7. LLM-powered Conversational Search

In this lesson

Learn about IBM watsonx Assistant's large language model (LLM)-powered Conversational Search: what it is, how it works, how to set it up, and how to use it.

Prerequisites

This section does not have to be done sequentially with the other lab sections. You may either use the watsonx Assistant instance from the previous sections of this lab or create a new one. This lesson will provision Assistant and Discovery services from the IBM Technology Zone (TechZone), so you don't have to start a Watson Discovery trial.

Table of Contents

Note: Be sure you are working with the latest version of this guide. Lab guidance will change as the integration between watsonx, watsonx Assistant, and LLMs evolve and IBM releases new versions of its purpose-built LLMs. The lab expands on this subject in greater detail.

What is Conversational Search?

Virtual assistants across every industry, geography, and company of all sizes are almost always built to answer a wide range of frequently asked questions (FAQs). In watsonx Assistant, builders have historically built anywhere from dozens to hundreds of Actions to answer FAQs.

FAQs typically require maintenance. Over time, answers change as an organization's processes, products, or services evolve. Builders must periodically manually review the Actions they have built to ensure their answers are accurate and up to date.

Actions that answer FAQs can also be challenging to build because end users can ask questions in many ways. For example:

- Tell me how to freeze my credit card
- Aren't I allowed to freeze credit cards?
- How do I freeze my credit card?
- Can you walk me through the process of freezing a credit card

One of these examples is an imperative command: "Tell me." One of these is a direct informational question: "How do I…?" One of these is a negative yes-or-no question: "Aren't I…?" As a builder, you might struggle to write an answer that addresses all these potential questions all at once without sounding robotic or unintelligent.

Conversational Search helps builders answer FAQs more easily by using generative AI (specifically, an LLM) to provide answers. To ensure that generative AI creates accurate, relevant, and up to date answers to FAQs, Conversational Search first *searches* for relevant information in the company's knowledge base and then *feeds* that information into the LLM so that the LLM can generate an answer *grounded* in the company-specific information.

General Benefits

What are the general benefits of Conversational Search? In other words, why would an organization want to use Conversational Search?

- **Faster build time:** Conversational Search essentially automates the process of

answering FAQs. Builders no longer need to manually write or build Actions to respond to FAQs, saving them dozens of hours of work for the initial build.

- **Less maintenance required:** Conversational Search generates answers grounded in the company's knowledge base. As the company updates its knowledge base, Conversational Search generates answers using that updated information. As a result, builders no longer need to manually maintain or update Actions to respond to FAQs, saving them hundreds of hours of maintenance – the virtual assistant automatically uses the most up-to-date information to answer questions!
- **Answers are truly conversational:** Conversational Search is *conversational*. Answers generated by Conversational Search are unlikely to sound robotic, stilted, or unintelligent; they are generated in real-time as a response to the end user's unique question using an LLM.

Framework: Retrieval-augmented generation

Conversational Search is a feature in watsonx Assistant that allows Assistant to generate conversational answers that are grounded in relevant, up-to-date, company-specific information in response to end users' questions.

Conversational Search uses an AI framework called retrieval-augmented generation (RAG). RAG is a very popular starting point for enterprises that are just beginning to deploy generative AI. In 2023 alone, Client Engineering and IBM Consulting engaged in 2000+ proofs of experience (PoXs) involving RAG.

RAG has two main benefits: It ensures that the LLM generating an answer to a question has access to the most current, reliable facts relevant to that question, and that users (both end users and builders) have access to the LLM's sources, ensuring that its answers can be traced, checked for accuracy, and ultimately trusted.

Process

Now that you understand RAG – the framework underpinning Conversational Search – we will review how Conversational Search specifically implements RAG in IBM watsonx Assistant.

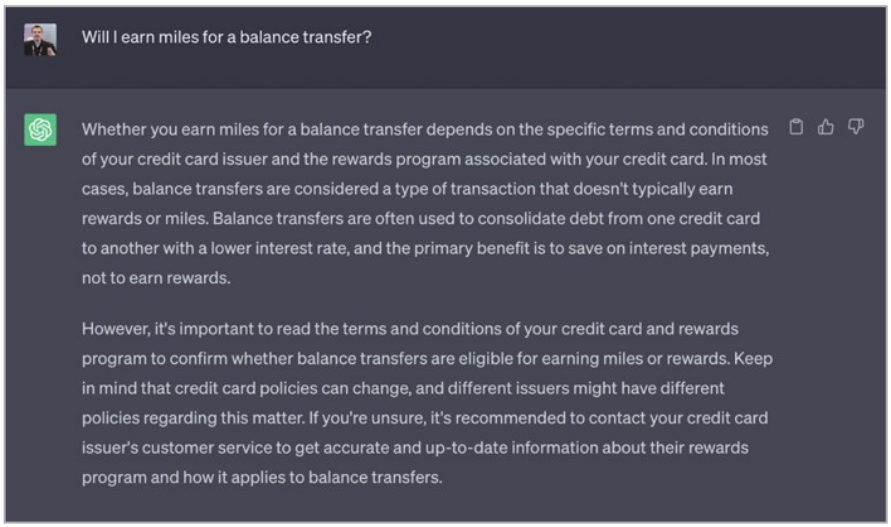
Please note that there are two general ways to implement RAG in watsonx Assistant.

First, you can use Conversational Search, the native, out-of-the-box, no-code feature built in to Assistant. Second, you can set up custom extensions to set up a custom implementation of RAG with low

code. This lab will primarily focus on Conversational Search, the native, no-code pattern built into watsonx Assistant.

Conversational Search has three core components: conversational AI, search (retrieval), and a fine-tuned or prompt-engineered LLM (generation).

Everyone's heard of ChatGPT, and you can ask it anything – for example, *will I earn miles for a balance transfer*? It queries its large language model and generates a response:



The answer is generic and demonstrates the shortcomings of the typical LLM. It is conversational, but not specific to any one enterprise or organization, and there is no way to trace the answer to its source to verify its accuracy and instill trust in the end user or builder that it is accurate and reliable.

With Conversational Search, watsonx Assistant searches a knowledge base for information relevant to a question and uses that relevant information to generate a conversational answer to the question.

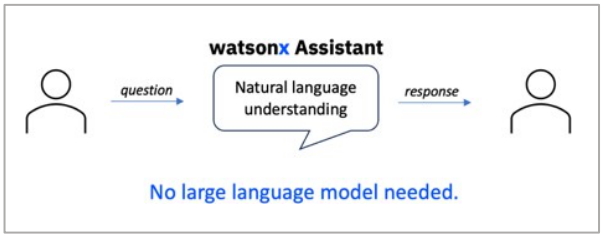
Let's walk through the order of operations in Conversational Search:

1. First, the end user asks the virtual assistant a question.

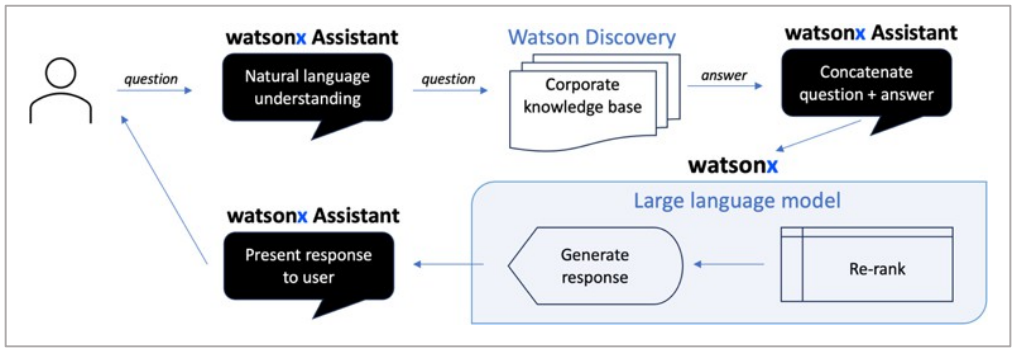
    - The virtual assistant uses its natural language understanding (NLU) model to determine whether it recognizes the question and whether it can answer

it using one of its actions for which it has been trained. For example, a bank's assistant is usually trained on answering a question such as, *help me locate the nearest branch*.

- If the virtual assistant recognizes the question, it answers it using the appropriate action (see lab *2. Actions and basic IAM*). Conversational Search is not needed. This is illustrated in the graphic below and is how the previous sections of this lab have handled user requests.



- However, if the assistant does not recognize the question, it will go to Conversational Search. The Conversational Search process is shown and explained in detail below.



2. With Conversational Search, the virtual assistant sends the end user's question, also known as a query or request, to a search tool – in this exercise, Watson Discovery. Watson Discovery has read and processed all relevant corporate documents.

3. The search tool (Watson Discovery) will then search its content and produce search results in response to the question.

4. The search tool passes these search results back to the virtual assistant in a list.

5. At this point, watsonx Assistant could display the results back to the user. However, they would not resemble natural speech; they would look more like a set of search results. Helpful – possibly – but not summarized and presented concisely and conversationally.

6. Therefore, watsonx Assistant sends the question and the list of search results to watsonx, which invokes an LLM.

   In some implementations of RAG, an LLM *re-ranks* the search results. It may

reorder or disregard some of the search results according to how relevant and useful it thinks the search results are to the question. For example, an LLM might decrease the ranking of a search result if it is from a document not recently updated, indicating the information may be outdated. This capability is often called a “neural re-ranker” and is coming soon to watsonx Assistant’s Conversational Search.

7. The LLM generates an answer to the question using the information in the search results, and it passes this answer back to watsonx Assistant.

8. The virtual assistant presents this conversational answer to the end user.

   Answer traceability. In some implementations of RAG, the virtual assistant may show the end user the search results that the LLM used to generate the answer. This allows the end user to trace the answer back to its source and confirm its accuracy. This capability is often called “answer traceability” and is available today in watsonx Assistant’s Conversational Search.

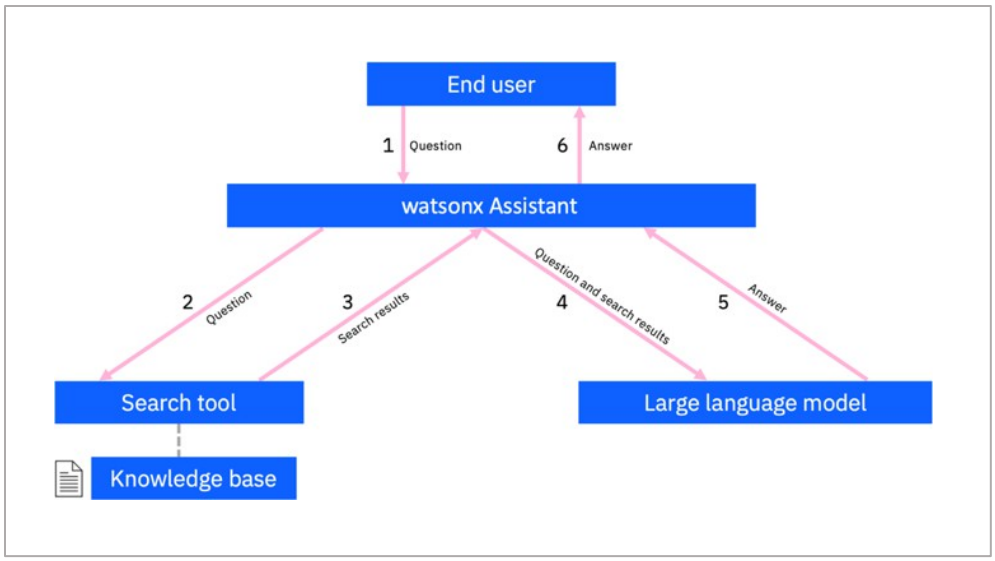
   Custom passage curation. In some implementations of RAG, the virtual assistant also shows the end-user snippets of the search results that the LLM used to generate the answer. These snippets might be direct quotes or 1-2 sentence summaries of the relevant information in each search result, allowing the end user to understand exactly what the LLM pulled out from the search result to craft its answer. This capability, often called “custom passage curation," is available today in watsonx Assistant’s Conversational Search.

Conversational Search is much more than document summarization or simple search. It includes the entire recognition, search, and answer generation process, sometimes also including a neural re-ranker, answer traceability, and custom passage curation.

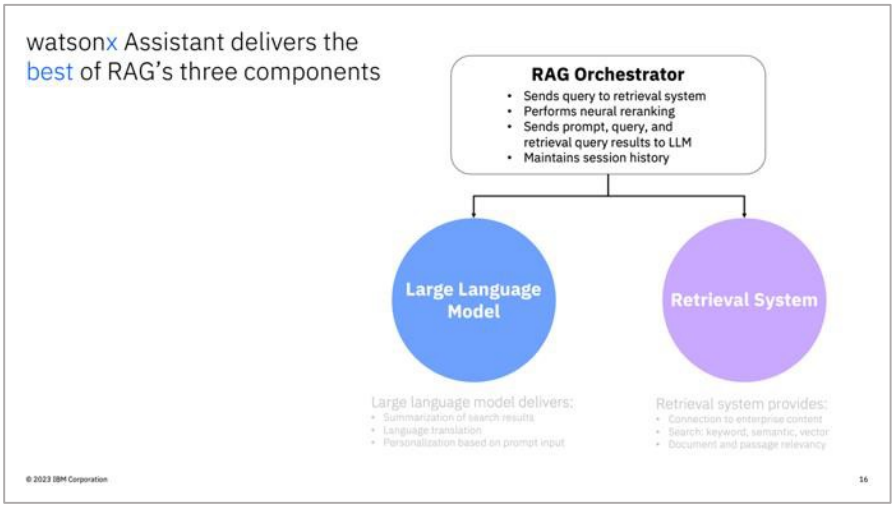
The **value-add** of watsonx Assistant in Conversational Search is its ability to orchestrate and connect every component of RAG using its NLU model, no-code actions, OOTB connectors, and custom extensions. We will discuss this in more detail in this lab – keep reading for more!

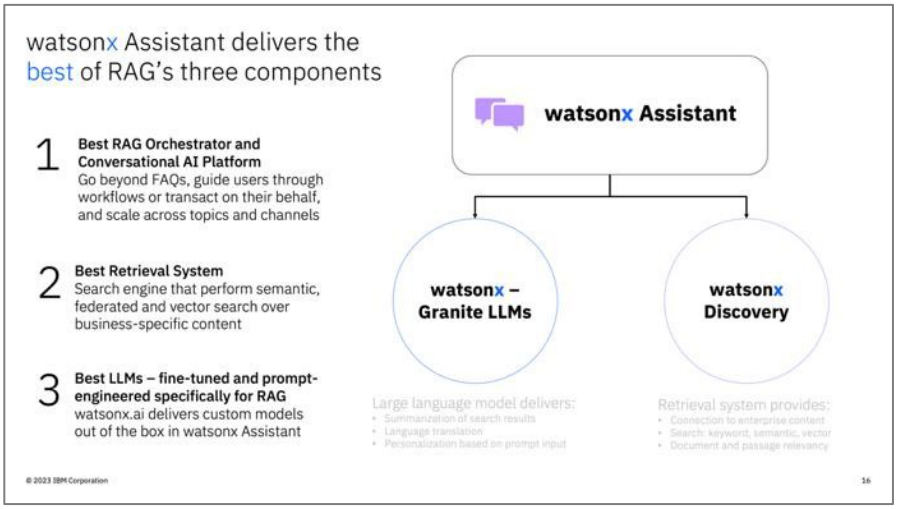
Architecture

Here is a generic architecture diagram for conversational search:



And this is another view:





LLMs in Conversational Search

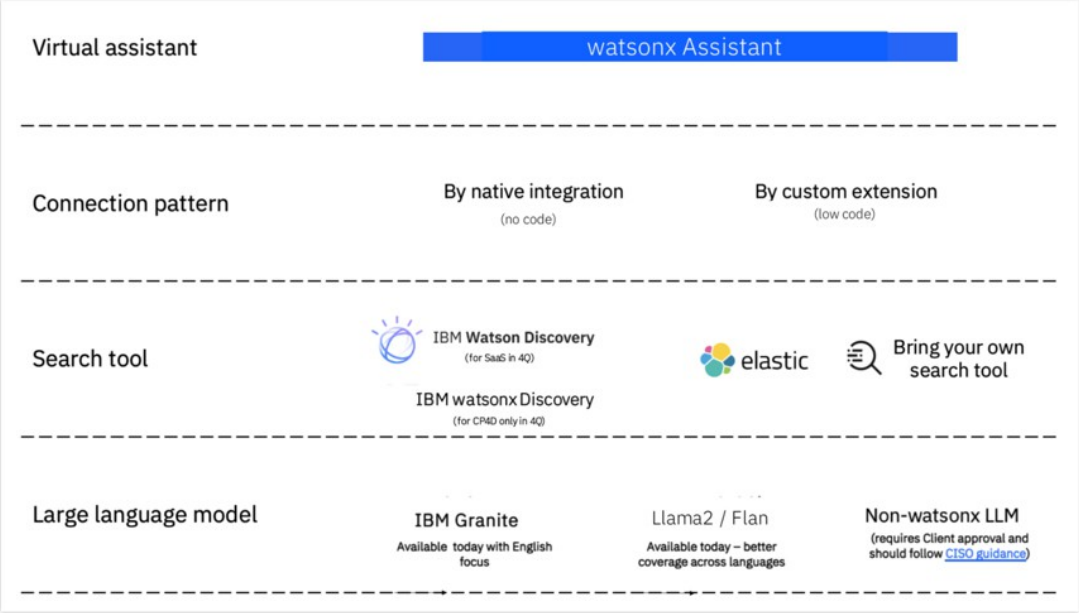
Other watsonx LLMs or non-watsonx LLMs can also be fine-tuned or prompt-engineered to perform RAG or other generative AI use cases. These LLMs can be integrated with watsonx Assistant via custom extensions. This pattern may be preferable to those who train their own

LLMs in-house or want customized functionality beyond what is available natively in watsonx Assistant through Conversational Search.

The **value-add** of watsonx.ai in Conversational Search is its customized large language model, designed and customized by the watsonx Assistant, IBM Research, and watsonx teams to perform well for conversational search.

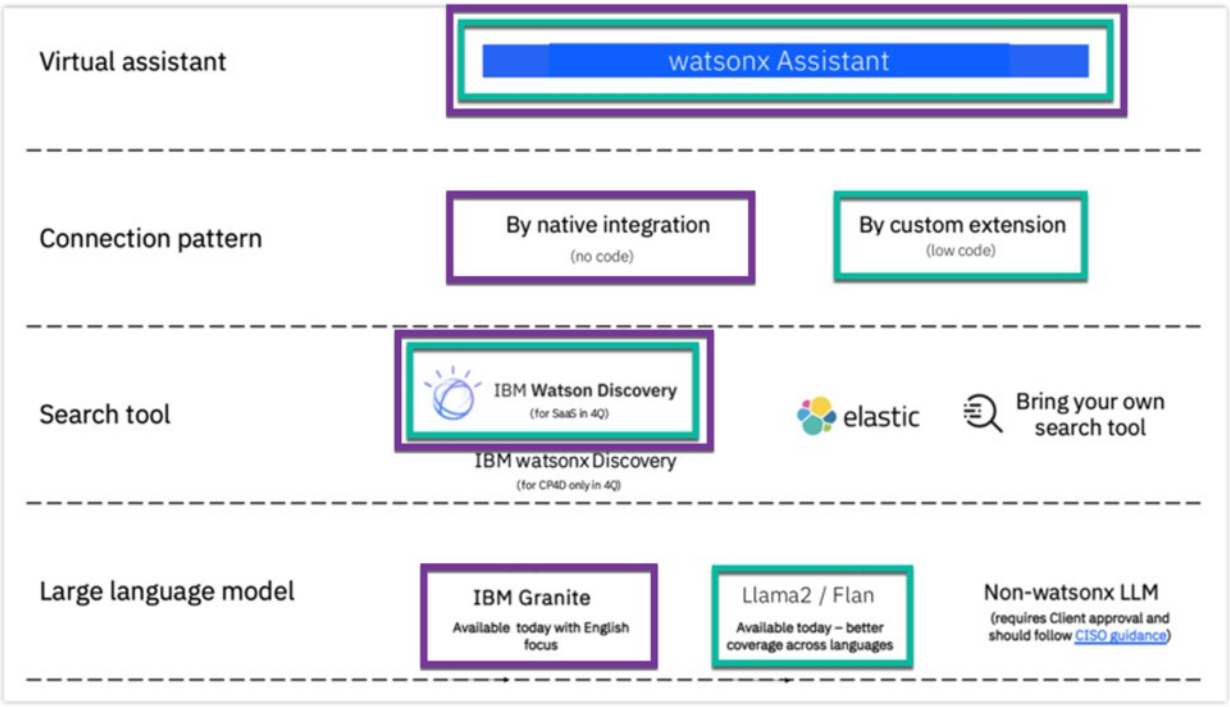
Demonstrating conversational search

Following are the options available to IBMers and IBM business partners that would like to demo Conversational Search and RAG:



This section of the lab gives you two options to implement Conversational Search with watsonx Assistant.

Option 1 uses Beta functionality, specifically the watsonx Assistant native Conversational Search feature, which does not require custom extensions or watsonx.ai. This option is emphasized in the graphic below with the purple boxes. The native Conversational Search capability is faster and easier to deploy; however, it is still in Beta.



Option 2 implements watsonx Assistant, by custom extension, Watson Discovery, and Llama 2, as emphasized above with the green boxes.

Lab Set Up

Let's walk through how to set up watsonx Assistant's native Conversational Search, which does not require the use of custom extensions or watsonx.ai. This native Conversational Search capability is fast and easy to deploy **but is still in Beta**. This will be GA'd in 1st Half 2024

* Please make sure you have an IBMId! -> https://www.ibm.com/account/us-en/signup/register.html

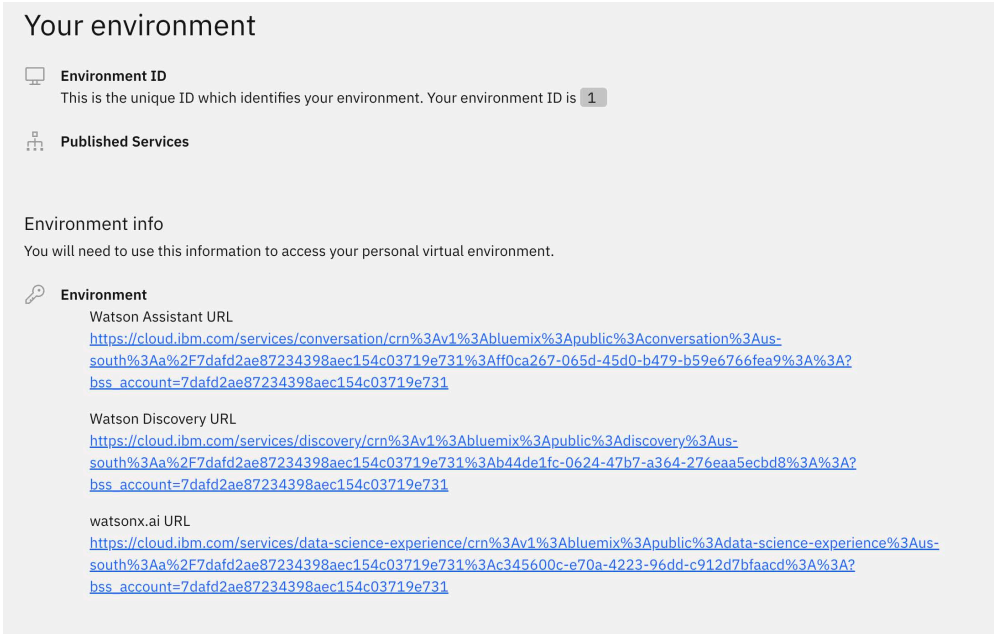
To access you environment, please go to **https://techzone.ibm.com/my/workshops/student/65f459fa2155fa001dfc9a6b**

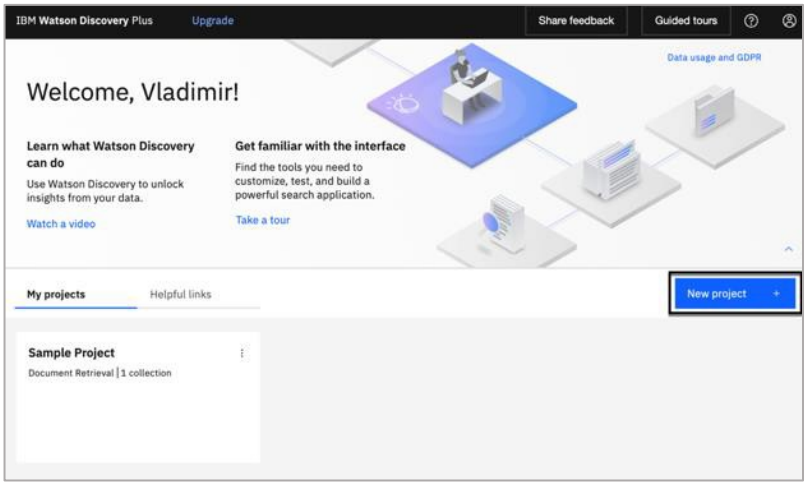
Password: **techxchange2024**

You should receive an email, **accept** the invite.

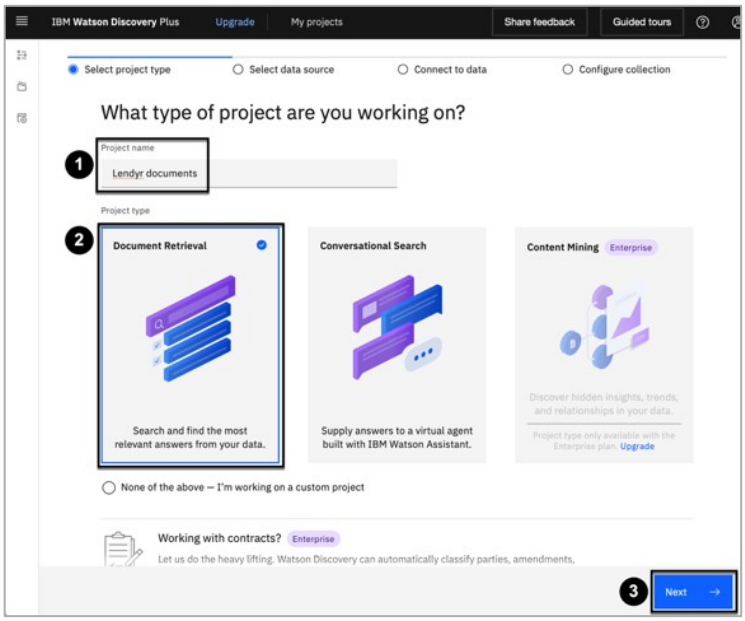
OR/AND

Click the Watson Discovery URL:

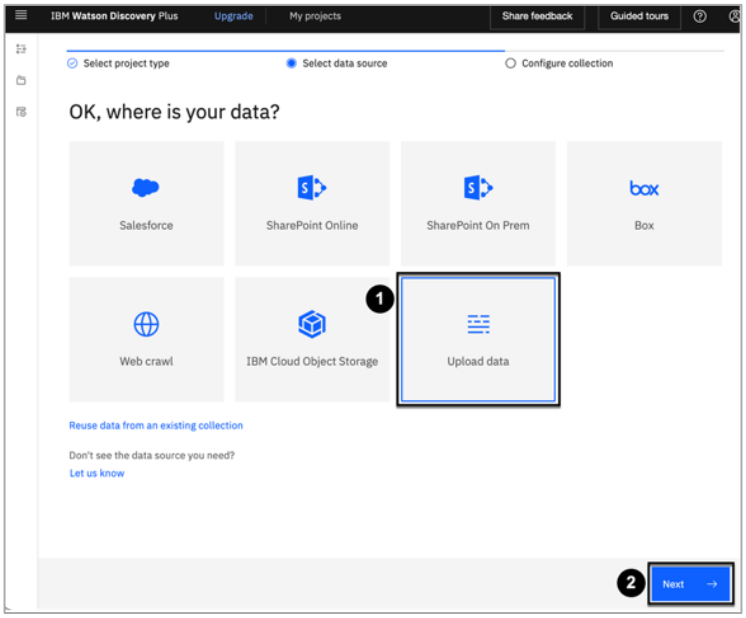




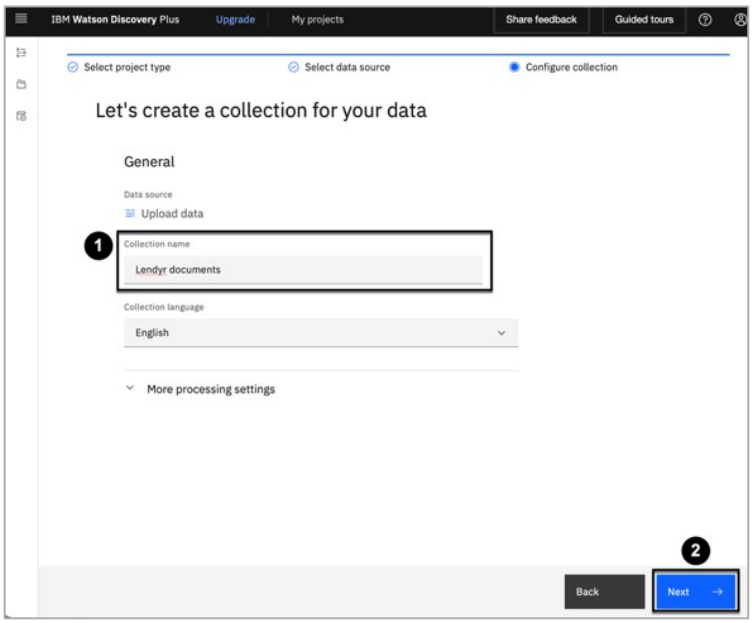
Enter a **project name (1)** such as “Lendyr documents” and select **Document Retrieval (2)**, then click **Next (3)**.



Download the [Lendyr FAQ document]. This is the file you will upload to Watson Discovery. Once downloaded, click **Upload data (1)** hit **Next (2)**.



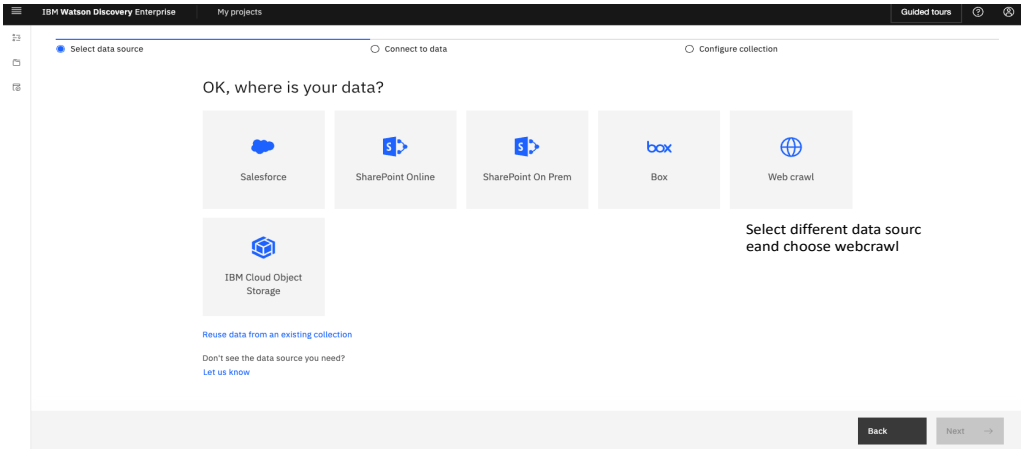
Enter a **collection name (1)** such as “Lendyr documents” and click **Next (2)**.



Now, **upload (1)** the FAQ file you downloaded, and click **Next (2)**.

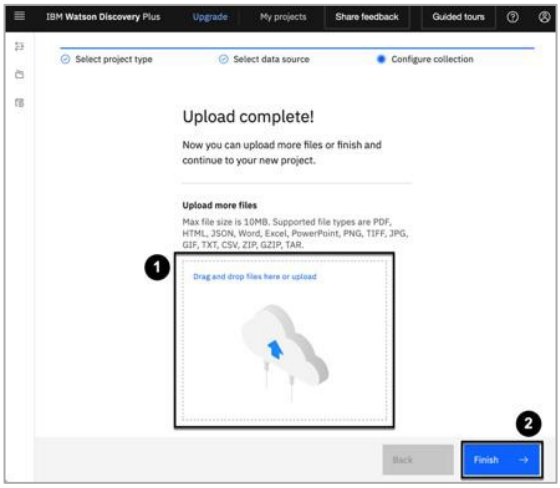
OR Do a crawl of the following website for a NASA website

https://www.nasa.gov/history/afj/ap11fj/a11-documents.html

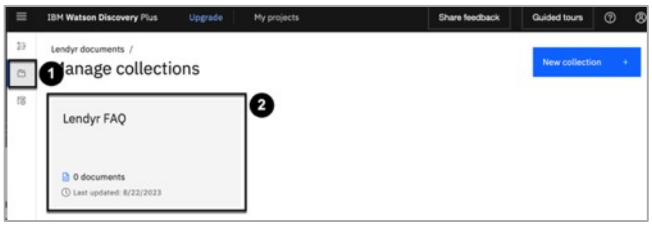


Ensure OCR is turned on

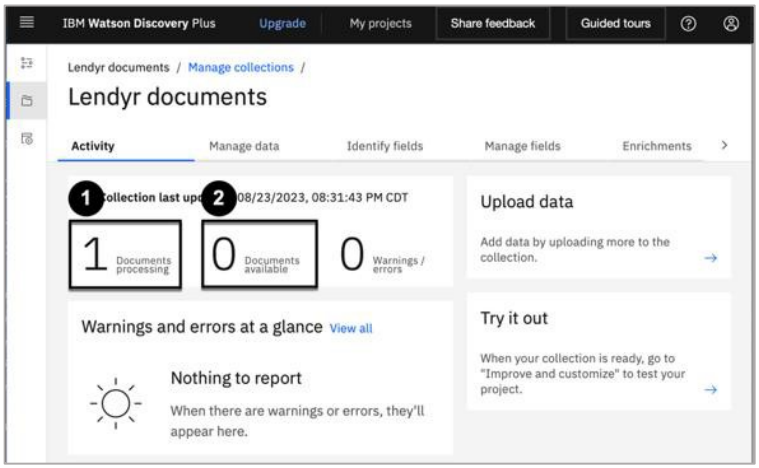
https://www.nasa.gov/history/afj/ap11fj/a11-documents.html



The file uploads relatively quickly, however it may take 15 minutes for Watson Discovery to process it. To monitor this progress, click **Manage collections (1)** and select the **Lendyr / Document FAQ** collection tile:

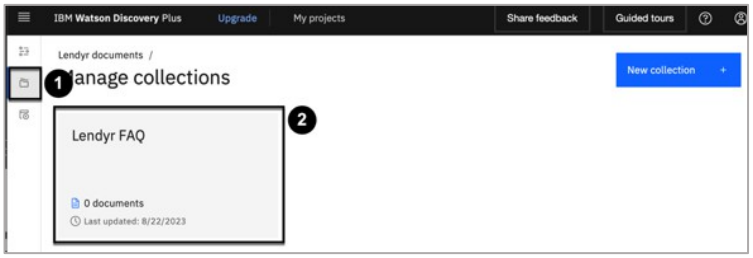


The next screen shows the collection state. Note that your document is still processing **(1)** and there are **0 Documents available (2)**. The processing will finish when the document becomes available.

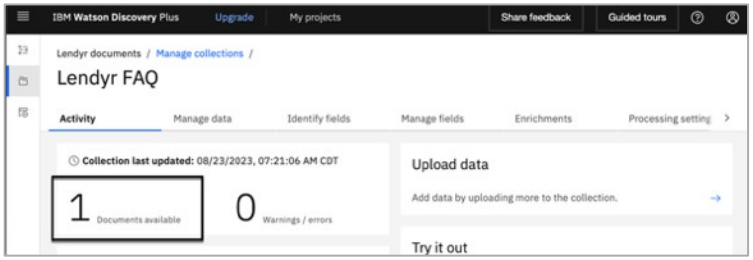


Continue to monitor the processing progress, by clicking **Manage collections (1)** and

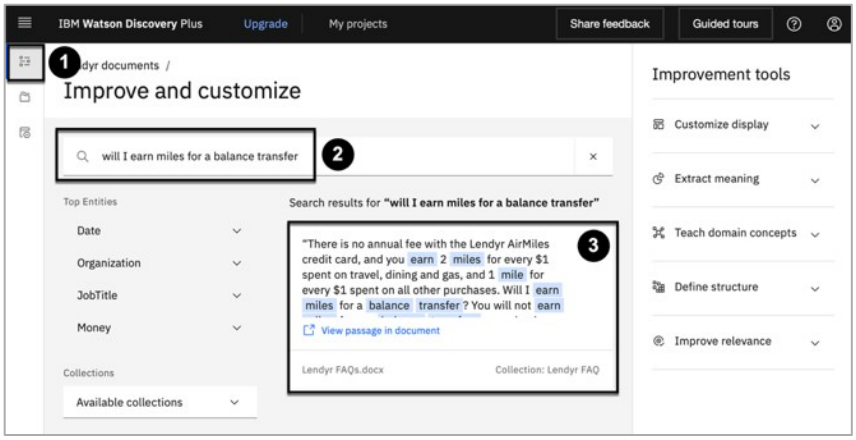
Lendyr / Document FAQ (2):



Once the screen says **1 Documents available**, you can proceed to the next step. This processing usually takes 5 to 15 minutes, so it may be a good time to take a short break.



Now that the processing is finished, let's ask Watson Discovery the same balance transfer question we asked ChatGPT at the beginning of this lab section: *will I earn miles for a balance transfer.* To do this, click on **Improve and Customize (1)**, then type **will I earn miles for a balance transfer (2)** in the query window. Note the **search result (3)**:



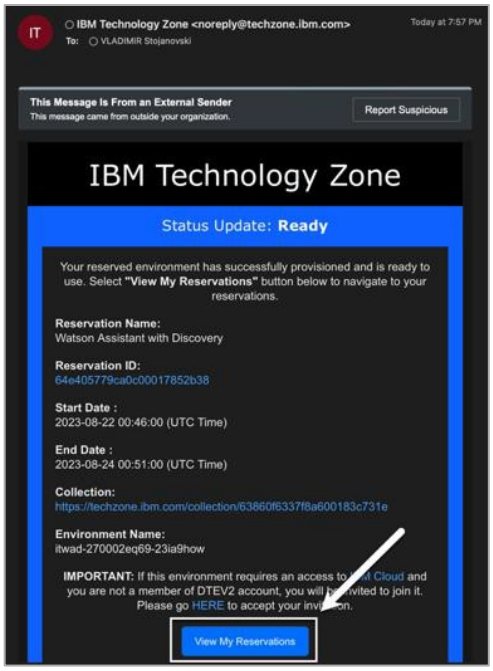
Watson Discovery provides a helpful, though not conversational, answer – an excerpt from a business document. Yet, it is more useful ("You will not earn miles for balance transfers …") than ChatGPT, because Discovery understands an organization's business content and context.

This lab will apply the power of generative language models to produce a more conversational and succinct answer to this question.

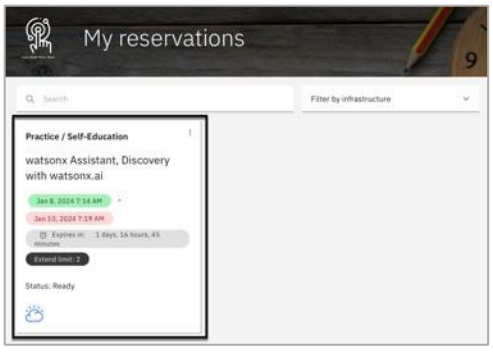
Set up the assistant (brand new instance)

Important: If you completed previous sections of this lab series and still have your TechZone environment running, skip this part and go to the header, *Use your existing assistant.*

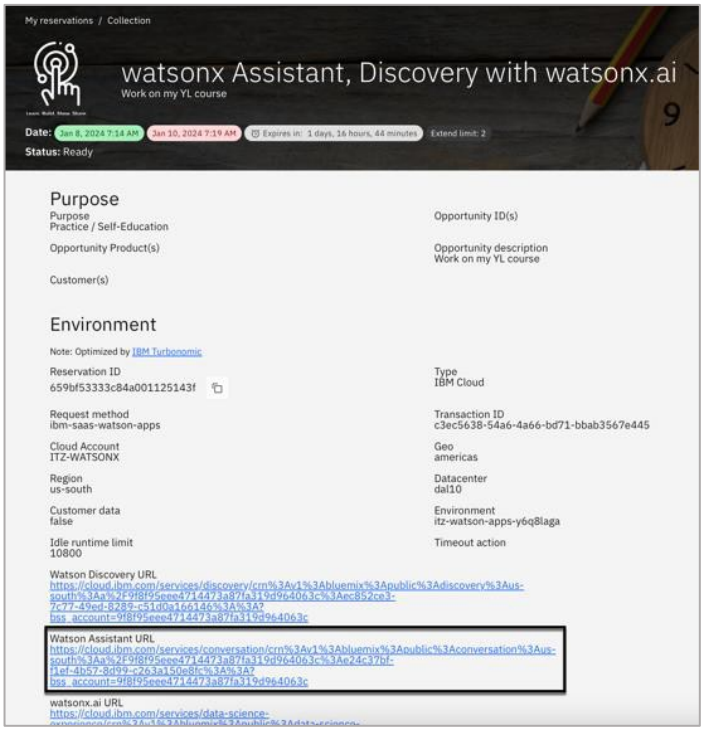
Go back to the email you received from TechZone, letting you know that your environment is ready. From that email, click on **View My Reservations**.



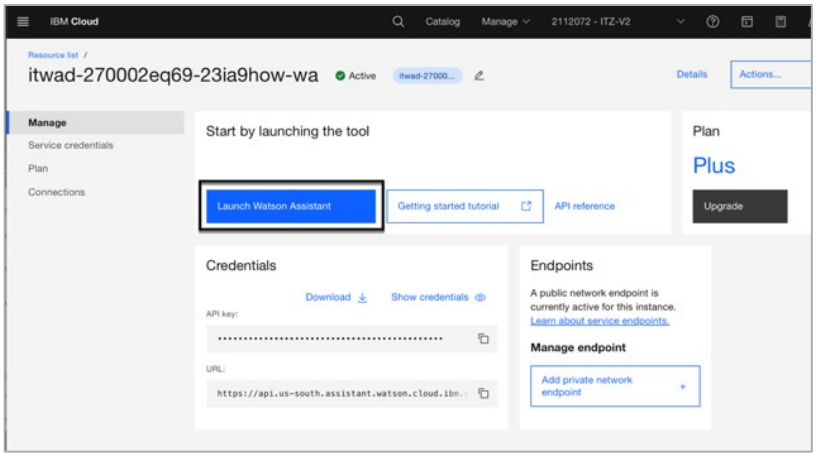
This takes you to the **My reservations** page on TechZone. Click your instance tile:



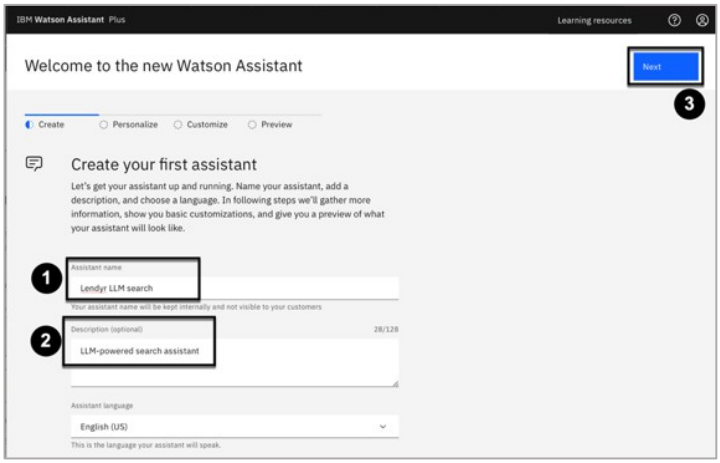
On your TechZone instance page. click the **watsonx Assistant URL**:



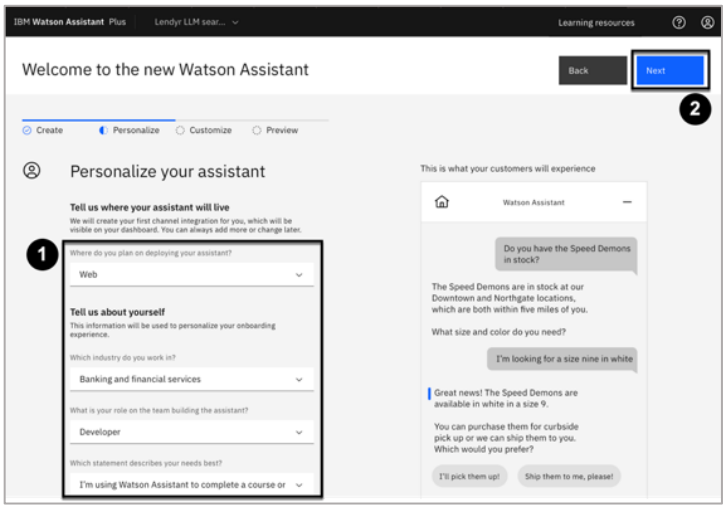
From the instance launch page, click **Launch watsonx Assistant**:



This is a new assistant, so you will first configure the basics. First, enter its **name (1)**, **description (2)**, and click **Next (3)**:



Fill out the form asking you how you will use the assistant **(1)** and click **Next (2)**:



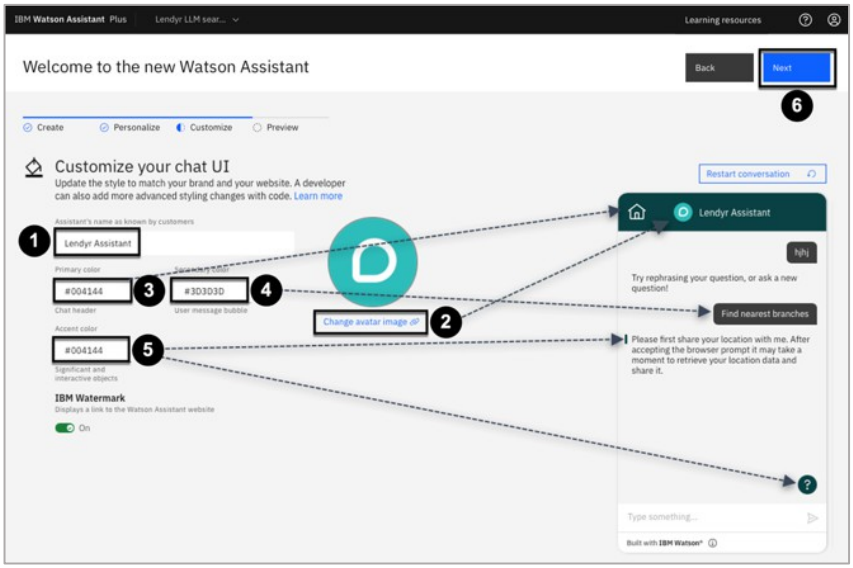
Customize your assistant to look like the Lendyr Bank website || OR Nasa || OR [insert your own branding here]

Lendyr (fictious bank) Example:

The next screen allows you to customize the look and feel of the Assistant; customize it to look like the Lendyr Bank assistant (you will recall going through these steps in Section 2 of this lab):

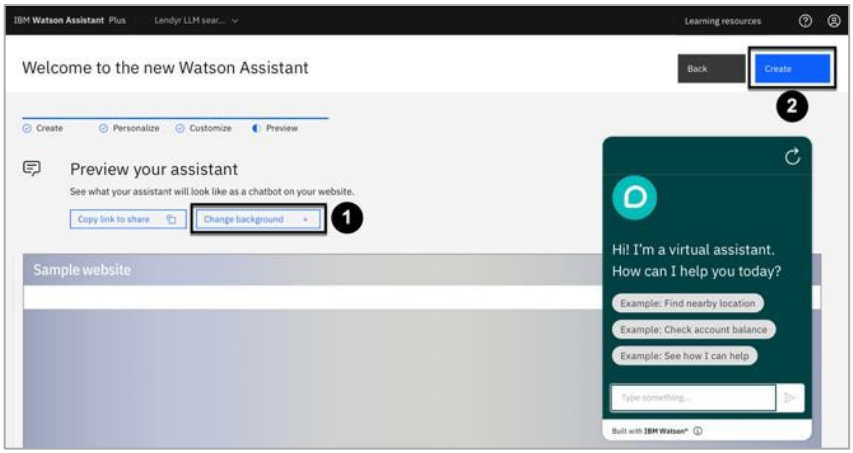
1. Change **Assistant's name as known by customers** to “Lendyr Assistant”.
2. Click on **Add an avatar image** and point to https://web-chat.global.assistant.watson.appdomain.cloud/assets/Lendyr-Avatar.png.
3. Change **Primary color** to: #004144. This is the color of the chat header.

4. Change **Secondary color** to: #3D3D3D. This colors the message bubble.
5. Change **Accent color** to: #004144. This is a tertiary color that accentuates certain assistant responses and icons, as shown below.



Your assistant should now look like the image above. (Note, some color changes may require that you restart the chat.)

On the following preview screen, you have the option to **change the background (1).** We won't do that because we'll embed the assistant into the Lendyr site, but this is a great option to customize the look of an assistant so it feels like it is embedded in a your website. Click **Create (2)**.



Important: Skip the next part, *Use your existing assistant,* and move directly to the header *Add the Search extension.*

Use your existing assistant

When using your existing instance, there are two steps to deploying the Conversational Search functionality:

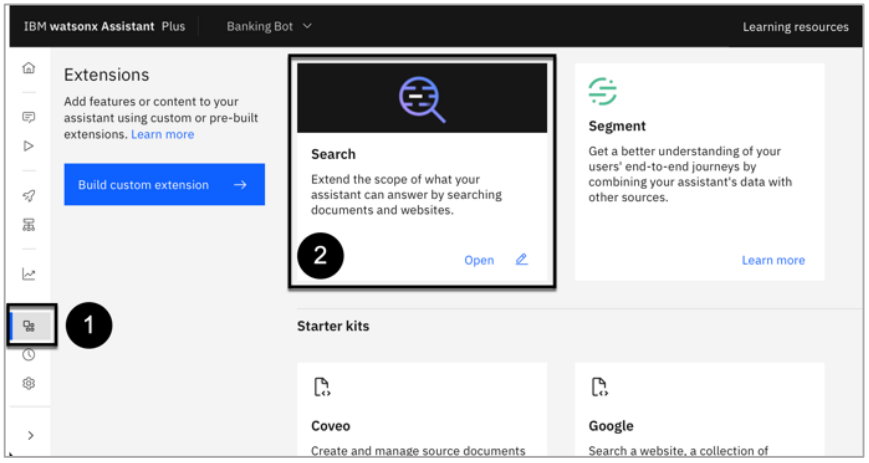
1. Add the Search extension, which will connect to your Watson Discovery instance.
2. Invoke the Conversational Search through one or more Actions.

Open your existing Assistant instance, and proceed to *Add the Search extension.*

Add the Search extension

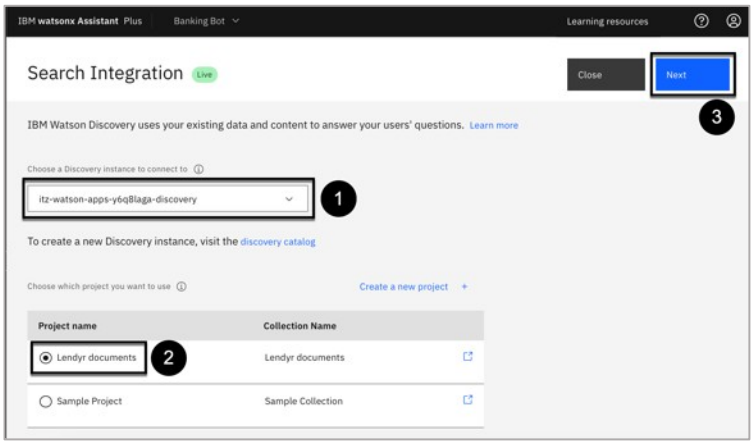
Now that you have a Watson Discovery instance setup and an Assistant configured, you can proceed to add the search extension to your assistant.

In watsonx Assistant, use the left menu to open the **Integrations (1)** page. Then, scroll down and click the **Search (2)** tile:



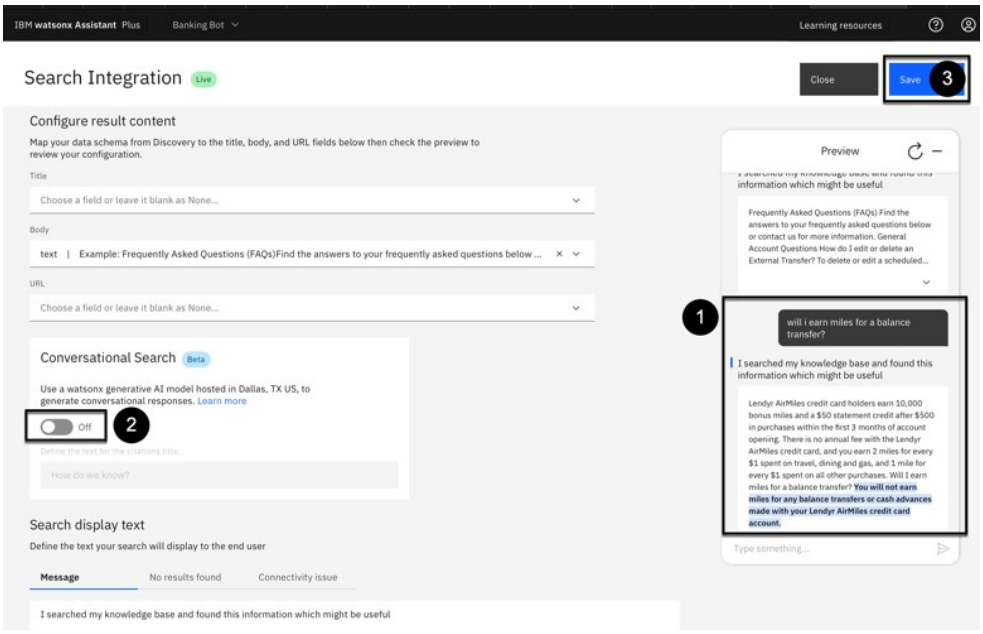
Select the **Watson Discovery** search extension:

Next, select the instance of Watson Discovery where you loaded the Lendyr FAQs || Nasa Website crawl || personal company hub's doc **(1)**, select the **documents** project **(2)**, and click **Next (3)**:



The following screen is where you map your data schema from Watson Discovery to the title, body, and URL fields that will be used by watsonx Assistant. (Review the product documentation here for more information about the options on this screen – you will leave the default values in this lab.)

First, test the search integration in the Preview window by typing, "Will I earn miles for a balance transfer?" **(1).** Note that the response is roughly the same answer you got in Watson Discovery earlier:



You have now successfully integrated Watson Discovery with watsonx Assistant! You will now take this integration one step further by toggling **Conversational Search**

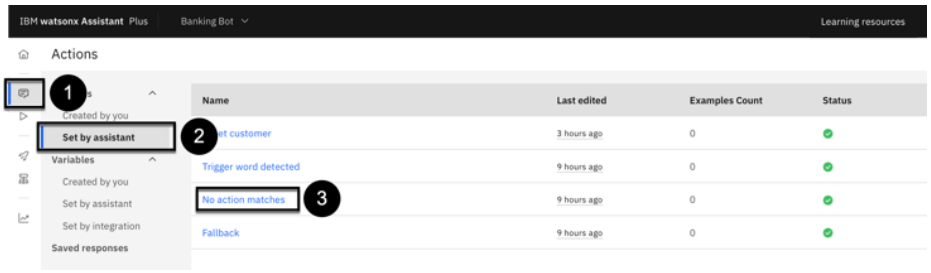
from **Off** to **On (2)** and clicking **Save (3)**.

Add a Search step to an Action

As you've experienced in this lab series, watsonx Assistant is powerful and versatile in its ability to *tell, show* and *do.* Often, Conversational Search will not be the first option for end users – for example, users who want to perform a transaction will likely be guided through a process managed through one or more Actions.

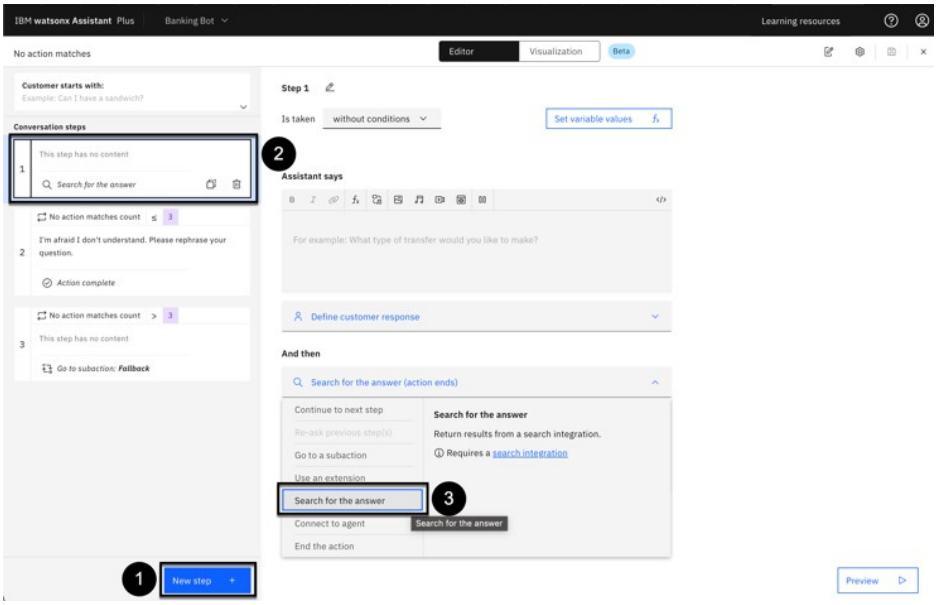
In this lab, you will set up Conversational Search to invoke when the assistant cannot match the end-user question to one of the Actions. The simplest way to do this is to add a Step to an Action provided automatically with each Assistant instance: the *No action matches* action.

To do this, go to the **Actions (1)** window, click on the actions which are **Set by assistant (2)**, and select the **No action matches (3)** action:

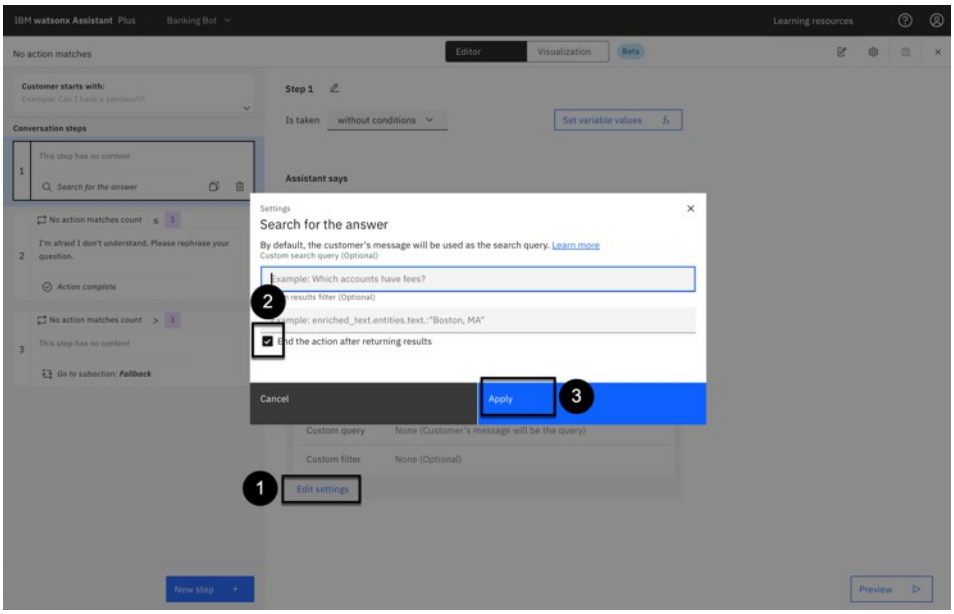


Then, as shown below,

1. Click **New step +**
2. Make sure this new step is the first in the sequence (you can drag and drop it in the list)
3. Under **And then** select **Search for the answer**



Next, click **Edit settings (1),** and in the ensuing popup, select **End the action after returning results (2)**, and click **Apply (3)**:



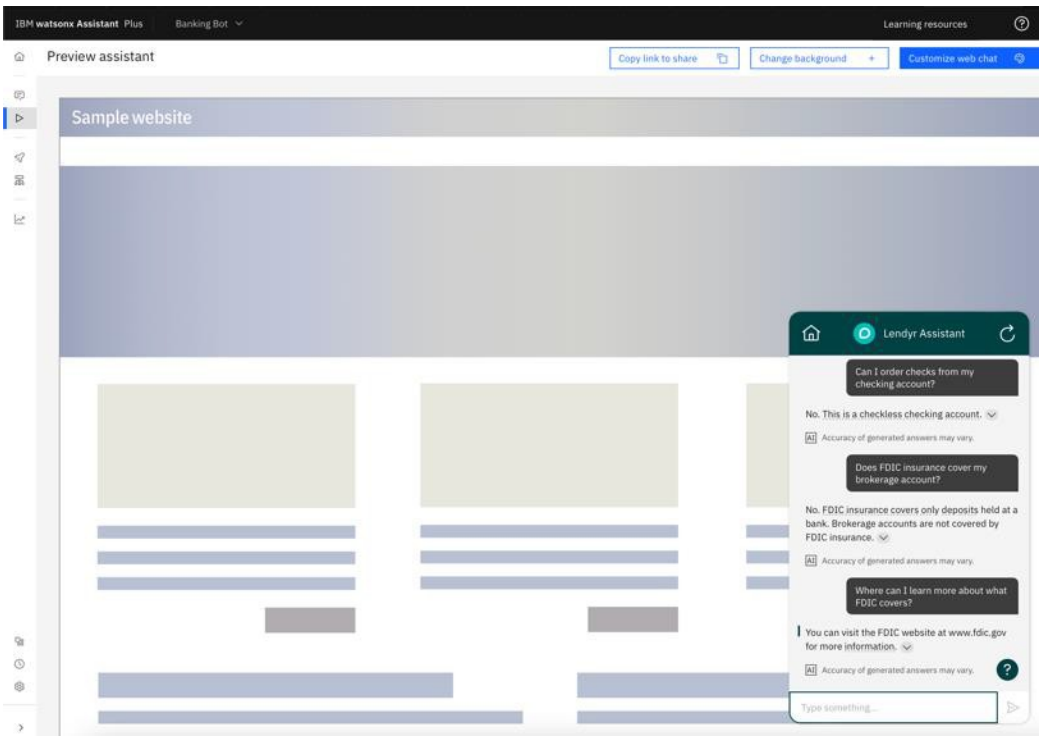
Save the action.

Try it out!

Now that you are finished configuring your assistant, try it out! Click on **Preview** and enter these questions. None of the actions have been trained to answer these questions:

- Will I earn miles for a balance transfer?
- What is Lendyr Bank's routing number for electronic transactions?
- How can I get the 20% discount with the Lendyr AirMiles credit card?
- Can I order checks for my Checking account?
- Does FDIC insurance cover my brokerage account? *
- Where can I learn more about what FDIC covers? *

* There are no answers to these questions in the FAQs you uploaded to Watson Discovery; the response is provided by the IBM Granite Large Language Model.



Note the clear, concise, and conversational answers. Compare this to the generic answers from ChatGPT and the excerpt answer from Watson Discovery earlier!

Option 2: Conversational search using Llama2

Let's walk through how to set up conversational search using watsonx Assistant, by custom extension, with Llama 2.

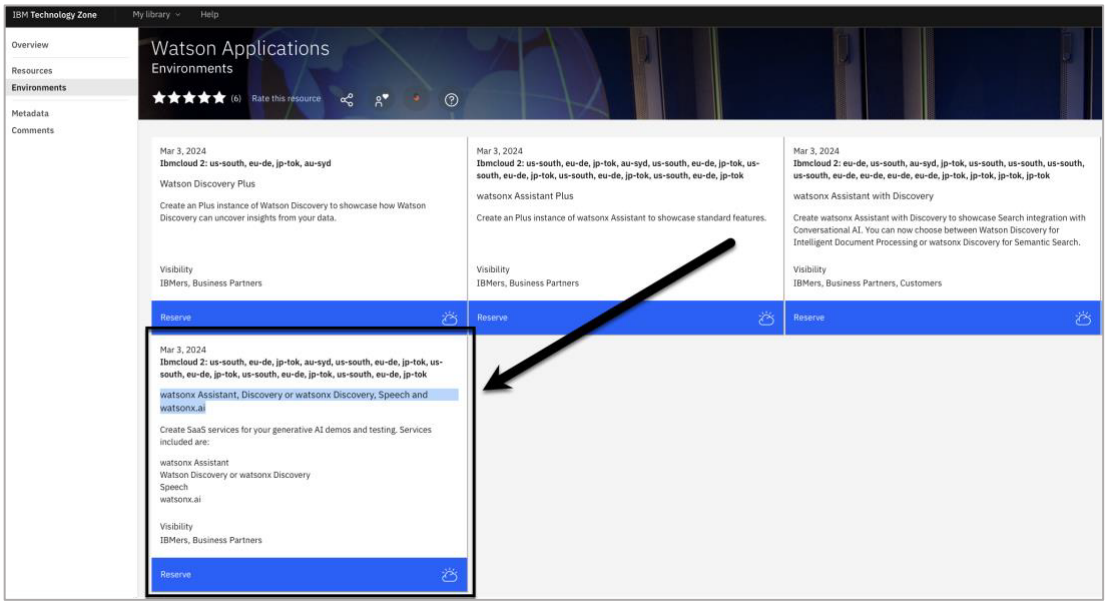
In this lab section, you will create a new virtual assistant. At the end of this exercise, you will have the ability to transfer the capabilities you will build in this section, to the assistant you built in Sections 1-6, but only if both assistants are running within the same instance of watsonx Assistant. You will copy-and-paste the actions and reconfigure the exensions.

For a preview and demonstration of what you are about to build in this section, view this video.

Provision the watsonx Assistant and related services on TechZone

Important: If you have just completed Section 6 of this lab series and still have your TechZone environment running, skip this part and go to the header, *Provision Watson Machine Learning and Object Storage services on IBM Cloud.*

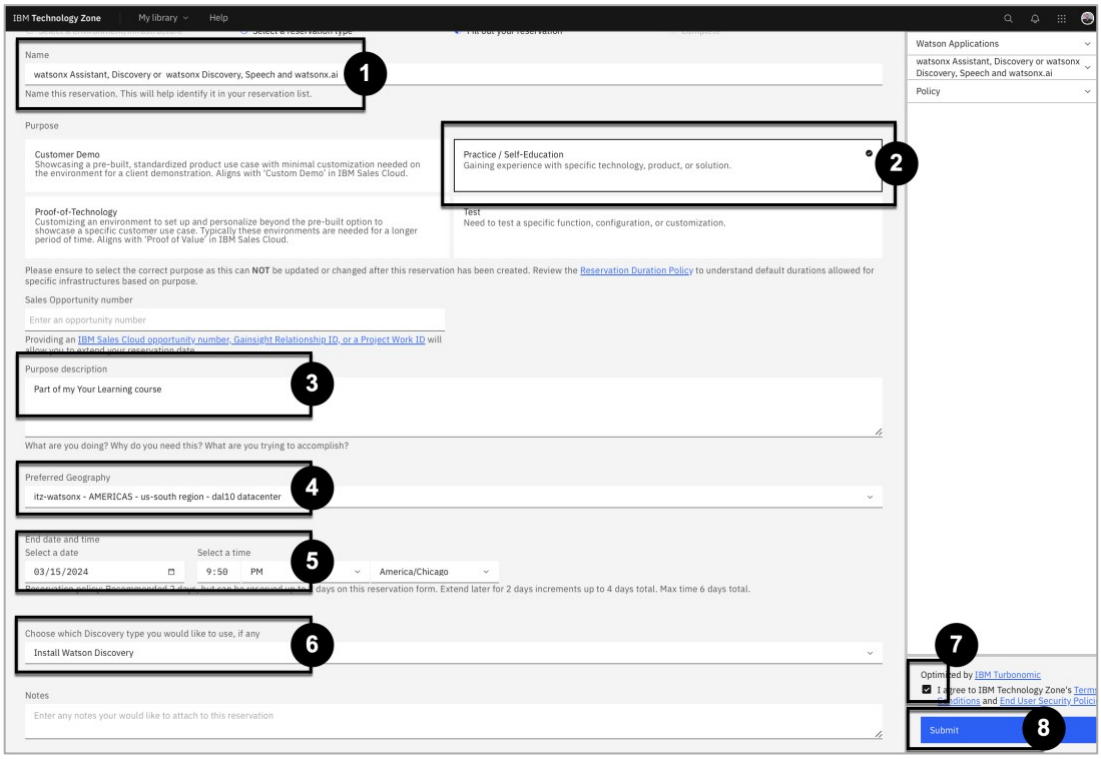
Navigate to the Watson Applications collection on TechZone, and click on the **watsonx Assistant, Discovery or watsonx Discovery, Speech and watsonx.ai** tile to reserve an environment:



Next, on the reservations page, select the **Reserve now** radio button (not shown). Then,

1. Enter a **name** for the reservation that is meaningful to you.
2. Under **Purpose**, select the **Practice / Self-Education** tile.
3. Enter a **Purpose description**.

4. Select a **Preferred geography**.
5. You can leave **End date and time** as-is, as it defaults at the maximum 2 days.
6. Choose to install **Watson Discovery**
7. Accept the **Terms & Conditions**.
8. Click **Submit**.



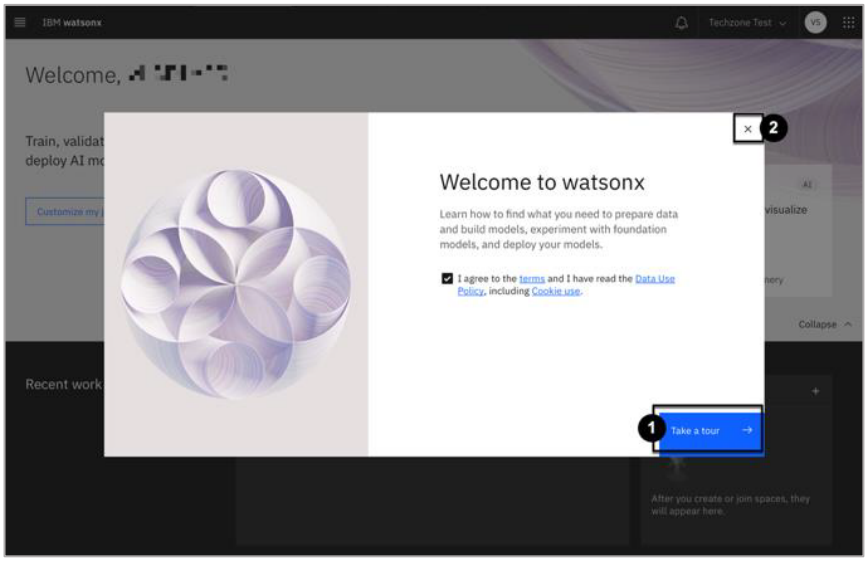
You will soon receive an email that your environment is provisioning. While you wait for it to be provisioned (usually less than 10 minutes), you can continue with the next steps.

Note: This environment can be extended up to two times, for a maximum of 6 days in total. You will receive multiple expiration warnings from TechZone before the environment expires, which will give you an opportunity to extend it, up to the 6-day maximum.

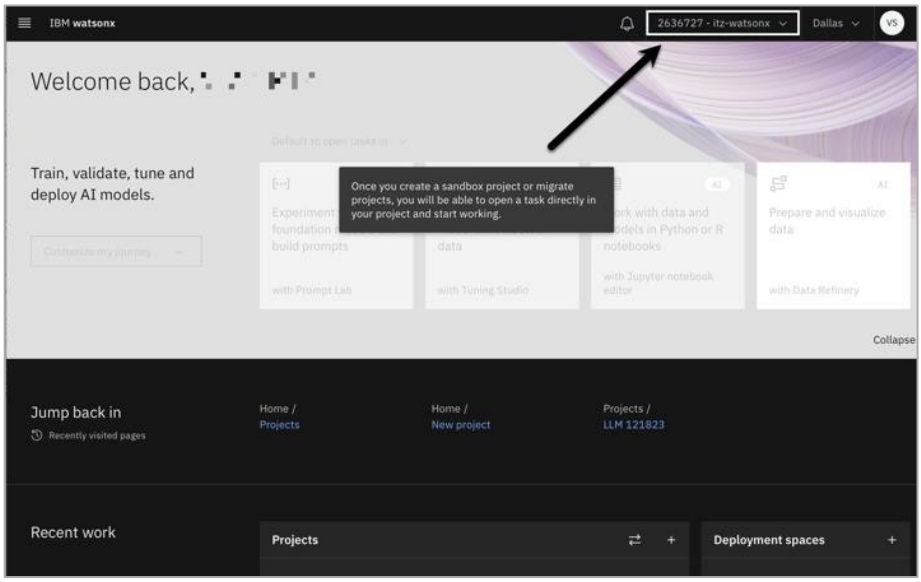
Create a watsonx project

IBM watsonx is an enterprise-ready AI and data platform designed to multiply the impact of AI across your business. It provides an API for interacting with generative language models. In this step, you will create a watsonx project, which will later allow you to connect watsonx Assistant to the watsonx API.

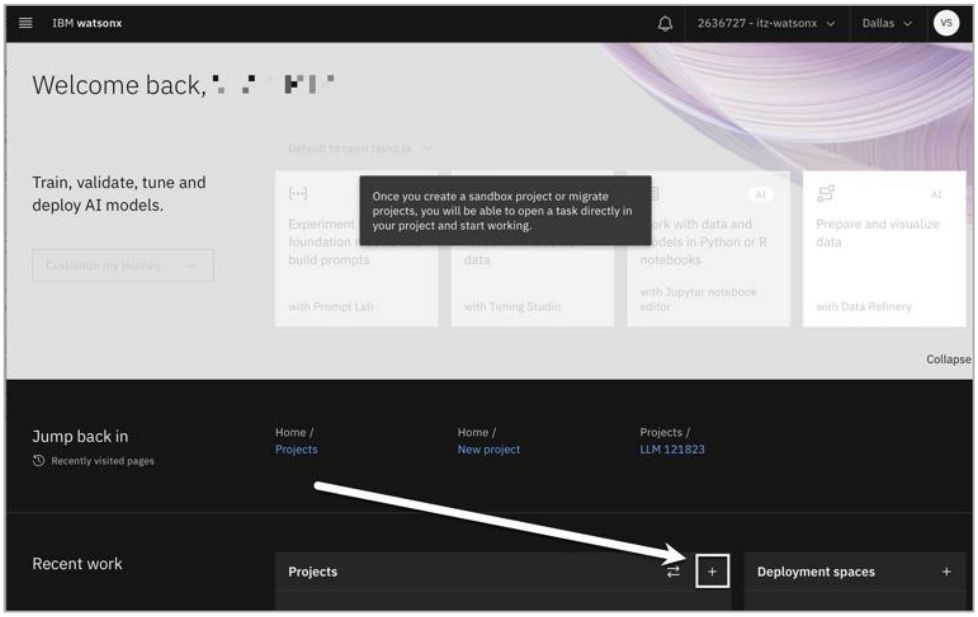
First, navigate to watsonx. You may see a splash screen like below. If so, you may **Take a tour**
(1) or **exit (2)** the screen.



Once on the main watsonx page, first, ensure you are under the same TechZone account that your watsonx Assistance is in (your TechZone account may be different from the screenshot). This will allow you to take advantage of the pre-created Object Storage and Watson Machine Learning services:

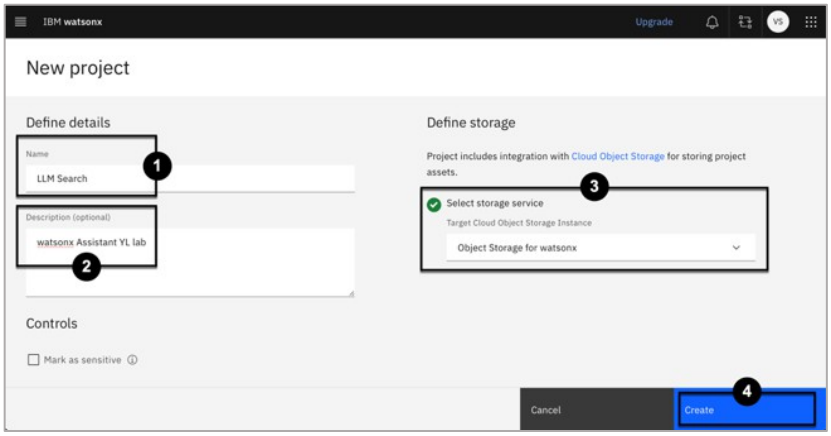


Then, create a new project by clicking the **+** button under **Projects**:

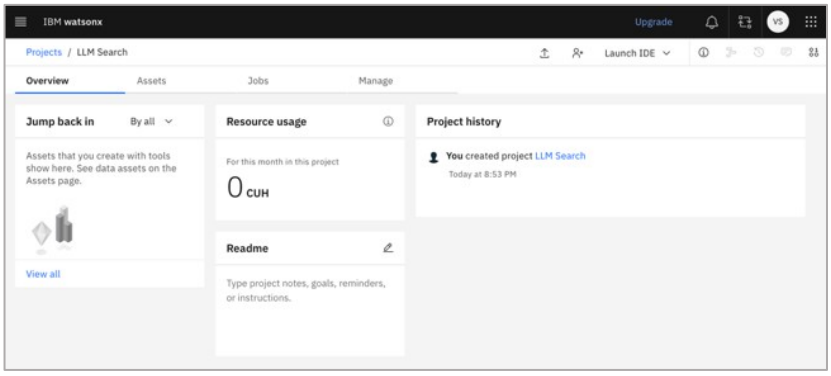


On the **New project** screen,

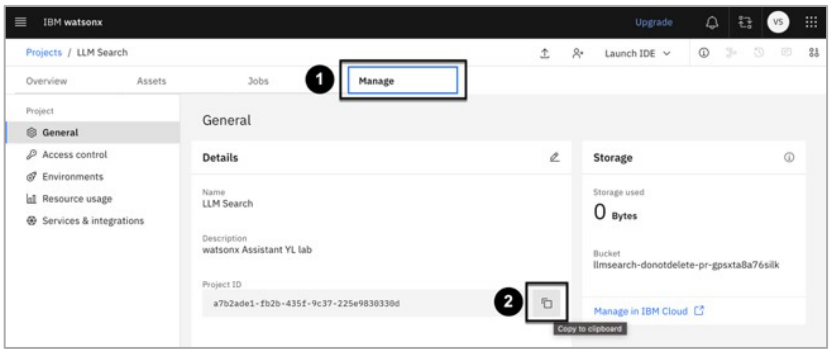
1. Enter a **Name** that is meaningful to you.
2. Optionally, enter a **Description**.
3. The **Object storage** service will be automatically populated (your service name will be different from the screenshot).
4. Click **Create**.



Next, the **Overview** tab of your new watsonx project screen is shown.

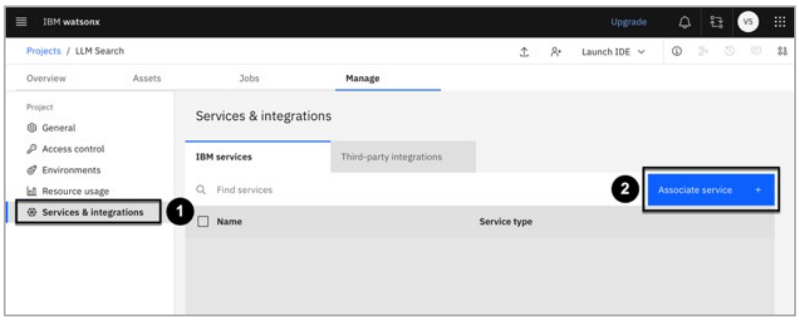


You will need the watsonx project ID to set up the action that calls the watsonx custom extension in Assistant later on. To get the watsonx project ID, select the **Manage (1)** tab, then click **copy (2)**.

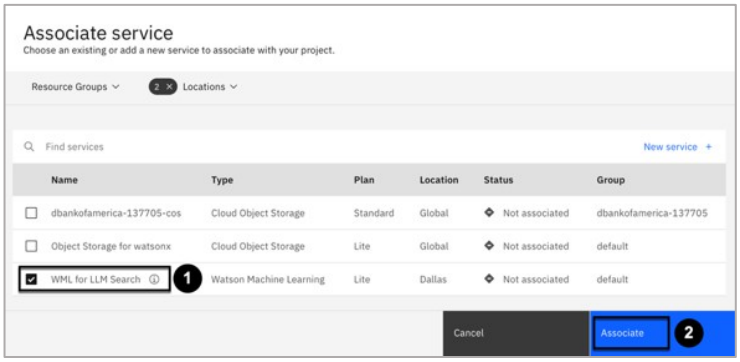


Save this project ID in a notepad for now, as you will need it shortly. Label it properly, as you will be copying a few more IDs and URLs.

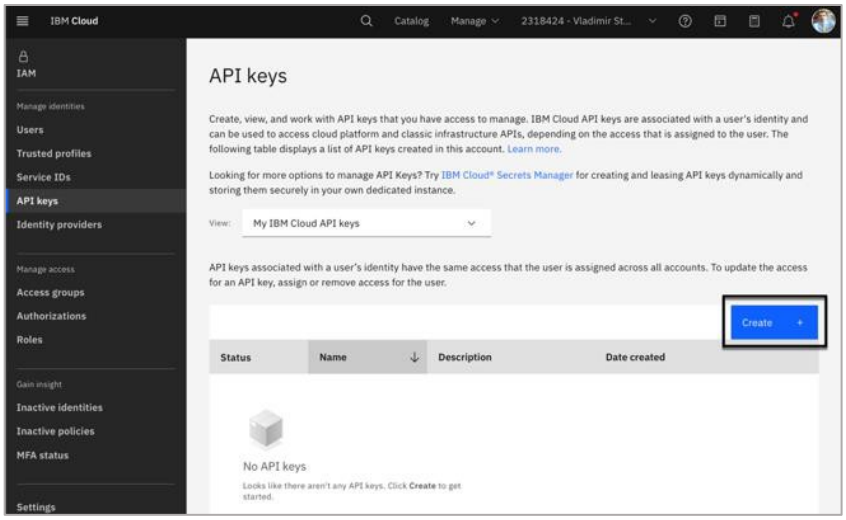
Next, you will link the automatically provisioned Watson Machine Learning service instance to this new watsonx project. To do this, select **Services & integrations (1)** and click **Associate service + (2)**:



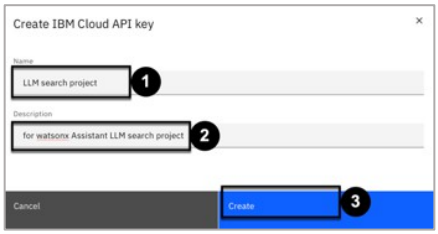
On the next screen, select the **Watson Machine Learning instance (1)** and click **Associate (2)**. Your service instance will have a different name than shown in the screenshot:



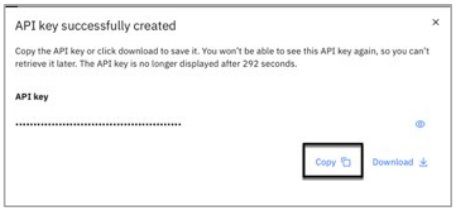
Setting up your watsonx extension in Assistant will also require an API key from your IBM Cloud account. To get the API key, in a new browser tab, navigate to Manage access and users - API Keys in IBM Cloud, which will take you to the following screen. Then, click **Create +**:



On the popup *Create IBM Cloud API key* screen, enter a **Name (1)** and **Description (2)** meaningful to you, then click **Create (3)**:



When you see the notification API key successfully created, click **copy**:



Save this API key somewhere safe and accessible. You will need this API key later to set up the watsonx custom extension in Assistant.

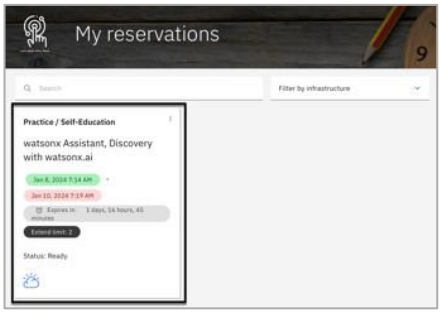
You can close this IBM Cloud browser tab.

Load a document into Watson Discovery

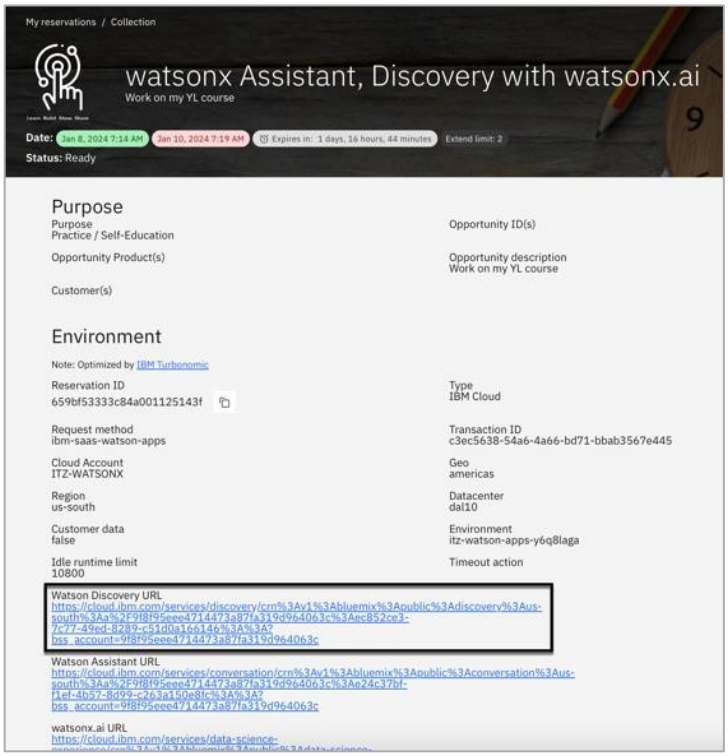
Next, you will open your Watson Discovery instance.

From your email notifying you that your TechZone environment is ready, click on **View My Reservations**.

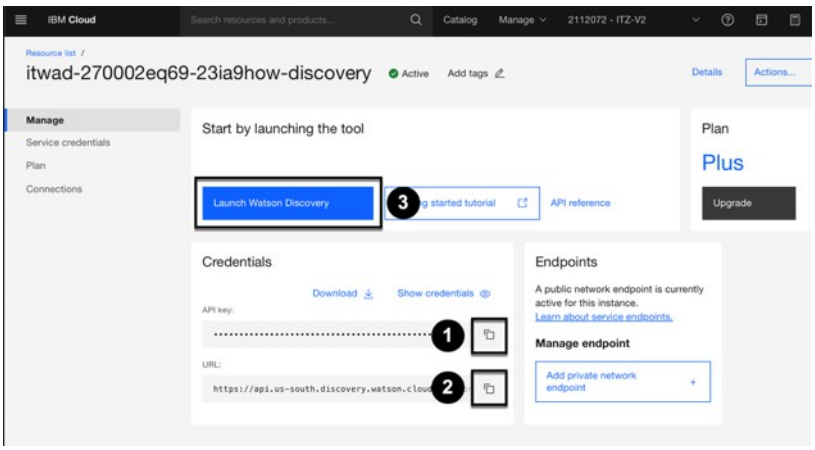
This takes you to the **My reservations** page on TechZone. Click on your instance tile:



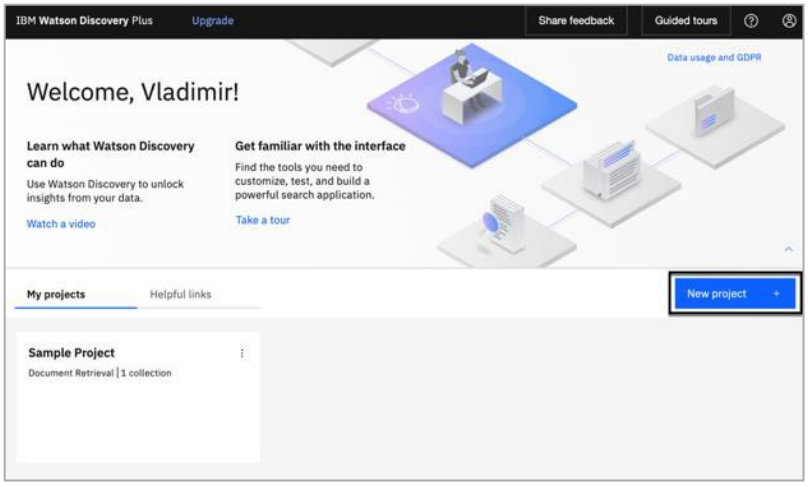
Next, your TechZone instance is shown, along with the links to your Assistant and Discovery instances. Click the **Watson Discovery URL**:



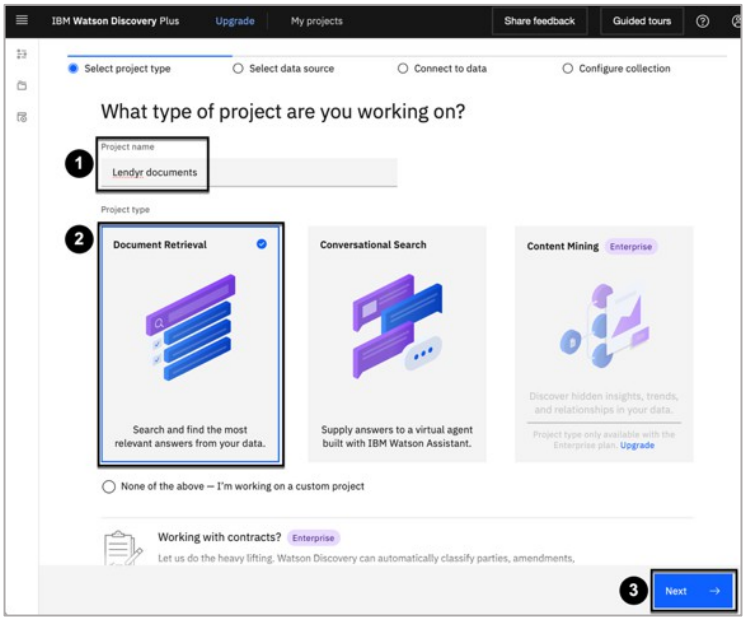
On the Watson Discovery launch page, copy the **API Key (1)** and service **URL (2)** and save them. You will need these to configure the Watson Discovery custom extension in watsonx Assistant. Then, click **Launch Watson Discovery (3)**.



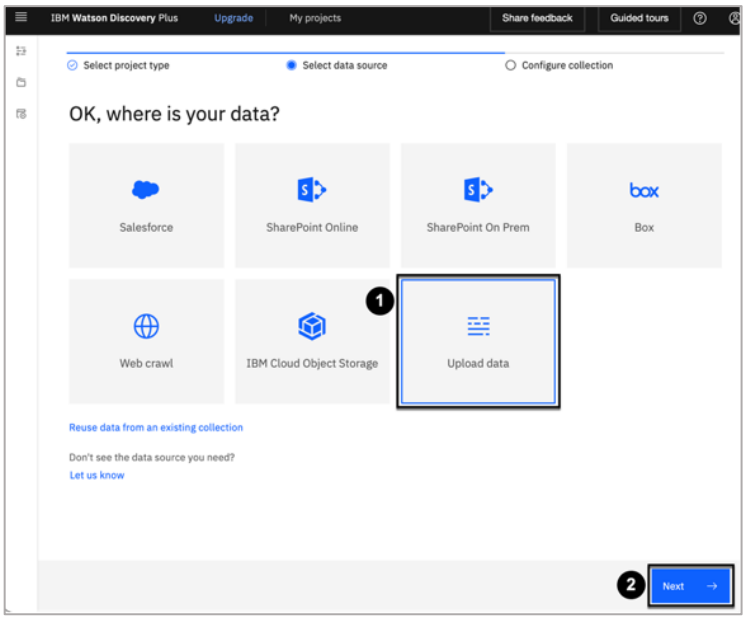
On the welcome page, click **New Project +**.



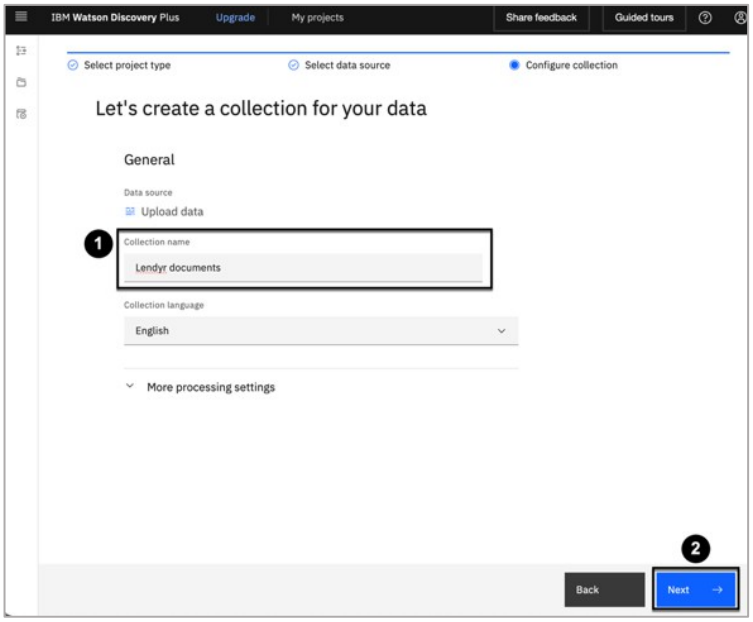
Enter a **project name (1)** such as “Lendyr documents” and select **Document Retrieval (2)**, then click **Next (3)**.



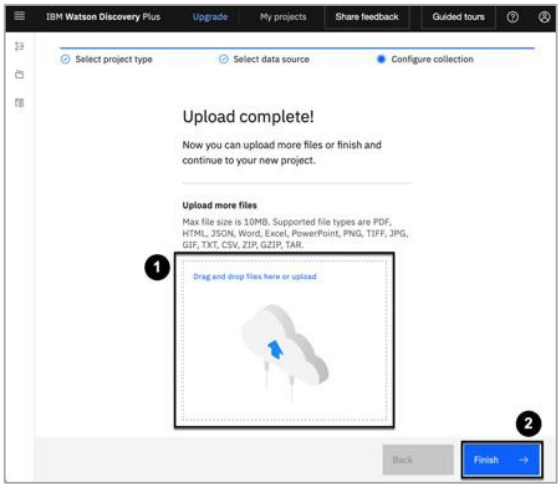
Download the [Lendyr FAQ document]. This is the file you will upload to Watson Discovery. Once downloaded, click **Upload data (1)** hit **Next (2)**.



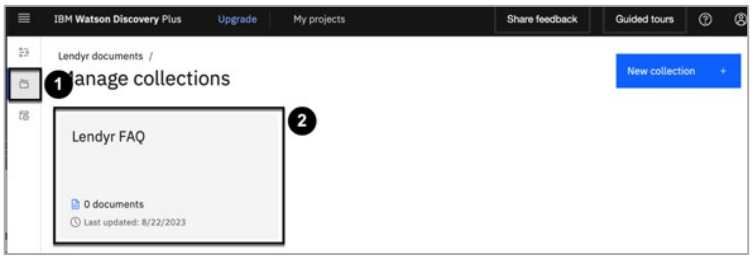
Enter a **collection name (1)** such as “Lendyr documents” and click **Next (2)**.



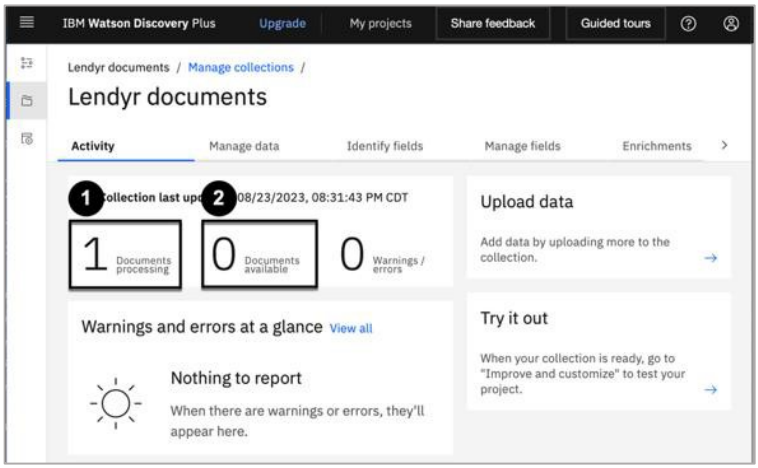
Now, **upload (1)** the FAQ file you downloaded, and click **Next (2)**.



The file uploads relatively quickly, however it may take 15 minutes for Watson Discovery to process it. To monitor this progress, click **Manage collections (1)** and select the **Lendyr FAQ** collection tile:

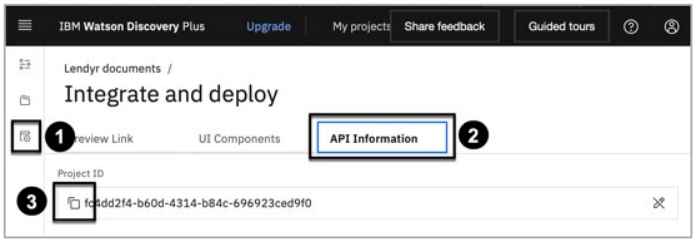


The next screen shows the collection state. Note that your document is still processing **(1)** and there are **0 Documents available (2)**. The processing will finish when the document becomes available.

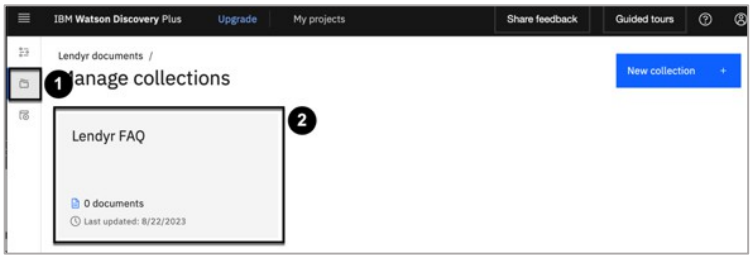


While you wait for the document to process, get the Watson Discovery project ID. You will need it when you configure the watsonx Assistant action.

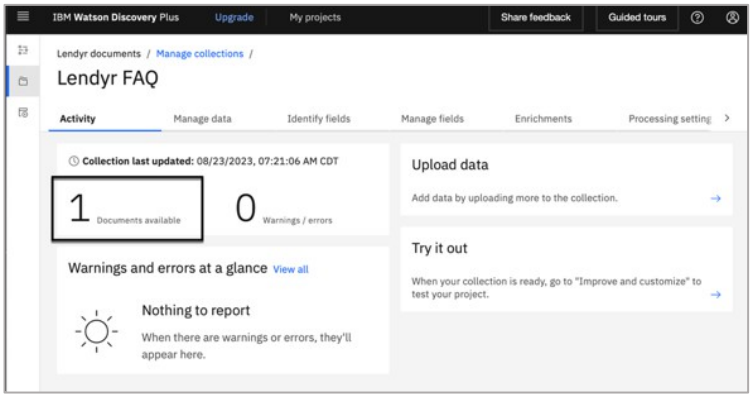
To get the project ID, select **Integrate and deploy (1)**, click the **API Information (2)** tab, and copy the **Project ID (3)**. Save it elsewhere.



Go back to the document collection, so you can continue to monitor its progress, by clicking
Manage collections (1) and **Lendyr FAQ (2)**:

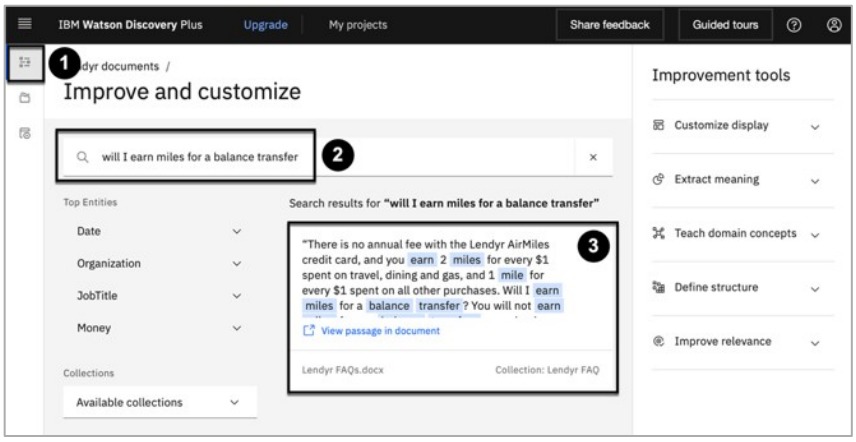


Once the screen says **1 Documents available**, you can proceed to the next step. This processing usually takes 5 to 15 minutes, so it may be a good time to take a short break.



Now that the processing is finished, let's ask Watson Discovery the same balance transfer question we asked ChatGPT at the beginning of this lab section: *will I earn miles for a balance transfer.*

To do this, click on **Improve and Customize (1)**, then type **will I earn miles for a balance transfer (2)** in the query window. Note the **search result (3)**:



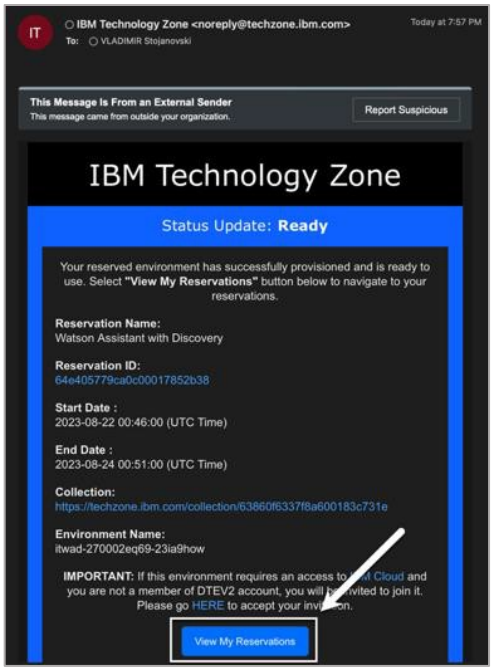
Watson Discovery provides a helpful answer, but not a conversational one; it's an excerpt from a business document that Watson Discovery read. Still, note that it is much more helpful and useful to the user ("You will not earn miles for any balance transfers …") than ChatGPT. This is because Watson Discovery understands the business content and context of an organization.

This lab will apply the power of generative language models to produce a more conversational and succinct answer to this question.

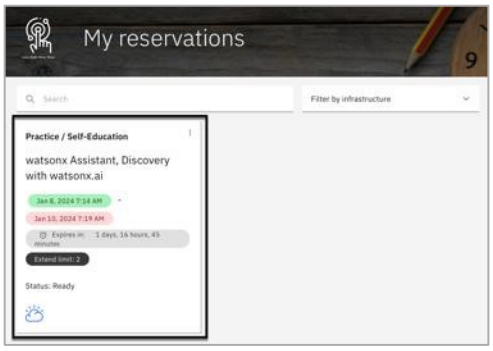
Set up the assistant (brand new instance)

Important: If you completed Section 6 of this lab series and still have your TechZone environment running, skip this part and go to the header, *Setup the assistant (existing instance)*.

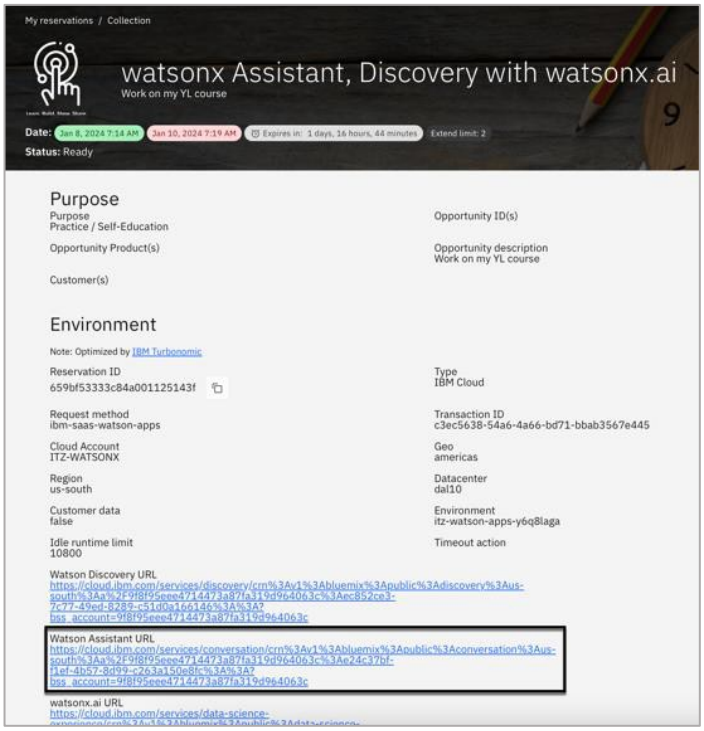
Go back to the email you received from TechZone, letting you know that your environment is ready. From that email, click on **View My Reservations**.



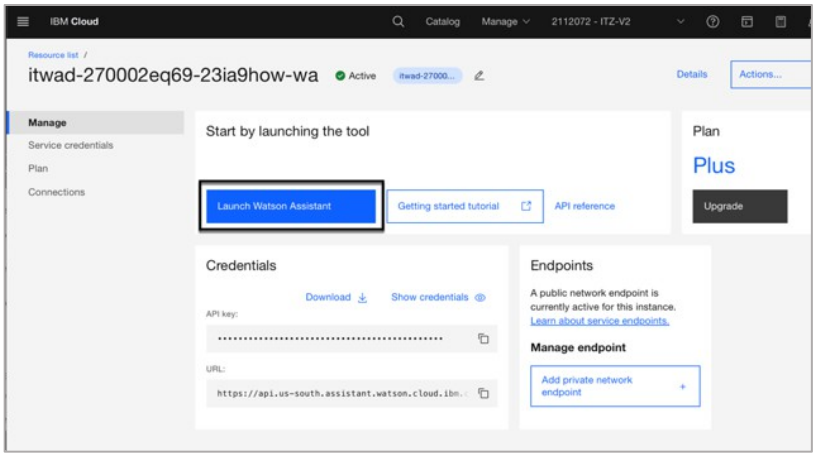
This takes you to the **My reservations** page on TechZone. Click your instance tile:



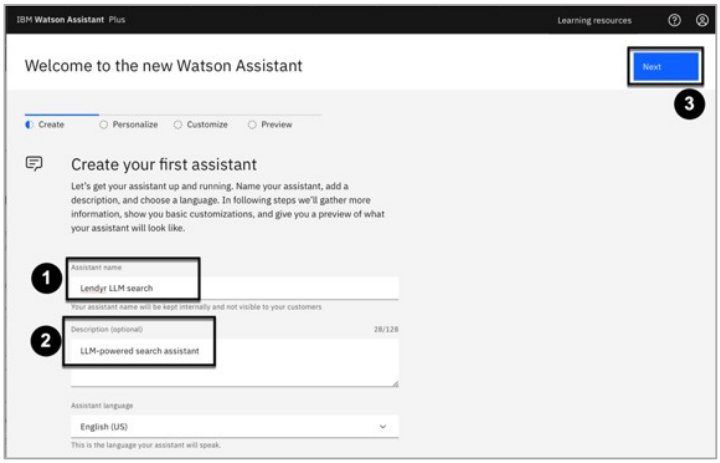
On your TechZone instance page. click the **watsonx Assistant URL**:



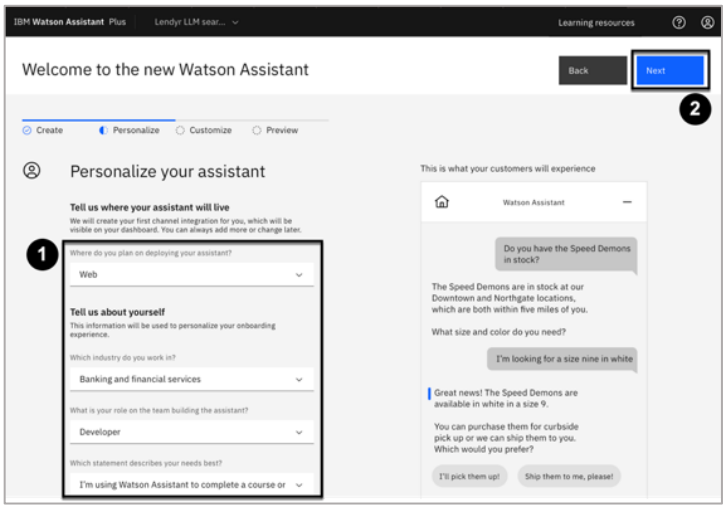
From the instance launch page, click **Launch watsonx Assistant**:



This is a new assistant, so you will first configure the basics. First, enter its **name (1)**, **description (2)**, and click **Next (3)**:



Fill out the form asking you how you will use the assistant **(1)** and click **Next (2)**:

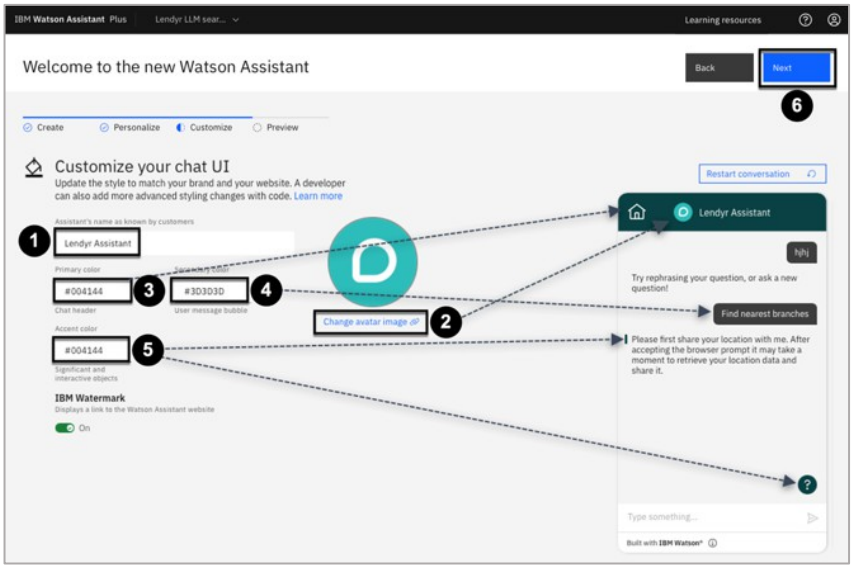


Customize your assistant to look like the Lendyr Bank website

The next screen allows you to customize the look and feel of the Assistant; customize it to look like the Lendyr Bank assistant (you will recall going through these steps in Section 2 of this lab):

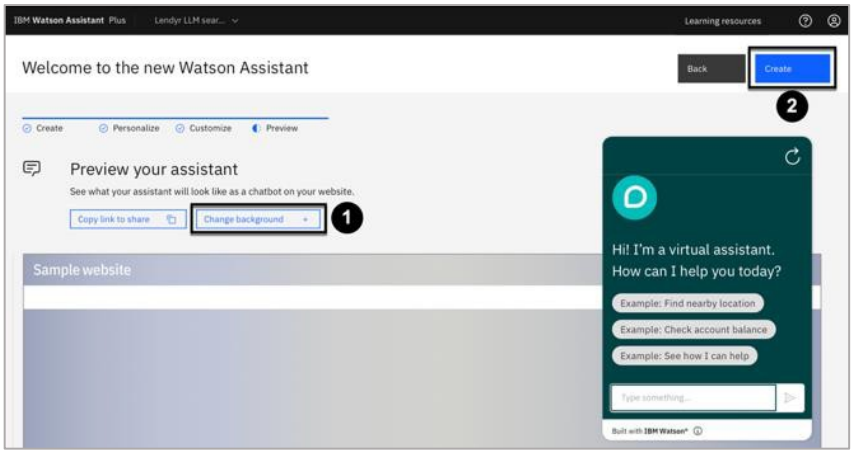
6. Change **Assistant's name as known by customers** to “Lendyr Assistant”.
7. Click on **Add an avatar image** and point to https://web-chat.global.assistant.watson.appdomain.cloud/assets/Lendyr-Avatar.png.
8. Change **Primary color** to: #004144. This is the color of the chat header.
9. Change **Secondary color** to: #3D3D3D. This colors the message bubble.
10. Change **Accent color** to: #004144. This is a tertiary color that accentuates

certain assistant responses and icons, as shown below.



Your assistant should now look like the image above. (Note, some color changes may require that you restart the chat.)

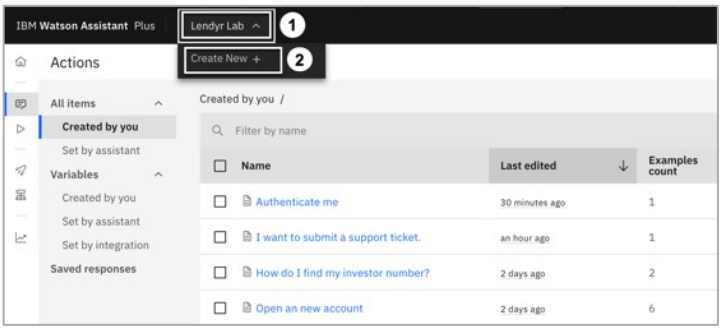
On the following preview screen, you have the option to **change the background (1).** We won't do that because we'll embed the assistant into the Lendyr site, but this is a great option to customize the look of an assistant so it feels like it is embedded in a your website. Click **Create (2)**.



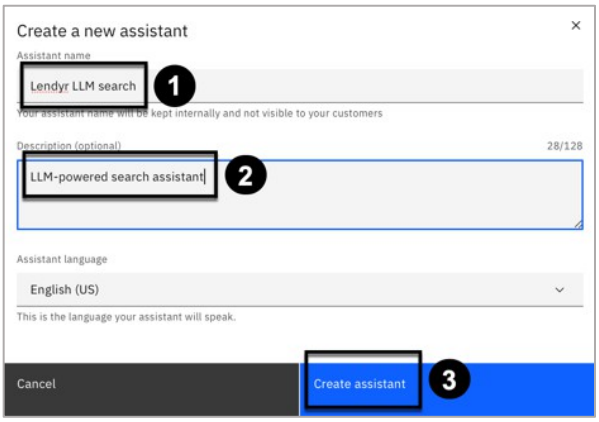
Important: Skip the next part, *Set up the assistant (existing instance),* and move directly to the header *Add the Watson Discovery extension.*

Set up the assistant (existing instance)

In your existing Lendyr Lab assistant, click on Lendyr Lab and Create New +. You will now create a new assistant for this lab section. Later, you can move this functionality into your original Lendyr Lab assistant.



This is a new assistant, so you will first configure the basics. First, enter its **name (1)**, **description (2)**, and click **Create assistant (3)**:

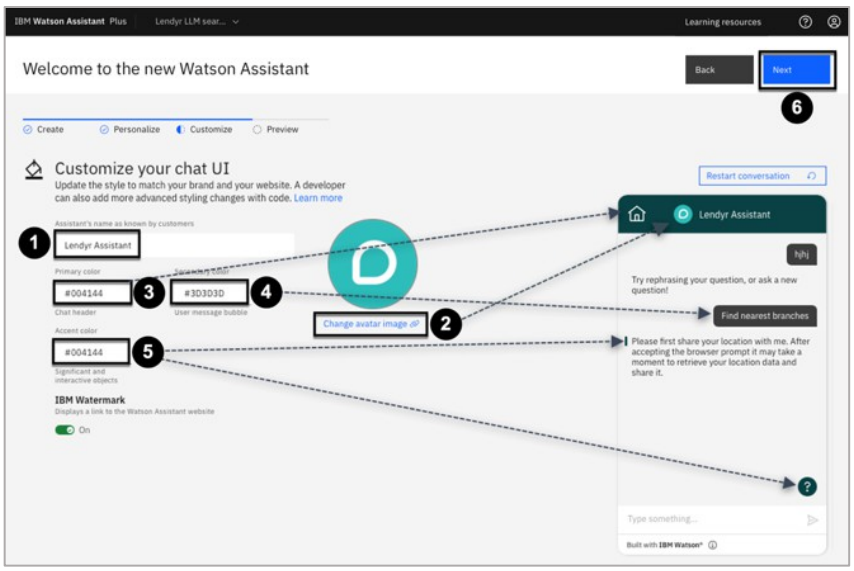


Customize the assistant to look like the Lendyr Bank website

Customize the look and feel of the Assistant to be like the Lendyr Bank assistant (you may recall going through these steps in Section 2 of this lab). If you are creating a new instance, the screen below will show automatically as part of the step-by-step wizard. If you are using an existing instance, you can find the look and feel options below by going to **Preview Customize webchat**:

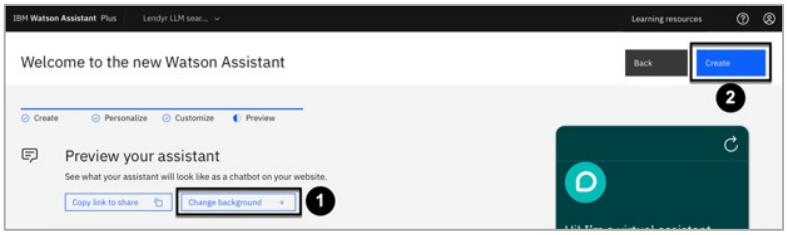
1. Change **Assistant's name as known by customers** to “Lendyr Assistant”.
2. Click on **Add an avatar image** and point to https://web-chat.global.assistant.watson.appdomain.cloud/assets/Lendyr-Avatar.png.
3. Change **Primary color** to: #004144. This is the color of the chat header.
4. Change **Secondary color** to: #3D3D3D. This colors the message bubble.

5. Change **Accent color** to: #004144. This is a tertiary color that accentuates certain assistant responses and icons, as shown below.

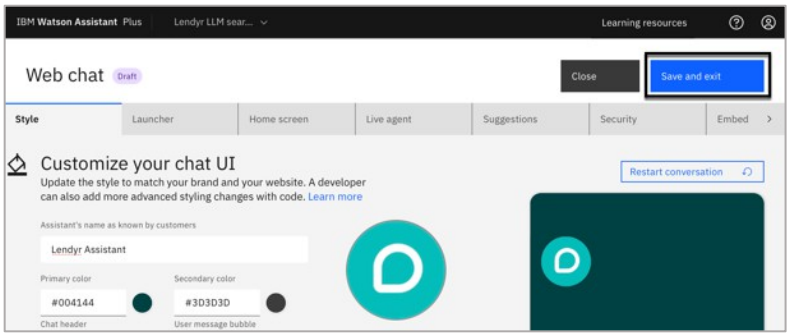


Your assistant should now look like the image above. (Note, some color changes may require that you restart the chat.)

If you are creating a new instance, you have the option to **change the background (1).** You won't do that because you'll embed the assistant into the Lendyr site, but this is a great option to customize the look of an assistant so it feels like it is embedded in a your website. Click **Create (2)**.



Otherwise, you are using an existing instance, click **Save and exit:**



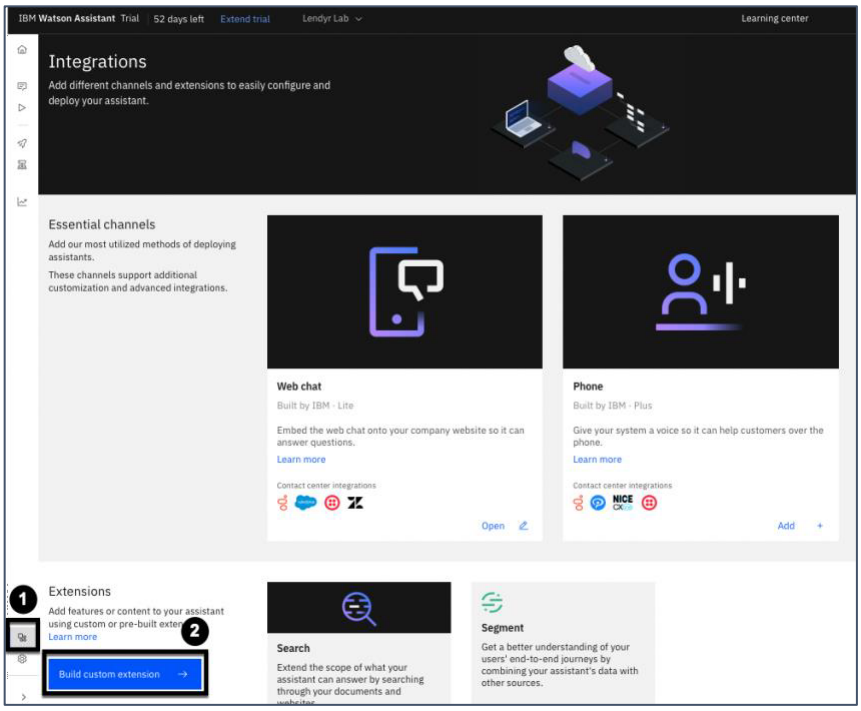
Add the Watson Discovery extension

Now that you have a Watson Discovery instance setup, a watsonx project ready, and an Assistant configured, you can proceed to add the custom extensions to your assistant.

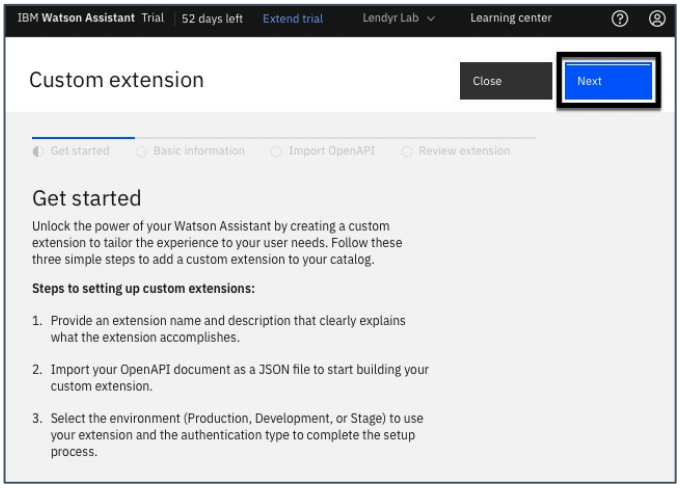
Note: Extensions were covered in detail in Section 5 of this lab.

For the Watson Discovery extension, download the OpenAPI specification watson-discovery- query-openapi.json. This is the API specification for the Watson Discovery custom extension.

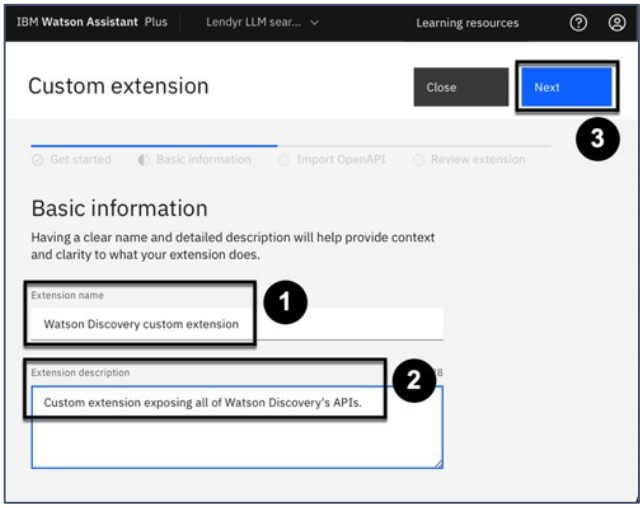
In watsonx Assistant, use the left menu to open the **Integrations (1)** page. Then, scroll down and click the **Build custom extension (2)** button:



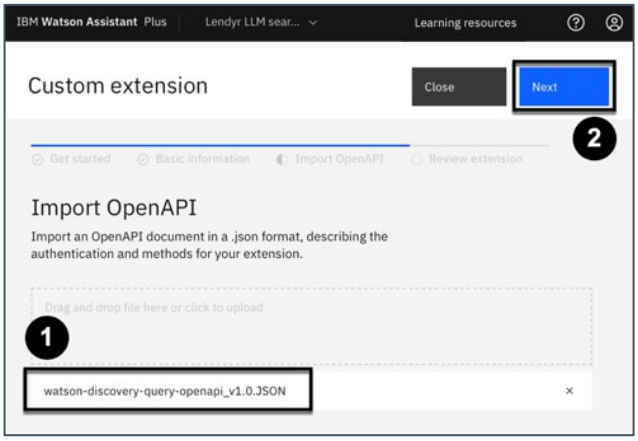
This first screen describes what you'll need to create the extension. Note that the OpenAPI JSON document, which you just obtained, is key to the setup. Click **Next** to proceed to the next screen:



The second screen asks you to name and describe the custom extension. Name the custom extension **Watson Discovery custom extension (1)** and add a description, like “Custom extension exposing all of Watson Discovery’s APIs” **(2)**. Click **Next (3)** to proceed to the next screen.

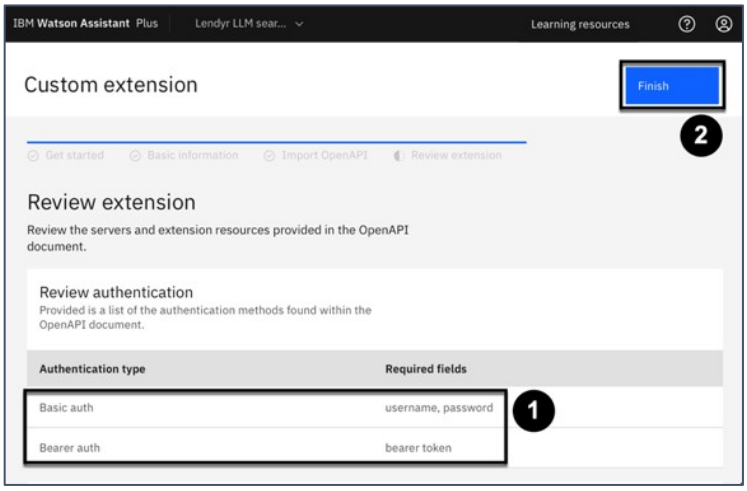


In the next screen, either drag-and-drop or click to upload the watson-discovery-query- openapi.json OpenAPI spec you downloaded **(1)**, then click **Next (2)** to proceed to the next screen:

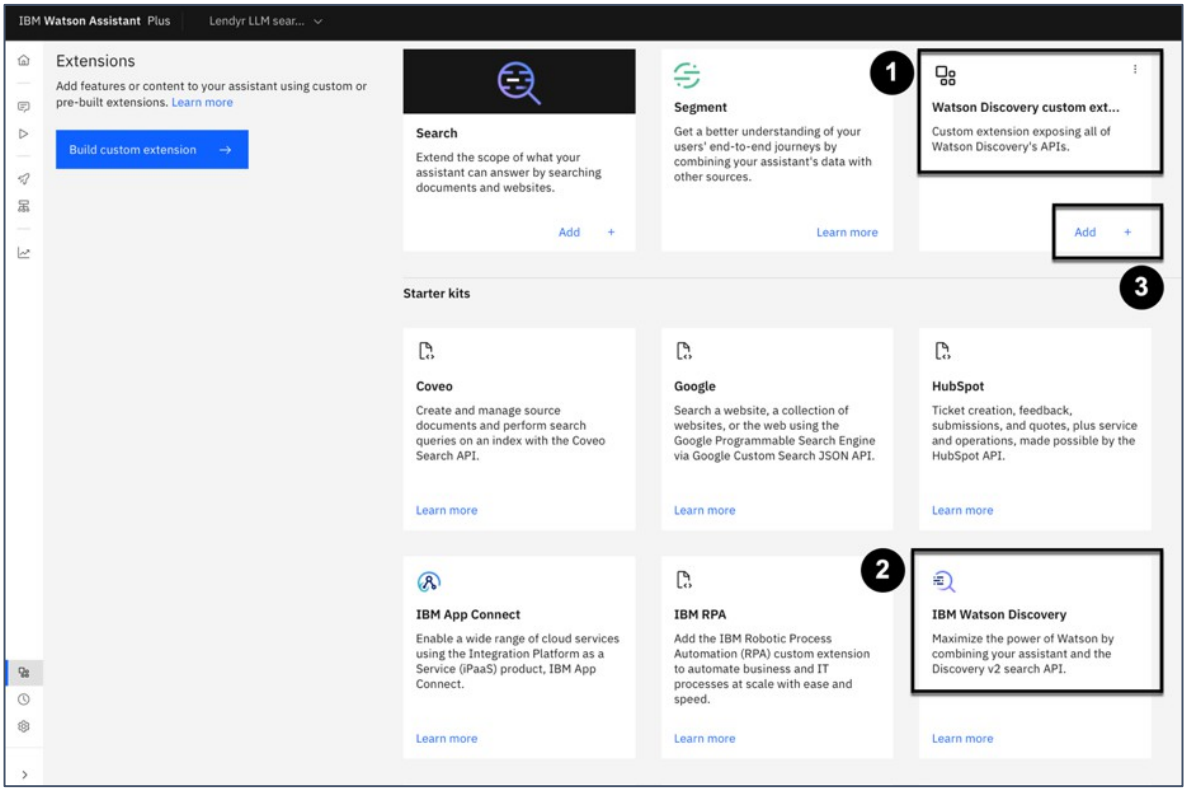


Note: If you get an error that the file is not in the proper JSON format, try downloading it again with another browser, and assure it has a **.json** extension.

Take a moment to review the extension on the following screen. This extension will allow us to call Watson Discovery by using its **API Key (1)** which you obtained earlier. Click **Finish (2)**.

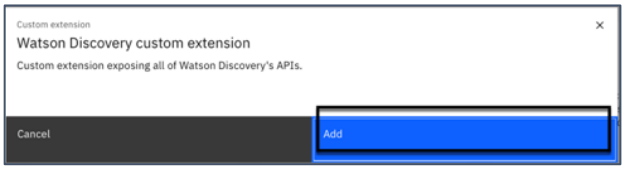


You should now be able to see the **Watson Discovery custom (1)** extension in your Integrations catalog:

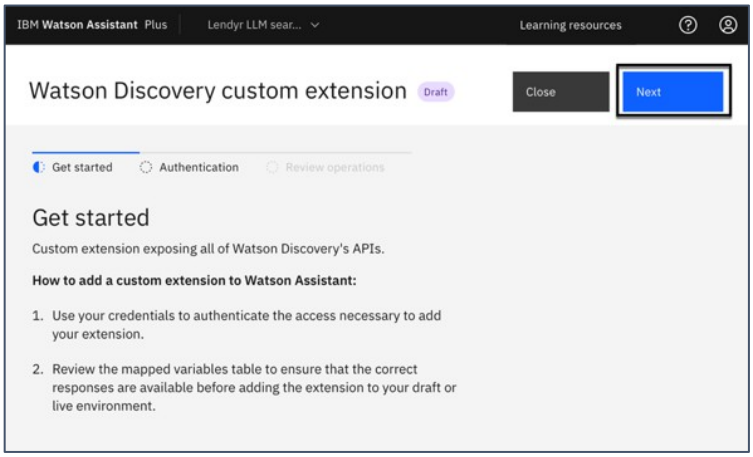


Note that it is different from the **IBM Watson Discovery (2)** extension, which is normally used as Assistant's search skill. The custom skill gives watsonx fuller access to Watson Discovery's APIs than what the search skill provides. Click **Add + (3)**.

You have created the Watson Discovery custom extension, and now you need to specify which Watson Discovery instance it will access. On the popup window, click **Add**.

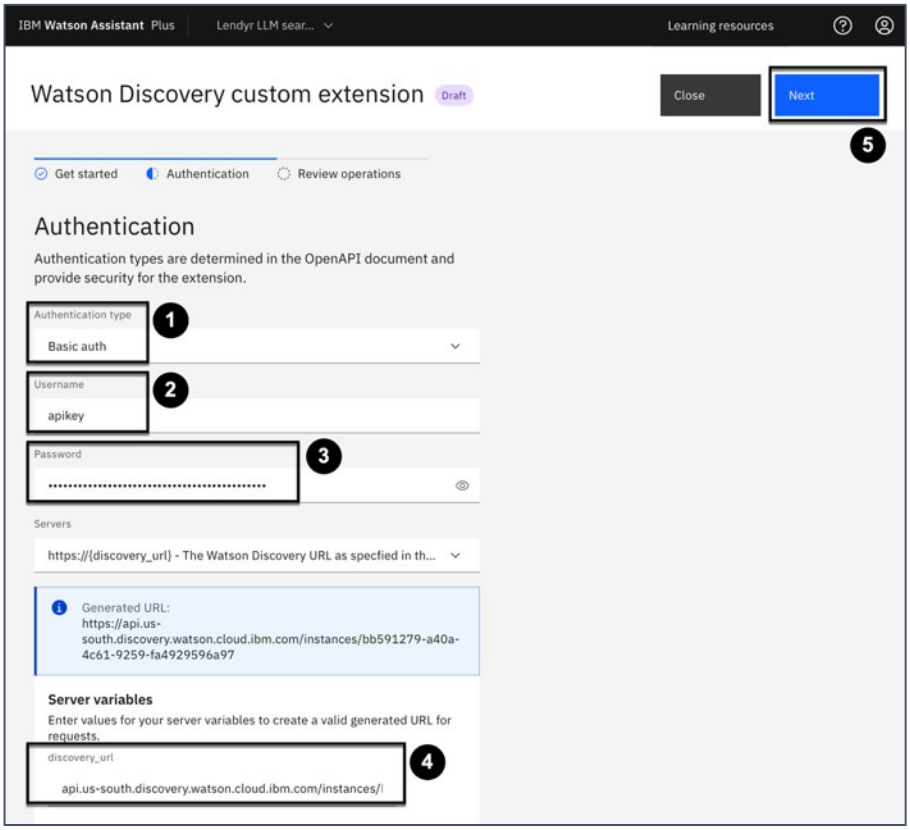


On the following screen, you can read about how to add the custom extension. When ready, click
Next:

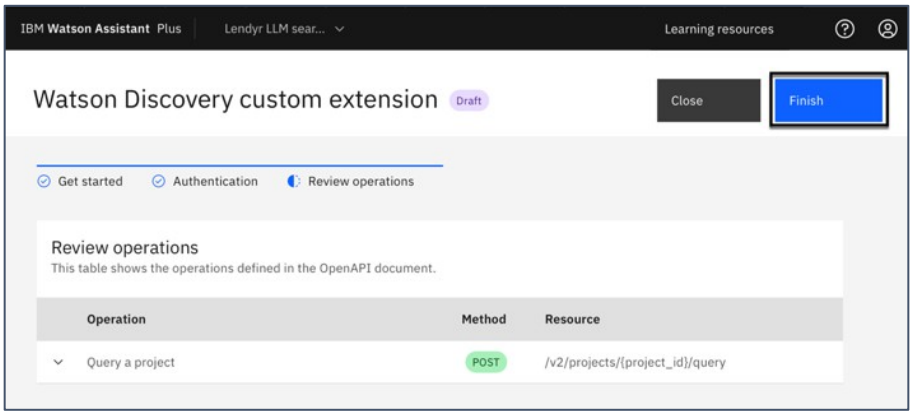


The following screen asks for **Authentication** details:

1. Select **Basic auth** as the authentication type.
2. In the **Username** field, enter **apikey**.
3. In the **Password** field enter the **Discovery API Key** you saved earlier.
4. At the bottom of the Authentication pane, under **discovery_url**, fill in the URL for your Discovery instance which you saved earlier. Make sure you enter it without
   the *https://* prefix.
5. Click **Next (5)**.



You'll see a summary view of your new custom extension. Click **Finish**:

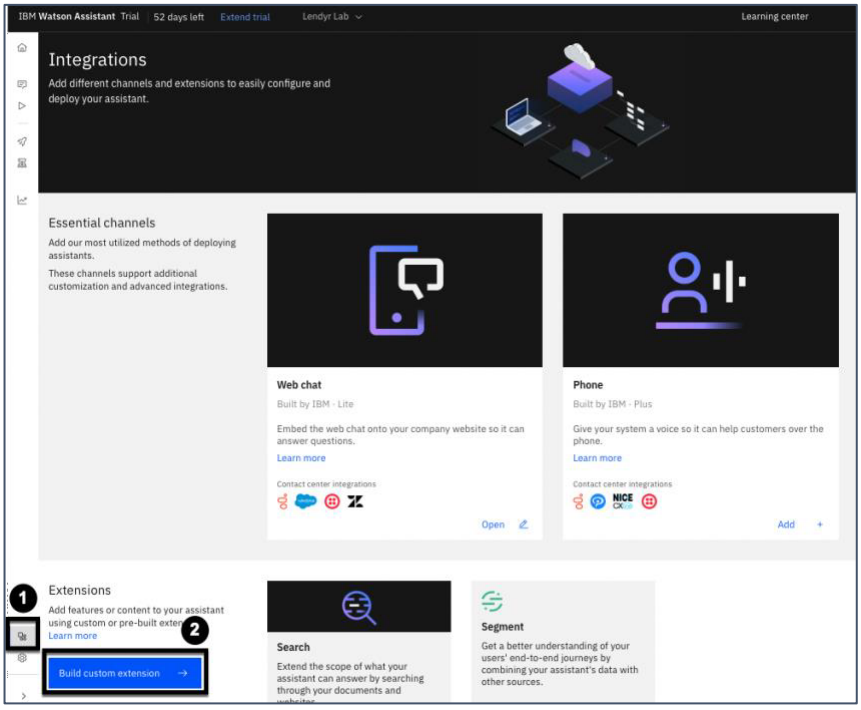


Add the watsonx custom extension

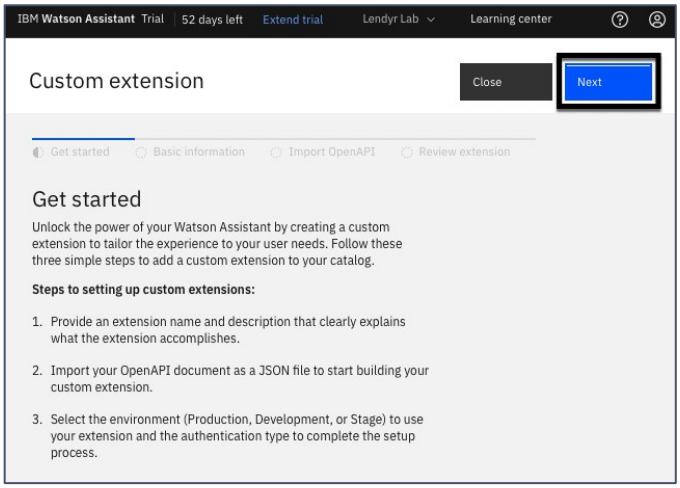
Next, you will create the watsonx custom extension. First, download the watsonx OpenAPI specification file. This JSON file defines the watsonx custom extension.

In watsonx Assistant, use the left menu to open the **Integrations (1)** page. Then, scroll

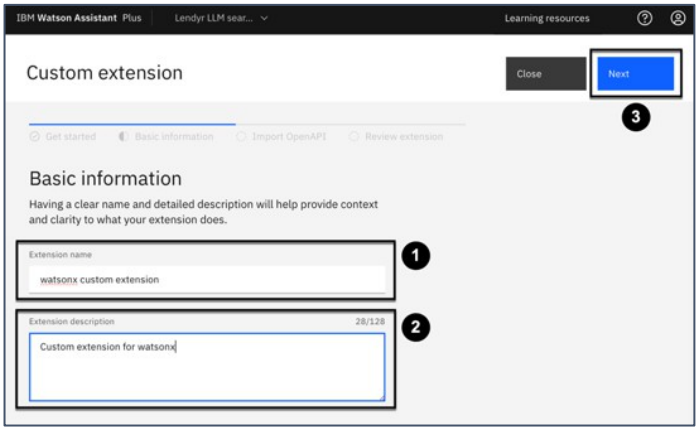
down and click the **Build custom extension (2)** button:



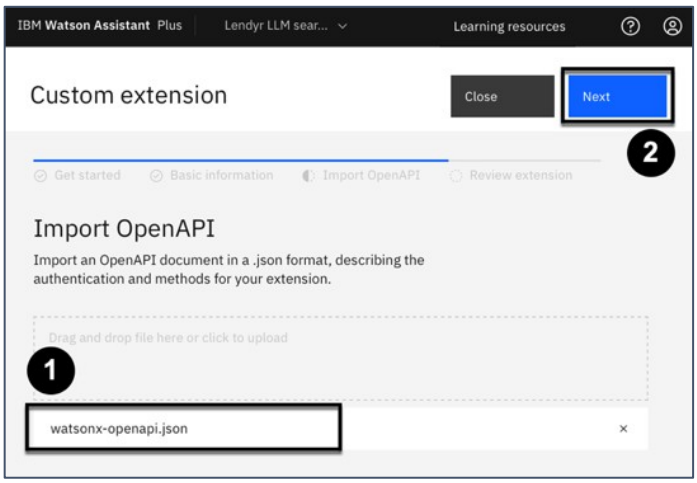
This first screen describes what you'll need to create the extension. Note that the watsonx OpenAPI JSON document, which you just obtained, is key to the setup. Click **Next** in the top right to proceed to the next screen:



The second screen asks you to name and describe the custom extension. Name it **watsonx custom extension (1)** and add a description, like "Custom extension for watsonx" **(2)**. Click **Next (3)** to proceed to the next screen.

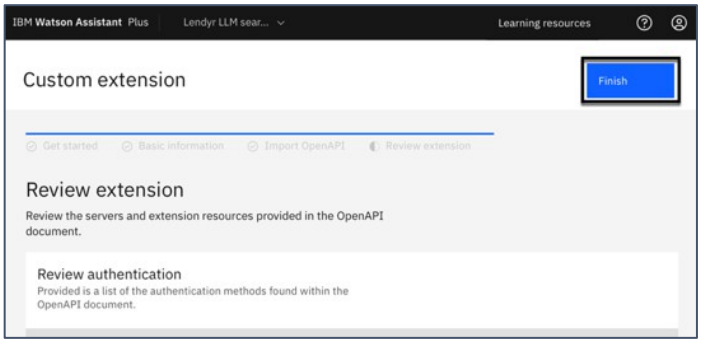


In the next screen, either drag-and-drop or click to upload the [watsonx OpenAPI specification file] **(1)** OpenAPI spec you downloaded, then click **Next (2)** to proceed to the next screen:

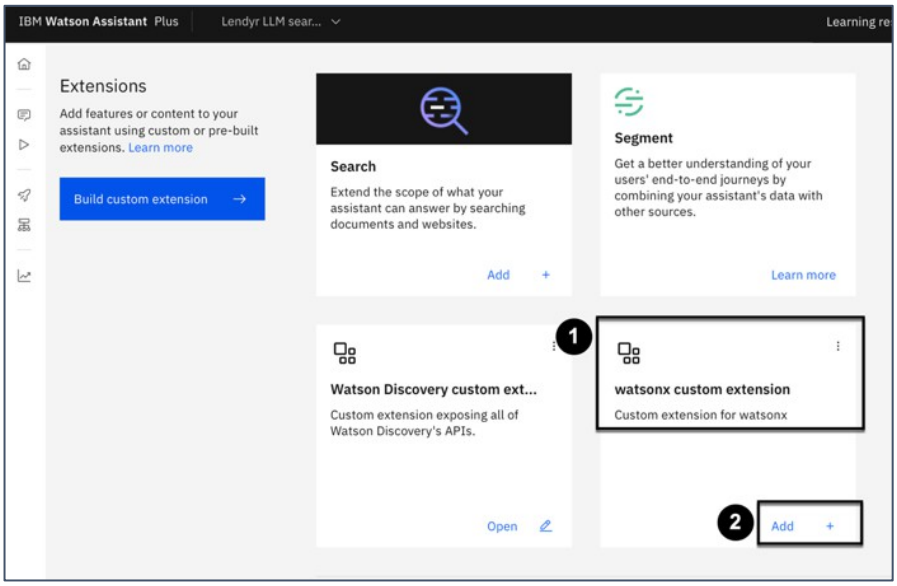


Note: If you get an error that the file is not in the proper JSON format, try downloading it again with another browser, and assure it has a **.json** extension, in lower case.

Click **Finish** to create the custom extension.

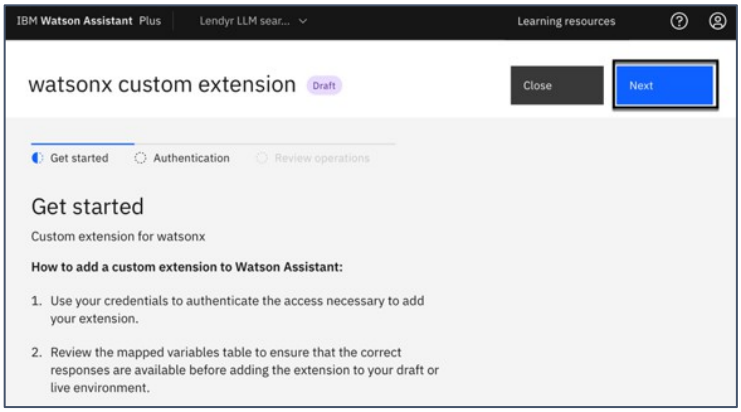


You should now be able to see the **watsonx custom extension (1)** in your Integrations catalog. Click **Add + (2)** so that you can configure a connection to your watsonx project:



Click **Add** on the popup screen.

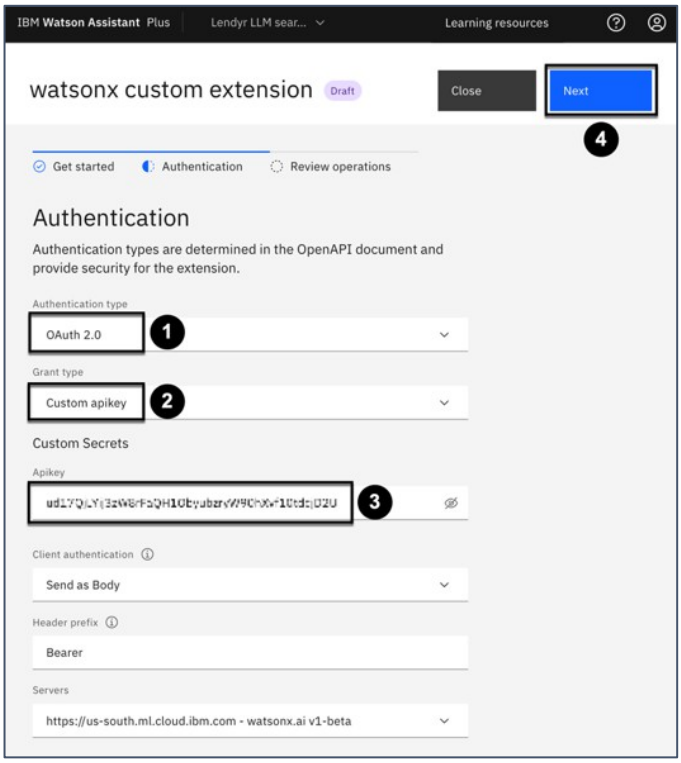
Hit **Next** on the following screen:



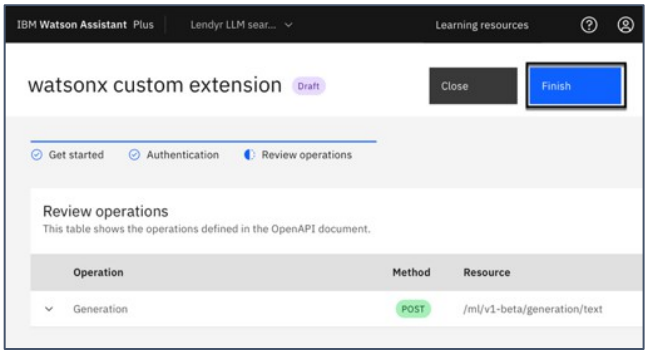
On the Authentication screen,

1. Choose **OAuth 2.0** as the Authentication type.
2. Select **Custom apikey** as the Grant type.
3. Copy and paste the **IBM Cloud API key** you saved earlier into the Apikey field.

4. Click **Next**.



Click **Finish**.



Click **Close** on the final review screen.

Upload and configure the watsonx actions

Next, you will upload the actions your assistant will need, so download the actions JSON file.

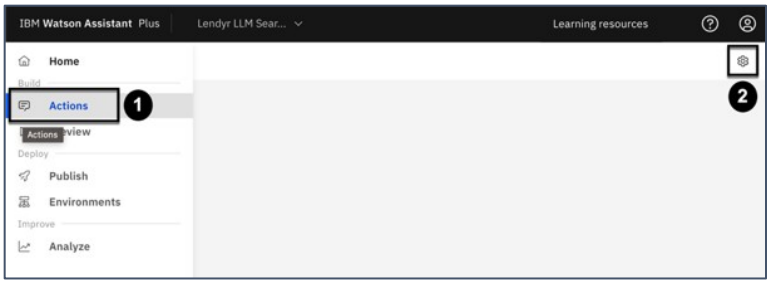
Notes: 1) You should not upload these actions directly into an *existing* assistant because doing so will overwrite your existing actions. 2) If the file downloads with an uppercase **.JSON** extension, you must rename it to lowercase **.json**, otherwise it will not upload correctly.

This file contains three actions:

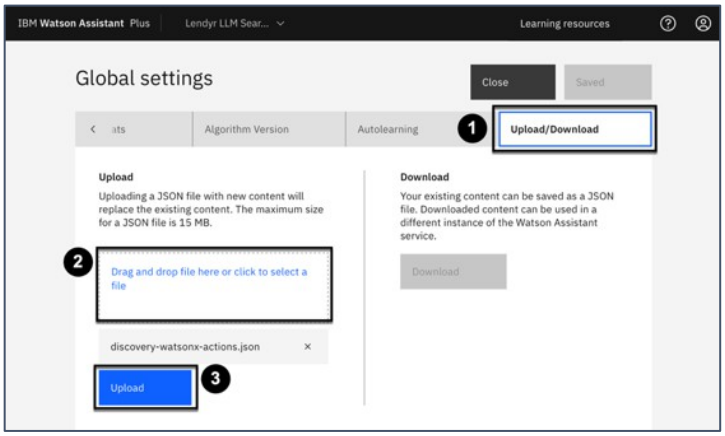
Action	Description
Search	Connects to Watson Discovery to search for documents related to the user query. The out-of-box "No Action Matches" action has been configured to send all input to this action, so whatever the user enters will be used as the search input. In turn, this action invokes the "Generate Answer" action to generate a response to the query.
Generate Answer	Configures the query prompt and document passages resulting from the Search action and calls the action "Invoke watsonx generation API." It is not meant to be invoked directly, but rather by the "Search" action.
Invoke watsonx generation API	Connects to watsonx and, using as context the documents resulting from the search, asks the language model to generate an answer to the user query. It is not meant to be invoked directly, but rather by the "Generate Answer" action.

The actions in this file will use search only when no action matches the user request. They search a complete watsonx project, and as such they are general-purpose and usable with any data set.

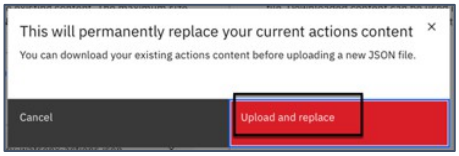
To upload the JSON file, click **Actions (1)**, and then **Global Settings (2)**:



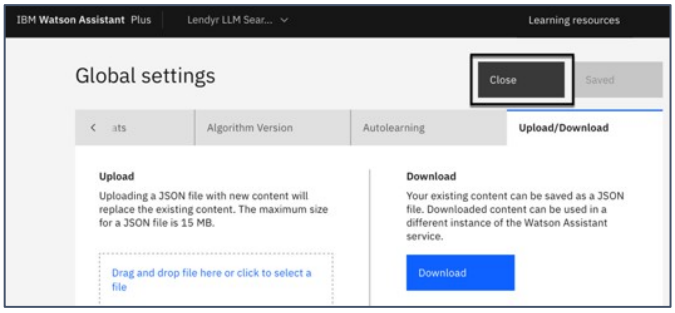
Scroll to the right until you are able to see and select the **Upload/Download (1)** tab. There, **drag and drop (2)** the actions JSON file, and click **Upload (3)**:



Click **Upload and replace** at the warning pop-up screen.

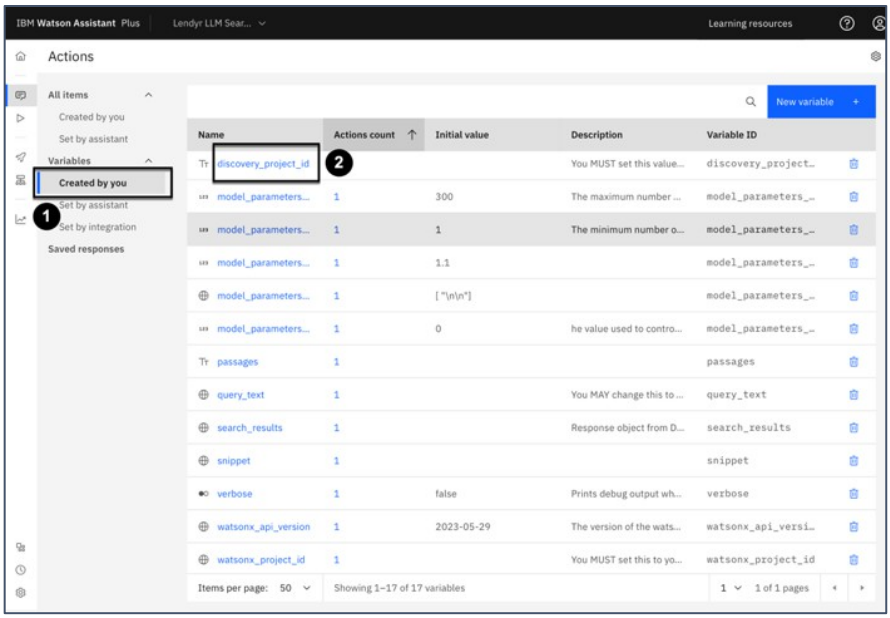


Hit **Close**.

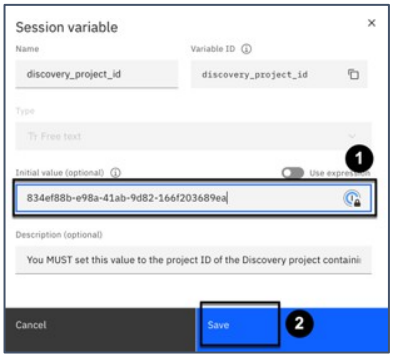


The actions JSON file you uploaded also includes variables used by the actions. You need to update two of them with your Discovery and watsonx project IDs.

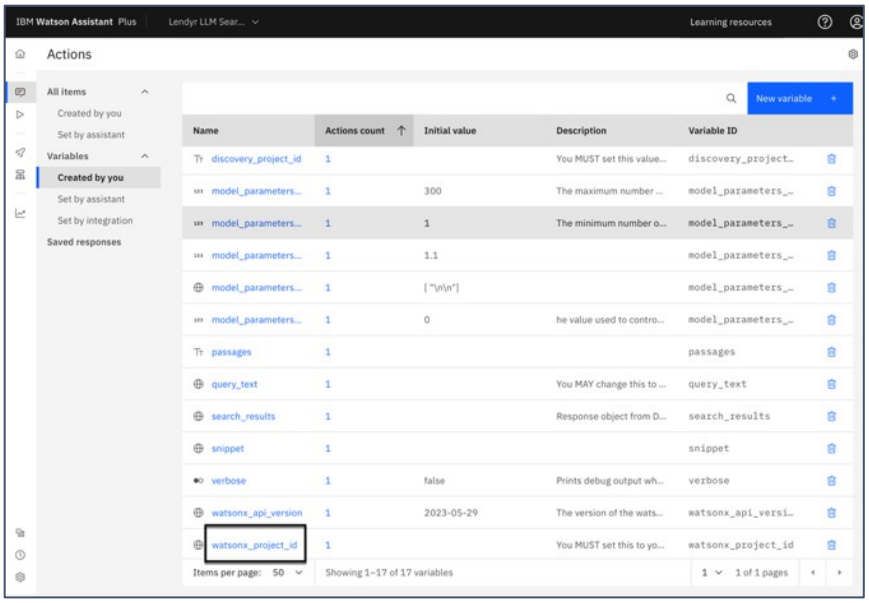
Within the Actions page, under Variables, select the **Created by you (1)** filter, which displays the variables you just uploaded. Click the **discovery_project_id (2)** session variable:



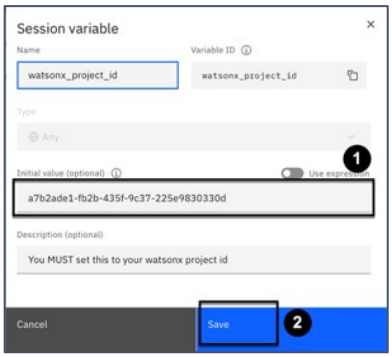
Copy and paste the **project ID (1)** value you obtained when configuring Watson Discovery and click **Save (2)**.



Next, click the **watsonx_project_id** variable:



Set the **watsonx_project_id (1)** variable to the watsonx project ID you obtained earlier and click
Save (2). This tells the assistant which watsonx project will be used for answer generation.



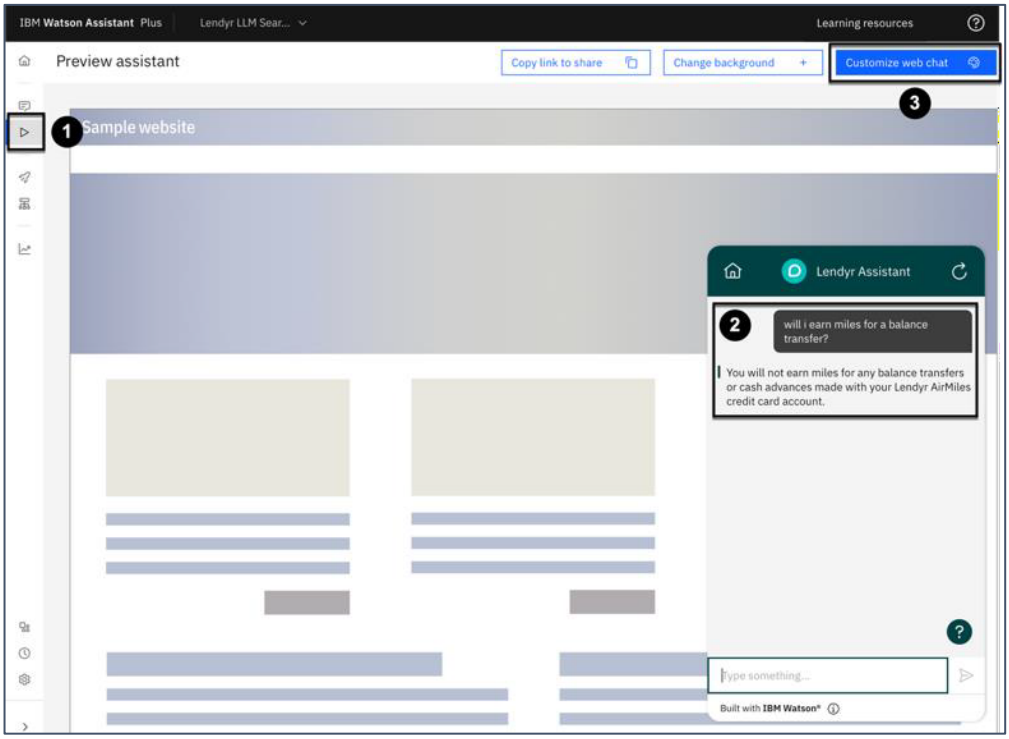
Those are all the variables you must set.

This list describes the available variables providing for greater control over your model:

- discovery_date_version: Discovery date versions are documented in the release notes.
- model_id: The id of the watsonx model that you select for this action. Defaults to meta- llama/llama-2-70b-chat.
- model_input: The input to the watsonx model. You may change this to do prompt engineering, but a default will be used by the model if you don't pass a prompt here, in Step 5 of the "Generate Answer" action
- model_parameters_max_new_tokens: The maximum number of new tokens to be generated. Defaults to 300.
- model_parameters_min_new_tokens: The minimum number of the new tokens to be generated. Defaults to 1.
- model_parameters_repetition_penalty: Represents the penalty for penalizing tokens that have already been generated or belong to the context. The range is 1.0 to 2.0 and defaults to 1.1.
- model_parameters_stop_sequences: Stop sequences are one or more strings which will cause the text generation to stop if/when they are produced as part of the output. Stop sequences encountered prior to the minimum number of tokens being generated will be ignored. The list may contain up to 6 strings. Defaults to []. For some models, you may want to set this to ["\n\n"] (i.e., stop whenever the model produces a double newline) because that can be a reliable indication that the model is shifting to a new topic. However, that is generally unhelpful for the chat model being used in this example (Llama 2).
- model_parameters_temperature : The value used to control the next token probabilities. The range is from 0 to 1.00; 0 makes it deterministic.
- model_response: The text generated by the model in response to the model input.
- passages: Concatenation of top search results.
- query_text: You may change this to pass queries to Watson Discovery. By default the Search action passes the user's input.text directly.
- search_results: Response object from Discovery query.
- snippet: Top results from the Watson Discovery document search.
- verbose: A boolean that will print debugging output if true. Default is false.
- watsonx_api_version - watsonx api date version. It currently defaults to 2023-05-29.
- watsonx_project_id: You must set this value to be a project ID value from watsonx.

Try it out!

Now that you are finished configuring your assistant, try it out! Click on **Preview (1)** and enter the same question as before: **will I earn miles for a balance transfer (2)**. Note the clear, concise, and conversational answer. Compare this to the generic answer from ChatGPT and the excerpt answer provided by Watson Discovery earlier!

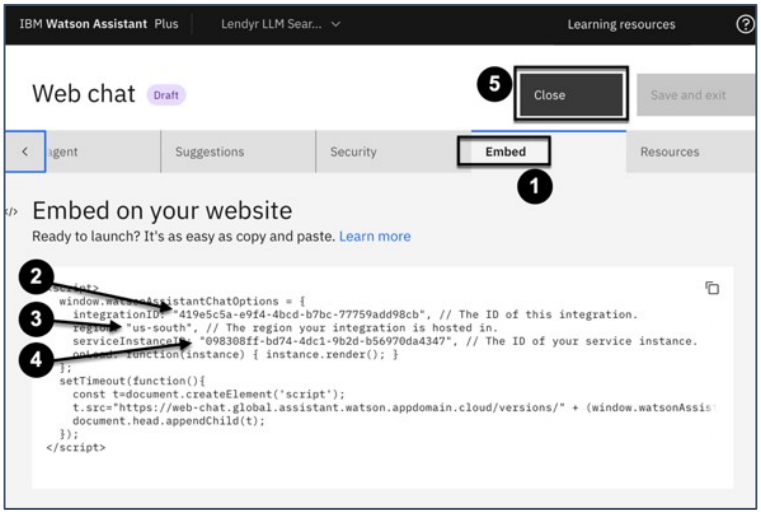


Preview your assistant on Lendyr Bank's website

You embedded your assistant on the Lendyr website in Section 2, so these directions may seem familiar.

On the **Preview** page (shown above), click **Customize web chat (3)**. Then, as shown below,

1. Click on the **Embed** tab.
2. Copy the value for your assistant's **integrationID** and paste it into a text document. You will use this value shortly.
3. Copy the value for your assistant's **region** and paste it into a text document. You will use this value shortly.
4. Copy the value for your assistant's **serviceInstanceId** and paste it into a text document. You will use this value shortly.
5. Finally, click **Close**.



Now, take your assistant's **integrationID**, **region**, and **serviceInstanceId**, and plug them into this URL:

https://www.ibm.com/products/watson-assistant/demos/lendyr/demo.html?integrationID=ID_HERE&region=REGION_HERE&serviceInstanceID=ID_HERE

Make sure there are no spaces (each paste adds a space!), and then copy and paste your custom URL in your browser.

If you would like to demo the capabilities built in this section, review the demo video for a recommended flow with other utterances to ask the assistant.

Congratulations! You have now built the demo and functionality shown in the demo video.

Copy the new functionality to your Lendyr assistant (optional step)

Note: There are two pre-requisites to being able to complete this last part of the lab. If both of these do not apply, simply skip to the last page of this document:

1. You must have completed sections 1 through 7 of the lab; and
2. Both assistants must be running in the same instance of watsonx Assistant (in other words, either under the same TechZone environment, or under the same instance under the same IBM Cloud user ID).

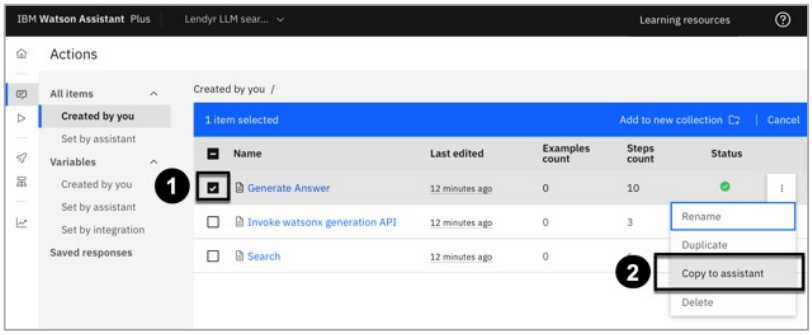
You can copy the LLM-powered search functionality you just created into your version of the Lendyr virtual assistant which you built in the previous lab sections.

Do not upload the actions to the assistant you previously built; this will overwrite your existing actions. Rather, you will copy-and-paste the actions.

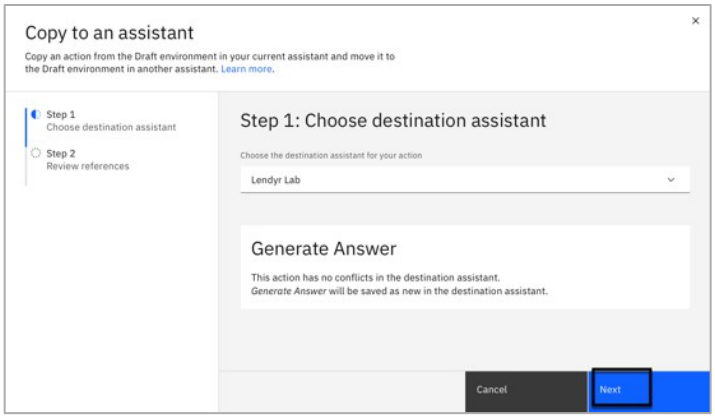
First, navigate to your Lendyr Lab assistant, built in sections 1-6. There, reconfigure the watsonx and Watson Discovery extensions by following the respective parts of this section:

- *Add the Watson Discovery extension*
- *Add the watsonx custom extension*

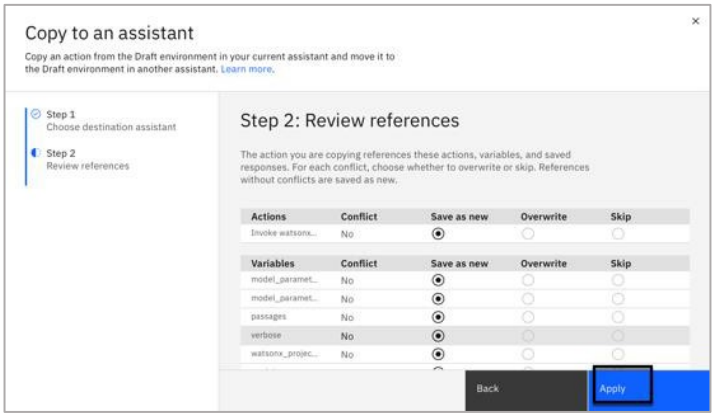
Then, navigate back to your new (LLM-powered) virtual assistant's **Actions** page. **Select (1)**
each of the three actions, one at a time, and select **Copy to assistant (2)**:



On the popup screen, check to see that you are pasting your Lendyr Lab assistant, and click
Next:



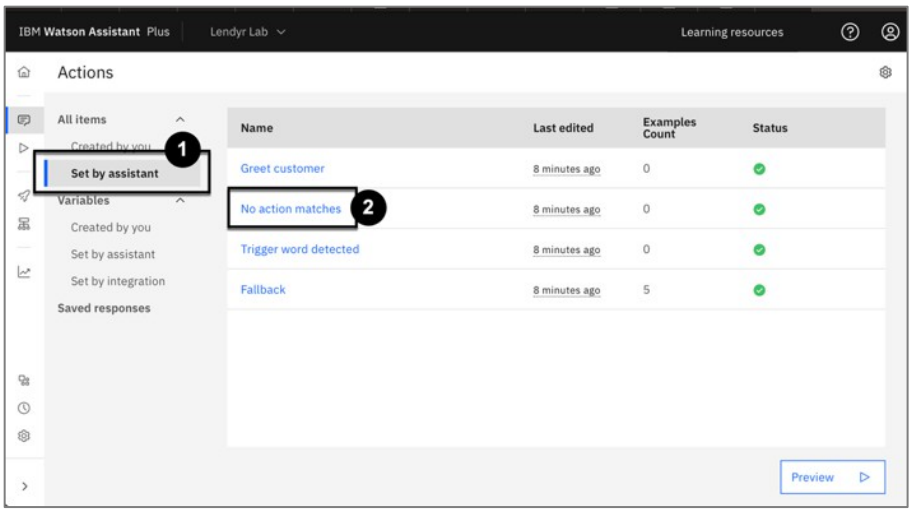
On the next screen, note that there are no "conflicts" and click **Apply**:



The virtual assistant may ask you to select *Overwrite* or *Save as new* for actions and variables. It will have default selections for each action and variable. Do not change the default selections.

Now that you have copied the actions created in this (Section 7) of the lab to the Lendyr assistant you created in Sections 1-6, in your Lendyr virtual assistant, navigate to the **Actions** page.

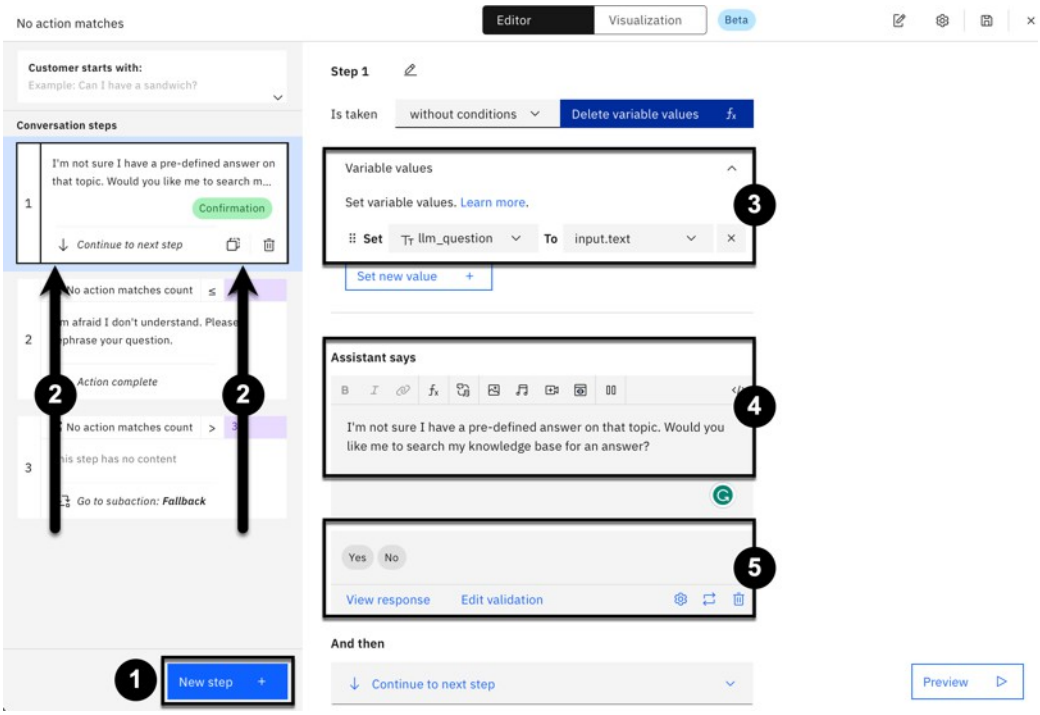
Here, you will create an action that will ask the user if they want to try conversational search when the assistant does not recognize the question. To build this action click **Set by assistant (1)**, and then open the **No action matches (2)** action:



As shown below,

1. Add a **New step +**

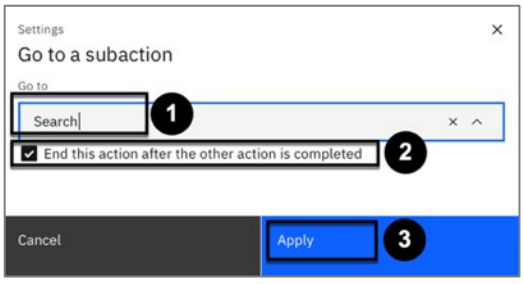
2. Drag it up so it’s the first step
3. Save the original question (**input.text**) to the **llm_question** variable so that you can later pass it to watsonx.ai
4. Under **Assistant says**, write: “I'm not sure I have a pre-defined answer on that topic. Would you like me to search my knowledge base for an answer?”
5. **Define the customer response** as a Confirmation.



Next,

1. Add another **New step +**
2. Drag it up so it's the second step
3. Add the **condition**: If Step 1 is Yes
4. Under **Assistant says**, write: "OK!"
5. Change **And then** to **Go to a sub-action**

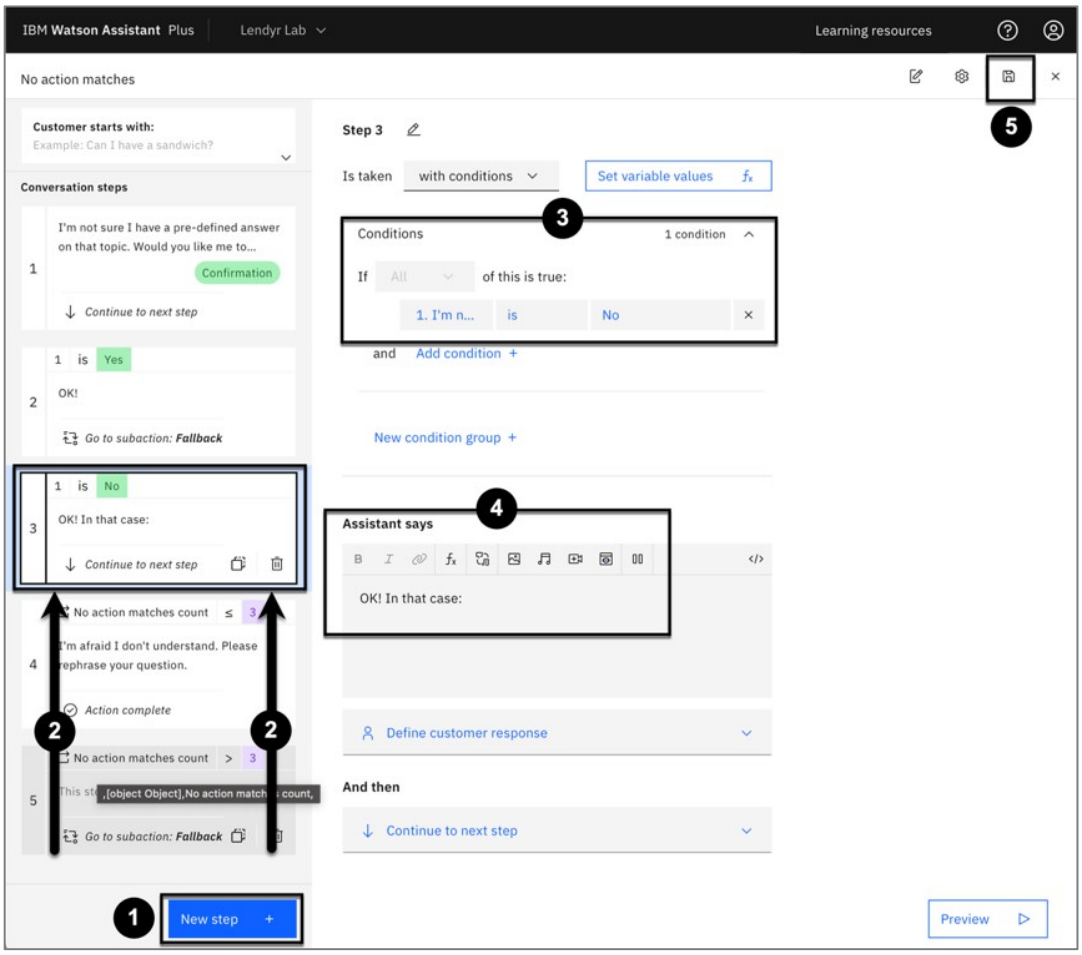
When the Go to a subaction search window appears, select **Search (1)** as the subaction, check the **End this action after the other action is completed (2)** checkbox, and click **Apply (3):**



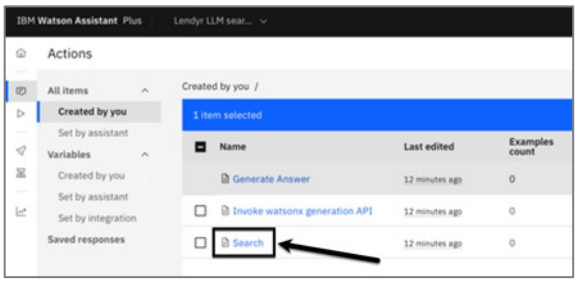
Finally,

1. Add another **New step +**
2. Drag it up so it's the third step
3. Add the **condition**: If Step 1 is No
4. Under **Assistant says**, write: "OK! In that case:"

5. **Save** the action.



There is one final step required. Save and exit the **No action matches** action, and open the
Search action:



In the Search action, set **query_text** to the variable **llm_question** you just created. This will ensure that the input question is passed to watsonx.ai.

Your Lendyr virtual assistant will now ask the end user if they want to try conversational search when the virtual assistant does not recognize their question.

Continue your learning

You have now completed this section of the lab, which also concludes this lab series. **You** have covered the fundamental concepts and features of watsonx Assistant, and are well-equipped to build and test a simple, powerful virtual assistant – with some practice, of course!

IBM clients that are interested in deeper learning should discuss education options with their IBM or IBM business partner representative.