

KYC Agent: Automate verification, risk profiling, and monitoring with Agentic AI

Learning Objective

Time Required : 90 Mins

You will acquire practical, hands-on experience building and deploying intelligent KYC agents using Watsonx Orchestrate Developer Edition and the Agent Development Kit (ADK). Specifically, you will be able to:

- ✓ **Design and deploy KYC agents** on your local machine using the Watsonx Orchestrate Developer Edition bundled with the ADK, enabling rapid iteration in an isolated dev environment before moving to production.
- ✓ **Build and integrate custom tools** using the ADK framework that fetch, validate, and process customer identity data in real time from both internal systems and external data sources to automate verification, profiling, and continuous monitoring workflows.
- ✓ **Build conversational experiences** using the built-in Watsonx Assistant within Watsonx Orchestrate to enable natural language interactions for querying KYC verification results, reviewing customer risk profiles, and performing real-time monitoring of customer activity and compliance indicators.
- ✓ **Use modular, low-code orchestration with ADK** to stitch together workflows that combine tools, conditional logic, LLMs, and prompts to deliver seamless KYC automation.
- ✓ **Apply these skills to real-world KYC use cases** by streamlining onboarding, ensuring ongoing risk monitoring, and improving compliance efficiency through end-to-end intelligent automation.

The Problem

FinTrust Bank, a mid-sized financial services provider, is experiencing growing inefficiencies in its customer onboarding and compliance processes. The current manual approach to KYC verification and customer monitoring is not scalable, leading to delays and increased operational risks. There is no unified mechanism for continuous customer risk profiling and monitoring, resulting in delayed compliance checks, missed red flags, and increased operational costs. With rising regulatory pressure and an expanding customer base, the bank requires an automated solution to streamline verification, enable real-time risk insights, and maintain audit-ready compliance.

Key Challenges:

- KYC verification is manual and time-consuming, slowing down customer onboarding timelines.
- Inconsistent processes across regions result in compliance gaps and operational inefficiencies.
- No centralized system for continuous customer risk profiling and monitoring
- Delays in detecting high-risk profiles or anomalies due to lack of real-time insights.
- High operational effort and cost due to dependence on manual reviews.
- Difficulty in maintaining audit trails and ensuring readiness for regulatory checks.

An intelligent, autonomous KYC Agent equipped with purpose-built tools such as identity verification, document extraction, customer profiling, and risk scoring can transform this process. By automating repetitive compliance tasks and enabling real-time access to customer data, the agent can provide faster onboarding, reduce operational effort, and ensure consistent adherence to regulatory standards. This approach also enhances decision-making by delivering proactive risk insights and eliminating the delays caused by manual review cycles.

Objective

FinTrust Bank plans to implement an AI-powered KYC Agent to support compliance and risk teams in continuously monitoring and managing customer risk profiles. The goal is to build an agentic AI solution that intelligently orchestrates KYC verification, transaction monitoring, and risk scoring systems to deliver proactive, real-time compliance insights. The KYC Agent will assist teams in executing the following tasks:

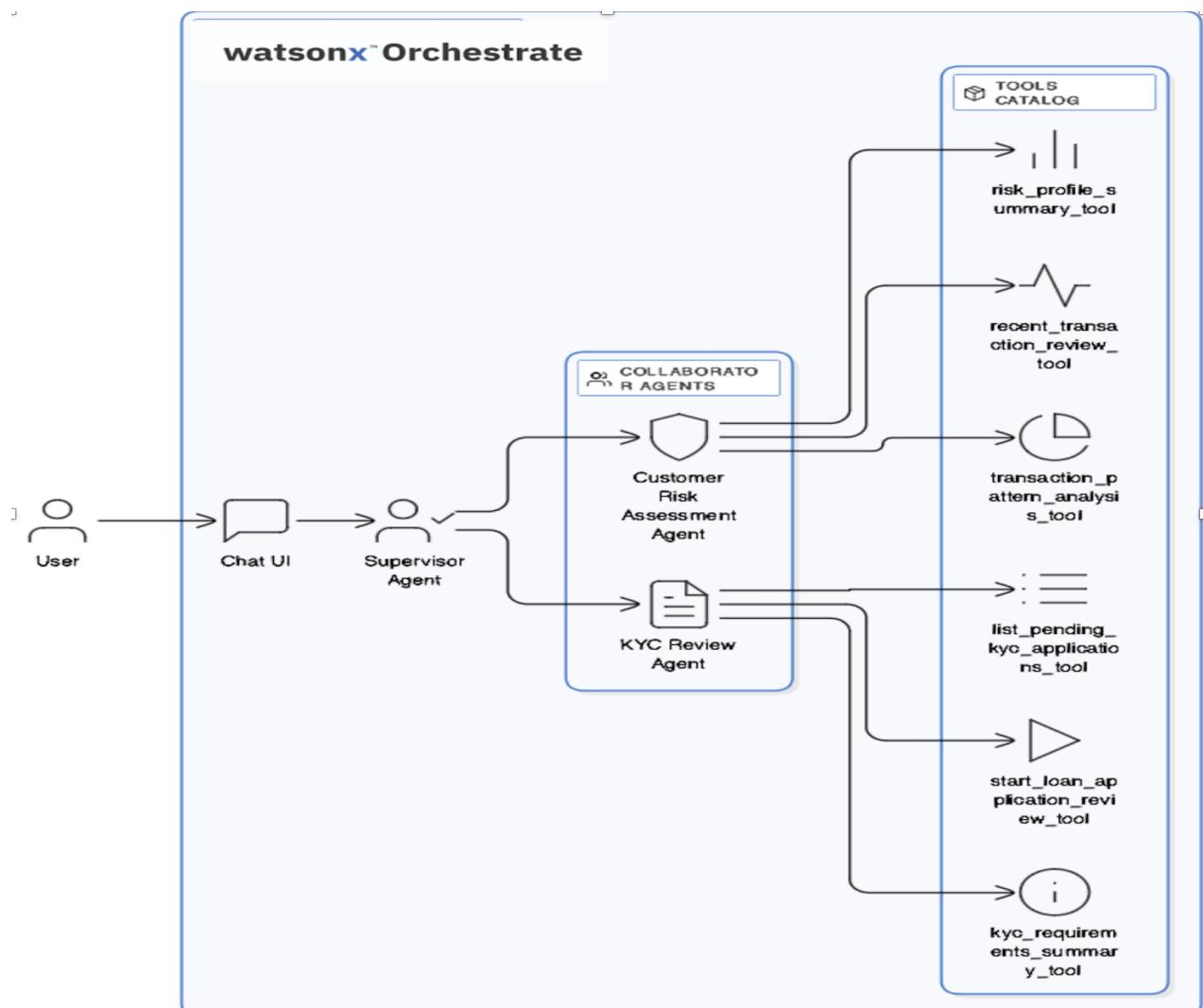
- Retrieve and validate customer identity data from internal and external sources
- Automate KYC checks including document verification, watchlist screening, and name matching.
- Monitor customer behaviour and transactions to identify anomalies or compliance risks.
- Assess and update customer risk profiles in real time based on evolving data.
- Provide contextual summaries and remediation suggestions to support faster decision-making

By automating these processes, the bank aims to reduce manual workload, improve compliance accuracy, and ensure consistent, audit-ready monitoring across the customer lifecycle.

Business Value

- **Accelerates customer onboarding** by removing manual steps and reducing delays.
- **Improves operational efficiency** by minimizing repetitive tasks and manual follow-ups.
- **Ensures up-to-date risk profiles**, helping teams stay audit-ready and compliant.
- **Supports real-time risk detection**, enabling early identification of red flags in customer behaviour.

Architecture



Pre-requisites

- Check with your instructor to make sure **all systems** are up and running before you continue.
- Validate that you have access to the right techzone environment for this lab.
- Validate that you have access to a credentials file that your instructor will share with you before starting the labs.

Accessing Your Watsonx Orchestrate Instance

- A TechZone environment has been reserved for you, and you will receive the access details via email.
- To access your Watsonx Orchestrate instance, **please check with your instructor** for the access link and the necessary steps.

Getting Started with the Agent Development Kit (ADK)

The **Agent Development Kit (ADK)** is a set of tools that helps you build, test, and manage AI agents for Watsonx Orchestrate. It gives developers full control using a simple CLI and a lightweight framework.

With ADK, you can:

- ✓ Define agents using YAML or JSON
- ✓ Create custom tools in Python
- ✓ Run and manage agents with just a few commands

In this bootcamp, you'll:

- ✓ Set up your local environment
- ✓ Install the ADK
- ✓ Deploy your first agent to a Watsonx Orchestrate instance

Before You Begin

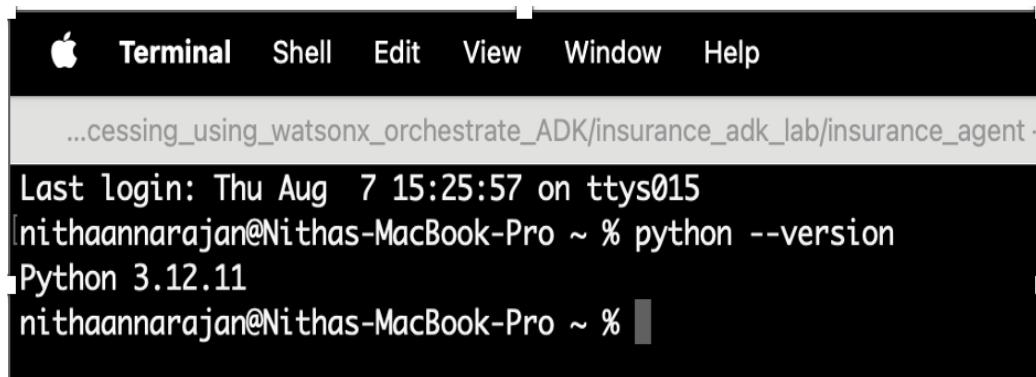
Before proceeding, ensure the following requirements are met:

1. Python

- ✓ **Version:** Python 3.11 or later

- ✓ **Check version:**

```
python3 --version
```



A screenshot of a macOS Terminal window. The window title is "Terminal". The menu bar includes "Terminal", "Shell", "Edit", "View", "Window", and "Help". The main pane shows the following text:
...cessing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent -
Last login: Thu Aug 7 15:25:57 on ttys015
[nithaannarajan@Nithas-MacBook-Pro ~ % python --version
Python 3.12.11
nithaannarajan@Nithas-MacBook-Pro ~ %

2. A Docker Engine

Ensure that you have a docker engine installed capable of running docker compose. The watsonx Orchestrate team recommend either Rancher or Colima.

Please make sure your instance of Rancher or Colima is configured with the following settings:

- [Rancher](#)
- [Colima](#)

Colima settings

M Series Mac

```
colima start --cpu-type host --arch host --vm-type=vz --mount-type  
virtiofs -c 8 -m 16
```

Intel Mac

```
colima start --cpu-type host --arch host --vm-type=vz --vz-rosetta -  
-mount-type virtiofs -c 8 -m 16
```

From ADK Installation to Agent Testing: Step-by-Step Setup Guide

You can install the ADK either directly on your system or inside a virtual environment. Installing it system-wide makes it available to all your Python projects, but using a virtual environment gives you better control over your setup and makes it easier to manage and share your agents and tools.

In this bootcamp, we will install the ADK inside a virtual environment to keep things clean, organized, and easy to manage.

Follow the steps below to get started with the virtual environment setup.

1. Installing ADK

Open your **terminal** to run the commands below.

a. Create your virtual environment

```
python -m venv adkenv
```

b. Activate your virtual environment

```
source ./adkenv/bin/activate
```

c. Install the ADK with the following command:

```
pip install ibm-watsonx-orchestrate
```

If you face issues after installing the ADK, you may need to update it to the latest version. Once installed or updated, you can check whether the installation was successful and explore the available commands using the CLI.

```
pip install --upgrade ibm-watsonx-orchestrate
orchestrate --version
orchestrate --help
```

```

Terminal Shell Edit View Window Help
...cessing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent --zsh ...
Last login: Thu Aug 7 15:25:57 on ttys015
nithaannarajan@Nithas-MacBook-Pro ~ % python --version
Python 3.12.11
nithaannarajan@Nithas-MacBook-Pro ~ % python -m venv adkenv
nithaannarajan@Nithas-MacBook-Pro ~ % source ./adkenv/bin/activate
((adkenv)) nithaannarajan@Nithas-MacBook-Pro ~ % pip install ibm-watsonx-orchestrate
Collecting ibm-watsonx-orchestrate
  Downloading ibm_watsonx_orchestrate-1.8.1-py3-none-any.whl.metadata (1.4 kB)
Collecting certifi>=2024.8.30 (from ibm-watsonx-orchestrate)
  Downloading certifi-2025.8.3-py3-none-any.whl.metadata (2.4 kB)
Collecting click<8.2.0,>=8.0.0 (from ibm-watsonx-orchestrate)
  Using cached click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting docstring_patcher-1.0.>=0.16 (from ibm-watsonx-orchestrate)

((adkenv)) nithaannarajan@Nithas-MacBook-Pro ~ % orchestrate --version
ADK Version: 1.4.2
Developer Edition Image Tags (if not overridden in env file)
  SERVER_TAG: 22-05-2025
  WORKER_TAG: 22-05-2025
  AI_GATEWAY_TAG: 20-05-2025
  DBTAG: 21-05-2025
  CM_TAG: 13-05-2025
  TRM_TAG: 23-05-2025
  TR_TAG: 23-05-2025
  BUILDER_TAG: 21-05-2025
  FLOW_RUNTIME_TAG: 13-05-2025
((adkenv)) nithaannarajan@Nithas-MacBook-Pro ~ %

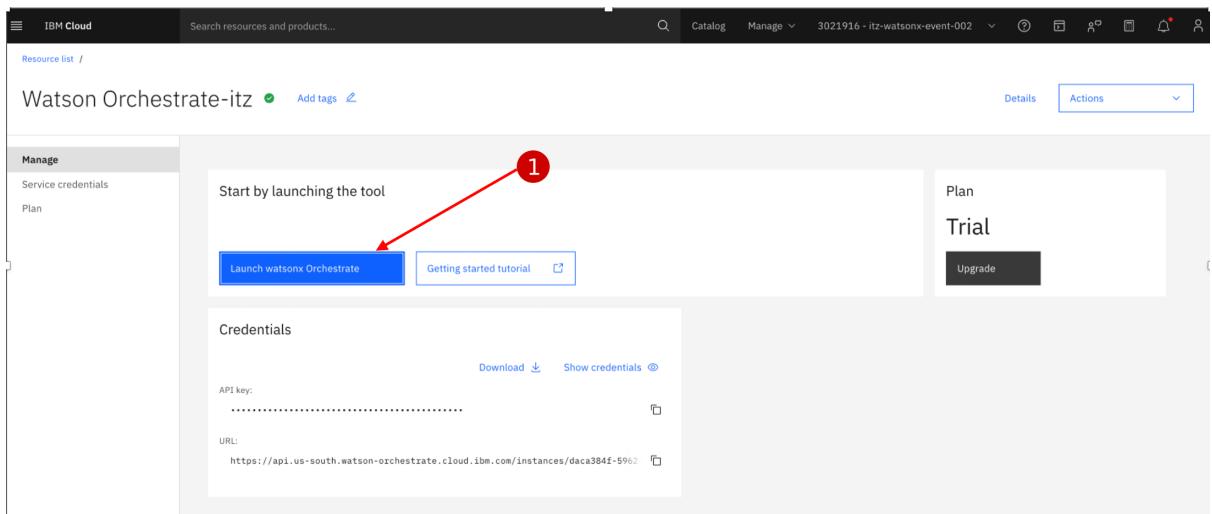
```

2. Configure your environment in the ADK

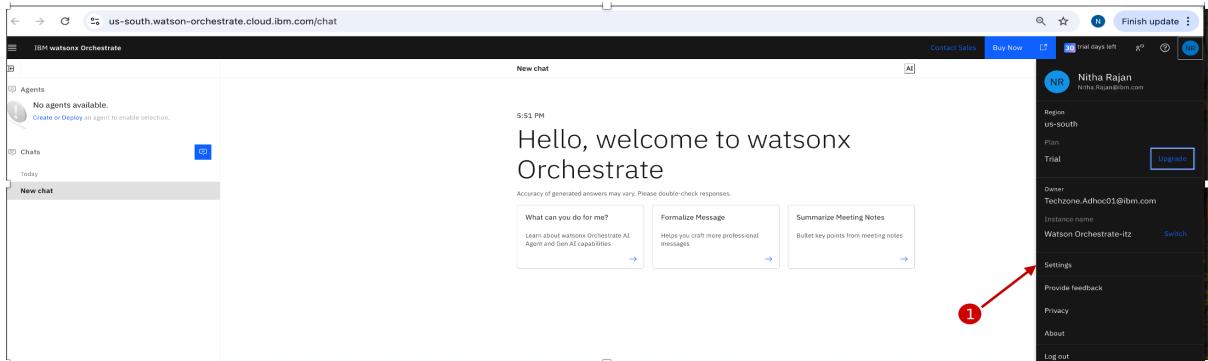
Before you can start creating agents, you need to configure your **watsonx Orchestrate environment** in the ADK.

To do this, you will need **credentials** specific to your environment (such as IBM Cloud, AWS, or on-premises).

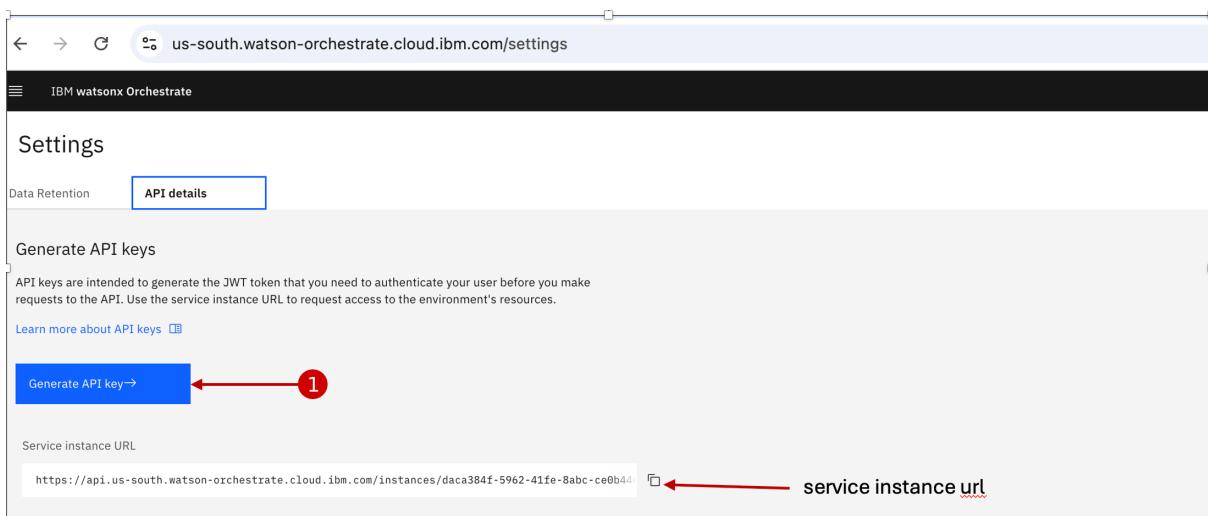
If you don't have these details, **please check with your bootcamp instructor** before proceeding.(Refer the screen shots below)



When you click **Launch watsonx Orchestrate**, it will take you to the **Orchestrate landing page**.



On this page, click on **Settings** as shown in the screenshot provided.



On the **Settings** page, note down the **Service Instance URL**. You will need this when setting up your environment in the ADK.

Click on **Generate API Key** to create the key required for authentication. Make sure to copy and save the API key somewhere safe. It will not be shown again.

Once you have the credentials, you can add the environment using this command:

```
orchestrate env add -n <environment-name> -u https://my-service-instance-url --type ibm_iam -activate
```

What each part means:

- <environment-name>: A name you choose to refer to this environment (e.g., `ibm-cloud`, `demo-env`)
- `https://my-service-instance-url`: The URL of your Orchestrate instance
- `--type ibm_iam`: The authentication type (IBM Cloud uses `ibm_iam`)
- `--activate`: This sets the environment as the active one right after adding it

Enter the API key generated in the previous step when prompted.

```
((adkenv) nithaannarajan@Nithas-MacBook-Pro ~ % orchestrate env add -n ibm-cloud-env -u https://api.us-south.watson-orchestrate.cloud.ibm.com/instances/daca384f-5962-41fe-8abc-ce0b44e38c02 --type ibm_iam --active
[INFO] - Environment 'ibm-cloud-env' has been created
Please enter IXO API key:
[INFO] - Environment 'ibm-cloud-env' is now active
((adkenv) nithaannarajan@Nithas-MacBook-Pro ~ % )
```

3. Activate your environment

Run the following command to activate the environment you created:

```
orchestrate env activate <environment-name>
```

Replace `<environment-name>` with the name you used earlier when adding the environment.

4. Clone the Repository and Retrieve the Code

Once your virtual environment is active and your environment is configured, clone the bootcamp repository to get the required tools and agent files:

```
git clone -b usecases git@github.ibm.com:ibm-client-engineering/fsm-agentic-ai-bootcamp.git
cd customer-risk-summary
```

5. Import Tools And Agents

You can choose either of the following options to import the tools and agents:

Option A: Use Manual Import Commands

To register your custom Python tools and agents in Watsonx Orchestrate, run the following commands in sequence:

```
orchestrate tools import -k python -f tools/risk_analysis_tools.py -r requirements.txt
orchestrate tools import -k python -f tools/kyc_application_tools.py -r requirements.txt
```

Where:

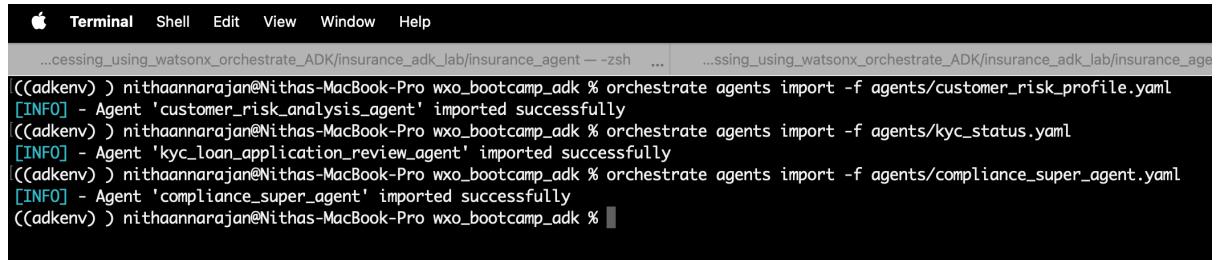
- `-k python`: Specifies it's a Python tool.
- `-f`: The path to your `.py` file that contains the `@tool` function.
- `-r`: The path to your `requirements.txt` file listing dependencies.

Below commands import your YAML agent configurations so they become available in the environment:

```
orchestrate agents import -f agents/customer_risk_profile.yaml
orchestrate agents import -f agents/kyc_status.yaml
orchestrate agents import -f agents/compliance_super_agent.yaml
```



```
...cessing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent --zsh ... ssing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent --zsh
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate tools import -k python -f tools/risk_analysis_tools.py -r requirements.txt
[INFO] - Using requirement file: "requirements.txt"
[INFO] - Tool 'recent_transaction_review_tool' imported successfully
[INFO] - Tool 'risk_profile_summary_tool' imported successfully
[INFO] - Tool 'transaction_pattern_analysis_tool' imported successfully
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate tools import -k python -f tools/kyc_application_tools.py -r requirements.txt
[INFO] - Using requirement file: "requirements.txt"
[INFO] - Tool 'high_risk_trigger_explanation_tool' imported successfully
[INFO] - Tool 'kyc_requirements_summary_tool' imported successfully
[INFO] - Tool 'list_pending_kyc_applications_tool' imported successfully
[INFO] - Tool 'start_loan_application_review_tool' imported successfully
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate agents import -f ./agents/customer_risk_analysis_agent.yaml
```



```
...cessing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent --zsh ... ssing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent --zsh
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate agents import -f agents/customer_risk_profile.yaml
[INFO] - Agent 'customer_risk_analysis_agent' imported successfully
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate agents import -f agents/kyc_status.yaml
[INFO] - Agent 'kyc_loan_application_review_agent' imported successfully
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk % orchestrate agents import -f agents/compliance_super_agent.yaml
[INFO] - Agent 'compliance_super_agent' imported successfully
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxa_bootcamp_adk %
```

Option B: Run Import Script

```
chmod +x import-all.sh
./import-all.sh
```

This script bundles all necessary CLI commands, importing your Python tools and agent YAMLs in one go.

```

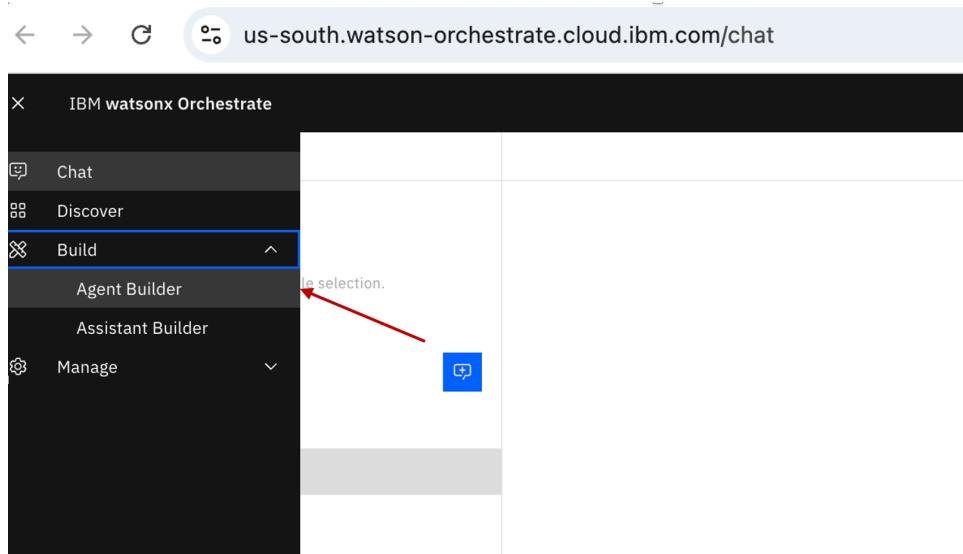
Terminal Shell Edit View Window Help
...cessing_using_watsonx_orchestrate_ADK/insurance_adk_lab/insurance_agent -- zsh ... ssing_using_watsonx
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxo_bootcamp_adk % orchestrate tools list
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxo_bootcamp_adk % chmod +x import-all.sh
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxo_bootcamp_adk % ./import-all.sh
Importing Python tools...
[INFO] - Using requirement file: "/Users/nithaannarajan/wxo_bootcamp_adk/requirements.txt"
[INFO] - Tool 'recent_transaction_review_tool' imported successfully
[INFO] - Tool 'risk_profile_summary_tool' imported successfully
[INFO] - Tool 'transaction_pattern_analysis_tool' imported successfully
[INFO] - Using requirement file: "/Users/nithaannarajan/wxo_bootcamp_adk/requirements.txt"
[INFO] - Tool 'high_risk_trigger_explanation_tool' imported successfully
[INFO] - Tool 'kyc_requirements_summary_tool' imported successfully
[INFO] - Tool 'list_pending_kyc_applications_tool' imported successfully
[INFO] - Tool 'start_loan_application_review_tool' imported successfully
Importing agent: customer_risk_profile
[INFO] - Agent 'customer_risk_analysis_agent' imported successfully
Importing agent: kyc_status
[INFO] - Agent 'kyc_loan_application_review_agent' imported successfully
Importing agent: compliance_super_agent
[INFO] - Agent 'compliance_super_agent' imported successfully
Import process complete. You can verify with:
    orchestrate tools list
    orchestrate agents list
((adkenv)) nithaannarajan@Nithas-MacBook-Pro wxo_bootcamp_adk %

```

6. How to Test Your Agent in the Watsonx Orchestrate UI

Once your agents are imported and deployed, you can test them using the built-in chat interface:

1. Log into **watsonx Orchestrate** via the web interface.
2. Click the “hamburger” menu (\equiv) in the UI navigation bar, then select **Build** → **Agent builder**.



3. From the *Agent* section, select **compliance_super_agent** to open its builder workspace.

4. **Test your agent** in the preview chat on the right side by asking the following questions and validating the responses. They should look similar to what is shown in the screenshots below:

- Show me the pending KYC applications for review
- Start review for Ryan Hogan's application
- Show the KYC requirements summary for this customer
- Why is the risk level of this customer high?
- Show the risk profile summary for customer id - cust001
- Review the recent transactions for this customer

| Applicant Name | Application ID | Type | Date Submitted | Status |
|----------------|----------------|---------------|----------------|---------|
| Ryan Hogan | KYC10234 | Loan | 2025-07-31 | Pending |
| Priya Rezada | KYC10235 | Credit Card | 2025-07-30 | Pending |
| Ahmad Saleh | KYC10236 | Overdraft | 2025-07-29 | Pending |
| Emily Chen | KYC10237 | Business Loan | 2025-07-29 | Pending |
| Liam Patel | KYC10238 | Mortgage | 2025-07-28 | Pending |

You 11:38 PM

Show the KYC requirements summary for this customer



compliance_super_agent 11:38 PM | Show Reasoning ▾

Based on Ryan Hogan's high risk level, the KYC requirements summary is as follows:

- KYC Level: High
- Required Documents:
 - Proof of identity
 - Business registration
 - Beneficial ownership declaration

You 11:39 PM

Why is the risk level of this customer high?



compliance_super_agent 11:39 PM | Show Reasoning ▾

The risk level of this customer is high due to the following triggers:

- Non-US nationality (Ireland)
- Revenue > \$1M
- Loan amount > \$1M



You 11:40 PM

Show the risk profile summary for customer id - cust001



compliance_super_agent 11:40 PM | Show Reasoning ▾

The risk profile summary for customer ID "cust001" is as follows:

- Risk Score: 82
- Risk Level: Critical
- Summary: Multiple high-value late-night transactions; urgent review required.

Please note that the risk level for this customer is critical, indicating a high risk of fraudulent activity. It is recommended to review the customer's transactions urgently to prevent any potential losses.

You 11:40 PM

Review the recent transactions for this customer



compliance_super_agent 11:40 PM | Show Reasoning ▾

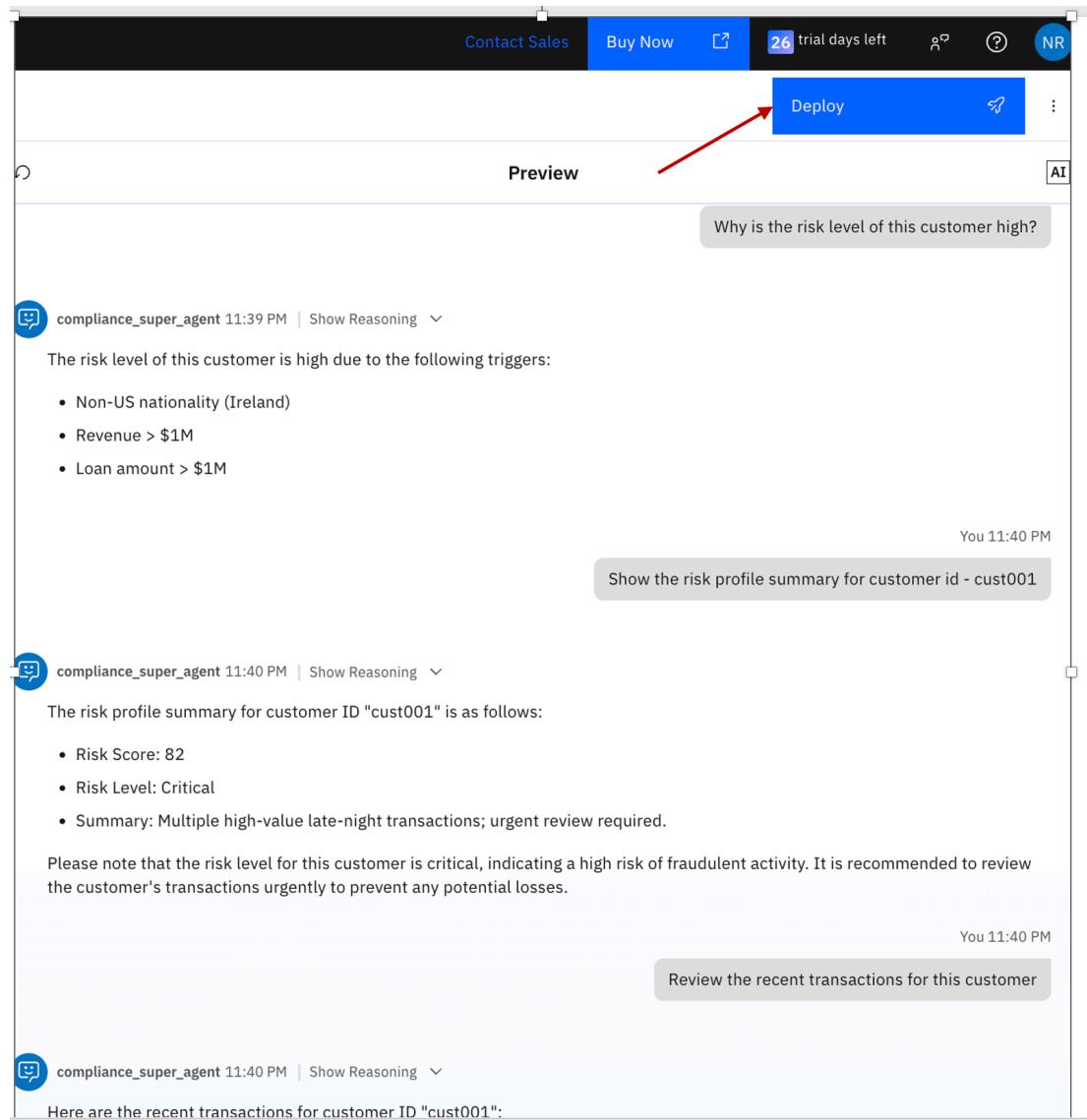
Here are the recent transactions for customer ID "cust001":

- Date: 2025-07-30, Amount: \$50,000, Description: Chip purchase, Risk Flag: True, Note: Largest single transaction this month.
- Date: 2025-07-29, Amount: \$17,000, Description: Cash advance, Risk Flag: True, Note: Unusual amount for this profile.

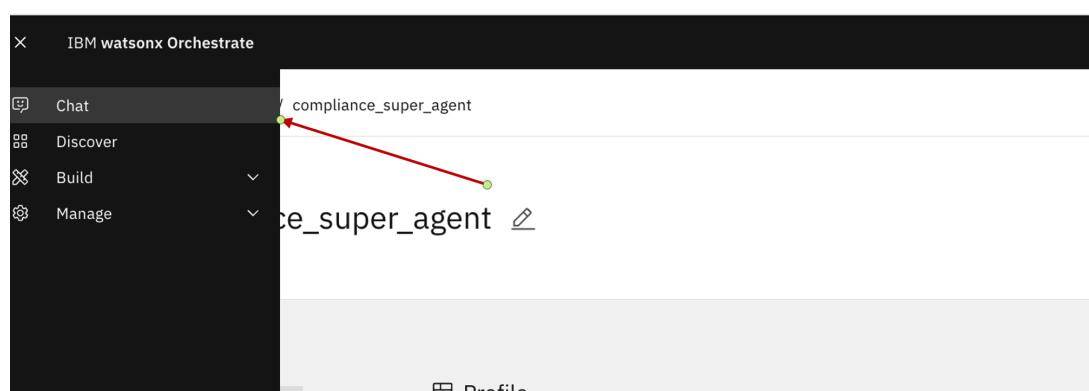
Please note that both transactions have been flagged as high-risk, indicating potential fraudulent activity. It is recommended to review these transactions further to prevent any potential losses.



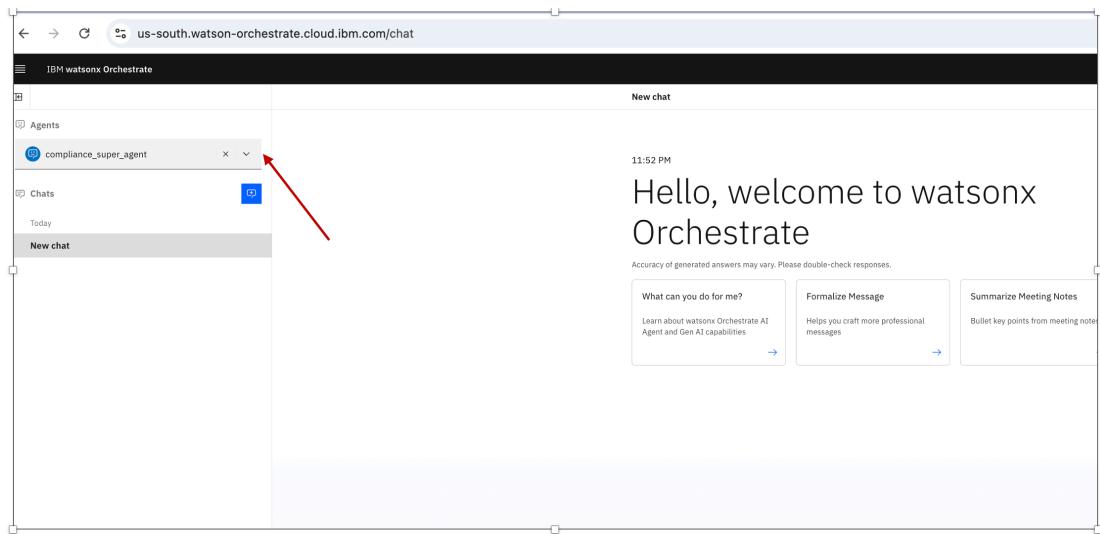
- Once you have validated the answers, click on Deploy in the top right corner to deploy your agent:



- Click on the hamburger menu in the top left corner and then click on Chat:



7. Make sure Compliance Super Agent is selected. You are now ready to test your agent:



Practice Lab: Create and Experiment with Your Own Tools

Now that you have successfully imported and tested the tools provided in the GitHub repository, try creating your own tools to extend the KYC Agent's capabilities. Follow the same **ADK tool creation process** you used earlier, using the given tools as a reference for structure, annotations, and requirements.

Here are some example tool ideas you can implement for practice:

- **risk_score_trend_tool** – Analyze how a customer's risk score changes over time.
- **suspicious_location_check_tool** – Flag transactions originating from unusual or high-risk locations.
- **fraud_alert_check_tool** – Scan for recent fraud alerts linked to a customer.
- **gaming_industry_risk_assessment_tool** – Evaluate compliance risk for customers operating in the gaming sector.
- **customer_activity_timeline_tool** – Generate a chronological view of a customer's activities.
- **multi_account_detection_tool** – Detect if a single customer is linked to multiple accounts.

Follow this checklist to build and integrate your own tool into Watsonx Orchestrate:

1. Create the Tool File

- In your tools directory, create a Python file (e.g., risk_score_trend_tool.py).
- Implement your function using the @tool decorator from ADK.
- Define inputs, outputs, and logic.

2. Add Dependencies (if any)

- If your tool needs extra Python packages, list them in your requirements.txt.

3. Test Locally

```
python your_tool_file.py
```

Ensure the function runs without errors.

4. Import Tool into Watsonx Orchestrate

```
orchestrate tool import -k python -f path/to/your_tool_file.py -r requirements.txt
```

5. Verify Tool Registration

```
orchestrate tool list
```

6. Integrate into Agent

- Edit your agent YAML to include the new tool.
- Re-import the agent:

```
orchestrate agent import -f path/to/agent.yaml
```

7. Test in Preview Chat

- Ask a query that triggers your tool.
- Validate the output matches expectations.

Wrapping Up & Next Steps

By now, you've successfully installed the ADK, configured your environment, imported tools and agents, and tested your KYC Agent in the watsonx Orchestrate UI. You've also explored how to extend the solution by building your own tools for additional compliance scenarios.

Your next steps:

- Continue experimenting with other custom tools .
- Combine your new tools into an extended KYC Agent for richer, real-world automation.

Congratulations! You've now completed the full cycle of building, deploying, and testing an intelligent KYC Agent with Watsonx Orchestrate and the ADK.

Take what you've learned here and continue experimenting. Enhance your agents, create new tools, and explore more automation possibilities in your environment.