Perfect, asta e exact abordarea corectă pentru cineva care vrea să crească! 💪  
Hai să-ți ofer un **ghid clar în pași**, fără cod, dar suficient de detaliat încât să știi ce ai de făcut.

**🔧 PROIECT: Log Parser & Forwarder (DevOps Simulation)**

**🔹 Obiectiv:**

Construiește un mini-sistem care:

1. **Parsează loguri** și extrage doar liniile importante (ERROR, CRITICAL).
2. **Trimite logurile** în batch-uri de câte 10 către un **API personalizat**.
3. Este complet automatizabil și containerizat, simulând un sistem real de DevOps/Observability.

**🧭 Faze și Pași**

**🟦 1. Log Parser în Python**

**Scop:** Script care:

* Citește fișierul de log (app.log).
* Filtrează doar liniile ce conțin „ERROR” sau „CRITICAL”.
* Le împarte în grupuri de câte 10.
* Le trimite către un endpoint /logs.

**Tehnologii:** Python + requests + json

📝 Pași:

* Creează scriptul de citire și filtrare.
* Creează o funcție care împarte lista în batch-uri.
* Creează o funcție care trimite un batch la API (POST request).
* Testează local cu un endpoint fals (ex: httpbin.org/post).

**🟩 2. Mini API în Node.js**

**Scop:** Un server care primește datele (loguri) și le scrie într-un fișier JSON sau le afișează.

**Tehnologii:** Node.js + Express

📝 Pași:

* Creează un endpoint POST /logs.
* Asigură-te că primește un JSON cu cheie logs: [linii...].
* Fie le salvează într-un fișier, fie le afișează în terminal.
* Testează cu Postman.

**🟨 3. Dockerizare + Docker Compose**

**Scop:** Rulezi parserul și API-ul în containere separate.

📝 Pași:

* Creează Dockerfile pentru parser și pentru API.
* Creează docker-compose.yml cu 2 servicii: parser și api.
* Montează fișierul app.log ca volum în containerul parserului.
* Rulează tot stack-ul și verifică dacă logurile ajung la API.

**🟥 4. Automatizare cu Jenkins**

**Scop:** Rulezi parserul automat prin Jenkins.

📝 Pași:

* Creează un Jenkinsfile:
  + Clonează repo-ul.
  + Rulează docker-compose up.
* Rulează-l fie pe un Jenkins local, fie GitHub + Jenkins Pipeline as Code.

**🟦 5. Automatizare cu Ansible (opțional)**

**Scop:** Simulezi provisioning și rulare automatizată.

📝 Pași:

* Scrie un playbook care:
  + Instalează dependințe.
  + Copiază fișierele de parser/API pe mașină remote/local.
  + Pornește serviciile cu Docker Compose.

**💡 Bonus: Extensii**

* Adaugă Prometheus + Grafana pentru a monitoriza statusul trimiterilor.
* Fă retry logic în parser dacă API-ul nu răspunde.
* Creează alerte dacă logurile conțin cuvinte cheie grave.

**📂 Structura finală**

bash

CopiazăEditează

log-forwarder/

├── api/ # Node.js

├── parser/ # Python

├── docker-compose.yml

├── jenkins/

├── ansible/

└── README.md # documentația proiectului tău

Dacă vrei, îți pot da și **un template de README** pe care să-l incluzi în portofoliu sau GitHub. Spune-mi când ești gata pentru pasul următor sau dacă vrei să discutăm despre deployment sau extensii.