром 1000 байтів. Зафіксувати помилку та пояснити її:

```
head -c 1048576 /dev/urandom > disk.vhd
az storage blob upload $A $C --file disk.vhd \
  --name disk.vhd --type page
head -c 1000 /dev/urandom > bad.vhd
az storage blob upload $A $C --file bad.vhd \
  --name bad.vhd --type page
```

5.6. Вивести тип, рівень доступу та розмір усіх blobs контейнера, а потім спробувати змінити рівень доступу append blob на Cool. Зафіксувати помилку та порівняти результат з block blob `report.txt`:

```
Q_TYPES="[].{name:name, \
  type:properties.blobType, \
  tier:properties.blobTier, \
  size:properties.contentLength}"
az storage blob list $A $C --query "$Q_TYPES" -o table
az storage blob set-tier $A $C --name app.log \
  --tier Cool
```

5.7. Пояснити у звіті, для яких сценаріїв призначено кожен тип blob, чому журнали зручно зберігати в append blobs і чому керовані диски віртуальних машин зберігаються як page blobs.

6. Налаштувати керування життєвим циклом та дослідити рівні доступу.

6.1. Перейти в **Data management** > **Lifecycle management** та натиснути **Add a rule**. Задати **Rule name** – `movetocool`, **Rule scope** – **Apply rule to all blobs in your storage account**, натиснути **Next**. На вкладці **Base blobs** задати умову: **Last modified** – **More than (days ago)** `30`, **Then** – **Move to cool storage**. Натиснути **Add**.

6.2. Переглянути правило в поданні **Code view** та додати JSON-визначення політики до звіту.

6.3. Завантажити в контейнер `data` (без папки) ще один файл `archive-test.txt` довільного вмісту. Для нього натиснути **…** >

Change tier, обрати **Cold** та **Save**; потім так само змінити рівень на **Archive**.

6.4. Спробувати завантажити (**Download**) файл `archive-test.txt` та зафіксувати помилку `BlobArchived`. Пояснити у звіті, чому blob недоступний для читання.

6.5. Для файлу натиснути **…** > **Change tier**, обрати **Access tier** – **Hot**, **Rehydrate priority** – **Standard** та **Save**. Переглянути стан відновлення в Azure Cloud Shell (очікуване значення – `rehydrate-pending-to-hot`); дочікуватися завершення відновлення не потрібно:

```
az storage blob show --account-name $SA \
  --container-name data --name archive-test.txt \
  --auth-mode login --query "[properties.blobTier, \
  properties.rehydrationStatus, rehydratePriority]"
```

7. Дослідити versioning та soft delete.

7.1. Створити на комп'ютері файл `version.txt` з текстом `v1` та завантажити його в контейнер `data`. Змінити вміст файлу на `v2` та завантажити повторно, встановивши прапорець **Overwrite if files already exist**.

7.2. Відкрити `version.txt`, перейти на вкладку **Versions** та переглянути попередню версію. Обрати попередню версію та натиснути **Make current version**. Переконатися (**Edit** або **Download**), що поточний вміст – `v1`.

7.3. Видалити `version.txt`. У списку blobs увімкнути **Show deleted blobs**, відкрити видалений blob, перейти на вкладку **Versions**, обрати потрібну версію та натиснути **Make current version**. Переконатися, що blob відновлено.

7.4. Створити контейнер `temp` та видалити його. На сторінці **Containers** увімкнути **Show deleted containers**, для `temp` натиснути **…** > **Undelete** та переконатися, що контейнер

відновлено. Пояснити у звіті, від яких ситуацій захищають versioning, soft delete для blobs та soft delete для контейнерів.

8. Налаштувати політику незмінності (immutable storage).
   8.1. Створити контейнер `immutable`. Для контейнера натиснути **…** > **Access policy**, у розділі **Immutable blob storage** натиснути **Add policy**: **Policy type** – **Time-based retention**, **Set retention period for** – `1` день. Натиснути **Save**. Політику не блокувати (вона залишається в стані **Unlocked**).
   8.2. Завантажити в контейнер `immutable` файл `worm.txt`. Спробувати видалити його, а також завантажити файл з тим самим іменем із прапорцем **Overwrite if files already exist**. Зафіксувати помилки та пояснити різницю між станами Unlocked і Locked політики time-based retention.
9. Керувати доступом за допомогою stored access policy, ключів доступу та анонімного доступу.
   9.1. Для контейнера `data` натиснути **…** > **Access policy**, у розділі **Stored access policies** натиснути **Add policy**: **Identifier** – `read-list`; **Permissions** – **Read** та **List**; **Start time** – вчорашня дата; **Expiry time** – завтрашня дата. Натиснути **OK**, а потім обов'язково **Save**.
   9.2. Для контейнера `data` натиснути **…** > **Generate SAS**: **Signing method** – **Account key**; **Signing key** – **Key 1**; **Stored access policy** – `read-list` (дозволи й терміни успадковуються з політики). Натиснути **Generate SAS token and URL** та скопіювати **Blob SAS URL**. Додати до URL параметри переліку вмісту контейнера й відкрити його в приватному вікні браузера – має відобразитися XML-перелік blobs:

```
<Blob SAS URL>&restype=container&comp=list
```

9.3. Видалити stored access policy `read-list` (**…** > **Delete**, потім **Save**), зачекати до 30 секунд та оновити сторінку з SAS URL. Зафіксувати помилку автентифікації та пояснити, як stored access policy дає змогу відкликати SAS без ротації ключів.

9.4. Завантажити до папки `securitytest` контейнера `data` другий файл. Для першого файлу згенерувати SAS із **Signing method** – **Account key** та **Signing key** – **Key 1**, а для другого – з **Signing key** – **Key 2** (дозвіл **Read**). Перевірити доступність обох SAS URL у приватному вікні браузера.

9.5. Перейти в **Security + networking** > **Access keys** та для **key1** натиснути **Rotate key**. Перевірити SAS URL, підписаний Key 1, SAS URL, підписаний Key 2, та user delegation SAS URL з кроку 4.4. Пояснити отримані результати та порядок ротації ключів без простою застосунків.

9.6. Перейти в **Settings** > **Configuration**, для **Allow Blob anonymous access** обрати **Enabled** та натиснути **Save**. На сторінці **Containers** обрати контейнер `data`, натиснути **Change access level**, обрати **Blob (anonymous read access for blobs only)** та **OK**.

9.7. Відкрити в приватному вікні браузера URL (без SAS) обох файлів у папці `securitytest`, а також URL самої папки `https://<ім’я>.blob.core.windows.net/data/securitytest`. Пояснити, чому файли доступні, а папка – ні. Повернути контейнеру рівень **Private (no anonymous access)** та вимкнути **Allow Blob anonymous access**.

10. Перенести дані за допомогою AzCopy та Azure Storage Explorer.

10.1. В Azure Cloud Shell створити каталог з кількома файлами та скопіювати його в новий контейнер `azcopy`, авторизуючись

через Microsoft Entra ID за допомогою облікових даних Azure CLI:

```
mkdir -p ~/azcopy-demo && cd ~/azcopy-demo
for i in 1 2 3; do echo "file $i" > file$i.txt; done
export AZCOPY_AUTO_LOGIN_TYPE=AZCLI
export AZCOPY_TENANT_ID=$(az account show \
  --query tenantId -o tsv)
URL=https://$SA.blob.core.windows.net/azcopy
az storage container create --account-name $SA \
  --name azcopy --auth-mode login
azcopy copy "$HOME/azcopy-demo/*" "$URL"
azcopy list "$URL"
```

10.2. Змінити вміст одного з файлів, видалити інший та синхронізувати каталог з контейнером. Пояснити у звіті відмінність між `azcopy copy` та `azcopy sync`:

```
echo "changed" > file1.txt && rm file2.txt
azcopy sync "$HOME/azcopy-demo" "$URL" \
  --delete-destination=true
azcopy list "$URL"
```

10.3. Встановити на локальний комп'ютер застосунок Azure Storage Explorer, увійти до облікового запису Azure (**Sign in with Azure**), розгорнути підписку Azure for Students та обліковий запис `stxxxyyyzzz`. Переглянути контейнери `data` та `azcopy` і завантажити з них будь-який файл на комп'ютер. Порівняти можливості Storage Explorer та Storage browser у звіті.

11. Налаштувати object replication.

11.1. Створити другий Storage Account у Resource group `rg-st-xxx-yyy-zzz`: **Storage account name** – `stxxxyyyzzzrep`; **Region** – регіон з кроку 1.2; **Preferred storage type** – **Azure Files**; **Performance** – **Standard**; **Redundancy** – **Locally-redundant storage (LRS)**; на вкладці **Data protection** встановити прапорець **Enable versioning for blobs**; інші параметри за замовчуванням. Цей обліковий запис також використовуватиметься в частині 4.

11.2. Створити контейнер `orsource` в обліковому записі `stxxxyyyzzz` та контейнер `replica` в обліковому записі `stxxxyyyzzzrep`.

11.3. В обліковому записі-джерелі `stxxxyyyzzz` перейти в **Data management** > **Object replication** та натиснути **Create replication rules**. Обрати **Destination subscription** та **Destination storage account** – `stxxxyyyzzzrep`; у розділі **Container pair details** обрати **Source container** – `orsource` та **Destination container** – `replica`; для **Copy over** натиснути **change** та обрати **Everything**. Натиснути **Save and apply**.

11.4. Завантажити в контейнер `orsource` кілька файлів. Через 2–5 хвилин переконатися, що вони з'явилися в контейнері `replica`. Відкрити один із файлів у джерелі та переглянути на вкладці **Overview** стан **Object replication**. Пояснити, чому для object replication необхідні versioning і change feed.

12. Налаштувати шифрування за допомогою encryption scope та customer-managed key.

12.1. В обліковому записі `stxxxyyyzzz` перейти в **Security + networking** > **Encryption**, відкрити вкладку **Encryption scopes** та натиснути **Add**: **Encryption scope name** – `scope-ms`; **Encryption type** – **Microsoft-managed keys**; встановити прапорець **Infrastructure encryption**. Натиснути **Create**.

12.2. Створити контейнер `scoped`, розгорнувши **Advanced**: **Encryption scope** – `scope-ms`, встановити прапорець **Use this encryption scope for all blobs in the container**. Завантажити в контейнер файл, відкрити його та переконатися, що у властивостях вказано encryption scope `scope-ms`.

12.3. Створити Key Vault: у головному пошуку знайти **Key vaults**, натиснути **Create** та задати **Resource group** – `rg-st-xxx-yyy-zzz`, **Key vault name** – `kv-xxx-yyy-zzz` (за потреби з суфіксом

`rrr`), **Region** – регіон з кроку 1.2, **Pricing tier** – **Standard**, **Days to retain deleted vaults** – `7`, **Purge protection** – **Enable purge protection**. На вкладці **Access configuration** переконатися, що обрано **Azure role-based access control**. Натиснути **Review + create** та **Create**.

12.4. У Key Vault перейти в **Access control (IAM)** та призначити власному обліковому запису роль **Key Vault Crypto Officer**. Після поширення ролі перейти в **Objects** > **Keys**, натиснути **Generate/Import**: **Name** – `cmk-storage`, **Key type** – **RSA**, **RSA key size** – `2048`, натиснути **Create**.

12.5. В обліковому записі `stxxxyyyzzz` перейти в **Security + networking** > **Identity**, на вкладці **System assigned** встановити **Status** – **On** та натиснути **Save**. Скопіювати **Object (principal) ID**.

12.6. У Key Vault перейти в **Access control (IAM)** та призначити роль **Key Vault Crypto Service Encryption User** керованій ідентичності облікового запису (**Assign access to** – **Managed identity**, **Select members** – **Storage account** – `stxxxyyyzzz`).

12.7. В обліковому записі `stxxxyyyzzz` перейти в **Security + networking** > **Encryption** та обрати **Customer-managed keys**. Обрати **Select from key vault** > **Select a key vault and key**, вказати Key Vault `kv-xxx-yyy-zzz` та ключ `cmk-storage`, **Identity type** – **System-assigned**. Натиснути **Save**. Переконатися, що портал відображає поточну версію ключа та ввімкнене автоматичне оновлення версії.

12.8. У Key Vault відкрити ключ `cmk-storage` та натиснути **New Version** > **Create**. Пояснити у звіті, коли Azure Storage почне використовувати нову версію ключа і що станеться з даними, якщо ключ вимкнути або видалити.

Частина 3. Azure Files та мережевий доступ

13. Перевірити доступ до вторинного регіону RA-GRS.

    13.1. Переглянути основне й вторинне розташування, стан вторинного регіону, вторинну кінцеву точку Blob Storage та час останньої синхронізації:

```
Q_GEO="{primary:primaryLocation, \
  secondary:secondaryLocation, \
  status:statusOfSecondary, \
  endpoint:secondaryEndpoints.blob, \
  lastSync:geoReplicationStats.lastSyncTime}"
az storage account show --name $SA --resource-group $RG \
  --expand geoReplicationStats --query "$Q_GEO" -o yaml
```

    13.2. Розв’язати імена основної та вторинної кінцевих точок і порівняти отримані IP-адреси:

```
nslookup $SA.blob.core.windows.net
nslookup $SA-secondary.blob.core.windows.net
```

    13.3. Визначити регіони Azure, яким належать обидві IP-адреси, за допомогою сервісу Azure Speed IP Lookup. Порівняти результат з виведенням `az storage account show`.

    13.4. У SAS URL файлу, підписаному Key 2 (крок 9.4), замінити `<ім’я>.blob.core.windows.net` на `<ім’я>-secondary.blob.core.windows.net` та відкрити отриманий URL у приватному вікні браузера. Пояснити, чому файл доступний, а для запису вторинна кінцева точка недоступна. Якщо файл ще не з’явився у вторинному регіоні, зачекати, оскільки реплікація асинхронна.

14. Створити Azure file share, знімок файлового ресурсу та відновити дані.

    14.1. В обліковому записі `stxxxyyyzzz` перейти в **Data storage** > **Classic file shares** та натиснути **Classic file share**: **Name** – `share1`, **Access tier** – **Transaction optimized**; на вкладці **Backup** зняти прапорець **Enable backup**. Натиснути **Review + create** та **Create**.

14.2. Перейти в **Storage browser** > **Classic file shares** > `share1`, створити папку `docs` (**Add directory**) та завантажити до неї файл `notes.txt` з текстом `original`. Якщо виникає помилка авторизації, змінити метод автентифікації на **Microsoft Entra user account**.

14.3. Відкрити `share1`, перейти в **Operations** > **Snapshots**, натиснути **Add snapshot** та створити знімок з коментарем `before-change`.

14.4. Змінити вміст `notes.txt` на `changed` (повторне завантаження з перезаписом). У знімку `before-change` знайти файл `docs/notes.txt`, натиснути **Restore**, обрати **Overwrite original file** та **OK**. Переконатися, що вміст файлу – `original`.

14.5. Створити другий файловий ресурс `share2`, видалити його, на сторінці **Classic file shares** увімкнути **Show deleted shares** та відновити `share2` (**Undelete**). Пояснити відмінність між знімком файлового ресурсу та soft delete.

15. Обмежити мережевий доступ до Storage Account за допомогою service endpoint.

15.1. У головному пошуку знайти **Virtual networks** та створити віртуальну мережу в Resource group `rg-st-xxx-yyy-zzz`: **Name** – `vnet-st-xxx-yyy-zzz`, **Region** – регіон з кроку 1.2, інші параметри – за замовчуванням (адресний простір `10.0.0.0/16`, підмережа `default`).

15.2. Відкрити віртуальну мережу, перейти в **Settings** > **Service endpoints**, натиснути **Add**, обрати **Service** – **Microsoft.Storage**, **Subnets** – `default` та натиснути **Add**.

15.3. В обліковому записі `stxxxyyyzzz` перейти в **Security + networking** > **Networking**, у розділі **Public network access** натиснути **Manage** та обрати **Public network access scope** – **Enable from selected networks**. Натиснути **Add a virtual**

network > **Add existing virtual network**, обрати `vnet-st-xxx-yyy-zzz` і підмережу `default`, натиснути **Add**. Переконатися, що в розділі **IPv4 Addresses** немає IP-адреси комп'ютера, та натиснути **Save**.

15.4. Через кілька хвилин оновити **Storage browser** та спробувати відкрити контейнер `data` і файловий ресурс `share1` з комп'ютера. Зафіксувати помилку та пояснити її.

16. Перевірити доступ до Blob Storage та Azure Files з віртуальної машини.

16.1. У головному пошуку знайти **Virtual machines** та обрати **Create** > **Virtual machine**. На вкладці **Basics** задати: **Resource group** – `rg-st-xxx-yyy-zzz`; **Virtual machine name** – `vm-xxx-yyy-zzz`; **Region** – регіон з кроку 1.2; **Availability options** – **No infrastructure redundancy required**; **Security type** – **Trusted launch virtual machines**; **Image** – **Ubuntu Server 24.04 LTS - x64 Gen2**; **Size** – розмір з кроку 1.2; **Authentication type** – **Password**; **Username** – ім'я користувача, аналогічне імені в Azure Cloud Shell; **Password** – надійний пароль (у звіті не вказувати); **Public inbound ports** – **Allow selected ports**, **Select inbound ports** – **SSH (22)**.

16.2. На вкладці **Networking** обрати **Virtual network** – `vnet-st-xxx-yyy-zzz` та **Subnet** – `default` (підмережа вже має service endpoint `Microsoft.Storage`). Натиснути **Review + create** та **Create**.

16.3. Скопіювати **Public IP address** віртуальної машини та підключитися до неї через SSH з Azure Cloud Shell або локального терміналу:

```
ssh <ім'я користувача>@<Public IP address>
```

16.4. У віртуальній машині завантажити файл за user delegation SAS URL з кроку 4.4 (за потреби згенерувати новий) та обчислити його хеш SHA-256:

```
curl -s -o downloaded.bin '<Blob SAS URL>'
sha256sum downloaded.bin
```

16.5. Обчислити хеш SHA-256 вихідного файлу на локальному комп'ютері (у Windows PowerShell – командою нижче, у macOS та Linux – `shasum -a 256` або `sha256sum`) і переконатися, що значення збігаються:

```
Get-FileHash <ім'я файлу> -Algorithm SHA256
```

16.6. В Azure Portal відкрити файловий ресурс `share1`, натиснути **Connect**, обрати **Linux** та **Show Script**. Скопіювати скрипт, виконати його в SSH-сеансі та переконатися, що ресурс підключено:

```
mount | grep share1
```

16.7. Перейти до каталогу, у який підключено ресурс (наприклад, `/mnt/share1`), та створити файл з відомостями про віртуальну машину:

```
cd /mnt/share1
IP=$(hostname -I | cut -d" " -f1)
OS=$(. /etc/os-release && echo "$PRETTY_NAME")
printf "Hostname: %s\nDistro: %s\nIP: %s\n" \
  "$(hostname)" "$OS" "$IP" | sudo tee os-info.txt
```

16.8. Тимчасово додати IP-адресу комп'ютера до правил firewall облікового запису (**Networking** > **Manage** > **IPv4 Addresses** > **Add your client IPv4 address**, **Save**), відкрити `share1` у **Storage browser**, завантажити `os-info.txt` та переглянути його вміст. Після перевірки видалити IP-адресу з правил і зберегти зміни.

17. Налаштувати private endpoint для Blob Storage.

17.1. В обліковому записі `stxxxyyyzzz` перейти в **Security + networking** > **Networking**, відкрити вкладку **Private endpoints** та натиснути **Private endpoint**. На вкладці **Basics** задати **Name** – `pep-blob-xxx-yyy-zzz`, **Region** – регіон з кроку 1.2; на

вкладці **Resource** – **Target sub-resource** – **blob**; на вкладці **Virtual Network** – **Virtual network** – `vnet-st-xxx-yyy-zzz`, **Subnet** – `default`, **Dynamically allocate IP address**; на вкладці **DNS** – **Integrate with private DNS zone** – **Yes**. Натиснути **Review + create** та **Create**.

17.2. Розв'язати ім'я кінцевої точки Blob Storage в SSH-сеансі віртуальної машини та в Azure Cloud Shell. Пояснити відмінність результатів і роль приватної DNS-зони `privatelink.blob.core.windows.net`:

```
nslookup <ім'я облікового запису>.blob.core.windows.net
```

17.3. У налаштуваннях **Networking** облікового запису натиснути **Manage** та обрати для **Public network access** – **Disable**, натиснути **Save**. Повторно виконати команду `curl` з кроку 16.4 у віртуальній машині та переконатися, що файл завантажується через private endpoint, а **Storage browser** з комп'ютера доступ до даних не надає.

17.4. Видалити віртуальну машину `vm-xxx-yyy-zzz` (разом з диском і мережевим інтерфейсом), щоб звільнити квоту vCPU для частини 4.

Частина 4. Azure File Sync

18. Розгорнути Storage Sync Service та файловий ресурс для синхронізації.

18.1. В обліковому записі `stxxxyyyzzzrep` створити файловий ресурс `syncshare` (**Classic file shares** > **Classic file share**, **Access tier** – **Transaction optimized**, резервне копіювання вимкнено). Переконатися, що в **Settings** > **Configuration** параметр **Allow storage account key access** має значення **Enabled**.

18.2. У головному пошуку знайти **Azure File Sync** та натиснути **Create**. Задати **Resource group** – `rg-st-xxx-yyy-zzz`, **Storage**

sync service name – `sss-xxx-yyy-zzz`, **Region** – регіон з кроку 1.2 (має збігатися з регіоном облікового запису `stxxxyyyzzzrep`). Натиснути **Review + create** та **Create**.

19. Підготувати Windows Server та зареєструвати його в Storage Sync Service.
    19.1. Створити віртуальну машину: **Resource group** – `rg-st-xxx-yyy-zzz`; **Virtual machine name** – `vm-fs-yyy-zzz` (ім'я віртуальної машини Windows не може перевищувати 15 символів); **Region** – регіон з кроку 1.2; **Image** – **Windows Server 2025 Datacenter - x64 Gen2**; **Size** – розмір з кроку 1.2; **Username** та **Password** – довільні надійні значення; **Public inbound ports** – **Allow selected ports**, **Select inbound ports** – **RDP (3389)**; на вкладці **Networking** – віртуальна мережа `vnet-st-xxx-yyy-zzz`, підмережа `default`. Після створення в **Networking** > **Network settings** змінити джерело правила RDP на **My IP address**.
    19.2. Підключитися до віртуальної машини через RDP (**Connect** > **Download RDP file**). У **Server Manager** обрати **Local Server**, натиснути посилання **IE Enhanced Security Configuration** та обрати **Off** для **Administrators** і **Users**.
    19.3. Завантажити агент Azure File Sync з Microsoft Download Center (пошуковий запит «Azure File Sync agent download») у версії для Windows Server 2025 та запустити MSI-пакет. У майстрі встановлення натиснути **Next**, прийняти ліцензійну угоду, залишити шлях за замовчуванням `C:\Program Files\Azure\StorageSyncAgent`, налаштування проксі та оновлень за замовчуванням, натиснути **Install** та **Finish**.
    19.4. Після встановлення автоматично відкривається вікно **Server Registration** (інакше запустити `C:\Program`

Files\Azure\StorageSyncAgent\ServerRegistration.exe). Обрати середовище **AzureCloud**, натиснути **Sign in**, увійти до облікового запису Azure, обрати **Azure Subscription**, **Resource Group** – `rg-st-xxx-yyy-zzz`, **Storage Sync Service** – `sss-xxx-yyy-zzz` та натиснути **Register**. Після повторного входу переконатися в успішній реєстрації.

19.5. В Azure Portal відкрити `sss-xxx-yyy-zzz`, перейти в **Sync** > **Registered servers** та переконатися, що сервер `vm-fs-yyy-zzz` має стан **Online**.

20. Створити sync group, cloud endpoint та server endpoint і перевірити синхронізацію.

20.1. У `sss-xxx-yyy-zzz` перейти в **Sync** > **Sync groups**, натиснути **Create a sync group**: **Sync group name** – `sg-xxx-yyy-zzz`; **Storage account** – **Select storage account** > `stxxxyyyzzzrep`; **Azure File Share** – `syncshare`. Натиснути **Create** та переконатися, що автоматично створено cloud endpoint.

20.2. На віртуальній машині створити каталог `C:\SyncData` та кілька текстових файлів у ньому. Відкрити групу `sg-xxx-yyy-zzz`, натиснути **Add server endpoint**: **Registered Server** – `vm-fs-yyy-zzz`; **Path** – `C:\SyncData`; **Cloud Tiering** – вимкнено (server endpoint на системному томі не підтримує cloud tiering); **Initial Sync** – злиття вмісту (**Merge the content of this server path with the content in the Azure file share**). Натиснути **Create**.

20.3. Через кілька хвилин відкрити `syncshare` у **Storage browser** облікового запису `stxxxyyyzzzrep` та переконатися, що файли з `C:\SyncData` синхронізовано. Створити на сервері ще один файл і перевірити його появу у файловому ресурсі.

20.4. Завантажити файл безпосередньо у `syncshare` через портал та перевірити, чи з'явився він у `C:\SyncData`. Пояснити результат з огляду на механізм change detection Azure File Sync.

21. Видалити всі ресурси, створені під час виконання практичної роботи.

21.1. У групі `sg-xxx-yyy-zzz` видалити server endpoint, потім cloud endpoint і саму sync group. У **Registered servers** для `vm-fs-yyy-zzz` обрати **Unregister server**. Без цих кроків видалення Storage Sync Service завершиться помилкою.

21.2. У контейнері `immutable` видалити політику time-based retention (**Access policy** > **…** > **Delete**), оскільки дані під дією політики блокують видалення облікового запису.

21.3. Видалити Resource group `rg-st-xxx-yyy-zzz` разом з усіма ресурсами та переконатися, що її видалено:

```
az group delete --name $RG --yes
az group list --output table
```

21.4. Переглянути Key Vault у стані soft delete (**Key vaults** > **Manage deleted vaults**) та пояснити, чому через увімкнений purge protection його неможливо остаточно видалити до завершення періоду зберігання. Видалити Resource group `NetworkWatcherRG`, якщо її було автоматично створено.

Контрольні запитання

1. Які служби містить Storage Account типу Standard general-purpose v2 та чим відрізняються Azure Blob Storage і Azure Files? Які вимоги до імені Storage Account і чому?
2. Порівняйте варіанти надлишковості LRS, ZRS, GRS, GZRS та їх RA-варіанти. Що таке Last sync time і як отримати доступ до даних у вторинному регіоні?
3. Чим відрізняються block blobs, append blobs та page blobs і для яких сценаріїв призначено кожен тип? Які рівні доступу існують для block blobs і що потрібно зробити, щоб прочитати blob у рівні Archive?
4. Як versioning, soft delete для blobs і контейнерів та знімки файлових ресурсів захищають дані? Від яких сценаріїв вони не захищають?
5. Що таке immutable storage? Чим відрізняються time-based retention і legal hold, а також стани Unlocked і Locked політики?
6. Які способи авторизації доступу до даних Azure Storage існують? Чим відрізняються ролі рівня керування та рівня даних в Azure RBAC?
7. Порівняйте user delegation SAS, service SAS та account SAS. Як можна відкликати кожен із них і яку роль відіграє stored access policy?
8. Як працюють customer-managed keys та encryption scopes? Навіщо Key Vault потрібен purge protection і як виконується ротація ключа?
9. Чим відрізняються правила firewall, service endpoint та private endpoint для Storage Account? Як змінюється розв’язання DNS-імені облікового запису після створення private endpoint?
10. З яких компонентів складається Azure File Sync і як відбувається синхронізація змін із сервера та з файлового ресурсу Azure? Що таке object replication і які умови необхідні для її налаштування?