Engineer Plattform

{ Infrastruktur som kod }

Servrar du kan bygga upp igen från versionshanterad konfiguration

Hela uppsättningen nedskriven som filer, så att den kan återskapas exakt och aldrig bara finns i en enda persons huvud.

Håller ni fortfarande på att reda ut vad ni behöver?

Berätta vad som inte fungerar eller vad ni försöker bygga. Vi säger vad det skulle kräva, ungefär vad det skulle kosta och om vi är rätt personer för uppgiften.

Från 1 400 DKK i timmen, exklusive moms — och mindre per timme vid en längre bokning.

Hör av er Se vad det kostar

Utvalt arbete som visar den här tjänsten.

Gör tjänsten till ett resultat

Låt ”Infrastruktur som kod” stödja ett affärsmål. Vi kan knyta det tekniska arbetet till användarna, driften och besluten det ska stödja. Hela uppsättningen nedskriven som filer, så att den kan återskapas exakt och aldrig bara finns i en enda persons huvud.

Hör av er

Granska det relevanta arbetet

Använd arbetet på sidan för att granska vårt angreppssätt. Fråga om avvägningarna och förutsättningarna bakom varje resultat. Skrev 1 000 RBAC-regler som säkrar Linux-servrar — 1,000 åtkomstregler, skrivna för att täppa till en säkerhetslucka

Skrev 1 000 RBAC-regler som säkrar Linux-servrar Hör av er →

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

Letar ni efter Infrastruktur som kod?

Använd de länkade arbetena för att bedöma relevant erfarenhet. Berätta vad ni behöver bygga; vi kan diskutera omfattning, arbetssätt och överlämning.

Hör av er Hela portföljen

Avgränsa uppdraget före beställning

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 →

Gå igenom era krav med oss

Berätta vad ni behöver inom ”Infrastruktur som kod”, vilken tidsram ni önskar och vilket team som ska förvalta lösningen. Vi kan diskutera en praktisk väg framåt.

Hör av er

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://platform.engineer.company/sv/services/infrastructure-as-code/