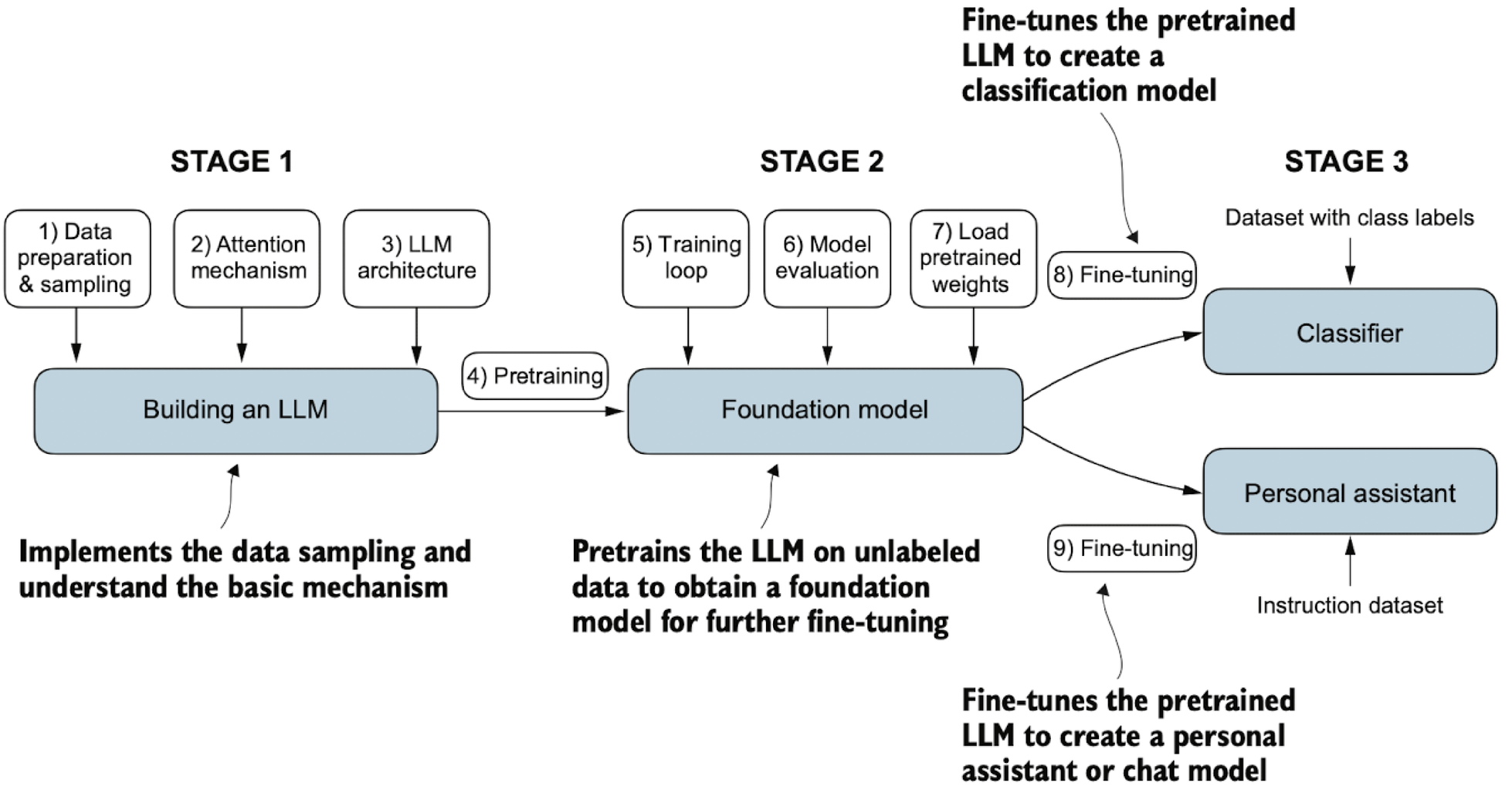
Running Large Language Models (LLMs) on IBM i

Adam Shedivy, IBM
Software Developer, AI solutions for IBM i
adam.shedivy@ibm.com

Goals for today

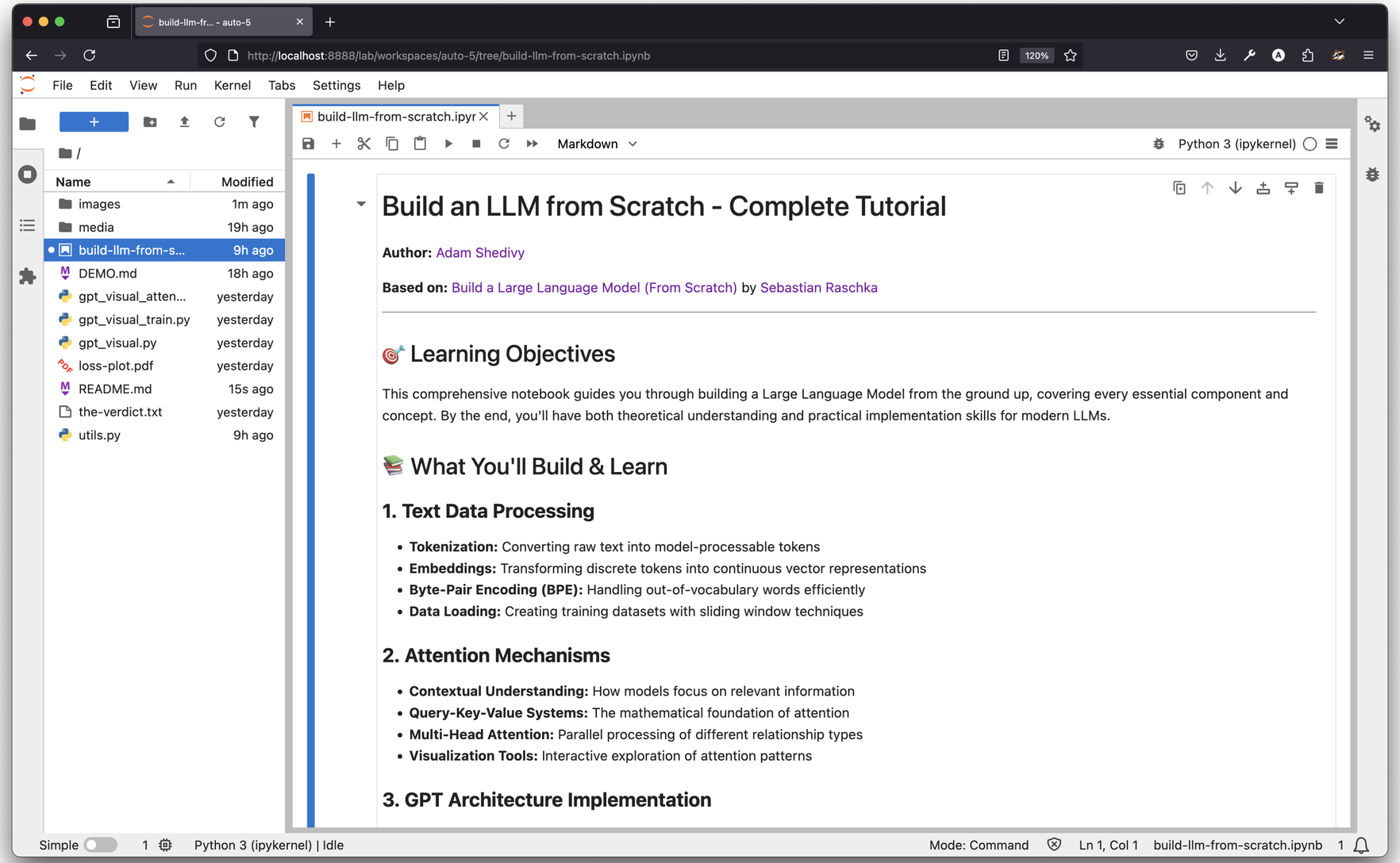
- Understand how LLMs work
- Run an LLM on IBM i
- Integrate with Db2 for i workloads

Overview



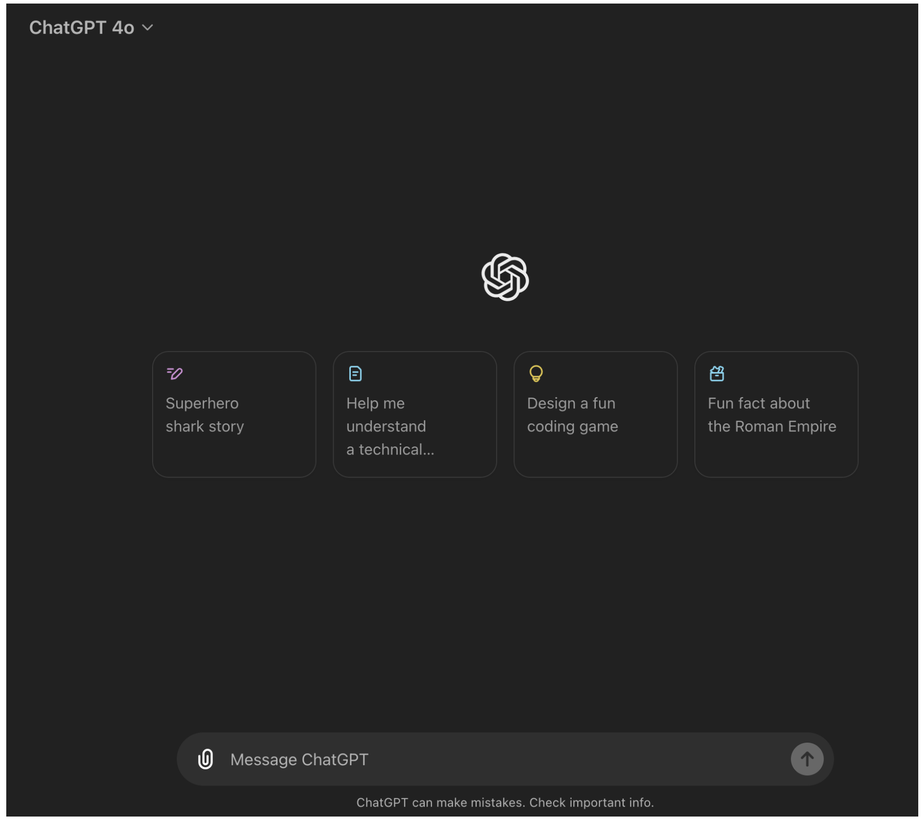
Code

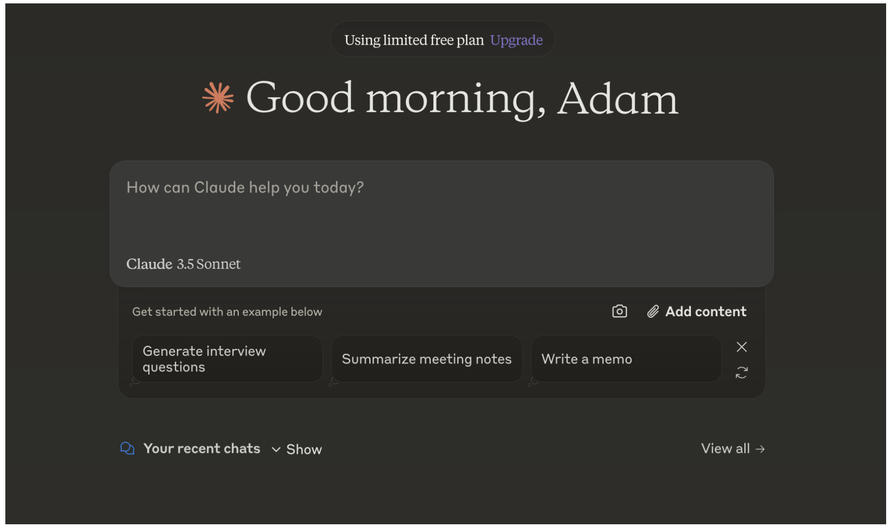
– Full Demo: https://github.com/ajshedivy/LLMs-from-scratch/tree/adam/demo

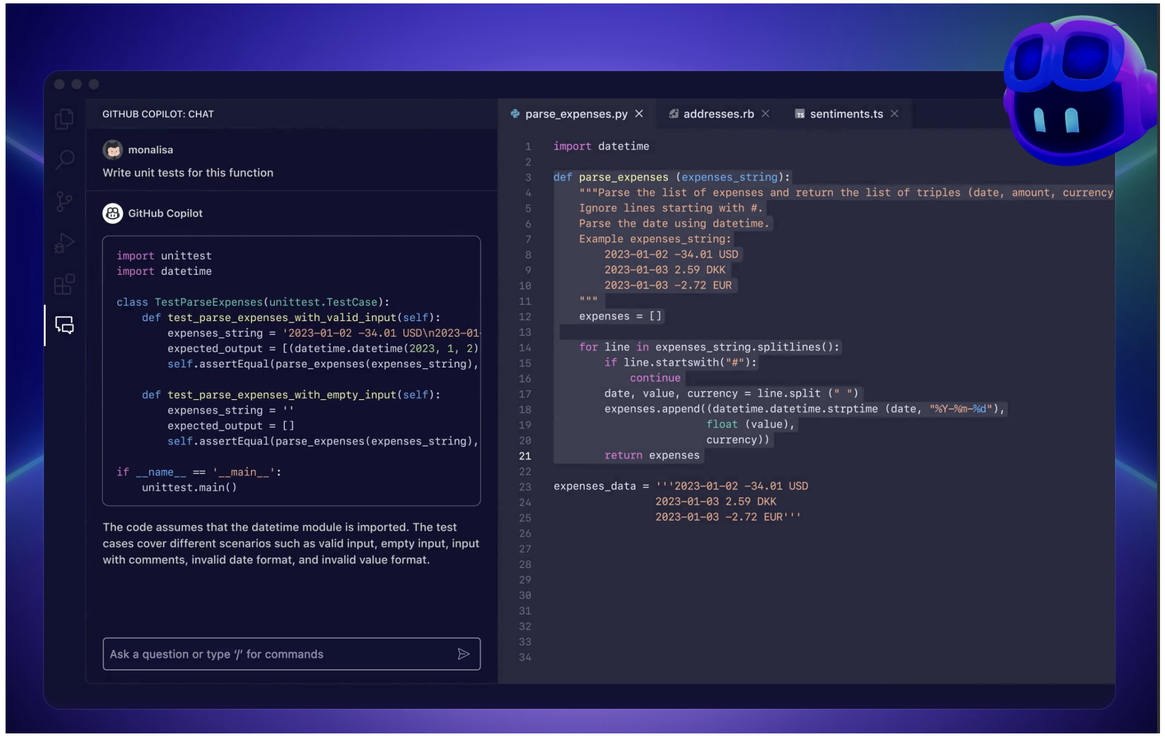


Local (PC) AI Inference

What tools exist?







How to use LLMs locally?

Lots of ways to run a LLM locally
– Llama.cpp, Ollama, transformers (Hugging Face), LMStudio, LangChain

Running LLMs on Power

- Run LLMs locally using Ollama


@8b44af8821f0:/
Shell
[root@8b44af8821f0 /]# ollama run granite3-dense:2b --verbose
>>> Write me a 300 word blurb about IBM Power10
nmon-16p [H for help] Hostname=mma-bench Refresh= 2secs 21:26.12
CPU Utilisation
CPU User% Sys% Wait% Idle
1 96.6 0.0 0.0 3.4
2 94.1 1.0 0.0 5.0
3 96.6 0.0 0.0 3.4
4 95.1 1.5 0.0 3.4
5 97.5 0.5 0.0 2.0
6 95.6 1.0 0.0 3.4
7 96.6 0.0 0.0 3.4
8 96.1 1.0 0.0 3.0
9 97.0 0.0 0.0 3.0
10 90.7 0.0 0.0 9.3
11 97.0 0.0 0.0 3.0
12 95.0 0.5 0.0 4.5
13 97.0 0.0 0.0 3.0
14 96.6 0.0 0.0 3.4
15 93.5 2.0 0.0 4.5
16 96.0 1.0 0.0 3.0
Avg 95.6 0.5 0.0 3.8

Performance Comparison 🔥 😎

Prompt: Why is the sky blue?

P10

```
total duration:       2.079801045s
load duration:        8.568136ms
prompt eval count:    13 token(s)
prompt eval duration: 137.423ms
prompt eval rate:     94.60 tokens/s
eval count:           78 token(s)
eval duration:        1.883978s
eval rate:            41.40 tokens/s
```

P9

```
total duration:       27.506958826s
load duration:        18.1912ms
prompt eval count:    13 token(s)
prompt eval duration: 3.922594s
prompt eval rate:     3.31 tokens/s
eval count:           79 token(s)
eval duration:        23.52334s
eval rate:            3.36 tokens/s
```

Running LLMs on IBM i

Running LLMs on IBM i

LLaMA C++

- Llama.cpp is supported on IBM i
- Llama.cpp is a light-weight pure C/C++ library for LLM inference
- Offers built in CLI (shown in GIF), Server, chat UI

Blog Post: https://ibm.biz/llamacpp-ibmi

Advantages

- Data Privacy & Security
- Performance and Cost
- Flexibility

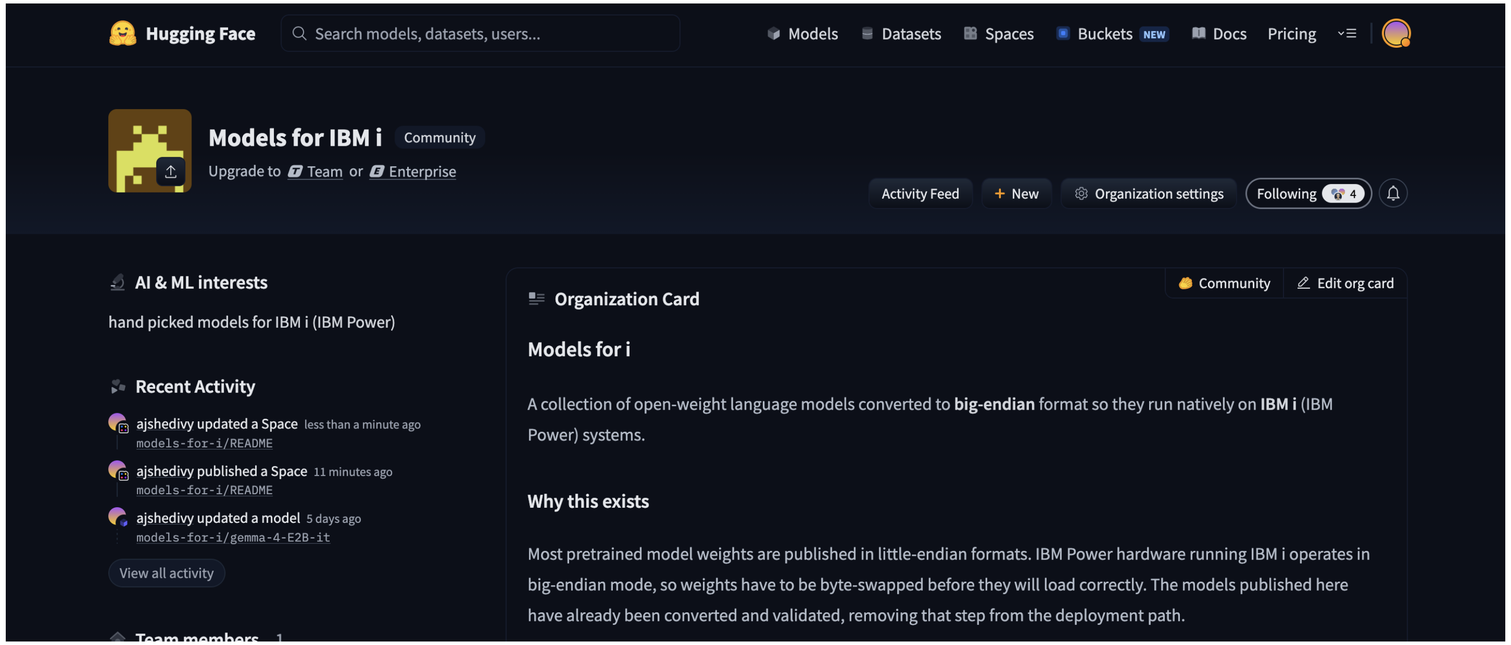
Drawbacks

- Complex setup and maintenance
- Limited Scalability
- Model Updates and latest features

Bigger isn’t always better.

How this works

1. Build/Install Llama.cpp (CPU) on IBM i
2. Transfer model to IBM i
3. Run Llama.cpp



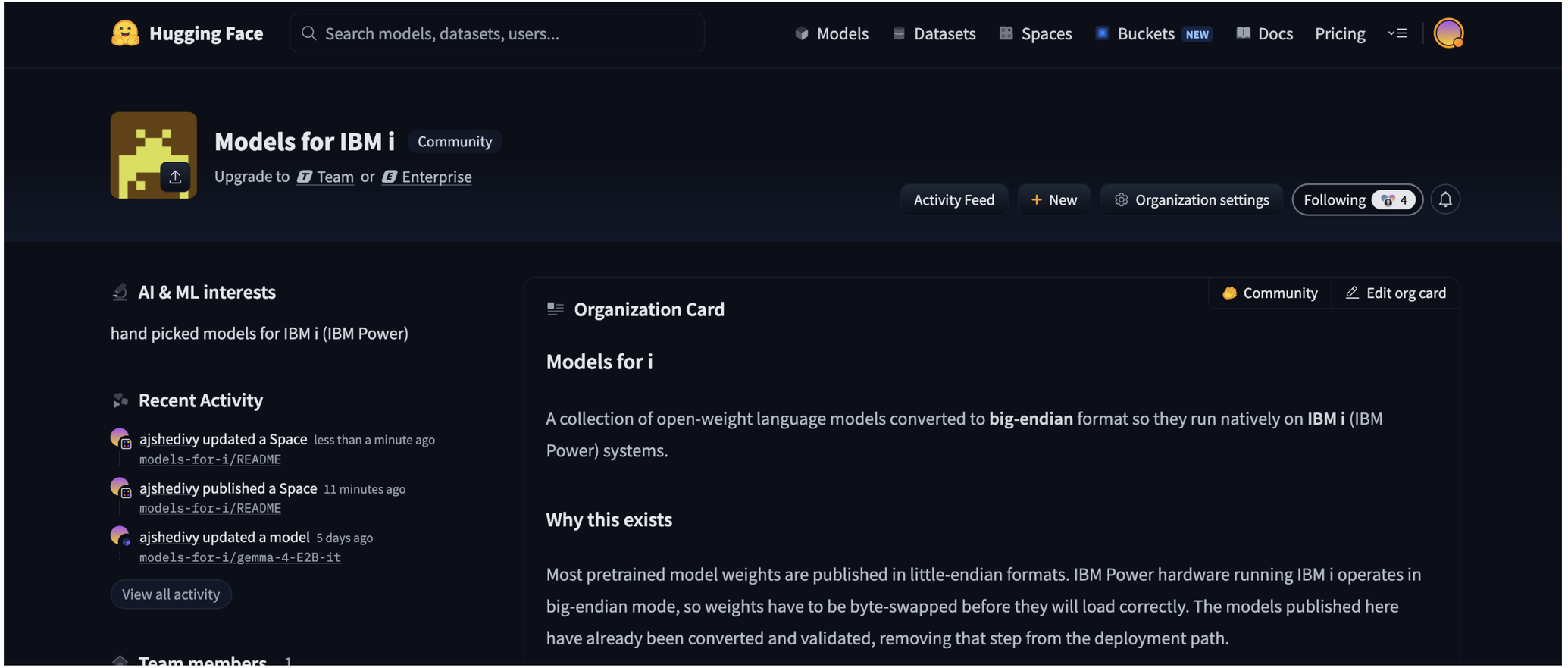
Build/Install Llama.cpp

```
yum install llama-cpp
```

Or, if you’re a weirdo….

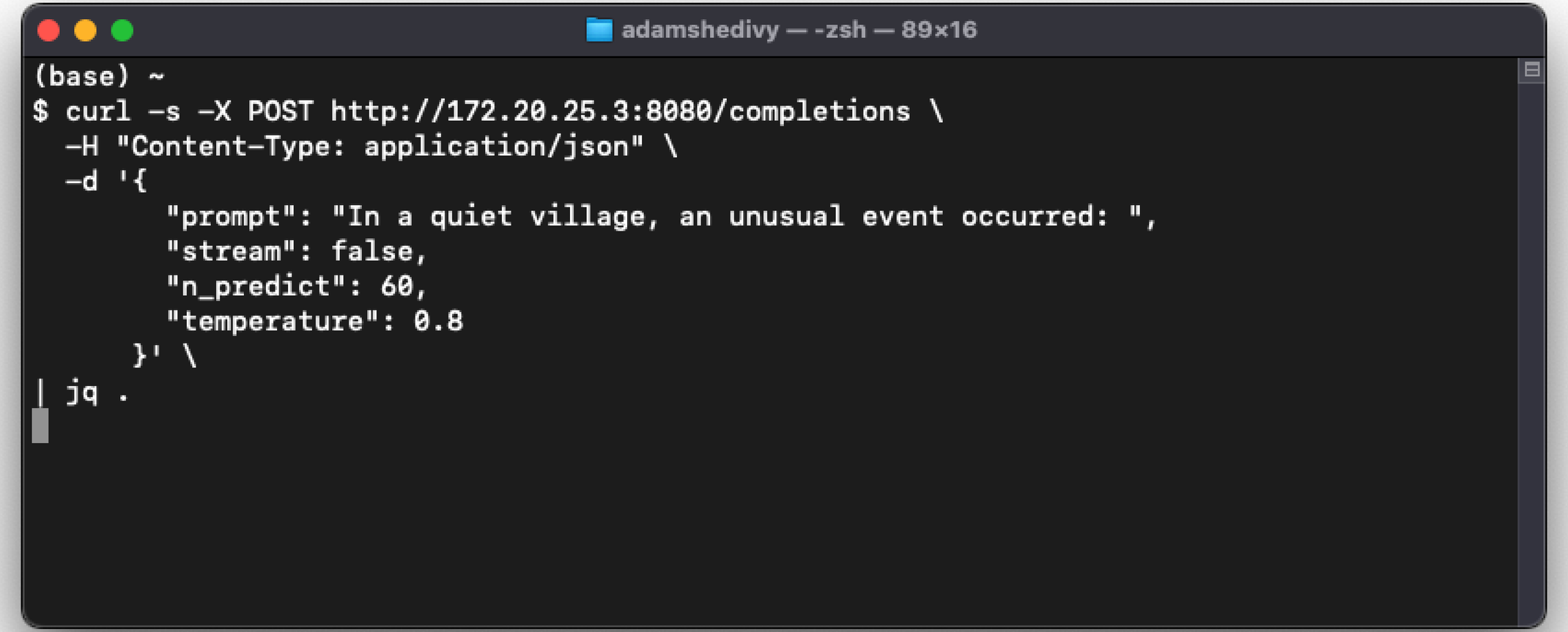
```
cmake -B build \
  -DCMAKE_AR="/QOpenSys/pkgs/bin/gcc-ar-12" \
  -DCMAKE_RANLIB="/QOpenSys/usr/bin/ranlib" \
  -DCMAKE_C_COMPILER=/QOpenSys/pkgs/bin/gcc-12 \
  -DCMAKE_CXX_COMPILER=/QOpenSys/pkgs/bin/g++-12 \
  -DCMAKE_CXX_FLAGS="-pthread" \
  -DCMAKE_EXE_LINKER_FLAGS="-pthread -lutil" \
  -DCMAKE_CXX_ARCHIVE_CREATE="<CMAKE_AR> -X64 qc <TARGET> <LINK_FLAGS> <OBJECTS>" \
  -DCMAKE_CXX_ARCHIVE_APPEND="<CMAKE_AR> -X64 q <TARGET> <LINK_FLAGS> <OBJECTS>" \
  -DCMAKE_CXX_ARCHIVE_FINISH="<CMAKE_RANLIB> <TARGET>" \
  -DCMAKE_BUILD_TYPE=Release \
   -DGGML_LLAMAFILE=OFF
```

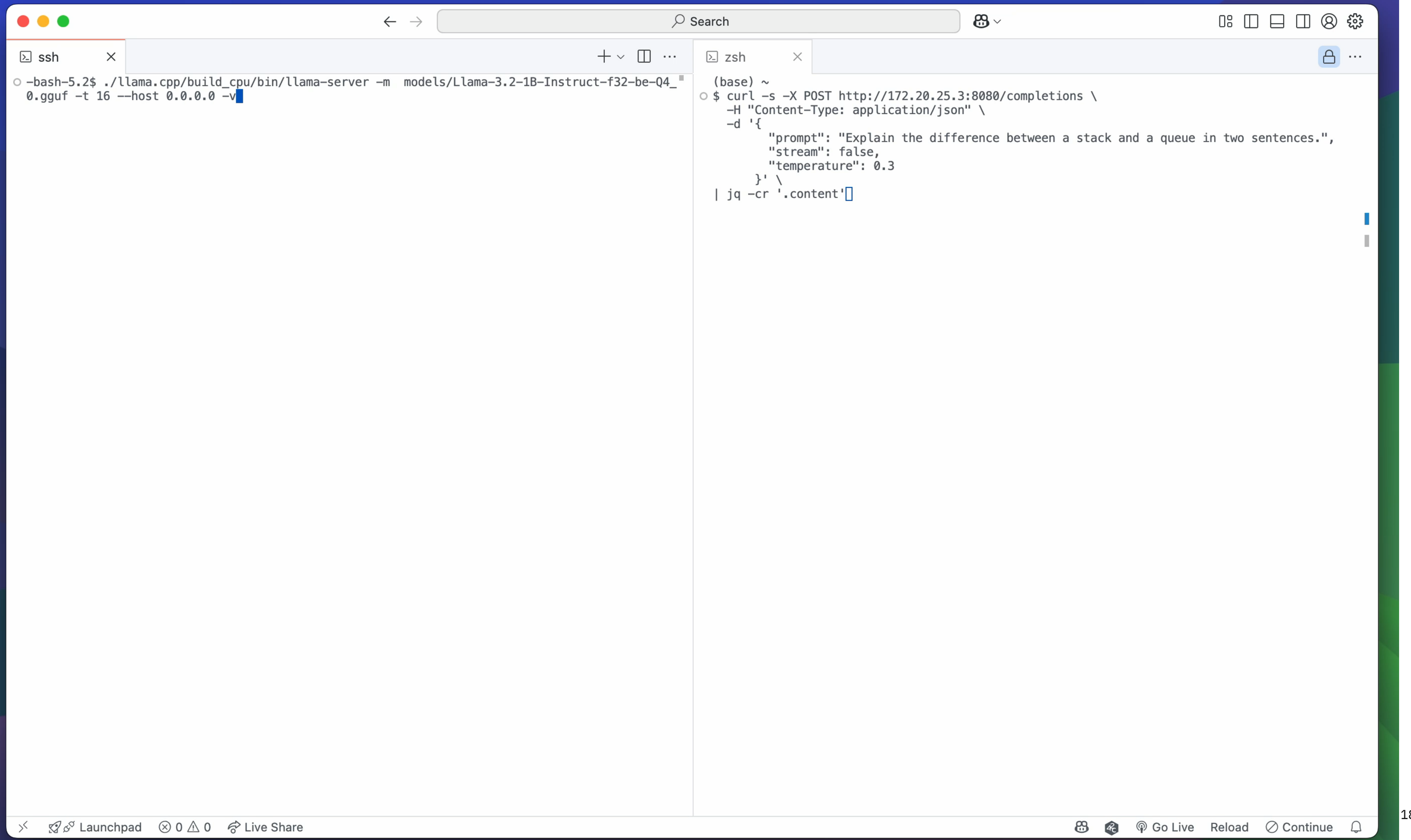
Download a model

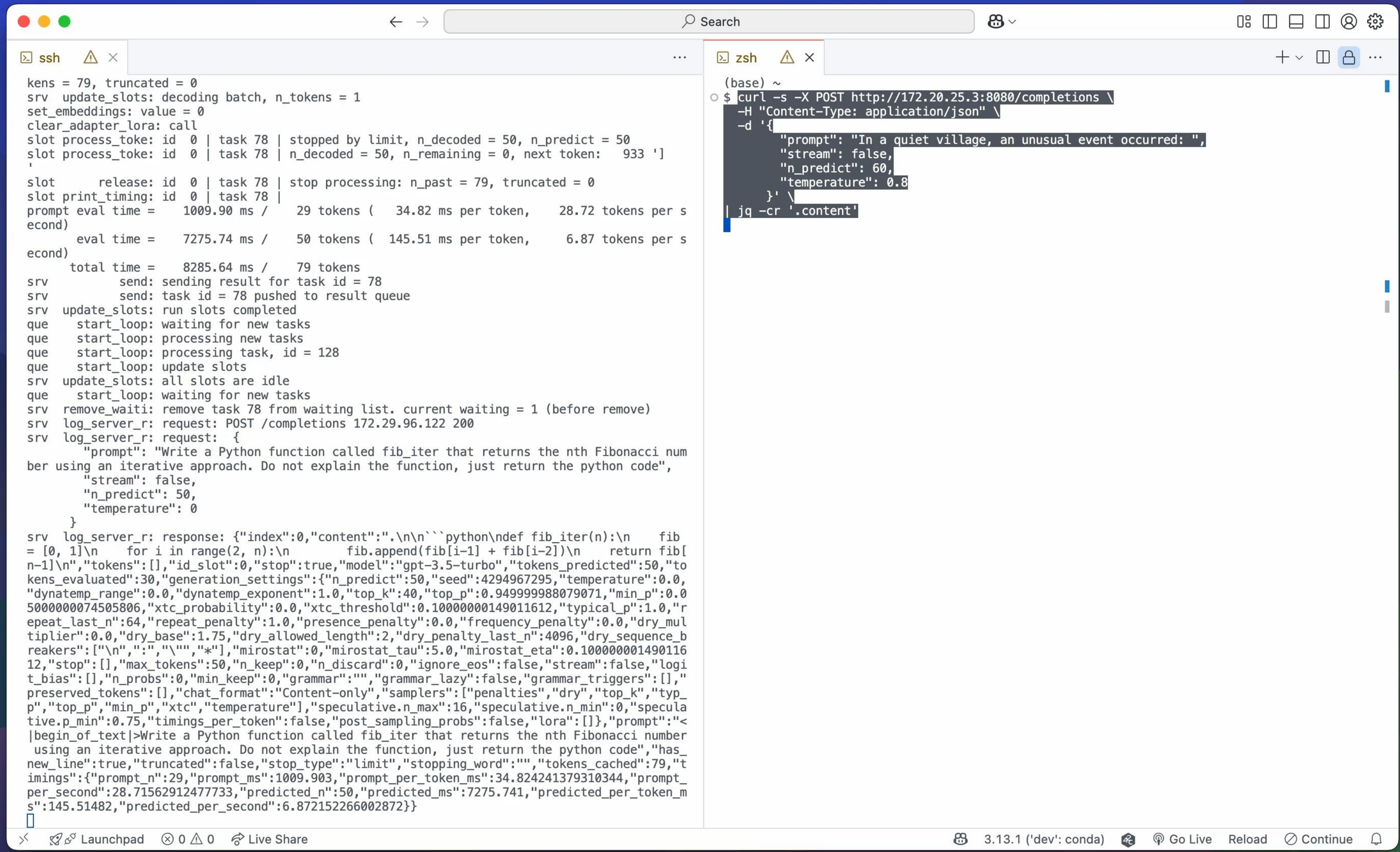


https://huggingface.co/models-for-i

Llama-server




Search
ssh
-bash-5.2$ ./llama.cpp/build_cpu/bin/llama-server -m models/Llama-3.2-1B-Instruct-f32-be-Q4_
0.gguf -t 16 --host 0.0.0.0 -v
zsh
(base) ~
$ curl -s -X POST http://172.20.25.3:8080/completions \
-H "Content-Type: application/json" \
-d '{
"prompt": "Explain the difference between a stack and a queue in two sentences.",
"stream": false,
"temperature": 0.3
}' \
| jq -cr '.content'
Launchpad
0 0
Live Share
Go Live
Reload
Continue


Search
ssh
kens = 79, truncated = 0
srv update_slots: decoding batch, n_tokens = 1
set_embeddings: value = 0
clear_adapter_lora: call
slot process_toke: id 0 | task 78 | stopped by limit, n_decoded = 50, n_predict = 50
slot process_toke: id 0 | task 78 | n_decoded = 50, n_remaining = 0, next token: 933 ']
'
slot   release: id 0 | task 78 | stop processing: n_past = 79, truncated = 0
slot print_timing: id 0 | task 78 |
prompt eval time = 1009.90 ms / 29 tokens (34.82 ms per token, 28.72 tokens per s
econd)
eval time = 7275.74 ms / 50 tokens (145.51 ms per token, 6.87 tokens per s
econd)
total time = 8285.64 ms / 79 tokens
srv send: sending result for task id = 78
srv send: task id = 78 pushed to result queue
srv update_slots: run slots completed
que start_loop: waiting for new tasks
que start_loop: processing new tasks
que start_loop: processing task, id = 128
que start_loop: update slots
srv update_slots: all slots are idle
que start_loop: waiting for new tasks
srv remove_waiti: remove task 78 from waiting list. current waiting = 1 (before remove)
srv log_server_r: request: POST /completions 172.29.96.122 200
srv log_server_r: request: {
"prompt": "Write a Python function called fib_iter that returns the nth Fibonacci num
ber using an iterative approach. Do not explain the function, just return the python code",
"stream": false,
"n_predict": 50,
"temperature": 0
}
srv log_server_r: response: {"index":0,"content":".\n\n```python\ndef fib_iter(n):\n fib
= [0, 1]\n for i in range(2, n):\n fib.append(fib[i-1] + fib[i-2])\n return fib[
n-1]\n","tokens":[],"id_slot":0,"stop":true,"model":"gpt-3.5-turbo","tokens_predicted":50,"to
kens_evaluated":30,"generation_settings":{"n_predict":50,"seed":4294967295,"temperature":0.0,
"dynatemp_range":0.0,"dynatemp_exponent":1.0,"top_k":40,"top_p":0.949999988079071,"min_p":0.0
5000000074505806,"xtc_probability":0.0,"xtc_threshold":0.10000000149011612,"typical_p":1.0,"r
epeat_last_n":64,"repeat_penalty":1.0,"presence_penalty":0.0,"frequency_penalty":0.0,"dry_mul
tiplier":0.0,"dry_base":1.75,"dry_allowed_length":2,"dry_penalty_last_n":4096,"dry_sequence_b
reakers":["\n",":","\"","*"],"mirostat":0,"mirostat_tau":5.0,"mirostat_eta":0.100000001490116
12,"stop":[],"max_tokens":50,"n_keep":0,"n_discard":0,"ignore_eos":false,"stream":false,"logi
t_bias":[],"n_probs":0,"min_keep":0,"grammar":"","grammar_lazy":false,"grammar_triggers":[],"
preserved_tokens":[],"chat_format":"Content-only","samplers":["penalties","dry","top_k","typ_
p","top_p","min_p","xtc","temperature"],"speculative.n_max":16,"speculative.n_min":0,"specula
tive.p_min":0.75,"timings_per_token":false,"post_sampling_probs":false,"lora":[]},"prompt":"<
|begin_of_text|>Write a Python function called fib_iter that returns the nth Fibonacci number
using an iterative approach. Do not explain the function, just return the python code","has_
new_line":true,"truncated":false,"stop_type":"limit","stopping_word":"","tokens_cached":79,"t
imings":{"prompt_n":29,"prompt_ms":1009.903,"prompt_per_token_ms":34.824241379310344,"prompt_
per_second":28.71562912477733,"predicted_n":50,"predicted_ms":7275.741,"predicted_per_token_m
s":145.51482,"predicted_per_second":6.87215266002872}}
zsh
(base) ~
$ curl -s -X POST http://172.20.25.3:8080/completions \
-H "Content-Type: application/json" \
-d '{
"prompt": "In a quiet village, an unusual event occurred: ",
"stream": false,
"n_predict": 60,
"temperature": 0.8
}' \
| jq -cr '.content'
Launchpad
0 0
Live Share
3.13.1 ('dev': conda)
Go Live
Reload
Continue

Conversations

+ New conversation

llama.cpp

Send a message to start

Overview

- You can run LLMs on IBM i using **Llama.cpp**
- Leverage Power hardware for running LLMs on CPU
- Build From source: https://ibm.biz/llamacpp-ibmi

Embedded AI Transactions

Embedding AI into a Db2 Transaction

Bringing AI into the transaction

Goal

- Embed AI inside a database transaction

Architecture

- Integrate LLM Provider REST APIs

Application

- Data summarization
- Data classification
- Data analysis

Today

- Watsonx, Ollama, OpenAI-Compatible support
- Manual Installation, basic usage

Soon...

- Other LLM providers
- Improved usage and support
- Better docs
- Integrations with other AI stacks

AI SDK for Db2 for i

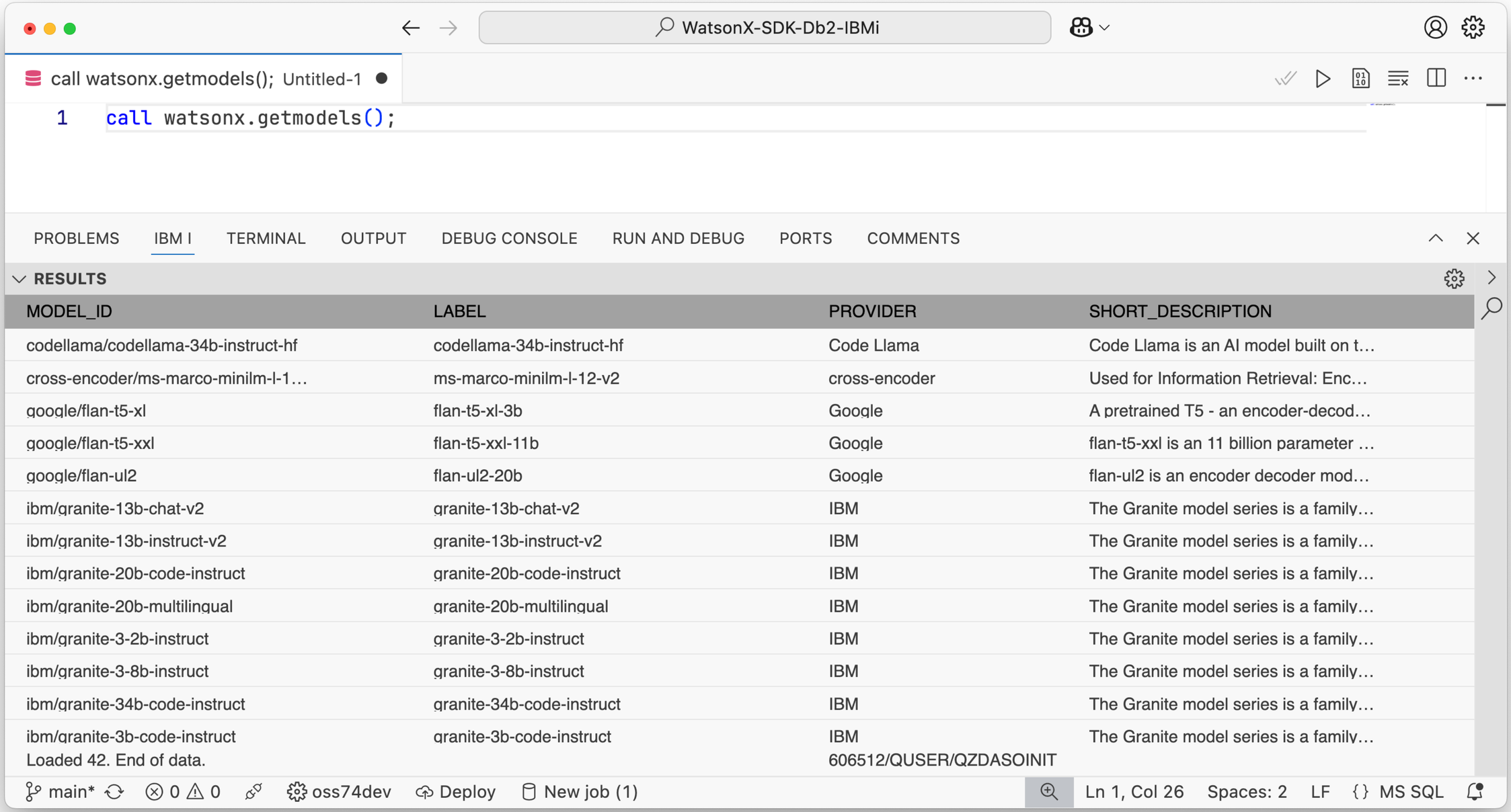
https://github.com/IBM/AI-SDK-Db2-IBMi

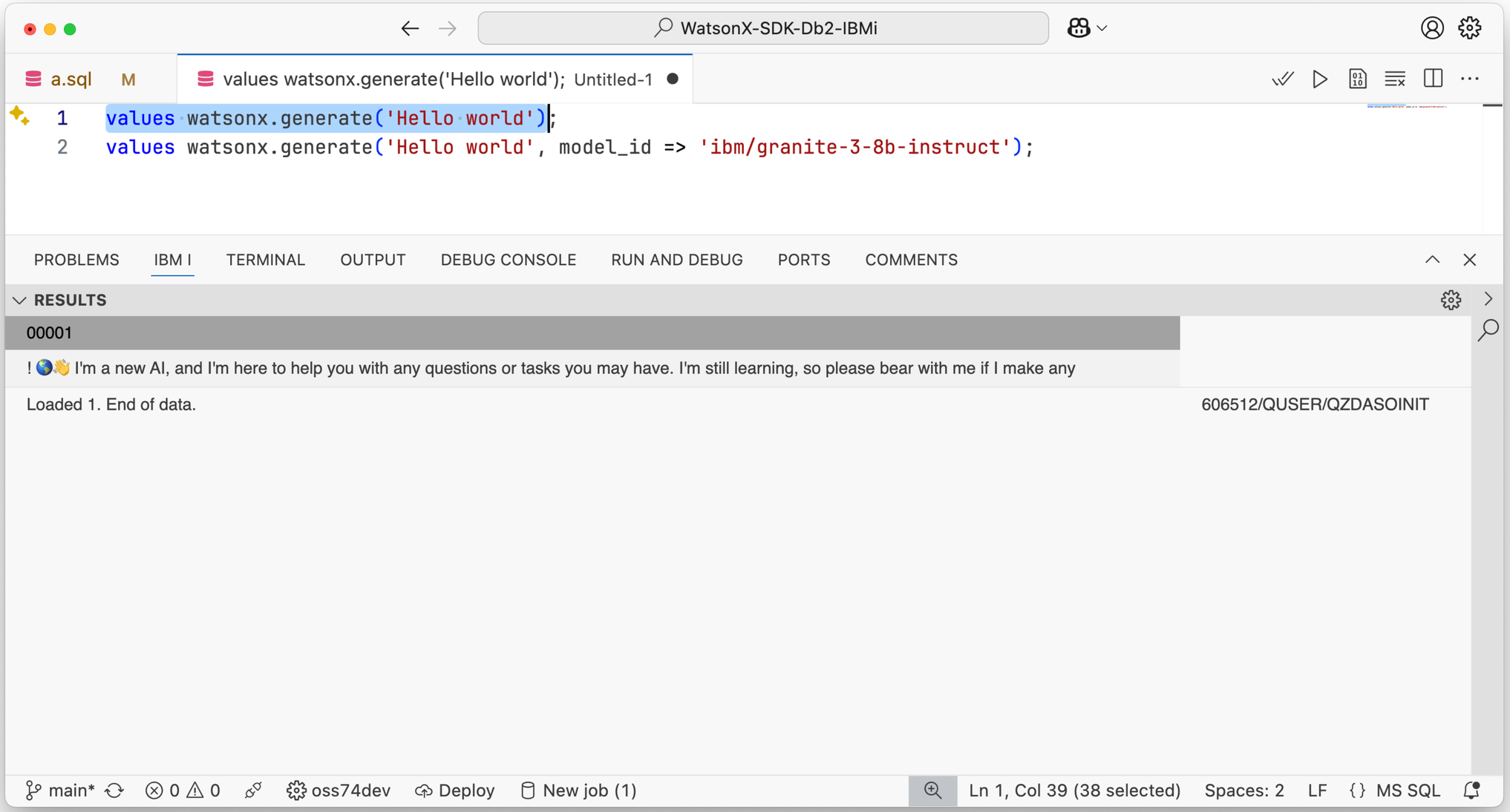
AI SDK: Watsonx

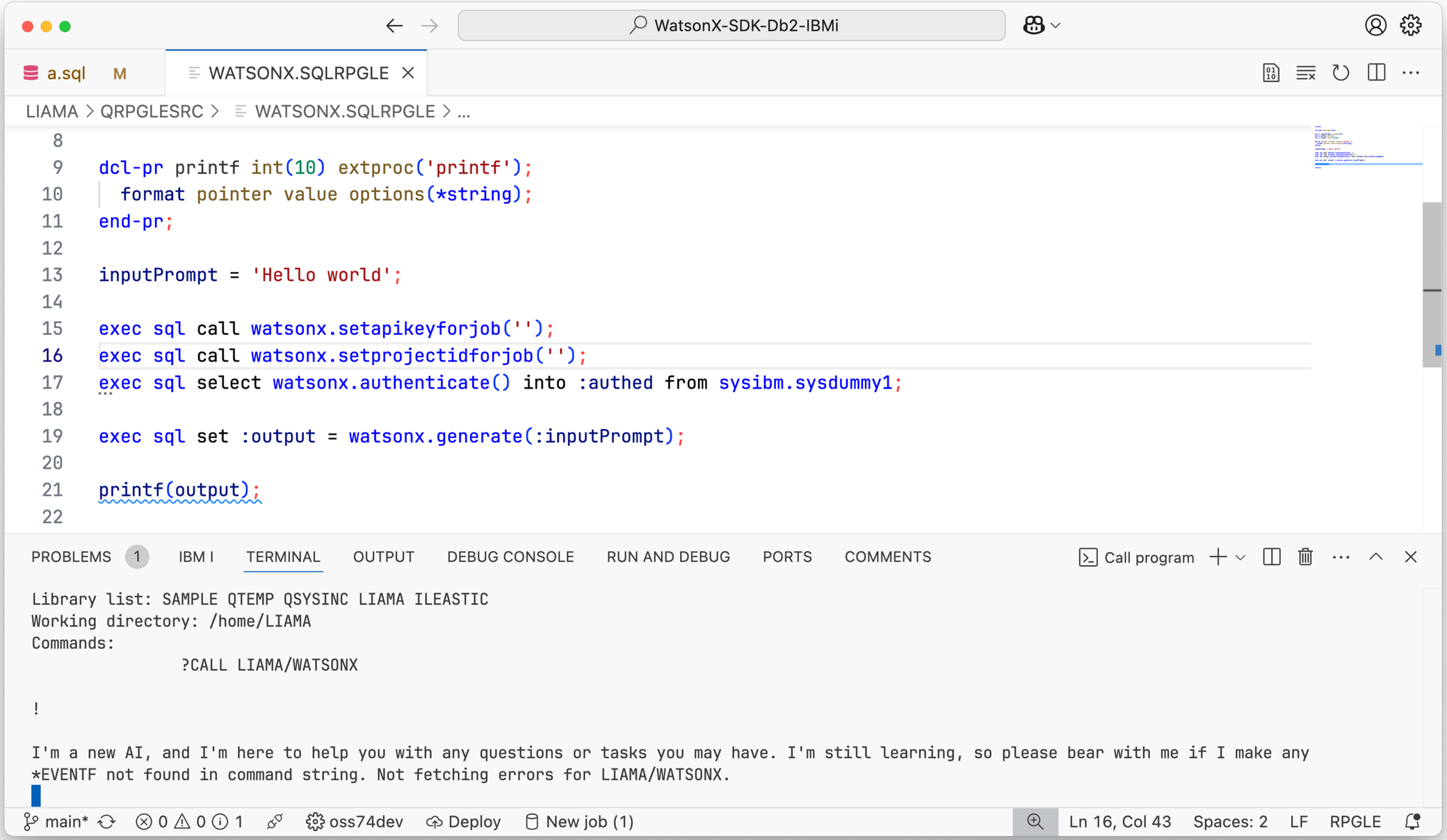
Set API key and project ID (configured in watsonx)

Authenticate with watsonx

```
call dbsdk_v1.wx_logoutjob();
call dbsdk_v1.wx_setapikeyforjob('x');
call dbsdk_v1.wx_setprojectidforjob('y');

-- Should return Y
values dbsdk_v1.wx_ShouldGetNewToken();

-- Should return Y
values dbsdk_v1.wx_authenticate();

-- Should retur
values dbsdk_v1.wx_ShouldGetNewToken();
```

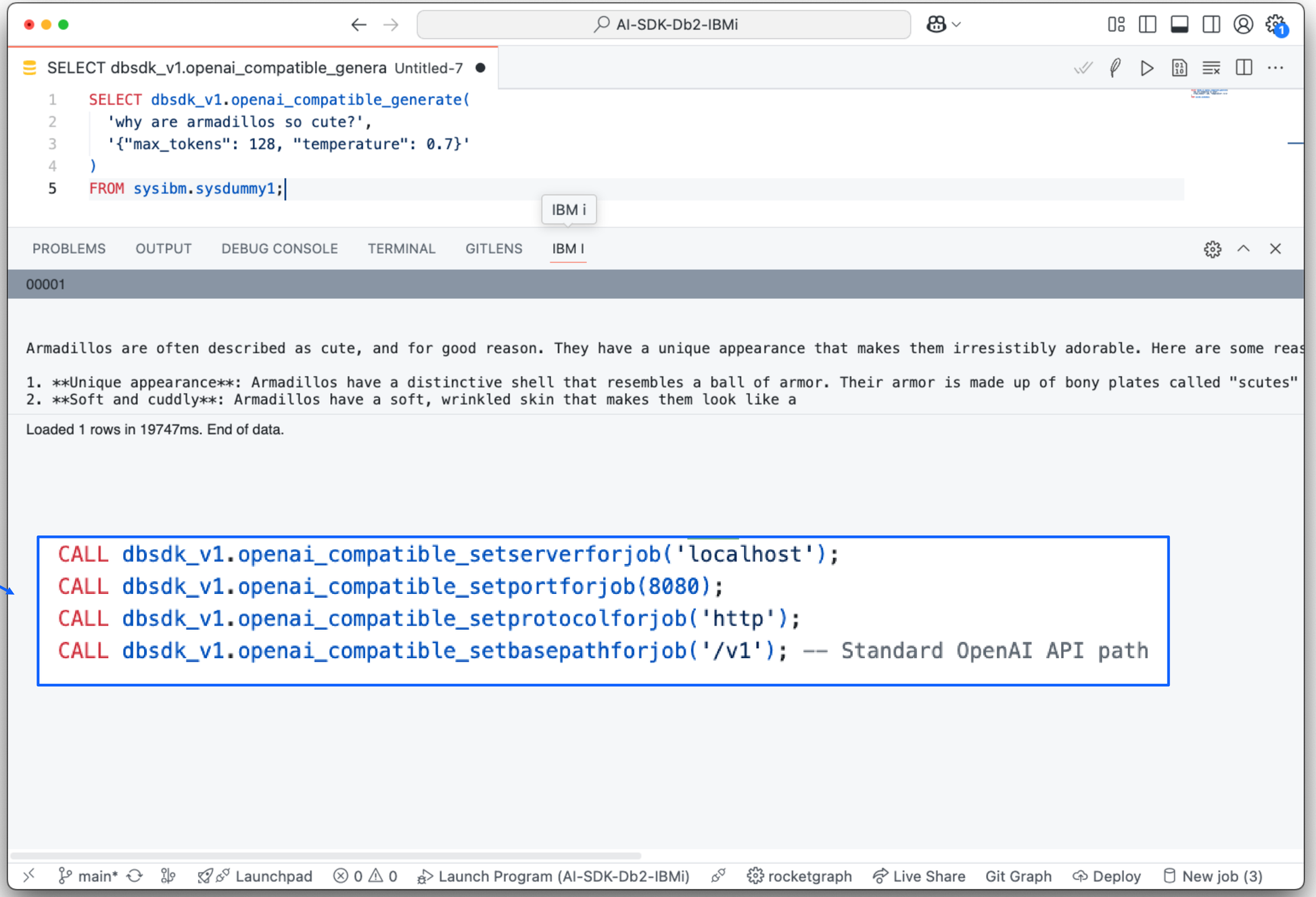

WatsonX-SDK-Db2-IBMi
call watsonx.getmodels(); Untitled-1
1 call watsonx.getmodels();
PROBLEMS
IBM I
TERMINAL
OUTPUT
DEBUG CONSOLE
RUN AND DEBUG
PORTS
COMMENTS
RESULTS
MODEL_ID | LABEL | PROVIDER | SHORT_DESCRIPTION
codellama/codellama-34b-instruct-hf | codellama-34b-instruct-hf | Code Llama | Code Llama is an AI model built on t…
cross-encoder/ms-marco-minilm-l-1… | ms-marco-minilm-l-12-v2 | cross-encoder | Used for Information Retrieval: Enc…
google/flan-t5-xl | flan-t5-xl-3b | Google | A pretrained T5 - an encoder-decod…
google/flan-t5-xxl | flan-t5-xxl-11b | Google | flan-t5-xxl is an 11 billion parameter …
google/flan-ul2 | flan-ul2-20b | Google | flan-ul2 is an encoder decoder mod…
ibm/granite-13b-chat-v2 | granite-13b-chat-v2 | IBM | The Granite model series is a family…
ibm/granite-13b-instruct-v2 | granite-13b-instruct-v2 | IBM | The Granite model series is a family…
ibm/granite-20b-code-instruct | granite-20b-code-instruct | IBM | The Granite model series is a family…
ibm/granite-20b-multilingual | granite-20b-multilingual | IBM | The Granite model series is a family…
ibm/granite-3-2b-instruct | granite-3-2b-instruct | IBM | The Granite model series is a family…
ibm/granite-3-8b-instruct | granite-3-8b-instruct | IBM | The Granite model series is a family…
ibm/granite-34b-code-instruct | granite-34b-code-instruct | IBM | The Granite model series is a family…
ibm/granite-3b-code-instruct | granite-3b-code-instruct | IBM | The Granite model series is a family…
Loaded 42. End of data.
606512/QUSER/QZDASOINIT
main*
0 0
oss74dev
Deploy
New job (1)
Ln 1, Col 26
Spaces: 2
LF
MS SQL


WatsonX-SDK-Db2-IBMi
a.sql M
values watsonx.generate('Hello world'); Untitled-1
1 values watsonx.generate('Hello world');
2 values watsonx.generate('Hello world', model_id => 'ibm/granite-3-8b-instruct');
PROBLEMS
IBM I
TERMINAL
OUTPUT
DEBUG CONSOLE
RUN AND DEBUG
PORTS
COMMENTS
RESULTS
00001
! 🌎👋 I'm a new AI, and I'm here to help you with any questions or tasks you may have. I'm still learning, so please bear with me if I make any
Loaded 1. End of data.
606512/QUSER/QZDASOINIT
main*
0 0
oss74dev
Deploy
New job (1)
Ln 1, Col 39 (38 selected)
Spaces: 2
LF
MS SQL


WatsonX-SDK-Db2-IBMi
a.sql M
WATSONX.SQLRPGLE
LIAMA > QRPGLESRC > WATSONX.SQLRPGLE > ...
8
9 dcl-pr printf int(10) extproc('printf');
10 format pointer value options(*string);
11 end-pr;
12
13 inputPrompt = 'Hello world';
14
15 exec sql call watsonx.setapikeyforjob('');
16 exec sql call watsonx.setprojectidforjob('');
17 exec sql select watsonx.authenticate() into :authed from sysibm.sysdummy1;
18
19 exec sql set :output = watsonx.generate(:inputPrompt);
20
21 printf(output);
22
PROBLEMS 1
IBM I
TERMINAL
OUTPUT
DEBUG CONSOLE
RUN AND DEBUG
PORTS
COMMENTS
Call program
Library list: SAMPLE QTEMP QSYSINC LIAMA ILEASTIC
Working directory: /home/LIAMA
Commands:
?CALL LIAMA/WATSONX
!
I'm a new AI, and I'm here to help you with any questions or tasks you may have. I'm still learning, so please bear with me if I make any
*EVENTF not found in command string. Not fetching errors for LIAMA/WATSONX.
main*
0 0 1
oss74dev
Deploy
New job (1)
Ln 16, Col 43
Spaces: 2
LF
RPGLE

AI SDK: OpenAI-Compatibility

- Plug in any OpenAI-compatible Endpoint
- Llama.cpp, Ollama, or any custom LLM deployment



Simple Demo

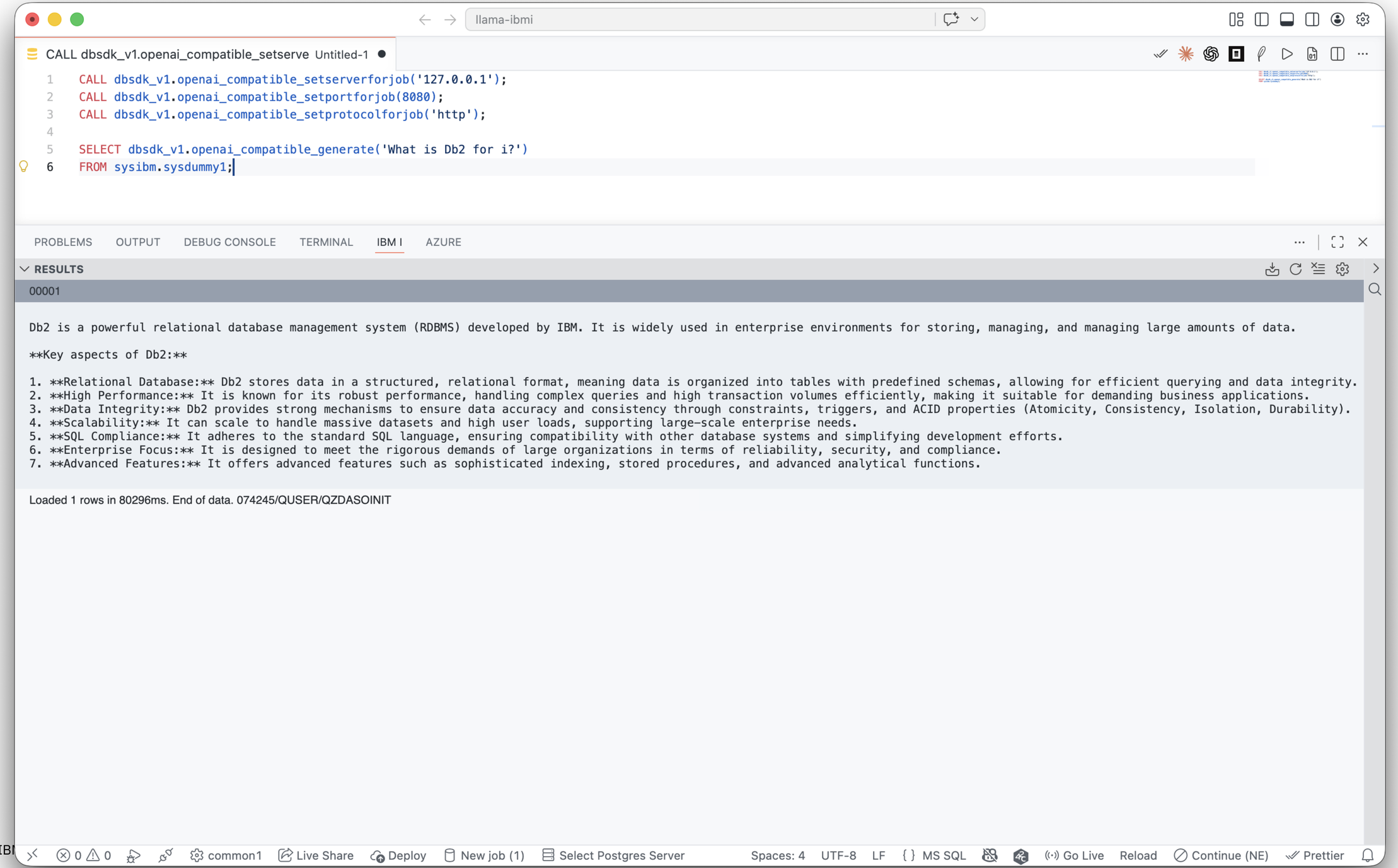
```
1  CALL dbsdk_v1.openai_compatible_setserverforjob('127.0.0.1');
2  CALL dbsdk_v1.openai_compatible_setportforjob(8080);
3  CALL dbsdk_v1.openai_compatible_setprotocolforjob('http');
4
5  SELECT dbsdk_v1.openai_compatible_generate('What is Db2 for i?')
6  FROM sysibm.sysdummy1;
```

Result set from response

RESULTS

00001

```
Db2 is a powerful relational database management system (RDBMS) developed by IBM. It is widely used in enterprise environments for storing, managing, and managing large amounts of data.

**Key aspects of Db2:**

1. **Relational Database:** Db2 stores data in a structured, relational format, meaning data is organized into tables with predefined schemas, allowing for efficient querying and data integrity.
2. **High Performance:** It is known for its robust performance, handling complex queries and high transaction volumes efficiently, making it suitable for demanding business applications.
3. **Data Integrity:** Db2 provides strong mechanisms to ensure data accuracy and consistency through constraints, triggers, and ACID properties (Atomicity, Consistency, Isolation, Durability).
4. **Scalability:** It can scale to handle massive datasets and high user loads, supporting large-scale enterprise needs.
5. **SQL Compliance:** It adheres to the standard SQL language, ensuring compatibility with other database systems and simplifying development efforts.
6. **Enterprise Focus:** It is designed to meet the rigorous demands of large organizations in terms of reliability, security, and compliance.
7. **Advanced Features:** It offers advanced features such as sophisticated indexing, stored procedures, and advanced analytical functions.
```


llama-ibmi
CALL dbsdk_v1.openai_compatible_setserve Untitled-1
1 CALL dbsdk_v1.openai_compatible_setserverforjob('127.0.0.1');
2 CALL dbsdk_v1.openai_compatible_setportforjob(8080);
3 CALL dbsdk_v1.openai_compatible_setprotocolforjob('http');
4
5 SELECT dbsdk_v1.openai_compatible_generate('What is Db2 for i?')
6 FROM sysibm.sysdummy1;
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
IBM I
AZURE
RESULTS
00001
Db2 is a powerful relational database management system (RDBMS) developed by IBM. It is widely used in enterprise environments for storing, managing, and managing large amounts of data.
Key aspects of Db2:
1. **Relational Database:** Db2 stores data in a structured, relational format, meaning data is organized into tables with predefined schemas, allowing for efficient querying and data integrity.
2. **High Performance:** It is known for its robust performance, handling complex queries and high transaction volumes efficiently, making it suitable for demanding business applications.
3. **Data Integrity:** Db2 provides strong mechanisms to ensure data accuracy and consistency through constraints, triggers, and ACID properties (Atomicity, Consistency, Isolation, Durability).
4. **Scalability:** It can scale to handle massive datasets and high user loads, supporting large-scale enterprise needs.
5. **SQL Compliance:** It adheres to the standard SQL language, ensuring compatibility with other database systems and simplifying development efforts.
6. **Enterprise Focus:** It is designed to meet the rigorous demands of large organizations in terms of reliability, security, and compliance.
7. **Advanced Features:** It offers advanced features such as sophisticated indexing, stored procedures, and advanced analytical functions.
Loaded 1 rows in 80296ms. End of data. 074245/QUSER/QZDASOINIT
0 0
common1
Live Share
Deploy
New job (1)
Select Postgres Server
Spaces: 4
UTF-8
LF
MS SQL
Go Live
Reload
Continue (NE)
Prettier

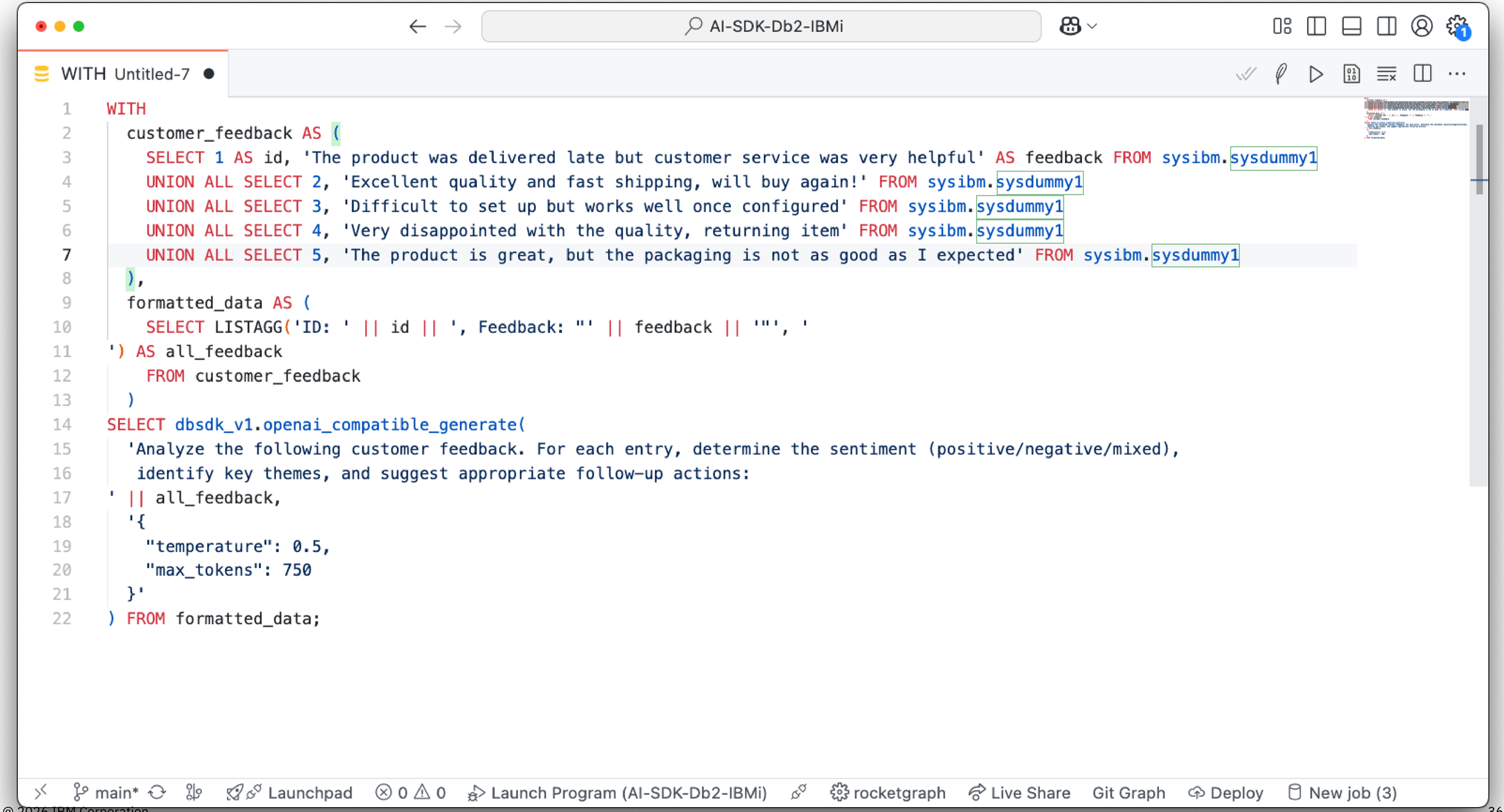
Use Cases

1. Summarize customer feedback
2. Analyze data trends
3. Generate and Store custom content

```sql
-- Create a temporary table with customer feedback
drop table coolstuff.feedback;

CREATE or REPLACE TABLE coolstuff.feedback (
  id INT PRIMARY KEY,
  comment VARCHAR(1000)
);

select * from coolstuff.feedback;

INSERT INTO coolstuff.feedback VALUES
  (1, 'I love your product, it works perfectly!'),
  (2, 'The software is good but the documentation could be better'),
  (3, 'This is terrible, nothing works as expected');

-- Classify each feedback
SELECT
  f.id,
  f.comment,
  dbsdk_v1.openai_compatible_generate(
    'Classify the following customer feedback into one of these categories: Positive, Neutral, Negative. Return just a JSON string with a single "category" field

    Feedback: ' || f.comment,
    '{
      "temperature": 0.1,
      "max_tokens": 30
    }'
  ) AS classification
FROM coolstuff.feedback f;
```

RESULTS

ID	COMMENT	CLASSIFICATION
1	I love your product, it works perfectly!	```json { "category": "Positive" } ```
2	The software is good but the documentation could be better	. ```json { "category": "Neutral" } ```
3	This is terrible, nothing works as expected	. ```json { "category": "Negative" } ```

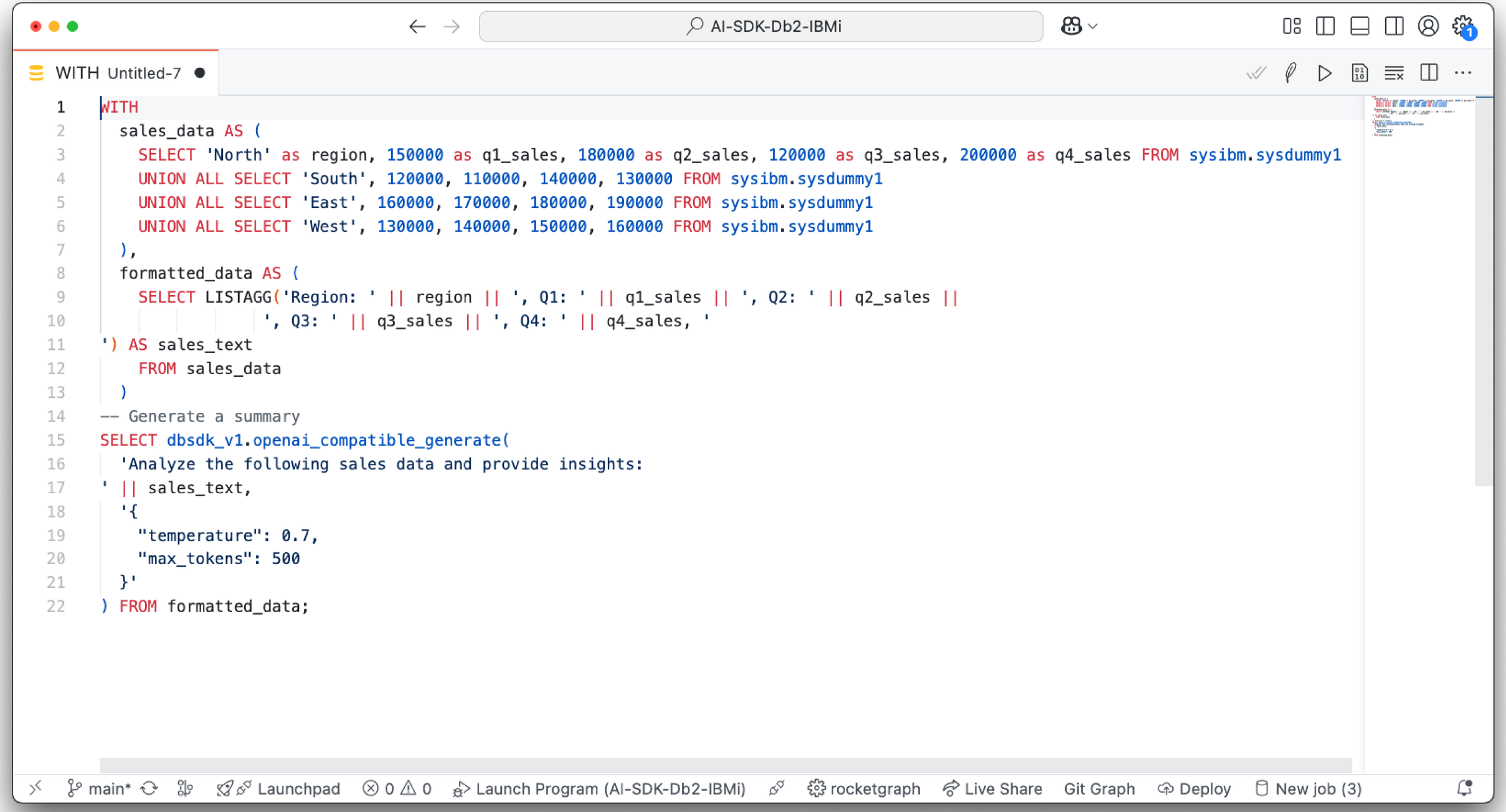


RESULTS

00001

```
---

### Analysis

**ID: 1**
*   **Sentiment:** Mixed
*   **Key Themes:** Delivery timeliness, Customer service responsiveness.
*   **Follow-up Actions:** Investigate the delay in delivery logistics. Acknowledge and commend the customer service team for their helpfulness.

**ID: 2**
*   **Sentiment:** Positive
*   **Key Themes:** Product quality, Shipping speed, Customer loyalty.
*   **Follow-up Actions:** Continue current processes for quality control and shipping efficiency. Encourage repeat purchases through targeted marketing.

**ID: 3**
*   **Sentiment:** Mixed
*   **Key Themes:** Ease of setup, Functionality after setup.
*   **Follow-up Actions:** Improve the initial setup instructions/tutorials. Gather feedback on the setup process to identify friction points.

**ID: 4**
*   **Sentiment:** Negative
*   **Key Themes:** Product quality, Return process.
*   **Follow-up Actions:** Conduct an immediate quality audit on the specific product batch. Review the return policy process for clarity and customer experience.

**ID: 5**
*   **Sentiment:** Mixed
*   **Key Themes:** Product satisfaction, Packaging quality.
*   ** Actions:** Review packaging materials and design to enhance protection during transit.

---

### Final Review

The analysis is thorough and the sentiment determination and suggested actions are logical and actionable. The structure is clear and directly addresses the prompt for each ID. No significant err
```

Use case: Analyze data trends (data summarization)

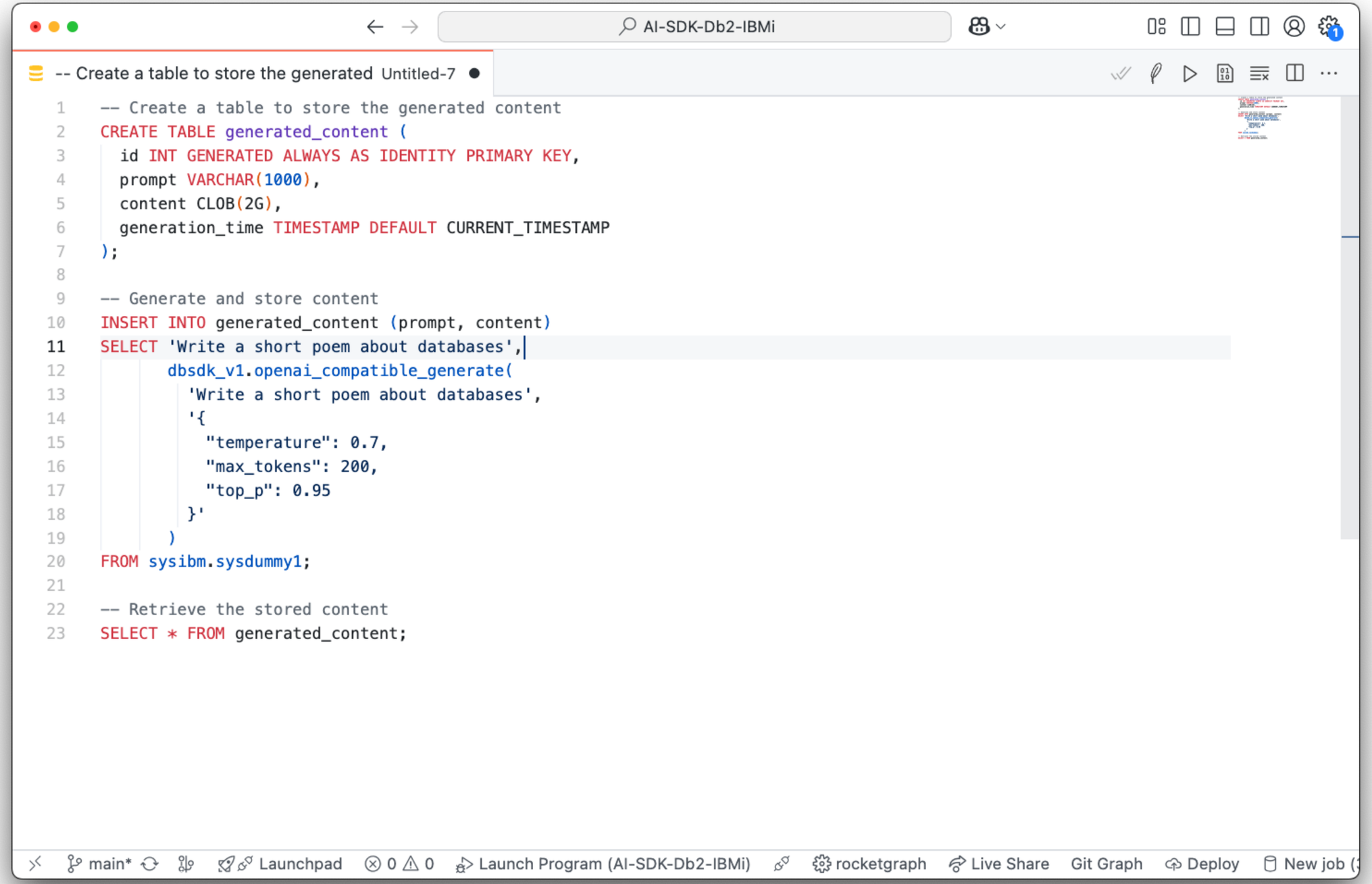


RESULTS

00001

```
---

## Sales Data Analysis

### 1. Overall Performance Overview

To get a holistic view, let's calculate the total sales for each region across all four quarters.

*   **North Total:** $150,000 + 180,000 + 120,000 + 200,000 = \$650,000$
*   **South Total:** $120,000 + 110,000 + 140,000 + 130,000 = \$500,000$
*   **East Total:** $160,000 + 170,000 + 180,000 + 190,000 = \$700,000$
*   **West Total:** $130,000 + 140,000 + 150,000 + 160,000 = \$580,000$

**Insight:** The **East** region generated the highest total sales ($\$700,000$), followed closely by the **North** ($\$650,000$). The **South** region had the lowest total sales ($\$500,000$).

### 2. Quarterly Trend Analysis (Seasonality)

Let's examine the trend within each region to identify seasonal patterns.

*   **North:** $150k \rightarrow 180k \rightarrow 120k \rightarrow 200k$ (Fluctuating, with a dip in Q3)
*   **South:** $120k \rightarrow 110k \rightarrow 140k \rightarrow 130k$ (Shows a dip in Q2)
*   **East:** $160k \rightarrow 170k \rightarrow 180k \rightarrow 190k$ (Consistent upward trend)
*   **West:** $130k \rightarrow 140k \rightarrow 150k \rightarrow 160k$ (Consistent upward trend)
```

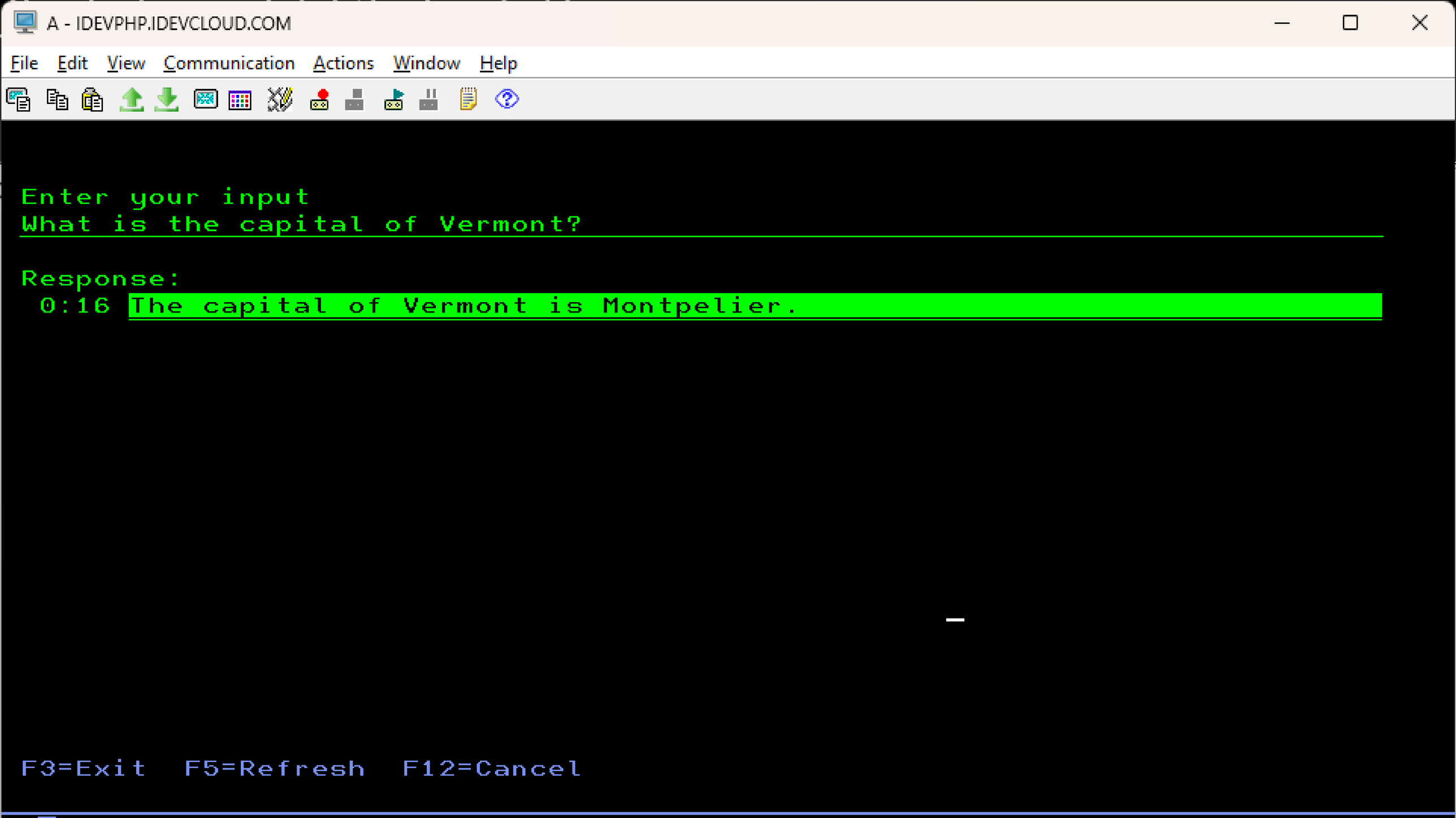
Use Case: Generate and Store custom content


AI-SDK-Db2-IBMi
-- Create a table to store the generated Untitled-7
-- Create a table to store the generated content
CREATE TABLE generated_content (
id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
prompt VARCHAR(1000),
content CLOB(2G),
generation_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- Generate and store content
INSERT INTO generated_content (prompt, content)
SELECT 'Write a short poem about databases',
dbsdk_v1.openai_compatible_generate(
'Write a short poem about databases',
'{
"temperature": 0.7,
"max_tokens": 200,
"top_p": 0.95
}'
)
FROM sysibm.sysdummy1;
-- Retrieve the stored content
SELECT * FROM generated_content;
main*
Launchpad
0 0
Launch Program (AI-SDK-Db2-IBMi)
rocketgraph
Live Share
Git Graph
Deploy
New job

RESULTS

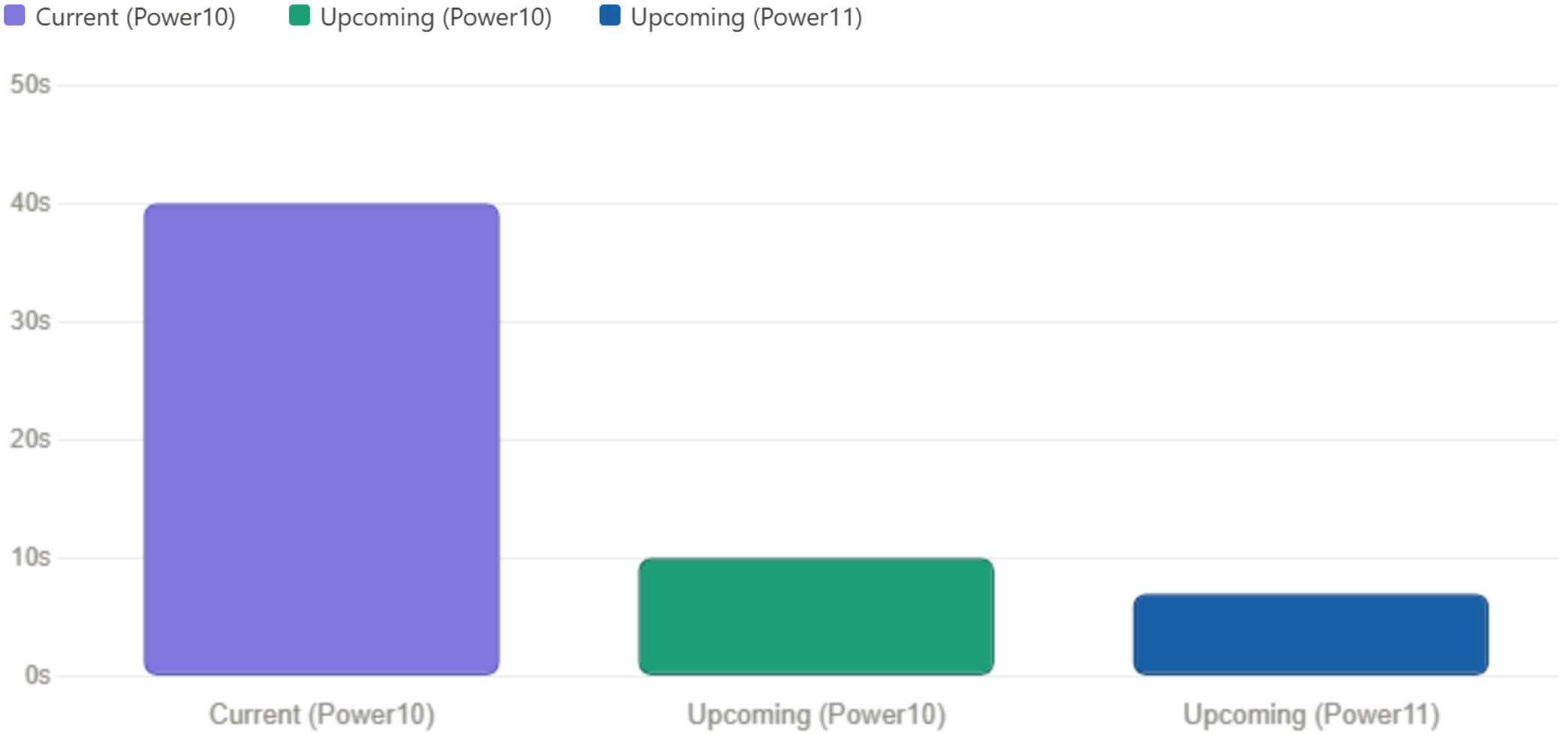
ID	PROMPT	CONTENT	GENERATION_TIME
1	Write a short poem about databases	. In structured rows, Data neatly stored, Indexed for quick retrieval, A digital treasure. Relational links bind, Tables in perfect alignment, SQL commands command, Knowledge in alignment. A digital vault, secure and deep, Where information safely sleeps.	2026-04-28 17:01:04.250550

```
exfmt test;
dow not (*in03 or *in12);
   if *in05;
      clear usertext;
      clear response;
   else;
     exec sql select
                LEFT( dbsdk_v1.wallaroo_generate(:usertext :text_null), 65)
                into :response :response_null
                from sysibm.sysdummy1;
     *in99 = *off;
     if sqlstate <> '00000';
        exec sql get diagnostics condition 1
                                    :errormsg = MESSAGE_TEXT;
        response = errormsg;
        *in99 = *on;
     elseif response_null <> 0;
        response = 'Null response';
     endif;
   endif;
   exfmt test;
enddo;
```


A - IDEVPHP.IDEVCLOUD.COM
File Edit View Communication Actions Window Help
Enter your input
What is the capital of Vermont?
Response:
0:16 The capital of Vermont is Montpelier.
F3=Exit F5=Refresh F12=Cancel

Virtual Assistant Use Case

(benchmarked with Wallaroo AI Starter Kit)



Other features...

A Db2 for i SDK for connecting to various services

Search Ctrl K

- Home
- IBM watsonx
- Ollama
- OpenAI-compliant endpoints
- Kafka
  - Requisites
  - Usage
  - Examples
- PASE
- Slack
- SMS (Twilio)
- Other Useful links (external)

Examples

Publish a message

Then at some future time (even a different session):

```
values(watsonx.kafka_publish( TOPIC   => 'fun_facts',
                              KEY     => 'baritone43',
                              MSGDATA => 'The baritone is a very interesting musical instru
```

Publish a message with a per-job topic

Then at some future time (even a different session):

```
call watsonx.kafka_settopicforjob('fun_facts');
values(watsonx.kafka_publish( KEY     => 'baritone43',
                              MSGDATA => 'The baritone is a very interesting musical instru
```

Other features...

```
23 select * from table(jesseg.callpase('echo "this\n\nthat'))
```

OUTPUT
this
that

Overview: AI SDK for Db2 for i

Easy Db2 calls for:

- watsonx.ai
- Ollama
- OpenAI and OpenAI-compliant solutions
  - LLaMa.cpp
  - Rocket CE
  - vLLM
  - LM Studio
- Kafka
- Slack
- Twilio (SMS)
- PASE commands

Project Link: https://ibm.github.io/AI-SDK-Db2-IBMi/

Links!

- Llama.cpp Blog: https://ibm.biz/llamacpp-ibmi
- AI SDK for Db2 for i: https://ibm.github.io/AI-SDK-Db2-IBMi/
- Models for i: https://huggingface.co/models-for-i