Operamos las plataformas **en las que despliega su software** y las reconstruimos desde código, las supervisamos y las documentamos.

Nube, infraestructura como código, CI/CD, contenedores, monitorización, copias de seguridad, redes y acceso: desde Copenhague.

Engineer company
Plataformas

Infraestructura que puede reconstruir desde un repositorio

+45 50 10 61 84 · Copenhague, Dinamarca

Creado con esmero y cariño por D.

Infraestructura de nube como código, despliegue automatizado y planificación de la recuperación para los sistemas de los que depende su negocio. Los registros de trabajo describen la situación inicial, lo realizado y el resultado.

Empiece con una vía de despliegue o un riesgo operativo. Podemos acordar una revisión o una implementación con un plan documentado de despliegue, reversión y responsabilidades, para que su equipo sepa qué debe operar y cómo.

Trabajos de infraestructura · Precios y modalidades de contratación · Hablemos de su proyecto

Empiece por el resultado que necesita

Cuéntenos qué necesita conseguir su empresa. Conectamos software, datos e infraestructura en torno a las personas que los utilizan.

Explore nuestros servicios Póngase en contacto →

Vea cómo abordamos el trabajo

Encuentre el servicio que encaja con su objetivo y vea ejemplos de lo que se entregó y cómo.

Casos prácticos Póngase en contacto →

Conozca el trabajo **en su contexto** — las decisiones, la entrega y los resultados comunicados.

Vea el catálogo completo

Un solo colaborador para **10 servicios** del primer servidor a la alerta de las 3 de la madrugada.

Explore el catálogo

{ Nube, infraestructura y DevOps }

El lugar donde funciona el software

Infraestructura en la nube y migración

Servidores dimensionados para el uso real, trasladados a la nube o fuera de ella, con la factura mensual calculada de antemano.

Segundo servidor aprovisionado desde código

Automaticé el aprovisionamiento de un segundo servidor en un segundo proveedor de nube y creé el cortafuegos antes que la máquina para que naciera protegida, con los cortafuegos de ambos proveedores configurados directamente mediante sus API REST para evitar una colección de terceros.

Contenedores con Podman y Quadlet

Desplegué una plataforma de contenedores con Podman y Quadlet bajo systemd en lugar de Docker, porque la publicación predeterminada de puertos en redes bridge de Docker puede eludir las reglas UFW, y asigné a los despliegues un usuario sin privilegios con una única orden fija, en vez de root.

Servicios públicos en un servidor pequeño

Alojé los servicios públicos de la empresa en un host de 512 MB y un solo núcleo — una forja git, un servidor web para siete dominios, Tor, dos servidores de protocolos alternativos y bloqueo de intrusiones — tomando los 464 MB de memoria utilizable como restricción de diseño; la automatización de copias de seguridad es una implementación aparte.

Ver los 9 logros

Cómo trabajamos

Primero, comprender

Conocemos el problema antes de proponer una solución. La descripción con la que llega inicia esa conversación; no la cierra.

Construir para durar

Durabilidad antes que rapidez, claridad antes que ingenio. El sistema debe seguir siendo manejable para las personas que lo hereden.

Responder por el trabajo

Acordamos la operación continuada o el soporte como parte del encargo. El traspaso deja claras las responsabilidades y la siguiente vía de contacto.

{ Nube, infraestructura y DevOps }

Servidores que se pueden reconstruir a partir de una configuración versionada

Infraestructura como código

Toda la configuración escrita en archivos para poder recrearla con exactitud y evitar que exista solo en la memoria de una persona.

Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

Ensayo del hook CI: 2 defectos detectados

Ensayé el hook CI del servidor de la forja y encontré dos defectos imposibles de alcanzar leyendo el archivo: una alternativa que introducía un argumento irresoluble en la entrada del hook y una variable de entorno de git que acompañaba al control hasta el checkout y hacía fallar 19 pruebas.

Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

Ver los 17 logros

{ Nube, infraestructura y DevOps }

Publicar cambios sin sobresaltos

DevOps y automatización CI/CD

Cambios probados, publicados y reversibles, para que lleguen a sus clientes un martes por la tarde en vez de un viernes por la noche.

Un criterio de historial para 4 repositorios

Apliqué un único criterio de historial a cuatro repositorios — commits convencionales, sin emojis ni líneas de atribución y con un hook de mensajes de commit —, junto con 48 documentos de instrucciones que regulan la forma de trabajar.

Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

La mitad sin utilizar del núcleo, implementada

Implementé la mitad del núcleo de mensajería que la aplicación no utilizaba — transferencia de copias de seguridad, mensajes temporales, edición y reenvío de mensajes, invitaciones verificadas, proxies y política de cifrado —, ejecutando cada prueba contra la biblioteca real, sin simulaciones.

Ver los 43 logros

{ Nube, infraestructura y DevOps }

Software que funciona igual en todas partes

Contenedores y orquestación

Software empaquetado para comportarse igual en un portátil, un servidor de pruebas y producción, y ampliar su capacidad cuando aumente la actividad.

Contenedores con Podman y Quadlet

Desplegué una plataforma de contenedores con Podman y Quadlet bajo systemd en lugar de Docker, porque la publicación predeterminada de puertos en redes bridge de Docker puede eludir las reglas UFW, y asigné a los despliegues un usuario sin privilegios con una única orden fija, en vez de root.

Menor exposición en cada unidad systemd

Reduje la exposición de aislamiento systemd en todas las unidades que instala la plataforma — un servicio de monitorización por ausencia de señal pasó de 9,6 UNSAFE a 1,5 y una forja git expuesta a Internet de 8,3 EXPOSED a 1,5 — y añadí una comprobación del analizador durante la convergencia tras encontrar una directiva mal escrita que se ignoraba silenciosamente en tres plantillas de unidades.

Flujos CI/CD en GitHub Actions

Desarrollé flujos CI/CD en GitHub Actions con una imagen de producción distroless para la interfaz y promoción entre varios entornos.

Ver los 5 logros

{ Nube, infraestructura y DevOps }

Saberlo antes que sus clientes

Fiabilidad de sistemas y monitorización

Monitorización y alertas que avisan cuando un fallo aún es pequeño, y cifras que muestran con qué frecuencia se mantiene así.

Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

13 direcciones medidas sin rastreo

Medí trece direcciones únicamente a partir del registro de accesos del servidor web — sin cookies, rastreadores ni servicios de terceros —, enmascarando la red del cliente al escribir cada línea e incorporando el resultado a un archivo permanente de 21 recuentos diarios y 20 facetas mensuales.

Un flujo de eventos limitado para una aplicación macOS

Limité los búferes de eventos y la generación de registros tras un incidente de macOS documentado con un crecimiento de 41 MB por segundo y 111 GB de páginas comprimidas en una máquina de 36 GB; una ejecución completa de pruebas documentada redujo las líneas de registro del núcleo de 610.996 a 1.411.

Ver los 23 logros

{ Nube, infraestructura y DevOps }

Recuperarlo todo después de un mal día

Copias de seguridad y recuperación ante desastres

Copias de seguridad que se prueban en vez de darse por buenas y un plan escrito para la mañana en que un servidor no vuelva a encenderse.

Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

Automatización de copias con un método explícito de restauración

Implementé automatización de copias con restic, retención, comprobaciones de integridad y un flujo mensual de prueba de restauración; documenté la activación del almacenamiento y la verificación de restauración real como pasos operativos separados.

Copias y recuperación ante desastres como código

Configuré la retención de las copias de seguridad de la base de datos como infraestructura como código, audité la capacidad de recuperación y documenté el procedimiento de restauración, dejando explícitas las carencias pendientes para evitar descubrirlas durante un incidente.

Ver los 5 logros

{ Nube, infraestructura y DevOps }

Mantenerlo todo en funcionamiento

Administración de sistemas

El cuidado diario de servidores, correo, cuentas y actualizaciones para que los servicios de los que depende todo el mundo sigan funcionando sin sorpresas.

Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

Seis recolectores declaran lo que omitieron

Hice que seis recolectores de informes declararan la parte de sus entradas que nunca habían leído — archivos no abiertos, sondeos no instalados y líneas del diario no reconocidas — y ajusté los umbrales a 116 muestras medidas tras constatar que una recolección ciega y un servidor sano generan la misma página.

Ensayo del hook CI: 2 defectos detectados

Ensayé el hook CI del servidor de la forja y encontré dos defectos imposibles de alcanzar leyendo el archivo: una alternativa que introducía un argumento irresoluble en la entrada del hook y una variable de entorno de git que acompañaba al control hasta el checkout y hacía fallar 19 pruebas.

Ver los 28 logros

{ Nube, infraestructura y DevOps }

Conectar sus oficinas y a su personal

Configuración de redes y VPN

Wi-Fi, cortafuegos y acceso remoto configurados para trabajar desde cualquier lugar y permitir el acceso solo a las personas autorizadas.

Segundo servidor aprovisionado desde código

Automaticé el aprovisionamiento de un segundo servidor en un segundo proveedor de nube y creé el cortafuegos antes que la máquina para que naciera protegida, con los cortafuegos de ambos proveedores configurados directamente mediante sus API REST para evitar una colección de terceros.

Zona DNS de 20 registros como código

Concilié declarativamente una zona DNS de 20 registros con la API de Cloudflare mediante puntos de entrada separados para auditoría y exportación BIND, y desactivé el proxy CDN por motivos de privacidad después de implementarlo.

Reglas de cortafuegos verificadas por posición

Verifiqué las reglas del cortafuegos por su posición, además de por su existencia, leyendo la lista numerada de reglas y la cadena activa del filtro de paquetes, porque una regla presente puede no recibir nunca un paquete.

Ver los 11 logros

{ Nube, infraestructura y DevOps }

Decidir quién puede acceder a qué

Seguridad y gestión de accesos

Cuentas, contraseñas y permisos organizados para que cada persona tenga exactamente el acceso que necesita y lo pierda el mismo día que deja la empresa.

Espejo onion de Tor con una dirección legible

Publiqué el sitio como espejo onion de Tor en una dirección legible que comienza por engineer, con la clave generada fuera del servidor para que nunca llegara a este repositorio ni a un portátil, y lo mantuve como la única de las cuatro puertas de entrada que nunca se contabiliza.

13 direcciones medidas sin rastreo

Medí trece direcciones únicamente a partir del registro de accesos del servidor web — sin cookies, rastreadores ni servicios de terceros —, enmascarando la red del cliente al escribir cada línea e incorporando el resultado a un archivo permanente de 21 recuentos diarios y 20 facetas mensuales.

Conocer la licencia de todo lo que se publica

Audité la compatibilidad de licencias de 644 crates Rust en cada compilación y probé que el control se activa al cambiar la licencia de una crate y al modificar la revisión fijada del núcleo sin regenerar los datos.

Ver los 31 logros

{ Nube, infraestructura y DevOps }

Que su IA actúe solo donde le corresponde

Seguridad de agentes de IA

El acceso de cada agente definido y limitado a lo necesario, cada acción registrada donde el agente no puede alterarla y cada protección comprobada en lugar de darse por segura.

Secretos operativos separados por servidor

Separé los cuatro secretos operativos de cada servidor tras constatar que dos servidores que comparten una señal de actividad alertan menos que dos señales independientes, y que una contraseña compartida de copias de seguridad puede exponer las copias de ambos servidores, incluso en repositorios separados, si alguien la obtiene y puede leer los datos cifrados y los archivos de claves de ambos repositorios.

Regla de alcance tras una fuga de credenciales

Incorporé al repositorio una regla de alcance tras una reorganización que había introducido el inventario, los permisos de cortafuegos y textos de otra empresa, y mantuve los residuos en cuarentena bajo el detector de secretos, sin excluirlos.

Comprobación de secretos en un segundo

Añadí una comprobación previa que ejecuta en aproximadamente un segundo el mismo código de convergencia sobre secretos locales del operador, después de que una ejecución de producción parcialmente aplicada fallara en su novena tarea con los ajustes de intercambio ya escritos en el servidor activo.

Ver los 9 logros

Trabajamos con **las plataformas** que su empresa ya utiliza.



Apple Microsoft DigitalOcean AWS Google

Vea toda la pila tecnológica

Revisión de seguridad de los agentes de IA incluidos en el alcance

Inventariados

Un inventario acordado de los agentes, credenciales y accesos a datos incluidos en el alcance, conservado junto a su configuración.

Protegidos

Cada agente con identidad propia y privilegios mínimos; el texto que lee se trata como datos, nunca como órdenes.

Registrados

Cada acción registrada junto a la persona para la que se realizó, donde el agente no puede modificarla, y alertas ante lo inusual.

Vea cómo se protegen los agentes Pregunte por sus agentes →

Cuéntenos qué está construyendo.

Sin formulario, sin embudo, sin secuencia de ventas: solo un correo electrónico a una persona que lo leerá.

Inicie la conversación

Esta dirección muestra solo los trabajos de infraestructura. La empresa también construye software, trabaja con datos y asesora a equipos: esa parte se encuentra en engineer.company.

Vea todo lo que hace Engineer ApS →

Empiece con una mejora concreta

Un flujo de trabajo, un flujo de datos o un sistema que necesita mejorar puede ser una primera tarea útil. Podemos acordar su alcance juntos.

Vea cuánto cuesta Póngase en contacto →

Prepare la primera conversación

Indíquenos el resultado que necesita, qué existe ya y cuándo le gustaría tenerlo listo. Con esa información podemos hablar del siguiente paso.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

En el servidor, trabajamos con **Linux de código abierto** — sin licencias que pagar ni nada oculto.

Ubuntu Debian

Cómo operamos los servidores

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://platform.engineer.company/es/