Савчук Олена Володимирівна, к.т.н., с.н.с.

Багнюк Юрій Іванович, студент

Київський політехнічний інститут ім.Ігоря Сікорського

**СИСТЕМА РОЗГОРТАННЯ, МОНІТОРИНГУ ТА КОНТРОЛЮ ВЕРСІЙНОСТІ ЗАСТОСУНКІВ В ХМАРНИХ СЕРВІСАХ**

Поглиблюючи тему розробки системи розгортання, моніторингу та контролю застосунків в хмарних сервісах [1], основну увагу було приділено саме контролю версійності застосунків. На відміну від моноліту, кількість мікросервісів чи лямбд може рости з часом розгортання, моніторингу і версійності безсерверних додатків у різних цільових середовищах (розробницьке,  тестове чи середовище з реальними користувачами) і бути досить значною. Тому виникла ідея створення системи, яка  інтегрувала б взаємодію з різними сервісами через зручний та інтуїтивний інтерфейс та надавала можливість керувати інфраструктурою складних проєктів.

Базою системи було вибрано AWS Lambda, розгортання якої може здійснюватися різними методами: на веб-сайті AWS Console, за допомогою AWS CLI, AWS SDK тощо. Найпростіше, але не завжди найбільш зручне у випадку великих проєктів, є розгортання за допомогою утиліти командної стрічки AWS CLI. Дещо ширші можливості надає, наприклад, утиліта Claudia [2]. За допомогою неї є можливість розгортати мікросервіси Node.js на AWS Lambda. Проте Claudia теж консольна утиліта, яка потребує певних технічних навичок.

Terraform дозволяє описати інфраструктуру. За допомогою Terraform  описується очікувана інфраструктура як код, навіть якщо вона охоплює кількох постачальників послуг. Cервери можуть бути від AWS, DNS - від CloudFlare тощо. Terraform створюватиме всі ці ресурси для всіх постачальників паралельно. Claudia утиліта підпадає під класифікацію більше як інструмент у категорії «Інструменти для мікросервісів», тоді як Terraform більше підпадає під визначення  «Інструмент для створення інфраструктури».

 Псевдонім функції схожий на вказівник. Кожен псевдонім вказує на певну версію функції, яку можна оновити за потреби. Покажчик версії є прозорим для споживачів, тому видавці можуть оновлювати функції без необхідності оновлювати запити споживачів. Подібно до версій, кожен псевдонім також має свій унікальний ARN. Псевдоніми можуть бути корисними для маршрутизації трафіку до нових версій після належного тестування. Команда розробників розгортає нову версію, перевіряє її належним чином, а потім вказує псевдонім на нову стабільну версію.

Стратегії розгортання Rolling або Canary можуть бути реалізовані за допомогою псевдонімів. Замість оновлення всіх запитів на використання нової версії Lambda одночасно, поточна та нова версії можуть співіснувати, кожна з яких отримує частину трафіку. Коли новий трафік спрямовується до нової версії, команда розробників може відстежувати та виявляти будь-які можливі проблеми з новим кодом. Коли є впевненість в нових версіях застосунків, можливе переведення 100% трафіку на нову версію та відмовлення від старої. Коли новий трафік спрямовується до нової версії, це може відстежуватися та виявлятися засобами моніторингу і логування. У випадку повної впевненості 100% трафіку може бути переведено на нову версію та реалізована відмова від старої.

На ринку достатньо велика кількість продуктів для організації безперервної інтеграції:  Jenkins, Azure DevOps, Bamboo, CircleCI, TeamCity, AWS CodeBuilds тощо. Вибрано Jenkins — програмну систему з відкритим вихідним кодом на Java, призначену для забезпечення процесу безперервної інтеграції програмного забезпечення [3].

Контроль версій (Source Version Control) є досить корисним для розробників і особливо на розробках промислового рівня. Він допомагає розробникам відстежувати зміни в коді програмного продукту і спільно керувати ними. Зі збільшенням програмного забезпечення із часом контроль версій коду стає все важливіший.

Система являє собою інтегроване рішення, яке взаємодіє з набором зовнішніх інтерфейсів : AWS API,  API серверів спільного використання коду GitHub, CI (Jenkins), API тощо [4].Основною метою даного застосунку є можливість надання в межах одного інтерфейсу доступу до управління і контролю інфраструктурою (в спрощеному виді, який би був зрозумілий і не технічним працівникам).

В програмі передбачається декілька ролей, зокрема і роль адміністратора. Додаток передбачає сторінку входу в систему та сторінку з переліком запущених сервісів, які розгорнуті в поточний момент в середовищі. Остання сторінка також містить технічну інформацію по сервісам в середовищі (версія, час останнього розгортання додатку тощо), перелік елементів управління по розгортанню сервісами (кнопки запуску розгортання тощо) і певна інформація з моніторингу сервісів (логування та інші).

Відповідно, наданий інтерфейс в одному додатку для контролю версійності застосунків, розгорнутих в хмарі, можливість розгортання останніх версій застосунків і моніторингу основних метрик цих додатків. Не знаючи інтерфейсу Jenkins, GitHub, AWS в одному агрегованому додатку можливе керування усім набором інструментів для контролю над інфраструктурою проєкту. Під версійністю мається на увазі конкретна версія коду (яка реалізує певний функціонал), яка міститься в кінцевому артефакті, і зазвичай позначається у назві артефакту. Наприклад, “my-service-1.4.5”, де “1.4.5” - версія коду, що містить певний функціонал. Зазвичай, вона збігається з версією, що вказана в файлі package.json (node.js) чи в pom.xml (java).

В системі використовуються в основному безкоштовні або умовно безкоштовні версії програмного забезпечення, бібліотек і фреймворків, що робить її дешевою у впровадженні. Jenkins вибрано як основний інструмент для розгортання, оскільки це найбільш поширене рішення в командах (серед безкоштовних, не SAAS рішень). Spring - це популярний фреймворк та IoC контейнер для платформи Java.

Для функціонування системи потрібні умови наявності встановленого деякого програмного забезпечення (terraform, бази даних тощо) та проведених налаштувань конфігурацій (зокрема Jenkins). Повинні також бути створені попередньо користувацькі акаунти і згенеровані токени (зокрема Jenkins, github токени). AWS client\_id, client\_secret мають бути сконфігуровані як зміні середовища сервера, на якому виконується додаток. Створено довгостроковий ключ доступу.

Користувачів з ролю  “Admin” варто мати в обмеженій кількості.

Основною ролю для користувачів даної системи є роль “User”.  Роль “Viewer” варто задавати користувачам, які ні в якому разі не повинні були б мати прав на запуск розгортання додатків системою. Користувачам з роллю “Viewer” доступні по суті лише можливості по перегляду інформації в системі. Можливість що-небудь редагувати у них є відсутня. Відповідно, дану роль доцільно надавати, наприклад, новим членам команди чи практикантам на проєкті.

Система має певні переваги перед конкурентними продуктами, насамперед це веб-інтерфейс. Даний продукт можна розглядати як мінімально життєздатний. Для реалізації продукту використані безкоштовні рішення, є можливість в декілька кліків у веб-інтерфейсі отримувати інформацію щодо версійності розгорнутих додатків, базові метрики і запускати розгортання; інтеграція з існуючими системами безперервної інтеграції.

Система може бути в нагоді як невеликим командам з обмеженим бюджетом, де відсутні release-інженери, та інженери, що займаються розгортанням версій додатків в середовищах, так і для більших проєктів з великою кількістю сервісів.

Література

1. Савчук О.В., Багнюк Ю.І. Система розгортання, моніторингу та контролю застосунків в хмарних сервісах/ Матеріали Міжнародної науково-практичної конференції (онлайн): Інформаційне суспільство: технологічні, економічні, та технічні аспекти становлення (вип.71),18-19 жовтня 2022, м. Тернопіль.- С.39-40.
2. Claudia vs Terraform: What are the differences? [електронний ресурс] - Режим доступу до ресурсу:  <https://stackshare.io/stackups/claudia-vs-terraform> (дата звернення: 10.12.2022)
3. Trigger Jenkins Job Remotely using Jenkins API [електронний ресурс] - Режим доступу до ресурсу: <https://narenchejara.medium.com/trigger-jenkins-job-remotely-using-jenkins-api-20973618a493> (дата звернення: 10.12.2022).
4. GitHub documentation [електронний ресурс] - Режим доступу до ресурсу:  <https://docs.github.com/en> (дата звернення: 10.12.2022)