

**Oracle Banking Enterprise Limits and Collateral
Management Cloud Service
Outbound Flow
Invoke SOAP Service
OnPrem (OBELCM CS)
Release 14.7.5.0.0
Part Number -G14683-01
September 2024**



Table of Contents

1. Configure OIC Agent Group in OIC.....	2-3
2. Target Connection: SOAP Adapter Connection with role "Invoke".....	4-10
3. Configure <i>App Driven Orchestration</i> Integration	11-13
4. Create Integration and configure below steps	
a. Source Trigger Configuration.....	13-18
b. Add Scope Configuration.....	18-20
c. Target Invoke Configuration.....	20-23
d. Data Transformation - Data Mapping for Request and Response.....	23-30
e. Scope - Handling Exception.....	30-44
5. Save and Activate Integration.....	45-47
6. Enable Tracing.....	48-50
7. Test the Integration.....	50-53

Base Doc:

<https://confluence.oraclecorp.com/confluence/display/BLA/Chapter+2.+Outbound+Flow+-+Invoke+SOAP+Service+OnPrem>

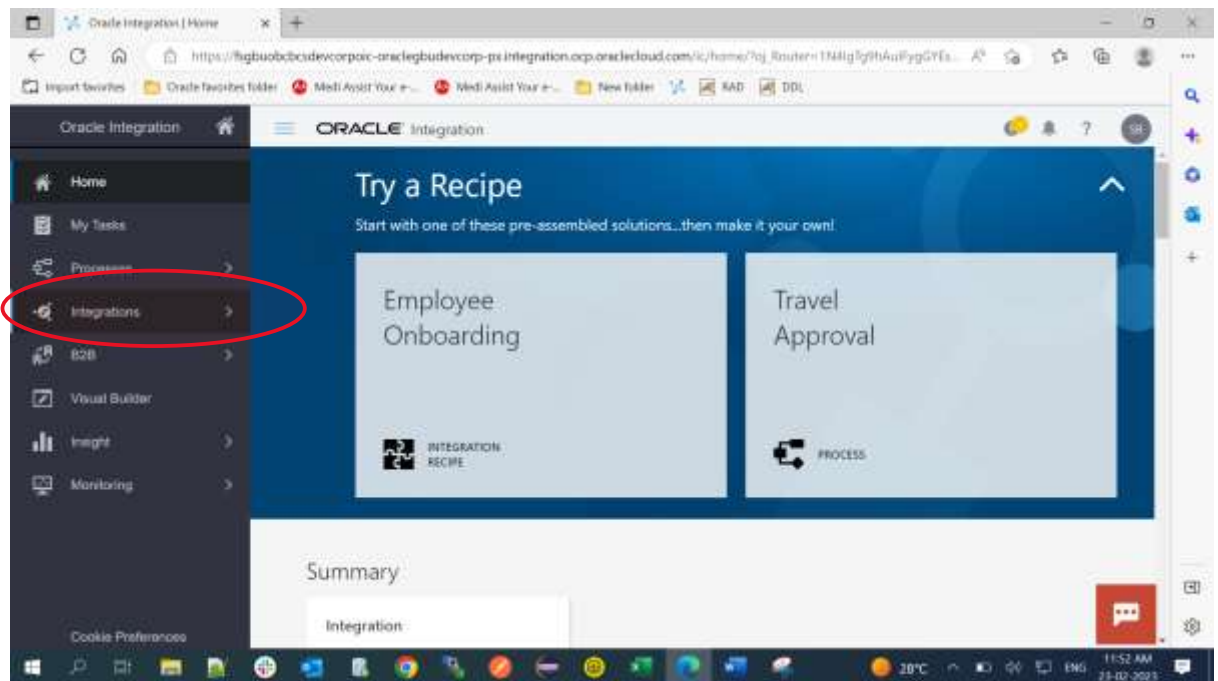
OIC DASHBOARD

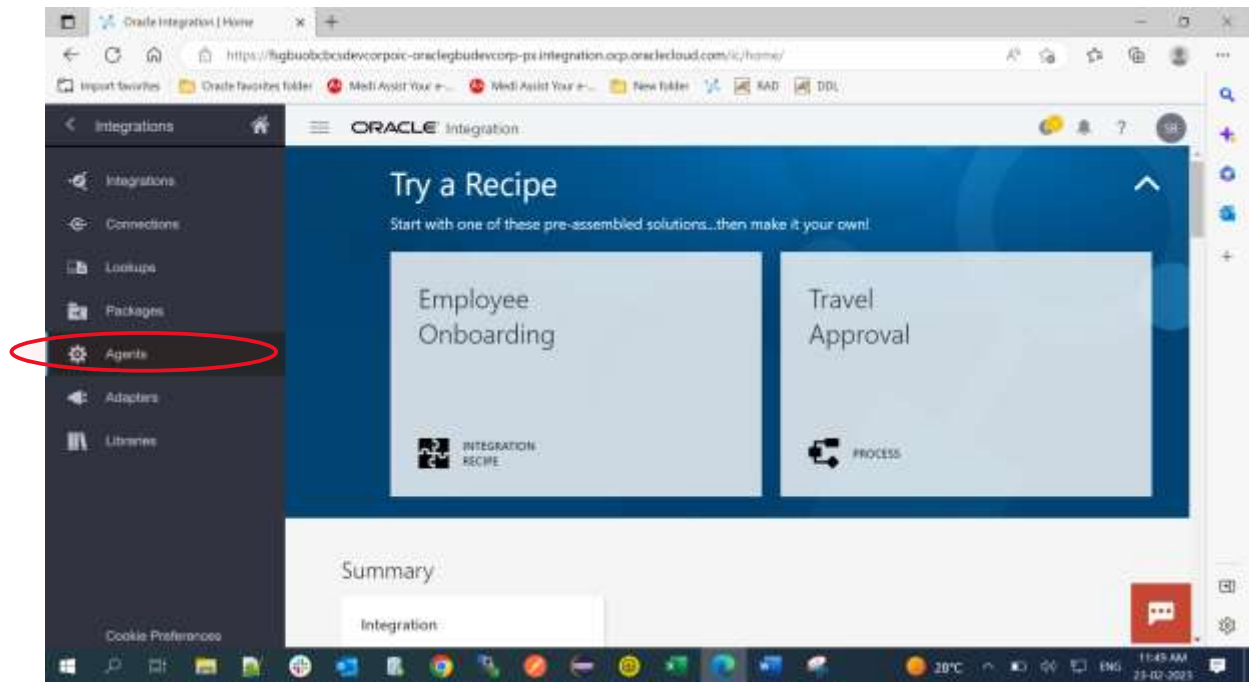
Login to the dashboard using the below URL with SSO login.

URL: <https://fsgbuobcbcsdevcorpoic-oraclegbudevcorp-px.integration.ocp.oraclecloud.com/ic/home>

1. To create an Agent in Oracle Integration:

- 1) In the left navigation pane, click on Navigation Menu > Integrations > Agents.

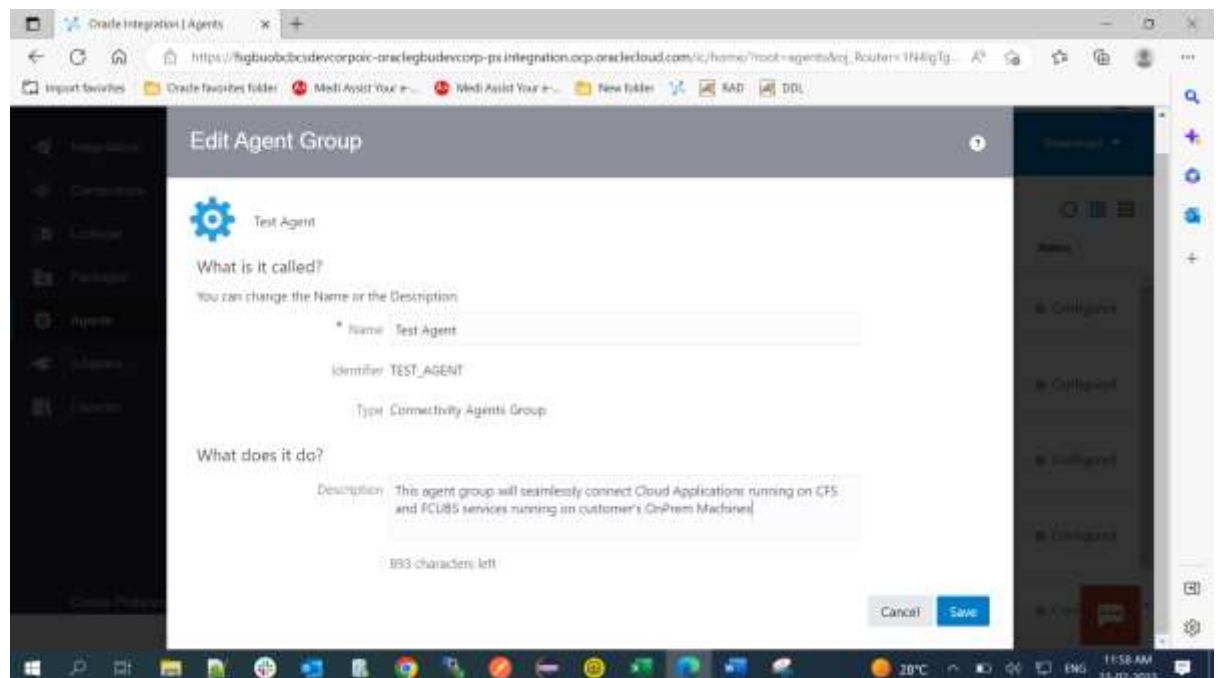




2) Click Create Agent Group.

The Create New Agent Group is displayed.

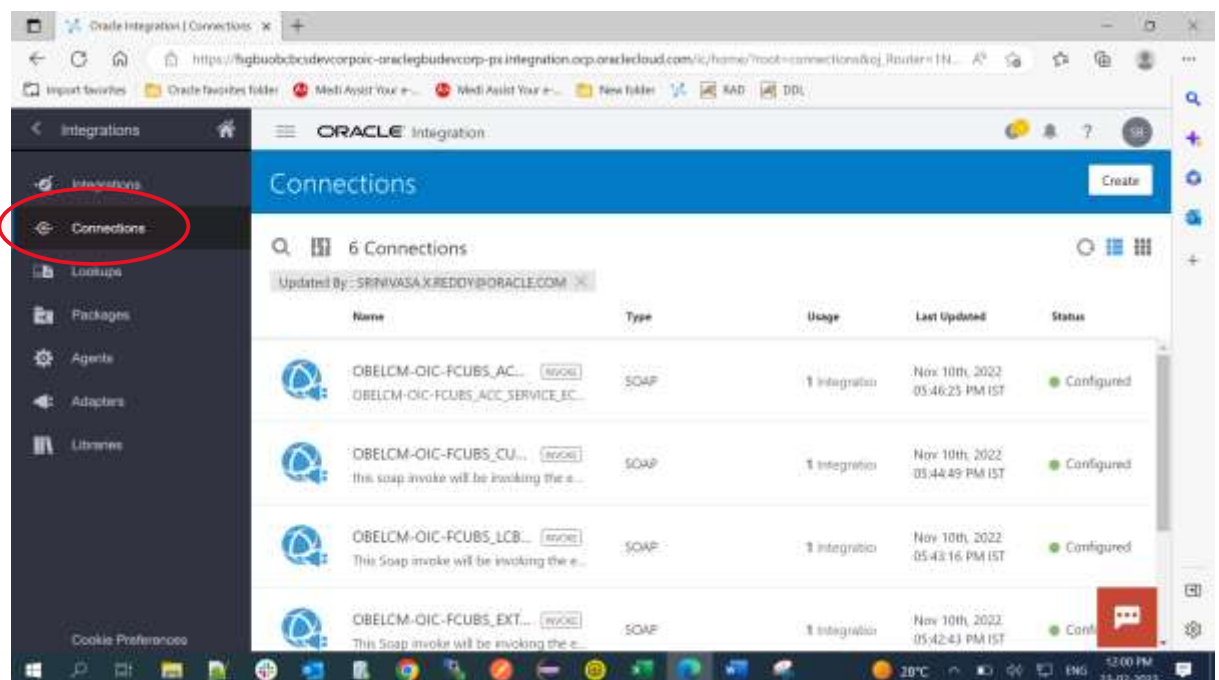
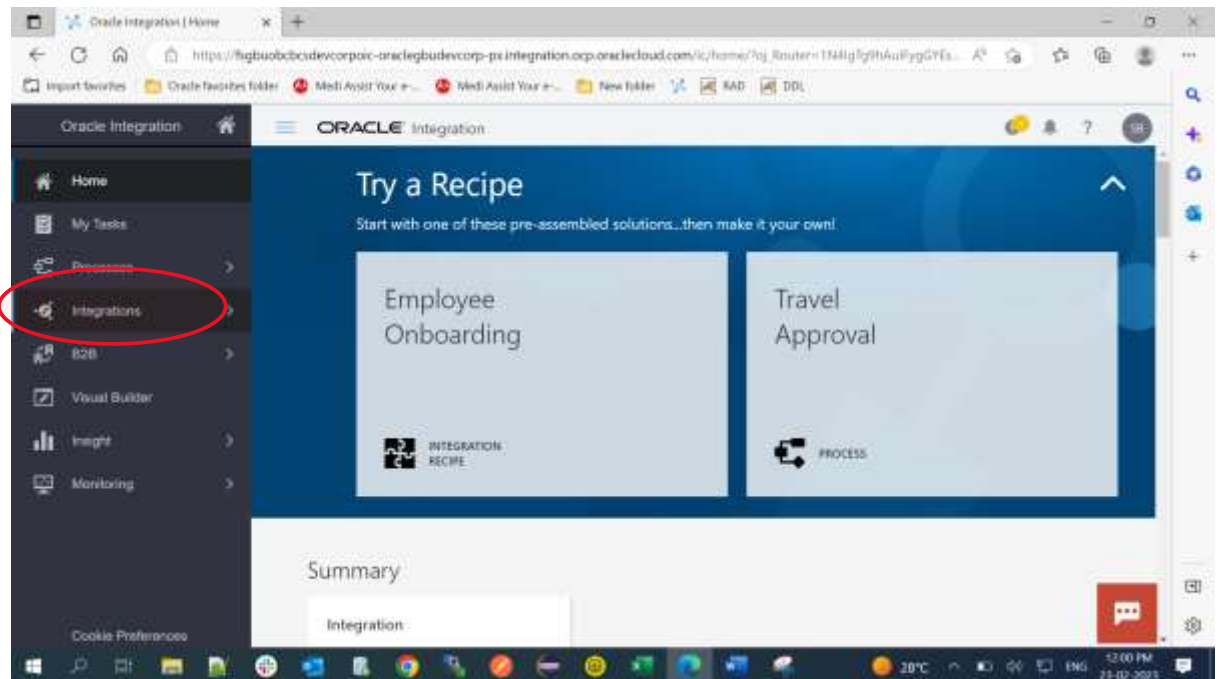
Enter the following information, then click Create.



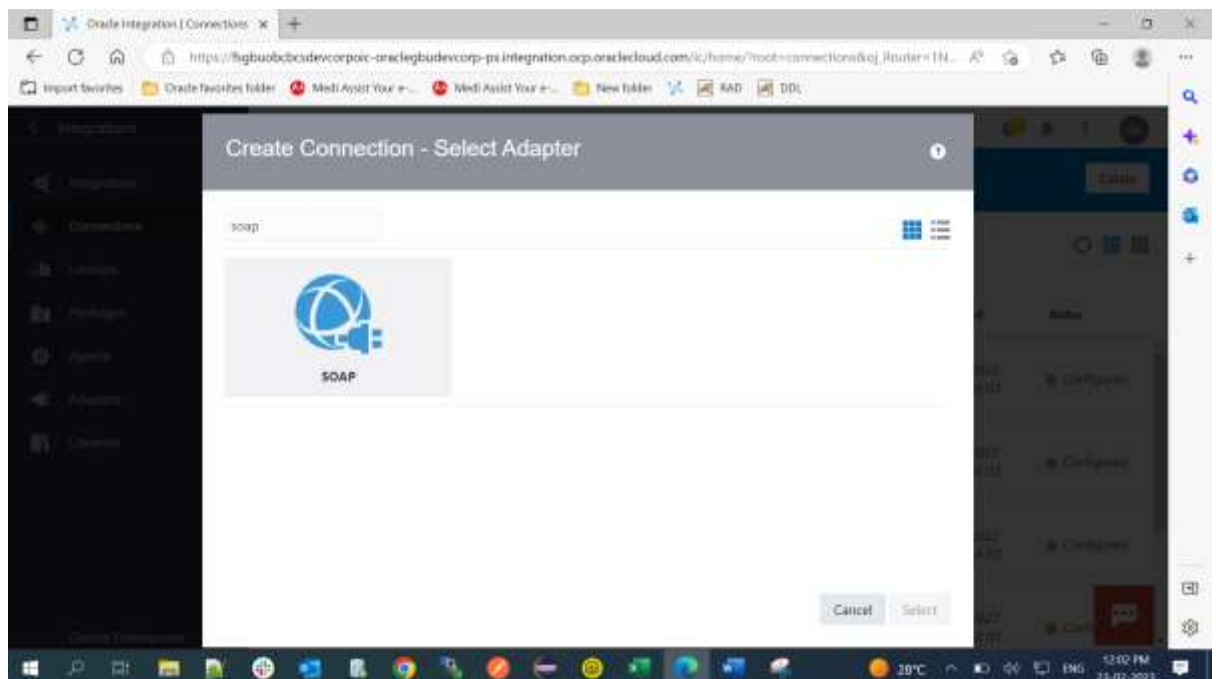
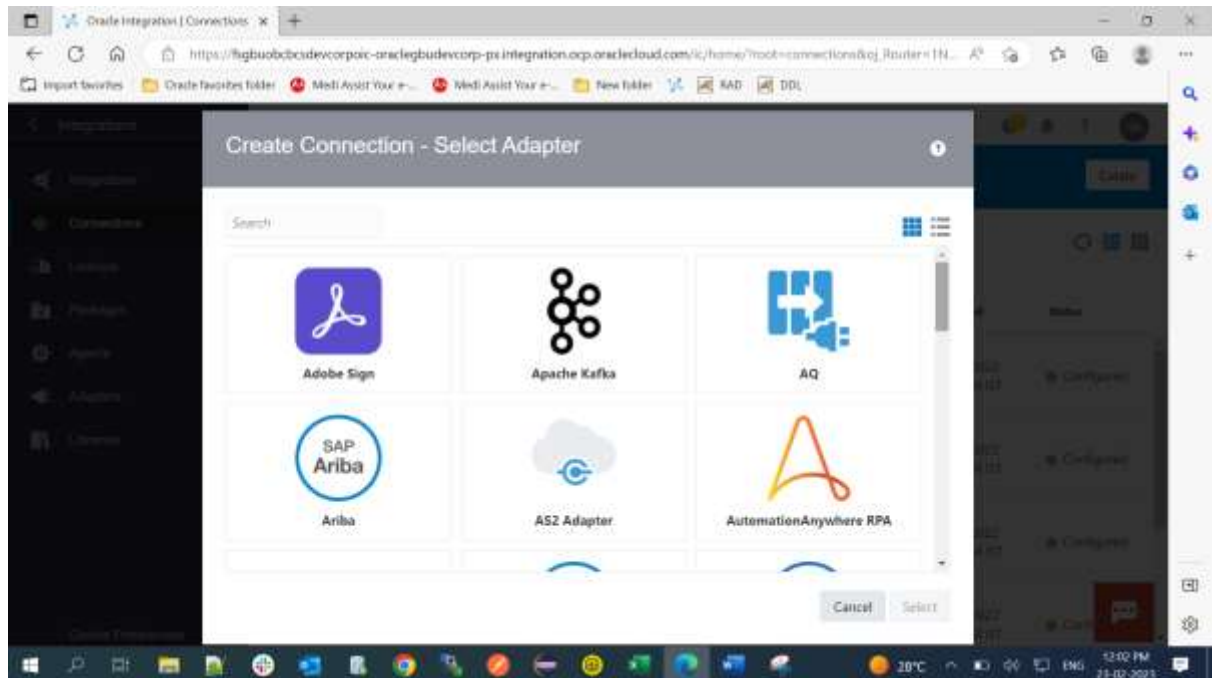
2. Target Connection: SOAP Adapter Connection with role "Invoke" - Required to configure the SOAP Webservices WSDL

To create a Connection in Oracle Integration:

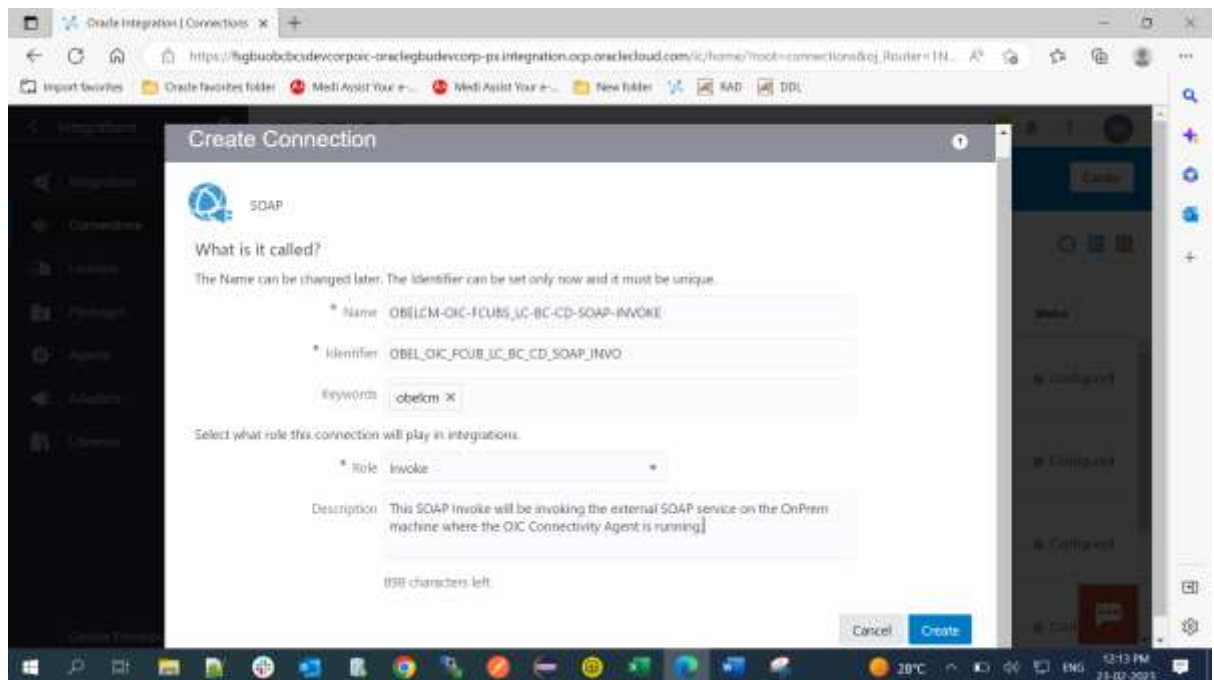
- 1) In the left navigation pane, click on Navigation Menu > Integrations > Connections.



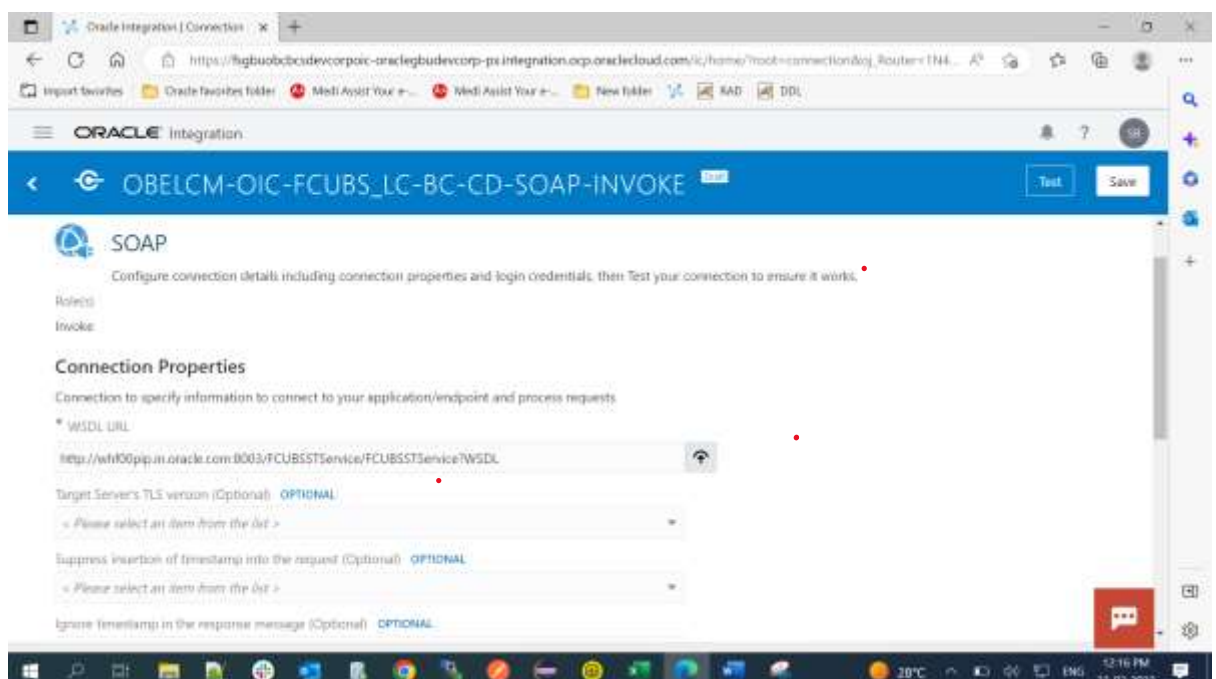
- 2) Click Create → type SOAP in the Search box and select the SOAP adapter and click the Select button

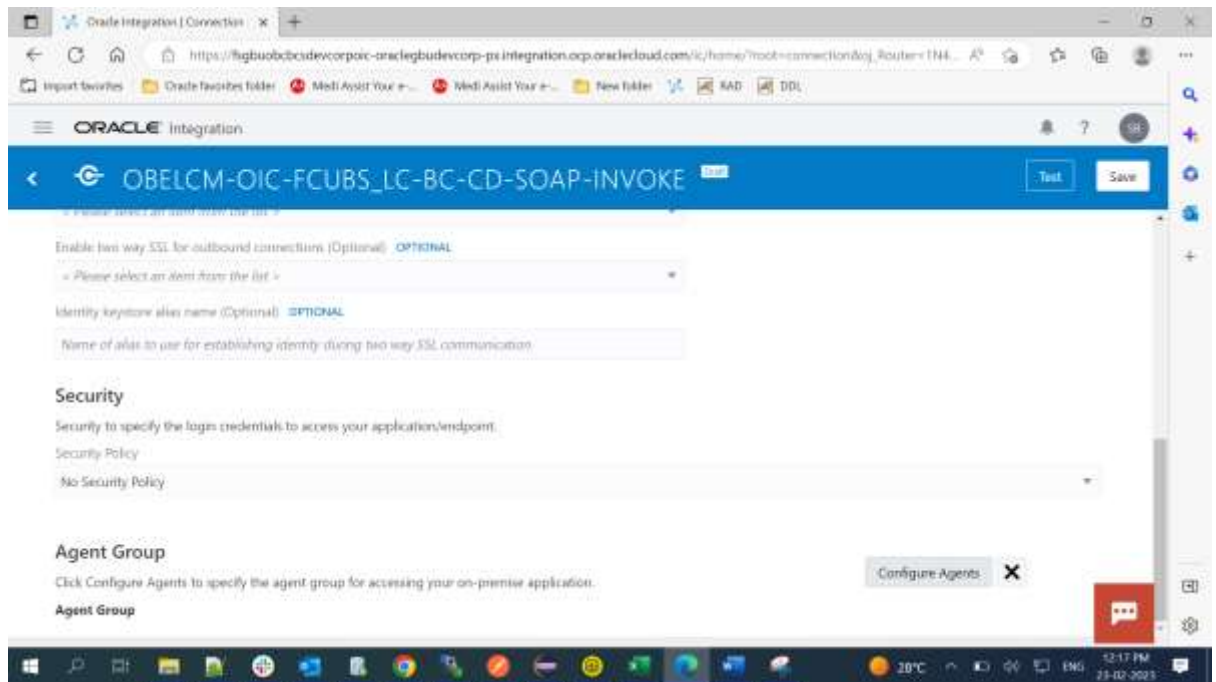


- 3) In the Create Connection dialog, enter the information that describes this connection and Click Create.



Configure Connection Properties for Invoke Connections and also Configure Connection Security as **No Security Policy**





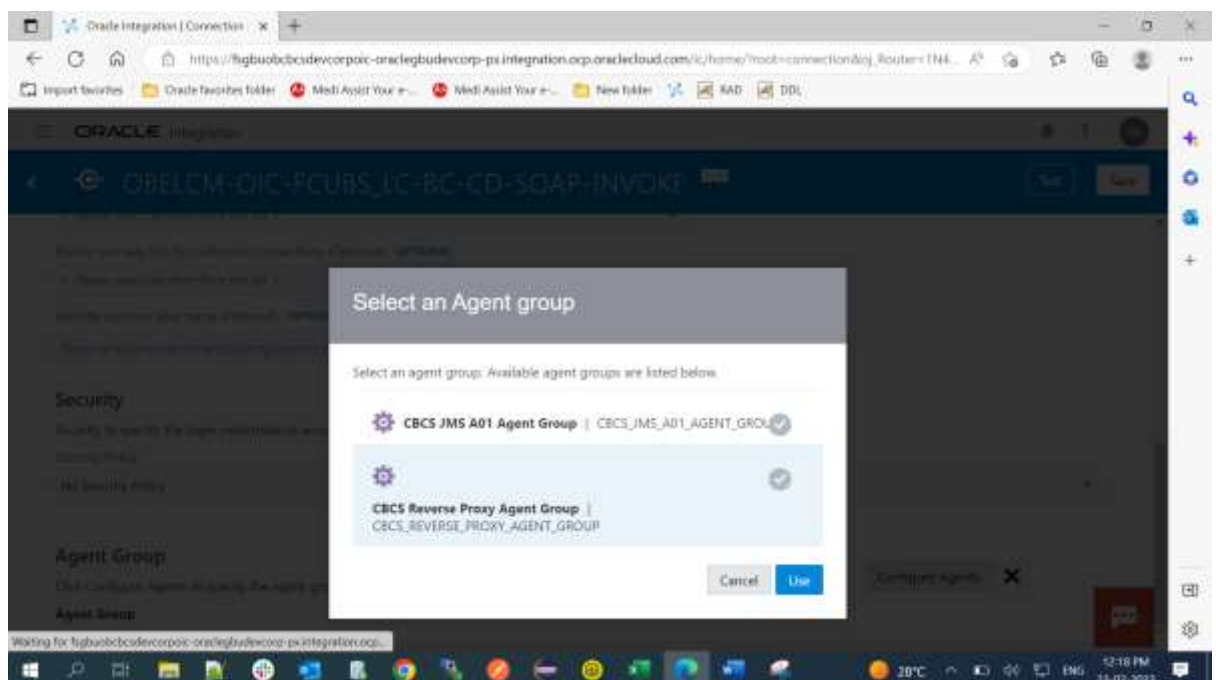
4) WSDL for the Connection:

<http://100.76.154.176:7001/FCUBSSTService/FCUBSSTService?WSDL>

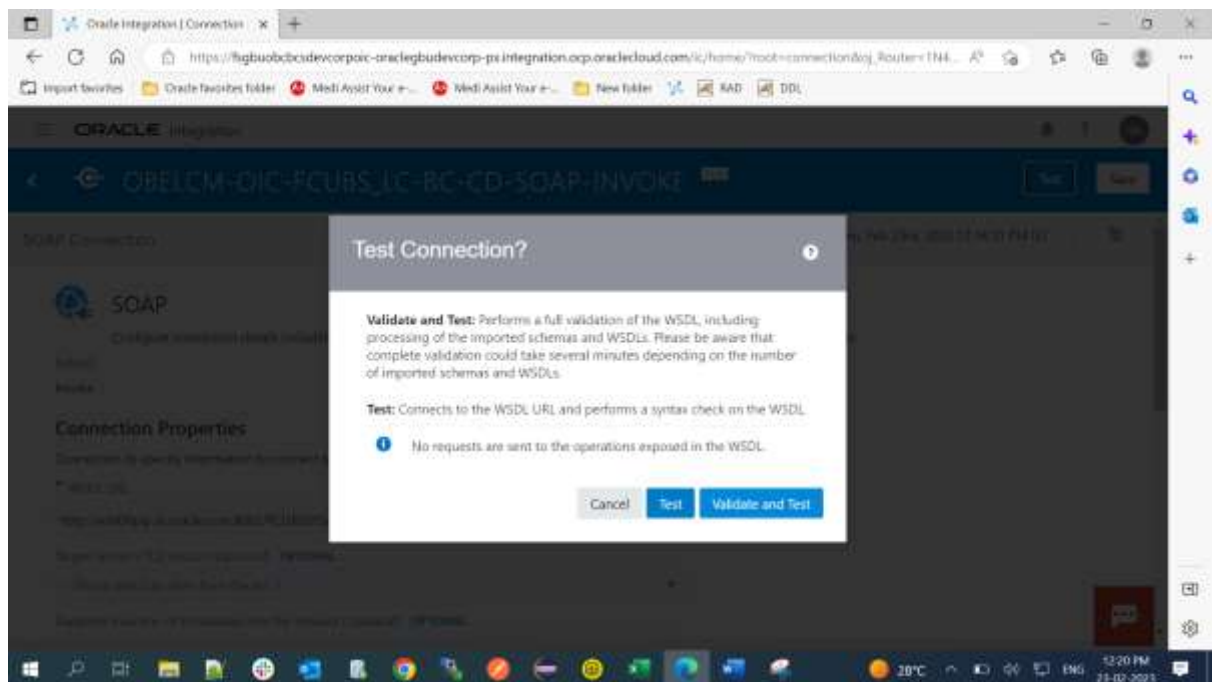
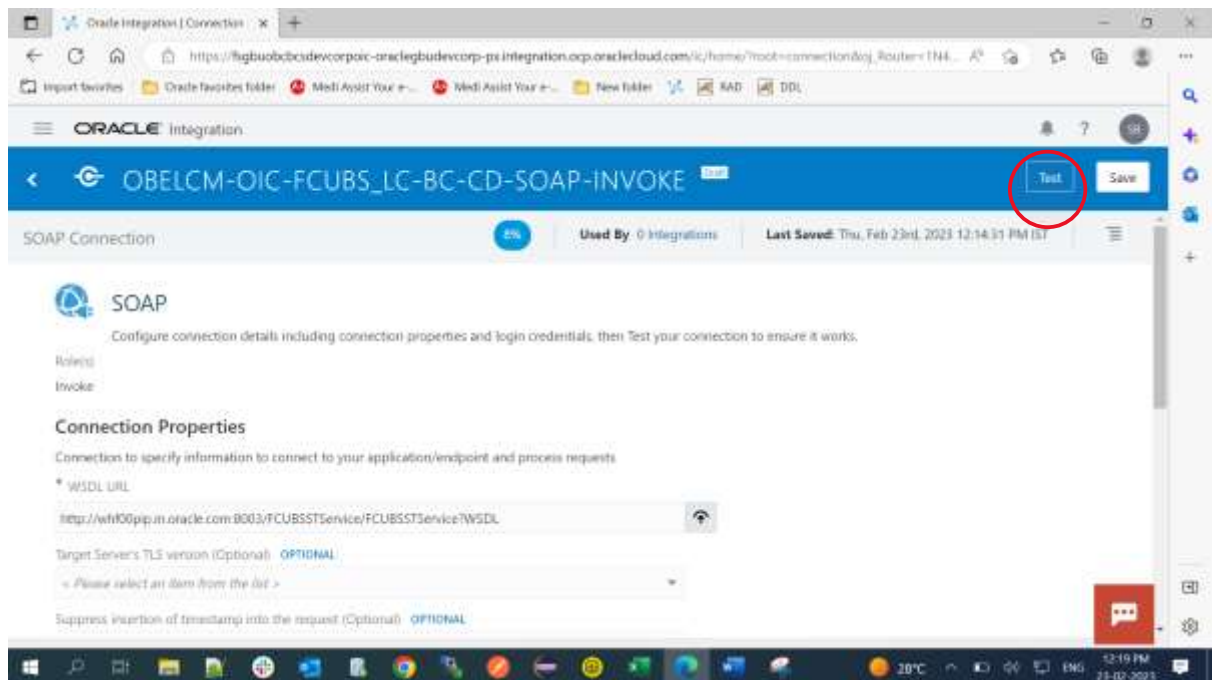
5) Configure an Agent Group

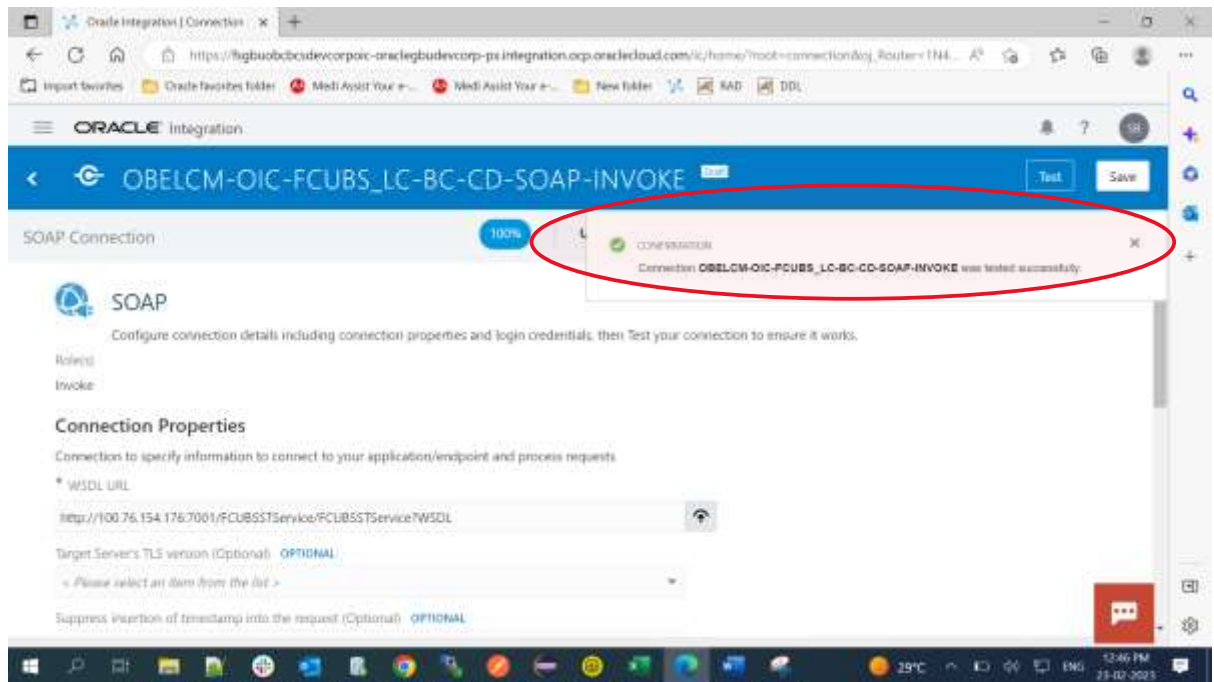
Configure an agent group for accessing the service hosted on your premises behind the fire wall.

- a) Click Configure Agents.
- b) The Select an Agent Group page appears.
- c) Click the name of the agent group. Click Use.

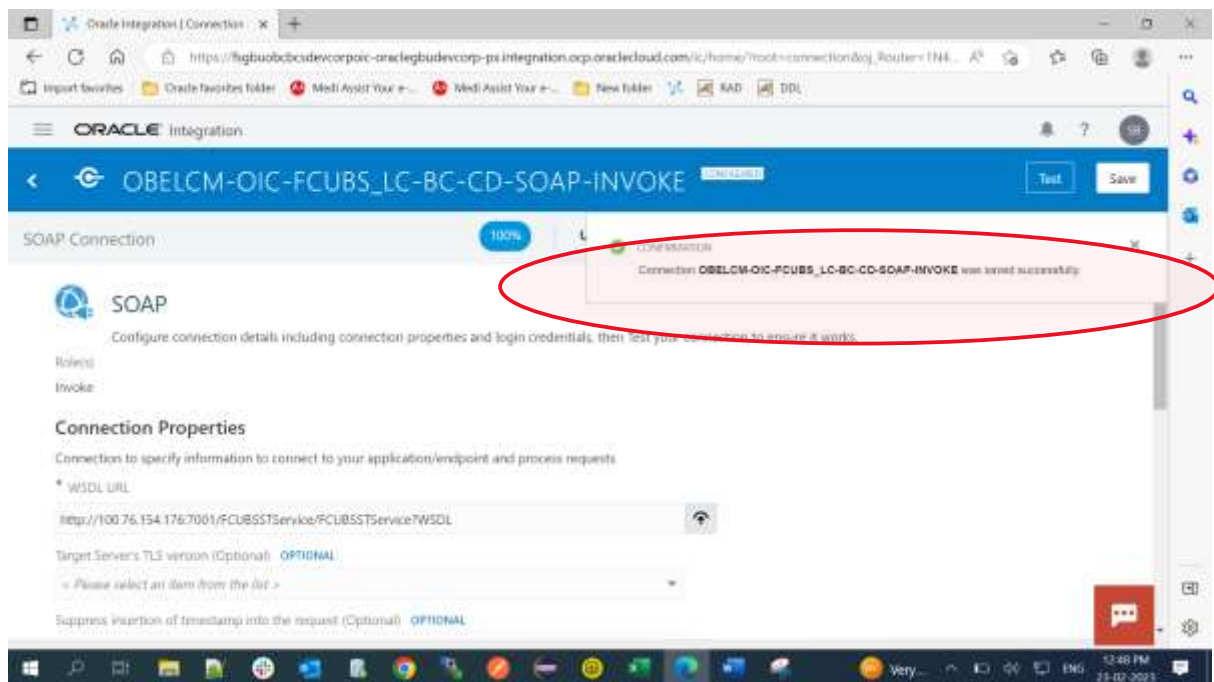


6) Validate and Test the Connection

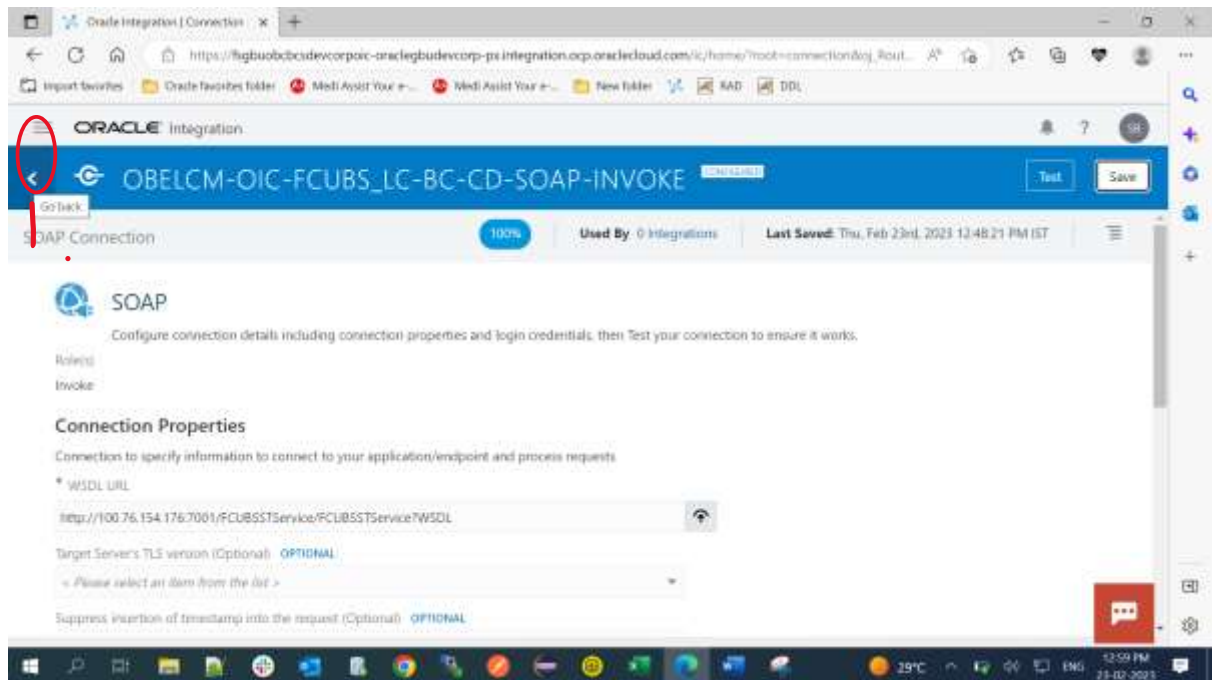




7) Save the Connection after Test is successful



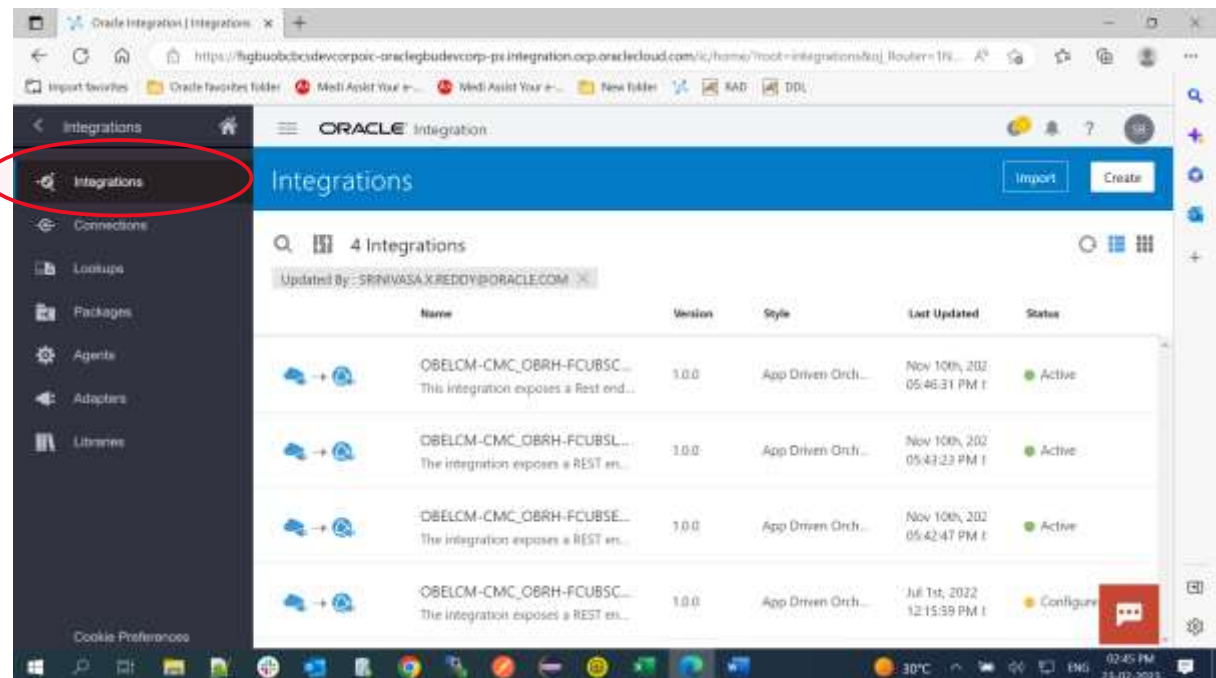
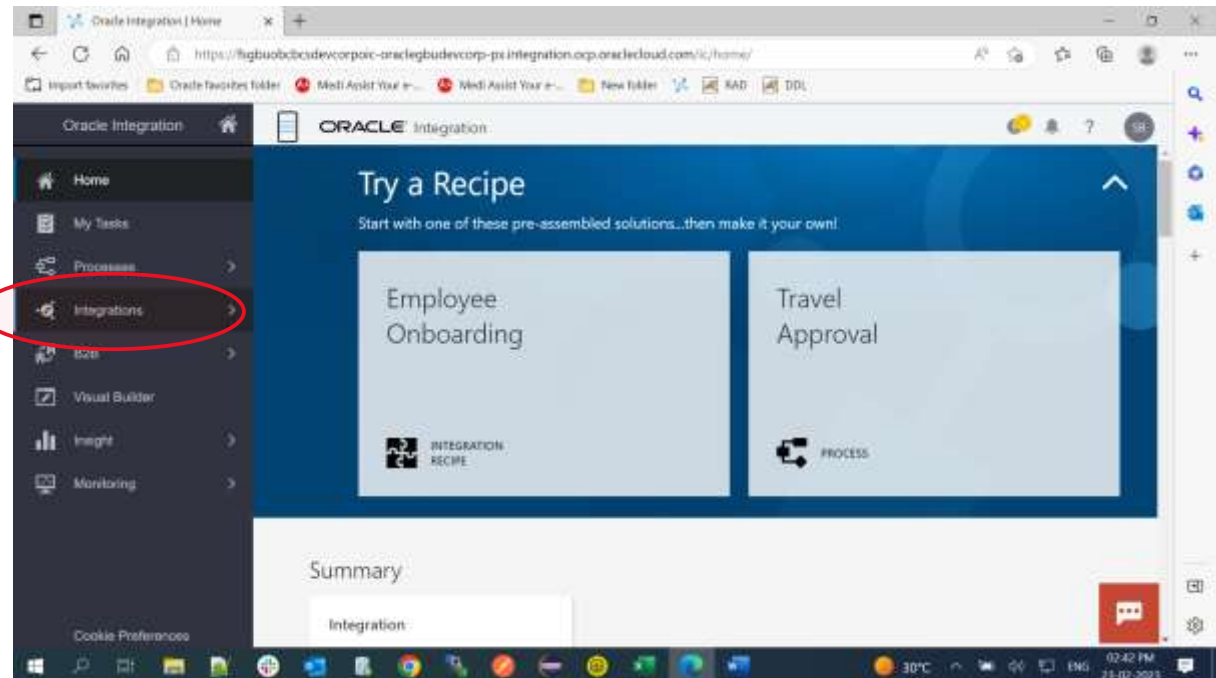
8) When complete, click Save, then click the Back button.



3. Configure App Driven Orchestration Integration Style for Integrating the OnPrem SOAP Services in OIC.

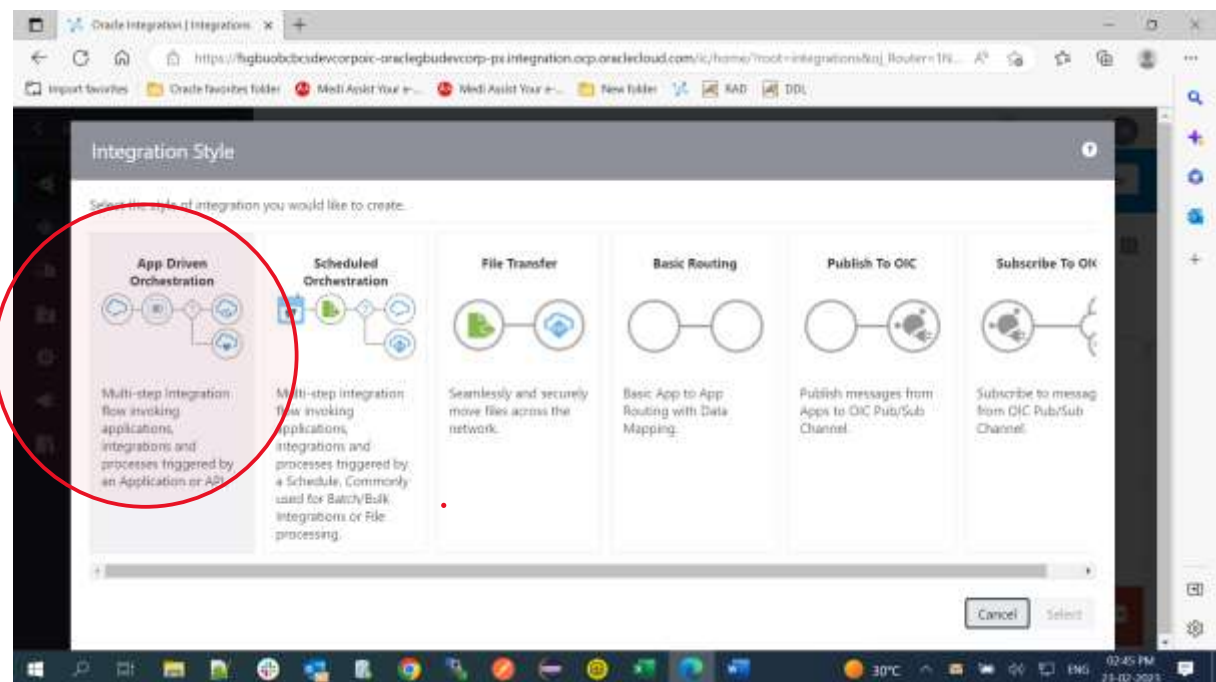
To create an Integration

- 1) In the left navigation pane, click on Navigation Menu > Integrations > Connections

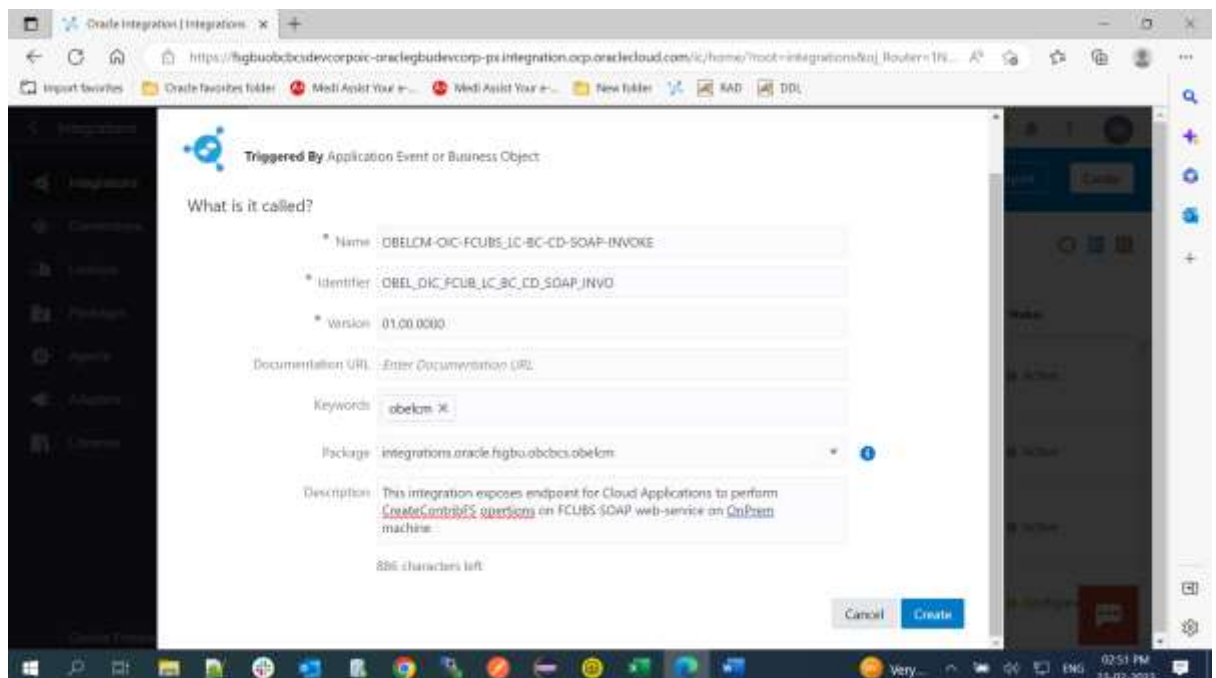


2) On the Integrations page, click Create.

Select the **App Driven Orchestration** type of integration style and click Select



3) Enter the following information



4) Click Create

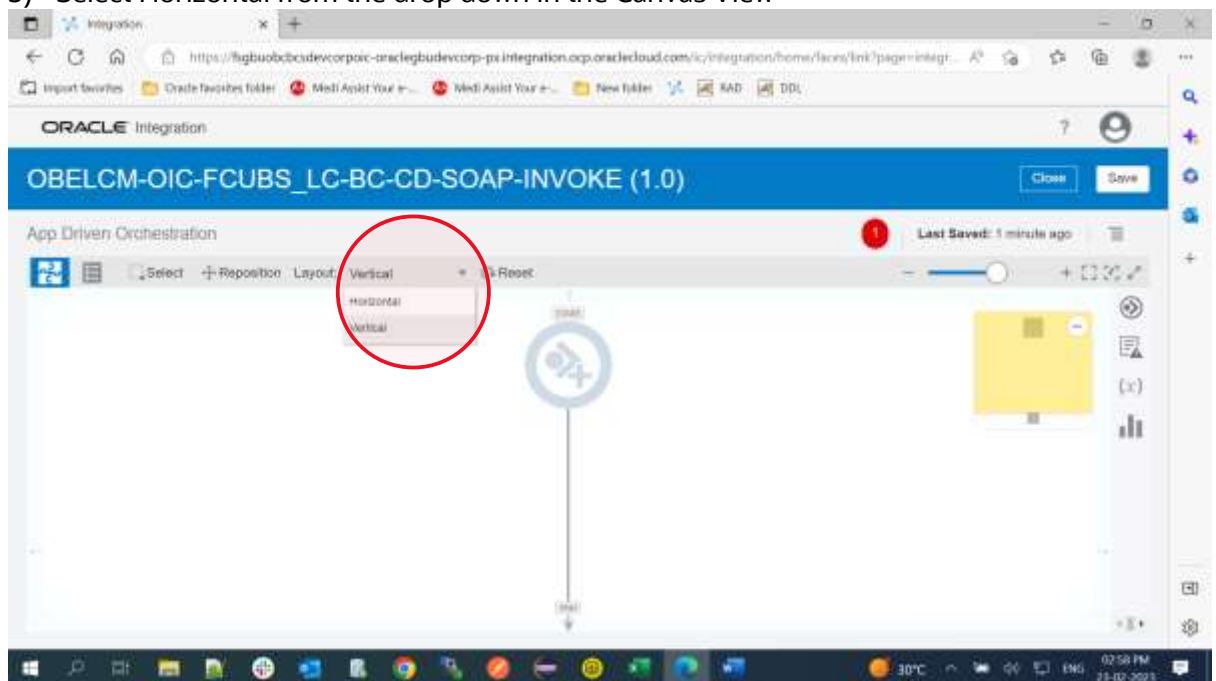
An empty integration canvas with the following sections is displayed



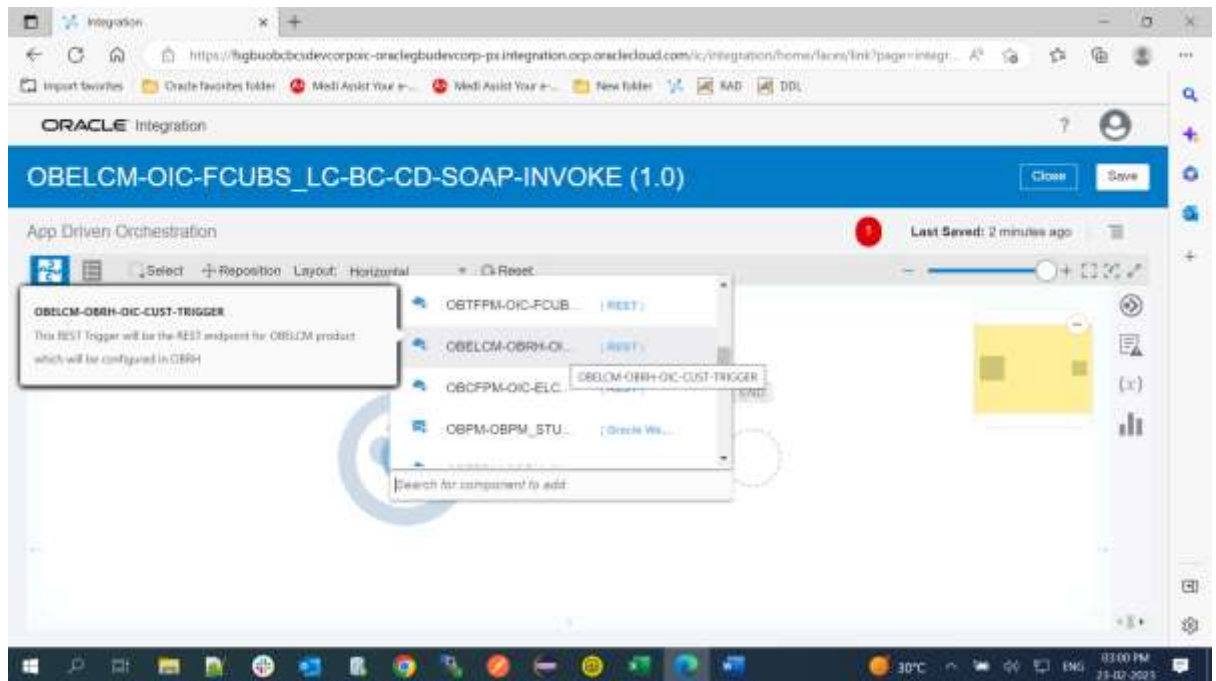
4. Create Integration and configure below steps

a. Source Trigger Configuration

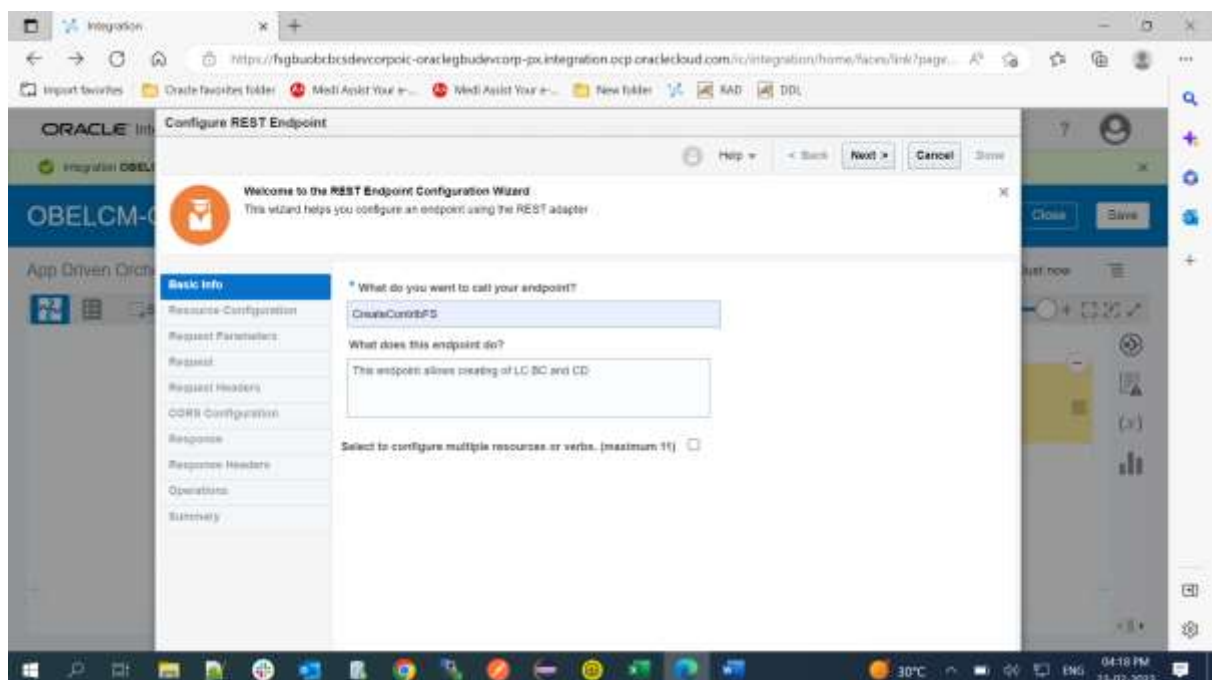
5) Select Horizontal from the drop down in the Canvas View



- 6) Click on + button and select trigger connection “**OBELCM-OBRIH-OIC-CUST-TRIGGER**” created

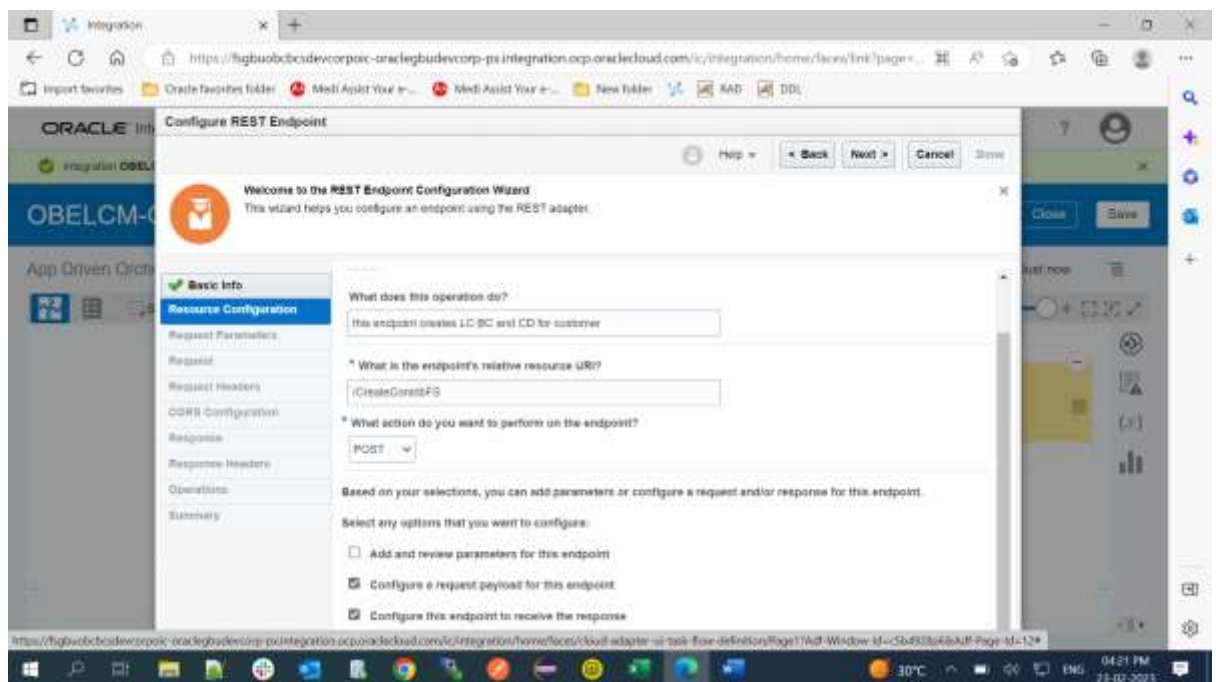


- 7) In **Configure Rest Endpoint → Basic Info** wizard, Enter Name for the endpoint and what the endpoint does then click Next

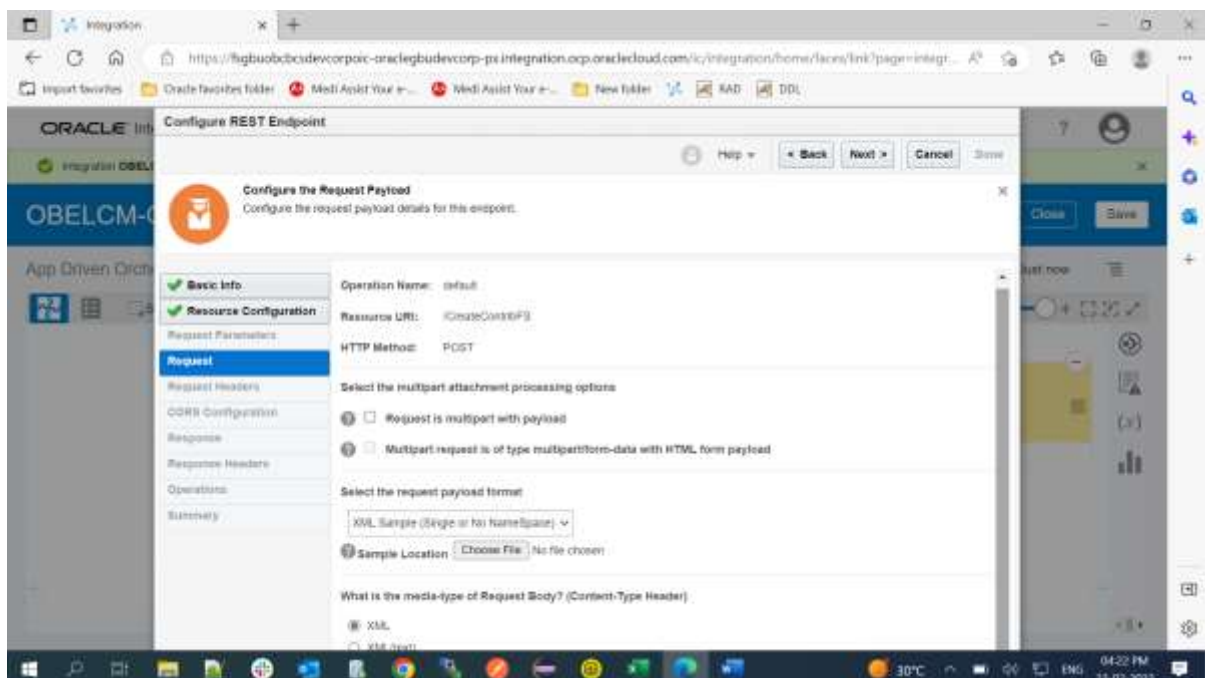


Note: If you have multiple resources to be configured in the same integration, then you check the select box for "Select to configure multiple resources or verbs (maximum 11)"

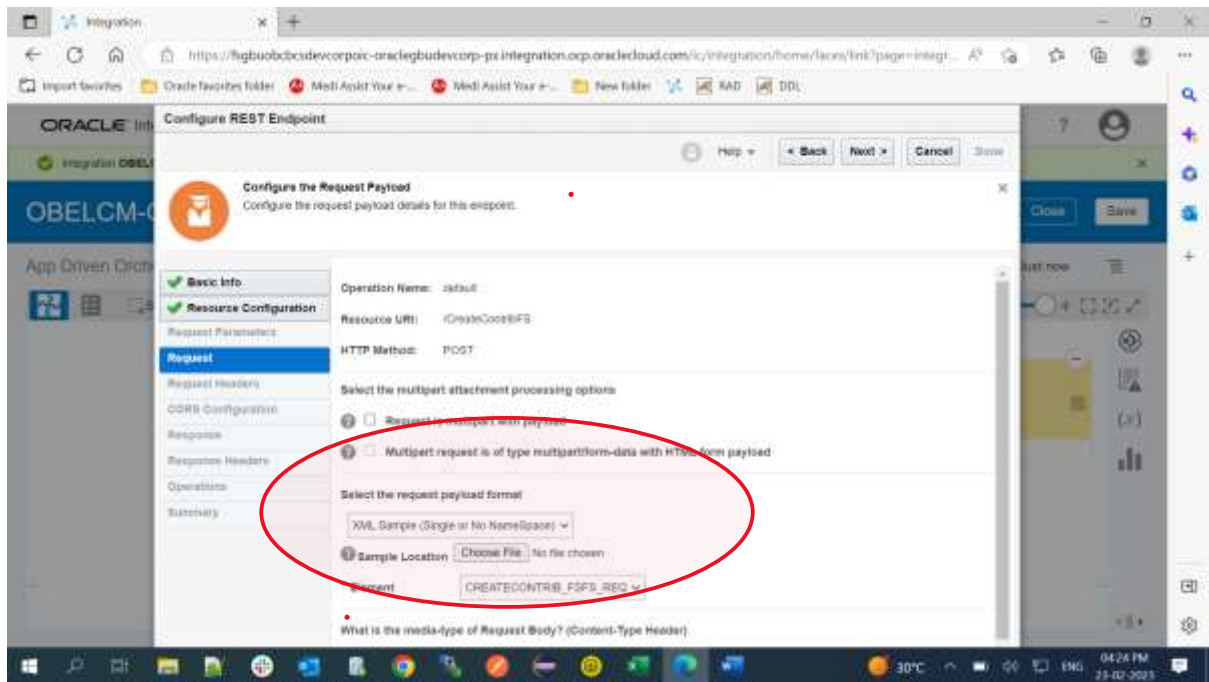
- 8) In **Configure Rest Endpoint → Resource Configuration** wizard, Enter the details as given below and click Next



- 9) In **Configure Rest Endpoint → Request** wizard, Select Details as shown below



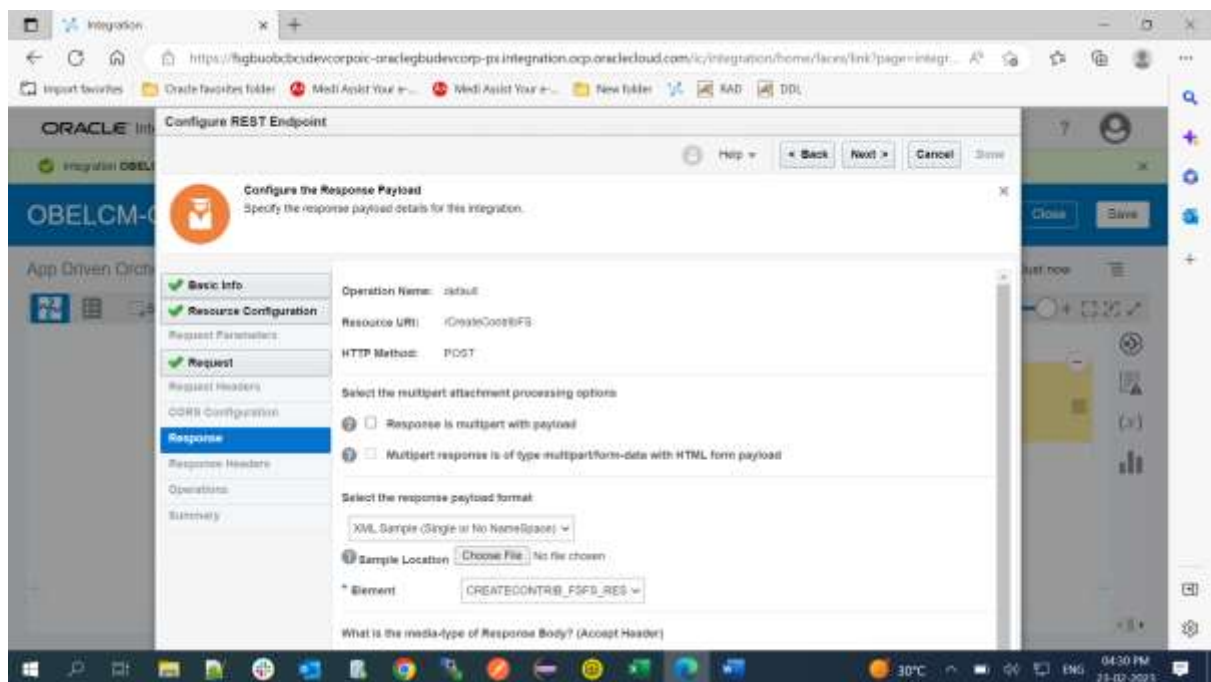
click Choose File, **select File and Open the request.xml** for this operation then click **ok** and **click Next**

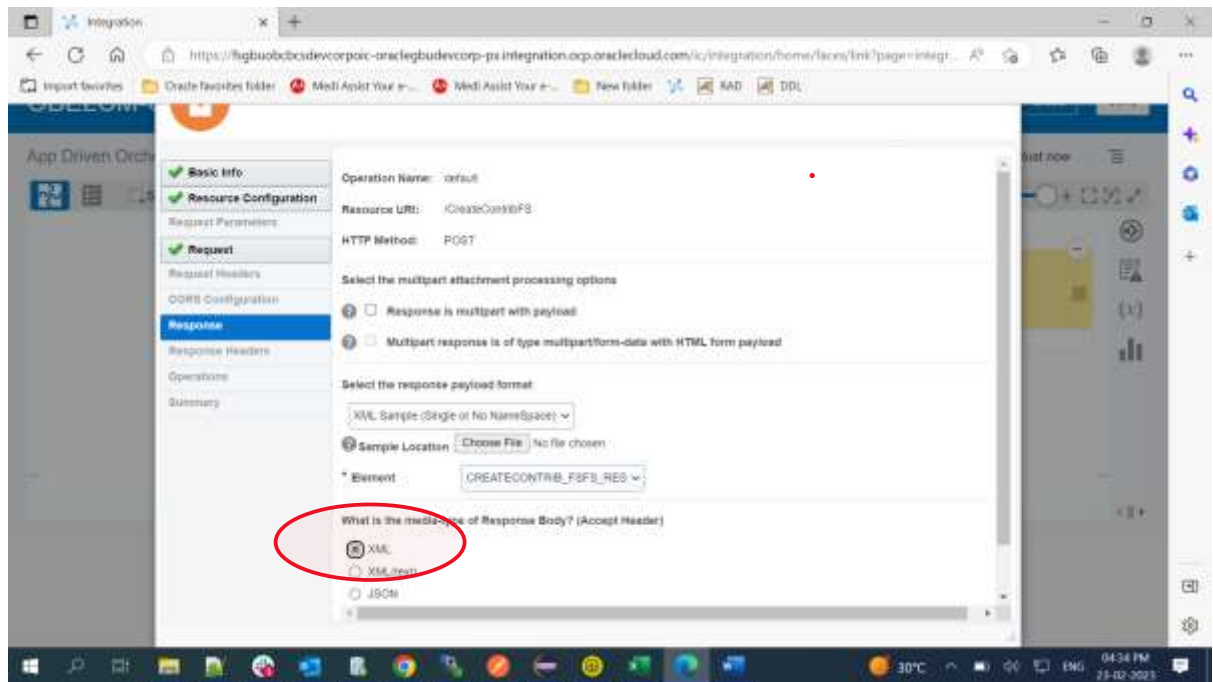


Request xml format

BCLCCD\BC LC CD BLK REQ.xml

10) In **Configure Rest Endpoint** → **Response** wizard, Select Details as shown below



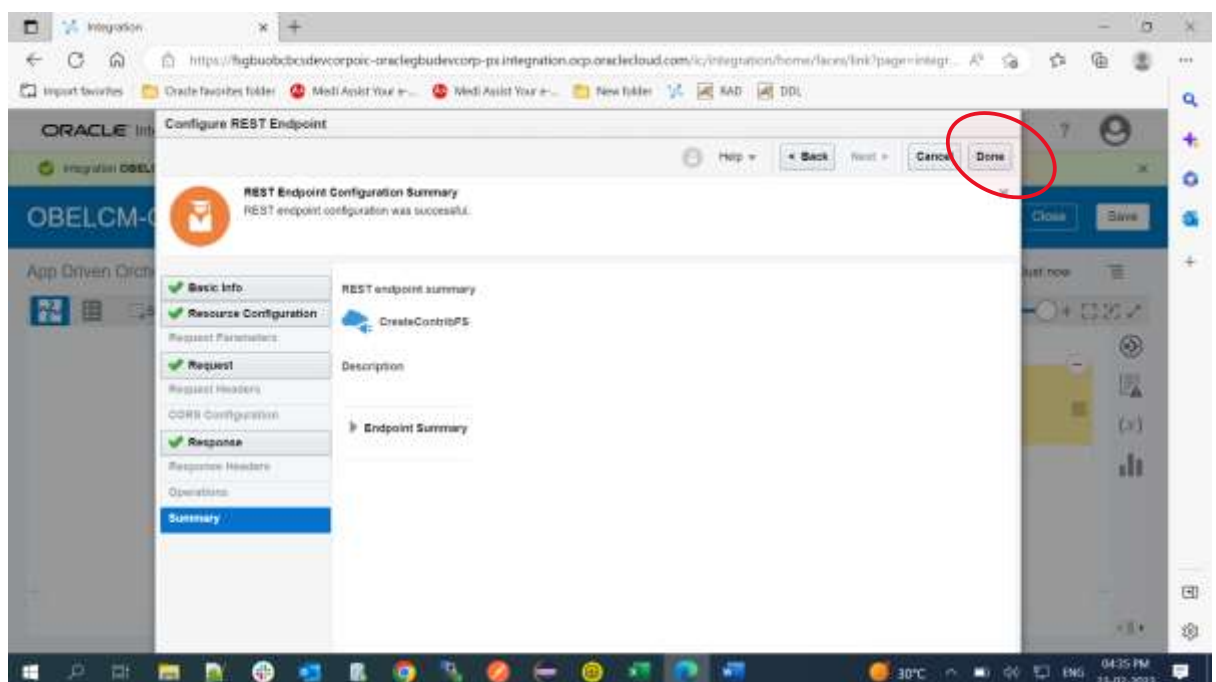


click Choose File, select File and Open the response.xml for this operation then click ok and click Next

Response xml format

BCLCCD\BC_LC_CD_BLK_RESP.xml

- 11) In **Configure Rest Endpoint** → **Summary** wizard, Click Done

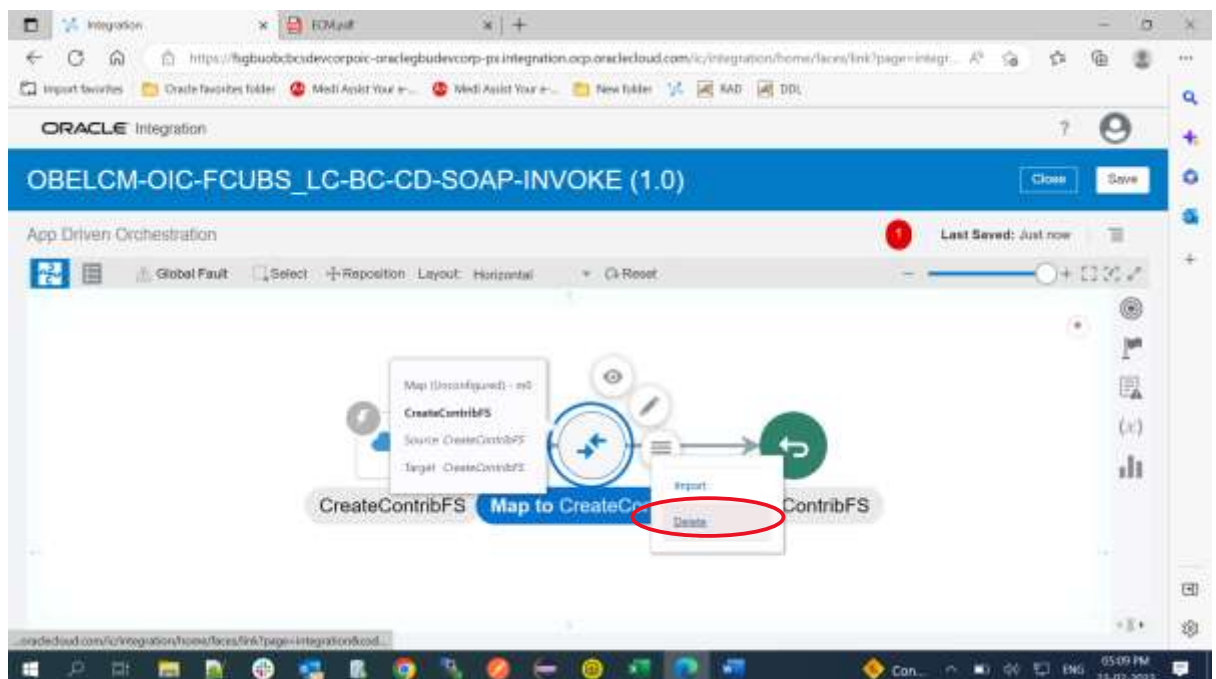


- 12) Trigger Connection setup completed in integration, you should see below Canvas screen - click Save



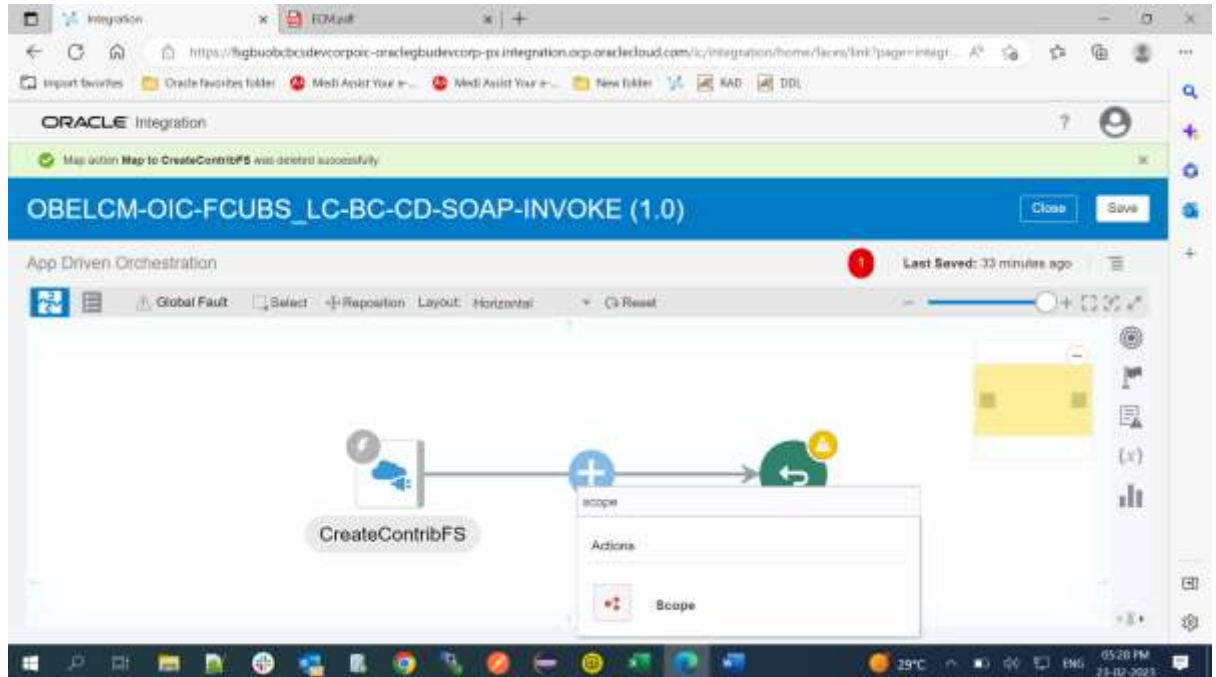
b. Add Scope Configuration

- 13) Delete Map to CreateContribFS, select the Map → click on More Actions Button → click Delete

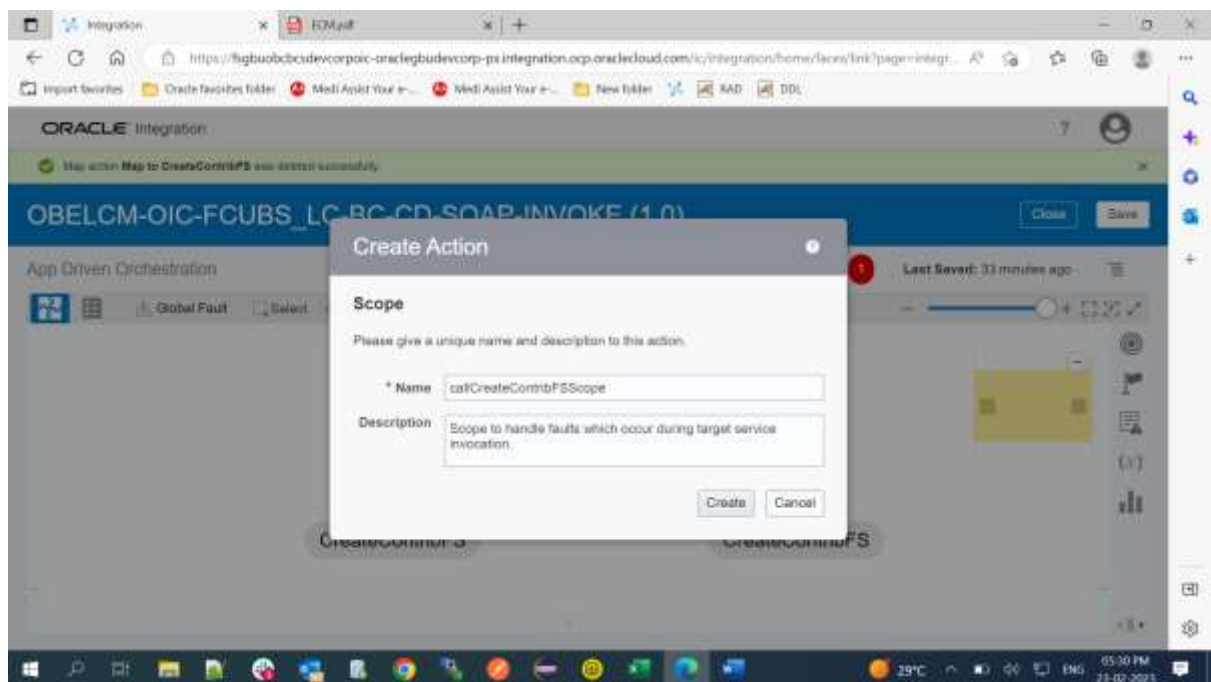


14) Click Save

15) Hover the mouse on the arrow as shown below, click + and search for Scope and Click on Scope component.



16) Enter the Following details for Scope and click **Create**



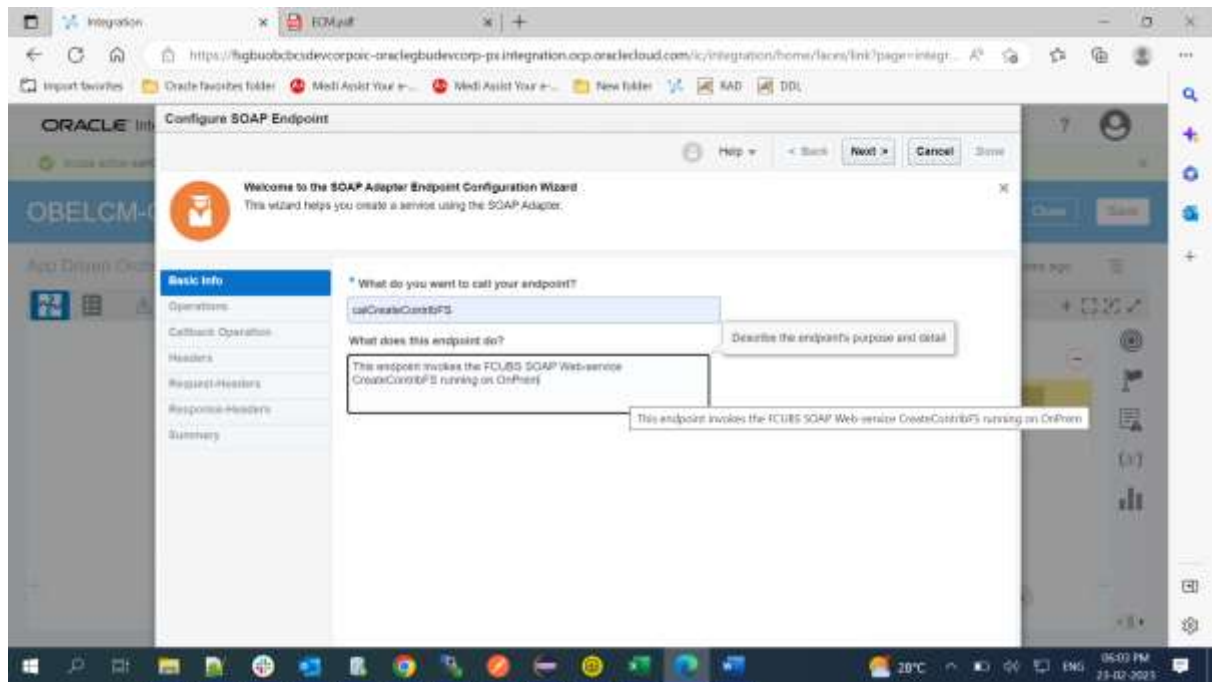


c. Target Invoke Configuration

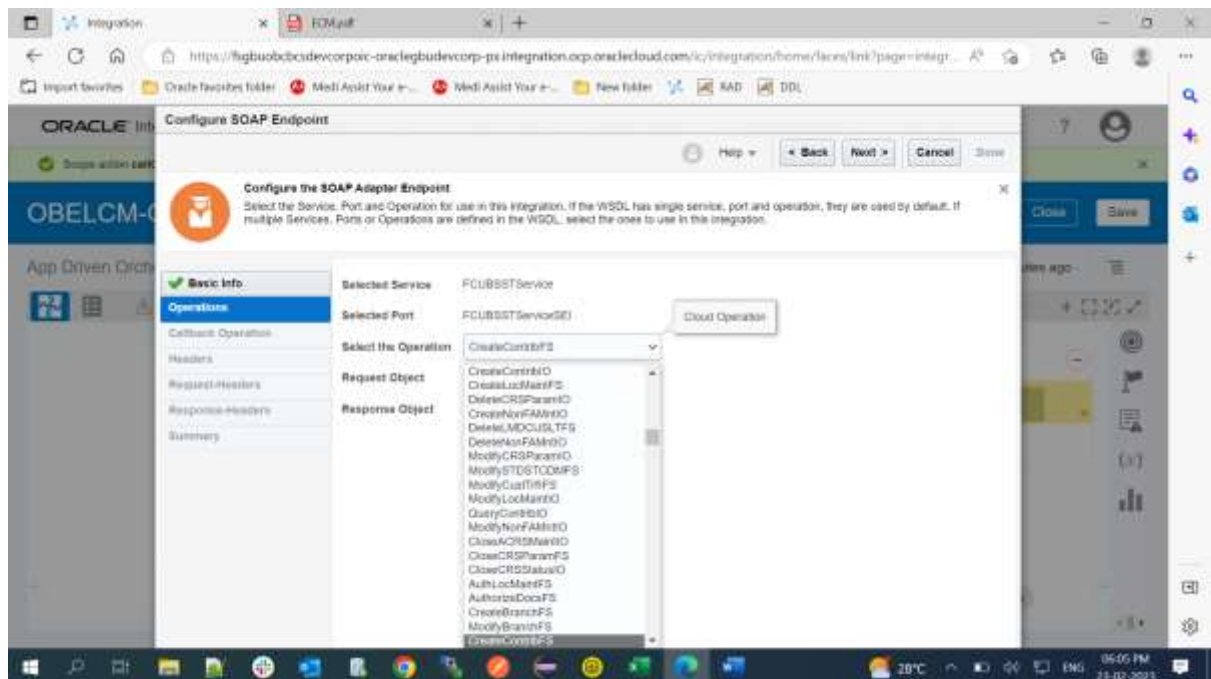
- 17) Click + button which is inside the callCreateContribFSScope and search for Invoke connection "OBELCM-OIC-FCUBS_LC-BC-CD -SOAP-INVOKE" created earlier

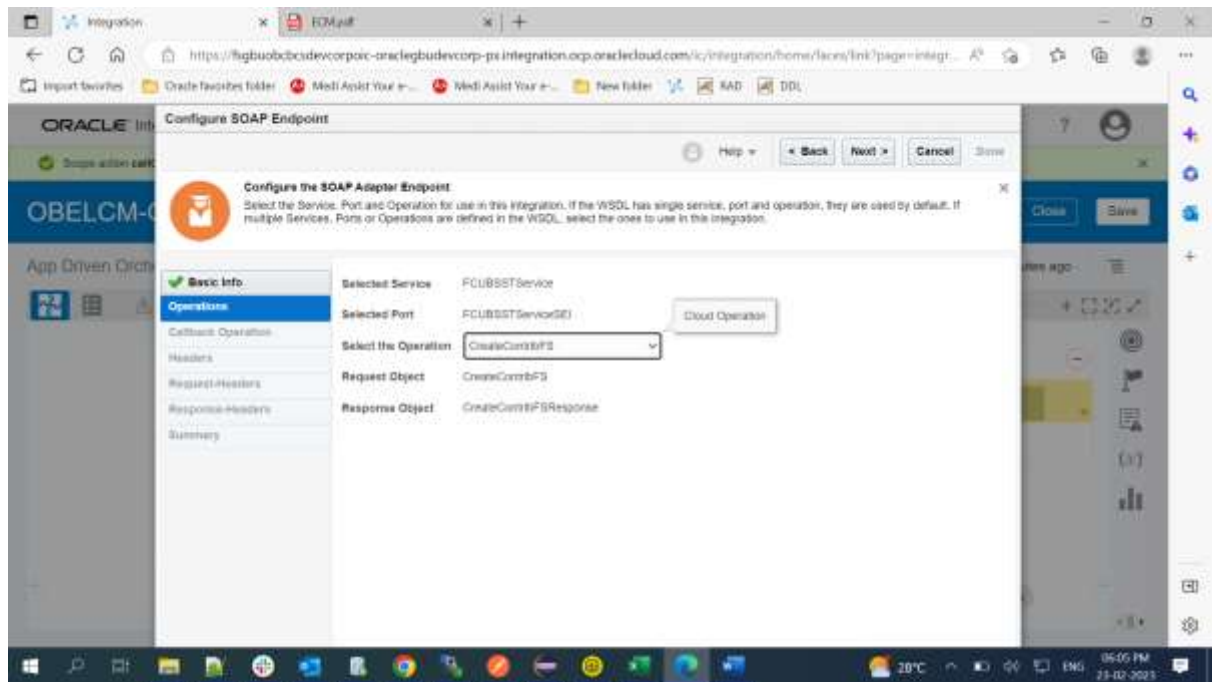


- 18) In **Configure SOAP Endpoint → Basic Info** wizard, Enter the below details and click **Next**

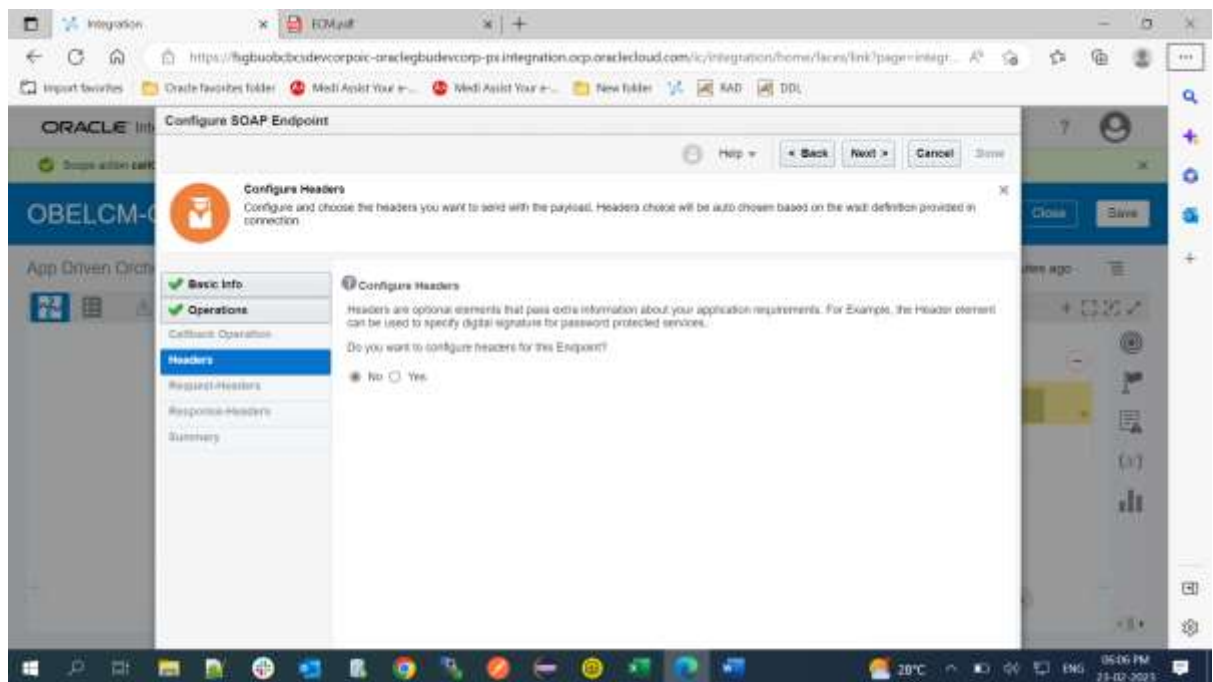


- 19) In **Configure SOAP Endpoint → Operations** wizard, Select CreateContribFS from the drop down as shown below and click Next

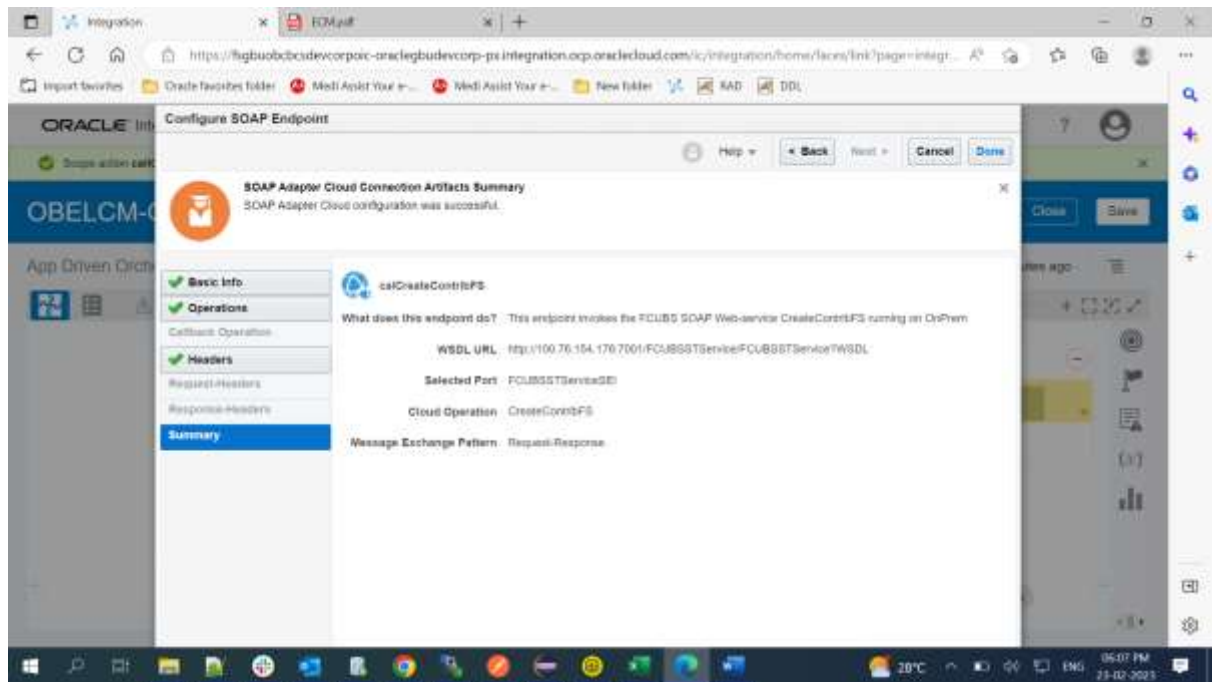




20) In **Configure SOAP Endpoint** → **Headers** wizard, select **No** and click **Next**



21) In **Configure SOAP Endpoint** → **Summary** wizard, verify the details and click **Done**



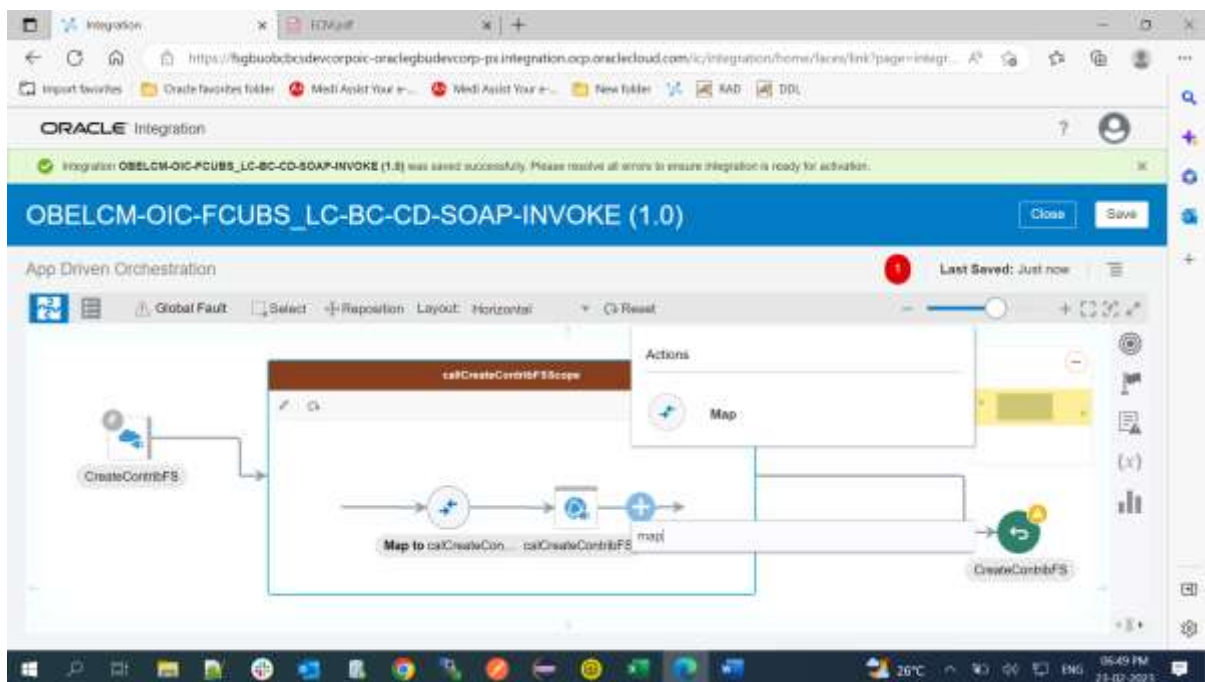
22) Invoke Connection setup completed in Integration for **callCreateContribFS**, you should see below Canvas screen - Click Save



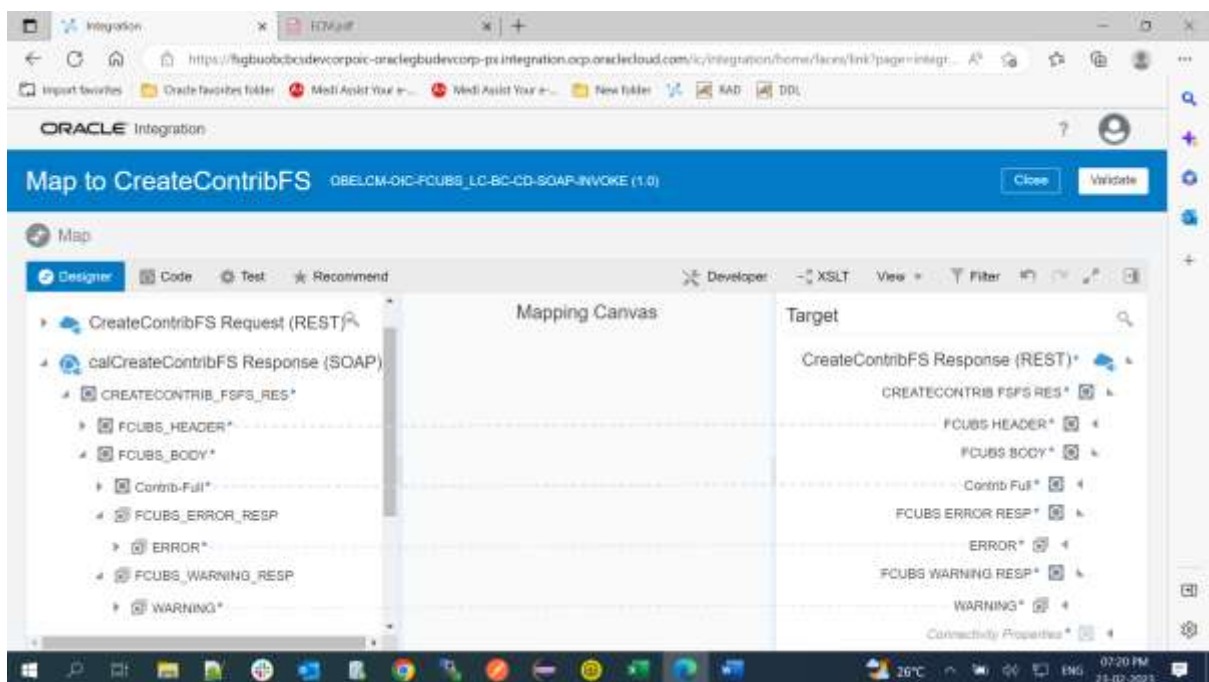
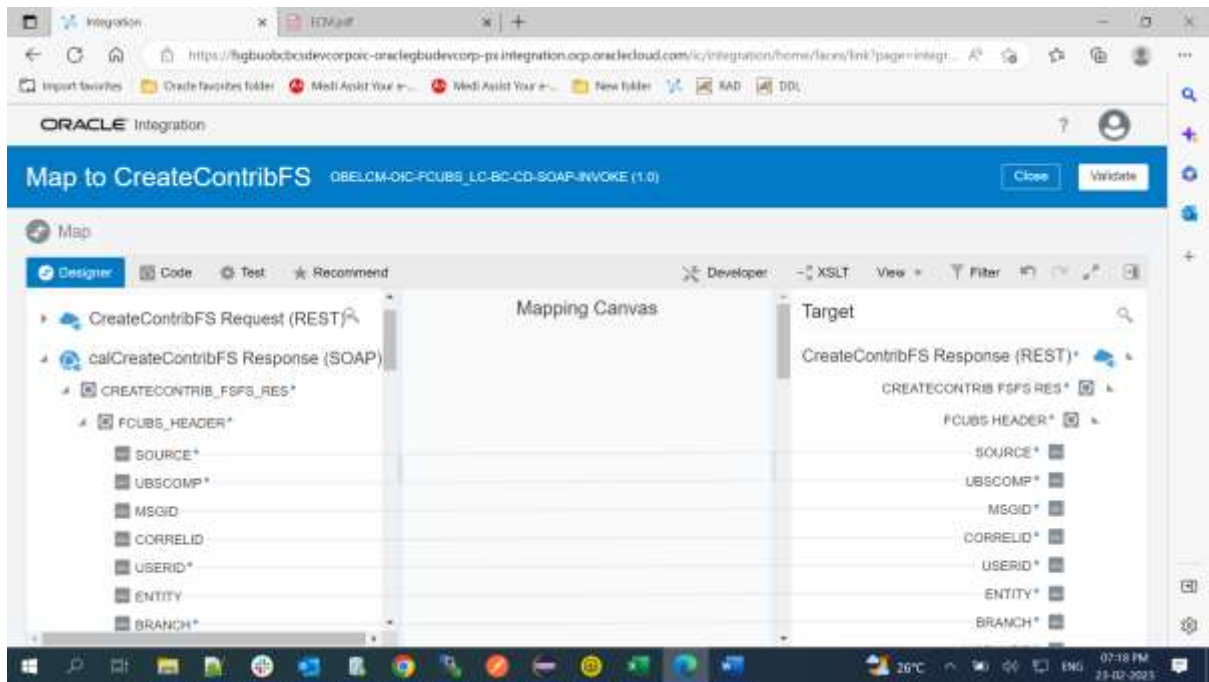
d. Data Transformation - Data Mapping for Request and Response

Mapping Response Data between callCreateContribFSScope (Target Invoke Response) and createContribFS (Source Trigger Response).

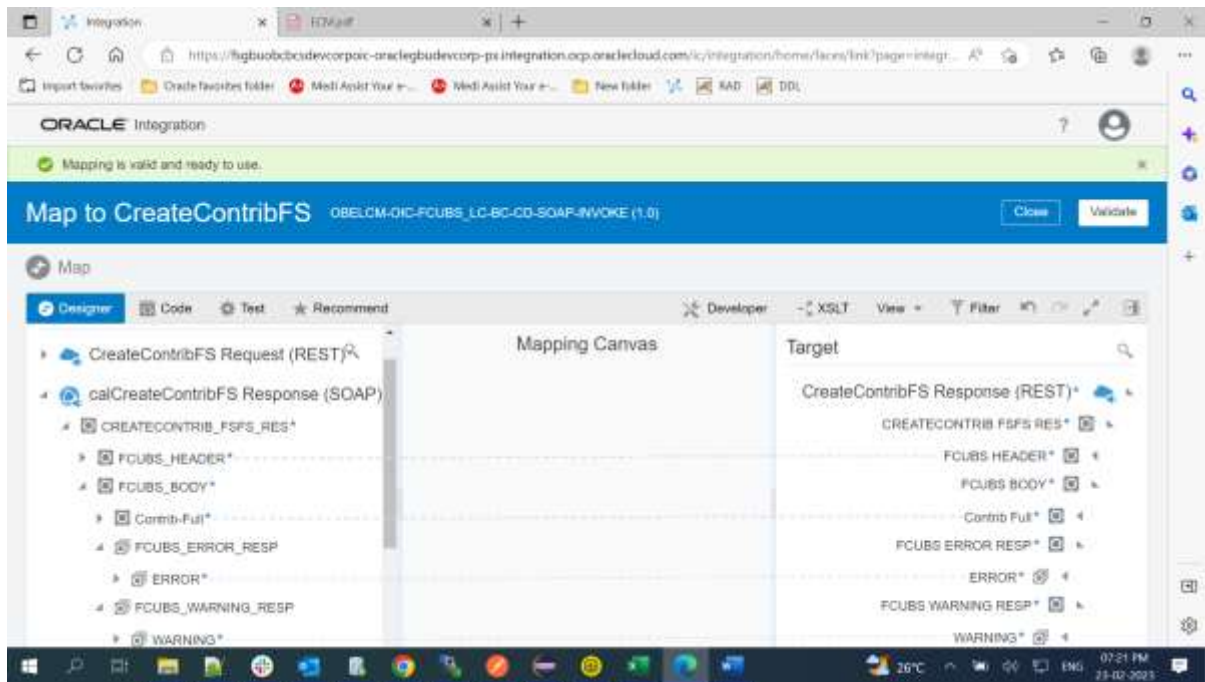
- a) Click + button which is inside the callCreateContribFSScope, next to callCreateContribFS and search for Map component and click Map



- b) In the Mapping Canvas, Mapping needs to be done for **all the Source Field** in **callCreateContribFS Response (SOAP)** to their corresponding **Target Fields** in **CreateContribFS Response (REST)**
- c) In Sources **callCreateContribFS** Response (REST) → click on Response Wrapper, Drag and Drop the Custom Headers from Source to Target **CreateContrib** Response (REST) → Response Wrapper one by one

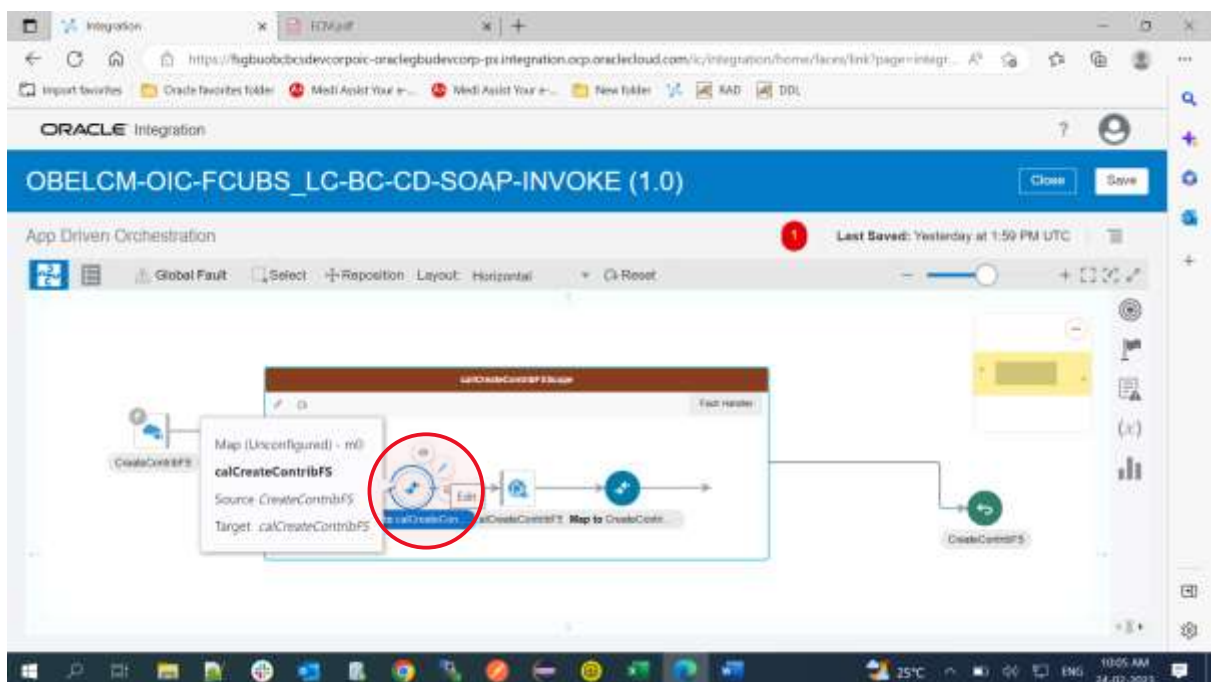
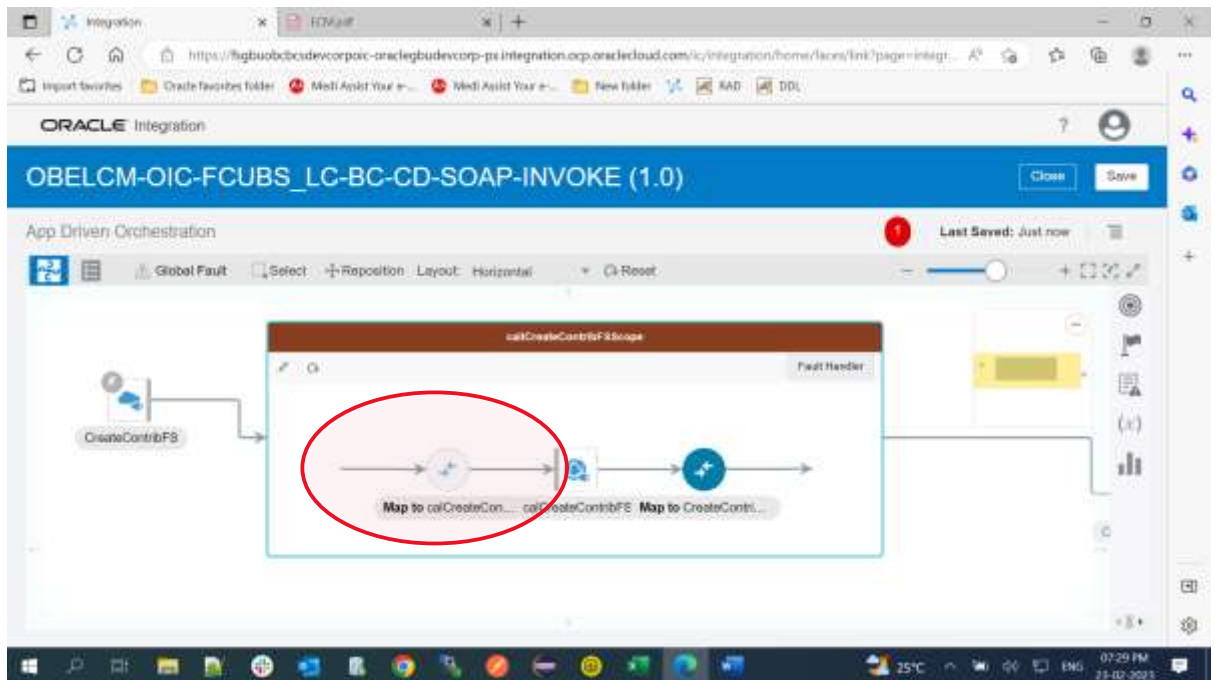


Click Validate and then Click Close

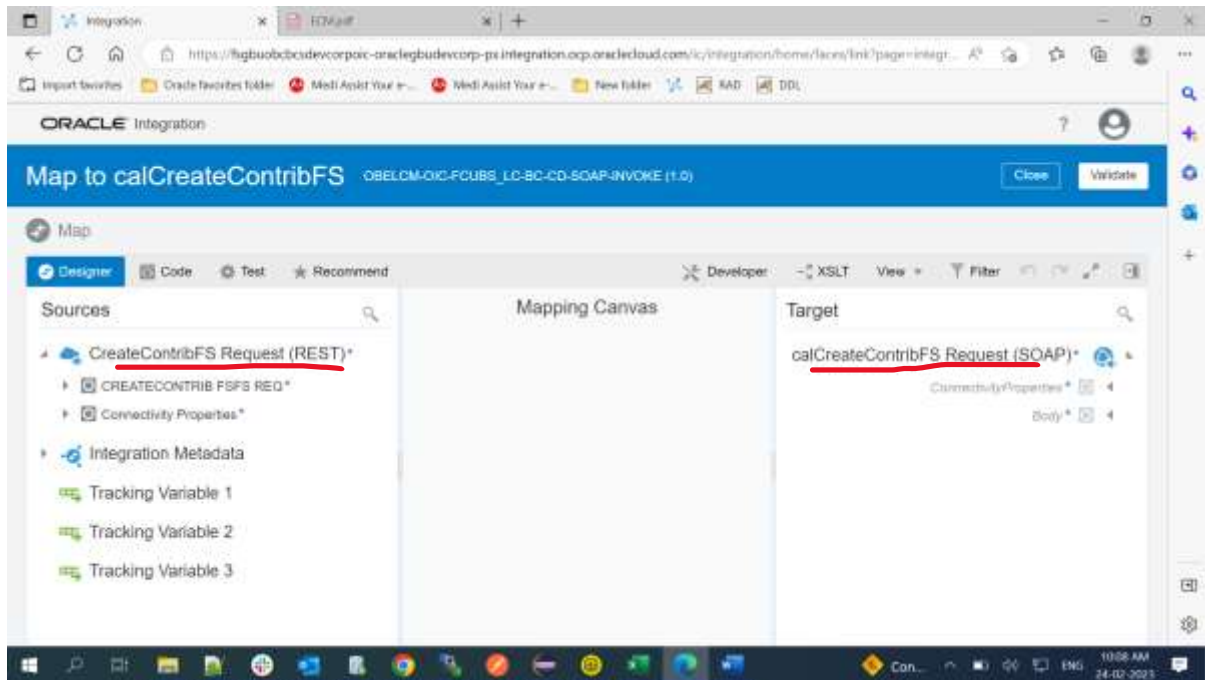


Mapping Request Data between CreateContribFS (Source Trigger Request) and callCreateContribFS (Target Invoke Request)

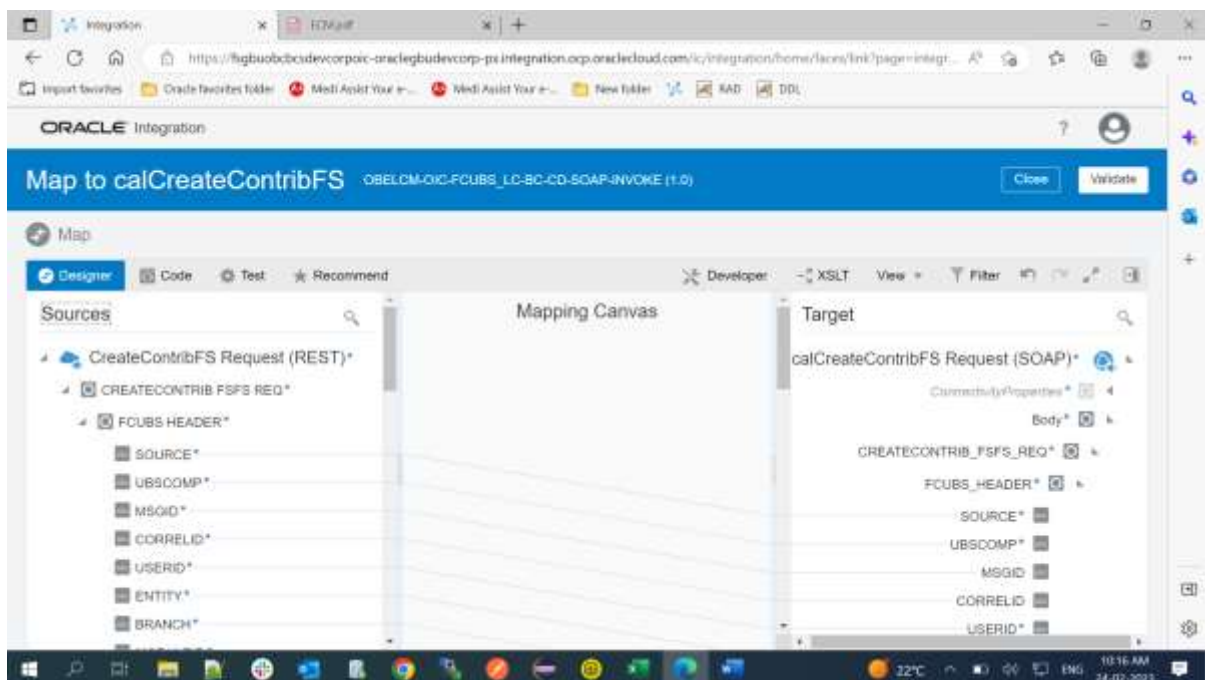
- 1) Click Map to callCreateContribFS which is inside the callCreateContribFSScope

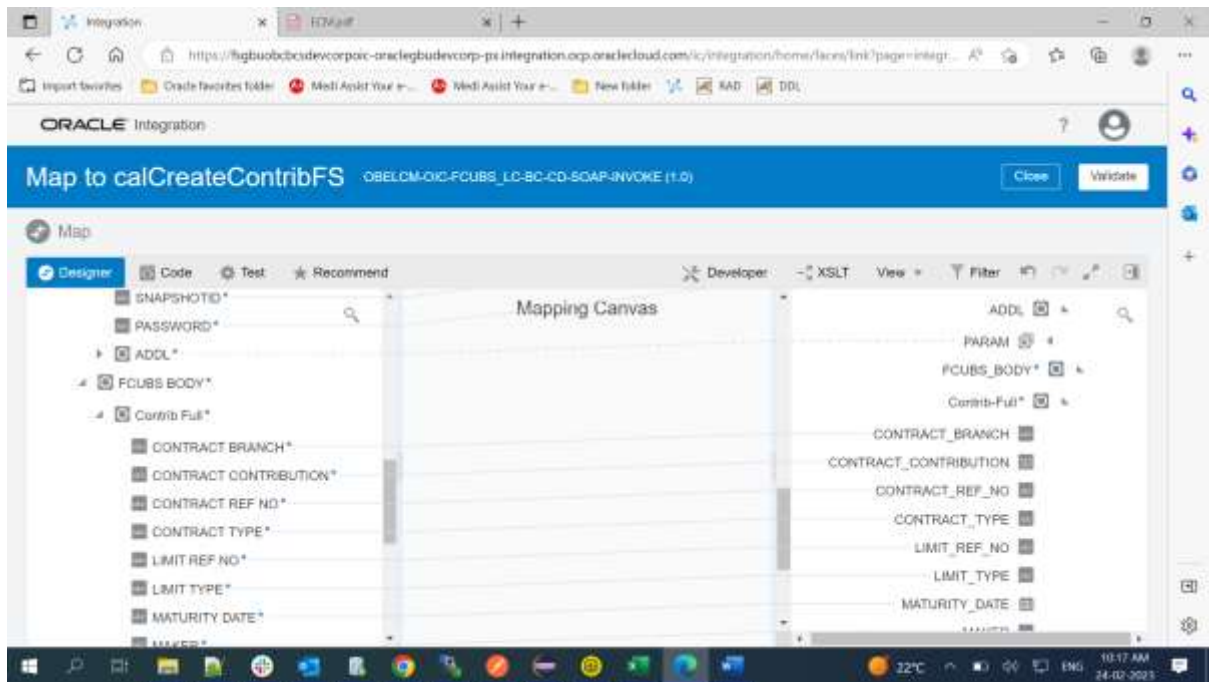


- 2) In the Mapping Canvas, Mapping needs to be done for all the Source Field in CreateContrib Request (REST) to their corresponding Target Fields in callCreateContrib Request (REST)

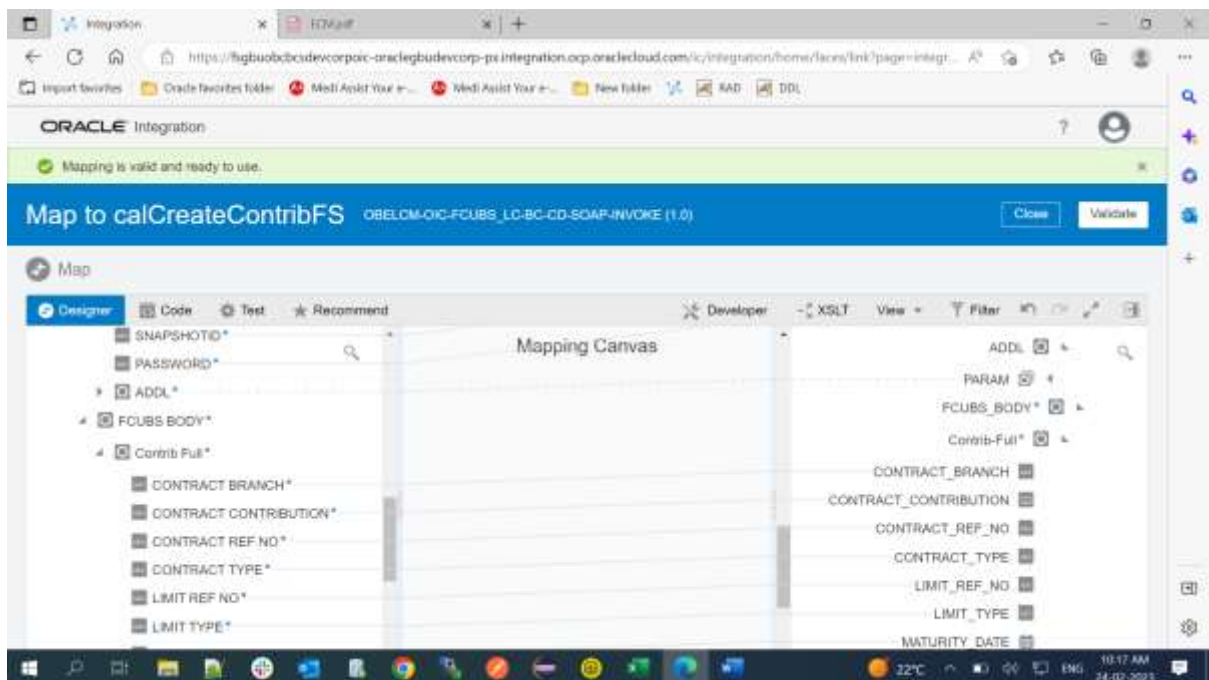


- 3) In Sources CreateContrib Request (REST), Drag and Drop the Custom Headers from Source to Target callCreateContribFS Request (SOAP)





Click Validate and then Click Close



Request Mapping Completed, you should see below Canvas screen - Click Save

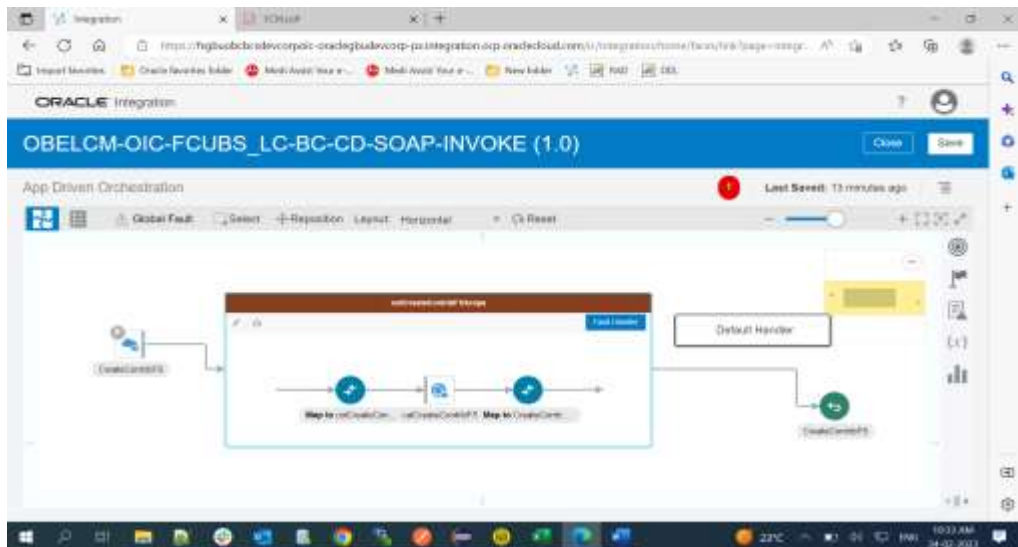


e. Configure Default Handler Fault Handler inside Scope

Fault Handlers	Links	Comments
Default Handler	Refer APIInvocationErr or Handler	Handle different error responses in Default Handler
Global Fault Handler	Refer Global Fault Handler	
Handle Status Code 200 OK	Refer Handle Status Code 200 OK	Handle 200 OK response flow

We can handle Faults occurred in Target Services with a scope using a Fault Handler.

- 1) Click on Fault Handler and then Click REST: APIInvocationError

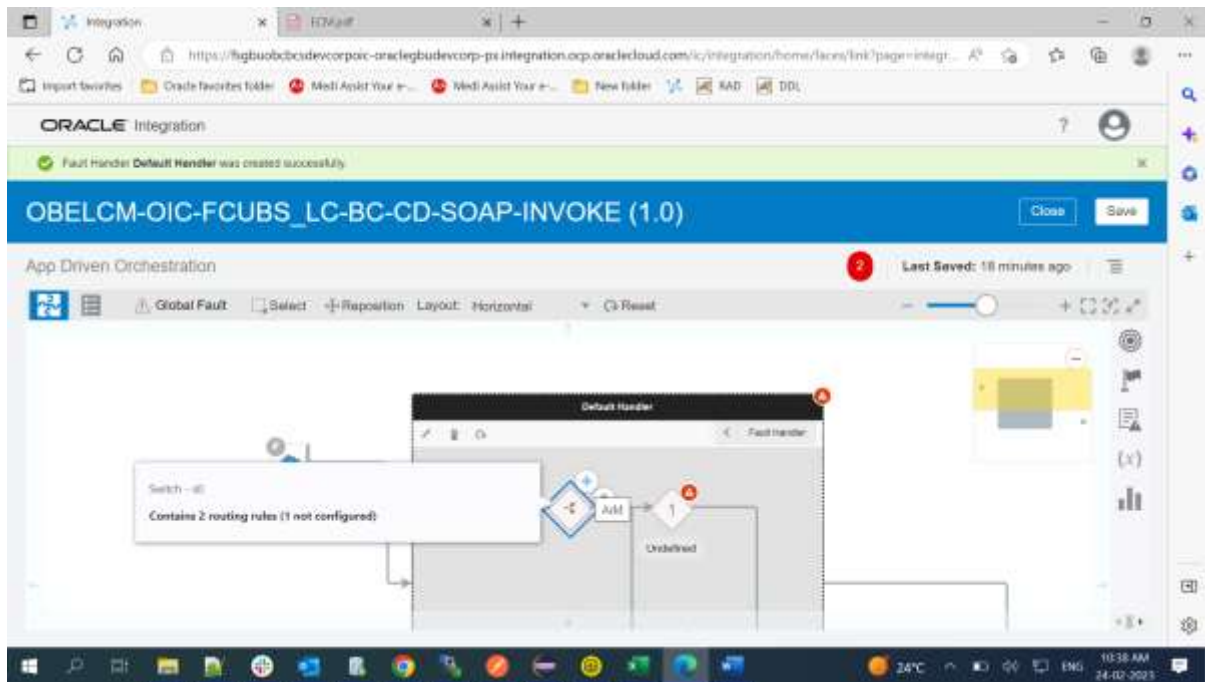


2) Click on + Button, search for Scope and select the Scope component



3) Click on Switch case component and click + button

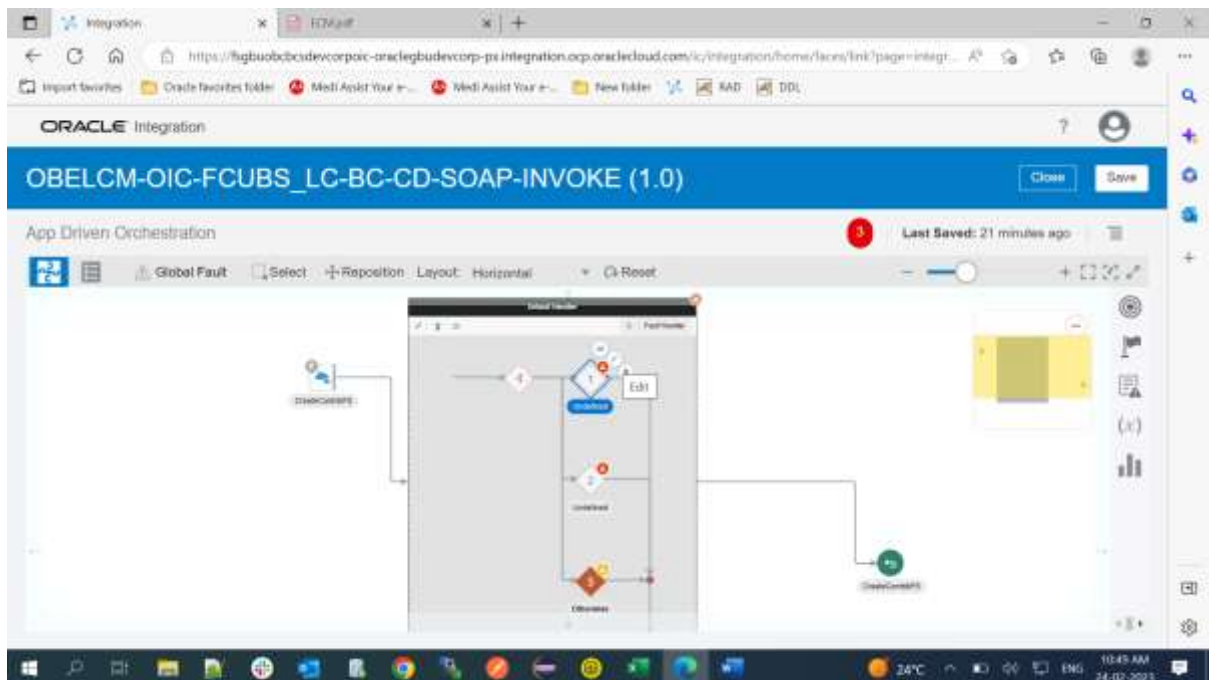
Note: you can handle multiple status codes or error Codes by clicking the + button



- 4) Now, we can handle following error codes and error statuses in the **APIInvocationError** Fault Handler

Http Status Code	Http Status	Response Body Field Name	Field Value	Case Name	Handled in Fault Handler Name/Scope
400	Bad Request	accountingMessageStatus	FAILURE	1	APIInvocationHandler Fault Handler
500	Internal Server error	Path	/fcubs-ext-accounting-services/service/v1/Accounting	2	APIInvocationHandler Fault Handler
200	OK			Otherwise	callCreateContrib Scope

4a. Click on the first case (1) (Handle Bad Request) and click the Pencil Edit button



4c. Click on \$CurrentFaultError and click APIInvocationError. Select the fieldname "**errorCode**" and click on move button.

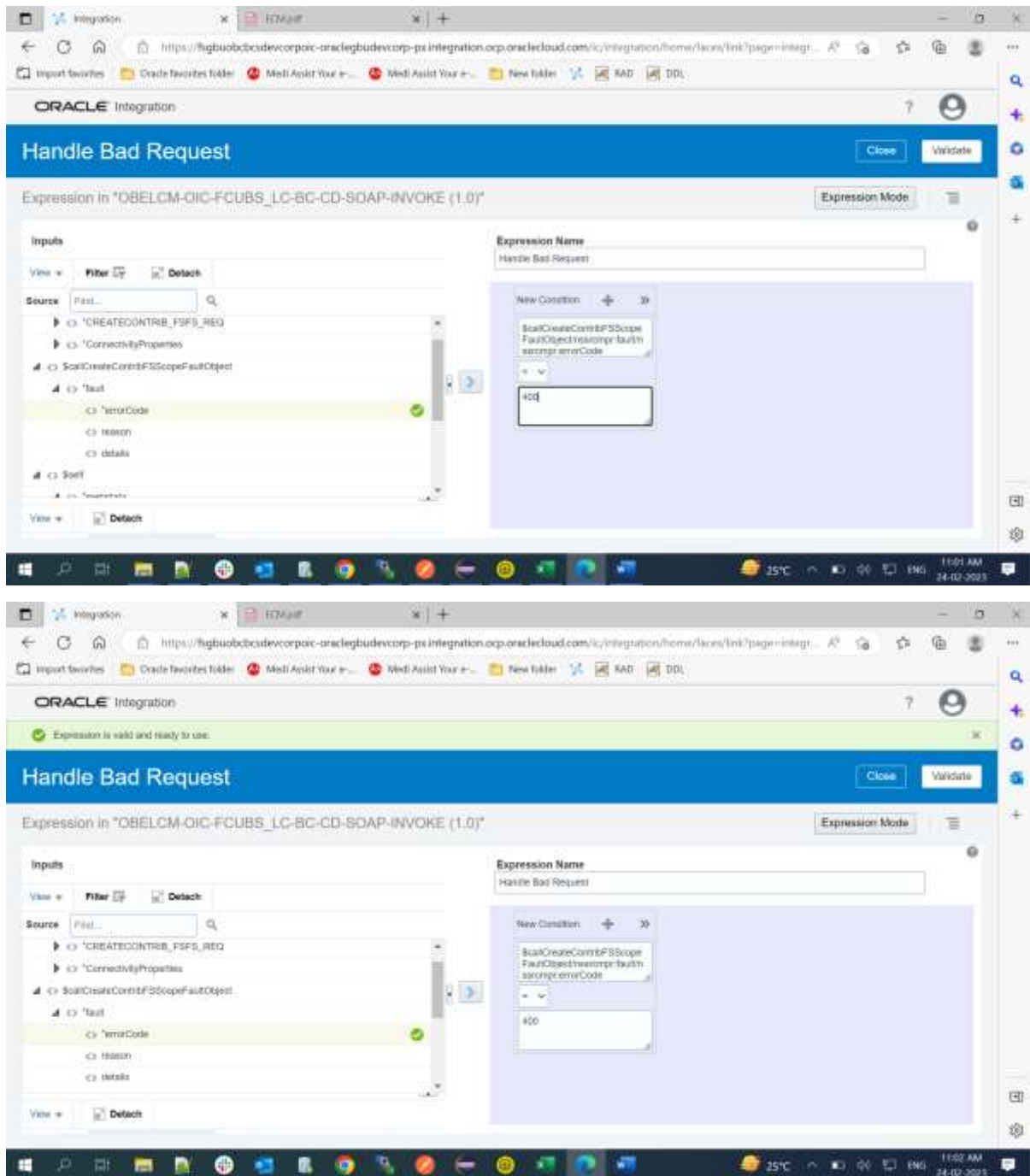
Provide the following details for the condition

Expression Name : Handle Bad Request

operator: =

Value: 400

Click Validate and Click Close.



4d. Click on the Second case 2 (Handle Internal Server Error) and click the Pencil Edit button



4e. Click on `$CurrentFaultError` and click `APIInvocationError`. Select the fieldname "errorCode" and click on move button

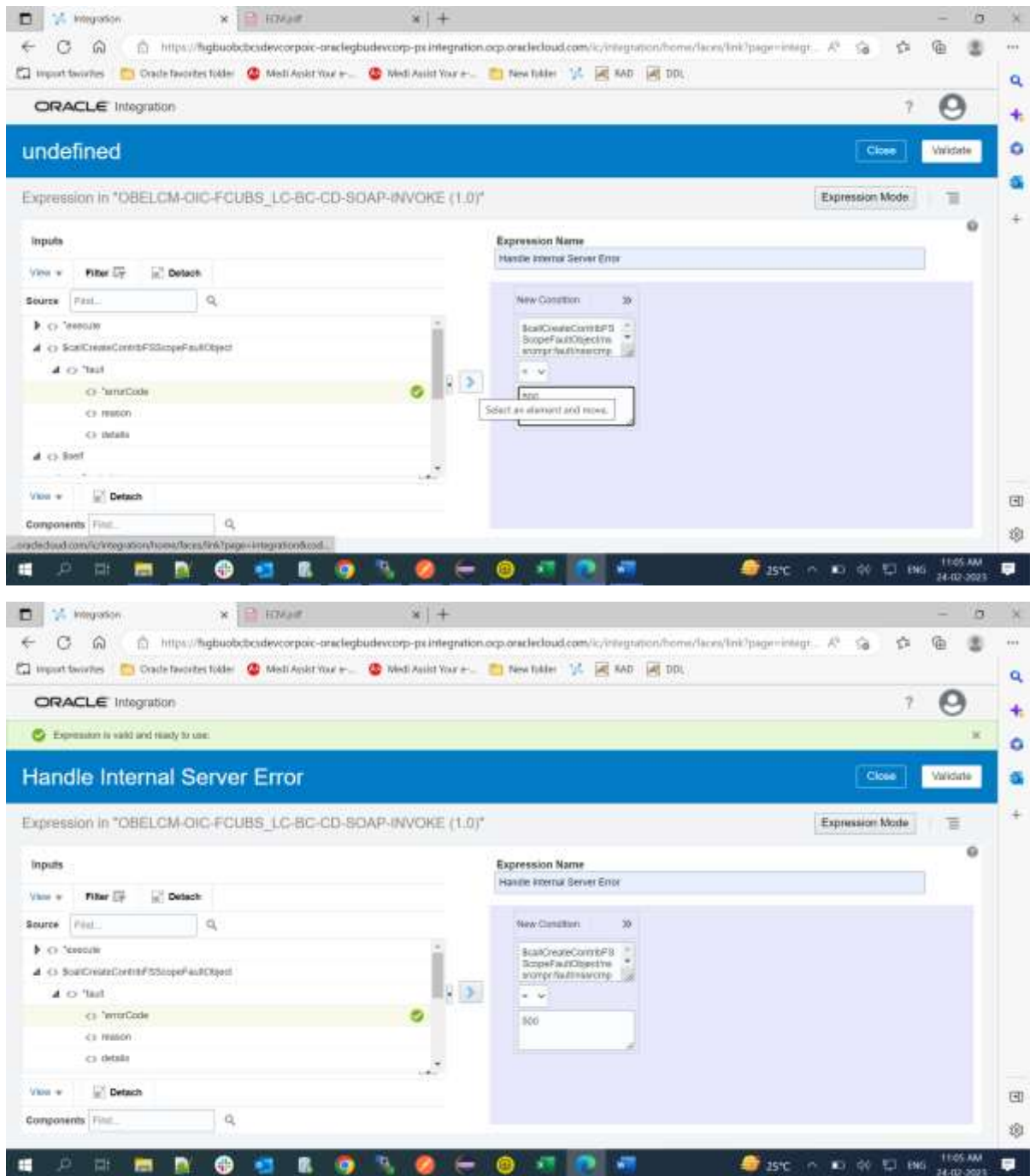
Provide the following details for the condition

Expression Name: Handle Internal Server Error

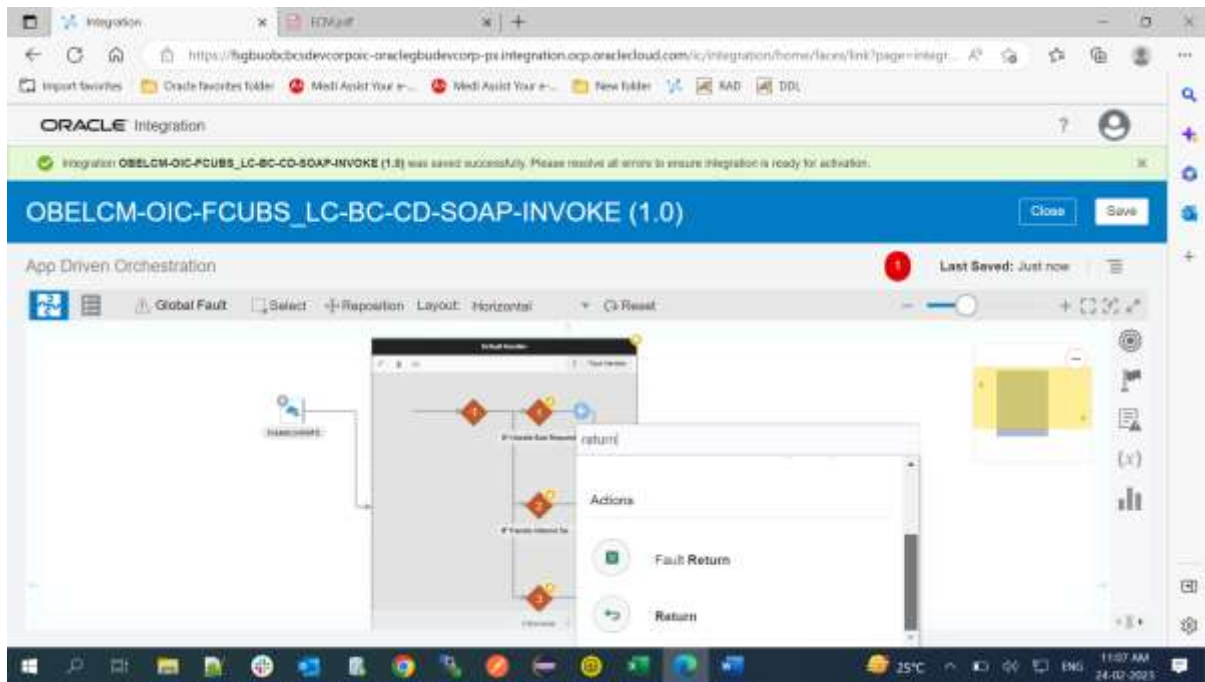
operator: =

Value: 500

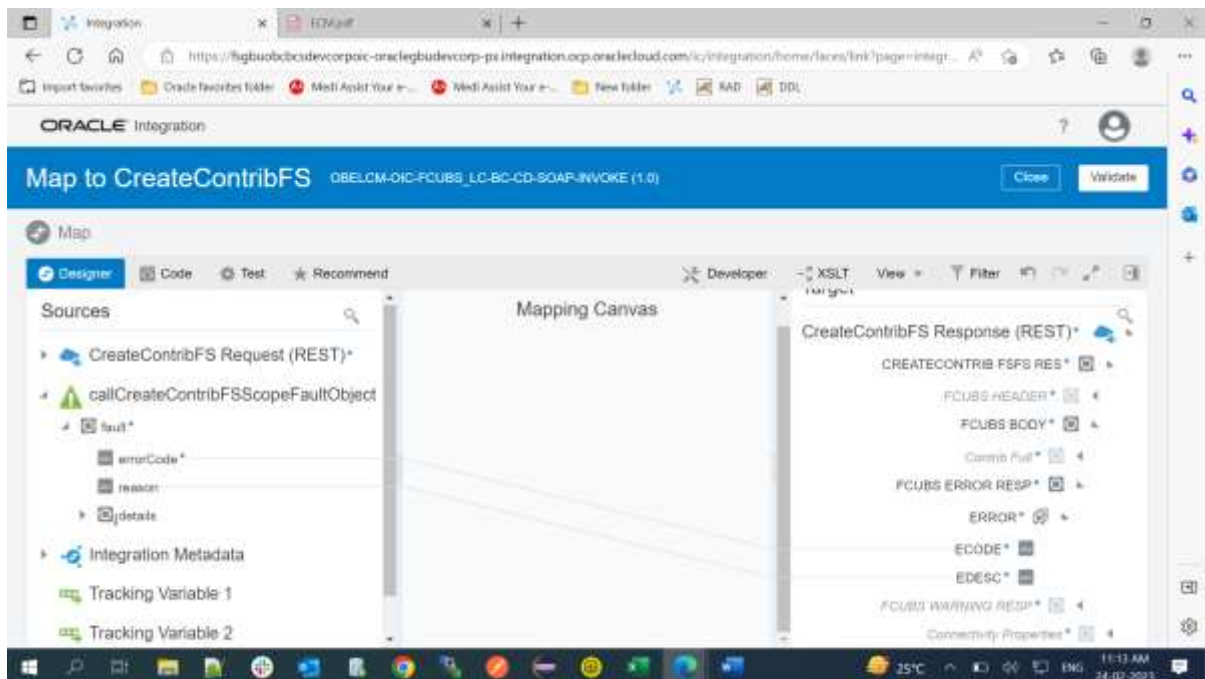
Click Validate and Click Close.



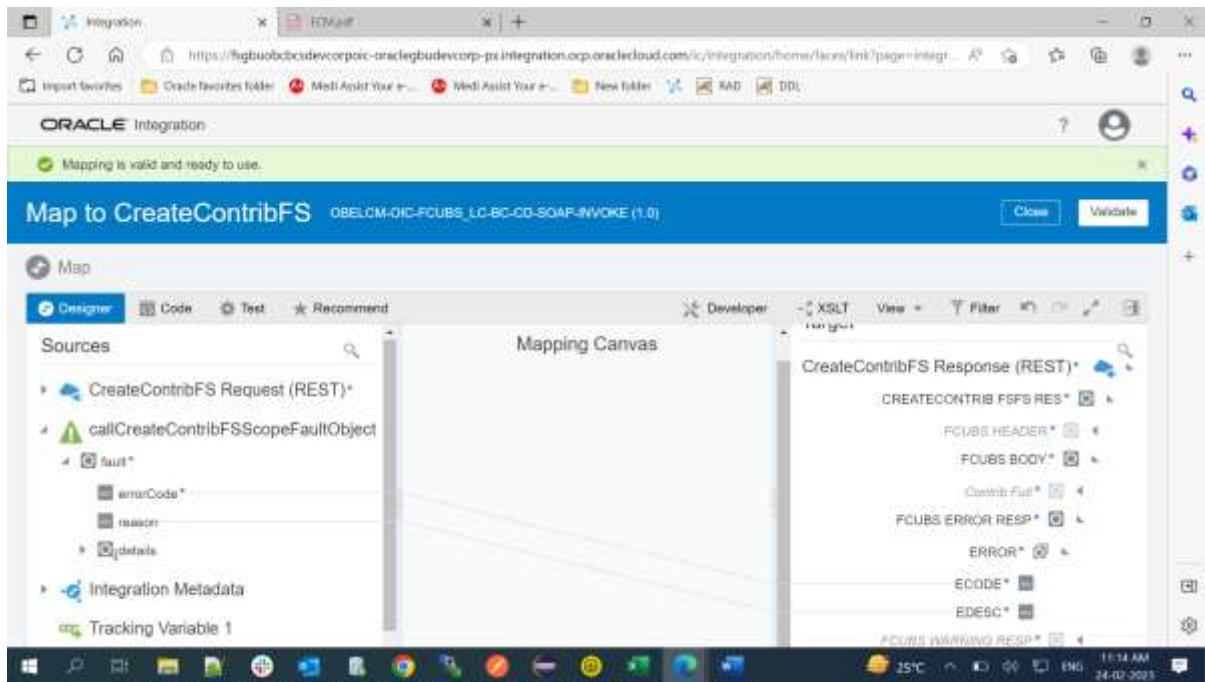
4f. Hover the mouse over the line which is coming out of Case 1 (IF Handle Bad Request) then Click on + button



4g. Click on **Map to CreateContrib**, Drag and Drop the Field names from Source to its Target Field Names as shown below

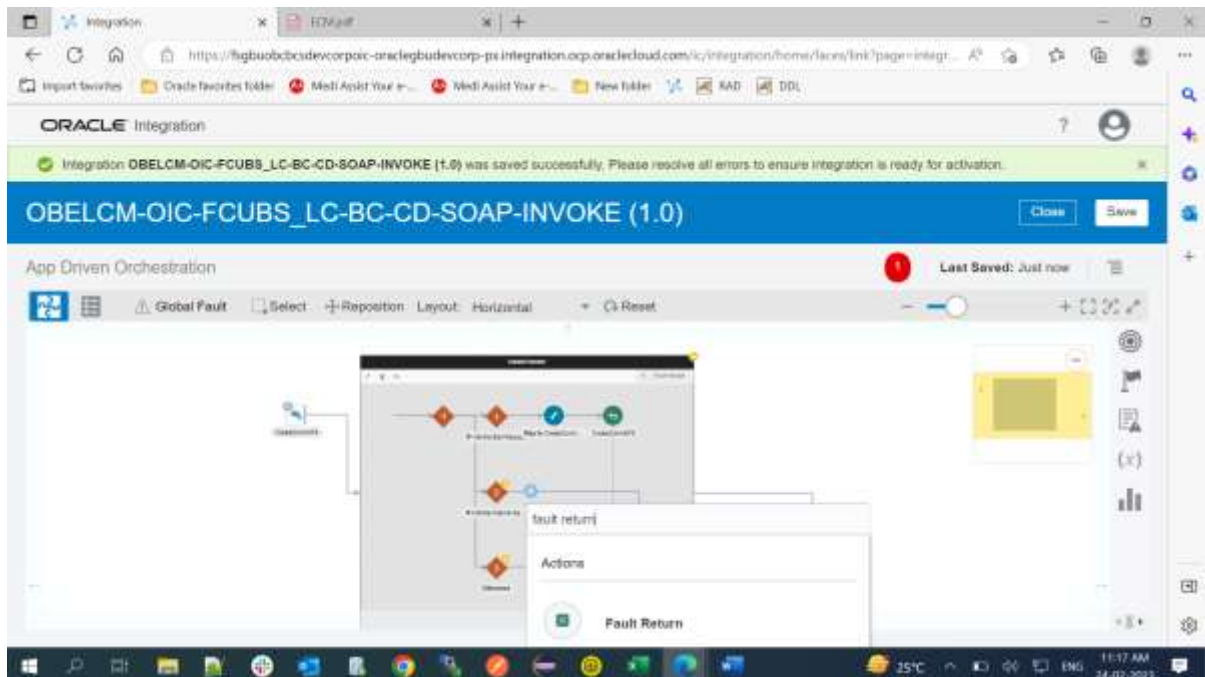


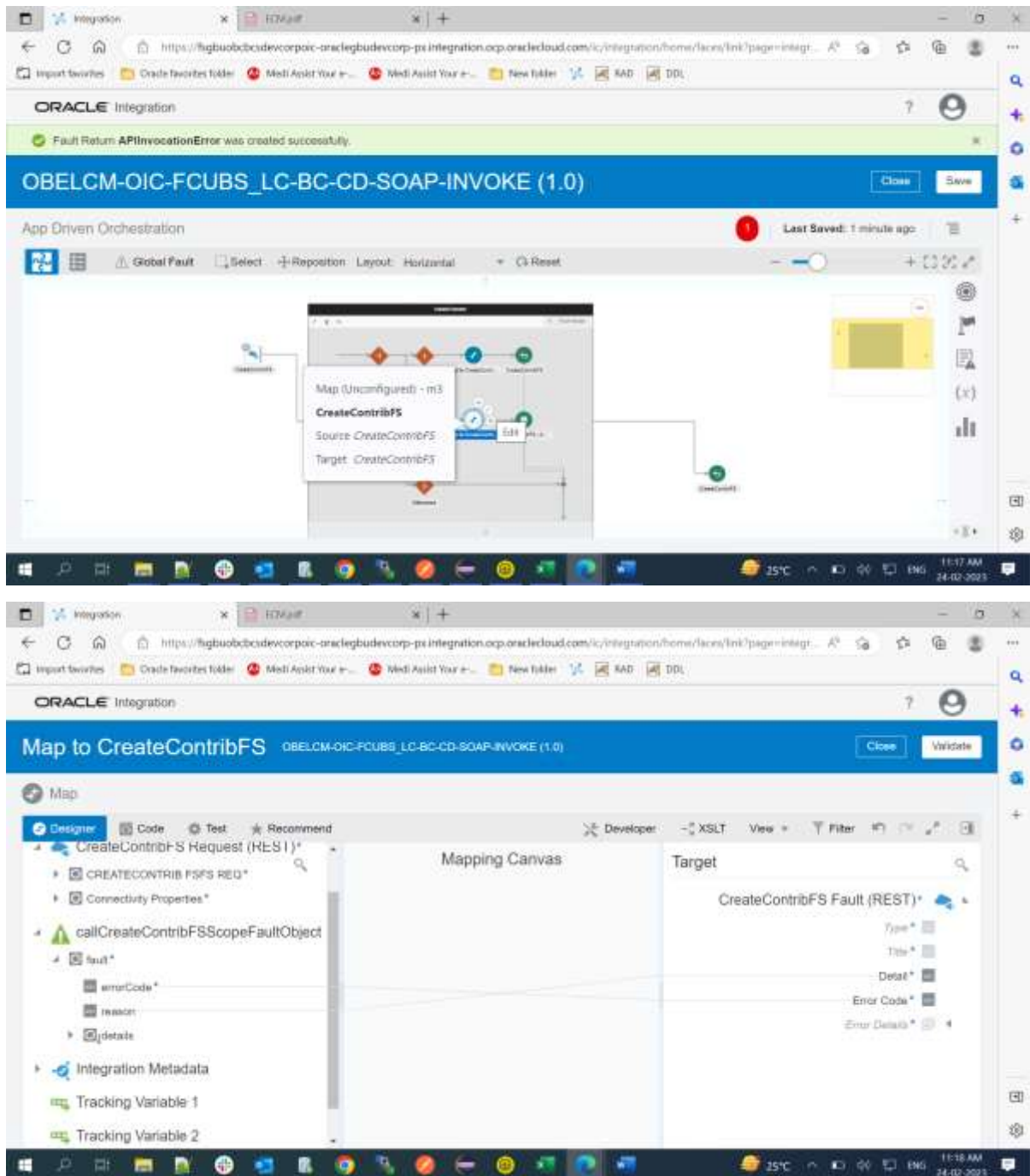
4h. Click Validate and Close



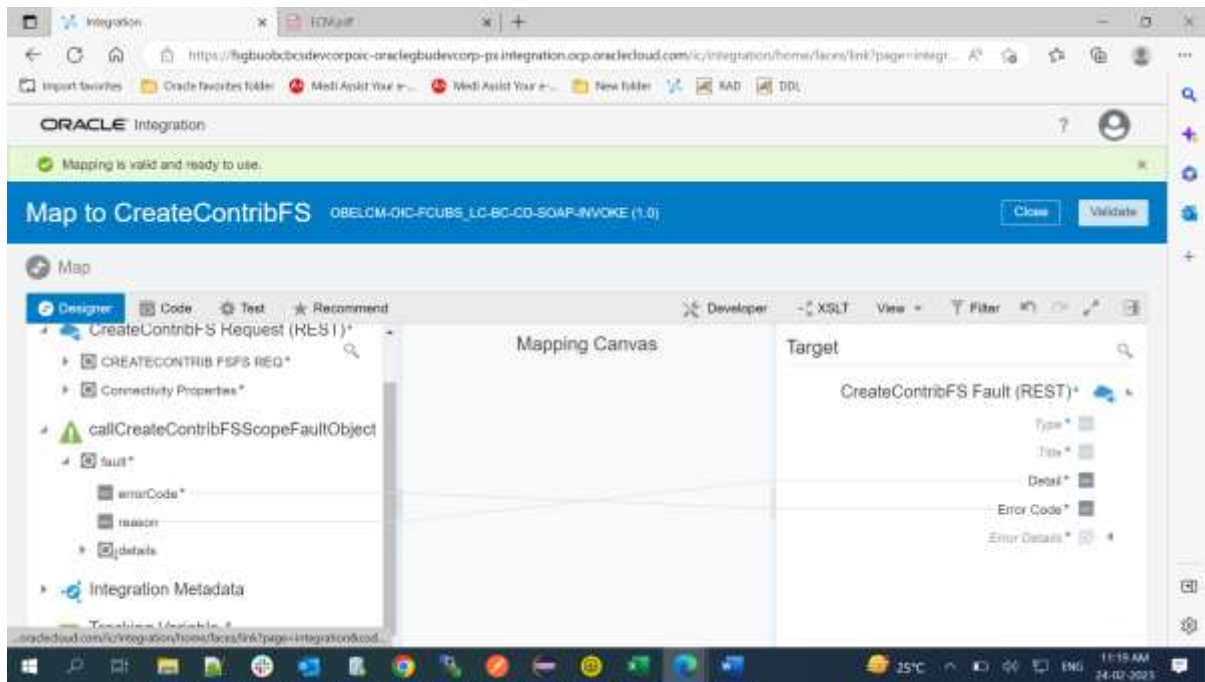
4i. Hover the mouse over the line which is coming out of Case 2 (IF Handle Internal Server Error) then Click on + button and type **Fault Return** and click **Fault Return** component

Click on **Map to CreateContrib**, Drag and Drop the Field names from Source to its Target Field Names as shown below





4j. Click Validate and Close

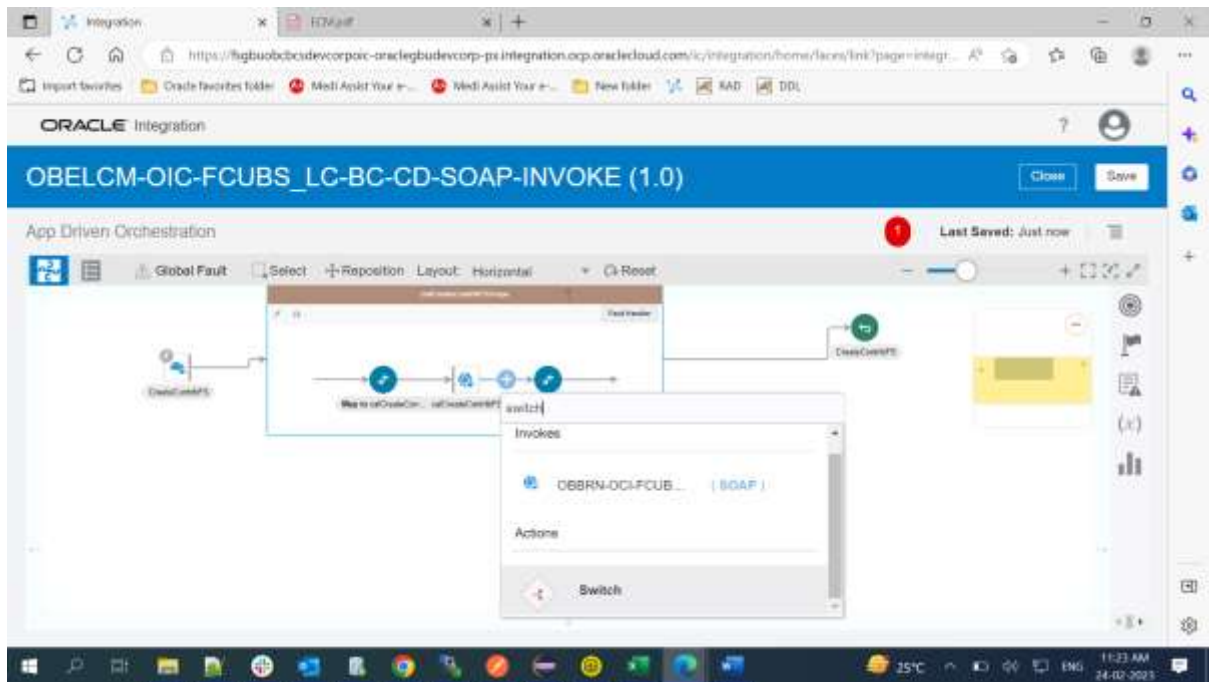


Handle Status Code 200 OK

4k. Click < button which is before Fault Handler



4l. Click + button between the **callCreateContribing** and **Map to CreateContribing**, search switch and click Switch component



4m. Click on the first case (1) (**Handle AccountingMessageStatus FAILED**) and click the Pencil Edit button



4n. Click on \$callCreateContribing and click executeResponse->responseWrapper. Select the

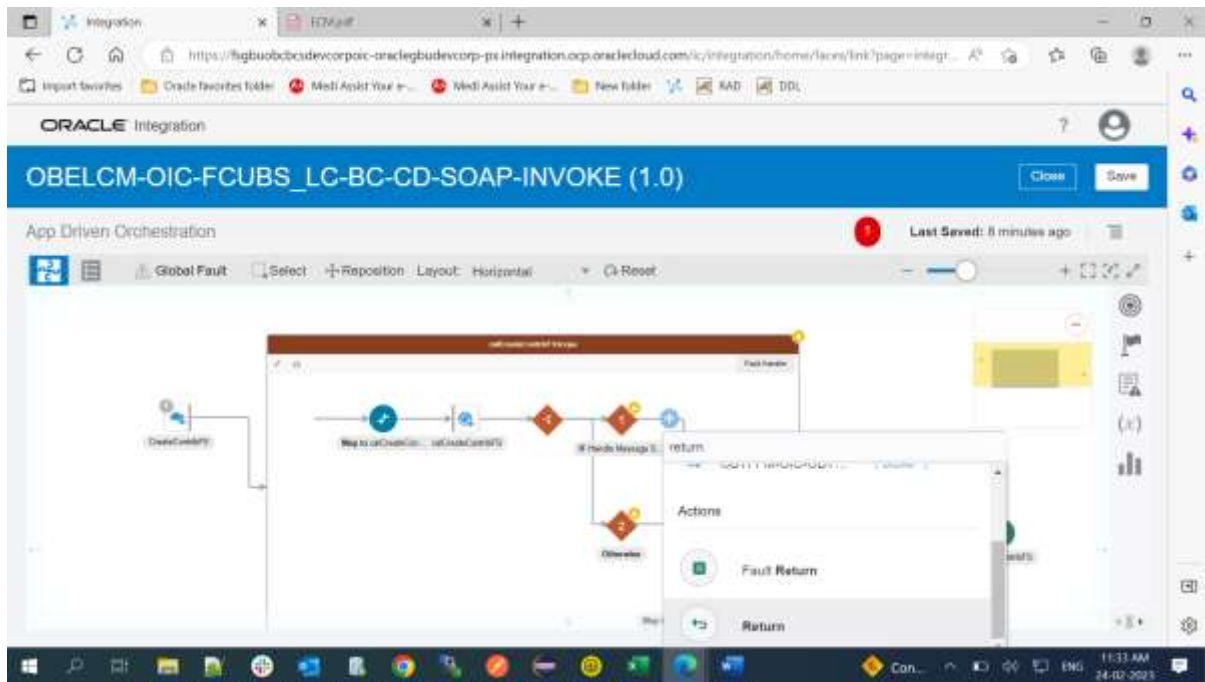


fieldname "**accountingMessageStatus**" and click on

Provide the following details for the condition

Expression Name: Handle AccountingMessageStatus FAILED

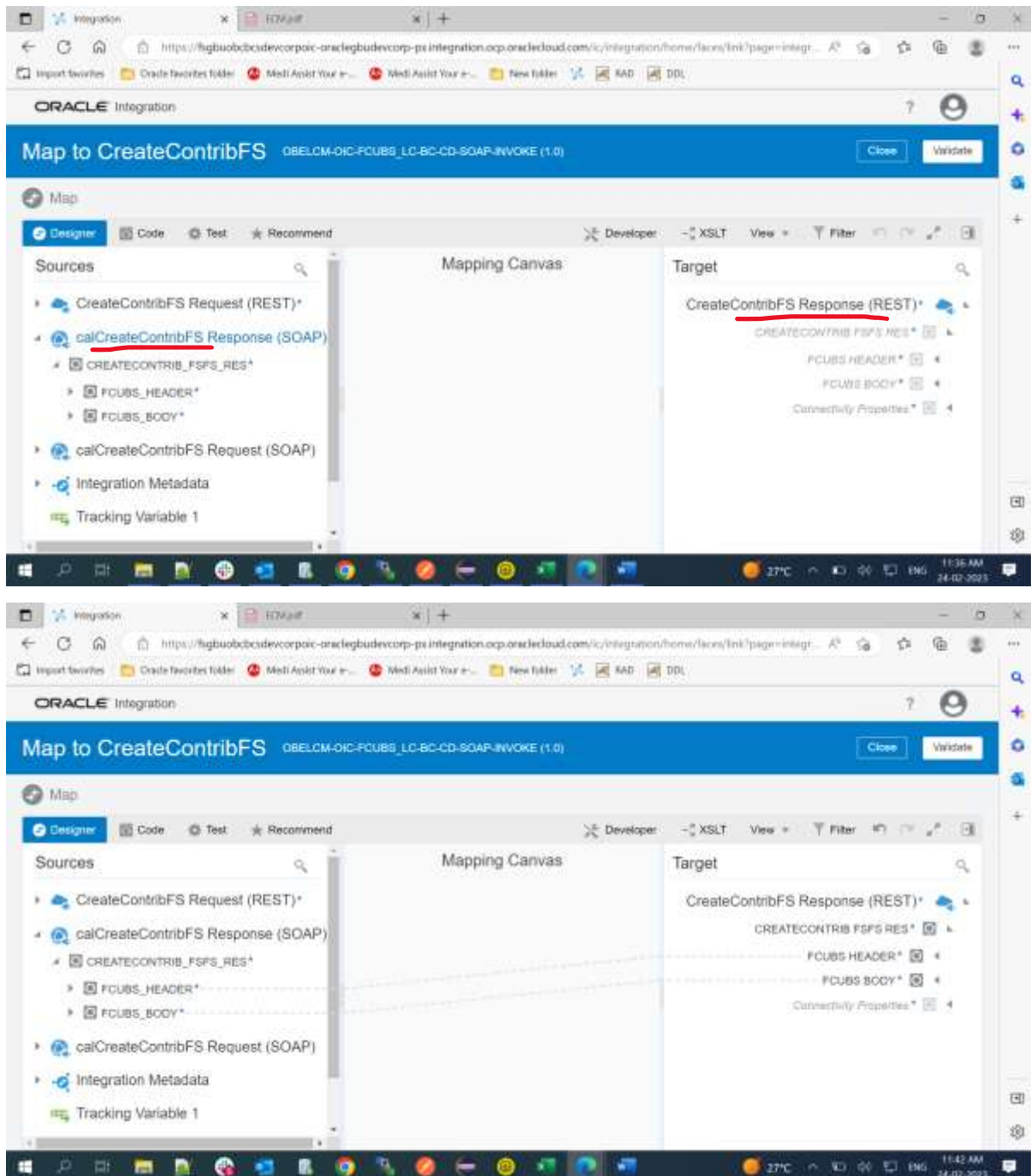
operator: =



4p. Click on **Map to CreateContrib**,



4r. Drag and Drop the Field names from Source to its Target Field Names as shown below

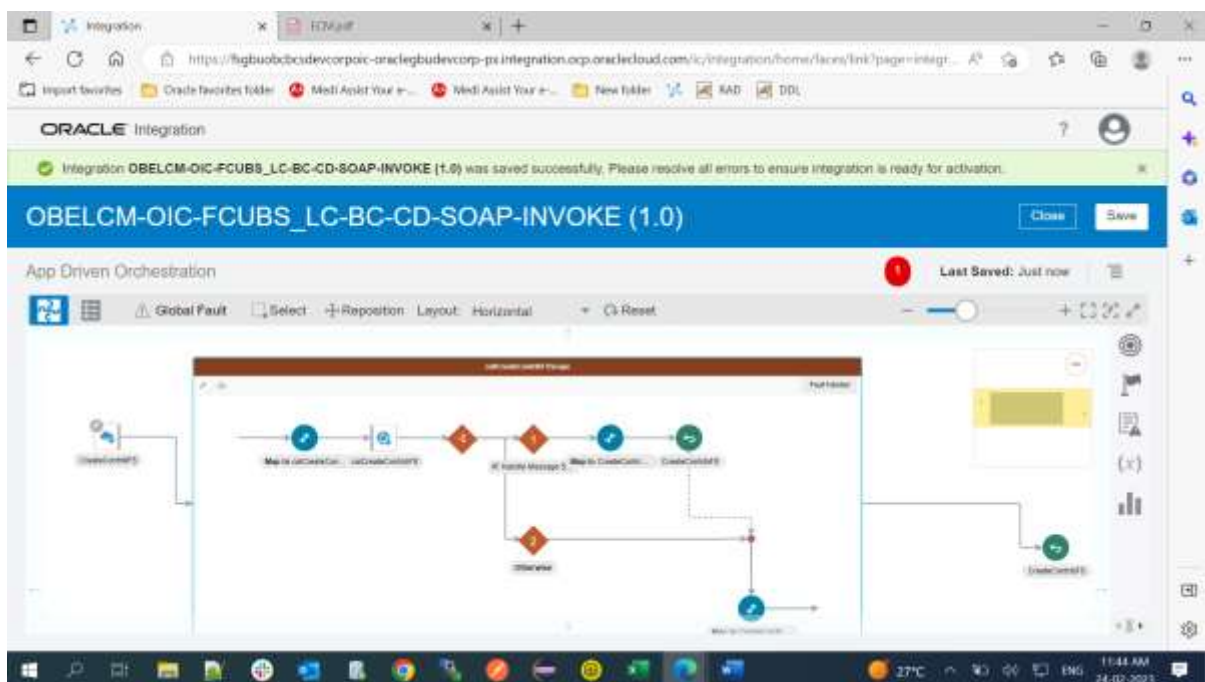


Click Validate and Close

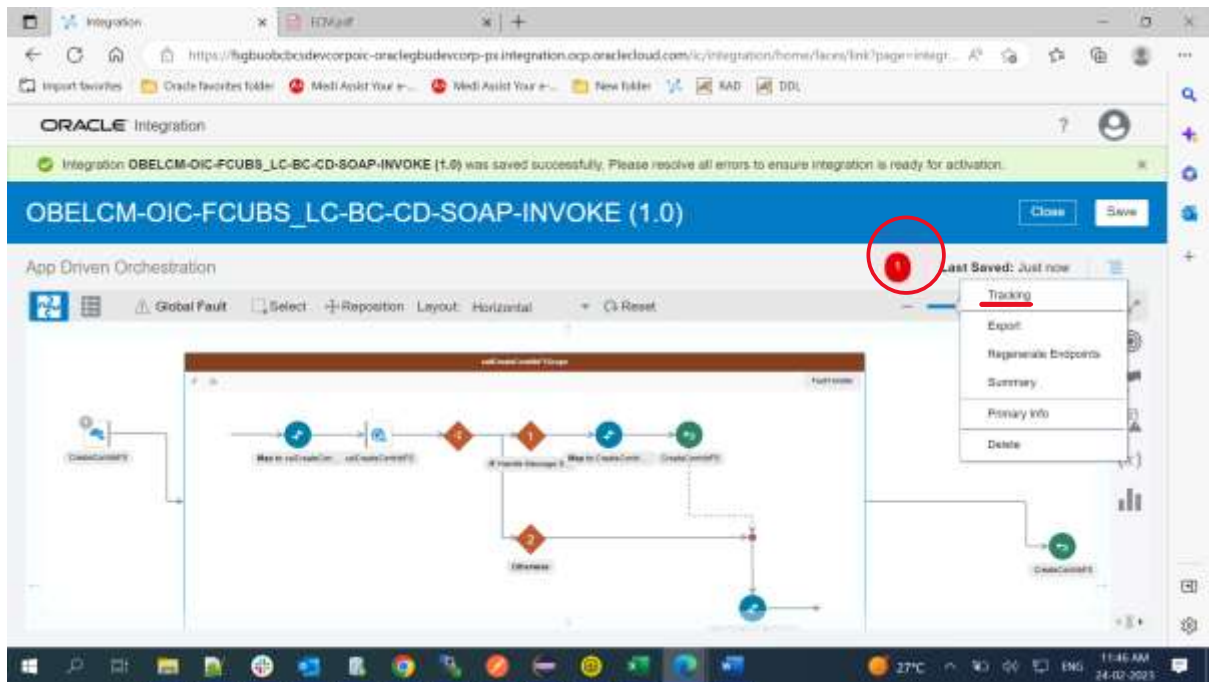
5. Save And Activate Integration



Click Save



you see one 1 Red Alert as shown below - We must add a tracking variable or combination of 2 or more variable. Click Tracking

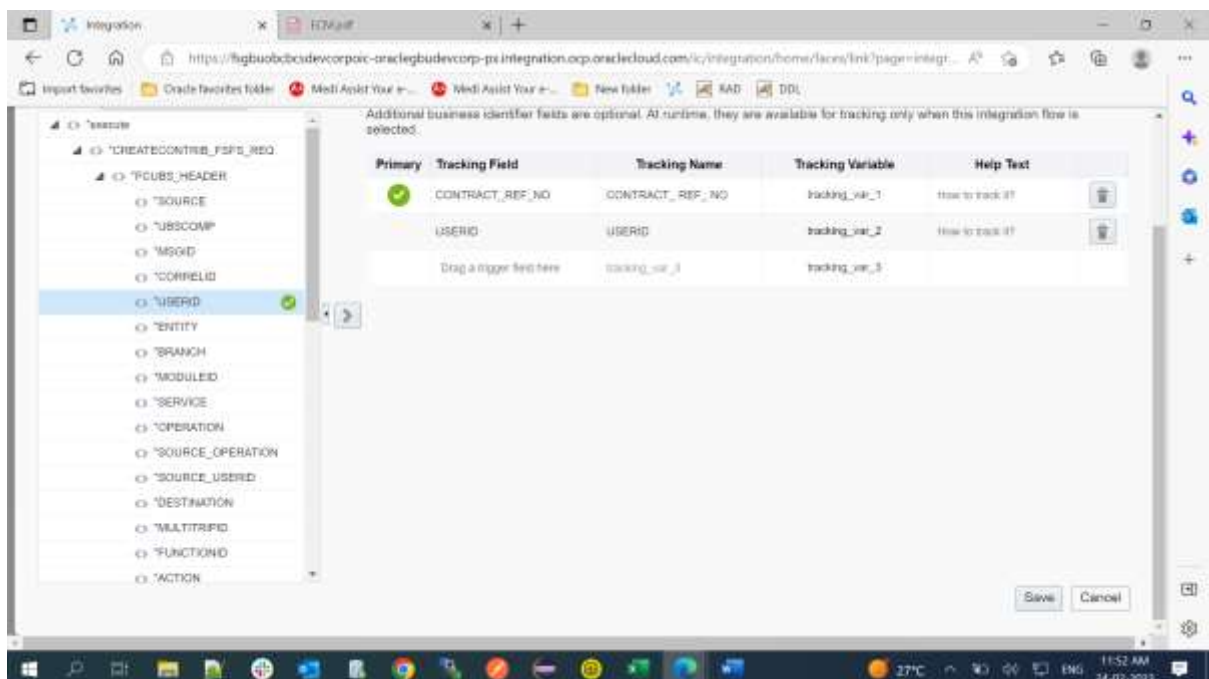


Drag and Drop fields from Request payload/Headers to the table shown below or select the



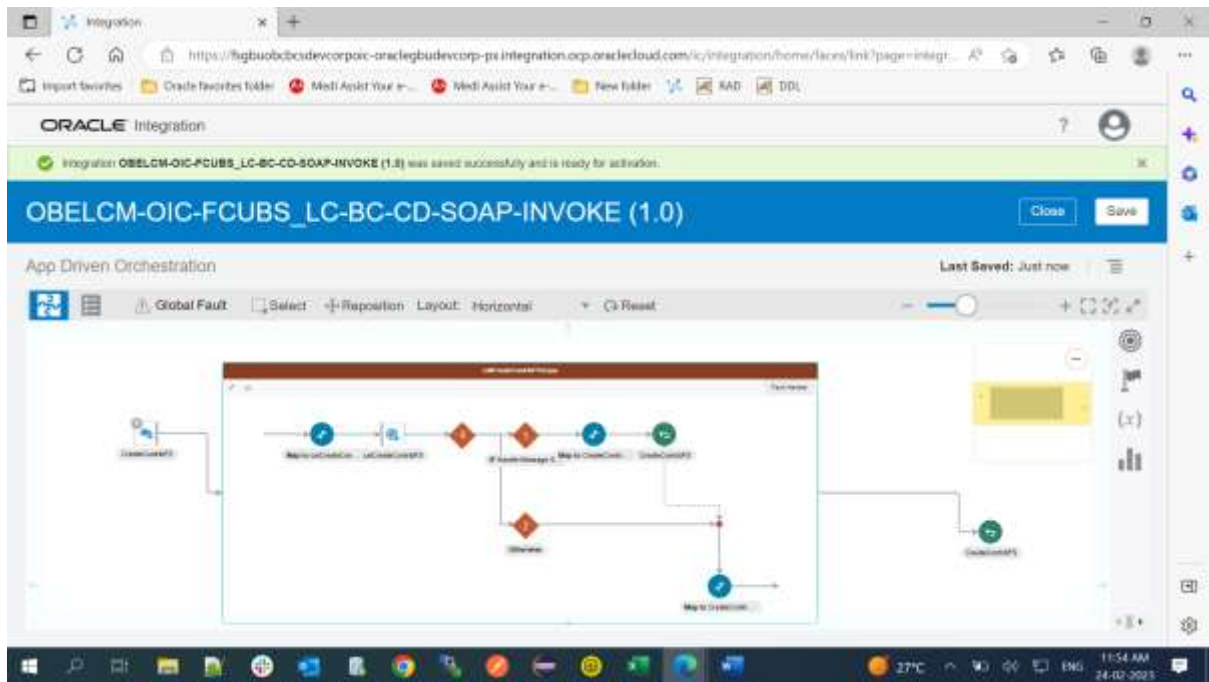
field and click

and click Save



Save And Activate Integration

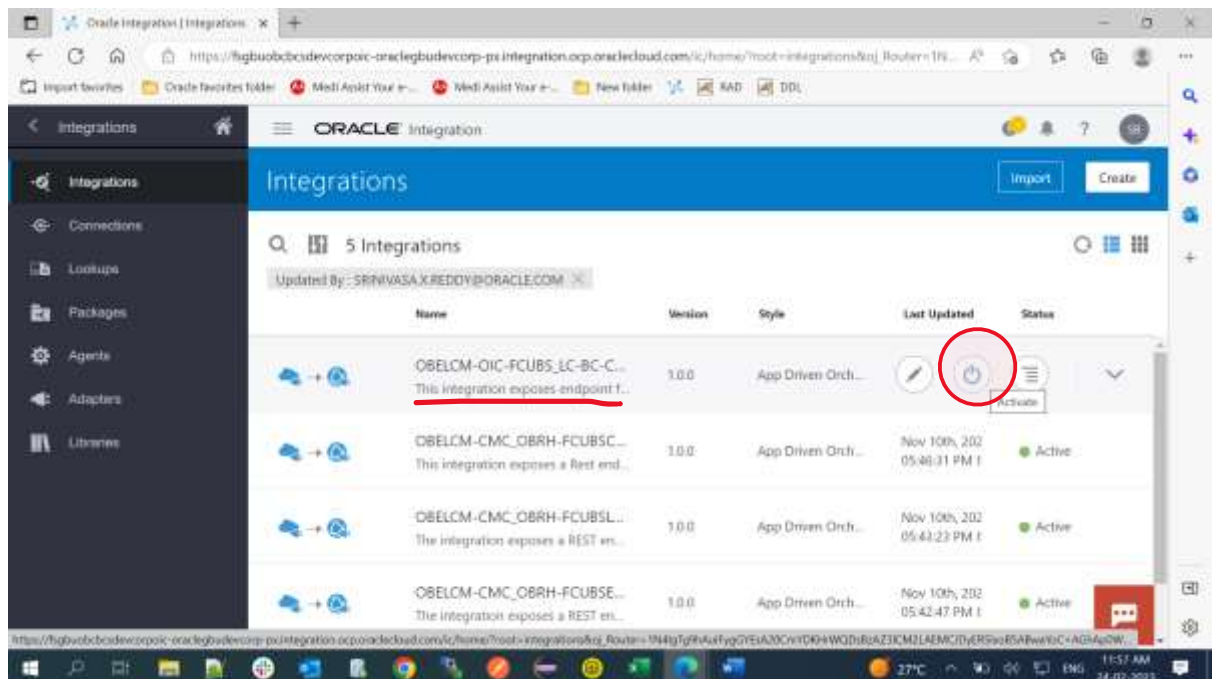
Click Save



- 1) In the Integration screen, first activate your integration by clicking on the

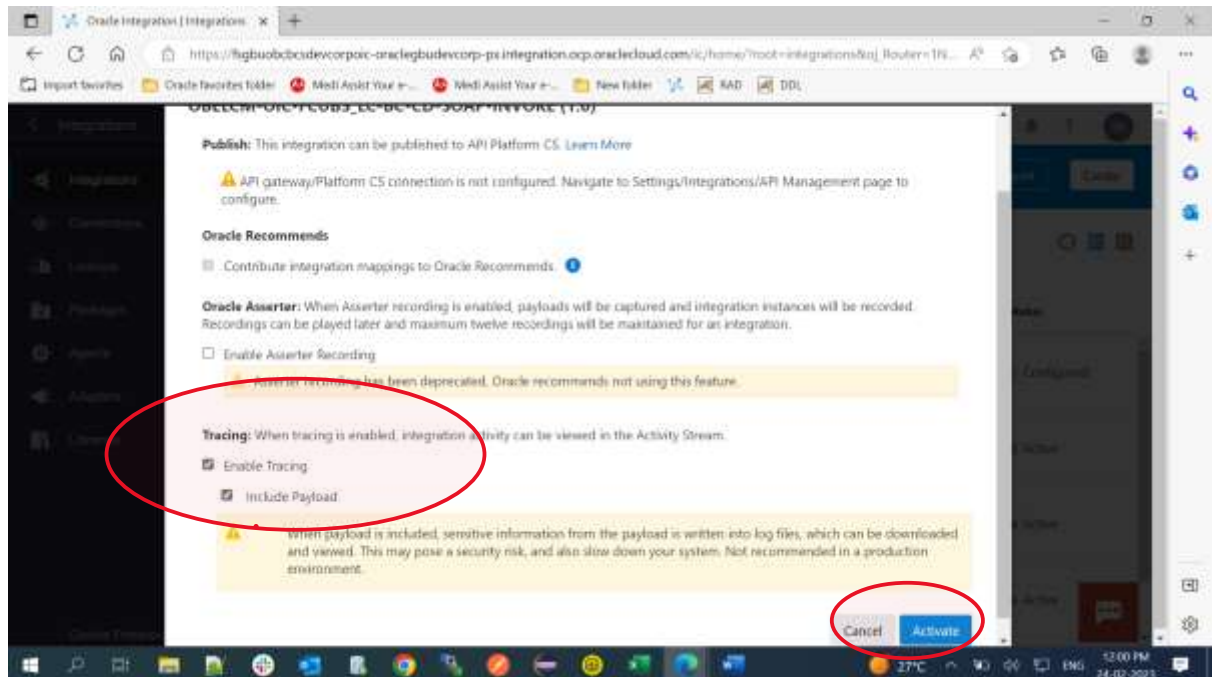


button shown below

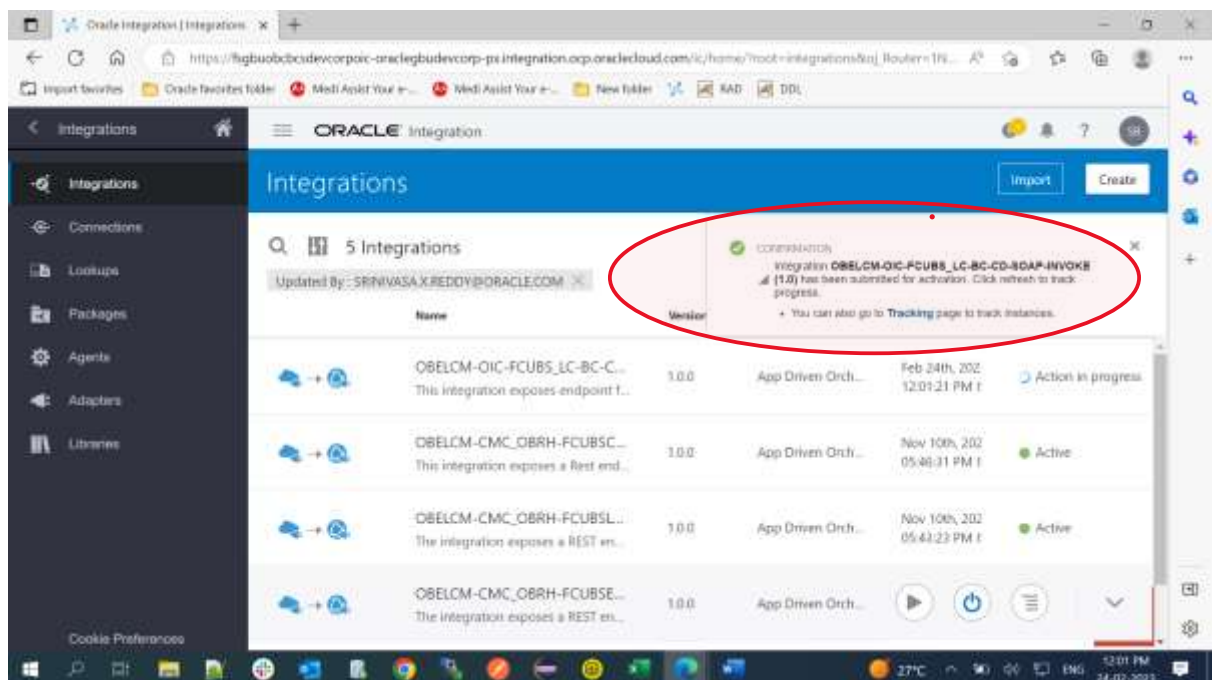


6. Enable Tracing

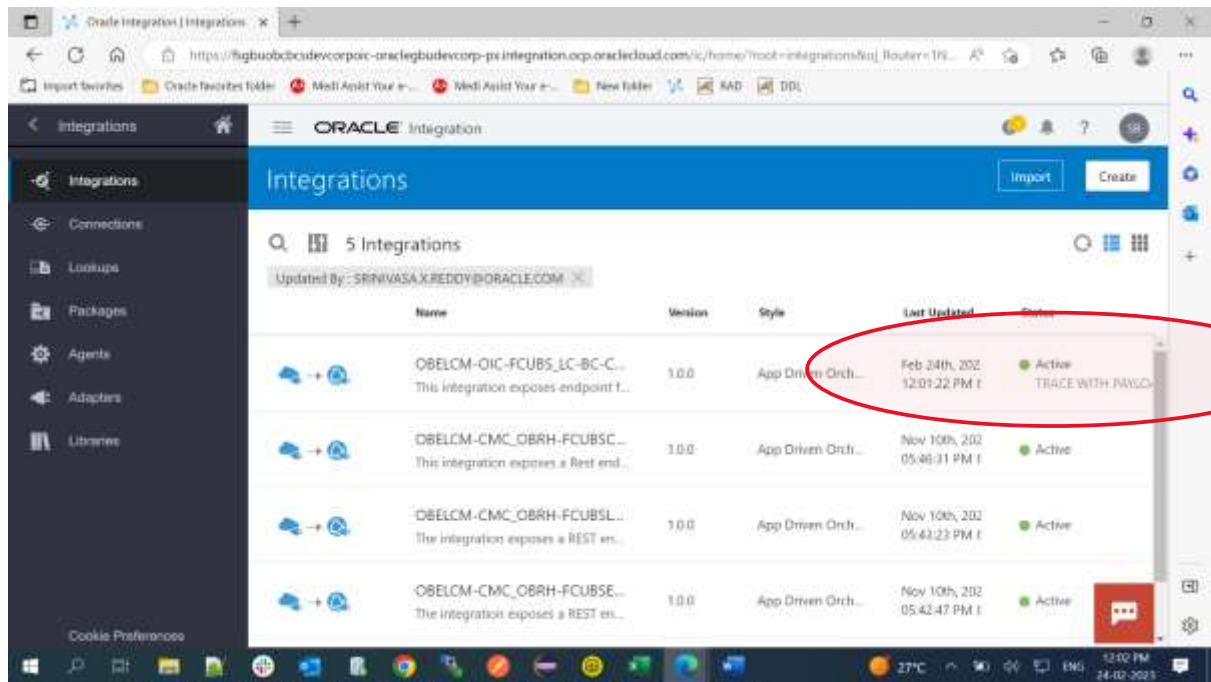
2) Enable Tracing and Include payload by click the check boxes and then click Activate



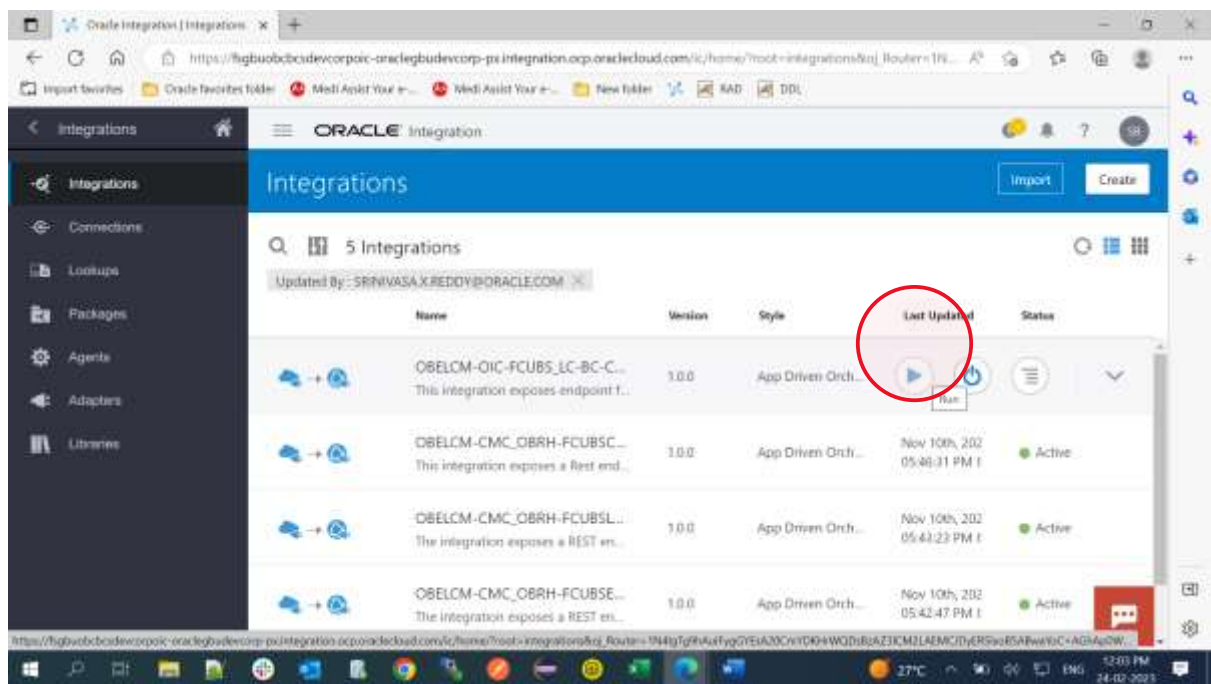
3) Activation is Successful.



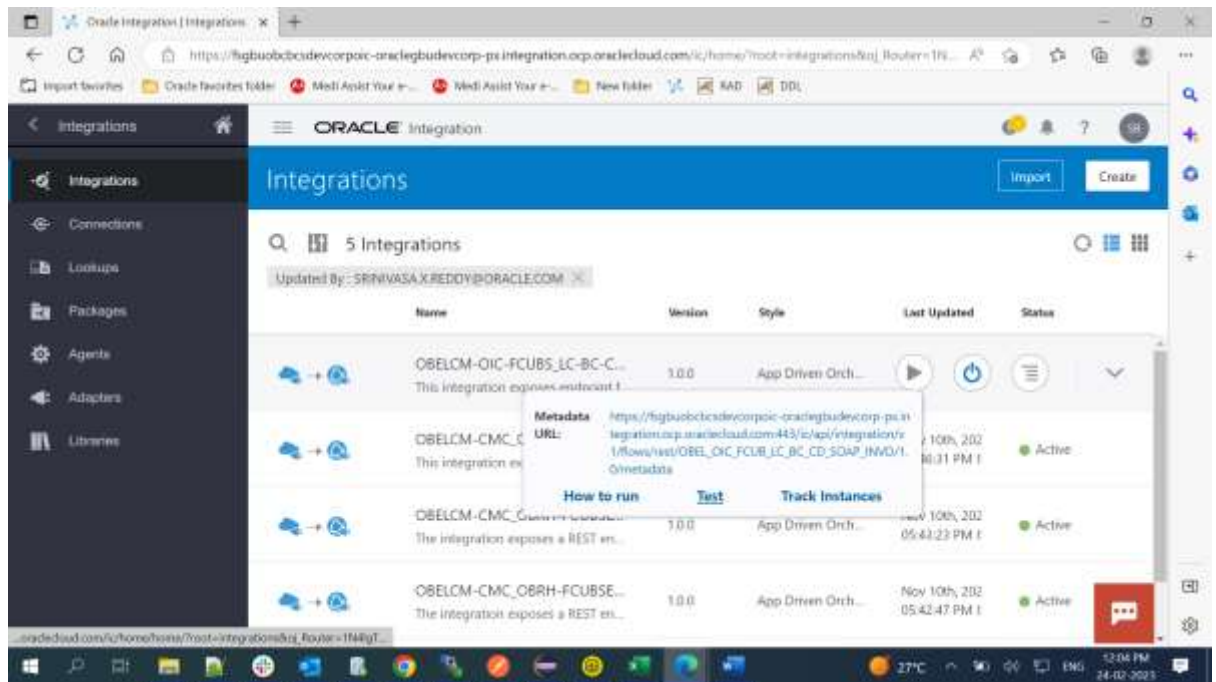
4) click on the refresh button to see the status **Active** for the integration



5) Click on Run

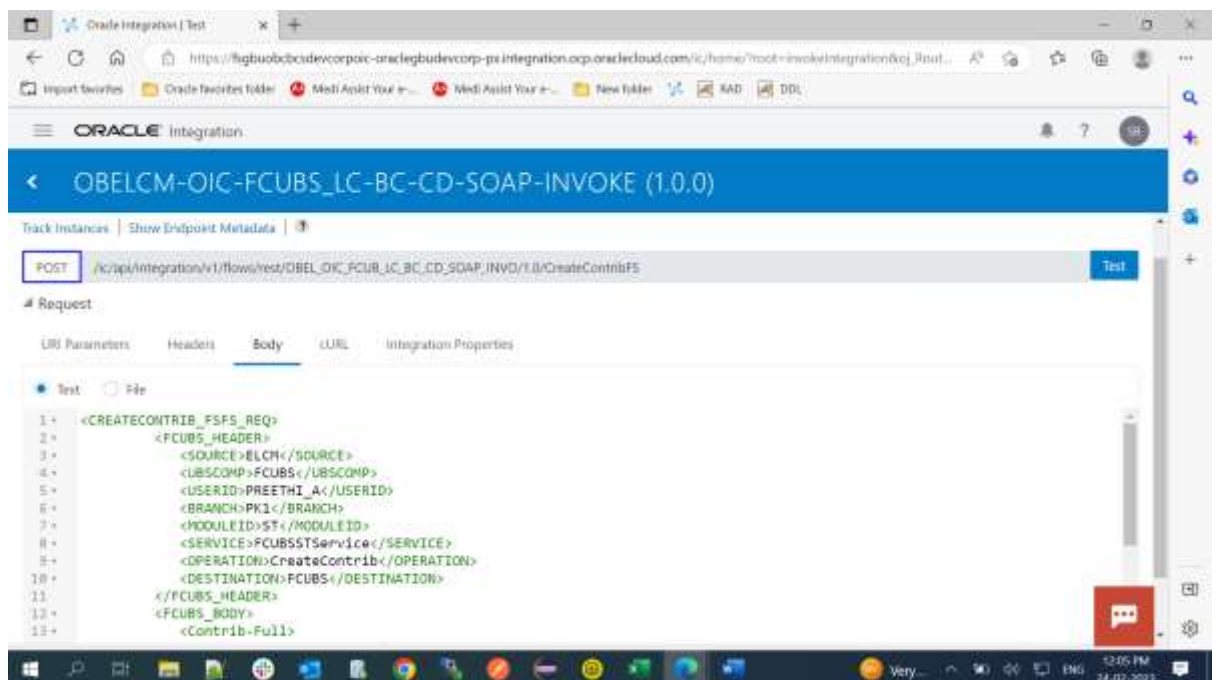


6) Click Test link.



7. Testing the Integration

7) Provide request xml in the body as shown below



8) Click Test and verify the response

The top screenshot shows the Oracle Integration Cloud (OIC) interface for the integration 'OBELCM-OIC-FCUBS_LC-BC-CD-SOAP-INVOKE (1.0.0)'. The 'Test' button is circled in red. The 'Request' section shows the following XML payload:

```

1 <CREATECONTRIB_FSFS_REQ>
2   <FCUBS_HEADER>
3     <SOURCE>ELCH</SOURCE>
4     <UBSCOMP>FCUBS</UBSCOMP>
5     <USERID>PREETHI_A</USERID>
6     <BRANCH>PK1</BRANCH>
7     <MODULEID>ST</MODULEID>
8     <SERVICE>FCUBSSTService</SERVICE>
9     <OPERATION>CreateContrib</OPERATION>
10    <DESTINATION>FCUBS</DESTINATION>
11  </FCUBS_HEADER>
12  <FCUBS_BODY>
13    <Contrib>Full

```

The bottom screenshot shows the 'Response' section with the following XML payload:

```

38 </Contrib>Full
39 </FCUBS_BODY>
40 </CREATECONTRIB_FSFS_REQ>

```

The 'Response' status is 200 OK. The 'Activity Stream' section shows the following log entries:

- 12:12:37.648 PM Message processed by trigger CreateContribFS
- 12:12:37.674 PM Message entered Scope callCreateContribFSScope
- 12:12:37.679 PM Data Mapping completed
- 12:12:37.898 PM Message received by invoke callCreateContribFS
- 12:12:38.380 PM Message entered Switch
- 12:12:39.356 PM Message routed through Switch Route with condition Otherwise
- 12:12:39.405 PM Message exited Switch
- 12:12:39.408 PM Data Mapping completed
- 12:12:39.418 PM Message exited Scope callCreateContribFSScope
- 12:12:39.485 PM Reply to CreateContribFS completed

You can track the use case payload by navigating Home→ Monitoring → Integrations → Tracking

