



IBM App Connect Enterprise

How to implement an AI RAG pattern using
IBM watsonx.ai and Pinecone Vector
Database Request nodes

Featuring:

ACE Toolkit, File Input node, IBM watsonx.ai Request node, Pinecone Vector Database Request node, Compute node.

1. INTRODUCTION.....	3
2. CONFIGURE IBM WATSONX.AI AND PINECONE.....	4
2.1 SET UP YOUR PINECONE VECTOR DATABASE.....	4
2.2 SET UP YOUR IBM WATSONX.AI ENVIRONMENT.....	6
3. CONFIGURE THE ACE TOOLKIT DEVELOPMENT ENVIRONMENT.....	14
3.1 IMPORT THE PROJECT INTERCHANGE FILE	20
3.2 DEPLOY THE POLICY PROJECT	29
3.3 UPDATE THE PHASE1_INDEXING MESSAGE FLOW	30
3.4 UPDATE THE PHASE2_RETRIEVALANDGENERATION MESSAGE FLOW.....	35
3.5 TEST THE MESSAGE FLOWS	45
END OF LAB GUIDE	50

1. Introduction

The purpose of this LAB is to provide existing developers who already have a degree of familiarity with ACE, a demonstration of two new ACE features:

- The IBM Watsonx.ai Request node (recently added in ACE 13.0.1.0)
- The Pinecone Vector Database Request node (recently added in ACE 13.0.3.0)

This means you must be at version ACE 13.0.3.0 or higher when completing the lab.

This lab assumes that you have at your disposal:

- A Windows VM Image which has ACE 13.0.3.0 installed.
- An IBMid with which to access the IBM Watsonx.ai cloud service
- A suitable id with which to access Pinecone (for example a gmail id)

2. Configure IBM watsonx.ai and Pinecone

2.1 Set up your Pinecone vector database

1. Open a web browser tab to the URL www.pinecone.io. The website should look similar to the picture below. Click the **Sign up** button:

The screenshot shows the Pinecone homepage with a navigation bar at the top. The main heading is "The vector database for scale in production". Below it are two buttons: "Start Building" and "Get a Demo". To the right is a detailed diagram of the Pinecone serverless architecture, which includes components like GPUs, Memory + SSD Index Cache, and Blob Storage, connected via WRITE, EMBEDDING, RANKING, and MODELS, INDEX BUILDERS, and QUERY WORKERS to Raw Data and Indexes.

2. You can choose whether you would like to create a dedicated account, but the example shown here chooses to use a Google account to continue. Click the **Continue with Google** button:

The screenshot shows the Pinecone sign-up page. It features a "Continue with Google" button highlighted with a red box. Other options include "Continue with GitHub" and "Continue with Microsoft". There is also a field for entering an email and a checkbox for accepting the Master Subscription Agreement. A "Continue >" button is present. On the right side, there is a dark panel with the text "Long-term memory for AI" and "Start building for free".

3. Type in your account Google account details or select it if already available:

The screenshot shows the 'Sign in with Google' interface. It displays a list of accounts: 'Ben Thompson' (selected) and 'Ben Thompson'. Below the accounts, there is a link to 'Continue to pinecone.io'.

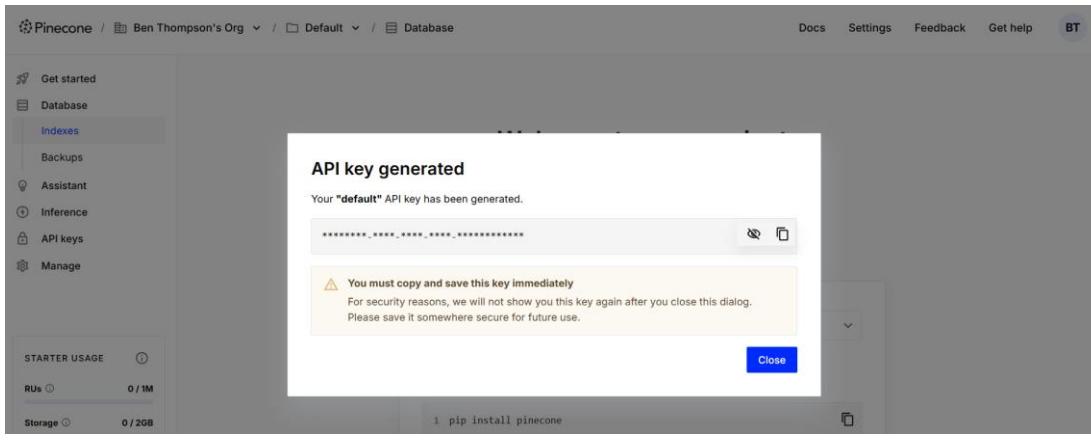
4. Click the **Continue** button:

The screenshot shows the 'Sign in to pinecone.io' page. It includes a note about Google sharing user information and links to manage sign-in settings. The 'Continue' button is highlighted with a red border.

5. Select your preferred coding language and then click Continue:

The screenshot shows the 'Create your account' form. It asks for first name ('Ben'), last name ('Thompson'), purpose of use ('Personal'), and preferred coding language ('Java'). The 'Preferred coding language' field and the 'Continue' button are highlighted with red borders.

6. Click the skip button and you will see a screen providing you with an API key which has been generated on your behalf. **Be sure to copy the key somewhere safe (such as a notepad file) as you will need it when connecting ACE in a few steps time:**

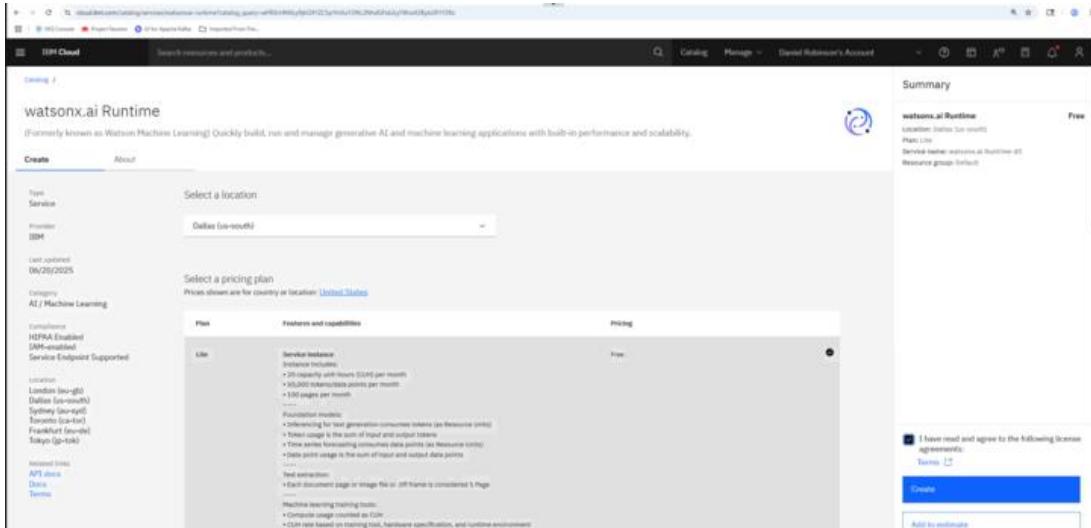


2.2 Set up your IBM watsonx.ai environment

7. We will start by creating an instance of the watsonx.ai runtime. In a new web browser tab, go to the following URL:

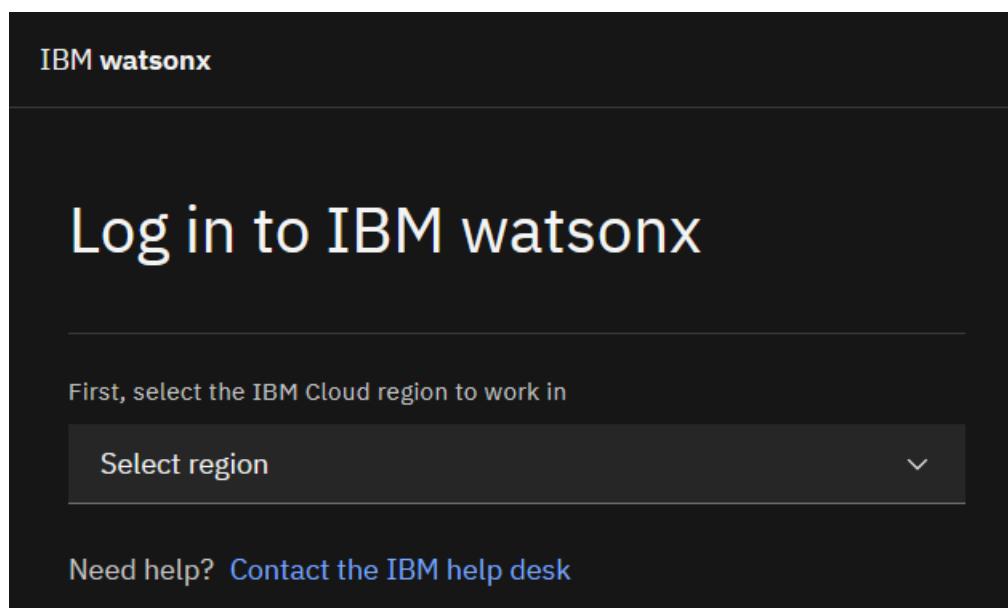
<https://cloud.ibm.com/catalog/services/watsonxai-runtime>

... and log in with your IBMid. Select the terms and conditions box on the right side of the screen and click the Create button:

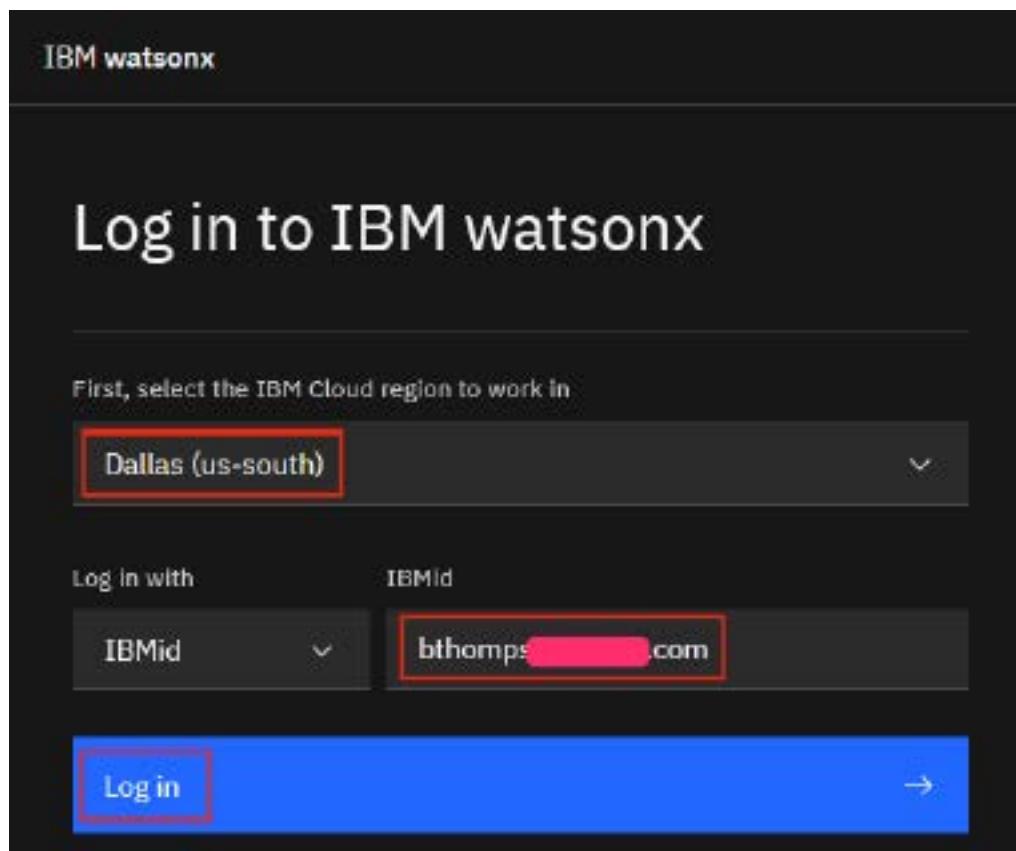


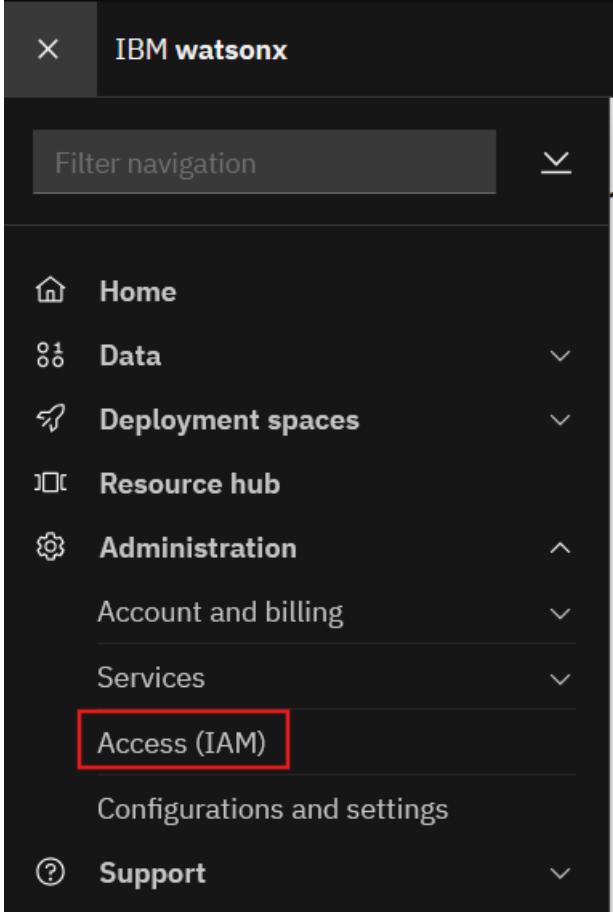
This will ensure that you have a watsonx.ai runtime available to associate with a service

8. Open a new tab in your web browser and go to the URL:
<https://dataplatform.cloud.ibm.com/wx/home?context=wx>

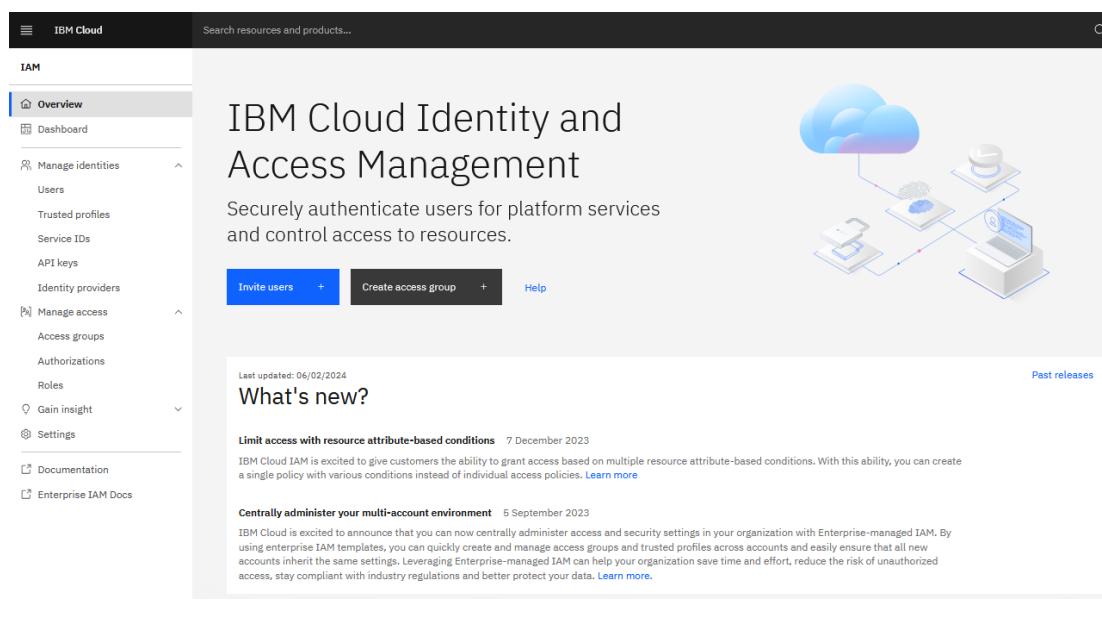


You may have your own preference for which IBM Cloud region you wish to work in. For simplicity, by default the ACE flows were set up to expect to use **Dallas (us-south)** when creating this Lab guide. If you have already set up access to watsonx previously in a different geography, then it's probably easier to change ACE to talk to a different watsonx region later in the lab rather than attempt to change your watsonx arrangements! You will also need to supply your IBMid as shown below:



	9. Once you are logged in, use the navigation menu in the top left corner of the web page and select the menu option Administration > Access (IAM) :
	

Choosing this option will take you to the **IBM Cloud Identity and Access Management** web page, which should look like this:



The screenshot shows the IBM Cloud Identity and Access Management (IAM) Overview page. The left sidebar contains navigation links for Overview, Dashboard, Manage identities (Users, Trusted profiles, Service IDs, API keys, Identity providers), Manage access (Access groups, Authorizations, Roles), Gain insight, Settings, Documentation, and Enterprise IAM Docs. The main content area features the title "IBM Cloud Identity and Access Management" and the subtext "Securely authenticate users for platform services and control access to resources." Below this are two buttons: "Invite users" and "Create access group". A "What's new?" section is present, noting the update "Last updated: 06/02/2024" and the addition of "Limit access with resource attribute-based conditions" on December 7, 2023. A "Centralized administer your multi-account environment" note is also shown.

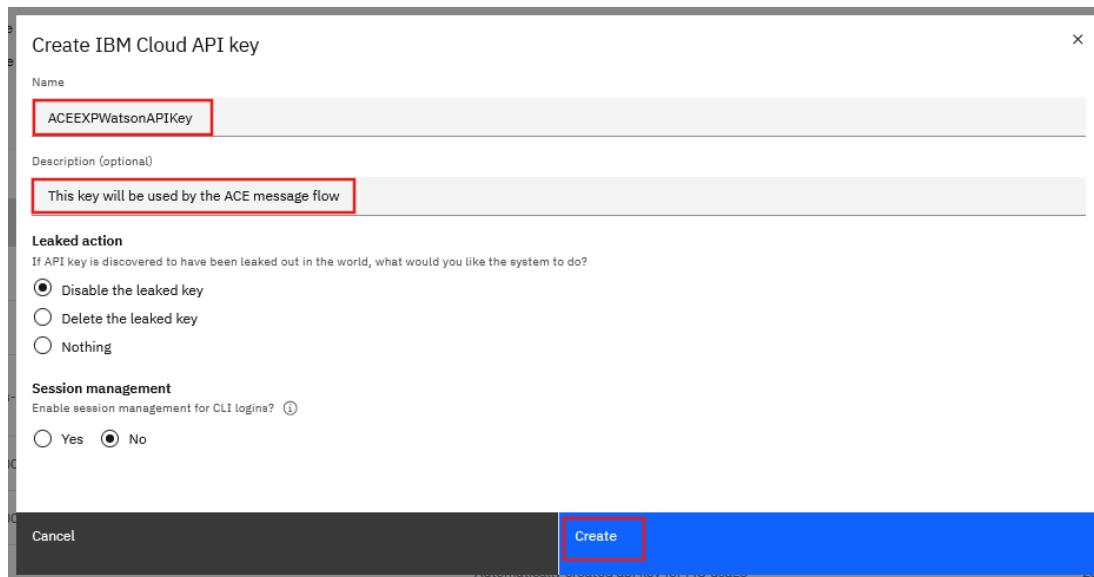
10. From the **IBM Cloud Identity and Access Management** web page, select the **API keys** option from the left navigation pane:

11. You will be shown a table containing API keys that you may have previously set up. Is this the first time you've used the IBM Cloud, then you may not have any pre-existing API keys. The top of the table will look like this:

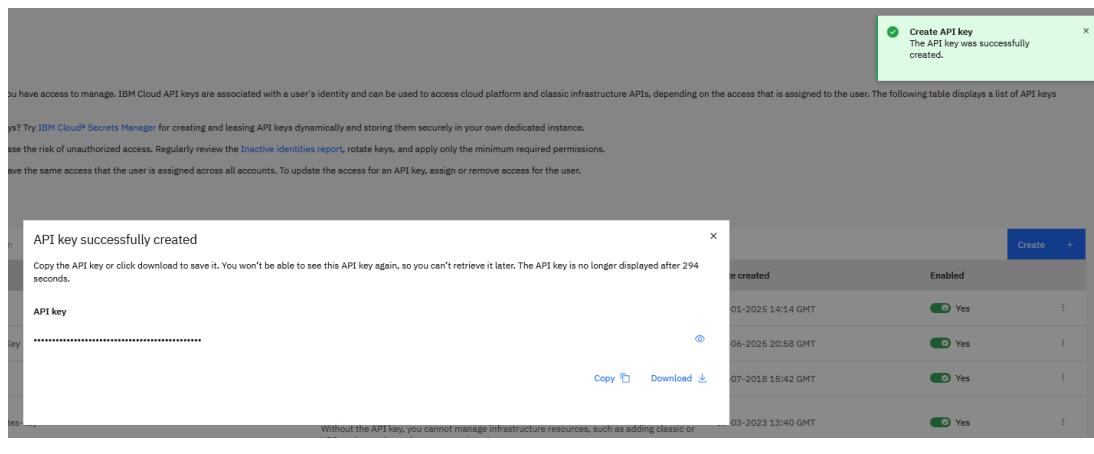
12. Click the Create button as shown below:

13. In the resulting dialog, provide a suitable Name and Description for the key.
 For example:
- **Name** = ACEEXPWatsonAPIKey
 - **Description** = This key will be used by the ACE message flow

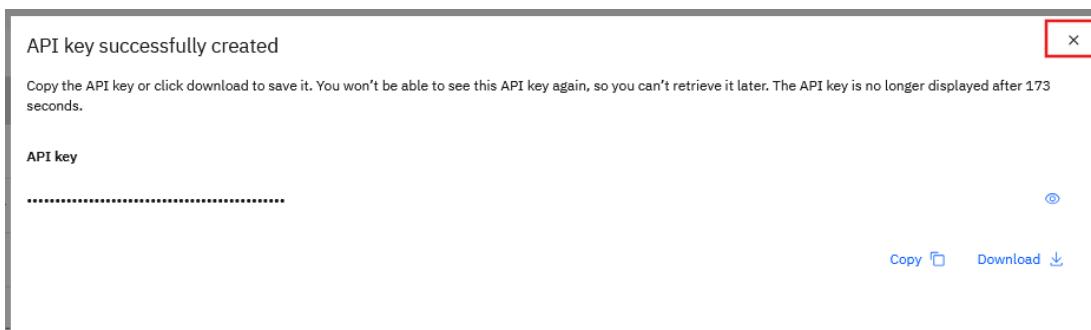
Click the **Create** button:



14. A success message will appear in the top right corner, and the main dialog will report that the API key has been successfully created. **Click the Copy button and paste the API Key in to a notepad file, as you will need to use it in a few steps time, when connecting from ACE.**



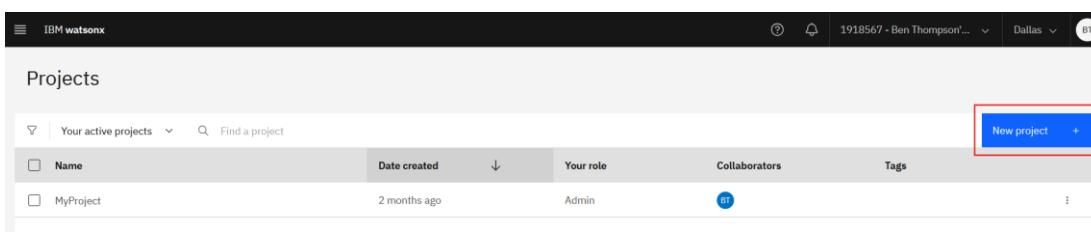
15. Once you have copied the API Key, close the dialog using the cross button in the top right corner:



16. We will now create a watsonx project. Return to the projects section of the main watsonx website page using the following URL:

<https://dataplatform.cloud.ibm.com/projects/?context=wx>

You may see other projects that you have created before in the view that is presented, or an empty table if you have not created any previous projects. Either way, click the button to Create a new project:



17. You can choose a suitable name for your project, for example:

- **Name** = ACEEXPWatsonProject

Click the **Create** button:

The screenshot shows the 'Create a project' dialog. On the left, there's a sidebar with '+ New' options: 'Local file' and 'Sample'. The main area has 'Define details' sections for 'Name' (containing 'ACEEXPWatsonProject'), 'Description (optional)', 'Tags (optional)', and 'Storage' (set to 'Cloud Object Storage-uq'). At the bottom right are 'Cancel' and 'Create' buttons, with 'Create' being highlighted by a red box.

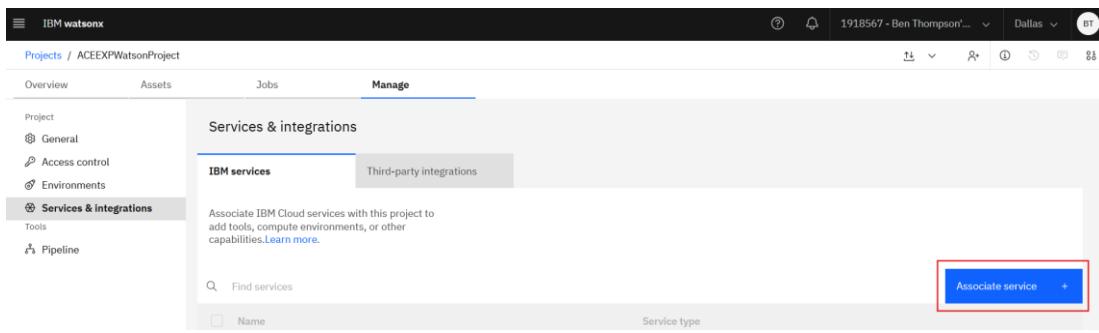
You will be shown a view of the created project like this:

The screenshot shows the 'Overview' tab of the project 'ACEEXPWatsonProject'. It features a 'Start working' section with four cards: 'Add users as collaborators', 'Add data to work with', 'Chat and build prompts with foundation models', and 'Tune a foundation model with labeled data'. Below this are sections for 'Jump back in', 'Resource usage' (showing 0 CUH), and 'Project history' (showing the creation date). The 'Manage' tab is highlighted with a red box at the top.

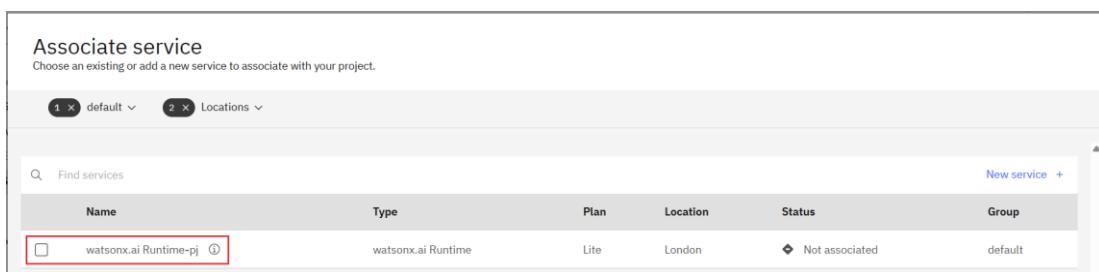
18. Click the **Services and Integrations** option from the left side menu:

The screenshot shows the 'General' tab of the project settings. The left sidebar has 'Project' and 'General' selected. Under 'General', there are sections for 'Details' (Name: ACEEXPWatsonProject, Description: 'What's the purpose of this project?', Tags: 'Add tags to make projects easier to find.', Project ID: c9d86947-11d6-4718-9643-607399f37923) and 'Storage' (Storage used: 0 Bytes, Bucket: aceexpwatsonproject-donotdelete-pr-7v8nlppssqzk29u). The 'Manage' tab is highlighted with a red box at the top.

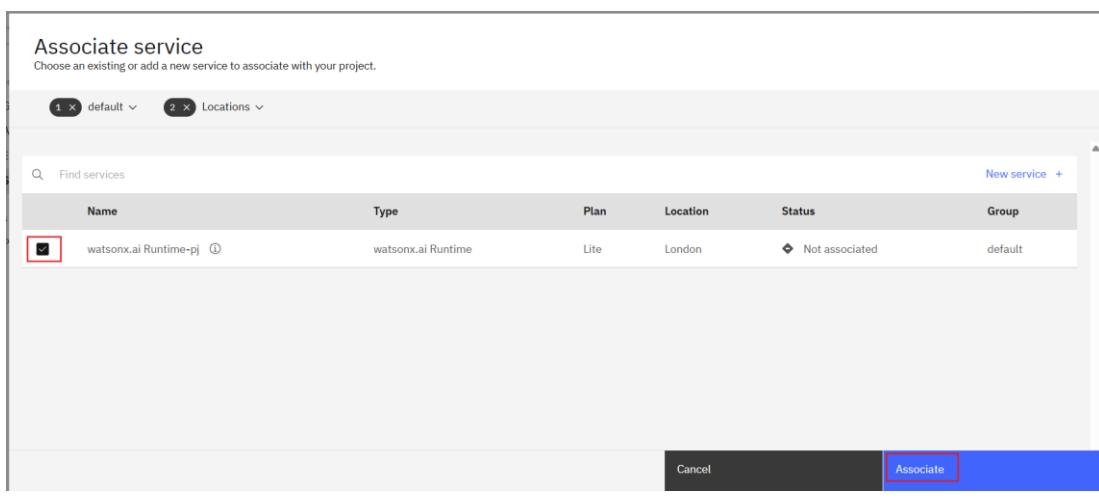
19. On the IBM services tab, click the **Associate service** button:



20. Associate the WatsonMachineLearning service:



Select the box and then click the Associate button:



21. The watsonx.ai runtime service will now be associated with the project. Next, sticking with the Manage tab, select the **General** option from the left nav as shown below:

The screenshot shows the IBM Watsonx interface with the 'Manage' tab selected. On the left, there's a sidebar with 'Project' and 'Tools' sections. Under 'Project', 'General' is highlighted with a red box. Under 'Tools', 'Pipeline' is listed. The main area is titled 'Services & integrations' and shows 'IBM services (1)'. A sub-section 'IBM services' lists 'watsonx.ai Runtime-pj' with a 'Service type' of 'watsonx.ai Runtime'. There's a search bar and a blue button labeled 'Associate service'.

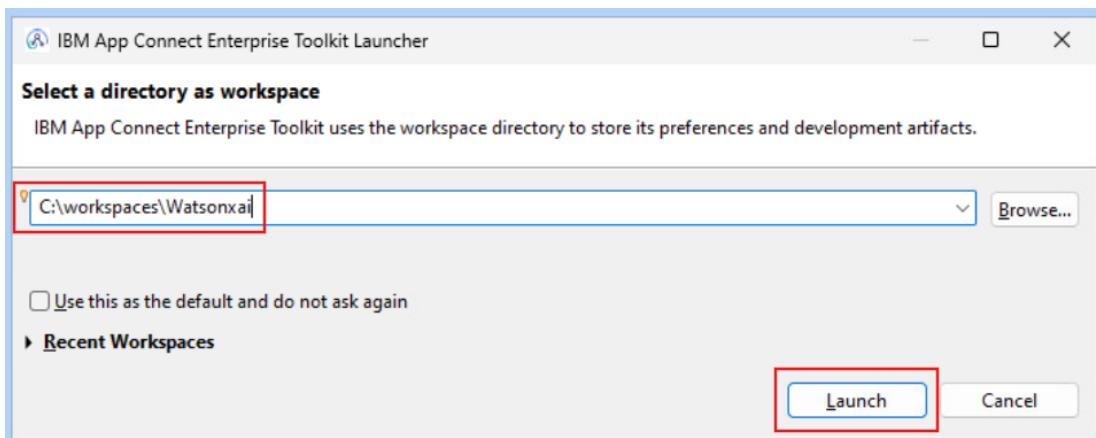
22. The **Project ID** will be displayed. Copy the Project ID value using the button next to the field, and save this away in a notepad file as you will need it in a moment, when configuring ACE:

The screenshot shows the 'General' settings page for the 'myACEProject' project. The 'General' section is selected in the sidebar. The 'Project ID' field contains the value '9e9c9a68-3381-48d4-ae0a-aec0d139b9b4', which is highlighted with a red box. Other fields include 'Name' (myACEProject), 'Description' (What's the purpose of this project?), and 'Tags' (Add tags to make projects easier to find).

- 23.

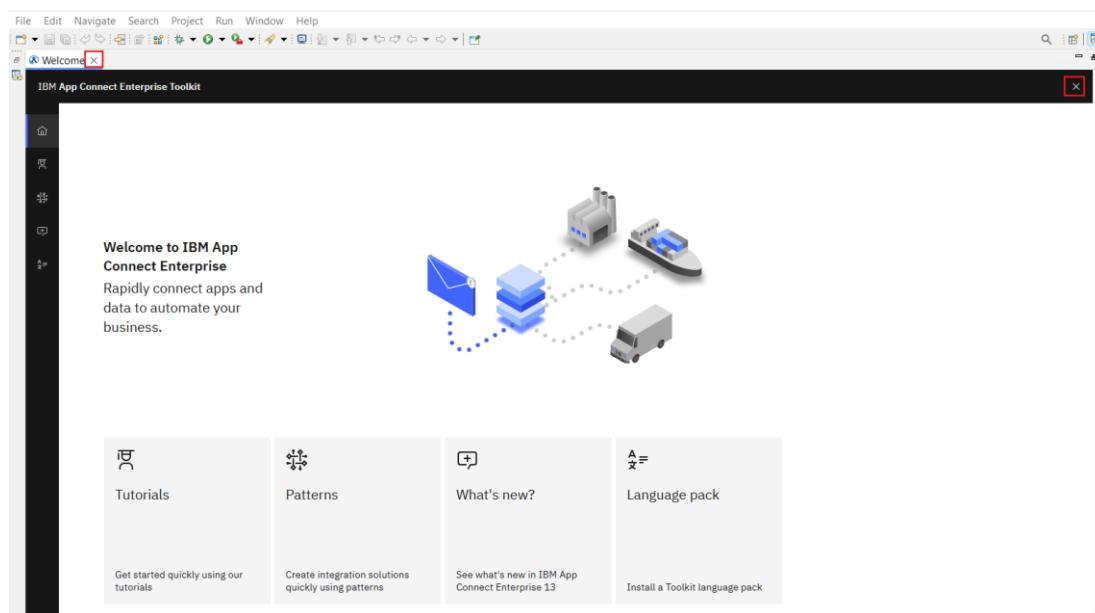
3. Configure the ACE Toolkit Development environment.

24. In the windows environment, open the v13.0.3.0 IBM App Connect Enterprise Toolkit. Create a new workspace for the work in this lab guide, named C:\workspaces\Watsonxai:

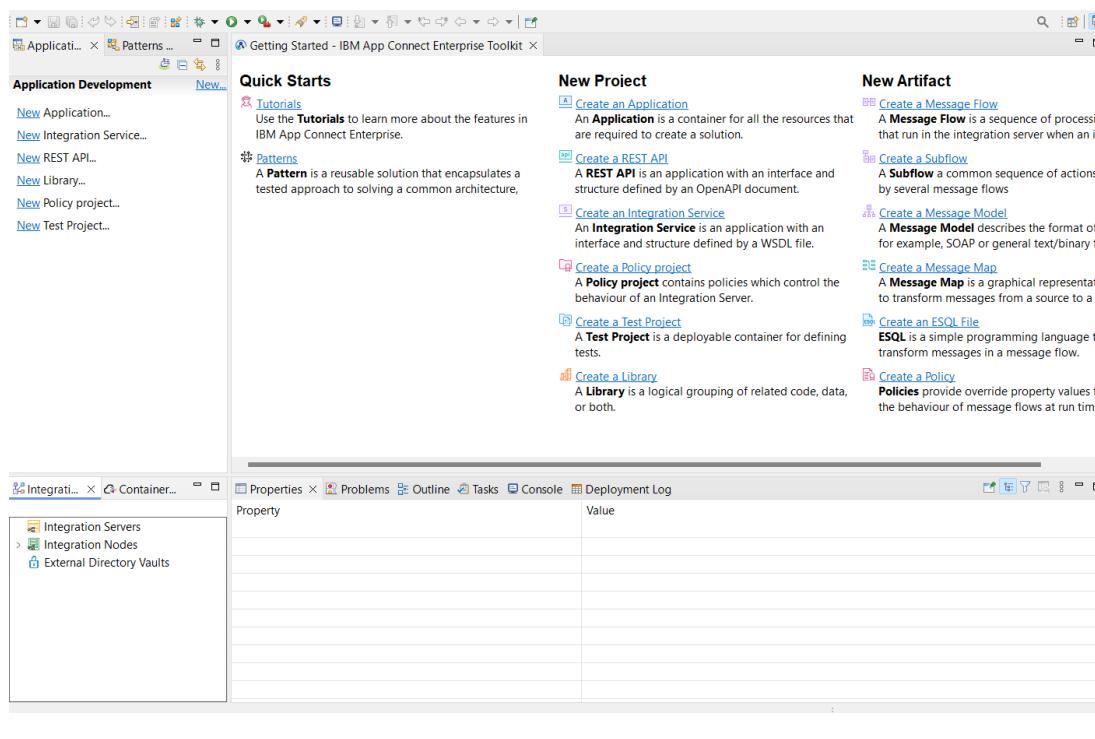


25. If this is the first time the Toolkit has been run after it was installed then when the Toolkit first launches you will see a pop up offering you the chance to exclude Toolkit from Windows Defender checks. This decision is up to you, but for best performance we recommend you choose the Exclude option. If you have already run the Toolkit, perhaps in earlier labs, then this warning will not appear.

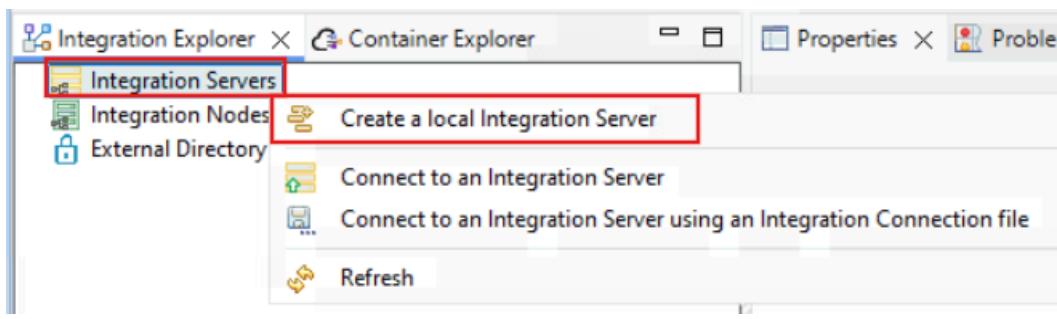
26. Once the Toolkit is started you will be looking at the “Welcome to IBM App Connect Enterprise Toolkit” window as shown below. Click one of the cross symbols (either one will work – they have the same effect, to dismiss the welcome screen):



Once the Welcome screen is closed you should see a view similar to this:

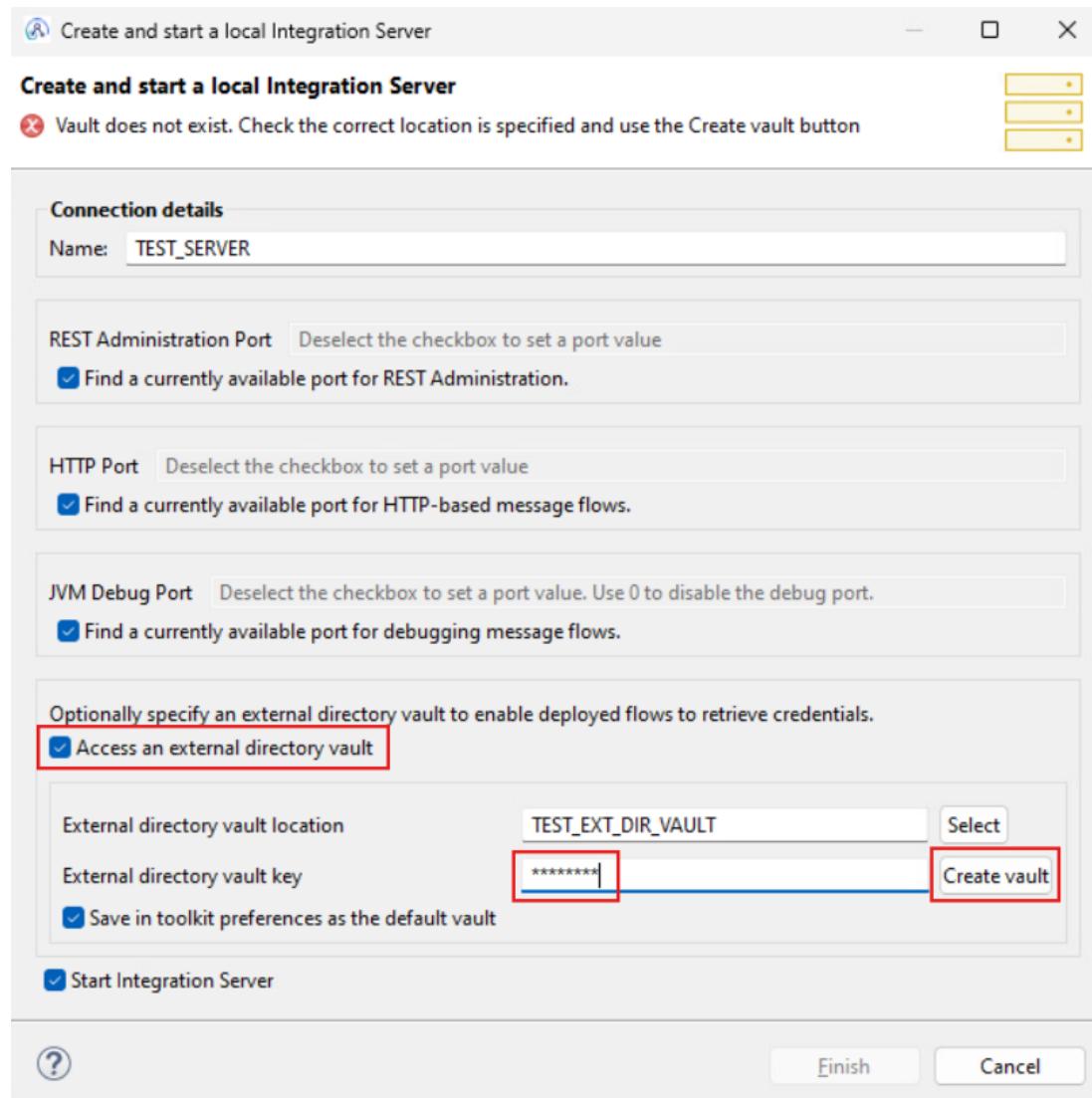


27. In the Integration Explorer window, right click on **Integration Servers** and select **Create a Local Integration Server**:

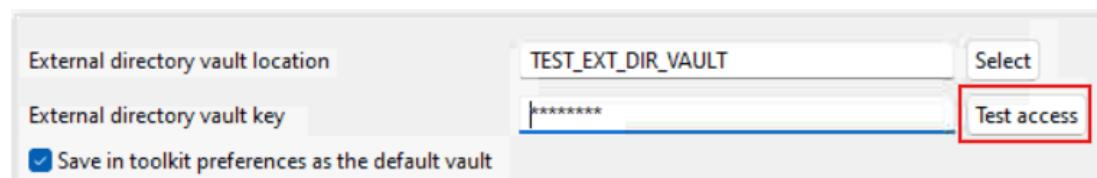


28. We will leave most of the settings with their defaults in the “**Create and start a local Integration Server**” dialog, however ...

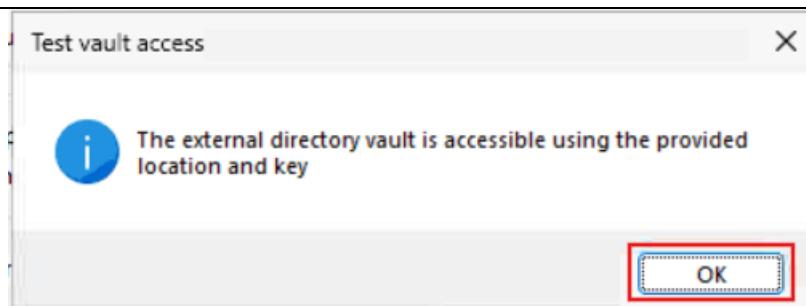
- Select the tick box to “**Access an external directory vault**”
- The “**external directory vault key property**” will become available. Type a value for the vault key. We recommend the value “**passw0rd**”. If you choose your own value please remember it as you may need it later.
- Click the “**Create vault**” button



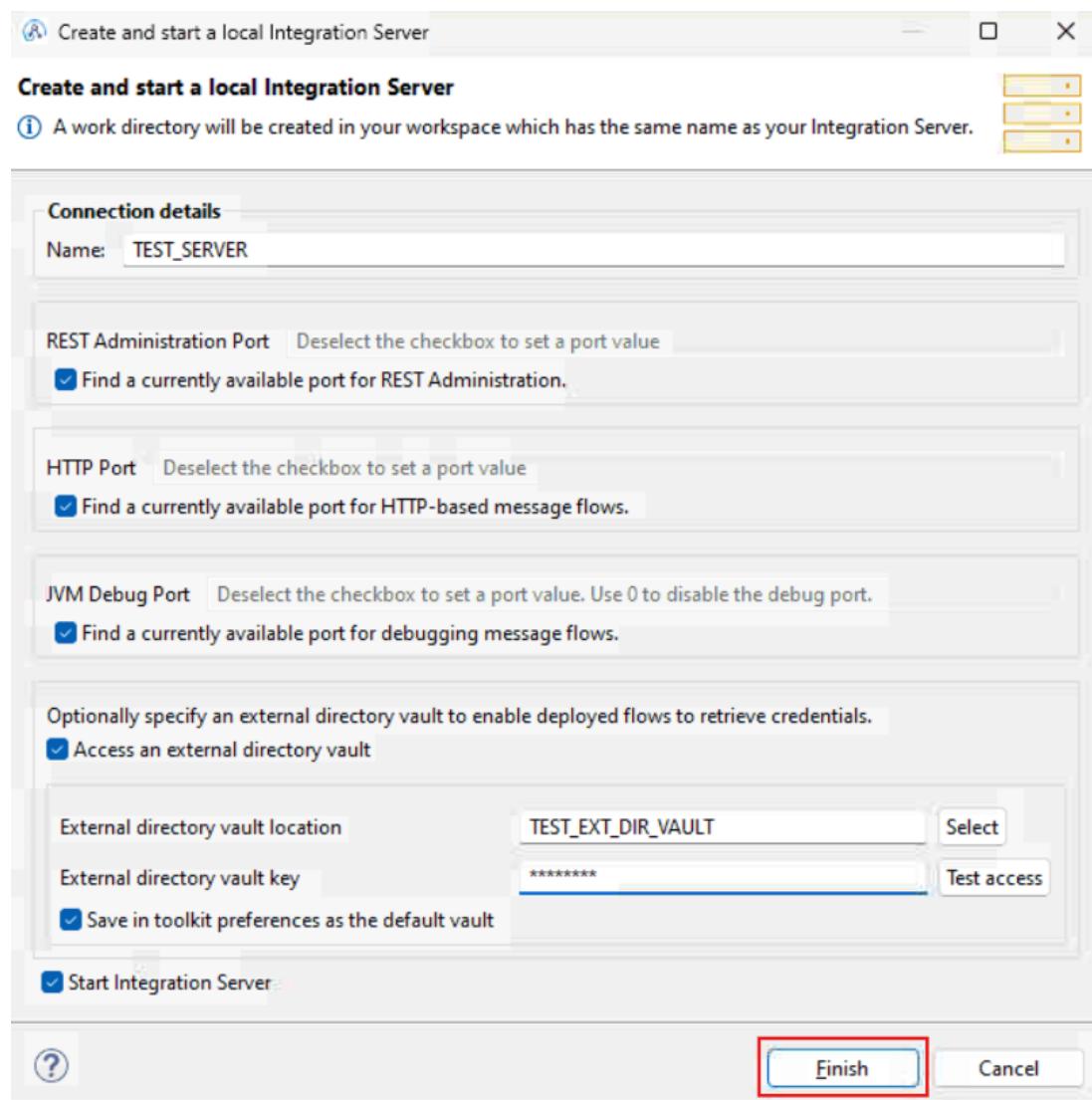
You will notice that the text on the button will change to say “**Test access**”



When you click the “Test access” button a new dialog box will open, showing a successful connection. Click OK.



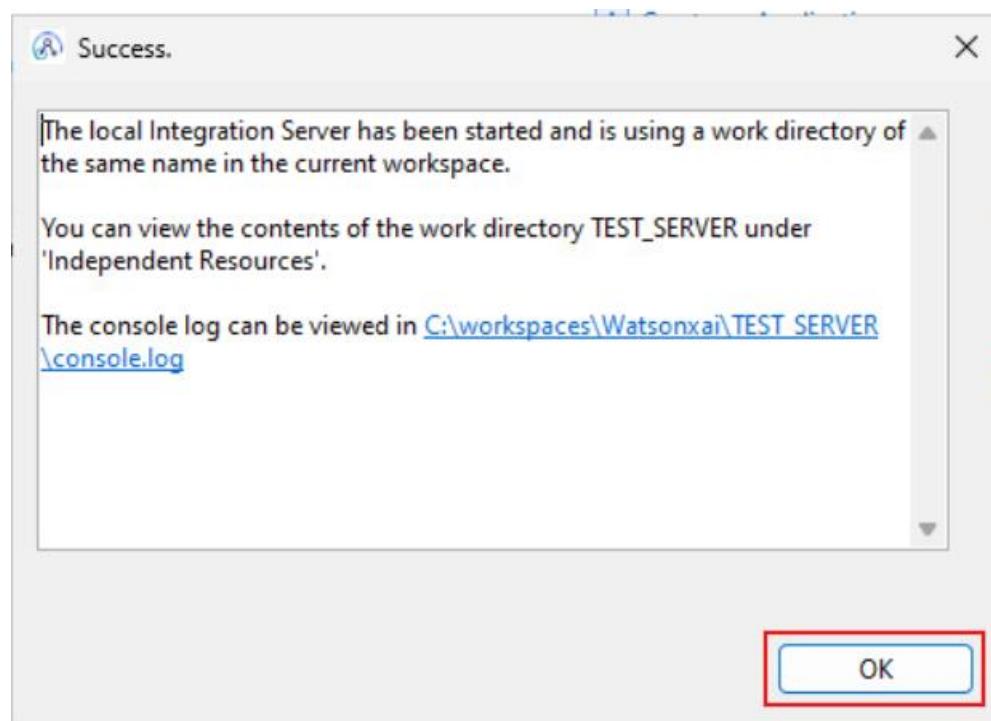
You will be returned to the “**Create and start a local Integration Server**” dialog, where you should click the **Finish** button:



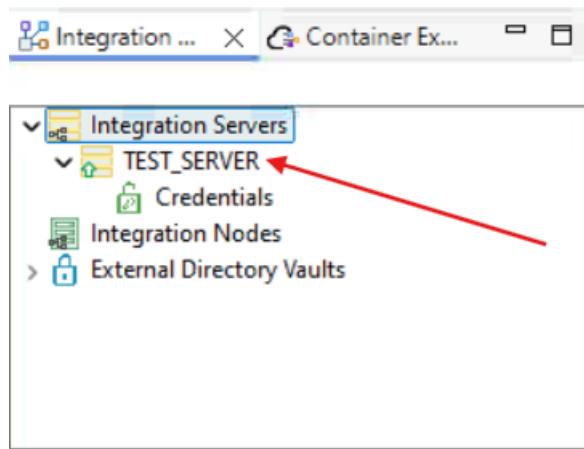
This will trigger a process to configure a local integration server (not managed by a node) and to start it.

Note: the default port for Administration of the server is **7600**. Since the “**Find a currently available port for REST administration**” is ticked, the process will start from 7600 and if not available, add one to this port number until an available port is found (*equivalent logic applies for the HTTP Port and JVM Debug Port*).

29. On successful start of the local Integration Server you will see a message similar to the picture below. Click the **OK** button to dismiss the window:



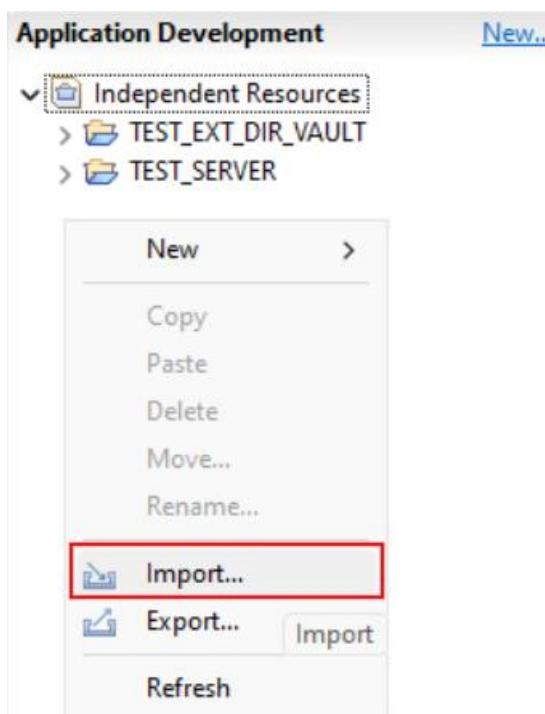
30. A connection to your new local Integration Server will appear in the **Integration Explorer** Window – the green arrow (pointing upwards) to the left of the server name indicates that the server is running:



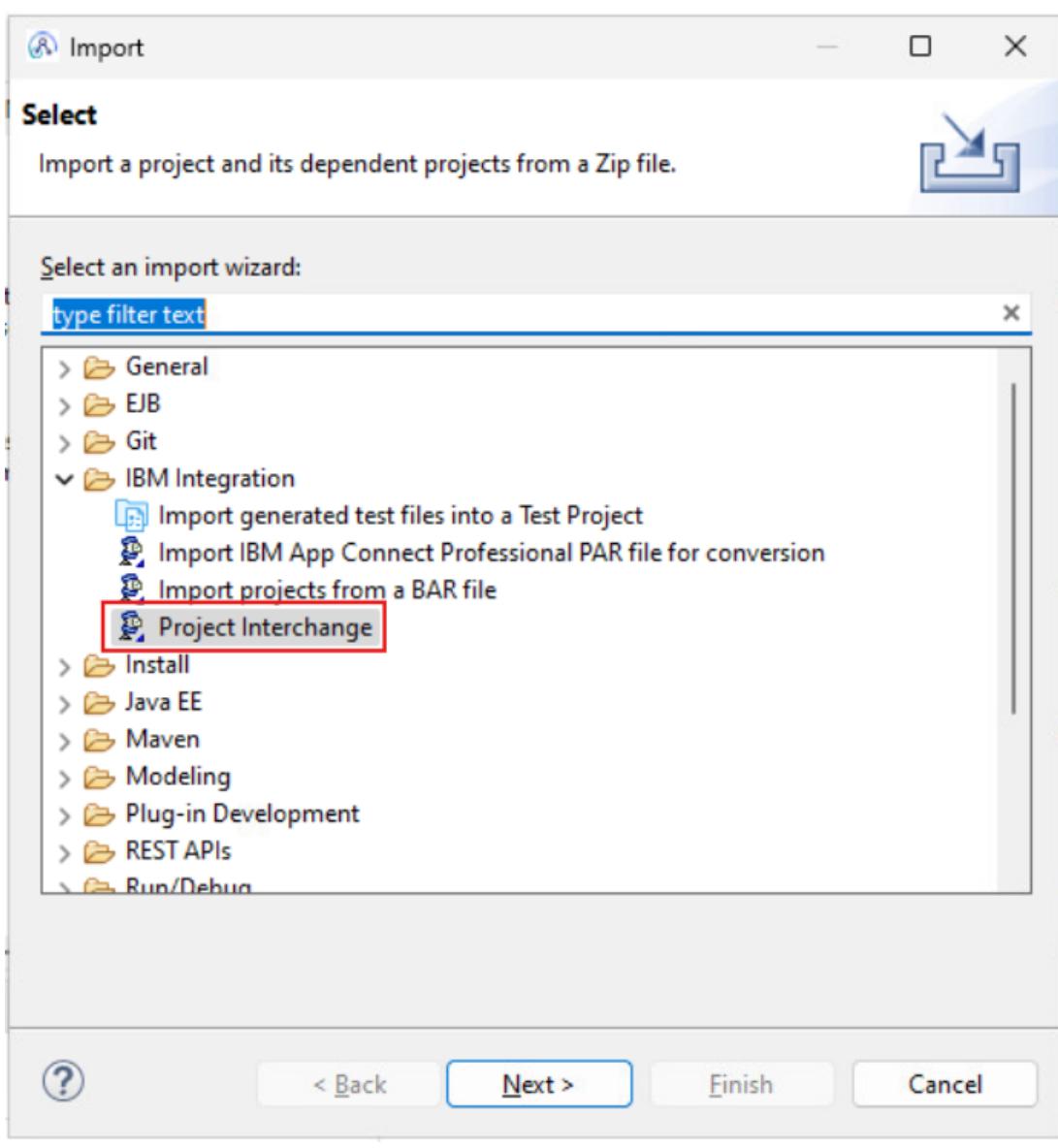
The local Integration Server should be started. If the server isn't showing as started, try right-clicking TEST_SERVER and refreshing. If this doesn't cause it to show as started, there is likely to be a problem with the default ports configured in your server.conf.yaml. If you cannot get the server started, seek help from your instructor.

3.1 Import the project interchange file

31. Right click on the background of the Application Development view and choose the menu option to Import:

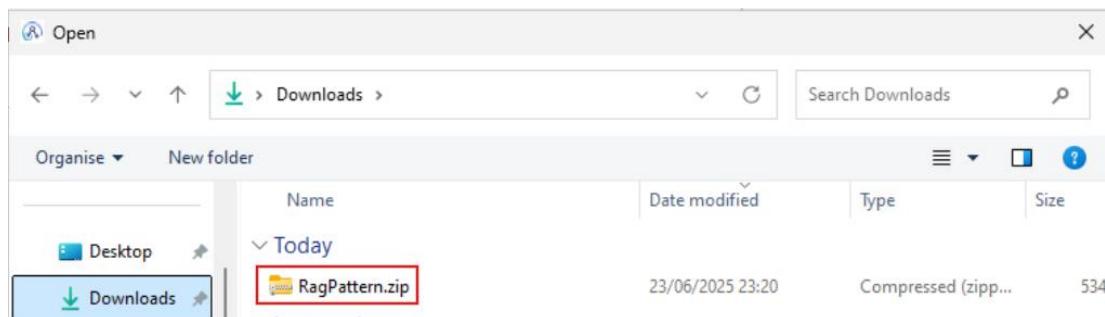


32. In the Import dialog, select the option **IBM Integration > Project Interchange**:



Click Next.

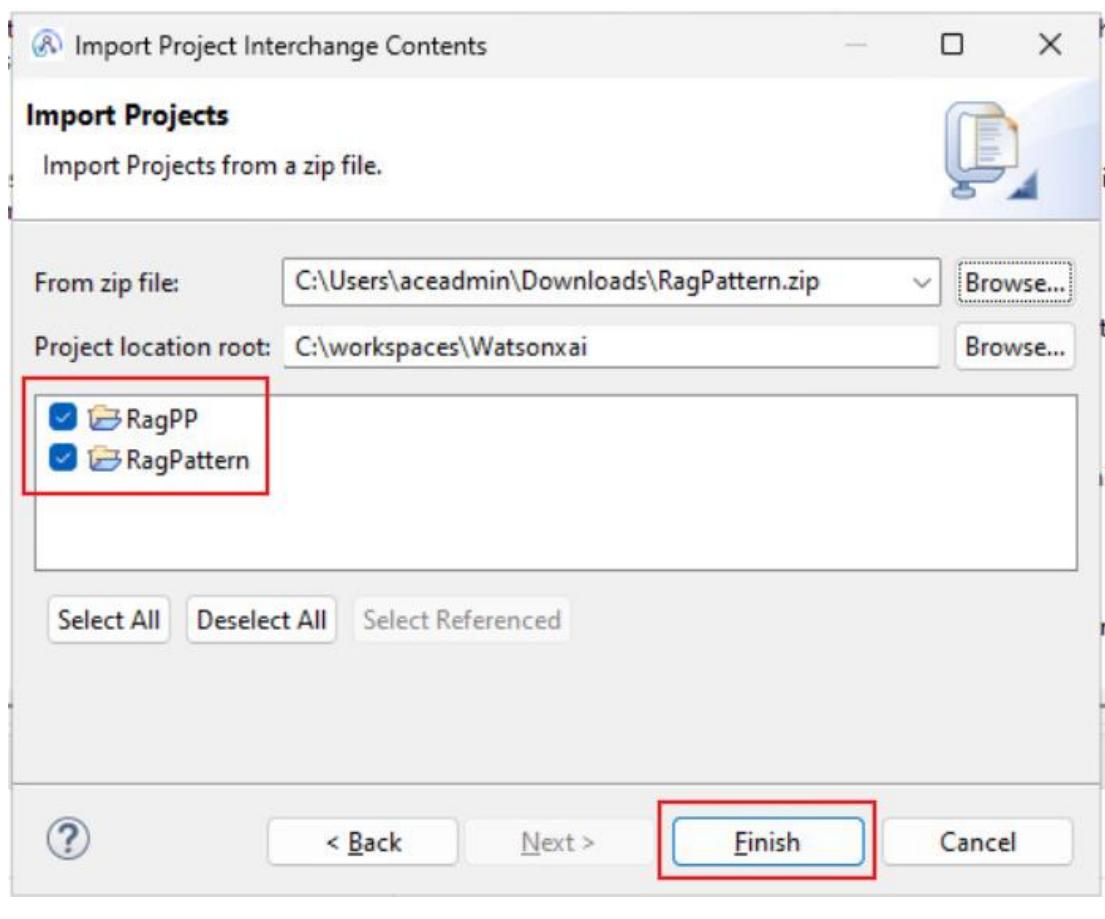
33. Navigate to the file **RagPattern.zip**, which will be provided to you over Slack (sorry ... unfortunately we were not able to get this file in to the VM image ahead of time!). In the picture below, the RagPattern.zip file is in the Windows download directory:



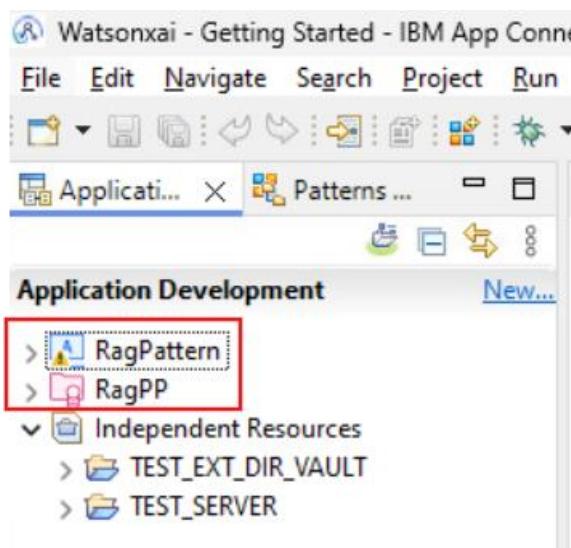
Having clicked on the file, click the Open button:



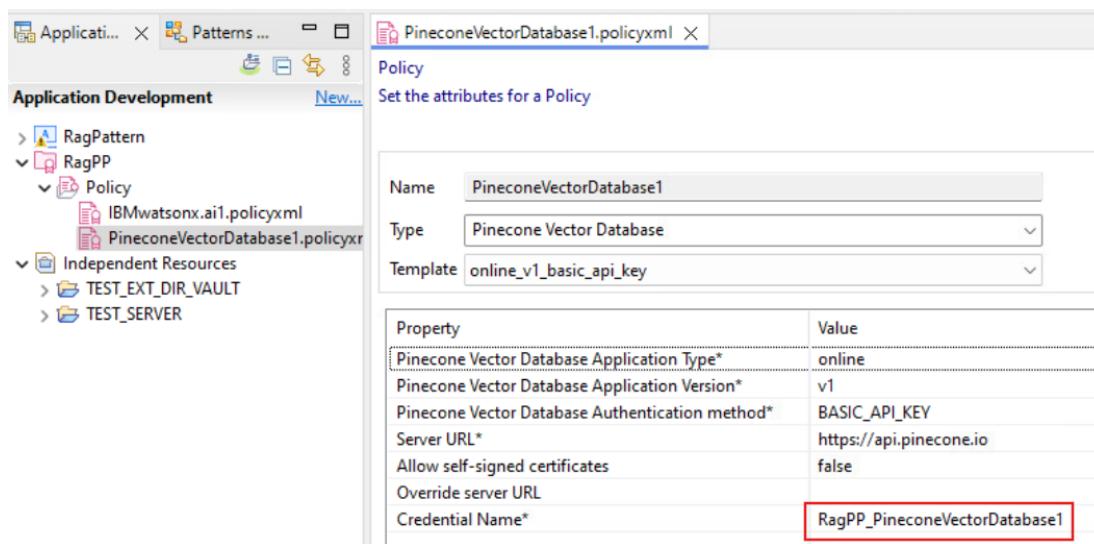
34. The Project Interchange file contents will be displayed. Select both projects and click Finish:



35. The projects will import and they will be displayed in the Application Development view as shown below. The project called **RagPattern** is an application project. The project called **RagPP** is a policy project:



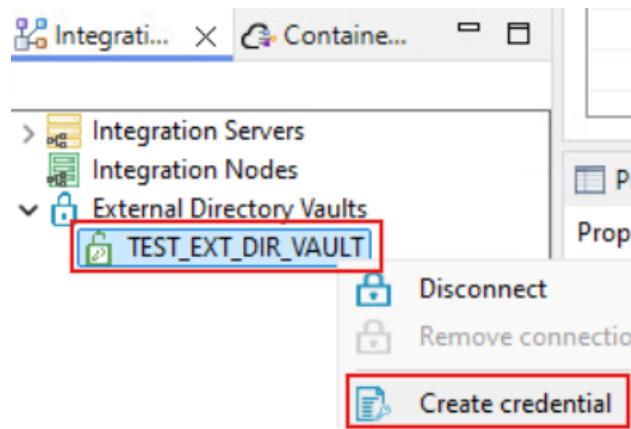
36. Expand the policy project and open the policy file named PineconeVectorDatabase1.policyxml. You will note that the policy has already been configured to reference a credential named **RagPP_PineconeVectorDatabase1**. In a moment we will create this credential:



37. In the same policy project, open the other policy which is provided named IBMwatsonx.ai1.policyxml. Note that this policy is configured to expect a credential named **RagPP_IBMwatsonx.ai1**:

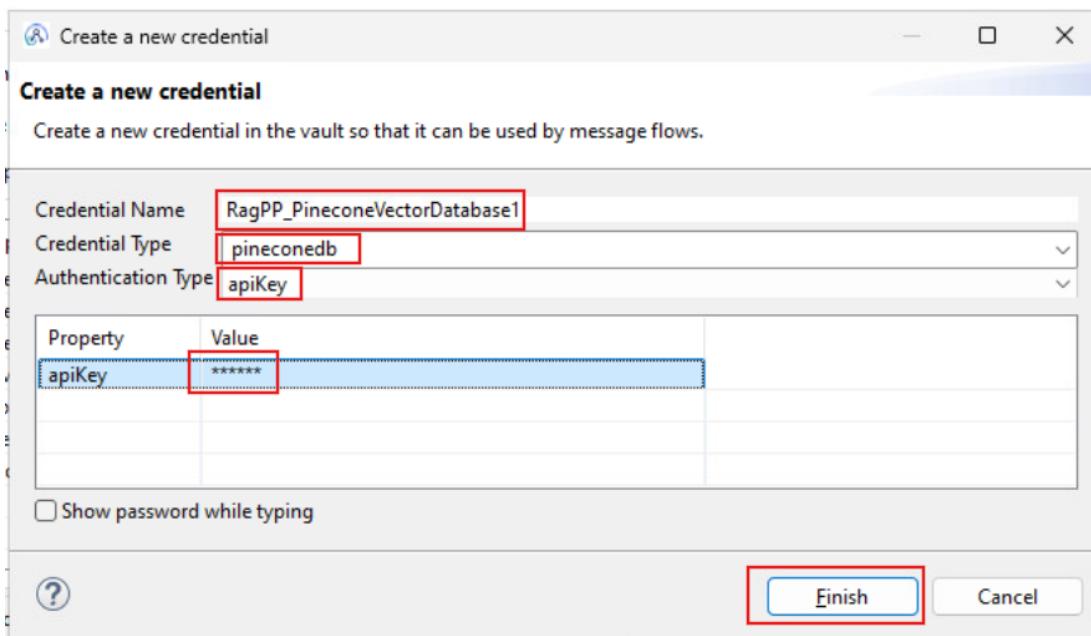
Name	IBMwatsonx.ai1
Type	IBM watsonx.ai
Template	online_v1_basic_api_key
Property	Value
IBM watsonx.ai Application Type*	online
IBM watsonx.ai Application Version*	v1
IBM watsonx.ai Authentication method*	BASIC_API_KEY
Server URL*	https://us-south.ml.cloud.ibm.com
Allow self-signed certificates	false
Override server URL	
Credential Name*	RagPP_IBMwatsonx.ai1

38. From the Application Development view in the bottom left corner of the Toolkit, right-click on the vault that you created earlier named **TEST_EXT_DIR_VAULT** and select the menu option to **Create credential**:

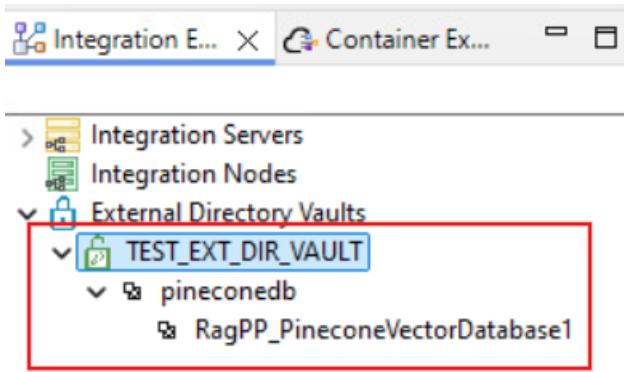


39. Create the Pinecone credential using the following values:
- **Credential Name** = RagPP_PineconeVectorDatabase1
 - **Credential Type** = pineconedb
 - **Authentication Type** = apiKey
 - **apiKey** = <value that you copied and stored earlier in the lab guide>

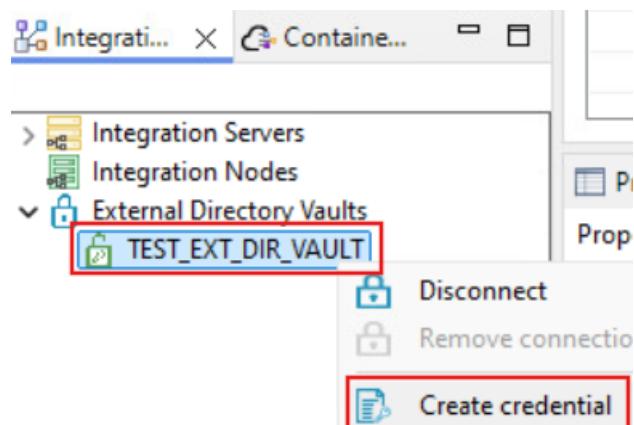
Click the **Finish** button:



40. The resulting credential will now be displayed in the TEST_EXT_DIR_VAULT hierarchy as shown below:

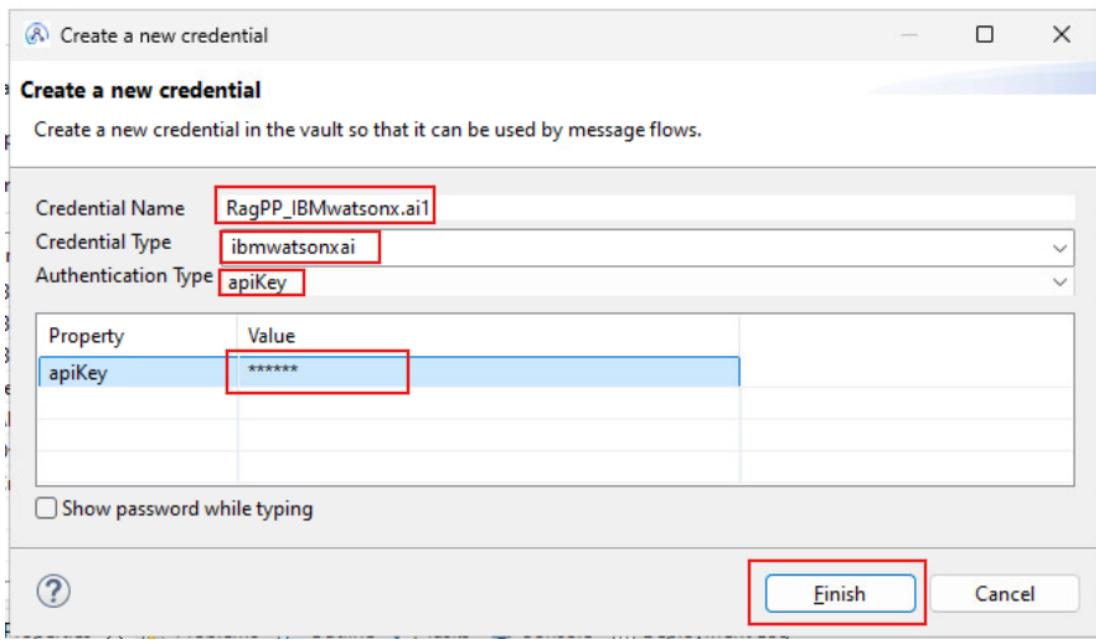


41. From the Application Development view in the bottom left corner of the Toolkit, we will repeat a similar process but create a credential for accessing watsonx.ai. Right-click on the vault **TEST_EXT_DIR_VAULT** and select the menu option to **Create credential**:

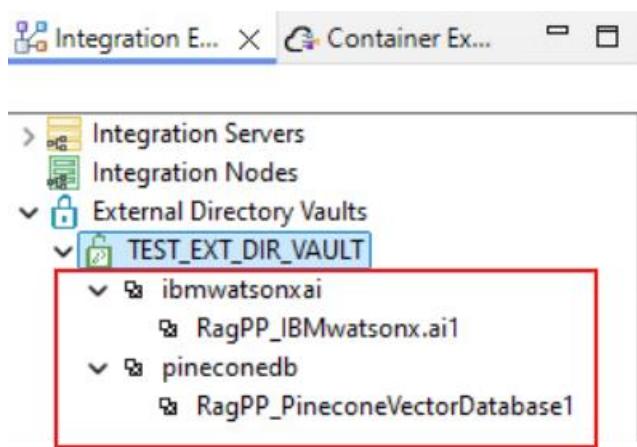


42. Create the watsonx.ai credential using the following values:
- **Credential Name** = RagPP_IBMwatsonx.ai1
 - **Credential Type** = ibmwatsonxai
 - **Authentication Type** = apiKey
 - **apiKey** = <value that you copied and stored earlier in the lab guide>

Click the **Finish** button:

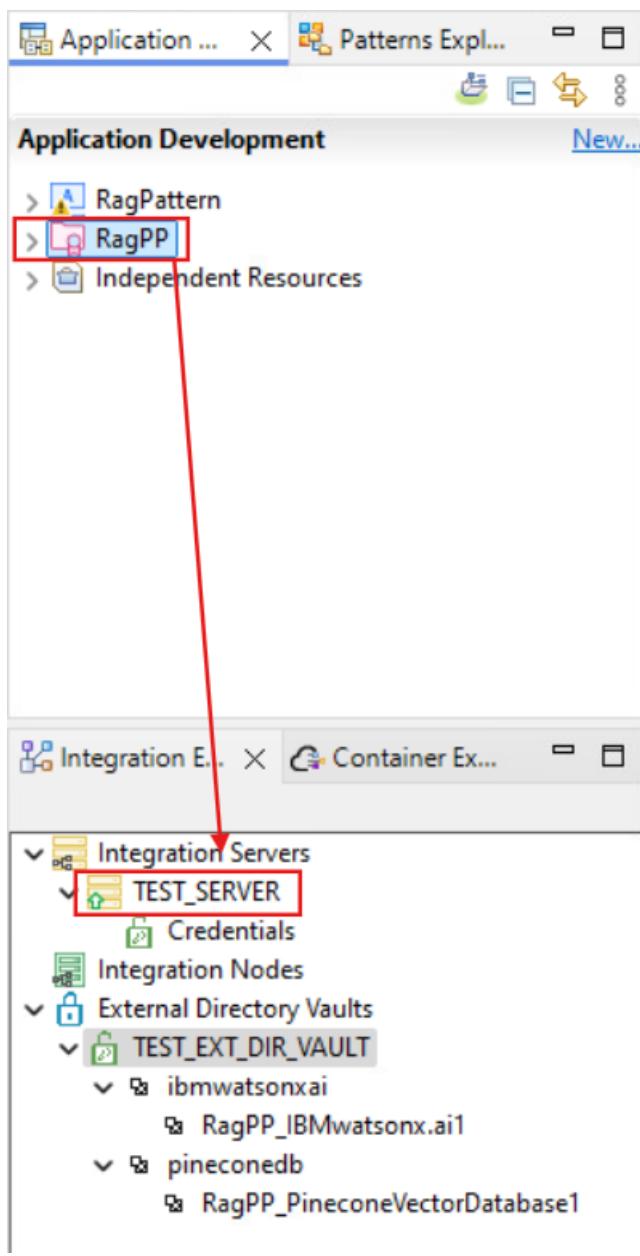


43. You will now see two credentials shown in the hierarchy under **TEST_EXT_DIR_VAULT**:

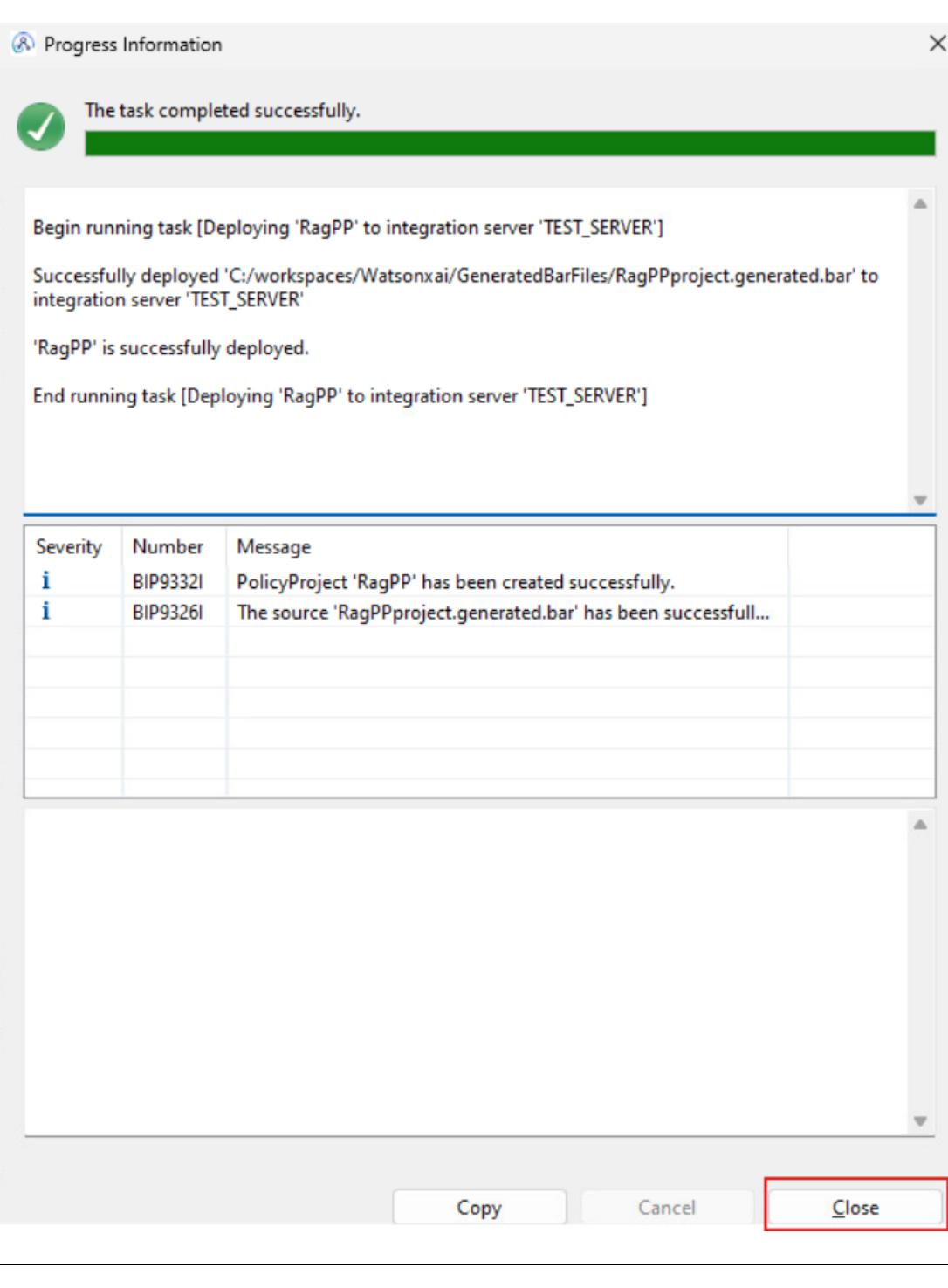


3.2 Deploy the Policy Project

44. Drag and drop deploy the policy project named RagPP on to the TEST_SERVER as shown below:



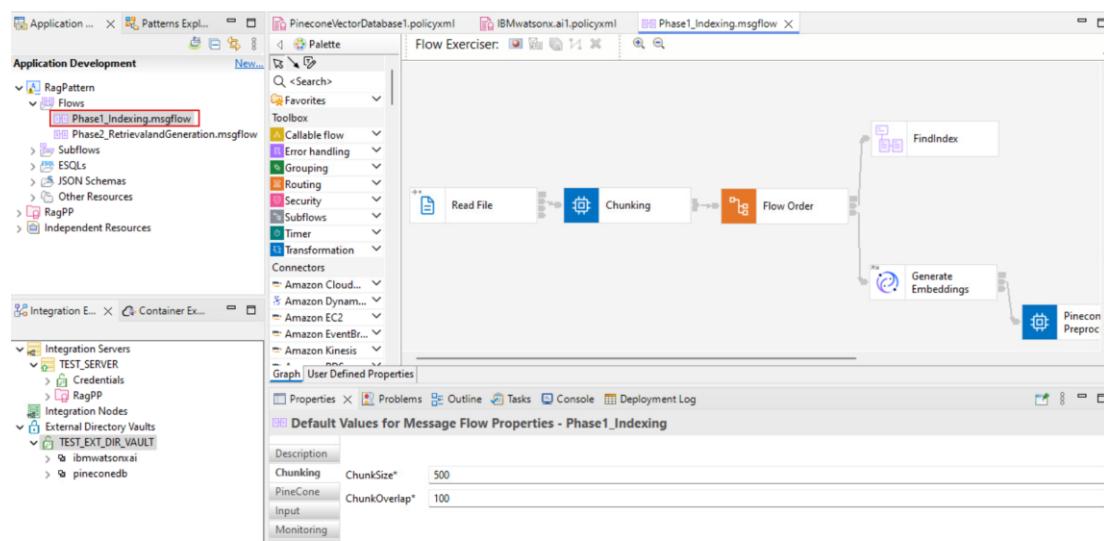
45. Click Close on the success confirmation window:



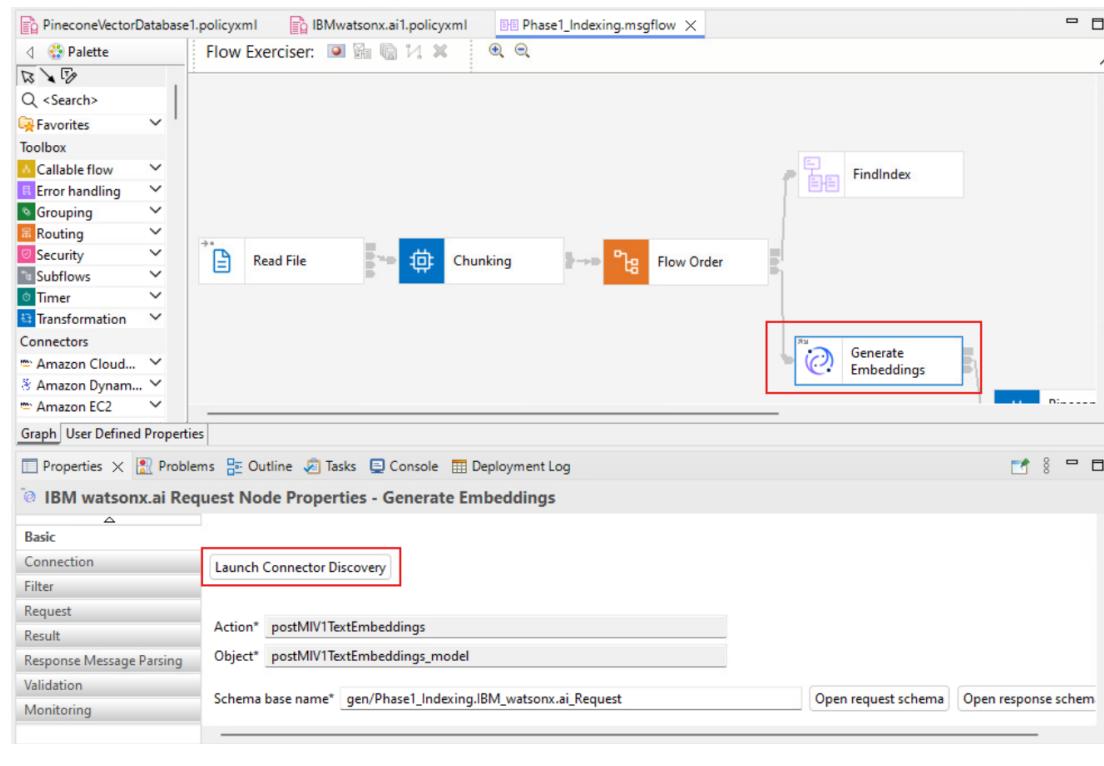
- 46.

3.3 Update the Phase1_Indexing message flow

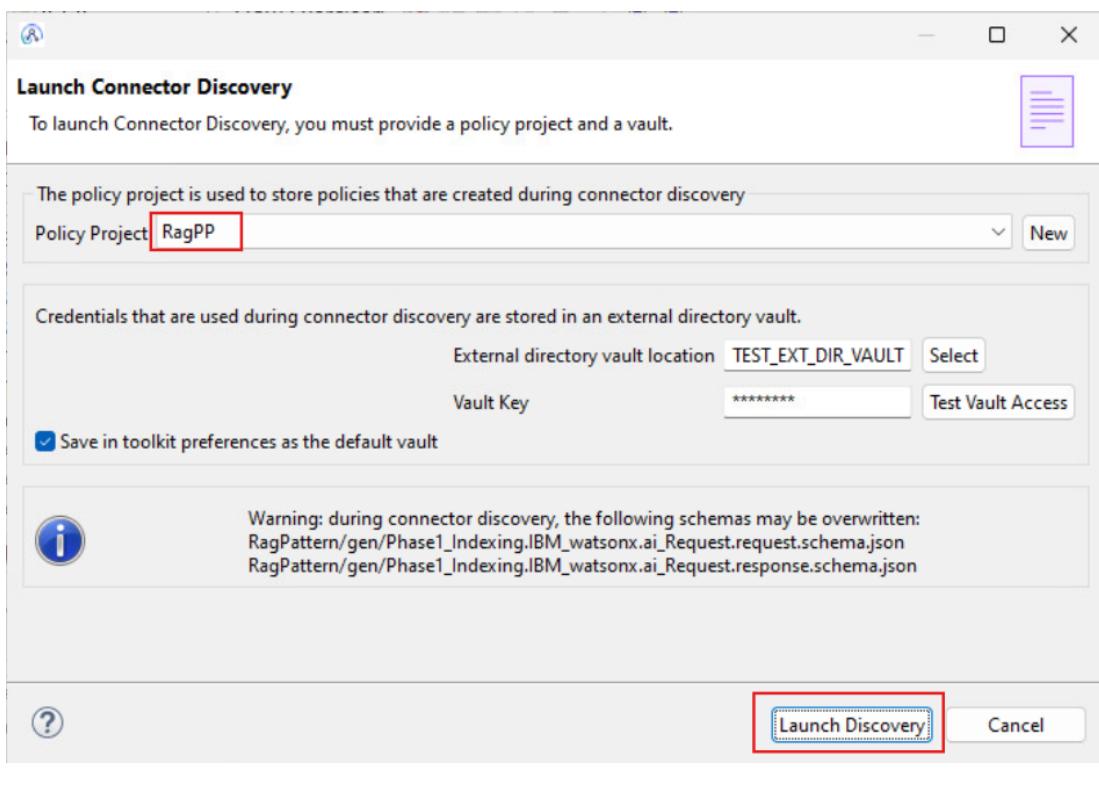
47. Expand the **RagPattern** application project and open the message flow named **Phase1_Indexing.msgflow**:



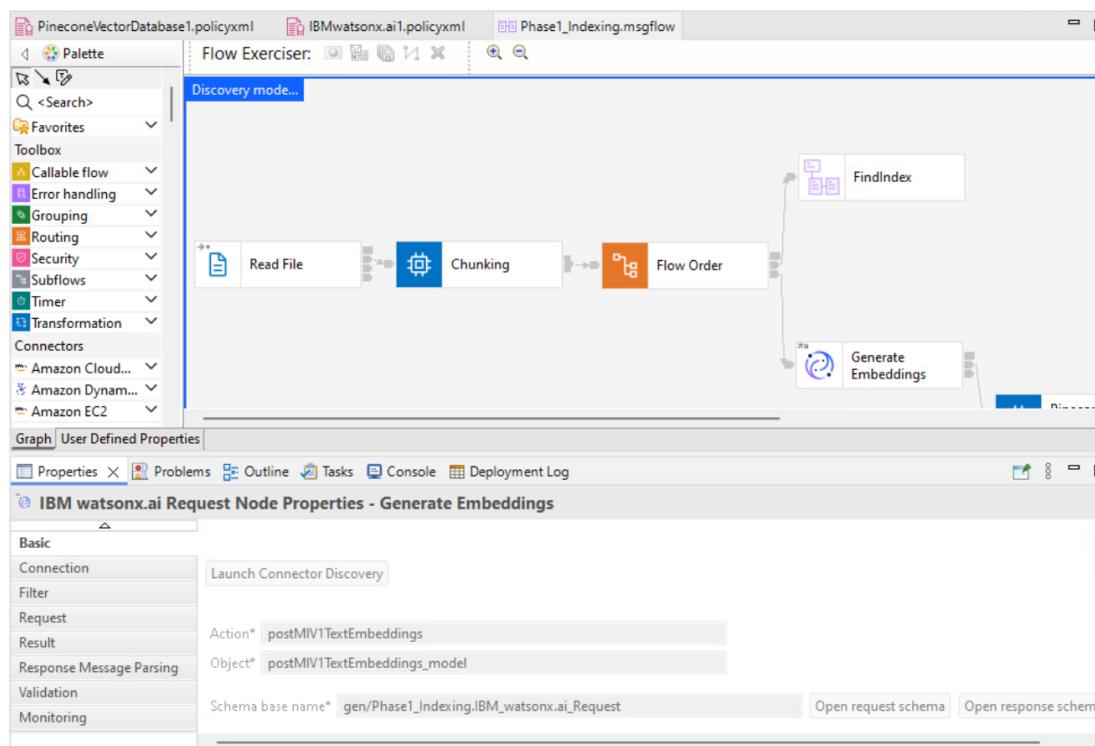
48. Click on the Watsonx.ai message flow node named **Generate Embeddings** and from the Properties panel click the button to **Launch Connector Discovery**:



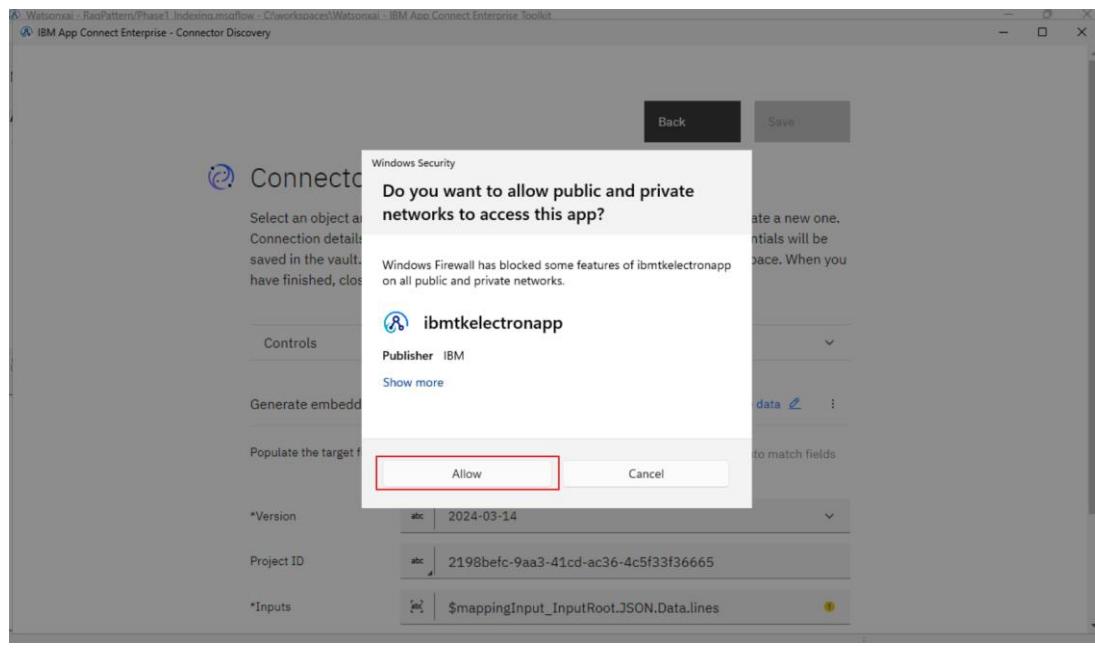
49. The Launch Connector Discovery wizard will open with the **RagPP Policy Project** already selected. Click the **Launch Discovery** button:



50. At this stage you will notice the message flow will receive a blue border around it due to entering Discovery mode. The properties of the node will be greyed out at the base of the screen. You will need to wait for several seconds at this stage for the next dialog to launch, so please be patient:



After a few seconds the screen may look like this ... click the Allow button if you are given this message (depending on what other labs you have run, you may not see this Windows Security warning):



51. The Connector Discovery window will have several values already pre-populated. You will need to update the Project ID with your own Project ID that you saved away in the previous steps in this Lab.

🔗 Connector Discovery

Select an object and action for **IBM watsonx.ai**. Use an existing connection or create a new one. Connection details will be saved in a policy inside policy project **RagPP** and credentials will be saved in the vault. Click Save to update the App Connect Enterprise Toolkit workspace. When you have finished, close the window.

Controls

Generate embeddings Sample data ⋮

Populate the target fields in IBM watsonx.ai ⓘ Auto match fields

*Version	abc 2024-03-14
Project ID	abc 2198befc-9aa3-41cd-ac36-4c5f33f36665
*Inputs	abc \$mappingInput_InputRoot.JSON.Data.lines !
<small>Warning: Must resolve to an array. No match for expression found in source data.</small>	
*Model ID	abc intfloat/multilingual-e5-large

52. Update the Project ID and then scroll back to the top of the page and click the Save button:



Connector Discovery

Select an object and action for **IBM watsonx.ai**. Use an existing connection or create a new one. Connection details will be saved in a policy inside policy project **RagPP** and credentials will be saved in the vault. Click Save to update the App Connect Enterprise Toolkit workspace. When you have finished, close the window.

Controls

Generate embeddings

Sample data [🔗](#)

:

Populate the target fields in IBM watsonx.ai [ⓘ](#)

Auto match fields

*Version

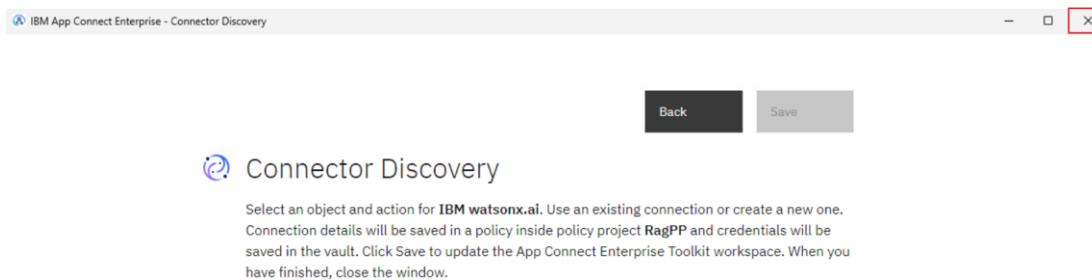
2024-03-14

Project ID

9e9c9a68-3381-48d4-ae0a-aec0d139b9b4

[ⓘ](#) 9e9c9a68-3381-48d4-ae0a-aec0d139b9b4

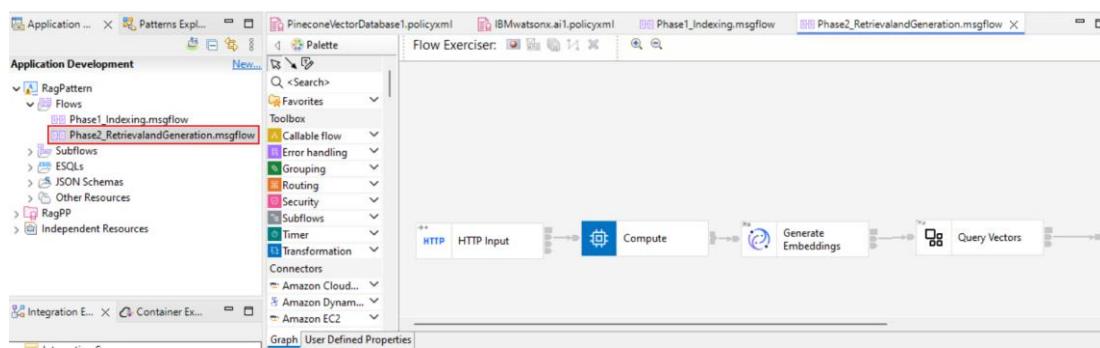
53. Close the window using the cross icon in the top right corner:



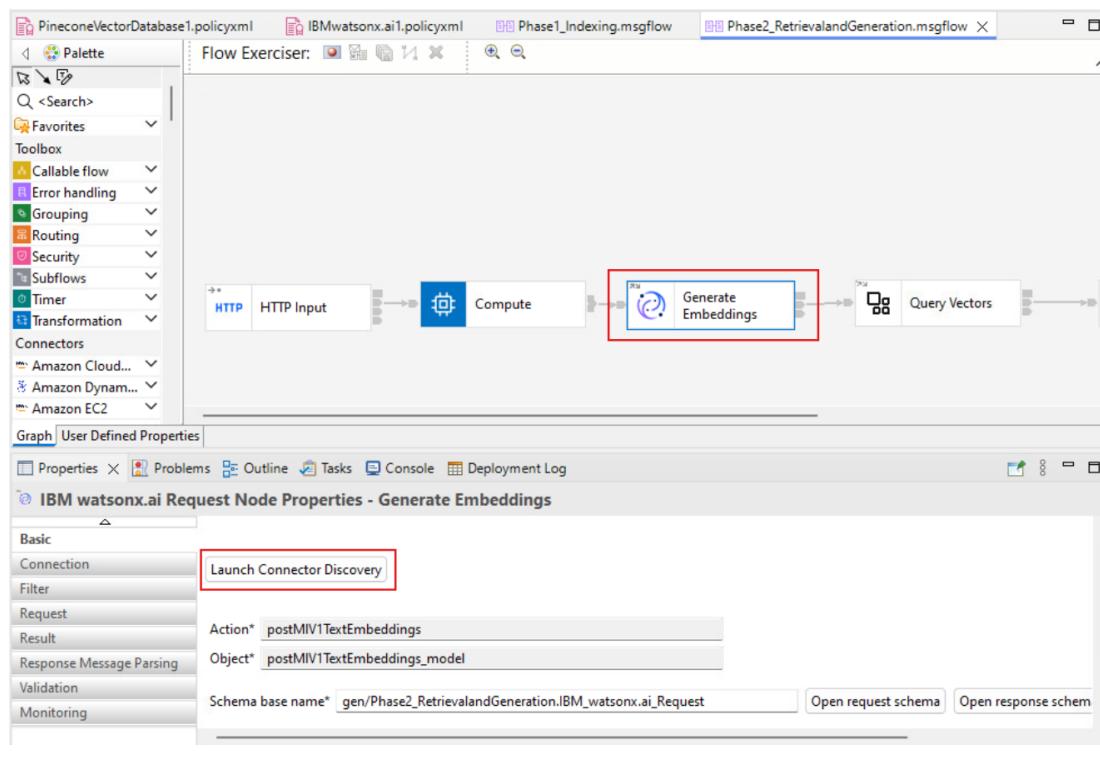
54. **Remember to now save the message flow using Ctrl-S**

3.4 Update the Phase2_RetrievalAndGeneration message flow

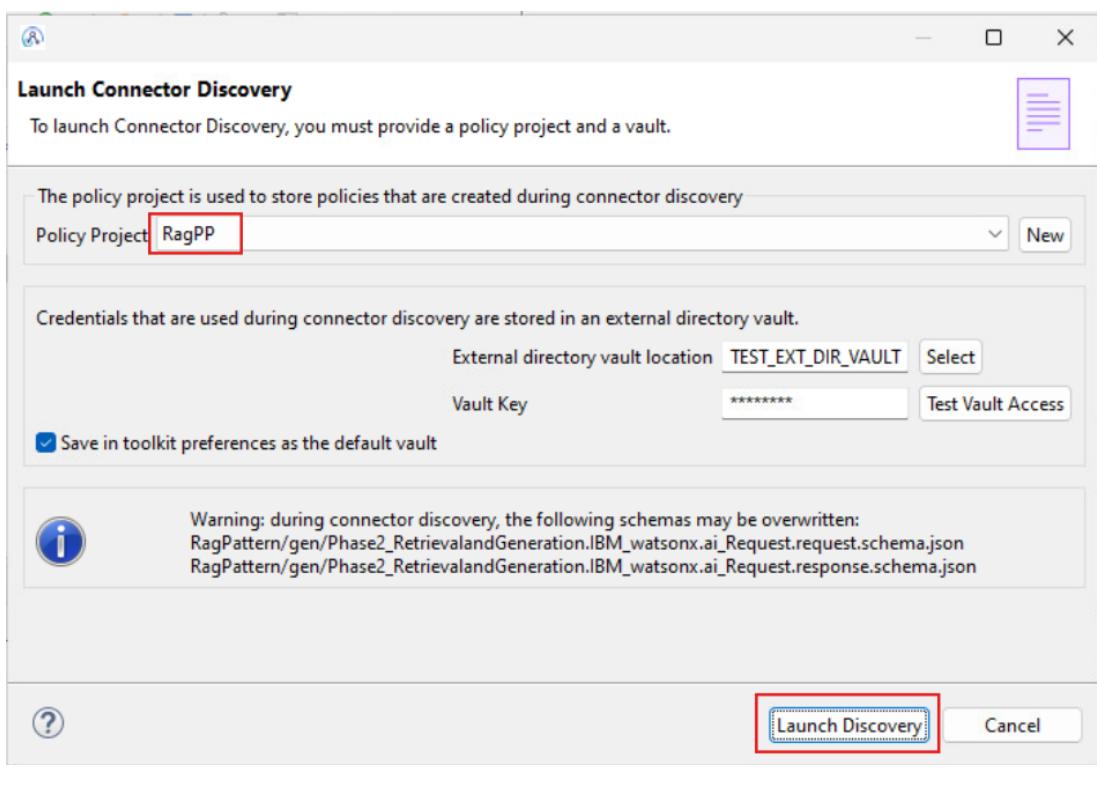
55. Expand the **RagPattern** application project and open the message flow named **Phase2_RetrievalandGeneration.msgflow**:



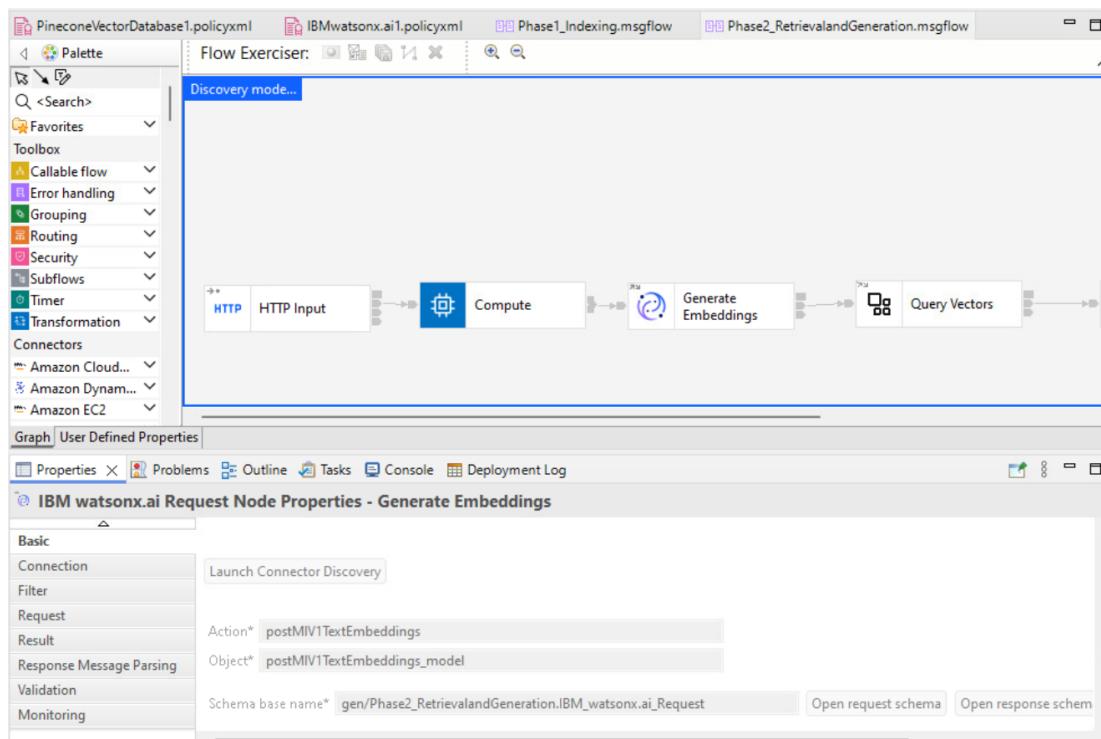
56. You will find that this message flow also has a Watsonx.ai message flow node named **Generate Embeddings**. Just like before, select this node and from the Properties panel click the button to **Launch Connector Discovery**:



57. The Launch Connector Discovery wizard will open with the **RagPP Policy Project** already selected. Click the **Launch Discovery** button:



58. At this stage you will notice the message flow will receive a blue border around it due to entering Discovery mode. The properties of the node will be greyed out at the base of the screen. You will need to wait for several seconds at this stage for the next dialog to launch, so please be patient:



After a few seconds the Connector Discovery page will launch. Just like before, you will need to update the Project ID field:

Back Save

Connector Discovery

Select an object and action for IBM watsonx.ai. Use an existing connection or create a new one. Connection details will be saved in a policy inside policy project RagPP and credentials will be saved in the vault. Click Save to update the App Connect Enterprise Toolkit workspace. When you have finished, close the window.

Controls

Generate embeddings Sample data [🔗](#) :

Populate the target fields in IBM watsonx.ai [ⓘ](#) Auto match fields

*Version	abc 2024-03-14	v
Project ID	abc f5c4b4f1-6c27-4cad-bcc3-8b56df83f335	
*Inputs	abc [\$mappingInput_Environment.Variable.query]	
*Model ID	abc intfloat/multilingual-e5-large	

59. The Connector Discovery window will have several values already pre-populated. You will need to update the Project ID with your own Project ID that you saved away in the previous steps in this Lab. Update the Project ID and then scroll back to the top of the page and click the Save button:



Connector Discovery

Select an object and action for **IBM watsonx.ai**. Use an existing connection or create a new one. Connection details will be saved in a policy inside policy project **RagPP** and credentials will be saved in the vault. Click Save to update the App Connect Enterprise Toolkit workspace. When you have finished, close the window.

Controls

Generate embeddings

Sample data :

Populate the target fields in IBM watsonx.ai

Auto match fields

*Version

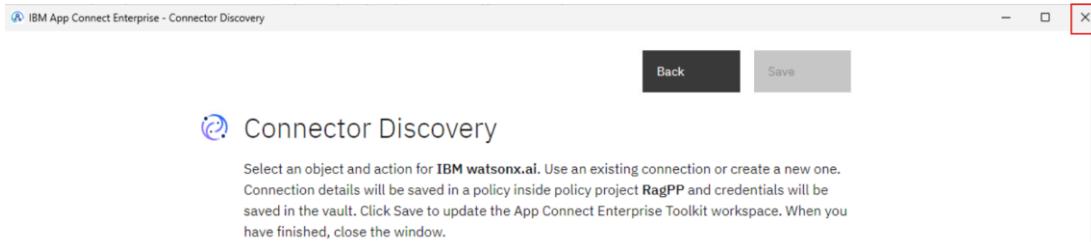
abc | 2024-03-14

Project ID

abc | 9e9c9a68-3381-48d4-ae0a-aec0d139b9b4

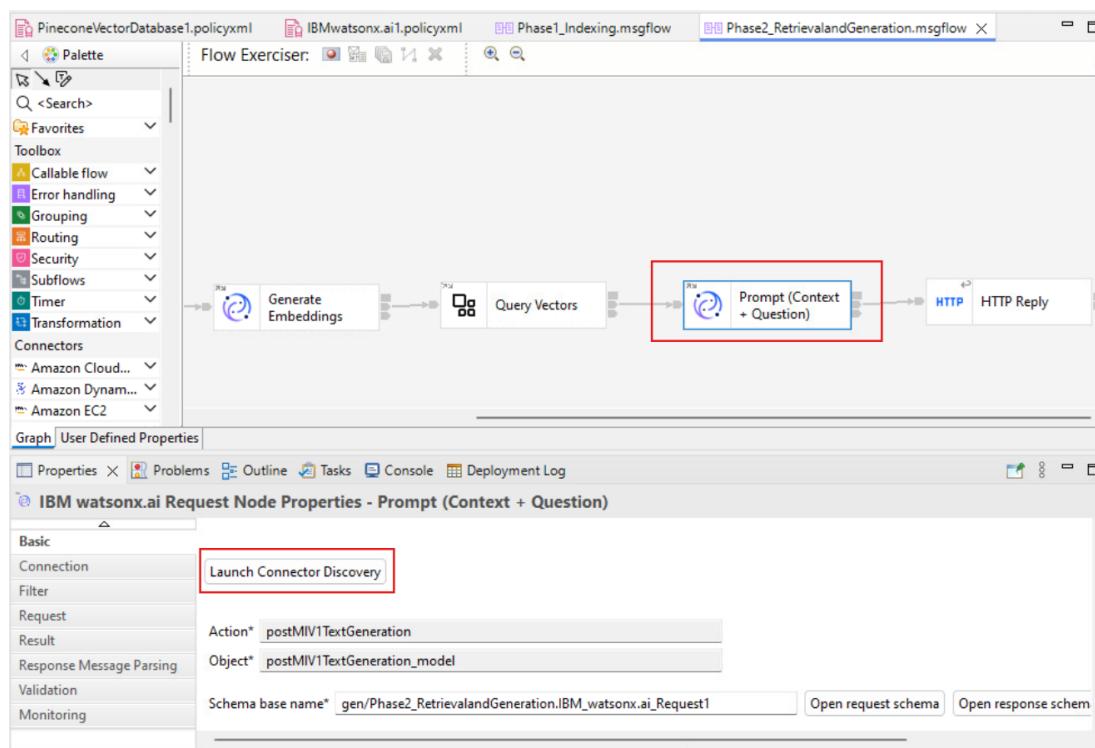
9e9c9a68-3381-48d4-ae0a-aec0d139b9b4

60. Close the window using the cross icon in the top right corner:

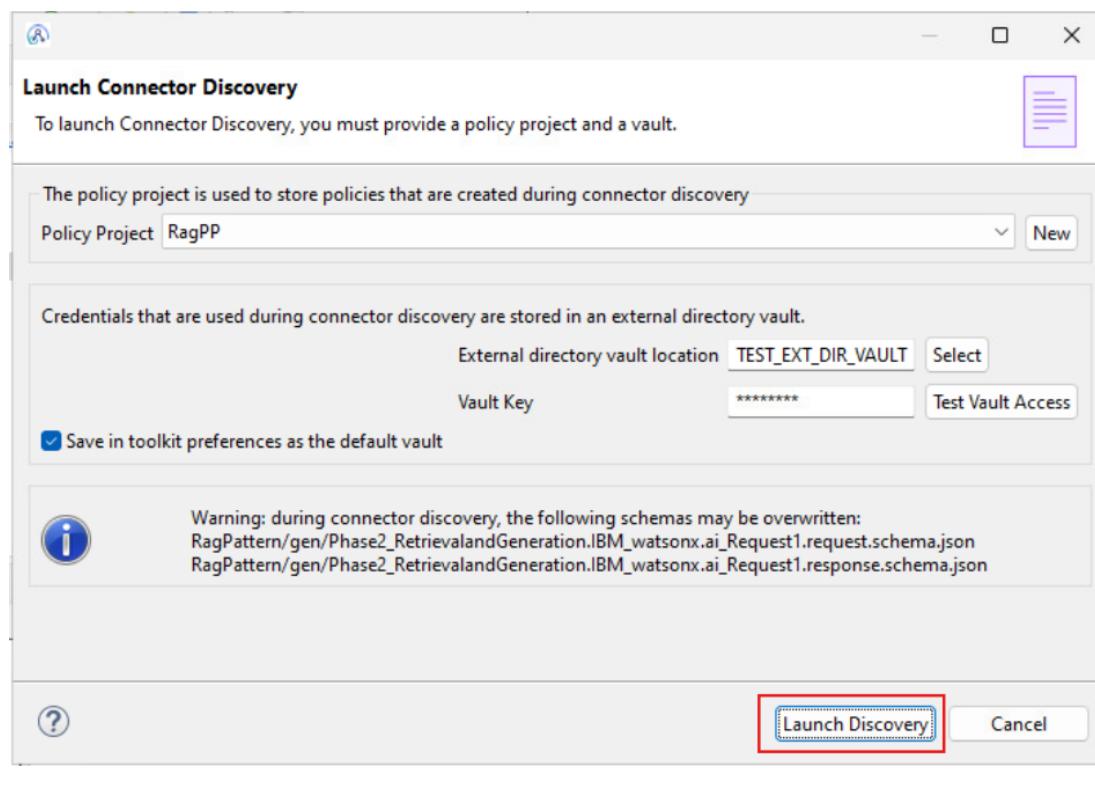


61. **Remember to now save the message flow using Ctrl-S**

62. Back in the message flow view, scroll to the right of the message flow and you will find there is a second watsonx.ai message flow node in the message flow named **Prompt (Context + Question)**. Select this node and then click the **Launch Connector Discovery** button:



63. The Launch Connector Discovery dialog will open just like before. Click **Launch Discovery**:



64. **This time, we must also make a second change...** Click the menu item shown and select the option to **Switch to advanced mode**:

The screenshot shows the 'Generate text' interface. At the top right are 'Sample data' and a three-dot menu. A red box highlights the 'Switch to advanced mode' button in the dropdown menu. A tooltip explains: 'Advanced mode enables you to dynamically populate the target destination and enter values into hidden fields'. Below the menu, there are four input fields: '*Version' (2024-03-14), 'Project ID' (9e9c9a68-3381-48d4-ae0a-aec0d139b9b4), '*Input' (You are an enterprise assistant that strictly uses the following context to answer questions. Only provide), and '*Model ID' (mistralai/mistral-large).

65. Scroll down the dialog, and update the Minimum and Maximum response token fields to each take the value of 200:

The screenshot shows the 'Parameters' section of the dialog. It includes fields for 'Space ID' (empty), 'Parameters' (with a grid icon and fx button), 'Minimum responses...' (set to 200, highlighted with a red box), and 'Maximum responses...' (set to 200). A note below says: 'If stop sequences are given, they are ignored until minimum tokens are generated. Default: 0.' Other parameters shown are 'Random seed' (123) and 'Decoding method' (abc).

66. Then, scroll back to the top of the window and click the Save button:

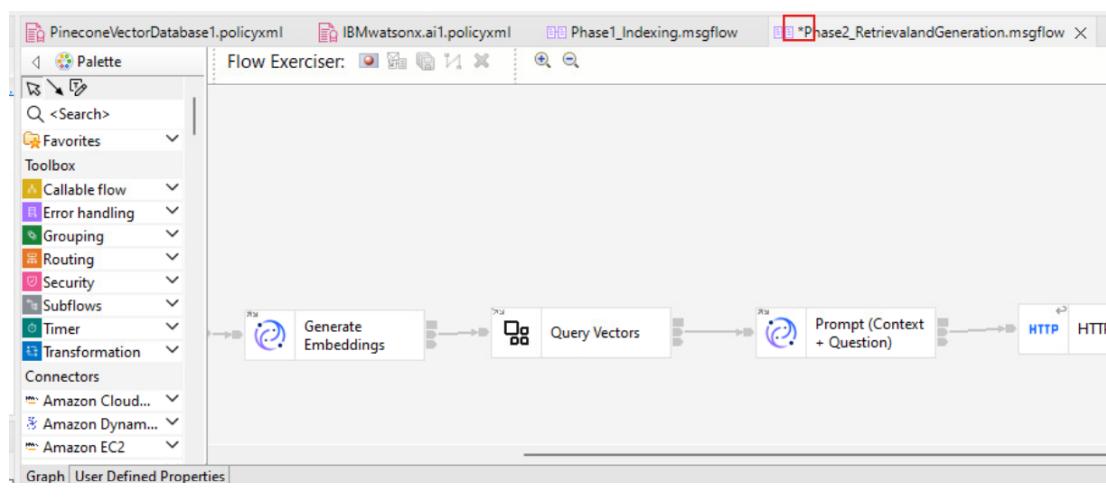


Connector Discovery

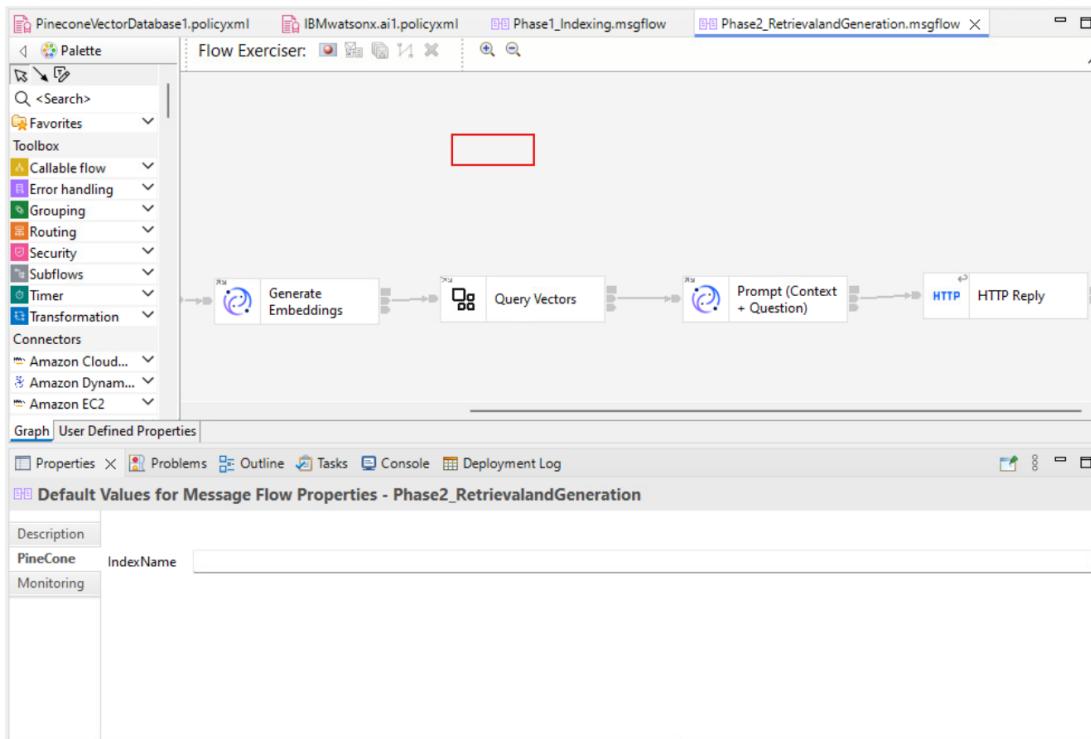
Select an object and action for **IBM watsonx.ai**. Use an existing connection or create a new one. Connection details will be saved in a policy inside policy project **RagPP** and credentials will be saved in the vault. Click **Save** to update the App Connect Enterprise Toolkit workspace. When you have finished, close the window.

.. and like before, once saved, click the cross icon to close the dialog.

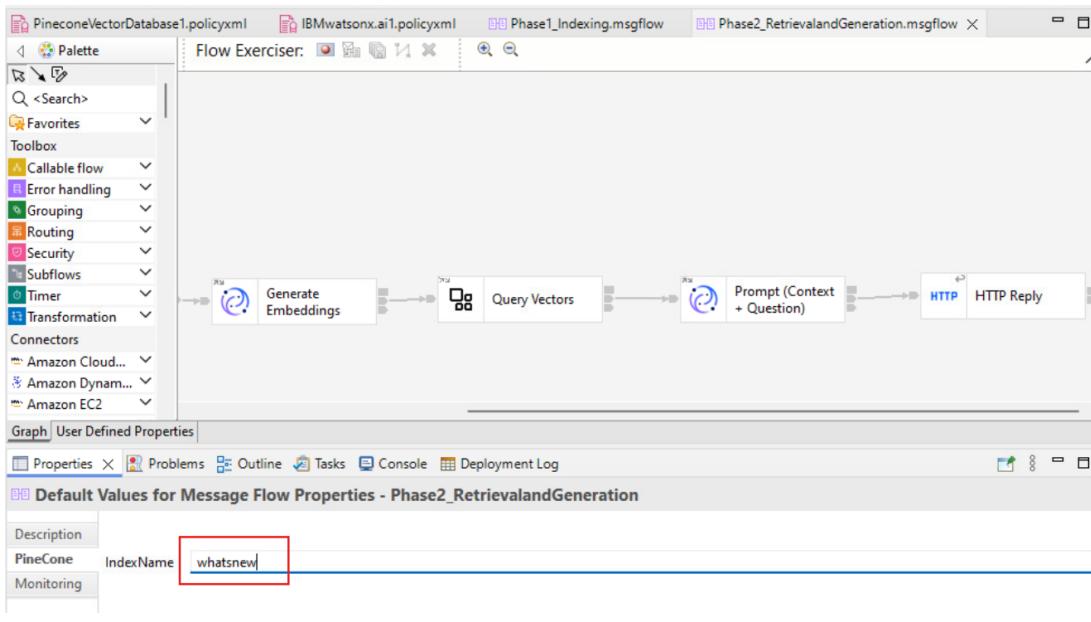
67. Back in the flow view, you will see that the message flow is marked as “dirty” (ie it has been updated but not yet saved). **Remember to save the flow at this stage using Ctrl-S.**



68. Next, click anywhere on the empty canvas of the message flow (as symbolised by the red box shown below ... basically anywhere on the grey background of the flow!) and the panel at the base will be updated to show you some User Defined Properties of the flow. In particular you will find on the PineCone tab, an IndexName field which is currently blank:



69. Update the IndexName with the value whatsnew:



Save the message flow again (Ctrl-S).

70. The message flow expects to read an input file from the directory **C:\temp\RagInput** ... you will need to create this directory on the file system. Open Windows Explorer and create the directory **C:\temp** and then a child directory named **RagInput**

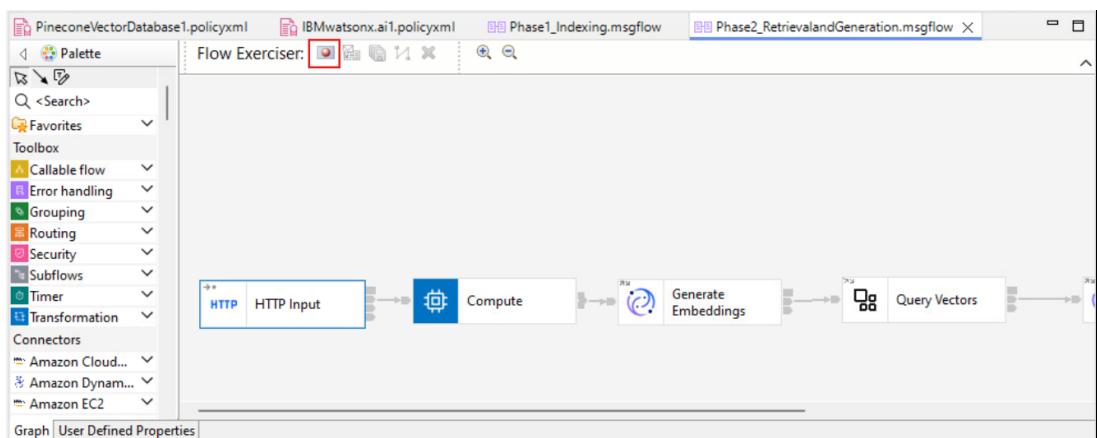


71.

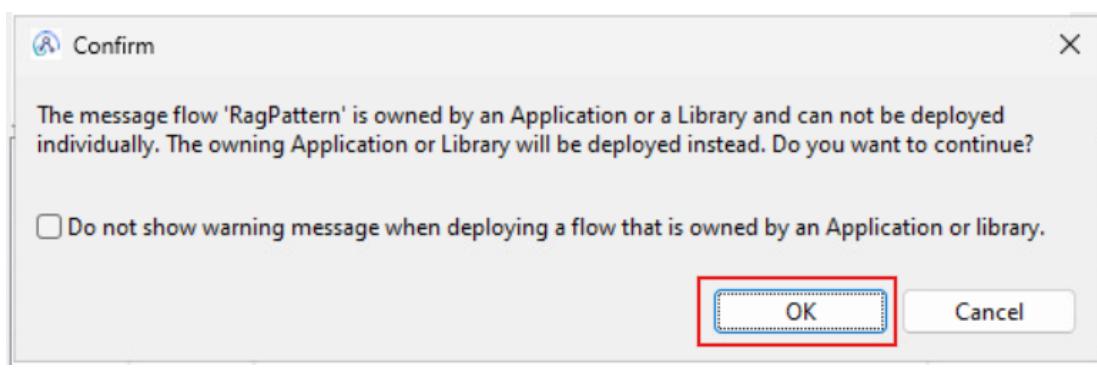
72.

3.5 Test the message flows

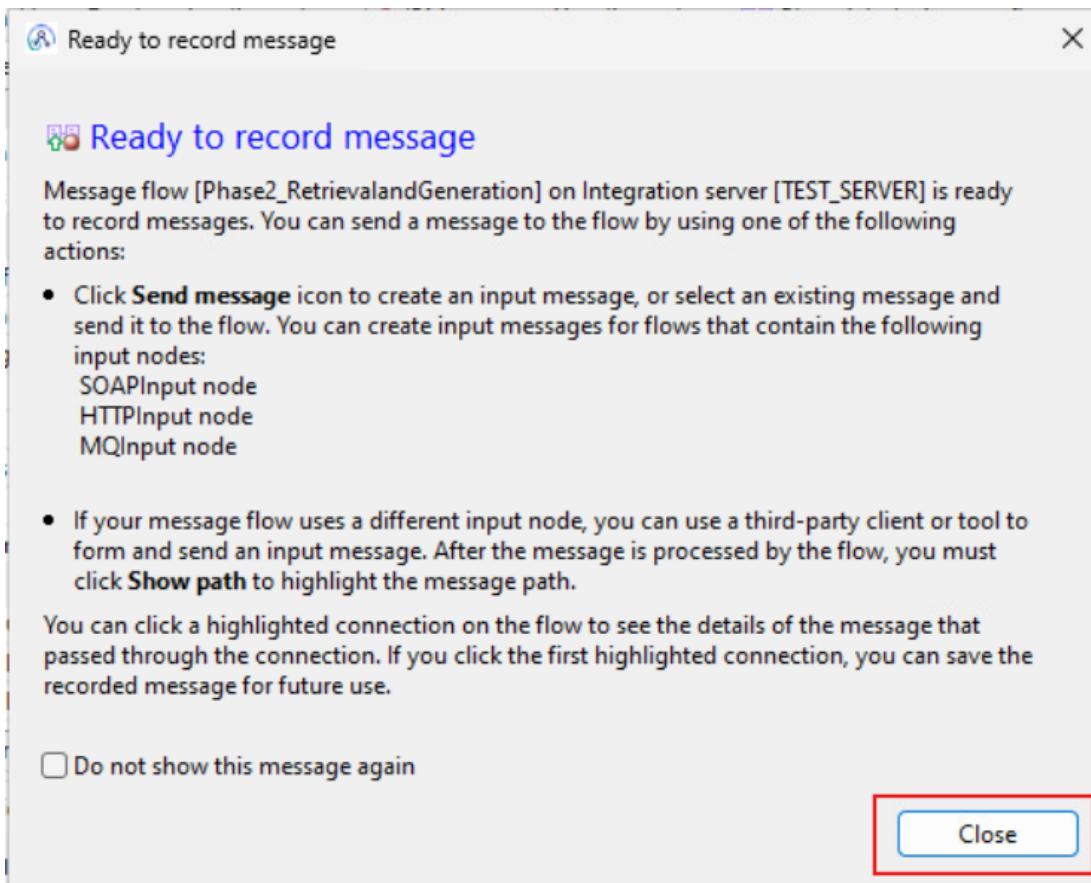
73. Return to the message flow named **Phase2_RetrievalandGeneration.msgflow** and click the Record button in the Flow Exerciser toolbar:



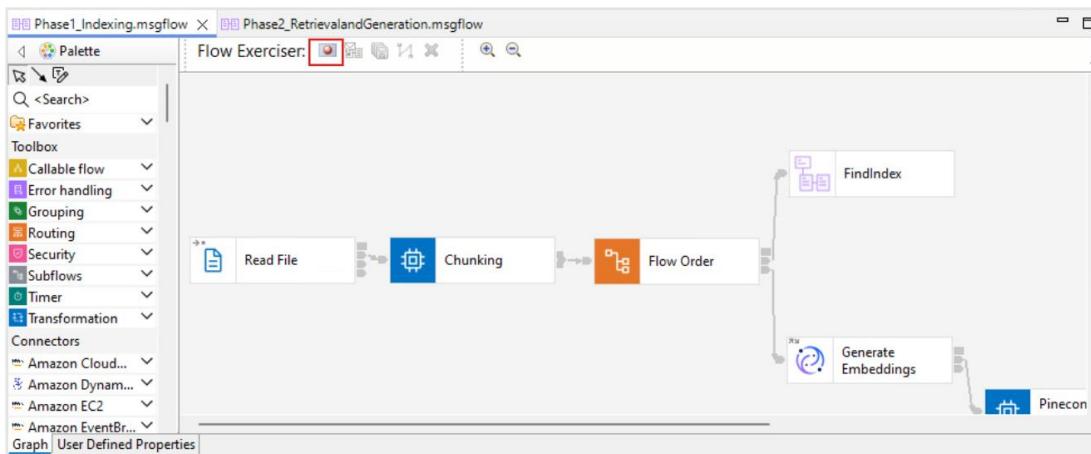
Click **OK** on the Confirm dialog:



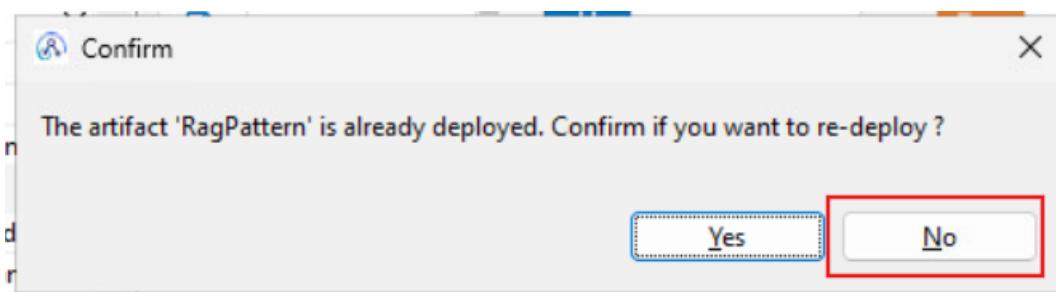
74. Click the Close button on the Ready to record message:



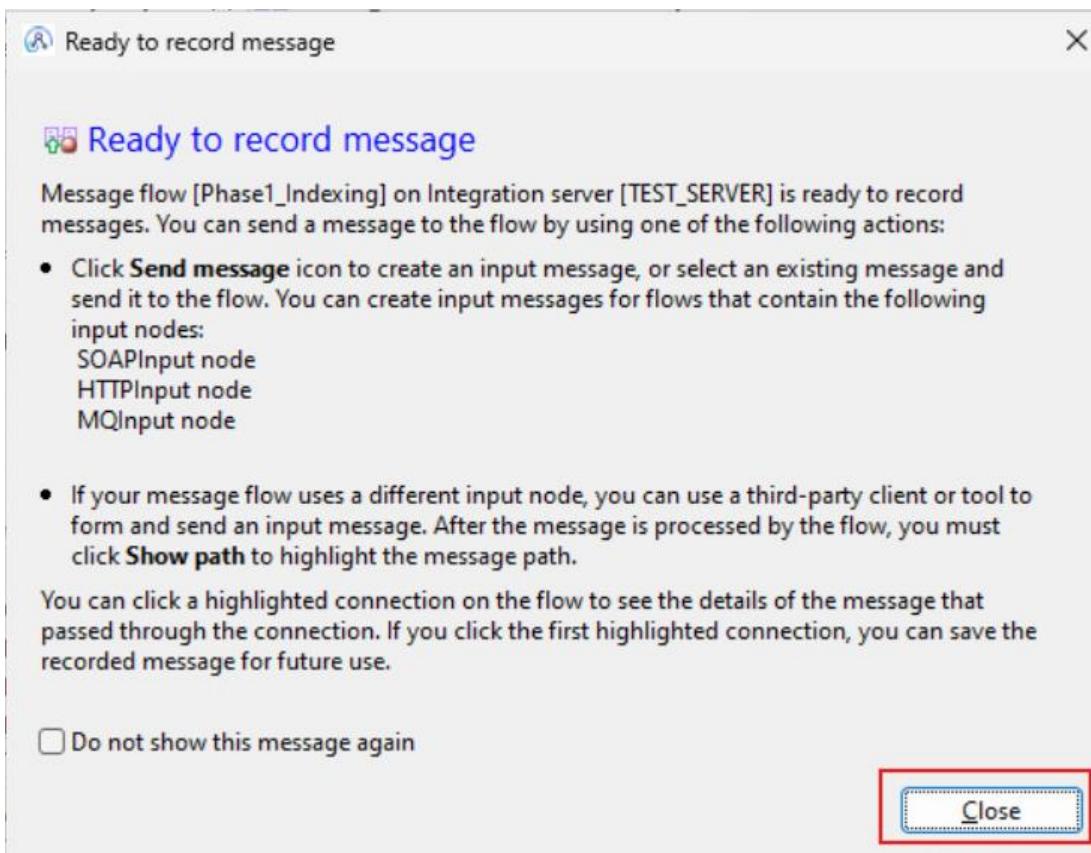
75. Go back to Phase1_Indexing.msgflow and click the Record icon in the taskbar of that flow as well:



76. You will receive a confirmation about whether you wish to redeploy ... you can click the No button:



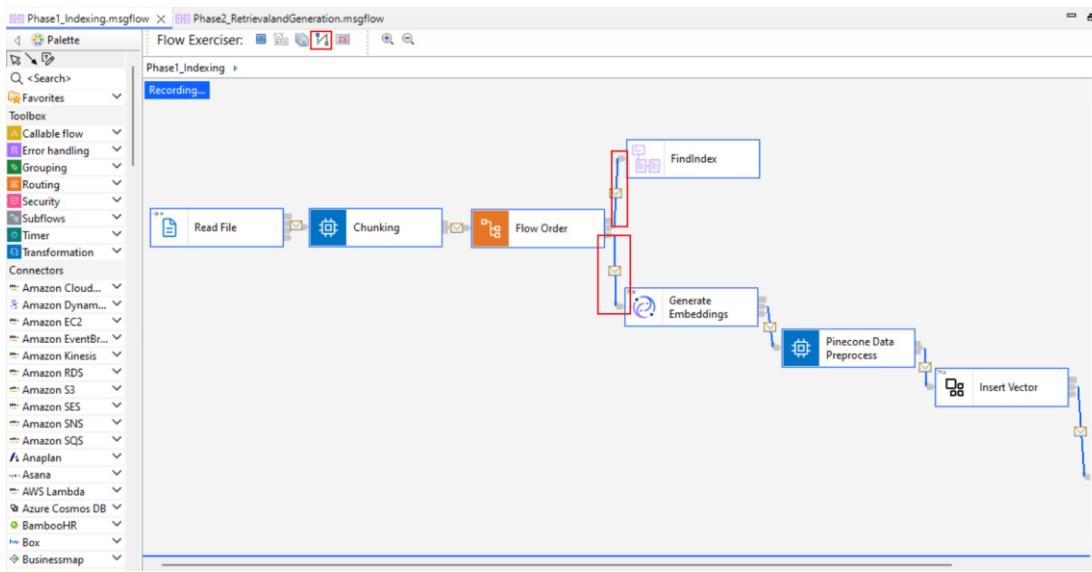
77. Click the Close button on the Ready to record message dialog:



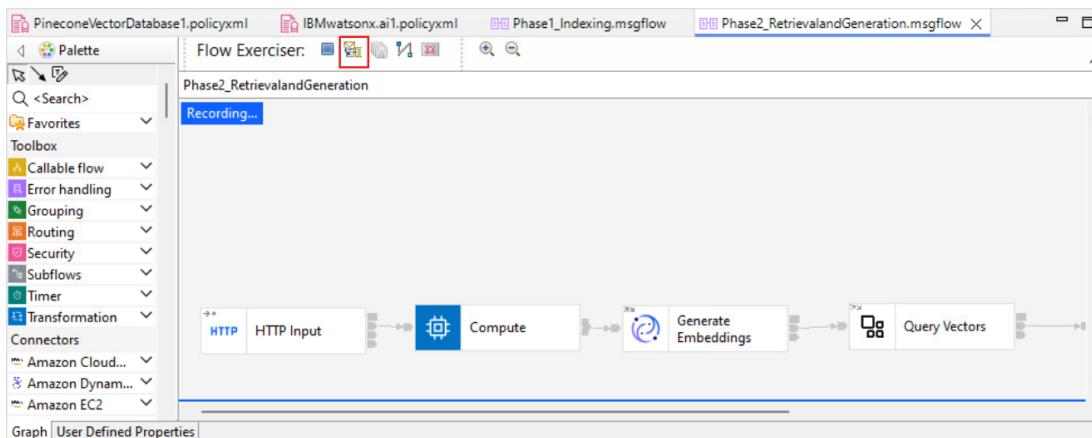
78. We will now send an example text document into the flow for analysis. The lab materials supplied for this lab include a text document named whatsnew1303.txt Copy and paste this file into the directory which will drive the File input node in the message flow Phase1_Indexing.msgflow which is C:\temp\RagInput

You will need to wait a few seconds and the file should disappear from the directory when it is read by the message flow.

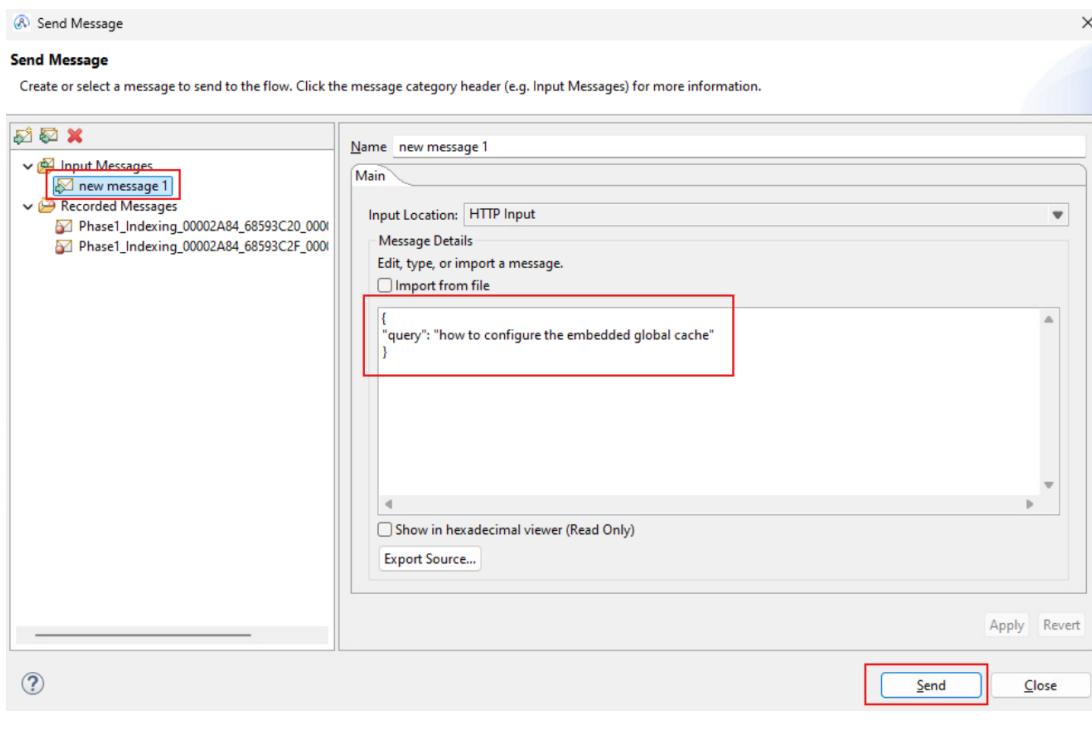
79. In the Toolkit, click the button in the Flow Exerciser task bar to show the path that the File has taken through the message flow. The flow will take about a minute to process ... you may need to click on the button a few times. It is important for both the branches after the FlowOrder node to have executed ... If you are too quick, you may see the upper blue line highlighted before the lower blue line:



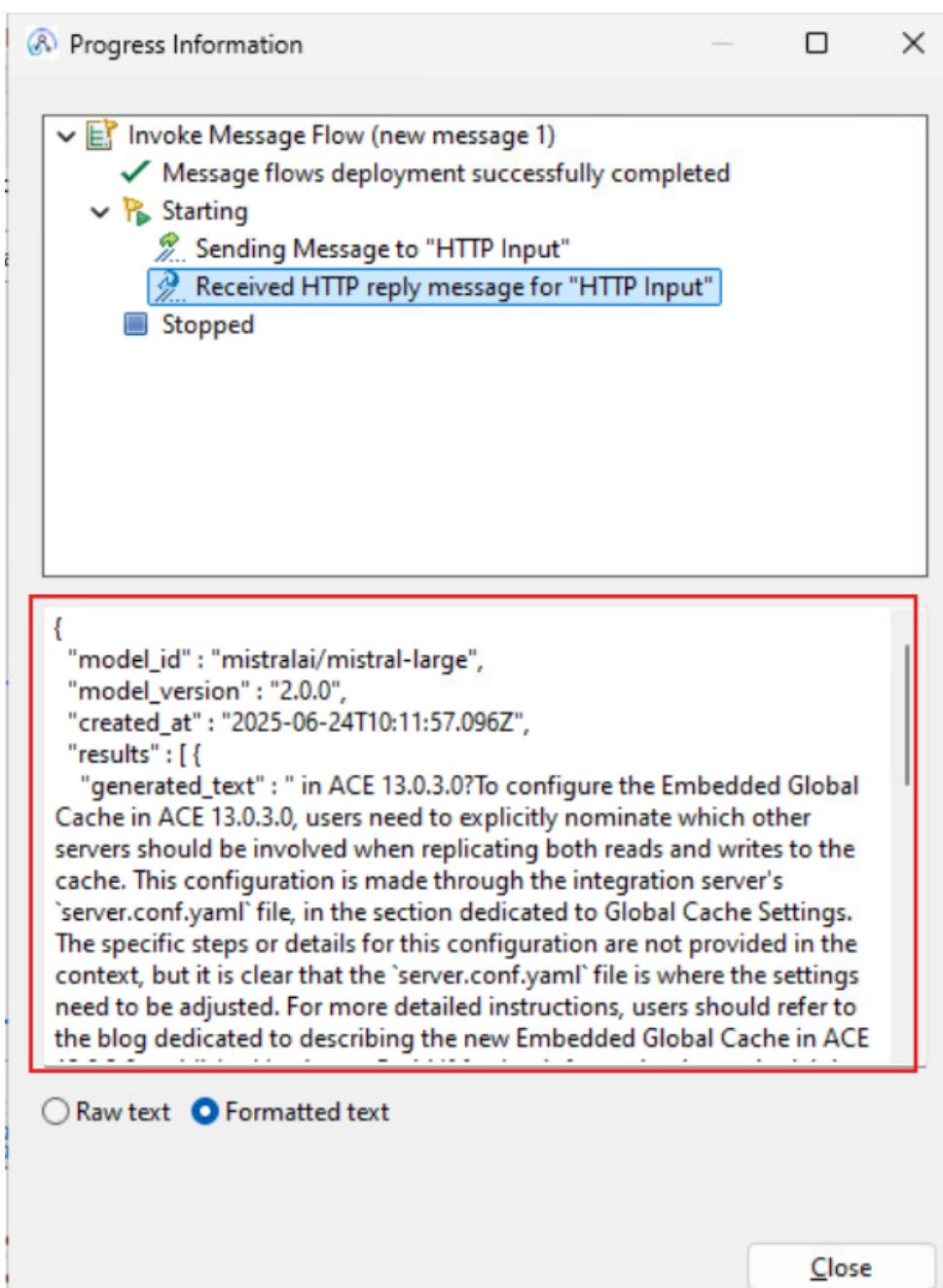
80. Now that we have indexed the input file, we will send in a question about the file. Switch your view to the second message flow named **Phase2_RetrievalandGeneration.msgflow**. Click the Envelope icon in the Flow Exerciser toolbar to send a message into the message flow:



81. In the Send Message dialog, use the example Input message which is provided named new message 1. Click Send button:



82. The query will go through the message flow and reply with the answer to the question!



83. You can experiment with different questions, and also different input data files. Within the Pinecone web browser interface you can also delete and recreate indexes or alternatively you can update the message flow UDPs to reference a new index name.

END OF LAB GUIDE