MySQL
ORACLE

MySQL HeatWave AI on Steroïd

Unlocking the Power of OCI GenAI Models in MySQL HeatWave

Frédéric Descamps
Community Manager
Oracle MySQL
preFOSDEM MySQL Belgian Days - January 2026

Who am I ?

about.me/lefred

Frédéric Descamps

- 𝕏 *@lefred*
- *@lefredbe.bsky.social*
- *@lefred@fosstodon.org*
- [matrix] *@lefred14:matrix.org*
- *MySQL Evangelist*
- *using MySQL since version 3.20*
- *devops believer*
- *living in* 🇧🇪
- *https://lefred.be*

MySQL HeatWave AI on Steroïd

Agenda

Agenda

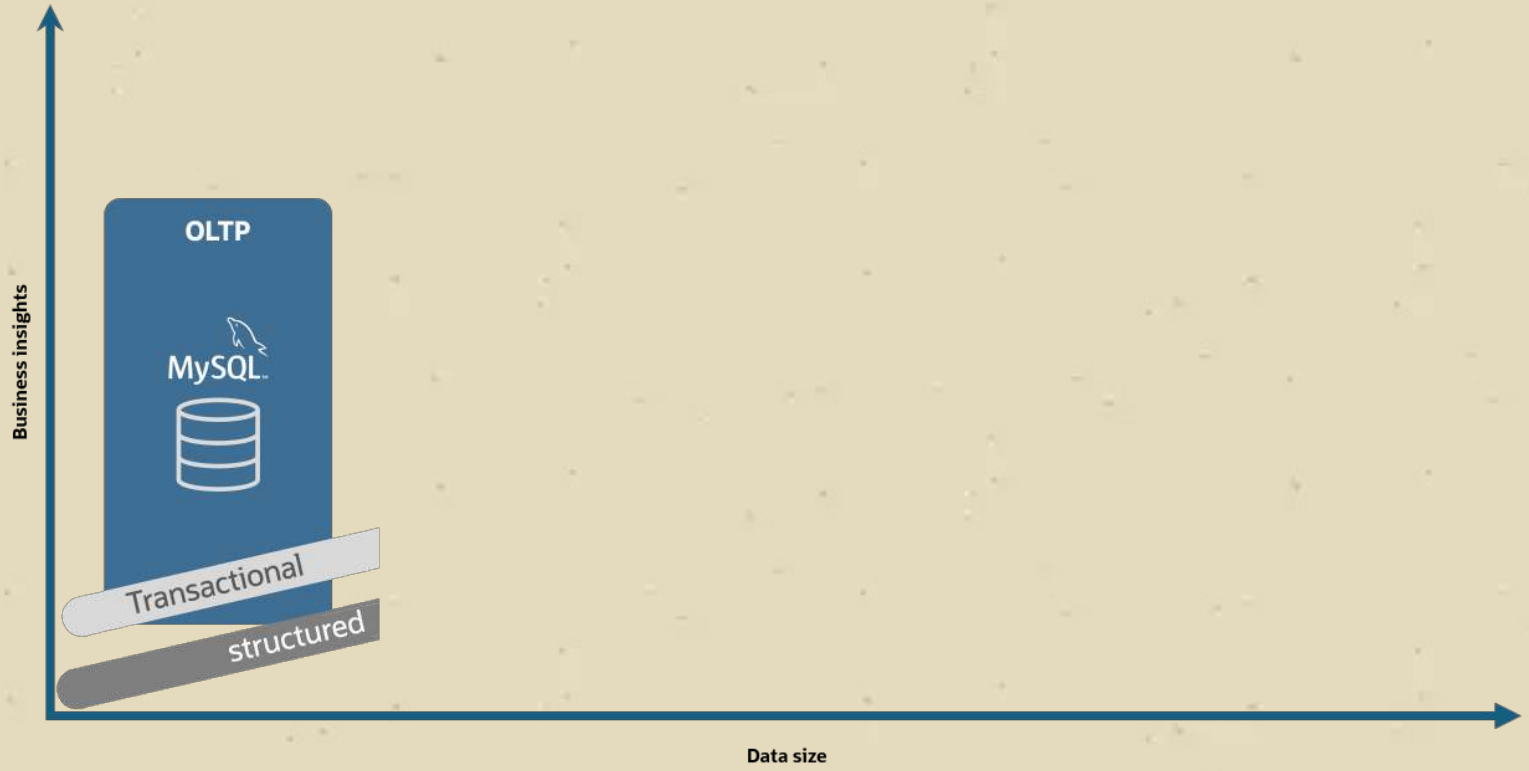
- *Introduction*
- *The Current HeatWave AI Capabilities*
- *Why OCI GenAI Models?*
- *Integrating OCI GenAI with MySQL HeatWave*
- *Benefits & Use Cases*
- *Comparative Performance*
- *Q&A Session*

MySQL HeatWave AI on Steroïd

Introduction

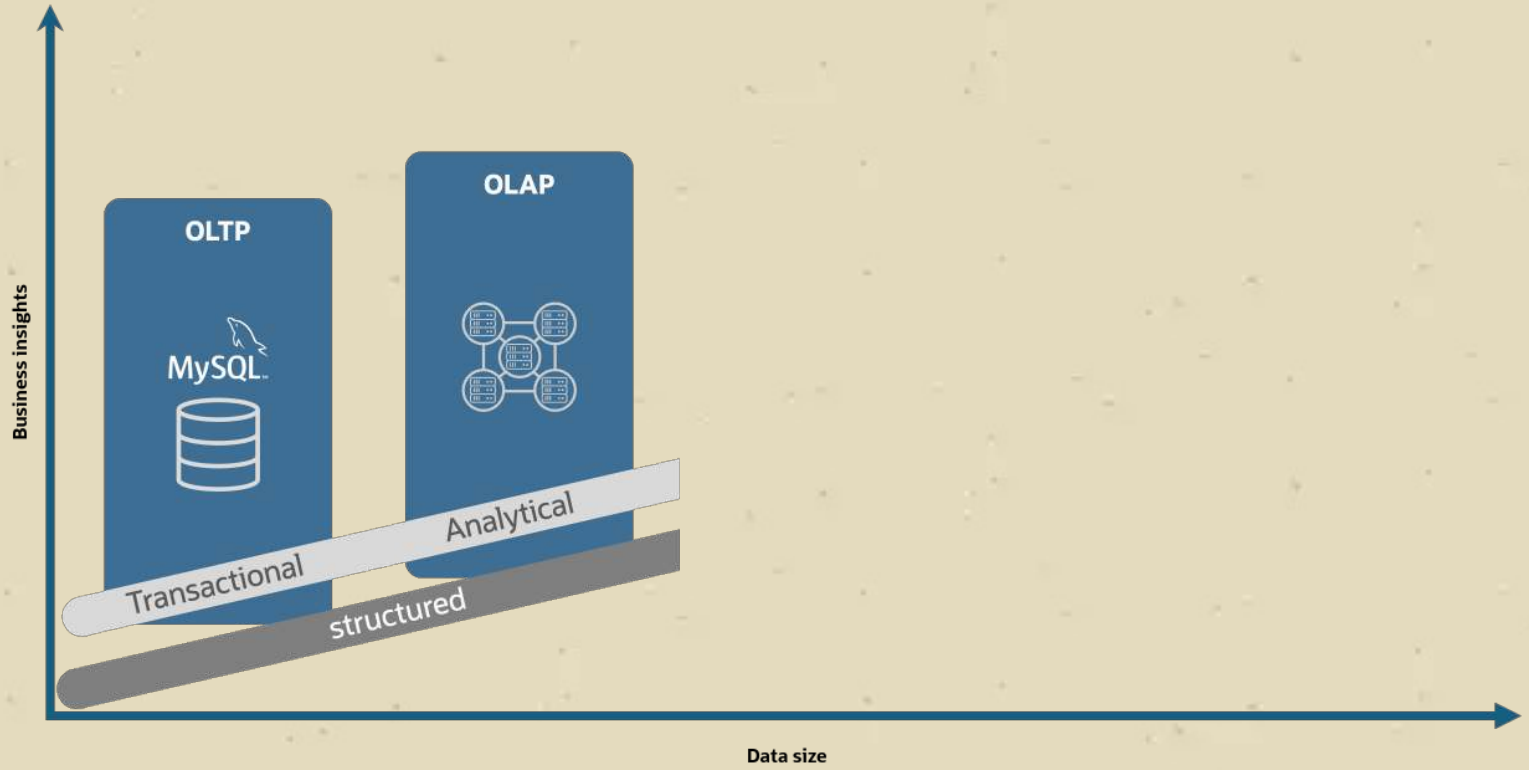
Introduction

MySQL HeatWave 🔥: DBaaS with high-performance in-memory query accelerator, ML capabilities, Lakehouse, AI and more.



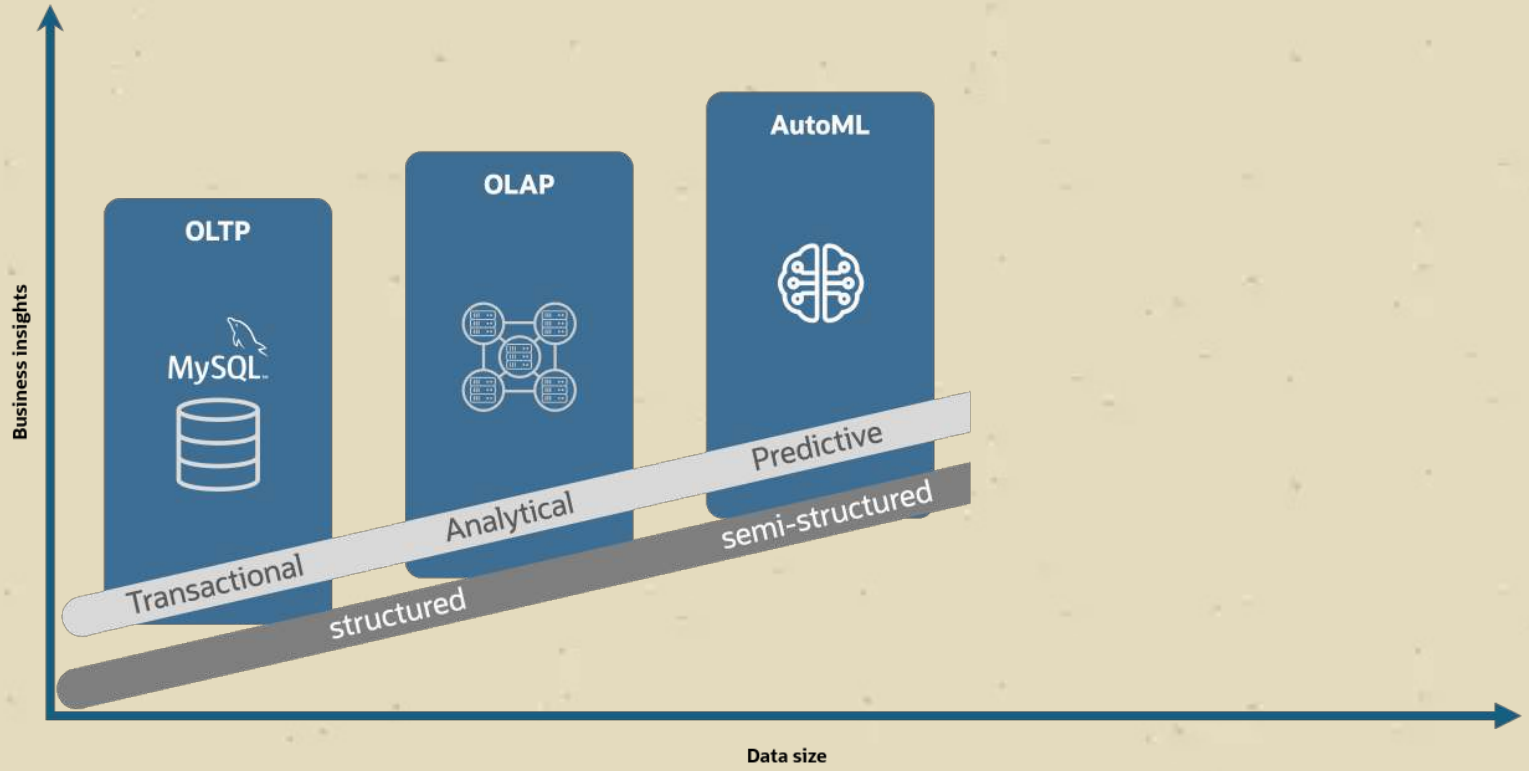
Introduction

MySQL HeatWave 🔥: DBaaS with high-performance in-memory query accelerator, ML capabilities, Lakehouse, AI and more.



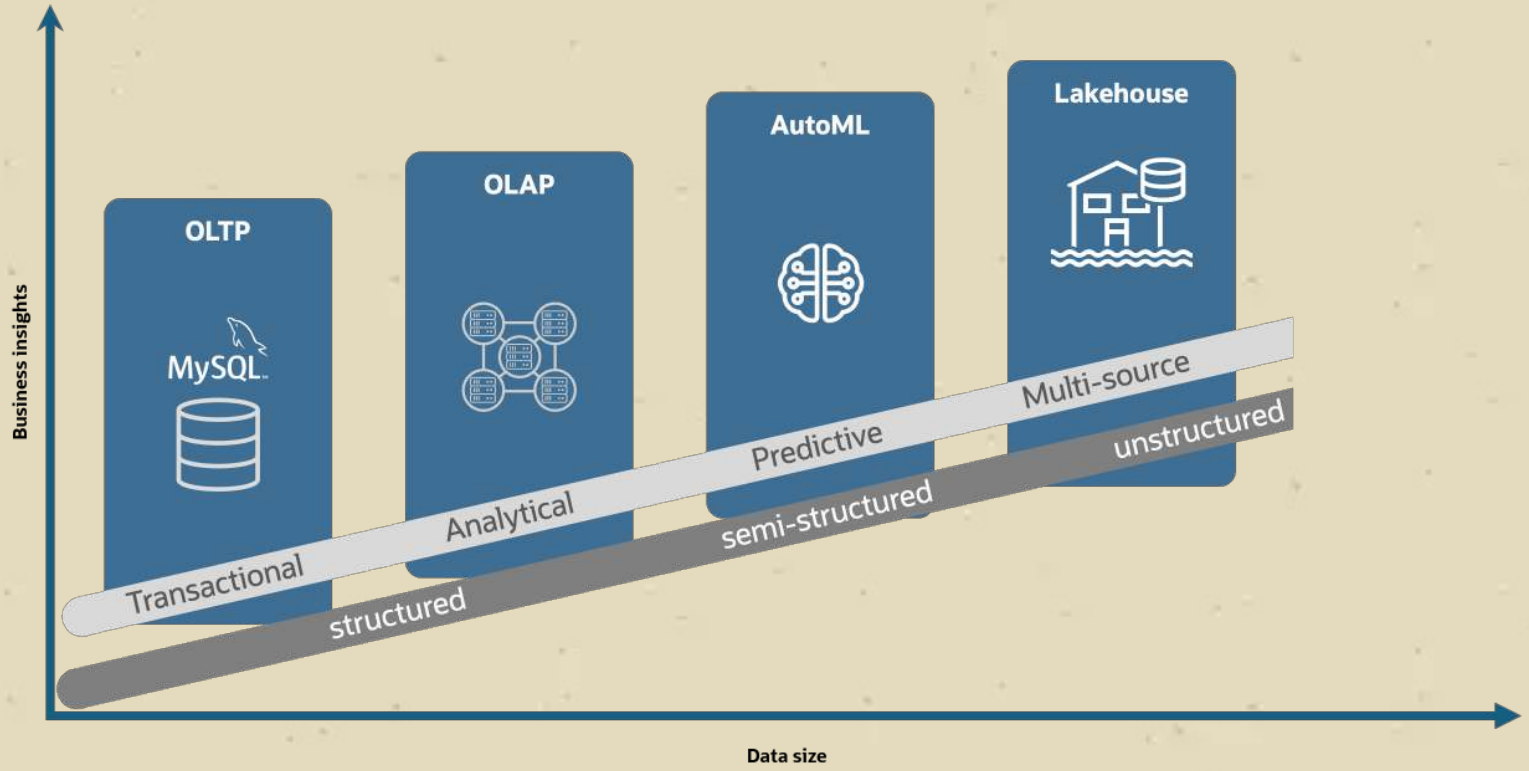
Introduction

MySQL HeatWave 🔥: DBaaS with high-performance in-memory query accelerator, ML capabilities, Lakehouse, AI and more.



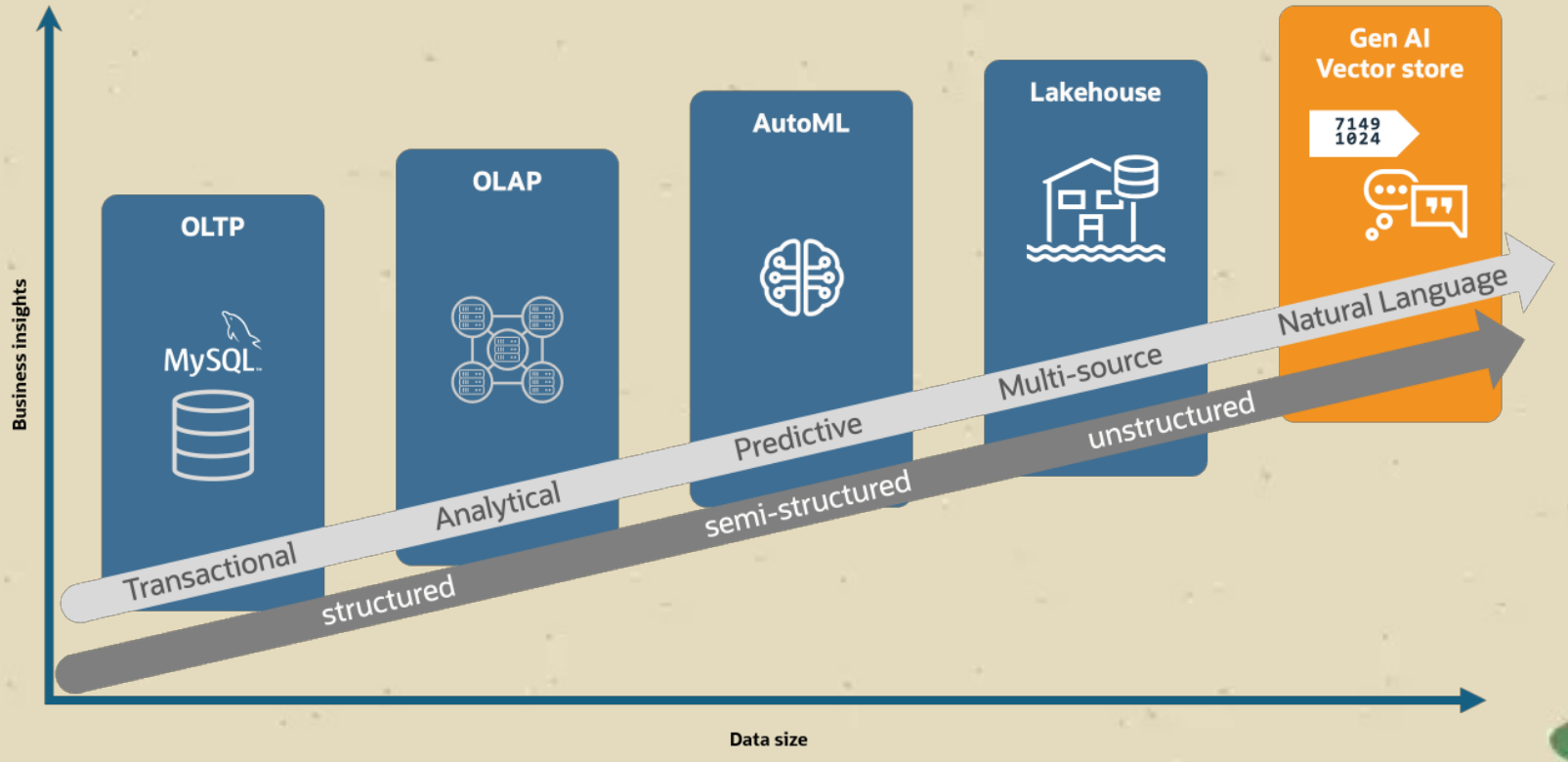
Introduction

MySQL HeatWave 🔥: DBaaS with high-performance in-memory query accelerator, ML capabilities, Lakehouse, AI and more.



Introduction

MySQL HeatWave 🔥: DBaaS with high-performance in-memory query accelerator, ML capabilities, Lakehouse, AI and more.



LLMs in MySQL HeatWave

Bundled CPU-optimized LLM models for AI tasks:

LLMs in MySQL HeatWave

Bundled CPU-optimized LLM models for AI tasks:

```
SQL> SELECT provider, model_id, availability_date, capabilities FROM sys.ML_SUPPORTED_LLMS;
+----------+--------------------------+-------------------+----------------------+
| provider | model_id                 | availability_date | capabilities         |
+----------+--------------------------+-------------------+----------------------+
| HeatWave | llama3.2-1b-instruct-v1  | 2025-05-20        | ["GENERATION"]       |
| HeatWave | llama3.2-3b-instruct-v1  | 2025-05-20        | ["GENERATION"]       |
| HeatWave | all_minilm_l12_v2        | 2024-07-01        | ["TEXT_EMBEDDINGS"]  |
| HeatWave | multilingual-e5-small    | 2024-07-24        | ["TEXT_EMBEDDINGS"]  |
+----------+--------------------------+-------------------+----------------------+
4 rows in set (1.9545 sec)
```

MySQL HeatWave AI on Steroïd

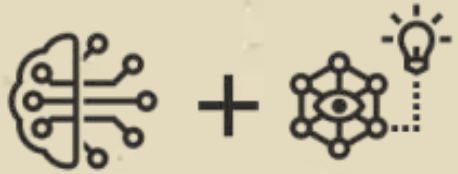
The Current HeatWave AI Capabilities

Current HeatWave AI Capabilities

Converstations in natural language interact with your data using simple text commands.

RAG & similarity search enhance data retrieval and analysis with advanced search capabilities.

Content generation and summarization generate reports, summaries, and insights from your data.



Synergy og integrated GenAI and ML combine generative AI with machine learning to save time and deliver more accurate results.

MySQL HeatWave AI on Steroïd

OCI GenAI Models

Current OCI GenAI Models

```
+-----------------------------------------+
| model_id                                |
+-----------------------------------------+
| meta.llama-4-maverick-17b-128e-instruct-fp8 |
| meta.llama-4-scout-17b-16e-instruct     |
| xai.grok-4-fast-non-reasoning           |
| xai.grok-4-fast-reasoning               |
| xai.grok-code-fast-1                    |
| xai.grok-3-mini-fast                    |
| xai.grok-3-fast                         |
| xai.grok-3                              |
| xai.grok-3-mini                         |
| xai.grok-4                              |
+-----------------------------------------+
```

Why OCI GenAI Models?

People might perfer OCI Generative AI Service models over the in-database HeatWave models for several reasons:

- *stronger output quality for harder tasks:*
  - *llama3.2 1B and 3B are good for basic tasks (small models)*
  - *OCI GenAI provides larger models with more parameters*
  - *trained on larger and more diverse datasets*
- *better in complex instructions and multi-step tasks*

OCI Generative AI Service gives you access to significantly larger / more capable families like Meta Llama 4 Scout/Maverick and xAI Grok.

Why OCI GenAI Models? (2)

- *Better specialization options (reasoning vs speed vs code)*
  OCI Generative AI Service provides variants that are explicitly positioned for different needs, e.g.:
  - *fast/low-latency options (Grok 4 Fast)*
  - *reasoning vs “non-reasoning”*
  - *code-focused model (e.g., xai.grok-code-fast-1)*

Grok 4 is strong for enterprise tasks like extraction, coding, summarization, and domain-heavy areas.

And Grok 4 Fast is quick (time-to-first-token/output speed) for real-time applications.

Why OCI GenAI Models? (3)

- *Availability and model breadth tend to move faster in OCI than in-DB*
  - *HeatWave in-database LLM availability can depend on your HeatWave cluster shape (and historically version/shape constraints).*
  - *Practically: OCI Generative AI Service is where Oracle can roll out new frontier models and variants faster and give you more choice without tying it to database node sizing.*

Integrating OCI GenAI Models with MySQL HeatWave

External LLMs are only available in selected regions and only after having authenticated HeatWave to OCI Generative AI Service.

Integrating OCI GenAI Models with MySQL HeatWave

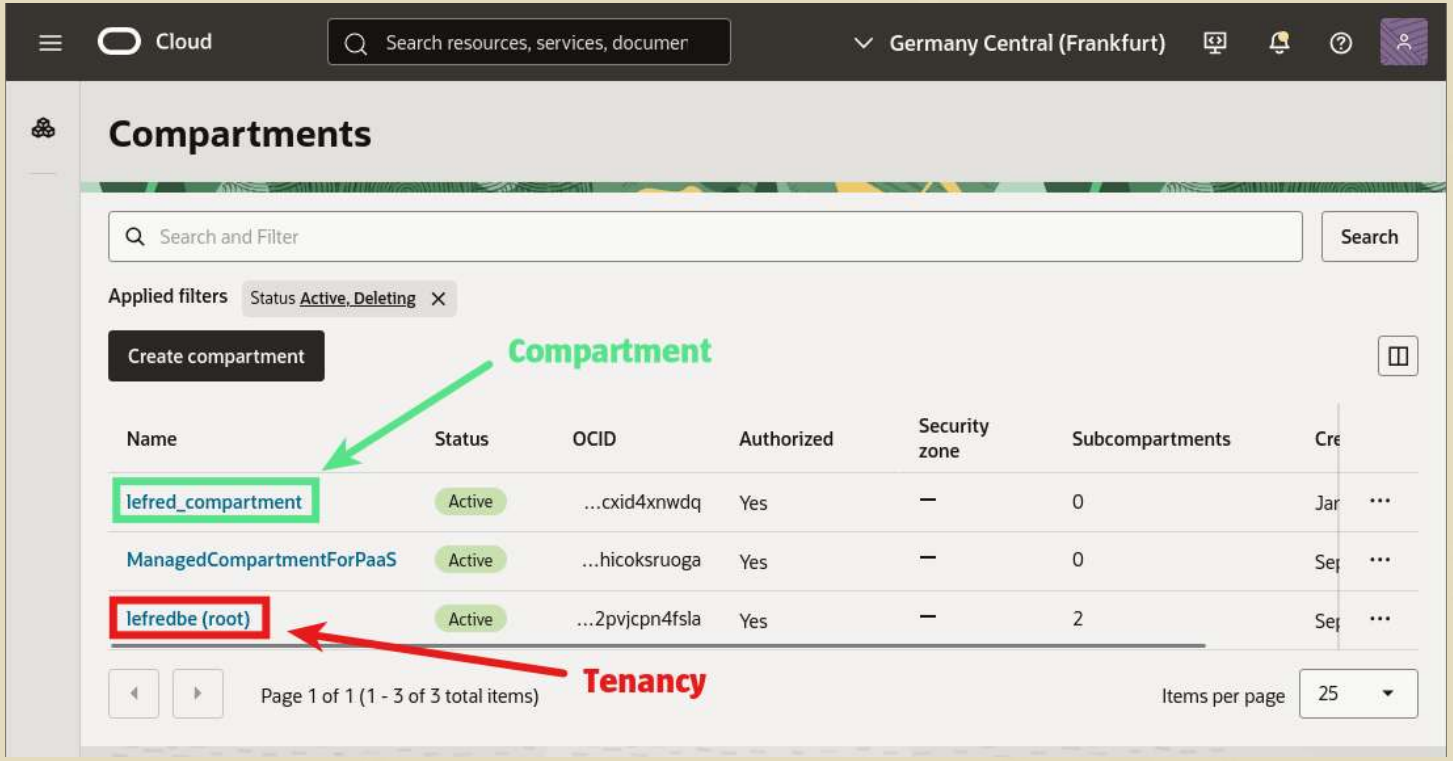
External LLMs are only available in selected regions and only after having authenticated HeatWave to OCI Generative AI Service.

... and that's probably the most complicated part !

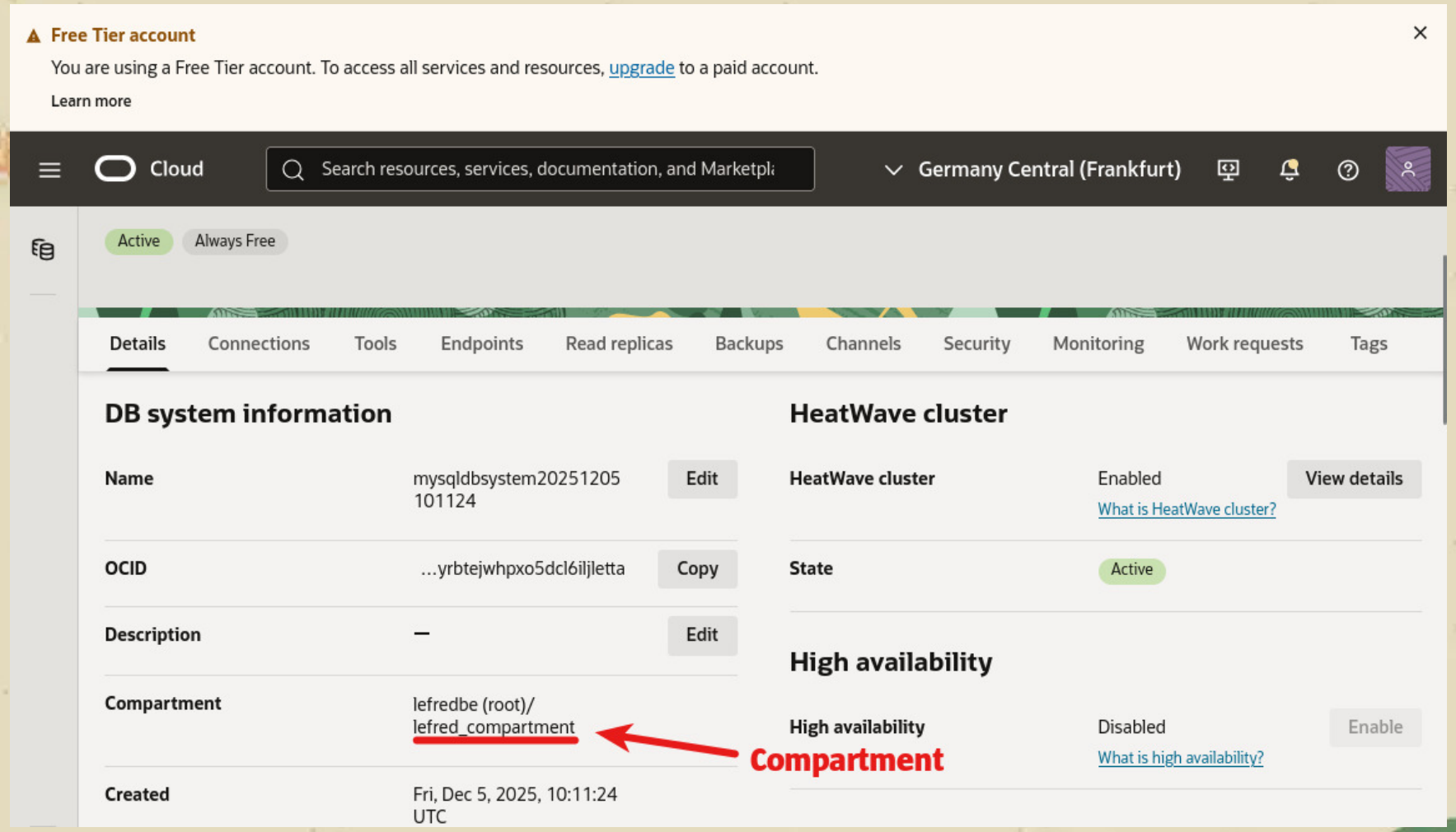
Authenticating MySQL HeatWave to OCI GenAI

Step 1

To grant such access, check whether your DB System is in your tenancy or in a dedicated compartment:

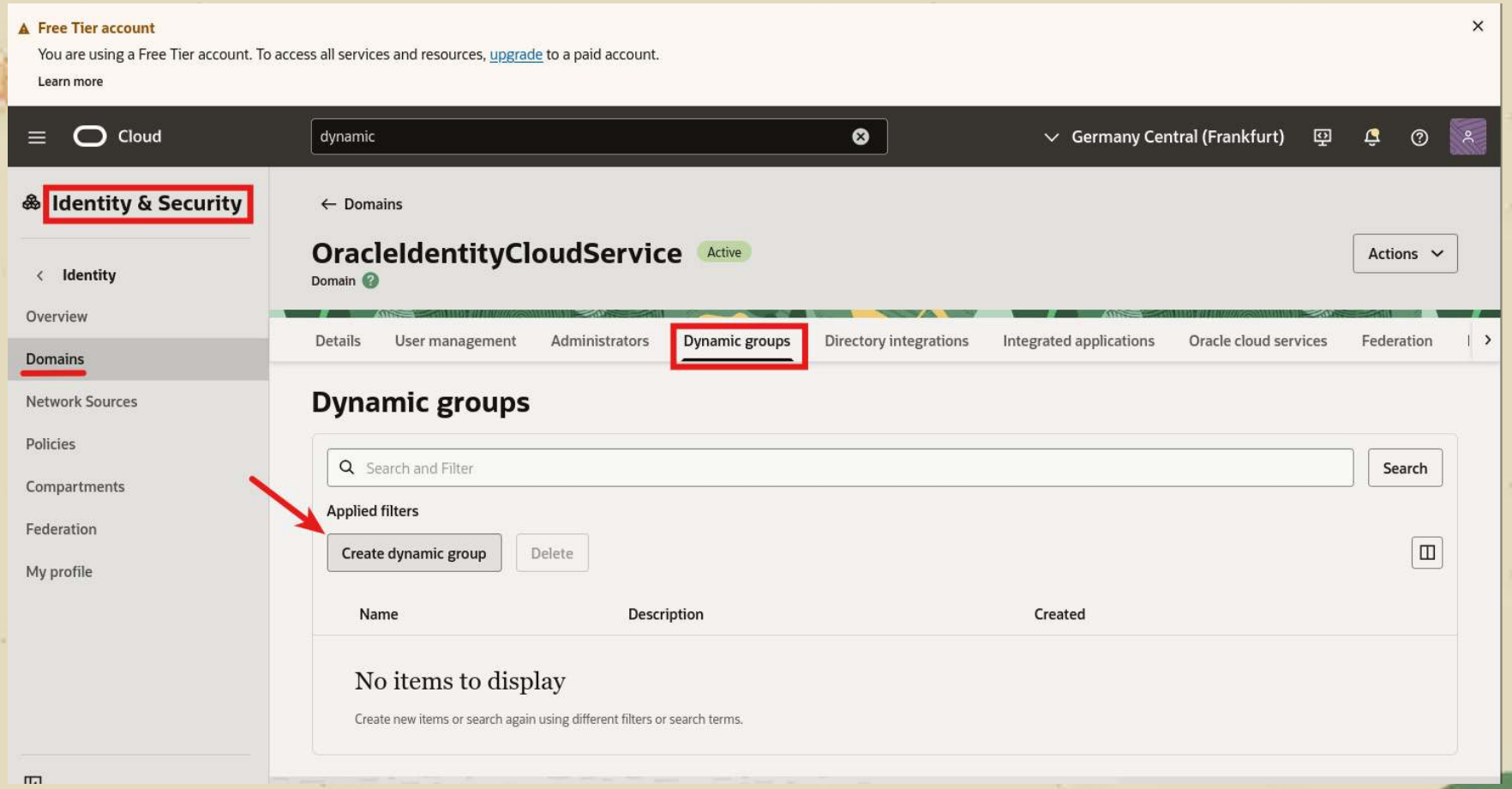


Authenticating MySQL HeatWave to OCI GenAI

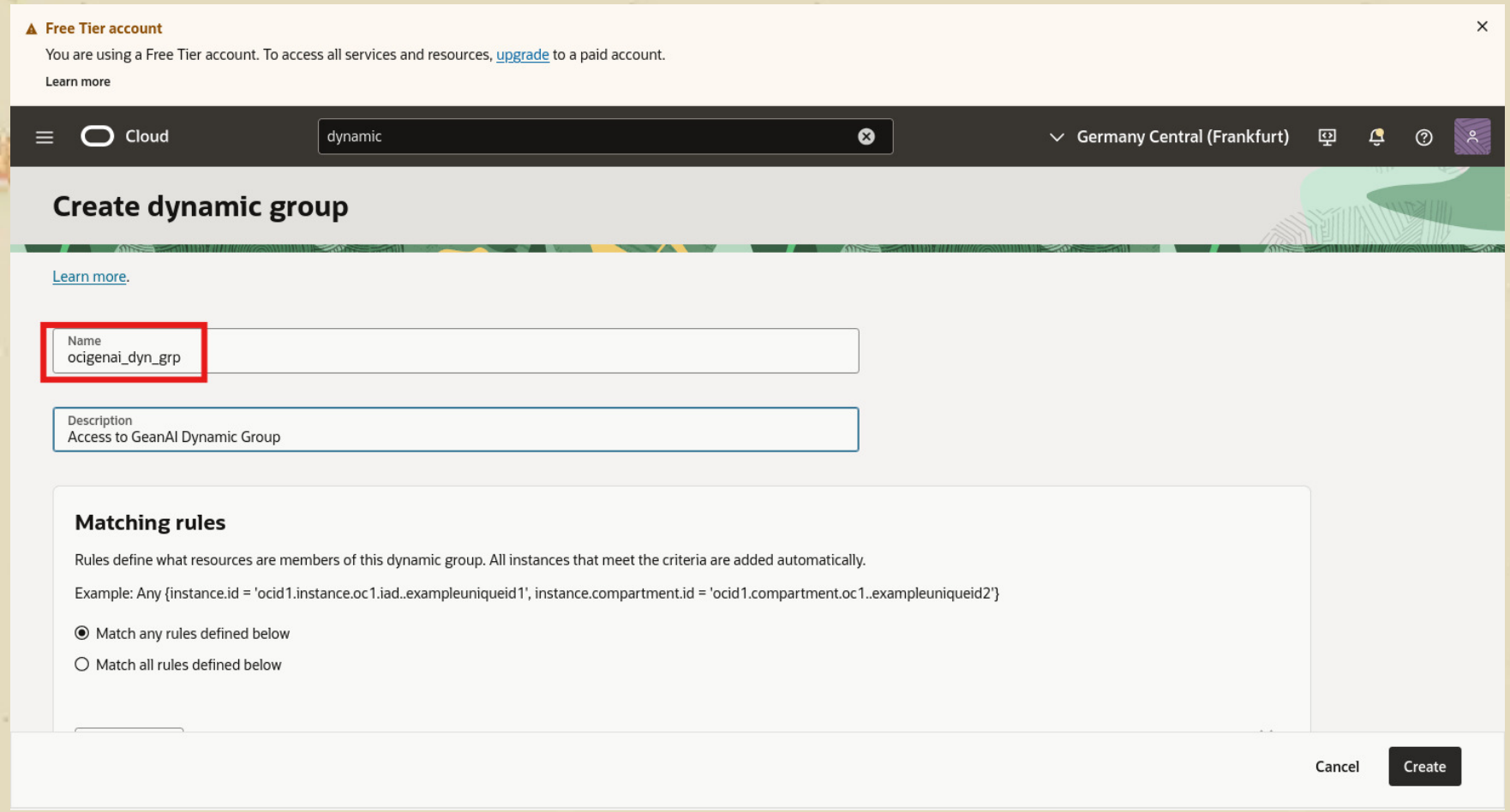


Authenticating MySQL HeatWave to OCI GenAI

Step 2 - Dynamic Group

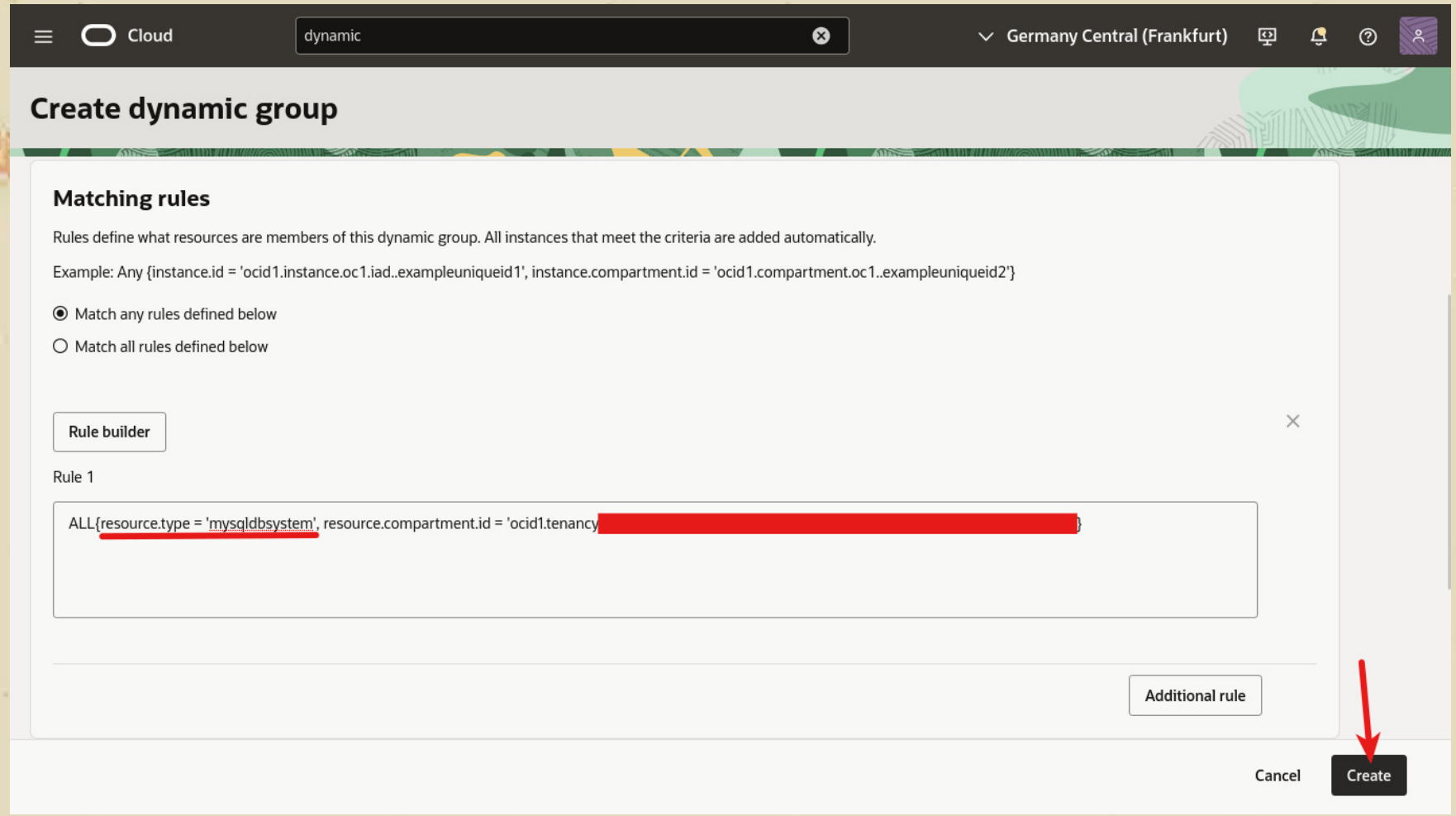


Authenticating MySQL HeatWave to OCI GenAI

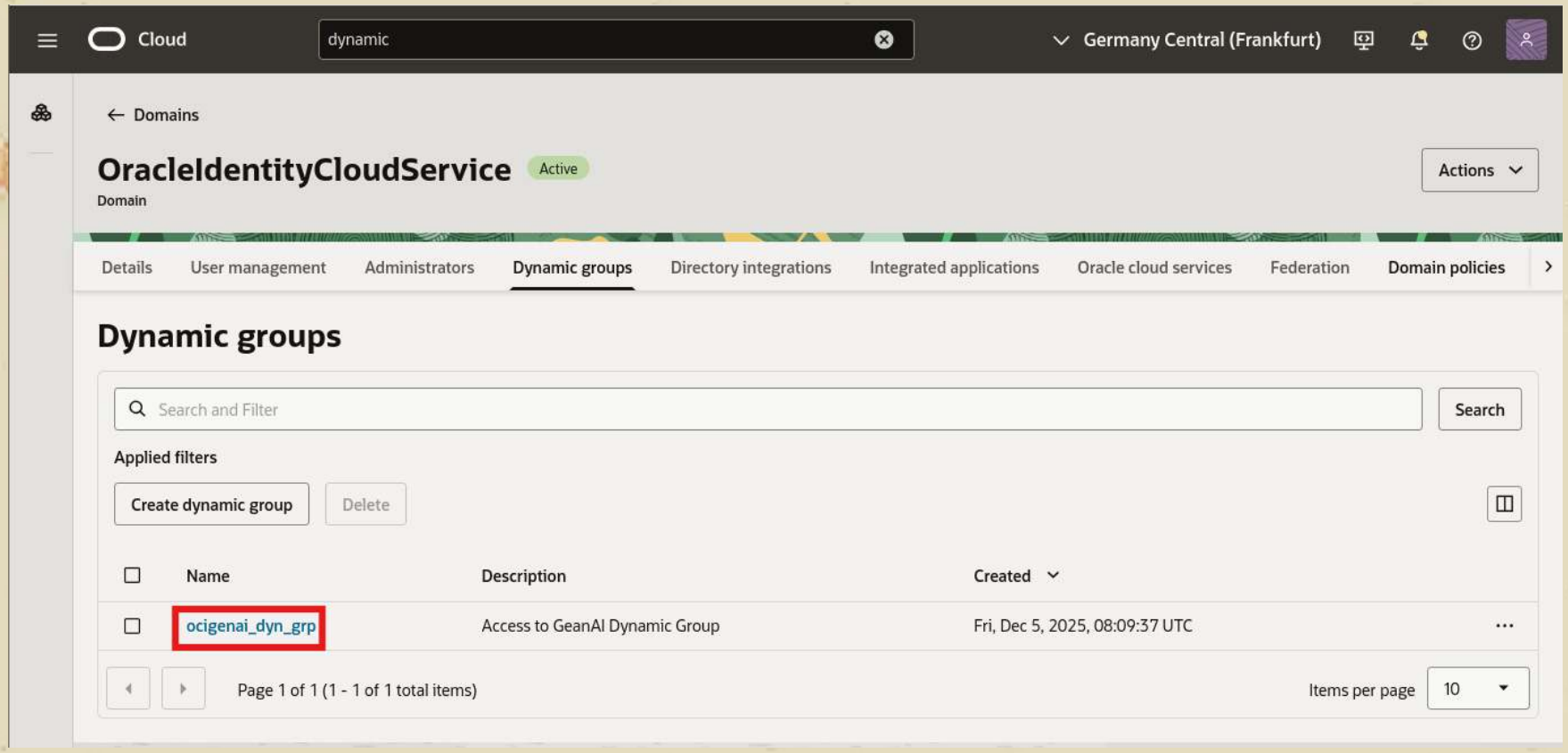


It's essential to remember the name used!

Authenticating MySQL HeatWave to OCI GenAI

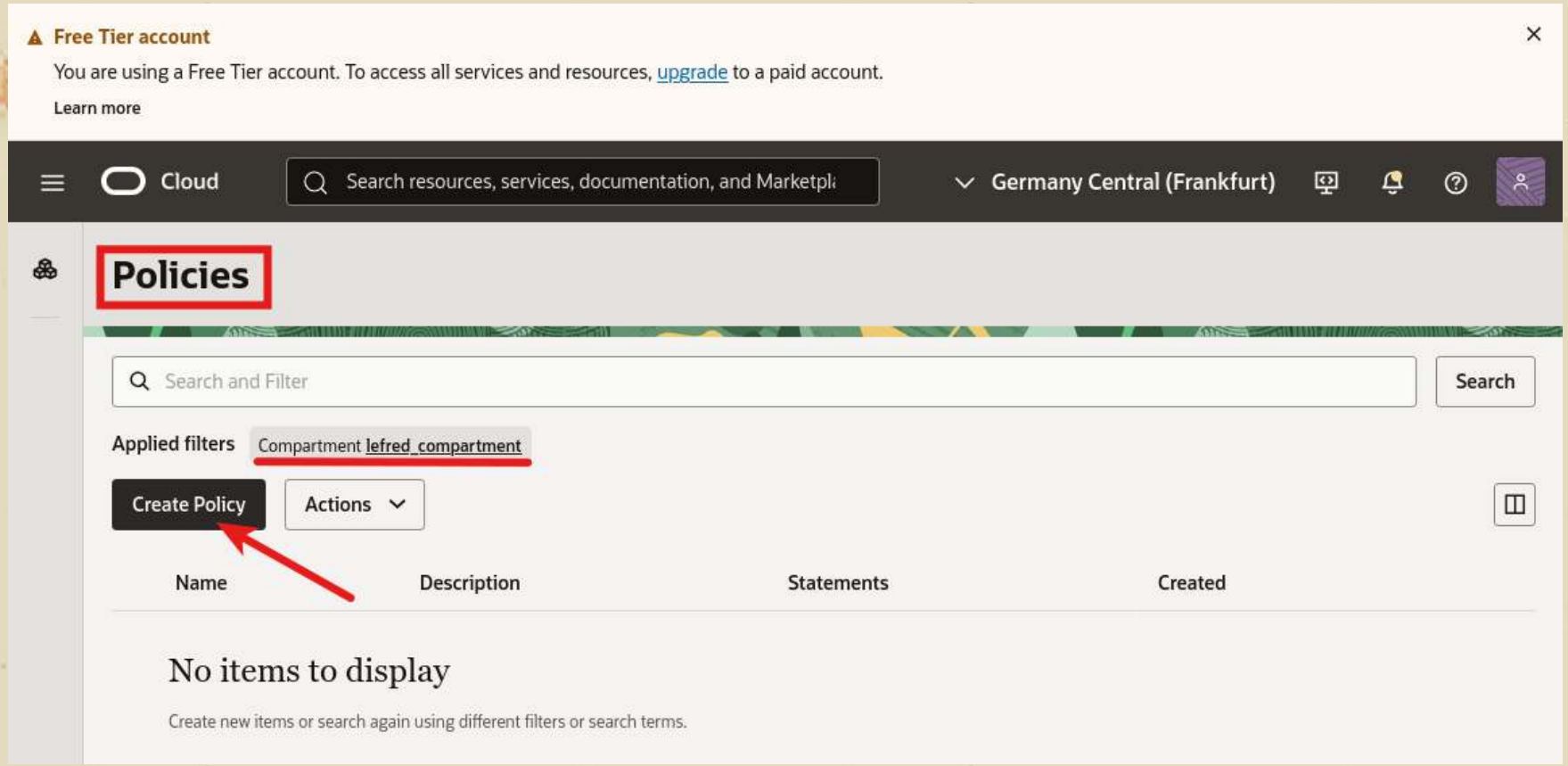


Authenticating MySQL HeatWave to OCI GenAI

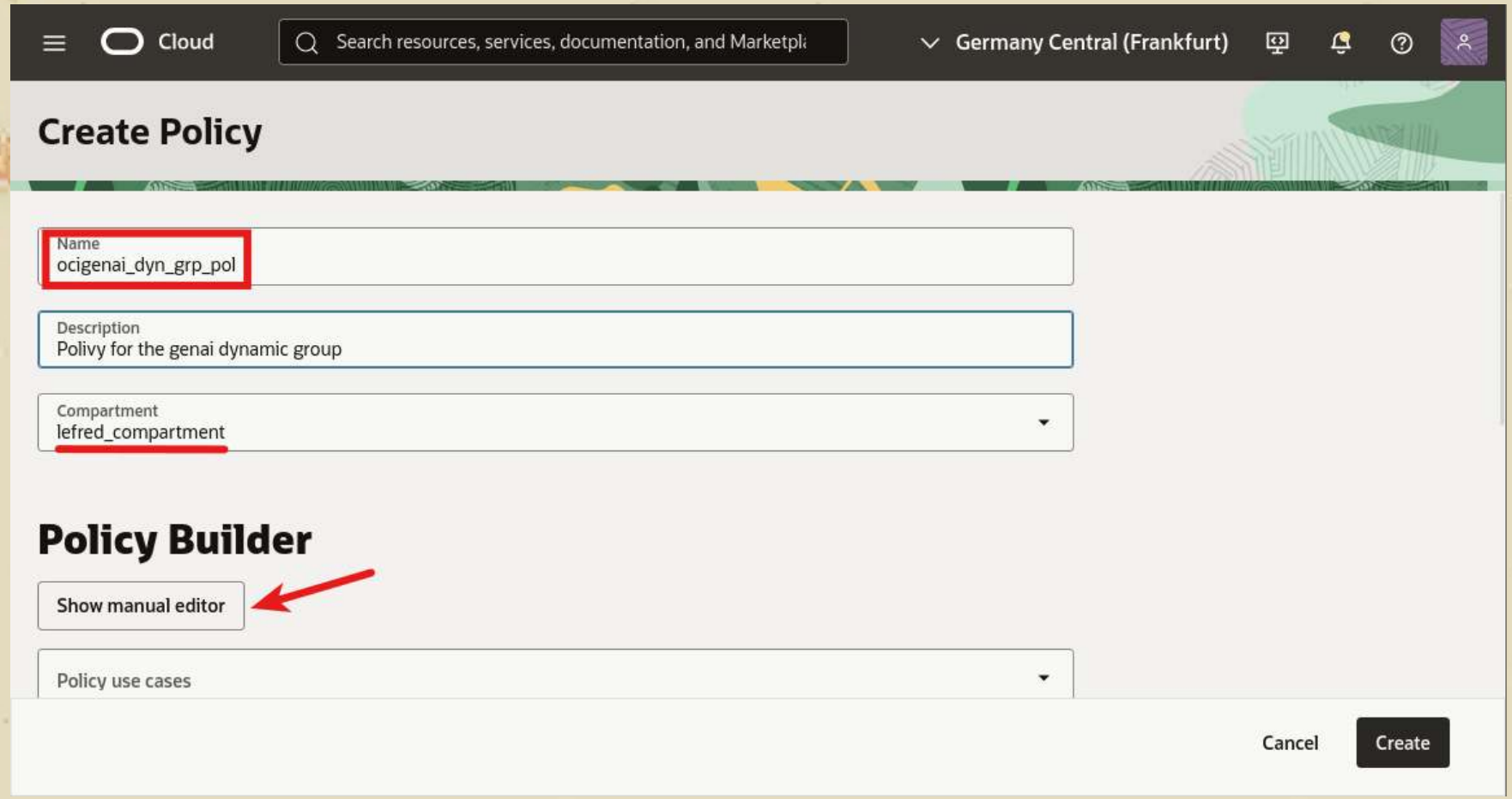


Authenticating MySQL HeatWave to OCI GenAI

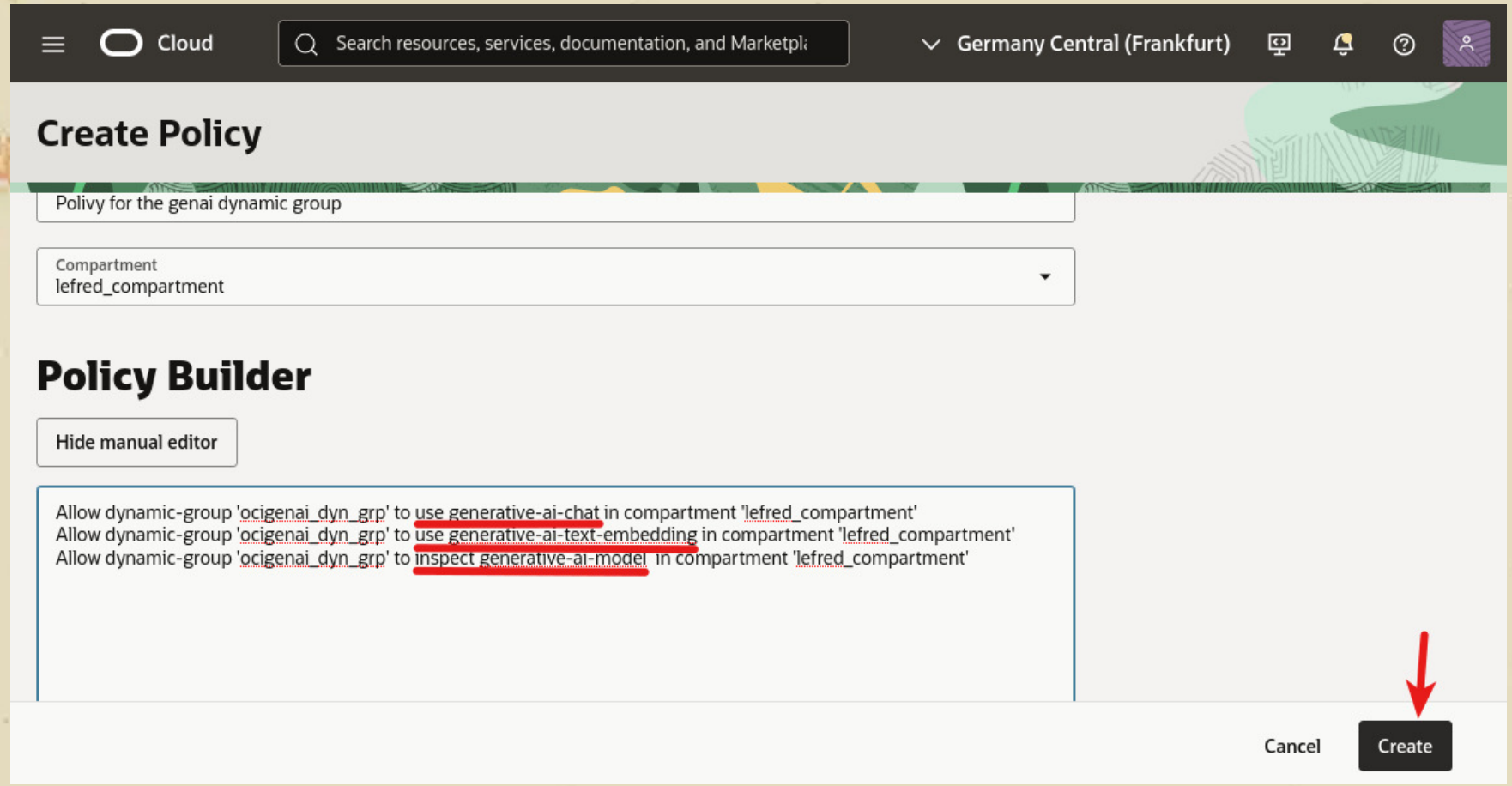
Step 3 - Policy



Authenticating MySQL HeatWave to OCI GenAI



Authenticating MySQL HeatWave to OCI GenAI



Authenticating MySQL HeatWave to OCI GenAI

Summary

DB System in a compartment:

Dynamic Group: `name_dyn_grp`

```
ALL{resource.type = 'mysqldbsystem', resource.compartment.id = 'ocid1.tenancy....'}
```

Policy:

```
Allow dynamic-group 'name_dyn_grp' to use generative-ai-chat in compartment 'compartment_name'
Allow dynamic-group 'name_dyn_grp' to use generative-ai-text-embedding in compartment 'compartment_name'
Allow dynamic-group 'name_dyn_grp' to inspect generative-ai-model in compartment 'compartment_name'
```

Authenticating MySQL HeatWave to OCI GenAI

Summary

DB System in tenancy's root:

Dynamic Group: `name_dyn_grp`

```
ALL{resource.type = 'mysqldbsystem'}
```

Policy:

```
Allow dynamic-group 'name_dyn_grp' to use generative-ai-chat in tenancy
Allow dynamic-group 'name_dyn_grp' to use generative-ai-text-embedding in tenancy
Allow dynamic-group 'name_dyn_grp' to inspect generative-ai-model in tenancy
```

Using OCI GenAI Models in MySQL HeatWave

And now we can access those models from MySQL HeatWave using SQL:

```
SQL> SELECT provider, model_id, availability_date, capabilities FROM sys.ML_SUPPORTED_LLMS;
+--------------------------+-----------------------------------------------+-------------------+----------------------+
| provider                 | model_id                                      | availability_date | capabilities         |
+--------------------------+-----------------------------------------------+-------------------+----------------------+
| HeatWave                 | llama3.2-1b-instruct-v1                       | 2025-05-20        | ["GENERATION"]       |
| HeatWave                 | llama3.2-3b-instruct-v1                       | 2025-05-20        | ["GENERATION"]       |
| HeatWave                 | all_minilm_l12_v2                             | 2024-07-01        | ["TEXT_EMBEDDINGS"]  |
| HeatWave                 | multilingual-e5-small                         | 2024-07-24        | ["TEXT_EMBEDDINGS"]  |
| OCI Generative AI Service | meta.llama-4-maverick-17b-128e-instruct-fp8 | 2025-11-12        | ["GENERATION"]       |
| OCI Generative AI Service | meta.llama-4-scout-17b-16e-instruct          | 2025-11-12        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-4-fast-non-reasoning                | 2025-09-25        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-4-fast-reasoning                    | 2025-09-25        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-code-fast-1                         | 2025-09-11        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-3-mini-fast                         | 2025-07-17        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-3-fast                              | 2025-07-17        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-3                                   | 2025-07-17        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-3-mini                              | 2025-07-17        | ["GENERATION"]       |
| OCI Generative AI Service | xai.grok-4                                   | 2025-07-17        | ["GENERATION"]       |
+--------------------------+-----------------------------------------------+-------------------+----------------------+
14 rows in set (1.7130 sec)
```

Benefits & Use Cases

The use cases are the same as for in-database models, but with better performance and quality.

The OCI GenAI models are more powerful and accurate, they run on GPUs and they were built with larger datasets and parameters.

Benefits & Use Cases

The use cases are the same as for in-database models, but with better performance and quality.

The OCI GenAI models are more powerful and accurate, they run on GPUs and they were built with larger datasets and parameters.

I would recommend:

- *using in-database models for simple tasks and prototyping*
- *using OCI GenAI models for more complex tasks and production workloads*

Comparative Performance

Shape:

- *MySQL.16 (16 ECPUs, 128 GB RAM)*
  - *HeatWave.32GB (32 GB Memory, 1 node)*

sys.HEATWAVE_CHAT()

Model	*Occurence*	*Time*	*Accuracy*
llama3.2-3b-instruct-v1	*1*	*2 min 36.6196 sec*	🔴
llama3.2-3b-instruct-v1	*2*	*51.2197 sec*	🟠
xai.grok-3-fast	*1*	*23.4348 sec*	🟢
xai.grok-3-fast	*2*	*21.9907 sec*	🟢
xai.grok-4-fast-reasoning	*1*	*20.4239 sec*	🟢

Comparative Performance

sys.ML_GENERATE_TABLE() with 10 rows

```
MySQL > select * from input_table;
+----+-------------------------------------------------------------------------------+
| id | input                                                                         |
+----+-------------------------------------------------------------------------------+
|  1 | Describe what is MySQL in 50 words.                                           |
|  2 | Describe Artificial Intelligence in 50 words.                                 |
|  3 | Describe MySQL HeatWave in 50 words.                                          |
|  4 | Describe why MySQL is the most popular open-source  database in 50 words.     |
|  5 | Describe why people should attend the preFOSDEM MySQL Belgian Days in 100 words. |
|  6 | Why should people use InnoDB in 10 points                                     |
|  7 | Provide me 10 activities to do in Brussels over the week-end                  |
|  8 | Which Belgian beers should I try and why?                                     |
|  9 | Describe why GenAI will not replace DBAs in 50 words                          |
| 10 | Who will win the FIFA World Cup in 2026 and who will score the most goals?    |
+----+-------------------------------------------------------------------------------+
```

Model	Time	Accuracy
llama3.2-1b-instruct-v1	1 min 59.5193 sec	🟠
llama3.2-3b-instruct-v1	4 min 20.0489 sec	🟠
xai.grok-4-fast-non-reasoning	29.6241 sec	🟢
xai.grok-4-fast-reasoning	50.8834 sec	🟢
xai.grok-code-fast-1	46.6858 sec	🟠🟠
xai.grok-3-mini-fast	1 min 53.7430 sec	🟠🟠
xai.grok-3-fast	1 min 8.8987 sec	🟠🟠
xai.grok-3	1 min 6.8184 sec	🟠🟠
xai.grok-3-mini	1 min 49.3835 sec	🟠🟠
xai.grok-4	1 min 21.0076 sec	🟢

Comparative Performance - NL2SQL

"For each categories of movies, list the actor who played in most movies of that category, display only one actor per category and sort by the highest participation first"

The best result with **`xai.grok-code-fast-1`** *in 6.9354 sec:*

```
SELECT `category`, `first_name`, `last_name`, `num_films`
FROM ( SELECT `c`.`name` AS `category`, `a`.`first_name`, `a`.`last_name`, COUNT(*) AS `num_films`,
  ROW_NUMBER() OVER (PARTITION BY `c`.`category_id` ORDER BY COUNT(*) DESC) AS `rn`
          FROM `sakila`.`category` AS `c` JOIN `sakila`.`film_category` AS `fc`
            ON `c`.`category_id` = `fc`.`category_id`
          JOIN `sakila`.`film_actor` AS `fa` ON `fc`.`film_id` = `fa`.`film_id`
          JOIN `sakila`.`actor` AS `a` ON `fa`.`actor_id` = `a`.`actor_id`
          GROUP BY `c`.`category_id`, `c`.`name`, `a`.`actor_id`,
                   `a`.`first_name`, `a`.`last_name`) AS `sub`
    WHERE `rn` = 1 ORDER BY `num_films` DESC
1 row in set (6.9354 sec)
```

Comparative Performance - NL2SQL (2)

Here, the results of OCI GenAI models is much better than those of in-database models:

Model	*Time*	*Accuracy*
llama3.2-3b-instruct-v1	*43.3887 sec*	🔴
xai.grok-code-fast-1	*6.9354 sec*	🟢

Comparative Performance - NL2SQL (3)

llama3.2-3b-instruct-v1

```
+----------+------------+
| actor_id | num_movies |
+----------+------------+
|      107 |         42 |
+----------+------------+
1 row in set (43.3887 sec)
```

xai.grok-code-fast-1

```
+-------------+-----------------+-----------+
| category    | actor           | num_films |
+-------------+-----------------+-----------+
| Sports      | BEN WILLIS      |         9 |
| Children    | HELEN VOIGHT    |         7 |
| Drama       | GRACE MOSTEL    |         7 |
| Foreign     | HUMPHREY WILLIS |         7 |
| Horror      | JULIA MCQUEEN   |         7 |
| New         | SIDNEY CROWE    |         7 |
| Sci-Fi      | GINA DEGENERES  |         7 |
| Action      | NATALIE HOPKINS |         6 |
| Animation   | MORGAN WILLIAMS |         6 |
| Classics    | GREG CHAPLIN    |         6 |
| Comedy      | BELA WALKEN     |         6 |
| Documentary | ED CHASE        |         6 |
| Family      | MAE HOFFMAN     |         6 |
| Games       | RIP WINSLET     |         5 |
| Music       | WARREN NOLTE    |         5 |
| Travel      | NICK STALLONE   |         5 |
+-------------+-----------------+-----------+
16 rows in set (6.9354 sec)
```

Comparative Performance - VLMs

Some of the models, unfortunately, not on HeatWave yet, support Vision Language tasks.

Of course it depends on the image and the model capabilities, but here are some results with a sample image.

VLMs - examples

medium photo 640x360 pixels 118.4 KB

`How many people are in the room?`

xai.grok-4: 10 (36.4546 sec)

xai.grok-4-fast-reasoning: 7 (7.2420 sec)

xai.grok-4-fast-non-reasoning: There are 9 people visible in the room in this photo. (1.4770 sec)

VLMs - examples (2)

large photo 4032x2268 pixels 2.2 MB

`How many people are in the room?`

xai.grok-4: There are 10 people visible in the room based on the photo. (24.0342 sec)

xai.grok-4-fast-reasoning: Based on the image, there are 8 people visibly present in the room. (11.4148 sec)

xai.grok-4-fast-non-reasoning: Based on the photo, there are 9 people visible in the room. (3.3931 sec)

VLMs - examples (3)

```
MySQL > select sys.ml_generate("Describe the image",
        JSON_OBJECT("model_id", "xai.grok-4-fast-reasoning", "image", @image))\G
 *************************** 1. row ***************************
sys.ml_generate("Describe the image", JSON_OBJECT("model_id", "xai.grok-4-fast-reasoning",
"image", @image)): {"text": "The image depicts an indoor event, likely a book signing
or promotional meet-and-greet, in a spacious room with white walls and tiled flooring.
At the center is a rectangular table covered in a bright blue cloth, cluttered with
stacks of white boxes (possibly containing books or merchandise), printed papers,
a green folder, and a small black key fob. A man with short dark hair, glasses, and a
trimmed beard, dressed in a dark blue jacket over a gray shirt, is seated at the table,
leaning forward attentively as he signs a document or book with a pen in his right hand.
Standing close to the table is a young woman with long blonde hair tied in a ponytail,
wearing a black long-sleeve top and black leggings, smiling warmly as she interacts with
the signer. She appears engaged and happy, with her hands clasped in front of her.
Surrounding them are several other attendees, mostly men in casual attire: one with light
brown hair and glasses in a light blue t-shirt stands nearby holding papers; another with
short blond hair in a gray hoodie leans in from the side; and a couple more in hoodies
and jeans observe from a short distance.\n\nIn the background, the room features rows of
empty white plastic chairs arranged in a semi-circle, a wooden podium on a small stage,
and a few more people milling about, including one man in a blue shirt standing with arms
crossed. The atmosphere seems relaxed and informal, with natural light filtering in from
off-frame windows or doors."}
1 row in set (6.9314 sec)
```

Bonus

They are not listed when querying `sys.ML_SUPPORTED_LLMS`*, but you can also use other older models hosted in MySQL HeatWave for generation tasks:*

- `mistral-7b-instruct-v1`
- `llama2-7b-v1`
- `llama3-8b-instruct-v1`

MySQL HeatWave AI on Steroïd

Embeddings

Embeddings

However, for embeddings, the in-database models are still the only option for now:

- `all_minilm_l12_v2`
- `multilingual-e5-small`

Nothing else is available from OCI Generative AI Service yet.

```
MySQL > set @text='preFOSDEM MySQL Belgian Days 2026';
MySQL > SELECT sys.ML_EMBED_ROW(@text,
  JSON_OBJECT("model_id", "all_minilm_l12_v2")) into @text_embedding;
Query OK, 1 row affected (0.4325 sec)
MySQL > SELECT sys.ML_EMBED_ROW(@text,
  JSON_OBJECT("model_id", "multilingual-e5-small")) into @text_embedding;
Query OK, 1 row affected (0.8245 sec)
```

Questions ?

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

A: No, currently MySQL HeatWave only supports the built-in models provided by HeatWave

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

A: No, currently MySQL HeatWave only supports the built-in models provided by HeatWave

Q: Can I load custom or other models into OCI Generative AI Service?

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

A: No, currently MySQL HeatWave only supports the built-in models provided by HeatWave

Q: Can I load custom or other models into OCI Generative AI Service?

A: Yes, OCI Generative AI Service supports custom or other model deployment (even from hugingface)

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

A: No, currently MySQL HeatWave only supports the built-in models provided by HeatWave

Q: Can I load custom or other models into OCI Generative AI Service?

A: Yes, OCI Generative AI Service supports custom or other model deployment (even from hugingface)

Q: Can I use them from MySQL HeatWave?

Questions ?

Q: Can I load custom or other models into MySQL HeatWave?

A: No, currently MySQL HeatWave only supports the built-in models provided by HeatWave

Q: Can I load custom or other models into OCI Generative AI Service?

A: Yes, OCI Generative AI Service supports custom or other model deployment (even from hugingface)

Q: Can I use them from MySQL HeatWave?

A: No :'-(