Engineer

Моніторинг

спостереження за системою, щоб вона повідомила першою

Що це таке

Система під моніторингом постійно звітує про себе, а щось безперервно читає ці звіти, порівнює їх із межами, які хтось провів, і вирішує, коли про це має дізнатися людина. Книга Google про Site Reliability Engineering визначає моніторинг як збирання, обробку, агрегування й відображення кількісних даних про систему в реальному часі: скільки запитів надходить і яких саме, скільки з них завершуються помилкою і якою, скільки часу вони тривають і скільки часу працює кожен сервер. Мета — певний порядок подій: система каже, що в неї біда, раніше, ніж це зробить користувач.

Дані надходять із двох боків. Моніторинг «білої скриньки» (white-box) читає те, що система сама показує про свій внутрішній стан, — сторінку з метриками, довжину черги чи кількість повторних спроб, — тож бачить, як назріває проблема, ще до того, як її хтось відчує. Моніторинг «чорної скриньки» (black-box) стоїть зовні й поводиться як користувач: запитує сторінку, розв’язує доменне ім’я чи проходить вхід і повідомляє, чи це вдалося і скільки тривало. Він бачить лише те, що вже зламалося, і саме тому є надійною підставою когось розбудити.

Більшість того, що зберігає система моніторингу, — це числа, виміряні з плином часу, бо їх дешево зберігати й легко порівнювати з порогом. Той самий розділ книги SRE називає чотири з них відправною точкою для всього, що обслуговує користувачів: затримка — час, який займає запит; трафік — скільки попиту надходить; помилки — частка запитів, що завершуються невдачею; і насичення — наскільки найзавантаженіший ресурс близький до вичерпання.

Навколо чисел є два види виходу. Дашборд — щоб дивитися, коли хтось уже щось підозрює. Тривога — щоб перервати, коли всі зайняті чимось іншим. В екосистемі Prometheus правила обчислює сервер, а Alertmanager групує, прибирає дублікати й надсилає далі те, що спрацювало.

Двох сусідів із ним легко сплутати. Журналювання записує окремі події, по рядку на кожну, і його читають уже після тривоги, щоб з'ясувати причину. Спостережуваність (observability), як її описує проєкт OpenTelemetry, — ширша здатність ставити системі нові запитання ззовні. Моніторинг — вужча й старша практика: запитання, визначені заздалегідь, які ставлять безперервно, з відповіддю, що може підняти людину.

Застосуйте поняття до рішення

Застосуйте «Що таке моніторинг?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

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. Запустила перший дашборд спостережуваності — перший огляд власних систем компанії

Запустила перший дашборд спостережуваності Зв'язатися з нами →

Чому це важливо

Альтернатива — дізнаватися про проблеми від тих, кого вони зачіпають, о тій годині, коли вони їх помітять. Моніторинг переносить цей момент раніше й передає його тому, хто може діяти.

Дефіцитний ресурс тут — увага, а не програмне забезпечення. Виклик чергового перериває роботу вдень і сон уночі, і книга SRE говорить прямо: коли виклики надходять надто часто, люди їх проглядають, сумніваються в них і зрештою ігнорують, іноді разом зі справжнім, захованим у шумі. Пораду проєкту Prometheus можна вмістити в одне речення: тримайте сповіщення простими, ставте тривоги на симптоми, майте добрі консолі для пошуку причин і не викликайте людину туди, де їй нема чого робити. Кожне правило, здатне розбудити людину, — це обіцянка, що коли воно спрацює, у неї буде щось термінове до виконання.

Тож головне рішення стосується того, що відчувають користувачі. Диск, заповнений на 85%, — це причина; оформлення замовлення, яке зривається, — симптом. Тривоги на симптоми тримають кількість правил

невеликою, бо способів зашкодити користувачам значно менше, ніж способів, якими можуть збоїти машини. Причинам усе одно місце на дашбордах і в тікетах, де вони допомагають шукати несправність, нікого не будячи. Чесний виняток — ємність: ресурс, близький до вичерпання, сьогодні нікому не шкодить, але скоро знадобиться людина.

Далі постає питання: тримати моніторинг власним чи орендувати. Власний означає сховище часових рядів, маршрутизатор тривог і людину, яка стежить за справністю обох. Хмарний сервіс забирає цю роботу й виставляє рахунок за обсягом надісланих даних, тож кожен новий хост з'являється в рахунку. Книга SRE зазначає, що навіть у Google в команді з 10–12 інженерів зазвичай один, а іноді два мають моніторинг основною роботою. Так чи інакше, це система з власним обслуговуванням, і розмір її слід добирати під систему, за якою вона стежить.

Як це виглядає на практиці

Дві попередні посади засновниці та власний хост компанії ілюструють ту саму ідею в дуже різних масштабах.

З 2015 до 2022 року, на посаді інженерки інфраструктури, завданням була цілодобова підтримка приблизно тисячі серверів IPTV/OTT-стрімінгової платформи, а також чималої кількості систем, що належали клієнтам, для замовників у Китаї, США та Німеччині. Із клієнтами в трьох часових поясах тихої години не існувало, і проблема о 3-й ночі за місцевим часом була прайм-таймом для когось іншого. Значна частина роботи полягала в тому, щоб помітити відхилення до того, як воно переросте у збій, і проблеми усували ще до того, як вони досягали екрана глядача. На такому сервісі успіх виглядає як відсутність подій, а моніторинг дає команді знати це, а не сподіватися.

На посаді інженерки баз даних у 2022–2024 роках, на 100 базах даних у PostgreSQL, MS SQL і BigQuery, автоматизований моніторинг скоротив час простою на 50%.

Моніторинг власного хоста публічних сервісів компанії означає роботу на протилежному кінці шкали. Одна машина з 512 МБ обслуговує ці публічні сервіси; близько 464 МБ її пам'яті придатні до використання, і саме пам'ять визначає кожне проєктне рішення на цьому хості. Звітність там працює за запитом, а не через агент, конвеєр і дашборд: сім команд, які зчитують стан

хоста й формують Markdown, нічого не змінюють і виконуються лише на запит. Рішення щодо onion-сервісу показує цей підхід: коли виникло питання, показники хоста збирали протягом доби, і за ними ухвалили рішення. У цих зафіксованих вибірках мінімум доступної пам’яті становив близько 310 МБ із 464, використання підкачування трималося на 2,6%, а процесор простоював 99,7% часу. За такого розміру моніторинг — це вимірювання саме тоді, коли воно потрібне для рішення.

Де виникають проблеми

Виклики надходять так часто, що люди перестають їх читати. Кожен інцидент залишає по собі правило, багато з них спрацьовують на причини, що минають самі, і за кілька місяців справжній виклик проглядають разом з рештою. Викликайте людину лише через те, що помітили б користувачі, в одній точці стеку, а все інше відправляйте в чергу тікетів чи на дашборд. Потім дайте умові протриматися якийсь час, перш ніж вона спрацює, щоб збій, який минає за хвилину, нікого не будив:

```
groups:
    - name: checkout
      rules:
          - alert: CheckoutErrorsHigh
            expr: job:checkout_errors:ratio5m > 0.02
            for: 10m
            labels:
                severity: page
```

Моніторинг зупиняється, а його мовчання сприймають як добру новину. Правило, яке ніколи не спрацьовує, виглядає точнісінько як система, що ніколи не ламається. Якщо падає сервер Prometheus, Alertmanager чи поштовий ретранслятор, усі тривоги зупиняються одночасно. Власна порада Prometheus — ставити тривоги на сам моніторинг, а ще краще — перевіряти весь шлях наскрізь, щоб одна перевірка доводила, що тривога може пройти від правила до поштової скриньки. У ту саму пастку потрапляють заплановані завдання: резервне копіювання, що перестало запускатися, не видає помилки, воно просто перестає залишати сліди. Викликайте людину, коли останній успіх надто давній, із запасом щонайменше на два повні запуски; для завдання, яке запускається кожні 4 години й триває годину, проєкт пропонує 10 годин:

```
time() - backup_last_success_timestamp_seconds > 10 * 3600
```

Середнє каже, що все гаразд, поки частина користувачів чекає. Середні значення легко показувати на графіку, і вони ховають хвіст. Приклад із книги SRE — сервіс із середньою затримкою 100 мс при 1 000 запитів на секунду, де 1% запитів цілком може тривати 5 секунд; коли сторінка залежить від кількох таких сервісів, 99-й перцентиль одного бекенда може стати медіаною фронтенда. Швидкі помилки погіршують картину, бо швидка відмова тягне середнє вниз. Записуйте затримку в бакети гістограми, ставте тривогу на високий перцентиль і рахуйте невдалі запити окремо від успішних.

Усі внутрішні показники зелені, а користувачі не можуть дістатися до сервісу. Метрики білої скриньки походять зсередини системи, тож поділяють її сліпі зони: прострочений сертифікат, зламаний запис DNS чи зміна в правилах брандмауера можуть залишити кожну з них здоровою. Перевірка чорної скриньки з машини поза мережею системи ловить те, що інакше лишається невидимим, і продовжує працювати, коли внутрішній моніторинг падає разом з усім іншим.

Що почитати далі

- Prometheus: Overview — що збирає система моніторингу на основі метрик, з яких частин вона складається і де точність до кожної події вимагає іншого інструмента.
- Alerting — поради проєкту Prometheus про те, через що варто викликати людину: від онлайн-сервісів і пакетних завдань до моніторингу самого моніторингу.
- Alertmanager — як тривоги очищують від дублікатів, групують, тимчасово вимикають, пригнічують і надсилають отримувачеві.
- Monitoring Distributed Systems — розділ 6 книги Google про Site Reliability Engineering: симптоми й причини, чотири золоті сигнали і хвіст.
- Observability primer — огляд OpenTelemetry про сигнали, SLI та SLO і про те, де спостережуваність сягає далі за моніторинг.

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 моніторинг?» 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/monitoring/