Engineer

# Infrastruktur in einem Repository definiert

Die eigene Unternehmensinfrastruktur als 19 Ansible-Playbooks und 34 Rollen mit 12.065 YAML-Zeilen aufgebaut, die einen laufenden Host mit ausschließlich idempotenten Plays in den deklarierten Sollzustand überführen.

August 2026

## Wie die Plattformkonfiguration angewendet wird

Überblick über die Implementierung der Bereitstellung; der Wiederaufbau hängt von Zugangsdaten und Voraussetzungen beim Anbieter ab.

1.

### Deklarierte Konfiguration

Host-Ebenen und Gruppenvariablen legen fest, welche Konfiguration jede Maschine erhalten soll.

2.

### Ansible-Rollen

Ansible-Rollen wenden die deklarierte Konfiguration auf die ausgewählten Hosts an.

3.

### Dienstdefinitionen

Systemd-Units und Dienstdefinitionen beschreiben, wie die bereitgestellten Dienste ausgeführt werden.

**19**

Playbooks führen einen laufenden Host in den Sollzustand

**Situation.** Engineer ApS pflegt die eigene Infrastruktur als Code. Das Repository umfasst bereitgestellte öffentliche Dienste und die Konfiguration weiterer Komponenten, darunter Sicherungs- und Mailautomatisierung, deren betriebliche Aktivierung ein getrennter Schritt ist. Dieser in Code festgehaltene Umfang macht die Einrichtung überprüfbar statt erinnerungsabhängig.

**Aufgabe.** Die gesamte Umgebung musste in einem Repository statt in einem Kopf beschrieben sein. Die Beschreibung sollte einen echten Host in den Sollzustand überführen, nicht lediglich dokumentieren.

**Massnahmen.** Die dokumentierte Implementierung umfasst 19 Playbooks und 34 Rollen mit 12.065 YAML-Zeilen. Diese Zahlen beschreiben den damals dokumentierten Stand, nicht die heutige Repositorygröße. Wichtiger als die Größe ist ihre Struktur. Die Zusammensetzung ist als Daten statt als Schalter festgelegt: Ein Host gehört zu einer Tier-Gruppe, deren Variablen seine Rollen angeben. Provisionierung ohne Argumente führt deshalb jeden Host in seinen deklarierten Zustand. Idempotenz ist eine Verpflichtung statt eines Wunsches: Ein bereits abgeglichener Host meldet null Änderungen; ein Play, das dies nicht leisten kann, ist nicht fertig. Vier Befehlsnamensräume halten die Zusagen getrennt: eine Qualitätsprüfung ohne Hostkontakt, Berichte mit ausschließlich lesendem Hostzugriff, Provisionierung zum Anpassen des Hosts an das Repository und Verifikation, die einen Host gezielt verändert und ein Urteil liefert.

**Ergebnis.** Das Repository liefert die Konfiguration zum Neuaufbau der deklarierten Umgebung. Was zuvor nur aufgrund persönlicher Erinnerung stimmte, wird nun durch Prüfungen abgesichert, die einen Lauf scheitern lassen. Der Preis ist real: Jede Änderung dauert länger als die direkte Bearbeitung einer Serverdatei, und ein halb ausgeführter Konvergenzlauf ist schlimmer als eine frühzeitige Verweigerung. Deshalb wurde später ein Preflight-Play vorgeschaltet. Diese Abwägung wurde bewusst getroffen und hat Bestand.


## Diese Erfahrung für Ihr Projekt nutzen

Nutzen Sie „Infrastruktur in einem Repository definiert“ als Ausgangspunkt für eine ähnliche Herausforderung in Ihrer Organisation. Die Leistungen hinter dieser Arbeit stehen auch für Ihr Projekt zur Verfügung. Server werden nach dem tatsächlichen Bedarf dimensioniert und in die Cloud oder aus ihr heraus verlagert — die monatlichen Kosten stehen vorher fest.

**Unsere Leistungen entdecken** Kontakt aufnehmen →

## Hinter das Ergebnis schauen

Lesen Sie die Situation, die Entscheidungen und das Ergebnis dieses Beispiels. Wir können besprechen, welche Teile sich auf Ihre Rahmenbedingungen übertragen lassen und welche einen anderen Ansatz erfordern. Infrastruktur in einem Repository definiert — 19 Playbooks führen einen laufenden Host in den Sollzustand

**Unsere Leistungen entdecken** Kontakt aufnehmen →

## Sollen Ihre Server in Code statt nur im Kopf beschrieben sein?

Besprechen Sie Ihr Projekt mit der Ingenieurin, die für die Umsetzung verantwortlich ist. Wir vereinbaren Umfang, Preisgrundlage und Übergabe schriftlich, bevor die Arbeit beginnt.

Rollenkontext: Ingenieurin für Infrastrukturautomatisierung. Die Arbeit baut auf Plattformarchitektur, DevOps, Infrastruktur, Automatisierung und CI/CD, Zuverlässigkeit und Sicherungen, Linux und Server, Systemadministration und Dokumentation. Und wenn Sie schon hier sind: Wir sind auch ziemlich gut in Cloud-Infrastruktur und Cloud-Migration, Infrastructure as Code, DevOps- und CI/CD-Automatisierung und Systemadministration.

**Kontakt aufnehmen** Unsere Leistungen entdecken

## Ein Teil einer **längeren Arbeitsbilanz** — vollständig auf dieser Website zu finden.

Jeder Eintrag folgt demselben Aufbau: die Situation, die Aufgabe, unser Vorgehen und die Veränderungen.

**Alle Arbeiten ansehen**

### Die benötigte Verbesserung definieren

Beschreiben Sie den Bereitstellungsweg, die Betriebsrisiken und die Bereitschaftsverantwortlichen. Wir können eine Änderung mit Prüfungen, einem Wiederherstellungsweg und klaren Zuständigkeiten abgrenzen.

**Preise ansehen** Kontakt aufnehmen →

# Sollen Ihre Server in Code statt nur im Kopf beschrieben sein?

Fragen Sie nach den Entscheidungen hinter „Infrastruktur in einem Repository definiert“. Sie sprechen mit den Menschen, die die Arbeit erklären und Ihr eigenes Projekt einschätzen können.

**Unsere Leistungen entdecken**

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

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

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/portfolio/servers-you-can-rebuild-from-scratch/