Comparison: AWS Pipeline vs. ColPali Pipeline

Feature	AWS Pipeline (Textract -> Kendra)	ColPali Pipeline (DuckDB -> Neo4j -> ColPali)
PDF Extraction	Textract (OCR-based, highly accurate)	pdfplumber (good for tables, lacks OCR)
Text Processing	Comprehend (AWS NLP)	Pandas + DuckDB (in-memory querying)
Search Mechanism	Kendra (Enterprise Search with ML)	Neo4j (Graph Search) + ColPali (Semantic Search)
NLP & Query Understanding	Lex (Conversational AI)	Custom Query Processor + ColPali
Gen AI / LLM	SageMaker (Fine-tuned models)	Hugging Face (ColPali for Retrieval-Augmented Search)
Infrastructure	Fully managed (AWS-hosted)	Local-first (Runs on GPU/CPU)
Cost	Pay-as-you-go (AWS pricing applies)	Free for local use (compute cost for GPUs)
Customization	Limited to AWS tools	Full control over models & indexing

---

Strengths & Weaknesses

Why AWS (Textract + Kendra + SageMaker)?

- Highly scalable (no need to manage infrastructure)
- Textract is state-of-the-art for OCR
- Kendra is powerful for search over large corpora
- Best for enterprise cloud-based solutions

Limitations of AWS:

- Expensive (AWS services charge per request/usage)
- Less control over search & retrieval mechanisms

Why ColPali + Neo4j + DuckDB?

- More control over data processing
- Local-first (runs on your infrastructure)
- Lower cost if running on-premise

Limitations of ColPali:

- More setup required (manual implementation)
- Needs GPU for best performance