An Integration Server is used to provide an isolated runtime environment for a set of deployed message flows and resources. Each integration server runs as a unique operating system process in a separate address space.

In other words, an integration server is like a super-smart middleman for computer programs and applications. Imagine it as a helpful translator that makes different apps understand each other and work together smoothly.
Each server is like a building with many private rooms, and each room is for a different program to work without causing trouble for the others. It keeps everything organized and running smoothly.

Learning objectives

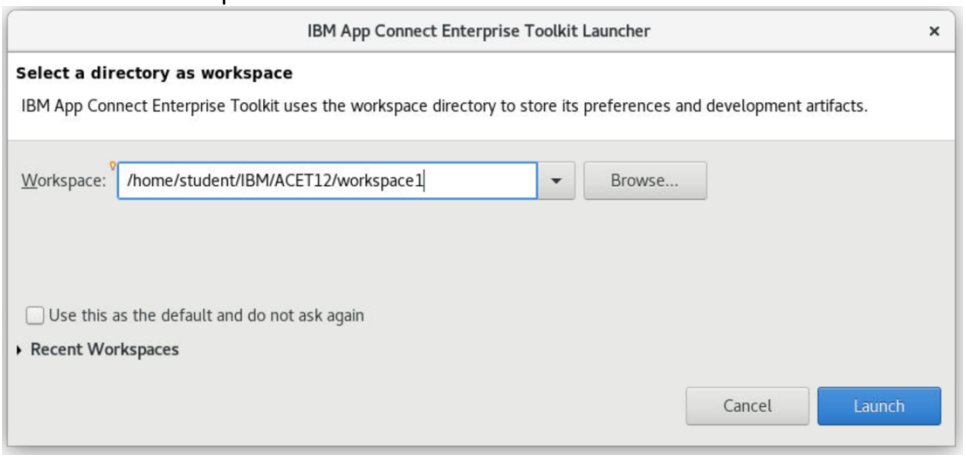
In this lab you will learned how to:

- Create and configure an Integration Server.
- Manage Integration Server from toolkit and Web Admin Console.
- Deploy BAR file in cp4i.

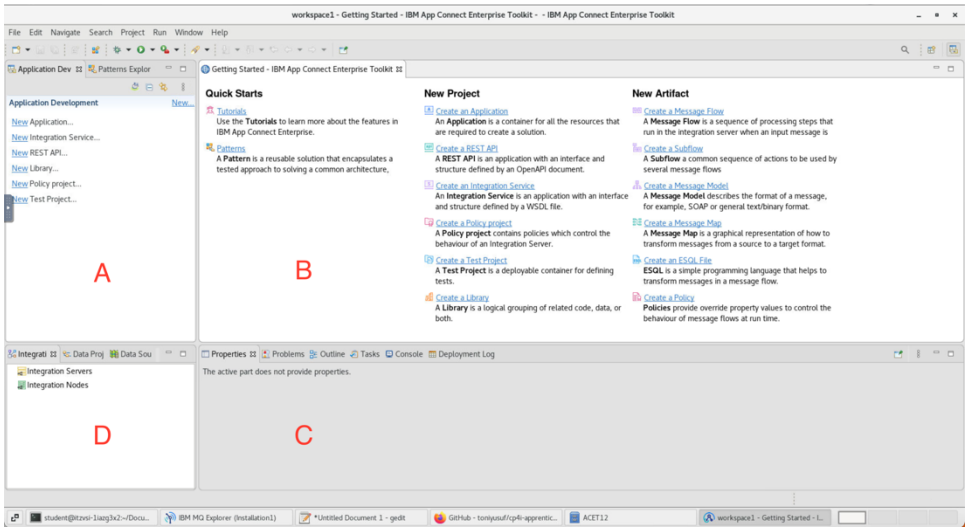
Step 1. Open the ACE Toolkit

1. In the terminal enter the following command
   `ace toolkit`

2. From the Workspace Launcher we will create a new workspace, for the work in this lab guide enter the Workspace name and click Launch.



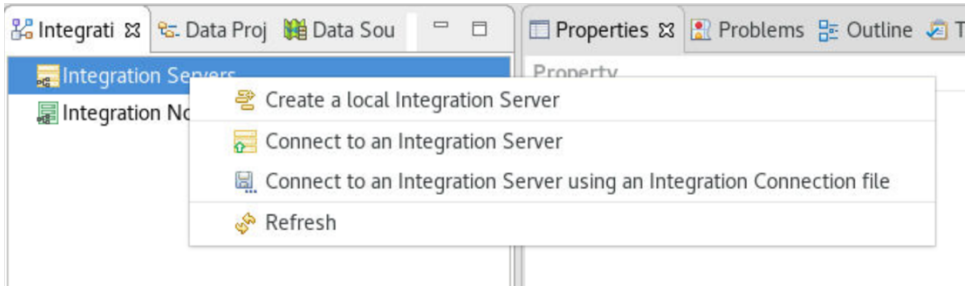
3. You may be prompted to Import Projects from File System or Archive, click Cancel
4. In the “Welcome to IBM App Connect Enterprise Toolkit” window click Close page
5. The Eclipse based **Integration Toolkit** will open:



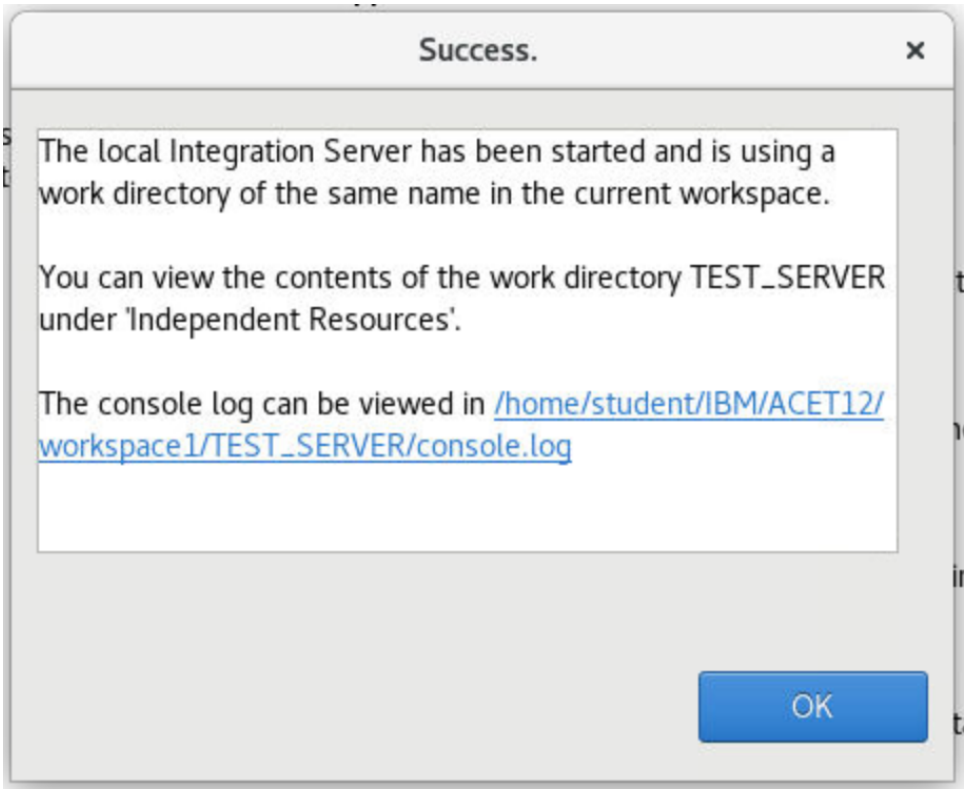
A) **The Application Development** window (A) is where your Applications, REST APIs, etc. will be shown in your workspace
B) Window (B) is where resources that you open (for example message flows) will be shown.
C) Window (C) is where properties of resources that you highlight in window (B) can be viewed
D) The **Integration Explorer** Window (D) is where you can view and manage deployed assets (for example Applications and Message flows).

Step 2. Create the Integration Server

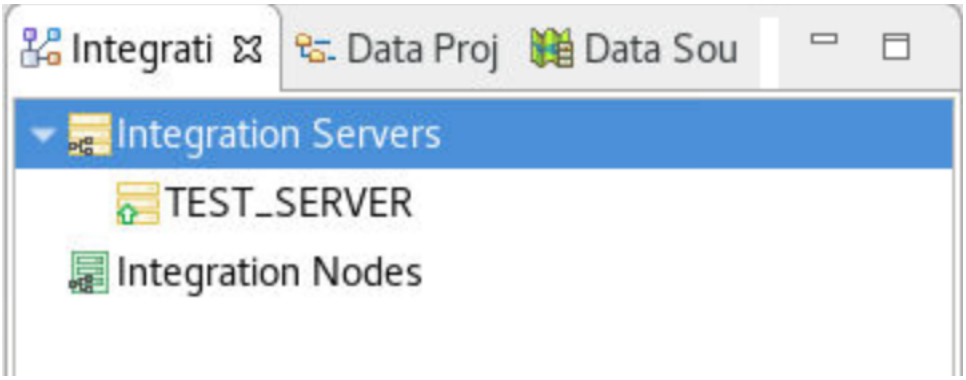
1. In the **Integration Explorer** window (D) right click on Integration Servers and select "Create a local Integration Server"



2. Accept the defaults in the ***"Create and start a local Integration Server"*** window and select Finish:
3. On successful start of the local Integration Server you will see a message similar to the following. Before dismissing the message , note the location of the console.log file:

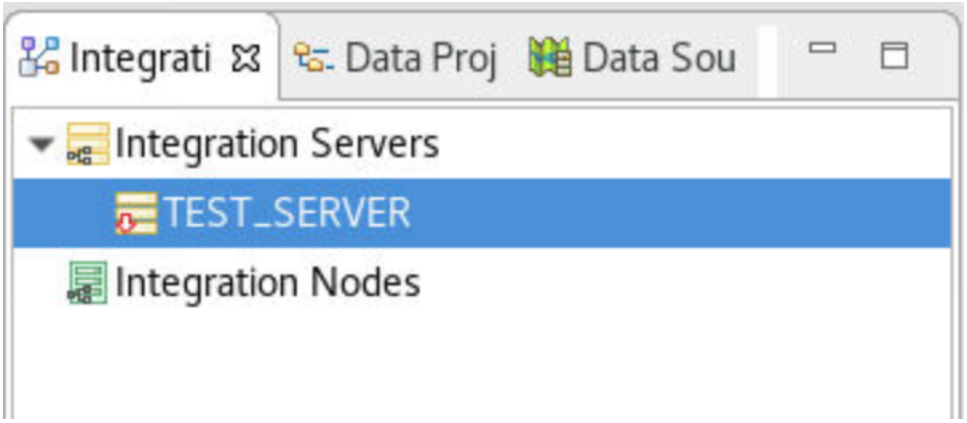


4. A connection to your new local Integration Server will appear in the Integration Explorer window – the green arrow (pointing upwards) to the left of the server name indicates that the server is running:

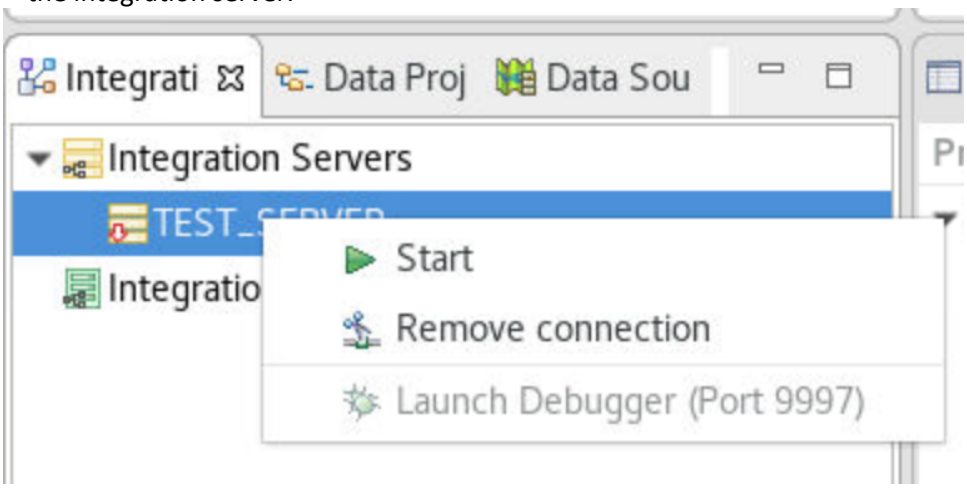


Step 3. Stopping and starting a local integration server

1. The integration server that you created in the previous section will be up and running (it will have a green arrow facing upwards in the **Integration Explorer** window. Right click on the server name and select Stop (this will stop the integration server):
2. A message will appear explaining that the integration server is shutting down:
3. When the integration server has shutdown, the **Integration Explorer** will show the TEST_SERVER with a red arrow pointing downwards:



4. In the **Integration Explorer** right click on the integration server and ***note*** the option to start the integration server.

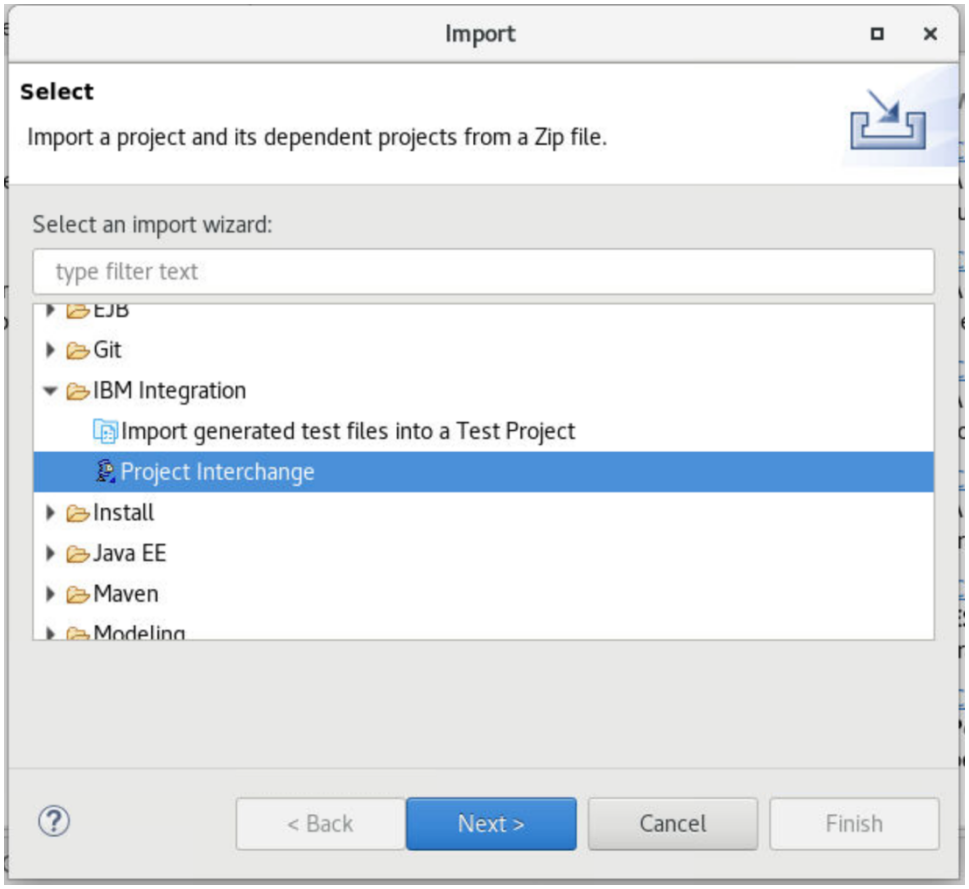


Step 3. Deploy a test application

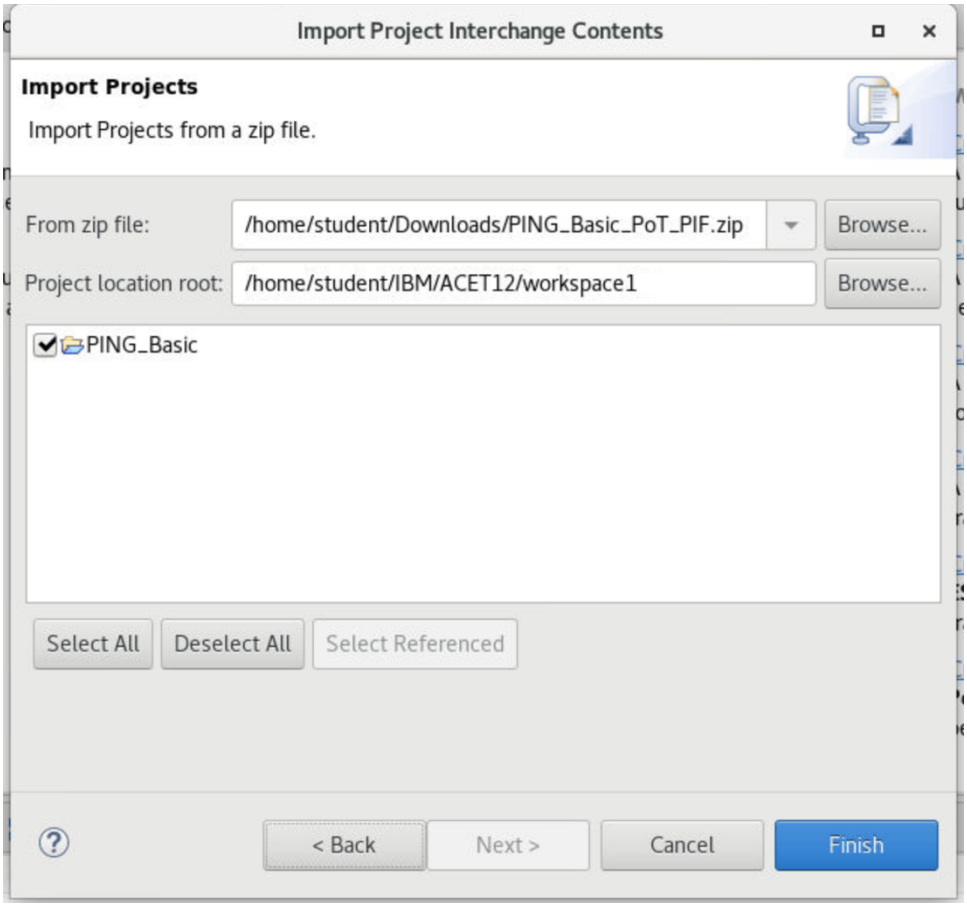
You now have a integration servers running: TEST_SERVER running with defaults settings; You will now review a very simple application called PING_Basic and deploy it to integration servers you have running in your environment.

Now download the **PING_Basic_PoT_PIF.zip** Click here and save the zip file -
PING_Basic_PoT_PIF.zip

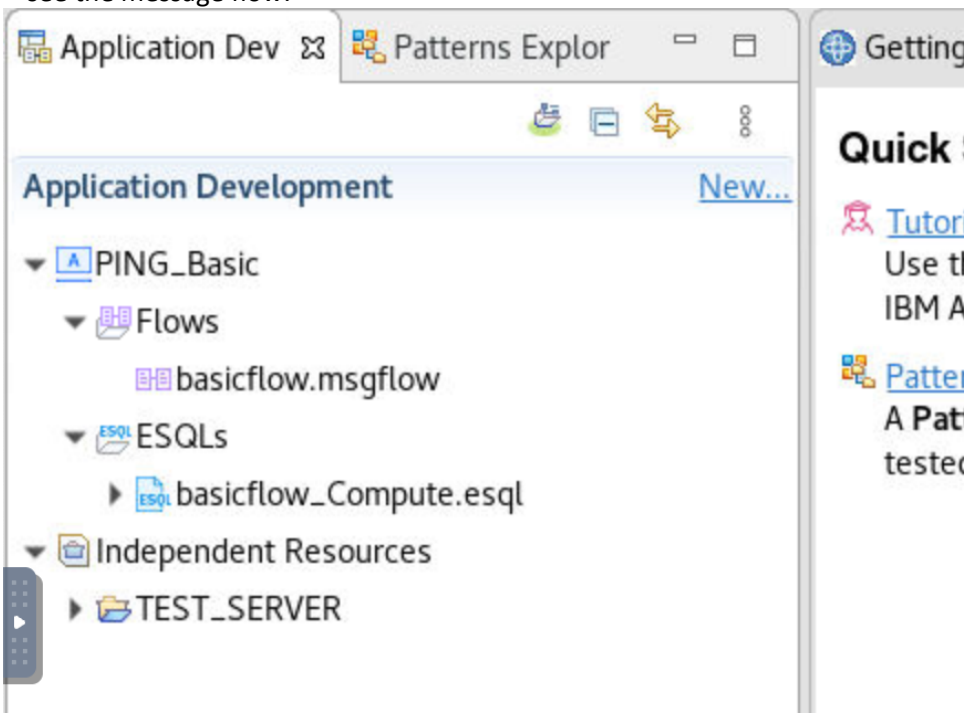
1. With your mouse right click on the background of the Application Development window and select “Import”
2. Select IBM Integration > Project Interchange then click the Next button:



Use the browse button to import the file **PING_Basic_PoT.zip** from where you had downloaded it.
Click Finish:



3. The **PING_Basic** Application will be imported into your workspace, expand the application to see the message flow:



4. Double click on basicflow.msgflow, this will open the message flow in window **(B)**. This flow has two paths. For this lab we will be focused on the HTTP Input.

5. Double click on the node called “Compute” to see the data that will be returned from the http request when the message flow us started:

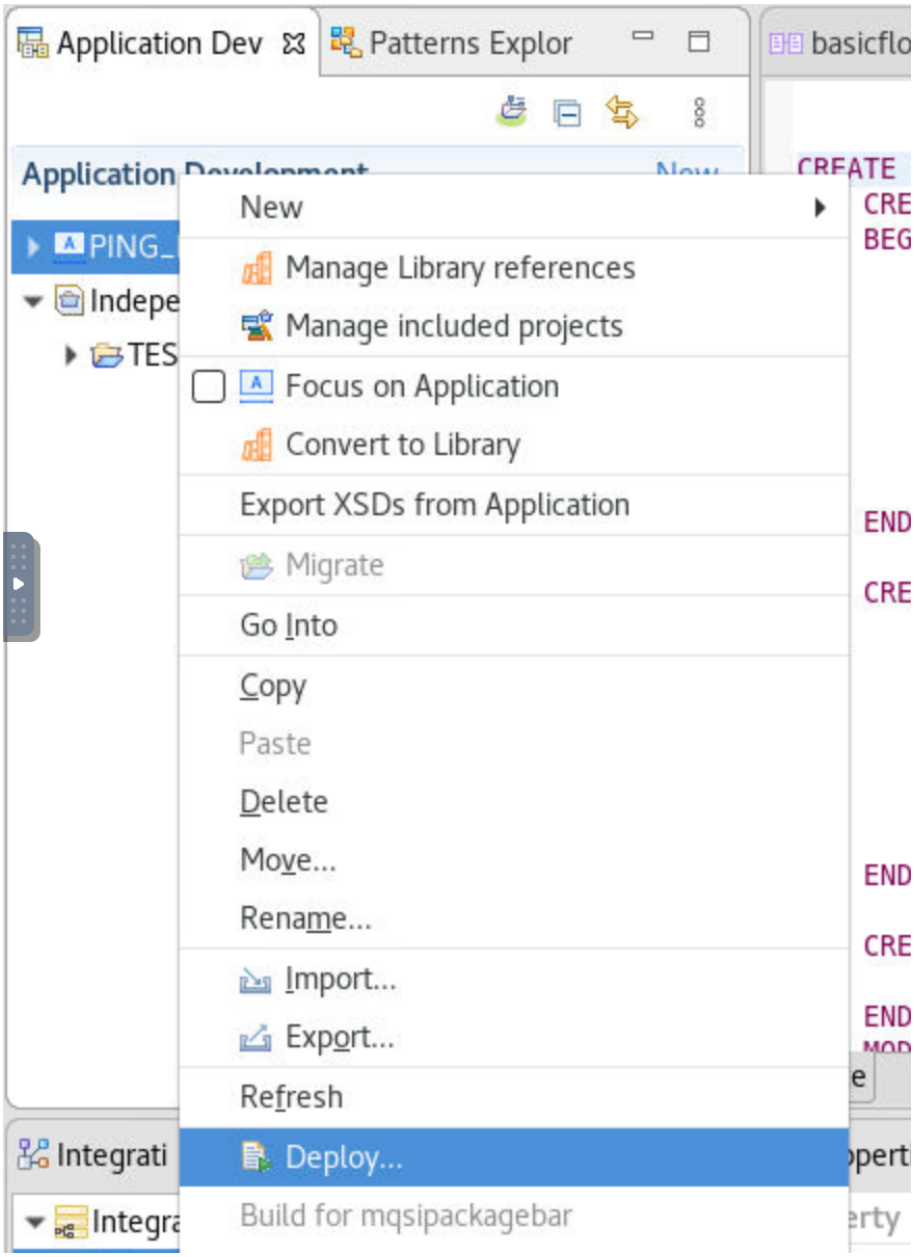
   The flow will return the following:

```
Set OutputRoot.JSON.Data.pingbasic.Server = ExecutionGroupLabel;Set OutputRoot.JSON.Data.pingbasic.WorkPath = WorkPath;Set OutputRoot.JSON.Data.pingbasic.MsgFlow = MessageFlowLabel;Set OutputRoot.JSON.Data.pingbasic.DateTime = CURRENT_TIMESTAMP;
```

   i.e. The server name that the flow is running on; the WorkPath of the server; the message flow name; the current time stamp;

6. Close the ESQL editor and the message flow without making any changes

7. Right click on the **PING_Basic** application and select **Deploy**



8. When prompted to choose an integration server, deploy the application to **TEST_SERVER**
9. Review the deploy messages and dismiss the Progress information window by pressing the close button

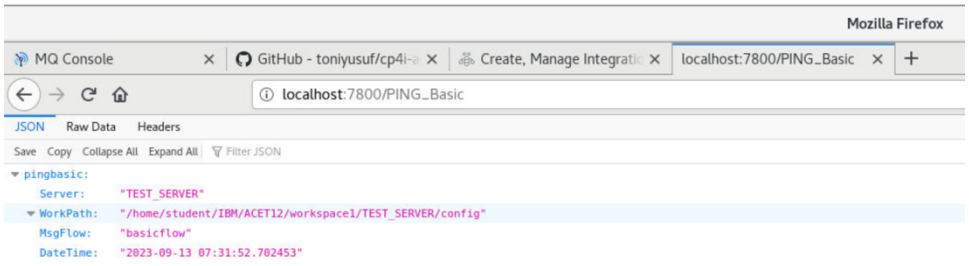
Step 3. Deploy a test application

1. Open a new tab in your Firefox browser window, and go to the following URL:

   http://localhost:7800/PING_Basic

   the request should return details of TEST_SERVER:

Lab 2 - Create, Manage Integration Server and Deploy & Test message flow



Congratulations you have just created an Integration server, deployed an application and tested it!