OpenSearchCon EUROPE

Building RAG with OpenSearch ML Plugins: From PDFs to Voice-Enabled Search

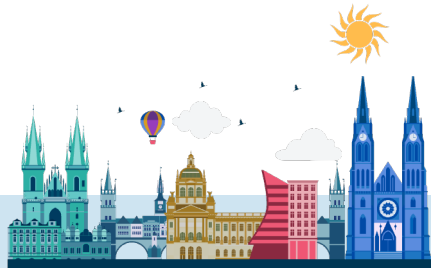
Kushagra Sharma

Staff Software Engineer @ Genesys

What You'll Learn Today

OpenSearchCon EUROPE

- Deploy embedding models directly on OpenSearch using ML Commons
- Process PDFs and generate embeddings within OpenSearch
- Build semantic search with knn_vector indices
- Integrate voice search using STT/TTS models
- Create a complete RAG system



Why OpenSearch ML for RAG?

OpenSearchCon EUROPE

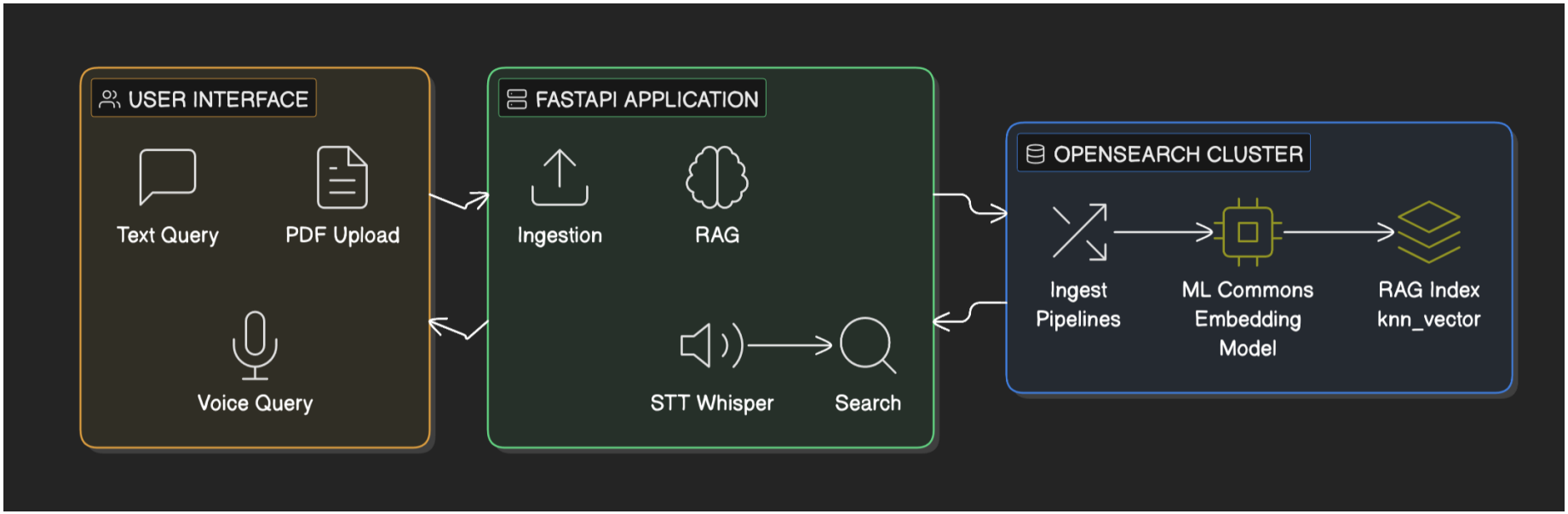
The Problem with Traditional RAG

- External Embedding APIs
- Vendor Lock-in
- Operational Overhead

OpenSearch ML Solution

- Native ML Capabilities
- Self-Contained System
- Production Ready

Architecture Overview



OpenSearch ML Commons

OpenSearchCon EUROPE

What is ML Commons?

ML Commons enables deploying and running
ML models directly on OpenSearch cluster

- Native model inference
- No external APIs required
- Integrated with OpenSearch ecosystem
- Production-ready and scalable

Supported Model Formats

- TorchScript (PyTorch)

- ONNX (Open Neural Network Exchange)

- Both formats supported for flexibility

Model Types Supported

- Text Embedding Models
  - Sentence transformers
  - BERT-based models
  - Multilingual models

- Sparse Encoding Models
  - For keyword-based search

- Cross-Encoder Models
  - For re-ranking results

Deploying an Embedding Model: Step-by-Step

OpenSearchCon EUROPE

Export Model to TorchScript/ ONNX

Download from Hugging Face
Convert to supported format
Package with tokenizer

Register Model in OpenSearch

Upload model ZIP file
Provide model metadata
Calculate SHA256 checksum

Deploy to Cluster Nodes

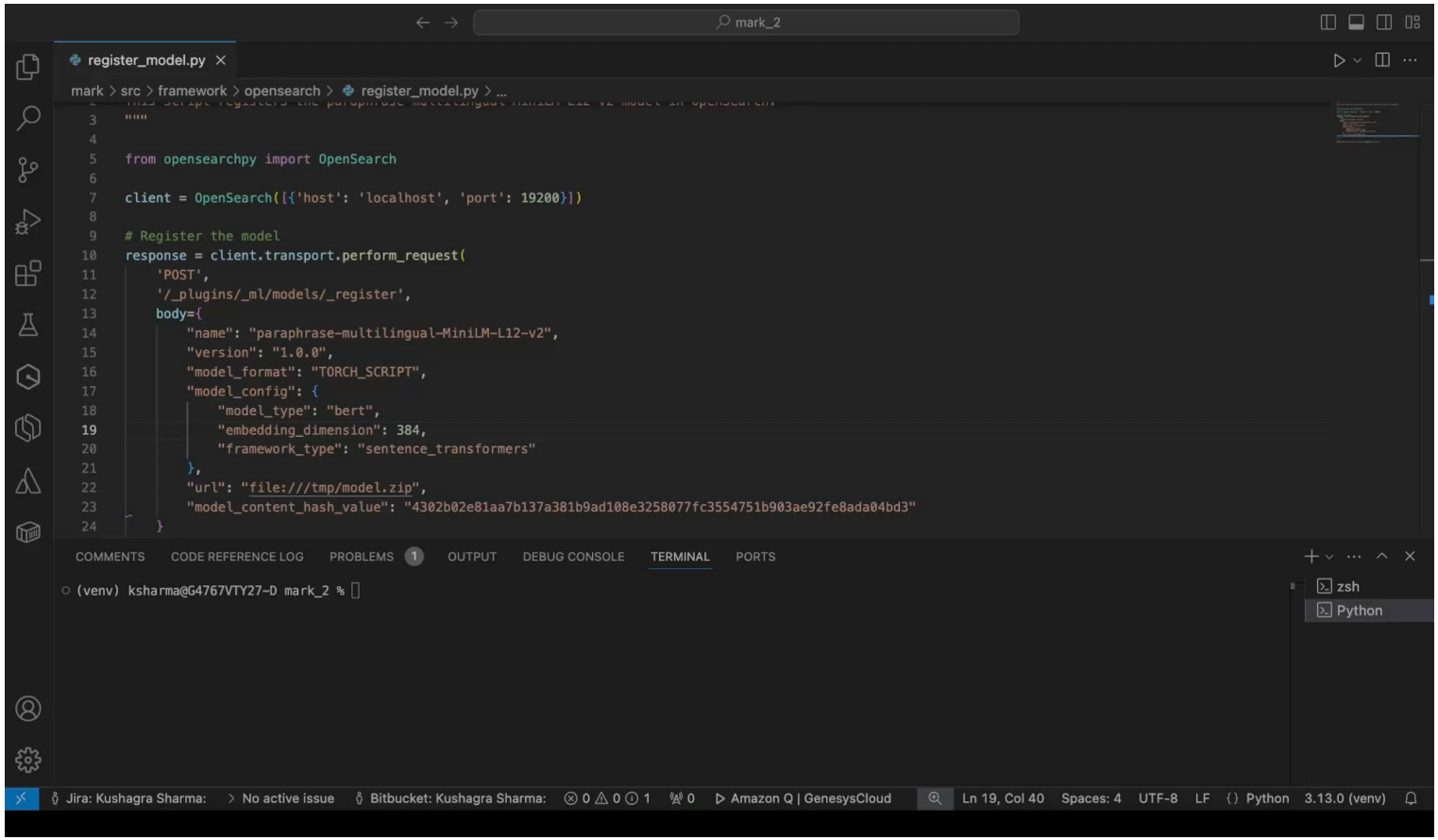
ML Commons loads model
Model available for inference
Verify deployment status

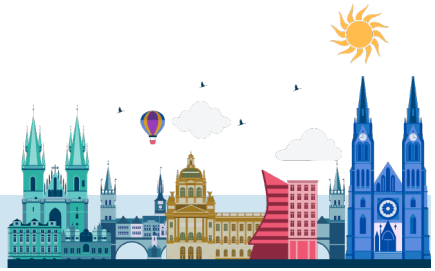
Generate Embeddings via API

POST /_plugins/_ml/_predict/text_embedding/{model_id}
Input: text documents
Output: vector embeddings

Model Deployment: Demo

OpenSearchCon EUROPE

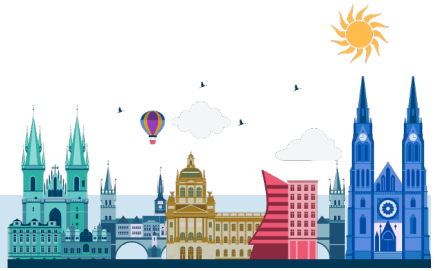




Document Ingestion Pipeline

OpenSearchCon EUROPE

PDF Processing	Text Chunking	Embedding Generation
• Extract text from PDFs • Use ingest-attachment plugin • Supports multiple formats • Handles complex documents	• Split documents into chunks • Configurable chunk size (500 words) • Overlap between chunks (50 words) • Preserve context across chunks	• Generate embeddings using ML model • Use deployed model via ML API • 384-dimensional vectors • Index with document metadata



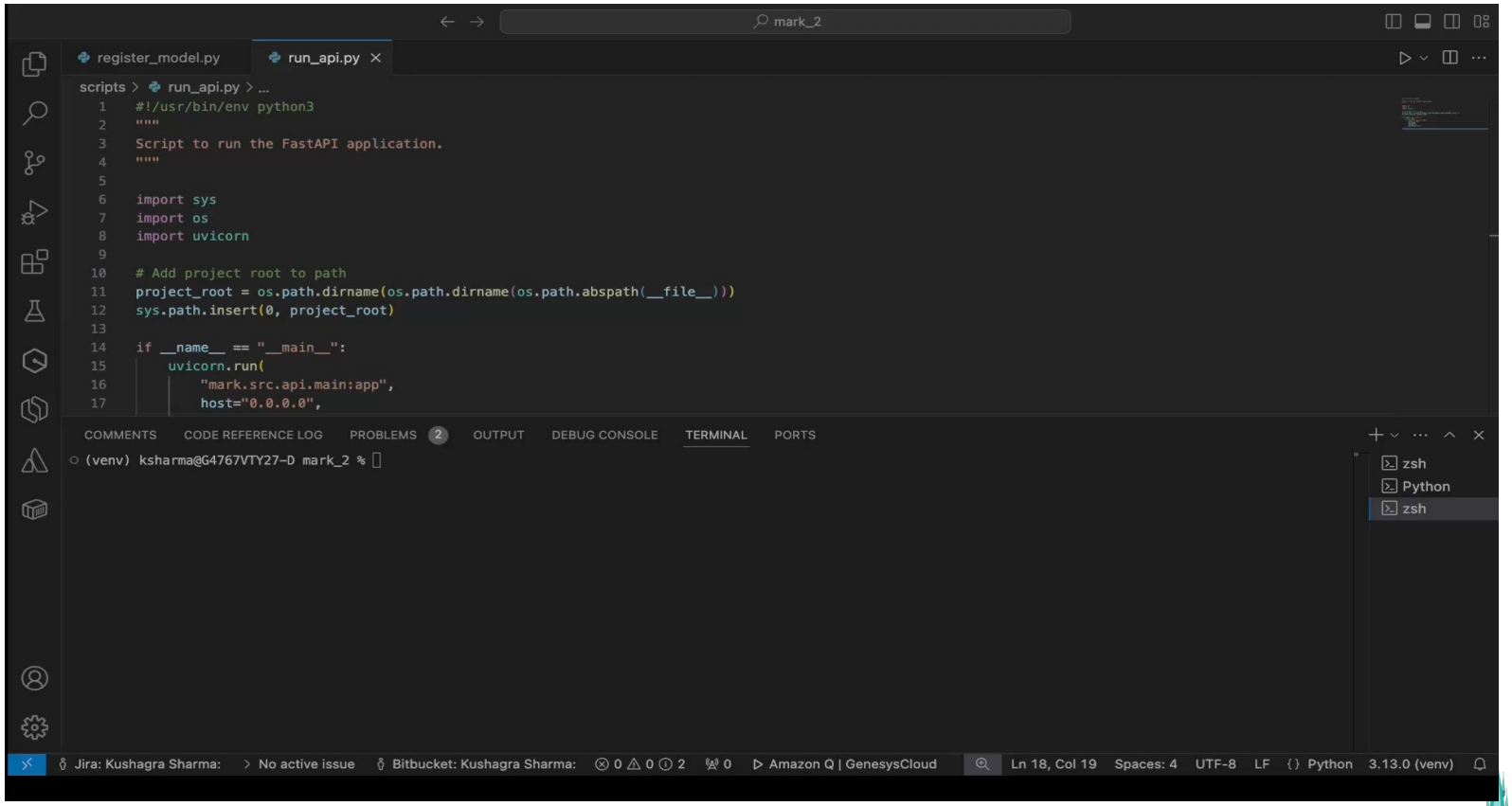
How Document Ingestion Works?

OpenSearchCon EUROPE

User uploads PDF	→ FastAPI receives file → Validates file format
Extract Text Content	→ OpenSearch Attachment Pipeline → Extracts text from PDF → Returns plain text
Split into Chunks	→ Text → Chunks (500 words) → Add overlap (50 words) → Add metadata (source, page)
Generate Embeddings	→ Each chunk → ML API → POST /_plugins/_ml/_predict/text_embedding/{model_id} → Returns 384-dimensional vector
Index in OpenSearch	→ Chunk + Embedding → Store in rag_documents index → knn_vector field stores embedding

Document Ingestion: Demo

OpenSearchCon EUROPE



Semantic Search: Finding Relevant Documents

OpenSearchCon EUROPE

Query Input

User enters a search query

Example: "What are the benefits of OpenSearch?"

Query Embedding

Convert query text to embedding using same ML model

Generate 384-dimensional vector

Vector Similarity Search

Use k-NN search in OpenSearch

Find documents with similar embeddings

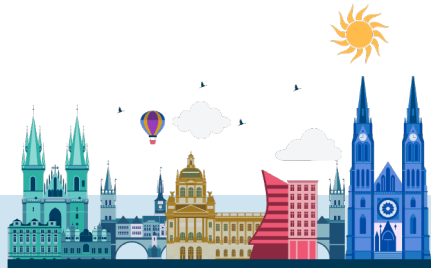
Return top-k most relevant chunks

Results Ranking

Results sorted by similarity score

Higher score = more relevant

Return top N most relevant chunks



Answer Generation: RAG in Action

OpenSearchCon EUROPE

User Query

- Natural language question (text or voice)
- Example: "What are the benefits of OpenSearch?"

Semantic Search

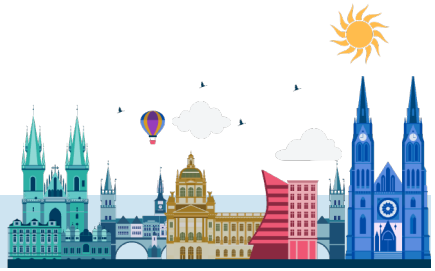
- Find top-k relevant document chunks
- Get: text, metadata, similarity scores
- Example: Top 5 chunks about OpenSearch benefits

Context Assembly

- Combine search results into context
- Include source citations
- Format for LLM prompt

LLM Generation

- Send query + context to LLM
- LLM generates answer based on context
- Include source citations in answer



Voice-Enabled Search: Making RAG Accessible

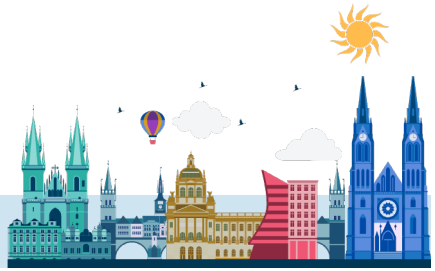
OpenSearchCon EUROPE

Voice Input
User speaks question into microphone

Speech-to-Text (STT)
OpenAI Whisper converts audio to text

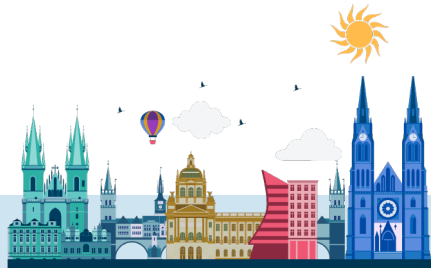
Text Processing
Transcribed text becomes search query & embeddings are generated for it

Answer Generation
LLM synthesizes information & generates answer from search results



Voice-Enabled Semantic Search with RAG: Demo

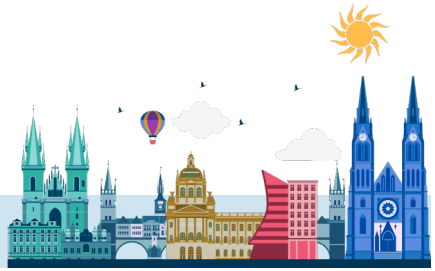
OpenSearchCon EUROPE



Key Takeaways

OpenSearchCon EUROPE

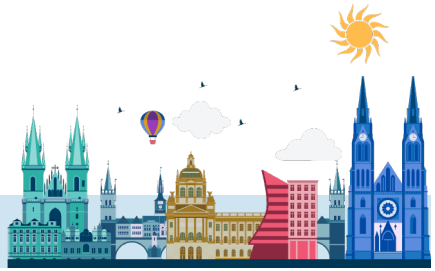
- OpenSearch ML Commons - Deploy embedding models directly on cluster - no external APIs
- Complete RAG Pipeline - PDF ingestion → Semantic search → Answer generation - all in OpenSearch
- Voice-Enabled Search - STT with Whisper, multilingual and accessible
- Self-Contained System - No external dependencies - private, cost-effective, and fully controlled



Benefits : Why this approach matters

OpenSearchCon EUROPE

- Data Privacy
  - No data sent to external services
  - Compliance-friendly (GDPR, HIPAA, etc.)
- Cost Efficiency
  - No per-request API fees
  - Open-source stack
- Performance
  - Low latency - no network calls
  - Fast embedding generation (<50ms)
- Flexibility
  - Customize models for your domain
  - Control over data and processing



Resources

OpenSearchCon EUROPE

OpenSearch ML Commons:

https://docs.opensearch.org/latest/ml-commons-plugin/

Mistral:

https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

Ollama:

https://docs.ollama.com/

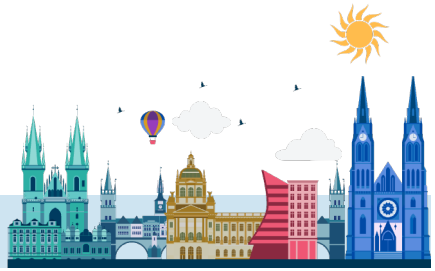
OpenAI Whisper:

https://huggingface.co/openai/whisper-base

Thank you

Contact:
kushshrma66@gmail.com

https://in.linkedin.com/in/kushagra-sharma-07




OpenSearchCon
EUROPE