

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-48-18>

УДК 004.738+004.43

Хомутник Дмитро Юрійович, бакалавр,

<https://orcid.org/0000-0002-3130-8839>

Марченко Олександр Іванович, к.т.н., доцент,

<https://orcid.org/0000-0002-4537-3420>

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

ВИСОКОРІВНЕВИЙ СПОСІБ ОПИСУ РЕСУРСІВ ХМАРНОЇ ІНФРАСТРУКТУРИ

Хомутник Д.Ю., Марченко О.І. Високорівневий спосіб опису ресурсів хмарної інфраструктури. У даній статті запропонований спосіб, який дозволяє стисло описувати хмарну інфраструктуру провайдера хмарних ресурсів AWS. Розглянутий спосіб базується на ідеї спрощення опису найпоширеніших видів хмарних ресурсів шляхом використання абстракцій високого рівня. Результатом роботи способу є згенерований код мовою Terraform, у якому використовуються абстракції найнижчого рівня.

Ключові слова: хмарна інфраструктура, опис хмарних сервісів, автоматизація, хмарні ресурси, комп'ютерні мережі, AWS, Terraform.

Khomutnyk D., Marchenko O. High-level technique for description of cloud infrastructure resources. This article proposes a technique that allows us to concisely describe a cloud infrastructure of the cloud resources provider AWS. The considered technique is based on the idea of simplified description of the most common cloud resources through using high-level abstraction. The result of the technique is the generated code in Terraform language that uses low-level abstractions.

Keywords: cloud infrastructure, cloud service description, automatization, cloud resources, computer networks, AWS, Terraform.

Постановка проблеми. Використання хмарних технологій набуло значної популярності за останні роки, і з часом ця популярність лише зростає. Все більше організацій з комп'ютерним парком різного розміру переносять свою комп'ютерно-мережеву інфраструктуру з локальних серверів на хмарні ресурси, що надаються постачальниками (провайдерами) хмарних послуг (рис. 1) [1]. На цьому рисунку по вертикальній осі позначений відсоток компаній у країнах ЄС, які користуються послугами хмарних провайдерів, по горизонтальній – рік проведення статистичних замірів. З набуттям все більшої популярності таких послуг у користувачів починають формуватися конкретні очікування від можливостей засобів опису комп'ютерно-мережевих інфраструктур, що надаються провайдерами, наприклад: можливість створення віртуальних машин у підмережах з публічними або приватними IP-адресами, можливість створення систем керування реляційними базами даних з декількома копіями для збільшення завадостійкості тощо. Зі зростаючою популярністю хмарних технологій постає потреба в реалізації зручних та простих способів опису створення та використання таких ресурсів, які були б орієнтовані на менш досвідчених користувачів хмарних послуг.

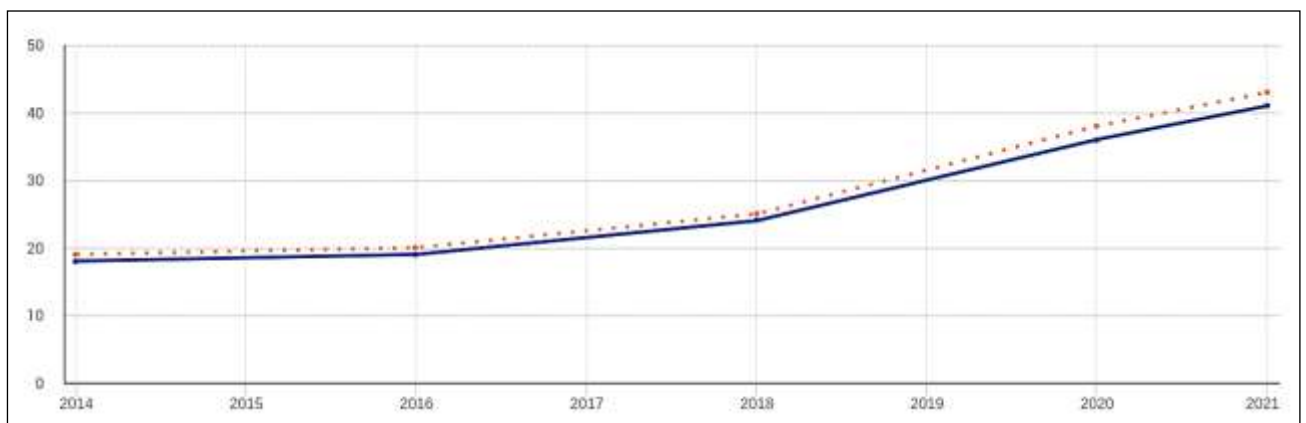


Рис. 1. Відсоток компаній у країнах ЄС, які використовують хмарні технології.

Аналіз останніх досліджень і публікацій. У роботі [2] описано загальне визначення хмарних сервісів та стандартні сервісні моделі — програмне забезпечення як сервіс, платформа як сервіс та інфраструктура як сервіс. Хоч це і стандартизує принципи моделей розповсюдження

хмарних ресурсів, але опис самих ресурсів залишається не визначеним. У роботі [3] описані можливі причини відсутності стандартних рішень для опису ресурсів у різних провайдерів. У роботі [4] наведено проблеми стандартизації опису хмарних ресурсів та запропоновано можливий варіант створення універсального способу для опису будь-яких ресурсів будь-яких провайдерів.

Попри всі проведені дослідження, досі не існує такого способу опису хмарної інфраструктури, який би надавав можливості опису ресурсів на більш високому абстрактному рівні та який був би орієнтований на менш досвідчених користувачів хмарних послуг.

Постановка завдання. Запропонувати новий спосіб опису хмарної інфраструктури, який буде відрізнятися від існуючих більш високим абстрактним рівнем опису ресурсів хмарної інфраструктури та генерацією на основі такого опису детальних низькорівневих скриптів мовою Terraform.

Термінологія.

Хмарні ресурси — ресурси, які фізично знаходяться на серверах, підключених до мережі Інтернет.

Провайдер хмарних послуг/сервісів — компанія, що надає користувачам можливість створювати та використовувати хмарні ресурси.

ОС — операційна система.

AWS – Amazon Web Services.

VPC — Virtual Private Cloud — ресурс, що об'єднує в умовну групу інші ресурси [5].

EC2 — Elastic Compute Cloud — ресурс, що надає користувачам віртуальні машини [6].

AMI — Amazon Machine Images — образ ОС, який використовується віртуальними машинами AWS [7].

Засоби автоматизації хмарної інфраструктури. Зі зростанням популярності хмарних технологій вже були вжиті певні кроки для полегшення роботи з опису та керування хмарними ресурсами. Одним з таких кроків є створення засобів для автоматизації “розгортання” хмарних ресурсів, які можна використовувати замість ручного налаштування кожного окремого ресурсу [8]. Цей підхід називається “інфраструктура як код” [9].

На сьогодні найбільші провайдери хмарних сервісів надають власні способи та засоби для опису хмарних інфраструктур. Наприклад: CloudFormation у AWS [10], Azure Resource Manager у Microsoft Azure [11], та Deployment Manager у Google Cloud Provider [12]. Всі ці засоби надають користувачам можливість описувати хмарну інфраструктуру у декларативний спосіб. Це дозволяє забезпечити однорідність створюваної інфраструктури (кожен раз створена інфраструктура буде ідентична попередній) та уникнути більшості помилок при ручному налаштуванні ресурсів.

Однак головними їх недоліками є велика детальність властивостей ресурсів, які треба описувати, та низький рівень власне способу опису цих ресурсів. Користувачу необхідно знати усі тонкощі роботи кожного способу опису, особливості роботи з певним засобом і конкретні деталі ресурсів провайдерів для створення базової інфраструктури. Це висуває високі вимоги до кваліфікації розробників та призводить до виникнення чималої кількості дрібних помилок через необхідність включення великої кількості цих деталей до опису інфраструктури, а також вимагає багато часу на дослідження структури кожного ресурсу та пошук помилок, які можуть виникнути.

Для полегшення роботи користувачів з ресурсами, що надаються провайдерами, компанією HashiCorp було розроблено спосіб опису на основі декларативної мови Terraform [13], що надає можливість описувати ресурси різних провайдерів у модульному стилі. Ця мова вирішує проблему, коли потрібно використовувати ресурси різних провайдерів, а засоби опису у кожного провайдера різні, та надає можливість модульного підходу до опису ресурсів. Однак основні проблеми – вимога високої кваліфікації розробника та необхідність робити описи ресурсів на детальному рівні – залишаються. Тому, новий користувач має витратити багато часу для вивчення специфіки опису ресурсів кожного провайдера.

Однією з причин високих вимог до детальності опису кожного ресурсу є відсутність стандарту опису для однакових ресурсів у різних хмарних провайдерів. Хоч хмарні технології і набувають популярності, ця сфера ще відносно нова для багатьох користувачів, і тому з часом все більше користувачів формують свої власні очікування від хмарних ресурсів, наприклад: створення віртуальної машини або контейнера, у якому будуть запускатись необхідні програми; створення реляційних або нереляційних баз даних з налаштуванням резервних копій тощо. Крім того, сама структура і організація провайдерів зазнавала значних змін у процесі формування і зростання ринку хмарних технологій, що унеможливило достатню стандартизацію опису ресурсів [9, 14].

Попри це, на сьогоднішній день найбільші провайдери хмарних ресурсів надають користувачам вже досить стандартний набір цих ресурсів. І хоч користувач знає, що у нього є можливість створити умовну віртуальну машину, на якій буде виконуватись користувацький код, йому все ж необхідно вивчити велику кількість конкретних деталей опису цього ресурсу того провайдера, послугами якого він хоче скористатись. І у випадку, якщо користувач захоче скористатись послугами іншого провайдера, цей процес необхідно буде повторити знову. Відповідно, навіть зі стандартними наборами хмарних ресурсів у окремих провайдерів, перед користувачами постає проблема з надмірною деталізацією і відмінністю їхніх описів, що й висуває вимогу високої кваліфікації розробника.

Опис простої інфраструктури різними способами. В якості прикладу опису інфраструктури для порівняння різних способів опису було обрано створення VPC з однією підмережею та однією віртуальною машиною EC2. У цієї віртуальної машини буде доступ до будь-якого вхідного та вихідного трафіку, однак вона буде прихована від зовнішнього світу. Для створення публічної підмережі необхідні додаткові налаштування таблиць маршрутизації та створення окремої публічної IP-адреси для EC2, що дещо ускладнює наведений приклад.

Для більш чіткого виділення відмінностей, а також підкреслення недоліків вище зазначених способів опису хмарних ресурсів, нижче наведено лістинги створення віртуальної машини у приватній підмережі для провайдера AWS за допомогою способів опису, які надають засіб CloudFormation (рис. 2) та мова Terraform (рис. 3). На цих рисунках числами в кружечках показані відповідності між оголошеннями ресурсів та їхніми параметрами в обох способах.

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ExampleVPC: ①
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 172.22.0.0/16 ②
  ExampleSubnet: ③
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref ExampleVPC ④
      CidrBlock: 172.22.10.0/24 ⑤
      AvailabilityZone: eu-north-1a ⑥
  ExampleSecurityGroup: ⑦
    Type: AWS::EC2::SecurityGroup
    Properties:
      VpcId: !Ref ExampleVPC ⑧
      GroupName: example-security-group ⑨
      GroupDescription: 'Allow all ingress traffic'
      SecurityGroupIngress:
        - IpProtocol: -1 ⑩
          FromPort: -1
          ToPort: -1
          CidrIp: 0.0.0.0/0
  ExampleNetworkInterface: ⑪
    Type: AWS::EC2::NetworkInterface
    Properties:
      SubnetId: !Ref ExampleSubnet ⑫
      GroupSet:
        - !Ref ExampleSecurityGroup ⑬
      PrivateIpAddresses:
        - PrivateIpAddress: 172.22.10.4 ⑭
          Primary: 'true'
  ExampleEC2: ⑮
    Type: AWS::EC2::Instance
    Properties:
      ImageId: 'ami-078e13ebe3b027f1c' ⑯
      InstanceType: t3.micro ⑰
      NetworkInterfaces:
        - NetworkInterfaceId: !Ref ExampleNetworkInterface ⑱
          DeviceIndex: '0'
```

Рис. 2. Спосіб опису хмарної інфраструктури засобу CloudFormation

Як видно з наведених рисунків, при використанні цих способів опису, кожна деталь повинна описуватись явно і точно. Оголошення VPC (1) потребує визначення простору IP-адрес (2) для ресурсів, що в ньому розміщуються. Оголошення підмережі (3) потребує прив'язки до VPC (4), а також визначення підпростору IP-адрес (5) для мережевих карт, що у ній розміщені. Також необхідно вказати одну із зон доступності (6) для визначення фізичного розташування ресурсів всередині цієї підмережі.

Для надання дозволу зовнішніх підключень до віртуальної машини оголошено групу правил безпеки (7). У ній також описується зв'язок з VPC (8) та інші параметри (9): ім'я групи, її опис тощо. Після цього описано правило для вхідного трафіку з вказанням дозволених HTTP-протоколів підключення, портів та просторів IP-адрес (10).

У віртуальної машини за замовчуванням відсутня мережева карта, і, відповідно, відсутній доступ до будь-яких підключень. Для створення і налаштування мережевої карти потрібно оголосити мережевий інтерфейс (11), який необхідно розмістити у підмережі (12), вказати йому групи правил безпеки (13) та виділити конкретну IP-адресу (14).

Наостанок, для оголошення віртуальної машини (15) необхідно визначити операційну систему (AMI), яка буде на ній працювати (16), а також її фізичні характеристики (17). Також необхідно прив'язати до віртуальної машини мережевий інтерфейс (18), що був оголошений раніше.

```
resource "aws_vpc" "ExampleVPC" { ①
  cidr_block = "172.22.0.0/16" ②
}

resource "aws_subnet" "ExampleSubnet" { ③
  vpc_id      = aws_vpc.ExampleVPC.id ④
  cidr_block  = "172.22.10.0/24" ⑤
  availability_zone = "eu-north-1a" ⑥
}

resource "aws_security_group" "ExampleSecurityGroup" { ⑦
  name        = "example-security-group" ⑨
  description = "Allow all ingress traffic"
  vpc_id      = aws_vpc.ExampleVPC.id ⑧

  ingress {
    protocol = "-1"
    from_port = 0 ⑩
    to_port   = 0
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "aws_network_interface" "ExampleNetworkInterface" { ⑪
  subnet_id      = aws_subnet.ExampleSubnet.id ⑫
  private_ips    = ["172.22.10.4"] ⑭
  security_groups = [aws_security_group.ExampleSecurityGroup.id] ⑬
}

resource "aws_instance" "ExampleEC2" { ⑮
  ami           = "ami-078e13ebe3b027f1c" ⑯
  instance_type = "t3.micro" ⑰

  network_interface { ⑱
    network_interface_id = aws_network_interface.ExampleNetworkInterface.id
    device_index          = 0
  }
}
```

Рис. 3. Спосіб опису хмарної інфраструктури мовою Terraform

Розглянуті приклади опису показують, що в обох випадках присутня однакова кількість оголошених ресурсів та їхніх параметрів. Хоч спосіб опису мовою Terraform і вирішує проблему модульності опису, але високі вимоги до кваліфікації розробників, що виконують перенесення

комп'ютерно-мережевої інфраструктури у хмару, та необхідність писати достатньо багато тексту залишаються на тому ж самому рівні.

Якщо віртуальну машину потрібно налаштувати з більш розширеним переліком ресурсів, ніж базовий, як в наведеному прикладі (а це необхідно майже у всіх випадках), то кількість явно прописаних зв'язків між ресурсами і їх параметрів також буде зростати відповідно. Такий підхід з явним описом абсолютно всіх ресурсів та їх параметрів може виявитись корисним, коли користувач точно знає які ресурси йому потрібні, як їх потрібно описувати, а також має достатню кваліфікацію, щоб зробити це. Однак в більшості випадків такі детальні описи є достатньо стандартними і можна обійтись більш стислим описом інфраструктури на більш високому абстрактному рівні з автоматичною генерацією детальних частин, що скоротить сам опис і прискорить його виконання, а також знизить вимоги до рівня кваліфікації розробника.

Запропонований спосіб опису. Авторами пропонується новий спосіб опису хмарної інфраструктури, який відрізняється від існуючих більш високим та наближеним до користувача рівнем опису ресурсів хмарної інфраструктури з генерацією скриптів мовою Terraform для провайдера AWS, приклад використання якого показаний на рисунку 4. На цьому рисунку, так само, як і на рис. 2 та 3, числами в кружечках показані відповідності між оголошеннями ресурсів та їхніми параметрами.

У оголошенні VPC (1) та присвоєнні йому простору IP-адрес (2) немає суттєвих відмінностей від попередніх способів опису. Однак замість визначення конкретних зон доступності для підмереж, користувач вказує лише регіон (6), у якому будуть розміщені всі ресурси в даному VPC. Хоча, при бажанні, у користувача залишається можливість і такого самого детального опису, як у попередніх способів.

Підвищення рівня абстракції опису та спрощення цього опису полягає в тому, що, вказавши регіон розміщення ресурсу VPC, користувач може опустити вибір конкретних зон доступності при оголошенні підмереж. У випадку, коли користувач оголошує декілька підмереж всередині однієї VPC, транслятором може бути згенеровано автоматичне розподілення цих підмереж по різних зонах доступності, що, крім спрощення опису, також збільшує надійність всієї інфраструктури.

```
VPC ExampleVPC { (1)
  cidr_block = "172.22.0.0/16" (2)
  region = "eu-north-1" (6)

  Subnet ExampleSubnet { (3)

    EC2 ExampleEC2 { (15)
      (16) ami = "ami-078e13ebe3b027f1c"
      (17) instanceType = "t3.micro"

      (11) networking {
        ingress = all (10)
      }
    }
  }
}
```

Рис. 4. Запропонований спосіб опису хмарної інфраструктури.

Завдяки вкладеній структурі описаних ресурсів користувачу не потрібно посилатись на ресурси в коді щоб створити зв'язки між ними. Підмережа описана в блоці VPC (3), а віртуальна машина описана в блоці підмережі (15). Таким чином, при генерації вихідних скриптів мовою Terraform з'являється можливість створювати потрібні зв'язки між ресурсами виходячи з їхньої вкладеності, а не явно прописаних зв'язків.

Крім того, зазначимо, що при використанні запропонованого способу опису серед параметрів оголошеної підмережі відсутній простір IP-адрес, адреси з якого будуть виділятися ресурсам в цій підмережі. Транслятор зможе сам генерувати ці простори для всіх підмереж, оголошених користувачем, враховуючи їхню належність до конкретних VPC та можливі конфлікти адрес цих просторів.

Ще одне спрощення полягає в тому, що, описуючи віртуальну машину, користувач може вказувати як конкретний ідентифікатор АМІ (16), так і більш загальну назву ОС (наприклад, Linux або Windows Server) – в такому випадку транслятор згенерує коректний ідентифікатор АМІ з обраною ОС в обраному регіоні. Користувач має можливість також вказати фізичні характеристики віртуальної машини (17).

Наступне спрощення опису стосується підключення віртуальної машини до мережі Інтернет, для чого необхідно визначити відповідний блок всередині EC2 (11), у якому можна описати необхідні правила доступу (10). При запропонованому способі немає необхідності описувати конкретні деталі створення мережевих інтерфейсів та правил доступу через мережу Інтернет. Для цього користувачу надається набір загальних ключових слів, якими можна замінити детальний опис найпоширеніших правил доступу до віртуальної машини (наприклад, доступ до підключень через SSH, HTTP/HTTPS, FTP тощо).

Запропонований спосіб надає можливість ще одного підвищення рівня опису хмарних ресурсів. Оголосивши ресурс VPC і вказавши у ньому блок внутрішніх IP-адрес, у нас з'являється можливість автоматично згенерувати відповідні IP-адреси для усіх підмереж і віртуальних машин, розташованих в цьому VPC. Користувач може вказати лише конкретні IP-адреси як для ресурсів EC2, так і для їхніх підмереж, а транслятор збиратиме усю необхідну інформацію з наявного опису та буде генерувати Terraform-скрипти з оптимальними налаштуваннями.

Слід наголосити, що запропонований спосіб дозволяє користувачу робити опис також і на такому самому рівні деталізації, як і при використанні мови Terraform. В такому випадку транслятор згенерує вихідні скрипти, в яких автоматичні налаштування будуть мінімальними. Однак, в даному випадку це не так важливо, оскільки метою нового способу опису ресурсів є полегшення процесу розробки інфраструктури шляхом приховування цих деталей від розробника при використанні високорівневих, а не детальних конструкцій опису.

Висновки. Авторами було запропоновано новий спосіб опису хмарної інфраструктури, який відрізняється від існуючих більш високим та наближеним до користувача рівнем опису ресурсів хмарної інфраструктури з генерацією скриптів мовою Terraform для провайдера AWS і дозволяє спростити процес опису та знизити вимоги до кваліфікації розробника. Для прикладу було обрано базові налаштування та створення однієї віртуальної машини всередині публічної підмережі. Шляхом порівняння нового способу опису з вже існуючими на конкретному прикладі було наведено основні відмінності між цими способами, а також показана більша стислість та більш високий рівень опису запропонованого способу.

У реальних комп'ютерно-мережевих інфраструктурах для ефективного використання створених ресурсів необхідно визначити різні ресурси на детальному рівні, наприклад: публічна IP-адреса для віртуальної машини, таблиця маршрутизації та її правильне налаштування в підмережі, перетворювачі адрес та інше. Описуючи це все мовою Terraform, розробнику необхідно детально розуміти усі принципи опису та використання цих ресурсів. Використання запропонованого способу опису дозволяє велику кількість деталей опису згенерувати автоматично, що усуне всі технічні помилки, які роблять розробники при детальному описі, збереже час розробника, який він мав би витратити на детальне вивчення синтаксису і структури описаних ресурсів, а також понизити рівень кваліфікації розробника до рівня користувача.

Одним з можливих шляхів подальшого розвитку запропонованого способу може бути автоматичне визначення найкращих налаштувань інфраструктури за такими параметрами як ціна, розмір віртуальних машин, швидкість створення інфраструктури та інші. Також запропонований спосіб можна розширити на використання для більш ніж одного провайдера хмарних послуг.

Список бібліографічного опису

1. Cloud computing services [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://ec.europa.eu/eurostat/databrowser/view/isoc_cicce_use/default/line/.
2. Mell P. The NIST definition of cloud computing [Електронний ресурс] / P. Mell, T. Grance // Special Publication 800-145. – 2011. – Режим доступу до ресурсу: <https://doi.org/10.6028/nist.sp.800-145/>.
3. Cloud interchangeability - redefining expectations [Електронний ресурс] / A. Monteiro, J. S. Pinto, C. Teixeira, T. Batista // Proceedings of the 1st International Conference on Cloud Computing and Services Science. – 2011. – Режим доступу до ресурсу: <http://dx.doi.org/10.5220/0003393301800183/>.
4. Ghazouani S. A survey on Cloud Service Description [Електронний ресурс] / S. Ghazouani, Y. Slimani // Journal of Network and Computer Applications. – 2017. – Режим доступу до ресурсу: <https://doi.org/10.1016/j.jnca.2017.04.013/>.
5. Amazon Virtual Private Cloud (Amazon VPC) [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/vpc/>.

6. Amazon EC2 [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/ec2/>.
7. Amazon Machine Images (AMI) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html>.
8. Prassanna J. A review of existing cloud automation tools [Електронний ресурс] / J. Prassanna, P. R. Anjali, V. Neelanarayanan // *Asian Journal of Pharmaceutical and Clinical Research*. – 2017. – Режим доступу до ресурсу: <https://doi.org/10.22159/ajpcr.2017.v10s1.20519/>.
9. Rahman A. A systematic mapping study of infrastructure as code research [Електронний ресурс] / A. Rahman, R. Mahdavi-Hezaveh, L. Williams // *Information and Software Technology*. – 2019. – Режим доступу до ресурсу: <https://doi.org/10.1016/j.infsof.2018.12.004/>.
10. AWS CloudFormation [Електронний ресурс] – Режим доступу до ресурсу: <https://aws.amazon.com/cloudformation/>.
11. Azure Resource Manager Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview/>.
12. Google Cloud Deployment Manager [Електронний ресурс] – Режим доступу до ресурсу: <https://cloud.google.com/deployment-manager/docs/>.
13. Terraform documentation [Електронний ресурс] // Terraform – Режим доступу до ресурсу: <https://www.terraform.io/docs>.
14. Shein E. The most important cloud advances of the decade [Електронний ресурс] / Esther Shein // *TechRepublic*. – 2019. – Режим доступу до ресурсу: <https://www.techrepublic.com/article/the-most-important-cloud-advances-of-the-decade/>.

References

1. “Cloud Computing Services.” *European Commission, Eurostat*, https://ec.europa.eu/eurostat/databrowser/view/isoc_cicce_use/default/line/. Accessed 5 Sep. 2022
2. Mell, Peter, and Timothy Grance. “The NIST Definition of Cloud Computing.” 2011, <https://doi.org/10.6028/nist.sp.800-145/>. Accessed 5 Sep. 2022
3. Monteiro, André, et al. “Cloud Interchangeability - Redefining Expectations.” *Proceedings of the 1st International Conference on Cloud Computing and Services Science*, 2011, pp. 180–183., <https://doi.org/10.5220/0003393301800183/>. Accessed 7 Sep. 2022
4. Ghazouani, Souad, and Yahya Slimani. “A Survey on Cloud Service Description.” *Journal of Network and Computer Applications*, vol. 91, 2017, pp. 61–74., <https://doi.org/10.1016/j.jnca.2017.04.013/>. Accessed 5 Sep. 2022
5. “Amazon Virtual Private Cloud (Amazon VPC).” *Amazon, Amazon Web Services, Inc.*, <https://aws.amazon.com/vpc/>. Accessed 6 Sep. 2022
6. “Amazon EC2.” *Amazon, Amazon Web Services, Inc.*, <https://aws.amazon.com/ec2/>. Accessed 6 Sep. 2022
7. “Amazon Machine Images (AMI).” *Amazon, Amazon Web Services, Inc.*, <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html/>. Accessed 6 Sep. 2022
8. J. Prassanna, et al. “A Review of Existing Cloud Automation Tools.” *Asian Journal of Pharmaceutical and Clinical Research*, vol. 10, no. 13, 2017, p. 471., <https://doi.org/10.22159/ajpcr.2017.v10s1.20519/>. Accessed 7 Sep. 2022
9. Rahman, Akond, et al. “A Systematic Mapping Study of Infrastructure as Code Research.” *Information and Software Technology*, vol. 108, 2019, pp. 65–77., <https://doi.org/10.1016/j.infsof.2018.12.004/>. Accessed 7 Sep. 2022
10. “AWS CloudFormation.” *Amazon, Amazon Web Services, Inc.*, <https://aws.amazon.com/cloudformation/>. Accessed 4 Sep. 2022
11. “Azure Resource Manager Overview.” *Azure Resource Manager Microsoft Docs*, <https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview/>. Accessed 5 Sep. 2022
12. “Google Cloud Deployment Manager Documentation.” *Google Cloud Deployment Manager*, Google, <https://cloud.google.com/deployment-manager/docs/>. Accessed 5 Sep. 2022
13. “Terraform Documentation.” *Terraform, HashiCorp*, <https://www.terraform.io/docs/>. Accessed 6 Sep. 2022
14. Shein, Esther. “The Most Important Cloud Advances of the Decade.” *TechRepublic*, 27 Nov. 2019, <https://www.techrepublic.com/article/the-most-important-cloud-advances-of-the-decade/>. Accessed 7 Sep. 2022