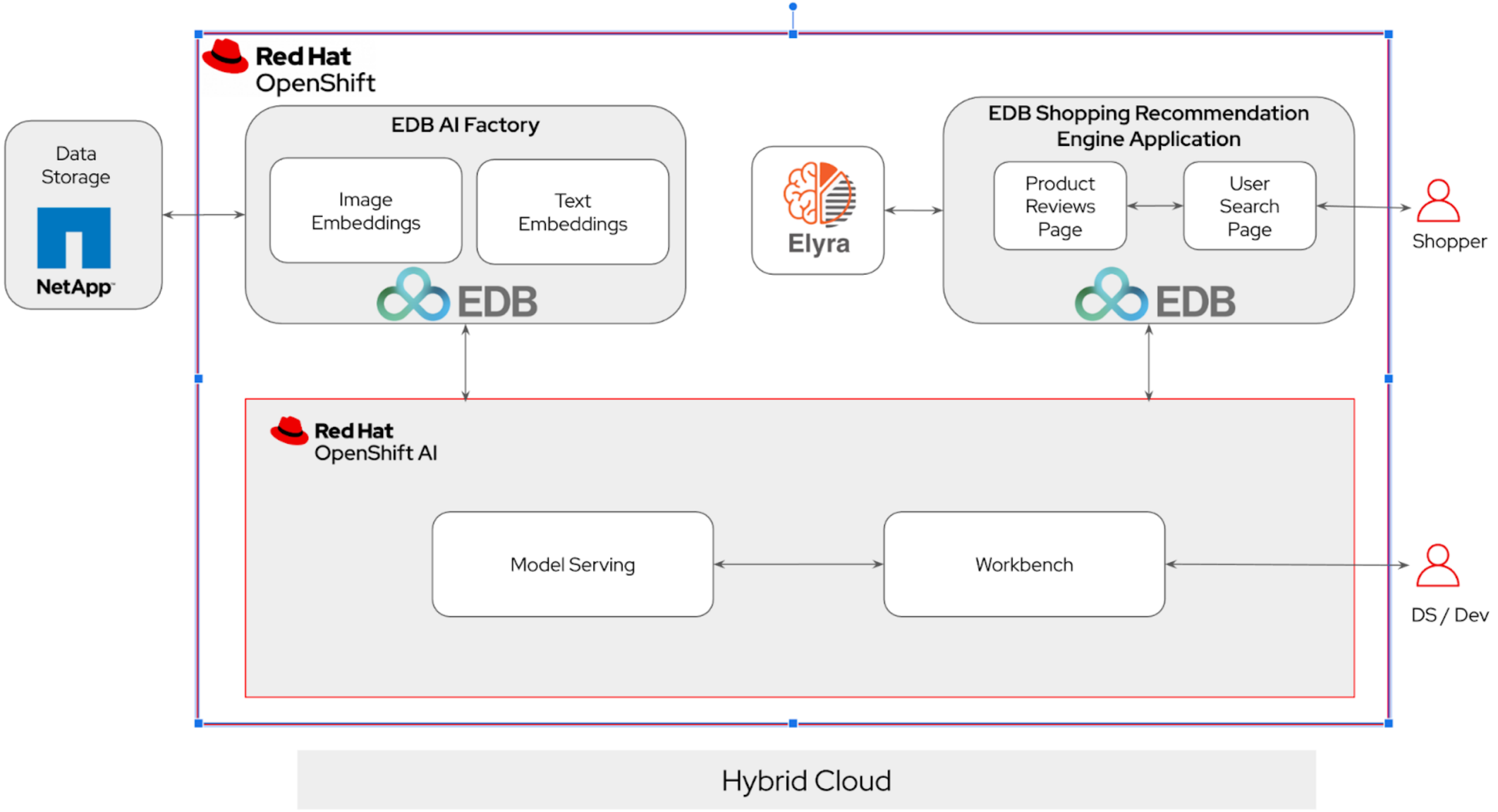
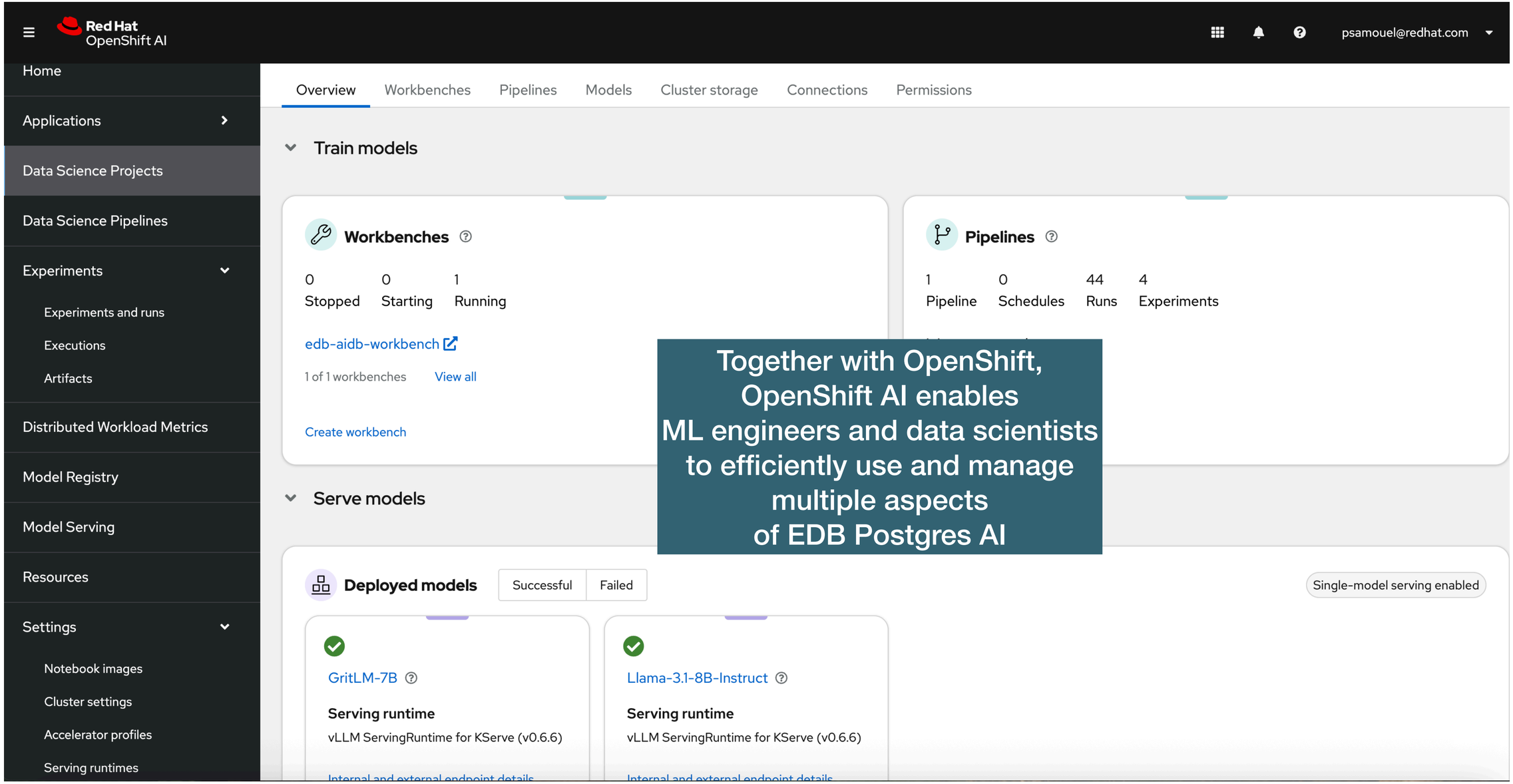
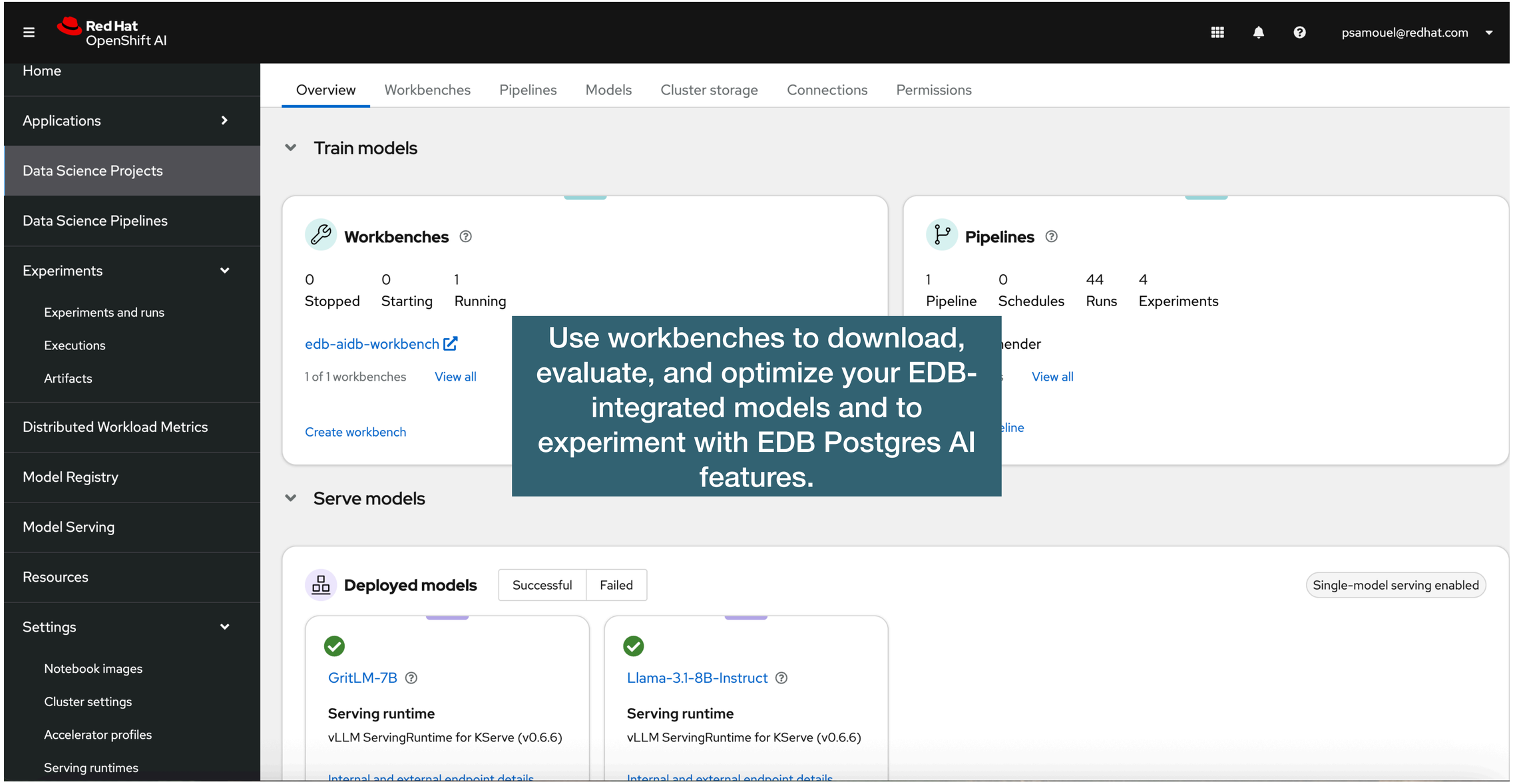

# Red Hat OpenShift AI

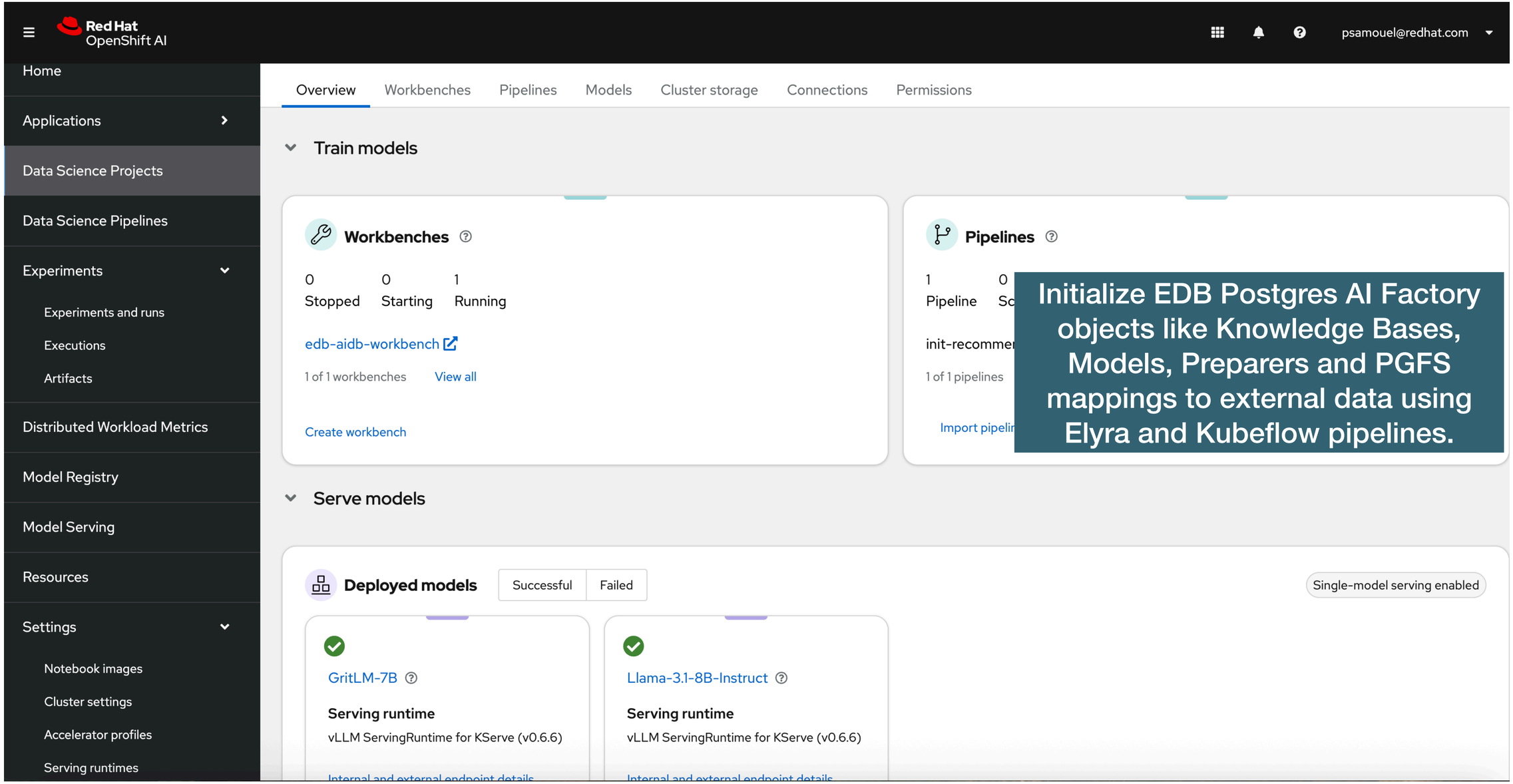
EDB Postgres AI

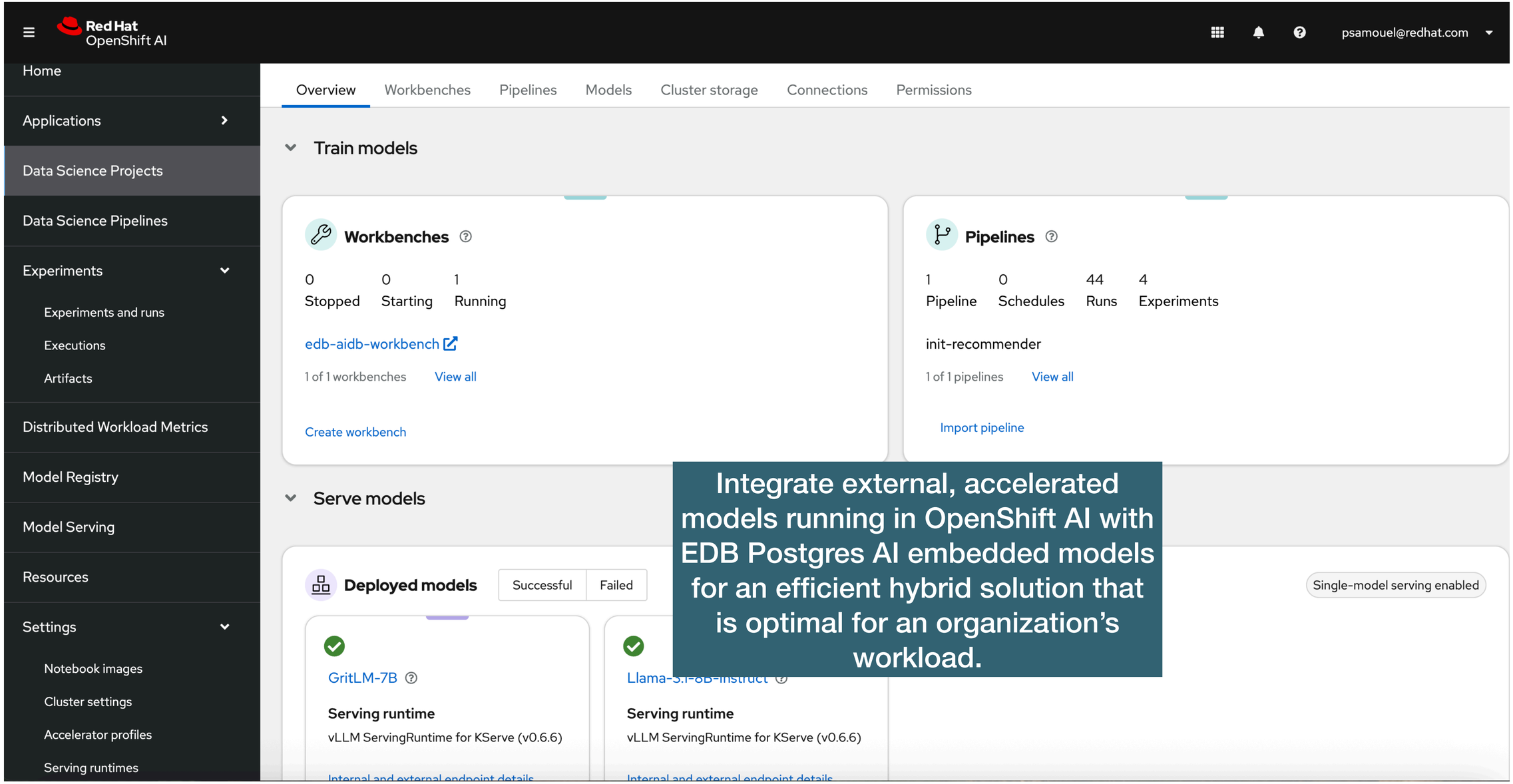
## OpenShift AI - A Comprehensive Platform for EDB Postgres AI

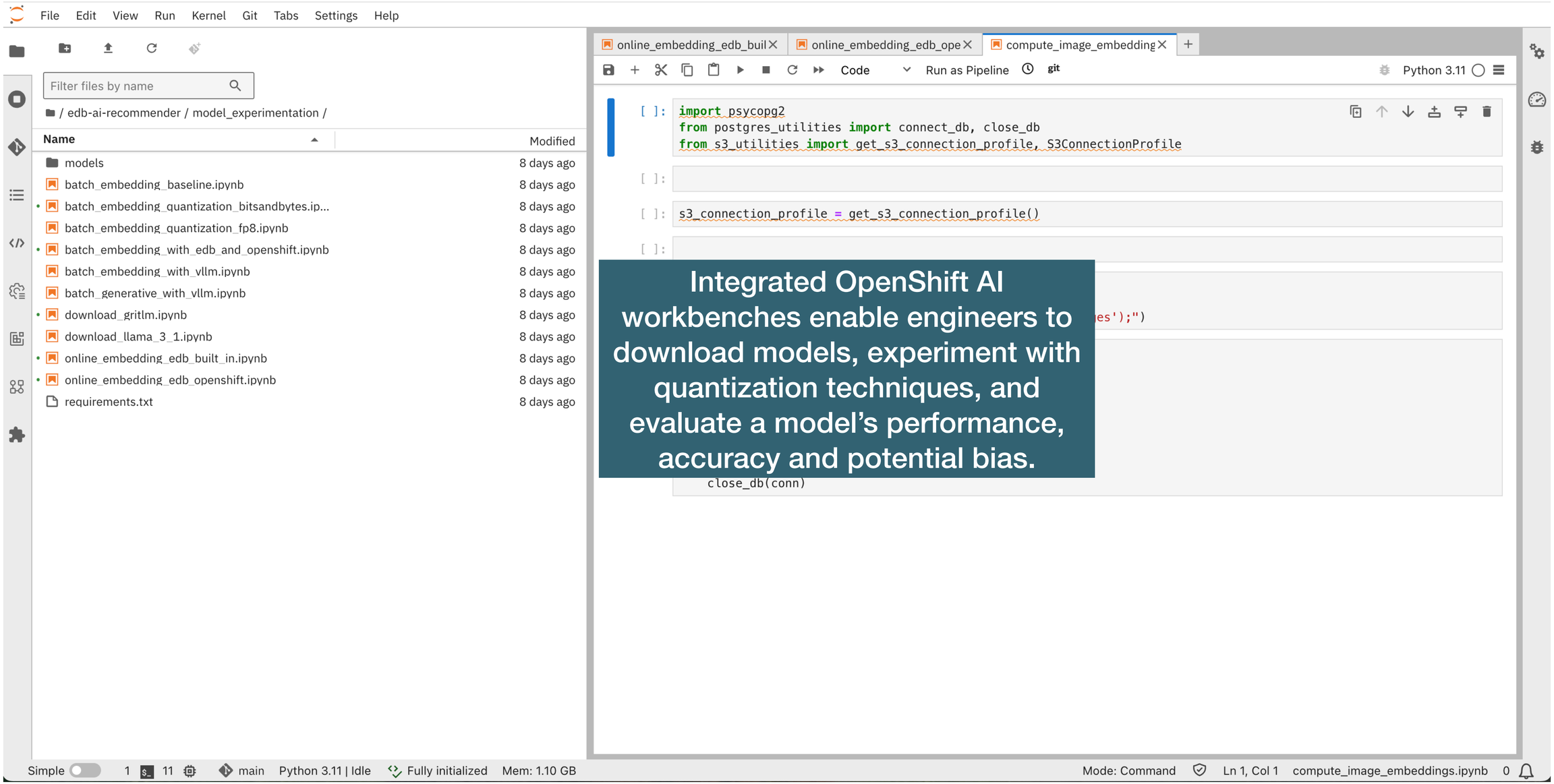

Red Hat
OpenShift
Data
Storage
NetApp
EDB AI Factory
Image
Embeddings
Text
Embeddings
EDB
Elyra
EDB Shopping Recommendation
Engine Application
Product
Reviews
Page
User
Search
Page
EDB
Shopper
Red Hat
OpenShift AI
Model Serving
Workbench
DS / Dev
Hybrid Cloud

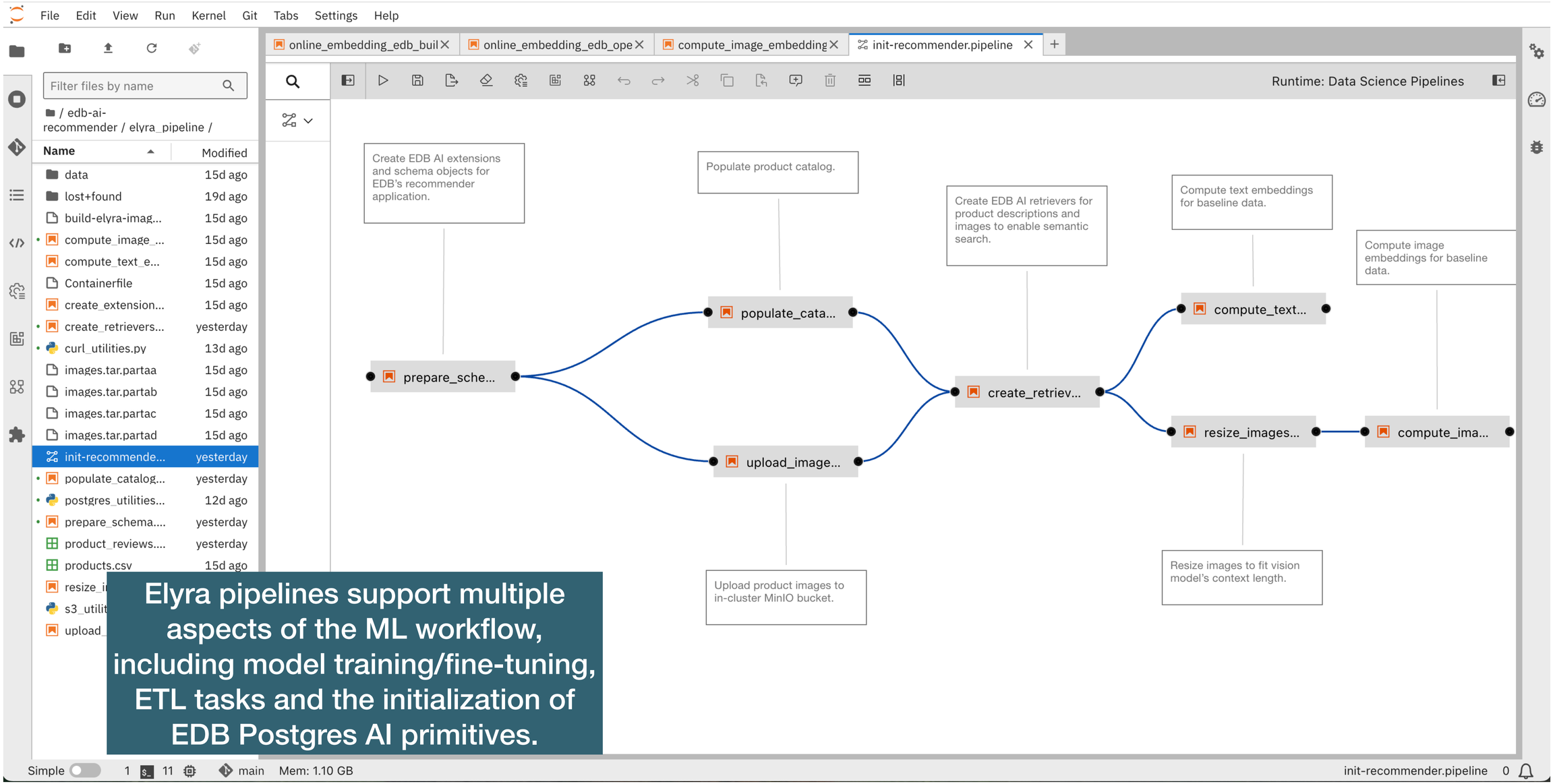

Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Notebook images
Cluster settings
Accelerator profiles
Serving runtimes
Overview
Workbenches
Pipelines
Models
Cluster storage
Connections
Permissions
Train models
Workbenches
0 Stopped
0 Starting
1 Running
edb-aidb-workbench
1 of 1 workbenches
View all
Create workbench
Pipelines
1 Pipeline
0 Schedules
44 Runs
4 Experiments
Together with OpenShift,
OpenShift AI enables
ML engineers and data scientists
to efficiently use and manage
multiple aspects
of EDB Postgres AI
Serve models
Deployed models
Successful
Failed
Single-model serving enabled
GritLM-7B
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)
Llama-3.1-8B-Instruct
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)

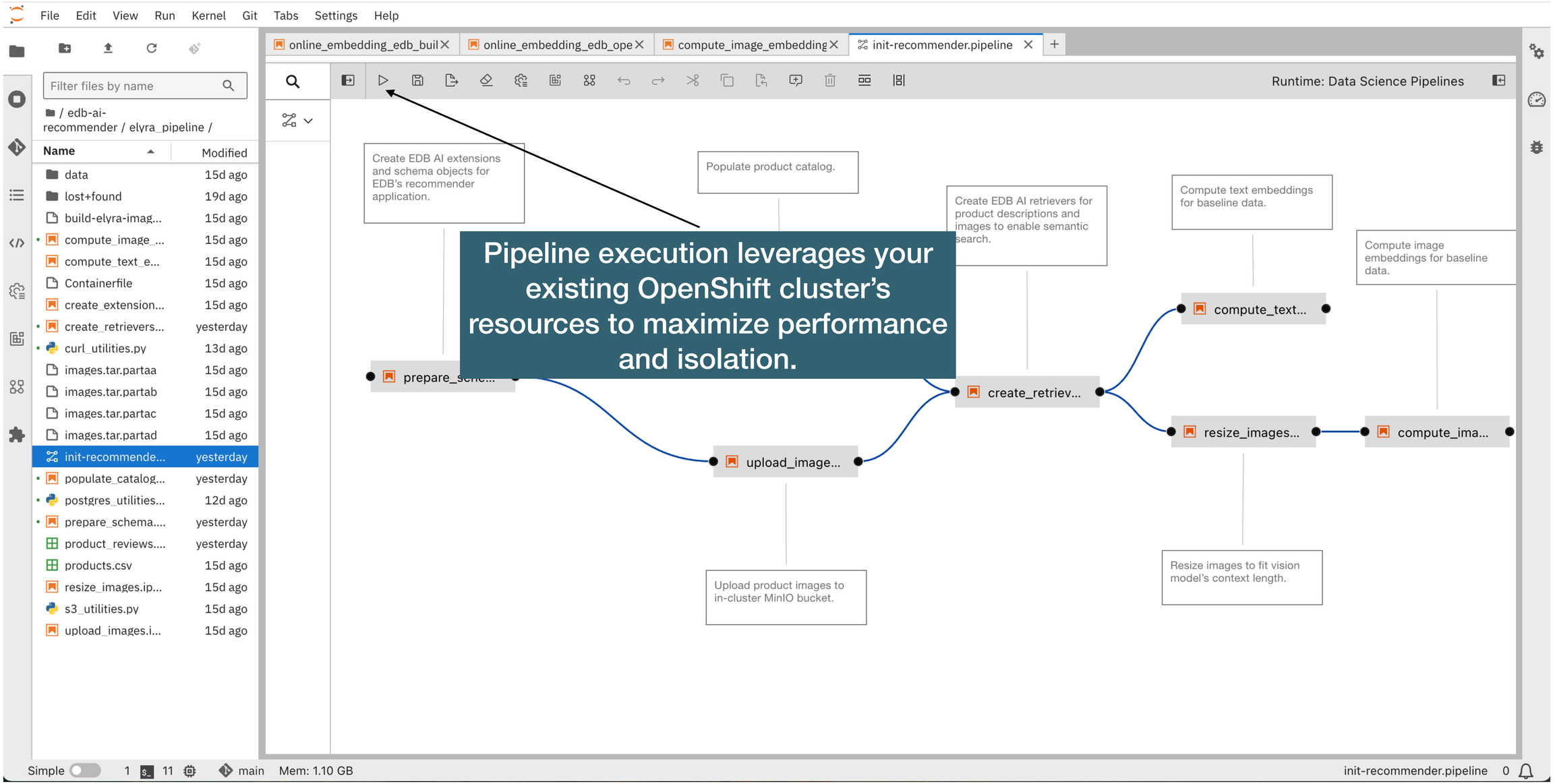

Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Notebook images
Cluster settings
Accelerator profiles
Serving runtimes
Overview
Workbenches
Pipelines
Models
Cluster storage
Connections
Permissions
Train models
Workbenches
0 Stopped
0 Starting
1 Running
edb-aidb-workbench
1 of 1 workbenches
View all
Create workbench
Pipelines
1 Pipeline
0 Schedules
44 Runs
4 Experiments
View all
Use workbenches to download, evaluate, and optimize your EDB-integrated models and to experiment with EDB Postgres AI features.
Serve models
Deployed models
Successful
Failed
Single-model serving enabled
GritLM-7B
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)
Llama-3.1-8B-Instruct
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)

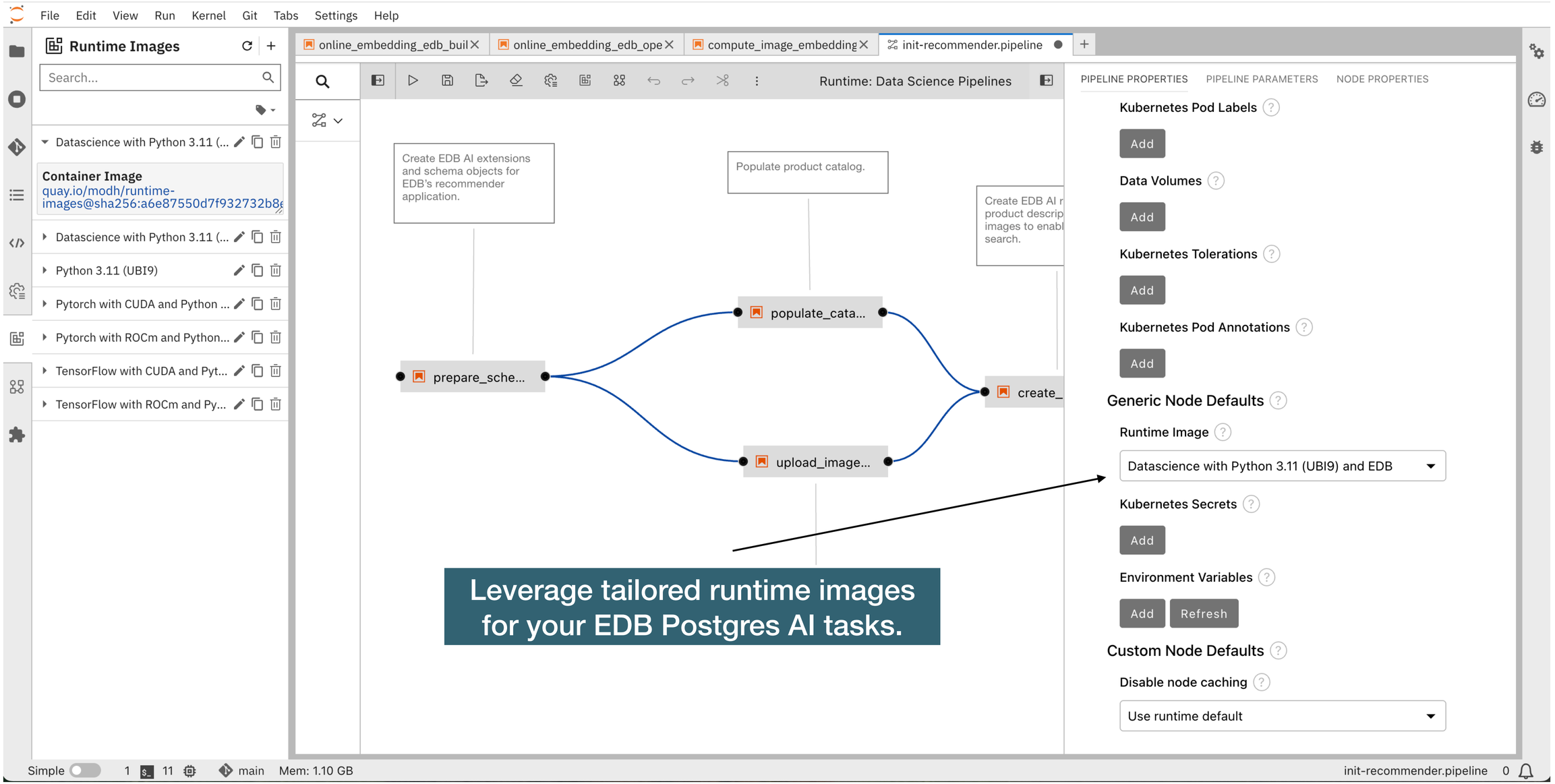

Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Notebook images
Cluster settings
Accelerator profiles
Serving runtimes
Overview
Workbenches
Pipelines
Models
Cluster storage
Connections
Permissions
Train models
Workbenches
0 Stopped
0 Starting
1 Running
edb-aidb-workbench
1 of 1 workbenches
View all
Create workbench
Pipelines
1 Pipeline
0
init-recommer
1 of 1 pipelines
Import pipelir
Initialize EDB Postgres AI Factory objects like Knowledge Bases, Models, Preparers and PGFS mappings to external data using Elyra and Kubeflow pipelines.
Serve models
Deployed models
Successful
Failed
Single-model serving enabled
GritLM-7B
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)
Llama-3.1-8B-Instruct
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)

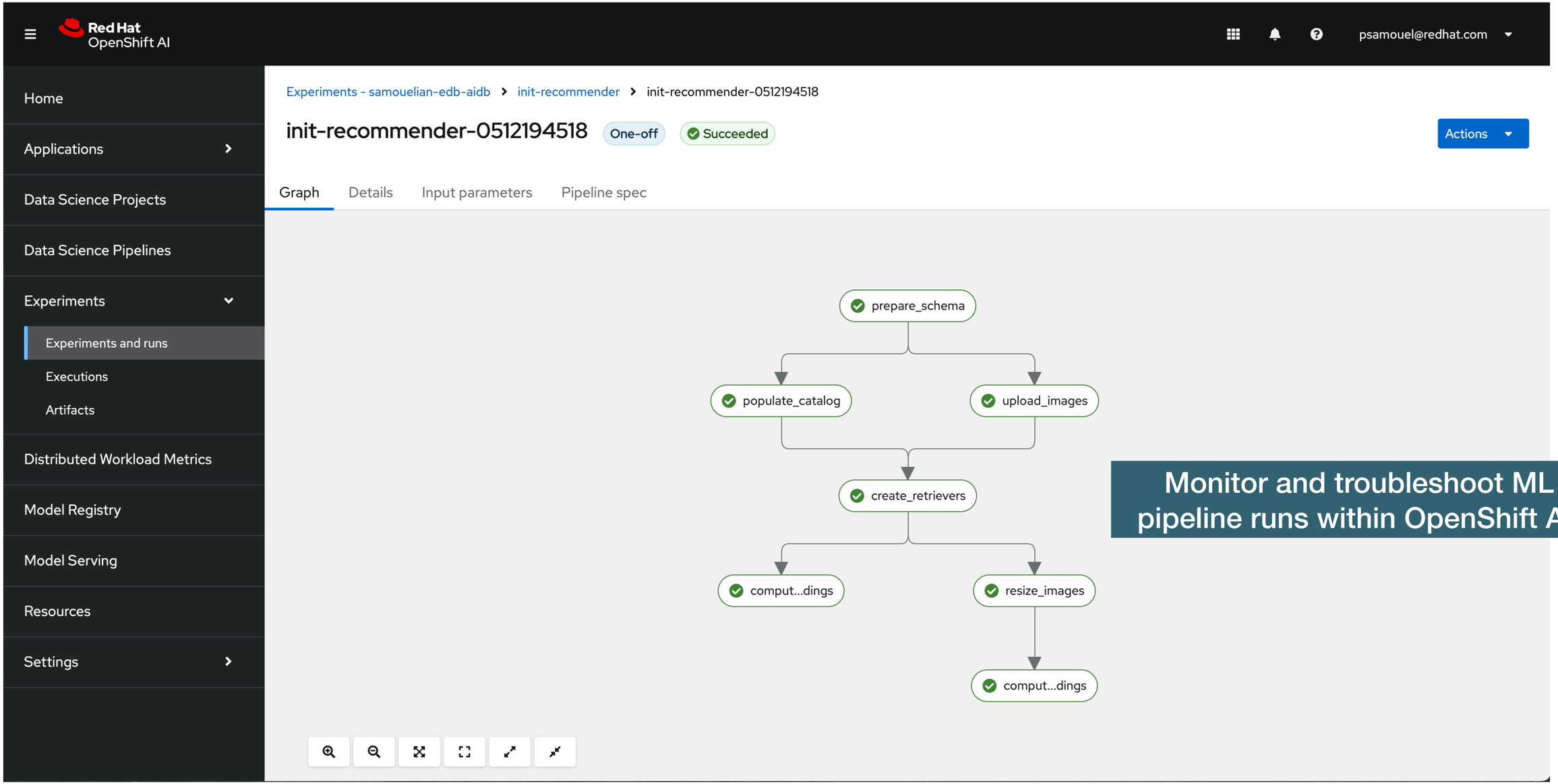

Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Notebook images
Cluster settings
Accelerator profiles
Serving runtimes
Overview
Workbenches
Pipelines
Models
Cluster storage
Connections
Permissions
Train models
Workbenches
0 Stopped
0 Starting
1 Running
edb-aidb-workbench
1 of 1 workbenches
View all
Create workbench
Pipelines
1 Pipeline
0 Schedules
44 Runs
4 Experiments
init-recommender
1 of 1 pipelines
View all
Import pipeline
Serve models
Deployed models
Successful
Failed
Single-model serving enabled
GritLM-7B
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)
Llama-3.1-8B-Instruct
Serving runtime
vLLM ServingRuntime for KServe (v0.6.6)
Integrate external, accelerated models running in OpenShift AI with EDB Postgres AI embedded models for an efficient hybrid solution that is optimal for an organization's workload.


File Edit View Run Kernel Git Tabs Settings Help
Filter files by name
/ edb-ai-recommender / model_experimentation /
Name
Modified
models 8 days ago
batch_embedding_baseline.ipynb 8 days ago
batch_embedding_quantization_bitsandbytes.ip... 8 days ago
batch_embedding_quantization_fp8.ipynb 8 days ago
batch_embedding_with_edb_and_openshift.ipynb 8 days ago
batch_embedding_with_vllm.ipynb 8 days ago
batch_generative_with_vllm.ipynb 8 days ago
download_gritlm.ipynb 8 days ago
download_llama_3_1.ipynb 8 days ago
online_embedding_edb_built_in.ipynb 8 days ago
online_embedding_edb_openshift.ipynb 8 days ago
requirements.txt 8 days ago
online_embedding_edb_buil
online_embedding_edb_ope
compute_image_embedding
Code
Run as Pipeline
git
Python 3.11
[ ]:
import psycopg2
from postgres_utilities import connect_db, close_db
from s3_utilities import get_s3_connection_profile, S3ConnectionProfile
[ ]:
[ ]:
s3_connection_profile = get_s3_connection_profile()
[ ]:
close_db(conn)
Integrated OpenShift AI workbenches enable engineers to download models, experiment with quantization techniques, and evaluate a model's performance, accuracy and potential bias.
Simple
1
11
main
Python 3.11 | Idle
Fully initialized
Mem: 1.10 GB
Mode: Command
Ln 1, Col 1
compute_image_embeddings.ipynb
0


File Edit View Run Kernel Git Tabs Settings Help
online_embedding_edb_buil
online_embedding_edb_ope
compute_image_embedding
init-recommender.pipeline
Runtime: Data Science Pipelines
Filter files by name
/ edb-ai-recommender / elyra_pipeline /
Name
Modified
data 15d ago
lost+found 19d ago
build-elyra-imag... 15d ago
compute_image_... 15d ago
compute_text_e... 15d ago
Containerfile 15d ago
create_extension... 15d ago
create_retrievers... yesterday
curl_utilities.py 13d ago
images.tar.partaa 15d ago
images.tar.partab 15d ago
images.tar.partac 15d ago
images.tar.partad 15d ago
init-recommende... yesterday
populate_catalog... yesterday
postgres_utilities... 12d ago
prepare_schema.... yesterday
product_reviews.... yesterday
products.csv 15d ago
Create EDB AI extensions and schema objects for EDB's recommender application.
Populate product catalog.
Create EDB AI retrievers for product descriptions and images to enable semantic search.
Compute text embeddings for baseline data.
Compute image embeddings for baseline data.
prepare_sche...
populate_cata...
upload_image...
create_retriev...
compute_text...
resize_images...
compute_ima...
Upload product images to in-cluster MinIO bucket.
Resize images to fit vision model's context length.
Elyra pipelines support multiple aspects of the ML workflow, including model training/fine-tuning, ETL tasks and the initialization of EDB Postgres AI primitives.
Simple
main
Mem: 1.10 GB
init-recommender.pipeline


File Edit View Run Kernel Git Tabs Settings Help
online_embedding_edb_buil
online_embedding_edb_ope
compute_image_embedding
init-recommender.pipeline
Runtime: Data Science Pipelines
Filter files by name
/ edb-ai-recommender / elyra_pipeline /
Name
Modified
data 15d ago
lost+found 19d ago
build-elyra-imag... 15d ago
compute_image_... 15d ago
compute_text_e... 15d ago
Containerfile 15d ago
create_extension... 15d ago
create_retrievers... yesterday
curl_utilities.py 13d ago
images.tar.partaa 15d ago
images.tar.partab 15d ago
images.tar.partac 15d ago
images.tar.partad 15d ago
init-recommende... yesterday
populate_catalog... yesterday
postgres_utilities... 12d ago
prepare_schema.... yesterday
product_reviews.... yesterday
products.csv 15d ago
resize_images.ip... 15d ago
s3_utilities.py 15d ago
upload_images.i... 15d ago
Create EDB AI extensions and schema objects for EDB's recommender application.
Populate product catalog.
Create EDB AI retrievers for product descriptions and images to enable semantic search.
Compute text embeddings for baseline data.
Compute image embeddings for baseline data.
Pipeline execution leverages your existing OpenShift cluster's resources to maximize performance and isolation.
prepare_sche...
upload_image...
create_retriev...
compute_text...
resize_images...
compute_ima...
Upload product images to in-cluster MinIO bucket.
Resize images to fit vision model's context length.
Simple
1
11
main
Mem: 1.10 GB
init-recommender.pipeline
0


File Edit View Run Kernel Git Tabs Settings Help
Runtime Images
Search...
Datascience with Python 3.11 (...
Container Image
quay.io/modh/runtime-images@sha256:a6e87550d7f932732b8
Datascience with Python 3.11 (...
Python 3.11 (UBI9)
Pytorch with CUDA and Python ...
Pytorch with ROCm and Python...
TensorFlow with CUDA and Pyt...
TensorFlow with ROCm and Py...
online_embedding_edb_buil
online_embedding_edb_ope
compute_image_embedding
init-recommender.pipeline
Runtime: Data Science Pipelines
Create EDB AI extensions and schema objects for EDB's recommender application.
Populate product catalog.
Create EDB AI r product descrip images to enabl search.
populate_cata...
prepare_sche...
create_
upload_image...
Leverage tailored runtime images for your EDB Postgres AI tasks.
PIPELINE PROPERTIES
PIPELINE PARAMETERS
NODE PROPERTIES
Kubernetes Pod Labels
Add
Data Volumes
Add
Kubernetes Tolerations
Add
Kubernetes Pod Annotations
Add
Generic Node Defaults
Runtime Image
Datascience with Python 3.11 (UBI9) and EDB
Kubernetes Secrets
Add
Environment Variables
Add
Refresh
Custom Node Defaults
Disable node caching
Use runtime default
Simple
1
11
main
Mem: 1.10 GB
init-recommender.pipeline
0


Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Experiments - samouelian-edb-aidb
init-recommender
init-recommender-0512194518
init-recommender-0512194518
One-off
Succeeded
Actions
Graph
Details
Input parameters
Pipeline spec
prepare_schema
populate_catalog
upload_images
create_retrievers
comput...dings
resize_images
comput...dings
Monitor and troubleshoot ML pipeline runs within OpenShift AI.

Red Hat OpenShift AI

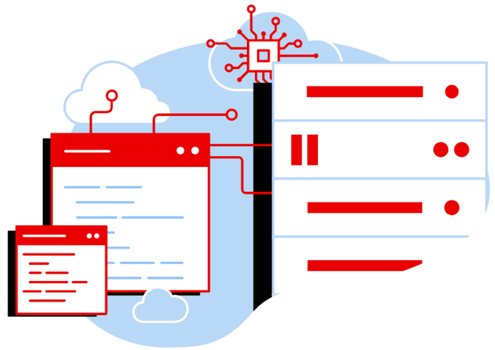
psamouel@redhat.com

- Home
- Applications
- Data Science Projects
- Data Science Pipelines
- Experiments
  - Experiments and runs
  - Executions
  - Artifacts
- Distributed Workload Metrics
- Model Registry
- Model Serving
- Resources
- Settings

# Model Registry

Select a model registry to view and manage your registered models. Model registries provide a structured and organized way to st

Model registry: model-registry-server01



## No models in selected registry

model-registry-server01 has no active registered models. Register a model in this registry, or select a different registry.

Register model

View archived models

**Manage model storage, versioning and provenance using OpenShift AI's model registry.**

**Model description**

## Version details

Configure details for the first version of this model.

**Version name** *

**Version description**

**Source model format**

Example, tensorflow

**Source model format version**

Example, 1

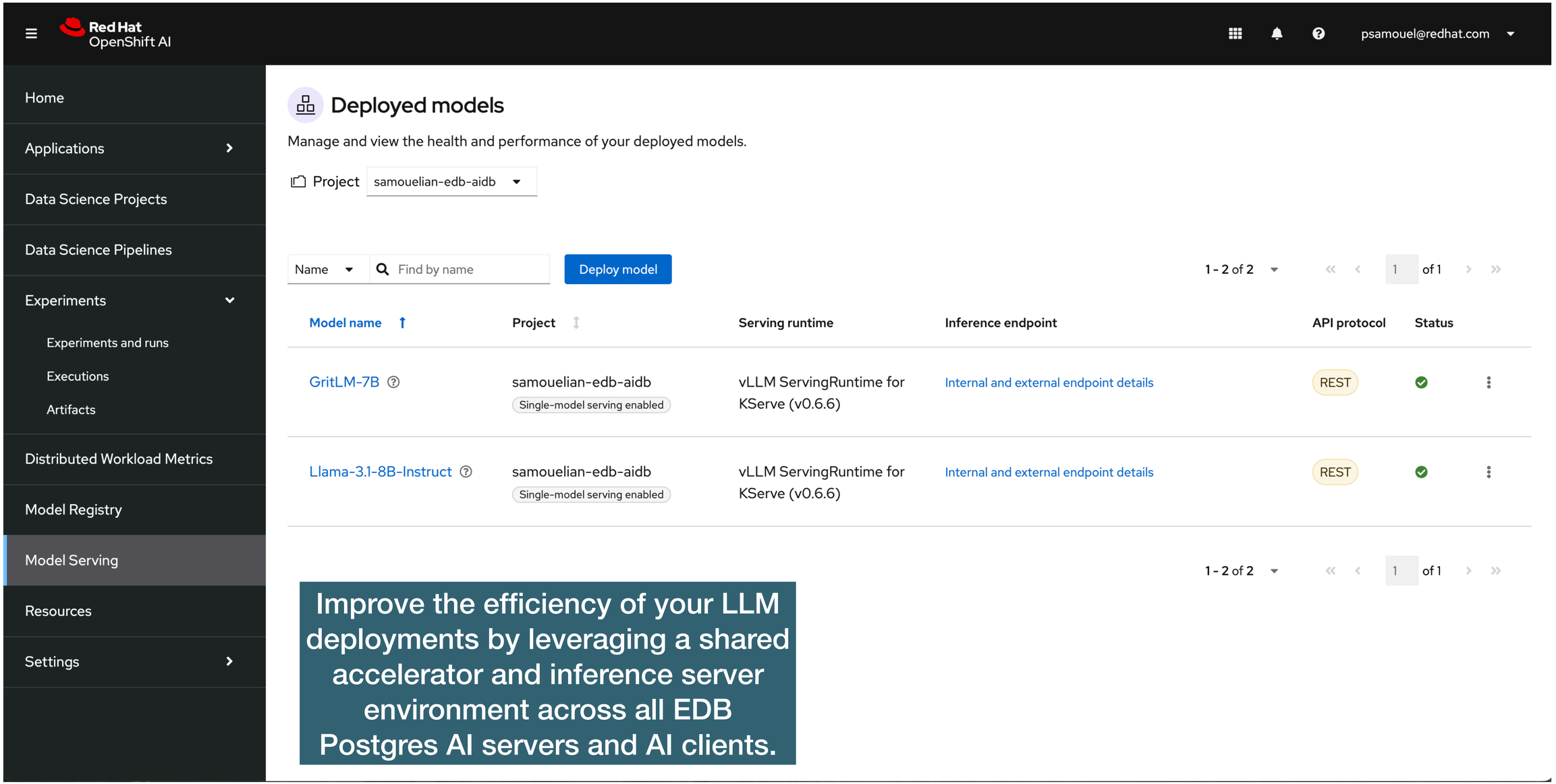
## Model location

Specify the model location by providing either the object storage de

Object storage

**Endpoint** *

Register model | Cancel


Red Hat
OpenShift AI
psamouel@redhat.com
Home
Applications
Data Science Projects
Data Science Pipelines
Experiments
Experiments and runs
Executions
Artifacts
Distributed Workload Metrics
Model Registry
Model Serving
Resources
Settings
Deployed models
Manage and view the health and performance of your deployed models.
Project
samouelian-edb-aidb
Name
Find by name
Deploy model
1 - 2 of 2
1
of 1
Model name
Project
Serving runtime
Inference endpoint
API protocol
Status
GritLM-7B
samouelian-edb-aidb
Single-model serving enabled
vLLM ServingRuntime for KServe (v0.6.6)
Internal and external endpoint details
REST
Llama-3.1-8B-Instruct
samouelian-edb-aidb
Single-model serving enabled
vLLM ServingRuntime for KServe (v0.6.6)
Internal and external endpoint details
REST
1 - 2 of 2
1
of 1
Improve the efficiency of your LLM deployments by leveraging a shared accelerator and inference server environment across all EDB Postgres AI servers and AI clients.

Red Hat OpenShift AI

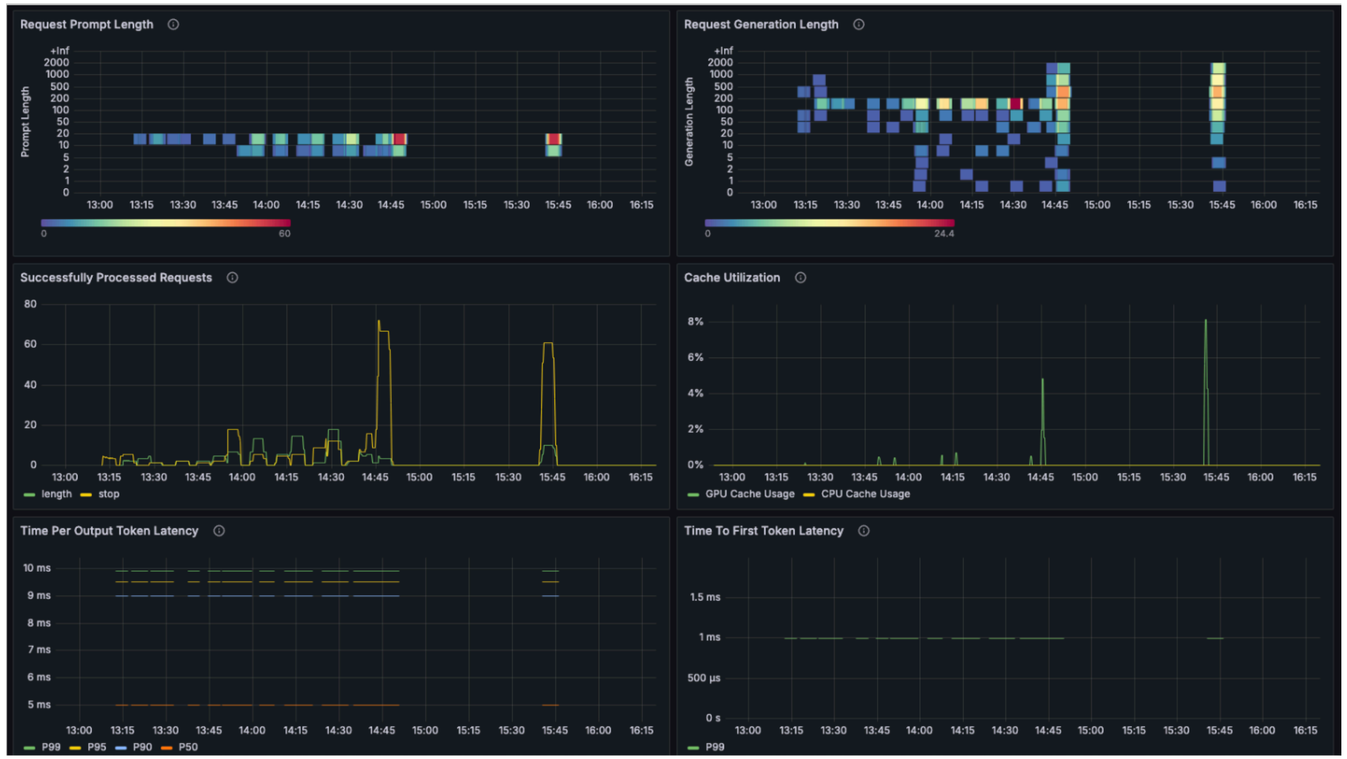
psamouel@redhat.com

- Home
- Applications
- Data Science Projects
- Data Science Pipelines
- Experiments
  - Experiments and runs
  - Executions
  - Artifacts
- Distributed Workload Metrics
- Model Registry
- Model Serving
- Resources
- Settings

# Distributed Workload Metrics

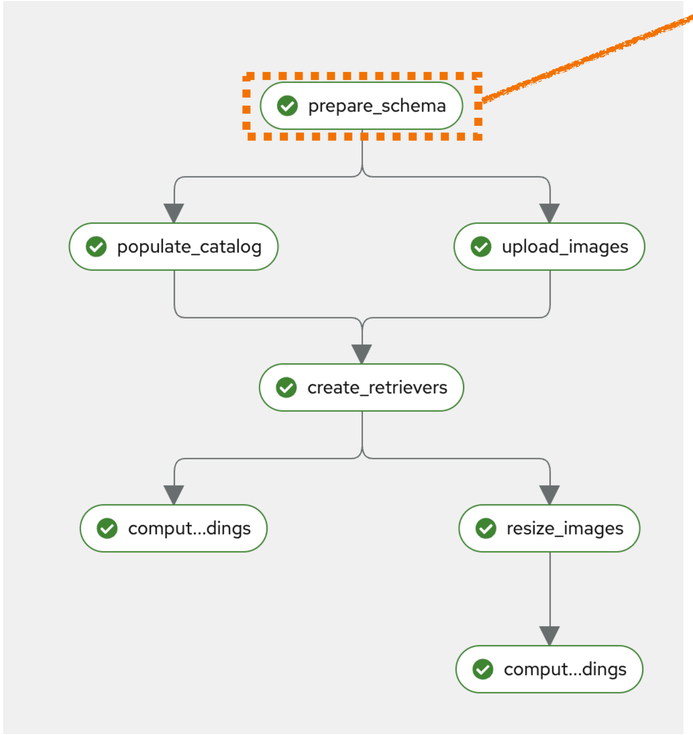
Monitor the metrics of your active resources.

Project: redhat-ods-monitoring




**Integrate with OpenShift AI workload metrics to monitor the resources your LLMs consume using Grafana dashboards.**

# Initial Step: Create Extensions




```python
with conn.cursor() as cur:
    cur.execute("CREATE EXTENSION IF NOT EXISTS aidb cascade;")
    cur.execute("CREATE EXTENSION IF NOT EXISTS pgfs;")

    cur.execute("DROP TABLE IF EXISTS products;")
    cur.execute("""
        CREATE TABLE IF NOT EXISTS products (
            img_id TEXT,
            gender VARCHAR(50),
            masterCategory VARCHAR(100),
            subCategory VARCHAR(100),
            articleType VARCHAR(100),
            baseColour VARCHAR(50),
            season TEXT,
            year INTEGER,
            usage TEXT NULL,
            productDisplayName TEXT NULL
        );
    """)

    cur.execute("DROP TABLE IF EXISTS product_review;")
    cur.execute(
    """CREATE TABLE IF NOT EXISTS product_review(
        user_id TEXT,
        product_id TEXT,
        rating INT,
        timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
        review TEXT
        );"""
    )
```

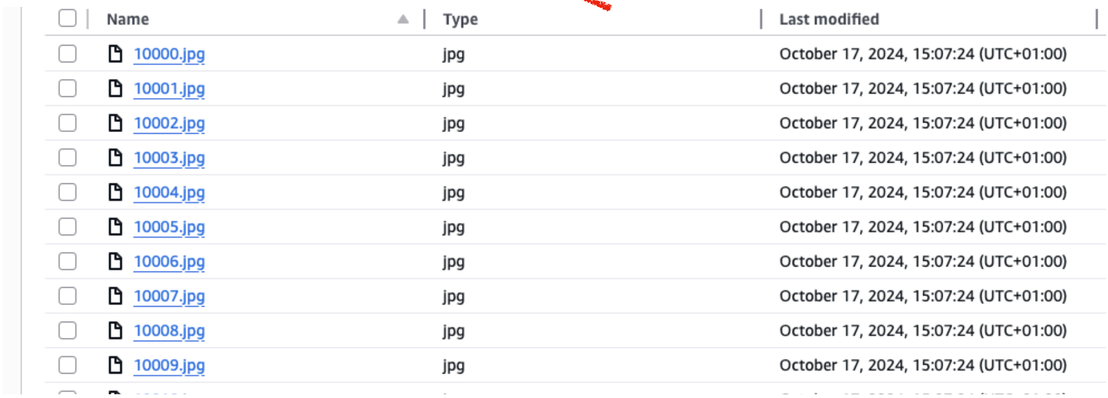
After connecting your PostgreSQL instance, create the two extensions of aidb and pgfs for volume and AI operations.

# Setting Up Object Storage Connection via pgfs

**First call pgfs functions to create connection with the object storage you're intended to use**

```
#Using EDB's pgfs extension, let's now connect to the in-cluster s3 bucket storage location
create_storage_location = (
    "SELECT pgfs.create_storage_location('images_s3_little', 's3://edb-aidb',"
    "msl_id => null, "
    f"options => '{{\"endpoint\": \"{s3_connection_profile.endpoint_url}\"}}', "
    f"credentials => '{{\"access_key_id\": \"{s3_connection_profile.access_key}\", \"secret_access_key\":\"{s3_connection_profile.secret_key}\"}}'); "
    )
cur.execute(create_storage_location)

#Now create a volume from a PGFS storage location for use as a data source in retrievers.
create_volume = f"SELECT aidb.create_volume('images_bucket_vol', 'images_s3_little', '{s3_connection_profile.recommender_images_path}/', 'Image');"
cur.execute(create_volume)
```



| Name | Type | Last modified |
|---|---|---|
| 10000.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10001.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10002.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10003.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10004.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10005.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10006.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10007.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10008.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |
| 10009.jpg | jpg | October 17, 2024, 15:07:24 (UTC+01:00) |

# Create a Local Model and a Retriever from a volume

Next, create the model you can use one of locally available multi-modal CLIP to use the model, later.

```
#Create a table to hold the image vectors
define_model = "SELECT aidb.create_model('recom_images', 'clip_local');"
```

```
create_retriever = (
    "SELECT aidb.create_retriever_for_volume("
    "name => 'recom_images',"
    "model_name => 'recom_images',"
    "source_volume_name => 'images_bucket_vol');"
   )
cur.execute(create_retriever)
```

aidb can generate embeddings for images from object storage with a single call once the volume name and AI model name have been provided as parameters. So, the next call is to create a retriever for volume.

```
    cur.execute("SELECT aidb.bulk_embedding('recom_images');")
```

To compute and store embeddings, all that needs to be run is bulk_embedding function by providing the name of the retriever.

# Create Model from Remote Source
# &
# Retriever from a table

```
define_model = (
    "select aidb.create_model("
    "'product_descriptions_embeddings',"
    "'embeddings',"
    f"'{{\"model\":\"{model_name}\", \"url\":\"{image_embedding_service}\","
    "\"dimensions\":4096}'::JSONB, "
    "'{\"api_key\":\"\"}'::JSONB, true); "
)
print(f"Create Model: {define_model}")

cur.execute(define_model)
```

Next we'll create a model that is provisioned by OpenShift AI model serving services. All that needs to be done is to pass the **model name** and the **model url** as parameters.

```
product_retriever = """SELECT aidb.create_retriever_for_table(
            name => 'recommend_products',
            model_name => 'product_descriptions_embeddings',
            source_table => 'products',
            source_key_column => 'img_id',
            source_data_column => 'productdisplayname',
            source_data_type => 'Text'
            );"""
```

Now, the retriever will be created for the products table in the database. The parameters require details like the table name, the source data column to be used to generate embeddings, and the AI model name to be used to generate embeddings.

```
cur.execute("SELECT aidb.bulk_embedding('recommend_products');")
```

To compute and store embeddings, all that needs to be run is bulk_embedding function by providing the name of the retriever.

EDB

# EDB Postgres AI & OpenShift AI

Select a Category:

Accessories

Enter search term:

flip flops for men

## 2go Active Gear USA Men Pack of Two Cushion Socks

Select the gender:

None

Or upload an image to search:

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG

Browse files

Search with Text

Reset

## ADIDAS Black Backpack

Results for 'flip flops for men':

Querying similar catalog took 0.9203 seconds.

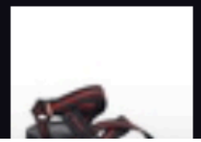
Number of elements retrieved: 5

**Numero Uno Men Black Flip Flops** Review

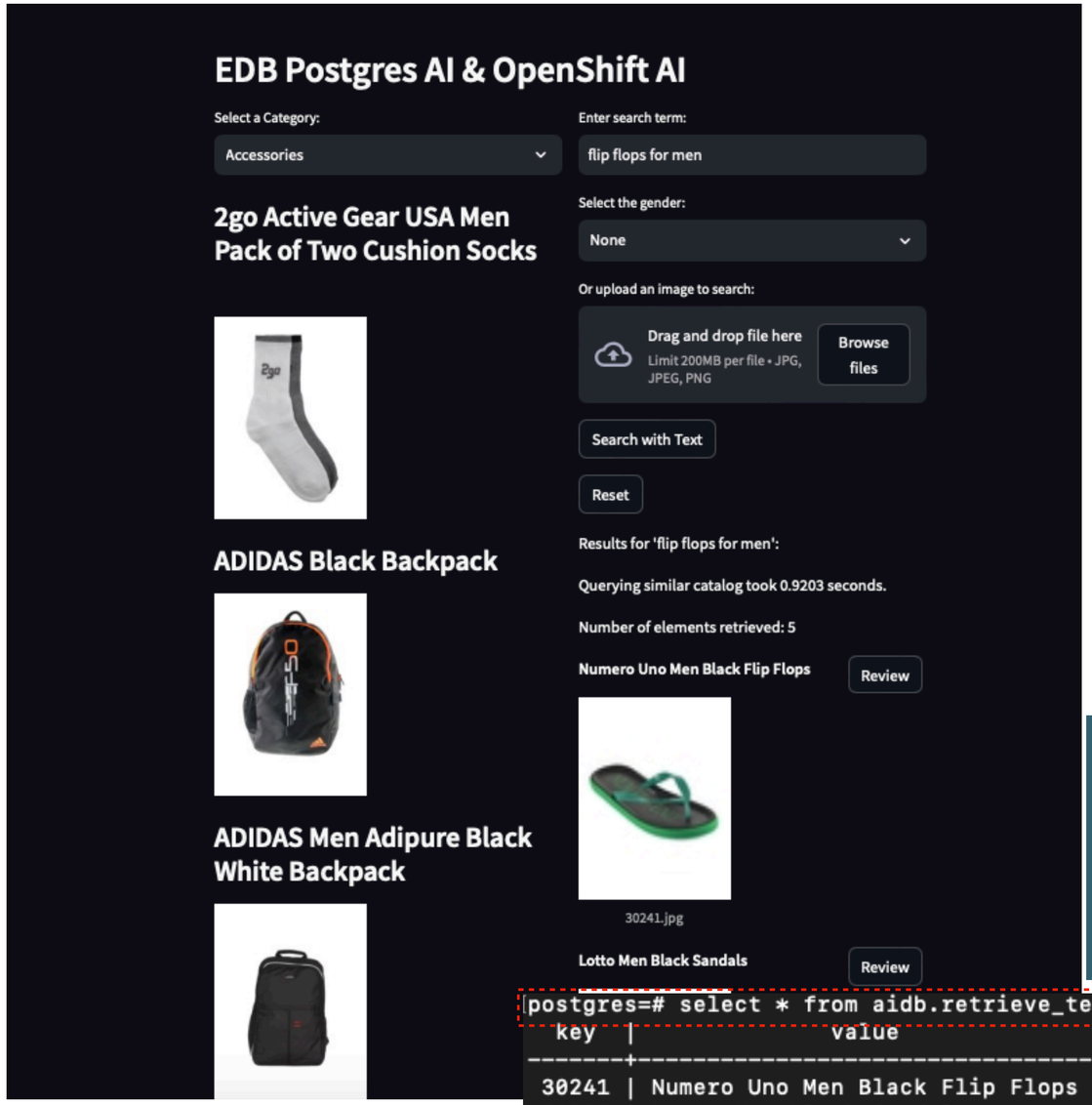
30241.jpg

## ADIDAS Men Adipure Black White Backpack

**Lotto Men Black Sandals** Review



EDB




```
postgres=# select * from aidb.retrieve_text('recommend_products', 'flip flops for men', 5);
  key  |              value               |      distance
-------+----------------------------------+--------------------
 30241 | Numero Uno Men Black Flip Flops  | 0.7788498440045555
 58227 | Lotto Men Black Sandals          | 0.7792318599034869
 11850 | Playboy Men Green Flip Flops     | 0.7822608892597847
 24105 | Playboy Men Green Flip Flops     | 0.7822608892597847
 24107 | Playboy Men Red Flip Flops       | 0.7840501196056017
(5 rows)
```

The aidb function call behind the scenes is in below terminal screenshot.

Usage:
aidb.retrieve_text(‘retriever_name’, ‘user_query’, topk);

EDB

## Numero Uno Men Black Flip Flops

30241.jpg

### Review Summary

Positive aspects of the Numero Uno Men Black Flip Flops include comfortable and supportive design, versatile color, soft straps, good grip, and durable construction. The flip flops are also lightweight, easy to clean, and offer a non-slip sole. While the reviews don't mention any major negative aspects, the reviewers note that they were pleasantly surprised by the quality given the affordable price.

### Review Labels

comfortable design | versatile color | soft straps | durability | quality

Once you click the Review button from the Search Page, Review Summary and Labels are generated live from user reviews on this page.

EDB

The aidb function call behind the scenes to generate review label is in below terminal screenshot.
Usage: aidb.decode_text(‘model_name’, ‘user_prompt’);

## Numero Uno Men Black Flip Flops

30241.jpg

### Review Summary

Positive aspects of the Numero Uno Men Black Flip Flops include comfortable and supportive design, versatile color, soft straps, good grip, and durable construction. The flip flops are also lightweight, easy to clean, and offer a non-slip sole. While the reviews don't mention any major negative aspects, the reviewers note that they were pleasantly surprised by the quality given the affordable price.

### Review Labels

comfortable design | versatile color | soft straps | durability | quality

```
postgres=#
postgres=# SELECT decode_text FROM aidb.decode_text('product_review_model', 'Extract a maximum of 5 concise labels (keywords or short phra
ses) from the following user review summary. The labels should represent both positive and negative aspects mentioned. Output *only* the l
abels as a single line of comma-separated values, with no introductory phrases, explanations, or formatting. Start outputting with Here ar
e the labels:Summary:The reviewer praises the sneakers for their genuine look, comfortable fit, and good performance, but notes some minor
 issues such as minor dust and dirt entry through ventilation, some colour fading, and a narrow tongue. Despite these issues, they find th
e sneakers to be a great value and a staple in their wardrobe. Overall, they recommend the Puma BasketBiz sneakers for their style, comfor
t, and performance.'
);
                                decode_text
-------------------------------------------------------------------------
 Good value, Comfortable fit, Colour fading, Narrow tongue, Minor dust entry
(1 row)

postgres=#
```

Learn more about how EDB and Red Hat partner to provide a powerful combination for the AI-driven enterprise.

Intelligent Shopping Experience powered by EDB Postgres AI & OpenShift AI