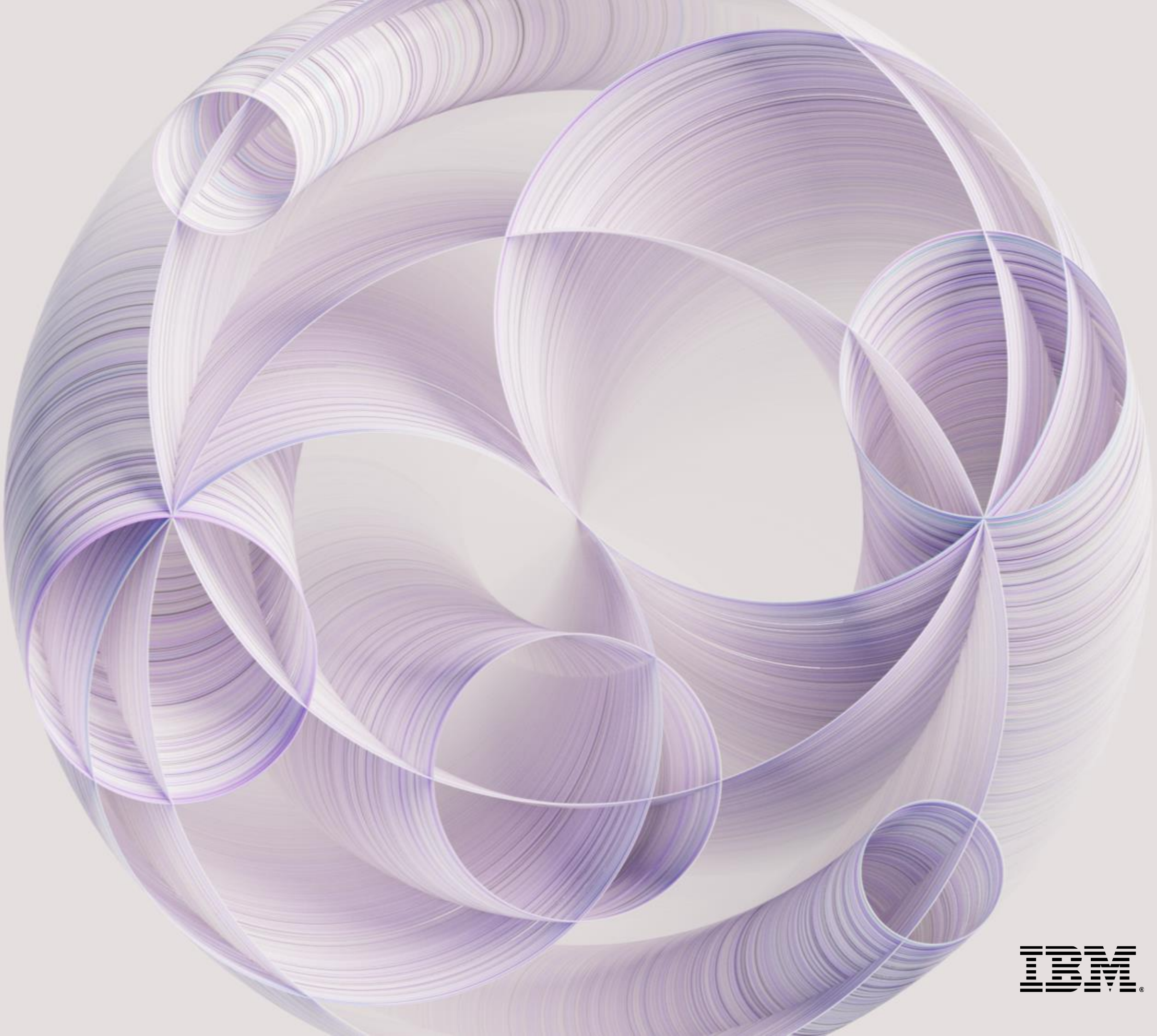
Integrating Watson Assistant and watsonx.ai

Randy Phoa
Client Engineering Leader



Prerequisites

Docker Related

- Create Docker Hub account
- Generate an access token from, https://hub.docker.com/settings/security
- Install Docker

IBM Related

- Provision Code Engine from TechZone
- Create a free Watson Assistant instance from IBM Cloud. Use the new version based on actions.

Hello Word Test – Docker

Test Locally

1. git clone https://github.com/randyphoa/watsonx.git
2. pip install --no-cache-dir torch --index-url https://download.pytorch.org/whl/cpu
3. cd watsonx
4. pip install --no-cache-dir --upgrade -r requirements.txt
5. uvicorn app.main:app --host 0.0.0.0 --port 8080 –reload
6. Go to http://localhost:8080 on your browser

Containerize and deploy

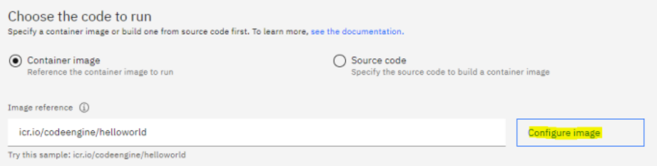
1. docker login --username <USERNAME> --password <ACCESS TOKEN>
2. docker build --rm --platform linux/amd64 --tag <USERNAME>/watsonx .
3. docker run --rm --name watsonx --publish 8080:8080 --platform linux/amd64 randyphoa/watsonx
4. Go to http://localhost:8080 on your browser
5. docker push <USERNAME>/watsonx

Hello Word Test – Code Engine

Code Engine

1. Go to the Code Engine project that was provisioned from TechZone
2. Click on Create Application
3. Click on Configure Image
4. Configure Image settings
5. Configure Resources & scaling
6. Test URL
7. Note the URL, this will be used for Watson Assistant integration

Step 3



Choose the code to run

Specify a container image or build one from source code first. To learn more, see the documentation.

● Container image
Reference the container image to run

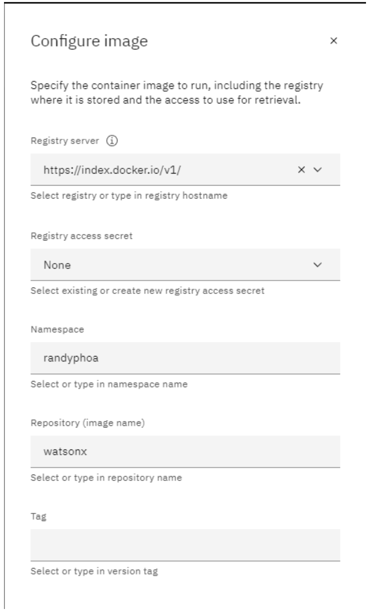
○ Source code
Specify the source code to build a container image

Image reference: icr.io/codeengine/helloworld

Configure image

Try this sample: icr.io/codeengine/helloworld

Step 4



Configure image ×

Specify the container image to run, including the registry where it is stored and the access to use for retrieval.

Registry server: https://index.docker.io/v1/
Select registry or type in registry hostname

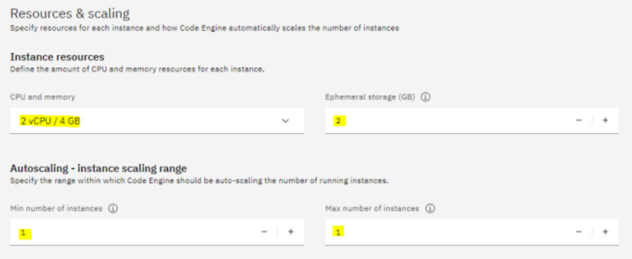
Registry access secret: None
Select existing or create new registry access secret

Namespace: randyphoa
Select or type in namespace name

Repository (image name): watsonx
Select or type in repository name

Tag:
Select or type in version tag

Step 5



Resources & scaling

Specify resources for each instance and how Code Engine automatically scales the number of instances

Instance resources

Define the amount of CPU and memory resources for each instance.

CPU and memory: 2 vCPU / 4 GB

Ephemeral storage (GB): 2

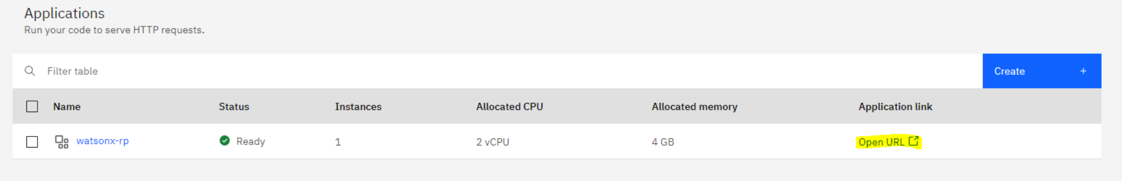
Autoscaling - instance scaling range

Specify the range within which Code Engine should be auto-scaling the number of running instances.

Min number of instances: 1

Max number of instances: 1

Step 6



Applications

Run your code to serve HTTP requests.

Filter table

Create +

□	Name	Status	Instances	Allocated CPU	Allocated memory	Application link
□	watsonx-rp	Ready	1	2 vCPU	4 GB	Open URL

Hello Word Test – Watson Assistant 1

1. Go to your Watson Assistant

2. Click on Actions, then Global Settings at the top right corner

3. Click on the Upload/Download tab and upload watson-assistant-action.json found in the Git repository

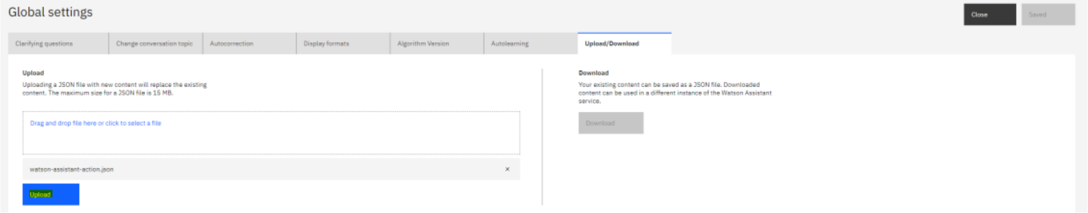
4. Click on Watson Assistant Home and go to Integrations

5. Click on Build custom extension

6. Go to https://<CODE ENGINE URL>/openapi and copy the generated JSON to openapi.json and upload it

7. After defining your custom extension, add it and click through the steps

Step 2



Global settings

Close

Saved

Clarifying questions | Change conversation topic | Autocorrection | Display formats | Algorithm Version | Autolearning | Upload/Download

Upload

Uploading a JSON file with new content will replace the existing content. The maximum size for a JSON file is 15 MB.

Drag and drop file here or click to select a file

watson-assistant-action.json ×

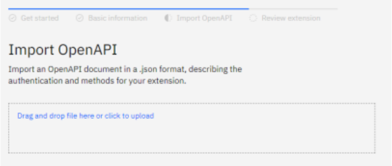
Upload

Download

Your existing content can be saved as a JSON file. Downloaded content can be used in a different instance of the Watson Assistant service.

Download

Step 6



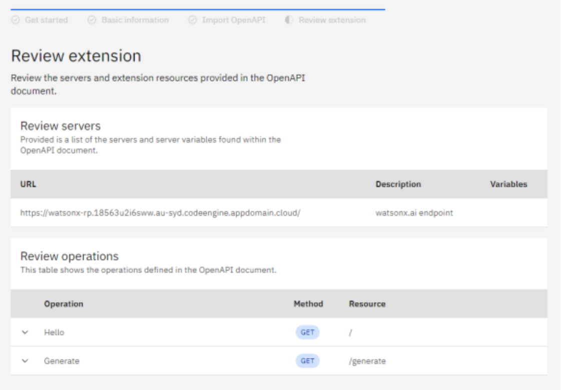
Get started | Basic information | Import OpenAPI | Review extension

Import OpenAPI

Import an OpenAPI document in a .json format, describing the authentication and methods for your extension.

Drag and drop file here or click to upload

Step 6: Final output



Get started | Basic information | Import OpenAPI | Review extension

Review extension

Review the servers and extension resources provided in the OpenAPI document.

Review servers

Provided is a list of the servers and server variables found within the OpenAPI document.

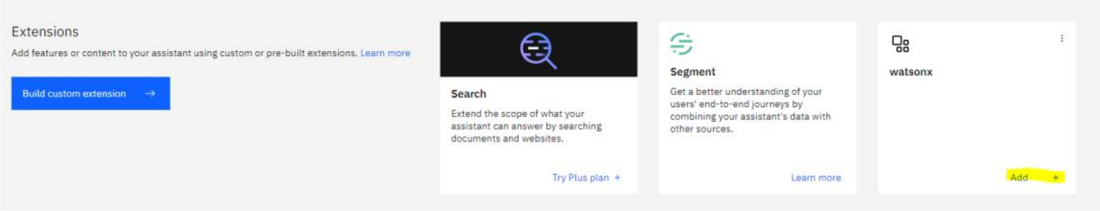
URL	Description	Variables
https://watsonx-rp.18563u2i6sww.au-syd.codeengine.appdomain.cloud/	watsonx.ai endpoint	

Review operations

This table shows the operations defined in the OpenAPI document.

Operation	Method	Resource
Hello	GET	/
Generate	GET	/generate

Step 7



Extensions

Add features or content to your assistant using custom or pre-built extensions. Learn more

Build custom extension →

Search

Extend the scope of what your assistant can answer by searching documents and websites.

Try Plus plan +

Segment

Get a better understanding of your users' end-to-end journeys by combining your assistant's data with other sources.

Learn more

watsonx

Add +

Hello Word Test – Watson Assistant 2

1. Go to your Watson Assistant

2. Click on Actions, then Global Settings at the top right corner

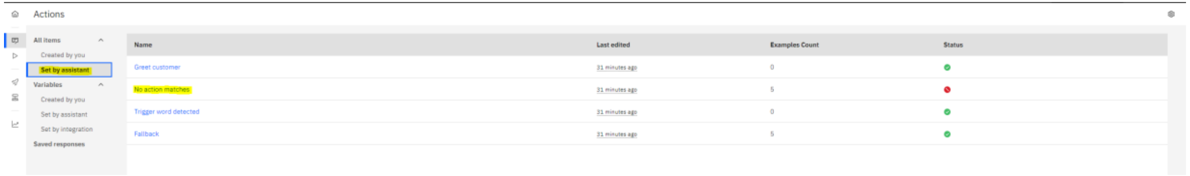
3. Click on the Upload/Download tab and upload watson-assistant-action.json found in the Git repository

4. Click on Watson Assistant Home and go to Integrations

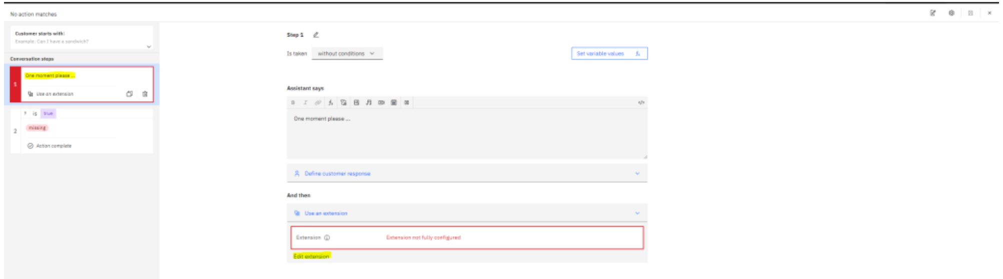
5. Click on Build custom extension

6. Go to https://<CODE ENGINE URL>/openapi and copy the generated JSON to openapi.json and upload it

7. After defining your custom extension, add it and click through the steps

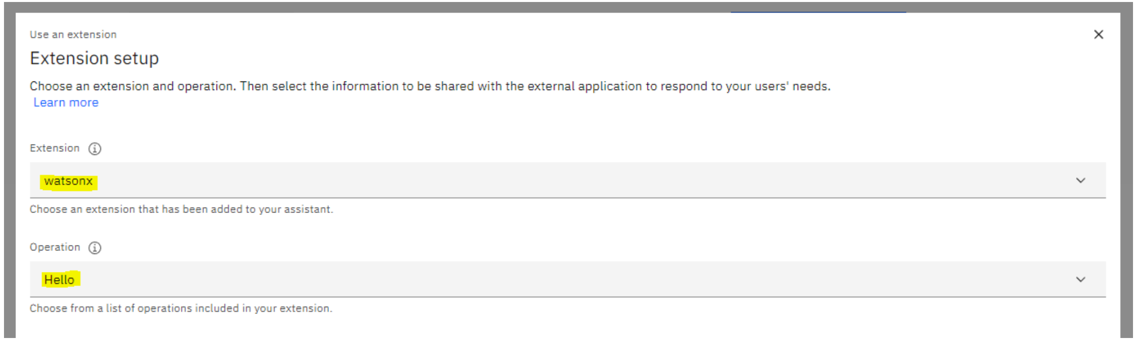


Name	Last edited	Examples Count	Status
Greet customer	31 minutes ago	0	●
No action matches	31 minutes ago	5	●
Trigger word detected	31 minutes ago	0	●
Fallback	31 minutes ago	5	●



One moment please ...

Extension ⓘ Extension not fully configured



Extension setup

Choose an extension and operation. Then select the information to be shared with the external application to respond to your users' needs.

Extension ⓘ: watsonx

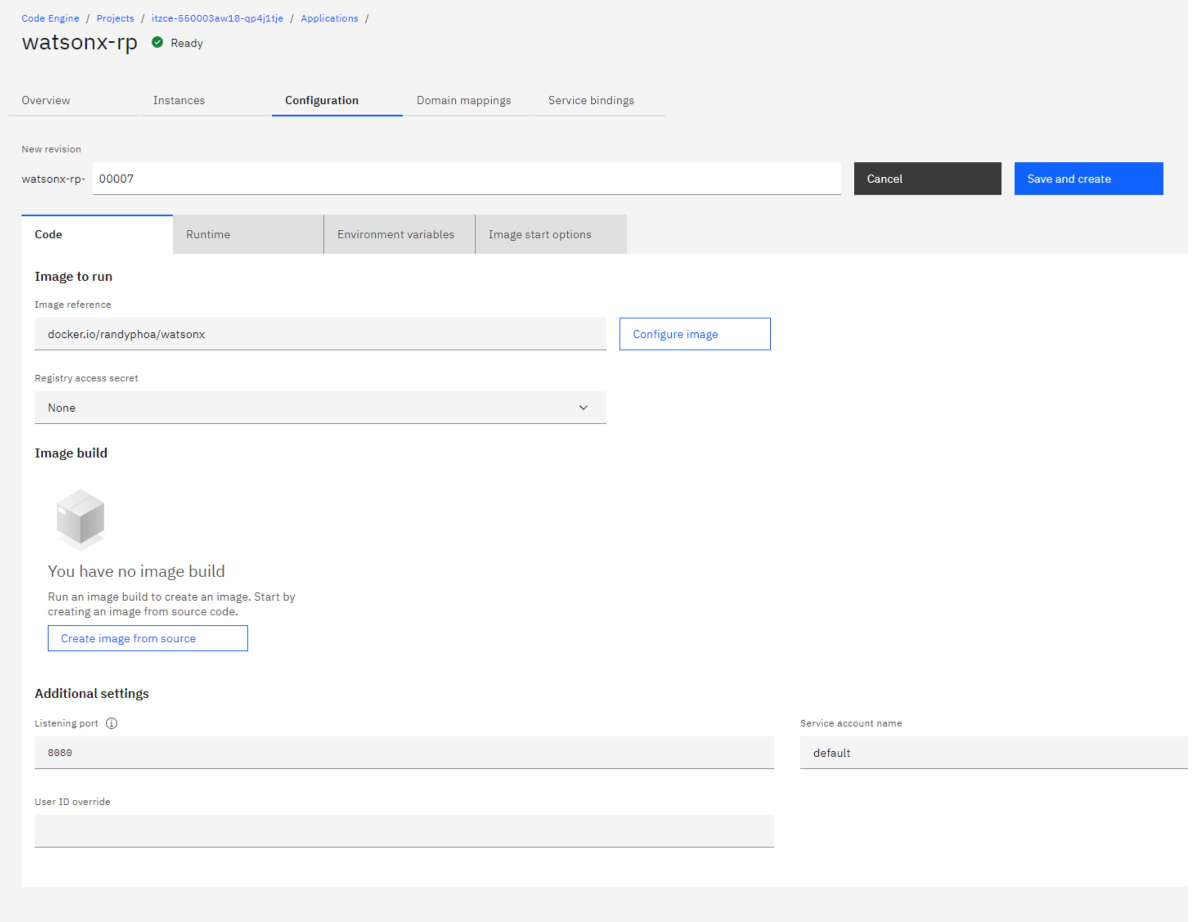
Choose an extension that has been added to your assistant.

Operation ⓘ: Hello

Choose from a list of operations included in your extension.

Hello Word Test – Making updates

1. Modify your code as required
2. Build Docker image and push to Docker Hub
3. Click on your application
4. Go to the Configuration tab and click Save and create
5. A new version base on your latest Docker image will be automatically deployed



New revision
watsonx-rp- 00007

Image to run

Image reference: docker.io/randyphoa/watsonx

Registry access secret: None

Image build

You have no image build

Run an image build to create an image. Start by creating an image from source code.

Additional settings

Listening port: 8080

Service account name: default

User ID override:

IBM