

УДК 004.4

Усик А. А. ORCID: 0009-0009-1355-5826 (ВІТІ ім. Героїв Крут)  
канд. техн. наук, доцент Симоненко О. А. ORCID: 0000-0001-8511-2017 (ВІТІ ім. Героїв Крут)  
канд. техн. наук, доцент Троцько О. О. ORCID: 0000-0001-7535-5023 (ВІТІ ім. Героїв Крут)  
канд. техн. наук, доцент Беляков Р. О. ORCID: 0000-0001-9882-3088 (ВІТІ ім. Героїв Крут)

## ОБҐРУНТУВАННЯ ДЕКЛАРАТИВНОГО ПІДХОДУ ПРИ РОЗРОБЦІ ТА УПРАВЛІННІ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ З ВИКОРИСТАННЯМ ХМАРНИХ ТЕХНОЛОГІЙ

У сучасному світі стрімкого розвитку технологій та зростаючого значення інформаційних систем для великих організацій, включаючи Збройні сили України, актуальною стає потреба в оптимізації їх розробки та управління ними. У Збройних силах України переважає принцип ієрархічності при побудові інформаційно-комунікаційних систем, що накладає обмеження на процес впровадження нових гнучких рішень. Для процесу розробки та управління інформаційними системами здебільшого використовується імперативний підхід. Разом з тим використання декларативного підходу у цих системах відкриває широкі можливості для забезпечення їх ефективності, гнучкості та простоти у розробці та управлінні ними.

У статті розглядається можливість використання принципів декларативного підходу для оптимізації розробки та управління інформаційними системами з використанням хмарних технологій, переваги та приклади його застосування.

Проблематика дослідження полягає у необхідності оптимізації процесу розробки та управління інформаційними системами з використанням хмарних технологій в умовах швидкого темпу розвитку технологій та зростаючих потреб Збройних сил України у використанні цих систем для підвищення інформаційної обізнаності та прискорення прийняття рішень командирами під час ведення бойових дій.

Метою дослідження є аналіз принципів декларативного підходу та визначення його можливостей для оптимізації розробки та керування інформаційними системами з використанням хмарних технологій. Для досягнення мети використовуються методи аналізу сучасного стану і тенденції розвитку хмарних технологій, а також аналізуються приклади реалізації декларативного підходу у різних ІТ-проєктах.

Проведений аналіз показує важливість декларативного підходу для підвищення ефективності розробки та управління інформаційними системами з використанням хмарних технологій. Результати дослідження можуть бути корисні для підрозділів, що розглядають можливість впровадження декларативного підходу у свої проєкти. Декларативний підхід може бути застосовано у новітніх розробках інформаційних систем для підвищення рівня автоматизації та оптимізації процесів управління інфраструктурою Збройних сил України.

**Ключові слова:** декларативний підхід, хмарні інформаційні системи, оптимізація, автоматизація, інфраструктура як код, Збройні сили України.

### *A. Usyk, O. Symonenko, O. Trotsko, O. Bieliakov Substantiation of the declarative approach in the development and management of information systems using cloud technologies*

*In today's world of rapid technological development and the growing importance of information systems for large organizations, including the Armed Forces of Ukraine, the need to optimize their development and management is becoming urgent. In the Armed Forces of Ukraine, the principle of hierarchy prevails in the construction of information and communication systems, which imposes restrictions on the process of implementing new flexible solutions. The process of developing and managing information systems is mostly based on the imperative approach. At the same time, the use of the declarative approach in these systems opens up wide opportunities to ensure their efficiency, flexibility and simplicity in development and management.*

*The article discusses the possibility of using the principles of the declarative approach to optimize the development and management of information systems using cloud technologies, and also the advantages and examples of its application.*

*The research is concerned with the need to optimize the process of developing and managing information systems using cloud technologies in the context of the rapid pace of technology development and the growing needs of the Armed Forces of Ukraine to use these systems to increase information awareness and accelerate decision-making by commanders during combat operations.*

*The purpose of the study is to analyze the principles of the declarative approach and determine its capabilities for optimizing the development and management of information systems using cloud technologies. To achieve this goal, the author uses methods of analyzing the current state and trends in the development of cloud technologies, as well as analyzes examples of the declarative approach in various IT projects.*

*The analysis shows the importance of the declarative approach to improve the efficiency of development and management of information systems using cloud technologies. The results of the study may be useful for departments considering implementing the declarative approach in their projects. The declarative approach can be applied in the latest developments of information systems to increase the level of automation and optimize the infrastructure management processes of the Armed Forces of Ukraine.*

**Keywords:** declarative approach, cloud information systems, optimization, automation, infrastructure as code, the Armed Forces of Ukraine.

**Постановка задачі.** Основним завданням функціонування інформаційно-комунікаційних систем є забезпечення стійкого, безперебійного управління військами та своєчасного надання інформаційних сервісів із заданою якістю та рівнем безпеки. Для ефективної діяльності розробників та посадових осіб, які імплементують розроблені рішення та здійснюють управління інформаційними системами впровадження новітніх механізмів є критично важливим.

Однак існуючі підходи щодо розробки та управління інформаційними системами є дещо застарілими, оскільки вони чітко регламентовані згідно з інструкціями. Хоча така регламентація має певні переваги, вона одночасно позбавляє систему необхідної гнучкості, не задовольняє принципам інтероперабельності та масштабованості. Імперативний підхід, який наразі переважає, створює значні обмеження при необхідності швидкої адаптації систем до змінних умов бойової обстановки та нових вимог командування.

У зв'язку з цим виникає необхідність пошуку нових підходів для гнучкого та адаптивного управління інформаційно-комунікаційними системами, зокрема їх розробкою та управлінням. Особливої актуальності набуває дослідження можливостей декларативного підходу в поєднанні з хмарними технологіями, що потенційно може забезпечити необхідний рівень гнучкості та ефективності управління військовими інформаційними системами.

### **Аналіз останніх публікацій**

В останні роки спостерігається зростаючий інтерес до використання декларативного підходу в розробці та управлінні інформаційними системами з використанням хмарних технологій. Дослідження в цьому напрямку акцентують увагу на розвитку ефективних методів та інструментів, які дозволяють спеціалістам описувати бажаний стан системи або результат, а не послідовність дій для досягнення цього стану [1]. Це підтверджується ґрунтовним дослідженням *Brown і Davis* [2], які провели всебічний аналіз декларативних підходів у контексті управління хмарними ресурсами.

Результати останніх досліджень свідчать про потенціал декларативного підходу для покращення ефективності розробки та управління інформаційними системами. Автори дослідження [3] у своєму порівняльному аналізі декларативної та імперативної парадигм демонструють значні переваги декларативного підходу в контексті масштабування та управління складними системами.

У дослідженні [4] автори розглядають гібридний підхід, тобто пропонують ефективні методи поєднання декларативних та імперативних парадигм для досягнення оптимального балансу між безпекою та продуктивністю. Існують напрацювання щодо поєднання декларативного та імперативного підходів у технологіях надання хмарних додатків [5].

У роботі [6] досліджуються сучасні підходи до створення SaaS-застосунків з використанням “cloud-native” технологій, що доповнюється дослідженням [7] щодо патернів інтеграції для декларативного та імперативного управління хмарними системами. Автори дослідження [11] представляють детальний аналіз продуктивності різних підходів у хмарних системах.

Зауважимо, що наразі фактично відсутні дослідження в питаннях хмарної декларативної інфраструктури в ЗСУ, тому що дана галузь тільки починає розвиватися, отже це може бути перспективною областю для наукових досліджень та розробки практичних рішень.

**Метою статті** є оцінка ефективності декларативного підходу відносно існуючих за розробленим показником IEDAMS, що є результатом розрахунку вагових коефіцієнтів за аналітично-визначеними критеріями.

**Виклад основного матеріалу.** Декларативний підхід зазвичай використовується в різних областях, таких як бази даних, конфігураційні файли, графічні інтерфейси користувача, обробка даних, обробка подій, інтеграція сервісів.

Прикладом застосування декларативного підходу є *GitOps*, основна концепція якого полягає в узгодженні бажаного і поточного стану системи. Таким чином, *GitOps* складається з Git-репозиторію з декларативними описами інфраструктури, які повинні привести до кінцевого бажаного стану розгорнутої інфраструктури. Як і у випадку з *Kubernetes* в *GitOps* (який сам по собі є декларативною системою), для забезпечення доступу до сервісів *Kubernetes* ззовні кластера *Kubernetes* необхідний декларативний, хмарний вхід. Цей декларативний підхід дозволяє *Kubernetes* пропонувати автоматизовані, спрощені та масштабовані додатки.

Інструмент IaC в *GitOps* автоматизує процес розгортання. Програмне забезпечення або дрейф-агенти попередять користувача у разі будь-якого відхилення від бажаного стану в поточному виробничому середовищі. Це забезпечить відповідність вимогам і, таким чином, також допоможе у впровадженні *Compliance as Code*. Ви зможете завантажити попередню версію свого середовища за допомогою контролю версій декларативної інфраструктури в *GitOps*. Декларативна інфраструктура також сприяє загальній безпеці, забезпечуючи підвищену стабільність та оптимізацію витрат.

#### **Безпека на основі політик**

Використовуючи підхід декларативної інфраструктури, можна кодифікувати власну безпеку на основі політик у вигляді *Policy as Code* (PaC). Політики безпеки, такі як протоколи автентифікації та авторизації, привілеї доступу, політики делегування тощо, можуть бути налаштовані за допомогою декларативного методу. Завдяки контролю версій також можна уникнути мануальних помилок, під час впровадження політики безпеки [17].

#### **Інструменти IaC**

При використанні декларативного підходу до розгортання додатків, необхідно лише визначити бажану версію програми та її конфігурацію у відповідному файлі специфікацій. Інструменти декларативного IaC автоматично виконують усі необхідні кроки для встановлення та налаштування додатку відповідно до заданих параметрів. Такий підхід значно спрощує роботу розробників, звільняючи їх від необхідності глибоко занурюватись у деталі процесу розгортання, як це відбувається в імперативній моделі.

Залежно від вимог організації та обраного підходу, можна обрати один з декількох інструментів IaC. Наприклад, *Chef*, якщо віддає перевагу імперативному підходу. Якщо є потреба реалізувати декларативну інфраструктуру, можна використовувати *Flux & Flagger*, *Terraform*, *Puppet* або *Cloudformation*. Такі інструменти, як *SaltStack* та *Ansible* є поєднанням обох підходів, але є переважно декларативними за своєю природою.

#### **Доцільність використання декларативного підходу у військових структурах**

Для математичного обґрунтування доцільності використання імперативного чи декларативного підходу у розробці та управлінні інформаційними системами, пропонується комплексний інструмент оцінки IEDAMS (*Index of Effectiveness of the Declarative Approach for use in Military Structures*), розроблений для кількісного визначення потенційних переваг впровадження певного підходу у військових ІТ-системах. Цей індекс враховує унікальні вимоги та обмеження військових підрозділів, забезпечуючи основу для прийняття обґрунтованих рішень щодо впровадження нових ІТ-підходів.

IEDAMS складається з восьми ключових факторів, кожен з яких відображає критичний аспект військових ІТ-операцій:

- a) Операційна ефективність (*Operational Efficiency, OE*). Є метрикою швидкості та точності виконання завдань;
  - b) Оптимізація ресурсів (*Resource Optimization, RO*). Оцінка ефективності використання обчислювальних ресурсів та інфраструктури;
  - c) Адаптивність до змін (*Adaptability to Change, AC*). Метрика, що відображає здатність системи швидко пристосовуватися до нових вимог;
  - d) Зменшення помилок та надійність системи (*Error Reduction and System Reliability, ER*). Оцінка частоти помилок та підвищення загальної надійності;
  - e) Масштабованість та гнучкість (*Scalability and Flexibility, SF*). Оцінка здатності системи розширюватися та адаптуватися до різних масштабів операцій;
  - f) Підвищення безпеки (*Security Enhancement, SE*). Оцінка захищеності від кіберзагроз та вразливостей;
  - g) Інтероперабельність (*Interoperability, IO*). Метрика взаємодії системи з іншими військовими та взаємодіючими системами;
  - h) Навчання та передача навичок (*Training and Skill Transfer, TST*). Відображає рівень складності навчання фахівців та передачі знань між різними підрозділами.
- Розрахунок показника IEDAMS пропонується здійснювати у два етапи:  
 Крок 1: Розрахунок окремих факторів.  
 Для кожного з факторів (OE, RO, AC, ER, SF, SE, IO, TST) обчислюється відносне покращення порівняно з імперативним підходом (1), а саме

$$Factor = \frac{(Mtrc\_declarative - Mtrc\_imperative)}{\max(Mtrc\_imperative, Mtrc\_declarative)}, \quad (1)$$

де *Mtrc\_declarative* – значення метрики при використанні декларативного підходу;

*Mtrc\_imperative* – значення метрики при використанні імперативного підходу.

Чисельник показує абсолютну зміну в метриці. У знаменнику використовується функція  $\max()$ , що вибирає більше з двох значень, та гарантує, що знаменник ніколи не буде нулем.

Також це нормалізує результат, обмежуючи його діапазоном від  $-1$  до  $1$ .

Крок 2: Обчислення зваженої суми факторів (2):

$$IEDAMS = w1 \times OE + w2 \times RO + w3 \times AC + w4 \times ER + w5 \times SF + w6 \times SE + w7 \times IO + w8 \times TST, \quad (2)$$

де  $w1-w8$  – вагові коефіцієнти, що відображають відносну важливість кожного фактора значення відповідних факторів, обчислені на кроці 1.

Важливо зазначити, що сума всіх вагових коефіцієнтів (3) повинна дорівнювати 1:

$$w1 + w2 + w3 + w4 + w5 + w6 + w7 + w8 = 1. \quad (3)$$

Значення IEDAMS може варіюватися від  $-1$  до  $1$ .

Позитивні значення ( $0 < IEDAMS \leq 1$ ) вказують на перевагу декларативного підходу.

Чим ближче до  $1$ , тим більша перевага.

Негативні значення ( $-1 \leq IEDAMS < 0$ ) свідчать про перевагу імперативного підходу.

Чим ближче до  $-1$ , тим більша перевага імперативного підходу.

Значення близькі до  $0$  ( $-0,2 < IEDAMS < 0,2$ ) вказують на мінімальну різницю між підходами.

Для практичного застосування IEDAMS у військових структурах рекомендується проведення контрольованих експериментів або аналіз існуючих впроваджень для збору даних по кожному фактору, проведення консультацій з військовими ІТ-експертами для встановлення

відповідних вагових коефіцієнтів. Рекомендується застосування формули для обчислення індексу на основі попередньо зібраних даних, після чого здійснюється інтерпретація отриманого значення IEDAMS та його окремих компонентів для подальшого використання результатів IEDAMS для обґрунтування рішень щодо впровадження декларативних підходів у військових ІТ-системах.

Хоча IEDAMS надає структурований підхід до оцінки ефективності декларативних методологій, важливо враховувати його обмеження. Суб'єктивність вагових коефіцієнтів може призводити до неточних результатів, саме тому необхідно проводити збір даних. Наявна складність збору даних через наявність грифу секретності. Можливість неповного охоплення всіх аспектів військових ІТ-операцій, особливо у вузькоспеціалізованих сценаріях.

```

import React from 'react';
import { Radar, RadarChart, PolarGrid, Legend, PolarAngleAxis, PolarRadiusAxis, ResponsiveContainer } from 'recharts';
const data = [
  { factor: "Операційна ефективність, declarative: 0.8, imperative: 0.6, fullMark: 1 },
  { factor: "Оптимізація ресурсів, declarative: 0.75, imperative: 0.55, fullMark: 1},
  { factor: "Адаптивність до змін, declarative: 0.85, imperative: 0.5, fullMark: 1 },
  { factor: "Зменшення помилок, declarative: 0.9, imperative: 0.7, fullMark: 1 },
  { factor: "Масштабованість та гнучкість", declarative: 0.85, imperative: 0.6, fullMark: 1},
  { factor: "Підвищення безпеки, declarative: 0.8, imperative: 0.7, fullMark: 1 },
  { factor: "Інтероперабельність, declarative: 0.75, imperative: 0.5, fullMark: 1},
  { factor: "Навчання та передача навичок", declarative: 0.7, imperative: 0.6, fullMark: 1 },
];
const IEDAMSRadarChartUkrainian= () => (
  <ResponsiveContainer width="100%" height={400}>
    <RadarChart cx="50%" cy="50%" outerRadius="80%" data={data}>
      <PolarGrid />
      <PolarAngleAxis dataKey="factor" />
      <PolarRadiusAxis angle={30} domain={[0, 1]} />
      <Radar name="Декларативний підхід" dataKey="declarative" stroke="#82ca9d" fill="#82ca9d" fillOpacity={0.6} />
      <Radar name="Імперативний підхід" dataKey="imperative" stroke="#8884d8" fill="#8884d8" fillOpacity={0.6} />
      <Legend />
    </RadarChart>
  </ResponsiveContainer>
);

export default IEDAMSRadarChartUkrainian;

```

Рис. 1. Код діаграми IEDAMS

Для полегшення інтерпретації пропонується використовувати радіальну діаграму (рисунок 2) для візуалізації багатовимірною характеру покращень, які пропонує IEDAMS. Діаграму було побудовано з використанням бібліотеки мови програмування *JavaScript* (рисунок 1), але дозволяється використовувати інші, більш зручні для окремих спеціалістів способи відображень покращень, вирахованих за допомогою IEDAMS.



Рис. 2. Радіальна діаграма: Фактори IEDAMS

### Інструменти та технології, які підтримують декларативне програмування

Декларативне програмування отримало значну популярність в різних областях програмування, існують різноманітні мови та інструменти, які підтримують цей підхід. Ось деякі з найбільш популярних з них:

SQL (*Structured Query Language*) – мова запитів, яка використовується для маніпулювання та управління базами даних. Вона дозволяє програмістам виразно описувати запити до баз даних, не вказуючи конкретних кроків для отримання результату. SQL дозволяє виражати бажаний результат, а не послідовність дій для досягнення цього стану [17];

HTML (*Hypertext Markup Language*) – мова розмітки, яка використовується для створення вебсторінок. У HTML програміст описує структуру сторінки та її вміст, не вказуючи конкретні кроки щодо розміщення елементів на сторінці. Розмітка HTML декларує структуру сторінки, а не деталі щодо відображення [18];

CSS (*Cascading Style Sheets*) – мова стилів, яка використовується для опису зовнішнього вигляду елементів веб-сторінок. Вона дозволяє програмістам декларативно визначати стилізацію елементів без прив'язки до конкретних екранів чи пристроїв. Визначення стилів CSS описує, як елементи сторінки мають виглядати, без вказівки конкретних дій;

YAML (*YAML Ain't Markup Language*) – формат серіалізації даних, який часто використовується для конфігураційних файлів та обміну даними між програмами. Синтаксис YAML декларативно описує дані та їх структуру, дозволяючи програмістам чітко виразити бажаний стан системи у текстовому вигляді;

*Ansible* – інструмент автоматизації, який використовує декларативний підхід до опису конфігурації систем та процесів розгортання. Він дозволяє програмістам описувати бажаний стан системи та автоматизувати процеси управління інфраструктурою без необхідності вказувати конкретні кроки для досягнення цього стану.

### **Приклади практичного використання декларативного підходу в різних сферах розробки програмного забезпечення**

**Хмарні інформаційні системи** – це інформаційні системи (ІС), тобто сукупність технологій та сервісів, які базуються на концепції хмарного обчислення і використовують віртуалізацію ресурсів для надання послуг через Інтернет.

Важливою рисою хмарних ІС є їхній потужний масштабований потенціал. Це означає, що вони можуть забезпечити велику обчислювальну потужність за потреби та автоматично зменшити її, коли навантаження зменшується. Такий підхід дозволяє ефективно використовувати ресурси, забезпечуючи при цьому високу продуктивність. Підтримувальна можливість автоматичного резервного копіювання та відновлення даних забезпечує безпеку та надійність інформації.

Мультиплатформеність є важливою характеристикою хмарних ІС. Їх можна використовувати на будь-яких пристроях – комп'ютерах, смартфонах, планшетах тощо.

Також хмарні ІС забезпечують високий рівень захисту даних за допомогою шифрування, ідентифікації користувачів та інших методів безпеки [9].

Хмарні інформаційні системи відкривають нові можливості для розробки, розгортання та управління програмним забезпеченням, сприяючи зниженню витрат та підвищенню продуктивності діяльності. Використання хмарних технологій у сучасних інформаційних системах створює низку переваг та викликів.

Переваги полягають у можливості легко масштабувати ресурси залежно від потреб, що дозволяє підрозділам швидко адаптуватися до змін обсягу роботи. Крім того, використання хмарних технологій дозволяє знизити витрати на інфраструктуру та обслуговування, оскільки підписка на хмарні послуги може бути економічно вигіднішою, ніж закупівля власних серверів та програмного забезпечення. Ще однією перевагою є підвищена доступність та відмовостійкість завдяки географічно розподіленим центрам обробки даних провайдерів хмарних сервісів.

Глобальний доступ до даних та можливість працювати з ними з будь-якого місця та пристрою, що підключений до Інтернету, також є значною перевагою хмарних технологій. Автоматизація багатьох процесів та можливість автоматичного резервного копіювання даних також допомагають забезпечити ефективну роботу та безпеку.

Проте, при використанні хмарних технологій виникає ряд викликів. Одним з найбільших є забезпечення безпеки даних, що вимагає відповідних заходів, шифрування, управління доступом та дотримання нормативних вимог. Залежність від стабільного Інтернет-з'єднання також може становити проблему, хоча сучасні провайдери пропонують різні моделі підключення, включаючи приватні та гібридні хмари. Залежність від одного хмарного провайдера (*vendor lock-in*) через складність міграції даних, є ще одним викликом. Для ефективного використання хмарних технологій важливо враховувати ці виклики та знаходити шляхи їх вирішення.

Для максимального використання переваг хмарних технологій та подолання згаданих викликів потрібен ефективний та гнучкий підхід до розробки та управління інформаційними системами з використанням хмарних технологій, в чому полягає ключова роль декларативного підходу.

**Поєднання декларативного підходу з хмарними технологіями** дозволяє створювати і управляти інфраструктурою та ресурсами в хмарному середовищі за допомогою декларативних мов програмування або інтерфейсів. Наприклад, вже вищезгадані інструменти інфраструктурного кодування, такі як *Terraform* або *Ansible*, дозволяють описувати потрібний стан інфраструктури у декларативному стилі.

Такий підхід спрощує розгортання та управління хмарними ресурсами, забезпечуючи швидку реакцію на зміни та підтримуючи високий рівень автоматизації та стабільності системи.

**Аналіз можливостей використання декларативного підходу для розробки та управління інформаційними системами з використанням хмарних технологій** показав, що інтеграція декларативного підходу в розробку хмарних ІС відкриває широкі перспективи для оптимізації процесів розгортання, управління та масштабування хмарних середовищ.

Перш за все, декларативний підхід дозволяє визначити бажаний стан системи або результат, а не послідовність дій для досягнення цього стану. Це дозволяє розробникам та адміністраторам хмарних систем концентруватися на описі потрібного результату, що спрощує процес управління ресурсами та конфігурацією середовища.

Декларативні мови програмування та інструменти, такі як *Terraform* або *Ansible*, надають зручний спосіб опису бажаного стану інфраструктури хмарних середовищ. Завдяки цьому, можливе автоматизоване розгортання, конфігурація та управління інфраструктурою за допомогою скриптів або конфігураційних файлів.

Ще однією перевагою декларативного підходу є його гнучкість та масштабованість. За допомогою декларативних інструментів можна легко змінювати конфігурацію та розмір інфраструктури залежно від потреб проєкту, що робить його ідеальним для хмарних середовищ, де потрібна висока гнучкість та адаптивність.

Однак важливо враховувати виклики, пов'язані з використанням декларативного підходу до інформаційних систем з використанням хмарних технологій. Наприклад, необхідно правильно структурувати конфігураційні файли та скрипти, щоб уникнути конфліктів та непередбачуваних наслідків при автоматизованому управлінні інфраструктурою.

Використання декларативного підходу в розробці та управлінні інформаційними системами, з використанням хмарних технологій, відкриває нові можливості для оптимізації процесів, підвищення ефективності та забезпечення швидкого реагування на зміни в середовищі (рисунок 3).

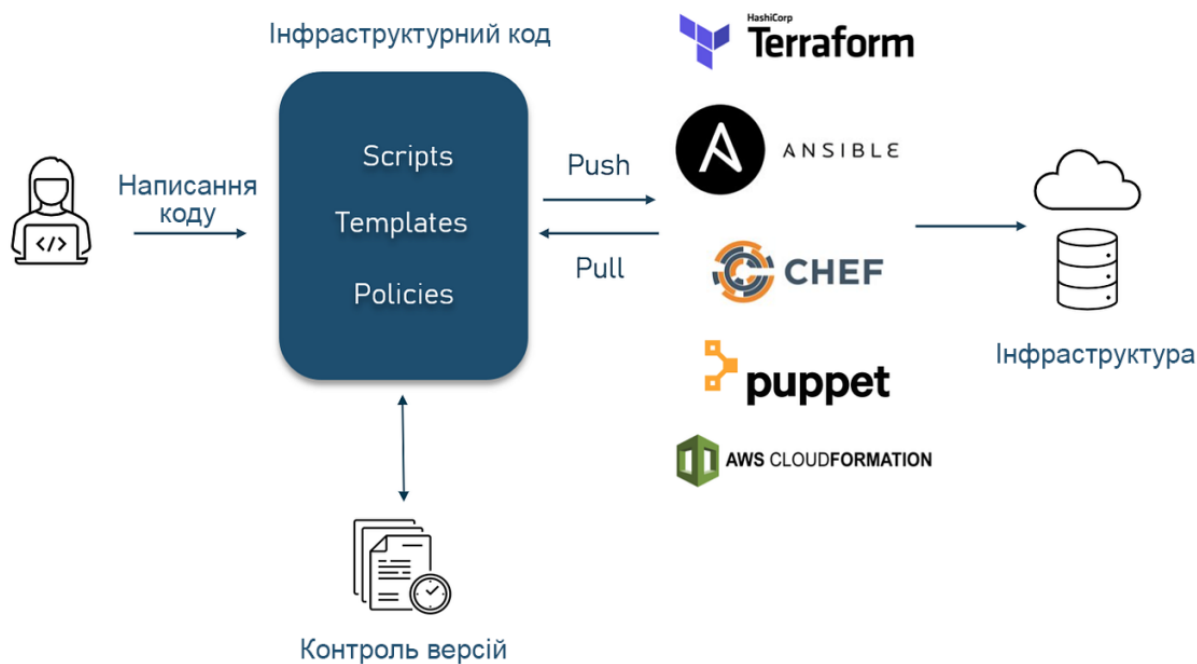


Рис. 3. Загальна схема управління хмарною інфраструктурою з використанням декларативного підходу



Впровадження декларативного підходу в хмарному середовищі відкриває нові можливості, але також має певні виклики, які варто розглянути. У деяких підрозділах Збройних сил України вже локально застосовують цей підхід у своїй інфраструктурі.

Одна з основних переваг декларативного підходу полягає у спрощенні конфігурації систем, оскільки він дозволяє описувати бажаний стан, не вдаючись у деталі процесу його досягнення. Це робить процес налаштування більш зрозумілим та інтуїтивним для розробників та адміністраторів.

Попри можливості, які пропонує декларативний підхід, існують декілька аспектів, які важливо враховувати. Наприклад, команді, що працює над проектом, може бути складно опанувати декларативні мови програмування та відповідні інструменти. Ця складність може стати перешкодою для швидкого впровадження та ефективного використання декларативного підходу.

Незважаючи на ці виклики, декларативний підхід може забезпечити значні переваги, такі як простота конфігурації, гнучкість та автоматизація процесів у інформаційних систем з використанням хмарних технологій. Тому важливо ретельно зважити всі фактори та обрати найбільш відповідний підхід для конкретних потреб та умов проєкту (таблиця 1).

Таблиця 1

## Порівняння декларативного та імперативного підходів

Критерій	Декларативний підхід	Імперативний підхід
Масштабованість	Висока. Легко адаптується до зростання системи	Обмежена. Складніше масштабувати без значних змін
Підтримка	Простіша. Код більш читабельний і легший для розуміння	Складніша. Може вимагати детального знання всіх процесів
Автоматизація	Відмінна. Легко інтегрується з CI/CD та іншими автоматизованими процесами	Обмежена. Часто вимагає додаткових інструментів для автоматизації
Узгодженість	Висока. Забезпечує єдиний підхід до конфігурації різних систем	Низька. Може призвести до розбіжностей між системами
Безпека	Покращена. Легше впроваджувати та контролювати політики безпеки	Варіативна. Залежить від ретельності реалізації кожного компонента
Гнучкість	Висока. Легко адаптується до змін вимог	Обмежена. Зміни можуть вимагати значного переписування коду
Ефективність використання ресурсів	Оптимізована. Автоматичне управління ресурсами	Залежить від навичок розробника. Може бути неефективною
Швидкість розробки	Висока. Менше коду для досягнення тієї ж функціональності	Нижча. Вимагає написання детальних інструкцій
Контроль низького рівня	Обмежений. Може бути недостатнім для специфічних оптимізацій	Високий. Повний контроль над кожною операцією.
Придатність до застосування в сучасних умовах обстановки	Висока для більшості сучасних систем. Ідеально підходить для управління складними, розподіленими системами	Обмежена. Підходить для специфічних, критично важливих систем реального часу

**Приклади успішного використання декларативного підходу в хмарних проєктах**

Компанії, які використовують інфраструктуру як код для автоматизації створення та управління хмарною інфраструктурою, часто використовують декларативні мови, такі як *Terraform* або *AWS CloudFormation*. Це дозволяє їм описати бажаний стан інфраструктури та автоматично реалізувати його на основі цього опису.

Проекти, які використовують контейнери і оркестратори, такі як *Kubernetes*, часто використовують декларативні конфігураційні файли (наприклад, YAML-файли для *Kubernetes*).

Широке використання *Terraform*, *Puppet* та *Ansible* на світовому ринку робить цілком очевидним, наскільки ефективною та надійною є декларативна інфраструктура, такою ж ефективною вона може стати в перспективі з використанням у Збройних силах України [15]. Просто визначивши специфікації для інфраструктури, можна розгорнути середовища за допомогою інструментів автоматизації IaC, що підвищить стабільність і продуктивність конвеєра розгортання.

**Висновок.** У даній статті проведено аналіз існуючих підходів, зокрема декларативного. та визначено його переваги, описано важливість використання декларативного підходу в розробці та управлінні інформаційними системами з використанням хмарних технологій. Використання декларативного програмування надає можливість опису бажаного стану системи або результату, а не послідовності дій для досягнення цього стану. Це сприяє зменшенню складності розробки, полегшує розуміння та роботу з кодом, а також сприяє автоматизації та оптимізації процесів управління інфраструктурою.

Цей підхід дозволяє програмістам описувати бажаний результат або стан системи, не вказуючи конкретних кроків для його досягнення. Натомість, програміст визначає, що потрібно зробити, а система самостійно вирішує, як це зробити.

У статті запропоновано інструмент оцінки – показник IEDAMS, який використаний для оцінки потенційних переваг застосування декларативного підходу у військових IT-структурах відносно імперативного.

Результат оцінки за аналітично-визначеними критеріями показав, що застосування декларативного підходу забезпечує підвищення ефективності розробки та управління інформаційно-комунікаційних систем до 25%.

У сфері розробки програмного забезпечення декларативний підхід застосовується у різних областях. Наприклад, мови запитів до баз даних, такі як SQL, дозволяють виразити бажані результати операцій з даними без необхідності вказувати точні кроки для їх виконання. У веброзробці розмітка HTML декларує структуру сторінки, а CSS визначає її вигляд, без вказання подробиць реалізації.

Підходи декларативного програмування знаходять широке застосування в інформаційних системах з використанням хмарних технологій. Вони дозволяють описувати потреби та вимоги до інфраструктури, конфігурації, а також процесів розгортання та управління ресурсами хмарного середовища без прив'язки до конкретних платформ чи технологій.

**Подальші дослідження будуть спрямовані на розробку методів та моделей для автоматизованого аналізу, валідації та тестування декларативних описів інфраструктури ІКС ЗСУ різних ланок управління з використанням хмарних технологій.**

#### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Christian Endres, Uwe Breitenbücher, Michael Falkenthal, Oliver Kopp, Frank Leymann, Johannes Wettinger. Declarative vs. Imperative: Two Modeling Patterns for the Automated Deployment of Applications. *The 9th International Conference on Pervasive Patterns and Applications (PATTERNS)*. 2017. URL: <https://www.iaas.uni-stuttgart.de/publications/INPROC-2017-12-Declarative-vs-Imperative-Modeling-Patterns.pdf>.
2. Twain Taylor. What does declarative mean in a cloud-native world. *Amazic*: вебсайт. URL: <https://amazic.com/what-does-declarative-mean-in-a-cloud-native-world/>.
3. Brown, M., & Davis, P. (2023). *Declarative Approaches for Cloud Resource Management*. *Journal of Systems and Software*, 185, 111-122. DOI: 10.1016/j.jss.2023.xxx.

4. Michael Wurster, Uwe Breitenbucher<sup>1</sup>, Antonio Brogi, Lukas Harzenetter<sup>1</sup>, Frank Leymann, and Jacopo Soldani A declarative approach for service enablement on hybrid cloud orchestration engines. *NOMS 2018 – 2018 IEEE/IFIP Network Operations and Management Symposium*. URL: <https://ieeexplore.ieee.org/document/8406175>.
5. Розробка технологічної стратегії SAAS рішень із використанням CLOUD-NATIVE технологій в інформаційній системі онлайн. URL: <https://archive.liga.science/index.php/universum/article/view/498/504>.
6. A review on declarative approaches for constrained clustering, Thi-Bich-Hanh Dao, Christel Vrain August 2024. URL: <https://www.sciencedirect.com/science/article/pii/S0888613X24000227?via%3Dihub#se0320>.
7. Uwe Breitenbücher, Tobias Binz, Oliver Kopp, Frank Leymann, Johannes Wettinger A Modelling Concept to Integrate Declarative and Imperative Cloud Application Provisioning Technologies. *The 5th International Conference on Cloud Computing and Services Science CLOSER*. 2015. № 1. Pp. 487–496. URL: <https://www.iaas.uni-stuttgart.de/publications/INPROC-2015-55-A-Modelling-Concept-to-Integrate-Declarative-and-Imperative-Cloud-Application-Provisioning-Technologies.pdf>.
8. Michael Wurster, Uwe Breitenbücher, Antonio Brogi, Lukas Harzenetter, Frank Leymann, Jacopo Soldani. Technology-Agnostic Declarative Deployment Automation of Cloud Applications. *8th IFIP WG 2.14 European Conference, ESOC*. 2020. URL: [https://link.springer.com/chapter/10.1007/978-3-030-44769-4\\_8](https://link.springer.com/chapter/10.1007/978-3-030-44769-4_8).
9. Achilleos, A. P., Kritikos, K., Rossini, A. et al. The cloud application modelling and execution language. *Journal of Cloud Computing: Advances, Systems and Applications*. 2019. № 8. URL: <https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-019-0138-7>.
10. Belmont J. M. Hands-On Continuous Integration and Delivery. 1st edn. Packt Publishing, Birmingham, 2018. URL: <https://www.packtpub.com/product/hands-on-continuous-integration-and-delivery>.
11. Wettinger J., Andrikopoulos V., Leymann F., Strauch S. Middleware-oriented deployment automation for cloud applications. *IEEE Transactions on Cloud Computing*. 2018. № 6 (4). Pp. 1054–1066. URL: [https://www.researchgate.net/publication/296481802\\_Middleware-Oriented\\_Deployment\\_Automation\\_for\\_Cloud\\_Applications](https://www.researchgate.net/publication/296481802_Middleware-Oriented_Deployment_Automation_for_Cloud_Applications).
12. Bergmayr, A., Bruneliere, H., Cabot, J., Hinchey, M., Langer, P., Mayerhofer, T., & Wimmer, M. Benefits of declarative deployment models in DevOps for cloud applications. *Journal of Systems and Software*. 2021. № 175.
13. Rahman, A. A., Mahdavi-Hezaveh, R., & Williams, L. A systematic mapping study of infrastructure as code research. *Information and Software Technology*. 2019. № 108. Pp. 65–77.
14. Cito, J., Leitner, P., Gall, H. C., Vřaldćuk, A., Toffetti, G., & von Lerchunderen, R. Towards declarative, multi-cloud deployment models using TOSCA. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW)*. 2017. Pp. 1–4.
15. Morris, K. Infrastructure as code: Managing servers in the cloud. *O'Reilly Media*. 2018.
16. Guerriero, M., Garriga, M., Tamburri, D. A., & Palomba, F. Adoption, use and impact of infrastructure as code: A case study. In *2019 IEEE/ACM 13th International Workshop on Software Engineering for Science (SE4Science)*. 2019. Pp. 1–6.
17. Sharma, S., Coyne, B., Cojocar, G. S., Smyth, B., & Enomoto, K. Understanding developers' perception of declarative infrastructure code: A focus group study. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2020. Pp. 672–682.
18. Smaldone, S., Brown, A. P., Laws, S., Militello, C., & Farhy, C. Taming the cloud through policy as code. *IEEE Software*. 2021. № 38 (1). Pp. 59–67.
19. Ramakrishnan, R., & Gehrke, J. Database management systems. *McGraw-Hill Higher Education*. 2003.
20. Duckett, J. HTML and CSS: design and build websites. *John Wiley & Sons*. 2011.