# Plan de Automatizare a Implementării (CI/CD)

**Obiectiv:** Crearea unui flux de lucru complet automatizat pentru a testa, compila și implementa aplicația "Sparky's Workshop Factory" în Google Cloud, eliminând nevoia de implementări manuale.

**Tehnologie Principală:** **GitHub Actions**

### **1. Filozofia de Bază: De la Manual la Automat**

În loc ca un dezvoltator să ruleze manual comenzi în terminal, vom crea "rețete" (workflows) pe care serverele GitHub le vor executa automat.

**Fluxul de lucru va fi următorul:**

1. Un dezvoltator face o modificare în cod.
2. Împinge modificarea pe GitHub în ramura principală (main).
3. GitHub detectează automat această modificare și pornește "linia de asamblare".
4. Aplicația este testată, compilată și implementată în Google Cloud.
5. Dezvoltatorul primește o notificare de succes sau de eșec.

### **2. Pregătirea: Conectarea Securizată între GitHub și Google Cloud**

Aceasta este cea mai importantă etapă de configurare.

* **Crearea unui Cont de Serviciu în Google Cloud:** Vom crea un cont de serviciu special pentru GitHub, cu permisiuni limitate de a implementa Funcții Cloud și de a scrie în Cloud Storage.
* **Generarea unei Chei JSON:** Vom genera o cheie de autentificare pentru acest cont.
* **Stocarea Cheii ca "Secret" în GitHub:** Conținutul acestei chei JSON nu va fi niciodată stocat în codul nostru. Va fi adăugat în secțiunea Settings > Secrets and variables > Actions a repository-ului GitHub, sub un nume precum GCP\_SA\_KEY. GitHub va folosi acest secret pentru a se autentifica în numele nostru în Google Cloud.

### **3. Planul de Automatizare (Workflow-uri)**

Vom crea două fișiere de configurare separate în directorul .github/workflows/ al proiectului nostru.

#### **3.1. Workflow 1: Implementarea Frontend-ului (deploy-frontend.yml)**

Acest flux de lucru se va declanșa de fiecare dată când există o modificare în folderul frontend/.

**Pașii executați de GitHub:**

1. **Checkout Code:** Descarcă cea mai recentă versiune a codului.
2. **Setup Node.js:** Pregătește mediul de lucru pentru o aplicație React.
3. **Install Dependencies:** Rulează npm install pentru a instala toate pachetele necesare.
4. **Build Application:** Rulează npm run build pentru a compila aplicația React într-un set de fișiere statice în directorul build/.
5. **Authenticate to Google Cloud:** Folosește "secretul" GCP\_SA\_KEY pentru a se autentifica.
6. **Deploy to Cloud Storage:** Rulează comanda gcloud storage cp -r build/\* gs://NUME\_BUCKET\_UNIC pentru a încărca noua versiune a site-ului.

#### **3.2. Workflow 2: Implementarea Backend-ului (deploy-backend.yml)**

Acest flux de lucru se va declanșa de fiecare dată când există o modificare într-unul din folderele funcțiilor (ex: functions/generate-text/).

**Pașii executați de GitHub:**

1. **Checkout Code:** Descarcă cea mai recentă versiune a codului.
2. **Setup Python:** Pregătește mediul de lucru pentru funcțiile noastre Python.
3. **Authenticate to Google Cloud:** Se autentifică folosind același "secret" GCP\_SA\_KEY.
4. **Deploy Cloud Function:** Rulează comanda gcloud functions deploy ... pentru funcția specifică care a fost modificată. Acest pas se poate repeta pentru fiecare funcție, sau poate fi configurat să implementeze doar funcțiile care au suferit modificări.

### **4. Noua Realitate a Dezvoltatorului**

După implementarea acestui sistem CI/CD, procesul de actualizare a aplicației devine extraordinar de simplu și sigur:

* **Scenariu:** Un utilizator raportează o mică greșeală de text în interfață.
  1. **Vechiul Proces:** Dezvoltatorul corectează textul, rulează npm run build, deschide terminalul, se autentifică manual în gcloud, rulează comanda gcloud storage cp..., verifică dacă a funcționat. Timp: 10-15 minute, cu risc de eroare manuală.
  2. **Noul Proces (CI/CD):** Dezvoltatorul corectează textul, scrie git commit -m "Corectat text", apoi git push. Asta e tot. În următoarele 3-5 minute, GitHub Actions face restul automat.

Acest plan de automatizare transformă proiectul nostru într-o operațiune profesională, permițând actualizări rapide, sigure și frecvente, esențiale pentru succesul pe termen lung al oricărui produs software.