

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра системного аналізу та управління

О.С. Мінеєв, Ю.О. Шевченко

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
галузі знань 12 (F) Інформаційні технології

Дніпро
НТУ «ДП»
2026

Аналіз програмного забезпечення [Електронний ресурс] : методичні рекомендації до виконання практичних робіт для здобувачів ступеня бакалавра галузі знань 12 (F) Інформаційні технології / уклад.: О.С. Мінєєв, Ю.О. Шевченко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2026. – 28 с.

Укладачі:

О.С. Мінєєв, канд. техн. наук, доц.;

Ю.О. Шевченко.

Затверджено кафедрою системного аналізу та управління (протокол № 1 від 05.01.2026).

Методичні рекомендації містять опис методики виконання практичних робіт з дисципліни «Аналіз програмного забезпечення» здобувачами галузі знань 12 (F) Інформаційні технології.

Орієнтовано на активізацію навчальної діяльності бакалаврів та закріплення практичних знань з даної дисципліни.

Відповідальний за випуск завідувач кафедри системного аналізу та управління Т.А. Желдак, канд. техн. наук, доц.

ЗМІСТ

ВСТУП.....	4
ПРАКТИЧНІ ЗАНЯТТЯ № 1.....	6
ПРАКТИЧНІ ЗАНЯТТЯ № 2.....	9
ПРАКТИЧНІ ЗАНЯТТЯ № 3.....	13
ПРАКТИЧНІ ЗАНЯТТЯ № 4.....	15
ПРАКТИЧНІ ЗАНЯТТЯ № 5.....	21
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	27
ДОДАТОК А Приклад оформлення титульної сторінки.....	28

ВСТУП

Методичні рекомендації призначені для здобувачів галузі знань 12 (F) Інформаційні технології, що вивчають дисципліну «Аналіз програмного забезпечення».

Методичні рекомендації включають низку частково взаємопов'язаних робіт, виконання яких дозволить здобувачам отримати широкий спектр знань і навичок у сфері сучасного світу ІТ індустрії. А саме:

- ~ розуміння системи функціонування ІТ компанії;
- ~ опанування базових знань аналітика якості програмного забезпечення;
- ~ розуміння як офіційних так і не офіційних правил поведінки в корпоративному житті;
- ~ робота з хмарними середовищами на приклади AWS;
- ~ розуміння систематизації мов і фреймворків;
- ~ розуміння CI\CD та контейнеризації;
- ~ розуміння Agile ідеології;
- ~ практичні навички в роботі по Scrum, Kanban.

Перед виконанням практичної роботи здобувачі повинні:

- ~ ознайомитися з методичними вказівками;
- ~ повторити лекційний матеріал, пов'язаний з практичною роботою;
- ~ підготувати відповіді на питання, які наведені у методичних вказівках наприкінці кожної практичної роботи.

Виконавши завдання практичної роботи, здобувач повинен оформити звіт за результатами даної практичної роботи, захистити його та здати викладачеві до Moodle під назвою *SurnameGroup_apzNetwork.pdf*, де Surname та Group – прізвище та група здобувача, наприклад: *Shevchenko_124-22-1_apz1.pdf* в встановлені терміни.

Для завантаження до Moodle і на ГітХаб звіт повинен бути у форматі ***.pdf**.

Не використовуйте програмне забезпечення, що використовується для об'єднання файлів у один архівний файл.

Звіт розміщується на сайті дистанційної освіти у відповідному розділі із завданням і оцінюється виходячи з 100 балів в ході захисту.

Загальна оцінка по курсу визначається як середнє значення балів, отриманих за 5 практичних робіт.

Оцінювання практичних робіт здійснюється за наступними критеріями:

Практична робота			
Бали	Критерій		
80	Виконання практичної частини	80	Здобувач правильно, в повному об'ємі виконує практичну роботу згідно із завданням в методичних інструкціях.
		-5	-5 балів за кожну помилку або невиконаний пункт в завданні до практичної роботи.
20	Оформлення звіту	20	Здобувач правильно оформляє звіт з практичної роботи згідно з вимогами та чинними стандартами та надає відповіді на всі питання при захисті.
		10	Здобувач неправильно оформляє звіт з практичної роботи або надає відповіді не на всі питання під час захисту.
		0	Здобувач не надав звіт до практичної роботи.
-1	Термін виконання	-1	При несвоєчасному виконанні завдання -1 бал за кожен прострочений день.
Максимальна кількість балів: 100			

ПРАКТИЧНА РОБОТА №1

Тема: Підписання персонального документа за допомогою кваліфікованого електронного підпису (КЕП).

Мета: закріпити теоретичні знання і розвинути практичні навички у підписанні документації із використанням КЕП.

За результатами виконання практичної роботи здобувачі мають набути вміння підписувати особисту документацію з використанням кваліфікованого електронного підпису за допомогою різних сервісів і додатків.

Короткі теоретичні відомості

Кваліфікований електронний підпис (КЕП) має таку ж юридичну силу, як і власноручний підпис, та має презумпцію його відповідності власноручному підпису.

Наразі в Україні використовується термін «кваліфікований електронний підпис» (КЕП), який прийшов на зміну терміну «електронний цифровий підпис» (ЕЦП) після набрання чинності Закону України «Про електронні довірчі послуги» у 2018 році.

Кваліфікований електронний підпис базується на кваліфікованому сертифікаті відкритого ключа та створюється за допомогою спеціалізованих програмних або апаратно-програмних засобів, що забезпечують найвищий рівень безпеки.

У стандартах Європейського інституту стандартів телекомунікацій (ETSI), визначено три формати розширеного підпису та один формат контейнера підпису:

- удосконалений (advanced) електронний підпис XML (XAdES), заснований на підписах XML;
- удосконалений (advanced) електронний підпис PDF (PAdES), заснований на підписах PDF;
- удосконалений (advanced) електронний підпис CMS (CAdES), заснований на Cryptographic Message Syntax (CMS);
- Асоційований контейнер підпису (ASiC) на основі формату ZIP та який підтримує формати підпису XAdES і CAdES.

Ці формати електронного підпису мають визнаватися як українськими так і європейськими органами державного сектору.

Завдання.

Створити документ у форматі *.pdf. В цьому документі написати кілька речень з фактами про себе. Наприклад: «Я Микола і я маю kota Димчика. А ще я обожнюю баскетбол». Підписати цей документ за допомогою кваліфікованого цифрового підпису (КЕП), використовуючи любі безкоштовні сервіси. Результат виконання завантажити викладачеві для перевірки.

Приклад накладання кваліфікованого електронного підпису.

Крок 1. Переходимо на сайт [«Акредитований центр сертифікації ключів АЦСК»](#) за посиланням. Далі обираємо пункт «Підписати документ» і завантажуюмо файл для підпису.



Рисунок 1.1 - Акредитований центр сертифікації ключів АЦСК

Крок 2. Виконуємо авторизацію через «Приват24» і підтверджуємо підписання документа. Після виконання цих дій ви можемо завантажити собі файл з підписом (рис. 2).

Вивантаження

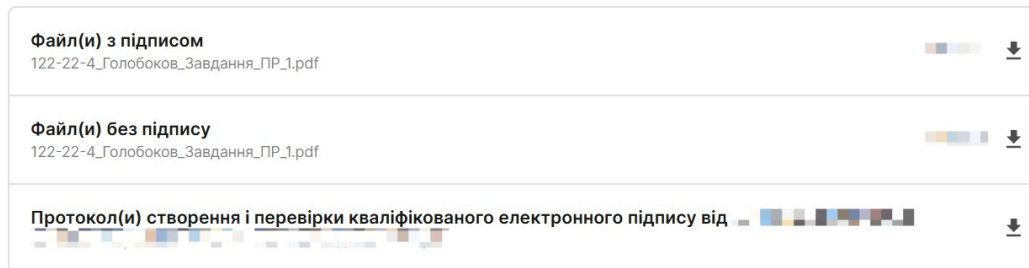


Рисунок 1.2 - Результат підписання КЕП

Крок 3. Для перевірки файлу на наявність КЕП скористуємося сайтом <https://czo.gov.ua/verify>.

Перевірити підпис

👍 Файл успішно перевірено. Усі дані цілі
Ви можете зберегти підписаний файл.

Рисунок 1.3 - Результат перевірки накладання КЕП

Додатково. Зверніть увагу, що на сайті [«Акредитований центр сертифікації ключів АЦСК»](#) ви можете також *отримати сертифікат* і *відкликати сертифікат* КЕП (рис. 1.1). Також **перевірити** підписаний вами документ на наявність КЕП перед відправкою викладачеві.

Інші сервіси. Для накладання і перевірки КЕП ви можете скористуватися іншими безкоштовними сервісами - Дія. Підпис, <https://czo.gov.ua/sign> та інші.

Контрольні питання:

1. Що таке КЕП?
2. Навіщо використовують КЕП?
3. Які сервіси і застосунки використовуються для створення КЕП?
4. Як перевірити документ на наявність КЕП?

ПРАКТИЧНА РОБОТА № 2

Тема: Створення і налаштування профілю у системі Git.

Мета: закріпити теоретичні знання і розвинути практичні навички у використанні системи контролю версій Git.

За результатами виконання практичної роботи здобувачі мають набути **уміння** у створенні, налаштуванні та підтримці власного профілю на найбільшому хостингу для сховищ Git.

Короткі теоретичні відомості

Git – це система для керування та контролю версіями. Це найпопулярніший та безкоштовний інструмент, в якому зберігається код та історія його змін.

До основних завдань Git належить збереження коду та історії змін, збереження інформації про користувачів, які змінюють код, можливість відкотити код до будь-якої версії, можливість об'єднувати різні версії, зміни версій, підготовка кінцевого коду до релізу.

Хостинг **GitHub** – це інструмент для зберігання та управління репозиторіями Git. Він дозволяє:

- зберігати код;
- використовувати інструменти для спільної роботи;
- оцінювати роботи інших розробників;
- створювати приватні та публічні репозиторії (за приватні стягується плата користування від трьох і більше осіб).

Завдання.

Створити власний репозиторій в GitHub. В подальшому усі результати своїх практичних робіт необхідно буде завантажувати у цей репозиторій. В репозиторії створити для кожної практичної роботи окрему папку і розмістити в ній звіт.

Якщо ви ще не маєте обліковий запис, то в професійному світі запис повинен виглядати, як поєднання першої літери ім'я та прізвище в повному виді. Наприклад, Микола Єфремов - «myefremov», а Тарас Шевченко - «tshevchenko». Посилання на репозиторій надаєте викладачеві.

Приклад виконання завдання.

Для створення облікового запису у системі GitHub. необхідно виконати декілька простих дій:

- Перейдіть на сайт <https://github.com/>
- Натисніть кнопку “Sign up” у верхньому правому куті сторінки.

- Заповніть реєстраційну форму, вказавши ваше ім'я користувача, адресу електронної пошти та пароль.
- Натисніть “Create account” та дотримуйтесь інструкцій на екрані для завершення реєстрації.

Рисунок 2.1 - Форма реєстрації на сайті GitHub

Далі необхідно створити нового проєкт (репозиторій) для розміщення всіх практичних робіт в ньому.

Репозиторій — це місце може містити файли коду, документацію, зображення і будь-які інші файли, які стосуються вашого проєкту.

Для **створення нового репозиторію** під назвою «arz» заходимо в наш обліковий запис на сайті <https://github.com> і створюємо репозиторій за допомогою кнопки «New» (рис. 2.1).

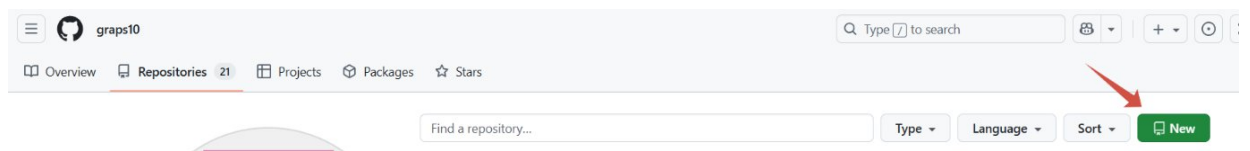


Рисунок 2.2 – Створення нового репозиторію на сайті GitHub

Вводимо назву репозиторію, робимо **публічний доступ (Public)** і натискаємо на кнопку «Create repository».

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

General

Owner * / Repository name *

✔ apz is available.

Great repository names are short and memorable. How about **musical-doodle**?

Description

0 / 350 characters

Configuration

Choose visibility *

Choose who can see and commit to this repository

Рисунок 2.3 – Створення нового публічного репозиторію на GitHub для розміщення практичних робіт

Зверніть увагу. У публічному репозиторії (**Public**) всі файли в ньому зможе побачити будь-яка людина в інтернеті, в той час, як приватний буде доступний лише тим людям, кому ви безпосередньо надаєте доступ.

Наступним кроком для кожної виконаної практичної роботи додаємо окрему папку у створеному репозиторії «apz» (рис. 2.4 - 2.5) і завантажуюмо в неї звіт за допомогою кнопки «Upload files» (рис. 2.6).

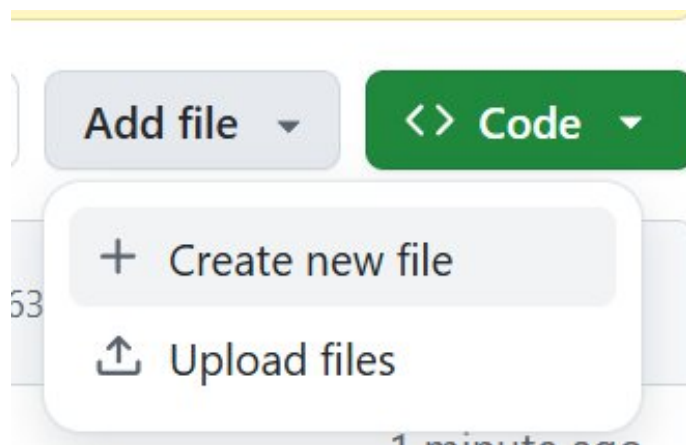


Рисунок 2.4 – Створення папки у репозиторії

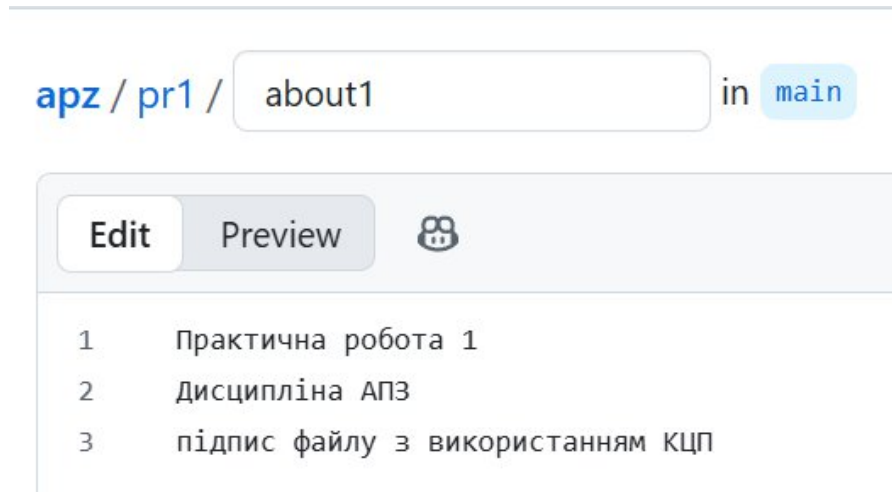


Рисунок 2.5 – Додавання папки pr1 у репозиторій apz з текстовим файлом about

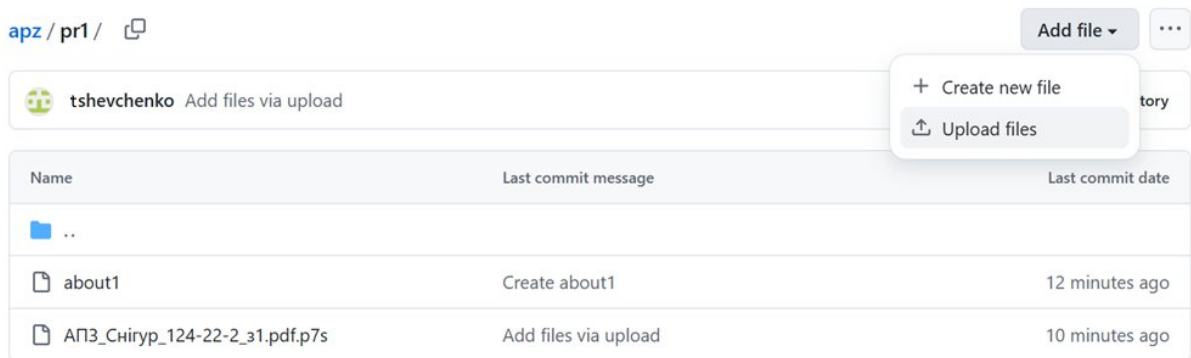


Рисунок 2.6 – Завантаження файлів у папку для практичної роботи 1

Важливо. Надалі всі практичні роботи будуть перевірятися викладачами тільки на сайті <https://github.com>.

Посилання на ваш публічний репозиторій зі звітами ви надаєте викладачеві під час захисту практичної роботи.

Контрольні питання:

1. Що таке контроль версій Git?
2. Що таке репозиторій у Git?
3. Які переваги використання Git?
4. Як можна створити репозиторій у Git?
5. Чи можна надати доступ до репозиторію сторонній людині?

ПРАКТИЧНА РОБОТА № 3

Тема: Написання тест-кейсів (**Test Case**).

Мета: закріпити теоретичні знання і розвинути практичні навички із роботи QA\QC.

За результатами виконання практичної роботи здобувачі мають набути вміння створювати чіткі, лаконічні та атомарні тест-кейси при тестування різних пристроїв та програм, а також вміння інтерпретувати результати виконання різних тестів.

Короткі теоретичні відомості

Написання тест-кейсів (**test cases**) — це мистецтво перетворення вимог до продукту на чіткі інструкції для перевірки.

Структура класичного тест-кейсу:

- ~ **ID:** Унікальний ідентифікатор (*наприклад*, TC-01).
- ~ **Назва** (Title): Короткий опис того, що перевіряється (*наприклад*, "Валідація входу з коректним паролем").
- ~ **Пріоритет** (Priority): High, Medium або Low.
- ~ **Попередні умови** (Pre-conditions): Стан системи до початку тесту (*наприклад*, "Користувач вже зареєстрований").
- ~ **Кроки** (Steps): Послідовність дій.
- ~ **Очікуваний результат** (Expected Result): Що саме має статись після виконання кроків.
- ~ **Фактичний результат** (Actual Result): Заповнюється під час тестування.

Правила хорошого тест-кейсу

- ~ Атомарність: один кейс перевіряє лише одну логічну річ.
- ~ Незалежність: виконання одного тесту не повинно залежати від результату попереднього.
- ~ Чіткість: Уникайте слів "красиво", "швидко", "правильно".

Результатом (фактичним) виконання тест-кейсу можуть бути:

- ~ Позитивний результат (**pass**) – фактичний результат дорівнює очікуваному результату.
- ~ Негативний результат (**fail**) – фактичний результат не дорівнює очікуваному результату. У цьому випадку знайдено помилку.
- ~ Виконання тесту заблоковано (**blocked**) – після одного з кроків продовження тесту неможливе. У цьому випадку також знайдено помилку.

Завдання.

Придумайте об'єкт тестування. Це повинен бути будь-який об'єкт, який складається мінімум з 5 частин. (годинник, скейт, велосипед, монітор і т.п.) Напишіть не менше 10 тест кейсів до цього об'єкту, що зможуть оцінити його якість з точки зору продукту. В звіті роботи коротко опишіть ваш об'єкт і тест-кейси.

Приклад виконання завдання.

Тест-кейс: ТС-101. **Авторизація зареєстрованого користувача**

Пріоритет: High (Високий) **Тип:** Positive (Позитивний) **Компонент:** Login Module **Пристрій:** Смартфон (Android/iOS).

Поле	Опис
Передумови (Pre-conditions)	1. Додаток встановлено на пристрій. 2. Користувач вже зареєстрований із логіном testuser@mail.com та паролем Password123. 3. Пристрій підключено до інтернету.
Кроки (Steps)	1. Відкрити мобільний додаток. 2. Натиснути на кнопку «Увійти» на головному екрані. 3. У поле «Email» ввести testuser@mail.com. 4. У поле «Пароль» ввести Password123. 5. Натиснути кнопку «Увійти».
Очікуваний результат (Expected Result)	1. Відбувається перехід на головний екран особистого кабінету. 2. З'являється вітальне повідомлення «Вітаємо, testuser!». 3. Дані користувача завантажуються коректно.
Постумови (Post-conditions)	Користувач авторизований у системі, сесія активна.

Контрольні питання

1. Навіщо потрібні тест-кейси?
2. Основні атрибути тест-кейсу?
3. Які типи тест-кейсів існують?
4. Що таке негативний тест-кейс?

ПРАКТИЧНА РОБОТА № 4

Тема: Amazon Simple Storage Service (AWS S3).

Мета: закріпити теоретичні знання і розвинути практичні навички із створення AWS S3.

За результатами виконання практичної роботи здобувачі мають набути **вміння** у створення і розміщенні власної статичної веб-сторінки на сервіс зберігання даних AWS S3.

Короткі теоретичні відомості

Amazon Simple Storage Service, відомий як **Amazon S3** — це один із найпопулярніших сервісів хмарного зберігання даних у світі. Якщо пояснювати просто: це "нескінченна флешка" в інтернеті, де ви можете зберігати будь-яку кількість файлів і мати до них доступ з будь-якої точки світу.

Цей сервіс дозволяє зберігати будь-який тип даних: документи, зображення, відео, резервні копії, веб-сторінки тощо.

Ключові можливості:

- ~ **Версійність (Versioning):** збереження кількох варіантів об'єкта в одному кошику для захисту від випадкового видалення.
- ~ **Безпека:** підтримка шифрування (SSE-S3, SSE-KMS), керування доступом через IAM-політики та ACL.
- ~ **Реплікація:** можливість автоматичного копіювання даних між регіонами (CRR) або всередині одного регіону (SRR).
- ~ **Статичний хостинг:** можливість розміщувати статичні вебсайти безпосередньо з кошика.

Завдання.

Зареєструйтесь на сервісі AWS (Amazon Web Services). Розмістити на цьому сервісі S3 статичну веб сторінку на якій буде вказана ваше ПІБ та обов'язково академічна група. Бакет на S3 в якому буде лежати ваша сторінка обов'язково в своєму імені повинен містити ваше прізвище та ім'я.

Наприклад, *myefremov-bucket-apz* для студента з іменем Михайло Єфремов.

Результатом виконання цієї роботи буде адреса на вашу веб-сторінку на подібні <https://myefremov-bucket-apz.amazonaws.com/index.html> на якій буде написано ваш ПІБ та група.

Приклад виконання завдання.

Крок 1. Реєстрація власного акаунт в системі AWS.

1. Перейдіть за посиланням <https://aws.amazon.com/> і натисніть «Create account» (рис. 4.1).
2. Введіть свій email, ім'я і прізвище і придумайте пароль. Далі підтвердіть свій акаунт через надану електронну пошту.
3. Введіть платіжні дані (AWS вимагає банківську картку), але з вас **не знімуться кошти** в межах безкоштовного рівня - **Free Tier** (рис. 4.2)).

Уточнення по банківській карті. Відкриваєте в будь-якому банку віртуальну Інтернет карту з мінімальною сумою на рахунку. Після закінчення курсу обов'язково необхідно буде її видалити для запобігання зняття коштів у подальшому використанні AWS.

4. Виконайте підтвердження особи (SMS/дзвінок).
5. Завершіть реєстрацію та увійдіть у власний профіль на <https://aws.amazon.com>

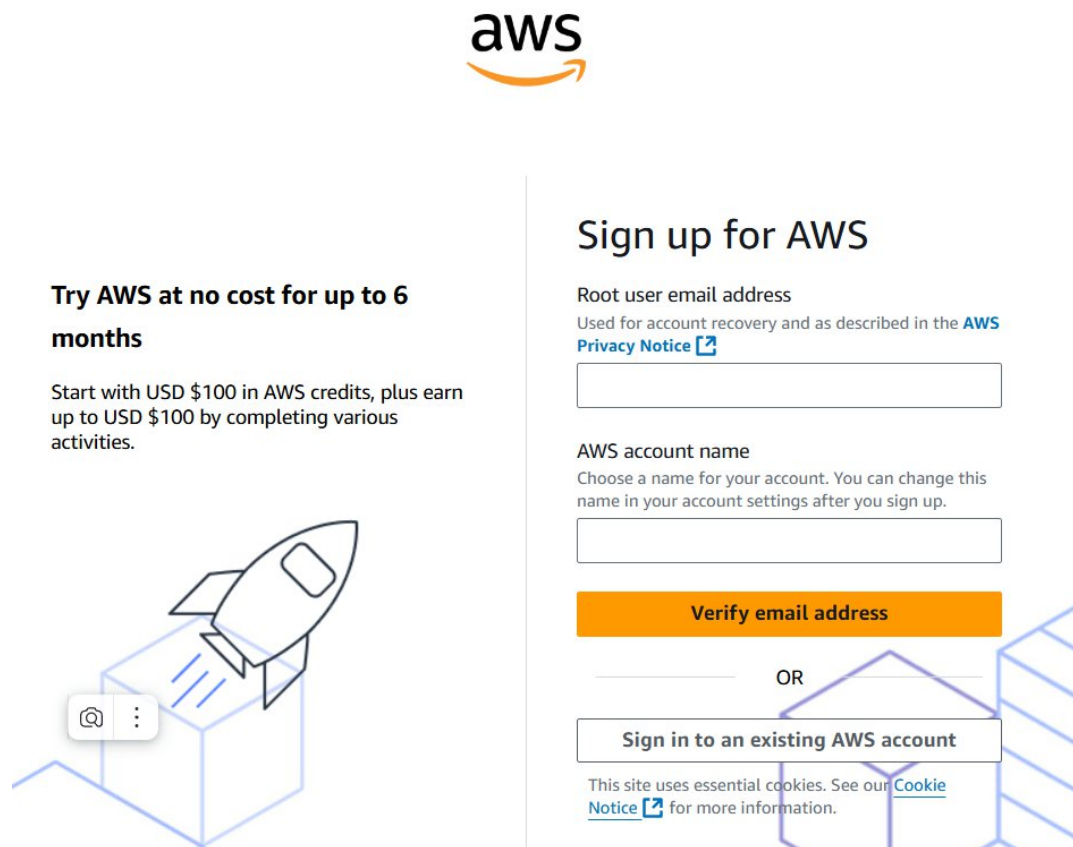





Рисунок 4.1 - Реєстрація власного акаунт в системі AWS



Sign up for AWS

Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)
[↗](#). You can change your plan anytime in the AWS Management Console.

<p><input checked="" type="radio"/> Basic support - Free</p> <ul style="list-style-type: none">• Recommended for new users just getting started with AWS• 24x7 self-service access to AWS resources• For account and billing issues only• Access to Personal Health Dashboard & Trusted Advisor 	<p><input type="radio"/> Developer support - From \$29/month</p> <ul style="list-style-type: none">• Recommended for developers experimenting with AWS• Email access to AWS Support during business hours• 12 (business)-hour response times 	<p><input type="radio"/> Business support - From \$100/month</p> <ul style="list-style-type: none">• Recommended for running production workloads on AWS• 24x7 tech support via email, phone, and chat• 1-hour response times• Full set of Trusted Advisor best-practice recommendations 
---	---	--



Need Enterprise level support?

From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more](#) [↗](#)

[Complete sign up](#)

Рисунок 4.2 - Вибір безкоштовного доступу **Free Tier** до послуг AWS

Крок 2. Створення S3 бакету.

1. Увійдіть свій акаунт на сайті AWS Console за посиланням <https://console.aws.amazon.com/s3/> і натисніть «**Create bucket**» (рис. 4.3).
2. Заповніть уважно всі дані:
 - **Bucket name**: **обов'язково** включіть ваше прізвище та ім'я латиницею, наприклад *kbaleiko-bucket-apz*.
 - **Region**: залиште за замовчуванням або оберіть найближчий (наприклад, Europe (Frankfurt))
3. **Обов'язково** зніміть галочку з "Block all public access" та підтвердьте галочкою, що ви *розумієте наслідки публічного доступу* і наприкінці натисніть «**Create bucket**».

Create bucket [Info](#)

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type [Info](#)

General purpose

Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket storage classes that redundantly store objects across multiple Availability Zones.

Bucket name [Info](#)

holobokov-bucket-apz

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

Рисунок 4.3 - Створення S3 бакету

Крок 3. Створення і розміщення статичної веб-сторінки на AWS.

1. Спочатку на власному ПК у редакторі блокнот створіть файл *index.html* з наступним змістом:

```
<html lang="uk">
<head>
<meta charset="UTF-8">
<title>Моя сторінка</title>
</head>
<body>
<h1>Голобоков Владіслав Юрійович</h1><br />
<p>група 122-22-4</p>
</body>
</html>
```

2. Поверніться на сайт <https://aws.amazon.com> і відкрийте створений на кроці 2 бакет.

3. Перейдіть у вкладку Objects → Upload та завантажте створений файл index.html (рис. 4.4).
4. **Обов'язково** після завантаження оберіть свій завантажений файл index.html і натисніть **Permissions** → **Make public** (рис 4.5).

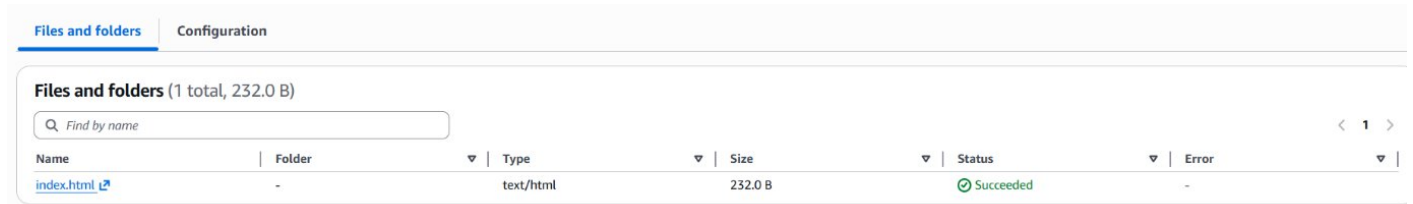


Рисунок 4.4 - Завантаження створеного файлу index.html

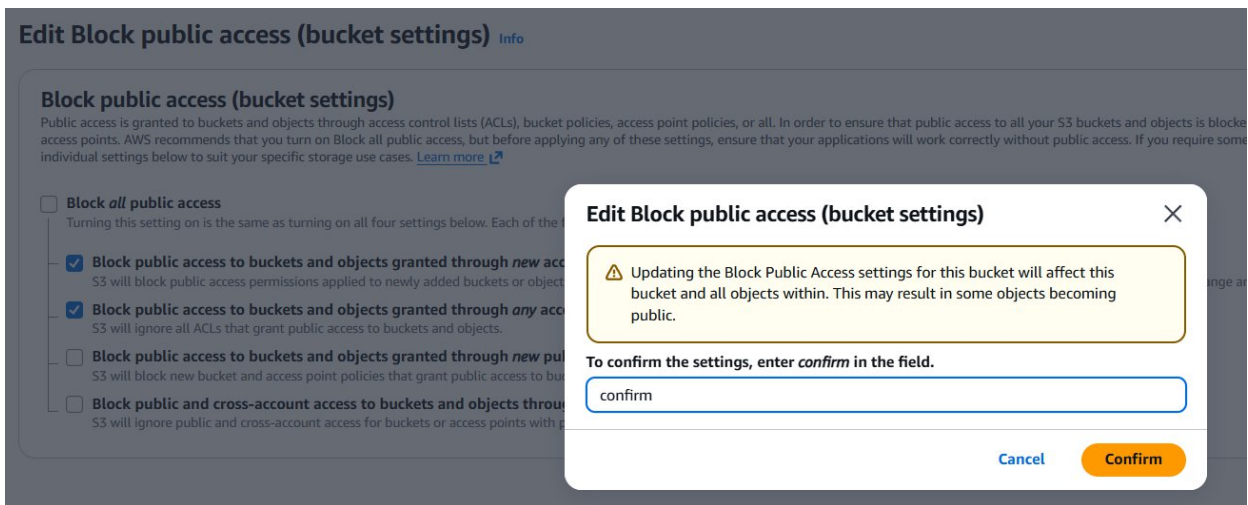


Рисунок 4.5 - Налаштування публічного доступу до сторінки

Крок 4. Налаштування хостингу і отримання публічної адреси сторінки.

1. Поверніться за посиланням <https://aws.amazon.com/> і відкрийте створений бакет (рис. 4.3).
2. Оберіть вкладку **Properties** та прокрутіть до поля **Static website hosting** (рис. 4.6), натисніть "Edit" і увімкніть "Enable".
3. У полі Index document вкажіть назву вашого файлу - index.html і натисніть "Save changes".
4. Для отримання адреси вашої сторінки для надання викладачеві знову перейдіть на вкладку **Properties** → **Static website hosting** і скопіюйте адресу у відповідному полі (наприклад, це буде подібно до <http://holobokov-bucket-apz.s3-website-us-east-1.amazonaws.com/>)

Як результат виконання практичної роботи ви надасте викладачу посилання на вашу веб-сторінку у форматі <http://holobokov-bucket-apz.s3-website-us-east-1.amazonaws.com/> (рис. 4.7).

Edit static website hosting [Info](#)

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

- Disable
 Enable

Hosting type

- Host a static website
Use the bucket endpoint as the web address. [Learn more](#)
- Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

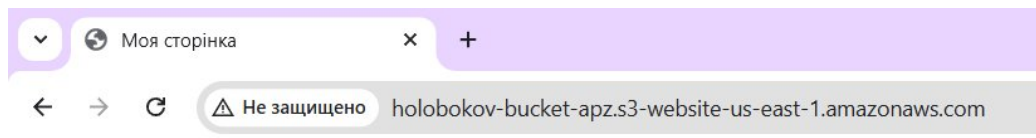
i For your customers to access content at the website endpoint, you must make all your content [Public Access](#)

Index document

Specify the home or default page of the website.

index.html

Рисунок 4.6 - Налаштування *Static website hosting* для сторінки



Голобоков Владіслав Юрійович

122-22-4

Рисунок 4.7 – Фінальний вигляд створеної веб-сторінки

Контрольні питання

1. Для чого найчастіше використовують AWS S3?
2. Які головні переваги використання AWS S3?
3. Що таке Buckets (кошки)?
4. Що в собі має об'єкт на сервісі AWS S3?
5. Як створити власний кошик у сервісі AWS S3?

ПРАКТИЧНА РОБОТА № 5

Тема: Створення віртуальної машини в сервісі Amazon EC2.

Мета: закріпити теоретичні знання і розвинути практичні навички із створення AWS EC2.

За результатами виконання практичної роботи здобувачі мають набути **вміння** у створення, налаштуванні та використанні віртуальної машини у сервісі Amazon EC2 (Elastic Compute Cloud).

Короткі теоретичні відомості

Сервіс **AWS EC2** (Elastic Compute Cloud) — це основний сервіс Amazon Web Services, який надає віртуальні машини в хмарі. По суті, це оренда комп'ютерів, якими ви можете керувати через інтернет.

Основні поняття сервісу EC2:

- **Екземпляр (Instance):** Це і є ваша віртуальна машина. Ви можете вибрати її характеристики: кількість процесорів (vCPU), обсяг оперативної пам'яті (RAM), тип і розмір сховища, а також операційну систему (Linux, Windows тощо).
- **АМІ (Amazon Machine Image):** Готовий шаблон, "знімок" операційної системи та встановленого програмного забезпечення. Використовується як відправна точка для запуску екземпляра.
- **Security Groups** (Групи безпеки): Віртуальний фаєрвол. Визначає, який мережевий трафік (наприклад, HTTP, SSH, RDP) дозволено вхідний та вихідний для вашого екземпляра.
- **EBS (Elastic Block Store):** Постійний дисковий простір (жорсткий диск), який можна приєднати до екземпляра. Дані на ньому зберігаються навіть після зупинки або завершення роботи віртуальної машини.

Завдання.

Вже в існуючому акаунті AWS (практична робота 4) створити віртуальну машину в сервісі EC2. Машину створити на операційній системі Windows. В цій віртуальній машині зробити найпростішу зміну – а саме, встановити на робочий стіл картинку с вашим ПІБ та номером групи.

Результатом роботи вашої машини буде її зовнішня IP адреса (не плутати з внутрішньою) та пароль адміністратора. *Наприклад*, дані для підключення:

IP: 13.60.68.81

User: Administrator

Password: !c7ugFMG;);zoovxF)-BkT87B!&mVU%o

Приклад виконання завдання.

Крок 1. Створення і налаштування екземпляру (Instance) ПК.

Для цього входимо в ваш профіль на сайті <https://aws.amazon.com> і обираємо вкладку EC2 – Instance (рис. 5.1).

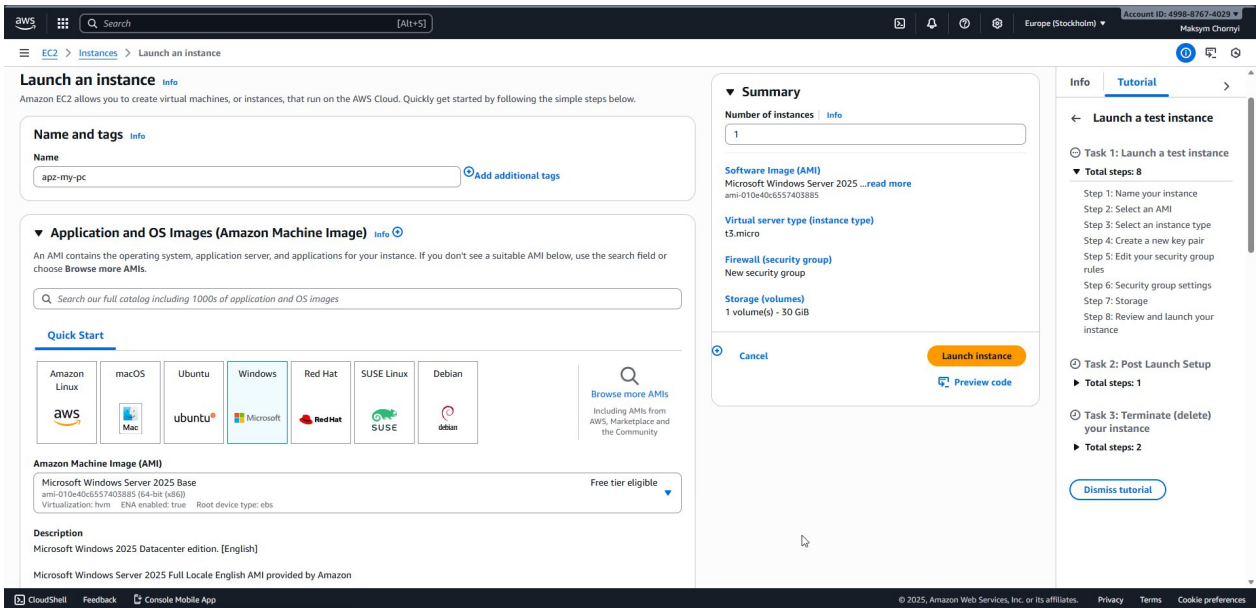


Рисунок 5.1 - Створення власного екземпляру віртуальної машини (Instance)

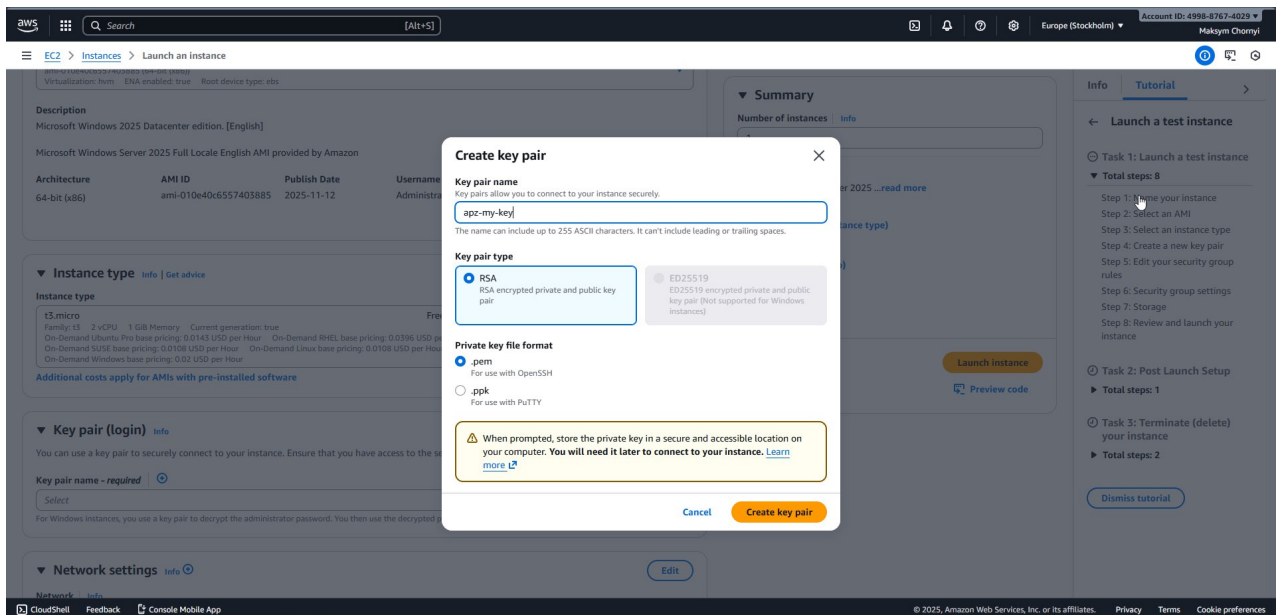


Рисунок 5.2 - Створення пари ключів (key pair)

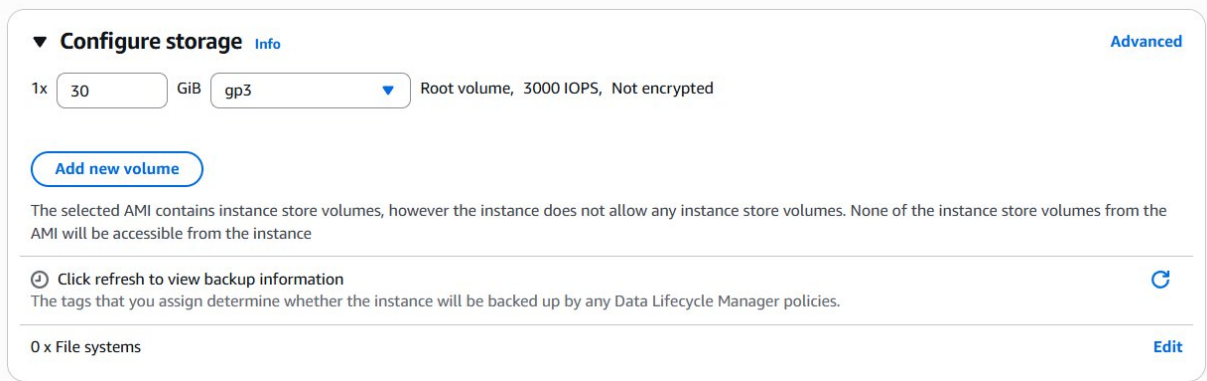


Рисунок 5.3 - Налаштування конфігурації машини (*Configure Storage*)

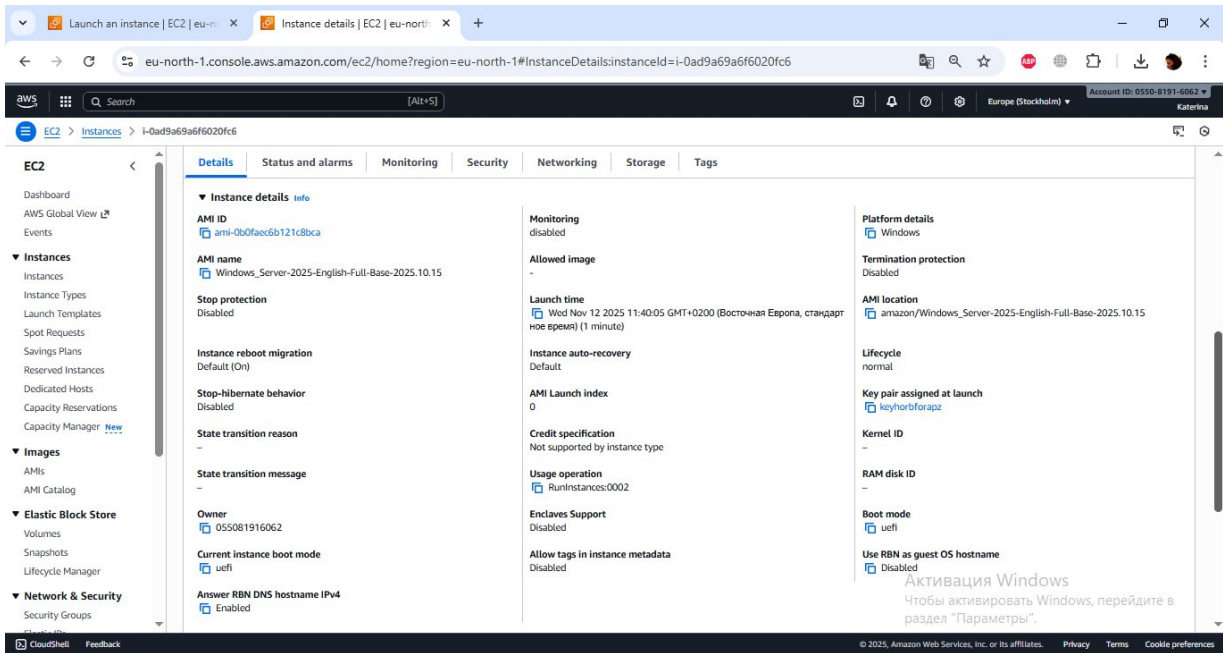


Рисунок 5.4 - Результат створення екземпляру (*Instance*) на AWS

Крок 2. Отримання зашифрованого паролю для віртуальної машини через консоль EC2:

- Виберіть пункт Instances і позначте потрібну віртуальну машину.
- Натисніть кнопку **Actions -> Security -> Get Windows password**.
- Натисніть на “Upload private key file” і обравши ваш файл *.pem натисніть на **Decrypt Password**.

Якщо все вірно, то ви побачите розшифрований пароль, який можна використовувати для входу на вашу віртуальну машину через протокол RDP (**Remote Desktop Protocol**) на будь-якому комп'ютері.

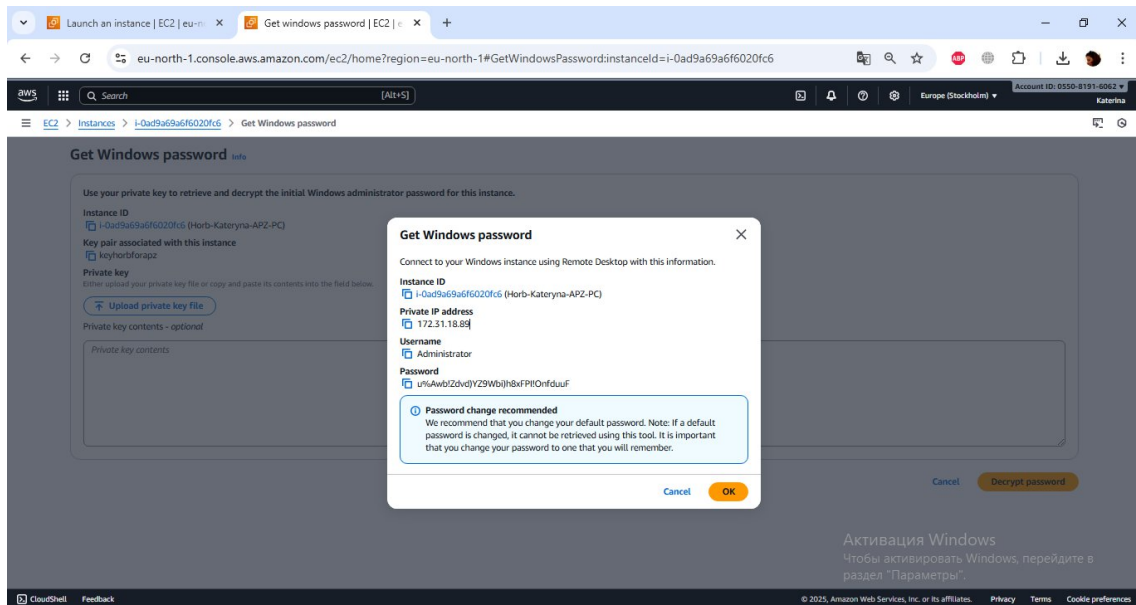


Рисунок 5.5 - Отримання даних машини і зашифрованого паролю

Крок 3. Підключення до створеної віртуальної машини.

Для підключення в ОС Windows необхідно:

- Натисніть комбінацію клавіш "Win + R", введіть команду "mstsc" у віконце (рис. 5.6).
- У полі Комп'ютер введіть Public IP вашої віртуальної машини (рис. 5.7) і натисніть Підключити (Connect). З'явиться вікно введення даних
- Натисніть "Більше варіантів" - "Використовувати інший обліковий запис".
- Введіть логін (зазвичай Administrator) та ваш розшифрований пароль, який ви отримали на попередньому кроці (рис. 5.8). Якщо з'явиться попередження про сертифікат безпеки, натисніть "Так".

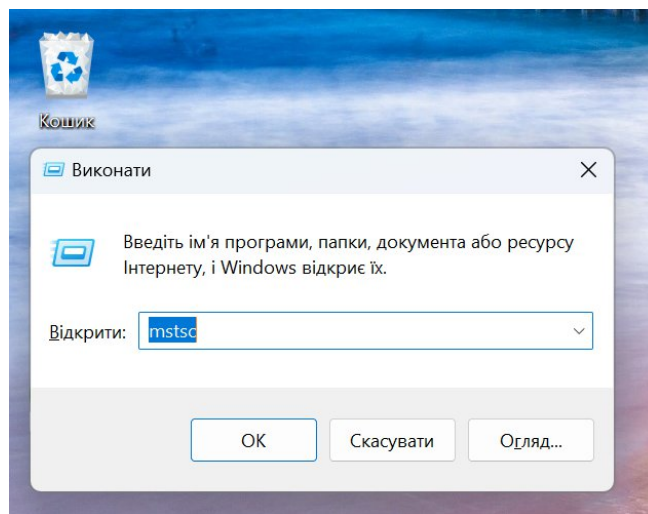


Рисунок 5.6 – Виклик командної строки в ОС Windows

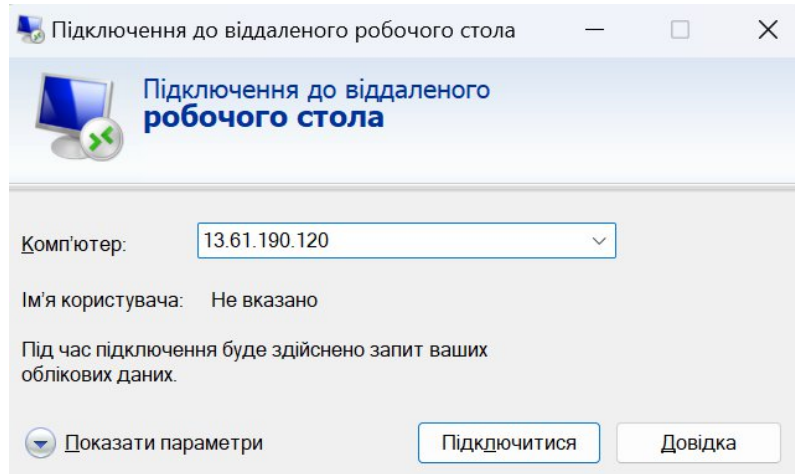


Рисунок 5.7 - Підключення до створеної віртуальної машини

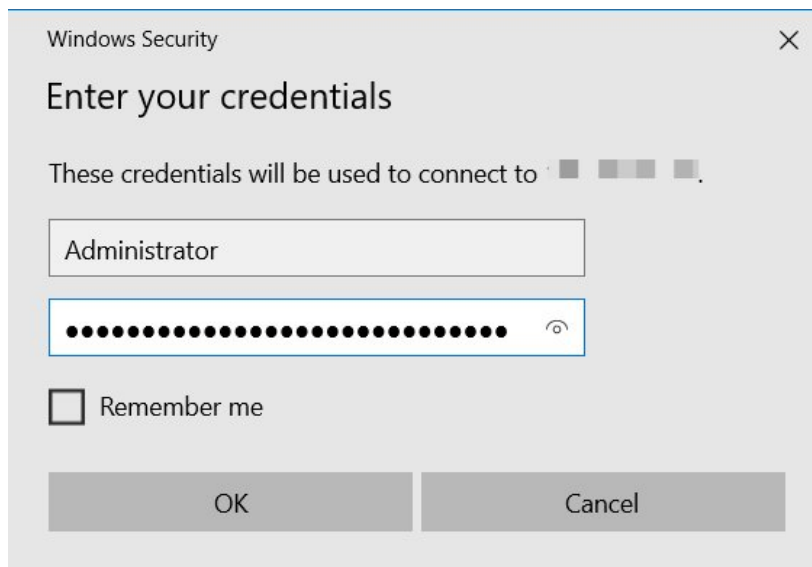


Рисунок 5.8 - Введення username/password

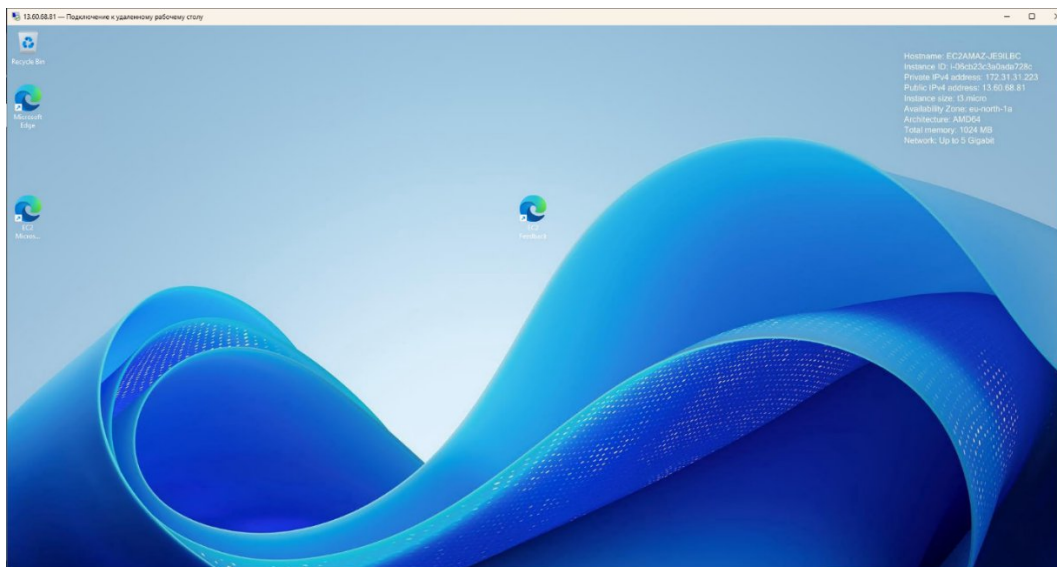


Рисунок 5.8 – Стандартний вигляд робочого стола віртуальної машини на AWS

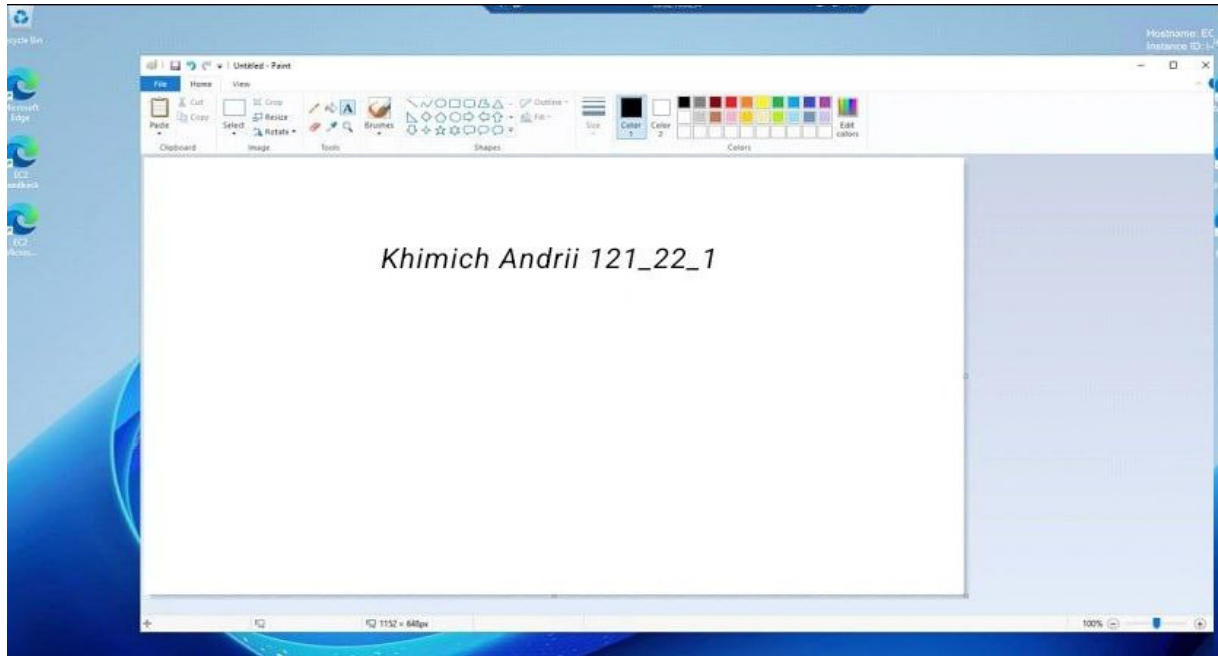


Рисунок 5.10 –Зміни на робочому столі віртуальної машини

Як **результат виконання роботи** викладачу надається для перевірки дані про створену віртуальну машину (IP, Username і Password – рис. 5.5), а також скріншот робочого стола, де показані якісь зміни на робочому столі вашої віртуальної машини (рис. 5.10).

Наприклад:

IP: 13.60.68.81

User: Administrator

Password: !c7ugFMG;);zoovxF)-BkT87B!&mVU%o

Контрольні питання

1. Яка різниця між "Stop" та "Terminate"?
2. Чим відрізняються On-Demand, Reserved та Spot інстанси?
3. Навіщо потрібен Security Group?
4. Що таке "Region" в AWS?
5. Як підключитися до Windows-інстансу?

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Кім Дж., Гамбл Дж., Дебуа П., Вілліс Дж. DevOps. Посібник із того, як зробити вашу IT-організацію максимально продуктивною, гнучкою та безпечною. Київ : Наш Формат, 2018. 464 с.
2. Джефф Сазерленд "Scrum: Мистецтво робити вдвічі більше за менший час". – Київ: Наш Формат, 2019. – 256 с.
3. Віттінг А., Віттінг М. Amazon Web Services in Action. Shelter Island : Manning Publications, 2019. 200 p.
4. Кангассу, Х. Забезпечення якості програмного забезпечення: Керівництво для інженерів і менеджерів / Х. Кангассу. – CRC Press, 2021.
5. Aniche M. Effective Software Testing: A developer's guide. Shelter Island : Manning Publications, 2022. 336 p.
6. Офіційний сайт Google: Compare AWS and Azure services to Google Cloud. 2023 url: <https://cloud.google.com/free/docs/aws-azure-gcp-servicecomparison>
7. Керцнер, Г. Управління проєктами: Комплексний підхід / Г. Керцнер. – Харків : Фоліо, 2019.
8. Коган, М. Agile Project Management: From Waterfall to Agile / М. Коган. – Харків : Фоліо, 2018.
9. Коен, М. Agile Estimating and Planning / М. Коен. – Київ : Академія, 2021.
10. Гопкінс, Д. Проєктний менеджмент для IT: Практичний підхід / Д. Гопкінс. – Харків : Фоліо, 2021. – 300.
11. Фоулер, М. Архітектура корпоративних додатків / М. Фоулер. – Київ : Наш Формат, 2021. – 544 с.
12. Very Verified: онлайн-курс з медіаграмотності | EdEra. EdEra – студія онлайн-освіти. URL: <https://study.ed-era.com/uk/courses/course/2118> (date of access: 09.01.2026).
13. Хом'як, Т. В., Шевченко, Ю. О., & Гаранжа, Д. М. (2026). Програмування та алгоритмічні мови [Електронний ресурс]: методичні рекомендації до виконання лабораторних робіт в 2-му семестрі для здобувачів ступеня бакалавра освітньо-професійної програми «Системний аналіз» зі спеціальності 124 Системний аналіз. URL: <https://ir.nmu.org.ua/handle/123456789/173149> (date of access: 03.02.2026).

ДОДАТОК А. ПРИКЛАД ОФОРМЛЕННЯ ТИТУЛЬНОЇ СТОРІНКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ДНІПРОВСЬКА ПОЛІТЕХНІКА»



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра системного аналізу та управління

ПРАКТИЧНА РОБОТА № 1

з дисципліни «Аналіз програмного забезпечення»

на тему: «Підписання персонального документа за допомогою
кваліфікованого електронного підпису (КЕП)»

Виконав:
здобувач групи СА-25-1
Шевченко Олег

Прийняв:
доц. Мінеєв О.С.

Дніпро
2026

Навчальне видання

Мінєєв Олександр Сергійович
Шевченко Юлія Олександрівна

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
галузі знань 12 (F) Інформаційні технології

Видано в авторській редакції

Електронний ресурс.
Підписано до видання 23.03.2026. Авт. арк. 0,67.

Національний технічний університет «Дніпровська політехніка»
49005, м. Дніпро, просп. Дмитра Яворницького, 19