Informatica

Build Enterprise Al Assistant with Amazon Bedrock & Informatica

Contents

Workshop Overview

Building enterprise Generative AI solutions, in production, require the solution to be anchored to the information within the organization, provide factually accurate response in the context of its business, account for and mitigate data-quality-based bias in the data, and adhere to governance and compliance requirements of the enterprise. To address these requirements, enterprises need a modern data management platform that can simplify the creation of data pipelines, facilitate data governance, and enhance application development with critical GenAI capabilities such as vector store integration. The platform should also provide trusted and curated enterprise context, metadata intelligence, and agentic RAG orchestration.

In this hands-on workshop, you will learn about Informatica's Generative AI blueprint for Amazon Bedrock. In Lab 1, using Informatica's no-code/low-code AI Agent Framework and its pre-built jumpstart recipes build an AI assistant to bring

- Bring additional context about the nuance behind the data (for e.g. data quality, data lineage, business glossary, field / column names for custom data objects etc.) to improves the overall accuracy and account for data quality-based bias
- Integrate trusted, high-quality data from Informatica Master Data Management & Business 360 solutions to enhance the accuracy and reliability of the response in the context of the business

As part of Lab 2, you will also build an Amazon Bedrock Agent and invoke Informatica's no-code/low-code AI Agent, that you built in Lab 1, to orchestrate a multi-agent workflow.

Lab 1 Building AI Assistant Using Informatica's no-code / low-code AI Agent Framework

Solution Overview

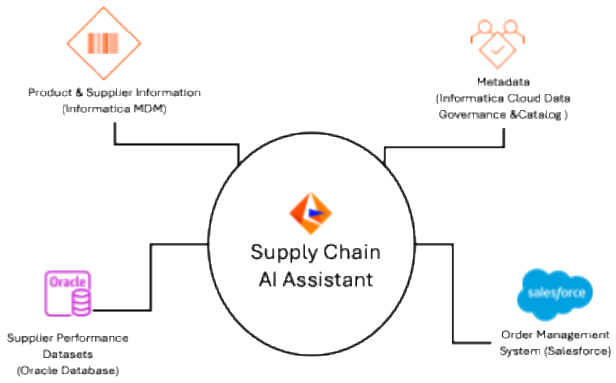


Figure 1: Solution Overview

In this you will build a simple Supply Chain AI assistant for a business that sells **build-to-order**' computer hardware for enterprise customers. Supply Chain Analyst will use the AI assistant to get information (e.g. regards to order details, product details, supplier details, and order lead time for different components) from various enterprise systems.

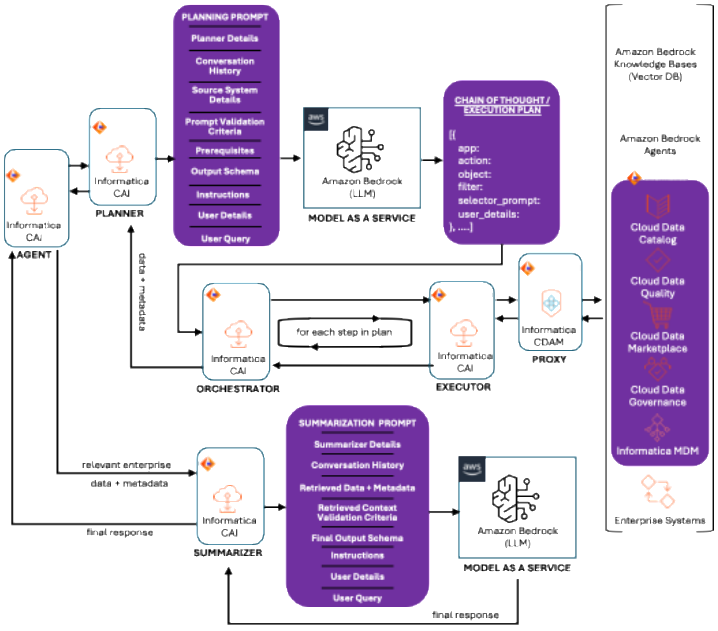
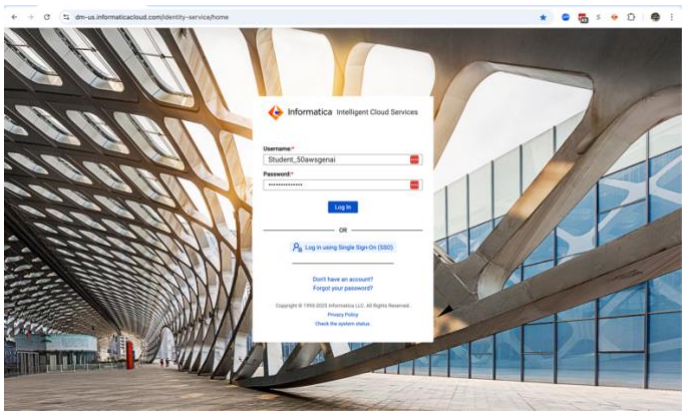


Figure 2: Informatica AI Agent Architecture

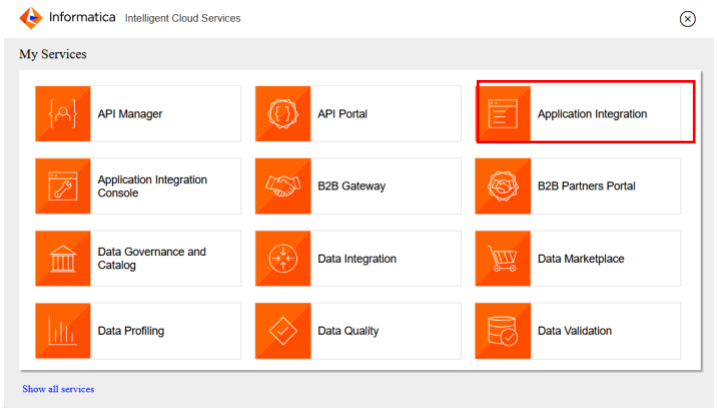
At the core of the solution is Informatica's low-code/no-code AI orchestration framework that is responsible for planning, orchestrating, executing and summarizing the information across one or more enterprise systems (incl. Informatica Product 360 & Informatica Supplier 360, relation Systems, etc.) based on the user intent. In addition to bringing data high quality trusted data from Informatica MDM, the solution leverages the data quality, business glossary, field / column names from with Informatica's Intelligent Data Management Cloud (IDMC) to dynamically select dataset based on quality threshold, improve summarization through business / enterprise context, and generate SQL queries based for custom object models, thus improving the accuracy, relevancy and reliability of the generated response.

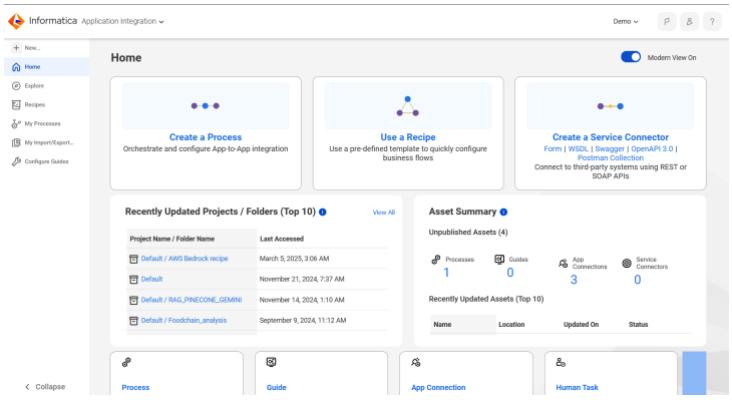
Prework

1. Login to [Informatica Intelligent Data Management Cloud] with the credentials provided.



2. From the list of services displayed, click on the **Application Integration** Service. You'll be taken to the Application Integration service.

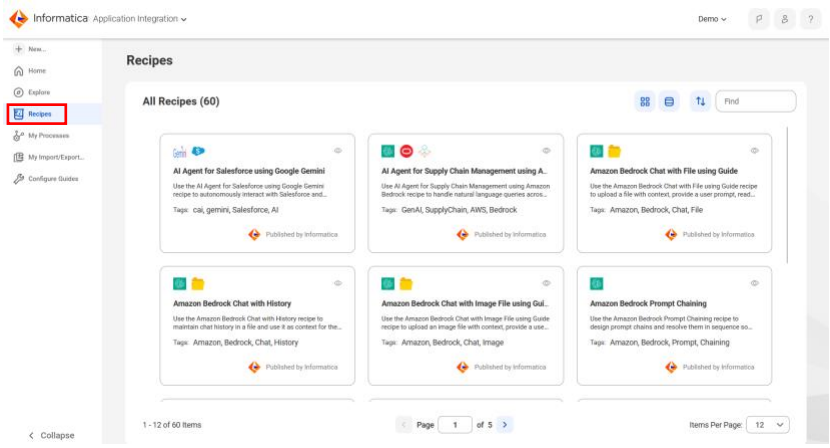




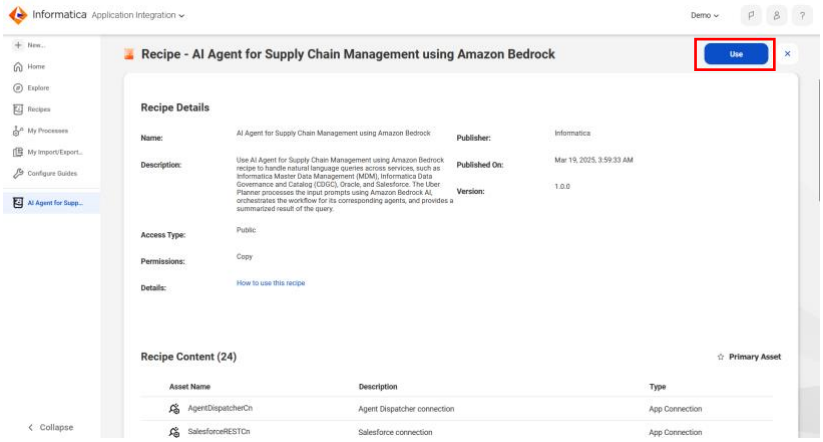
Step 1: Import the Recipe

1. Click on the **Recipes** menu option on left of the page. This will display the list of available Recipes.

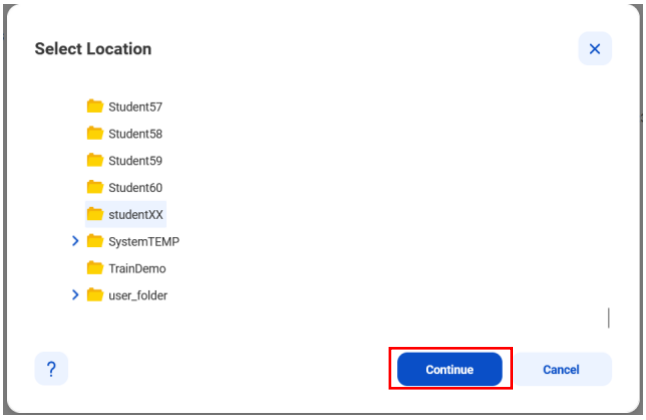
   a. Search for **AI Agent for Supply Chain management using Amazon Bedrock** using the search bar near the top right and select the Recipe.



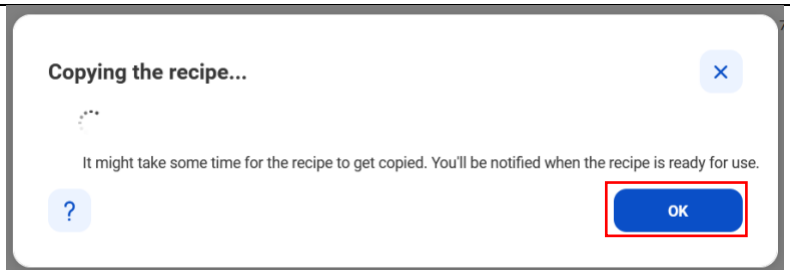
   b. Click the **Use** button.



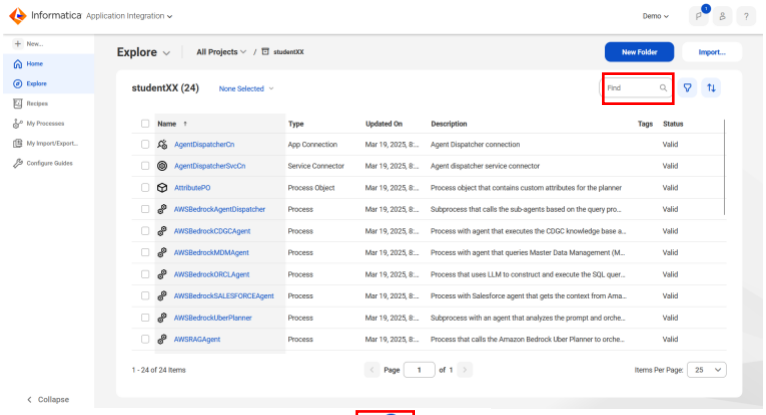
   c. A pop-up will appear asking for the location where the Recipe assets will be imported. Select the **Project Folder** that has been assigned to you (i.e. **student01** for Student01) and click **Continue**. As mentioned previously, the lab documentation will use the studentXX folder. Please **DO NOT USE** this project folder.



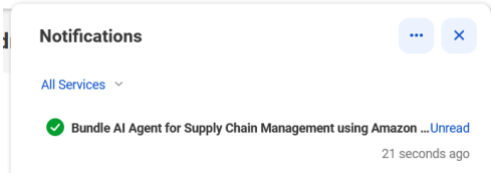
d. A pop-up will appear stating the Recipe is being copied. Click **OK** and close the recipe. The recipe is now getting installed in the folder. This may take 2-3 minutes.



2. Click on **Explore** on the left-hand side, and search for your assigned project folder using the **Find** search bar near the top right. Select the project folder where you deployed the recipe.



You will see all the Recipe assets have been created. If they do not appear, refresh the page until you see a notification appear in the top right, which will indicate the recipe has been copied.



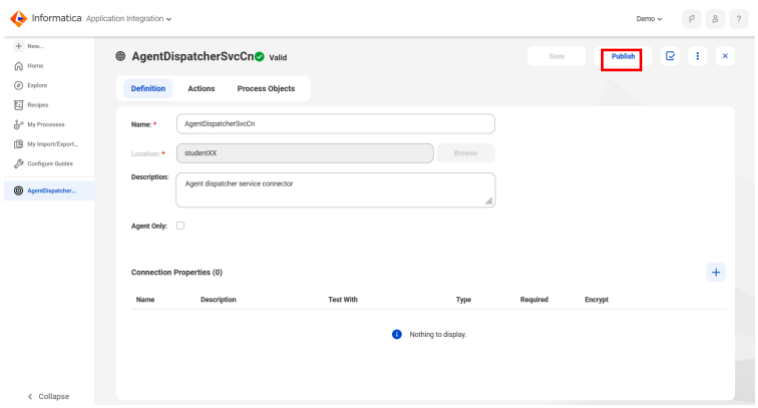
Please take some time to review the assets in the recipes

Asset Name	Description
Connector Assets	
CDAMSvcCn	Service connector for Informatica's Cloud Data Access Management Agent Proxy for governed data access to Oracle database
CDAMCn	Connector for Informatica's Cloud Data Access Management Agent Proxy for governed data access to Oracle database
BedrockCn	Amazon Bedrock Connector
AgentDispatcherSvcCn	Service connector to connect to sub-agent /task (AWSBedrockMDMAgent, AWSBedrockCDGCAgent, or AWSBedrockORCLAgent)
AgentDispatcherCn	Connects to the appropriate sub-agent /task (AWSBedrockMDMAgent, AWSBedrockCDGCAgent, or AWSBedrockORCLAgent) based on the execution plan
Agent Orchestrator Assets	
AWSBedrockAgentDispatcher	Subprocess that calls the sub-agents / task based on the query prompt and dispatches the workflow to the right agent at run time.
Agent Assets	
AWSRAGAgent	Main Agent Process that is the responsible for planning, orchestrating, executing and summarizing the information across multiple systems based on the user prompt
AWSBedrockUberPlanner	Subprocess with an agent that analyzes the prompt and orchestrates the workflow to invoke sub-agents
Supply Chain AI Assistant	Simple chat interface for Supply Chain AI Assistant
Bedrock Summarizer	Subprocess that summarizes the information retrieved from multiple systems using LLM
ContextForAWSBedrock	Sub-process to retrieve the base64 encoded prompt template for the planner, summarizer and selectors
Sub Agents / Task	
AWSBedrockMDMAgent	Agent process that gets product and supplier details from Informatica Master Data Management (MDM)
AWSBedrockCDGCAgent	Agent process that fetches metadata (incl. data quality business glossary, field / column names for custom data objects) from Informatica Cloud Data Governance and Catalog
AWSBedrockORCLAgent	Agent process that leverages the metadata from Informatica Cloud Data Governance and Catalog and LLM to construct a SQL query to fetch governed data from Oracle database
MDM Knowledge Bases Executor	Sub-Process to get product and supplier details from Amazon Bedrock Knowledge Base
CDGC Knowledge base Executor	Sub-Process gets metadata (incl. data quality business glossary, field / column names for custom data objects) from Amazon Bedrock Knowledge Base
Process Objects	
UberPlannerPO	Process object that contains custom attributes for the Uber Planner
PlannerPO	Process object that is contains attributes for the Planner
AtrributePO	Process object that contains custom attributes for the planner

Step 2: Configure the connectivity

In this step you will configure the connectivity to AWS Bedrock, Oracle and Salesforce. After configuring the connectors, we will publish them and the associated processes, which will be used in the next lab.

1. From your specific project folder, select the **AgentDispatcherSvcCn** Service Connector and select **Publish**.



   You should see a popup saying the asset was published successfully. Press on **'X'** on top right to close the Connector window.

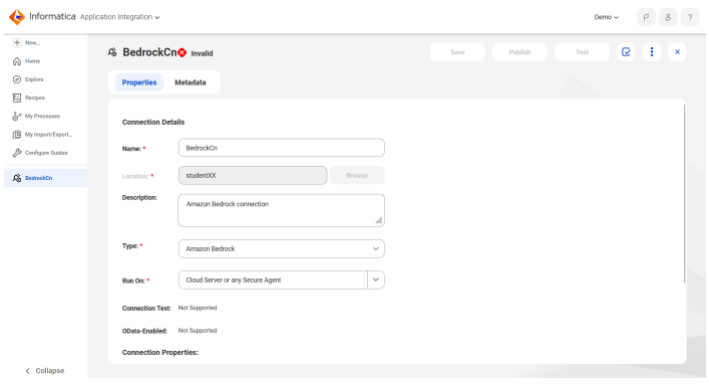
2. Repeat this process for **CDAMSvcCn** Service Connector.

3. From your project folder, select the **BedrockCn** App Connection.

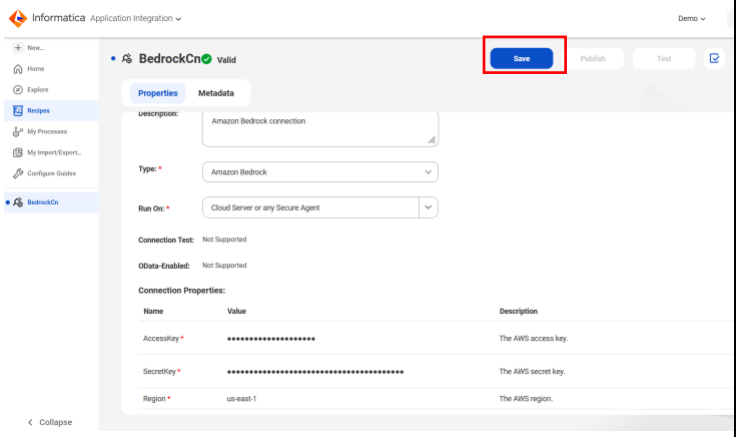
   Scrolling down, enter the below **Connection Properties**

   AccessKey: AKIA4LKFNB2TWFIYRJ4P
   SecretKey: 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
   Region: us-east-1

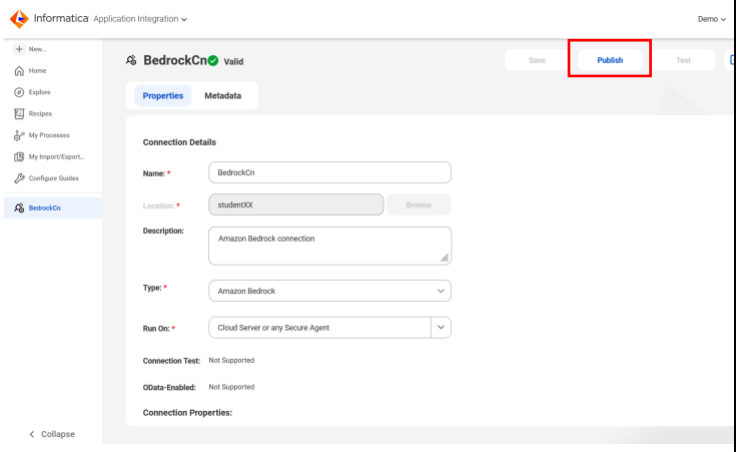
   Note: To configure the property, click on the space under **Value** next to property name and type the value.



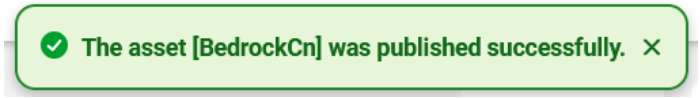
Click the **Save** button. The connection should now be **Valid**.



Once saved successfully, the connection can now be published. Click the **Publish** button.



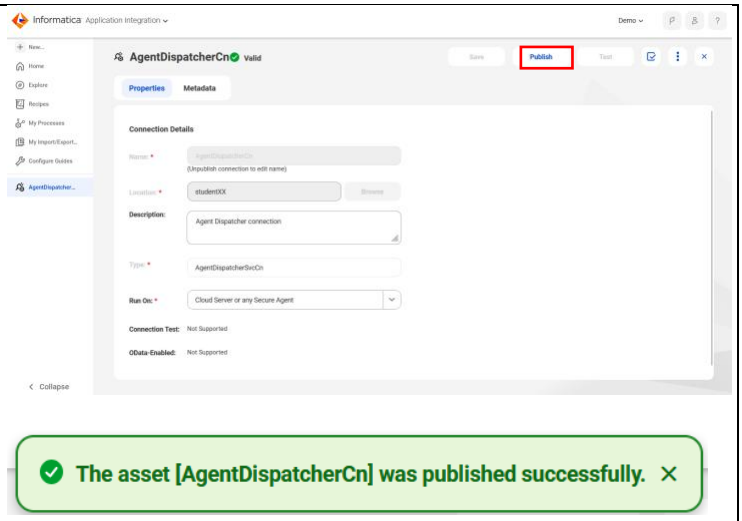
A popup will appear confirming the publication was successful.



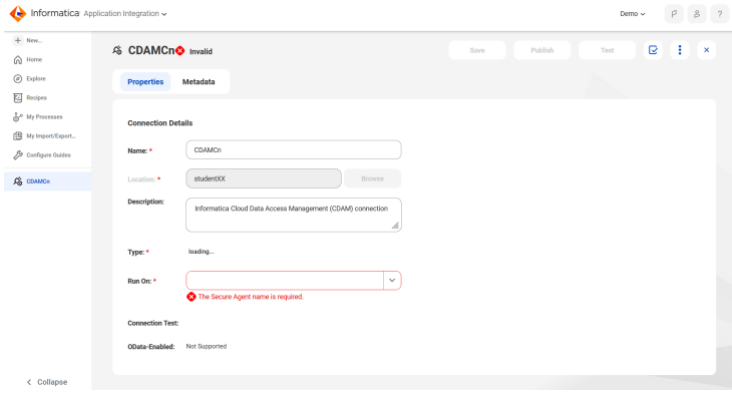
Close the BedrockCn connection.

4. From your specific student project, select the **AgentDispatcherCn** App Connection. We will use the default configuration for this connection, so click the **Publish** button. Again, the confirmation popup should appear.

   Close the AgentDispatcherCn connection.



5. From your specific student project, select the **CDAMCn** App Connection.

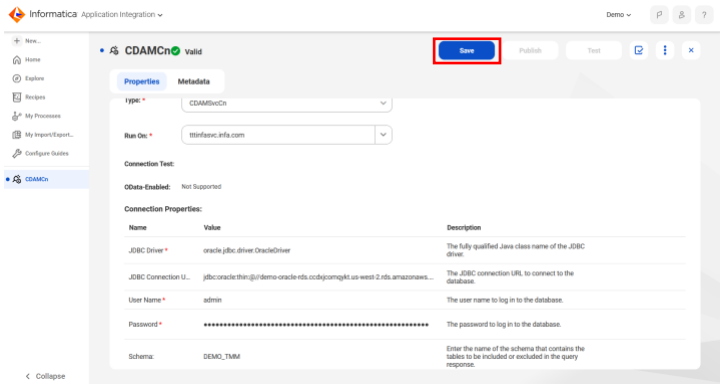


   Enter the following connection configurations:

   Run on: select modulabs.master
   JDBC Driver: oracle.jdbc.driver.OracleDriver
   JDBC Connection URL: jdbc:oracle:thin:@//demo-oracle-rds.ccdxjcomqykt.us-west-2.rds.amazonaws.com:1521/ORCL
   Username: admin
   Password: Password12345
   Schema: DEMO_TMM

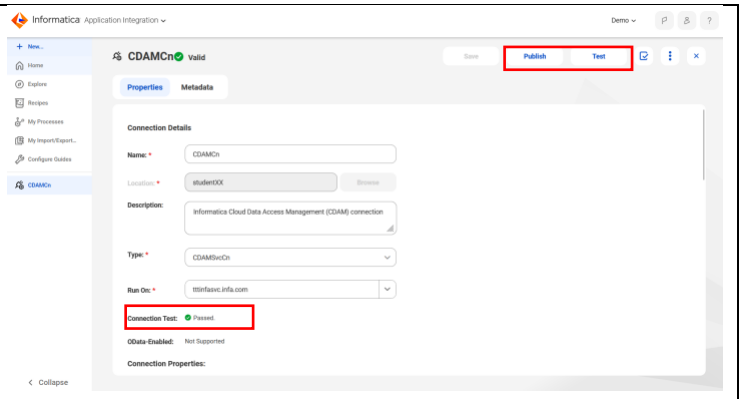
   Click **Save**.

   Note: To configure the property, click on the space under **Value** next to property name and type the value.



Once successfully saved, the **Test** and **Publish** buttons should now be available. Click the **Test** button and wait until it passes successfully. This may take a few minutes on the first test.

After testing successfully, click the **Publish** button.



The confirmation popup should appear.
Close the CDAMCn connection.

Step 3: Setup and Publish the Processes

In this step you will publish the processes that leverage the connectivity you just setup, which will dynamically orchestrate the required AI requests from the various data sources.

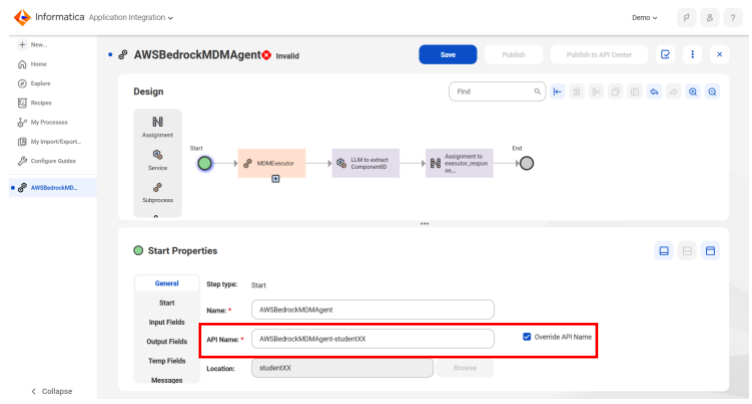
1. From your specific student project, select the **AWSBedrockMDMAgent** Process.

   In the **General** configuration towards the bottom of the screen, tick the **Override API Name** check-box, and add **-studentXX** to the **API Name** value, but replace the **XX** with your specific project folder name (e.g. if you are student01, then you would have **AWSBedrockMDMAgent-student01)**

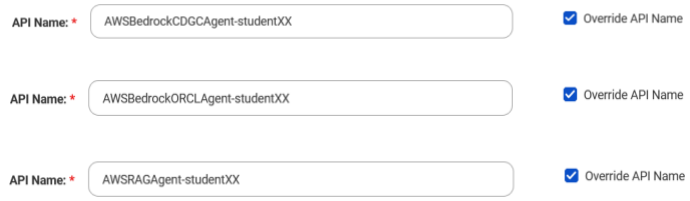
   Click the **Save** button.

   You will see the popup saying it was saved successfully. Notice this Process is still Invalid, this is because there are dependent sub-processes that are not published yet.

   Close the process.

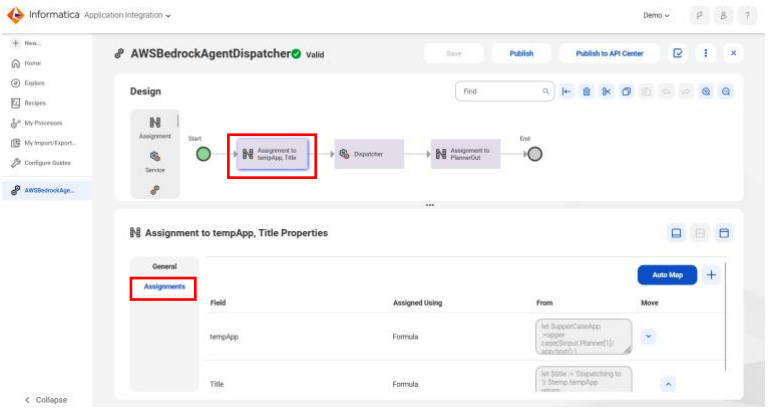


2. Repeat the above instructions for the **AWSBedrockCDGCAgent, AWSBedrockORCLAgent** and **AWSRAGAgent** sub-processes. Some of these processes may already be Valid.

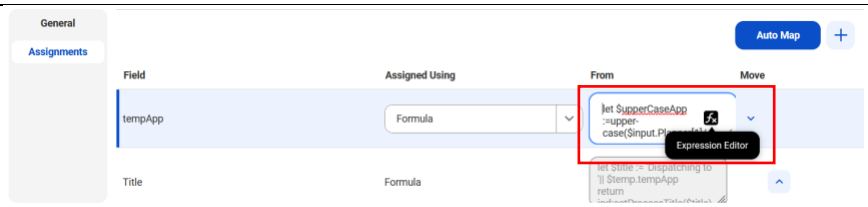


3. From your specific student project, select the **AWSBedrockAgentDispatcher** Process. Select the first step in the process after the start, labeled **Assignment to tempApp, Title**. In the bottom configuration section, select **Assignments** on the left-hand side.

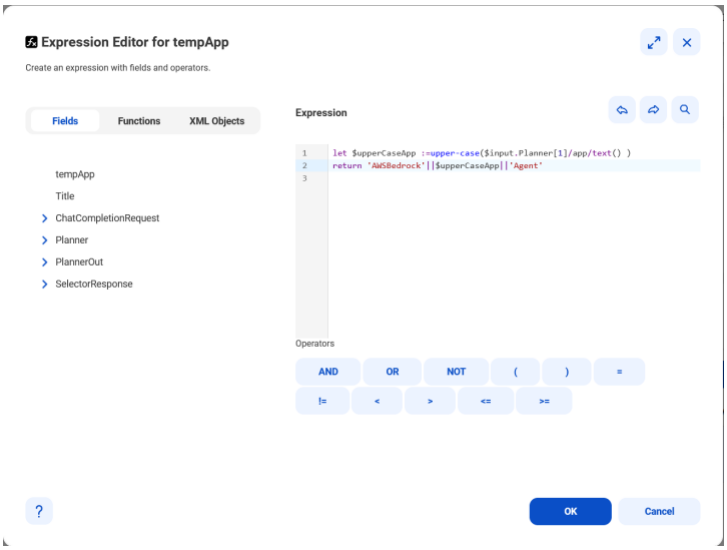
   Click on the text area for the **tempApp** field in the **From** column. Click on the black function button **fx** that appears.

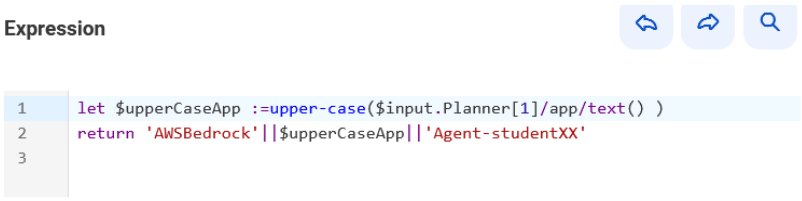


The function editor will popup.



In the expression, append **-studentXX** to the word **Agent**, within the single quotes, making sure you use your project folder name instead of the **XX** in the lab guide (e.g. if you are student 01, then the text should read 'Agent-student01'). Make sure the case matches the same suffix used in step 2 above. Click the **OK** button and click anywhere on the screen to deselect the text area.



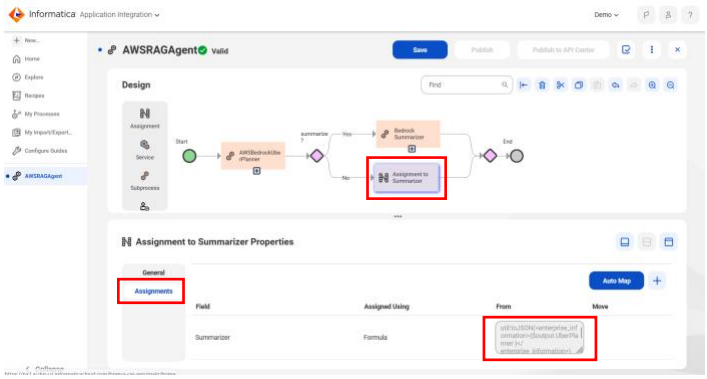


Click **Save** and **Publish**. The popup confirming the publication was successful will appear.

4. From your specific student project, select the **AWSRAGAgent** Process.

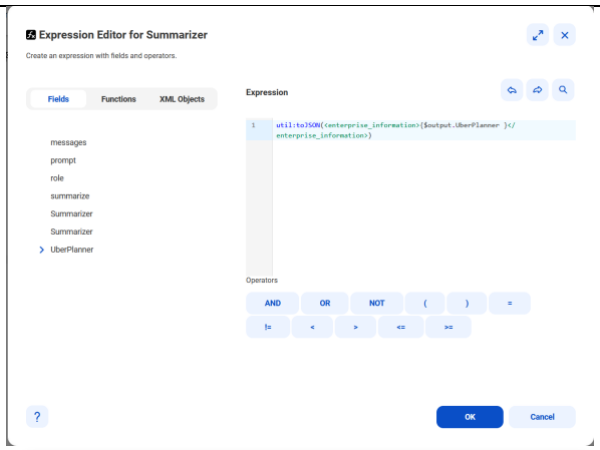
   Select the **Assignment to Summarizer** Assignment step in the process and select the **Assignment** configuration at the bottom of the screen. Ensure the **Assigned Using** for the **Summarizer** Field is set to **Formula**.

   Click on the **From** field and click on the black function button **fx** that appears.

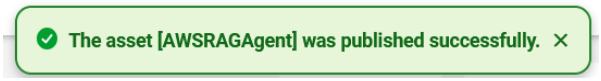


The function editor popup will appear and enter the following and click **OK**.

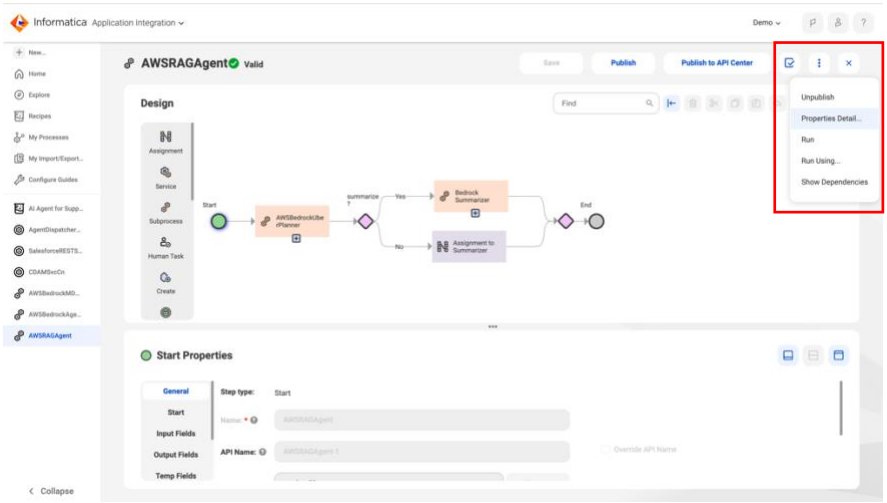
util:toJSON(<enterprise_information><UberPlannerPO>{$output.UberPlanner/}</UberPlannerPO></enterprise_information>)*



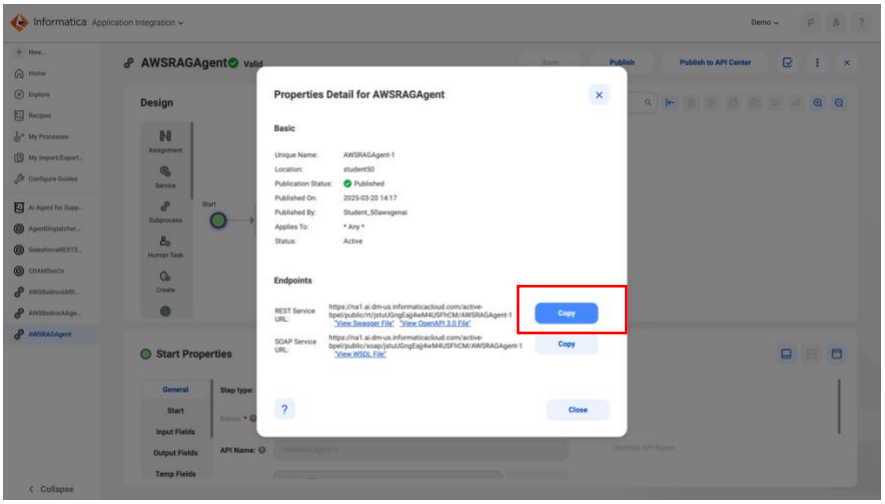
Click the **Save** and **Publish** buttons. Again, the confirmation popup should appear.



5. Click the ellipsis (3 dots) button at the top right of the screen and navigate to **Properties Detail…**

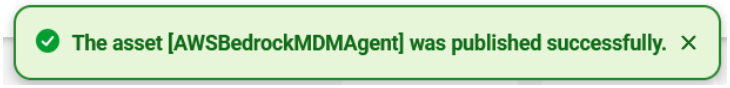


Copy the REST Service URL and paste it to a notepad for Lab 2: Build high data quality GenAI assistants using Amazon Bedrock agents and Informatica).

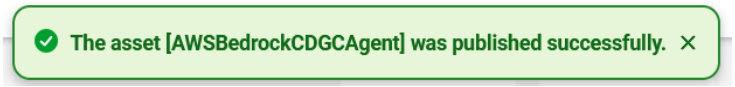


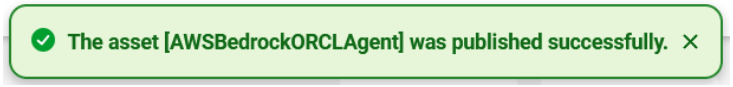
Close the Process.

6. From your specific student project, select the **AWSBedrockMDMAgent** process. Click the **Publish** button. Again, the confirmation popup should appear.

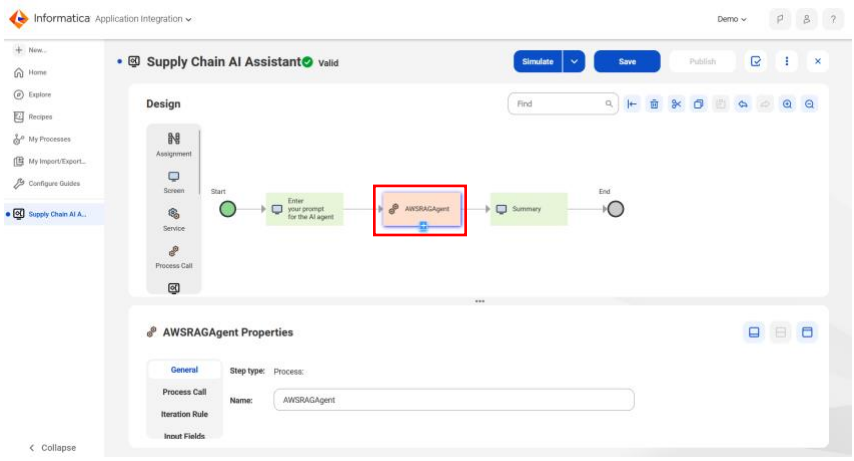


7. Repeat Step 6 to Publish the following **AWSBedrockCDGCAgent** and **AWSBedrockORCLAgent**

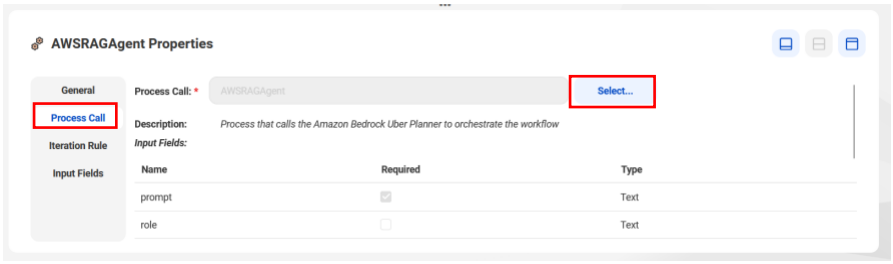




8. From your specific student project, select the **Supply Chain AI Assistant** Guide asset.

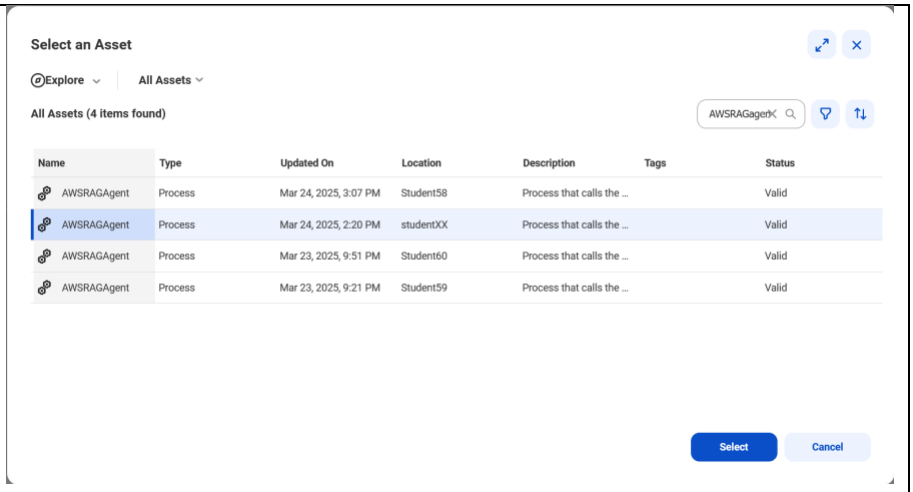


Select the **AWSRAGAgent** Process step and click on t he **Process Call** configuration option on the bottom left.

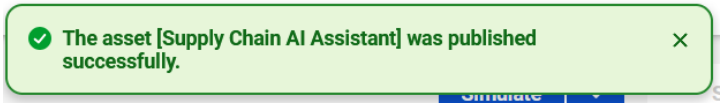


Click the **Select** option and the Process selection popup will appear. Select the

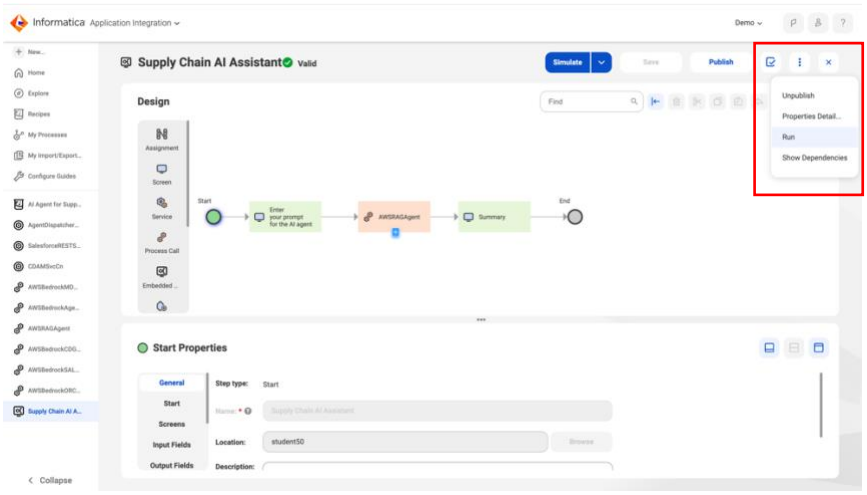
AWSRAGAgent that is in your specific student folder. Click the **Select** button.



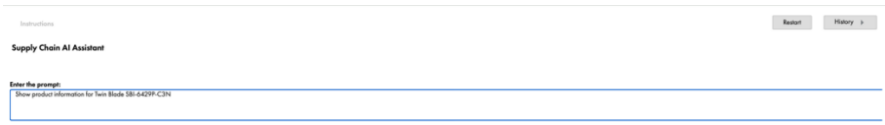
9. Click the **Publish** button. Again, the confirmation popup will appear.



10. Click the ellipsis (3 dots) icon at the top right of the screen and select **Run**



11. Enter the following prompt:
    a. Show product information for Twin Blade SBI-6429P-C3N
    b. Click **Continue**



12. The Guide will return a summary response with product information in JSON format.

Click the **Restart** button

13. Repeat steps 10 with the following prompts:
    - **a.** Who is the supplier for SYS-621H-TN12R?
    - **b.** What is the lead time for Montage for delivering component SYS-621H-TN12R?

**********Congratulations you have completed Lab 1. Please proceed to Lab 2 - Link*********