IBM Integration Bus

Distributing IIB Workload using IBM Integration Bus on Cloud and Callable Flows

Featuring:

- Running a pre-built REST API on IIB on Cloud
- Callable Flow running locally “on prem”
- Configuration of an IIB Switch server and Connectivity agent to enable a REST API running on IIB on Cloud to access Callable message flows running in a Windows environment

September 2016

Hands-on lab built at product
Version 10.0.0.6 and IIB on Cloud

1. INTRODUCTION 3

1.1 SCENARIO OVERVIEW 3
1.2 OUTLINE OF TASKS 4
1.3 OPEN THE WINDOWS LOG MONITOR FOR IIB 4
1.4 IMPORTANT NOTE 4
1.5 CONFIGURE INTEGRATION BUS NODE TO WORK WITH DB2 5

2. DEPLOY BAR FILE WITH CALLABLE APPLICATION (ON PREMISE) 6

2.1 OPEN THE INTEGRATION NODE'S WEB USER INTERFACE 6
2.2 DEPLOY BAR FILE FOR CALLABLE APPLICATION 7

3. IIB ON CLOUD INTEGRATION 9

3.1 DEPLOY APPLICATION TO IIB ON CLOUD 9
3.2 CONNECT IIB ON CLOUD TO IIB ON PREMISE 14
3.2.1 Set up an agent 14
3.2.2 View connected flows on IIB on Cloud 17

4. TEST YOUR CLOUD INTEGRATION 19

4.1 START YOUR IIB ON CLOUD INTEGRATION 19
4.3 TEST THE REST API IN THE BROWSER 22

END OF LAB GUIDE 23

1. Introduction

IIB V10.0.0.4 introduces Callable Flows. This enables a message flow (or integration service, or REST API) to invoke a separate message flow in a call/return (blocked wait) programming model.

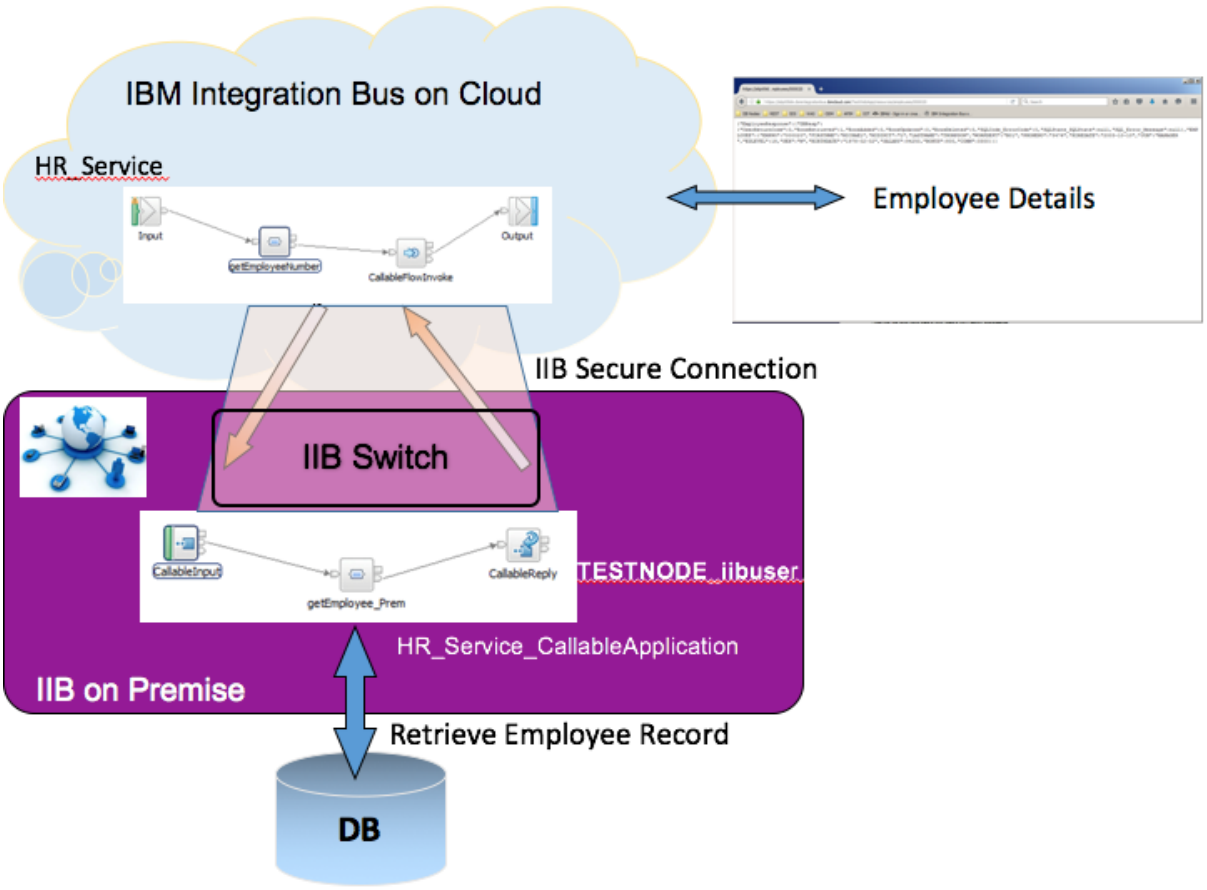
The calling and called message flows could operate on remote Integration Nodes, including one on IIB on Cloud (IIB on Cloud) and one On-Premise.

1.1 Scenario Overview

Lab 16L10 - Callable Flows Development from this series, added an operation to an existing REST API and developed a Callable Application. This is one of three Callable Flows labs in this series, where the REST API runs on the IIB on Cloud runtime and the callable application is “On-premises”.

In this lab, you will run the REST API that you modified in 16L10 (HR_Service) on IIB on Cloud invoking a flow in an application, which is running on an Integration Node “On-premises”. The callable application will access a database, located on the same host as its Integration Node, and retrieve employee details.

The REST API and the Callable application are provided for you and the lab design and logic is contained in the Callable Flows Development lab 16L10.



Provided by IBM BetaWorks

1.2 Outline of tasks

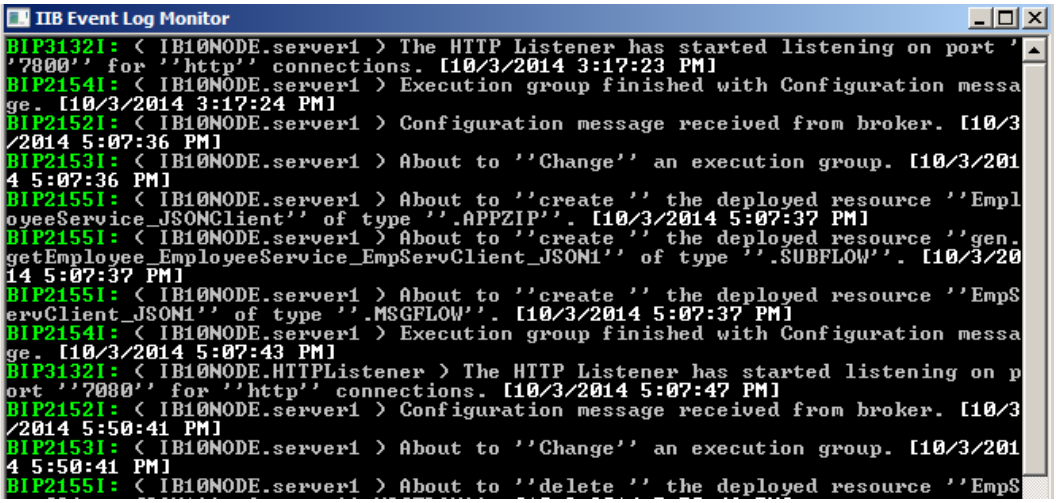
The tasks to complete in this lab are the following:

1. Import and deploy the Callable Application “HR_Service_CallableApplication” to your local Integration Node.
2. Access IIB on Cloud and upload the REST API HR_Service bar file.
3. Configure an IIB Switch server and Connectivity agent to enable IIB on Cloud to connect to your local IIB environment
4. Use the HR_Service REST API to obtain EMPLOYEE data (on-premises) by calling the HR_Service_CallableApplication.

1.3 Open the Windows Log Monitor for IIB

A useful tool for IIB development on Windows is the IIB Log Viewer. This tool continuously monitors the Windows Event Log, and all messages from the log are displayed immediately.

From the Start menu, click IIB Event Log Monitor. The Monitor will open; it is useful to have this always open in the background.



This tool is not shipped as part of the IIB product; please contact us directly if you would like a copy.

1.4 Important note

This lab will require an ‘IIB on Cloud’ account.

In other scenarios, you can use the Free Trial version, which you can request at *http://www-03.ibm.com/software/products/en/ibm-integration-bus-on-cloud*.

If you are using the provided VMWare image, you should use the Windows user "iibuser". This user is a member of mqbrkrs and mqm, but is not a member of Administrators. The user "iibuser" can create new IIB nodes and do all required IIB development work. However, installation of the IIB product requires Administrator privileges (not required in this lab).

1.5 Configure Integration Bus node to work with DB2

Note 1

Login to Windows VM Image as the user "iibuser", password = "passw0rd".

Start the IIB Toolkit from the Start menu.

Note 2

If you have already done Lab 1 in this series (create a REST API using a Swagger document), you can skip straight to Import Applications on the next page.

The HRDB database, and the EMPLOYEE and DEPARTMENT tables have already been created on the supplied VMWare image. If you wish to recreate your own instance of this database, the commands

1_Create_HRDB_database.cmd and
2_Create_HRDB_Tables.cmd

are provided for this. If used in conjunction with the VM image, these commands must be run under the user "iibadmin". Appropriate database permissions are included in the scripts to GRANT access to the user iibuser.

To run this lab, the Integration Bus node must be enabled to allow a JDBC connection to the HRDB database.

1. Open an IIB Command Console (from the Start menu), and navigate to

   c:\student10\Create_HR_database

2. Run the command

   3_Create_JDBC_for_HRDB

   Accept the defaults presented in the script. This will create the required JDBC configurable service for the HRDB database.

3. Run the command

   4_Create_HRDB_SecurityID

4. Stop and restart the node to enable the above definitions to be activated

   mqsistop TESTNODE_iibuser

   mqsistart TESTNODE_iibuser

This will create the necessary security credentials enabling TESTNODE_iibuser to connect to the database.

2. Deploy BAR file with Callable Application (on premise)

If you have just done the Callable Flows in Docker lab 16L11, you can go directly to step 3.

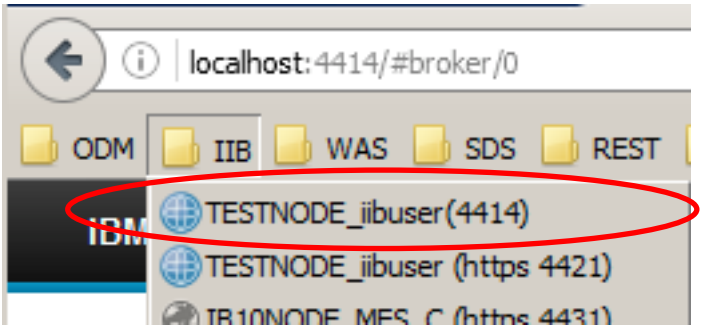
In this section you will deploy the Callable Application to the Integration Node On premise.

Please note that a pre-built barfile is provided for this purpose. The barfile containing the REST API will be deployed directly to IIB on Cloud from the file system.

2.1 Open the Integration Node's Web User Interface

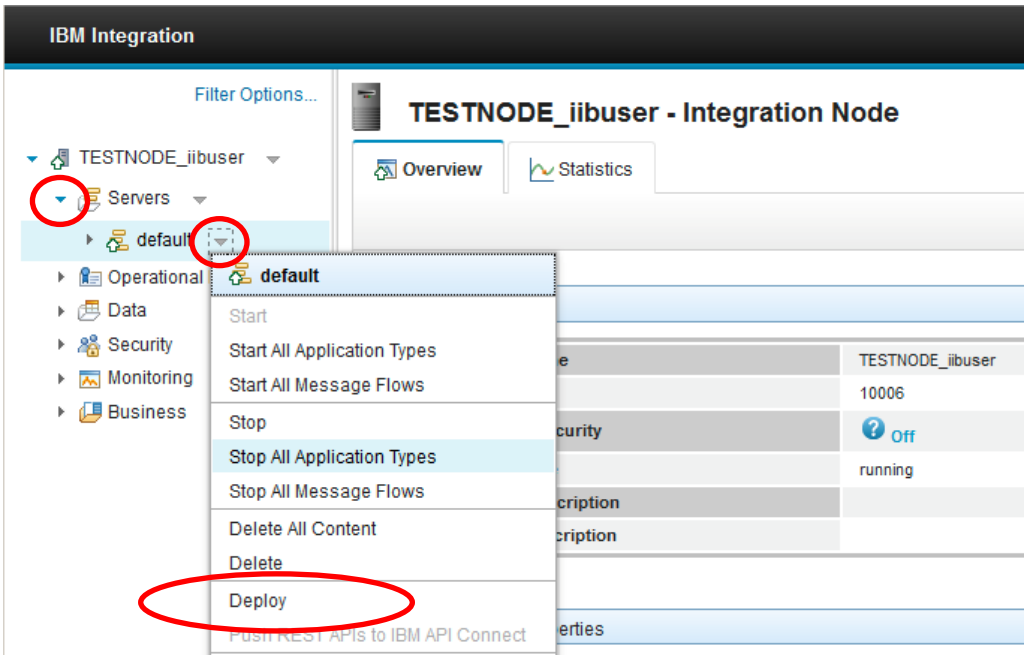
1. Open a browser window and click IIB tab in the Toolbar, then the TESTNODE_iibuser bookmark (http://localhost:4414/).

   If you are using your own IIB installation, you may need to specify different hostname and port.



2. In the Web UI for the Integration Node expand Servers and then open the context menu of the 'default' integration server.

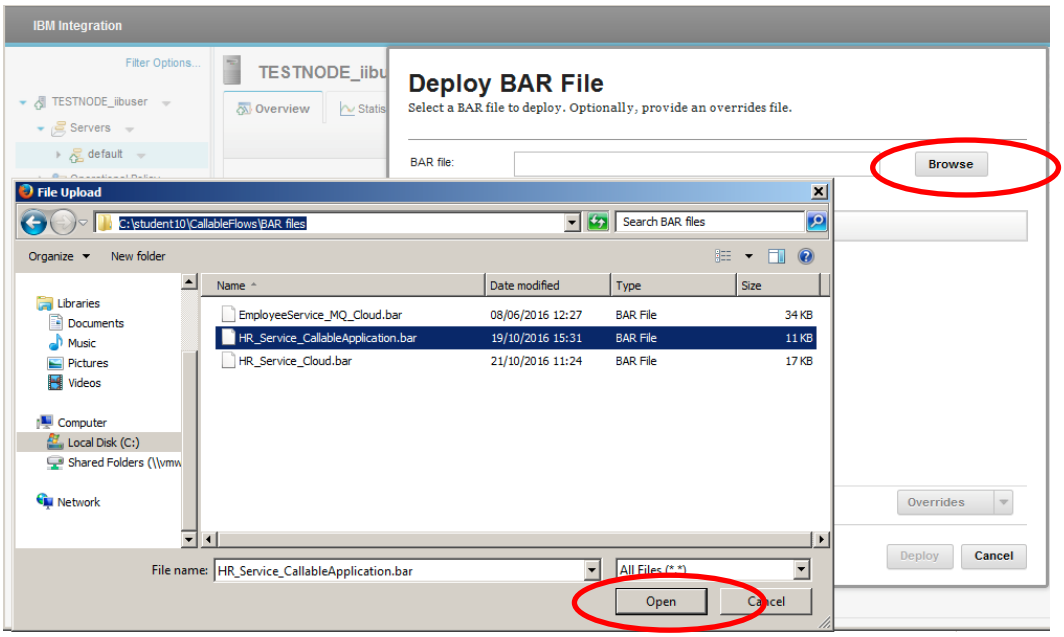
   Click on 'Deploy'.



2.2 Deploy BAR file for Callable Application

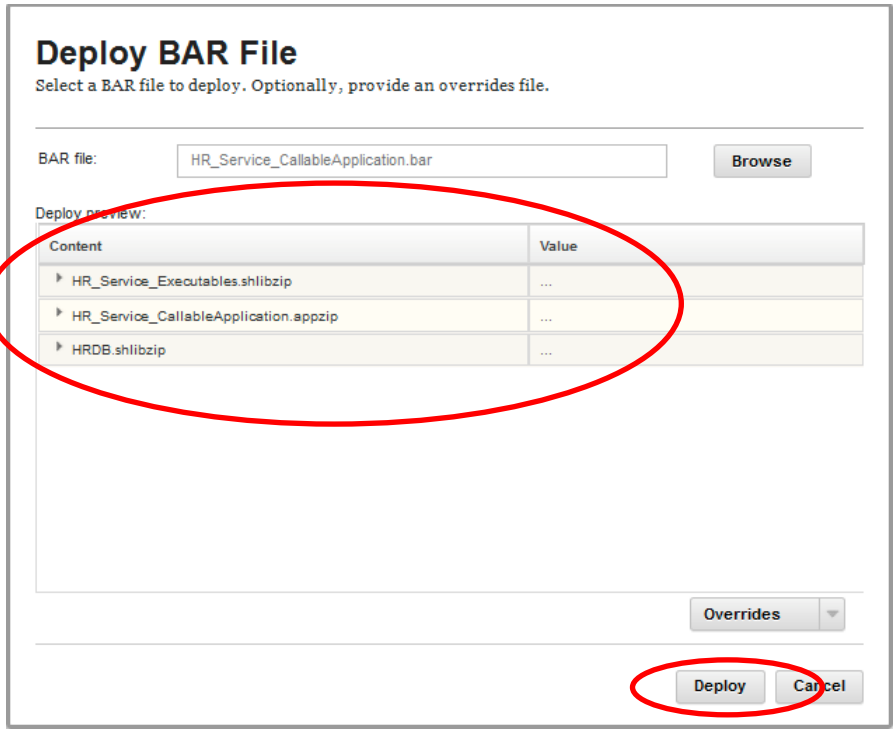
1. In the open dialog, click 'Browse' and navigate to **C:\student10\CallableFlows\BAR files** directory.

   Select **HR_Service_CallableApplication.bar** and click 'Open'.

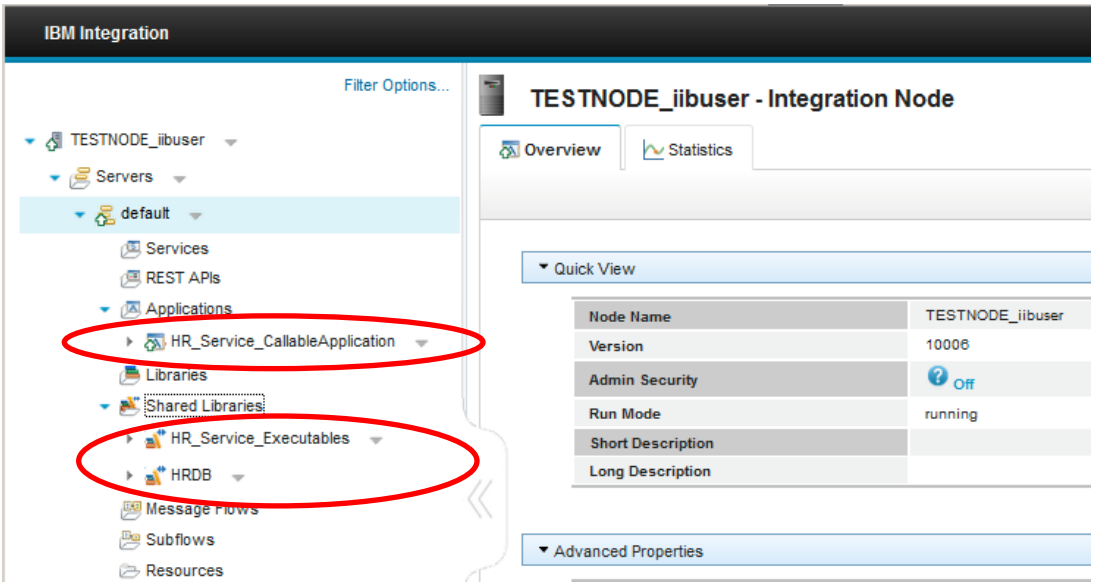


2. In the 'Deploy preview' you will see the callable application and the shared libraries which will be deployed to the integration server.

   Click 'Deploy'.



3. Confirm the successful deployment of the application and libraries (they should appear in the left navigation window in the browser):



3. IIB on Cloud Integration

In this part of the lab you will connect to the IIB on Cloud user interface. You will then deploy your integration (REST API) to IIB on Cloud and connect it to the IIB Integration Node on Premise.

If you have an active account for IIB on Cloud, you can use it for the rest of the lab.

For reference, if you do not have an account visit the IIB on Cloud web page at:

http://www-03.ibm.com/software/products/en/ibm-integration-bus-on-cloud

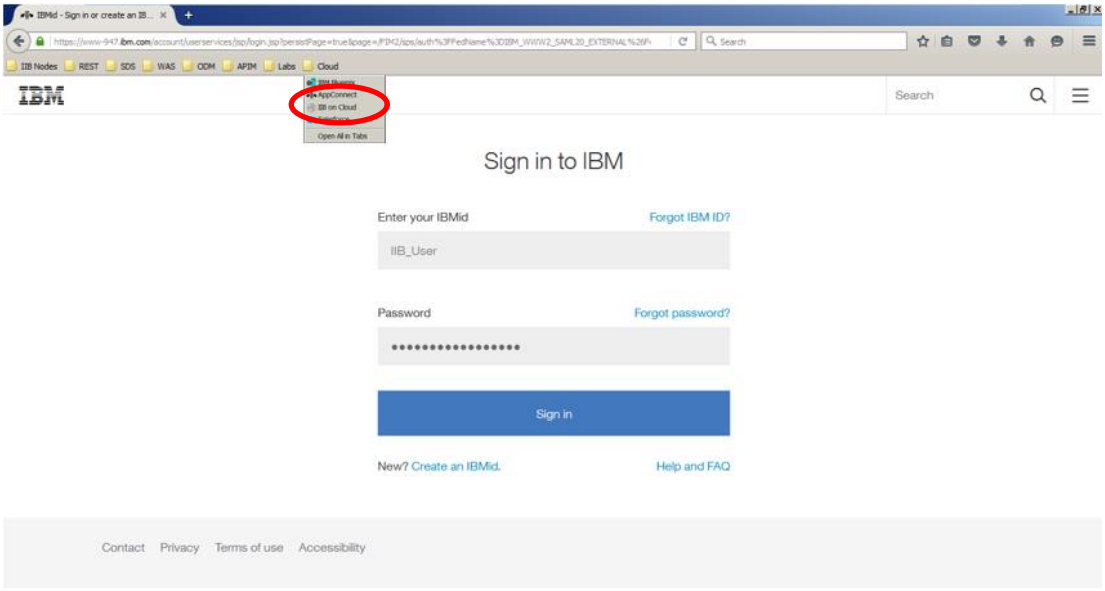
3.1 Deploy application to IIB on Cloud

1. Open a browser window and in the Bookmarks Toolbar click Cloud and then the "IIB on Cloud" bookmark.

   If you are not using the workshop VMWare Windows system, use the following URL:

   https://ibm-cloud-ui.ibmintegrationbus.ibmcloud.com/

   Enter your 'IBM Id' and password and click '**Sign In**'.



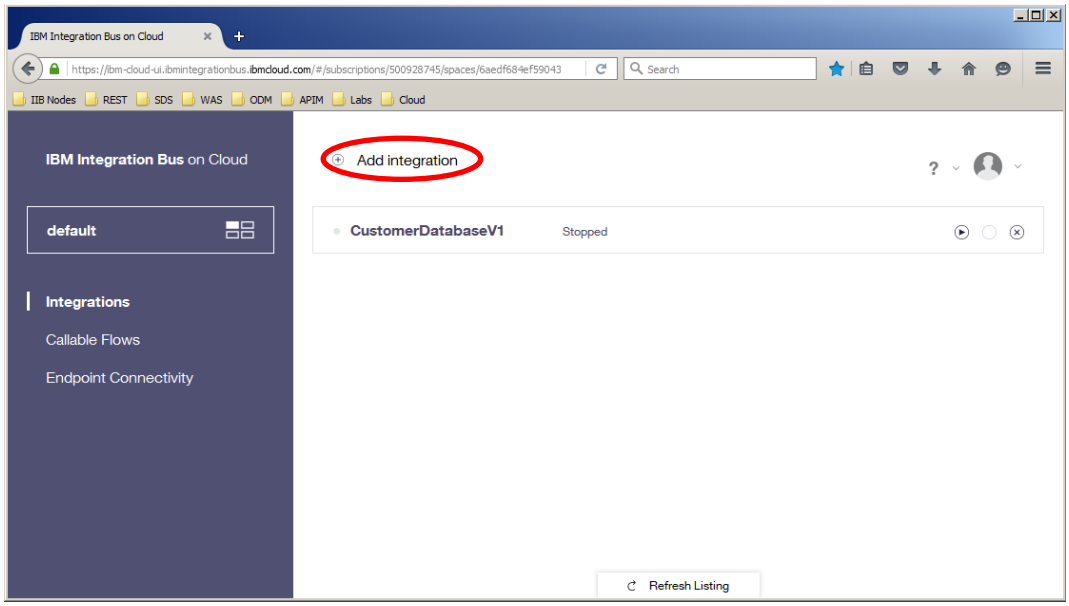
2. You now have your IIB on Cloud default integration space opened.

   If you have done the introduction lab for IIB on Cloud from this series you will have the sample integration CustomerDatabaseV1 shown in this view.

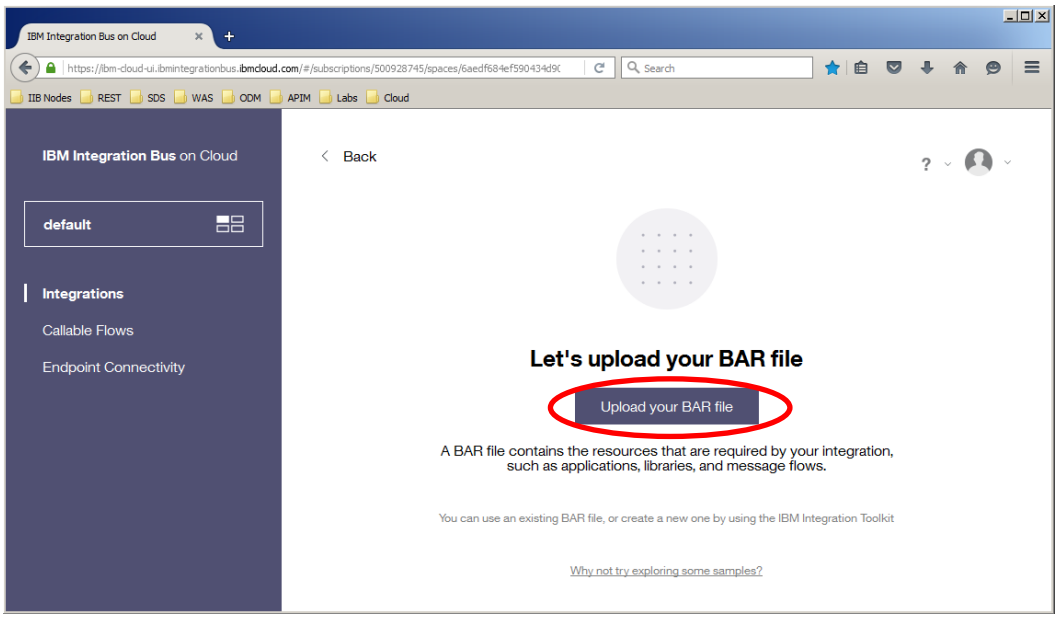
   The trial version for IIB on Cloud allows only one integration to run at one time. Therefore, if you have previous applications running, please make sure that they are all stopped.

   You will now upload a BAR file with the REST API integration that you created in earlier in earlier lab in this series. The BAR file is provided for you.

   Click '**Add Integration**'.



3. You are presented with the Web UI for uploading a BAR file.
   Click '**Upload your BAR file**'.

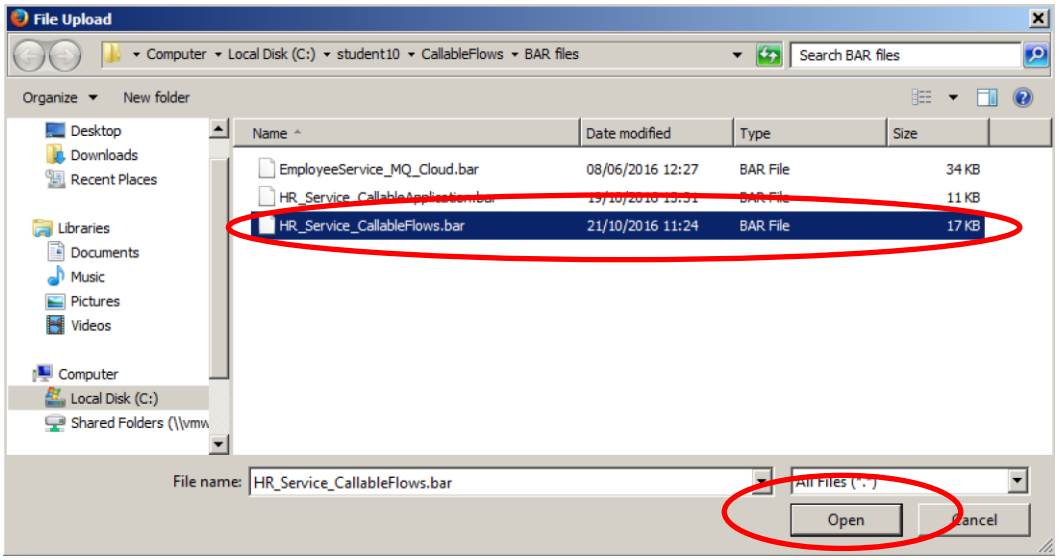


4. A File Upload window is opened.

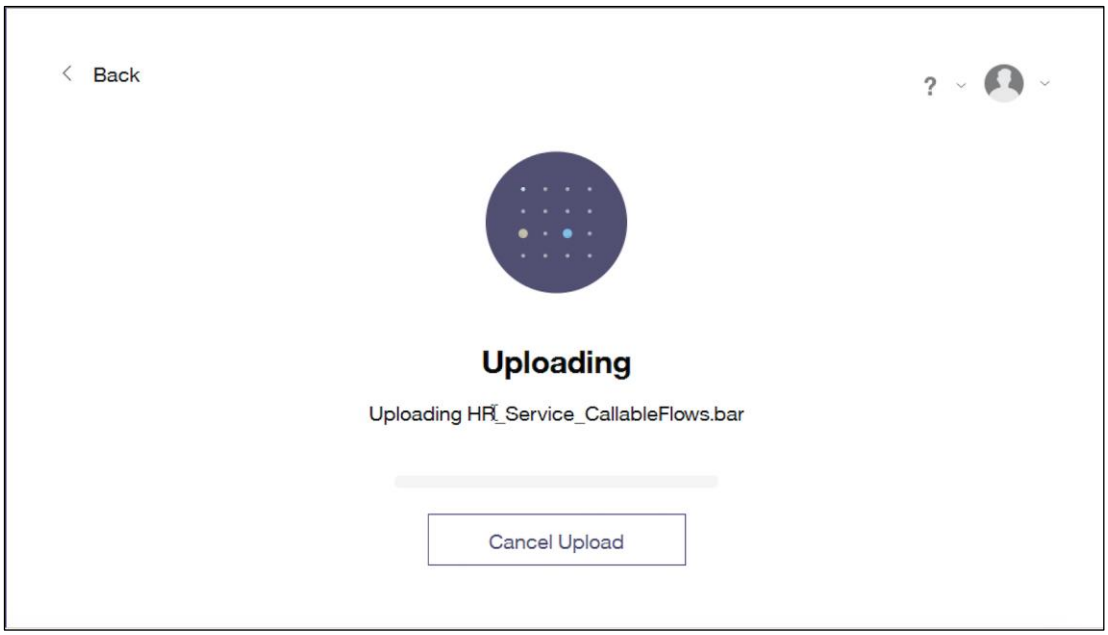
   In case you did not complete the Callable Flows Development Lab) we have provided a pre-built solution of the BAR file in the **Student10\CallableFlows\BAR Files** directory.

   Navigate to C:\student10\CallableFlows\BAR files and select **HR_Service_CallableFlows.bar**.

   Click **Open:**



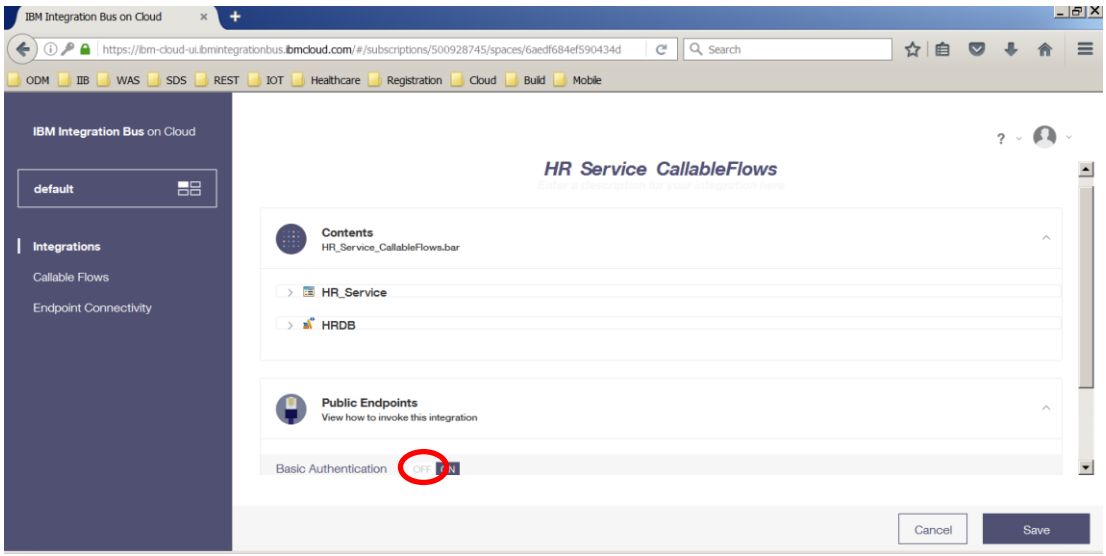
5. The BAR File will be uploaded to IIB on Cloud and then validated:



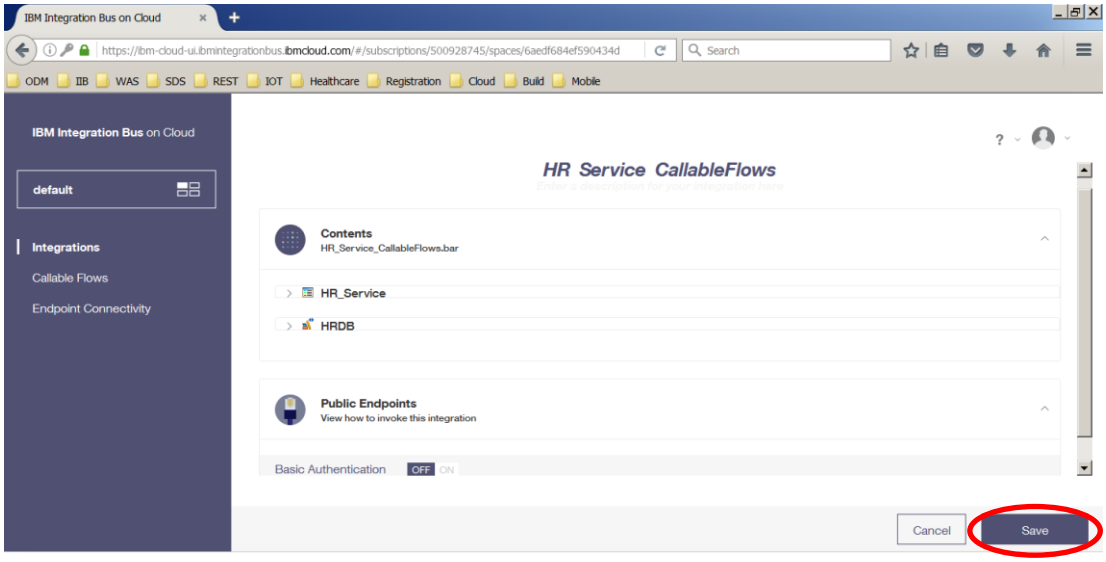
6. When the upload and validation is complete, you are presented with the integration content that will be deployed to IIB on Cloud.

   You can view the artefacts that will be deployed by clicking **'HR_Service'.**

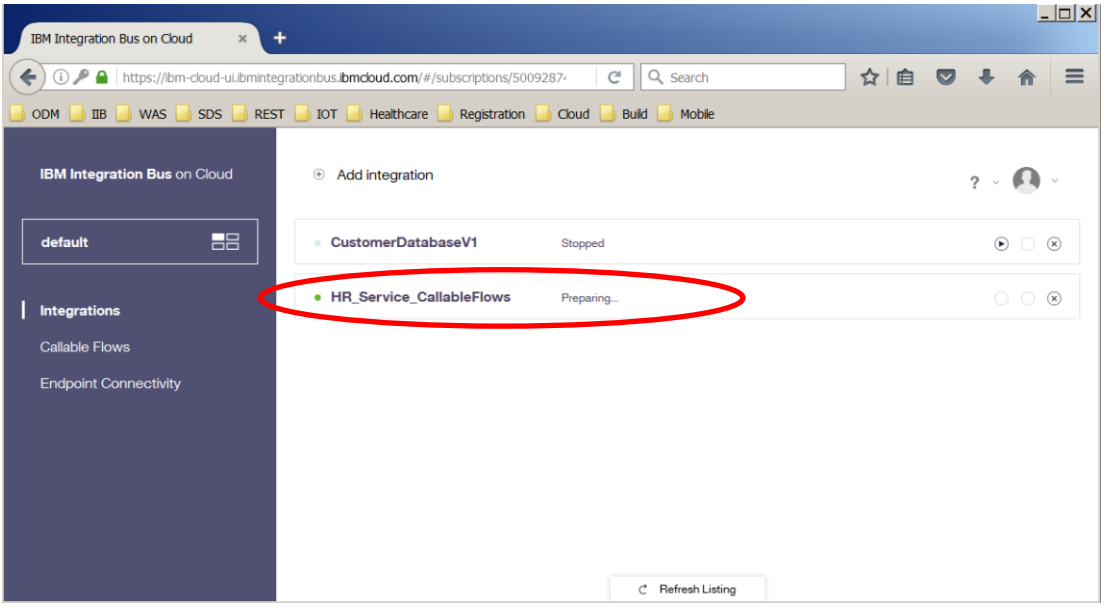
   Turn off the Basic Authentication for this integration by clicking the '**Off**' button as shown below (It is '**On**' by default):



7. When ready, **Save** the integration.



8. The Web user interface will show that the integration is '**Preparing**'. This may take a minute or two to complete.



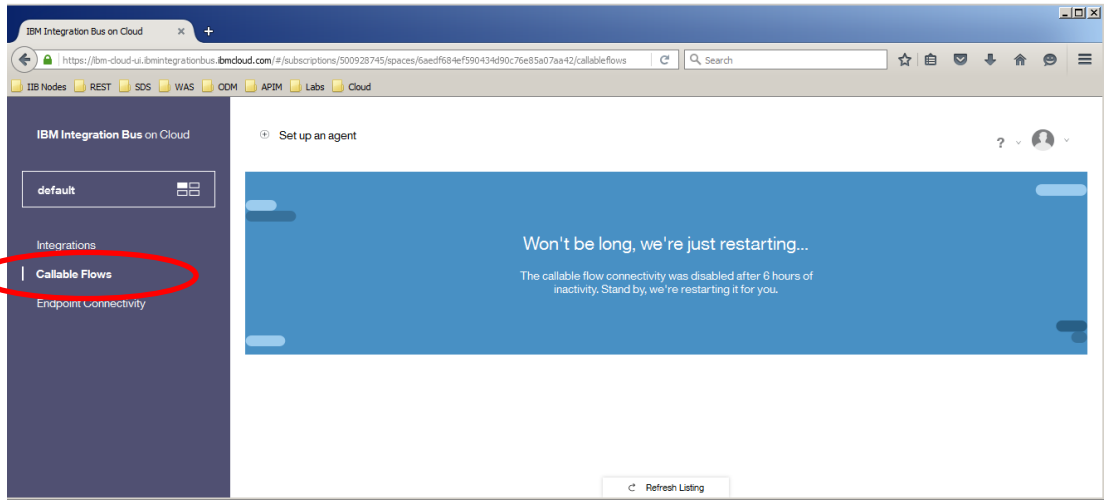
Whilst the integration is being prepared, proceed with the rest of the lab.

3.2 Connect IIB on Cloud to IIB on Premise

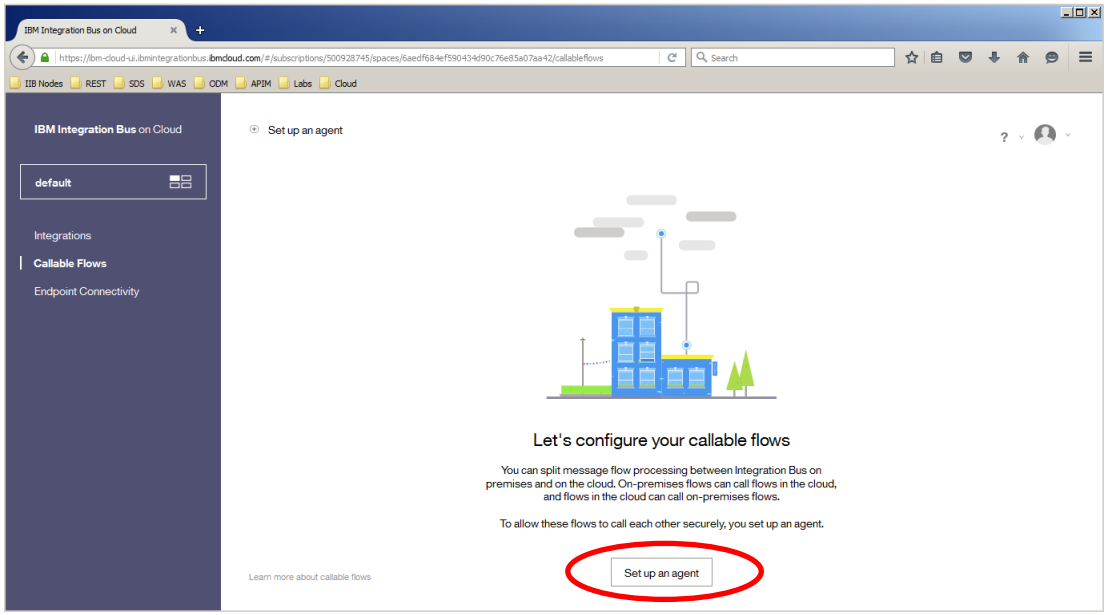
In this part of the lab, you will create a 'switch' which will connect your IIB on Cloud system to your local IIB environment. This will allow the integration running on IIB on Cloud to connect to a flow running locally using a secure connection.

3.2.1 Set up an agent

1. Click on the **Callable Flows** tab. As this may be your first time setting up an agent it may take a minute or two for the utilities to restart.

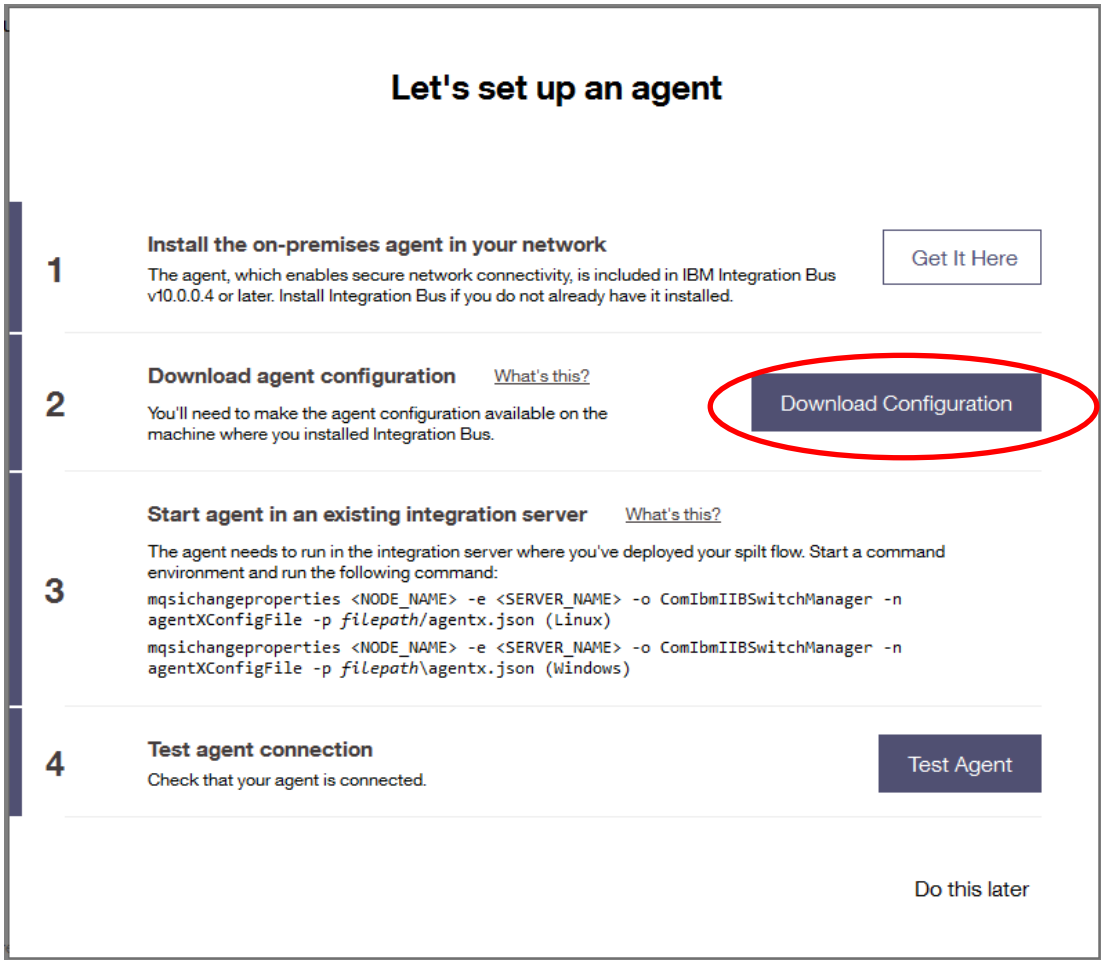


2. Click '**Set up an agent**'.

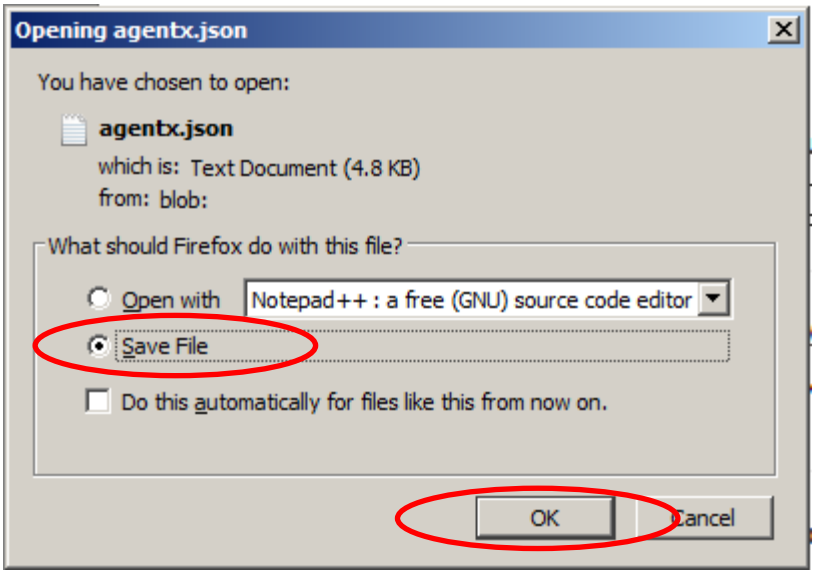


3. The pop-up shows the steps required for setting up the connection agent.
We already have a local installation of IIB, so the first step is not required.

   Click '**Download Configuration**'.

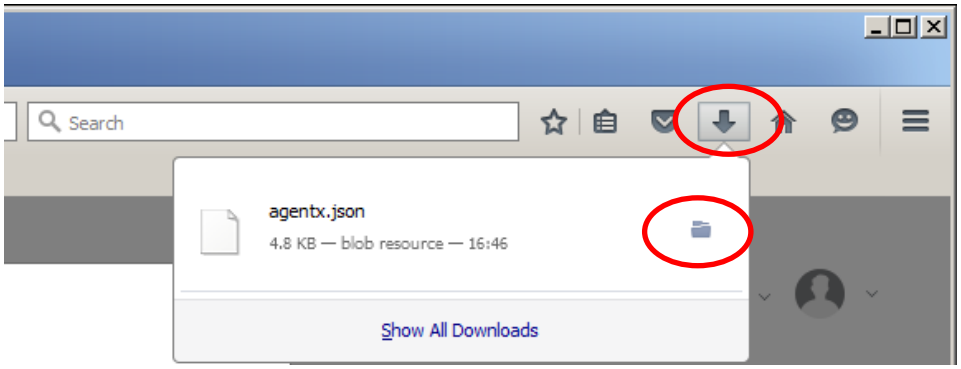


4. Select to **Save** the file and then click '**OK**'.

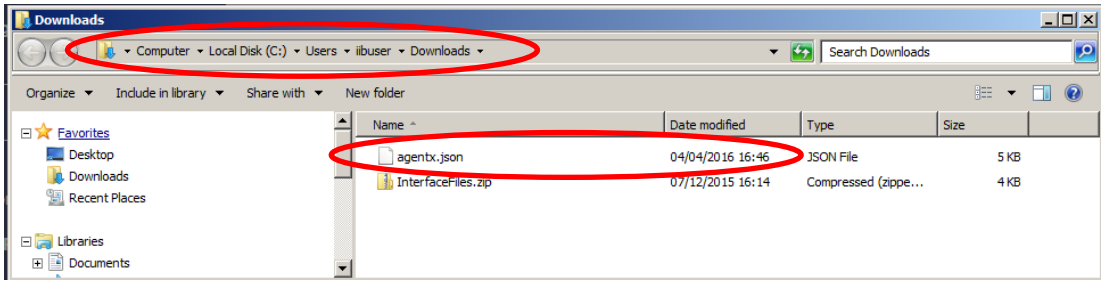


5. When the download has finished, view the folder in which the **agentx.json** file has been saved.

   In a Firefox browser, click the '**Download**' icon and then the small '**Folder**' icon.



6. If you are using the IIB workshop VM Image, the download folder is **C:\Users\iibuser\Downloads.**



7. In the next step, you will start an agent in the default integration server in the **TESTNODE_iibuser** Integration Node. You will then deploy the callable application '**HR_Service_CallableApplication**' (step 2.2.1) to this integration server.

   In an Integration Console, type the following command (***all on one line***):

```
mqsichangeproperties TESTNODE_iibuser
                     -e default
                     -o ComIbmIIBSwitchManager
                     -n agentXConfigFile
                     -p C:\Users\iibuser\Downloads\agentx.json
```

   Make sure that you see: `BIP8071I: Successful command completion`

8. Now that you have set up the agent, make sure that IIB on Cloud can connect to your local Integration Server using the agentx configuration.

   In your browser window click '**`Test Agent`**' (step 4)

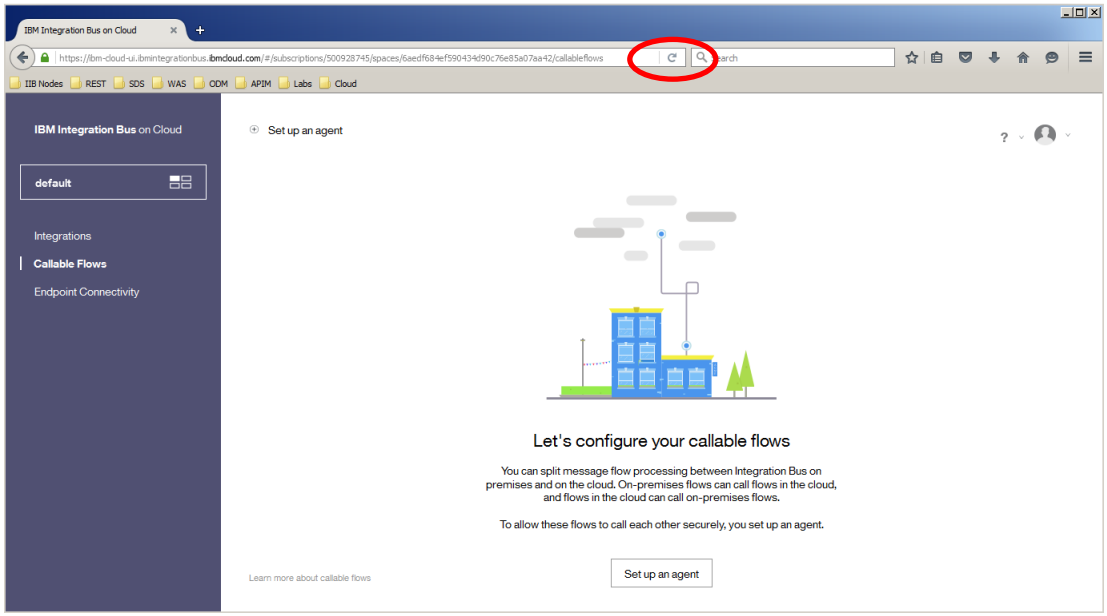
9. If successful, you will see the message '**1 agent connected**' in a green box:

Click **Finish**.

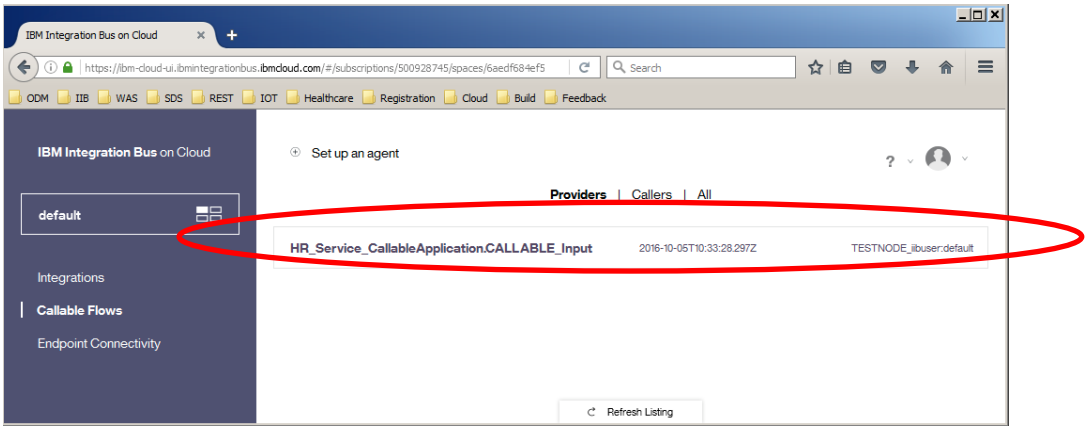
3.2.2 View connected flows on IIB on Cloud

Now that IIB on Cloud is able to connect to your local IIB environment, any callable flows that are running locally can be seen by IIB on Cloud. Note: *depending on timing and network performance, you may need to stop and restart your local IIB node to enable IIB on Cloud to see the Callable Flows on TESTNODE_iibuser.*

1. In your browser, you should still have the '**Callable Flows**' view. **Refresh** your browser window.



2. You should now see a list of callable flows (in this case with just one flow) that are running On Premise, which are connected to your IIB on Cloud.

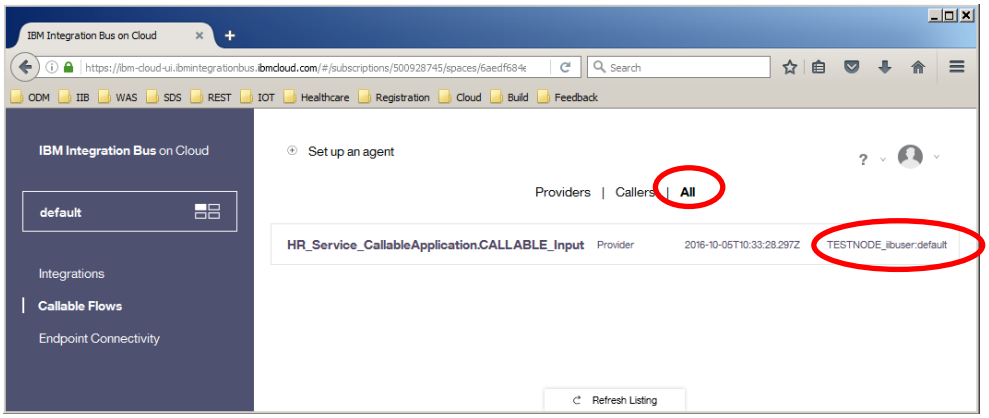


Note: If you do not see your callable flows registered in IIB on Cloud, stop and start your local default integration server. This will ensure that you are able to see your Callable Flows in the Web UI of the IIB on Cloud system.

3. In IIB on Cloud you will see that there are three tabs, **Providers**, **Callers** and **All**, which list the connected flows based on the type of integration they provide. In this lab the message flow running locally is being called and is '*providing*' a response to the integration deployed on IIB on Cloud.

   Click 'All'.

   This will show a combined view of all the flows and indicating if they are Providers or Callers.



   The Web UI also shows to which on-premise Integration Node and server the flow is deployed.

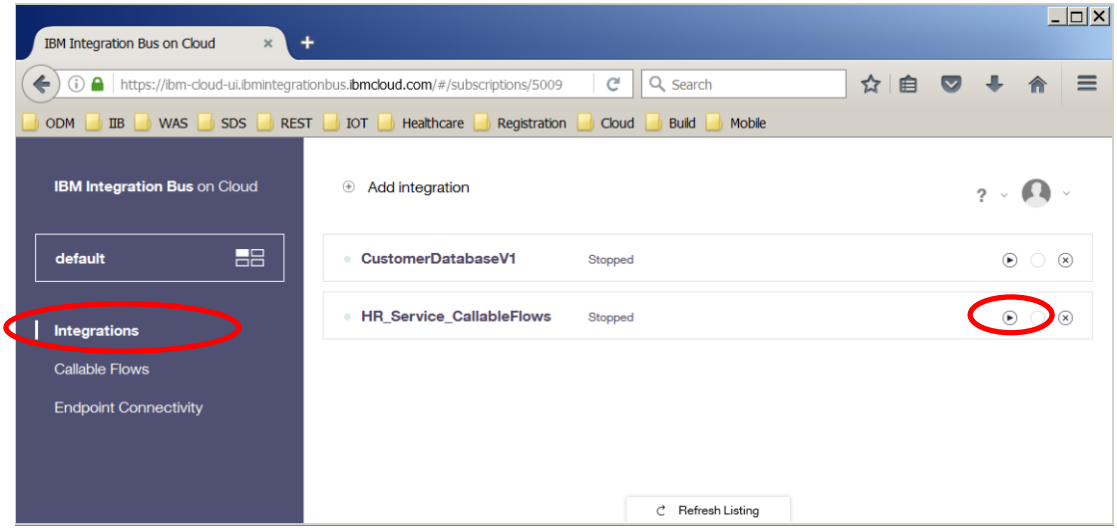
4. Test your Cloud Integration

In this part of the lab you will test your IIB on Cloud integration. The REST API HR_Service will call HR_Service_CallableApplication through the IIB Secure connection.

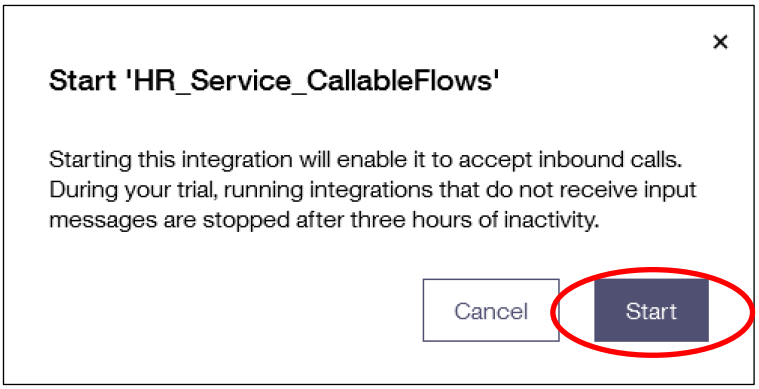
HR_Service_CallableApplication will retrieve Employee data from the EMPLOYEE table and return the response to IIB on Cloud.

4.1 Start your IIB on Cloud integration

1. In the IIB on Cloud Web UI, click **Integrations**, then start **HR_Service_CallableFlows** integration:

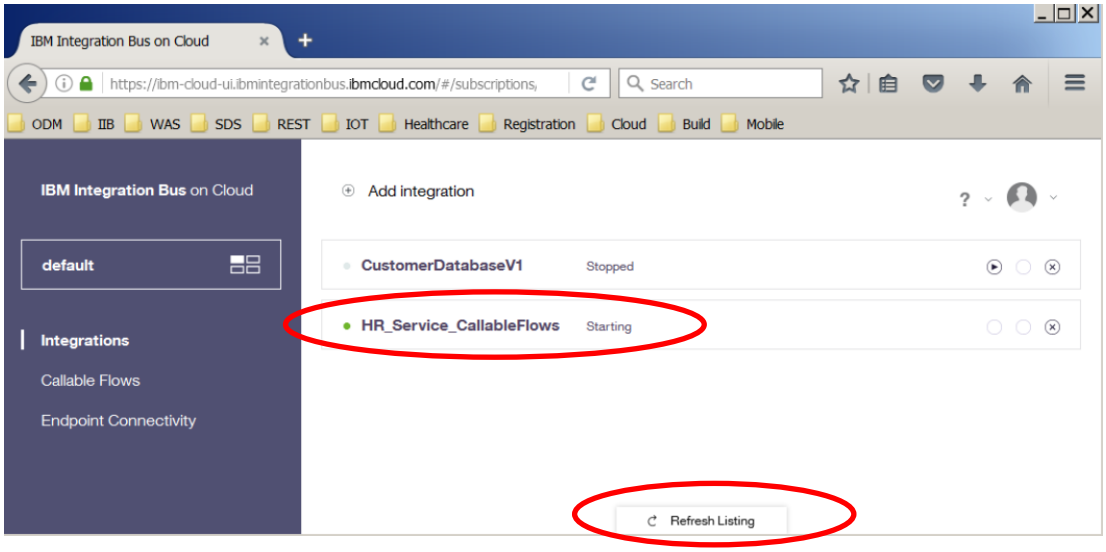


2. Confirm that the application will be started:

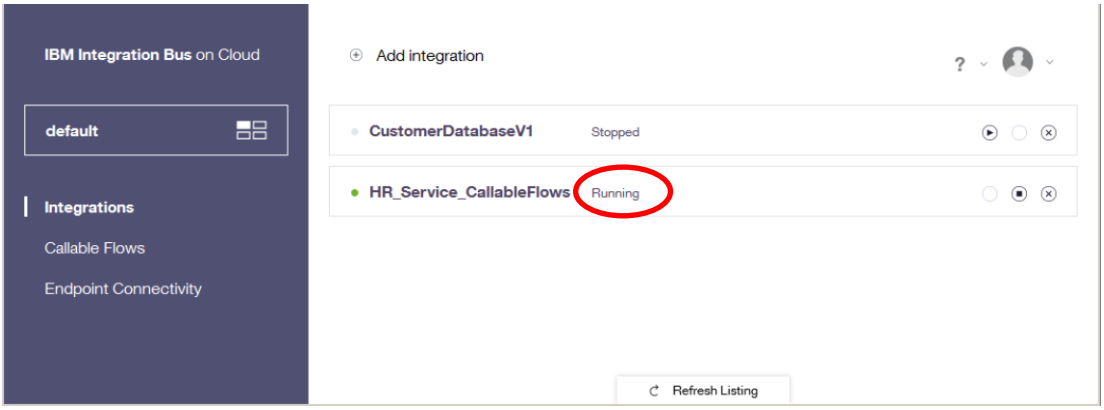


3. The IIB on Cloud browser will show that the application is starting.

   It may take a minute or two for the integration to start completely and you can check its status by click on '**Refresh Listing**':



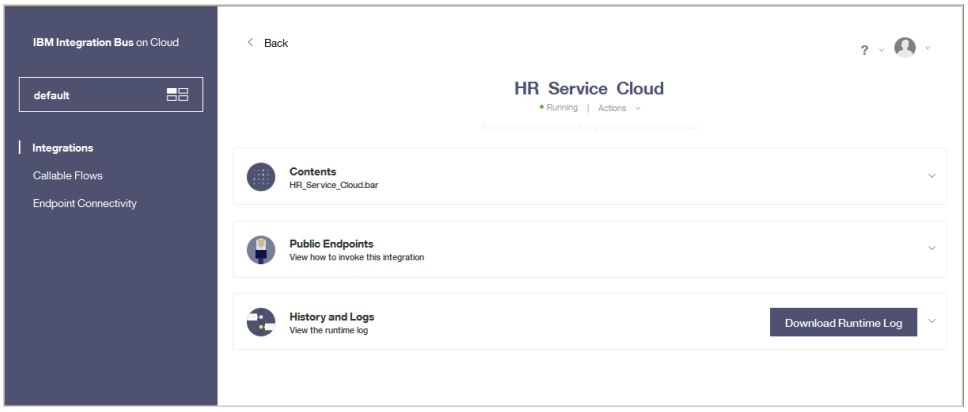
4. When it has started, the browser will be updated:



5. Click on **HR_Service_CallableFlows** integration.

   Here, you can see further details of the integration: Contents; Public Endpoints; History and Logs.

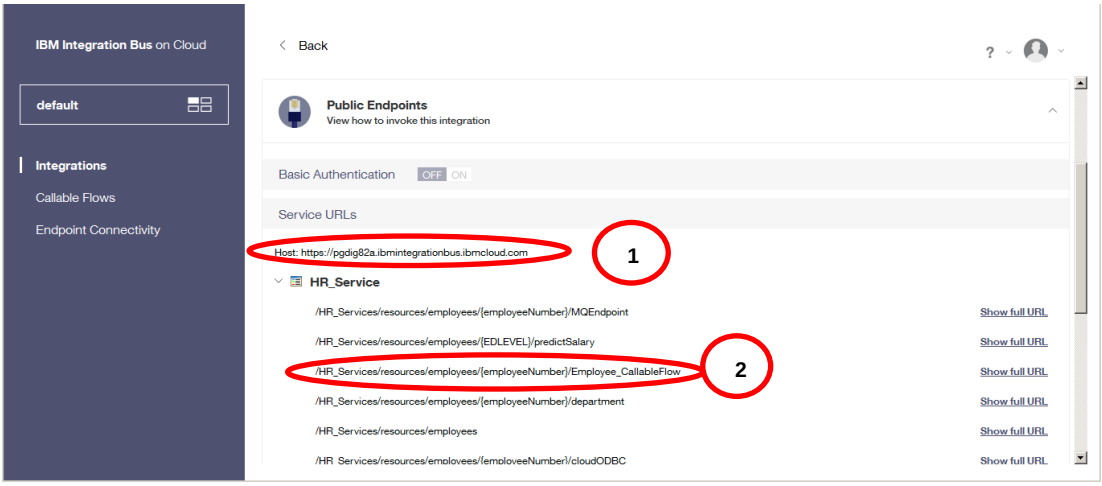
   Click '**Public Endpoints**':



6. Expand **HR_Service** endpoints under **Service URLs**.

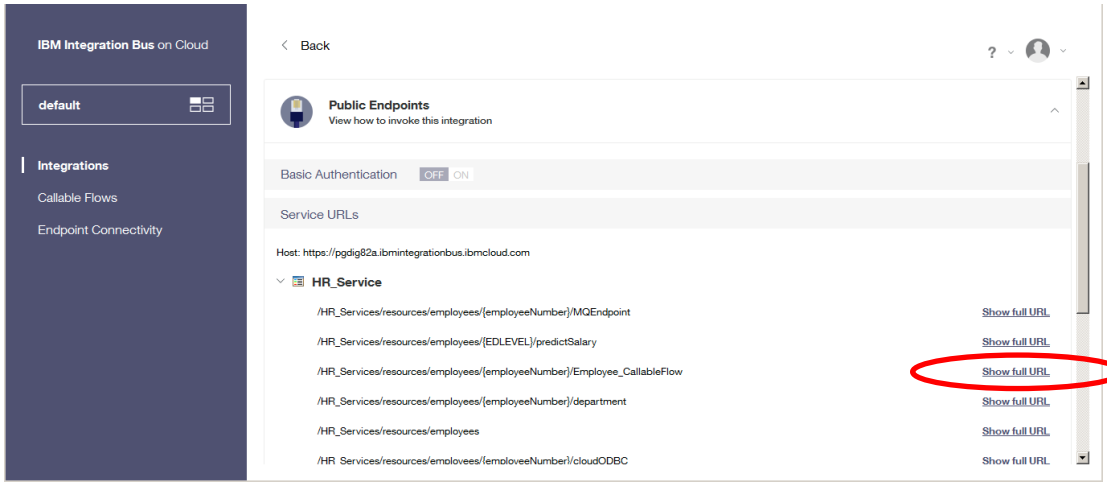
   Here you can see the main service URL for the REST API (1) and the specific endpoints for the operation available in the deployed Integration.

   However, in the REST API, the operation that we need for this lab is (Employee_CallableFlow) – (2).



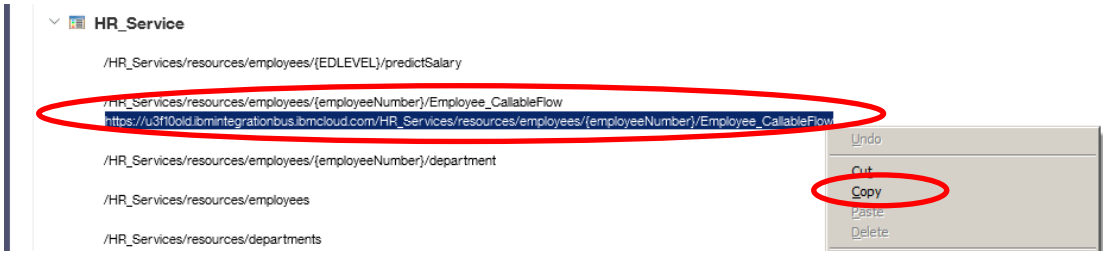
4.3 Test the REST API in the Browser

1. Click '**Show full URL**' to the right of the implemented operation – **/HR_Services/resources/employees/{employeeNumber}/Employee_CallableFlow**

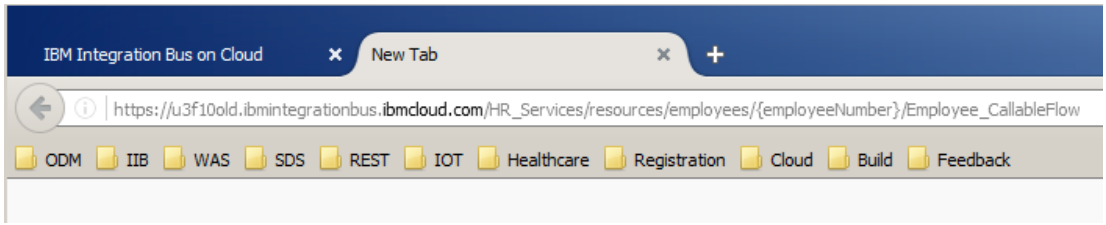


2. This will show and highlight the unique URL for this operation.

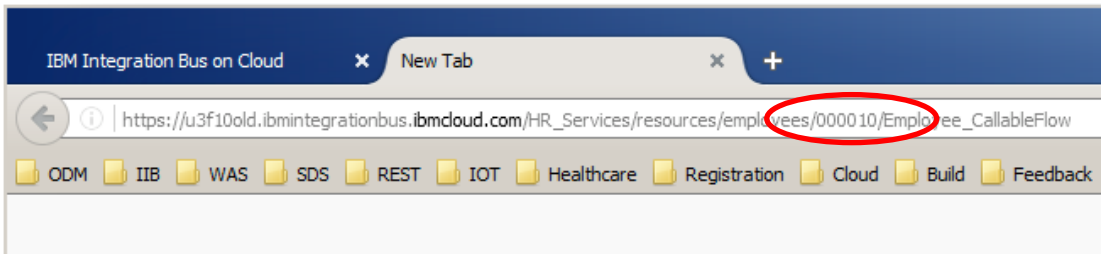
   Right-click the selected line and then click **Copy.**



3. Open a new tab in your browser. Paste the link you have just copied into the browser URL field.

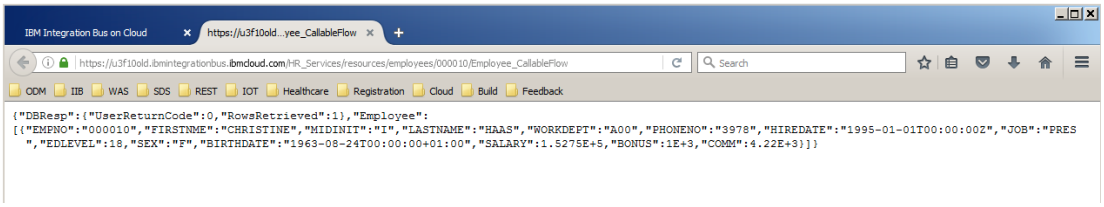


4. Replace **`{employeeNumber}`** with `000010`.



Hit **Enter**.

5. You should see the retrieved data…



END OF LAB GUIDE