✦ Member-only story

Vespa: The Open-Source Engine Powering Search, Recommendations, and Real-Time Data

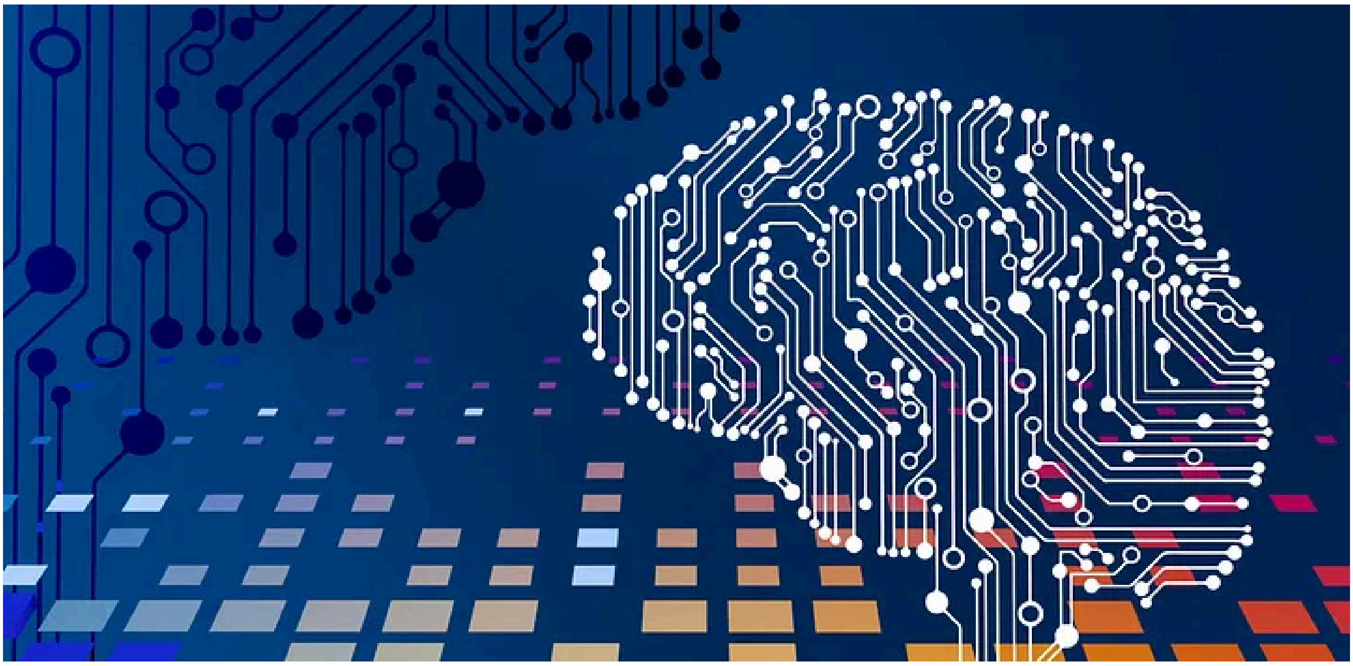
3 min read · 8 hours ago

Civil Learning Follow

Listen Share More

A few weeks ago, I discovered **Vespa** — an open-source search and recommendation engine used by *large companies*. Think of it as a robust system that finds data quickly, ranks it intelligently, and remains reliable even when handling millions of records.

I wanted to find out what all the fuss was about, so I decided to set it up myself. Here's how it went.



· · ·

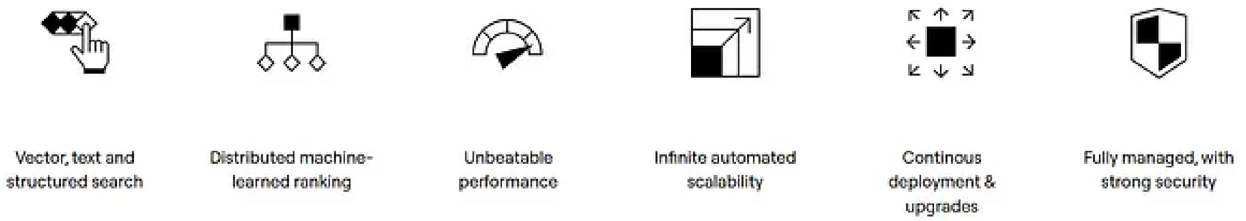
What Vespa Actually Does

If you've ever created a search bar or a recommendation system, you understand the challenge:
You have tons of data, some of it changing every second, and you still need to answer queries instantly.

That is precisely where **Vespa** excels.
It takes your structured data, text, and embeddings (vectors, tensors — all that ML stuff), and helps you:

- search through it,
- run models on it,
- and return meaningful results — in **under 100 ms**.

And while you're doing that, Vespa quietly maintains accessibility by distributing everything across multiple nodes. Pretty neat.


Vector, text and structured search
Distributed machine-learned ranking
Unbeatable performance
Infinite automated scalability
Continous deployment & upgrades
Fully managed, with strong security

· · ·

Getting Vespa Up and Running

Alright, let's get our hands dirty.

1. What You'll Need

You'll need a few basics before handling Vespa:

- **Linux, macOS, or Windows 10 Pro** (x86_64 or ARM64)
- **Docker or Podman** (because Vespa runs in containers)

- **Java 17**
- **Apache Maven**
- **Homebrew** (for installing the Vespa CLI)

If you’re on macOS, the CLI setup is as easy as:

```
brew install vespa-cli
```

You can also obtain it from GitHub if you prefer to install manually.

· · ·

2. Create a Tenant

Go to console.vespa-cloud.com and create a **tenant**.
Think of it as your project workspace on Vespa Cloud — everything you deploy is stored here.

· · ·

3. Configure the Vespa CLI

Once the CLIs are ready, point them to your new tenant:

```
vespa config set target cloud
vespa config set application tenant-name.myapp
```

(Replace `tenant-name` with your own. I used `myapp` because creativity isn’t my strong suit before coffee.)

· · ·

4. Log In

Authenticate with Vespa Cloud:

```
vespa auth login
```

Follow the browser prompt and boom — you’re in.

• • •

5. Clone a Sample App

Now for the fun part. Vespa offers sample apps, so you don’t have to start from scratch. I chose the **album recommendation** app.

```
vespa clone album-recommendation-java myapp
cd myapp
```

If you’re more of a tinkerer, explore the other samples as well — they’re excellent templates.

• • •

6. Add a Public Certificate

This gives your app secure read/write access:

```
vespa auth cert
```

Vespa automatically creates a self-signed certificate and adds it to your package.

• • •

7. Build and Deploy

Build it with Maven:

```
mvn -U package
```

Then deploy it to Vespa Cloud:

```
vespa deploy --wait 600
```

The initial deployment takes some time (nodes are being provisioned). After that, redeployments are much quicker.

8. Feed Some Data

Once the app's up, let's push some sample data:

```
vespa feed src/test/resources/*.json
```

You will see logs confirming that the documents are being indexed.

9. Run a Query

Let's try searching:

```
vespa query "select * from music where album contains 'head'"
```

Or a more personalised one:

```
vespa query \
  "select * from music where true" \
  "ranking=rank_albums" \
  "ranking.features.query(user_profile)={{cat:pop}:0.8,{cat:rock}:0.2,{cat:jazz
```

That last one allows Vespa to adjust rankings based on user preferences. It’s essentially saying, *“show me pop and rock first, maybe sprinkle in a little jazz.”*

Why Vespa Feels Different

Most search systems excel at one thing — either retrieving data quickly or ranking intelligently. Vespa accomplishes both.

It’s designed for massive, real-time workloads and can run machine-learned ranking models directly within the engine. That’s significant when you need personalisation at scale.

And yeah, it’s open source under **the Apache 2.0 license**, which makes it even more appealing.

Wrapping Up

So, that’s how I got Vespa running.
Was it complicated? A little.
Was it worth it? Definitely.

It’s not a simple plug-and-play toy — it’s a **powerful engine** designed for applications that require quick thinking and response.

If your app needs *real-time intelligence*, give Vespa a spin.
Start here: github.com/vespa-engine/vespa

And don’t be surprised if you find yourself spending the rest of your weekend experimenting with it.

AI Ai Agent Ai Agent Development Llm Llm Applications

Follow

Written by Civil Learning

2.8K followers · 6 following

We share what you need to know. Shared only for information.

Responses (1)

Bgerby

What are your thoughts?

See all responses