**Køreplan for Hosting af Cognify Learning Suite på Google Cloud**

Denne plan beskriver en moderne, sikker og skalerbar metode til at deployere og hoste jeres webapplikation ved hjælp af Google Clouds serverless-teknologier.

**Fase 1: Klargøring og Infrastruktur**

Her skaber vi det fundament, applikationen skal leve på. Vi vælger en serverless tilgang for at sikre automatisk skalering og reducere driftsomkostninger.

1. **Opret et Google Cloud Project:**
   * Alt starter med et dedikeret projekt i dit Google Cloud Console. Dette isolerer ressourcer, fakturering og tilladelser for platformen.
   * **Anbefaling:** Navngiv det f.eks. cognify-learning-suite-prod.
2. **Valg af Hosting Service: Cloud Run**
   * Vi bruger **Cloud Run** til at hoste applikationen. Cloud Run er en fuldt administreret, serverless platform, der automatisk skalerer op og ned – endda helt ned til nul, så du ikke betaler for inaktiv tid.
   * **Hvorfor Cloud Run?** Det er ideelt for denne applikation, da det fjerner behovet for at administrere servere, og det er bygget på containere, hvilket sikrer et konsistent og sikkert kørselsmiljø.
3. **Containerisering med Docker:**
   * For at køre på Cloud Run skal vi pakke applikationens kode (HTML, CSS, JS) ind i en **Docker-container**. Dette er en standardiseret "kasse", der indeholder alt, hvad applikationen behøver for at køre.
   * Vi opretter en Dockerfile-fil i projektets rod, der beskriver, hvordan containeren bygges. Den vil typisk bruge en letvægts-webserver som Nginx til at servere de statiske filer.
4. **Opbevaring af Container: Artifact Registry**
   * Når Docker-containeren er bygget, skal den gemmes et sikkert sted. Vi bruger **Artifact Registry** til at opbevare vores container images. Det fungerer som et privat, sikkert bibliotek for jeres applikations-versioner.

**Fase 2: Automatiseret Deployment Pipeline (CI/CD)**

Her automatiserer vi processen fra kodeændring til live-implementering. Dette sikrer hurtige, pålidelige og sporbare opdateringer.

1. **Source Control (Kildekode-styring):**
   * Al kode til applikationen skal ligge i et Git-repository, f.eks. på **GitHub** eller **Google Cloud Source Repositories**.
2. **Automatisk Byg og Deployment: Cloud Build**
   * Vi opsætter **Cloud Build** til at skabe en automatisk CI/CD-pipeline. Cloud Build kan konfigureres til at lytte efter ændringer i jeres Git-repository (f.eks. på main-branchen).
   * **Processen:**
     1. En udvikler pusher en ændring til Git.
     2. Cloud Build starter automatisk.
     3. Den bygger Docker-containeren baseret på Dockerfile.
     4. Den uploader den nye container til Artifact Registry.
     5. Den implementerer den nye container-version på Cloud Run.
   * Hele denne proces defineres i en cloudbuild.yaml-fil.

**Fase 3: Sikkerhed og Konfiguration**

Dette er den mest kritiske fase, især med din fokus på datasikkerhed.

1. **Sikker Håndtering af API-nøgle: Secret Manager**
   * **Vigtigt:** Gemini API-nøglen må **ALDRIG** være skrevet direkte i koden.
   * Vi bruger **Secret Manager** til at opbevare API-nøglen sikkert. Secret Manager er en central, krypteret tjeneste til hemmeligheder.
   * Under deployment-processen i Cloud Build konfigurerer vi Cloud Run-tjenesten til at hente nøglen fra Secret Manager og gøre den tilgængelig for applikationen som en **miljøvariabel**. Dette er best practice og sikrer, at nøglen aldrig eksponeres.
2. **Eget Domæne og SSL:**
   * Vi opsætter en **Global External HTTPS Load Balancer** foran vores Cloud Run-tjeneste.
   * Dette giver os mulighed for at tilknytte et brugerdefineret domæne (f.eks. app.cognify-suite.dk).
   * Load balancere inkluderer **gratis, automatisk fornyede Google-managed SSL-certifikater**, hvilket sikrer, at al trafik er krypteret (HTTPS).
3. **Adgangskontrol (IAM):**
   * Vi bruger **Identity and Access Management (IAM)** til at definere præcist, hvem der har tilladelse til at administrere de forskellige dele af cloud-opsætningen, hvilket minimerer risikoen for uautoriserede ændringer.

**Opsummering**

Ved at følge denne plan opnår vi en professionel cloud-arkitektur, der er:

* **Sikker:** Med hemmeligheder adskilt fra kode og krypteret trafik.
* **Skalerbar:** Klar til at håndtere alt fra én til tusindvis af brugere uden manuel indgriben.
* **Automatiseret:** Med en CI/CD-pipeline, der gør opdateringer simple og pålidelige.
* **Omkostningseffektiv:** Ved kun at betale for de ressourcer, der rent faktisk bruges.

Dette er det tekniske fundament, der matcher den høje kvalitet af selve Cognify Learning Suite. Skal jeg gå videre og udarbejde de specifikke konfigurationsfiler (Dockerfile og cloudbuild.yaml) til dig?