УДК004.416

ВИКОРИСТАННЯ ХМАРНИХТЕХНОЛОГІЙ ДЛЯ КОЛАБОРАТИВНОГО ІНТЕГРОВАНОГО СЕРЕДОВИЩА РОЗРОБКИ

Линник Микола Миколайович
к.т.н.,доцентКомлеваНаталіяОлегівна
Національнийуніверситет“Одеськаполітехніка”,УКРАЇНА

Анотація: У даній роботі представлено рішення для побудови масштабованої та супроводжуваної інфраструктури для проекту колаборативного інтегрованого середовища за допомогою хмарних веб-сервісів Amazon та налаштування CI/CD за допомогою Gitlab та Terraform. Також було проведено моделювання даних для БД на основі AWS Aurora.

Вступ. Більшість сучасних програмних систем мають розвинутий функціонал, який дозволяєкористувачамефективновиконуватизавданняурізнихпредметнихобластях[1,2].Під часпланнуваннярозробкипроектуколаборативногоінтегрованогосередовищабулорозглянутоі нефункціональні вимоги. Одними з найважливіших нефункціональних вимог було поставлено масштабованість, супроводжуваність та безпеку. Для забезпечення цих вимог було вирішено використовувати хмарні сервіси та впровадити CI/CD для забезпечення автоматизованого процесу поєднання коду розробників, виконання автоматичних тестів та постійної доставки програмного забезпечення будь-якої складності [3,4,8]. Це передбачає побудову надійної інфраструктури,оптимізаціюпроцесіврозгортаннятаоновленнязастосункутавідслідковування змін в коді для забезпечення стабільної роботи програми в режимі онлайн.

Мета роботи. Метою даної роботи є впровадження ефективних методів налаштування хмарних сервісів, а також реалізація стратегії DevOps для забезпечення масштабованості та супроводжуваності системи проекту колаборативного інтегрованого середовища.

Основна частина роботи.Вендором хмарних послуг було обрано *AWS*. Попри важкість використання він забезпечує високу надійність, безпеку та масштабованість інфраструктури.

В проекті використовується *AWS Aurora* — це високопродуктивна, масштабована хмарна СУБД, яка є сумісною з *MySQL* та *PostgreSQL*. На рис. 1 зображено *ER*-діаграму сутностей, що зберігаються в БД.

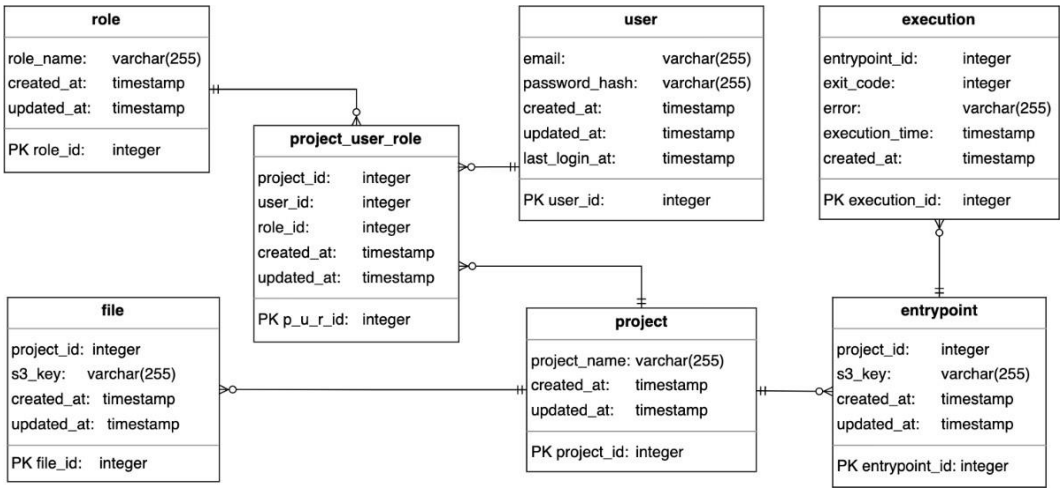

Рисунок1–ERдіаграмаданих

В проекті також використовується *Amazon S3 (Simple Storage Service)* у якості об’єктного сховища. Ця технологія є необхідною для розробленої системи — для збереження програмних проектівкористувачів,атакожїхподальшоїобробкиіншимикомпонентамисистеми.*AmazonS3*

також використовується для зберігання внутрішніх файлів проекту, наприклад, стану інфраструктури.

Не менш важливою, ніж збереження програмних проектів, є їх збірка, з трансляцією коду. Для цього, а також для виконання коду програмних проектів користувачів використовується такий хмарний сервіс як *Amazon Lambda* — хмарний сервіс безсерверних обчислень, що надається *AWS* і дозволяє користувачам запускати код заподіями (викликами чи тригерами) без необхідності піднімання та налаштування повноцінних серверів [5].

Оскількицехмарнийсервіс*AWS*,вінлегкоінтегруєтьсязіншимихмарнимисервісами*AWS*, наприклад в проекті інстанси *Amazon Lambda*звертаються до *Amazon S3*, отримуючи об'єкти — файли проектів користувачів для їх подальшої трансляції. Результат виконання коду також зберігається в окремих об'єктах *Amazon S3*.

Задля підвищення надійності та швидкості розробки, у проекті використовуються принципи *CI/CD*.Конфігурація цього процесу знаходиться у файлі *.gitlab-ci.yml* в корніпроекту [6].Віноописуєсписокдій,якіпотрібнозробитипіслякожногокоміту.Дляньогоналаштовано3 стадії.

1. *build* — у цій стадії виконуються роботи по збірці образів для подальшого розгортання. Кожен сервіс має свою налаштовану роботу, яка автоматично виконує збірку лише у випадку наявності змін у поточному коміті.
2. *deploy* — у цій стадії зібрані образи розгортаються на AWS за допомогою технології Terraform.
3. *cleanup* — у поточній реалізації для кожного сервісу видаляє всі образи крім останнього. Це потрібно для економії під час розробки, але є опціональною стадією, тому викликається власноруч.

У проекті було вирішено використовувати інфраструктуру як код (*IaC*). Для цього був обраний *Terraform* [7].

Terraform зберігає стан інфраструктури, що дозволяє йому самому виявляти які зміни в конфігурації проекту потрібно внести. Такий підхід, однак, потребує налаштування зберігання стану інфраструктури. Це робиться за рахунок *Terraform backend*, який включає в себе наступні компоненти:

1. *S3bucket*потрібнийдлязберіганнясамогостану
2. *DynamoDB* таблиця з колонкою *LockID* для запобігання конфліктів під час паралельних викликів;

В проекті *Terraform* використовується для автоматизації створення та оновлення більшої кількості інстансівсервісів,утомучислістворенняпотрібних *roles*та*policies*.Аледеякісервіси повинні бути налаштовані власноруч для налаштування власне *Terraform* та інших керівних техгологій проекту.

Проектдужепростийуналаштуванні.Списокнеобхіднихдляцьогокомпонентівскладає:

1. Створений*AWSIAM*користувачзнеобхіднимиправами
2. Створені*Gitlabvariables*,щовключаютьвсебеключікористувача,регіонрозгортаннята логін з паролем для бази даних.
3. Створений*AWSS3bucket*та*AWSDynamoDB*таблицядлязберіганнястанурозгортання *Terraform*.

Налаштувавшиусіцікомпоненти,можнапіднятиробочийпроект.

Нижчеприведенийалгоритмвзаємодіїсервісівпідчастрансляціїкодукористувача.

1. Основнийсерверзберігаєфайлзпрограмнимкодому *S3*сховищі.
2. Основнийсерверзберігаєметаданіпрофайлубазіданих.
3. Основнийсервервикликає*AWSLambda*функціюйпередаєїйшляхдопроекту,який потрібно виконати.
4. *AWSLambda*отримуєфайлйпрограмнийкодзісховища*S3*.
5. *AWSLambda*виконуєпрограмнийкодйотримуєйогорезультат.
6. *AWSLambda*зберігаєрезультатвиконаннякодудо *S3*сховища.

7. Основнийсерверотримуєрезультатвиконаннякодуз*S3*сховища.
8. Основнийсерверзберігаєметаданіпровиконаннякодудобазиданих.

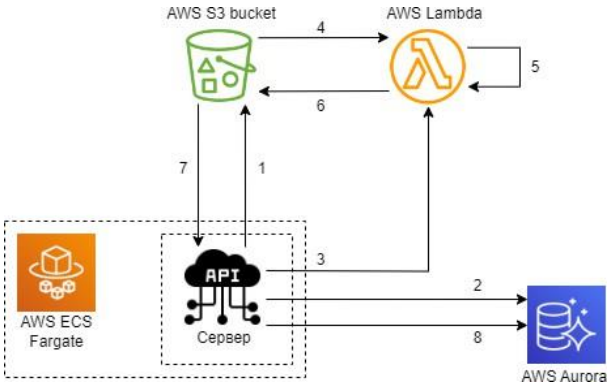

Рисунок2–Діаграмавзаємодіїсервісів

Висновок. У результаті даної наукової роботи було впроваджено ефективні методи налаштування хмарних сервісів за рахунок використання *AWS*, а також реалізовано стратегії *DevOps* для забезпечення масштабованості та супроводжуваності системи проекту колаборативногоінтегрованогосередовищазарахуноквикористання*GitlabCI/CD*та*Terraform*. Також було проведено моделювання даних для бази даних на *AWS Aurora*.

СПИСОКВИКОРИСТАНИХДЖЕРЕЛ

1. Novikova,N.O.“ChangingandTracingofSoftwareRequirementsatLevelofConceptual Classes”.*AppliedAspectsofInformationTechnology.Publ.NaukaiTekhnika*.Odessa:Ukraine.2020;Vol.3No.1: 393–404. DOI: https://doi.org/10.15276/aait.01.2020.2
2. Liubchenko, V., Komleva, N., Zinovatna, S. & Pysarenko, K. “Framework for Systematization of Data Science Methods”. *Applied Aspects of Information Technology. Publ. Nauka iTekhnika*.Odessa: Ukraine.2021; Vol.4 No.1: 80–90. DOI: https://doi.org/10.15276/aait.01.2021.7.
3. Paulin, O. N., Komleva, N. O., Marulin, S. U. & Nikolenko, A. A. “Method for Constructing the Model of Computing Process Based on Petri Net”.*Applied Aspects of Information Technology. Publ. Science i Technical*. Odessa: Ukraine.2019; Vol. 2 No.4: 260–270. DOI: https://doi.org/10.15276/aait.04.2019.1
4. KomlevaN.O.,TereshchenkoO.I.“Requirementsforthedevelopmentofsmartcontractsandanoverviewofsmart contract vulnerabilities at the Solidity code level on the Ethereum platform”. Herald of Advanced Information Technology. Publ. Nauka i Tekhnika. Odessa: Ukraine. 2023; Vol. 6 No. 1: 54–68. DOI: https://doi.org/10.15276/hait.06.2023.4
5. AWS [Electronic source] — AWS documentation — https://docs.aws.amazon.com/?nc2=h_ql_doc_do— 04.05.23
6. Gitlab[Electronicsource]—Gitlabdocumentation—https://docs.gitlab.com/—04.05.23
7. Terraform [Electronic source] — Terraform documentation — https://developer.hashicorp.com/terraform/docs— 04.05.23
8. Galchonkov O. N., Babych M. I., Plachinda A. V., Majorova A. R. “Reducing cloud infrastructure costs through taskmanagement”. Applied Aspects of Information Technology. Publ.Nauka iTekhnika. Odessa: Ukraine. 2021; Vol. 4 No. 4: 366– 376. DOI: https://doi.org/10.15276/aait.04.2021.6

USINGCLOUDTECHNOLOGIESFORACOLLABORATIVEINTEGRATED DEVELOPMENTENVIRONMENT

Lynnyk Mykola
PhD,AssociateProfessoroftthedepartmentofSENataliiaKomleva National University “Odessa Polytechnic”, UKRAINE

ANNOTATION. Thispaperpresentsasolutionforbuildingascalableandmaintainableinfrastructure for a collaborative integrated environment project using Amazon cloud web services and setting up CI/CD using Gitlab and Terraform. Data modeling forthe database based on AWS Aurora was also performed.