

Лабораторна робота №6

РОЗГОРТАННЯ ОБЧИСЛЮВАЛЬНИХ СЕРВІСІВ В AZURE ТА УПРАВЛІННЯ НИМИ (AZURE VIRTUAL MACHINES, APP SERVICE, CONTAINER INSTANCE)

Мета: ознайомитися з основними обчислювальними сервісами платформи Microsoft Azure та навчитися розгортати і управляти ними на практиці. Дослідити принципи створення і конфігурації віртуальних машин (Azure Virtual Machines, Azure Virtual Machine Scale Sets), хмарних вебзастосунків (Azure App Service) та контейнерних екземплярів (Azure Container Instances). Сформувані навички вибору відповідного сервісу залежно від сценарію використання, оптимізації параметрів продуктивності та вартості, а також управління станом, масштабуванням і доступом до обчислювальних ресурсів у хмарному середовищі.

Хід роботи

1. Перевірити політику Allowed resource deployment regions та обрати один з регіонів. Створити Virtual Machine Azure Portal. В головному пошуку на Azure Portal знайти Virtual Machine та обрати опцію Create. Далі заповнити поля:

Basics (змінювати тільки вказані поля):

- **Subscription:** обрати підписку Azure for Students;
- **Resource group:** створити нову, назвавши rg-vm-xxx-yyy-zzz, де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, відповідно до цього файлу, що містить відповідність між повною та скороченою назвою регіону, а yyy – ініціали студента, zzz – номер варіанту;
- **Virtual machine name:** vm-xxx-yyy-zzz-001, де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, відповідно до цього

- [файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;
- **Region:** обрати серед доступних, що вказані в політиці регіонів, яка застосовується для навчальних Subscriptions Azure for Students;
 - **Availability options:** Availability set;
 - **Availability set:** Create new
 - **Name:** as-xxx-ууу-zzz-001, де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;
 - **Fault domains:** 2;
 - **Update domains:** 6;
 - Зберегти налаштування Availability set;
 - **Security type:** Trusted launch virtual machines;
 - **Image:** Windows Server 2025 Datacenter – x64 Gen2;
 - **Size:** Standard_B2s – 2 vcpus, 4GiB memory;
 - **Username:** аналогічний username, як і в Azure Cloud Shell, типу ki251_mvuv;
 - **Password:** довільний безпечний пароль не менше 12 символів, що буде містити букви, цифри, спец.символи (у звіті не вказувати);
 - **Public inbound ports:** None;

Disks (змінювати тільки вказані поля):

- **OS disk type:** Standard SSD (locally-redundant storage);
- **Create and attach a new disk:**
 - **Standard SSD (locally-redundant storage);**
 - **Size: 32GiB;**
 - **Delete disk with VM:** check;
 - Зберегти зміни, натиснувши **Ok**;

Networking (змінювати тільки вказані поля):

- **Virtual network:** Edit virtual network
 - Замінити адресний простір на 10.xxx.0.0/16, де, а xxx – номер варіанту групи;

- Редагувати існуючий subnet, обравши в **IPv4 address range** попередньо створений (10.xxx.0.0/16) та зменшити кількість доступних адрес в subnet, **обравши префікс (Size) – /29**;
 - Зберегти зміни;
 - Перейти у вкладку **Review + Create** та після валідації натиснути **Create**.
2. Дозволити доступ за протоколом RDP до новоствореної віртуальної машини. Перейти в налаштування **Virtual Machine > Networking > Network settings** та натиснути **Create port rule > Inbound port rule**.
- Далі вказати наступне:
- **Source:** My IP address;
 - **Source port ranges:** *;
 - **Destination:** Any;
 - **Service:** RDP;
 - **Action:** Allow;
 - **Priority:** 110;
 - **Name:** allow-rdp;
 - **Description:** Allow RDP connection from my current public IP address
 - Зберегти правило.
3. В налаштування віртуальної машини перейти у **Connect > Connect** та натиснути **Check access** та переконатись, що ви маєте доступ за протоколом RDP до віртуальної машини, про що буде сигналізувати надпис «Port 3389 is accessible from source IP(s)». Після цього натиснути **Download RDP file**.
4. Підключитись до віртуальної машини, використовуючи Microsoft Remote Desktop Connection client, відкривши завантажений файл. Після цього ввести логін та пароль, що були задані під час створення віртуальної машини для підключення до неї.
5. Відкрити File Explorer та зафіксувати наявні Devices and drives (Windows (C:), Temporary Storage (D:), DVD Drive (E:)). Важливо зазначити, що на даному етапі відсутній Data drive, що було створено під час створення віртуальної машини. Для того, щоб він відображався у File Explorer та його можна було використовувати, необхідно клікнути правою кнопкою миші по Start Menu Icon та обрати **Disk Management**, а де система запропонує ініціалізувати наявний диск. Натискаємо ОК. Далі знаходимо наш диск (орієнтовно Disk 2, розміром 31.98GB), клікаємо по нерозміченій області, та обираємо **Create new simple volume** і далі

погоджуємось з усіма налаштуваннями за замовчуванням і натискаємо **Finish**. У File Explorer переконаємось, що появився новий Volume, розміром у 31.9GB.

6. Відкрити PowerShell, та виконати команду:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

Після її виконання, відкрити Azure Portal, перейти до створеної раніше віртуальної машини та у вкладці **Overview**, скопіювати значення **Primary NIC public IP** та відкрити вебсторінку, доступну за цією адресою у новій вкладці браузеру, використовуючи протокол HTTP (не HTTPS!). Очікувано, ми отримаємо помилку.

7. Дозволити доступ за протоколом HTTP до віртуальної машини. Перейти в налаштування **Virtual Machine > Networking > Network settings** та натиснути **Create port rule > Inbound port rule**.

Далі вказати наступне:

- **Source:** Any;
- **Source port ranges:** *;
- **Destination:** Any;
- **Service:** HTTP;
- **Action:** Allow;
- **Priority:** 120;
- **Name:** allow-http;
- **Description:** Allow HTTP connections from any location
- Зберегти правило.

8. У новому вікні браузері вставити попередньо скопійовану IP-адресу та переконатись, що ви отримали доступ до стандартної сторінки вебсерверу Internet Information Services (IIS).

9. Відключитись від RDP-підключення та видаляємо усі ресурси, що було попередньо створено.

10. В головному пошуку на Azure Portal знайти App Services та обрати опцію Create > Web App. Далі заповнити поля:

Basics (змінювати тільки вказані поля):

- **Subscription:** обрати підписку Azure for Students;
- **Resource group:** створити нову, назвавши rg-app-linux-php-xxx-yyy-zzz, де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між

повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;

- **Name:** app-linux-php-xxx-ууу-zzz-001 (наприклад, app-linux-php-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту та вимкнути *Try a secure unique default hostname*;
- **Publish:** Code;
- **Runtime stack:** PHP 8.4;
- **Operating system:** Linux;
- **Region:** обрати серед доступних, що вказані в політиці регіонів, яка застосовується для навчальних Subscriptions Azure for Students;
- **App Service Plan** обрати Create new та задати наступні значення:
 - **Name:** asp-linux-php-xxx-ууу-zzz-001 (наприклад, asp-linux-php-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;
 - **Pricing plan:** клікнути на Explore pricing plans та обрати Standard S1;
- **Zone redundancy:** Disabled;
- Перейти у вкладу **Review + Create** та після валідації натиснути **Create**.

Переконатись у тому, що додаток функціонує справно, перейшовши до новогоствореного **App Service > Overview > Browse**.

11. Створити Deployment slot для тестового середовища. У налаштуваннях App Service перейти до **Deployment > Deployment slots**, натиснути Add Slot та вказати наступне:

- **Name:** staging;
- **Clone settings from:** Do not clone settings.

Перейти до новогоствореного Deployment Slot та зафіксувати його Default domain. Порівняйте з Default domain Deployment Slot, який позначений, який *production*. В головному пошуку на Azure Portal знайти App Service plan, знайти попередньо створений Apps Service

plan, перейти у **Overview > Apps(s) / Slots** та зафіксувати наявності там.

12. Налаштувати розгортання оновлень версії додатку. Перейти в **Settings > Configuration (preview)**, поставити галочку у **Platform settings > SCM Basic Auth Publishing Credentials** та зберегти зміни, натиснувши **Appy**. В налаштуваннях App Service перейти до **Deployment > Deployment slots** та перейти до Deployment Slot, який називається *staging*. Далі в налаштуваннях саме staging Deployment slot (важливо не переплутати!) перейти у **Deployment > Deployment Center**. Далі обрати наступне:

- **Source:** External Git;
- **Repository:** <https://github.com/Azure-Samples/php-docs-hello-world>;
- **Branch:** master;
- **Repository type:** Public;
- Зберегти зміни, натиснувши **Save**, потім натиснути Sync.

Зачекати до 5хв, поки додаток буде розгорнуто, то перейти у Overview нашого staging Deployment Slot та натиснути Browse. Переконайтесь, що «Hello World!» додаток було успішно розгорнуто.

13. Виконати переключення Deployment Slots. Перейти в налаштування **App Service > Deployment > Deployment slots**, натиснути **Swap**, переконайтесь, що у **Source** у нас обрано *staging*, а у **Target** – *production*, та натиснути **Start Swap**. Після завершення переключення Deployment Slots, перейти до сторінки **Overview** production Deployment Slot, натиснути **Browse** та переконайтесь, що заміна Deployment Slots пройшла успішно. Аналогічну дію повторити зі staging Deployment Slots. Результат зафіксувати, обов'язково вказавши Default domain.

14. Налаштувати автоматичне горизонтальне масштабування на основі метрик для App Service. В налаштуваннях App Service перейти у **App Service plan > Scale out** та обрати **Scale out method – Rule Based** та клікнути **Configure**. Далі обрати наступне:

- **Choose how to scale your resource:** Custom autoscale;
- **Scale mode:** Scale based on a metric;
- **Instance limits:**
 - **Minimum:** 1;
 - **Maximum:** 2;
 - **Default:** 1;
- **Rules:** Add a rule:

- **Metric namespace:** Standard metrics;
- **Metric name:** CPU Percentage;
- **Operator:** =;
- **Dimension Values:** All values;
- **Operator:** Greater than;
- **Metric threshold to trigger scale action:** 70%;
- **Duration (minutes):** 5;
- **Time grain statistics:** Average;
- **Time Aggregation:** Average;
- **Operation:** Increase count by;
- **instance count:** 1;
- **Cooldown (minutes):** 5.

Натиснути **Add** та зберегти зміни, натиснувши вгорі **Save**.

15. Перевірити масштабування, створивши Load Test. У налаштуваннях App Service перейти у Diagnose and solve problems > Load Test your App > Create Load Test > Create. Залишити значення Subscription, Resource group за замовчуванням, вказати Name – lt-linux-php-xxx-yyy-zzz-001 (наприклад, lt-linux-php-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а yyy – ініціали студента, zzz – номер варіанту; Region – той же, що й було обрано для App Service та App Service Plan. Перейти у вкладку **Review + Create** та після валідації натиснути **Create**. Після створення Load Test, перейти у вкладку **Overview** створеного ресурсу, знайти **Create by adding HTTP requests**, далі ми попадаємо у налаштування URL-based test. Там обираємо **Test plan > Requests > Add request** та в полі **URL** задаємо Default domain URL нашого App Service (обов'язково має починатись з https://) та натиснути **Add**, потім натиснути **Review + Create** та знову **Create**. Далі потрібно зачекати, поки тест буде створено. Тим часом, поки тест запущено, відкрити у іншій вкладці браузеру App Service, обрати створений раніше **App Service > App Service plan > Scale out > Configure > Run history** та переконатись, що через певний час масштабування було здійснено. Додатково переконатись в цьому, перейшовши в App Service Plan, знайти раніше створений App Service Plan на у вкладці Overview переконатись, що Instance count: 2. Далі в

Activity log App Service Plan знайти подію, яка підтверджує, що масштабування відбулось.

16. Зупинити Load Test та видалити попередньо створені ресурси, видаливши усю групу ресурсів.

17. Створити публічний Github пустий репозиторій (створити обліковий запис на Github за його відсутності). В процесі створення вказати наступне:

- **Repository name:** app-win-dotnet-xxx-yyy-zzz-001 (наприклад, app-wind-dotnet-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а yyy – ініціали студента, zzz – номер варіанту;
- **Description:** Lab 6 .NET app deployment test;
- **Choose visibility:** Public;
- **Add README:** On;
- **Add .gitignore:** Dotnet;
- **Add license:** Apache License 2.0;

Та натиснути **Create repository**.

18. Запустити Azure Cloud Shell та в середовищі PowerShell виконати:

```
dotnet -info | Select-String version
```

Зафіксувати версію .NET, яка встановлена в Azure Cloud Shell.

19. В головному пошуку на Azure Portal знайти App Services та обрати опцію Create > Web App. Далі заповнити поля:

Basics (змінювати тільки вказані поля):

- **Subscription:** обрати підписку Azure for Students;
- **Resource group:** створити нову, назвавши rg-app-win-dotnet-xxx-yyy-zzz, де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а yyy – ініціали студента, zzz – номер варіанту;

- **Name:** app-win-dotnet-xxx-yuu-zzz-001 (наприклад, app-linux-php-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту та вимкнути *Try a secure unique default hostname*;
- **Publish:** Code;
- **Runtime stack:** обрати версію .NET з пункту 27;
- **Operating system:** Windows;
- **Region:** обрати серед доступних, що вказані в політиці регіонів, яка застосовується для навчальних Subscriptions Azure for Students;
- **App Service Plan** обрати Create new та задати наступні значення:
 - **Name:** asp-win-dotnet-xxx-yuu-zzz-001 (наприклад, asp-win-dotnet-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;
 - **Pricing plan:** клікнути на Explore pricing plans та обрати Standard S1;
- **Zone redundancy:** Disabled;
- Перейти у вкладку **Review + Create** та після валідації натиснути **Create**.

Переконатись у тому, що додаток функціонує справно, перейшовши до новоствореного **App Service > Overview > Browse**.

20. Відкрити сторінку з репозиторієм на Github, натиснути **Code** та скопіювати посилання у вкладці **HTTPS**. Відкрити Azure Cloud Shell (PowerShell) та виконати (замінити REPO URL на попередньо скопійоване посилання): `git clone <REPO URL>`. Створити новий .NET Core web application, виконавши `dotnet new webapp` та запустити його, виконавши `dotnet run --urls "http://localhost:5005"`. У результаті виконання `dotnet run`, у виводі можна побачити рядок `Переконатись, що додаток працює, відкривши нову сесію Azure Cloud Shell та виконати curl http://localhost:5005` та `curl -I http://localhost:5005`. Зупинити виконання додатку, натиснувши `Ctrl + C`.

21. Створити Personal Access Token (PAT) на Github. Перейти на Github, клікнути вгорі справа на зображення профілю > Settings > Developer Settings > Personal access tokens > Fine-grained token та натиснути Generate new token та задати наступне:

- **Token name:** pat-app-win-dotnet-xxx-yyy-zzz-001 (наприклад, pat-app-win-dotnet-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а yyy – ініціали студента, zzz – номер варіанту;
- **Repository access:** Only select repositories та обрати репозиторій, що було створено раніше;
- **Add permissions:** та обрати мінімально необхідні доступи для виконання `git push` (дослідити самостійно) та натиснути **Generate token**.

Зберегти токен в нотатках, оскільки він буде використовуватись надалі.

22. Закомітити зміни у репозиторій Github. виконавши наступні команди та налаштувати параметри для роботи з репозиторіями:

```
git config --global user.email "you@example.com" # задати пошту, що використовується в обліковому записі Github
```

```
git config --global user.name "Your Name" # Full Name, вказаний у профілі Github
```

```
git add . # додати всі файли в індекс (підготувати до коміту)
```

```
git commit -m "add initial .NET app" # створити коміт з повідомленням
```

```
git push # відправити коміт у віддалений репозиторій (Github) (під час першого коміту треба буде вказати Github username в якості username, а в якості password – попередньо згенерований PAT)
```

Перейти на Github та переконатись, що файли .NET проєкту появились там.

23. Перейти на Azure Portal, знайти попередньо створений App Service та в його налаштуваннях перейти у **Deployment > Deployment Center** та вказати наступне:

- **Source:** Github;
- **Repository:** вказати репозиторій, створений раніше в рамках роботи;
- **Branch:** main;
- **Workflow option:** Add a workflow;
- **Authentication type:** User-assigned identity;
- **Subscription:** обрати доступну підписку Azure for Students.
- **Identity:** залишити значення за замовчуванням.

Натиснути Save та дочекатись виконання синхронізації коду з репозиторію у App Service. Переконатись в тому, що додаток було успішно розгорнуто, перейшовши в **Overview > Browse** нашого App Service.

24. Видалити попередньо створені ресурси в Azure (обов'язково) та репозиторій в Github (опціонально).

25. Встановити [Azure CLI](#) та [Docker Desktop](#) (обрати інструкцію для інсталяції для актуальної версії ОС самостійно) локально.

26. Використовуючи наступні команди Azure CLI у середовищі WSL або в терміналі Bash виконати наступні команди:

```
az login # для здійснення входу в Azure
```

```
az provider register --namespace  
'Microsoft.ContainerRegistry' # реєструємо провайдер для  
роботи з Azure Container Registry
```

```
# задати значення наступним змінним  
LOCATION=<вказати регіон, який використовувався у попередніх  
завданнях>  
RG_NAME=<rg-dotnet-containers-xxx-yyy-zzz, де xxx - скорочена  
назва обраного, серед доступних в політиці, регіонів, яка  
застосовується для навчальних Subscriptions Azure for  
Students, відповідно до цього файлу, що містить відповідність  
між повною та скороченою назвою регіону, а yyy - ініціали  
студента, zzz - номер варіанту>  
ACR_NAME=<acr-dotnet-containers-xxx-yyy-zzz, де xxx - скорочена  
назва обраного, серед доступних в політиці, регіонів, яка  
застосовується для навчальних Subscriptions Azure for  
Students, відповідно до цього файлу, що містить відповідність  
між повною та скороченою назвою регіону, а yyy - ініціали  
студента, zzz - номер варіанту>  
ACR_SKU=Standard
```

```
az group create -l $LOCATION -n $RG_NAME # створюємо групу  
ресурсів
```

```
az acr create -l $LOCATION -n $ACR_NAME -g $RG_NAME --sku  
Standard # створюємо Azure Container Registry
```

```
az acr login -n $ACR_NAME # увійти в Azure Container Registry
```

27. Склонувати репозиторій (<https://github.com/Azure-Samples/dotnetcore-docs-hello-world>), виконати білд Dockerfile, який знаходиться в репозиторії та розмістити його в Azure Container Registry, виконавши наступні команди:

```
docker build -f Dockerfile.linux -t  
$ACR_NAME.azurecr.io/dotnetcore-docs-hello-world-linux . #  
виконуємо білд образу контейнера
```

```
docker images | grep $ACR_NAME # отримуємо інформацію про образ
```

```
docker push $ACR_NAME.azurecr.io/dotnetcore-docs-hello-world-linux:latest
```

Переконайтесь, що образ контейнеру було розміщено в Azure Container Registry. В головному пошуку Azure Portal знайти Azure Container Registry, обрати раніше створений Azure Container Registry, перейти у **Services > Repositories**, де має бути репозиторій з образом контейнеру, яка попередньо було розміщено там за допомогою команд Azure CLI.

Якщо при білді образу контейнеру (виконанні команди `docker build`) ви отримуєте помилку, то можна спробувати замінити оригінальний вміст файлу `Dockerfile.linux` на наступний:

```
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /source
SHELL ["/bin/sh", "-c"]
COPY . ./dotnetcore-docs-hello-world/
WORKDIR /source/dotnetcore-docs-hello-world
RUN dotnet publish -c Release -o /app
```

```
FROM mcr.microsoft.com/dotnet/aspnet:8.0
WORKDIR /app
COPY --from=build /app/ ./
ENV ASPNETCORE_URLS=http://0.0.0.0:80
EXPOSE 80
ENTRYPOINT ["dotnet", "dotnetcoresample.dll"]
```

28. Створити контейнеризований App Service, що буде використовувати попередньо створений образ контейнеру. В головному пошуку на Azure Portal знайти App Services та обрати опцію **Create > Web App**. Далі заповнити поля:

Basics (змінювати тільки вказані поля):

- **Subscription**: обрати підписку Azure for Students;
- **Resource group**: обрати попередньо створену Resource group, в якій був розміщений Azure Container Registry;

- **Name:** app-dotnet-linux-container-xxx-yyy-zzz-001 (наприклад, app-dotnet-linux-frc-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту та вимкнути *Try a secure unique default hostname*;
- **Publish:** Container;
- **Operating system:** Linux;
- **Region:** обрати серед доступних, що вказані в політиці регіонів, яка застосовується для навчальних Subscriptions Azure for Students;
- **App Service Plan** обрати Create new та задати наступні значення:
 - **Name:** asp-dotnet-linux-container-xxx-yyy-zzz-001 (наприклад, asp-dotnet-linux-container-fra-mvv-006-001), де xxx – скорочена назва обраного, серед доступних в політиці, регіонів, яка застосовується для навчальних Subscriptions Azure for Students, [відповідно до цього файлу](#), що містить відповідність між повною та скороченою назвою регіону, а ууу – ініціали студента, zzz – номер варіанту;
 - **Pricing plan:** клікнути на Explore pricing plans та обрати Standard S1;
- **Zone redundancy:** Disabled;

Container (змінювати тільки вказані поля):

- **Image Source:** Azure Container Registry;
- **Name:** main;
- **Azure container registry options:** вказати назву попередньо створеного Azure Container Registry;
- **Authentication:** Managed identity;
- **Identity:** обрати запропоноване створення нової identity;
- **Image:** dotnetcore-docs-hello-world-linux;
- **Tag:** latest;
- **Port:** 80;
- Перейти у вкладку **Review + Create** та після валідації натиснути **Create**.

Після створення Web App, потрібен певний час на його ініціалізацію (до 5хв). Також варто перевірити логи в двох місцях:

- **Deployment > Deployment Center** та на місці існуючого контейнеру (**main**) в правому кутку натиснути на **View logs** (так є змога переглядати логи контейнеру в режимі реального часу);
- **Deployment > Deployment Center** та клікнути у вкладку **Logs**, де буде змога переглянути логи саме App Service. За умови, якщо все пішло добре, то очікувано побачити в логах наступне:

```
qualified image name:
2025-11-04T23:34:24.9945183Z Starting container: b387a20be138_app-dotnet-linux-
nwe-mvv-006-001_main.
2025-11-04T23:34:25.8427311Z Starting watchers and probes.
2025-11-04T23:34:25.8906907Z Starting metrics collection.
2025-11-04T23:34:25.8923708Z Container is running.
2025-11-04T23:34:26.8215519Z Container start method finished after 4485 ms.
2025-11-04T23:34:29.4291995Z Site startup probe succeeded after 2.5587854
seconds.
2025-11-04T23:34:29.5699472Z Site started.
```

Переконатись у тому, що додаток функціонує справно, перейшовши до новогоствореного **App Service > Overview > Browse**.

29. Видалити App Service та App Service Plan в Resource Group. В Resource Group мають залишитись Azure Container Registry та Managed Identity.

30. Для можливості запуску контейнерів, використовуючи Azure Container Instance, на локальній машині виконати команди:

```
az provider register --namespace Microsoft.ContainerInstance #
zareestruvati Container Instance namespace na pivni
Subscription
```

```
az acr update -n acrdotnetcontainergwckmob --admin-enabled
true # uvimknuti privilejovanyj oblikovij zapis dla Azure
container registry (ne rekomendujetsja robiti v production)
```

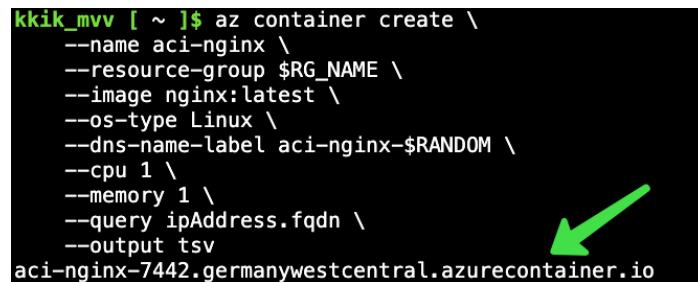
```
ACR_USERNAME=$(az acr credential show --name $ACR_NAME --query
username -o tsv) # zapisujemo u zminnu Username Azure Container
Instance (pered виконанням переконатись, що змінна ACR_NAME не
пуста)
```

```
ACR_PASSWORD=$(az acr credential show --name $ACR_NAME --query
"passwords[0].value" -o tsv) # zapisujemo u zminnu Password
Azure Container Instance (rekomendujetsja zberigati chutlivu
informaciju u servisax, typu Azure Key Vault)
```

31. Запустити контейнер NGINX з публічного репозиторію Dockerhub, використовуючи Azure Container Instance, використовуючи наступні команди:

```
az container create \  
--name aci-nginx \  
--resource-group $RG_NAME \ # переконатись, що RG_NAME, яка була  
оголошена раніше не пуста  
--image nginx:latest \  
--os-type Linux \  
--dns-name-label aci-nginx-$RANDOM \  
--cpu 1 \  
--memory 1 \  
--query ipAddress.fqdn \  
--output tsv
```

Після виконання даної команди має бути виведене fully qualified domain name (FQDN) (див. рис.1). Його необхідно скопіювати, та відкрити у новій вкладці браузеру. Можливо, треба буде зачекати до однієї хвилини, по пройде ініціалізація контейнеру.



```
kkik_mv [ ~ ]$ az container create \  
--name aci-nginx \  
--resource-group $RG_NAME \  
--image nginx:latest \  
--os-type Linux \  
--dns-name-label aci-nginx-$RANDOM \  
--cpu 1 \  
--memory 1 \  
--query ipAddress.fqdn \  
--output tsv  
aci-nginx-7442.germanywestcentral.azurecontainer.io
```

A green arrow points to the output FQDN: `aci-nginx-7442.germanywestcentral.azurecontainer.io`.

Рис. 1. Отриманий FQDN в результаті створення контейнеру Azure Container Instance

Очікуваний результат зображено на рис. 2.

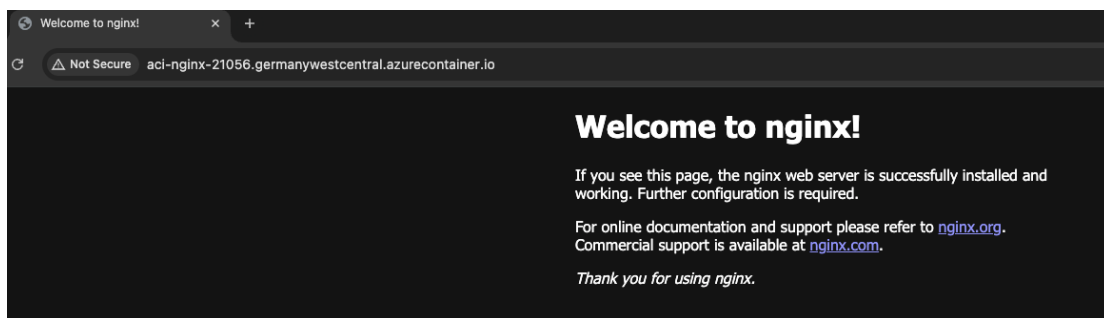


Рис. 2. Отриманий доступ до запущеного контейнеру NGINX в середовищі Azure Container Instance

Перейти в Azure Portal, в Resource Group, та клікнути по Azure Container Instance, який буде містити назву нашого створеного контейнеру. Після переходу вас буде направлено у вкладку **Overview**. Зафіксувати у звіті скріншоти вкладок **Overview**, **Settings > Containers**, **Settings > Properties** та **Overview**, **Settings > Containers**, **Settings > Logs**.

32. Видалити Azure Container Instance, використовуючи Azure Portal (не потрібно видаляти решту ресурсів, тільки Azure Container Instance, що містить назву aci-nginx). Перейти в **Resource Group > Activity log**, та знайти подію з **Operation name** «Delete Container Group» (дана подія може з'явитись через 3-5хв після фактичного видалення ресурсу). Розгорнути деталі та переконатись, що ресурс, над яким було виконано операцію видалення, містить назву контейнеру.
33. Запустити контейнер з приватного Azure Container Registry. Згідно виконання попереднього завдання, ви вже маєте бути автентифіковані в Azure Container Registry. Для надійності можна виконати команди (перед виконанням команд, переконатись, що змінні не пусті):

```
ACR_NAME=$(az acr list --query [].name -o tsv) # записуємо в змінну ACR_NAME ім'я нашого єдиного Azure Container Registry
az acr login -n $ACR_NAME # проходимо автентифікацію в Azure Container Registry
```

```
ACR_REPO_NAME=$(az acr repository list -n $ACR_NAME -o tsv)
# записуємо в змінну ACR_REPO_NAME ім'я нашого єдиного репозиторію в Azure Container Registry
```

```
ACR_IMAGE_TAG=$(az acr repository show-tags --name $ACR_NAME --repository $ACR_REPO_NAME --output tsv) # записуємо в змінну image tag
```

```

az container create \
  --name aci-dotnet-container \
  --resource-group $RG_NAME \
  --image
$ACR_NAME.azurecr.io/$ACR_REPO_NAME:$ACR_IMAGE_TAG \
  --os-type Linux \
  --dns-name-label aci-dotnet-container-$RANDOM \
  --registry-login-server $ACR_NAME.azurecr.io \
  --registry-username $ACR_USERNAME \
  --registry-password $ACR_PASSWORD \
  --cpu 1 \
  --memory 1 \
  --query ipAddress.fqdn \
  --output tsv

```

Після виконання даної команди має бути виведене fully qualified domain name (FQDN) (див. рис.3). Його необхідно скопіювати, та відкрити у новій вкладці браузеру. Можливо, треба буде зачекати до однієї хвилини, по пройде ініціалізація контейнеру.



```

kkik_mv [ ~ ]$ az container create \
  --name aci-dotnet-container \
  --resource-group $RG_NAME \
  --image $ACR_NAME.azurecr.io/$ACR_REPO_NAME:$ACR_IMAGE_TAG \
  --os-type Linux \
  --ports 80 8080 \
  --dns-name-label aci-dotnet-container-$RANDOM \
  --registry-login-server $ACR_NAME.azurecr.io \
  --registry-username $ACR_USERNAME \
  --registry-password $ACR_PASSWORD \
  --cpu 1 \
  --memory 1 \
  --query ipAddress.fqdn \
  --output tsv
aci-dotnet-container-12355.germanywestcentral.azurecontainer.io

```

Рис. 3. Отриманий FQDN в результаті створення контейнеру Azure Container Instance

Примітка: якщо в результаті перевірки роботи додатку він не відкривається, то варто перевірити на якому порту слухає .NET додаток всередині контейнеру, перейшовши у створений Azure Container Instance > Settings > Containers > Logs та знайти рядок, подібний рядку на рис. 4.

The screenshot shows the Azure portal interface for a container instance named 'aci-dotnet-container'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Settings, Containers (selected), Identity, Properties, Locks, Monitoring, Metrics, Alerts, Logs, Automation, CLI / PS, Tasks, and Export template. The main content area displays a table with container information:

Name	Image	State	Previous state	Start time	Restart count
aci-dotnet-container	acrdotnetcontainergwkmo6.azurecr.i...	Running	-	2025-11-11T21:09:34.491Z	0

Below the table, there is a log viewer showing the following messages:

```
warn: Microsoft.AspNetCore.DataProtection.Repositories.FileSystemXmlRepository[60]
Storing keys in a directory '/root/.aspnet/DataProtection-Keys' that may not be persisted outside of the container. Protected data will be un
e when container is destroyed. For more information go to https://aka.ms/aspnet/dataprotectionwarning
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
No XML encryptor configured. Key (e1e0e1af-21ef-4923-9f55-9a6b5c9abeb1) may be persisted to storage in unencrypted form.
info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://[::]:8080
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
Content root path: /app
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
Failed to determine the https port for redirect.
```

Рис. 4. Логи контейнеру, з яких видно на якому порту працює додаток
Якщо додаток працює не на порту 80, то потрібно видалити поточний Azure Container Instance та перестворити його, відкривши доступ до порту, який буде використовуватись нашим додатком всередині контейнеру:

```
az container create \  
  --name aci-dotnet-container \  
  --resource-group $RG_NAME \  
  --image $ACR_NAME.azurecr.io/$ACR_REPO_NAME:$ACR_IMAGE_TAG \  
  --os-type Linux \  
  --ports 80 8080 \  
  --dns-name-label aci-dotnet-container-$RANDOM \  
  --registry-login-server $ACR_NAME.azurecr.io \  
  --registry-username $ACR_USERNAME \  
  --registry-password $ACR_PASSWORD \  
  --cpu 1 \  
  --memory 1 \  
  --query ipAddress.fqdn \  
  --output tsv
```

Після виконаних маніпуляцій перевірити роботу контейнеризованого додатку. З виводу команди скопіювати адресу, вставити в адресний рядок браузеру та перейти.

Очікуваний результат зображено на рис. 5.

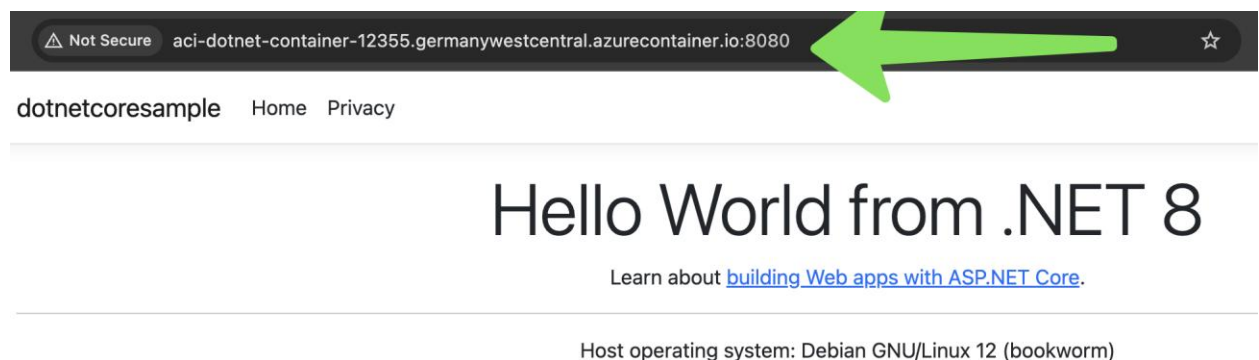


Рис. 5. Запущений працюючий контейнеризований .NET додаток у середовищі Azure Container Instance

Перейти в Azure Portal, в Resource Group, та клікнути по Azure Container Instance, який буде містити назву нашого створеного контейнеру. Після переходу вас буде направлено у вкладку **Overview**. Зафіксувати у звіті скріншоти вкладок **Overview**, **Settings > Containers**, **Settings > Properties** та та **Overview**, **Settings > Containers**, **Settings > Logs**.

34.Видалити усі ресурси, що було створені в рамках даної лабораторної.

Контрольні запитання

1. Які параметри (Availability set, тип безпеки, образ, розмір, регіон) потрібно враховувати під час створення Virtual Machine, і як вони впливають на доступність та продуктивність?
2. Як планування адресного простору (CIDR-нотація, вибір префікса /29 чи /24) впливає на кількість доступних IP-адрес у віртуальній мережі та можливість масштабування інфраструктури?
3. У чому різниця між дозволом RDP через Inbound port rule лише з Мого IP і відкриттям порту для Any? Які ризики пов'язані з обома підходами?
4. Чому після додавання додаткового диска в налаштуваннях VM він не з'являється у File Explorer одразу? Опишіть кроки ініціалізації й монтування в Disk Management.
5. Ви встановили IIS і отримали помилку при відкритті вебсторінки за публічною IP-адресою. Чому це відбулося та як правильно налаштувати правила NSG для доступу за HTTP?
6. Які відмінності між Virtual Machine Scale Set у режимі Flexible orchestration та звичайними VM? Як у цьому сценарії відбувається автоматичне налаштування вебсервера через cloud-init?
7. Як працює Azure Load Balancer у зв'язці з VMSS (frontend IP, backend pool, health probe, load-balancing rule)? Як можна перевірити балансування запитів на практиці?
8. Як налаштовується autoscale для VMSS та App Service? За якими метриками та порогами відбувається масштабування, який cooldown встановлено та як перевірити його дію на практиці?
9. У чому різниця між App Service (Code) та App Service (Container)? Як реалізується безперервне розгортання застосунку через GitHub та механізм Deployment Slots?
10. Опишіть процес роботи з Azure Container Registry та Azure Container Instances: автентифікація, публікація Docker-образу, запуск контейнера та перевірка результату.