# Ми супроводжуємо платформи, **на яких працює ваш софтвер,** — відтворювані з коду, під наглядом і задокументовані.

Хмара, інфраструктура як код, CI/CD, контейнери, моніторинг, резервні копії, мережі та доступи — з Копенгагена.

---

Engineer company
Платформа

**Інфраструктура, яку можна відтворити з репозиторію**

+45 50 10 61 84 · Копенгаген, Данія

**Дбайливо зроблено з любов'ю — D.**

Хмарна інфраструктура як код, автоматизоване розгортання та планування відновлення систем, від яких залежить ваш бізнес. Описи робіт містять початкову ситуацію, виконані дії та результат.

Почніть з одного процесу розгортання або експлуатаційного ризику. Узгодимо аналіз чи реалізацію з документованим планом розгортання, відкату й відповідальності, щоб ваша команда знала, чим і як керувати.

Інфраструктурні роботи · Ціни та формати співпраці · Обговорити ваш проєкт

## Почніть із потрібного результату

Розкажіть, чого має досягти ваш бізнес. Ми поєднуємо програмне забезпечення, дані та інфраструктуру для людей, які ними користуються.

**Переглянути наші послуги** Зв'язатися з нами →

## Дізнайтеся, як ми працюємо

Знайдіть послугу відповідно до своєї мети й перегляньте приклади того, що було виконано та як.

**Кейси** Зв'язатися з нами →

## Розгляньте роботу **в її контексті** — рішення, виконання та описані результати.

**Подивитися весь каталог**

## Один партнер — **10 послуг** від першого сервера до сповіщення о третій ночі.

**Переглянути каталог**

{ Хмара, інфраструктура та DevOps }

# Там, де працює програма

Хмарна інфраструктура та міграція

Сервери під те навантаження, яке у вас справді є, перенесені в хмару чи назад, із порахованим наперед місячним рахунком.

### Підготувала другий сервер із коду

Автоматизувала підготовку (provisioning) другого сервера в другого хмарного провайдера, створюючи фаєрвол раніше за машину, щоб вона від самого початку була за ним, а фаєрволи обох провайдерів налаштовуються безпосередньо через їхні REST API, без сторонньої колекції.

### Запустила контейнери на Podman і Quadlet

Розгорнула контейнерну платформу на Podman і Quadlet під systemd, а не на Docker, бо типове публікування портів Docker у мережі bridge може обходити правила UFW, — а для розгортань виділила непривілейованого користувача з однією фіксованою командою замість root.

### Публічні служби на невеликому сервері

Розмістила публічні служби компанії на одноядерному хості з 512 МБ — git-сервер, вебсервер для семи доменів, Tor, два сервери альтернативних протоколів і блокування вторгнень, — використовуючи 464 МБ доступної пам'яті як проєктне обмеження; автоматизація резервування є окремою реалізацією.

**Переглянути всі 9 досягнень**

# Як ми працюємо

## Спочатку зрозуміти

Ми розбираємося в проблемі, перш ніж пропонувати рішення. Запит, з яким ви приходите, — це початок розмови, а не її кінець.

## Будувати надовго

Довговічне замість швидкого, ясність замість вигадливості. Із системою мають упоратися й ті, кому вона дістанеться далі.

## Відповідати за зроблене

Поточну експлуатацію чи підтримку узгоджуємо в межах співпраці. Передача результатів визначає відповідальність і подальший шлях зв’язку.

{ Хмара, інфраструктура та DevOps }

## Сервери, які можна відтворити з версійованої конфігурації

Infrastructure as Code

Уся конфігурація записана у файлах, тож її можна відтворити точно, і вона не живе лише в чиїйсь голові.

### Опублікувала читабельне onion-дзеркало в Tor

Опублікувала сайт як onion-дзеркало в Tor за читабельною адресою, що починається з engineer, з ключем, згенерованим поза хостом, тож він ніколи не потрапив ні до цього репозиторію, ні на ноутбук, і залишила цей вхід єдиним із чотирьох, відвідування якого ніколи не рахуються.

### Перевірила CI-хук git-сервера, знайшла 2 дефекти

Відрепетирувала CI-хук на боці git-сервера й знайшла два дефекти, яких не виявити читанням файлу: запасний варіант, що передавав на вхід хука аргумент, який неможливо розв'язати, і власну змінну середовища git, яка потрапляла слідом за гейтом у checkout і робила 19 тестів червоними.

### Розділила всі операційні секрети за хостами

Розділила чотири операційні секрети так, щоб кожен хост мав власні, з'ясувавши, що два хости зі спільним dead man's switch сповіщають про збої рідше, ніж два окремі перемикачі, а не частіше, і що спільна парольна фраза резервних копій може відкрити доступ до копій обох хостів навіть в окремих репозиторіях, якщо хтось отримає її та зможе прочитати зашифровані дані й файли ключів обох репозиторіїв.

**Переглянути всі 17 досягнень**

{ Хмара, інфраструктура та DevOps }

# Випускати зміни без нервів

DevOps та автоматизація CI/CD

Перевірено, випущено і з можливістю відкату, тож зміни доходять до клієнтів у вівторок удень, а не пізно в п'ятницю.

### Привела 4 репозиторії до одного стандарту історії

Привела чотири репозиторії до одного стандарту історії — conventional-коміти, без емодзі, без рядків атрибуції, з контролем через хук повідомлень комітів, — поряд із 48 документами інструкцій, що визначають, як виконується робота.

### Знайте ліцензію всього, що випускаєте

Запровадила перевірку сумісності ліцензій 644 крейтів Rust під час кожної збірки й довела, що вона спрацьовує, переписавши ліцензію одного крейта та змістивши зафіксовану ревізію ядра без перегенерації.

### Задіяла невикористану половину ядра

Задіяла ту половину ядра обміну повідомленнями, якою застосунок ніколи не користувався, — передавання резервної копії, зникомі повідомлення, редагування та повторне надсилання, підтверджені запрошення, проксі та політику шифрування, — проганяючи кожен тест на справжній бібліотеці без заглушок.

**Переглянути всі 43 досягнення**

{ Хмара, інфраструктура та DevOps }

# Програма, що всюди працює однаково

Контейнеризація та оркестрація

Запакована так, що поводиться однаково на ноутбуці, на тестовому сервері й у продакшні, і масштабується, коли настає напружений тиждень.

### Запустила контейнери на Podman і Quadlet

Розгорнула контейнерну платформу на Podman і Quadlet під systemd, а не на Docker, бо типове публікування портів Docker у мережі bridge може обходити правила UFW, — а для розгортань виділила непривілейованого користувача з однією фіксованою командою замість root.

### Посилила ізоляцію systemd для кожного юніта

Знизила оцінку відкритості (exposure) у пісочниці systemd для кожного юніта, який встановлює платформа: служба dead man's switch — з 9,6 UNSAFE до 1,5, відкритий в інтернет git-сервер — з 8,3 EXPOSED до 1,5, — і додала перевірку парсером під час converge, після того як знайшла у трьох шаблонах юнітів директиву з орфографічною помилкою, яку systemd мовчки ігнорував.

### Побудувала пайплайни CI/CD на GitHub Actions

Побудувала пайплайни CI/CD на GitHub Actions з distroless-образом продакшн-фронтенду та просуванням релізів між кількома середовищами.

**Переглянути всі 5 досягнень**

{ Хмара, інфраструктура та DevOps }

## Дізнаватися раніше за клієнтів

Надійність та моніторинг (SRE)

Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

### Змусила шість збирачів звітувати про пропущене

Змусила шість збирачів звітів повідомляти ту частину свого входу, яку вони так і не прочитали, — файли, що лишилися закритими, невстановлені зонди, нерозпізнані рядки журналу, — і підігнала пороги до 116 виміряних зразків, з'ясувавши, що сліпий збір і справний хост дають однакову сторінку.

### Виміряла 13 адрес без трекінгу

Виміряла тринадцять адрес лише за власним журналом доступу вебсервера — без кукі, без трекерів і без сторонніх сервісів — маскуючи мережу клієнта в момент запису рядка та згортаючи результат у постійний архів із 21 денного показника та 20 місячних фасетів.

### Обмежений потік подій у застосунку macOS

Обмежила буфери подій і журналювання після зафіксованого інциденту macOS зі зростанням на 41 МБ за секунду й 111 ГБ стиснених сторінок на машині з 36 ГБ; зафіксований повний тестовий запуск зменшив журнал ядра з 610 996 до 1 411 рядків.

**Переглянути всі 23 досягнення**

{ Хмара, інфраструктура та DevOps }

# Повернути все після поганого дня

Резервне копіювання та відновлення

Резервні копії, які перевірено, а не просто вважають робочими, і записаний план на той ранок, коли сервер не ввімкнеться.

### Розділила всі операційні секрети за хостами

Розділила чотири операційні секрети так, щоб кожен хост мав власні, з'ясувавши, що два хости зі спільним dead man's switch сповіщають про збої рідше, ніж два окремі перемикачі, а не частіше, і що спільна парольна фраза резервних копій може відкрити доступ до копій обох хостів навіть в окремих репозиторіях, якщо хтось отримає її та зможе прочитати зашифровані дані й файли ключів обох репозиторіїв.

### Автоматизація резервування з явним способом відновлення

Реалізувала автоматизацію резервування restic з політикою зберігання, перевіркою цілісності та щомісячним тестом відновлення; задокументувала активацію сховища й перевірку справжнього відновлення як окремі операційні кроки.

### Налаштувала резервні копії та відновлення як код

Налаштувала зберігання резервних копій бази даних у вигляді інфраструктури як коду (infrastructure-as-code), провела аудит готовності до відновлення й задокументувала процедуру відновлення — чітко назвавши наявні прогалини, а не залишивши їх на час інциденту.

**Переглянути всі 5 досягнень**

{ Хмара, інфраструктура та DevOps }

# Щоб усе просто працювало

Системне адміністрування

Щоденний догляд за серверами, поштою, обліковими записами й оновленнями, щоб те, на що всі покладаються, працювало тихо й без сюрпризів.

### Опублікувала читабельне onion-дзеркало в Tor

Опублікувала сайт як onion-дзеркало в Tor за читабельною адресою, що починається з engineer, з ключем, згенерованим поза хостом, тож він ніколи не потрапив ні до цього репозиторію, ні на ноутбук, і залишила цей вхід єдиним із чотирьох, відвідування якого ніколи не рахуються.

### Змусила шість збирачів звітувати про пропущене

Змусила шість збирачів звітів повідомляти ту частину свого входу, яку вони так і не прочитали, — файли, що лишилися закритими, невстановлені зонди, нерозпізнані рядки журналу, — і підігнала пороги до 116 виміряних зразків, з'ясувавши, що сліпий збір і справний хост дають однакову сторінку.

### Перевірила CI-хук git-сервера, знайшла 2 дефекти

Відрепетирувала CI-хук на боці git-сервера й знайшла два дефекти, яких не виявити читанням файлу: запасний варіант, що передавав на вхід хука аргумент, який неможливо розв'язати, і власну змінну середовища git, яка потрапляла слідом за гейтом у checkout і робила 19 тестів червоними.

**Переглянути всі 28 досягнень**

{ Хмара, інфраструктура та DevOps }

## З'єднати ваші офіси і ваших людей

Налаштування мереж та VPN

Wi-Fi, фаєрволи і віддалений доступ налаштовані так, щоб працювати можна було звідусіль, а заходити — лише своїм.

### Підготувала другий сервер із коду

Автоматизувала підготовку (provisioning) другого сервера в другого хмарного провайдера, створюючи фаєрвол раніше за машину, щоб вона від самого початку була за ним, а фаєрволи обох провайдерів налаштовуються безпосередньо через їхні REST API, без сторонньої колекції.

### Описала зону DNS із 20 записів як код

Налаштувала декларативне узгодження зони DNS із 20 записів з API Cloudflare, з окремими точками входу для аудиту та експорту в BIND, а побудувавши проксі CDN, знову вимкнула його з міркувань приватності.

### Перевірила правила фаєрвола за позицією

Перевірила правила фаєрвола за позицією, а не лише за наявністю, зчитуючи пронумерований перелік правил і чинний ланцюжок фільтрації пакетів, бо наявне правило ще не означає, що до нього доходить хоч один пакет.

**Переглянути всі 11 досягнень**

{ Хмара, інфраструктура та DevOps }

# Визначити, хто до чого має доступ

Безпека та керування доступом

Облікові записи, паролі й права налаштовані так, щоб доступ відповідав роботі, а звільнені втрачали його того ж дня.

### Опублікувала читабельне onion-дзеркало в Tor

Опублікувала сайт як onion-дзеркало в Tor за читабельною адресою, що починається з engineer, з ключем, згенерованим поза хостом, тож він ніколи не потрапив ні до цього репозиторію, ні на ноутбук, і залишила цей вхід єдиним із чотирьох, відвідування якого ніколи не рахуються.

### Виміряла 13 адрес без трекінгу

Виміряла тринадцять адрес лише за власним журналом доступу вебсервера — без кукі, без трекерів і без сторонніх сервісів — маскуючи мережу клієнта в момент запису рядка та згортаючи результат у постійний архів із 21 денного показника та 20 місячних фасетів.

### Знайте ліцензію всього, що випускаєте

Запровадила перевірку сумісності ліцензій 644 крейтів Rust під час кожної збірки й довела, що вона спрацьовує, переписавши ліцензію одного крейта та змістивши зафіксовану ревізію ядра без перегенерації.

**Переглянути 31 досягнення**

{ Хмара, інфраструктура та DevOps }

# Щоб ШІ не чіпав того, чого не слід

Безпека ШІ-агентів

Доступ кожного агента описано й обмежено необхідним, кожну дію записано в журнал, якого агент не може змінити, і кожен захист перевірено на ділі, а не взято на віру.

### Розділила всі операційні секрети за хостами

Розділила чотири операційні секрети так, щоб кожен хост мав власні, з'ясувавши, що два хости зі спільним dead man's switch сповіщають про збої рідше, ніж два окремі перемикачі, а не частіше, і що спільна парольна фраза резервних копій може відкрити доступ до копій обох хостів навіть в окремих репозиторіях, якщо хтось отримає її та зможе прочитати зашифровані дані й файли ключів обох репозиторіїв.

Записала правило меж після витоку облікових даних

Записала в репозиторій правило про межі його вмісту після того, як під час реструктуризації туди потрапили інвентар, дозволи фаєрвола та тексти іншої компанії, — і залишила відокремлений у карантин залишок під наглядом сканера секретів, а не виключила його.

Додала секундну попередню перевірку секретів

Додала попередній плей (preflight), який приблизно за секунду проганяє той самий код, що й converge, на локальних секретах оператора, — після того як продакшн-запуск обірвався на дев'ятому завданні, застосувавшись лише наполовину, а налаштування swap уже було записано на робочий хост.

**Переглянути всі 9 досягнень**

Ми працюємо з **платформами** —
тими, на яких уже працює ваш бізнес.



Apple Microsoft DigitalOcean AWS Google

**Переглянути всі технології**

## Аналіз безпеки ШІ-агентів у погодженому обсязі

### Перелічено

Узгоджений перелік агентів, облікових даних і доступів до даних у межах роботи зберігається разом із вашою конфігурацією.

### Замкнено

Кожен агент має власний обліковий запис і найменші привілеї, а текст, який він читає, вважається даними, а не наказами.

### Записано

Кожна дія записана в журнал, який агент не може змінити, із зазначенням людини, для якої її виконано, а на незвичне спрацьовують сповіщення.

**Як ми захищаємо агентів** Запитайте про своїх агентів →

Розкажіть, що ви будуєте.

Жодних форм, жодних воронок продажів, жодних автоматичних розсилок — лише один лист людині, яка його прочитає.

**Почати розмову**

Ця адреса показує лише роботу з інфраструктурою. Компанія також розробляє програмне забезпечення, працює з даними та консультує команди — це на engineer.company.
Подивитися все, що робить Engineer ApS →

### Почніть із конкретного вдосконалення

Робочий процес, потік даних або система, що потребує вдосконалення, може стати корисним першим завданням. Його обсяг можемо погодити разом.

**Дивитися, скільки це коштує** Зв'язатися з нами →

## Підготуйтеся до першої розмови

Розкажіть, який результат потрібен, що вже є та коли ви хотіли б його отримати. На цій основі обговоримо наступний крок.

**Переглянути наші послуги**

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

На сервері ми працюємо з **Linux із відкритим кодом** — без платних ліцензій і без прихованого коду.


Ubuntu


Debian

**Як ми обслуговуємо сервери**


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://platform.engineer.company/uk/**