Structured Job Task Update in Maximo Using Job Plan Number

Table of Contents

1. *Background* 2
2. *Architecture Diagram* 2
3. *Technologies used:* 3
4. *Create API key* 3
5. *Watsonx.AI Configurations* 4
6. *Watsonx Assistant Configurations* 6
7. *Adding Custom Extensions* 8
8. *Import Actions* 14
9. *Conclusion* 22

1. Background

In IBM Maximo, a **job plan** is identified by a unique **job plan number**, which serves as a reference for predefined sets of maintenance tasks. Each job plan consists of multiple **job tasks**, where each task represents a specific step or action, typically stored in the description field.

In many cases, job plans are created with a **single long description** or a **high-level instruction** under the plan, rather than a structured breakdown of individual steps. This makes it harder for technicians to follow the procedure in a clear, step-by-step manner during execution.

This use case aims to improve the clarity and usability of existing job plans by:

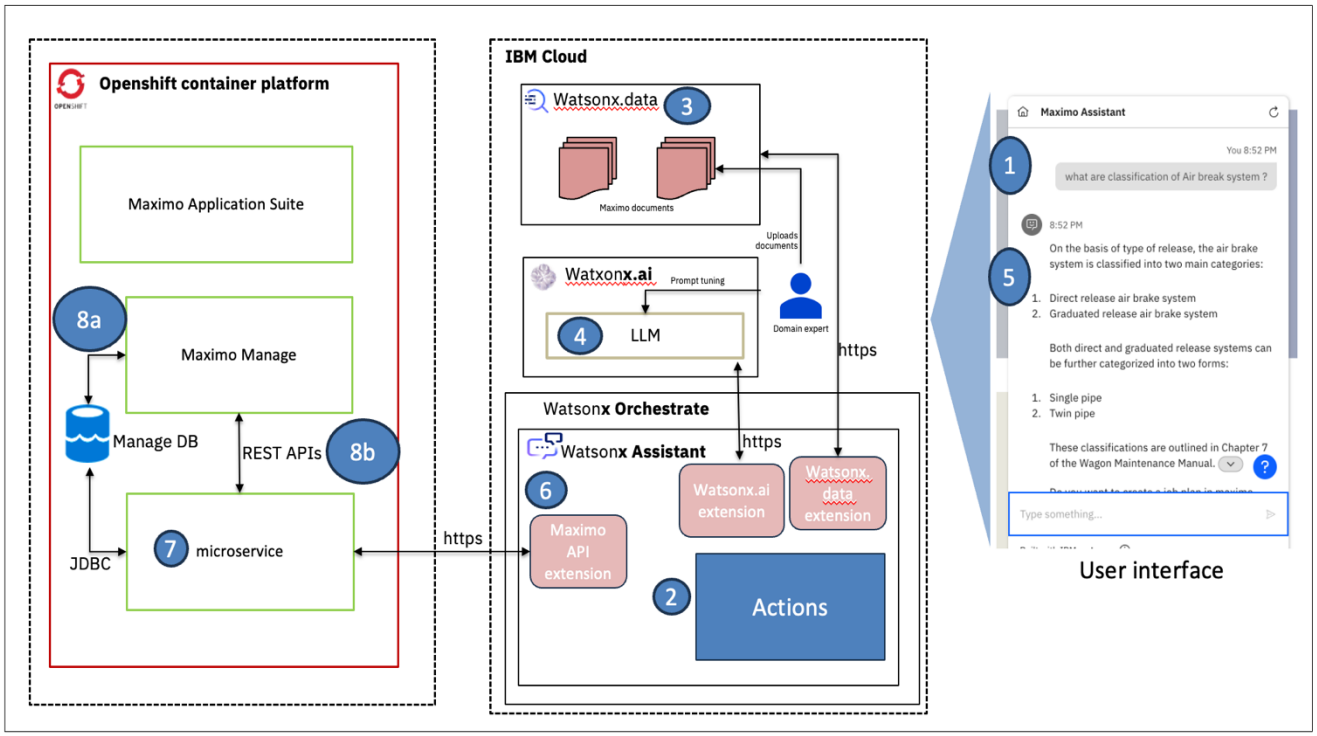
- Retrieving the job plan using the **provided job plan number**
- Analysing the long, unstructured description
- **Breaking it down into smaller, individual job tasks**
- Updating the Maximo job plan with this structured list of tasks

By transforming vague or condensed instructions into actionable task items, this approach:

- Enhances technician understanding and execution
- Improves compliance with procedures
- Reduces the chances of human error
- Aligns job plans more closely with real-world operational workflows

This solution leverages automation and integration with Maximo to streamline the update process—ensuring job plans are not only technically accurate but also user-friendly and execution-ready.

2. Architecture Diagram



3. Technologies used:

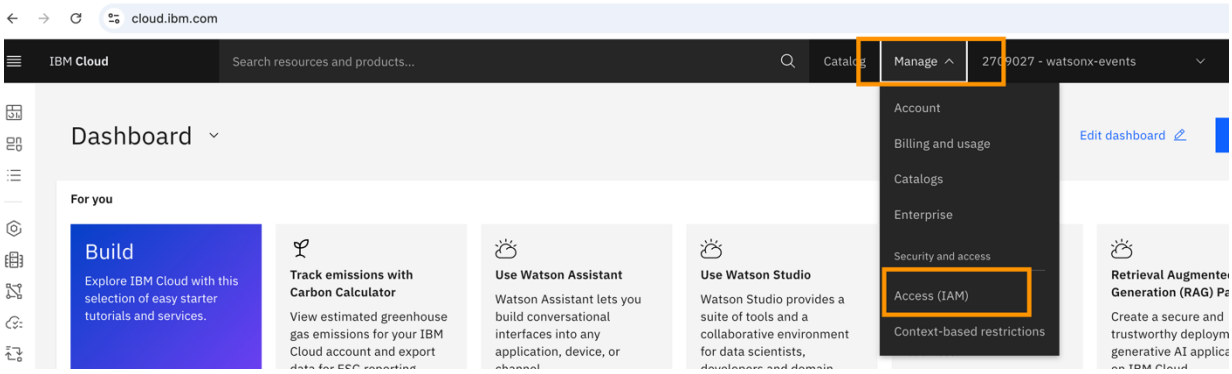
Services	Purpose
Watsonx.ai Studio	To generate structured responses using LLMs
Watsonx.ai Runtime	To host and run foundation models
Watsonx Assistant	For building conversational search applications
Maximo	Asset Management Tool

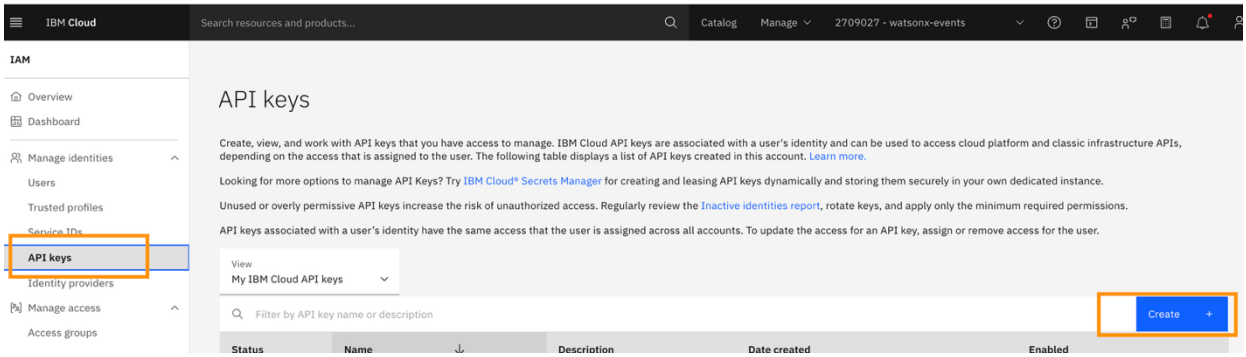
4. Create API key

Why to Create API key: An API key is required to securely authenticate and authorize communication between services in your Watsonx and Maximo environment.

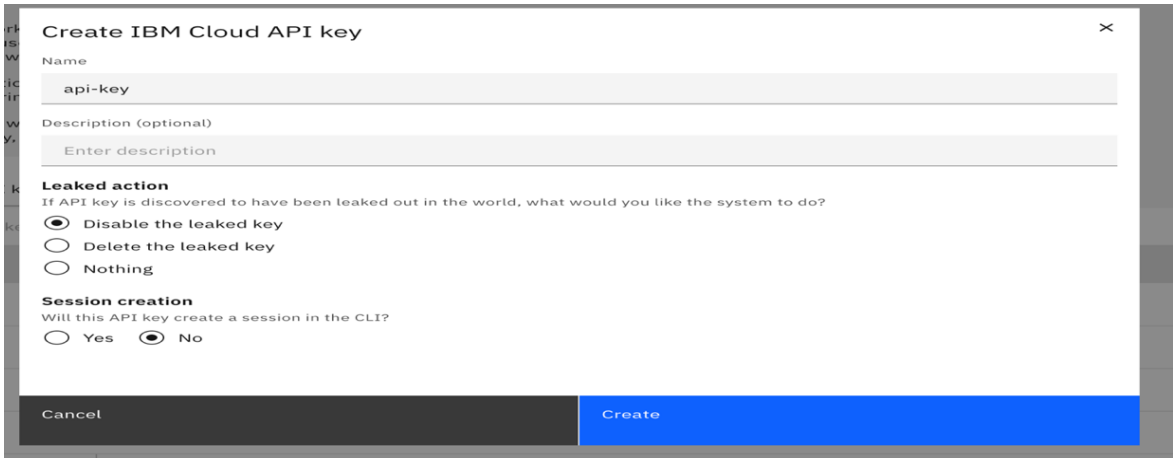
- Always keep API keys confidential.
- Rotate them periodically.
- Never hard-code them directly into source code (use .env or secrets managers instead).

4.1 Go to *IBM Cloud* →click on *Manage* →click on *Access (IAM)* →then click on *API Keys* on the left side →Click on *Create*





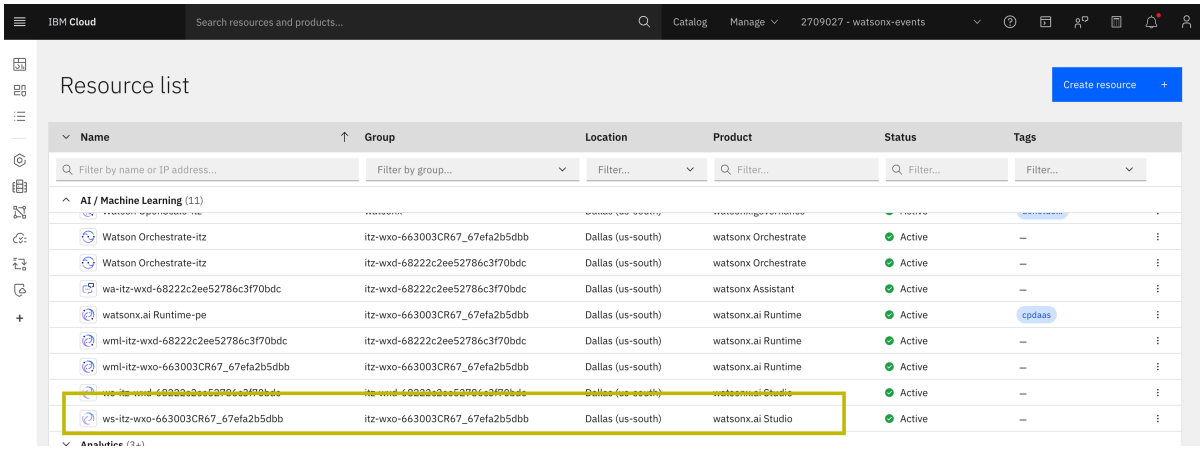
4.2 Provide *Name* and *Description* →Click on *Create* → Copy & download the API key for future reference



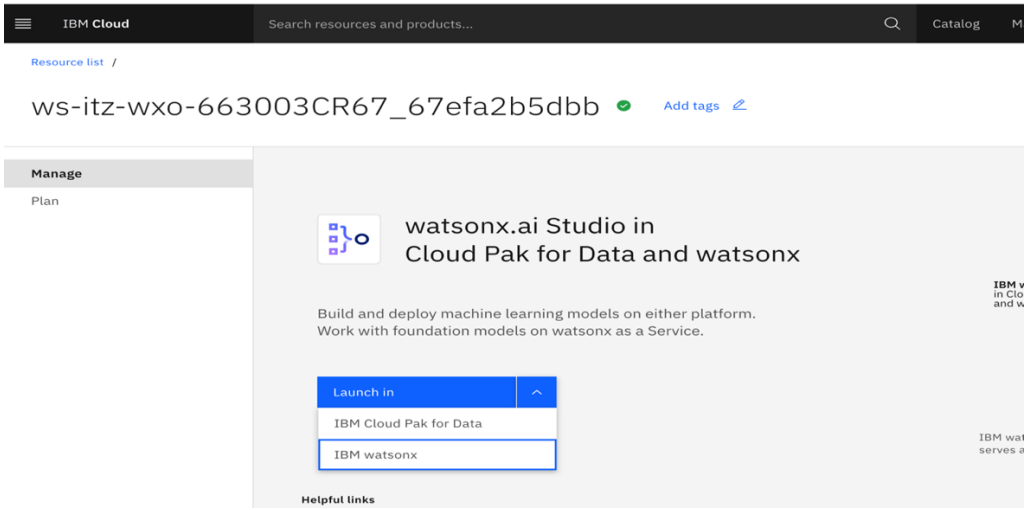
5. Watsonx.AI Configurations

5.1 Launch watsonx.ai

Click on *Resource list* from left side panel in *IBM Cloud* →Click on ***watsonx.ai Studio*** under ***AI / Machine Learning*** resource. (highlighted in image)

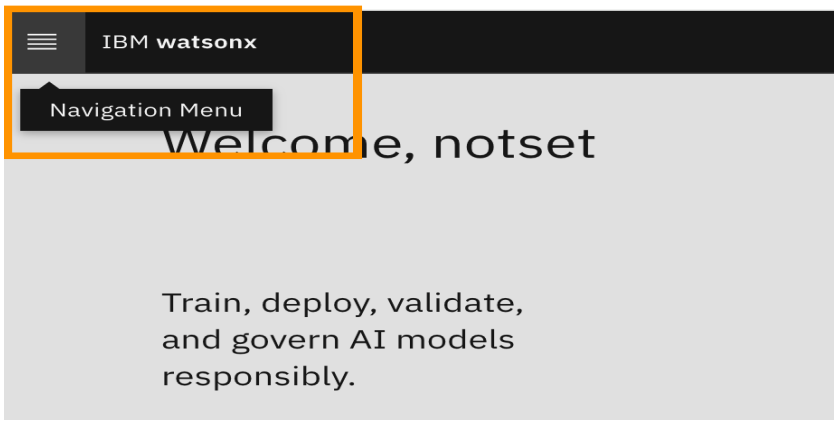


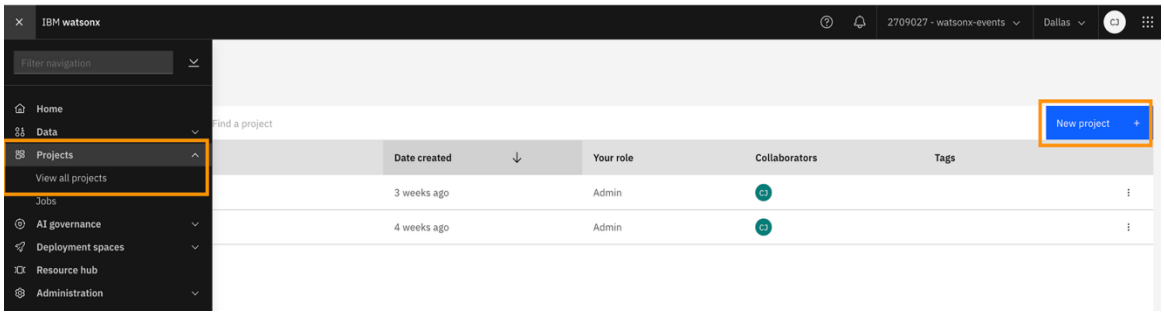
5.2 Select Launch in *IBM watsonx*



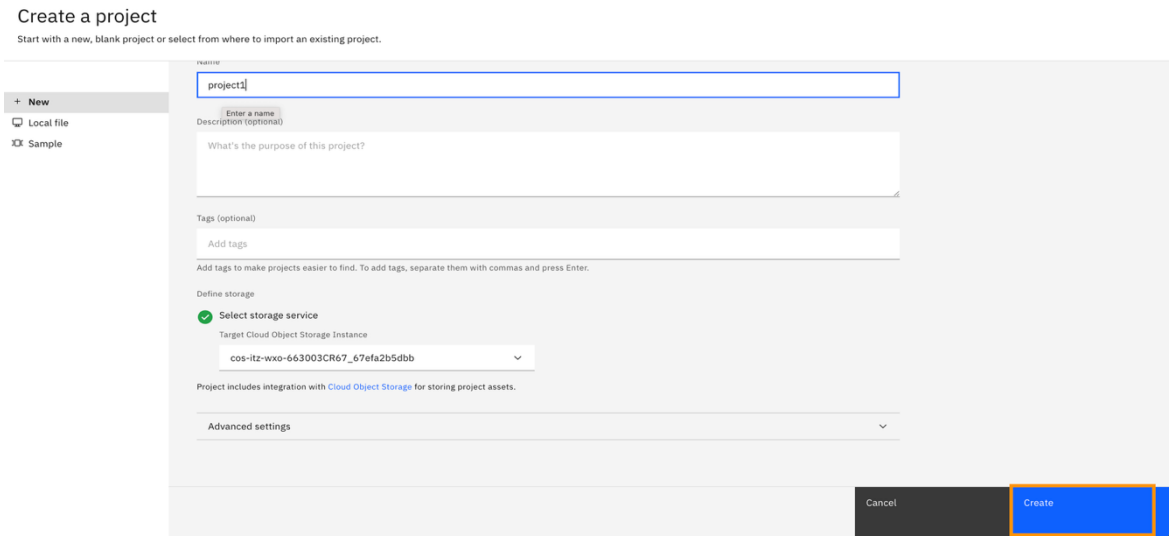
5.3 Create New project

Click on Navigation Menu at top left side → Click on *Projects* → Click on *View all projects* → Click on *New project*.

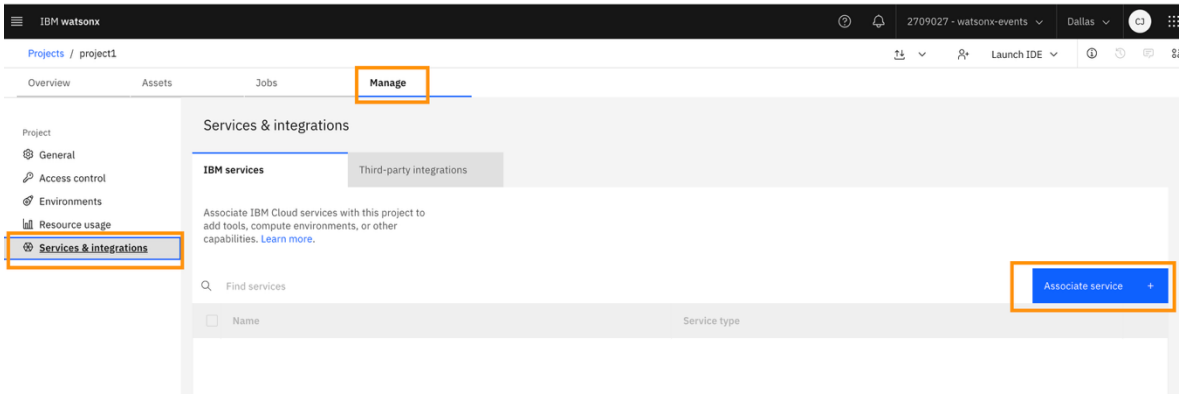




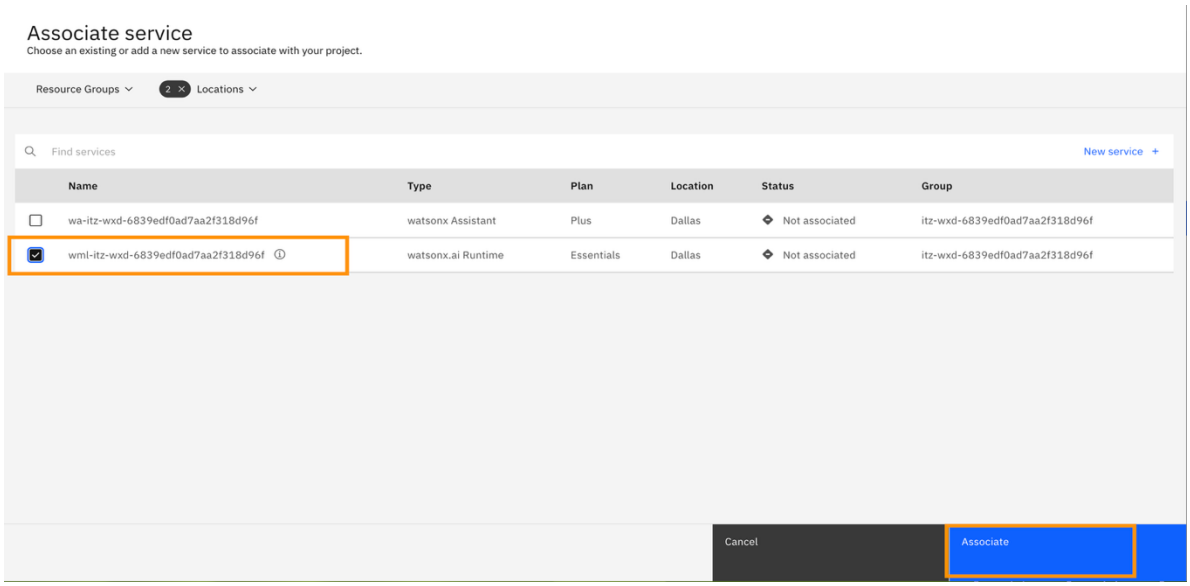
5.4 Provide *Name* and *Select storage service* from drop down →Click on *Create*.



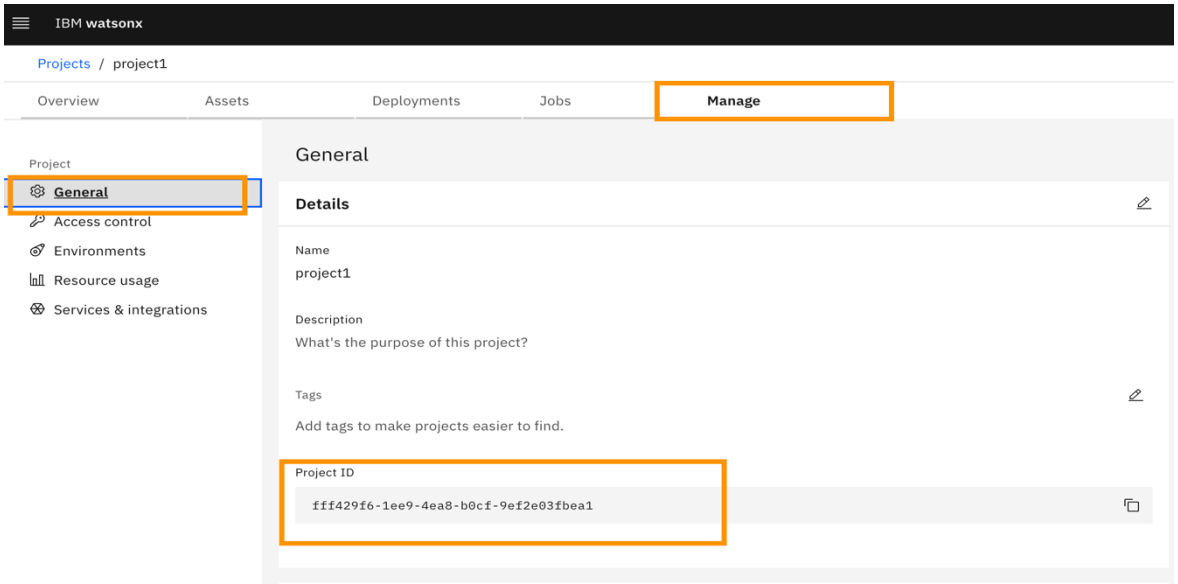
5.5 Once project created, Navigate on *Manage* tab → select *Services & Integrations* (left side panel) → click on *Associate service* (highlighted in image)



5.6 Select service name starts with wml(ex: wml-xxx-xxx-xxxxxxx) → click on *Associate*.



5.7 Navigate to *Manage* tab → Click on *General* → Copy and Save *Project ID* for the future use

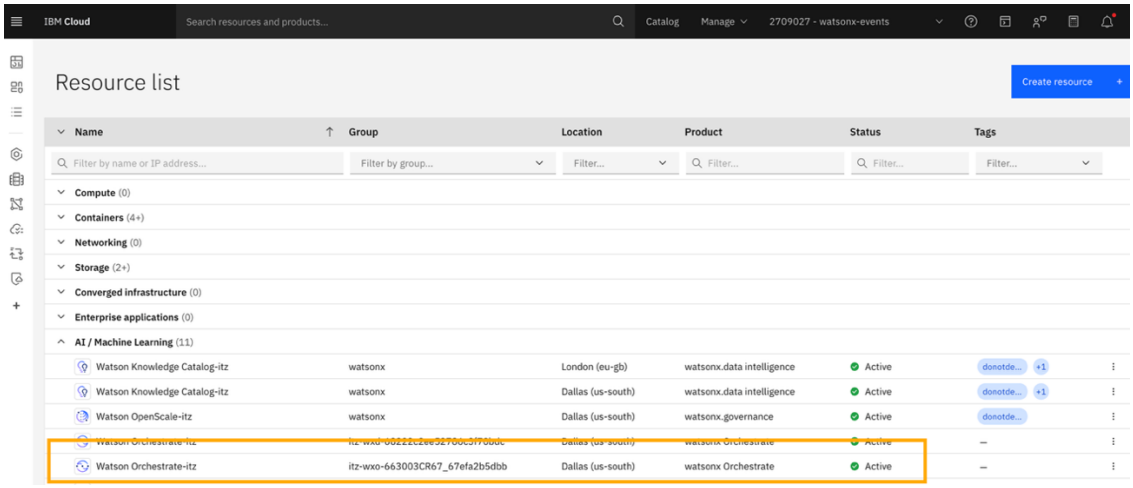


6. Watsonx Assistant Configurations

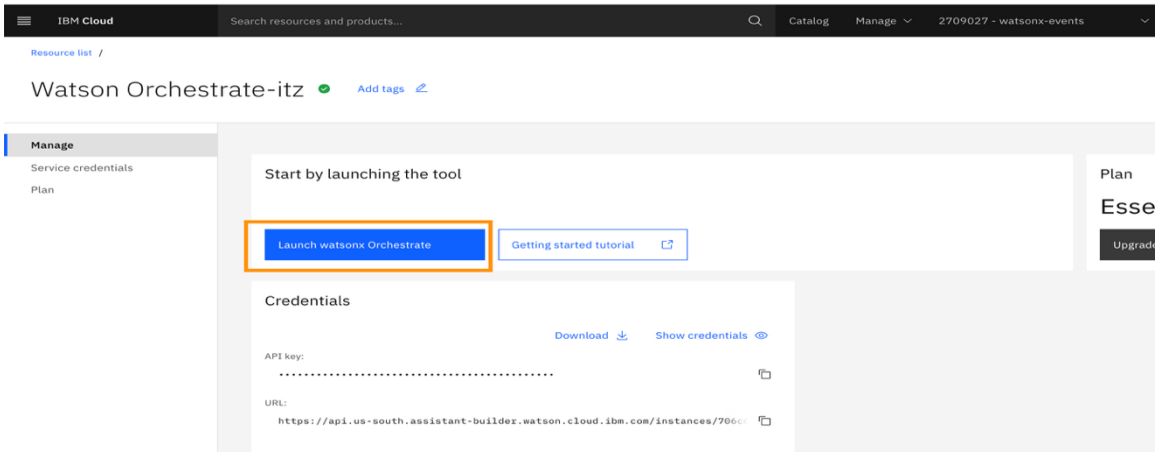
6.1 Launch watsonx assistance in orchestrate

Go to Resource list in *IBM Cloud* →Expand *AI/Machine learning* →Click on *Watson Orchestrate-*

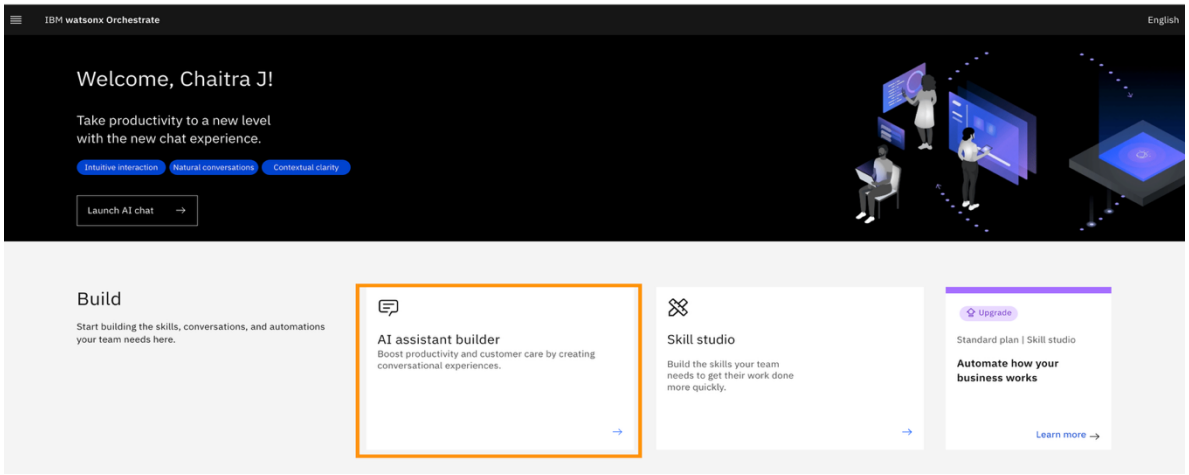
(highlighted in image)



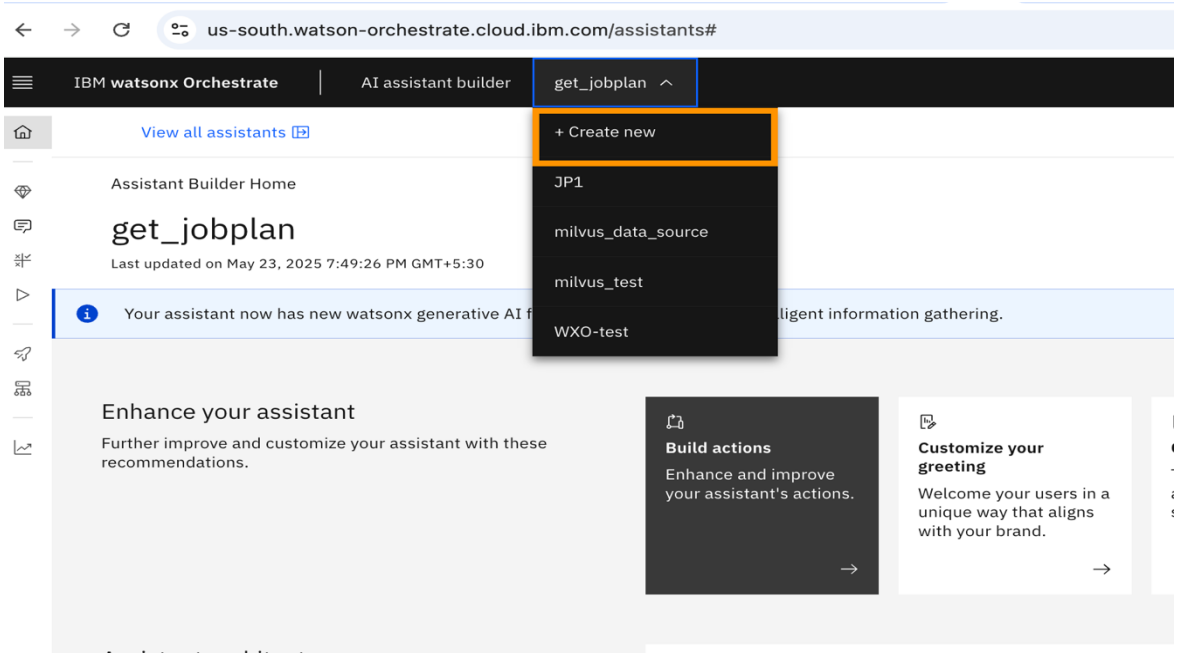
6.2 Click on *Launch watsonx Orchestrate*



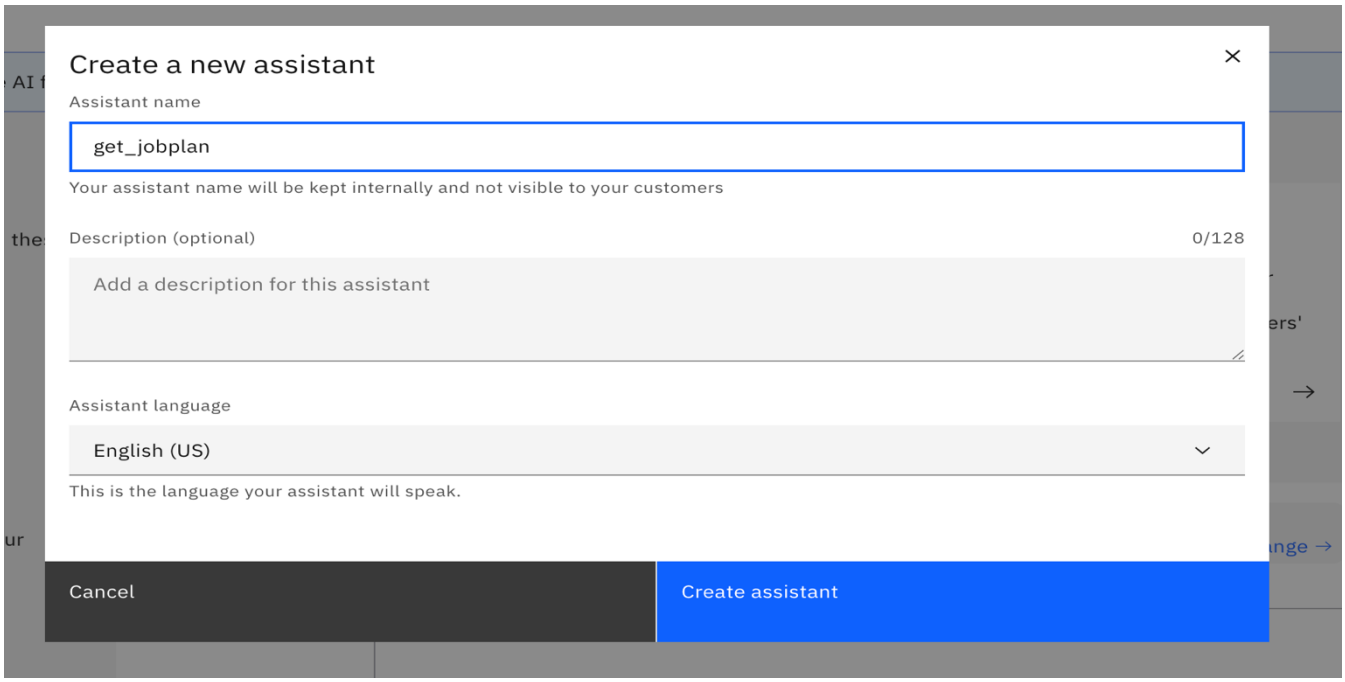
6.3 Click on *AI assistant builder*



6.4 Click on drop down next to the *AI assistant builder* →Click on *Create new* for creating new Assistant



Provide *Assistant name* (ex: get_jobplan)

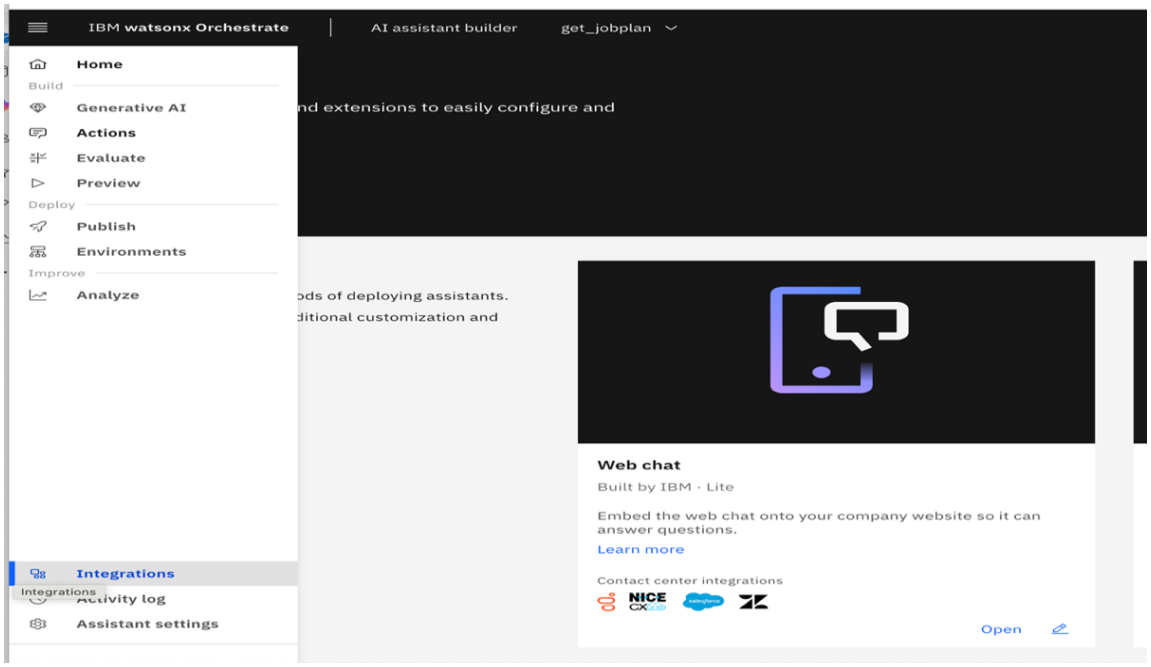


7. Adding Custom Extensions

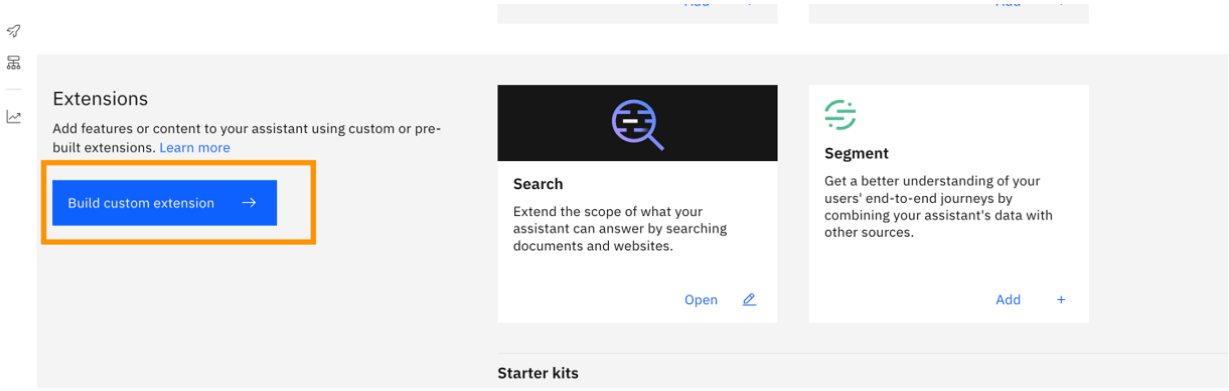
7.1 Watsonx AI integration

7.1.1 Create Api key in *IBM Cloud* for authentication →follow Step3 to *Create API key*.

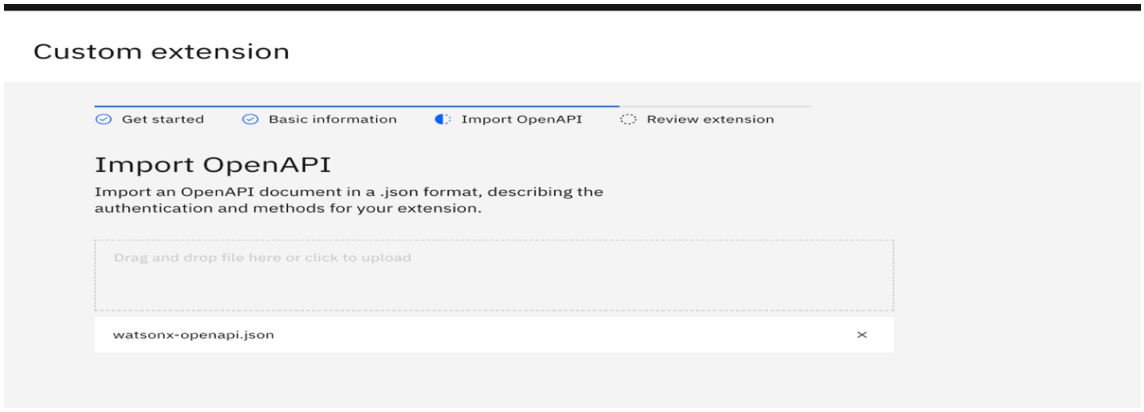
7.1.2 From Left Navigation, select *Integrations* option



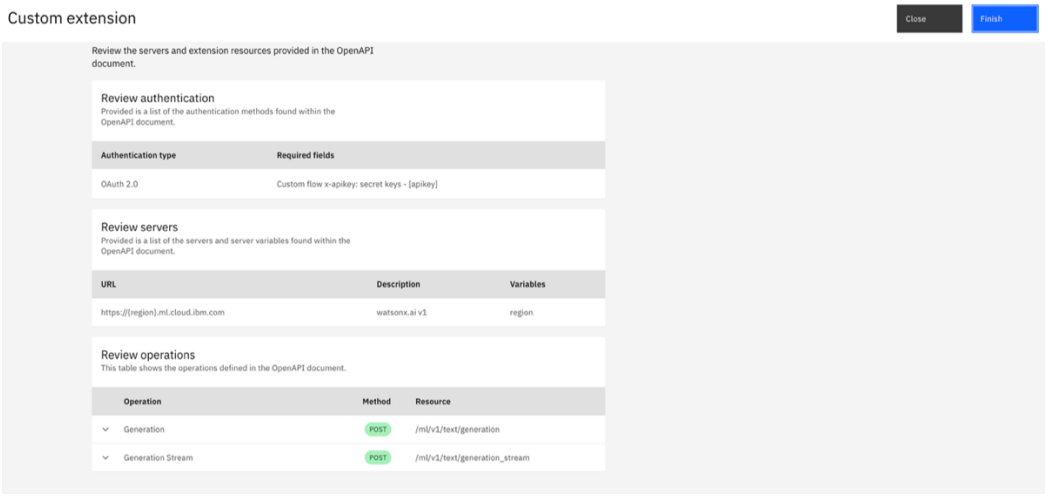
7.1.3 Click on *Build custom extension*



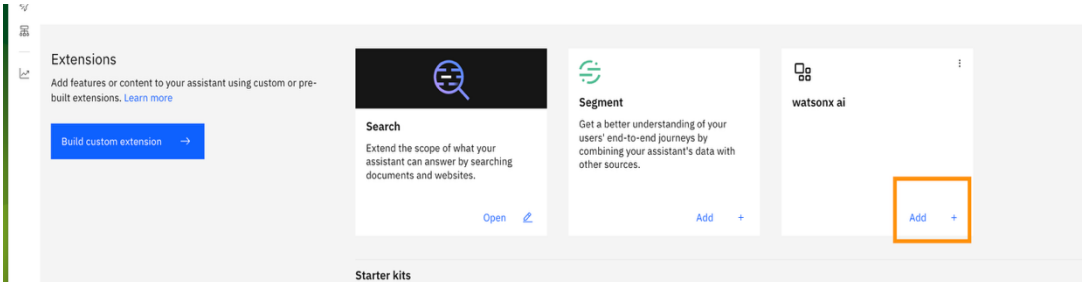
7.1.4 Use [watsonx-openapi.json] Open api spec & import it.

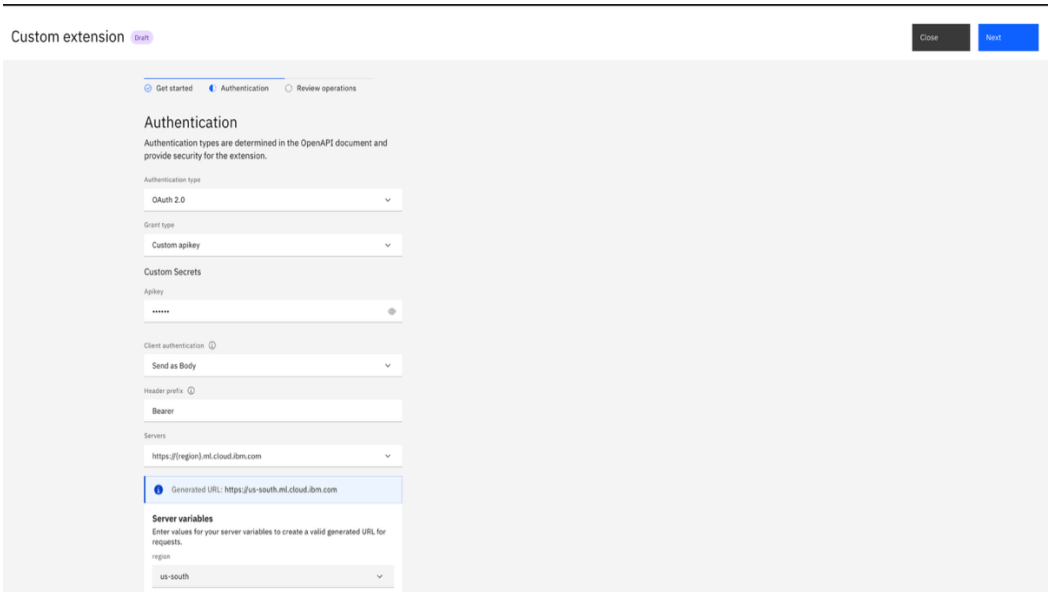


7.1.5 Review and click on *Finish*



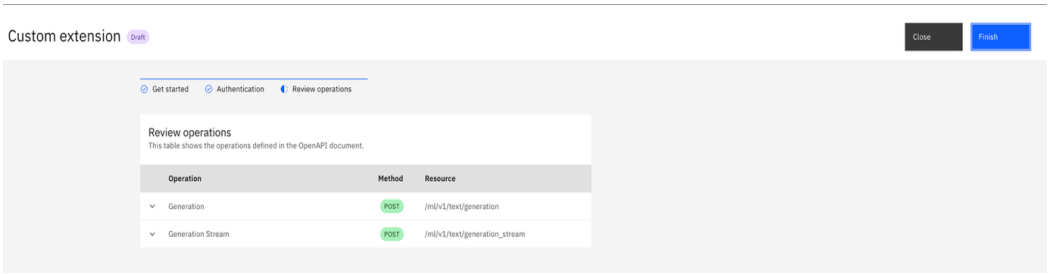
7.1.6 Click on *Add*





7.1.7 Select/Provide below fields

- Authentication type: OAuth 2.0
- Grant type: Custom apikey
- Api key: provide apikey created in step 6.1.1
- Client authentication: sent as body



7.1.8 Click on next & *Finish*

7.2 Maximo custom extension

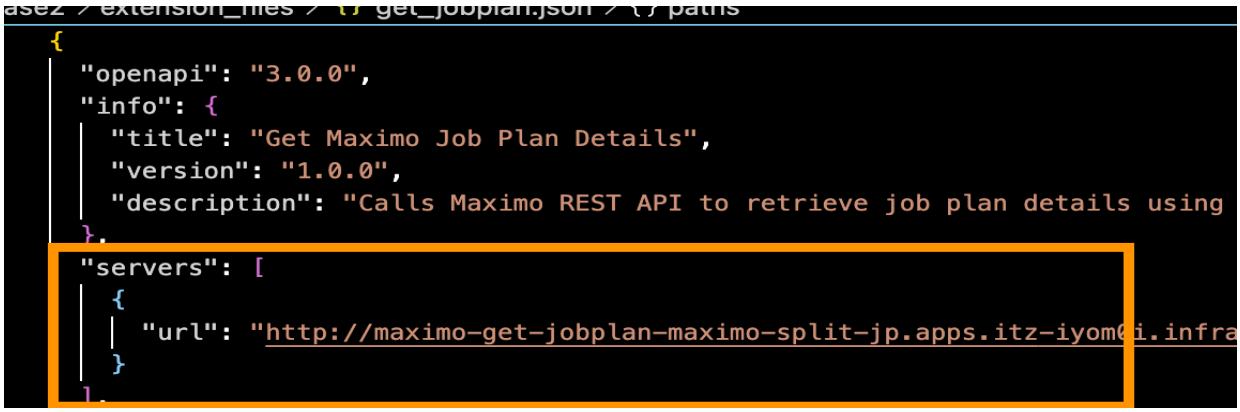
We need to build and deploy code in Maximo to support the following three scenarios

- Retrieve job plan details
- Update job plan status
- Update job plan tasks

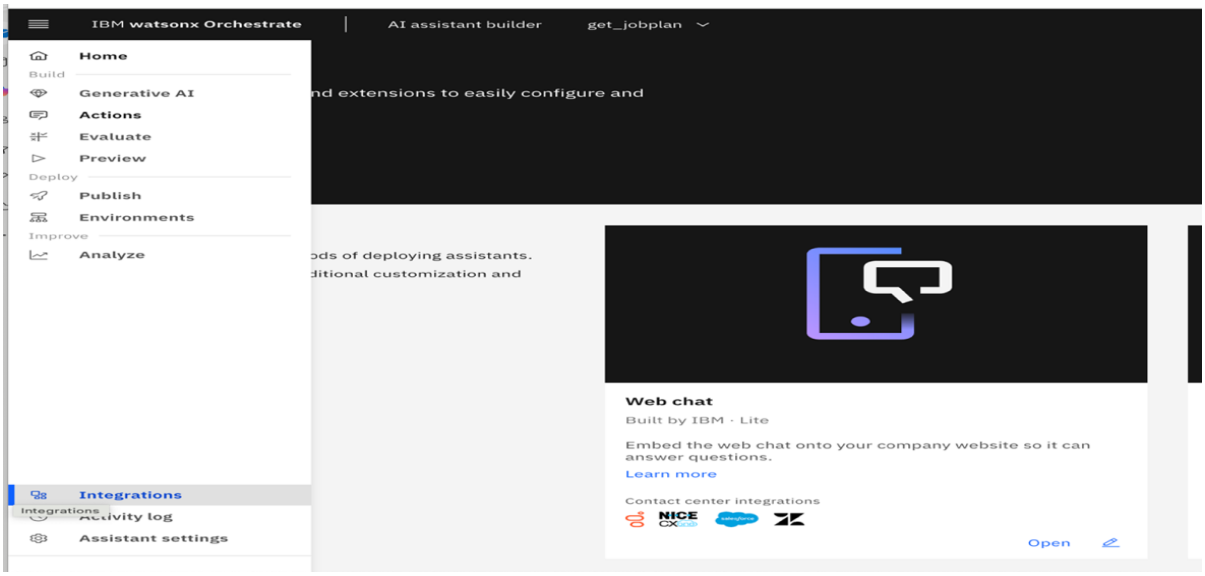
The steps outlined below apply to ***one(Retrieve job plan details)*** scenario. Please follow the same steps for the remaining two scenarios as well.

7.2.1 Perform the steps provided in the README.md for deploying get_jobplan.py python code file.

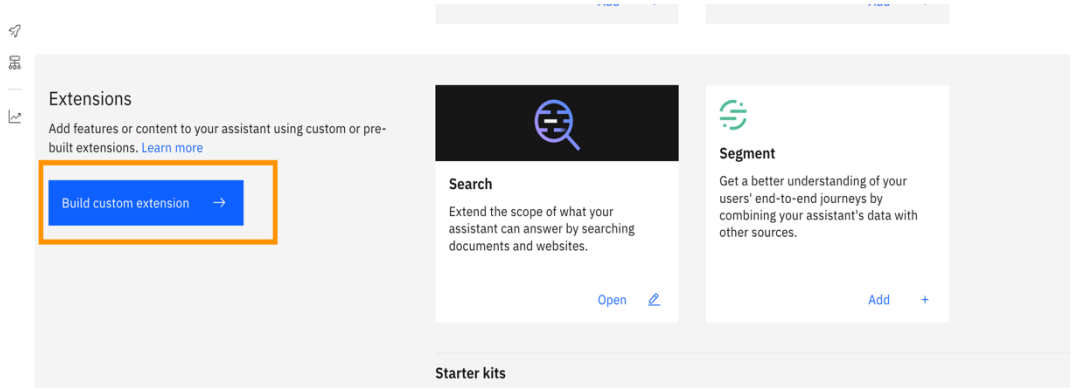
7.2.2 Update the url field in the **get_jobplan.json** OpenAPI file using the URL generated in the step **7.2.1**



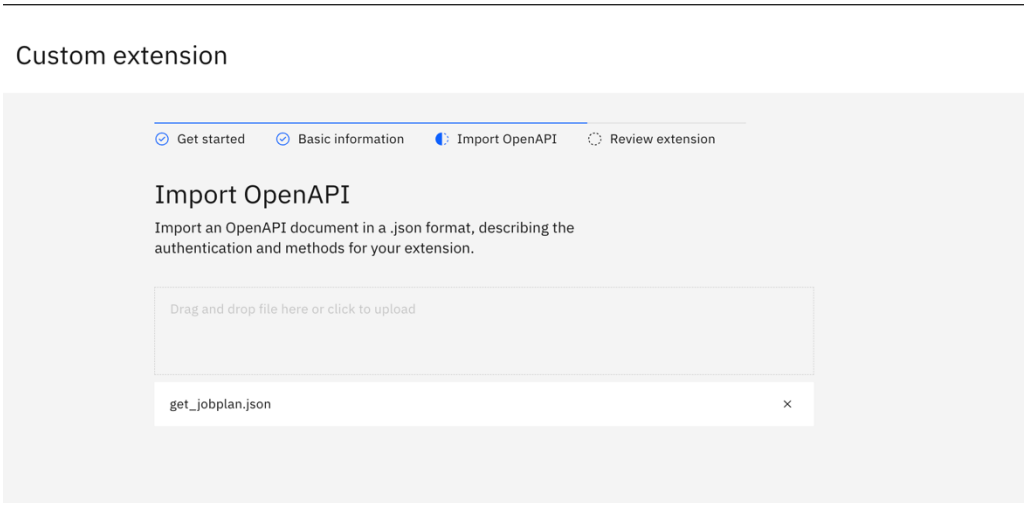
7.2.3 Use **get_jobplan.json** OpenAPI file to add Maximo extension to retrieve the job plan details for the given job plan number.
From Left Navigation, select *Integrations* option



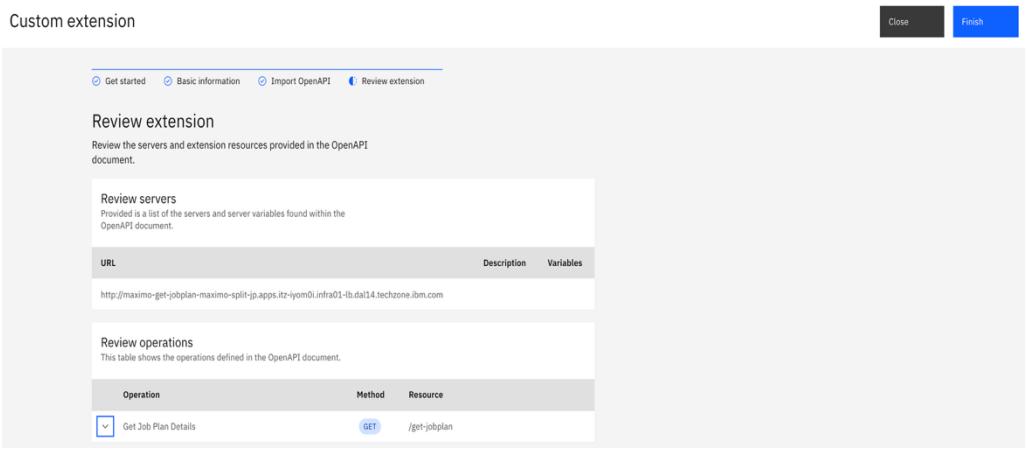
7.2.4 Click on *Build custom extension*



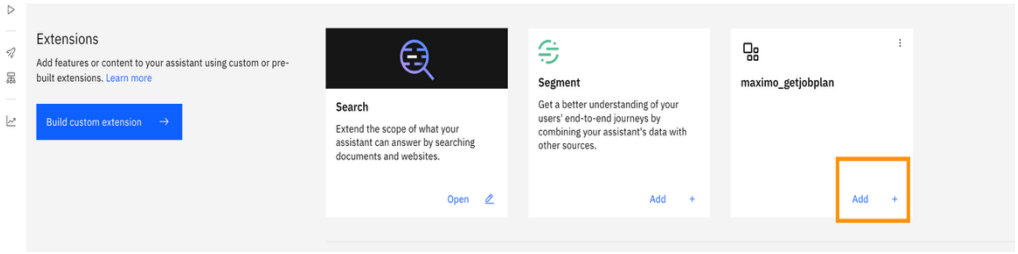
7.2.5 *Import open API* file



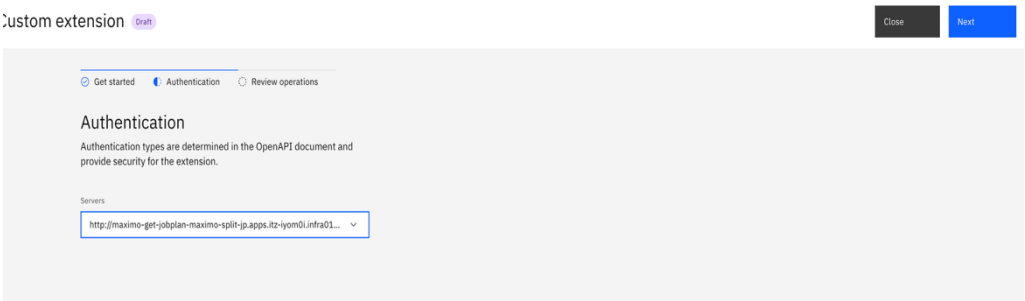
7.2.6 Review and click on *finish*



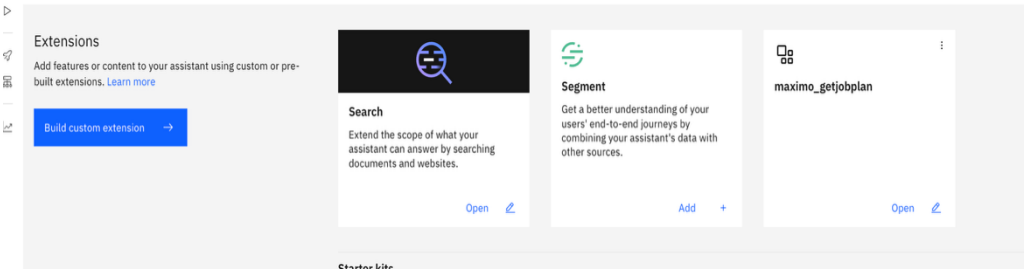
7.2.7 Click on *Add*



7.2.8 Make sure under *Authentication you* can see the Maximo URL (same URL will be present in the openAPI file)



7.2.9 Click Next and *Finish*.



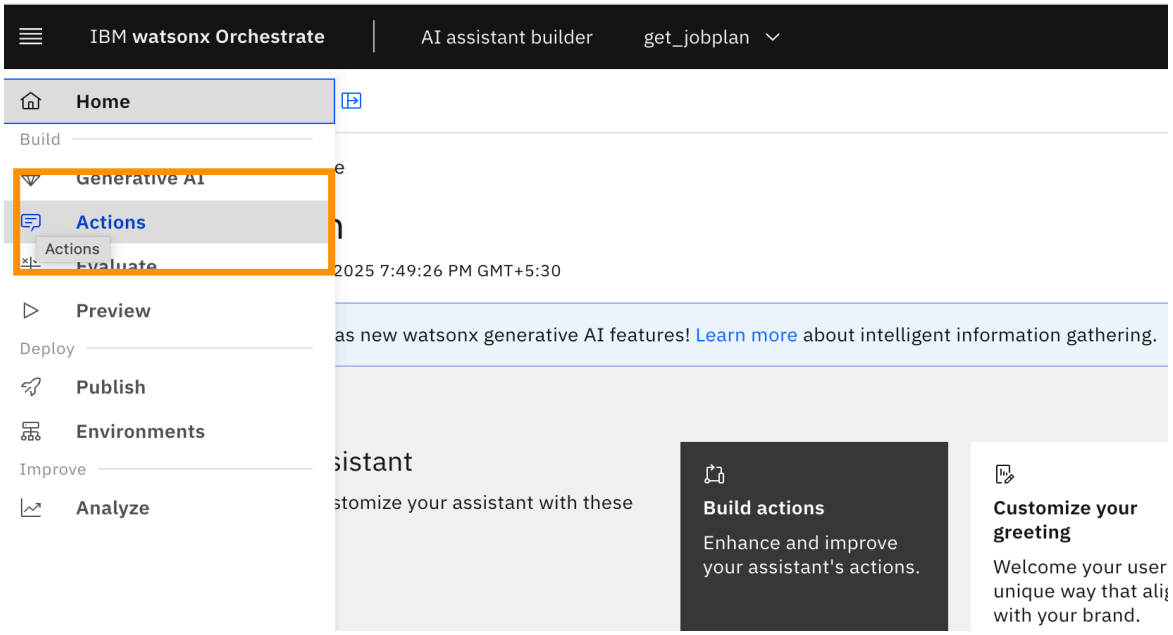
7.2.10 Follow same steps from 6.2.1 to 6.2.7 for the remaining 2 scenarios

Task	Python code file	Open API file	Custom extension name
Update job plan status	update_status.py	update_status.json	update_status_maximo
Update job plan tasks	update_jobtask.py	update_jobtask.json	update_jobtasks

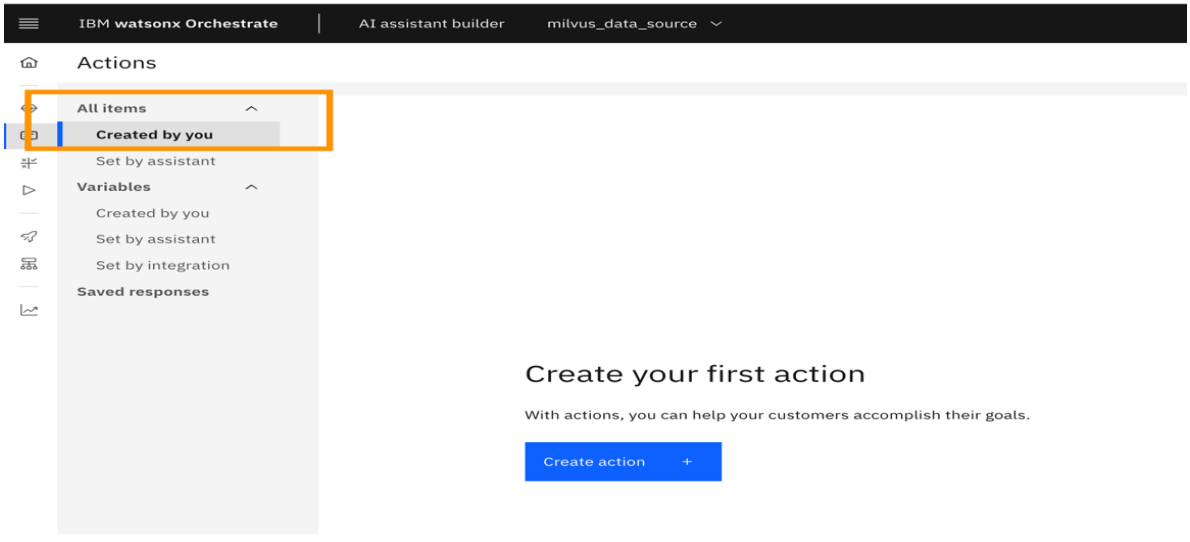
8. Import Actions

8.1 Use the file **actions.json** file

8.1.1 Select *Actions* form the left side panel

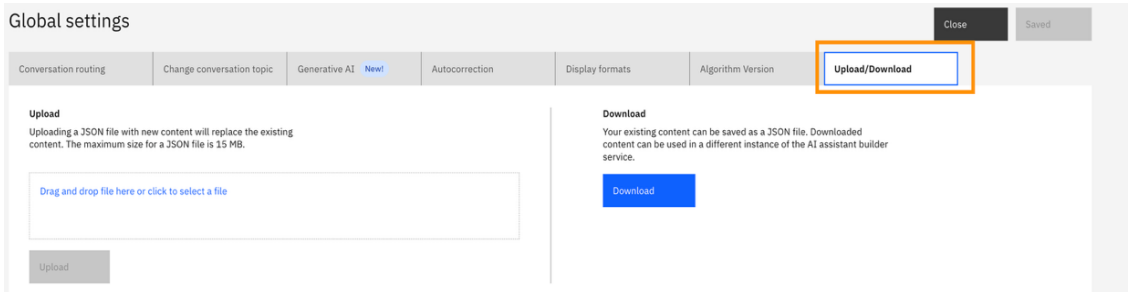


8.1.2 Click on *Created by You*

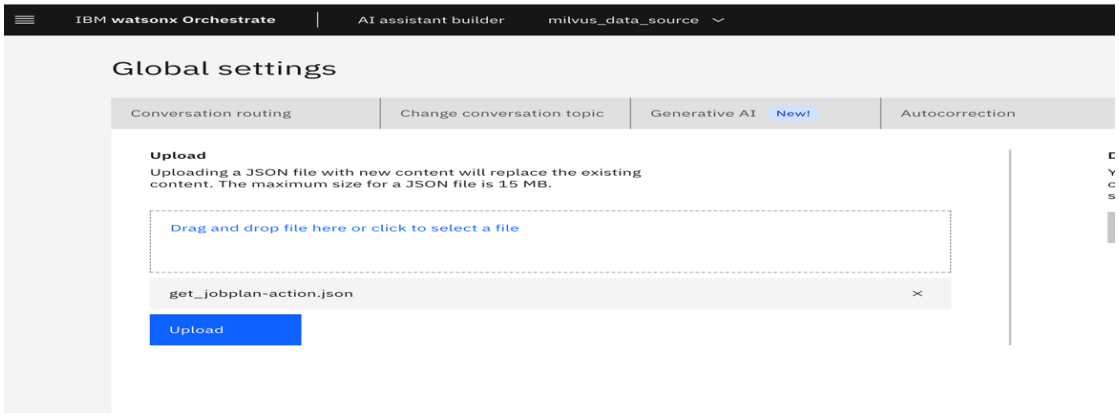


8.1.3 Select *Global setting*s on right top corner

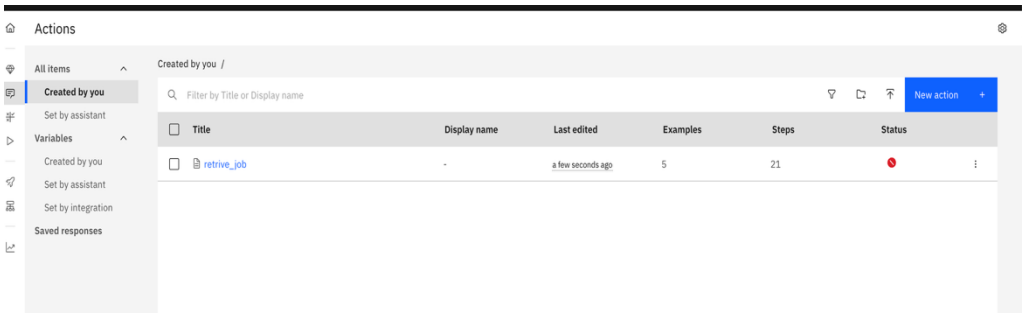
8.1.4 Navigate to *Upload/Download* option



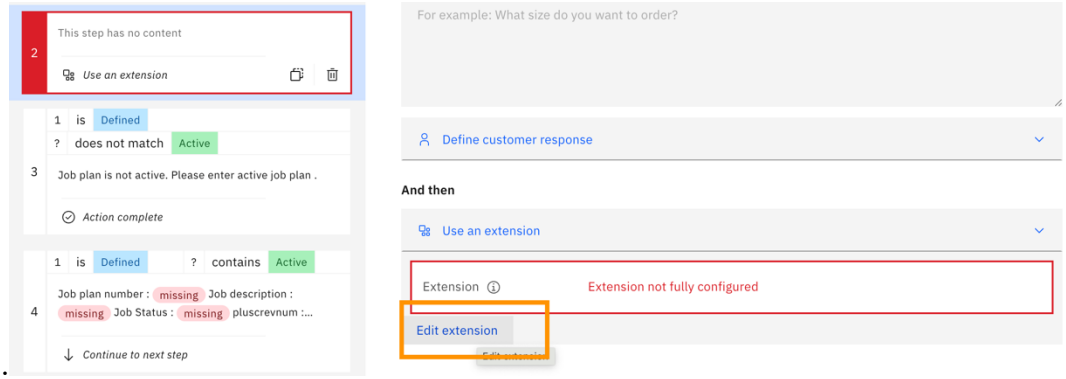
8.1.5 Click on *Drag and drop file here* option & select ***actions.json*** file →Click on *Upload*



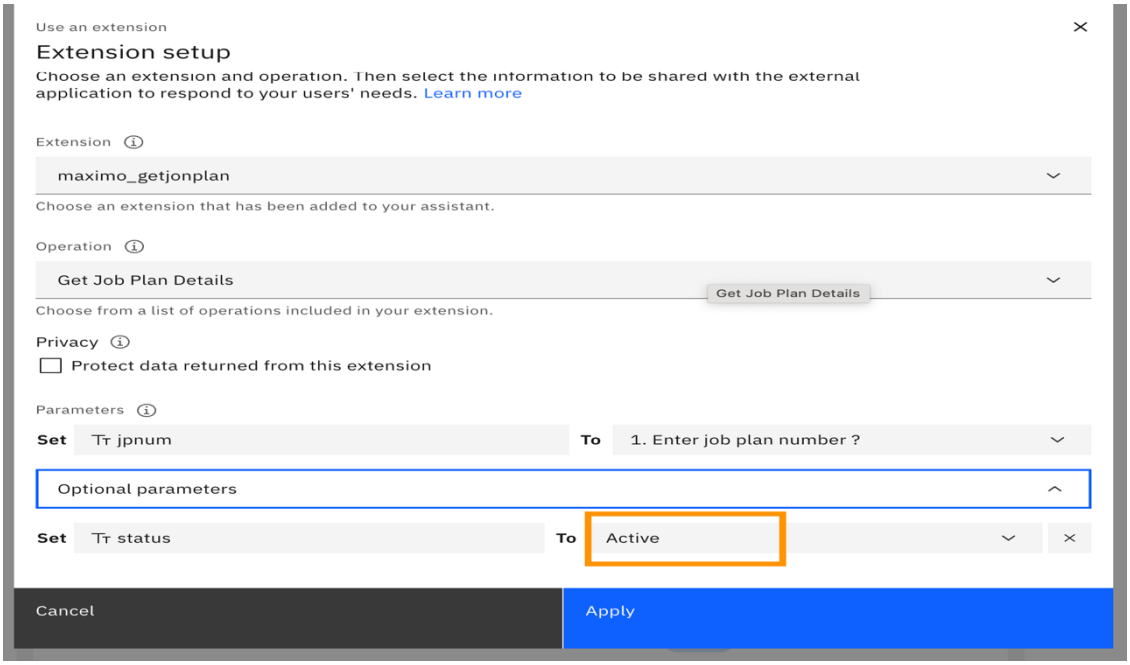
8.1.6 Click on *retrive_job* title

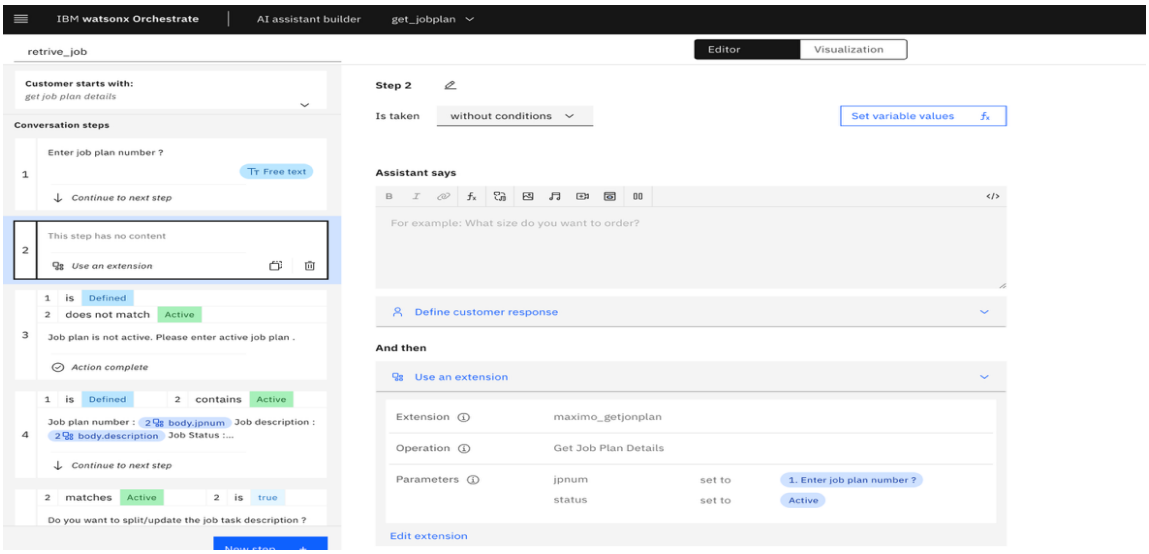


8.1.7 Click on Step2 and select *Edit settings* →select **maximo_getjobplan** Maximo extension

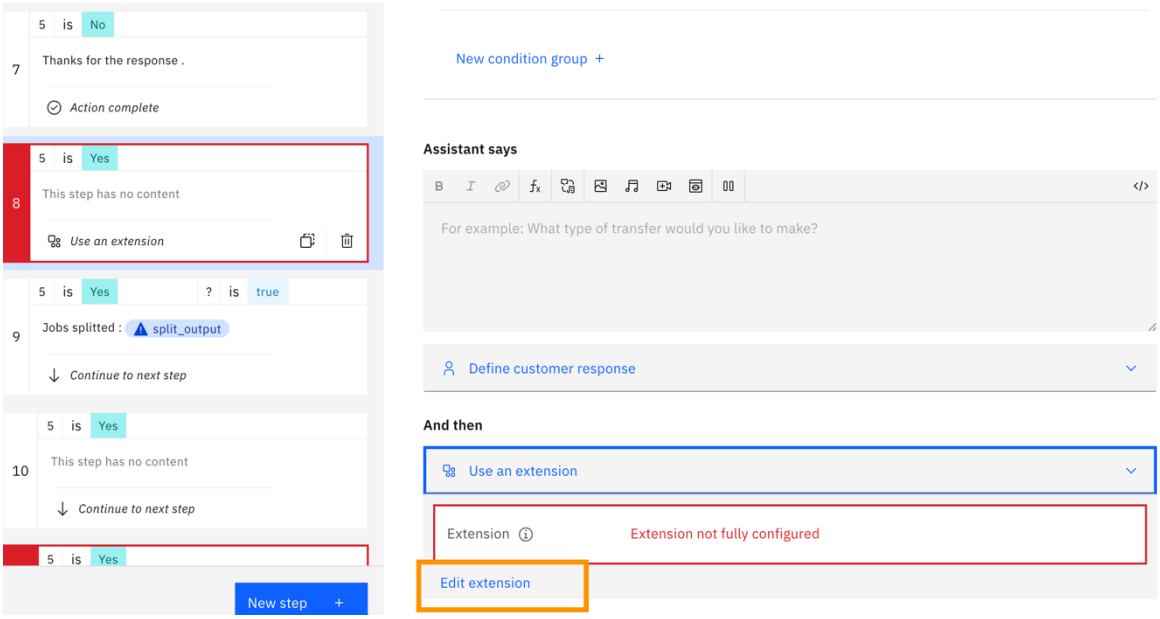


Set *Status to Active* →Click on Apply



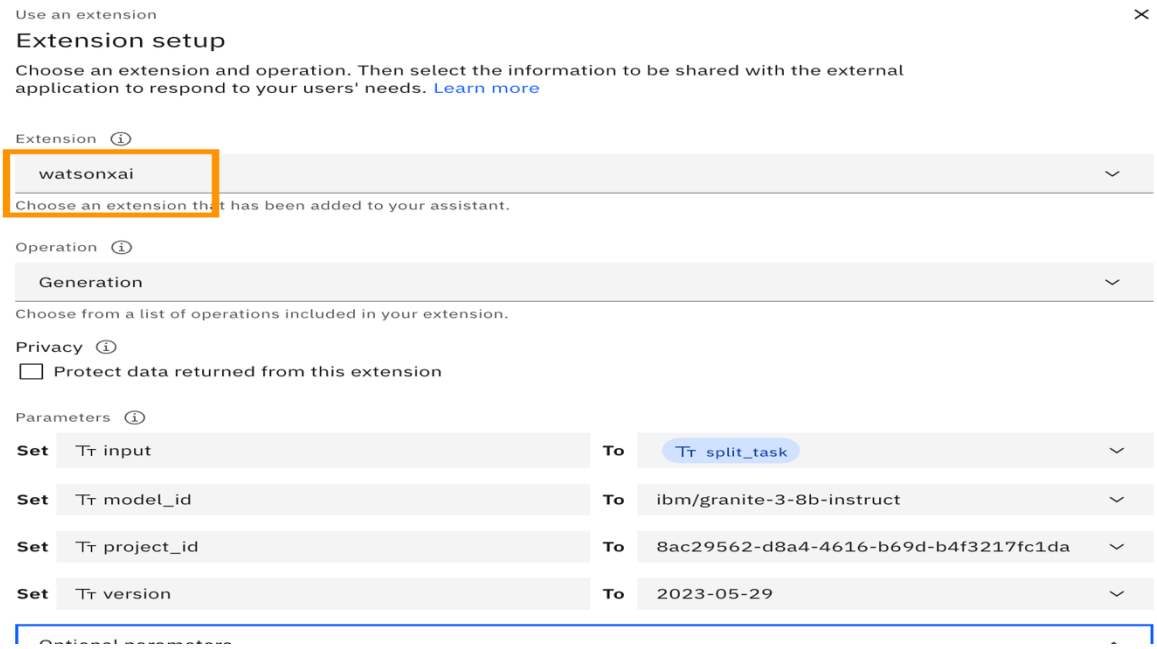


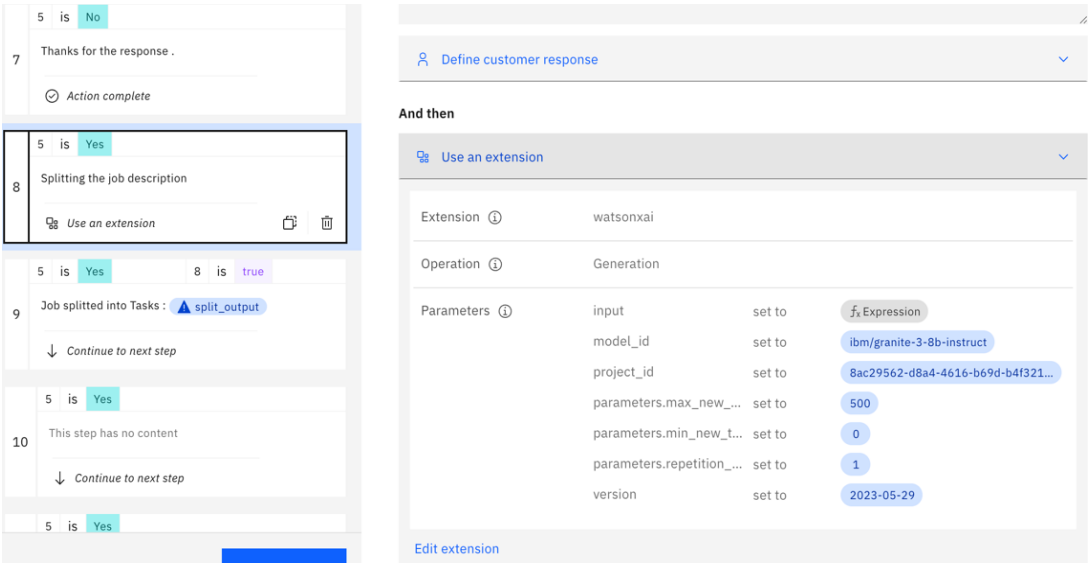
8.1.8 Select Step 8 and click on *Edit extension* and use ***watonx.ai extension.***



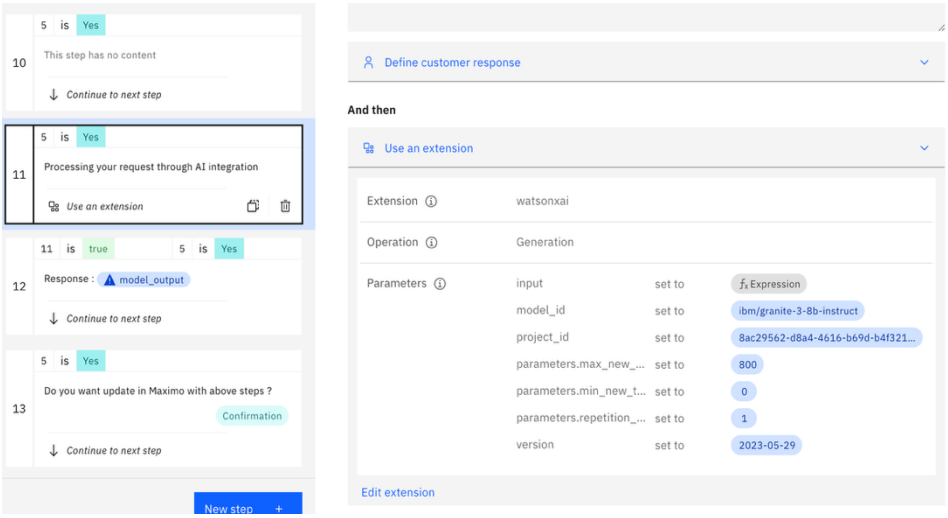
8.1.9 Update below fields

Field name	Fetching from
input	Previous step output (Ex: split_task)
Project_id	Project id from watsonx.ai →Step
Model_id	ibm/granite-3-8b-instruct
Version	2023-05-29
Max_tokens	500 & above (based on generated answer)
Min_tokens	0

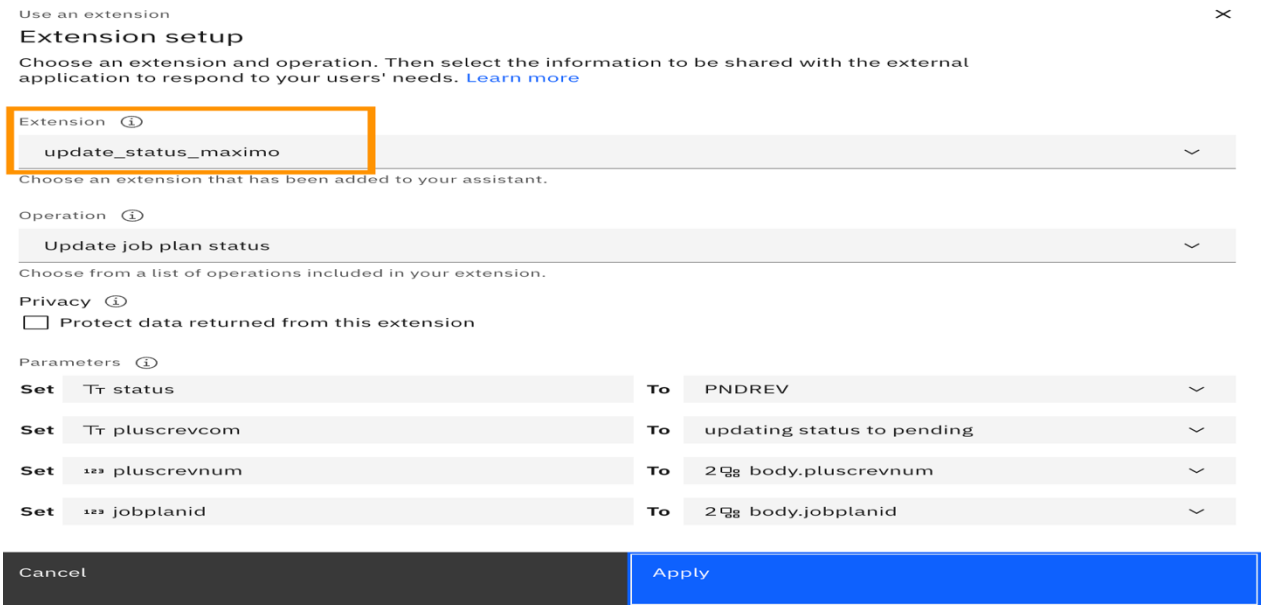


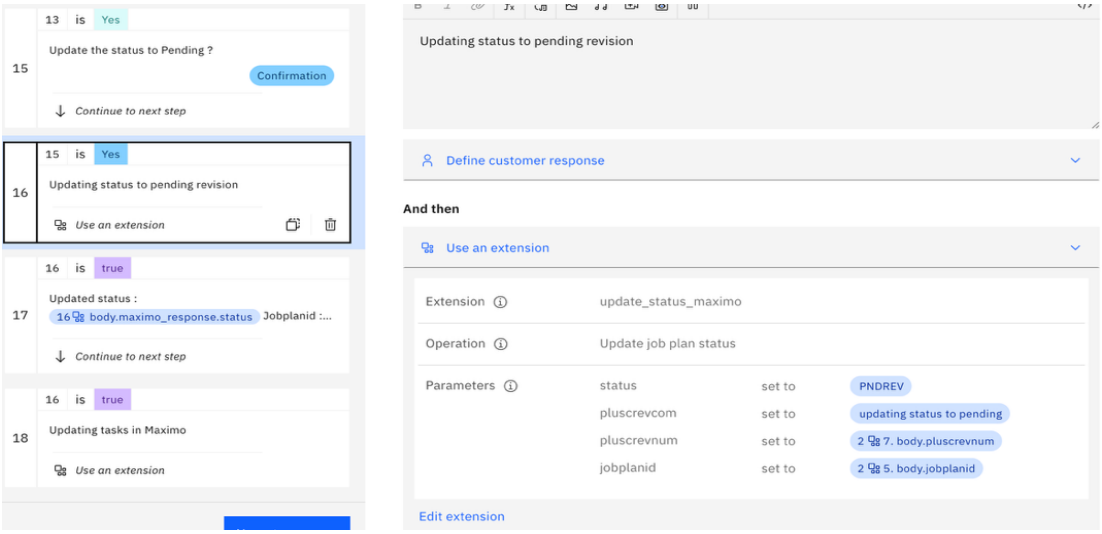


8.1.10 Select Step11 and Click on *Edit extension* and use ***watonx.ai extension***. (follow 7.1.8 &7.1.9)

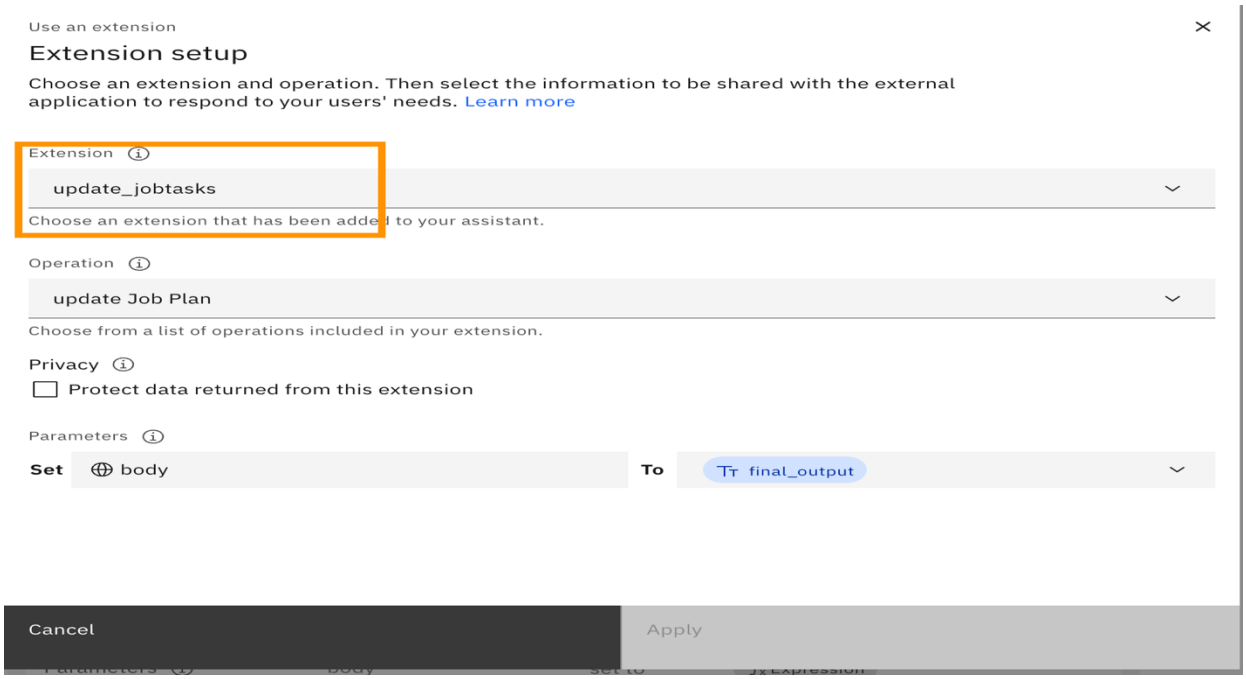


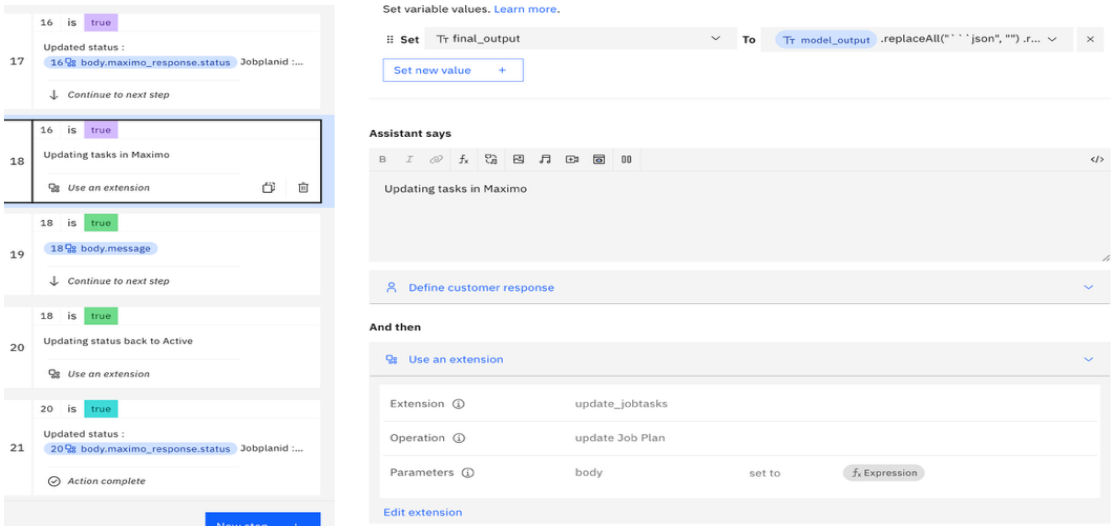
8.1.11 Select Step 16 and click on *Edit extension* and use Maximo **update_status_maximo** extension to update the *status from Active to Pending revised.*



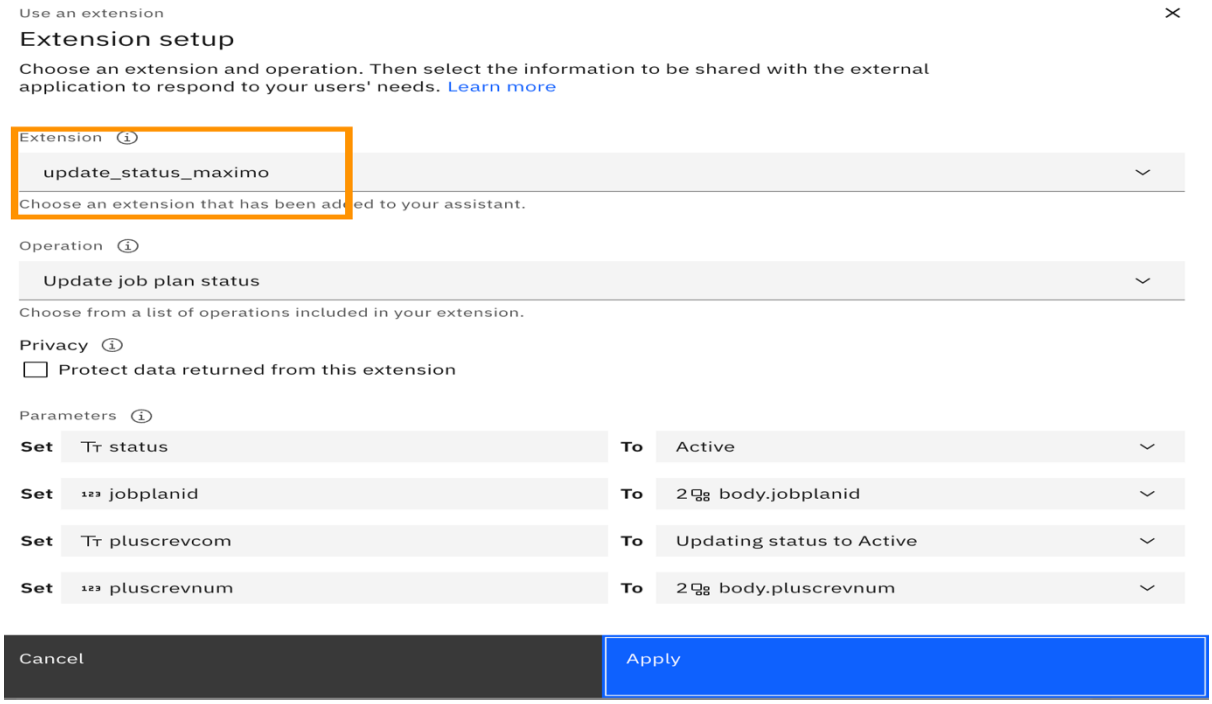


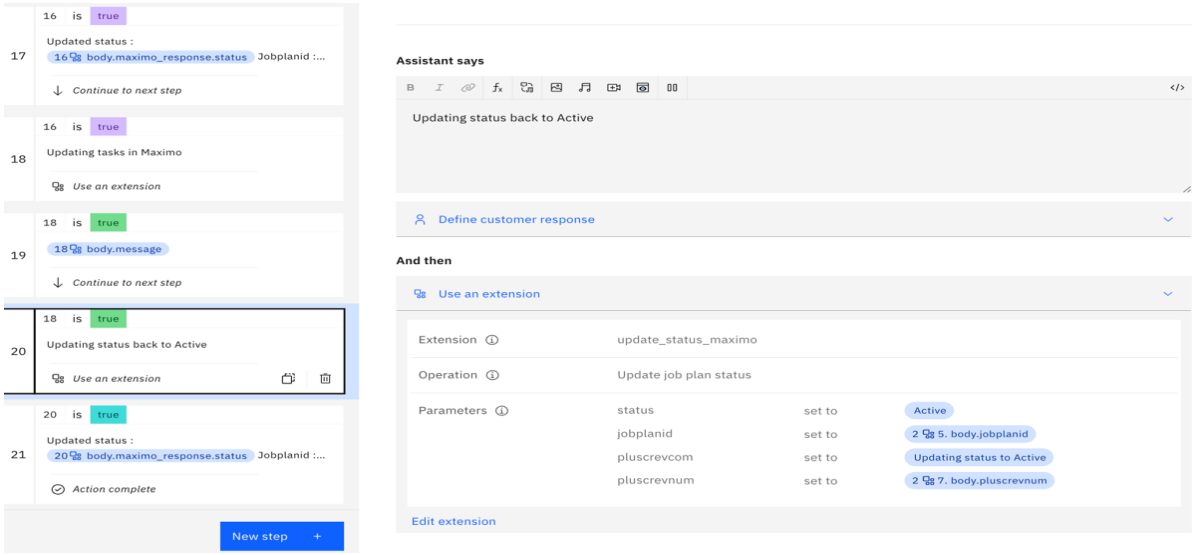
8.1.12 select Step 18 and click on *Edit extension* and use Maximo **update_jobtasks** extension to update the job tasks in Maximo.



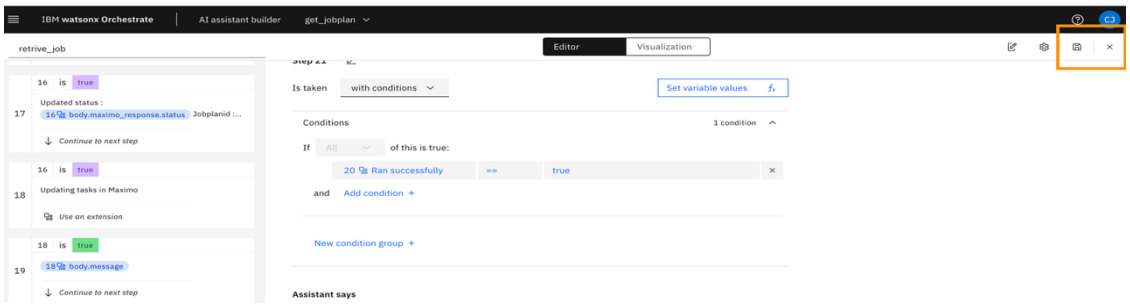


8.1.13 Select Step 20 and click on *Edit extension* and use Maximo **update_status_maximo** extension to update the *status from Pending revised to Active*.

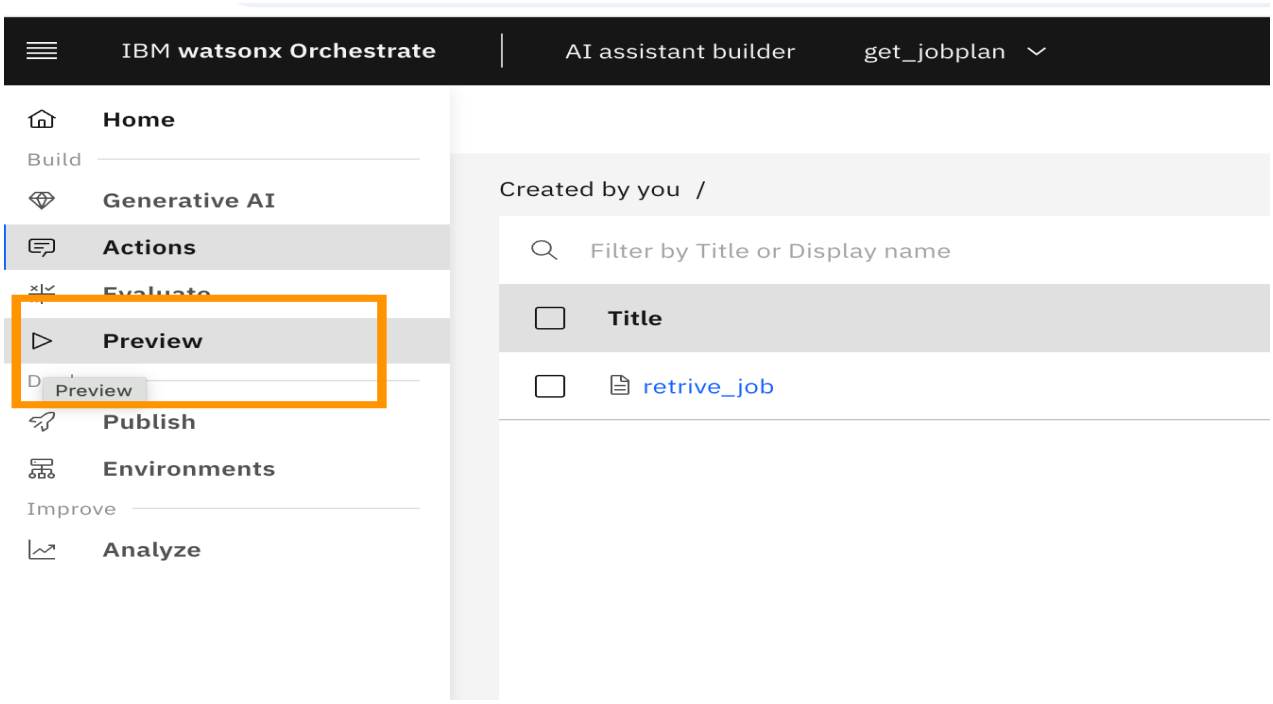




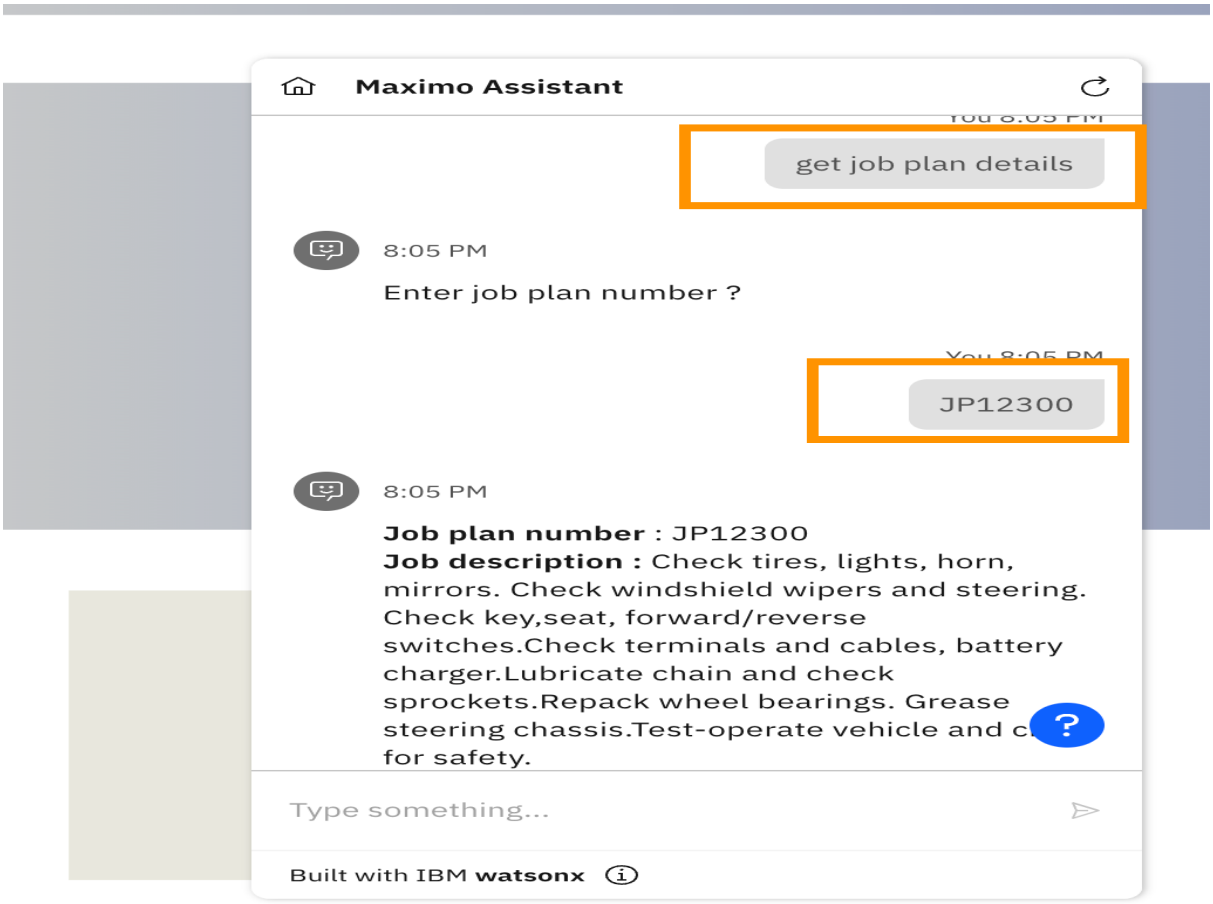
8.1.14 Save the *Actions*. → Click on *Close*



8.1.15 From *left* Navigation Click on *Preview*



8.1.16 Ask Assistant to fetch job plan details and provide job plan number



9. Conclusion

Transforming long, unstructured job plan descriptions into clearly defined job tasks significantly improves the usability and effectiveness of maintenance planning in IBM Maximo.

By automating this process based on the provided job plan number, organizations can:

- Convert complex instructions into simple, actionable steps
- Enhance clarity for technicians executing the work
- Ensure consistency in task execution across teams
- Reduce interpretation errors and operational delays

This approach not only improves field-level execution but also strengthens the integrity and maintainability of job plans in the system. As a result, maintenance teams can operate with greater efficiency, confidence, and alignment to best practices—making Maximo job plans a more powerful tool in day-to-day asset management operations.