IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide

Description	IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide
Author(s)	Andrew R. Jones (andrewj@us.ibm.com) and Max Weiss (maxwell.g.weiss@ibm.com)
Copyright	Copyright © 2024 IBM - V 2.0

Table of Contents

- Verify the document that is ingested is now returned as a source file for a query

Adding skills to the assistant

Getting started with skills and actions

- Environments
  - Watsonx Orchestrate
  - Ansible Automation Platform and Wazi as a Service

Explore Ansible Automation Platform

- Access the

Importing skills from Ansible Automation Platform

- Import skills into your assistant

Connecting skills to your assistant

Creating actions for your assistant

Configure the number of input fields

- Create actions

Verify the job in the Ansible Automation Platform console

- Troubleshooting

Creating skill flows

- Add the utility skill
- Add the skills to your Personal skills
- Create the skill flow
- Enable the skill flow in your assistant

Creating custom-built actions

Importing pre-packaged z/

Publishing and deploying your assistant

- Publish the assistant
- Configure the live environment
- Connect the skills to the live environment
- Deploy the assistant

Adding other integrations

Next steps

- Additional resources
- Earn the badge

Welcome

Welcome to the IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide (lab guide). The lab guide is part of the IBM watsonx Assistant for Z for Technical Sales Level 4 learning plan for IBM and Business Partner Technical Sales and related badge. The learning plan is intended to teach technical sellers and Business Partners how to conduct a proof of experience (PoX) for a client.

> **In a fast-paced world, things change**
>
> The products and services may appear differently than what is shown in the lab guide. This can occur if the product or service is updated with a new version.
>
> Responses generated by IBM watsonx Assistant for Z are likely to change over time. The responses you see when you run the queries in this section may differ from the screen images captured in the lab guide.

This lab guide covers the setup, configuration, and usage of watsonx Assistant for Z. This lab guide uses the IBM watsonx Assistant for Z Velocity collection and the three Velocity Pilot lab environments in IBM Technology Zone (ITZ).

The lab guide also enables dedicated lab environments for customized client PoXs and demonstrations. If you are preparing for an actual pilot engagement, refer to the Pilot Scoping Guide for watsonx Assistant for Z for additional information.

The lab guide provides the following guidance to:

- Provisioning the lab environments
- Creating an assistant and configuring conversational search
- Configuring assistant settings
- Testing conversational search
- Deploying a dedicated instance of OpenSearch for client document ingestion (Optional)
- Importing skills for z/OS automations
- Connecting apps to assistants
- Creating assistant actions
- Creating skill flows
- Learn about custom-built actions
- Learn about Importing pre-packaged z/OS skills
- Publishing and deploying your assistant

Not all capabilities of the offering are covered in the lab guide.

This lab guide covers many features and capabilities of IBM Watson Assistant for Z, but not all. Some uncovered capabilities may be available in ITZ environments, while others may not, such as using skills for OMEGAMON.

Support

Think something is down? Check the applicable status pages for any known issues such as a site or service not being available:

- IBM Technology Zone status

For issues with provisioning the ITZ environment for this lab (for example, a failed reservation request due to insufficient quota capacity), open a ticket with ITZ support:

- Web: IBM Technology Zone ticket system
- Email: techzone.help@ibm.com

For issues related to specific steps found in the demonstration guide after the ITZ environment is provisioned, contact the authors:

- Slack:
- #watsonx-assistant-z-technical - IBM only
- #wxo-practitioners - IBM only - for questions that are related to the software as a service (SaaS) instance of watsonx Orchestrate
- Email: andrewj@us.ibm.com and maxwell.g.weiss@ibm.com

Business Partners, use the IBM Training live Chat Support service or other support methods that are found on the IBM Training portal here.

Using the demonstration guide

Use these helpful tips to take full advantage of the IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide.

Printing the demonstration guide

Printed or saved copies can be out of date

The IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide changes regularly to match the IBM watsonx Assistant for Z offering and associated ITZ environment. Printed or saved copies of the demonstration guide can become out-of-date quickly and result in failed steps.

A ready-to-print PDF version of the IBM watsonx Assistant for Z for Technical Sales Level 4 Lab Guide is here.

Viewing images

Images in the demonstration guide can be enlarged by clicking on the image. Press the Esc key or click the **X** to dismiss the enlarged image.

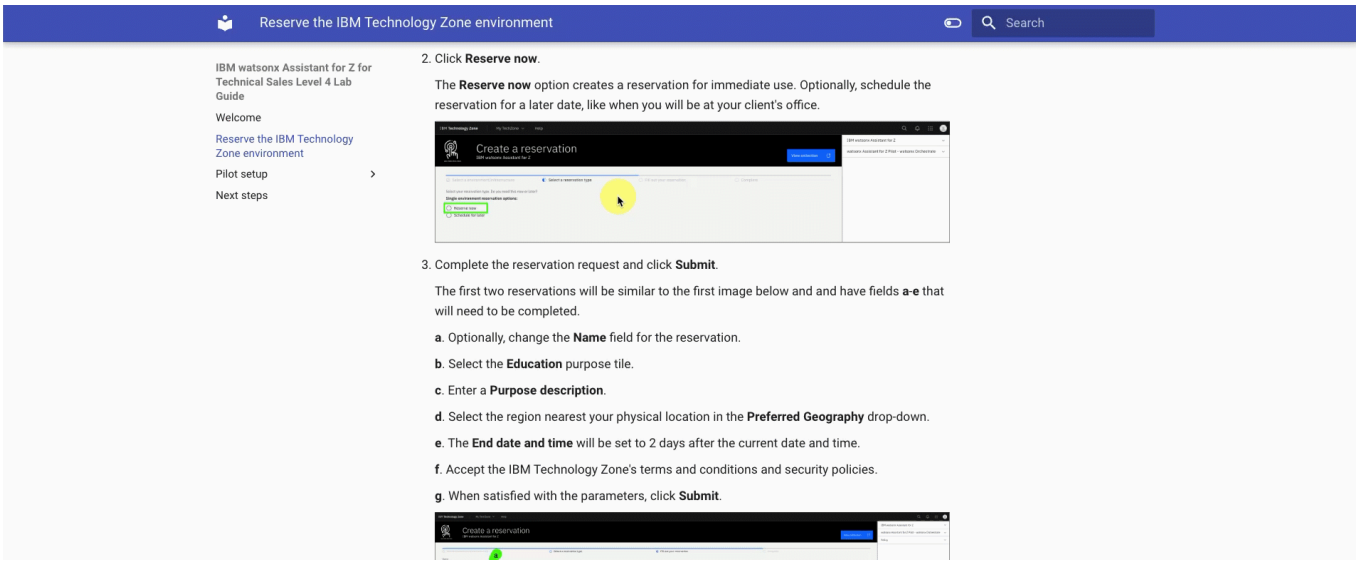
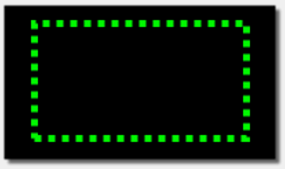


Image highlighting

In some images, the following styles of highlighting are used:

- **Solid highlight box**: This style of box highlights where to click, enter, or select an item.

- **Dash highlight box**: This style of box highlights one of two things: the path to follow to get to a specific location in the user interface, or areas to explore on your own.



Copying commands and prompts

Copying and pasting commands and prompts from this demonstration guide is easy and can eliminate typographical errors.

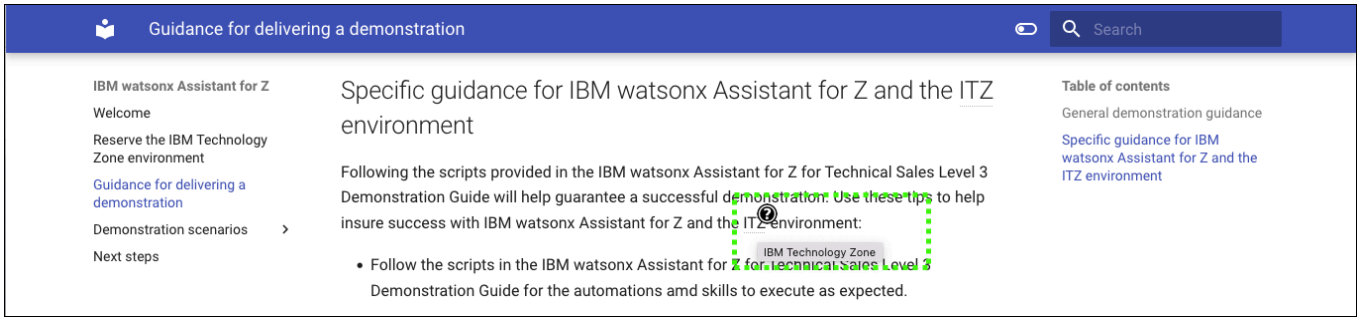
Click the highlighted copy icon and then use your operating system's paste function. For example, `^ Ctrl` + `V`, or right-click and select `Paste`.

Prompt 1

```
What is the APF list in z/OS? Provide a detailed explanation.
```

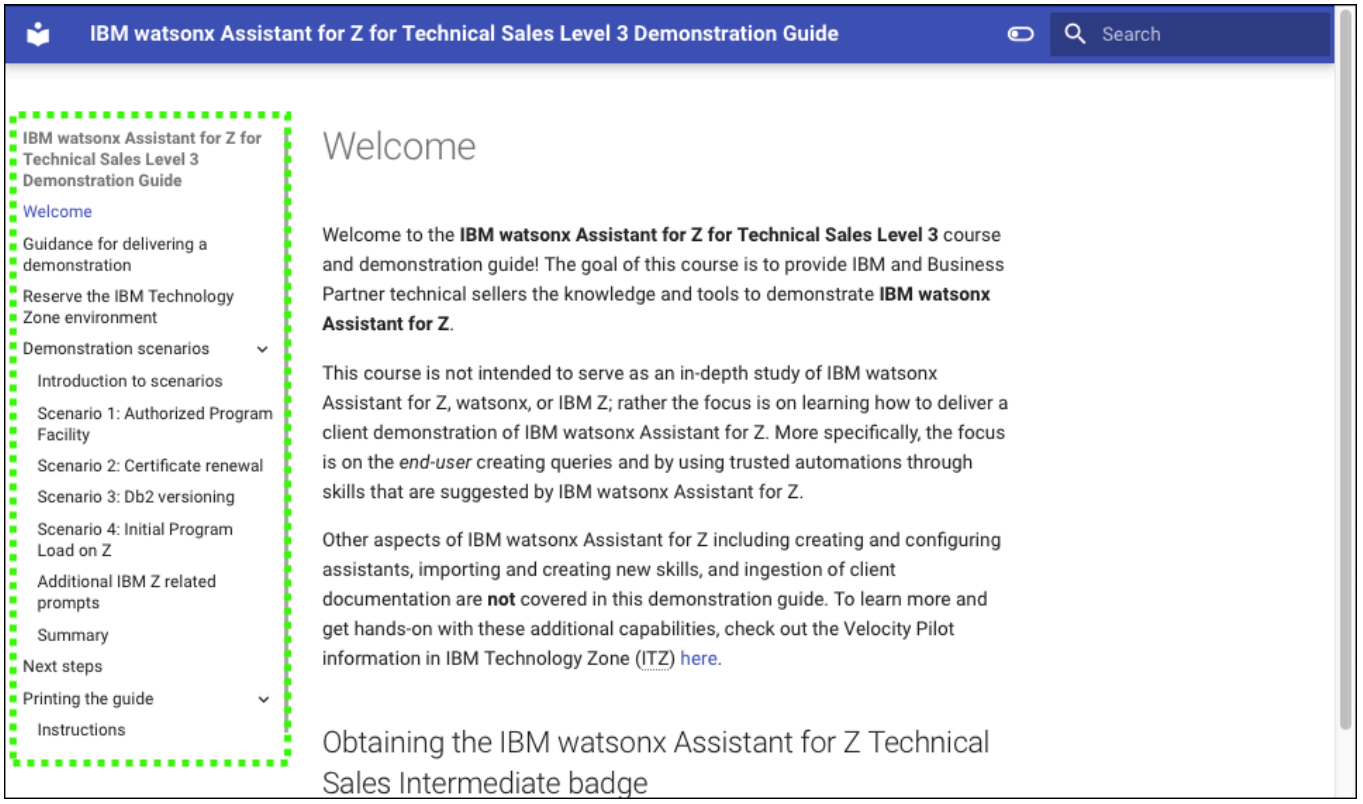
Acronyms and terminology

IBM employees and the tech industry in general, tend to use acronyms. In the demonstration guide, most acronyms will appear with a dashed underline. Hover over the acronym to learn its meaning. A question mark (?) icon will first appear and after a second the tool tip with the acronym's meaning is displayed. Try it here: LPAR.



The Lab Guide table of contents

This **Demonstration Guide** uses a responsive browser-based interface to ensure pages are usable on various devices with different screen sizes. The Demonstration Guide table of contents may be displayed as highlighted in the green dashed box in this image:



However, if the browser window is sized smaller, the table of contents can be accessed by clicking the main menu icon (☰):

Click the main menu icon (☰) to expand the table of contents.

Continue to the Reserve the IBM Technology Zone environments section to begin the journey to obtain the IBM watsonx Assistant for Z Technical Sales Advanced badge.

IBM Technology Zone environment

To enable sellers to learn how to deliver client pilots of IBM watsonx Assistant for Z, three environments are available in IBM Technology Zone (ITZ). The environments are part of the watsonx Assistant for Z Velocity lab collection and can be found in the IBM watsonx Assistant for Z collection.

- **Watsonx Assistant for Z lab – watsonx Orchestrate**: provides a dedicated environment on IBM Cloud where you can create and configure the assistant, set up conversational search, import skills, and configure actions.
- **Ansible Automation Platform (AAP) & z/OS**: provides a pre-configured instance of AAP and Wazi z/OS. This environment includes Ansible playbooks, which you can import as skills within WatsonX Orchestrate and connect to your assistant. Preinstalled templates for various use cases are also available (covered in later sections). Learn more about AAP here. Learn more about Wazi, here.
- **Single Node OpenShift with NFS storage**: provisions a single-node Red Hat OpenShift cluster (SNO) on IBM Cloud. This cluster installs a dedicated instance of OpenSearch for Watson Assistant for Z, enabling ingestion of client-supplied documents.-

> ⚠ **All activities in this lab guide are required.**
>
> To earn the IBM Watsonx Assistant for Z Technical Sales Advanced badge and complete the Level 4 learning plan, you must provision all three ITZ environments and finish every section in the lab guide. Disregard any statements in the ITZ collection that suggest optional environments or tasks.

Follow the instructions to create new reservation requests, extend the reservations, and access the ITZ demonstration environments. Provisioning the SNO environment in ITZ can take several hours, while the other two environments typically provisioning in under 30 minutes.

Create a reservation request

1. Click each of the links that follow to open a browser to the reservation pages of the **IBM watsonx Assistant for Z** ITZ environments.

   > ⚠ **You may be asked to authenticate to IBM Technology Zone.**
   >
   > The steps to authenticate to ITZ are not detailed here as they may vary between users.

   Watsonx Assistant for Z lab – watsonx Orchestrate - reservation page

   Ansible Automation Platform (AAP) & z/OS - reservation page

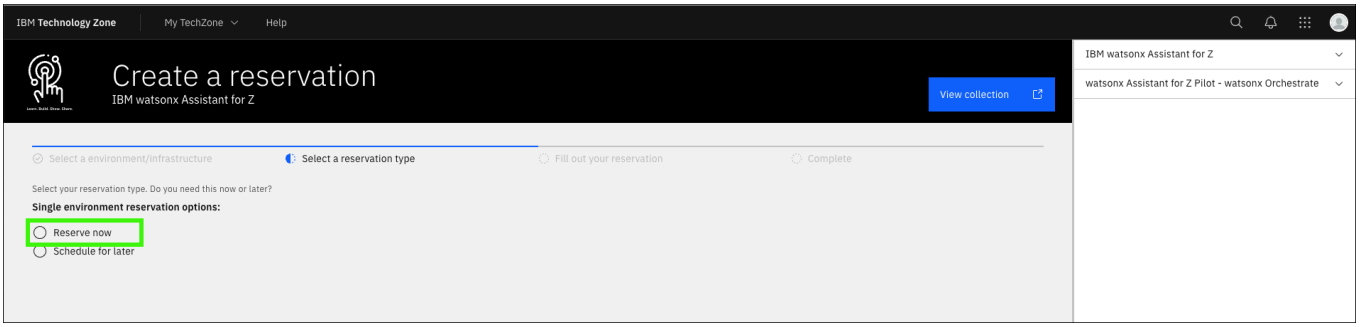
   Single Node OpenShift with NFS storage - reservation page

The next two steps are for one of the three environments. Repeat for all three environments.

Follow the steps to create a reservation in ITZ for all three environments.

2. Select **Reserve now**.

   The **Reserve now** option creates a reservation for immediate use. Optionally, schedule the reservation for a later date, for example, when you are at your client's office to start a pilot.



3. Complete the reservation request form and then click **Submit**.

   The first two reservations are similar to the first image and have fields **a**-**e** that need to be completed.

   a. Name: specify a name for the reservation (optional).

   b. Purpose: select the **Education** purpose tile.

> **For client pilots...**
>
> For client pilots, set the **Purpose** field in the reservation to **Pilot** and provide an opportunity number to receive a longer reservation.

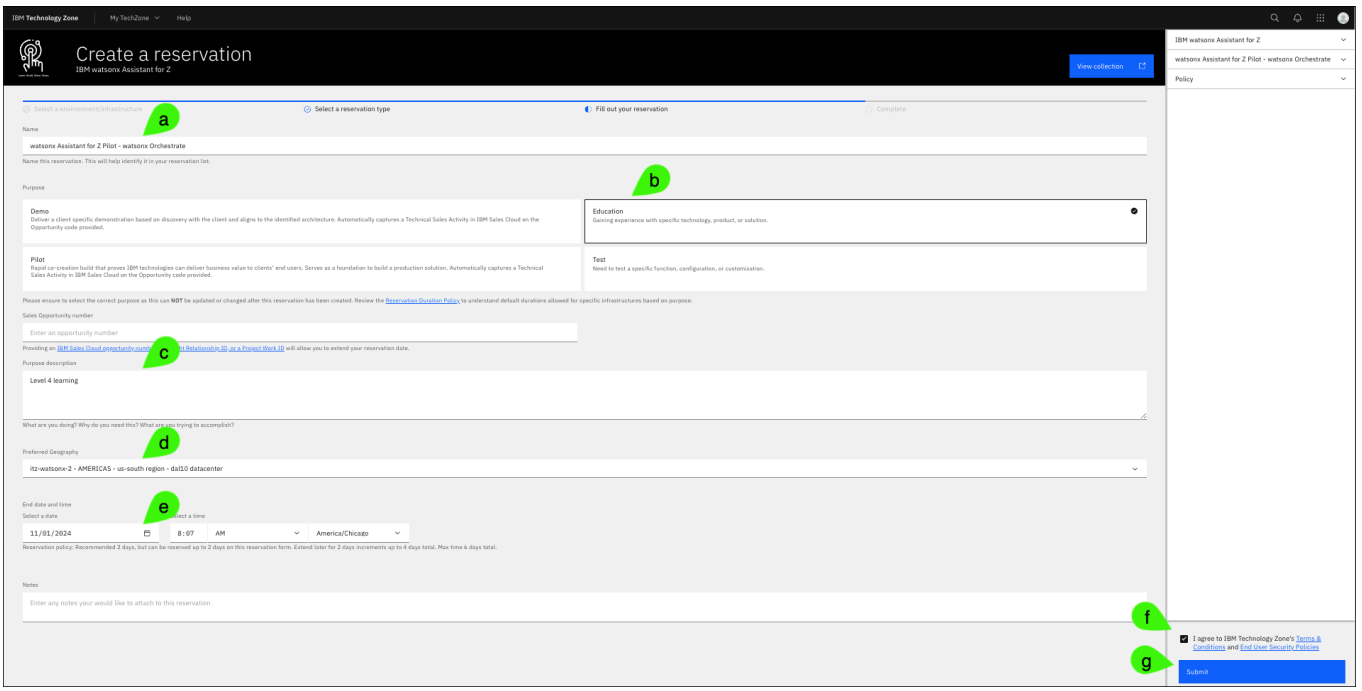
   c. Purpose description: enter a description, for example: Level 4 education.

   d. Preferred geography: select the region nearest to your physical location for improved performance and reduced network latency.

   e. End date and time: the initial maximum will be set to two days after the current date and time. Instructions follow to extend the reservation end date.

   f. Accept the IBM Technology Zone's terms and conditions and security policies.

   g. Click **Submit**.



In addition to the preceding fields, the reservation for the **Single Node OpenShift with NFS storage** has these additional fields:

h. OCP/Kubernetes cluster network: leave the default setting of **10.128.0.0/14**.

i. Enable FIPS security: leave the default setting of **No**. Learn more about the Federal Information Processing Standards (FIPS) here.

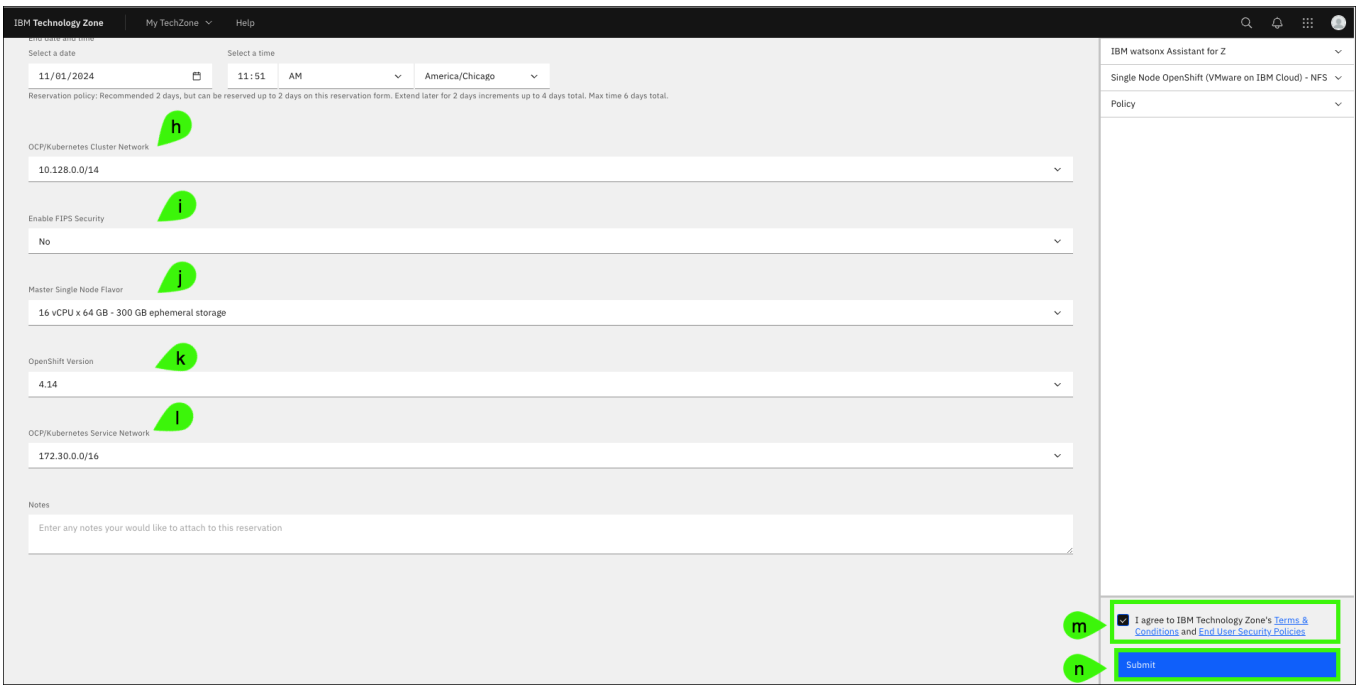
j. Master single node flavor: select **16 vCPU x 64 GB - 300 GB ephemeral storage**.

k. OpenShift version: select **4.14**.

l. OCP/Kubernetes service network: leave the default setting of **172.30.0.0/16**.

m. Accept the IBM Technology Zone's terms and conditions and security policies.

n. Click **Submit**.


IBM Technology Zone
My TechZone
Help
Select a date
11/01/2024
Select a time
11:51
AM
America/Chicago
Reservation policy: Recommended 2 days, but can be reserved up to 2 days on this reservation form. Extend later for 2 days increments up to 4 days total. Max time 6 days total.
h
OCP/Kubernetes Cluster Network
10.128.0.0/14
i
Enable FIPS Security
No
j
Master Single Node Flavor
16 vCPU x 64 GB - 300 GB ephemeral storage
k
OpenShift Version
4.14
l
OCP/Kubernetes Service Network
172.30.0.0/16
Notes
Enter any notes your would like to attach to this reservation
IBM watsonx Assistant for Z
Single Node OpenShift (VMware on IBM Cloud) - NFS
Policy
m
I agree to IBM Technology Zone's Terms & Conditions and End User Security Policies
n
Submit

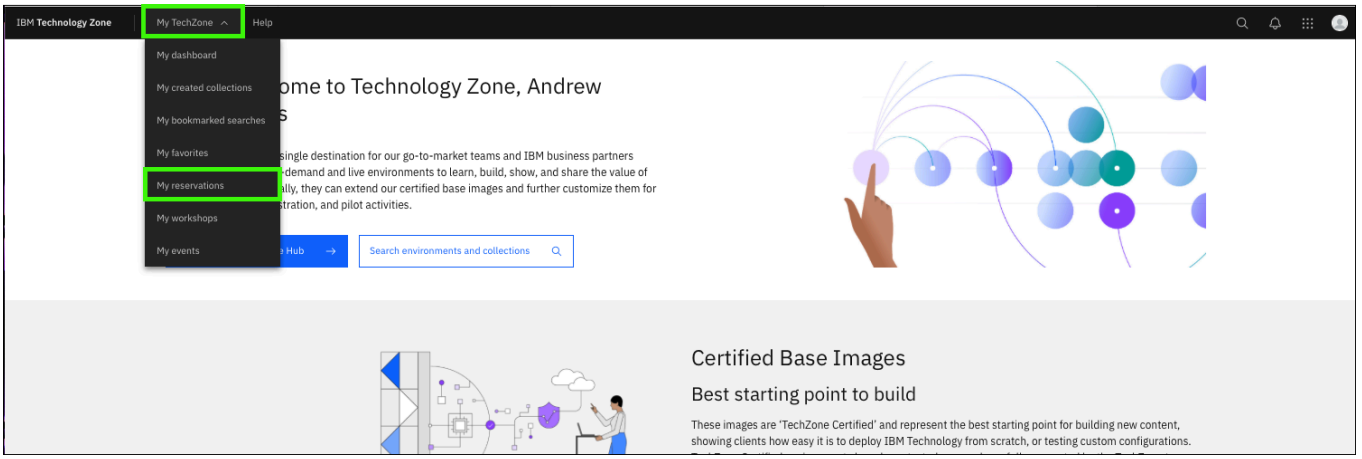
During the provisioning process, multiple emails are sent to you from ITZ as the provisioning process runs. One email states the reservation is provisioning and the other email states that the environment is **Ready**.

In rare cases, the provisioning process may fail. If you receive an email stating the reservation failed, try again by repeating Steps 1-3 for the environment that failed to provision. In addition, review the Troubleshooting section that follows. If issues continue, open an ITZ support ticket by using the methods that are mentioned in the Support section.

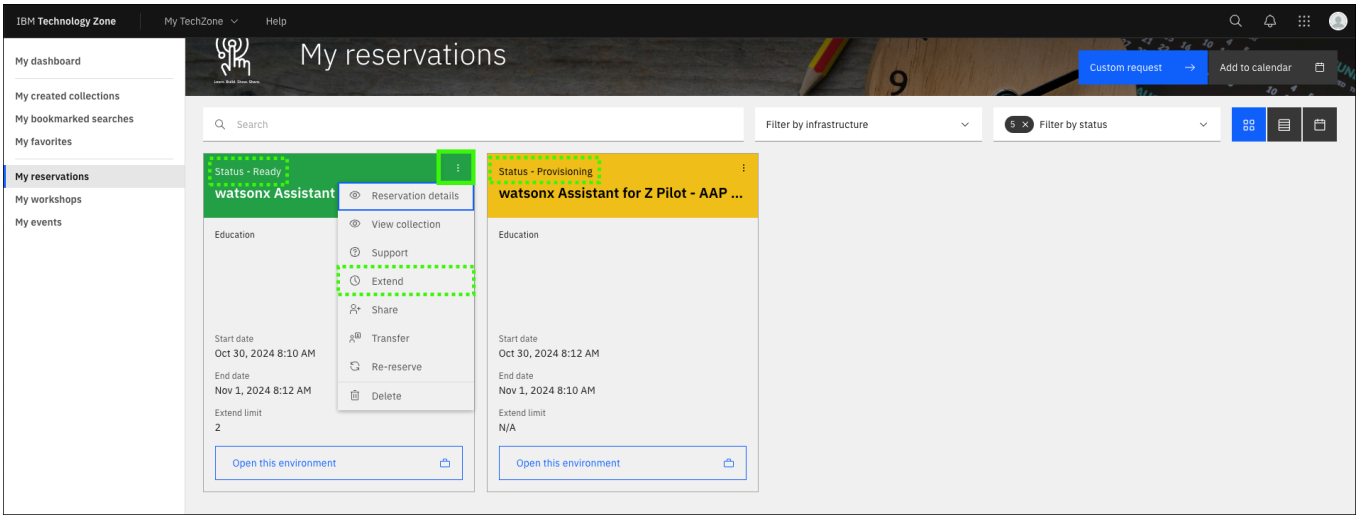
Extend the reservation

When the reservations are in the **Ready** state, you can extend each reservation beyond its original end date. The duration of the extension will vary by reservation.

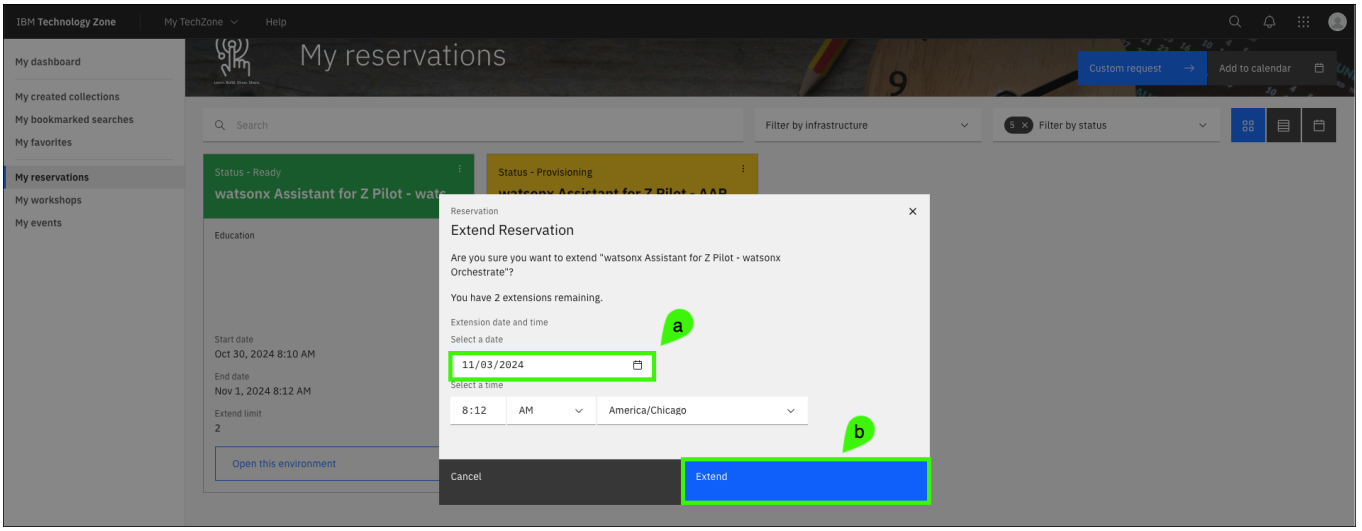
1. In the IBM Technology Zone portal, expand **My TechZone** and select **My Reservations**.



2. Click the **overflow icon** (⋮) on the reservation tile and select **Extend**.



3. Click the **Select a date** option, (a) specify the date to extend to, and then (b) click **Extend**.



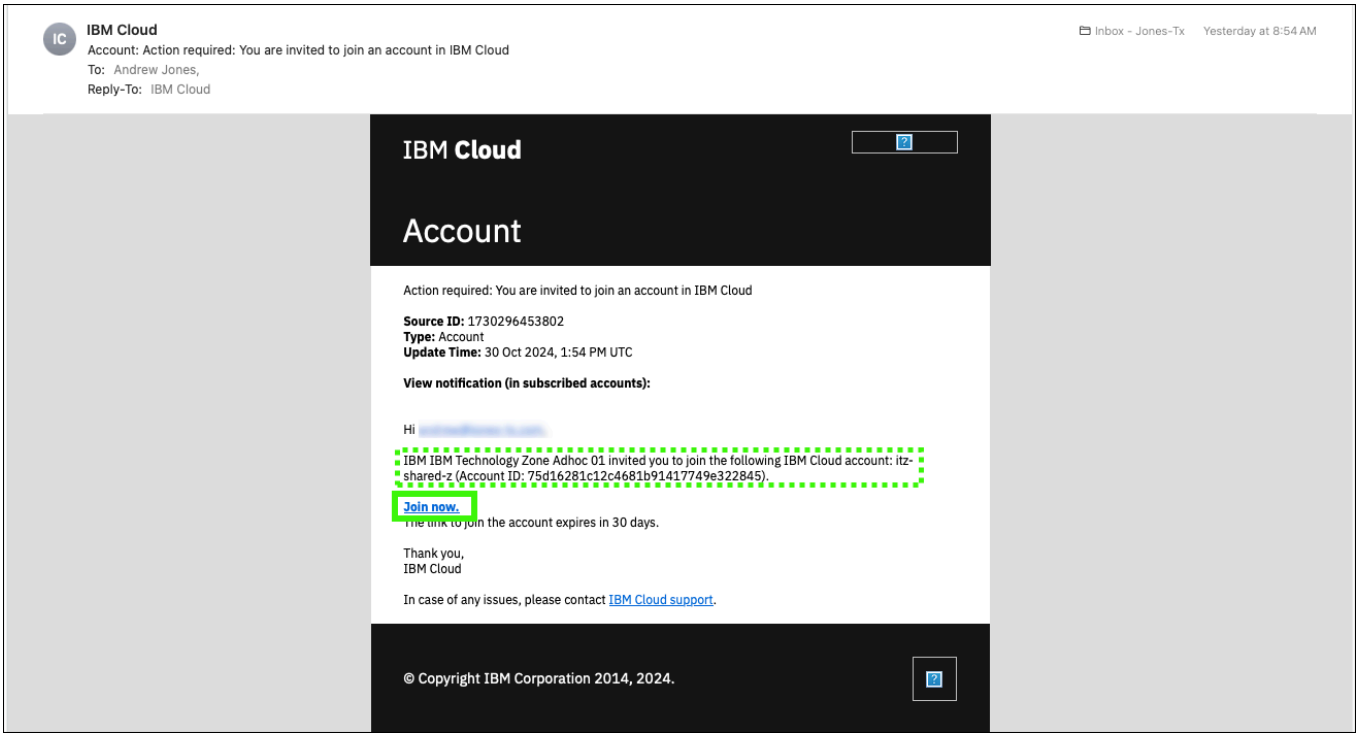
If you anticipate needing more time, repeat Steps 5-6 to extend the reservation to the maximum allowed. Repeat these steps for the other two reservations.

Join the ITZ IBM Cloud account

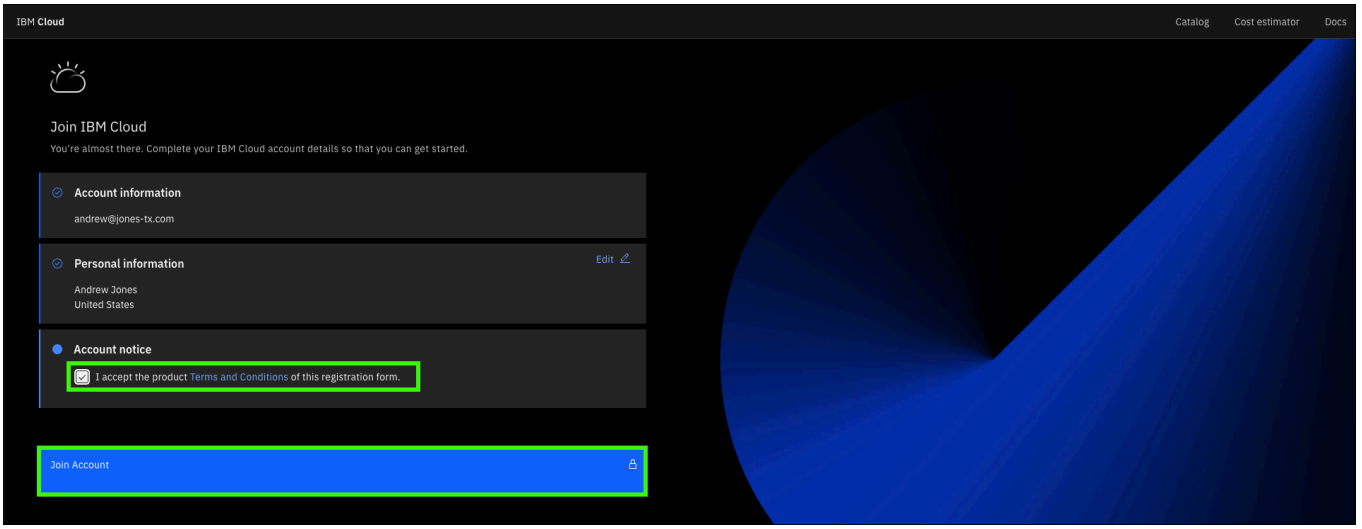
Both the **watsonx Assistant for Z lab – watsonx Orchestrate** and the **Ansible Automation Platform (AAP) & z/OS** environments add you to an IBM Cloud account while your reservation is active. During the provisioning process of these ITZ environments, you receive two emails from IBM Cloud.

You only need to accept the invitation to the **watsonx Assistant for Z lab – watsonx Orchestrate** environment.

1. Open the email from **IBM Cloud** and click the **Join now** links.



2. In the **Join IBM Cloud** browser window that opens, select the **I accept the product Terms and Conditions** of the registration form, and then click **Join Account**.

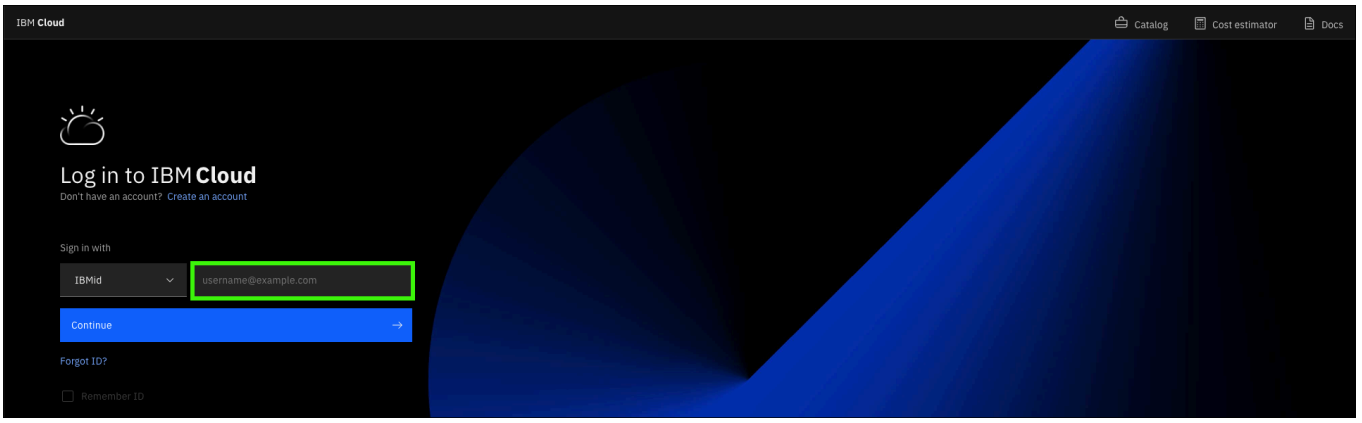


After joining the account, verify that the account appears in your available account list in the IBM Cloud portal.

3. Click the following link to open a browser to the IBM Cloud portal.

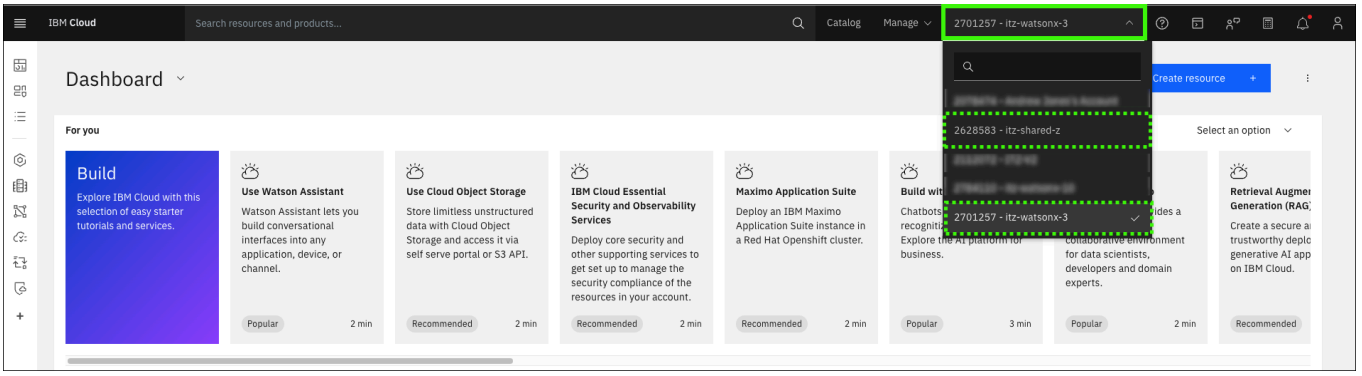
   IBM Cloud portal

4. Follow the directions to complete the authentication to IBM Cloud using the same email address you used to login to ITZ. The login steps vary depending on any two-factor authentication methods enabled.



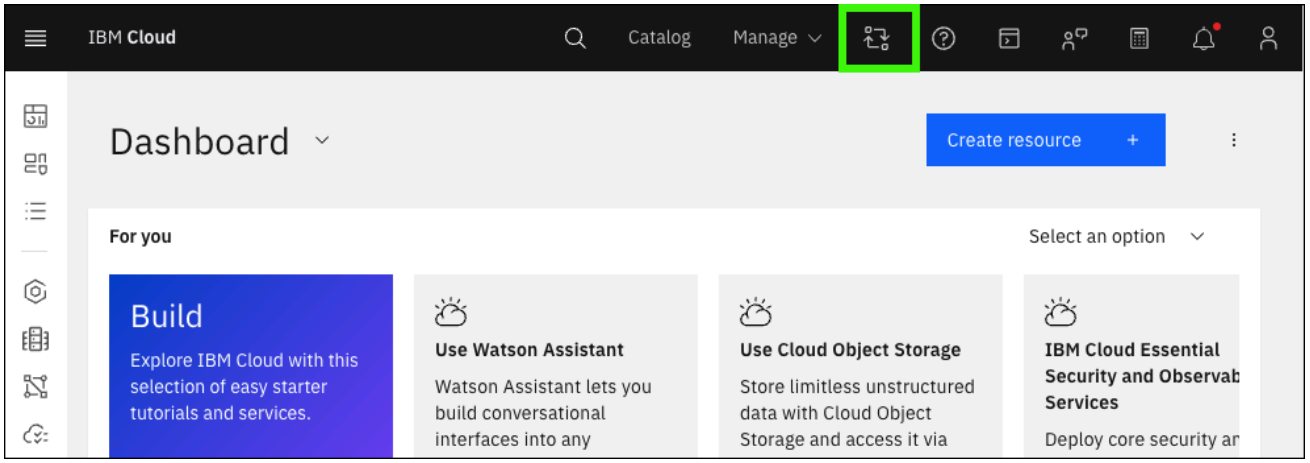
5. Click the **account** menu and verify access to the IBM Cloud account listed in your ITZ reservation.

> **The account may change within ITZ.**
>
> Over time, the account may change for the environments. The account names should align with the account named in the invitation email you received.



Does your IBM Cloud portal view look different?

If your IBM Cloud portal looks different from the images above, it could be because the IBM Cloud portal has gone through a design change, or your browser window is set to smaller size. Instead of the current selected account appearing in the top menu, you may see this **change account** icon: . Click this icon to view the list of accounts you can access.



Accessing the environments

Each reservation provides access to its respective environment. Details for accessing each environment are provided in the **Pilot setup** sections that follow in the lab guide.

After all three reservations are in the **Ready** state and you accept the invitations to the IBM Cloud accounts, proceed to the next section to complete the pilot setup.

Troubleshooting

If your reservation for the Single Node OpenShift environment fails...

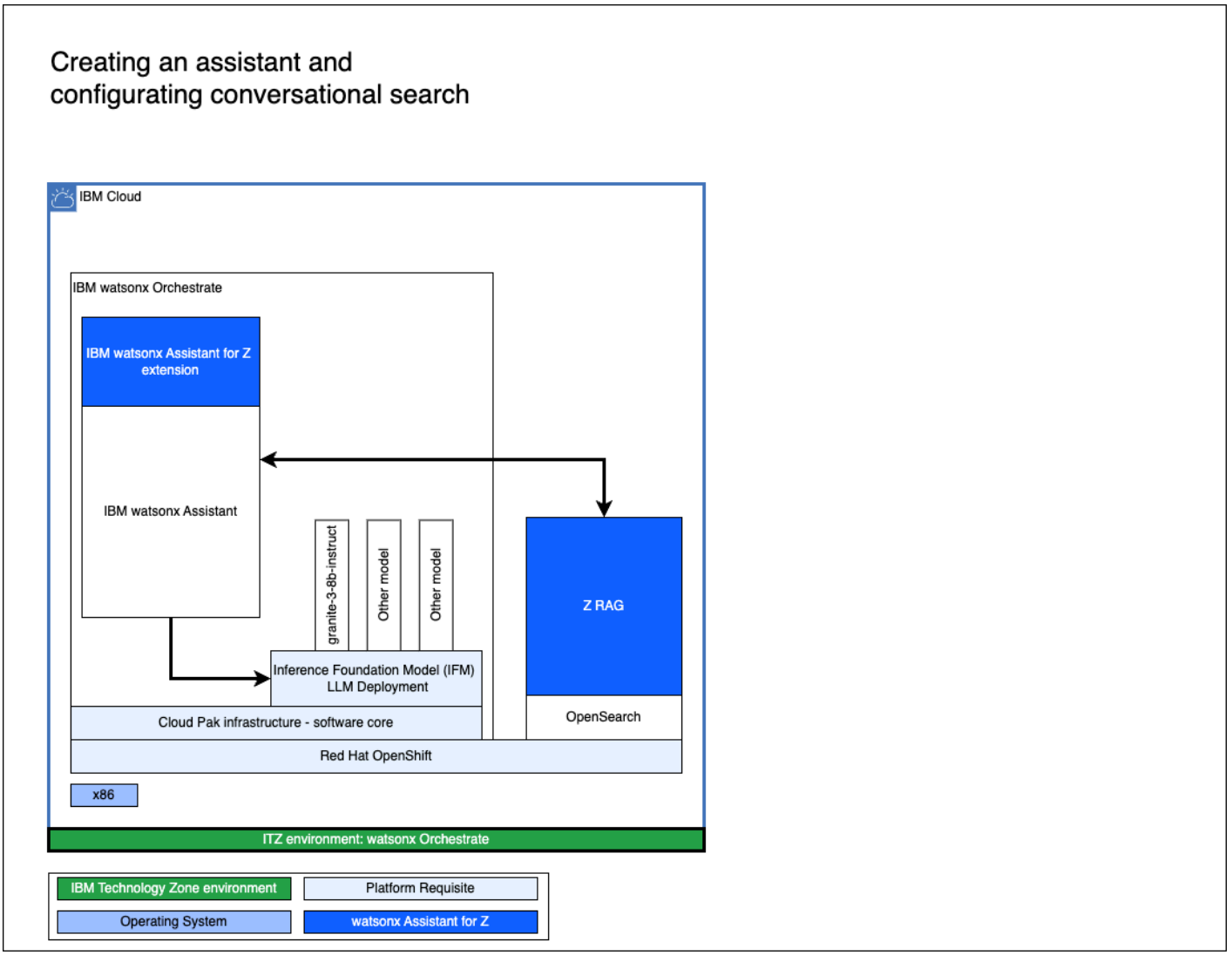
If your reservation for the Single Node OpenShift environment fails, try selecting one of the **eu-gb region** options as the **Preferred Geography**.

Pilot setup

Creating an assistant and configuring conversational search

Watsonx Orchestrate enables you to create and configure an assistant with conversational search capabilities. Configure your assistant to use conversational search by leveraging a hosted OpenSearch instance. The pre-configured instance of watsonx Orchestrate in IBM Technology Zone (ITZ) boasts over 220 knowledge sources and supports Retrieval Augmented Generation (RAG). The large language model (LLM) providing conversational AI augments this knowledge based on IBM Z documentation, generating IBM Z context-aware responses to queries with content-grounded knowledge.

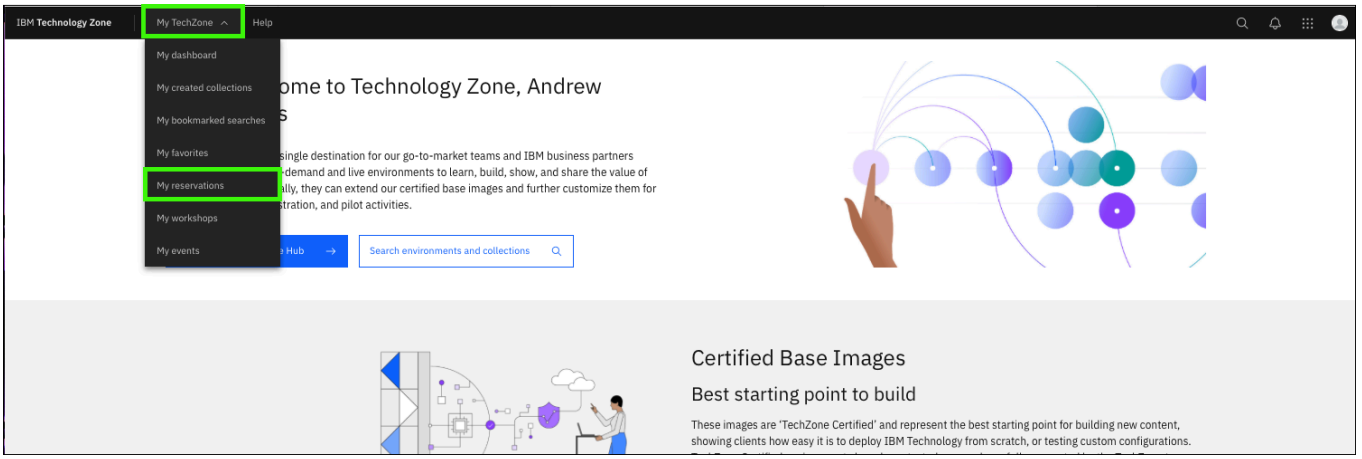
Below is a high-level, logical architecture of the environment you will deploy in this section.



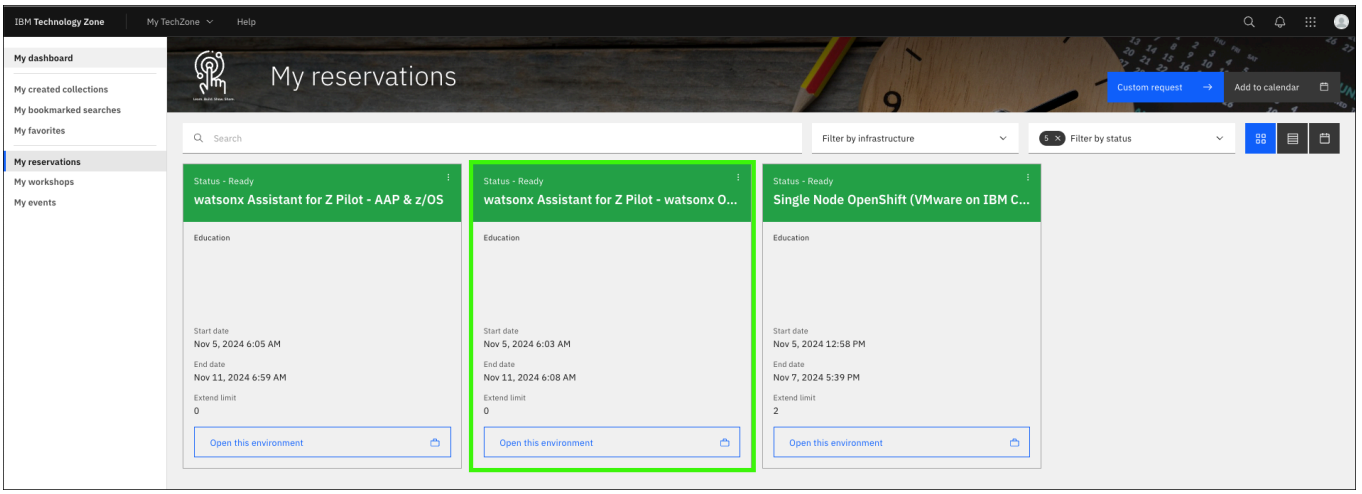
Access the ITZ IBM Cloud account for the watsonx Assistant for Z Pilot environment

1. In the IBM Technology Zone portal, expand **My TechZone** and select **My Reservations**, or click the following link.

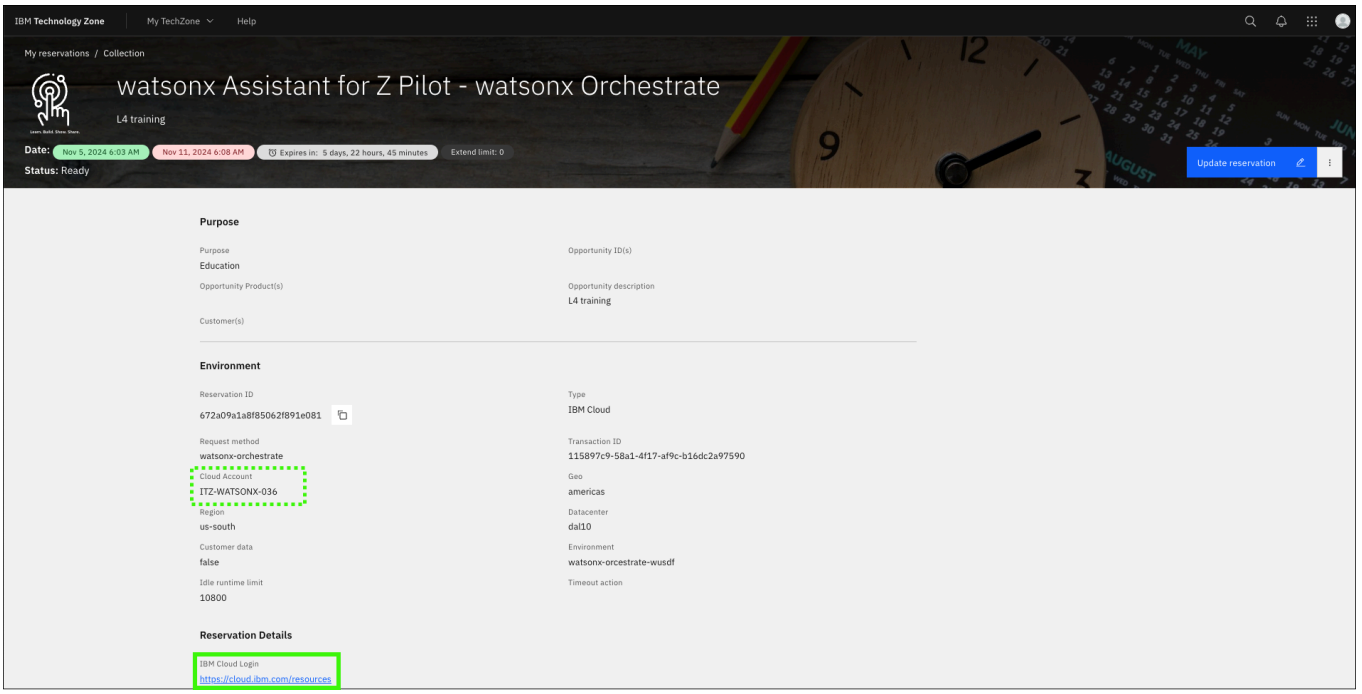
   ITZ My reservations



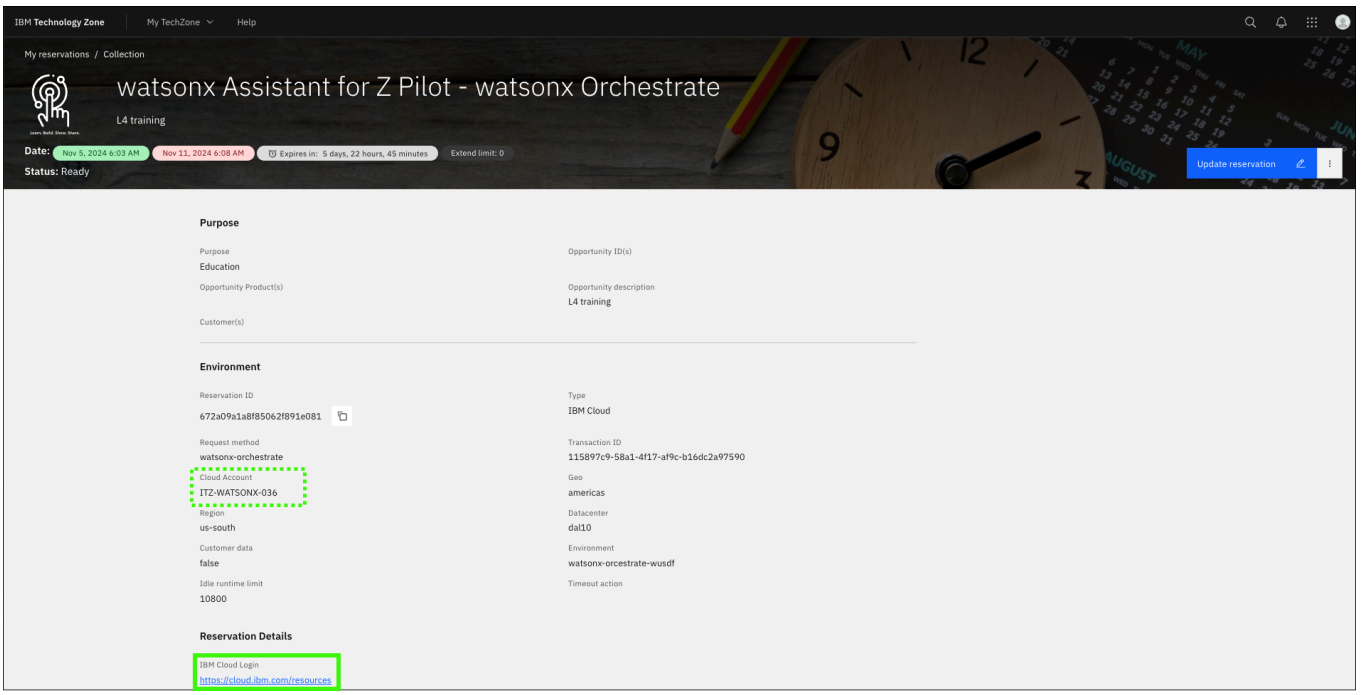
2. Click the **watsonx Assistant for Z Pilot - watsonx Orchestrate** tile.



3. Record the ITZ IBM Cloud account name associated with the reservation.



4. Click the **IBM Cloud Login** link.

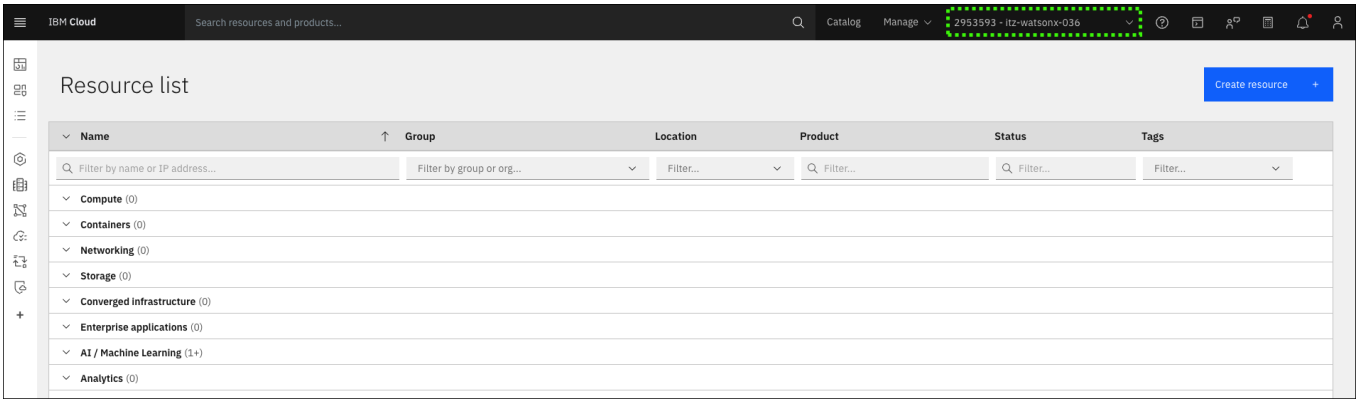


Steps to authenticate to IBM Cloud are not illustrated here.

You may need to authenticate to IBM Cloud after clicking the link. These steps are not shown here as they may vary by individual.

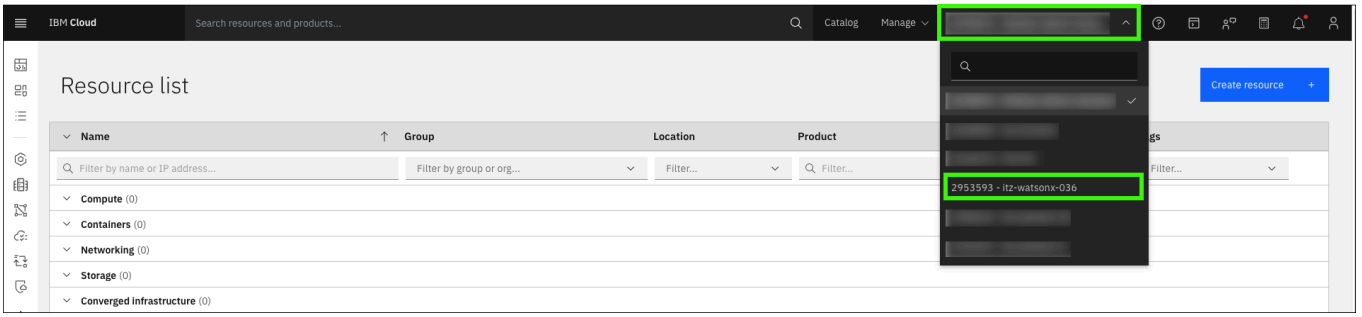
5. Verify that the current IBM Cloud account is the same as the account name recorded in step 3. If the account is not the same, switch to the proper account.

   Note: The formatting of the name can appear differently than what is shown in the ITZ reservation.



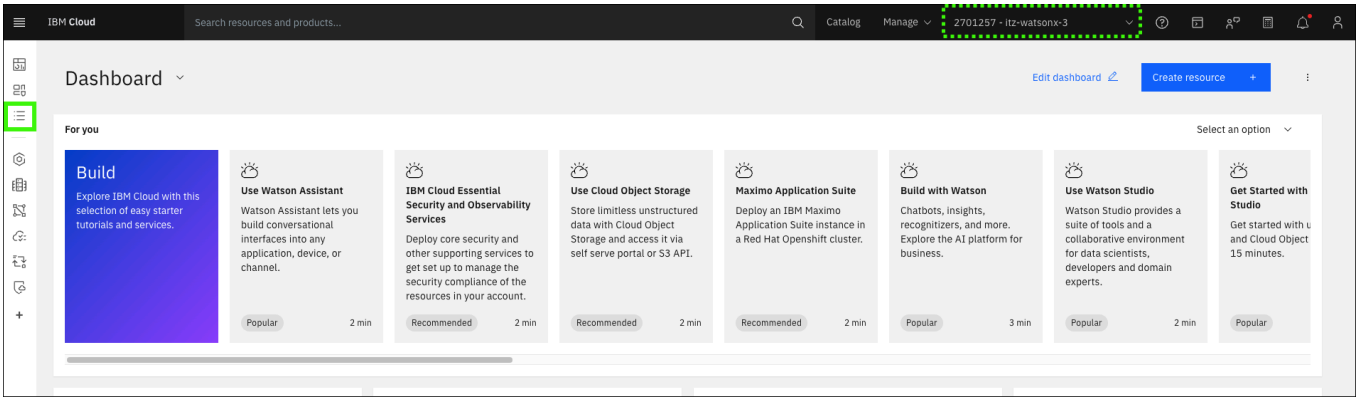
   If the proper account is not listed, click the account drop down and select the proper account.

   Note: If your browser window is narrow, the account drop down can be depicted with the Switch Account icon ( ).

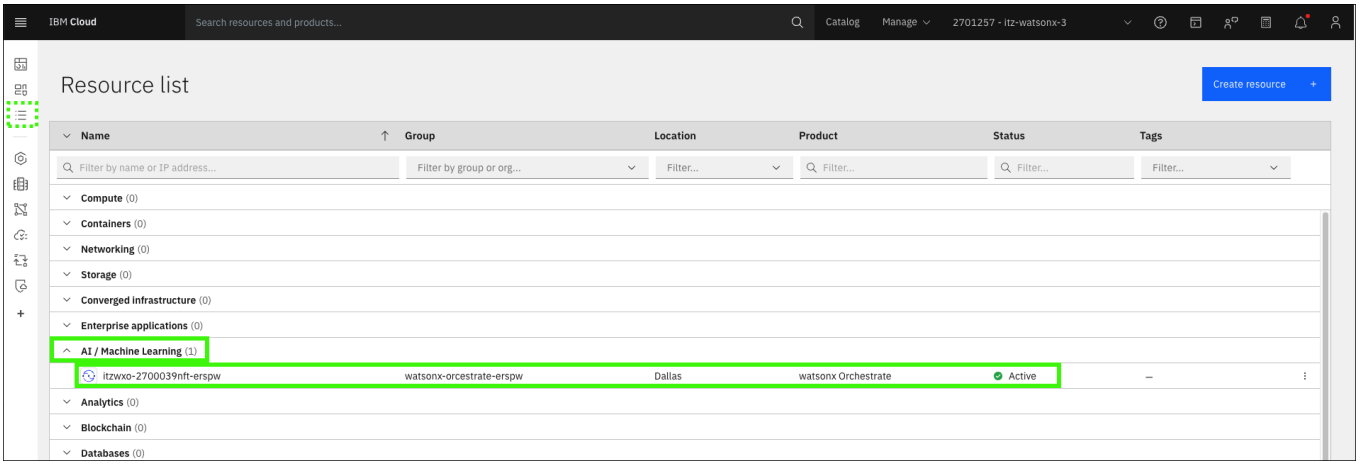


Create your Assistant

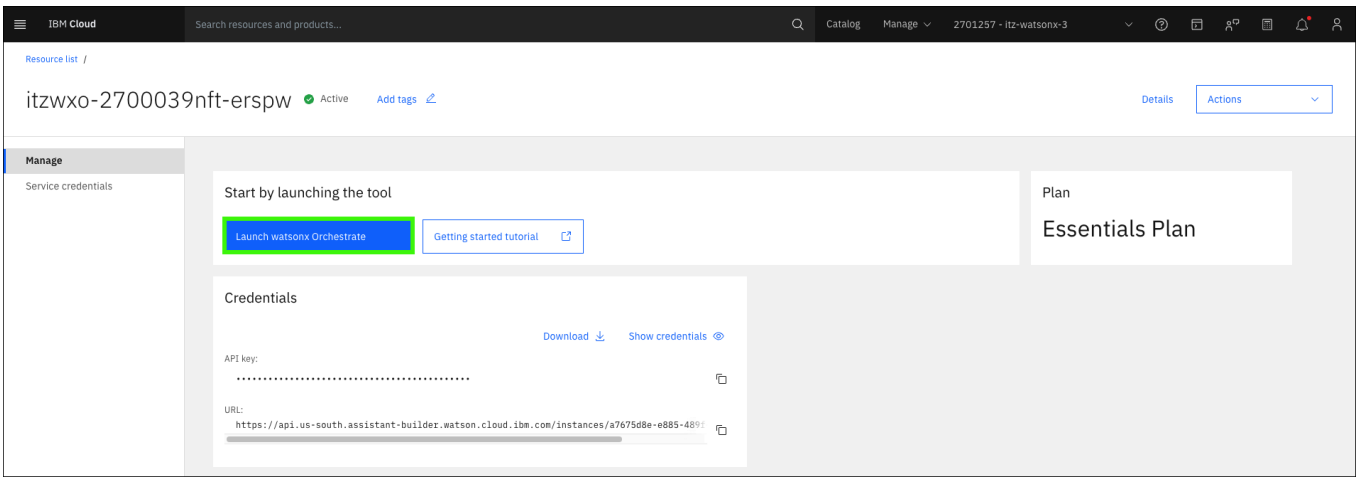
1. Click the **Resources** icon ( ).



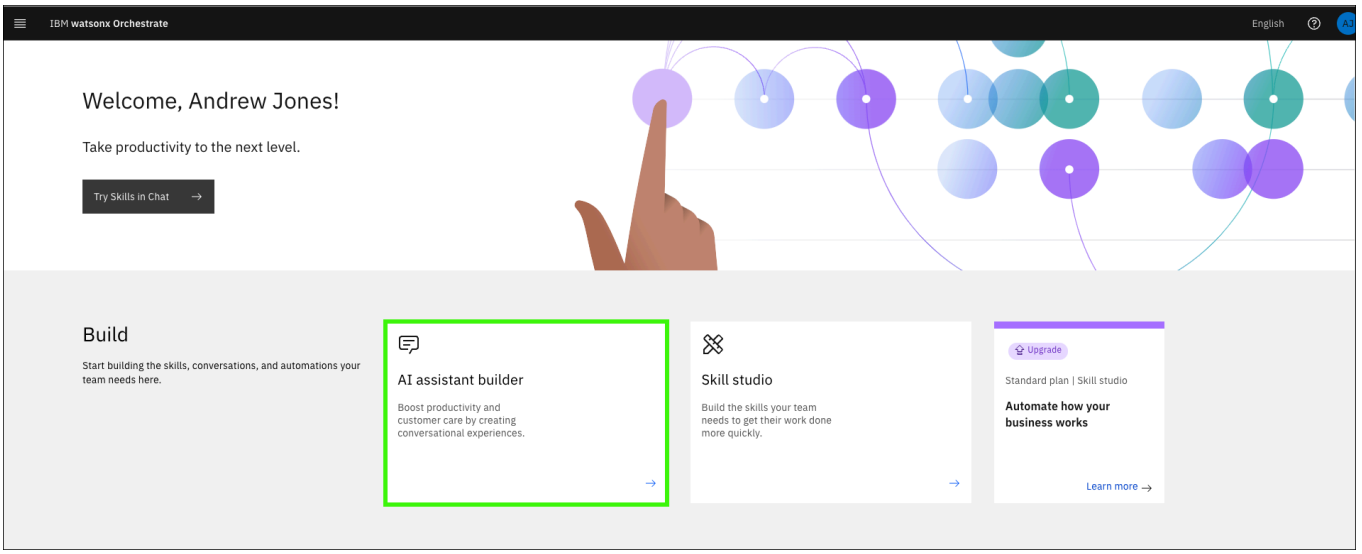
2. Expand the **AI / Machine Learning** section and click the **watsonx Orchestrate** instance listed (the instance name is different than shown in the following image).



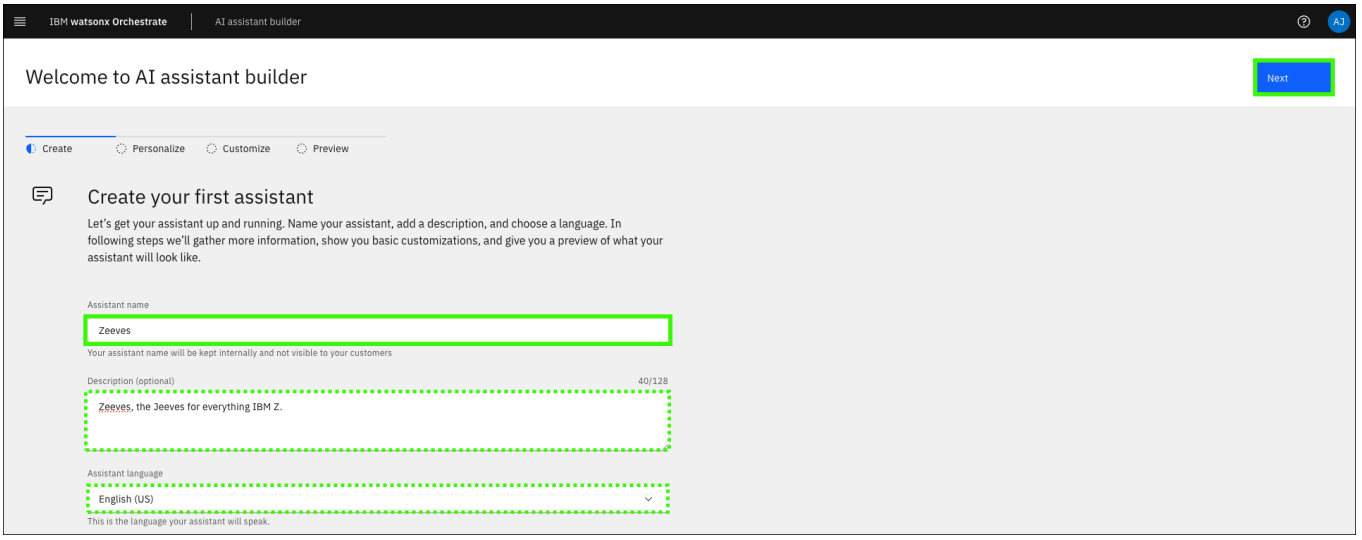
3. Click **Launch watsonx Orchestrate**.



4. Click the **AI assistant builder** tile to start creating a new assistant.



5. Enter a name and optional description for your assistant and click **Next**.



6. Complete the **Personalize your assistant** form and click **Next**.

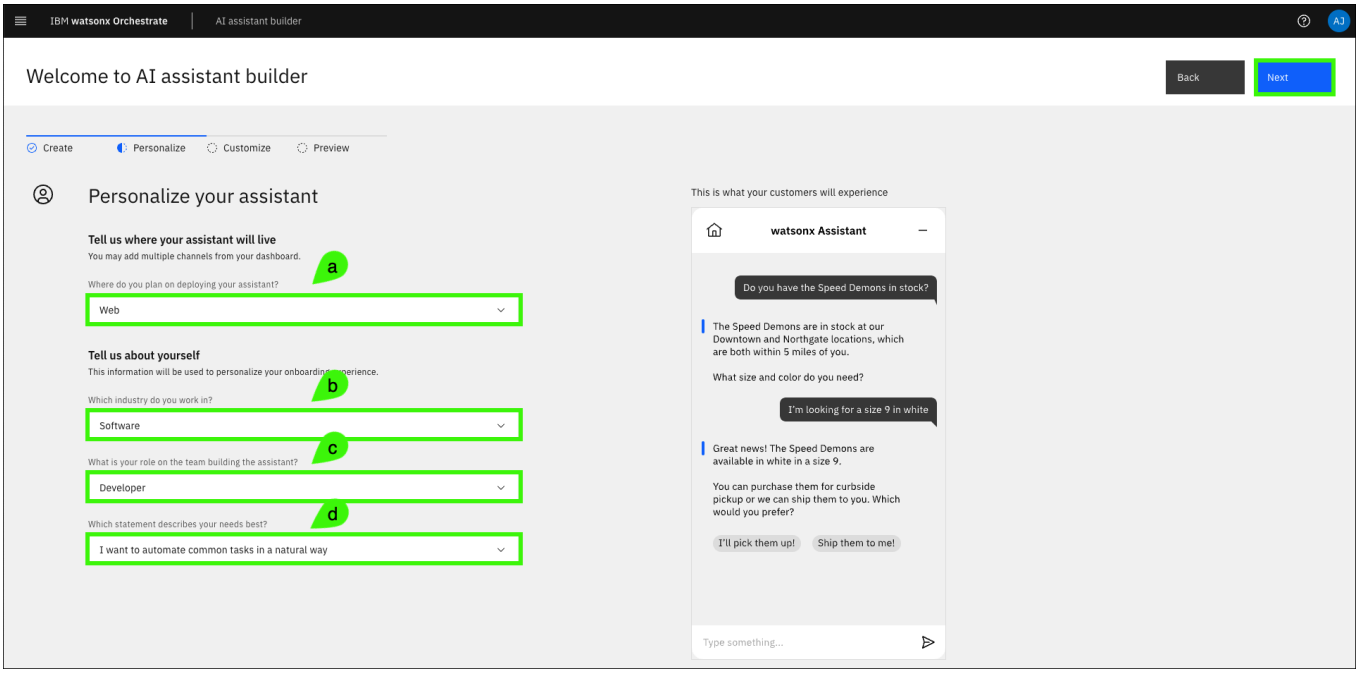
   Explore the personalization options. In creating an assistant for a client pilot, consider specifying attributes that align with the client's business.

a. Select **Web**.

b. Select the industry of your choice.

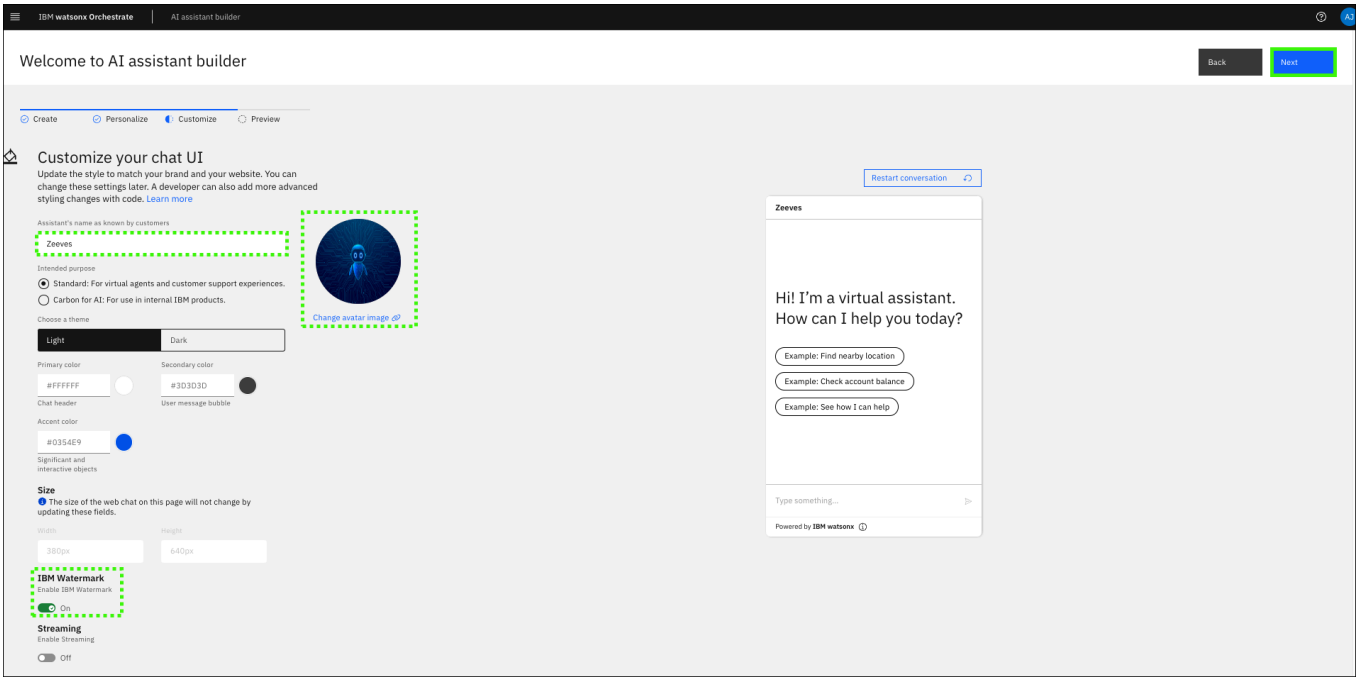
c. Select the role of your choice.

d. Select the need of your choice.

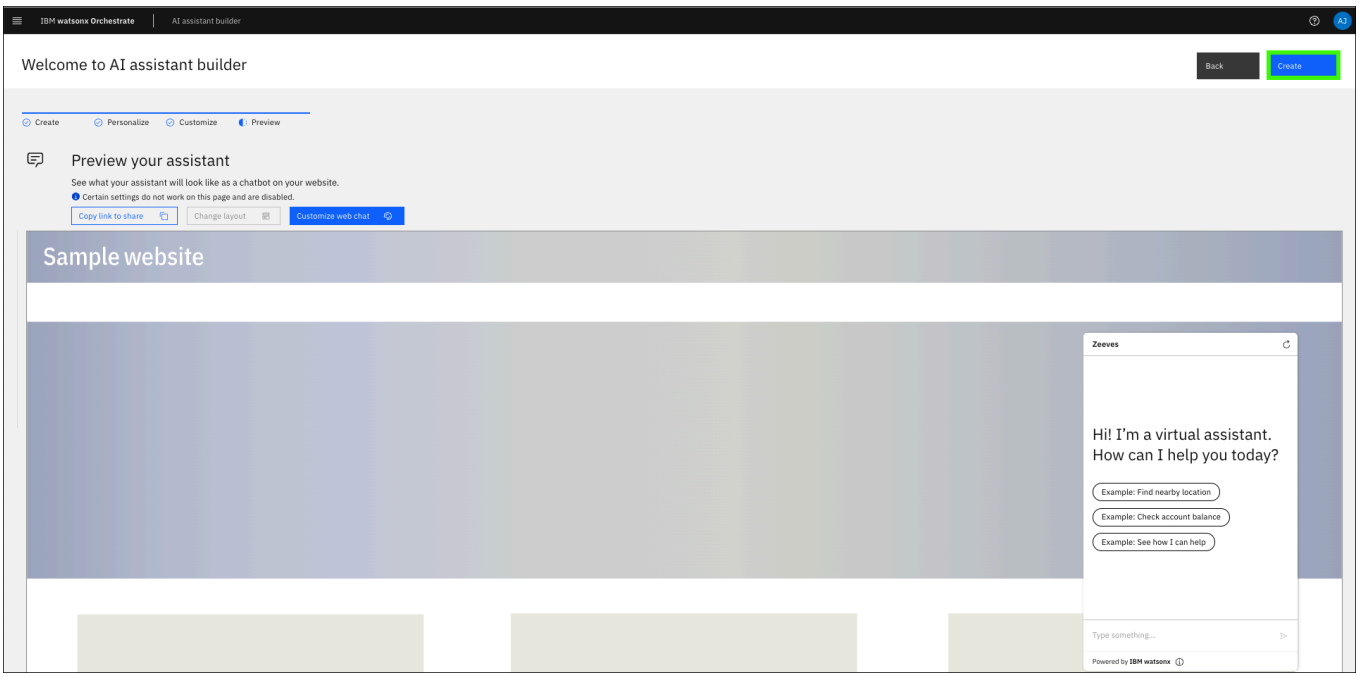


7. Complete the **Customize your chat UI** form and click **Next**.

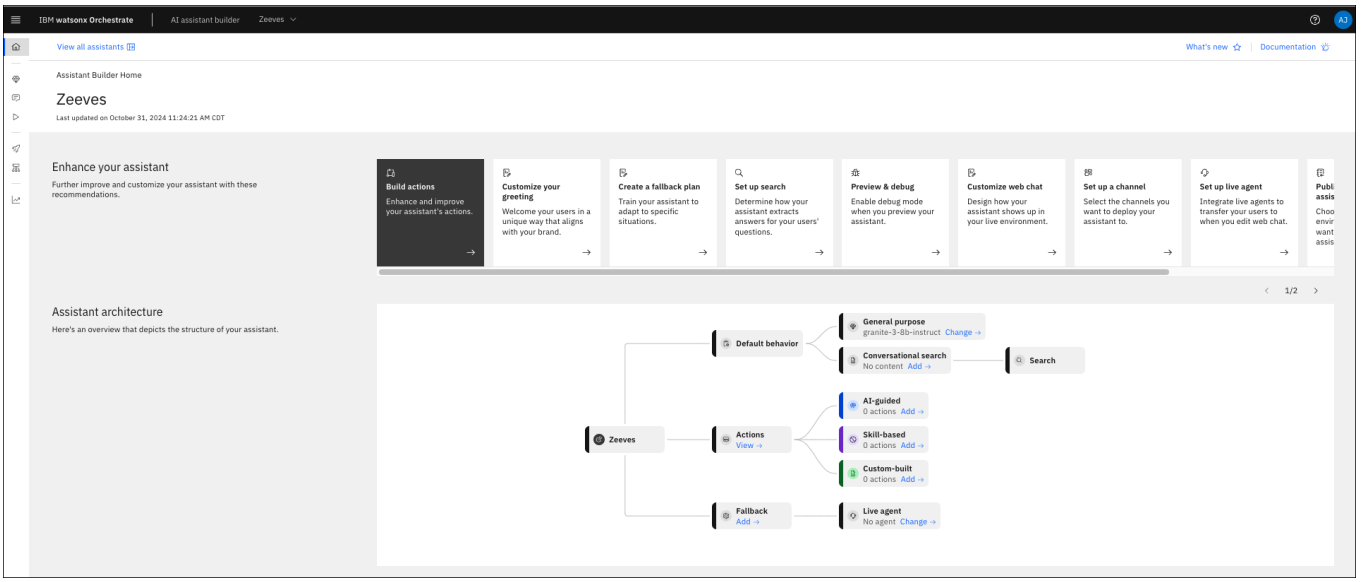
   Explore the customization options. When creating an assistant for a client pilot, consider specifying attributes that align with the client (for example, colors and logos).



8. Preview your assistant and then click **Create**.



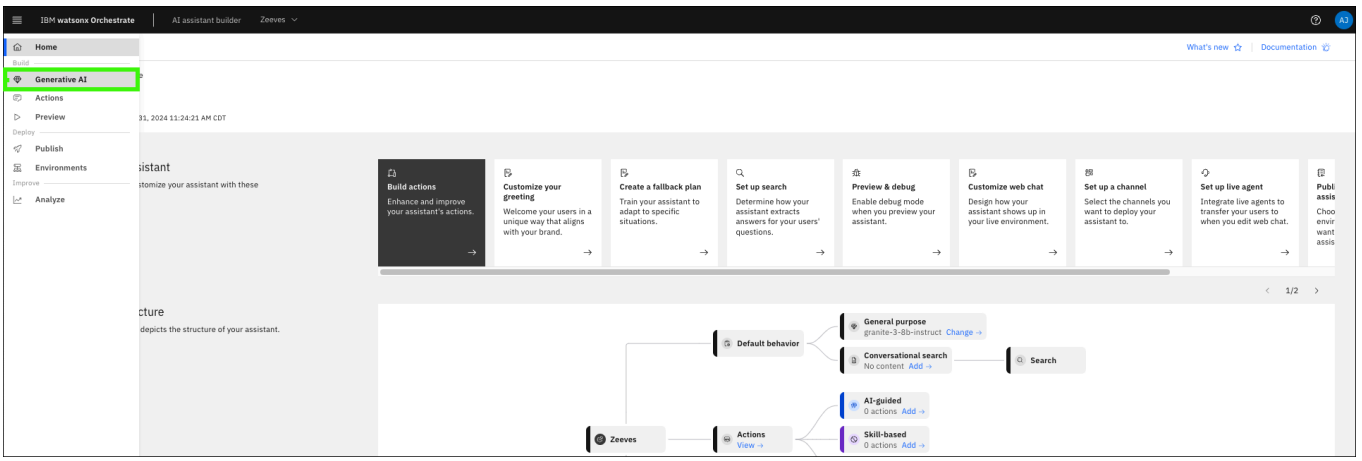
The assistant is now created.



Configure conversational search

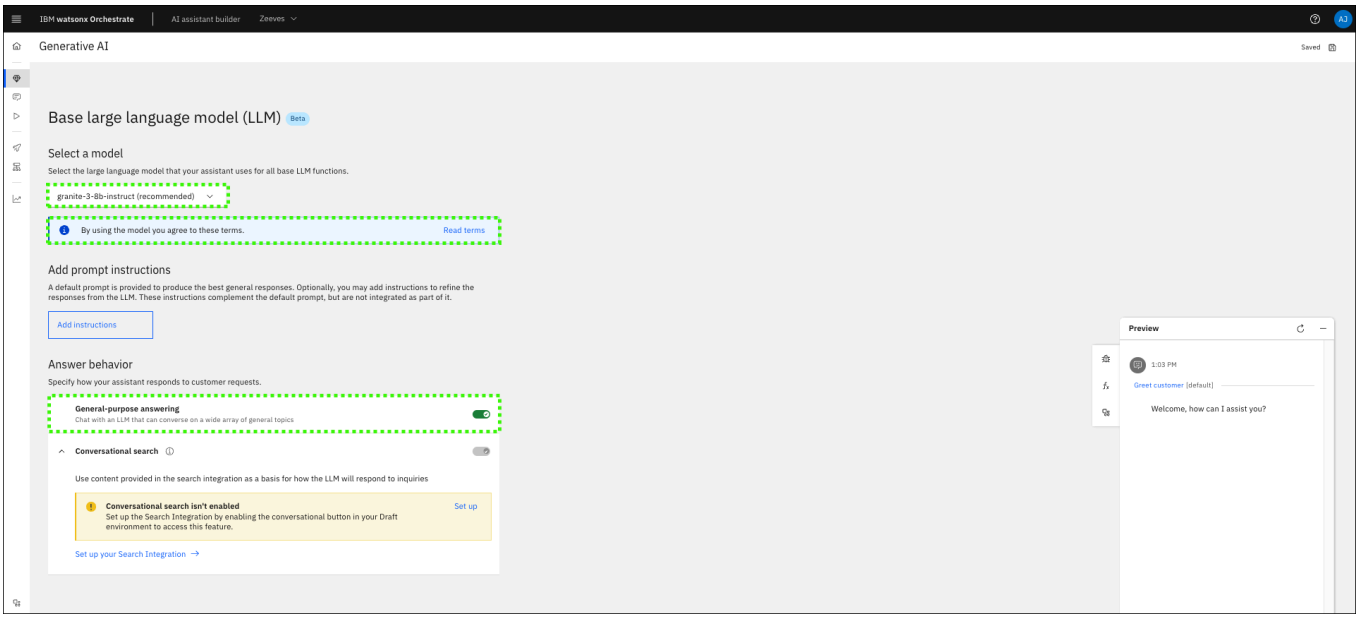
In the next steps you will be to configure **conversational search** for your assistant that uses a hosted instance of OpenSearch.

1. Click **Generative AI** menu item ( ) in the left navigation.



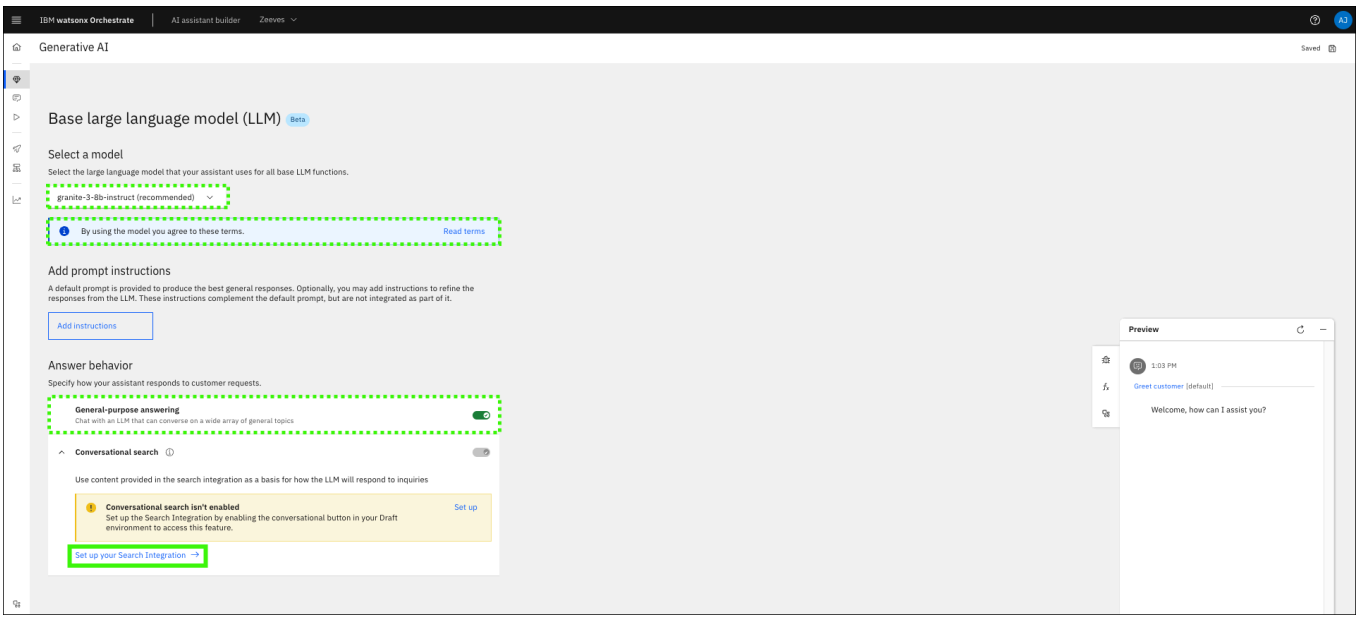
2. Select the base large language model (LLM) settings.

> **If available, set the LLM model to granite-13b-chat-v2.**
>
> Notice their are several LLM models available. **The current recommendation is for pilots, to use the** granite-13b-chat-v2 **model as it provides the best results in recent testing.** However, this model has been deprecated as of 2024-11-04 and will may not be available after 2025-01-19. This recommendation will be updated to reflect any changes in recommended models. **Note**: screen captures that follow show the **granite-3-8b-instruct** LLM as being selected.

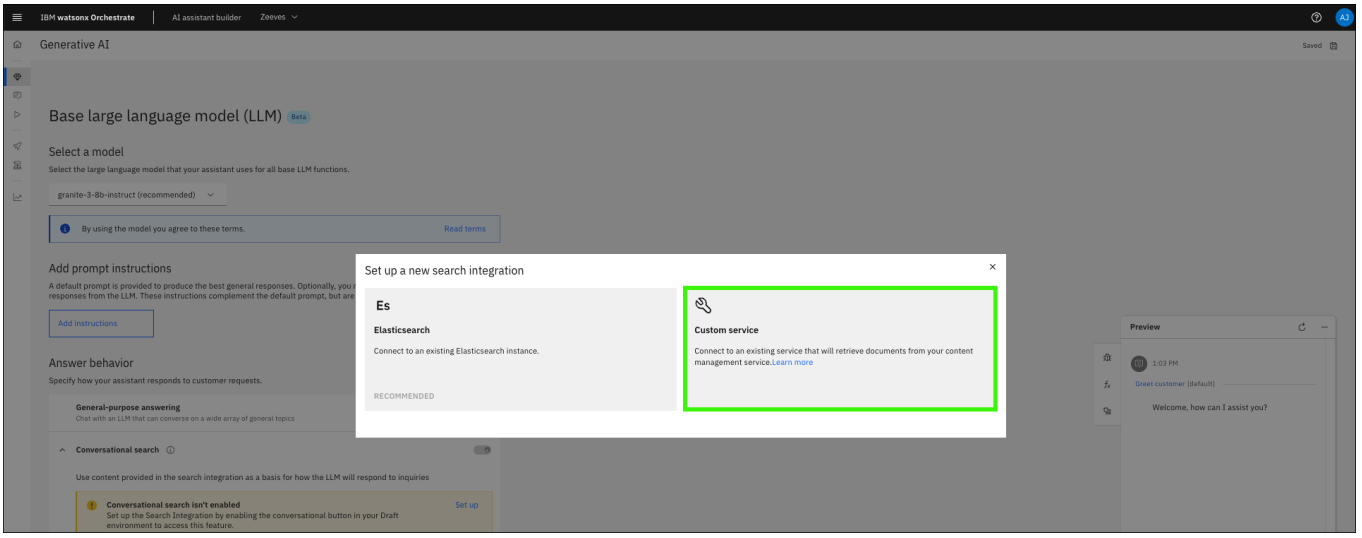


3. Click **Set up your Search Integration**.

   By default, conversational search is not enabled when an assistant is created. Conversational search takes priority over general-purpose answering if both are enabled. Learn more about conversational search in watsonx here.



4. Click **Custom service**.



5. Complete the **Custom service** (**a-e**) form and then click **Next** (**f**).

   a. Select **By providing credentials**.

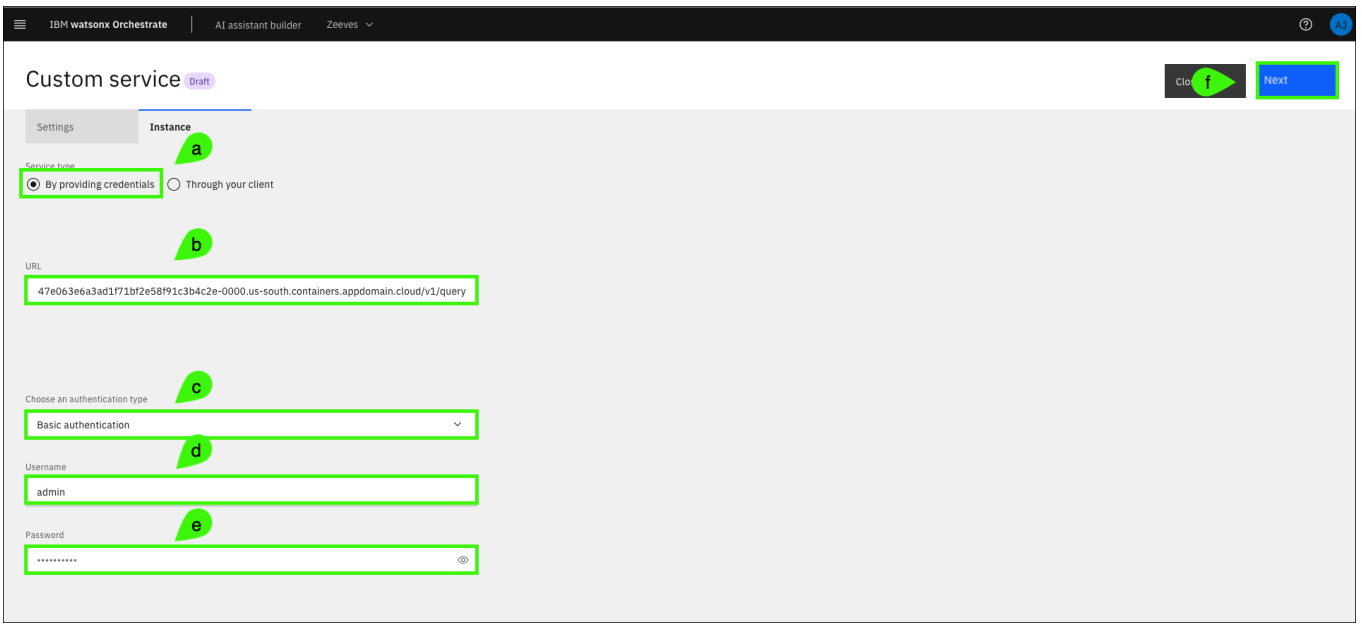
   b. Enter the following value in the **URL** field (use the copy icon to avoid typographical errors). This is the URL for the a shared OpenSearch instance. In later sections you will created and customize a dedicated instance.

   ```
   https://wxa4z-opensearch-wrapper-wxa4z-demo-v2-1-0.wxo4z-opc-opensearch-clus-47e063e6a3ad1f71bf2e58f91c3b4c2e-0000.us-south.containers.appdomain.cloud/v1/query
   ```

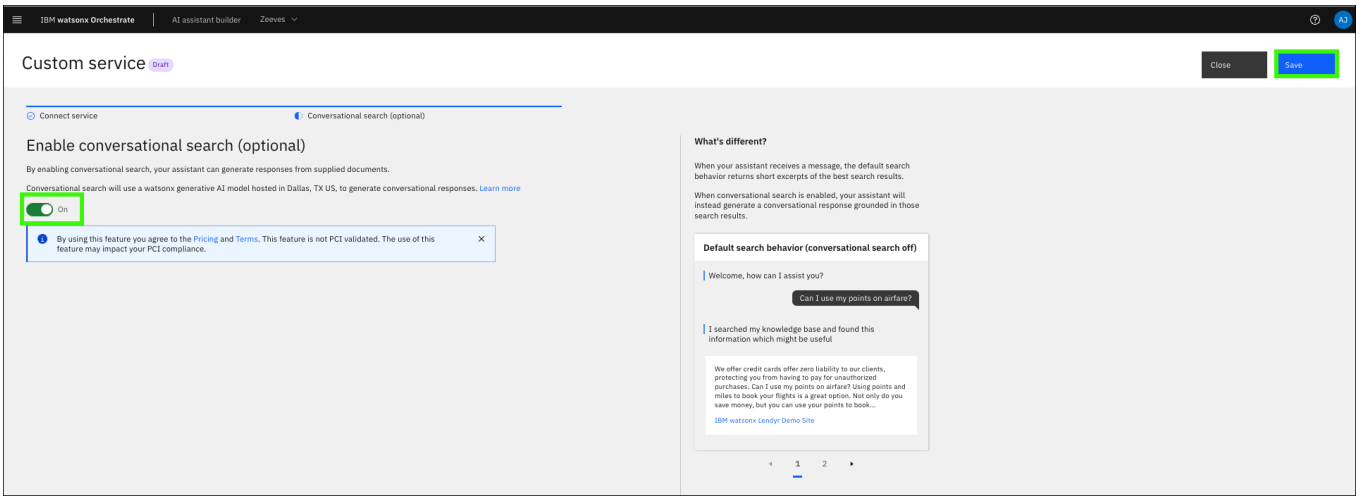
   c. Select **Basic authentication** in the **Choose an authentication type** drop-down list.

   d. Enter `admin` in the **Username** field.

   e. Enter `secureP@ssw0rd!` in the **Password** field.



6. Enable **conversational search** and then click **Save**.



7. Update the conversational search **custom service** settings based on your requirements.

   Note: The **Settings** page is divided into two sections in the following images to enhance the visibility of the screen captures.

   Learn more about these **custom service** settings here.

   The following settings are proven to work well. You can experiment with these settings to see how they affect queries for your client's pilot.

   a. Enable **Conversational search**.

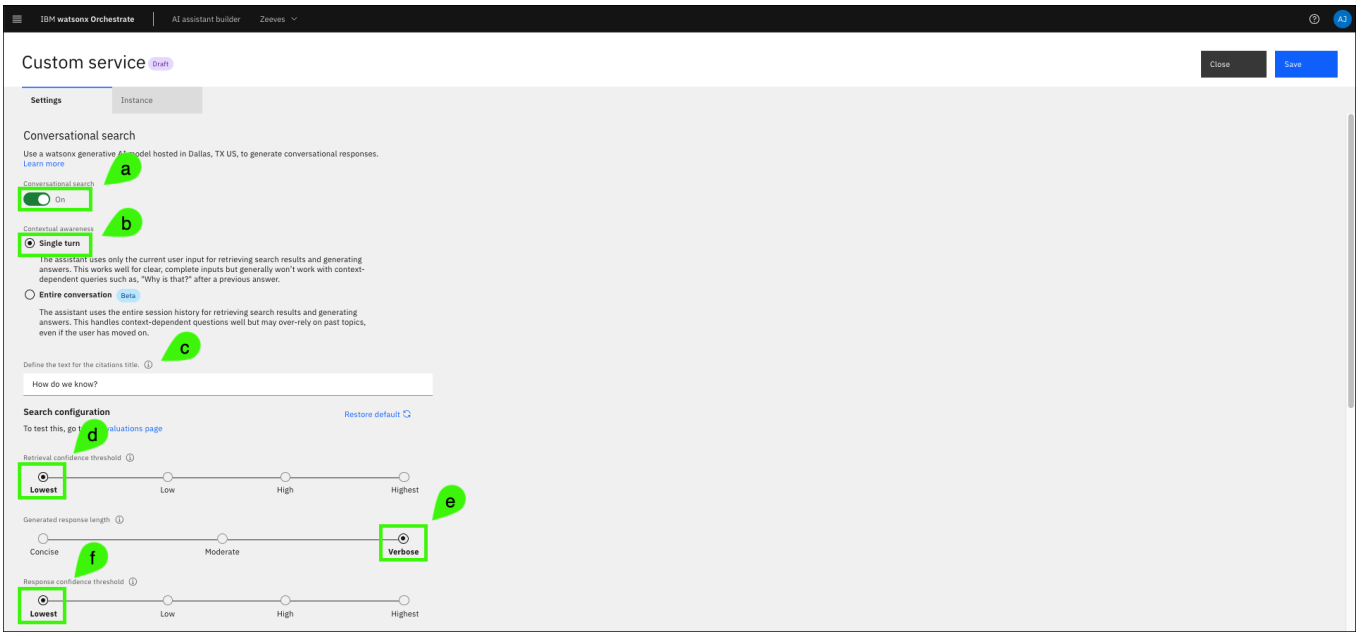
   b. Select **Single turn**. **Multi-turn conversation** (by selecting Entire conversation) is supported by the offering, but has not been fully included in the lab guide. See the callout in the Testing conversational search section below.

   c. Specify the text that appears to instruct the user to expand the list of citations in the assistant (except web chat client).

   d. Select **Lowest** for the **retrieval confidence threshold** setting. This setting checks the confidence of the retrieved citations before a response is generated.

e. Select **Verbose** for the **generated response length**. This setting affects the average response length. Depending on user input, variations from the selected length can occur.

f. Select **Lowest** for the **response confidence threshold**. This setting checks the confidence of the generated citations after the response is generated.

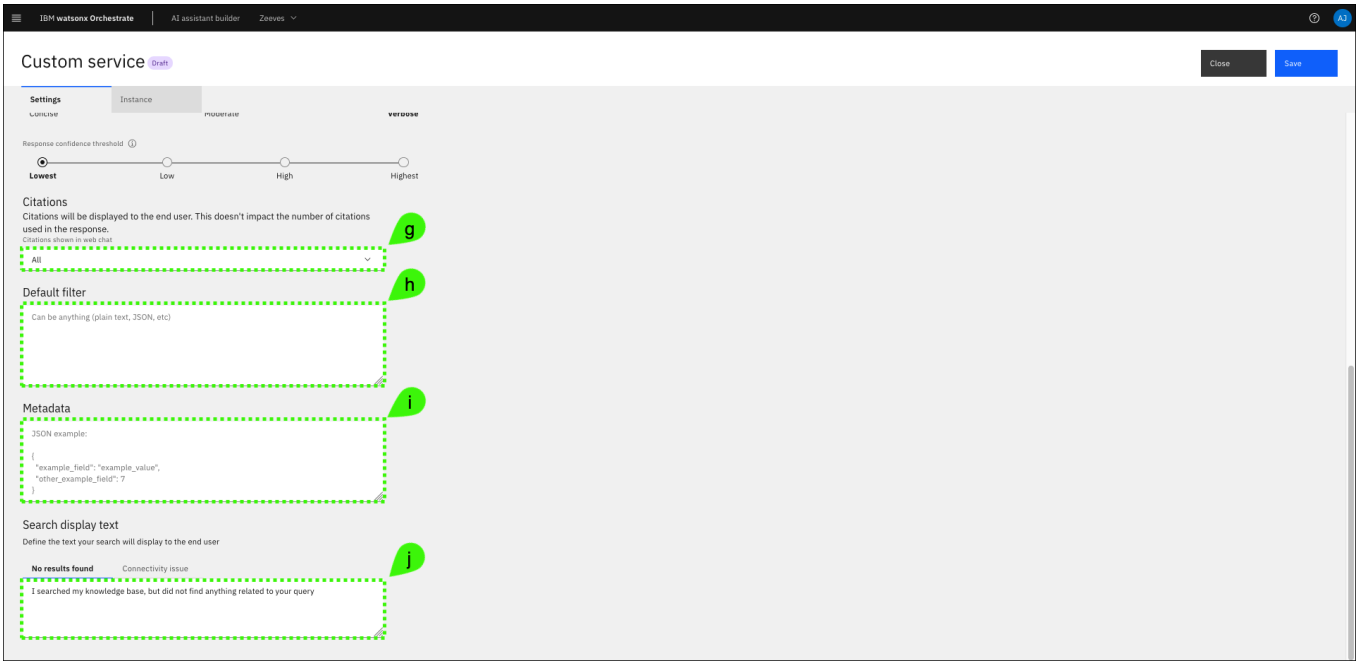


g. Keep the default setting of **All** for the listing of citations.

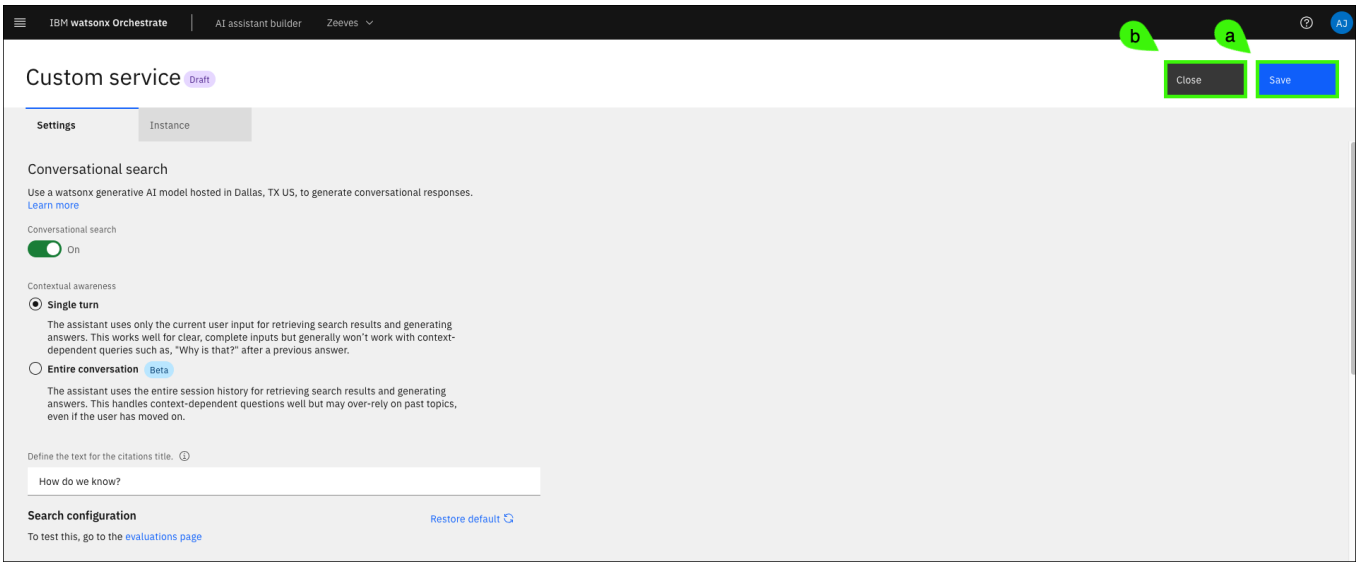
h. Keep the **Default filter** field empty.

i. The **Metadata** field provides a way to adjust your assistant's behavior during conversational search for your OpenSearch instance. This option is explored in detail in the Installing and using zassist to ingest client documents. Leave the field empty for now.

j. The **Search display text** options specify the default text displayed when no results are found or when connectivity issues to the backend search service occur. You can keep the defaults or customize the service.



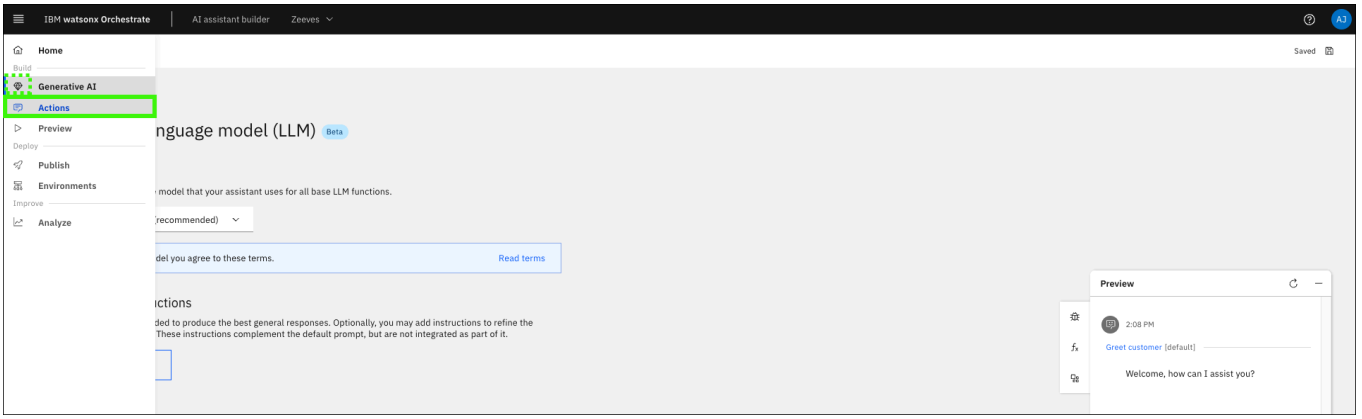
8. Click **Save** (**a**) and then click **Close** (**b**).



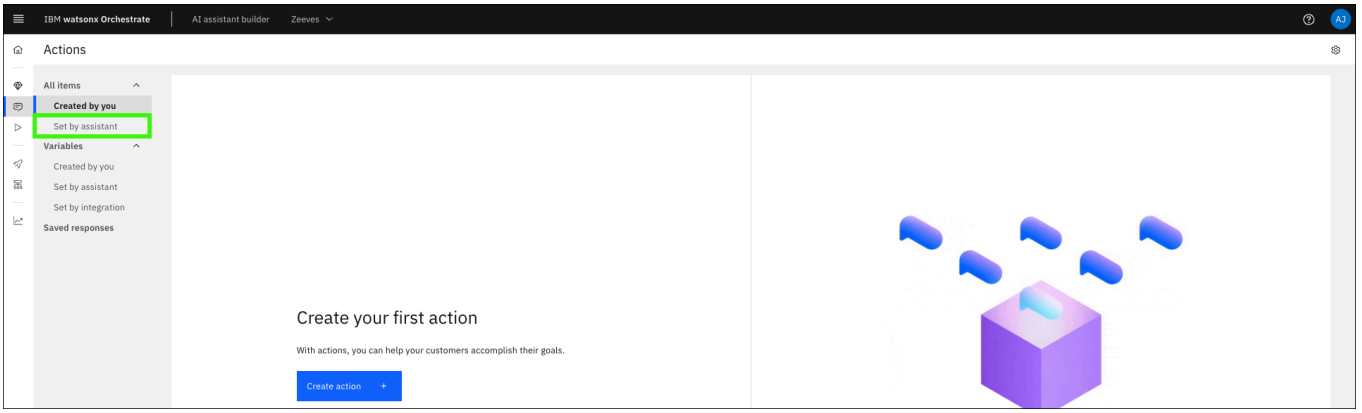
Complete the configuration

After you save and close the **Conversational search** configuration page, a few more configurations are needed to get the best experience from your conversational chat. Details on these settings are available here.

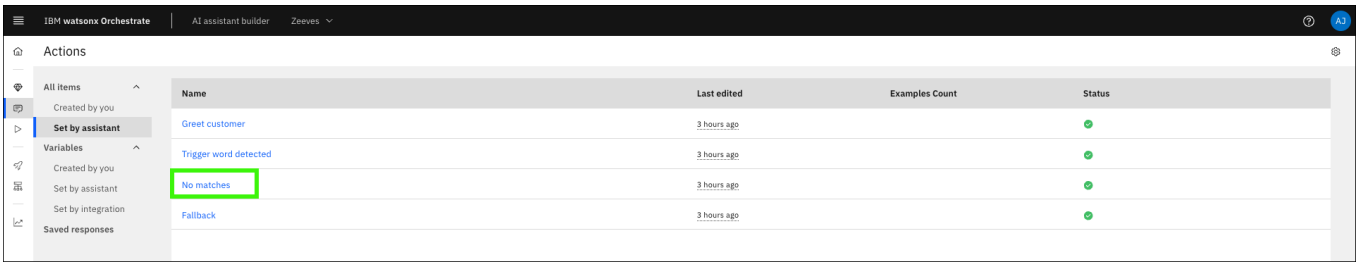
1. Hover over the **Generative AI** icon ( ) in the left navigation and click **Actions**.



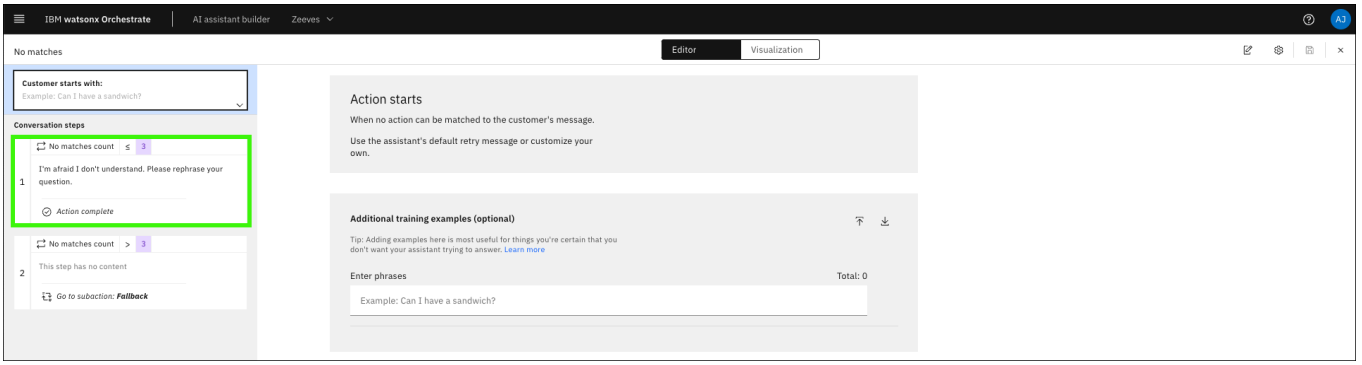
2. Click **Set by assistant** under the **All items** menu.



3. Click **No matches**.

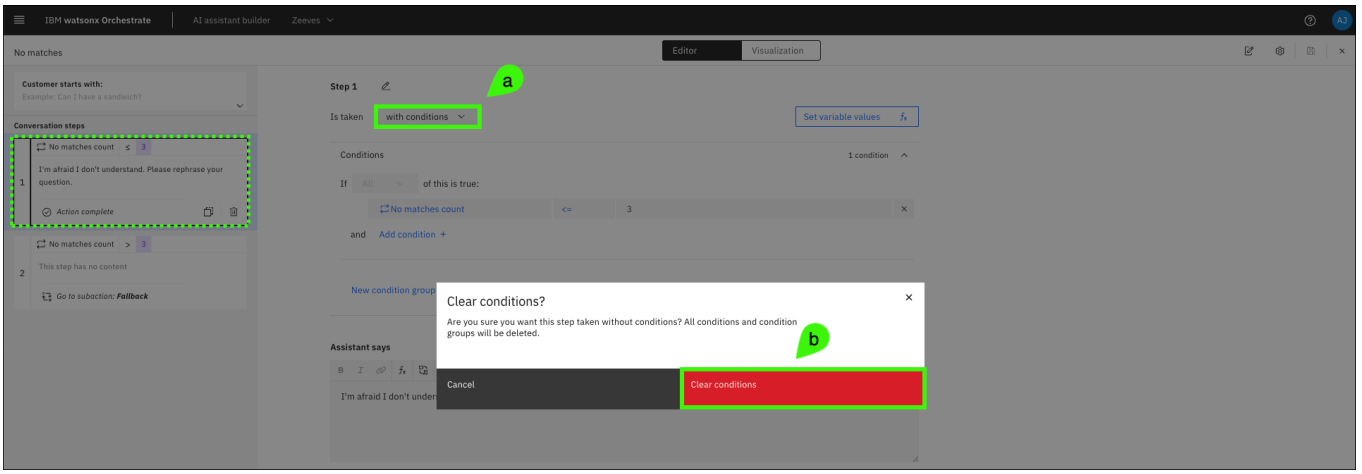


4. Click **Step 1** under **Conversation steps**.

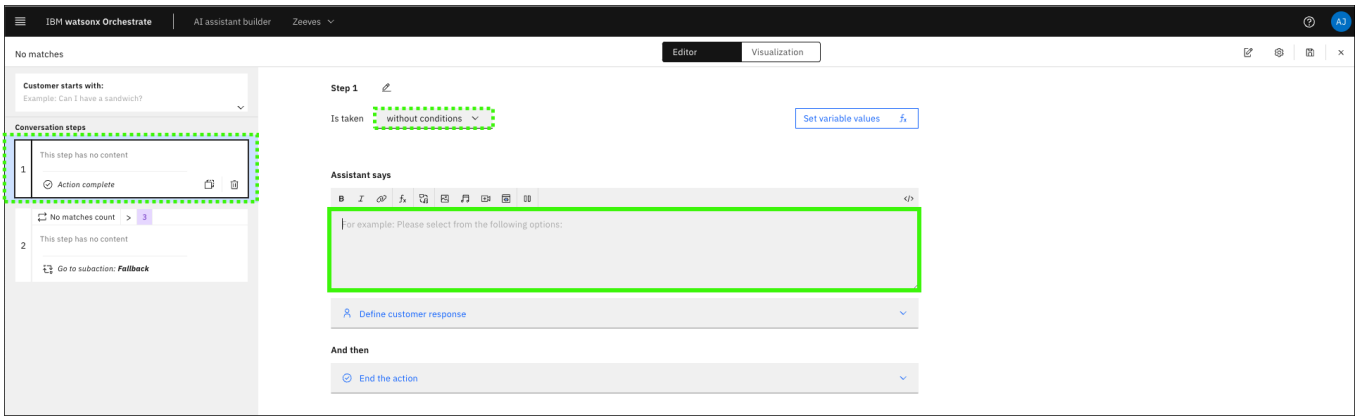


5. Select **without conditions** (**a**) in the **Is taken** drop-down menu and then click **Clear conditions** (**b**).

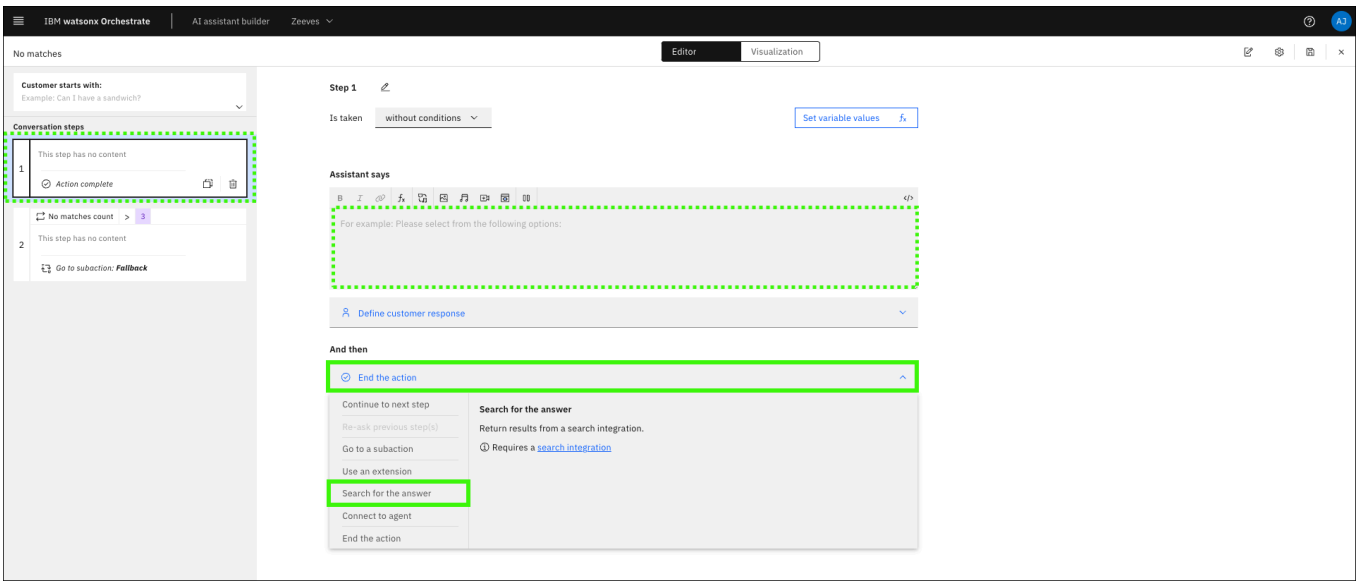
   Note: the **Is taken** value does not change from **with conditions** after selecting **without conditions**.



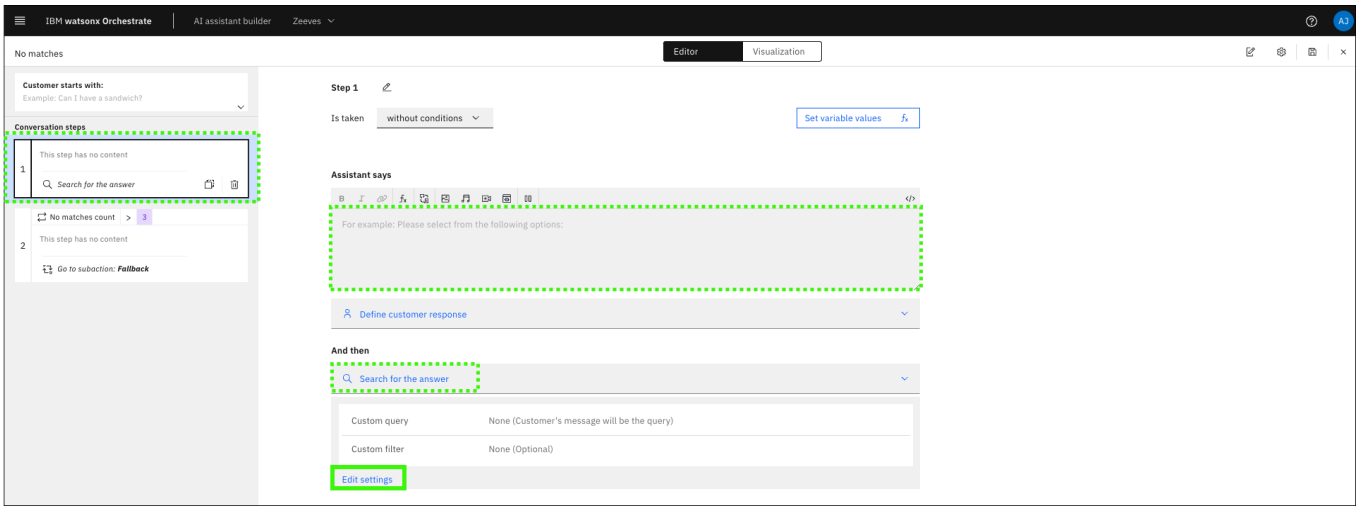
6. Delete the default text in the **Assistant says** entry field.



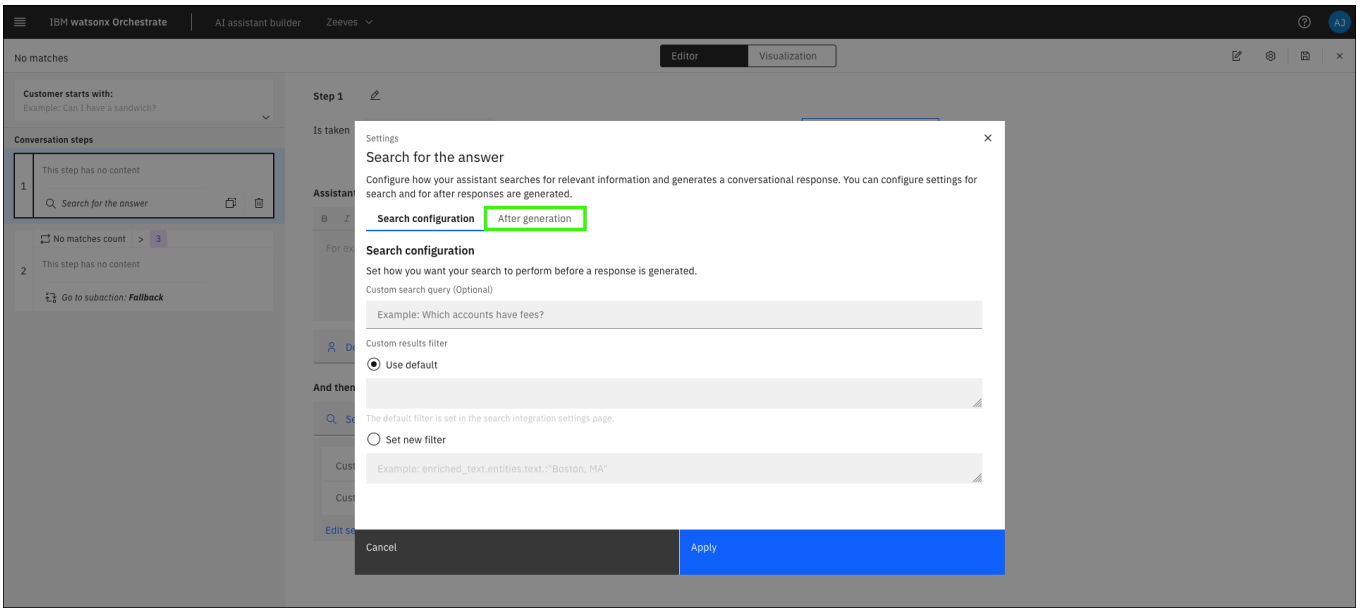
7. Expand the **And then** drop-down menu and select **Search for the answer**.



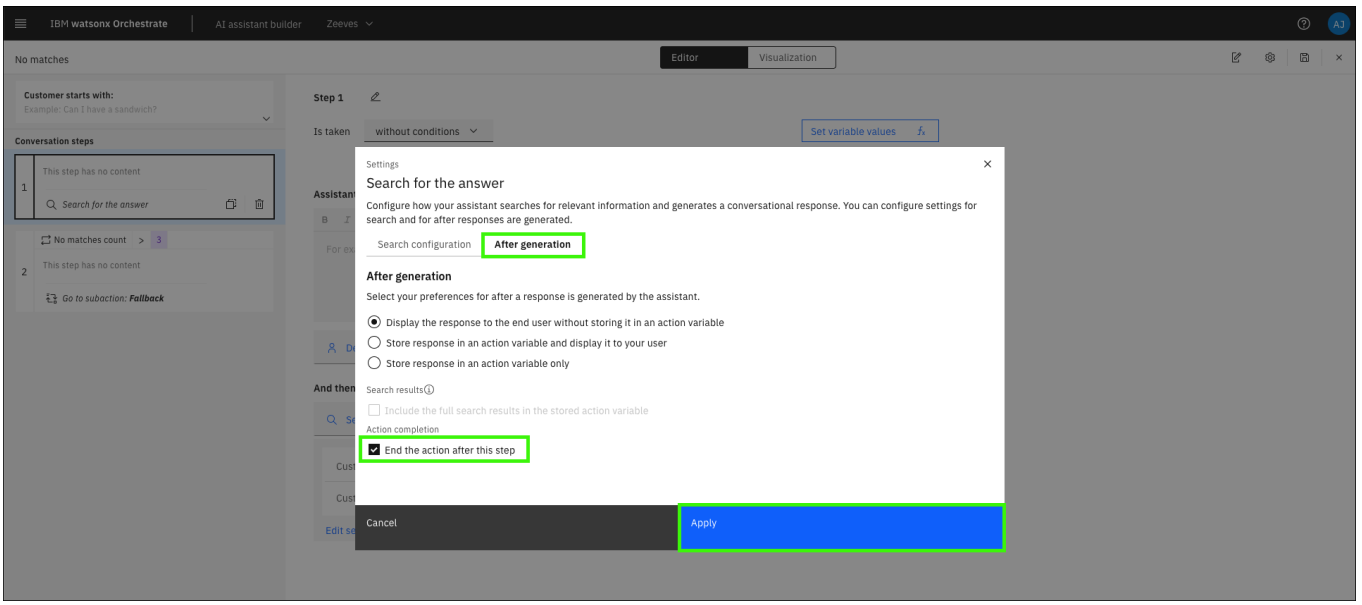
8. Click **Edit settings**.



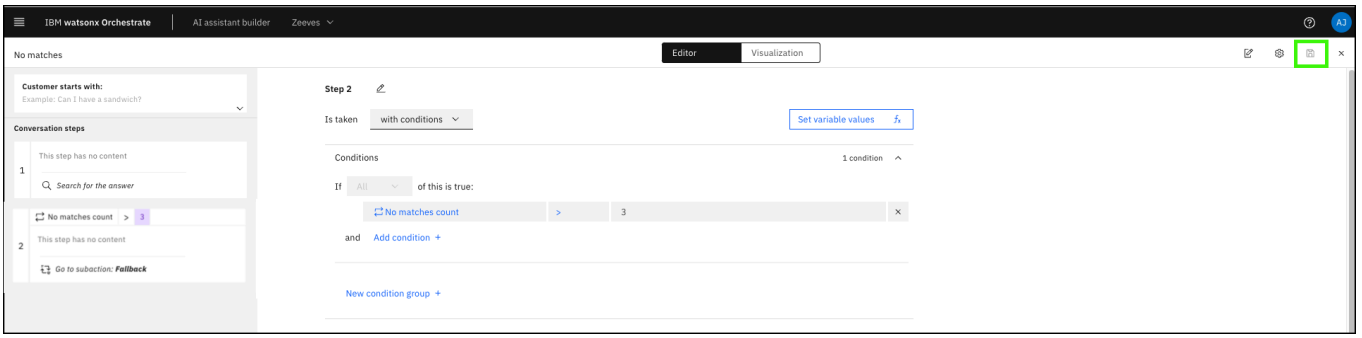
9. Click **After generation**.



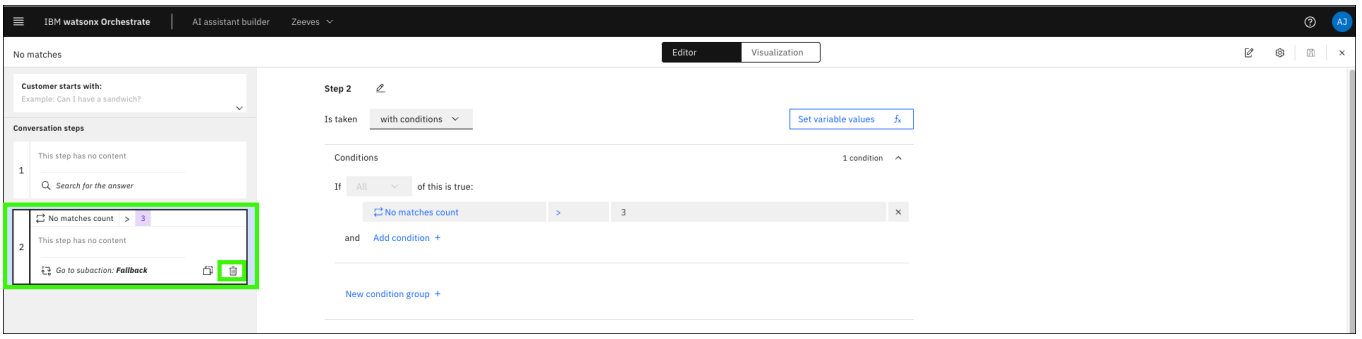
10. Select **End the action after this step** and then click **Apply**.



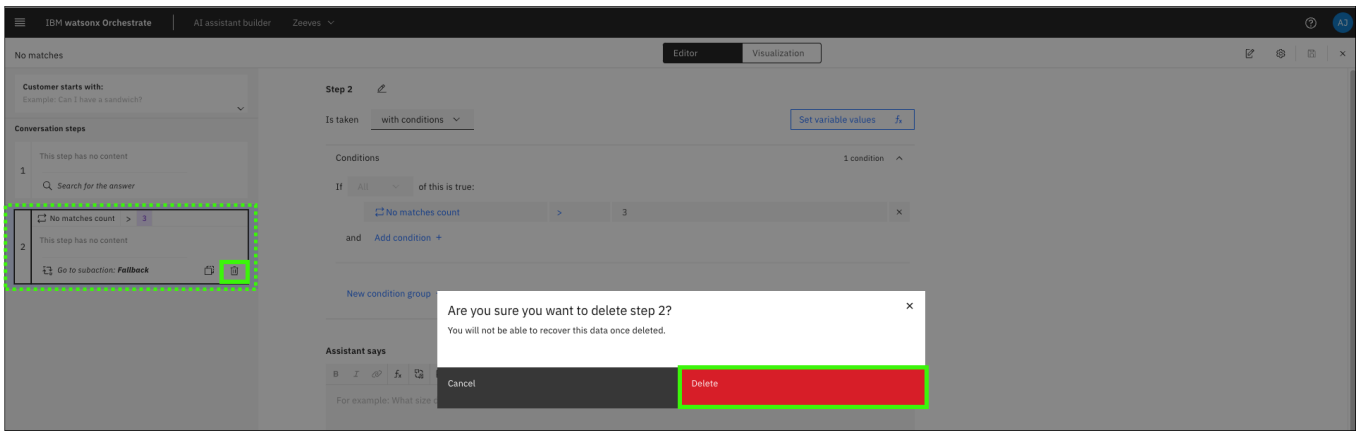
11. Click **Save** ( ).



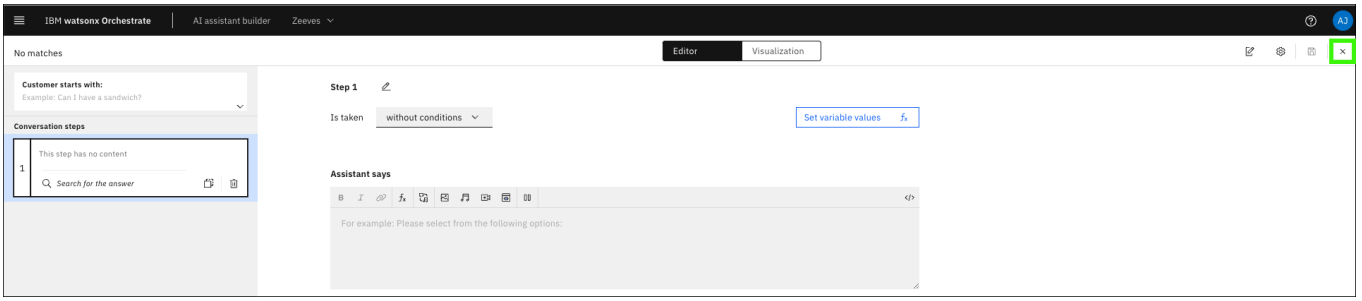
12. Select **Step 2** (**No matches count**) under **Conversation steps** and click **delete** ( ).



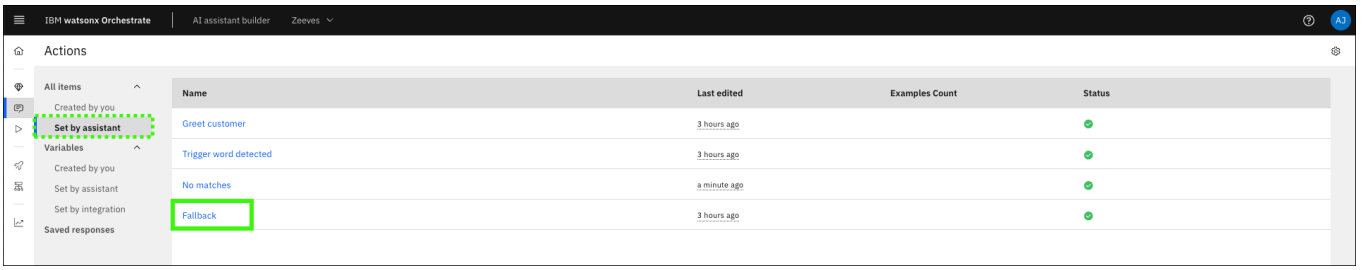
13. Click **Delete** in the confirmation dialog to delete **Conversation step 2**.



14. Click **Close** (the **x** icon) the **Editor** window.

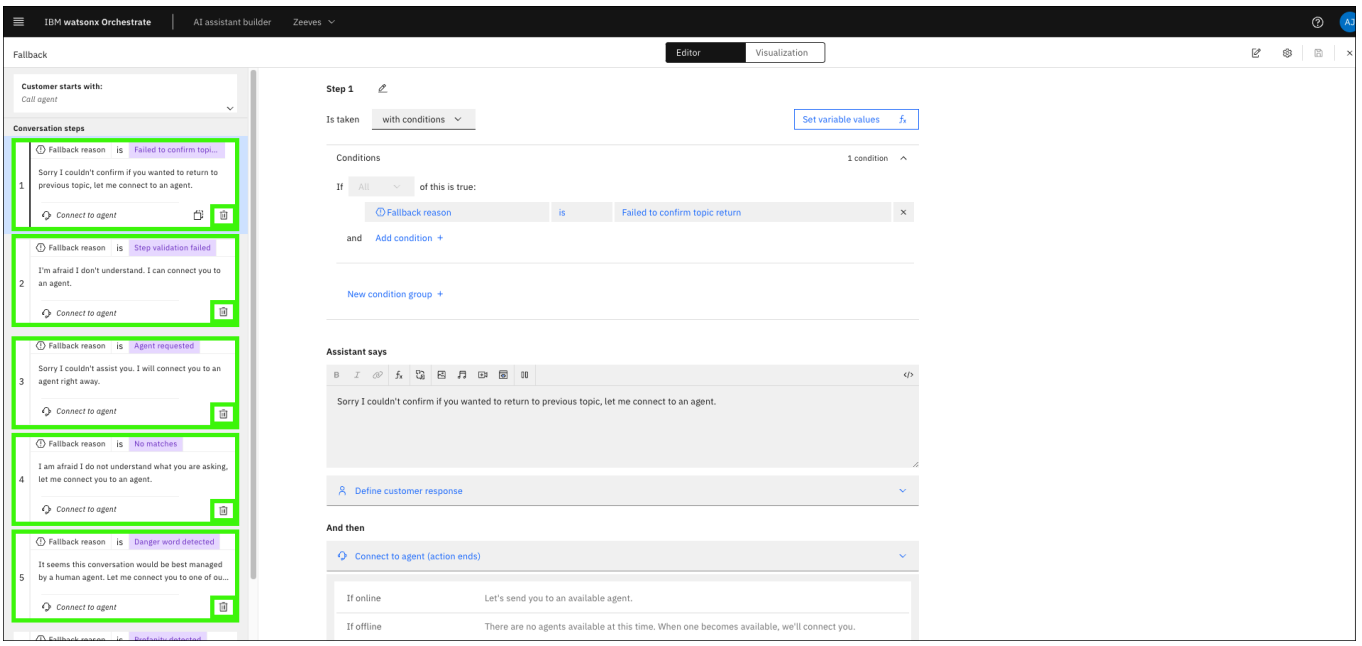


15. Click **Fallback** in the **Actions** table.

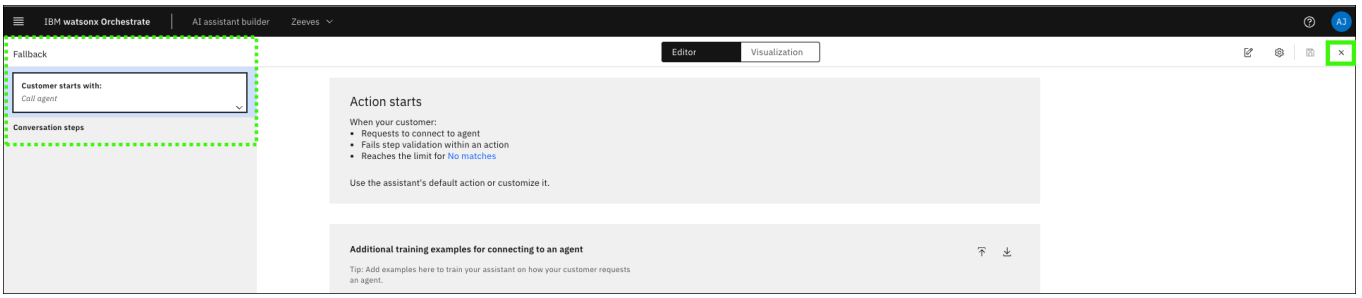


16. Delete **all** of the **Conversation steps**.

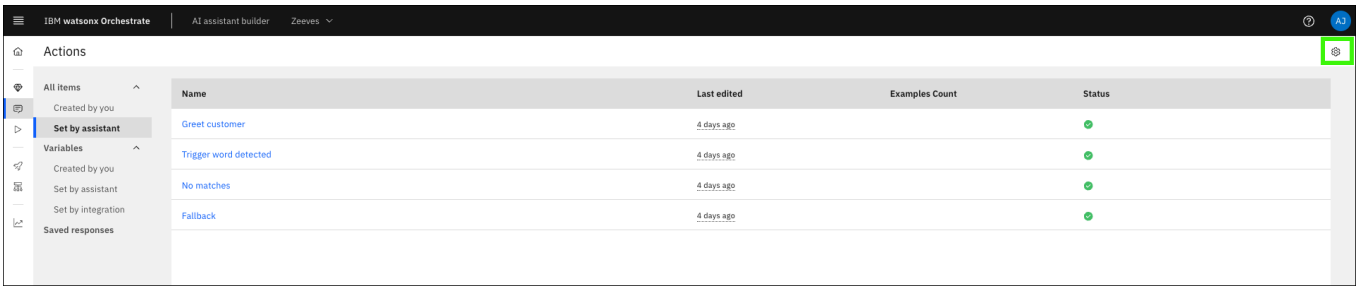
    Note: The following image is edited. Only five steps are shown, but all six need to be deleted. You need to select each step individually. Click **delete** (🗑) and confirm the deletion.



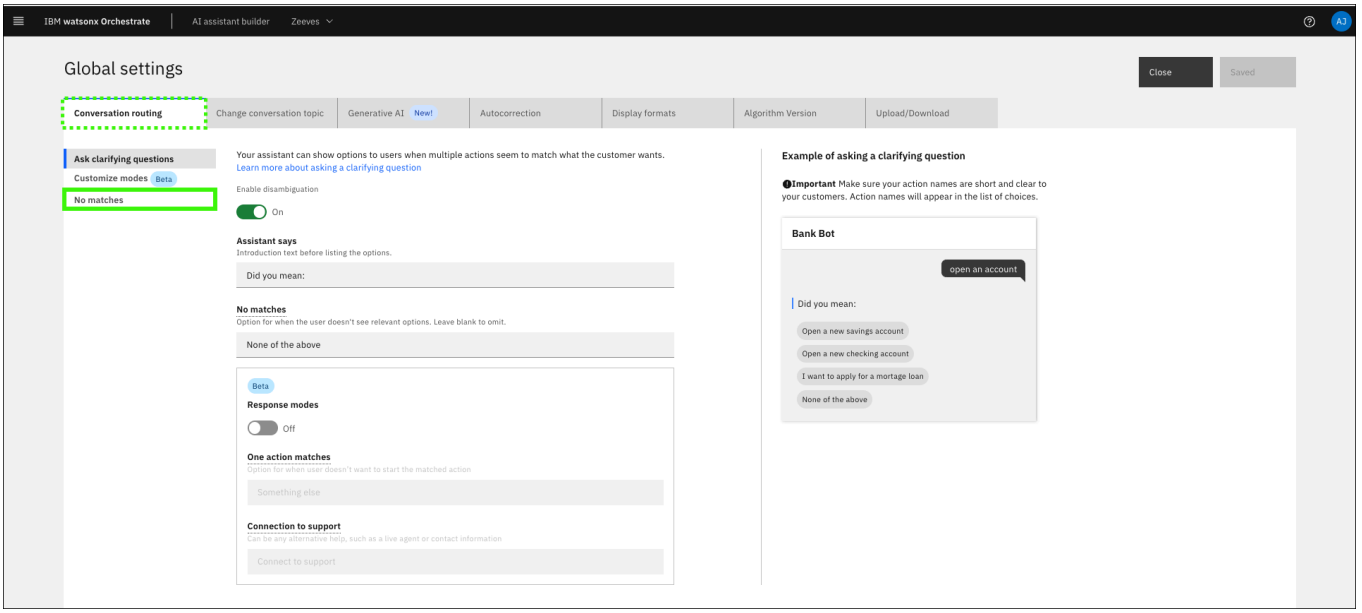
17. Verify that all **Conversation steps** are deleted and then click the **x** to close the **Editor** window.



18. Click the **Global settings** ( ).

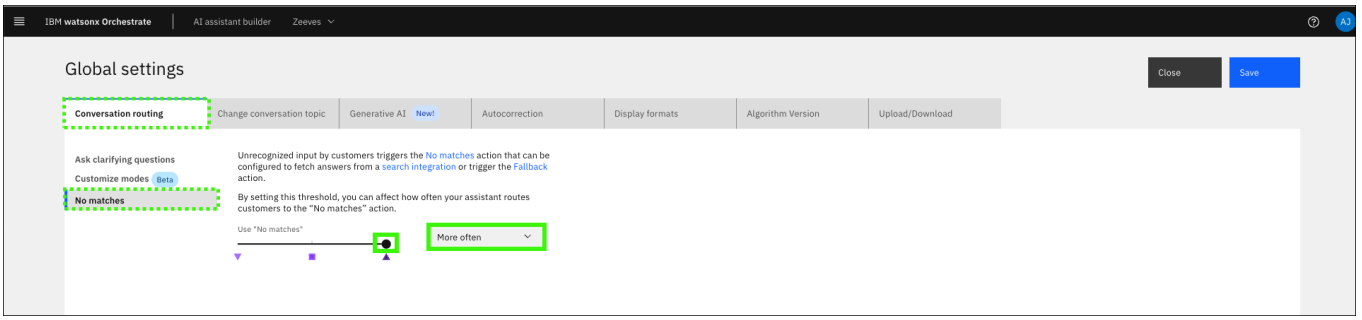


19. Click **No matches** under the **Conversation routing** tab.

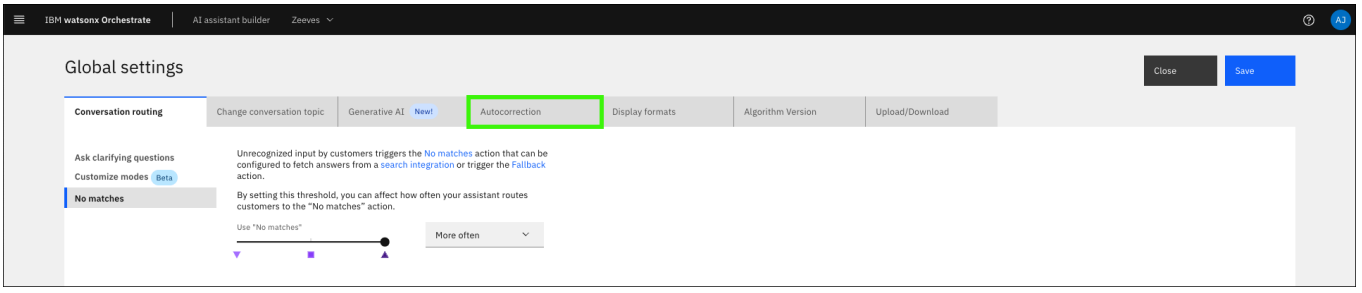


20. Move the slider to **More often** (or select **More often** in the drop-down).

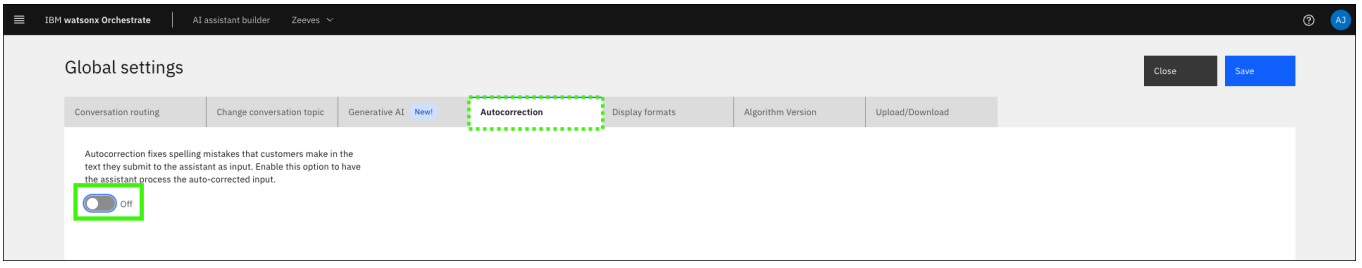
    The setting helps ensure that actions are triggered less often unless the user's query specifically matches the action's input.



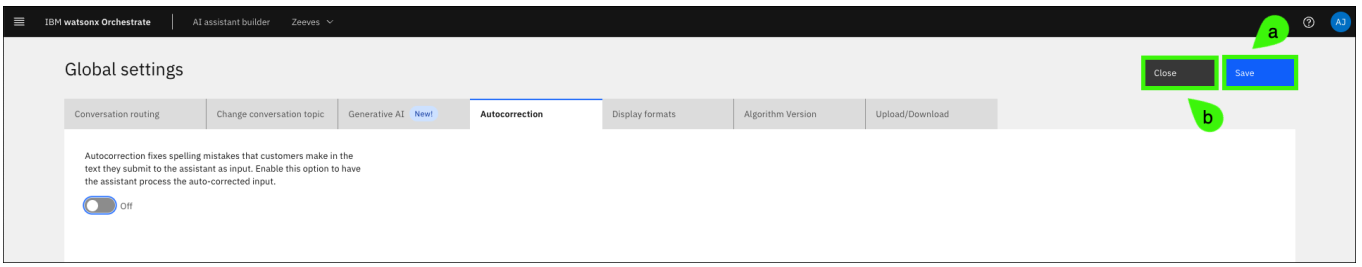
21. Click **Autocorrection**.



22. Click the autocorrection toggle to turn the feature **Off**.

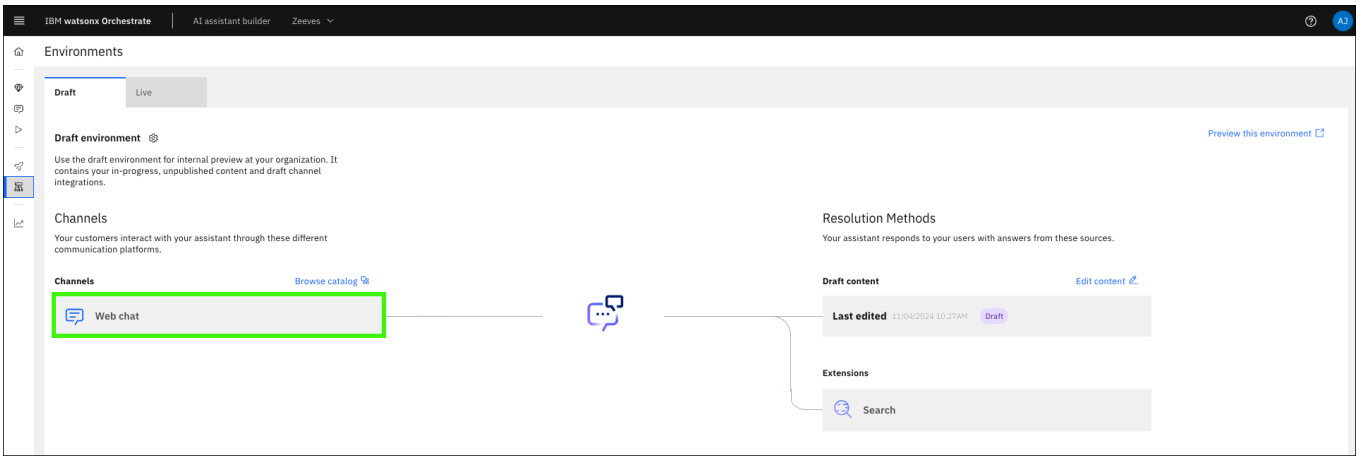


23. Click **Save** (**a**) and then **Close** (**b**).

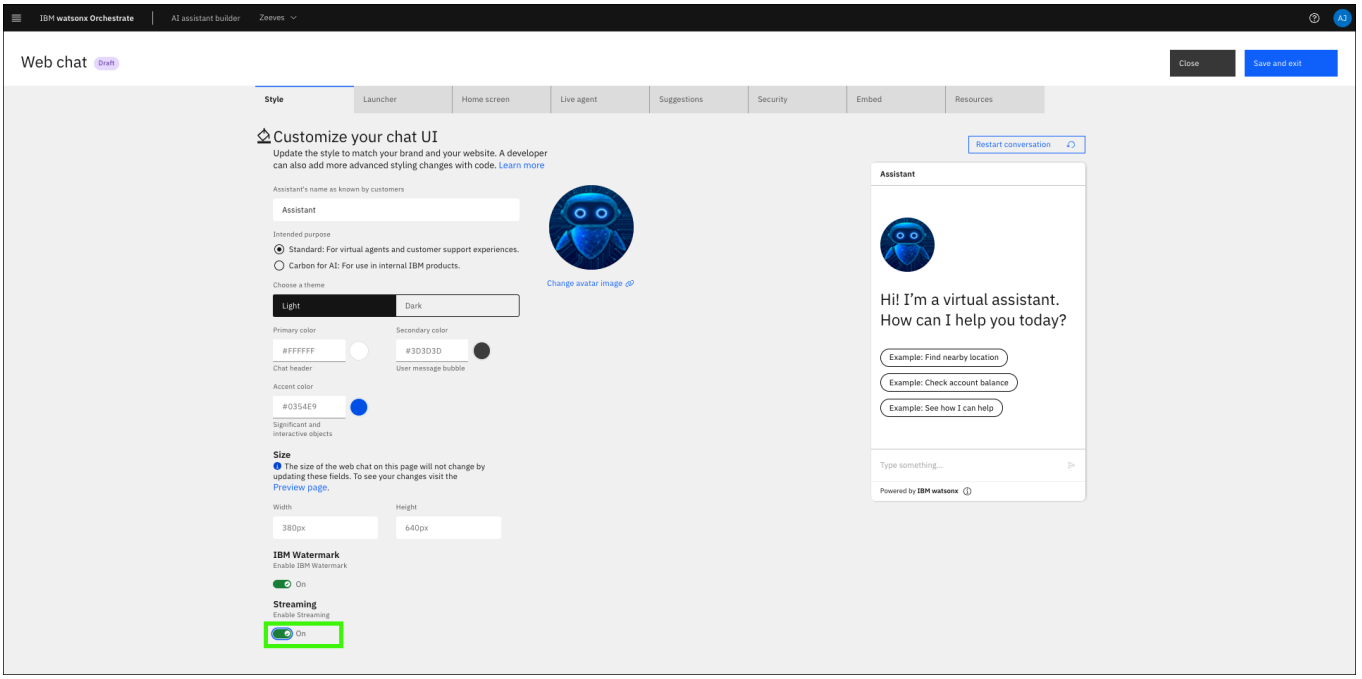


24. Hover over the **Home** ( ) and click **Environments**.

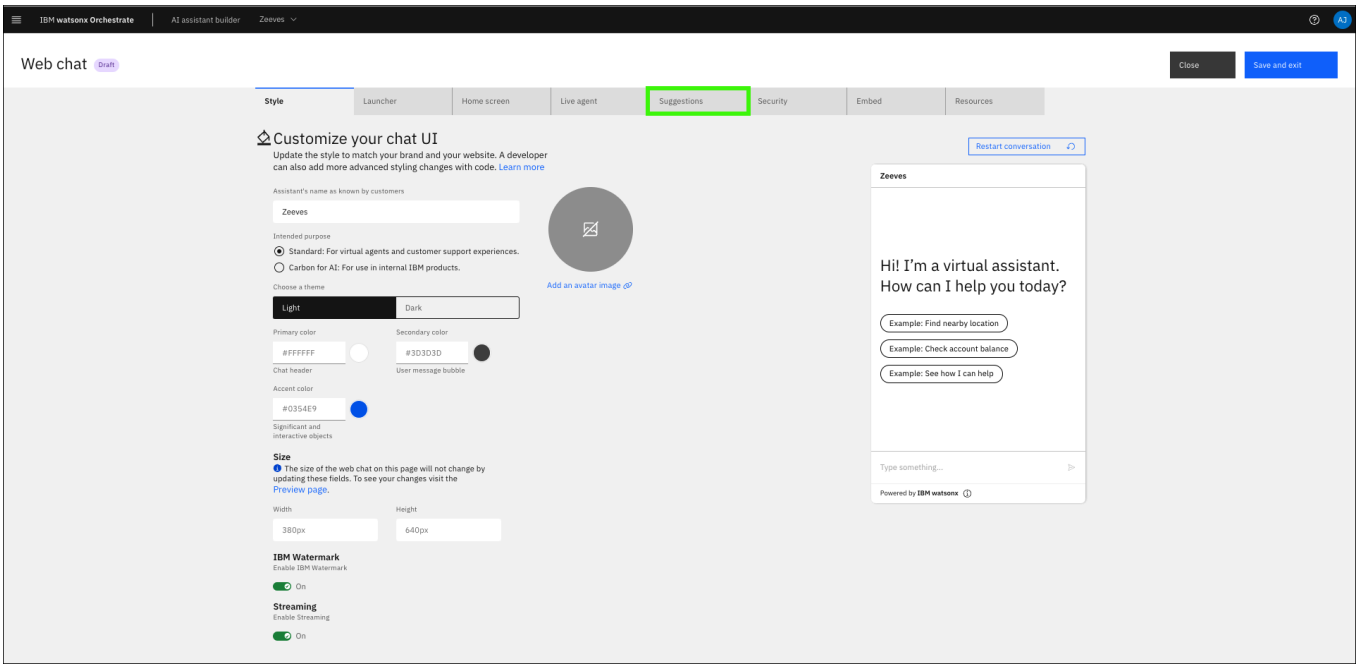
25. Click **Web chat**.



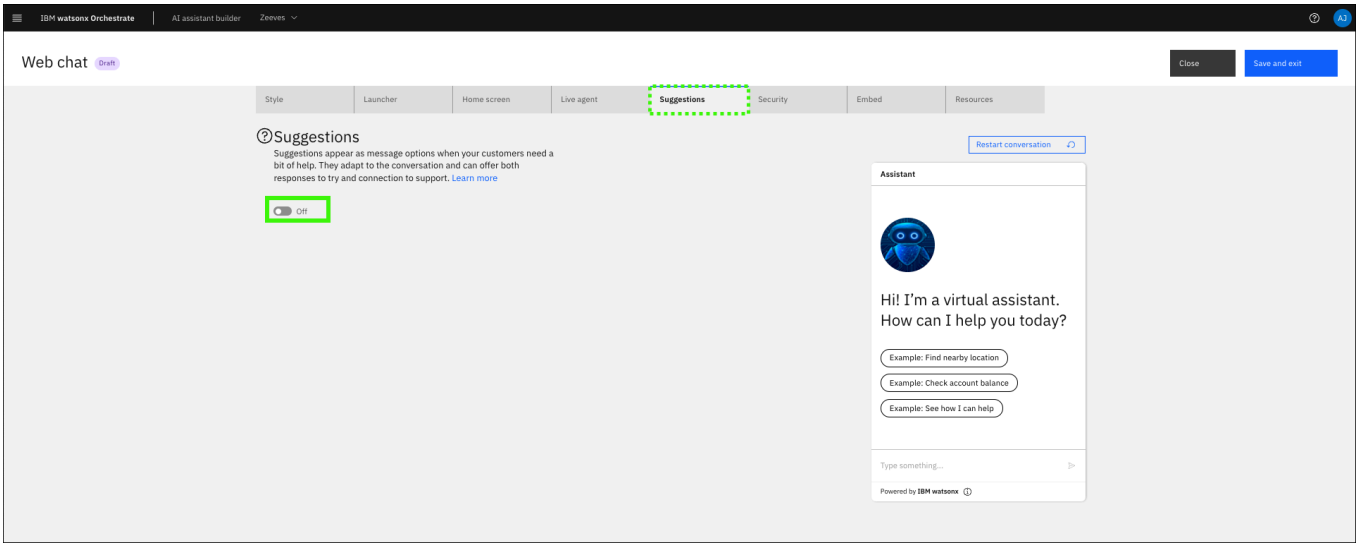
26. On the **Style** tab, click the **Streaming** toggle to enable streaming.



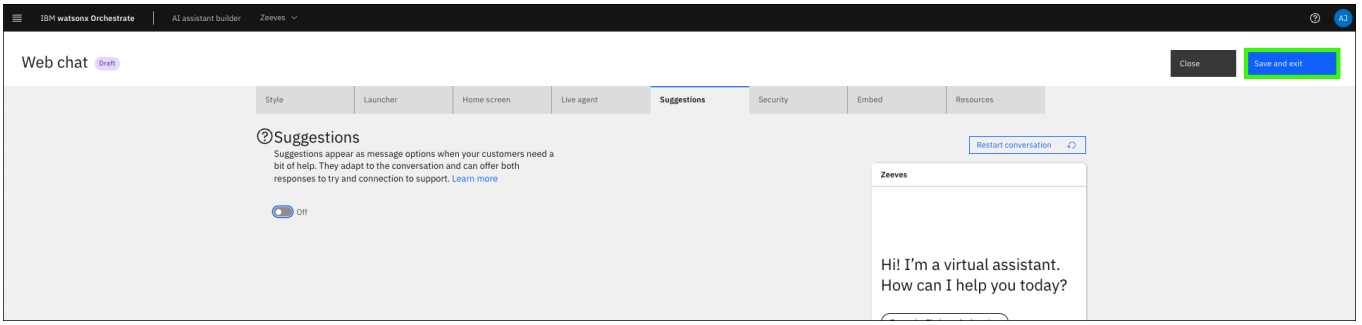
27. Click **Suggestions**.



28. Click the **Suggestions** toggle to turn this feature **Off**.



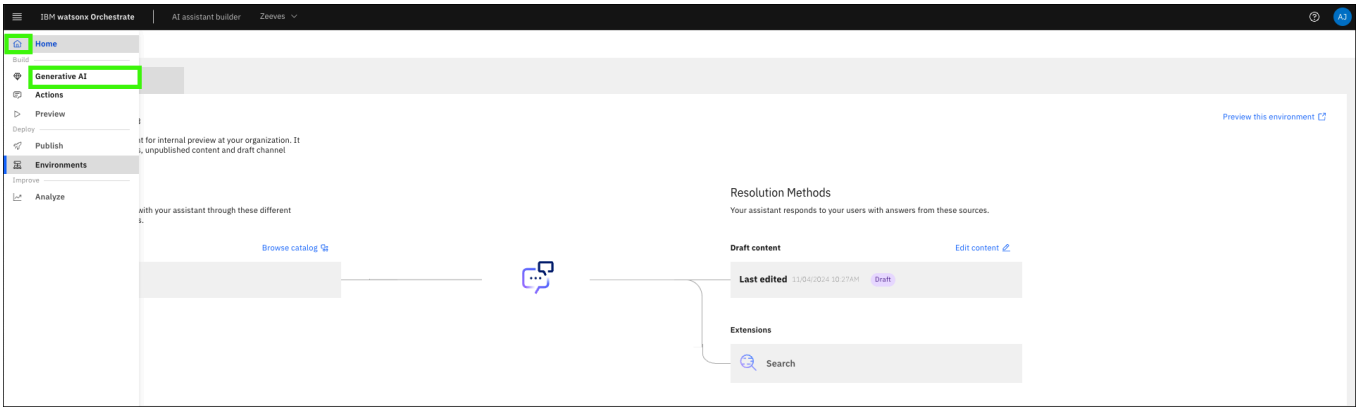
29. Click **Save and exit**.



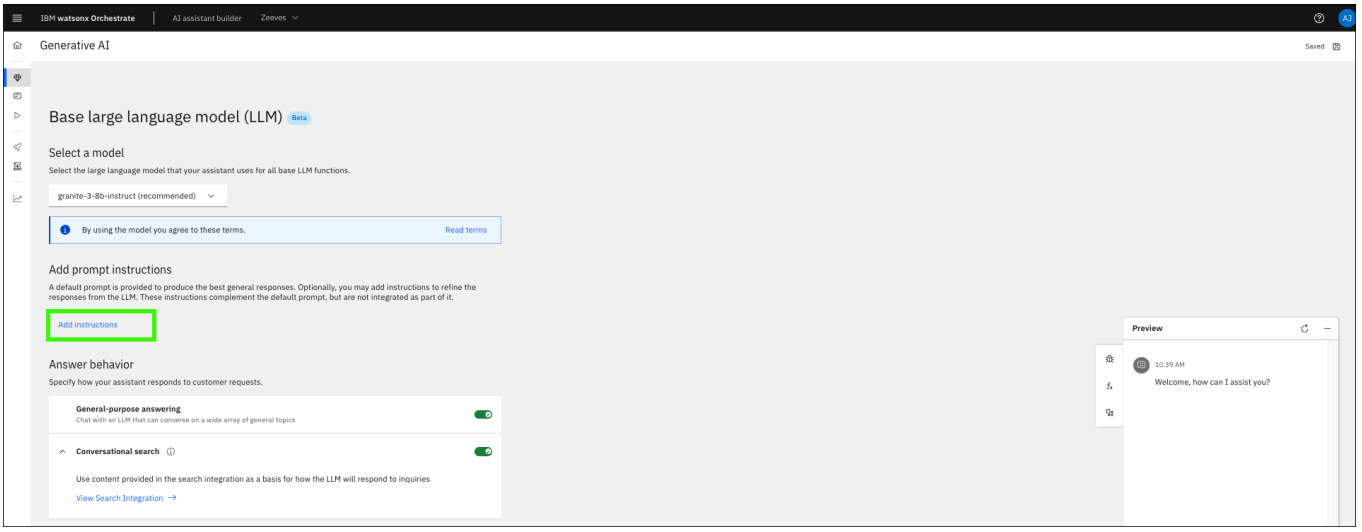
Configure the base large language model

After the preceding steps are completed, there are enhancements you can make to configure how the large language model (LLM) responds to your queries including adding prompt instructions and configuring the LLM's answer behavior. These options can be summarized here.

1. Hover over the **Home** ( ) and click **Generative AI**.



2. Click **Add instructions**.



3. Enter a **prompt instruction**.

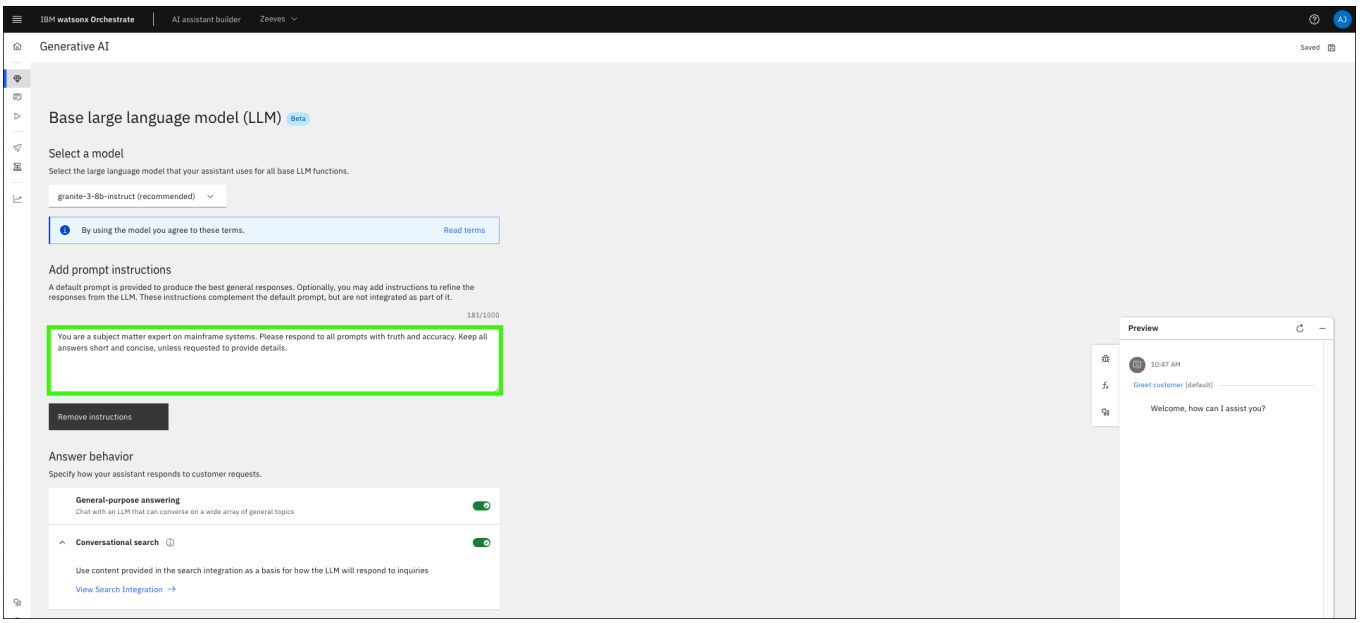
   Your assistant's LLM gives refined responses by following the prompt's instructions, which clarify how to achieve the end-goal of an action.

   Enter prompt instructions in the field. The maximum number of characters you can enter in the prompt instruction field is 1,000.

   The following is an example prompt instruction that works well. Experiment with different prompt instructions.

```
You are a subject matter expert on mainframe systems. Please respond to all prompts with
truth and accuracy. Keep all answers short and concise, unless requested to provide details.
```

   Note: When the instructions are typed in, they are automatically saved and the LLM is immediately trained on them.



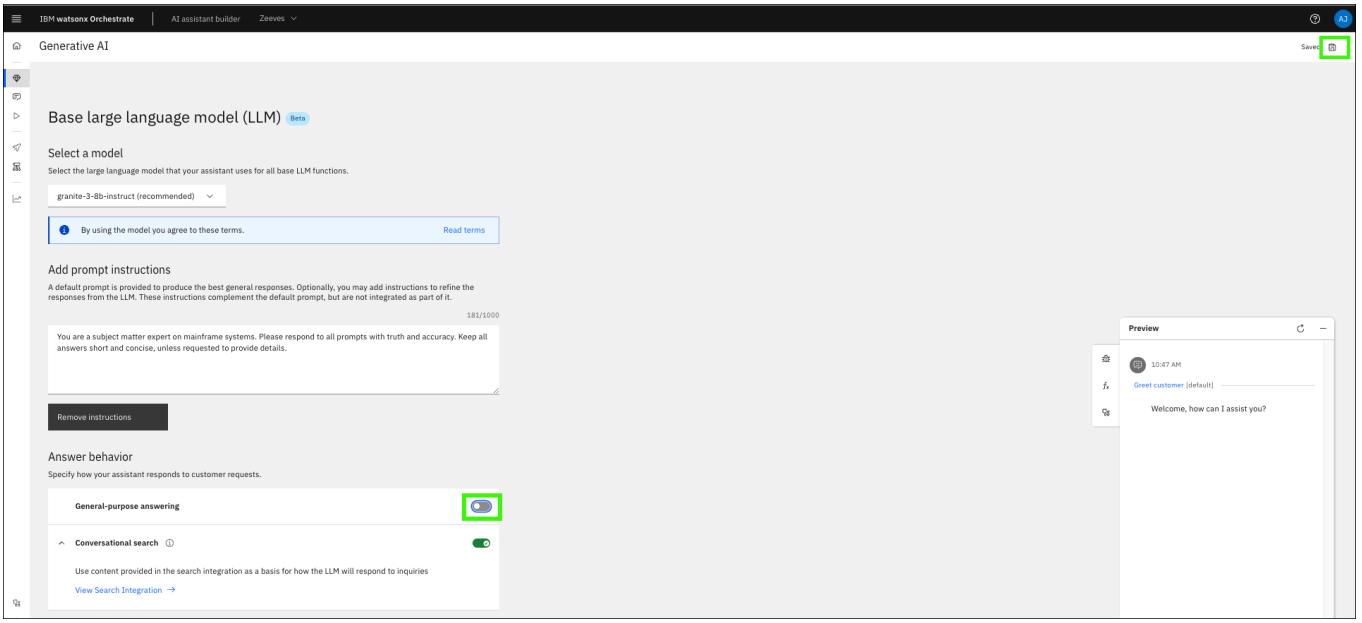
4. Toggle **General-purpose answering** to **Off** and then click **Save** ( ).

   The ability exists to configure the answering behavior of your assistant to provide responses that are based on the preinstalled content or general content.

   On the **Generative AI** page (under **Prompt Instructions**), you see the **Answer behavior** section. After you configure **Conversational search**, you see that it is enabled (toggled on) with the search integration added.

   If you enable both general-purpose answering as well as conversational search, the conversational search answering takes precedence over General-purpose answering.

   Recommendation: For purposes of retrieving Z-specific answers and responses, it is recommended that you turn off general-purpose answering and leave only conversational search turned on.



Testing conversational search

Now you can begin issuing queries to test the assistant's responses.

Important: Modify settings iteratively based on your assessment of response quality. Review and change them at any time. For example, add extra prompt instructions, change response verbosity, and modify OpenSearch indexes.

1. Hover over the **Home** ( ) and click **Preview**.

2. Experiment with different prompts and validate that the answers are reasonable and related to IBM Z.

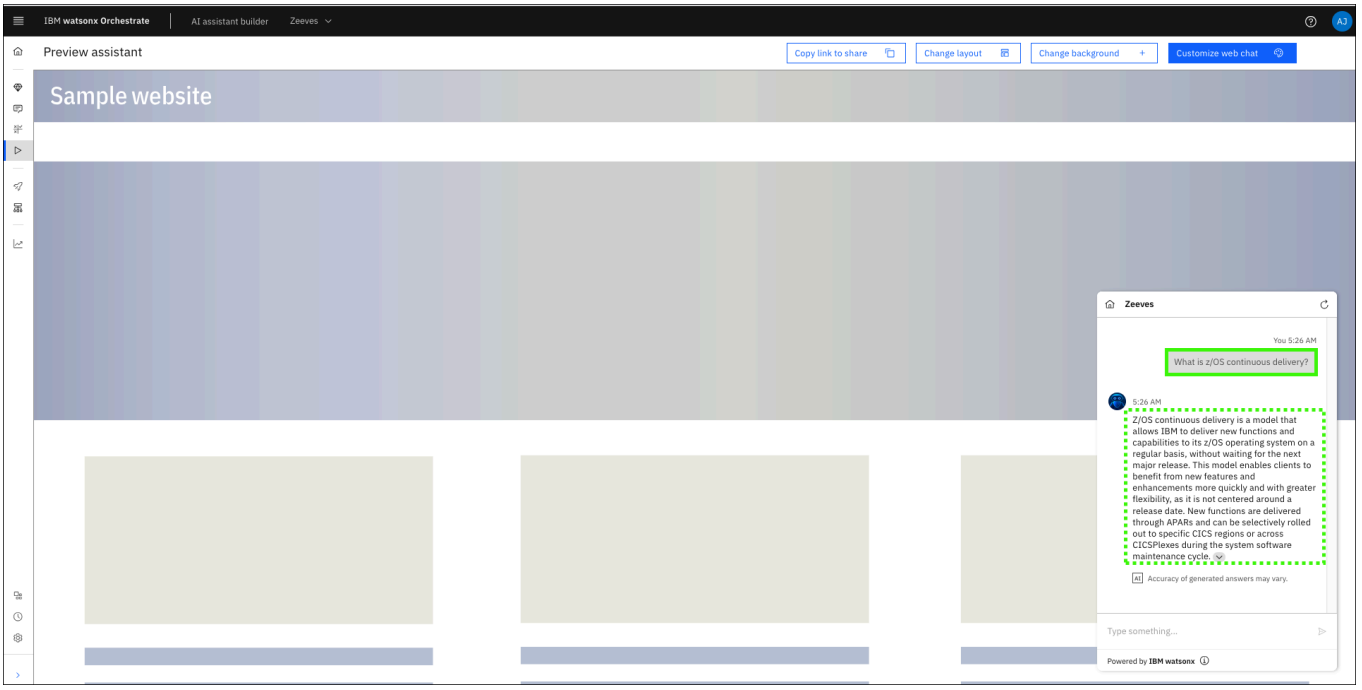
   Other prompts and responses follow.

   Note: The responses that you receive can vary from the ones shown.

   Prompt:

   ```
   What is z/OS continuous delivery?
   ```

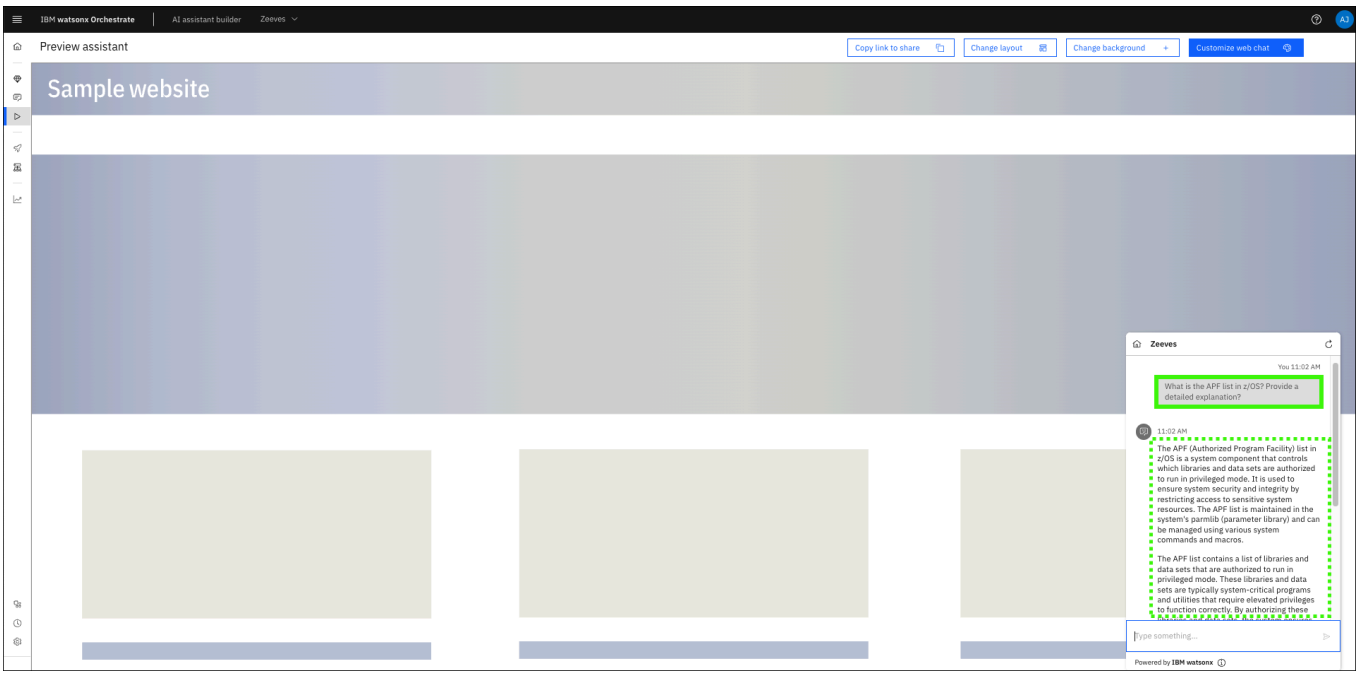
   Example output:



   Prompt:

   ```
   What is the APF list in z/OS? Provide a detailed explanation?
   ```

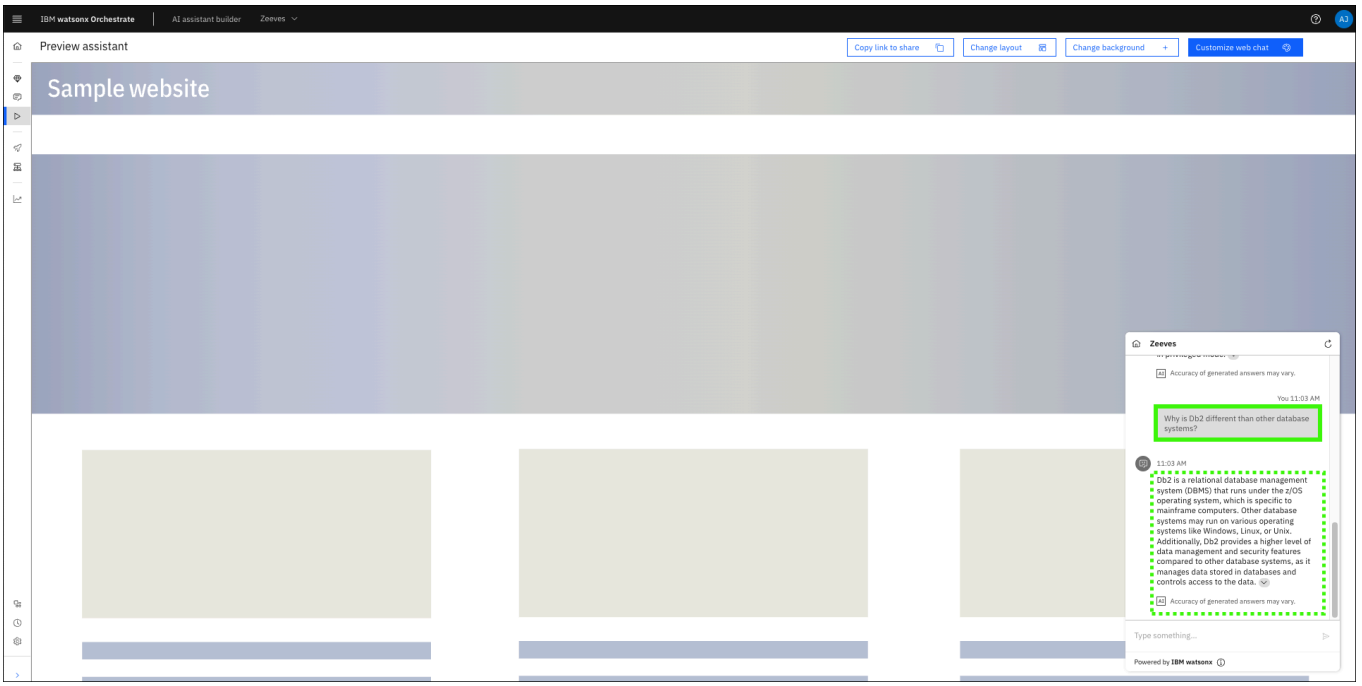
   Example output:



Prompt:

```
Why is Db2 different than other database systems?
```

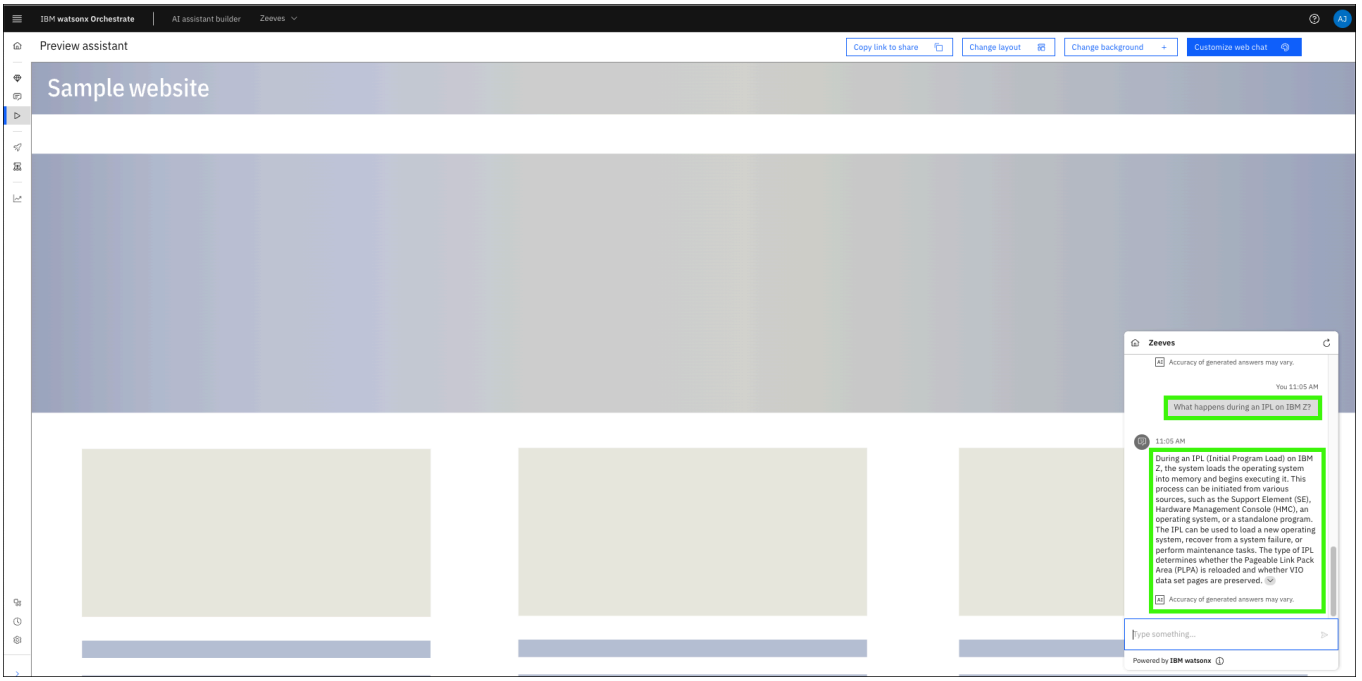
Example output:



Prompt:

```
What happens during an IPL on IBM Z?
```

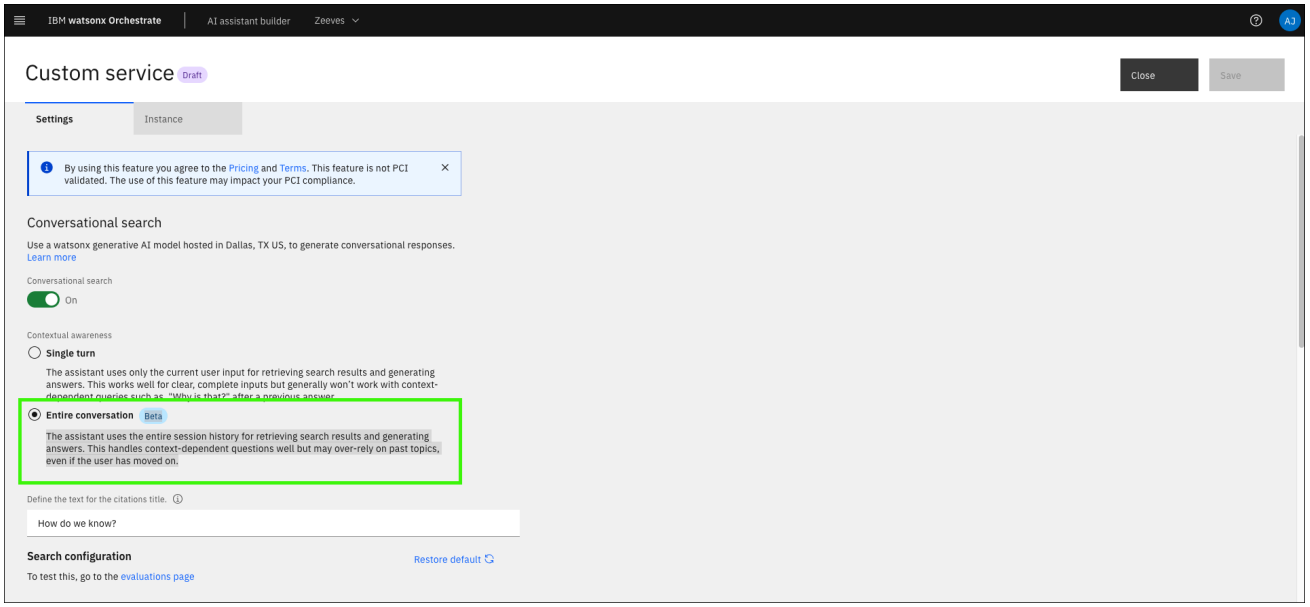
Example output:


IBM watsonx Orchestrate
AI assistant builder
Zeeves
Preview assistant
Copy link to share
Change layout
Change background
Customize web chat
Sample website
Zeeves
Accuracy of generated answers may vary.
You 11:05 AM
What happens during an IPL on IBM Z?
11:05 AM
During an IPL (Initial Program Load) on IBM Z, the system loads the operating system into memory and begins executing it. This process can be initiated from various sources, such as the Support Element (SE), Hardware Management Console (HMC), an operating system, or a standalone program. The IPL can be used to load a new operating system, recover from a system failure, or perform maintenance tasks. The type of IPL determines whether the Pageable Link Pack Area (PLPA) is reloaded and whether VIO data set pages are preserved.
Accuracy of generated answers may vary.
Type something...
Powered by IBM watsonx

Experiment with multi-turn (entire conversation) contextual awareness.

In the December 2024 release of IBM watsonx Assistant for Z support for multi-turn contextual awareness was added. This capability enables the assistant to use an entire session history for retrieving search results and generating answers. This handles context-dependent questions well but may over-rely on past topics, even if the user has moved on.

Experiment with this setting by changing your custom service contextual awareness setting from **Single turn** to **Entire conversation**.



Once enabled, try sequential prompts like:

```
What are some features of z/OS?
```

```
Give me an itemized list?
```

```
Tell me more about item 3.
```

You have a working assistant that uses IBM Watson Assistant for Z. Explore different prompt instructions and settings. If you encounter issues, refer to the Troubleshooting section that follows for resolution.

Continue to the Creating a stand-alone OpenSearch instance for document ingestion to learn how to configure a dedicated OpenSearch instance for ingesting client-specific documentation into the RAG model.

Troubleshooting

The following are issues that you may encounter. If the provided resolutions do not work, contact support by using the methods that are mentioned in the Support section.

Assistant responds to all prompts with, "I might have information related to your query to share, but am unable to connect to my knowledge base at the moment"

This Assistant is unable to connect to the custom service URL specified. This could be a network issue, the service may be down, the service may be restarting, or the service is no longer running at that URL.

Before reaching out to Support, try the following:

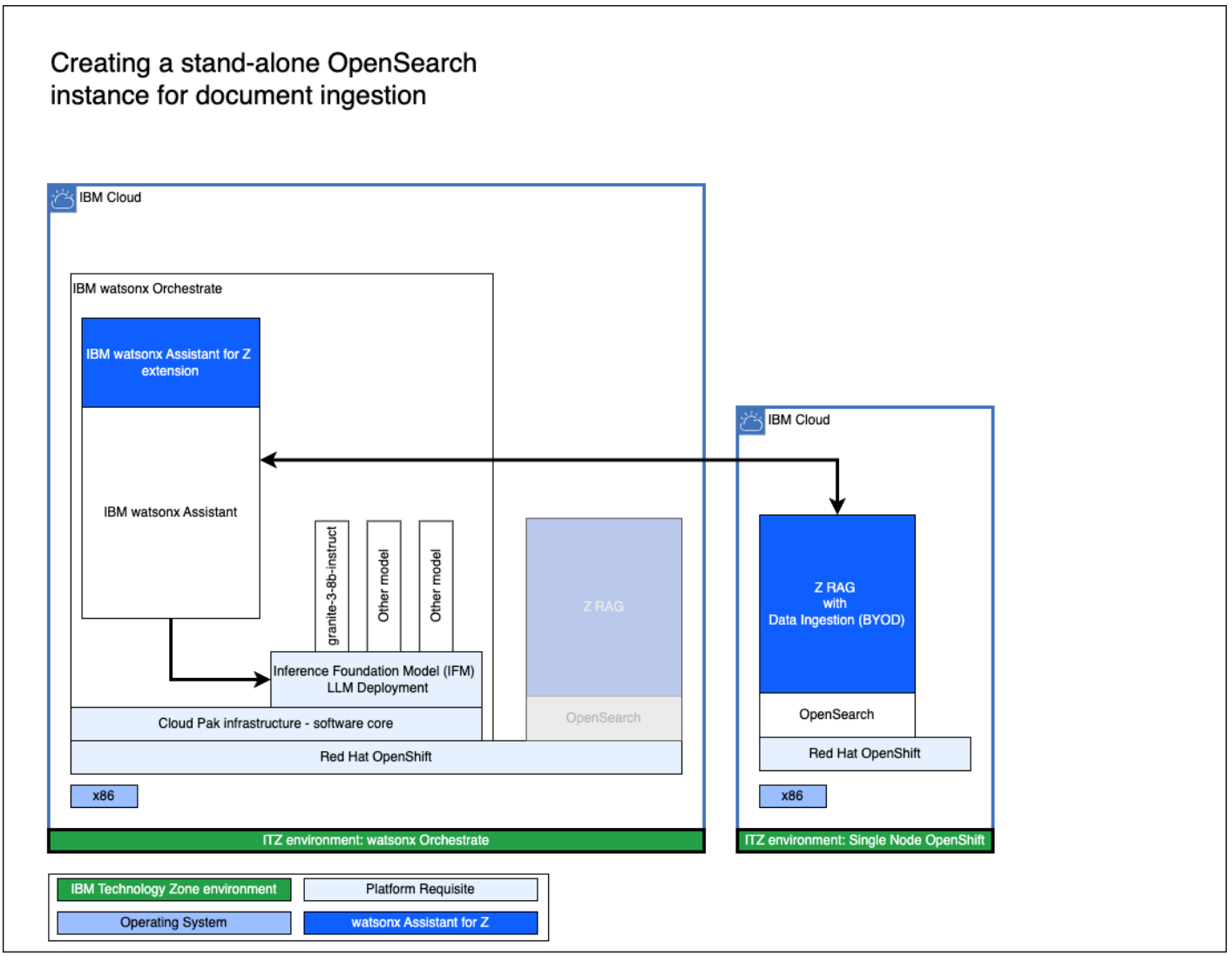
- Wait a few minutes and try again. It may be the service was in the process of restarting.
- If you printed this demonstration guide or saved a copy, verify you are using the most current version of the lab guide and the correct service URL (https://wxa4z-opensearch-wrapper-wxa4z-demo-v2-1-0.wxo4z-opc-opensearch-clus-47e063e6a3ad1f71bf2e58f91c3b4c2e-0000.us-south.containers.appdomain.cloud/v1/query). The URL may have changed since you saved or printed the lab guide.

Ingesting client documents

Creating a stand-alone OpenSearch instance for document ingestion

In this section, learn to enable clients to ingest their own documentation into the Retrieval Augmented Generation (RAG) used by IBM Watson Assistant for Z by deploying a dedicated OpenSearch instance, referred to as bring-your-own-search (BYOS).

Below is a high-level, logical architecture of the environment you will deploy in this section.



Earlier, you provisioned three IBM Technology Zone (ITZ) environments. One of which was a single-node Red Hat OpenShift (SNO) cluster. If you have not reserved this environment, or it is not in the **Ready** state, return to the IBM Technology Zone environment section to complete the reservation.

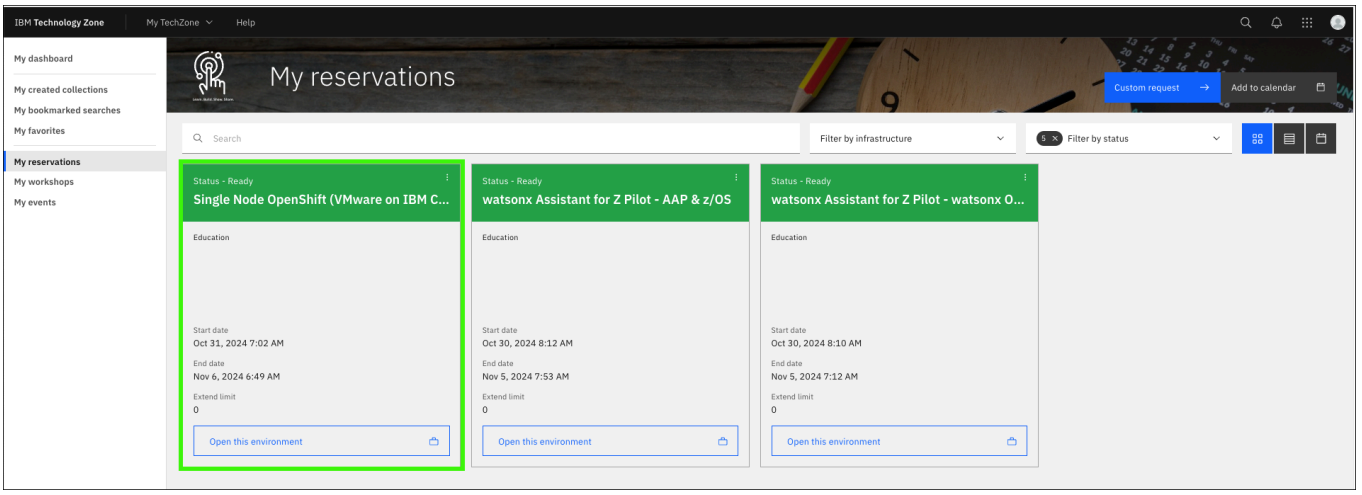
Install the Red Hat OpenShift command-line interface utility

The Red Hat OpenShift command-line interface (CLI) utility, which is known as **oc**, must be installed on your local workstation. If you already installed the **oc** utility, you can proceed to log in to the SNO cluster.

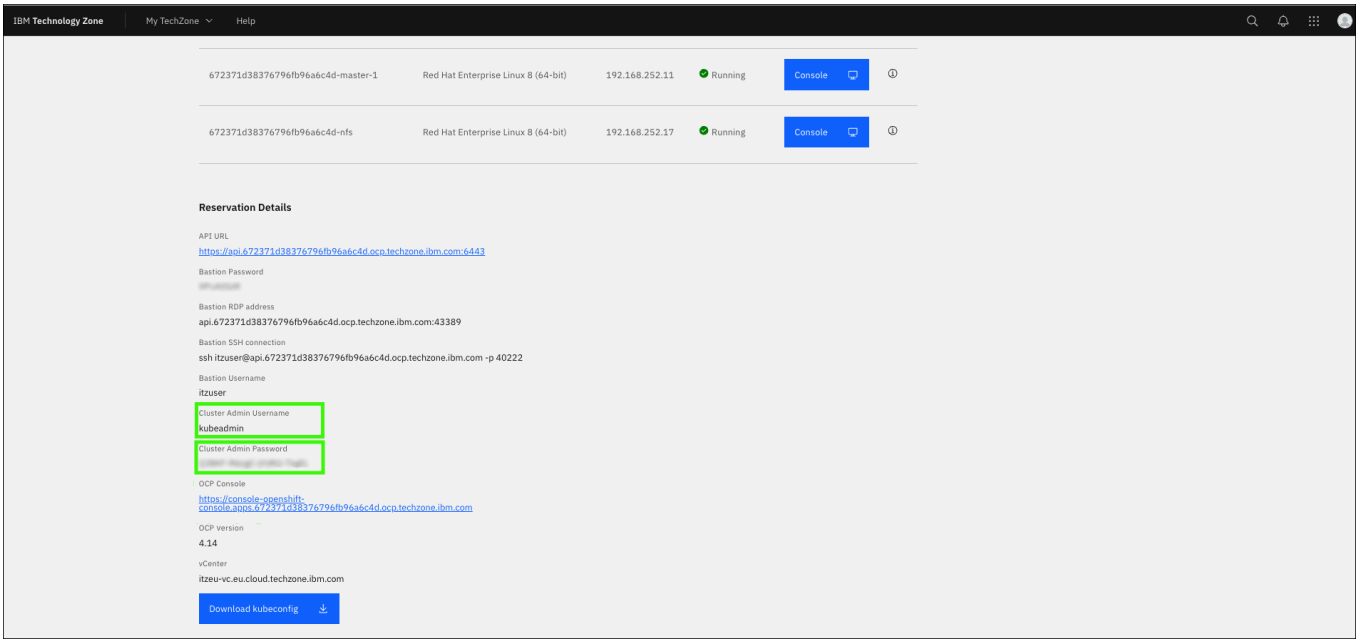
1. Click the following link to open a browser window to your ITZ reservations.

   ITZ My reservations

2. Click the **Single Node OpenShift** tile.

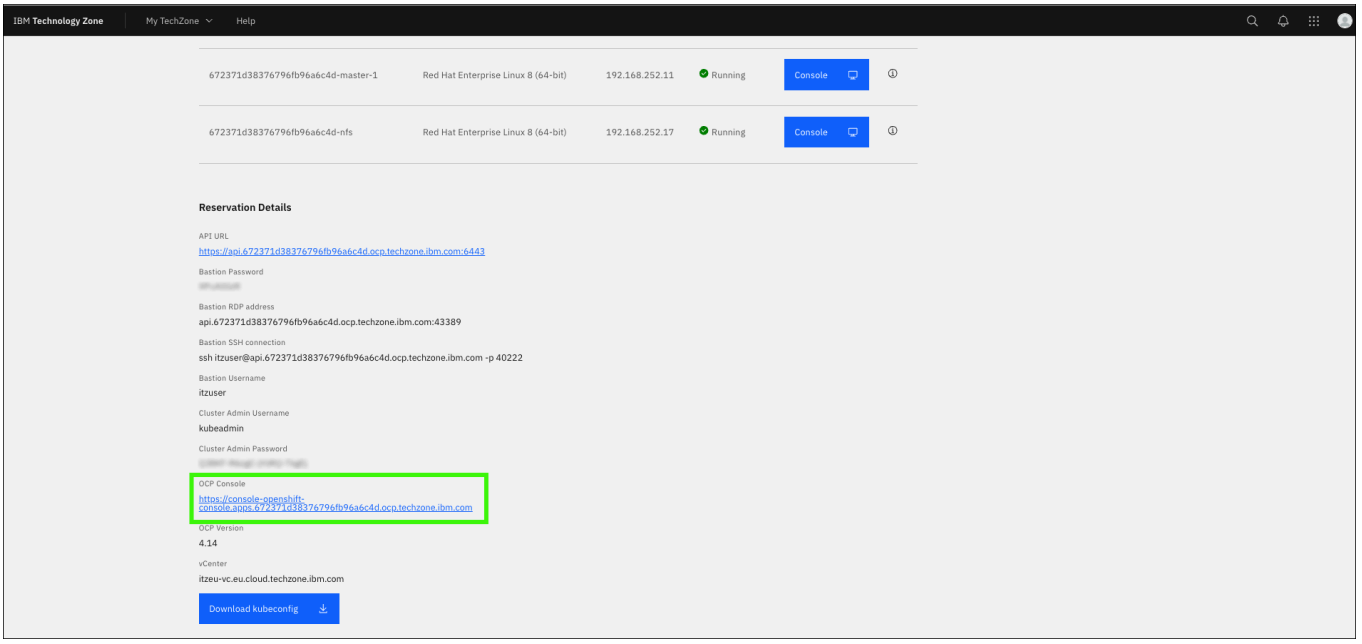


3. Scroll down and record the **Cluster Admin Username** and **Cluster Admin Password**.

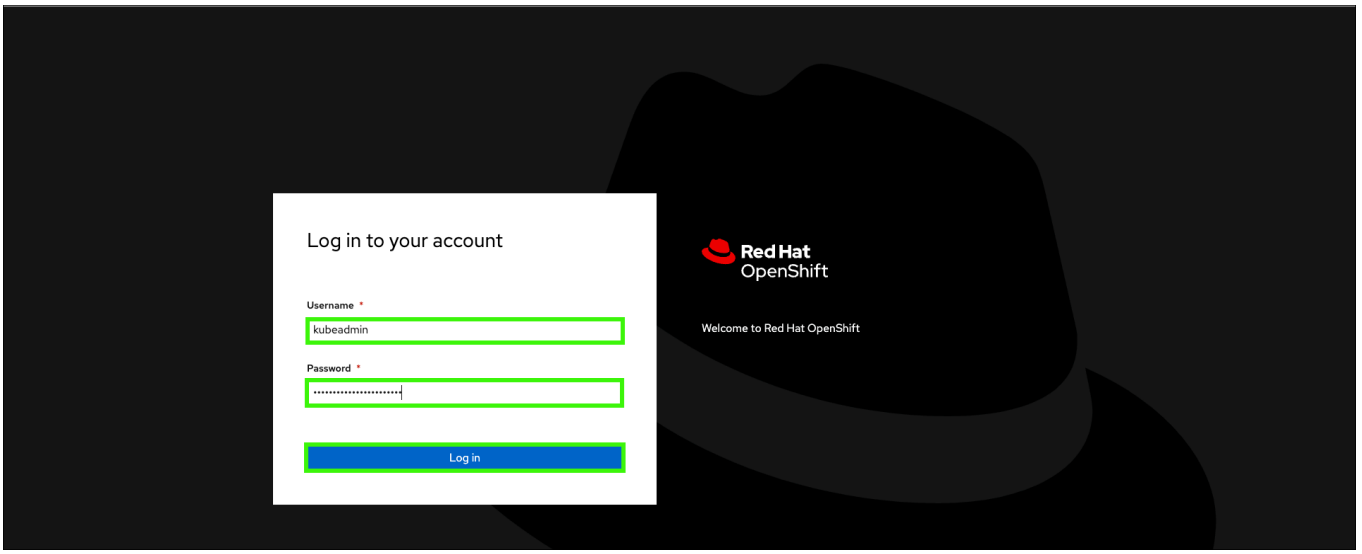


4. Click the **OCP Console** link.

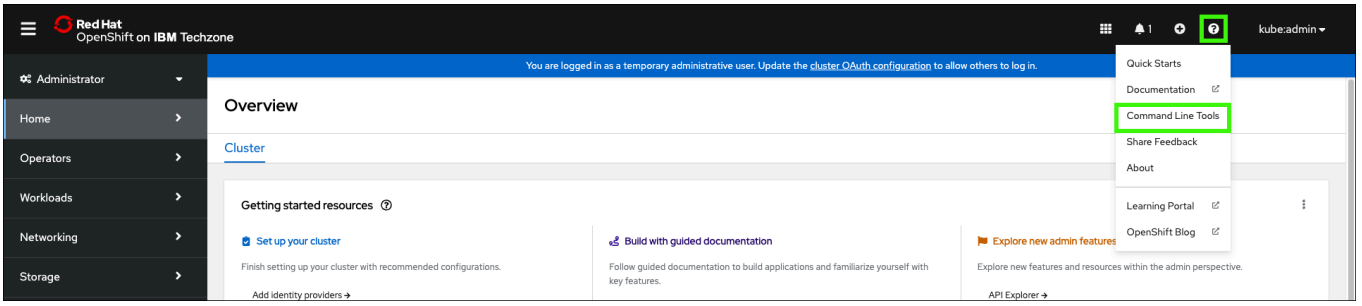
   Note: OCP stands for OpenShift Container Platform.



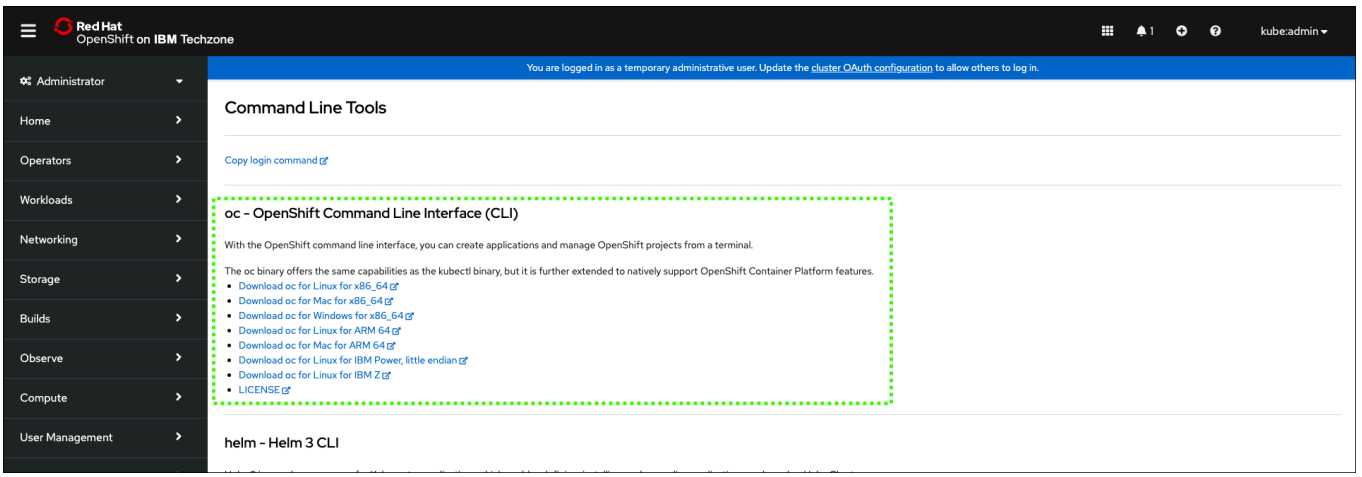
5. Enter the **Cluster Admin Username** and **Cluster Admin Password** values from step 3 and click **Log in**.



6. Click **Help** (?) and then click **Command Line Tools**.



7. Click the link under **oc - OpenShift Command Line Interface (CLI)** for the operating system of your local machine.



Clicking the preceding link automatically downloads either a **.zip** or **.tar** file specific to your operating system. Unzip or untar the file. Place the **oc** binary for your operating system (OS) in a directory that is in your default PATH, or set the PATH environment variable to include the location of the **oc** binary.

8. Verify the installation by running the **oc** command on your local workstation.

```
oc --help
```

Sample output:

```
andrewjones@Andrews-MBP ~ % oc --help
OpenShift Client

This client helps you develop, build, deploy, and run your applications on any
OpenShift or Kubernetes cluster. It also includes the administrative
commands for managing a cluster under the 'adm' subcommand.

Basic Commands:
  login             Log in to a server
  new-project       Request a new project
  new-app           Create a new application
  status            Show an overview of the current project
  project           Switch to another project
  projects          Display existing projects
  explain           Get documentation for a resource

Build and Deploy Commands:
  rollout           Manage a Kubernetes deployment or OpenShift deployment
config
  rollback          Revert part of an application back to a previous deployment
```

Mac/OS users may need to adjust security settings.

The **oc** binary may cause a security exception. Adjust the security settings by opening the **System Settings** utility and clicking **Privacy & Security**. Under **Security** locate the message about the **oc** binary and click **Allow Anyway**. Return to the terminal window and try the `oc --help` command again and click **Allow Anyway** when prompted.

Prepare to ingest documents

Before ingesting documents, complete the following setup steps.

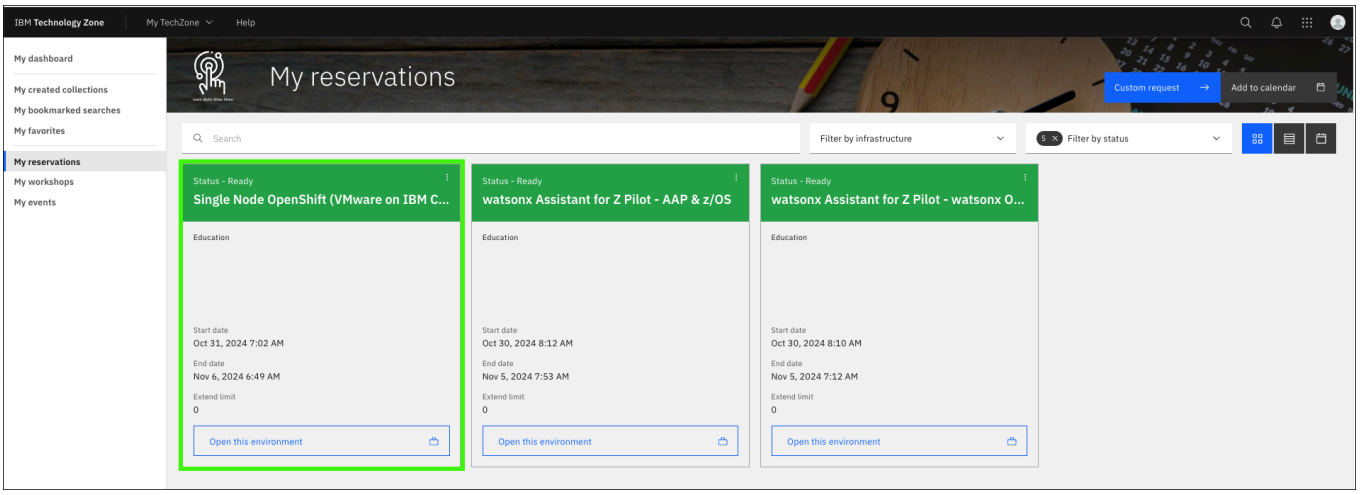
Log in to the OpenShift cluster from your local terminal

Note: If you just installed the **oc** utility, skip the next 5 steps.

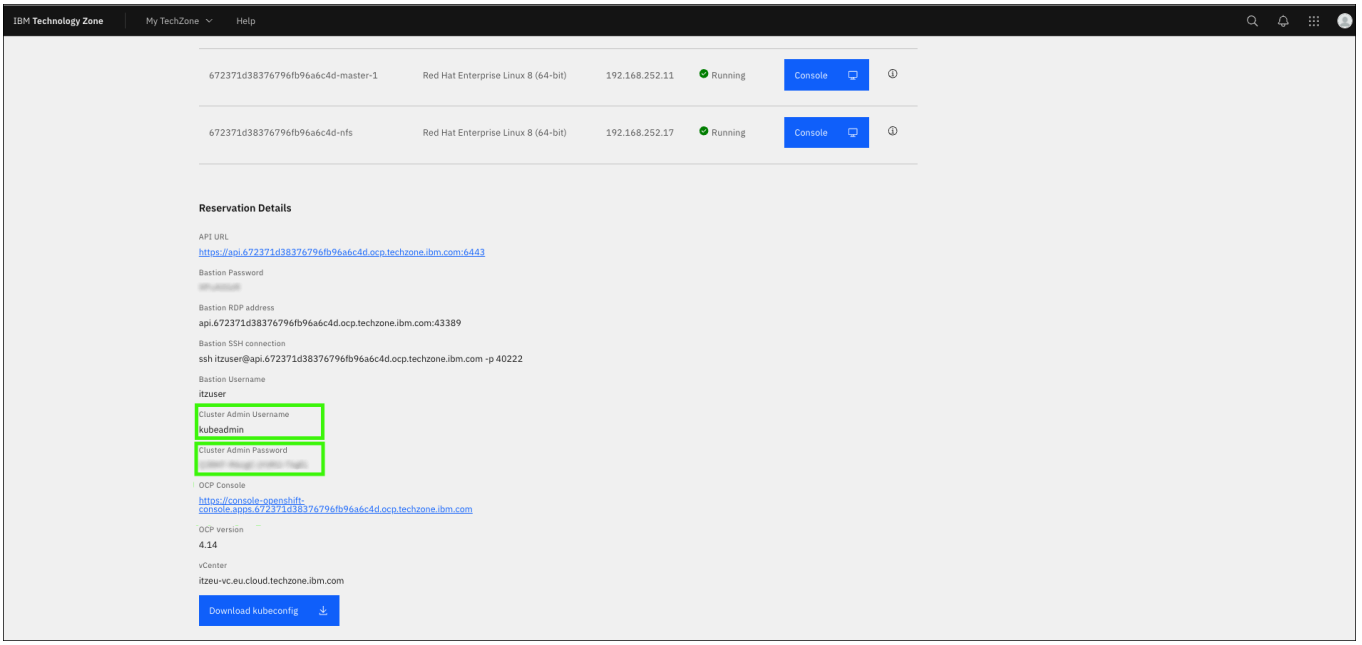
1. Click the following link to open a browser window to your ITZ reservations.

   ITZ My reservations

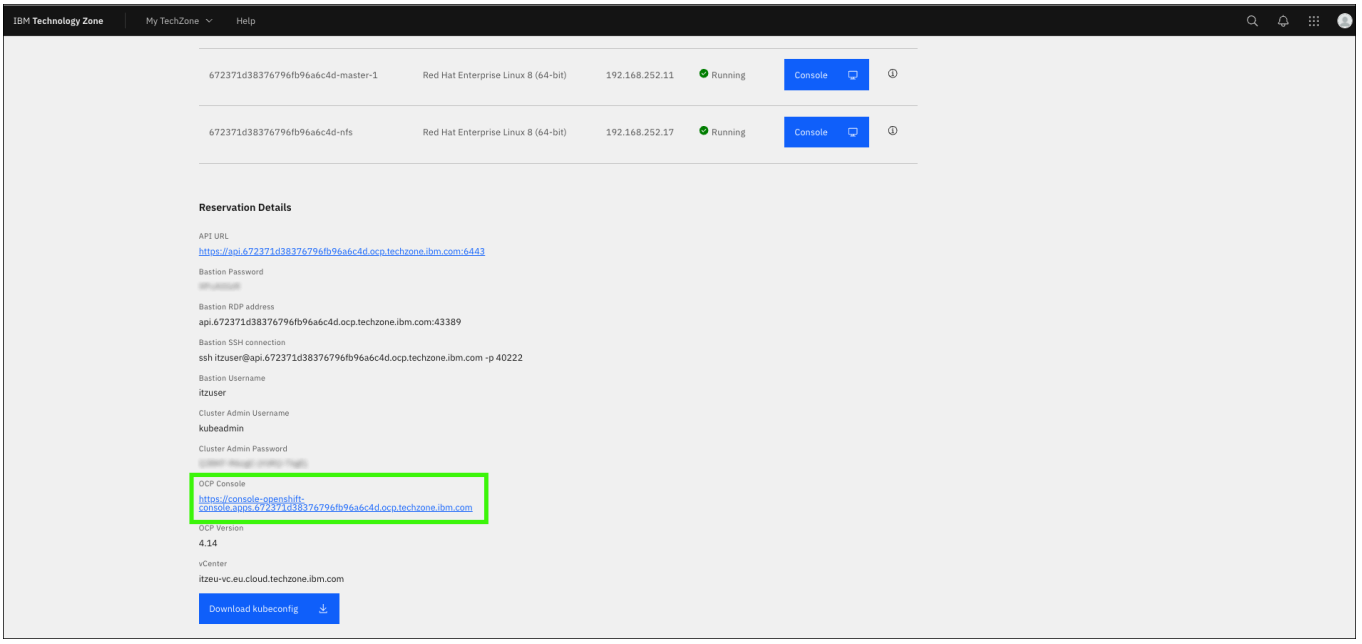
2. Click the **Single Node OpenShift** tile.



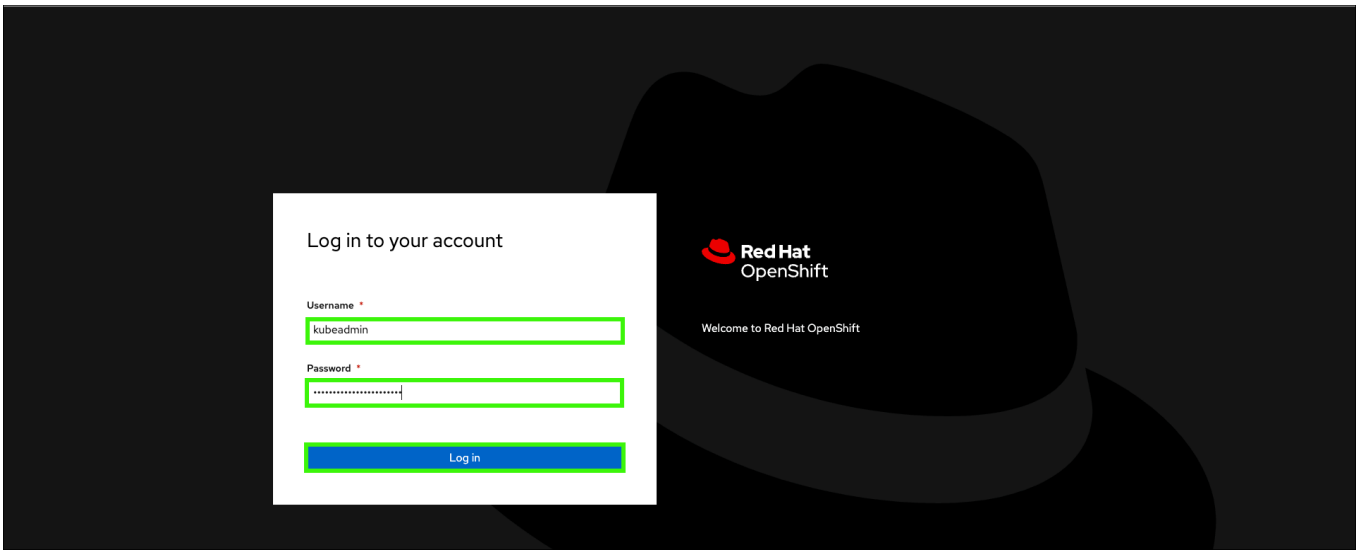
3. Scroll to the bottom of the reservation page and record the **Cluster Admin Username** and **Cluster Admin Password**.



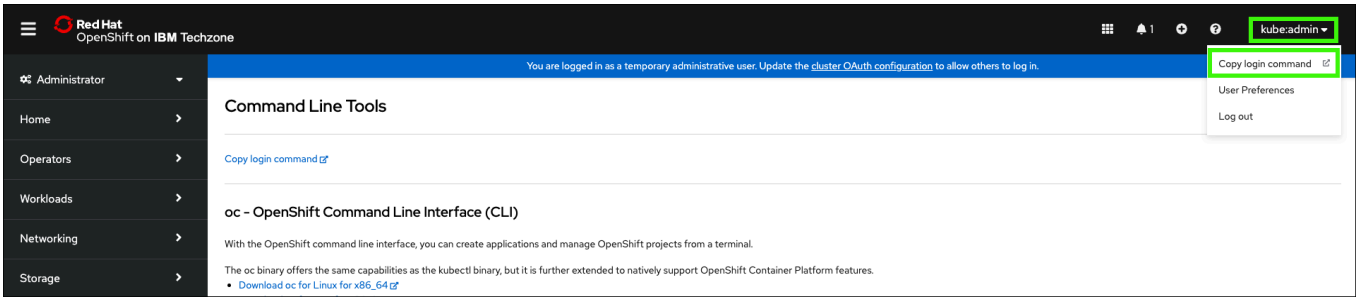
4. Click the **OCP Console** link.



5. Enter the **Cluster Admin Username** and **Cluster Admin Password** values from step 3 and click **Log in**.



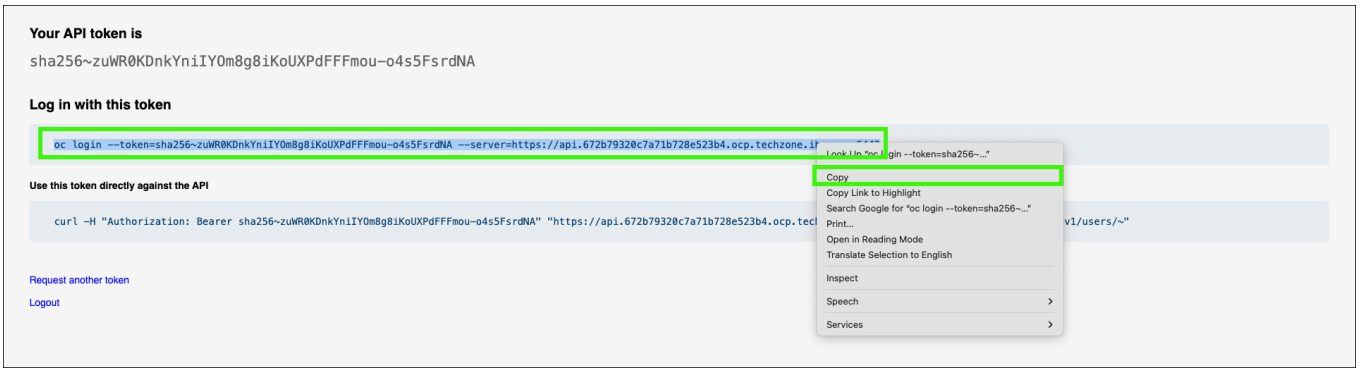
6. Click the **kube:admin** profile drop-down and click **Copy login command**.



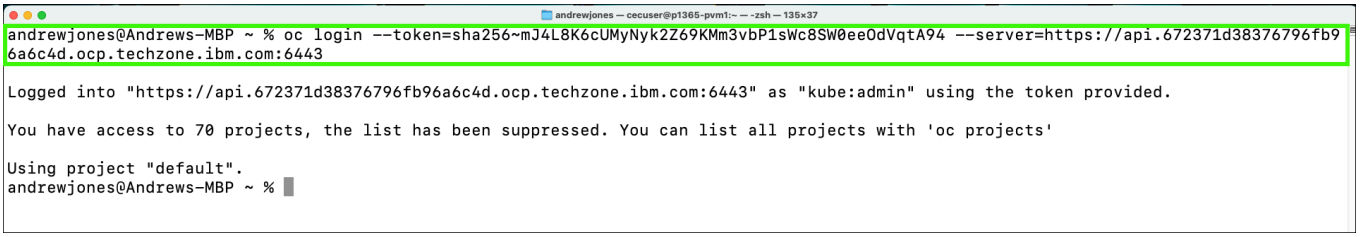
7. Click **Display Token**.

8. Select and copy the **Log in with this token** string.

For most operating systems, double-click the value, then right-click and select **Copy**.



9. Open a command prompt or terminal window on your local workstation.
10. Paste the login command and press **enter**.



Create a working directory

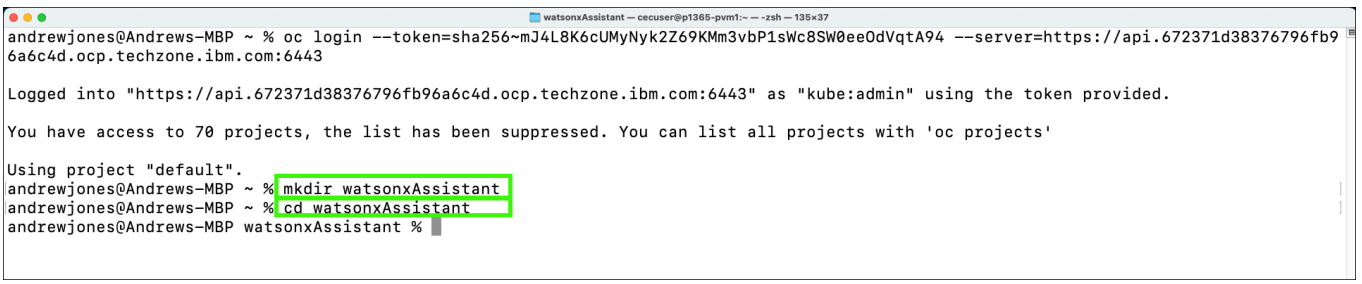
1. Create a directory to store the configuration files that you will create in the next steps.

> **Instructions vary by your local workstation's operating system.**
>
> The directions that follow may vary depending on your operating system. The examples provided are based upon MacOS.

```
mkdir watsonxAssistant
```

2. Change to the new directory.

```
cd watsonxAssistant
```



Install IBM Certificate Manager on Red Hat OpenShift

1. In a text editor, create a file named `catalogCertManager.yaml` and paste the following text in the file.

> **Formatting of the yaml file is critical!**
>
> The content of the YAML file must be formatted exactly as shown. Use the **Copy** icon to prevent typographical errors.

File name:

```
catalogCertManager.yaml
```

File contents:

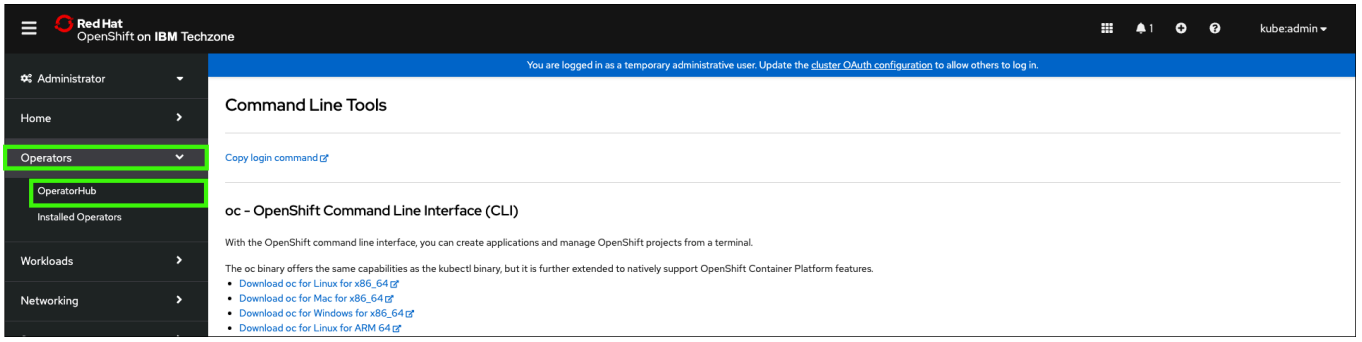
```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-cert-manager-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-cert-manager-4.2.7
  grpcPodConfig:
    securityContextConfig: restricted
  image: icr.io/cpopen/ibm-cert-manager-operator-
catalog@sha256:4dcf4ace4b5f166f83b31063f7e6404dbf78d8e98a9d4fcf52fedf576a55ca6c
  publisher: IBM
  sourceType: grpc
  updateStrategy:
    registryPoll:
      interval: 30m0s
```

2. Install the IBM Certificate Manager operator in the Red Hat OpenShift cluster.

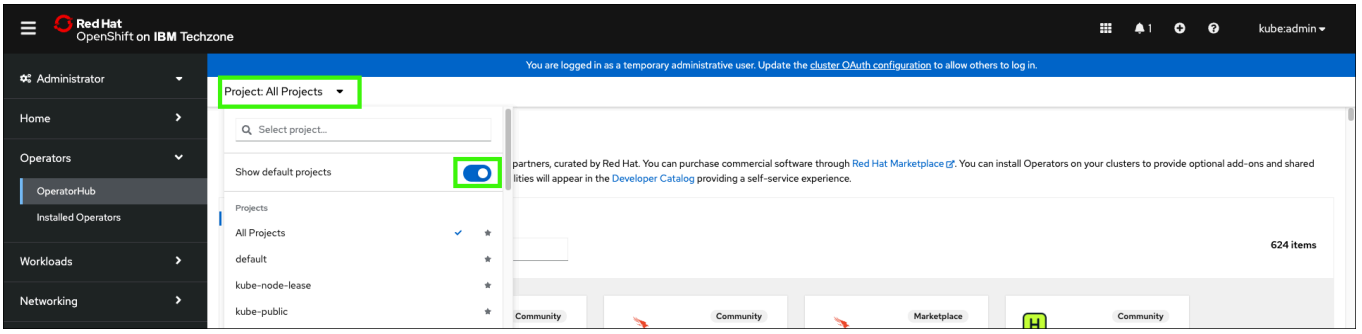
```
oc apply -f catalogCertManager.yaml
```

The preceding command returns a message that states the **ibm-cert-manager-catalog** was created.

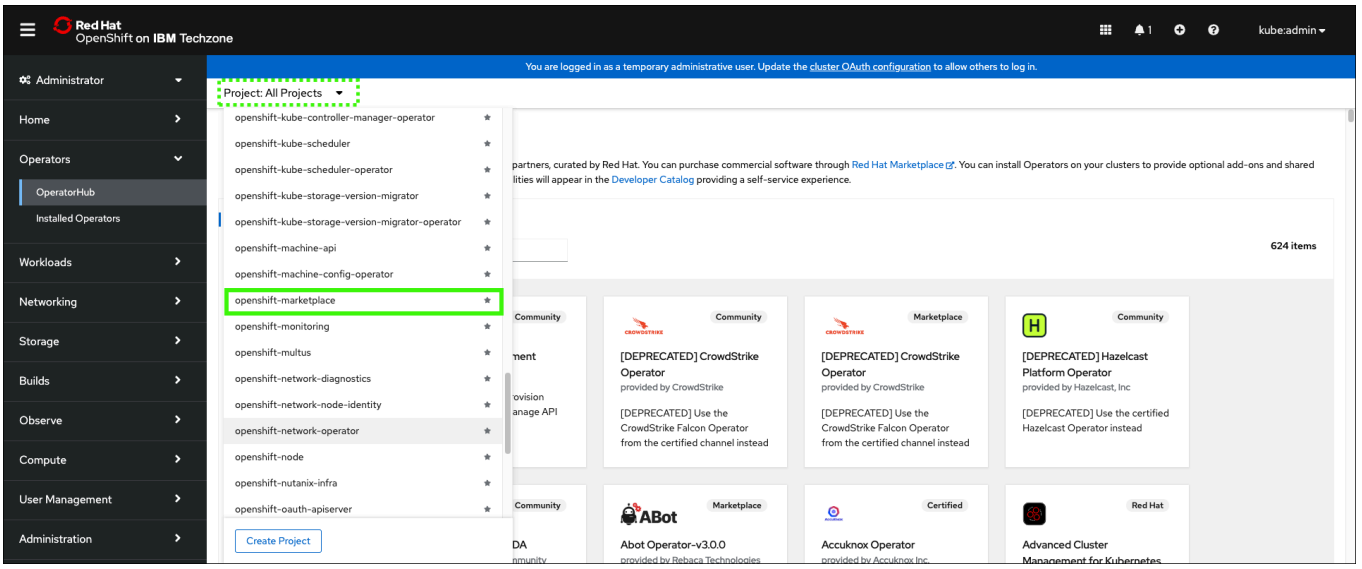
3. In the OpenShift web console, click **Operators** and then select **OperatorHub**.



4. Click the **Project** to pull-down menu and click the **Show default projects** toggle.



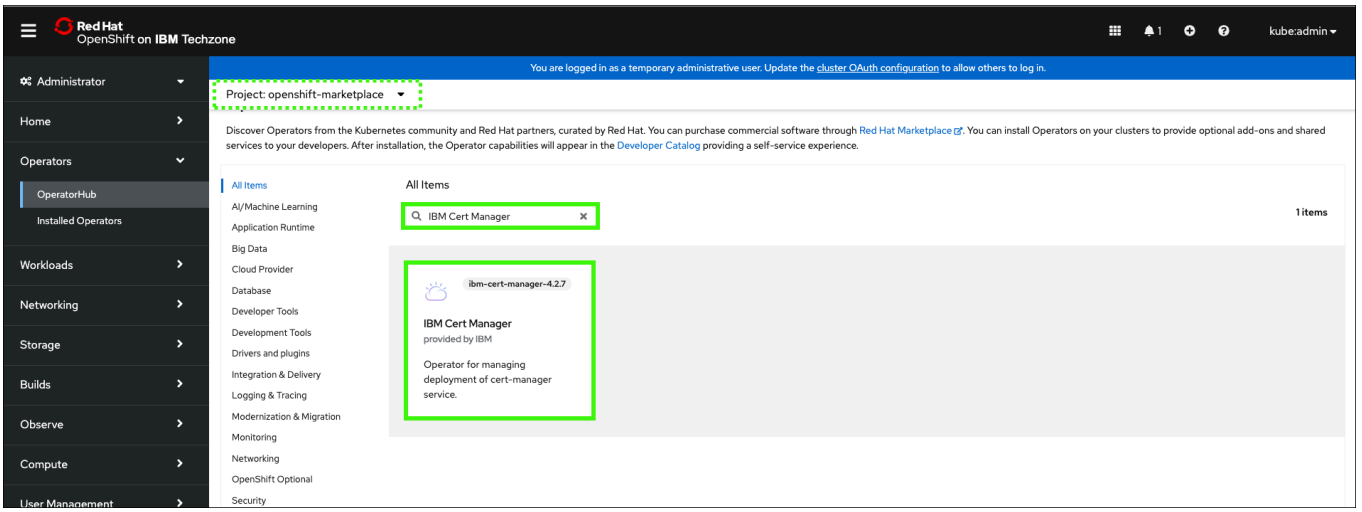
5. Scroll down and select **openshift-marketplace**.



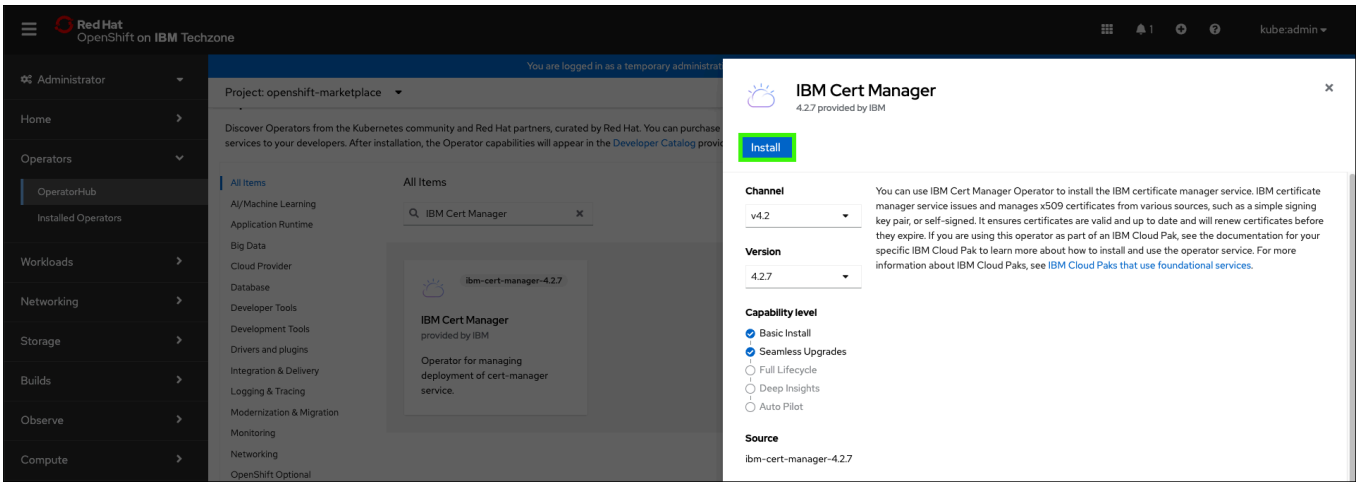
6. Enter **IBM Cert Manager** in the search field and then click the **IBM Cert Manager** tile.

> **Be patient.**
>
> It may take a minute or two for the **IBM Cert Manager** tile to appear.

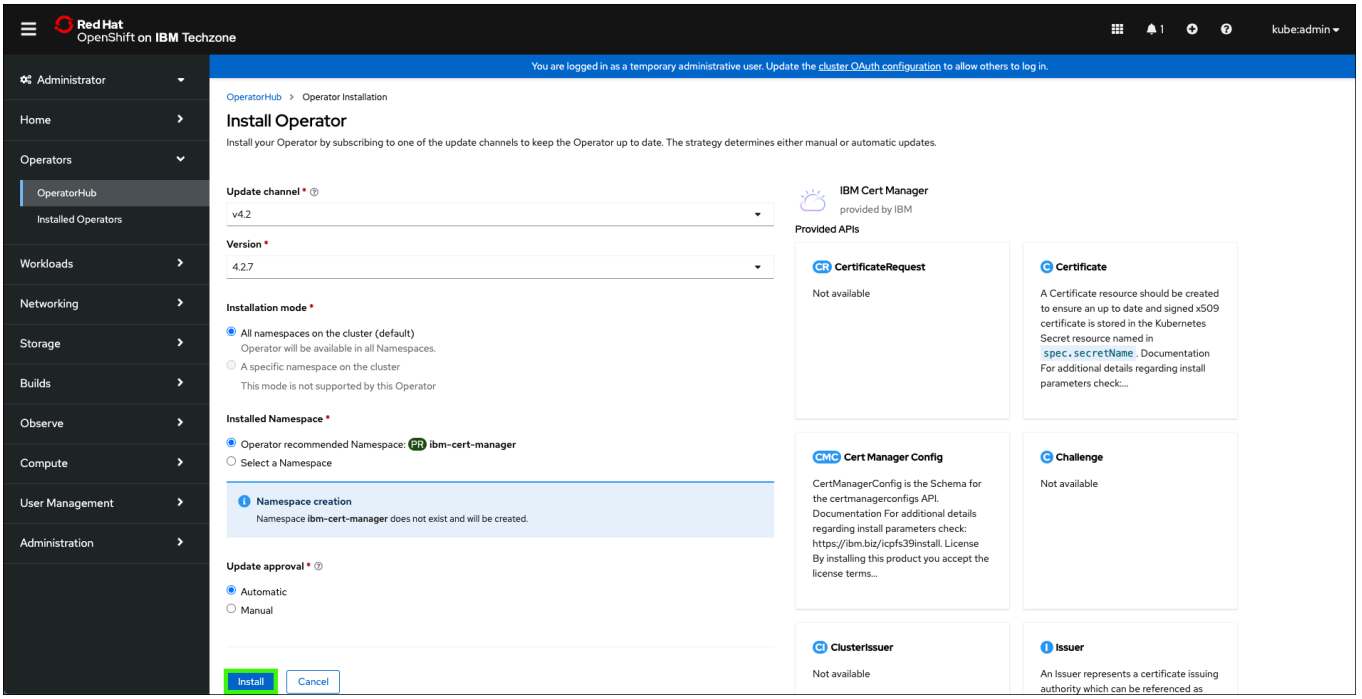
Note: The current version of the operator may differ than the one shown in the image below. Select the most current version.



7. Click **Install**.

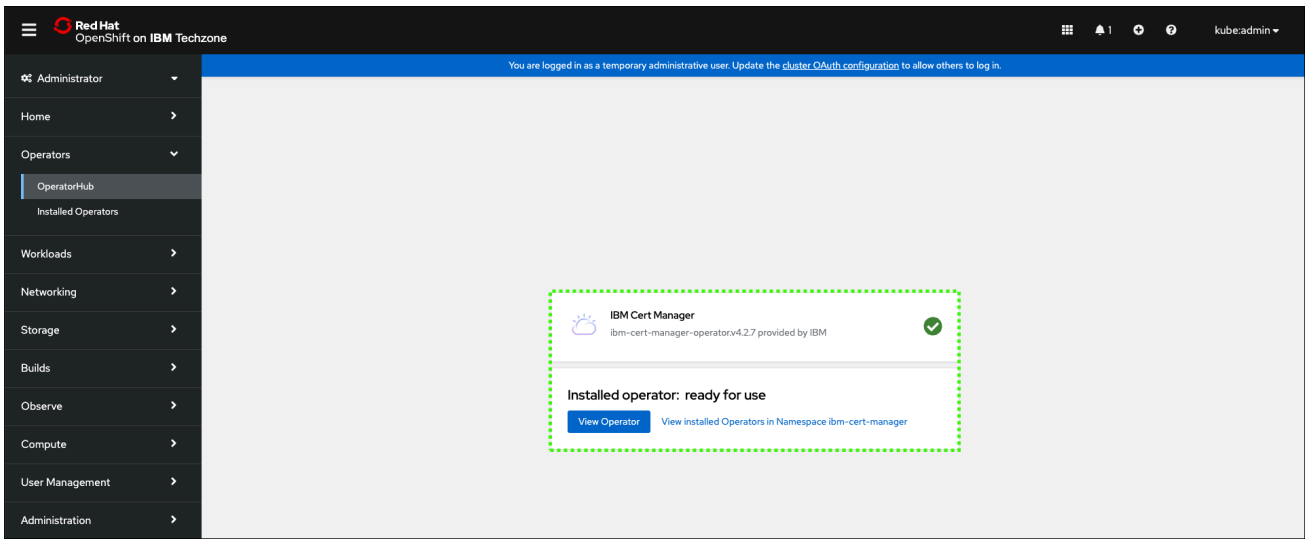


8. Keep the default settings and click **Install**.



Do not continue until...

The installation process takes a few minutes. Do not continue until you see the following message: **Installed operator: ready for use**.



Install the watsonx Assistant for Z Operator (for OpenSearch)

1. In your command prompt or terminal window, create a new namespace called `wxa4z-byos` in the Red Hat OpenShift cluster.

   ```
   oc create namespace wxa4z-byos
   ```

2. Create or obtain your IBM Container Software production entitlement key.

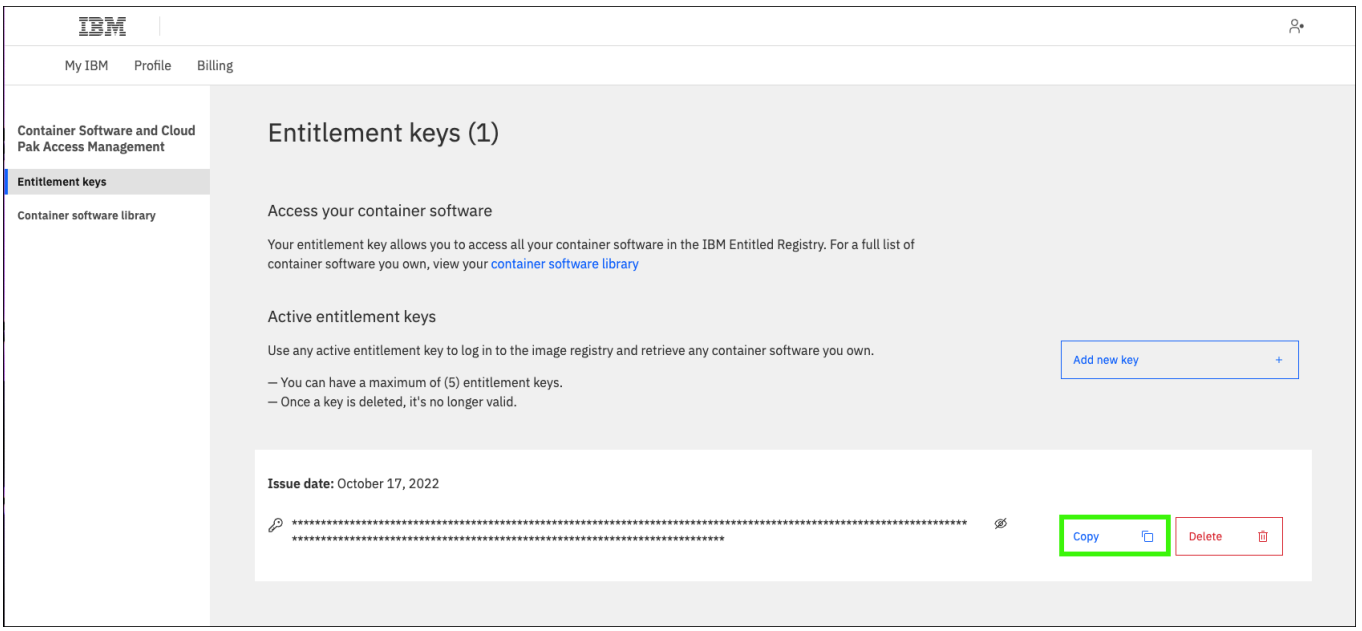
   A production entitlement key is required to pull the container images that get deployed by the operator.

   To create or retrieve your existing entitlement key, follow the instructions here.

   If additional assistance is needed, refer to this site.

   After locating your existing key or creating a new key, continue to the next step.

3. Click **copy** and record your entitlement key for future use in a secure location.



4. In your command prompt or terminal window, set an environment variable with your production entitlement key.

   Substitute your production entitlement key copied in the last step for `<entitlement key>`.

```
export IBM_CS_ENT_KEY=<entitlement key>
```

5. Enter the following command to create a pull secret for the **Container Registry**.

```
oc -n wxa4z-byos create secret docker-registry icr-pull-secret --docker-server=cp.icr.io --
docker-username=cp --docker-password=$IBM_CS_ENT_KEY
```

6. In a text editor, create a file named `catalogSource.yaml` and paste the following text in the file.

> **Formatting of the yaml file is critical!**
>
> The content of YAML files must be formatted exactly as shown. Use the copy icon to prevent typographical errors.

File name:

```
catalogSource.yaml
```

File contents:

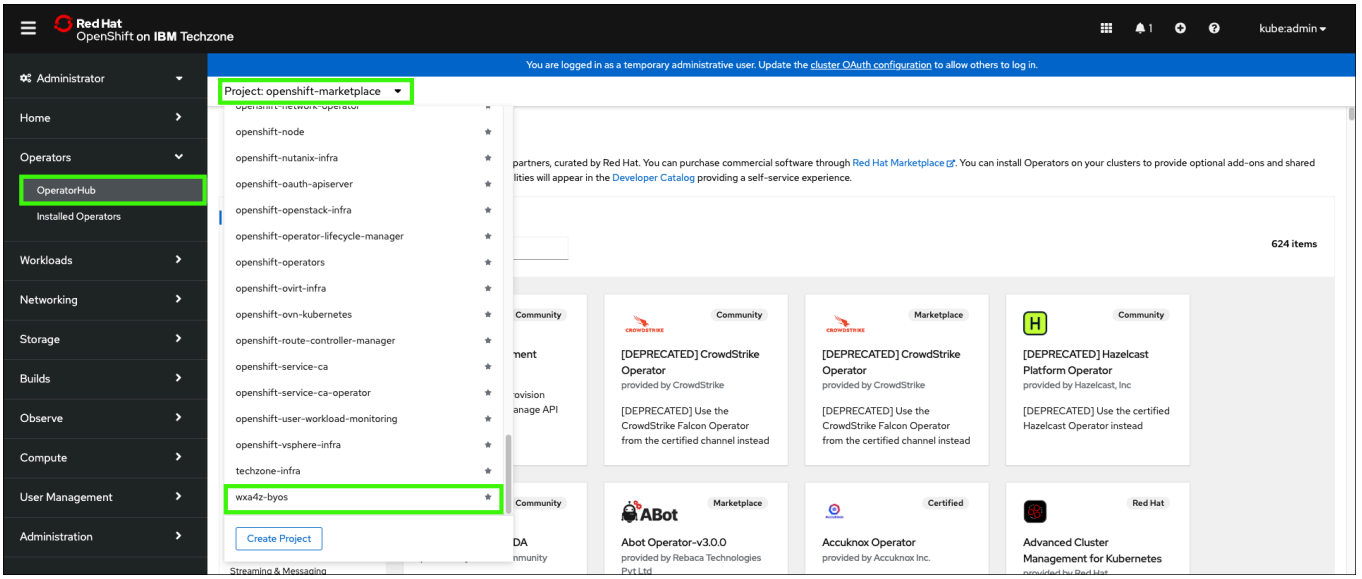
```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-wxa4z-operator-catalog
  namespace: wxa4z-byos
spec:
  displayName: "IBM watsonx Assistant for Z Operator Catalog"
  image: icr.io/cpopen/ibm-wxa4z-
catalog:v2.1.0@sha256:a085d360b6aa0e40cf86a632eb5cd190a0407d1c54ec1b2d1d2fb5507f39a524
  publisher: 'IBM'
  sourceType: grpc
  secrets:
  - icr-pull-secret
```

7. Create your document catalog in the Red Hat OpenShift operator.

```
oc apply -f catalogSource.yaml
```

```
watsonxAssistant — cecuser@p1365-pvm1:~ — -zsh — 135×37
andrewjones@Andrews-MBP watsonxAssistant % oc create namespace wxa4z-byos
namespace/wxa4z-byos created
andrewjones@Andrews-MBP watsonxAssistant % export IBM_CS_ENT_KEY=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiL
CJpYXQiOjE2NjYwMTk1ODAsImp0aSI6IjNkOWUyMzZjZTAzMDQzMzVhNTJhYTkzMWNmOTcyMDRlIn0.4M3XRD4XzkHMSOkFNJ4uKVcWZ6SnEAOZO3eL_11A2xY
andrewjones@Andrews-MBP watsonxAssistant % oc -n wxa4z-byos create secret docker-registry icr-pull-secret --docker-server=cp.icr.io --d
ocker-username=cp --docker-password=$IBM_CS_ENT_KEY
secret/icr-pull-secret created
andrewjones@Andrews-MBP watsonxAssistant % vi catalogSource.yaml
andrewjones@Andrews-MBP watsonxAssistant % vi catalogSource.yaml
andrewjones@Andrews-MBP watsonxAssistant % oc apply -f catalogSource.yaml
catalogsource.operators.coreos.com/ibm-wxa4z-operator-catalog created
andrewjones@Andrews-MBP watsonxAssistant %
```

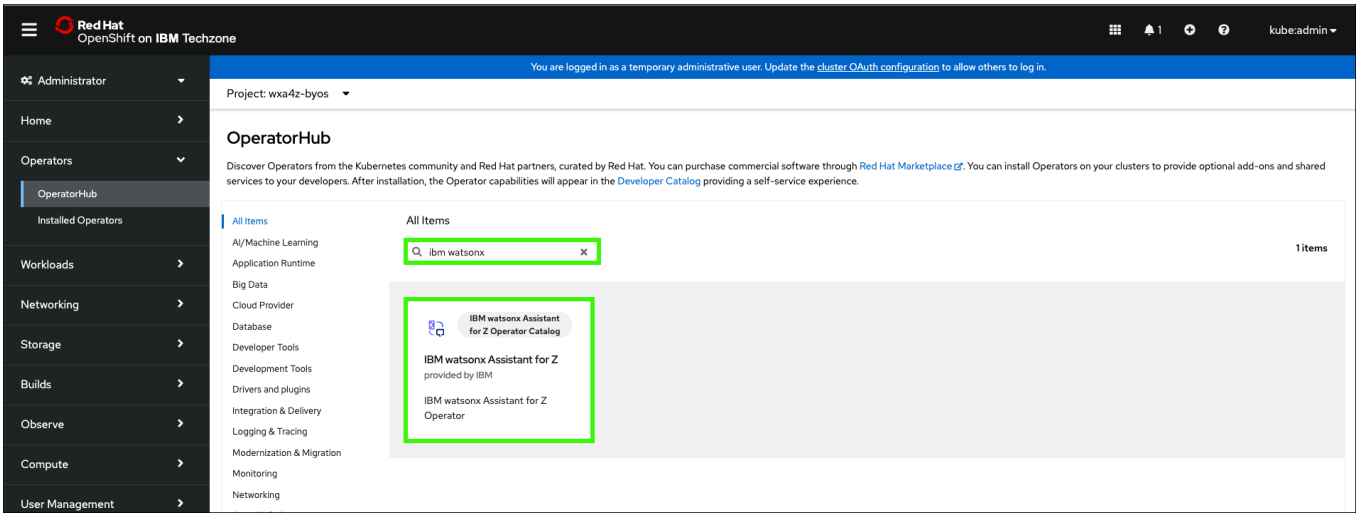
8. In the Red Hat OpenShift web console, click **OperatorHub** and select the **wxa4z-byos** project.



9. Enter **ibm watsonx** in the search field and the click the **IBM watsonx Assistant for Z Operator Catalog** tile.

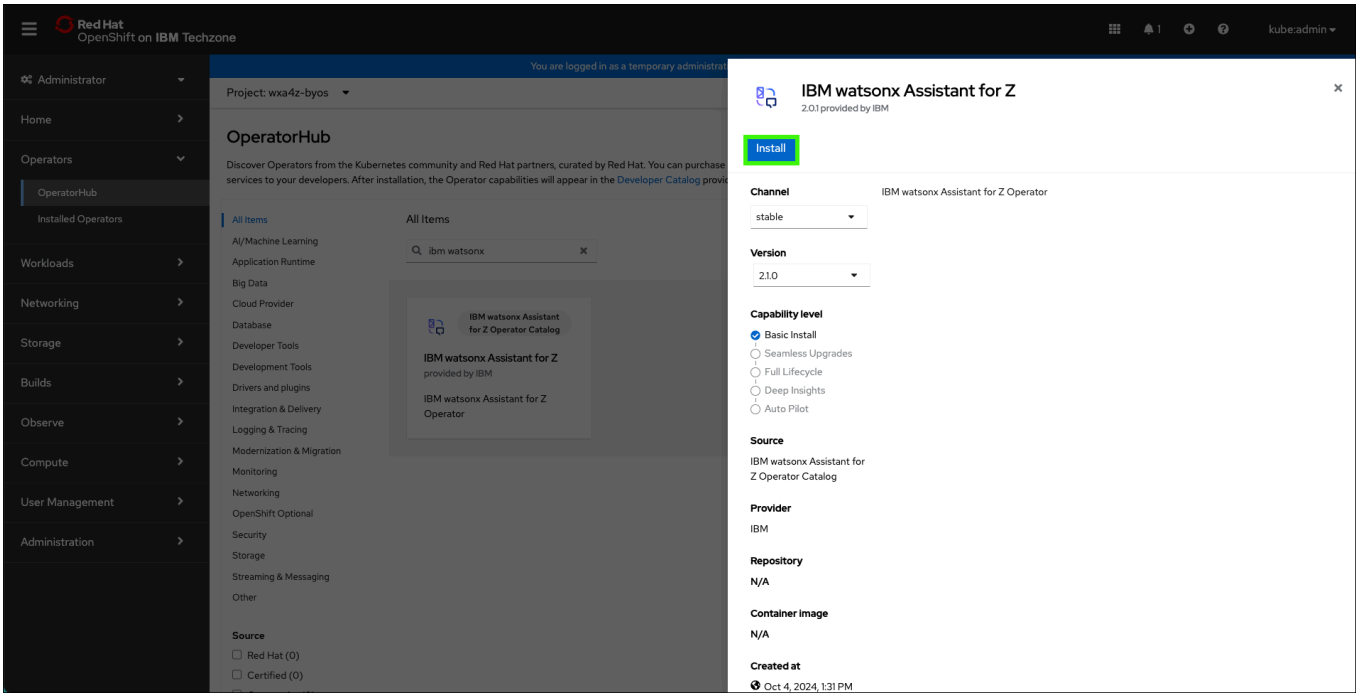
> **Be patient.**
>
> It may take a minute or two for the **IBM watsonx Assistant for Z Operator Catalog** tile to appear.

Note: The current version of the operator may differ than the one shown in the image below.

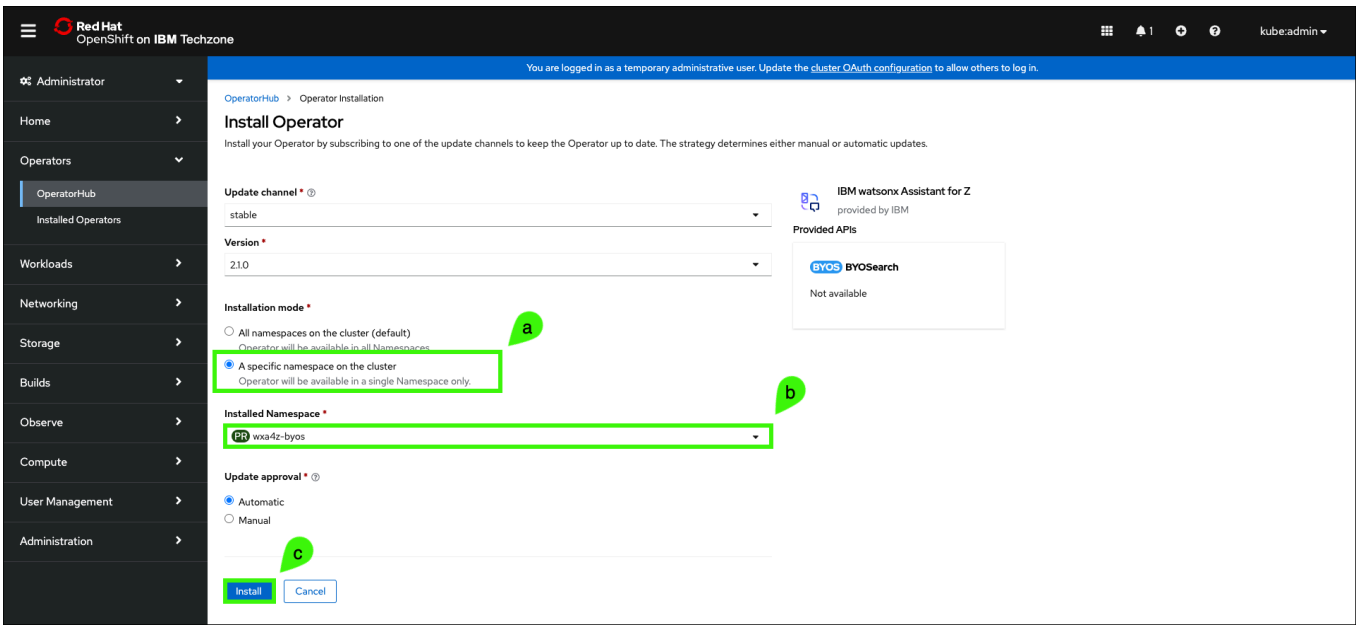


10. Click **Install**.

    Note: The current version of the operator may differ than the one shown in the image below. Select the most current version.

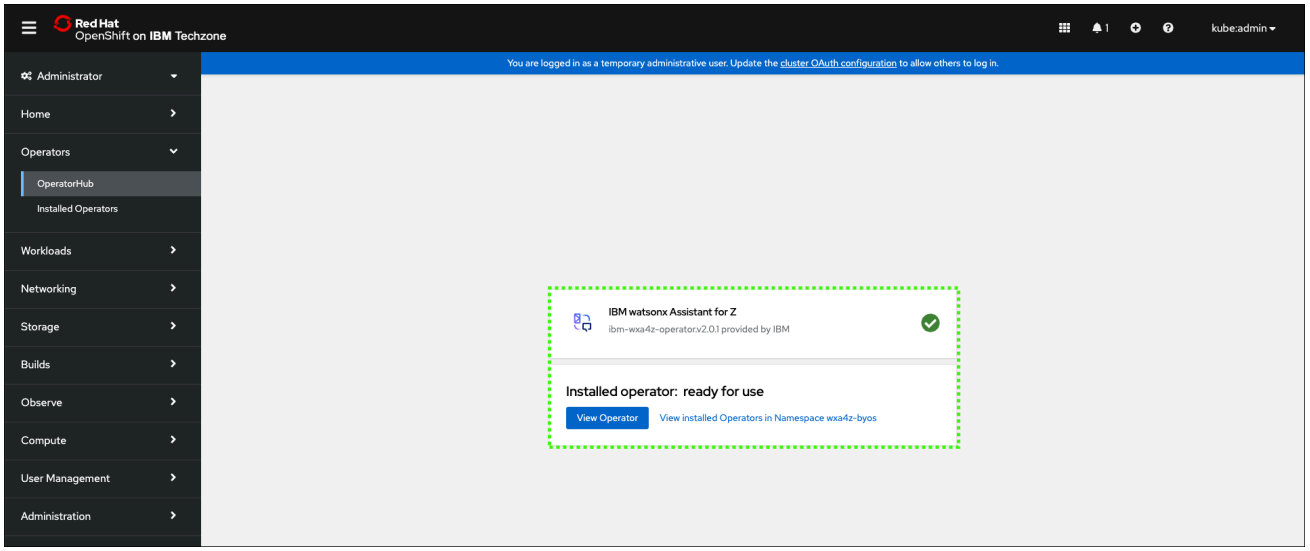


11. Select **A specific namespace on the cluster** (**a**) under **Installation mode** and **wxa4z-byos** (**b**) for the **Installed Namespace**, then click **Install** (**c**).



Do not continue until...

The installation process takes a few minutes. Do not continue until you see the following message: **Installed operator: ready for use**.



12. In your command prompt or terminal window, run the following commands to add the Container Registry credential to the operator's service account.

```
oc project wxa4z-byos
```

For MacOS users:

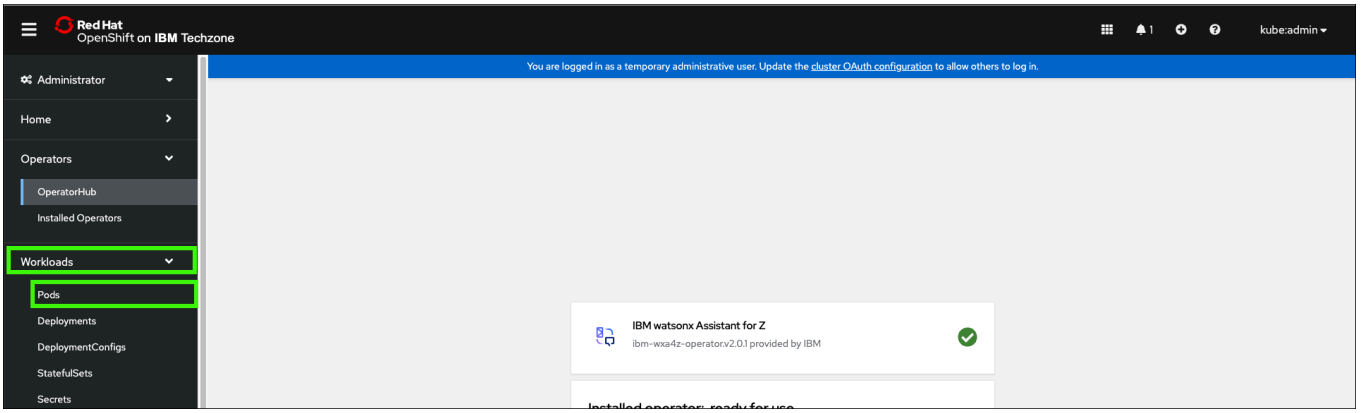
```
oc patch serviceaccount ibm-wxa4z-operator-controller-manager --type merge -p
'{"imagePullSecrets": [{"name": "icr-pull-secret"}]}'
```

For Microsoft Windows users:

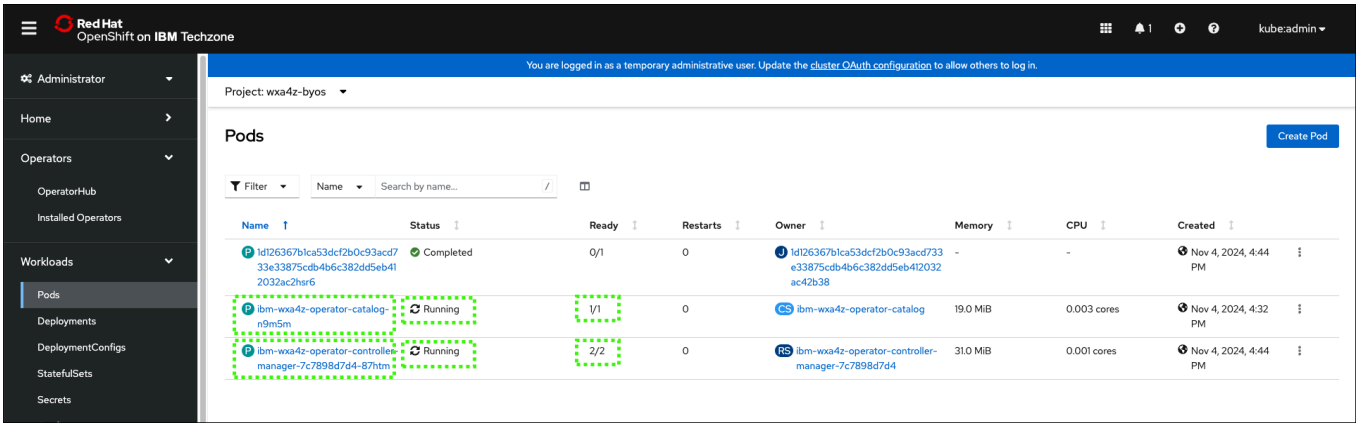
```
oc patch serviceaccount ibm-wxa4z-operator-controller-manager --type merge -p "
{\"imagePullSecrets\":[{\"name\":\"icr-pull-secret\"}]}"
```

```
watsonxAssistant — cecuser@p1365-pvm1:~ — -zsh — 135×37
andrewjones@Andrews-MBP watsonxAssistant % oc create namespace wxa4z-byos
namespace/wxa4z-byos created
andrewjones@Andrews-MBP watsonxAssistant % export IBM_CS_ENT_KEY=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiL
CJpYXQiOjE2NjYwMTk1ODAsImp0aSI6IjNkOWUyMzZjZTAzMDQzMzVhNTJhYTkzMWNmOTcyMDRlIn0.4M3XRD4XzkHMSOkFNJ4uKVcWZ6SnEAOZO3eL_11A2xY
andrewjones@Andrews-MBP watsonxAssistant % oc -n wxa4z-byos create secret docker-registry icr-pull-secret --docker-server=cp.icr.io --d
ocker-username=cp --docker-password=$IBM_CS_ENT_KEY
secret/icr-pull-secret created
andrewjones@Andrews-MBP watsonxAssistant % vi catalogSource.yaml
andrewjones@Andrews-MBP watsonxAssistant % vi catalogSource.yaml
andrewjones@Andrews-MBP watsonxAssistant % oc apply -f catalogSource.yaml
catalogsource.operators.coreos.com/ibm-wxa4z-operator-catalog created
andrewjones@Andrews-MBP watsonxAssistant % oc project wxa4z-byos
Now using project "wxa4z-byos" on server "https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443".
andrewjones@Andrews-MBP watsonxAssistant % oc patch serviceaccount ibm-wxa4z-operator-controller-manager --type merge -p '{"imagePullSe
crets": [{"name": "icr-pull-secret"}]}'
serviceaccount/ibm-wxa4z-operator-controller-manager patched
andrewjones@Andrews-MBP watsonxAssistant %
```

13. In the Red Hat OpenShift web console, under **Workloads**, click **Pods**.

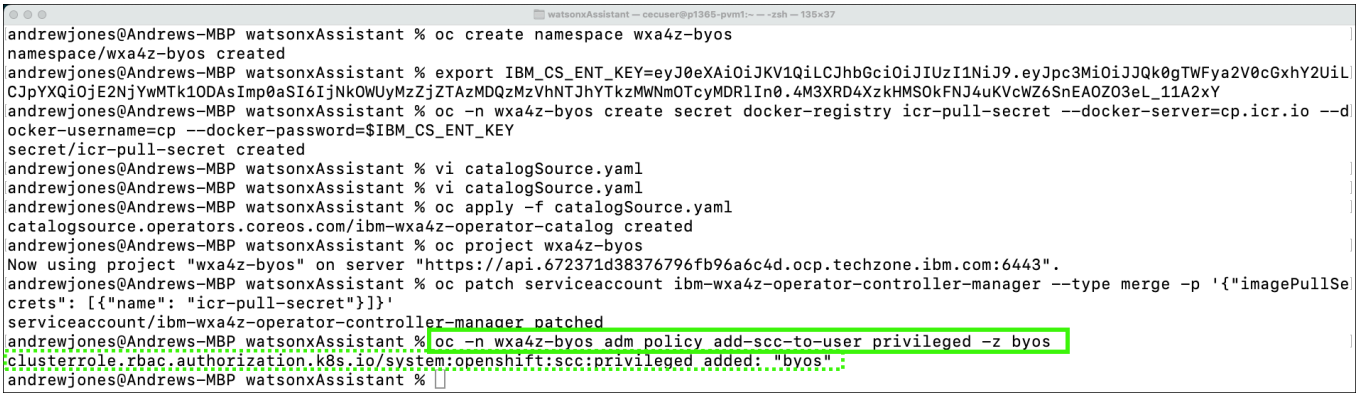


14. Verify the two pods that start with **ibm-wxa4z-operator** have a status of **Running** and that all pods are **Ready**.



15. Run the following command to set the administrative policy for the workspace.

```
oc -n wxa4z-byos adm policy add-scc-to-user privileged -z byos
```



Deploy required secrets and the custom bring-your-own-search (BYOSearch) resources

1. In a text editor, create a file named `os-secret.yaml` and paste the following text in the file.

   File name:

   ```
   os-secret.yaml
   ```

   Substitute a secure password of your choosing for the string `<OPENSEARCH_PASSWORD>`.

   File contents:

   ```
   apiVersion: v1
   stringData:
     password: <OPENSEARCH_PASSWORD>
   kind: Secret
   metadata:
     name: opensearch-creds
     namespace: wxa4z-byos
   type: Opaque
   ```

2. Create the secret by running the following command.

   ```
   oc apply -f os-secret.yaml
   ```

3. In a text editor, create a file named `client-ingestion-secret.yaml` and paste the following text in the file.

   File name:

   ```
   client-ingestion-secret.yaml
   ```

   Substitute a secure authentication key of your choosing for the string `<CLIENT_INGESTION_AUTHKEY>`. The authentication key can be a random password.

   File contents:

```
apiVersion: v1
stringData:
  authkey: <CLIENT_INGESTION_AUTHKEY>
kind: Secret
metadata:
  name: client-ingestion-authkey
  namespace: wxa4z-byos
type: Opaque
```

4. Create the secret by running the following command.

```
oc apply -f client-ingestion-secret.yaml
```

5. In a text editor, create a file named `wrapper-creds.yaml` and paste the following text in the file.

   File name:

   ```
   wrapper-creds.yaml
   ```

   Substitute a secure password credential of your choosing for the string `<WRAPPER_PASSWORD>`. The password can be a random password. Use this password in the following steps when you configure your BYOS connection in your assistant to connect to the network route.

   File contents:

   ```
   apiVersion: v1
   stringData:
     username: admin
     password: <WRAPPER_PASSWORD>
   kind: Secret
   metadata:
     name: wrapper-creds
     namespace: wxa4z-byos
   type: Opaque
   ```

6. Create the secret by running the following command.

```
oc apply -f wrapper-creds.yaml
```

7. Obtain and record your cluster domain that is used for routes by running the following command.

```
oc -n openshift-ingress-operator get ingresscontroller default -o jsonpath="{.status.domain}"
```

> ⚠ **The output from the command does not include a newline.**
>
> The value returned for the cluster domain does not include a newline. When copying the value do not include the character or characters used for your command line prompt. Do not include the your prompt in the next step!

Note: The output of the command will be a string similar to:
apps.672b79320c7a71b728e523b4.ocp.techzone.ibm.com

8. In a text editor, create a file named `byos.yaml` and paste the following text in the file.

   File name:

   ```
   byos.yaml
   ```

   Substitute the domain name recorded in the previous step for the string `<YOUR_CLUSTER_DOMAIN>`.

   File contents:

```yaml
apiVersion: wxa4z.watsonx.ibm.com/v1
kind: BYOSearch
metadata:
  name: byosearch
  namespace: wxa4z-byos
spec:
  imagePullSecrets:
  - name : icr-pull-secret
  namespace: wxa4z-byos
  clusterName: wxa4z-byos-cluster
  clusterDomain: <YOUR_CLUSTER_DOMAIN>

  opensearch:
    secretName: opensearch-creds

    persistence:
      enabled: true
      storageClass: "managed-nfs-storage"
      accessModes:
        - ReadWriteOnce
      size: 24Gi

  wrapper:
    createRoute: true
    resources:
      requests:
        cpu: 2
        memory: "500Mi"
    limits:
        cpu: 2
        memory: "1Gi"

  clientIngestion:
    secretName: client-ingestion-authkey

    resources:
      limits:
        cpu: "500m"
        memory: 2Gi
        nvidia.com/gpu: "0"
      requests:
        cpu: "500m"
        memory: 1Gi
        nvidia.com/gpu: "0"
    pvc:
      storageClass: "managed-nfs-storage"
      enabled: true
      size: 24Gi
```

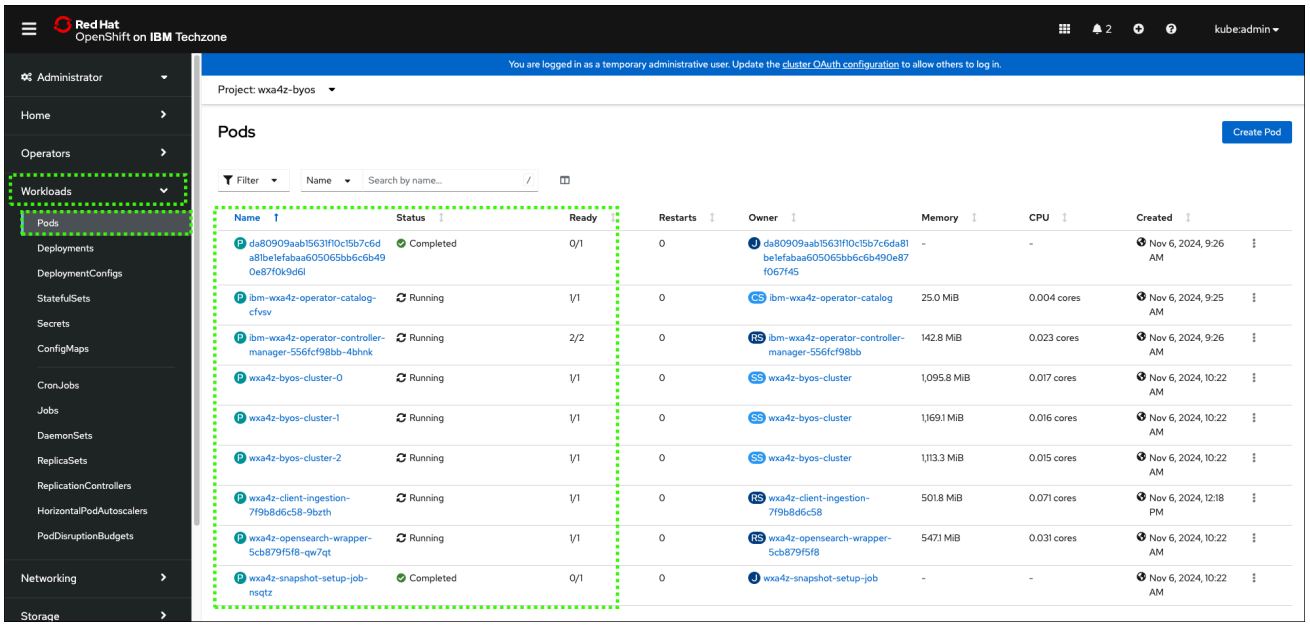
9. Run the following command to deploy BYOS on your cluster.

```
oc apply -f byos.yaml
```

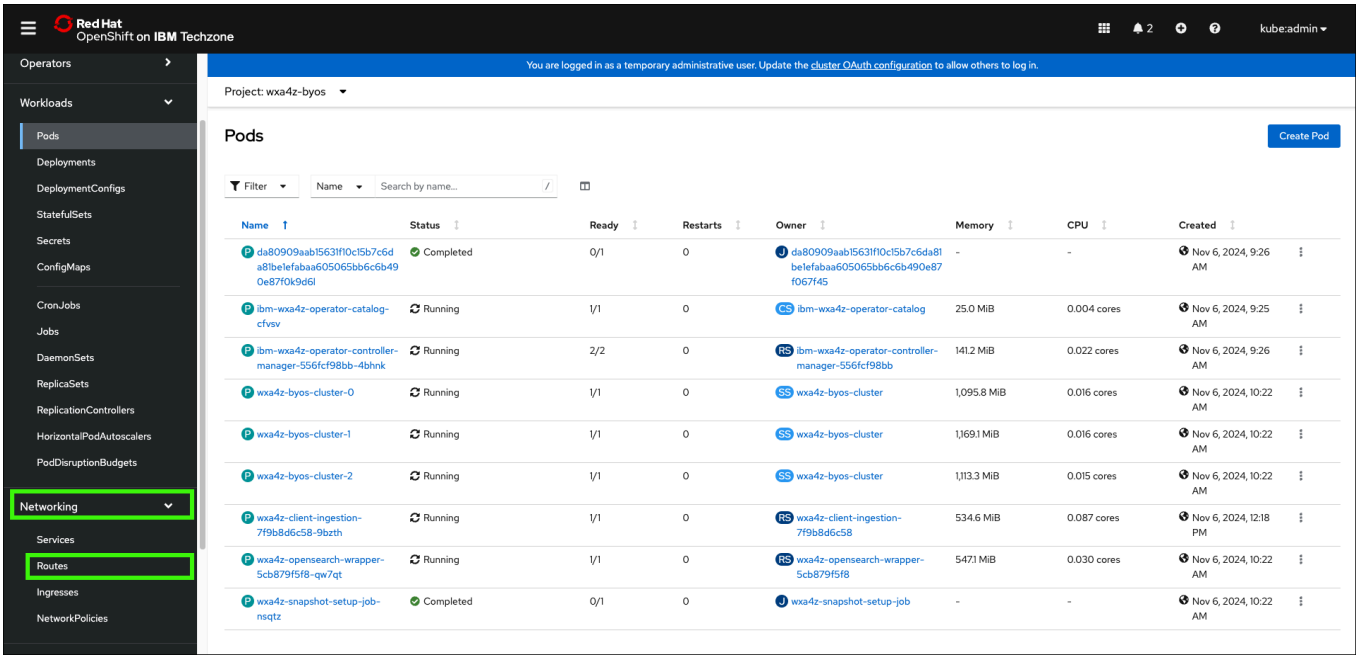
Verify all the required pods are running and get the network route to your BYOS instance

1. In the OCP console, verify that all pods have the status of **Running** or **Completed**.

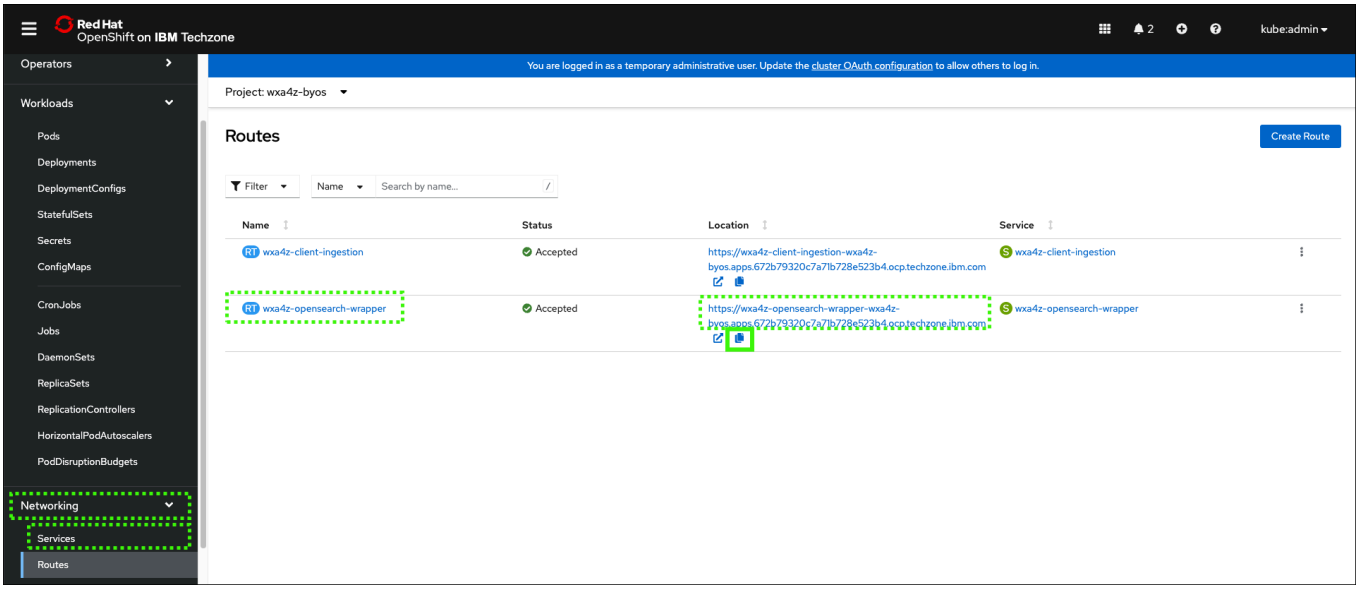
> **Do not continue until...**
>
> The BYOS deployment can take 20 minutes or more to complete. Do not continue until all the pods have a status of “Running” or "Completed". The next step is to retrieve your BYOS endpoint URL.



2. Under **Networking**, click **Routes**.



3. Copy and record the location for the **wxa4z-opensearch-wrapper** route.



Update your assistant with the new BYOS instance route

You are now ready to configure your assistant with the route to your BYOS instance.

1. Using the network route for your BYOS instance, append the string **/v1/query** to complete the URL endpoint.

   The URL should look similar to:

   `https://wxa4z-opensearch-wrapper-wxa4z-byos.apps.672b79320c7a71b728e523b4.ocp.techzone.ibm.com/v1/query`

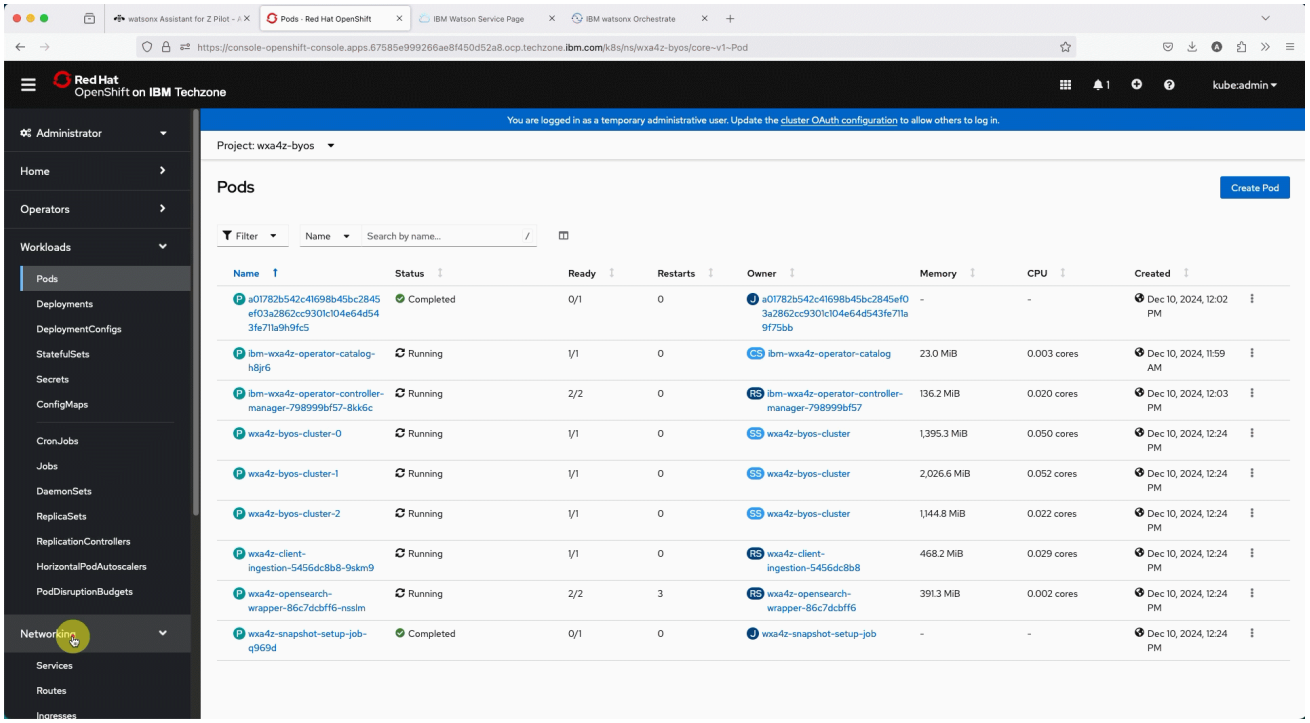
   Important: The above URL will not work for you. Use the value of your specific OpenSearch instance that is recorded in the previous step.

2. Update your assistant's custom search integration URL.

   Next, you need to return to your assistant in the watsonx Orchestrate AI assistant builder and update the custom search integration URL. This time, instead of setting the authentication type to **None**, you need to set it to **Basic authentication**. Use **admin** for the **Username** and the **Password** will be the password you specified in the `wrapper-creds.yaml` file.

   The steps to update the URL are illustrated in the animated gif that follows. You can review the steps to accomplish this here (be sure to use your BYOS URL and not the shared URL specified in the lab guide).

How to set the custom search integration URL.



Troubleshooting

The following are issues that you may encounter. If the provided resolutions do not work, contact support by using the methods that are mentioned in the Support section.

Pods have a status of ErrImagePull or ImagePullBackoff

If the pods starting with **ibm-wxa4z-operator** have a status of “ErrImagePull” or “ImagePullBackoff”, you can delete the pod and it will automatically restart and pull the image successfully. Wait until the pod is re-created successfully.

The wxa4z-client-ingestion pod does not start

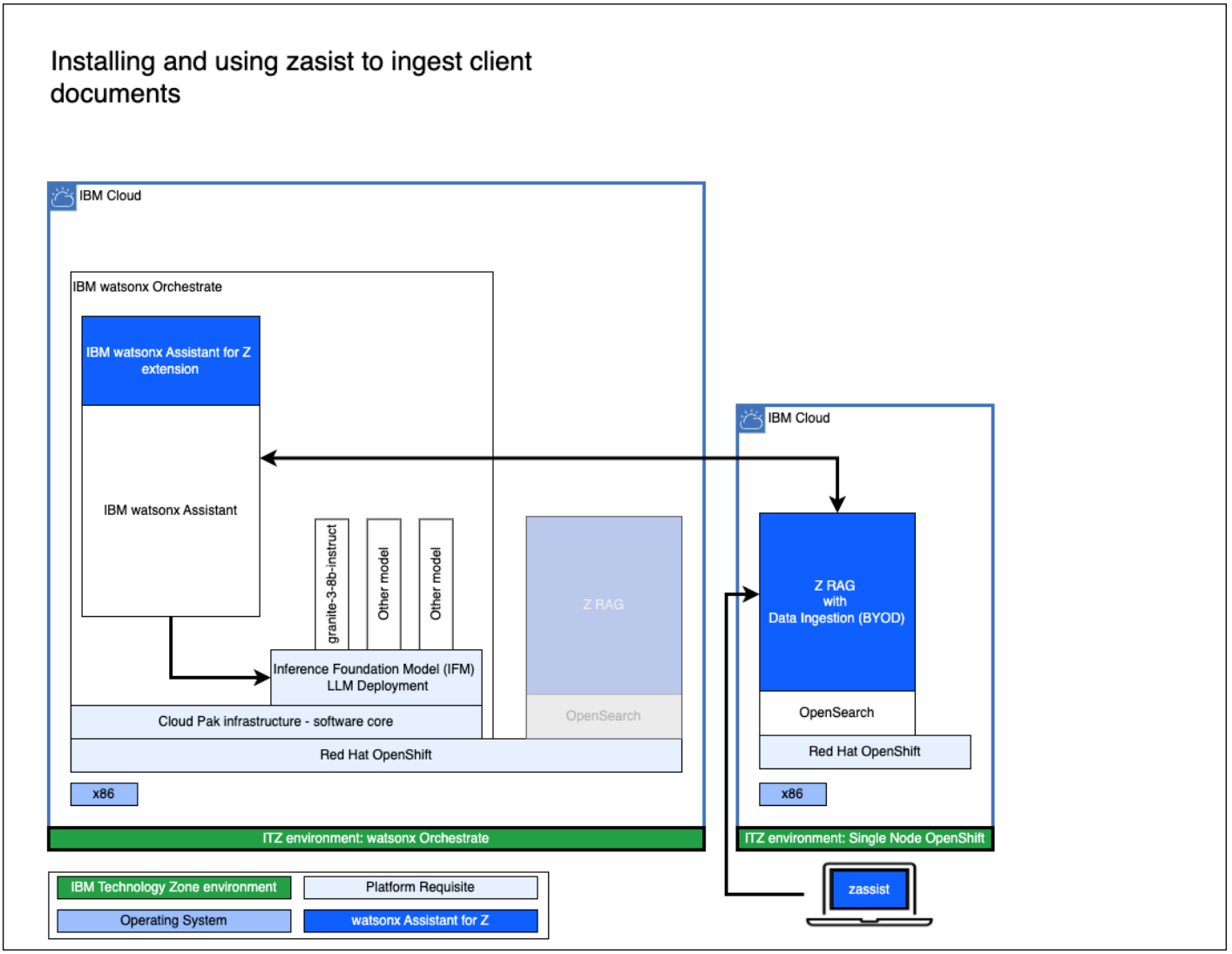
Did you include the **%** character in the **clusterDomain** name when creating the **byos.yaml**? To resolve, edit the **byos.yaml** file and run the following command again. The current pod will be terminated and a new one started. This will take about 20 minutes to start.

```
oc apply -f byos.yaml
```

Installing and using zassist to ingest client documents

With bring-your-own-search (BYOS) installed and configured in your assistant, you can now prepare for document ingestion. Currently, only PDF, HTML, and DOCX file formats are supported for ingestion.

Below is a high-level, logical architecture of the environment you will deploy in this section.



To prepare for document ingestion, you can also reference the setup instructions that are located here.

Install the zassist utility

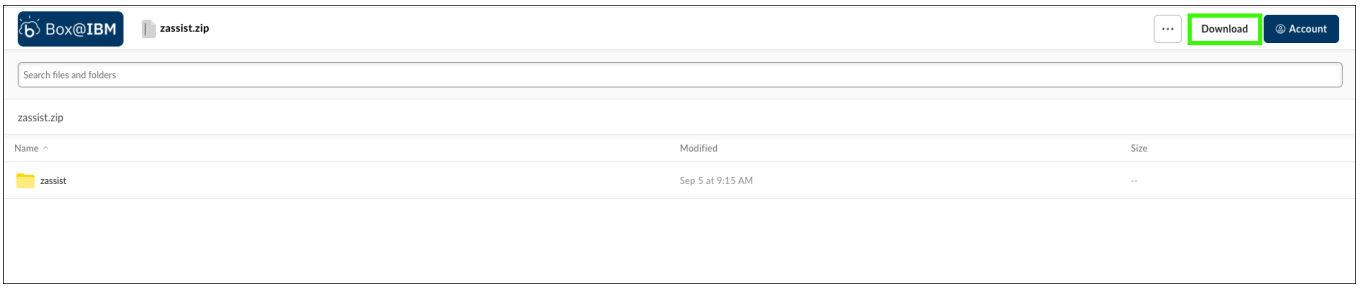
The **zassist** utility is an executable program that automates the ingestion of client documentation into the RAG for watsonx Assistant for Z. A version of zassist is available for download for IBMers and Business Partners for conducting pilots. Follow the steps below to download and install **zassist**.

How do clients get the zassist utility?

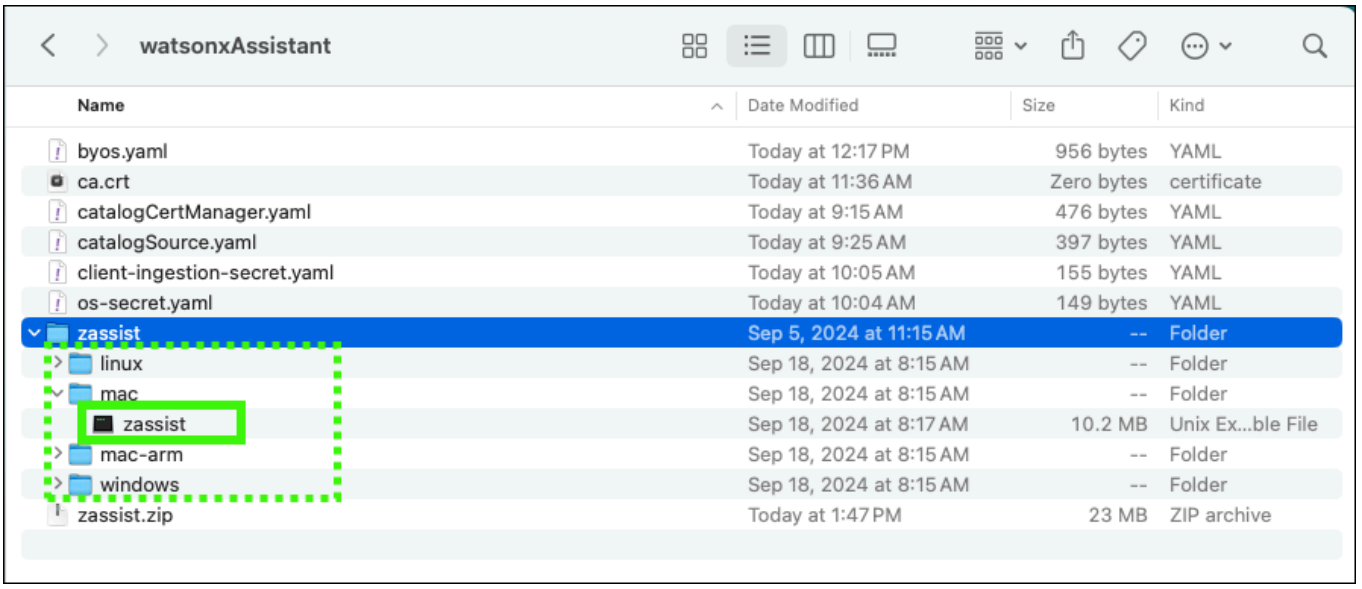
The utility is available to clients through IBM Passport Advantage.

1. Click the following link and download the **zassist.zip** file.

   https://ibm.box.com/s/j3nt5iw4fqd5w2jgcqwxnjlsu8bpvl77



2. Extract the **zassist.zip** file.
3. Locate the appropriate file for your local workstation's operating system.

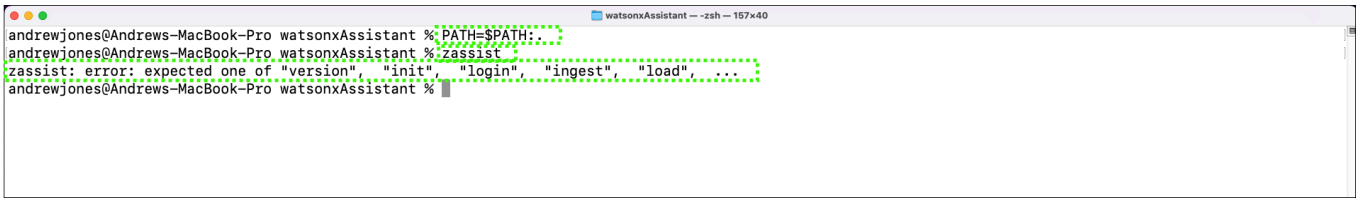


4. Either copy the appropriate **zassist** file to a directory in your PATH, or add the appropriate directory to your PATH environment variable.

   Additional information for running the preceding tasks can be found here.

5. Run the **zassist** command to verify it is working.

```
zassist
```



Mac/OS users may need to adjust security settings.

The **zassist** binary may cause a security exception. Adjust the security settings by opening the **System Settings** utility and clicking **Privacy & Security**. Under **Security** locate the message about the **zassist** binary and click **Allow Anyway**. Return to the terminal window and try running teh command again.

Ingest client documentation using zassist

With the **zassist** command installed, you are now able to begin ingesting data.

Step-by-step guidance for ingesting documents using zassist is provided in the IBM watsonx Assistant for Z documentation.

1. Follow the directions here to ingest documents using zassist.

   The steps are not repeated in this lab guide. The following video illustrates the steps to ingest a single document. The document that is ingested in the video is a compressed PDF of the **IBM z/OS Continuous Delivery** Red Piece. You can download a copy of this document here.

   Note: The video has no audio.

Don't see the video in the PDF version of the lab guide?

If you are viewing the PDF of the lab guide, you can access the video here.

Adjusting the search behavior

Do you recall the **Metadata** field when you configured your assistant?

The Metadata field provides a way to adjust your assistant's behavior during conversational search for your OpenSearch instance. Now that you have your own docs that are ingested for conversational search, you can set the metadata field for your assistant to use those documents in its content-grounded search. If you leave the metadata field empty, then it defaults to settings found to perform well but may not use the ingested documents as part of the search results.

If you leave the Metadata field empty, OpenSearch will rely on the default settings, which means OpenSearch will search all of the default IBM-provided documentation and all of the ingested customer documentation using the following value:

```
{"ibm_indices":"*_ibm_docs_slate,*_ibm_redbooks_slate",
“customer_indices”:“customer_*”}
```

Replacing the wildcard string with an explicit list of indices allows for personalization. The metadata setting is where you can input specific indices (pointing to the underlying documentation) that you want your assistant to use for the content-grounded search. There are over 220 products and topics that the OpenSearch instance has IBM Documentation for. You can find those indices and products here.

You can input a subset of indices into the “Metadata” field in cases where you only want your assistant to gather context for specific IBM products or topics. The specific indices can be listed out in this format:

```
{“ibm_indices”:“<comma separated index values>”,“customer_indices”:“customer_*”}
```

For example, if you only want your assistant to reference documentation for “Db2 Analytics Accelerator for z/OS” and no ingested client documentation, you can enter the following into the metadata field:

```
{“ibm_indices”:“ss4lq8_ibm_docs_slate”}
```

If you have a mix of IBM Documentation and client documentation ingested, then there's an optional search string that you can use to set the "weights" used for each.

For example:

```
{"doc_weight":
{"product_docs":0.5,
"customer_docs":0.5},
"ibm_indices":"*_ibm_docs_slate,*_ibm_redbooks_slate",
"standardize":true,
"customer_indices":"customer_*"
}
```

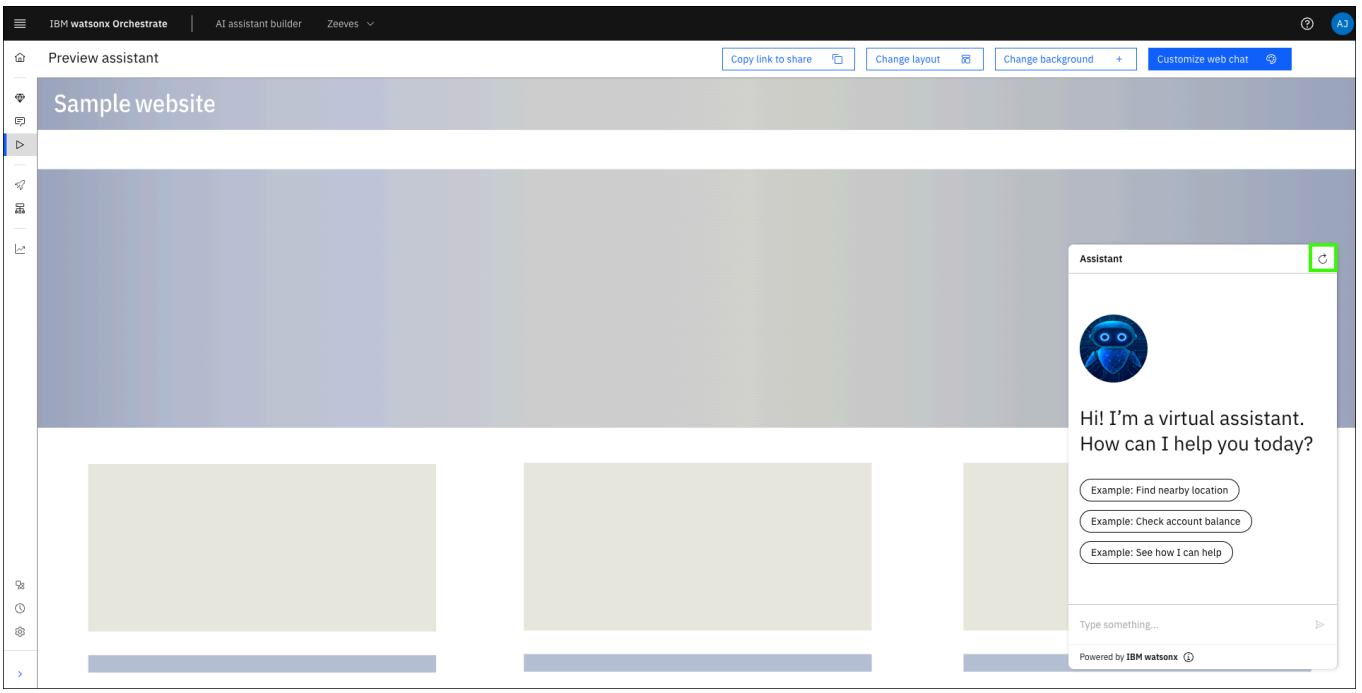
In this case, "product_docs" is the weight that is assigned to "ibm_indices" and "customer_docs" is the weight that is assigned to "customer_indices". For more information on customizing the metadata field for conversational search, refer to this supplemental video found here.

Verify the document that is ingested is now returned as a source file for a query

Use the watsonx Orchestrate AI assistant builder to verify your document ingestion.

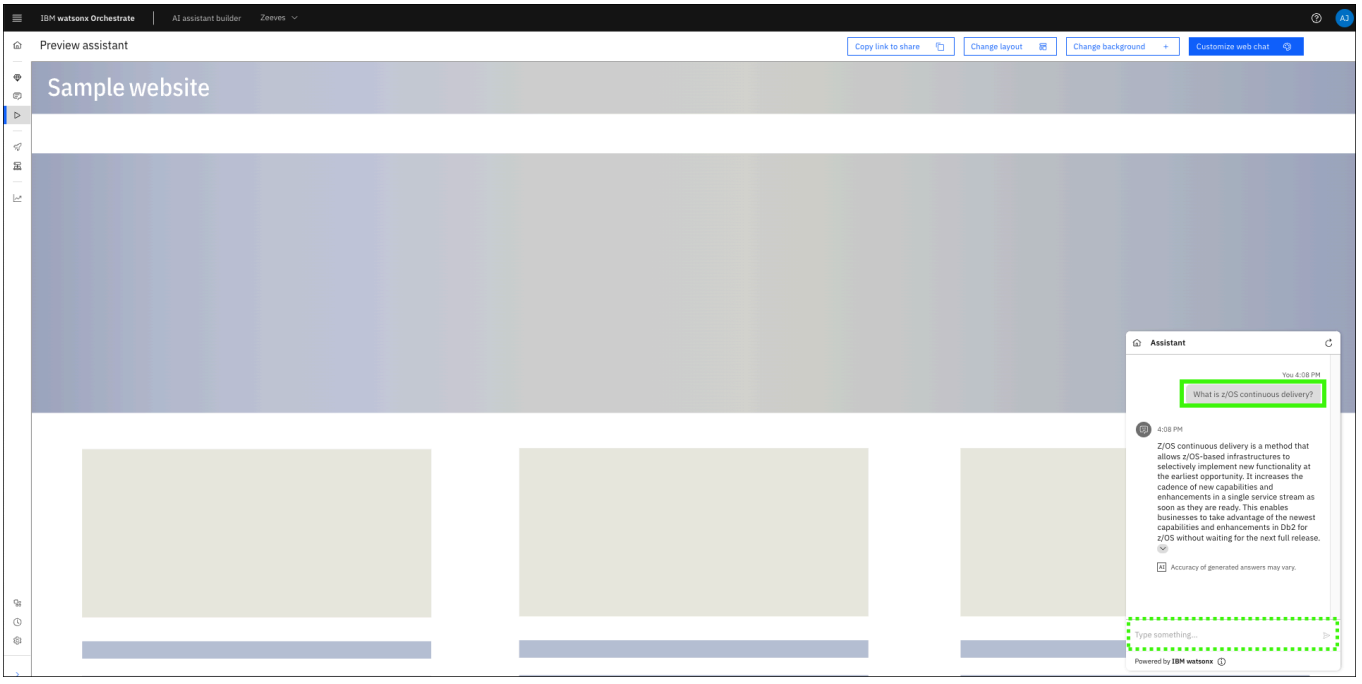
> ⚠ **You may not receive the same results as shown below.**
>
> In the 4Q 2024 release of IBM watsonx Assistant for Z, additional IBM documents were added to the RAG including many IBM RedBooks. The new data changes the results returned when using the sample IBM Red Piece ingested earlier. To reproduce the results shown, you can modify the Metadata field for your assistant to remove the IBM Redbooks from the IBM indicies:
>
> ```
> {"doc_weight":
> {"product_docs":0.5,
> "customer_docs":0.5},
> "ibm_indices":"*_ibm_docs_slate",
> "standardize":true,
> "customer_indices":"customer_*"
> }
> ```

1. Hover over the **Home** (⌂) icon and click **Preview**.
2. Click the **Restart conversation** (↻) icon.

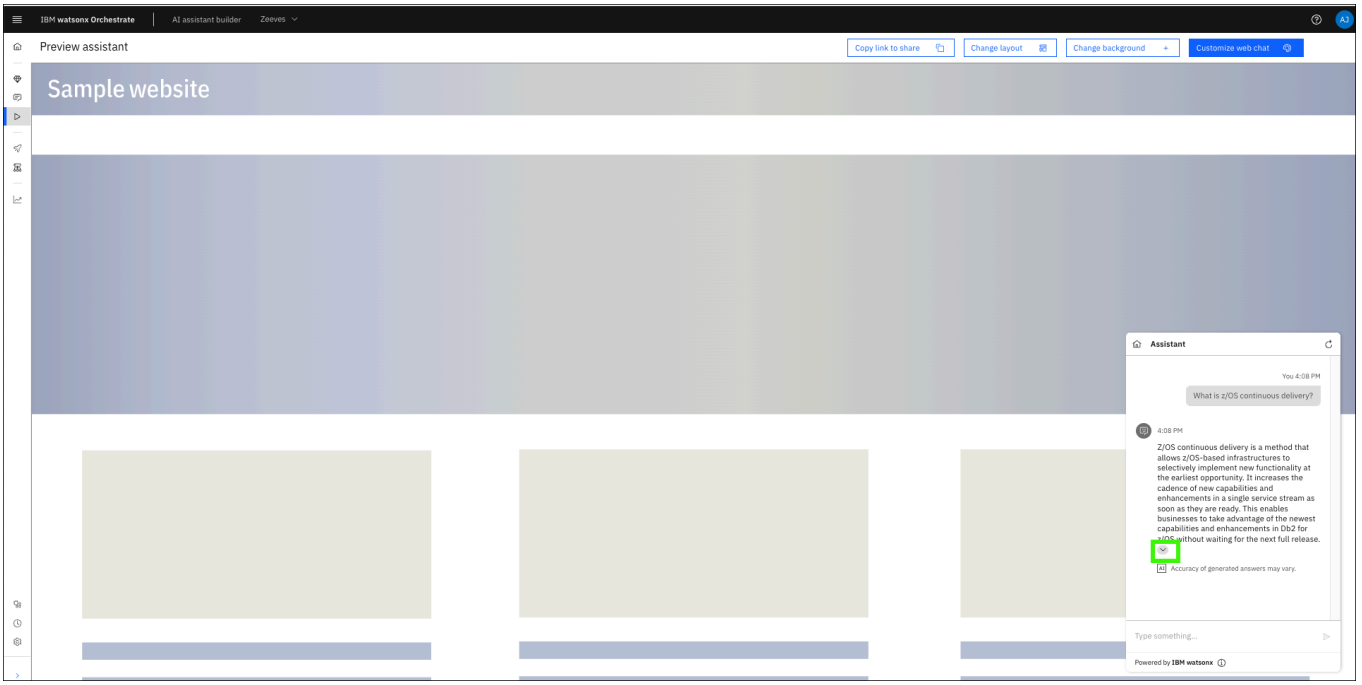


3. Enter the following prompt in your assistant.

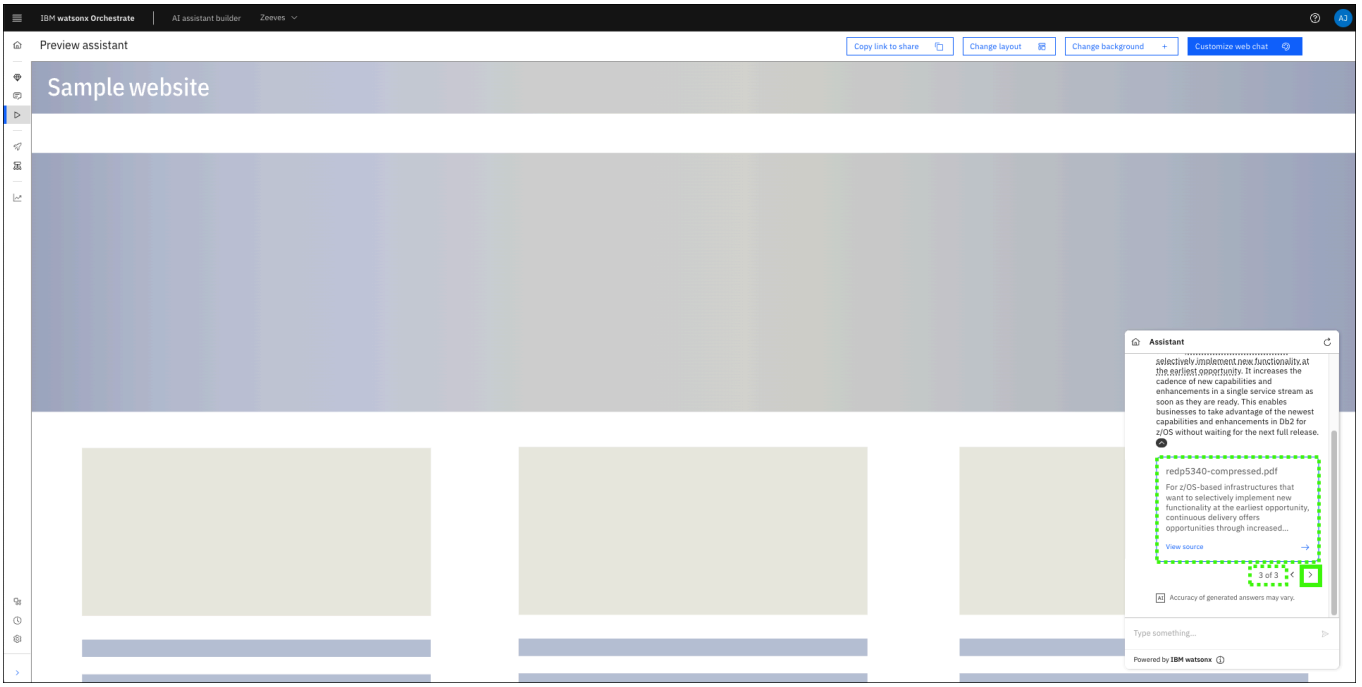
```
What is z/OS continuous delivery?
```



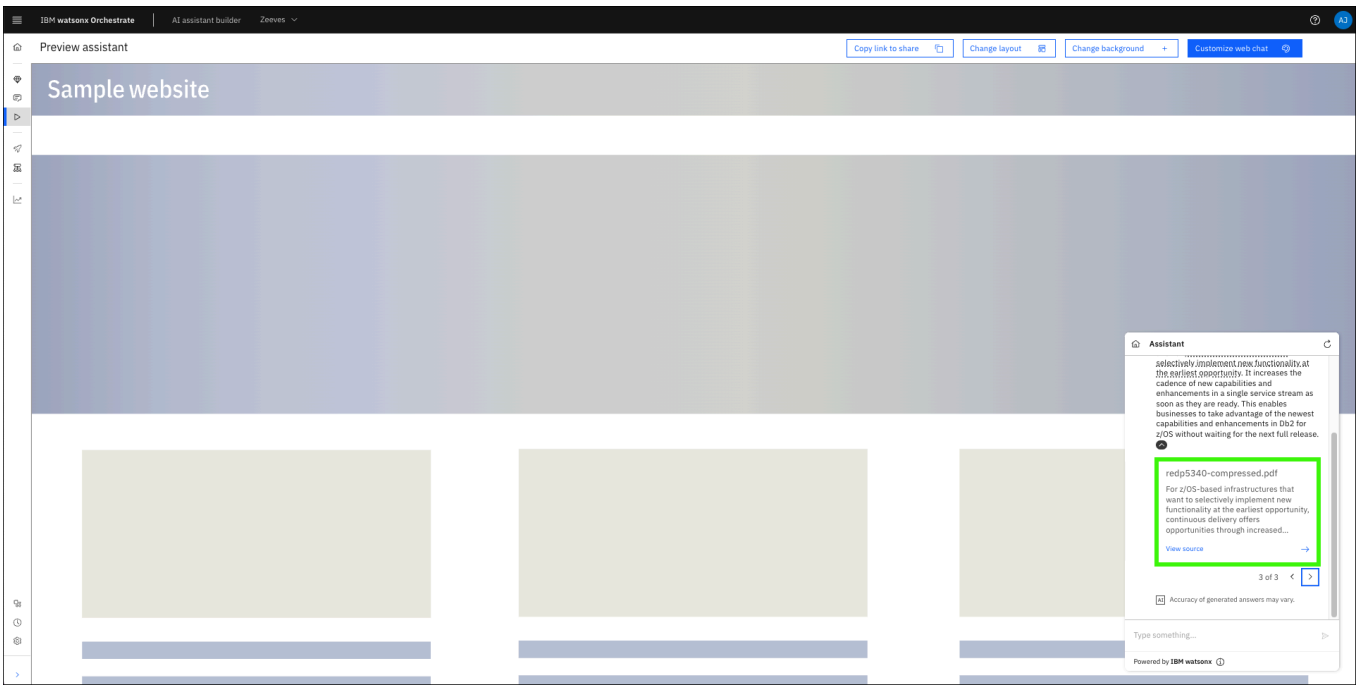
4. Clicking the **Down arrow** ( ).



5. Click through the list of resources and find the reference to the Red Piece document you ingested.

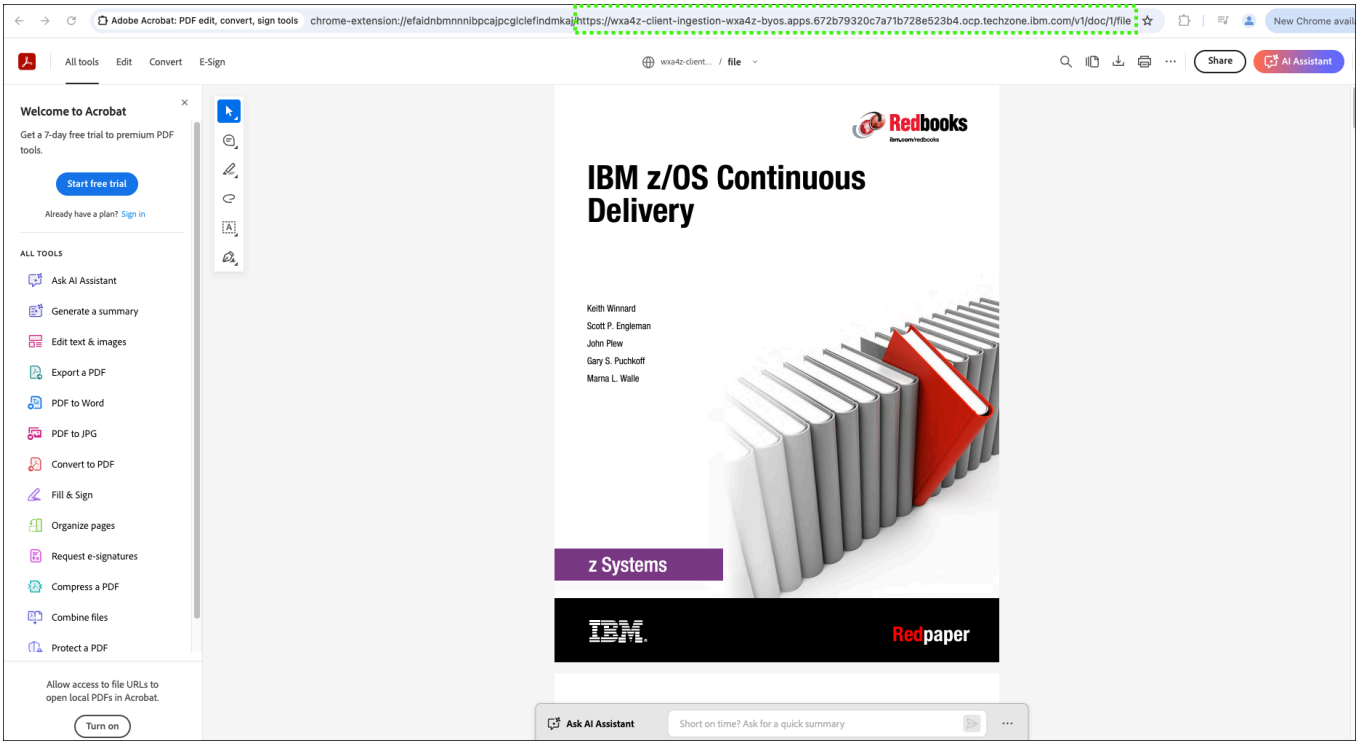


6. Click the ingested document reference.



7. Accept the security risk to view the source document.

   The steps to accept the security risk for the document are not shown. The risk occurs because the certificate for the connection to the SNO instance is not secure. Notice that the URL contains the path to your SNO instance route.



You are encouraged to experiment with the metadata field! Try setting the metadata field to the following, which weights ingested docs higher than the product docs. Note, if the sample metadata below includes the IBM Redbooks:

```
{"doc_weight":
{"product_docs":0.2,
"customer_docs":0.8},
"ibm_indices":"*_ibm_docs_slate,*_ibm_redbooks_slate",
"standardize":true,
"customer_indices":"customer_*"
}
```

After you have configured all the settings for Conversational Search on the page, be sure to click **Save** in the upper-right of the page.

For client pilots

If you or your client have other documents to ingest, you can do so by repeating the steps using zassist. The Velocity Pilot ITZ environment is limited in compute and storage capacity. The following limits should be adhered to:

- Loading documents can take a long time, especially with > 100 MB of text.
- It is recommended to run large loads late at night.
- When loading, ensure your workstations does not sleep during the process.
- If you receive a **batch time error**, set the batch size to a lower number for that command. For example:

```
zassist ingest . -s 50
```

After ingesting all your additional documents, proceed to the next section to learn about adding skills to your assistant.

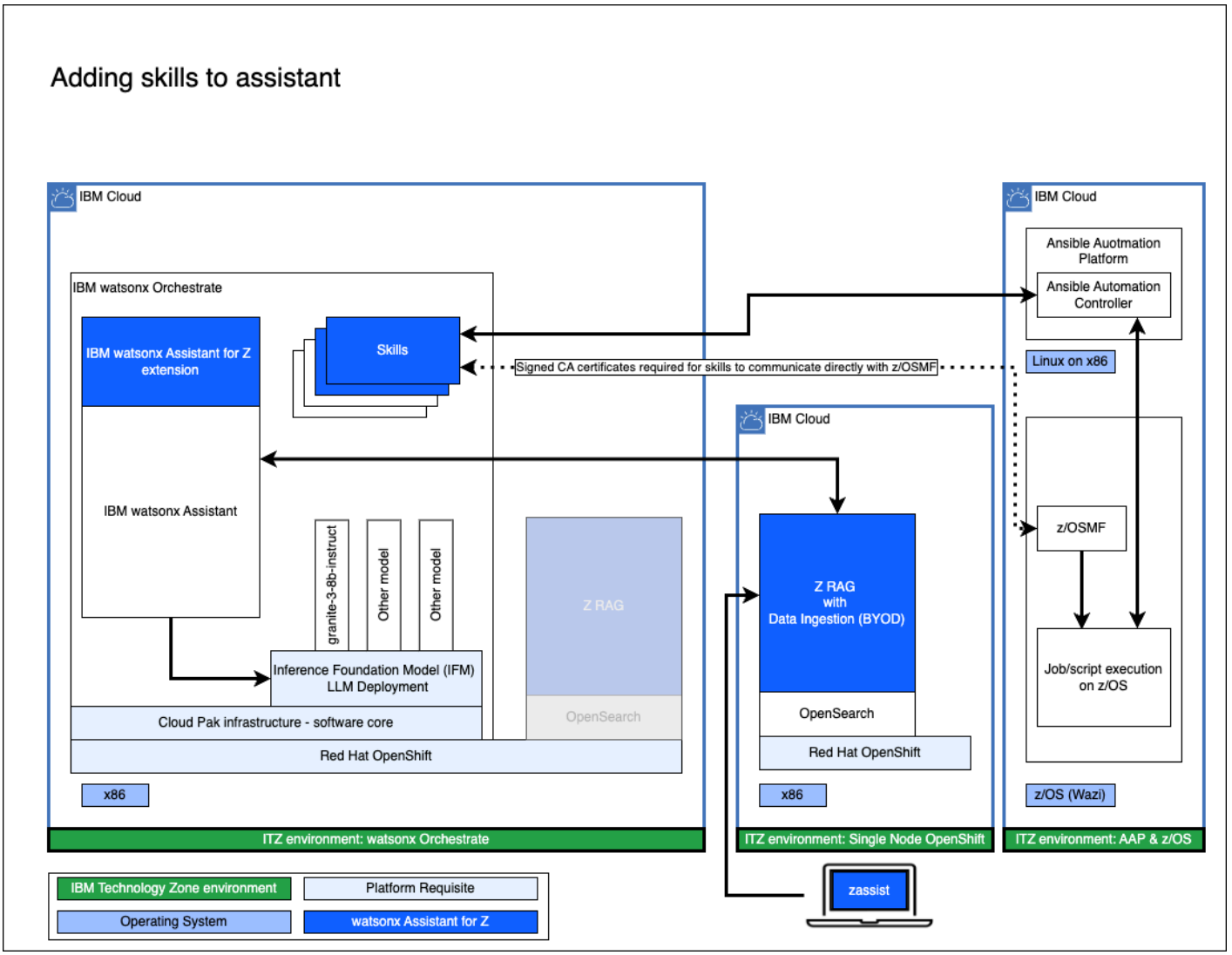
Adding skills to the assistant

Getting started with skills and actions

Watsonx Assistant for Z automates a range of IBM Z related tasks through assistant interactions by importing skills. Each skill is a pre-defined automation that accomplishes some unit or units of work by running tasks. For example, skills can view z/OS IPL information or work with z/OS datasets.

Watsonx Assistant for Z extends watsonx Orchestrate, allowing users to build new skills from Ansible Automation platform or z/OS Management Facility (z/OSMF) through the Z Skills Accelerator extension. This extension connects Ansible and z/OS application programming interfaces (APIs) and imports automation as Ansible Playbooks, JCL, or REXX as skills.Learn more importing and building skills here.

Below is a high-level, logical architecture of the environment you will deploy in this section.



Environments

Watsonx Orchestrate

The Z Skills Accelerator extension is already configured in your watsonx Orchestrate IBM Technology Zone (ITZ) environment. You can use this component to import new skills.

Ansible Automation Platform and Wazi as a Service

To import skills for automations, you use Ansible Automation Platform (AAP) and Wazi as a Service (Wazi aaS) to serve as the z/OS back-end. Learn more about AAP here. Learn more about Wazi, here.

The two resources are provisioned together in the ITZ environment that you reserved earlier. This environment enables the ability to manage and automate z/OS tasks and subsystems with various preinstalled ansible playbooks. It includes a z/OS back-end (Wazi as a Service) with all needed prerequisites.

The playbooks provided cover various use cases for automating z/OS management. Ansible's capabilities for automating various Z-specific tasks are not limited to the use cases that are preinstalled in the AAP instance. The preinstalled playbooks are tasks from the 'IBM z/OS core collection'. Using this environment accelerates the ability to showcase the value of watsonx assistant for Z, and to get started with simple automations that can be expanded.

The ITZ environment gives you access to AAP, which is preconfigured to target the accompanying z/OS Wazi system, along with web-based access to AAP to experiment with different playbook templates. These templates are imported into watsonx Orchestrate as skills and connected to your assistant.

For more information on the AAP and Wazi z/OS environments, refer to this document.

The playbook templates that are preinstalled in AAP cover various use cases, which you can explore, including:

- z/OS Certificate Management (create, delete, list, and renew certificates)
- dataset management (create, delete, fetch datasets)
- Submit JCL
- Run Operator commands
- Run TSO commands
- And more

Each of the sections that follow build upon each other. Complete each to successfully enhance your assistant by starting with Explore Ansible Automation Platform.

Explore Ansible Automation Platform

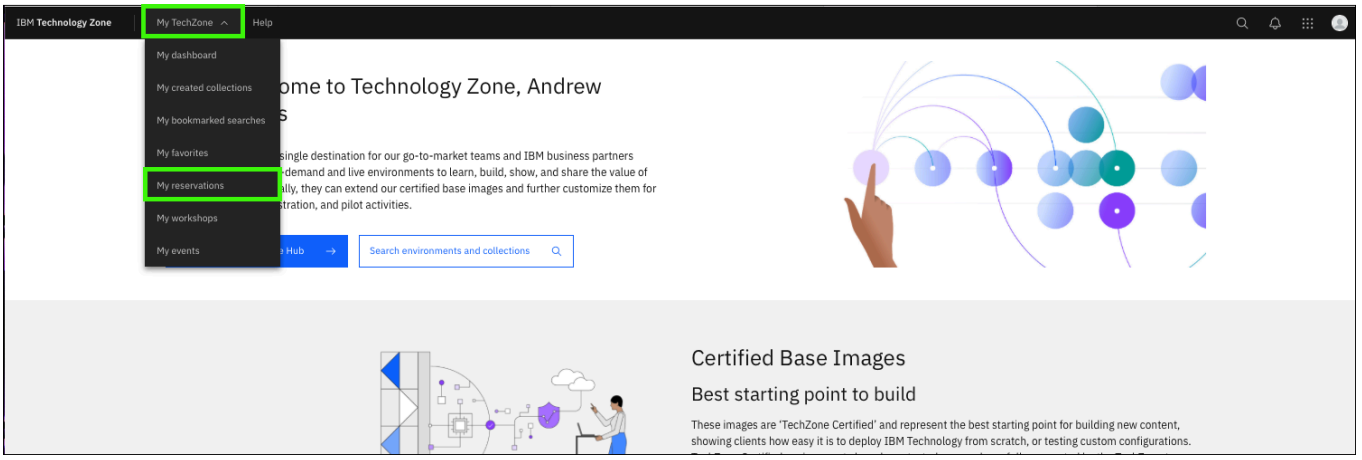
After you reserved the Ansible Automation Platform (AAP) and Wazi z/OS environment in IBM Technology Zone (ITZ) and the reservation is in the **Ready** state, follow these steps to explore AAP.

Access the AAP and Wazi as a Service environment

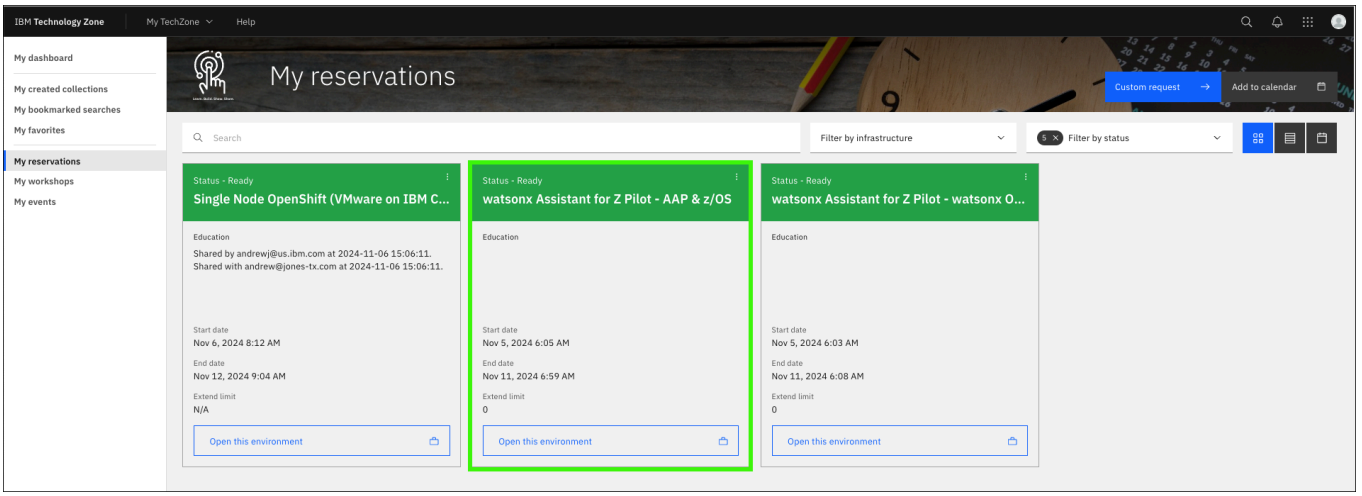
> **Be sure to record the information as instructed**
>
> Several of the steps below instruct you to record values from your ITZ reservation. Be sure to do this as they will not only be used in this section, but also in later sections of the lab guide.

1. In the IBM Technology Zone portal, expand **My TechZone** and select **My Reservations**, or click the following link.

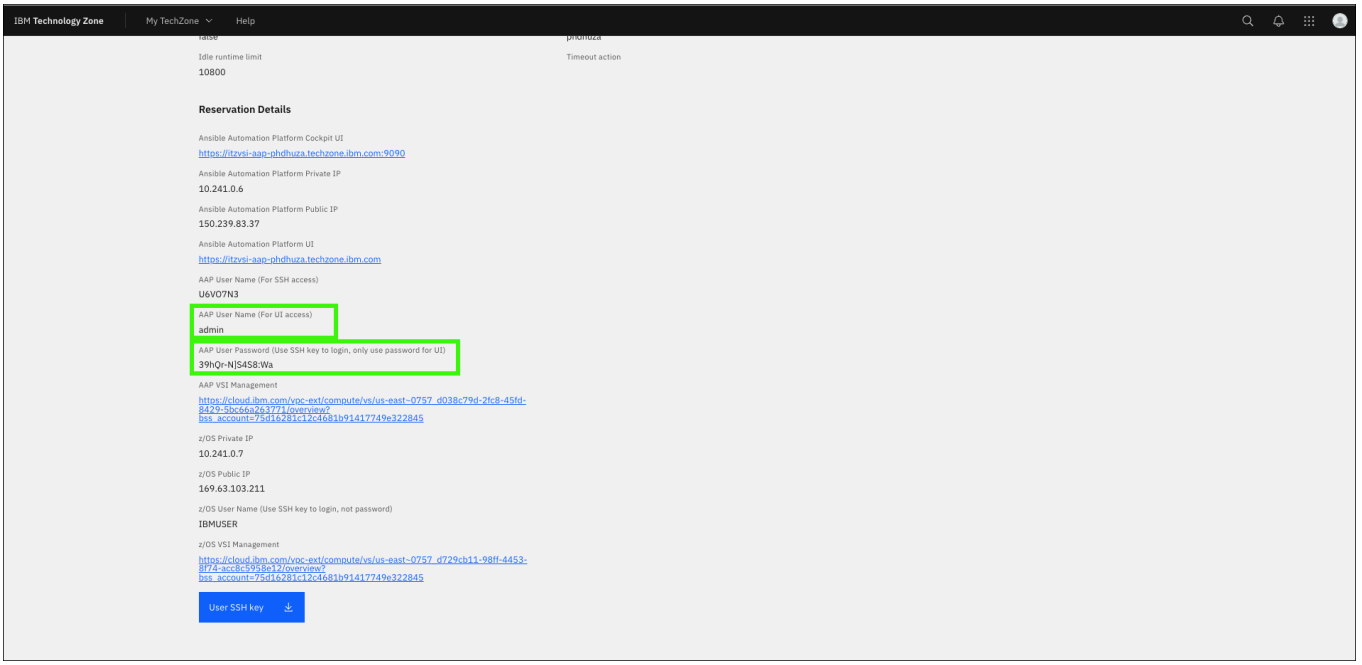
   ITZ My reservations



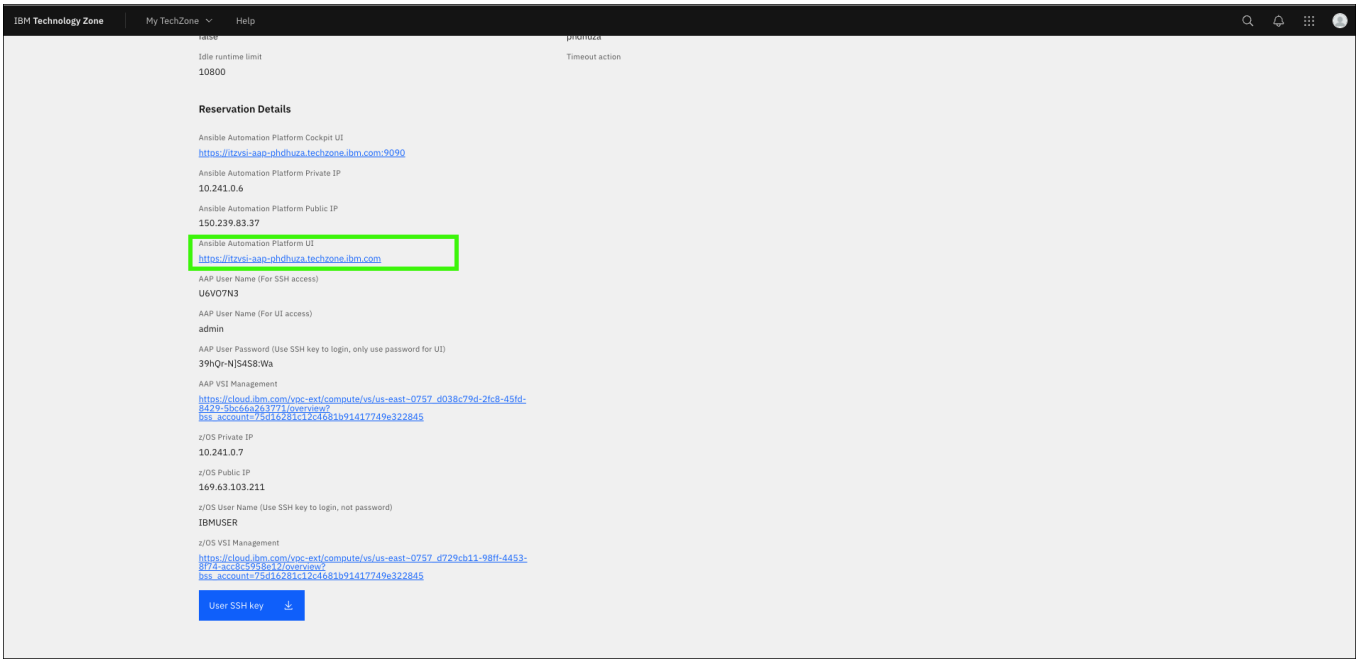
2. Click the **watsonx Assistant for Z Pilot - AAP & z/OS** tile.



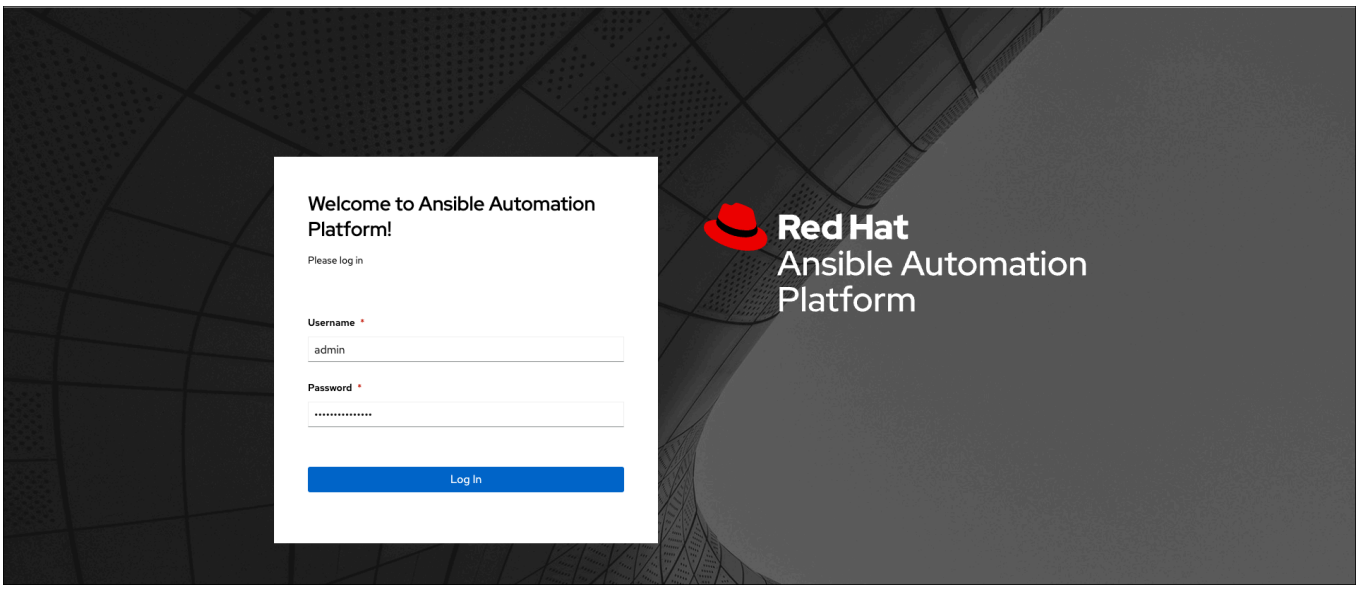
3. Locate and record the **AAP User Name (For UI access)** and **AAP User Password** fields.



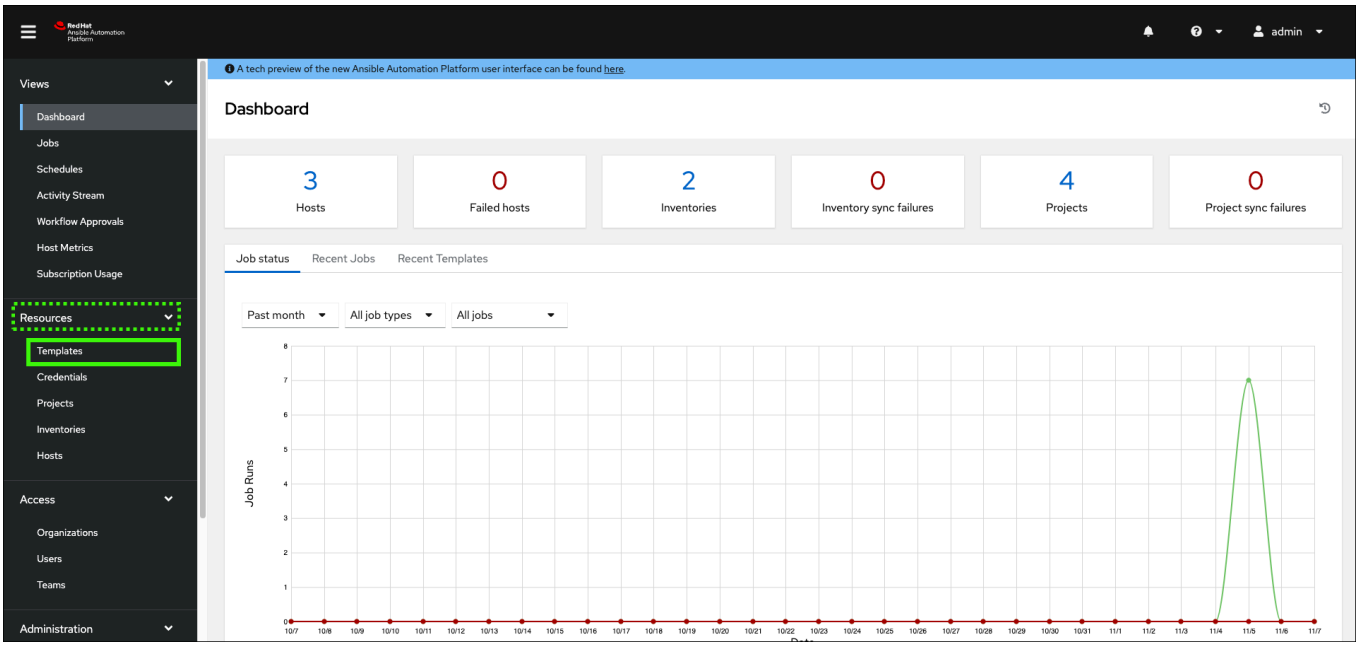
4. Record and then click the **Ansible Automation Platform UI** link.



5. Enter the **Username** and **Password** that is recorded in step 3 and click **Log In**.

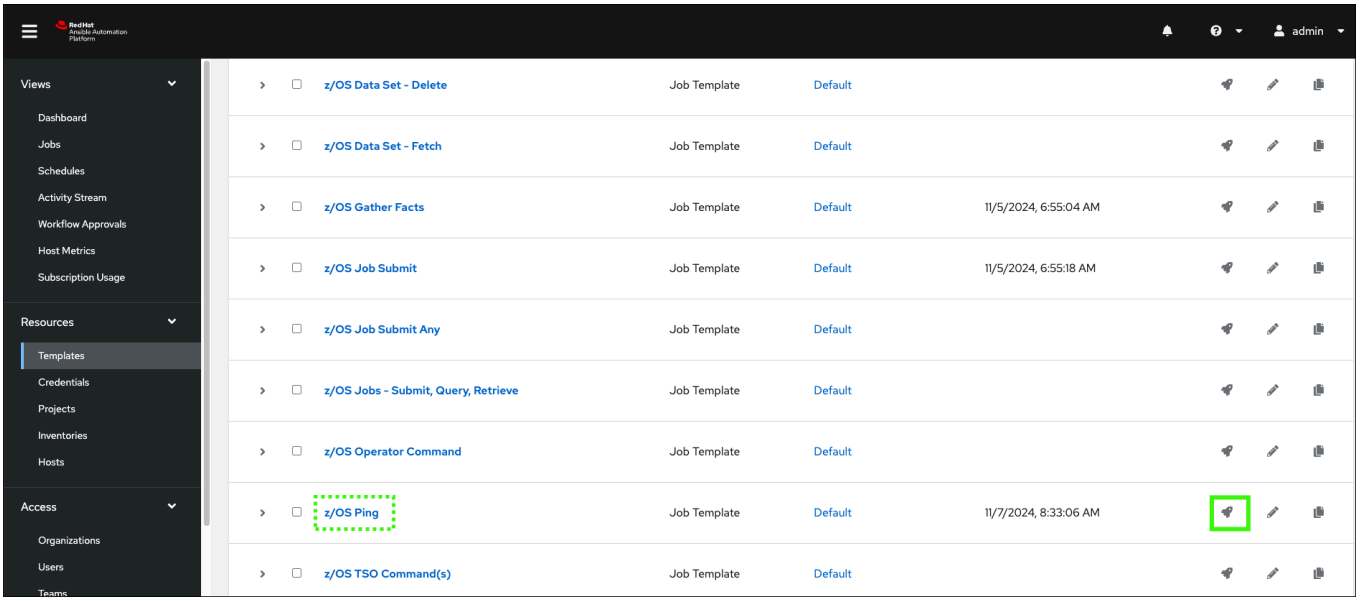


6. Click **Templates** under the **Resources** section.

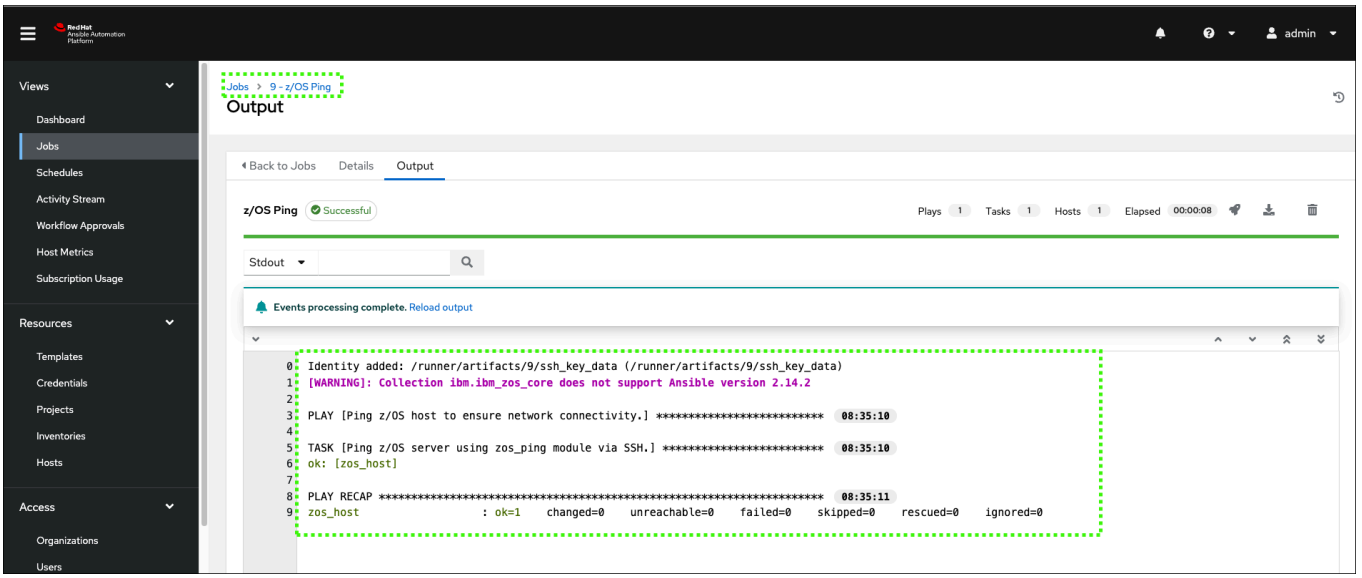


> **The AAP instance is preconfigured to the Wazi aaS instance**
>
> Note that because the AAP instance and the back-end z/OS system are preconfigured, no changes are needed to execute the templates and they will target your provisioned z/OS system by default.

7. Locate the **z/OS Ping** template and click the rocket (🚀) icon to start the template.



8. Observe the **z/OS Ping** job run.



Take time to explore the other templates that are ready to use. Learn more about the automation capabilities here.

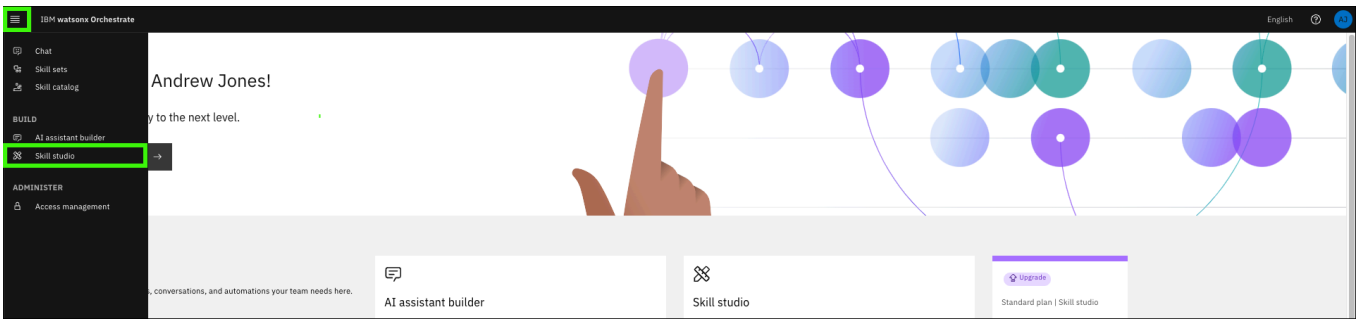
Importing skills from Ansible Automation Platform

Now that you understand Ansible Automation Platform (AAP) and the preinstalled automations available, you can import them as skills into your watsonx Orchestrate instance, which is used for assistant guided actions.

Import skills into your assistant

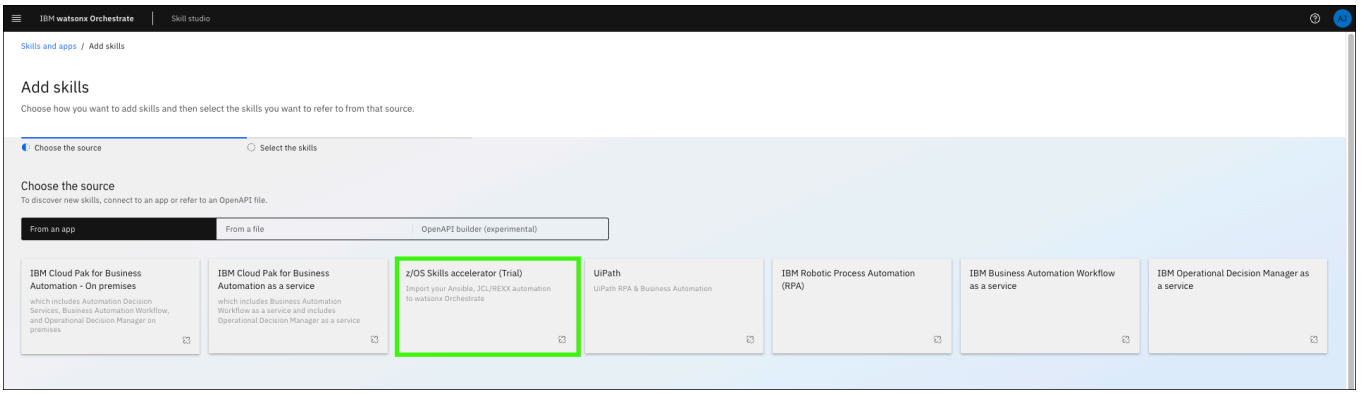
The next steps assume that you have an active browser window to the watsonx Orchestrate ITZ cloud account. If you do not, refer to the initial steps in Creating an assistant and configuring conversational search.

1. Return to your watsonx Orchestrate instance and expand the main menu and click **Skill studio**.



2. Expand **Create** and click **Import API**.

3. Click the **z/OS Skills accelerator (Trial)** tile.



4. Enter the following values in the **z/OS Skills accelerator** form and then click **Connect**.

   Use the **URL**, **User Name**, and **Password** values recorded in the Explore Ansible Automation Platform section earlier.

a: Connection Type: `ansible`

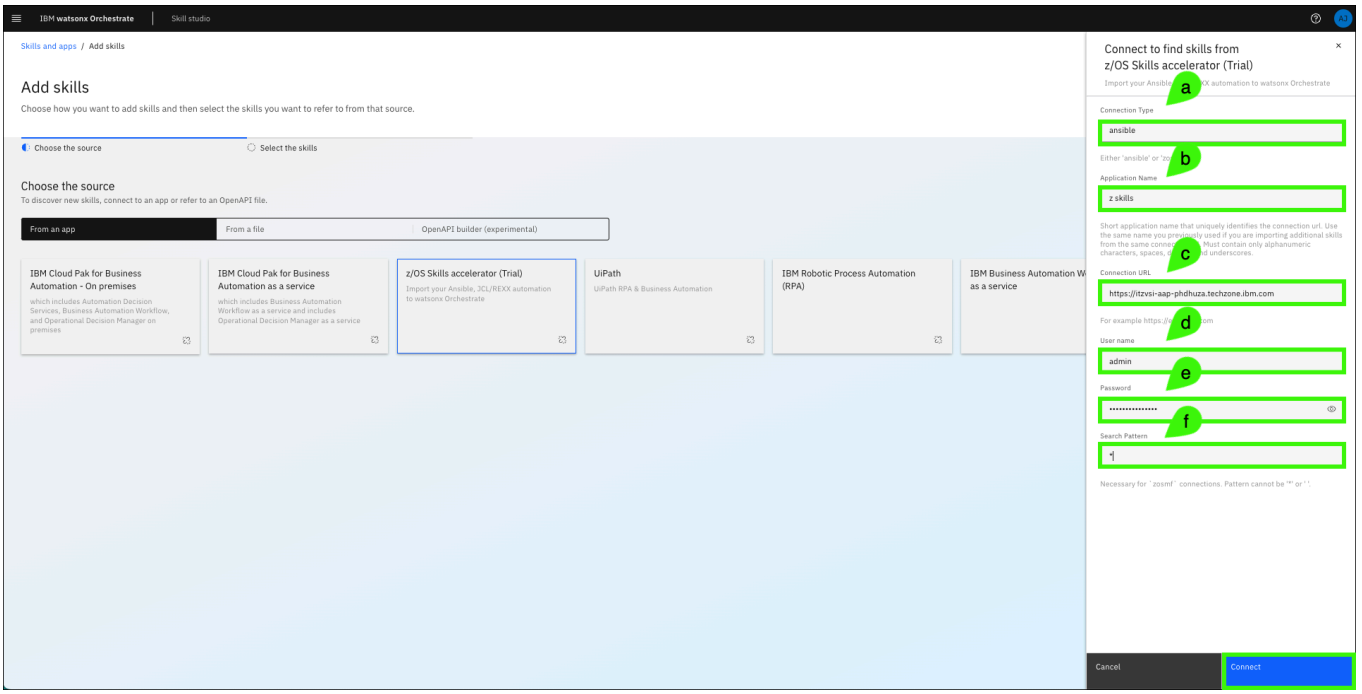
b: Application Name: <enter a meaningful name for the skills that you will import> - *be sure to remember this name, you will need in the next section*

c: Connection URL: <enter the URL for your AAP UI>

d: User Name: <enter the AAP User Name (for UI access)>

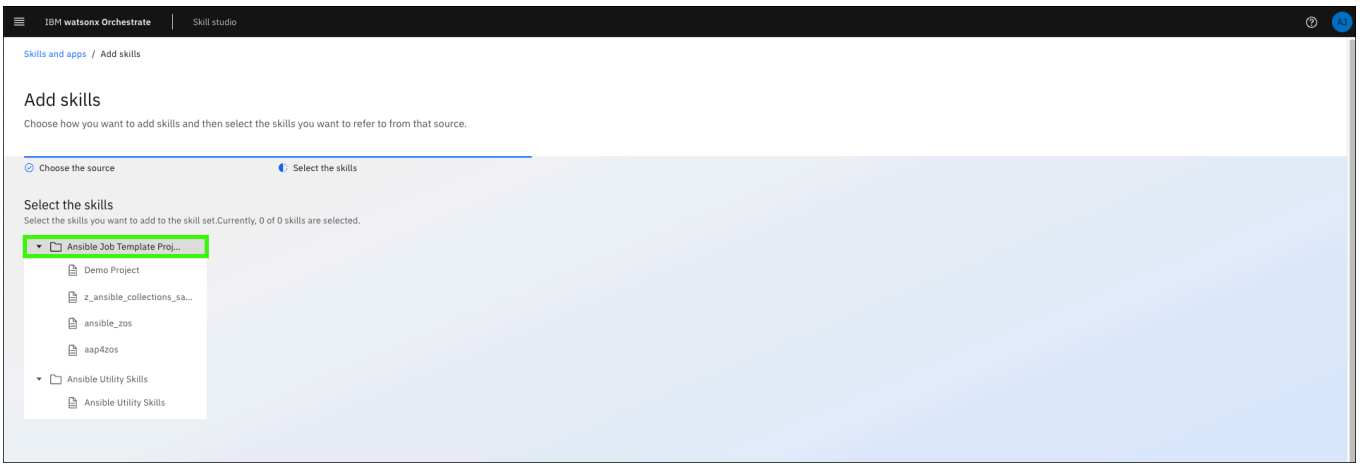
e: Password: <enter the AAP User Password>

f: Search Pattern: `*`

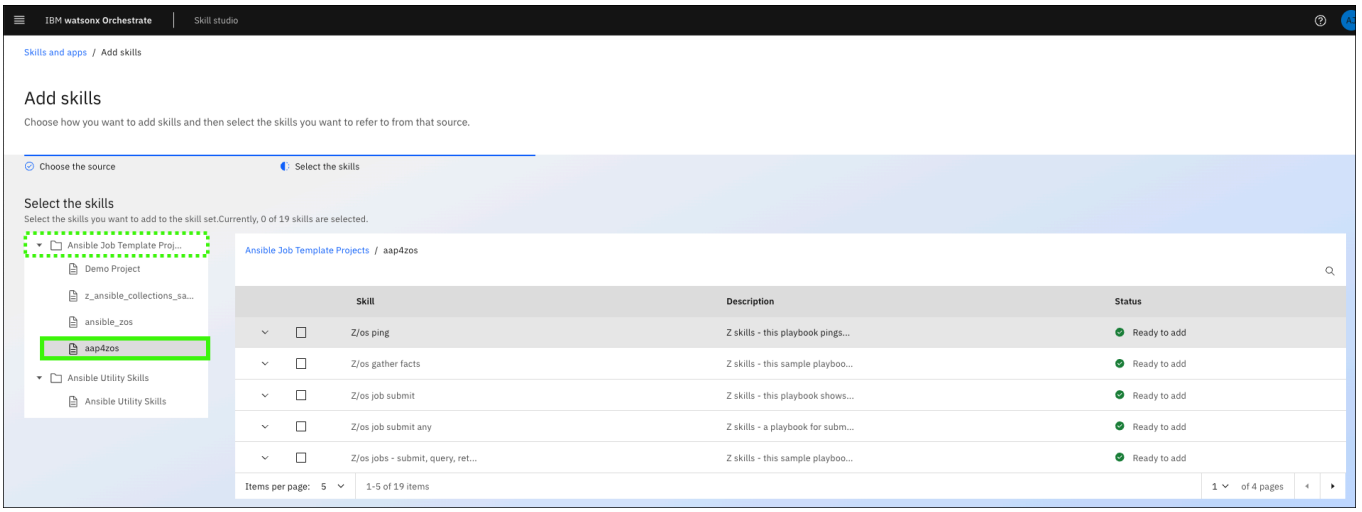


5. Expand the **Ansible Job Template Proj...** folder.

> **Explore the other available skills**
>
> Take time to explore the available skills. There are many utility skills provided out of the box with the “Z Skills Accelerator” which are needed for actions such as retrieving the output of an Ansible skill. Consider importing these utility skills to enable more complete automation execution flows.

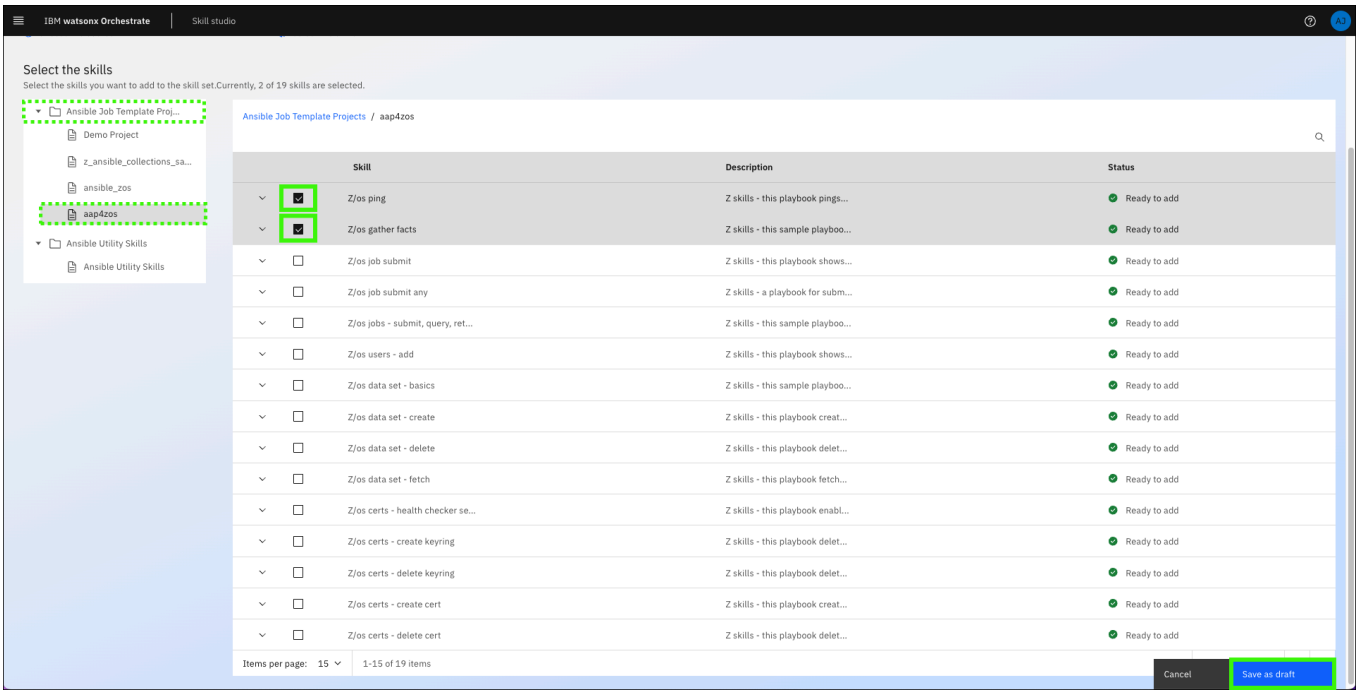


6. Click **aap4zos**.

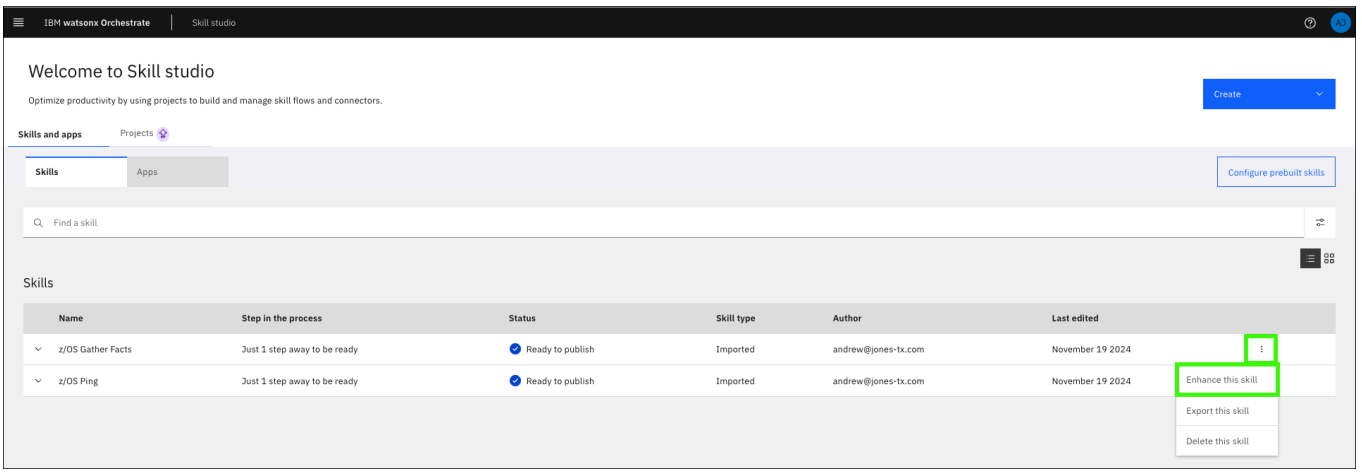


7. Select the skills you want to import into your application and then click **Save as draft**.

   For this lab, select the **Z/os ping** and **Z/os gather facts** skills.

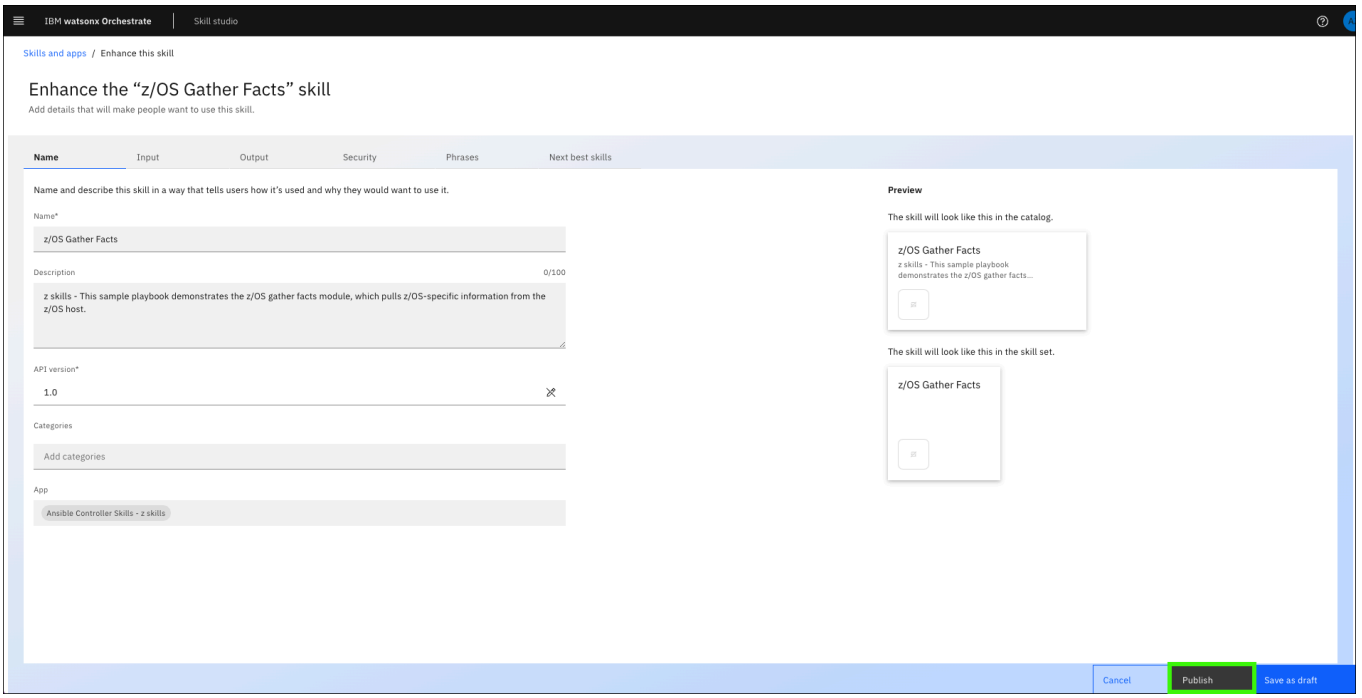


8. Click the ellipses (⋮) for the first skill and select **Enhance this skill**.

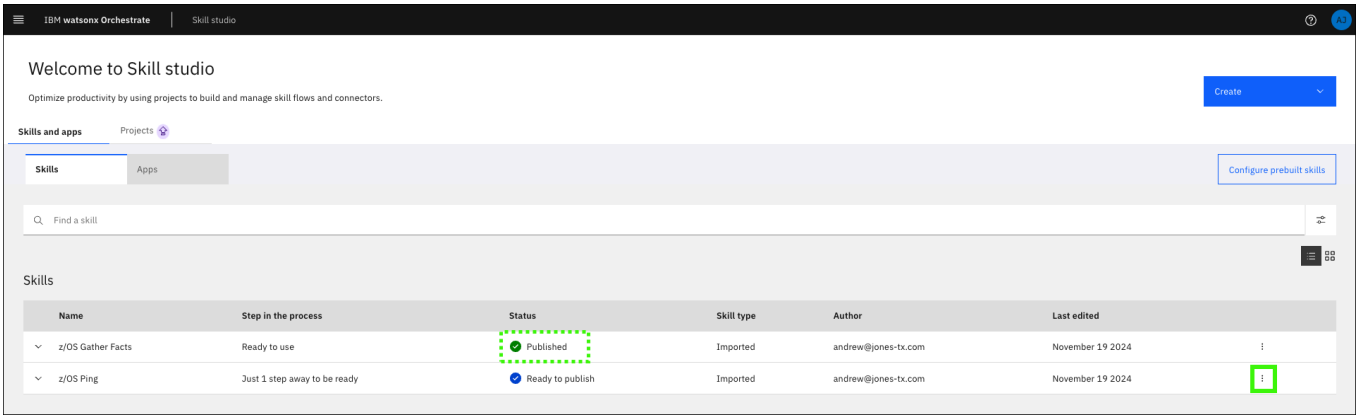


9. Review the skill enhancement options and then click **Publish**.

   On the **Enhance this skill** page, you can specify enhancements to the default skill. Refer to this documentation for more information on enhancing skills.



10. Repeat steps 8 and 9 for each skill you imported.

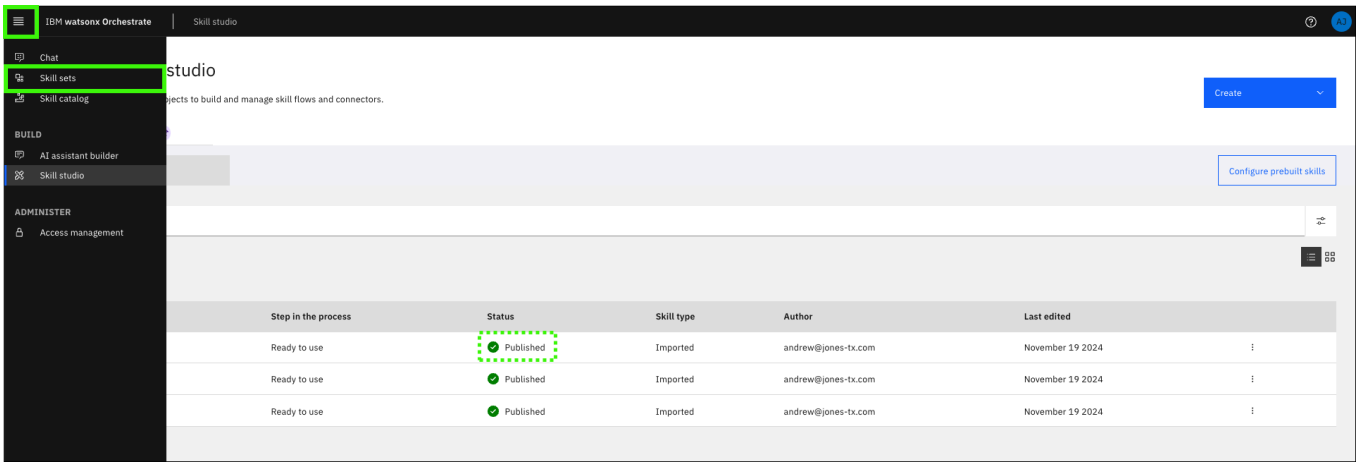


The selected skills are now ready for use and available to your assistant. In the next section, learn how to connect them to your assistant.

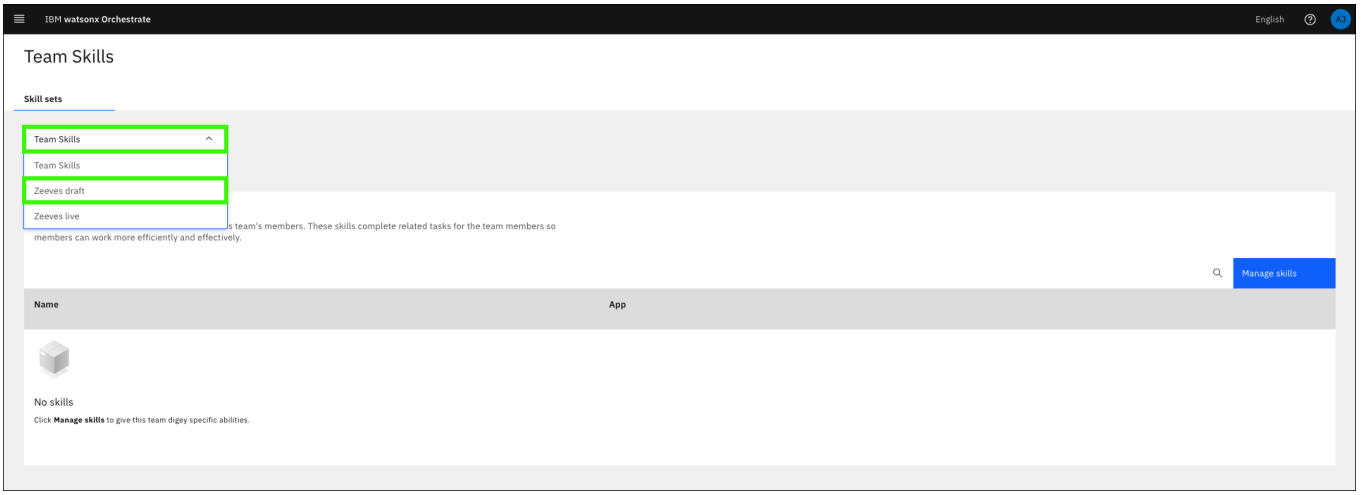
Connecting skills to your assistant

Once you have a subset of skills published, the application you created can be connected to your assistant.

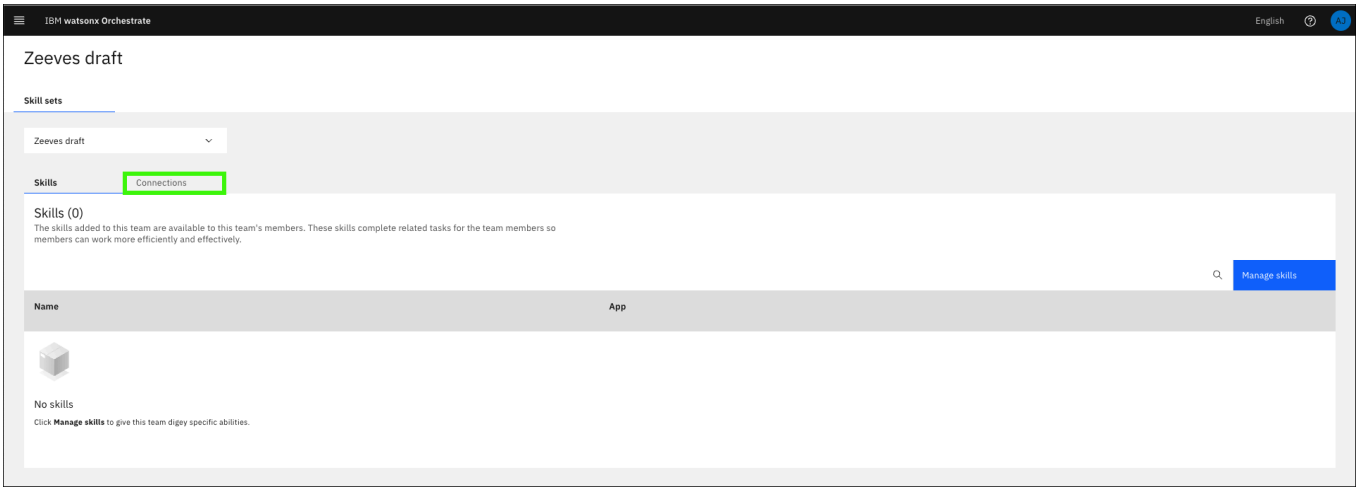
1. Expand the main menu and select **Skill sets**.



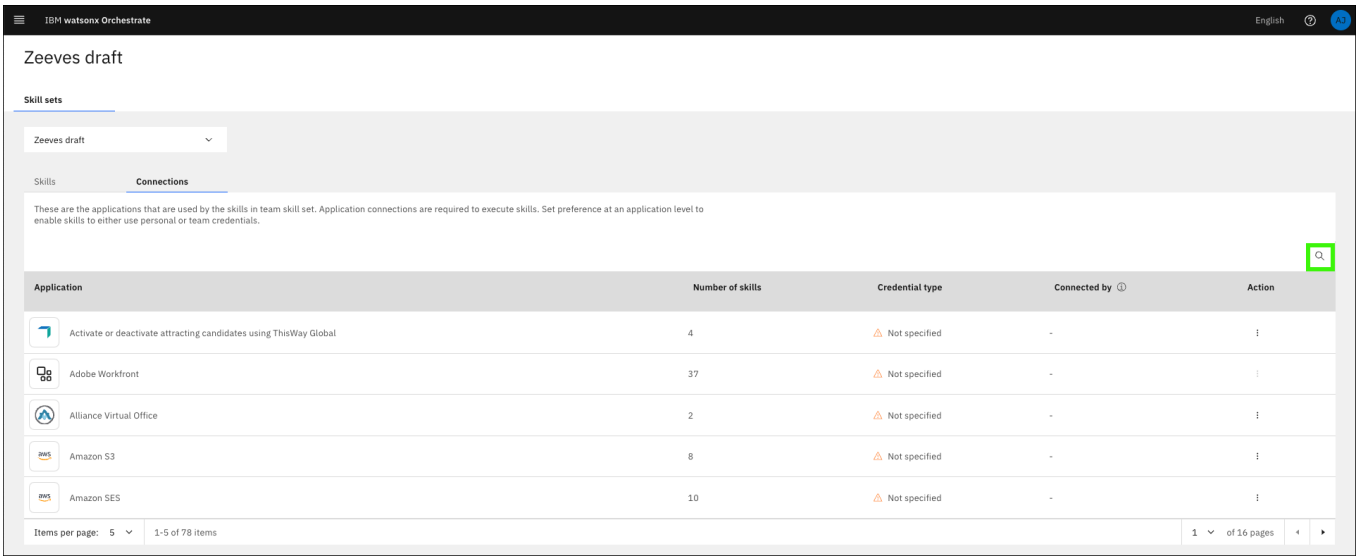
2. Click the **Team Skills** drop-down and select the **Draft** of your assistant.



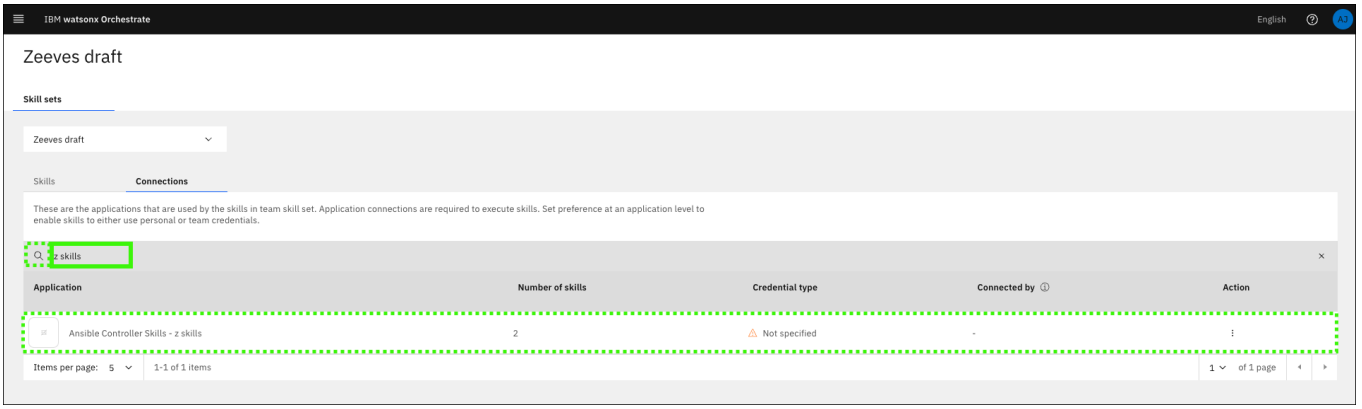
3. Click the **Connections** tab.



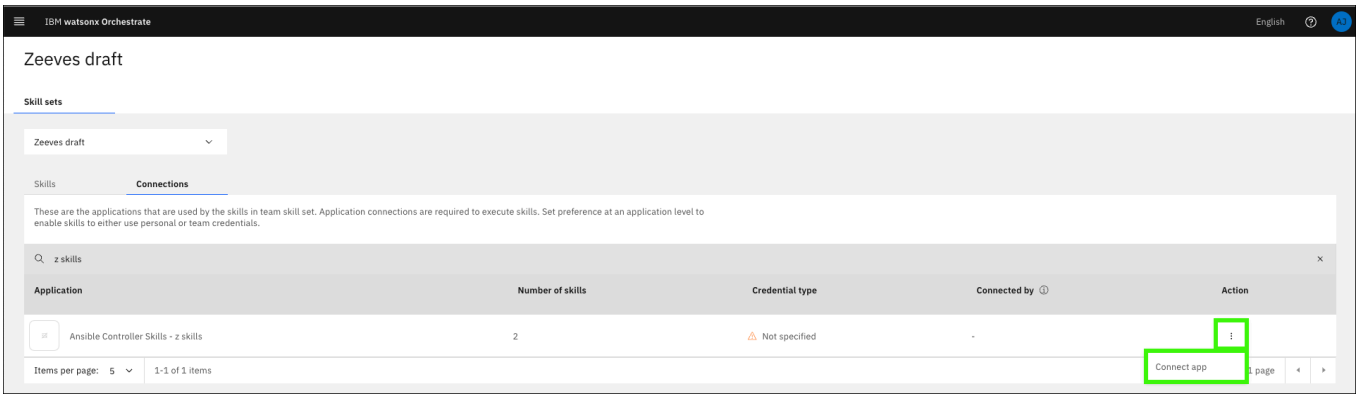
4. Click the **Search** ( ) icon.



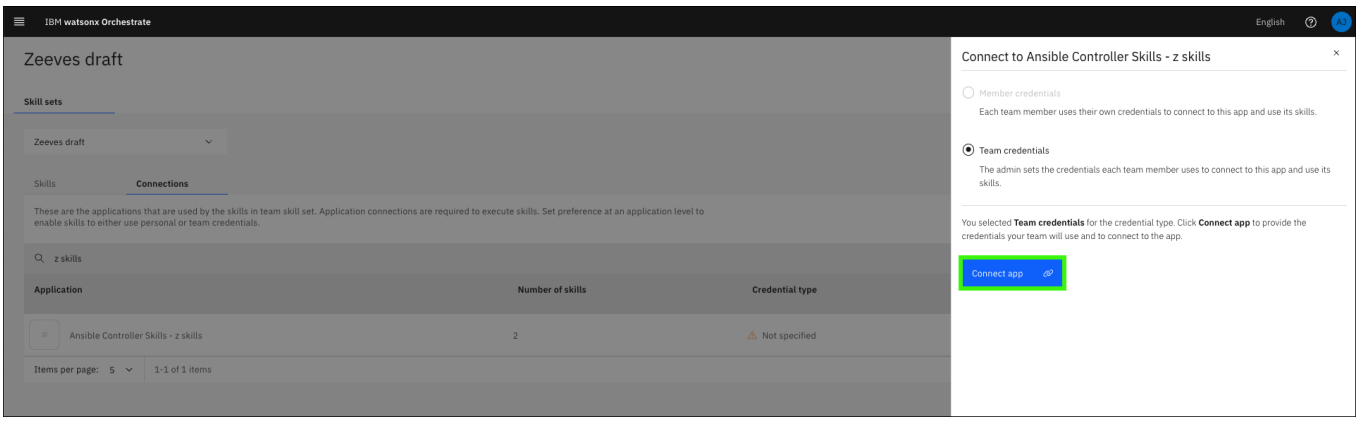
5. Search for the application name you specified in the previous section.



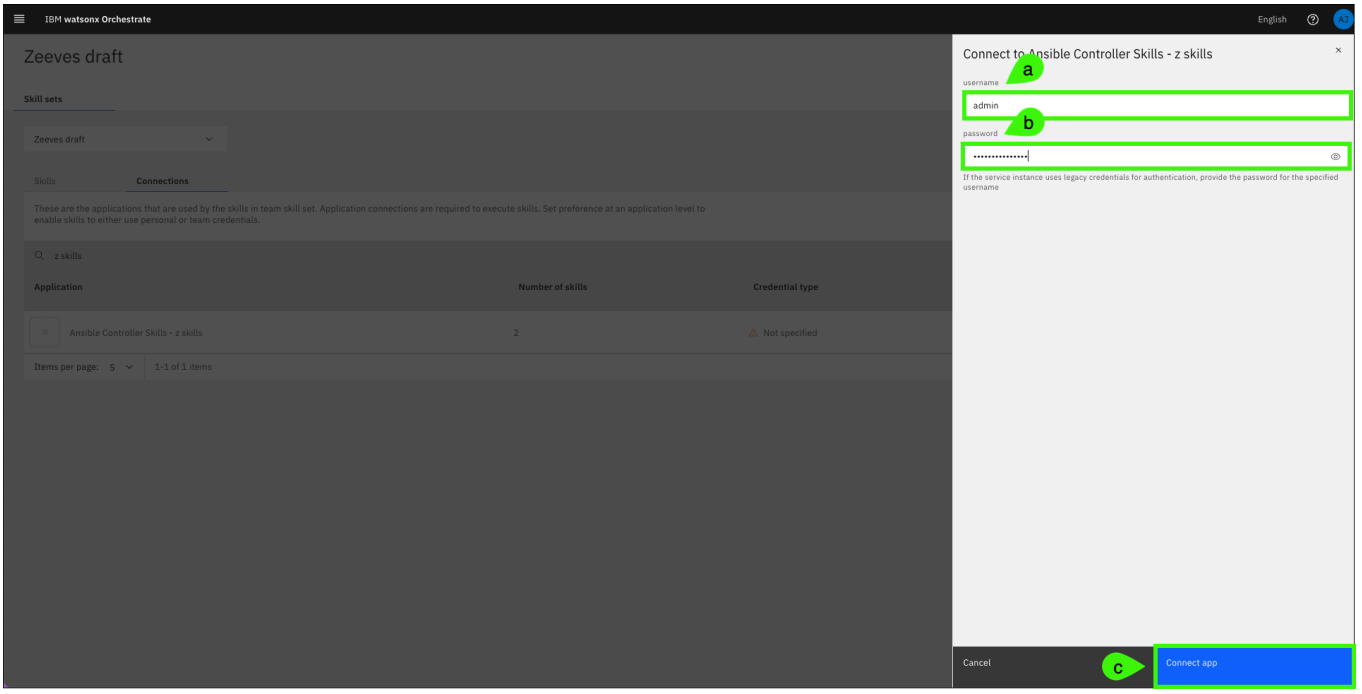
6. Click the ellipses ( ) and click **Connect app**.



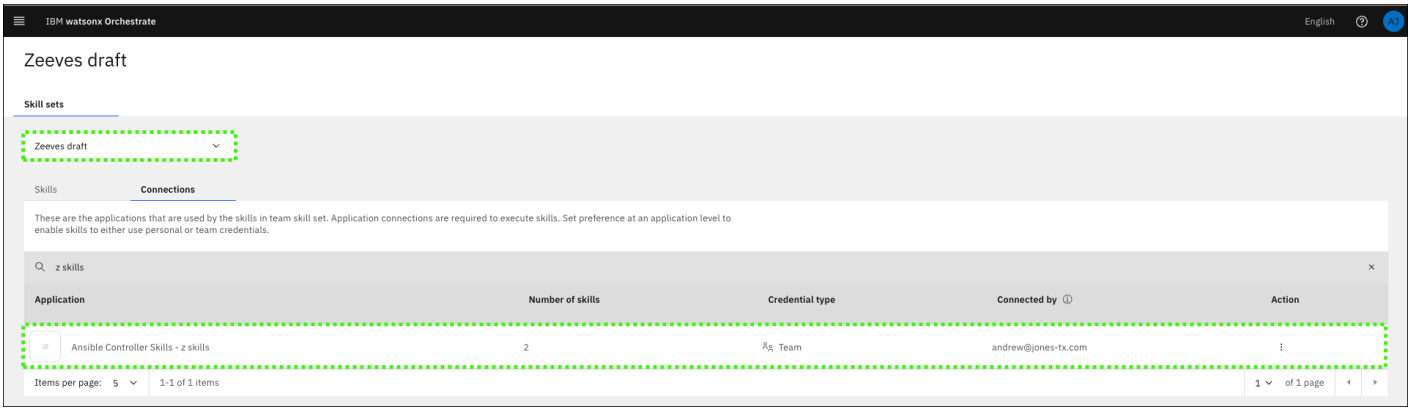
7. On the **Connect to Ansible Controller Skills** form, keep the defaults and click **Connect app**.



8. Enter the **username** (**a**) and **password** (**b**) using the username (`admin`) and password for your IBM Technology Zone (ITZ) *watsonx Assistant for Z Pilot - AAP & z/OS* reservation, and then click **Connect app** (**c**).



The application is now connected to the draft version of your assistant.



Continue to the next section to create actions for your assistant.

Creating actions for your assistant

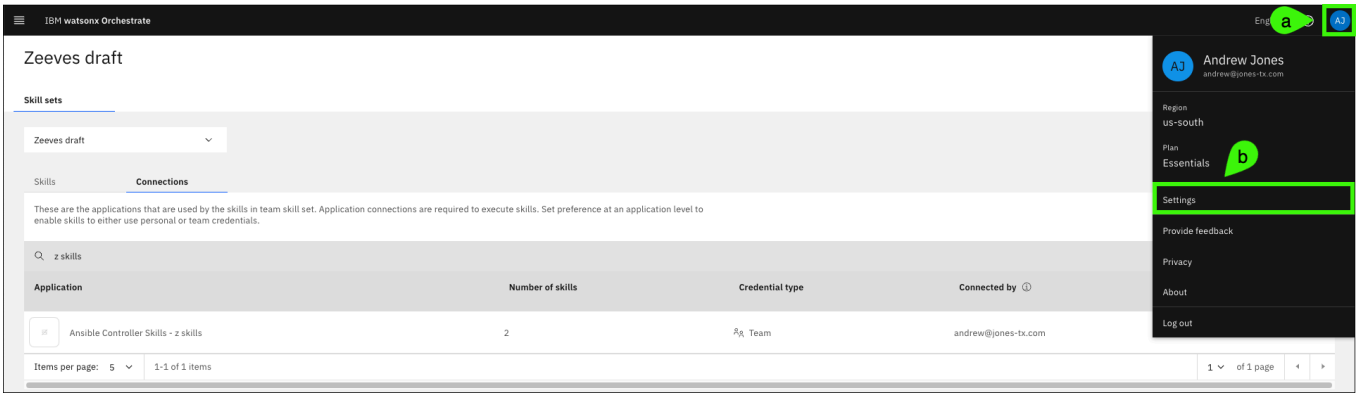
Once the skills in your application are connected to your assistant, you're ready to begin creating actions tied to those skills. Learn more about building actions here

Configure the number of input fields

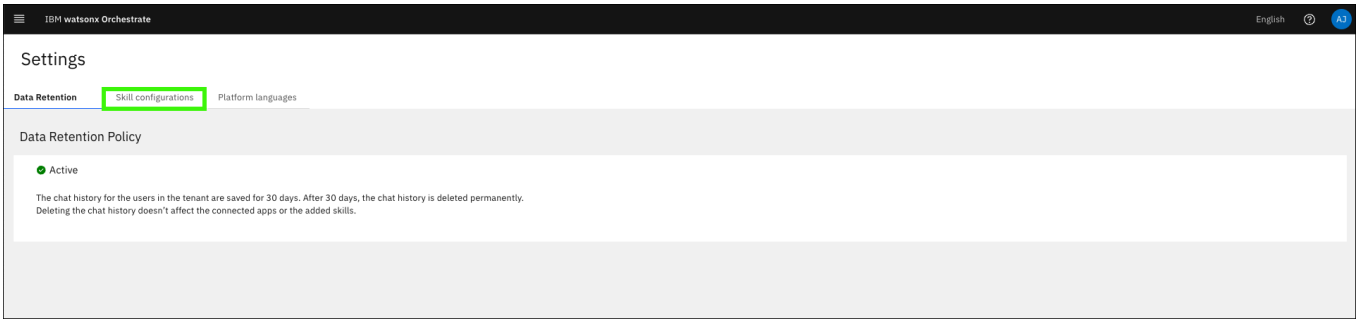
Before configuring actions, it's important to modify a setting within watsonx Orchestrate that allows triggered skills to display as forms (versus conversational skills).

1. Click your (a) profile icon and then click (b) **Settings**

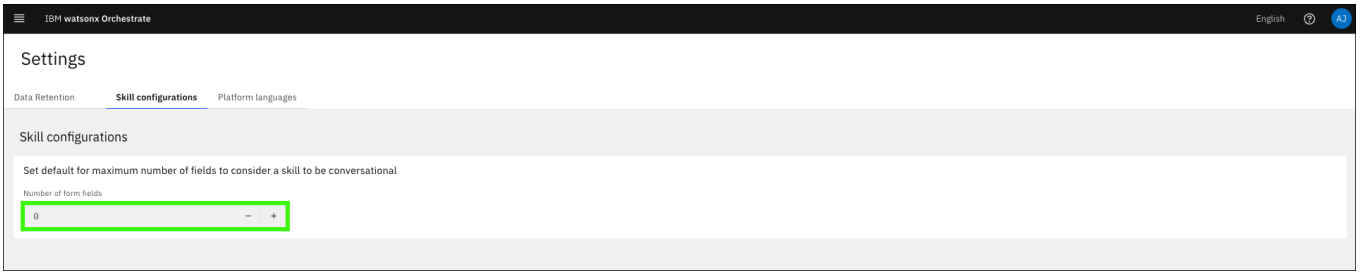
   Learn more about configuring input fields here.



2. Click the **Skill configurations** tab.

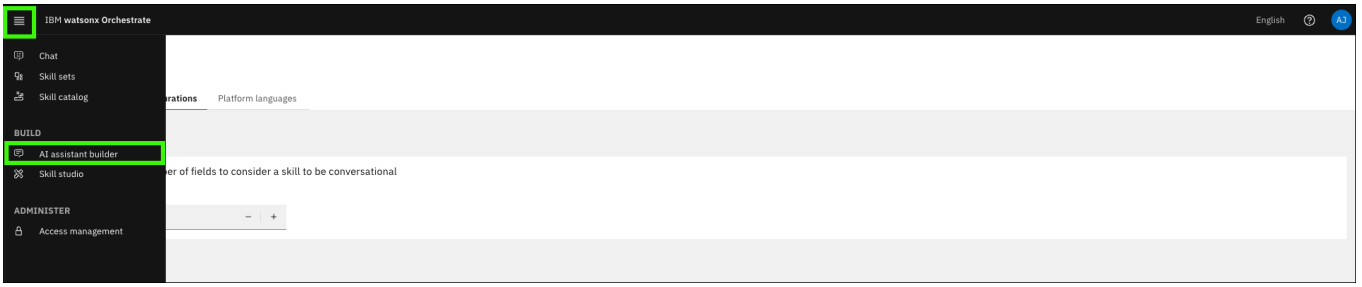


3. Enter **0** for the **Number of form fields**.

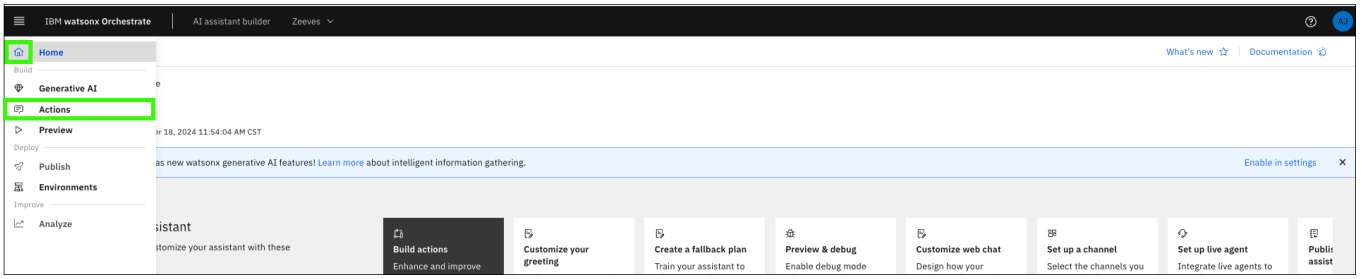


Create actions

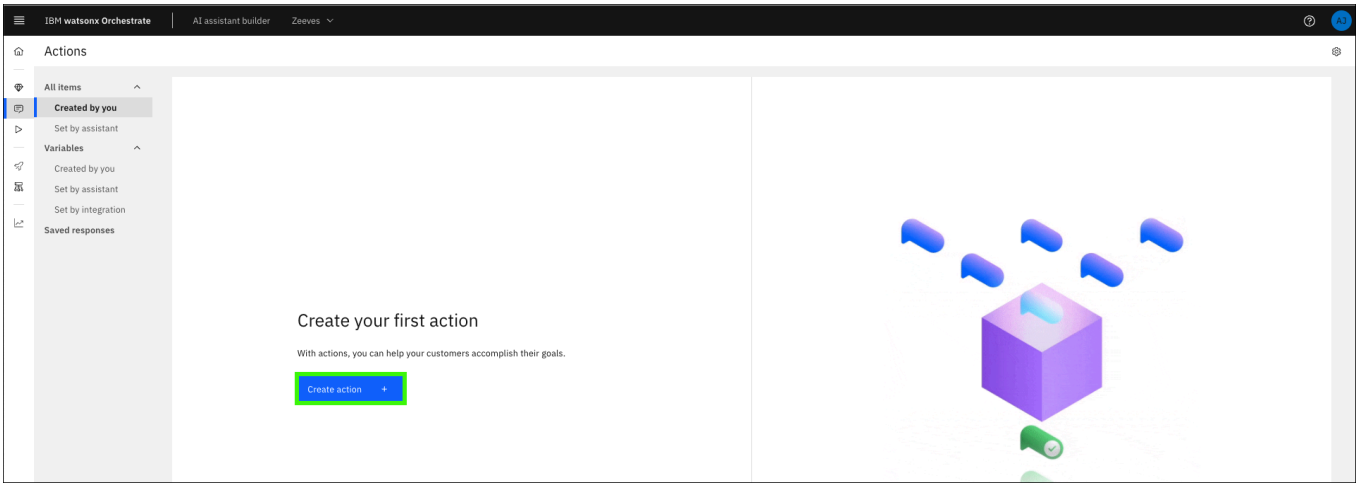
1. Click the main menu and select **AI assistant builder**.



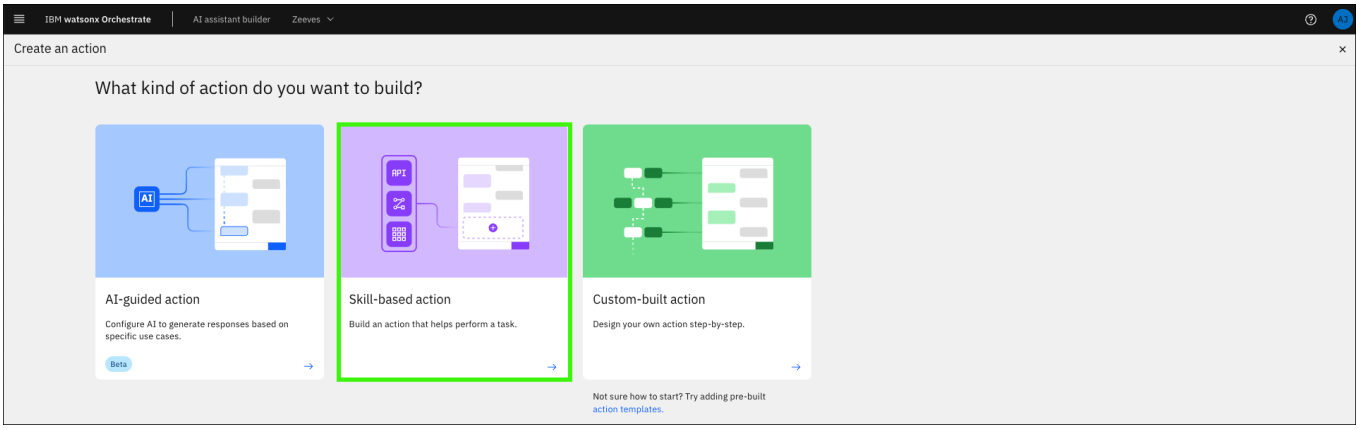
2. Hover over the **Home** icon ( ) and click **Actions**.



3. Click **Create action**.

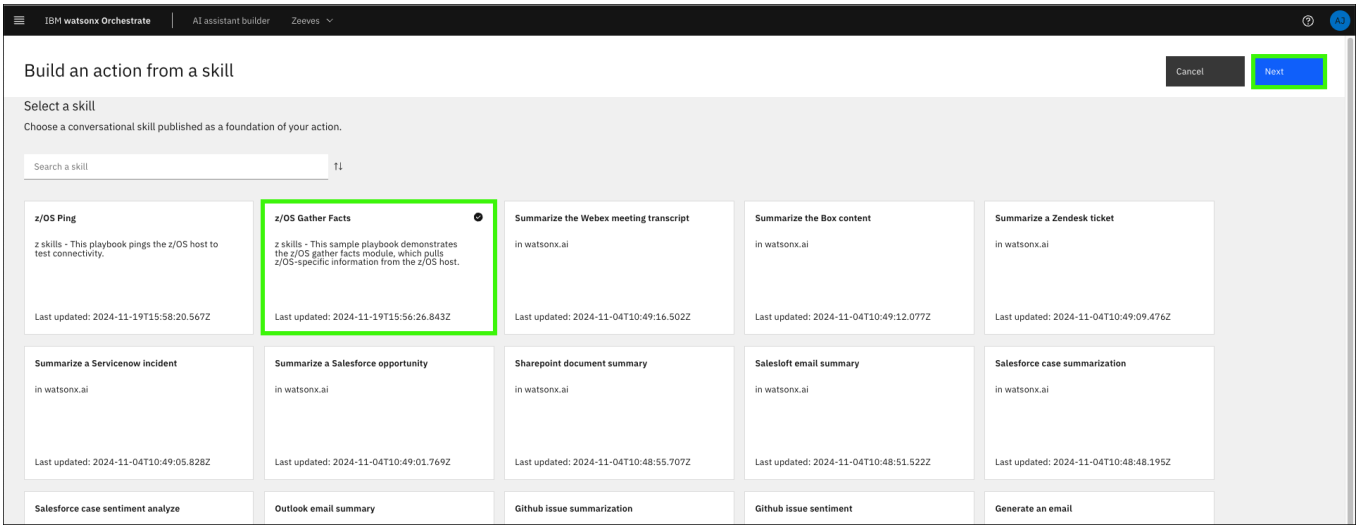


4. Click the **Skill-based action** tile.



5. Select the **z/OS Gather Facts** tile and click **Next**.

Note, it may take a minute for the page to display the action tiles. The date shown in the **z/OS Gather Facts** tile reflects when you added the skill to your application.



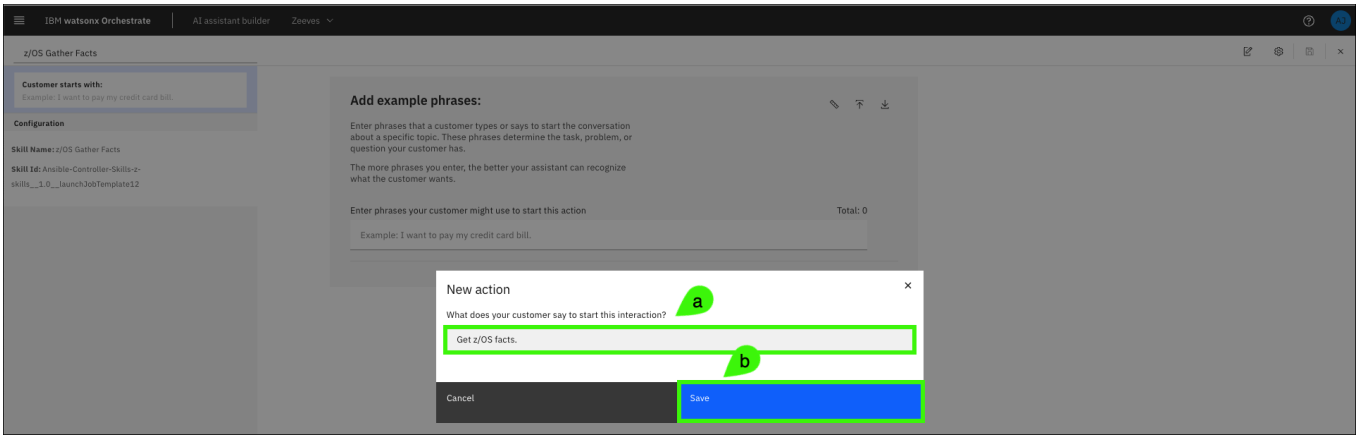
6. On the **New action** dialog, (a) enter a prompt a user of the assistant might use to initiate the action and then (b) click **Save**.

> **Be careful with the sample phrases you specify.**
>
> During the development of the lab guide, it was discovered that some sample phrases with a `/` character can cause issues with the actions. Avoid using **z/OS** in the your sample phrases. This issue has been reported to the offering team.

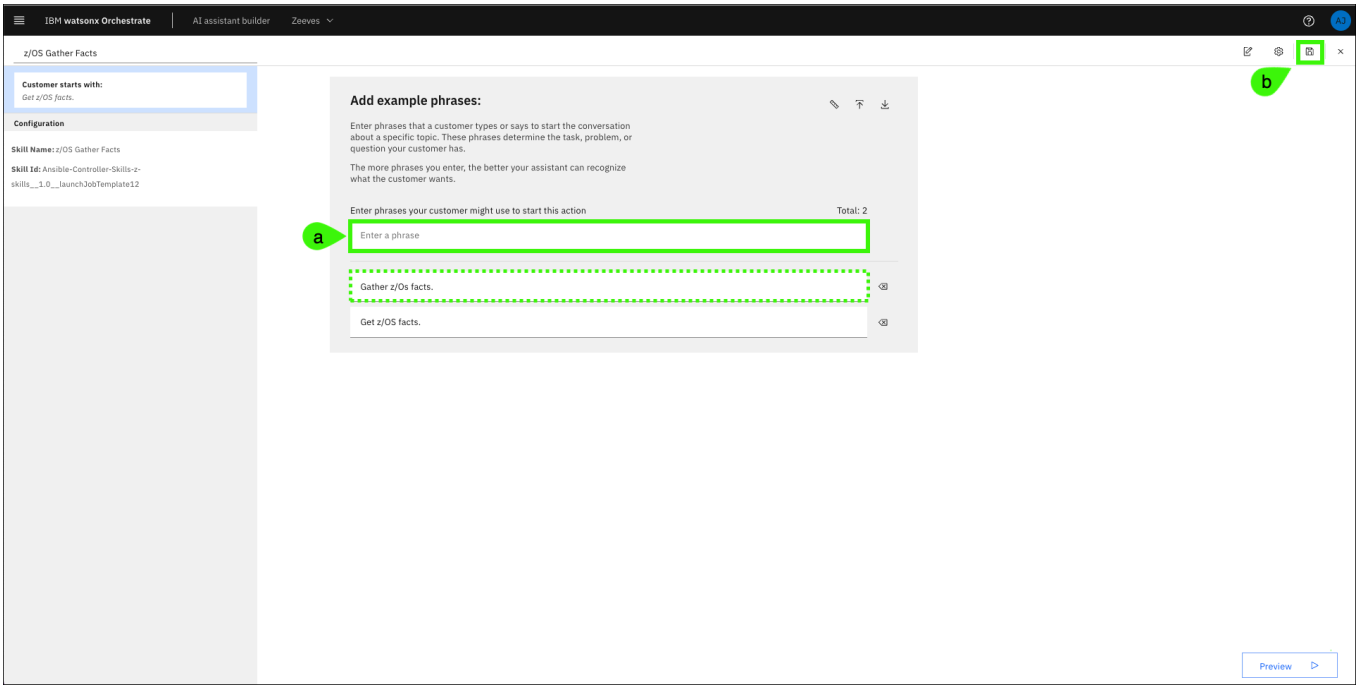
Sample prompts:

```
Get zOS facts
```

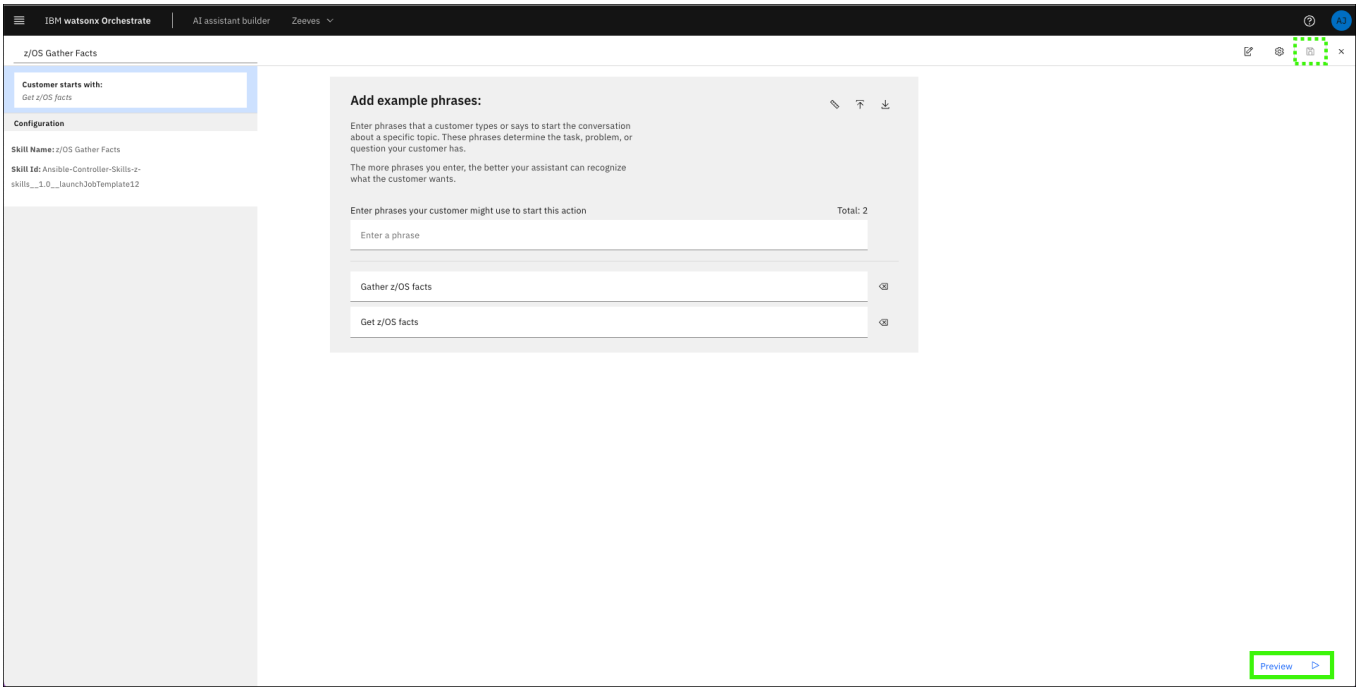
```
Gather zOS facts
```



7. Add any (a) additional prompts and then (b) click the save ( ).



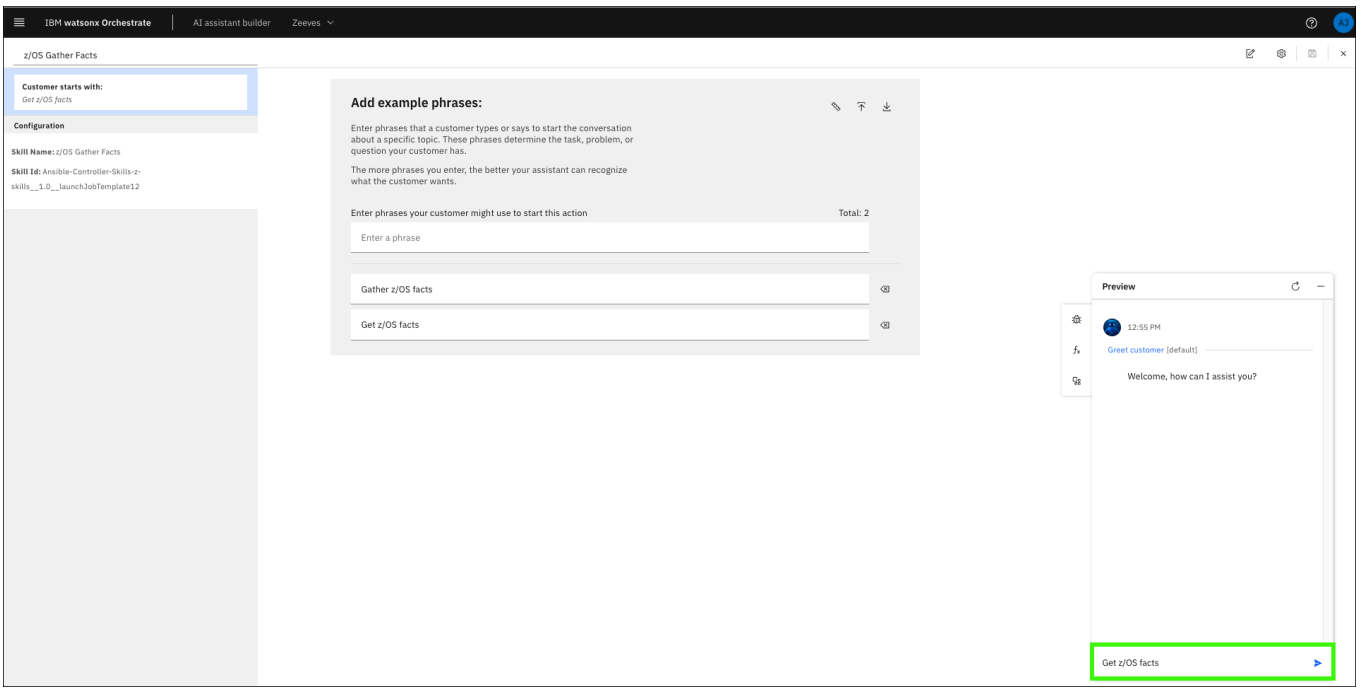
8. Click **Preview**.



9. Enter one of the prompts you specified in step 9 or 10.

   Prompt:

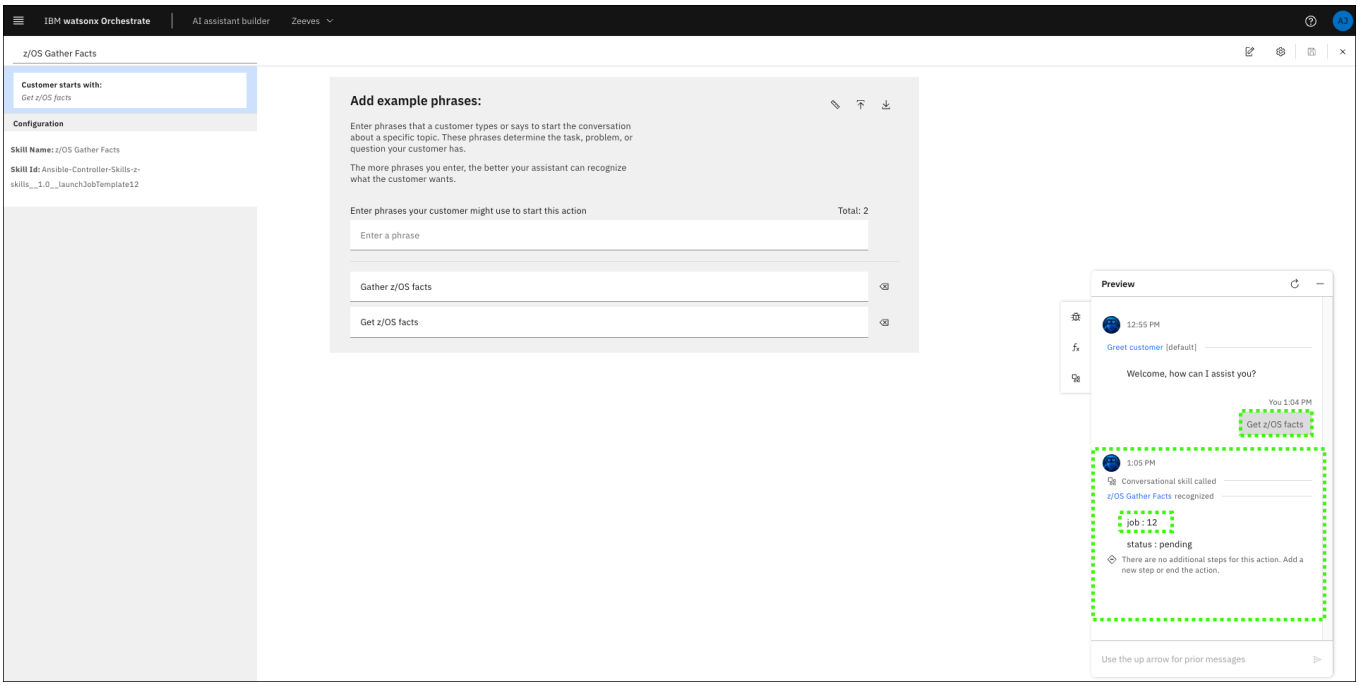
```
Get zOS facts
```



10. Review the returned results and record the **job** number.

    In the execution of this skill-based action, the skill executed properly and the output is the job id.

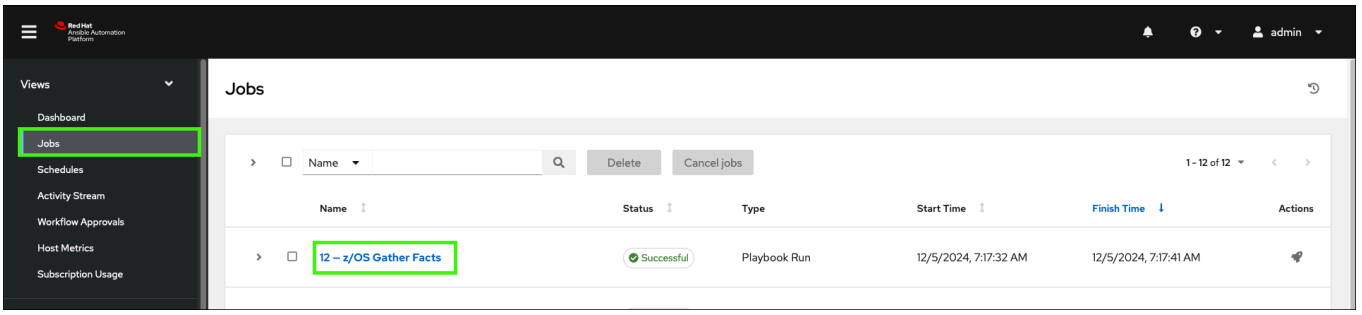
> ⚠ **If an error is generated or the action is not performed and only search results are returned, review the Troubleshooting section below.**



Verify the job in the Ansible Automation Platform console

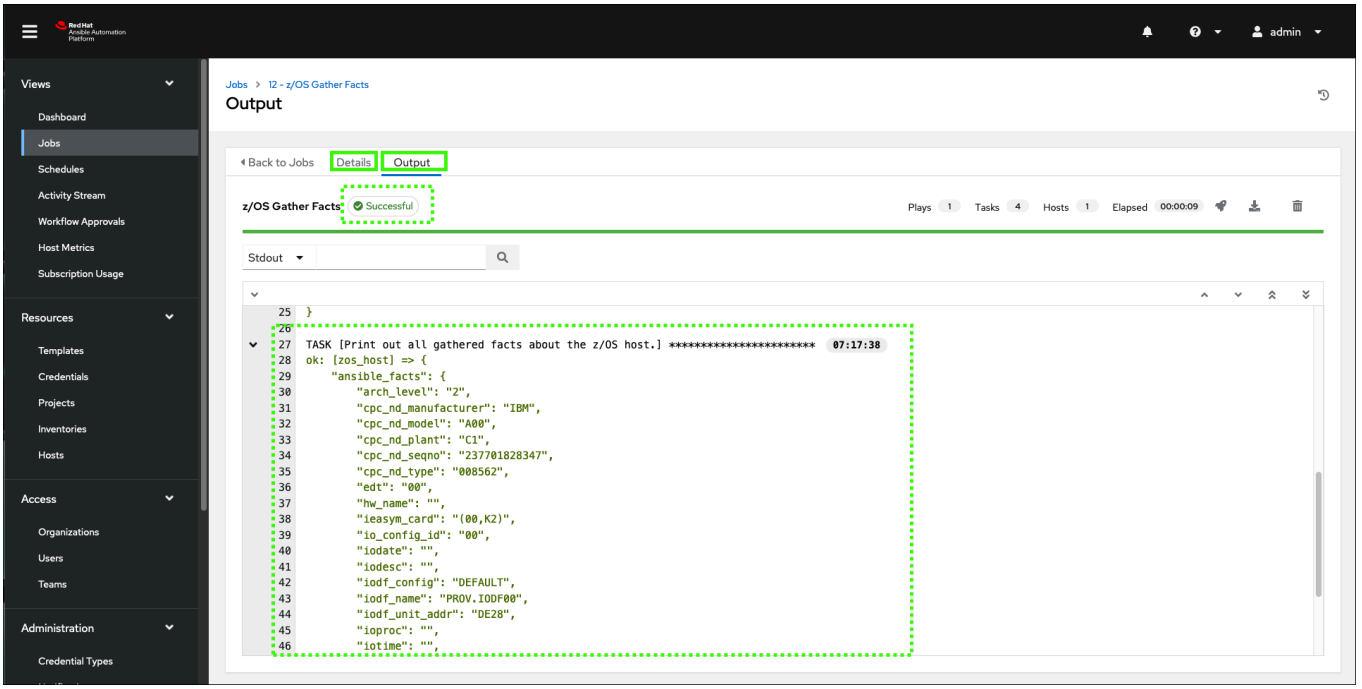
Return to the Ansible Automation Platform (AAP) console and review the job information.

1. Click **Jobs** and then click the **job** number recorded in the previous step for the **z/OS Gather Facts** skill.



2. Review both the **Details** and **Output** for the **z/OS Gather Facts** job.

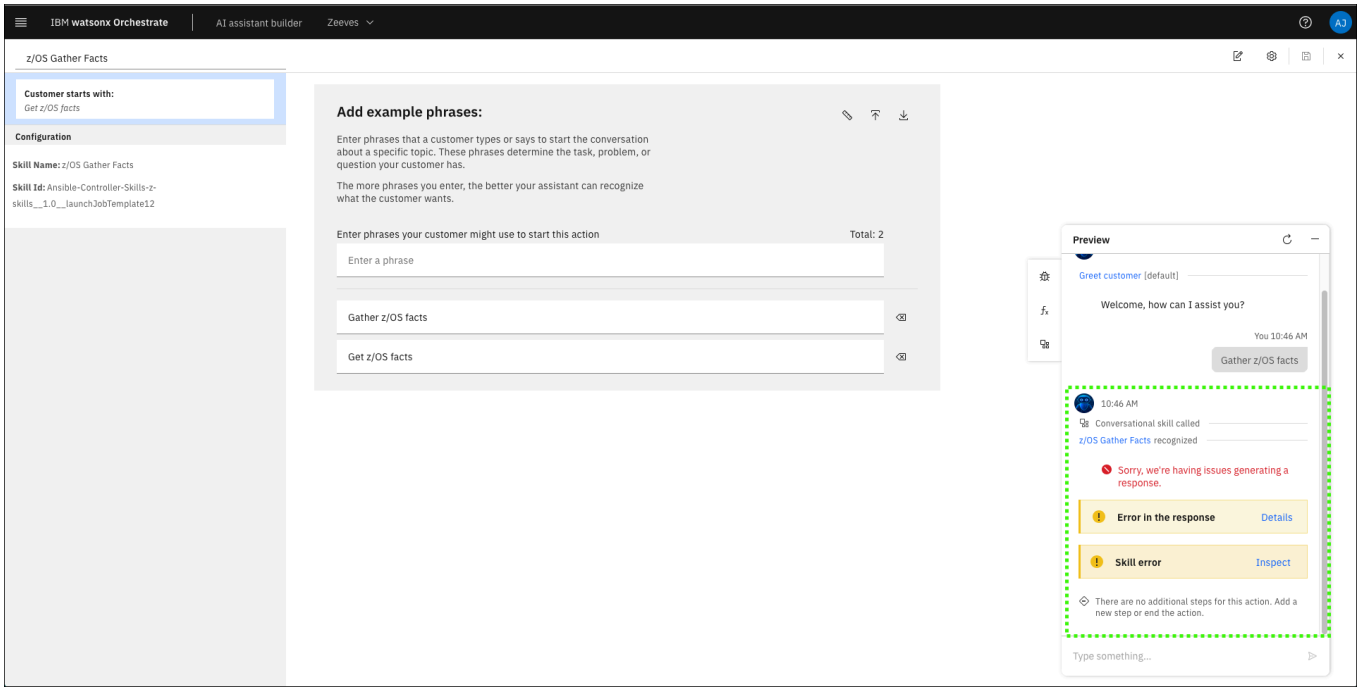
   Recall, that in the assistant, the contents shown in the **Output** of the Ansible job were not displayed.



IBM watsonx Assistant for Z provides utility skills to retrieve the job output. It is also possible to create a skill flow that executes the **z/OS Gather Facts** skill followed by the **Retrieve job output** utility skill in sequence; passing the job id from the first skill to the second, in order to view the output within the assistant. Creating a skill flow is covered in the next section.

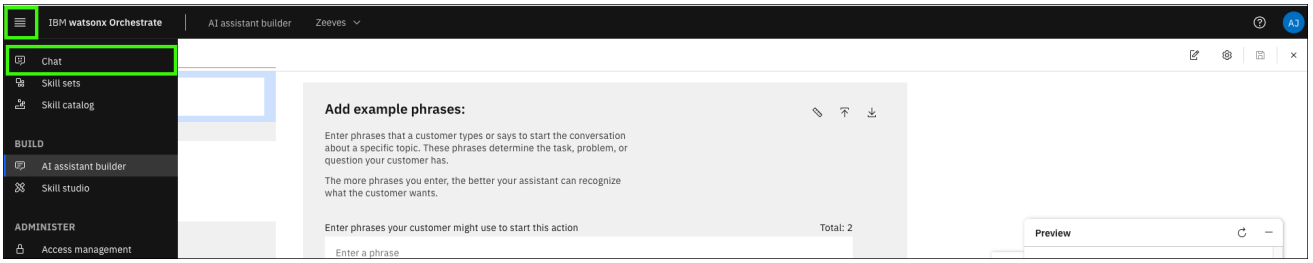
Troubleshooting

Skill returns "*Sorry, we're having issues generating a response*" or the action is not performed and only search results are returned.

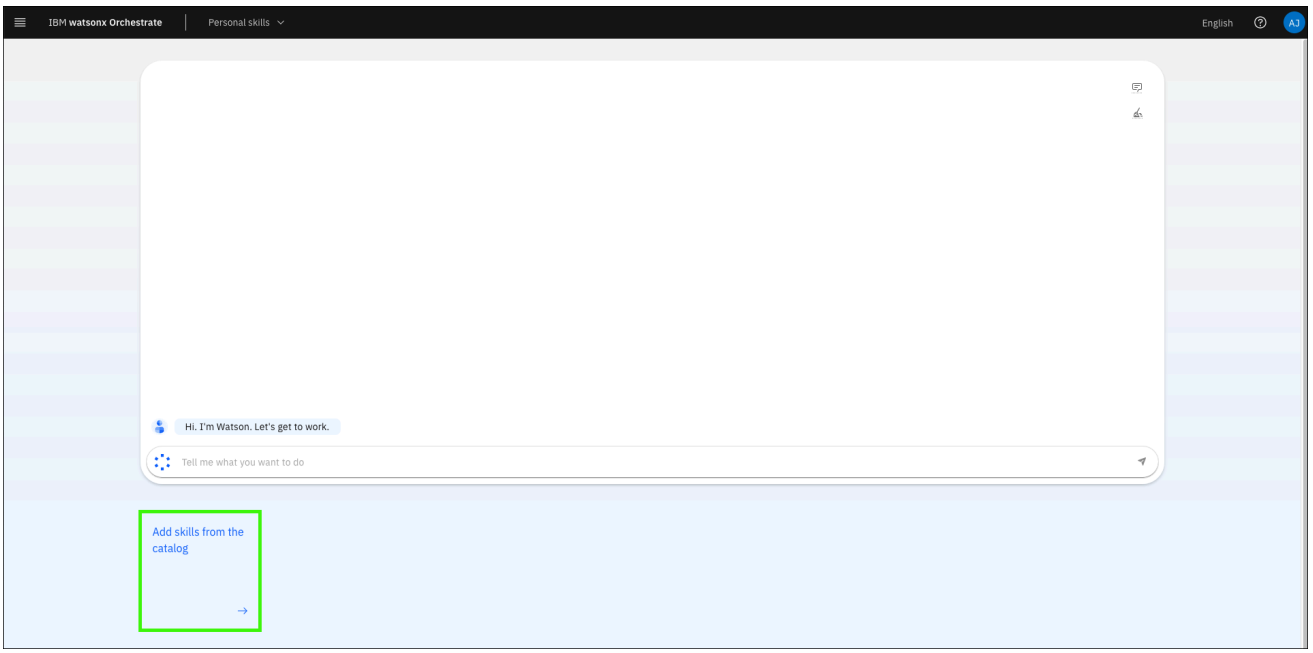


This error appears to be an intermittent issue when a skill is first added. To resolve, add the skill to your personal skills catalog using the steps that follow. If you encounter the issue, try the steps that follow:

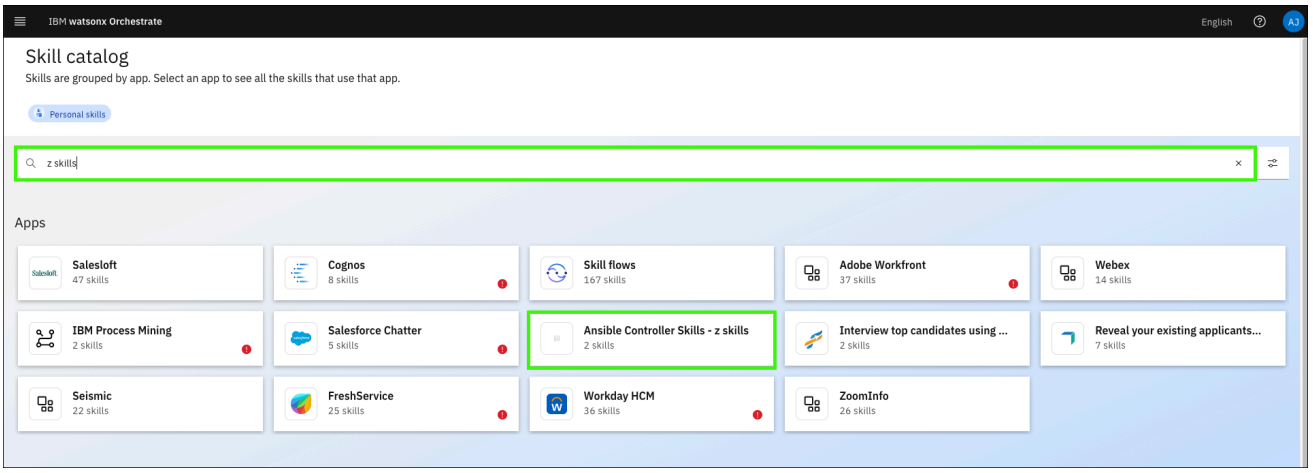
1. Expand the main menu and select **Chat**.



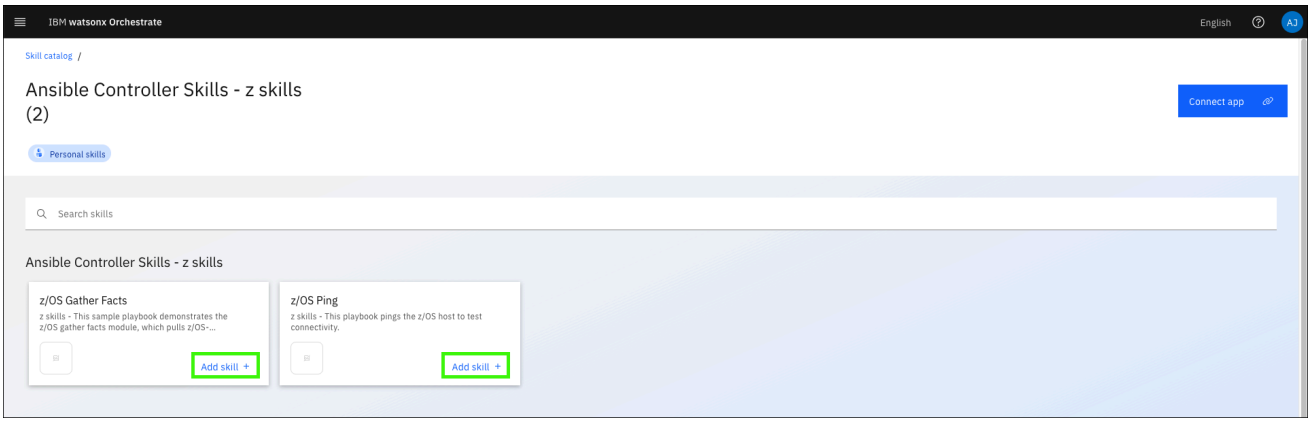
2. Click **Add skills from the catalog**.



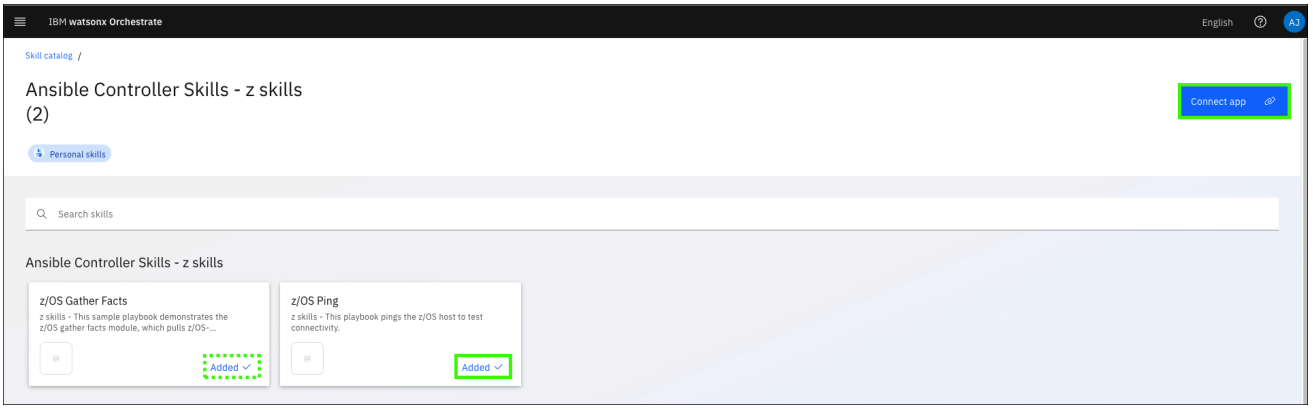
3. Search for the skill app you created earlier and click the tile for your app.



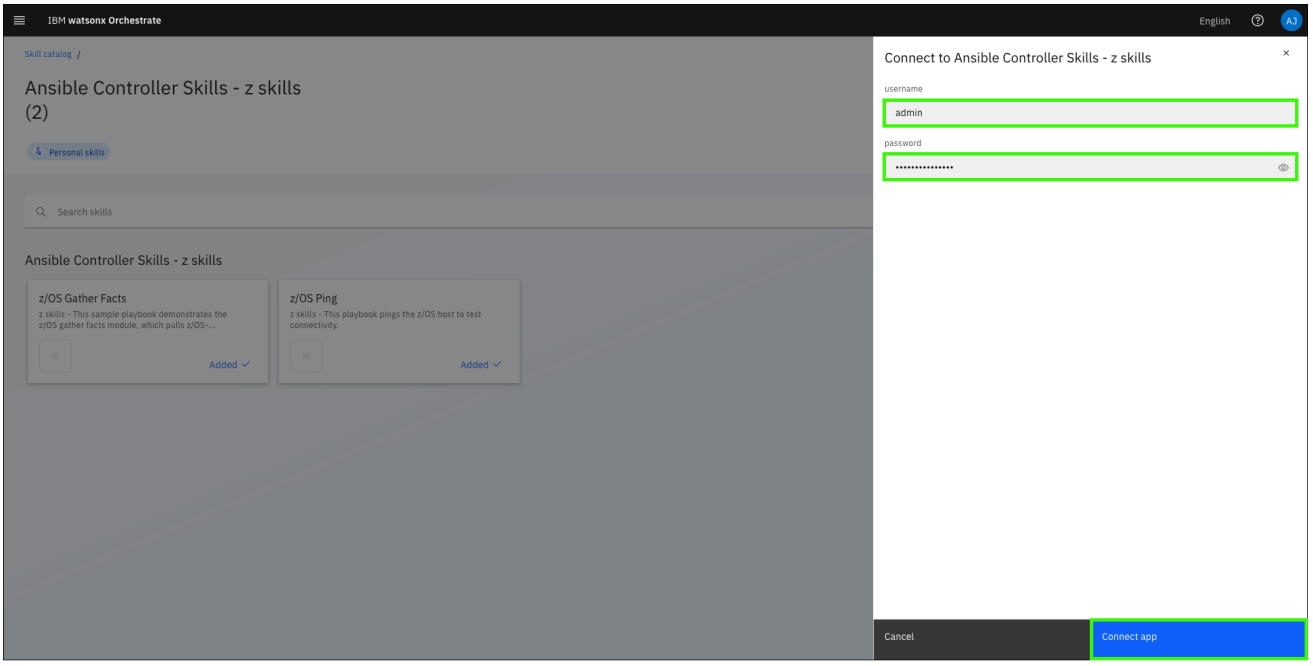
4. Click **Add skill** for all the skills you want to add.



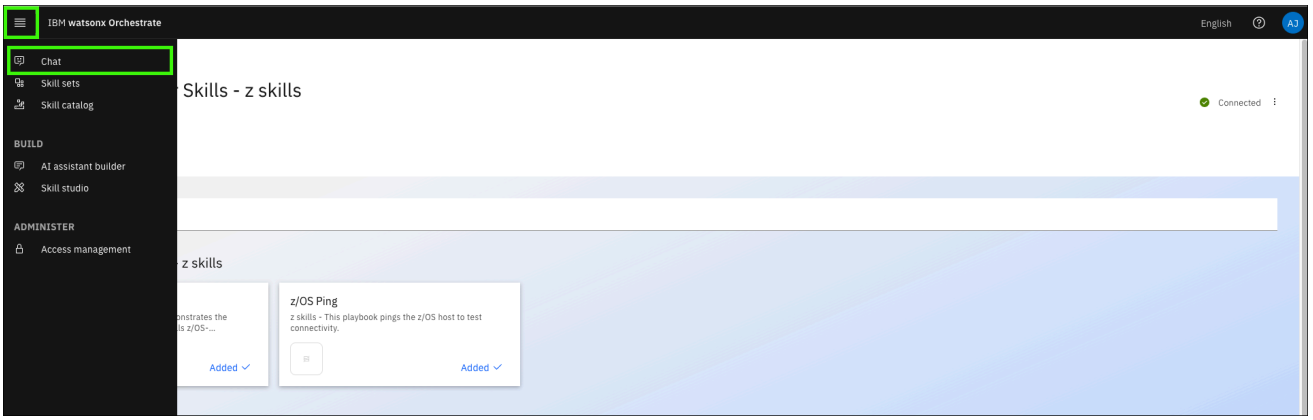
5. Click **Connect app**.



6. Enter the (a) **username** and (b) **password** using the username (admin) and password for your IBM Technology Zone (ITZ) watsonx Assistant for Z Pilot - AAP & z/OS reservation (AAP User Password (Use SSH key to login, only use password for UI)), and then click **Connect app**.



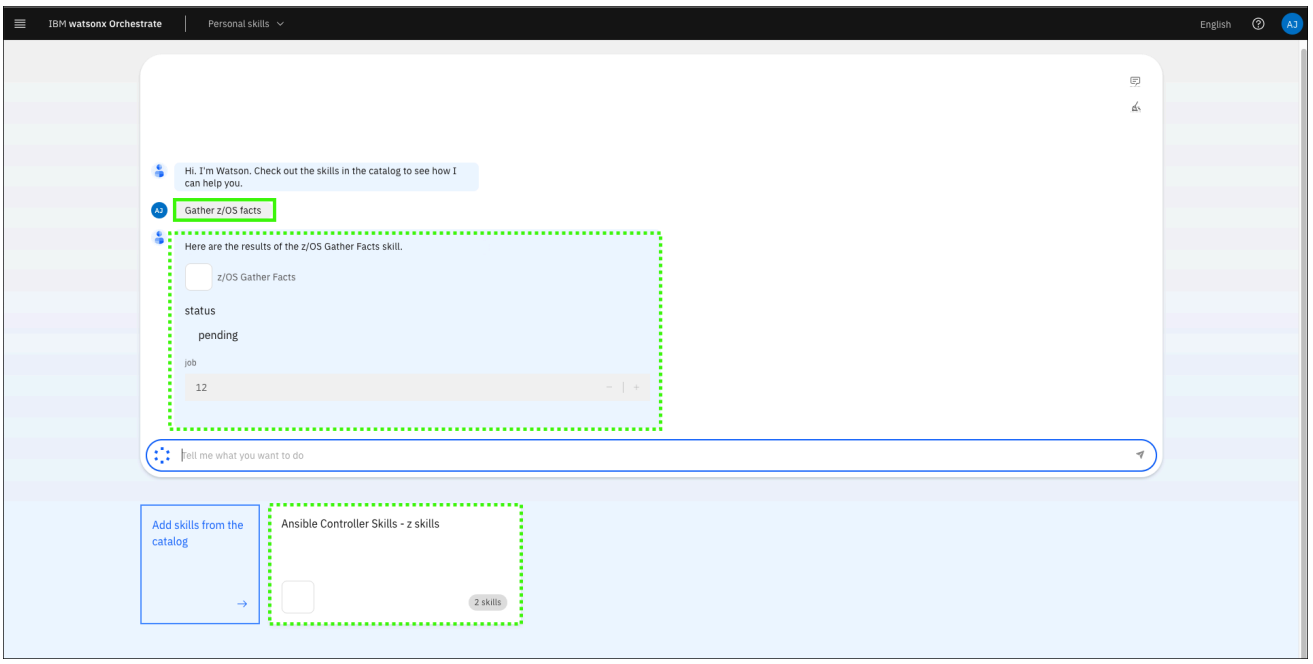
7. Expand the main menu and select **Chat**.



8. Try one of the prompts you created for your skill.

   Prompt:

```
Gather zOS facts
```



You should now be able to run the skill through the assistant preview.

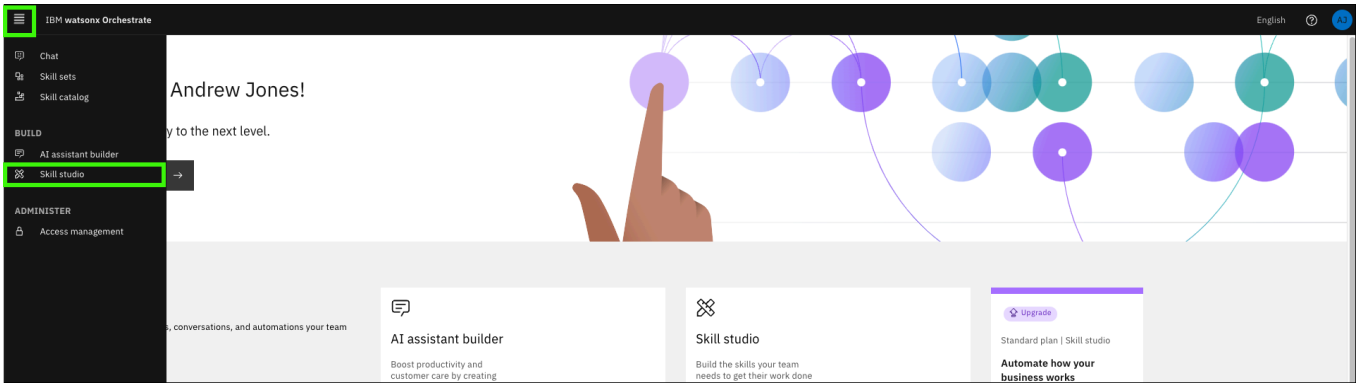
Creating skill flows

As seen in the previous section, running the Ansible skill to **Gather z/OS facts**, the skill executed successfully and was verified within the Ansible Automation Platform (AAP) console by viewing the job output. However, the output wasn't displayed by the assistant. To enable this scenario, a skill flow is needed. Skills are often more valuable when combined with other skills. You can create a skill flow to use two or more skills together to finish a task (like returning the output of a previous skill). When you create a skill flow, you map the output of one skill as the input for subsequent skills. Learn more about creating skill flows here.

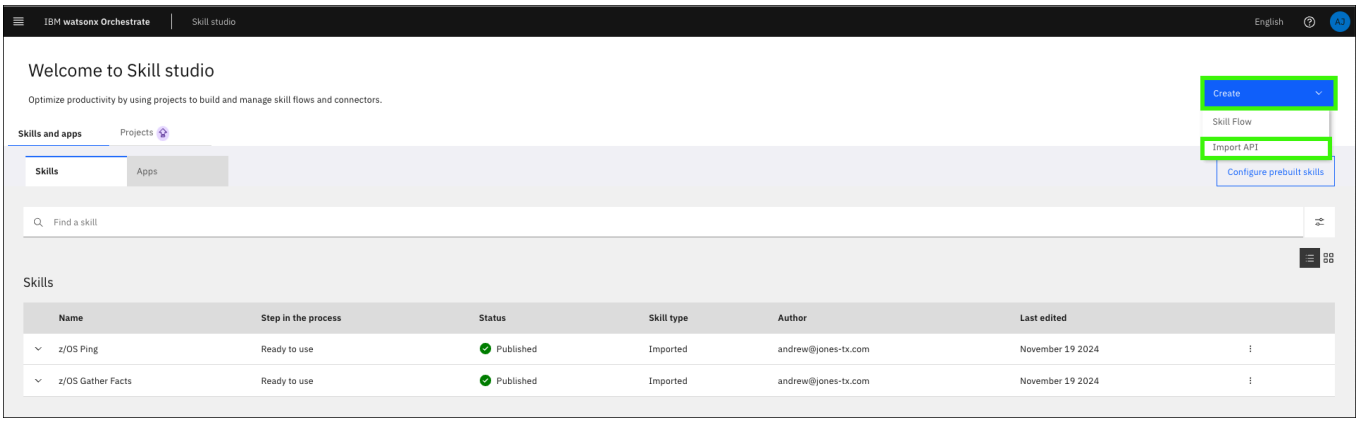
As mentioned in a previous section, there are some default utility skills that are provided out of the box with the "Z Skills Accelerator" which are leveraged to return the output of a skill. To accomplish this, we will import the Ansible Utility skill called **Retrieve job output**.

Add the utility skill

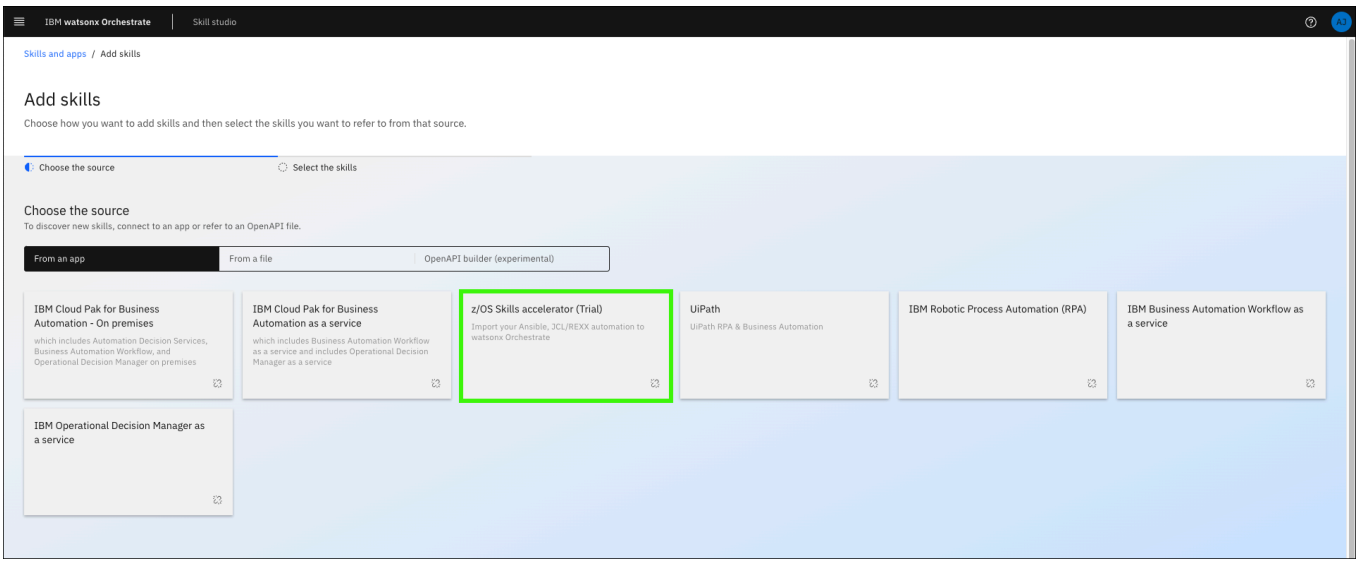
1. Open IBM watsonx Orchestrate **Skill studio**.



2. Expand **Create** and click **Import API**.



3. Click the **z/OS Skills accelerator (Trial)** tile.



4. Enter the following values in the **z/OS Skills accelerator** form and then click **Connect**.

   Use the **URL**, **User Name**, and **Password** values recorded in the Explore Ansible Automation Platform section earlier.

   a: Connection Type: `ansible`

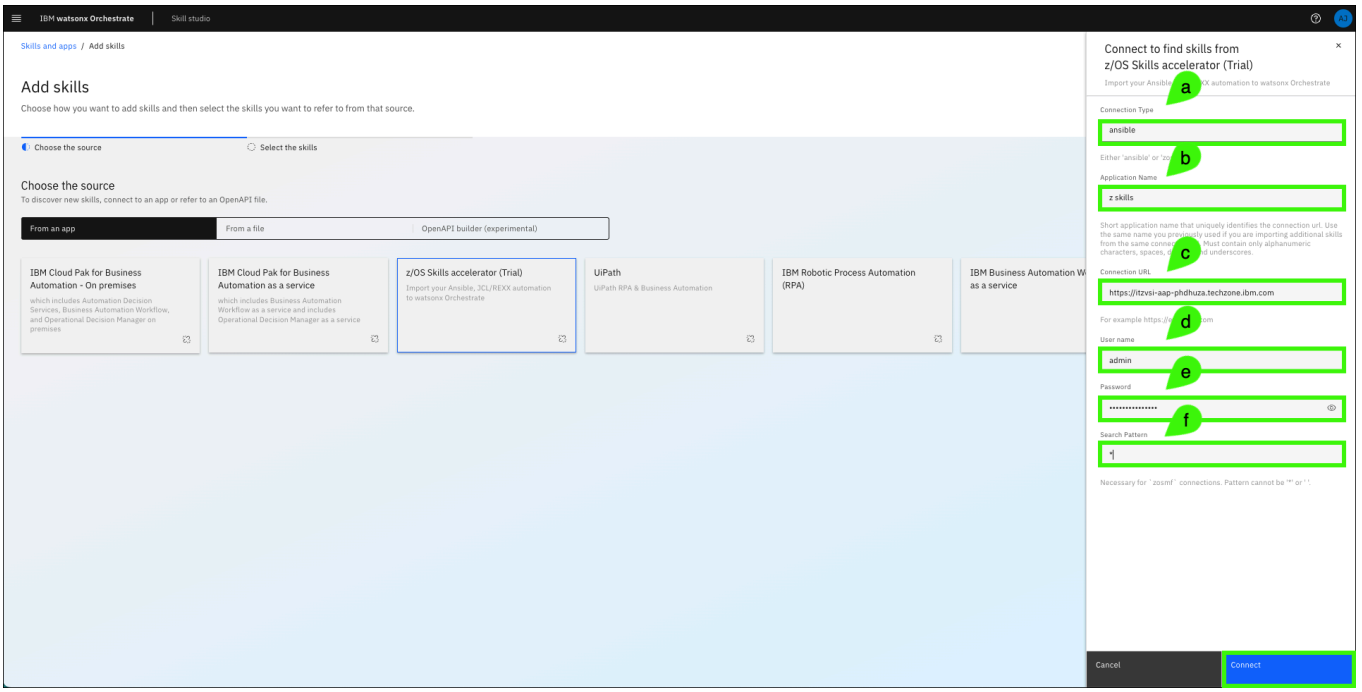
   b: Application Name: <use the same application name as in previous section>

   c: Connection URL: <enter the URL for your AAP UI>

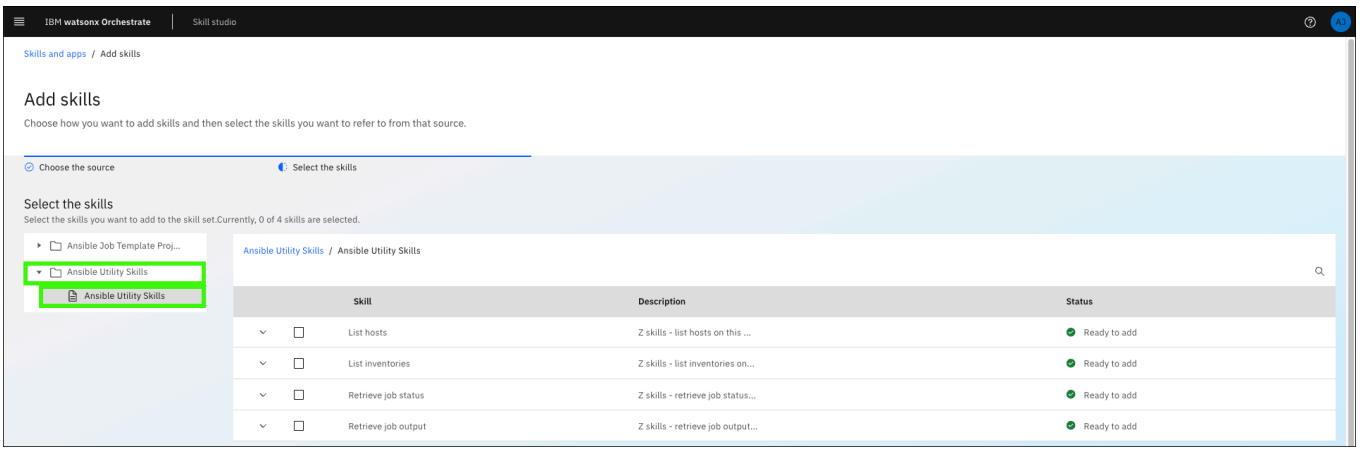
   d: User Name: <enter the AAP User Name (for UI access)>

   e: Password: <enter the AAP User Password>

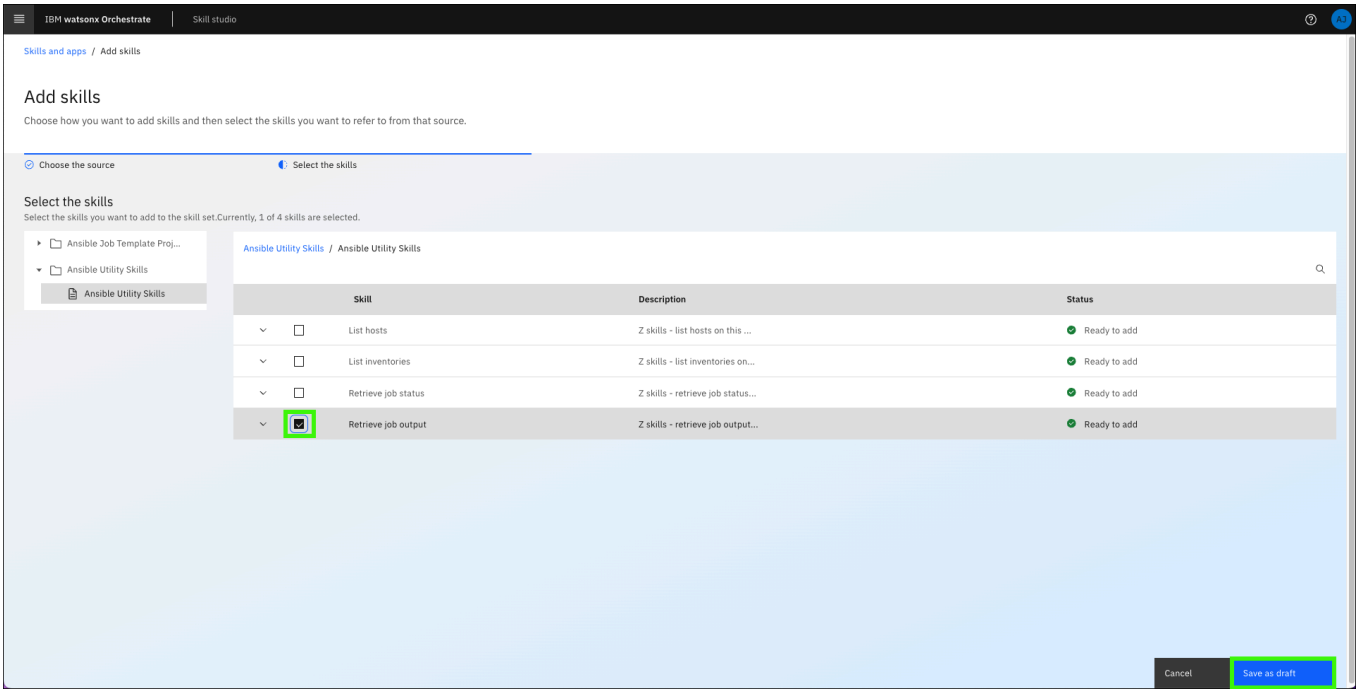
   f: Search Pattern: `*`



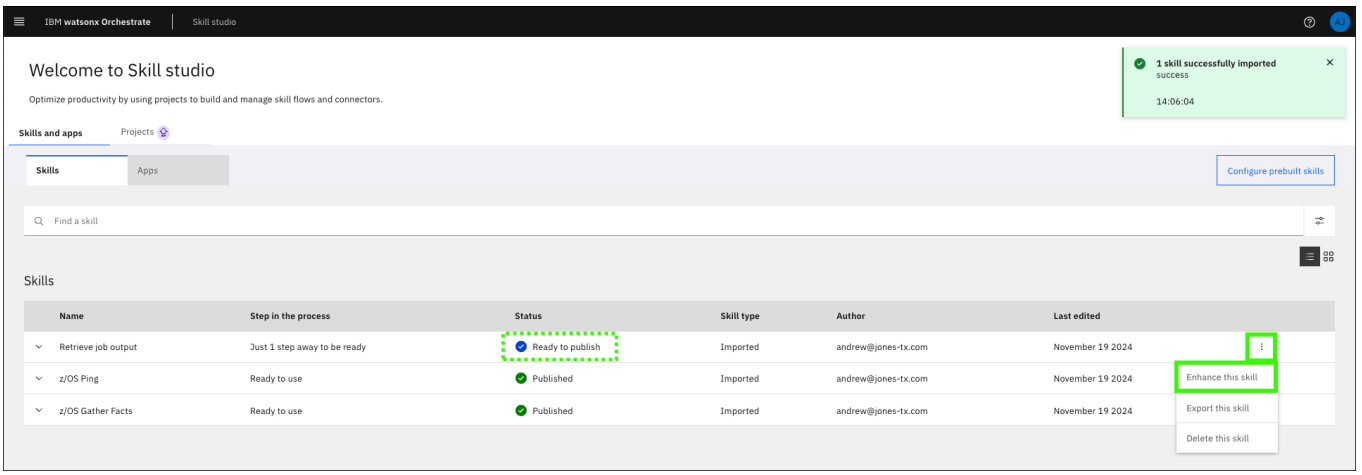
5. Expand **Ansible Utility Skills** and click **Ansible Utility Skills**.



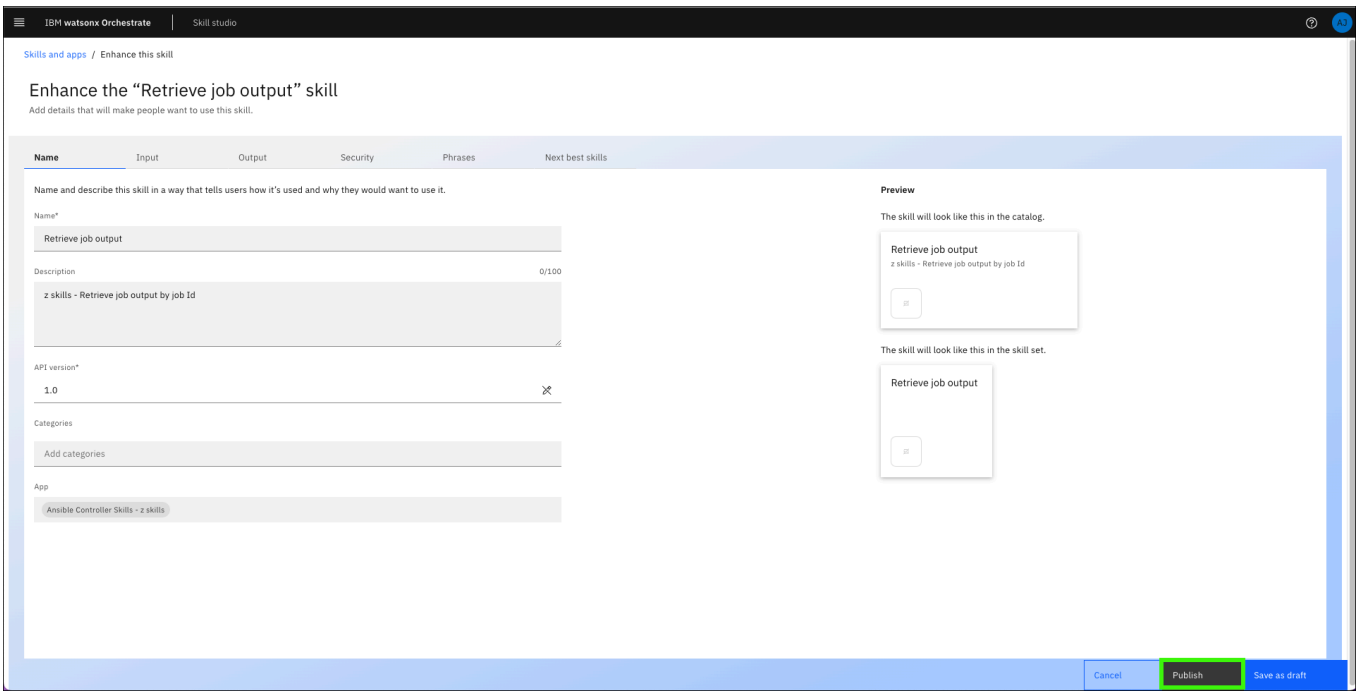
6. Select **Retrieve job output** and click **Save as draft**.



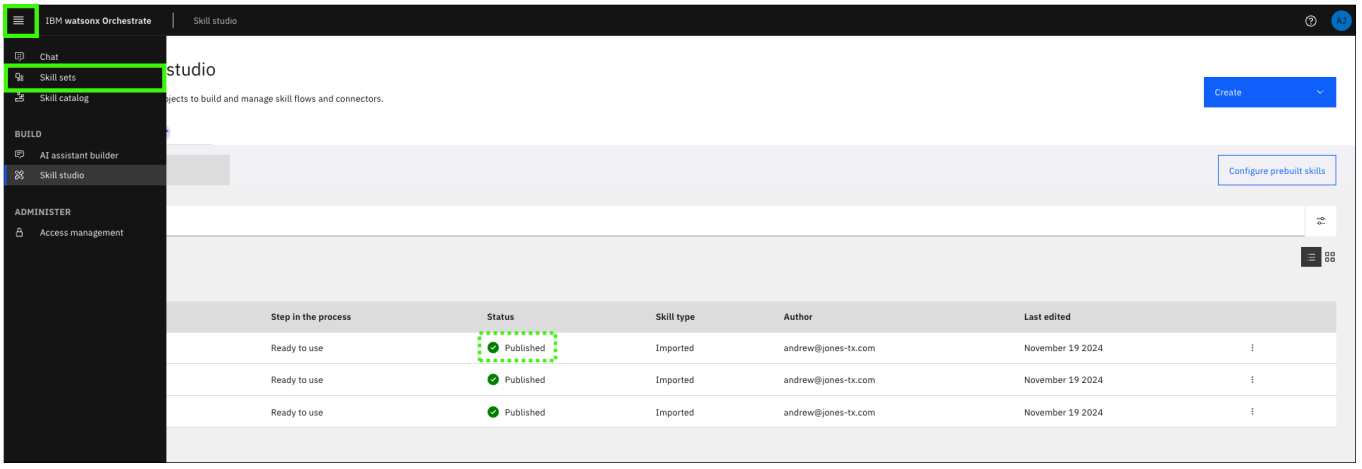
7. Click the ellipses (⋮) for the **Retrieve job output** skill and select **Enhance this skill**.



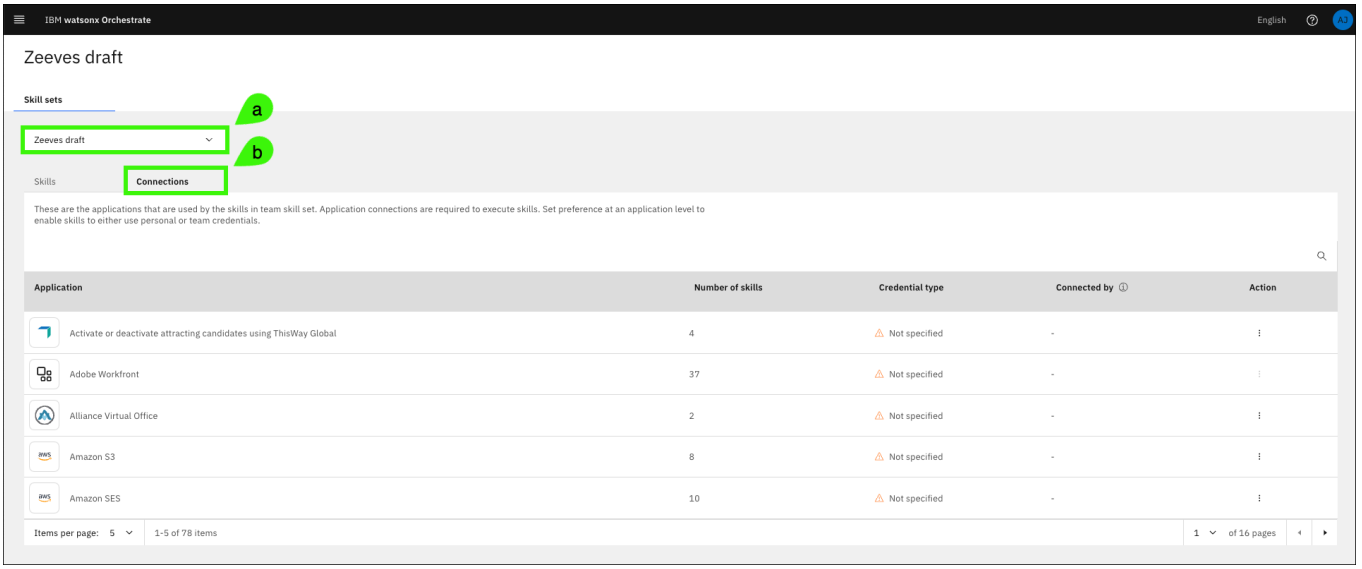
8. Review the skill settings and then click **Publish**.



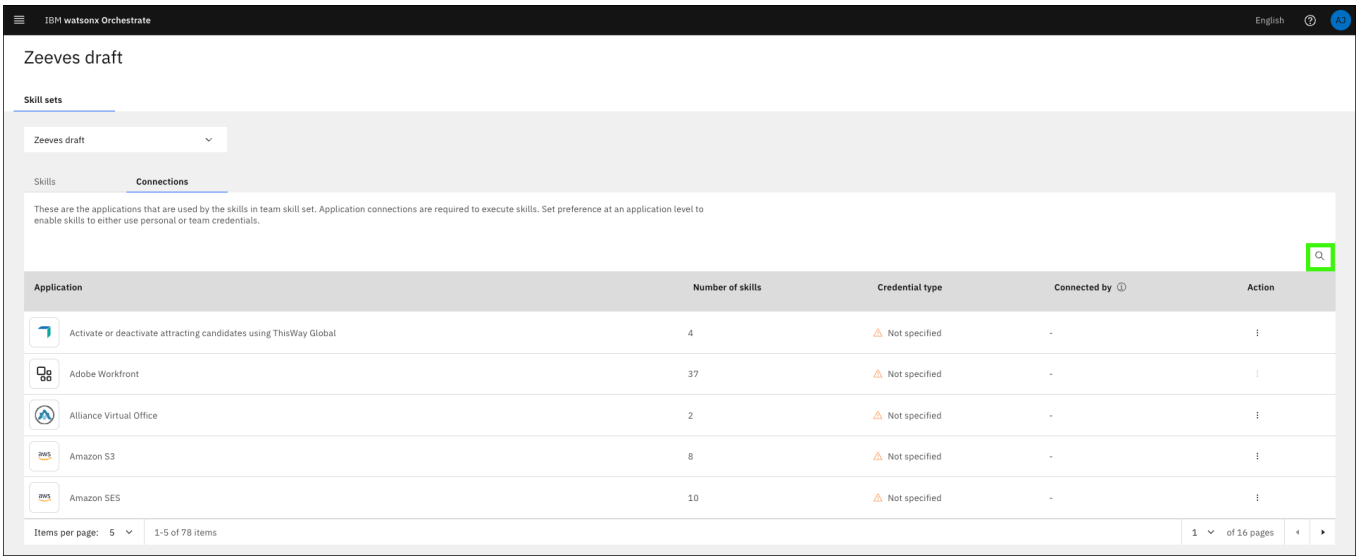
9. Select **Skill sets** from the main menu.



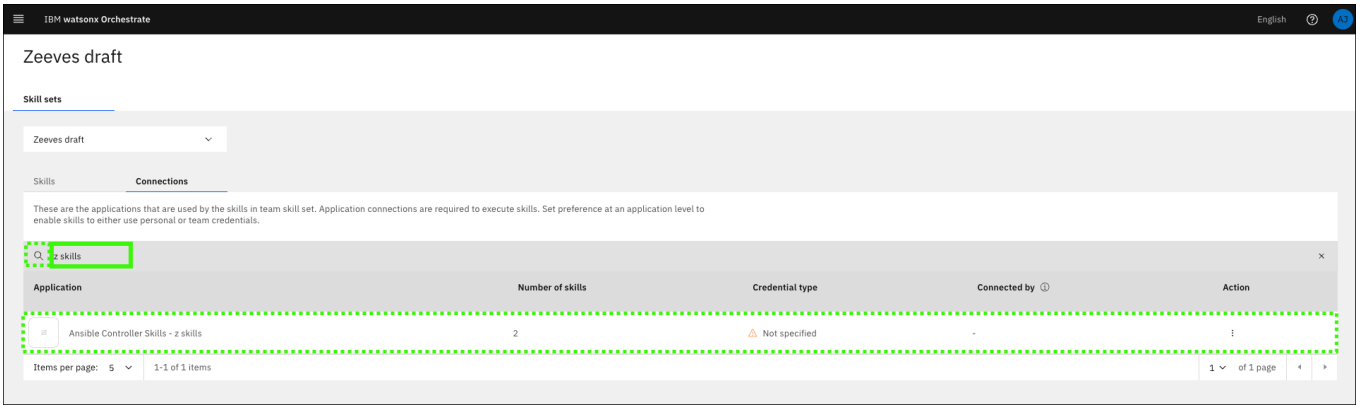
10. Select (a) your draft assistant in the **Team Skills** drop-down list and (b) click the **Connections** tab.



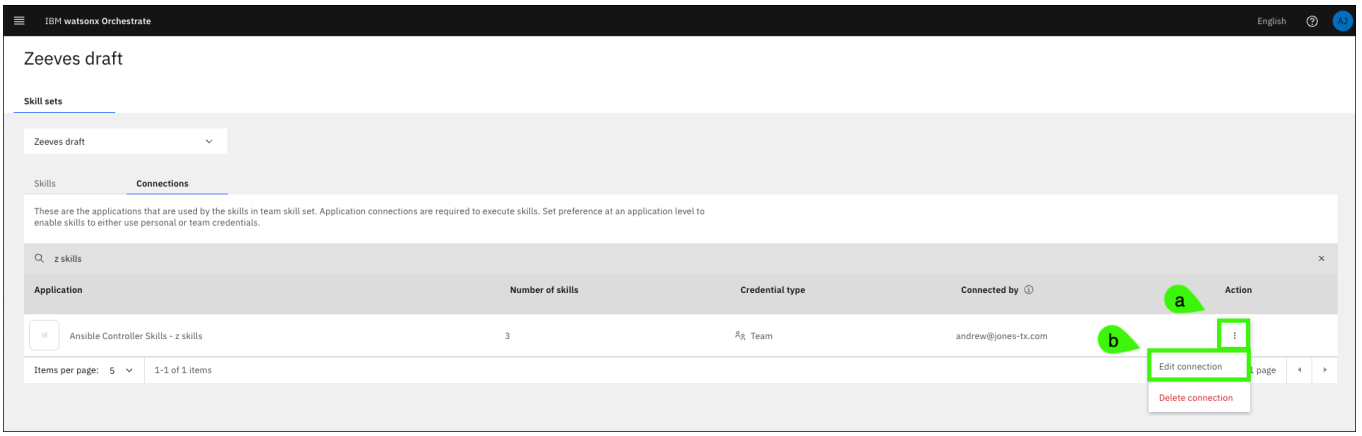
11. Click the **Search** (🔍) icon.



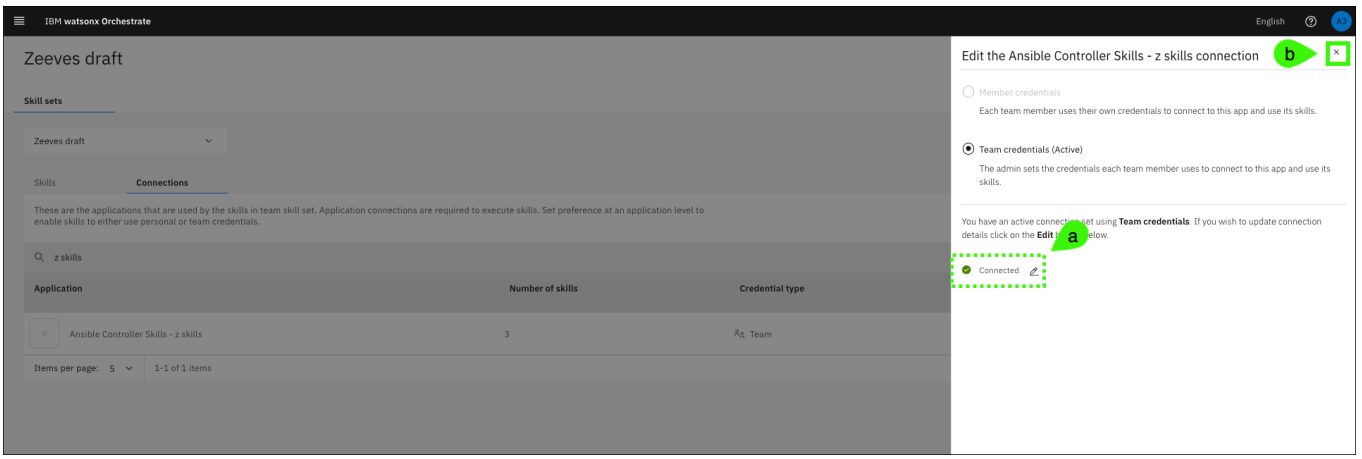
12. Search for the application name you specified earlier.



13. Click the (a) ellipses (⋮) for your application and (b) click **Edit connection**.

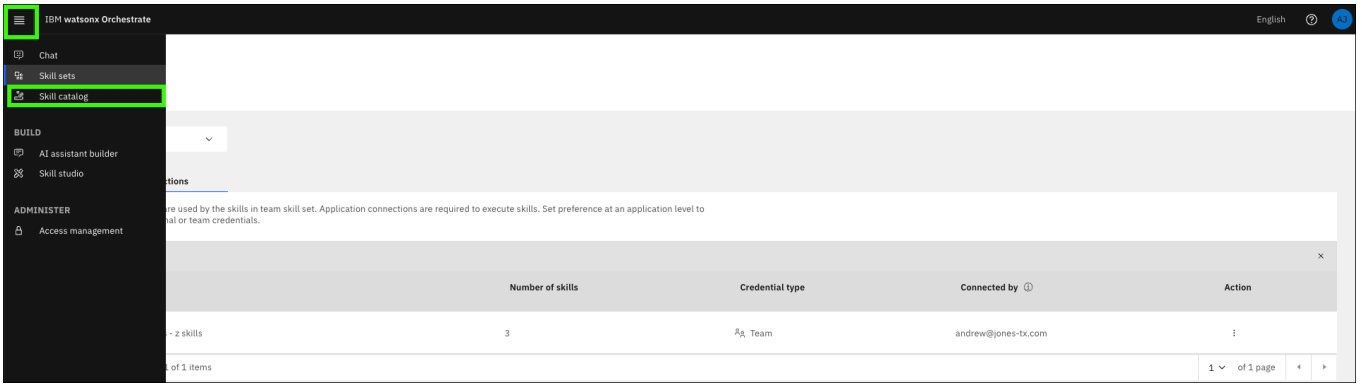


14. Verify the application is **Connected** (**a**) and then click **Close** (**b**).

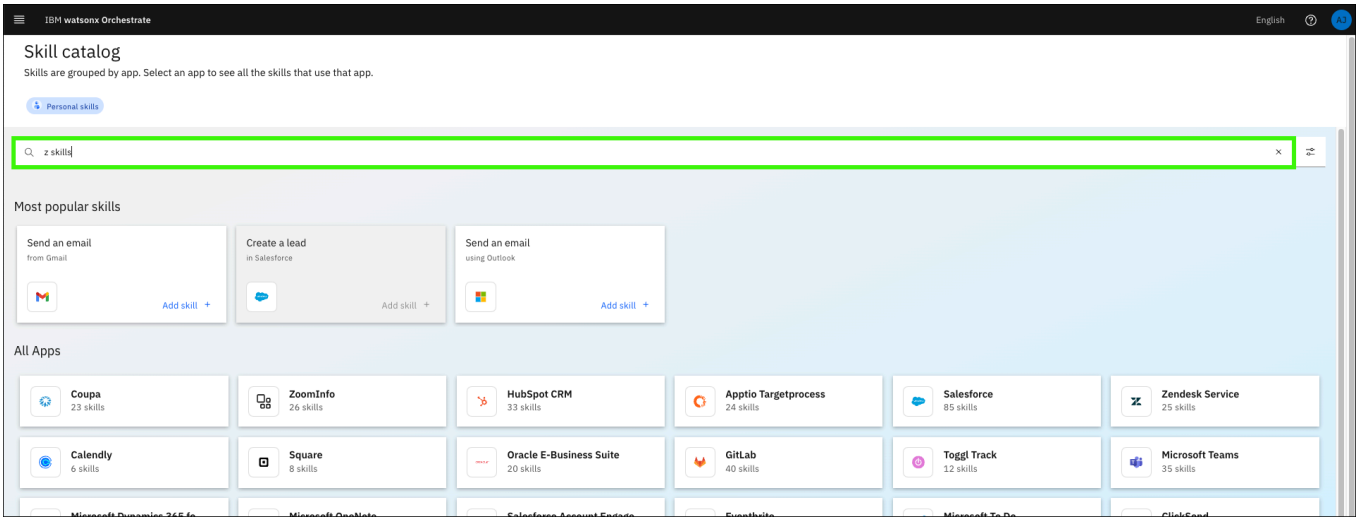


Add the skills to your Personal skills

1. Click **Skill catalog** in the main menu.

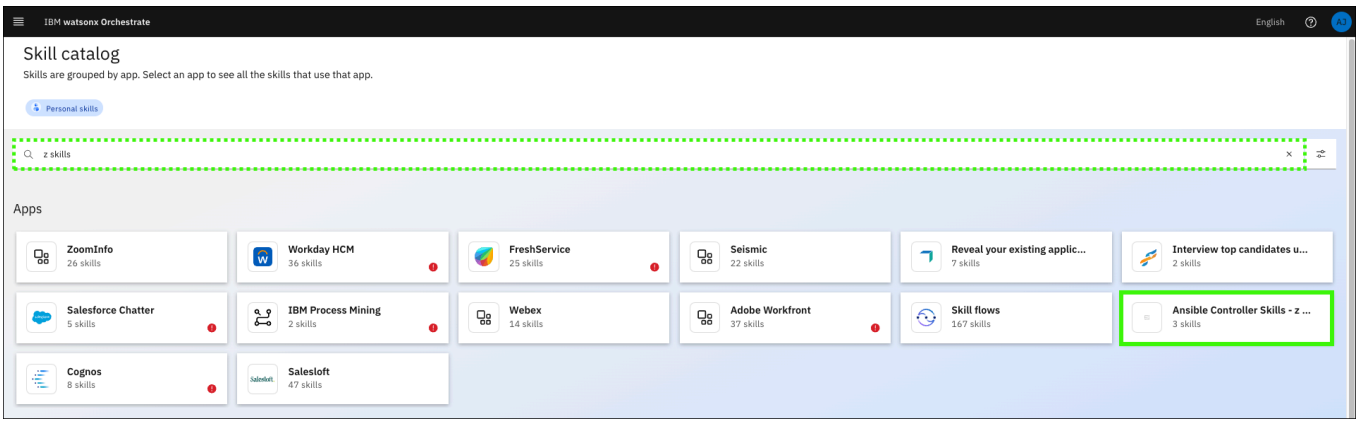


2. Search for the application name you specified earlier.

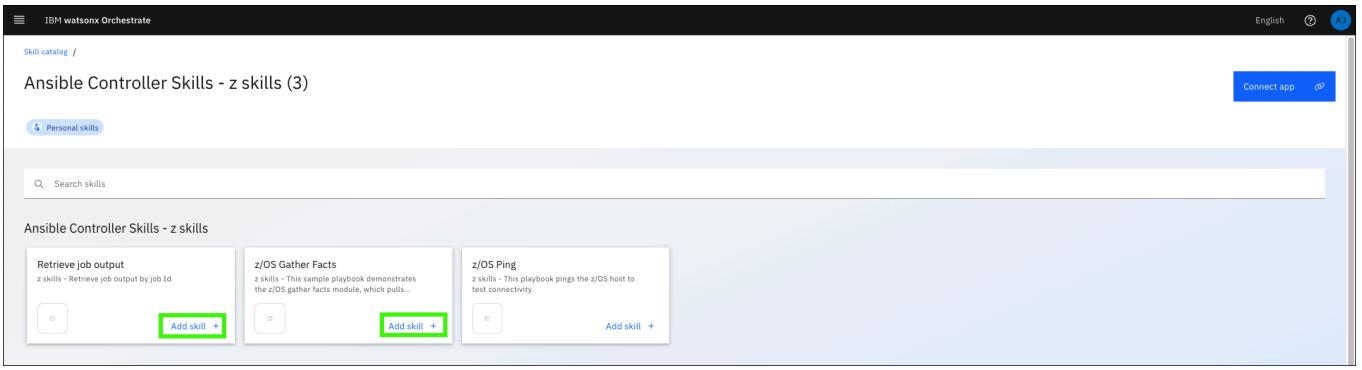


3. Click the tile for your application.

   Note, the tile name is proceeded by **Ansible Controller Skills**.

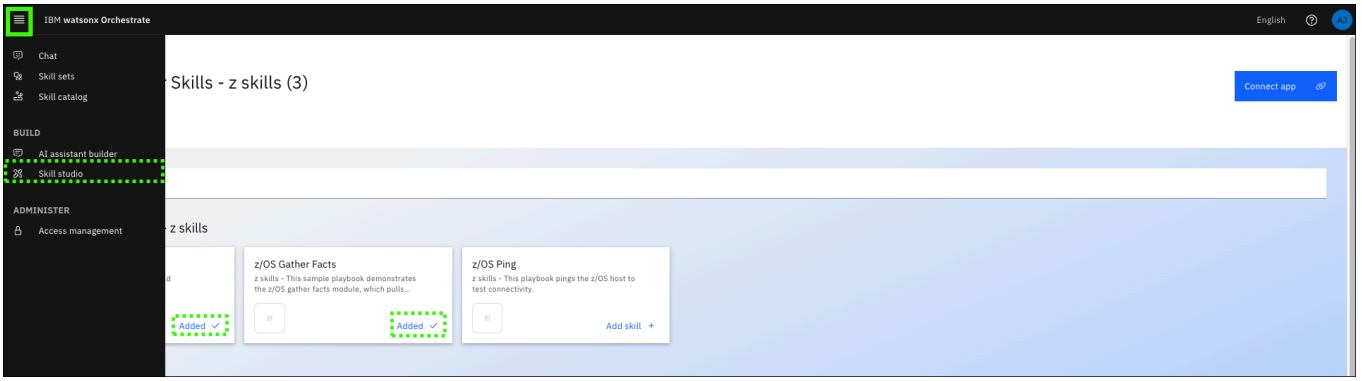


4. Click **Add skill** for each of the skills you want to add to the flow.

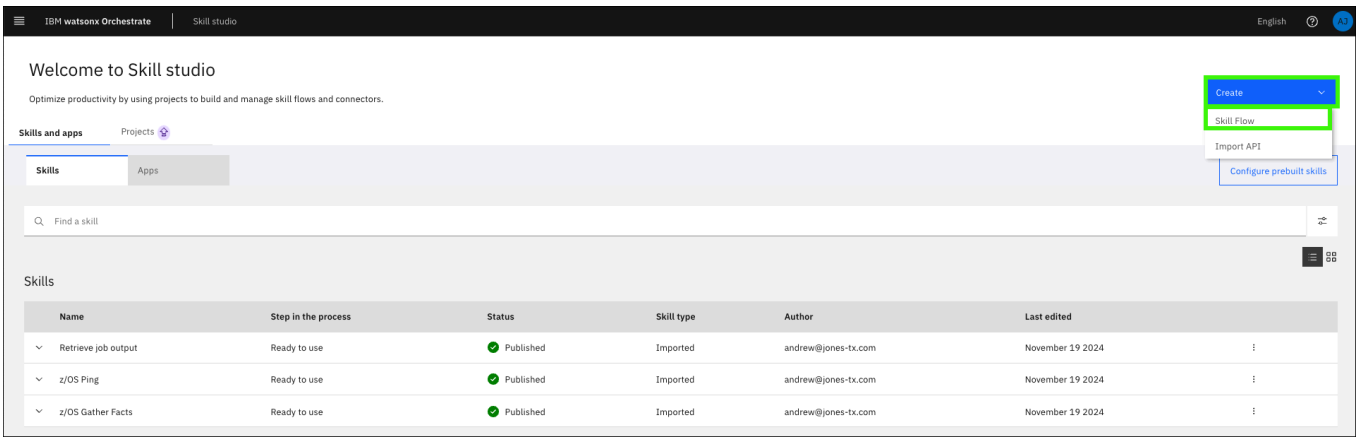


Create the skill flow

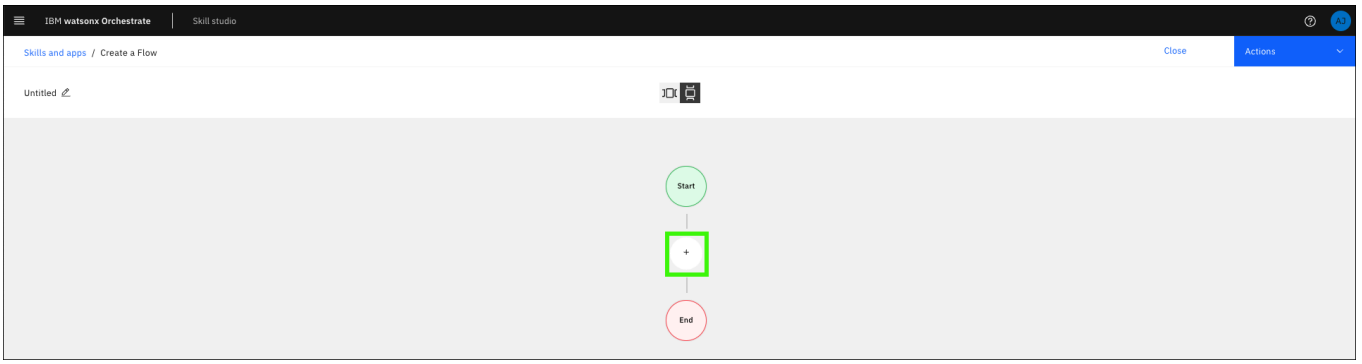
1. Click **Skill studio** in the main menu.



2. Expand the **Create** drop-down menu and click on **Skill flow**.

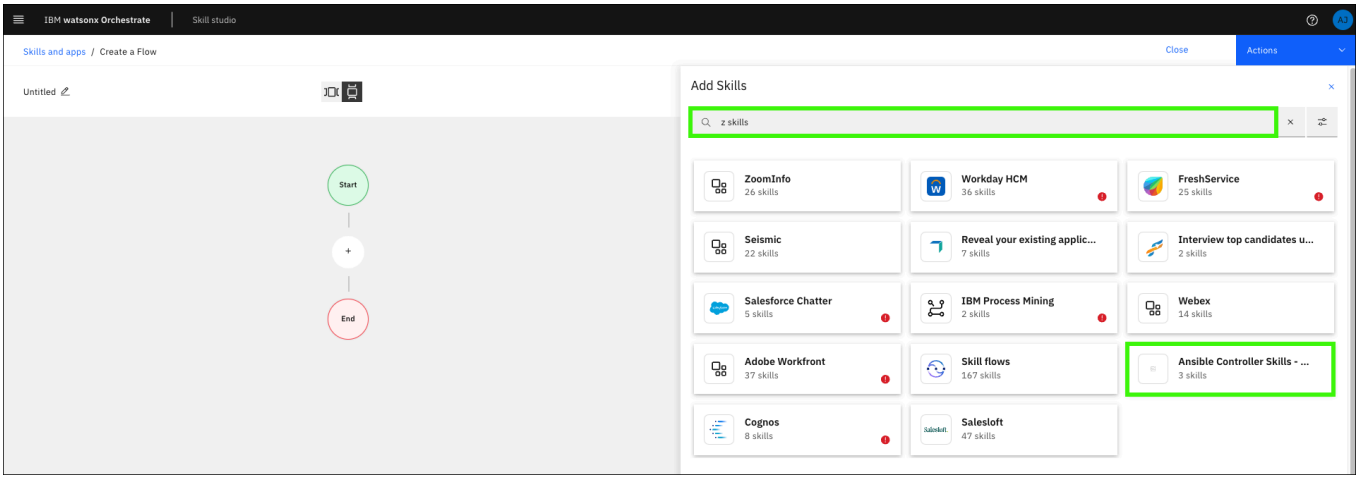


3. Click the **+** icon.

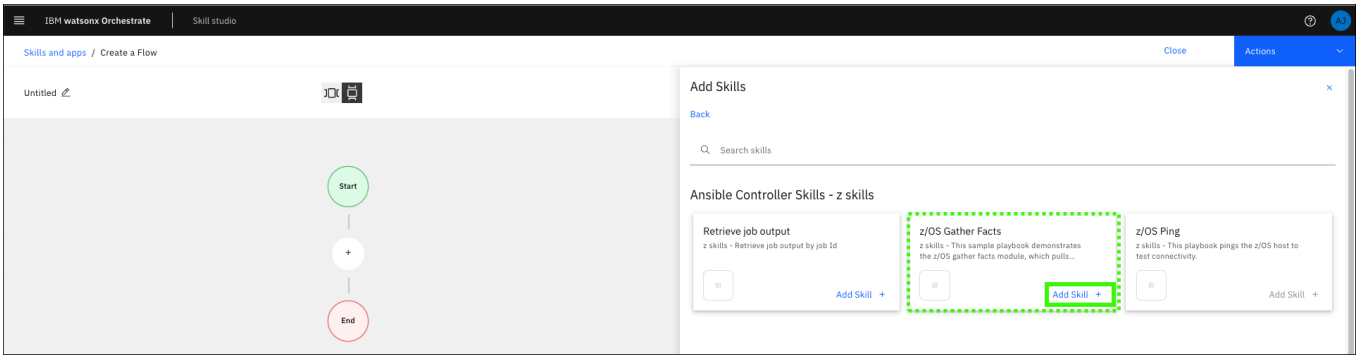


Next, you need to add the **z/OS Gather Facts** skill and the **Retrieve job output** skill to the skill flow. Use the **Search apps** function to locate the skills.

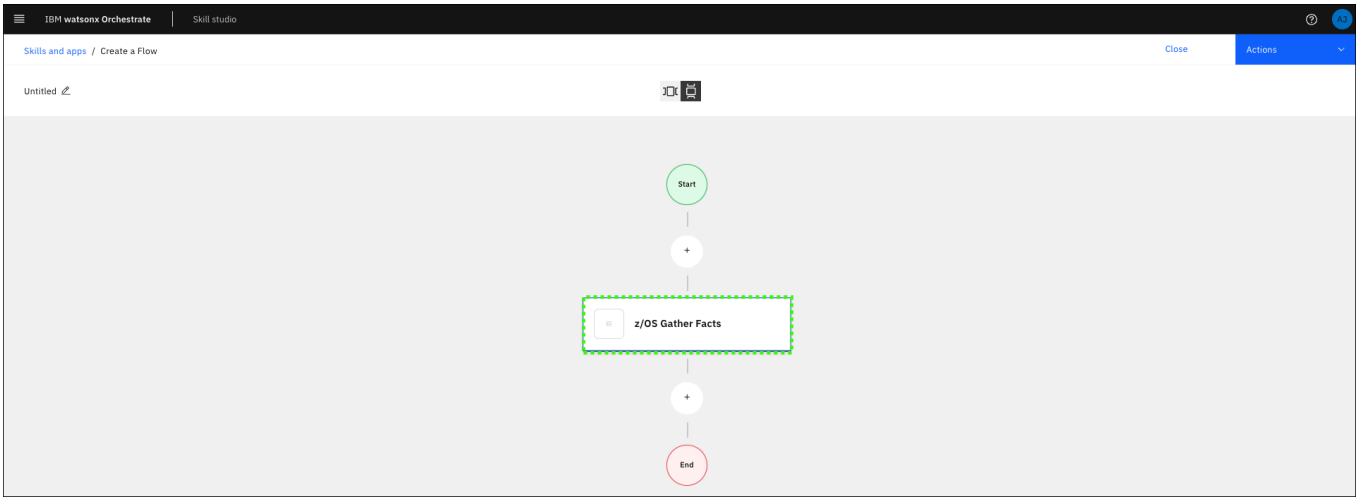
4. Search for the application name you specified earlier and click it's tile.



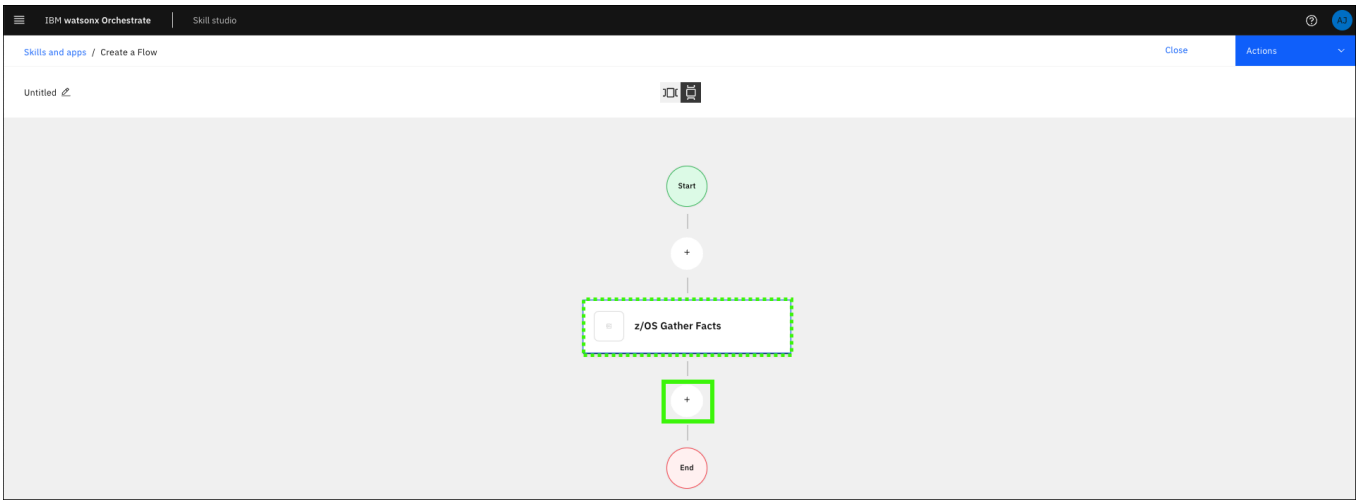
5. Click **Add Skill** in the **z/OS Gather Facts** tile.



6. Verify the **z/OS Gather Facts** skill is added to the skill flow.

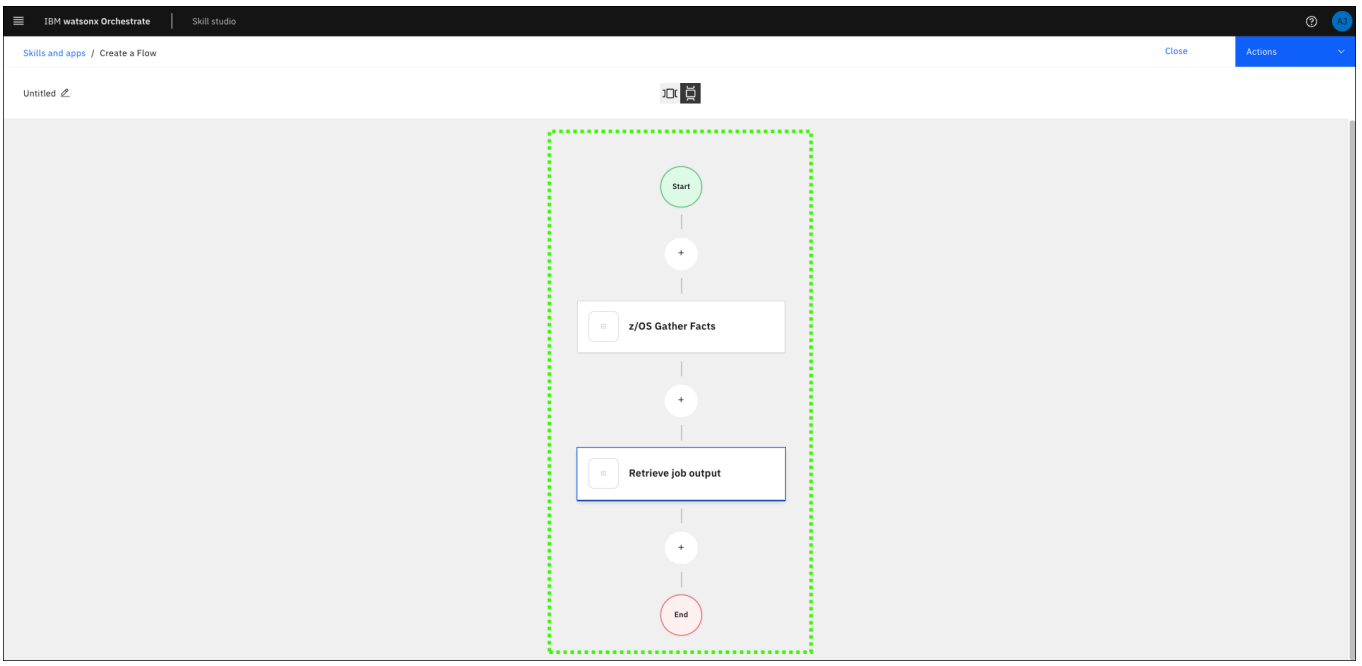


7. Click the **+** icon **after** the **z/OS Gather Facts** tile.



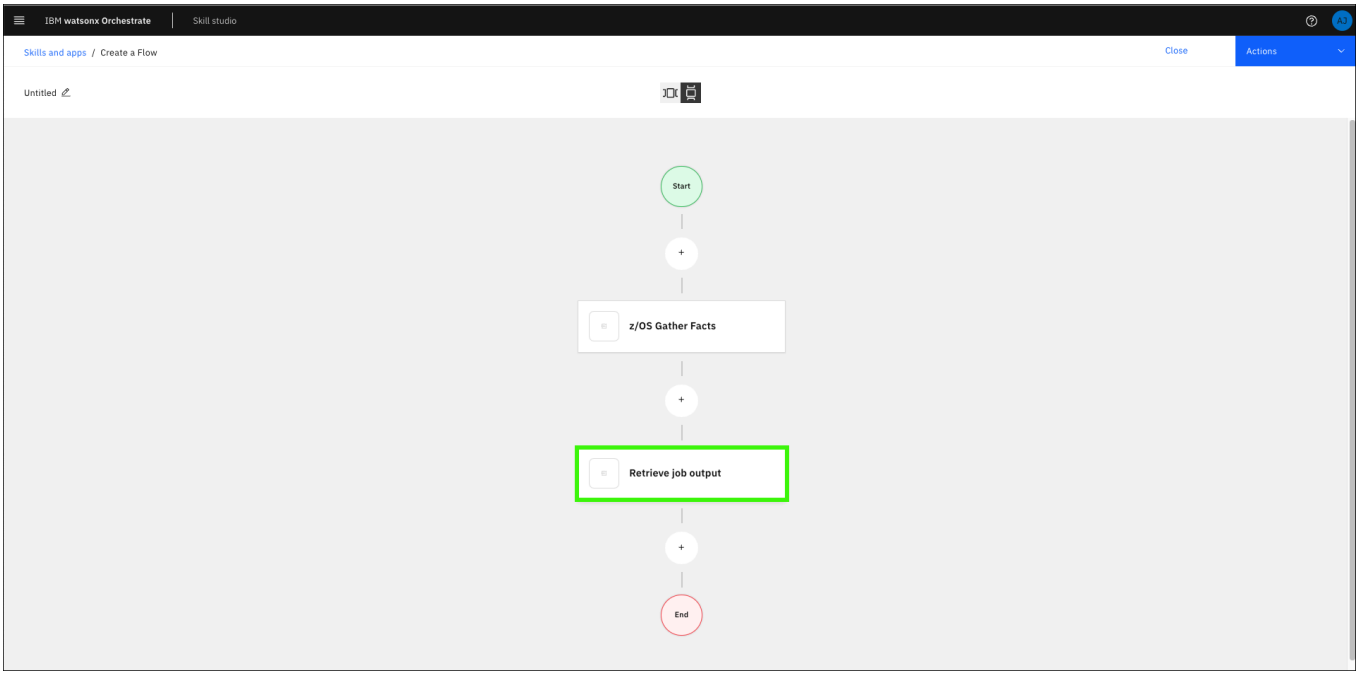
8. Repeat steps 5 and 6 for the **Retrieve job output** skill.

   After adding the **Retrieve job output** skill, your skill flow should like like:

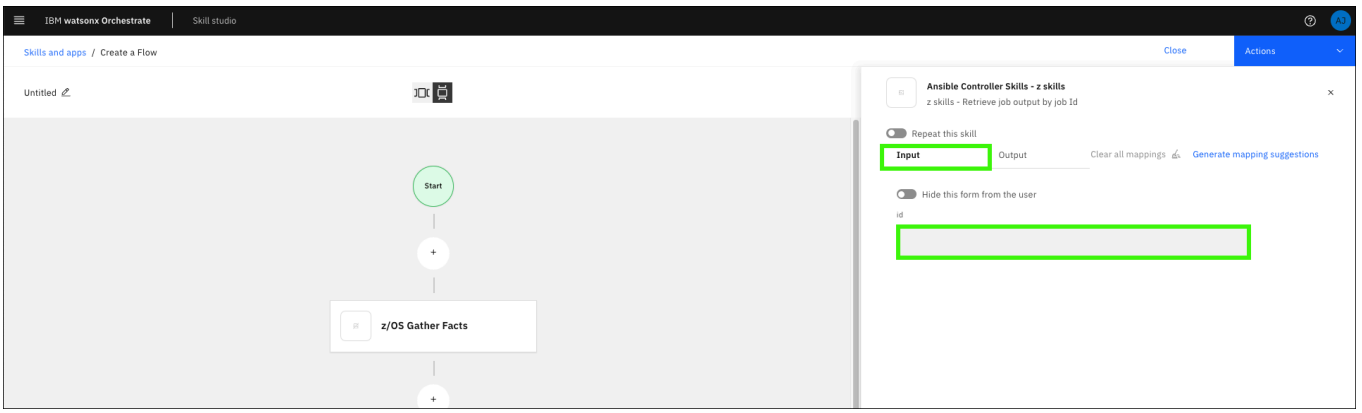


Next you must map the output values of the first skill to the input of the second skill. In this case, pass the “job id” output from **z/OS Gather Facts** as an input for **Retrieve job output**.

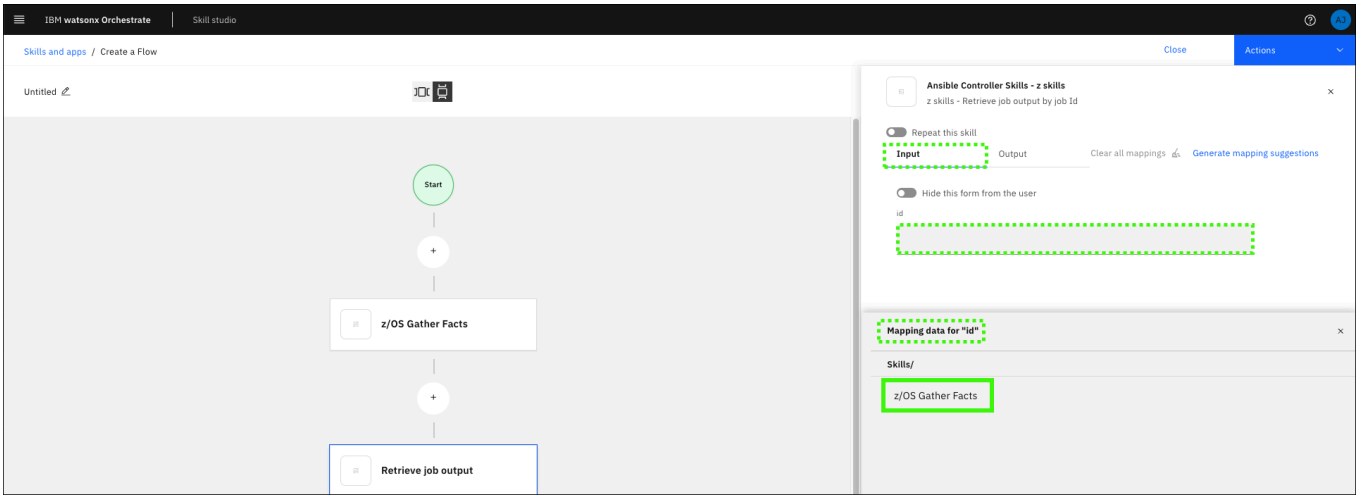
9. Click the **Retrieve job output** tile.



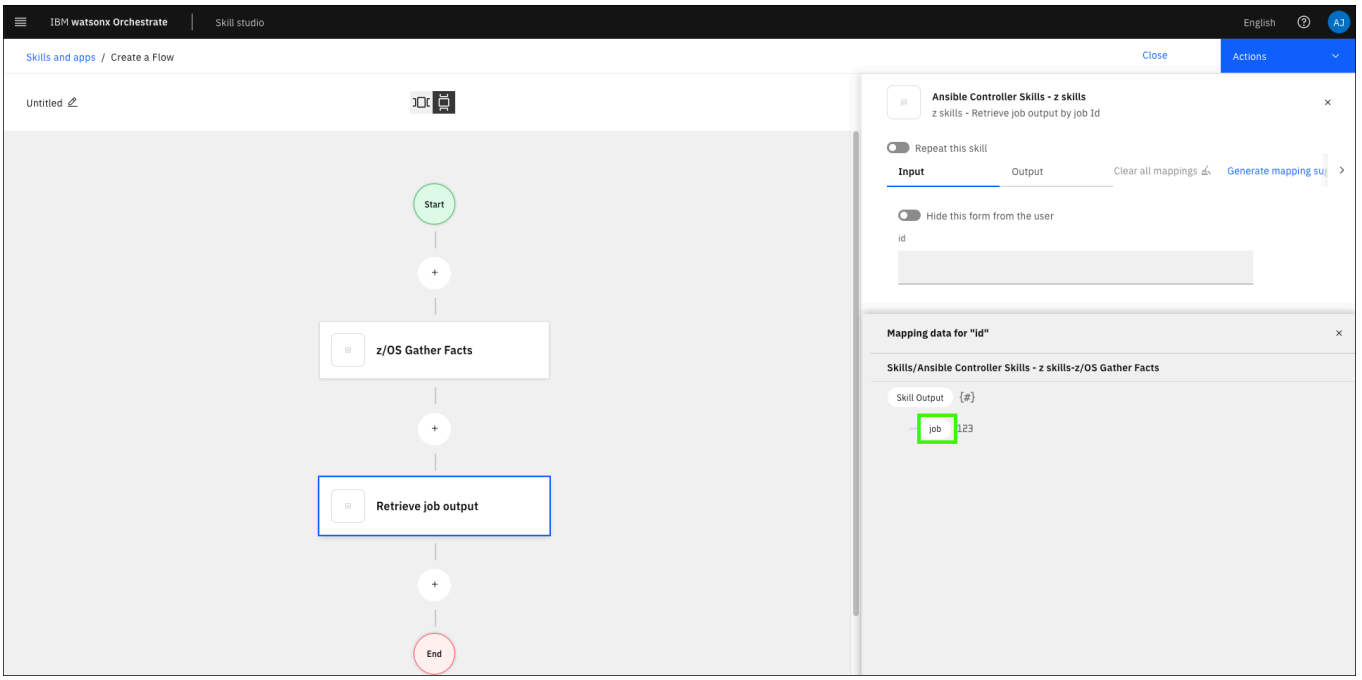
10. Select the **Input** tab and click in the **id** field.



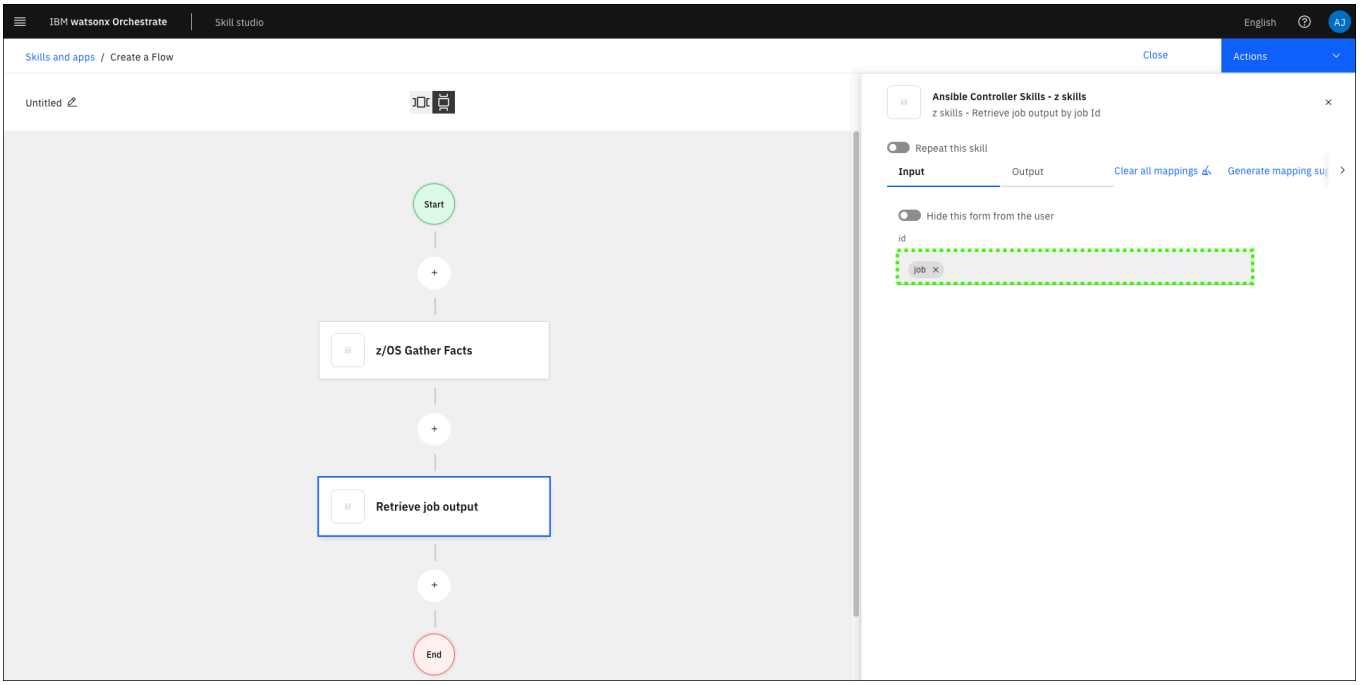
11. Click the **z/OS Gather Facts** skill in the **Mapping data for "id"** section.



12. Click the **job** icon.

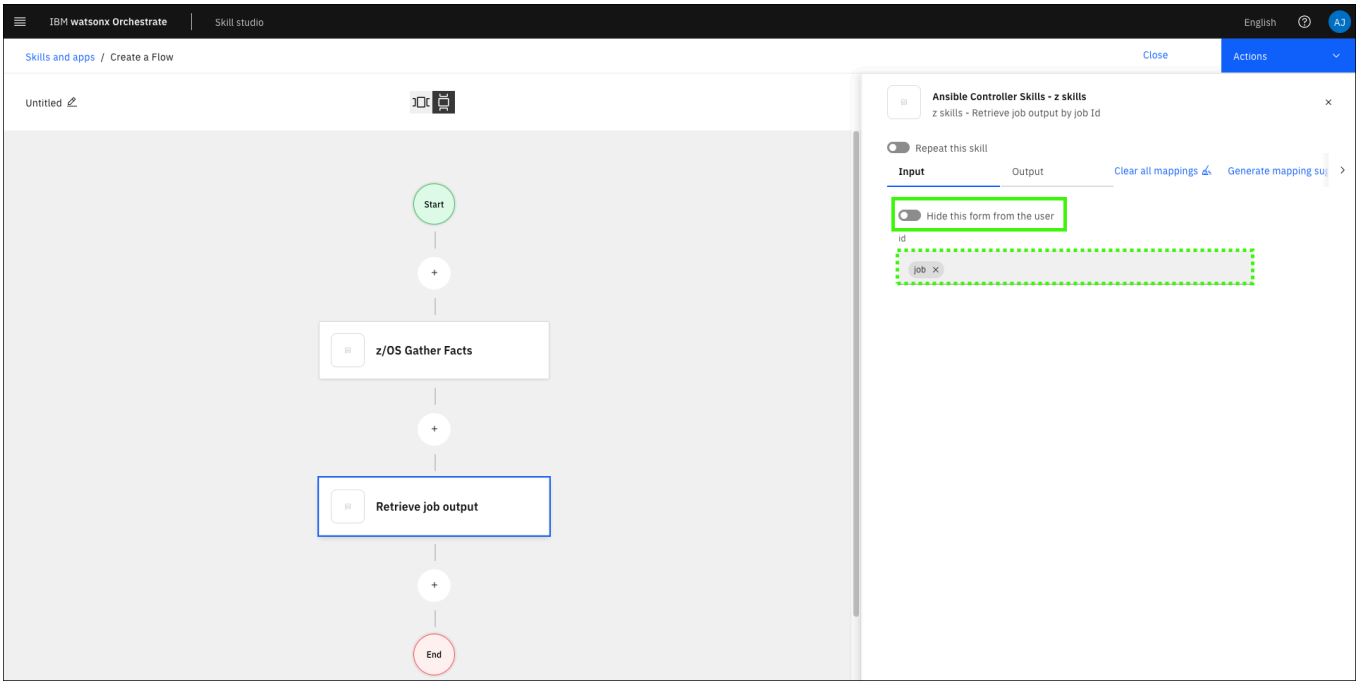


13. Verify the **job** appears in the **id** field.

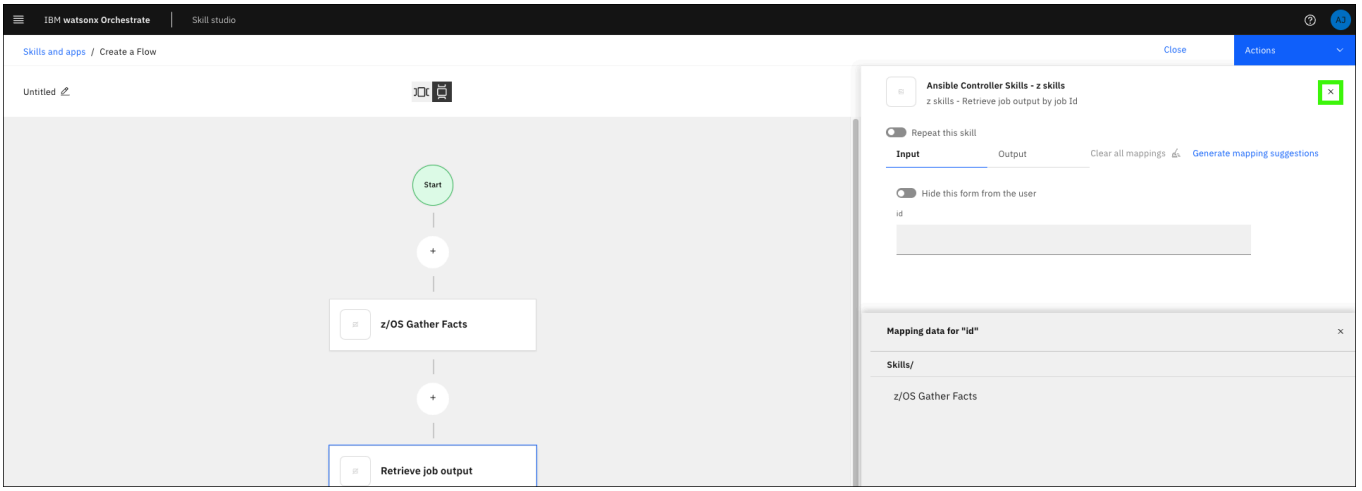


14. Optionally, toggle the **Hide this from from the user** setting.

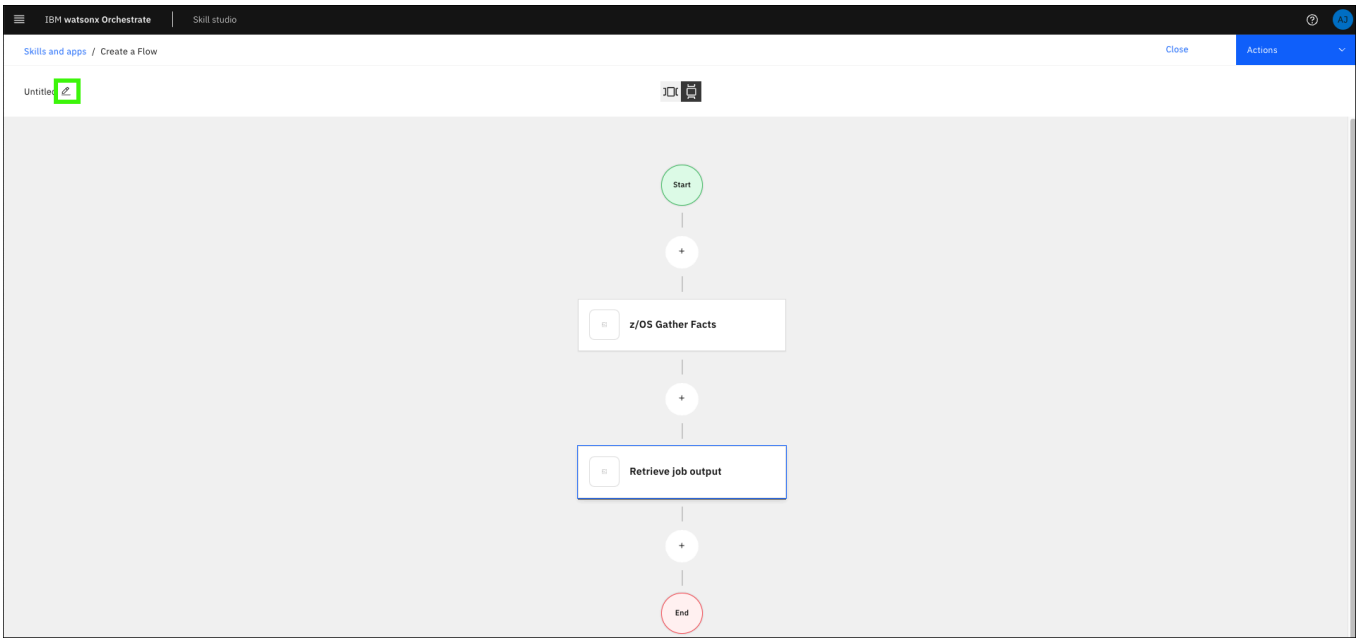
    For this lab guide, this option is left disabled. Learn more about this option here.



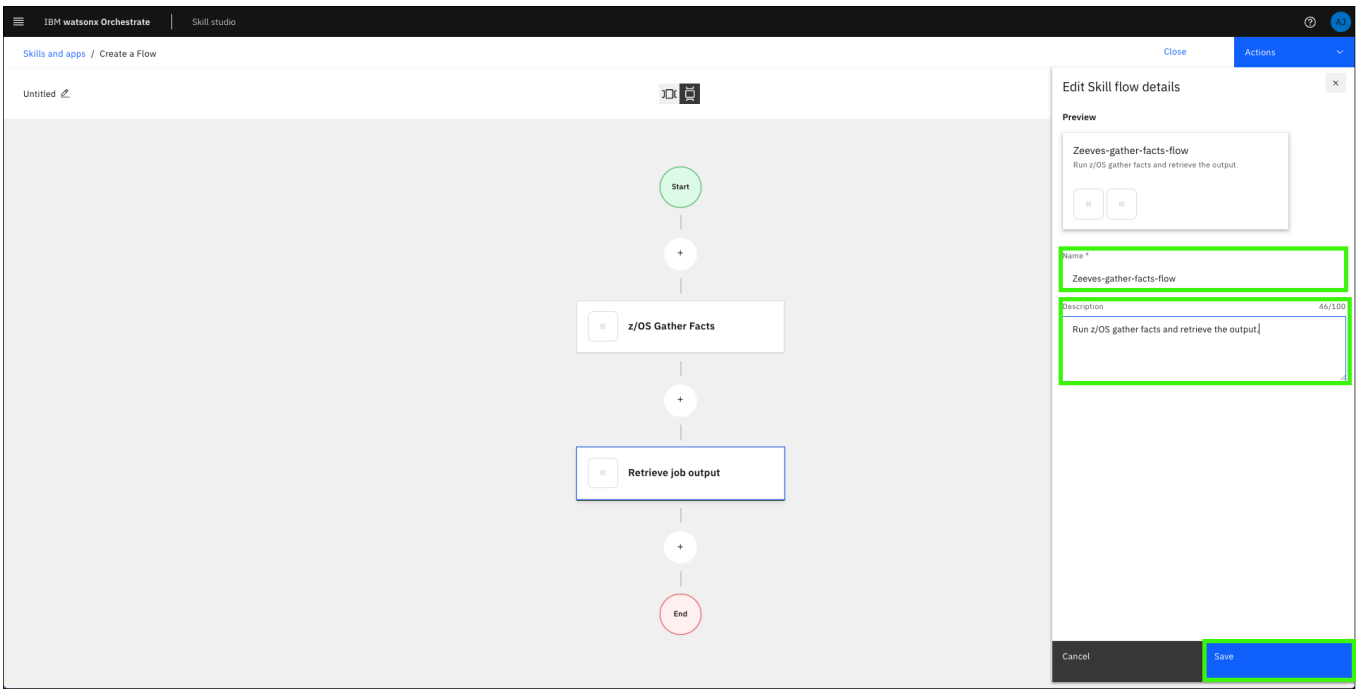
15. Click the **x** to close mapping window.



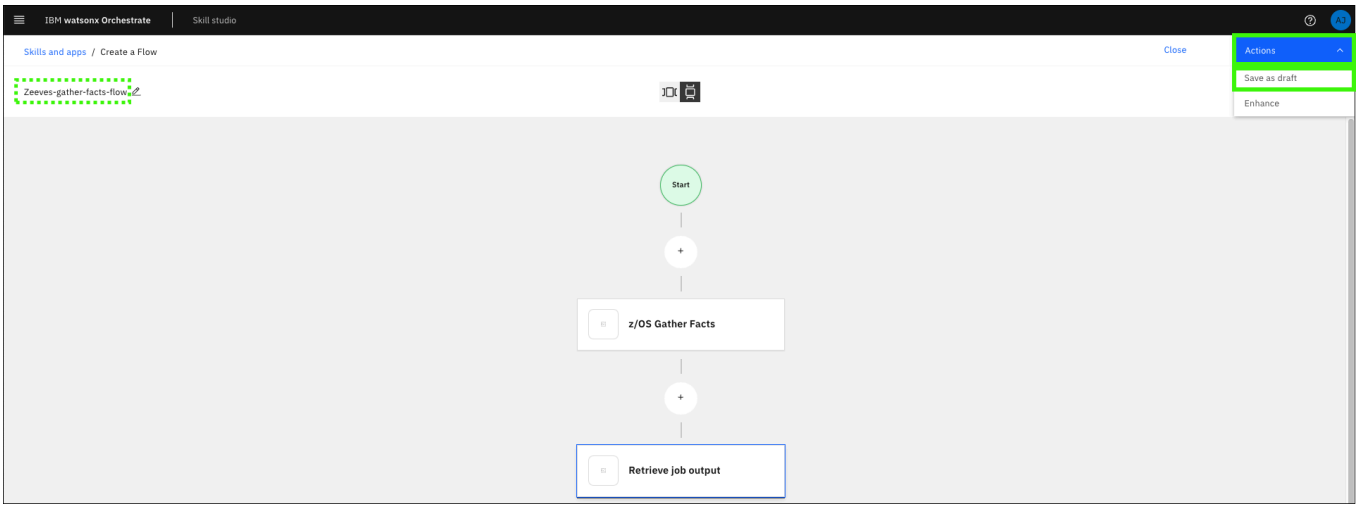
16. Click the pencil (✎).



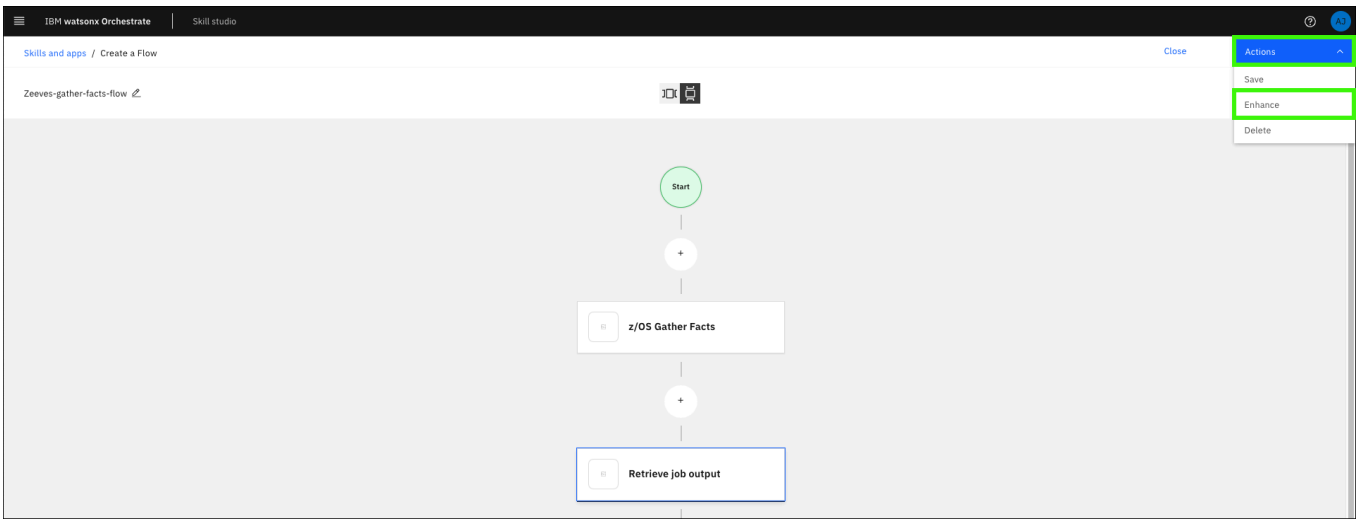
17. Enter a (a) **Name** and (b) **Description** for your skill flow and then (c) click **Save**.



18. Expand the **Actions** pull-down list and click **Save as draft**.



19. Expand the **Actions** pull-down list and click **Enhance**.



On the **Enhancing the skill** pages, you can:

- modify the skill name, description, and version
- add phrases (prompts) that will be recognized by the assistant to call the skill flow

20. Click the **Phrases** tab.

21. Replace the existing **phrases** (prompts) and then click **Publish**.

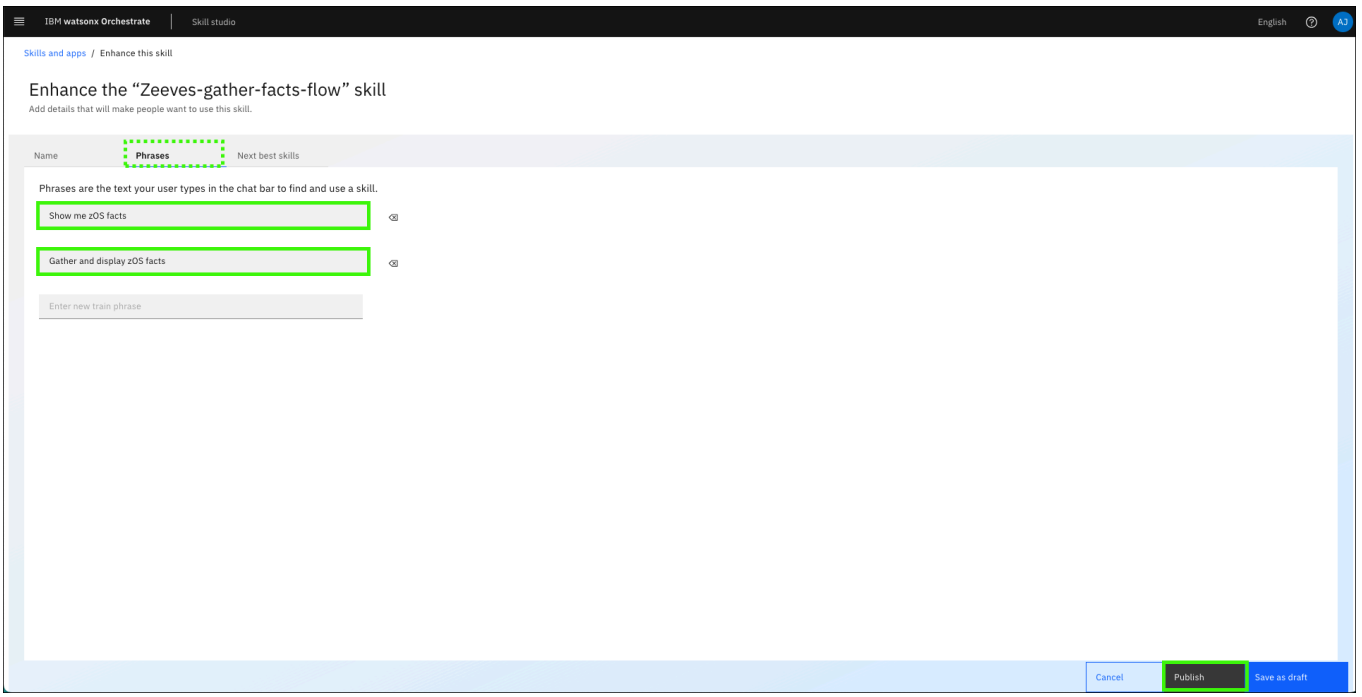
    Notice the default prompts are either not very intuitive (the skill flow name) or a bit verbose. Replace the existing phrases with phrases that you anticipate users will enter.

> **Be careful with the sample phrases you specify.**
>
> During the development of the lab guide, it was discovered that some sample phrases with a `/` character can cause issues with the actions. Avoid using **z/OS** in the your sample phrases. This issue has been reported to the offering team.

Example prompts:

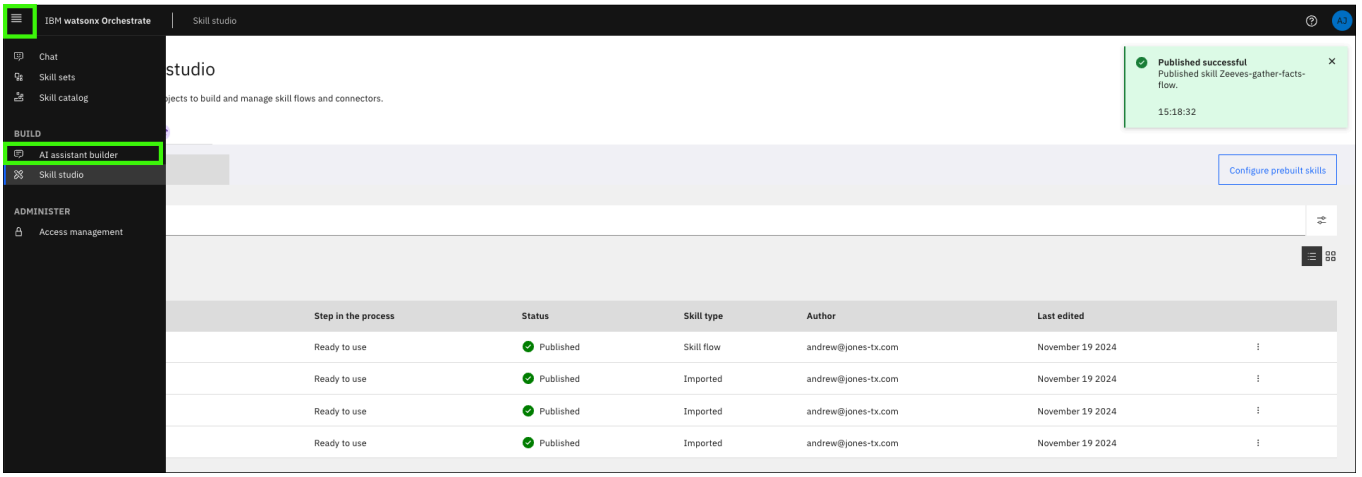
```
Show me zOS facts
```

```
Gather and display zOS facts
```

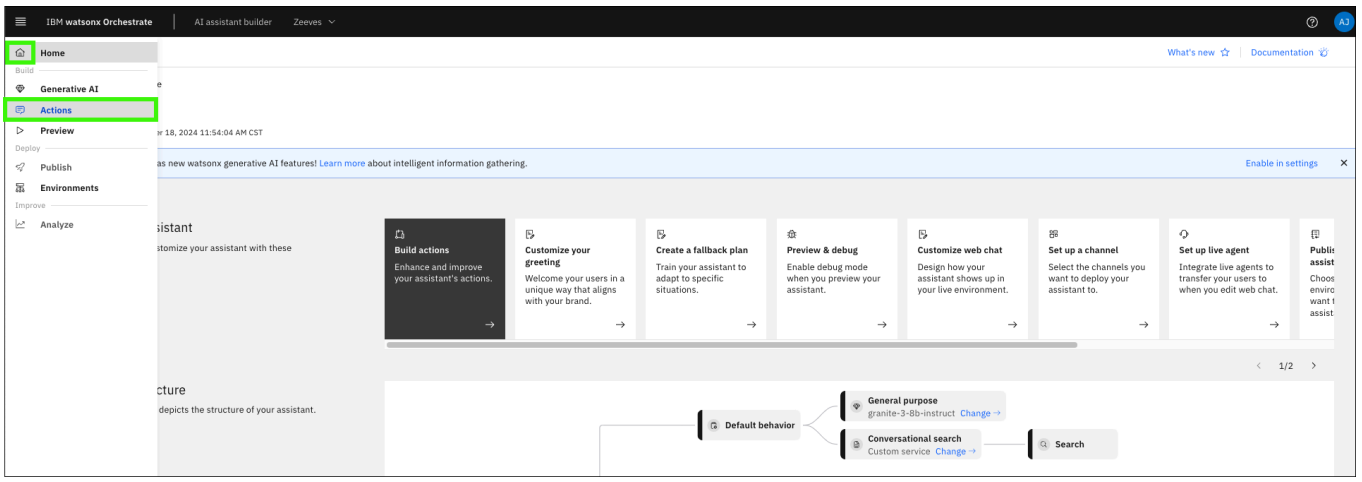


Enable the skill flow in your assistant

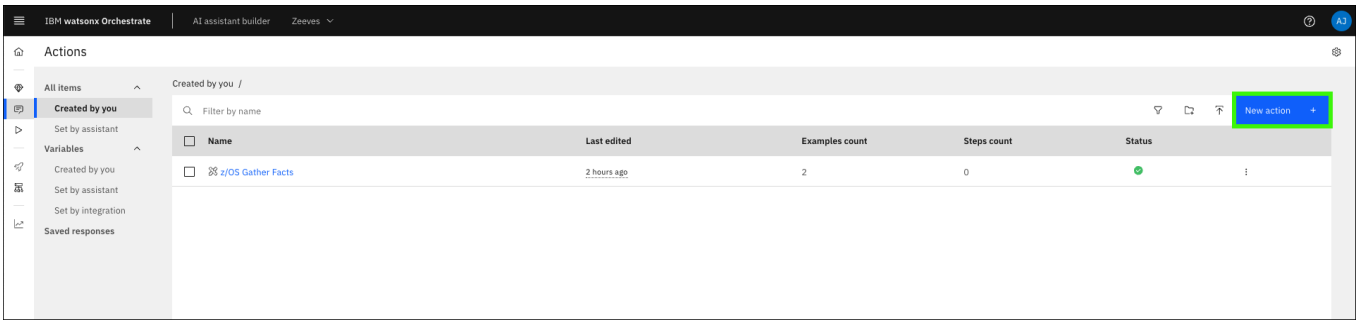
1. Click **AI assistant builder** in the main menu.



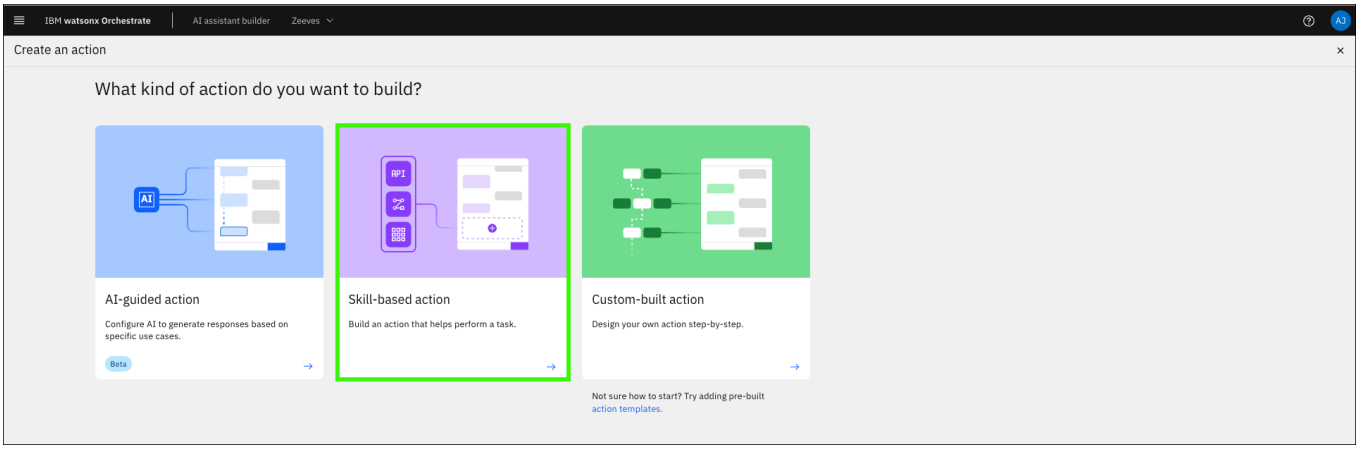
2. Hover over the **Home** ( ) and click **Actions**.



3. Click **New action**.

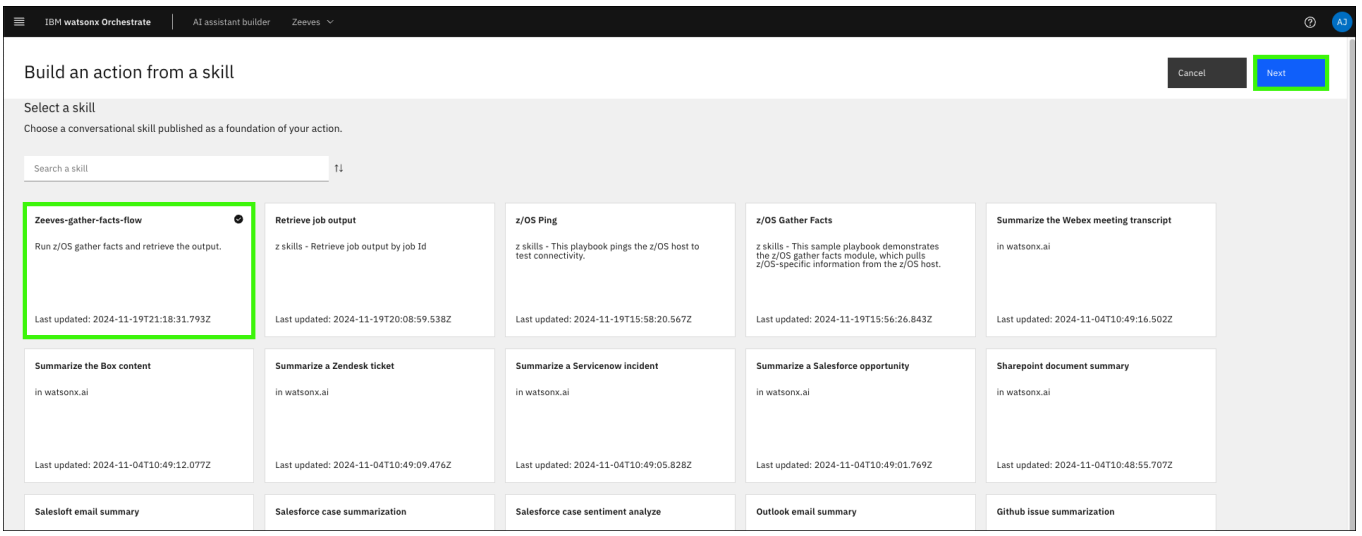


4. Click the **Skill-based action** tile.



5. Click the skill flow you created earlier and then click **Next**.

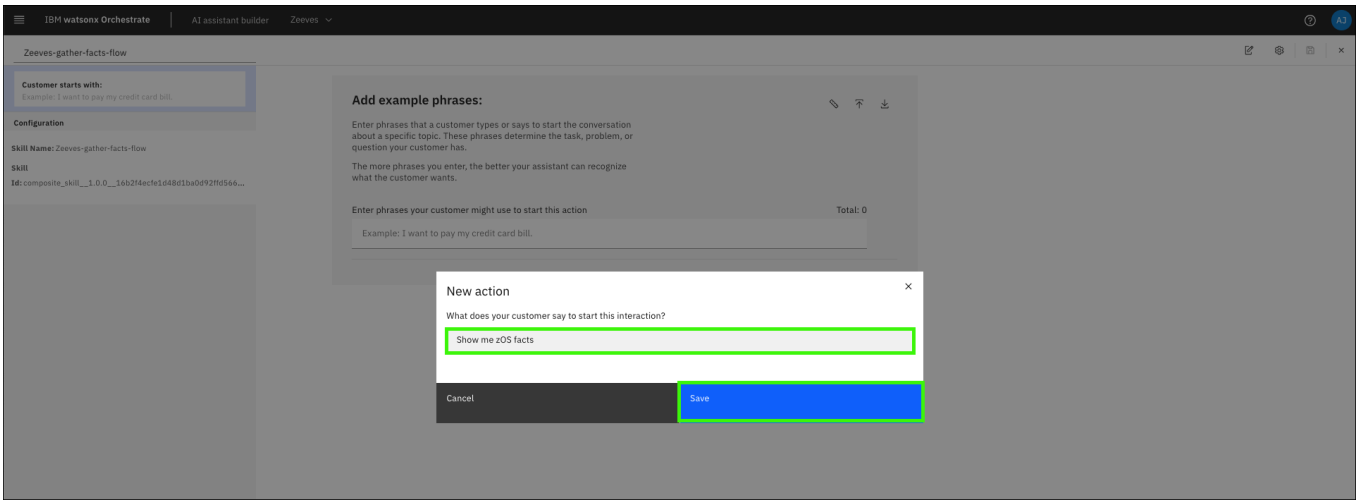
   Note: it may take a minute for the tiles to appear on the screen.



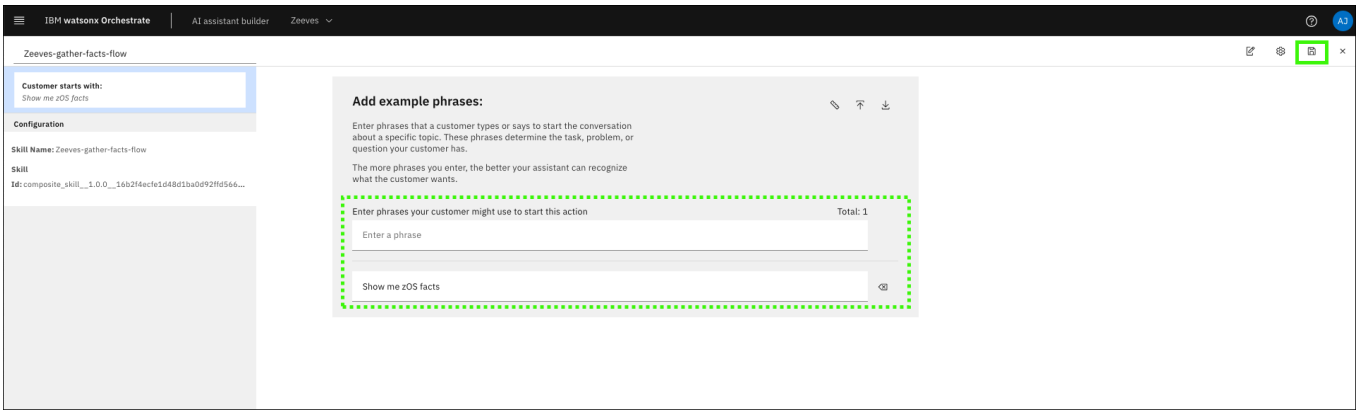
6. Enter an example prompt for the skill and click **Save**.

   You can use one of the prompts you used earlier for the skill flow.

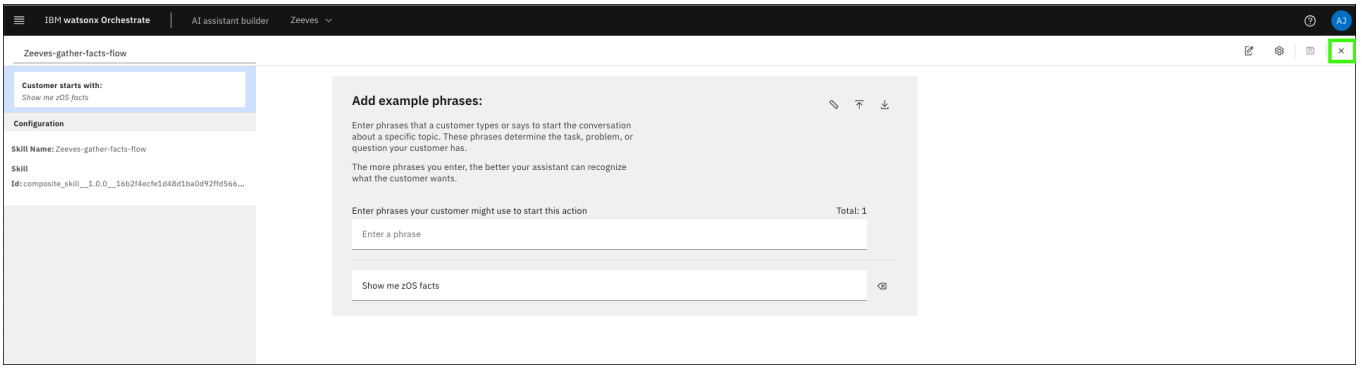
```
Show me zOS facts
```



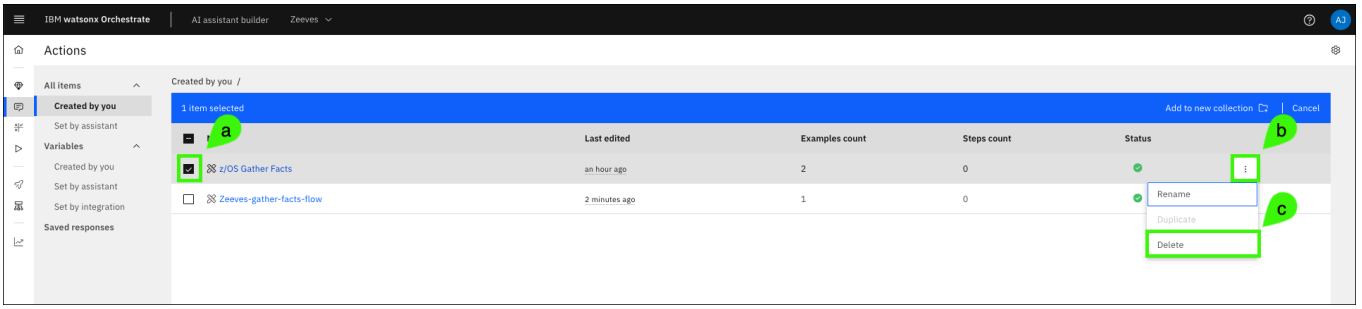
7. Enter any additional phrases (prompts) and then click the **save** ( ).



8. Click close (**x**).

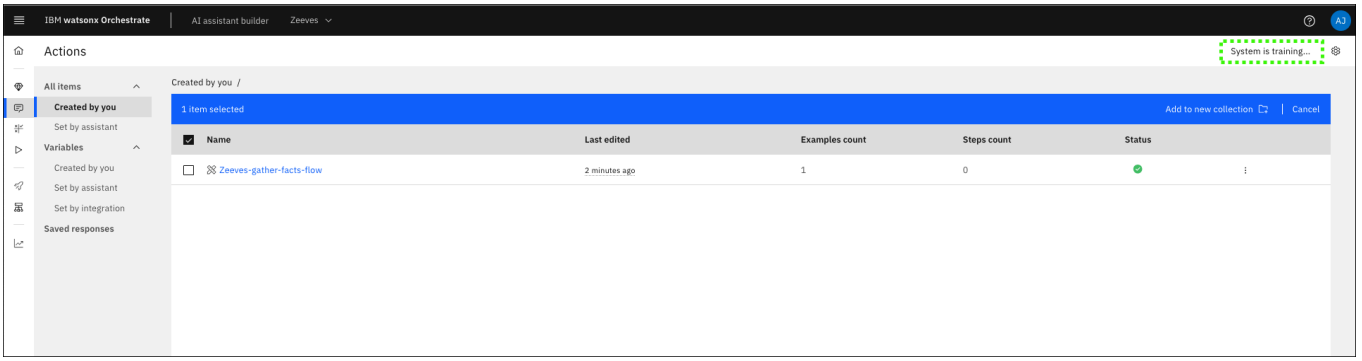


9. Select the original skill you created (**a**) (not the skill flow you just created), click the ellipses (**b**), and then click **Delete** (**c**).

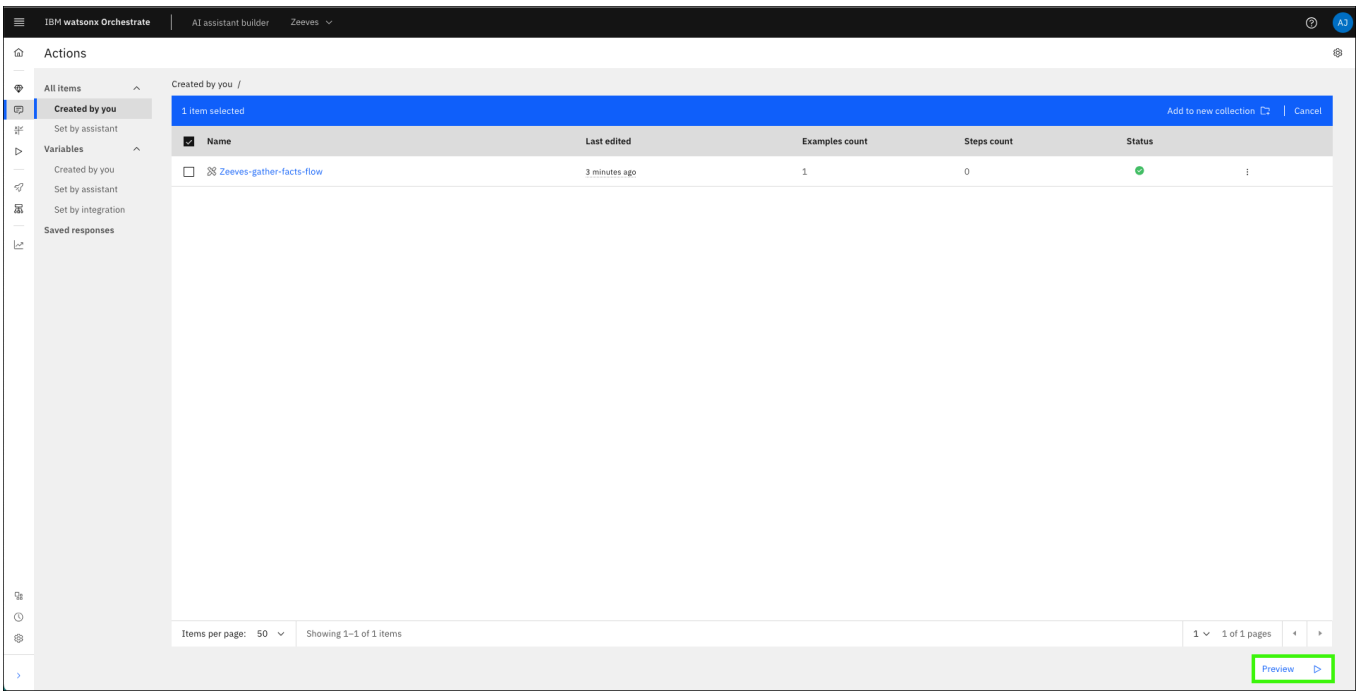


10. Wait for system training to complete.

    Note: The message will change to "System is trained" and then disappear.

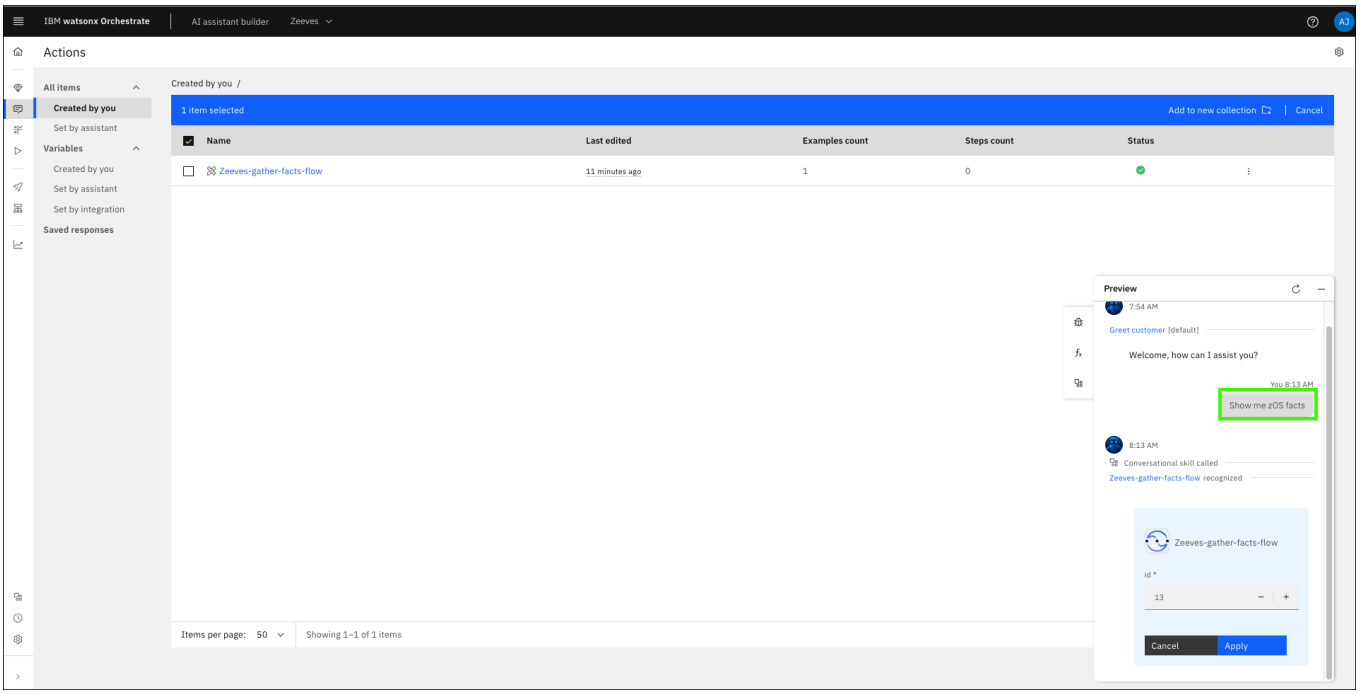


11. Click **Preview**.



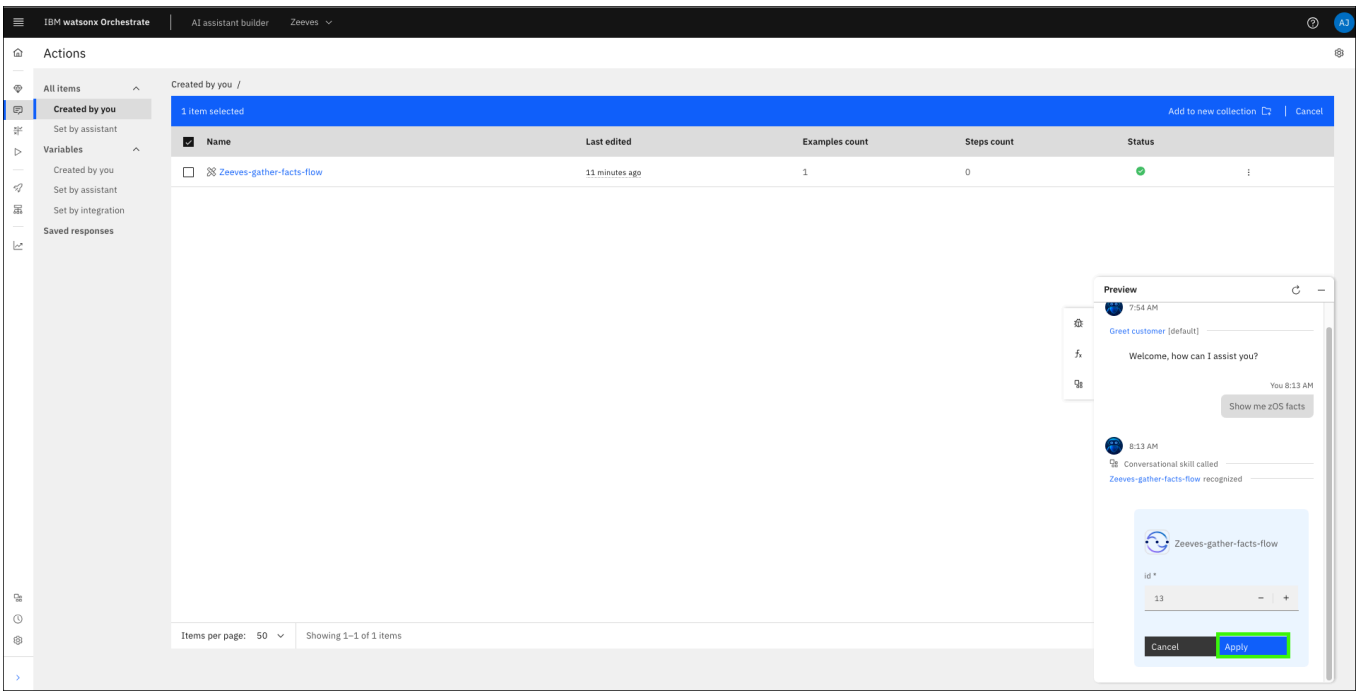
12. Enter one of the prompts you specified into the assistant preview.

```
Show me zOS facts
```



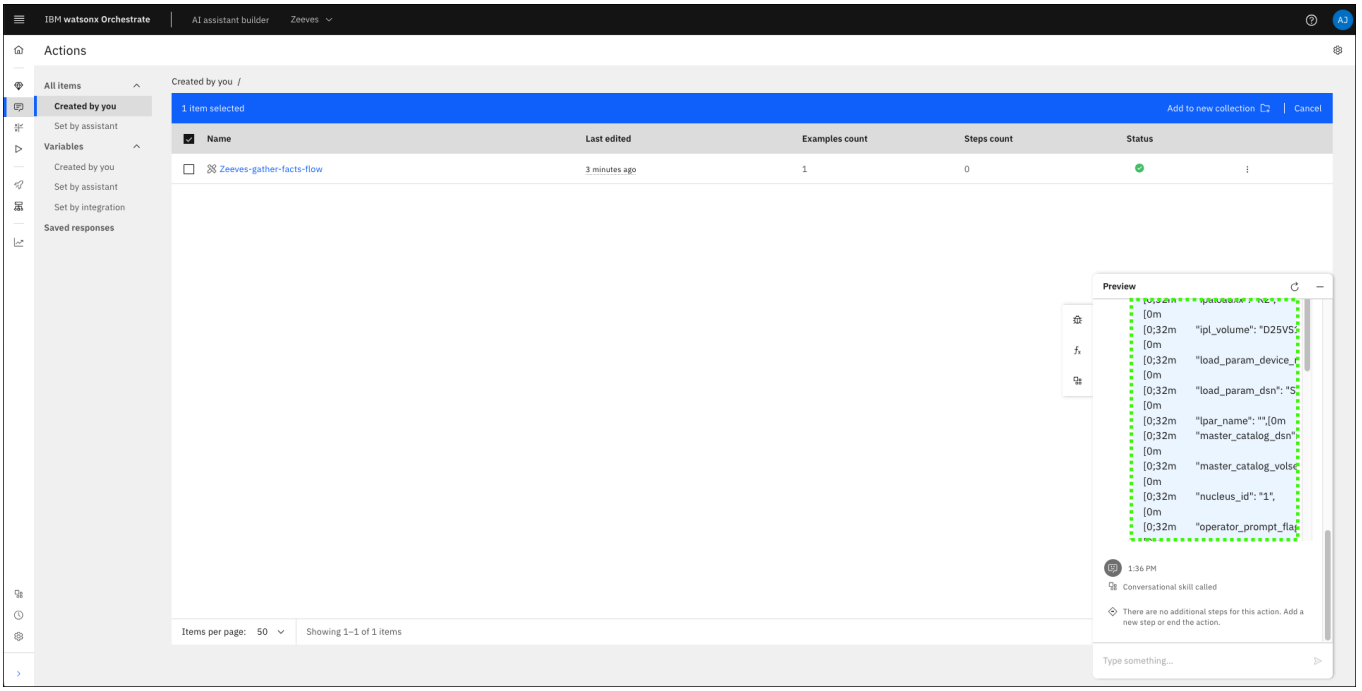
13. **Wait 10 seconds** and then click **Apply**.

    Note: It is important to wait for the first job to complete before submitting the second job in the flow.



14. Review the results from the skill flow.

    Use both scroll bars in the assistant preview to review all of the returned information. The output should be similar to what was seen in the AAP web console. The character strings like `[0;32m` are special characters that are not properly displayed in the assistant preview interface.



Sample output form the Z/OS gather facts flow.

Content

Identity added: /runner/artifacts/16/ssh_key_data (/runner/artifacts/16/ssh_key_data)
[1;35m[WARNING]: Collection ibm.ibm_zos_core does not support Ansible version 2.14.2[0m

PLAY [Gather z/OS-specific facts.] *********

TASK [Gather all facts about z/OS host.] ********* [0;32mok: [zos_host][0m

TASK [Print gathered facts about the master catalog.] **** [0;32mok: [zos_host] => {[0m [0;32m "msg": [[0m [0;32m "master catalog dsn: CATALOG.VS01.MASTER",[0m [0;32m "master catalog volser: OPEVS1"[0m [0;32m][0m [0;32m}[0m

TASK [Print only CPC and IODF info from gathered z/OS facts.] ****** [0;32mok: [zos_host] => {[0m [0;32m "msg": [[0m [0;32m "manufacturer: IBM",[0m [0;32m "model: A00",[0m [0;32m "plant: C1",[0m [0;32m "iodf name: PROV.IODF00",[0m [0;32m "iodf config: DEFAULT"[0m [0;32m][0m [0;32m}[0m

TASK [Print out all gathered facts about the z/OS host.] ***** [0;32mok: [zos_host] => {[0m [0;32m "ansible_facts": {[0m [0;32m "arch_level": "2",[0m [0;32m "cpc_nd_manufacturer": "IBM",[0m [0;32m "cpc_nd_model": "A00",[0m [0;32m "cpc_nd_plant": "C1",[0m [0;32m "cpc_nd_seqno": "20D90792EB76",[0m [0;32m "cpc_nd_type": "008562",[0m [0;32m "edt": "00", [0m [0;32m "hw_name": "",[0m [0;32m "ieasym_card": "(00,K2)",[0m [0;32m "io_config_id": "00",[0m [0;32m "iodate": "",[0m [0;32m "iodesc": "",[0m [0;32m "iodf_config": "DEFAULT",[0m [0;32m "iodf_name": "PROV.IODF00",[0m [0;32m "iodf_unit_addr": "DE28",[0m [0;32m "ioproc": "",[0m [0;32m "iotime": "",[0m [0;32m "ipaloadxx": "K2",[0m [0;32m "ipl_volume": "D25VS1",[0m [0;32m "load_param_device_num": "DE28",[0m [0;32m "load_param_dsn": "SYS0.IPLPARM",[0m [0;32m "lpar_name": "",[0m [0;32m "master_catalog_dsn": "CATALOG.VS01.MASTER",[0m [0;32m "master_catalog_volser": "OPEVS1",[0m [0;32m "nucleus_id": "1",[0m [0;32m "operator_prompt_flag": "M",[0m [0;32m "parmlib_dsn": "K2.PARMLIB",[0m [0;32m "parmlib_volser": "USRVS1",[0m [0;32m "primary_jes": "JES2",[0m [0;32m "product_mod_level": "00",[0m [0;32m "product_name": "z/OS",[0m [0;32m "product_owner": "IBM CORP",[0m [0;32m "product_release": "05",[0m [0;32m "product_version": "02",[0m [0;32m "smf_name": "VS01",[0m [0;32m "sys_name": "VS01",[0m [0;32m "sysplex_name": "LOCAL",[0m [0;32m "tsoe_rel": "05",[0m [0;32m "tsoe_ver": "4",[0m [0;32m "vm_name": ""[0m [0;32m }[0m [0;32m}[0m

PLAY RECAP ********** [0;32mzos_host[0m : [0; 32mok=4
[0m changed=0 unreachable=0 failed=0 skipped=0 rescued=0 ig nored=0

The scenario shown above may or may not be relevant for your client's use case. It is intended to show you how to sequence skills together in a skill flow to create an action that your assistant triggers based on prompts using the pre-configured Ansible automation templates. You are encouraged to create your own skill flows and prompts using other skills available within the AAP instance. As an example, create a skill flow for the **z/OS Ping** skill. Be sure to add the **Retrieve job output** skill to view the results.

Next, learn about custom-built actions.

Creating custom-built actions

To this point, you have learned how to:

- import skills into watsonx Orchestrate
- add applications with those skills to your assistant
- create skill-based actions for your assistant
- combine skills in a skill flow

There is also the ability to create **custom-built** actions. Custom-built actions allow you to create new actions with different steps to take in conversations and form sequences of prompts that define the conversation experience. The steps can be defined with or without conditions, which help control the custom responses. Steps within the custom action can end with routing to conversational search, triggering another existing sub-action, and other actions. This is a powerful way of customizing the end-user's experience.

Learn more about creating custom-built actions here.

Importing pre-packaged z/OS skills

Provided with Version 2 of watsonx Assistant for Z is a set of pre-packaged skills which can be used to automate various tasks on z/OS, such as running different console commands and retrieving logs from batch jobs.

The list of pre-packaged skills available include:

- Authorized program list
- z/OS IPL Information
- Display zOS parmlib datasets
- Unix System services options
- Display zOS subsystems
- List spool files
- Retrieve dataset content
- Retrieve spool file content
- Retrieve z/OS Management Facility (OSMF) job status

IBM watsonx Orchestrate requires that any OSMF environment you connect to for skill execution has certificate authority (CA) signed certificates. In the case of the Ansible Automation Platform (AAP) & Wazi z/OS environment provisioned in IBM Technology Zone (ITZ), the z/OS system is not currently using CA signed certificates and therefore cannot execute the pre-packaged skills on your own z/OS system. For demo purposes, it is still recommended to import them so that the pre-packaged skills can be shown.

Work is in progress to modify the ITZ environments so that they are enabled for skill execution using these pre-packaged skills. In the meantime, the underlying automation for these pre-packaged skills can still be demonstrated using the Ansible skill 'z/OS Operator Command' which is available to import. This works because the pre-packaged skills are executing console commands directly using OSMF APIs which can also be run using the Ansible template skill 'z/OS Operator command'. For example, here are the console commands being used in some of the pre-packaged skills:

- Authorized Program list – `operator command -> d prog,lnklist`
- z/OS IPL Information - `operator command -> d iplinfo`
- Display zOS parmlib datasets - `operator command -> d parmlib`

You can import the pre-packaged skills into your sandbox environment by downloading the .zip file from here and following these instructions.

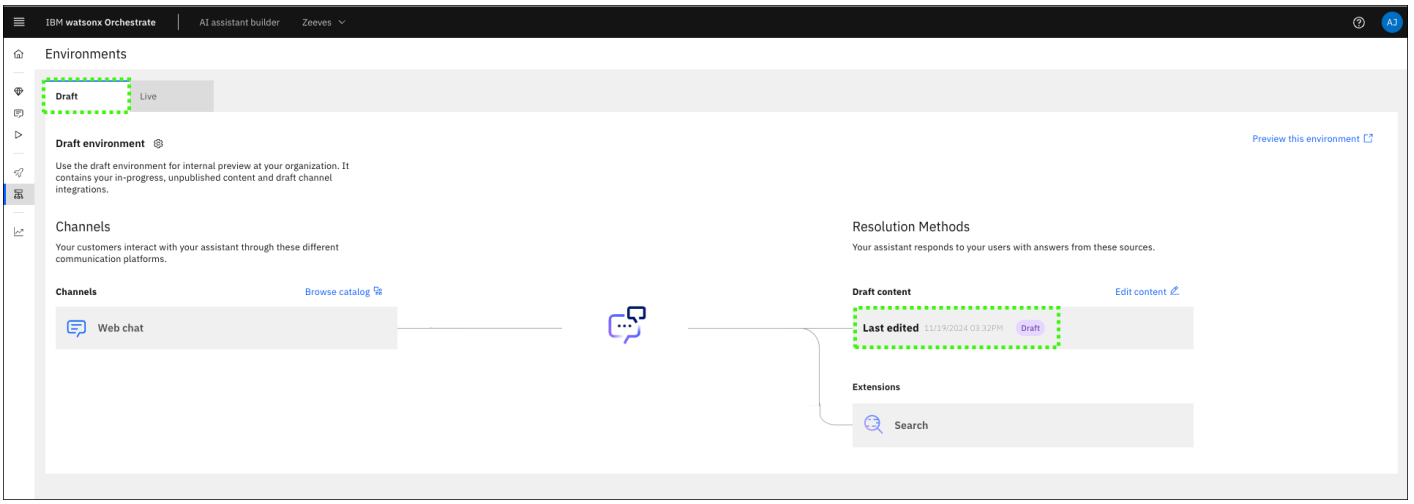
You must extract the imbedded JSON file and modify the file for your environment by following these instructions.

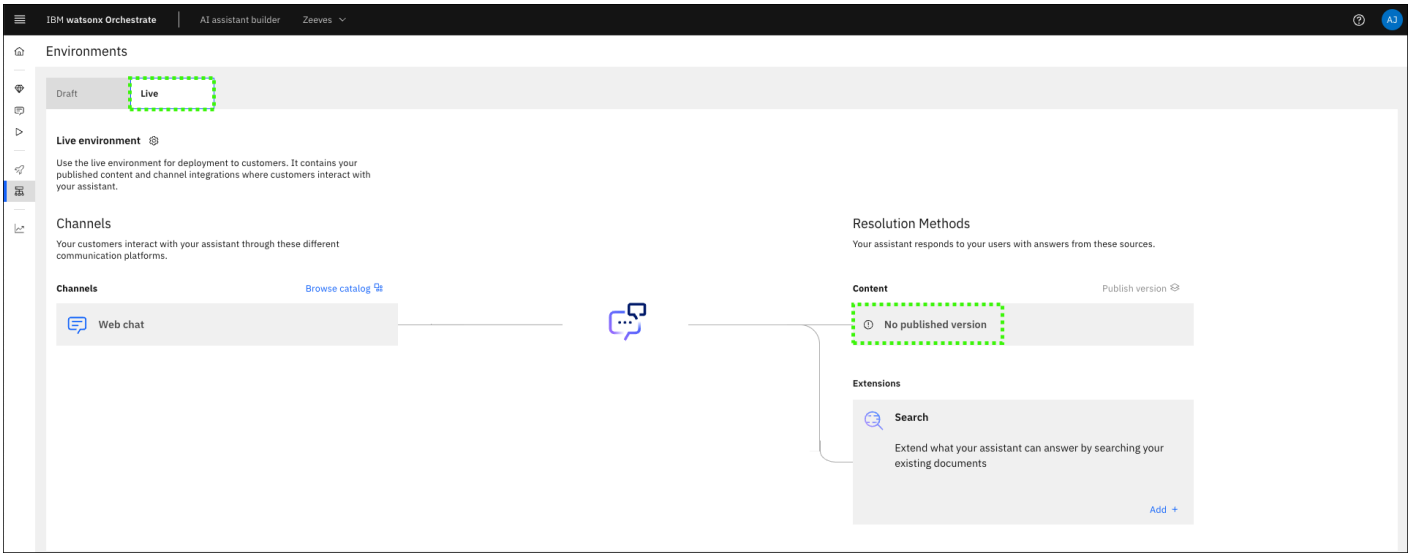
Publishing and deploying your assistant

To this point, acting as an Assistant Builder, you have built out the assistant, configured conversational search, and added skills and automations. While doing so, you have been testing your assistant using the **preview** capability of AI Assistant Builder. The **preview** capability is a closed environment for experimenting with prompts.

After your assistant is finalized, you can publish it to make it available to end-users. Each assistant you create comes with two **environments**: *draft* and *live*. You have been configuring your assistant in the draft environment. Each environment has its own set of IDs, URLs, and service credentials that can be referenced by external services.

The **Environments** page in the AI assistant builder has tabs for managing both the **Draft environment** and the **Live environment**:





The **Draft environment** contains all your in-progress work in the Actions, Preview, and Publish pages. Use the **Draft environment** tab to build out your assistant and use for internal testing before deployment. Any integrations you use (i.e. channels) for the **Draft environment** are unique to that environment, and changes to draft integrations don't affect the **Live environment**.

Publish the assistant

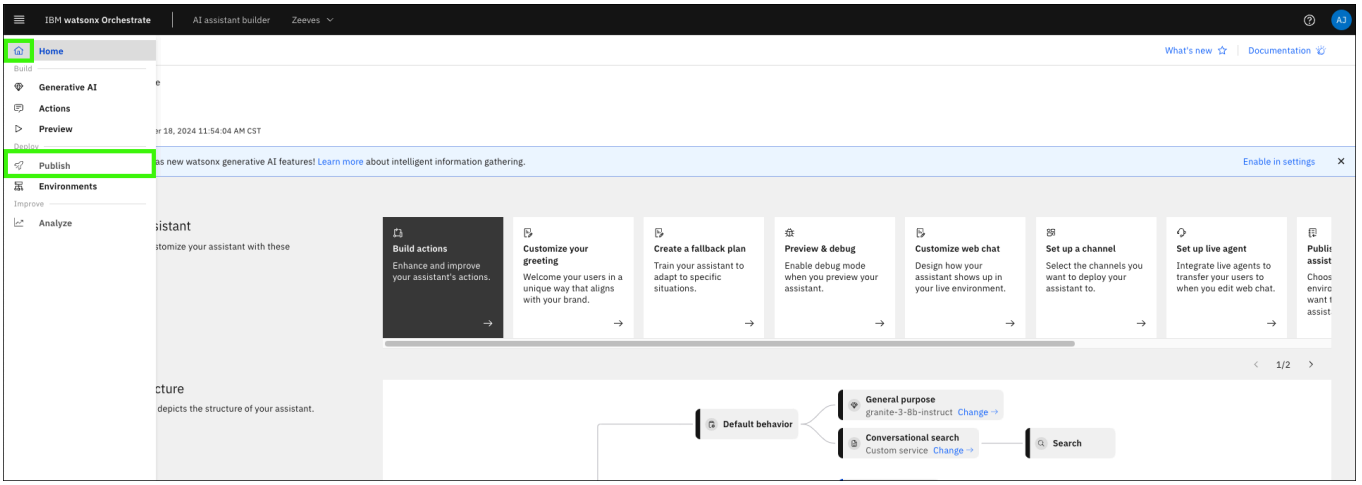
Each time that you publish, you're creating a new version of the assistant, for example *V1*. When you publish your content, you're creating a snapshot of the draft content, resulting in a version.

> **Versions do not contain integration configurations or environment settings**
>
> Published versions contain all of the content from actions, including settings and variables. **However, versions do not contain integration configurations or environment settings.** Integration configurations and environment settings must be configured manually in each environment.

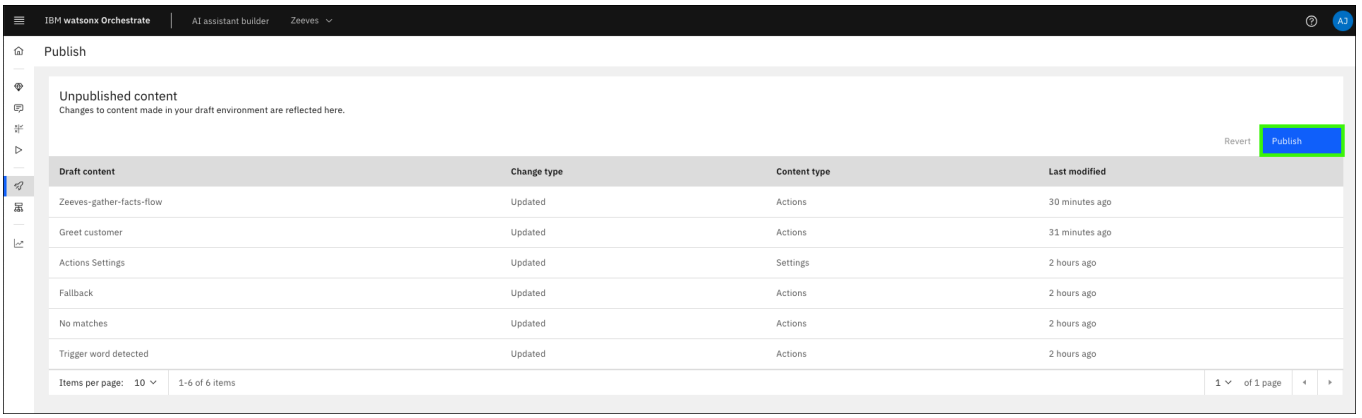
For managing quality-control and versioning, the Live environment is the version of the assistant you should be giving access to the end-user.

Follow these steps to publish the first version of your assistant using Assistant Builder:

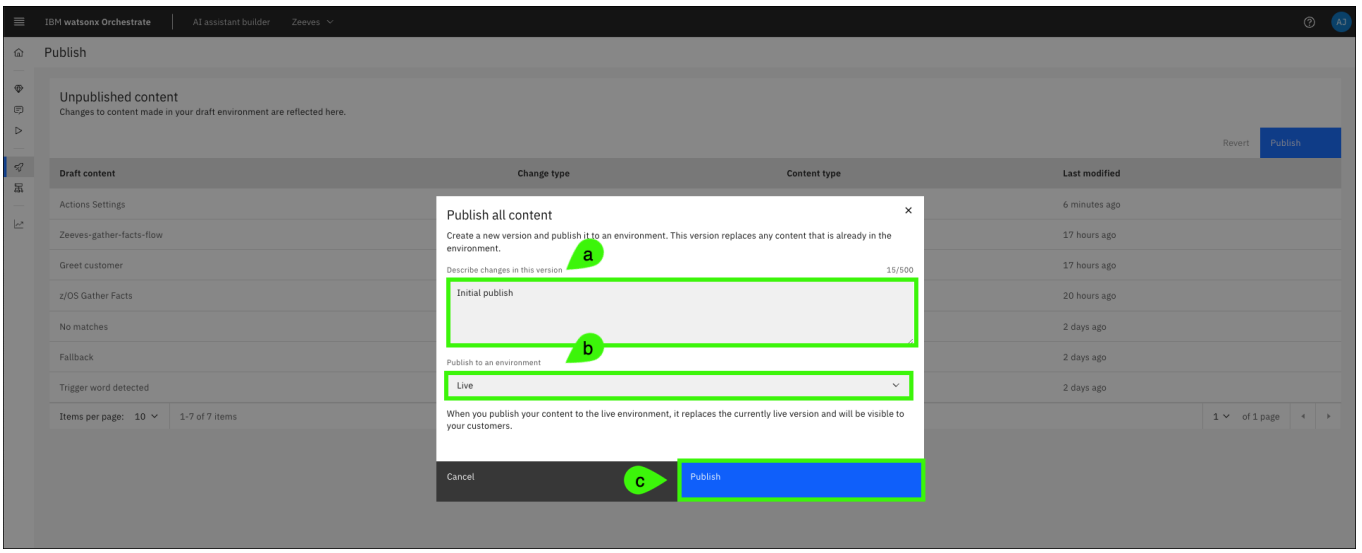
1. Hover over the **Home** icon ( ) and click **Publish**.



2. Click **Publish**.



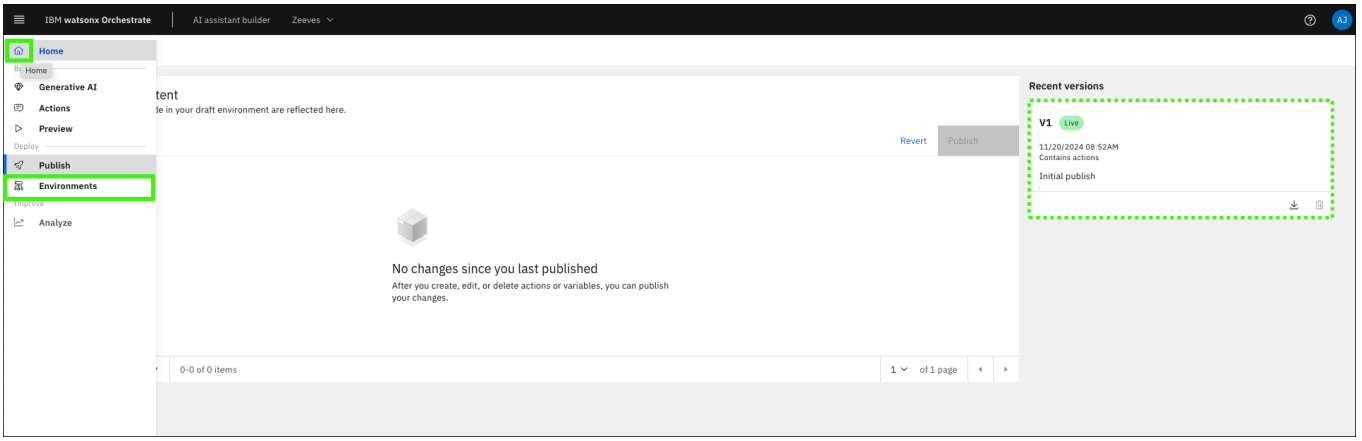
3. Enter a description of the changes (**a**), set the environment to **Live** (**b**), and then click **Publish** (**c**).



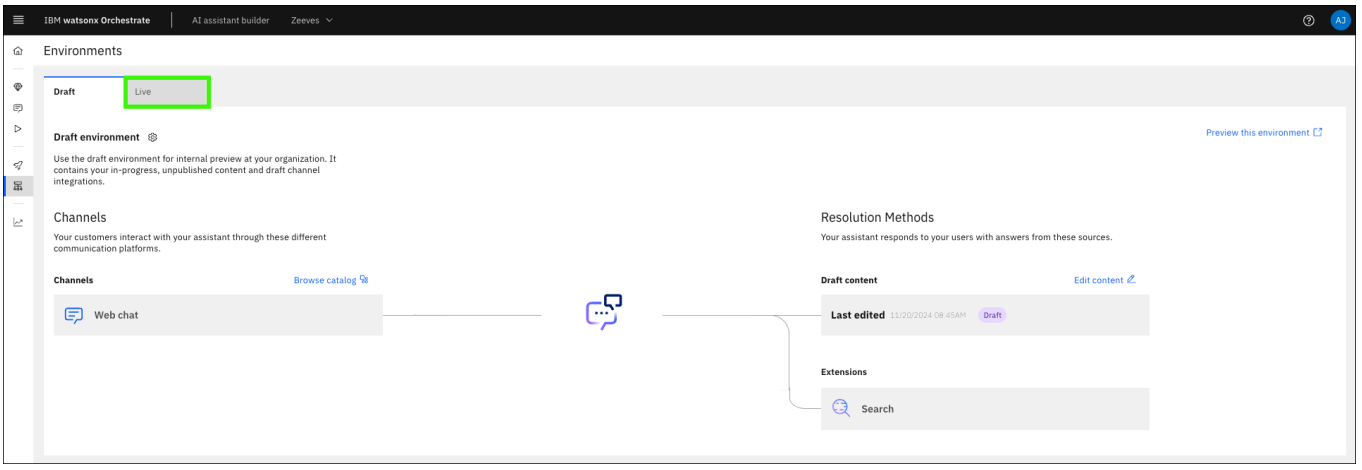
Configure the live environment

Important: When the live environment is created, the environment settings from the draft environment are not carried over (for example the configuration of the OpenSearch instance used for conversational search).

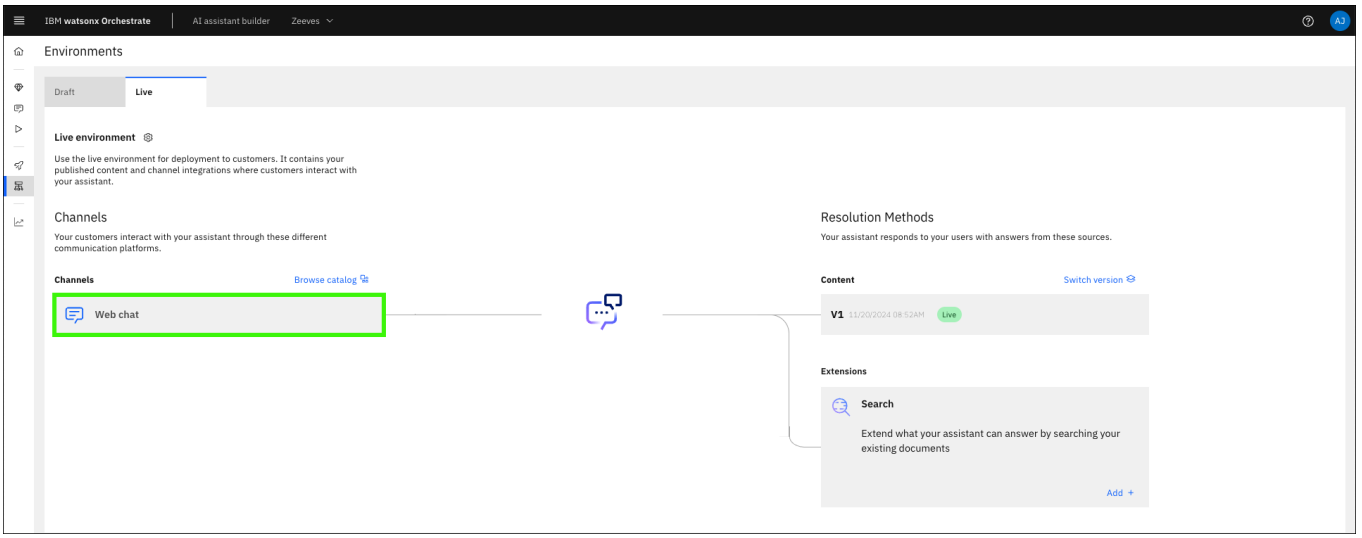
1. Hover over the **Home** icon ( ) and click **Environments**.



2. Click **Live**.



3. Click **Web chat**.



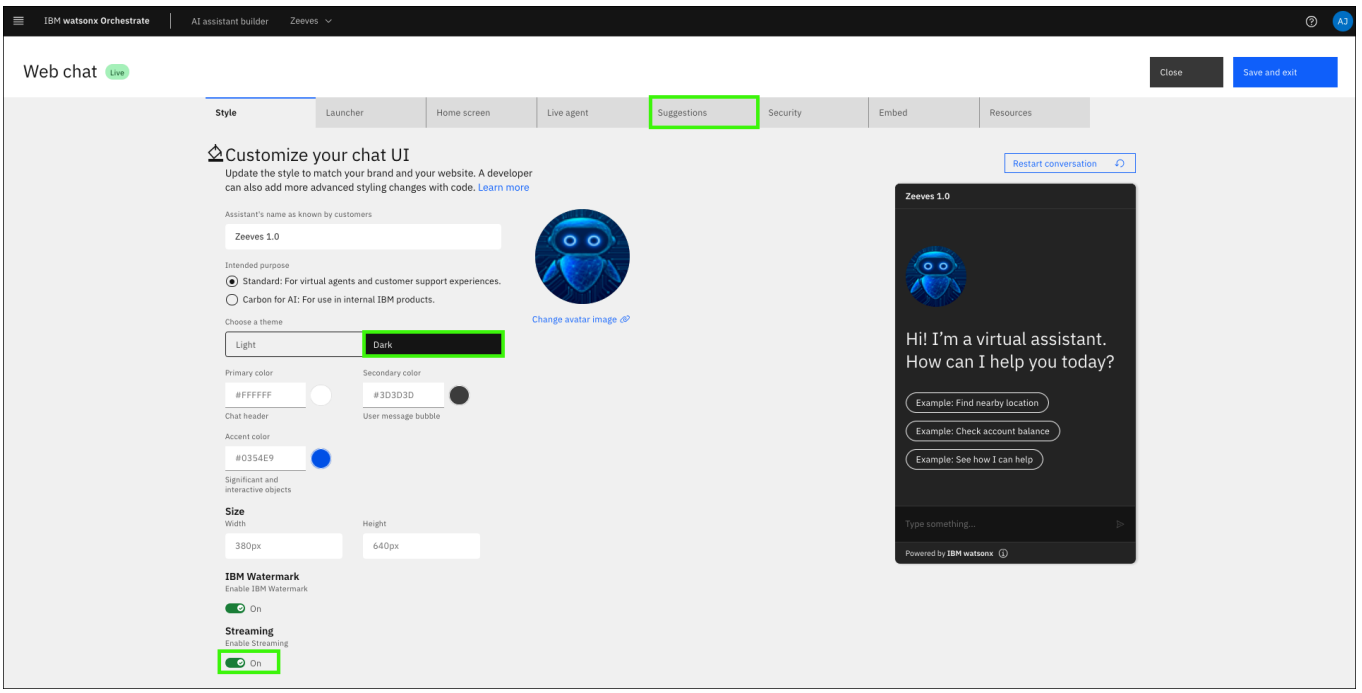
4. Customize the live assistant as you see fit.

   On the **Style** tab, you're able to set the Assistant name which will be displayed at the top of the chat window when end-users are interacting with the assistant. For pilots or demos, you may want to personalize this name for the client. Also in the **Style** tab, you have the ability set the themes and display settings of the chat windows, including the ability to enable the IBM Watermark and enable streaming (recommended).

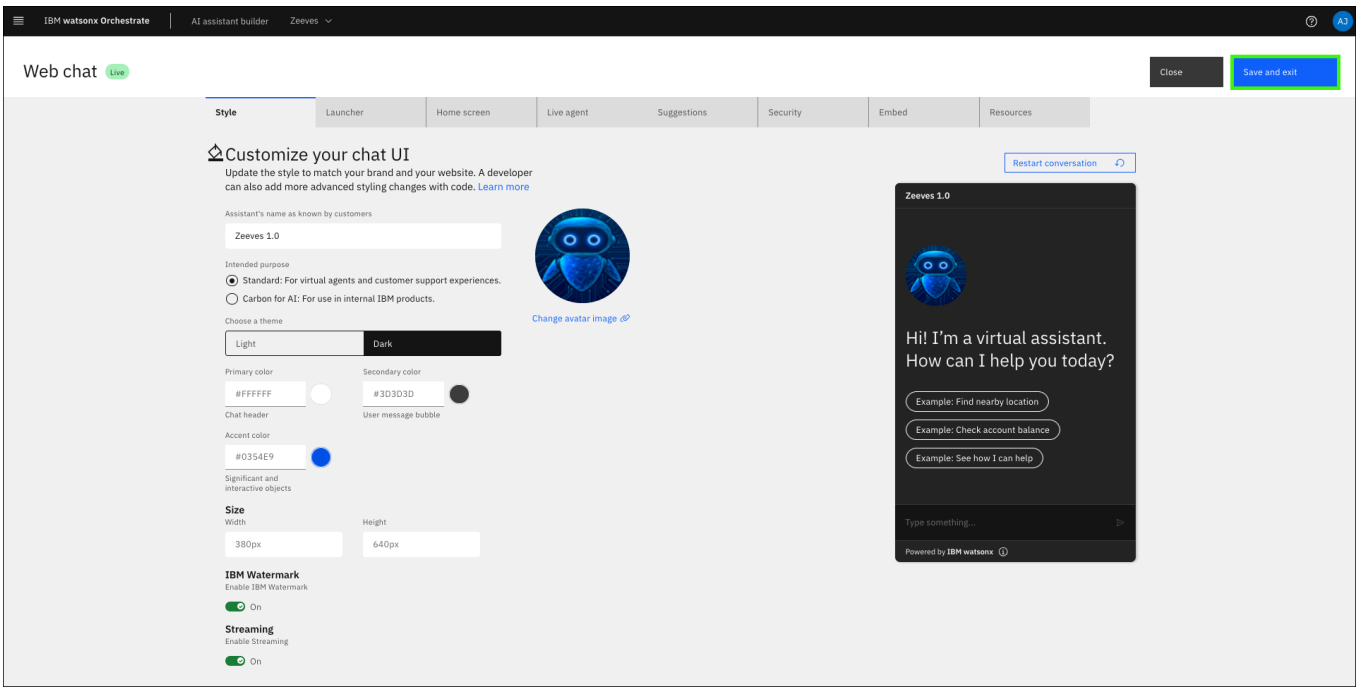
   On the **Home** tab, you enable and customize a default greeting message from the assistant when the user accesses the assistant chat. You're also able to set Conversation starters that will be displayed in the chat window. When selected by the end-user, the text of these conversation starters are sent as prompts, so it is important that your assistant is trained and tested to answer appropriately. It is highly recommended to remove these default conversation starters and to consider creating your own as long as they're able to be executed as actions and provide value to the end-user. At the bottom of the Home screen tab, you will also see the ability to add a Background style for the assistant chat window.

   Explore all the other tabs.

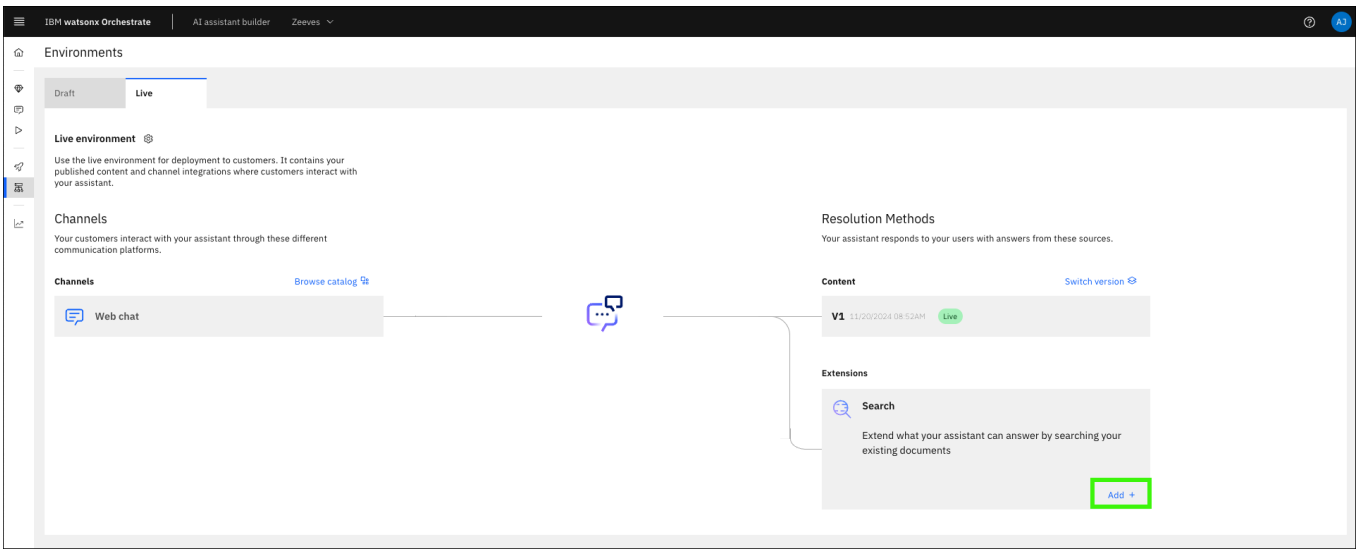
> **Customize your live environment.**
>
> For this lab, toggle **Streaming** on and turn **Suggestions** off on the **Suggestions** tab. You may also want to change the theme to **Dark** to differentiate your draft and live environments.



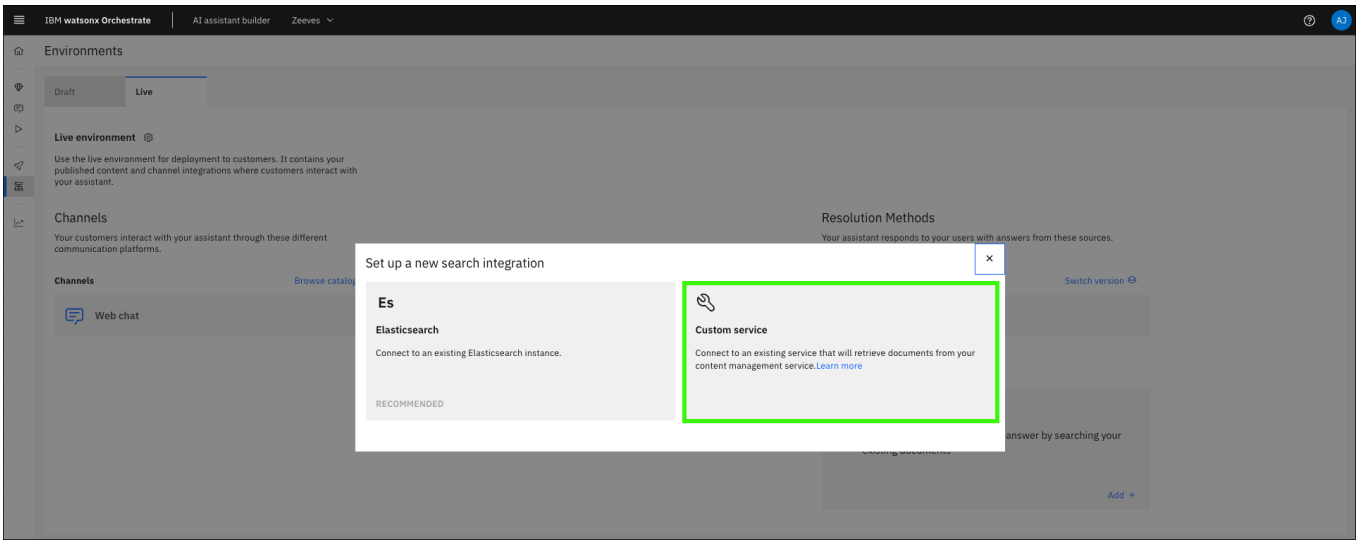
5. Click **Save and exit**.



6. Click **Add** in the **Search** tile.

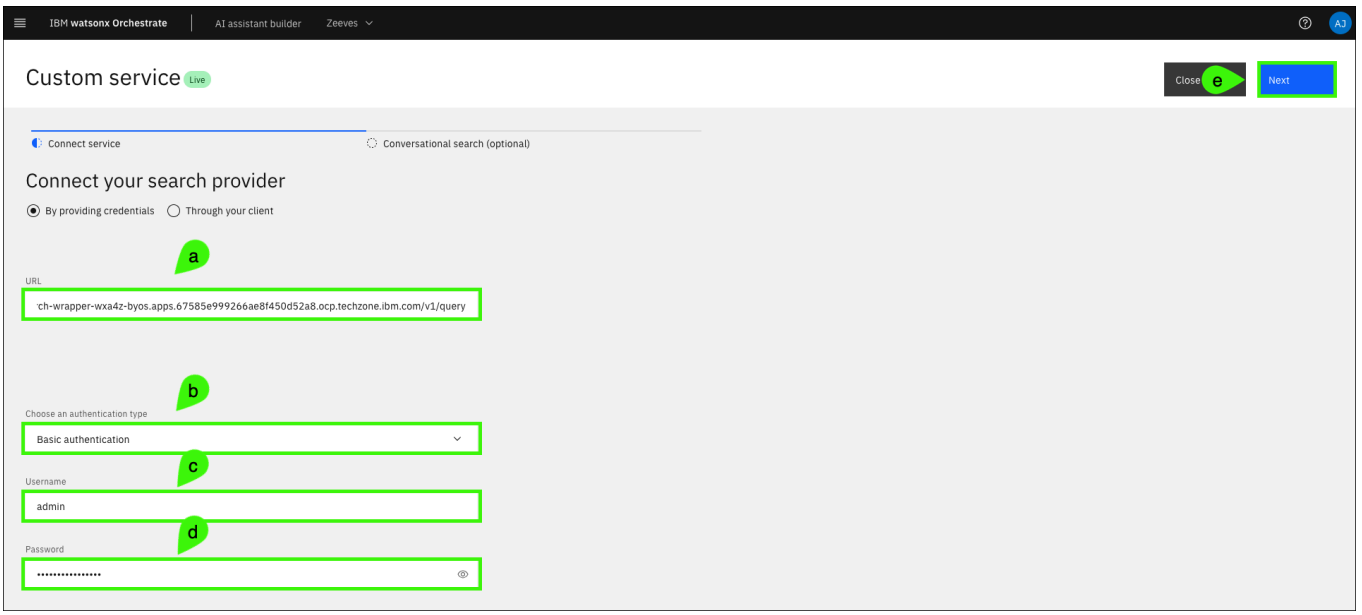


7. Click **Custom service**.

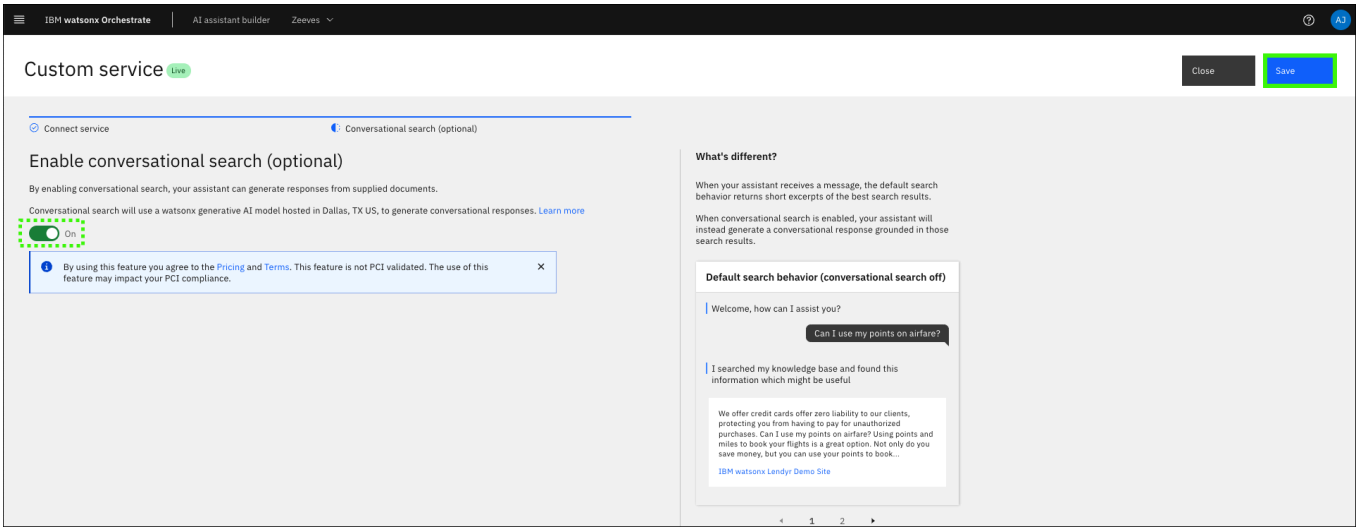


8. Enter the URL for your bring-your-own-search (BYOS) engine (**a**), select **Basic authentication** for the **authentication type** (**b**), enter **admin** for the **Username** (**c**), enter the password you specified in the `wrapper-creds.yaml` file for the **Password** (**d**), and then click **Next** (**e**).

> ⚠ **Use the correct URL and authentication type!**
>
> Use the URL and credentials for your BYOS OpenSearch engine created earlier here.



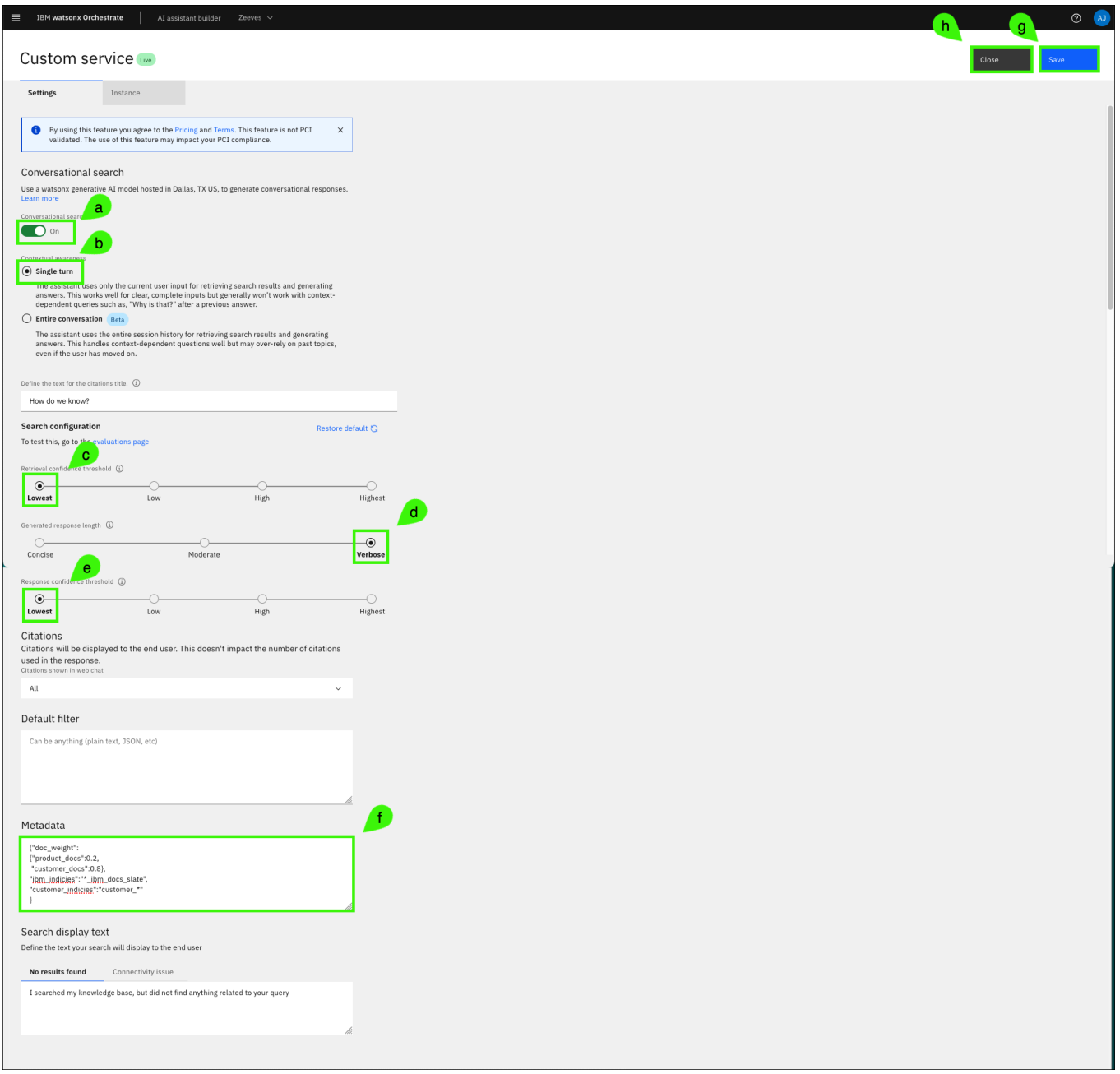
9. Verify **conversational search** is enabled and click **Save**.



10. Update the **Custom service** settings (**a-f**), click **Save** (**g**), and then click **Close** (**h**).

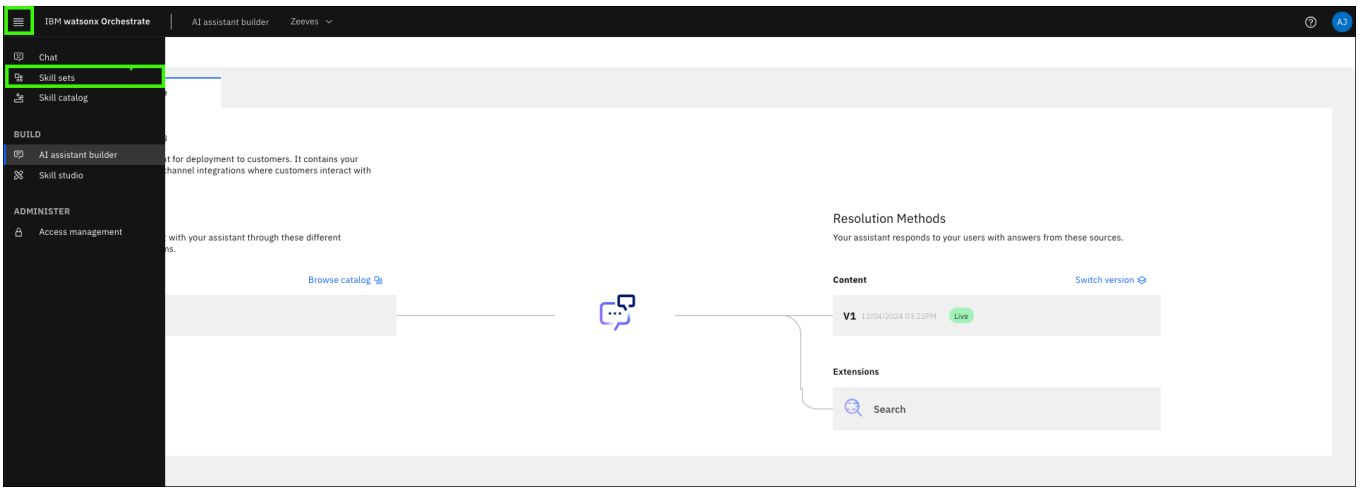
> **Feel free to customize the settings.**
>
> This is your assistant. Feel free to customize the settings. The settings shown below reflect the changes made earlier in the lab guide to the draft version of the assistant. This includes the **Metadata** field to weigh ingested client documents higher using:

```
{"doc_weight":
{"product_docs":0.2,
"customer_docs":0.8},
"ibm_indices":"*_ibm_docs_slate,*_ibm_redbooks_slate",
"standardize":true,
"customer_indices":"customer_*"
}
```

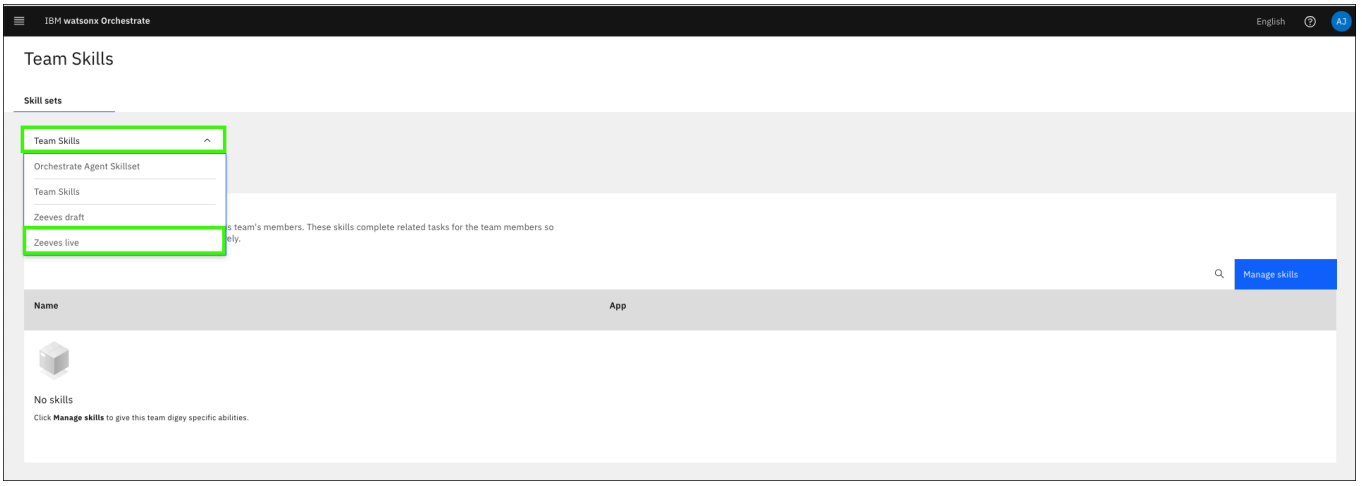


Connect the skills to the live environment

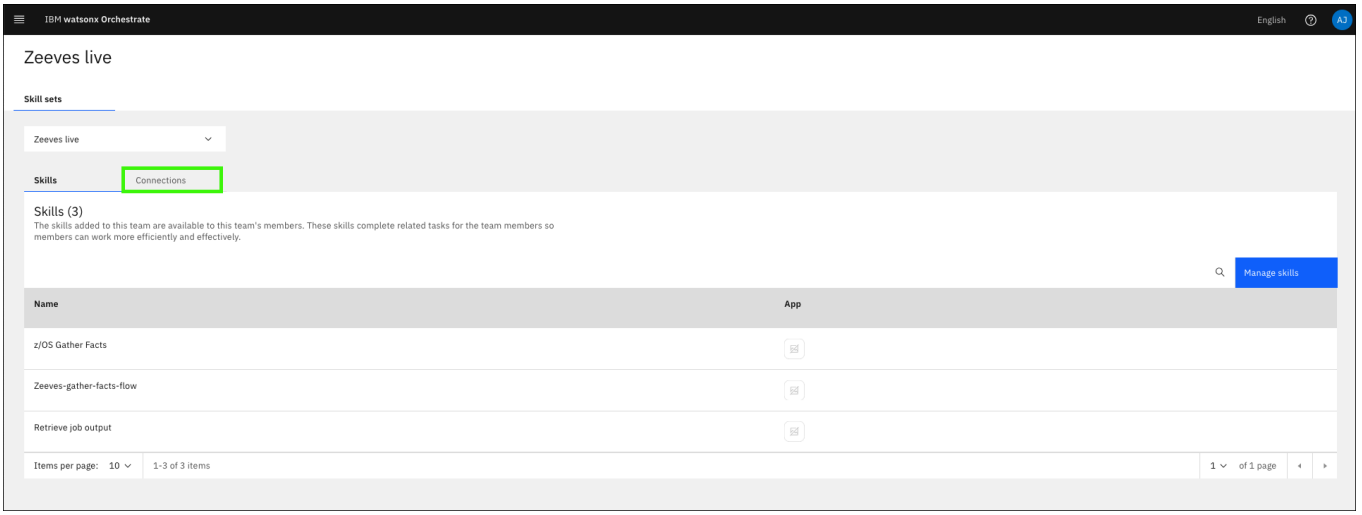
1. Click **Skill sets** in the main menu.



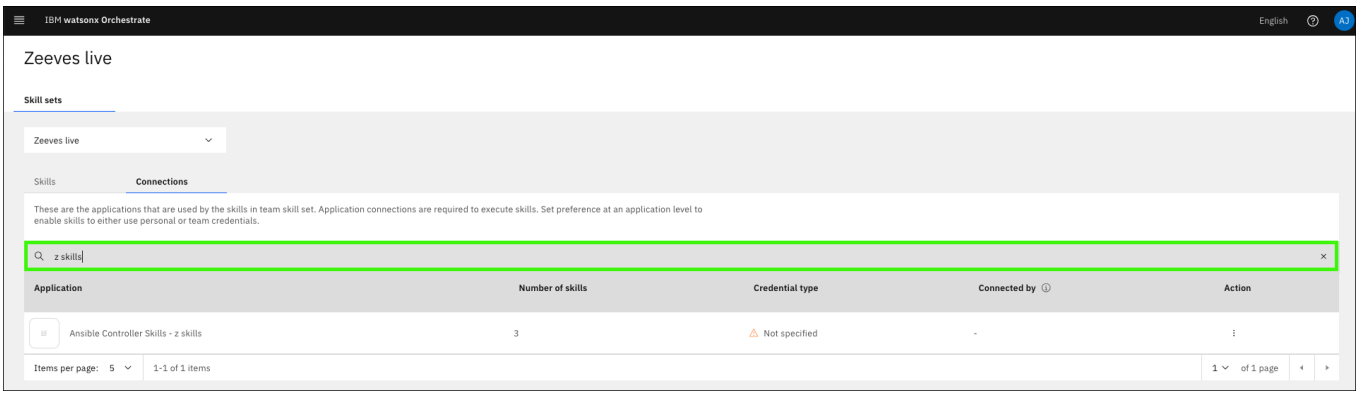
2. Select your assistant's live instance in the **Skill sets** list.



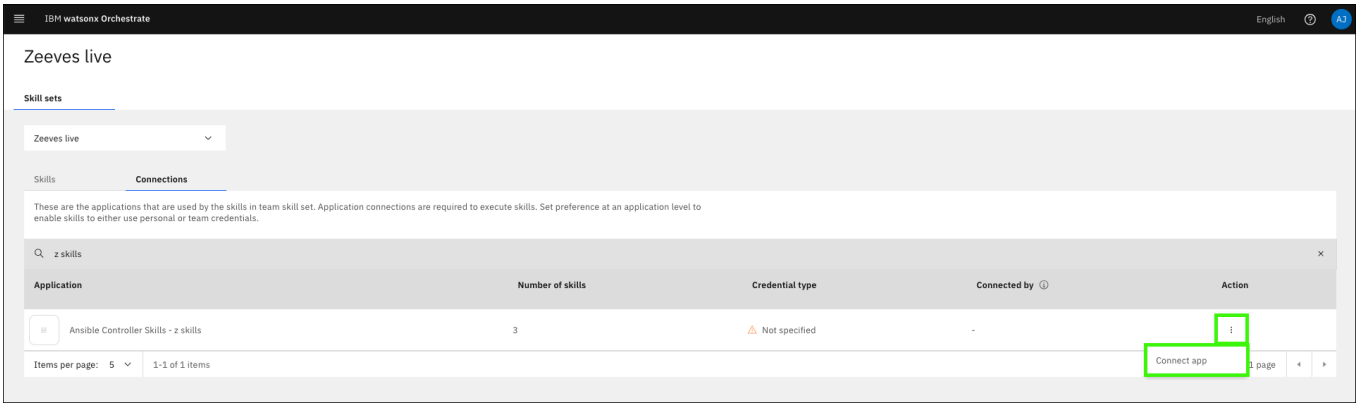
3. Click **Connections**.



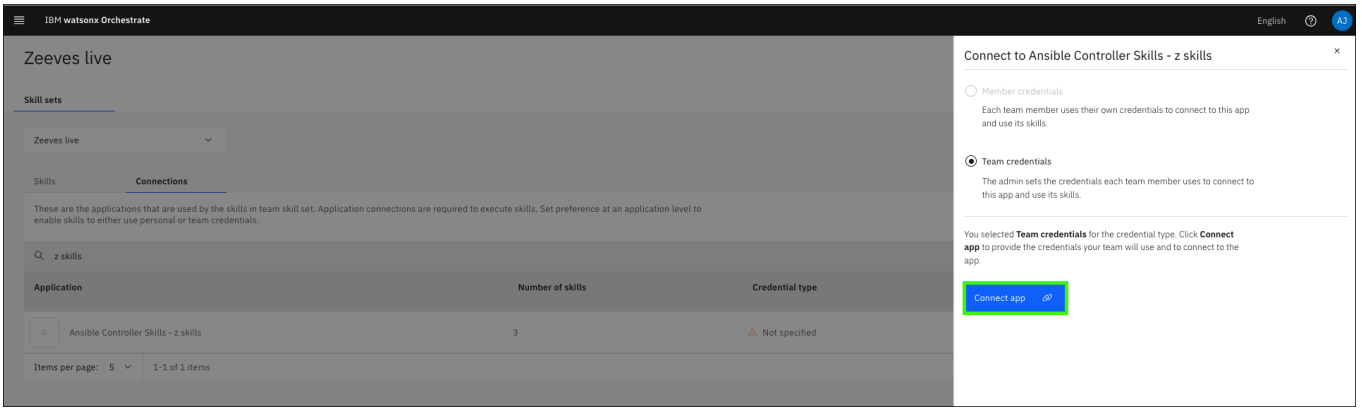
4. Search for the application name you specified earlier.



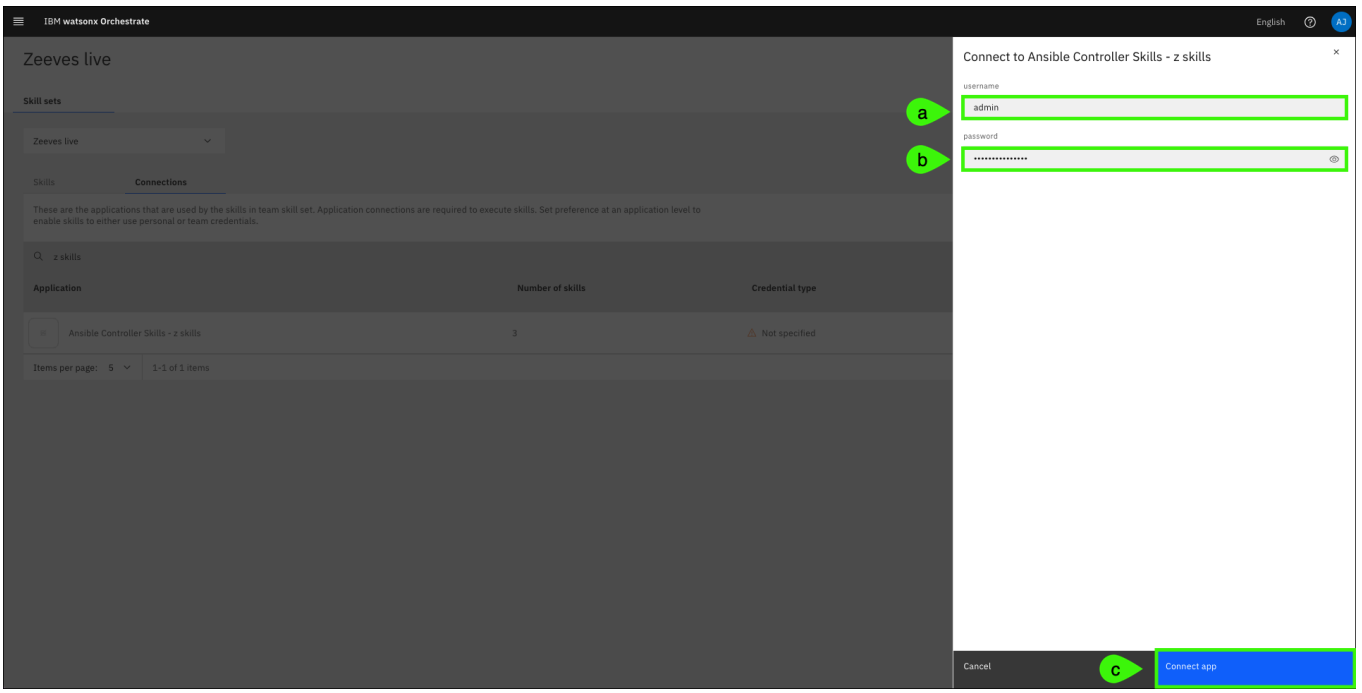
5. Click the ellipses (⋮) for your app and then click **Connect app**.



6. Click **Connect app**.



7. Enter the **username** (**a**) and **password** (**b**) using the username (`admin`) and password for your IBM Technology Zone (ITZ) *watsonx Assistant for Z Pilot - AAP & z/OS* reservation, and then click **Connect app** (**c**).



Learn more about publishing your assistant and creating live environments here.

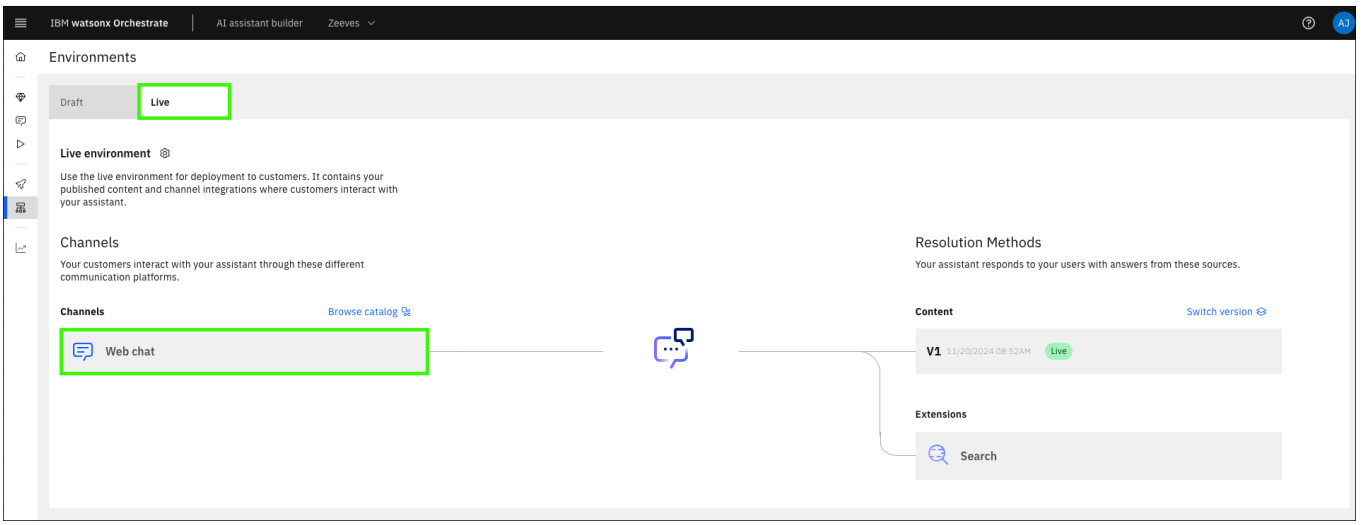
Deploy the assistant

After configuring your assistant's settings and publishing, the final step is to deploy your assistant which can be done across various channels depending on the use case.

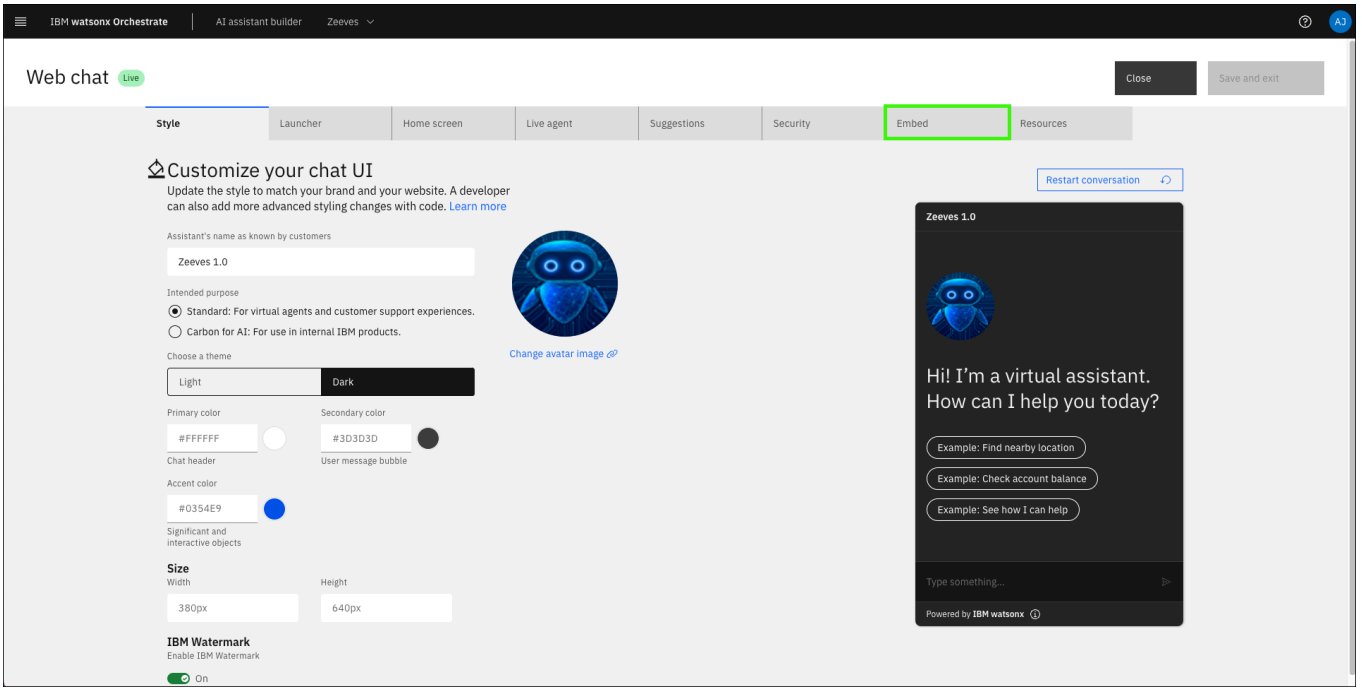
There are several options for deploying your assistant through channels and integrations to satisfy the use cases that you might address. Learn more about all the deployment options here.

For this lab, you will deploy the assistant using the web chat integration. The web chat integration provides an assistant interface that can integrate with your website. There is a lot of flexibility with how you may want to integrate it. Learn more about the web chat integration here.

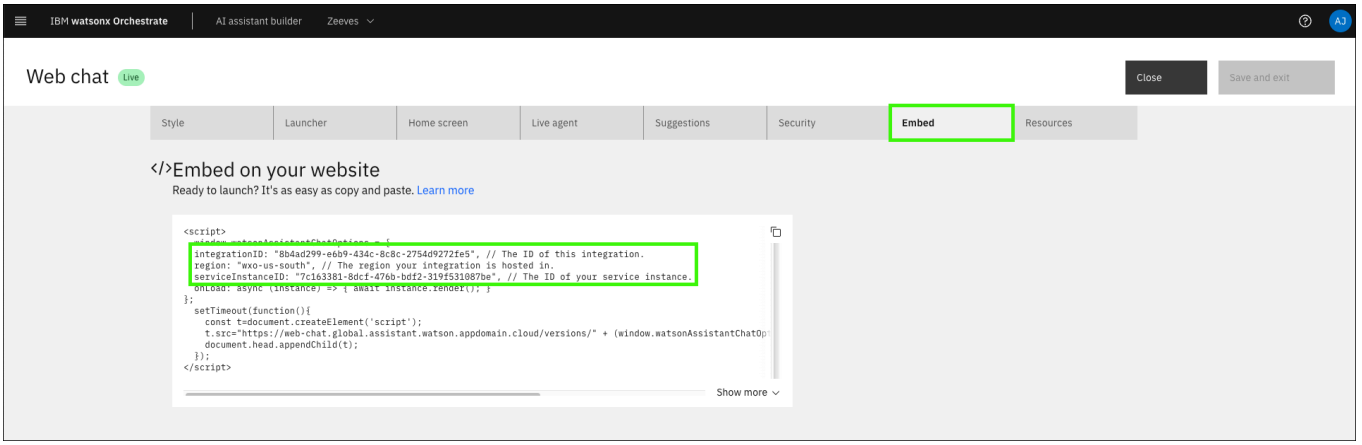
1. Open the **Environments** page in **AI assistant builder**.
2. Click **Web chat** for the **Live** environment.



3. Click the **Embed** tab.



4. Copy and record the `integrationID`, `region`, and `serviceInstanceID` values.



5. In a text editor, create a file named `Watson Assistant Chat.html` and paste the following text in the file.

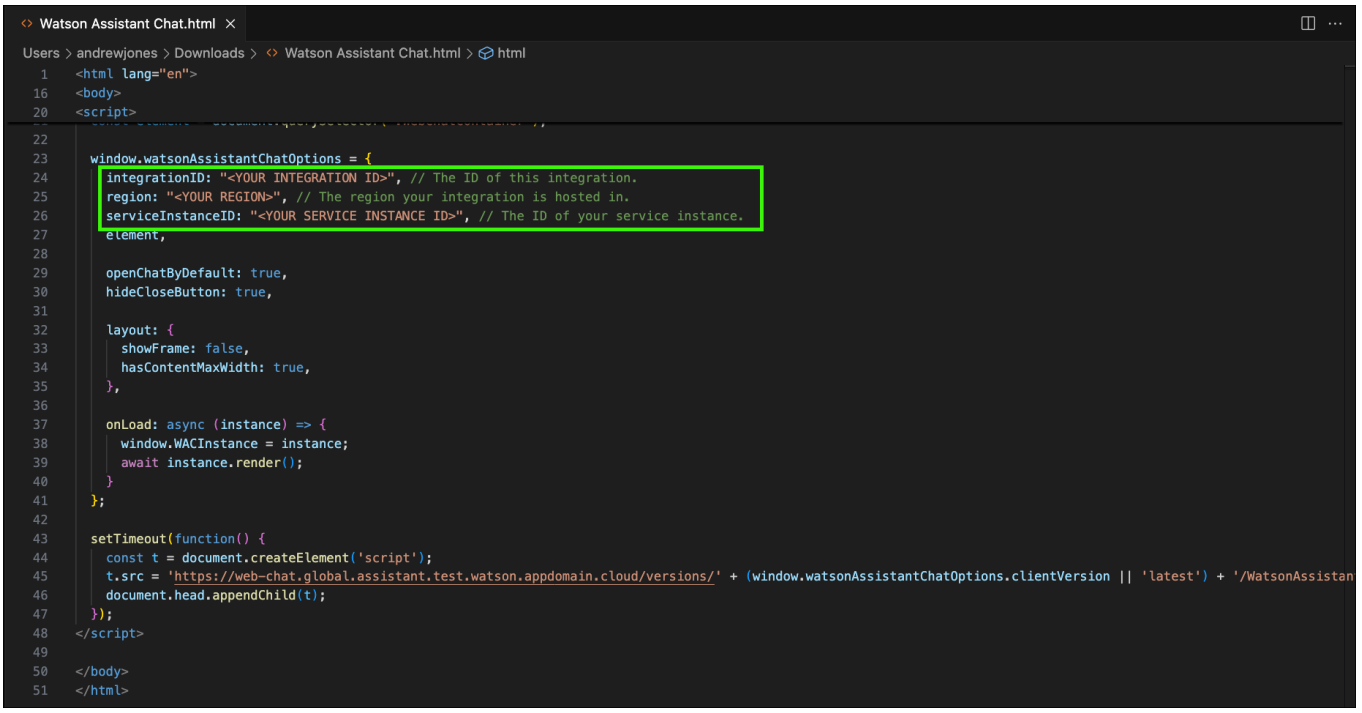
File name:

```
Watson Assistant Chat.html
```

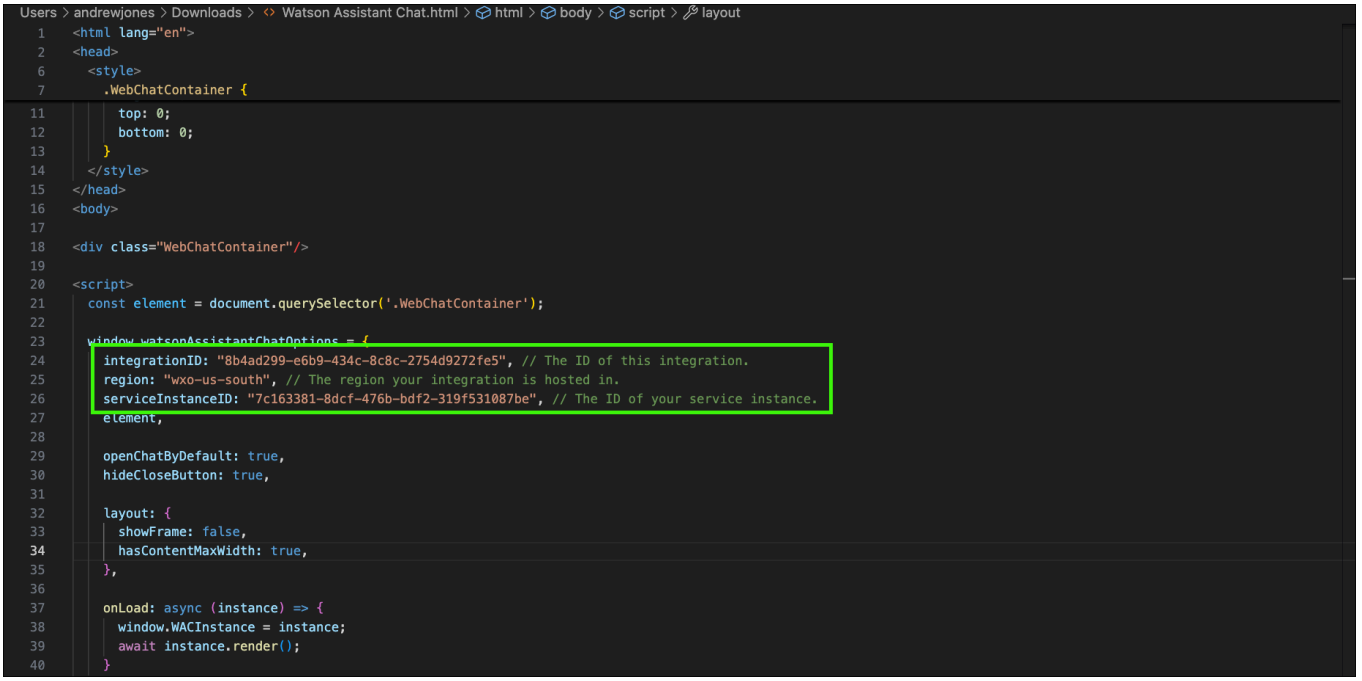
File contents:

```
<html lang="en">
<head>
<title>Watson Assistant Chat</title>
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>
  .WebChatContainer {
    position: absolute;
    left: 0;
    right: 0;
    top: 0;
    bottom: 0;
  }
</style>
</head>
<body>

<div class="WebChatContainer"/>

<script>
const element = document.querySelector('.WebChatContainer');

window.watsonAssistantChatOptions = {
    integrationID: "<YOUR INTEGRATION ID>", // The ID of this integration.
    region: "<YOUR REGION>", // The region your integration is hosted in.
    serviceInstanceID: "<YOUR SERVICE INSTANCE ID>", // The ID of your service instance.
    element,

    openChatByDefault: true,
    hideCloseButton: true,

    layout: {
      showFrame: false,
      hasContentMaxWidth: true,
    },

    onLoad: async (instance) => {
      window.WACInstance = instance;
      await instance.render();
    }
};

setTimeout(function() {
    const t = document.createElement('script');
    t.src = 'https://web-chat.global.assistant.test.watson.appdomain.cloud/versions/' +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
'/WatsonAssistantChatEntry.js';
    document.head.appendChild(t);
});
</script>

</body>
</html>
```

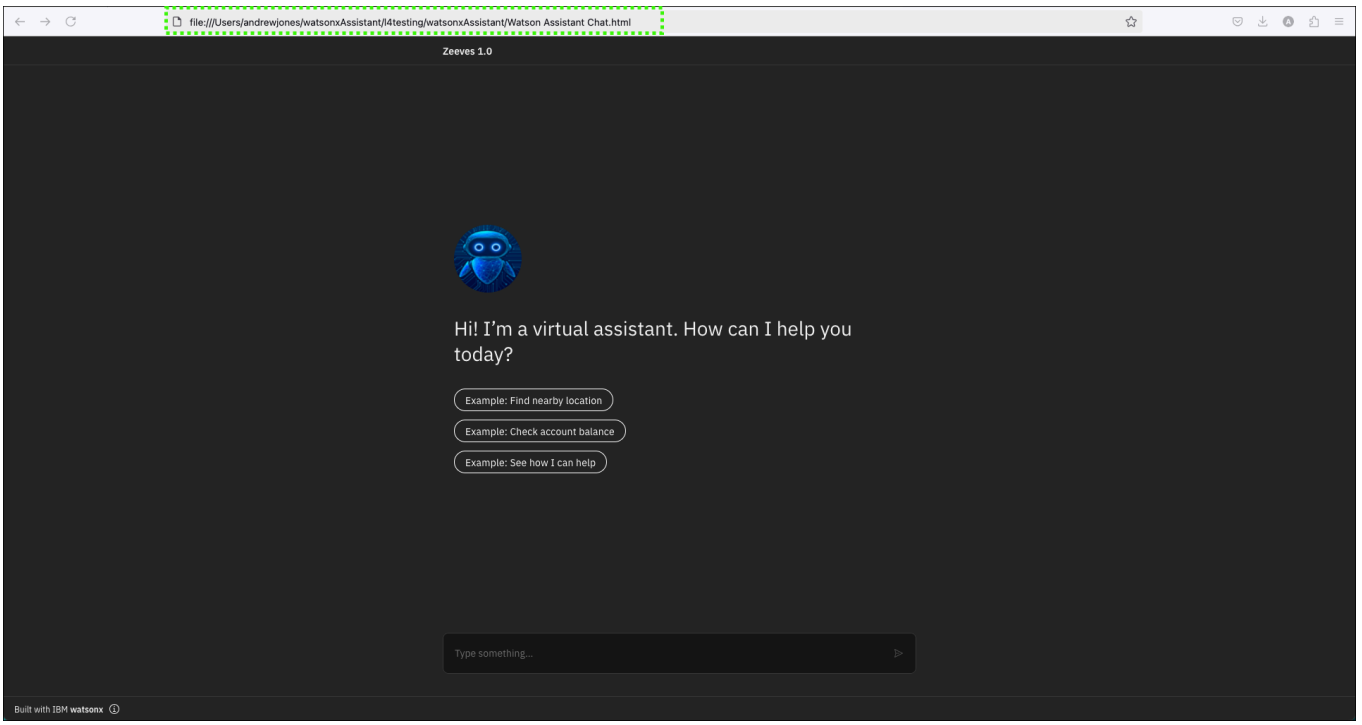
Before modification:



After modification:



6. Open the `Watson Assistant Chat.html` file in a web browser.



Your assistant is now live. Explore some of the earlier prompts to verify the assistant is accessing the ingested documents and your skills and skill flows are active.

> ⚠ **Wait 5-10 seconds before clicking apply on skill actions.**

Prompts to try:

```
What is z/OS continuous delivery?
```

```
Get z/OS facts
```

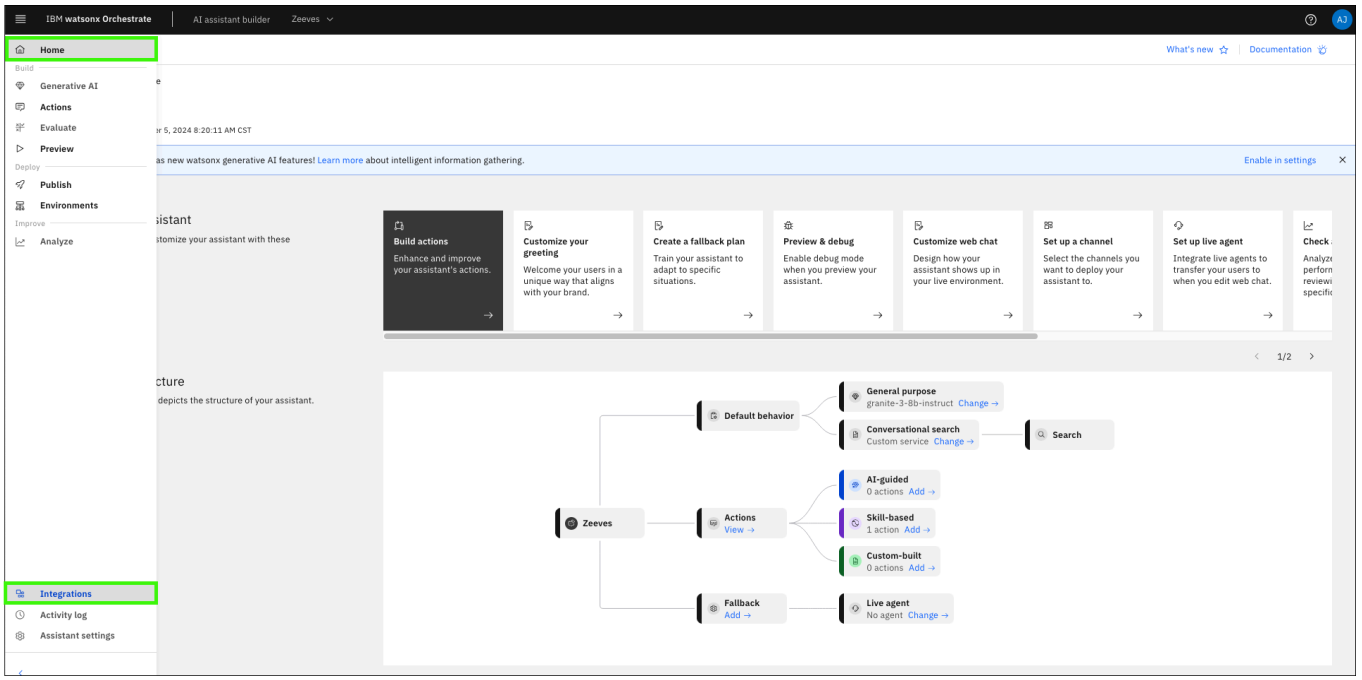
```
Show me z/OS facts
```

```
Gather and display z/OS facts
```

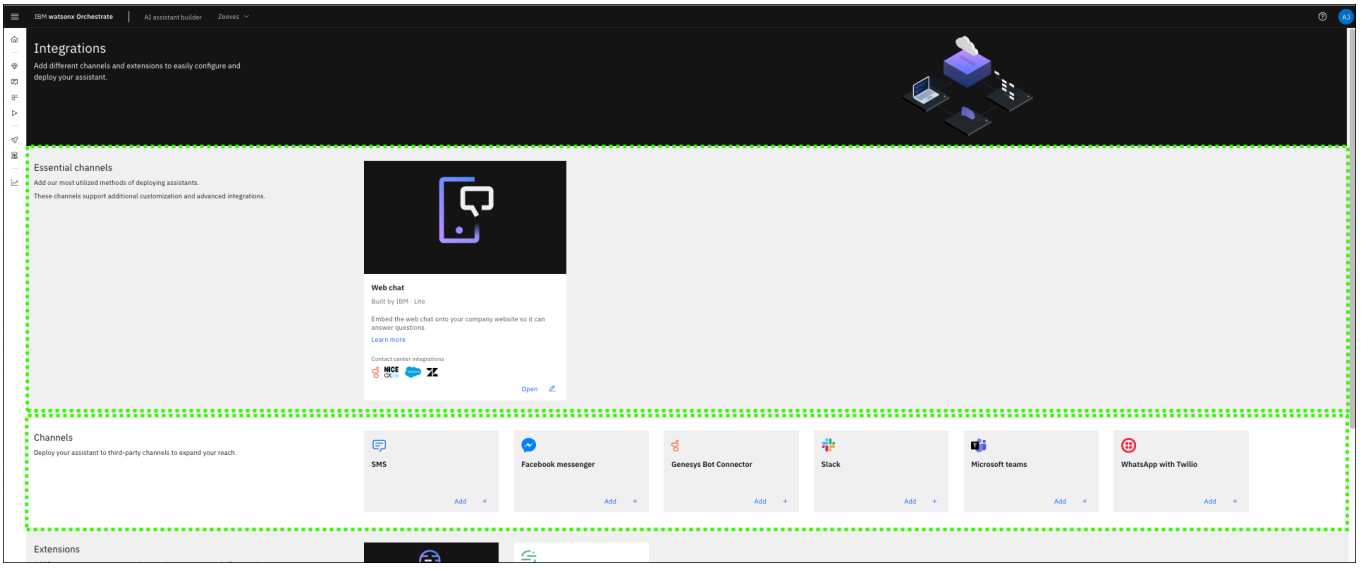
Adding other integrations

IBM watsonx Assistant for Z can integrate with other delivery channels beyond a web page. Other channels include Slack, Microsoft teams, WhatsApp, and many others. Integrating with these and other channels are not covered in the lab guide. However, follow the steps after this to find the current channels that are supported and where to get more information.

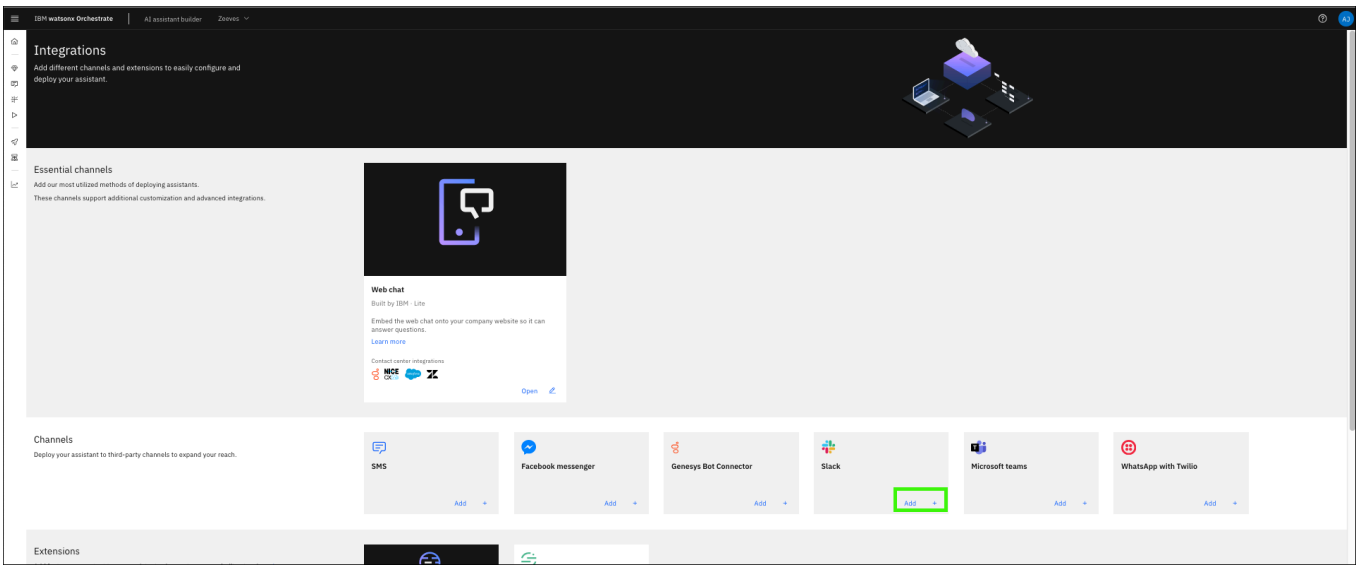
1. Hover over the **Home** ( ) and click **Integrations**.



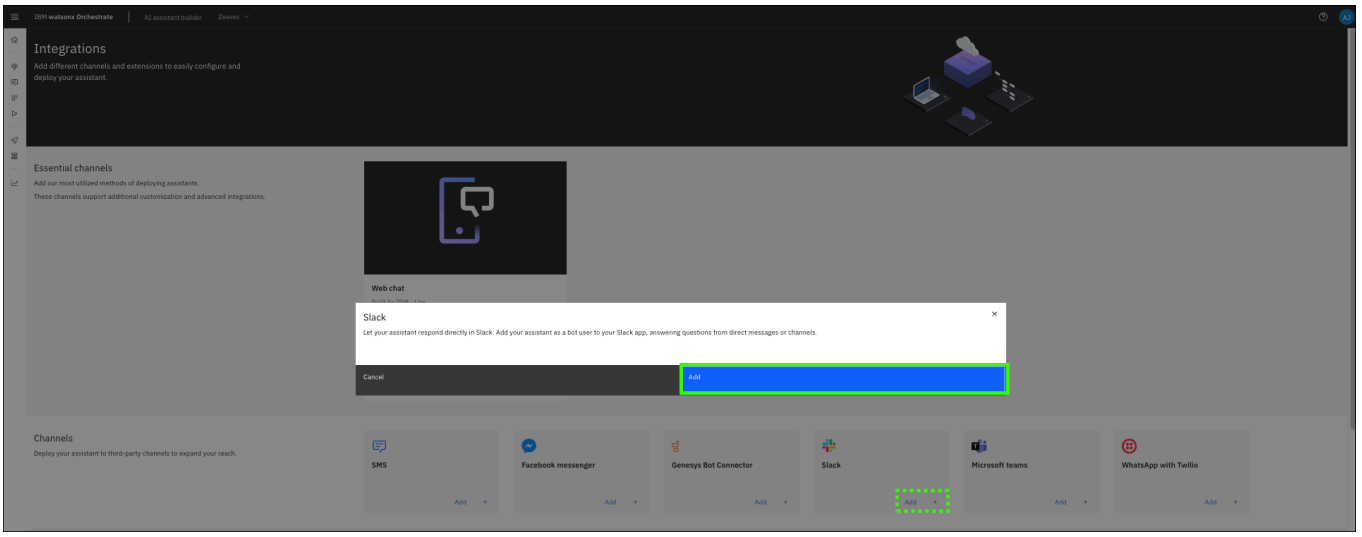
2. Explore the **Essential channels** and **Channels** sections.



3. Click **Add** on the **Slack** tile.

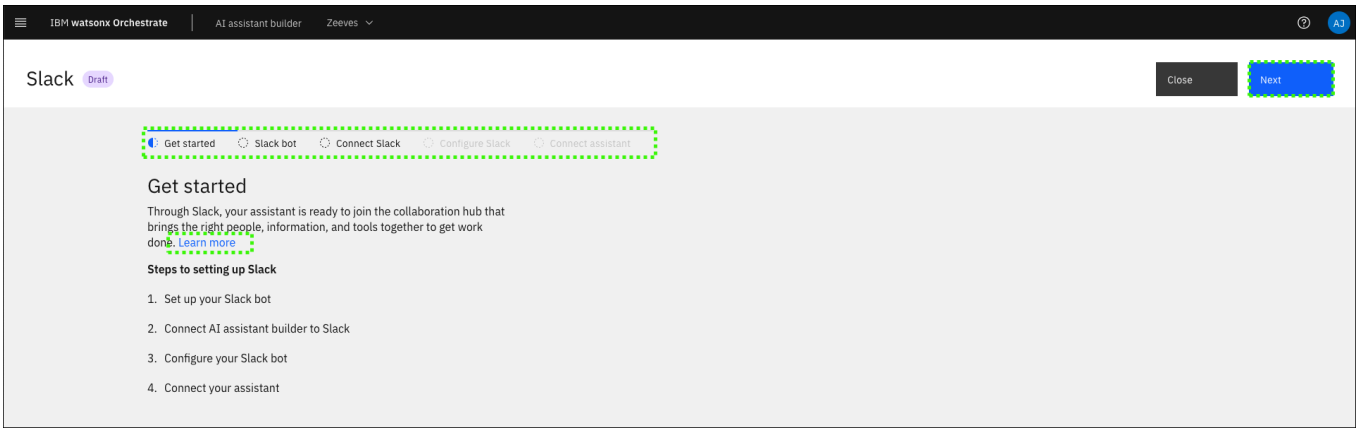


4. Click **Add**.



5. Review the step-by-step instructions and additional information available for adding a Slack integration.

   Note: Most users will not have permissions to integrate with your enterprise slack deployment as doing so requires administrative rights.



Take time to further explore the next steps for adding a Slack integration channel and the other supported integration channels.

Learn more about adding integrations here.

Next steps

This lab guide covered many of the IBM watsonx Assistant for Z capabilities and provides a good base for conducting basic client pilots. However, there is still more to learn about IBM watsonx Assistant for Z. Updates and new releases of the offering and the underlying offerings like watsonx Orchestrate rollout on a regular basis. Be sure to stay informed by bookmarking and regularly reviewing the offering landing page and the product documentation.

Additional resources

The following resources are also available; however, not all are available to business partners:

Slack channel: #watsonx-assistant-z-technical Wiki: Development team's wiki Box: wxa4z Q&A - questions with high-quality responses.

Earn the badge

Finally, remember to earn the IBM watsonx Assistant for Z Technical Sales Advanced you must complete the IBM watsonx Assistant for Z for Technical Sales Level 4 learning plan.

IBM technical sellers: Your Learning learning plan

Business Partners IBM Training learning plan