

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



Факультет інформаційних технологій
Кафедра системного аналізу і управління

О.С. Мінесєв, Ю.О. Шевченко

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Методичні рекомендації
до практичних занять з дисципліни

для здобувачів спеціальності F4 Системний аналіз та наука про дані

Електронне мережеве навчальне видання

Дніпро
НТУ «ДП»
2025

Аналіз програмного забезпечення [Електронний ресурс]: методичні рекомендації до практичних занять з дисципліни для здобувачів в ступеня бакалавра спеціальності F4 Системний аналіз та наука про дані / уклад.: О.С. Мінеєв, Ю.О. Шевченко ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2025. – 66 с.

Укладачі:

О.С. Мінеєв, канд. техн. наук, доц.;

Ю.О. Шевченко, асист.

Затверджено до видання науково-методичною комісією зі спеціальності F4 Системний аналіз (протокол № 3 від 07.05.2025) за поданням кафедри системного аналізу та управління (протокол № 7 від 07.05.2025).

Наведено матеріал лекційних занять відповідно до освітньо-професійної програми підготовки бакалаврів зі спеціальності F4 Системний аналіз та наука про дані.

Орієнтовано на активізацію навчальної діяльності здобувачів у галузі навчань F Інформаційні технології та закріплення практичних навичок у засвоєнні дисципліни.

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

Зміст

	стор.
Вступ	4
Практична робота №1. ЕЦП.....	5
Практична робота №2. GIT.....	17
Практична робота №3. Написання Test cases.....	24
Практична робота №4. AWS S3.....	43
Практична робота №5. AWS EC2.....	51
Практична робота №6. Scrum гра.....	66
Оцінювання результатів навчання.....	78
Рекомендовані джерела інформації.....	80

ВСТУП

На сучасному етапі розвитку ІТ кожен робітник компанії повинен розуміти як працює система і як кожен долар та кожне бажання проходить від замовника через всю систему до кінцевого робітника. Також ІТ компанії на від мінус від інших впроваджують багато речей що покращує якість продукту, зменшує бюрократичне навантаження та покращують психологічний настрій в компанії. Студенти відчують себе в тілі кожного члена команди та зможуть в повному обсязі зробити для себе висновки хто і чим займається.

Мета дисципліни – надання знань та вмінь щодо розуміння системи функціонування ІТ компанії розуміння як офіційних так і не офіційних правил поведінки в корпоративному житті . Результатами вивчення даної дисципліни є придбання навичок зі знання та вміння вже з першого дня включитись до корпоративного життя і без великих консультацій почати свою кар'єру для покращення якості своєї роботи і якості програмного забезпечення.

Очікувані результати навчання за дисципліною:

– знати ;

– знати і володіти поняттями CI\CD та контейнеризації.

–

– застосовувати методи і засоби роботи з даними і знаннями, методи математичного, логіко-семантичного, об'єктного та імітаційного моделювання.

– застосовувати методи і засоби роботи технології системного і статистичного аналізу;

– вміти використовувати хмарні середовища на приклади AWS.

–

Практичні роботи виконуються кожним здобувачем індивідуально згідно завдання.

Практична робота №1

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

Мета: Набування навичок підписання особистої документації з використанням кваліфікованого електронного підпису.

Очікувані результати навчання: уміння 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.

Перевірити накладання підпису на е-документ можна на сайті центрального засвідчувального органу Міністерства цифрової трансформації України <https://czo.gov.ua/verify> або за допомогою спеціального програмного забезпечення для звітності та електронного документообігу.

Завдання.

Створити документ формату *.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 безкоштовні сервісів - <https://sign.dia.gov.ua/> чи Дія. Результат виконання у форматі *.p7s надати викладачеві для перевірки.

Приклад щодо накладання кваліфікованого електронного підпису.

Крок 1. Переходимо за посиланням: <https://czo.gov.ua/> (Центральний засвідчувальний орган Міністерство цифрової трансформації України). Далі переходимо на сторінку підписання документації <https://czo.gov.ua/sign>.

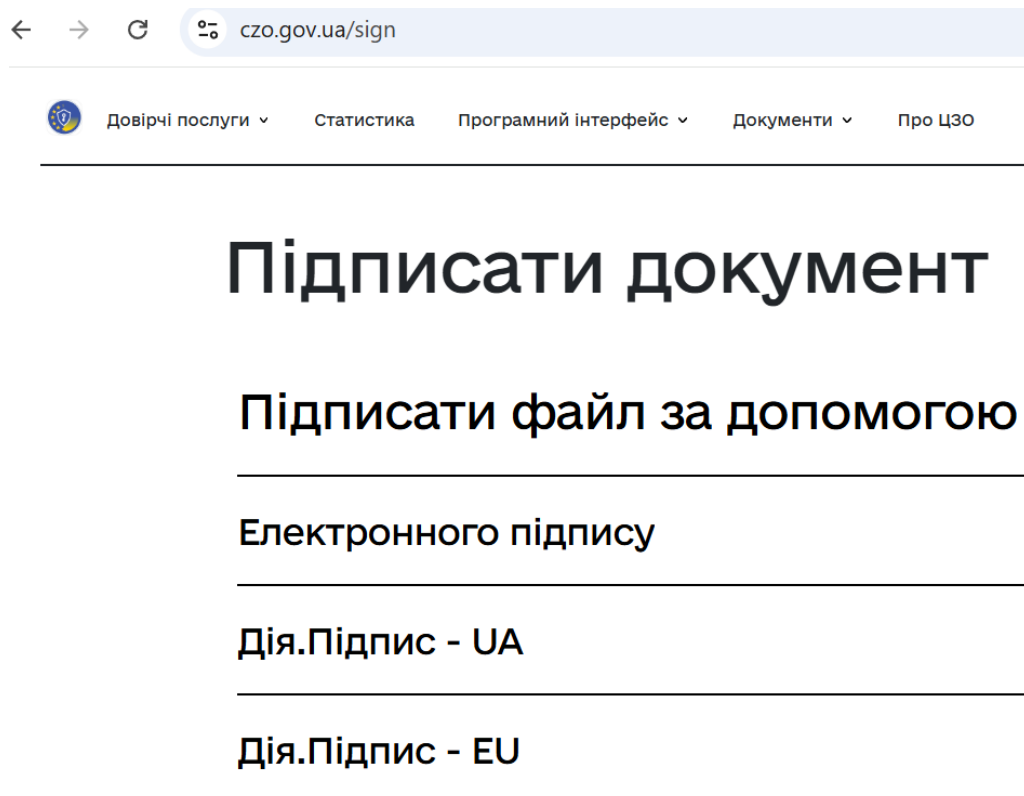


Рис. 1.1 – Сайт Міністерства цифрової трансформації України

а) Обираєте спосіб підписання. Наприклад за допомогою електронного ключа, який ви створили завчасно на сайті ПриватБанку (рис. 1.2).

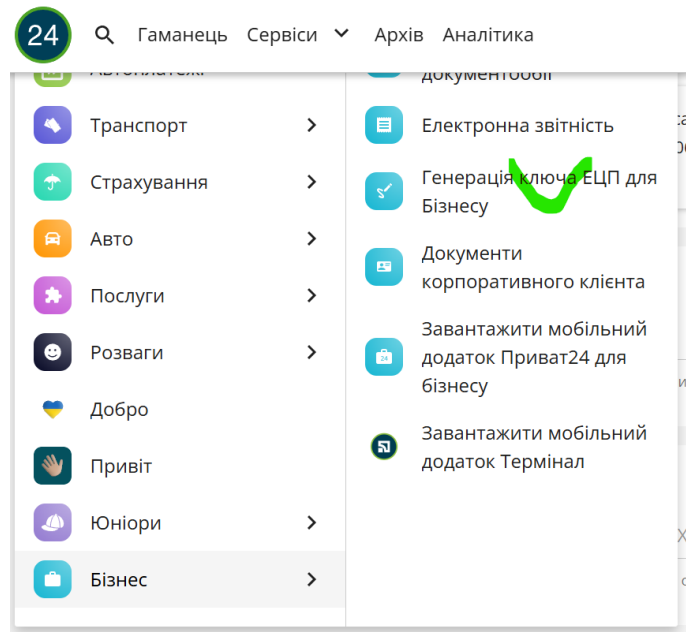


Рис. 1.2 – Генерація ЕЦП

б) Виконуємо дії по ідентифікації електронного ключа: перетягнути файл ключа та ввести пароль від файлу ключа.

Підписати документ

Файловий Токен Хмарний

Що таке файловий носій?

Кваліфікований надавач електронних довірчих послуг

Визначити автоматично

pb_2891608122.jks
Змінити

Ім'я ключа
pb_sign_2891608122(ШЕВЧЕНКО ЮЛІЯ ОЛЕКСАНДРІВНА)

Пароль захисту ключа

Назад Зчитати

Рис. 1.3 – Ідентифікації електронного ключа

в) Натискаємо кнопку «Зчитати ключ».

г) Виконуємо підписання документа (**Крок 2**).

Крок 1. Обираємо формат документа, який буде згенерований після підписання документа.

Підписати документ

Крок 3 з 4

Підписати та зберегти

Що таке ASiC?

ASiC – структурований контейнер, що дозволяє зберігати набір файлових об'єктів з пов'язаними е-підписами та/або е-позначками часу, що відповідає специфікації ZIP.

ASiC-S дозволяє зберігати один файловий об'єкт з пов'язаним е-підписом та в подальшому додавати нові. Також дає можливість додавати файли для захисту е-позначок часу.

ASiC-E дозволяє зберігати один або кілька файлових об'єктів з пов'язаними е-підписами та в подальшому додавати файлові об'єкти, файли е-підпису та е-позначки часу.

👍 **Рекомендуємо підписувати документи у форматі ASiC-E.**

Це уніфікований формат електронного документообігу, який гарантує, що ваші документи прийматимуть всі держоргани.

Підписати та зберегти

Що таке ASiC?

👍 **Рекомендуємо підписувати документи у форматі ASiC-E.**

Це уніфікований формат електронного документообігу, який гарантує, що ваші документи прийматимуть всі держоргани.

Так, підписати в форматі ASiC-E

Ні, обрати інший формат

Крок 2. Підписання документа.

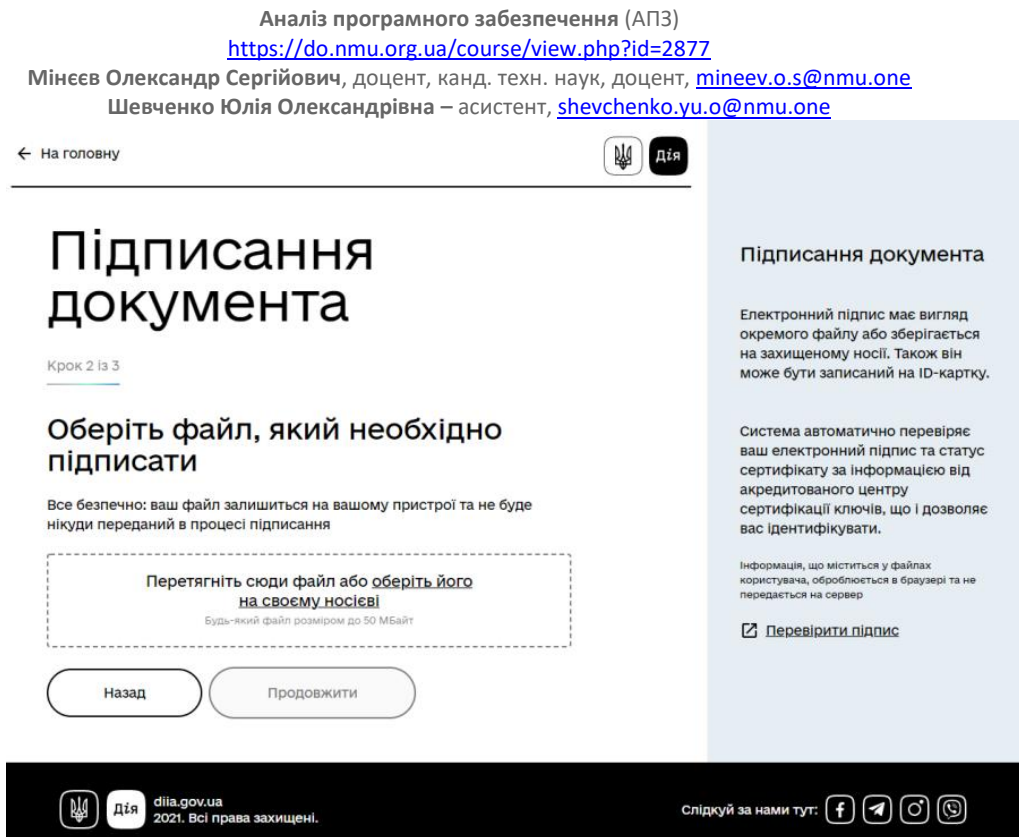


Рис. 1.2 – Обрання файлі для підписання

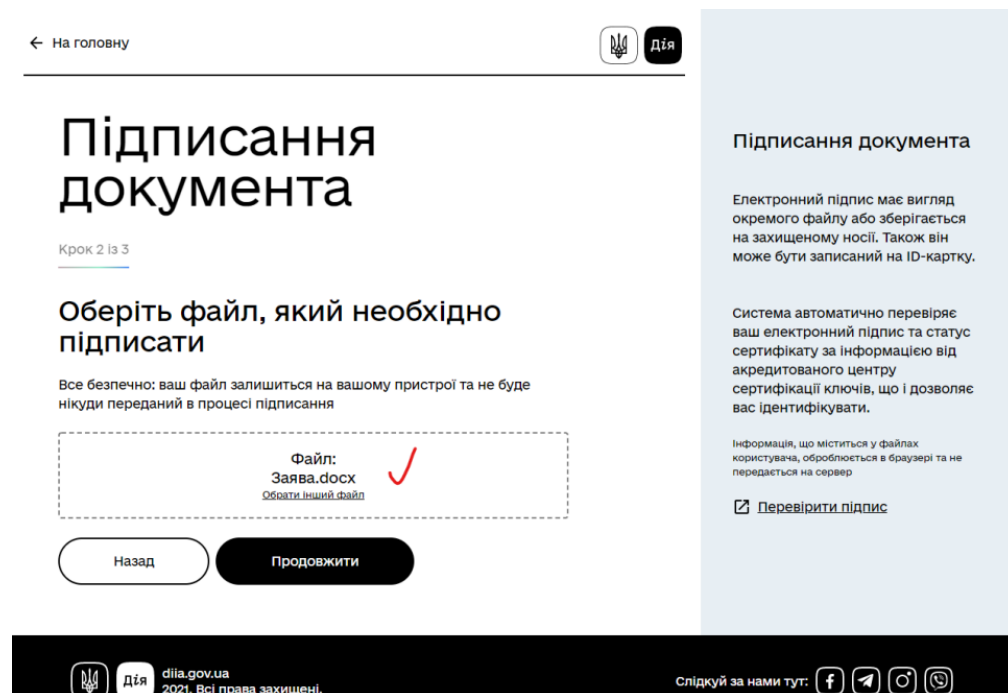


Рис. 1.3 – Підтвердження обрання файлу для підписання

Після вибору файлу натискаємо кнопку «Продовжити».

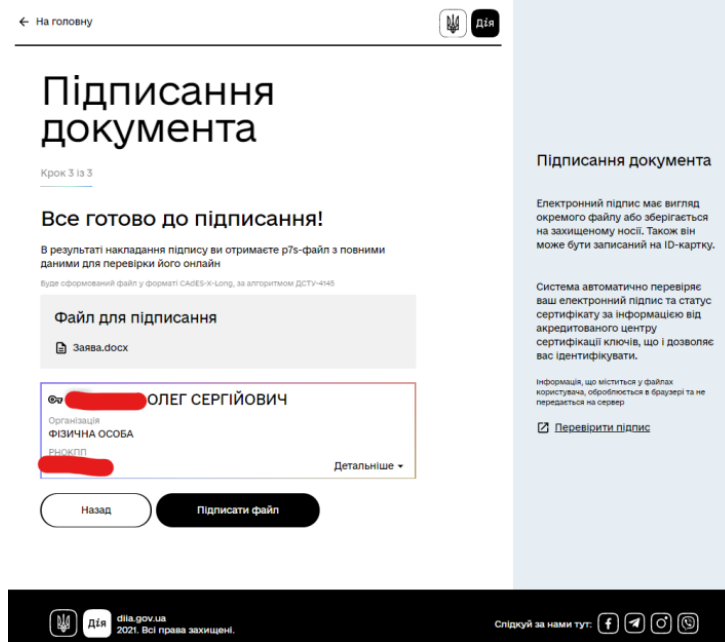


Рис. 1.4 - Обрання файлу для підписання

Наступним кроком натискаємо кнопку «**Підписати файл**».

Після натискання «Підписати файл» відкривається вікно із наступною інформацією:

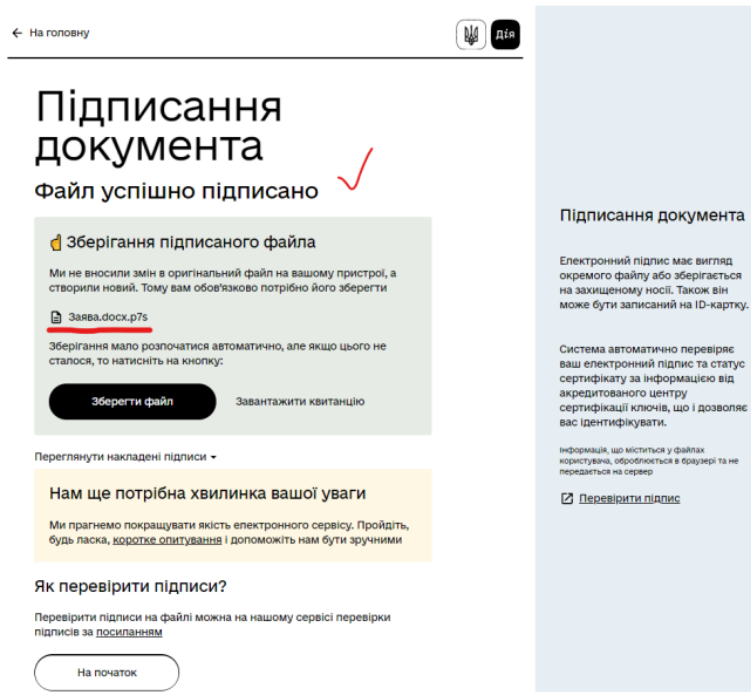


Рис. 1.5 Результат підписання на документ ЕЦП

У результат підписання створюється файл із цифровим підписом із

розширенням *.p7s. Наприклад від буде мати назву «Заява.p7s»

Далі для збереження і завантаження файлу для особистого використання і надання викладачеві на перевірку натискаємо «**Зберегти файл**».

Контрольні питання

1. Що таке ЕЦП?
2. Навіщо використовують КЕП?
3. Які сервіси і застосунки використовуються для створення ЕЦП?
4. Який формат має документ після підписання ЕЦП?

Практична робота №2

Тема: Створення і налаштування профілю у системі Git.

Мета: Набування навичок при реєстрації та налаштуванню облікового запису (account) на хостінгу GitHub.

Очікувані результати навчання: уміння створити, налаштувати та підтримувати власний профіль на найбільшому хостінгу 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.

GitHub – це інструмент для зберігання та управління репозиторіями Git.

GitHub 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 завантажувати у цей репозиторій. В репозиторії створити для кожної практичної роботи окрему папку і розмістити звіт. Якщо ви ще не маєте обліковий запис, будь ласка, в професійному світі, запис повинен виглядати, як поєднання першої літери ім'я та прізвище в повному виді. Наприклад Микола Єфремов буде «myefremov», а Тарас Шевченко «tshevchenko».

Приклад виконання завдання.

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

- 1) Перейдіть на сайт <https://github.com/>
- 2) Натисніть кнопку “Sign up” у верхньому правому куті сторінки.
- 3) Заповніть реєстраційну форму, вказавши ваше ім'я користувача, адресу електронної пошти та пароль.
- 4) Натисніть “Create account” та дотримуйтесь інструкцій на екрані для завершення реєстрації.

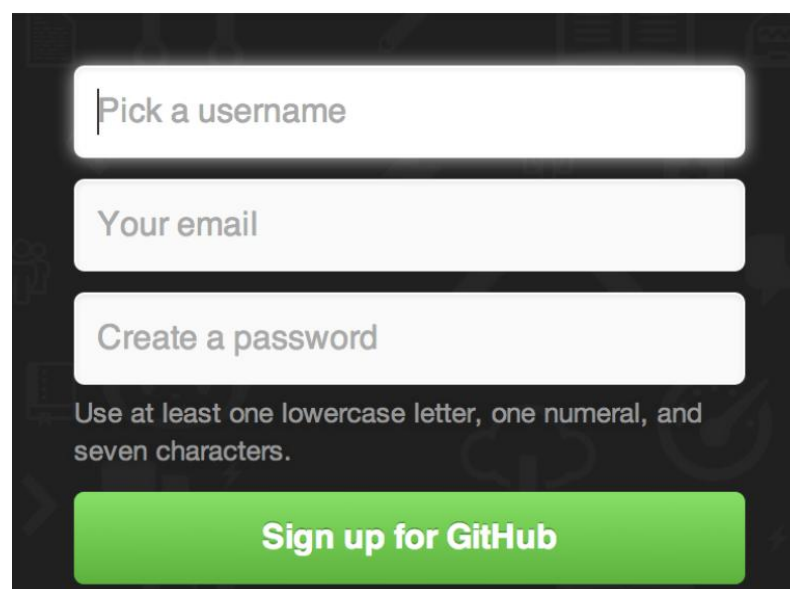


Рис. 2.1 - Форма реєстрації на сайті GitHub

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 проєкту.

Для того, щоб створити **новий репозиторій** на GitHub, виконайте наступні дії:

- 1) Увійдіть у ваш обліковий запис на GitHub - “Sign up” у верхньому правому куті сторінки.
- 2) Натисніть на кнопку “New” у верхньому правому куті сторінки, щоб створити новий репозиторій.
- 3) Введіть назву вашого репозиторію і, за бажанням, його опис.
- 4) Виберіть, щоб ваш репозиторій був репозиторій **публічним** (*Public*).

Публічний репозиторій і всі файли в ньому зможе побачити будь-яка людина в інтернеті, в той час, як приватний буде доступний лише тим людям, кому ви безпосередньо надасте доступ.

- 5) Натисніть 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 *

 tshevchenko / apz

✓ 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


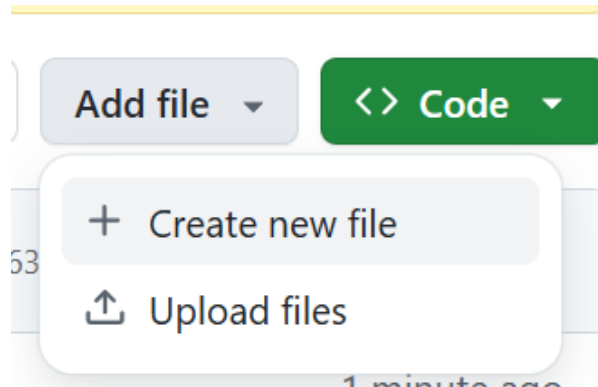
 Public

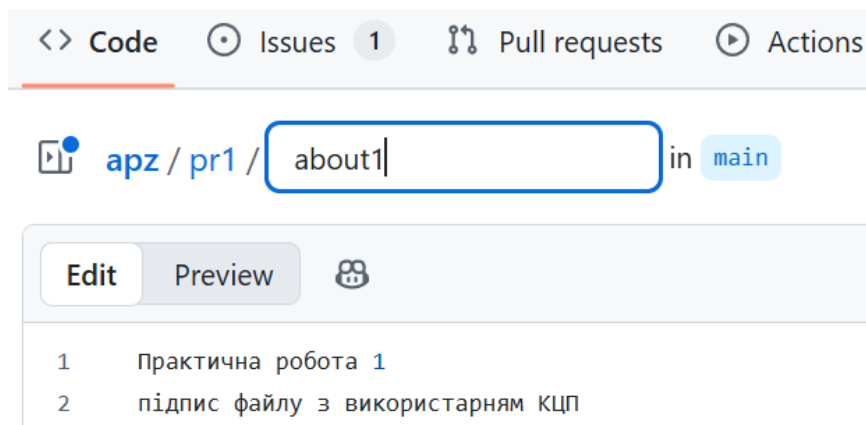
Рис. 2.2 – Створення нового публічного репозиторію на GitHub для розміщення практичних робіт

Далі для кожної лабораторної роботи зробіть окрему папку і розмістите в ній звіт для перевірки викладачем:

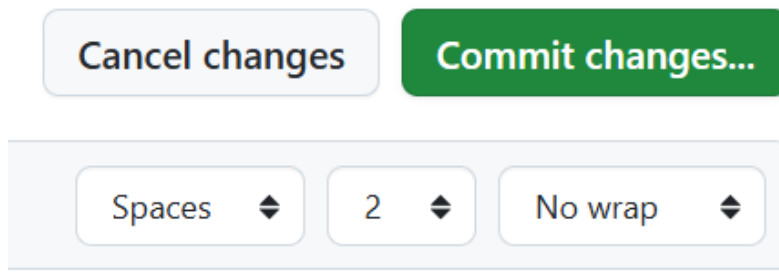
- 1) Створити новий файл, наприклад текстовий документ (рис. 2.3), який потім при необхідності можна видалити чи замінити на звіт з практичної роботи.



а)



б)



в)

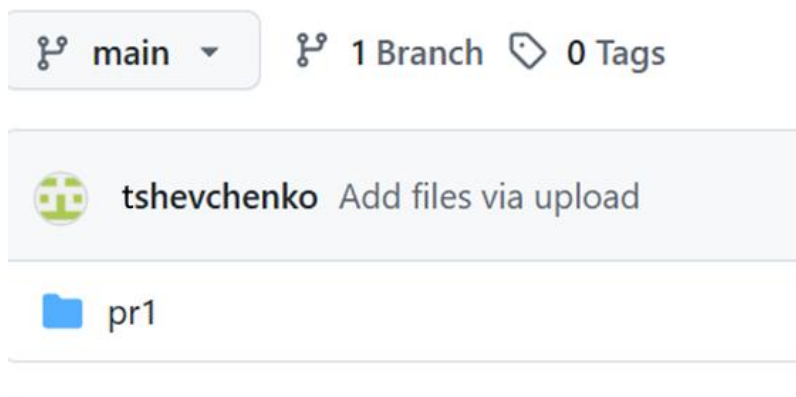


Рис. 2.3 – Послідовність створення окремих папок (а,б,в) у репозиторії для кожної практичної роботи

- 2) В кожен окрему папку додаємо файл звіту, скориставшись «**Add file**» у верхньому куті екрану.

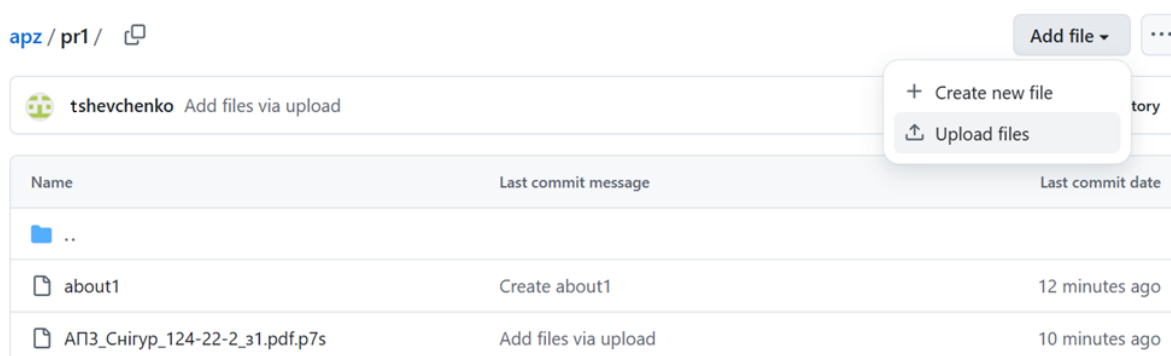


Рис. 2.4 – Додавання звітів з практичних робіт у репозиторій

Контрольні питання

1. Що таке GIT?
2. Що таке репозиторій у GIT?
3. Які переваги використання GIT?
4. Яка мова використовується в GIT?
5. Як можна створити репозиторій у Git?
6. Яка команда використовується для видалення гілки?
7. Що таке контроль версій GIT?
8. Як можна виправити несправний комміт?
9. Як ви дізнаєтесь у GIT, чи гілку вже об'єднано в master?

Практична робота №3

Тема: Написання тест-кейсів (Test Case).

Мета: Набування навичок у написанні тест-кейсів різних пристроїв.

Очікувані результати навчання: уміння підписувати особисту документацію з використанням єдиного цифрового підпису за допомогою різних сервісів і додатків.

Короткі теоретичні відомості

Тест-кейс – це професійна документація тестувальника, це послідовність дій, спрямована на перевірку будь-якого функціоналу, що описує як прийти до очікуваного результату.

Тема кейса – описова назва тесту, яка спрощує його пошук та розуміння його змісту.

У темі тест-кейсі не повинно бути залежностей від інших тест-кейсів і також нечітких формулювань та зайвої деталізації.

Навички, необхідні для написання тест-кейсів:

- *Уміння збирати та аналізувати вимоги до продукту.* Якщо немає формально описаних вимог (специфікацій) – потрібно вміти їх збирати у розробників, аналітиків, користувачів, тощо.
- *Уміння розділяти систему на складові, тобто робити декомпозицію.* Тобто, потрібно не тільки вміти бачити систему як ціле, а й уміти розкласти її на складові частини.
Це дуже корисна навичка для проведення функціонального тестування, де перевіряється кожна складова продукту.
- *Уміння розставляти пріоритети.* Тест-дизайнер повинен уміти відрізнати важливіше від менш важливого, а також розставляти пріоритети тестування.
- *Уміння формулювати свої думки (письмово та усно).* Це вміння важливе для тестувальника в принципі. Тест-дизайнеру воно дуже допомагає при створенні тест-кейсів.
- *Знання технік тест-дизайну, а також уміння застосовувати їх на практиці.*

Результатом 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.

Важливі моменти при складанні тест-кейса:

Розуміння вимоги, за якою складається кейс

Скласти й описати ефективний тест-кейс не вийде, якщо немає розуміння, яким саме чином повинен бути реалізований той чи інший функціонал програми. Ефективніше буде витратити сили та час на уточнення неточностей і в підсумку написати «робочий» тест-кейс, ніж спершу скласти абсолютно неправильний, зрозуміти в процесі тестування, що він не спрацював як треба, знову витратити час на уточнення, і тільки після цього повністю переробити та отримати робочий варіант.

Простота та легкість для розуміння

Важливо, щоб тест-кейс був описаний зрозумілими словами, без використання спеціальної термінології та складних мовних конструкцій, так як його можуть проходити фахівці різного рівня кваліфікації та спеціалізації (досвідчені тестувальники та початківці, керівники проекту, розробники, аналітики та навіть співробітники замовника, не завжди володіють достатнім рівнем знань у галузі комп'ютерних технологій).

Окрема увага приділяється наборам тестових даних

Якщо тест передбачає використання будь-яких даних, слід звернути увагу на правильність їх складання та подальші вказівки. В іншому випадку такий тест є марним

Завдання.

Придумайте об'єкт тестування. Це повинен буде будь-який об'єкт, який складається мінімум з 5 частин. (годинник, скейт, велосипед, монітор і т.п.) Напишіть не менше 20 тест кейсів до цього об'єкту, що зможуть оцінити його якість з точки зору продукту. Створіть *.pdf документ, де кратко опишіть ваш об'єкт та потім опишіть ваші тест-кейси.

Приклад Test Case: «Обчислення $1 + 1$ в калькуляторі».

Назва: Перевірка обчислення $1 + 1$.

Pre-condition: Відкрито стандартний калькулятор, система готова до введення виразів.

Кроки:

- Натиснути кнопку “1”.
- Натиснути кнопку “+”.
- Натиснути кнопку “1”.
- Натиснути кнопку “=”.

Expected Result: Відображається число 2

Post-condition: Екран калькулятора показує результат, система готова до наступного обчислення.

Контрольні питання

1. Навіщо потрібні тест-кейси?
2. Основні атрибути Test Case?
3. Типи тест-кейсів.
4. Що таке негативний тест-кейс?
5. Що повинен знати тестувальник?
6. Скільки основних принципів тестування?

Завдання

Зареєструватись в системі AWS (Amazon web services). Далі розмістити на 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-arz` для студента з іменем Михайло Єфремов. Результатом виконання цієї лабораторної роботи буде адреса на вашу веб-сторінку на подібні `myefremov-bucket-arz.amazonaws.com/index.html` на якій буде написано ваш ПІБ та група. Ось сторінка на моєму бакеті <https://omineievbuck1.s3.amazonaws.com/index.html> але з іменем неіснуючого студента Миколаєнка

Хід роботи

1. Знайомство з AWS:

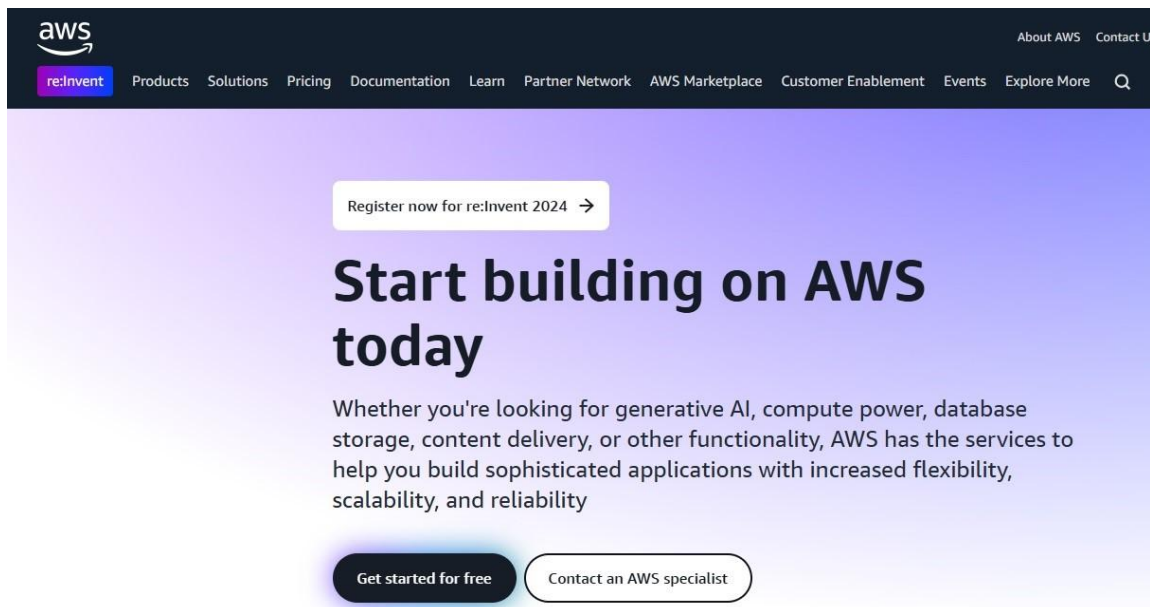


Рисунок 1 - Знайомство

2. Створення акаунту:

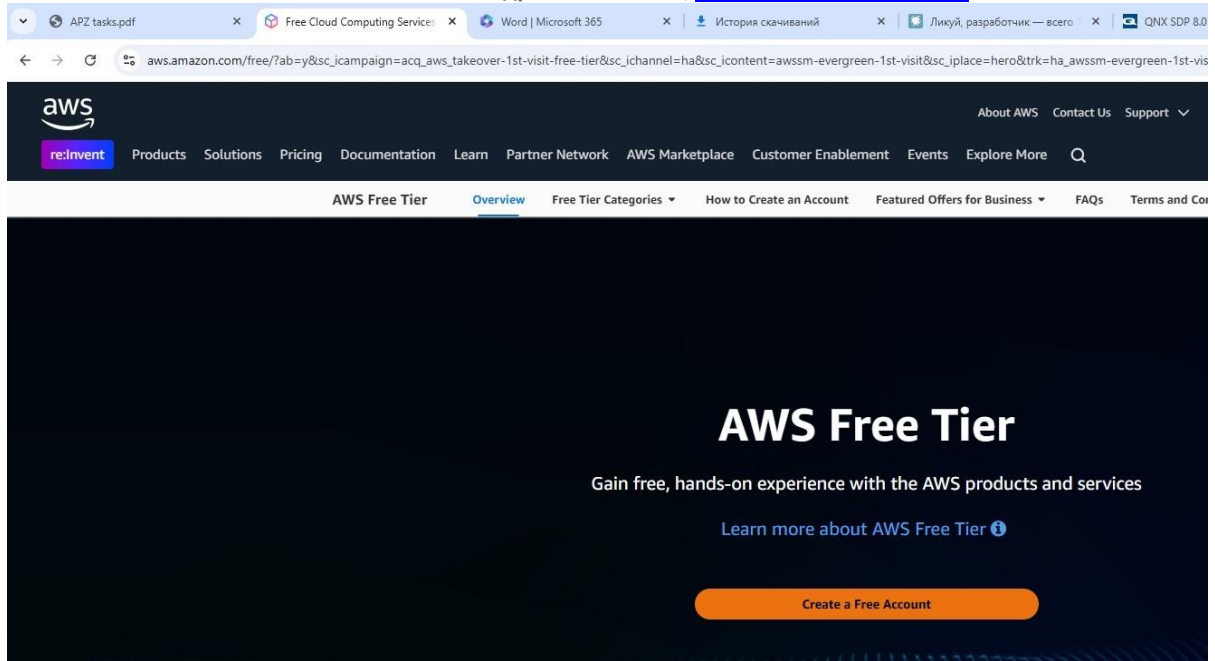


Рисунок 2 - Безкоштовний аккаунт

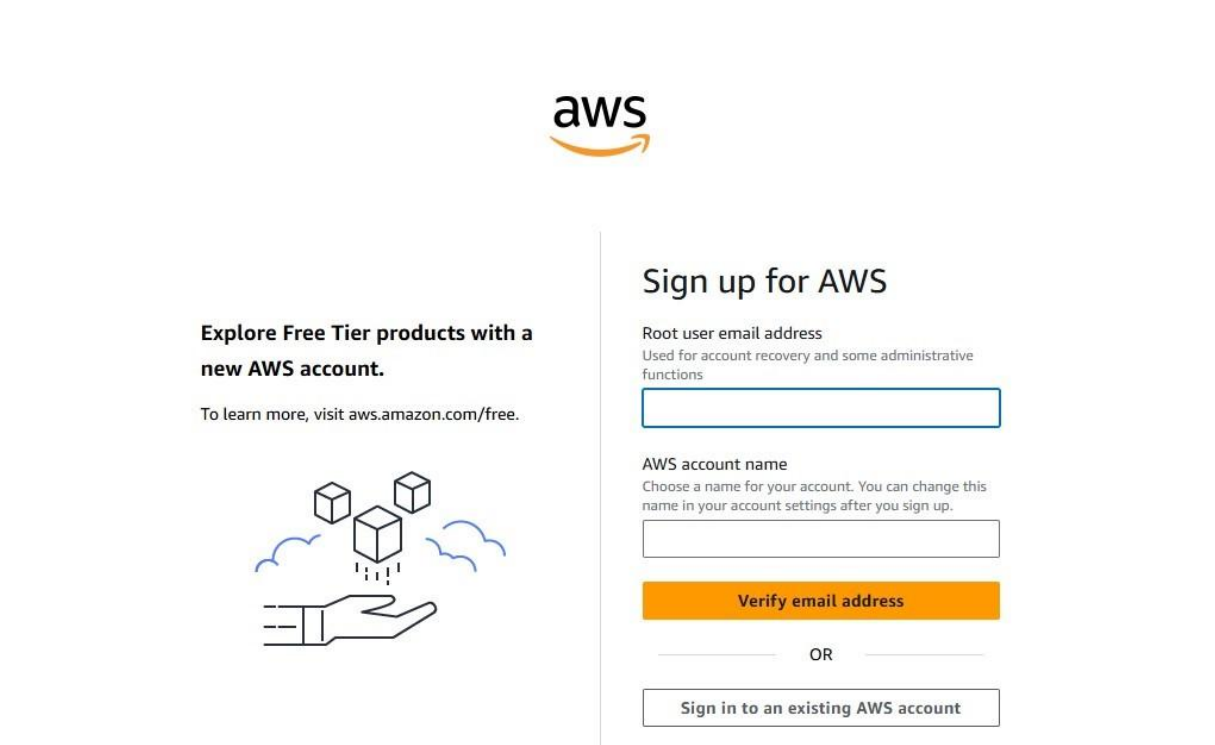


Рисунок 3 - Введення даних



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.

☒ **Basic support - Free**

- 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

☐ **Developer support - From \$29/month**

- Recommended for developers experimenting with AWS
- Email access to AWS Support during business hours
- 12 (business)-hour response times

☐ **Business support - From \$100/month**

- 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 - Узгодження ціни

3. Створення бекету:

Рисунок 5 - Створення

4. Додавання сайту (<http://kbaleiko-bucket-apz.s3-website.eu-north-1.amazonaws.com/>):

5.



Рисунок 6 - Посилання

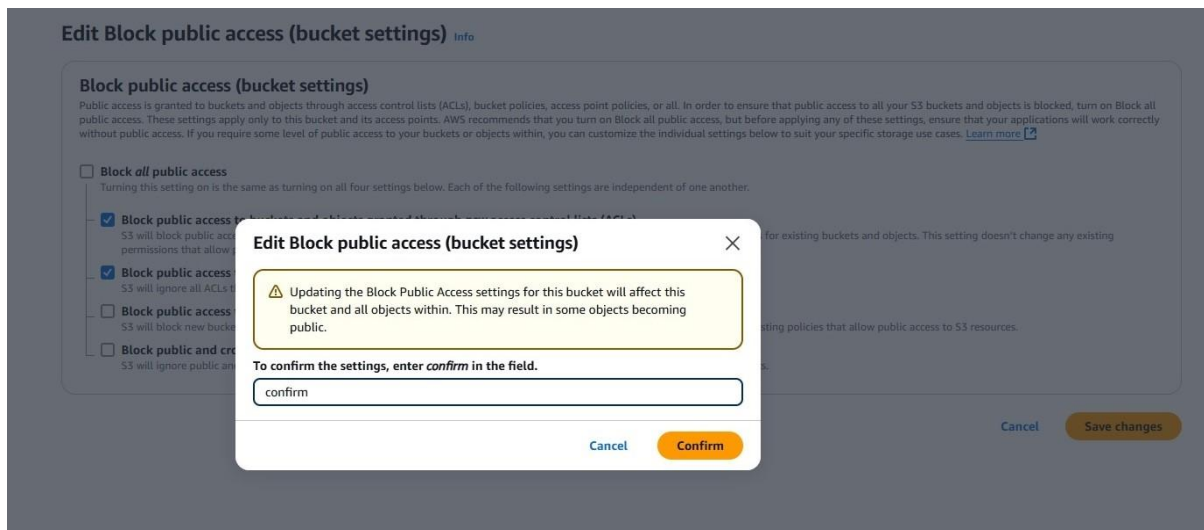


Рисунок 7 - Прибрання обмежень

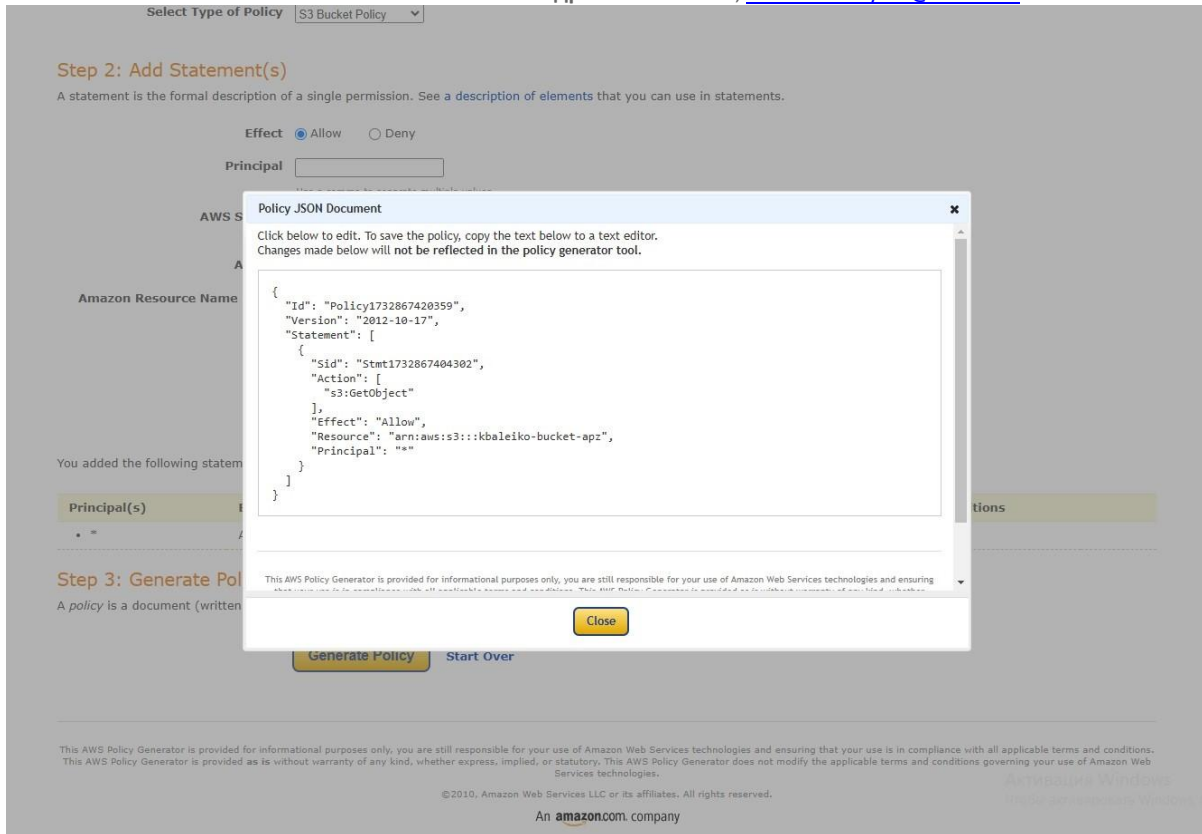


Рисунок 8 - Генерація коду

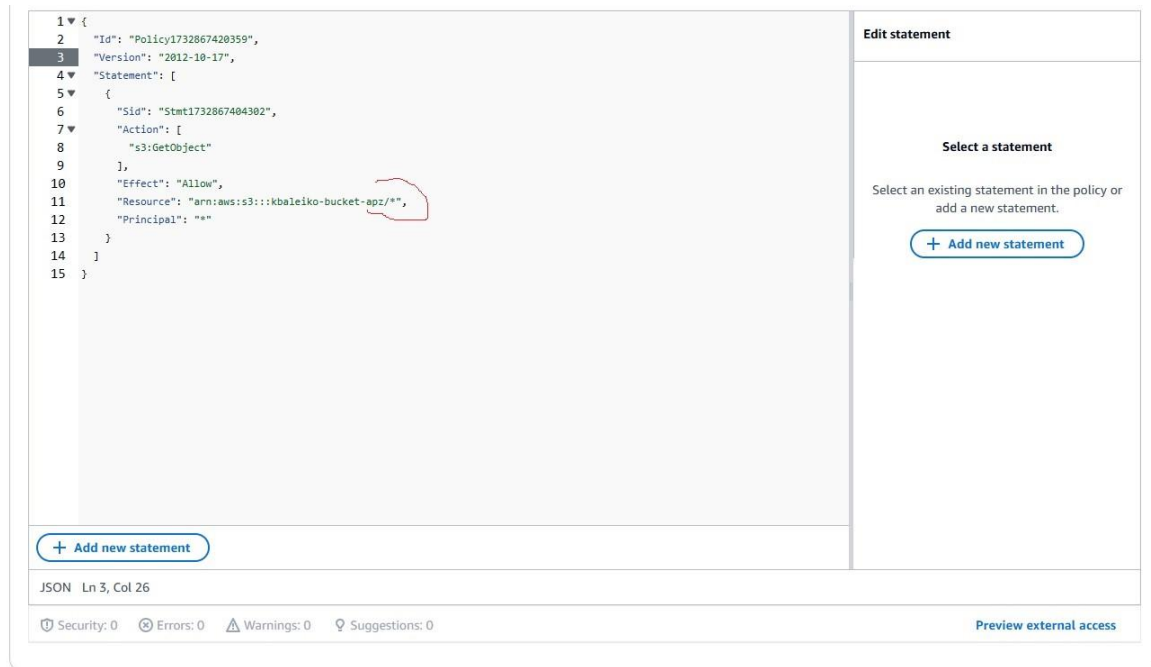


Рисунок 9 - Вільний доступ

Балейко Кирило Вячеславович, 122-21-4

Рисунок 10 - Вигляд

Висновок: ознайомився з AWS, зробив бакет та завантажив туди файл.

авдання

Вже в існуючому аккаунті AWS 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. Мій приклад: 54.91.84.196 [password:jNBM=%tQeOm3Ix46h5GT\(J5984@=aj.D.](https://www.base64encode.org/)

Хід роботи

1. Створення машини:

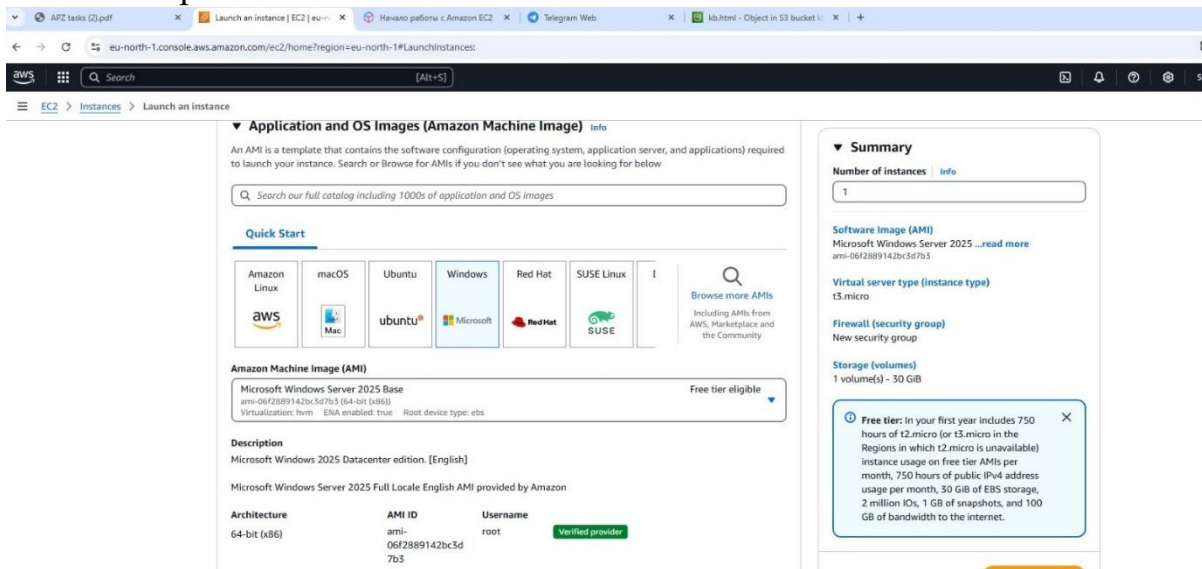


Рисунок 1 - Створення

2. Запуск (актуальні дані:

login: Administrator

password: [fbb\\$0TeUnn;&E@YWzTi*-V.jx8dUY&cT](https://www.base64encode.org/)

url: ec2-13-60-218-198.eu-north-1.compute.amazonaws.com

ip: 13.49.76.29)

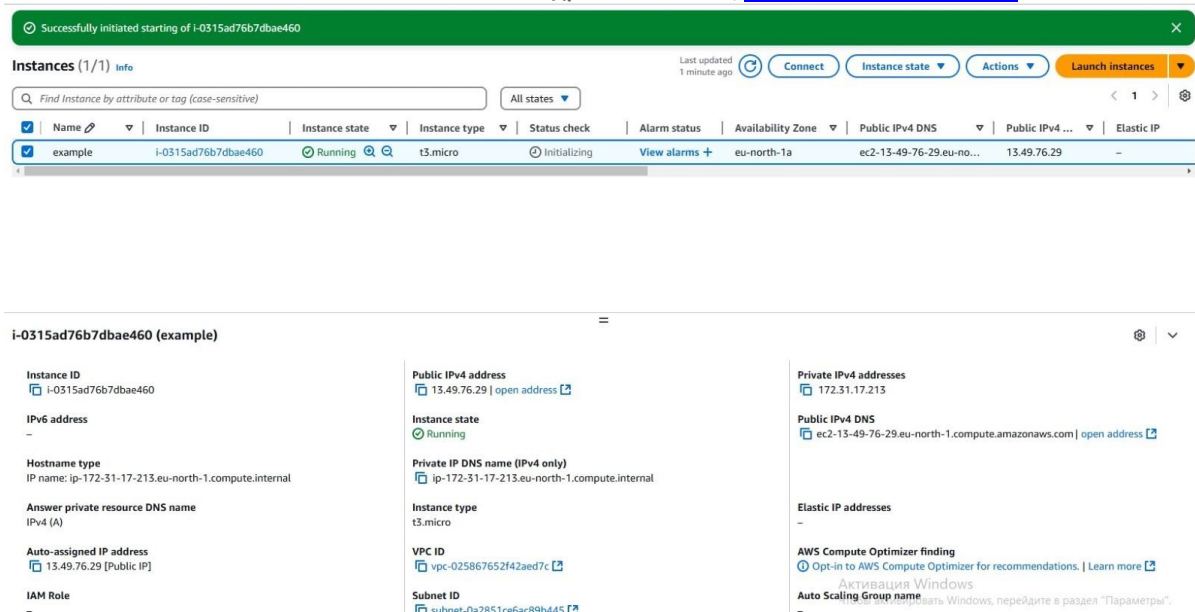


Рисунок 1 - Запуск



Балейко Кирило Вячеславович
122-21-4

Рисунок 2 - Вигляд

Висновок: створено віртуальну машину на EC2

Аналіз програмного забезпечення (АПЗ)

<https://do.nmu.org.ua/course/view.php?id=2877>

Мінеєв Олександр Сергійович, доцент, канд. техн. наук, доцент, mineev.o.s@nmu.one

Шевченко Юлія Олександрівна – асистент, shvchenko.yu.o@nmu.one

Рекомендовані джерела інформації

Базова:

1. Google. Порівняння хмарних сервісів: AWS, Azure та Google Cloud. – cloud.google.com, 2023.
2. Andreas Witting. Amazon Web Services in Action / A. Witting, M/ Witting – Shelter Island, 2019 – 200p.;
3. Сазерленд Джефф. Scrum. Навчись робити вдвічі більше за менший час. – Київ: Видавництво «Наш Формат», 2022. – 256 с.
4. Книга Agile. Як працювати по-справжньому гнучко / Нейл Перкін, Пітер Ебрахам – Київ: Видавництво «Книголав», 2021. – 368 с.
5. Мартін Роберт С. Чистий код: Створення, аналіз та оптимізація — Київ: Видавництво «Наш Формат», 2020. – 464 с.Роїк, О. М. Системний аналіз. Навчальний посібник / О. М. Роїк, А. А. Шиян, Л.О. Нікіфорова – Вінниця : ВНТУ, 2015. – 83 с.
6. Грицюк П.М. Основи теорії систем і управління: Навч. посібник /П.М. Грицюк, О.І. Джоші, О.М. Гладка. – Рівне: НУВГП, 2021. – 272 с.
7. Біловодська, О.А. Системний аналіз і прийняття інноваційних рішень : курс лекцій для студ. факультету економіки та менеджменту спец. 8.000014 "Управління інноваційною діяльністю" денної форми навчання / О.А. Біловодська, О.Ф. Грищенко. — Суми : СумДУ, 2010. — 106 с. — 9-88. Електронна версія:
<http://www.essuir.sumdu.edu.ua/retrieve/5823/Bilovodska.doc>
8. Варенко В. М. Системний аналіз інформаційних процесів: Навч. посіб. / В. М. Варенко, І. В. Братусь, В. С. Дорошенко, Ю. Б. Смольников, В.О. Юрченко. – К.: Університет «Україна», 2013. – 203с.

Додаткова:

1. Проєктний менеджмент для ІТ: Практичний підхід / Гопкінс Джоанна. – Харків: Видавництво «Фоліо», 2021. – 300 с.
2. Фоулер Мартін. Архітектура корпоративних додатків — Київ: Видавництво «Наш Формат», 2021. – 544 с.

Навчальне видання

Малієнко Андрій Вікторович
Владико Олександр Борисович
Козир Світлана Василівна
Гаранжа Дмитро Миколайович

СИСТЕМНИЙ АНАЛІЗ

Методичні рекомендації до виконання практичних робіт
для здобувачів ступеня бакалавра
зі спеціальності 124 Системний аналіз

Видано в авторській редакції.

Електронний ресурс.

Підписано до видання 11.06.2024. Авт. арк. 6,2

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