

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

Skills and skill flows

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

Use cases

Use case: Retrieve

- Download and customize the watsonx Assistant for Z OpenAP JSON file
- Set the
- Import and publish the pre-packaged skills as an OpenAPI file
- Configure your appURL to connect to the WaaS server
- Test the skill
- Connect the app to your assistant

- Create an action for your assistant
- Create a custom-built action
- Test the custom-built action

COMING SOON!

COMING SOON!

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.



Read and follow all the directions.

It is important to read and follow all the documented steps. Skipping steps or sections can cause issues with completing the lab.

Also, invest a few minutes in reading the tips in the [Using the demonstration guide](#) section. The tips will save you time and frustration in completing the lab.

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 learn how to:

- 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
- Publish and deploy an assistant

In addition, you will import pre-packaged z/OS skills and create custom-built skills to deliver an assistant that handles 3 use cases:

- Retrieving initial program load (IPL) information
- Managing certificates
- Resource Access Control Facility (RACF) administration support



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

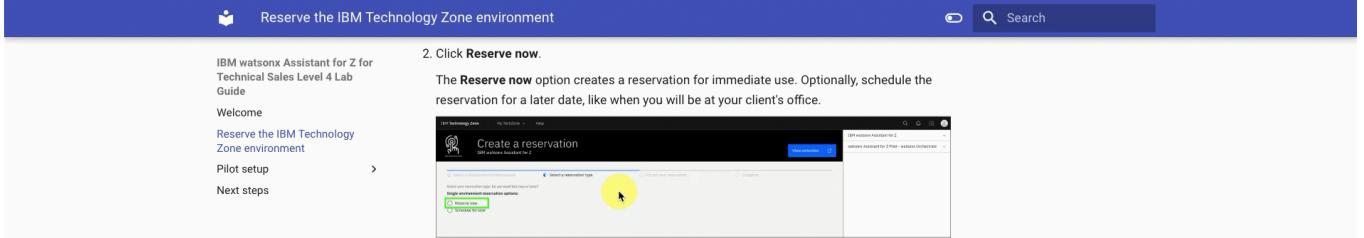
inetegrationID:

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.

- Optionally, 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**.



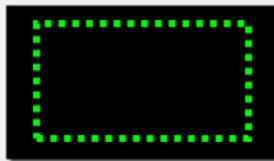
Image highlighting

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. The main content area has a green dashed box highlighting the left sidebar, which contains the Table of Contents. The sidebar lists chapters such as "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. At the top, there are four tabs: 'Select an environment/infrastructure' (selected), 'Select a reservation type' (radio button), 'Fill out your reservation' (radio button), and 'Complete' (radio button). Below these tabs, there is a note: 'Select your reservation type. Do you need this now or later?'. Underneath, it says 'Single environment reservation options:' and shows two radio buttons: 'Reserve now' (selected) and 'Schedule for later'.

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 time | 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.

OCP/Kubernetes Cluster Network: 10.128.0.0/14 | **h**

Enable FIPS Security: No | **i**

Master Single Node Flavor: 16 vCPU x 64 GB - 300 GB ephemeral storage | **j**

OpenShift Version: 4.14 | **k**

OCP/Kubernetes Service Network: 172.30.0.0/16 | **l**

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

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

Submit

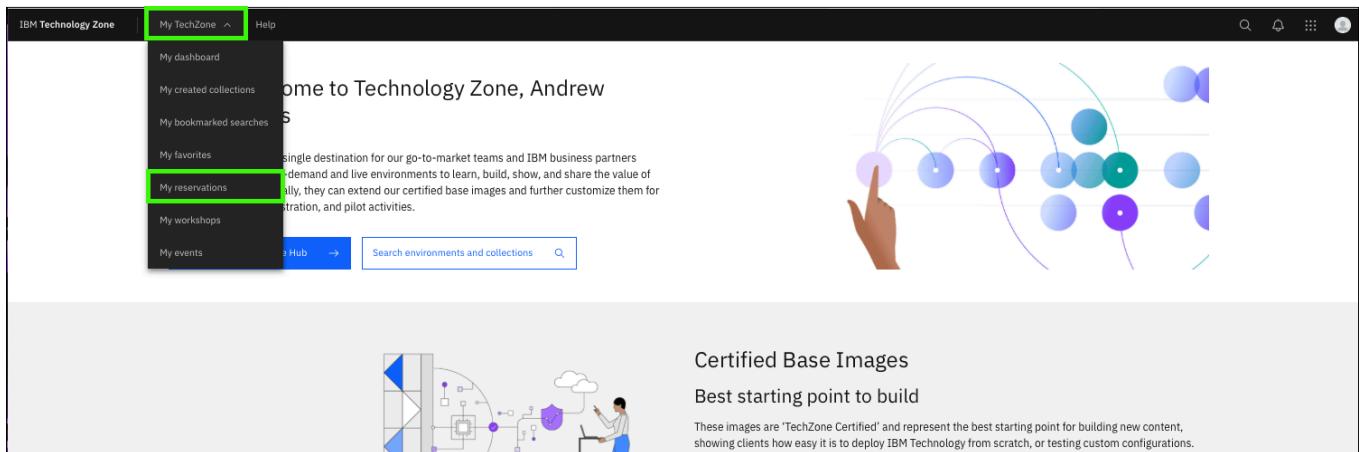
During the provisioning process, multiple emails are sent to you from ITZ as the provisioning process runs. One email states the reservation is provisioning and the other email states that the environment is **Ready**.

In rare cases, the provisioning process 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

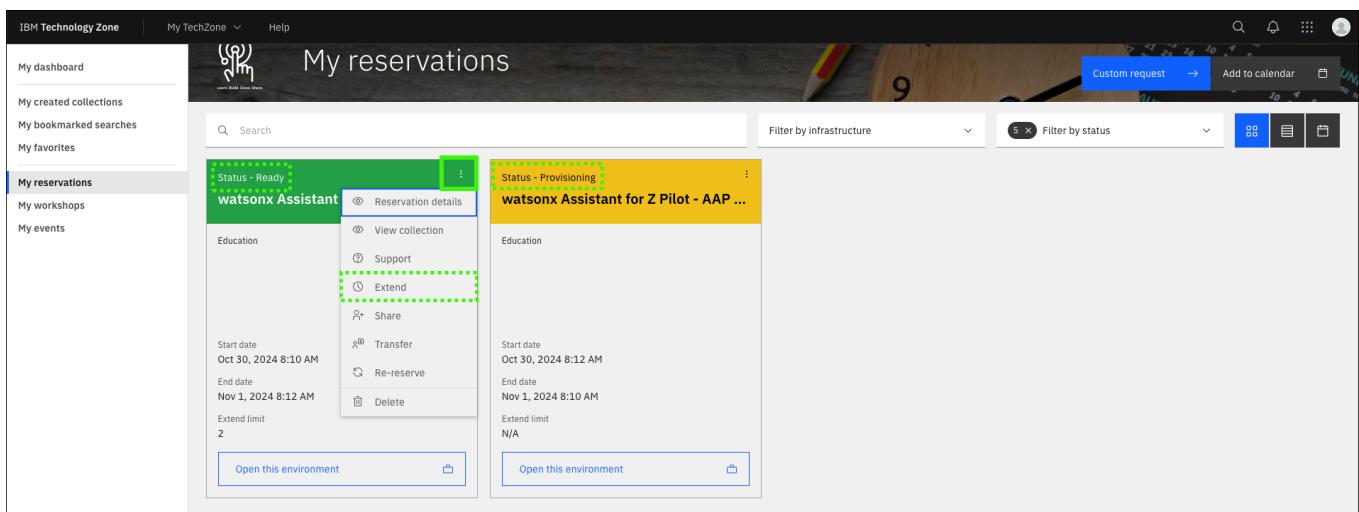
When the reservations are in the **Ready** state, you can extend each reservation beyond its original end date. The duration of the extension will vary by reservation.

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



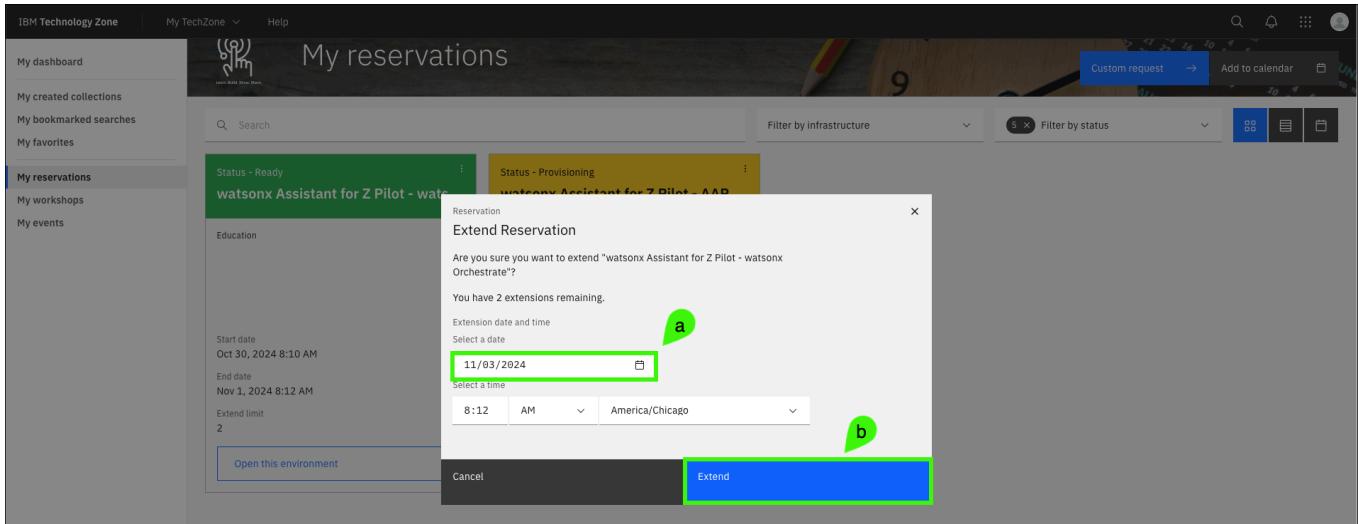
The screenshot shows the IBM Technology Zone portal. The top navigation bar includes 'IBM Technology Zone', 'My TechZone ▾', 'Help', and a search bar. The main content area features a welcome message 'Welcome to Technology Zone, Andrew' and a section titled 'Certified Base Images' with the subtext 'Best starting point to build'. On the left, a sidebar menu lists 'My dashboard', 'My created collections', 'My bookmarked searches', 'My favorites', **My reservations** (which is selected and highlighted with a green box), 'My workshops', and 'My events'. A hand cursor is pointing at the three-dot overflow icon on the 'My reservations' tile.

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, for 'watsonx Assistant', has a green box around its status indicator 'Status - Ready' and the '⋮' overflow icon. A green dashed box highlights the 'Extend' option in the dropdown menu that appears when the icon is clicked. The second card, for 'watsonx Assistant for Z Pilot', shows standard reservation details: Start date Oct 30, 2024 8:10 AM, End date Nov 1, 2024 8:12 AM, and Extend limit N/A. Both cards feature a 'Transfer' and 'Re-reserve' button below their respective details. At the bottom of the page are buttons for 'Open this environment' and a download icon.

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



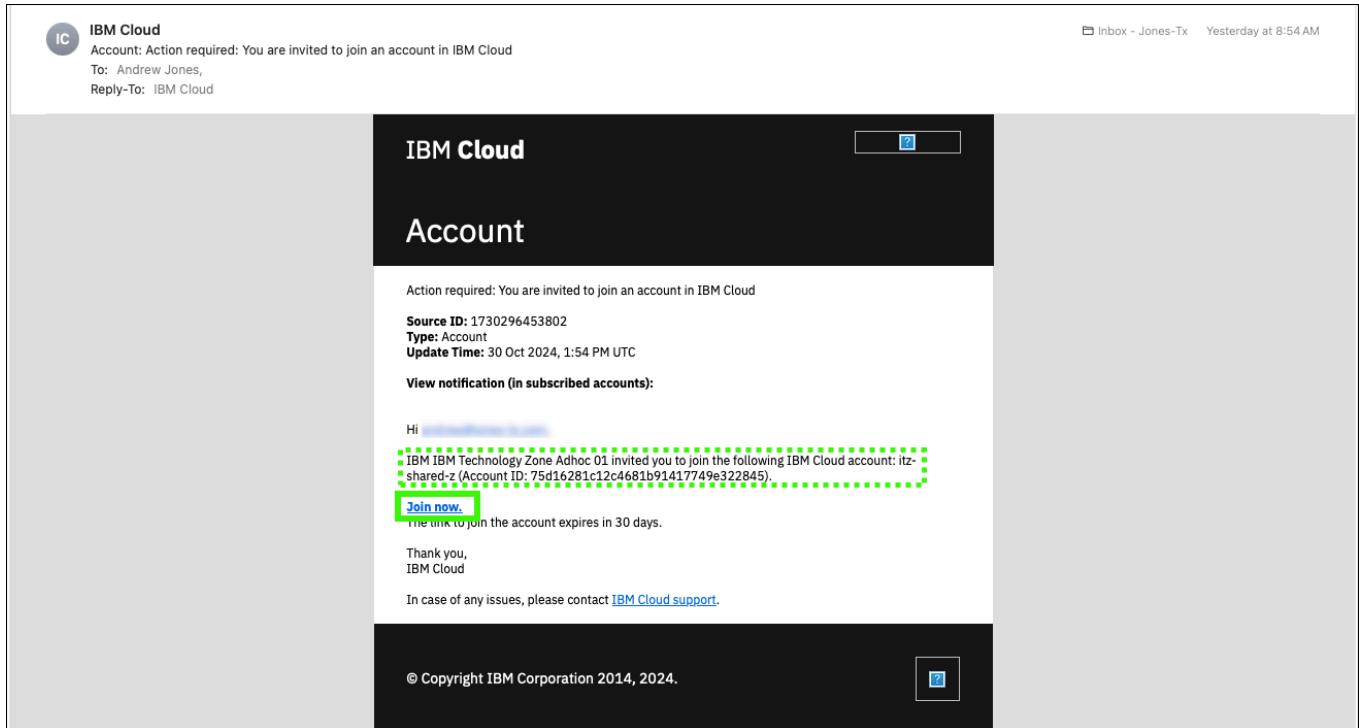
If you anticipate needing more time, repeat Steps 5-6 to extend the reservation to the maximum allowed. Repeat these steps for the other two reservations.

Join the ITZ IBM Cloud account

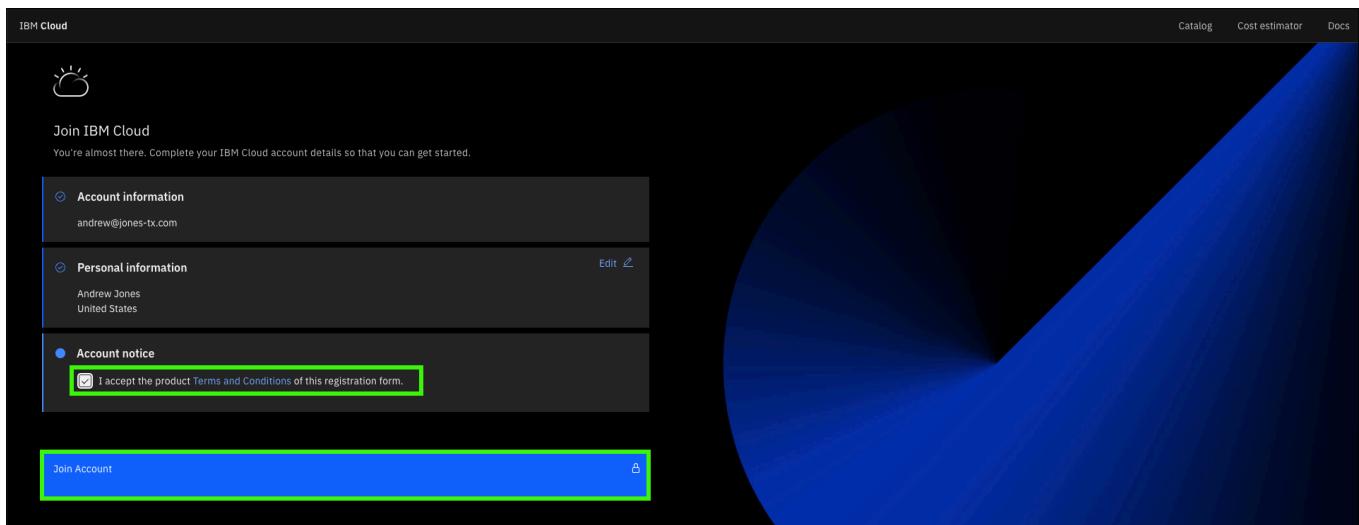
Both the **watsonx Assistant for Z lab – watsonx Orchestrate** and the **Ansible Automation Platform (AAP) & z/OS** environments add you to an IBM Cloud account while your reservation is active. During the provisioning process of these ITZ environments, you receive two emails from IBM Cloud.

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**.

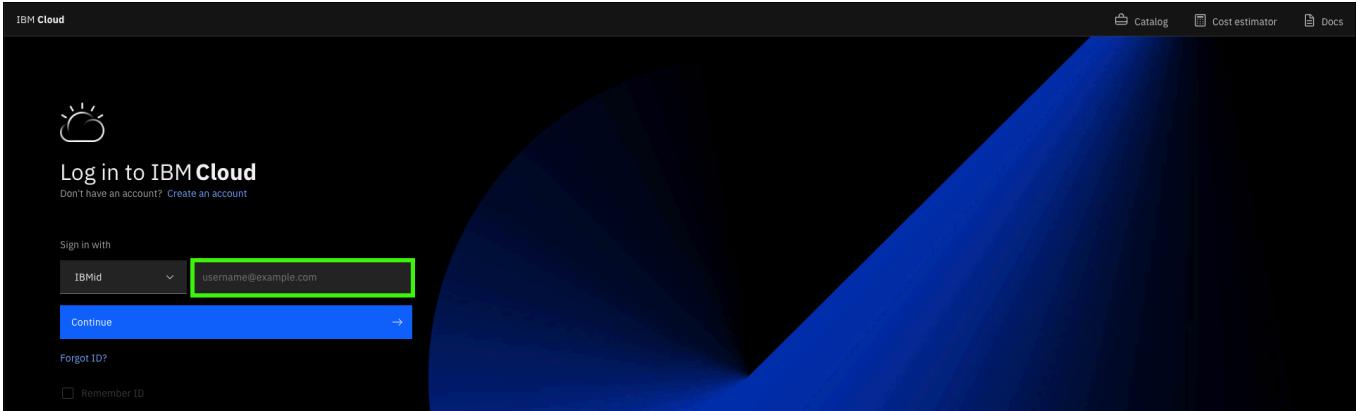


After joining the account, verify that the account appears in your available account list in the IBM Cloud portal.

- 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.

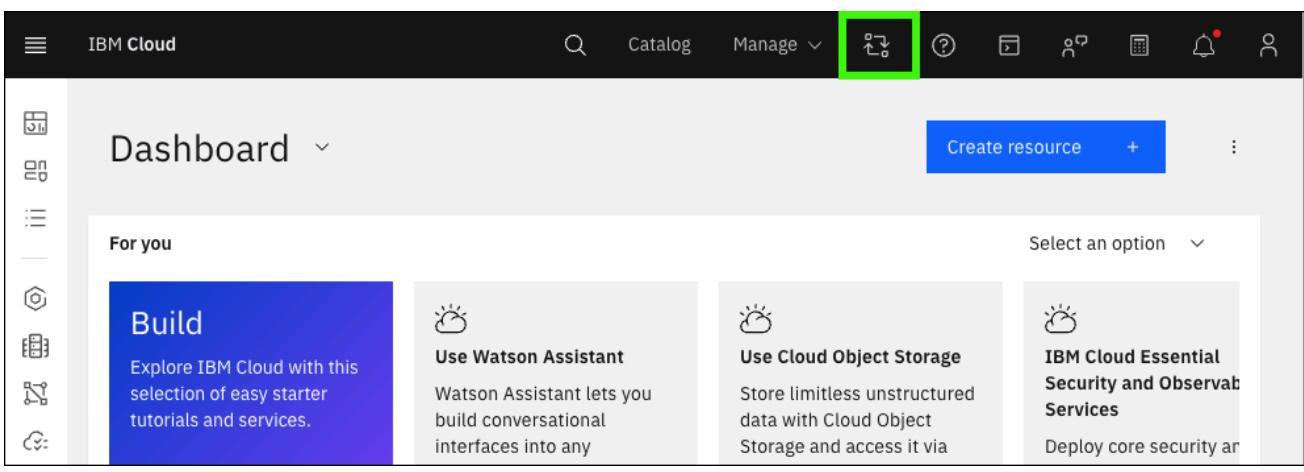


The account may be different.

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...

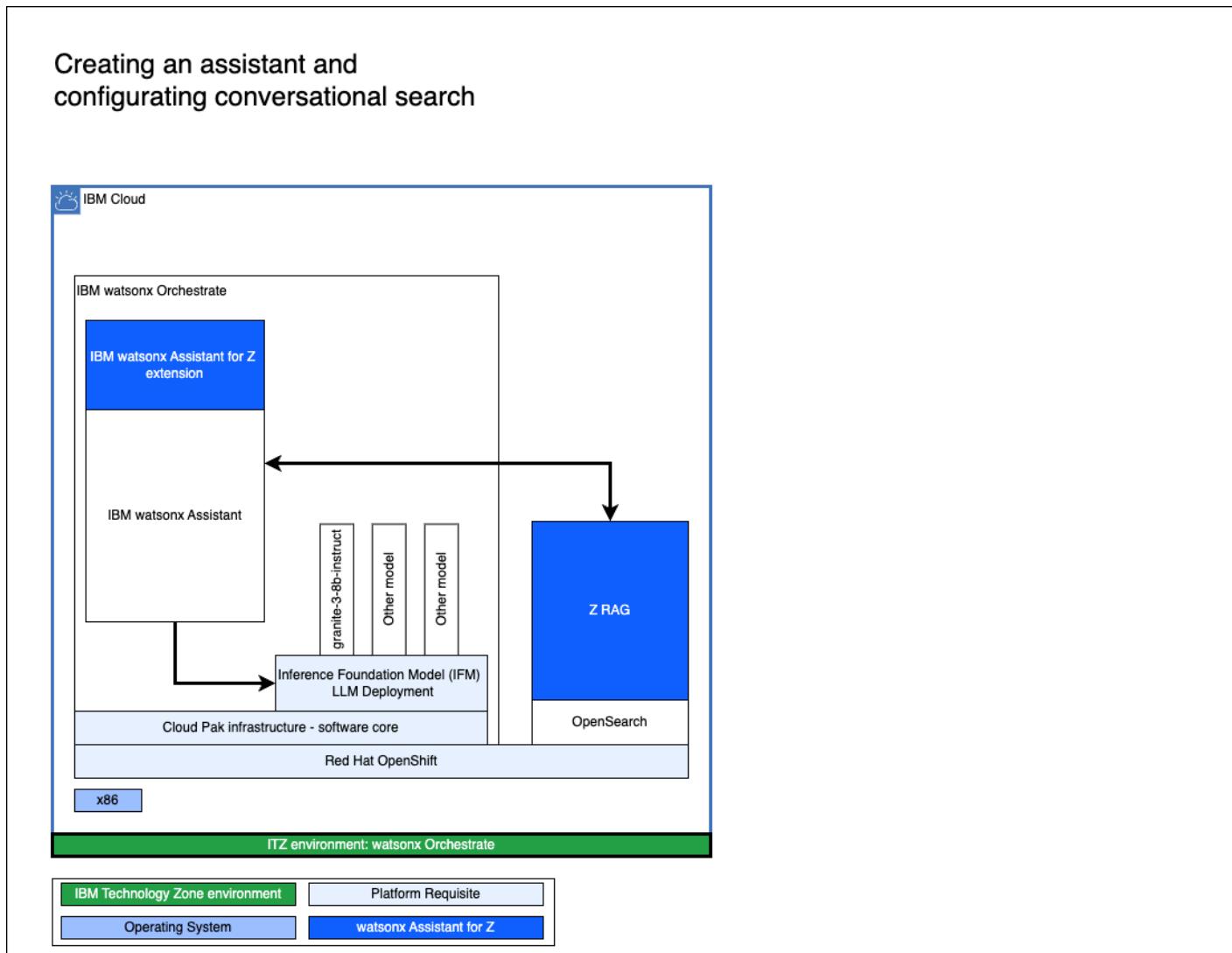
If your reservation for the Single Node OpenShift environment fails, try selecting one of the **eu-gb region** options as the **Preferred Geography**.

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. In the top navigation bar, there are tabs for Catalog and Manage. On the far right, there is a search bar, a 'Create resource' button, and several other icons. The main area is titled 'Resource list' and contains a table with columns: Name, Group, Location, Product, Status, and Tags. The 'Compute' section is expanded in the sidebar, showing sub-options like Containers, Networking, Storage, and Converged infrastructure. A green box highlights the search bar and the 'Create resource' button. Another green box highlights the 'Compute' section in the sidebar.

Create your Assistant

1. Click the **Resources** icon (☰).

The screenshot shows the IBM Cloud Dashboard. In the top navigation bar, there are tabs for Catalog and Manage. On the far right, there is a search bar, a 'Create resource' button, and several other icons. The main area is titled 'Dashboard' and contains a 'For you' section with various cards. One card is highlighted with a green box and labeled 'Build'. Below this, there are other cards for Watson Assistant, IBM Cloud Essential Security & Observability Services, Cloud Object Storage, Maximo Application Suite, Build with Watson, Use Watson Studio, and Get Started with Studio. A green box highlights the 'Build' section in the sidebar.

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. In the top navigation bar, there are tabs for Catalog and Manage. On the far right, there is a search bar, a 'Create resource' button, and several other icons. The main area is titled 'Resource list' and contains a table with columns: Name, Group, Location, Product, Status, and Tags. The 'Enterprise applications' section is expanded in the sidebar, showing sub-options like AI / Machine Learning, Analytics, Blockchain, and Databases. A green box highlights the 'AI / Machine Learning' section in the sidebar. Another green box highlights the specific 'watsonx-orchestrate-erspw' instance in the table.

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):

40/128

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?
Web

Tell us about yourself
This information will be used to personalize your onboarding experience.

Which industry do you work in?
Software

What is your role on the team building the assistant?
Developer

Which statement describes your needs best?
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 Add IBM Watermark

Streaming
Enable Streaming
 Off

Zeeves

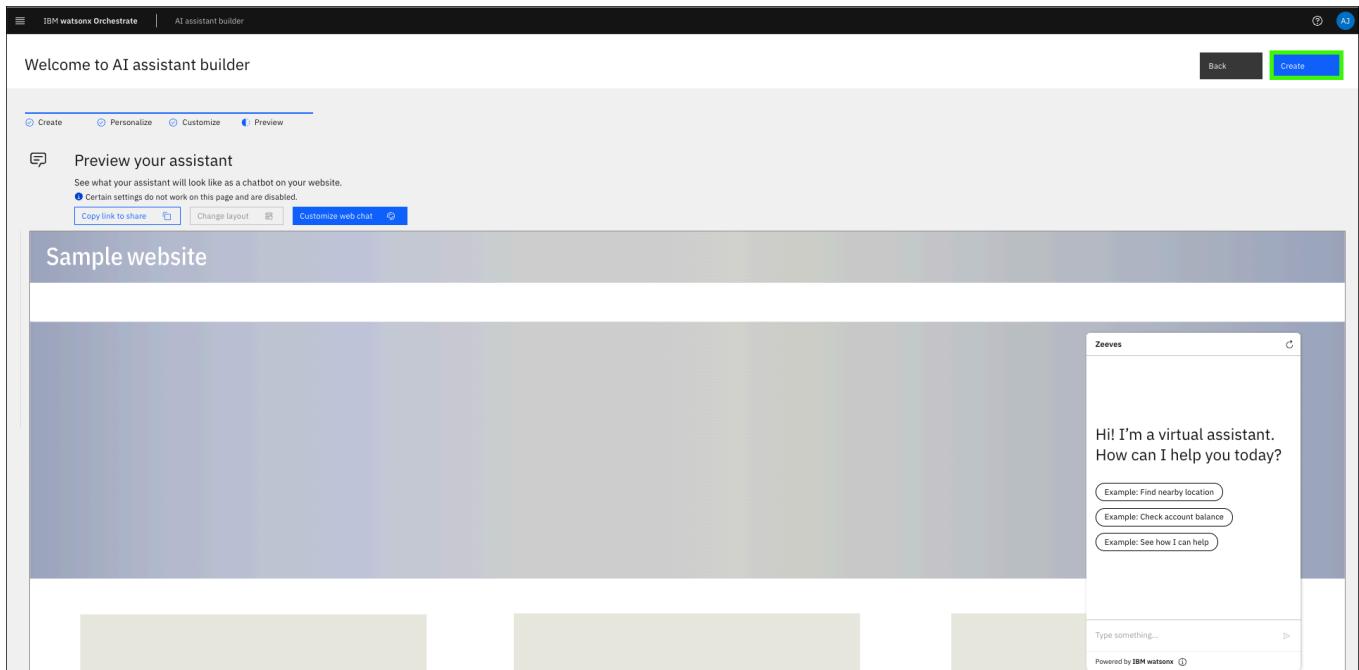
Hi! I'm a virtual assistant.
How can I help you today?

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

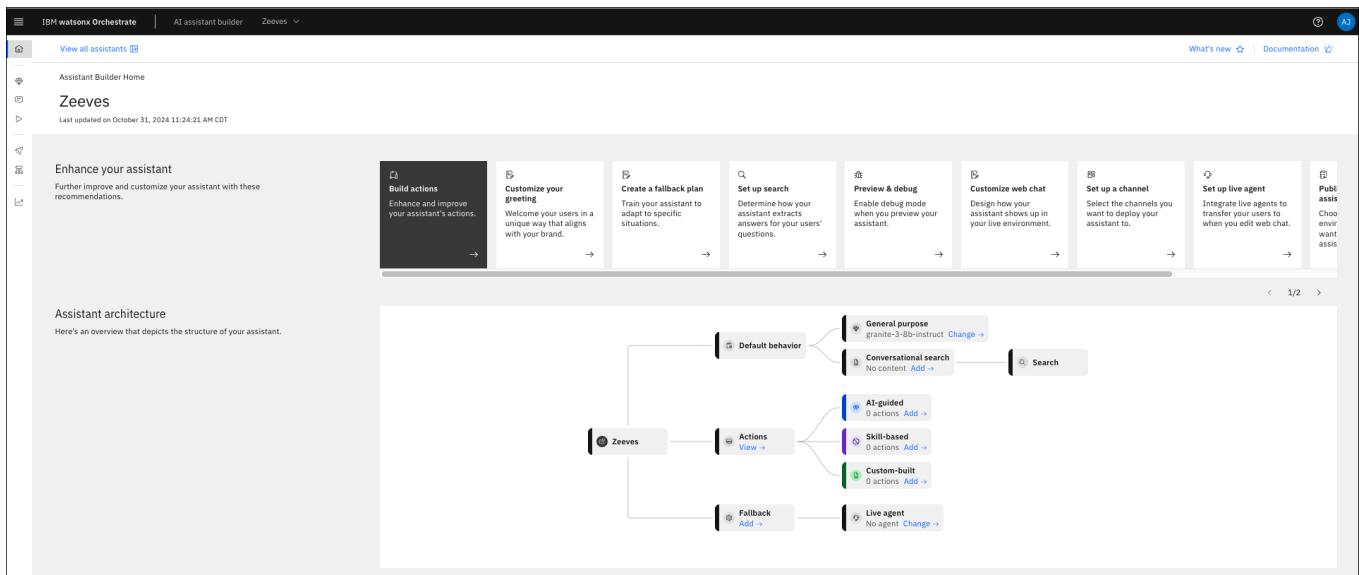
Type something... ➤

Powered by IBM watsonx

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



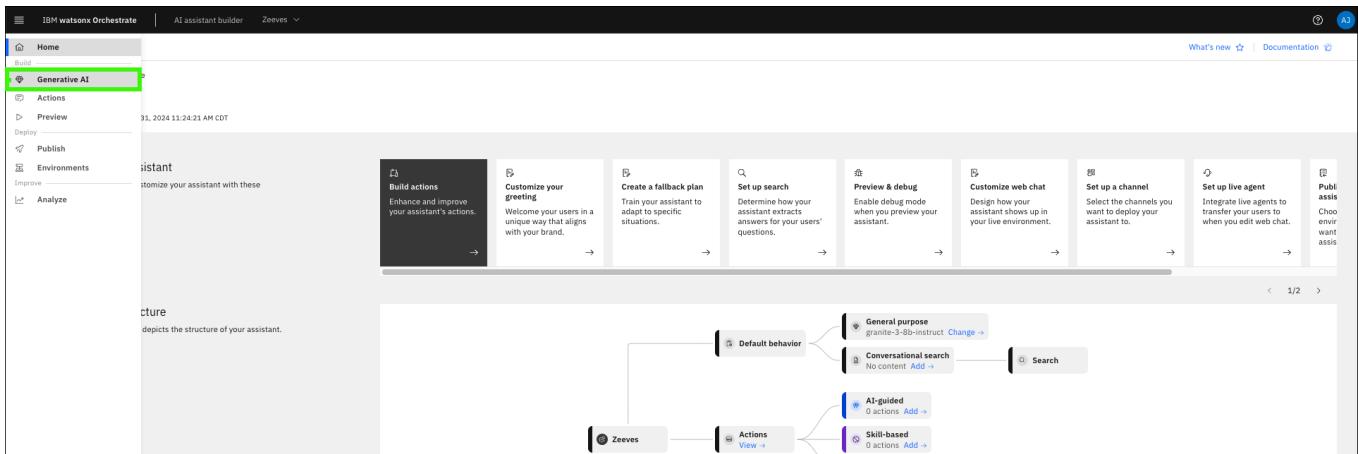
The assistant is now created.



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.

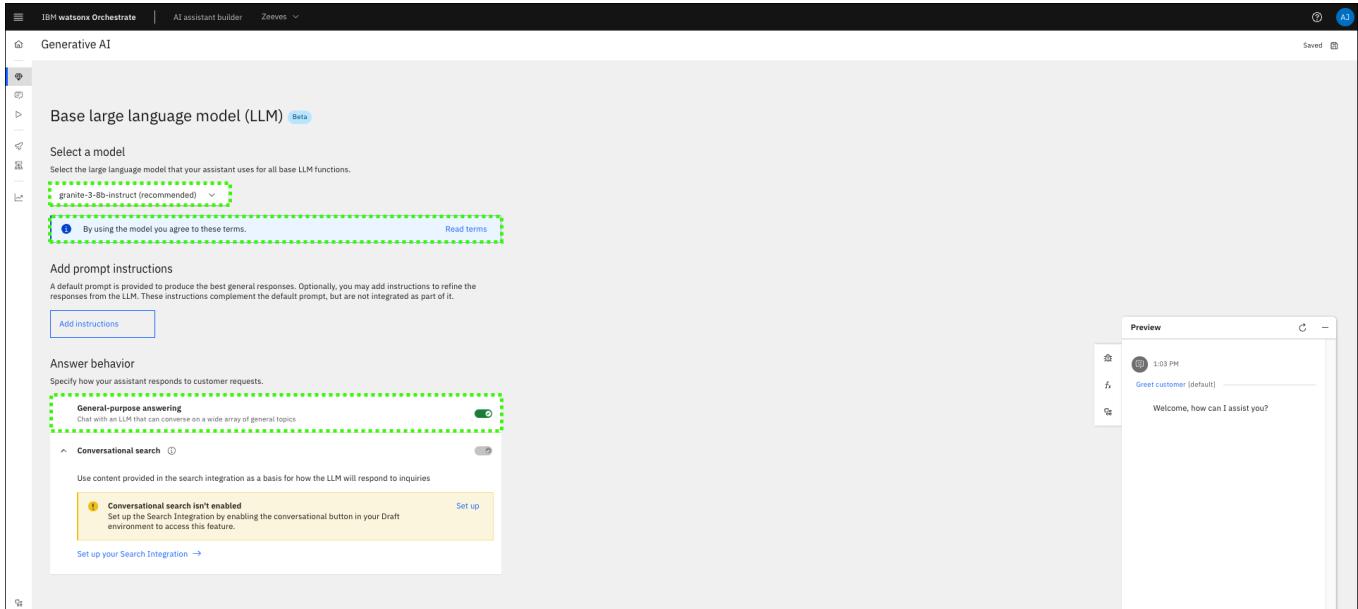


2. Select the base large language model (LLM) settings.



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

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

granite-3-8b-instruct (recommended)

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

Add prompt instructions

A default prompt is provided to produce the best general responses. Optionally, you may add instructions to refine the responses from the LLM. These instructions complement the default prompt, but are not integrated as part of it.

[Add instructions](#)

Answer behavior

Specify how your assistant responds to customer requests.

General-purpose answering

Chat with an LLM that can converse on a wide array of general topics

Conversational search

Use content provided in the search integration as a basis for how the LLM will respond to inquiries

Conversational search isn't enabled

Set up the Search Integration by enabling the conversational button in your Draft environment to access this feature.

[Set up your Search Integration →](#)

Preview

1:03 PM Greet customer [default]

Welcome, how can I assist you?

4. Click Custom service.

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

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

granite-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

A default prompt is provided to produce the best general responses from the LLM. These instructions complement the prompt provided by the LLM.

[Add instructions](#)

Answer behavior

Specify how your assistant responds to customer requests. Consider enabling both General-purpose answering and Conversational search for better performance.

General-purpose answering

Chat with an LLM that can converse on a wide array of general topics

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

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.

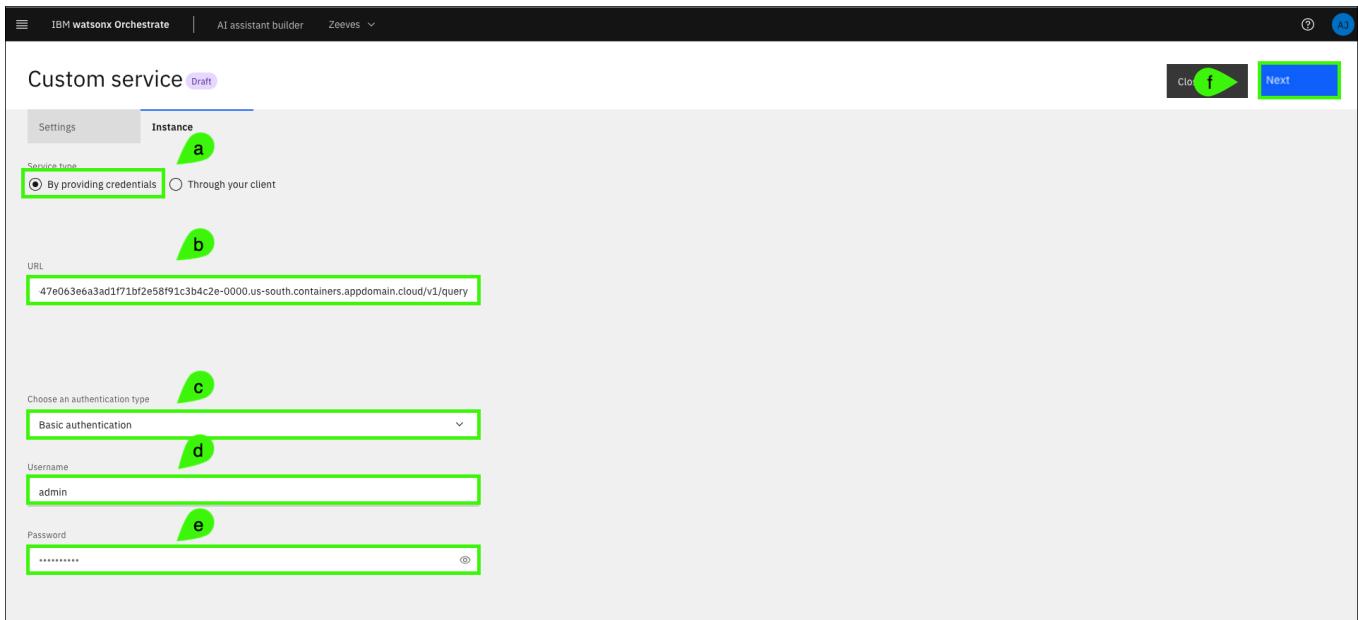
```
https://wxa4z-opensearch-wrapper-wxa4z-demo-v2-1-0.wxo4z-opc-opensearch-clus-47e063e6a3ad1f71bf2e58f91c3b4c2e-0000.us-south.containers.appdomain.cloud/v1/query
```

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

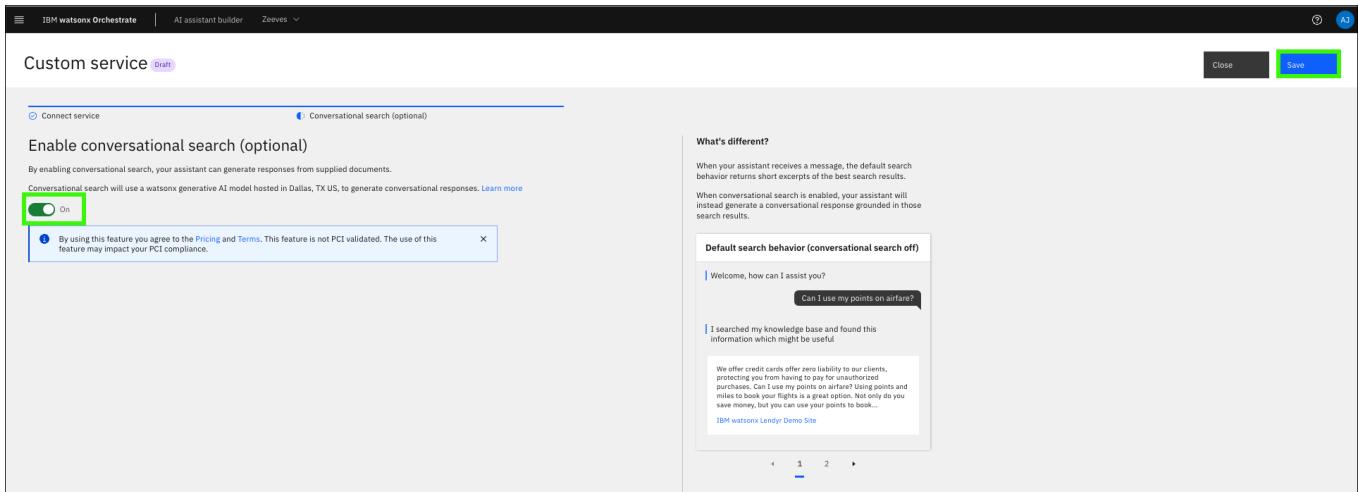
d. Enter **admin** in the **Username** field.

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](#).

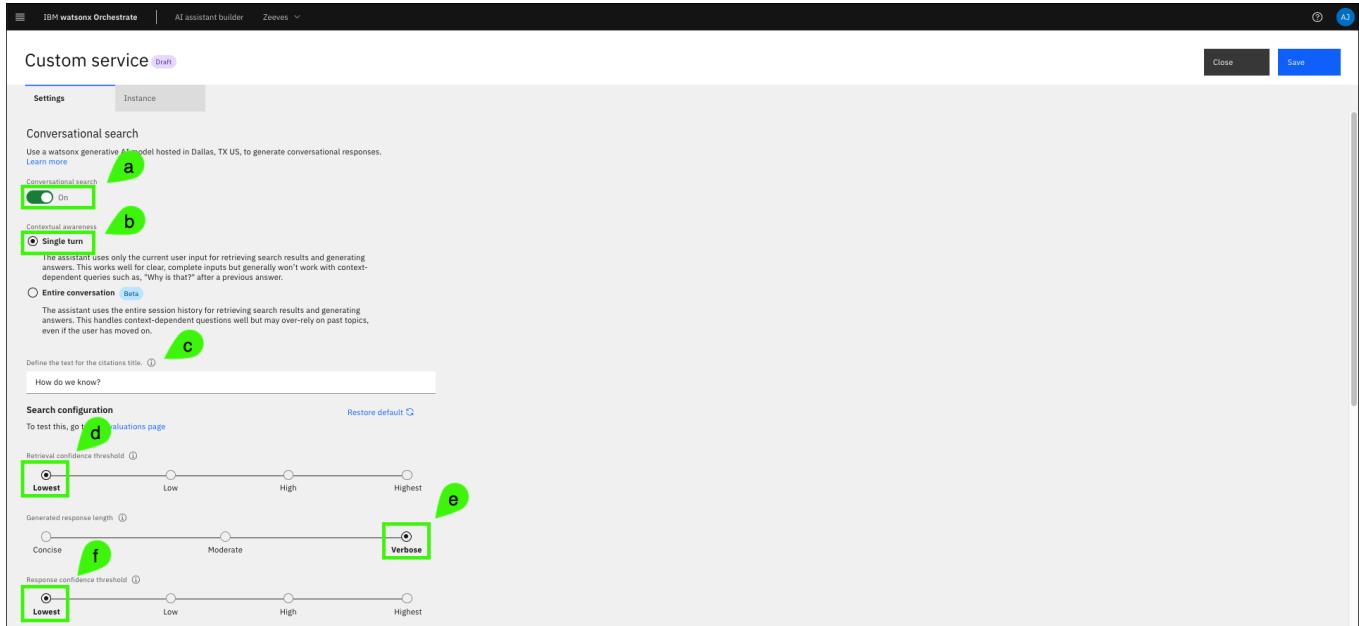
The following settings are proven to work well. You can experiment with these settings to see how they affect queries for your client's pilot.

- 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.

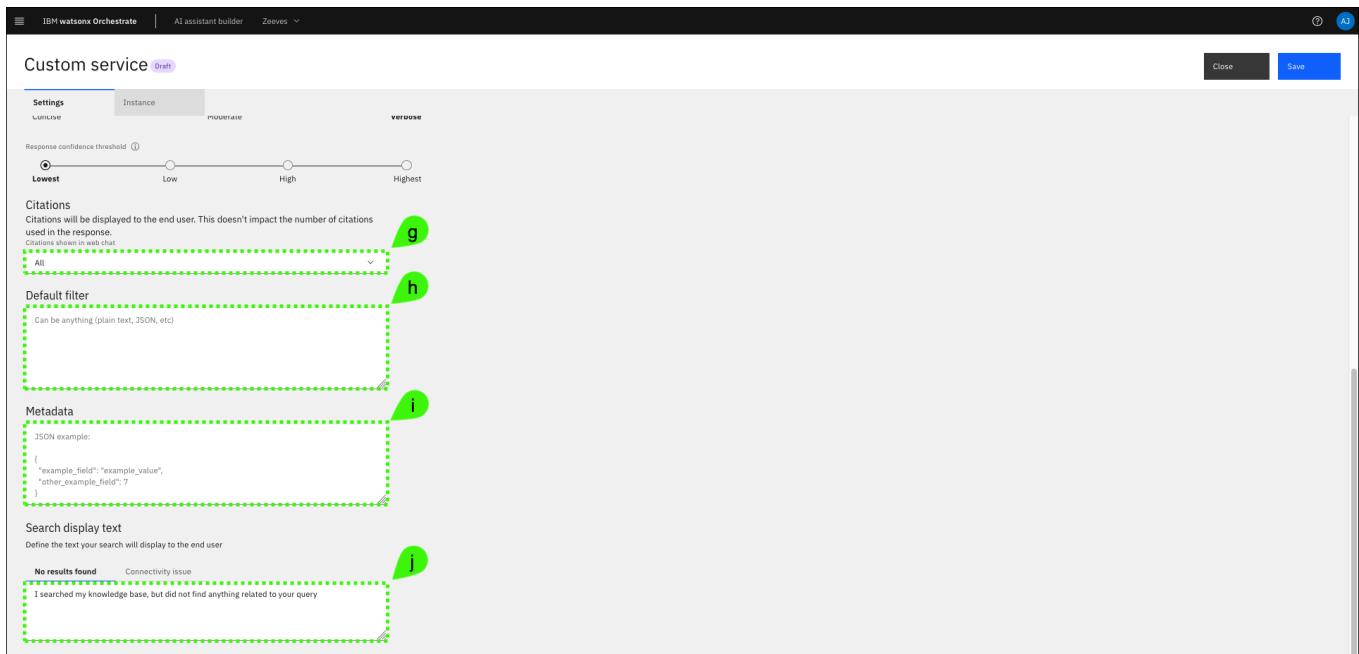


g. Keep the default setting of **All** for the listing of citations.

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**.

Generative AI

Actions

language model (LLM) Beta

model that your assistant uses for all base LLM functions.

recommended)

del you agree to these terms. [Read terms](#)

ctions

ded to produce the best general responses. Optionally, you may add instructions to refine the 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.

All items

Created by you

Set by assistant

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. It displays two steps: Step 1 and Step 2. Step 1 is highlighted with a green box. The 'Action starts' section indicates that no action can be matched to the customer's message, and it includes a tip about using a default retry message or customizing it.

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 currently set to 'with conditions'. A modal dialog box 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 configuration of Step 1. The 'Is taken' dropdown has been changed to 'without conditions'. The 'Assistant says' field is highlighted with a green box. The field contains placeholder text: '| or example: Please select from the following options:'.

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 'Conversation steps' section, step 1 is highlighted with a green dashed border. The 'Assistant says' section contains a text input field with placeholder text: '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 selected and highlighted with a green box. A sub-menu for 'Search for the answer' is open, listing 'Continue to next step', 'Re-ask previous step(s)', 'Go to a subaction', 'Use an extension', and 'Search for the answer'. The 'Search for the answer' option is also highlighted with a green box.

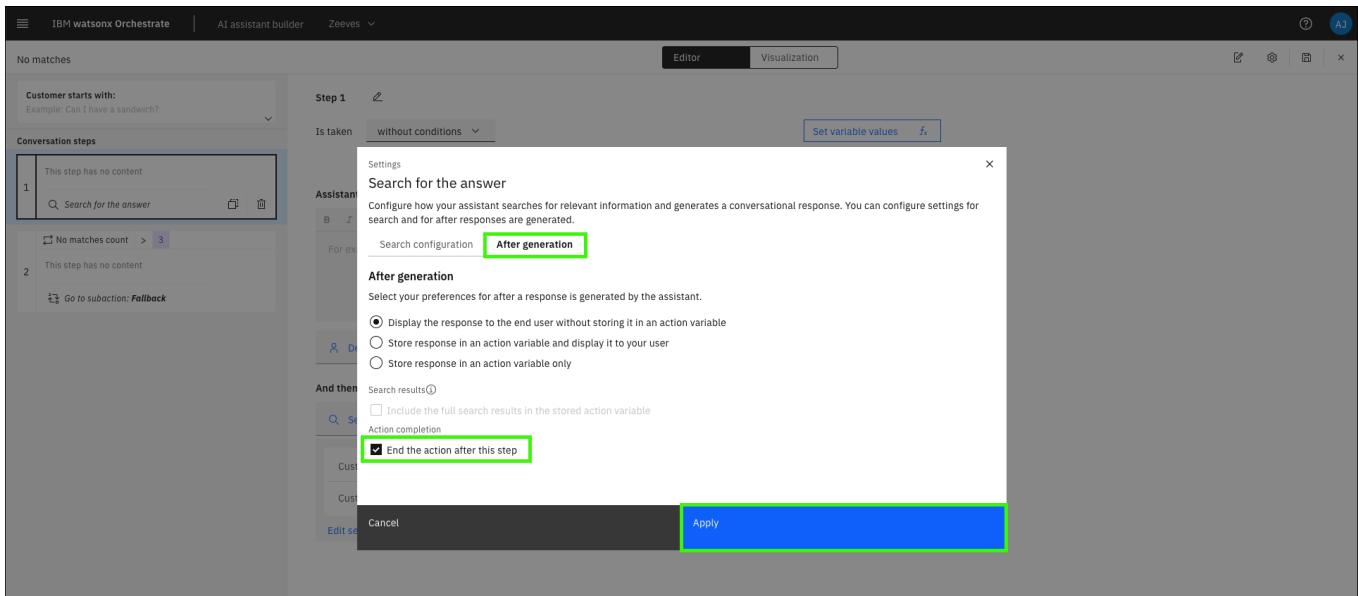
8. Click Edit settings.

This screenshot is similar to the previous one but shows the 'Edit settings' button at the bottom of the 'Search for the answer' sub-menu highlighted with a green box.

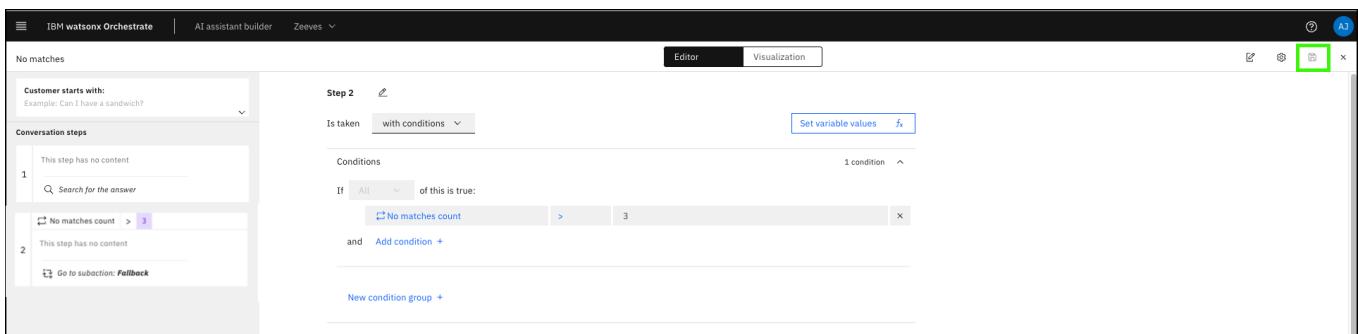
9. Click After generation.

A modal dialog box titled 'Search configuration' is displayed over the main interface. It has tabs for 'Search configuration' (which is selected) and 'After generation'. The 'Search configuration' tab contains fields for 'Custom search query (Optional)' and 'Custom results filter' (with 'Use default' selected). The 'After generation' tab contains a note about configuring settings for search and responses. At the bottom of the dialog are 'Cancel' and 'Apply' buttons, with 'Apply' being the active button.

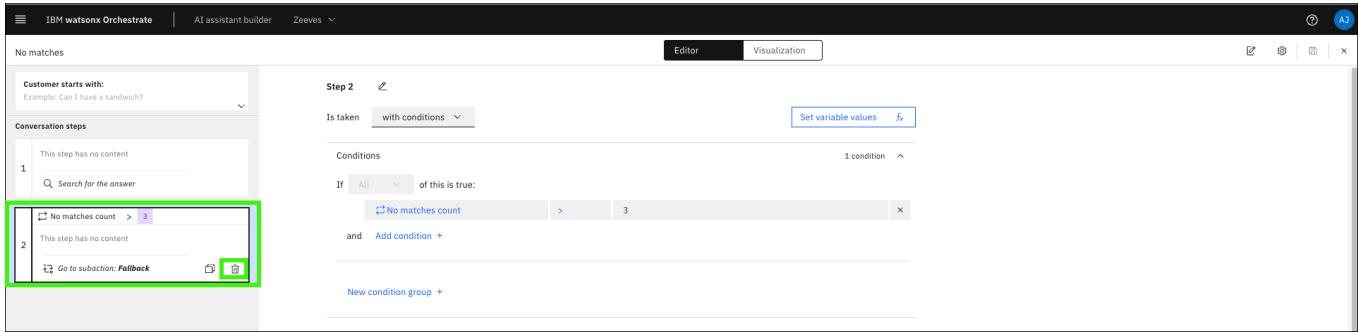
10. Select End the action after this step and then click Apply.



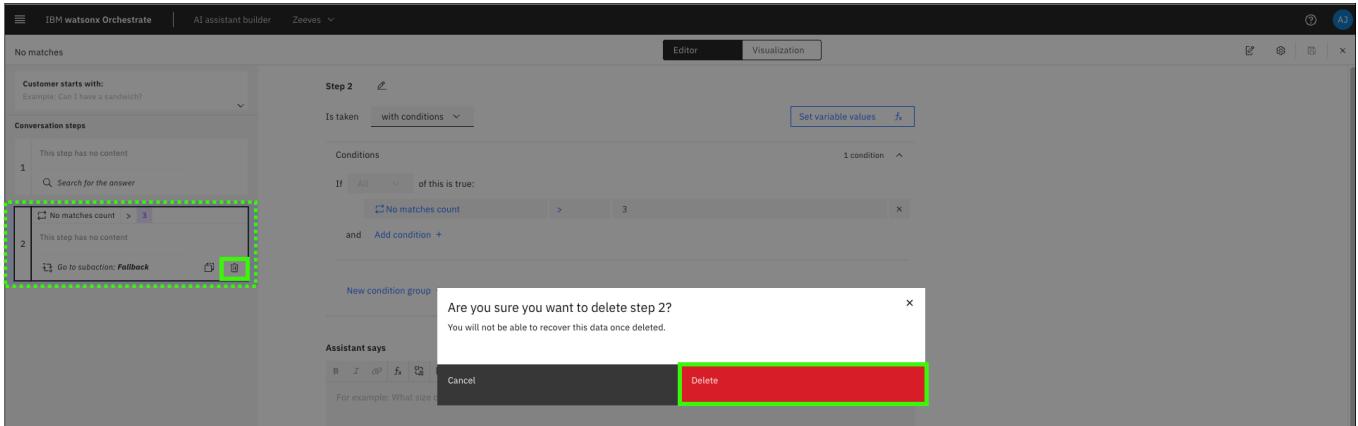
11. Click Save (💾).



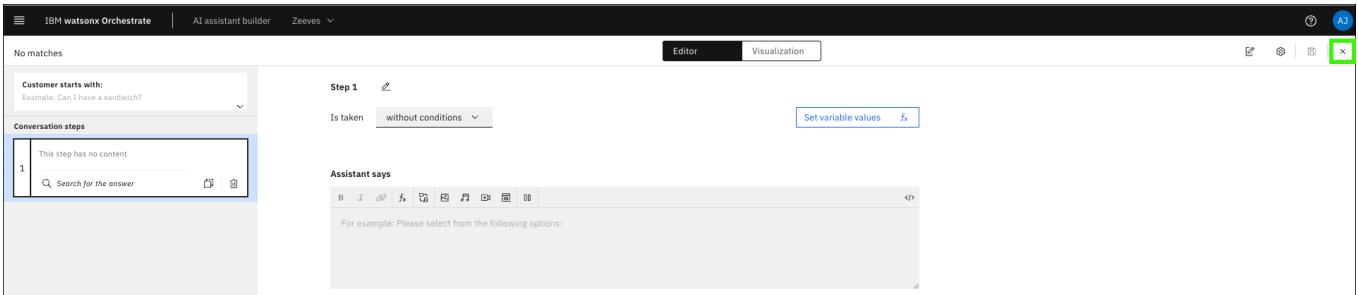
12. Select Step 2 (No matches count) under Conversation steps and click delete (🗑).



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.

The screenshot shows the AI assistant builder interface. On the left, under 'Conversation steps', there are five steps listed, each with a green border around it. Step 1: 'Fallback reason is Failed to confirm topic'. Step 2: 'Fallback reason is Step validation failed'. Step 3: 'Fallback reason is Agent requested'. Step 4: 'Fallback reason is No matches'. Step 5: 'Fallback reason is Danger word detected'. To the right, the 'Editor' tab is open, showing the configuration for Step 1. It includes conditions like 'If All of this is true:' and 'Assistant says' (Sorry I couldn't confirm if you wanted to return to previous topic, let me connect to an agent.). Below this, 'And then' actions like 'Connect to agent (action ends)' are defined.

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

The screenshot shows the AI assistant builder interface after the steps have been deleted. The 'Conversation steps' section is now empty, indicated by a dashed green border. The 'Editor' window is still open on the right, showing the configuration for Step 1, but the steps themselves are no longer present.

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

The screenshot shows the 'Actions' section of the Global settings page. On the left, a sidebar lists actions categorized by creator: 'All items', 'Created by you', 'Set by assistant', 'Variables', 'Created by you', 'Set by assistant', 'Set by integration', and 'Saved responses'. On the right, a table lists items with columns: Name, Last edited, Examples Count, and Status. The items listed are 'Greet customer', 'Trigger word detected', 'No matches', and 'Fallback', all last edited 4 days ago and in a green status.

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

No matches (Beta)

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

Customize modes [Beta](#)

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

No matches (Beta)

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

No matches (Beta)

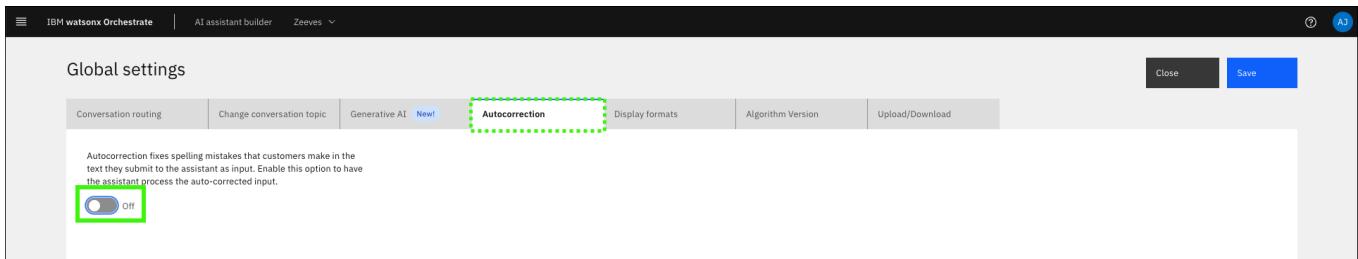
Unrecognized input by customers triggers the **No matches** action that can be configured to fetch answers from a [search integration](#) or trigger the [Fallback](#) action.

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

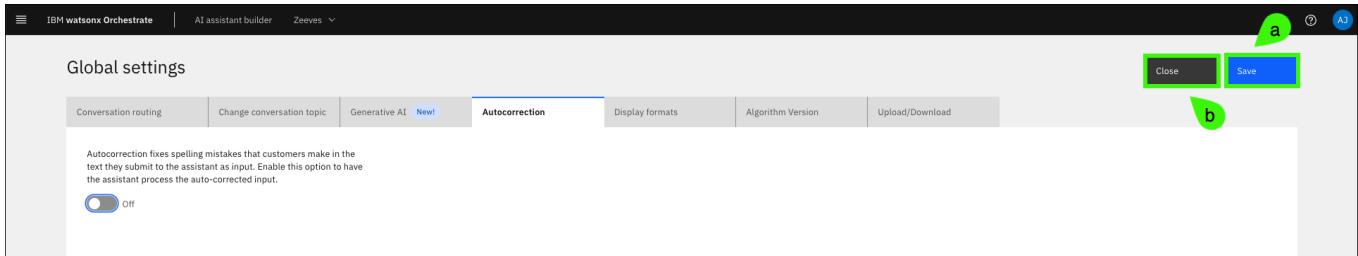
Use "No matches"

Save

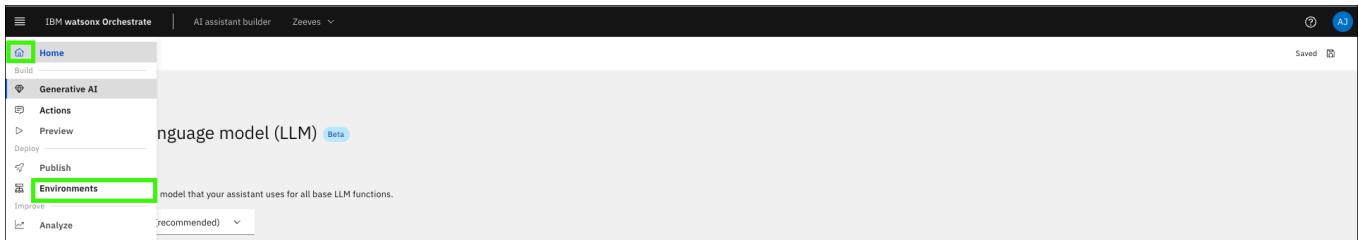
22. Click the autocorrection toggle to turn the feature **Off**.



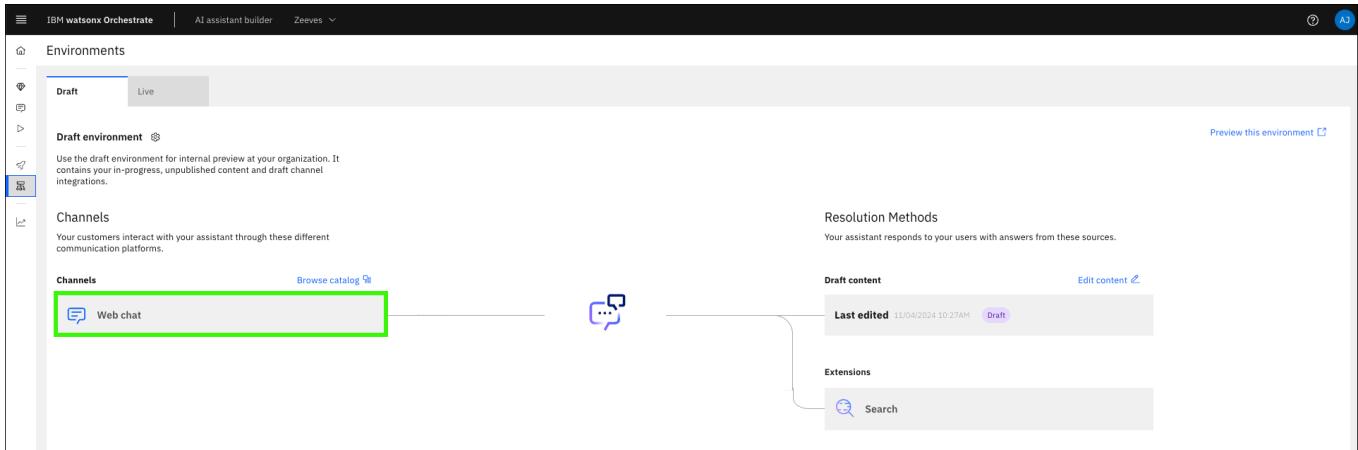
23. Click **Save** (a) and then **Close** (b).



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

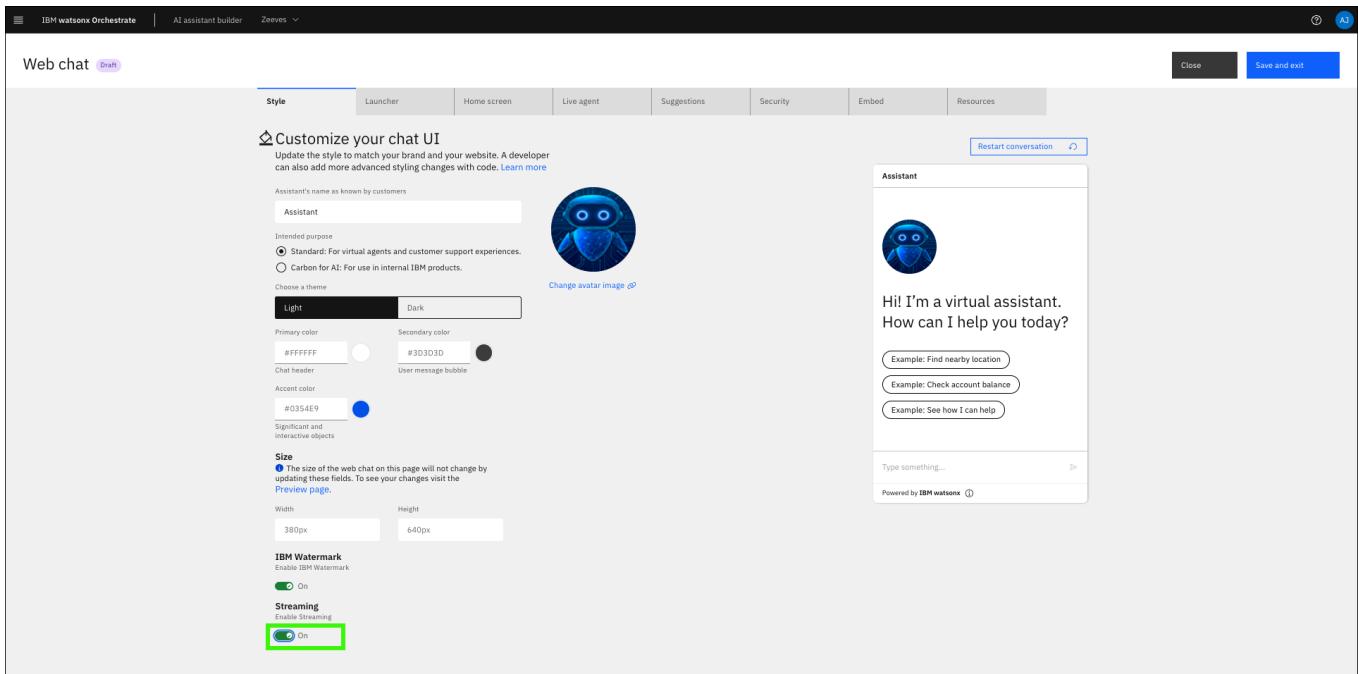


25. Click **Web chat**.

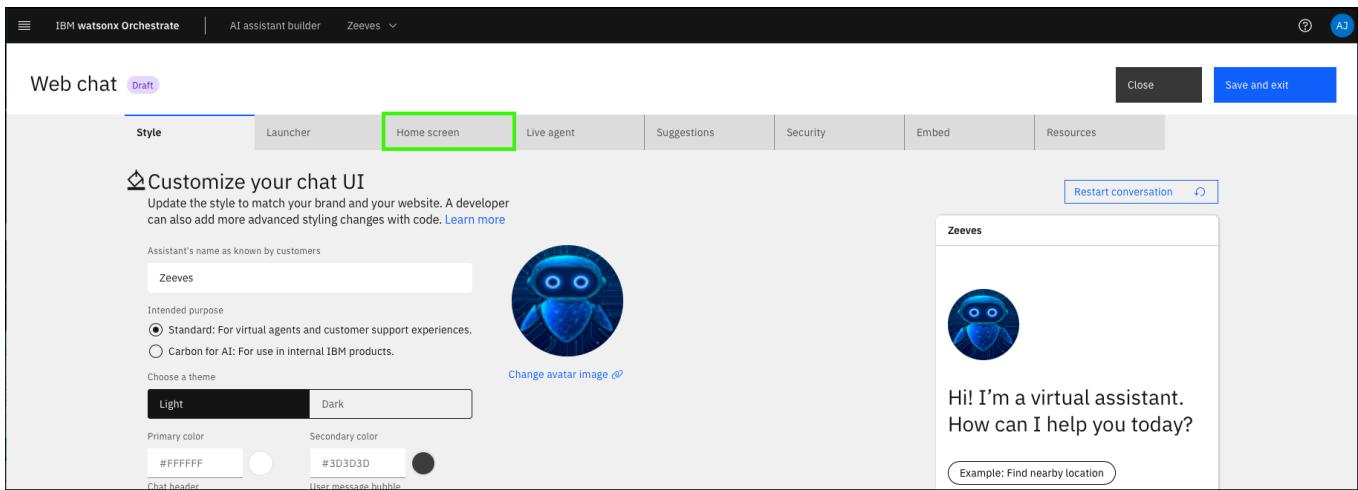


26. On the **Style** tab, click the **Streaming** toggle to enable streaming.

The streaming setting allows responses to be streamed to the assistant and displayed as they are generated versus waiting until the full response is received and then displayed.



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

Add a starter (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

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

Zeeves

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 ⓘ

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?

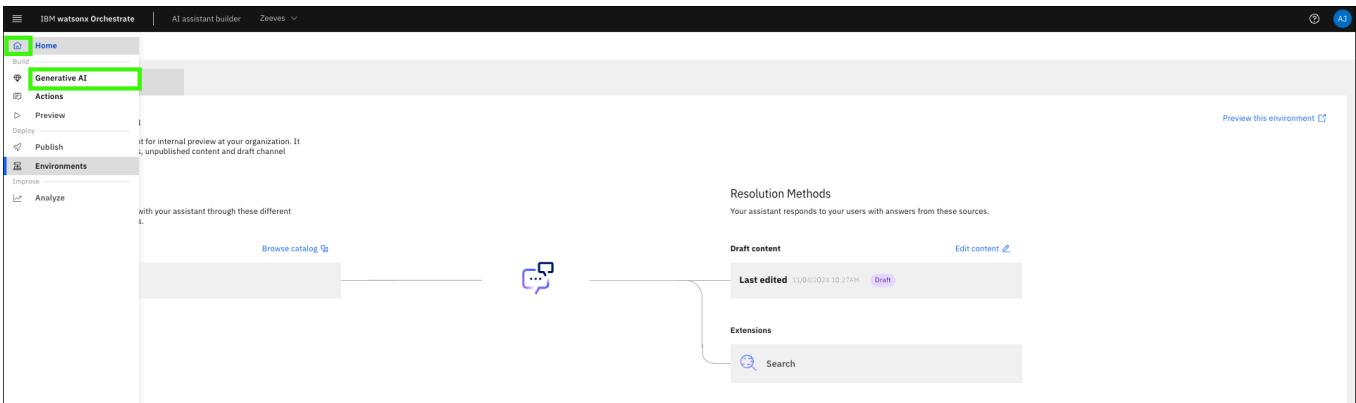
Type something... ▶

Built with IBM watsonx ⓘ

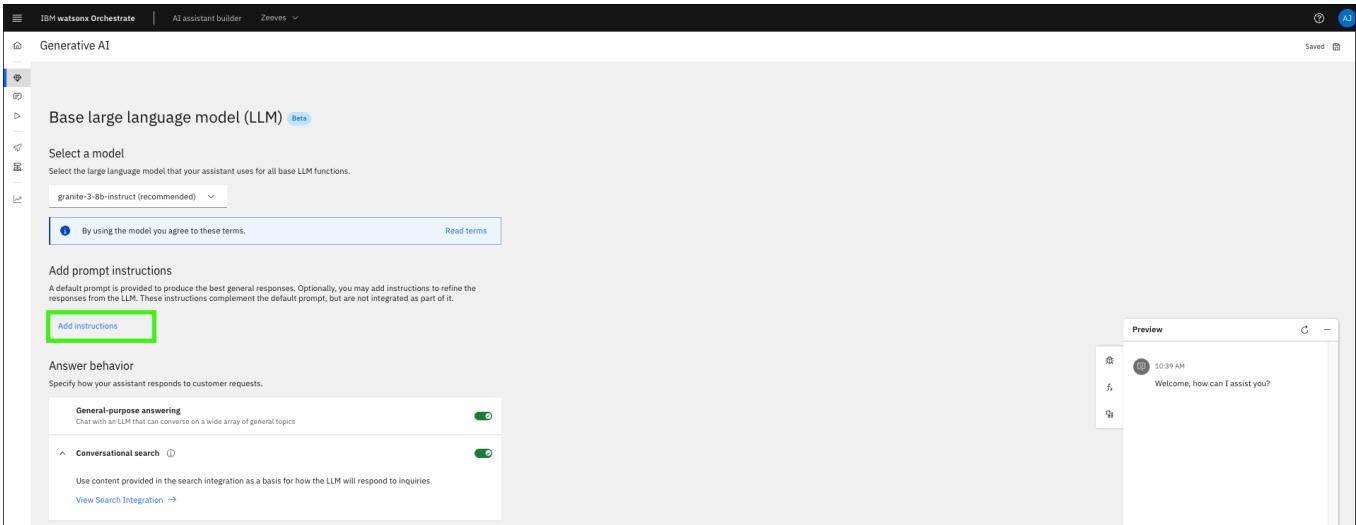
Configure the base large language model

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 green 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.

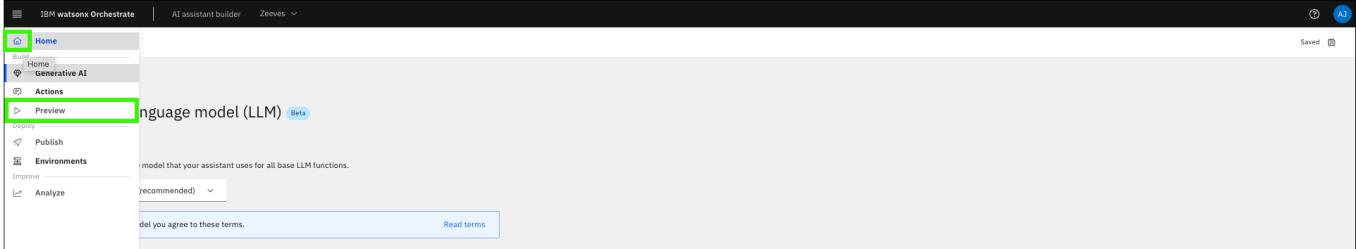
The screenshot shows the 'Generative AI' configuration page. In the 'Answer behavior' section, the 'General-purpose answering' toggle switch is turned off (indicated by a greyed-out appearance). The 'Conversational search' toggle switch is turned on (indicated by a green background). A preview window 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 green 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 Orchestrate interface. On the left, there's a sidebar with icons for preview, AI assistant builder, and Zeeves. The main area is titled "Sample website". On the right, there's a "Zeeves" chat window. The chat history shows a user asking "What is the APF list in z/OS? Provide a detailed explanation?" and the AI assistant responding with a detailed explanation of the APF list, mentioning its purpose in controlling access to system resources.

Prompt:

Why is Db2 different than other database systems?

Example output:

The screenshot shows the IBM Watsonx Orchestrate interface. On the left, there's a sidebar with icons for preview, AI assistant builder, and Zeeves. The main area is titled "Sample website". On the right, there's a "Zeeves" chat window. The chat history shows a user asking "Why is Db2 different than other database systems?" and the AI assistant responding with an explanation, mentioning that Db2 is a database management system (DBMS) that runs under the z/OS operating system, which is specific to mainframe computers. Other database systems may run on various operating systems like Linux, Windows, or Unix.

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'.

On the left, there's a sidebar with icons for preview, copy link, change layout, change background, and customize web chat. The main content area is a large, mostly blank space with a light blue gradient background.

To the right, there's a chat window titled 'Zeeves'. It shows a message from 'You' at 11:05 AM: 'What happens during an IPL on IBM Z?'. A green box highlights this message. The AI response starts with '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 management console. 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.' A green box highlights the beginning of the response. Below the response, there's a 'Type something...' input field and a 'Powered by IBM Watson' footer.



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 box around the 'Entire conversation' radio button. The 'Entire conversation' option is selected, while 'Single turn' is unselected. A tooltip for 'Entire conversation' provides a detailed description of how it works.

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

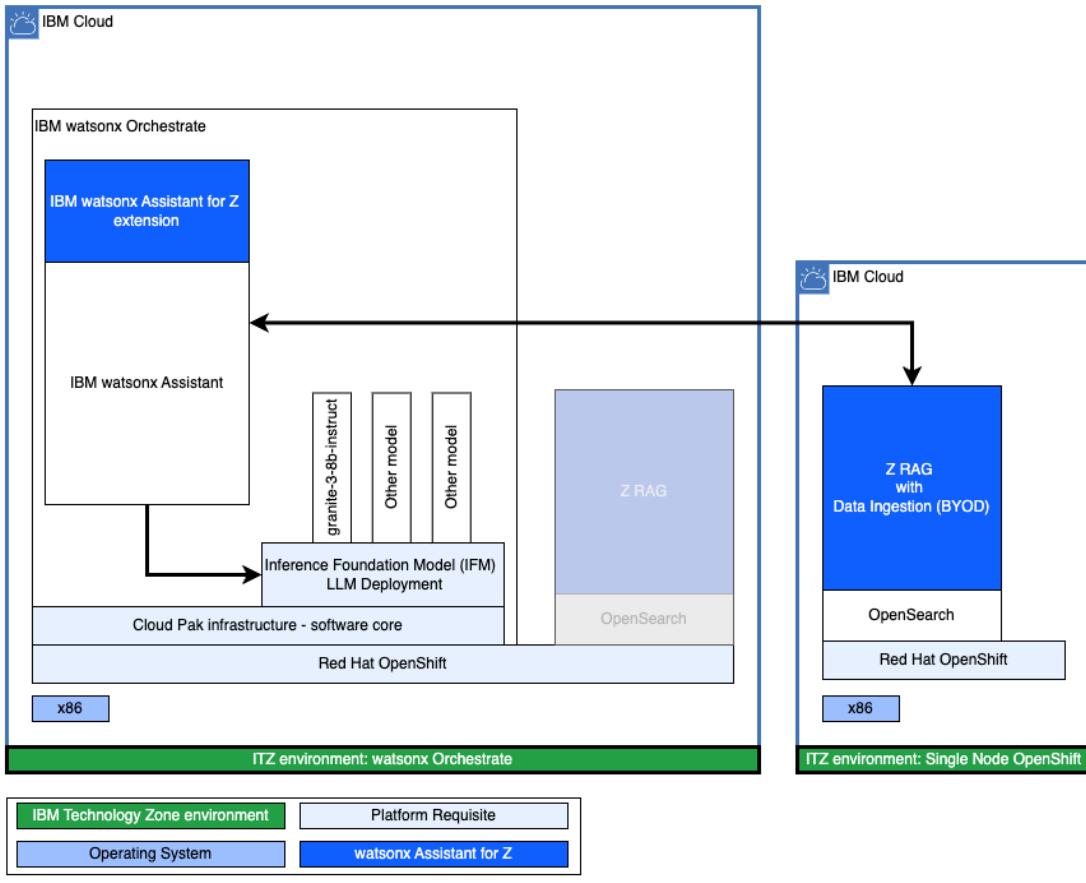
Now that you created and deployed your own assistant with conversational search capabilities, your client can understand how watsonx Assistant for Z provides its content-grounded responses to any Z-related questions. In the previous section, you configured your assistant to use a pre-configured Z RAG 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](#)

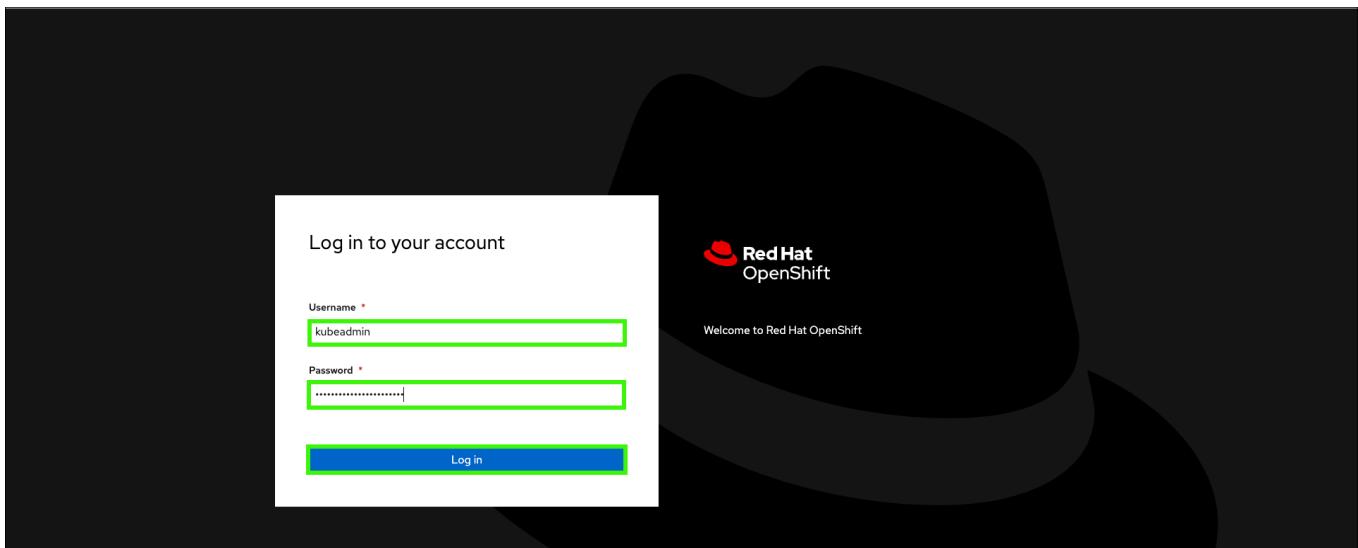
2. Click the **Single Node OpenShift** tile.

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

4. Click the OCP Console link.

Note: OCP stands for OpenShift Container Platform.

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



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

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

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

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](#)
- [Download oc for Mac for x86_64](#)
- [Download oc for Windows for x86_64](#)
- [Download oc 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](#)

helm - Helm 3 CLI

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

5. 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

6. 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

7. Click **Display Token**.

Display Token

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

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

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), 'OperatorHub' (which is also selected), 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 this, there's a section titled 'Command Line Tools' with a 'Copy login command' button. Another section titled 'oc - OpenShift Command Line Interface (CLI)' provides instructions on how to use the CLI and links to download it for various platforms (Linux, Mac, Windows, ARM).

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

The screenshot shows the Red Hat OpenShift interface on IBM Techzone. The left sidebar has 'Administrator' selected. Under 'Operators', 'OperatorHub' is selected. The main area shows a dropdown menu for 'Project: All Projects'. A green box highlights the 'Project: All Projects' dropdown. Below it is a search bar with 'Select project...' and a 'Show default projects' toggle switch. A list of projects follows, including 'All Projects', 'default', 'kube-node-lease', and 'kube-public'. At the bottom right, there are tabs for 'Community', 'Marketplace', and 'Community' again.

5. Scroll down and select openshift-marketplace.

This screenshot shows the same interface as above, but the 'openshift-marketplace' operator is now selected in the list. A green box highlights 'openshift-marketplace'. The interface includes a 'Create Project' button at the bottom left. The right side shows a grid of operator cards, with one card for 'Abot Operator-v3.0.0' highlighted by a green box.

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 search results for 'IBM Cert Manager'. The search bar at the top right contains 'IBM Cert Manager'. A green box highlights the search bar. Below it, a list of items shows 'ibm-cert-manager-4.2.7' selected, which is also highlighted with a green box. The item details show it's an 'IBM Cert Manager' operator provided by IBM, used 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, Workloads, Networking, Storage, Builds, Observe, and Compute. Under Operators, 'OperatorHub' is selected. In the main content area, a search bar at the top right says 'All Items' and has 'IBM Cert Manager' typed into it. Below the search is a list of operators, with 'ibm-cert-manager-4.2.7' by IBM highlighted. To the right of this list is a detailed view of the 'IBM Cert Manager' operator. It shows the 'Channel' as v4.2 and 'Version' as 4.2.7. Under 'Capability level', 'Basic Install' and 'Seamless Upgrades' are selected. The 'Source' is listed as 'ibm-cert-manager-4.2.7'. A large green 'Install' button is prominently displayed at the top of this detailed view.

8. Keep the default settings and click Install.

The screenshot shows the 'Install Operator' page for the 'IBM Cert Manager' operator. The left sidebar is identical to the previous screenshot. The main form has several fields: 'Update channel' set to 'v4.2', 'Version' set to '4.2.7', and 'Installation mode' set to 'All namespaces on the cluster (default)'. Under 'Installed Namespace', 'Operator recommended Namespace' is selected with 'ibm-cert-manager' chosen. The 'Namespace creation' section indicates that 'Namespace ibm-cert-manager does not exist and will be created'. The 'Update approval' section has 'Automatic' selected. At the bottom of the form are two buttons: a green 'Install' button and a blue 'Cancel' button. To the right of the form, there are four cards labeled 'Provided APIs': 'CR CertificateRequest', 'CMC Cert Manager Config', 'ClusterIssuer', and 'Issuer', each with a brief description and status.

**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 'andrewjones' at 'Andrews-MBP'. They run 'oc create namespace wxa4z-byos' which creates the namespace. Then they run 'export IBM_CS_ENT_KEY=' followed by the copied entitlement key. Finally, they run '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' which creates the secret 'icr-pull-secret'.

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'. Below the search bar, there's a list of operators categorized by provider: Community, Marketplace, and Red Hat. One operator from the Red Hat category is highlighted with a green border.

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. Reload the browser page if the operator is not listed.

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'. Below the search bar, there's a list of operators. One specific operator, 'IBM watsonx Assistant for Z Operator Catalog', is highlighted with a green border.

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

Install

Channel: stable

Version: 2.1.0

Capability level: Basic Install

- 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.

Installed Namespace*: wxa4z-byos

Update approval*:

- Automatic
- 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: '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. The text 'Installed operator: ready for use' is displayed below the card, followed by two buttons: 'View Operator' and 'View installed Operators in Namespace wxa4z-byos'. A green dashed box surrounds the 'Installed operator: ready for use' message and the operator card.

- 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 %
```

- 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`, paste the following text in the file, and then modify the default password.

File name:

```
os-secret.yaml
```

Substitute a secure password of your choosing for the string <OPENSEARCH_PASSWORD>. Record this value for later use.

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`, paste the following text in the file, and then modify the default password.

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. Record this value for later use.

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`, paste the following text in the file, and then modify the default password.

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. Record this value for later use. 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	da80909aab1563ff0c15b7c6d81 a8bbefabaa605065bb6c6b490e87 0e87f0a96f	-	-	Nov 6, 2024, 9:26 AM
ibm-wxa4z-operator-catalog-cfvsv	Running	1/1	0	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	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	wxa4z-byos-cluster	1,095.8 MiB	0.017 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-1	Running	1/1	0	wxa4z-byos-cluster	1,169.1 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-2	Running	1/1	0	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	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	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	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	da80909aab1563ff0c15b7c6d81 a8bbefabaa605065bb6c6b490e87 0e87f0a96f	-	-	Nov 6, 2024, 9:26 AM
ibm-wxa4z-operator-catalog-cfvsv	Running	1/1	0	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	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	wxa4z-byos-cluster	1,095.8 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-1	Running	1/1	0	wxa4z-byos-cluster	1,169.1 MiB	0.016 cores	Nov 6, 2024, 10:22 AM
wxa4z-byos-cluster-2	Running	1/1	0	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	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	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	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.

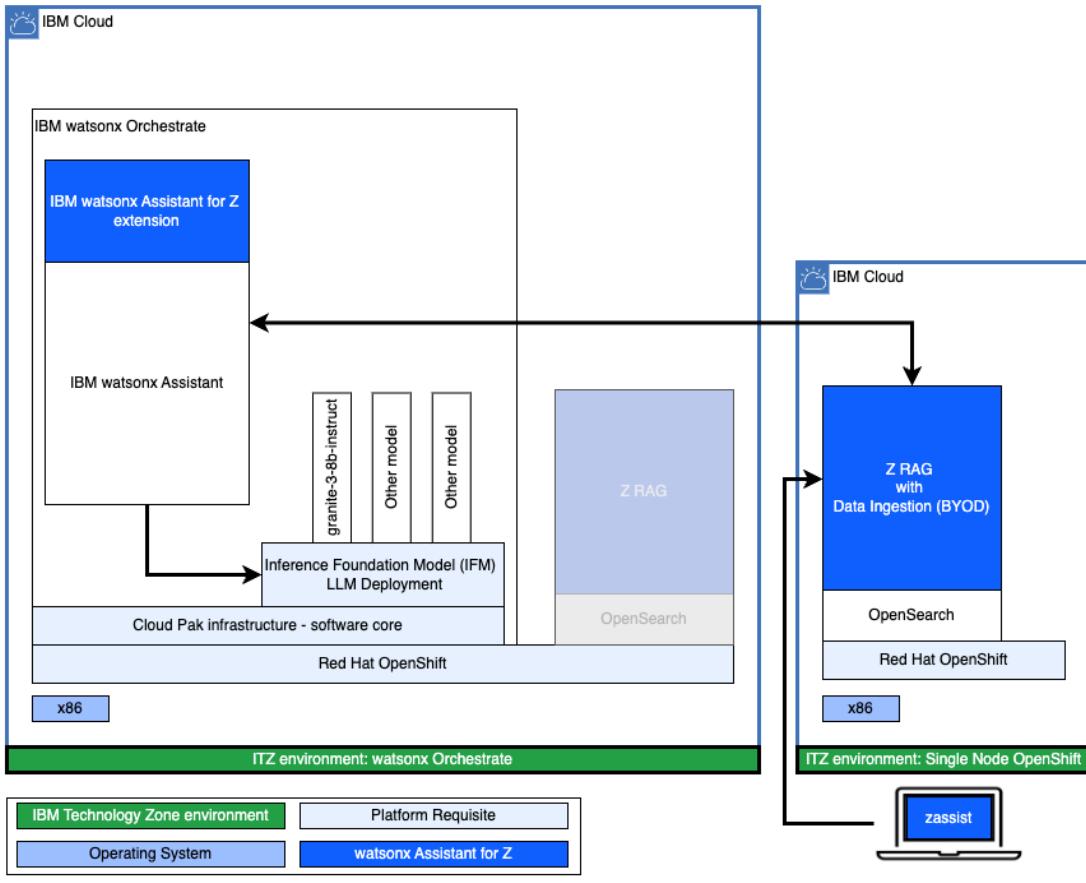
As an example, a client mentioned that their developers often need reference material on company-specific legacy code or company-specific syntax. The users must search through volumes of documentation to find it or look at old code. Also, a need for their operational support group to quickly determine 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 named "zassist.zip" in a Box folder. The file was modified on Sep 5 at 9:15 AM and has a size of ... bytes.

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

The screenshot shows the contents of the "zassist" folder. It contains subfolders for "linux", "mac", "mac-arm", and "windows", and a file named "zassist.zip". The "mac" folder is highlighted with a green dashed box, and the "zassist" file within it is also highlighted with a green box.

Name	Date Modified	Size	Kind
byos.yaml	Today at 12:17 PM	956 bytes	YAML
ca.crt	Today at 11:36 AM	Zero bytes	certificate
catalogCertManager.yaml	Today at 9:15 AM	476 bytes	YAML
catalogSource.yaml	Today at 9:25 AM	397 bytes	YAML
client-ingestion-secret.yaml	Today at 10:05 AM	155 bytes	YAML
os-secret.yaml	Today at 10:04 AM	149 bytes	YAML
zassist	Sep 5, 2024 at 11:15 AM	--	Folder
linux	Sep 18, 2024 at 8:15 AM	--	Folder
mac	Sep 18, 2024 at 8:15 AM	--	Folder
zassist	Sep 18, 2024 at 8:17 AM	10.2 MB	Unix Executable File
mac-arm	Sep 18, 2024 at 8:15 AM	--	Folder
windows	Sep 18, 2024 at 8:15 AM	--	Folder
zassist.zip	Today at 1:47 PM	23 MB	ZIP archive

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](#).



Windows users may need to rename the file **zassist** file!

If the **zassist** file does not execute properly, rename the file to **zassist.exe**.

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

```
zassist
andrewjones@Andrews-MacBook-Pro watsonxAssistant % PATH=$PATH:./zassist
andrewjones@Andrews-MacBook-Pro watsonxAssistant % ./zassist
zassist: error: expected one of "version", "init", "login", "ingest", "load", ...
andrewjones@Andrews-MacBook-Pro watsonxAssistant %
```



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.

Mac OS:

```
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.

Windows OS (this method can also be used by Mac OS users):

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.

Name	Status	Location	Service
wxa4z-client-ingestion	Accepted	https://wxa4z-client-ingestion-wxa4z-byos.apps.67850b0240c621133ba8749.ocp.techzone.ibm.com	wxa4z-client-ingestion
wxa4z-opensearch-wrapper	Accepted	https://wxa4z-opensearch-wrapper-wxa4z-byos.apps.67850b0240c621133ba8749.ocp.techzone.ibm.com	wxa4z-opensearch-wrapper

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 IBM Watsonx Orchestrate AI assistant builder interface. The top navigation bar includes 'IBM Watsonx Orchestrate', 'AI assistant builder', and 'Zeeves'. Below the navigation is a toolbar with 'Copy link to share', 'Change layout', 'Change background', and 'Customize web chat'. On the left, there's a sidebar with various icons. The main area is titled 'Preview assistant' and shows 'Zeeves'. A blue circular icon of an owl is displayed. Below it, the text 'Hi! I'm Zeeves. How can I help you today?' is shown. At the bottom, a text input field contains the prompt: 'The customer application is failing with ERR-CBL-001, what does this internal error mean?'. To the right of the input field is a blue send button with a white arrow.

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

The customer application is failing with 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.

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

CDBL0001E
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...

Mainframe_COBOL_Error_Codes....
ERR-CBL-001: Null value encountered in mandatory field.



Take note of the order of the response citations!

4. 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.

Company-Specific Mainframe COBOL Error Codes

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: SOCA 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.

5. 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 AI assistant builder. The 'Metadata' field is highlighted with a green dashed box. The field contains a JSON example:

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

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](#).

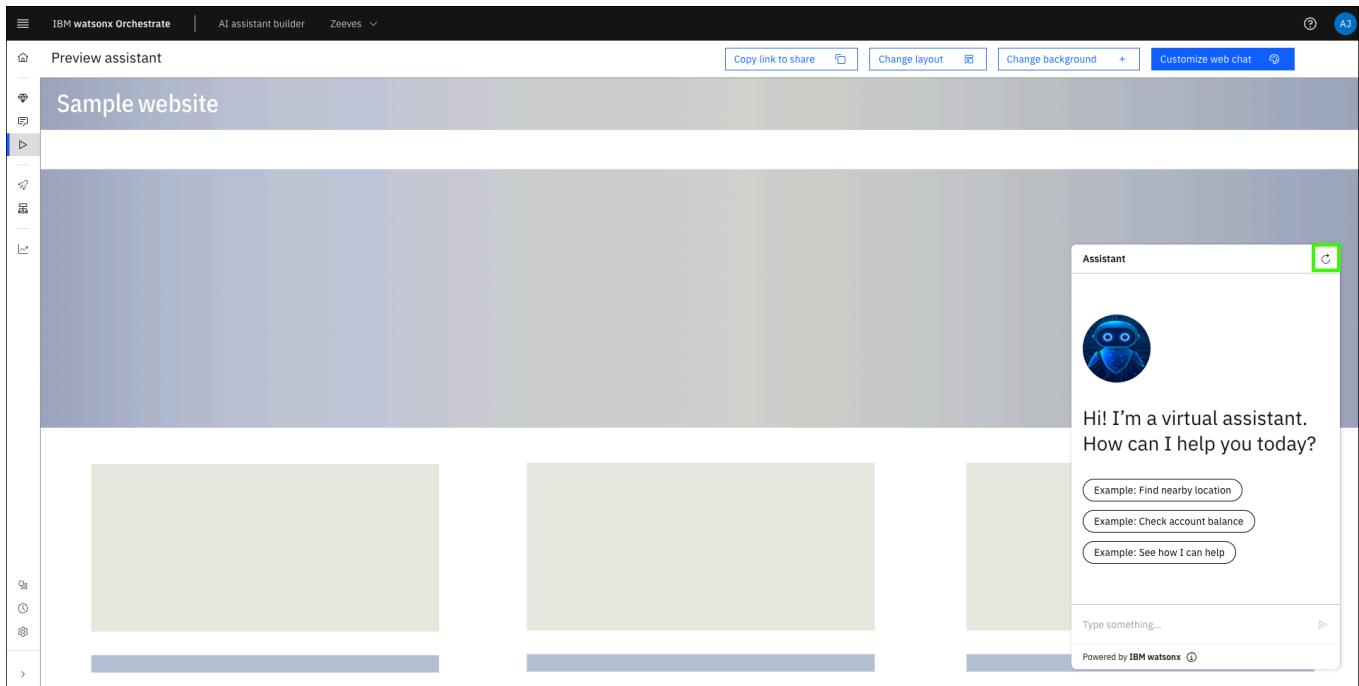
1. Set the (a) **Metadata** field for your BYOS custom search instance to the following value, click (b) **Save**, and then click (c) **Close**. Notice the weight for **customer_docs** is heavier than the weight for **product_docs**.

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

The screenshot shows the 'Custom service' configuration page in the IBM Watsonx Orchestrate interface. The 'Metadata' field is highlighted with a green box and labeled 'a'. The 'Save' and 'Close' buttons are labeled 'b' and 'c' respectively. The 'Metadata' field contains the JSON code from the previous code block.

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
```

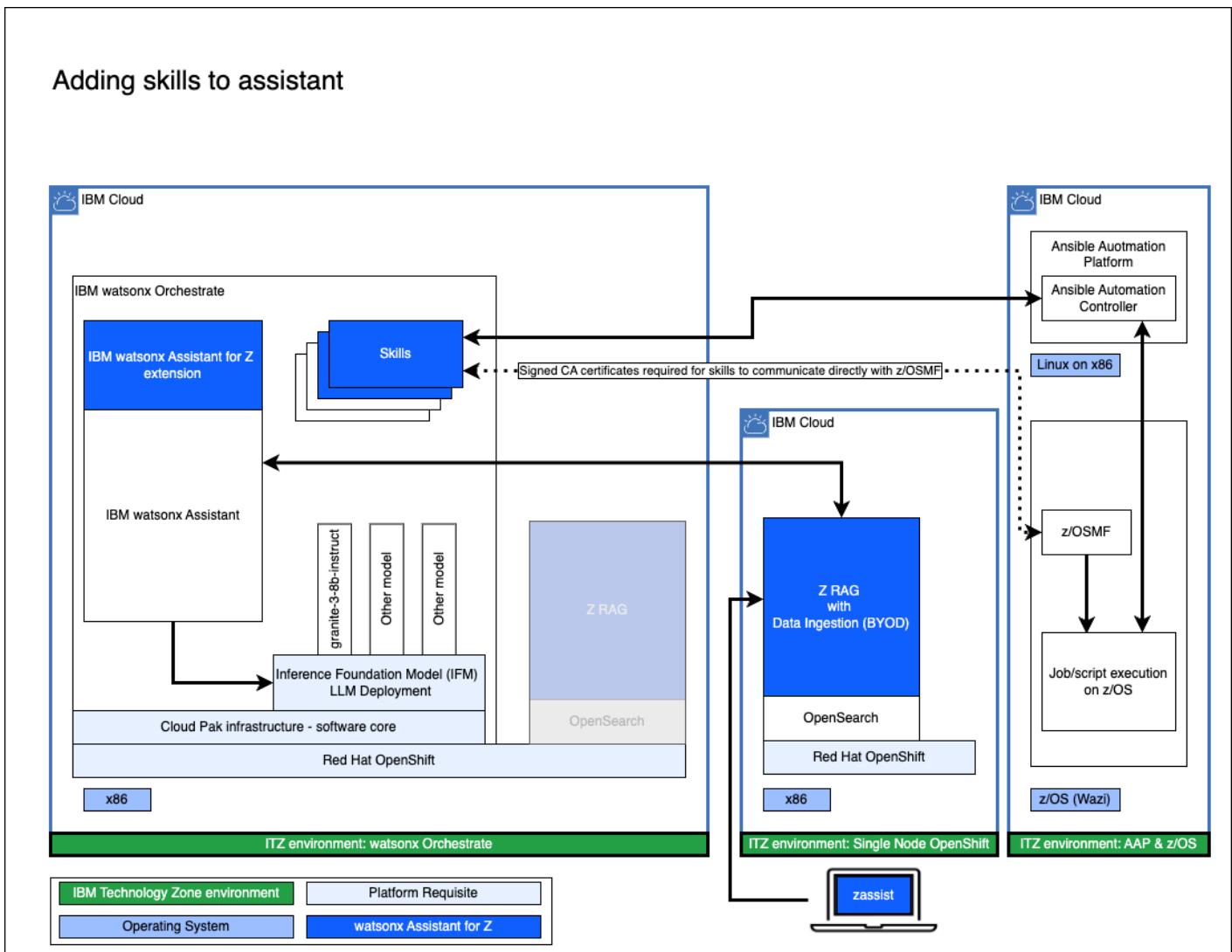
Skills and skill flows

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.

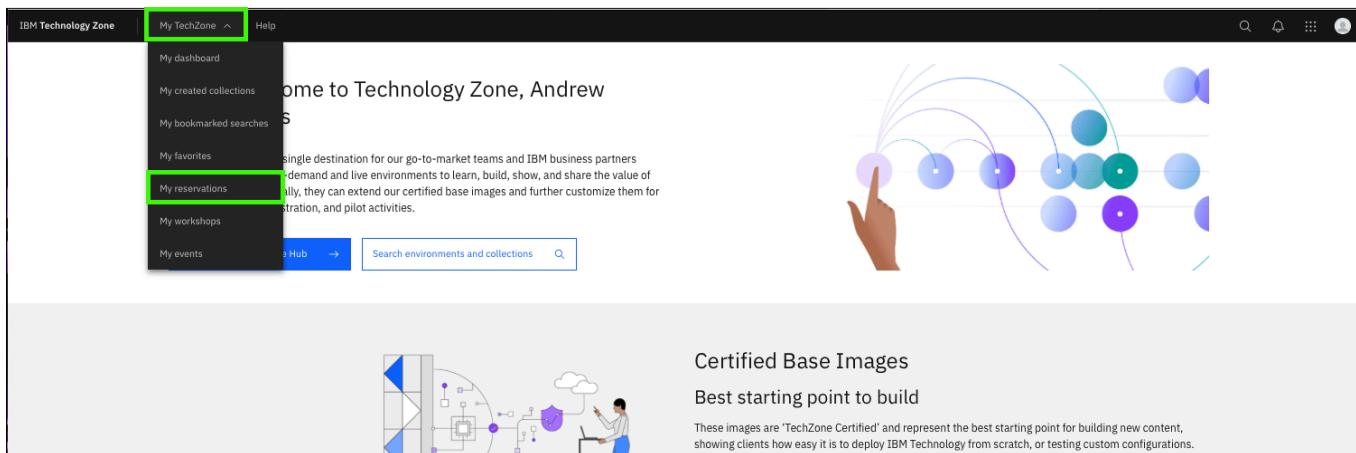
Access the AAP and Wazi as a Service environment

Be sure to record the information as instructed

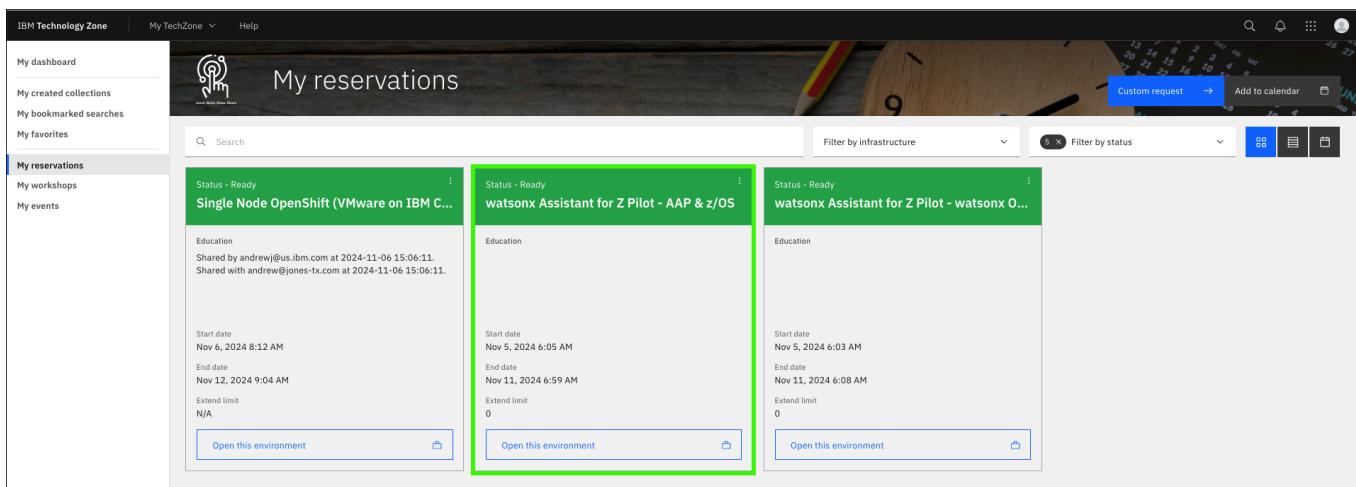
Several of the steps below instruct you to record values from your ITZ reservation. Be sure to do this as they will not only be used in this section, but also in later sections of the lab guide.

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

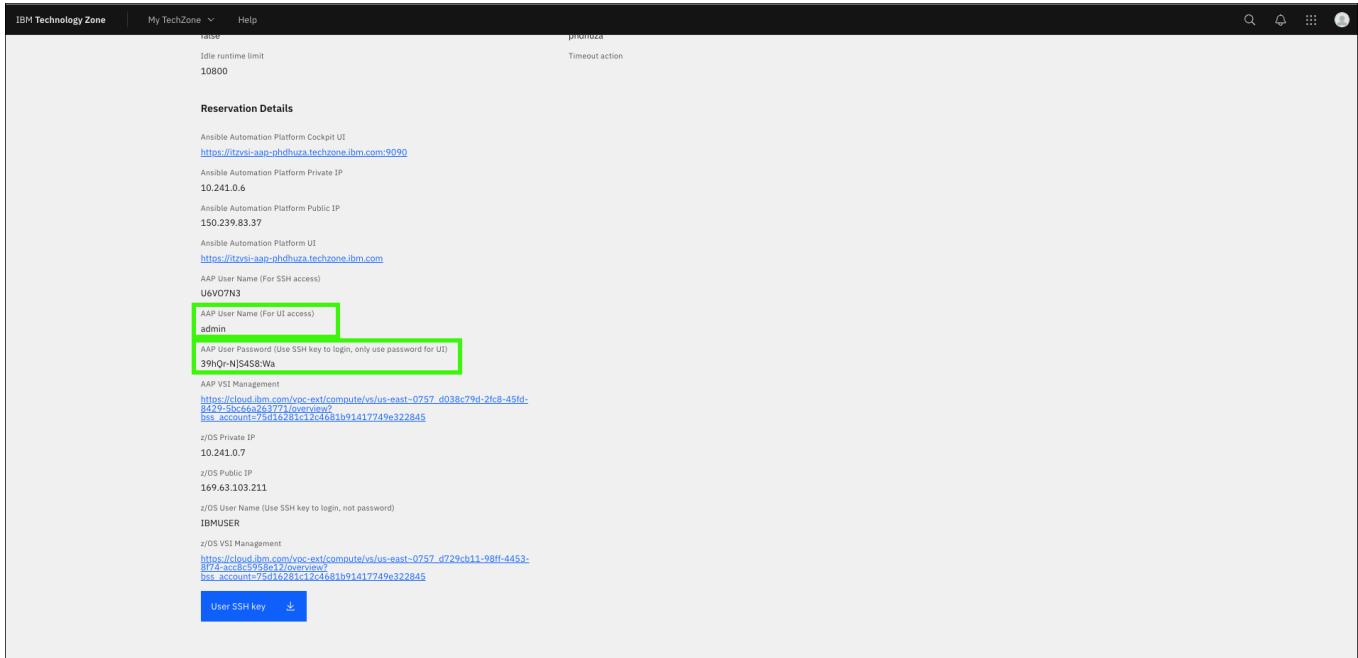
[ITZ My reservations](#)



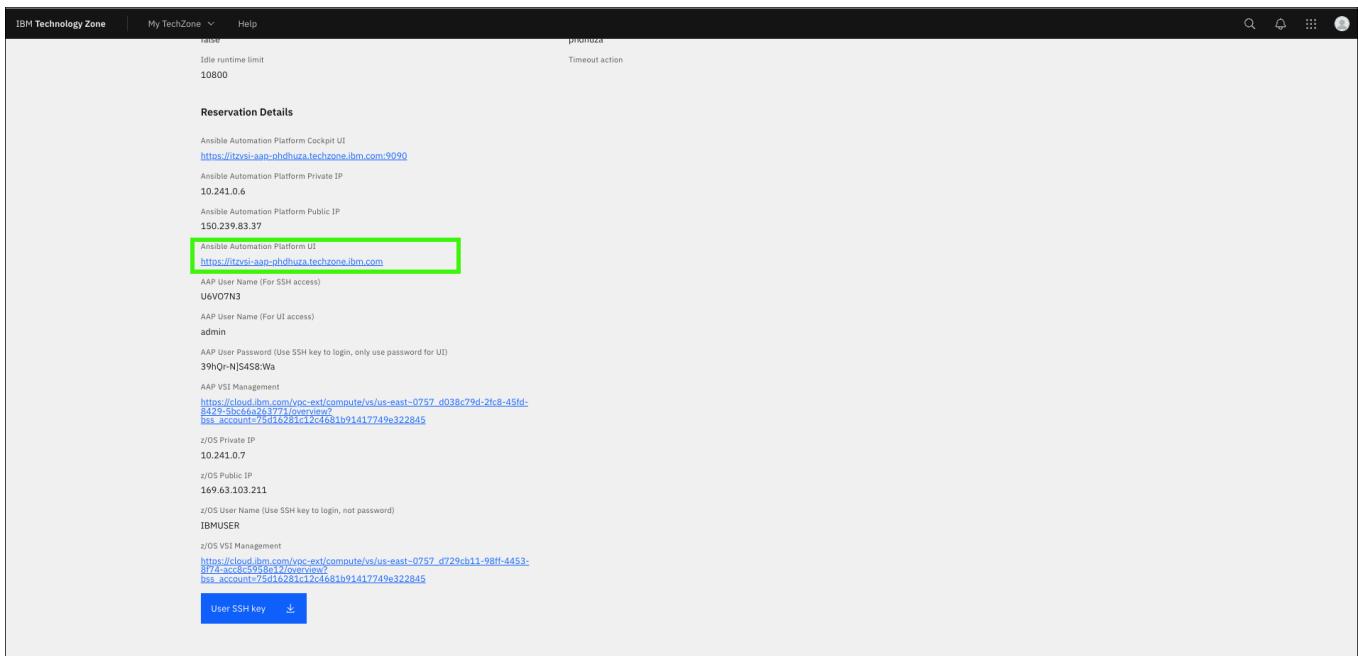
2. Click the **watsonx Assistant for Z Pilot - AAP & z/OS** tile.



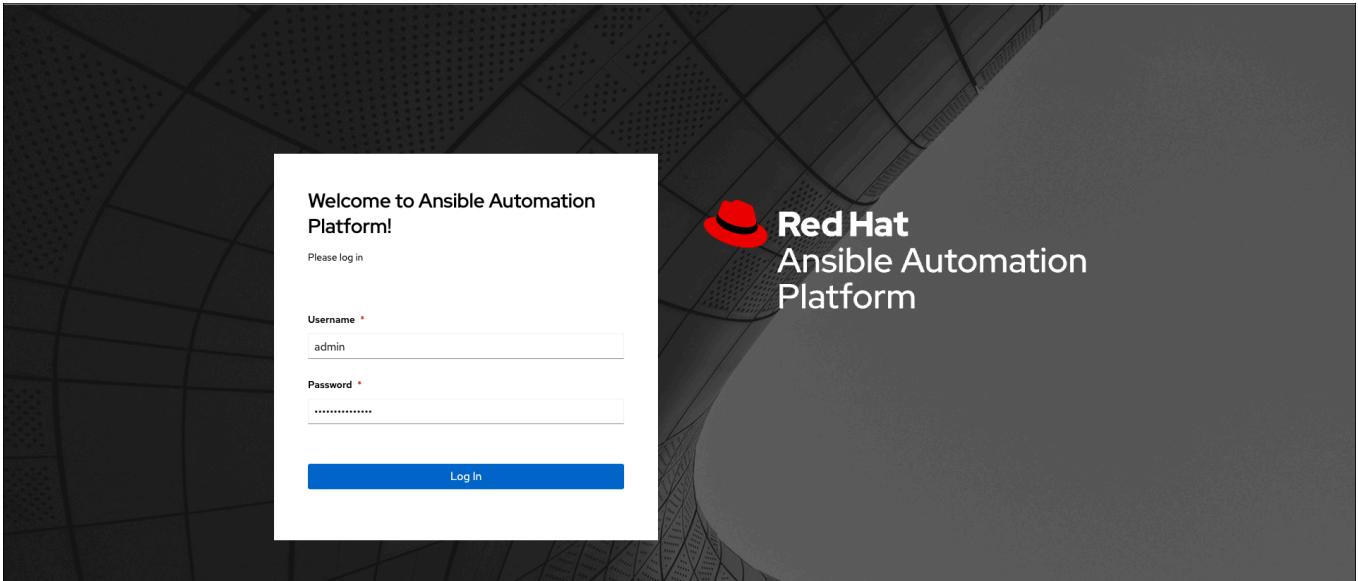
3. Locate and record the **AAP User Name (For UI access)** and **AAP User Password** fields.



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.

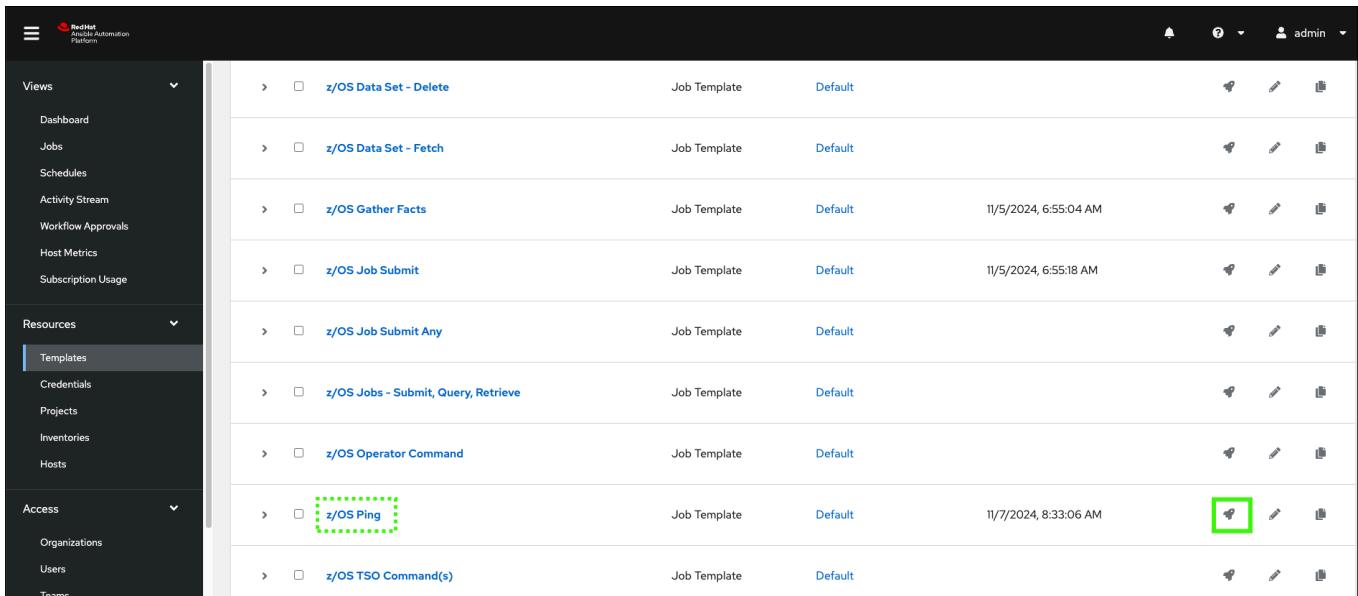


6. Click **Templates** under the **Resources** section.

The AAP instance is preconfigured to the Wazi aaS instance

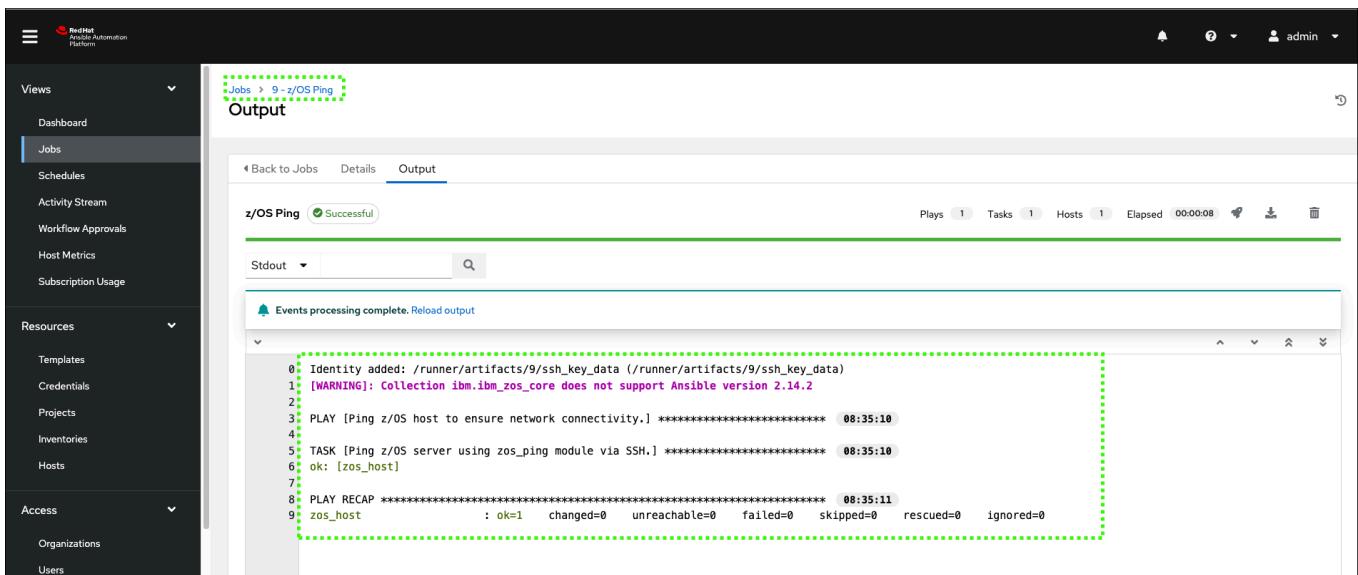
Note that because the AAP instance and the back-end z/OS system are preconfigured, no changes are needed to execute the templates and they will target your provisioned z/OS system by default.

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](#).

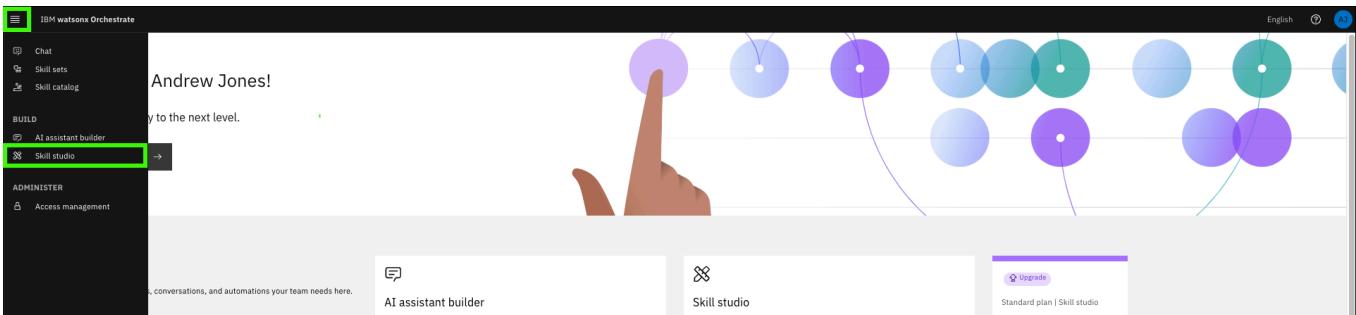
Importing skills from Ansible Automation Platform

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.

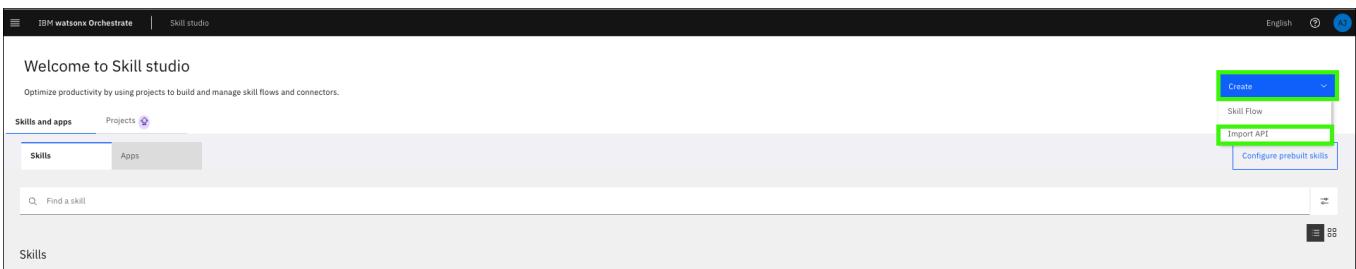
Import skills into your assistant

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](#).

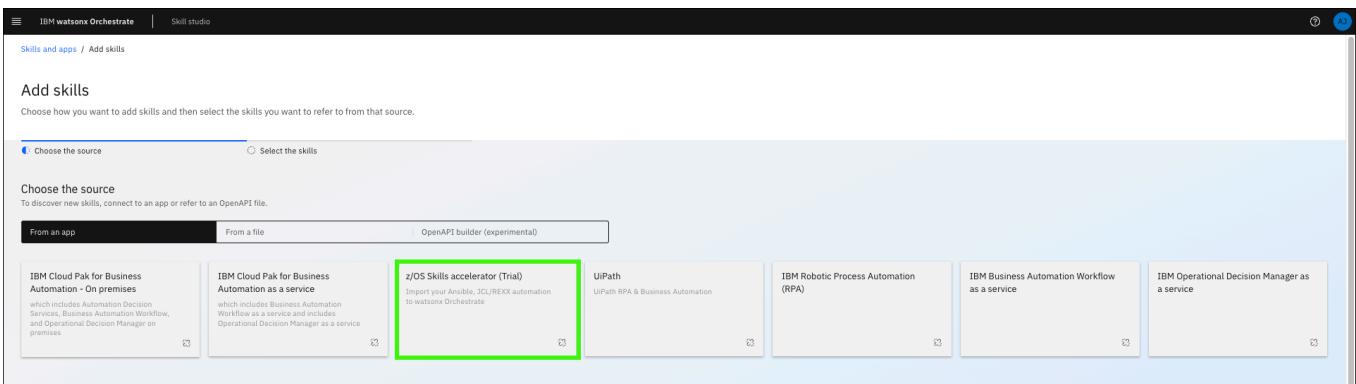
1. Return to your watsonx Orchestrate instance and expand the main menu and click **Skill studio**.



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

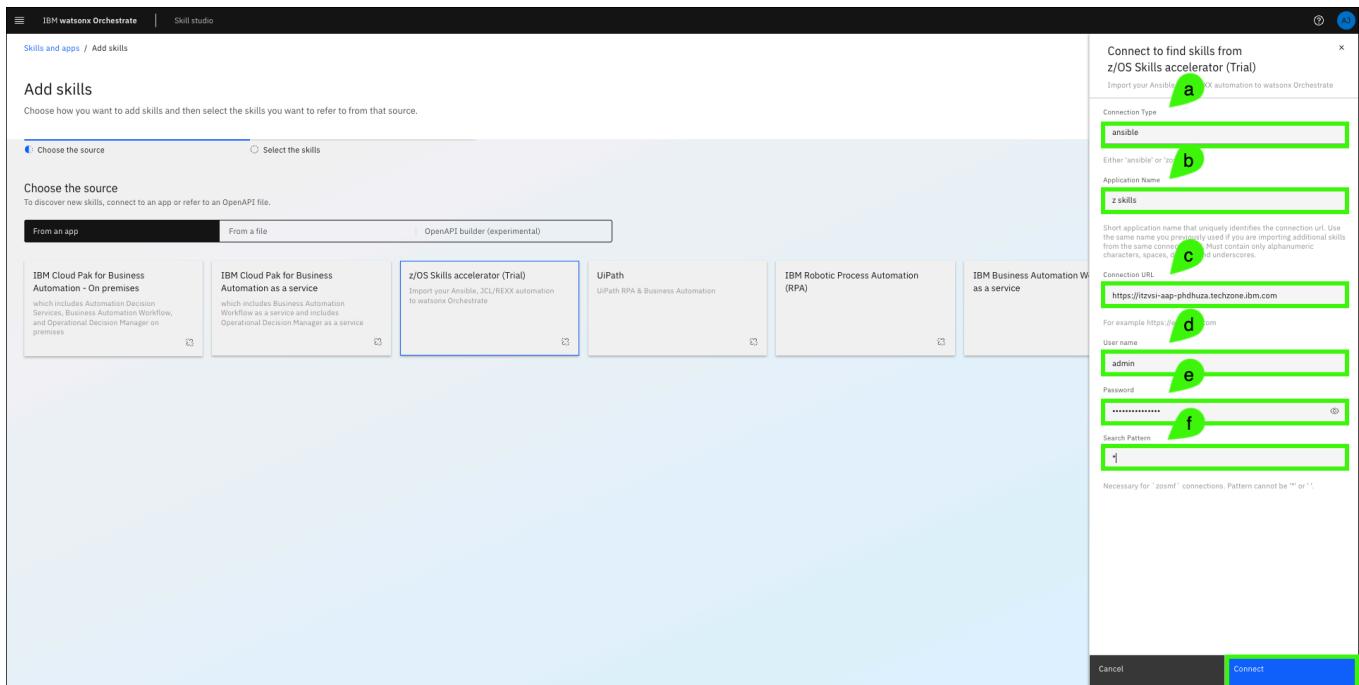
b: Application Name: <enter a meaningful name for the skills that you will import> - record this name, you will need in the next section

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: *

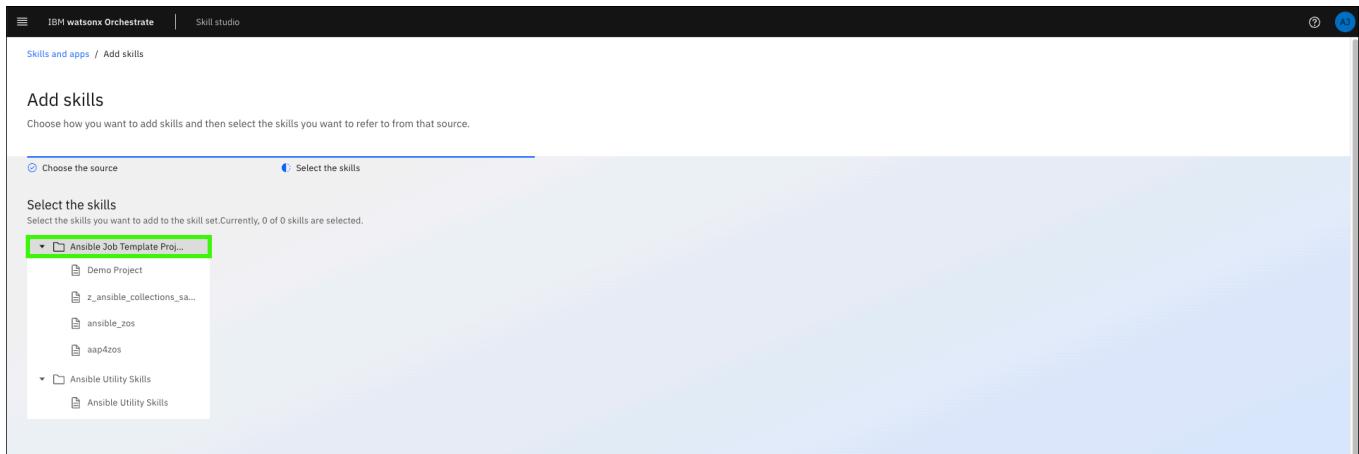


5. Expand the Ansible Job Template Proj... folder.



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

Enhance this skill

Export this skill

Delete this skill

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

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

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* 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

App

API version* 1.0

Preview

The skill will look like this in the catalog.

The skill will look like this in the skill set.

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. The left sidebar has sections for Chat, Skill sets (which is highlighted with a green box), and Skill catalog. The main area is titled "Skill studio" and contains a table of 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 top navigation bar includes "Skill sets" and "Team Skills". The "Team Skills" dropdown is open, showing "Team Skills" and "Zeeves draft" (which is highlighted with a green box). Below the dropdown, there is a note about team skills and a "Manage skills" button. The main content area shows a table with columns "Name" and "App".

3. Click the **Connections** tab.

The screenshot shows the "Zeeves draft" connections page. The top navigation bar includes "Skill sets" and "Connections" (which is highlighted with a green box). The main content area shows a table with columns "Name" and "App".

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 | ⌂ ⌃

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 other buttons. Below it, the main area has a title 'Zeeves draft' and a 'Skill sets' tab selected. Underneath, there's a table showing skill details: 'Ansible Controller Skills - z skills' (2 skills, Team credential type, connected by 'andrew@jones-tx.com'). On the right side, there's a sidebar with options 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 corner.

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

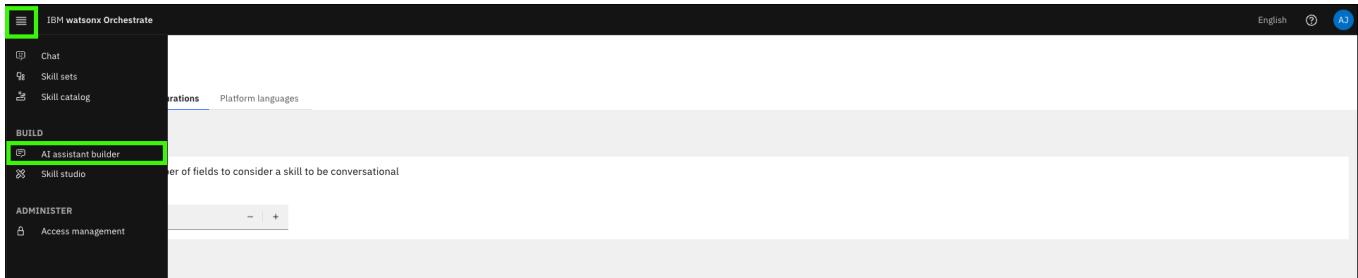
This screenshot shows the 'Settings' page in the IBM Watsonx Orchestrate interface. The 'Skill configurations' tab is active, highlighted with a green box. Other tabs available are 'Data Retention' and 'Platform languages'. The 'Skill configurations' section contains a note: 'Set default for maximum 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**.

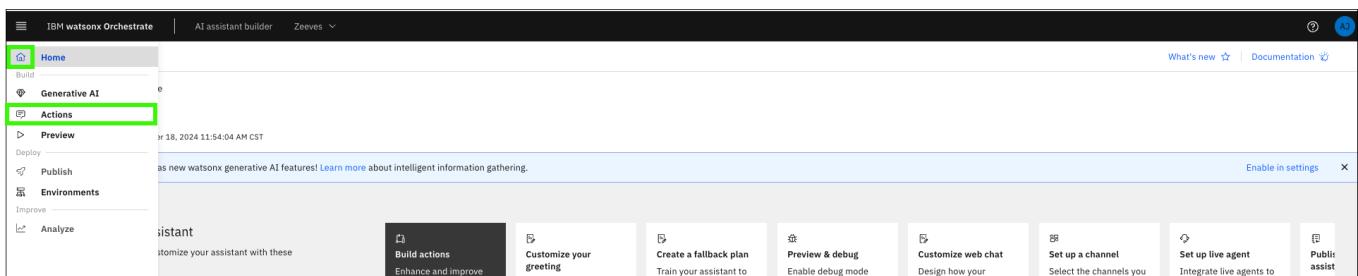
This screenshot is identical to the previous one, showing the 'Settings' page with the 'Skill configurations' tab selected. The 'Number of form fields' input field now contains the value '0', indicating the configuration has been updated.

Create actions

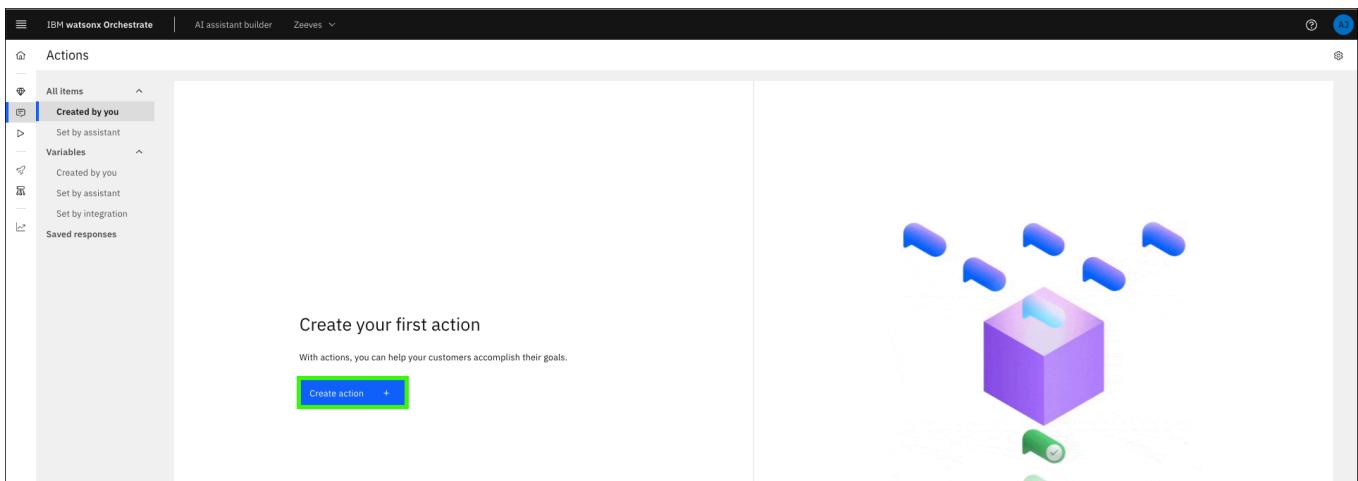
1. Click the main menu and select **AI assistant builder**.



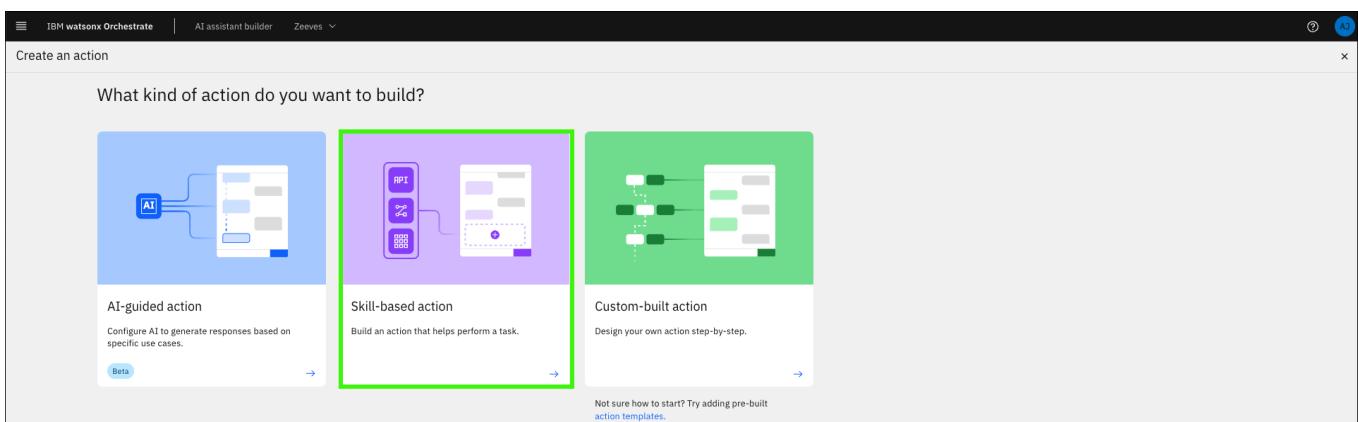
2. Hover over the **Home** icon () and click **Actions**.



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 skills under the heading "Build an action from a skill". The "Select a skill" section says "Choose a conversational skill published as a foundation of your action." A search bar is present. Below is a grid of cards:

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**.

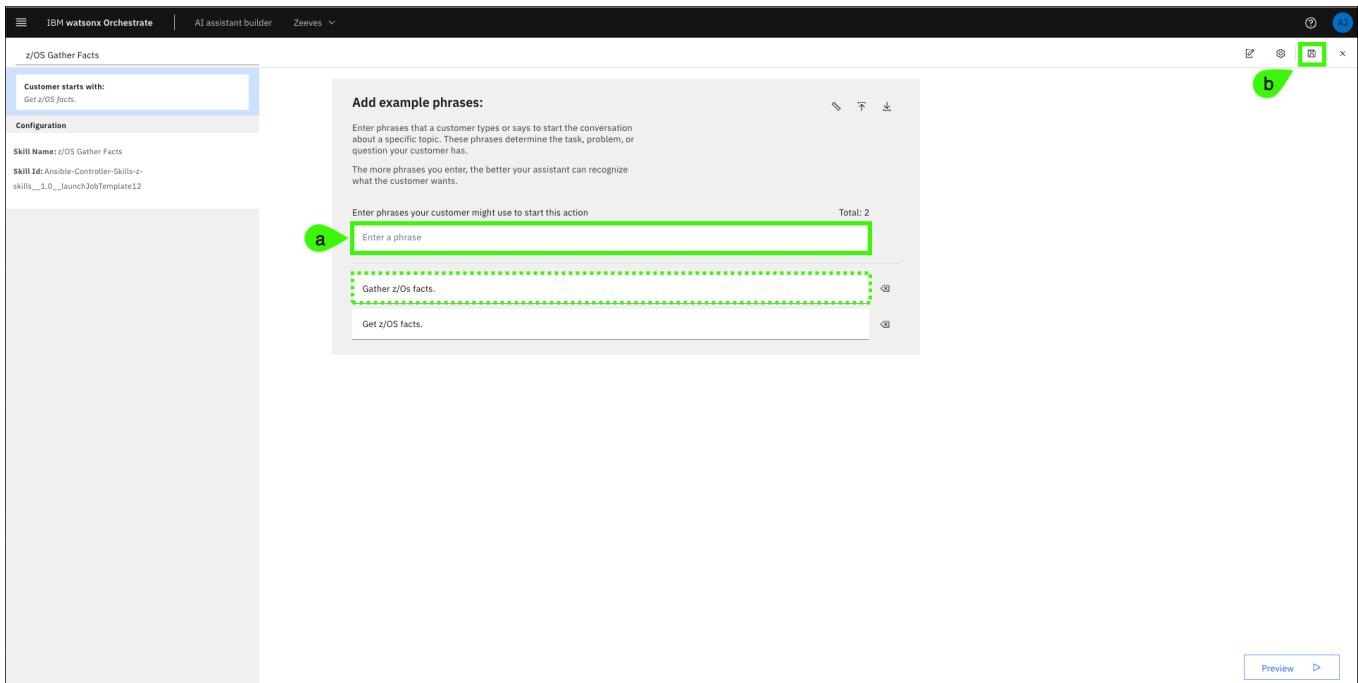
Sample prompts:

Get z/OS facts

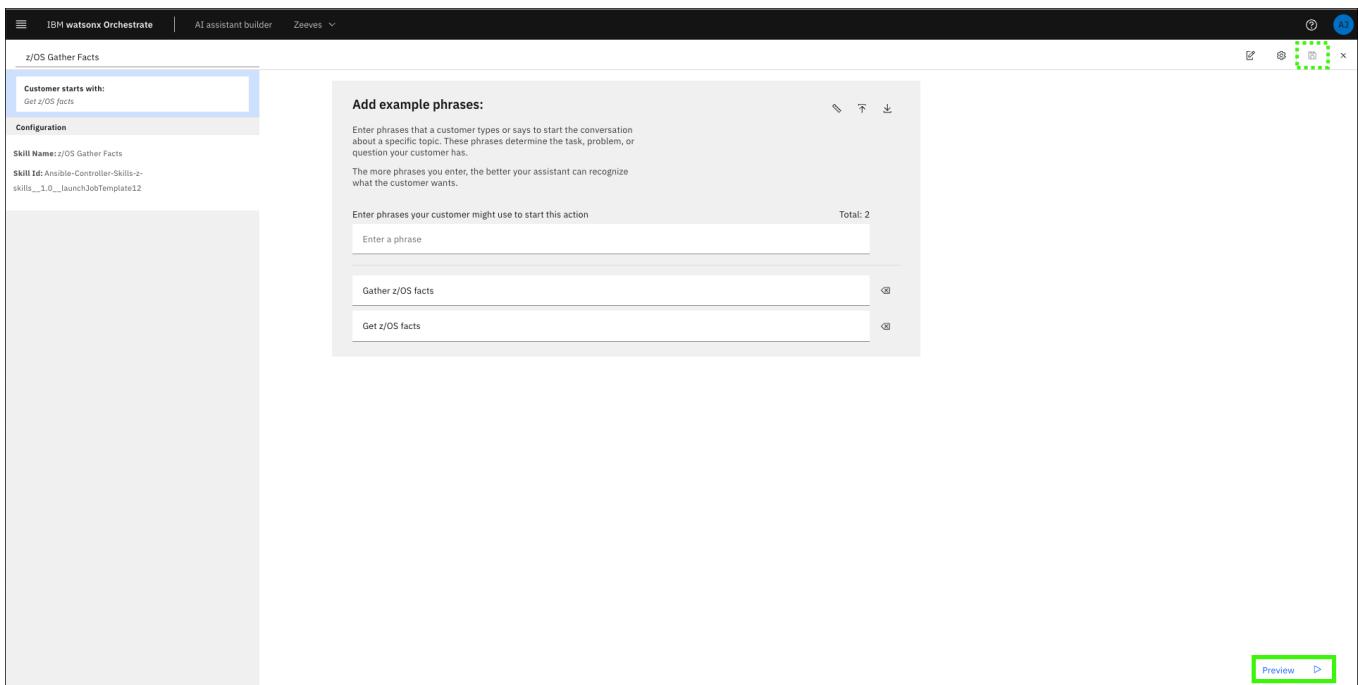
Gather z/OS facts

The screenshot shows the configuration for the "z/OS Gather Facts" skill. On the left, there's a sidebar with "Customer starts with:" containing "Example: I want to pay my credit card bill," and a "Configuration" section with "Skill Name: z/OS Gather Facts" and "Skill ID: Ansible-Controller-Skills-z-skills_1.0_launchJobTemplate12". The main area has a heading "Add example phrases:" with instructions: "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." Below this is a text input field with "Enter phrases your customer might use to start this action" and "Total: 0". The input field contains "Example: I want to pay my credit card bill.". A modal window titled "New action" is overlaid. It asks "What does your customer say to start this interaction?" and has a text input field containing "Get z/OS facts." A green circle labeled "a" is positioned above the input field in the modal. A blue circle labeled "b" is positioned above the "Save" button in the modal.

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



8. Click Preview.



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

Prompt:

Get z/OS facts

The screenshot shows the configuration interface for the 'z/OS Gather Facts' skill. In the 'Customer starts with:' field, 'Get z/OS facts' is entered. The 'Add example phrases:' section contains two entries: 'Gather z/OS facts' and 'Get z/OS facts'. On the right, a 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 same configuration interface as before, but now with a red box highlighting the error message in the preview window: 'You 1:04 PM Get z/OS facts'. Below this, another message from the bot is shown: '1:05 PM Conversational skill called z/OS Gather Facts recognized job: 12 status: pending'. A note at the bottom states: 'There are no additional steps for this action. Add a new step or end the action.' A green box highlights the user's message 'Get z/OS facts'.

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.

```

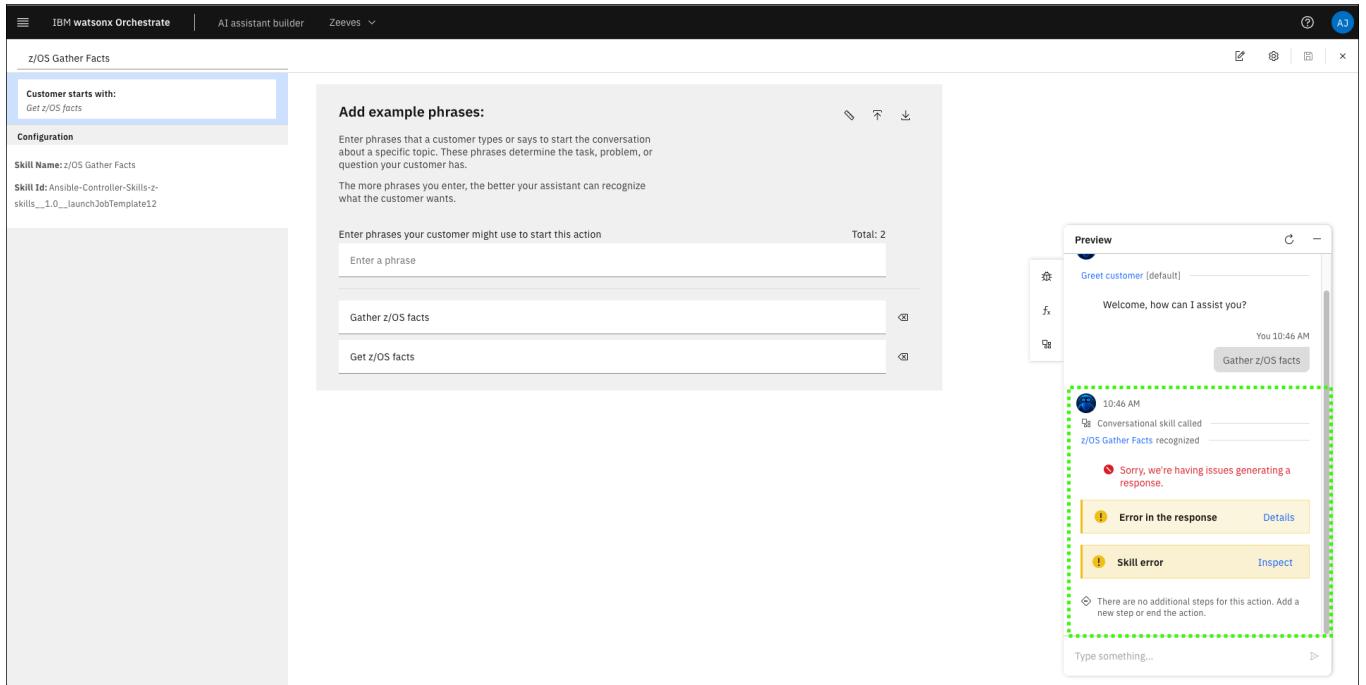
25 }
26 TASK [Print out all gathered facts about the z/OS host.] **** 07:17:38
27 ok: [zos_host] => {
28     "ansible_facts": {
29         "arch_level": "2",
30         "cpc_nd_manufacturer": "IBM",
31         "cpc_nd_model": "A00",
32         "cpc_nd_plant": "C1",
33         "cpc_nd_seqno": "237701828347",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

```

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

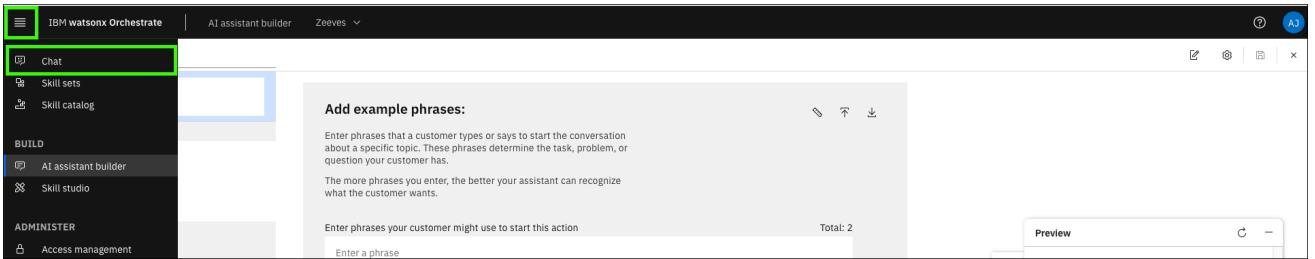
 Skill returns "Sorry, we're having issues generating a response" or the action is not performed and only search results are returned.



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", with a note "Total: 2". 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". A message from the bot says "10:46 AM Conversational skill called z/OS Gather Facts recognized". Below this, two error messages are highlighted with red circles and dashed green boxes: "Sorry, we're having issues generating a response." and "Error in the response". Another message "Skill error" is also visible. A note at the bottom of the preview window states "There are no additional steps for this action. Add a new step or end the action."

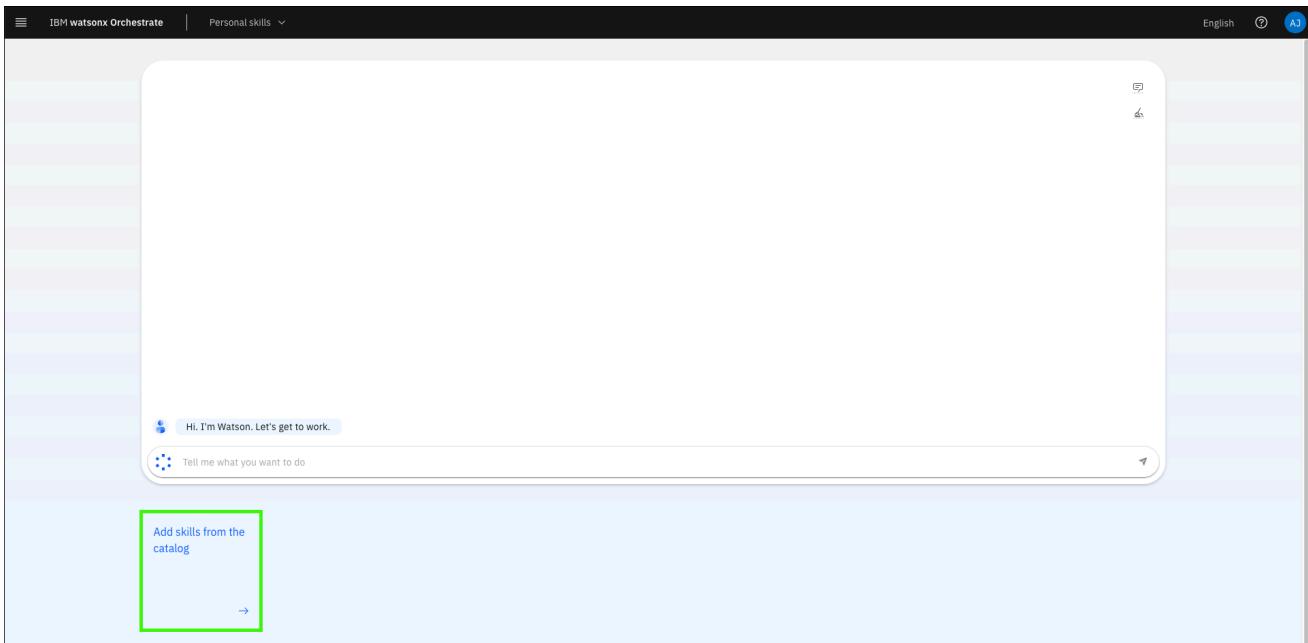
This error appears to be an intermittent issue when a skill is first added. To resolve, add the skill to your personal skills catalog using the steps that follow. If you encounter the issue, try the steps that follow:

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:" and "Enter phrases your customer might use to start this action" with "Total: 2".

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. At the top, it says "Skills are grouped by app. Select an app to see all the skills that use that app." Below is a search bar with the text "z skills". The main area is titled "Apps" and shows several skill categories in a grid:

- Salessoft: 47 skills
- Cognos: 8 skills
- Skill flows: 167 skills
- Adobe Workfront: 37 skills
- Webex: 14 skills
- IBM Process Mining: 2 skills
- Salesforce Chatter: 5 skills
- Ansible Controller Skills - z skills: 2 skills** (This tile is highlighted with a green box.)
- Interview top candidates using ...: 2 skills
- Reveal your existing applicants...: 7 skills
- Seismic: 22 skills
- FreshService: 25 skills
- Workday HCM: 36 skills
- ZoomInfo: 26 skills

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. At the top, it says "(2)". Below is a search bar with the text "Search skills". The main area shows two skill cards:

- z/OS Gather Facts**: "z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS-...". It has an "Add skill +" button highlighted with a green box.
- z/OS Ping**: "z skills - This playbook pings the z/OS host to test connectivity.". It also 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 z/OS 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 settings for "English" and a blue "Aj" button. 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 green button labeled "Gather z/OS facts" is highlighted. A green dashed box encloses the results of this skill: "z/OS Gather Facts", "status: pending", and "job: 12". At the bottom, there's a blue button to "Tell me what you want to do". To the left, a box says "Add skills from the catalog" with an arrow pointing to another box labeled "Ansible Controller Skills - z skills" which contains "2 skills".

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

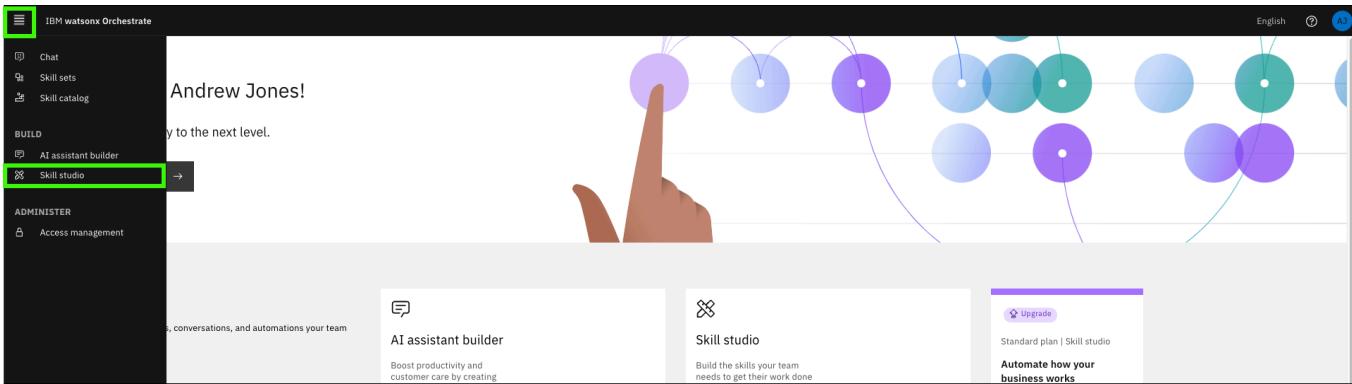
Creating skill flows

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](#).

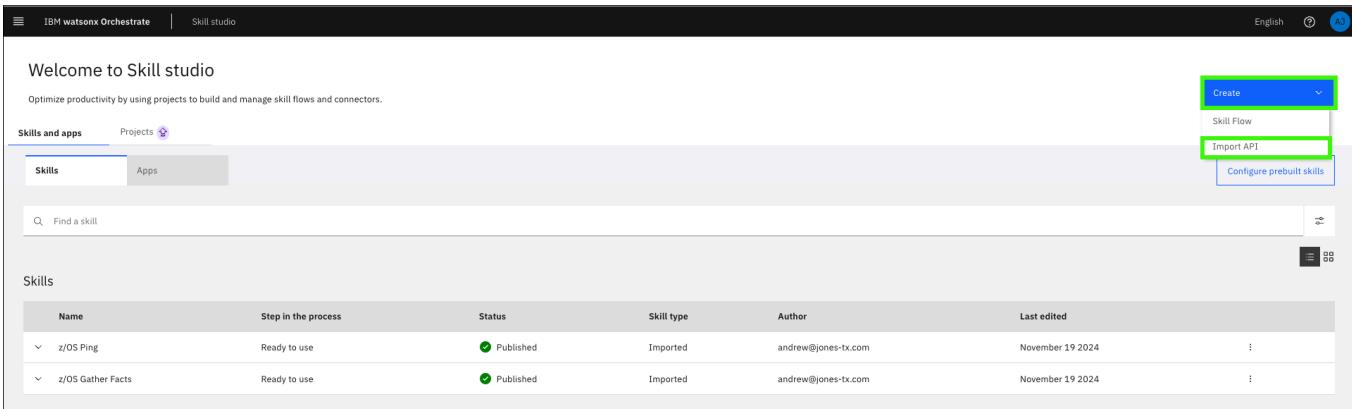
As mentioned in a previous section, default utility skills that are provided with the watsonx Assistant for Z skills collection. The **Retrieve job output** utility skill is used to return the output of a skill.

Add the utility skill

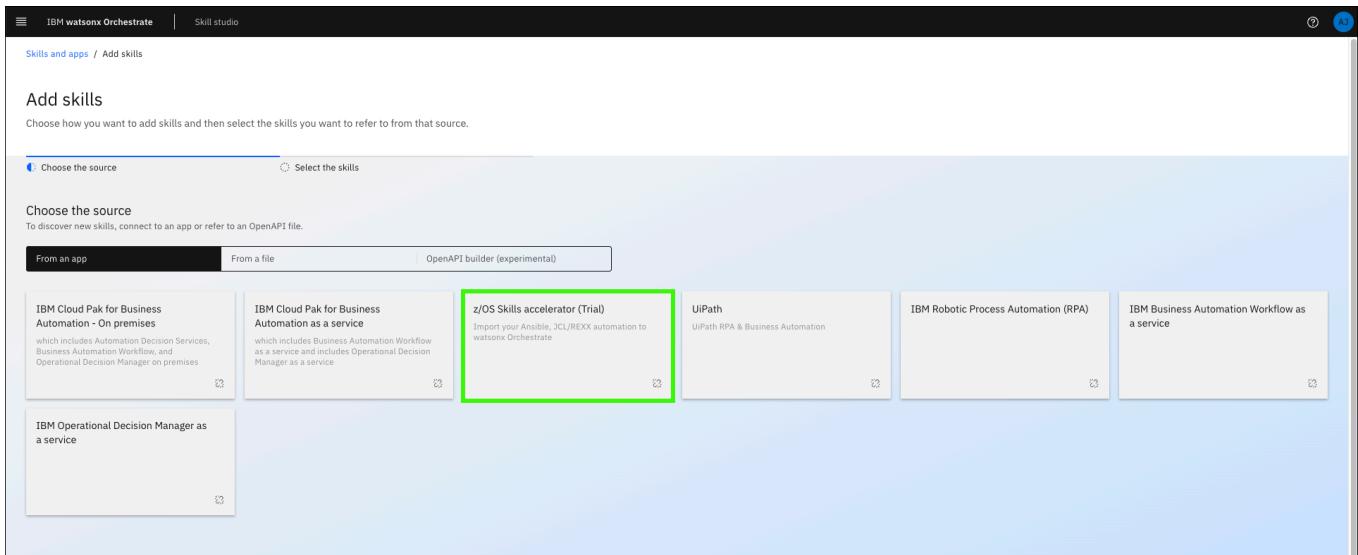
1. Open IBM watsonx Orchestrate **Skill studio**.



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`

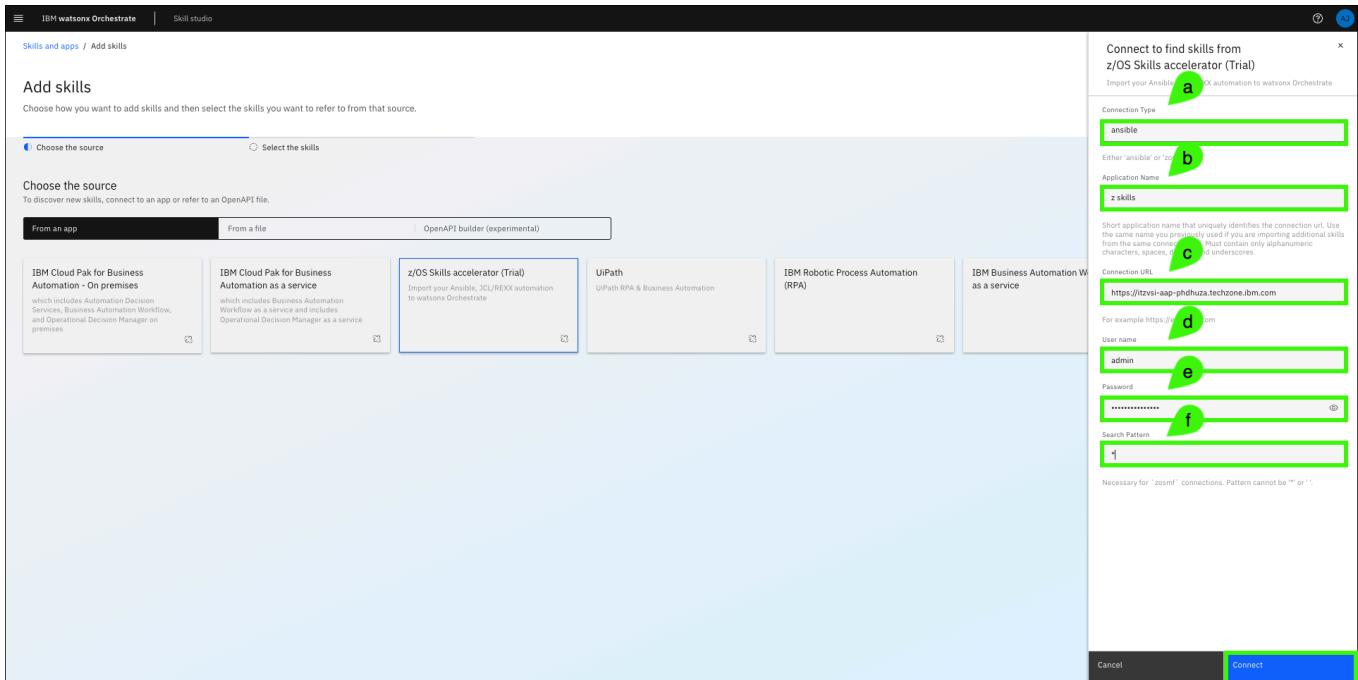
b: Application Name: <use the same application name from the previous section>

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

Description: z skills - Retrieve job output by job Id

API version*: 1.0

Categories: Add categories

App: Ansible Controller Skills - z skills

Publish

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	-	⋮

Items per page: 5 | 1-5 of 78 items

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	a ⋮ b

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)**.



Connect the application if it is not connected.

Use the AAP user name (admin) and the AAP password for your ITZ reservation.

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

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

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.

a Personal skills

a 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	Ansible Controller Skills - z ... 3 skills
Cognos 8 skills	Salessoft 47 skills				

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

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

Connect app

Personal skills

Search skills

Ansible Controller Skills - z skills

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

Create the skill flow

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

IBM Watsonx Orchestrate

English

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

Connect app

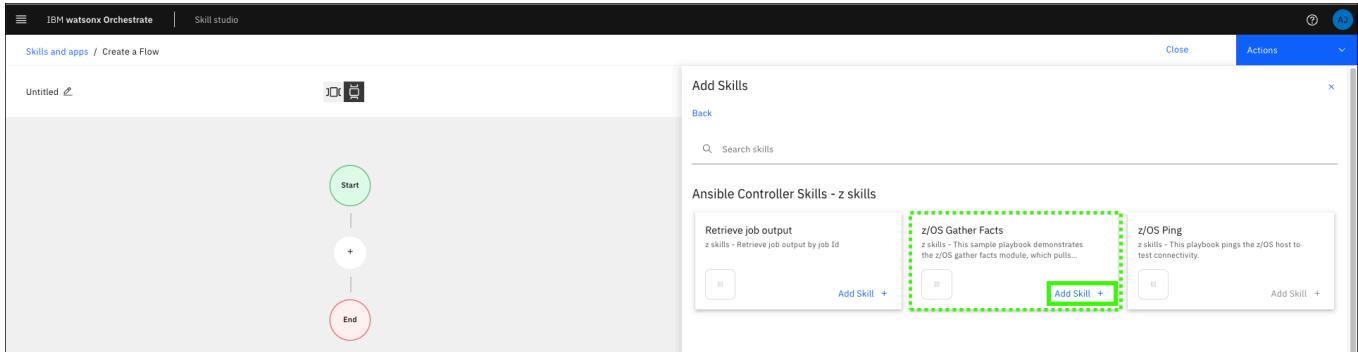
- Expand the **Create** drop-down menu and click **Skill flow**.

3. Click the + icon.

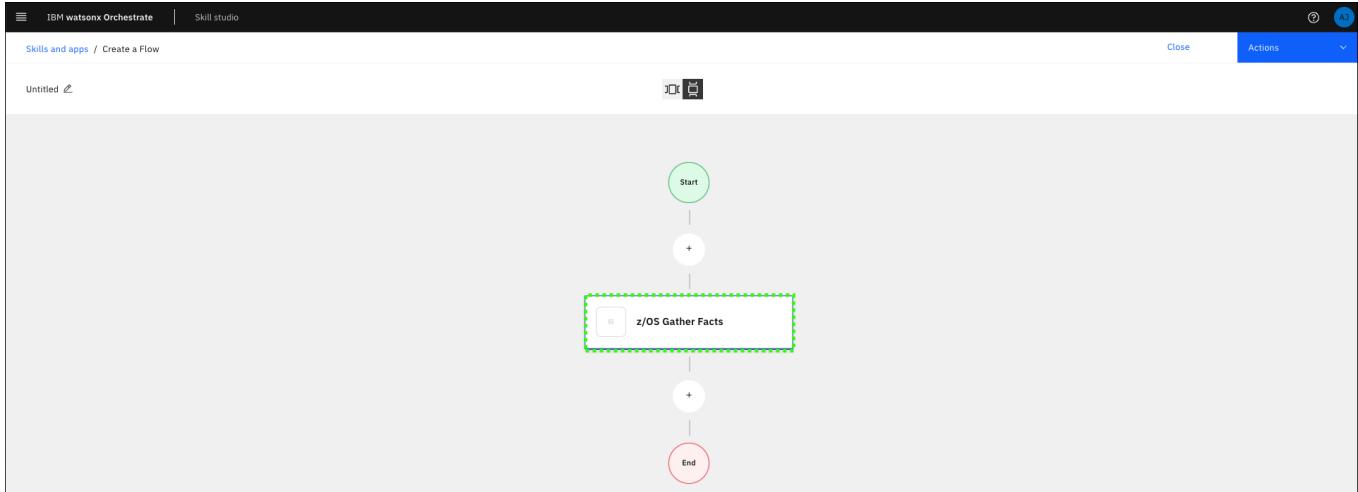
Next, you need to add the [z/OS Gather Facts](#) skill and the **Retrieve job output** skill to the skill flow. Use the **Search apps** function to locate the skills.

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

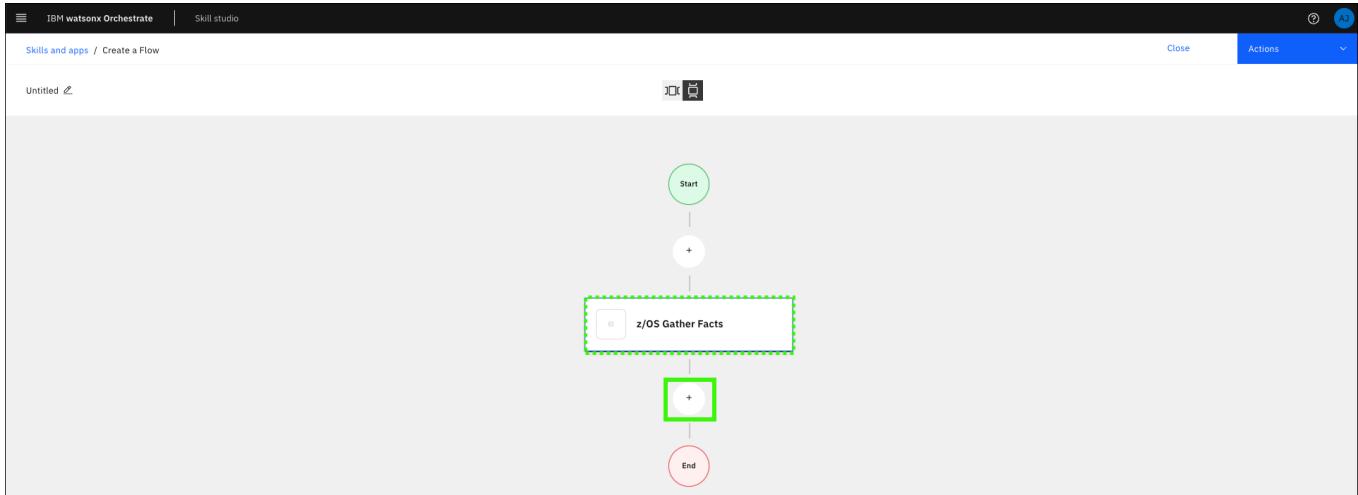
5. Click **Add Skill** in the [z/OS Gather Facts](#) tile.



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

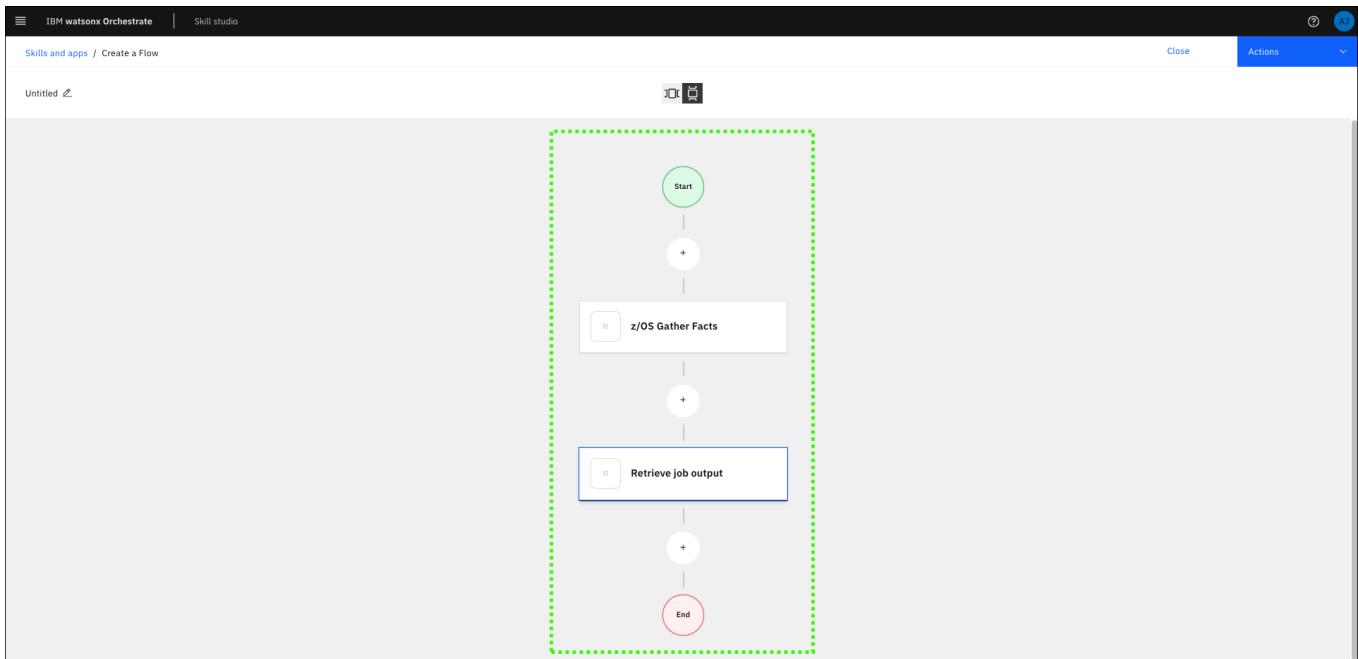


7. Click the **+** icon after the **z/OS Gather Facts** tile.



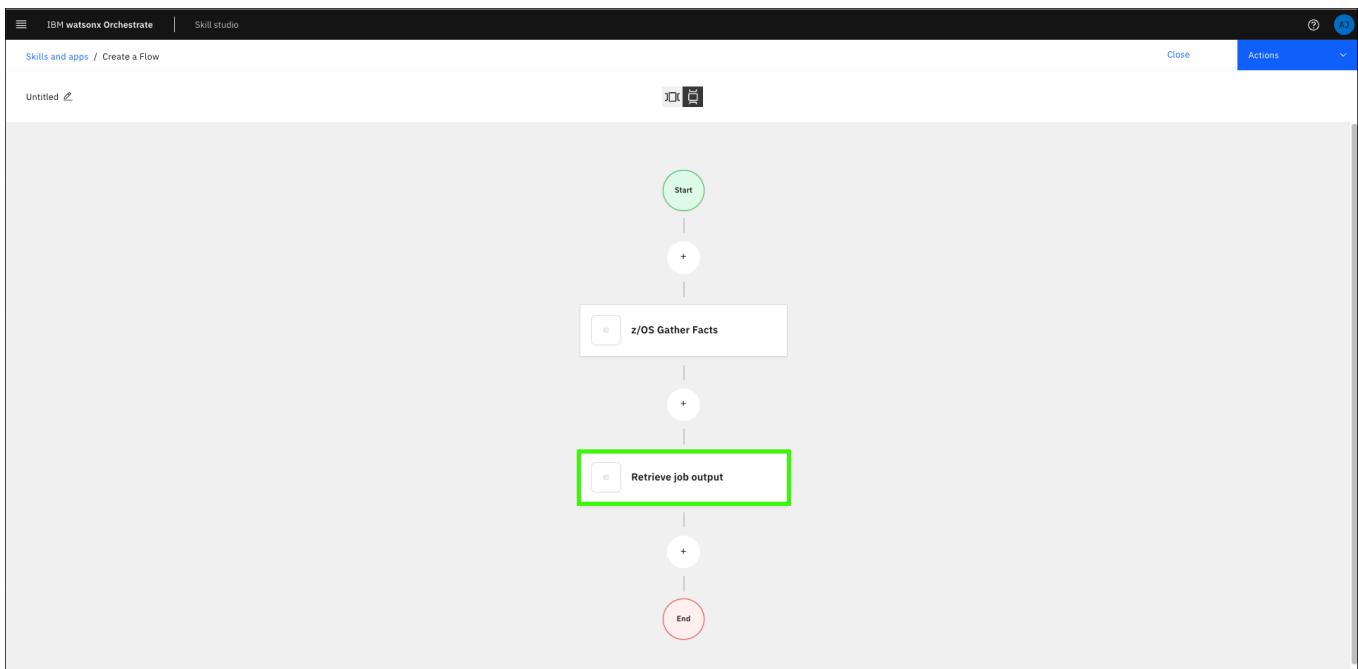
8. Repeat steps 5 and 6 for the **Retrieve job output** skill.

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

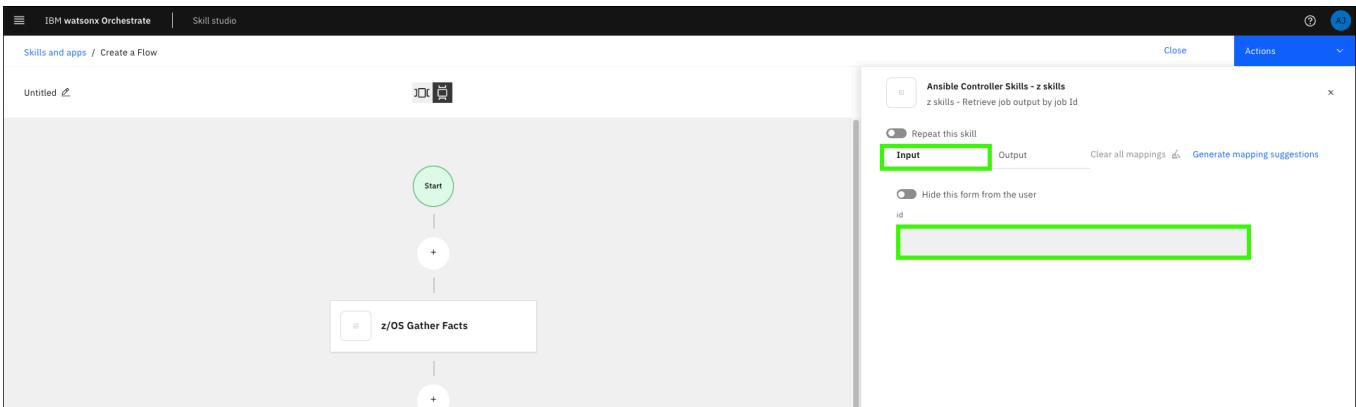


Next you must map the output values of the first skill to the input of the second skill. In this case, pass the job ID output from **z/OS Gather Facts** as an input for **Retrieve job output**.

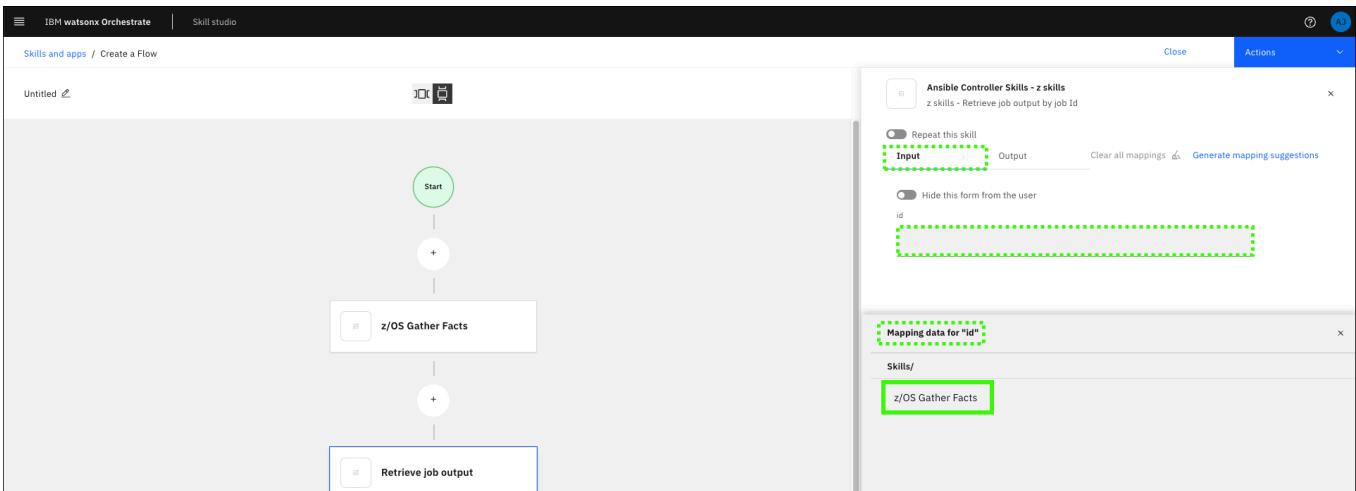
9. Click the **Retrieve job output** tile.



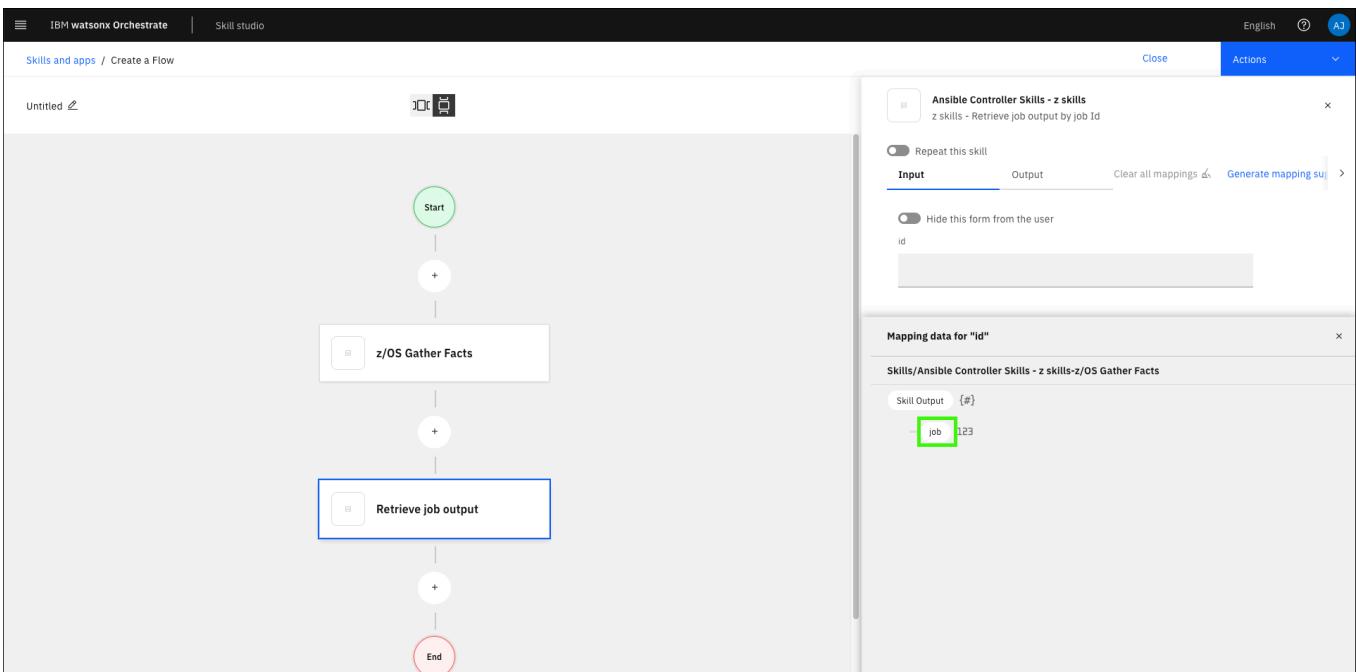
10. Select the **Input** tab and click the **id** field.



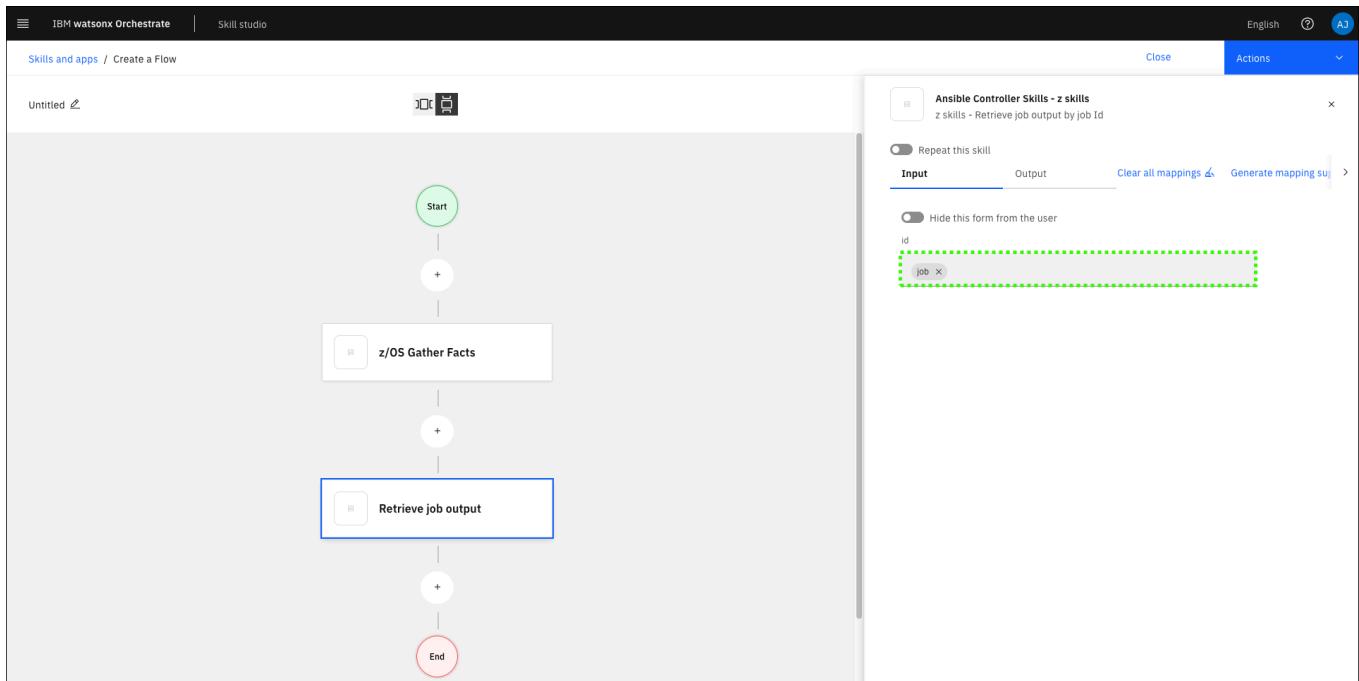
11. Click the **z/OS Gather Facts** skill in the **Mapping data for "id"** section.



12. Click the **job** icon.

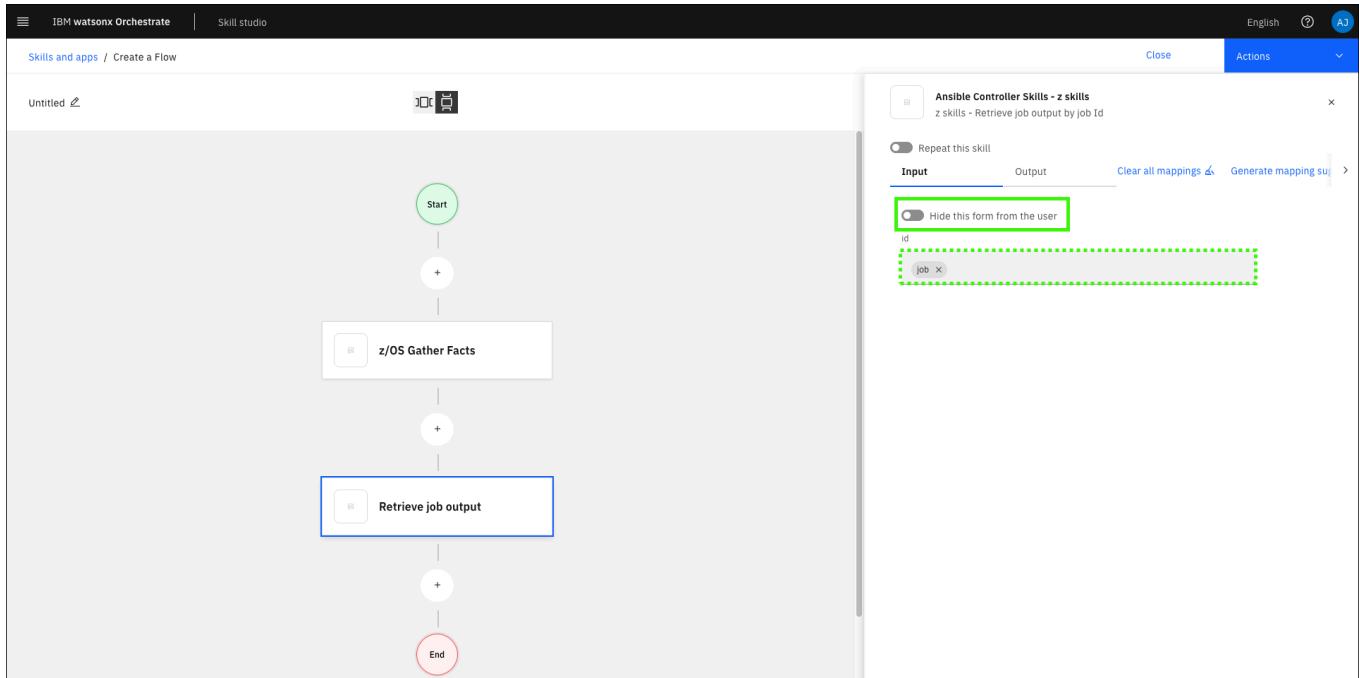


13. Verify that the **job** appears in the **id** field.

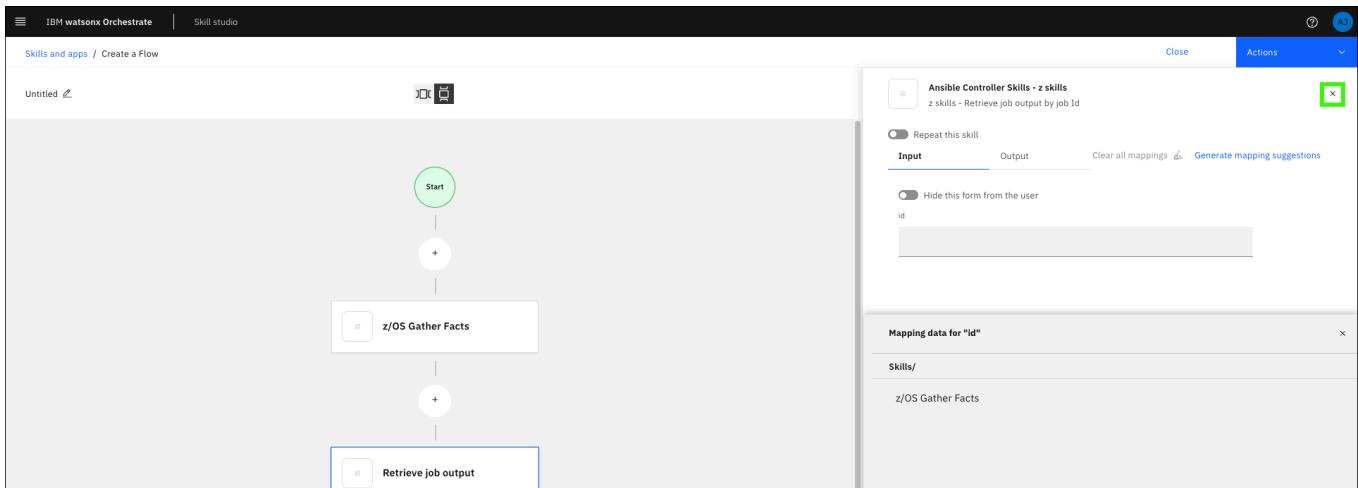


14. Optionally, toggle the **Hide this from the user** setting.

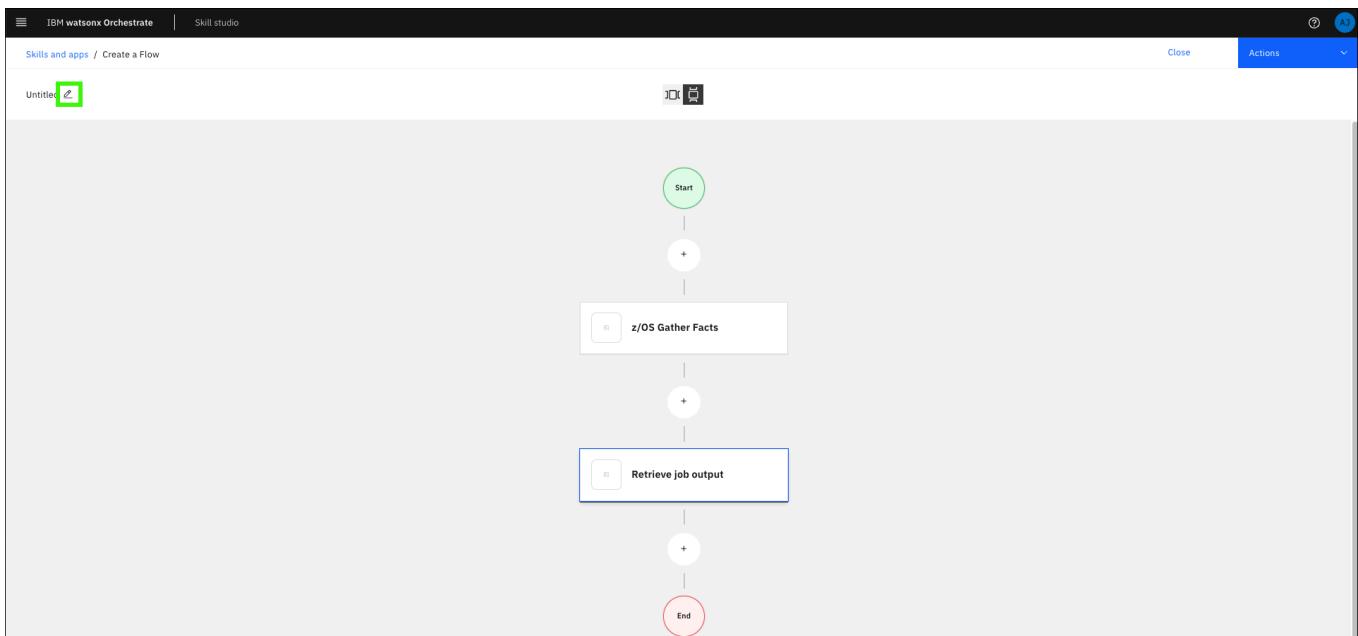
For this lab guide, this option is left disabled. Learn more about this option [here](#).



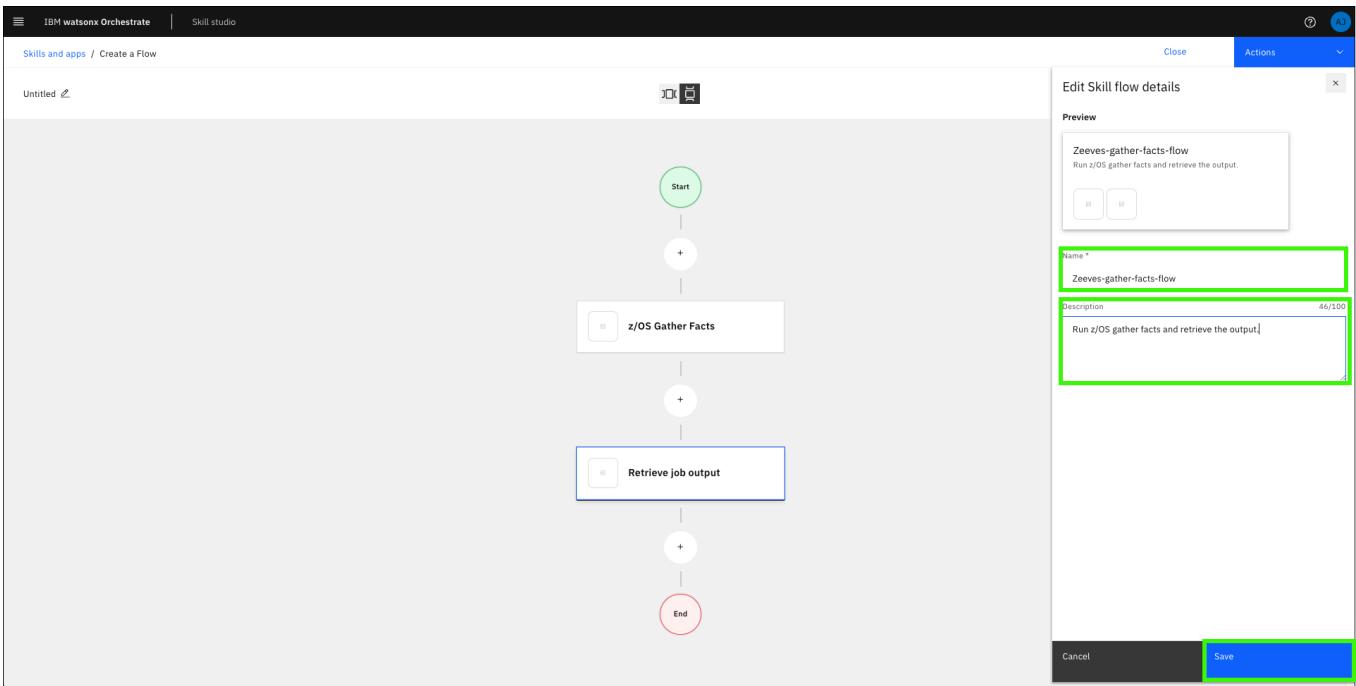
15. Click the **x** to close mapping window.



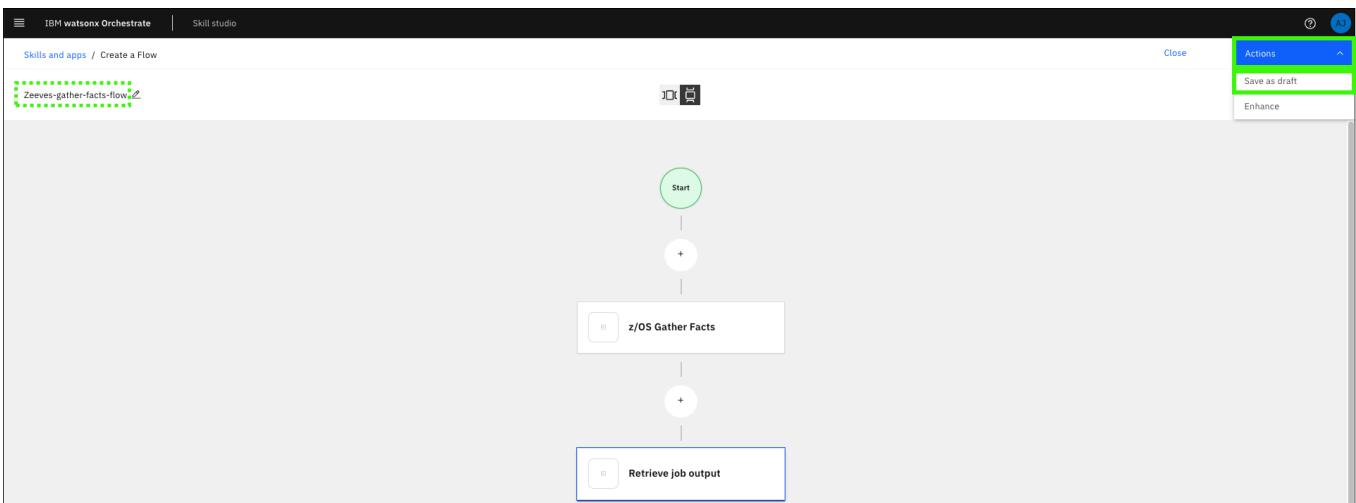
16. Click the pencil (✍).



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.

Skills and apps / Enhance this skill

Enhance the "Zeeves-gather-facts-flow" skill

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

Name Phrases Next best skills

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

Zeeves-gather-facts-flow

Run z/OS gather facts and retrieve the output.

Enter new train phrase

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.

Example prompts:

Show me z/OS facts

Gather and display z/OS facts

Skills and apps / Enhance this skill

Enhance the "Zeeves-gather-facts-flow" skill

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

Name Phrases Next best skills

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

Show me z/OS facts

Gather and display z/OS 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.

The screenshot shows the IBM Watsonx Orchestrate interface. The left sidebar has a dark theme with several sections: Chat, Skill sets, Skill catalog, BUILD (with AI assistant builder highlighted), ADMINISTER, and ACCESS MANAGEMENT. The main area is titled "Skill studio" and contains a message: "Welcome to the skill studio. This is where you can build and manage skill flows and connectors." Below this is a table with four rows of skill information:

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

A green box highlights the "AI assistant builder" option in the sidebar.

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

The screenshot shows the "Actions" section of the AI assistant builder. The left sidebar has sections: Home (highlighted), Build, Generative AI, and Actions (highlighted). The main area displays a series of cards with actions:

- Build actions**: Enhance and improve your assistant's actions.
- Customize your greeting**: Welcome your users in a unique way that aligns with your brand.
- 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 the channels you want to deploy your assistant to.
- Set up live agent**: Integrate live agents to transfer your users to when you edit web chat.

Below these cards is a diagram showing a "Default behavior" node branching into "General purpose" and "Conversational search" nodes, which then lead to a "Search" node.

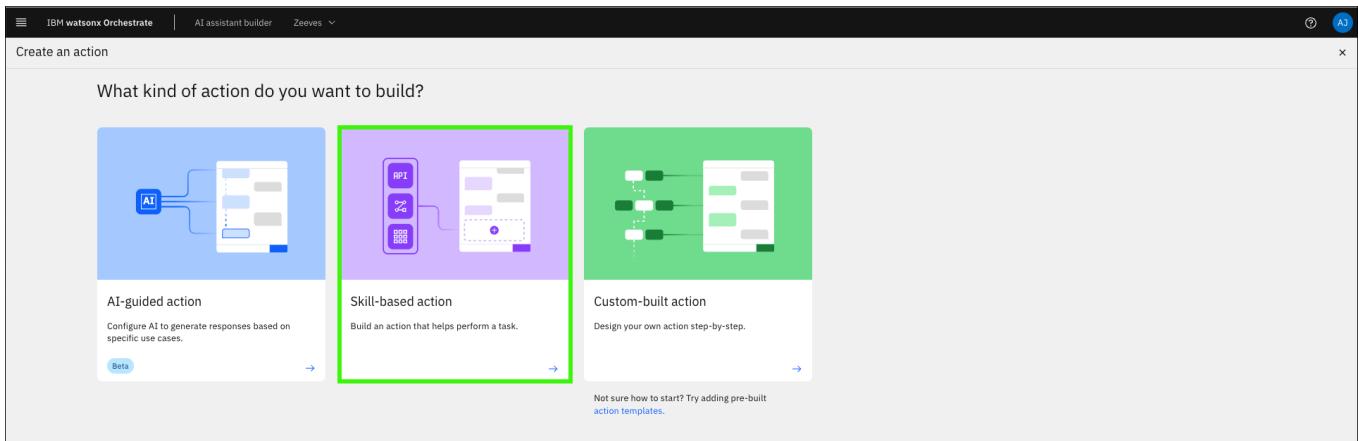
3. Click **New action**.

The screenshot shows the "Actions" list interface. The left sidebar has sections: Actions (highlighted), All items, Created by you (highlighted), Variables, and Saved responses. The main area lists actions under "Created by you /":

Name	Last edited	Examples count	Steps count	Status
z/OS Gather Facts	2 hours ago	2	0	Green circle icon

A green box highlights the "New action" button in the top right corner of the list area.

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



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.

The screenshot shows the 'Build an action from a skill' screen. It lists several skills:

Skill Name	Description	Last updated
Zeeves-gather-facts-flow	Run z/OS gather facts and retrieve the output.	2024-11-19T21:18:31.793Z
Retrieve job output	z skills - Retrieve job output by job Id	2024-11-19T20:08:59.538Z
z/OS Ping	z skills - This playbook pings the z/OS host to test connectivity.	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.	2024-11-19T15:56:26.843Z
Summarize the Webex meeting transcript	in watsonx.ai	2024-11-04T10:49:16.502Z
Summarize the Box content	in watsonx.ai	2024-11-04T10:49:12.077Z
Summarize a Zendesk ticket	in watsonx.ai	2024-11-04T10:49:09.476Z
Summarize a ServiceNow incident	in watsonx.ai	2024-11-04T10:49:05.828Z
Summarize a Salesforce opportunity	in watsonx.ai	2024-11-04T10:49:01.769Z
Sharepoint document summary	in watsonx.ai	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.

The screenshot shows the configuration screen for the 'Zeeves-gather-facts-flow' skill. It includes sections for 'Customer starts with:' and 'Add example phrases:'.

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.

New action
What does your customer say to start this interaction?
Show me z/OS facts

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

The screenshot shows the AI assistant builder interface for a skill flow named 'Zeeves-gather-facts-flow'. In the configuration pane, it specifies 'Customer starts with: Show me zOS facts'. The 'Add example phrases' section contains a single phrase: 'Show me zOS facts'. A green box highlights the close button (X) in the top right corner of the configuration pane.

8. Click close (x).

This screenshot is identical to the previous one, showing the configuration of the 'Zeeves-gather-facts-flow' skill flow. The 'Add example phrases' section contains the phrase 'Show me zOS facts'. A green box highlights the close button (X) in the top right corner of the configuration pane.

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).

The screenshot shows the Actions page in the IBM Watsonx Orchestrator. The left sidebar shows 'Actions' under 'Created by you'. A skill named 'z/OS Gather Facts' is selected, indicated by a green box 'a'. A context menu is open over this skill, with the 'Delete' option highlighted by a green box 'c'. Other options in the menu include 'Rename' and 'Duplicate'. A green box 'b' highlights the three-dot ellipsis icon in the menu header.

10. Wait for system training to complete.

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

The screenshot shows the Actions page in the IBM Watsonx Orchestrator. The left sidebar shows 'Actions' under 'Created by you'. A skill named 'Zeeves-gather-facts-flow' is selected. A status message 'System is training...' is displayed at the top right of the main pane. The skill row for 'Zeeves-gather-facts-flow' shows it was last edited 2 minutes ago, has 1 example, and 0 steps.

11. Click **Preview**.

The screenshot shows the IBM Watsonx Orchestrate interface. In the top navigation bar, it says "IBM Watsonx Orchestrate", "AI assistant builder", and "Zeeves". On the left, there's a sidebar titled "Actions" with sections for "All items", "Created by you", "Variables", and "Saved responses". The main area is titled "Created by you /" and shows 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 3 minutes ago, with 1 example and 0 steps, and a green status icon. At the bottom right of the main area, there's a "Preview" button, which is highlighted with a green box.

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

The screenshot shows the IBM Watsonx Orchestrate interface. The "Preview" window is open, displaying a conversation between a customer and an AI assistant. The customer message "Show me z/OS facts" is highlighted with a green box. The AI response "Welcome, how can I assist you?" is shown below. The "Preview" window has a close button and a "Cancel" button at the bottom.

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. On the left, there's a sidebar with 'Actions' selected. Under 'Created by you', there's a list item 'Zeeves-gather-facts-flow'. On the right, a preview window shows a conversation between a user and a bot. The user says 'Show me zOS facts' at 8:13 AM, and the bot responds with 'Zeeves-gather-facts-flow' at 8:13 AM.

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.

The screenshot shows the same interface as before, but the preview window now displays a large amount of JSON-like data. A portion of this data is highlighted with a green dashed box. The data includes parameters like 'ipl_volume', 'load_param_device', 'load_param_dsn', 'operator_prompt_flag', and 'nucleus_id'.



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 Watson Assistant 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 'Draft' status indicator. A callout box points to this 'Draft' button. Below it is an 'Edit content' link. The 'Extensions' section contains a 'Search' component.

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 'No published version' message. A callout box points to this message. Below it is a 'Publish version' link. The 'Extensions' section contains a 'Search' component with a note about extending what the assistant can answer by searching existing documents. There's also an 'Add +' button.

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, a sidebar has "Publish" highlighted with a green border. The main area displays several cards: "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". Below these cards is a diagram showing a "Default behavior" node connected to "General purpose" and "Conversational search" nodes, which both point to a "Search" node. The status bar at the bottom indicates "1/2" of the page.

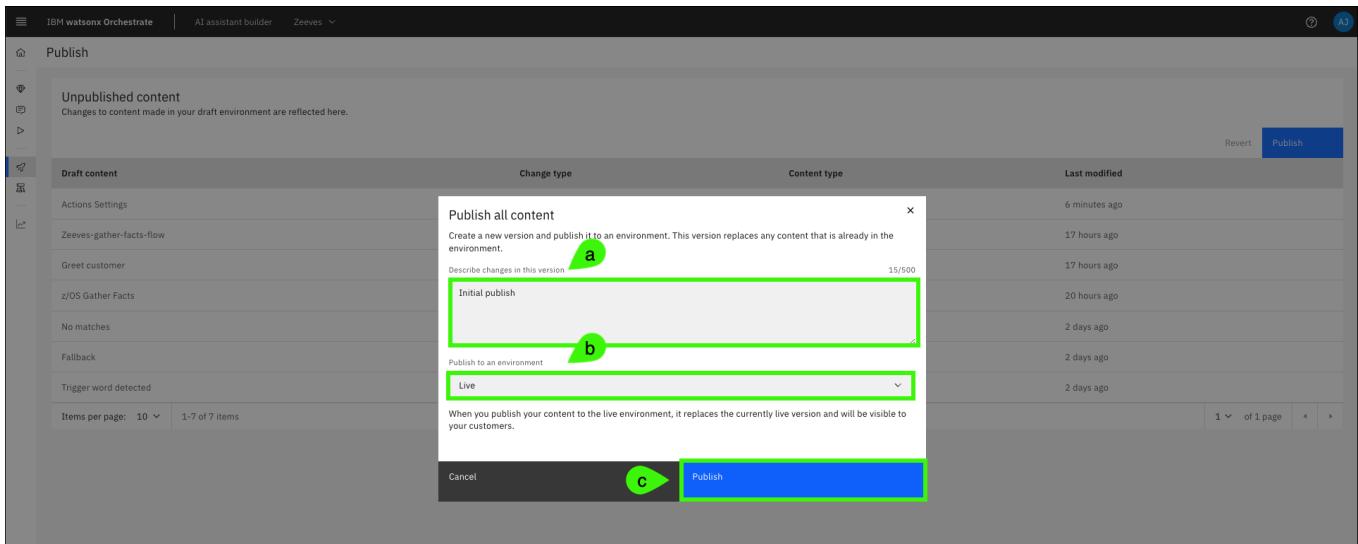
2. Click **Publish**.

The screenshot shows the "Publish" screen with the "Draft content" table. The table lists changes made to the assistant's components:

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

At the bottom, there are buttons for "Revert" and "Publish", with "Publish" highlighted with a blue border.

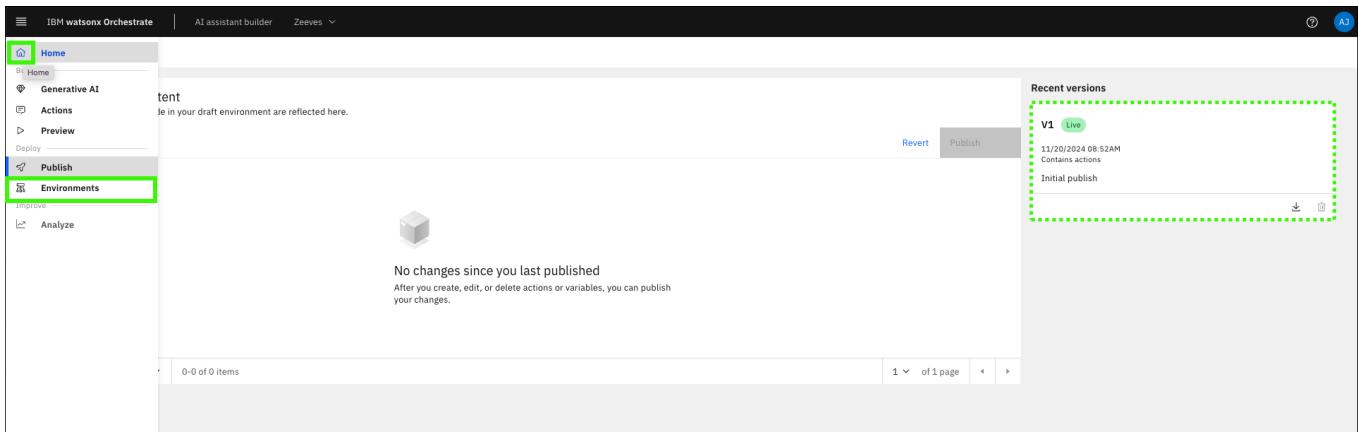
3. Enter a description of the changes (a), set the environment to **Live** (b), and then click **Publish** (c).



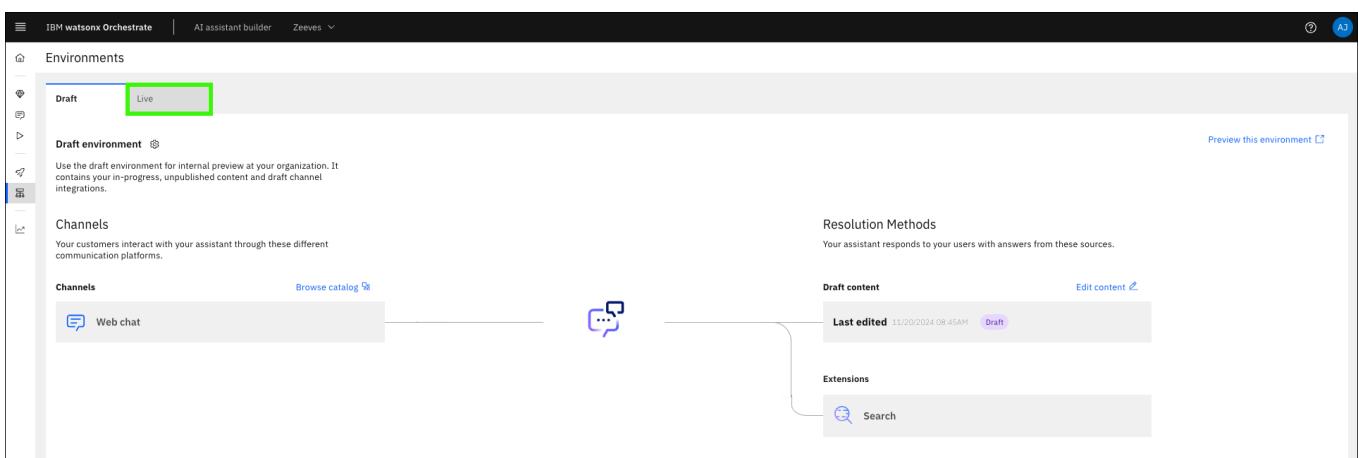
Configure the live environment

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 'Live' environment configuration in the IBM Watsonx Orchestrate interface. The 'Web chat' channel is highlighted with a green box. Other channels listed are 'Email', 'SMS', 'Facebook', 'Twitter', and 'LinkedIn'. The 'Resolution Methods' section indicates the assistant responds from various sources. The 'Content' section shows version V1 from 11/20/2024 at 08:52AM in 'Live' mode. The 'Extensions' section includes a 'Search' extension for extending the assistant's answers.

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.

Style

Suggestions (highlighted with a green box)

Restart conversation

Save and exit

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 1.0

Intended purpose: Standard: For virtual agents and customer support experiences.

Choose a theme: Light (highlighted with a green box) Dark

Primary color: #FFFFFF Secondary color: #3D3D3D

Chat header: User message bubble

Accent color: #035AE9

Significant and interactive objects

Size: Width 380px Height 640px

IBM Watermark: Enable IBM Watermark (On)

Streaming: Enable Streaming (On)

Zeeves 1.0

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

Type something... ▶

Powered by IBM watsonx ⓘ

5. Click Save and exit.

Style

Save and exit (highlighted with a green box)

Restart conversation

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 1.0

Intended purpose: Standard: For virtual agents and customer support experiences.

Choose a theme: Light (highlighted with a green box) Dark

Primary color: #FFFFFF Secondary color: #3D3D3D

Chat header: User message bubble

Accent color: #035AE9

Significant and interactive objects

Size: Width 380px Height 640px

IBM Watermark: Enable IBM Watermark (On)

Streaming: Enable Streaming (On)

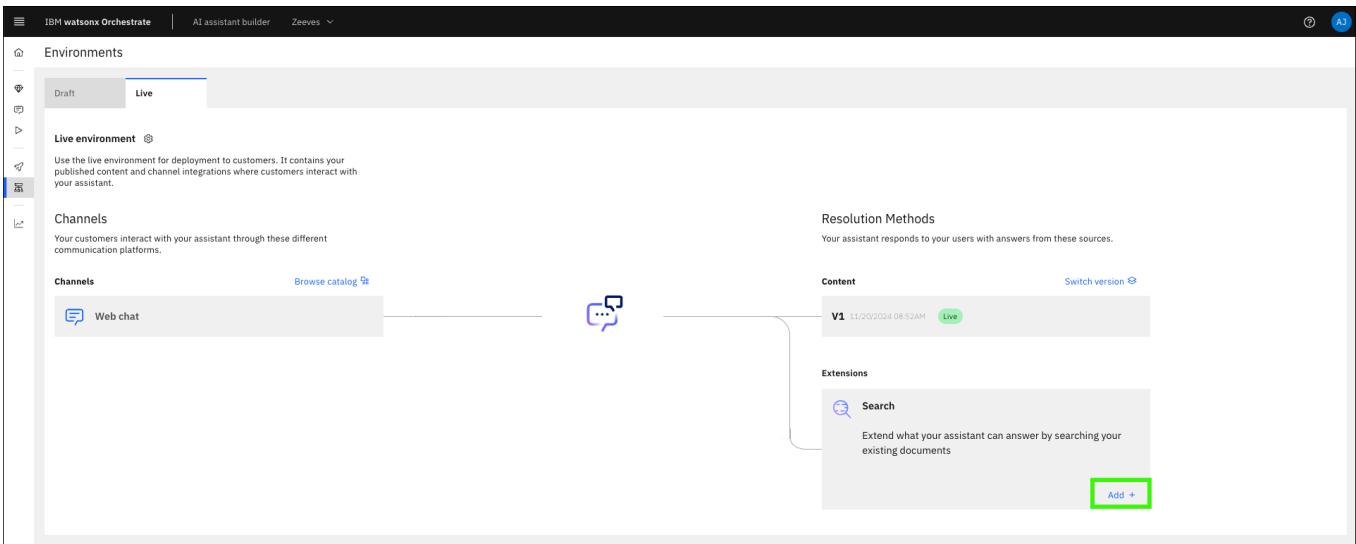
Zeeves 1.0

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

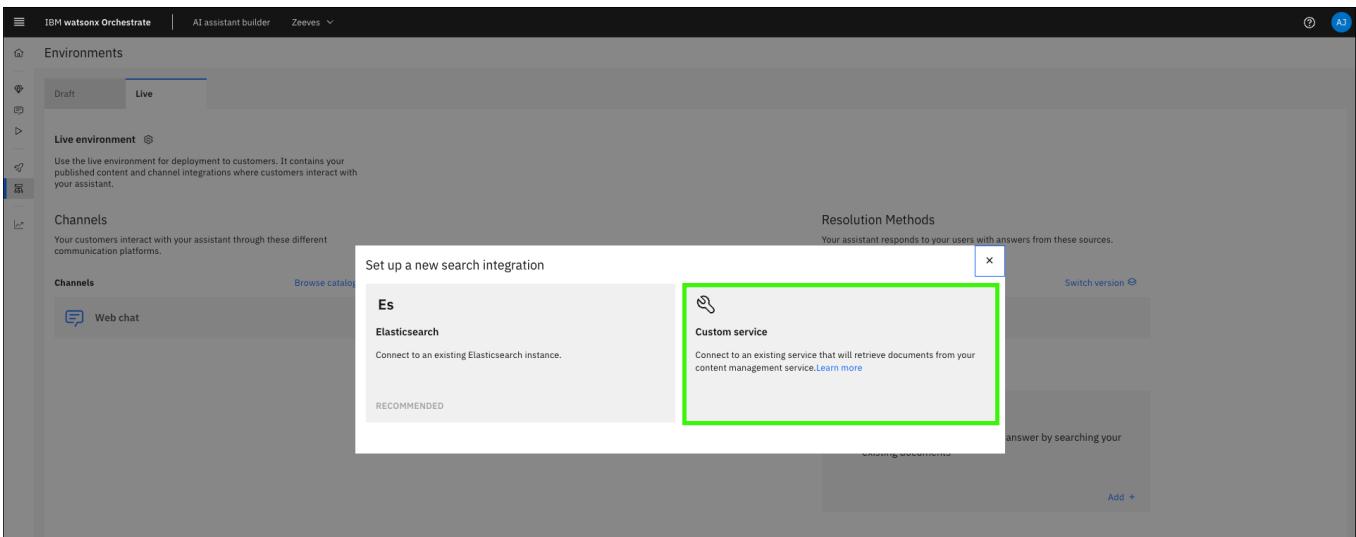
Type something... ▶

Powered by IBM watsonx ⓘ

6. Click Add in the Search tile.



7. Click Custom service.

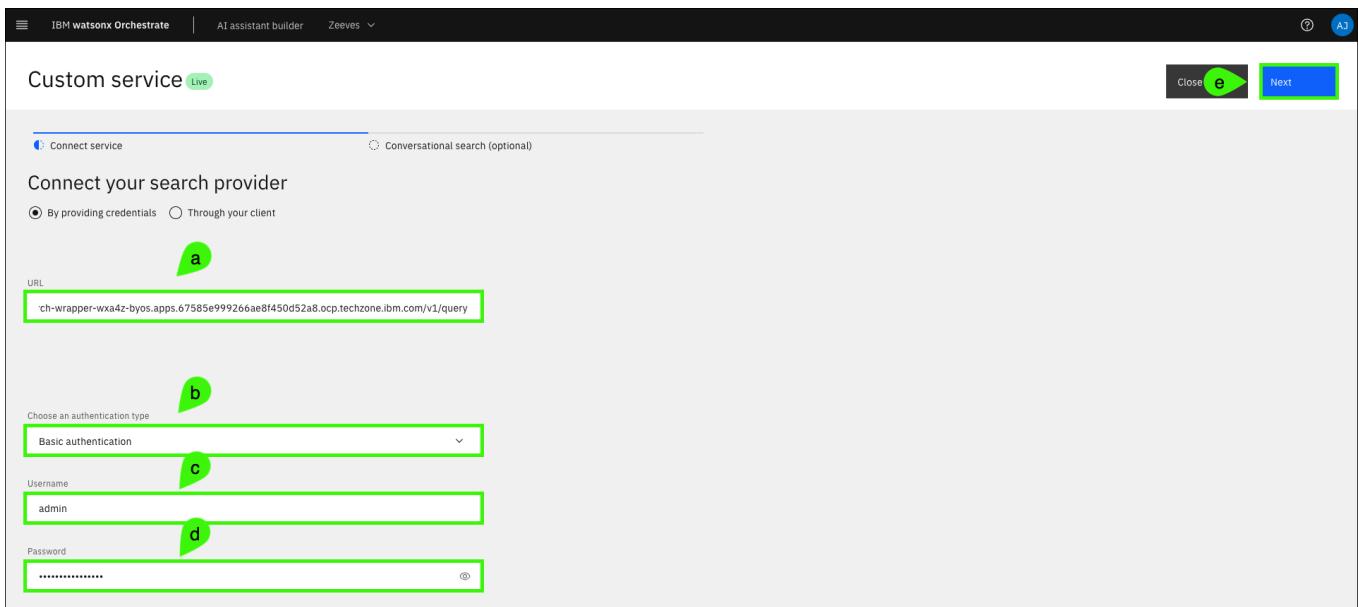


8. Enter the URL for your bring-your-own-search (BYOS) engine (a), select **Basic authentication** for the **authentication type** (b), enter **admin** for the **Username** (c), enter the password that you specified in the `wrapper-creds.yaml` file for the **Password** (d), and then click **Next** (e).

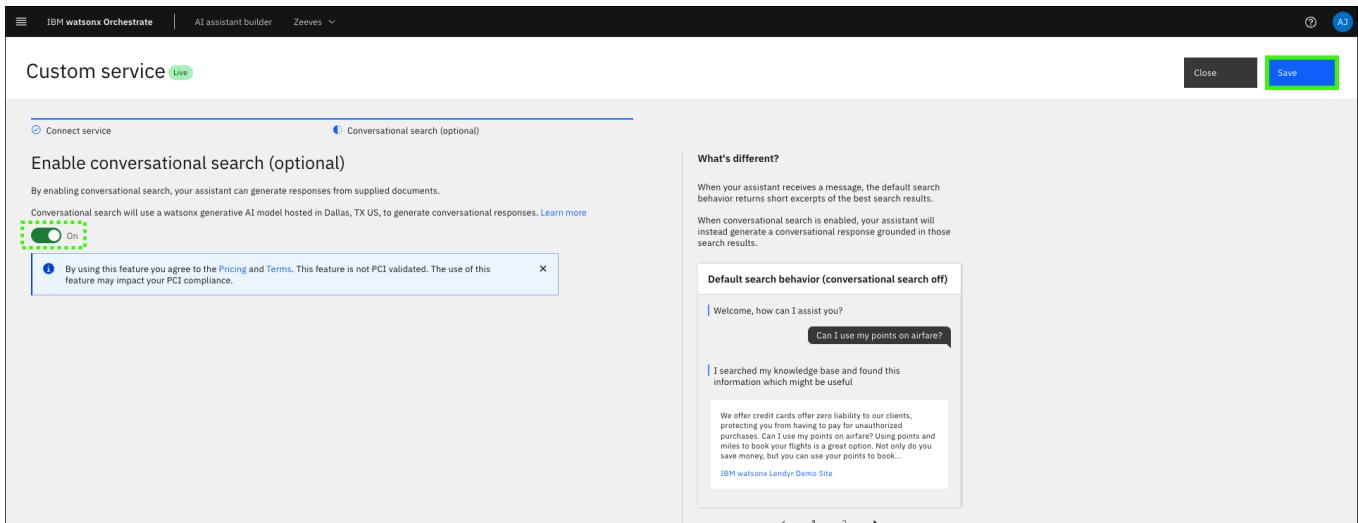


Use the correct URL and authentication type!

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_*" }
```

Custom service Live

Settings **Instance**

By using this feature you agree to the [Pricing](#) and [Terms](#). This feature is not PCI validated. The use of this feature may impact your PCI compliance. X

Conversational search

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

Conversation length

a On

b Off

Context-dependent questions

c 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.

d 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

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

Retrieval confidence threshold ⓘ

e Lowest Low High Highest

Generated response length ⓘ

Concise Moderate **f** Verbose

Response confidence threshold ⓘ

g Lowest Low High Highest

Citations

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

Citations shown in web chat

All

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

Close **Save** A3

Connect the skills to the live environment

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

The screenshot shows the IBM Watsonx Orchestrate interface. On the left, there's a sidebar with sections: Chat, Skill sets (highlighted with a green border), Skill catalog, BUILD (AI assistant builder, Skill studio), and ADMINISTER (Access management). The main area has a header: AI assistant builder Zeeves. Below the header, there's a 'Resolution Methods' section 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: 'The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.'

3. Click **Connections**.

The screenshot shows the 'Zeeves live' skill set page. The 'Skills' tab is selected (highlighted with a green border), while 'Connections' is also present. Below the tabs, there's a note: 'Skills (3) The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.' Below this, there's a 'Name' section with a placeholder 'No skills' and a 'Manage skills' button. A note says: 'The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.' Below this, there's a 'Name' section with a placeholder 'No skills' and a 'Manage skills' button. A note says: 'The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.'

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

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	-	⋮

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

6. Click **Connect app**.

Skill sets

Zeeves live

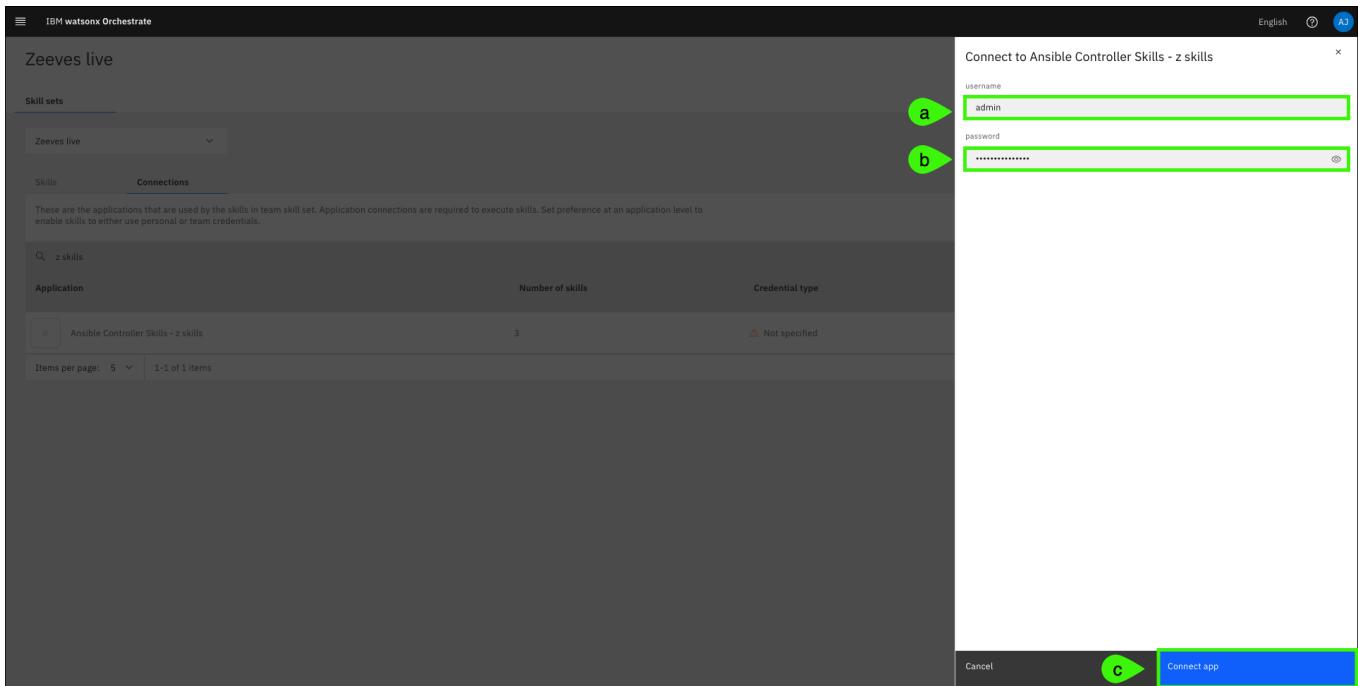
Skills **Connections**

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

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

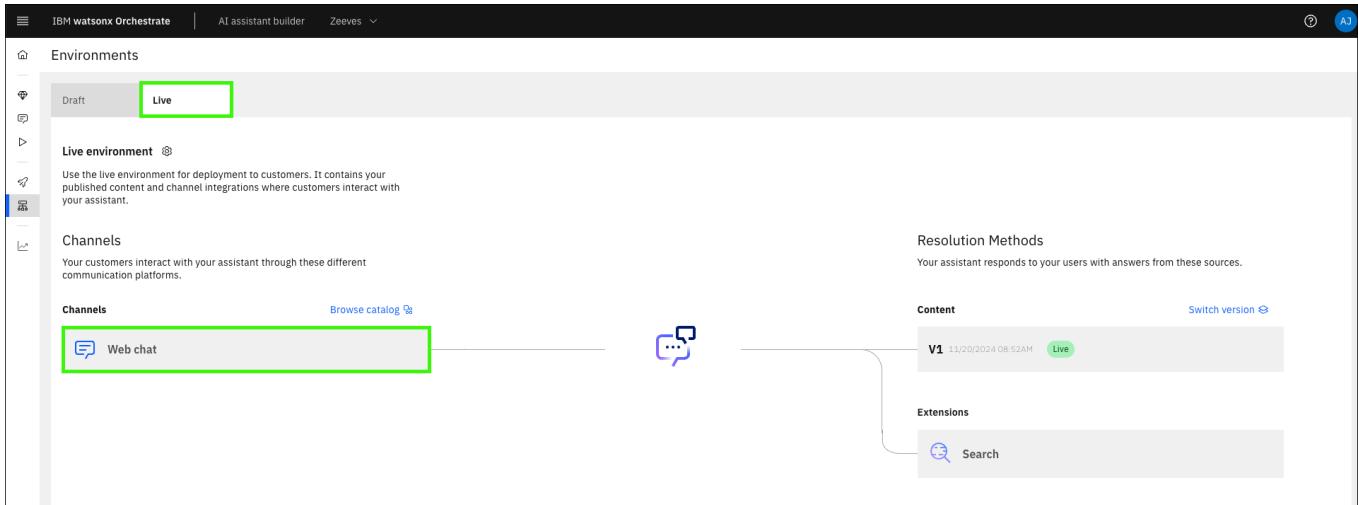
After configuring your assistant's settings and publishing, the final step is to deploy your assistant, which can be done across various channels depending on the use case.

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 fields for 'Assistant's name as known by customers' (Zeeves 1.0), 'Intended purpose' (Standard: For virtual agents and customer support experiences), 'Choose a theme' (Light selected), 'Primary color' (#FFFFFF), 'Secondary color' (#3D3D3D), 'Chat header' (Accent color #0354E9), 'Significant and interactive objects' (Size: Width 380px, Height 640px), and 'IBM Watermark' (Enabled). A 'Change avatar image' button is also present.

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. 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. Below the preview, a code editor displays the embed script:

```
<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-bdf2-319f531087be', // 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,
}

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') + '/WatsonAssistant';
  document.head.appendChild(t);
});
</script>
</body>
</html>

```

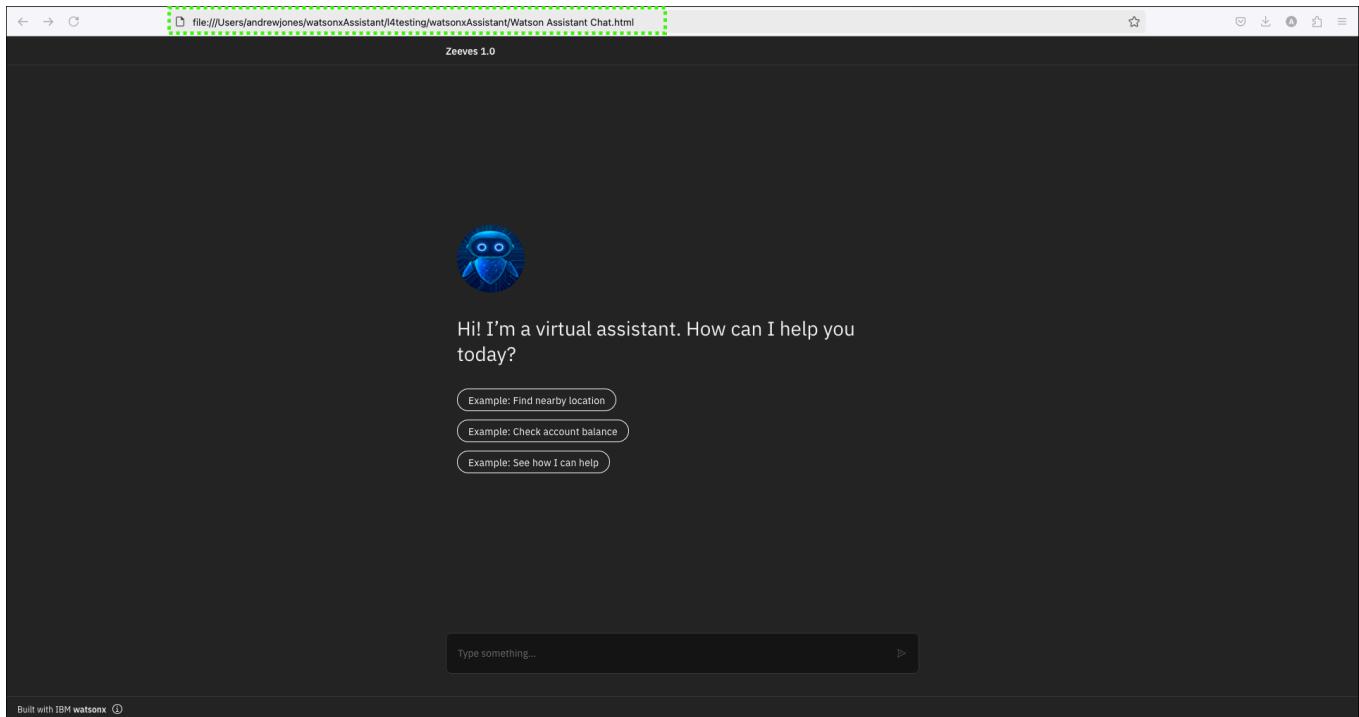
After modification:

```

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: "wxs-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

Use cases

Use case: Retrieve IPL information

Next, explore a use case to improve productivity for early-tenure system programmers (SysProg) who are preparing for an upcoming initial program load (IPL) for a logical partition (LPAR).

To prepare for the IPL, the SysProgs need to familiarize themselves with the process. Rather than spending time reading through the wide array of documentation available online, they would like to use watsonx Assistant for Z. The content-grounded capabilities that are provided by watsonx Assistant for Z return accurate responses to their questions quickly and uses automations to perform actions.

As part of the pilot, they already explored prompting the assistant with questions. In one example prompt, they asked the assistant is “*What information is needed to perform an IPL on a z/OS LPAR?*” In reading the response, the SysProg learns they need information about their system in preparation for the IPL. For example, the IPL Volume and the IPL LOAD PARM.

For this use case, show how a simple automation for retrieving this type of information can be infused in a natural conversation with watsonx Assistant for Z. You use pre-packaged skills to automate various tasks on z/OS. The pre-packaged skills are provided as an OpenAPI JSON file. Learn more about OpenAPI [here](#). The file includes skill definitions that can be uploaded to the Skill Studio within watsonx Orchestrate to import the pre-packaged skills. First, the file must first be customized for your z/OS server.

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 (z/OSMF) job status

A great value of watsonx Orchestrate is the ability to build skills that anyone can use. You can build your own custom skills by importing an OpenAPI file into watsonx Orchestrate as a JSON or YAML file. For more information on building skills by importing OpenAPI files, refer to the documentation [here](#).

Watsonx Orchestrate also makes it possible to build, edit, and generate OpenAPI specifications by using the OpenAPI builder. With the OpenAPI Builder, you can use the AI function to simplify the process of generating these specifications. For more information on using the OpenAPI Builder, refer to the documentation [here](#).

For this use case, you are importing the skill for retrieving a z/OS server’s IPL information. The next steps walk you through the process of downloading that OpenAPI JSON file and customizing it for your environment.

Download and customize the watsonx Assistant for Z OpenAP JSON file

1. Download the watsonx Assistant for Z OpenAP JSON file.

wxa4z-skillpak-prepackaged-skills.json.zip

2. Extract the file.

3. In a text editor, open the `wxa4z-skillpak-prepackaged-skills.json`, modify the server's url field as described, and save the file.

File name:

wxa4z-skillpak-prepackaged-skills.json

Substitute your Wazi as a Service (WaaS) instance URL for the string <your z/OSMV URL>. Your WaaS URL is based on your ITZ AAP URL that can be obtained from your watsonx Assistant for Z Pilot - AAP & z/OS ITZ reservation.

The AAP URL is similar to:

<https://itzvsi-aap-ppxbcno.techzone.ibm.com>

Change the `aap` string to `zos` and append `:10443` to the URL value. **Record this value for later use!** Your new URL is similar to:

<https://itzvsi-zis-ppxbcn0.techzone.ibm.com:10443>

Before:

{ wxa4z-skillpak-prepackaged-skills.json }

Users > andrewjones > watsonxAssistant > finaltest > watsonxAssistant > {} wxa4z-skillpak-prepackaged-skills.json > ...

```
1 [ "openapi": "3.0.1",
2   "servers": [
3     {
4       "url": "<your z/OSMF URL>"
5     }
6   ],
7 ],
8   "info": {
9     "description": "z/OS operations",
10    "version": "1.0.0",
11    "title": "IBM watsonX Assistant for Z skills pak",
12    "x-ibm-annotations": "1.0.0",
13    "x-ibm-application-name": "z/OS operations",
14    "x-ibm-application-icon": "",
15    "x-ibm-application-id": "zos-operations",
16    "x-ibm-skill-subtype": "public",
17    "x-ibm-skill-type": "imported",
18    "x-ibm-disable-default-server-url": "true"
19  },
20  "tags": [
21    {
22      "name": "zOS operations"
23    }
24  ],
25  "paths": {
26    "/zosmf/restconsoles/consoles/{consoleName}?authProgList": {
27      "put": {
```

After:

```
{} wxa4z-skillpak-prepackaged-skills.json ●
Users > andrewjones > watsonxAssistant > finaltest > watsonxAssistant > {} wxa4z-skillpak-prepackaged-skills.json > {} info
1  {
2    "openapi": "3.0.1",
3    "servers": [
4      {
5        "url": "https://itzvsi-zos-pwgabob.techzone.ibm.com:10443"
6      }
7    ],
8    "info": [
9      "description": "z/OS operations",
10     "version": "1.0.0",
11     "title": "IBM watsonX Assistant for Z skills pak",
12     "x-ibm-annotations": "1.0.0",
13     "x-ibm-application-name": "z/OS operations",
14     "x-ibm-application-icon": "",
15     "x-ibm-application-id": "zos-operations",
16     "x-ibm-skill-subtype": "public",
17     "x-ibm-skill-type": "imported",
18     "x-ibm-disable-default-server-url": "true"
19   ],
20   "tags": [
21     {
22       "name": "zOS operations"
23     }
24   ],
25   "paths": {
26     "/zosmf/restconsoles/consoles/{consoleName}?authProgList": {

```



In watsonx Orchestrate, each app is associated with a single URL.

If you have not imported previous skills into an app called `z/OS operations`, the default values in the `info` section are fine. If you already have an app named "z/OS operations", modify the `info` section to meet your needs.

For more information on modifying the OpenAPI JSON file, see the instructions [here](#).

Set the RACF passphrase for z/OSMF authentication

For your skills to run successfully on your WaaS instance, you need to ensure that you can authenticate to it from watsonx Orchestrate. To achieve successful authentication, setup a new RACF Passphrase for the `IBMUSER` ID that is a pre-defined user ID on the WaaS server. The following steps take you through the steps of setting a new passphrase for your user and verifying access.

1. Open and log in to the Ansible Automation Platform (AAP) web console.



Don't remember how?

Refer to the first 5 steps in [Explore Ansible Automation Platform](#).

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

Name	Type	Organization	Last Ran	Actions
z/OS Certs - Add CA Cert	Job Template	Default	1/17/2025, 5:17:58 AM	
z/OS Certs - Create Cert	Job Template	Default		
z/OS Certs - Create Keyring	Job Template	Default		
z/OS Certs - Delete Cert	Job Template	Default		
z/OS Certs - Delete Keyring	Job Template	Default		

3. Click the launch icon () for the **z/OS TSO Command(s)** template.

Name	Type	Organization	Last Ran	Actions
z/OS TSO Command(s)	Job Template	Default		

4. Replace the default command with the text that follows and substituting a password of your choosing for the string `YOUR PASSWORD PHRASE` and then click **Next**.

```
ALTUSER IBMUSER PHRASE('YOUR PASSWORD PHRASE') NOEXPIRE RESUME
```

Avoid typographical errors later... keep the password simple.

If you type the command yourself, be sure to include the single quotes before and after the password. **Record the password as it will be needed later.** We will refer to this as your **WaaS password**."

Launch | z/OS TSO Command(s)

A playbook to execute one or multiple TSO commands on the target z/OS system.

1 Survey 2 Preview

Command(s) (?)

```
ALTUSER IBMUSER PHRASE('YOUR PASSWORD PHRASE') NOEXPIRE RESUME
```

Actions

Next Back Cancel

5. Click Launch.

Launch | z/OS TSO Command(s)

A playbook to execute one or multiple TSO commands on the target z/OS system.

1 Survey 2 Preview

Name	z/OS TSO Command(s)	Description	Type	Job Template
Timeout	0 min 0 sec	Job Type	Run	Organization Default
Inventory	AAP z/OS	Project	aap4zos	Execution Environment ap4zos EE
Playbook	zos_tso_command.yml	Forks	0	Verbosity 1 (Verbose)
Show Changes	Off	Job Slicing	1	
Credentials	SSH: z/OS Host SSH ...			
Created	1/17/2025, 5:17:49 AM by admin	Last Modified	1/17/2025, 5:17:49 AM by admin	

Actions

Launch Back Cancel

6. Verify that the job is **Successful** by locating the message `"failed": false` in the job output.

The screenshot shows the Red Hat Ansible Automation Platform interface. On the left, there's a sidebar with sections like Views, Jobs, Schedules, Activity Stream, Workflow Approvals, Host Metrics, Subscription Usage, Resources, Access, and Administration. The main area is titled 'Jobs > 13 - z/OS TSO Command(s)' and has a sub-section titled 'Output'. It shows a successful job run with 1 play, 2 tasks, and 1 host. The output tab displays a JSON response from a task. A specific line in the JSON, which contains the word 'failed', is highlighted with a green dashed rectangle.

7. Verify that you can log in to z/OSMF in a new browser tab.

Use the WaaZ URL created earlier and append `/zosmf` to the string. The URL is similar to: <https://itzvsi-zos-pwgabob.techzone.ibm.com:10443/zosmf>.

The screenshot shows a web browser window with a warning message. The address bar shows the URL <https://itzvsi-zos-pwgabob.techzone.ibm.com:10443/zosmf>. The warning message is: "This Connection Is Not Private. This website may be impersonating "itzvsi-zos-pwgabob.techzone.ibm.com" to steal your personal or financial information. You should close this page." There are "Show Details" and "Close Page" buttons at the bottom of the message box.



Accept any connection not private messages to open the page.

8. Enter (a) IBMUSER for the z/OS USER ID, (b) the password you specified in step 4 for the z/OS PASSWORD, and then (c) click LOG IN.

Welcome to z/OS

The highly secure, scalable and resilient enterprise operating system for the IBM z Systems mainframe.

z/OS USER ID
IBMUSER

z/OS PASSWORD
.....

LOG IN

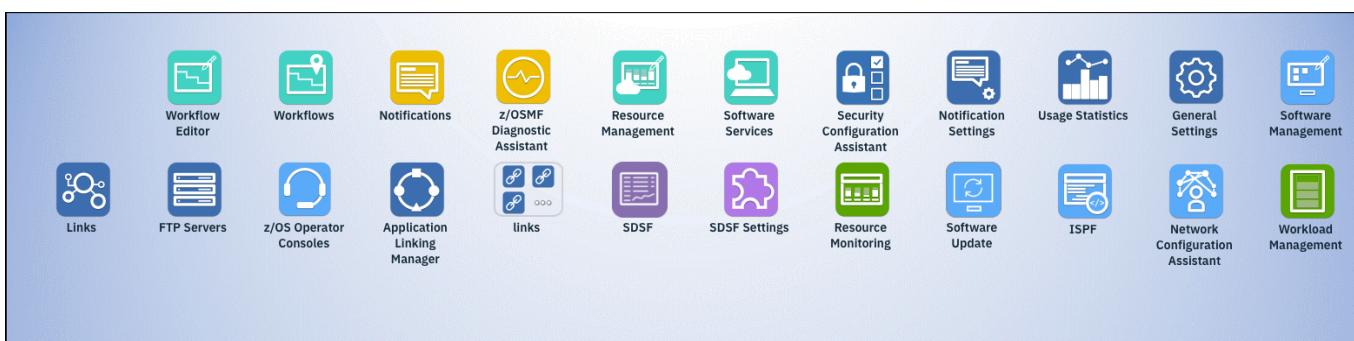
Shopz
IBM Support

z Systems Redbooks
z/OSMF Home Page

WSC Flashes and Techdocs
z/OS Home Page

IBM z/OS documentation

9. Close the new browser tab after verifying a screen similar to the image that follows is displayed.



Import and publish the pre-packaged skills as an OpenAPI file

1. Open **Skills studio** in watsonx Orchestrate.

Welcome, Andrew Jones!

Take productivity to the next level.

Intuitive interaction Natural conversations Contextual clarity

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.

Standard plan | Skill studio

Automate how your business works

Learn more →

Explore

2. Click **Create** and then click **Import API**.

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

Create

Import API

Skill Flow

Configure prebuilt skills

3. Click **From a file**.

Add skills

Choose how you want to add skills and then select the skills you want to refer to from that source.

Choose the source Select the skills

Choose the source

To discover new skills, connect to an app or refer to an OpenAPI file.

From an app **From a file** OpenAPI builder (exp...)

IBM Cloud Pak for Business Automation - On premises
which includes Automation Decision Services, Business Automation Workflow, and Operational Decision Manager on-premises

IBM Cloud Pak for Business Automation as a service
which includes Business Automation Workflow as a service and includes Operational Decision Manager as a service

z/OS Skills accelerator (Trial)
Import your Ansible, JCL/REXX automation to watsonx Orchestrate

4. Click Drag and drop files here or click to upload in the Import a skill file window.

Skills and apps / Add skills

Add skills

Choose how you want to add skills and then select the skills you want to refer to from that source.

Choose the source Select the skills

Choose the source

To discover new skills, connect to an app or refer to an OpenAPI file.

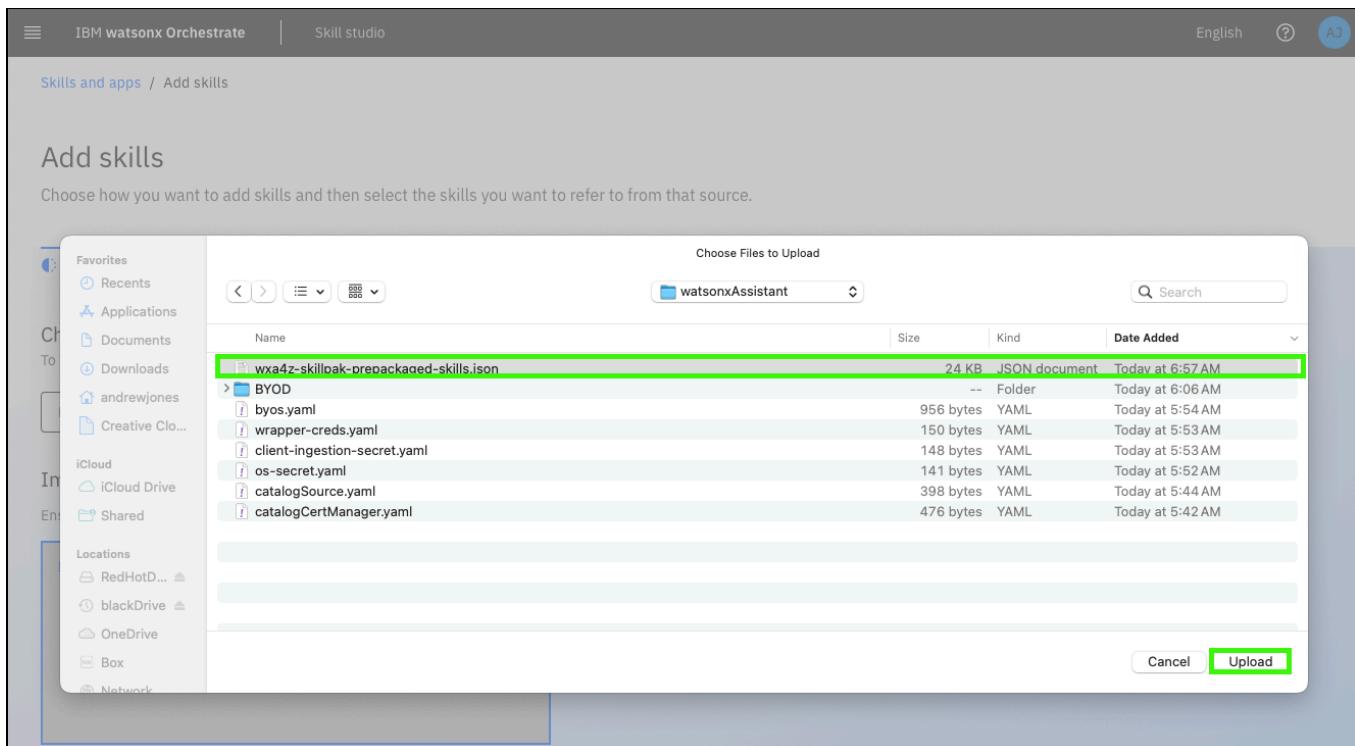
From an app **From a file** OpenAPI builder (exp...)

Import a skill file

Ensure your file is in the .json, or yaml format and no larger than 50 MB.

Drag and drop files here or click to upload

5. Locate and select the JSON file that you modified earlier and then click **Upload**.



6. Verify that you receive the message **The Open file or skill package is good to go!** and then click **Next**.

Skills and apps / Add skills

Add skills

Choose how you want to add skills and then select the skills you want to refer to from that source.

Choose the source Select the skills

Choose the source
To discover new skills, connect to an app or refer to an OpenAPI file.

From an app From a file OpenAPI builder (exp...)

Import a skill file
Ensure your file is in the .json, or yaml format and no larger than 50 MB.

wxa4z-skillpak-prepackaged-skills.json

The OpenAPI file or skill package is good to go! ✓

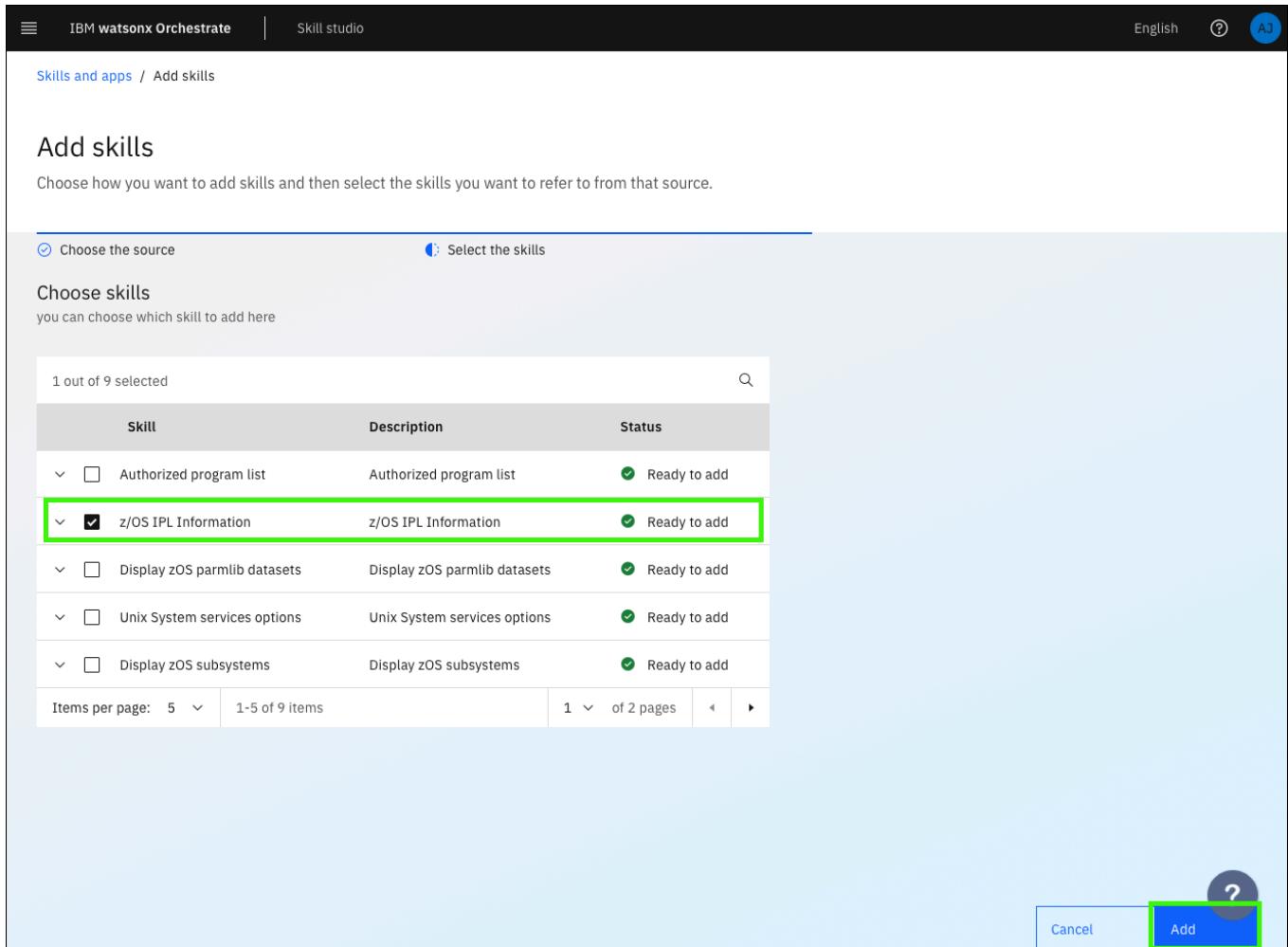
Cancel

✖️ Not good to go?

If the file does not load properly you will need to verify not formatting or errors were made in your json file. Return to the previous section to verify the file contents and then reload the JSON file.

7. Select the **z/OS IPL Information** skill and then click **Add**.

Note: Only the **z/OS IPL Information** skill is required for this use case, but you can add as many skills as you like for testing purposes.



The screenshot shows the 'Skill studio' interface for adding skills. At the top, there are navigation links for 'Skills and apps' and 'Add skills'. On the right, there are language and help buttons. The main area is titled 'Add skills' with a sub-instruction: 'Choose how you want to add skills and then select the skills you want to refer to from that source.' Below this, there are two tabs: 'Choose the source' (selected) and 'Select the skills'. Under 'Choose skills', it says 'you can choose which skill to add here'. A table lists nine skills, with 'z/OS IPL Information' checked and highlighted with a green border. The table columns are 'Skill', 'Description', and 'Status'. The status for all skills is 'Ready to add'. At the bottom, there are pagination controls for 'Items per page' (5), 'Page 1 of 2 pages', and navigation arrows. The 'Add' button at the bottom right is also highlighted with a green border.

8. Click the ellipses icon (~⋮~) for the **z/OS IPL Information** skill and then click **Enhance this skill**.

Welcome to Skill studio

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

Create

Skills and apps Projects

Skills Apps

Configure prebuilt skills

Find a skill

Skills

Name	Step in the process	Status	Skill type	Author	Last edited	More
z/OS IPL Information	Just 1 step away to be ready	Ready to publish	Imported	andrew@jones-tx.com	January 17 2025	⋮
Zeeves-gather-facts-flow	Ready to use	Published	Skill flow	andrew@jones-tx.com	January 17 2025	Enhance this skill
Retrieve job output	Ready to use	Published	Imported	andrew@jones-tx.com	January 17 2025	Export this skill
z/OS Ping	Ready to use	Published	Imported	andrew@jones-tx.com	January 17 2025	Delete this skill
z/OS Gather Facts	Ready to use	Published	Imported	andrew@jones-tx.com	January 17 2025	⋮

9. Review the skill enhancements options and then click Publish.

Skills and apps / Enhance this skill

Enhance the “z/OS IPL Information” skill

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

Name	Input	Output	Security	Phrases	Next best skills
z/OS IPL Information					

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 IPL Information

Description 20/100

z/OS IPL Information

API version*

1.0.0

Categories

Add categories

App

z/OS operations

Preview

The skill will look like this in the catalog.

z/OS IPL Information
z/OS IPL Information

The skill will look like this in the skill set.

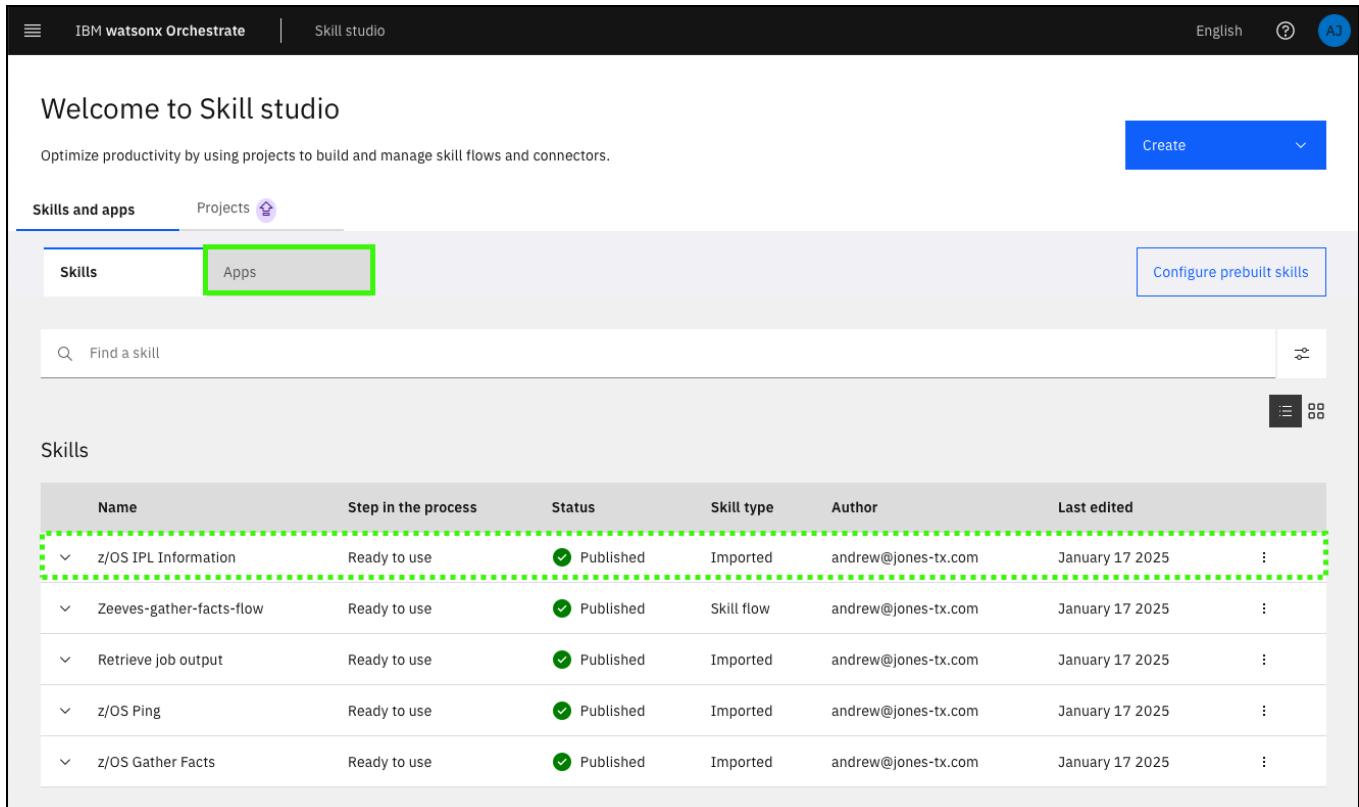
z/OS IPL Information

Cancel Publish Save as draft

10. If you added other skills in step 7, repeat the previous 2 steps for each skill added.

Configure your appURL to connect to the WaaS server

1. In **Skill studio**, click the **Apps** tab.



The screenshot shows the 'Skill studio' section of the IBM Watsonx Orchestrate interface. At the top, there's a navigation bar with 'IBM Watsonx Orchestrate', 'Skill studio', 'English', and a help icon. Below the navigation is a header 'Welcome to Skill studio' with a sub-instruction 'Optimize productivity by using projects to build and manage skill flows and connectors.' To the right of the header is a 'Create' button with a dropdown arrow. Below the header, there are two tabs: 'Skills' and 'Apps', with 'Skills' being the active tab. A green box highlights the 'Apps' tab. To the right of the tabs is a 'Configure prebuilt skills' button. Below the tabs is a search bar with a magnifying glass icon and the placeholder 'Find a skill'. To the right of the search bar is a filter icon. The main area is titled 'Skills' and contains a table with the following data:

Name	Step in the process	Status	Skill type	Author	Last edited
z/OS IPL Information	Ready to use	✓ Published	Imported	andrew@jones-tx.com	January 17 2025
Zeeves-gather-facts-flow	Ready to use	✓ Published	Skill flow	andrew@jones-tx.com	January 17 2025
Retrieve job output	Ready to use	✓ Published	Imported	andrew@jones-tx.com	January 17 2025
z/OS Ping	Ready to use	✓ Published	Imported	andrew@jones-tx.com	January 17 2025
z/OS Gather Facts	Ready to use	✓ Published	Imported	andrew@jones-tx.com	January 17 2025

2. In the search field, enter the name of the application in the search field. Unless you modified the `info` section of the JSON file, the default name is **z/OS operations**.

Welcome to Skill studio

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

Create

Skills and apps Projects

Skills **Apps** Configure prebuilt skills

z/OS operations

Select an app to configure the settings for all the app's skills.

Name	Description	Configuration status
Workday HCM		Not Configured
Salesloft		Not Configured
Adobe Workfront		Not Configured
Webex		Not Configured
ZoomInfo		Not Configured

- Click the ellipses icon (~⋮) for the **z/OS Operations** and then click **Edit**.

Welcome to Skill studio

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

Create

Skills and apps Projects

Skills **Apps** Configure prebuilt skills

z/OS operations

Select an app to configure the settings for all the app's skills.

Name	Description	Configuration status
z/OS operations	z/OS IPL Information	Not Configured

- Click the **Configuration** tab.

Skills and apps / Enhance this skill

Configure the “z/OS operations” app

Details **Configuration**

Change the app's name and icon to whatever makes more sense for your organization. Be sure to provide the connection information.

Name*
z/OS operations

Description
0/100
z/OS IPL Information

App icon

Preview

The app will look like this in the skill set.

z/OS operations
1 skill

The app will look like this in the skill set.

z/OS operations
1 skill

5. Click **Test connection**.

Note: the **Server URL** field must match the URL you created for your WaaS server. It is similar to:

<https://itzvsi-zos-pwgabob.techzone.ibm.com:10443> .

Skills and apps / Enhance this skill

Configure the “z/OS operations” app

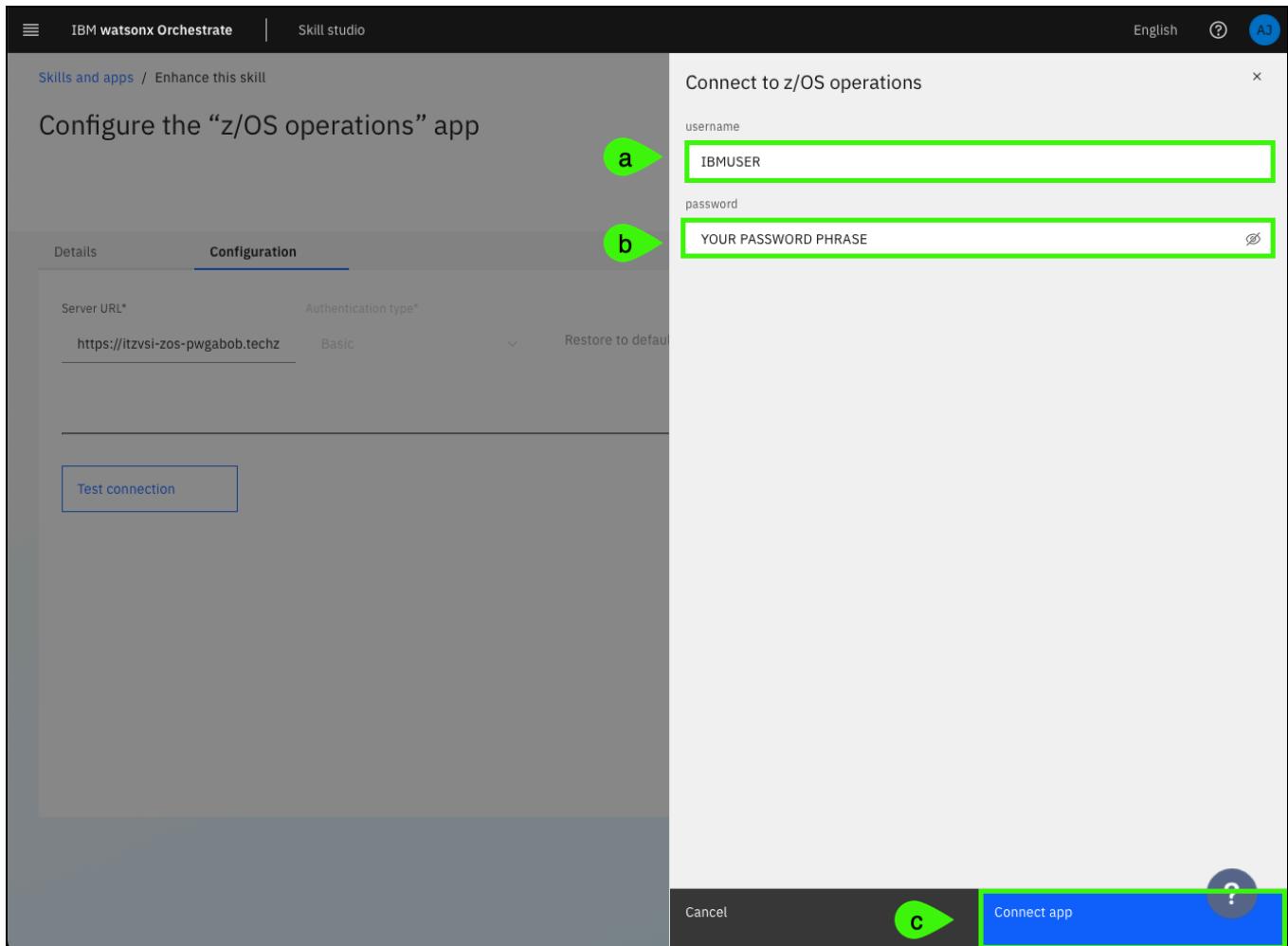
Details **Configuration**

Server URL* https://itzvsi-zos-pwgabob.techz **Authentication type*** Basic

Restore to default ↻

Test connection

6. Enter (a) **IBMUSER** in the **username** field, (b) your **WaaS password** created earlier in the **password** field, and then (c) click **Connect app**.



7. Verify that the connection is successful and then click **Save**.

Skills and apps / Enhance this skill

Configure the “z/OS operations” app

Configuration

Server URL* https://itzvsi-zos-pwgabob.techz Authentication type* Basic Restore to default ↻

Test connection

Connection successful. Valid URL 08:58:14

Cancel Save

8. Repeat step 2 above to verify that the **Configuration status** is **Configured**.

Welcome to Skill studio

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

Skills and apps Projects

Skills Apps

Configure prebuilt skills

z/OS operations

Select an app to configure the settings for all the app's skills.

Name	Description	Configuration status
z/OS operations	z/OS IPL Information	Configured

Test the skill

1. Open **Chat** in watsonx Orchestrate.

The screenshot shows the IBM Watsonx Orchestrate interface. The left sidebar has a dark theme with white text and icons. The 'Chat' option is highlighted with a green border. Other options like 'Skill sets' and 'Skill catalog' are also visible. The main area is titled 'Skill studio' and contains a brief description: 'Build skill flows and connectors to build and manage skill flows and connectors.' A 'Create' button is at the top right. Below it is a 'Configure prebuilt skills' button. A table is present with one row, showing a skill named 'z/OS IPL Information' with a configuration status of 'Configured' and a checkmark icon.

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

The screenshot shows the 'Chat' interface. At the top, there's a message from Watson: 'Hi. I'm Watson. Check out the skills in the catalog to see how I can help you.' Below is a search bar with the placeholder 'Tell me what you want to do'. In the bottom left corner, there's a button labeled 'Add skills from the catalog' with a green border and a blue arrow pointing right. To its right is a card for 'Ansible Controller Skills - z skills', which includes a small icon and a '2 skills' button. A question mark icon is in the bottom right corner.

3. In the search apps field, enter the name of the application. Unless you modified the `info` section of the JSON file, the default name is **z/OS operations**.

The screenshot shows the 'Skill catalog' page in IBM Watsonx Orchestrate. At the top, there's a search bar with the text 'z/OS operations' highlighted by a green box. Below the search bar, a section titled 'Most popular skills' displays three cards: 'Send an email from Gmail', 'Create a lead in Salesforce', and 'Send an email using Outlook'. Underneath this, a section titled 'All Apps' shows a grid of app tiles, each with a logo, name, and skill count. The tiles include Coupa (23 skills), ZoomInfo (26 skills), HubSpot CRM (33 skills), Apptio Targetprocess (24 skills), Salesforce (85 skills), Zendesk Service (25 skills), Calendly (6 skills), and Square (8 skills).

- Locate and click the tile for your app (z/OS operations).

This screenshot shows the same 'Skill catalog' page after the user has located the 'z/OS operations' tile. The tile is now highlighted with a green box. The rest of the interface remains the same, with the search bar containing 'z/OS operations', the 'Most popular skills' section, and the 'All Apps' grid.

5. Click **Add skill +** for the **z/OS IPL Information** skill.

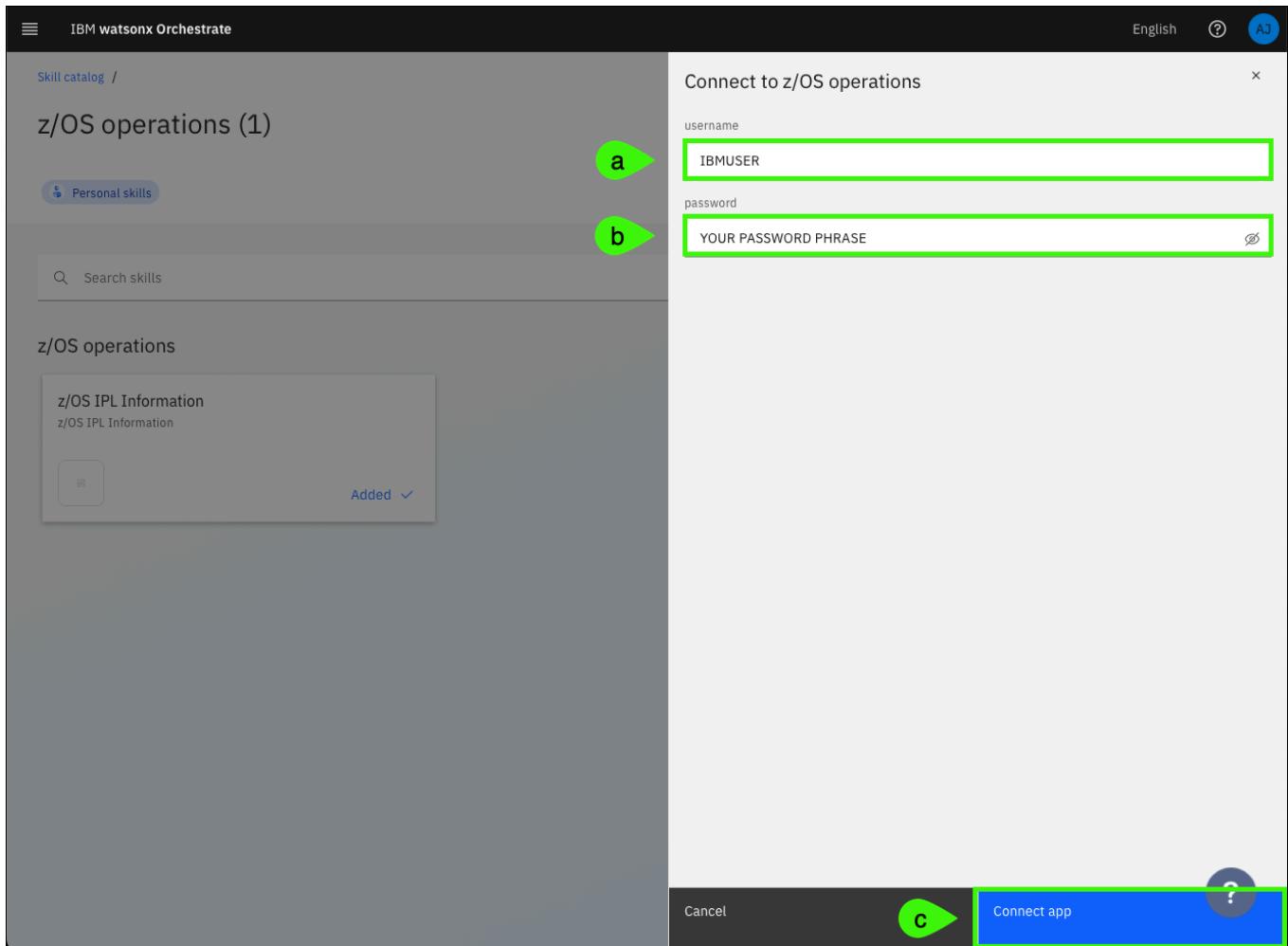
The screenshot shows the IBM Watsonx Orchestrate interface. At the top, there's a navigation bar with 'Skill catalog /' and other options like 'English', '?', and 'AJ'. Below it, the title 'z/OS operations (1)' is displayed. Underneath, there's a search bar with 'Search skills'. The main content area shows a single skill card for 'z/OS IPL Information'. This card has a small icon on the left and a 'Add skill +' button on the right, which is highlighted with a green box. The entire screenshot is set against a light blue background.

6. Repeat step 5 for any additional skills you added to the app.

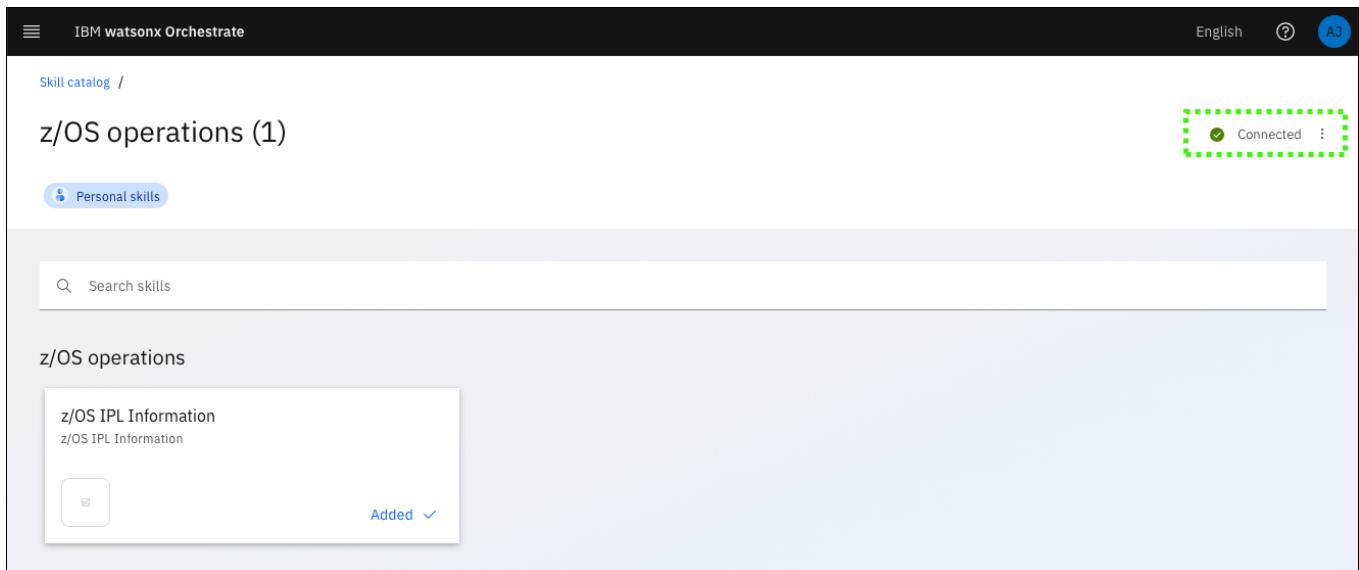
7. Click **Connect app**.

This screenshot is similar to the previous one but shows the result of adding the skill. The 'z/OS IPL Information' skill card now has a green box around its status indicator 'Added ✓'. The 'Connect app' button at the top right is also highlighted with a green box. The rest of the interface remains the same, with the 'Skill catalog /' header and the 'z/OS operations (1)' title.

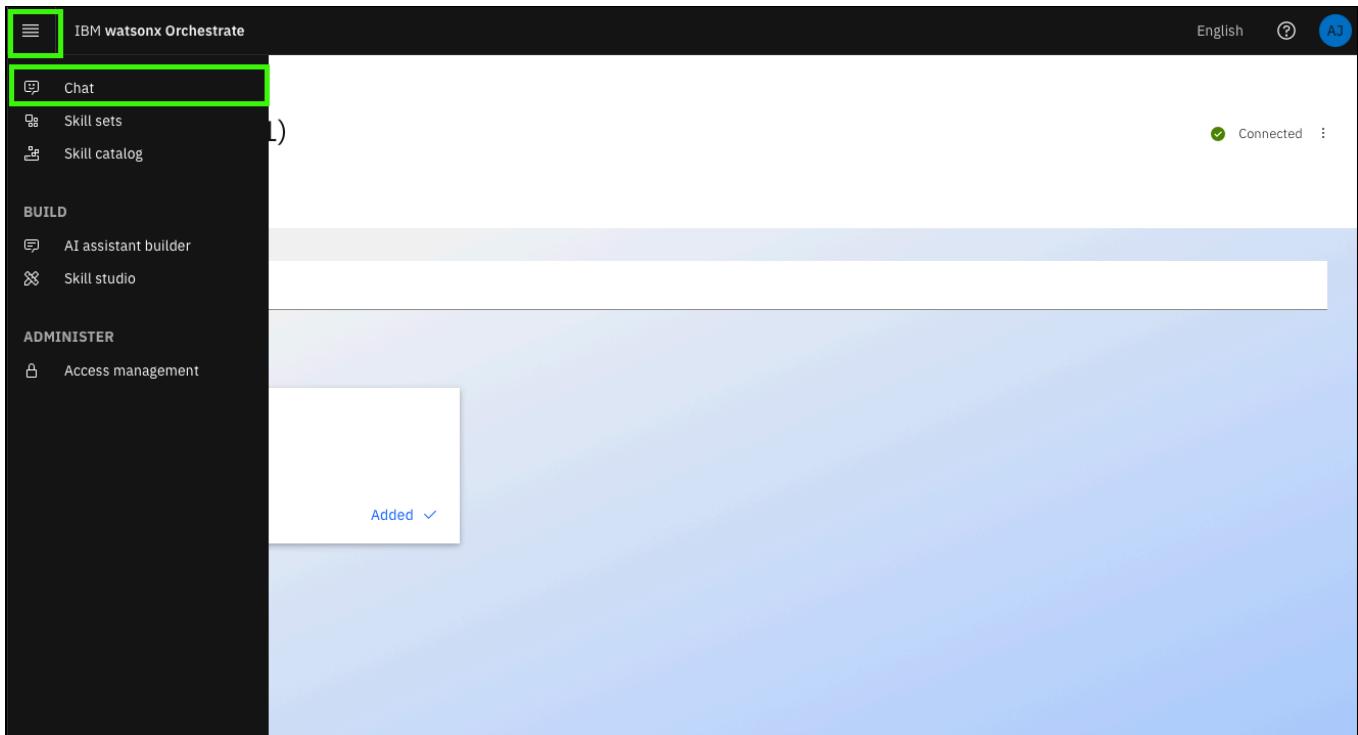
8. Enter (a) **IBMUSER** in the **username** field, (b) your **WaaS password** created earlier in the **password** field, and then (c) click **Connect app**.



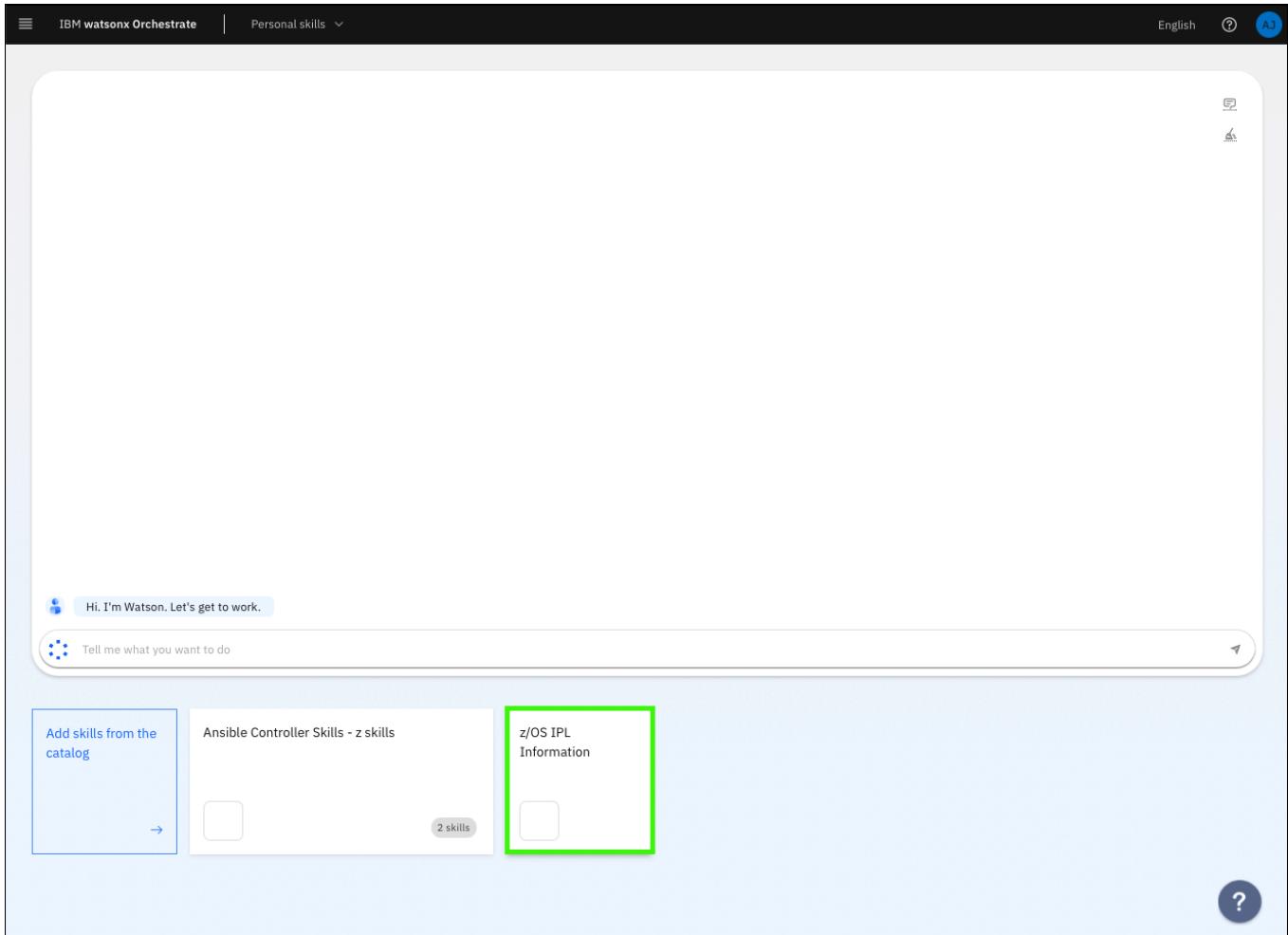
9. Verify that the skill is connected.



10. Open **Chat** in watsonx Orchestrate.



11. Click the **z/OS IPL Information** tile.





The name of the tile may be different.

If you added multiple skills to your app, the tile name may be **z/OS operations** and the number of skills included will be shown.

12. Using the defaults in the form, click **Apply**.

Hi, I'm Watson. Let's get to work.

z/OS IPL Information

You just need to complete this form first.

z/OS IPL Information

consoleName *

cmd *

sol-key *

Cancel **Apply**

Tell me what you want to do

Add skills from the catalog →

Ansible Controller Skills - z skills

z/OS IPL Information

2 skills

13. Review the returned information.

The provided output shows information the early-tenure SysProg needs to prepare for an IPL on their z/OS LPAR. For example, the date and time the system was last IPL'ed, the z/OS release level, the IPL volume, the IPL LOAD PARM used during the IPL, and other details.

If you loaded other pre-packaged skills, test them now.



```

z/OS IPL Information
sol-key-timeout
sol-key-detected
cmd-response
IEE2541 10.14.21 IPLINFO DISPLAY 514
SYSTEM IPLED AT 06.48.48 ON 01/17/2025
RELEASE z/OS 02.05.00 LICENSE = z/OS
USED LOADK2 IN SYS0.IPLPARM ON ODE28
ARCHLVL = 2 MTLSHARE = N
VALIDATED BOOT: NO
IEASYM LIST = (00,K2)
IEASYS LIST = (00) (OP)
IODEVICE: ORIGINAL(ODE28) CURRENT(ODE28)
IPL DEVICE: ORIGINAL(ODE27) CURRENT(ODE27) VOLUME(D25VS1)
VM CPID = zHYPaaS
VM UUID = 3C9BD7F3-8A47-4A47-B62C-298DA45F40EB
VM NAME = k8s_75d1
VM EXT NAME = k8s_75d16281c12c4681b91417749e322845_0727_3c9bd7f3-8a47
-4a47-b62c-298da45f40eb

```

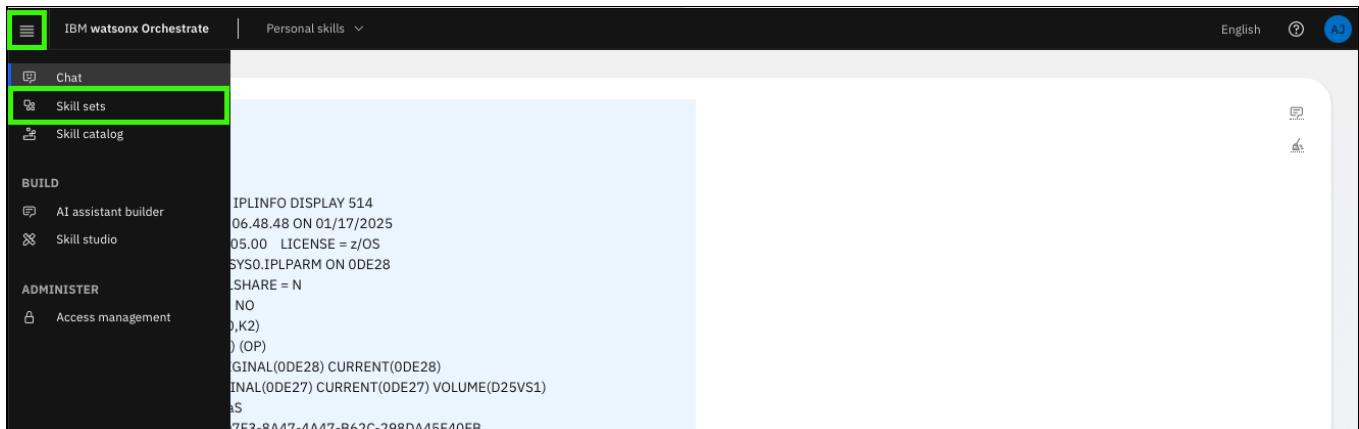
cmd-response key

Tell me what you want to do

Connect the app to your assistant

Before configuring the **z/OS IPL Information** skill as an assistant action, the app containing the skill must first be connected to the assistant.

1. Open Skill sets in watsonx orchestrate.



2. Click the **Skill sets** drop-down list and select the **Draft** of your assistant.

Team Skills

Skill sets

- Team Skills
- Orchestrate Agent Skillset
- Team Skills
- Zeeves draft
- Zeeves live

The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.

Name	App
Zeeves draft	

No skills

Click **Manage skills** to give this team digest specific abilities.

3. Click Connections.

Zeeves draft

Skill sets

- Zeeves draft

Skills (2)

The skills added to this team are available to this team's members. These skills complete related tasks for the team members so members can work more efficiently and effectively.

Name	App
Retrieve job output	
Zeeves-gather-facts-flow	

Items per page: 10 ▾ 1-2 of 2 items 1 of 1 page

4. Enter your skill app name (z/OS operations) in the search bar.

Skill sets

Zeeves draft

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
z/OS operations	1	Not specified	-	⋮

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

5. Click the ellipses icon (~⋮~) for the z/OS operations app and then click **Connect app**.

Skill sets

Zeeves draft

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
z/OS operations	1	Not specified	-	⋮

Items per page: 5 1-1 of 1 items Connect app ⋮

6. In the dialog, click **Connect app**.

Skill sets

Zeeves draft

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
z/OS operations	1	Not specified

Items per page: 5 1-1 of 1 items

Connect to z/OS operations

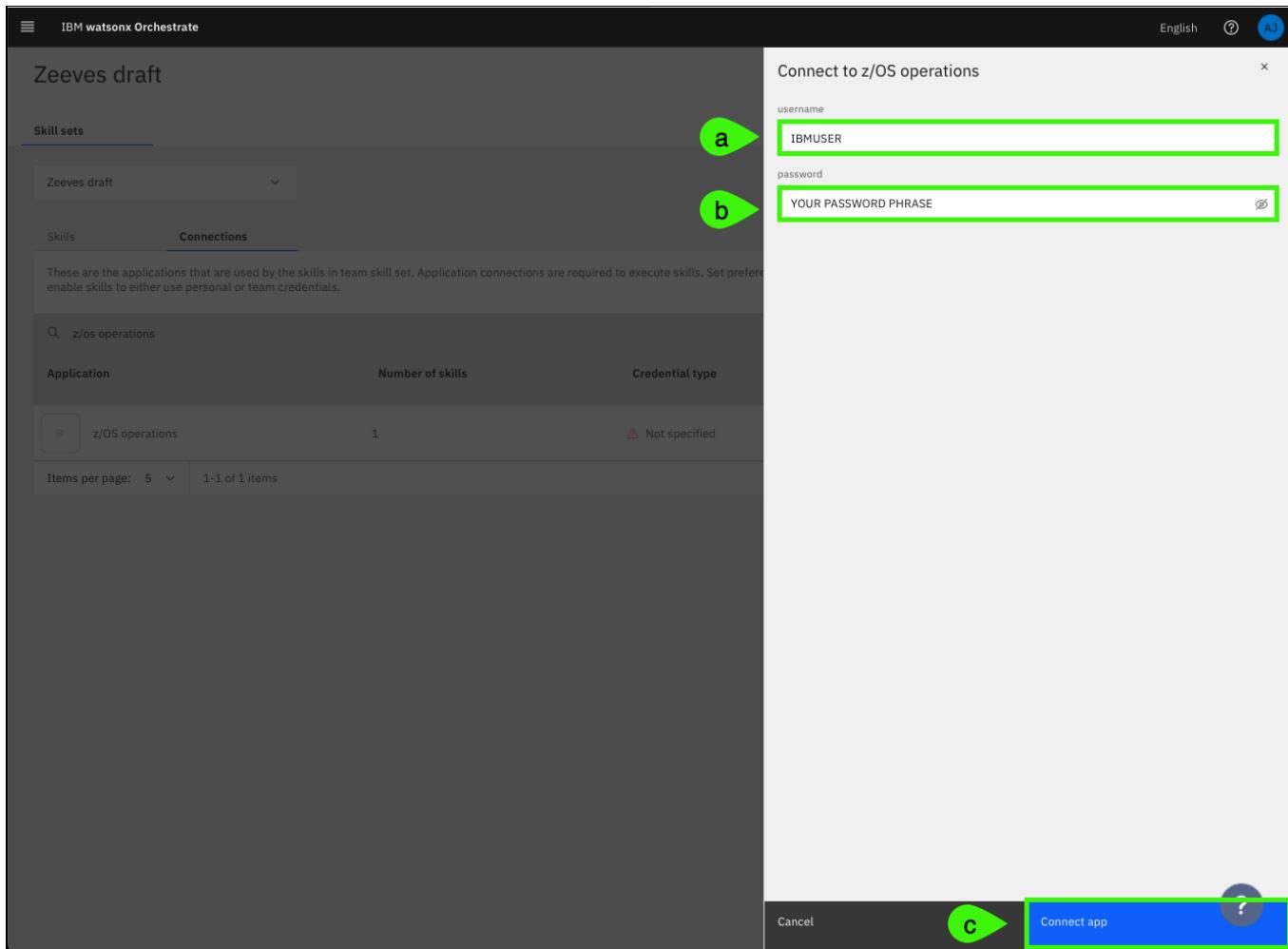
Member credentials
Each team member uses their own credentials to connect to this app and use its skills.

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 (a) **IBMUSER** in the username field, (b) your **WaaS password** in the password field, and then (c) click **Connect app**.



Create an action for your assistant

Next, create a skill-based action that uses the **z/OS IPL Information**. Recall, with the **z/OS Gather Facts** skill flow that was created earlier, adding the skill as a skill-based action allows the skill to run based on user prompts to the assistant.

Skill-based actions also serve another purpose. After creating the skill-based action, you can then call that action from a custom-built action. This is accomplished through subactions. For this use case, create an action that triggers the **z/OS IPL Information** skill on z/OS to display the LPARs IPL information. That action is integrated into another custom-built action (as a subaction) to provide a customized user experience.

1. Open [AI assistant builder](#) in watsonx Orchestrate.

The screenshot shows the IBM Watsonx Orchestrate interface. On the left, there's a navigation sidebar with sections like Chat, Skill sets (which is highlighted with a green box), Skill catalog, BUILD (with AI assistant builder selected), and ADMINISTER (with Access management). The main content area displays a table of skill sets. One row is visible, showing 1 skill, connected by Team, and associated with the email andrew@jones-tx.com. A message at the top states: "Skills are used by the skills in team skill set. Application connections are required to execute skills. Set preference at an application level to use your own credentials or team credentials." At the bottom right of the table, there are pagination controls showing 1 of 1 items.

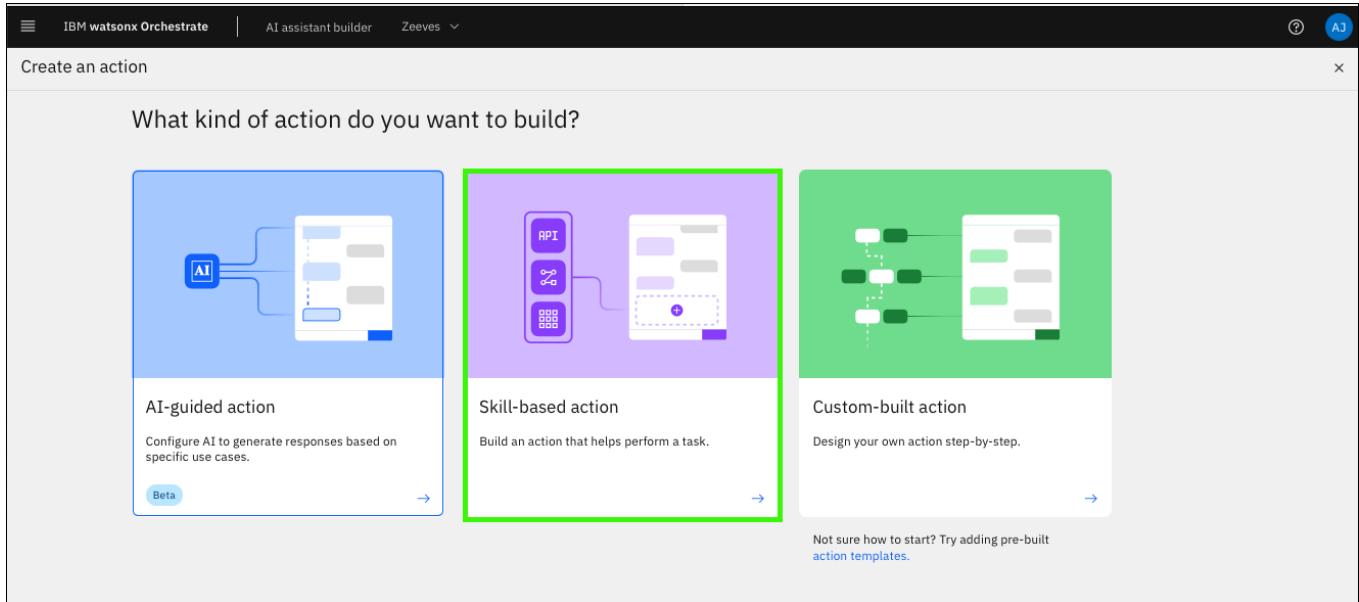
2. Select Actions.

This screenshot shows the AI assistant builder interface under the 'Actions' section. The left sidebar includes options like Home, Generative AI (selected), Actions (highlighted with a green box), Evaluate, Preview, Deploy, Publish, Environments, Improve, and Analyze. The main area displays several actions: Build actions, Customize your greeting, Create a fallback plan, Preview & debug, Customize web chat, and Set. Each action has a brief description and a small icon. A message at the top right says "Enable in settings".

3. Click New action +.

This screenshot shows the Actions list interface. The left sidebar lists filters: All items, Created by you (highlighted with a green box), Set by assistant, Variables, Created by you, Set by assistant, Set by integration, and Saved responses. The main area is a table of actions. One row is visible, showing a name (Zeeves-gather-facts-flow), last edited (2 days ago), examples count (1), steps count (0), and status (green checkmark). At the bottom right of the table, there is a blue button labeled "New action +" with a green box around it.

4. Click Skill-based action.



5. Click **z/OS IPL Information** and then click **Next**.

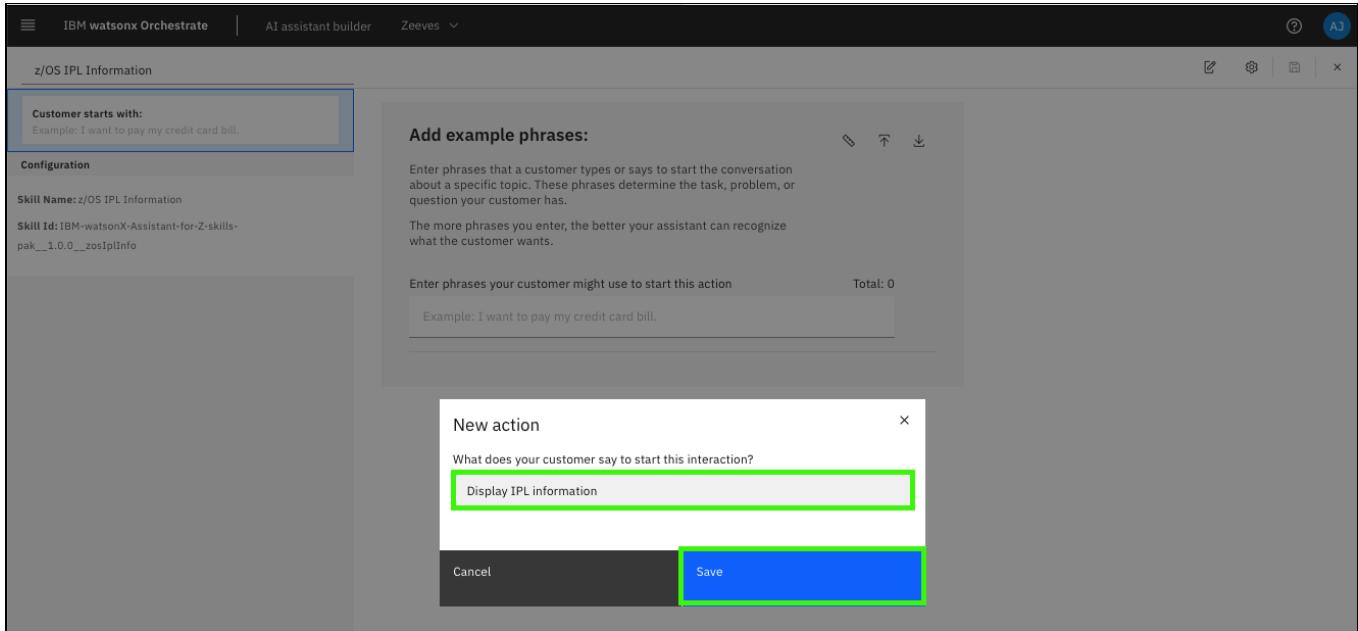
The screenshot shows the 'Build an action from a skill' screen. At the top, it says 'Select a skill' and 'Choose a conversational skill published as a foundation of your action.' A search bar contains the text 'z/OS IPL Information'. Below the search bar is a grid of skill cards:

z/OS IPL Information	Zeeves-gather-facts-flow	Retrieve job output	z/OS Ping
z/OS IPL Information	Run z/OS gather facts and retrieve the output.	z skills - Retrieve job output by job Id	z skills - This playbook pings the z/OS host to test connectivity.
Last updated: 2025-01-17T14:39:36.042Z	Last updated: 2025-01-17T12:24:16.672Z	Last updated: 2025-01-17T12:18:14.714Z	Last updated: 2025-01-17T12:13:33.194Z
z/OS Gather Facts	Summarize the Webex meeting transcript	Summarize the Box content	Summarize a Zendesk ticket
z skills - This sample playbook demonstrates the z/OS gather facts module, which pulls z/OS-specific information from the z/OS host.	in watsonx.ai	in watsonx.ai	in watsonx.ai

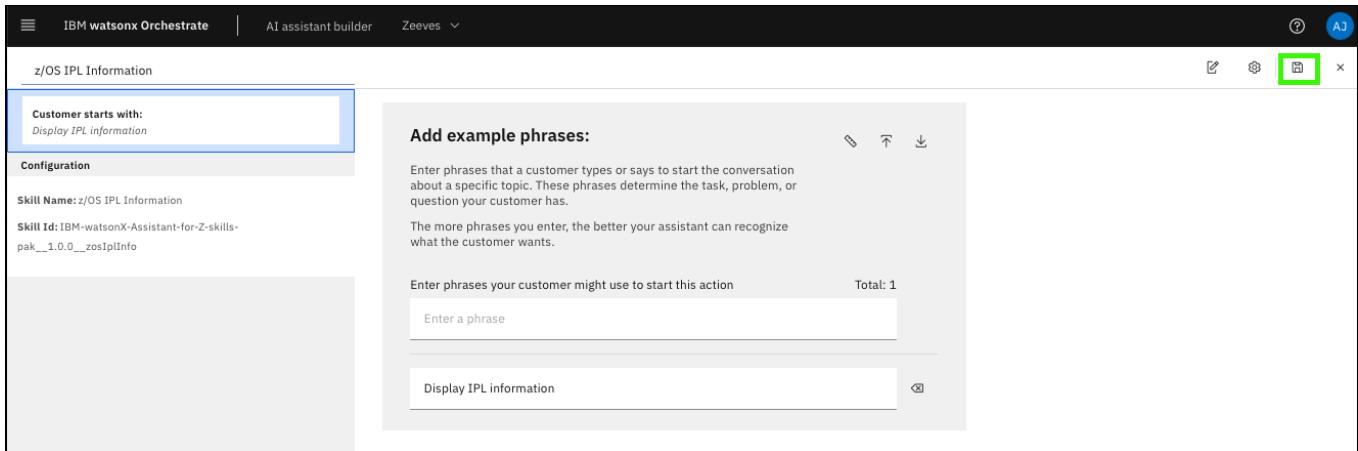
6. Enter a prompt like **Display IPL information** that starts the skill and then click **Save**.

Prompt:

Display IPL information



7. Click the **Save** icon (💾).



8. Click **Preview**.

The screenshot shows the IBM WatsonX Assistant builder interface. On the left, there's a sidebar with 'IBM WatsonX Orchestrate' and 'AI assistant builder' tabs, and a dropdown for 'Zeeves'. The main area is titled 'z/OS IPL Information'. Under 'Configuration', it shows 'Skill Name: z/OS IPL Information' and 'Skill Id: IBM-watsonX-Assistant-for-Z-skills-pak_1.0.0_zosIplInfo'. A large blue box highlights the 'Customer starts with:' section, which contains 'Display IPL information'. To the right, under 'Add example phrases:', it says '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.' Below this, it says 'The more phrases you enter, the better your assistant can recognize what the customer wants.' There's a text input field with 'Enter a phrase' placeholder and a list item 'Display IPL information'. At the bottom right of the configuration panel, there's a 'Preview' button with a question mark icon, which is also highlighted with a green box.



Wait for the Your changes are being added. message to disappear before proceeding.

9. Enter the prompt (**Display IPL information**) to test the skill.

z/OS IPL Information

Customer starts with:
Display IPL information

Configuration

Skill Name: z/OS IPL Information

Skill Id: IBM-watsonX-Assistant-for-Z-skills-pak_1.0.0_zosIplInfo

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

Display IPL information

Preview

8:46 AM
Greet customer [default]

Welcome, how can I assist you?

Display IPL information

10. Using the defaults in the form, click **Apply**.

The screenshot shows the IBM WatsonX Assistant for Z AI assistant builder interface. The top navigation bar includes 'IBM WatsonX Orchestrate', 'AI assistant builder', 'Zeeves', and a user icon. The main area is titled 'z/OS IPL Information'. On the left, there's a sidebar with 'Customer starts with:' and 'Display IPL information' under 'Add example phrases:'. Below this is a 'Configuration' section with 'Skill Name: z/OS IPL Information' and 'Skill Id: IBM-watsonX-Assistant-for-Z-skills-pak_1.0.0_zosIplInfo'. The right side features a 'Preview' window showing a simulated conversation:

9:02 AM
Conversational skill called
[z/OS IPL Information](#) recognized

z/OS IPL Information

consoleName *
iserVS01

cmd *
d iplinfo

sol-key *
JES

Cancel **Apply**

11. Verify the results.

Customer starts with:
Display IPL information

Configuration

Skill Name: z/OS IPL Information
Skill Id: IBM-watsonX-Assistant-for-Z-skills-pak_1.0.0_zosIplInfo

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

Display IPL information

Preview

z/OS IPL Information

cmd-response
IEE254I 11.04.51 IPLINFO DISPLAY
SYSTEM IPLED AT 06.48.48 ON 01/1
RELEASE z/OS 02.05.00 LICENSE =
USED LOADK2 IN SYS0.IPLPARM ON
ARCHLVL = 2 MTLSHARE = N
VALIDATED BOOT: NO
IEASYM LIST = (00,K2)
IEASYS LIST = (00) (OP)
IODF DEVICE: ORIGINAL(ODE28) CUI
IPL DEVICE: ORIGINAL(ODE27) CURF
VM CPID = zHYPaa5
VM UUID = 3C9BD7F3-8A47-
4A47-B62C-298DA45F40EB
VM NAME = k8s_75d1
VM EXT NAME = k8s_75d16281c12c
8a47

12. Click the **delete** icon (✖) to remove the example phrase (**Display IPL information**) from the skill.

Next, you will add this skill-based action as a subaction to a custom-build action. To have the custom-built action started rather than the skill-based action, the existing *example phrases* need to be removed.

Customer starts with:
Display IPL information

Configuration

Skill Name: z/OS IPL Information
Skill Id: IBM-watsonX-Assistant-for-Z-skills-pak_1.0.0_zosIplInfo

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

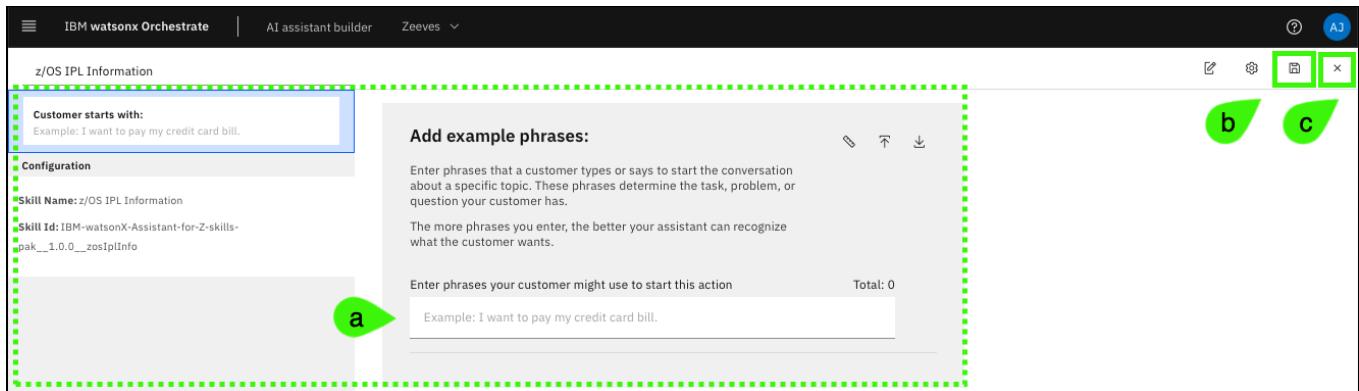
Display IPL information

Preview

z/OS IPL Information

cmd-response
IEE254I 11.04.51 IPLINFO DISPLAY
SYSTEM IPLED AT 06.48.48 ON 01/1
RELEASE z/OS 02.05.00 LICENSE =
USED LOADK2 IN SYS0.IPLPARM ON
ARCHLVL = 2 MTLSHARE = N
VALIDATED BOOT: NO
IEASYM LIST = (00,K2)
IEASYS LIST = (00) (OP)
IODF DEVICE: ORIGINAL(ODE28) CUI
IPL DEVICE: ORIGINAL(ODE27) CURF
VM CPID = zHYPaa5
VM UUID = 3C9BD7F3-8A47-
4A47-B62C-298DA45F40EB
VM NAME = k8s_75d1
VM EXT NAME = k8s_75d16281c12c
8a47

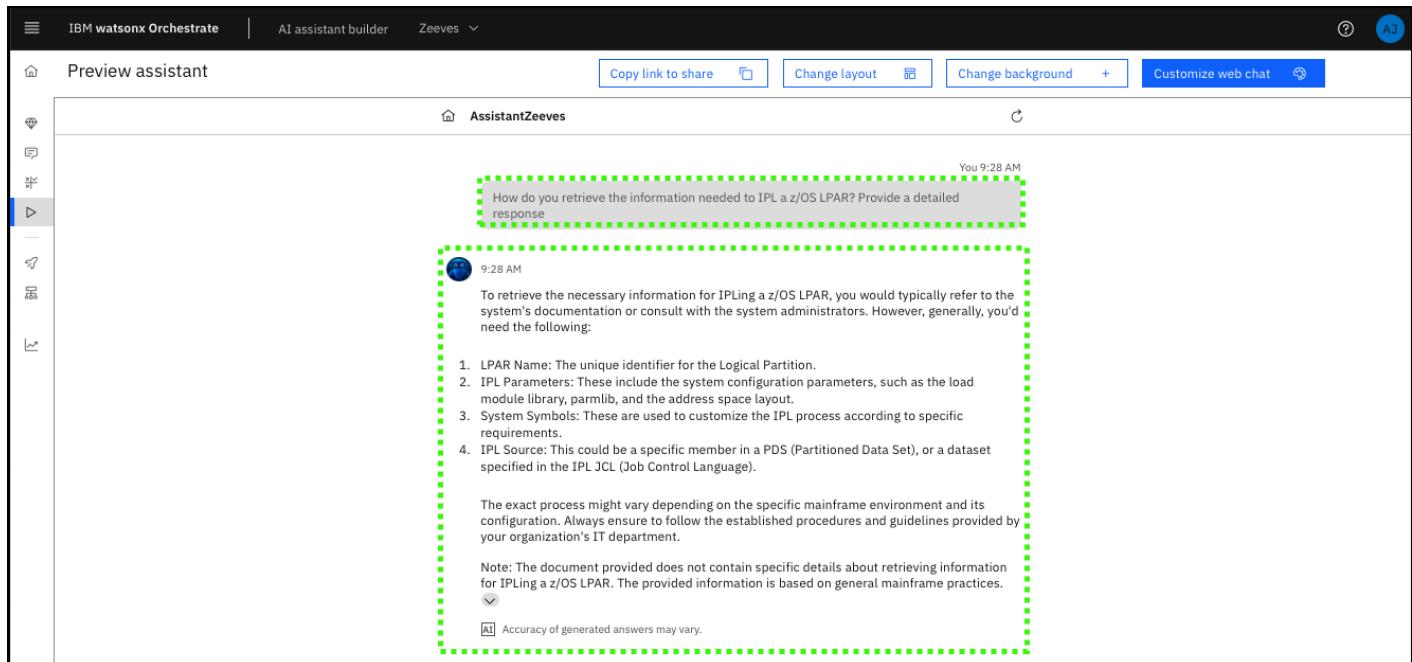
13. Verify (a) all **example phrases** are deleted, (b) click the **Save** icon (💾), and then (c) click **x** to close the action.



Create a custom-built action

Custom-built actions allow you to define each step of a conversation with your assistant. You can define steps with or without conditions to control the user responses. You can include your skill-based actions as sub-actions. You can pass generated values between the subactions.

Recall the use case of an early-tenure SysProg who is preparing for an upcoming IPL on one of their LPARs. After going through documentation, the SysProg found that there is information that is needed about their LPAR before they can begin the IPL process. To gather that information, the SysProg might ask the assistant "How do you retrieve the information needed to IPL a z/OS LPAR? Provide a detailed response". A response the assistant might return is shown in the following image.



A custom-built action can help the SysProg to take the next steps that are required by adding next-step action suggestions and start skills to gather the needed information.

1. Click New action +.

The screenshot shows the 'Actions' section of the IBM Watsonx Orchestrate interface. On the left, there's a sidebar with categories like 'All items', 'Created by you', 'Variables', and 'Saved responses'. The main area lists actions under 'Created by you': 'z/OS IPL Information' (last edited 17 minutes ago, 0 examples, 3 steps, status green) and 'Zeeves-gather-facts-flow' (last edited 2 days ago, 1 example, 0 steps, status green). At the top right, there's a 'New action' button.

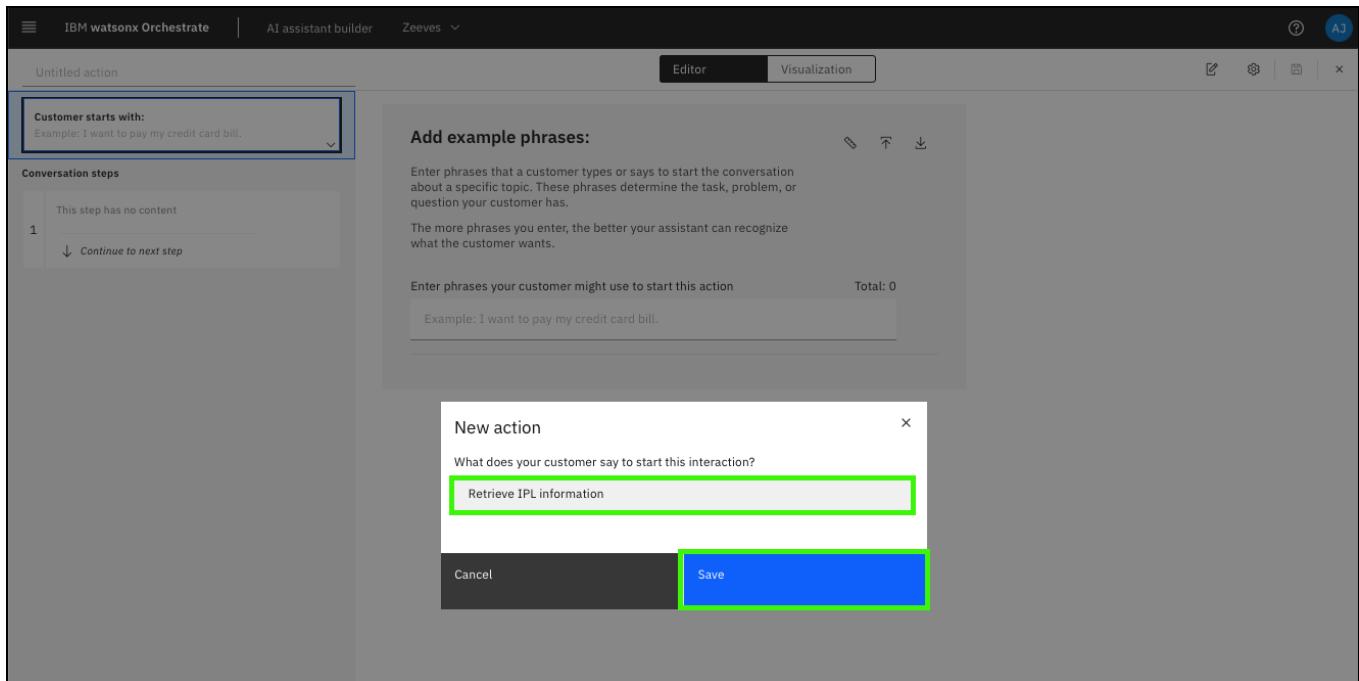
2. Click Custom-built action.

The screenshot shows the 'Create an action' dialog. It asks 'What kind of action do you want to build?' and lists three options: 'AI-guided action' (blue box), 'Skill-based action' (purple box), and 'Custom-built action' (green box, highlighted with a green border). Each option has a description and a 'Beta' button. Below the boxes is a note: 'Not sure how to start? Try adding pre-built action templates.'

3. Enter an example phrase (**Retrieve IPL information**) to prompt the assistant to start the custom-built action and click **Save**.

Prompt:

Retrieve IPL information



4. Review the form to create a custom-built action.

A custom-built action can consist of multiple steps with each step that is taken with or without conditions. Each step specifies the assistant's response and the next step to start.



For client demonstrations and pilots...

For client demonstrations and pilots, it is possible to add a custom search query such that when conversational search is executed in the 1st step, the query being used is hard-coded and isn't necessarily what the end-user inputted. This can be done by clicking on the Edit settings option under Search for the answer and specifying a custom search query that specifies the exact query to be used.

5. Click the **And then** drop-down list and select **Search for the answer**.

Customer starts with:
Retrieve IPL information

Conversation steps

Step 1

Is taken without conditions Set variable values

Assistant says

For example: Please select from the following options:

Define customer response

And then

Continue to next step
Re-ask previous step(s)
Go to a subaction
Use an extension
Search for the answer
Connect to agent
End the action

6. Click **Edit settings**.

Customer starts with:
Retrieve IPL information

Conversation steps

Step 1

Is taken without conditions Set variable values

Assistant says

For example: Please select from the following options:

Define customer response

And then

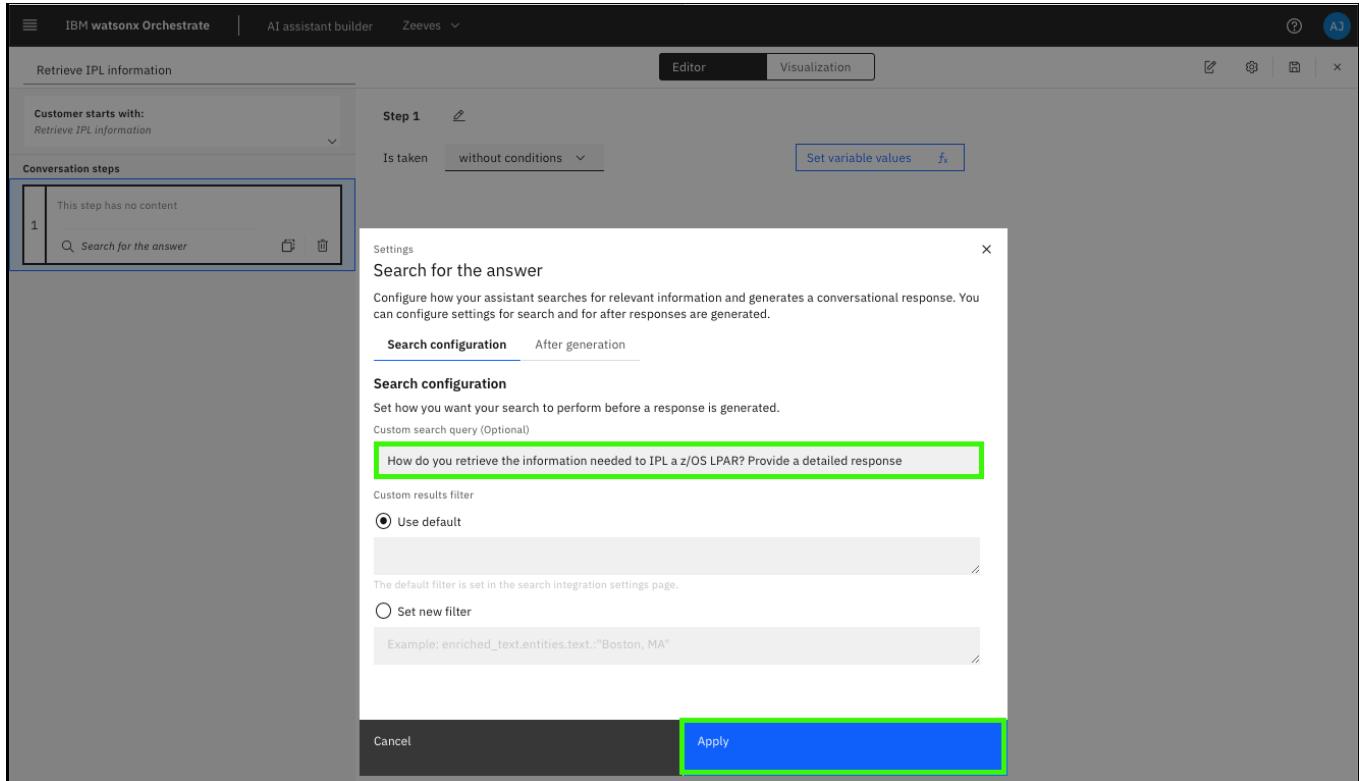
Search for the answer

Custom query	None (Customer's message will be the query)
Custom filter	None (Optional)
Edit settings	

7. Enter a **Custom search query** and click **Apply**.

Custom search query:

How do you retrieve the information needed to IPL a z/OS LPAR? Provide a detailed response



8. Click Next step +.

The new custom-built action only responds with a description of how to retrieve the IPL information. Now, infuse automation into the conversation and ask the user if they want to perform the action.

The screenshot shows the IBM Watsonx Orchestrate AI assistant builder interface. The top navigation bar includes 'IBM Watsonx Orchestrate', 'AI assistant builder', 'Zeeves', and a user icon. The main area is titled 'Retrieve IPL information'. On the left, under 'Conversation steps', there is one step labeled '1' with the placeholder 'This step has no content' and a search bar 'Search for the answer'. In the center, 'Step 1' is selected, with the condition 'Is taken without conditions'. Below it, the 'Assistant says' section contains a rich text editor with placeholder text: 'For example: Please select from the following options:' and a dropdown menu 'Define customer response'. Under 'And then', another step is defined with the action 'Search for the answer', a custom query ('How do you retrieve the information needed to IPL a z/OS LPAR? Provide a detailed response'), and a custom filter ('None (Optional)'). A green box highlights the 'New step +' button at the bottom left.

9. In the **Assistant says** field, enter a response asking if the user wants to display the IPL information.

Assistant says:

Would you like to display your LPARs IPL information?

The screenshot shows the continuation of the AI assistant builder interface. The top navigation bar and title 'Retrieve IPL information' remain the same. The conversation now includes two steps: '1' and '2'. Step 1 is identical to the previous screenshot. Step 2 starts with the question 'Would you like to display your LPARs IPL information?'. This question is highlighted with a green dashed box. Below it, the 'Assistant says' section contains a rich text editor with the same question, and a dropdown menu 'Define customer response'. Under 'And then', a step is defined with the action 'Continue to next step'. A green box highlights the 'Continue to next step' link.

10. Click the **Define customer response** drop-down and select **Confirmation**.

Now, after providing a conversational search-based response to the original question, the assistant asks the user if they want to display the IPL information and prompts the user to select **Yes** or **No**.

The screenshot shows the IBM Watsonx Orchestrate AI assistant builder interface. At the top, it says "IBM Watsonx Orchestrate" and "AI assistant builder Zeeves". Below that, the title is "Retrieve IPL information". The interface has tabs "Editor" and "Visualization", with "Editor" selected.

Conversation steps:

- Step 1: "Customer starts with: Retrieve IPL information". It says "This step has no content".
- Step 2: "Is taken without conditions". It has a "Set variable values" button.

Assistant says:

"Would you like to display your LPARs IPL information?"

Define customer response:

System: Options (highlighted with a green border)

Options: Enable customers to select from a set of choices. If more than 4, options show as a list.

Confirmation: Checked (highlighted with a green border)

As buttons: Buttons labeled "Checking", "Savings", "401 (k)", and "Roth IRA".

As a list: List item "Pay Bill".

11. Click **Next step +**.

The screenshot shows the IBM Watsonx Orchestrate interface. In the top navigation bar, it says "IBM watsonx Orchestrate" and "AI assistant builder". The main area is titled "Retrieve IPL information". On the left, there's a sidebar for "Conversation steps" with two steps defined: Step 1 (This step has no content) and Step 2 (Would you like to display your LPARs IPL information?). Step 2 includes a "Confirmation" button and a "Continue to next step" link. The main workspace is titled "Step 2" and contains the following configuration:

- Is taken:** A dropdown menu set to "without conditions".
- Assistant says:** A text input field containing "Would you like to display your LPARs IPL information?". Below it are "Yes" and "No" buttons.
- View response** and **Edit validation** buttons.
- And then:** A dropdown menu set to "Continue to next step".

12. Click the **Is taken** drop-down and select **with conditions**.

The screenshot shows the same interface after Step 2 has been modified. The "Is taken" dropdown in Step 3 now has "with conditions" selected, while "without conditions" is highlighted. The rest of the configuration for Step 3 remains the same as in the previous screenshot.

13. Review the **Conditions**.

Notice that the default condition is based on **Step 2** and the user clicking **Yes** when prompted.

The screenshot shows the IBM Watsonx Orchestrate AI assistant builder interface. On the left, the "Conversation steps" pane displays three steps:

- Step 1: "This step has no content". A search bar "Q Search for the answer" is present.
- Step 2: "Would you like to display your LPARs IPL information?". A "Confirmation" button is shown below the question.
- Step 3: "2 is Yes". This step also has no content.

In the center, the "Step 3" configuration pane is open. It shows the condition "Is taken with conditions" and a green-dashed box containing the condition "If All of this is true: 2. Would you like to d... is Yes". Below this is a "New condition group +".

On the right, the "Assistant says" pane contains the text "For example: What size do you want to order?" and a dropdown menu set to "Define customer response".

14. Enter **Retrieving your system's IPL information...** in the **Assistant says** field.

Assistant says:

Retrieving your system's IPL information...

The screenshot shows the same interface as above, but the "Assistant says" field has been updated. The text "Retrieving your system's IPL information..." is now displayed in the text area, and it is highlighted with a thick green border. The rest of the interface remains the same, including the conversation steps and the step configuration pane.

15. Click the **And then** drop-down and select **Go to a subaction**.

The screenshot shows the IBM Watsonx Orchestrate interface. In the top navigation bar, it says "IBM watsonx Orchestrate" and "AI assistant builder". The main area is titled "Retrieve IPL information". On the left, there's a sidebar for "Conversation steps" with three steps defined:

1. "Search for the answer" (with a "Confirmation" button).
2. "Would you like to display your LPARs IPL information?" (with a "Confirmation" button).
3. "Retrieving your system's IPL information..." (with a "Continue to next step" button).

In the center, under "Step 3", the condition "Is taken with conditions" is set to "If All of this is true:". The condition "2. Would you like to d... is Yes" is selected. Below this, there's a "New condition group +" button.

Under "Assistant says", there's a text input field containing "Retrieving your system's IPL information...".

On the right, there's a "Define customer response" section with a dropdown menu. The "And then" options are listed, and the "Continue to next step" option is highlighted with a green box. A tooltip for "Continue to next step" says "Follow the action's flow to whichever step is ordered next." Other options include "Re-ask previous step(s)", "Go to a subaction" (which is also highlighted with a green box), "Use an extension", "Search for the answer", "Connect to agent", and "End the action".

At the bottom right, there's a "Preview" button and a question mark icon.

16. Click (a) the Go to drop-down, (b) select your skill-based action ([z/OS IPL Information](#)), and then (c) click **Apply**.

This screenshot shows the same interface as the previous one, but with a modal dialog box open over it. The dialog is titled "Go to a action" and has a dropdown menu labeled "Go to". The item "z/OS IPL Information" is selected and highlighted with a green box. There is also an unchecked checkbox "End this action after the b action is completed". At the bottom of the dialog, there are "Cancel" and "Apply" buttons, with "Apply" being highlighted with a blue box.

17. Click **Save** (💾) and then click **x**.

The screenshot shows the IBM Watsonx Orchestrate interface. On the left, there's a sidebar with 'Customer starts with: Retrieve IPL information' and a 'Conversation steps' section. Step 1: 'This step has no content'. Step 2: 'Would you like to display your LPARS IPL information?' with a 'Confirmation' button. Step 3: 'Retrieving your system's IPL information...' with a link to 'z/OS IPL Information'. The main area shows 'Step 3' with a condition 'Is taken with conditions' set to 'with conditions'. A condition is defined: 'If All of this is true: 2. Would you like to d... is Yes'. Below this, there's a 'New condition group +' button and an 'Assistant says' section with rich text editor icons.

Test the custom-built action

You can now practice demonstrating the flow of this use case. Recall the steps that were taken and the scenario of the early-tenure SysProg being tasked with preparing for an upcoming IPL.

1. Click Preview.

The screenshot shows the 'Actions' menu in the sidebar with 'Preview' highlighted. The main area displays a table of actions created by the user:

Name	Last edited	Examples count	Steps count	Status
z/OS IPL Information	an hour ago	0	3	Green checkmark
Zeeves-gather-facts-flow	2 days ago	1	0	Green checkmark
Retrieve IPL information	15 minutes ago	1	3	Green checkmark

2. Enter How do you retrieve the information needed to IPL a z/OS LPAR? in the assistant prompt.

Prompt:

How do you retrieve the information needed to IPL a z/OS LPAR?



Use the Change layout option to open a full page view of the assistant.

The screenshot shows the 'Preview assistant' view in the IBM Watsonx Orchestrate interface. The top navigation bar includes 'IBM Watsonx Orchestrate', 'AI assistant builder', 'Zeeves', and a help icon. Below the navigation is a toolbar with 'Copy link to share', 'Change layout' (which is highlighted with a green dotted border), 'Change background', and 'Customize web chat'. The main area displays the AI assistant's profile picture (an owl) and its greeting: 'Hi! I'm Zeeves. How can I help you today?'. At the bottom, a message card contains the question 'How do you retrieve the information needed to IPL a z/OS LPAR?' followed by a blue '▶' button.

3. Review the response.

By using the conversational search capability of Watsonx Assistant for Z, grounded on Z-domain knowledge in the Z RAG, the user is able to quickly find an answer to their question. The assistant returns a response that shows the exact operator command they might use, and the types of details it provides which is relevant for preparing for an upcoming IPL. For example, the system's release level, the LOAD LIB information used for the IPL, the IODF device, the IPL device, and other system information.

You are also able to improve their productivity by infusing automations into the natural conversation that will provide them exactly the information they need to accomplish their task.

How do you retrieve the information needed to IPL a z/OS LPAR?

To retrieve the necessary information for IPLing a z/OS LPAR, you would typically refer to the system's documentation or consult with the system administrators. However, generally, you'd need the following:

1. LPAR Name: The unique identifier for the Logical Partition.
2. IPL Parameters: These include the system configuration parameters, such as the load module library, parmlib, and the address space layout.
3. System Symbols: These are used to customize the IPL process according to specific requirements.
4. IPL Source: This could be a specific member in a PDS (Partitioned Data Set), or a dataset specified in the IPL JCL (Job Control Language).

The exact process might vary depending on the specific mainframe environment and its configuration. Always ensure to follow the established procedures and guidelines provided by your organization's IT department.

Note: The document provided does not contain specific details about retrieving information for IPLing a z/OS LPAR. The provided information is based on general mainframe practices.

Would you like to display your LPARs IPL information?

4. Click Yes.

How do you retrieve the information needed to IPL a z/OS LPAR?

To retrieve the necessary information for IPLing a z/OS LPAR, you would typically refer to the system's documentation or consult with the system administrators. However, generally, you'd need the following:

1. LPAR Name: The unique identifier for the Logical Partition.
2. IPL Parameters: These include the system configuration parameters, such as the load module library, parmlib, and the address space layout.
3. System Symbols: These are used to customize the IPL process according to specific requirements.
4. IPL Source: This could be a specific member in a PDS (Partitioned Data Set), or a dataset specified in the IPL JCL (Job Control Language).

The exact process might vary depending on the specific mainframe environment and its configuration. Always ensure to follow the established procedures and guidelines provided by your organization's IT department.

Note: The document provided does not contain specific details about retrieving information for IPLing a z/OS LPAR. The provided information is based on general mainframe practices.

Would you like to display your LPARs IPL information?

5. Using the defaults in the form, click Apply.

The screenshot shows the IBM Watsonx Orchestrate interface with the AI assistant builder tab selected. The main area displays an AI-generated response titled "AssistantZeeves". The response discusses retrieving information from the IPL JCL and provides a note about the general nature of the information. It asks if the user would like to display their LPAR's IPL information, with "Yes" and "No" buttons. A timestamp "You 10:37 AM" and a "Yes" button are visible. Below this, a modal window titled "z/OS IPL Information" is shown, containing fields for "consoleName" (set to "iservS01"), "cmd" (set to "d iplinfo"), and "sol-key" (set to "JES"). The "Apply" button at the bottom right of the modal is highlighted with a green border.

6. Review the response to the custom-built action.

The output provides relevant pieces of information the early-tenure SysProg needs to prepare for an IPL on their z/OS LPAR. For example, when the system was last IPLed, the z/OS release they have running (2.5), the IPL volume (D25VS1), the IPL LOAD PARM (LOADK2) used during the last IPL, and other relevant details.

The screenshot shows the IBM Watsonx Assistant interface. At the top, there are tabs for "IBM Watsonx Orchestrate", "AI assistant builder", and "Zeeves". On the right side of the header are buttons for "Copy link to share", "Change layout", "Change background", and "Customize web chat". The main content area is titled "AssistantZeeves". It displays a list of items with checkboxes:

- z/OS IPL Information
- sol-key-timeout
- sol-key-detected

Below this is a "cmd-response" section containing the following text:

```

cmd-response
IEE254I 12.40.03 IPLINFO DISPLAY 770
SYSTEM IPLED AT 06.48.48 ON 01/17/2025
RELEASE z/OS 02.05.00 LICENSE = z/OS
USED LOADK2 IN SYS0.IPLPARM ON ODE28
ARCHLVL = 2 MTLSHARE = N
VALIDATED BOOT: NO
IEASYM LIST = (00,K2)
IEASYS LIST = (00) (OP)
IODF DEVICE: ORIGINAL(ODE28) CURRENT(ODE28)
IPL DEVICE: ORIGINAL(ODE27) CURRENT(ODE27) VOLUME(D25VS1)
VM CPID = zHYPaaS
VM UUID = 3C9BD7F3-8A47-4A47-B62C-298DA45F40EB
VM NAME = k8s_75d16281c12c4681b91417749e322845_0727_3c9bd
8a47
        -4a47-b62c-298da45f40eb

cmd-response-key
C4443879

```

At the bottom of the content area, there is a timestamp "10:39 AM" and a text input field with placeholder text "Type something...".

The ability to infuse skills and automations into the conversation allows the user to issue the operator command on their system directly within the assistant itself. The assistant provides a single interface for retrieving details that are required for the IPL process to go smoothly and increases the user's efficiency.

COMING SOON!

COMING SOON!

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 under 'Build'. The main area displays a 'Build actions' card and a 'Zeeves' integration diagram. The 'Integrations' section is highlighted with a green border. Below it are 'Activity log' and 'Assistant settings'.

Integrations section details:

- Zeeves** integration diagram:
 - Default behavior branches to:
 - General purpose (granite-3-8b-instruct)
 - Conversational search (Custom service)
 - Search
 - Actions View > branches to:
 - AI-guided (0 actions)
 - Skill-based (1 action)
 - Custom-built (0 actions)
 - Fallback branches to:
 - Live agent (No agent)

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

The screenshot shows the 'Integrations' section with a green dashed box around the 'Essential channels' and 'Channels' sections.

Essential channels section (highlighted by a green dashed box):

- Web chat**: Built by IBM - Lite. Embed the web chat onto your company website so it can answer questions. Contact center integrations: NICE, Genesys, Twilio.

Channels section (highlighted by a green dashed box):

- Deploy your assistant to third-party channels to expand your reach.
- Icons for SMS, Facebook messenger, Genesys Bot Connector, Slack, Microsoft teams, and WhatsApp with Twilio, each with an 'Add' button.

Extensions section (highlighted by a green dashed box):

- Icons for Web browser and Cloud integration, 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, Genesys, Twilio

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 + (highlighted with a green box)	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

Slack
Let your assistant respond directly in Slack. Add your assistant as a bot user to your Slack app, answering questions from direct messages or channels.

Cancel **Add** (highlighted with a green box)

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 + (highlighted with a green dashed box)	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 box)
Slack bot Connect Slack Configure Slack Connect assistant

Get started
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 dashed 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](#)