МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Київський національний університет будівництва і архітектури

GRID-системи та хмарні технології

Методичні вказівки
до виконання лабораторних робіт № 1-7
для підготовки здобувачів другого магістерського рівня вищої освіти за спеціальністю 121 «Інженерія програмного забезпечення»

Київ 2024

УДК 004.042

G

Укладач: О.Л. Соловей, канд. техн. наук

Рецензент О.А. Поплавський, канд. техн. наук, доцент

Відповідальна за випуск Т.А. Гончаренко, канд. тех. наук, доцент

Затверджено на засіданні кафедри інформаційних технологій, протокол № 9 від 27 березня 2024 року.

В авторській редакції.

GRID-системи та хмарні технології: Методичні вказівки до виконання лабораторних робіт / Уклад. О.Л. Соловей. – Київ: КНУБА, 2024. – 41 с

Містять теоретичні відомості і рекомендації щодо виконання лабораторних робіт з дисципліни та вимоги до оформлення звіту. Спрямовані на організацію самостійної роботи студентів.

Призначені для студентів спеціальності 121 "Інженерія програмного забезпечення" для практичного використання при виконанні лабораторних робіт.

© КНУБА, 2024

Зміст

Загальні положення 4
Лабораторна робота №1 5
Лабораторна робота №2 9
Лабораторна робота №3 14
Лабораторна робота №4 18
Лабораторна робота №5 24
Лабораторна робота №6 30
Лабораторна робота №7 34
Література. 40

Загальні положення

Лабораторні роботи є логічним продовженням лекційного курсу з дисципліни “GRID-системи та хмарні технології” і є перехідною ланкою від теоретичного курсу до набуття практичних навичок роботи з хмарними технологіями платформи Azure.

Мета курсу є навчити студентів використовувати основні прийоми створення розподілених застосувань з урахуванням сучасних концепцій і тенденцій розвитку хмарних технологій.

Під час вивчення курсу студенти набувають знань та умінь фахівець використання Грід – технології для вирішення складних науково – практичних завдань.

Кожна лабораторна робота містить наступні види робіт:

- аналіз умови задачі;
- виконання задача в хмарному середовищі Azure;
- демонстрацію виконаного завдання;
- відповіді на контрольні запитання;
- складання і захист звіту.

Завершивши вивчення дисципліни здобувач повинен вміти створювати віртуальні машини (ВМ) на платформі Microsoft Azure за допомогою Azure Services та Azure Cloud Shell; підключатись до віртуальної мережі Azure методом «точка – мережа»; створювати канали ExpressRoute для підключень між локальною мережею та «хмарою» Майкрософт; отримувати зображення Докера та розгортати екземпляр контейнера в Azure за допомогою порталу Azure; повинен вміти працювати зі сховищами даних Azure та вміти працювати з Azure Data Factory.

Лабораторна робота №1
Створення віртуальної машини (ВМ) на платформі Microsoft Azure за допомогою Azure Services та Azure Cloud Shell

Мета роботи: Здобути навики створення ВМ на порталі Azure за допомогою Azure Services та Azure Cloud.

Завдання

1. Створити віртуальну машину на порталі Azure.
2. Підключитися до віртуальної машини за допомогою протоколу віддаленого робочого столу (remote desktop protocol - RDP).
3. Створіть другу віртуальну машину на порталі Azure за допомогою Azure Cloud Shell.
4. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання..

Теоретичні відомості

Віртуальні машини (ВМ) Azure - це масштабовані хмарні обчислювальні ресурси, призначені для полегшення процесу міграції існуючих Windows Server додатків в "хмару". Створення віртуальної машини на порталі Azure пов’язано з вивченням її конфігурації, а саме: регіону, розміру, типу сховища даних, конфігурації мережі, правил груп безпеки.

Регіон – це група «доступних зон» в яких знаходяться центри обробки даних. До рекомендованих регіонів належать регіони, в яких доступні майже всі сервіси Azure (такі регіони підтримують три різні доступні зони). Альтернативні регіони це регіони з функціями «аварійного відновлення» (не підтримують різні доступні зони).

Розміри віртуальних машин згруповані за категоріями, починаючи із серії «B» для найпростішого тестування та закінчуючи серією «H» для складних обчислювальних завдань. Розмір віртуальної машини слід вибирати відповідно до необхідного робочого навантаження. Розмір віртуальної машини можна змінити після створення, але для цього необхідно спочатку завершити її роботу, тому краще відразу вибрати правильний розмір, якщо це можливо.

За умовчанням для віртуальної машини Windows створюються два віртуальні жорсткі диски:

1) диск операційної системи - це основний диск або C з максимальною ємністю 2048 ГБ;

2) тимчасовий диск – цей диск призначений для тимчасового зберігання ОС чи додатків, позначається літерою D і за замовчуванням його розмір залежить від розміру віртуальної машини.

Під час створення віртуальної машини можна вибрати існуючу віртуальну мережу у вибраному регіоні або створити нову. Створення мережі разом із віртуальною машиною – найпростіший варіант, але в більшості ситуацій він не ідеальний. Найкраще спланувати вимоги до мережі для всіх компонентів архітектури і створити структуру віртуальної мережі окремо. Після цього можна створити віртуальні машини та помістити їх у вже створені віртуальні мережі.

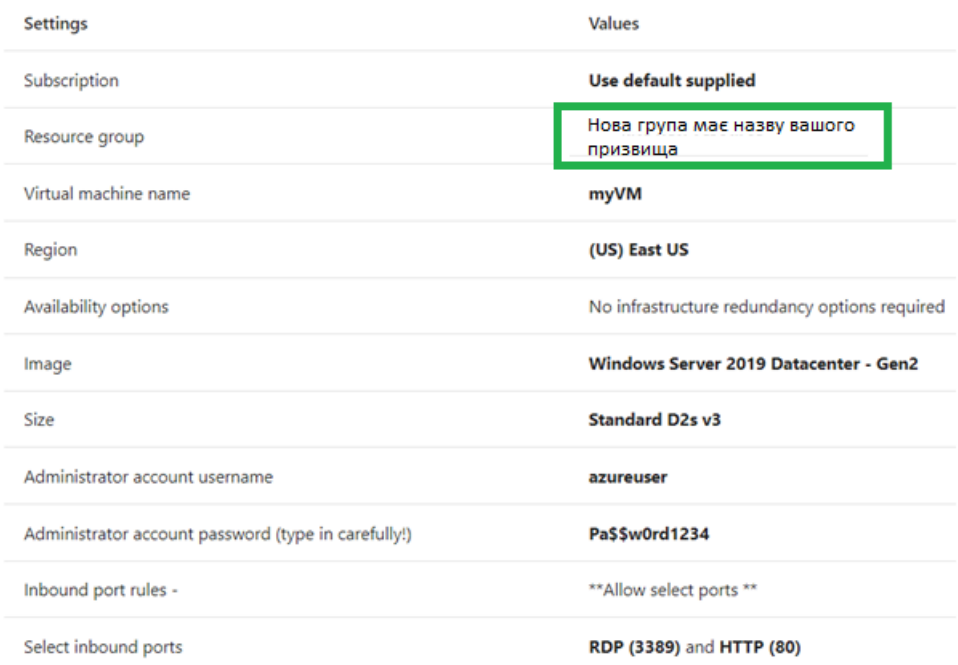
Віртуальна мережа необхідна для:

1) обміну даними між ресурсами Azure;
2) взаємодії із локальними ресурсами;
3) фільтрування мережного трафіку;
4) маршрутизації мережевого трафіку;
5) інтеграції зі службами Azure.

Azure Cloud Shell - призначена для створення ресурсів Azure та керування ними з командного рядка або за допомогою скриптів. Оскільки Cloud Shell створена для інтерактивних сеансів, оболонка автоматично завершує роботу після 20 хвилин бездіяльності. Azure Cloud Shell дозволяє створити та розгорнути ВМ за допомогою низки команд.

Хід роботи

1. Зайдіть на портал Azure - https://portal.azure.com
2. Створіть віртуальну мережу – визначте ім’я вузла «Бастіон»; початкова адреса – залиште за замовченням 10.0.0.0; розмір під-мережі – залиште за замовченням /24(256 адресів).
3. Створіть віртуальну машину - у колонці «Усі служби» в меню порталу знайдіть і виберіть «Віртуальні машини», і натисніть «Створити» та виберіть «Віртуальна машина Azure» зі спадного списку.
4. На вкладці «Основні» введіть інформацію з рис. 1



Settings	Values
Subscription	**Use default supplied**
Resource group	Нова група має назву вашого призвища
Virtual machine name	**myVM**
Region	**(US) East US**
Availability options	No infrastructure redundancy options required
Image	**Windows Server 2019 Datacenter - Gen2**
Size	**Standard D2s v3**
Administrator account username	**azureuser**
Administrator account password (type in carefully!)	**Pa$$w0rd1234**
Inbound port rules -	**Allow select ports **
Select inbound ports	**RDP (3389)** and **HTTP (80)**

Рис. 1. Інформація при створенні ВМ

5. Перейдіть на вкладку «Мережа» та визначте наступні параметри:

1) визначте створену вами віртуальну мережу;

2) визначте підмережу;

3) в розділі «Вибрати вхідні порти» визначте HTTP (80) і RDP (3389).

6. Перейдіть на вкладку «Керування» та в її розділі «Моніторинг» виберіть таке налаштування: «вимкнути діагностику завантаження». Залиште решту значень за замовчуванням, а потім натисніть кнопку «Переглянути + створити» внизу сторінки.

7. Після проходження перевірки натисніть кнопку Створити.

8. «Перейти до ресурсу», коли розгортання машини закінчиться успішно та натисніть «Підключити» та виберіть RDP зі спадного списку.

9. На сторінці «Підключення до віртуальної машини» збережіть параметри за замовчуванням для з'єднання з загальнодоступною IP-адрес через порт 3389 і натисніть «Завантажити файл RDP».

10. Відкрийте завантажений файл RDP (розташований у нижній лівій частині вашої лабораторної машини) і натисніть «Підключитися», коли з'явиться запит.

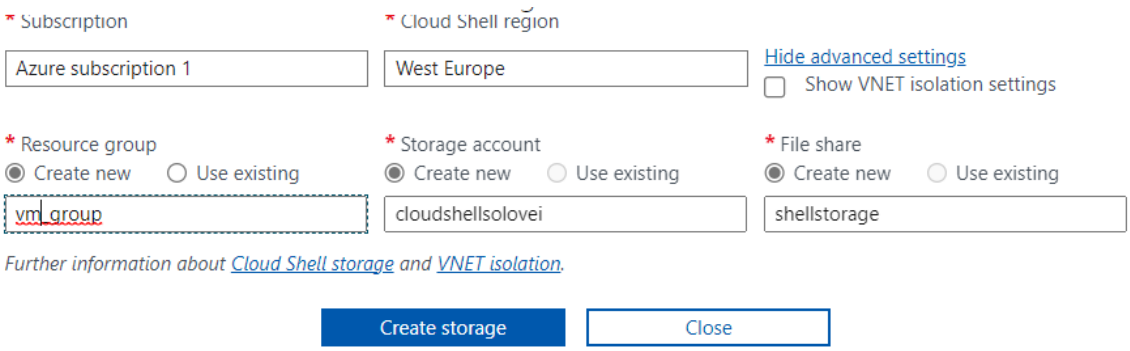
11. У вікні безпеки Windows увійдіть, використовуючи облікові дані адміністратора, які ви використовували під час створення віртуальної машини azureuser, і пароль Pa$$w0rd1234.

12. На порталі Azure відкрийте Azure Cloud Shell, за допомогою значки у верхньому правому куті порталу Azure.

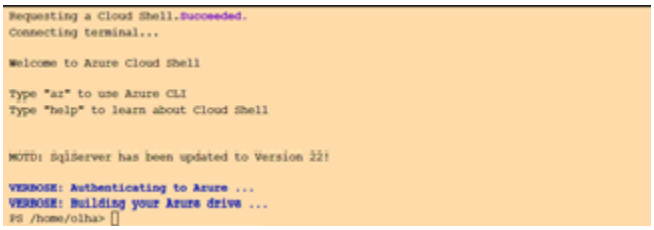


13. У вікні «Welcome to Azure Cloud Shell» виберіть «Cloud Shell».

14. У вікні «You have no storage mounted» натисніть “Advanced settings” і заповніть як показано нижче. Поле «storage account» має включати ваше прізвище.



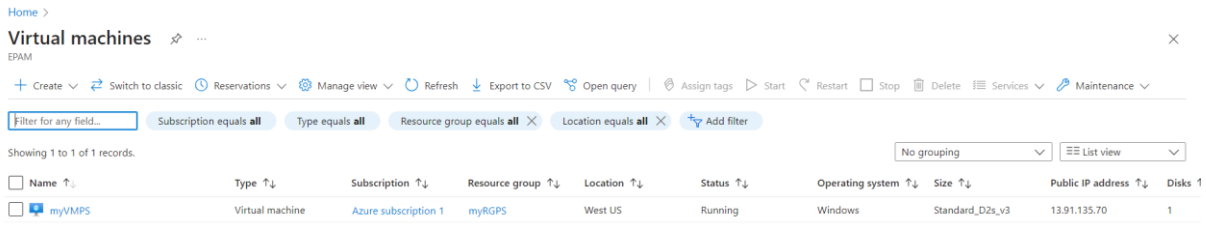
15. Отримуйте підтвердження в PowerShell: «Welcome to Azure Cloud Shell».



16. Створіть віртуальну машину через набір команд New-AzVm -ResourceGroupName "myRGPS"-Name "myVMPS"-Location "East US"-VirtualNetworkName "myVnetPS"-SubnetName "mySubnetPS"-SecurityGroupName "myNSGPS"-PublicIpAddressName "myPublicIpPS".

17. Коли буде запропоновано, введіть ім’я користувача (azureuser) і пароль (Pa$$w0rd1234). Процес створення ресурсів для віртуальної машини буде розпочати і закінчено повідомленням «ProvisioningState: Succeeded».

18. Після створення віртуальної машини закрийте панель Cloud Shell сеансу PowerShell. На порталі Azure знайдіть віртуальні машини та переконайтеся, що ВМ запущено. Це може зайняти кілька хвилин.



19. На порталі Azure відкрийте Azure Cloud Shell, клацнувши значок у верхньому правому куті порталу Azure. Отримайте інформацію про вашу віртуальну машину, включаючи назву, групу ресурсів, розташування та статус, виконавши команду: Get-AzVM -name myVMPS -status | Format-Table –autosize

Контрольні запитання.

1. Яка роль віртуальної машини Azure?
2. Що означає термін «доступна зона»?
3. На які категорії поділяються регіони Azure?
4. Поясність для чого використовують «домени збою» та «домени оновлення»?
5. Поясніть за яким принципом визначається розмір ВМ Azure?
6. Поясніть конфігурацію сховищ ВМ Azure за замовченням?

Лабораторна робота №2
Підключення до віртуальної мережі Azure методом «точка – мережа»

Мета роботи: Здобути навики створювати підключення до віртуальної мережі Azure методом «точка – мережа».

Завдання

1. Створити віртуальну мережу з VPN-шлюзом та виконати підключення «віддаленого робочого місця» до віртуальної машини в хмарному середовищі Azure методом "точка-мережа".
2. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання.

Теоретичні відомості

Локальний комп'ютер можна підключити до віртуальної мережі за допомогою з’єднання типу «точка – мережа», через створення VPN-шлюзу. Конфігурація з’єднання типу «точка – мережа» представлена на рисунку 1 і складається з: Azure VPN, Client VPN або OnPrem VPN – для встановлення віртуального приватного мережевого з’єднання; P2S VPN Tunnel, IPSec/IKEv2 VPN Tunnel - шлюзів для передачі даних; Radius Server, Domain Server – сервери, дані з яких використовують при виконанні аутентифікації клієнтів.

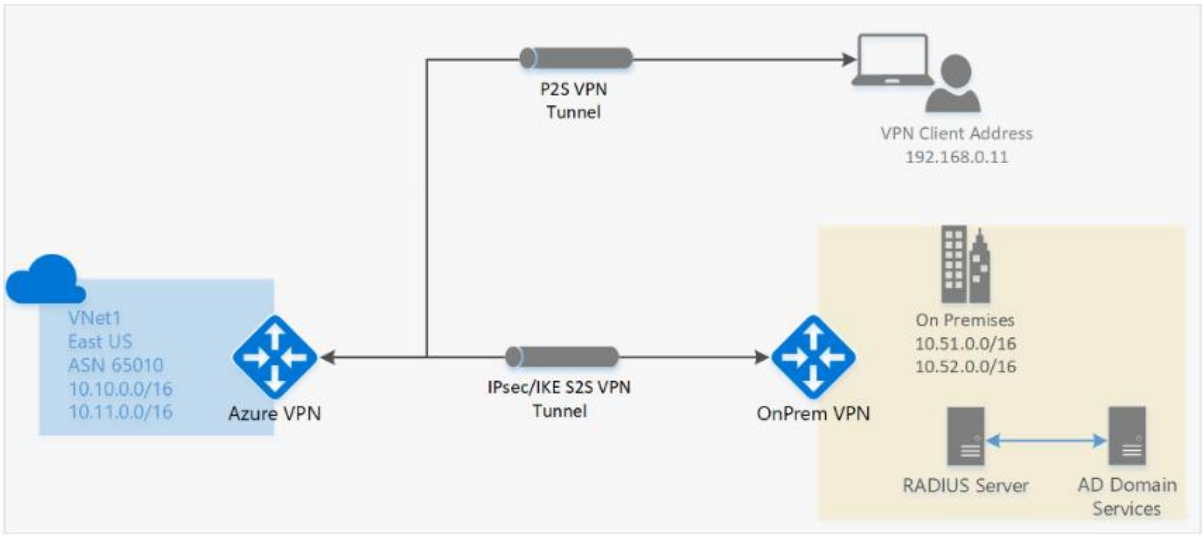


Рис. 1. Конфігурація з’єднання типу «точка – мережа»

Обмін даними між комп'ютером та віртуальною мережею здійснюється через Інтернет за допомогою зашифрованого тунелю. Підключення типу «точка – мережа» можливе за одним з протоколів: OpenVPN - на основі SSL/TLS, підключення дозволяє трафіку проходити через брандмауер - порт 443; SSTP (Secure Socket Tunneling Protocol)-підключення за протоколом TLS (Transport Layer Security), підходить тільки для пристроїв Windows; IPSec/IKEv2 VPN — рішення VPN на основі стандартів Ipsec (IP Security - на відміну від SSL та TLS, працюють на мережевому рівні; IKE (Internet Key Exchange) - стандартний протокол набору протоколів IPsec, який використовується для забезпечення безпеки взаємодії в віртуальних приватних мережах). IKEv2 VPN можна використовують для пристроїв - на платформі Mac.

На момент встановлення підключення типу «точка – мережа» дані клієнта перевіряються VPN шлюзом. Аутентифікація клієнтів виконується трьома способами:

1) перевірка сертифікату Azure клієнта;

2) перевірка дійсного облікового запису клієнта Azure, тобто перевірка Microsoft Entra ID;

3) перевірка облікового домену організації. Для перевірки облікового запису домену організації – додатково використовується сервер RADIUS, який інтегрований з сервером Azure з обліковими записами доменів організацій.

Вимоги до конфігурації клієнта залежать від налаштувань VPN-клієнта, в таблиці 1 наведені конфігурації для способів аутентифікації способами 1-2.

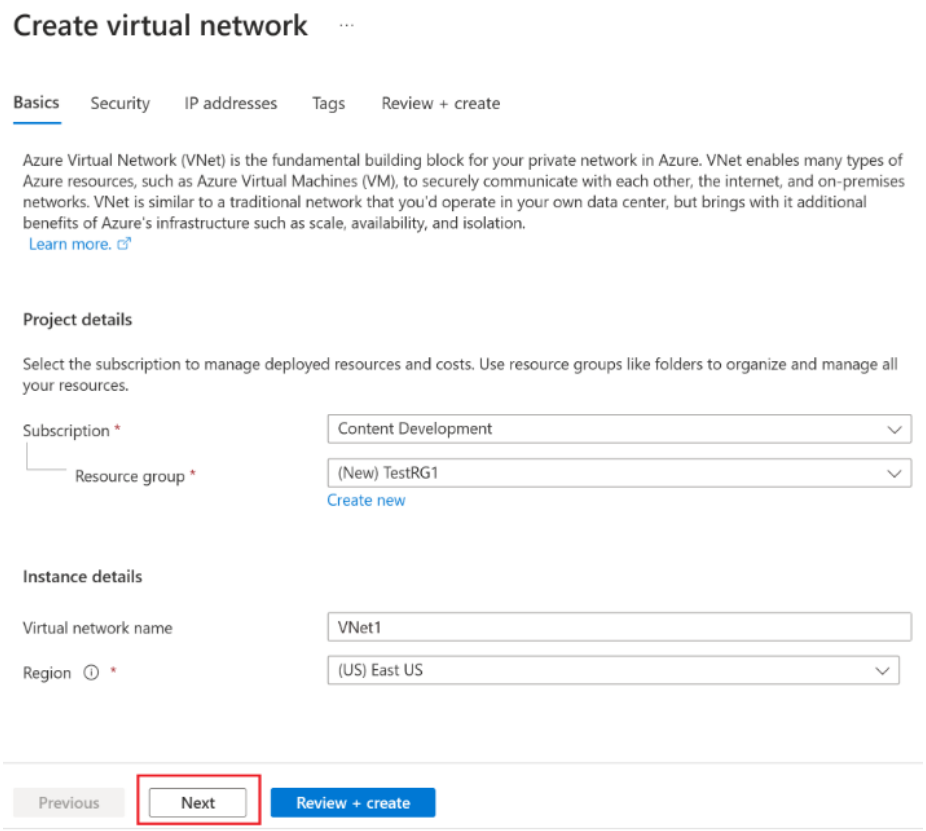
Таблиця 1.

Вимоги до конфігурації клієнта

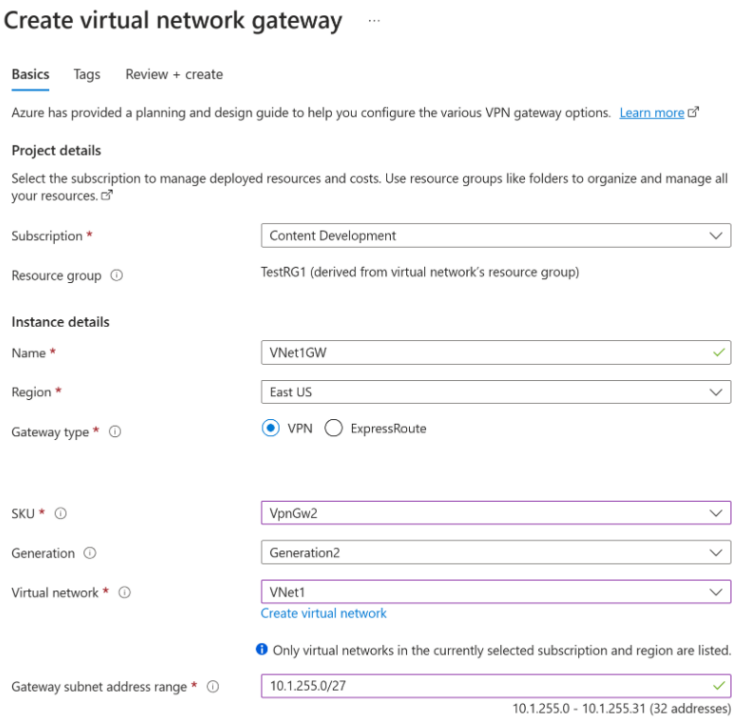
Спосіб аутентифікації	Тип тунелю	Файлова конфігурація	Налаштування VPN-клієнта
Сертифікат Azure	IKEv2, SSTP	Windows	власний VPN-клієнт
Сертифікат Azure	OpenVPN	Windows	Клієнт OpenVPN VPN-клієнт Azure
Сертифікат Azure	IKEv2, OpenVPN	macOS-iOS	macOS-iOS
Сертифікат Azure	IKEv2, OpenVPN	Linux	Linux
Microsoft Entra ID	OpenVPN (SSL)	Windows	Windows
Microsoft Entra ID	OpenVPN (SSL)	macOS	macOS

Хід роботи

1. Зайдіть на портал Azure - https://portal.azure.com.
2. Створіть віртуальну мережу зазначивши необхідні параметри: «ім’я», «регіон», «IP-адрес».

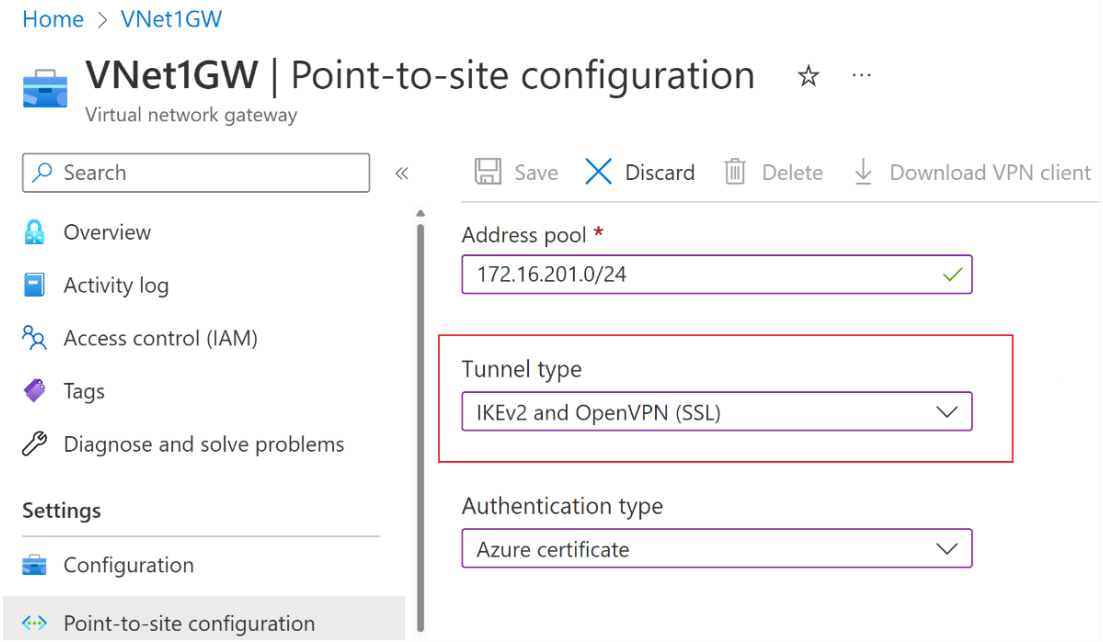


3. Створіть VPN-шлюз зазначивши необхідні параметри: «підписку», «групу», «імя», «регіон», «тип шлюзу», «номер SKU», «віртуальну мережу», «діапазон адрес під-мережі», вкажіть загальнодоступний IP-адрес, який буде зв’язуватись з VPN – шлюзом.



4. Визначте пул IP адрес – на сторінці «Конфігурація» додайте діапазон приватних IP адрес які ви хочете використовувати. VPN – клієнт буде отримувати IP-адресу з вказаного пулу. Мінімальне значення для налаштування маски: 29 біт в режимі «активний — пасивний» і 28 біт в режимі «активний — активний».

5. Визначте тип тунелю та спосіб аутентифікації.



6. Перевірте VPN підключення – для цього виконайте команду ipconfig/all. В отриманій відповіді – перевірте, що отриманий IP-адрес, це один з IP адрес з визначеного вами пулу IP адрес.

```
PPP adapter VNet1:
   Connection-specific DNS Suffix .:
   Description.....................: VNet1
   Physical Address................:
   DHCP Enabled....................: No
   Autoconfiguration Enabled.......: Yes
   IPv4 Address....................: 172.16.201.3(Preferred)
   Subnet Mask.....................: 255.255.255.255
   Default Gateway.................:
   NetBIOS over Tcpip..............: Enabled
```

7. Перевірте VPN підключення до ВМ, яку ви створили в лабораторній роботі №1. Для цього використовуйте підключення типу RDP. Далі, для підключення «віддаленого робочого місця» введіть *IP адресу ВМ.*

Контрольні запитання.

1. За якими протоколами можливе підключення «точка - мережа»?
2. При підключенні «мережа-мережа», в чому різниця налаштувань VPN- тунелю: "активний - резервний" та "активний - активний"?
3. У яких випадках рекомендовано використовувати конфігурацію паралельного підключення?
4. В якому порядку оброблюються правила підмережевого інтерфейсу та інтерфейсу мережі для вхідного та вихідного трафіків?

Лабораторна робота №3
Створення каналу ExpressRoute для підключень між локальною мережею та «хмарою» Майкрософт

Мета роботи: Здобути навики створювати канал ExpressRoute для підключень локальною мережи до віртуальної мережі Azure.

Завдання

Створити канал ExpressRoute для підключень між локальною мережею та «хмарою» Майкрософт. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання.

Теоретичні відомості

Канал ExpressRoute дозволяє підключення локальної мережі до Майкрософт «хмарних» послуг таких як Office 365, Dynamics 365 та інші через партнерів. Архітектура з’єднання каналом ExpressRout визначає наступні компоненти: партнери встановлюють підключення до кінцевої точки розташування ExpressRout шляхом пірингу локальних мереж з віртуальними – ці з’єднання називають ExpressRout Circuit (рис.1). ExpressRout Circuit – забезпечує фізичне підключення для передачі даних через «прикордонні» канали постачальника (Partner Edge) на «прикордонні» канали Майкрософт (Microsoft Edge). «прикордонні» канали Майкрософт надають доступ в область Microsoft.

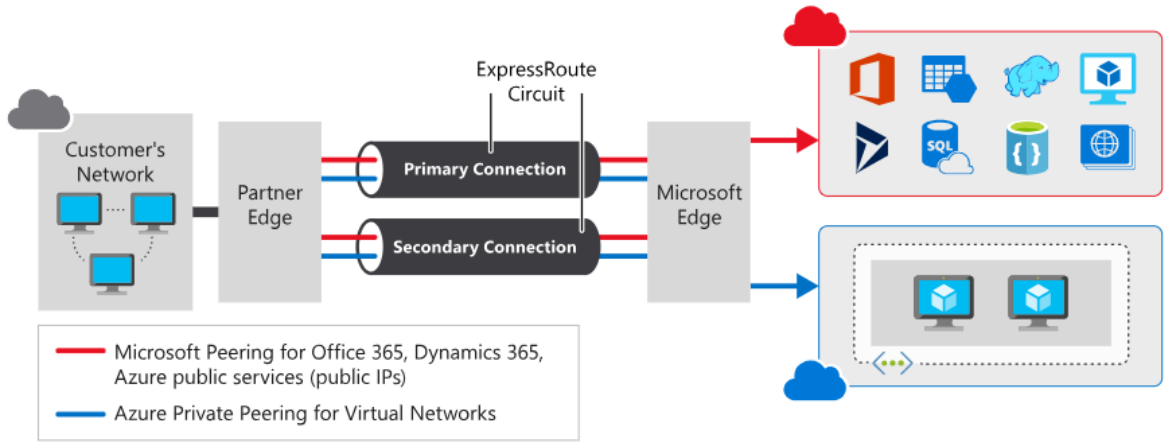


Рис. 1. Архітектура з’єднання каналом ExpressRout

Канал ExpressRoute є логічним зв'язком між локальною інфраструктурою та «хмарою» Майкрософт через постачальника послуг зв'язку. Канал, який однозначно ідентифікується стандартним GUID, називається ключем служби (s-key).

Ключ служба — це єдиний фрагмент інформації, який передається між Майкрософт, постачальником послуг підключення та локальною мережею при співвідношення між каналом ExpressRoute і ключем 1:1. Канали ExpressRoute можуть підтримувати три незалежних типу пірингу: приватний і Microsoft Azure, загальний Кожен канал має фіксовану пропускну здатність (50 Мбіт/с, 100 Мбіт/с, 200 Мбіт/с, 500 Мбіт/с, 5 Гбіт/с, 10 Гбіт/с) і підтримується з постачальником підключення та піринговим розташуванням. Вибрана пропускна здатність поширюється на всі пірингові канали. Кількість префіксів IPv4, IPv6 та діапазон IP адрес залежно від типу пірингу наведені в таблиці 1.

Таблиця 1

Кількість префіксів IPv4, IPv6 та діапазон IP адрес залежно від типу пірингу

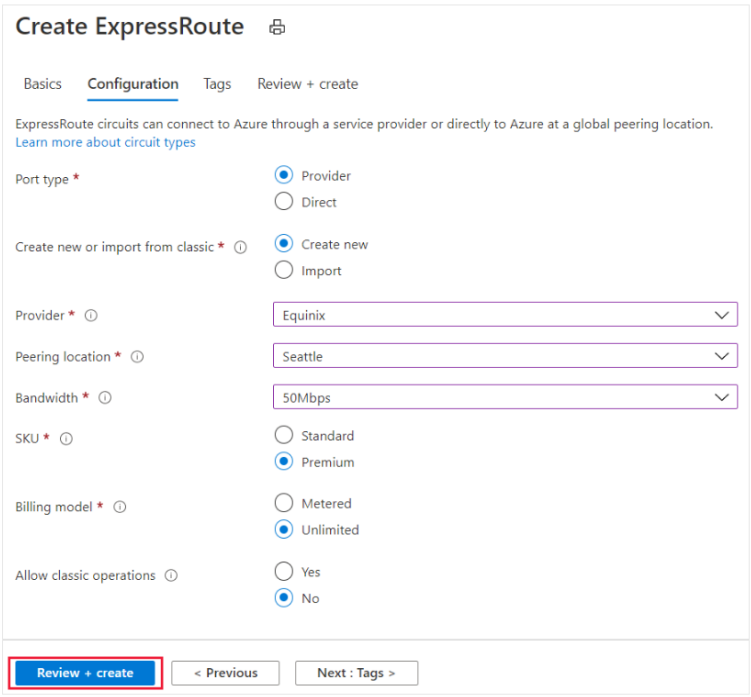
	Приватний	**Microsoft Azure**	**Загальний**
Кількість префіксів IPv4, для одного пірінгу	400 за замовчення	200	200
Кількість префіксів IPv6, для одного пірінгу	100	200	-

Продовження таблиці 1

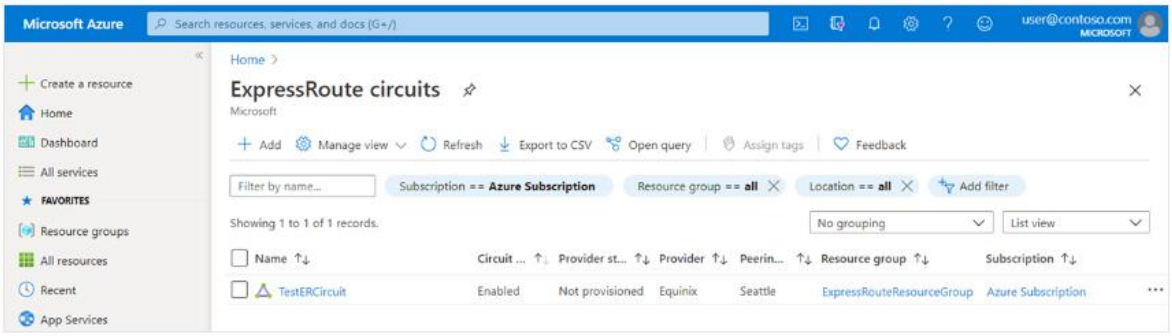
Діапазон IP адрес	допустимий IP-адрес у глобальній мережі	Загальнодоступні IP-адреси	Загальнодоступні IP-адреси
IP протокол	IPv4, IPv6	IPv4, IPv6	IPv4

Хід роботи

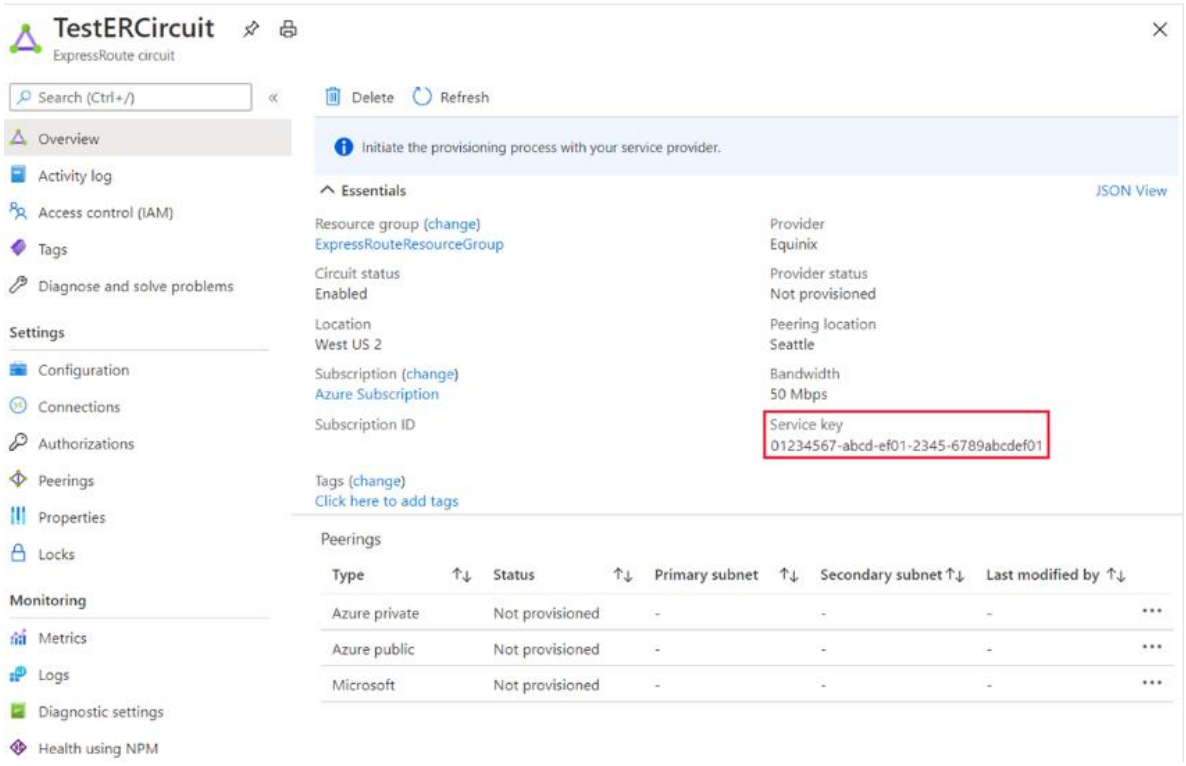
1. Зайдіть на портал Azure - https://portal.azure.com.
2. Створіть канал ExpressRout, визначивши необхідні параметри: «групу ресурсів», «ім'я», «регіон». Перейдіть в розділ конфігурації і вкажіть: «тип порту», «постачальника послуги», «розташування пірінгу», виберіть «стандартний» SKU, модель оплати послуги – «з врахуванням трафіку».



3. Створіть канал, натиснувши «Review+Create».
4. Перевірте, що канал створено – для цього перейдіть до розділу «ExpressRoute» circuits.



5. Оберіть створений вами канал, і перевірте, що для нього визначений Security ID – цей ключ передається постачальнику для отримання послуг ExpressRout.



Контрольні запитання

1. Поясніть відмінності в конфігурації типів пірингу?
2. Наведіть розміри префіксів загальнодоступних IP-адрес?
3. З чого складається архітектура паралельного підключення типу «точка-мережа» і ExpressRout?
4. Які типи «ExpressRoute» з’єднання вам відомі?
5. Які переваги «ExpressRoute» з’єднання?

Лабораторна робота №4
Отримання зображення Докера та розгортання екземпляра контейнера в Azure за допомогою порталу Azure

Мета роботи: Здобути навики отримання зображення Докера та розгортання екземпляра контейнера в Azure за допомогою порталу Azure

Завдання

1. Отримати зображення Докера.
2. Розгорнути екземпляр контейнера в Azure за допомогою порталу Azure.
3. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання.

Теоретичні відомості

Архітектура контейнеру Docker в хмарному середовищі Azure складається з Docker Hub, Docker Host (Докер Вузол) та контейнерів (рис. 1).

Сервер Docker це керуюча програма, яка називається dockerd, яка відповідає на запити клієнта за допомогою REST API Docker та може взаємодіяти з іншими програмами та відповідає за відстеження життєвого циклу контейнерів.

Docker Hub – це реєстр контейнерів Docker. *Реєстри Docker* – це репозиторії, які служать для зберігання та розповсюдження створюваних образів контейнерів. Docker Hub є загальнодоступним реєстром Docker, який використовується за умовчанням для керування зображеннями контейнера.

Зображення контейнера — це пакет, що містить програмне забезпечення, бібліотеки, конфігураційні файли і т.п. Під час запуску він стає контейнером. Зображення контейнера є незмінним. Єдиний спосіб змінити зображення це створити нове.

Докер файл – це текстовий файл, який включає інструкції щодо компіляції та запуску зображення докера.

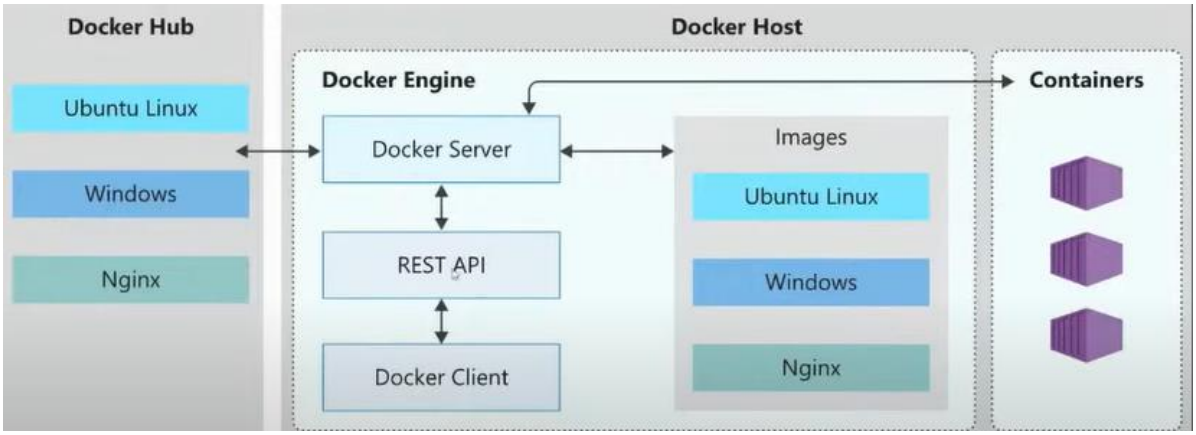


Рис. 1. Архітектура контейнеру Docker в хмарному середовищі Azure

Під час розгортання контейнеру в Azure вказується джерело зображення в конфігурації розгортання або шаблонах. Потім середовище виконання контейнера (наприклад, Docker) використовує цю інформацію, щоб отримати вказане зображення контейнера з джерела та запустити його як контейнер у вашому середовищі Azure.

Хоча контейнер використовує ядро операційної системи сервера, але не отримує необмежений доступ до ядра. Натомість, контейнер отримує ізольоване, а в деяких випадках віртуалізоване уявлення системи.

Наприклад, контейнер може звертатися до віртуалізованої версії файлової системи та реєстру, але будь-які зміни стосуються лише контейнера та видаляються при його зупинці. Щоб зберегти дані, контейнер може підключити постійне сховище.

В таблиці 1 наведені типи програм, які не можна перемістити в контейнер Windows. В таблиці 2, навпаки, визначені типи програм «ідеальні» для контейнеризації.

Таблиця 1

Типи програм, які не можна перемістити в контейнер Windows

Типи програм	**Причини**
Додатки, для яких потрібні можливості робочого столу	Контейнери не підтримують графічний інтерфейс користувача (GUI), навіть якщо в самому додатку немає графічного інтерфейсу користувача, але є установник який його використовує.
Програми, що використовують	Оскільки протокол віддаленого робочого столу (RDP) призначений для створення

протокол віддаленого робочого столу (RDP)	інтерактивного візуального сеансу, обмеження графічного інтерфейсу користувача, описане вище, застосовується і до нього.
Програми з базами даних	Контейнери не передбачають збереження даних. Такі типи програм можна контейнеризувати лише в тому випадку, якщо ви ізолюєте потрібні дані з одного сеансу в наступному та зберігаєте їх у постійному сховищі.
Програми, які використовують .NET Framework версії 2.0 або пізнішої	Для підтримки .NET Framework потрібні певні образи контейнерів, при цьому підтримуються лише пізніші версії, а версії, які передують 2.0, не підтримуються в принципі.

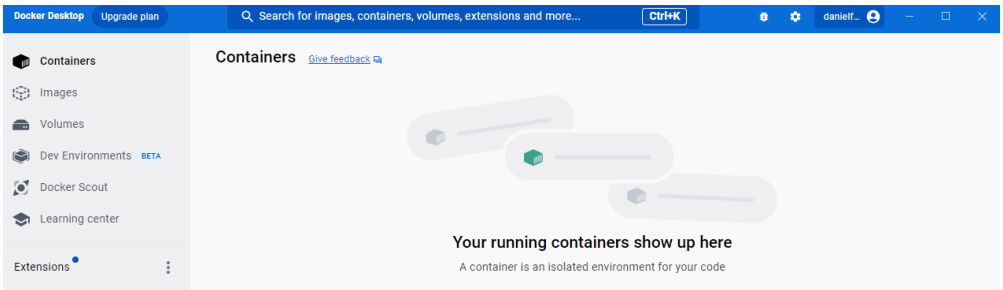
Таблиця 2

Типи програм, які «ідеальні» для контейнеризації

Типи програм	**Причини**
Консольні програми	Консольні програми, які не мають обмежень за графічним інтерфейсом користувача
Служби Windows	Оскільки це фонові процеси, які не потребують прямої взаємодії з користувачем
Веб-програми	Веб-програми по суті є фоновими службами, які прослуховують певний порт, і тому є відмінними кандидатами для контейнеризації, тому що вони можуть використовувати переваги масштабованості, пропоновані контейнерами.

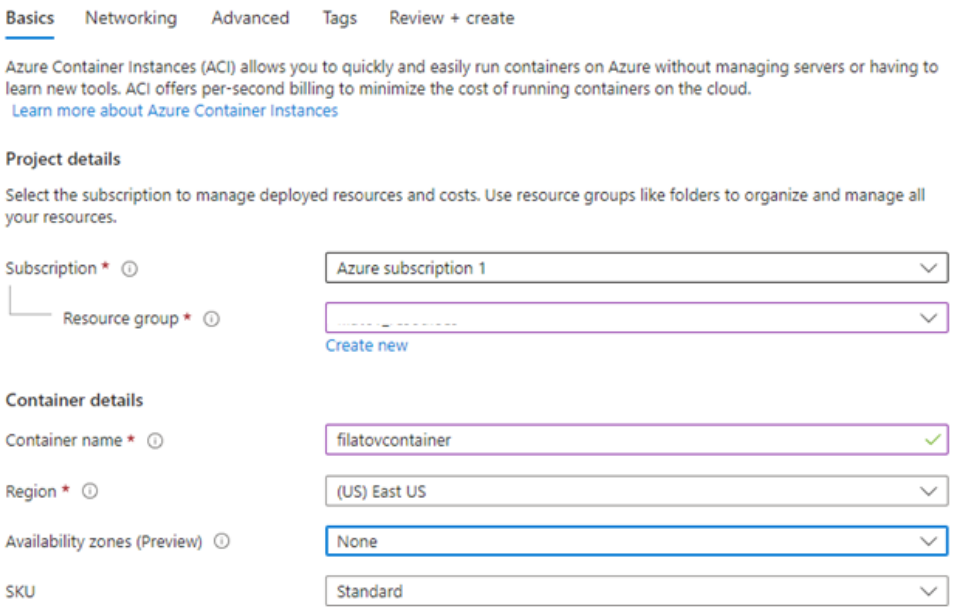
Хід виконання роботи

1. Розгорніть Docker на своєму комп’ютері.

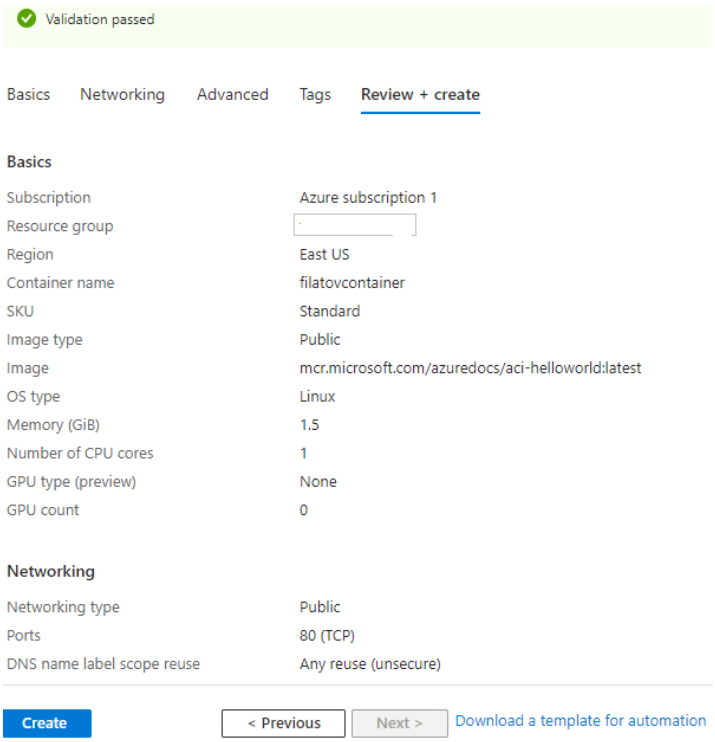


2. Увійдіть на портал Azure - https://portal.azure.com.

3. Створіть екземпляр контейнеру- на вкладці “Основи” (Basics) вкажіть необхідні параметри для контейнера. (ім’я контейнера має включати ваше прізвище; Мережа – порт – 80, порт протокол – TCP)

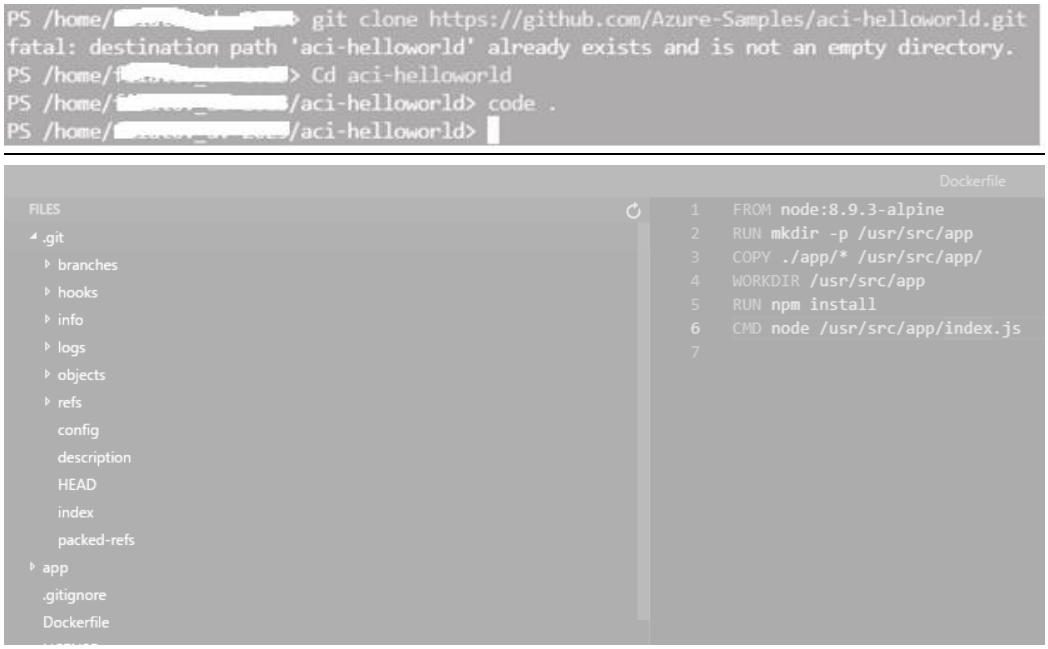


4. Натисніть “Перевірити+Створити”, щоб розпочати процес автоматичної перевірки. Після завершення перевірки, натисніть «Створити».

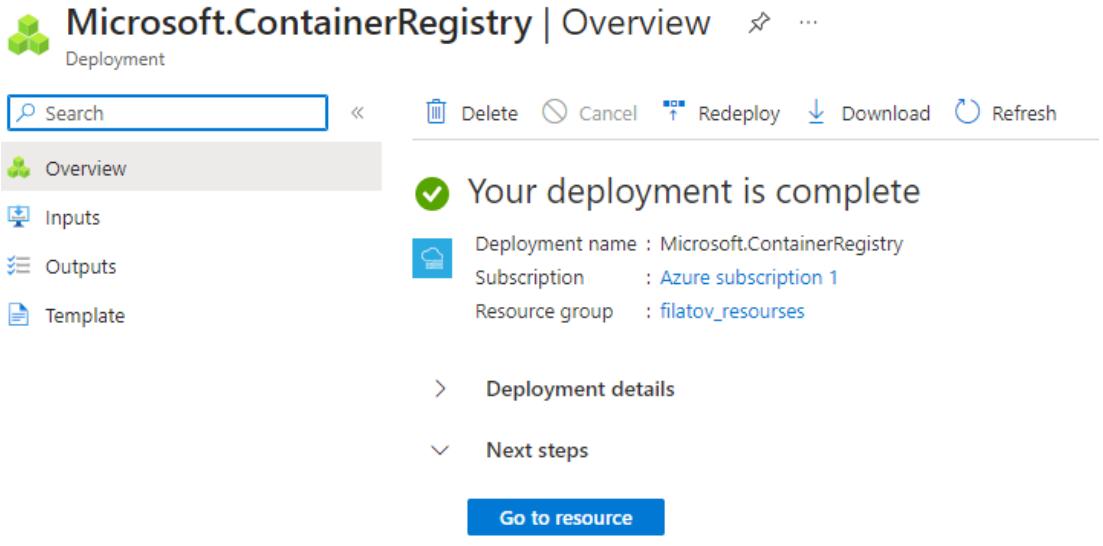


5. «Перейдіть до ресурсу», скопіюйте IP адресу та виконайте в новому браузері – маєте побачити привітання «Welcome to Azure Container Instances»!

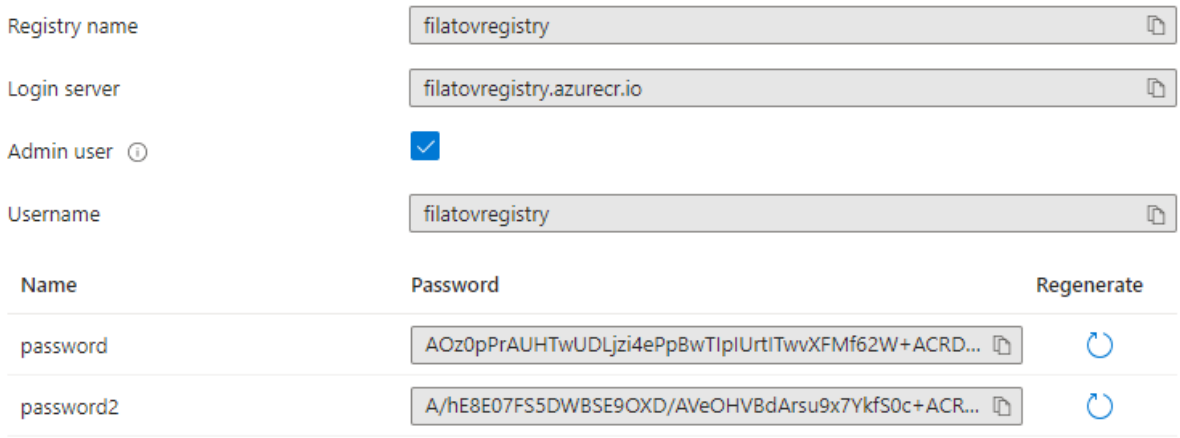
6. Перегляньте DockerFile для цього в PowerShell виконайте наступні команди, git clone https://github.com/Azure-Samples/aci-helloworld.git. Зайдіть в папку - **Cd aci-helloworld.** Виконайте команду - **Code**



7. Створіть «Реєстр Контейнерів» для цього на панелі «Усі служби» знайдіть і виберіть «Реєстр Контейнерів» (container registry) і натисніть «Створити».



8. Перейдіть до ресурсу та в меню Налаштування («Settings») активуйте параметр «Адміністратор» (Admin user), в результаті відобразиться ім'я користувача та пароль для реєстру контейнерів. Запишіть Ім'я реєстру (Registry name), Сервер входу (Login server), Ім'я користувача (user name) та Пароль (password) для реєстру контейнерів.



9. В PowerShell виконайте наступні команди, попередньо замінивши <registry-name> на ваше ім’я реєстру.

```
docker tag reservationsystem:latest <registry-name>.azurecr.io/reservationsystem:latest
```

10. Перевірити, що «зображення» створене введіть команду.

```
Docker image ls
```

11. В результаті на екрані відобразиться.

```
REPOSITORY TAG IMAGE ID CREATED SIZE
reservationsystem latest a56281e7038f 4 minutes ago 1.76GB
<registry-name>.azurecr.io/reservationsystem latest a56281e7038f 4 minutes ago 1.76GB
```

12. Вкажіть сервер входу виконавши команду - docker login <login-server>.

13. Поверніться на портал Azure - https://portal.azure.com.

14. Перейдіть до вашого ресурсу «Реєстр Контейнерів» і зайдіть в меню репозиторій – маєте побачити зображення «reservationsystem».

Контрольні запитання.

1. Що таке Докер? З яких компонентів складається архітектура Докера?

2. Для чого призначений екземпляр контейнеру?

3.Чи треба робити зміни в коді програмного доданку при його контейнеризації? Якщо ні, то яка технологія це забезпечує?

4.У чому принципова різниця між ВМ і контейнерами?

5.Чи можна контейнеризувати доданок з графічним інтерфейсом? Доданок з базою даних?

6.Які програми є «ідеальними» для контейнеризації?

7.Чим відрізняються зона DNS від домену DNS?

8.Що означає захоплення піддомену? Як уникнути захоплення піддомену?

Лабораторна робота №5.
Робота зі сховищами даних Azure

Мета роботи: Здобути навики роботи зі сховищами даних Azure

Завдання

1. Створити обліковий запис сховища Azure.
2. Побудувати процес відправки даних протоколом HTTP в хмарне сховище та збереження отриманих даних у хмарному сховище.
3. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання.

Теоретичні відомості

Azure об'єднує чотири служби сховища даних: BLOB-об'єкти Azure; файли Azure; черги Azure; таблиці Azure. Об'єднання служб даних в обліковий запис зберігання дозволяє керувати ними як групою. Параметри які вказуються під час створення облікового запису, або зміни які вносяться після створення, застосовуються до всіх служб облікового запису. При видаленні облікового запису зберігання видаляються всі дані, що зберігаються всередині.

Обліковий запис зберігання визначає політику, яка застосовується до всіх служб сховища в обліковому записі. Так, параметр «реплікація» - визначає стратегію копіювання даних для захисту від збоїв обладнання та стихійних лих. Мінімальна реплікація називається локально надлишковим сховищем (LRS), що забезпечує захист від збоїв обладнання, але не від події, здатної вивести з ладу весь центр обробки даних. Більш надійна альтернатива - геонадлишкове сховище (GRS), при цьому налаштуванні копіювання даних виконується в різні центри обробки даних по всьому світу.

Параметр рівень доступу визначає як швидко дані можуть бути отримані з «хмарного» сховища.

«Гарячий рівень доступу» оптимізований для зберігання даних до яких часто звертаються або які часто змінюють. «Холодний рівень доступу» оптимізовано для зберігання даних до яких рідко звертаються або змінюють. «Гарячий рівень доступу» підтримується лише для великих двійкових об'єктів, він задається за замовченням.

Параметр «Потрібне безпечне переміщення» визначає прийом запитів лише від безпечних з'єднань при цьому будь-які запити, виконані за протоколом HTTP, будуть відхилені. При включенні зворотного видалення великих об'єктів для облікового запису зберігання необхідно вказати термін зберігання віддалених об'єктів в діапазоні від 1 до 365 днів. Термін зберігання показує, як довго дані залишаються доступними після видалення або перезапису. Годинник для зберігання запускається в момент видалення або перезапису об'єкта.

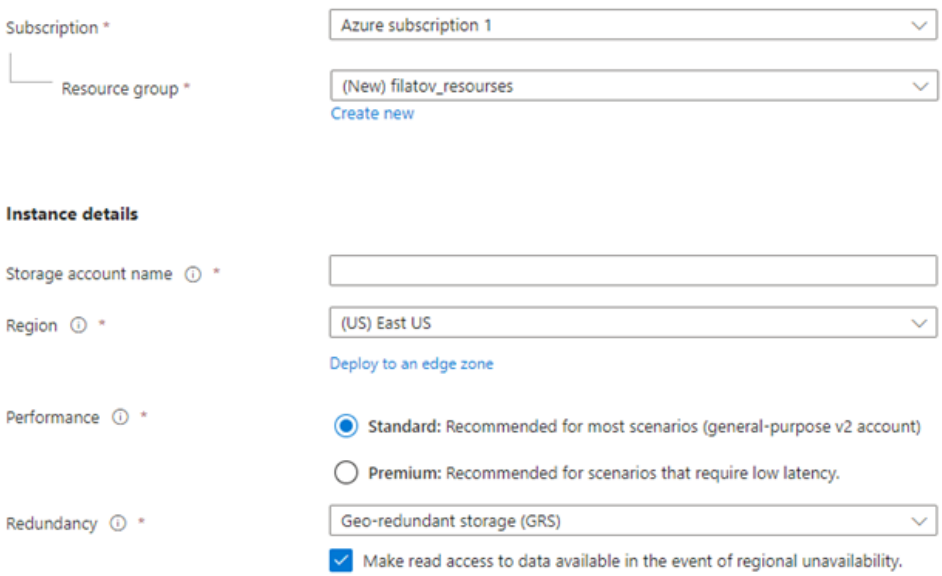
Azure Logic Apps — це застосунок, який дозволяє автоматизувати робочі процеси та інтегрувати різні програми, служби та системи в хмарних і локальних середовищах. Logic Apps легко інтегрується з іншими службами хмарного середовища Azure такими як функції Azure, службова шина Azure, база даних SQL Azure тощо це дозволяє створювати складні робочі процеси, поєднуючи потужність кількох компонентів Azure.

Створення складного робочого процесу в застосунку Azure Logic Apps починається з визначення способу, яким буде ініційовано початок робочого циклу, тобто тригеру. Тригер представляє подію або умову, яка запускає робочий процес, наприклад, тригери типу: «Коли надходить електронний лист», «Коли новий елемент додається до списку SharePoint» або «Коли надходить запит HTTP» починають робочий цикл як тільки відбулась

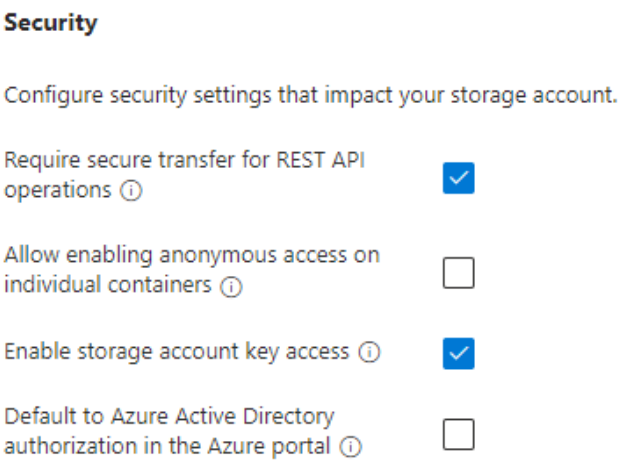
визначена подія. Наступними елементами робочого циклу є дії та з’єднувачі, які визначають сам процес та його кінцевий результат.

Хід виконання роботи

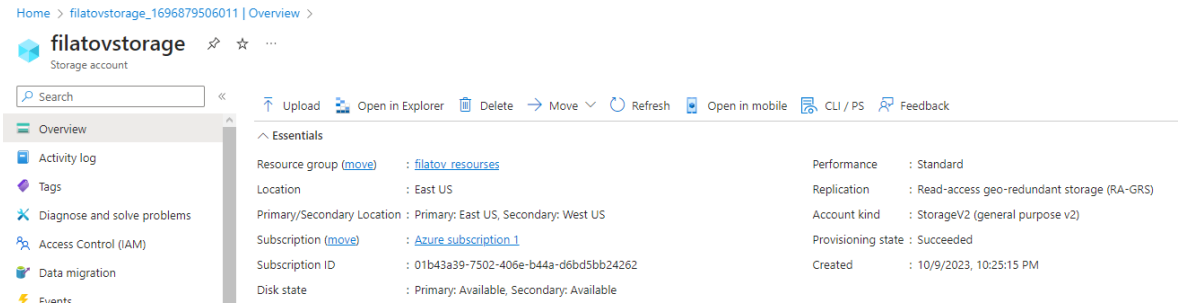
1. Увійдіть на портал Azure.
2. На панелі «Усі служби» знайдіть і виберіть «Служба сховища» (Storage Account) і натисніть «Створити».



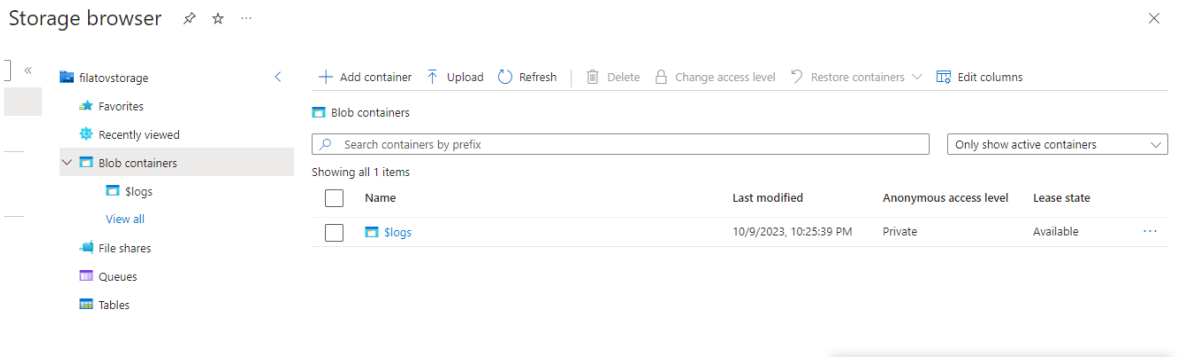
3. Надайте основні відомості для створення нової служби сховища (storage account name – має включати ваше прізвище). В розділі «Безпека» (“Security”) виберіть "Enable storage account key access"; в розділі – «Мережі» ("Network connectivity") виберіть "Enable public access from all networks".



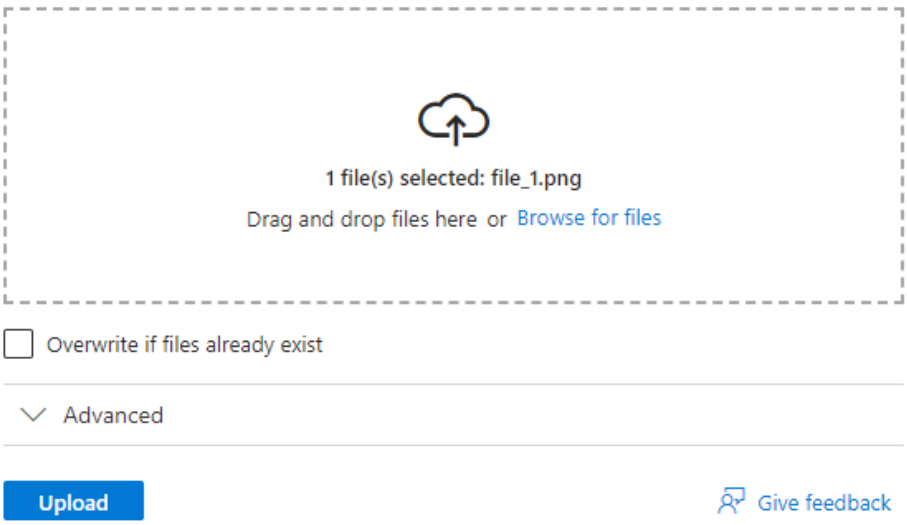
4. Натисніть “Переглянути та створити”, щоб розпочати процес автоматичної перевірки. Після завершення розгортання перейдіть за посиланням «Перейти до ресурсу».



5. Перейдіть до меню «Storage Browser/Blob containers».



6. Створіть власний Blob Container, визначне рівень доступу – «anonymous read access for containers and blobs». Завантажте файл у створений контейнер.



7. Скопіюйте URL адресу завантаженого файлу, та виконайте її в новому вікні браузера – файл має стати доступний для читання та скачування.

8. Створить власну таблицю в «Службі сховища». Таблиця має включати, щонайменше 3 рядки.

9. Створить процес в Logic Apps конструкторі - додайте метод «When a HTTP request is received» та визначте схему JSON, яка буде описувати ваш файл з даними, METHOD=POST. Додайте метод «Insert or Merge Entity» та визначте всі обов'язкові поля (рис.1).

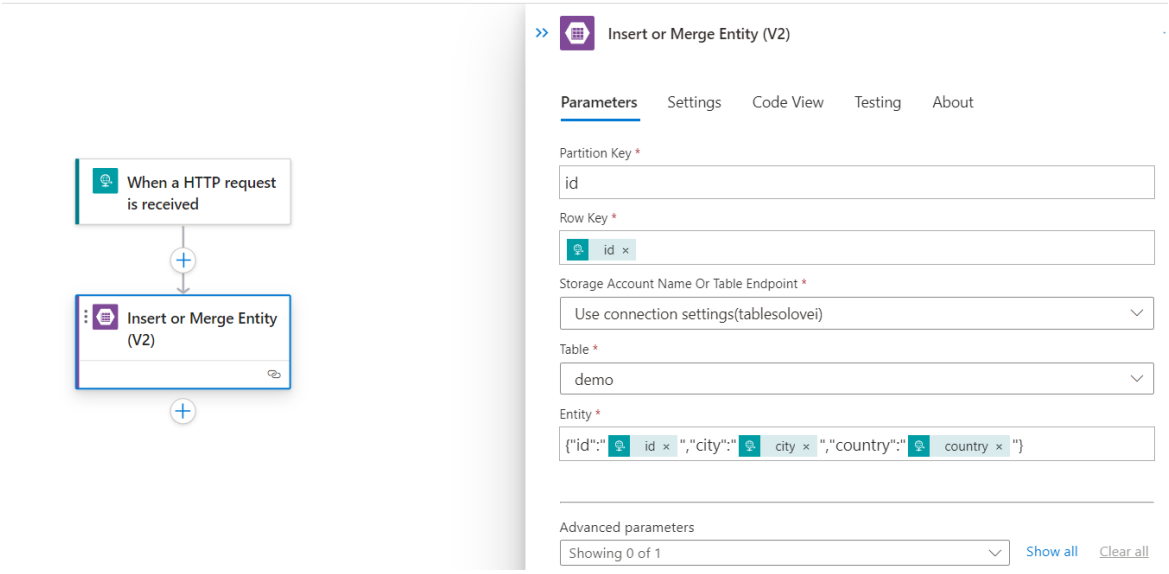


Рис. 1. Приклад для визначення методу «Insert or Merge Entity»

10. Збережіть створений процес. Після збереження скопіюйте HTTP POST URL.

11. Відкрийте Postman Web та заповніть всі дані для нового POST request. (рис 2). Натисніть «Send» та впевнитесь, що Status = 202.

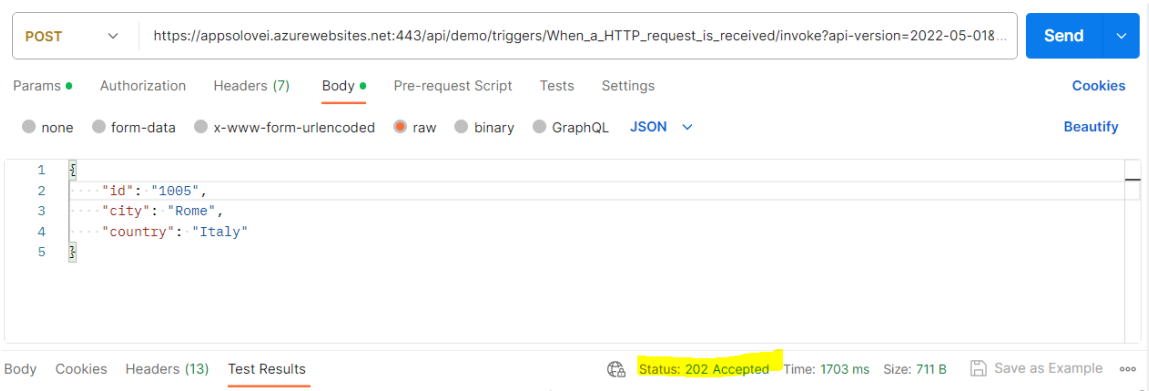


Рис. 2. Приклад для визначення методу POST в Postman Web.

12. Відкрийте Logic Apps Azure Overview Run History та перевірте, що процес виконано без помилок (рис.3).

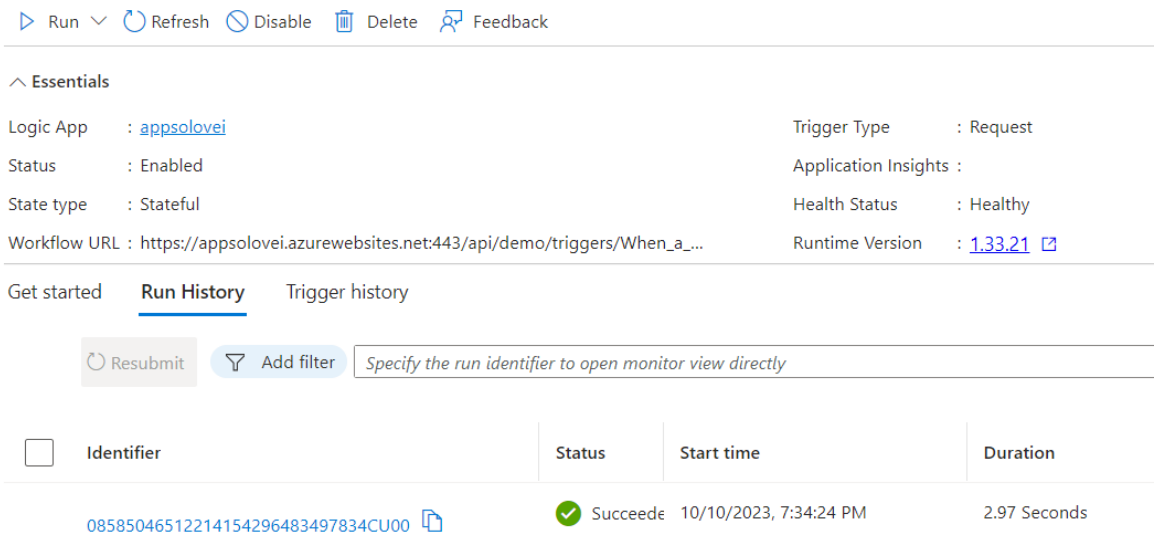


Рис. 3. Перевірка статусу виконаного процесу в Logic Apps Azure Overview Run History

13. Перейдіть до сховища даних, перевірте що запис додано чи оновлено (рис 4).

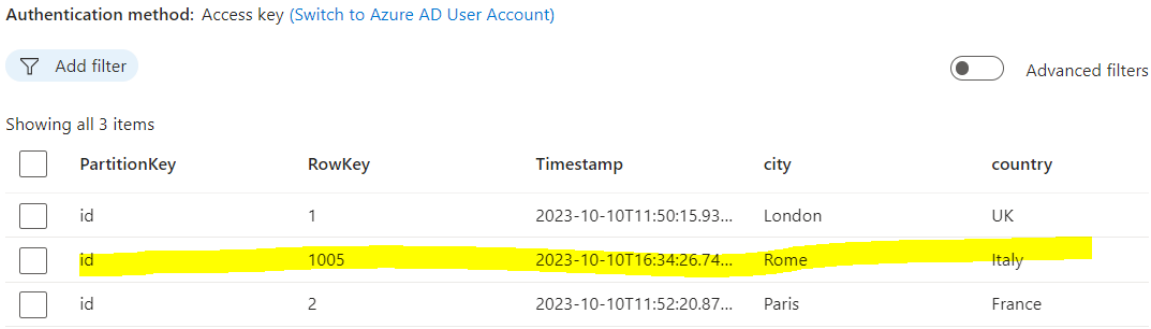


Рис. 4. В базу даних додано новий запис з RowKey=1005

Контрольні запитання.

1. Які 4 служби об’єднує сховище Azure?
2. Який параметр облікового запису зберігання визначає які служби даних можна використовувати у вашому обліковому записі зберігання?
3. Які типи стратегій копіювання даних для захисту від збоїв обладнання та стихійних лих включають параметр облікового запису зберігання? У чому різниця між типами?

4. Які доступні рівні доступу до даних в обліковому записі зберігання? Коли треба застосовувати якій рівень?

5. Які три параметри застосовуються до самого облікового запису, а не до даних у ньому?

6. Протягом якого терміну можна відновити видалені дані, якщо, при створенні облікового запису було визначено термін зворотного видалення 7 днів?

7. Є два відеофайли, які зберігаються як великі двійкові об'єкти. Один із відеофайлів критично важливий, інший відеофайл – ні. Скільки облікових записів зберігання Вам потрібно створити і з якими типами стратегії копіювання даних для захисту від збоїв?

Лабораторна робота №6.
Робота з Azure Cosmos DB

Мета роботи: Здобути навики роботи з Azure Cosmos DB за допомогою мови запитів SQL і клієнтських бібліотек для .NET, JavaScript, Python і Java.

Завдання

Визначити рівень споживання ресурсів при роботі з таблицями Azure Cosmos DB

Теоретичні відомості

Azure Cosmos DB - це платформа баз даних, яка є сукупністю логічно взаємопов'язаних баз даних, розподілених у комп'ютерній мережі. Програмні доданки працюють з базами даних Azure Cosmos DB за допомогою інтерфейсів (API), які розроблені також для таких баз даних, як MongoDB, PostgreSQL, Apache Cassandra, Apache Germlin.

Залежно від типу бази даних інтерфейси зберігають дані в різних форматах, так:

- API для MongoDB зберігає дані у структурі документа у форматі BSON (Binary JSON);
- API для Cassandra зберігає дані у схемі, орієнтованій на стовпці;
- API для Gremlin дозволяє зберігати дані у вигляді ребер та вершин;
- API для таблиць зберігає дані у форматі "ключ-значення".

Azure Cosmos DB пропонує доступ до даних зі швидким відгуком під час читання та запису на глобальному рівні. Для забезпечення такого рівня доступності дані копіюються між регіонами.

Копіювання розподілених баз даних для забезпечення високого рівня їх доступності та низької затримки передбачає компроміс між узгодженістю читання та такими параметрами, як доступність, час затримки та пропускна спроможність. Такий підхід пов'язаний з проблемою забезпечення узгодженості даних. Наприклад, для копіювання даних з регіону US до регіону India потрібно 2хв, протягом цього часу, користувач з регіону US буде читати дані, які відрізняються від даних доступних для регіону India, тобто дані будуть не узгоджені. Для забезпечення узгодженості даних Azure Cosmos DB пропонує п'ять чітко визначених рівнів «узгодженості даних» від сильних до слабких:

1. Рівень узгодженості "Строгий" (Strong).
2. Обмежений із запізненням (Bounded staleness).
3. Сеанс (Session).
4. Узгодженість префіксів (Consistent prefix).
5. Рівень узгодженості "Підсумковий" (Eventual).

Таким чином, при роботі з даними, які зберігаються в Azure Cosmos DB важливо визначити, який з визначених типів «узгодженості даних» підтримується для того, щоб правильно оцінювати очікувану «неузгодженість».

Для оцінювання завантаженості CPU при роботі з Azure Cosmos DB використовують міру «одиниця запиту» за секунду (оз/с). 1 оз/с потрібна для читання 1кБ даних з Azure Cosmos DB. На рівень споживання ресурсів при роботі з Azure Cosmos DB впливають наступні фактори:

1) Розмір елемента. У міру збільшення розміру елемента число оз/с необхідне для читання чи запису елемента, також збільшується.

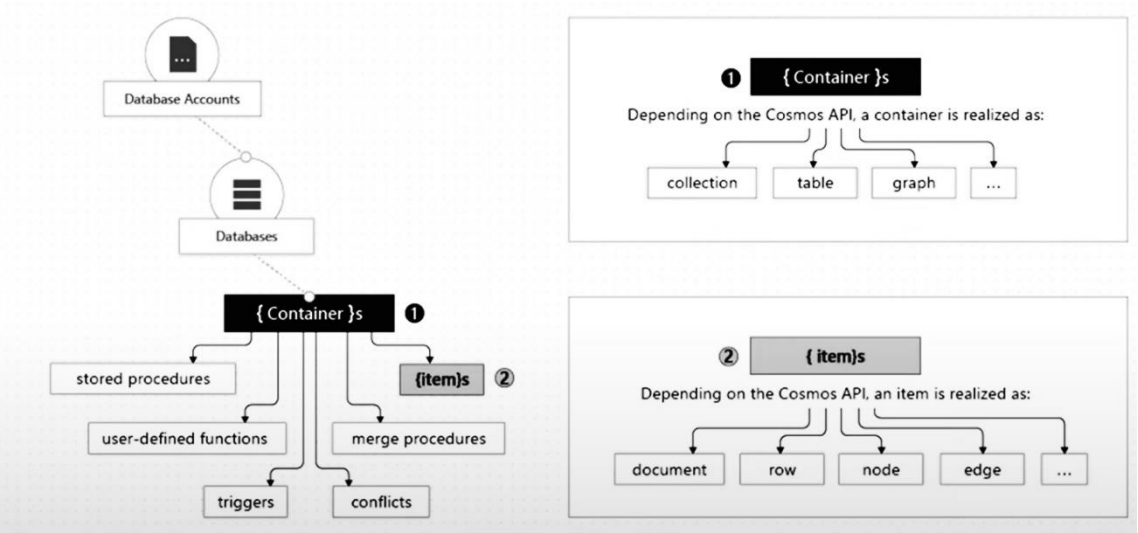
2) Індексація елементів. За замовчуванням кожен елемент в таблиці Azure Cosmos DB автоматично індексується, таким чином зменшення кількості індексованих елементів сприяє зменшенню оз/с.

3) Узгодженість даних. «Строгий» рівень узгодженості потребує приблизно вдвічі більше оз/с при виконанні операцій читання порівняно з іншими рівнями узгодженості.

4) Типи операцій читання та шаблони запитів. Чинники, що впливають на вартість операцій запитів: кількість результатів запиту; кількість предикатів; характер предикатів; кількість функцій, що

визначаються користувачем; розмір вихідних даних; розмір результуючого набору; проекції.

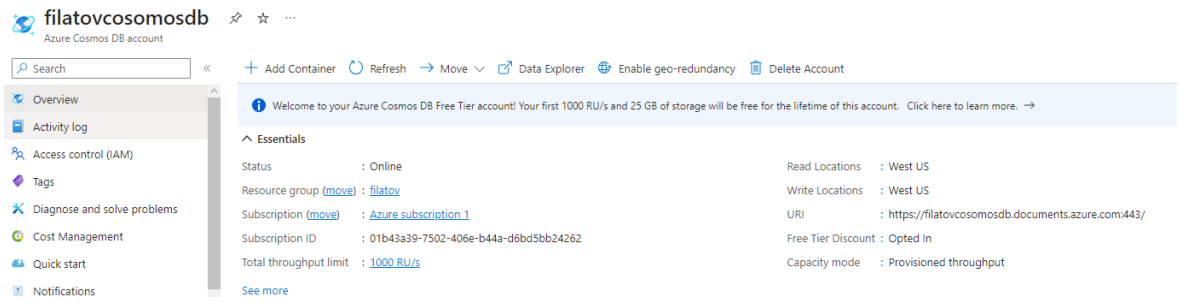
Структура бази даних Azure Cosmos DB, представлена контейнерами, де кожен контейнер складається з даних, процедур, функції та тригерів



Дані одного контейнера розподіляються по різних «логічних секція» для забезпечення масштабування даних. Логічні секції формуються з урахуванням значення ключа секції, який є у кожного елемента у контейнері. Усі елементи у логічній секції мають однакове значення ключа секції. Крім ключа секції кожен елемент в контейнері також має ідентифікатор, який є унікальним в межах логічної секції. Поєднання ключа секції та ідентифікатора елемента створює індекс, що однозначно визначає елемент.

Хід виконання

1. Створіть обліковий запис Azure Cosmos DB.



2. Виділіть один об’єкт, який є частиною вашого програмного доданку з предмету "Системна інженерія програмного забезпечення" і опишіть його властивості за допомогою JSON документу. (1 JSON має включати не менше 3-х секцій, кожна з яких включає більше ніж один об’єкт).

3. За допомогою однієї з клієнтських бібліотек створіть базу даних Cosmos DB та контейнер.

Взаємодія з Cosmos DB починається з екземпляра класу CosmosClient. Для створення екземпляра об’єкта клієнту потрібен обліковий запис та відповідно заповнити змінні середовища в файлі конфігурації. Провести перевірку клієнта можливо якщо передавати облікові дані в ClientSecretCredential або використовувати DefaultAzureCredential:

```
from azure.cosmos import CosmosClient
from azure.identity import ClientSecretCredential, DefaultAzureCredential

import os
url = os.environ['ACCOUNT_URI']
tenant_id = os.environ['TENANT_ID']
client_id = os.environ['CLIENT_ID']
client_secret = os.environ['CLIENT_SECRET']

# Using ClientSecretCredential
aad_credentials = ClientSecretCredential(
    tenant_id=tenant_id,
    client_id=client_id,
    client_secret=client_secret)

# Using DefaultAzureCredential (recommended)
aad_credentials = DefaultAzureCredential()

client = CosmosClient(url, aad_credentials)
```

Після перевірки CosmosClient можна працювати з будь-яким ресурсом в облікового запису. Наведений нижче фрагмент коду створює базу даних API SQL, яка використовується за умовчанням, якщо при виклику create_database не вказано API.

```
from azure.cosmos import CosmosClient, exceptions
import os

URL = os.environ['ACCOUNT_URI']
KEY = os.environ['ACCOUNT_KEY']
client = CosmosClient(URL, credential=KEY)
DATABASE_NAME = 'testDatabase'
try:
    database = client.create_database(DATABASE_NAME)
except exceptions.CosmosResourceExistsError:
    database = client.get_database_client(DATABASE_NAME)
```

Для створення контейнера з параметрами за замовчуванням необхідно скористатись методом

container = database.create_container (id=CONTAINER_NAME, partition_key=PartitionKey (path="/productName")).

4. Завантажте JSON документ; визначте ключ контейнера.
5. Визначте ОЗ/с необхідні для виконання SQL запитів:
   1) читання з фільтром;
   2) читання з 2-хтаблиць;
   3) оновлення значень.

Контрольні запитання.

1. Що означає узгодженість даних?
2. Які типи узгодженості даних підтримує Azure Cosmos DB?
3. Що означає, коли для типу узгодженості даних «Обмежена із запізненням» обрано інтервал часу T=10сек?
4. Який об’єм даних можна прочитати витративши 100 ОЗ/с.
5. Яким чином Cosmos DB забезпечує горизонтальне масштабування?
6. Наведіть приклад, коли невдало визначений ключ секції приводить до зниження продуктивності бази даних?

Лабораторна робота №7.
Робота з Azure Data Factory

Мета роботи: Здобути навики роботи з компонентами Azure Data Factory для організації процесу отримання, перетворення та завантаження даних в хмарному середовище Azure.

Завдання

1. Створіть процес вилучення, перетворення та завантаження (Extract, Transform, Load - ETL) даних в Azure Data Factory.
2. Оформити звіт, який включає: знімок екрану з результатом отриманим на кожному кроці виконання лабораторної роботи; відповіді на контрольні запитання.

Теоретичні відомості

Фабрика даних Azure – це керована хмарна служба, створена для реалізації складних гібридних ETL процесів. До ключових компонентів Фабрики даних Azure належать:

1) конвеєр;

2) дії;

3) набори даних які визначають тип даних для яких створюється ETL процес;

4) пов'язані служби які визначають способи підключення до джерел та сховищ даних.

Зазначені компоненти для реалізації процесу ETL застосовуються наступним чином: компонент конвеєр об’єднує логічну групу дії, які можуть бути пов’язані послідовно або виконуватись паралельно і незалежно одна від одної. На рисунку 1 схематично представлено конвеєр, який об’єднує послідовні дії копіювання даних з одних сховищ (зелений колір) в інше (синій колір).

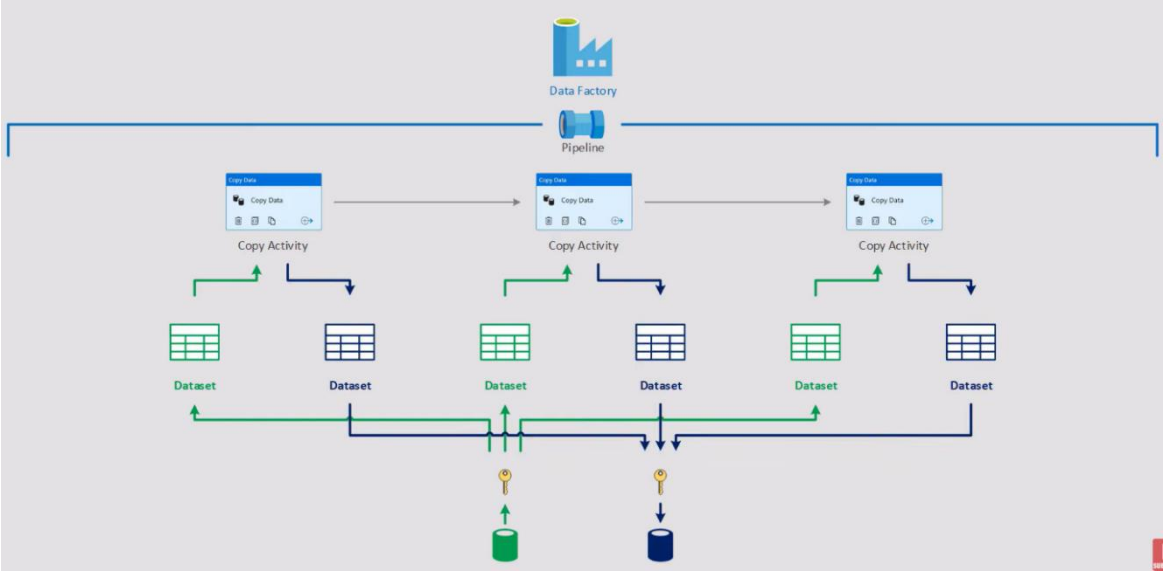


Рис. 1. Конвеєр який об’єднує послідовні дії копіювання даних

Дії для копіювання даних використовують для копіювання даних між локальними та хмарними сховищами, відповідно до процесу:

1) отримання даних з джерела даних;

2) виконання дій серіалізації або десеріалізації, стиснення або розпакування, співставлення стовпців тощо;

3) запис даних до приймача або цільового сховища даних.

Налаштування правил відмовостійкості для дії копіювання визначаються відповідно таблиці 1.

Таблиця 1

Налаштування правил відмовостійкості для дії копіювання

Тип відмово стійкості	**Збій виникає у випадку, коли**	**Управління**
fileMissing	Дані копіюються з файлу, який видаляється іншими доданками	True: інші файли копіюються, окрім того, який видаляється False: копіювання даних зупиняється
fileForbidden	Дані копіюються з файлу, який вимагає вищого рівня дозволів, ніж у налаштованого підключення.	True: необхідно скопіювати решту файлів, пропускаючи ці. False: Ви повинні перервати копіювання при виникненні проблеми з дозволами для папок або файлів.
dataInconsistency	При копіюванні неузгоджених даних між вихідним та цільовим сховищами	True: необхідно скопіювати решту даних, пропускаючи неузгоджені. False: Необхідно перервати копіювання при виявленні неузгоджених даних.
invalidFileName	При копіюванні, якщо імена файлів неприпустимі для цільового сховища.	True: потрібно скопіювати інші файли, пропускаючи файли з неприпустимими іменами. False: необхідно перервати копіювання при виявленні неприпустимих імен файлів.

Потоки даних дозволяють реалізовувати логіку перетворення даних без написання коду. Потоки даних створюються так само, як конвеєри та набори даних. Створений потік даних складається з 3х складових:

- панель управління,
- граф потоку даних,
- панель конфігурації.

Потік зазвичай починається з визначення джерела, при необхідності можна додати будь-яку кількість джерел. Всі додані джерела з'єднуються за допомогою перетворення з'єднання, пошуку або об'єднання.

Тип джерела визначається параметром: «Набір даних» або «Вбудований набір даних».

Об'єкти «Набір даних» - це сутності, які багаторазово використовуються, також в інших потоках даних і діях, таких як копіювання.

Об'єкти «Набір даних» особливо корисні при використанні фіксованої схеми. Вбудовані набори даних - рекомендуються при використанні гнучких схем, одноразових екземплярів або параметризованих джерел. Панель конфігурації визначає схему вихідних даних та правила трансформації даних.

Хід виконання роботи

1. Створіть XML- документ для опису компонент користувацького інтерфейсу програмного доданку з дисципліни «Системна інженерія програмного забезпечення» (приклад в таблиці 1) та завантажте в Azure Blob Storage.

Таблиця 1

Фрагмент XML- документ для опису компонент користувацького інтерфейсу

```
    <form>
info="Main form"/>
        <description info="Booking a room in hotel"/>
        <content info="Content of main form">
            <inputField name="Text_input" lable="Enter name"
    type="text"/>
            <inputField name="Text_input" lable="Enter
    surname" type="text"/>
```

```
        <inputField name="Date_input" lable="Enter start
date" type="date"/>
        <inputField name="Date_input" lable="Enter end
date" type="date"/>
        <inputField name="Text_input" lable="E-mail"
type="text"/>
        <inputField name="Number_input" lable="Enter
phone number" type="number"/>
        <inputField name="Button" lable="Submit"
type="submit"/>
    </content>
</form>
```

2. Створіть таблиці в Azure SQL database для збереження даних XML-документу, окремо в 2-х таблицях: в 1-шу таблицю треба завантажити дані 1-го рівня ієрархії XML-документу; в 2-гу – 2-го рівня ієрархії XML-документу (в таблиці 2 наведено приклад SQL команд для створення таблиць)

Таблиця 2

Команди для створення таблиць в Azure SQL database

```
CREATE TABLE input_fields (
  id INT NOT NULL IDENTITY(1,1),
  name VARCHAR(40) NOT NULL,
  label VARCHAR(40) NOT NULL,
  type VARCHAR(40) NOT NULL,
  PRIMARY KEY (id)
);

CREATE TABLE form_description (
  id INT NOT NULL IDENTITY(1,1),
  info VARCHAR(40) NOT NULL,
  PRIMARY KEY (id)
);
```

3. За допомогою потоку перетворення даних в Azure Data Factory завантажте XML- документ з Azure Blob Storage, виконайте «нормалізацію»

даних XML- документу та збережіть в підготовлених таблицях Azure SQL database. (приклад створено потоку в Azure Data Factory наведено на рис. 1).

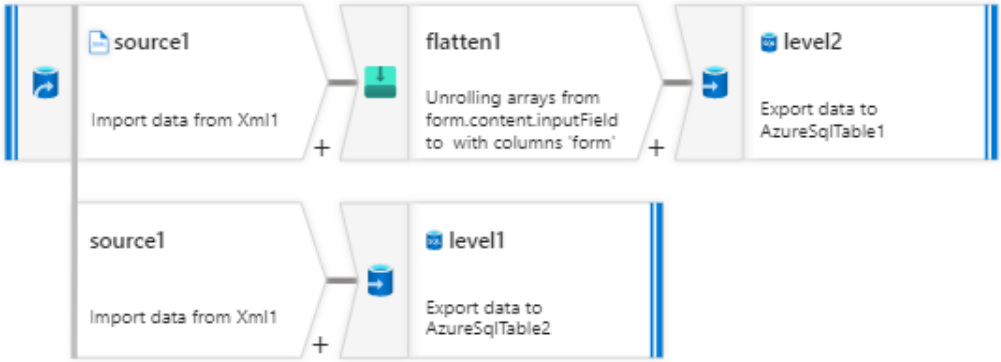


Рис. 1. Схема потоку даних для реалізації процесу ETL

Контрольні запитання

1. Які ключові компоненти включено в Azure Data Factory?
2. Коли варто використовувати дію «Копіювання даних», а коли «Потік перетворення даних»?
3. Який параметр конфігурації дії копіювання треба використовувати для запобігання збою при копіюванні даних, при умові, що збій виник коли «Файли, які копіюються службою, видаляються іншими додатками»?
4. Яку функцію «Потоку перетворення даних» треба використовувати для зміни типу стовбця?
5. Які типи з'єднань підтримуються Azure Data Factory для об'єднання даних з різних джерел?

Література

1. Sreeram PK. Azure Serverless Computing Cookbook: Build and monitor Azure applications hosted on serverless architecture using Azure functions. Packt Publishing Ltd; 2020 Jun 19. https://www.packtpub.com/en-ua/product/azure-serverless-computing-cookbook-9781800206601?type=print
2. Seara DA, Milano F, Dominici D. Microsoft Azure Data Solutions-An Introduction. Microsoft Press; 2021 Jul 14. https://www.oreilly.com/library/view/microsoft-azure-data/9780137252725/
3. L'Esteve RC. The Definitive Guide to Azure Data Engineering: Modern ELT, DevOps, and Analytics on the Azure Cloud Platform. Apress; 2021. https://www.oreilly.com/library/view/the-definitive-guide/9781484271827/
4. Seara DA, Milano F. Exam Ref DP-900 Microsoft Azure Data Fundamentals. Microsoft Press; 2021 Mar 12. https://www.oreilly.com/library/view/exam-ref-dp-900/9780137252145/
5. Хмарні технології [Електронний ресурс]. – Режим доступу: http://j.parus.ua/ua/358.
6. Ю. І. Корольова Переваги та недоліки використання 88 хмарних технологій підприємствами України [Електронний ресурс]. – Режим доступу: http://www.bsfa.edu.ua/files/konf2013/62.p.df.
7. В.Я. Юрчишин Хмарні та грід – технології [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/server/api/core/bitstreams/6db9b134-8c26-4687-9f98-7d3c8e85096e/content

Навчально-методичне видання

GRID-системи та хмарні технології

Методичні вказівки
до виконання лабораторних робіт № 1-7
для підготовки здобувачів другого магістерського рівня вищої освіти
спеціальності 121 «Інженерія програмного забезпечення»

Укладач: О.Л. Соловей

Комп’ютерне верстання

Підписано до друку 22.02.2024 Формат 60 × 84 1/ 16
Ум. друк. арк. 1,16. Обл.-вид. арк. 1,25.
Електронний документ. Вид № 59/ІІІ-17.

Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Повітрофлотський проспект, 31, Київ, Україна, 03037
Свідоцтво про внесення до Державного реєстру суб’єктів
видавничої справи ДК № 808 від 13.02.2002 р.