Lecture 27: Redis Vector Store Config

Why Redis?

- Redis is a popular **in-memory data structure store**.
- Very fast for vector operations.
- Not ideal for permanent databases, but widely used for caching and vector search.
- We will use it as a **Redis Vector Store** with Spring AI.

Required Dependencies:

Add these two dependencies in pom.xml:

```
<dependency>
          <groupId>org.springframework.ai</groupId>
          <artifactId>spring-ai-starter-vector-store-redis</artifactId>
</dependency>

 <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-redis</artifactId>
 </dependency>
```

Docker Setup:

- Run the Redis stack using Docker:
    - Image will run on **port 6379**.
    - Docker pulls the latest Redis image and starts the container.
    - Run the command given in the terminal:

docker run -d --name redis-stack -p 6379:6379 -p 8001:8001 redis/redis-stack:latest

Application Properties:

```
spring.data.redis.host=localhost
spring.data.redis.port=6379

spring.ai.vectorstore.redis.index-name=product-index
spring.ai.vectorstore.redis.prefix=product
```

- host → Localhost
- port → 6379 (Docker default)
- index-name → Index name used for storing embeddings
- prefix → Key prefix for Redis storage

Key Notes:

- Redis needs **Jedis client** (Spring Data Redis handles pooling).

- Initialize schema by setting property (default = false).
- Configuration is simpler compared to PGVector.
- Vector search logic remains the same; only the **storage backend changes**.