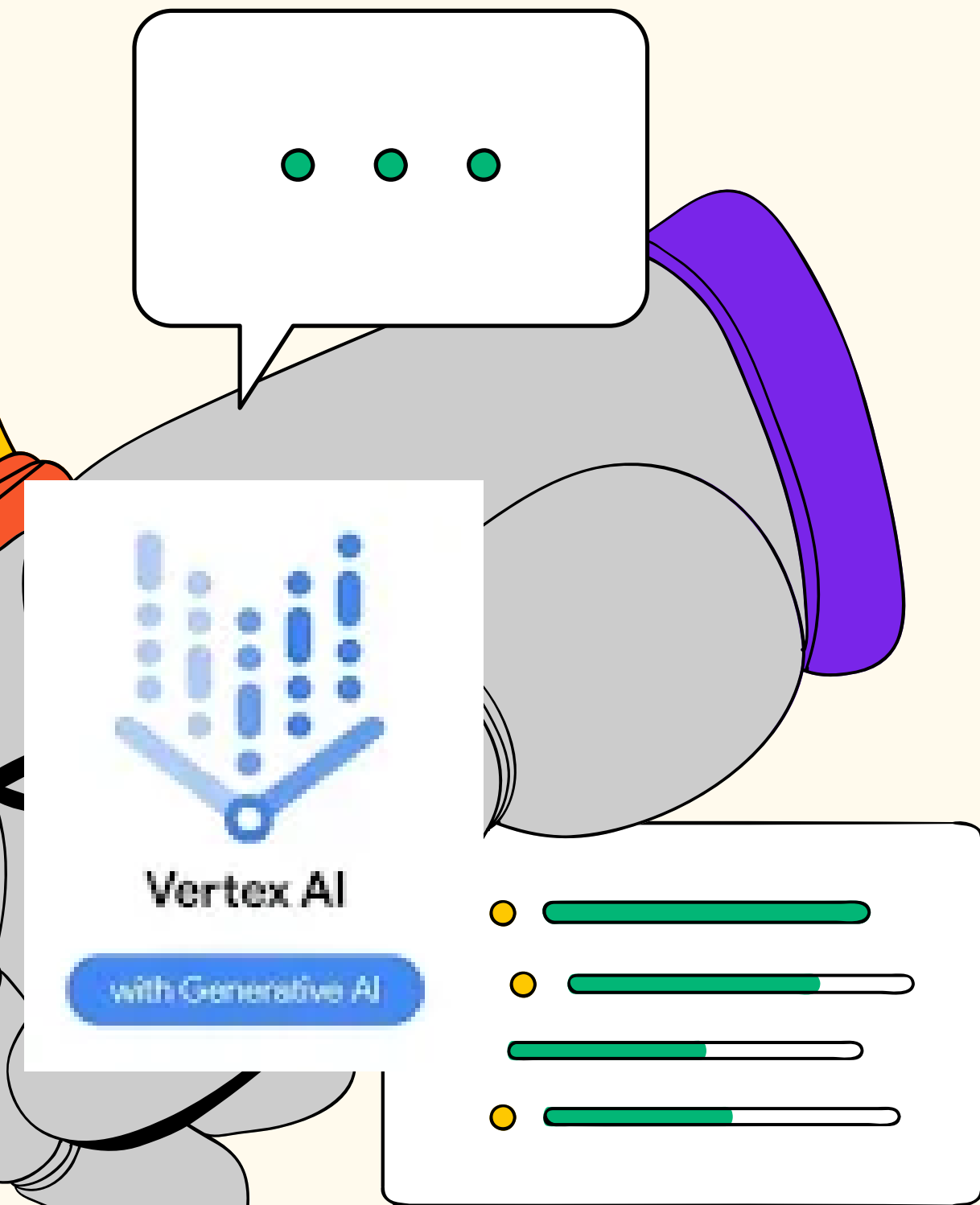
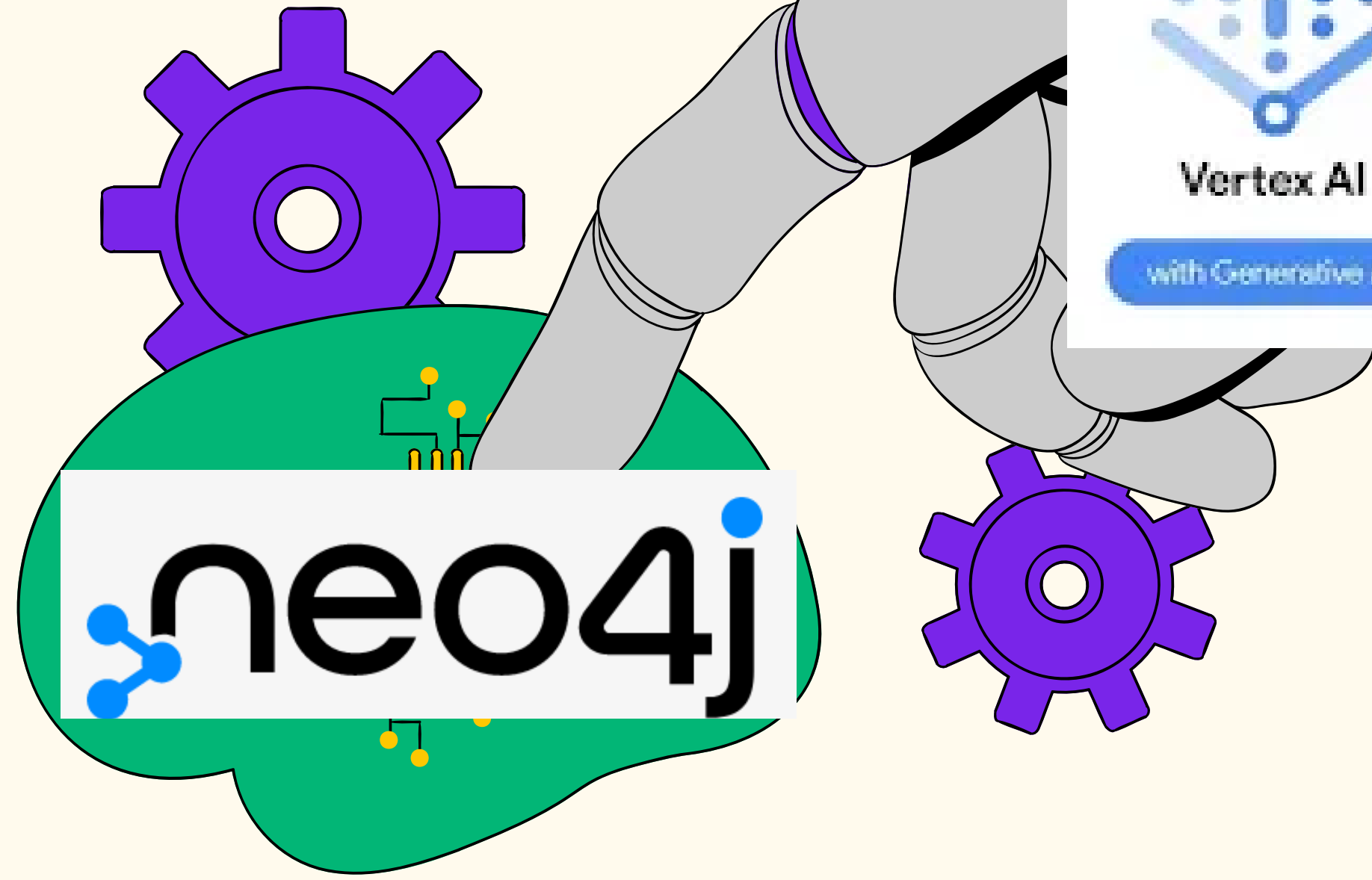


# IMPLEMENTING DOCUMENT QA WITH NEO4J & VERTEX AI STEP BY STEP INTRODUCTION OF GETTING LLM TO WORK WITH NEO4J



# CHALLENGE SOLVED: EMBEDDING RELATIONSHIPS

- NEO4J STORES THE NODES AND RELATIONSHIPS BETWEEN THE NODES AS GRAPHS. OUR OBJECTIVE IS TO EMBED THESE RELATIONSHIPS INTO VECTORS AND USE IT FOR DOCUMENT RETRIEVAL
- RECENTLY NEO4J HAS RELEASED NEW PROCEDURES FOR CREATING THESE EMBEDDINGS USING BOTH VERTEXAI & OPENAI EMBEDDING MODELS.
- NEO4J GRAPH DATASCIENCE LIBRARY HAS COSINE DISTANCE ALGORITHMS WHICH CAN BE USED FOR FINDING THE QUERY VS DOCUMENT VECTOR DISTANCE
- NEO4J BROWSER INTERFACE ITSELF CAN BE USED FOR DOING THE DOCUMENT RETRIEVAL. ALSO IT CAN BE LINKED WITH JUPYTER NOTEBOOK USING NEO4J LIBRARY4
- TEXT CREATED USING THE NODES AND THE RELATION IS ONLY EMBEDDED. NOT THE NODES OR RELATIONS

# STEPS INVOLVED IN EMBEDDING RELATIONSHIPS

1) ENABLING VERTEXAI API ACCESS IN GCP & USING APPLICATION DEFAULT CREDENTIALS ROUTE TO GET THE ACCESS TOKENS

LINK: [HTTPS://CLOUD.GOOGLE.COM/DOCS/AUTHENTICATION/USE-CASES](https://cloud.google.com/docs/authentication/use-cases)

2) INSTALLING THE VERTEXAI PROCEDURES IN THE NEO4J DOCKER INSTANCE BY DOWNLOADING THE LATEST APOC LIBRARY

LINK: [HTTPS://GITHUB.COM/NEO4J-CONTRIB/NEO4J-APOC-](https://github.com/neo4j-contrib/neo4j-apoc-procedures/releases/download/5.8.1/apoc-5.8.1-extended.jar)

[PROCEDURES/RELEASES/DOWNLOAD/5.8.1/APOC-5.8.1-EXTENDED.JAR](https://github.com/neo4j-contrib/neo4j-apoc-procedures/releases/download/5.8.1/apoc-5.8.1-extended.jar)

3) CREATING THE NEO4J DOCKER INSTANCE IN CLOUD VIRTUAL MACHINE, AND LOADING THE APOC JAR FILES INTO THE PLUGIN DIRECTORY

COMMAND :`DOCKER DOCKER RUN --NAME TESTNEO4J -P 7474:7474 -P 7687:7687 -D -V`

`$HOME/NEO4J/DATA:/DATA -V $HOME/NEO4J/LOGS:/LOGS -V`

`$HOME/NEO4J/IMPORT:/VAR/LIB/NEO4J/IMPORT -V`

`$HOME/NEO4J/PLUGINS:/VAR/LIB/NEO4J/PLUGINS --ENV NEO4J_AUTH=NEO4J/PASSWORD --ENV`

`NEO4J_PLUGINS='["APOC","GRAPH-DATA-SCIENCE"]' NEO4J:LATEST`

**[HTTPS://GITHUB.COM/INSIGHTBUILDER](https://github.com/InsightBuilder)**

# CYPHER NEEDS TO BE BROKEN DOWN

- APOC (AWESOME PROCEDURES ON CYPHER) GENERAL FUNCTIONS FOR DATA MANIPULATION, MATH CALCULATIONS
- GDS (GRAPH DATA SCIENCE) GRAPH ALGORITHMS AND DATA SCIENCE IMPLEMENTATIONS
- BOTH ARE LIBRARIES THAT HAS MANY FUNCTIONS THAT CAN SUPER CHARGE THE DATA AND SCIENCE FLOW, ADDING THE POWER OF GRAPHS ON TOP OF IT

## SOME FUNCTIONS USED IN CODE

- COLLECT : TAKES A SERIES OF DATA, AND RETURNS A LIST
- REDUCE : TAKES A LIST AND REDUCES IT BY APPLYING ONE OPERATION REPEATEDLY ON CONSECUTIVE ELEMENTS
- COALESCE: SIMILAR TO SQL COALESCE. RETURNS NON NULL VALUES IN GIVEN LIST OF ELEMENTS
- SUBSTRING: RETURN PART OF THE STRING, BASED ON THE LENGTH PROVIDED
- APOC.TEXT.JOIN: JOINS THE LIST ELEMENTS INTO A STRING
- APOC.PERIODIC.ITERATE: ITERATES OVER THE NODES
- APOC.ML.VERTEXAI.EMBEDDING: CREATES EMBEDDINGS OUT OF THE TEXT PROVIDED TO IT
- APOC.ML.VERTEXAI.COMPLETION: CALLS THE VERTEXAI COMPLETION END POINT
- GDS.SIMILARITY.COSINE(QUERY\_EMBED, STORED\_EMBEDDING) PROVIDES THE SCORE

# WHAT IS NEXT?

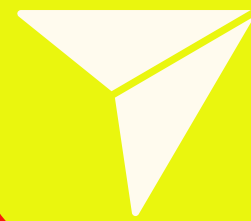
- CREATE NEO4J DOCKER INSTANCE IN CLOUD VIRTUAL MACHINE. ENSURE THE LATEST APOC / GDS LIBRARIES ARE INCLUDED IN THE CONFIGURATION
- [HTTPS://NEO4J.COM/DOCS/CYPHER-CHEAT-SHEET/5/AURADB-ENTERPRISE/](https://neo4j.com/docs/cypher-cheat-sheet/5/auradb-enterprise/)
- LOAD THE RECENT APOC LIBRARY INTO THE PLUGINS FOLDER AND RESTART DOCKER
- LOAD EXAMPLE DATASET FROM THE INTERNET / NEO4J REPOSITORY INTO NEO4J INSTANCE
- SEE HOW THE HOW EMBEDDING IS CREATED PRACTICALLY
- USE THE APOC.PERIODIC.ITERATE PROCEDURE TO CREATE & STORE EMBEDDING FOR THE NODES AND RELATIONSHIPS
- DO THE QA:
  - THE USER QUERY,
  - EMBED IT WITH VERTEXAI,
  - LOCATE THE N NEAREST DOCS USING COSINE ROUTINE USING GDS LIBRARY
  - RETURN THE NEAREST DOCUMENTS

# THANKS FOR WATCHING

REMEMBER TO PRACTICE WITH EXAMPLES



**LIKE**



**SHARE**



**SUBSCRIBE**