

Enabling Business User Interaction with IBM OpenPages through AI Agent

Automating compliance management in OpenPages with an AI
agent

Krishna Damarla, Aug 9, 2025

Summary

This presentation demonstrates how to seamlessly integrate AI agents into existing enterprise software platforms to drive automation and enhance user support.

It details the architecture and step-by-step methodology for building, deploying and embedding AI agents within IBM OpenPages—a leading Governance, Risk, and Compliance (GRC) solution.

AI agents built within IBM Watsonx Orchestrate are designed to help business users in regulated industries to efficiently manage compliance requests (BPM), streamline business process workflows and improve operational efficiency.

Use Case: Automating Compliance Tasks

The following is an example use case for enabling business users to interact with OpenPages via an AI agent to automate the resolution of compliance issues

Compliance Issue	Business User Action	Agent Action in OpenPages UI
New compliance issues reported	Assign all new compliance issues to Agent	Create new cases in OpenPages
Compliance issue investigation complete	Approving all compliance issues that have been investigated and have a low-risk rating	Update case status in OpenPages
Compliance issue remediation required	Taking action to mitigate all high-risk compliance issues	Create new action task in OpenPages

Steps Involved

1: Define the Requirements

- Identify the specific compliance issues that the business user needs to take action on
- Determine the actions that the business user needs to perform (e.g. assign, approve, reject)
- Define the rules and workflows that govern the compliance issue lifecycle

2: Configure the IBM OpenPages Environment

- Set up the OpenPages environment to allow for agent-based interaction
- Configure the necessary workflows, rules, and permissions to enable business user interaction
- Ensure that the OpenPages environment is properly integrated with other relevant systems

3: Develop the Agent Chatbot Interface

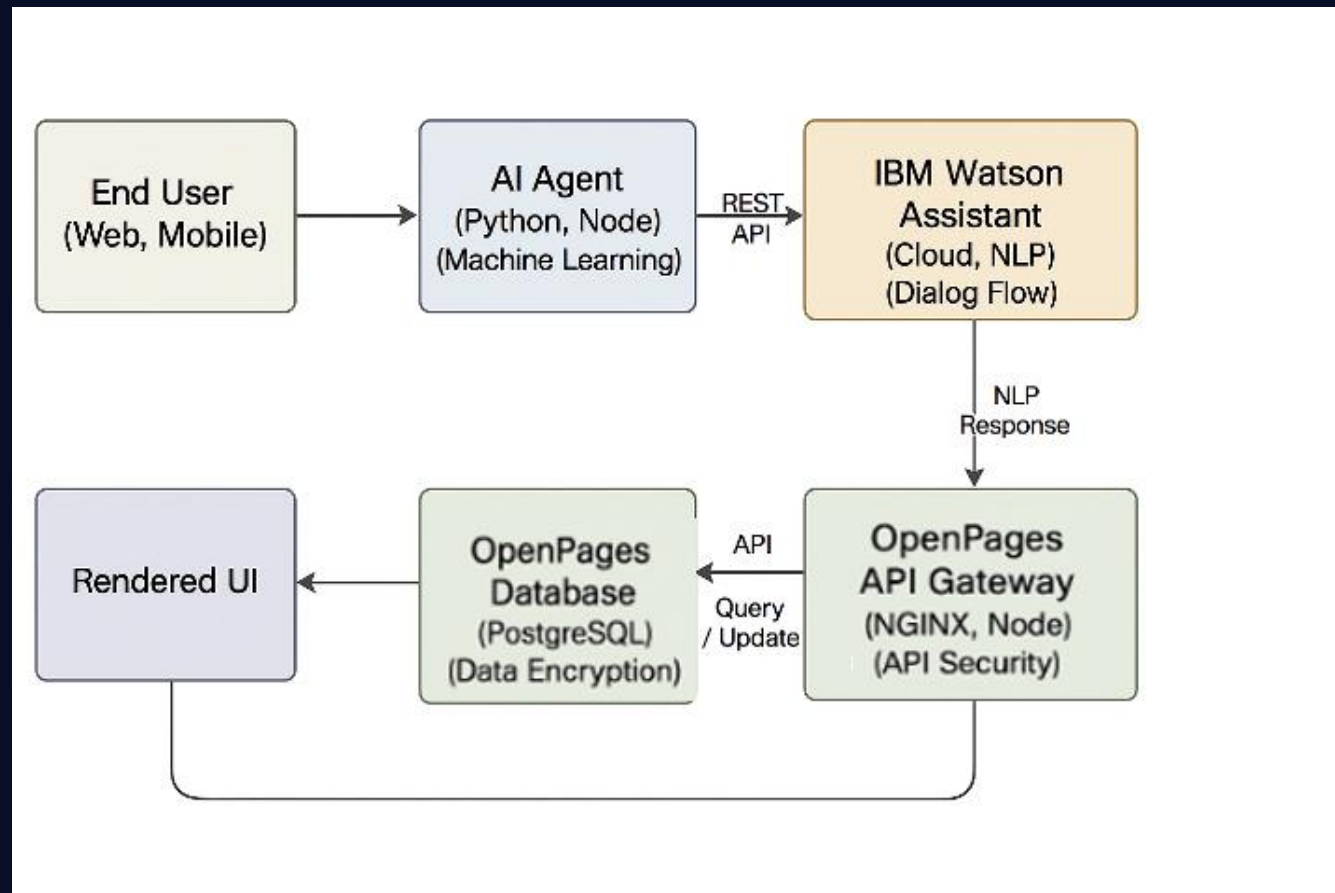
- Develop Watsonx AI agent & Chatbot interface that allows business users to interact with OpenPages
- Ensure that the interface provides clear guidance on the actions that can be taken

4: Implement Workflow Automation

- Implement workflow automation to streamline the compliance issue lifecycle
- Use BPM tools to define and execute workflows
- Ensure that workflows are properly integrated with the OpenPages environment

Architecture Overview: AI Agent Integration with OpenPages

Architecture

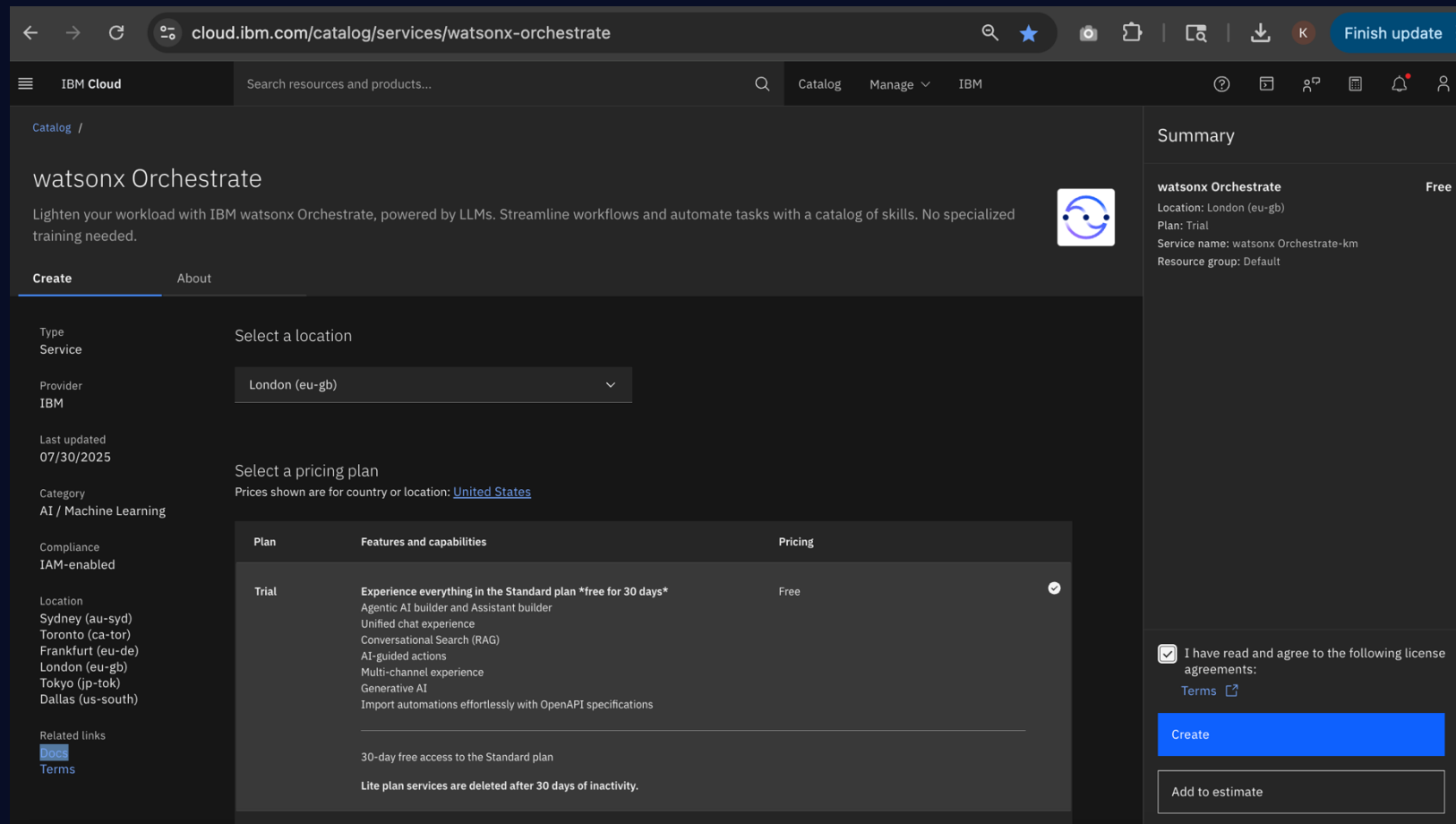


In this diagram, the flow is as follows:

- The **End User** interacts with the **AI Agent** to get insights and automate tasks in OpenPages.
- The **AI Agent** sends REST API calls to the **IBM Watson Assistant** to leverage its natural language processing (NLP) capabilities.
- The **IBM Watson Assistant** processes the user's input using NLP and sends the output to the **OpenPages API Gateway**.
- The **OpenPages API Gateway** receives the API calls and authenticates the request.
- The **OpenPages API Gateway** then calls the **OpenPages Database** to retrieve or update data.
- The **OpenPages Database** stores and retrieves data related to OpenPages.
- The **OpenPages UI** provides a user interface for users to interact with the system.

Building & Deploying AI Agent in Watsonx Orchestrate

Create watsonx orchestrate service in IBM Cloud



The screenshot shows the IBM Cloud catalog page for the watsonx Orchestrate service. The page is titled "watsonx Orchestrate" and includes a description: "Lighten your workload with IBM watsonx Orchestrate, powered by LLMs. Streamline workflows and automate tasks with a catalog of skills. No specialized training needed." The page is divided into two main sections: "Create" and "About". The "Create" section is active and shows a form for selecting a location and a pricing plan. The "About" section is visible on the right side of the page.

Summary

watsonx Orchestrate Free

Location: London (eu-gb)
Plan: Trial
Service name: watsonx Orchestrate-km
Resource group: Default

Create About

Type: Service

Provider: IBM

Last updated: 07/30/2025

Category: AI / Machine Learning

Compliance: IAM-enabled

Location: Sydney (au-syd), Toronto (ca-tor), Frankfurt (eu-de), London (eu-gb), Tokyo (jp-tok), Dallas (us-south)

Related links: [Docs](#), [Terms](#)

Select a location

London (eu-gb)

Select a pricing plan

Prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Trial	Experience everything in the Standard plan *free for 30 days* Agentic AI builder and Assistant builder Unified chat experience Conversational Search (RAG) AI-guided actions Multi-channel experience Generative AI Import automations effortlessly with OpenAPI specifications	Free

30-day free access to the Standard plan

Lite plan services are deleted after 30 days of inactivity.

☒ I have read and agree to the following license agreements:
[Terms](#)

[Create](#)

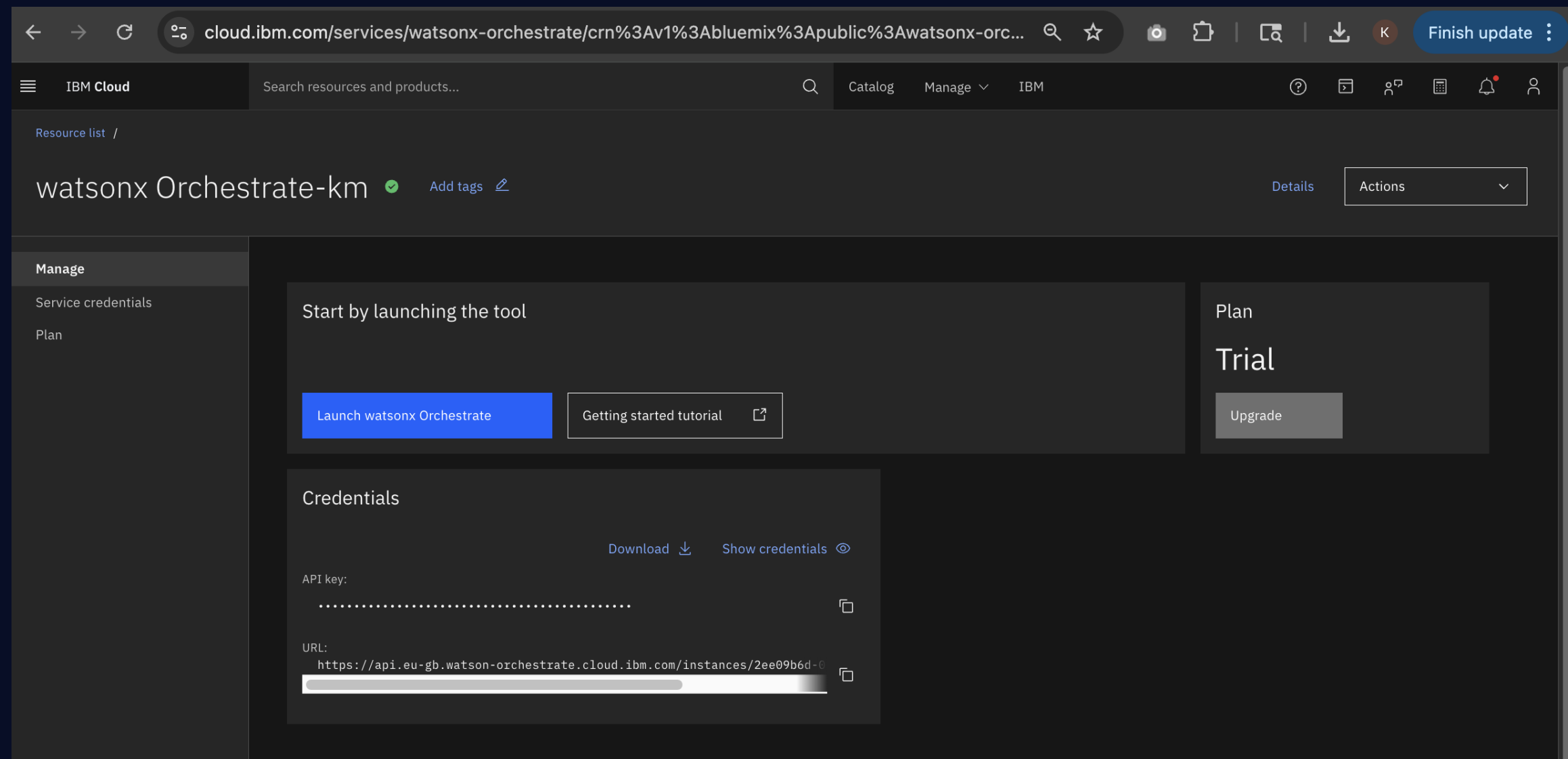
[Add to estimate](#)

Steps Involved

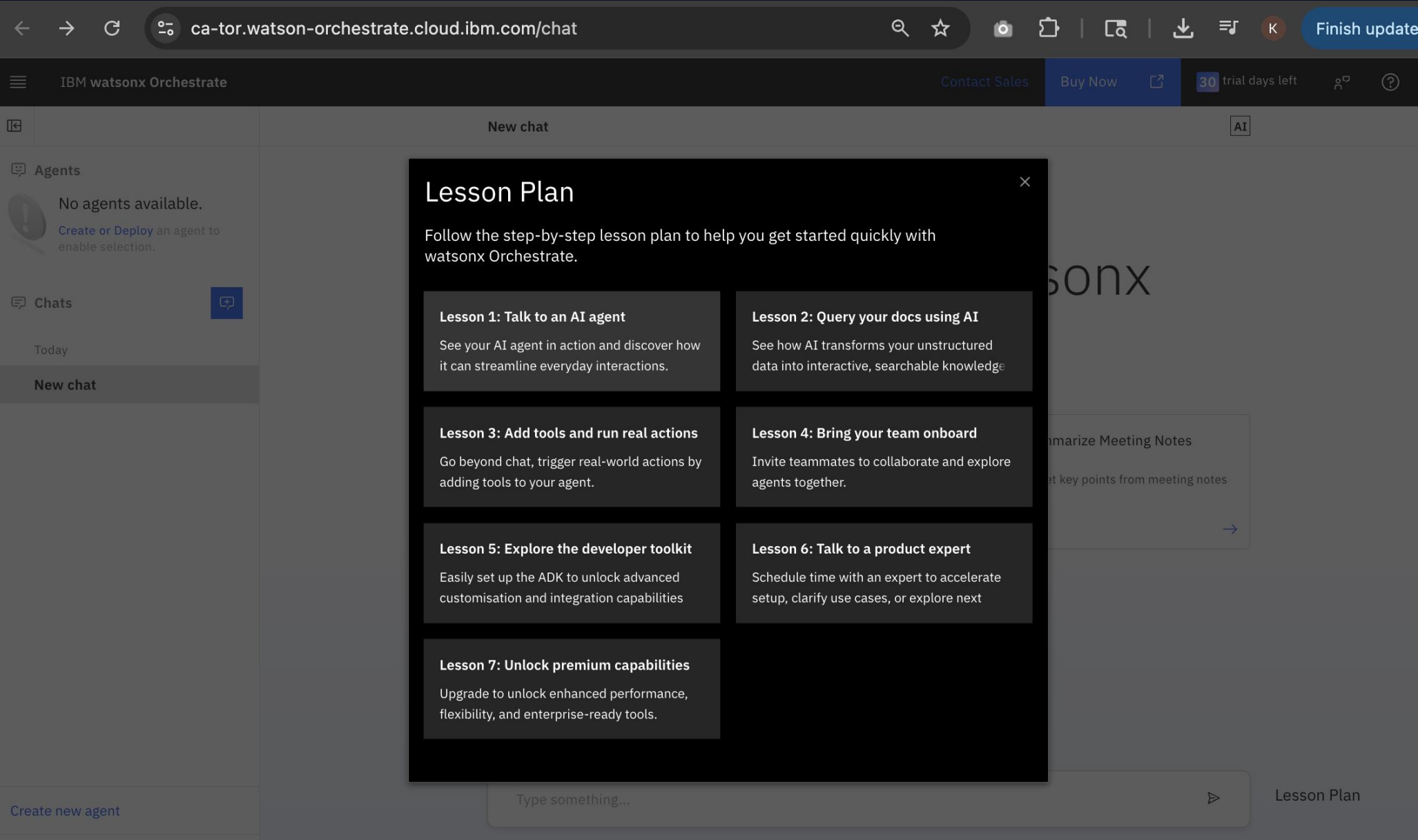
- Log in to the Watson Orchestrate dashboard and navigate to the "Agents" tab.
- Click on "Create a new agent" and follow the prompts to create a new AI agent.
- Configure the agent to use the Watson Orchestrate API.

Ref: <https://www.ibm.com/docs/en/watsonx/watson-orchestrate/base>

Launch watsonx orchestrate service in IBM Cloud



Watsonx Orchestrate Capabilities



Build AI Agent in Watsonx Orchestrate

IBM watsonx Orchestrate

[Contact Sales](#)[Buy Now](#)

30 trial days left

KD

[Agent chat](#) /

Build agents and tools

Total messages ⓘ

2

▲ 100% from the previous day

Failed messages ⓘ

0

▼ 0% from the previous day

Latency average ⓘ

3792_{ms}

▲ 100% from the previous day


[View all →](#)

All agents
1

All tools
7

Search for an agent

[Discover](#) →[Create agent](#) +



File management

An agent to manage files in Microsoft Sharepoint, with capabilities to create folders, and rename and download files.

Tools
7

Agents
—

→

Deploy Agent

IBM watsonx Orchestrate

Contact SalesBuy Now30 trial days left

KD

Agent chat / Manage agents / File management

Deploy

File management

Model : llama-3-2-90b-visi...

ProfileKnowledgeToolsetBehaviorChannels Preview

Describe what your agent does so people and other agents know when to use it.

Description*

An agent to manage files in Microsoft Sharepoint, with capabilities to create folders, and rename and download files.

Describe your agent, including its purpose, tools, and agents, to help other agents and users know when to use it.

What makes a good description?

Agent style

Define how your agent interprets prompts and decide what action to take.

Default Recommended

Relies on the models intrinsic ability to understand, plan and call tools and knowledge.

ReAct

Enables the model to think, act, observe and refine its approach until a task is completed.

Good for: High stakes apps

Preview

AI

Hello, welcome to watsonx Orchestrate

Accuracy of generated answers may vary. Please double-check responses.

What can you do for me?

Learn about watsonx Orchestrate AI Agent and Gen AI capabilities

→

Formalize Message

Helps you craft more professional messages

→

Summarize Meeting Notes

Bullet key points from meeting notes

→

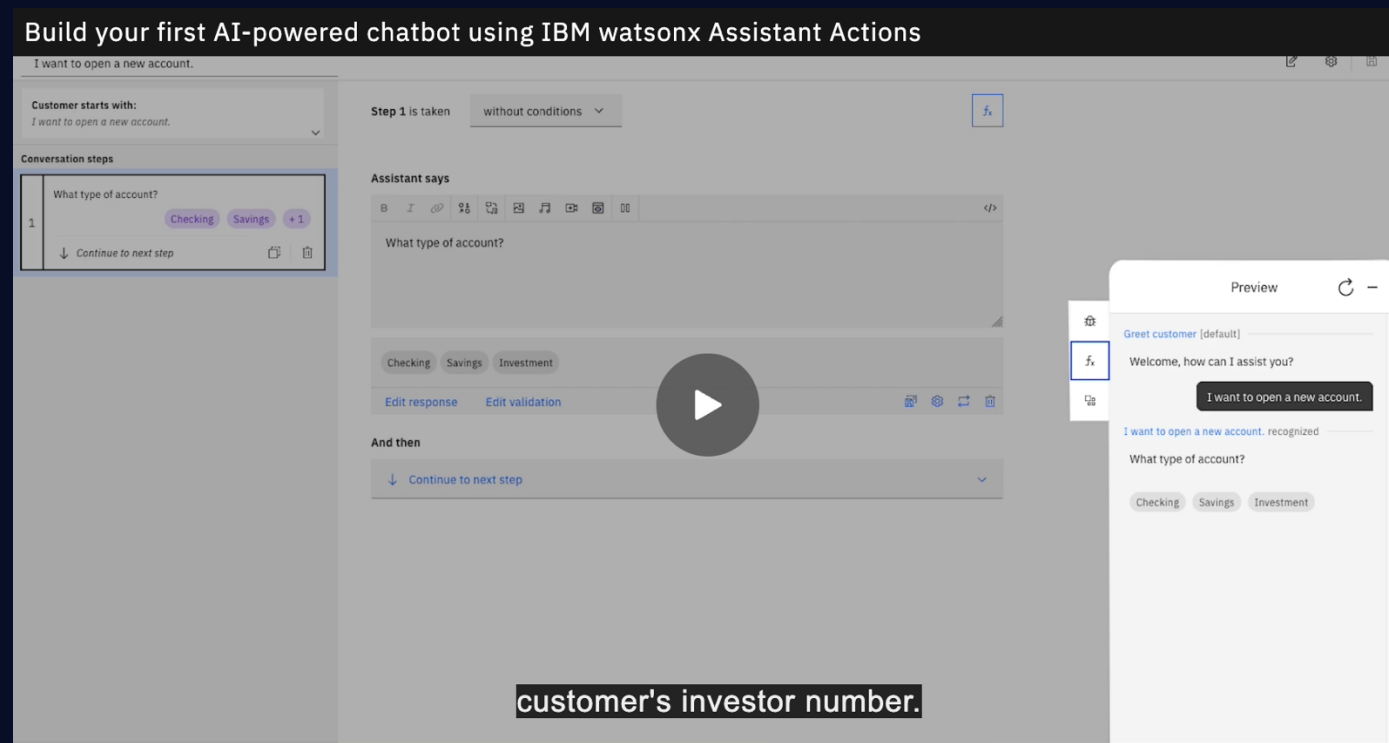
Type something...

A nam 'Next'

2/46

Integrate Watsonx Assistant with AI agent

Create Watsonx Assistant Chatbot Service

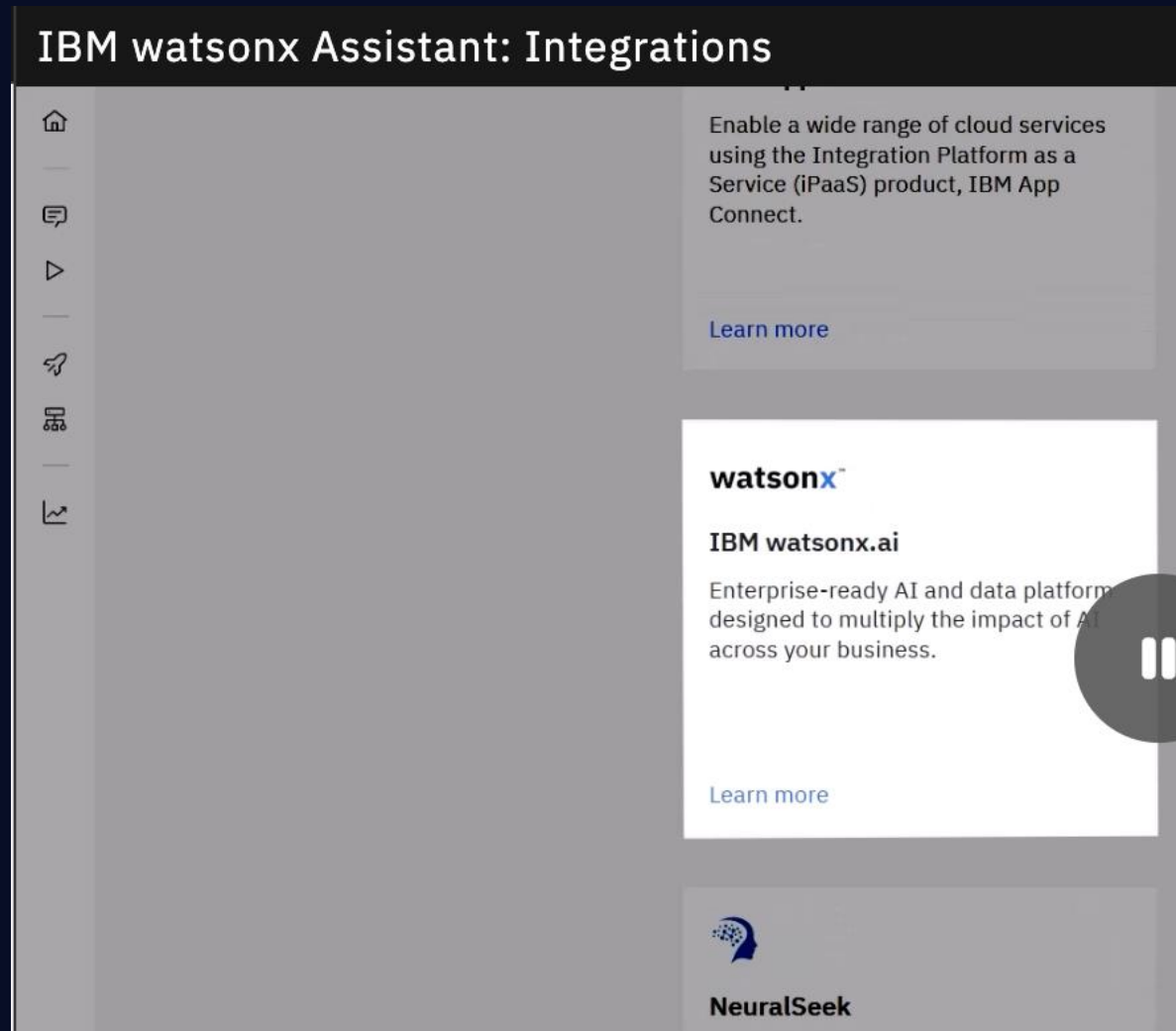


Steps Involved

- Log in to the IBM Cloud dashboard and navigate to the Watson Assistant service.
- Click on "Create a new assistant" and follow the prompts to create a new chatbot.
- Configure the chatbot to use the Watson Assistant API.

Ref: <https://www.ibm.com/products/watson-assistant/demos/lendyr/demo.html?page=build-experience-tour&ion=build-experience&panel=build-experience-building-conversation>

Integrate Watsonx Assistant Chatbot with AI Agent



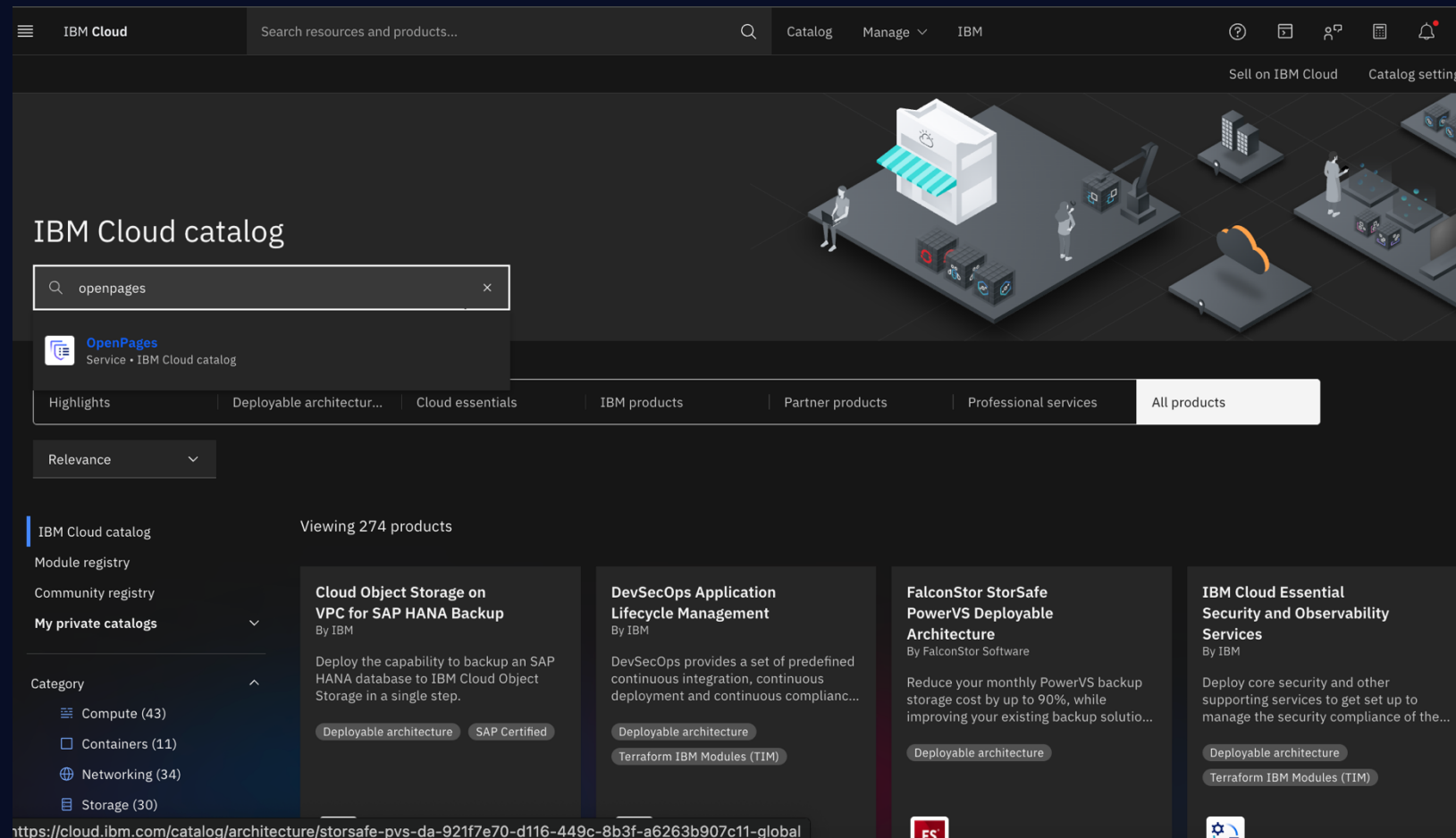
Steps Involved

Configure the Watson Assistant chatbot to use the AI agent

- In the Watson Assistant dashboard, navigate to the "Settings" tab.
- Click on "Integrations" and select "Watson Orchestrate" as the integration type.
- Enter the Watson Orchestrate API credentials and configure the integration settings.
- Configure the chatbot to use the AI agent's intents and entities.
- In the Watson Orchestrate dashboard, navigate to the "Integrations" tab.
- Click on "Add a new integration" and select "Watson Assistant" as the integration type.
- Enter the Watson Assistant API credentials and configure the integration settings.
- Map the AI agent's intents and entities to the Watson Assistant chatbot's intents and entities.

Integrating Watsonx Assistant with IBM OpenPages

Search for OpenPages in IBM Cloud Catalog

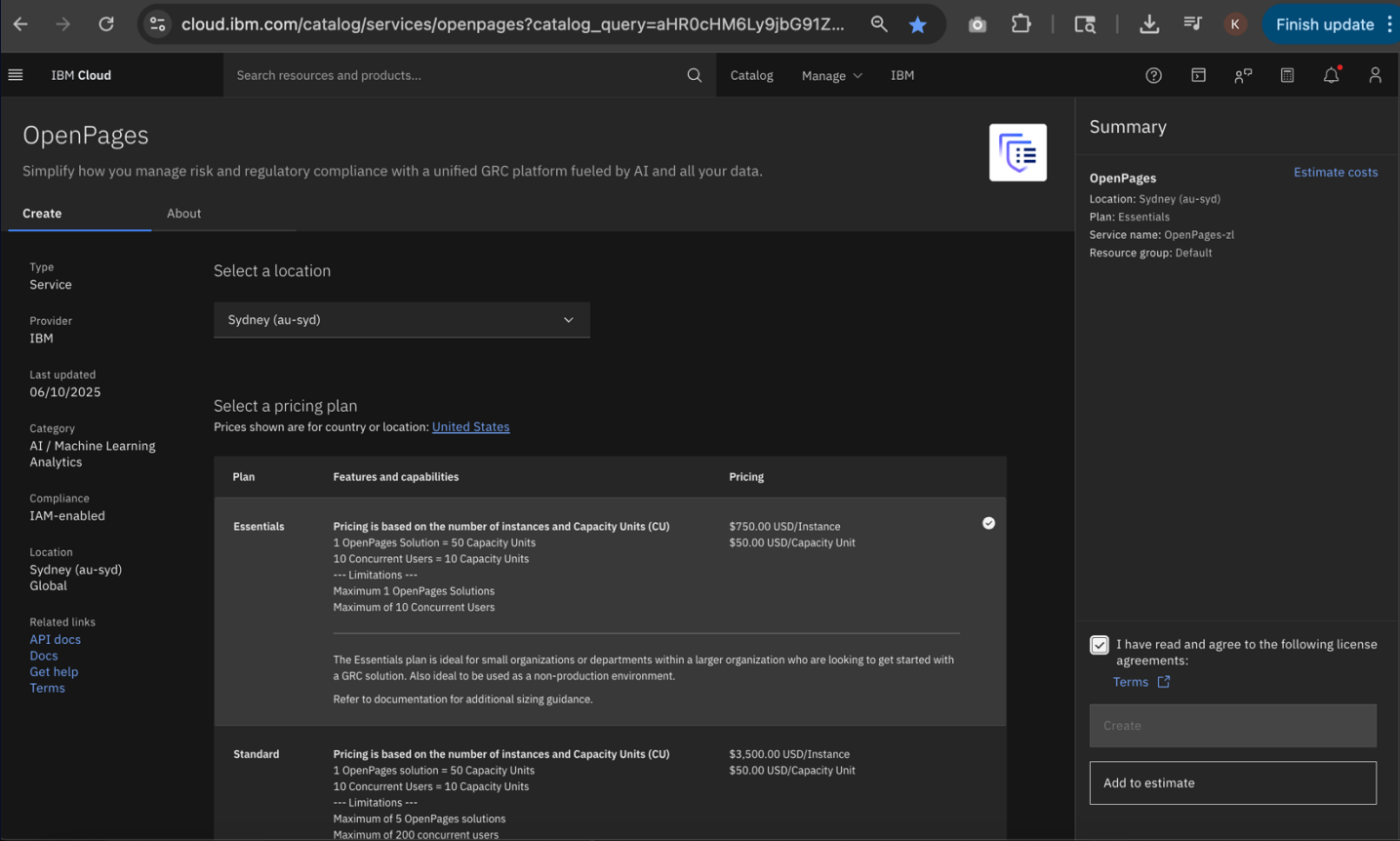


<https://cloud.ibm.com/docs/openpages?topic=openpages-gettingstartedtutorial>

<https://www.ibm.com/docs/en/openpages/9.1.x?topic=guide-openpages>

<https://cloud.ibm.com/apidocs/openpages>

Create OpenPages Service in IBM Cloud

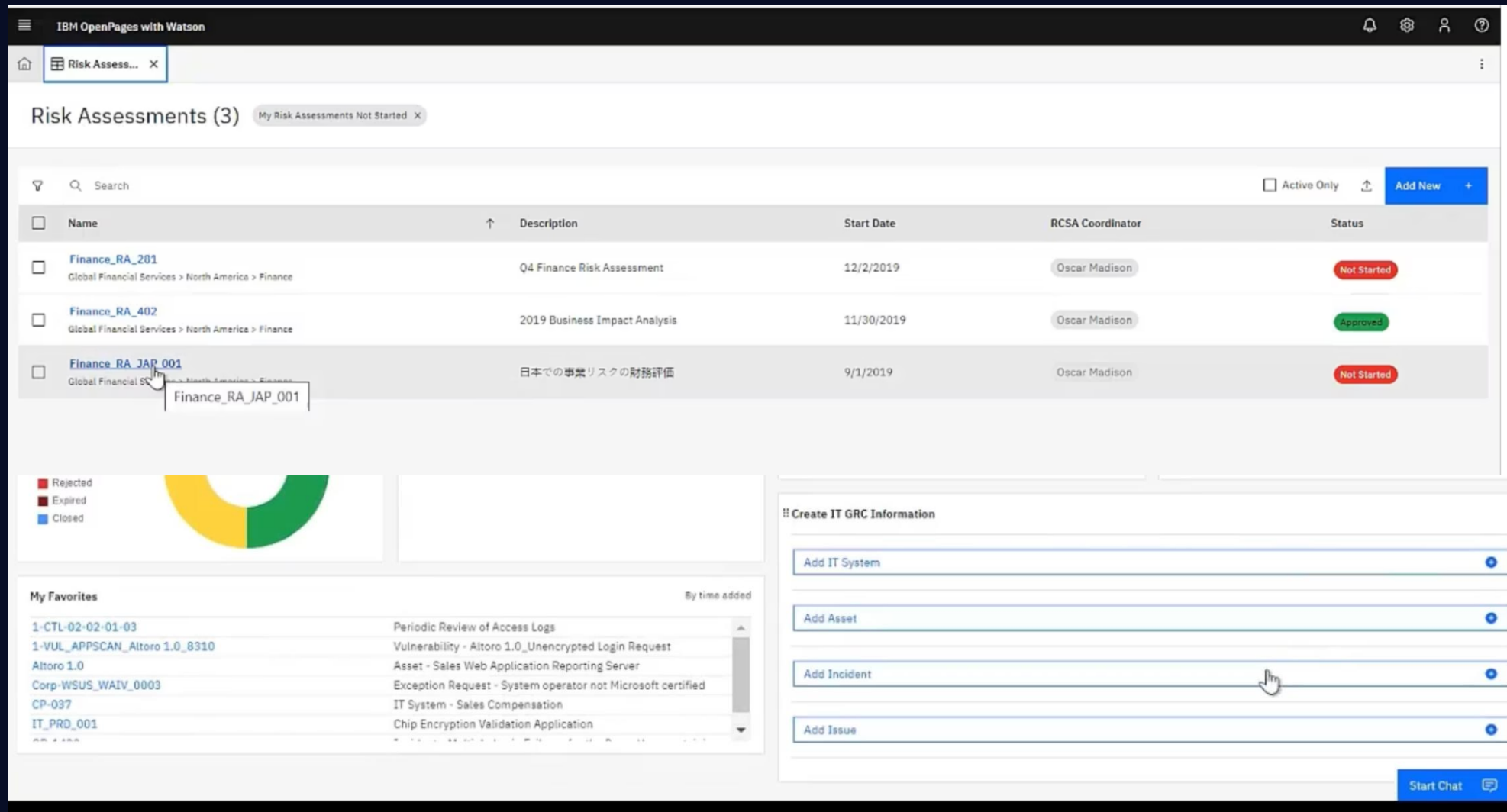


- In the IBM OpenPages dashboard, navigate to the "Settings" tab.
- Click on "Integrations" and select "Watson Assistant" as the integration type.
- Enter the Watson Assistant API credentials and configure the integration settings.
- Configure the OpenPages application to use the Watson Assistant chatbot.

<https://www.ibm.com/docs/en/openpages/9.1.x?topic=guide-openpages>

<https://cloud.ibm.com/apidocs/openpages>

Compliance & Risk Assessment Tasks in OpenPages



The screenshot displays the IBM OpenPages with Watson interface for Risk Assessments. The top navigation bar includes the IBM logo, the text "IBM OpenPages with Watson", and icons for notifications, settings, user profile, and help. Below the navigation bar, a tab labeled "Risk Assess..." is active. The main content area is titled "Risk Assessments (3)" and includes a sub-tab "My Risk Assessments Not Started". A search bar and a filter for "Active Only" are present. A table lists three risk assessments, with the third one, "Finance_RA_JAP_001", highlighted. Below the table, a donut chart shows the status distribution: Rejected (red), Expired (dark red), and Closed (blue). On the left, a "My Favorites" section lists various items. On the right, a "Create IT GRC Information" section provides buttons to add IT System, Asset, Incident, and Issue. A "Start Chat" button is located at the bottom right.

Name	Description	Start Date	RCSA Coordinator	Status
Finance_RA_281 Global Financial Services > North America > Finance	Q4 Finance Risk Assessment	12/2/2019	Oscar Madison	Not Started
Finance_RA_402 Global Financial Services > North America > Finance	2019 Business Impact Analysis	11/30/2019	Oscar Madison	Approved
Finance_RA_JAP_001 Global Financial Services > North America > Finance	日本での事業リスクの財務評価	9/1/2019	Oscar Madison	Not Started

My Favorites

Item ID	Description
1-CTL-02-02-01-03	Periodic Review of Access Logs
1-VUL_APPSCAN_Altoro 1.0_8310	Vulnerability - Altoro 1.0_Unencrypted Login Request
Altoro 1.0	Asset - Sales Web Application Reporting Server
Corp-WSUS_WAIV_0003	Exception Request - System operator not Microsoft certified
CP-037	IT System - Sales Compensation
IT_PRD_001	Chip Encryption Validation Application

Create IT GRC Information

- Add IT System
- Add Asset
- Add Incident
- Add Issue

Start Chat

Watsonx Assistant Chatbot in OpenPages

The screenshot shows the IBM OpenPages dashboard interface. The browser tabs include 'IBM OpenPages with Watson', 'ibmwin16 - IBM - DTE Skytap', and 'Login - IBM OpenPages with Watson'. The address bar shows the URL 'services-uscentral.skytap.com:16508/openpages/app/jspview/react/grc/dashboard/Home'. The dashboard features several widgets: 'IT Compliance Libraries' with counts for Industry Standards (165), Control Standards (382), IT Policies (2), and Baseline Library (71); 'IT Application and Asset Management' with counts for Managed Applications (4), Assets (313), and Monthly Vulnerabilities (35); 'Issues by Priority' showing a bar chart with 133 issues; 'IT Risk Assessment Activities' with counts for Service Level KRIs (32), My Risks (17), and My IT Controls (8); and 'Reporting & Analytics' with a search bar. On the right, the 'Watson Assistant' chatbot is open, displaying a conversation with GRACE, a Governance, Risk, and Compliance Expert. The chatbot has responded to the user's question about the difference between inherent and residual risk, providing definitions and a list of sources for creating issues.

Home

Dashboard | My Tasks (471) | Subscription Tasks (20) | Oversight Tasks (6)

IT Compliance Libraries

Industry Standards / Regulations by Type: 165

Control Standards by Risk Area: 382

IT Policies (published): 2

Baseline Library: 71

IT Application and Asset Management

Managed Applications by Criticality: 4

Assets by Risk Ranking: 313

Monthly Vulnerabilities by Risk Ranking: 35

Issues by Priority

133

Medium | Not Determined

IT Risk Assessment Activities

Service Level KRIs by Breach Status: 32

My Risks by Risk Rating: 17

My IT Controls by Effectiveness: 8

Reporting & Analytics

Search

Watson Assistant

Hi, my name is GRACE, (Governance, Risk And Compliance Expert). How can I help you today?

what is the difference between inherent and residual risk?

Inherent Risk is typically defined as the level of risk in place in order to achieve an entity's objectives and before actions are taken to alter the risk's impact or likelihood.

Residual Risk is the remaining level of risk following the development and implementation of the entity's response.

I need to create an issue

An issue is a control weakness that highlights a deficiency in the mitigation of a key risk. Issues need to be created if controls have failed through either

- Control testing
- An incident
- Audit finding

You can create an Issue from [here](#) or see your existing issues [here](#)

Further documentation can be found [here](#)

Implementation code

Code

Example code snippet on how to integrate the AI agent deployed within Watsonx Orchestrate to Watsonx Assistant chatbot within IBM OpenPages using Python

```
import requests

# Watson Orchestrate API credentials
orchestrate_url = "https://api.us-south.orchestrate.watson.cloud.ibm.com"
orchestrate_username = "your_username"
orchestrate_password = "your_password"

# Watson Assistant API credentials
assistant_url = "https://api.us-south.assistant.watson.cloud.ibm.com"
assistant_username = "your_username"
assistant_password = "your_password"

# IBM OpenPages API credentials
openpages_url = "https://your_openpages_instance.com"
openpages_username = "your_username"
openpages_password = "your_password"

# Define the AI agent's intents and entities
intents = ["assign_compliance_issue", "approve_compliance_issue"]
entities = ["compliance_issue_id", "assignee"]

# Define the Watson Assistant chatbot's intents and entities
assistant_intents = ["assign_compliance_issue", "approve_compliance_issue"]
assistant_entities = ["compliance_issue_id", "assignee"]

# Integrate the AI agent with Watson Assistant chatbot
def integrate_ai_agent_with_assistant():
    # Authenticate with Watson Orchestrate
    orchestrate_auth = requests.auth.HTTPBasicAuth(orchestrate_username, orchestrate_password)

    # Authenticate with Watson Assistant
    assistant_auth = requests.auth.HTTPBasicAuth(assistant_username, assistant_password)

    # Map the AI agent's intents and entities to the Watson Assistant chatbot's intents and entities
    for intent in intents:
        for entity in entities:
            # Create a new intent in Watson Assistant
            intent_url = f"{assistant_url}/v1/workspaces/{assistant_workspace_id}/intents"
            intent_data = {
                "intent": intent,
                "entities": [entity]
            }
            response = requests.post(intent_url, auth=assistant_auth, json=intent_data)

            # Create a new entity in Watson Assistant
            entity_url = f"{assistant_url}/v1/workspaces/{assistant_workspace_id}/entities"
            entity_data = {
                "entity": entity,
                "values": [entity]
            }
            response = requests.post(entity_url, auth=assistant_auth, json=entity_data)

# Integrate the Watson Assistant chatbot with IBM OpenPages
def integrate_assistant_with_openpages():
    # Authenticate with IBM OpenPages
    openpages_auth = requests.auth.HTTPBasicAuth(openpages_username, openpages_password)

    # Configure the OpenPages application to use the Watson Assistant chatbot
    openpages_url = f"{openpages_url}/api/v1/chatbot"
    openpages_data = {
        "chatbot_id": assistant_workspace_id,
        "chatbot_url": assistant_url
    }
    response = requests.post(openpages_url, auth=openpages_auth, json=openpages_data)

integrate_assistant_with_openpages()

integrate_ai_agent_with_assistant()
```

Code

Example code snippet on how you can implement the capabilities for the AI agent to create a new case in OpenPages, assign it to an investigator, and provide any necessary documentation or evidence using Python

```
1 import watson_orchestrate
2
3 # Create a new agent
4 agent = watson_orchestrate.Agent(
5     name="Compliance Issue Agent",
6     description="This agent creates a new case, assigns it to an investigator, and provides documentation or evidence"
7 )
8
9 # Define the agent's steps
10 agent.steps = [
11     watson_orchestrate.Step(
12         name="Create Case",
13         description="Create a new case in OpenPages",
14         action=create_case
15     ),
16     watson_orchestrate.Step(
17         name="Assign Case",
18         description="Assign the case to an investigator",
19         action=assign_case
20     ),
21     watson_orchestrate.Step(
22         name="Provide Documentation",
23         description="Provide documentation or evidence for the case",
24         action=provide_documentation
25     )
26 ]
27
28 # Save the agent
29 agent.save()
```

```
import watson_orchestrate
# Trigger the agent to run
agent = watson_orchestrate.Agent.get("Compliance Issue Agent")
agent.trigger()
```

```
import requests
import json

# Define the OpenPages API endpoint and credentials
openpages_url = "https://your-openpages-instance.com/api/v1"
openpages_username = "your-username"
openpages_password = "your-password"

# Define the case details
case_title = "New Compliance Issue"
case_description = "This is a new compliance issue"
case_investigator = "investigator@example.com"

# Create a new case in OpenPages
def create_case(case_title, case_description, case_investigator):
    # Set the API endpoint and headers
    endpoint = f"{openpages_url}/cases"
    headers = {
        "Content-Type": "application/json",
        "Authorization": f"Basic {openpages_username}:{openpages_password}"
    }

    # Set the case data
    case_data = {
        "title": case_title,
        "description": case_description,
        "investigator": case_investigator
    }

    # Make the API call to create the case
    response = requests.post(endpoint, headers=headers, json=case_data)

    # Check if the case was created successfully
    if response.status_code == 201:
        print("Case created successfully")
        return response.json()["id"]
    else:
        print("Error creating case:", response.text)
        return None

# Assign the case to an investigator
def assign_case(case_id, investigator):
    # Set the API endpoint and headers
    endpoint = f"{openpages_url}/cases/{case_id}/assign"
    headers = {
        "Content-Type": "application/json",
        "Authorization": f"Basic {openpages_username}:{openpages_password}"
    }

    # Set the assignment data
    assignment_data = {
        "investigator": investigator
    }

    # Make the API call to assign the case
    response = requests.put(endpoint, headers=headers, json=assignment_data)

    # Check if the case was assigned successfully
    if response.status_code == 200:
        print("Case assigned successfully")
    else:
        print("Error assigning case:", response.text)

# Provide documentation or evidence for the case
def provide_documentation(case_id, documentation):
    # Set the API endpoint and headers
    endpoint = f"{openpages_url}/cases/{case_id}/documentation"
    headers = {
        "Content-Type": "application/json",
        "Authorization": f"Basic {openpages_username}:{openpages_password}"
    }

    # Set the documentation data
    documentation_data = {
        "documentation": documentation
    }

    # Make the API call to provide the documentation
    response = requests.post(endpoint, headers=headers, json=documentation_data)

    # Check if the documentation was provided successfully
    if response.status_code == 201:
        print("Documentation provided successfully")
    else:
        print("Error providing documentation:", response.text)

# Create a new case, assign it to an investigator, and provide documentation
case_id = create_case(case_title, case_description, case_investigator)
if case_id:
    assign_case(case_id, case_investigator)
    provide_documentation(case_id, "This is some documentation for the case")
```

End User Training & Support

IBM OpenPages with Watson Training Material



IBM OpenPages with Watson - GRC Essentials - Sales Foundation

Issued by [IBM](#)

This badge earner is a GRC sales professional who has developed fundamental competency in the offering and sales essentials for IBM OpenPages with Watson. They understand the OpenPages platform and solutions that make up the offering in order to make a sales pitch to potential clients, along with understanding the various use cases the offering supports.

- <https://www.credly.com/badges/ad433cae-d940-4e82-9812-cd3b7cd619d6>
- <https://learn.ibm.com/mod/video/view.php?id=165039>
- IBM AI ML Ops for IT Operations <https://www.ibm.com/solutions/ai-assistants#ai-assistants>