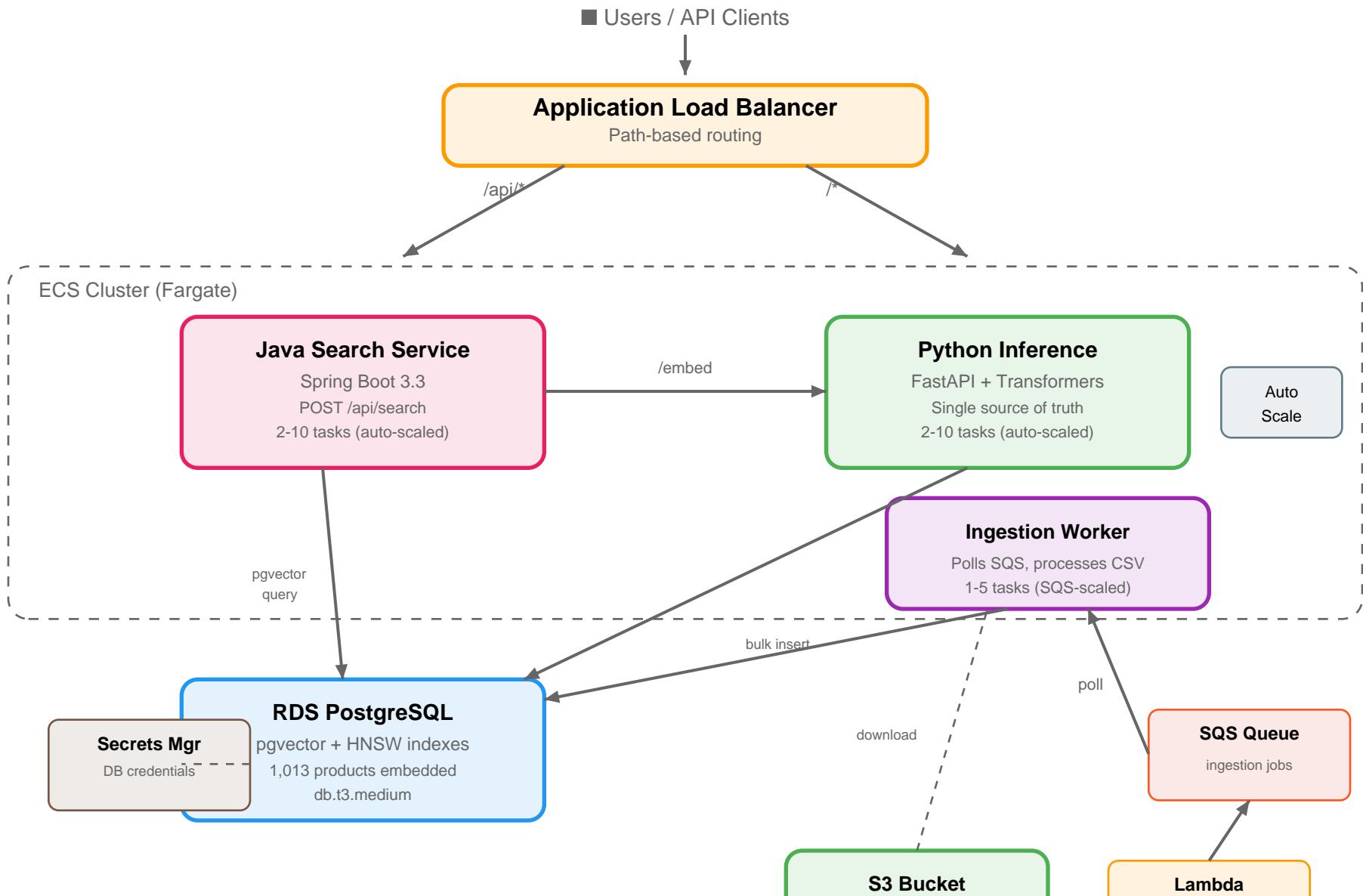


# LLM Vector Search Platform

AWS Architecture - ECS Fargate Deployment



**Data Flow:** CSV → S3 → Lambda → SQS → Worker → Embeddings → PostgreSQL → Search APIs

All services run on ECS Fargate with auto-scaling. Java delegates embeddings to Python (single source of truth for the model).

# Component Details

## Application Load Balancer

- Routes traffic based on URL path: /api/\* → Java, /\* → Python
- Health checks: /api/health (Java), /health (Python)
- Sticky sessions enabled (1 hour), idle timeout 120s
- Target groups with IP-based routing to ECS tasks

## Java Search Service (Spring Boot 3.3)

- Endpoints: POST /api/search, GET /api/health, GET /api/info
- Delegates embedding generation to Python /embed endpoint
- Runs pgvector cosine similarity queries directly against PostgreSQL
- Lightweight: ~512MB memory, ~5 second startup (no local model)

## Python Inference Service (FastAPI)

- Single source of truth for embedding model (all-MiniLM-L6-v2, 384 dims)
- Endpoints: /embed, /search, /search/records, /rag, /generate
- Loads model at startup using HuggingFace Transformers
- Memory: ~2GB (includes model), startup: ~60 seconds

## Ingestion Worker (Python)

- Long-running process that polls SQS with 20-second long polling
- Downloads CSV from S3, auto-detects column mappings
- Generates embeddings in batches (32 texts), bulk inserts (100 rows)
- Moves processed files to completed/ or failed/ prefix

## RDS PostgreSQL + pgvector

- Tables: documents, ingested\_records, ingestion\_jobs
- HNSW indexes on content\_embedding and title\_embedding columns
- Cosine similarity search via <=> operator
- Instance: db.t3.medium (2 vCPU, 4GB RAM)

## **S3 → Lambda → SQS Pipeline**

- S3: Bucket with incoming/, completed/, failed/ prefixes
- Lambda: Triggers on s3:ObjectCreated, sends SQS message
- SQS: Queue with 900s visibility timeout, 4-day retention

# Auto Scaling & Resource Configuration

## Auto Scaling Rules

### Inference Services (Java & Python)

- Min: 2 tasks, Max: 10 tasks
- Scale out when CPU > 60%
- Scale out when Memory > 70%
- Scale out when requests > 100 per target
- Scale-out cooldown: 60-120 seconds
- Scale-in cooldown: 300 seconds

### Ingestion Worker

- Min: 1 task, Max: 5 tasks
- Scale out when SQS queue depth > 5 messages
- Scale in when queue is empty for 5 minutes
- CloudWatch alarms trigger step scaling policies

## Resource Configuration

Service	CPU	Memory	Min Tasks	Max Tasks	Notes
Java Search Service	512 vCPU	1 GB	2	10	Lightweight proxy
Python Inference	4096 vCPU	16 GB	2	10	Owns the model
Ingestion Worker	4096 vCPU	8 GB	1	5	SQS-driven
RDS PostgreSQL	2 vCPU	4 GB	—	—	db.t3.medium

## Security

- Database credentials stored in AWS Secrets Manager (llm-db-credentials)
- IAM roles per service with least-privilege permissions
- VPC security groups restrict traffic between components
- RDS instance not publicly accessible (private subnets only)
- ALB handles TLS termination (HTTPS can be added)