

Maximo Agents

Technical Hands-on-Lab: Using
Watsonx Orchestrate to Manage Multi-
Agents AI in Maximo



1. Overview

This guide is designed to help you quickly set up and deploy AI agents that interact seamlessly with IBM Maximo to automate and enhance asset management tasks.

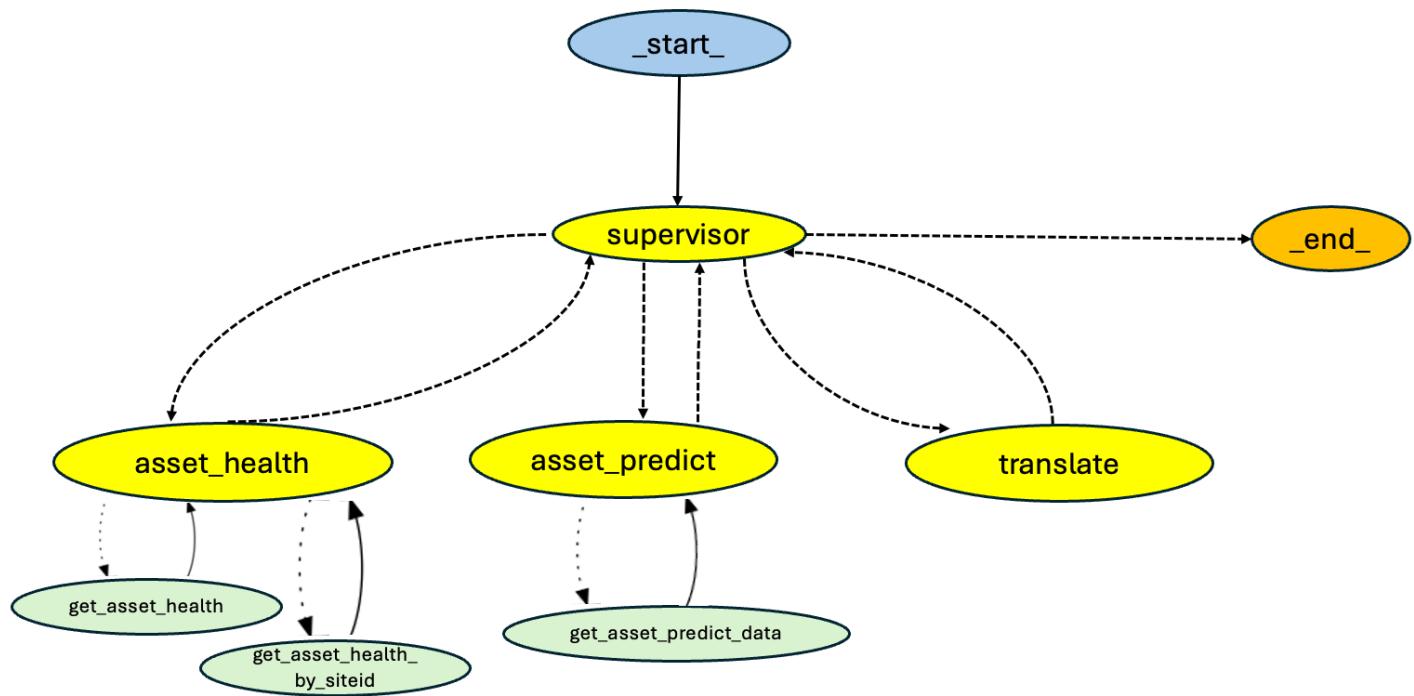
These agents, powered by large language models from the watsonx.ai platform, demonstrate the power of generative AI in enterprise environments. In this guide, you'll learn how to configure AI agents capable of:

- Retrieving asset health data for specific assets
- Listing all assets at a given location
- Generating predictive insights using historical and real-time data
- Translating responses to support multilingual users

Each agent can invoke tools through a unified user interface, simplifying interactions and decision-making for operations teams. While prebuilt tools are available, the true value comes from customizing and extending the toolkit to meet your unique business needs—accelerating innovation and delivering tangible results.

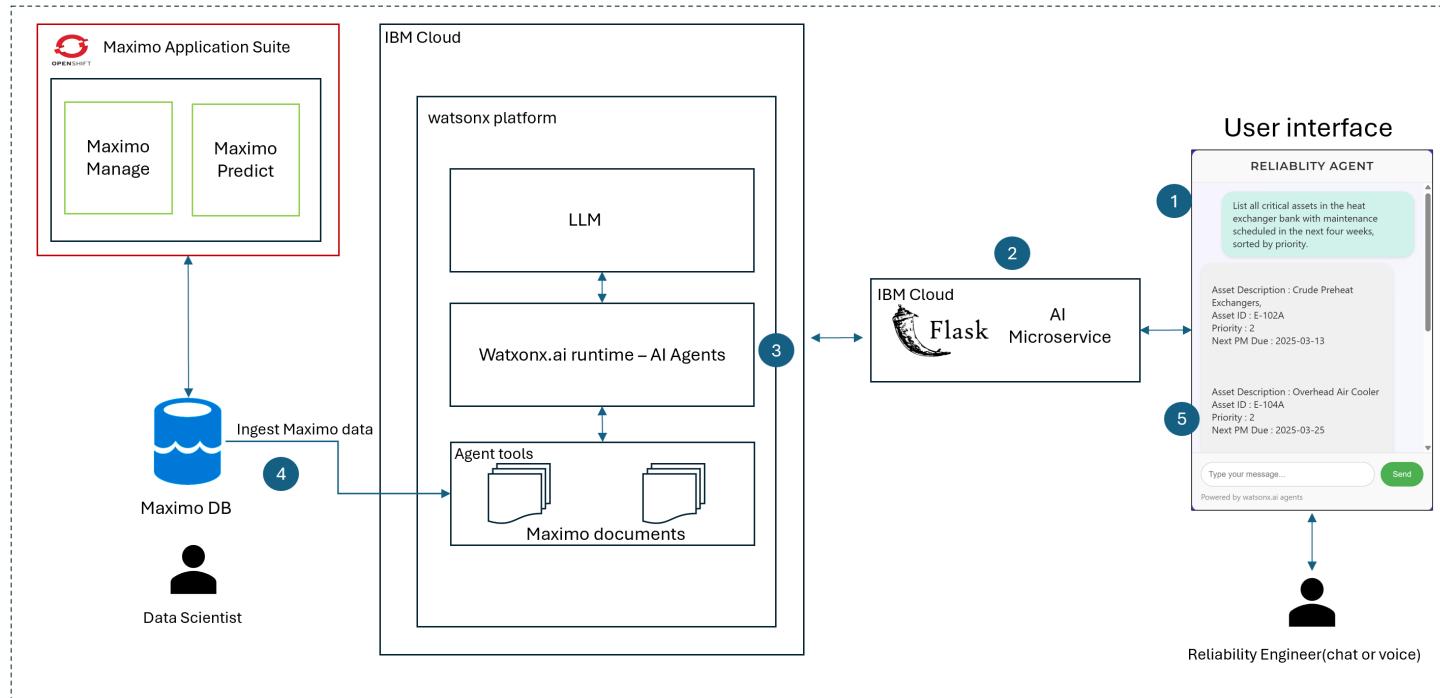
2. Agent Flow

The illustration below defines the control flow used by the wxo agents:



3. AI Agents Setup

Reference Architecture



3.1 Pre-requisite steps

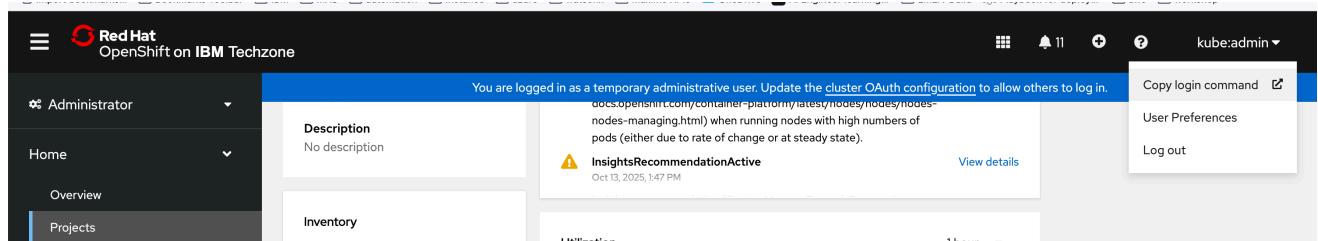
3.1.1

```
docker build -f Dockerfile --platform linux/amd64 -t <username>/maximo-agents:latest .
```

```
mehgajingar@Meghas-MacBook-Pro asset_health_by_assetnum % docker build -f Dockerfile --platform linux/amd64 -t md8911/get-asset-health:latest
[+] Building 28.7s (11/11) FINISHED
   => [internal] load build definition from Dockerfile
   => => transferring dockerfile: 559B
   => [internal] load metadata for docker.io/library/python:3.11-slim
   => [auth] library/python:pull token for registry-1.docker.io
   => [internal] load .dockerignore
   => => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fd2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> => resolve docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fd2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> [internal] load build context
=> => transferring context: 4.94KB
=> CACHED [2/5] WORKDIR /app
=> [3/5] COPY requirements.txt .
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt
=> [5/5] COPY .
=> exporting to image
=> => exporting layers
=> => writing image sha256:98d7e6b7bdf4d88c53dfed7b2d3cebc923fe373fe5282634d14ba2c00cdd9510c
=> => naming to docker.io/md8911/get-asset-health:latest
```

```
docker push <username>/get-asset-health:latest
```

```
[mehhajingar@Meghas-MacBook-Pro asset_health_by_assetnum % docker push md8911/get-asset-health:latest
The push refers to repository [docker.io/md8911/get-asset-health]
dce2f27d57bb: Pushed
15ed3ca5db77: Pushed
eefb578d096d8: Pushed
033ccb348dc8: Mounted from md8911/maximo-jobplan
b2738b04de4b: Mounted from md8911/maximo-jobplan
dba5cbed1e08: Mounted from md8911/maximo-jobplan
c9cf0647c388: Mounted from md8911/maximo-jobplan
1d46119d249f: Mounted from md8911/maximo-jobplan
latest: digest: sha256:b747f9b3d3816db03e35e9536a23b183d5cb47b592ac217821332a6133327e1b size: 1991
mehhajingar@Meghas-MacBook-Pro asset health by assetnum %
```



Your API token is

[Log in with this token](#)

```
oc login --token=sha256~x1157_kTz7oeVxtibm-uvM0a1X0T5qFHi1oJbU9DDA4 --server=https://api-68e7b6dc72e06a9c307a9ded.am1.techzone.ibm.com:6443
```

```
[mehajingar@Megha's-MacBook-Pro asset_health_by_assetnum % oc login --token=sha256-xJl57_kTz7oeVxtjhm-uyMqa1XQT5gFHloJhU9DDdA4 --server=https://api.68e7b6dc72e06a9c307a9ded.aml.techzone.ibm.com:6443
Logged into "https://api.68e7b6dc72e06a9c307a9ded.aml.techzone.ibm.com:6443" as "kube:admin" using the token provided.

You have access to 91 projects, the list has been suppressed. You can list all projects with 'oc projects'

Using project "default".
```

```
oc create secret generic maximo-env-secret --from-env-file=.env --namespace=maximo-agent-lab
```

```
[meghajingar@Meghas-MacBook-Pro asset_health_by_assetnum % oc create secret generic maximo-env-secret --from-env-file=.env --namespace=maximo-agent-lab
secret/maximo-env-secret created
meghajingar@Meghas-MacBook-Pro asset_health_by_assetnum %
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: get-asset-health
  namespace: maximo-agent-lab
spec:
  replicas: 1
  selector:
    matchLabels:
      app: get-asset-health
  template:
    metadata:
      labels:
        app: get-asset-health
    spec:
      containers:
        - name: get-asset-health
          image: md8911/get-asset-health:latest
          ports:
            - containerPort: 5000
          envFrom:
            - secretRef:
                name: maximo-env-secret # Reference to your secret
```

The screenshot shows the OpenShift web interface. On the left, there's a sidebar with navigation links: API Explorer, Events, Operators, Workloads (with sub-options like Pods, Deployments, DeploymentConfigs, StatefulSets, Secrets, ConfigMaps), CronJobs, Jobs, DaemonSets, ReplicaSets, ReplicationControllers, HorizontalPodAutoscalers, PodDisruptionBudgets, and Networking. The 'Deployments' link under 'Workloads' is highlighted. The main content area has a header bar with 'Project: maximo-agent-lab' and a message about cluster OAuth configuration. Below this, it says 'Deployments > Deployment details' and shows a card for 'get-asset-health'. The card has tabs for Details, Metrics, YAML, ReplicaSets, Pods, Environment, and Events. The 'Details' tab is selected. It displays deployment details: Name (get-asset-health), Namespace (maximo-agent-lab), Labels (No labels), Pod selector (app=get-asset-health), Node selector (None), Update strategy (RollingUpdate), Max unavailable (25% of 1 pod), Max surge (25% greater than 1 pod), Progress deadline seconds (600 seconds), and Min ready seconds (Not configured). A large blue circle icon indicates 1 Pod.

```
apiVersion: v1
kind: Service
metadata:
  name: get-asset-health
  namespace: maximo-agent-lab
spec:
  selector:
    app: get-asset-health
  ports:
```

```
- protocol: TCP
  port: 80
  targetPort: 5000
type: ClusterIP
```

Project: maximo-agent-lab ▾

Import YAML

Drag and drop YAML or JSON files into the editor, or manually enter files and use `---` to separate each definition.

Opt + F1 Accessibility help

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: get-asset-health
5   namespace: maximo-agent-lab
6 spec:
7   selector:
8     app: get-asset-health
9   ports:
10    - protocol: TCP
11      port: 80
12      targetPort: 5000
13    type: ClusterIP
```

Create

Cancel

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: maximo-agent-lab ▾

Services > Service details

get-asset-health

Actions ▾

Details YAML Pods

Service details		Service routing			
Name	get-asset-health	Hostname	get-asset-health.maximo-agent-lab.svc.cluster.local		
Namespace	maximo-agent-lab	Accessible within the cluster only			
Labels	No labels	Service address	Type: Cluster IP, Location: 172.30.65.104		
Pod selector	app=get-asset-health	Annotations	0 annotations		
Annotations	0 annotations	Session affinity			
Session affinity		Service port mapping			
		Name	Port	Protocol	Pod port or name
		-	S 80	TCP	P 5000

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: maximo-agent-lab ▾

Routes

No Routes found

Create Route

- CronJobs
- Jobs
- DaemonSets
- ReplicaSets
- ReplicationControllers
- HorizontalPodAutoscalers
- PodDisruptionBudgets

Networking ▾

- Services
- Routes**
- Ingresses
- NetworkPolicies

 Red Hat
OpenShift on IBM Techzone

You are logged in as a temporary administrative user. Update the [cluster OAuth configuration](#) to allow others to log in.

Project: maximo-agent-lab ▾

Name *
get-asset-health-by-assetnum

A unique name for the Route within the project.

Hostname
www.example.com

Public hostname for the Route. If not specified, a hostname is generated.

Path
/

Path that the router watches to route traffic to the service.

Service *
S get-asset-health

Service to route to.

Target port *
80 → 5000 (TCP)

Target port for traffic.

Security

Secure Route
Routes can be secured using several TLS termination types for serving certificates.

Create **Cancel**

- DeploymentConfigs
- StatefulSets
- Secrets
- ConfigMaps
- CronJobs
- Jobs
- DaemonSets
- ReplicaSets
- ReplicationControllers
- HorizontalPodAutoscalers
- PodDisruptionBudgets

Networking ▾

- Services
- Routes**
- Ingresses
- NetworkPolicies

Storage ▾

Builds ▾

Project: maximo-agent-lab ▾

Routes > Route details

RT get-asset-health-by-assetnum Accepted

Details Metrics YAML

Route details

Field	Value
Name	get-asset-health-by-assetnum
Namespace	NS maximo-agent-lab
Labels	No labels
Annotations	1 annotation
Service	S get-asset-health
Target port	5000
Location	http://get-asset-health-by-assetnum-maximo-agent-lab.apps.68e7b6dc72e06a9c307a9ded.am1.techzone.ibm.com
Status	Accepted
Host	get-asset-health-by-assetnum-maximo-agent-lab.apps.68e7b6dc72e06a9c307a9ded.am1.techzone.ibm.com
Path	-
Router canonical hostname	router-default.apps.68e7b6dc72e06a9c307a9ded.am1.techzone.ibm.com

Copy to clipboard

3.2 Login to Cloud Account

3.1.1 Open -> <https://cloud.ibm.com/>

3.1.2 Select the appropriate cloud environment.

The screenshot shows the IBM Cloud dashboard with the URL 'cloud.ibm.com' in the address bar. The top navigation bar includes links for Import bookmarks, Bookmarks Toolbar, IBM, MAS, automation, Instance, azure, watsonx, Maximo APIs, ISL Learning Experi..., OneDrive, AI Engineer learning..., and EMEA-Build. The main menu has 'IBM Cloud' selected. A search bar says 'Search resources and products...'. The left sidebar shows 'Dashboard' and other options like 'Compute', 'Containers', 'Networking', 'Storage', 'Converged infrastructure', 'Enterprise applications', 'AI / Machine Learning', and 'Analytics'. The central area is titled 'Resource list' with a 'Create resource' button. It features a table with columns: Name, Group, Location, Product, Status, and Tags. The table contains four entries under 'AI / Machine Learning': Watson OpenScale-itz, Watson Orchestrate-itz, wml-itz-wxo-6841271fc37d1f363a7064, and ws-itz-wxo-6841271fc37d1f363a7064. All entries are active.

Name	Group	Location	Product	Status	Tags
Watson OpenScale-itz	watsonx	Dallas (us-south)	watsonx.governance	active	-
Watson Orchestrate-itz	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx Orchestrate	active	-
wml-itz-wxo-6841271fc37d1f363a7064	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx.ai Runtime	active	-
ws-itz-wxo-6841271fc37d1f363a7064	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx.ai Studio	active	-

3.1.3 Select Resource list on the left menu and expand the AI/ Machine Learning.

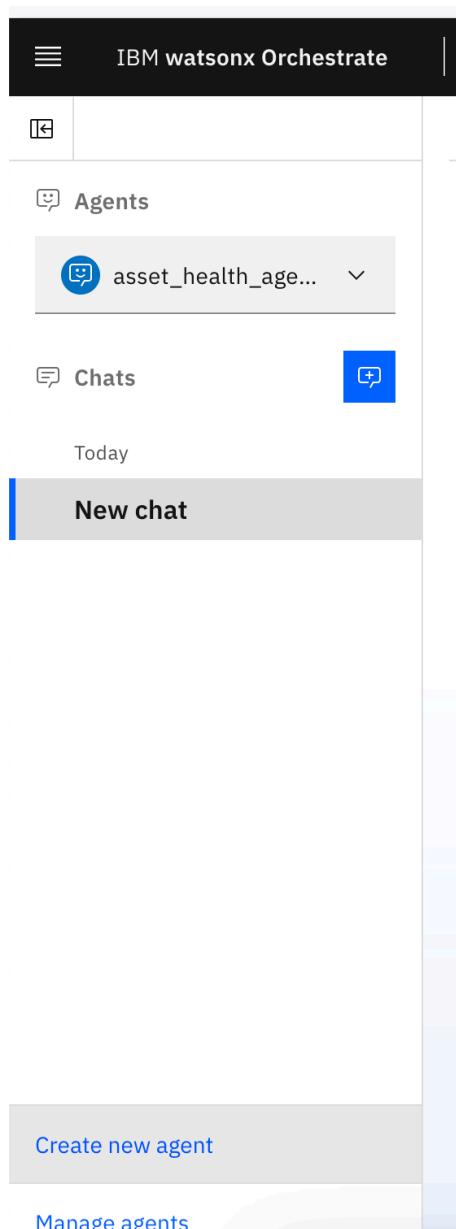
This screenshot shows the 'Resource list' table from the previous image, specifically focusing on the 'AI / Machine Learning' section. The table has columns: Name, Group, Location, Product, Status, and Tags. There are four entries listed:

Name	Group	Location	Product	Status	Tags
Watson OpenScale-itz	watsonx	Dallas (us-south)	watsonx.governance	active	-
Watson Orchestrate-itz	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx Orchestrate	active	-
wml-itz-wxo-6841271fc37d1f363a7064	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx.ai Runtime	active	-
ws-itz-wxo-6841271fc37d1f363a7064	itz-wxo-6841271fc37d1f363a7064	Tokyo (jp-tok)	watsonx.ai Studio	active	-

3.3 Setting up your Agent

3.2.1 Open your IBM Watsonx Orchestrate instance.

3.2.2 Click on “Create new agent”



- 3.2.3 Enter the name as “asset_health_agent” and description as “You are an API agent that supports Maximo operations. You handle back-end interactions such as retrieving asset health details.”

Click on Create button.

Create an agent

Create from scratch

Build your custom agent step by step to create a custom solution.

Create from template

Browse the catalog and use the attributes of another agent as a template to create your agent.

Name*

Description*

You are an API agent that supports Maximo operations. You handle back-end interactions such as retrieving asset health details.

Describe your agent's purpose.

Cancel
Create

3.2.3 The agent is created and it should look like this.

The screenshot shows the IBM Watsonx Orchestrate interface. At the top, there is a navigation bar with the title "IBM Watsonx Orchestrate". Below the navigation bar, the URL "Agent chat / Manage agents / asset_health_agent" is visible. On the right side of the header, there are several icons: a trial days left indicator (26), a deployment status icon, a help icon, and a user profile icon. A blue "Deploy" button is also present in the header area. The main content area is titled "asset_health_agent" with a preview icon. The interface is divided into sections: "Profile", "Knowledge", "Toolset", and "Behavior". The "Profile" section is currently active, showing a "Profile" card with a "Description" field containing the text: "You are an API agent that supports Maximo operations. You handle back-end interactions such as retrieving asset health details.". The "Knowledge" section contains a "Knowledge" card with a note about the importance of descriptions. To the right of the "Profile" section, there is a "Preview" panel displaying a welcome message: "Hello, welcome to watsonx Orchestrate." and a text input field "Type something...".

3.2.4 Scroll down and under the toolset, click on “Add tool” button.

The screenshot shows the 'asset_health_agent' configuration interface. On the left, there's a vertical navigation bar with tabs: Profile, Knowledge, Toolset (which is selected and highlighted in blue), and Behavior. The main content area has a header 'Toolset' with a sub-header 'Add tools and other agents to help your agent take action.' Below this is another section titled 'Tools' with the sub-header 'Give your agent access to your organization's tool to help it accomplish tasks.' A large central box contains a placeholder icon of a cube and the text 'Start by adding a tool' followed by 'To add a tool, click Add tool.' At the top right of this central box is a button labeled 'Add tool' with a '+' sign.

3.2.5 Select the “import” tile.

This screenshot shows the 'asset_health_agent' configuration interface with the 'Toolset' tab selected. A modal dialog box titled 'Add a new tool' is open over the main content. The dialog has a 'Preview' tab at the top right. Inside, there are four options: 'Add from catalog' (selected, shown in light gray), 'Add from local instance' (shown in white), 'Import' (shown in white), and 'Create a new flow' (shown in white). The 'Import' option is described as 'Import an external tool.' The 'Add from catalog' option is described as 'Choose from a catalog of existing tools.' The 'Add from local instance' option is described as 'Choose from a list of tools available on this instance.' The 'Create a new flow' option is described as 'Use the tool builder to create a sequence of steps that utilizes conditional controls and activities like document processors.'

3.2.4 Browse and select “get-asset-health-openapi.json” file and click on Next button.

This screenshot shows the 'Import tool' dialog. At the top, it says 'Import tool'. Below that are two buttons: 'Upload file' (selected) and 'Select operations'. A note below says 'Upload an OpenAPI specification file. Max file size is 500kb. Supported file types are .json and .yaml.' A large input field is present with the placeholder 'Drag and drop an OpenAPI file here or click to upload.' Below this, a file named 'get-asset-health-openapi.json' is listed with a delete 'x' icon. A green checkmark icon indicates 'Validation successful'. At the bottom are 'Cancel' and 'Next' buttons, with 'Next' being highlighted in blue.

3.2.5 Select the operations and click on Done.

The screenshot shows the 'Import tool' interface. At the top, there are two buttons: 'Upload file' and 'Select operations'. Below them, a section titled 'Operations' contains a table with one item: 'Get asset health by asset number' (Method: GET). At the bottom right of the table are buttons for 'Cancel' and 'Done'.

3.2.6 Now, in the preview box, ask something like this “can you get me health details for asset?”, Watson Orchestrate should ask for asset number, Enter the asset number e.g 11430

The screenshot shows the Watson Orchestrate interface. On the left, under 'Toolset', there is a 'Tools' section with a 'Get asset health by asset number (2)' entry. On the right, in the 'Preview' window, a conversation is shown between a user and an agent named 'asset_health_agent'. The user asks 'can you get me health details for asset?'. The agent responds with 'What is assetnum?' and then provides the health details for asset 11430, including its description as a 'Centrifugal Pump 100GPM/60FT HD'.

3.2.7 Click on the three dots in tools and select “Edit details” to update the description.

The screenshot shows the 'Tools' configuration interface. It lists two tools: 'Get asset health by assetnum' and 'Get asset health by site ID'. The 'Get asset health by assetnum' tool has a 'More' button (three dots) which is highlighted with a blue box. A tooltip 'Edit details' is shown above the 'More' button.

Name	Get asset health by assetnum
Description	<p>Operation: <code>get__get-asset_{assetnum}</code></p> <p>Use the Get asset health by assetnum tool to fetch health metadata for a specific asset. The user will provide an assetnum.</p>

	<p>The response will contain LLM-friendly field names like <code>asset_number</code>, <code>health_score</code>, <code>install_date</code>, and <code>criticality_score</code> for easier reasoning and summarization. <code>health_score</code> is a mandatory column.</p> <p>Provide the output in easy to read format each field in next line.</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tool information

Name your tool, give it a description and define its parameters

Importance of descriptions
A clear and detailed tool description helps an AI agent understand when to use the tool, what inputs are required, and how to interpret the output, leading to more accurate and relevant responses of users.

Name
Get asset health by assetnum

Description
Operation: `get_asset_(assetnum)`
Use the Get asset health by `assetnum` tool to fetch health metadata for a specific asset. The user will provide an `assetnum`. The response will contain LLM-friendly field names like `asset_number`, `health_score`, `install_date`, and `criticality_score`

Detailed descriptions will help guide your agents to effectively use your tools

Inputs (1)

Type	Name	Description	Default	Required
string	path_assetnum	Asset number to fetch health info		On

Cancel Save changes

Select Save changes.

3.2.8 we can try with few other questions as showed below.

Can you give me description for asset 11430?

Preview AI

You 03:38 PM

can you give me description for asset 11430?

asset_health_agent 03:38 PM | Show Reasoning ▾
The description for asset 11430 is:
Centrifugal Pump 100GPM/60FT HD

You 03:38 PM

what's the priority of asset 11430?

asset_health_agent 03:38 PM | Show Reasoning ▾
The priority of asset 11430 is 2.

You 03:46 PM

what's the installation date ?

asset_health_agent 03:46 PM | Show Reasoning ▾
The installation date of asset 11430 is April 7, 1995.