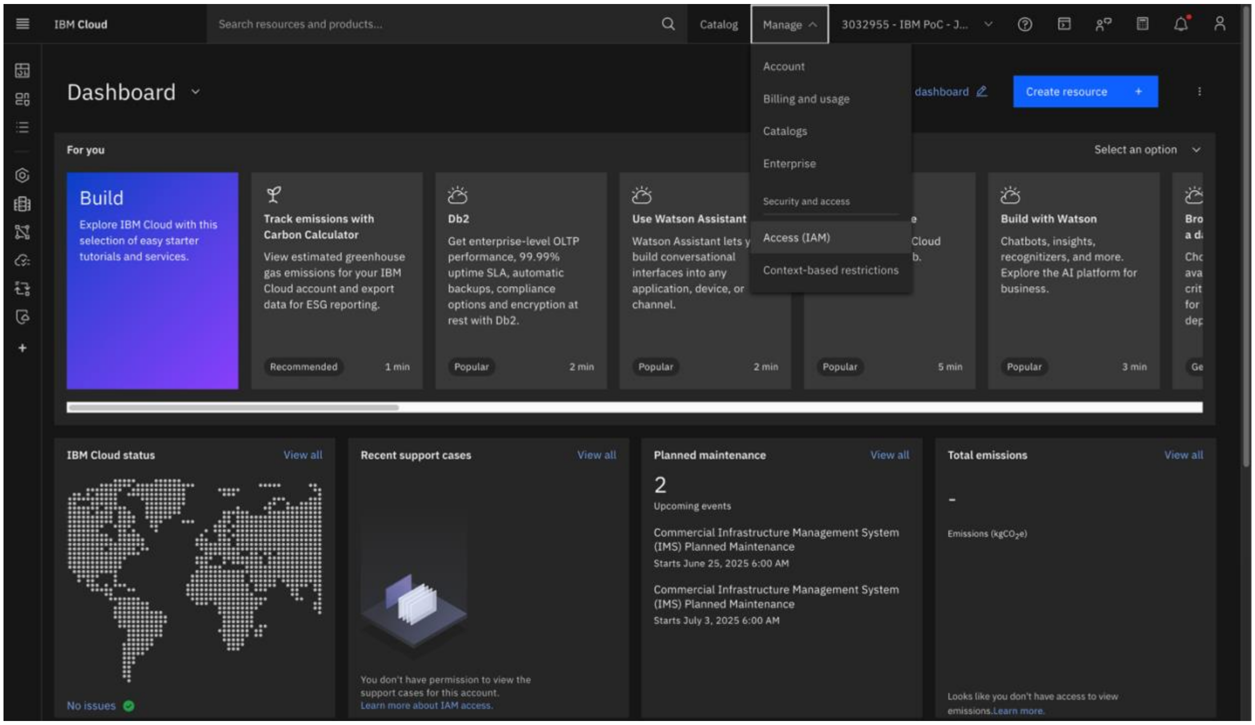
Step 0: Create a new project

1. Open your browser and go to IBM Cloud Login: https://cloud.ibm.com/login
2. Sign in with your IBM Cloud credentials.
3. Expand Manage → click Access (IAM).



Build
Explore IBM Cloud with this selection of easy starter tutorials and services.

Track emissions with Carbon Calculator
View estimated greenhouse gas emissions for your IBM Cloud account and export data for ESG reporting.

Db2
Get enterprise-level OLTP performance, 99.99% uptime SLA, automatic backups, compliance options and encryption at rest with Db2.

Use Watson Assistant
Watson Assistant lets y build conversational interfaces into any application, device, or channel.

Build with Watson
Chatbots, insights, recognizers, and more. Explore the AI platform for business.

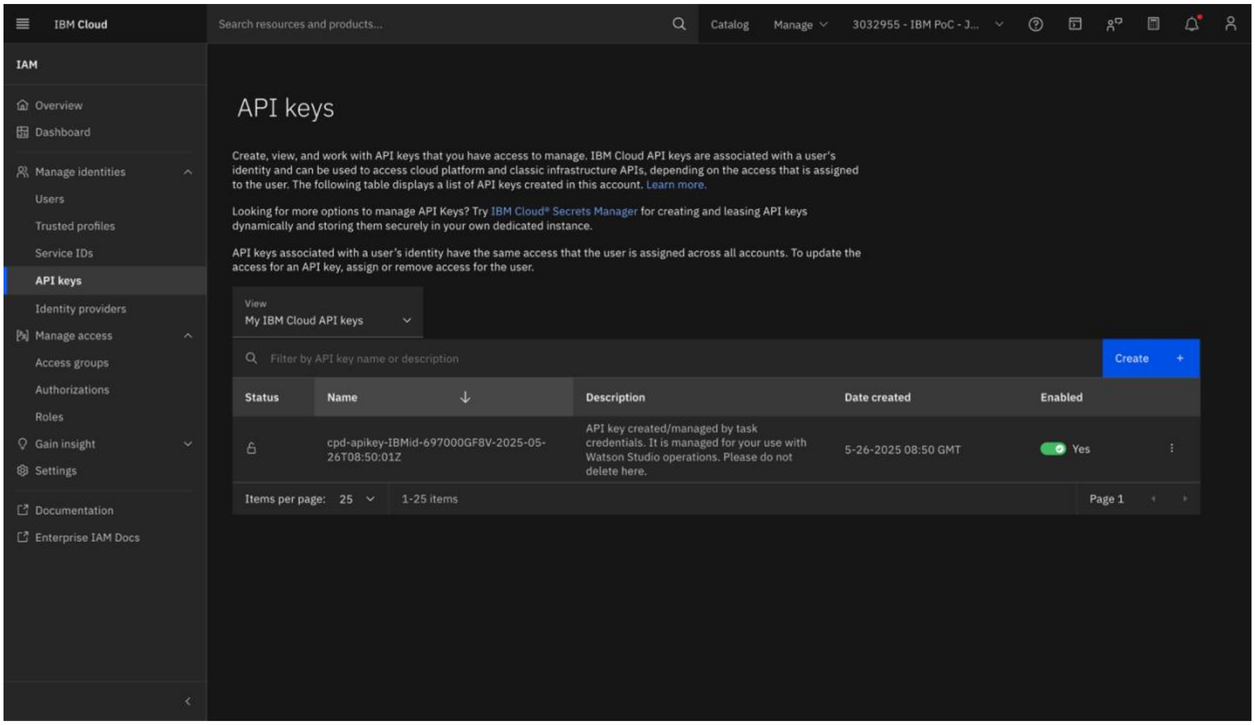
You don't have permission to view the support cases for this account. Learn more about IAM access.

Commercial Infrastructure Management System (IMS) Planned Maintenance
Starts June 25, 2025 6:00 AM

Commercial Infrastructure Management System (IMS) Planned Maintenance
Starts July 3, 2025 6:00 AM

Looks like you don't have access to view emissions.Learn more.

4. Select API Keys.



API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

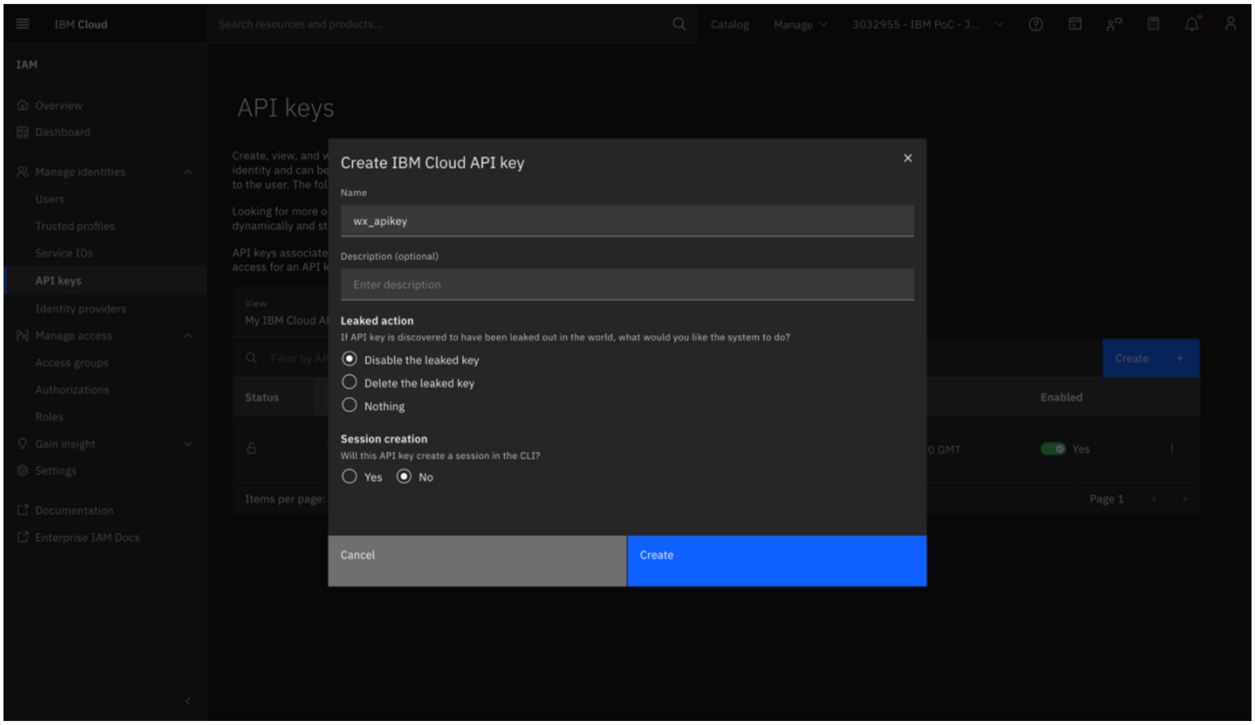
Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

View: My IBM Cloud API keys

Status	Name	Description	Date created	Enabled
	cpd-apikey-IBMid-697000GF8V-2025-05-26T08:50:01Z	API key created/managed by task credentials. It is managed for your use with Watson Studio operations. Please do not delete here.	5-26-2025 08:50 GMT	Yes

5. Click Create and enter the name of key: wx_apikey.



Create IBM Cloud API key

Name: wx_apikey

Description (optional): Enter description

Leaked action

If API key is discovered to have been leaked out in the world, what would you like the system to do?

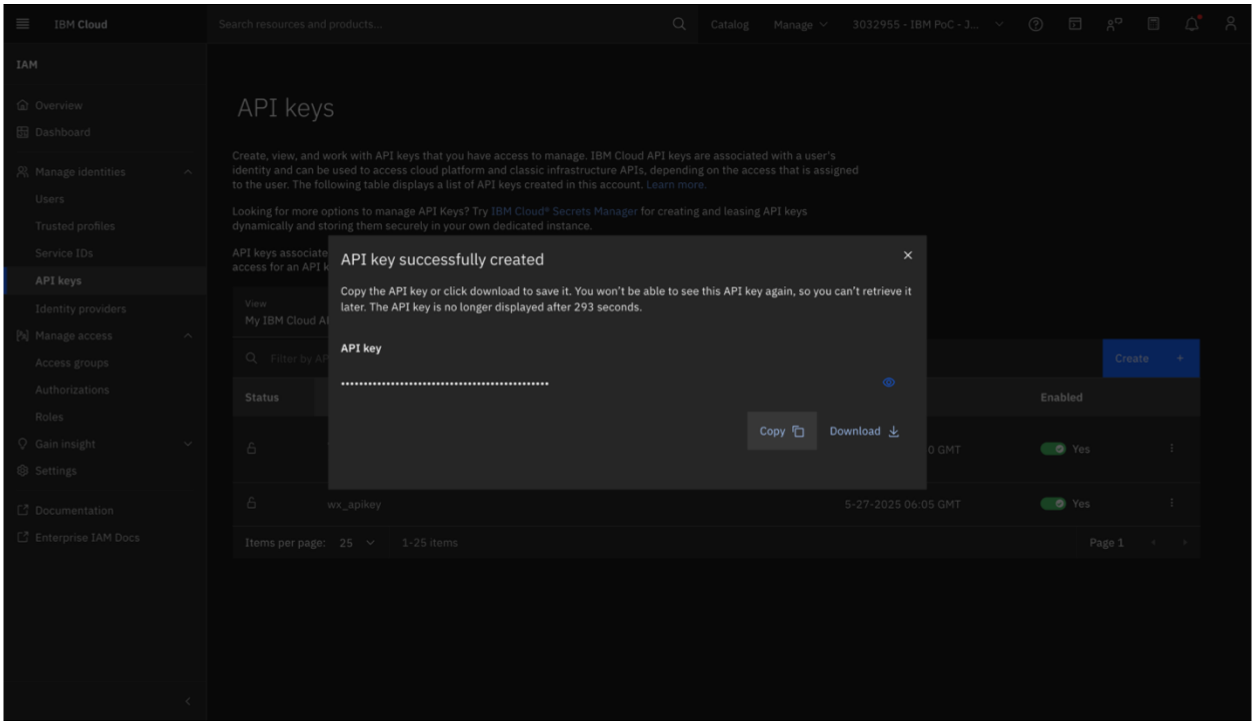
● Disable the leaked key
○ Delete the leaked key
○ Nothing

Session creation

Will this API key create a session in the CLI?

○ Yes ● No

6. Copy the displayed API key and save it **securely**.



API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. Learn more.

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

API keys associate... access for an API k...

View: My IBM Cloud AP...

API key successfully created

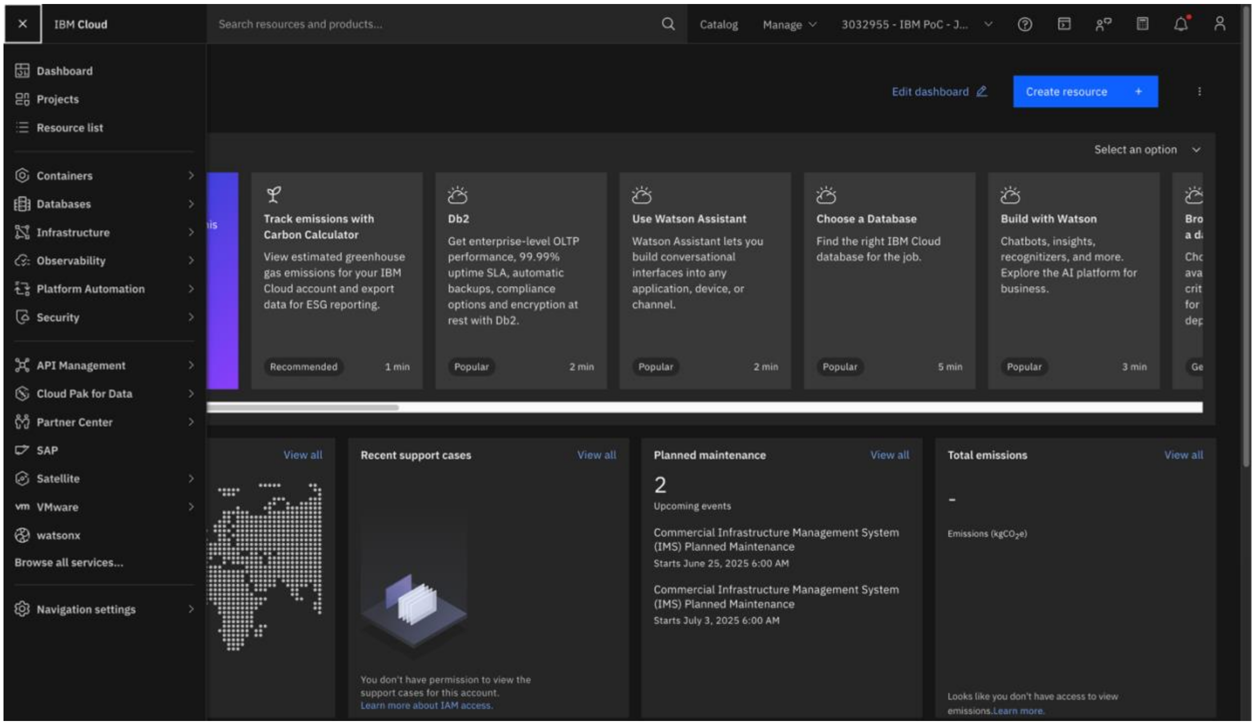
Copy the API key or click download to save it. You won't be able to see this API key again, so you can't retrieve it later. The API key is no longer displayed after 293 seconds.

API key: ••

Status	Name	Created	Enabled
		...0 GMT	Yes
	wx_apikey	5-27-2025 06:05 GMT	Yes

Items per page: 25 1-25 items Page 1

7. In the left-hand sidebar, click Resource list.



Track emissions with Carbon Calculator — View estimated greenhouse gas emissions for your IBM Cloud account and export data for ESG reporting. Recommended 1 min

Db2 — Get enterprise-level OLTP performance, 99.99% uptime SLA, automatic backups, compliance options and encryption at rest with Db2. Popular 2 min

Use Watson Assistant — Watson Assistant lets you build conversational interfaces into any application, device, or channel. Popular 2 min

Choose a Database — Find the right IBM Cloud database for the job. Popular 5 min

Build with Watson — Chatbots, insights, recognizers, and more. Explore the AI platform for business. Popular 3 min

Recent support cases — View all

You don't have permission to view the support cases for this account. Learn more about IAM access.

Planned maintenance — View all

2 Upcoming events

Commercial Infrastructure Management System (IMS) Planned Maintenance
Starts June 25, 2025 6:00 AM

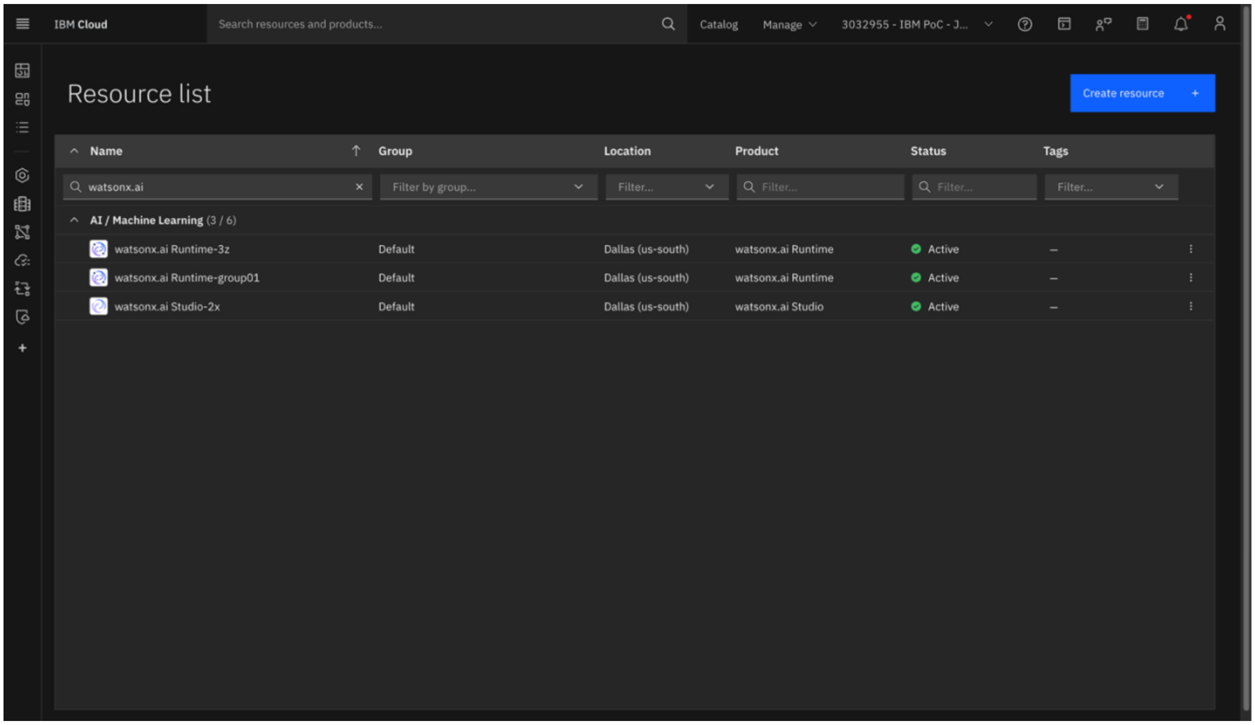
Commercial Infrastructure Management System (IMS) Planned Maintenance
Starts July 3, 2025 6:00 AM

Total emissions — View all

-
Emissions (kgCO$_2$e)

Looks like you don't have access to view emissions. Learn more.

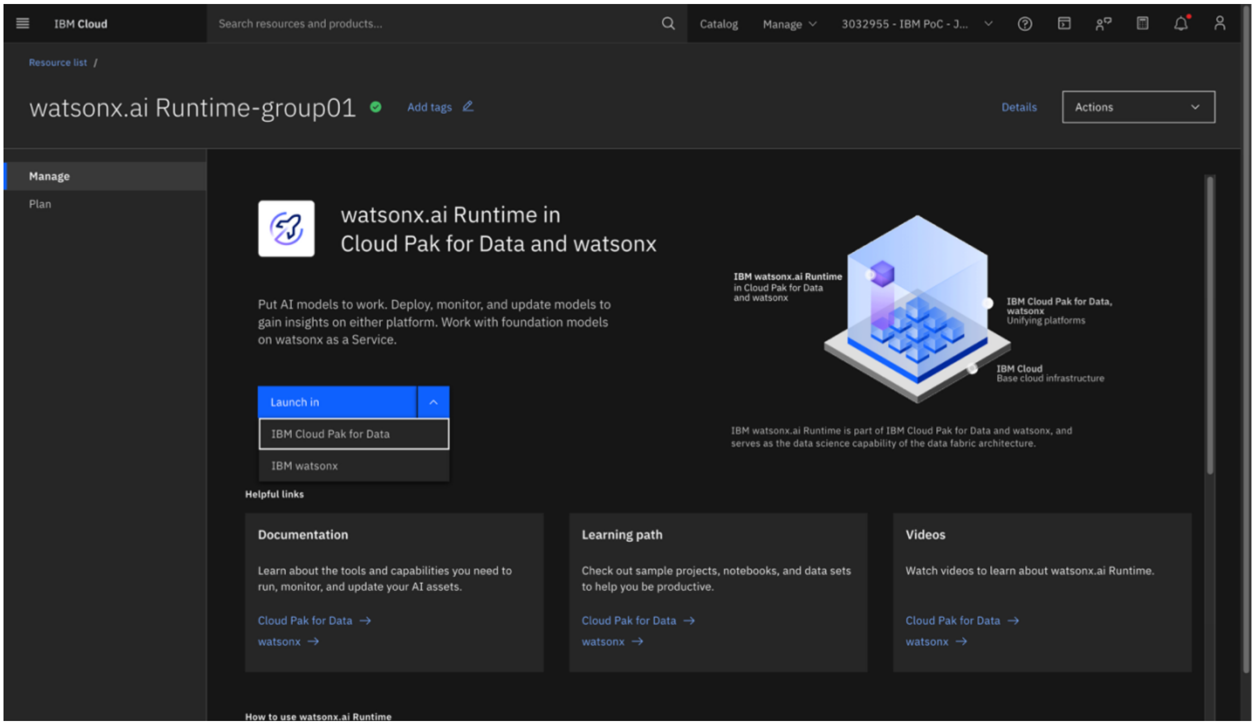
8. In the search box, type watsonx.ai, then from the results select your assigned service watsonx.ai group.



Resource list

Name	Group	Location	Product	Status	Tags
watsonx.ai	Filter by group...	Filter...	Filter...	Filter...	Filter...
AI / Machine Learning (3 / 6)					
watsonx.ai Runtime-3z	Default	Dallas (us-south)	watsonx.ai Runtime	Active	–
watsonx.ai Runtime-group01	Default	Dallas (us-south)	watsonx.ai Runtime	Active	–
watsonx.ai Studio-2x	Default	Dallas (us-south)	watsonx.ai Studio	Active	–

9. Click Launch in → **IBM watsonx** to open the watsonx console.



watsonx.ai Runtime-group01

watsonx.ai Runtime in Cloud Pak for Data and watsonx

Put AI models to work. Deploy, monitor, and update models to gain insights on either platform. Work with foundation models on watsonx as a Service.

Launch in
IBM Cloud Pak for Data
IBM watsonx
IBM watsonx.ai Runtime in Cloud Pak for Data and watsonx
IBM Cloud Pak for Data, watsonx
Unifying platforms
IBM Cloud
Base cloud infrastructure

IBM watsonx.ai Runtime is part of IBM Cloud Pak for Data and watsonx, and serves as the data science capability of the data fabric architecture.

Helpful links

Documentation

Learn about the tools and capabilities you need to run, monitor, and update your AI assets.

Learning path

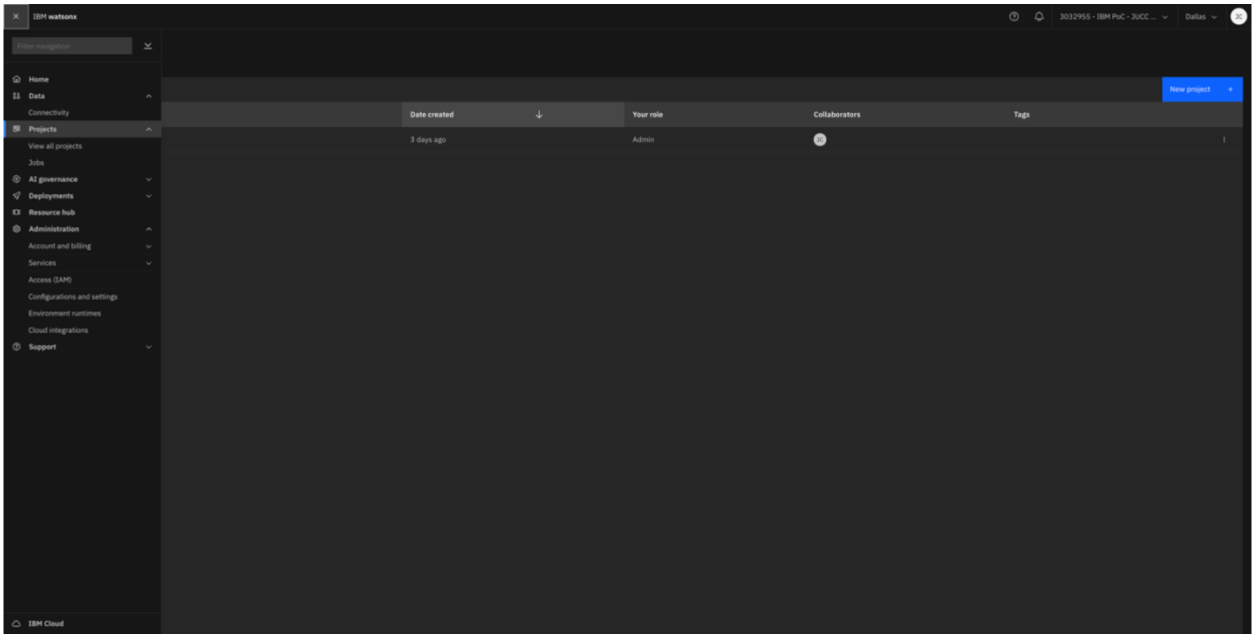
Check out sample projects, notebooks, and data sets to help you be productive.

Videos

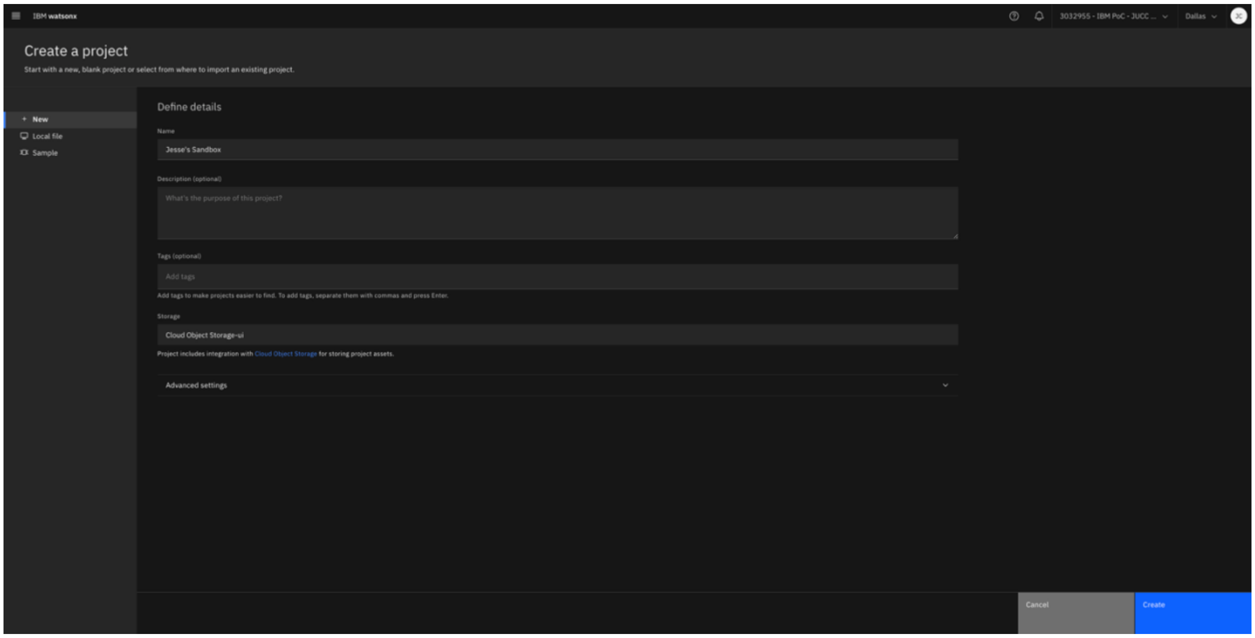
Watch videos to learn about watsonx.ai Runtime.

How to use watsonx.ai Runtime

10. In the IBM watsonx console sidebar, click Projects.
11. Click the New project button in the top-right.
12. Ensure the New tab is selected.
13. In Name, enter <Your Name>'s Sandbox.
14. (Optional) Fill in Description and Tags as desired.



Date created	Your role	Collaborators	Tags
3 days ago	Admin	JC	



Create a project

Start with a new, blank project or select from where to import an existing project.

Define details

Name: Jesse's Sandbox

Description (optional): What's the purpose of this project?

Tags (optional): Add tags

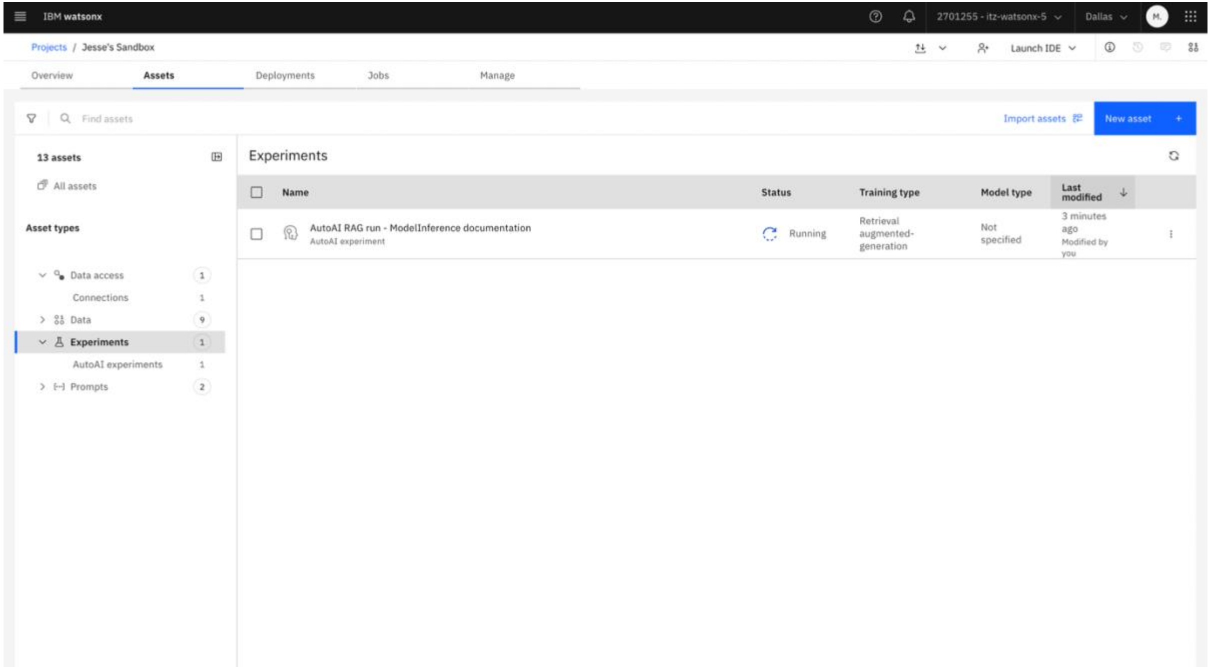
Add tags to make projects easier to find. To add tags, separate them with commas and press Enter.

Storage: Cloud Object Storage-ui

Project includes integration with Cloud Object Storage for storing project assets.

Step 0.5: Import Elasticsearch Connection

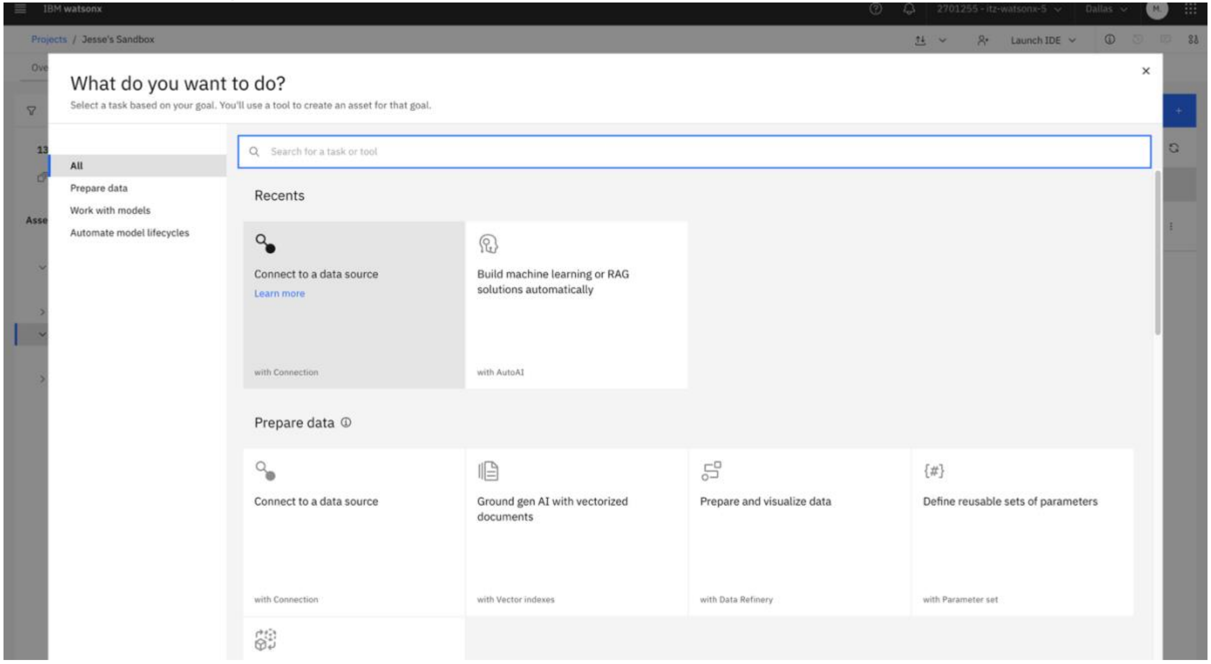
1. Click on the “New Asset” button in the project.



Experiments

	Name	Status	Training type	Model type	Last modified
□	AutoAI RAG run - ModelInference documentation AutoAI experiment	Running	Retrieval augmented-generation	Not specified	3 minutes ago Modified by you

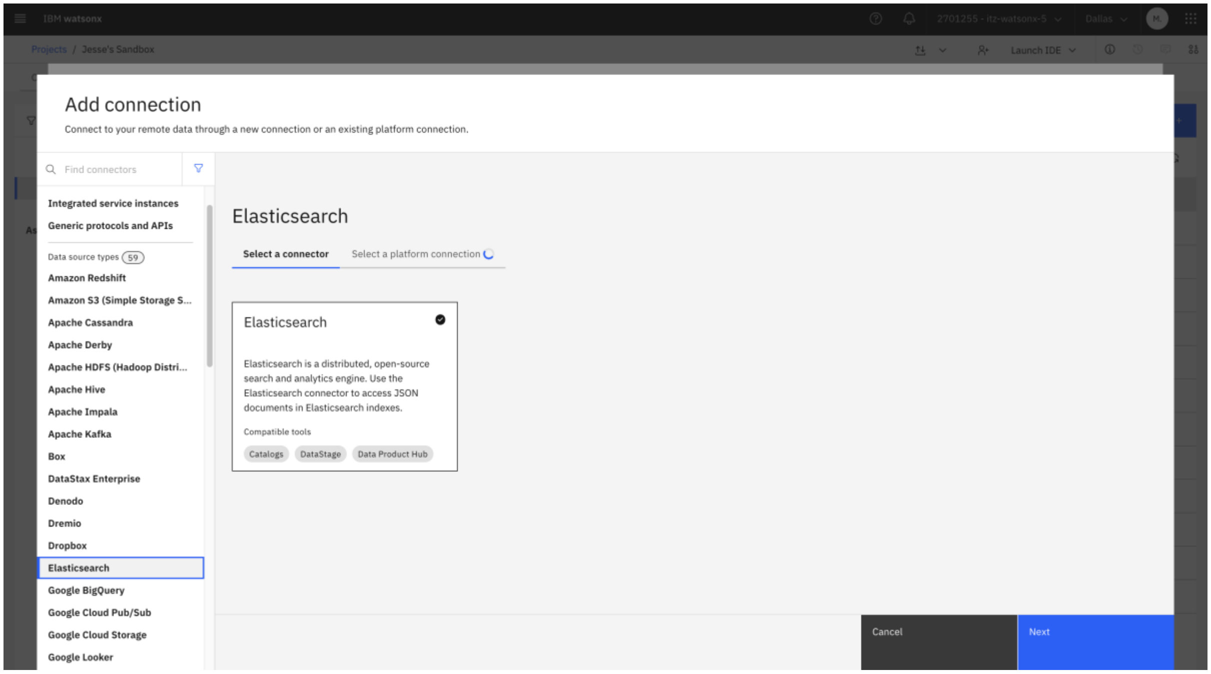
2. Now choose the option “Connect to a data source”.



What do you want to do?

Select a task based on your goal. You'll use a tool to create an asset for that goal.

3. Select Elasticsearch from the options on left panel. Then click on Next.



Add connection

Connect to your remote data through a new connection or an existing platform connection.

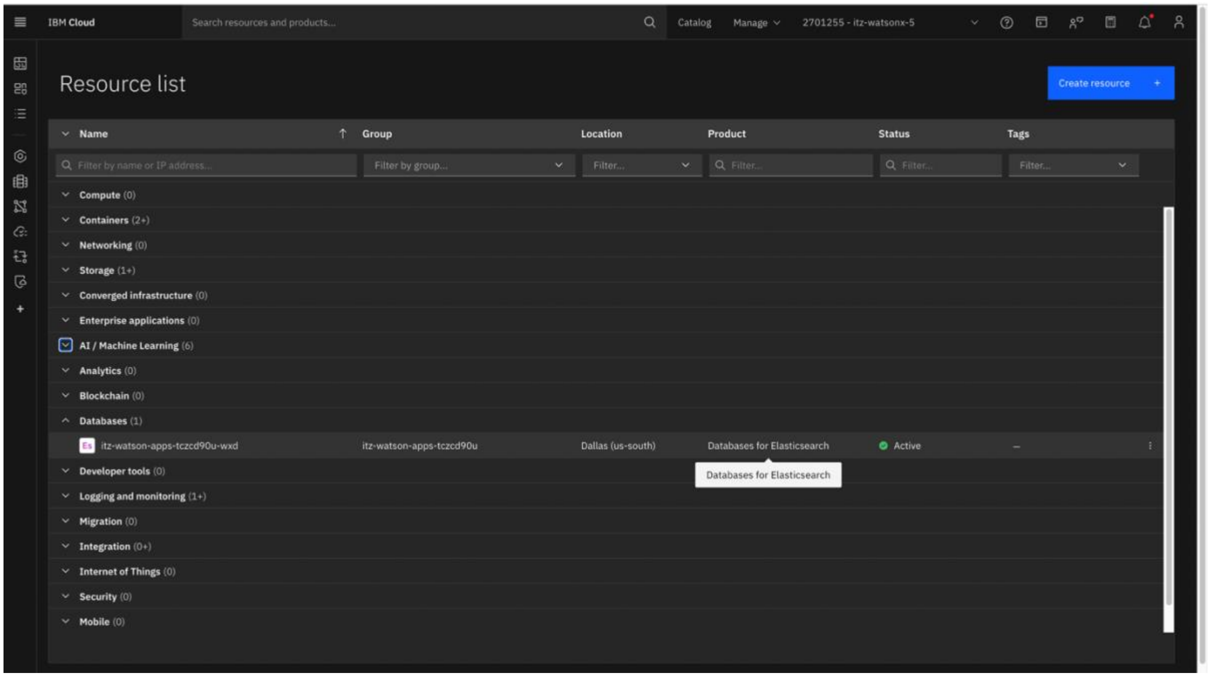
Elasticsearch

Elasticsearch

Elasticsearch is a distributed, open-source search and analytics engine. Use the Elasticsearch connector to access JSON documents in Elasticsearch indexes.

Compatible tools

4. In a separate tab, navigate to the Databases for Elasticsearch service on IBM Cloud and copy the hostname, port, and TLS certificate under the Endpoints section of the Overview tab.

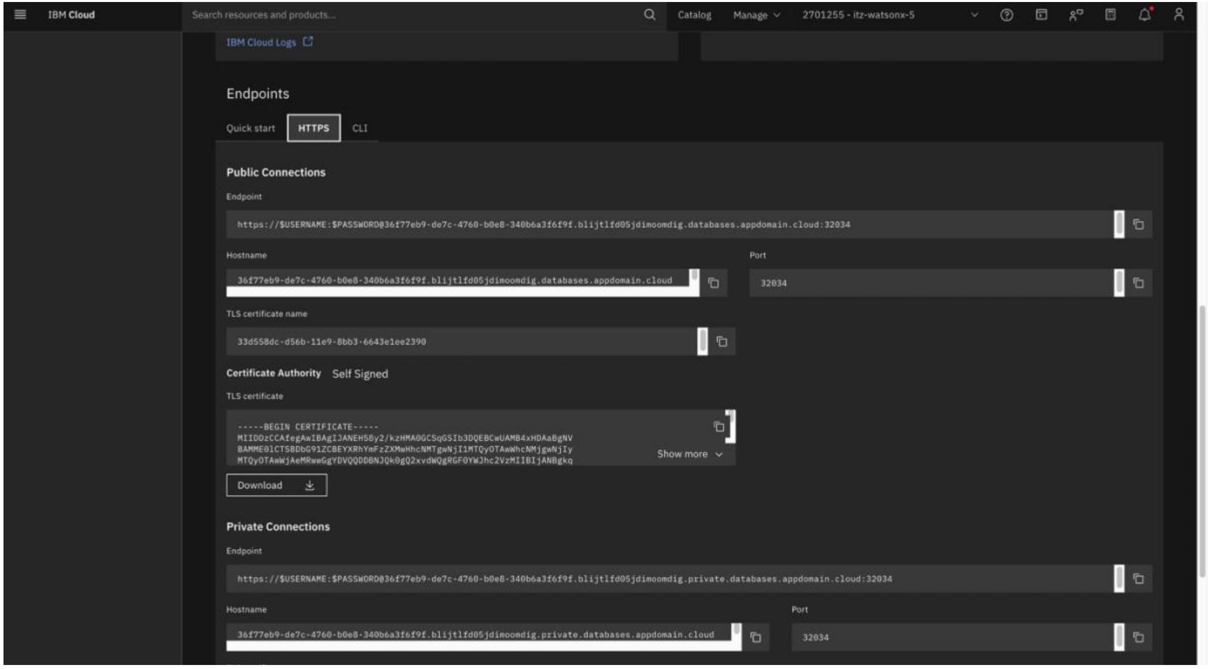


Resource list

Name	Group	Location	Product	Status	Tags
Compute (0)					
Containers (2+)					
Networking (0)					
Storage (1+)					
Converged infrastructure (0)					
Enterprise applications (0)					
AI / Machine Learning (6)					
Analytics (0)					
Blockchain (0)					
Databases (1)					
itz-watson-apps-tczcd90u-wxd	itz-watson-apps-tczcd90u	Dallas (us-south)	Databases for Elasticsearch	Active	–
Developer tools (0)					
Logging and monitoring (1+)					
Migration (0)					
Integration (0+)					
Internet of Things (0)					
Security (0)					
Mobile (0)					

Copy the TLS certificate and add it to Elasticsearch connection on the watsonx.ai platform.

5. From the HTTPS tab you can note down the Hostname and Port number.



Endpoints

Public Connections

Endpoint

`https://$USERNAME:$PASSWORD@36f77eb9-de7c-4760-b0e8-340b6a3f6f9f.blijtlfd05jdimoomdig.databases.appdomain.cloud:32034`

Hostname: `36f77eb9-de7c-4760-b0e8-340b6a3f6f9f.blijtlfd05jdimoomdig.databases.appdomain.cloud`

Port: 32034

TLS certificate name: `33d558dc-d56b-11e9-8bb3-6643e1ee2390`

Certificate Authority Self Signed

TLS certificate

```
-----BEGIN CERTIFICATE-----
MIIDDzCCAfegAwIBAgIJANEH58y2/kzHMA0GCSqGSIb3DQEBCwUAMB4xHDAaBgNV
BAMME0lCTSBDbG91ZCBEYXRhYmFzZXMwHhcNMTgwNjI1MTQyOTAwWhcNMjgwNjIy
MTQyOTAwWjAeMRwwGgYDVQQDDBNJQk0gQ2xvdWQgRGF0YWJhc2VzMIIBIjANBgkq
```

Private Connections

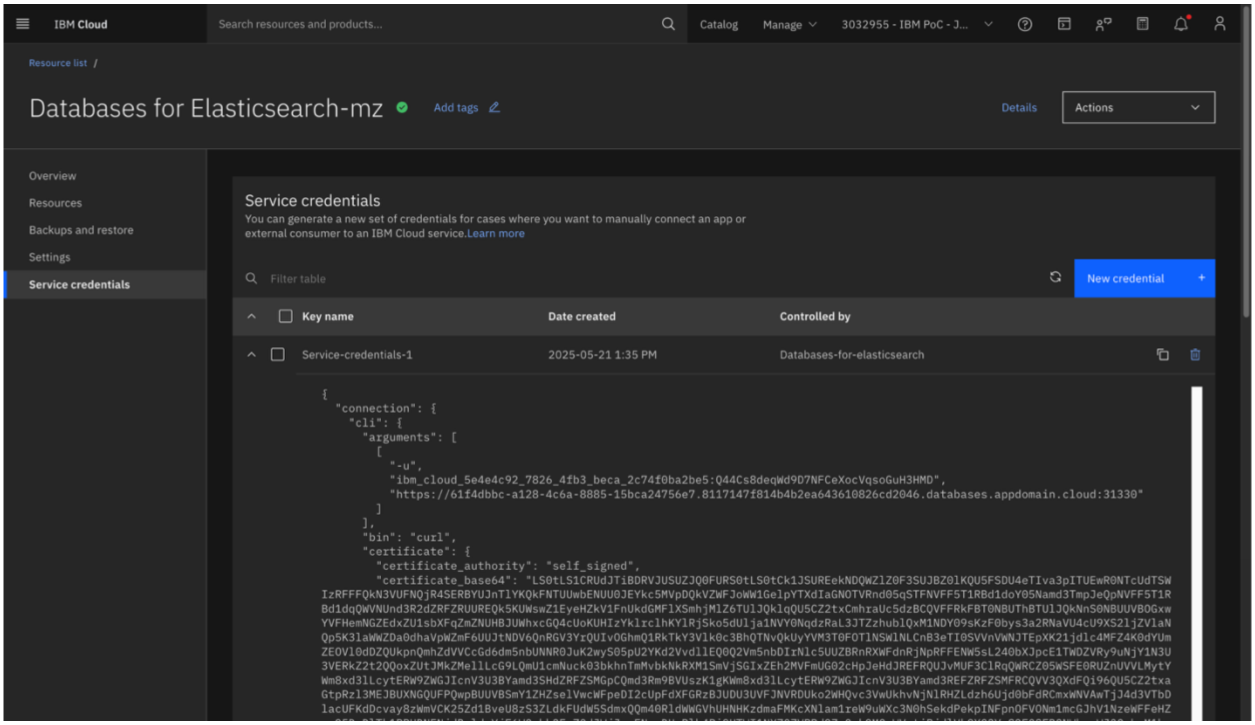
Endpoint

`https://$USERNAME:$PASSWORD@36f77eb9-de7c-4760-b0e8-340b6a3f6f9f.blijtlfd05jdimoomdig.private.databases.appdomain.cloud:32034`

Hostname: `36f77eb9-de7c-4760-b0e8-340b6a3f6f9f.blijtlfd05jdimoomdig.private.databases.appdomain.cloud`

Port: 32034

6. Go to the service credentials tab of the service, click on the arrow to expand it. Copy the username and password under "**connection.https.authentication**" in the service credential JSON.



Databases for Elasticsearch-mz

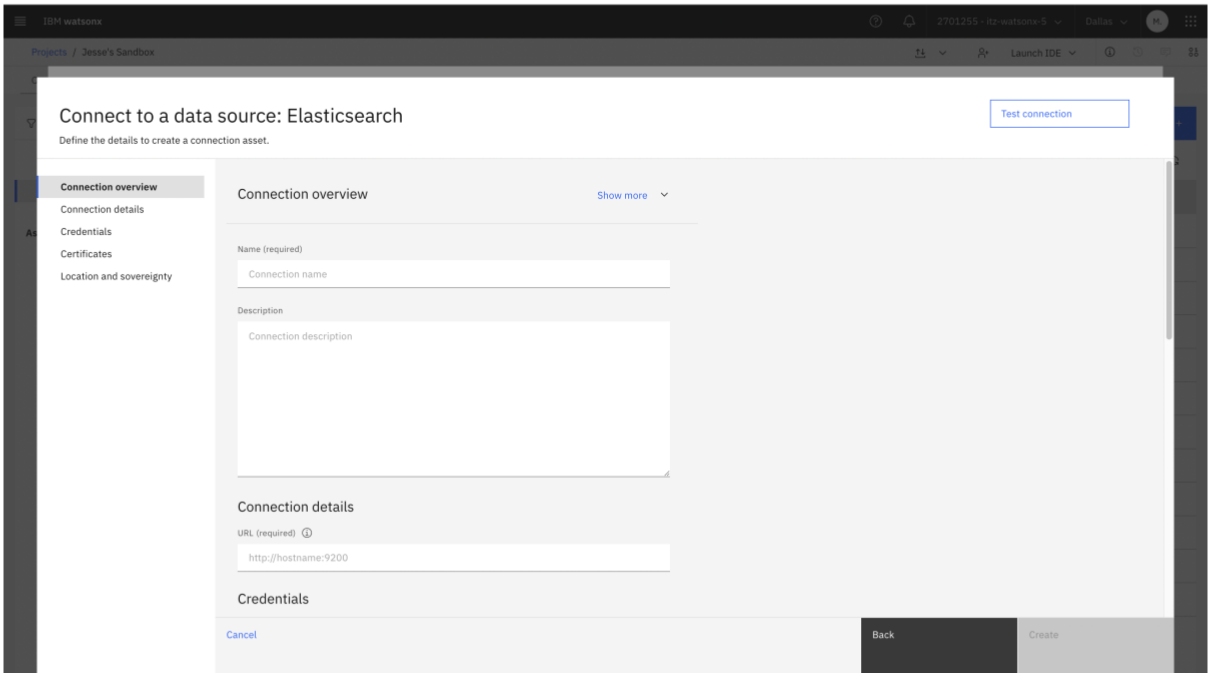
Service credentials

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. Learn more

Key name	Date created	Controlled by
Service-credentials-1	2025-05-21 1:35 PM	Databases-for-elasticsearch

```
{
  "connection": {
    "cli": {
      "arguments": [
        [
          "-u",
          "ibm_cloud_5e4e4c92_7826_4fb3_beca_2c74f0ba2be5:Q44Cs8deqWd9D7NFCeXocVqsoGuH3HMD",
          "https://61f4dbbc-a128-4c6a-8885-15bca24756e7.8117147f814b4b2ea643610826cd2046.databases.appdomain.cloud:31330"
        ]
      ],
      "bin": "curl",
      "certificate": {
        "certificate_authority": "self_signed",
        "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUREekNDQWZlZ0F3SUJBZ0lKQU5FSDU4eTIva3pITUEwR0NTcUdTSW
IzRFFFQkN3VUFNQjR4SERBYUJnTlYKQkFNTUUwbENUU0JEYkc5MVpDQkVZWFJoWW1GelpYTXdIaGNOTVRnd05qSTFNVFF5T1RBd1doY05Namd3TmpJeQpNVFF5T1R
Bd1dqQWVNUnd3R2dZRFZRUURQk5KUWswZ1EyeHZkV1FnUkdGMFlXSmhjMlZ6TUlJQklqQU5CZ2txCmhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBOGxw
YVFHemNGZEdxZU1sbXFqZmZNUHBJUWhxcGQ4cUoKUHIzYklrclhKYlRjSko5dUlja1NVY0NqdzRaL3JTZzhublQxM1NDY09sKzF0bys3a2RNaVU4cU9XS2ljZVlaN
Qp5K3laWWZDa0dhaVpWZmF6UUJtNDV6QnRGV3YrQUIvOGhmQ1RkTkY3Vlk0c3BhQTNvQkUyYVM3T0FOTlNSWlNLCnB3eTI0SVVnVWNJTEpXK21jdlc4MFZ4K0dYUm
ZEOVl0dDZQUkpnQmhZdVVCcGd6dm5nbUNNR0JuK2wyS05pU2YKd2VvdllEQ0Q2Vm5nbDIrNlc5UUZBRnRXWFdnRjNpRFFENW5sL240bXJpcE1TWDZVRy9uNjY1N3U
3VERkZ2t2QQoxZUtJMkZMellLcG9LQmU1cmNuck03bkhnTmMvbkNkRXM1SmVjSGIxZEh2MVFmUG02cHpJeHdJREFRQUJvMUF3ClRqQWRCZ05WSFE0RUZnUVVLMytY
Wm8xd3lLcytERW9ZWGJIcnV3U3BYamd3SHdZRFZSMGpCQmd3Rm9BVUszK1gKWm8xd3lLcytERW9ZWGJIcnV3U3BYamd3REFZRFZSMFRCQVV3QXdFQi96QU5CZ2txa
GtpRzl3MEJBUXNGQUFPQwpBUUVBSmY1ZHZselVwcWFpeDI2cUpFdXFGRzBJUDU3UVFJNVRDUko2WHQvc3VwUkhvNjNlRHZLdzh6Ujd0bFdRCmxWNVAwTjJ4d3VTbD
lacUFKdDcvay8zWmVCK25Zd1BveU8zS3ZLdkFUdW5SdmxQQm40RldWWGVhUHNHKzdmaFMKcXNlam1reW9uWXc3N0hSekdPekpINFpnOFVONm1mcGJhV1NzeWFFeHZ
```

7. Return to the project connection and set the following fields. Give the connection a name of: “Elasticsearch connection”



Connect to a data source: Elasticsearch

Define the details to create a connection asset.

Test connection

- Connection overview
- Connection details
- Credentials
- Certificates
- Location and sovereignty

Connection overview

Show more

Name (required)

Connection name

Description

Connection description

Connection details

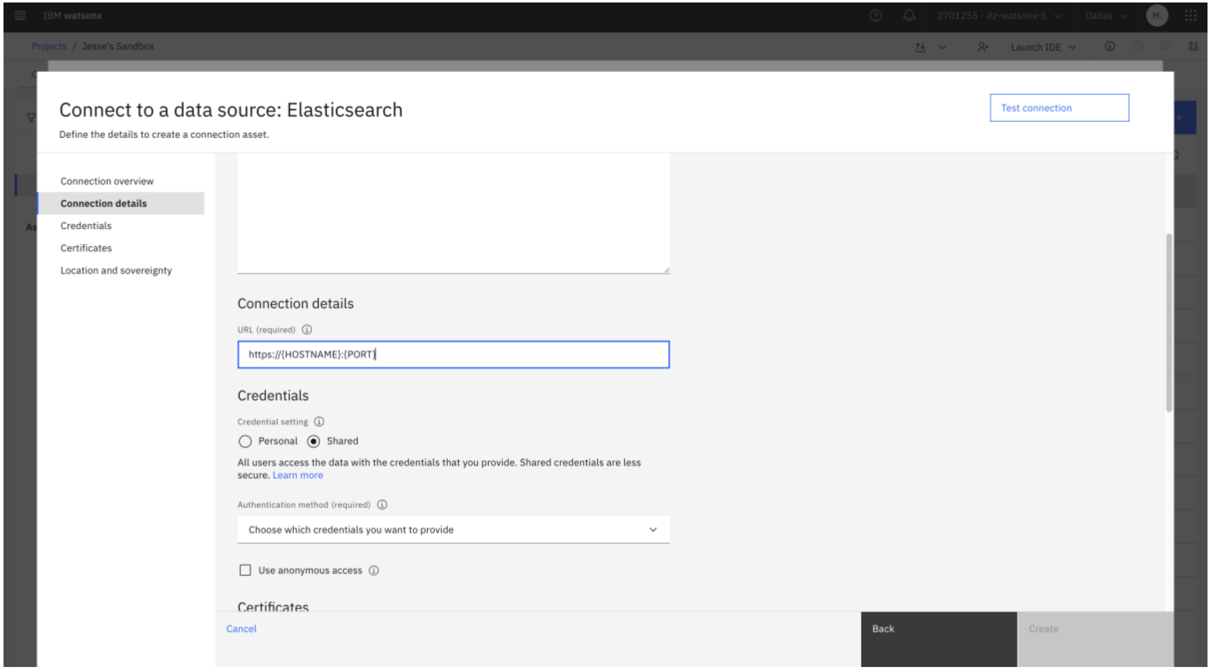
URL (required)

http://hostname:9200

Credentials

Cancel | Back | Create

8. The URL should be formatted as https://{HOSTNAME}:{PORT}.



Connect to a data source: Elasticsearch

Define the details to create a connection asset.

Test connection

- Connection overview
- Connection details
- Credentials
- Certificates
- Location and sovereignty

Connection details

URL (required)

https://{HOSTNAME}:{PORT}

Credentials

Credential setting

○ Personal ● Shared

All users access the data with the credentials that you provide. Shared credentials are less secure. Learn more

Authentication method (required)

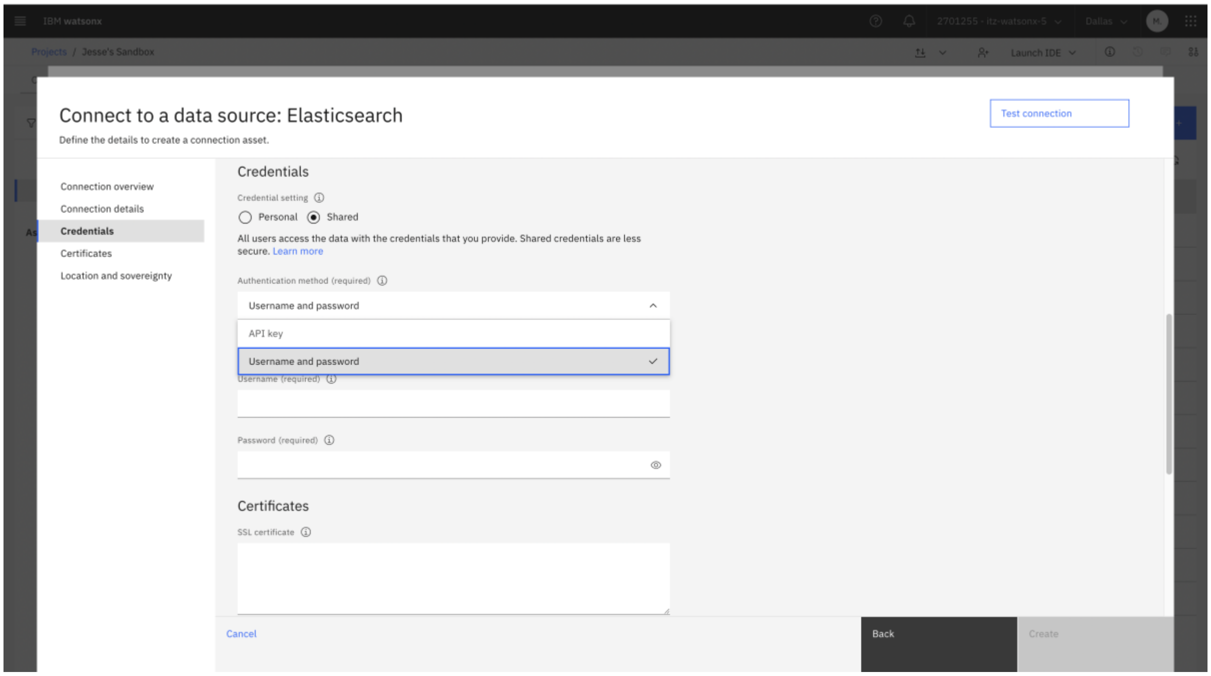
Choose which credentials you want to provide

□ Use anonymous access

Certificates

Cancel | Back | Create

9. Now for authentication, from the dropdown select Username and password.



Connect to a data source: Elasticsearch

Define the details to create a connection asset.

Credentials

Credential setting

○ Personal ● Shared

All users access the data with the credentials that you provide. Shared credentials are less secure. Learn more

Authentication method (required): Username and password

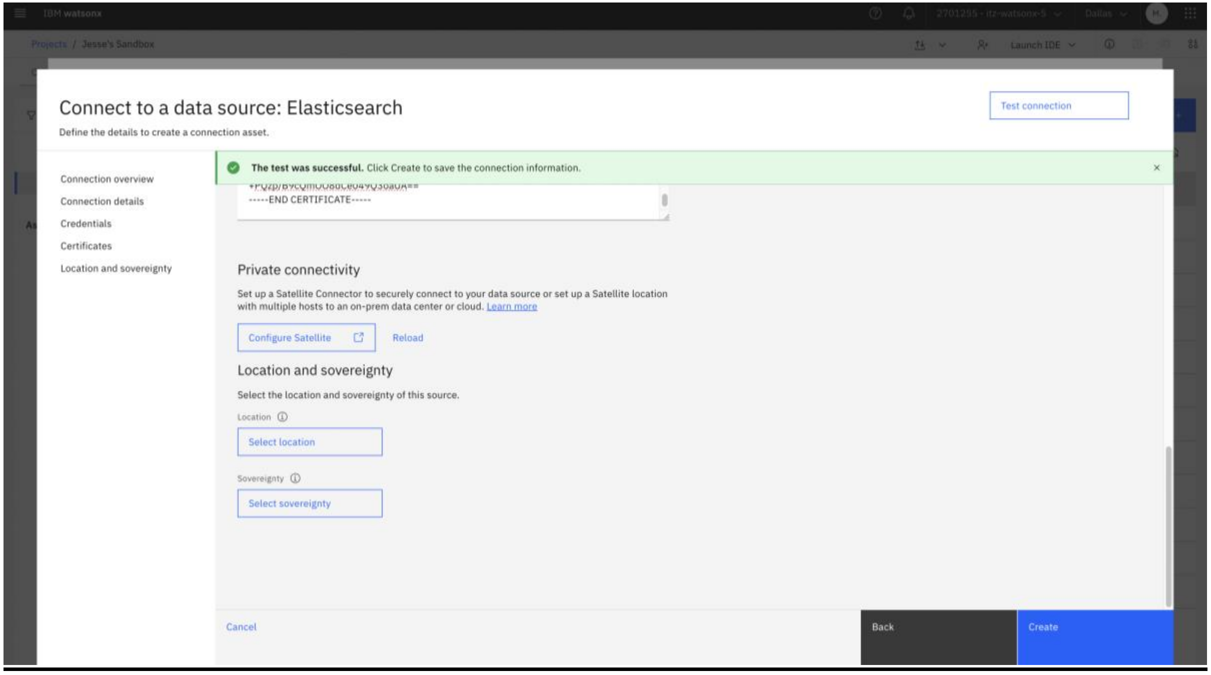
Username (required)

Password (required)

Certificates

SSL certificate

10. The username and password should be the ones copied from the service credentials.
11. The SSL certificate should be the TLS certificate.
12. Click on Test connection on the top, if that is successful then click on create.



Connect to a data source: Elasticsearch

Define the details to create a connection asset.

The test was successful. Click Create to save the connection information.

```
-----END CERTIFICATE-----
```

Private connectivity

Set up a Satellite Connector to securely connect to your data source or set up a Satellite location with multiple hosts to an on-prem data center or cloud. Learn more

Location and sovereignty

Select the location and sovereignty of this source.

Location

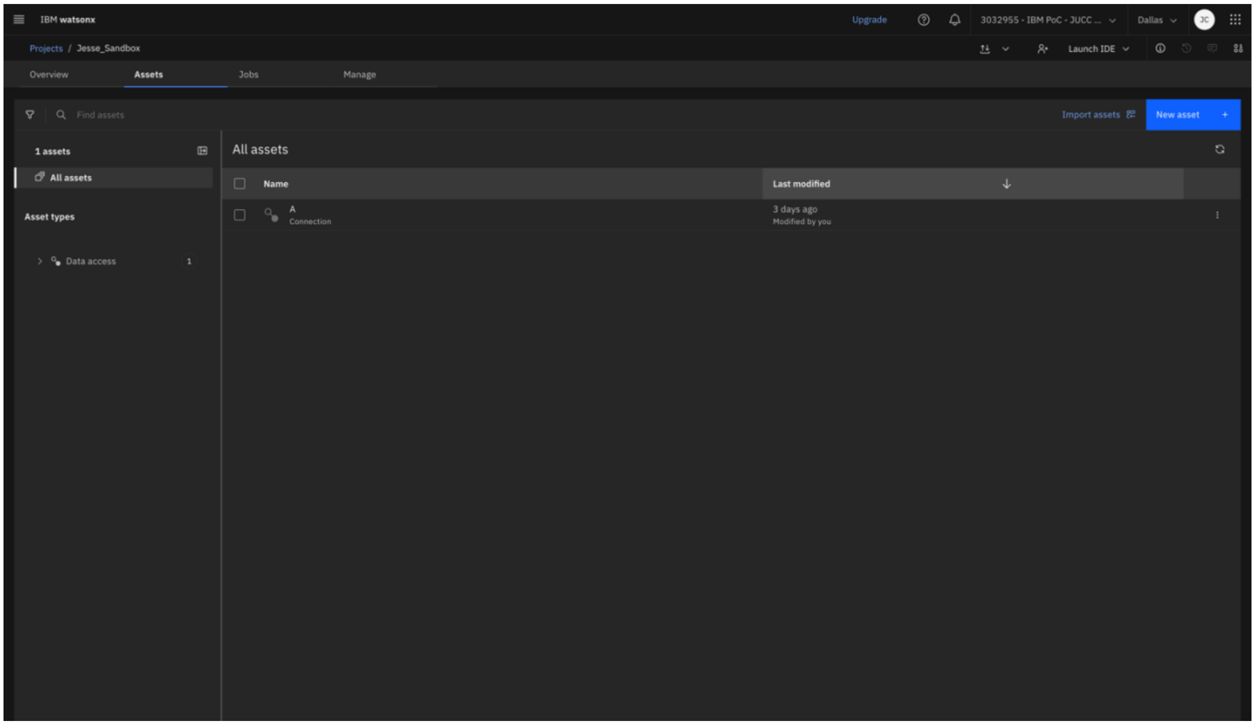
Sovereignty

Step 0.5: Import Data Asset

1. Navigate to the GitHub repo at

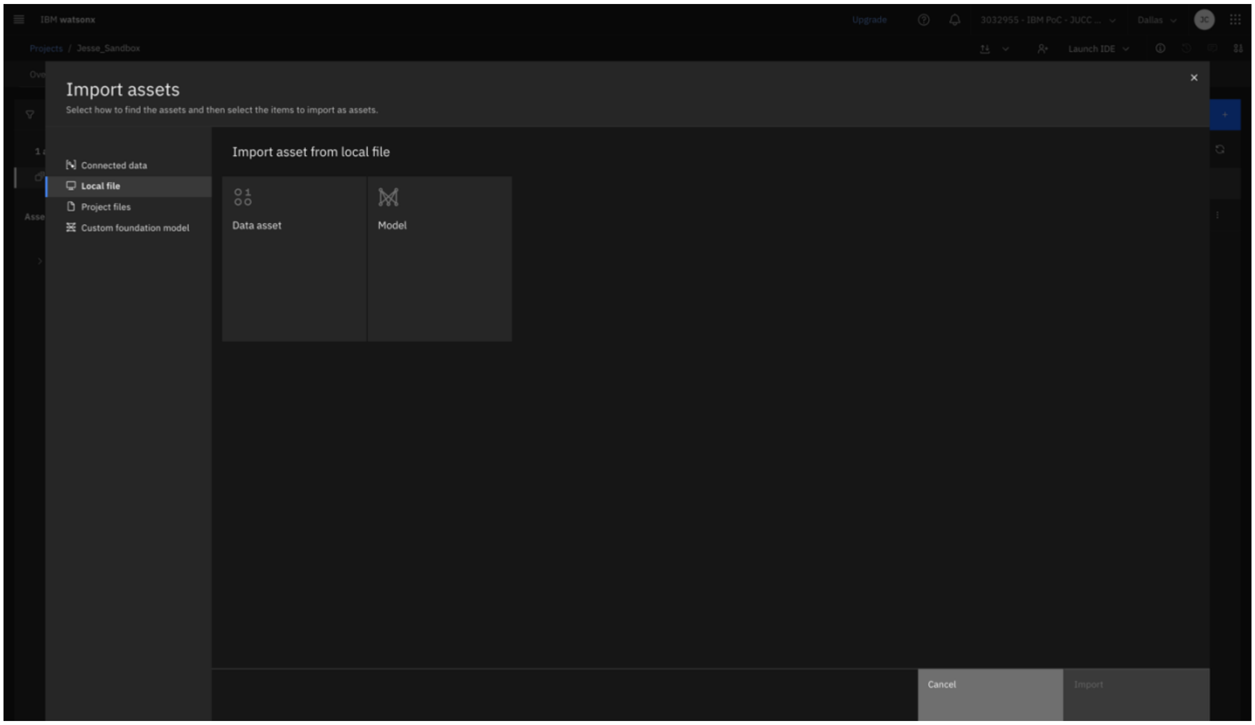
   https://github.com/JesseChan5171/JUCC-workshop#

2. Click Code → Download ZIP (or run git clone)
3. In the extracted directory, locate the “evaluation_dataset” folder.
4. In your project, click the **Assets** tab.
5. Click **Import assets** in the top-right.



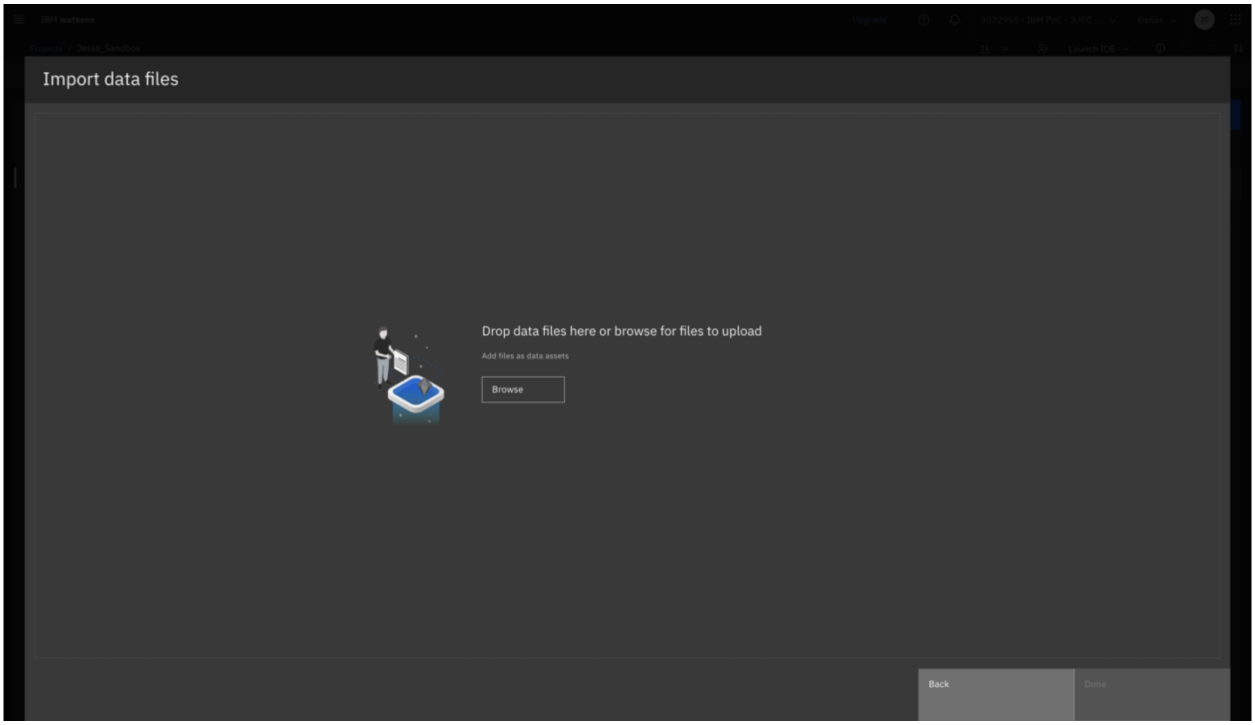
Name	Last modified
A Connection	3 days ago Modified by you

6. In the Import assets dialog, select **Local file** in the left menu.
7. Click the **Data asset** tile.



Select how to find the assets and then select the items to import as assets.

8. Click **Browse**, navigate to and select your dataset file (e.g. my_data.csv).
9. Click **Import** to upload the all data file from "**evaluation_dataset**" folder (Step **3**).

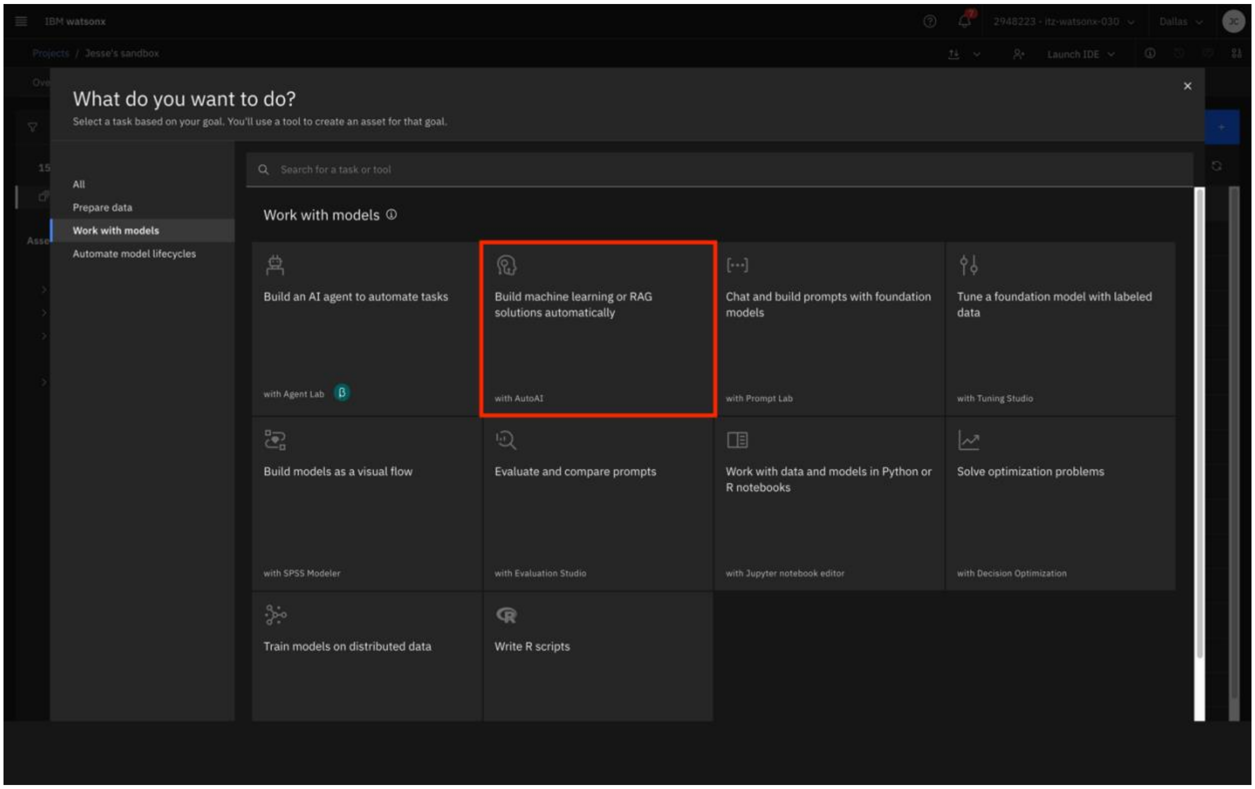


Drop data files here or browse for files to upload

Add files as data assets

Step 1: Select the Task

1. Navigate to the Assets Page inside projects in IBM watsonx.
2. Select **"Work with models"** from the list of categories.
3. Click on **"Build machine learning or RAG solutions automatically"** using the AutoAI tool.



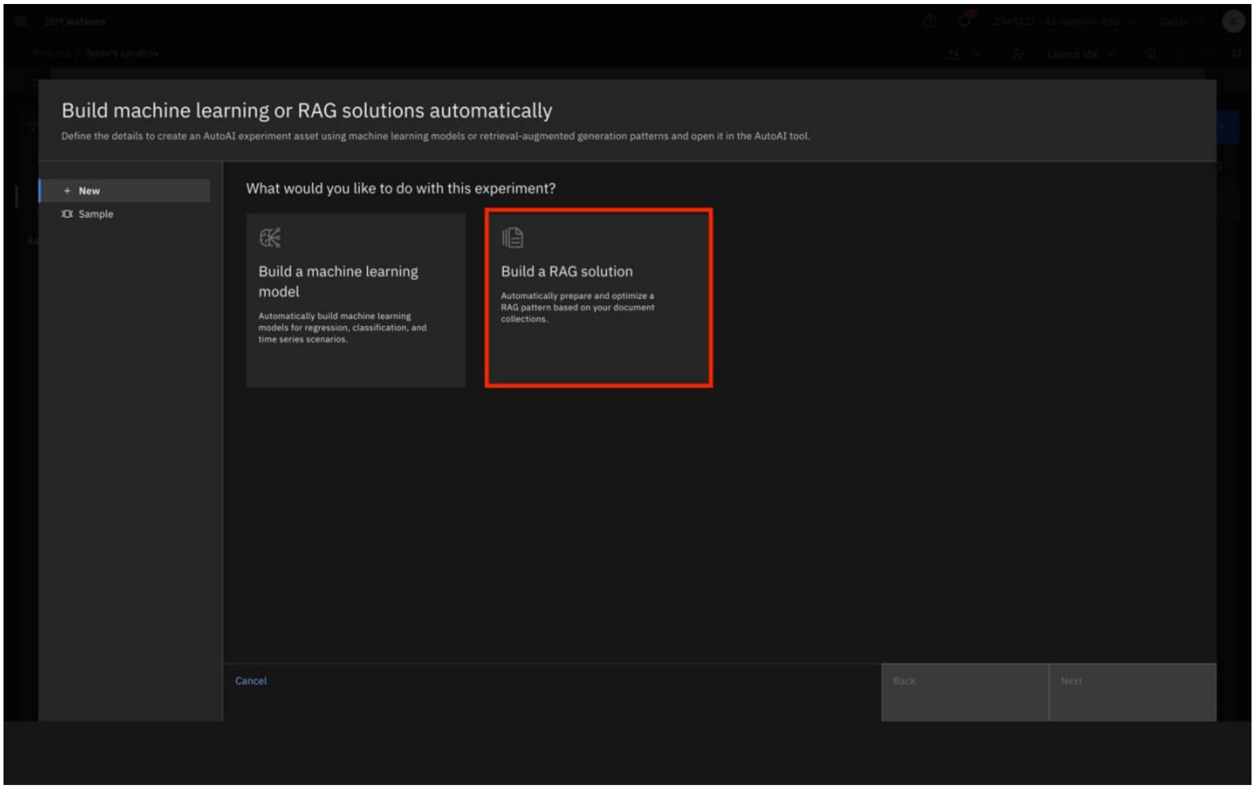
What do you want to do?

Select a task based on your goal. You'll use a tool to create an asset for that goal.

IBM watsonx
2948223 - itz-watsonx-030
Dallas
Projects / Jesse's sandbox
Launch IDE
Search for a task or tool
All
Prepare data
Work with models
Automate model lifecycles
Work with models
Build an AI agent to automate tasks
with Agent Lab
Build machine learning or RAG solutions automatically
with AutoAI
Chat and build prompts with foundation models
with Prompt Lab
Tune a foundation model with labeled data
with Tuning Studio
Build models as a visual flow
with SPSS Modeler
Evaluate and compare prompts
with Evaluation Studio
Work with data and models in Python or R notebooks
with Jupyter notebook editor
Solve optimization problems
with Decision Optimization
Train models on distributed data
Write R scripts

Step 2: Choose the Experiment Type

1. In the **AutoAI** experiment setup screen, choose what you want to build:
    - Select **"Build a RAG solution"** to prepare and optimize RAG patterns based on document collections.
2. Click **Next** to proceed.



Build machine learning or RAG solutions automatically

Define the details to create an AutoAI experiment asset using machine learning models or retrieval-augmented generation patterns and open it in the AutoAI tool.

What would you like to do with this experiment?

Build a machine learning model

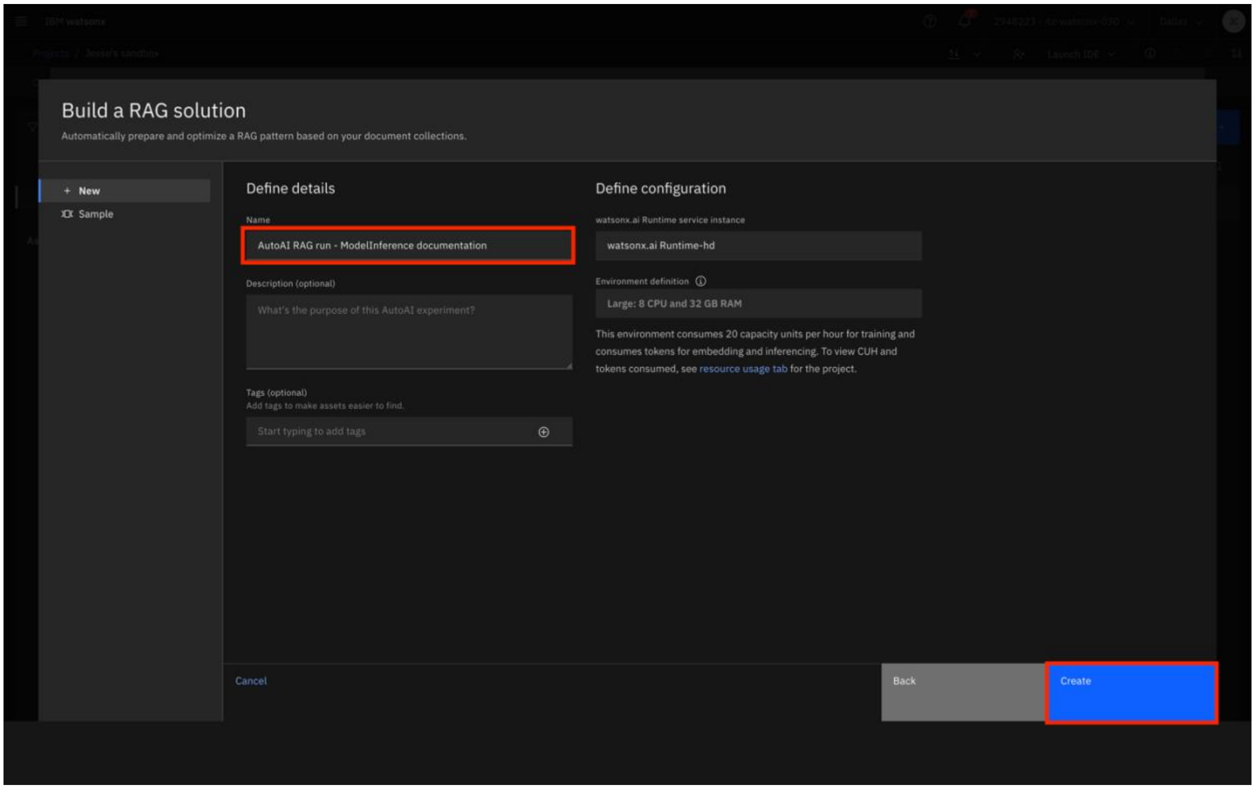
Automatically build machine learning models for regression, classification, and time series scenarios.

Build a RAG solution

Automatically prepare and optimize a RAG pattern based on your document collections.

Step 3: Define Experiment Details

1. Provide a name for your experiment (e.g., *AutoAI RAG run - ModelInference documentation*).
2. Optionally, add a description and tags for better organization and searchability.
3. In the configuration section:
   - Select the **watsonx.ai Runtime** service instance.
   - Choose the default environment definition
4. Click **Create** to proceed to the next step.



Build a RAG solution

Automatically prepare and optimize a RAG pattern based on your document collections.

Define details

Name: AutoAI RAG run - ModelInference documentation

Description (optional): What's the purpose of this AutoAI experiment?

Tags (optional)
Add tags to make assets easier to find.
Start typing to add tags

Define configuration

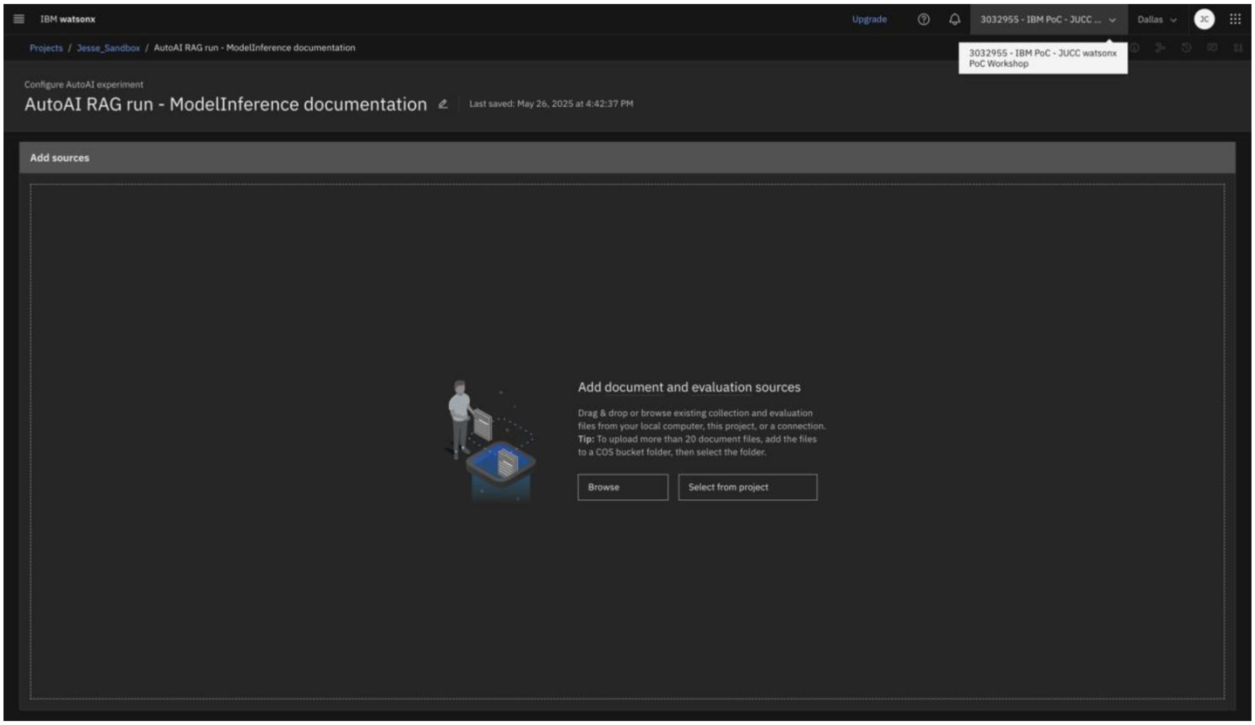
watsonx.ai Runtime service instance: watsonx.ai Runtime-hd

Environment definition: Large: 8 CPU and 32 GB RAM

This environment consumes 20 capacity units per hour for training and consumes tokens for embedding and inferencing. To view CUH and tokens consumed, see resource usage tab for the project.

Step 4: Add Document Sources

1. In the **Add sources** screen, upload or select the required documents for the RAG solution:
    - Click **Select from project**, choose **Data asset** in the left-hand Categories pane to upload files from your project.
2. Select the upload all files and click **Open** to upload it.

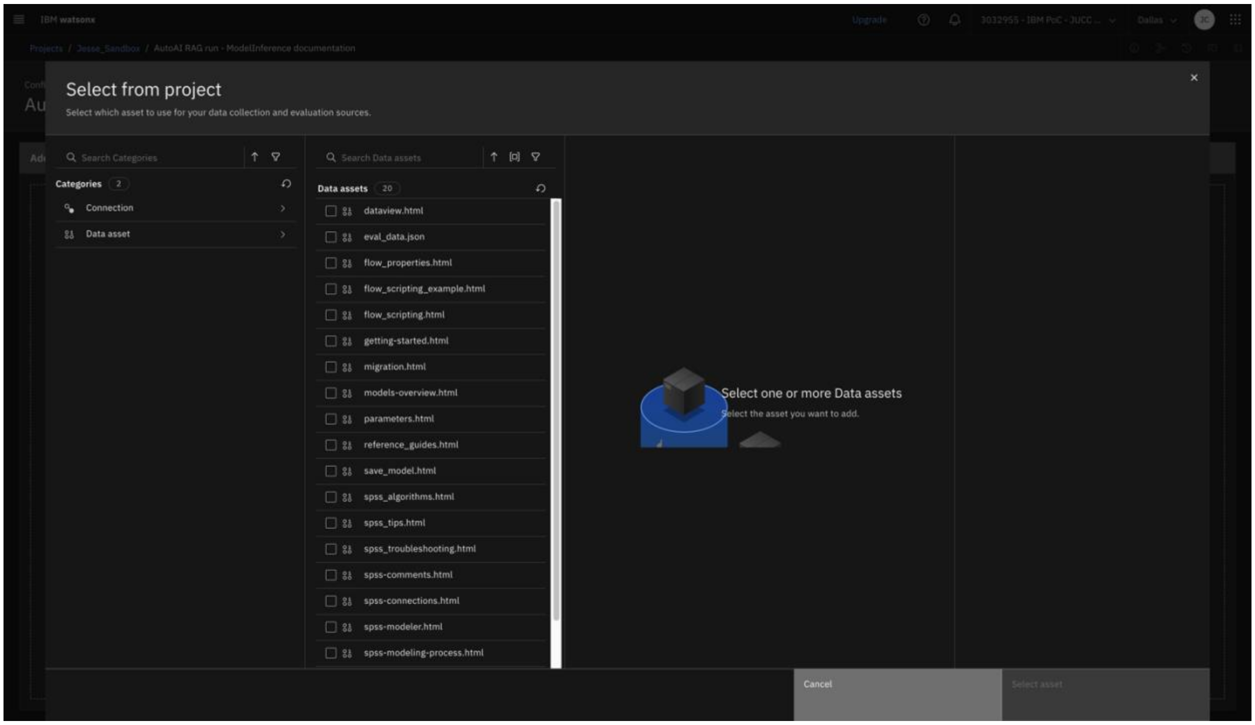

IBM watsonx
Upgrade
3032955 - IBM PoC - JUCC ...
Dallas
JC
Projects / Jesse_Sandbox / AutoAI RAG run - ModelInference documentation
3032955 - IBM PoC - JUCC watsonx PoC Workshop
Configure AutoAI experiment
Add sources
Browse
Select from project

AutoAI RAG run - ModelInference documentation

Last saved: May 26, 2025 at 4:42:37 PM

Add document and evaluation sources

Drag & drop or browse existing collection and evaluation files from your local computer, this project, or a connection. **Tip:** To upload more than 20 document files, add the files to a COS bucket folder, then select the folder.



IBM watsonx
Upgrade
3032955 - IBM PoC - JUCC ...
Dallas
JC
Projects / Jesse_Sandbox / AutoAI RAG run - ModelInference documentation

Select from project

Select which asset to use for your data collection and evaluation sources.

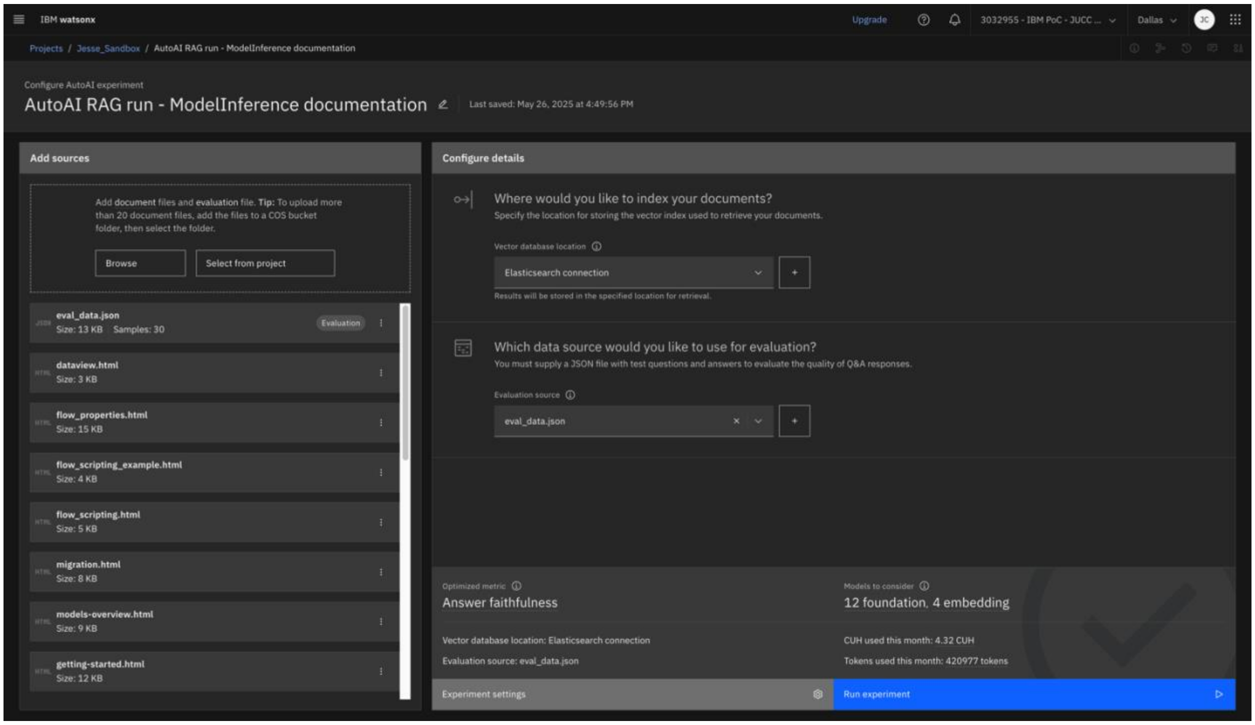
- □ dataview.html
- □ eval_data.json
- □ flow_properties.html
- □ flow_scripting_example.html
- □ flow_scripting.html
- □ getting-started.html
- □ migration.html
- □ models-overview.html
- □ parameters.html
- □ reference_guides.html
- □ save_model.html
- □ spss_algorithms.html
- □ spss_tips.html
- □ spss_troubleshooting.html
- □ spss-comments.html
- □ spss-connections.html
- □ spss-modeler.html
- □ spss-modeling-process.html

Select one or more Data assets

Select the asset you want to add.

Step 5: Configure Indexing and Evaluation Sources

1. In the **Configure details** page:
    - For indexing documents:
        - Choose the **‘Elasticsearch connection’** to store the vector index.
2. Once the source is uploaded, confirm its inclusion in the experiment.



IBM watsonx

Upgrade

3032955 - IBM PoC - JUCC ...

Dallas

Projects / Jesse_Sandbox / AutoAI RAG run - ModelInference documentation

Configure AutoAI experiment

AutoAI RAG run - ModelInference documentation

Last saved: May 26, 2025 at 4:49:56 PM

Add sources

Add document files and evaluation file. Tip: To upload more than 20 document files, add the files to a COS bucket folder, then select the folder.

Browse

Select from project

eval_data.json Size: 13 KB Samples: 30 Evaluation

dataview.html Size: 3 KB

flow_properties.html Size: 15 KB

flow_scripting_example.html Size: 4 KB

flow_scripting.html Size: 5 KB

migration.html Size: 8 KB

models-overview.html Size: 9 KB

getting-started.html Size: 12 KB

Configure details

Where would you like to index your documents?

Specify the location for storing the vector index used to retrieve your documents.

Vector database location: Elasticsearch connection

Results will be stored in the specified location for retrieval.

Which data source would you like to use for evaluation?

You must supply a JSON file with test questions and answers to evaluate the quality of Q&A responses.

Evaluation source: eval_data.json

Optimized metric: Answer faithfulness

Models to consider: 12 foundation, 4 embedding

Vector database location: Elasticsearch connection

Evaluation source: eval_data.json

CUH used this month: 4.32 CUH

Tokens used this month: 420977 tokens

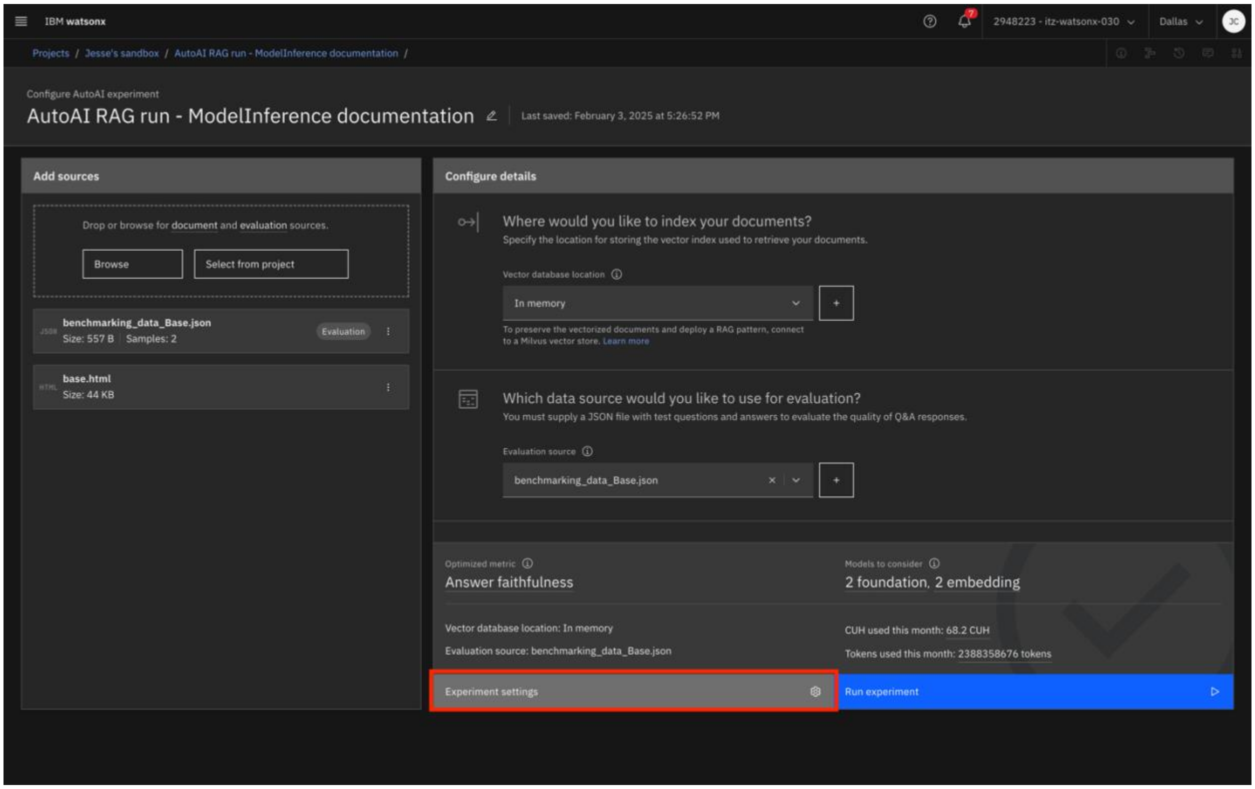
Experiment settings

Run experiment

Step 6: Configure Indexing and Experiment Settings

1. Ensure that the documents and evaluation sources are listed correctly in the **Configure details** page.

2. Click **Experiment settings**:



Configure AutoAI experiment

AutoAI RAG run - ModelInference documentation

Last saved: February 3, 2025 at 5:26:52 PM

Drop or browse for document and evaluation sources.

benchmarking_data_Base.json
Size: 557 B | Samples: 2

base.html
Size: 44 KB

Where would you like to index your documents?
Specify the location for storing the vector index used to retrieve your documents.

Vector database location: In memory

To preserve the vectorized documents and deploy a RAG pattern, connect to a Milvus vector store. Learn more

Which data source would you like to use for evaluation?
You must supply a JSON file with test questions and answers to evaluate the quality of Q&A responses.

Evaluation source: benchmarking_data_Base.json

Optimized metric: Answer faithfulness

Models to consider: 2 foundation, 2 embedding

Vector database location: In memory

Evaluation source: benchmarking_data_Base.json

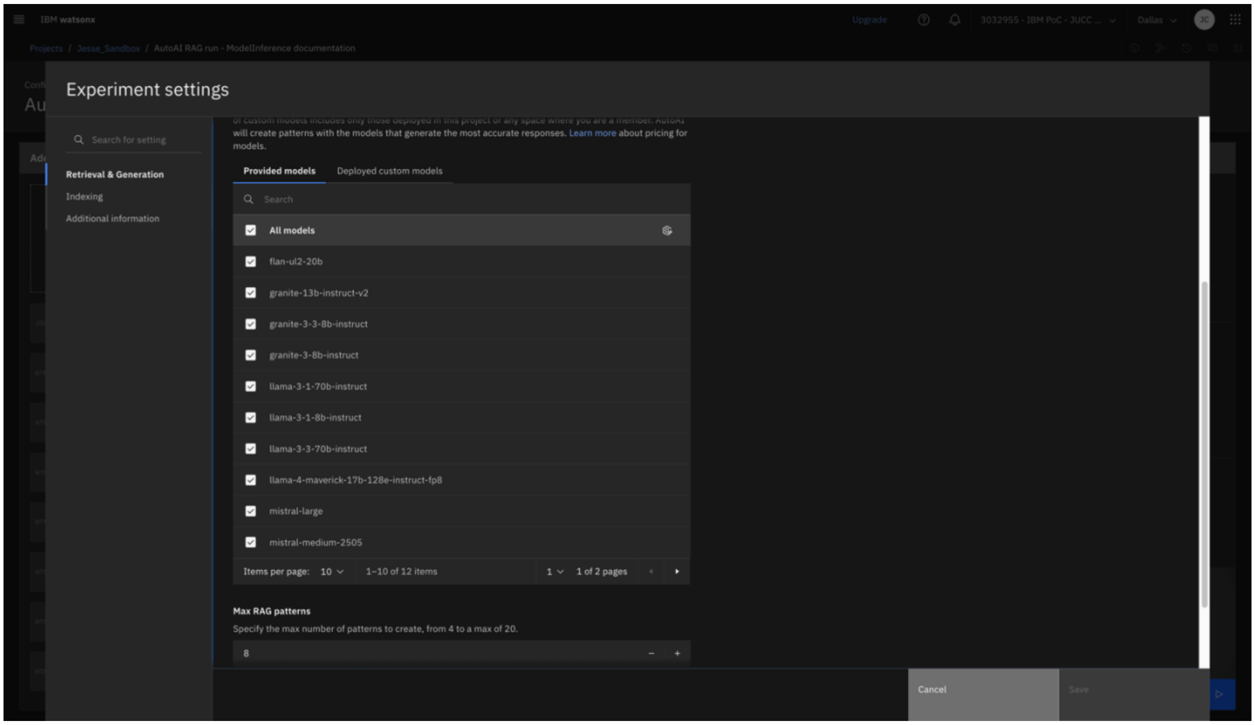
CUH used this month: 68.2 CUH

Tokens used this month: 2388358676 tokens

In **Experiment settings**:

- Review the **Optimized metric** (**Answer faithfulness**) to prioritize the quality of generated answers.
- Review the **Retrieval method** (**Window** or **Simple**) for data retrieval.
- Review the **Foundation models** (llama-2-70b-instruct, flan-ul2-20b) for experimentation.

2. Save the changes.



Experiment settings

of custom models includes only those deployed in this project or any space where you are a member. AutoAI will create patterns with the models that generate the most accurate responses. Learn more about pricing for models.

Provided models | Deployed custom models

Search

☑ All models
☑ flan-ul2-20b
☑ granite-13b-instruct-v2
☑ granite-3-3-8b-instruct
☑ granite-3-8b-instruct
☑ llama-3-1-70b-instruct
☑ llama-3-1-8b-instruct
☑ llama-3-3-70b-instruct
☑ llama-4-maverick-17b-128e-instruct-fp8
☑ mistral-large
☑ mistral-medium-2505

Items per page: 10 | 1–10 of 12 items | 1 | 1 of 2 pages

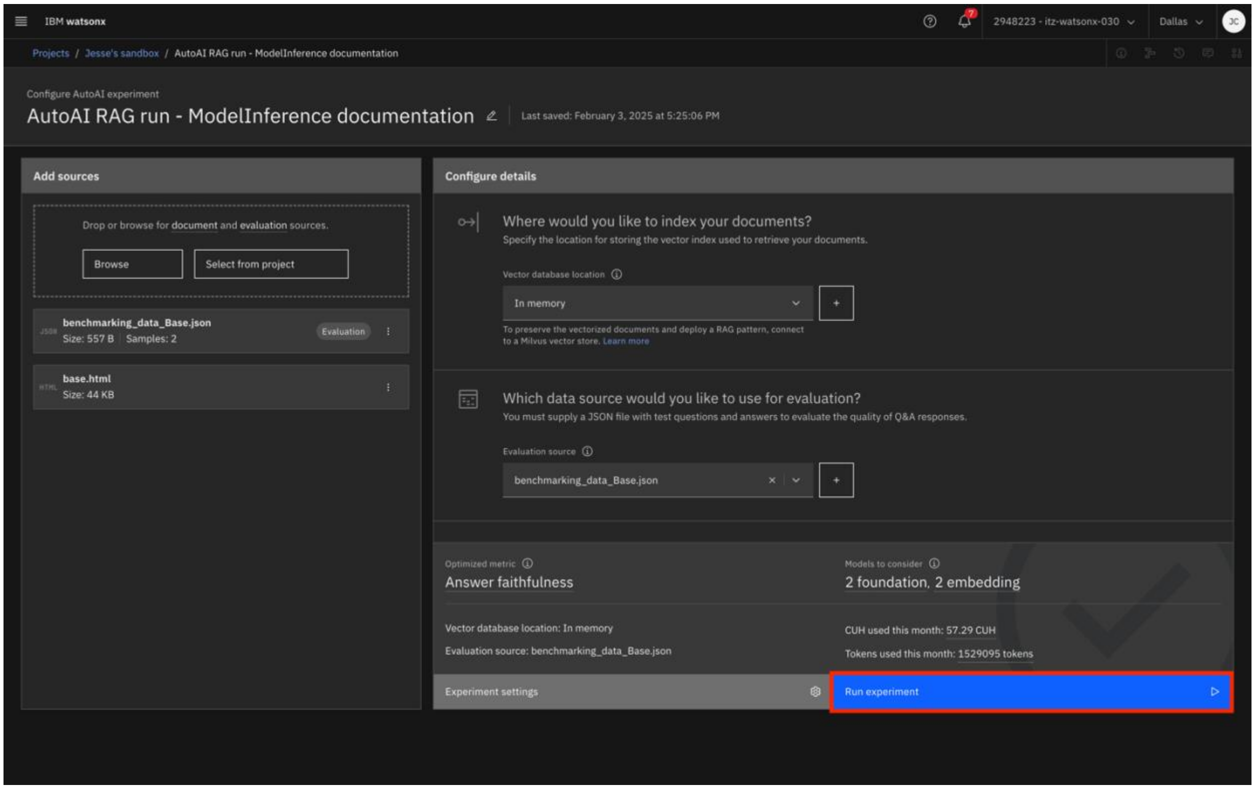
Max RAG patterns

Specify the max number of patterns to create, from 4 to a max of 20.

8

Step 7: Run the Experiment

1. Return to the **Configure details** page.
2. Verify all configurations and sources, then click **Run experiment**.
3. Monitor the execution through the **Progress map**.


IBM watsonx
2948223 - itz-watsonx-030
Dallas
JC
Projects / Jesse's sandbox / AutoAI RAG run - ModelInference documentation

Configure AutoAI experiment

AutoAI RAG run - ModelInference documentation

Last saved: February 3, 2025 at 5:25:06 PM

Add sources
Drop or browse for document and evaluation sources.
Browse
Select from project

benchmarking_data_Base.json
Size: 557 B | Samples: 2

base.html
Size: 44 KB

Where would you like to index your documents?
Specify the location for storing the vector index used to retrieve your documents.

Vector database location: In memory

To preserve the vectorized documents and deploy a RAG pattern, connect to a Milvus vector store. Learn more

Which data source would you like to use for evaluation?
You must supply a JSON file with test questions and answers to evaluate the quality of Q&A responses.

Evaluation source: benchmarking_data_Base.json

Optimized metric: Answer faithfulness

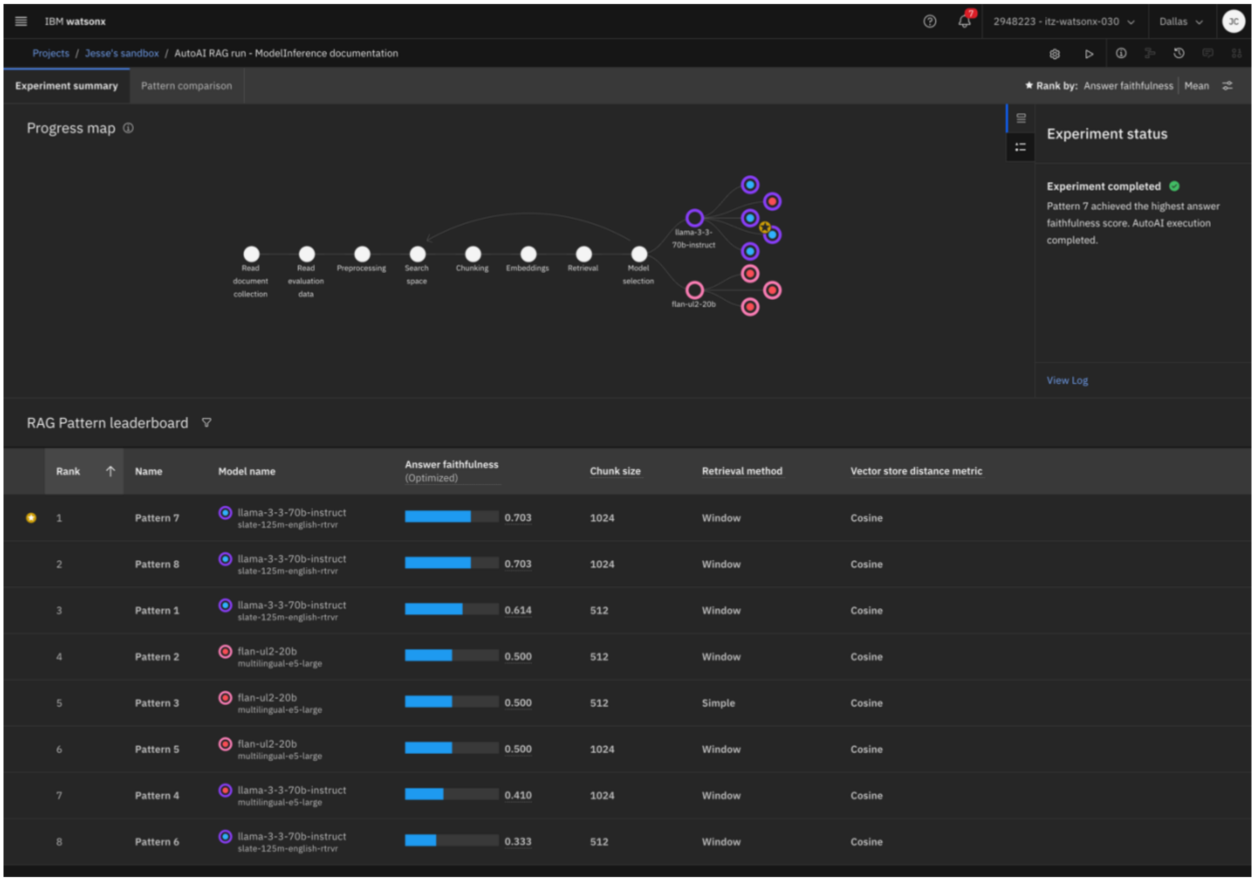
Models to consider: 2 foundation, 2 embedding

Vector database location: In memory
Evaluation source: benchmarking_data_Base.json

CUH used this month: 57.29 CUH
Tokens used this month: 1529095 tokens

Step 8: Review Results

1. Navigate to the **Experiment summary**:
   - View the **Progress map** to confirm the experiment stages were completed successfully.
   - Check the **RAG Pattern leaderboard** for ranked results based on performance metrics such as **Answer faithfulness** and **Chunk size**.
2. Identify the top-performing pattern, such as Pattern 7 with llama-2-70b-instruct.



Experiment status

Experiment completed

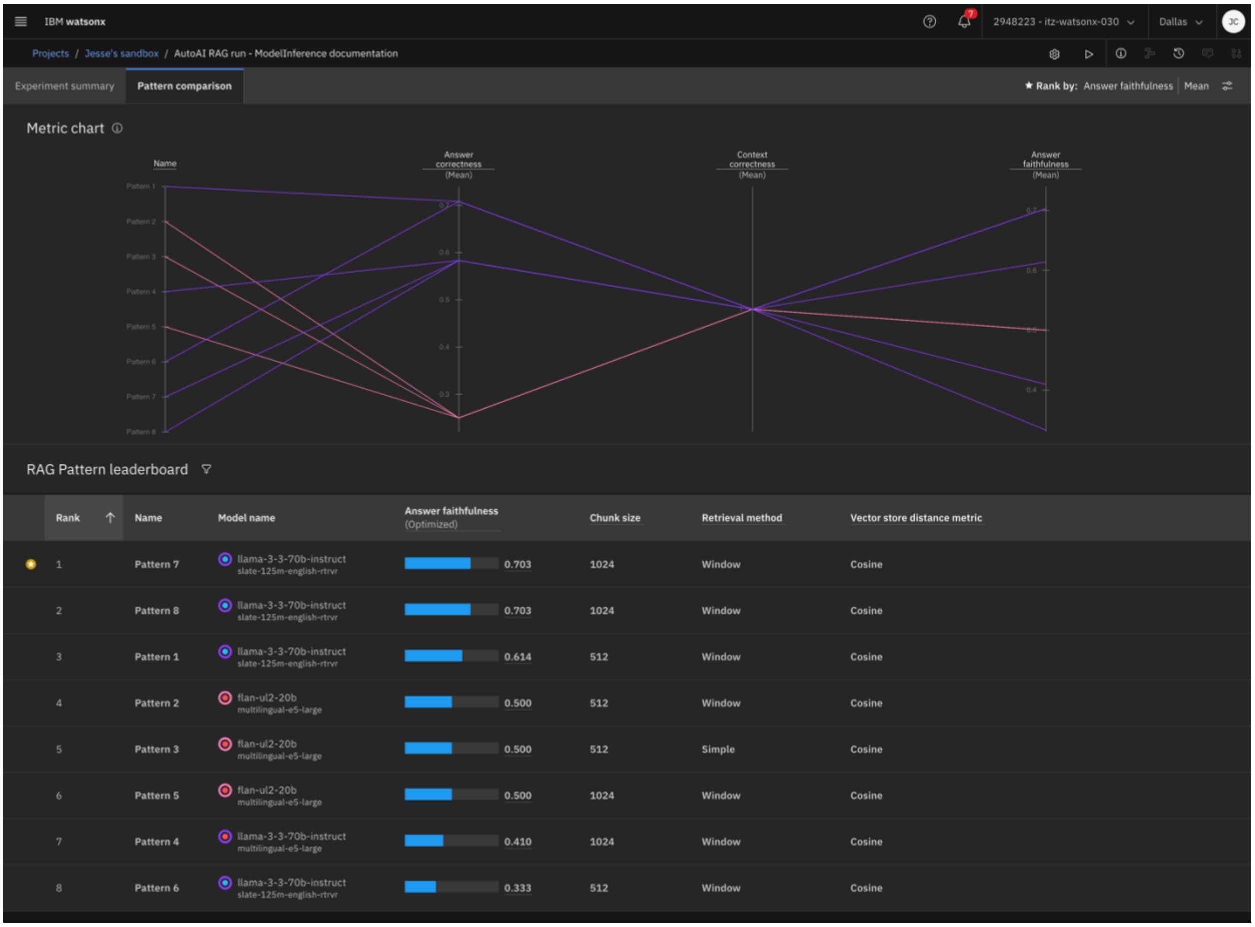
Pattern 7 achieved the highest answer faithfulness score. AutoAI execution completed.

RAG Pattern leaderboard

	Rank	Name	Model name	Answer faithfulness (Optimized)	Chunk size	Retrieval method	Vector store distance metric
★	1	Pattern 7	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine
	2	Pattern 8	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine
	3	Pattern 1	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.614	512	Window	Cosine
	4	Pattern 2	flan-ul2-20b multilingual-e5-large	0.500	512	Window	Cosine
	5	Pattern 3	flan-ul2-20b multilingual-e5-large	0.500	512	Simple	Cosine
	6	Pattern 5	flan-ul2-20b multilingual-e5-large	0.500	1024	Window	Cosine
	7	Pattern 4	llama-3-3-70b-instruct multilingual-e5-large	0.410	1024	Window	Cosine
	8	Pattern 6	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.333	512	Window	Cosine

Step 9: Pattern Comparison

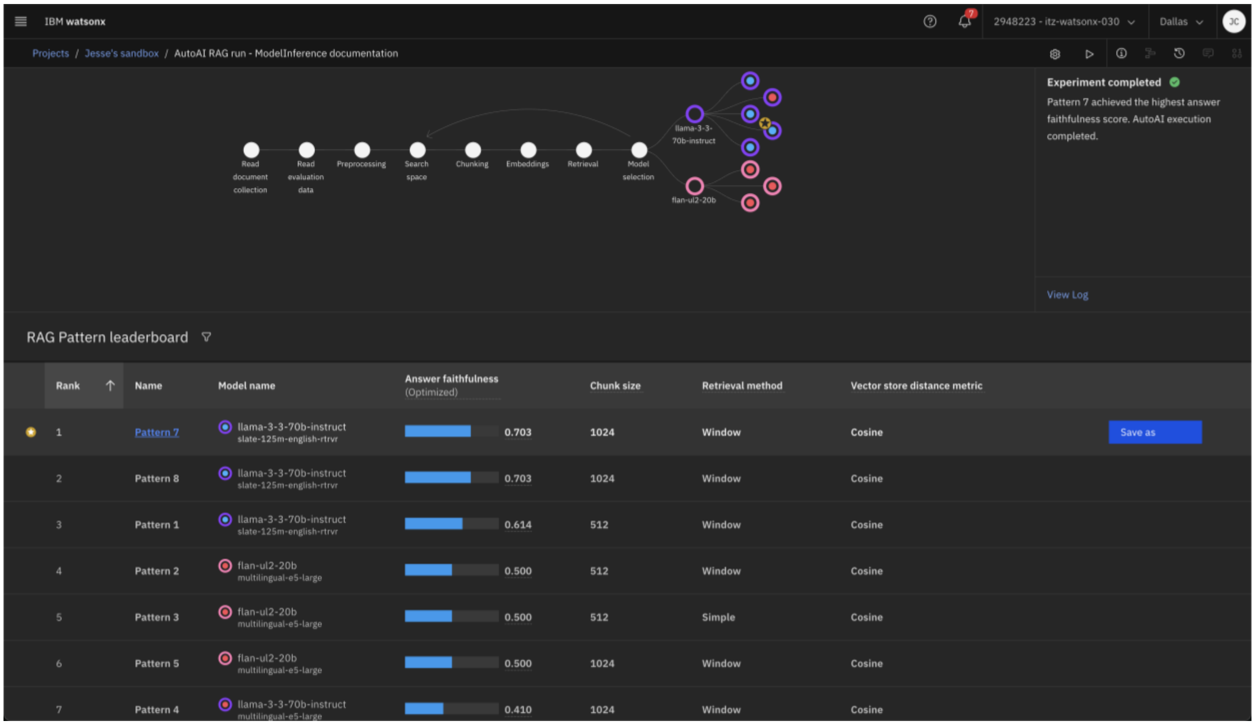
1. Switch to the **Pattern comparison** tab for detailed insights:
    - Use the **Metric chart** to compare patterns based on metrics like **Answer faithfulness** and **Correctness**.
    - Analyze the trends to validate the optimal configuration for deployment.


IBM watsonx
7
2948223 - itz-watsonx-030
Dallas
JC
Projects / Jesse's sandbox / AutoAI RAG run - ModelInference documentation
Experiment summary
Pattern comparison
★ Rank by: Answer faithfulness | Mean
Metric chart
Name
Answer correctness (Mean)
Context correctness (Mean)
Answer faithfulness (Mean)
Pattern 1
Pattern 2
Pattern 3
Pattern 4
Pattern 5
Pattern 6
Pattern 7
Pattern 8
0.7
0.6
0.5
0.4
0.3
0.7
0.6
0.5
0.4
RAG Pattern leaderboard

	Rank	Name	Model name	Answer faithfulness (Optimized)	Chunk size	Retrieval method	Vector store distance metric
★	1	Pattern 7	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine
	2	Pattern 8	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine
	3	Pattern 1	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.614	512	Window	Cosine
	4	Pattern 2	flan-ul2-20b multilingual-e5-large	0.500	512	Window	Cosine
	5	Pattern 3	flan-ul2-20b multilingual-e5-large	0.500	512	Simple	Cosine
	6	Pattern 5	flan-ul2-20b multilingual-e5-large	0.500	1024	Window	Cosine
	7	Pattern 4	llama-3-3-70b-instruct multilingual-e5-large	0.410	1024	Window	Cosine
	8	Pattern 6	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.333	512	Window	Cosine

Step 10: Save the Results as a Notebook

1. After the experiment is completed:
   - Review the **RAG Pattern leaderboard** to confirm the top-performing pattern.
2. Click on the **Save as** button to generate a detailed notebook of the experiment results.



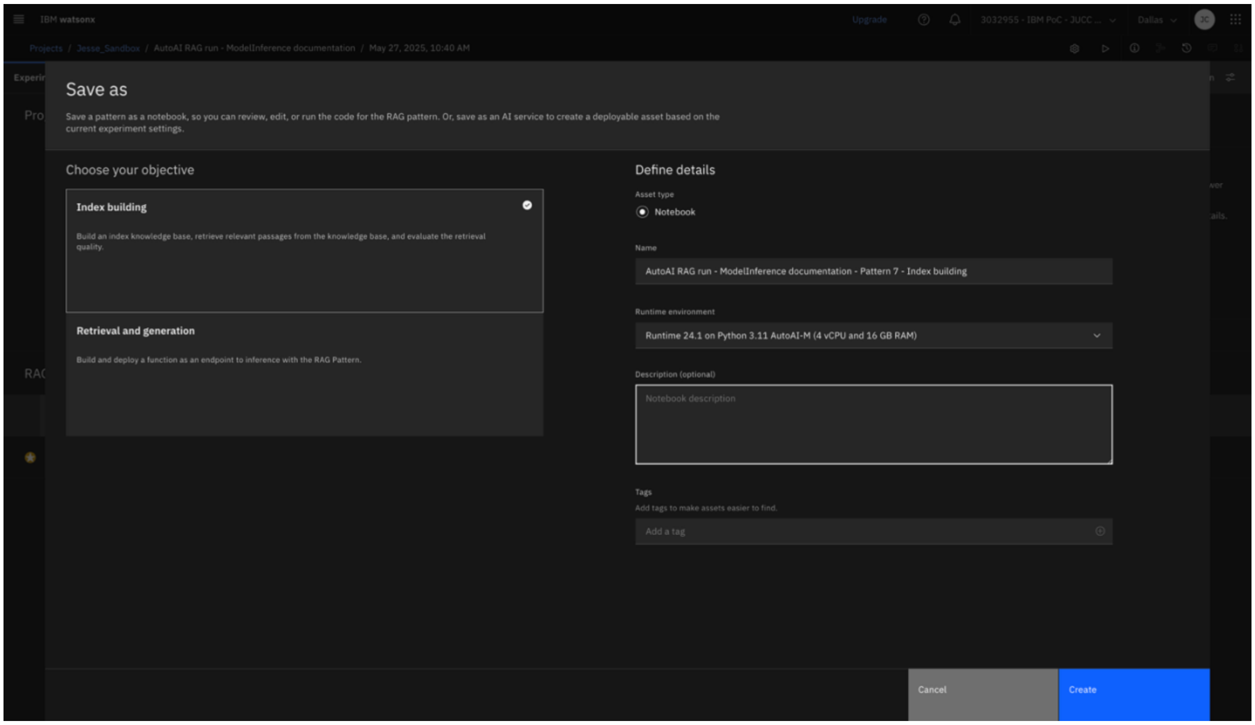
Experiment completed

Pattern 7 achieved the highest answer faithfulness score. AutoAI execution completed.

RAG Pattern leaderboard

	Rank	Name	Model name	Answer faithfulness (Optimized)	Chunk size	Retrieval method	Vector store distance metric	
★	1	Pattern 7	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine	Save as
	2	Pattern 8	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.703	1024	Window	Cosine	
	3	Pattern 1	llama-3-3-70b-instruct slate-125m-english-rtrvr	0.614	512	Window	Cosine	
	4	Pattern 2	flan-ul2-20b multilingual-e5-large	0.500	512	Window	Cosine	
	5	Pattern 3	flan-ul2-20b multilingual-e5-large	0.500	512	Simple	Cosine	
	6	Pattern 5	flan-ul2-20b multilingual-e5-large	0.500	1024	Window	Cosine	
	7	Pattern 4	llama-3-3-70b-instruct multilingual-e5-large	0.410	1024	Window	Cosine	

3. In the **Save notebook** dialog:
   - Under Choose your objective, select **Index building** (and Retrieval and generation if you want both).
   - Provide a name for the notebook (e.g., *AutoAI RAG - Top-performing Pattern 7*).
   - Choose the runtime environment and add an optional description or tags for easier identification.
4. Click **Create** to save the notebook for further use or sharing.



IBM watsonx
Upgrade
3032955 - IBM PoC - JUCC ...
Dallas
Projects / Jesse_Sandbox / AutoAI RAG run - ModelInference documentation / May 27, 2025, 10:40 AM

Save as

Save a pattern as a notebook, so you can review, edit, or run the code for the RAG pattern. Or, save as an AI service to create a deployable asset based on the current experiment settings.

Choose your objective

Index building ✓

Build an index knowledge base, retrieve relevant passages from the knowledge base, and evaluate the retrieval quality.

Retrieval and generation

Build and deploy a function as an endpoint to inference with the RAG Pattern.

Define details

Asset type

● Notebook

Name: AutoAI RAG run - ModelInference documentation - Pattern 7 - Index building

Runtime environment: Runtime 24.1 on Python 3.11 AutoAI-M (4 vCPU and 16 GB RAM)

Description (optional): Notebook description

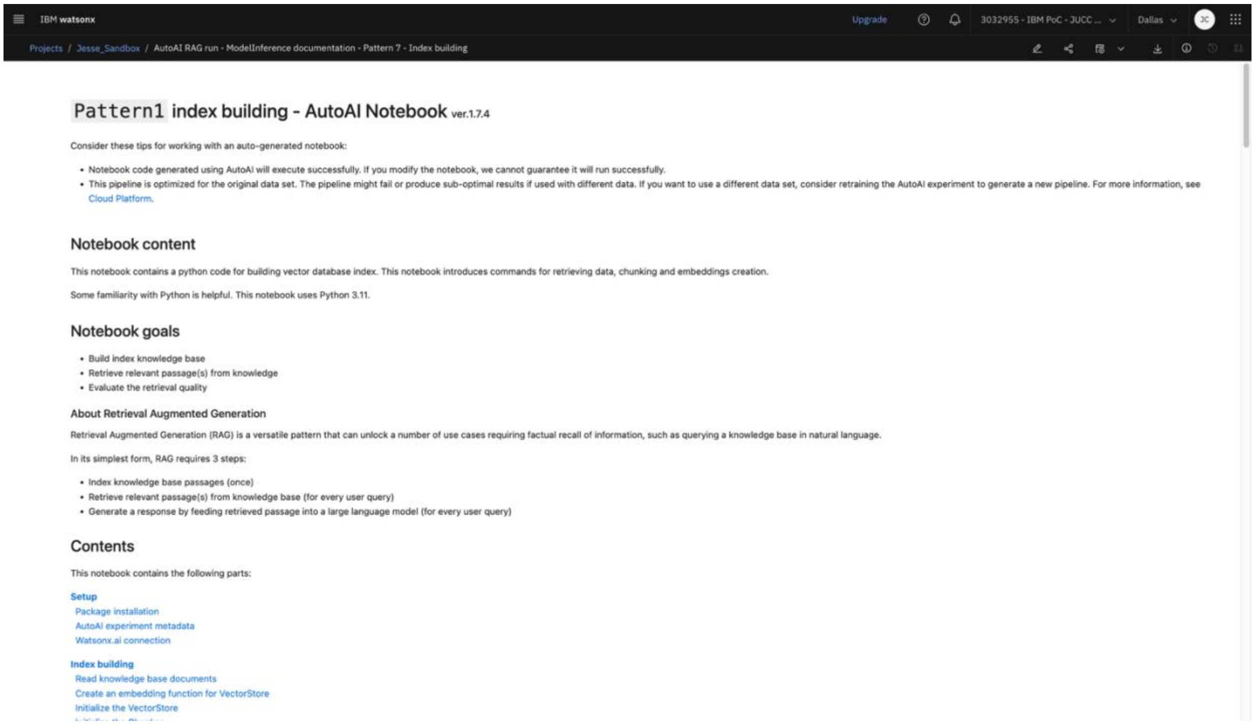
Tags

Add tags to make assets easier to find.

Add a tag

Step 11: Open and Explore the Notebook

1. Once the notebook is created:
   - ○ Open the notebook to explore the detailed steps, data set, configurations, and metrics used in the experiment.


IBM watsonx
Upgrade
3032955 - IBM PoC - JUCC ...
Dallas
JC
Projects / Jesse_Sandbox / AutoAI RAG run - ModelInference documentation - Pattern 7 - Index building

`Pattern1` index building - AutoAI Notebook ver.1.7.4

Consider these tips for working with an auto-generated notebook:

- Notebook code generated using AutoAI will execute successfully. If you modify the notebook, we cannot guarantee it will run successfully.
- This pipeline is optimized for the original data set. The pipeline might fail or produce sub-optimal results if used with different data. If you want to use a different data set, consider retraining the AutoAI experiment to generate a new pipeline. For more information, see Cloud Platform.

Notebook content

This notebook contains a python code for building vector database index. This notebook introduces commands for retrieving data, chunking and embeddings creation.

Some familiarity with Python is helpful. This notebook uses Python 3.11.

Notebook goals

- Build index knowledge base
- Retrieve relevant passage(s) from knowledge
- Evaluate the retrieval quality

About Retrieval Augmented Generation

Retrieval Augmented Generation (RAG) is a versatile pattern that can unlock a number of use cases requiring factual recall of information, such as querying a knowledge base in natural language.

In its simplest form, RAG requires 3 steps:

- Index knowledge base passages (once)
- Retrieve relevant passage(s) from knowledge base (for every user query)
- Generate a response by feeding retrieved passage into a large language model (for every user query)

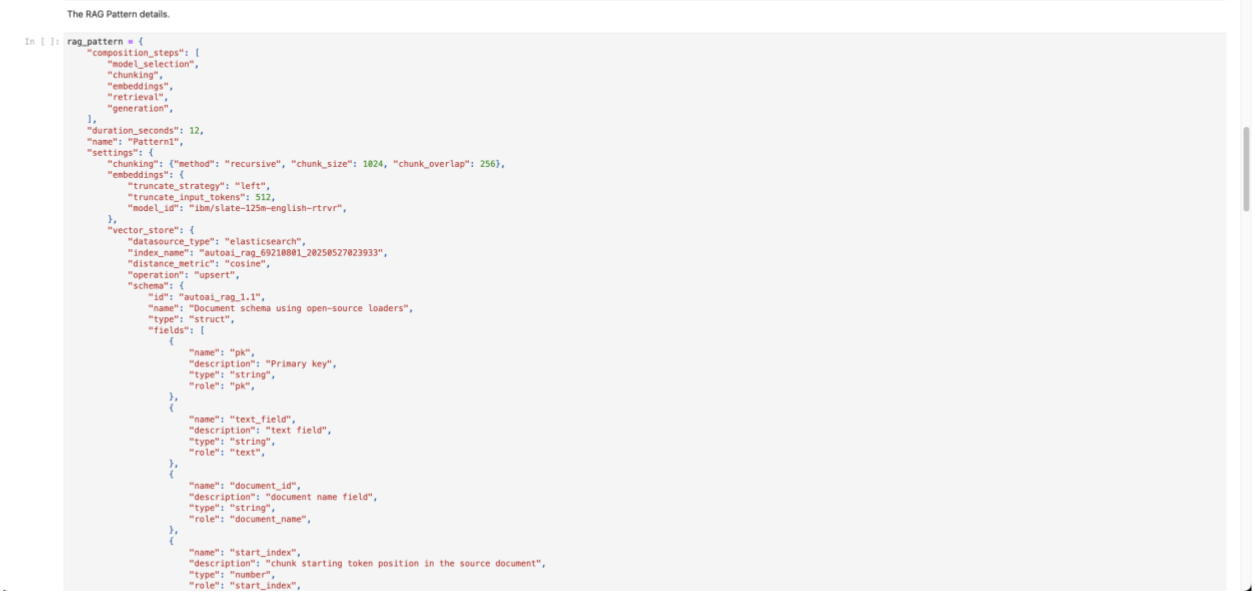
Contents

This notebook contains the following parts:

2. Run the notebook with following:
   - ○ **Run the notebook cells**, check Instructions for setting up the vector store and index building.
   - ○ **Watsonx.ai API key** for connection **(step 0)**
   - ○ Details on RAG retrieval and generation processes.
   - ○ Code snippets to reproduce the experiment and generate results.
3. Use the notebook to further refine or replicate the experiment as needed.

Step 12: Extract RAG Pattern Configuration Keys

1. Open the code or notebook cell containing your rag_pattern = { … } definition.



The RAG Pattern details.

```
In [ ]: rag_pattern = {
            "composition_steps": [
                "model_selection",
                "chunking",
                "embeddings",
                "retrieval",
                "generation",
            ],
            "duration_seconds": 12,
            "name": "Pattern1",
            "settings": {
                "chunking": {"method": "recursive", "chunk_size": 1024, "chunk_overlap": 256},
                "embeddings": {
                    "truncate_strategy": "left",
                    "truncate_input_tokens": 512,
                    "model_id": "ibm/slate-125m-english-rtrvr",
                },
                "vector_store": {
                    "datasource_type": "elasticsearch",
                    "index_name": "autoai_rag_69210801_20250527023933",
                    "distance_metric": "cosine",
                    "operation": "upsert",
                    "schema": {
                        "id": "autoai_rag_1.1",
                        "name": "Document schema using open-source loaders",
                        "type": "struct",
                        "fields": [
                            {
                                "name": "pk",
                                "description": "Primary key",
                                "type": "string",
                                "role": "pk",
                            },
                            {
                                "name": "text_field",
                                "description": "text field",
                                "type": "string",
                                "role": "text",
                            },
                            {
                                "name": "document_id",
                                "description": "document name field",
                                "type": "string",
                                "role": "document_name",
                            },
                            {
                                "name": "start_index",
                                "description": "chunk starting token position in the source document",
                                "type": "number",
                                "role": "start_index",
```

2. Locate the top-level keys and the nested settings that drive each module.
3. Copy out the following key names for use in your module in later lab:

IBM watsonx.ai Embeddings Key

	Where to get it?	Value
watsonx API Endpoint	->	https://us-south.ml.cloud.ibm.com
API Key	Step 0	(get your own config)
Project ID	Find 'experiment_metadata'	(get your own config) e.g. cd2dadd2-68e5-4680-8d79-cf8db08cc8c8
Embedding Model Name	rag_pattern: embeddings -> model_id	(get your own config) e.g. intfloat/multilingual-e5-large

IBM watsonx.ai Key

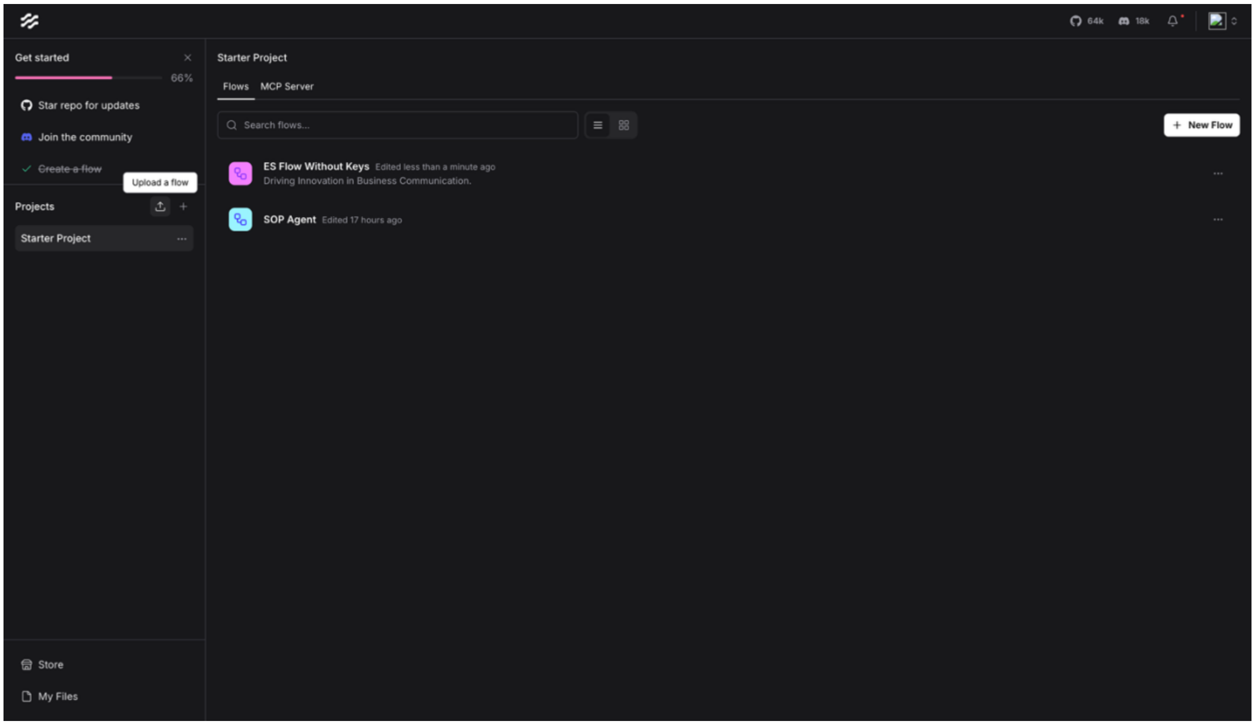
	Where to get it?	Value
watsonx API Endpoint	->	https://us-south.ml.cloud.ibm.com
Model Name	rag_pattern: generation -> model_id	(get your own config) e.g. meta-llama/llama-3-3-70b-instruct

Elasticsearch Vector Store

	Where to get it?	Value
Elasticsearch URL	->	https://us-south.ml.cloud.ibm.com
Index Name	rag_pattern: vector_store -> index_name	(get your own config) e.g. autoai_rag_69210801_20250527023933
Username	Step 0 -> sub-step 6	(get your own config) e.g. ibm_cloud_c06543e4_08e4_42d9_836e_9de74a8cff8d
Password	Step 0 -> sub-step 6	(get your own config) e.g. eGq3GJdEh9kHhQHfetUu3gOE6hwGIP7l

Step 13: Upload the ES Flow into Lang flow

1. In your browser, navigate to the Langflow UI: http://158.176.6.193:7860/
2. In the left-hand sidebar, click Upload a flow.
3. browse to the local JUCC-Workshop folder you downloaded earlier, select **ES Flow Without Keys.json**, and click Open.
4. Langflow will import the flow and display it on the canvas for you to review and configure

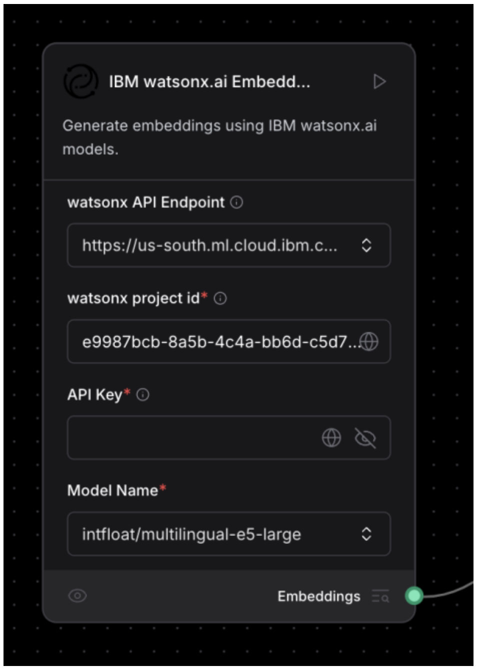


ES Flow Without Keys Edited less than a minute ago
Driving Innovation in Business Communication.

SOP Agent Edited 17 hours ago

Step 14: Configure Each Langflow Component and Understand Its Role

1. **watsonx.ai Embeddings**
   Role: Transforms text (documents or queries) into vector representations that capture semantic meaning, enabling similarity-based search in later steps.
   - watsonx API Endpoint: https://us-south.ml.cloud.ibm.com
   - watsonx project id: your project ID
   - API Key: your API key
   - Model Name: your embeddings model (e.g. intfloat/multilingual-e5-large)



IBM watsonx.ai Embedd...

Generate embeddings using IBM watsonx.ai models.

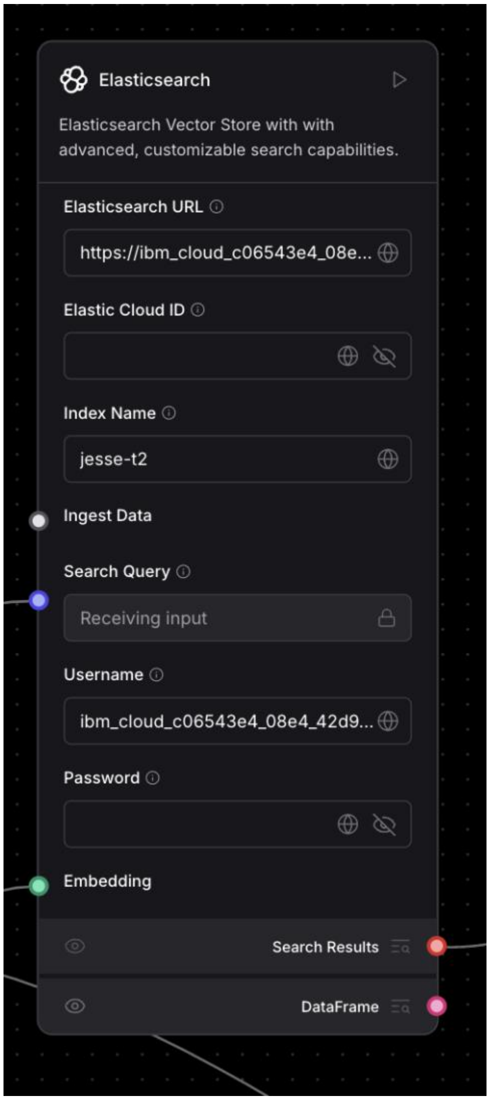
watsonx API Endpoint: https://us-south.ml.cloud.ibm.c...

watsonx project id*: e9987bcb-8a5b-4c4a-bb6d-c5d7...

API Key*:

Model Name*: intfloat/multilingual-e5-large

2. Elasticsearch Vector Store
   Role: Stores and indexes those embeddings; given a query vector, it retrieves the top most similar document chunks to serve as context.
   - Elasticsearch URL: your Elasticsearch URL
   - Index Name: your index name (e.g. autoai_rag_69210801_20250527023933)
   - Username: your Elasticsearch username
   - Password: your Elasticsearch password



Elasticsearch

Elasticsearch Vector Store with with advanced, customizable search capabilities.

Elasticsearch URL: https://ibm_cloud_c06543e4_08e...

Elastic Cloud ID:

Index Name: jesse-t2

Ingest Data

Search Query: Receiving input

Username: ibm_cloud_c06543e4_08e4_42d9...

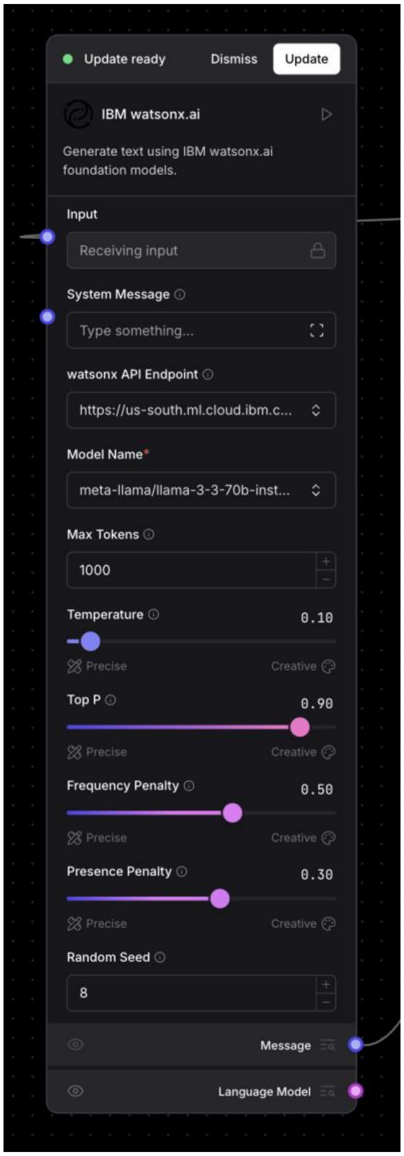
Password:

Embedding

Search Results

DataFrame

3. IBM watsonx.ai (Generation)
   Role: Consumes the retrieved chunks plus the user's query to generate a coherent, context-grounded answer using a foundation language model.
   - watsonx API Endpoint: https://us-south.ml.cloud.ibm.com
   - Model Name: your generation model (e.g. meta-llama/llama-3-3-70b-instruct)



Update ready

IBM watsonx.ai

Generate text using IBM watsonx.ai foundation models.

Input: Receiving input

System Message: Type something...

watsonx API Endpoint: https://us-south.ml.cloud.ibm.c...

Model Name*: meta-llama/llama-3-3-70b-inst...

Max Tokens: 1000

Temperature: 0.10

Top P: 0.90

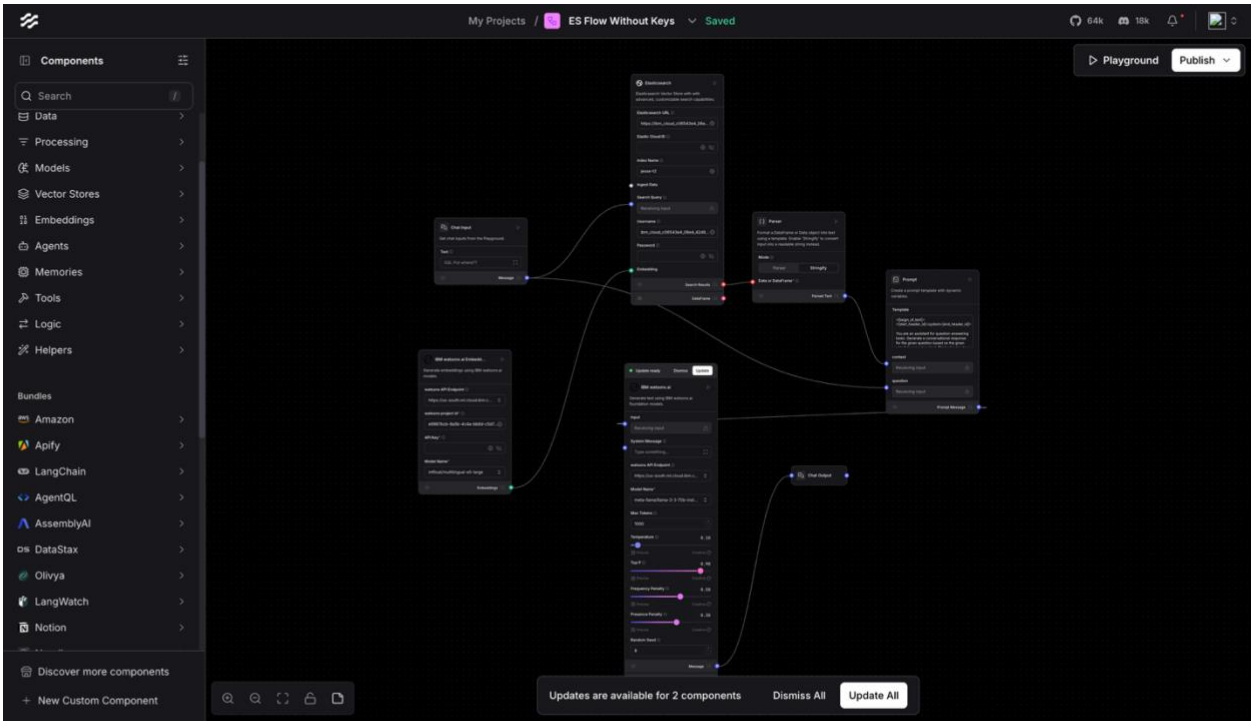
Frequency Penalty: 0.50

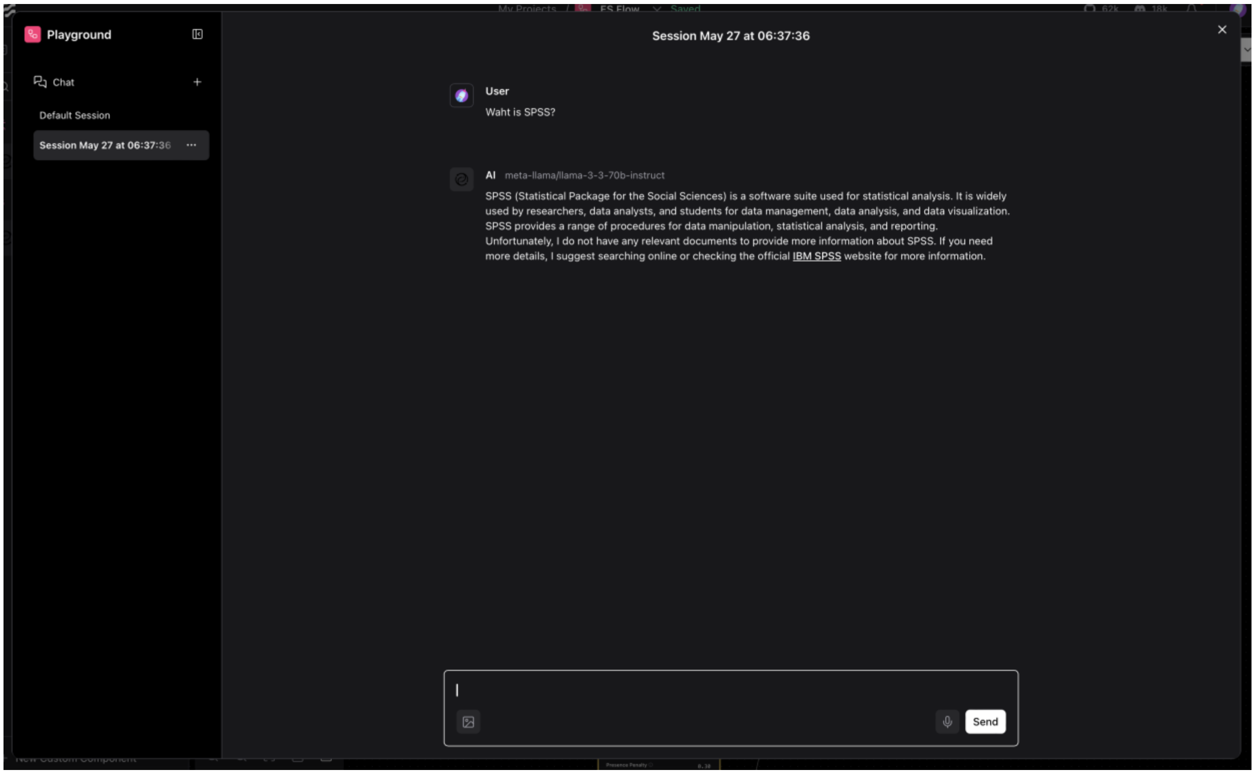
Presence Penalty: 0.30

Random Seed: 8

Step 15: Test Your RAG Flow in the Playground

1. In the top-right corner of Langflow, click **Playground**.
2. In the “New chat” prompt that appears, type your test question (for example, “What is SPSS?”).
3. Press **Enter** or click **Send**.
4. Observe how the flow:
   - Converts your question into an embedding,
   - Retrieves the most relevant document chunks from Elasticsearch,
   - Feeds those chunks + your question into the generation model,
   - Returns a grounded answer in the chat pane.
5. Review the **Session** sidebar to inspect earlier queries and responses or start a fresh chat by clicking the + next to **Chat**.





Session May 27 at 06:37:36

User

Waht is SPSS?

AI meta-llama/llama-3-3-70b-instruct

SPSS (Statistical Package for the Social Sciences) is a software suite used for statistical analysis. It is widely used by researchers, data analysts, and students for data management, data analysis, and data visualization. SPSS provides a range of procedures for data manipulation, statistical analysis, and reporting.
Unfortunately, I do not have any relevant documents to provide more information about SPSS. If you need more details, I suggest searching online or checking the official IBM SPSS website for more information.