Engineer

DNS

🗺️ довідник, що зіставляє назву з адресою

Що це таке

DNS, або Domain Name System, — це розподілений довідник, до якого інтернет звертається раніше, ніж відбувається майже будь-що інше. Програма, що має назву на кшталт `example.com`, питає в нього адресу для з'єднання, а та сама система відповідає ще на кілька запитань про цю назву: які сервери приймають її пошту, який текст опублікував власник і які сервери за неї відповідають.

Довідник має форму дерева, яке читають справа наліво. Угорі корінь, під ним домени верхнього рівня, як-от `.com` чи `.ua`, нижче — домени, які реєструють люди й компанії, а ще нижче — піддомени, що їх створюють власники. Жоден сервер не тримає дерево цілком. Кожен рівень передає відповідальність за наступний названим серверам, а частину дерева, за яку відповідає один набір серверів, називають зоною. Хто керує зоною, той і вирішує, куди вказують її назви.

Роботу ділять між собою два типи серверів. Авторитетні сервери зберігають зону й дають щодо неї остаточну відповідь. Рекурсивні резолвери, які тримають інтернет-провайдери, компанії чи публічні сервіси, проходять увесь шлях замість клієнта: питають кореневий сервер, де шукати `.com`, питають `.com`, де шукати `example.com`, питають сервери самого домену про потрібний запис і повертають відповідь. Операційна система на вашому комп'ютері має лише невеликий stub-резолвер, який передає запитання рекурсивному й чекає.

Відповіді — це типізовані записи, а файл зони — класичний спосіб їх записати:

```
$ORIGIN example.com.
$TTL 3600
@     IN  SOA    ns1.example.net. hostmaster.example.com. (
                 2026101701 7200 900 1209600 300 )
@     IN  NS     ns1.example.net.
@     IN  A      192.0.2.10
@     IN  AAAA   2001:db8::10
www   IN  CNAME  example.com.
@     IN  MX     10 mail.example.net.
@     IN  TXT    "v=spf1 include:_spf.example.net -all"
```

`A` та `AAAA` дають адреси IPv4 та IPv6. `CNAME` робить одну назву псевдонімом іншої. `MX` вказує сервер, що приймає пошту, а `TXT` несе довільний текст, тобто на практиці правила автентифікації пошти й докази того, що хтось контролює домен. `NS` і `SOA` описують саму зону.

Кожен запис має час життя, TTL, тут 3600 секунд, і саме це число визначає поведінку DNS. Резолвер, який отримав відповідь, може використовувати її повторно, доки не мине TTL, і не питати знову. У результаті більшість запитів отримує відповідь із кешу поблизу клієнта, а система продовжує відповідати, навіть коли авторитетний сервер ненадовго зникає. Ціна така: змусити світ забути стару відповідь раніше ніхто не може.

Застосуйте поняття до рішення

Застосуйте «Що таке DNS?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Wi-Fi, фаєрволи і віддалений доступ налаштовані так, щоб працювати можна було звідусіль, а заходити — лише своїм.

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. Керувала 30 інстансами Ubuntu Linux VPS — 30 серверів Linux під управлінням, із планом відновлення

Керувала 30 інстансами Ubuntu Linux VPS Зв'язатися з нами →

Чому це важливо

Кожне з’єднання з сервісом, що має назву, починається з пошуку в DNS. Коли DNS налаштовано неправильно, сервери можуть бути цілком справними, а сайт, API і пошта однаково виглядають недоступними, бо до них ніщо не доходить. Коли DNS налаштовано правильно, його ніхто не помічає, і тому саме ця частина системи зазвичай описана найгірше.

У DNS також здійснюють і доводять контроль над доменом. Поштові сервіси читають правила SPF, DKIM і DMARC із записів `TXT`, щоб вирішити, чи справжній лист, який нібито надійшов із вашого домену, тож хибний або відсутній запис відправляє справжню пошту в спам. Центри сертифікації й хмарні сервіси просять додати запис із певним значенням як доказ того, що назва ваша. Хто може редагувати зону, той може перенаправити сайт, перехопити пошту й отримати сертифікати для домену, тож доступ до облікового запису DNS такий самий чутливий, як доступ до серверів.

Кешування впливає на кожну зміну. Те, що називають поширенням, насправді є тисячами резолверів, кожен з яких чекає, поки спливе термін його власної копії. Щоб перенести сайт якомога м'якше, TTL знижують приблизно за добу до зміни, чекають, поки мине старий TTL, вносять зміну, а коли все усталиться, знову його підвищують.

Класичний DNS передається відкритим текстом, і ні відповіді, ні запитання не є приватними. DNSSEC дає резолверу змогу перевірити, що відповідь справді надійшла від власника зони й дорогою її не змінили, але, за власною специфікацією, конфіденційності не забезпечує. Зашифровані транспорти, як-от DNS over HTTPS, ховають запитання від мережі на шляху, проте резолвер на іншому кінці однаково бачить кожну назву, яку в нього питають. Обрати резолвер означає обрати, хто отримає цей список.

Як це виглядає на практиці

Власна зона компанії показує найпоширенішу відправну точку і спосіб вибратися з неї. У ній було близько двадцяти записів, які зберігалися у вебконсолі провайдера: вебприсутність, git-піддомен, два особисті перенаправлення, пошта через хостованого постачальника поштових скриньок, домен для транзакційних листів і записи автентифікації для обох. У консолі поточний стан — це те, що лишила остання людина, яка там клацала. Тепер зона — це дані в репозиторії: живі записи, записи застосунків, записи транспортної політики та окремий перелік записів, призначених до вилучення. Один плей Ansible узгоджує це оголошення з API провайдера. Ще дві точки входу запускаються лише за явно вказаними тегами: перевірка, що показує різницю й нічого не змінює, і експорт, що виписує зону в стандартному форматі файлу зони, аби її могло прочитати щось, окрім репозиторію. Найчастіше використовують саме перевірку, бо знати різницю потрібно частіше, ніж її усувати.

Той самий провайдер має свідомо вузьке завдання. Сторінка подяк сайту пояснює, що Cloudflare відповідає для сайту за DNS і більше ні за що, без проксі попереду, і прямо наводить адресу сервера, бо DNS однаково повідомляє її кожному, хто запитає.

Читання DNS входить і до звичайного опису інфраструктури. Серед семи ролей лише для читання, що відтворюють живий хост у Markdown, є інвентаризація провайдерів, яка читає DNS поряд із двома хмарними API та API виділених серверів. Правило для всіх однакове: жоден звіт не друкує числа, якого його запуск не виміряв.

Де виникають проблеми

Збої DNS часто відкладені, закешовані або розділені, тож та сама назва працює в однієї людини й не працює в іншої. Ось ті, що повторюються.

Зміна не набуває чинності. Хтось редагує запис із TTL в одну добу й чекає, що нова адреса запрацює одразу, але резолвери, які закешували стару відповідь, віддають її, доки не спливе їхня копія. Тихіший варіант чекає на нові назви: якщо назву шукали ще до появи запису, кешується й негативна відповідь, на стільки, скільки дозволяє `SOA` зони (RFC 2308). Знижуйте TTL перед запланованими змінами і створюйте записи раніше, ніж їх хтось шукатиме.

Псевдонім стоїть там, де йому не місце. RFC 1034 каже, що назва із записом `CNAME` не може мати жодних інших даних. Голий домен, тобто сам `example.com`, завжди має записи `SOA` і `NS`, тож зона, яка дотримується стандарту, не може зробити корінь домену псевдонімом. Те саме правило порушує `CNAME` на піддомені, якому також потрібні записи `MX` чи `TXT`. Провайдери пропонують «сплощені» записи або `ALIAS` як власний обхідний шлях, і кожен поводиться трохи по-своєму, тож перевірте, що саме віддає ваш.

Автентифікація пошти ламається. Кожен `include` у записі SPF спричиняє запити до DNS, а RFC 7208 обмежує їхню загальну кількість десятьма. Ще один сервіс розсилок чи служба підтримки може виштовхнути запис за межу, і тоді отримувачі вважають правило постійною помилкою, а пошту, яку воно мало засвідчувати, оцінюють без нього. Підраховуйте запити щоразу, коли додаєте нового відправника.

Запис переживає свою ціль. Піддомен через `CNAME` вказує на хмарне сховище чи розміщений застосунок, які вже видалили. Якщо провайдер дозволяє будь-кому зайняти цю назву ресурсу, хтось сторонній може її забрати й показувати власний вміст під вашим піддоменом, прикриваючись репутацією вашого домену. Видаляйте запис у тій самій зміні, яка прибирає те, на що він вказує.

Зона живе тільки в консолі. Немає ні історії, ні рецензії, ні порівняння, тож випадково видалений запис `MX` помічають, коли перестає надходити пошта, а попереднє значення доводиться відновлювати з пам'яті. Регулярний експорт зони, а ще краще — її опис у коді, дає кожній зміні слід і шлях назад.

Що почитати далі

- RFC 1035: Domain Names, Implementation and Specification — формати записів, синтаксис файлу зони й мережевий протокол.
- RFC 1034: Domain Names, Concepts and Facilities — дерево, зони, делегування і те, як резолвери ними проходять.
- RFC 9499: DNS Terminology — точні визначення слів, якими користується ця сторінка.
- RFC 2308: Negative Caching of DNS Queries — як довго резолвер може пам'ятати, що назви не існує.
- RFC 4033: DNS Security Introduction and Requirements — що захищає DNSSEC, а чого не торкається.
- RFC 8484: DNS Queries over HTTPS — запити до DNS усередині звичайного зашифрованого вебтрафіку.

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 DNS?» 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/dns/