Практична робота №6

РОЗГОРТАННЯ ОБЧИСЛЮВАЛЬНИХ СЕРВІСІВ AZURE ТА УПРАВЛІННЯ НИМИ: AZURE VIRTUAL MACHINES, VIRTUAL MACHINE SCALE SETS, APP SERVICE, CONTAINER REGISTRY, CONTAINER INSTANCES, CONTAINER APPS

Мета: ознайомитися з основними обчислювальними сервісами платформи Microsoft Azure та набути практичних навичок їх розгортання й керування ними. Дослідити створення та конфігурацію віртуальних машин (Azure Virtual Machines): вибір розміру з урахуванням квот, розміщення в Availability set та Availability zones, керування керованими дисками та їх шифруванням, зміну розміру та переміщення між Resource groups; горизонтальне масштабування за допомогою Virtual Machine Scale Sets і Azure Load Balancer. Навчитися розгортати вебзастосунки в Azure App Service з використанням deployment slots, власного домену з TLS-сертифікатом, резервного копіювання та автоматичного масштабування, а також працювати з контейнерами за допомогою Azure Container Registry, Azure Container Instances та Azure Container Apps. Сформувати навички вибору обчислювального сервісу залежно від сценарію використання, вимог до доступності та вартості.

Теоретичні відомості

Обчислювальні сервіси Azure

Azure надає обчислювальні сервіси з різним рівнем абстракції. У моделі IaaS (Infrastructure as a Service) користувач отримує віртуальну машину та самостійно керує операційною системою, оновленнями й програмним забезпеченням. У моделі PaaS (Platform as a Service) платформа бере на себе операційну систему та середовище виконання, а користувач розгортає лише код або контейнер. Serverless-сервіси додатково звільняють

від керування кількістю екземплярів: платформа масштабує застосунок автоматично, зокрема до нуля. Порівняння сервісів, що розглядаються в практичній роботі, наведено в табл. 1.

Таблиця 1

Порівняння обчислювальних сервісів Azure

Сервіс	Модель	Одиниця розгортання	Типові сценарії
Azure Virtual Machines	IaaS	Віртуальна машина з обраним образом ОС	Перенесення наявних серверів, повний контроль над ОС, спеціалізоване ПЗ
Virtual Machine Scale Sets	IaaS	Група однотипних віртуальних машин	Горизонтально масштабовані вебсервери та обчислювальні вузли
Azure App Service	PaaS	Вебзастосунок (код або контейнер) в App Service plan	Вебсайти, REST API, бекенди мобільних застосунків
Azure Container Instances	PaaS	Container group (один або кілька контейнерів)	Короткочасні задачі, прості сервіси, пакетна обробка
Azure Container Apps	Serverless	Container app в Container Apps environment	Мікросервіси, API, подієво-керовані застосунки з масштабуванням до нуля

Для оркестрації великих контейнерних застосунків з повним доступом до Kubernetes API призначено Azure Kubernetes Service (AKS).

Azure Virtual Machines: розміри, квоти та образи

Розмір віртуальної машини (VM size) визначає кількість vCPU, обсяг оперативної пам’яті, кількість дисків та пропускну здатність мережі. Розміри об’єднані в серії за призначенням: B – burstable-машини, що накопичують кредити CPU в періоди низького навантаження та витрачають їх у пікові періоди; D – машини загального призначення; E – оптимізовані за пам’яттю; F – оптимізовані за обчисленнями. Ім’я розміру кодує його характеристики: наприклад, у `Standard_B2als_v2` цифра 2 означає кількість vCPU, літера a – процесор AMD, l – зменшений обсяг пам’яті на vCPU, s – підтримку Premium SSD, v2 – версію серії.

Кількість vCPU, яку можна використати, обмежена квотами: загальною регіональною квотою (Total Regional vCPUs) та квотою для кожного сімейства розмірів. Для Subscription Azure for Students квоти невеликі (як правило, 6 vCPU на регіон), а окремі розміри можуть бути недоступні в регіоні (обмеження типу Location) або в окремих зонах доступності (обмеження типу Zone). Крім того, навіть доступний розмір може тимчасово не мати вільних потужностей у регіоні, про що свідчить помилка `AllocationFailed` або `SkuNotAvailable`.

Розмір можна змінити після створення віртуальної машини (resize). Якщо новий розмір доступний на поточному апаратному кластері, зміна виконується з перезавантаженням; інакше віртуальну машину необхідно попередньо зупинити зі звільненням ресурсів (deallocate). Під час зміни розміру віртуальна машина недоступна, а вміст тимчасового диска втрачається. Образ віртуальної машини (image) визначає операційну систему та покоління гіпервізора (Gen1 або Gen2). Тип безпеки **Trusted launch** для образів Gen2 вмикає Secure Boot та віртуальний TPM (vTPM), що захищає від руткітів і буткітів.

Керовані диски та їх шифрування

Віртуальна машина Azure використовує керовані диски (managed disks): диск операційної системи, необов’язкові диски даних (data disks) та, для розмірів з локальним сховищем, тимчасовий диск (temporary disk), вміст якого не зберігається під час зупинки зі звільненням ресурсів чи перенесення на інший хост. Тимчасовий диск мають розміри з літерою `d` у назві (наприклад, `Standard_D2ds_v5`); розміри без неї (зокрема серія B v2, `Standard_B2ls_v2`) тимчасового диска не мають. Тип керованого диска (табл. 2) визначає продуктивність, затримку та вартість. Змінити тип диска можна, якщо віртуальна машина зупинена зі звільненням ресурсів або диск від’єднано (detach). Від’єднаний диск даних зберігається як окремий ресурс

і може бути під’єднаний до цієї або іншої віртуальної машини в тому самому регіоні.

Таблиця 2

Типи керованих дисків Azure

Тип диска	Характеристика	Типове застосування
Standard HDD	Магнітні накопичувачі, найнижча вартість, змінна затримка	Резервні копії, некритичні середовища
Standard SSD	Стабільніша затримка порівняно з HDD	Вебсервери, середовища розробки й тестування
Premium SSD	Висока продуктивність і низька затримка; потрібен розмір VM з підтримкою Premium Storage (літера s в імені)	Промислові навантаження, бази даних
Premium SSD v2	Незалежне налаштування ємності, IOPS та пропускної здатності; не підтримується як диск ОС	Бази даних, аналітика
Ultra Disk	Найвища продуктивність, субмілісекундна затримка; не підтримується як диск ОС	Високонавантажені СКБД, транзакційні системи

Дані керованих дисків завжди шифруються на стороні сховища (server-side encryption, SSE) ключами платформи (platform-managed keys) або ключами клієнта в Azure Key Vault (customer-managed keys). SSE не охоплює тимчасовий диск та кеш дисків на хості. Функція **encryption at host** розширює шифрування на тимчасовий диск і кеш, тобто забезпечує наскрізне шифрування даних віртуальної машини без навантаження на її CPU; перед першим використанням її потрібно зареєструвати для Subscription як feature `Microsoft.Compute/EncryptionAtHost`. Azure Disk Encryption (ADE), що шифрує томи всередині ОС за допомогою BitLocker або DM-Crypt, буде виведено з експлуатації 15 вересня 2028 року; для нових віртуальних машин Microsoft рекомендує encryption at host.

Доступність віртуальних машин: Availability set та Availability zones

Fault domain (домен збою) – група апаратного забезпечення зі спільним живленням і мережевим комутатором, тобто стійка серверних

шаф; Update domain (домен оновлення) – група хостів, які перезавантажуються одночасно під час планового обслуговування платформи. **Availability set** розподіляє віртуальні машини між до трьох fault domains (залежно від регіону) та до 20 update domains (за замовчуванням 5) в межах одного центру обробки даних. **Availability zones** – фізично відокремлені групи центрів обробки даних у регіоні з незалежним живленням, охолодженням і мережею. Розміщення віртуальних машин у різних зонах захищає від відмови цілого центру обробки даних. Для двох і більше віртуальних машин Microsoft гарантує доступність 99,95 % в Availability set та 99,99 % у різних Availability zones. Віртуальна машина може належати або до Availability set, або до зони, але не до обох одночасно.

Переміщення ресурсів

Ресурси можна перемістити в іншу Resource group або Subscription. Переміщення змінює лише метадані: ресурс залишається в тому самому регіоні та продовжує працювати, проте його ідентифікатор ресурсу (resource ID) змінюється, тому скрипти та шаблони, що посилаються на ресурс, потрібно оновити. Під час переміщення вихідна та цільова Resource groups блокуються для операцій зміни. Залежні ресурси (диски, мережеві інтерфейси, Availability set) переміщуються разом із віртуальною машиною. Для перенесення в інший регіон використовується Azure Resource Mover, який створює копію ресурсів у цільовому регіоні.

Virtual Machine Scale Sets та автоматичне масштабування

Virtual Machine Scale Sets (VMSS) дає змогу створювати й керувати групою віртуальних машин з однаковою конфігурацією та змінювати їх кількість вручну, за розкладом або залежно від метрик. Режим оркестрації **Uniform** створює ідентичні екземпляри з єдиної моделі, а режим **Flexible** (рекомендований для нових розгортань) дозволяє поєднувати різні розміри, розподіляти екземпляри між зонами та керувати ними як звичайними

віртуальними машинами. Початкове налаштування екземплярів, наприклад встановлення вебсервера, виконується за допомогою cloud-init, конфігурація якого передається як Custom data.

Автоматичне масштабування (autoscale) Azure Monitor ґрунтується на правилах: правило scale-out збільшує кількість екземплярів, коли метрика (наприклад, Percentage CPU) перевищує поріг протягом заданого часу (duration), а правило scale-in зменшує її, коли метрика нижче порогу. Межі масштабування задаються мінімальною, максимальною та типовою (default) кількістю екземплярів. Період очікування (cool down) після кожної дії масштабування запобігає частим змінам; пороги scale-out та scale-in мають бути достатньо віддалені, щоб уникнути «коливань» (flapping). Історія дій доступна на вкладці **Run history**.

Azure Load Balancer розподіляє вхідний трафік між екземплярами VMSS. Його складові: frontend IP configuration (публічна IP-адреса), backend pool (екземпляри VMSS), health probe (перевірка стану екземплярів) та load-balancing rule (зв’язок порту фронтенду з портом бекенду). Трафік надходить лише на екземпляри, що успішно проходять перевірку health probe.

Azure App Service та App Service plan

Azure App Service – PaaS-сервіс для хостингу вебзастосунків, REST API та бекендів на Windows або Linux, з розгортанням коду (.NET, Java, Node.js, PHP, Python) чи контейнера. Застосунки працюють в App Service plan, що визначає регіон, операційну систему, кількість і розмір екземплярів та цінову категорію (pricing tier). Усі застосунки одного плану використовують одні й ті самі екземпляри та масштабуються разом. Можливості цінових категорій наведено в табл. 3.

Таблиця 3

Можливості цінових категорій App Service plan

Можливість	Free, Shared	Basic	Standard	Premium v3, v4	Isolated v2
Виділені екземпляри	Ні	Так	Так	Так	Так
Максимальна кількість екземплярів	–	3	10	30	100
Власний домен та App Service Managed Certificate	Лише домен (Shared)	Так	Так	Так	Так
Резервне копіювання (Backups)	Ні	Так*	Так	Так	Так
Deployment slots	Ні	Ні	Так	Так	Так
Autoscale (Rules Based)	Ні	Ні	Так	Так	Так
Automatic scaling	Ні	Ні	Ні	Так	Ні

* У категорії Basic резервне копіювання доступне лише для production slot.

Масштабування App Service буває вертикальним (scale up) – зміна цінової категорії плану, та горизонтальним (scale out) – зміна кількості екземплярів. Горизонтальне масштабування виконується вручну (Manual), за правилами Azure Monitor на основі метрик чи розкладу (**Rules Based**) або автоматично за HTTP-трафіком без правил (**Automatic**, лише Premium v2–v4). Одночасно для плану має діяти лише один метод масштабування.

Нові застосунки, створені в Azure Portal, отримують **secure unique default hostname** у форматі `<ім’я>-<хеш>.<регіон>.azurewebsites.net`. Хеш унеможливлює повторне використання цього імені іншою організацією, що захищає від захоплення піддомену (subdomain takeover) через «висячі» DNS-записи.

Deployment slots, розгортання та резервне копіювання App Service

Deployment slot – окремий живий застосунок з власним іменем хоста в тому самому App Service plan (категорії Standard і вище). Новий код розгортається в неробочий (non-production) слот (наприклад, `staging`), перевіряється, після чого слоти міняються місцями (swap). Під час swap платформа спочатку прогріває екземпляри вихідного слоту з

налаштуваннями цільового, і лише потім перемикає маршрутизацію, тому оновлення відбувається без простою; у разі проблем повторний swap повертає попередню версію. Код може надходити з GitHub, Azure Repos, зовнішнього Git-репозиторію (External Git), ZIP-пакета або контейнерного реєстру. Розгортання з External Git та FTP потребує базової автентифікації (**SCM Basic Auth Publishing Credentials**), яка для нових застосунків вимкнена за замовчуванням, оскільки є менш безпечною за автентифікацію Microsoft Entra ID.

Щоб застосунок відповідав на власному домені, у DNS створюються два записи: CNAME (або A для кореневого домену), що вказує на ім'я хоста застосунку, та TXT-запис `asuid.<піддомен>` з ідентифікатором перевірки (Custom Domain Verification ID), який підтверджує володіння доменом. Для захисту домену HTTPS використовується прив'язка TLS/SSL (binding) типу SNI SSL з сертифікатом. **App Service Managed Certificate** – безкоштовний сертифікат, який App Service випускає та автоматично поновлює для домену (категорії Basic і вище); його не можна експортувати, а для піддомену CNAME має вказувати безпосередньо на `*.azurewebsites.net`.

App Service підтримує два типи резервних копій: автоматичні (щогодинні, зберігаються 30 днів, не потребують налаштування) та власні (custom), що створюються на вимогу або за розкладом у контейнері Azure Storage Account і можуть бути завантажені. Резервну копію можна відновити в той самий застосунок, інший застосунок або deployment slot.

Контейнери в Azure: Container Registry, Container Instances, Container Apps

Azure Container Registry (ACR) – приватний реєстр образів контейнерів та OCI-артефактів. Доступні цінові категорії Basic, Standard та Premium, які відрізняються обсягом сховища, пропускною здатністю та можливостями (geo-replication і private endpoints – лише Premium). Під час

створення реєстру обирається **Domain name label scope**: для всіх варіантів, крім Unsecure, до імені сервера входу (login server) додається хеш, наприклад `myregistry-abc123.azurecr.io`. Образи потрапляють до реєстру через `docker push`, через збірку в хмарі за допомогою ACR Tasks (`az acr build`) або імпортом з іншого реєстру (`az acr import`), який не потребує Docker. Запуск ACR Tasks для підписок, що фінансуються безкоштовними кредитами (зокрема Azure for Students), тимчасово призупинено, тому в практичній роботі використовується імпорт. Для доступу сервісів до реєстру рекомендовано використовувати managed identity з роллю **AcrPull** замість облікового запису адміністратора (admin user).

Azure Container Instances (ACI) запускає контейнери без керування віртуальними машинами. Одиницею розгортання є container group – набір контейнерів на одному хості зі спільними мережею, сховищем і життєвим циклом. Для групи задаються ресурси CPU та пам’яті, політика перезапуску (Always, OnFailure, Never) та, за потреби, публічна IP-адреса з DNS-міткою. ACI не має вбудованого автоматичного масштабування: щоб змінити ресурси, групу потрібно створити заново.

Azure Container Apps – serverless-платформа на базі Kubernetes, KEDA, Dapr та Envoy. Застосунки (container apps) розгортаються в Container Apps environment – спільній межі мережі та журналювання. Кожна зміна шаблону застосунку створює нову ревізію (revision). Масштабування задається межами (мінімальна та максимальна кількість реплік) і правилами (HTTP, TCP або власні правила KEDA); за мінімальної кількості 0 застосунок масштабується до нуля і не тарифікується за використання під час простою.

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); **N** – номер варіанту без ведучих нулів (наприклад, 15). Скорочення типів ресурсів в іменах відповідають рекомендаціям Cloud Adoption Framework. Усі ресурси та Resource groups створюються в регіоні, обраному в кроці 1.2. Команди виконуються в Azure Cloud Shell (Bash), якщо не зазначено інше. У командах Azure CLI позначення `xxx-yyy-zzz` задаються змінною `SUFFIX` (наприклад, `SUFFIX=esc-mvv-015`), яку разом з іншими змінними необхідно задати в кроці 1.3 та повторно після кожного перезапуску Azure Cloud Shell.

Через обмежену квоту vCPU підписки Azure for Students ресурси кожної частини роботи необхідно видаляти перед переходом до наступної частини (відповідні кроки наведено в завданнях). Звіт має містити скріншоти та виведення команд, що підтверджують виконання кожного кроку.

Частина 1. Azure Virtual Machines

1. Обрати регіон та розмір віртуальної машини і 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. Для кожного розміру виводяться обсяг

пам’яті (RAM), зони доступності 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], \
  join(',', locationInfo[0].zones || \`[]\`)]"

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 ZONES; do
    Q=$(grep -i "^$FAMILY[[:space:]]" ~/quota-$L.txt) &&
      echo "  $SKU  RAM=${MEM}GB  zones=$ZONES  ($Q)"
  done
done
```

1.2. Обрати регіон, у якому доступні розміри з квотою щонайменше 6 vCPU в сімействі (бажано із зонами доступності 1, 2 і 3 для завдання 8). У цьому регіоні обрати два розміри одного сімейства з 2 vCPU на процесорах Intel, переважно серії B v2: **основний** – з обсягом пам’яті не менше 4 ГіБ (для Windows Server), наприклад `Standard_B2ls_v2`, та **розмір для масштабування** – з більшим обсягом пам’яті, наприклад `Standard_B2s_v2`. Не обирати розміри на процесорах AMD (з літерою `a` після кількості vCPU, наприклад `Standard_B2als_v2`, `Standard_B2as_v2`), розгортання яких у підписці Azure for Students може завершуватися помилкою, а також розміри серій DC, EC та L, для яких квота підписки, як правило, дорівнює нулю. Обрані значення зафіксувати у звіті.

1.3. Задати змінні для обраних регіону та розміру, змінну `SUFFIX` з позначеннями `xxx-yyy-zzz` і перевірити, що розмір підтримує encryption at host (очікуване значення – `True`):

```
LOC=<обраний регіон, наприклад spaincentral>
SIZE=<основний розмір, наприклад Standard_B2ls_v2>
SUFFIX=<xxx-yyy-zzz, наприклад esc-mvv-015>
Q_EAH="[0].capabilities \
  [?name=='EncryptionAtHostSupported'].value"
az vm list-skus --location $LOC --size $SIZE \
  --resource-type virtualMachines --query "$Q_EAH" -o tsv
```

1.4. Зареєструвати для Subscription функцію encryption at host: зареєструвати feature, дочекатися стану **Registered** (кілька хвилин, команду `az feature show` повторювати до зміни стану) та оновити реєстрацію resource provider:

```
az feature register --namespace Microsoft.Compute \
  --name EncryptionAtHost
az feature show --namespace Microsoft.Compute \
  --name EncryptionAtHost --query properties.state -o tsv
az provider register --namespace Microsoft.Compute
```

2. Створити віртуальну машину Windows Server за допомогою Azure Portal.

2.1. У головному пошуку Azure Portal знайти **Virtual machines** та обрати **Create** > **Virtual machine**. На вкладці **Basics** змінити лише вказані поля:

– **Subscription** – Azure for Students;
– **Resource group** – **Create new**, `rg-vm-xxx-yyy-zzz`;
– **Virtual machine name** – `vm-xxx-yyy-zzz` (ім’я віртуальної машини Windows не може перевищувати 15 символів);
– **Region** – регіон з кроку 1.2;
– **Availability options** – **Availability set**; **Availability set** – **Create new**: **Name** – `avail-xxx-yyy-zzz`, **Fault domains** – `2`, **Update domains** – `5`; натиснути **OK**. Варіант **Availability set** доступний не в усіх регіонах; якщо його немає в списку, обрати **No infrastructure redundancy required** та зазначити це у звіті;

– **Security type** – **Trusted launch virtual machines**;
– **Image** – **Windows Server 2025 Datacenter - x64 Gen2**;
– **Size** – основний розмір з кроку 1.2 (за відсутності розміру в списку натиснути **See all sizes**);
– **Username** – ім'я користувача, аналогічне імені в Azure Cloud Shell (наприклад, `ki251_mvv`); **Password** – надійний пароль довжиною не менше 12 символів, що містить великі й малі літери, цифри та спеціальні символи (у звіті не вказувати);
– **Public inbound ports** – **None**.

2.2. Перейти на вкладку **Disks** та змінити лише вказані поля: **OS disk type** – **Standard SSD (locally-redundant storage)**; встановити прапорець **Encryption at host**. Диск даних на цьому етапі не додавати.

2.3. Перейти на вкладку **Networking**. Для **Virtual network** обрати **Create new** (або **Edit virtual network**) та задати: **Name** – `vnet-vm-xxx-yyy-zzz`; адресний простір – `10.N.0.0/16`; для наявної підмережі задати **Name** – `snet-vm` та діапазон `10.N.0.0/29` (8 адрес, з яких 5 резервує Azure). Зберегти зміни. Встановити прапорець **Delete public IP and NIC when VM is deleted**.

2.4. Перейти на вкладку **Review + create** та після успішної валідації натиснути **Create**. Після завершення розгортання натиснути **Go to resource**. Додати до звіту скріншот сторінки **Overview** віртуальної машини, на якому видно Availability set (якщо його обрано), розмір та стан **Encryption at host** (розділ **Properties**).

3. Дозволити доступ до віртуальної машини за протоколом RDP та підключитися до неї.

3.1. У налаштуваннях віртуальної машини перейти в **Networking** > **Network settings** та обрати **Create port rule** > **Inbound port rule**. Задати: **Source** – **My IP address**; **Source port ranges** – *;

Destination – **Any**; **Service** – **RDP**; **Action** – **Allow**; **Priority** – `110`; **Name** – `allow-rdp`; **Description** – `Allow RDP from my current public IP address`. Натиснути **Add**.

3.2. Перейти в **Connect** > **Connect**, натиснути **Check access** та переконатися у відображенні повідомлення про доступність порту 3389 з IP-адреси джерела. Натиснути **Download RDP file**.

3.3. Відкрити завантажений файл у клієнті Remote Desktop Connection (у macOS – Windows App) та підключитися, ввівши ім'я користувача та пароль, задані під час створення віртуальної машини.

4. Розгорнути вебсервер IIS та дозволити доступ до нього за протоколом HTTP.

4.1. У віртуальній машині відкрити PowerShell від імені адміністратора та виконати:

```
Install-WindowsFeature -Name Web-Server `
  -IncludeManagementTools
```

4.2. В Azure Portal на сторінці **Overview** віртуальної машини скопіювати значення **Public IP address** та відкрити в новій вкладці браузера адресу `http://<Public IP address>` (саме HTTP, а не HTTPS). Зафіксувати, що сторінка не відкривається, та пояснити причину у звіті.

4.3. Створити правило в **Networking** > **Network settings** > **Create port rule** > **Inbound port rule**: **Source** – **Any**; **Source port ranges** – *; **Destination** – **Any**; **Service** – **HTTP**; **Action** – **Allow**; **Priority** – `120`; **Name** – `allow-http`. Натиснути **Add**, оновити сторінку в браузері та переконатися, що відкривається стандартна сторінка Internet Information Services (IIS). Додати до звіту скріншот.

5. Додати диск даних, змінити розмір віртуальної машини та тип диска.

5.1. У налаштуваннях віртуальної машини перейти в **Settings** > **Disks** та обрати **Create and attach a new disk**. Задати **Disk name**

– `disk-data-xxx-yyy-zzz`, **Storage type** – **Standard HDD**, **Size (GiB)** – `32`, встановити прапорець **Delete with VM** (за наявності). Натиснути **Apply**.

5.2. У віртуальній машині відкрити File Explorer та зафіксувати наявні диски: Windows (C:), DVD Drive та, якщо обраний розмір має тимчасовий диск, Temporary Storage (D:). Новий диск даних поки не відображається, оскільки не має розділів і файлової системи.

5.3. Натиснути правою кнопкою миші на кнопку **Start** та обрати **Disk Management**. У вікні **Initialize Disk** залишити тип розділів **GPT** та натиснути **OK**. Натиснути правою кнопкою миші на нерозподілену область нового диска (приблизно 32 ГБ), обрати **New Simple Volume**, погодитися з усіма значеннями за замовчуванням та натиснути **Finish**. Переконатися, що в File Explorer з’явився новий том, та створити на ньому текстовий файл `test.txt` з довільним вмістом. Відключитися від RDP-сеансу.

5.4. В Azure Portal на сторінці **Overview** віртуальної машини натиснути **Stop** та підтвердити дію. Дочекатися стану **Stopped (deallocated)**.

5.5. Перейти в **Availability + scale** > **Size**, обрати розмір для масштабування з кроку 1.2 (того самого сімейства, з більшим обсягом пам’яті) та натиснути **Resize**. Зафіксувати у звіті старий і новий розміри та їх характеристики.

5.6. Перейти в **Settings** > **Disks**, для диска `disk-data-xxx-yyy-zzz` натиснути **Detach** (піктограма від’єднання) та **Apply**. Диск залишається в Resource group як окремий ресурс.

5.7. У головному пошуку знайти **Disks**, відкрити `disk-data-xxx-yyy-zzz` та перейти в **Settings** > **Size + performance**. Змінити тип сховища на **Standard SSD** та натиснути **Save**.

5.8. Повернутися до віртуальної машини, перейти в **Settings** > **Disks**, обрати **Attach existing disks**, вибрати `disk-data-xxx-yyy-zzz`, переконатися, що для нього вказано **Standard SSD**, та натиснути **Apply**.

5.9. Запустити віртуальну машину (**Start**), повторно підключитися через RDP та переконатися, що том і файл `test.txt` збереглися. Пояснити у звіті, чи має обраний розмір тимчасовий диск (за назвою розміру та значенням **Temp storage** у списку **Size**) і чому вміст тимчасового диска після зупинки зі звільненням ресурсів не гарантується.

6. Перемістити віртуальну машину в іншу Resource group.

6.1. Створити Resource group `rg-vm-moved-xxx-yyy-zzz` в тому самому регіоні.

6.2. Відкрити Resource group `rg-vm-xxx-yyy-zzz` та виділити всі ресурси, встановивши прапорець у заголовку списку (віртуальна машина переміщується разом із залежними ресурсами: дисками, мережевим інтерфейсом, публічною IP-адресою, NSG, віртуальною мережею та Availability set, якщо його створено).

6.3. Натиснути **Move** > **Move to another resource group**. Вказати як цільову Resource group `rg-vm-moved-xxx-yyy-zzz` та натиснути **Next**. Дочекатися завершення перевірки (validation) та натиснути **Next**.

6.4. Підтвердити розуміння того, що інструменти та скрипти, які посилаються на ресурси, необхідно оновити, і натиснути **Move**. Дочекатися сповіщення про завершення переміщення.

6.5. Переконатися, що віртуальна машина продовжує працювати (вебсторінка IIS відкривається за публічною IP-адресою) та що її **Resource ID** на сторінці **Properties** містить нову Resource group. Видалити порожню Resource group `rg-vm-xxx-yyy-zzz`.

7. Створити віртуальну машину Linux за допомогою Azure CLI.

7.1. В Azure Cloud Shell (Bash) задати змінні (значення `LOC` та `SIZE` – з кроку 1.3) та створити віртуальну машину Ubuntu з увімкненим encryption at host, без публічної IP-адреси:

```
RG=rg-vm-moved-$SUFFIX
az vm create --resource-group $RG --location $LOC \
  --name vm-cli-$SUFFIX \
  --image Ubuntu2404 --size $SIZE \
  --admin-username azureuser --generate-ssh-keys \
  --encryption-at-host true \
  --public-ip-address "" --nsg "" \
  --os-disk-delete-option Delete --nic-delete-option Delete
```

7.2. Переглянути розмір, Availability set та стан encryption at host створеної віртуальної машини, а також порівняти їх із віртуальною машиною Windows:

```
Q_VM="[].{name:name, size:hardwareProfile.vmSize, \
  avset:availabilitySet.id, \
  eah:securityProfile.encryptionAtHost}"
az vm list --resource-group $RG --query "$Q_VM" -o table
```

7.3. Зупинити віртуальну машину зі звільненням ресурсів та переконатися в стані `VM deallocated`:

```
az vm deallocate --resource-group $RG \
  --name vm-cli-$SUFFIX
az vm get-instance-view --resource-group $RG \
  --name vm-cli-$SUFFIX \
  --query "instanceView.statuses[1].displayStatus" -o tsv
```

7.4. Видалити Resource group з усіма ресурсами частини 1 та переконатися, що її видалено:

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

Частина 2. Virtual Machine Scale Sets

8. Створити Virtual Machine Scale Set з балансувальником навантаження та вебсервером NGINX за допомогою Azure CLI.

   8.1. Інтерфейс створення Virtual Machine Scale Set в Azure Portal може не відображати частину регіонів, дозволених політикою підписки, а регіон з кроку 1.3 може не мати вільної ємності чи квоти для набору віртуальних машин. Тому ресурси цього завдання створюються в Azure Cloud Shell (Bash) скриптом, що перебирає всі дозволені регіони та розміри до першого успішного розгортання. Задати змінні (`SUFFIX` – позначення `xxx-yyy-zzz`, `N` – номер варіанту без ведучих нулів) та створити файл cloud-init, який встановлює NGINX і записує в стандартну сторінку ім’я хоста екземпляра:

```
SUFFIX=<xxx-yyy-zzz, наприклад esc-mvv-015>
N=<номер варіанту, наприклад 15>
VMSS=vmss-$SUFFIX
LB=lb-vmss-$SUFFIX
NSG=nsg-vmss-$SUFFIX
VNET=vnet-vmss-$SUFFIX
PIP=pip-lb-vmss-$SUFFIX
cat > cloud-init.yaml <<'EOF'
#cloud-config
package_update: true
packages:
  - nginx
runcmd:
  - echo "Hello from $(hostname)" > /var/www/html/index.html
  - systemctl enable --now nginx
EOF
```

   8.2. Оголосити функцію `deploy`, яка в регіоні `$L` створює Resource group `$RG`, віртуальну мережу з підмережею `snet-vmss`, NSG з правилом `allow-http`, публічний Standard Load Balancer (backend pool `bepool`, health probe `probe-http` за протоколом HTTP, правило `rule-http` для порту 80) та Virtual Machine Scale Set у режимі оркестрації Flexible з двома екземплярами розміру

`$SIZE`. Публічна IP-адреса кожного екземпляра забезпечує вихідний доступ до Інтернету для встановлення пакетів:

```
deploy() {
  az group create -n $RG -l $L -o none &&
  az network vnet create -g $RG -n $VNET -l $L \
    --address-prefix 10.$((N+100)).0.0/20 \
    --subnet-name snet-vmss \
    --subnet-prefix 10.$((N+100)).0.0/24 -o none &&
  az network nsg create -g $RG -n $NSG -l $L -o none &&
  az network nsg rule create -g $RG --nsg-name $NSG \
    -n allow-http --priority 1010 --access Allow \
    --protocol Tcp --direction Inbound \
    --destination-port-ranges 80 -o none &&
  az network vnet subnet update -g $RG \
    --vnet-name $VNET -n snet-vmss \
    --network-security-group $NSG -o none &&
  az network lb create -g $RG -n $LB -l $L \
    --sku Standard --public-ip-address $PIP \
    --frontend-ip-name fe-http \
    --backend-pool-name bepool -o none &&
  az network lb probe create -g $RG --lb-name $LB \
    -n probe-http --protocol Http --port 80 \
    --path / -o none &&
  az network lb rule create -g $RG --lb-name $LB \
    -n rule-http --protocol Tcp \
    --frontend-port 80 --backend-port 80 \
    --frontend-ip-name fe-http \
    --backend-pool-name bepool --probe-name probe-http \
    --disable-outbound-snat true -o none &&
  az vmss create -g $RG -n $VMSS -l $L \
    --orchestration-mode Flexible $ZONES \
    --instance-count 2 --image Ubuntu2404 \
    --vm-sku $SIZE --storage-sku StandardSSD_LRS \
    --admin-username azureuser --generate-ssh-keys \
    --vnet-name $VNET --subnet snet-vmss \
    --lb $LB --backend-pool-name bepool \
    --public-ip-per-vm --custom-data cloud-init.yaml \
    -o none
}
```

8.3. Виконати скрипт, який для кожного регіону з політики Allowed resource deployment regions формує перелік кандидатів так само, як скрипт з кроку 1.1: розміри з 2 vCPU без обмеження Location, крім розмірів на процесорах AMD та Arm (літери `a` чи `p` після кількості vCPU) і серій A, DC, EC та L, для яких вільна квота сімейства та регіональна квота становлять щонайменше 4 vCPU (два екземпляри). Для кожного кандидата скрипт визначає

наявність зон доступності та запускає `deploy`: якщо зон щонайменше дві, спочатку з розміщенням екземплярів у зонах 1 і 2, а в разі помилки – без зон. Кожна спроба виконується в окремій Resource group з номером спроби (`rg-vmss-xxx-yyy-zzz-1`, `-2` тощо), оскільки повторне створення ресурсів у щойно видаленій Resource group з тим самим ім'ям може завершуватися помилкою `ResourceNotFound`; Resource group невдалої спроби видаляється у фоновому режимі (`--no-wait`), і скрипт одразу переходить до наступного варіанта. Ім'я Resource group успішної спроби зберігається в змінній `RG` і використовується в наступних кроках. Перелік розмірів і квоти не враховують поточної нестачі ємності в регіоні чи зоні (помилка `SkuNotAvailable` з повідомленням про Capacity Restrictions), яка виявляється лише під час розгортання, тому перебір кандидатів необхідний. Зафіксувати у звіті невдалі

спроби з повідомленнями про помилки та регіон і розмір, у яких розгортання виконано успішно:

```
REGIONS=$(az policy assignment show \
  --name sys.regionrestriction \
  --query "parameters.listOfAllowedLocations.value" -o tsv)
Q_USAGE="[].[name.value, currentValue, limit]"
Q_C="[?length(restrictions[?type=='Location'])==\`0\` \
  && capabilities[?name=='vCPUs' && value=='2']] \
  .[name, family, length(locationInfo[0].zones || \`[]\`)]"
EXCL='^Standard_(A|DC|EC|L)|^Standard_[A-Z]+2[a-z]*[ap]'
OK=""; I=0
for L in $REGIONS; do
  az vm list-usage -l $L --query "$Q_USAGE" \
    -o tsv > ~/quota-$L.txt
  TOT=$(awk '$1=="cores"{print $3-$2}' ~/quota-$L.txt)
  [ "${TOT:-0}" -ge 4 ] || continue
  az vm list-skus -l $L --resource-type virtualMachines \
    --query "$Q_C" -o tsv | grep -Ev "$EXCL" | sort \
    > ~/cand-$L.txt
  while read SIZE FAMILY NZ; do
    FREE=$(awk -v f="$FAMILY" \
      'tolower($1)==tolower(f){print $3-$2}' ~/quota-$L.txt)
    [ "${FREE:-0}" -ge 4 ] || continue
    ZN=""; [ "$NZ" -ge 2 ] && ZN="--zones 1 2"
    for ZONES in "$ZN" ""; do
      I=$((I+1)); RG=rg-vmss-$SUFFIX-$I
      echo "=== $RG $L $SIZE ${ZONES:-no zones} free=$FREE"
      if deploy < /dev/null; then OK=1; break 3; fi
      az group delete -n $RG --yes --no-wait < /dev/null
      [ -z "$ZN" ] && break
    done
  done < ~/cand-$L.txt
done
[ -n "$OK" ] && echo "OK: $RG $L $SIZE $ZONES" \
  || echo "Не вдалося розгорнути в жодному регіоні"
```

8.4. Перевірити створені ресурси: регіон, режим оркестрації, кількість і розмір екземплярів, їх зони та стан, а також правило й health probe балансувальника:

```
Q_SS="{loc:location, mode:orchestrationMode, \
  sku:sku.name, capacity:sku.capacity, \
  zones:join(',', zones || \`[]\`)}"
az vmss show -g $RG -n $VMSS --query "$Q_SS" -o table
Q_INST="[].{name:name, zone:zones[0], \
  state:provisioningState}"
az vmss list-instances -g $RG -n $VMSS \
  --query "$Q_INST" -o table
Q_LB="{rules:loadBalancingRules[].name, \
  probes:probes[].name, \
  pool:backendAddressPools[0].name}"
az network lb show -g $RG -n $LB --query "$Q_LB" -o json
```

8.5. В Azure Portal відкрити `vmss-xxx-yyy-zzz` та додати до звіту скріншот сторінки **Overview** (Location, Orchestration mode, Availability zones, кількість екземплярів).

9. Перевірити роботу балансувальника навантаження.

9.1. В Azure Portal відкрити `lb-vmss-xxx-yyy-zzz`, перейти в **Monitoring** > **Insights** та дочекатися, доки health probe позначатиме обидва екземпляри як справні (healthy); встановлення NGINX через cloud-init триває кілька хвилин після створення екземплярів.

9.2. Отримати в Azure Cloud Shell публічну IP-адресу балансувальника (її також видно в **Settings** > **Frontend IP configuration**) та виконати 10 HTTP-запитів:

```
LBIP=$(az network public-ip show -g $RG \
  -n $PIP --query ipAddress -o tsv)
for i in $(seq 1 10); do curl -s http://$LBIP; done
```

9.3. Переконатися, що відповіді надходять від різних екземплярів (різні значення hostname). Додати виведення до звіту та пояснити, на якому рівні моделі OSI працює Azure Load Balancer.

10. Налаштувати автоматичне масштабування Virtual Machine Scale Set.
    10.1. Перейти в **Instances**, обрати один з екземплярів та натиснути **Delete**, щоб у Virtual Machine Scale Set залишився один екземпляр.
    10.2. Перейти в **Availability + scale** > **Scaling**, обрати **Custom autoscale** та **Scale based on a metric**. В **Instance limits** задати: **Minimum** – `1`, **Maximum** – `2`, **Default** – `1` (максимальне значення обмежено квотою vCPU).
    10.3. Натиснути **Add a rule** та створити правило scale-out: **Metric source** – **Current resource**; **Metric namespace** – **Virtual Machine Host**; **Metric name** – **Percentage CPU**; **Operator** – **Greater than**; **Metric threshold to trigger scale action** – `70`; **Duration (minutes)** – `5`; **Time grain statistic** – **Average**; **Time aggregation** – **Average**; **Operation** – **Increase count by**; **Cool down (minutes)** – `5`; **Instance count** – `1`. Натиснути **Add**.
    10.4. Натиснути **Add a rule** та створити правило scale-in: **Operator** – **Less than**; **Metric threshold to trigger scale action** – `30`; **Duration (minutes)** – `5`; **Operation** – **Decrease count by**; **Instance count** – `1`. Натиснути **Add**, після чого натиснути **Save** у верхній частині сторінки.
    10.5. Згенерувати навантаження на CPU. Перейти в **Instances**, відкрити екземпляр, що залишився, перейти в **Operations** > **Run command** > **RunShellScript**, вставити скрипт, який запускає по одному нескінченному циклу на кожне vCPU, та натиснути **Run**:

```
for i in $(seq $(nproc)); do
  nohup sh -c 'while :; do :; done' >/dev/null 2>&1 &
done
```

    10.6. Через 5–10 хвилин оновити сторінку **Instances** та переконатися, що кількість екземплярів збільшилася до двох. На сторінці

Scaling перейти на вкладку **Run history** та додати до звіту скріншот з подією масштабування. Якщо масштабування не відбулося, переконатися на сторінці **Metrics** екземпляра, що значення Percentage CPU перевищує 70 %.

10.7. Завершити процеси навантаження, виконавши в **Run command** > **RunShellScript** того самого екземпляра:

```
pkill -f 'while :; do :; done'
```

10.8. Дочекатися, доки правило scale-in зменшить кількість екземплярів до одного, та додати до звіту скріншот **Run history**.

10.9. Видалити Resource group успішної спроби (змінна `RG`) та переконатися, що Resource groups невдалих спроб також видалено (порожнє виведення другої команди) і в Subscription не залишилося віртуальних машин (порожнє виведення третьої команди):

```
az group delete -n $RG --yes
Q_RG="[?starts_with(name, 'rg-vmss-')].name"
az group list --query "$Q_RG" -o tsv
az vm list --query "[].name" -o tsv
```

Частина 3. Azure App Service

11. Створити вебзастосунок Azure App Service.

11.1. У головному пошуку Azure Portal знайти **App Services** та обрати **Create** > **Web App**. На вкладці **Basics** задати:

– **Resource group** – **Create new**, `rg-app-xxx-yyy-zzz`;

– **Name** – `app-php-xxx-yyy-zzz`; переконатися, що під полем відображається ім’я хоста у форматі secure unique default hostname, та зафіксувати його у звіті;

– **Publish** – **Code**; **Runtime stack** – найновіша доступна версія PHP 8 (наприклад, **PHP 8.4**); **Operating System** – **Linux**; **Region** – регіон з кроку 1.2;

– **Linux Plan** – **Create new**, `asp-php-xxx-yyy-zzz`;

– **Pricing plan** – натиснути **Explore pricing plans** та обрати **Premium V3 P0V3**. Це найменша категорія, що підтримує всі можливості, використані в цій частині (deployment slots, autoscale, власний домен із сертифікатом, резервне копіювання), а в більшості регіонів вона дешевша за **Standard S1** за більшого обсягу пам'яті; порівняти ціни обох категорій у вікні **Explore pricing plans** та зафіксувати їх у звіті. Якщо P0V3 недоступна в регіоні або виникає помилка квоти, обрати **Standard S1**;
– **Zone redundancy** – **Disabled**.

11.2. На вкладці **Monitor + secure** для **Enable Application Insights** обрати **No**. Натиснути **Review + create** та **Create**. Після розгортання відкрити застосунок, на сторінці **Overview** натиснути **Browse** та переконатися, що відображається стандартна сторінка App Service.

12. Створити deployment slot та розгорнути в ньому застосунок з Git-репозиторію.

12.1. У налаштуваннях застосунку перейти в **Deployment** > **Deployment slots** та натиснути **Add**. Задати **Name** – `staging`, **Clone settings from** – **Do not clone settings**, натиснути **Add**, після чого – **Close**.

12.2. Відкрити слот `staging` та порівняти його **Default domain** з доменом production slot. У головному пошуку знайти **App Service plans**, відкрити `asp-php-xxx-yyy-zzz` та переглянути **Apps** – слот відображається як окремий застосунок того самого плану. Додати до звіту скріншот.

12.3. У слоті `staging` (важливо не плутати зі production slot) перейти в **Settings** > **Configuration** > **General settings**, у розділі **Platform settings** встановити для **SCM Basic Auth Publishing Credentials** значення **On** та зберегти зміни (**Apply** або **Save**).

12.4. У слоті `staging` перейти в **Deployment** > **Deployment Center** > **Settings** та задати: **Source** – **External Git**; **Repository** – `https://github.com/Azure-Samples/php-docs-hello-world`; **Branch** – `master`; **Repository Type** – **Public**. Натиснути **Save**.

12.5. Дочекатися завершення розгортання (вкладка **Logs** у Deployment Center), відкрити **Overview** слоту `staging` та натиснути **Browse**. Переконатися, що відображається сторінка **Hello World!**, а production slot досі показує стандартну сторінку.

13. Виконати swap deployment slots.

13.1. Перейти в **Deployment** > **Deployment slots** застосунку та натиснути **Swap**.

13.2. Переконатися, що **Source** – `staging`, **Target** – production slot (`app-php-xxx-yyy-zzz`), переглянути вкладки **Source slot changes** та **Target slot changes** і натиснути **Start Swap**. Після завершення натиснути **Close**.

13.3. Відкрити production slot та слот `staging` у браузері та переконатися, що вони помінялися вмістом. Зафіксувати у звіті Default domain обох слотів та пояснити, чому swap не спричиняє простою застосунку.

14. Під’єднати власний домен та захистити його TLS-сертифікатом.

14.1. Використати наявний домен у зоні pp.ua (далі – `<домен>`). Якщо для домену ще немає публічної зони Azure DNS, створити її: у головному пошуку знайти **DNS zones**, натиснути **Create**, задати **Resource group** – **Create new**, `rg-dns-xxx-yyy-zzz`, **Name** – `<домен>`, натиснути **Review + create** та **Create**; на сторінці **Overview** зони скопіювати чотири **Name servers** та вказати їх як NS-записи домену в панелі реєстратора (наприклад, NIC.UA).

Поширення змін NS може тривати від кількох хвилин до кількох годин.

14.2. У production slot перейти в **Settings** > **Custom domains** та натиснути **Add custom domain**. Задати: **Domain provider** – **All other domain services**; **TLS/SSL certificate** – **App Service Managed Certificate**; **TLS/SSL type** – **SNI SSL**; **Domain** – `app-yyy-zzz.<домен>`. Не натискати **Validate**.

14.3. У розділі **Domain validation** скопіювати значення записів CNAME (Default domain застосунку) та TXT (Custom Domain Verification ID) і створити їх у DNS-зоні – в Azure Portal (**DNS zones** > `<домен>` > **Recordsets** > **Add**) або в Azure Cloud Shell:

```
ZONE=<домен>
RG_DNS=rg-dns-$SUFFIX
az network dns record-set cname set-record \
  --resource-group $RG_DNS --zone-name $ZONE \
  --record-set-name app-yyy-zzz \
  --cname <Default domain застосунку>
az network dns record-set txt add-record \
  --resource-group $RG_DNS --zone-name $ZONE \
  --record-set-name asuid.app-yyy-zzz \
  --value <Custom Domain Verification ID>
```

14.4. Повернутися до вікна **Add custom domain**, натиснути **Validate**, переконатися, що обидва записи позначено як перевірені, та натиснути **Add**. Дочекатися, доки стан домену зміниться на **Secured** (випуск сертифіката може тривати кілька хвилин).

14.5. Відкрити `https://app-yyy-zzz.<домен>` у браузері та переглянути сертифікат (видавця, термін дії та ім’я суб’єкта). Переглянути сертифікат у **Settings** > **Certificates** > **Managed certificates**. Додати до звіту скріншоти.

15. Налаштувати резервне копіювання застосунку.

15.1. У production slot перейти в **Backups** та переглянути перелік автоматичних резервних копій (перша автоматична копія з’являється протягом години після створення застосунку).

15.2. Натиснути **Configure custom backups**. У полі **Storage account** обрати **Create new** та задати ім'я `stbkpxxxyyyzzz` (лише малі літери та цифри; якщо ім'я зайняте, додати суфікс `rrr` з 3 випадкових символів a-z0-9), у полі **Container** обрати **Create new** та задати `backups`. Натиснути **Configure**.

15.3. Натиснути **Backup Now** та дочекатися стану **Succeeded** для нової резервної копії.

15.4. Відновити створену резервну копію в слот `staging`: для резервної копії натиснути **Restore**, у розділі **Choose a destination** обрати **Deployment slot** – `staging` та натиснути **Restore**. Після завершення переконатися, що слот `staging` знову показує сторінку **Hello World!**.

16. Налаштувати автоматичне масштабування App Service plan та перевірити його за допомогою Azure Load Testing.

16.1. У production slot перейти в **Settings** > **Scale out (App Service plan)**, обрати метод **Rules Based** та перейти до налаштування правил (**Configure** або **Manage rules based settings**). Обрати **Custom autoscale** та **Scale based on a metric**, в **Instance limits** задати **Minimum** – `1`, **Maximum** – `2`, **Default** – `1`.

16.2. Натиснути **Add a rule** та створити правило scale-out: **Metric namespace** – **Standard metrics**; **Metric name** – **CPU Percentage**; **Operator** – **Greater than**; **Metric threshold to trigger scale action** – `70`; **Duration (minutes)** – `5`; **Time grain statistic** – **Average**; **Time aggregation** – **Average**; **Operation** – **Increase count by**; **Instance count** – `1`; **Cool down (minutes)** – `5`. Натиснути **Add**.

16.3. Додати правило scale-in: **Operator** – **Less than**; **Metric threshold to trigger scale action** – `30`; **Duration (minutes)** – `5`; **Operation** – **Decrease count by**; **Instance count** – `1`. Натиснути **Add** та **Save**.

16.4. У production slot перейти в **Diagnose and solve problems**, знайти **Load Test your App** та обрати **Create Load Test**. Залишити запропоновані Subscription та Resource group, задати **Name** – `lt-xxx-yyy-zzz`, **Region** – регіон застосунку. Натиснути **Review + create** та **Create**.

16.5. Після створення ресурсу Azure Load Testing на сторінці **Overview** обрати **Create by adding HTTP requests** (URL-based test). На вкладці **Test plan** натиснути **Add request** та вказати в полі **URL** адресу production slot, що починається з `https://` (Default domain або власний домен). Натиснути **Add**, потім **Review + create** та **Create**. Тест запускається автоматично після створення.

16.6. Під час виконання тесту спостерігати метрики **Virtual users**, **Response time** та **Requests/sec**. В іншій вкладці відкрити **Scale out (App Service plan)** > налаштування правил > **Run history** та переконатися, що відбулося масштабування. Додатково перевірити кількість екземплярів на сторінці **Overview** App Service plan та знайти подію масштабування в **Activity log** плану. Додати до звіту скріншоти.

16.7. Зупинити тест (**Stop**) та підтвердити дію.

16.8. Видалити DNS-записи `app-yyy-zzz` (CNAME) та `asuid.app-yyy-zzz` (TXT), щоб не залишати «висячих» записів, що вказують на видалений застосунок. Видалити Resource group `rg-app-xxx-yyy-zzz` (та Resource group ресурсу Azure Load Testing, якщо вона відрізняється):

```
az network dns record-set cname delete \
  --resource-group $RG_DNS --zone-name $ZONE \
  --name app-yyy-zzz --yes
az network dns record-set txt delete \
  --resource-group $RG_DNS --zone-name $ZONE \
  --name asuid.app-yyy-zzz --yes
az group delete --name rg-app-$SUFFIX --yes --no-wait
```

Частина 4. Контейнери в Azure

17. Створити Azure Container Registry та імпортувати до нього образ.

17.1. Зареєструвати resource providers, необхідні для цієї частини (портал реєструє їх автоматично, проте для команд Azure CLI це потрібно зробити явно):

```
for NS in Microsoft.ContainerRegistry \
          Microsoft.ContainerInstance Microsoft.App \
          Microsoft.OperationalInsights \
          Microsoft.ManagedIdentity Microsoft.Web; do
  az provider register --namespace $NS
done
```

17.2. У головному пошуку Azure Portal знайти **Container registries** та натиснути **Create**. На вкладці **Basics** задати: **Resource group** – **Create new**, `rg-ctr-xxx-yyy-zzz`; **Registry name** – `crxxxyyyzzz` (лише літери та цифри, без дефісів); **Location** – регіон з кроку 1.2; **Pricing plan** – **Basic**; **Domain name label scope** – **Tenant Reuse**; **Role assignment permissions mode** – **RBAC Registry Permissions**. Інші параметри залишити за замовчуванням, натиснути **Review + create** та **Create**.

17.3. Відкрити реєстр та зафіксувати значення **Login server** на сторінці **Overview** (воно містить хеш, доданий через Domain name label scope).

17.4. Імпортувати до реєстру публічний образ з Microsoft Container Registry та переглянути репозиторії й теги:

```
ACR=cr${SUFFIX//-/}
az acr import --name $ACR \
  --source \
  mcr.microsoft.com/azuredocs/aci-helloworld:latest \
  --image aci-helloworld:v1
az acr repository list --name $ACR --output table
az acr repository show-tags --name $ACR \
  --repository aci-helloworld --output table
```

17.5. В Azure Portal перейти в **Services** > **Repositories** реєстру та переконатися в наявності репозиторію `aci-helloworld` з тегом `v1`. Додати до звіту скріншот.

18. Розгорнути контейнер з Azure Container Registry в Azure App Service.
    18.1. У головному пошуку знайти **App Services** та обрати **Create** > **Web App**. На вкладці **Basics** задати: **Resource group** – `rg-ctr-xxx-yyy-zzz`; **Name** – `app-ctr-xxx-yyy-zzz`; **Publish** – **Container**; **Operating System** – **Linux**; **Region** – регіон з кроку 1.2; **Linux Plan** – **Create new**, `asp-ctr-xxx-yyy-zzz`; **Pricing plan** – **Basic B1** (найменша виділена категорія; deployment slots та autoscale у цьому завданні не потрібні).
    18.2. На вкладці **Container** задати: **Image Source** – **Azure Container Registry**; **Registry** – `crxxxyyyzzz`; **Authentication** – **Managed identity**; **Identity** – створити нову user-assigned managed identity з іменем `id-ctr-xxx-yyy-zzz`; **Image** – `aci-helloworld`; **Tag** – `v1`; **Port** – `80`. Натиснути **Review + create** та **Create**. Портал автоматично призначить managed identity роль AcrPull на реєстр.
    18.3. Після створення застосунку переглянути журнали в **Deployment** > **Deployment Center** > **Logs**, дочекатися повідомлення про успішний запуск контейнера, відкрити **Overview** та натиснути **Browse**. Переконатися, що відображається сторінка застосунку **Welcome to Azure Container Instances!**.
    18.4. Відкрити реєстр `crxxxyyyzzz` > **Access control (IAM)** > **Role assignments** та переконатися, що identity `id-ctr-xxx-yyy-zzz` має роль **AcrPull**. Пояснити у звіті, чому використання managed identity безпечніше, ніж облікового запису адміністратора реєстру.
    18.5. Видалити застосунок `app-ctr-xxx-yyy-zzz` та App Service plan `asp-ctr-xxx-yyy-zzz`.

19. Розгорнути контейнер в Azure Container Instances.

19.1. У головному пошуку знайти **Container instances** та натиснути **Create**. На вкладці **Basics** задати: **Resource group** – `rg-ctr-xxx-yyy-zzz`; **Container name** – `ci-xxx-yyy-zzz`; **Region** – регіон з кроку 1.2; **Availability zones** – **None**; **Image source** – **Quickstart images**; **Image** – `mcr.microsoft.com/azuredocs/aci-helloworld:latest (Linux)`; **Size** – натиснути **Change size** та задати `1` vCPU і `1` ГіБ пам'яті.

19.2. На вкладці **Networking** задати: **Networking type** – **Public**; **DNS name label** – `ci-yyy-zzz`; **DNS name label scope reuse** – **Tenant**; **Ports** – `80` (TCP).

19.3. На вкладці **Monitoring** зняти прапорець **Enable container instance logs** (журнали контейнера залишаються доступними в порталі, а Log Analytics workspace не створюється). Інші параметри залишити за замовчуванням, натиснути **Review + create** та **Create**.

19.4. Після розгортання відкрити ресурс та на сторінці **Overview** переконатися, що **Status** – **Running**. Скопіювати **FQDN**, відкрити його в браузері та кілька разів оновити сторінку.

19.5. Перейти в **Settings** > **Containers** та переглянути вкладки **Properties** (образ, ресурси, політика перезапуску) та **Logs** (HTTP-запити GET від браузера). Переглянути виділені ресурси в Azure CLI:

```
Q_CI="{state:instanceView.state, fqdn:ipAddress.fqdn, \
  restart:restartPolicy, \
  cpu:containers[0].resources.requests.cpu, \
  memoryGB:containers[0].resources.requests.memoryInGB}"
az container show --resource-group rg-ctr-$SUFFIX \
  --name ci-$SUFFIX --query "$Q_CI" -o table
```

19.6. На сторінці **Overview** натиснути **Stop**, переконатися, що FQDN перестав відповідати, та знову натиснути **Start**. Пояснити у

звіті, як у Azure Container Instances змінюються ресурси CPU та пам’яті і чи підтримується автоматичне масштабування.

19.7. Видалити container group `ci-xxx-yyy-zzz` в Azure Portal. Перейти в **Activity log** Resource group та знайти подію **Delete Container Group** (може з’явитися через 3–5 хвилин після видалення).

20. Розгорнути застосунок в Azure Container Apps та налаштувати масштабування.

20.1. У головному пошуку знайти **Container Apps** та обрати **Create** > **Container App**. На вкладці **Basics** задати: **Resource group** – `rg-ctr-xxx-yyy-zzz`; **Container app name** – `ca-xxx-yyy-zzz`; **Deployment source** – **Container image**; **Region** – регіон з кроку 1.2; **Container Apps environment** – **Create new**, **Environment name** – `cae-xxx-yyy-zzz`, **Zone redundancy** – **Disabled**.

20.2. На вкладці **Container** встановити прапорець **Use quickstart image** та залишити **Simple hello world container**. На вкладці **Ingress** переконатися, що **Ingress** увімкнено, **Ingress traffic** – **Accepting traffic from anywhere**, **Target port** – `80`. Натиснути **Review + create** та **Create**.

20.3. Після розгортання натиснути **Go to resource**, перейти за посиланням **Application Url** та переконатися, що відкривається сторінка застосунку. Додати до звіту скріншот.

20.4. Перейти в **Application** > **Scale** та обрати **Edit and deploy**. На вкладці **Scale** задати межі **Min replicas** – `0`, **Max replicas** – `3`. Натиснути **Add**, задати **Rule name** – `http-rule`, **Type** – **HTTP Scaling**, **Concurrent requests** – `10`, натиснути **Add**, а потім **Create**. Буде створено нову ревізію застосунку.

20.5. Згенерувати одночасні запити до застосунку з Azure Cloud Shell:

```
URL=https://<Application Url без https://>
for i in $(seq 1 300); do
  curl -s -o /dev/null $URL &
done; wait
```

20.6. Перейти в **Application** > **Revisions and replicas**, обрати активну ревізію та переглянути кількість реплік. Переглянути метрику **Replica Count** в **Monitoring** > **Metrics**. Через 5–10 хвилин без запитів переконатися, що застосунок масштабувався до нуля реплік. Додати до звіту скріншоти та порівняти масштабування Azure Container Apps з autoscale App Service та Virtual Machine Scale Sets.

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

21.1. Видалити Resource group `rg-ctr-xxx-yyy-zzz` разом з усіма ресурсами (реєстр, managed identity, Container Apps environment, Log Analytics workspace):

```
az group delete --name rg-ctr-$SUFFIX --yes --no-wait
```

21.2. Видалити Resource group `NetworkWatcherRG`, якщо її було автоматично створено під час виконання практичної роботи. Публічну DNS-зону `<домен>` у Resource group `rg-dns-xxx-yyy-zzz` можна залишити для подальшого використання.

21.3. Переконатися, що в Subscription не залишилося ресурсів, створених у практичній роботі (Resource groups `rg-vm-*`, `rg-vmss-*`, `rg-app-*`, `rg-ctr-*` відсутні), та додати до звіту виведення команди:

```
az group list --output table
```

Контрольні запитання

1. Які параметри визначають розмір віртуальної машини Azure? Як квоти vCPU та обмеження розмірів у регіоні впливають на вибір розміру і що означають помилки QuotaExceeded та AllocationFailed?
2. Чим відрізняються Availability set та Availability zones? Що таке fault domain та update domain і яку доступність гарантує кожен із варіантів розміщення?
3. Які типи керованих дисків існують в Azure і за яких умов можна змінити тип диска або розмір віртуальної машини? Чому вміст тимчасового диска не гарантується?
4. Чим відрізняються server-side encryption, encryption at host та Azure Disk Encryption? Чому для нових віртуальних машин рекомендовано encryption at host?
5. Що відбувається з ресурсом під час переміщення в іншу Resource group? Які ресурси потрібно переміщувати разом із віртуальною машиною та як перенести її в інший регіон?
6. Чим відрізняються режими оркестрації Uniform та Flexible у Virtual Machine Scale Sets? Як працює автоматичне масштабування за метрикою (поріг, duration, cool down, межі кількості екземплярів)?
7. Як цінова категорія App Service plan впливає на доступні можливості? Чим відрізняються методи масштабування Manual, Rules Based та Automatic?
8. Що таке deployment slots і як виконується swap? Чому swap дає змогу оновлювати застосунок без простою?
9. Які DNS-записи потрібні для під’єднання власного домену до App Service і для чого використовується TXT-запис asuid? Що таке App Service Managed Certificate і secure unique default hostname?

10. Порівняйте Azure Container Registry, Azure Container Instances та Azure Container Apps. Як образ потрапляє до реєстру без Docker і чому для доступу до реєстру рекомендовано managed identity?