Engineer

AWS

хмарна платформа Amazon

Що це таке

AWS, скорочено від Amazon Web Services, — це послуга Amazon, що здає в оренду обчислювальні ресурси з оплатою за години й гігабайти: віртуальні сервери, сховища файлів, керовані бази даних, приватні мережі, черги й довгий каталог інших сервісів, які вмикають через один API й оплачують з одного облікового запису. Amazon продає її з 2006 року. Два найвідоміші сервіси пояснюють чимало з решти. Amazon EC2 здає в оренду віртуальні машини, які AWS називає інстансами (instances), а Amazon S3 зберігає файли як об'єкти в контейнерах, що звуться бакетами (buckets).

У термінах визначення NIST AWS — публічна хмара: будь-хто з платіжною карткою може відкрити обліковий запис, ресурси з'являються за кілька хвилин, а кожна секунда й кожен гігабайт споживання обліковуються. Її зазвичай порівнюють з Azure від Microsoft і з Google Cloud, які продають майже ті самі будівельні блоки під іншими назвами. Те, що AWS називає інстансом EC2, в Azure є віртуальною машиною, а в Google Cloud — інстансом Compute Engine.

Кожен ресурс AWS розташований у конкретному місці. Сервіс поділено на регіони (Regions), кожен з яких спроєктовано ізольованим від інших, і в кожному регіоні є кілька зон доступності (Availability Zones). Зона — це один або кілька окремих дата-центрів із резервованим живленням, мережею та підключенням, фізично відокремлених від інших зон регіону. Ресурс створюють в одному регіоні, часто в одній зоні, а місце обирають з огляду на затримку, законодавство і ціну.

Доступ до всього цього проходить через AWS Identity and Access Management, або IAM. Обліковий запис починається з root-користувача, якому дозволено все, а IAM додає користувачів, групи, ролі та політики, що

визначають, який виклик API дозволено для якого ресурсу і за яких умов. Більшість питань про безпеку в AWS зрештою виявляються питаннями про ці політики.

Застосуйте поняття до рішення

Застосуйте «Що таке AWS?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Сервери під те навантаження, яке у вас справді є, перенесені в хмару чи назад, із порахованим наперед місячним рахунком.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Вела 20 застосунків Docker у Google Cloud і AWS — 20 контейнеризованих застосунків розгорнуто й поставлено під моніторинг

Вела 20 застосунків Docker у Google Cloud і AWS Зв'язатися з нами →

Чому це важливо

AWS обирають за широту. Продукт, якому потрібні керований PostgreSQL, черга, мережа доставки контенту, об’єктне сховище і місце для запуску контейнерів, може отримати все це в одному рахунку й одній системі ідентифікації, з документацією за спільним зразком. Ця широта має і свою ціну. Кожен сервіс має власні тарифні параметри, власні ліміти й власний куточок мови політик IAM, тож команда, яка запровадила десять сервісів, відтепер підтримує десять невеликих інтеграцій з інтерфейсами Amazon. Переїзд згодом означає замінити кожну з них.

Друге рішення — що лишається на боці клієнта. AWS публікує модель спільної відповідальності й проводить межу двома англійськими фразами: AWS відповідає за безпеку «of the cloud», тобто за обладнання, програмне забезпечення, мережі й приміщення, на яких працюють його сервіси, а клієнт — за безпеку «in the cloud». Де саме пролягає межа, залежить від сервісу. На інстансі EC2 клієнт оновлює гостьову операційну систему, доглядає за кожною встановленою програмою і налаштовує security group, тобто брандмауер перед інстансом. У S3 чи DynamoDB операційну систему й платформу веде AWS, а клієнтові лишаються дані, параметри їх шифрування і дозволи, що визначають, хто може їх читати.

Третє — вартість, яка в AWS є властивістю архітектури більше, ніж рахунком, що надходить згодом. Well-Architected Framework від Amazon називає шість опор, за якими варто оцінювати навантаження: операційна досконалість, безпека, надійність, ефективність продуктивності, оптимізація витрат і

сталість. Вартість потрапила до цього переліку, бо ту саму систему можна збудувати дешево або дорого рішеннями першого тижня: розміри інстансів, кількість зон, обсяг даних, що виходять за межі регіону, і вибір керованих сервісів, які замінюють процес на звичайному сервері.

Як це виглядає на практиці

На сторінці технологічного стека AWS стоїть поруч із Google Cloud як одна з двох хмар, у яких компанія працює за замовчуванням, а Azure додається там, де клієнт уже в ньому працює.

Один проєкт використовував обидві ці хмари водночас: двадцять Docker-застосунків працювали в Google Cloud і AWS, серед них аналітичні застосунки на R. Їх розгортали й підтримували з однаковою конфігурацією та моніторингом в обох хмарах, щоб жоден застосунок не перетворився на окремий виняток. Коли щось ламалося, усунення несправностей ішло через Podman, а налагодження оркестрації — через Kubernetes, і аналітичні застосунки лишалися стабільними й відтворюваними в продакшні з обох боків.

Інше замовлення рухалося у протилежний бік. Для міжнародного клієнта середовище AWS було надмірно забезпечене ресурсами й накопичило невикористовувані сервіси, а витрати зросли непропорційно до потреб бізнесу. У межах роботи перевірили, які сервіси використовувалися, прибрали зайві, скоригували решту потужностей і перенесли навантаження з AWS. Саме ці виконані інфраструктурні зміни описує запис. Він не встановлює співвідношення витрат за порівнянних умов, показника продуктивності чи факту міграції без збоїв у роботі. Приклад корисний методом дослідження наявного середовища та зміни того, що потрібно бізнесу для роботи, а не обіцянкою певної економії.

Обидва випадки зводяться до однієї звички. Сервіс з оплатою за використання виставляє рахунок за все, що ввімкнено, тож середовище в AWS лишається здоровим, коли хтось регулярно звіряє те, що працює, з тим, чим бізнес справді користується.

Де виникають проблеми

AWS рідко втрачає дані чи довго лежить. Дорогі інциденти зазвичай починаються всередині облікового запису самого клієнта.

Root-користувач виконує щоденну роботу. Кожен обліковий запис починається з root-користувача, тобто адреси електронної пошти й пароля, якими його відкрили, з необмеженим доступом до всіх ресурсів і налаштувань, аж до закриття облікового запису. Якщо входити як root для рутинних задач, один пароль, що витік, відкриває все. AWS радить берегти облікові дані root, захистити їх багатофакторною автентифікацією і тримати для тих небагатьох задач, які їх потребують, а щоденну роботу вести через федеративний вхід і ролі.

Довготривалі ключі доступу витікають. Пара ключів доступу працює, доки її хтось не видалить, і легко мандрує: у dotfiles на ноутбуці, у змінну CI, в образ контейнера, у коміт. Перелік найкращих практик IAM починає з протилежного боку. Люди входять через постачальника ідентичності й отримують тимчасові облікові дані, а навантаження беруть тимчасові облікові дані з ролі IAM, прив'язаної до машини, на якій вони працюють. Де довготривалий ключ справді потрібен, його регулярно змінюють, а ключі, користувачів і ролі, якими давно не користуються, видаляють.

Сповіщення бюджету вважають гальмом. AWS Budgets надсилає сповіщення, коли фактичні чи прогнозовані витрати перетинають поріг, а дані оновлюються до трьох разів на добу, тож сповіщення повідомляє про гроші, які вже витрачено. Ресурси працюють далі й далі нараховують плату. Budgets може й виконати дію, наприклад застосувати політику IAM чи service control policy, що забороняє створювати нові ресурси, або втрутитися в роботу певних інстансів EC2 чи RDS. Але лише тоді, коли цю дію налаштували заздалегідь і дали їй роль, через яку діяти.

Усе стоїть в одній зоні доступності. Один інстанс з одним диском в одній зоні — найдешевша схема, і вона поділяє долю цього єдиного місця. Зони існують для того, щоб збій живлення чи мережі в одній лишав інші працювати, і документація AWS радить розподіляти застосунок між кількома. Важлива база даних отримує резервний екземпляр у другій зоні, а вебрівень — інстанси щонайменше у двох, за балансувальником навантаження, що охоплює обидві.

Бакет, відкритий для демонстрації, так і лишається відкритим. Нові бакети S3 за замовчуванням блокують публічний доступ, але політика бакета чи дозвіл на об'єкт змінюють це за хвилину, і посилання, яке надіслали клієнтові на один день, лишається публічним і через рік. S3 Block Public

Access, увімкнений для всього облікового запису чи організації, переважає ці політики. Вмикайте його скрізь, де обліковому запису нема чого публікувати, а публічні файли віддавайте з бакета, створеного саме для цього.

Що почитати далі

- AWS Documentation — покажчик посібників користувача, довідників API та навчальних матеріалів для кожного сервісу AWS.
- Shared responsibility — пояснення з опори безпеки Well-Architected, які рівні захищає AWS, а які клієнт.
- The pillars of the framework — шість напрямів, за якими AWS пропонує оцінювати навантаження, зокрема вартість і надійність.
- AWS Regions and Availability Zones — як ізольовано регіони й зони і чому застосунку варто охоплювати кілька зон.
- Security best practices in IAM — федерація, ролі, багатофакторна автентифікація, root-користувач і найменші привілеї в одному переліку.
- Managing your costs with AWS Budgets — бюджети на фактичні та прогнозовані витрати, їхні сповіщення і дії, які вони можуть виконати.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке AWS?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/aws/