

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

Welcome

- Support
- Using the demonstration guide

IBM Technology Zone environment

- Create a reservation request
- Extend the reservation
- Join the
- Accessing the environments
- Troubleshooting

Creating an assistant and configuring conversational search

- Access the
- Create your Assistant
- Configure conversational search
- Complete the configuration
- Configure the base large language model
- Testing conversational search
- Troubleshooting

Bring your own search and documents

Creating a stand-alone OpenSearch instance for document ingestion

- Install the Red Hat OpenShift command line interface utility
- Prepare to ingest documents
 - Log in to the OpenShift cluster from your local terminal
 - Create a working directory
 - Install IBM Certificate Manager on Red Hat OpenShift
 - Install the watsonx Assistant for Z Operator (for OpenSearch)
 - Deploy required secrets and the custom bring-your-own-search (BYOSearch) resources
 - Verify all the required pods are running and get the network route to your
 - Update your assistant with the new
- Troubleshooting

Installing and using zassist to ingest client documents

- Install the zassist utility
- Ingest client documentation using zassist
- Verify that the assistant is using the ingested documents
- Adjusting the search behavior

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

- Other 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.

Using the lab guide, you will:

- Provision the lab environments
- Create an assistant and configuring conversational search
- Configure assistant settings
- Test conversational search
- Deploy a dedicated instance of OpenSearch for client document ingestion (Optional)
- Import skills for z/OS automations
- Connect apps to assistants
- Create assistant actions
- Create skill flows
- Create custom-built actions
- Import pre-packaged z/OS skills
- Publish and deploy an 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](#).



Create a reference card for storing user IDs, passwords, and links for your ITZ environments.



You will be creating and using multiple user IDs, passwords, links, and other content throughout the lab. To save time, it is strongly suggested you create a simple text file to store this data so it is readily available and you can easily cut and paste the data when needed. Here is a template to get you started.

watsonx Assistant for Z - Level 4 shortcuts:

watsonx Orchestrate

IBM Cloud account:

IBM Cloud resources: <https://cloud.ibm.com/resources>

watsonx Orchestrate URL:

Assistant name:

Assistant description:

Assistant icon: https://ibm.github.io/SalesEnablement-L4-watsonx-AssistantForZ/Setup/_attachments/Zeeves75x75.png

OpenShift

Cluster Admin Username: kubeadmin

Cluster Admin Password:

OCP Console:

IBM Cloud container entitlement key:

OS-secret password:

Client ingestion AuthKey:

Wrapper password:

Cluster domain for routes:

Ingestion route (append /v1/query):

Ansible

Ansible Automation Platform URL:

AAP User Name: admin

AAP password:

Wazi User: IBMUSER

Wazi Password:

Wazi URL:

Live Embed

inetegrationID:

region:

serviceInstanceID:

Viewing images

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



2. Click **Reserve now**.

The **Reserve now** option creates a reservation for immediate use. Optionally, schedule the reservation for a later date, like when you will be at your client's office.

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

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

- Optional, change the **Name** field for the reservation.
- Select the **Education** purpose tile.
- Enter a **Purpose description**.
- Select the region nearest your physical location in the **Preferred Geography** drop-down.
- The **End date and time** will be set to 2 days after the current date and time.
- Accept the IBM Technology Zone's terms and conditions and security policies.
- When satisfied with the parameters, click **Submit**.



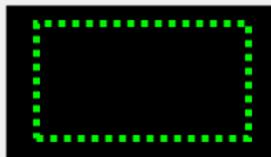
Image highlighting

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.



Guidance for delivering a demonstration



Search

IBM watsonx Assistant for Z

Welcome

Reserve the IBM Technology Zone environment

[Guidance for delivering a demonstration](#)

Demonstration scenarios >

Next steps

Specific guidance for IBM watsonx Assistant for Z and the ITZ environment

Following the scripts provided in the IBM watsonx Assistant for Z for Technical Sales Level 3 Demonstration Guide will help guarantee a successful demonstration. Use these tips to help insure success with IBM watsonx Assistant for Z and the ITZ environment:

- Follow the scripts in the IBM watsonx Assistant for X for technical sales Level 3 Demonstration Guide for the automations and skills to execute as expected.

Table of contents

General demonstration guidance

[Specific guidance for IBM watsonx Assistant for Z and the ITZ environment](#)



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:

The screenshot shows a browser window for the "IBM watsonx Assistant for Z for Technical Sales Level 3 Demonstration Guide". The title bar includes the course name, a refresh icon, and a search bar. On the left, a green dashed box highlights the Table of Contents sidebar. The sidebar lists chapters: "IBM watsonx Assistant for Z for Technical Sales Level 3 Demonstration Guide", "Welcome", "Guidance for delivering a demonstration", "Reserve the IBM Technology Zone environment", "Demonstration scenarios", "Introduction to scenarios", "Scenario 1: Authorized Program Facility", "Scenario 2: Certificate renewal", "Scenario 3: Db2 versioning", "Scenario 4: Initial Program Load on Z", "Additional IBM Z related prompts", "Summary", "Next steps", "Printing the guide", and "Instructions". The main content area displays the "Welcome" page, which introduces the course goal and provides information about its focus and additional resources.

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

The screenshot shows the same browser window after the main menu icon (≡) was clicked. The sidebar is now collapsed, and the main content area displays the "Welcome" page. The main menu icon is highlighted with a green box.

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.

The screenshot shows the 'Create a reservation' page in the IBM Technology Zone. At the top, there are tabs for 'IBM Technology Zone', 'My TechZone', and 'Help'. On the right, there are search and filter icons. The main title is 'Create a reservation' with a subtitle 'IBM watsonx Assistant for Z'. Below the title, there are four steps: 'Select a environment/infrastructure', 'Select a reservation type', 'Fill out your reservation', and 'Complete'. Under 'Single environment reservation options:', there are two radio buttons: 'Reserve now' (which is checked) and 'Schedule for later'. A 'View collection' button is located in the top right corner of the main form area.

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**.

Create a reservation

IBM Watsonx Assistant for Z

Name a

Select a reservation type b

Purpose c

Sales Opportunity number d

Preferred Geography e

End date and time f

Notes

I agree to IBM Technology Zone's [Terms & Conditions](#) and [End User Security Policies](#) g

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

Enter date and time | Select a date | Select a time | 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.

Select a date: 11/01/2024 | Select a time: 11:51 AM | America/Chicago

OCP/Kubernetes Cluster Network: 10.128.0.0/14 | Enable FIPS Security: No

Master Single Node Flavor: 16 vCPU x 64 GB - 300 GB ephemeral storage | OpenShift Version: 4.14

OCP/Kubernetes Service Network: 172.30.0.0/16

Notes: Enter any notes you would like to attach to this reservation

I agree to IBM Technology Zone's [Terms & Conditions](#) and [End User Security Policies](#)

Submit

h

i

j

k

l

m

n

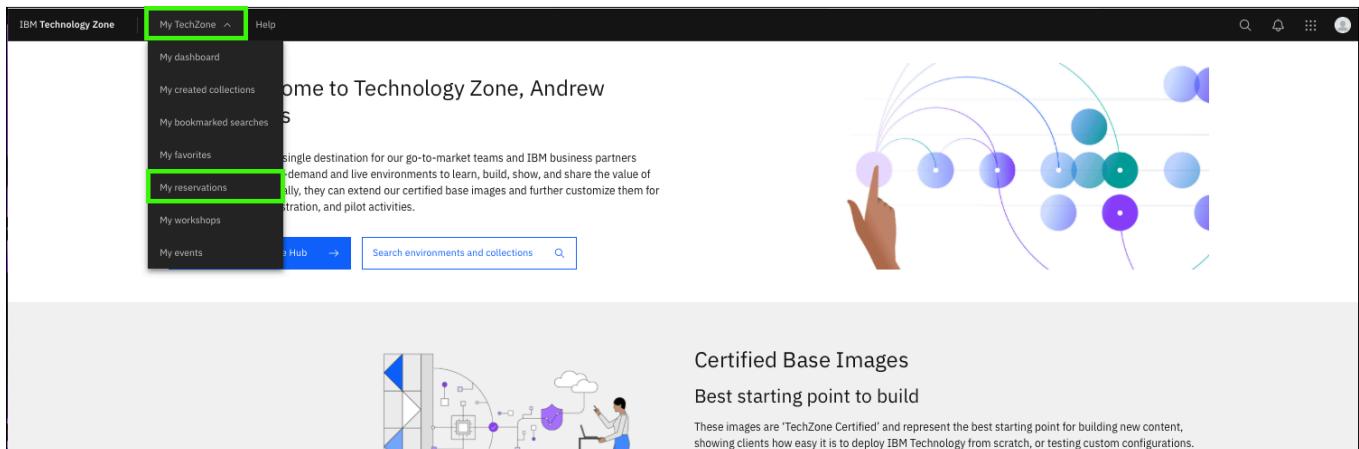
During the provisioning process, multiple emails are sent to you from ITZ as the provisioning process runs. One email states the reservation is provisioning and the other email states that the environment is **Ready**.

In rare cases, the provisioning process can 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

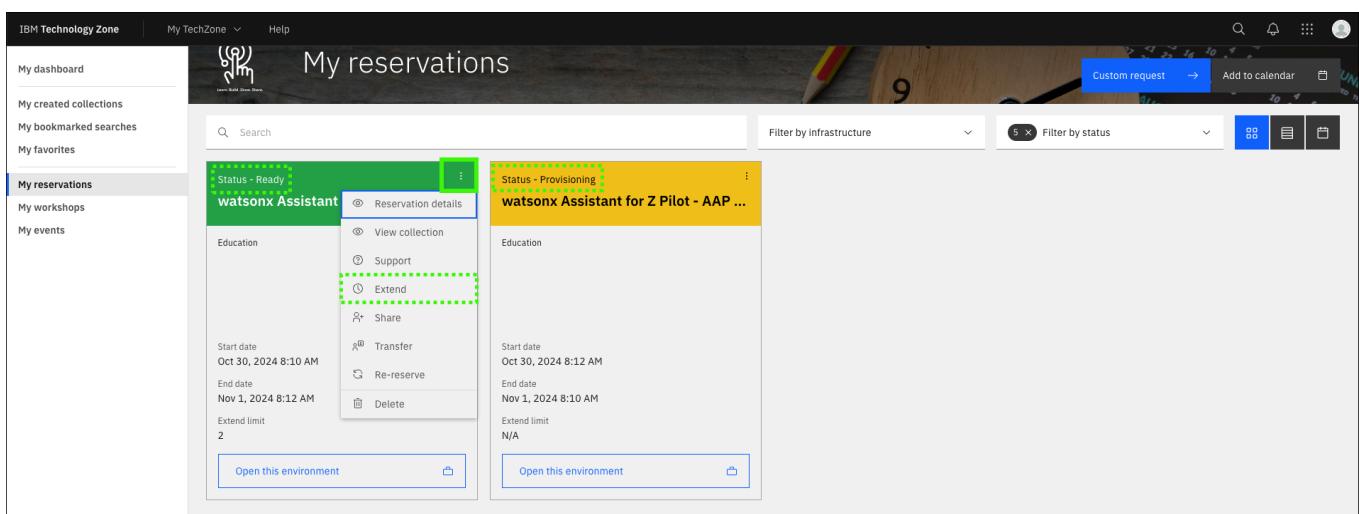
When the reservations are in the **Ready** state, you can extend each reservation beyond its original end date. The duration of the extension will vary by reservation.

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



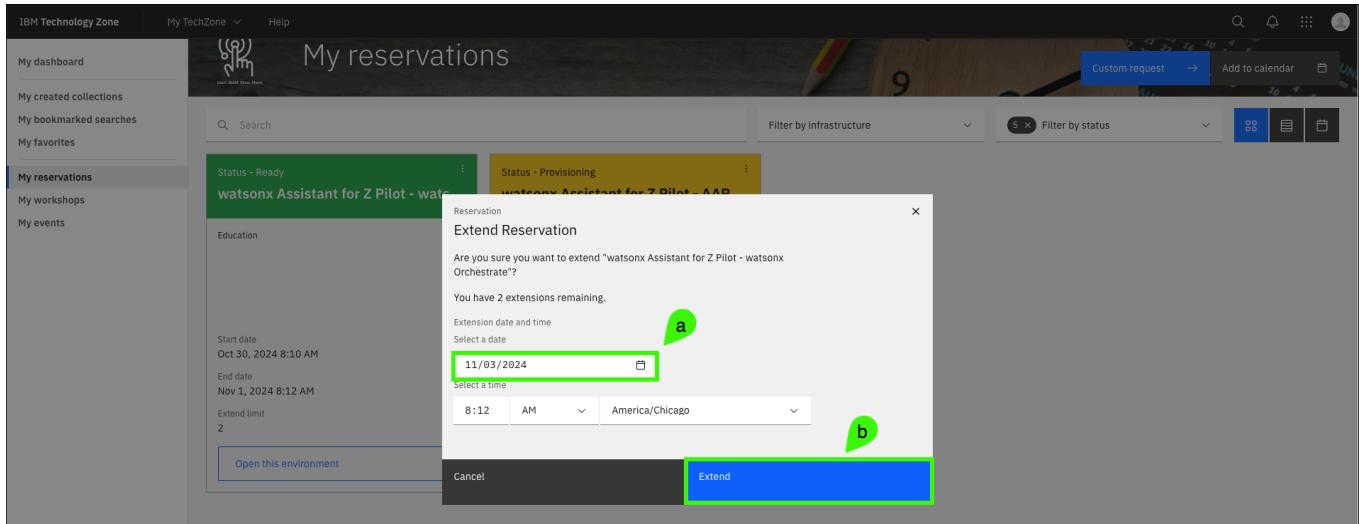
The screenshot shows the IBM Technology Zone portal homepage. The left sidebar has a dark theme with white text. The 'My TechZone' dropdown is expanded, showing options like 'My dashboard', 'My created collections', 'My bookmarked searches', 'My favorites', 'My reservations' (which is highlighted with a green box), 'My workshops', and 'My events'. The main content area has a light gray background. It features a welcome message 'Welcome to Technology Zone, Andrew' and a section titled 'Certified Base Images' with the subtext 'Best starting point to build'. There's also a decorative graphic of a hand pointing at a cluster of colored circles.

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



The screenshot shows the 'My reservations' page. The left sidebar is identical to the previous screenshot. The main area displays two reservation cards. The first card is for 'watsonx Assistant' and has a green border around its status indicator 'Status - Ready'. The second card is for 'watsonx Assistant for Z Pilot' and has a yellow border around its status indicator 'Status - Provisioning'. Both cards show basic details like start date, end date, and extend limit. Below each card is a blue button labeled 'Open this environment'. On the right side of the page, there are filters for 'Filter by infrastructure' and 'Filter by status', along with several icons for search, calendar, and other functions.

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



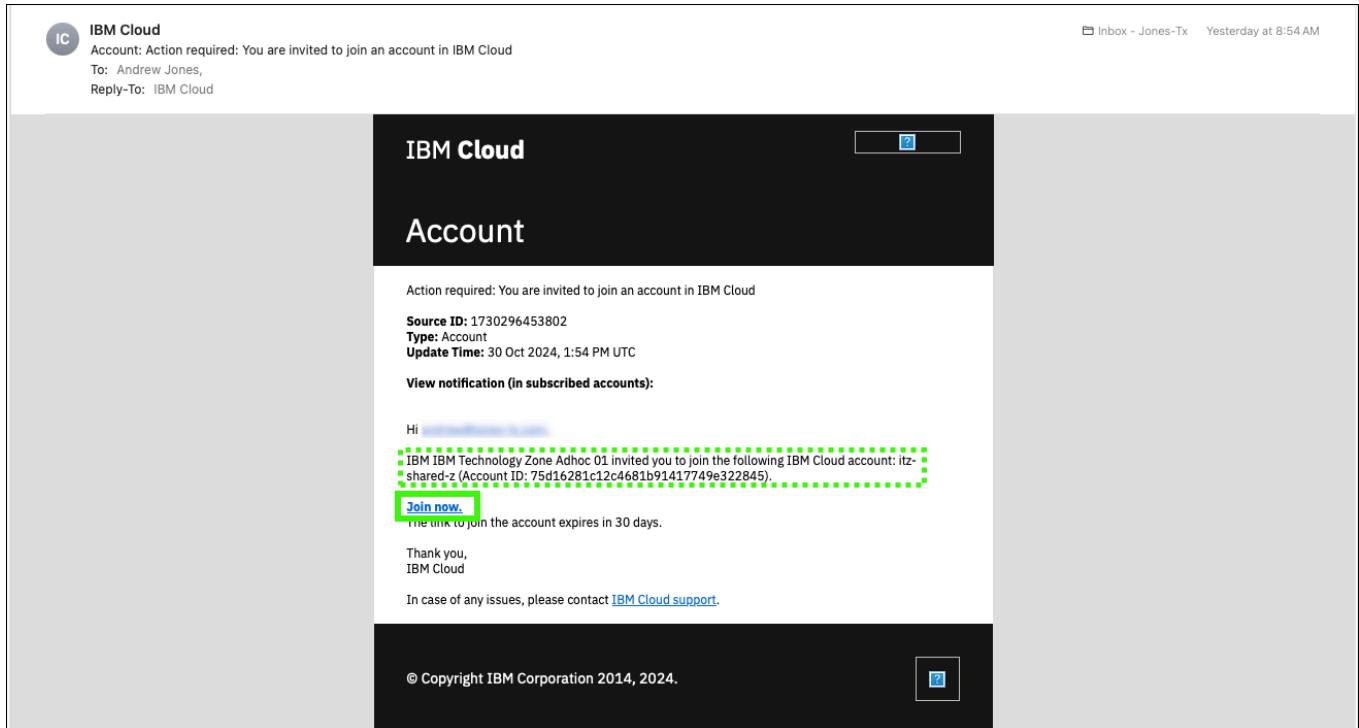
If you anticipate needing more time, repeat Steps 5-6 to extend the reservation to the maximum allowed. Repeat these steps for the other two reservations.

Join the ITZ IBM Cloud account

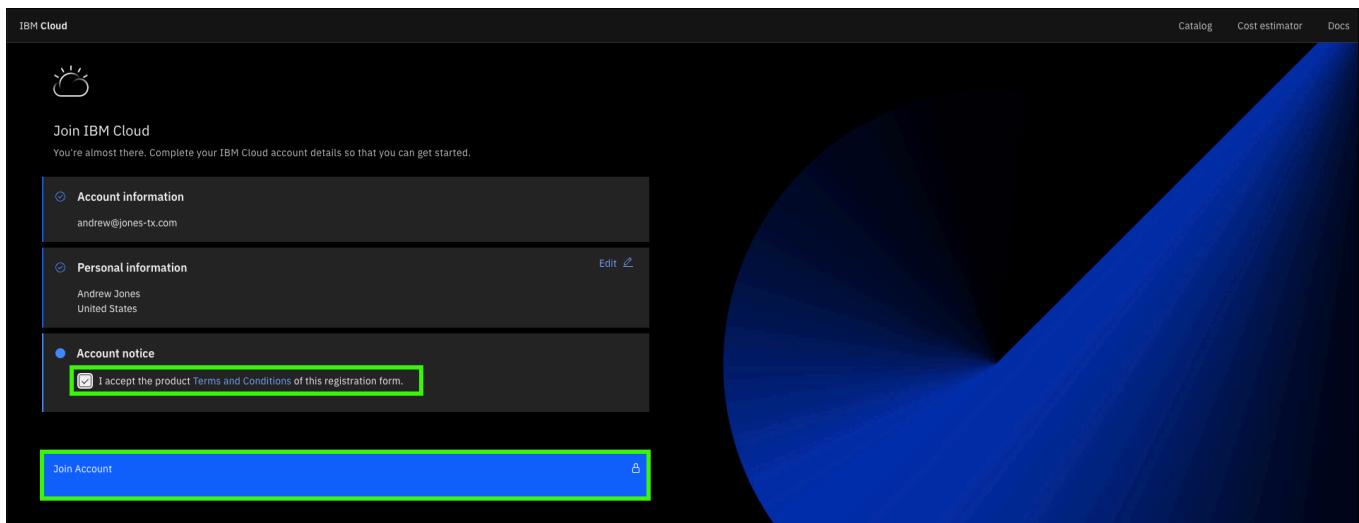
Both the **watsonx Assistant for Z lab – watsonx Orchestrate** and the **Ansible Automation Platform (AAP) & z/OS** environments add you to an IBM Cloud account while your reservation is active. During the provisioning process of these ITZ environments, you receive two emails from IBM Cloud.

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**.

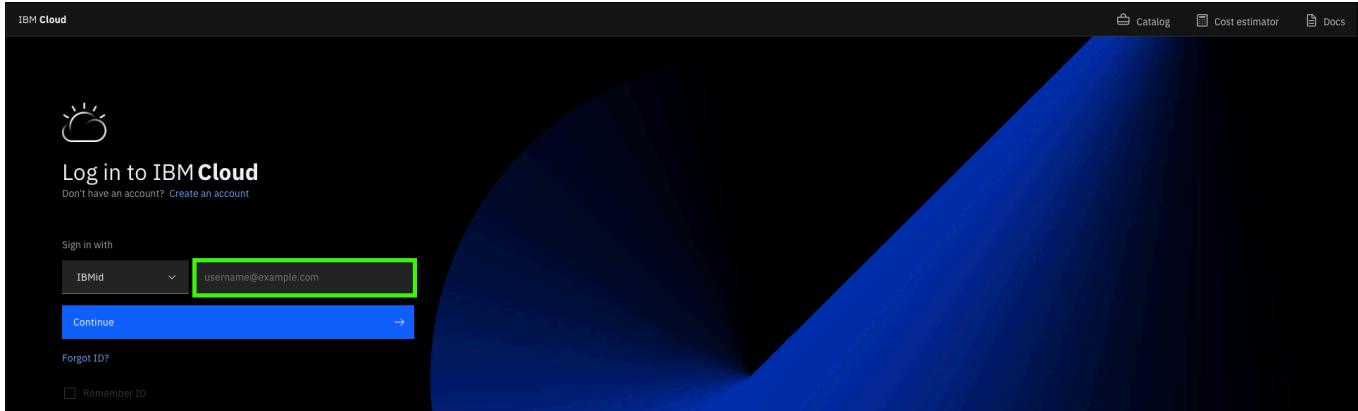


After joining the account, verify that the account appears in your available account list in the IBM Cloud portal.

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

IBM Cloud portal

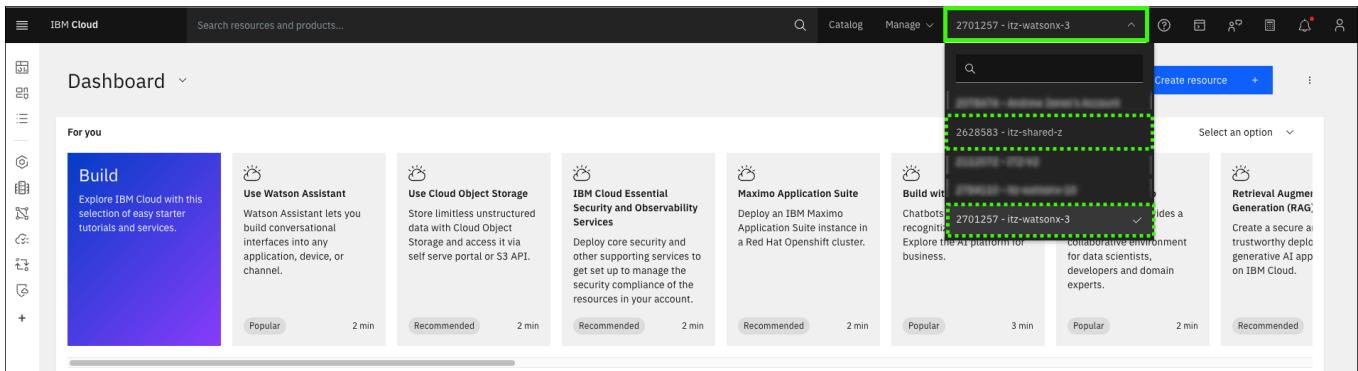
- Follow the directions to complete the authentication to IBM Cloud using the same email address you used to login to ITZ. The login steps vary depending on any two-factor authentication methods enabled.



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

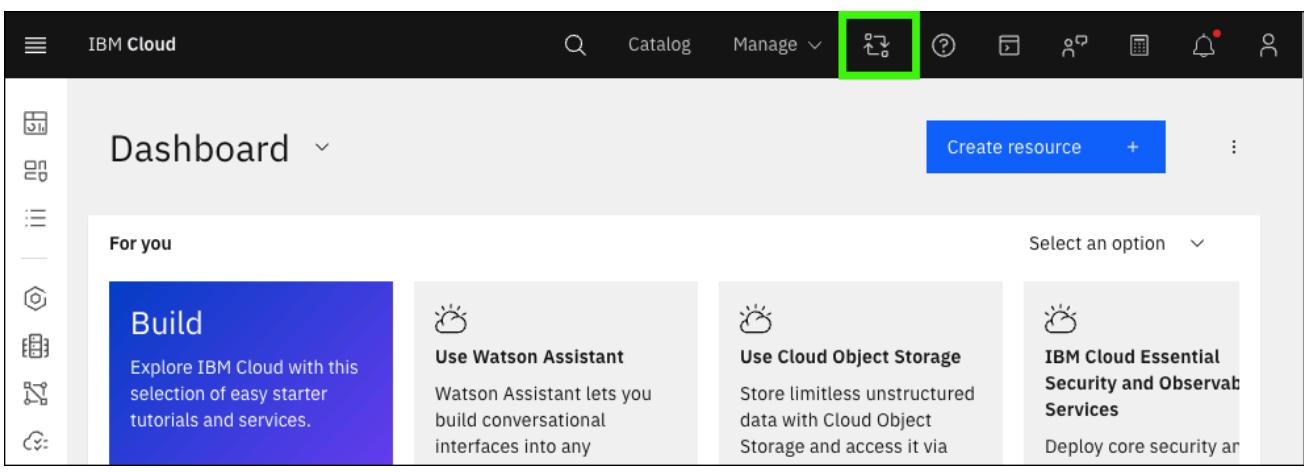
The account may be different.

The account name 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...

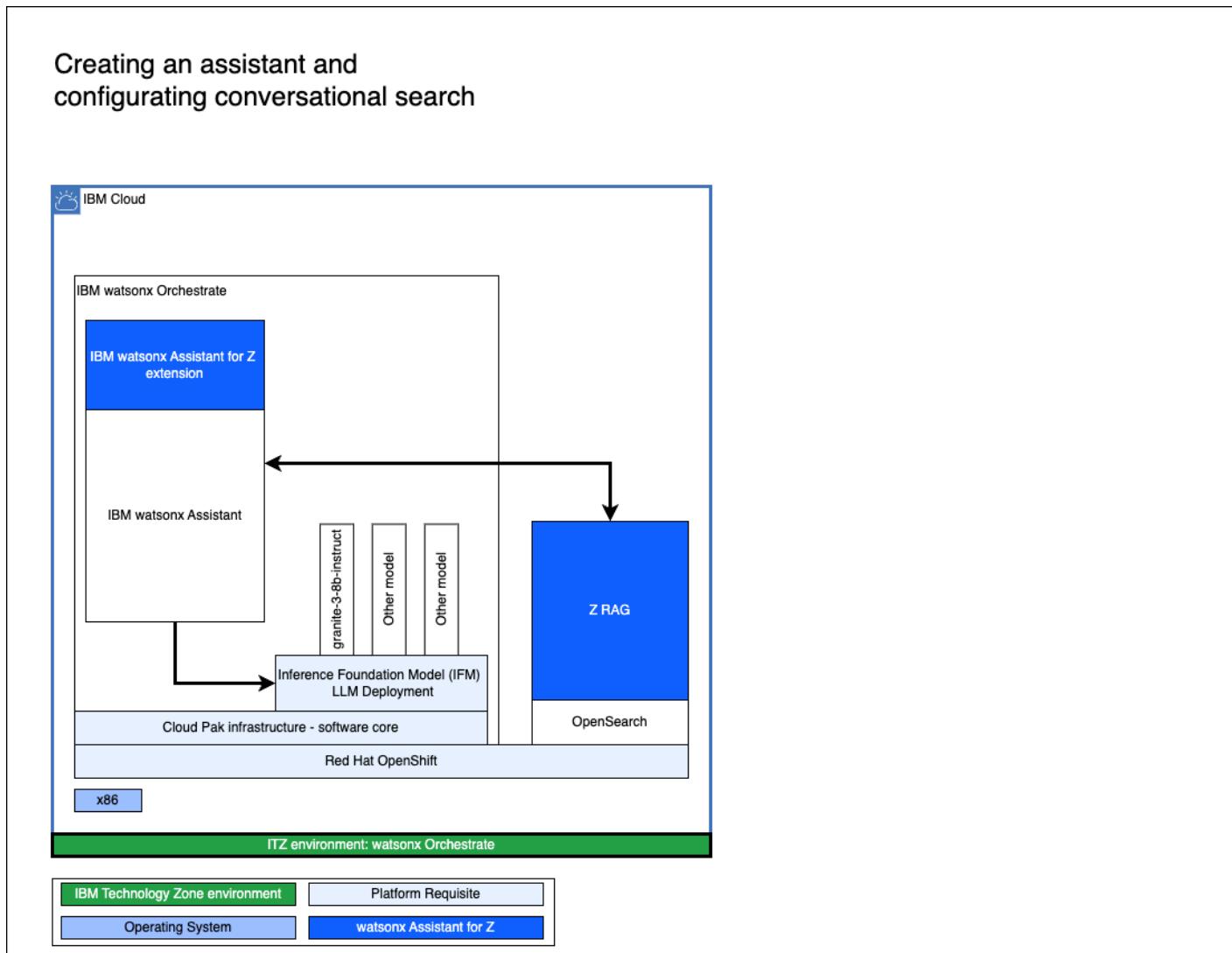
If your reservation for the Single Node OpenShift environment fails, try selecting one of the **eu-gb region** options as the **Preferred Geography**.

Creating an assistant and configuring conversational search

Watsonx Orchestrate allows you to create and configure an assistant with conversational search capabilities.

Configure your assistant to use conversational search by using 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.

A high-level, logical architecture of the environment is illustrated in the following diagram.



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.

Did you read the tip on the welcome page about creating a reference card? Check it out [here](#).

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

Purpose

- Purpose
- Education
- Opportunity Product(s)
- Customer(s)

Environment

- Reservation ID: 672a091a8f85062f81e081
- Type: IBM Cloud
- Request method: watsonx-orchestrate
- Region: us-south
- Customer data: false
- Idle runtime limit: 10800
- Transaction ID: 115897c9-58a1-4f17-af9c-b16dc2a97590
- Geo: americas
- Datcenter: dal10
- Environment: watsonx-orchestrate-wusdf
- Timeout action:

Reservation Details

IBM Cloud Login
<https://cloud.ibm.com/resources>



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.

- 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.

Resource list

Name	Group	Location	Product	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Compute (0)					
Containers (0)					
Networking (0)					
Storage (0)					
Converged infrastructure (0)					
Enterprise applications (0)					
AI / Machine Learning (1+)					
Analytics (0)					

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 ().

The screenshot shows the IBM Cloud Resource list interface. At the top, there's a search bar labeled "Search resources and products..." and several navigation buttons: Catalog, Manage, and a user icon. Below the search bar is a large search input field with a dropdown menu. To the right of the search field is a "Create resource" button. On the left, there's a sidebar with icons for Compute, Containers, Networking, Storage, and Converged infrastructure. The main area displays a table with columns: Name, Group, Location, Product, Status, and Tags. A row in the table is highlighted with a green box, showing the value "2953593 - itz-watsonx-036".

Create your Assistant

1. Click the **Resources** icon (☰).

The screenshot shows the IBM Cloud Dashboard. The left sidebar has a "Resources" icon highlighted with a green box. The main area is titled "Dashboard" and contains several cards with service descriptions. One card for "Build" is highlighted with a green box. The dashboard also includes a "Create resource" button.

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).

The screenshot shows the IBM Cloud Resource list interface. The sidebar has a "Resources" icon highlighted with a green box. The main area shows a table with columns: Name, Group, Location, Product, Status, and Tags. The "AI / Machine Learning" section is expanded, showing a list of instances. One instance, "itzwxo-2700039nft-erspw", is highlighted with a green box. This instance is located in Dallas, running on watsonx Orchestrate, and is marked as Active.

3. Click **Launch watsonx Orchestrate**.

Resource list / itzwxo-2700039nft-erspw Active Add tags [Edit](#)

Manage

Service credentials

Start by launching the tool

[Launch Watsonx Orchestrate](#) [Getting started tutorial](#)

Credentials

API key: [Download](#) [Show credentials](#)

URL: <https://api.us-south.assistant-builder.watson.cloud.ibm.com/instances/a7675d8e-e885-4891-92f3-1a2a2a2a2a2a>

Plan

Essentials Plan

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

Welcome, Andrew Jones!

Take productivity to the next level.

[Try Skills in Chat](#)

Build

Start building the skills, conversations, and automations your team needs here.

AI assistant builder

Boost productivity and customer care by creating conversational experiences.

Skill studio

Build the skills your team needs to get their work done more quickly.

Upgrade

Standard plan | Skill studio

Automate how your business works

[Learn more](#)

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

Welcome to AI assistant builder

Create Personalize Customize Preview

Create your first assistant

Let's get your assistant up and running. Name your assistant, add a description, and choose a language. In following steps we'll gather more information, show you basic customizations, and give you a preview of what your assistant will look like.

Assistant name:

Your assistant name will be kept internally and not visible to your customers.

Description (optional):

Assistant language:

This is the language your assistant will speak.

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.

- Select **Web**.
- Select the industry of your choice.
- Select the role of your choice.
- Select the need of your choice.

Welcome to AI assistant builder

Personalize your assistant

Tell us where your assistant will live
You may add multiple channels from your dashboard.
Where do you plan on deploying your assistant?
a Web

Tell us about yourself
This information will be used to personalize your onboarding experience.
Which industry do you work in?
b Software
What is your role on the team building the assistant?
c Developer
Which statement describes your needs best?
d I want to automate common tasks in a natural way

This is what your customers will experience

watsonx Assistant

Do you have the Speed Demons in stock?
The Speed Demons are in stock at our Downtown and Northgate locations, which are both within 5 miles of you.
What size and color do you need?
I'm looking for a size **9** in white
Great news! The Speed Demons are available in white in a size 9.
You can purchase them for curbside pickup or we can ship them to you. Which would you prefer?
I'll pick them up! Ship them to me!

Type something... ➤

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).

Welcome to AI assistant builder

Customize your chat UI

Update the style to match your brand and your website. You can change these settings later. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers
Zeeves

Intended purpose
 Standard: For virtual agents and customer support experiences.
 Carbon for AI: For use in internal IBM products.

Choose a theme
 Light
 Dark

Primary color #FFFFFF Secondary color #3D3D3D

Chat header User message bubble

Accent color #035AE9

Significant and interactive objects

Size
 The size of the web chat on this page will not change by updating these fields.
Width 380px Height 640px

IBM Watermark
Enable IBM Watermark

Streaming
Enable Streaming
 Off

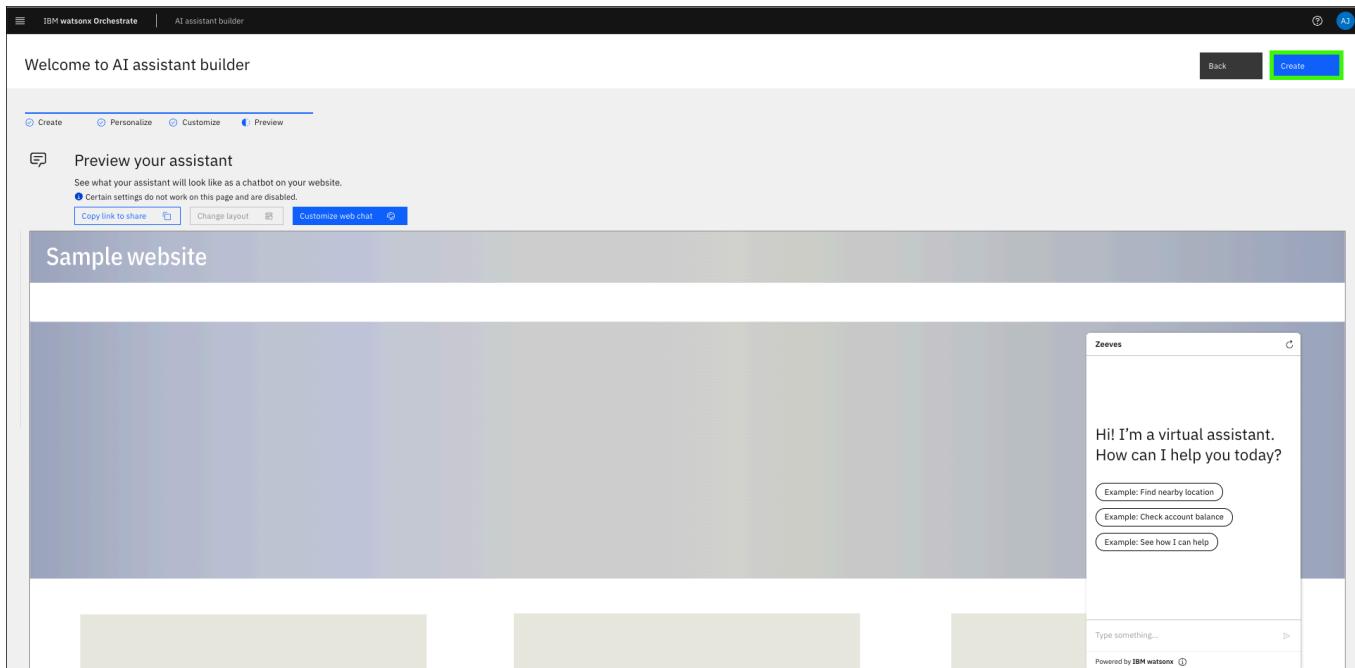
Hi! I'm a virtual assistant.
How can I help you today?

Example: Find nearby location
Example: Check account balance
Example: See how I can help

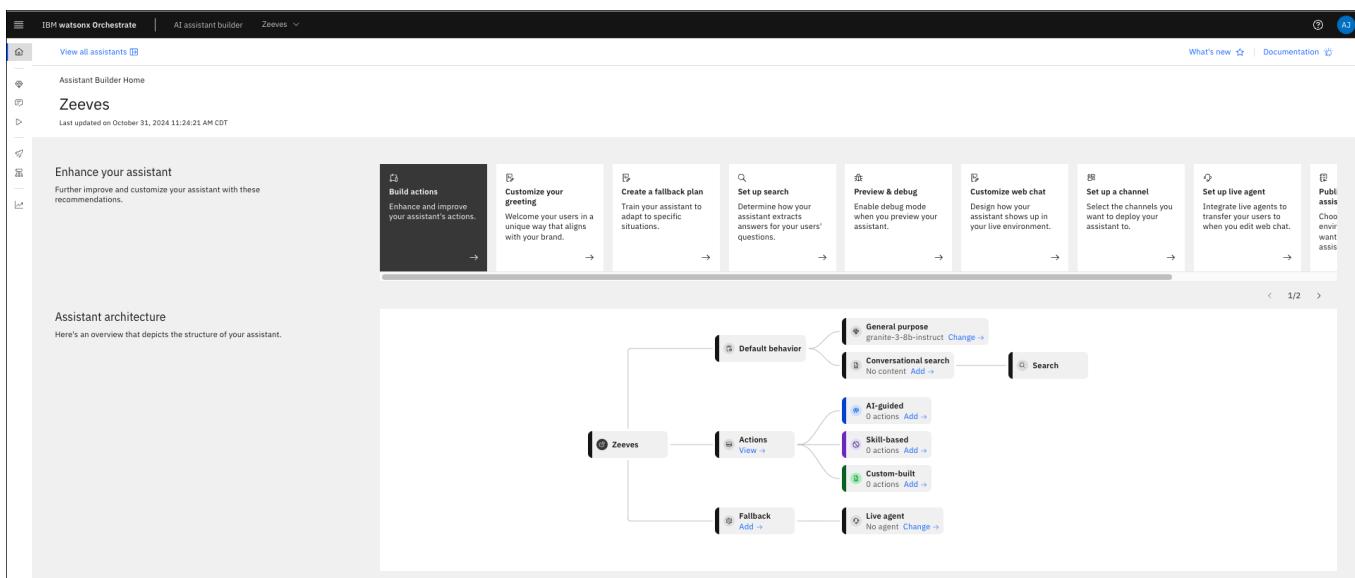
Type something... ➤

Powered by IBM watsonx

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



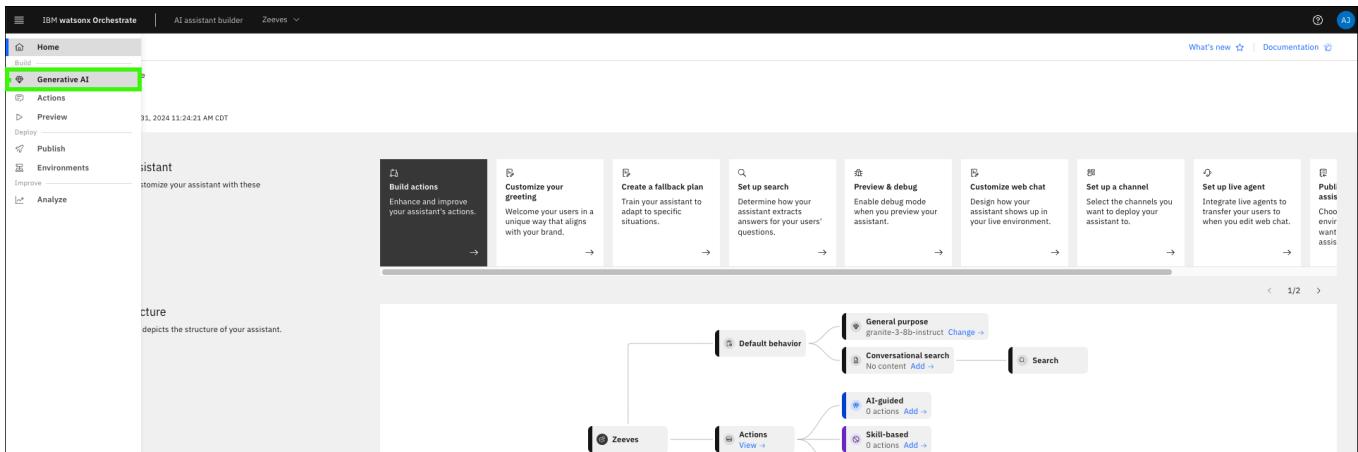
The assistant is now created.



Configure conversational search

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

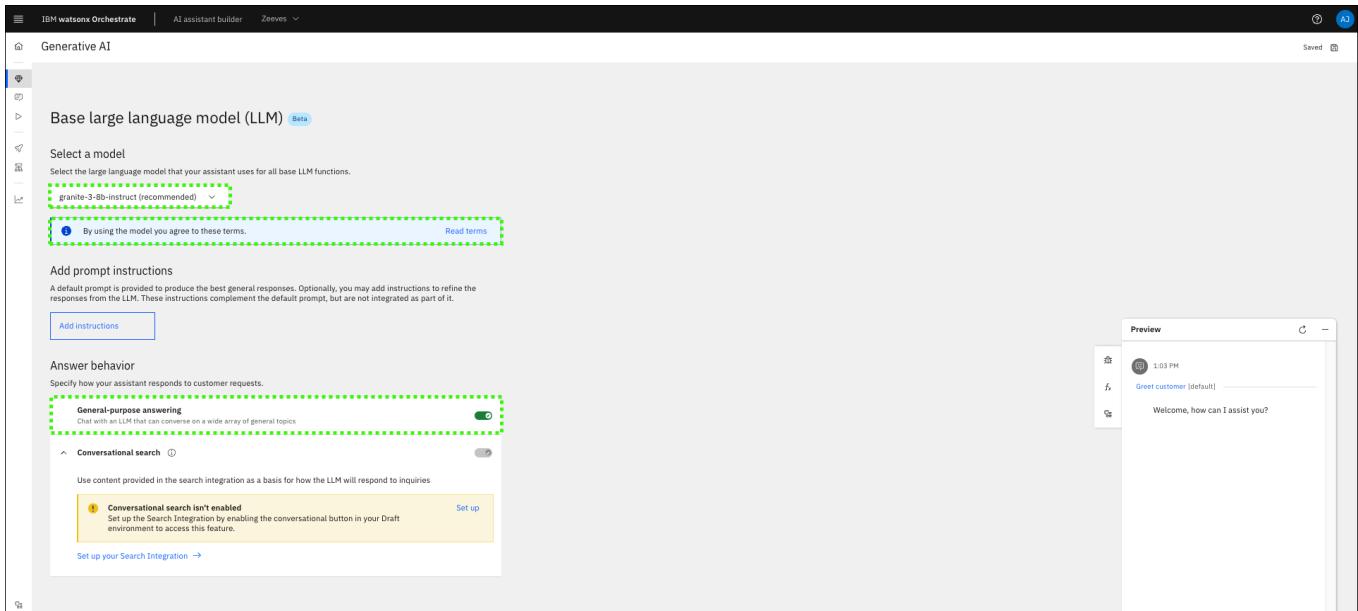
1. Click **Generative AI** menu item (💡) in the left navigation.



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

If available, set the LLM model to **granite-13b-chat-v2**.

Notice there 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. **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](#).

Generative AI

Base large language model (LLM) Beta

Select a model

grainne-3-8b-instruct (recommended)

By using the model you agree to these terms. [Read terms](#)

Add prompt instructions

General-purpose answering

Conversational search

Preview

1:03 PM Greet customer [default] Welcome, how can I assist you?

4. Click Custom service.

IBM Watsonx Orchestrate | AI assistant builder | Zeeves

Watsonx.ai model deprecation The model granite-13b-chat-v2 will be deprecated on 2024-11-04 and not available after 2025-01-19. Recommended alternative: granite-3-8b-instruct. You will be switched automatically on 2025-01-19.

Base large language model (LLM) Beta

Select a model

grainne-13b-chat-v2

Model will not be available after 2025-01-19.

By using the model you agree to these terms. [Read terms](#)

Add prompt instructions

General-purpose answering

Conversational search

Set up a new search integration

Es Elasticsearch Connect to an existing Elasticsearch instance. RECOMMENDED

Milvus Milvus Connect to an existing Milvus instance.

Custom service Connect to an existing service that will retrieve documents from your content management service. [Learn more](#)

Preview

7:45 AM Greet customer [default] Welcome, how can I assist you?

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 shared **OpenSearch** instance. In later sections, you create and customize a dedicated instance.

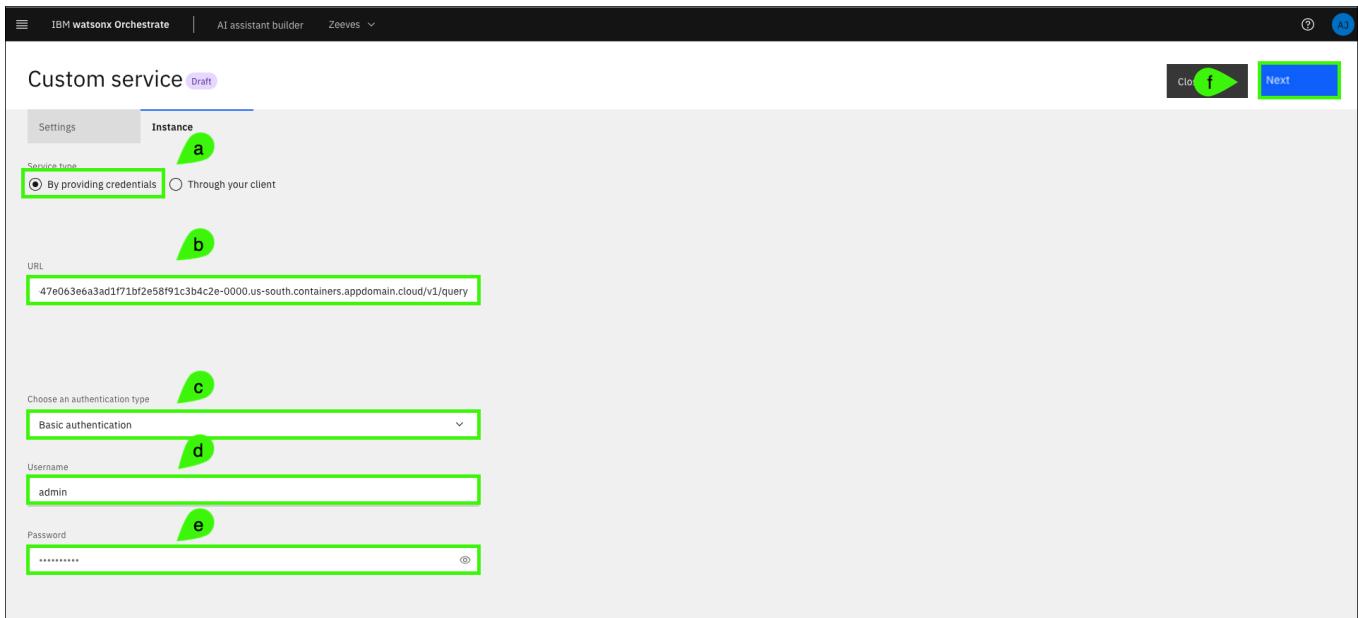
```
https://wxa4z-opensearch-wrapper-wxa4z-demo-v2-1-0.wxo4z-opc-opensearch-clus-47e063e6a3ad1f71bf2e58f91c3b4c2e-0000.us-south.containers.appdomain.cloud/v1/query
```

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

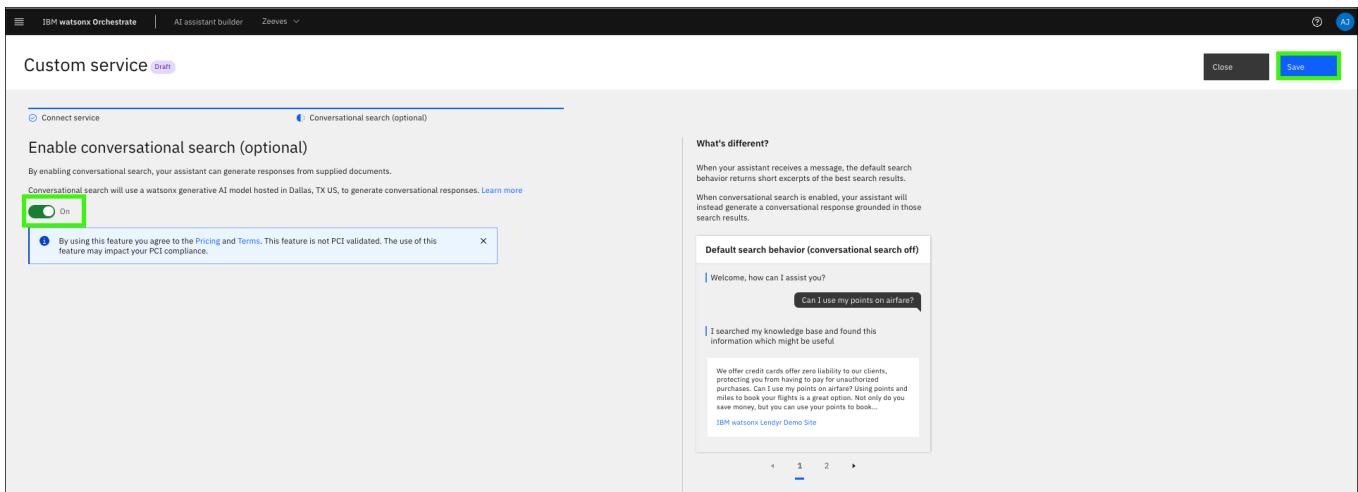
d. Enter **admin** in the **Username** field.

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

```
secureP@ssw0rd!
```



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](#).

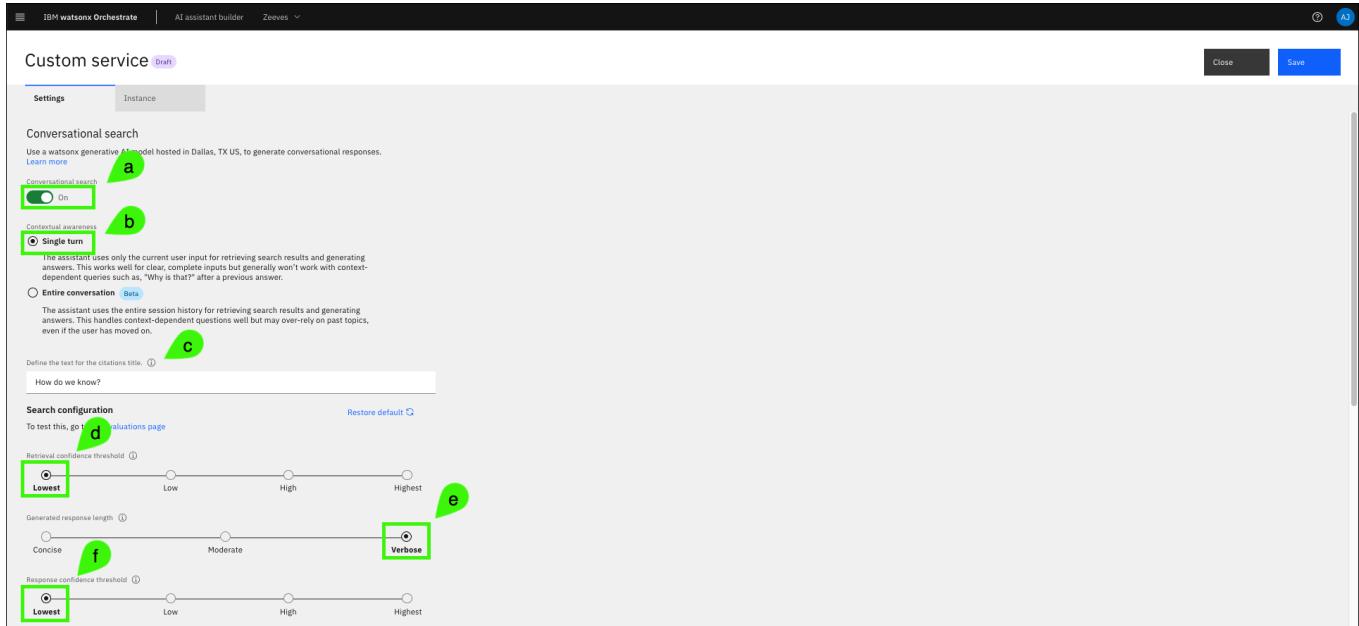
The following settings are proven to work well. You can experiment with these settings to see how they affect queries for your client's pilot.

- 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.

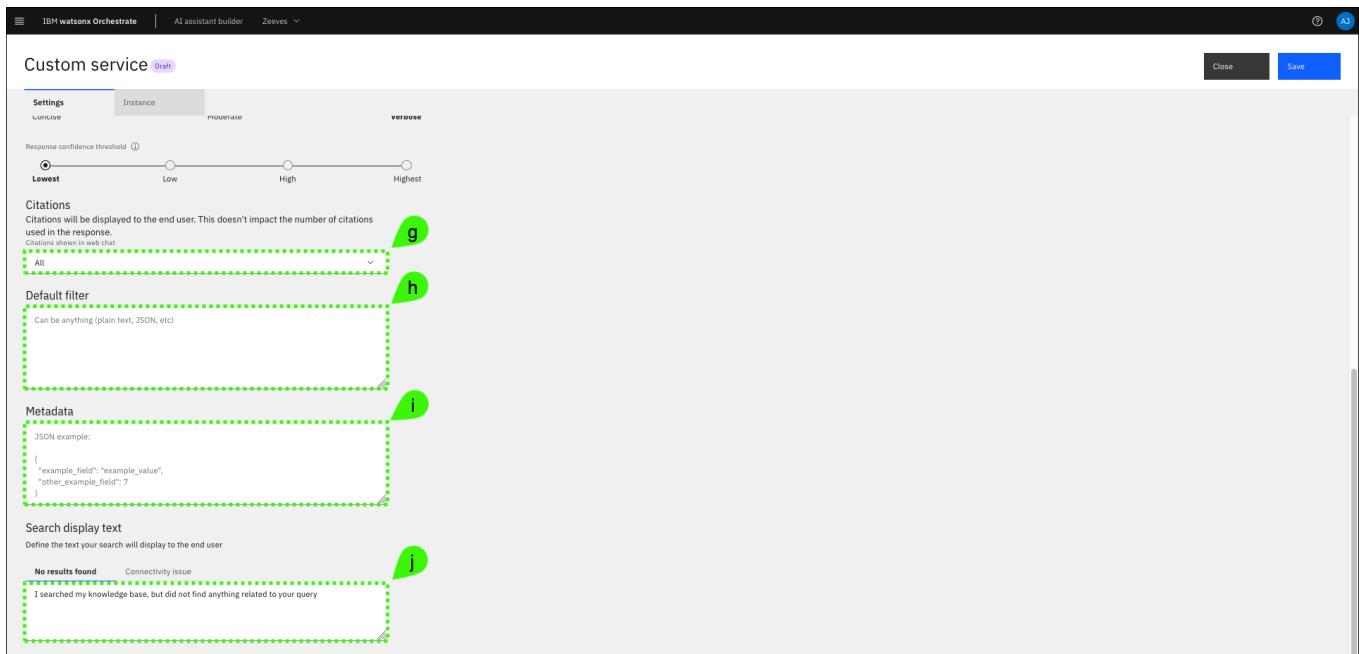


g. Keep the default setting of **All** for the listing of citations.

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).

Custom service Draft

Settings **Instance**

Conversational search

Use a watsonx generative AI model hosted in Dallas, TX US, to generate conversational responses. [Learn more](#)

Conversational search On

Contextual awareness

Single turn
The assistant uses only the current user input for retrieving search results and generating answers. This works well for clear, complete inputs but generally won't work with context-dependent queries such as, "Why is that?" after a previous answer.

Entire conversation Beta
The assistant uses the 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.

Define the text for the citations title. (?)

How do we know?

Search configuration Restore default

To test this, go to the [evaluations page](#)

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**.

IBM Watsonx Orchestrate | AI assistant builder Zeeves

Build

Generative AI Actions

Actions

- Preview
- Deploy
- Publish
- Environments
- Improve
- Analyze

Language model (LLM) Beta

language model that your assistant uses for all base LLM functions.

recommended

I accept the terms of service and agree to these terms. [Read terms](#)

Instructions

These instructions complement the default prompt, but are not integrated as part of it.

Preview

2:08 PM Greet customer [default]

Welcome, how can I assist you?

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

IBM Watsonx Orchestrate | AI assistant builder Zeeves

Actions

All items

Created by you

- Set by assistant** a
- Variables
- Created by you
- Set by assistant
- Set by integration

Saved responses

Create your first action

With actions, you can help your customers accomplish their goals.

Create action +

3. Click No matches.

The screenshot shows the 'Actions' section of the AI assistant builder. A table lists various actions with their names, last edit times, example counts, and statuses. The 'No matches' action is highlighted with a green box around its row.

Name	Last edited	Examples Count	Status
Greet customer	3 hours ago	1	Green
Trigger word detected	3 hours ago	1	Green
No matches	3 hours ago	1	Green
Fallback	3 hours ago	1	Green

4. Click Step 1 under Conversation steps.

The screenshot shows the 'Conversation steps' section for the 'No matches' action. Step 1 is highlighted with a green box. It contains a condition 'No matches count <= 3' and a response 'I'm afraid I don't understand. Please rephrase your question.' Below it, there's another step with a condition 'No matches count > 3' and a response 'Action complete'.

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.

The screenshot shows the configuration of Step 1. The 'Is taken' dropdown is set to 'with conditions'. A modal dialog titled 'Clear conditions?' is open, asking if the user wants to proceed. The 'Clear conditions' button in the dialog is highlighted with a red box.

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

The screenshot shows the 'Assistant says' field for Step 1. The default text 'or example: Please select from the following options:' is highlighted with a green box. Below it, there are other options like 'Define customer response' and 'End the action'.

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

The screenshot shows the AI assistant builder interface with the 'Editor' tab selected. In the 'Step 1' section, the 'Is taken' dropdown is set to 'without conditions'. The 'Assistant says' field contains a placeholder message: 'For example: Please select from the following options:'. Below this, the 'And then' section is expanded, showing a list of actions. The 'End the action' option is highlighted with a green border. A sub-menu for 'End the action' is open, listing several options: 'Continue to next step', 'Re-ask previous steps', 'Go to a subaction', 'Use an extension', 'Search for the answer', 'Connect to agent', and 'End the action'. The 'Search for the answer' option is also highlighted with a green border.

8. Click Edit settings.

The screenshot shows the AI assistant builder interface with the 'Editor' tab selected. The 'Step 1' configuration is identical to the previous screenshot. In the 'And then' section, the 'Search for the answer' action is selected and highlighted with a green border. A modal dialog titled 'Edit settings' is open over the interface. This dialog is specifically for the 'Search for the answer' action, showing its configuration options. It includes fields for 'Custom query' (set to 'None (Customer's message will be the query)') and 'Custom filter' (set to 'None (Optional)'). At the bottom of the dialog, there is a 'Edit settings' button, which is also highlighted with a green border.

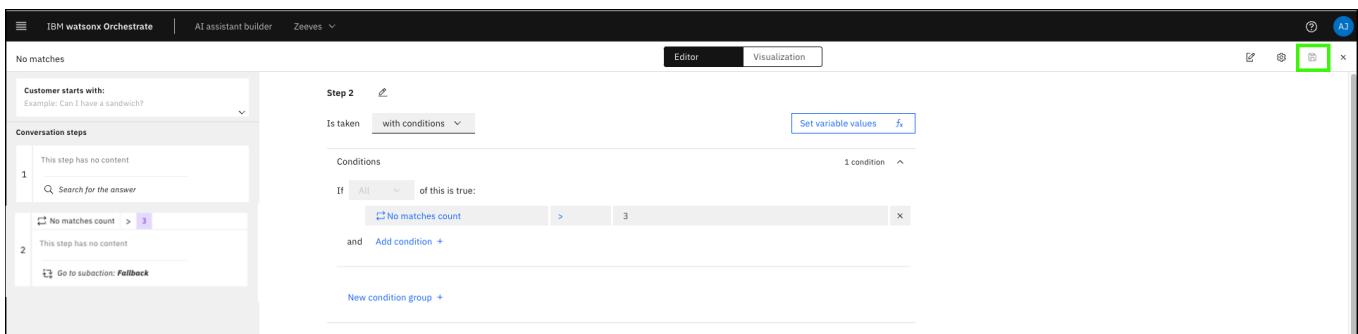
9. Click After generation.

The screenshot shows the AI assistant builder interface with the 'Editor' tab selected. The 'Step 1' configuration is identical to the previous screenshots. In the 'And then' section, the 'Search for the answer' action is selected and highlighted with a green border. A modal dialog titled 'Search configuration' is open, showing the 'After generation' tab selected. This dialog allows configuring how the search performs before a response is generated. It includes fields for 'Custom search query (Optional)' (with an example placeholder) and 'Custom results filter' (with radio buttons for 'Use default' and 'Set new filter'). At the bottom of the dialog, there are 'Cancel' and 'Apply' buttons, with 'Apply' being highlighted with a blue background.

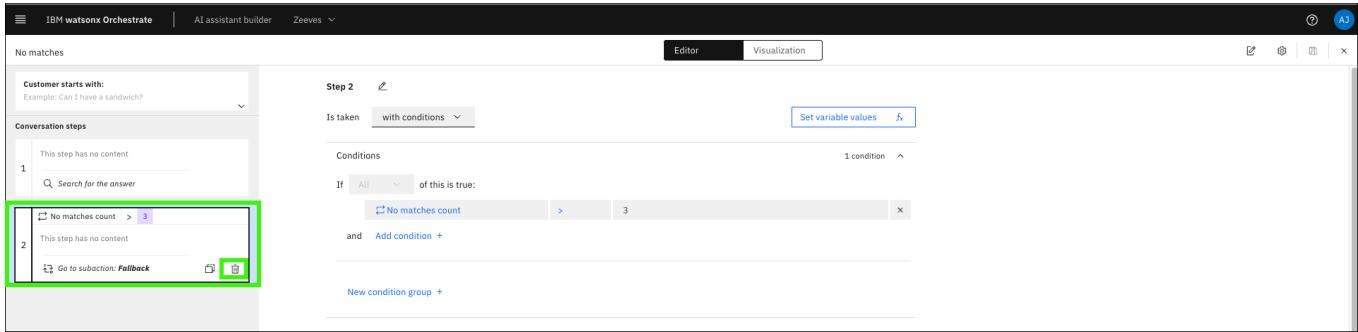
10. Select End the action after this step and then click Apply.



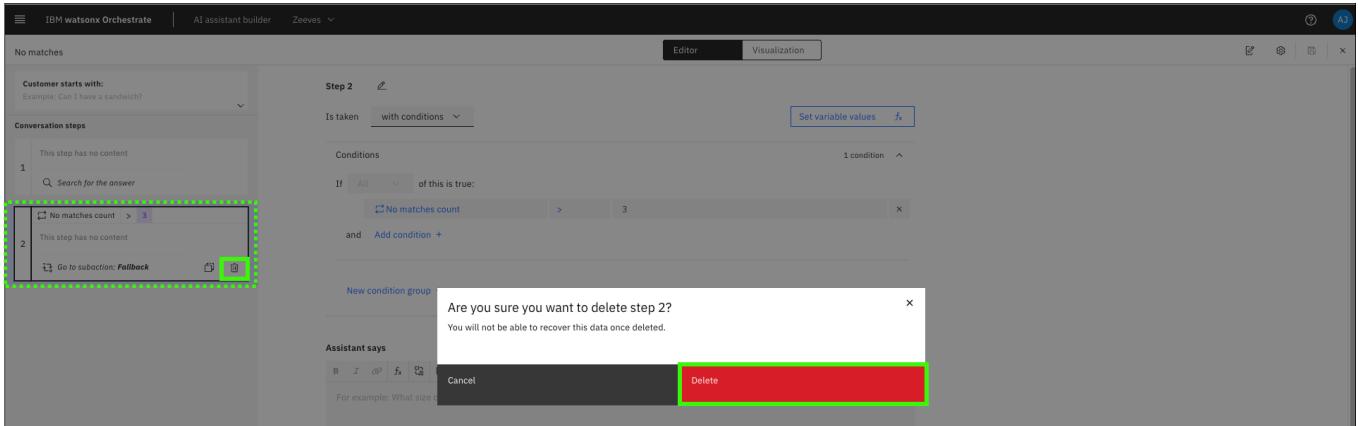
11. Click Save (💾).



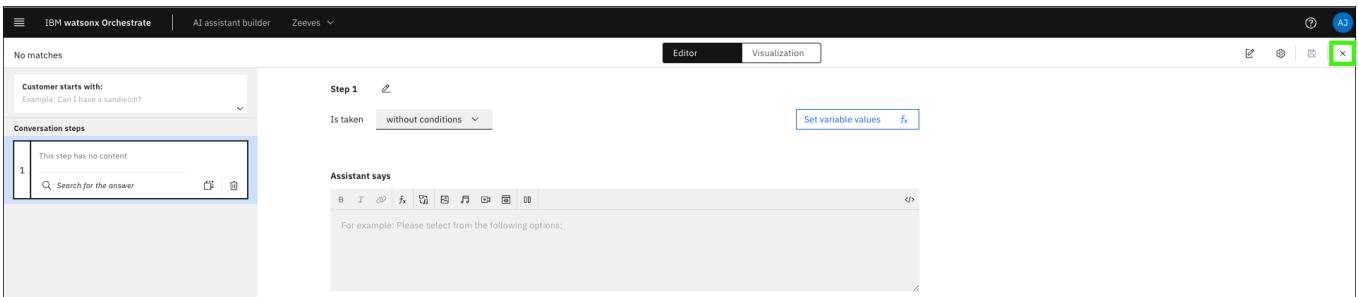
12. Select Step 2 (No matches count) under Conversation steps and click delete (🗑).



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.

Actions				
	Name	Last edited	Examples Count	Status
All items	Greet customer	3 hours ago	0	Green
Created by you	Trigger word detected	3 hours ago	0	Green
Set by assistant	No matches	a minute ago	0	Green
Variables	Fallback	3 hours ago	0	Green
Created by you				
Set by assistant				
Set by integration				
Saved responses				

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** (trash bin icon) 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.

Global settings

Conversation routing Change conversation topic Generative AI New! Autocorrection Display formats Algorithm Version Upload/Download

Ask clarifying questions

Customize modes Beta

No matches (highlighted with a green box)

Search routing

Your assistant can show options to users when multiple actions seem to match what the customer wants. [Learn more about asking a clarifying question](#)

Enable disambiguation On

Assistant says

Introduction text before listing the options.

Did you mean:

No matches Option for when the user doesn't see relevant options. Leave blank to omit.

None of the above

Response modes

Off

One action matches

Option for when user doesn't want to start the matched action.

Something else

Connection to support

Can be any alternative help, such as a live agent or contact information.

Connect to support

Example of asking a clarifying question

Important Make sure your action names are short and clear to your customers. Action names will appear in the list of choices.

Bank Bot

open an account

Did you mean:

- Open a new savings account
- Open a new checking account
- I want to apply for a mortgage loan
- None of the above

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.

Global settings

Conversation routing Change conversation topic Generative AI New! Autocorrection Display formats Algorithm Version Upload/Download

Ask clarifying questions

Customize modes Beta

No matches (highlighted with a green box)

Search routing

Unrecognized input by customers triggers the **No matches** action that can be configured to fetch answers from a [search integration](#) or trigger the [Fallback](#) action.

By setting this threshold, you can affect how often your assistant routes customers to the "No matches" action.

Use "No matches"

Save

21. Click **Autocorrection**.

Global settings

Conversation routing Change conversation topic Generative AI New! Autocorrection Display formats Algorithm Version Upload/Download

Ask clarifying questions

Customize modes Beta

No matches (highlighted with a green box)

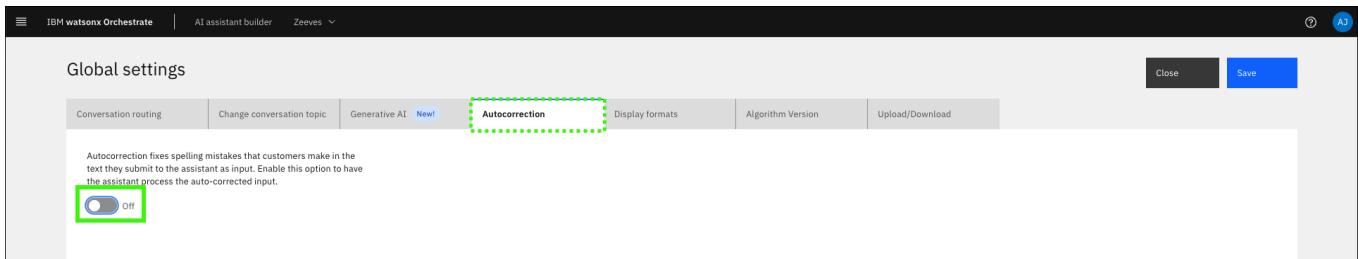
Unrecognized input by customers triggers the **No matches** action that can be configured to fetch answers from a [search integration](#) or trigger the [Fallback](#) action.

By setting this threshold, you can affect how often your assistant routes customers to the "No matches" action.

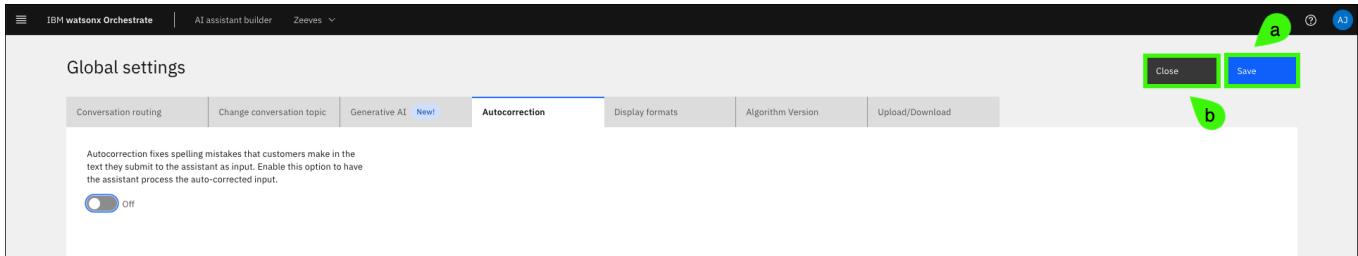
Use "No matches"

Save

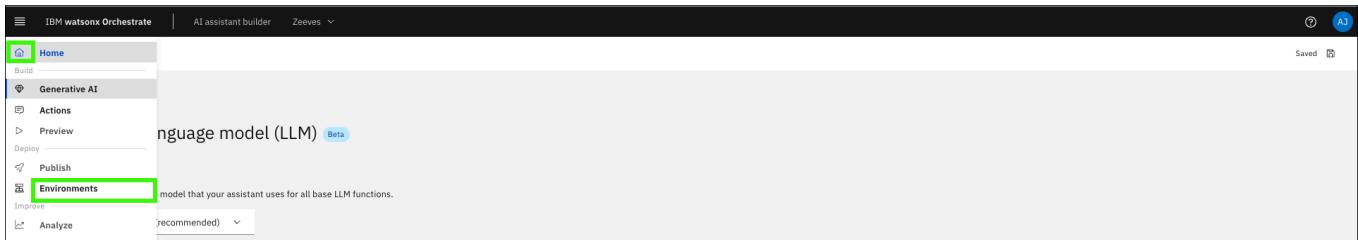
22. Click the autocorrection toggle to turn the feature **Off**.



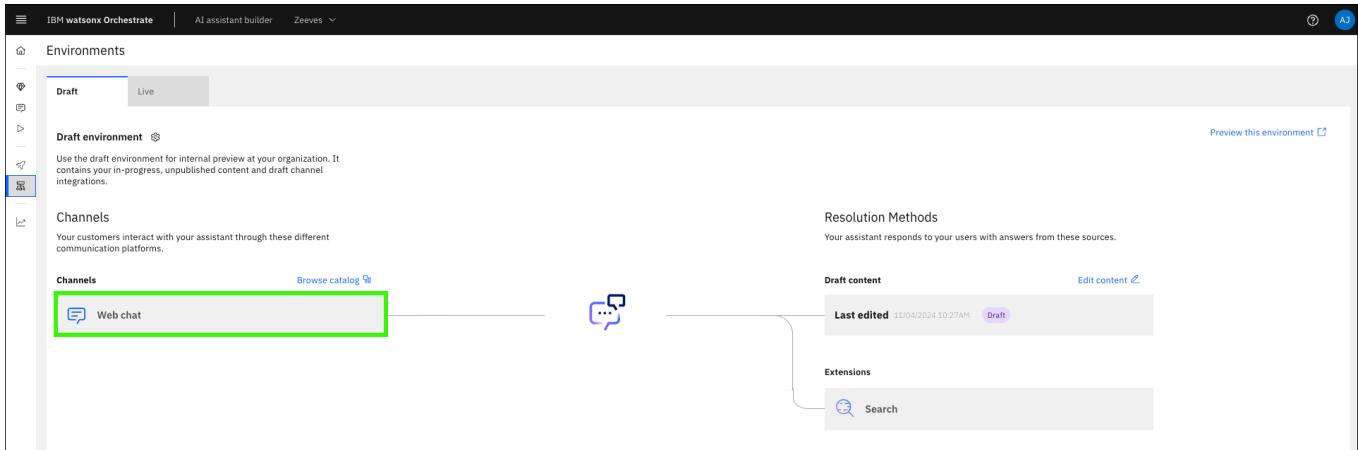
23. Click **Save** (a) and then **Close** (b).



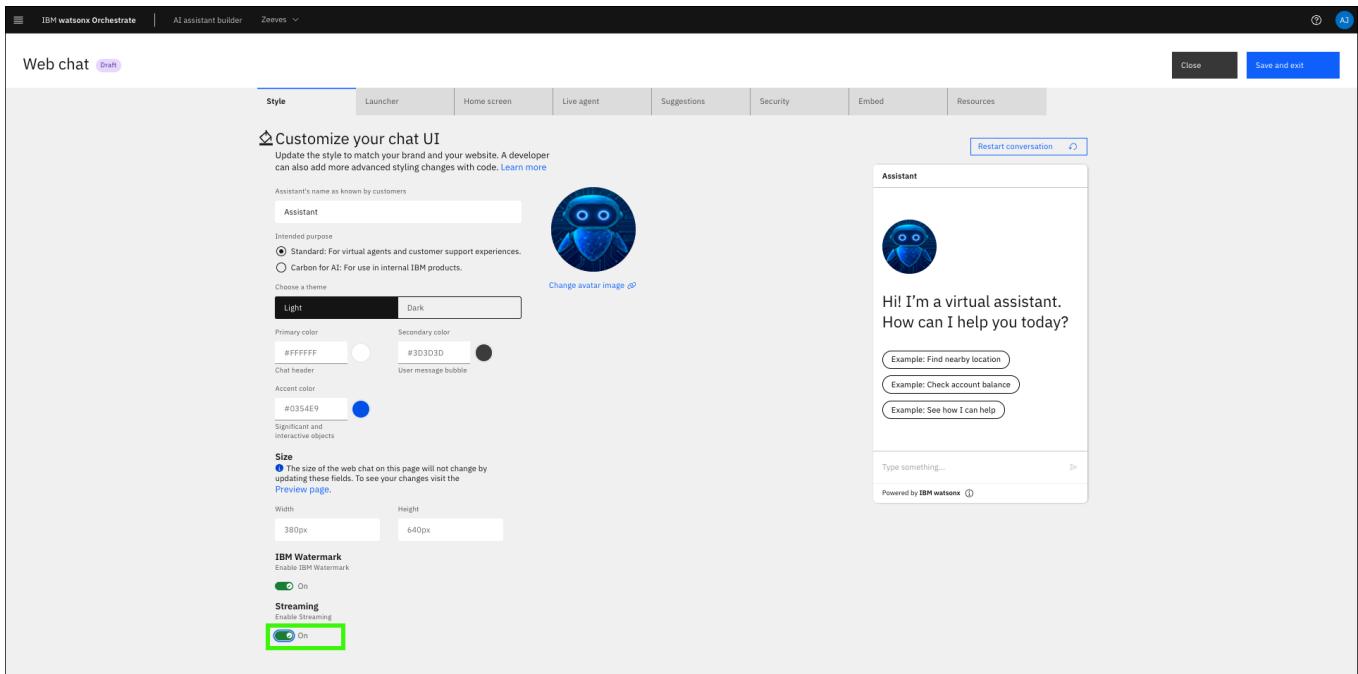
24. Hover over the **Home** (🏠) and click **Environments**.



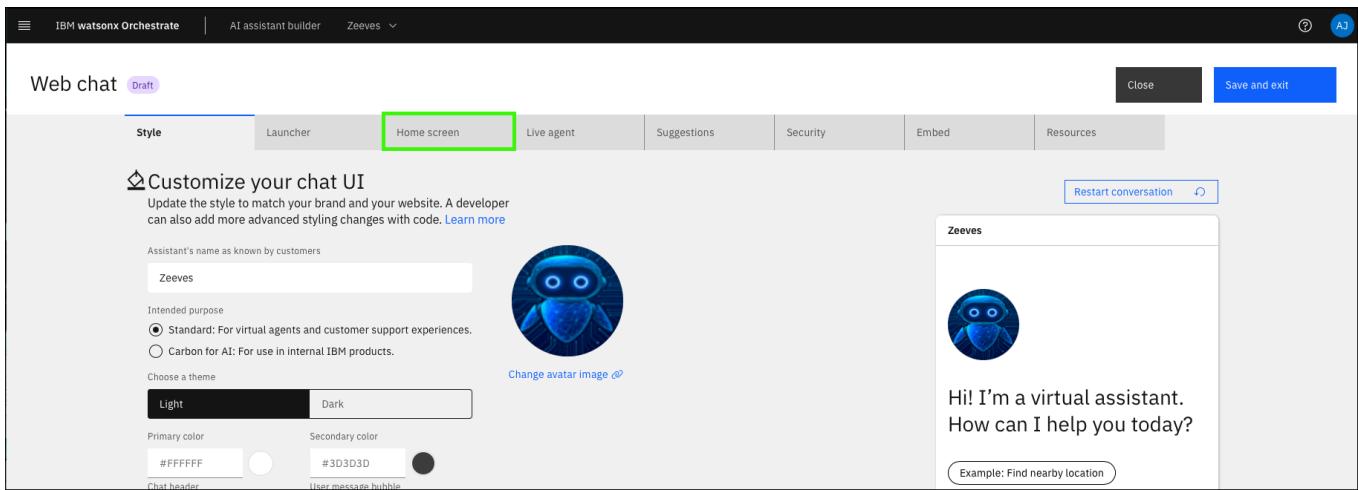
25. Click **Web chat**.



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



27. Click the **Home screen tab.**



28. Customize the **Home screen by setting a custom **Greeting message** and deleting the default **Conversation starters**. Optionally, adjust the **Background style**.**

Web chat Draft

Style **Launcher** **Home screen** **Live agent** **Suggestions** **Security** **Embed** **Resources**

Home screen

Make a good first impression and get customers the help they need more quickly. [Learn more](#)

On

Greeting

Good greetings are welcoming, actionable, and expressive of your assistant's personality.

ⓘ This message will override the dialog welcome node and Actions greeting.

Greeting message

Hi! I'm Zeeves. How can I help you today?

Conversation starters

Help your customers get going. When selected, the text of these starters is sent as a message. Each should be something your assistant is trained and tested to answer.

1 Example: Check account balance

(1/5)

Background style

Choose a style for your home screen's background.

None

Accent color

Gradient from the top corner

Gradient from the bottom

Zeeves

Restart conversation

Hi! I'm Zeeves. How can I help you today?

Example: Check account balance

Type something...

Built with IBM watsonx

29. Click **Suggestions**.

Web chat Draft

Style **Launcher** **Home screen** **Live agent** **Suggestions** **Security** **Embed** **Resources**

Customize your chat UI

Update the style to match your brand and your website. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers

Zeeves

Intended purpose

Standard: For virtual agents and customer support experiences.

Carbon for AI: For use in internal IBM products.

Choose a theme

Light

Dark

Primary color #FFFFFF

Secondary color #303030

Accent color #0354E9

Significant and interactive objects

Size

ⓘ The size of the web chat on this page will not change by updating these fields. To see your changes visit the [Preview page](#).

Width 380px

Height 640px

IBM Watermark

Enable IBM Watermark

On

Streaming

Enable Streaming

On

Zeeves

Restart conversation

Hi! I'm a virtual assistant.
How can I help you today?

Example: Find nearby location

Example: Check account balance

Example: See how I can help

Type something...

Powered by IBM watsonx

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

Web chat Draft

Suggestions

Suggestions appear as message options when your customers need a bit of help. They adapt to the conversation and can offer both responses to try and connection to support. [Learn more](#)

Off

Zeeves

Hi! I'm Zeeves. How can I help you today?

Type something... ▶

Built with IBM watsonx ⓘ

Close Save and exit

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

Web chat Draft

Suggestions

Suggestions appear as message options when your customers need a bit of help. They adapt to the conversation and can offer both responses to try and connection to support. [Learn more](#)

Off

Zeeves

Hi! I'm Zeeves. How can I help you today?

Type something... ▶

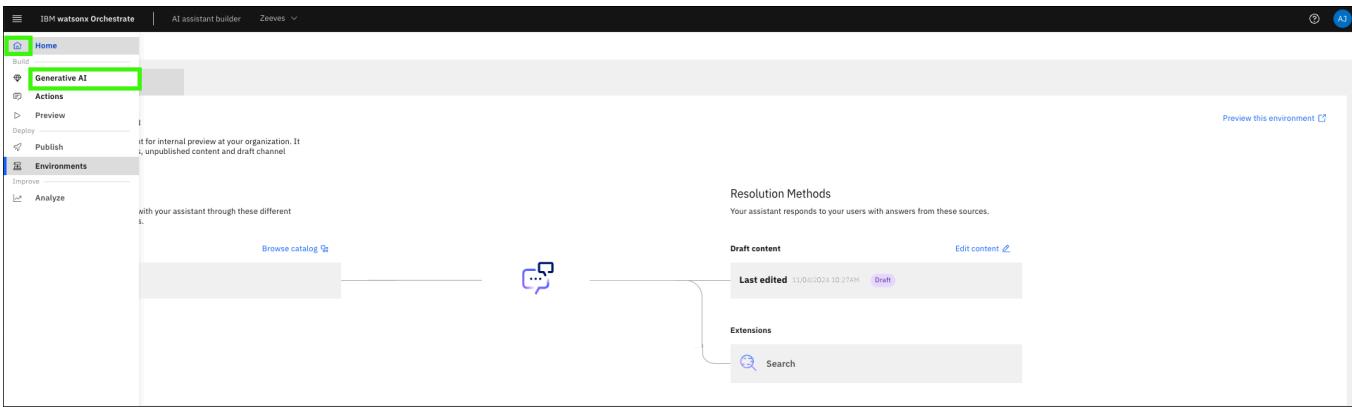
Built with IBM watsonx ⓘ

a Close **b** Save and exit

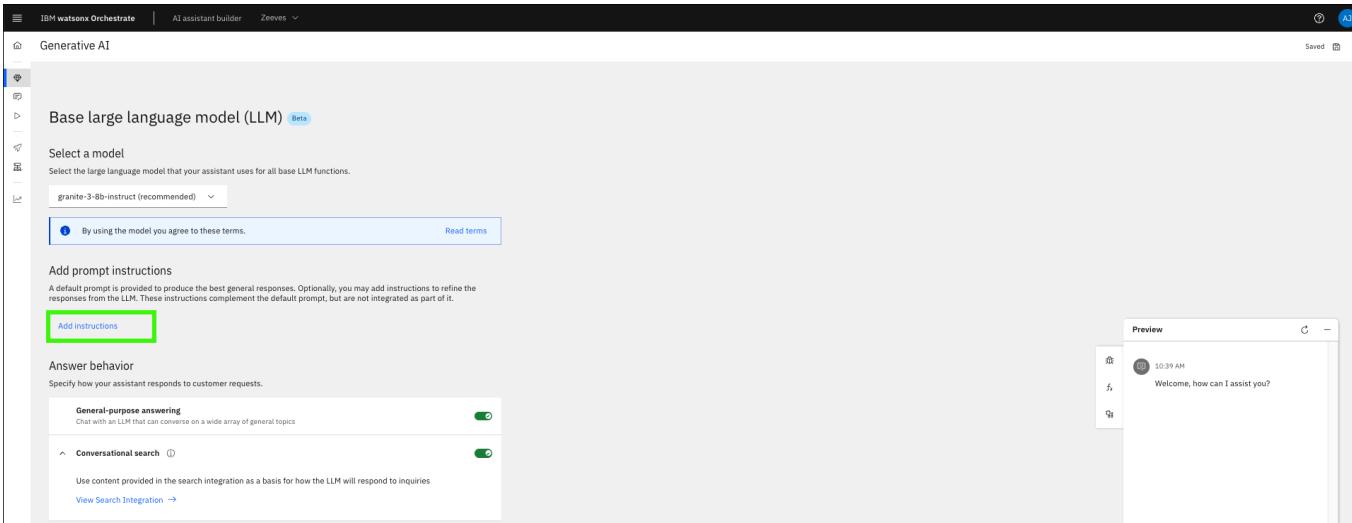
Configure the base large language model

There are enhancements that 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. The options are 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.

The screenshot shows the 'Generative AI' configuration page. In the 'Answer behavior' section, the 'General-purpose answering' toggle switch is turned off (indicated by a greyed-out appearance). The 'Conversational search' toggle switch is turned on (indicated by a green background). A preview window on the right shows a message from 'Greet customer [default]' at 10:47 AM: 'Welcome, how can I assist you?'. The 'Save' button in the top right corner is highlighted with a blue box.

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 and conversational search, 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.

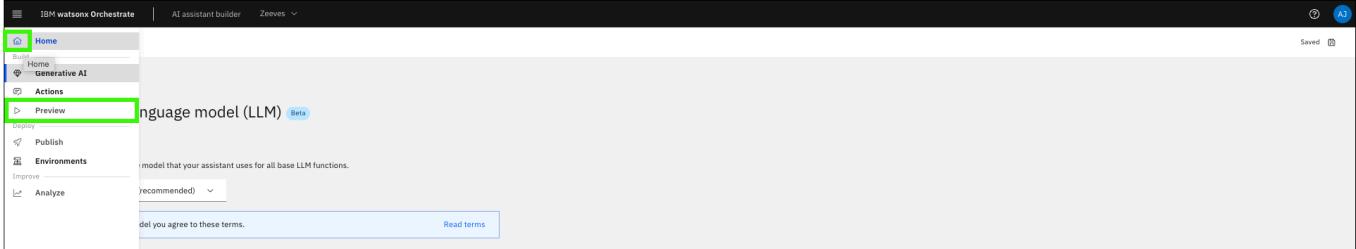
The screenshot shows the 'Generative AI' configuration page. In the 'Answer behavior' section, the 'General-purpose answering' toggle switch is turned off (indicated by a greyed-out appearance). The 'Conversational search' toggle switch is turned on (indicated by a green background). A preview window on the right shows a message from 'Greet customer [default]' at 10:47 AM: 'Welcome, how can I assist you?'. The 'Save' button in the top right corner is highlighted with a blue box.

Testing conversational search

Now, you can begin issuing queries to test the assistant's responses. For more detailed responses, try appending "Please provide a detailed response." to the end of your question.

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:

The screenshot shows the IBM Watsonx Assistant interface. On the left, there's a preview of a "Sample website" which is mostly blank. On the right, there's a chat window titled "Zeeves". A message from "You" at 11:02 AM asks, "What is the APF list in z/OS? Provide a detailed explanation?" Below it, a response from "Zeeves" at 11:02 AM provides a detailed explanation of the APF list, mentioning its role in controlling access to system resources and how it can be managed using system commands and macros. The response is highlighted with a green dashed box.

Prompt:

Why is Db2 different than other database systems?

Example output:

The screenshot shows the IBM Watsonx Assistant interface. On the left, there's a preview of a "Sample website" which is mostly blank. On the right, there's a chat window titled "Zeeves". A message from "You" at 11:03 AM asks, "Why is Db2 different than other database systems?" Below it, a response from "Zeeves" at 11:03 AM explains that Db2 is a database management system (DBMS) that runs under the z/OS operating system, which is specific to mainframe computers. It compares Db2 to other database systems like Oracle and MySQL, noting that Db2 provides a higher level of data management and security features compared to other relational systems, as it manages data stored in databases and controls access to the data. The response is highlighted with a green dashed box.

Prompt:

What happens during an IPL on IBM Z?

Example output:

The screenshot shows the IBM Watson Assistant interface. At the top, there's a navigation bar with 'IBM Watson Orchestrate', 'AI assistant builder', and 'Zeeves'. Below the navigation is a toolbar with icons for preview, copy link, change layout, change background, and customize web chat. The main area is titled 'Sample website' and contains three large, empty rectangular boxes. To the right of the website preview is a chat window titled 'Zeeves'. The chat log shows a message from the user 'What happens during an IPL on IBM Z?' and a response from the AI assistant. The AI response is highlighted with a green box. The response text is as follows:

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 open systems host, or a z/OS host. 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 Primary Link Pack Area (PLPA) is reloaded and whether VTO data set pages are preserved.



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**.

The screenshot shows the 'Custom service' settings page in the IBM Watsonx Orchestrate interface. The 'Contextual awareness' section is highlighted with a green border. The 'Entire conversation' option is selected, while 'Single turn' is unselected. A tooltip for 'Entire conversation' provides a detailed description of how it works.

Entire conversation Beta
The assistant uses the 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.

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.

Bring your own search and documents

Creating a stand-alone OpenSearch instance for document ingestion

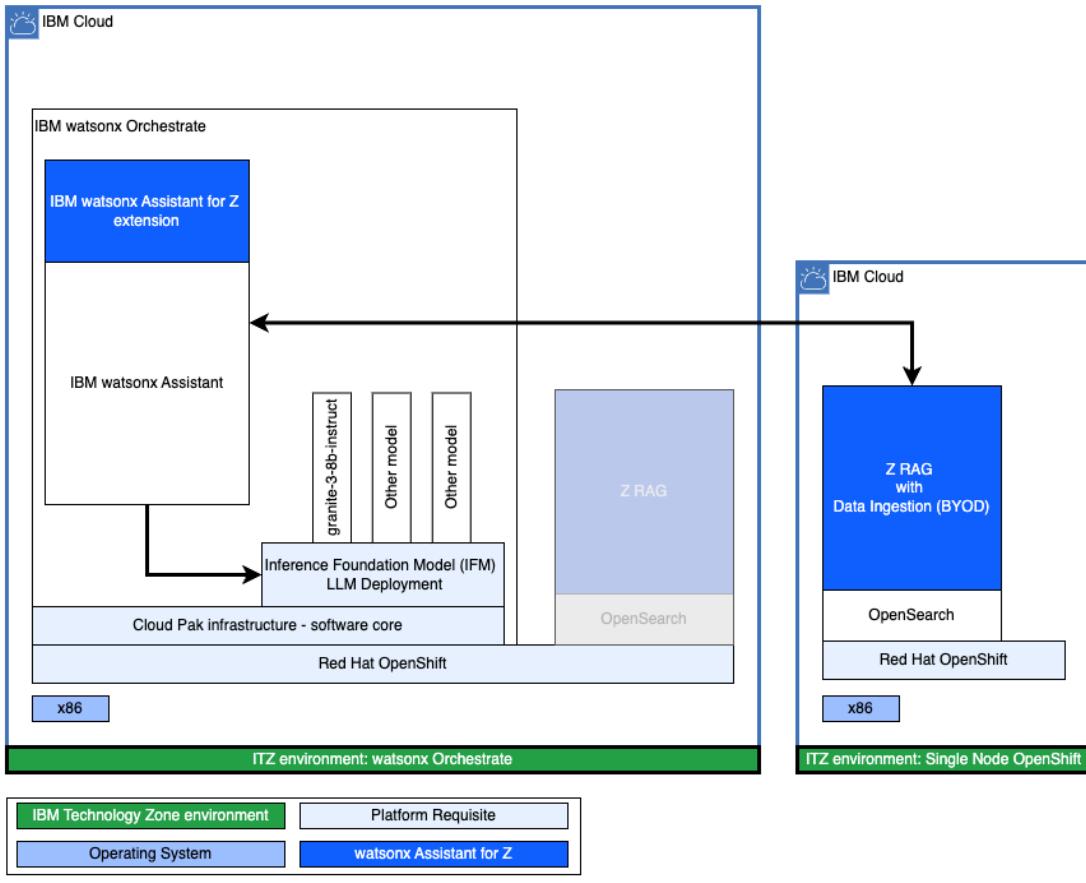
Now that you created and deployed your own assistant with conversational search capabilities, your client can understand how watsonx Assistant for Z provides its content-grounded responses to any Z-related questions. In the previous section, you configured your assistant to use a pre-configured Z RAG that has over 220 knowledge sources, and uses this knowledge to provide AI-generated responses.

Next, learn to enable clients to personalize the assistant with an internal knowledge base that contains documentation they add to the Retrieval Augmented Generation (RAG). This helps provide a level of context-awareness for their own environment when environment-specific questions are asked to the assistant.

Now, install and configure a “Z RAG” on Red Hat OpenShift enabling the bring-your-own-search (BYOS) and bring-your-own-documentation (BYOD) capability to ingest other documentation. In doing so, you deploy a dedicated OpenSearch instance (BYOS). Then, connect your assistant to the new RAG database to provide responses based on the ingested documentation (BYOD).

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

Creating a stand-alone OpenSearch instance for document ingestion



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.

My reservations

Status - Ready
Single Node OpenShift (VMware on IBM Cloud)

Education

Start date: Oct 31, 2024 7:02 AM
End date: Nov 6, 2024 6:49 AM
Extend limit: 0

[Open this environment](#)

Status - Ready
watsonx Assistant for Z Pilot - AAP & z/OS

Education

Start date: Oct 30, 2024 8:12 AM
End date: Nov 5, 2024 7:53 AM
Extend limit: 0

[Open this environment](#)

Status - Ready
watsonx Assistant for Z Pilot - watsonx O...

Education

Start date: Oct 30, 2024 8:10 AM
End date: Nov 5, 2024 7:12 AM
Extend limit: 0

[Open this environment](#)

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

Reservation Details

API URL: <https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443>

Bastion Password:

Bastion RDP address: api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:43389

Bastion SSH connection: ssh itzuser@api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com -p 40222

Bastion Username: itzuser

Cluster Admin Username: kubeadmin

Cluster Admin Password: [REDACTED]

OCP Console: <https://console.nginx.ingress.k8s.itzuser-vc.eu.cloud.techzone.ibm.com>

OCP Version: 4.14

vCenter: itzeu-vc.eu.cloud.techzone.ibm.com

[Download kubeconfig](#)

4. Click the OCP Console link.

Note: OCP stands for OpenShift Container Platform.

Reservation Details

API URL
<https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443>

Bastion Password [REDACTED]

Bastion RDP Address
api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:43389

Bastion SSH connection
ssh itzuser@api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com -p 40222

Bastion Username
itzuser

Cluster Admin Username
kubeadmin

Cluster Admin Password [REDACTED]

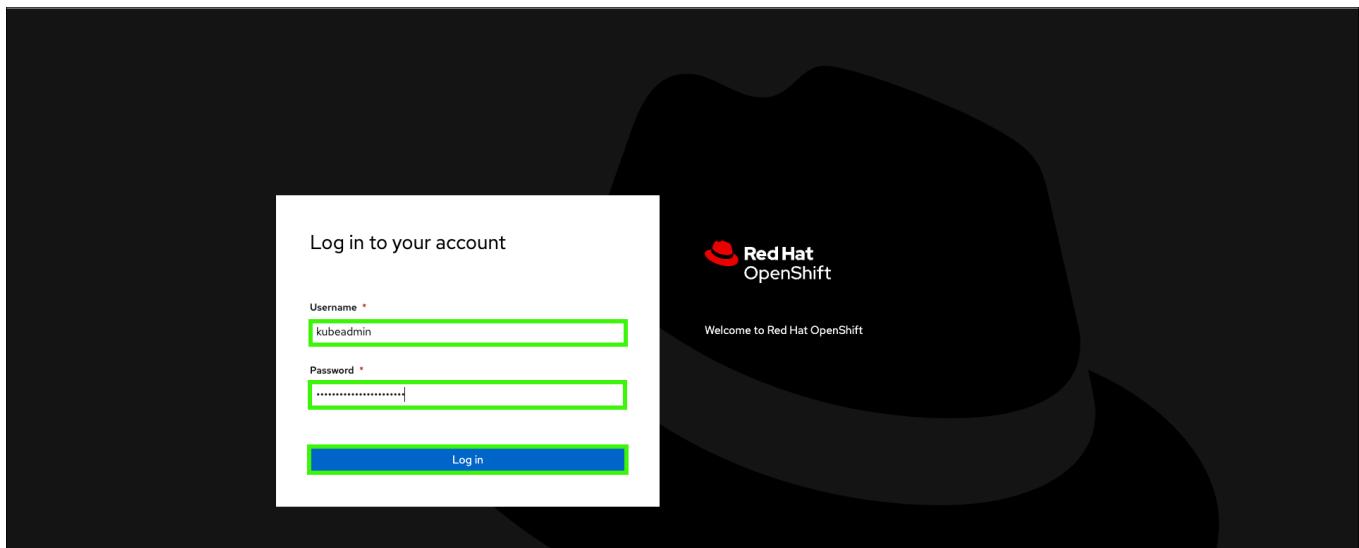
OCP Console
<https://console.openshift-console.apps.672371d38376796fb96a6c4d.ocp.techzone.ibm.com>

OCP Version
4.14

vCenter
itzeu-vc.eu.cloud.techzone.ibm.com

Download kubeconfig [Download](#)

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



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

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Overview

Cluster

Getting started resources [Set up your cluster](#)

Finish setting up your cluster with recommended configurations.
[Add identity providers](#)

Build with guided documentation

Follow guided documentation to build applications and familiarize yourself with key features.

Explore new admin features

Explore new features and resources within the admin perspective.

Quick Starts

- Documentation
- Command Line Tools**
- Share Feedback
- About
- Learning Portal
- OpenShift Blog
- API Explorer

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

The screenshot shows the 'Command Line Tools' section of the Red Hat OpenShift interface. A green dashed box highlights the 'oc - OpenShift Command Line Interface (CLI)' section, which contains the following text and links:

With the OpenShift command line interface, you can create applications and manage OpenShift projects from a terminal.

The oc binary offers the same capabilities as the kubectl binary, but it is further extended to natively support OpenShift Container Platform features.

- [Download oc for Linux for x86_64](#)
- [Download oc for Mac for x86_64](#)
- [Download oc for Windows for x86_64](#)
- [Download oc for Linux for ARM 64](#)
- [Download oc for Mac for ARM 64](#)
- [Download oc for Linux for IBM Power, little endian](#)
- [Download oc for Linux for IBM Z](#)
- [LICENSE](#)

Clicking the preceding link automatically downloads either a **.zip** or **.tar** file specific to your operating system. Extract the file's content. 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           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.

Reservation Details

API URL
<https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443>

Bastion Password [REDACTED]

Bastion RDP Address
api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:43389

Bastion SSH connection
ssh itzuser@api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com -p 40222

Bastion Username
itzuser

Cluster Admin Username
kubeadmin

Cluster Admin Password [REDACTED]

OCP Console
<https://console-openshift-console.apps.672371d38376796fb96a6c4d.ocp.techzone.ibm.com>

OCP Version
4.14

vCenter
itzeu-vc.eu.cloud.techzone.ibm.com

Download kubeconfig

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

Log in to your account

Welcome to Red Hat OpenShift

Red Hat OpenShift

Username *

Password *

Log in

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

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Administrator

Home >

Operators >

Workloads >

Networking >

Storage >

Command Line Tools

Copy login command

oc - OpenShift Command Line Interface (CLI)

With the OpenShift command line interface, you can create applications and manage OpenShift projects from a terminal.

The oc binary offers the same capabilities as the kubectl binary, but it is further extended to natively support OpenShift Container Platform features.

- Download oc for Linux for x86_64

- Click **Display Token**.

Display Token

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

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

Your API token is
sha256~zuWR0KDnkYniIY0m8g8iKoUXPdFFFmou~o4s5FsrDNA

Log in with this token

```
oc login --token=sha256~zuWR0KDnkYniIY0m8g8iKoUXPdFFFmou~o4s5FsrDNA --server=https://api.672b79320c7a71b728e523b4.ocp.techzone.ibm.com:6443
```

Use this token directly against the API

```
curl -H "Authorization: Bearer sha256~zuWR0KDnkYniIY0m8g8iKoUXPdFFFmou~o4s5FsrDNA" "https://api.672b79320c7a71b728e523b4.ocp.techzone.ibm.com:6443/v1/users/~"
```

Request another token

[Logout](#)

A context menu is open over the token value, with 'Copy' highlighted.

9. Open a command prompt or terminal window on your local workstation.

10. Paste the login command and press **enter**.

```
andrewjones@Andrews-MBP ~ % oc login --token=sha256~mJ4L8K6cUMyNyk2Z69KMm3vbP1sWc8SW0eeOdVqtA94 --server=https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443
Logged into "https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443" as "kube:admin" using the token provided.
You have access to 70 projects, the list has been suppressed. You can list all projects with 'oc projects'
Using project "default".
andrewjones@Andrews-MBP ~ %
```

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
```

```
andrewjones@Andrews-MBP ~ % oc login --token=sha256~mJ4L8K6cUMyNyk2Z69KMm3vbP1sWc8SW0eeOdVqtA94 --server=https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443
Logged into "https://api.672371d38376796fb96a6c4d.ocp.techzone.ibm.com:6443" as "kube:admin" using the token provided.
You have access to 70 projects, the list has been suppressed. You can list all projects with 'oc projects'
Using project "default".
andrewjones@Andrews-MBP ~ % mkdir watsonxAssistant
andrewjones@Andrews-MBP ~ % cd watsonxAssistant
andrewjones@Andrews-MBP watsonxAssistant %
```

Install IBM Certificate Manager on Red Hat OpenShift

1. In a text editor, create a file that is 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**.

The screenshot shows the Red Hat OpenShift web console interface. The left sidebar has a navigation menu with items like 'Administrator', 'Home', 'Operators' (which is currently selected and highlighted in green), 'OperatorHub' (which is also highlighted in green), and 'Installed Operators'. The main content area has a blue header bar with the text 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' Below the header, there's a section titled 'Command Line Tools' with a 'Copy login command' button. Another section titled 'oc - OpenShift Command Line Interface (CLI)' provides information about the 'oc' binary and links to download it for various platforms.

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

The screenshot shows the Red Hat OpenShift OperatorHub interface. On the left, there's a navigation sidebar with options like Home, Operators, Workloads, Networking, etc. The 'OperatorHub' section is currently selected. In the main area, there's a dropdown menu labeled 'Project: All Projects'. A green box highlights this dropdown. Below it is a search bar with the placeholder 'Select project...'. There's also a toggle switch for 'Show default projects'. The main content area lists 'Projects' under 'All Projects', including 'default', 'kube-node-lease', and 'kube-public'. At the bottom right, it says '624 items'. Navigation tabs at the bottom include 'Community', 'Marketplace', and 'Community' again.

5. Scroll down and select openshift-marketplace.

This screenshot shows the same interface as the previous one, but with the 'openshift-marketplace' operator selected from the list of installed operators. A green box highlights this selection. The main content area now displays various operators categorized by provider: Community, Marketplace, and Community again. The 'Community' category includes operators like 'CrowdStrike Operator' and 'Abot Operator-v3.0.0'. The 'Marketplace' category includes operators like 'Hazelcast Platform Operator' and 'Accuknox Operator'. The 'Community' category again includes 'Advanced Cluster Management for Kubernetes'.

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 shown in the image below. Select the most current version.

This screenshot shows the Red Hat OpenShift OperatorHub interface with a search query 'IBM Cert Manager' entered into the search bar. A green box highlights the search bar. The results show a single item: 'ibm-cert-manager-4.2.7' under the 'All Items' section. This item is highlighted with a green box. The description for this operator states: 'Operator for managing deployment of cert-manager service.'

7. Click Install.

The screenshot shows the Red Hat OpenShift OperatorHub interface. On the left, a sidebar navigation includes sections like Home, Operators (selected), Workloads, Networking, Storage, Builds, Observe, and Compute. The main content area displays a search bar with 'ibm-cert-manager' and a list of operators. One operator, 'ibm-cert-manager-4.2.7' by IBM, is highlighted. A modal window for this operator is open, showing its details: Channel v4.2, Version 4.2.7, Capability level (Basic Install, Seamless Upgrades selected), and Source ibm-cert-manager-4.2.7. The 'Install' button in the top right of the modal is highlighted with a green box.

8. Keep the default settings and click Install.

The screenshot shows the 'Operator Installation' page for the 'ibm-cert-manager' operator. The left sidebar is identical to the previous screen. The main form has several fields: 'Update channel' set to 'v4.2', 'Version' set to '4.2.7', 'Installation mode' set to 'All namespaces on the cluster (default)', 'Installed Namespace' set to 'Operator recommended Namespace: ibm-cert-manager' (with a note about namespace creation), and 'Update approval' set to 'Automatic'. On the right, there are four cards: 'CR CertificateRequest' (Not available), 'CMC Cert Manager Config' (Not available, with a note about license terms), 'ClusterIssuer' (Not available), and 'Issuer' (Not available). At the bottom are 'Install' and 'Cancel' buttons, with 'Install' being highlighted.

**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 extra assistance is needed, refer to this [site](#).

Locate your existing key or create a new one and continue to the next step.

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

The screenshot shows the IBM Container Software and Cloud Pak Access Management interface. On the left, there's a sidebar with 'My IBM', 'Profile', 'Billing', 'Container Software and Cloud Pak Access Management', and 'Entitlement keys' (which is selected). Below that is 'Container software library'. The main area has a title 'Entitlement keys (1)'. It says 'Access your container software' and explains that an entitlement key allows access to all container software in the IBM Entitled Registry. It shows one active entitlement key issued on October 17, 2022. The key itself is a long string of characters. To the right of the key are 'Copy' and 'Delete' buttons, with 'Copy' highlighted by a green border.

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> .

Mac OS:

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

Microsoft Windows:

```
set IBM_CS_ENT_KEY=<entitlement key>
```

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

Mac OS:

```
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
```

Microsoft Windows:

```
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%
```

The terminal window shows the user running commands to create a namespace and a pull secret. The pull secret command is highlighted with a green border. The output shows the secret was created successfully.

```
andrewjones@Andrews-MBP watsonxAssistant % oc create namespace wxa4z-byos  
namespace/wxa4z-byos created  
andrewjones@Andrews-MBP watsonxAssistant % export IBM_CS_ENT_KEY=  
andrewjones@Andrews-MBP watsonxAssistant % 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  
secret/icr-pull-secret created  
andrewjones@Andrews-MBP watsonxAssistant %
```

6. In a text editor, create a file that is 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
```

```
andrewjones@Andrews-MBP watsonxAssistant % oc create namespace wxa4z-byos
namespace/wxa4z-byos created
andrewjones@Andrews-MBP watsonxAssistant % export IBM_CS_ENT_KEY=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE2NjYwMTk1ODAsImp0aSI6IjNkOWUyMzjZTAzMDQzMzVhNTJhYTkzMWNmOTcyMDR1In0.4M3XRDXzkHMSOkFNJ4uKVcWZ6SnEA0Z03eL_11A2xY
andrewjones@Andrews-MBP watsonxAssistant % 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
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.operator.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.

The screenshot shows the Red Hat OpenShift OperatorHub interface. The left sidebar has 'OperatorHub' selected. The search bar at the top contains 'wxa4z-byos'. The main area displays a grid of operator cards. One card is highlighted with a green border and labeled 'IBM Watsonx Assistant for Z Operator Catalog'. Other visible cards include 'CrowdStrike Falcon Operator', 'Abot Operator-v3.0.0', 'Accuknox Operator', and 'Advanced Cluster Management for Kubernetes'.

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 that shown in the image below.

The screenshot shows the Red Hat OpenShift OperatorHub interface. The search bar at the top contains 'ibm watsonx'. The main area displays a grid of operator cards. One card is highlighted with a green border and labeled 'IBM Watsonx Assistant for Z Operator Catalog'. Other visible cards include 'IBM Watsonx Assistant for Z Operator Catalog' (without a green border), 'IBM Watsonx Assistant for Z Operator Catalog', and 'IBM Watsonx Assistant for Z Operator Catalog'.

10. Click **Install**.

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

You are logged in as a temporary administrator.

Project: wxa4z-byos

OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog provided by IBM.

All Items

Search: ibm watsonx

IBM watsonx Assistant for Z Operator Catalog

provided by IBM

IBM watsonx Assistant for Z Operator

Channel: stable

Version: 2.1.0

Capability level: Basic Install

- Seamless Upgrades
- Full Lifecycle
- Deep Insights
- Auto Pilot

Source: IBM watsonx Assistant for Z Operator Catalog

Provider: IBM

Repository: N/A

Container image: N/A

Created at: Oct 4, 2024, 1:31 PM

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

You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

OperatorHub > Operator Installation

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel*: stable

Version*: 2.1.0

Installation mode*:

- All namespaces on the cluster (default)
Operator will be available in all Namespaces
- A specific namespace on the cluster
Operator will be available in a single Namespace only. (selected)

Installed Namespace*: wxa4z-byos

Update approval*:

- Automatic (selected)
- Manual

Install **Cancel**

**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.**

The screenshot shows the Red Hat OpenShift web console interface. On the left, there's a sidebar with various navigation options like Home, Operators, Workloads, Networking, Storage, Builds, Observe, Compute, User Management, and Administration. The 'Operators' section is expanded, and 'OperatorHub' is selected. In the main content area, there's a message at the top: 'You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.' Below this, a card displays the status of an operator: 'IBM watsonx Assistant for Z' (version v2.0.1) is listed as 'ibm-wxa4z-operator:v2.0.1 provided by IBM'. A green checkmark icon is next to the card. Below the card, the text 'Installed operator: ready for use' is displayed, followed by two buttons: 'View Operator' and 'View installed Operators in Namespace wxa4z-byos'. A green dashed box surrounds the entire 'Installed operator' message and its components.

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

Mac OS and Microsoft Windows:

```
oc project wxa4z-byos
```

Mac OS:

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

Microsoft Windows:

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

```
andrewjones@Andrews-MBP watsonxAssistant % oc create namespace wxa4z-byos
namespace/wxa4z-byos created
andrewjones@Andrews-MBP watsonxAssistant % export IBM_CS_ENT_KEY=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJJQk0gTWFya2V0cGxhY2UiLCJpYXQiOjE2NjYwMTk1ODAsImp0aS16IjNkOWUyNzZjZTAzMDQzMzNhNTJhYTkzMWNmOTcyMDR1In0.4M3XRD4XzkHMSOkFNJ4uKVcWZ6SnEA0Z03el_11A2xY
andrewjones@Andrews-MBP watsonxAssistant % 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
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.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 '{"imagePullSecrets": [{"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

- In a text editor, create a file that is 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 that is 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 that is 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 that is 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:

```

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.

Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
da80909aab1563ff0c15b7c6d	Completed	0/1	0	ds80909aab1563ff0c15b7c6da81 a8bbefabaa605065bb6c6b490e87 0e87f0a96f	-	-	Nov 6, 2024, 9:26 AM
ibm-wxa4z-operator-catalog-cfvsv	Running	1/1	0	CS ibm-wxa4z-operator-catalog	25.0 MiB	0.004 cores	Nov 6, 2024, 9:25 AM
ibm-wxa4z-operator-controller-manager-556fcf98bb-4bhmk	Running	2/2	0	RS ibm-wxa4z-operator-controller-manager-556fcf98bb	142.8 MiB	0.023 cores	Nov 6, 2024, 9:26 AM
wxa4z-byos-cluster-0	Running	1/1	0	SS wxa4z-byos-cluster	1,095.8 MiB	0.017 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-1	Running	1/1	0	SS wxa4z-byos-cluster	1,169.1 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-2	Running	1/1	0	SS wxa4z-byos-cluster	1,113.3 MiB	0.015 cores	Nov 6, 2024, 10:22 AM
wxa4z-client-ingestion-7f98d86c58-9bzth	Running	1/1	0	RS wxa4z-client-ingestion-7f98d86c58	501.8 MiB	0.071 cores	Nov 6, 2024, 12:18 PM
wxa4z-opensearch-wrapper-5cb879f5fb-qw7qt	Running	1/1	0	RS wxa4z-opensearch-wrapper-5cb879f5fb	547.1 MiB	0.031 cores	Nov 6, 2024, 10:22 AM
wxa4z-snapshot-setup-job-nsqlz	Completed	0/1	0	U wxa4z-snapshot-setup-job	-	-	Nov 6, 2024, 10:22 AM

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

Name	Status	Ready	Restarts	Owner	Memory	CPU	Created
da80909aab1563ff0c15b7c6d	Completed	0/1	0	ds80909aab1563ff0c15b7c6da81 a8bbefabaa605065bb6c6b490e87 0e87f0a96f	-	-	Nov 6, 2024, 9:26 AM
ibm-wxa4z-operator-catalog-cfvsv	Running	1/1	0	CS ibm-wxa4z-operator-catalog	25.0 MiB	0.004 cores	Nov 6, 2024, 9:25 AM
ibm-wxa4z-operator-controller-manager-556fcf98bb-4bhmk	Running	2/2	0	RS ibm-wxa4z-operator-controller-manager-556fcf98bb	141.2 MiB	0.022 cores	Nov 6, 2024, 9:26 AM
wxa4z-byos-cluster-0	Running	1/1	0	SS wxa4z-byos-cluster	1,095.8 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-1	Running	1/1	0	SS wxa4z-byos-cluster	1,169.1 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-2	Running	1/1	0	SS wxa4z-byos-cluster	1,113.3 MiB	0.015 cores	Nov 6, 2024, 10:22 AM
wxa4z-client-ingestion-7f98d86c58-9bzth	Running	1/1	0	RS wxa4z-client-ingestion-7f98d86c58	534.6 MiB	0.087 cores	Nov 6, 2024, 12:18 PM
wxa4z-opensearch-wrapper-5cb879f5fb-qw7qt	Running	1/1	0	RS wxa4z-opensearch-wrapper-5cb879f5fb	547.1 MiB	0.030 cores	Nov 6, 2024, 10:22 AM
wxa4z-snapshot-setup-job-nsqlz	Completed	0/1	0	U wxa4z-snapshot-setup-job	-	-	Nov 6, 2024, 10:22 AM

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

Name	Status	Location	Service
wxa4z-client-ingestion	Accepted	https://wxa4z-client-ingestion-wxa4z-byos.apps.672b79320c7a71b728e523b4.ocp.techzone.ibm.com	wxa4z-client-ingestion
wxa4z-opensearch-wrapper	Accepted	https://wxa4z-opensearch-wrapper-wxa4z-byos.apps.672b79320c7a71b728e523b4.ocp.techzone.ibm.com	wxa4z-opensearch-wrapper

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. Use the URL from the network route (with **/v1/query**) appended. Use **admin** for the **Username** and the **Password** will be the password that you specified in the `wrapper-creds.yaml` file.



Don't recall how to set the customer search URL?

Refer back to [Creating an assistant and configuring conversational search](#) if you don't remember how to specify the customer search URL.

3. Test your assistant and verify that it is still answering questions that are related to IBM Z.

Experiment with different prompts and validate that the answers provided are reasonable, and that you can view the documentation that was sourced. If responses are not received as expected, verify that the URL is formatted correctly and you specified the `wrapper-creds.yaml` password as the **admin** password.

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 (bring-your-own-documents (BYOD)). BYOD demonstrates how clients can augment their assistant's conversational search by creating an internal knowledge base with their documentation. Using the client's documentation allows the assistant to provide valuable responses to a range of questions not possible with the default documentation available.

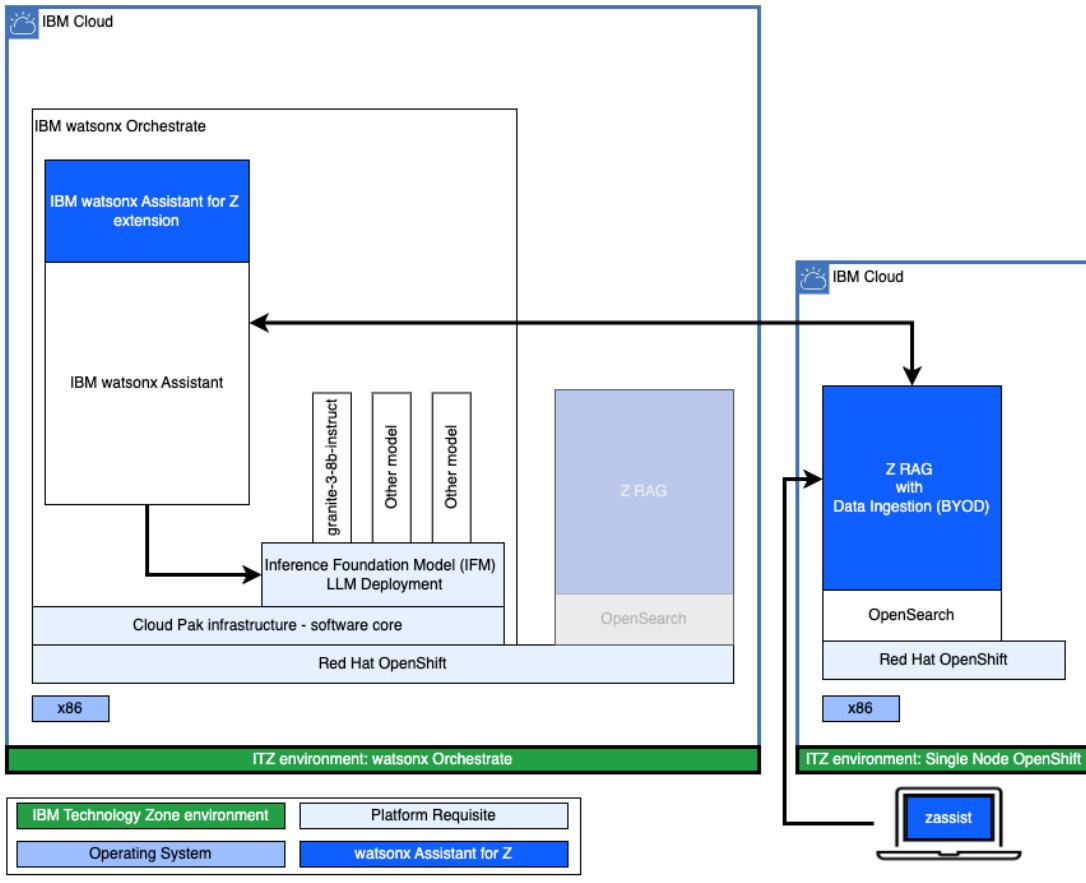
As an example, a client mentioned that their developers often need reference material on company-specific legacy code or company-specific syntax. The users must search through volumes of documentation to find it or look at old code. Also, a need for their operational support group to quickly determine how to resolve technical issues using runbooks exists.

You can show your client how watsonx Assistant for Z can assist developers and operational support personnel in finding answers about internal processes for code development and deployment.

Currently, only PDF, HTML, and DOCX file formats are supported for ingestion.

A high-level, logical architecture of the environment is illustrated in the following diagram.

Installing and using zassist to ingest client documents



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 to download and install **zassist**.

i 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>

The screenshot shows a file browser window titled 'Box@IBM'. Inside, there is a single file entry: 'zassist.zip' with a yellow folder icon. Below the file name, it says 'Modified Sep 5 at 9:15 AM' and 'Size --'. At the top right of the browser, there are buttons for 'Download' (highlighted in green) and 'Account'.

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

The screenshot shows a file browser window titled 'watsonxAssistant'. The main pane displays the contents of the 'zassist' folder, which has been extracted from 'zassist.zip'. The files listed are: byos.yaml, ca.crt, catalogCertManager.yaml, catalogSource.yaml, client-ingestion-secret.yaml, os-secret.yaml, linux, mac (which contains zassist), mac-arm, windows, and zassist.zip. The 'zassist' file within the 'mac' folder is highlighted with a red box. The 'zassist' folder itself is also highlighted with a red box. The browser interface includes standard file management icons at the top.

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 that it is working.

The screenshot shows a terminal window titled 'watsonxAssistant -- zsh -- 157x40'. The user runs the command 'zassist' and receives an error message: 'zassist: error: expected one of "version", "init", "login", "ingest", "load", ...'. The terminal window has a light gray background and a dark gray header bar.

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 the 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. Download the `BYOD.zip` file.

[BYOD.zip](#)

What is in the sample client documentation?

Three sample documents are included:

- `Mainframe_COBOL_Error_Codes.pdf`

This is a document containing company-specific mainframe COBOL error codes for their application.

Developers within the organization typically review this document to quickly diagnose issues based on the application error codes returned.

- `Mainframe_Operational_Incidents_Log.pdf`

This document is leveraged by the organization's operational support team and contains historical records of production-level incidents that occurred. For each incident, there's a record of what the incident was, the date, how it was resolved and who was involved in resolving the incident.

- `COBOL-CICS-to-Java-Internal-Framework.pdf`

This document is leveraged by the development team and contains details about the organization's internal framework for developing applications consisting of legacy COBOL CICS interoperating with new Java code. Within the document contains company-specific coding practices and code syntax that the developers frequently reference.

2. Extract the `BYOD.zip` file.

3. Change to the `BYOD` directory.

4. Set the `TLS_VERIFY` environment variable to `false`.

Mac OS:

```
export TLS_VERIFY=false
```

Windows OS:

```
set TLS_VERIFY=false
```

5. Initialize the `zassist` environment.

```
zassist init
```

6. Retrieve the server URL for the client ingestion server.

```
echo https://$(oc -n wxa4z-byos get route wxa4z-client-ingestion -o jsonpath=".spec.host")"
```

The output of this command is your unique URL for your client ingestion server.

 If the command doesn't work for you...

You can retrieve the URL in your OCP Web console by navigating to Networking Routes and then copy the URL for the wxa4z-client-ingestion route.

7. Retrieve the client-ingestion-authkey .

```
oc -n wxa4z-byos get secret client-ingestion-authkey -o jsonpath=".data.authkey" | base64 -d
```

The output of this command is your unique auth-key that you had previously set. You will need the output of both previous commands in the next step.

 If the command doesn't work for you...

You can find your authkey value by viewing the client-ingestion-secret.yaml file you created and copying the value set for the authkey parameter.

8. Login to your server. Replace <server url> with the value from step 6.

```
zassist login <server url>
```

9. When prompted, enter the password from step 7. Verify that a Success message is received.

10. Verify zassist is ready to ingest documents by checking the status.

```
zassist status
```

```
andrewjones@Andrews-MBP BYOD % export TLS_VERIFY=false
andrewjones@Andrews-MBP BYOD % zassist init
andrewjones@Andrews-MBP BYOD % echo https://$(oc -n wxa4z-byos get route wxa4z-client-ingestion -o jsonpath=".spec.host")
https://wxa4z-client-ingestion-wxa4z-byos.apps.67850b0240c6211133ba8749.ocp.techzone.ibm.com
andrewjones@Andrews-MBP BYOD % oc -n wxa4z-byos get secret client-ingestion-authkey -o jsonpath=".data.authkey" | base64 -d
[REDACTED]
andrewjones@Andrews-MBP BYOD % zassist login https://wxa4z-client-ingestion-wxa4z-byos.apps.67850b0240c6211133ba8749.ocp.techzone.ibm.com
Enter Password: *
Success.
Gathering local files...
Local Ingested Loaded Path
yes no no dev/COBOL-CICS-to-Java-Internal-Framework.pdf
yes no no dev/Mainframe_COBOL_Error_Codes.pdf
yes no no ops/Mainframe_Operational_Incidents_Log.pdf
andrewjones@Andrews-MBP BYOD %
```

11. Ingest the documentation using the commands.

For the next steps, you must be in the root directory called `BYOD`.

```
zassist ingest dev
```

```
zassist ingest ops
```

12. Upload the ingested documents.

```
zassist load dev
```

```
zassist load ops
```

13. Verify that all documents were successfully ingested and loaded.

```
zassist status
```

```
andrewjones@Andrews-MBP BYOD % zassist ingest dev
Gathering local files...
Comparing uploaded files with local files...
Uploaded 2 documents
Started 2 embedding jobs
Completed 2 embedding jobs
andrewjones@Andrews-MBP BYOD % zassist ingest ops
Gathering local files...
Comparing uploaded files with local files...
Uploaded 1 documents
Started 1 embedding jobs
Completed 1 embedding jobs
andrewjones@Andrews-MBP BYOD % zassist load dev
Gathering local files...
Loaded 2 documents into Opensearch database
andrewjones@Andrews-MBP BYOD % zassist load ops
Gathering local files...
Loaded 1 documents into Opensearch database
andrewjones@Andrews-MBP BYOD % zassist status
Gathering local files...
Local Ingested Loaded Path
yes yes yes dev/COBOL-CICS-to-Java-Internal-Framework.pdf
yes yes yes dev/Mainframe_COBOL_Error_Codes.pdf
yes yes yes ops/Mainframe_Operational_Incidents_Log.pdf
andrewjones@Andrews-MBP BYOD %
andrewjones@Andrews-MBP BYOD %
```

Verify that the assistant is using the ingested documents

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

1. Enter the following prompt in your assistant and **record the response** (cut and paste into a text file on your local machine).

The customer application is failing with ERR-CBL-001, what does this internal error mean?

The screenshot shows the 'Preview assistant' tab in the IBM Watsonx Orchestrate interface. The assistant is named 'Zeeves'. The user's message is: 'The customer application is failing with ERR-CBL-001, what does this internal error mean?'. The AI response is: 'Hi! I'm Zeeves. How can I help you today?'. A green box highlights the user's message, and a blue box highlights the AI response.

2. Click the **Down arrow** to view the citations for the response.

The screenshot shows the same interface after clicking the down arrow. The AI response now includes a detailed citation: 'The customer application is encountering the internal error ERR-CBL-001, which means that a null value was encountered in a mandatory field. This error typically occurs when a required field in the application is left blank or contains a null value. It is important to investigate and resolve this issue as soon as possible to ensure the proper functioning of the application.' A green box highlights this citation.

3. Click **View source** for the **Mainframe_COBOL_Error_Codes-...** citation.

The screenshot shows the IBM Watsonx Assistant interface. At the top, there are tabs for 'IBM Watsonx Orchestrate', 'AI assistant builder', and 'Zeeves'. Below the tabs are buttons for 'Copy link to share', 'Change layout', 'Change background', and 'Customize web chat'. The main area is titled 'Zeeves' and shows a message from 'You 1:57 PM': 'The customer application is failing with ERR-CBL-001, what does this internal error mean?'. Below this, there are two code snippets. The first snippet is titled 'CDBL0001' and contains: 'The z/TPF debugger works as a client in debug sessions, where TPF Toolkit is listening for the requests from the z/TPF debugger. pgmname The name of the program that the error originated...'. It has a 'View source' button. The second snippet is titled 'Mainframe_COBOL_Error_Codes....' and contains: 'ERR-CBL-001: Null value encountered in mandatory field.' It also has a 'View source' button. A note at the bottom says 'AI Accuracy of generated answers may vary.'



Take note of the order of the response citations!

- Accept the security risk to view the source document for any ingested document cited.

The steps to accept the security risk for the document are not shown as it varies by the browser you are using. 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.

The screenshot shows a browser window with a list of error codes. A green dashed box highlights the title 'Company-Specific Mainframe COBOL Error Codes' and the first few items in the list:

- List of Error Codes
- ERR-CBL-001: Null value encountered in mandatory field.
- ERR-CBL-002: Overflow in numeric computation.
- ERR-CBL-003: File not found in JCL input directory.
- ERR-CBL-004: Duplicate record detected during WRITE operation.
- ERR-CBL-005: Index out of bounds in table processing.
- ERR-CBL-006: Invalid REDEFINE clause detected in structure.
- ERR-CBL-007: SOC4 abend due to invalid memory reference.
- ERR-CBL-008: Division by zero encountered in arithmetic operation.
- ERR-CBL-009: INSUFFICIENT MEMORY: Unable to allocate storage for SORT operation.
- ERR-CBL-010: File status 35: File not opened correctly before access.
- ERR-CBL-011: Data mismatch error in READ INTO clause.
- ERR-CBL-012: Invalid key on SEARCH ALL for indexed file.
- ERR-CBL-013: SOC7 abend: Data exception encountered.
- ERR-CBL-014: Lock timeout in CICS transaction.
- ERR-CBL-015: File status 90: General file handling error.
- ERR-CBL-016: Job failed due to exceeded time allocation (MAXCC=12).
- ERR-CBL-017: Attempt to WRITE beyond maximum file size.
- ERR-CBL-018: Illegal MOVE operation detected on COMP-3 field.
- ERR-CBL-019: Misaligned DD statement in JCL.

- Repeat the preceding steps for the following prompts in your assistant and **record the responses** (cut and paste into a text file on your local machine).

Are there any production incidents that were resolved in relation to Data corruption in the production database. If yes who can I collaborate with to resolve a similar issue today and what are their names?

What specific syntax changes do I need to make in COBOL to call Java using the internal framework? Please provide a detailed explanation.

What is the internal git lab link to execute the Java on z/OS pipeline?

Adjusting the search behavior

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

The screenshot shows the 'Custom service' configuration page in the IBM Watsonx Assistant builder. The 'Instance' tab is selected. In the 'Metadata' section, there is a JSON example provided:

```
{
  "example_field": "example_value",
  "other_example_field": 7
}
```

A green dotted rectangle highlights this JSON example. Below the JSON example, there is a 'Search display text' field containing the message: 'I searched my knowledge base, but did not find anything related to your query'.

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 relies on the default settings, which means OpenSearch searches all 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 want your assistant to only 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 want your assistant to reference only 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.

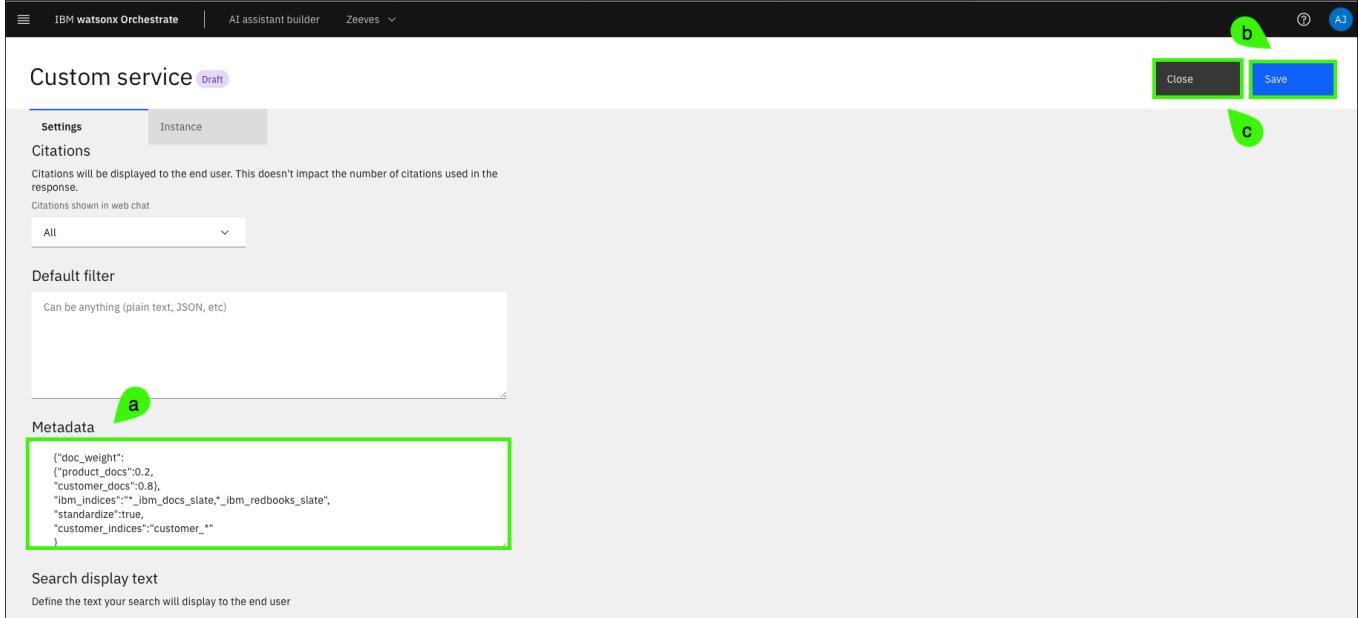
For example:

```
{"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](#).

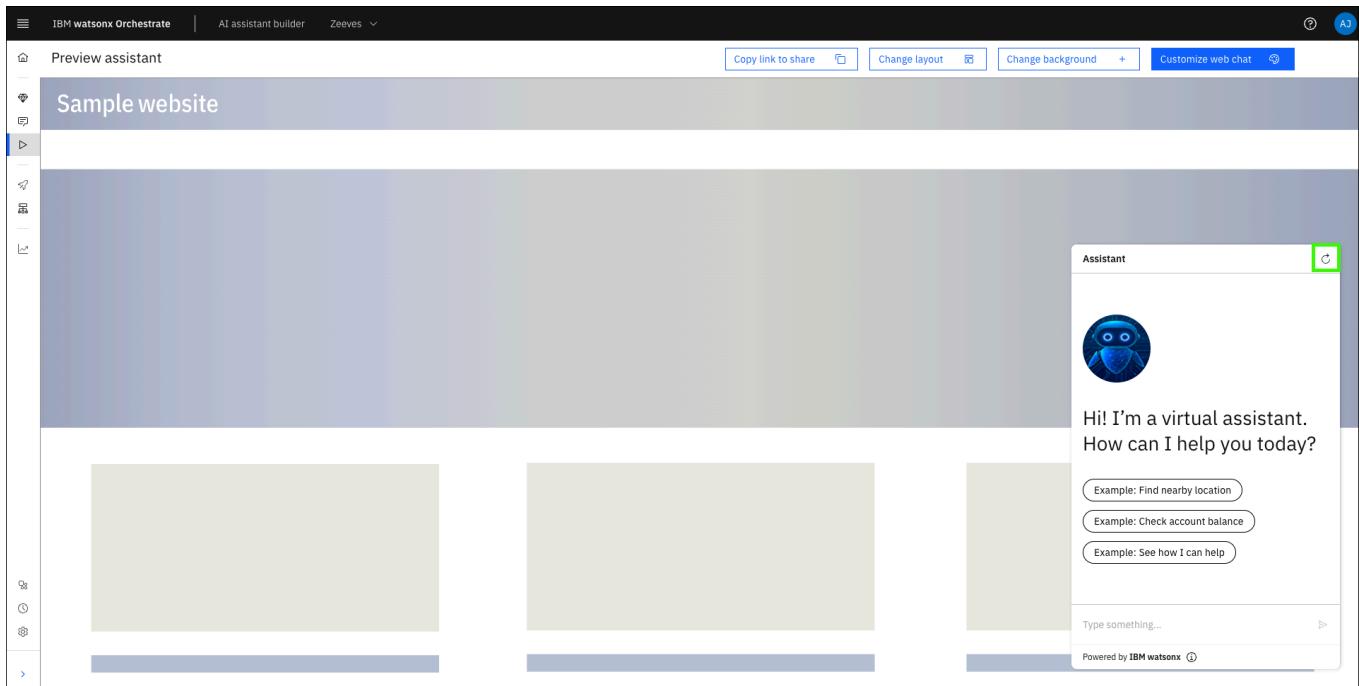
1. Set the **(a) Metadata** field for your BYOS custom search instance to the following value, click **(b) Save**, and then click **(c) Close**. Notice the weight for **customer_docs** is heavier than the weight for **product_docs**.

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



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

3. Click the **Restart conversation** () icon.



4. Repeat the queries four queries run earlier and record the results and the order of the response citations.

Compare the two sets of results. Notice how the answers changed based on the weighting of the ingested documents versus the IBM product documentation. Were the ingested documents always the first document cited? If not, why do you think that is?

Before proceeding, experiment with different metadata and other configuration settings for your custom service instance.

 **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
```

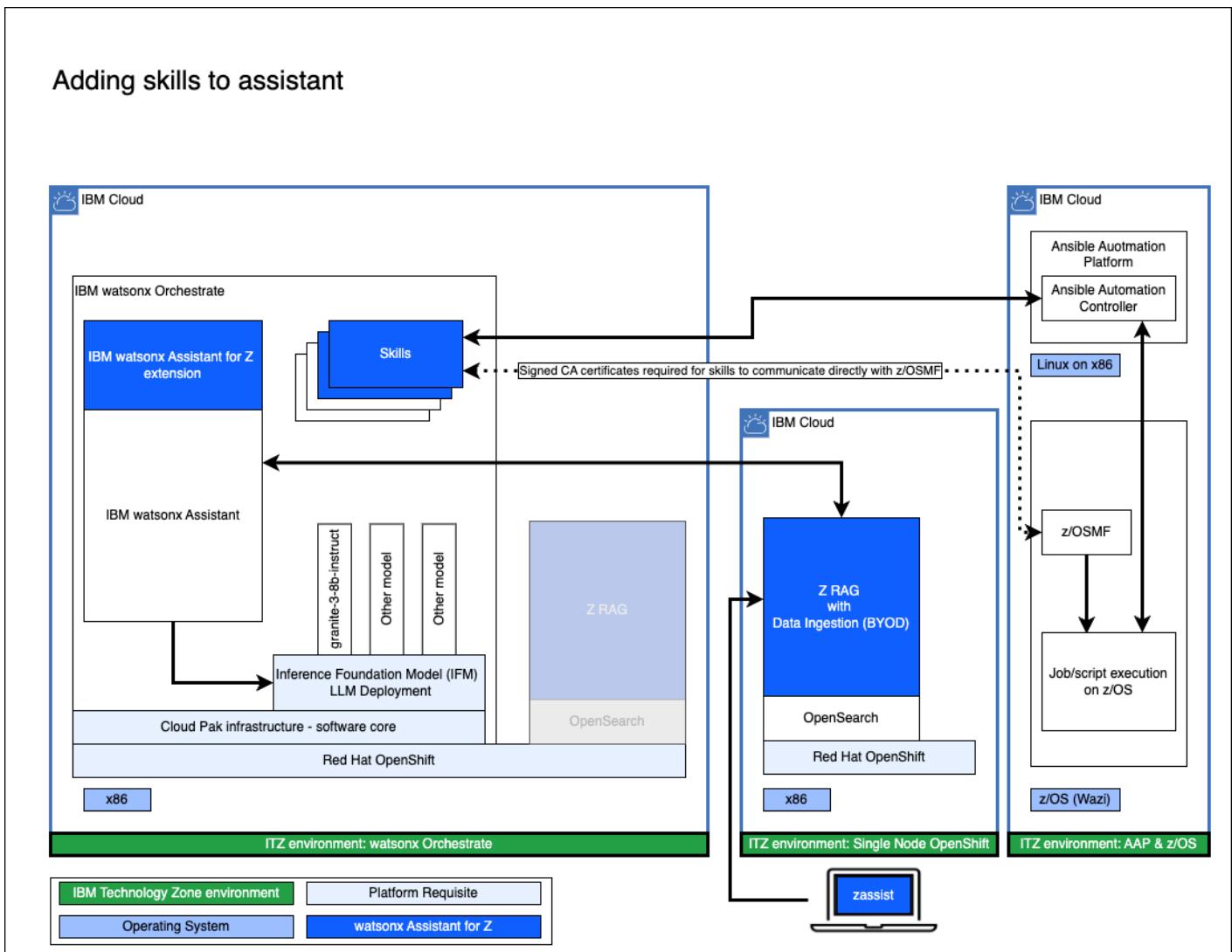
Adding skills to the assistant

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. The Z Skills Accelerator 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](#).

A high-level, logical architecture of the environment is illustrated in the figure that follows.



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.

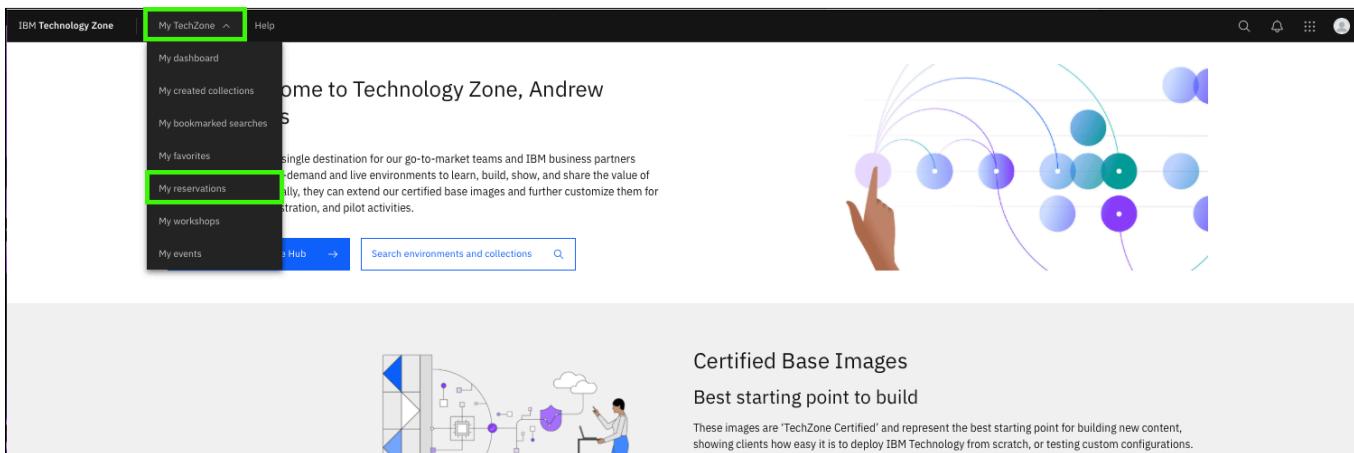
Access the AAP and Wazi as a Service environment

Be sure to record the information as instructed

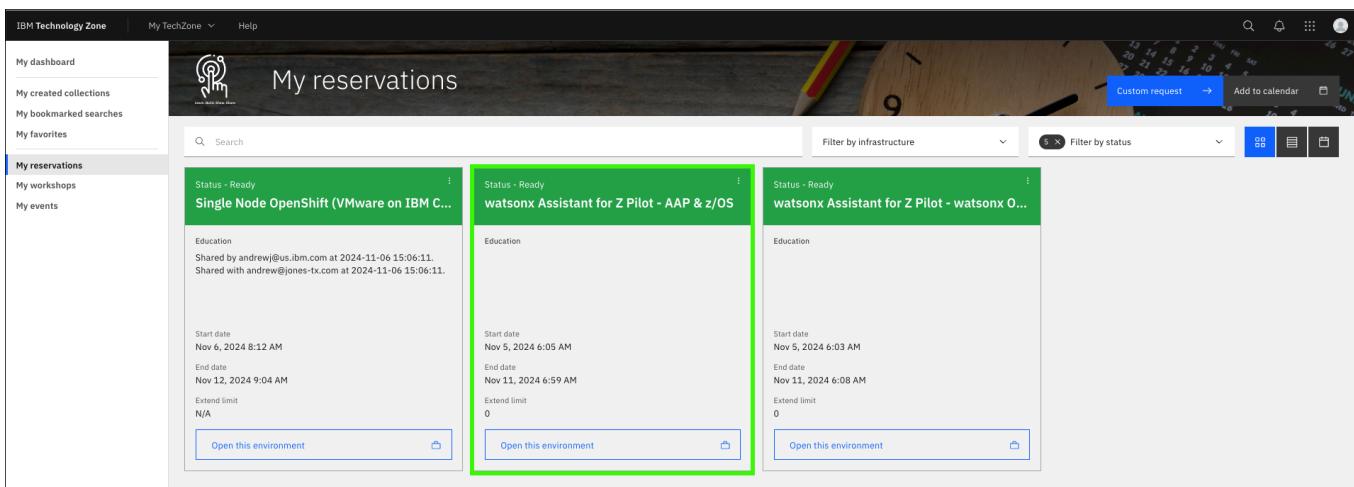
Several of the steps below instruct you to record values from your ITZ reservation. Be sure to do this as they will not only be used in this section, but also in later sections of the lab guide.

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

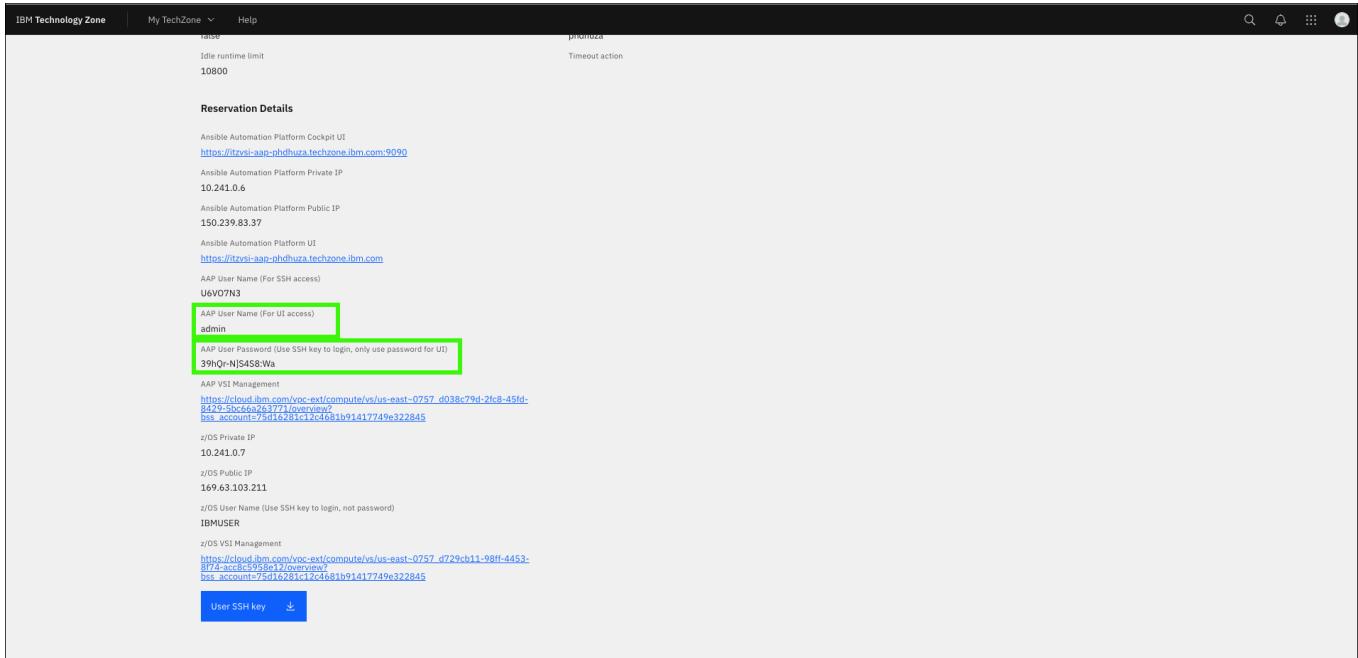
[ITZ My reservations](#)



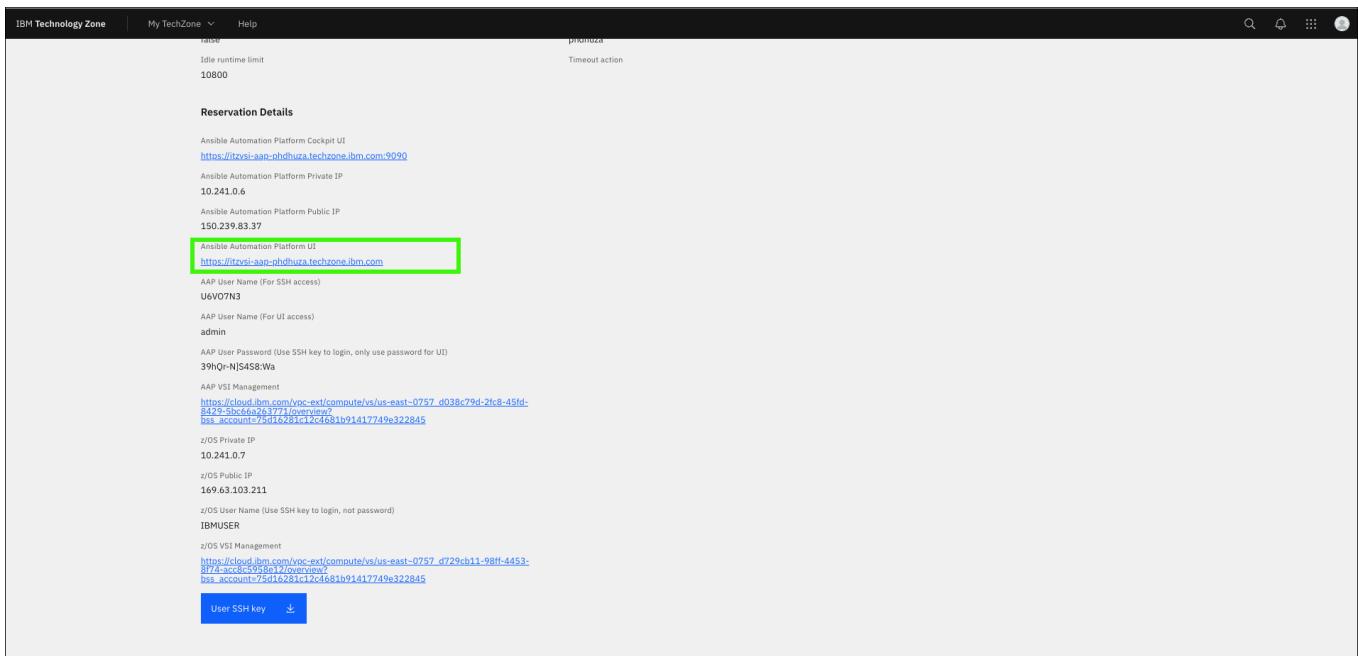
2. Click the **watsonx Assistant for Z Pilot - AAP & z/OS** tile.



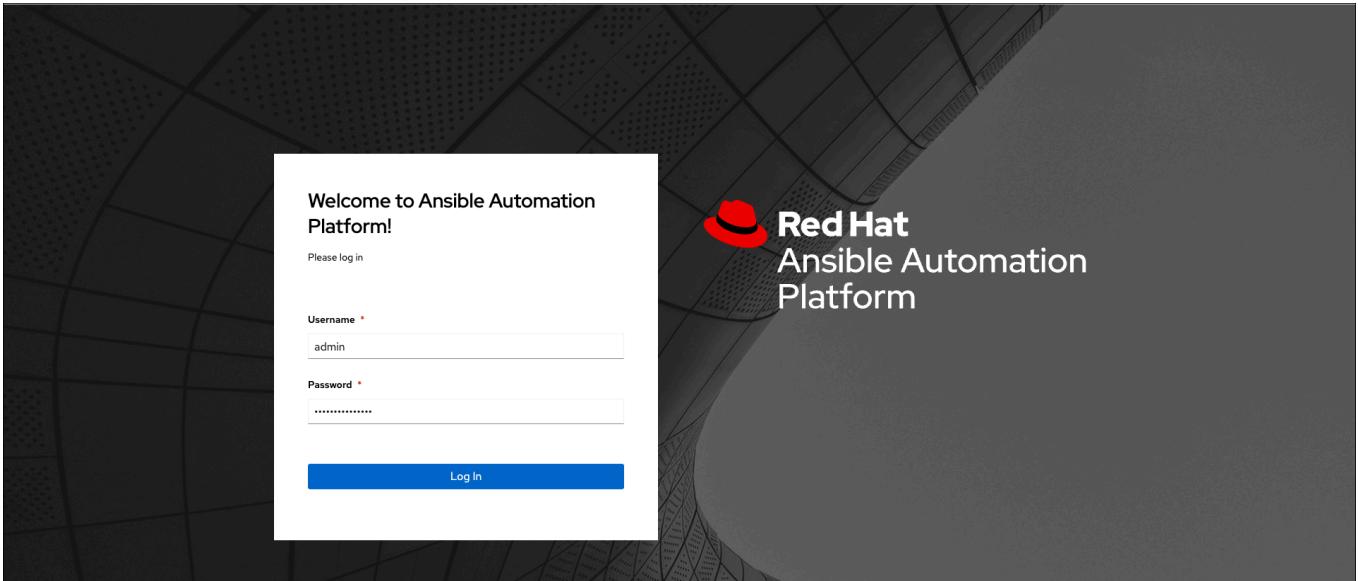
3. Locate and record the **AAP User Name (For UI access)** and **AAP User Password** fields.



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.



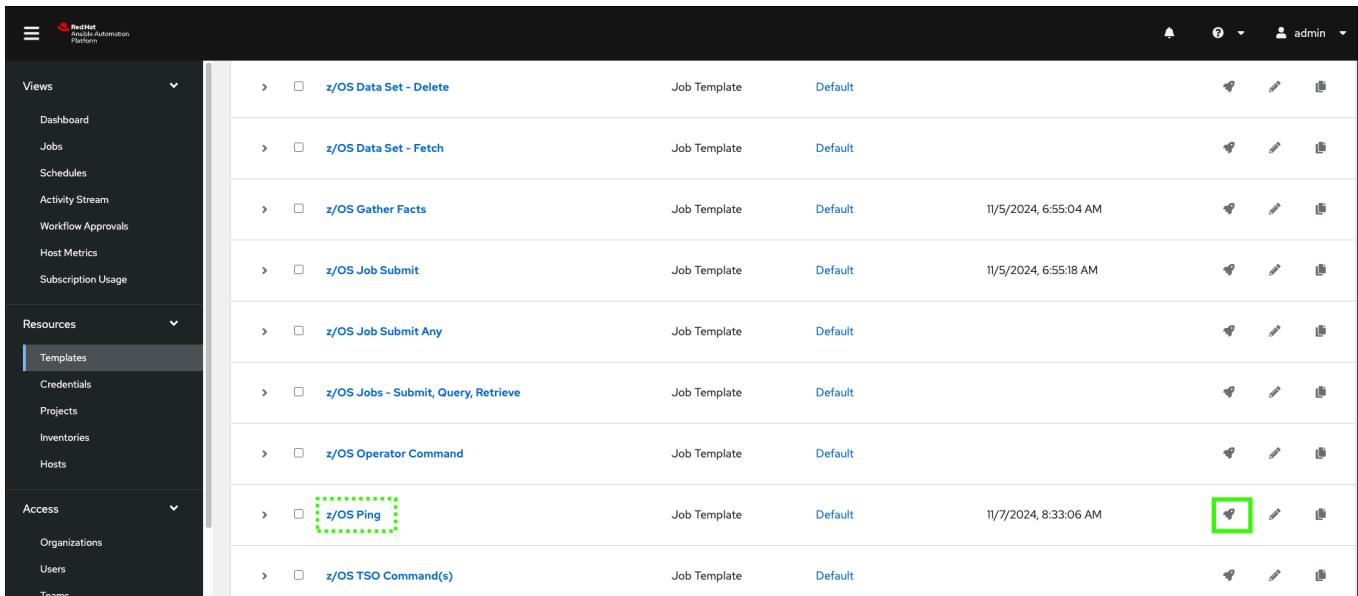
6. Click **Templates** under the **Resources** section.



The AAP instance is preconfigured to the Wazi aaS instance

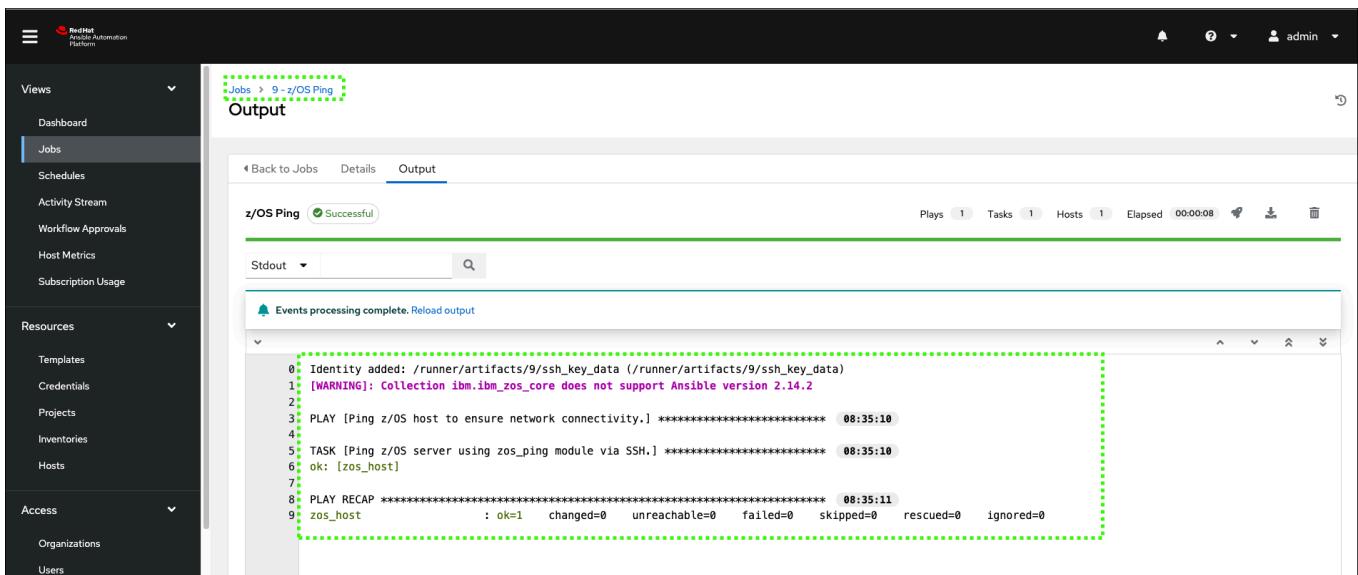
Note that because the AAP instance and the back-end z/OS system are preconfigured, no changes are needed to execute the templates and they will target your provisioned z/OS system by default.

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



Views	z/OS Data Set - Delete	Job Template	Default			
Dashboard	z/OS Data Set - Fetch	Job Template	Default			
Schedules	z/OS Gather Facts	Job Template	Default	1/5/2024, 6:55:04 AM		
Activity Stream	z/OS Job Submit	Job Template	Default	1/5/2024, 6:55:18 AM		
Workflow Approvals	z/OS Job Submit Any	Job Template	Default			
Host Metrics	z/OS Jobs - Submit, Query, Retrieve	Job Template	Default			
Subscription Usage	z/OS Operator Command	Job Template	Default			
Resources	z/OS Ping	Job Template	Default	1/7/2024, 8:33:06 AM		
Templates	z/OS TSO Command(s)	Job Template	Default			
Credentials						
Projects						
Inventories						
Hosts						
Access						
Organizations						
Users						
Teams						

8. Observe the z/OS Ping job run.



Jobs > 9 - z/OS Ping

Output

Events processing complete. Reload output

```

0 Identity added: /runner/artifacts/9/ssh_key_data (/runner/artifacts/9/ssh_key_data)
1 [WARNING]: Collection ibm.ibm_zos_core does not support Ansible version 2.14.2
2
3 PLAY [Ping z/OS host to ensure network connectivity.] **** 08:35:10
4
5 TASK [Ping z/OS server using zos_ping module via SSH.] **** 08:35:10
6 ok: [zos_host]
7
8 PLAY RECAP **** 08:35:11
9 zos_host : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

```

Take time to explore the other templates that are ready to use. Learn more about the automation capabilities [here](#).

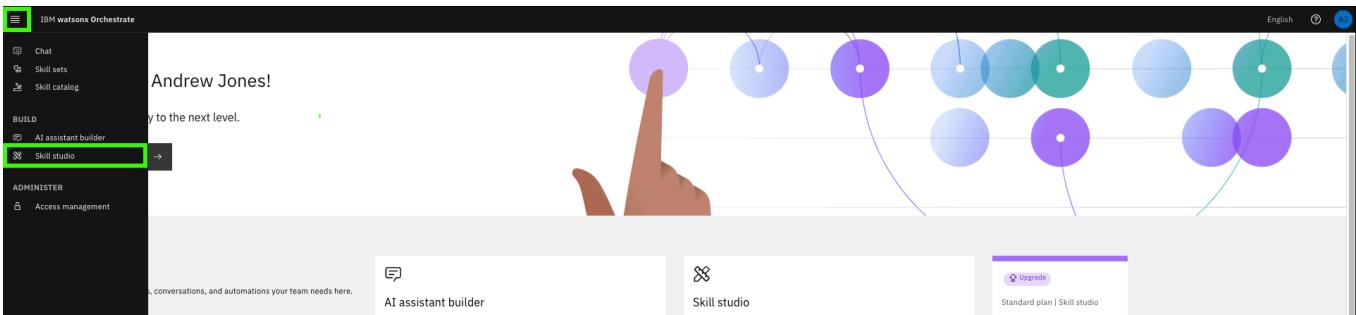
Importing skills from Ansible Automation Platform

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.

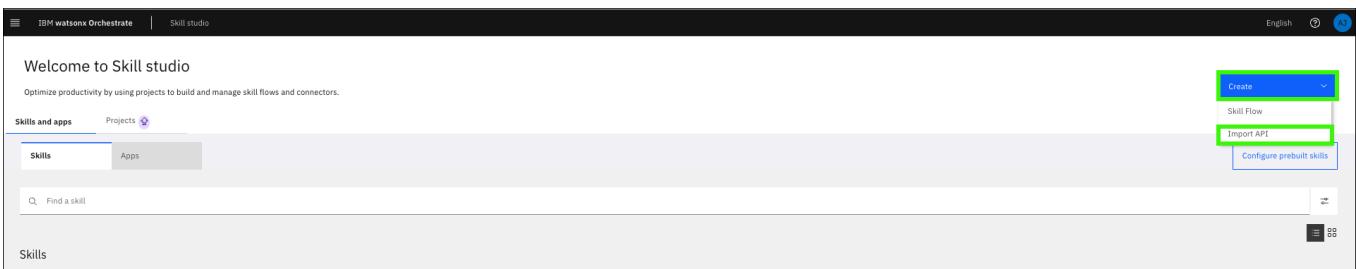
Import skills into your assistant

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](#).

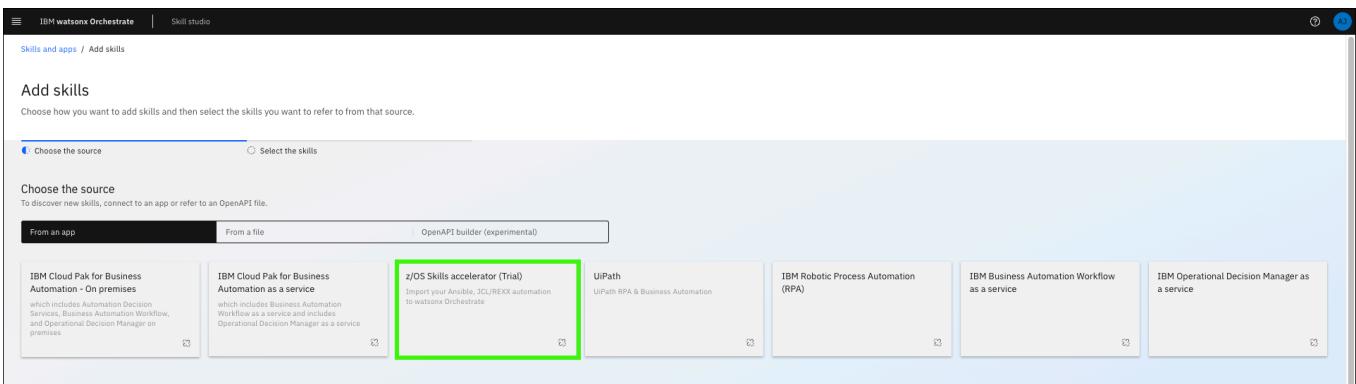
1. Return to your watsonx Orchestrate instance and expand the main menu and click **Skill studio**.



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

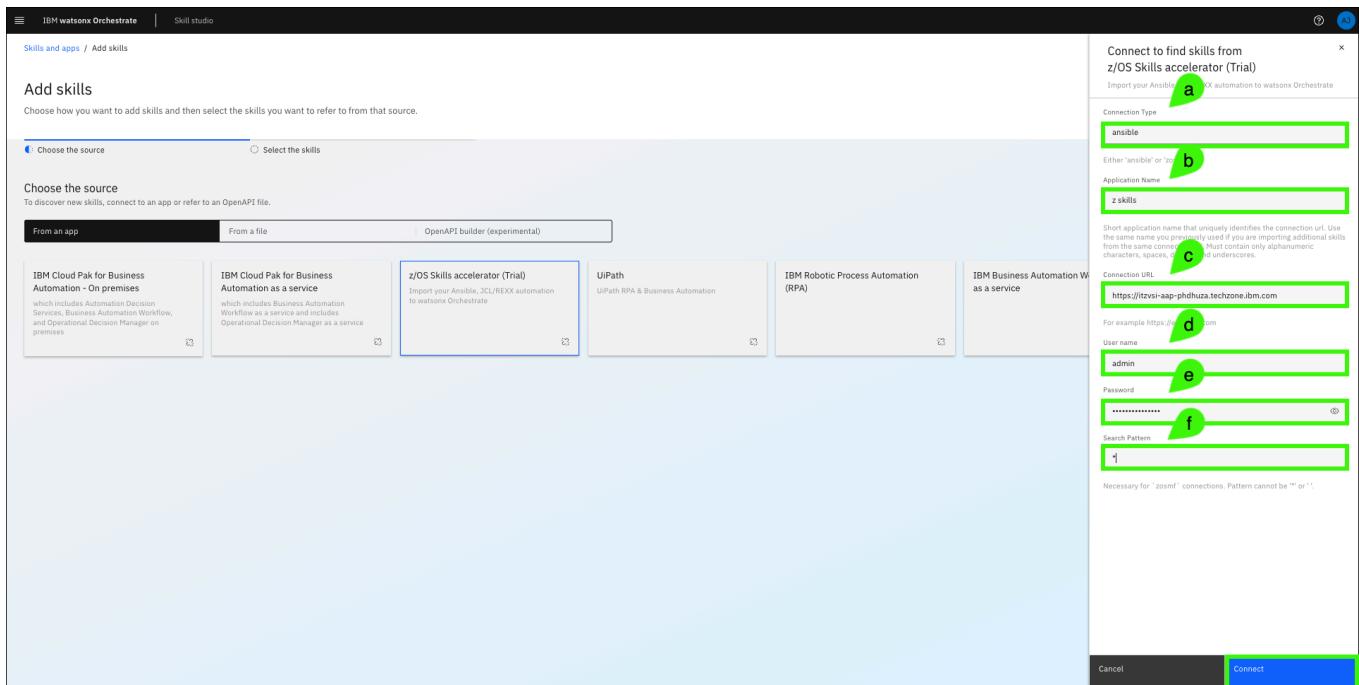
b: Application Name: <enter a meaningful name for the skills that you will import> - record this name, you will need in the next section

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: *

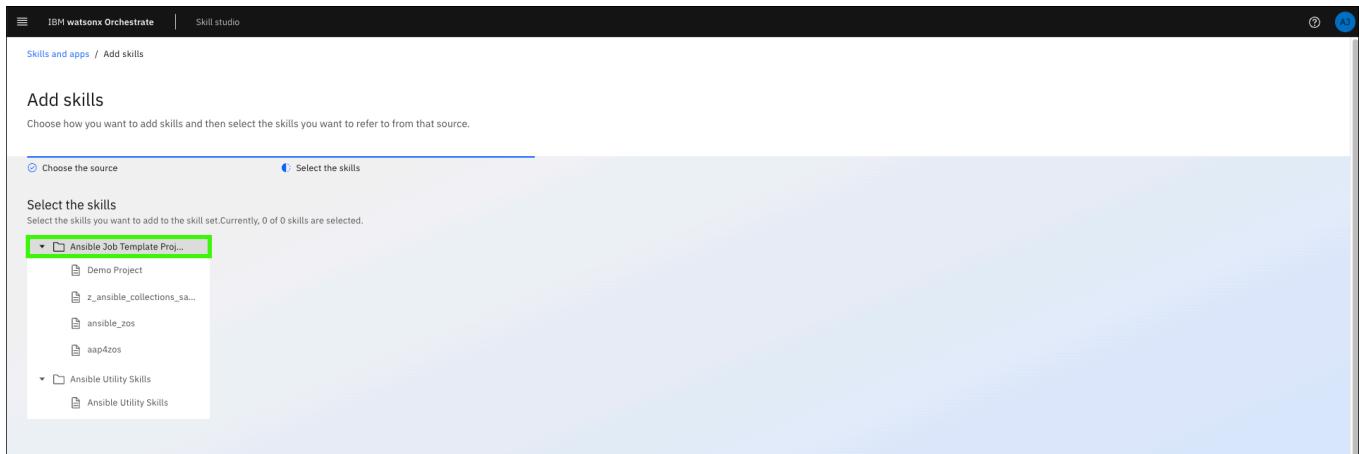


5. Expand the Ansible Job Template Proj... folder.



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**.

Skill	Description	Status
Z/os ping	Z skills - this playbook pings...	Ready to add
Z/os gather facts	Z skills - this sample playbook...	Ready to add
Z/os job submit	Z skills - this playbook shows...	Ready to add
Z/os job submit any	Z skills - a playbook for subm...	Ready to add
Z/os jobs - submit, query, ret...	Z skills - this sample playbook...	Ready to add

7. Select the skills that 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.

Skill	Description	Status
<input checked="" type="checkbox"/> Z/os ping	Z skills - this playbook pings...	Ready to add
<input checked="" type="checkbox"/> Z/os gather facts	Z skills - this sample playbook...	Ready to add
<input type="checkbox"/> Z/os job submit	Z skills - this playbook shows...	Ready to add
<input type="checkbox"/> Z/os job submit any	Z skills - a playbook for subm...	Ready to add
<input type="checkbox"/> Z/os users - add	Z skills - this playbook shows...	Ready to add
<input type="checkbox"/> Z/os data set - basics	Z skills - this sample playbook...	Ready to add
<input type="checkbox"/> Z/os data set - create	Z skills - this playbook creat...	Ready to add
<input type="checkbox"/> Z/os data set - delete	Z skills - this playbook delet...	Ready to add
<input type="checkbox"/> Z/os data set - fetch	Z skills - this playbook fetch...	Ready to add
<input type="checkbox"/> Z/os certs - health checker se...	Z skills - this playbook enabl...	Ready to add
<input type="checkbox"/> Z/os certs - create keyring	Z skills - this playbook delet...	Ready to add
<input type="checkbox"/> Z/os certs - delete keyring	Z skills - this playbook delet...	Ready to add
<input type="checkbox"/> Z/os certs - create cert	Z skills - this playbook creat...	Ready to add
<input type="checkbox"/> Z/os certs - delete cert	Z skills - this playbook delet...	Ready to add

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

Welcome to Skill studio

Optimize productivity by using projects to build and manage skill flows and connectors.

Skills and apps Projects

Skills Apps

Find a skill

Skills

Name	Step in the process	Status	Skill type	Author	Last edited
z/OS Gather Facts	Just 1 step away to be ready	Ready to publish	Imported	andrew@jones-tx.com	November 19 2024
z/OS Ping	Just 1 step away to be ready	Ready to publish	Imported	andrew@jones-tx.com	November 19 2024

Configure prebuilt skills

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.

Skills and apps / Enhance this skill

Enhance the "z/OS Gather Facts" skill

Add details that will make people want to use this skill.

Name Input Output Security Phrases Next best skills

Name and describe this skill in a way that tells users how it's used and why they would want to use it.

Name* z/OS Gather Facts

Description 0/100
z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS-specific information from the z/OS host.

Categories

API version* 1.0

Preview

The skill will look like this in the catalog.

z/OS Gather Facts
z skills - This sample playbook demonstrates the z/OS gather facts...

The skill will look like this in the skill set.

z/OS Gather Facts

Cancel Publish Save as draft

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

Welcome to Skill studio

Optimize productivity by using projects to build and manage skill flows and connectors.

Skills and apps Projects

Skills Apps

Find a skill

Skills

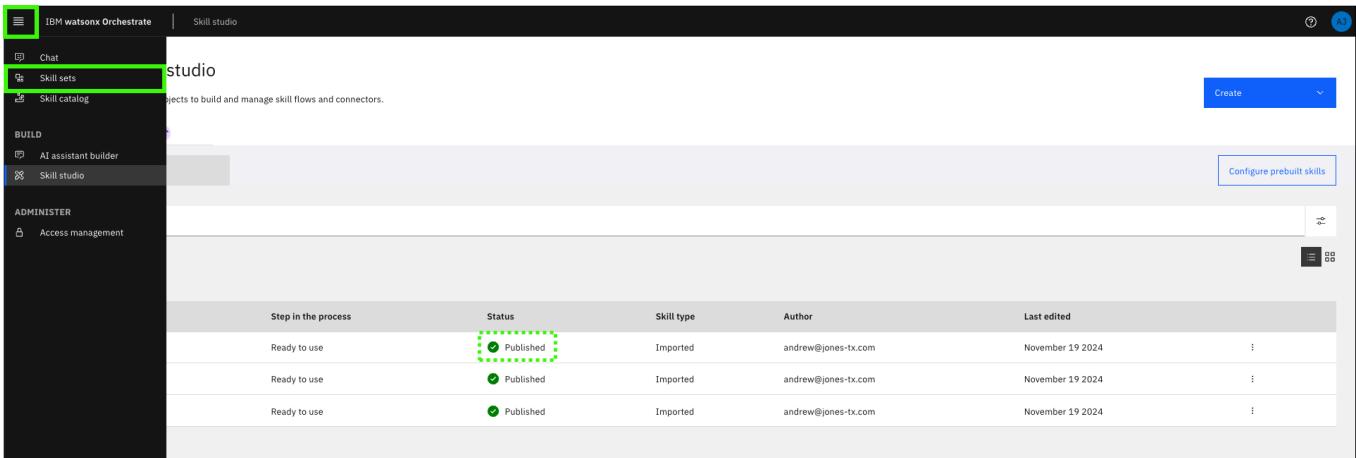
Name	Step in the process	Status	Skill type	Author	Last edited
z/OS Gather Facts	Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
z/OS Ping	Just 1 step away to be ready	Ready to publish	Imported	andrew@jones-tx.com	November 19 2024

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 that are published, the application you created can be connected to your assistant.

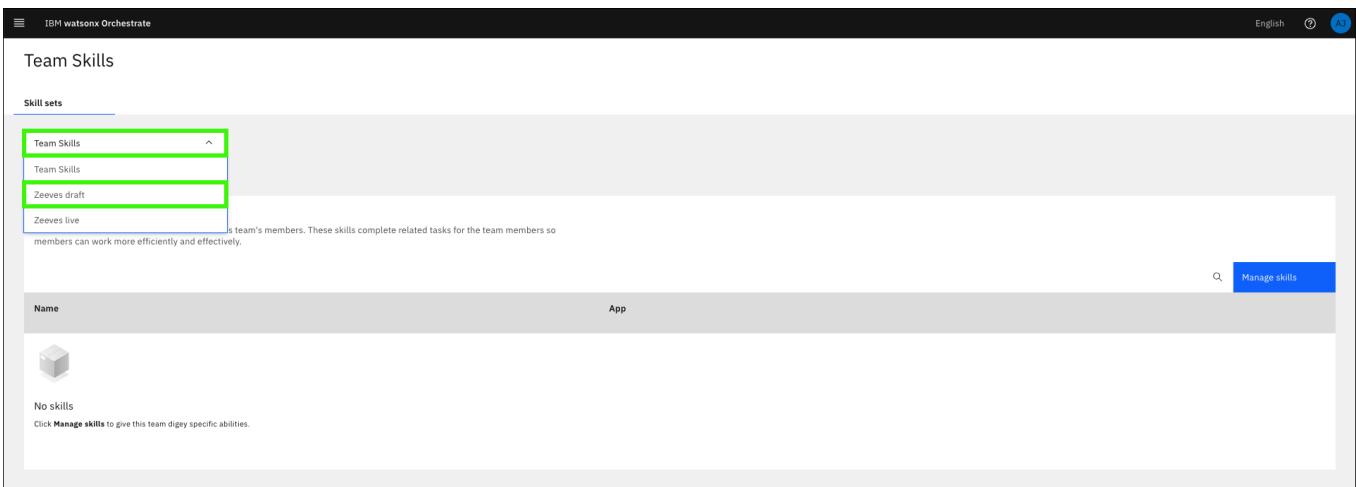
1. Expand the main menu and select **Skill sets**.



The screenshot shows the IBM Watsonx Orchestrate interface with the 'Skill studio' tab selected. In the left sidebar, the 'Skill sets' option is highlighted with a green box. The main area displays a table of published skills:

Step in the process	Status	Skill type	Author	Last edited
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024

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

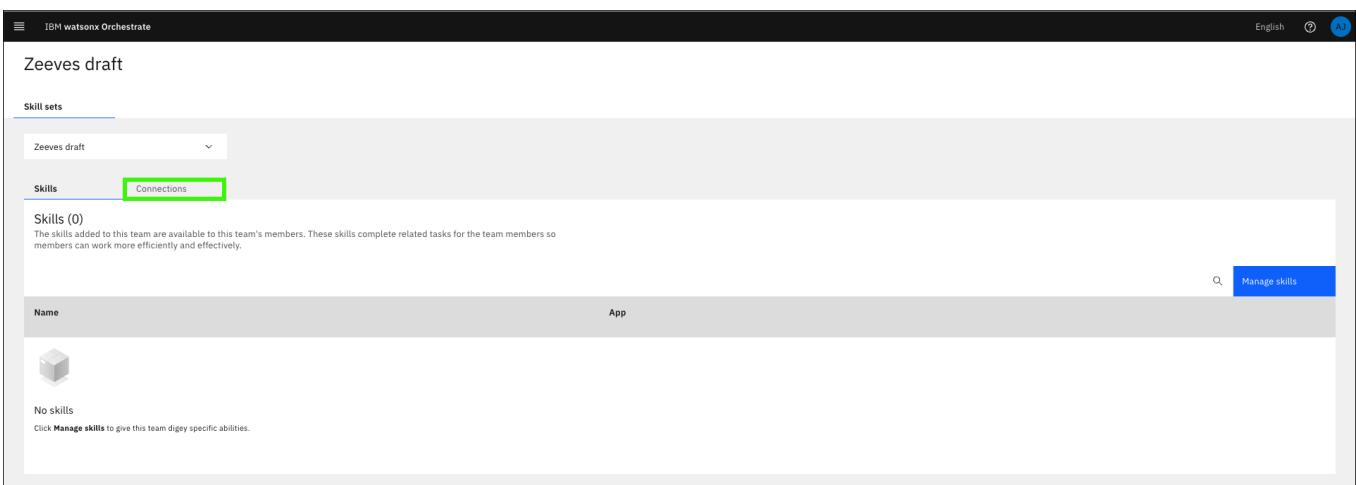


The screenshot shows the 'Team Skills' page. The 'Skill sets' dropdown is open, showing 'Team Skills' selected. Below it, 'Zeeves draft' is highlighted with a green box. The main area displays a table with one row:

Name	App

Below the table, it says 'No skills' and 'Click Manage skills to give this team digest specific abilities.'

3. Click the **Connections** tab.



The screenshot shows the 'Zeeves draft' connections page. The 'Skills' tab is selected, but the 'Connections' tab is highlighted with a green box. The main area displays a table with one row:

Name	App

Below the table, it says 'No skills' and 'Click Manage skills to give this team digest specific abilities.'

4. Click the **Search (🔍)** icon.

Application	Number of skills	Credential type	Connected by ⓘ	Action
Activate or deactivate attracting candidates using ThisWay Global	4	⚠️ Not specified	-	⋮
Adobe Workfront	37	⚠️ Not specified	-	⋮
Alliance Virtual Office	2	⚠️ Not specified	-	⋮
Amazon S3	8	⚠️ Not specified	-	⋮
Amazon SES	10	⚠️ Not specified	-	⋮

Items per page: 5 | 1-5 of 78 items | 1 ⚏ of 16 pages | ⌂ ⌃

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

Application	Number of skills	Credential type	Connected by ⓘ	Action
Ansible Controller Skills - z skills	2	⚠️ Not specified	-	⋮

Items per page: 5 | 1-1 of 1 items | 1 ⚏ of 1 page | ⌂ ⌃

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

Application	Number of skills	Credential type	Connected by ⓘ	Action
Ansible Controller Skills - z skills	2	⚠️ Not specified	-	⋮

Items per page: 5 | 1-1 of 1 items | Connect app | 1 ⚏ of 1 page | ⌂ ⌃

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

Zeeves draft

Skill sets

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Q z skills

Application	Number of skills	Credential type
Ansible Controller Skills - z skills	2	Not specified

Items per page: 5 1-1 of 1 items

Connect to Ansible Controller Skills - z skills

Member credentials
Each team member uses their own credentials to connect to this app and use its skills.

Team credentials
The admin sets the credentials each team member uses to connect to this app and use its skills.

You selected **Team credentials** for the credential type. Click **Connect app** to provide the credentials your team will use and to connect to the app.

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)**.

Zeeves draft

Skill sets

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Q z skills

Ansible Controller Skills - z skills

Items per page: 5 1-1 of 1 items

Connect to Ansible Controller Skills - z skills

a
admin

b

If the service instance uses legacy credentials for authentication, provide the password for the specified username.

Cancel Connect app c

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

Zeeves draft

Skill sets

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Q z skills

Application	Number of skills	Credential type	Connected by	Action
Ansible Controller Skills - z skills	2	Team	andrew@jones-tx.com	⋮

Items per page: 5 1-1 of 1 items

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

Creating actions for your assistant

Now that the skills in your application are connected to your assistant, you are ready to create actions that are tied to those skills. Learn more about building actions [here](#)

Configure the number of input fields

Before configuring actions, it is 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](#).

The screenshot shows the IBM Watsonx Orchestrate web interface. At the top, there's a navigation bar with a profile icon and some settings. Below it, the main area is titled 'Skill sets' and shows a single skill set named 'Zeeves draft'. The 'Skills' tab is selected. On the right side, there's a sidebar with a user profile for 'Andrew Jones' and various links like 'Region', 'Plan', 'Settings' (which is highlighted with a green box), 'Provide feedback', 'Privacy', 'About', and 'Log out'. A green box also highlights the user profile icon in the top right.

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

This screenshot shows the 'Settings' page in the IBM Watsonx Orchestrate interface. The 'Skill configurations' tab is active and highlighted with a green box. Other tabs available are 'Data Retention' and 'Platform languages'. The main content area is titled 'Skill configurations' and contains a note: 'Set default for maximum number of fields to consider a skill to be conversational'. Below this is a text input field labeled 'Number of form fields' with the value '0', which is also highlighted with a green box. There are minus and plus buttons next to the input field.

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

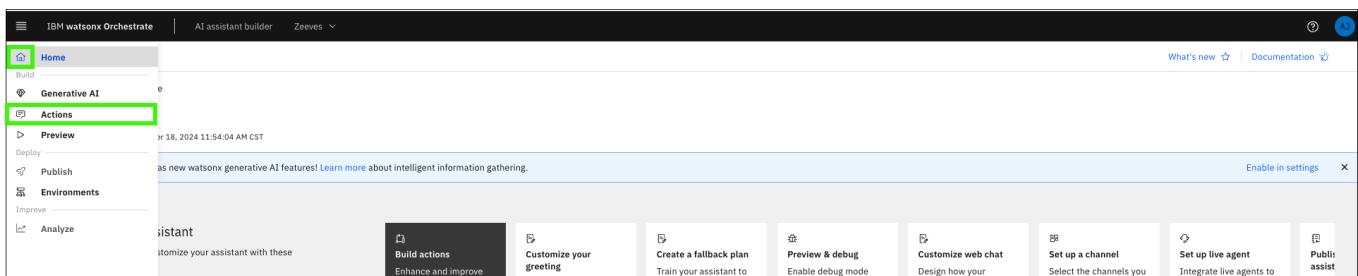
This screenshot is identical to the previous one, showing the 'Settings' page with the 'Skill configurations' tab selected. The 'Number of form fields' input field now contains the value '0', indicating that the user has completed the configuration step.

Create actions

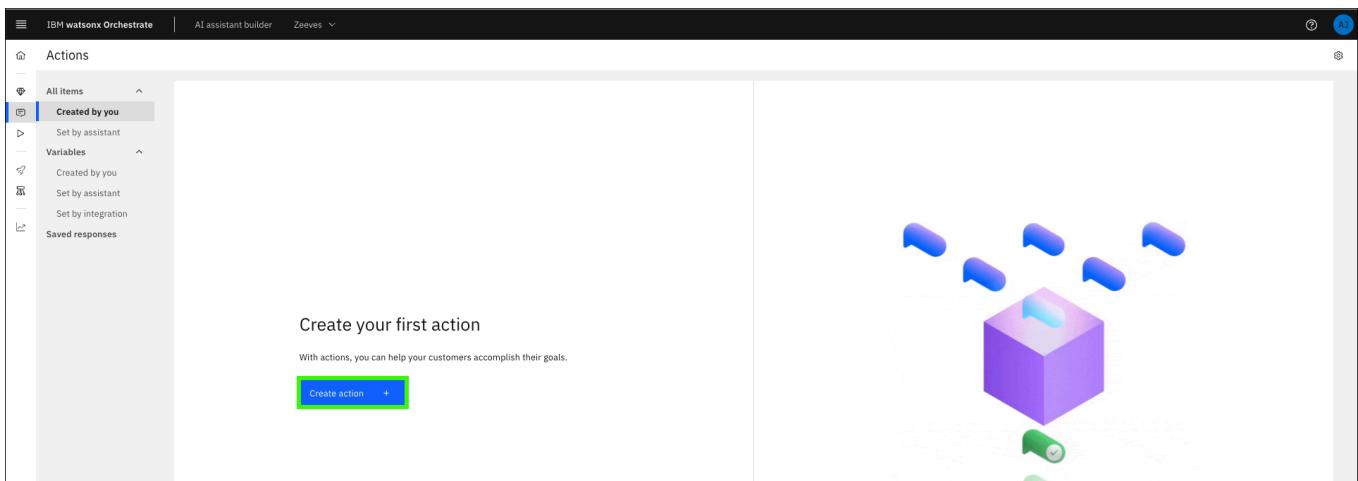
1. Click the main menu and select **AI assistant builder**.



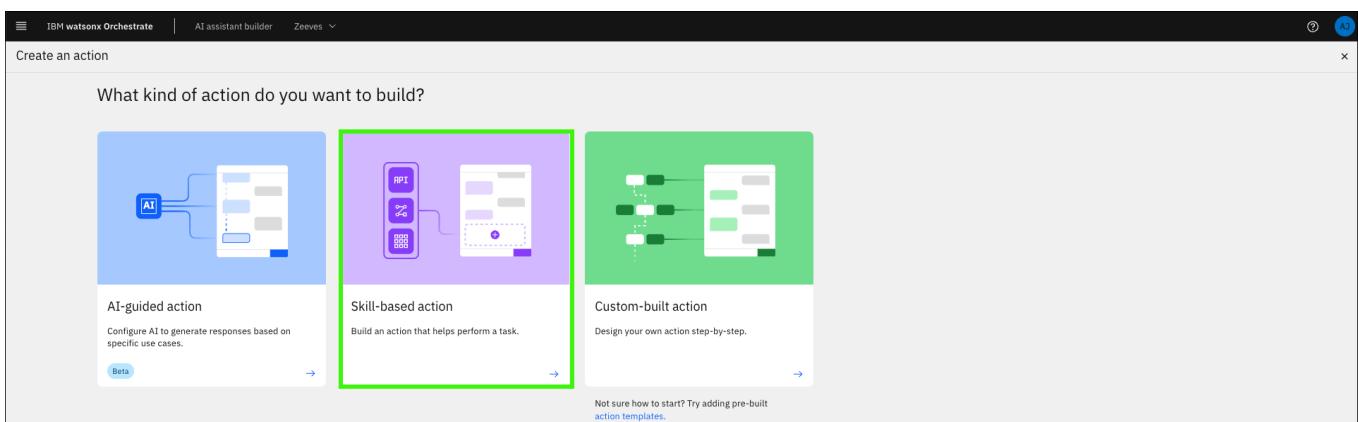
2. Hover over the **Home** icon (🏠) and click **Actions**.



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 that is shown in the **z/OS Gather Facts** tile reflects when you added the skill to your application.

The screenshot shows a list of actions under the heading "Build an action from a skill". The "Select a skill" section includes a search bar and a note: "Choose a conversational skill published as a foundation of your action." Below is a grid of tiles:

z/OS Ping z skills - This playbook pings the z/OS host to test connectivity. Last updated: 2024-11-19T15:58:20.567Z	z/OS Gather Facts z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS-specific information from the z/OS host. Last updated: 2024-11-19T15:56:26.843Z	Summarize the Webex meeting transcript in watsonx.ai Last updated: 2024-11-04T10:49:16.502Z	Summarize the Box content in watsonx.ai Last updated: 2024-11-04T10:49:12.077Z	Summarize a Zendesk ticket in watsonx.ai Last updated: 2024-11-04T10:49:09.476Z
Summarize a ServiceNow incident in watsonx.ai Last updated: 2024-11-04T10:49:05.828Z	Summarize a Salesforce opportunity in watsonx.ai Last updated: 2024-11-04T10:49:01.769Z	Sharepoint document summary in watsonx.ai Last updated: 2024-11-04T10:48:55.707Z	Salesloft email summary in watsonx.ai Last updated: 2024-11-04T10:48:51.522Z	Salesforce case summarization in watsonx.ai Last updated: 2024-11-04T10:48:48.195Z
Salesforce case sentiment analyze	Outlook email summary	Github issue summarization	Github issue sentiment	Generate an email

- 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 that 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 your sample phrases. This issue has been reported to the offering team.

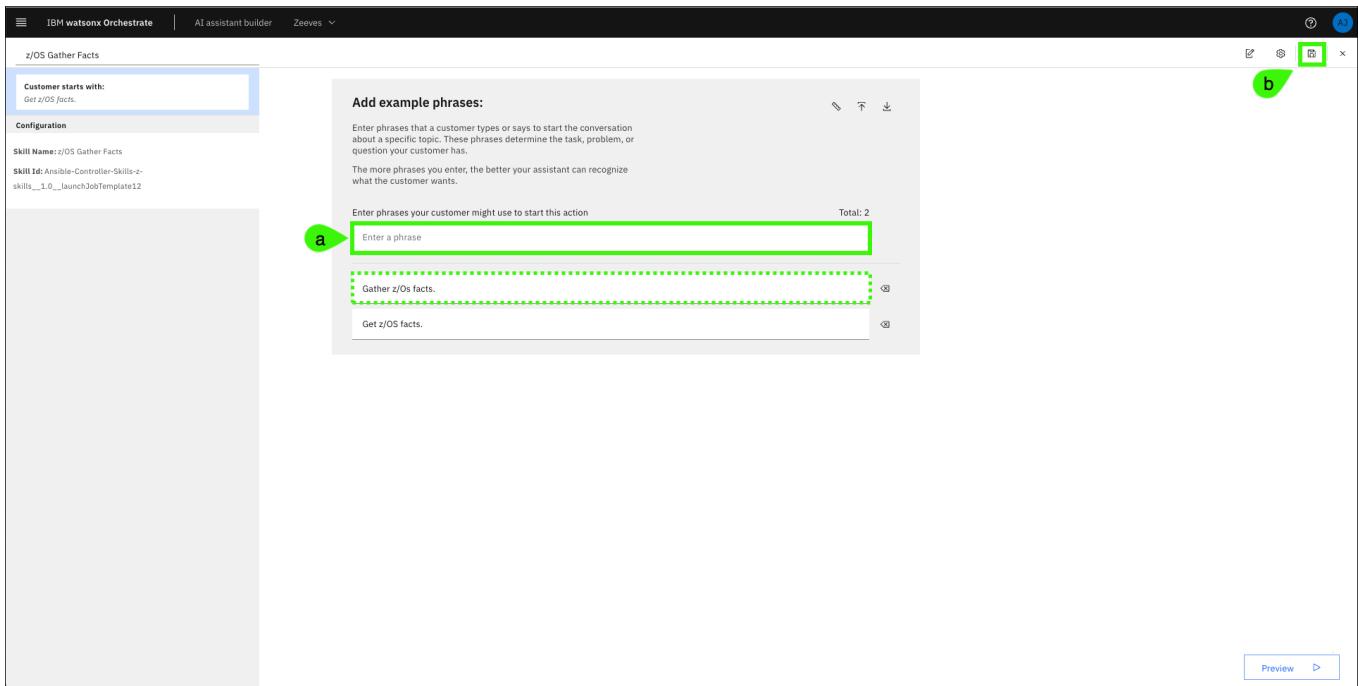
Sample prompts:

Get zOS facts

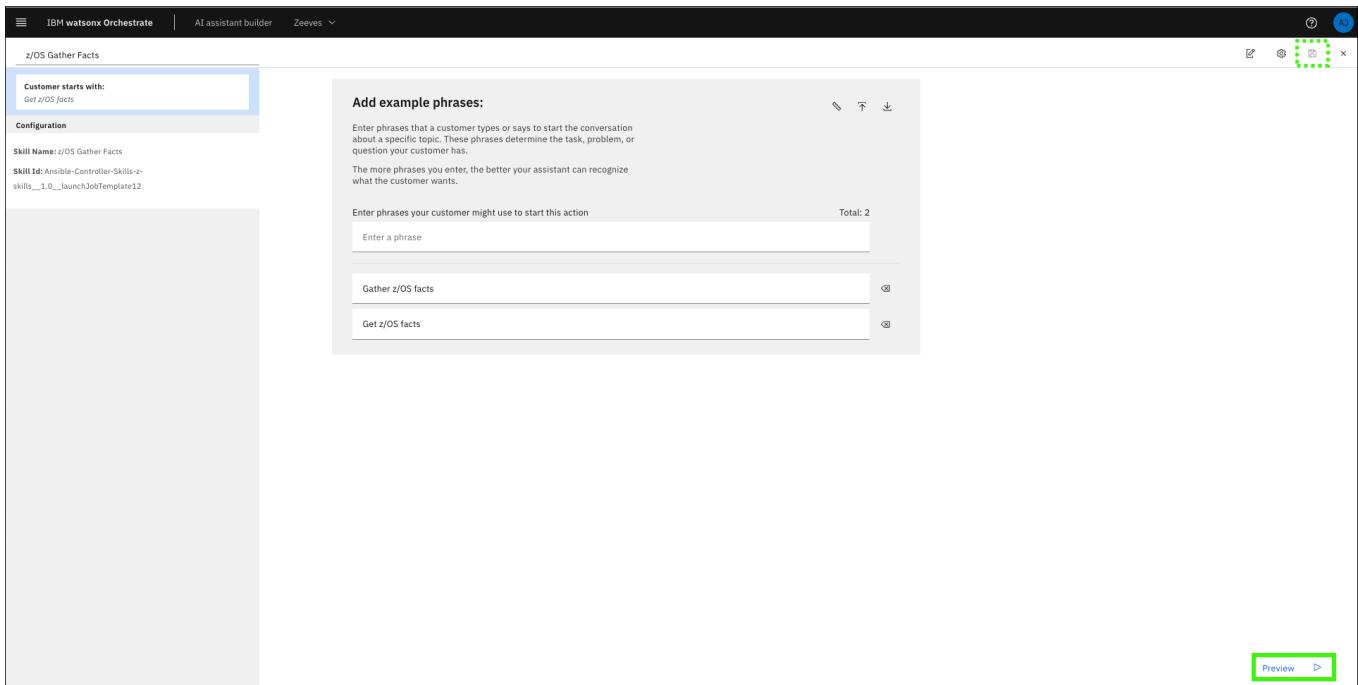
Gather zOS facts

The screenshot shows the "New action" dialog box. The input field for "What does your customer say to start this interaction?" contains the text "Get z/OS facts." The "Save" button at the bottom right is highlighted with a blue background and a callout pointing to it.

- Add any extra prompts (a) and then click the save (b).



8. Click Preview.



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

Prompt:

Get zOS facts

The screenshot shows the configuration interface for a skill named "z/OS Gather Facts". The "Customer starts with:" field contains "Get z/OS facts". The "Add example phrases:" section has two entries: "Gather z/OS facts" and "Get z/OS facts". The "Preview" window shows a conversation where the user says "Get z/OS facts" and the bot responds with "Welcome, how can I assist you?". A green box highlights the user's message "Get z/OS facts".

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

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

The screenshot shows the configuration interface for the same skill. The "Customer starts with:" field still contains "Get z/OS facts". The "Add example phrases:" section now includes "You 1:04 PM Get z/OS facts" (with a green dashed box around it). The "Preview" window shows a conversation where the user says "Get z/OS facts" and the bot responds with "Welcome, how can I assist you?". Below this, a message from the bot at 1:05 PM states "You 1:04 PM Get z/OS facts" (with a green dashed box around it), followed by "1:05 PM Conversational skill called z/OS Gather Facts recognized" and "job : 12 status : pending". A green dashed box also surrounds the message "There are no additional steps for this action. Add a new step or end the action." A note at the bottom says "Use the up arrow for prior messages".

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 that you recorded in the previous step for the **z/OS Gather Facts** skill.

Name	Status	Type	Start Time	Finish Time	Actions
12 - z/OS Gather Facts	Successful	Playbook Run	12/5/2024, 7:17:32 AM	12/5/2024, 7:17:41 AM	

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.

```

25 }
26
27 TASK [Print out all gathered facts about the z/OS host.] **** 07:17:38
28 ok: [zos_host] => {
29     "ansible_facts": {
30         "arch_level": "2",
31         "cpc_nd_manufacturer": "IBM",
32         "cpc_nd_model": "A00",
33         "cpc_nd_plant": "C1",
34         "cpc_nd_seqno": "237701828347",
35         "cpc_nd_type": "008562",
36         "edt": "00",
37         "hw_name": "...",
38         "ieasym_card": "(00,K2)",
39         "io_config_id": "00",
40         "iodate": "...",
41         "iodesc": "...",
42         "iodf_config": "DEFAULT",
43         "iodf_name": "PROV.IODF00",
44         "iodf_unit_addr": "DE28",
45         "ioproc": "...",
46         "iotime": "..."
}

```

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 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.

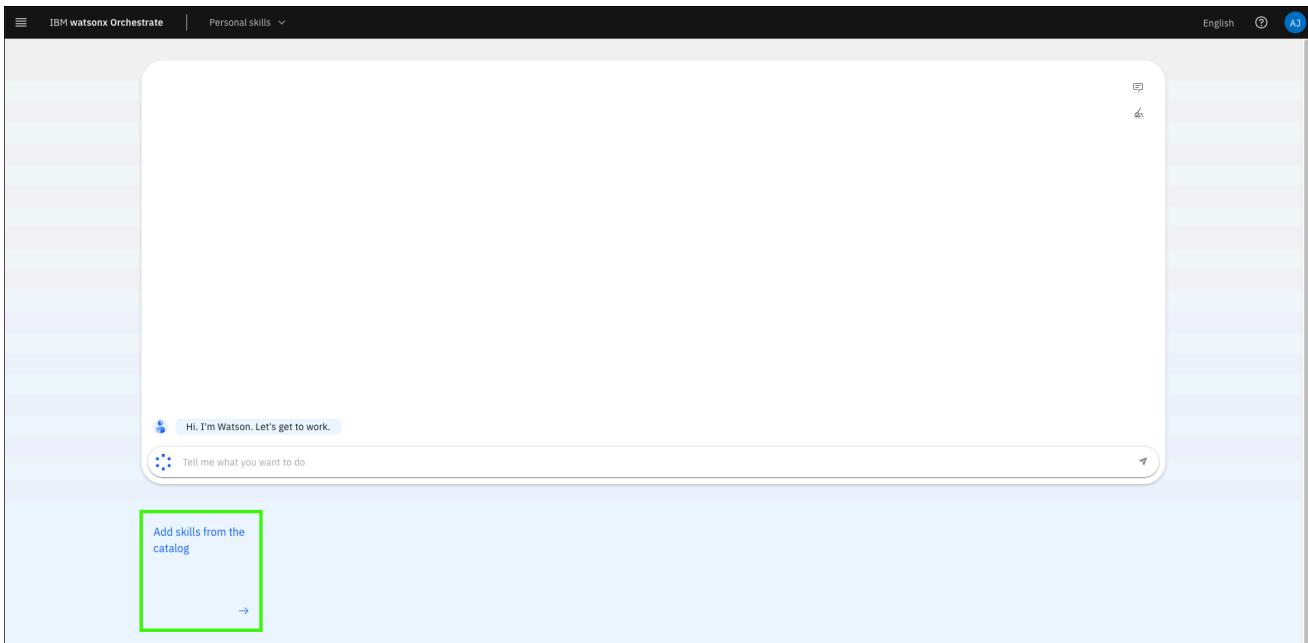
The screenshot shows the AI assistant builder interface. On the left, there's a sidebar with sections like Configuration, Customer starts with:, and Add example phrases:. Under Configuration, the Skill Name is set to "z/OS Gather Facts" and the Skill Id is "Ansible-Controller-Skills-z-skills_1.0_launchJobTemplate1". In the main area, there are two input fields: "Enter phrases your customer might use to start this action" containing "Get z/OS facts" and "Gather z/OS facts". To the right, a "Preview" window shows a simulated chat session. The customer says "Welcome, how can I assist you?", and the bot responds with "Gather z/OS facts". Below this, a message from the bot says "Sorry, we're having issues generating a response." A green dashed box highlights this error message. Below it, two yellow boxes show "Error in the response" and "Skill error", each with a "Details" and "Inspect" button. A note at the bottom says "There are no additional steps for this action. Add a new step or end the action."

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.

The screenshot shows the main menu of IBM Watsonx Assistant. The "Chat" option is highlighted with a green box. Other menu items include "Skill sets", "Skill catalog", "AI assistant builder", "Skill studio", "Access management", and "Administrator". The "AI assistant builder" section is expanded, showing "Add example phrases:", "Enter phrases your customer might use to start this action" (with "Get z/OS facts" entered), and a "Preview" window.

2. Click Add skills from the catalog.



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

The screenshot shows the "Skill catalog" page. The search bar contains "z skills". The "Apps" section displays several tiles, including "Salesloft", "Cognos", "Skill flows", "Adobe Workfront", "Webex", "IBM Process Mining", "Salesforce Chatter", "Ansible Controller Skills - z skills" (which is highlighted with a green box), "Interview top candidates using ...", "Reveal your existing applicants...", "Seismic", "FreshService", "Workday HCM", and "ZoomInfo".

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

The screenshot shows the details for the "Ansible Controller Skills - z skills" app. It lists two skills: "z/OS Gather Facts" and "z/OS Ping". Each skill card has an "Add skill +" button highlighted with a green box.

5. Click **Connect app**.

Ansible Controller Skills - z skills (2)

z/OS Gather Facts
z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS...
z/OS Ping
z skills - This playbook pings the z/OS host to test connectivity.

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**.

Connect to Ansible Controller Skills - z skills

username: admin
password: *****

Cancel Connect app

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

IBM Watsonx Orchestrate

Chat

Skills - z skills

z/OS Ping
z skills - This playbook pings the z/OS host to test connectivity.

Connected

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

Prompt:

Gather zOS facts

The screenshot shows the IBM Watsonx Assistant interface. At the top, it says "IBM Watsonx Orchestrate" and "Personal skills". On the right, there are language and accessibility settings. The main area has a message from Watson: "Hi, I'm Watson. Check out the skills in the catalog to see how I can help you." Below this, a button labeled "Gather z/OS facts" is highlighted with a green border. A callout box contains the text: "Here are the results of the z/OS Gather Facts skill." It lists one job named "z/OS Gather Facts" with a status of "pending". A progress bar indicates the job is at step 12. At the bottom of this section is a blue input field with the placeholder "Tell me what you want to do". To the left, a button says "Add skills from the catalog" with an arrow pointing right. To the right, another section titled "Ansible Controller Skills - z skills" is shown, with a note "2 skills" and a small icon.

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

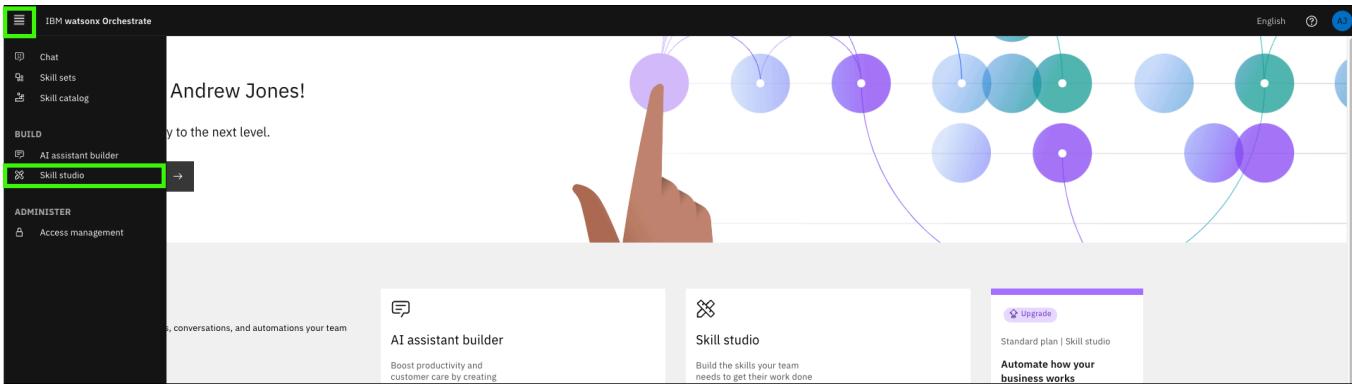
Creating skill flows

In the previous section, you ran the **Gather z/OS facts** skill, but the output was not displayed in the assistant. To both run the action and display the results, 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](#).

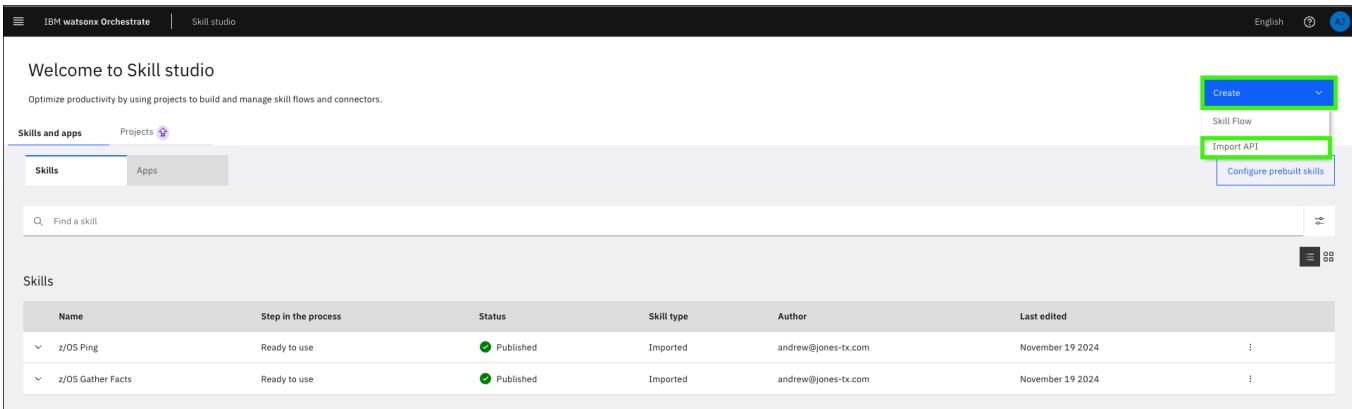
As mentioned in a previous section, default utility skills that are provided with the watsonx Assistant for Z skills collection. The **Retrieve job output** utility skill is used to return the output of a skill.

Add the utility skill

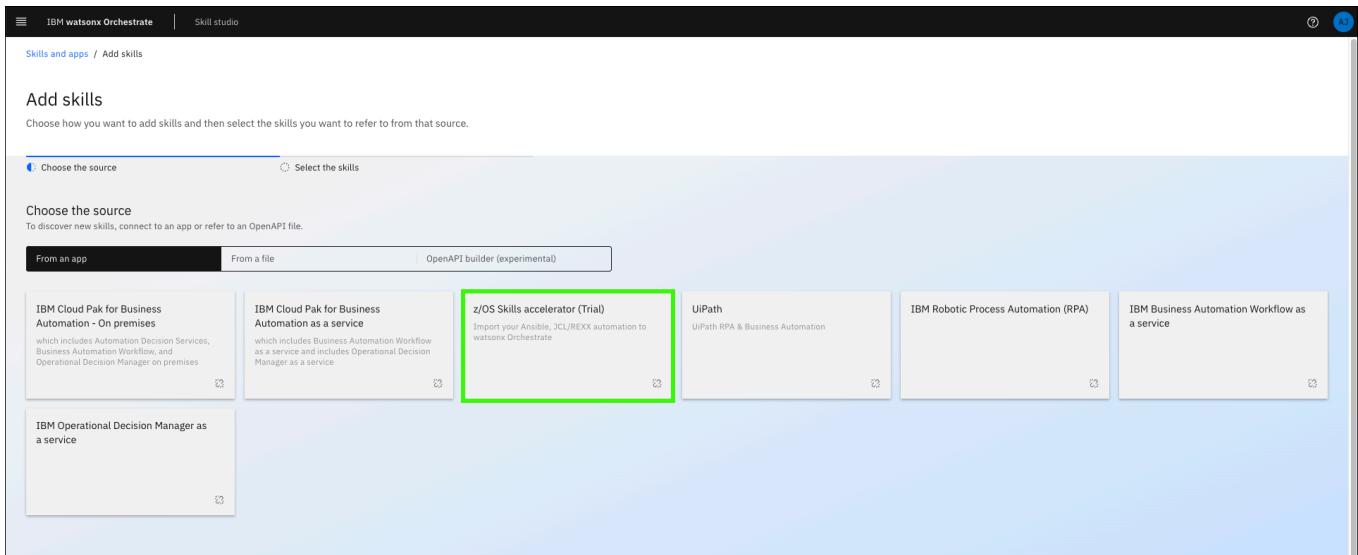
1. Open IBM watsonx Orchestrate **Skill studio**.



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`

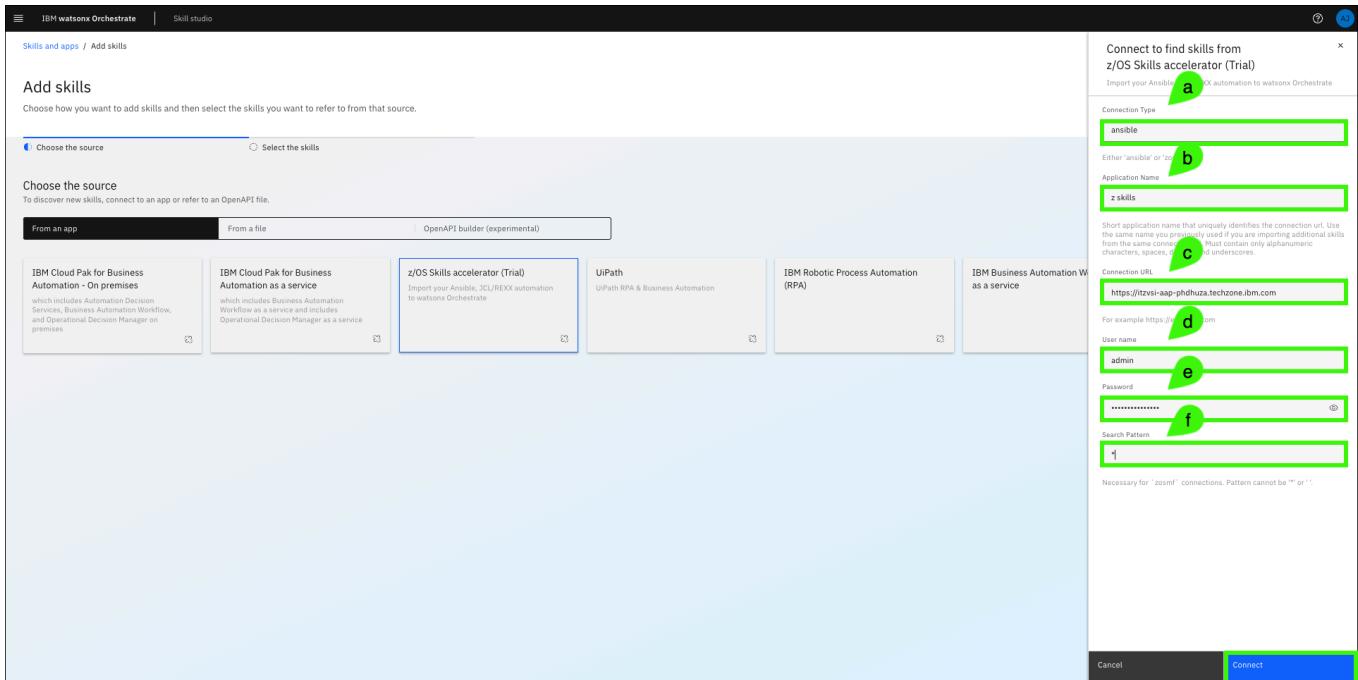
b: Application Name: <use the same application name from the previous section>

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**.

Name: Retrieve job output
Input: 0/100
Output: 0/100
Security: None
Phrases: None
Next best skills: None
API version: 1.0
Categories: None
App: Ansible Controller Skills - z skills

9. Select Skill sets from the main menu.

Step in the process	Status	Skill type	Author	Last edited
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024

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

Application	Number of skills	Credential type	Connected by	Action
Activate or deactivate attracting candidates using ThisWay Global	4	Not specified	-	
Adobe Workfront	37	Not specified	-	
Alliance Virtual Office	2	Not specified	-	
Amazon S3	8	Not specified	-	
Amazon SES	10	Not specified	-	

11. Click the **Search (🔍)** icon.

Application	Number of skills	Credential type	Connected by ⓘ	Action
Activate or deactivate attracting candidates using ThisWay Global	4	⚠️ Not specified	-	⋮
Adobe Workfront	37	⚠️ Not specified	-	⋮
Alliance Virtual Office	2	⚠️ Not specified	-	⋮
Amazon S3	8	⚠️ Not specified	-	⋮
Amazon SES	10	⚠️ Not specified	-	⋮

Items per page: 5 ⏺ 1-5 of 78 items 1 ⏺ of 16 pages ⏺

12. Search for the application name you specified earlier.

Application	Number of skills	Credential type	Connected by ⓘ	Action
Ansible Controller Skills - z skills	2	⚠️ Not specified	-	⋮

Items per page: 5 ⏺ 1-1 of 1 items 1 ⏺ of 1 page ⏺

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

Application	Number of skills	Credential type	Connected by ⓘ	Action
Ansible Controller Skills - z skills	3	⚠️ Team	andrew@jones-tx.com	⋮

Items per page: 5 ⏺ 1-1 of 1 items 1 ⏺ of 1 page ⏺

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

Zeeves draft

Skill sets

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Application	Number of skills	Credential type
Ansible Controller Skills - z skills	3	Team

Items per page: 5 | 1-1 of 1 items

Edit the Ansible Controller Skills - z skills connection **b**

Member credentials
Each team member uses their own credentials to connect to this app and use its skills.

Team credentials (Active)
The admin sets the credentials each team member uses to connect to this app and use its skills.

You have an active connection set using **Team credentials**. If you wish to update connection details click on the **Edit** **a** below.

Connected **a**

Add the skills to your Personal skills

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

Chat

Skill sets

Skill catalog **a**

AI assistant builder

Skill studio

Administrations

Access management **b**

Number of skills	Credential type	Connected by a	Action
3	Team	andrew@jones-tx.com	⋮

1 of 1 items

2. Search for the application name you specified earlier.

Skill catalog

Skills are grouped by app. Select an app to see all the skills that use that app.

Personal skills

z skills

Most popular skills

Send an email from Gmail	Create a lead in Salesforce	Send an email using Outlook
--------------------------	-----------------------------	-----------------------------

All Apps

Coupa	ZoomInfo	HubSpot CRM	Apptio Targetprocess	Salesforce	Zendesk Service
Calendly	Square	Oracle E-Business Suite	GitLab	Toggl Track	Microsoft Teams

3. Click the tile for your application.

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

Skill catalog
Skills are grouped by app. Select an app to see all the skills that use that app.

Personal skills

z skills

Apps

- ZoomInfo (26 skills)
- Workday HCM (36 skills)
- FreshService (25 skills)
- Seismic (22 skills)
- Reveal your existing applic... (7 skills)
- Interview top candidates u... (2 skills)
- Salesforce Chatter (5 skills)
- IBM Process Mining (2 skills)
- Webex (14 skills)
- Adobe Workfront (37 skills)
- Skill flows (167 skills)
- Cognos (8 skills)
- Salessoft (47 skills)
- Ansible Controller Skills - z ... (3 skills)**

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

Skill catalog / Ansible Controller Skills - z skills (3)

Personal skills

Search skills

Ansible Controller Skills - z skills

- Retrieve job output
- z/OS Gather Facts**
- z/OS Ping**

Add skill + **Add skill +** **Add skill +**

Create the skill flow

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

IBM Watsonx Orchestrate

Chat

Skill sets

Skill catalog

BUILD

AI assistant builder

Skill studio

ADMINISTER

Access management

Skills - z skills (3)

z skills

- z/OS Gather Facts**
- z/OS Ping**

Added ✓ **Added ✓** Add skill +

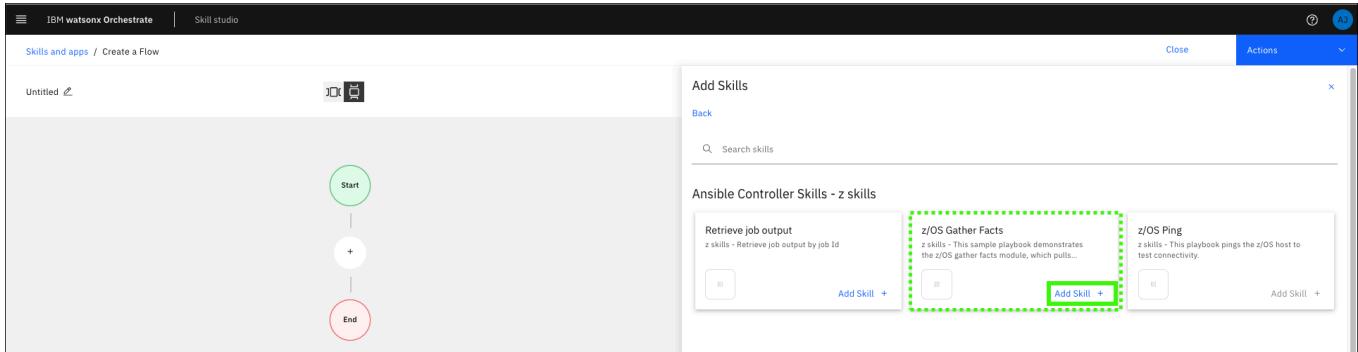
- Expand the **Create** drop-down menu and click **Skill flow**.

3. Click the + icon.

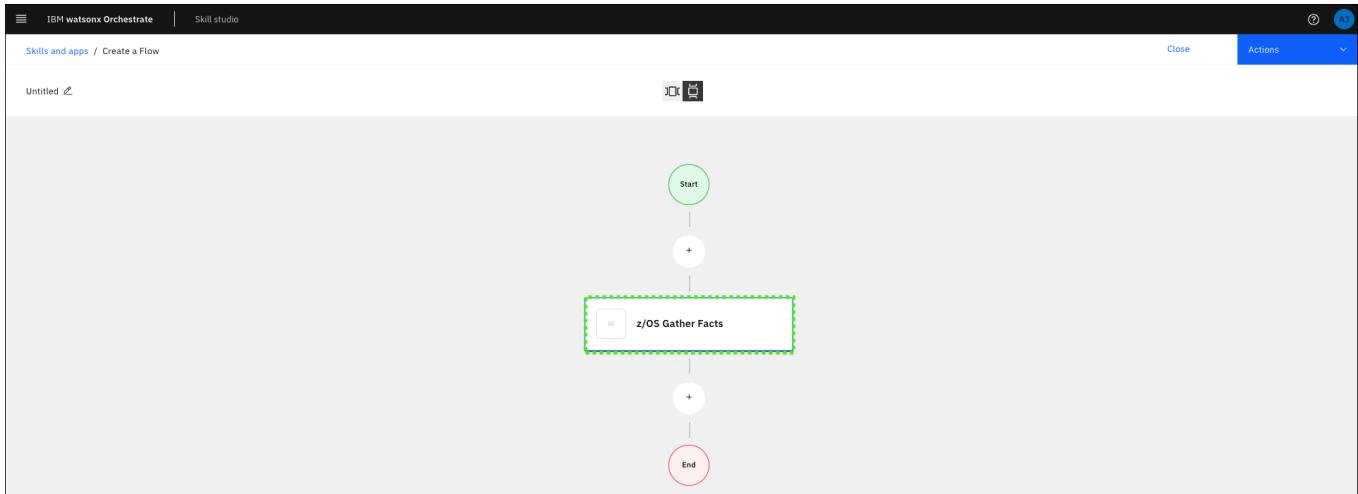
Next, you need to add the [z/OS Gather Facts](#) skill and the [Retrieve job output](#) skill to the skill flow. Use the **Search apps** function to locate the skills.

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

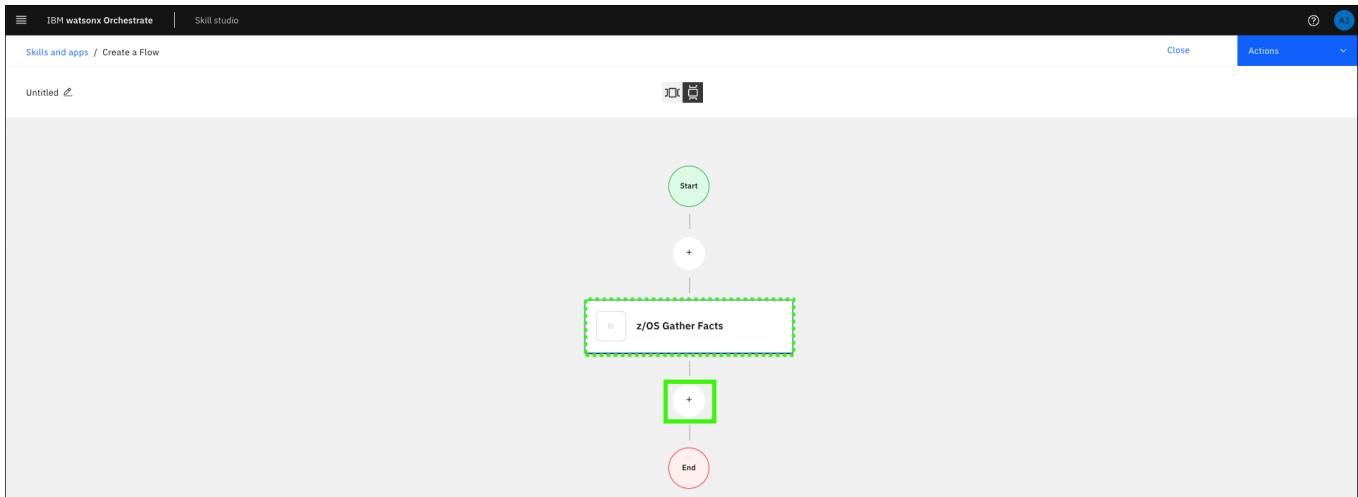
5. Click **Add Skill** in the [z/OS Gather Facts](#) tile.



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

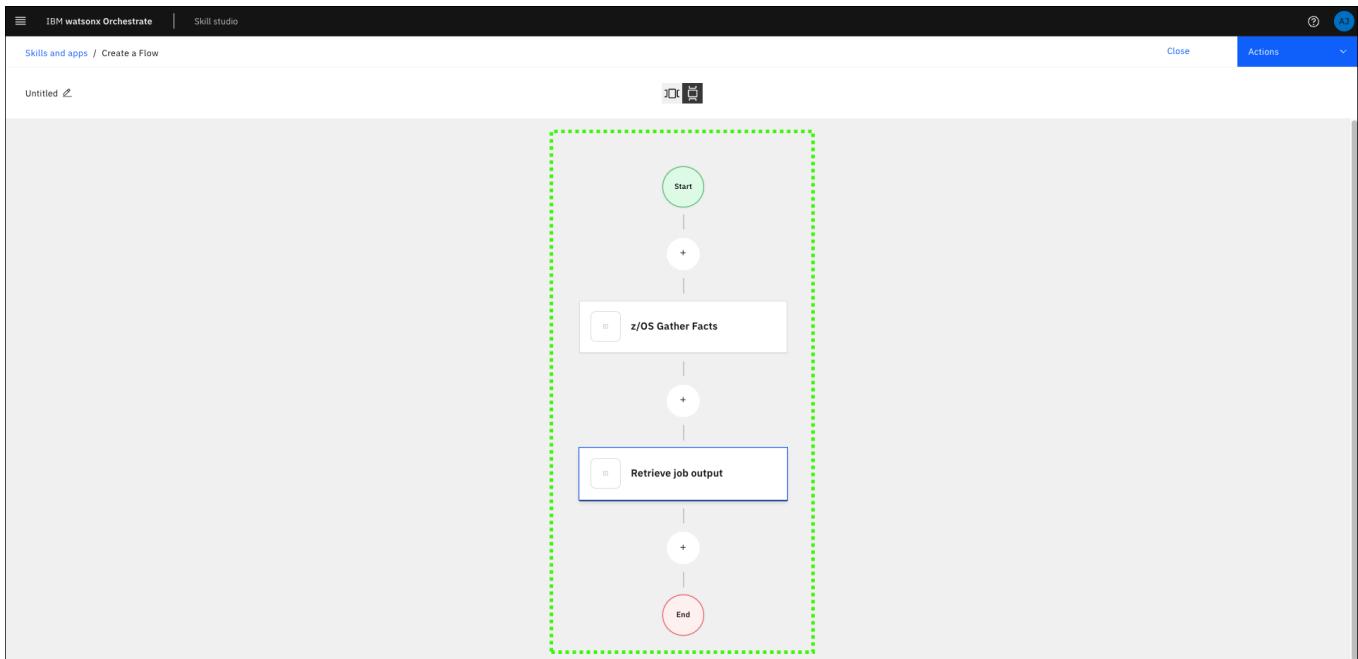


7. Click the **+** icon after the **z/OS Gather Facts** tile.



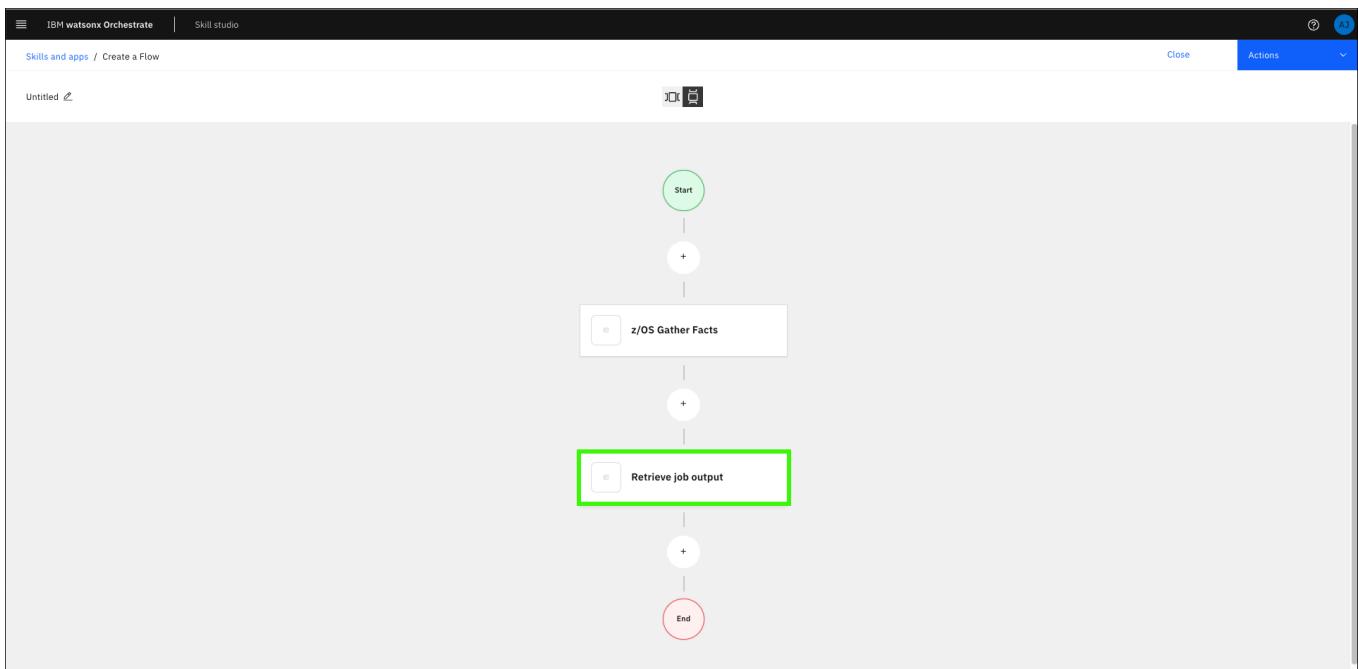
8. Repeat steps 5 and 6 for the **Retrieve job output** skill.

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

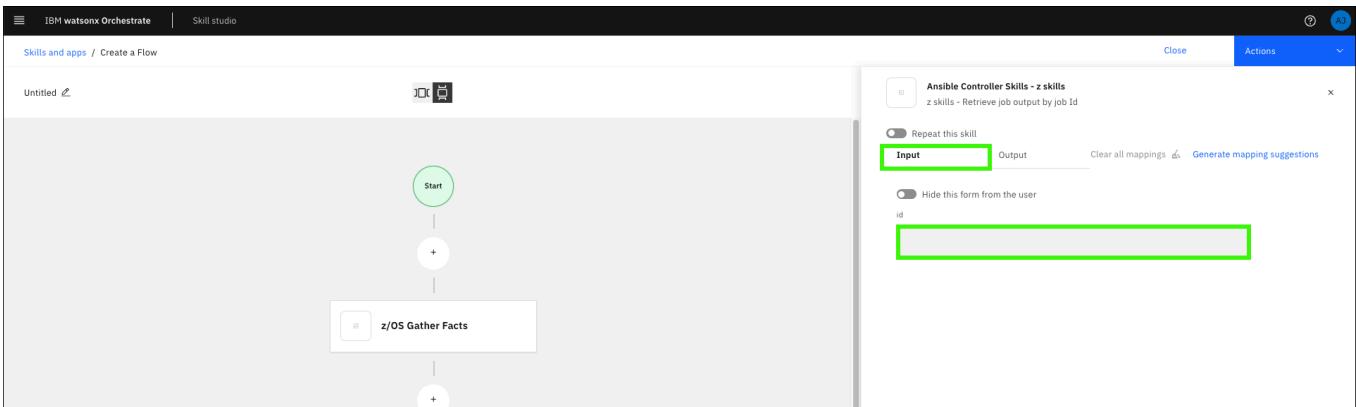


Next you must map the output values of the first skill to the input of the second skill. In this case, pass the job ID output from **z/OS Gather Facts** as an input for **Retrieve job output**.

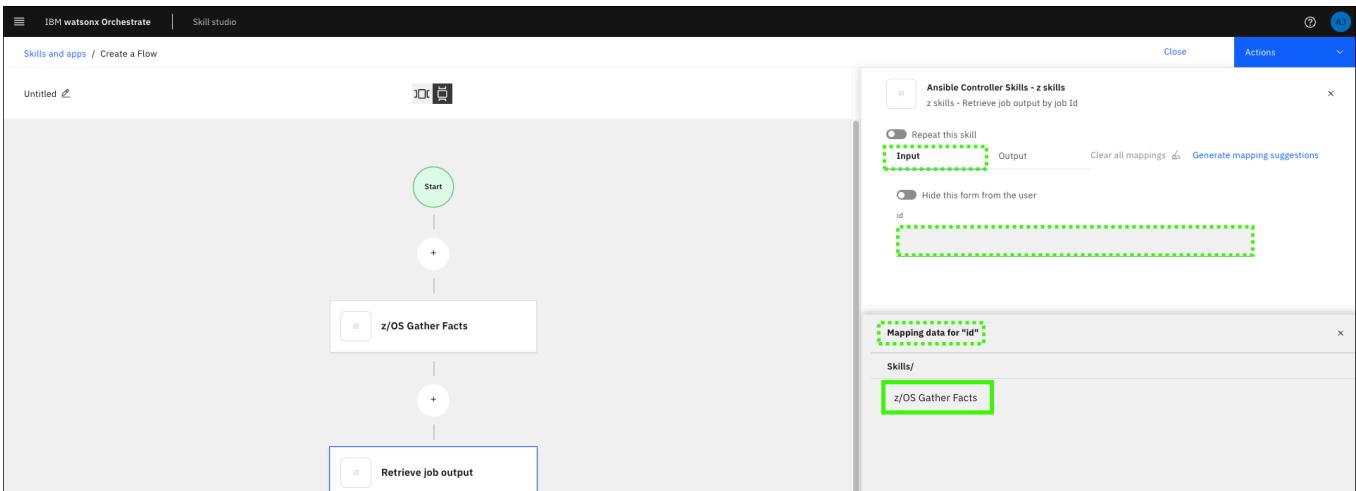
9. Click the **Retrieve job output** tile.



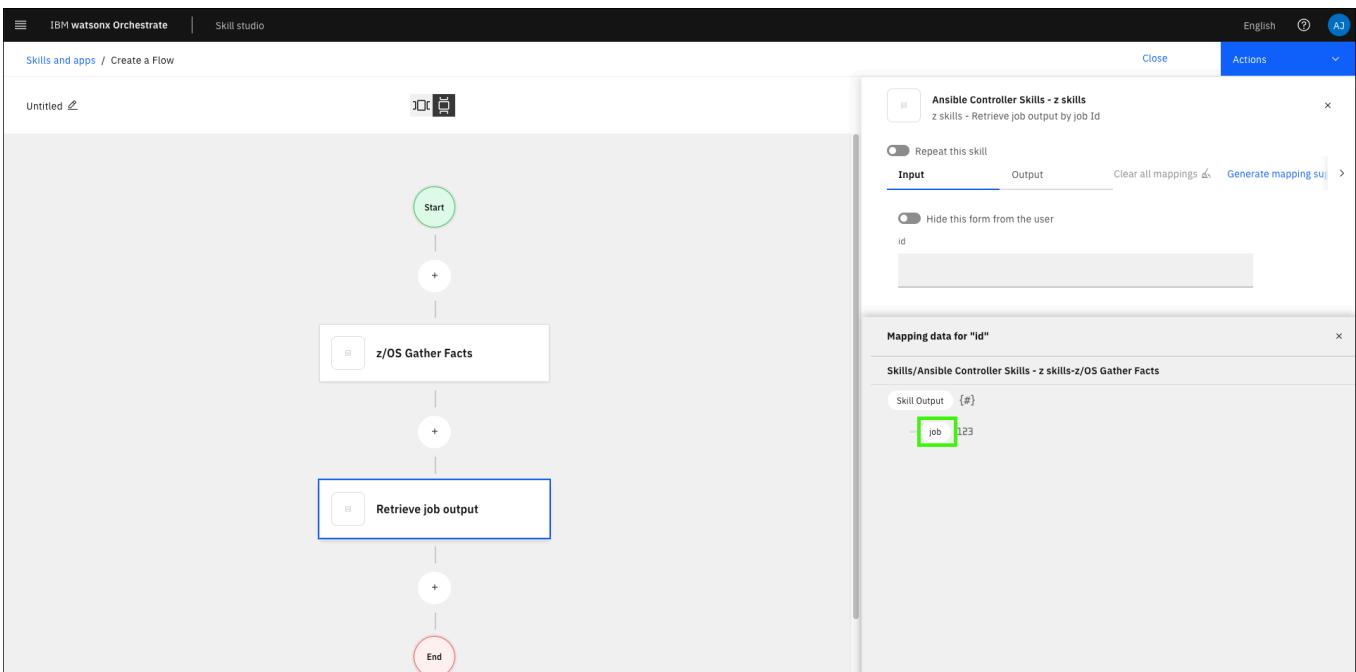
10. Select the **Input** tab and click the **id** field.



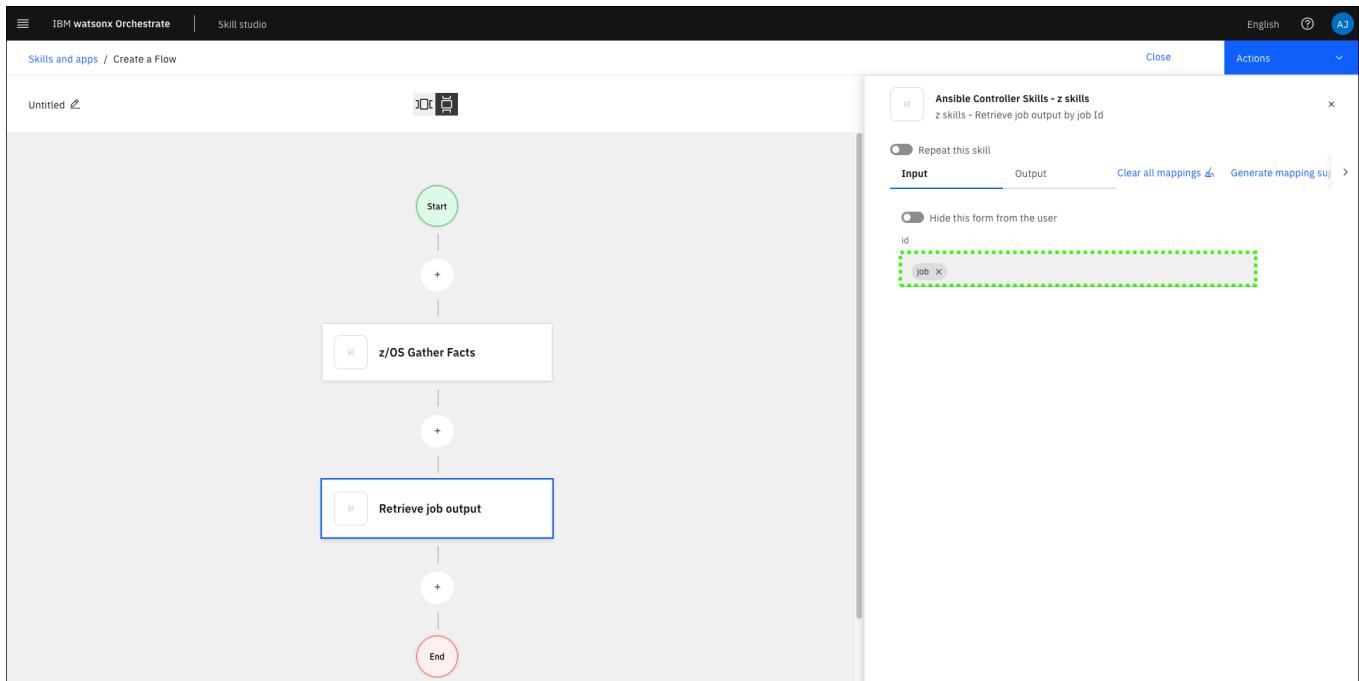
11. Click the **z/OS Gather Facts** skill in the **Mapping data for "id"** section.



12. Click the **job** icon.

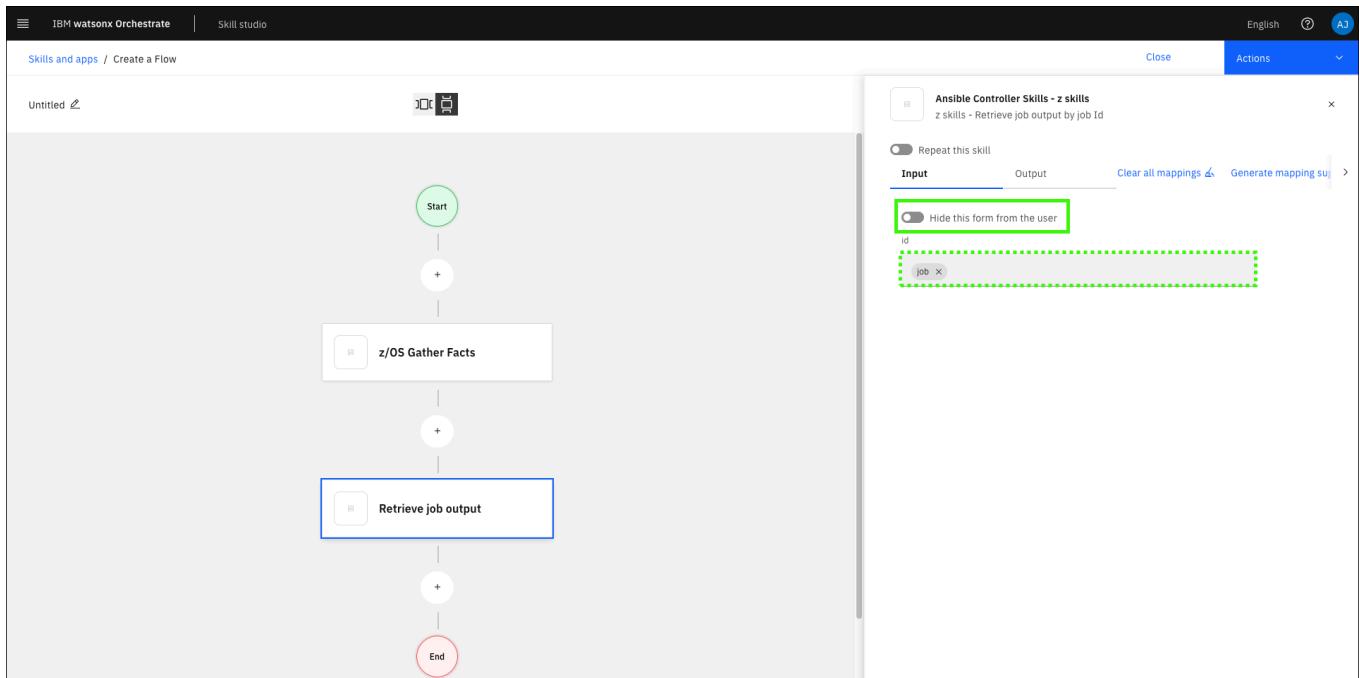


13. Verify that the **job** appears in the **id** field.

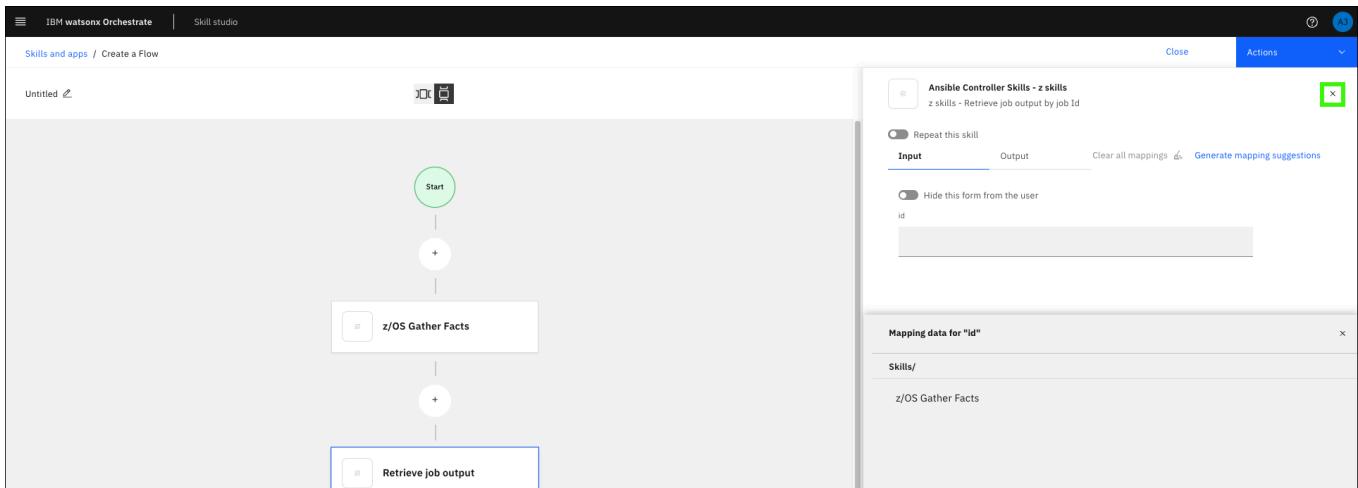


14. Optionally, toggle the **Hide this from the user** setting.

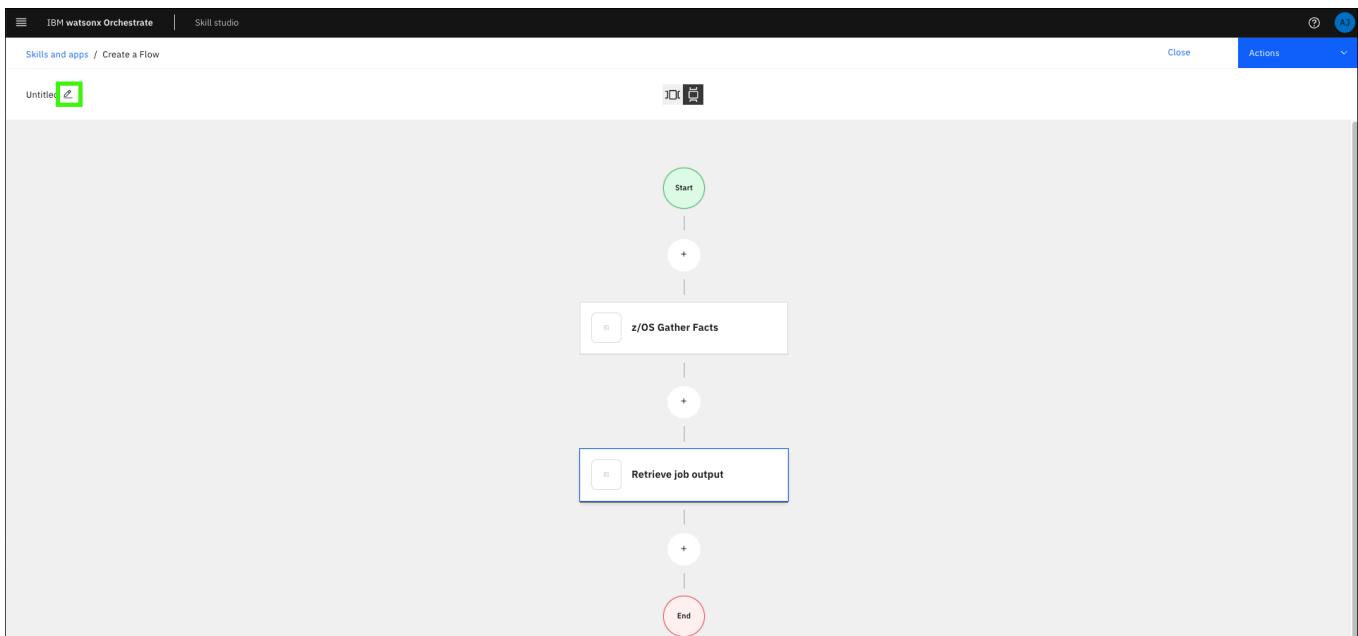
For this lab guide, this option is left disabled. Learn more about this option [here](#).



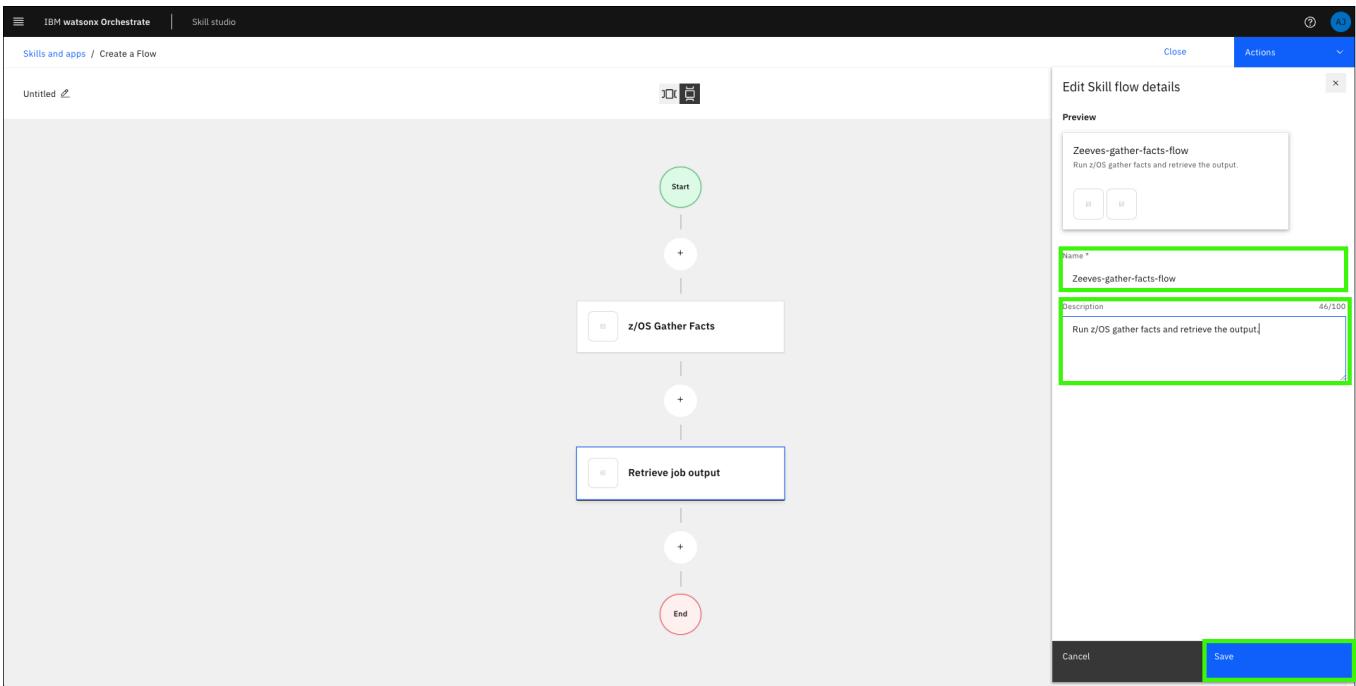
15. Click the **x** to close mapping window.



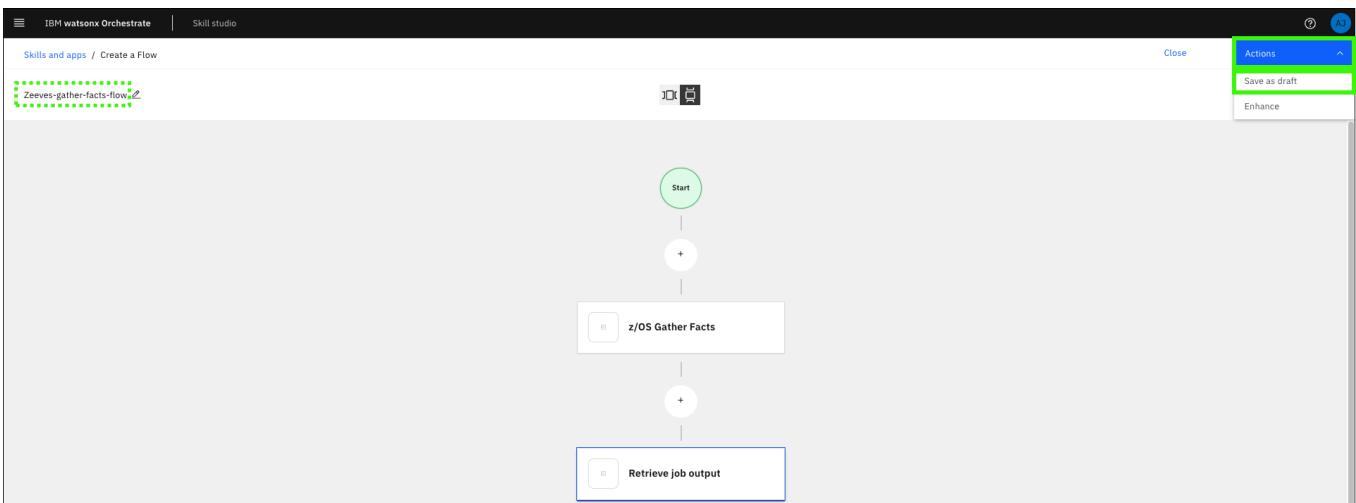
16. Click the pencil ().



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



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



19. Expand the **Actions** pull-down 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 that the default prompts are either not intuitive (the skill flow name) or a bit verbose. Replace the existing phrases with phrases that you anticipate users will use.



Be careful with the sample phrases that 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 your sample phrases. This issue has been reported to the offering team.

Example prompts:

Show me zOS facts

Gather and display zOS facts

Name **Phrases** Next best skills

Phrases are the text your user types in the chat bar to find and use a skill.

Show me zOS facts

Gather and display zOS facts

Enter new train phrase

Cancel Publish Save as draft

Enable the skill flow in your assistant

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

Step in the process	Status	Skill type	Author	Last edited
Ready to use	Published	Skill flow	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024
Ready to use	Published	Imported	andrew@jones-tx.com	November 19 2024

Published successful
Published skill Zeeves-gather-facts-flow.
15:18:32

Configure prebuilt skills

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

The screenshot shows the 'Actions' section of the AI assistant builder. On the left, there's a sidebar with 'Actions' selected. The main area displays several cards: 'Build actions' (enhance and improve your assistant's actions), 'Customize your greeting' (welcome users in a unique way), 'Create a fallback plan' (train your assistant to adapt to specific situations), 'Preview & debug' (enable debug mode when you preview your assistant), 'Customize web chat' (design how your assistant shows up in your live environment), 'Set up a channel' (select channels for deployment), 'Set up live agent' (integrate live agents to transfer users to when you edit web chat), and 'Public assist' (choose environment for assistance). A 'New' button is located in the top right corner of the 'Actions' card.

3. Click New action.

The screenshot shows the 'Actions' list page. The left sidebar has 'Actions' selected under 'Created by you'. The main area is a table listing actions. A 'New action +' button is highlighted in the top right corner of the table header.

4. Click the Skill-based action tile.

The screenshot shows the 'Create an action' wizard. It asks 'What kind of action do you want to build?'. Three options are shown: 'AI-guided action' (configure AI to generate responses based on specific use cases, marked as 'Beta'), 'Skill-based action' (build an action that helps perform a task, highlighted with a green border), and 'Custom-built action' (design your own action step-by-step).

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

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

Build an action from a skill

Select a skill
Choose a conversational skill published as a foundation of your action.

Search a skill

Zeeves-gather-facts-flow	Retrieve job output	z/OS Ping	z/Gather Facts	Summarize the Webex meeting transcript
Run z/OS gather facts and retrieve the output. Last updated: 2024-11-19T21:18:31.793Z	z skills - Retrieve job output by job Id Last updated: 2024-11-19T20:08:59.538Z	z skills - This playbook pings the z/OS host to test connectivity. Last updated: 2024-11-19T15:58:20.567Z	z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS-specific information from the z/OS host. Last updated: 2024-11-19T15:56:26.843Z	in watsonx.ai Last updated: 2024-11-04T10:49:16.502Z
Summarize the Box content	Summarize a Zendesk ticket	Summarize a ServiceNow incident	Summarize a Salesforce opportunity	Sharepoint document summary
in watsonx.ai Last updated: 2024-11-04T10:49:12.077Z	in watsonx.ai Last updated: 2024-11-04T10:49:09.476Z	in watsonx.ai Last updated: 2024-11-04T10:49:05.828Z	in watsonx.ai Last updated: 2024-11-04T10:49:01.769Z	in watsonx.ai Last updated: 2024-11-04T10:48:55.707Z
Salesloft email summary	Salesforce case summarization	Salesforce case sentiment analyze	Outlook email summary	Github issue summarization

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

Customer starts with:
Example: I want to pay my credit card bill.

Add example phrases:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action Total: 0

Example: I want to pay my credit card bill.

New action

What does your customer say to start this interaction?
Show me zOS facts

Cancel Save

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

Customer starts with:
Show me zOS facts

Add example phrases:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action Total: 1

Enter a phrase

Show me zOS facts

Save

8. Click close (x).

Customer starts with:
Show me zOS facts

Configuration

Skill Name: Zeeves-gather-facts-flow
Skill ID: composite_skill_1.0.0_16b2f4acfe1048d1ba0d92fd566...

Add example phrases:

Enter phrases that a customer types or says to start the conversation about a specific topic. These phrases determine the task, problem, or question your customer has.

The more phrases you enter, the better your assistant can recognize what the customer wants.

Enter phrases your customer might use to start this action Total: 1

Enter a phrase

Show me zOS facts

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

All items

Created by you

z/OS Gather Facts

Last edited: an hour ago Examples count: 2 Steps count: 0 Status: Green

b

c

10. Wait for system training to complete.

Note: The message changes to "System is trained" and then disappears.

All items

Created by you

Name

Last edited: 2 minutes ago Examples count: 1 Steps count: 0 Status: Green

System is training...

11. Click **Preview**.

The screenshot shows the 'Actions' section of the IBM Watsonx Orchestrator. On the left is a sidebar with categories like 'All items', 'Created by you', 'Variables', and 'Saved responses'. In the main area, a table lists one item: 'Zeeves-gather-facts-flow'. The table columns include 'Name', 'Last edited', 'Examples count', 'Steps count', and 'Status'. The 'Status' column for this item shows a green circle with a checkmark. At the bottom right of the main area, there is a 'Preview' button.

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

The screenshot shows the 'Preview' window of the AI assistant builder. It displays a simulated conversation between a customer and an AI assistant. The customer asks 'Show me zOS facts'. The AI assistant responds with 'Show me zOS facts'. The 'Preview' window is overlaid on the main interface, which shows the 'Actions' list and the 'Zeeves-gather-facts-flow' item.

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.

The screenshot shows the IBM Watsonx Orchestrate interface. In the top navigation bar, it says "IBM Watsonx Orchestrate" and "AI assistant builder". Below this, the workspace name "Zeeves" is shown. On the left, there's a sidebar with sections like "Actions", "All items", "Created by you", "Variables", and "Saved responses". The main area displays a table with one item selected. The table columns are "Name", "Last edited", "Examples count", "Steps count", and "Status". The selected item is "Zeeves-gather-facts-flow", last edited 11 minutes ago, with 1 example and 0 steps, and a green status. To the right of the table is a "Preview" pane showing a simulated conversation between a user and a bot. The user asks "Show me zOS facts" and the bot responds with "Zeeves-gather-facts-flow".

14. Review the results from the skill flow.

Use both scroll bars in the assistant preview to review all the returned information. The output is 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.

This screenshot is similar to the previous one but shows a different conversation. The user asks "Show me zOS facts" and the skill responds with a large block of JSON data. The JSON output is highlighted with a green dashed box. The JSON content includes various parameters and their values, such as "ipl_volume": "D25VS", "load_param_device_id": "5", and "operator_prompt_flag": "true".



Sample output from 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.] *****
TASK [Print gathered facts about the master catalog.] *****
[0;32mok: [zos_host][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 previous scenario might or might not be relevant for your client's use case. The scenario illustrates how to sequence skills together in a skill flow to create an action that your assistant triggers based on prompts that use the pre-configured Ansible automation templates. You are encouraged to create your own skill flows and prompts that use 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 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

You can also create **custom-built** actions. Custom-built actions have 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 subaction, and other actions. Custom-built actions are a powerful way of customizing the 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. These skills are 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.

As an example, the following are console commands that are 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 to your sandbox environment by downloading the compressed file [here](#) and following [these instructions](#).

Extract the embedded 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 built the assistant, configured conversational search, and added skills and automations. You tested your assistant by 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 users. Each assistant that you create comes with two **environments**: *draft* and *live*. You configured your assistant in the draft environment. Each environment has its own set of IDs, URLs, and service credentials referenced by external services.

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

This screenshot shows the 'Environments' page in the AI assistant builder. The 'Draft' tab is selected, indicated by a green dashed border. The 'Live' tab is also present. On the left, there's a sidebar with icons for environments, channels, and extensions. The main area has sections for 'Draft environment', 'Channels' (listing 'Web chat'), and 'Resolution Methods'. Under 'Resolution Methods', there's a 'Draft content' section with a 'Last edited' timestamp and a 'Edit content' link. A callout box points to the 'Last edited' timestamp and the 'Draft' link. Below this is an 'Extensions' section with a 'Search' button.

This screenshot shows the same 'Environments' page, but the 'Live' tab is selected, indicated by a green dashed border. The 'Draft' tab is now unselected. The main area has sections for 'Live environment', 'Channels' (listing 'Web chat'), and 'Resolution Methods'. Under 'Resolution Methods', there's a 'Content' section with a radio button labeled 'No published version' and a 'Publish version' link. A callout box points to the 'No published version' radio button. Below this is an 'Extensions' section with a 'Search' button and a note about extending what the assistant can answer by searching existing documents. There's also an 'Add +' button at the bottom right of the extensions section.

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 (channels) that you use 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 to give to users.

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

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

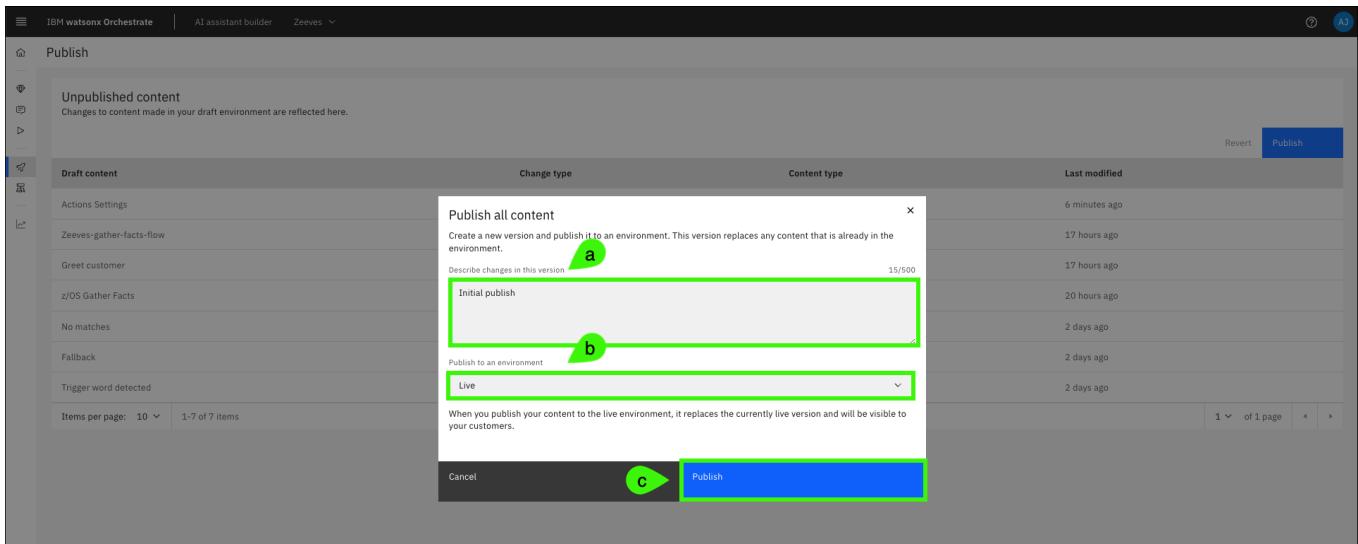
The screenshot shows the IBM Watsonx Orchestrate interface with the "AI assistant builder" tab selected. On the left, there's a sidebar with "Home" highlighted. The main area has a "Publish" button at the top, which is also highlighted with a green border. Below it, there are several cards for customizing the assistant, such as "Build actions", "Customize your greeting", "Create a fallback plan", "Preview & debug", "Customize web chat", "Set up a channel", "Set up live agent", and "Publish assist". At the bottom, there's a diagram showing a "Default behavior" node connected to "General purpose" and "Conversational search" nodes, which then lead to a "Search" node. A watermark in the background says "Watson AI Assistant" and "Watson AI Assistant".

2. Click **Publish**.

The screenshot shows the "Publish" content page. It displays a table of "Draft content" changes. The table has columns for "Change type", "Content type", and "Last modified". Changes listed include "Zeeves-gather-facts-flow", "Greet customer", "Actions Settings", "Fallback", "No matches", and "Trigger word detected", all updated within the last few hours. At the bottom right, the "Publish" button is highlighted with a blue border.

Draft content	Change type	Content type	Last modified
Zeeves-gather-facts-flow	Updated	Actions	30 minutes ago
Greet customer	Updated	Actions	31 minutes ago
Actions Settings	Updated	Settings	2 hours ago
Fallback	Updated	Actions	2 hours ago
No matches	Updated	Actions	2 hours ago
Trigger word detected	Updated	Actions	2 hours ago

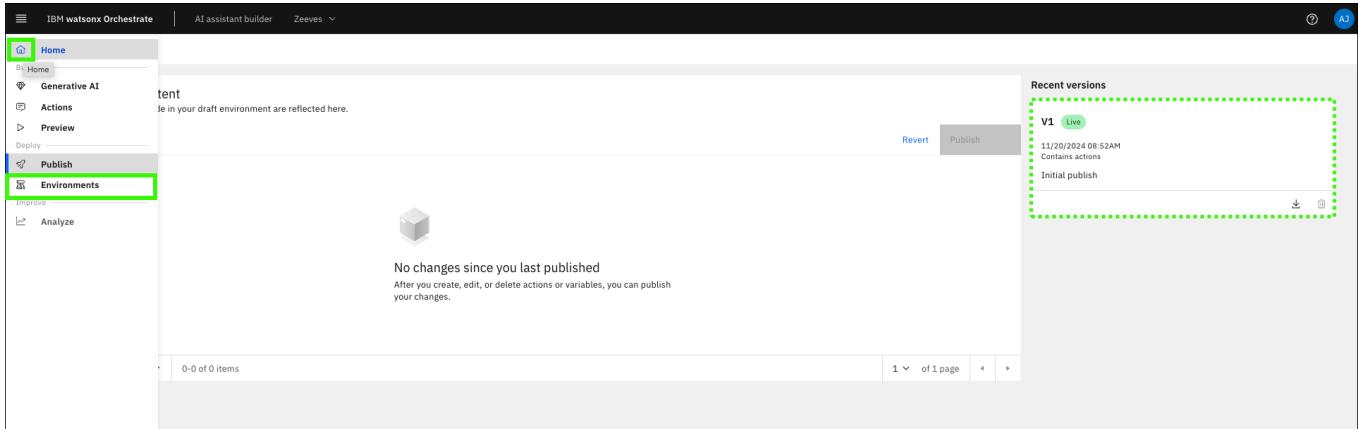
3. Enter a description of the changes (a), set the environment to **Live** (b), and then click **Publish** (c).



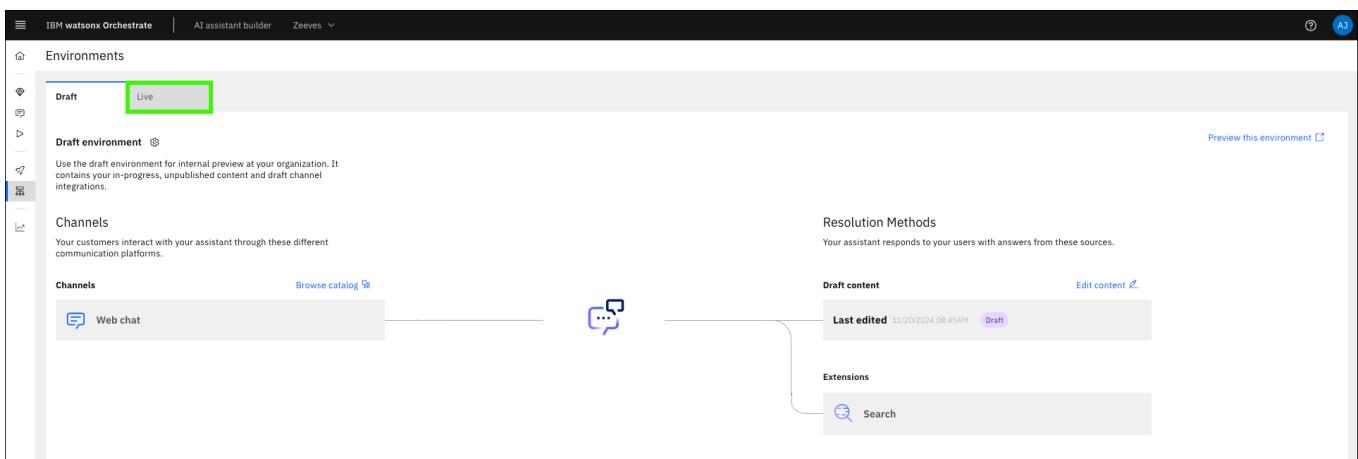
Configure the live environment

Important: When the live environment is created, the 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.

The screenshot shows the IBM Watsonx Orchestrate interface with the 'Live' tab selected. In the 'Channels' section, the 'Web chat' option is highlighted with a green box. Other channels listed are 'Email', 'SMS', 'Facebook', 'Twitter', and 'LinkedIn'. To the right, there are sections for 'Live environment', 'Resolution Methods', 'Content' (version V1, 11/20/2024 08:52AM, Live), and 'Extensions' (Search). A central icon of two speech bubbles is positioned between the 'Channels' and 'Content' sections.

4. Customize the live assistant as you see fit.

On the **Style** tab, you're able to set the Assistant name that is displayed on the chat window when users are interacting with the assistant. For pilots or demos, consider personalizing this name for the client. Also in the **Style** tab, you can set the themes and display settings of the chat windows, including the ability to enable the IBM Watermark and enable streaming.

On the **Home** tab, you enable and customize the greeting message from the assistant when the user accesses the assistant chat. You are also able to set **Conversation starters** that are displayed in the chat window. When selected by the user, the text of the conversation starter is sent as a prompt, so it is important that your assistant is trained and tested to answer correctly. It is highly recommended to remove the default conversation starters and create your own. The ability to add a Background style for the assistant chat window is on the home screen tab.

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.

The screenshot shows the 'Style' tab selected in the navigation bar. The 'Suggestions' button is highlighted with a green box. The preview window on the right shows a dark-themed web chat interface with a blue owl avatar and sample messages.

5. Click Save and exit.

The screenshot shows the 'Save and exit' button highlighted with a green box. The interface and preview window remain the same as in the previous step.

6. Click Add in the Search tile.

The screenshot shows the 'Live' environment configuration in IBM Watsonx Orchestrate. It includes sections for 'Live environment', 'Channels' (Web chat), 'Resolution Methods' (Content, V1, Live), and 'Extensions' (Search). A green box highlights the 'Add +' button under Extensions.

7. Click Custom service.

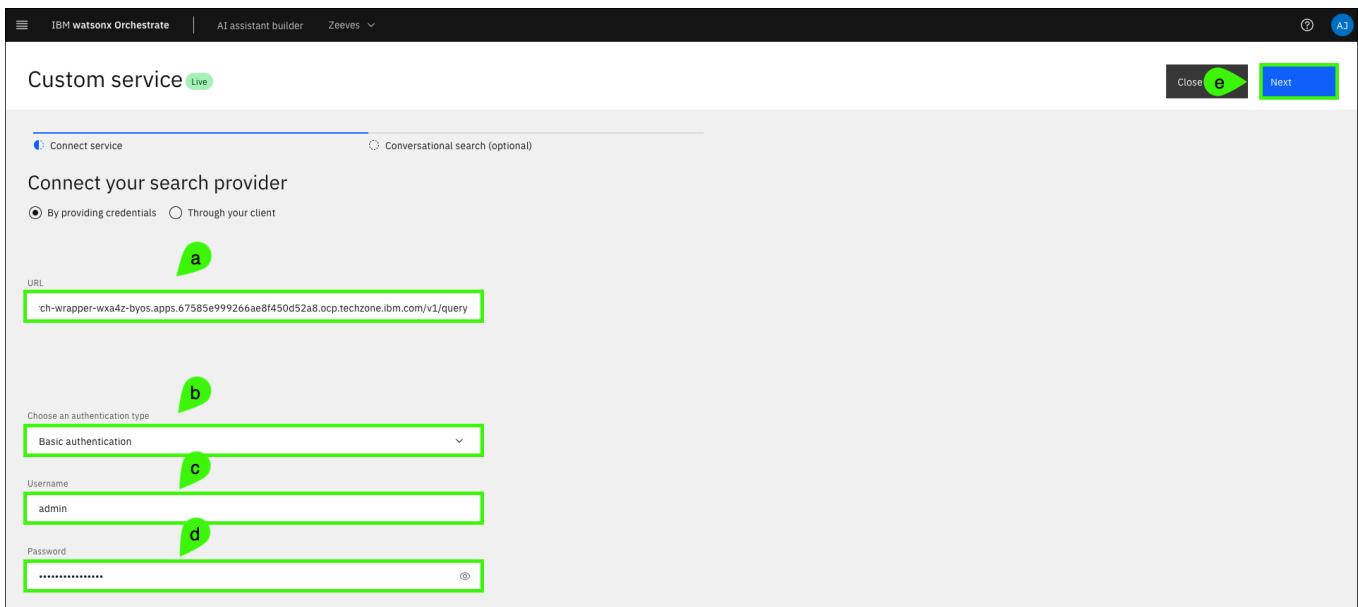
The screenshot shows the 'Set up a new search integration' dialog box. It lists 'Es' (Elasticsearch) and 'Custom service'. The 'Custom service' option is highlighted with a green box.

- 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 that you specified in the wrapper-creds.yaml file for the Password (d), and then click Next (e).

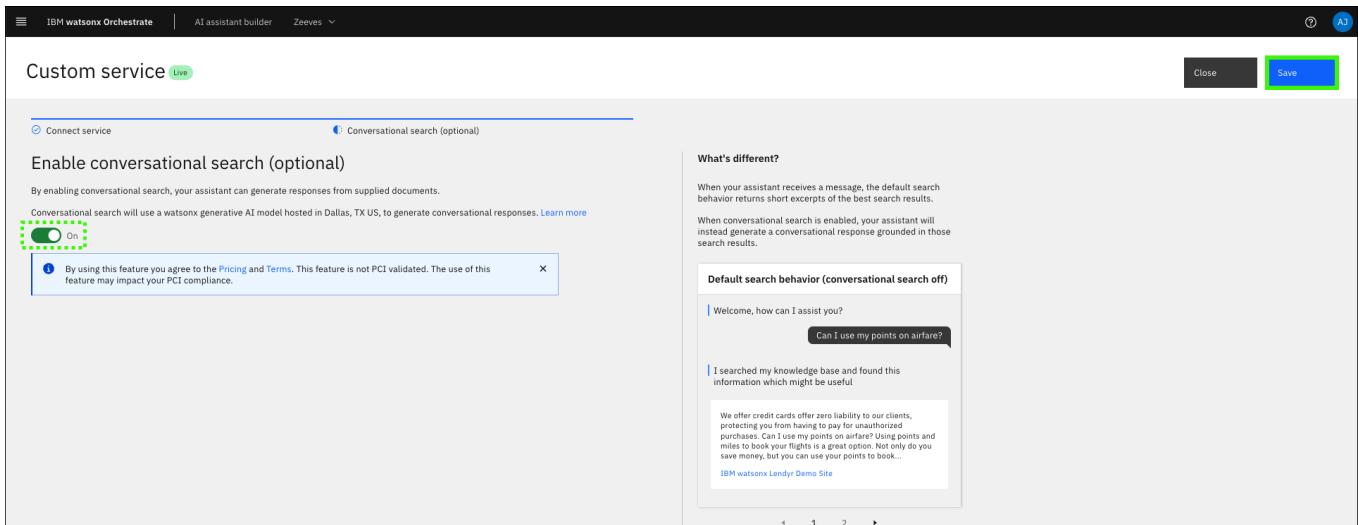


Use the correct URL and authentication type!

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).

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_*"
  }
}
```

The screenshot shows the 'Custom service' configuration page in the IBM Watsonx Orchestrate interface. The top navigation bar includes 'IBM Watsonx Orchestrate', 'AI assistant builder', and 'Zeeves'. The right side features a toolbar with 'Close' (button h) and 'Save' (button g).

Conversational search

- Conversation length:** A slider with 'On' (button a) and 'Off' (button b) options.
- Context-dependent:**
 - Single turn** (radio button c, highlighted with a green circle)
 - Entire conversation** (radio button d, highlighted with a green circle)

Text: "The assistant uses only the current user input for retrieving search results and generating answers. This works well for clear, complete inputs but generally won't work with context-dependent queries such as, 'Why is that?' after a previous answer."

Search configuration

- Retrieval confidence threshold:** A slider with 'Lowest' (radio button e, highlighted with a green circle) selected.
- Generated response length:** A slider with 'Concise' (radio button f, highlighted with a green circle) selected.
- Response confidence threshold:** A slider with 'Lowest' (radio button g, highlighted with a green circle) selected.

Citations

Citations will be displayed to the end user. This doesn't impact the number of citations used in the response.

Default filter

Can be anything (plain text, JSON, etc.)

Metadata

```
{"doc_weight": 1.0,
"product_docs": 0.2,
"customer_docs": 0.8,
"ibm_indices": "**.ibm_docs.state",
"customer_indices": ".customer_**"}
```

Search display text

Define the text your search will display to the end user

No results found	Connectivity issue
I searched my knowledge base, but did not find anything related to your query	

Connect the skills to the live environment

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

The screenshot shows the IBM Watsonx Orchestrate interface. The left sidebar has sections: Chat, Skill sets (highlighted with a green border), Skill catalog, BUILD (AI assistant builder, Skill studio), and ADMINISTER (Access management). The main content area shows 'Resolution Methods' with 'Content' (V1, 12/04/2024 03:22PM, Live) and 'Extensions' (Search). A central icon is a blue speech bubble with three dots.

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

The screenshot shows the 'Team Skills' page. The 'Skill sets' section lists 'Team Skills' (highlighted with a green border), 'Orchestrator Agent Skillset', 'Team Skills', 'Zeeves draft', and 'Zeeves live'. Below this, there's a 'Name' section with a placeholder 'No skills' and a 'Manage skills' button. A note says 'Click Manage skills to give this team digest specific abilities.'

3. Click **Connections**.

The screenshot shows the 'Zeeves live' skill set page. The 'Skills' tab is selected (highlighted with a green border). Below it, there's a 'Skills (3)' section with a note about skills being available to team members. The 'Name' section lists 'z/OS Gather Facts', 'Zeeves-gather-facts-flow', and 'Retrieve job output'. At the bottom, there are pagination controls: 'Items per page: 10' and '1-3 of 3 items'.

4. Search for the application name you specified earlier.

Skill sets

Zeeves live

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Application	Number of skills	Credential type	Connected by	Action
Ansible Controller Skills - z skills	3	Not specified	-	⋮

Items per page: 5 1-1 of 1 items 1 of 1 page ⌂ ⌃ ⌄ ⌅

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

Skill sets

Zeeves live

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Application	Number of skills	Credential type	Connected by	Action
Ansible Controller Skills - z skills	3	Not specified	-	⋮ i Connect app 1 page ⌂ ⌃ ⌄ ⌅

Items per page: 5 1-1 of 1 items

6. Click **Connect app**.

Skill sets

Zeeves live

Skills Connections

These are the applications that are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to enable skills to either use personal or team credentials.

Application	Number of skills	Credential type
Ansible Controller Skills - z skills	3	Not specified

Items per page: 5 1-1 of 1 items

Connect to Ansible Controller Skills - z skills

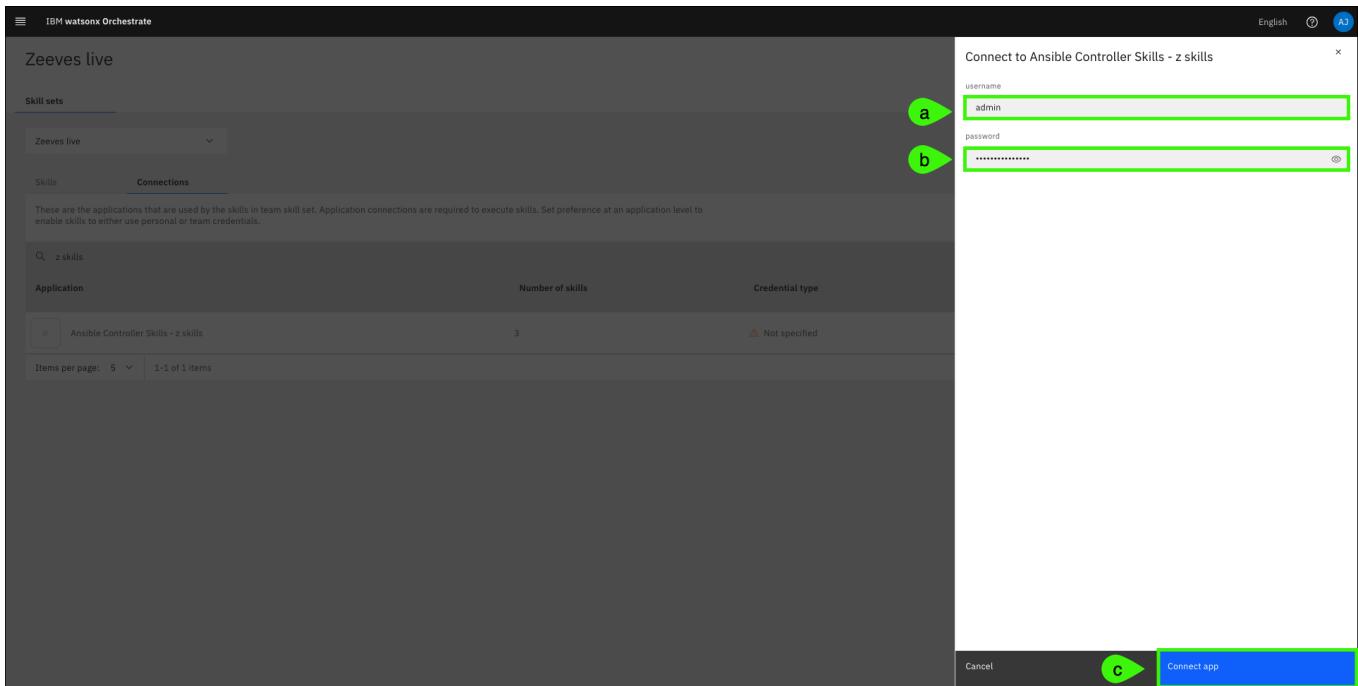
Member credentials
Each team member uses their own credentials to connect to this app and use its skills.

Team credentials
The admin sets the credentials each team member uses to connect to this app and use its skills.

You selected **Team credentials** for the credential type. Click **Connect app** to provide the credentials your team will use and to connect to the app.

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

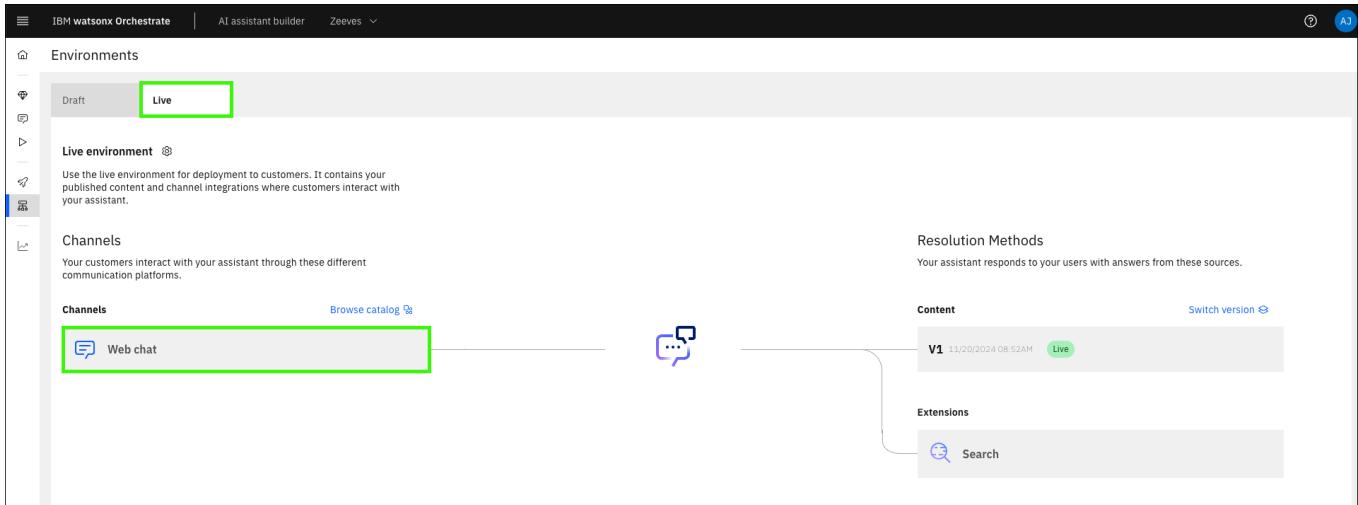
After configuring your assistant's settings and publishing, the final step is to deploy your assistant, which can be done across various channels depending on the use case.

Several options exist for deploying your assistant through channels and integrations to satisfy the use cases that you might encounter. Learn more about all the deployment options [here](#).

For this lab, deploy the assistant by using the web chat integration. The web chat integration provides an assistant interface that can integrate with a website. Learn more about the web chat integration [here](#).

1. Open the **Environments** page in the **AI assistant builder**.

2. Click **Web chat** for the **Live** environment.



3. Click the **Embed** tab.

The screenshot shows the 'Web chat' configuration page in the IBM Watsonx Orchestrate interface. The 'Embed' tab is selected and highlighted with a green border. On the right side, there is a preview window titled 'Zeeves 1.0' showing a dark-themed chat interface with a blue owl icon and sample messages. The configuration options include:

- Style:** Set to 'Light'.
- Assistant's name as known by customers:** 'Zeeves 1.0'.
- Intended purpose:** Standard: For virtual agents and customer support experiences.
- Choose a theme:** Set to 'Light'.
- Primary color:** #FFFFFF.
- Secondary color:** #3D3D3D.
- Chat header:** Set to 'Accent color' (#0354E9).
- Accent color:** #0354E9.
- Significant and interactive objects:** Set to 'Accent color' (#0354E9).
- Size:**
 - Width:** 380px.
 - Height:** 640px.
- IBM Watermark:** Enabled (On).

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

The screenshot shows the 'Web chat' configuration page in the IBM Watsonx Orchestrate interface. The 'Embed' tab is selected and highlighted with a green border. A code snippet for embedding the chat is displayed in a text area:

```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: '8b4ad299-e6b9-434c-8c8c-2754d9272fe5', // The ID of this integration.
    region: 'wco-us-south', // The region your integration is hosted in.
    serviceInstanceId: '7c1e3381-8dcf-476b-bdf2-319f531087ba', // The ID of your service instance.
    onLoad: async (instance) => { await instance.render(); }
  };
  setTimeout(function() {
    const t=document.createElement('script');
    t.src='https://web-chat.global.assistant.watson.appdomain.cloud/versions/' + (window.watsonAssistantChatOptions.version || 'latest') + '/index.js';
    document.head.appendChild(t);
  });
</script>
```

5. In a text editor, create a file that is 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:

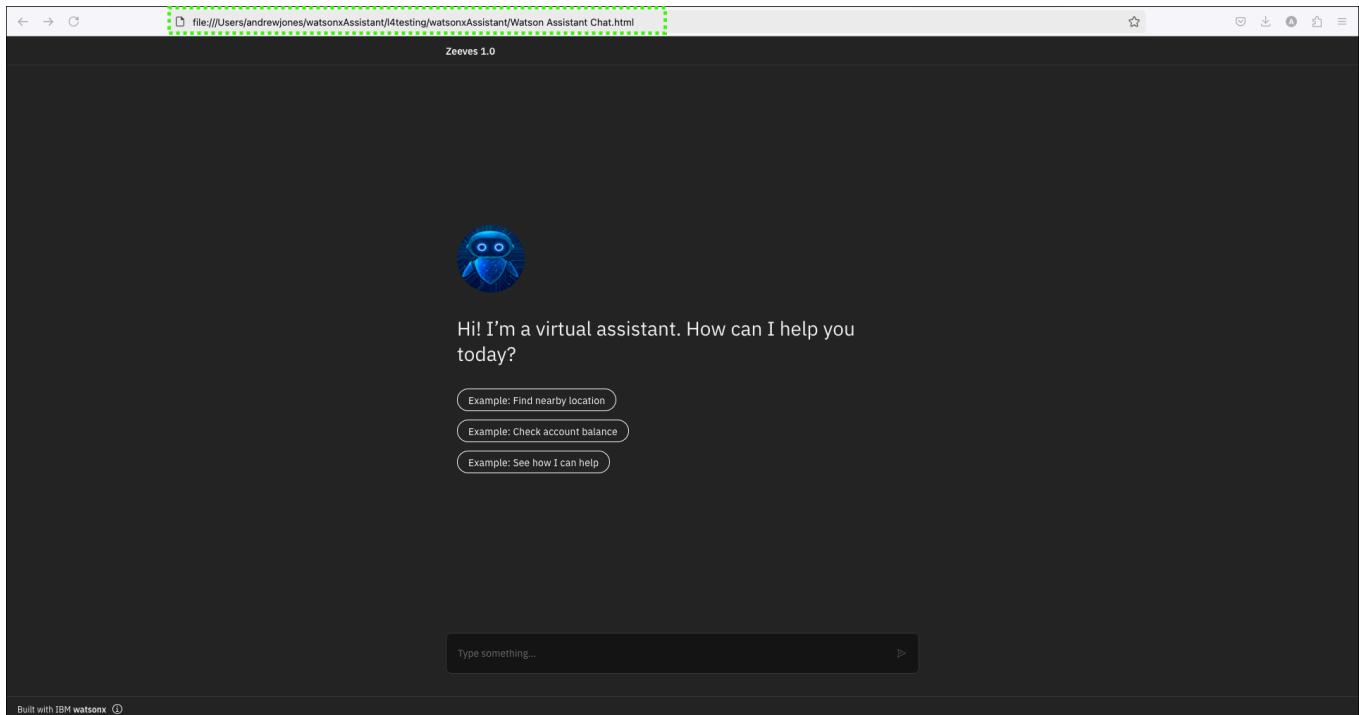
```
Watson Assistant Chat.html ×
Users > andrewjones > Downloads > Watson Assistant Chat.html > html
1  <html lang="en">
2  <body>
3  <script>
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
```

```
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,
```

After modification:

```
Users > andrewjones > Downloads > Watson Assistant Chat.html > html > body > script > layout
1  <html lang="en">
2  <head>
3    <style>
4      .WebChatContainer {
5        top: 0;
6        bottom: 0;
7      }
8    </style>
9  </head>
10 <body>
11 <div class="WebChatContainer">
12
13 <script>
14   const element = document.querySelector('.WebChatContainer');
15
16   window.watsonAssistantChatOptions = {
17     integrationID: "8b4ad299-e6b9-434c-8c8c-2754d9272fe5", // The ID of this integration.
18     region: "wxi-us-south", // The region your integration is hosted in.
19     serviceInstanceId: "7c163381-8dcf-476b-bdf2-319f531087be", // The ID of your service instance.
20     element,
21
22     openChatByDefault: true,
23     hideCloseButton: true,
24
25     layout: {
26       showFrame: false,
27       hasContentMaxWidth: true,
28     },
29
30     onLoad: async (instance) => {
31       window.WACInstance = instance;
32       await instance.render();
33     }
34   }
35
36
37
38
39
40
```

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 that 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 to find the current channels that are supported and where to get more information.

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

The screenshot shows the IBM watsonx Orchestrate interface. The left sidebar has a tree view with 'Home' (selected), 'Build', 'Actions', 'Evaluate', 'Preview', 'Publish', 'Environments', 'Improve', and 'Analyze'. The main area shows a 'Build actions' card with 'Enhance and improve your assistant's actions.' Below it is a 'Zeeves' diagram illustrating the flow from 'Default behavior' through 'Actions' (with sub-options like 'General purpose', 'Conversational search', 'AI-guided', 'Skill-based', and 'Custom-built') to 'Fallback' and finally 'Live agent'. The bottom section shows 'Essential channels' and 'Channels' cards, and a 'Extensions' section with tiles for 'SMS', 'Facebook messenger', 'Genesys Bot Connector', 'Slack', 'Microsoft teams', and 'WhatsApp with Twilio'.

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

This screenshot focuses on the 'Integrations' section. It highlights the 'Essential channels' and 'Channels' sections with green dashed boxes. The 'Essential channels' section features a 'Web chat' tile with a phone icon. The 'Channels' section features a grid of tiles for 'SMS', 'Facebook messenger', 'Genesys Bot Connector', 'Slack', 'Microsoft teams', and 'WhatsApp with Twilio', each with an 'Add' button.

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

Integrations
Add different channels and extensions to easily configure and deploy your assistant.

Essential channels
Add our most utilized methods of deploying assistants. These channels support additional customization and advanced integrations.

Web chat
Built by IBM Lite
Embed the web chat onto your company website so it can answer questions.
[Learn more](#)

Contact center integrations
NICE CX One Zixi

Channels
Deploy your assistant to third-party channels to expand your reach.

SMS	Facebook messenger	Genesys Bot Connector	Slack	Microsoft teams	WhatsApp with Twilio
Add +	Add +	Add +	Add +	Add +	Add +

Extensions

4. Click Add.

Integrations
Add different channels and extensions to easily configure and deploy your assistant.

Essential channels
Add our most utilized methods of deploying assistants. These channels support additional customization and advanced integrations.

Web chat
Built by IBM Lite
Embed the web chat onto your company website so it can answer questions.
[Learn more](#)

Contact center integrations
NICE CX One Zixi

Channels
Deploy your assistant to third-party channels to expand your reach.

SMS	Facebook messenger	Genesys Bot Connector	Slack	Microsoft teams	WhatsApp with Twilio
Add +	Add +	Add +	Add +	Add +	Add +

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

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

Slack Draft

Get started (highlighted with a green dotted box)

Through Slack, your assistant is ready to join the collaboration hub that brings the right people, information, and tools together to get work done. [Learn more](#)

Steps to setting up Slack

1. Set up your Slack bot
2. Connect AI assistant builder to Slack
3. Configure your Slack bot
4. Connect your assistant

Close **Next** (highlighted with a green dotted box)

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 regularly. Be sure to stay informed by bookmarking and regularly reviewing the [offering landing page](#) and the [product documentation](#).

Other 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](#)