Engineer

Infrastruktur

49 resultat

Koppla samman systemets delar

Arbetet under ”Infrastruktur” visar en del av ett fungerande system. Vi kan hjälpa er att koppla den delen till resten av verksamheten. Övervakning och larm som säger till medan ett fel fortfarande är litet, och siffrorna som visar hur ofta det förblir så.

Se våra tjänster Hör av er →

Hitta en jämförbar utmaning

Öppna ett exempel som liknar er situation. Genomgången förklarar besluten bakom resultatet, utöver teknikens namn. Byggde företagets första observerbarhetsdashboard — företagets första överblick över de egna systemen

Byggde företagets första observerbarhetsdashboard Hör av er →

En infrastrukturplan för 14 avdelningar

Utformade ett heltäckande infrastrukturramverk för ett forskningsinstitut som berörde 14 avdelningar, med flexibla moduler, enhetliga datapipelines och strukturerade supportstrategier för ett långsiktigt införande.

Byggde företagets första observerbarhetsdashboard

Utvecklade och lanserade företagets första dashboard för observerbarhet, med realtidsinsikt i systemens prestanda och datavisualisering på den stora tv:n på kontoret.

Drev 10 PostgreSQL- och MS SQL-servrar på Ubuntu

Utformade, driftsatte och underhöll 10 PostgreSQL- och MS SQL-servrar på VPS:er med Ubuntu Linux och säkerställde optimal serverprestanda och tillförlitlighet.

Skrev 1 000 RBAC-regler som säkrar Linux-servrar

Stärkte datasäkerheten genom att införa 1 000 RBAC-regler för utvecklare, applikationsinstanser, PostgreSQL, MS SQL och andra Linux-servrar, vilket förhindrade obehörig åtkomst; dokumenterade reglerna som automatisering i Ansible.

Automatiserade 100 kritiska säkerhetskopior

Automatiserade 100 kritiska säkerhetskopior av data med Barman, Google Cloud, Bash och Python och säkerställde dataintegriteten i samtliga databaser.

Drev 20 Docker-appar på Google Cloud och AWS

Driftsatte och underhöll 20 containeriserade Docker-applikationer, felsökte med Podman och Kubernetes och förvaltade R-baserade appar på Google Cloud och AWS.

Förvaltade 30 VPS-instanser med Ubuntu Linux

Förvaltade 30 VPS-instanser med Ubuntu Linux, införde strategier för katastrofåterställning och säkerställde optimala nätverkskonfigurationer.

Ledde behörighetshanteringen i M365 och SSO

Förhindrade säkerhetsintrång genom att leda initiativ för behörighetshantering med M365, 1Password, Red Hat SSO och OKTA SSO.

Minskade driftrisken med Grafana och Prometheus

Minskade driftriskerna genom att införa en dashboard för övervakning med Grafana och Prometheus, vilket förbättrade systemets tillförlitlighet.

Förvaltade och felsökte 8 VPN-anslutningar

Förvaltade och felsökte 8 VPN-anslutningar över WireGuard och IPsec och säkerställde säker kommunikation i Google Cloud och på Linux-system.

Förnyade interna processer, sparade 8 000 timmar

Gjorde en genomgripande översyn av de interna processerna och sparade 8 000 timmar genom att förbättra mjukvaruarkitekturen, systemen och effektiviteten i schemaläggningen.

Administrerade nätverksinfrastrukturen

Administrerade nätverksinfrastrukturen för över 1 000 servrar och säkerställde optimal driftsättning, säkerhet och felsökning av systemen.

Automatiserade SSL/TLS-cert för 100 Docker-appar

Automatiserade skapandet av SSL/TLS-certifikat för 100 Docker-applikationer och säkerställde säkra anslutningar på alla värdar med Ubuntu Linux.

Konfigurerade och driftsatte 1 000 Wi-Fi-routrar

Konfigurerade och driftsatte 1 000 Wi-Fi-routrar och förbättrade nätverkets tillgänglighet och prestanda för kunderna.

Administrerade 100 Bare Bone-servrar

Administrerade 100 Bare Bone-servrar, fysiska nätverk och system för IP-telefoni och säkerställde en robust infrastruktur för företagets tillväxt.

Administrerade 40 webbplatser på Apache och Nginx

Administrerade 40 webbplatser på värdservrar med Ubuntu Linux, Apache och Nginx och säkerställde hög tillgänglighet och prestanda.

En kartprodukt, buren hela vägen

Utformade, utvecklade och stödde infrastruktur, databearbetning och en kartapplikation under 2 år, med både programvaruleveranser och driftunderhåll.

Förenklad infrastruktur utanför AWS

Omarbetade en internationell kunds infrastruktur genom att ta bort oanvända tjänster, anpassa kapaciteten och flytta arbetslaster från AWS. Grundaren beskriver en minskning av molnkostnaderna med en faktor 10 utan förlust av produktivitet.

Provisionerade Azure-infrastruktur som kod

Provisionerade Azure-infrastruktur som kod med Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF och nätverk — i utvecklings-, staging- och produktionsmiljöerna.

Ansvarade för plattformens Azure-driftsättningar

Ansvarade för plattformens driftsättningar till Azure från början till slut och hanterade releaser i utvecklings-, staging- och produktionsmiljöerna.

Byggde säkerhetskopior och återställning som kod

Konfigurerade lagringstiden för databasens säkerhetskopior som infrastruktur som kod, granskade sedan återställningsförmågan och dokumenterade rutinen för återläsning — och pekade ut de återstående luckorna i stället för att låta dem upptäckas under en incident.

Gav support 24/7 för ca 1 000 IPTV/OTT-servrar

Gav infrastruktursupport dygnet runt, 24/7, för en IPTV/OTT-strömningsplattform och administrerade ca 1 000 servrar samt kundägda system för globala kunder i Kina, USA och Tyskland.

Höll strömmade IPTV-signaler igång oavbrutet

Säkerställde oavbruten leverans av strömmade IPTV-signaler mellan leverantörer och kunder och övervakade och underhöll strömningsnätet och IP-telefonin dygnet runt.

Byggde infrastruktur som ännu körs år senare

Planerade och implementerade ny infrastrukturfunktionalitet för interna och externa system och byggde lösningar som höll tillräckligt väl för att fortfarande vara i drift flera år senare med minimala ändringar.

Byggde hela kärninfrastrukturen från noll

Utformade och byggde, som en av de första anställda, hela kärninfrastrukturen och de stödjande processerna från noll för en SaaS-startup inom grön energi och lade grunden för snabb tillväxt.

Införde en företagsgemensam lösenordshanterare

Integrerade ett företagsövergripande system för lösenordshantering, vilket stärkte säkerheten och effektiviserade åtkomstkontrollen.

Byggde egen felövervakning och spårning

Byggde egen felövervakning och OpenTelemetry-spårning i stället för att köpa dem — sanering av nyttolaster, upptäckt av toppar och regressioner, symbolisering och ett syntetiskt hjärtslag — bakom 11 operatörsvyer.

Infrastruktur beskriven i ett kodarkiv

Byggde företagets egen infrastruktur som 19 Ansible-playbooks och 34 roller på 12 065 rader YAML, som konvergerar en värd i drift mot ett deklarerat tillstånd, där varje play är idempotent.

Offentliga tjänster på en liten server

Drev företagets offentliga tjänster på en värd med en kärna och 512 MB — en git-forge, en webbserver för sju domäner, Tor, två servrar för alternativa protokoll och intrångsblockering — med 464 MB användbart minne som designbegränsning; backupautomatik är en separat implementation.

Tre lås på dörren, inget av dem låst

Hittade och åtgärdade tre skydd mot brute force-attacker över SSH som aldrig hade fungerat: en jail som bevakade port 22 medan daemonen lyssnade på 1986, en hastighetsbegränsning som skuggades av en bredare regel ovanför den och en spärråtgärd vars binärfil aldrig gick att hitta, så att ingen spärr någonsin hade tillämpats.

Härdade SSH till 24 validerade direktiv

Härdade SSH till 24 kontrollerade direktiv med validering i tre steg — kandidatfilen, den sammansatta konfigurationen och sedan daemonens egen återläsning — efter att återläsningen hade avslöjat att den körande servern tyst åsidosatte två av de tjugofyra.

Bevisade att intrångsspärren fungerar varje gång

Bevisade spärrkedjan mot intrång från början till slut vid varje härdningskörning genom att spärra en reserverad testadress, läsa den kärnregel som uppstod och häva spärren i ett städblock som alltid körs, så att en jail som slutar fungera fäller körningen i stället för att rapportera att allt är i ordning.

Verifierade brandväggsregler efter position

Verifierade brandväggsregler efter position snarare än förekomst och läste den numrerade regellistan och den aktiva paketfilterkedjan, eftersom en regel som finns inte är detsamma som en regel som något paket når.

Backupautomatik med en tydlig återläsningsmetod

Implementerade restic-backupautomatik med lagringspolicy, integritetskontroll och ett flöde för månatliga återläsningstest; dokumenterade aktivering av lagring och kontroll av en verklig återläsning som separata driftsteg.

Byggde värdövervakning med dödmansgrepp

Byggde övervakning med dödmansgrepp som pingar bara så länge minne och disk är i gott skick, så att en försämrad värd larmar genom att tystna — och upptäckte sex variabelnamn som sa ”free” där kontrollen korrekt mätte ”available”, en storleksordning isär på en maskin med 464 MB.

Fick Ansibles check mode att tala sanning

Fick check mode att tala sanning i hela plattformen efter att ha hittat sex sonder som fattade beslut utifrån ett värde som värddatorn aldrig gav, eftersom command-modulen i Ansible rapporterar framgång under --check medan den hoppar över kommandot helt.

Kontrollerade hemligheter i förväg på en sekund

Lade till en play för förhandskontroll som kör samma kod som konvergeringen mot operatörens lokala hemligheter på ungefär en sekund, efter att en halvt tillämpad produktionskörning kraschade vid sin nionde uppgift med swapinställningarna redan skrivna till värden i drift.

Stämde av en DNS-zon med 20 poster som kod

Stämde av en DNS-zon med 20 poster deklarativt mot Cloudflare-API:t med separata ingångar för granskning och BIND-export, och stängde av CDN-proxyn igen av integritetsskäl efter att ha byggt den.

Minskade systemd-exponeringen för varje enhet

Minskade exponeringen i systemd-sandlådan för varje enhet som plattformen installerar — en tjänst för dödmansgrepp från 9,6 UNSAFE till 1,5, en internetexponerad git-forge från 8,3 EXPOSED till 1,5 — och lade till en parserkontroll vid konvergering efter att ha hittat ett felstavat direktiv som tyst ignorerades i tre enhetsmallar.

Byggde 7 skrivskyddade rapportroller för värdar

Byggde sju skrivskyddade rapportroller som återger en värd i drift som Markdown — fakta, åtkomst, git, mätvärden, trafik, säkerhet och leverantörsinventering — under regeln att inget tal skrivs ut som körningen inte har mätt.

Er kod på er egen server

Driftsatte företagets egen git-forge på Soft Serve, privat som standard, utan webbpanel och med SSH-porten bunden till loopback bakom en hoppserver, och gjorde landningssidan framför den till en byggartefakt från huvudwebbplatsen i stället för en manuellt underhållen kopia.

Körde containrar på Podman och Quadlet

Driftsatte en containermiljö på Podman och Quadlet under systemd i stället för Docker, eftersom Dockers standardpublicering av portar i bridge-nätverk kan kringgå UFW-regler — och gav driftsättningarna en oprivilegierad användare med ett enda fast kommando i stället för root.

Provisionerade en andra server från kod

Automatiserade provisioneringen av en andra server hos en andra molnleverantör och skapade brandväggen före maskinen så att den föds bakom en, med båda leverantörernas brandväggar skrivna direkt mot deras REST-API:er för att undvika en collection från tredje part.

Skrev en avgränsningsregel efter läckta uppgifter

Skrev in en avgränsningsregel i repot efter att en omstrukturering hade fört in ett annat företags värdförteckning, brandväggsundantag och text i det — och höll de karantänsatta resterna kvar under skanningen efter hemligheter i stället för att undanta dem.

Delade upp varje driftshemlighet per värd

Delade upp de fyra driftshemligheterna per värd efter att ha konstaterat att två värdar som delar ett dödmansgrepp larmar mindre än två separata grepp, inte mer, och att en delad lösenfras för säkerhetskopior kan ge tillgång till båda värdarnas kopior, även i separata repon, om någon får tag på den och kan läsa båda reponas krypterade data och nyckelfiler.

Speglade webbplatsen till Gemini och Gopher

Byggde Gemini- och Gopher-utdata från samma webbkälla; resultatbeskrivningen anger 777 dokument i varje format vid det skedet och noll ändrade HTML-byte genom tillägget av formaten.

Repeterade forgens CI-hook och hittade 2 fel

Repeterade CI-hooken på forge-sidan och hittade två fel som inte gick att se genom att läsa filen: en reservväg som lade ett argument som inte gick att lösa upp i hookens indata, och en miljövariabel från git själv som följde med grinden in i utcheckningen och gjorde 19 tester röda.

Mätte 13 adresser helt utan spårning

Mätte tretton adresser enbart utifrån webbserverns egen åtkomstlogg — ingen cookie, ingen spårare och ingen tjänst från tredje part — maskerade klientens nätverk när varje rad skrivs och förde in resultatet i ett permanent arkiv med 21 dagliga räkningar och 20 månatliga fasetter.

Publicerade en läsbar onion-spegel i Tor

Publicerade webbplatsen som en onion-spegel i Tor på en läsbar adress som börjar med engineer, med nyckeln genererad utanför värden så att den aldrig nådde detta repo eller en bärbar dator, och lät den förbli den enda av fyra ytterdörrar som aldrig räknas.

Fokusera den tekniska insatsen

Ta med driftsättningsvägen, driftsriskerna och dem som har jour. Vi kan avgränsa en förändring med kontroller, en väg tillbaka och tydligt ansvar.

Se vad det kostar Hör av er →

Sätt ämnet i ert sammanhang

Berätta hur ”Infrastruktur” ingår i ert system. Vi kan hjälpa er att välja rätt tjänst och avgränsa arbetet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/categories/infrastructure/