Extend your back-end integrations securely to partners and developers.

Extending access via APIs to your back-end integrations empower your partners and developer community to create new business value, technical value, and customer experiences for your products and offerings. Spur innovations where a number of technologies are combined to create something new, for example, extending the ability apply for loan pre-approvals that can be used within apps that search for cars or real estate. To do this, you must first create the back-end integrations, which combine data from existing core systems, disparate assets, or SaaS services with the ability to send critical data between systems reliably. Second, you need to provide APIs to your back-end integrations that secure access and apply rate limits.

In this tutorial, you learn how to deploy an app integration flow that takes data from one source and sends it to a message queue for reliable delivery. Then, you expose this integration as a rate-limited API secured by a key and secret. This integration flow is deployed **quickly and easily** as an independent, **auto-scalable microservice** running on **containers**. By using RedHat OpenShift as a foundation.

In this tutorial, you will explore the following key capabilities:

- Explore multiple integration capabilities within a single platform.
- Create a queue using MQ Console
- Create an integration flow between a public cloud service and on-prem message queue.
- Deploy the integration flow as a microservice using Helm.
- Provide access to the integration flow as a secure API.
- Explore Operations Dashboard for APIC, IBM MQ and App Connect Enterprise.

Task 1 - Start the Environment

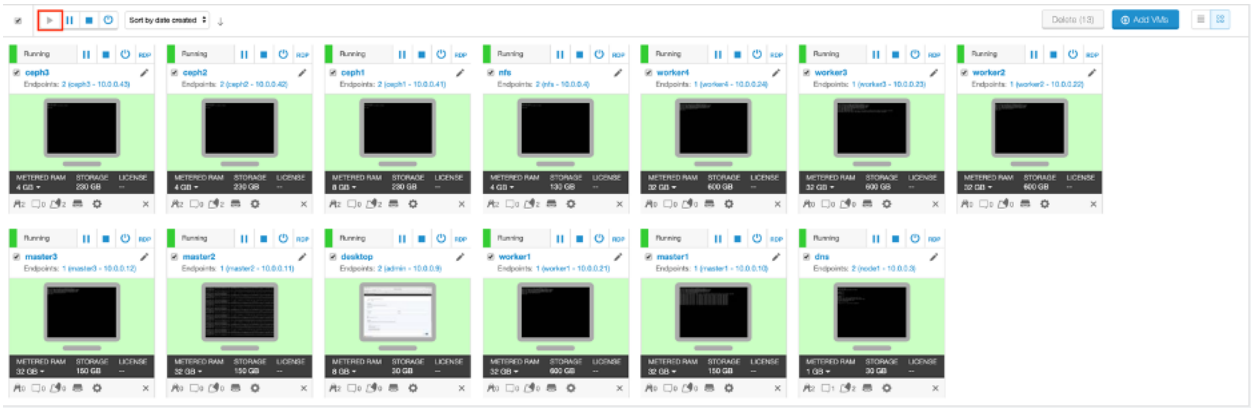
As this is a new deployment of the Cloud Pak for Integration, you must execute some steps to prepare the environment. Initial setup steps

are only needed for a fresh installation of the platform. They do not need to be repeated.

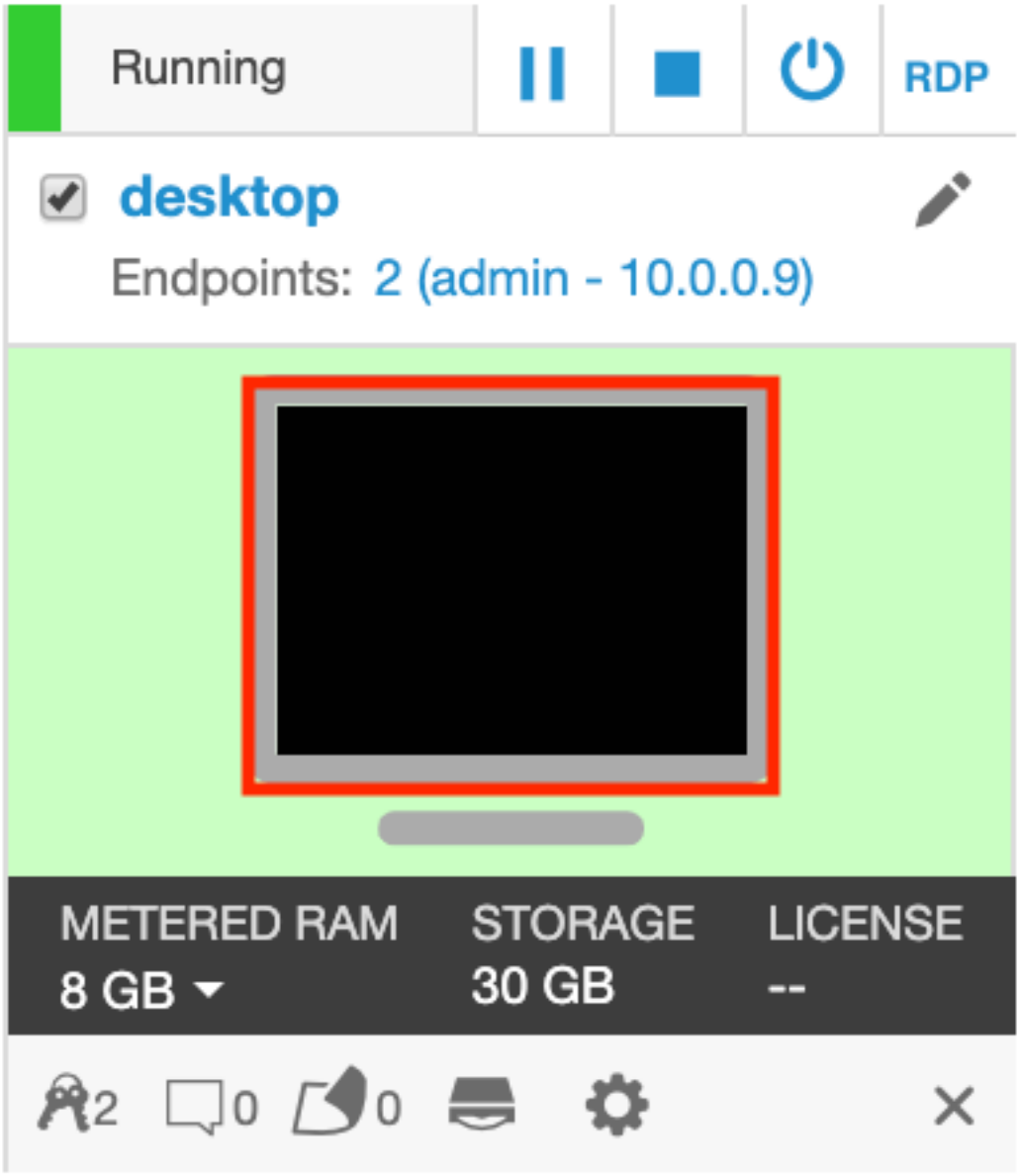
All work for this lab is done on the Developer Machine. Open the **developer Machine** VM by clicking the tile.

1. If the environment is already up and running when you open your reservation link, skip to step 3. If it is not running, continue to step 2.

2. **Click** the Run VM(s) button as shown below to start the virtual machine environment that is used for this lab.



3. Once the virtual machine has started, click the **desktop** Machine tile to start your lab exercise.

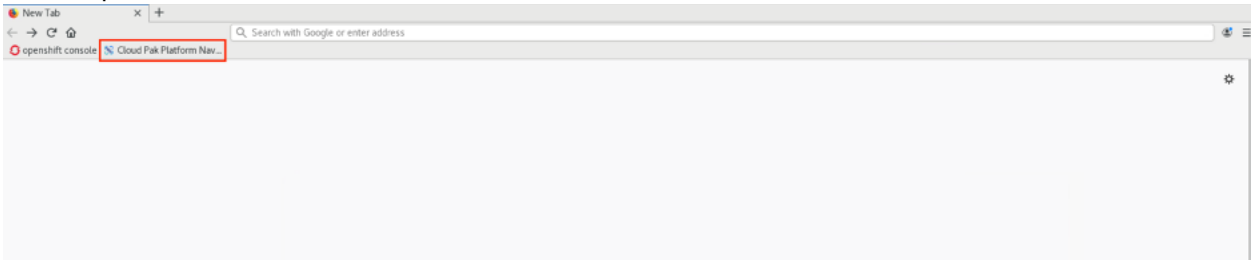


4. Log in as user **ibmuser**, password **engageibm**

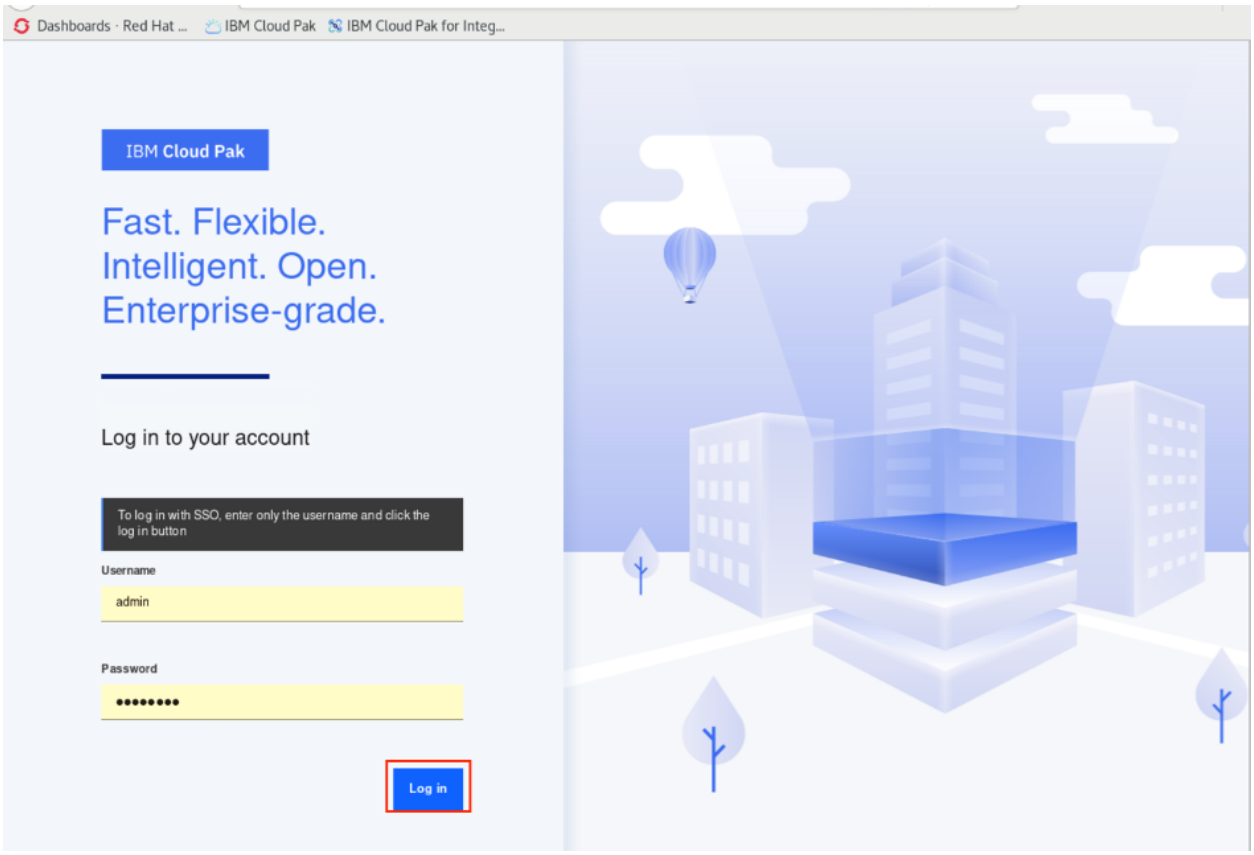
Task 2 - Configure Message Queue (MQ) to Authorize and Accept Data. As this is a brand-new deployment of the Cloud Pak for Integration, all instances of integrations, message queues, and event streams are deployed as microservices. We need to authorize the Message Queue service to accept incoming data from the integration running on a separate server.

Cloud Pak for Integration provides a single solution for all of your enterprise integration needs. The platform provides a comprehensive set of industry-leading capabilities. Use any of them on their own or together through a single interface. Create, manage, and monitor all of your integrations across SaaS applications, messaging, streams, APIs, high-speed transfer, and more.
Unlock the power of your data and support the scale required for all of your integration and digital transformation initiatives.

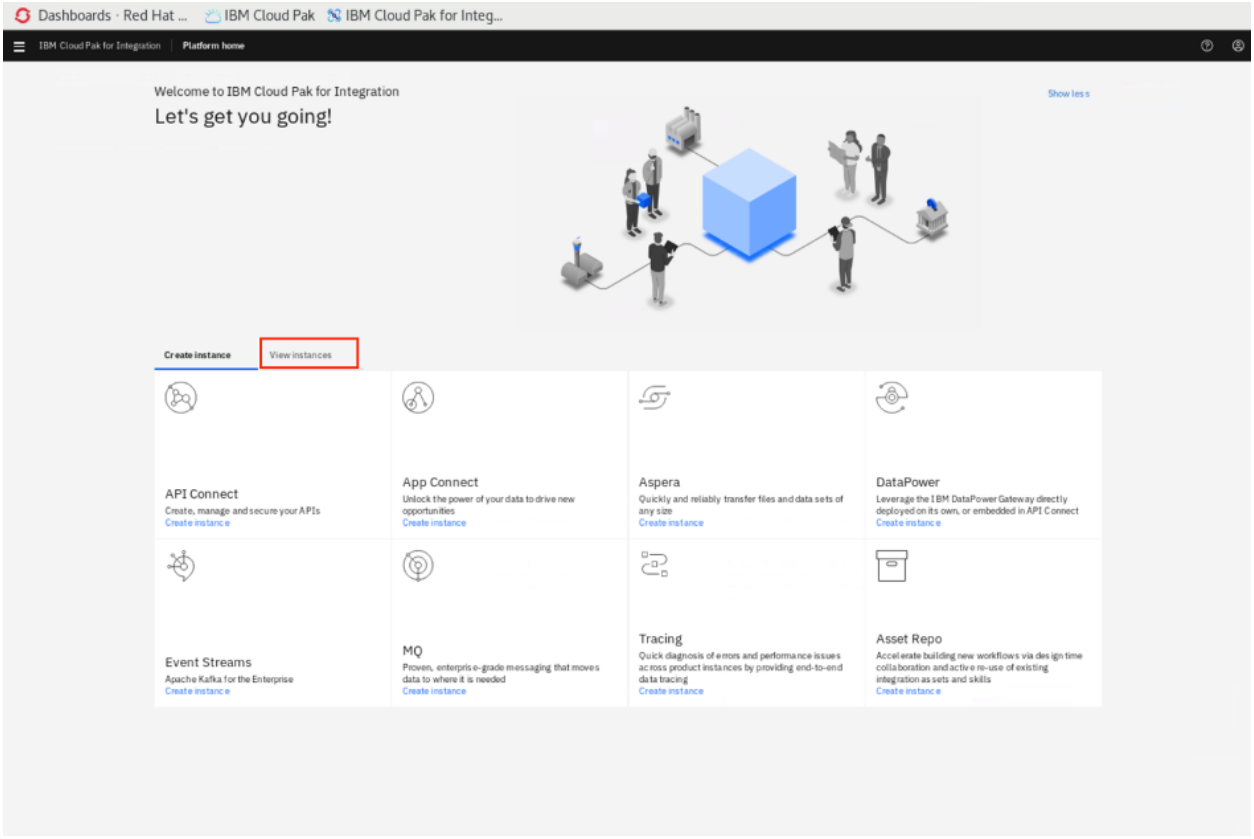
1. Click the IBM Cloud Pak for Integration bookmark in the bookmarks bar at the top.



2. The login screen for IBM Cloud Pak might be displayed, **you don't need to enter** username: **admin** and Password: **passw0rd**. Click **log in**.



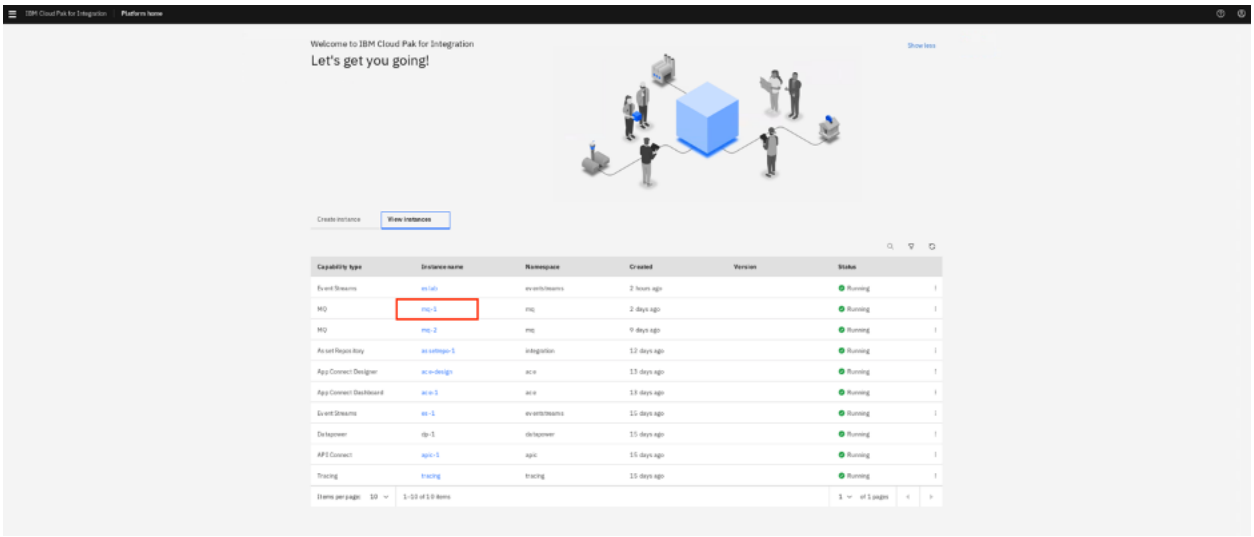
3. After logging in, the **Cloud Pak for Integration** home page is loaded. From here you are able to navigate to all the integration and development technology contained within the platform. Today, this technology includes: API Connect, App Connect integration, Aspera, DataPower, Event Streams, MQ, Tracing and Asset Repo . Click the **View Instances l**ink.



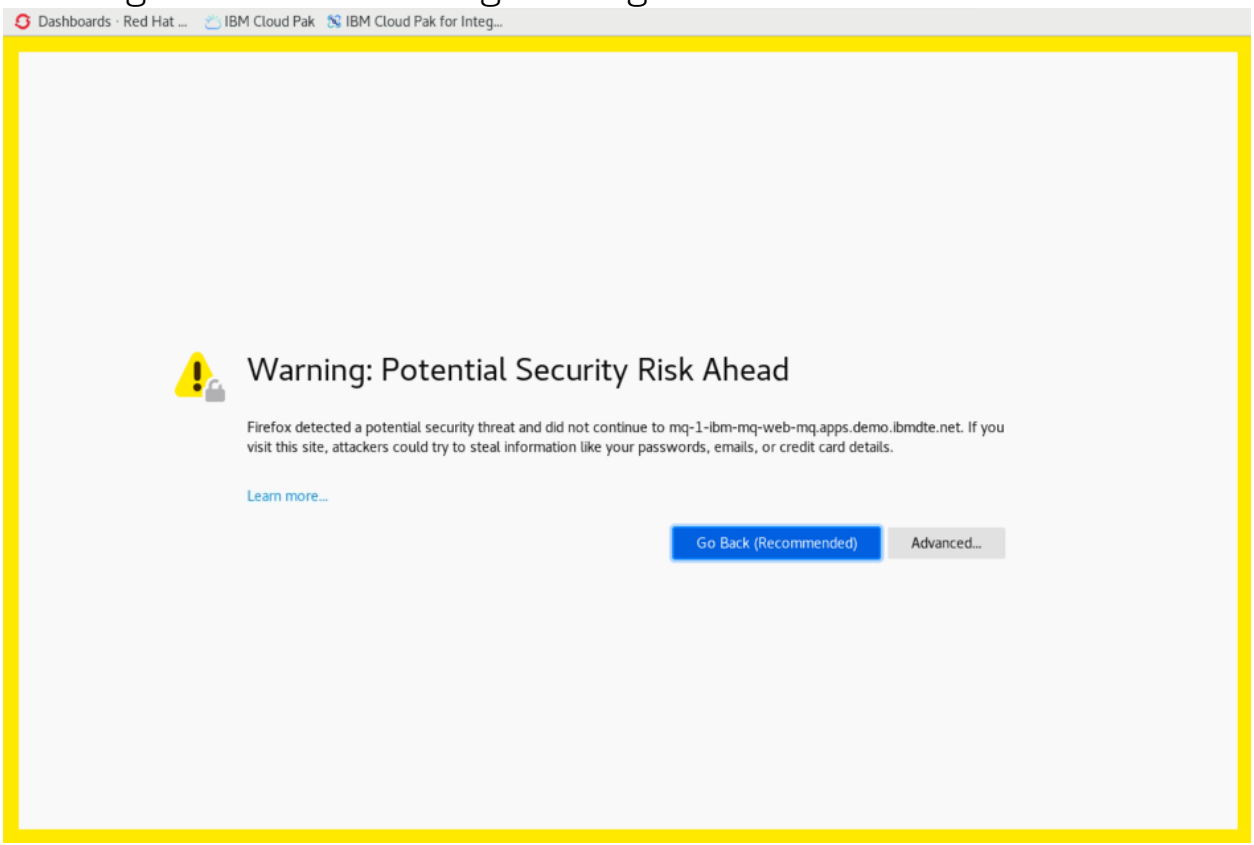
Task 3 - Creating a queue in MQ

This task covers administering and creating a new queue in MQ. MQ for Cloud Pak for Integration has a Web GUI, which the Integration Developer, with security authorization, will be able to manage the different MQ objects (queues, channels, topics and so on)

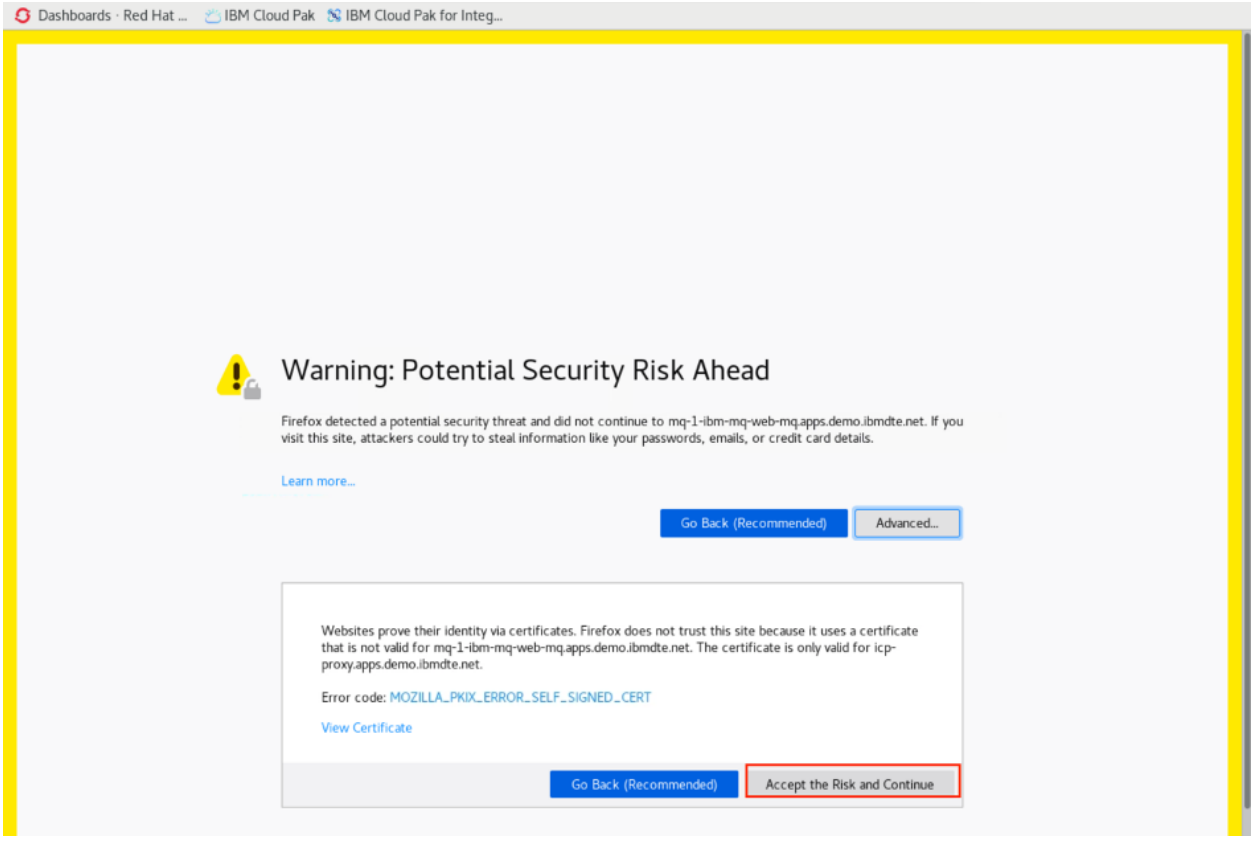
1. Click in **View Instances** and then menu, **mq-1**, instance to open MQ console.



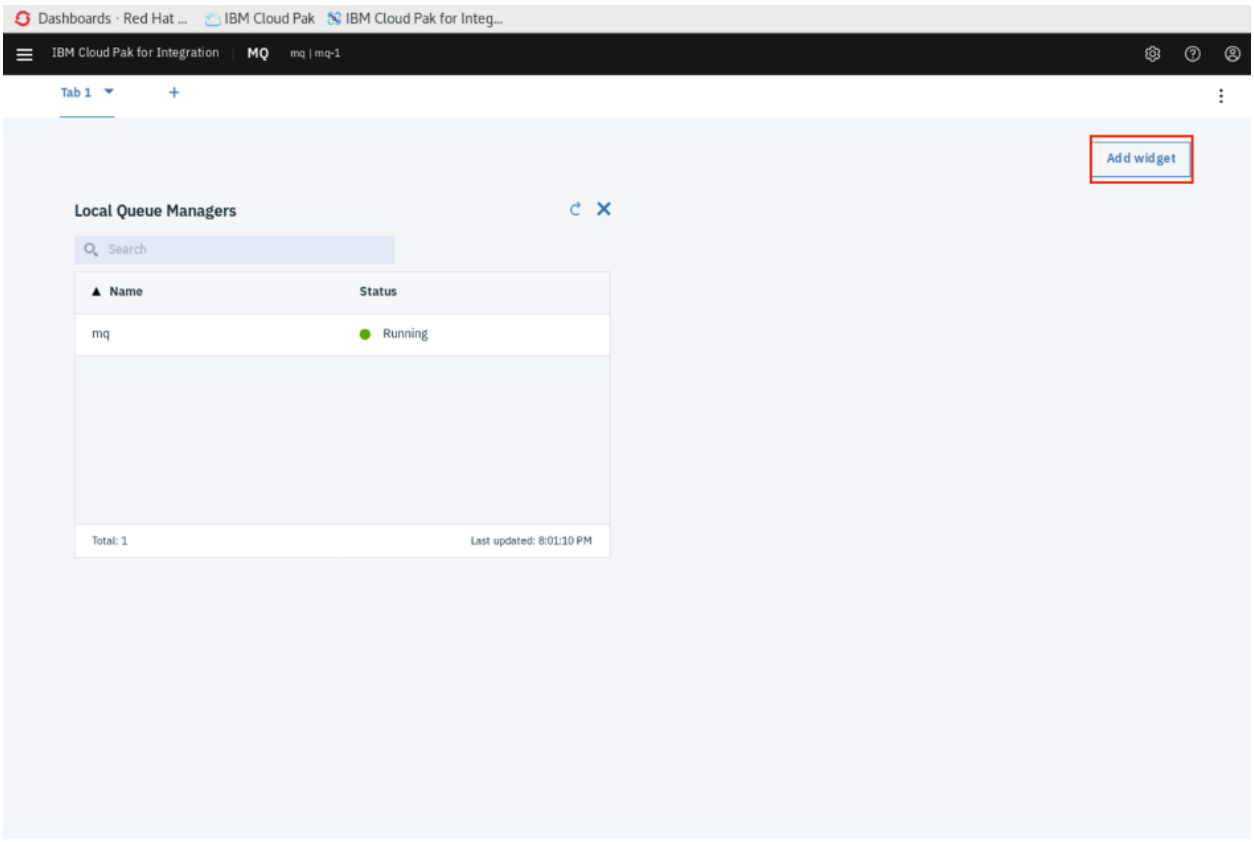
You might receive a warning message. Click **Advance**.



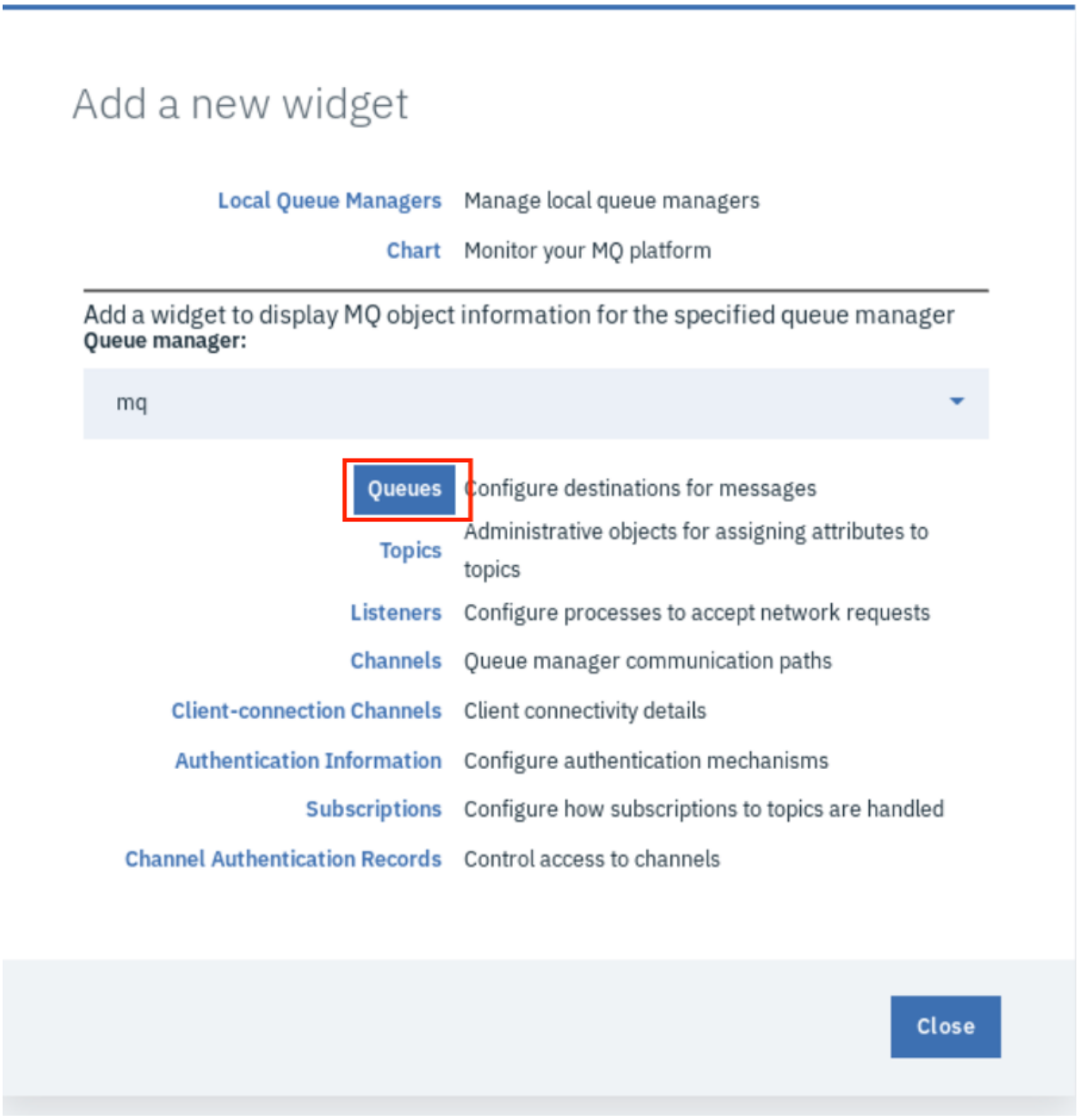
Click **Accept the Risk and Continue** in the Firefox browser. Click the **Proceed (unsafe) link** in the Chrome browser.



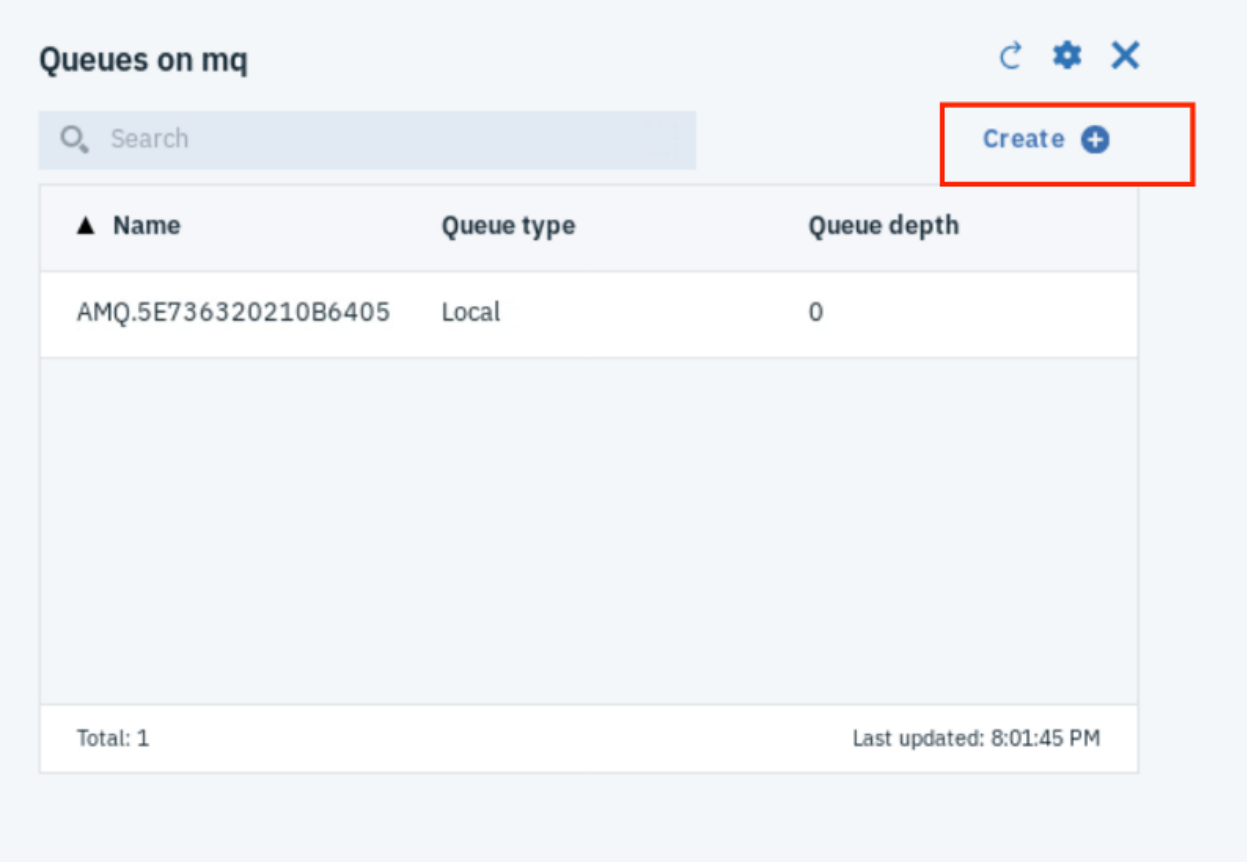
2. You see the MQ Console and MQ server running as **mq.** You can add more widgets for more information on MQ objects, click **Add Widget** button.



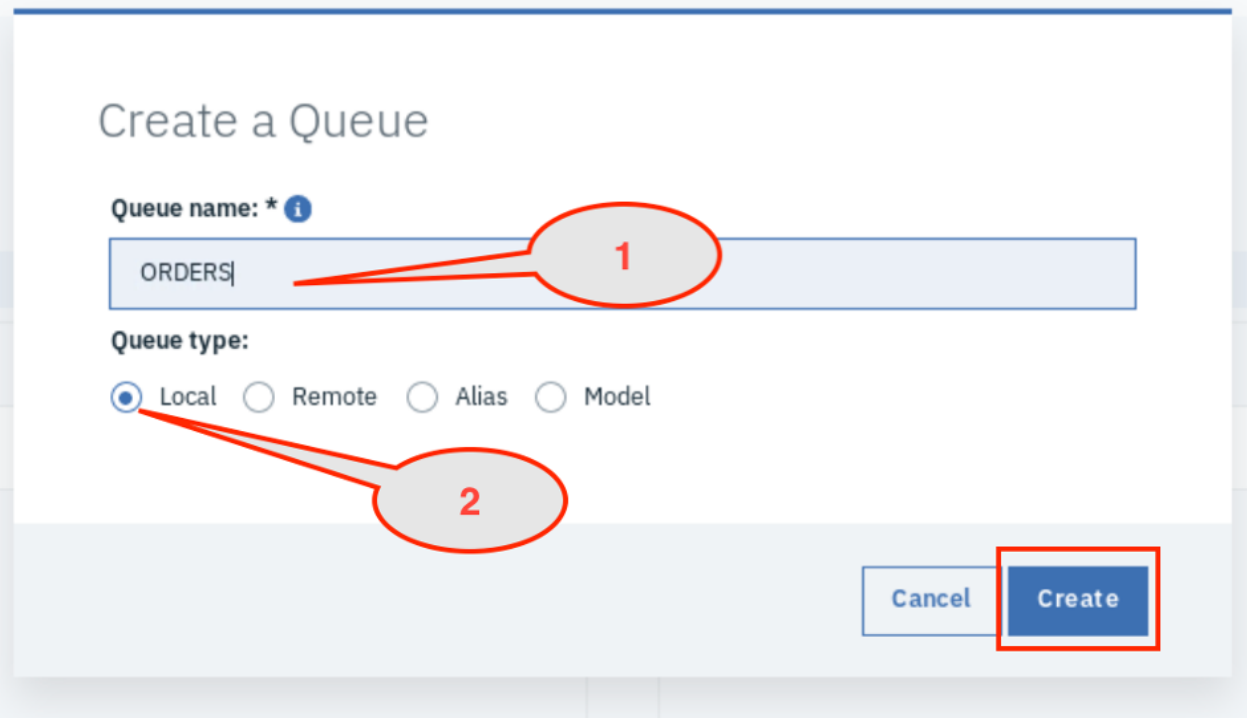
3. In the window, you can select the **Queues** widget.



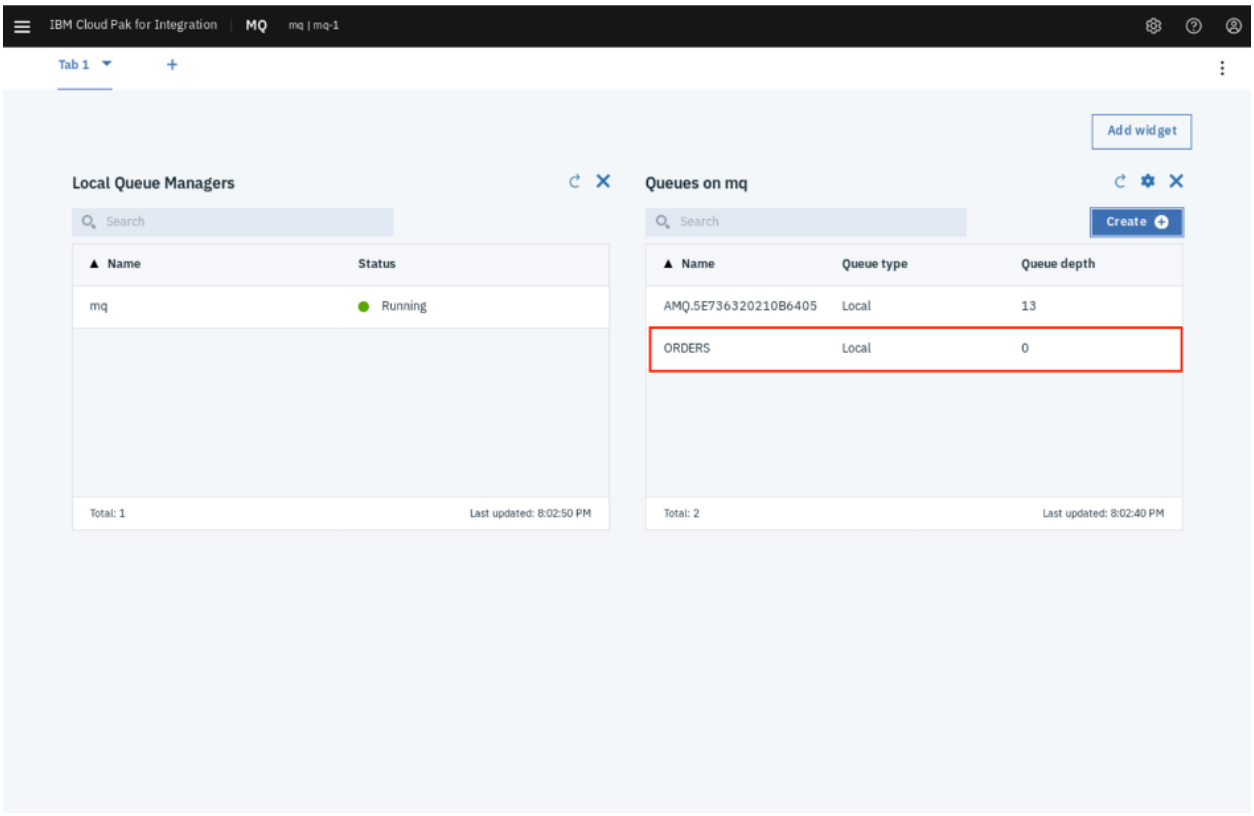
4. You see the queues window.You see queue names, type and queue depth. Click **Create (+)** to create a queue.



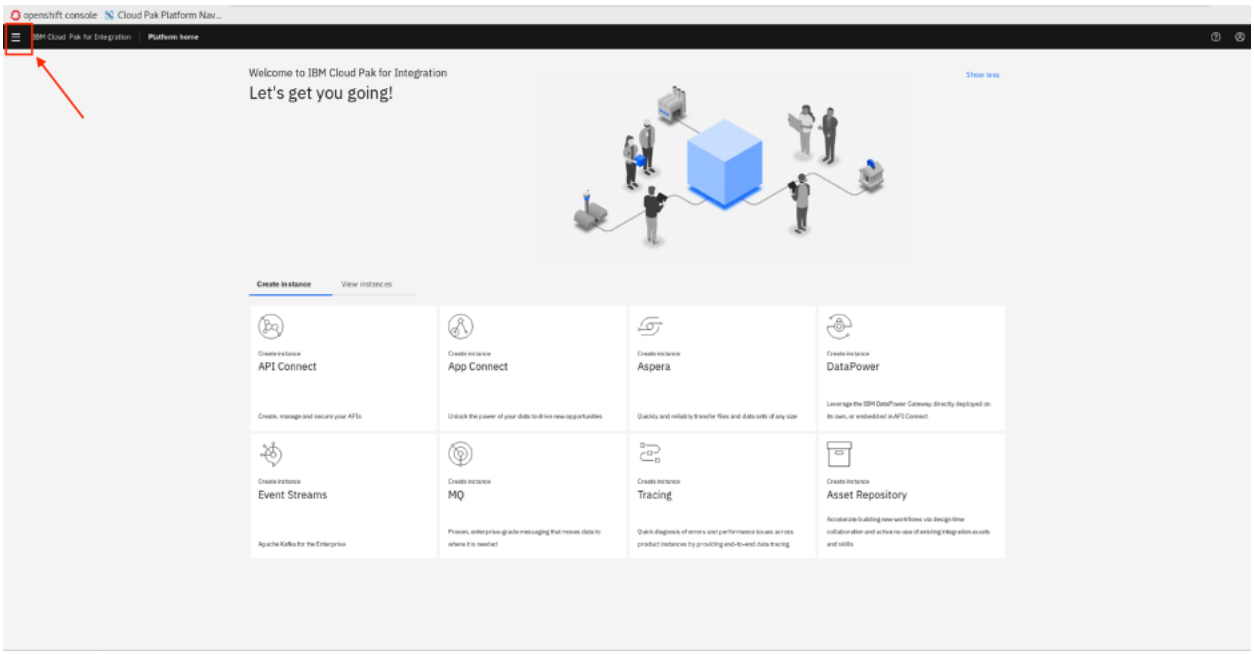
5. Enter the queue name: **ORDERS (this is case-sensitive).** Here you are able to create Local, Remote, Alias and Model queue. In our lab check **Local.**



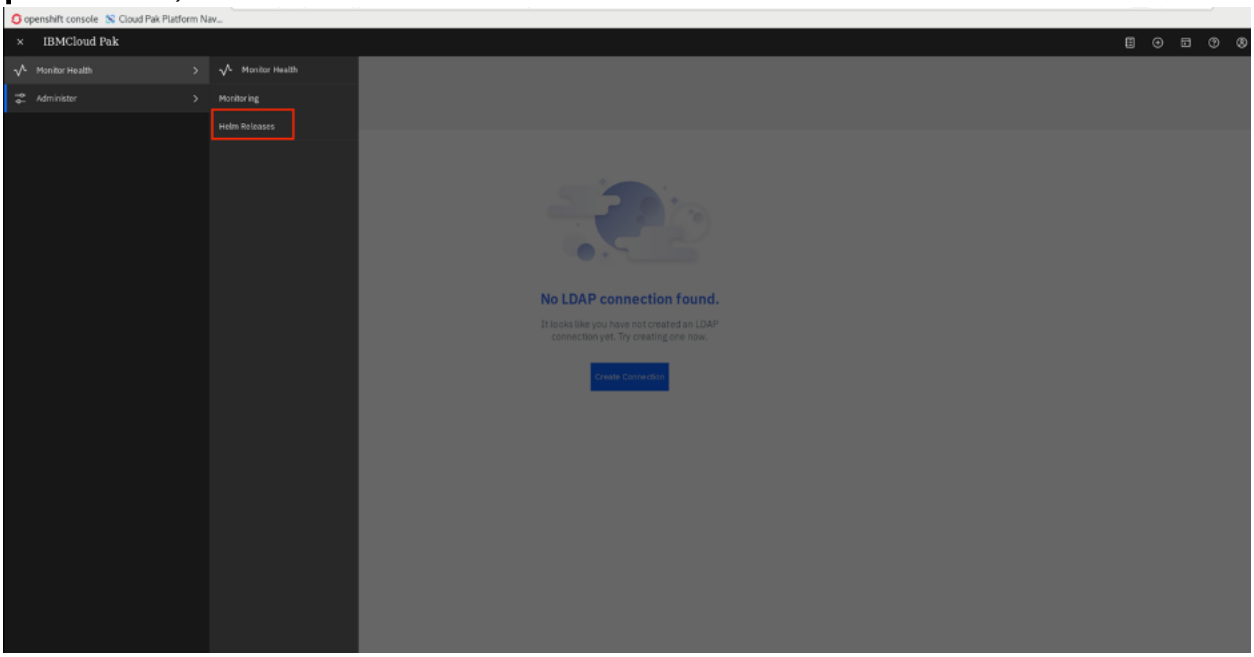
6. Look at **Queues on mq** window and check the queue created (Local and Queue depth).



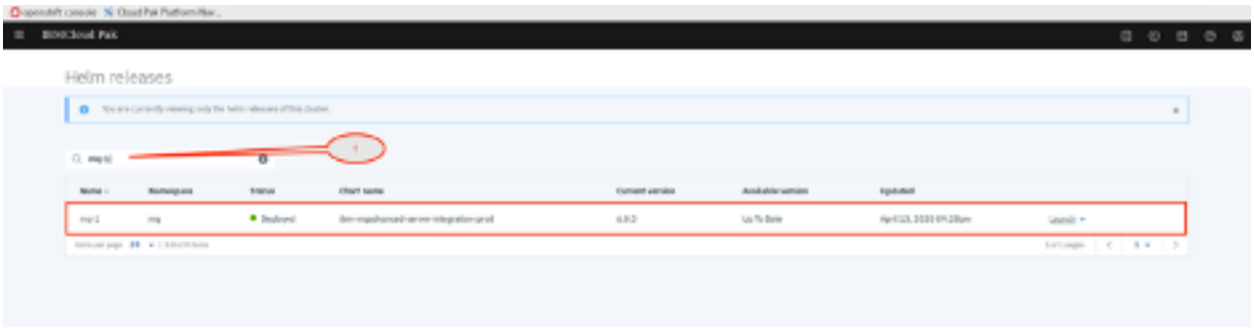
7. You need additional information about the MQ connection, click the “Hamburger” menu and then select **Cloud Pak Foundation .**
   Note: You can navigate using the **Hamburger Menu** or **View Instances** in the Cloud Pak Navigator page



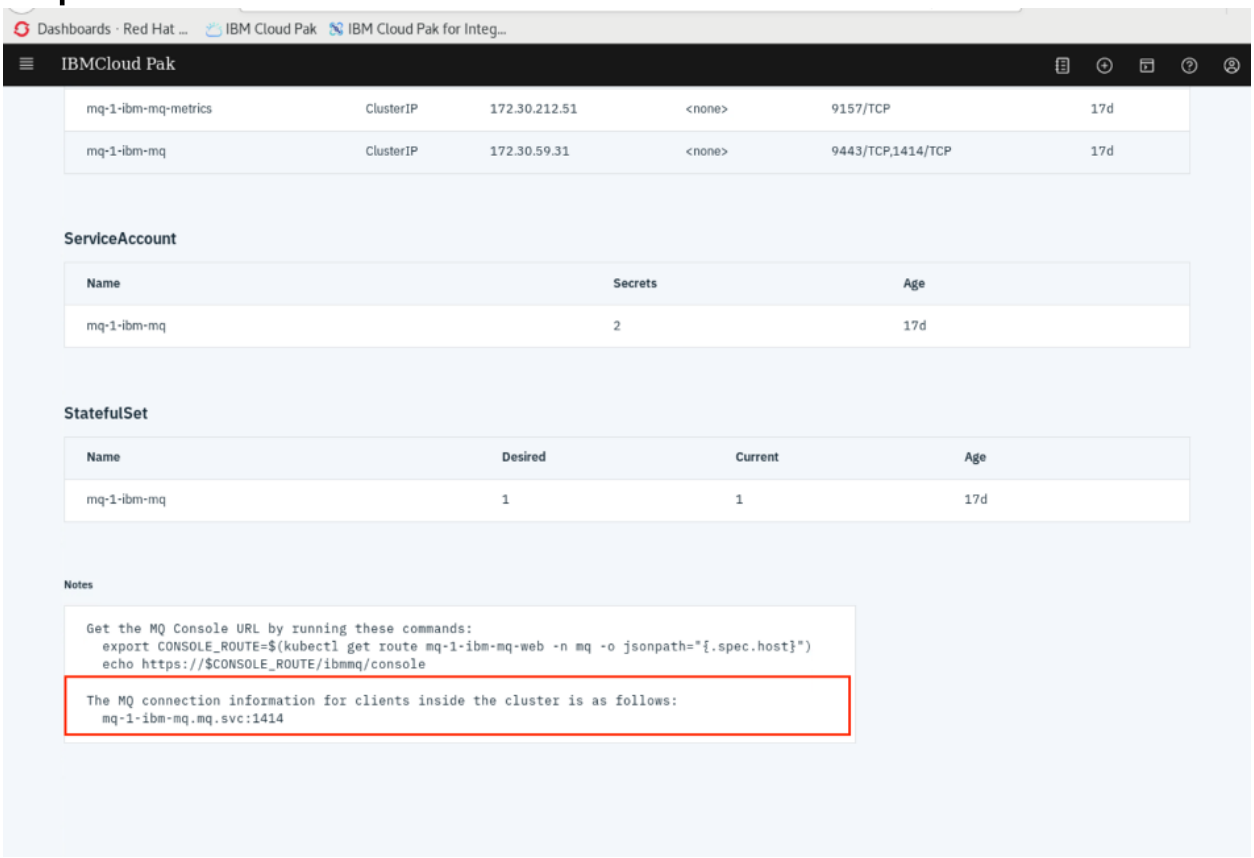
8. Open “**Hambuger**” menu again and select **Administer->Helm Repositories**) .



9. On the **Helm Releases** page, type **mq** in the search box. The search finds mq helm releases. Click the line of **mq-1.**



10. Scroll down to the bottom of the page and check the MQ connection. You see the host address: **mq-1-ibm-mq.mq.svc** and the port **1414**. App Connect Enterprise will use a mq connection client for **mq-1 .**

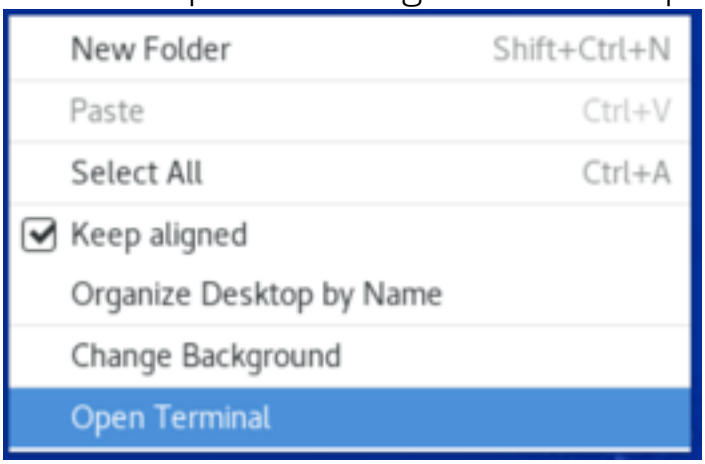


Task 4 - Configuring the app integration flow

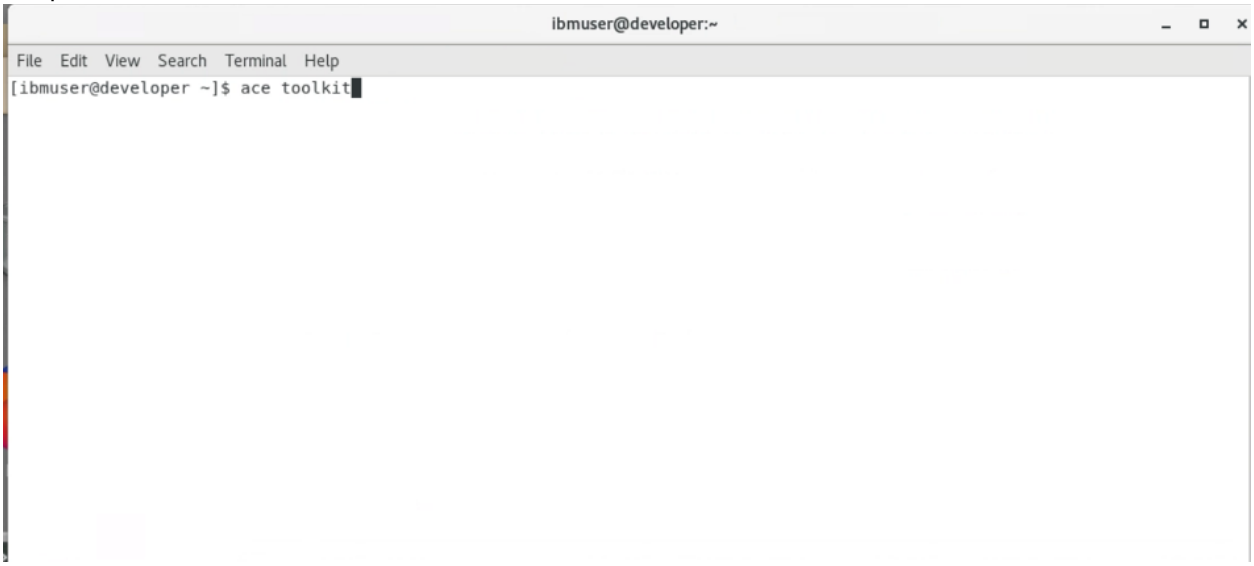
This task covers opening and examining an application integration flow in the IBM App Connect Enterprise Toolkit. With the Toolkit you can build powerful and complex integration applications, services, and

APIs quickly and easily using a visual designer. Your integration solutions can be directly deployed to the Cloud Pak for Integration on IBM Cloud Pak running on-premise, in any cloud, or combinations of both.

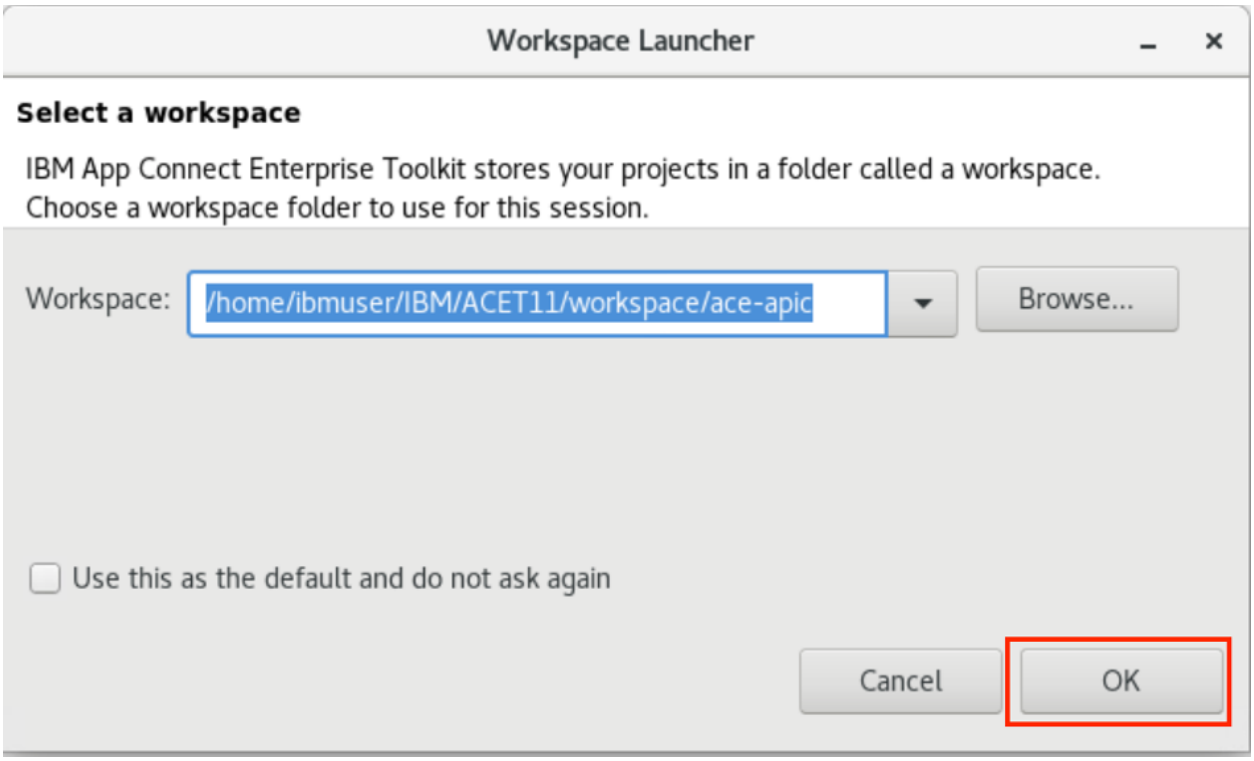
1. In the desktop click the right mouse to open a terminal window.



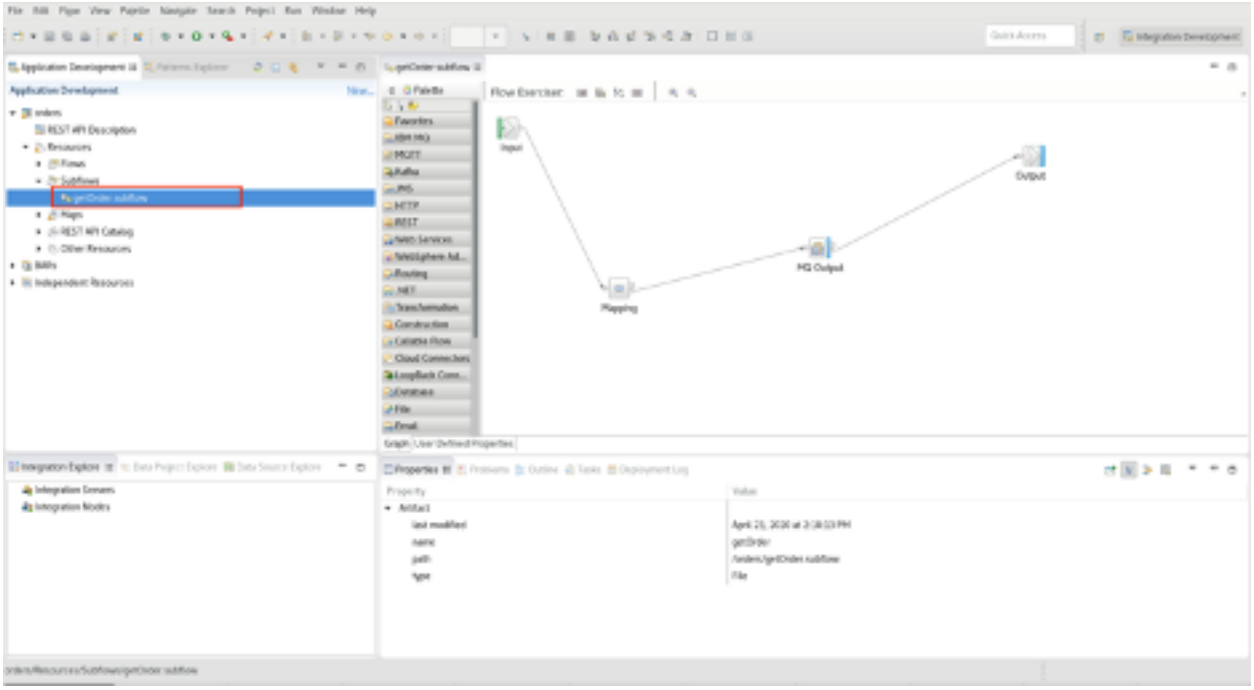
2. In the terminal window, type **ace toolkit** to open the App Connect Enterprise Toolkit.



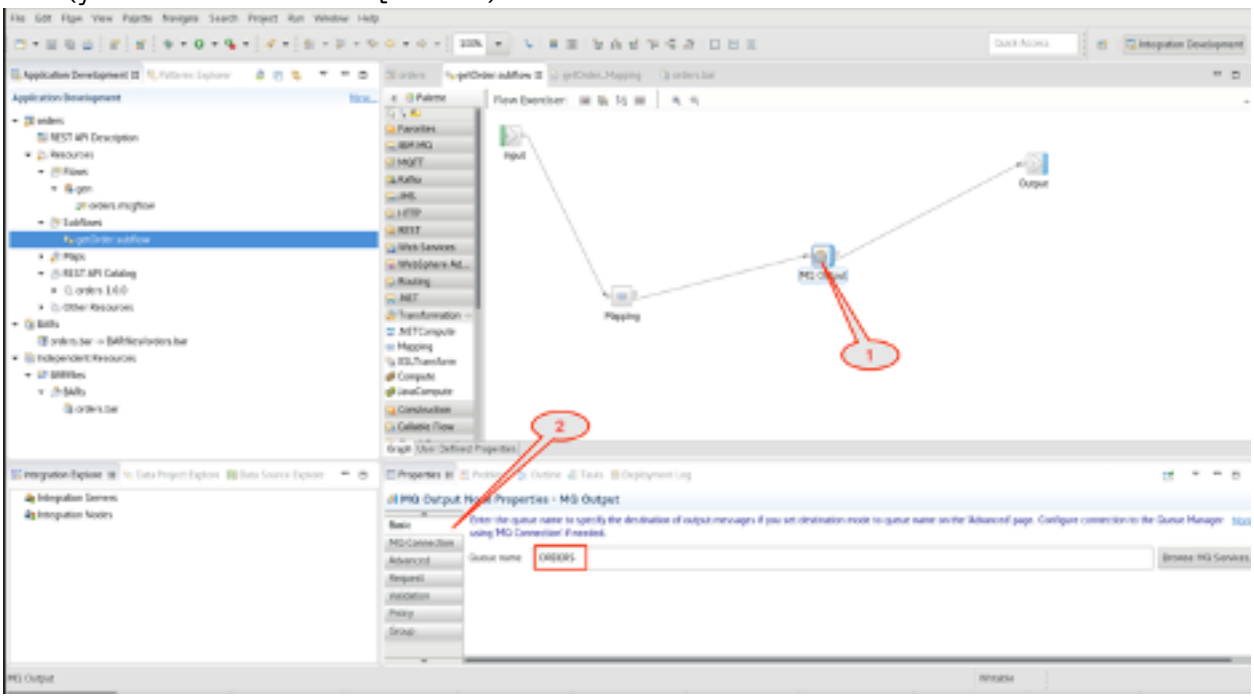
3. In the Workspace Launcher window, choose the workspace /home/student/IBM/ACET11/workspace/ace-apic.Verify the path routes to the folder: **ace-apic.** Click **OK**.



4. The toolkit opens the project. To view the integration flow that you deploy, click **orders -> Resources -> Subflows -> getOrder.subflow**



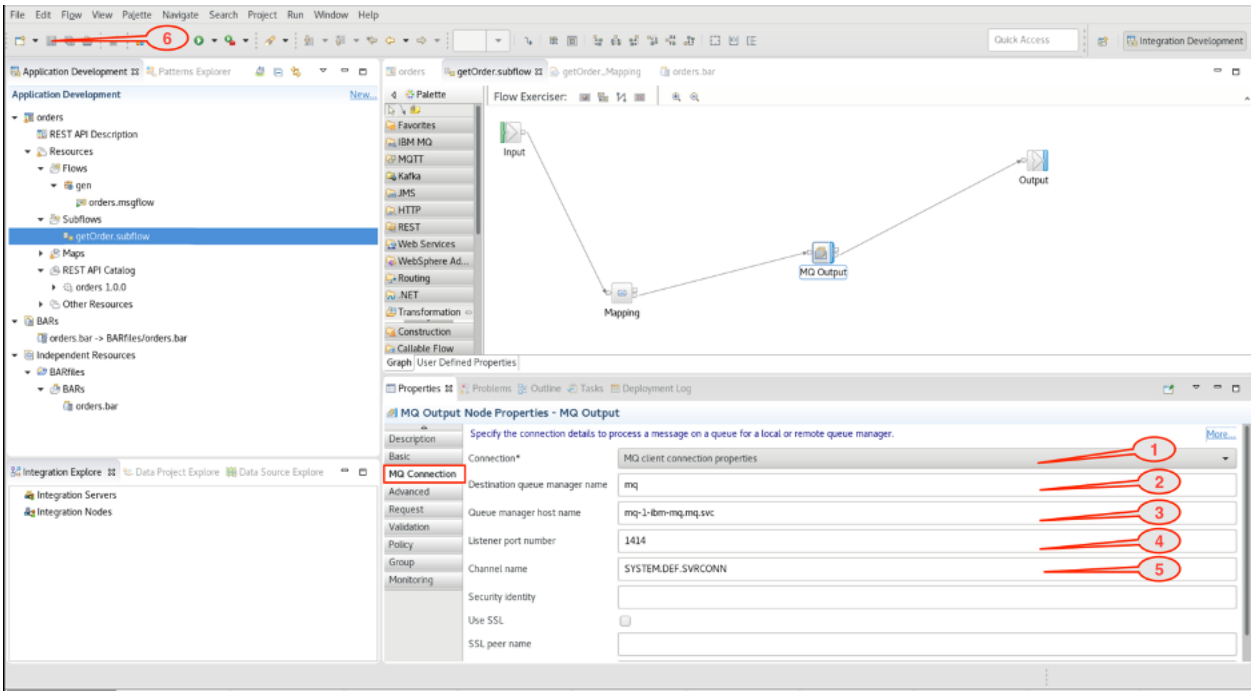
5. Check the MQ node by clicking the **MQ Output Node.** Click **Properties** and select **Basic**, If necessary type the queue name (you created in MQ Task) **ORDERS .**



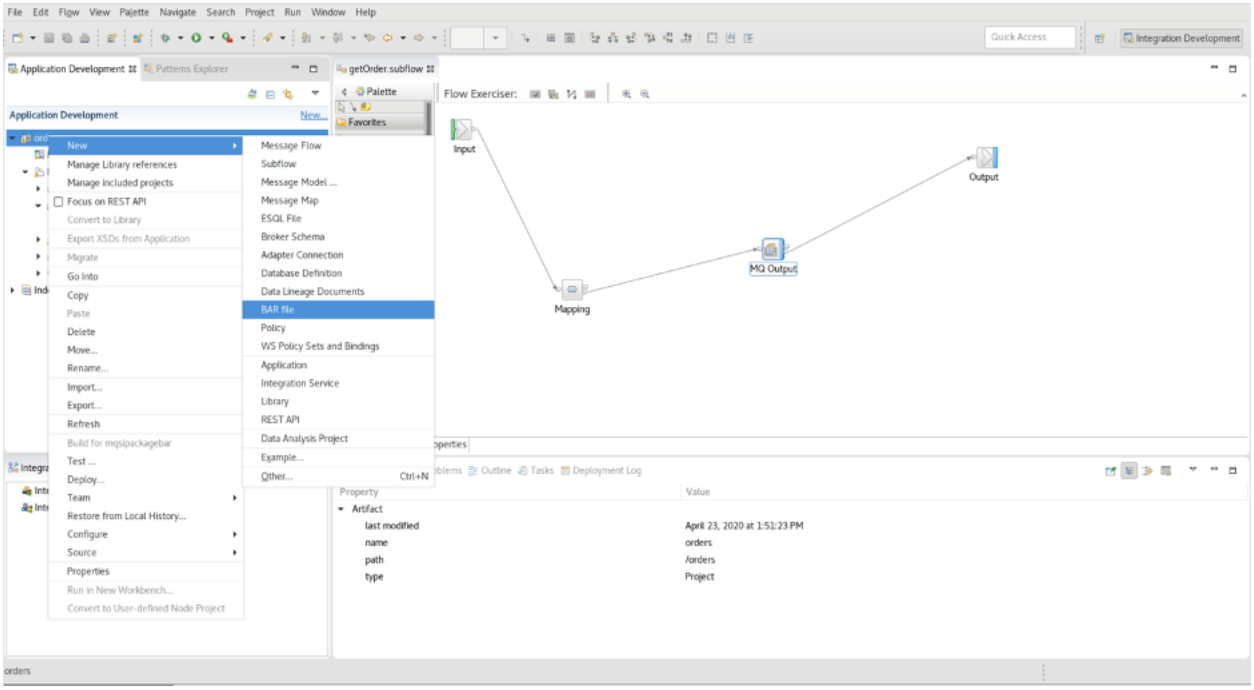
6. You have configured the queue name. You need to configure how the App Connect Enterprise server connects to the MQ server. Click **MQ Connection.**

   Obs: All parameters are case-sensitive.

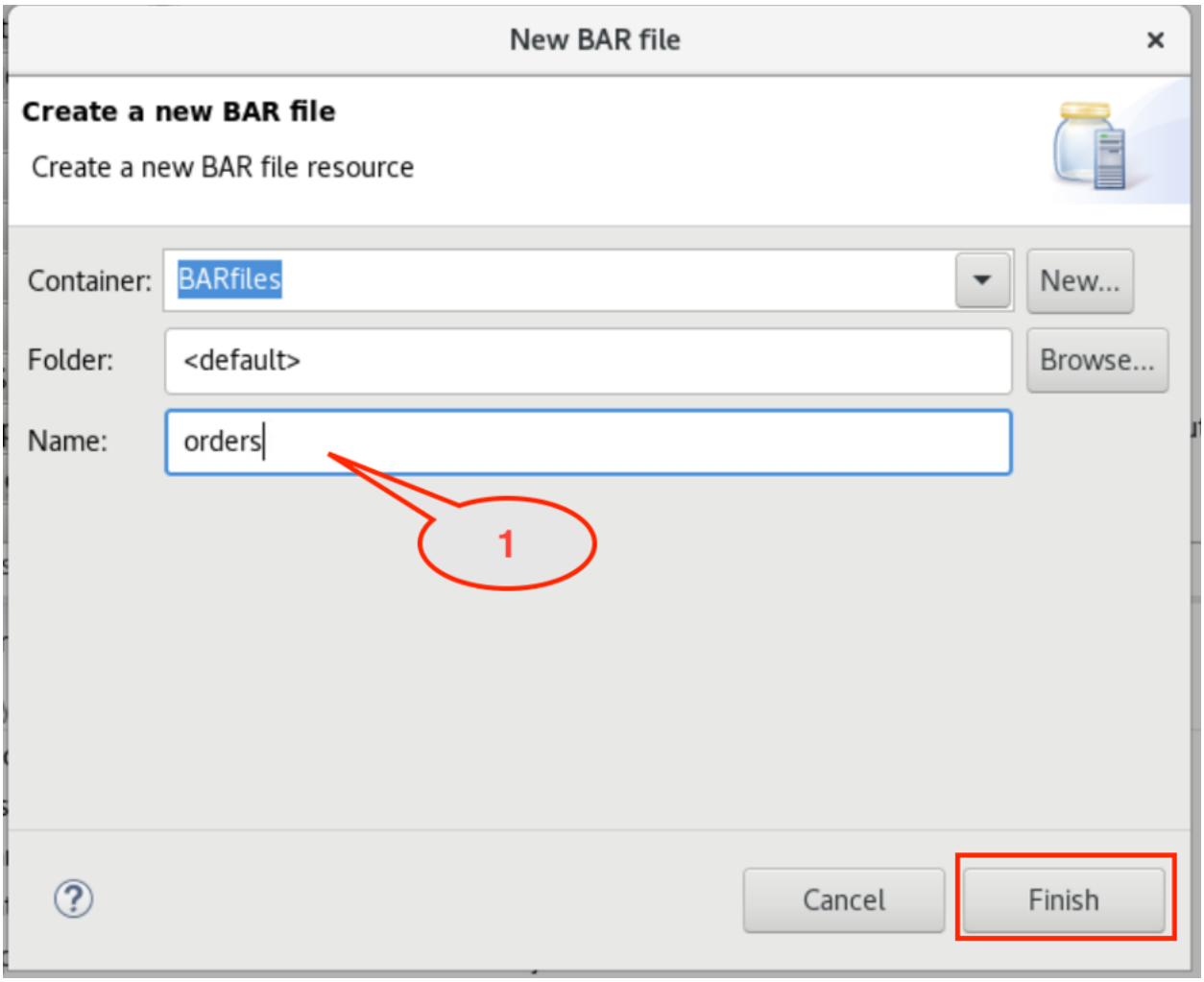
1. Select the **MQ** connection tab. App Connect Enterprise uses a mq client connection. Select **MQ client** connection properties
2. Type Destination queue manager name: **mq**
3. Type Queue manager host name: **mq-1-ibm-mq.mq.svc**
4. Type port number: 1414
5. Type Channel name: **SYSTEM.DEF.SVRCONN**
6. If necessary **Save the flow.**



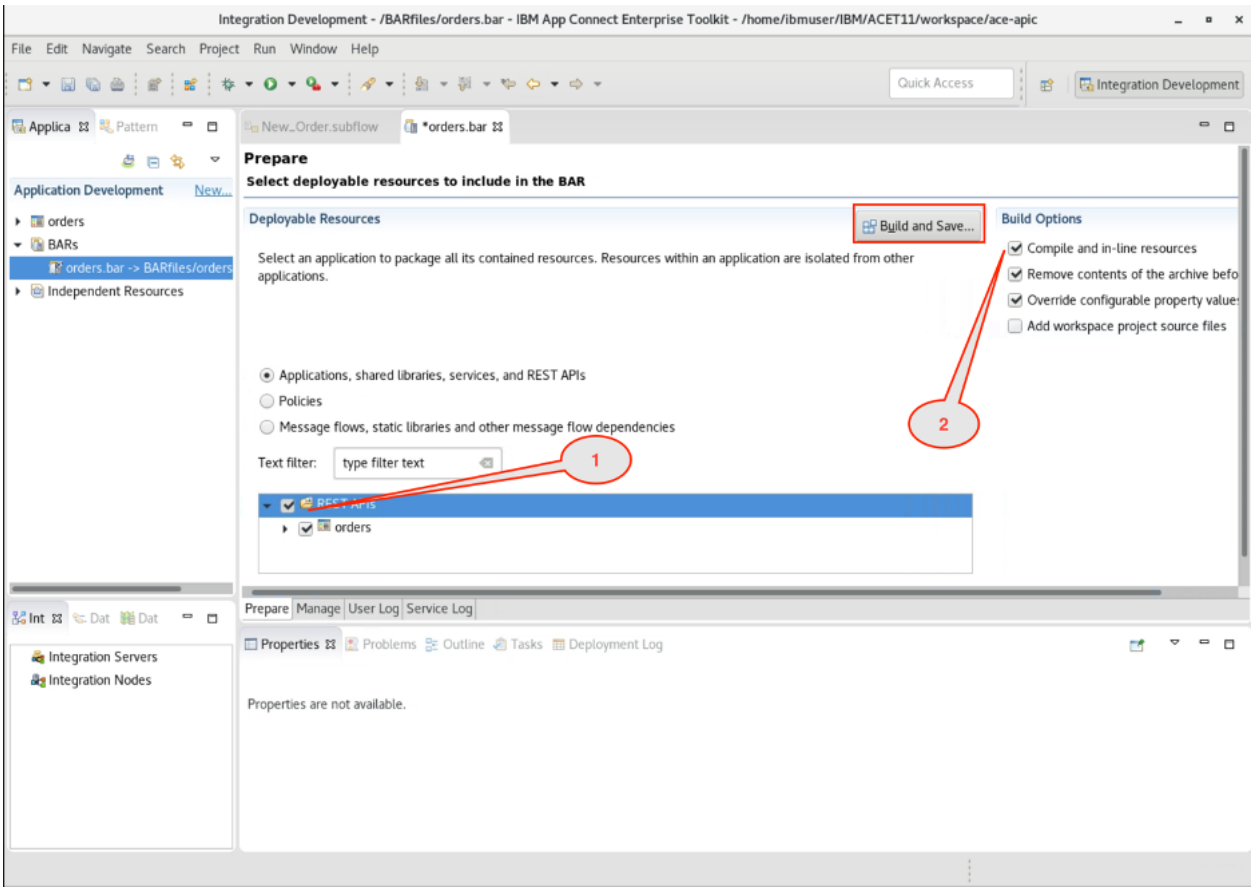
7. In the Application Development view on the left, on orders flow, right-click and then select **New** -> **BAR file**.



8. Type the name of BAR file: **orders** and click **Finish**. App Connect Enterprise is creating an empty BAR file



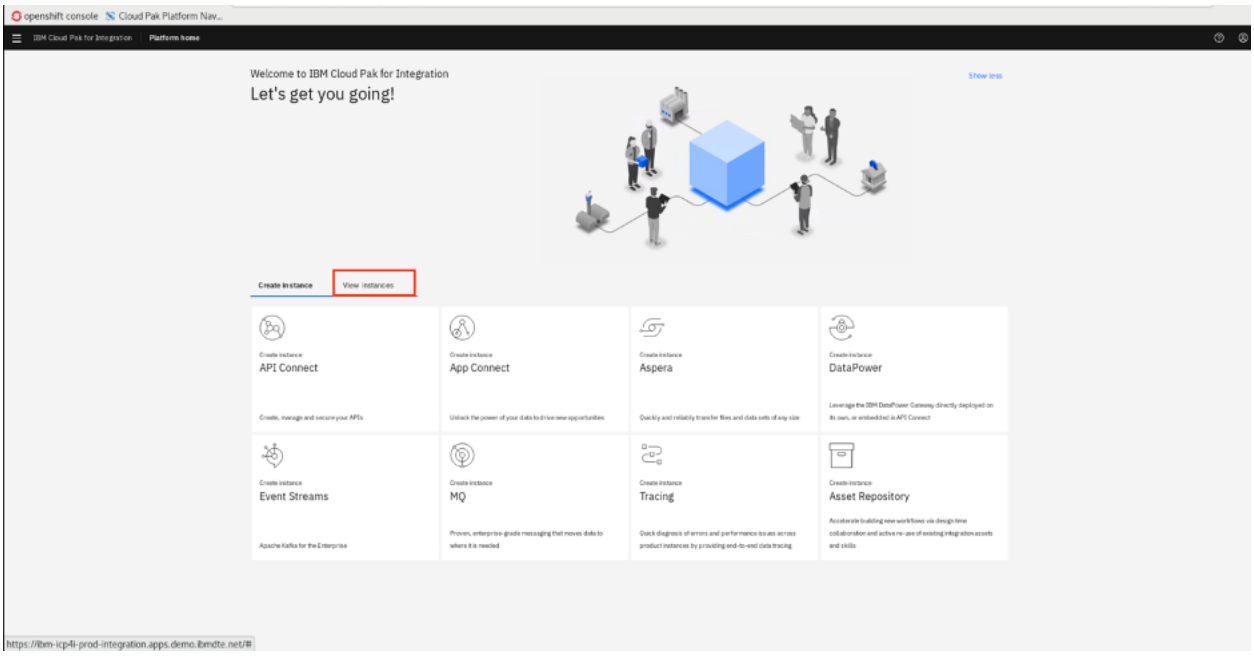
9. You need to configure which artifacts are compiled in the BAR file. Check **orders** and check **Compile and in-line resources**, then click the **Build and Save** button. A pop-up window displays “Operation completed successfully.” Click **OK.**



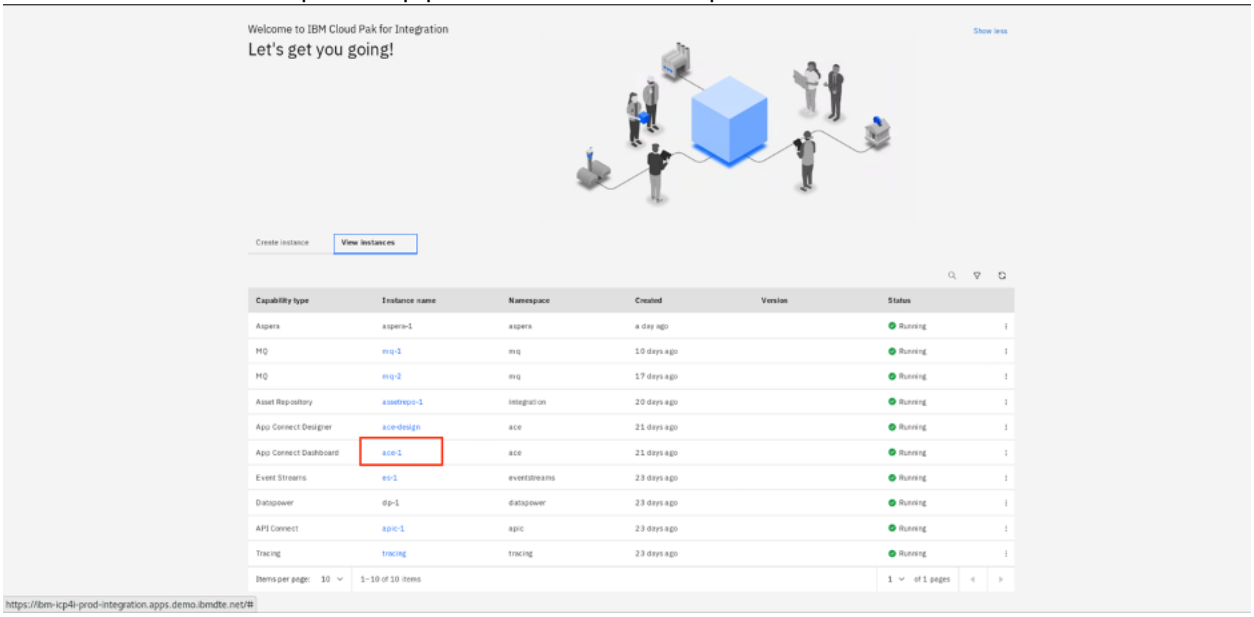
Task 5 - Deploy Integration BAR file as containers.

In this task, you deploy a BAR file in App Connect Enterprise Dashboard.

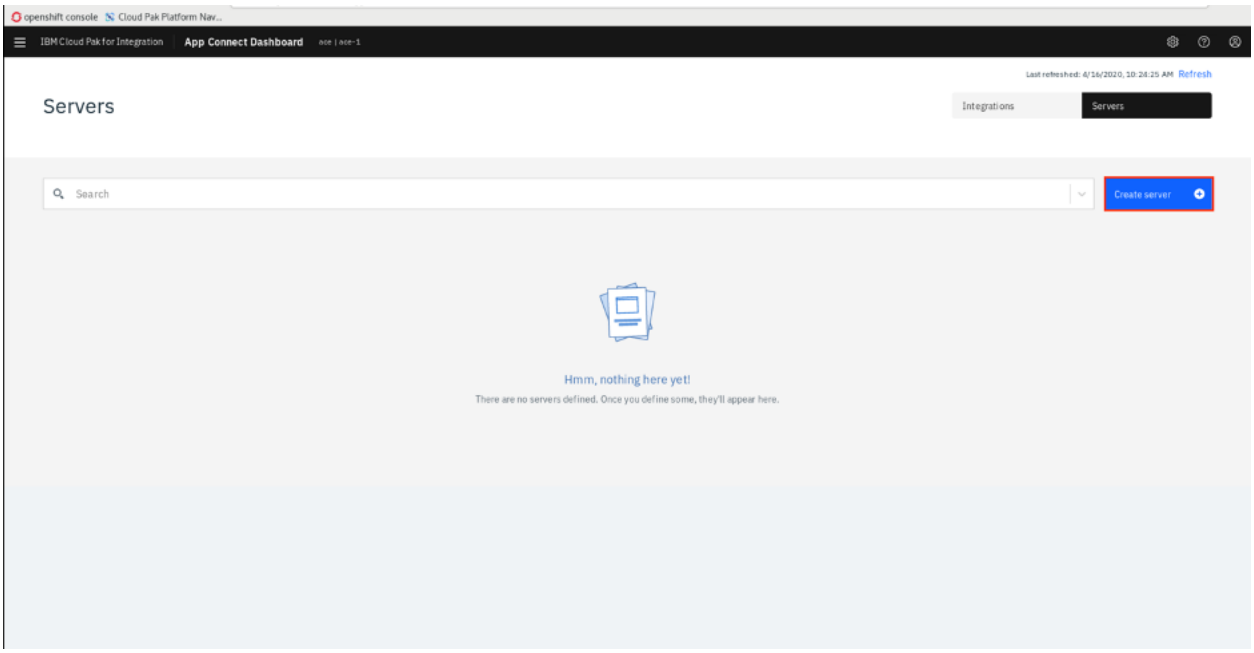
1. Click the bookmark bar **Cloud Pak Platform Navigator** in the browser. Click **Skip Welcome** , Open **View instances**



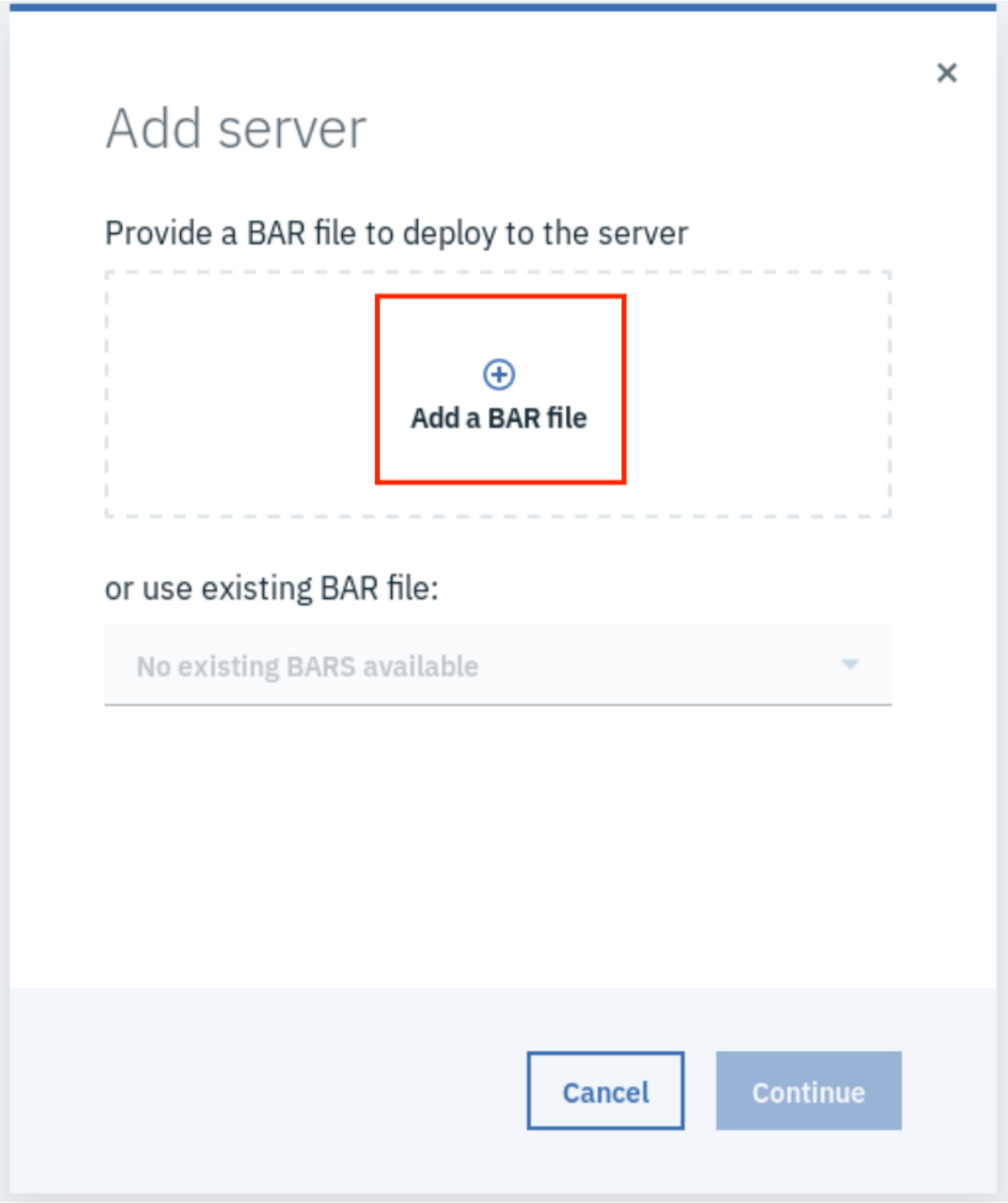
2. Click ace-1 link to open App Connect Enterprise Dashboard.



3. In the App Connect Enterprise Dashboard, you see the Integration Server deployed. To deploy the orders.bar file you saved and complied above click **Create server**.

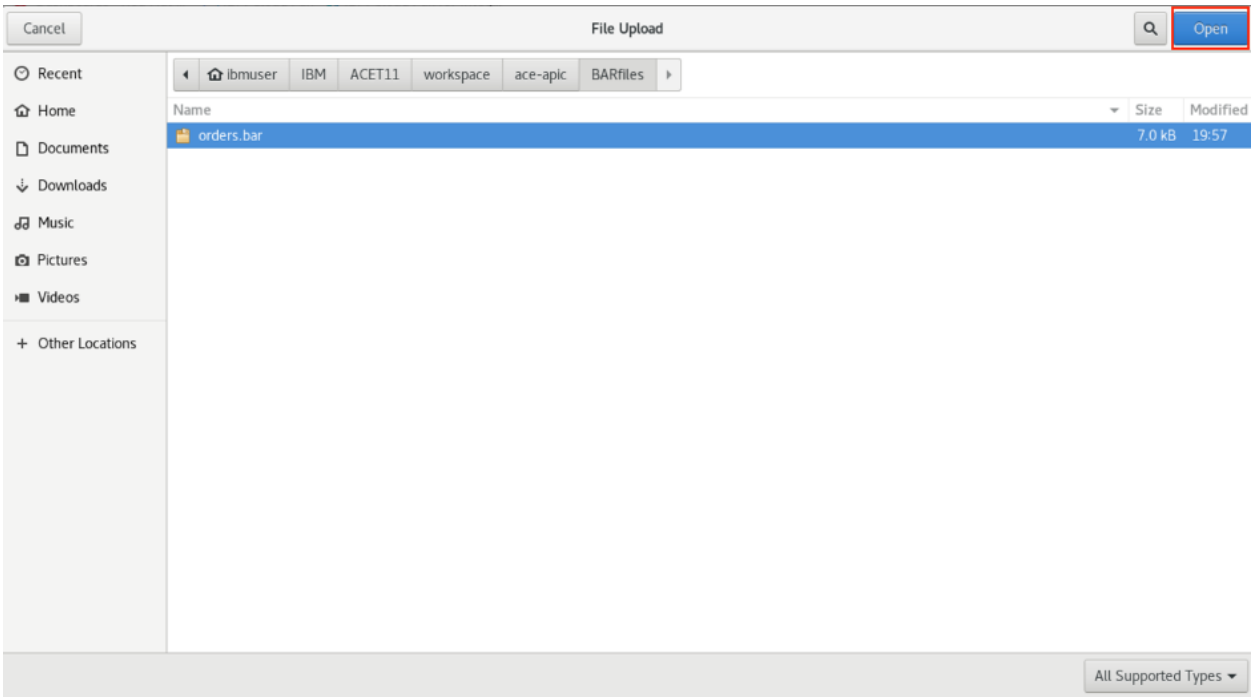


4. Click the box **Add a BAR file**.

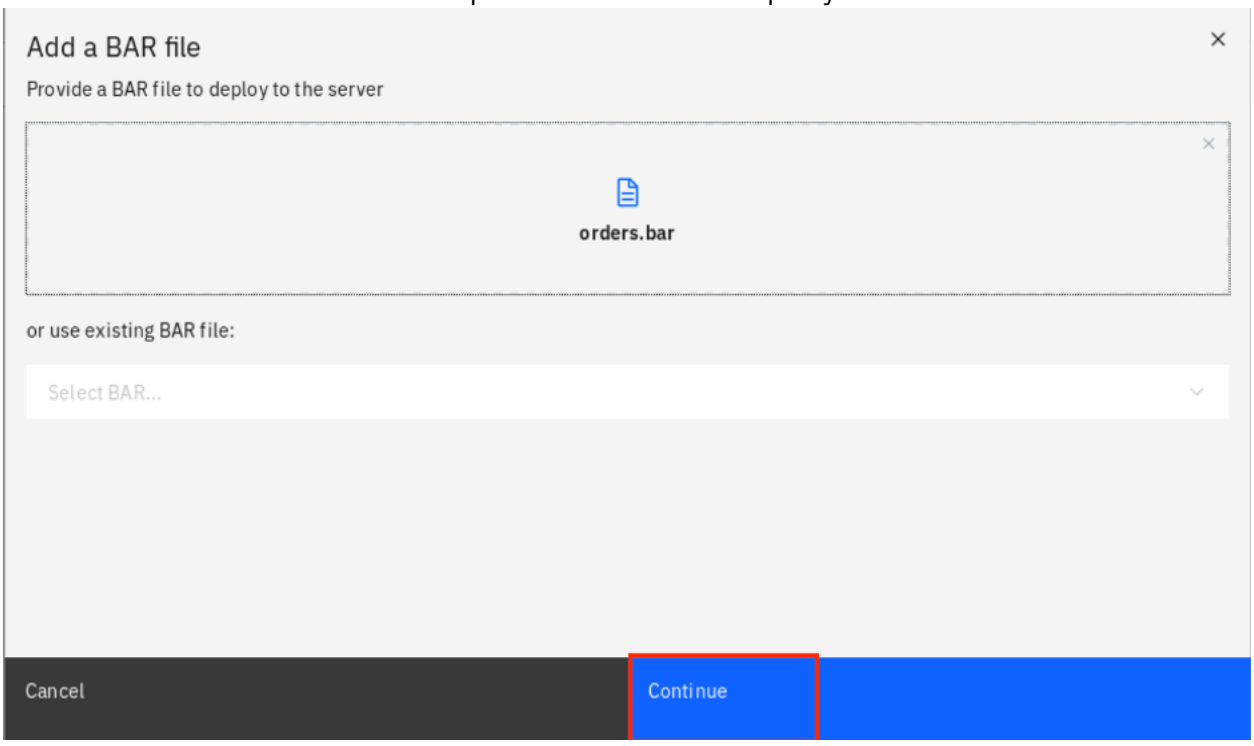


5. In the File Upload window. Open /home/ibmuser/IBM/ACET11/workspace/ace-apic/BARfiles andselect

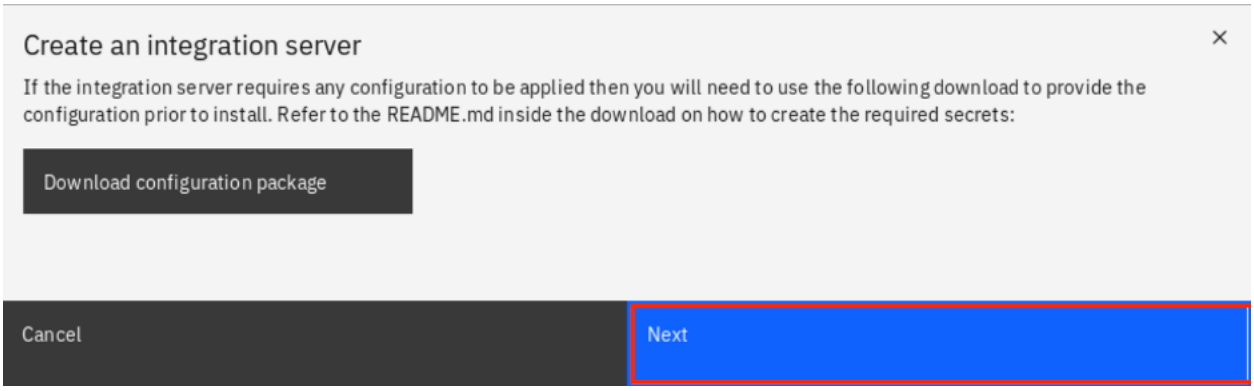
the file **orders.bar** and click **Open**. Verify the path routes to the **ace-apic** folder.



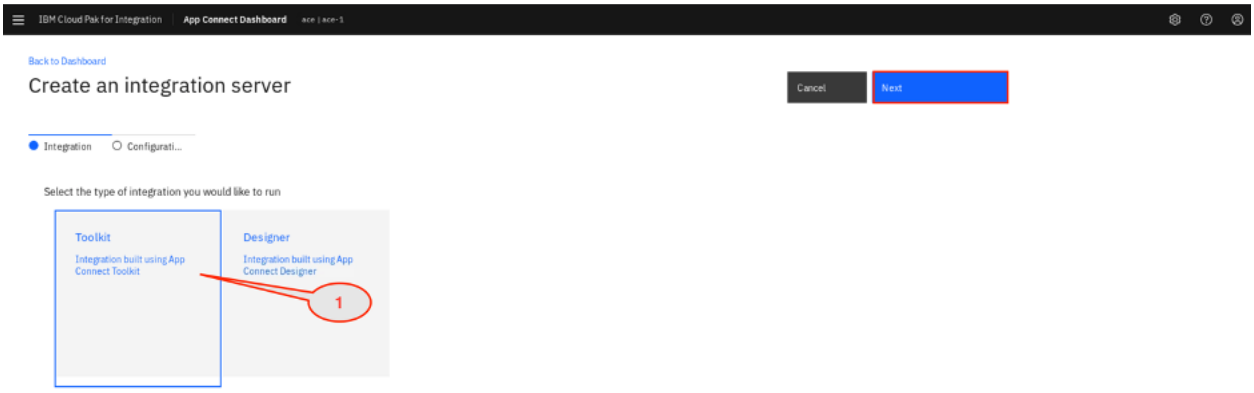
6. You see that orders.**bar** has uploaded to be deployed. Click **Continue**.



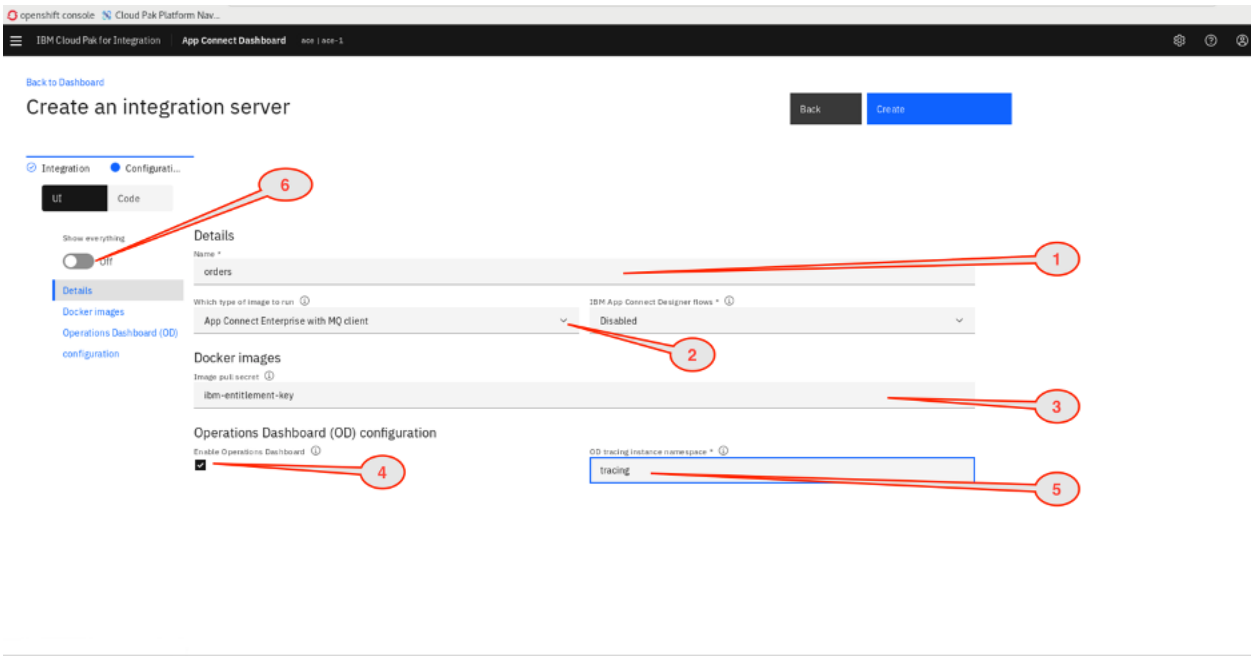
7. You don’t need to download configuration package (Configuration package contains the files that you can use for App Connect Enterprise works with Databases, Event Streams, etc) click **Next .**



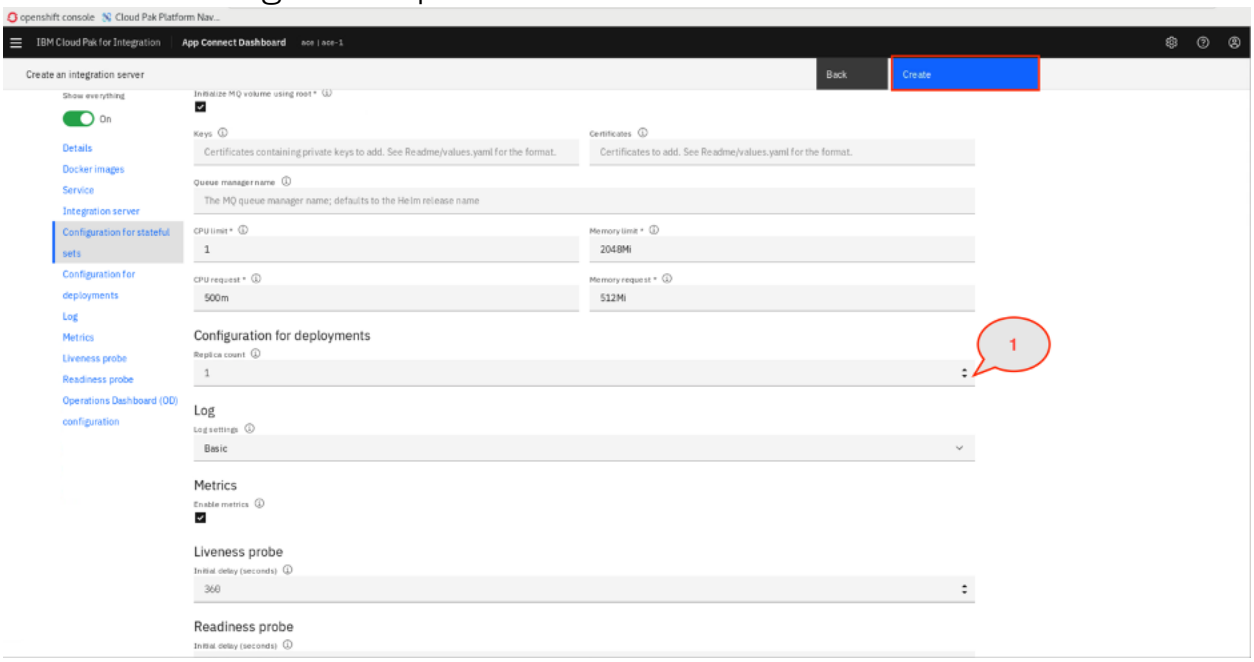
8. In the **Create an integration server** page. You have two option to deploy a BAR file. Deploy a BAR file from **App Connect Toolkit** or a BAR file from **App Connect Designer**. In this lab you deploy BAR file from App Connect Toolkit. Select **Toolkit** link and then click **NEXT .**



9. Enter the parameters:
1. Integration Server name: **orders**
2. Which type of image to run: **App Connect Enterprise with MQ Client**
3. Enter as Image pull secret: **ibm-entitlement-key**
4. **Check Enable Operations Dashboard**
5. Enter OO tracing instance namespace: **tracing**
6. Click **Show Everything to On .**

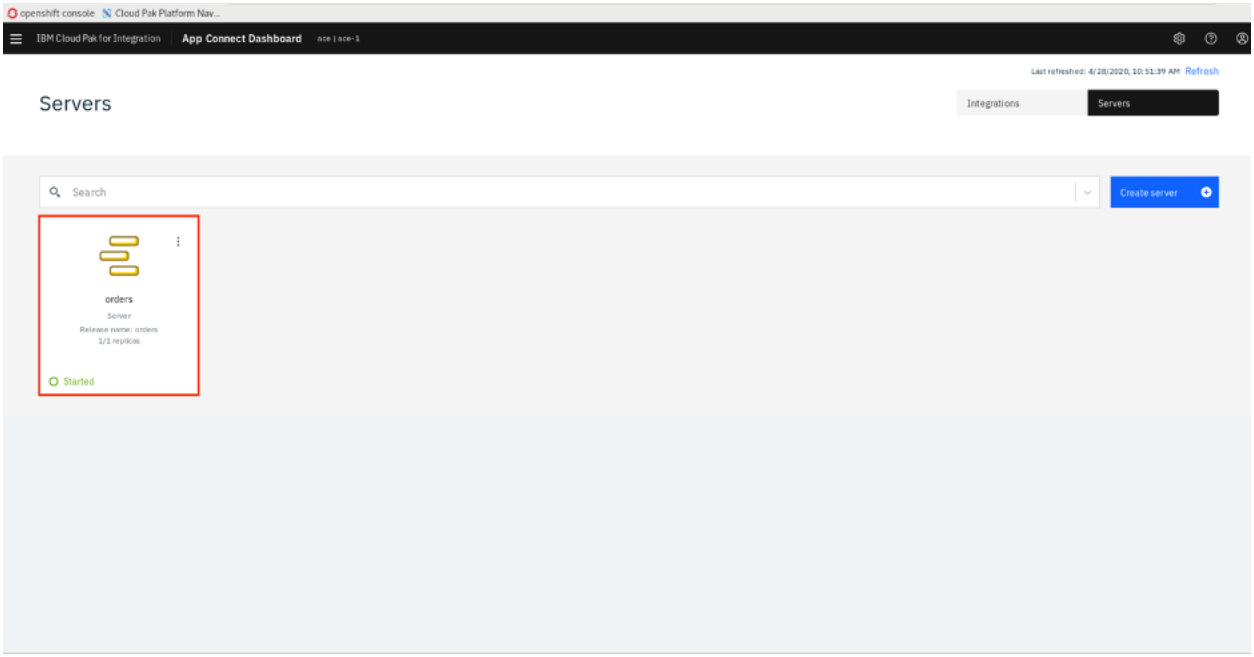


10. Scroll down and locate **Configuration for deployments** and click the **arrow** to change the replica count to **1** and then click **Create.**

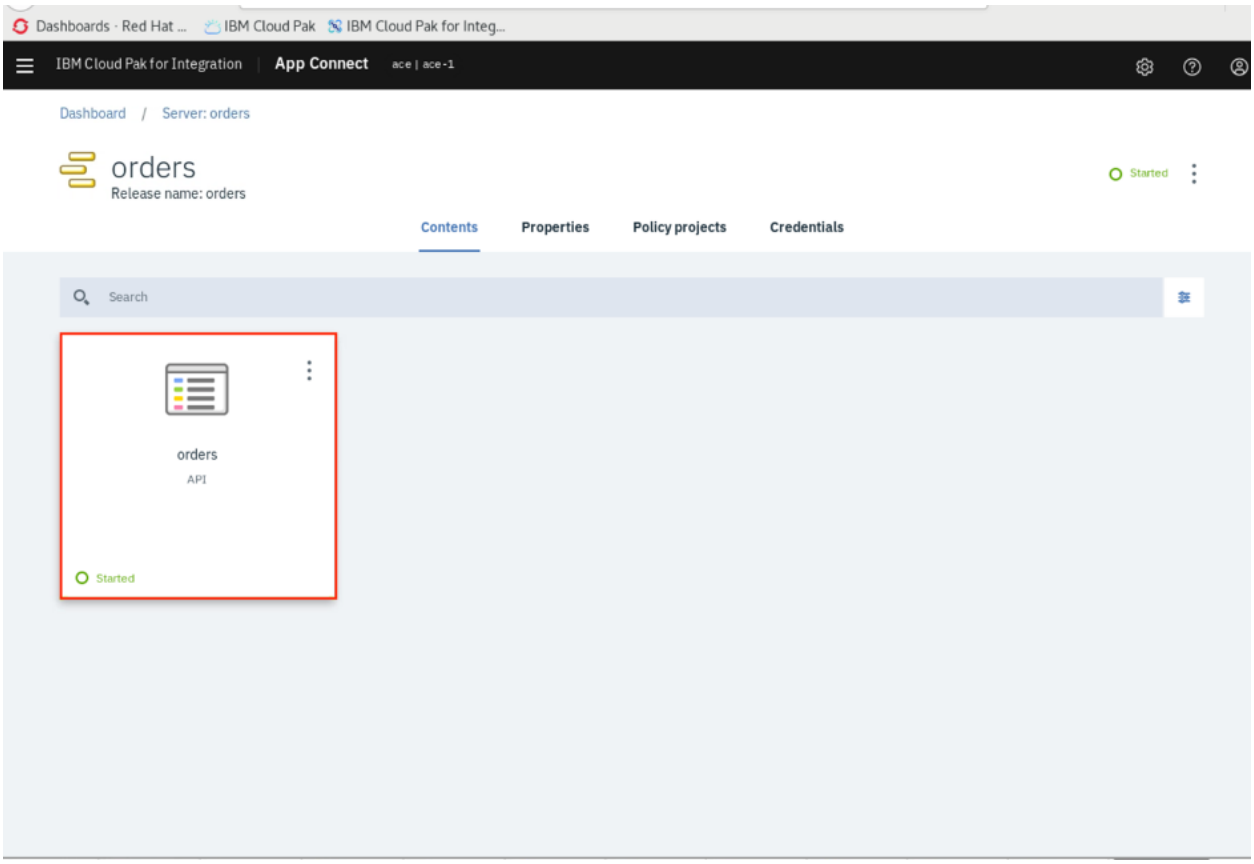


11. During the deployment process, App Connect Enterprise opens the servers page. You see the App Connect Enterprise Dashboard with the Integration Server **orders** deployed and started .Click the **orders** server icon.

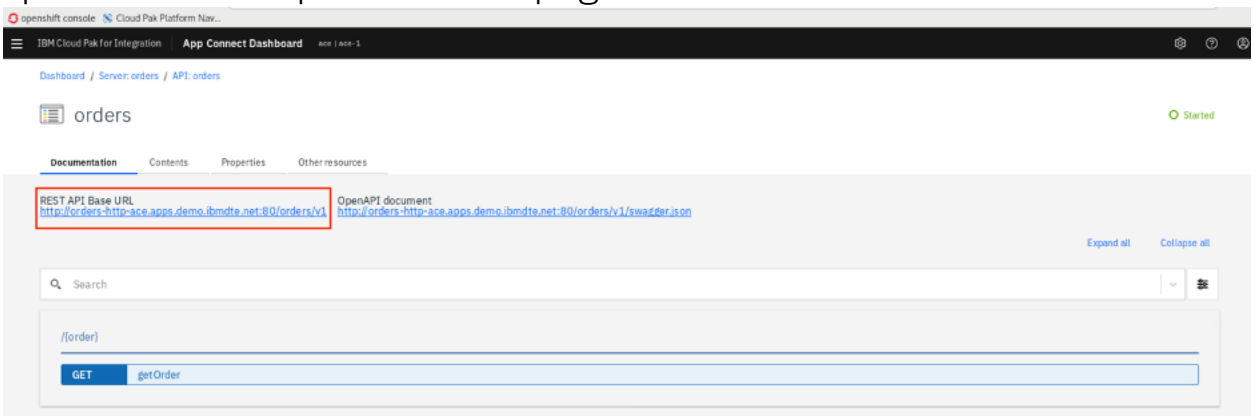
Note: The deployment process takes 2-3 minutes, refresh the browser to see the BAR file .



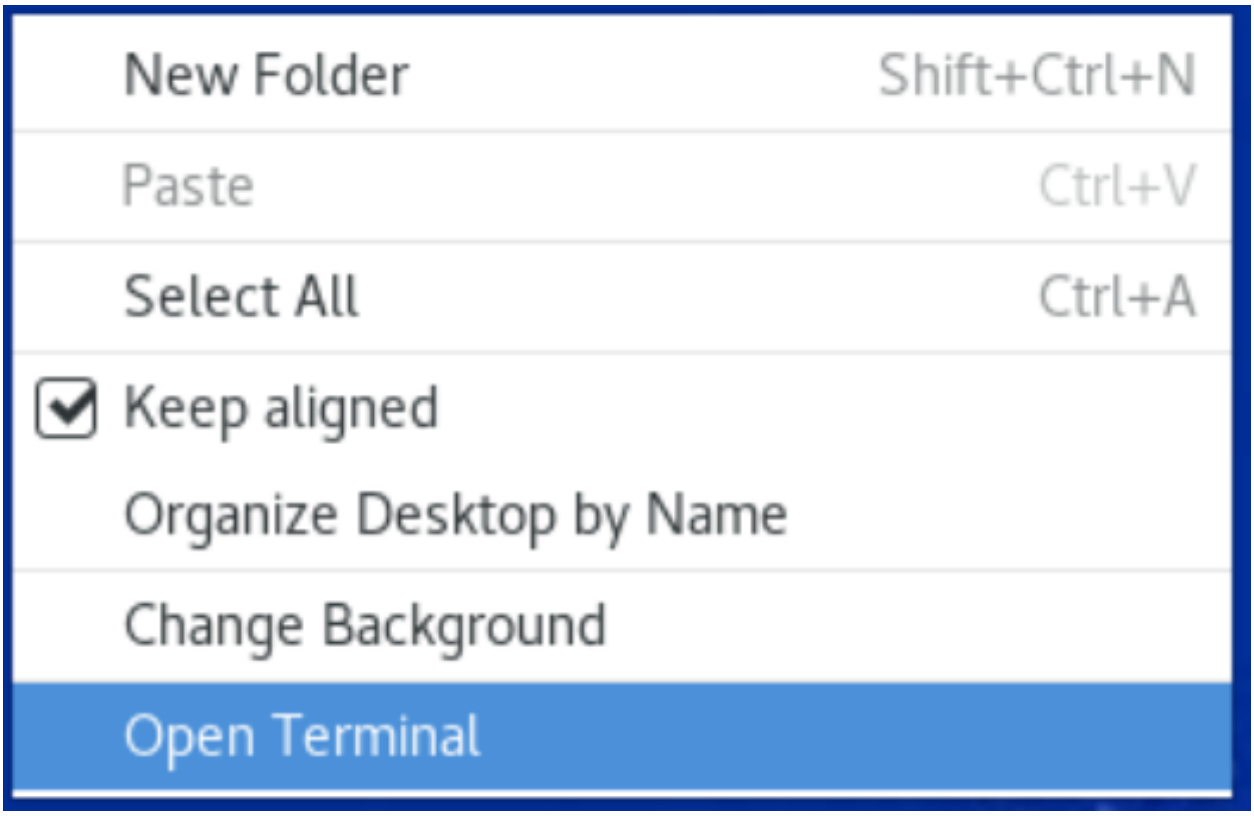
12. Click the **orders** server. Click **orders** API.



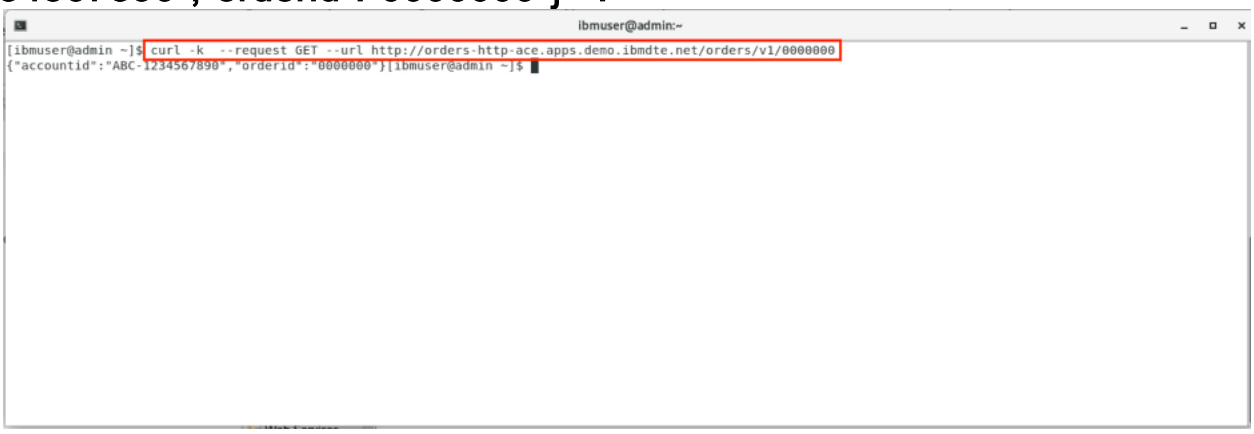
13. This page displays the REST API Base URL. you use the base URL (in the example below: http://orders-http-ace.apps.demo.ibmdte.net:80/orders/v1). Keep the browser opened in this page.



14. Open a Terminal Window, right mouse on desktop workspace and select **Open Terminal.**

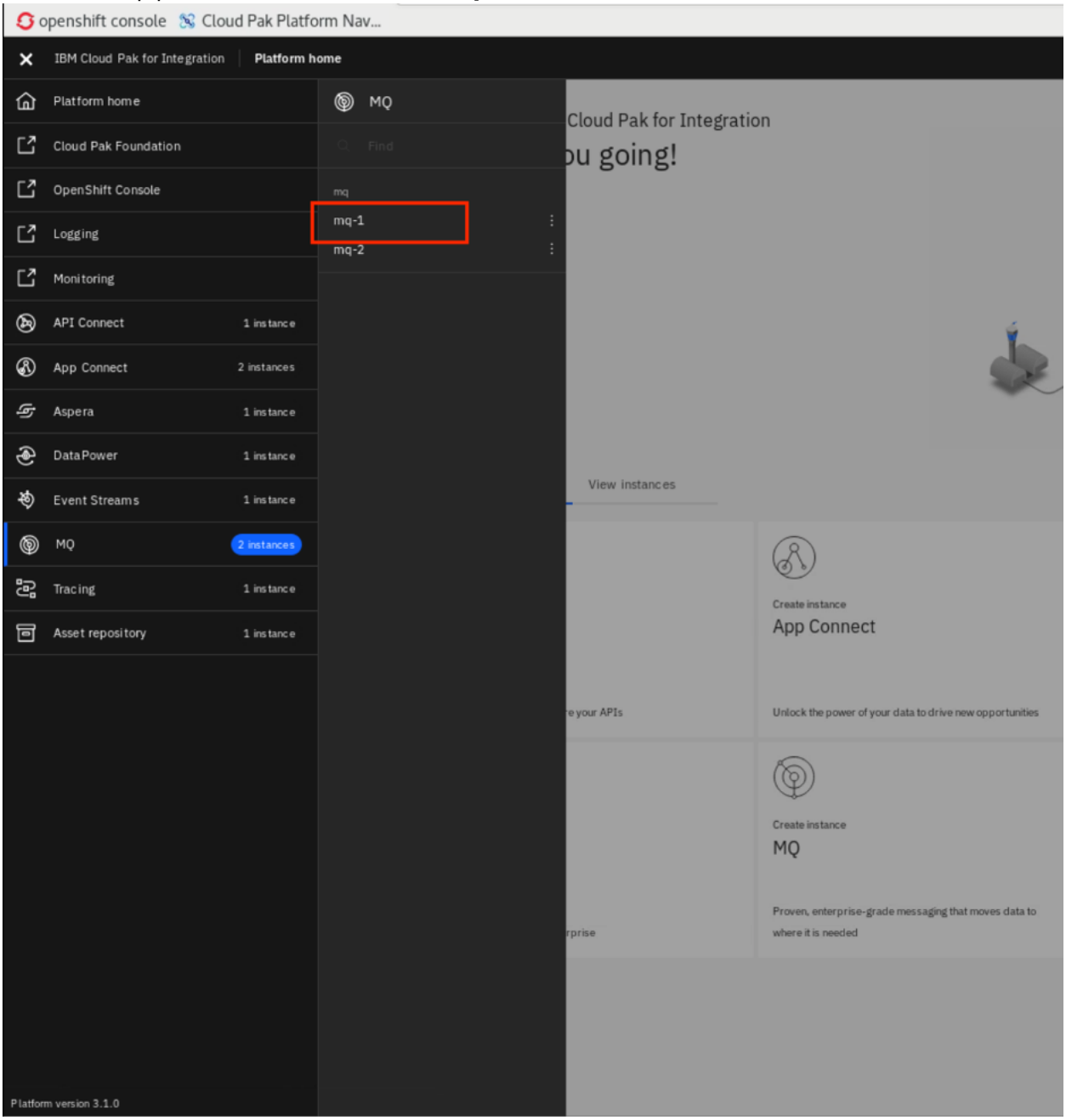


15. From the terminal window, execute the following curl command :
curl -k --request GET – url http://orders-http-ace.apps.demo.ibmdte.net/orders/v1/0000000
If the API call is successful, you see JSON reply with **{"accountid":"ABC-1234567890","orderid":"0000000"}** .

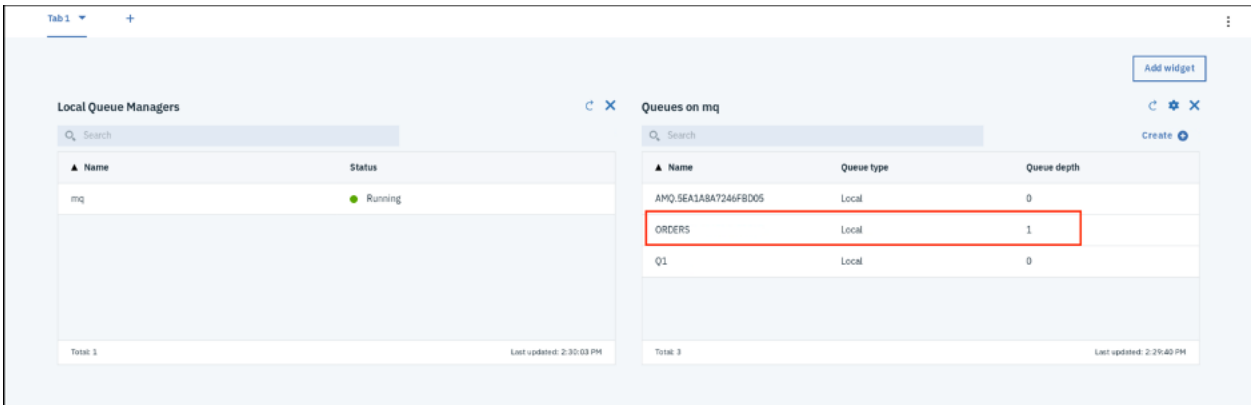


16. You check the message arrived in queue **ORDERS** in MQ (**mq-1**). You can check using MQ Console. Open a browser and click **Cloud Pak**

Platform Navigator bookmark bar and click “**Hambuger**” menu, and then select MQ application and click **mq-1**.



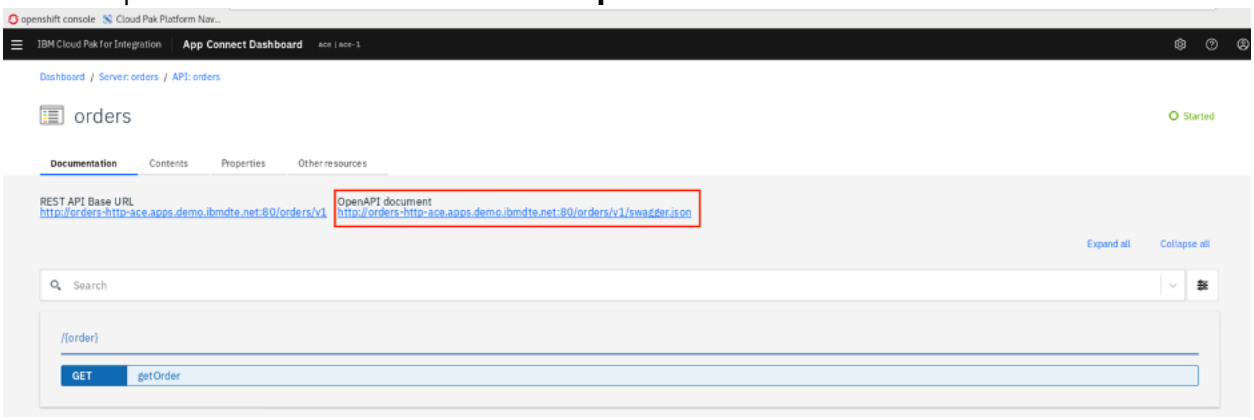
17. You see in **Queues on mq** window. The queue **ORDERS** has a message (look at **Queue Depth**)



Task 6 – Configuring API Connect to test the integration

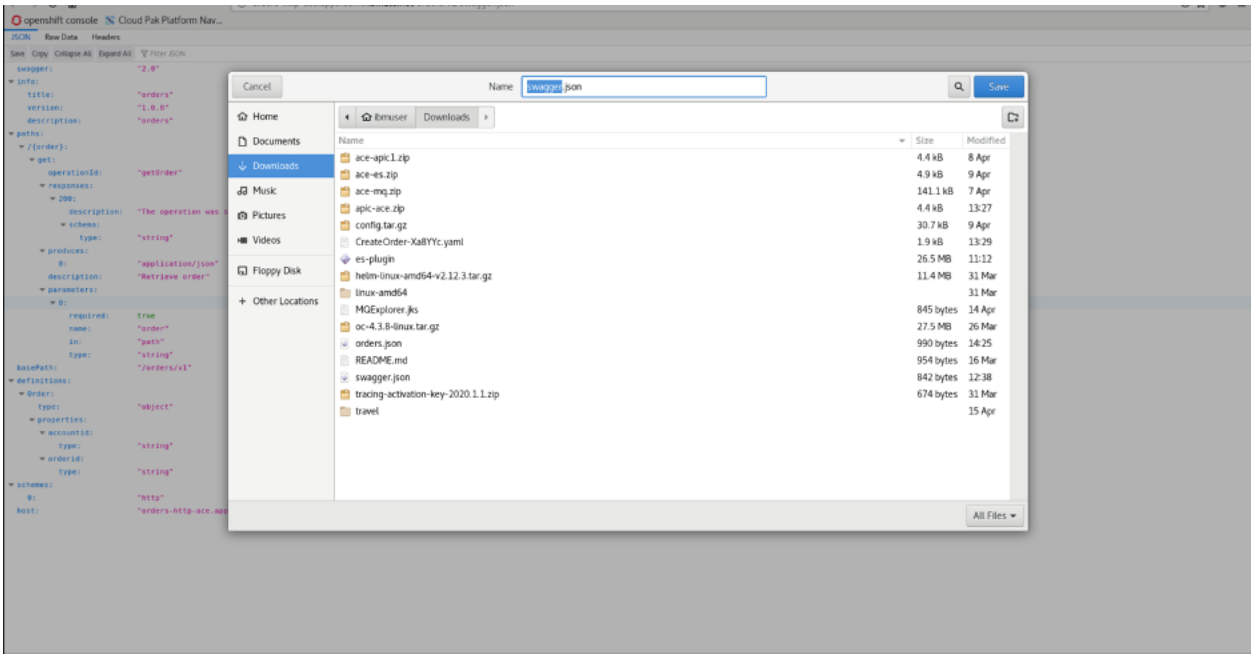
You've created an application integration flow and successfully called it via a REST API call! Now, to make it accessible to the rest of the world, it's important to add security around it—at least in the form of a client ID .This way, in addition to access control, you can get insights into which teams or customers are the least and most active. Adding security to an API is simply done via an OpenAPI configuration parameter. We can add rate limits to the API to increase the calls per second, minute, or hour to scale up as much as you need.

1. Open the browser window opened where you have the orders API window open. Click the URL under **OpenAPI document.**

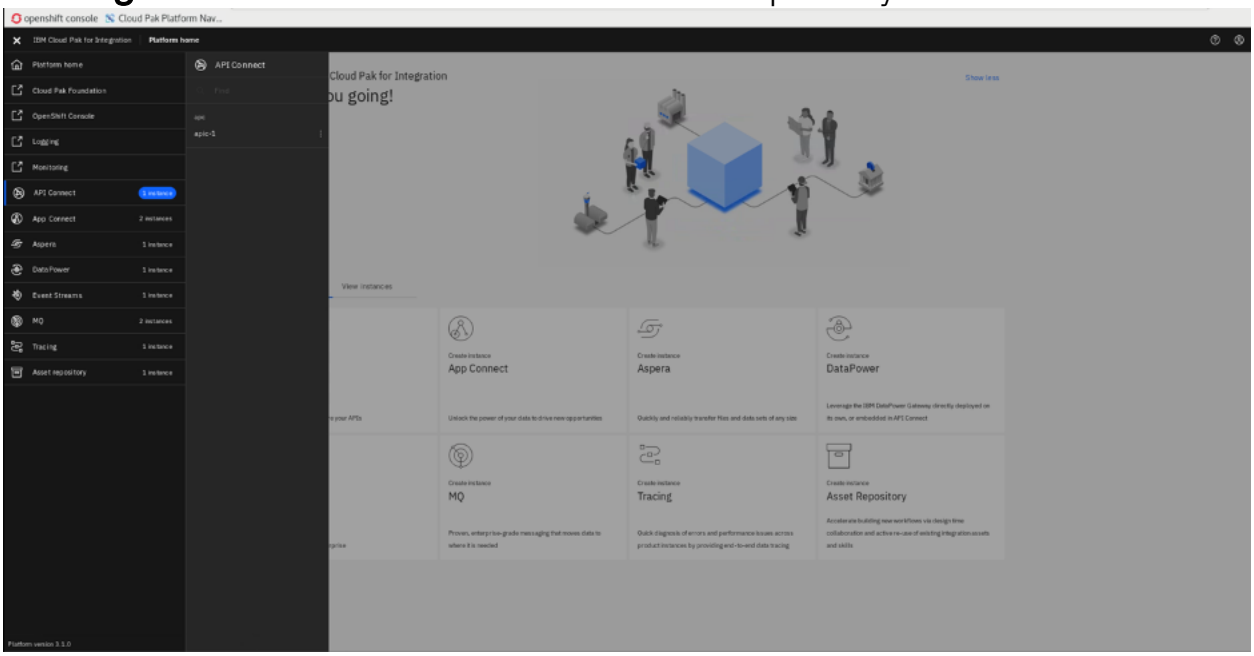


2. A new browser tab opens, containing the swagger file. Right-click anywhere in the body of the text and choose **Save as**. Click **Save** to save the swagger.json file.

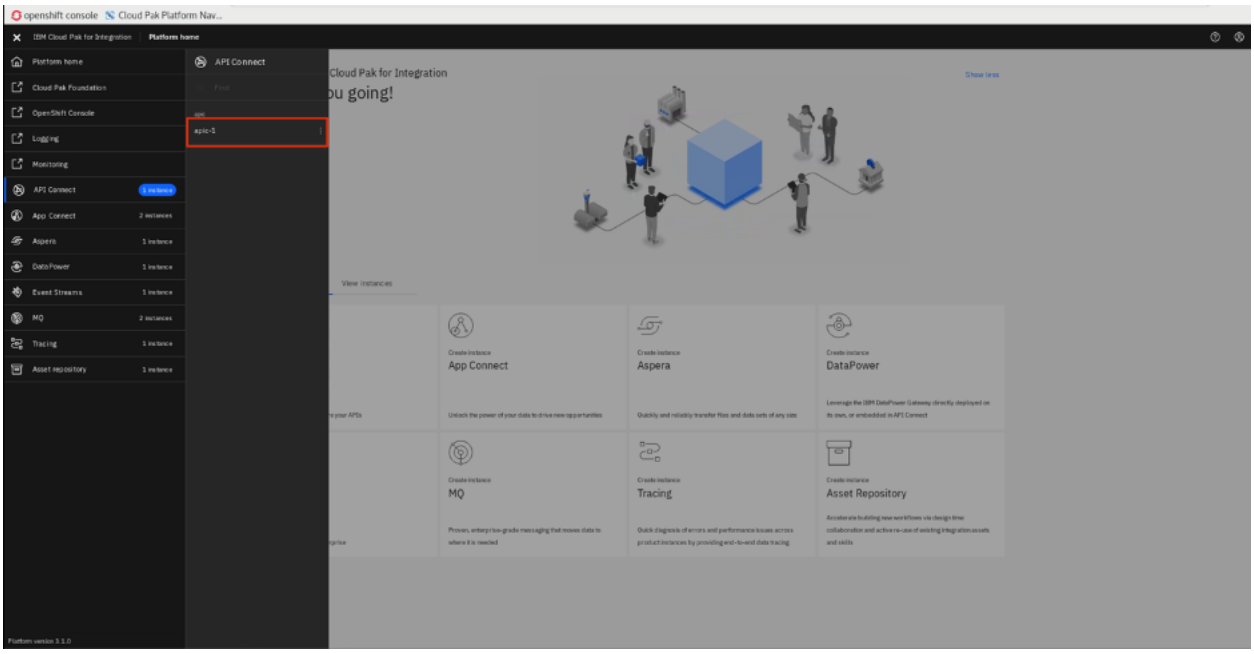
   Note: You might need to confirm to replace.



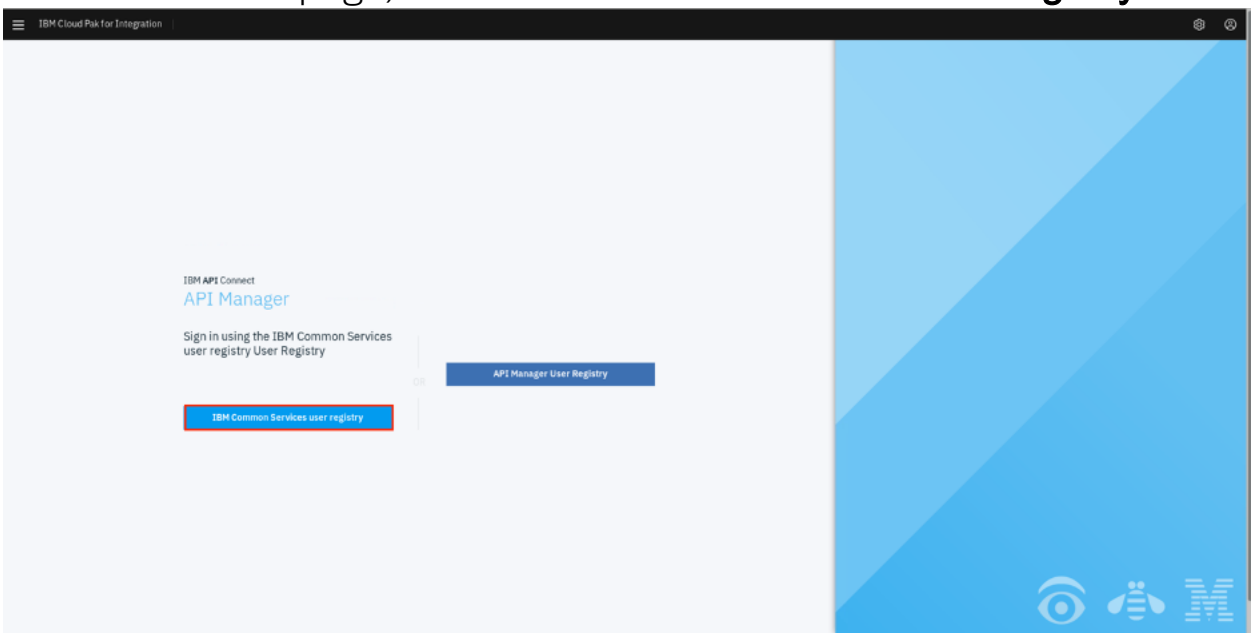
3. Cloud Pak Navigator has a bookmark that you access all services. Click "**Hamburger"** menu and select API Connect capability.



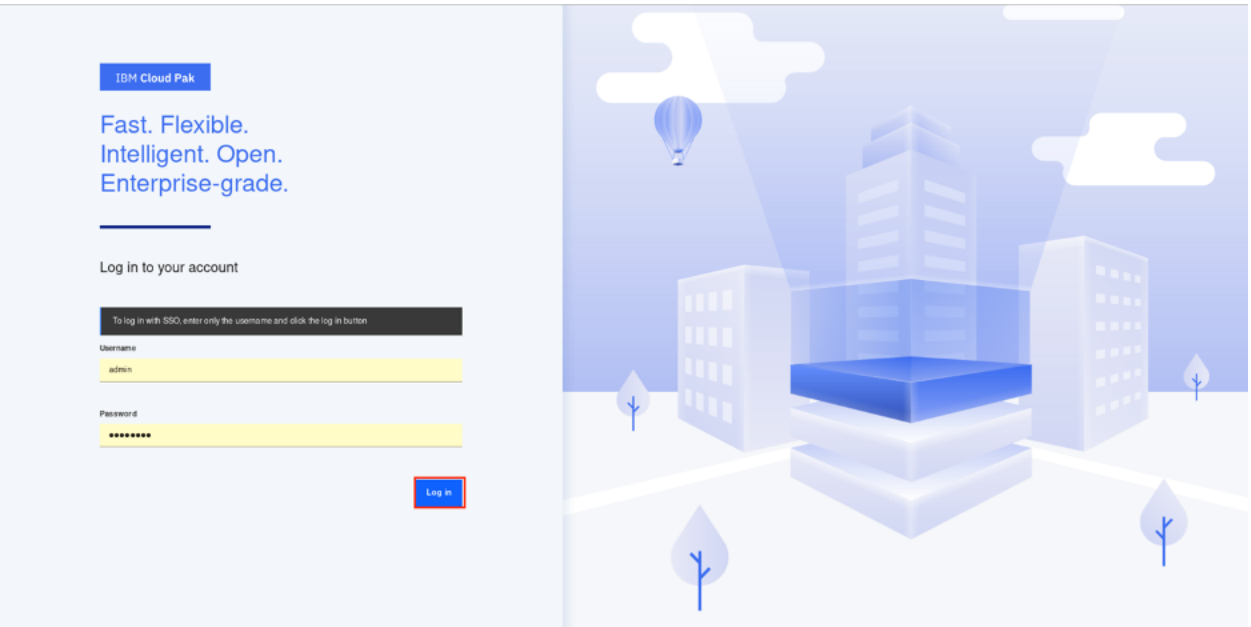
4. Click **apic-1** instance.



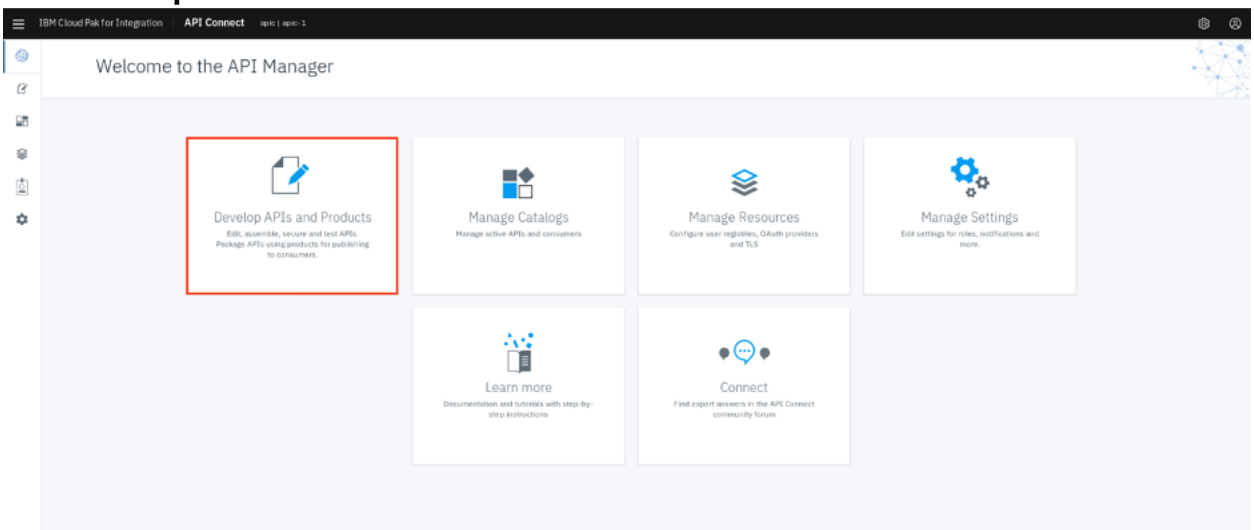
5. In the API Connect page, click **IBM Common Services user registry**.



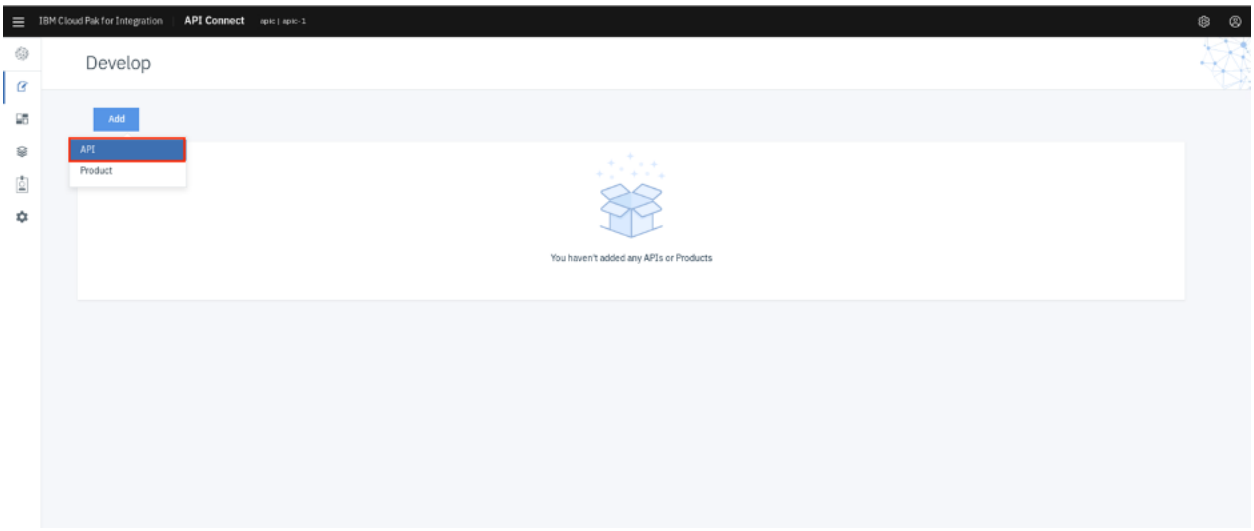
6. You might need to login to IBM Common Services. Juist click **Log in**



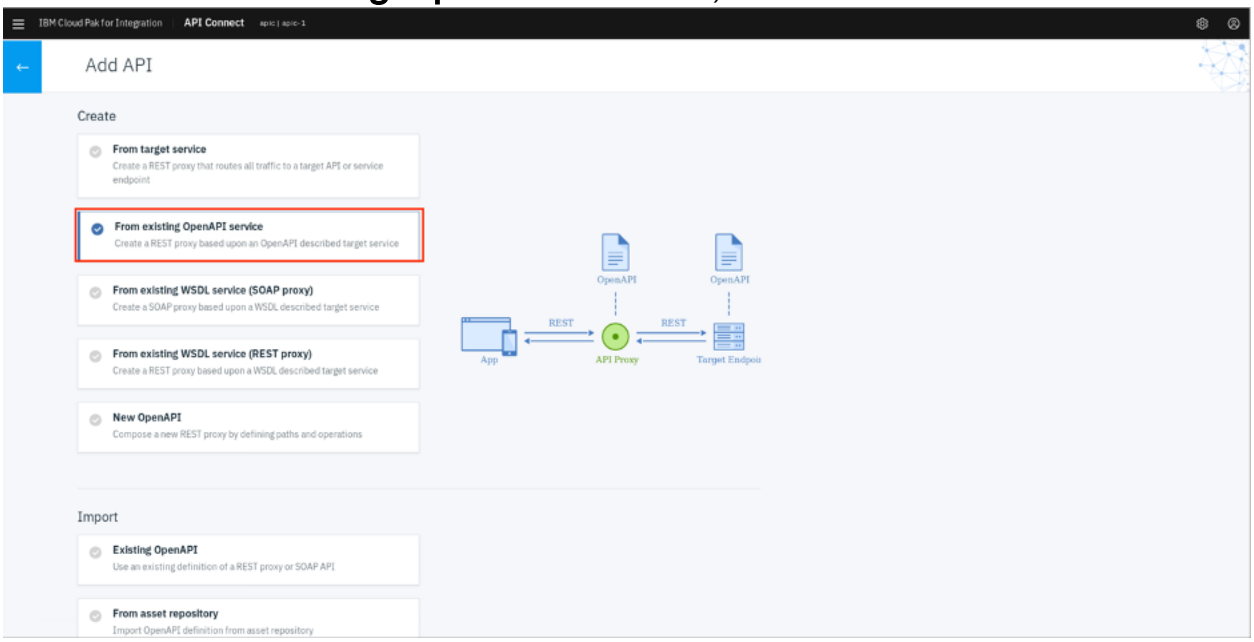
7. Click **Develop APIs and Products .**



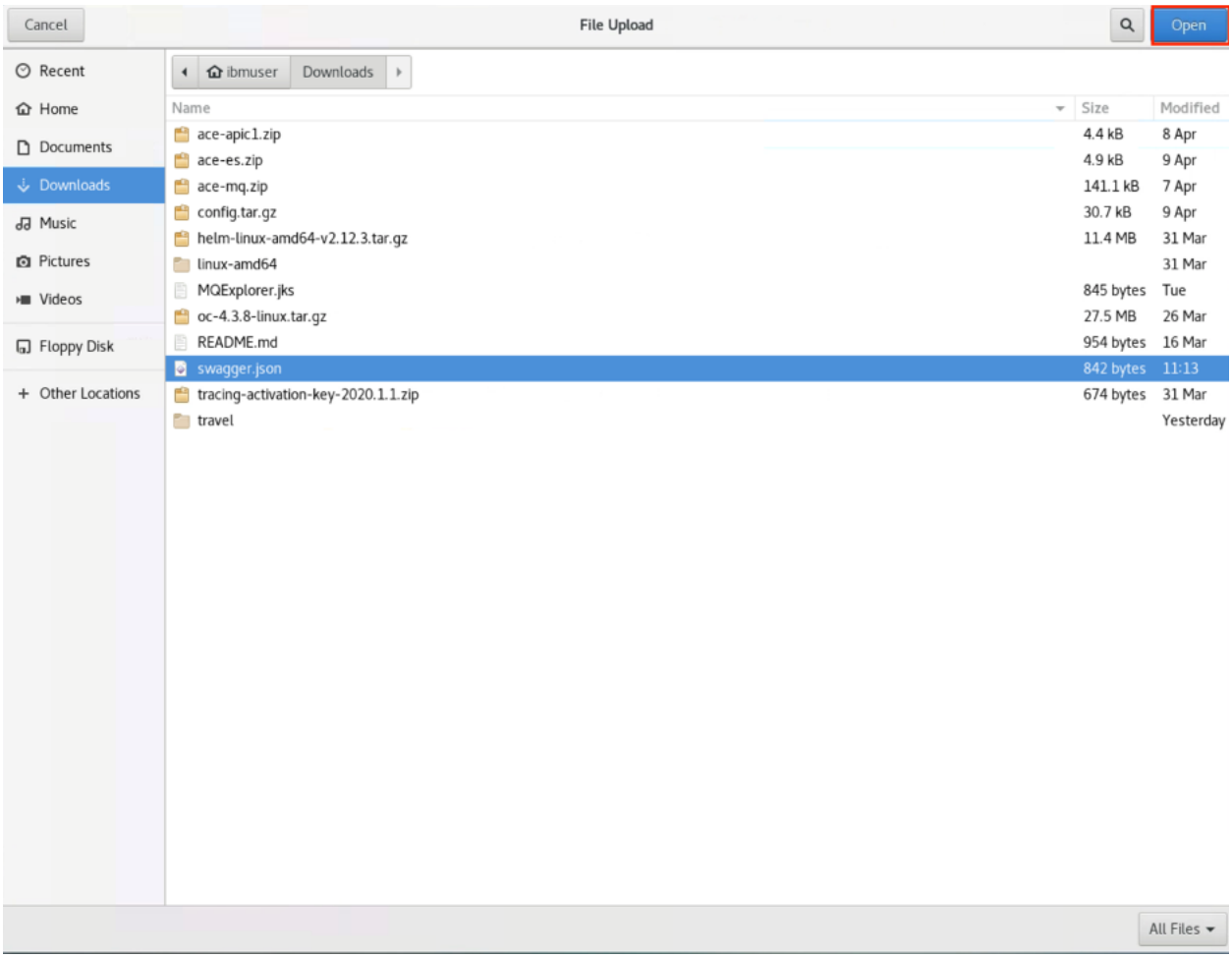
8. Click **Add**, then choose **API** from the drop-down menu.



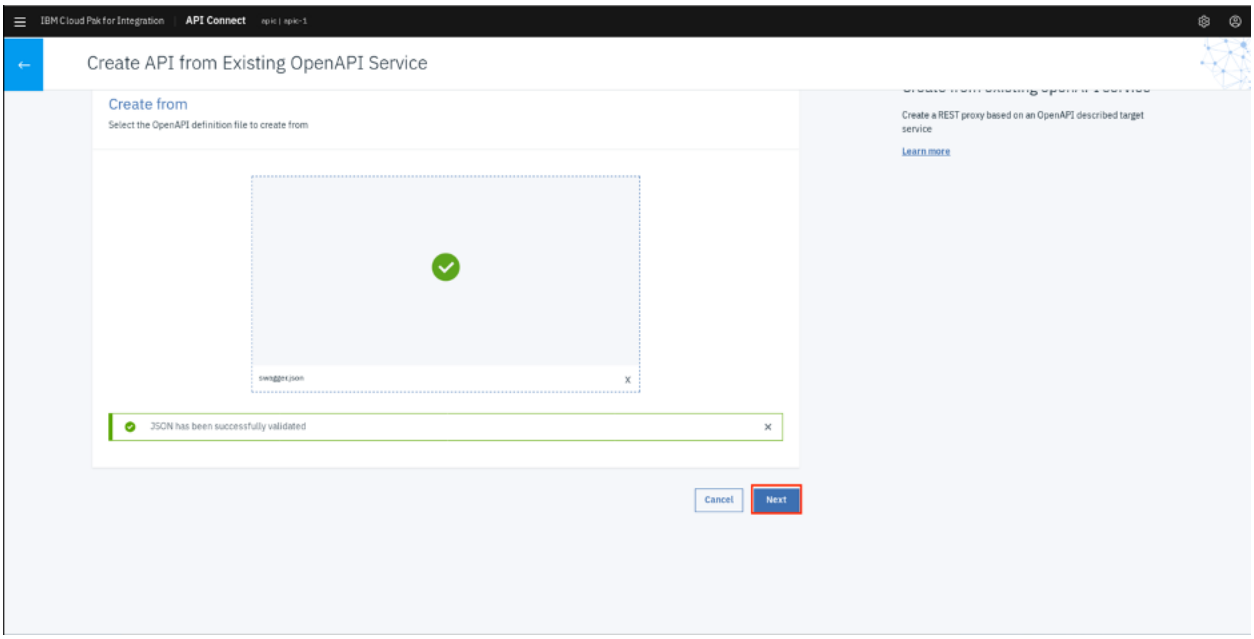
9. Choose **From an existing OpenAPI service** , scroll down and click **Next** .



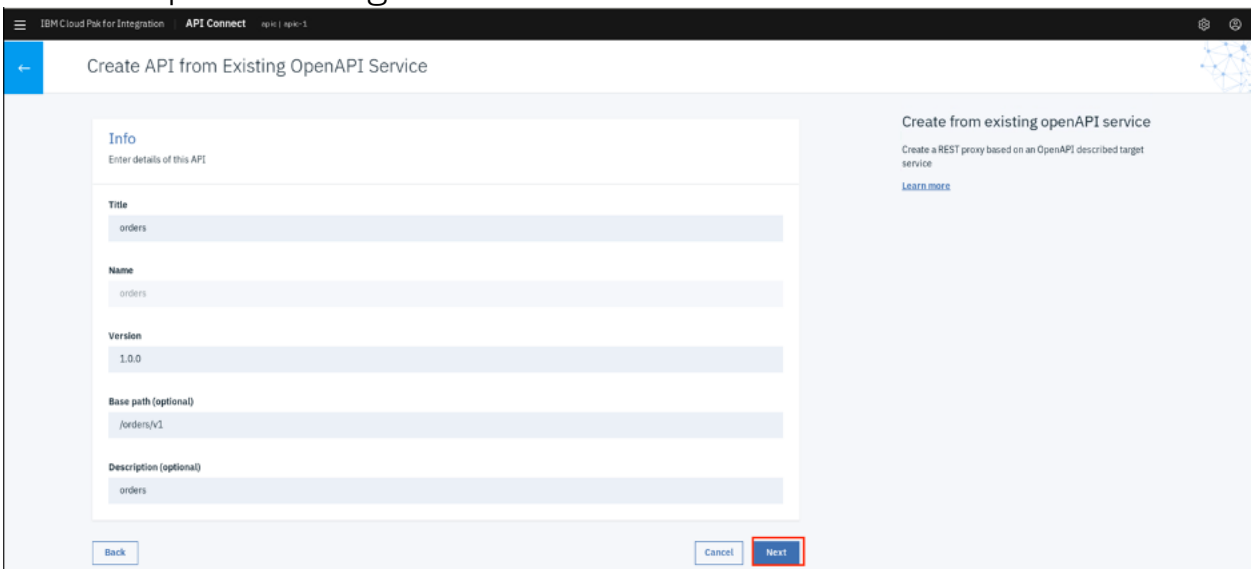
10. Click **Browse**, select the downloaded **swagger.json** file (The file is located under the /home/ibmuser/Downloads/swagger.json directory) and click **Open.**



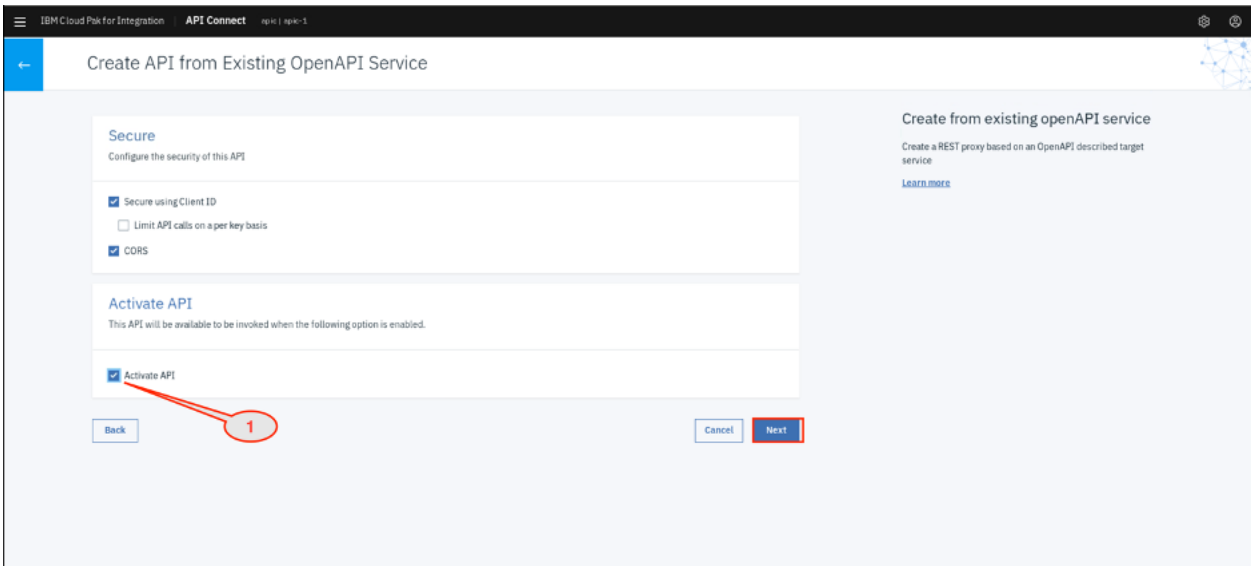
11. Make sure the JSON is successfully validated and then click **Next.**



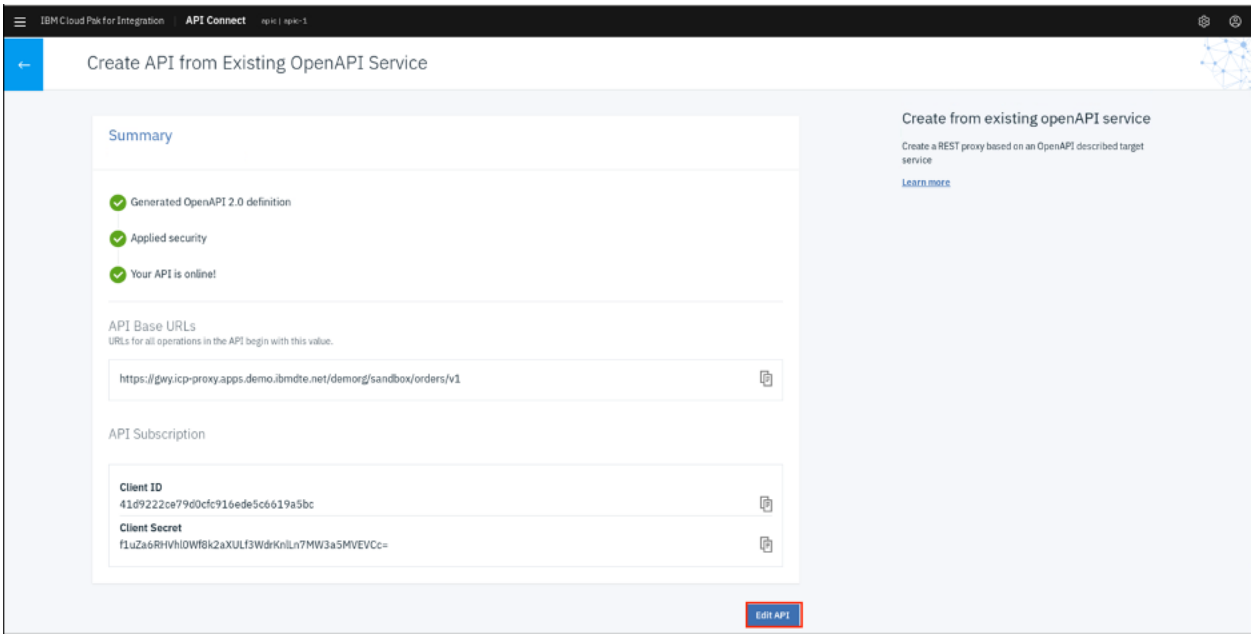
12. Keep the settings and click **Next.**



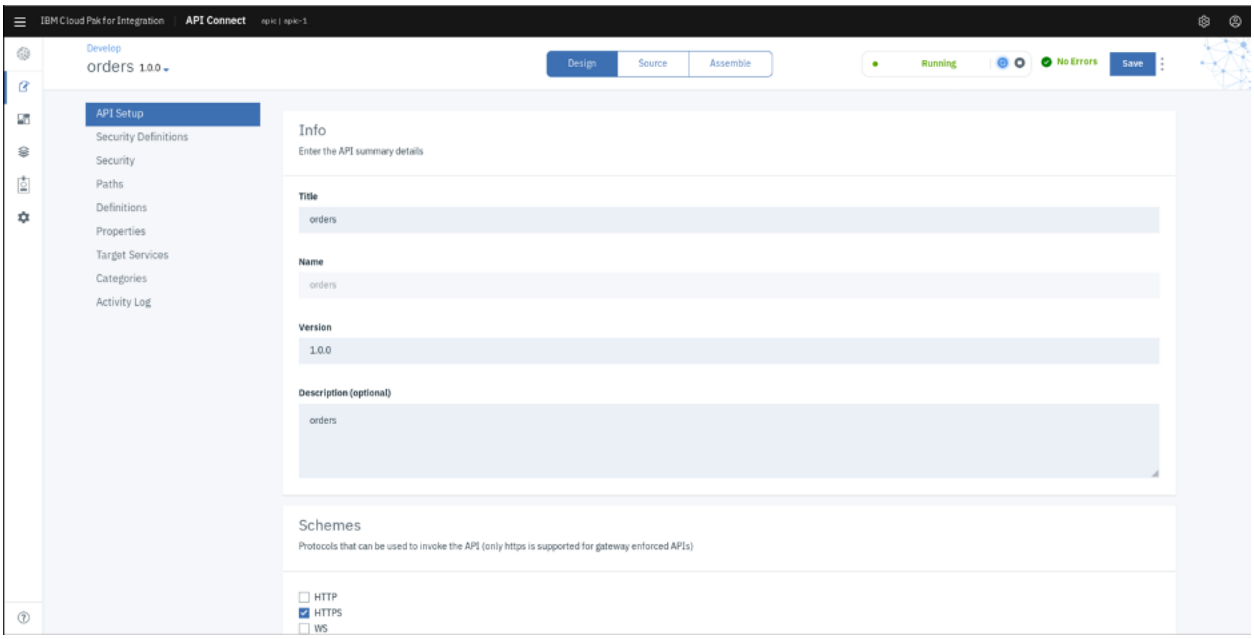
13. Under "**Secure using Client ID**", "**CORS**" and Check "**Activate API**". Click **Next**.



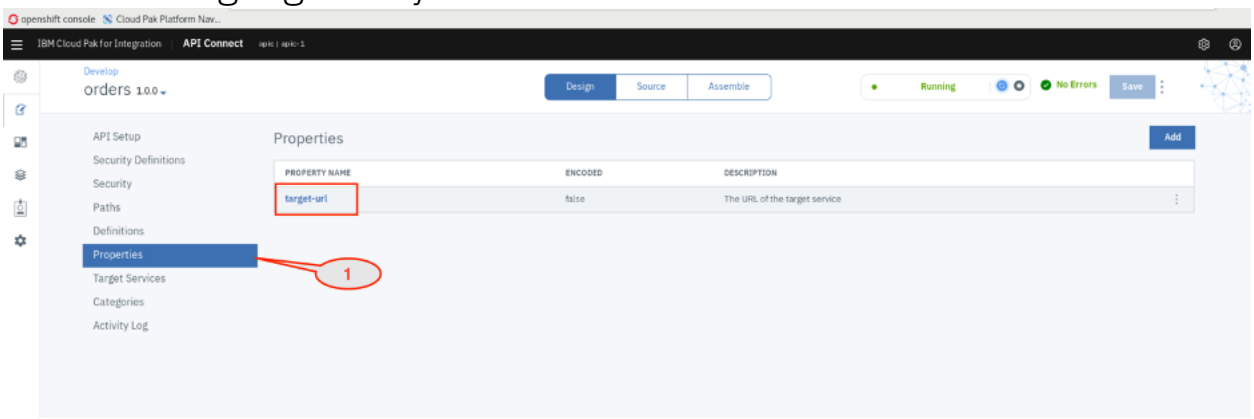
14. Your API with Client ID is created! Click **Edit API**.



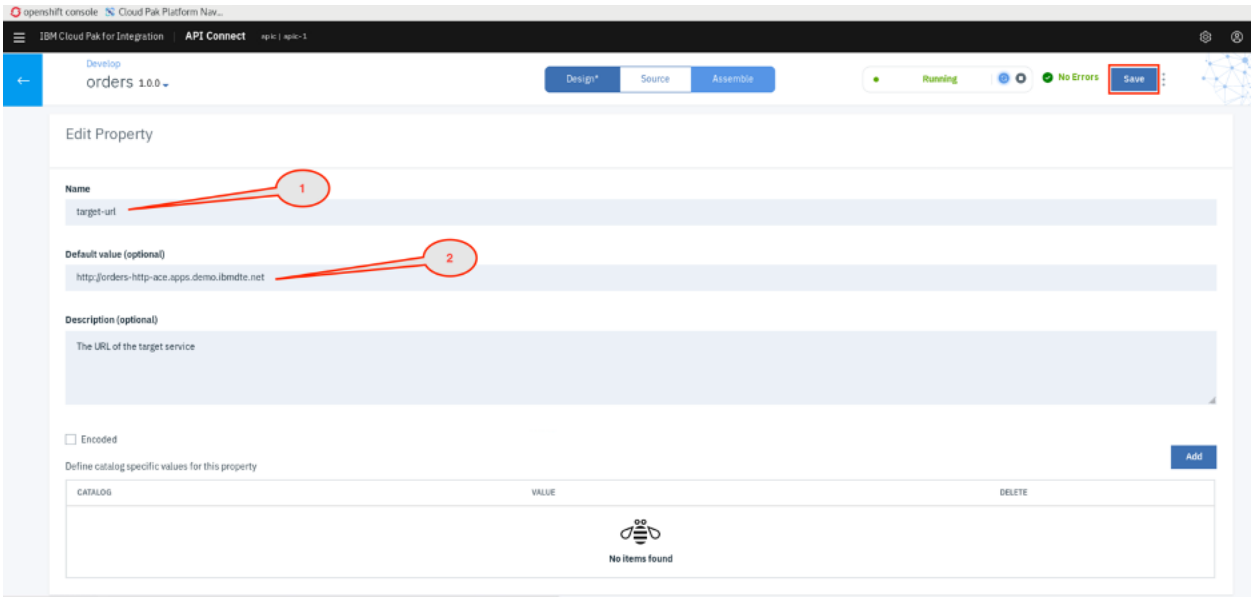
15. In the API Setup page. You configure your API.



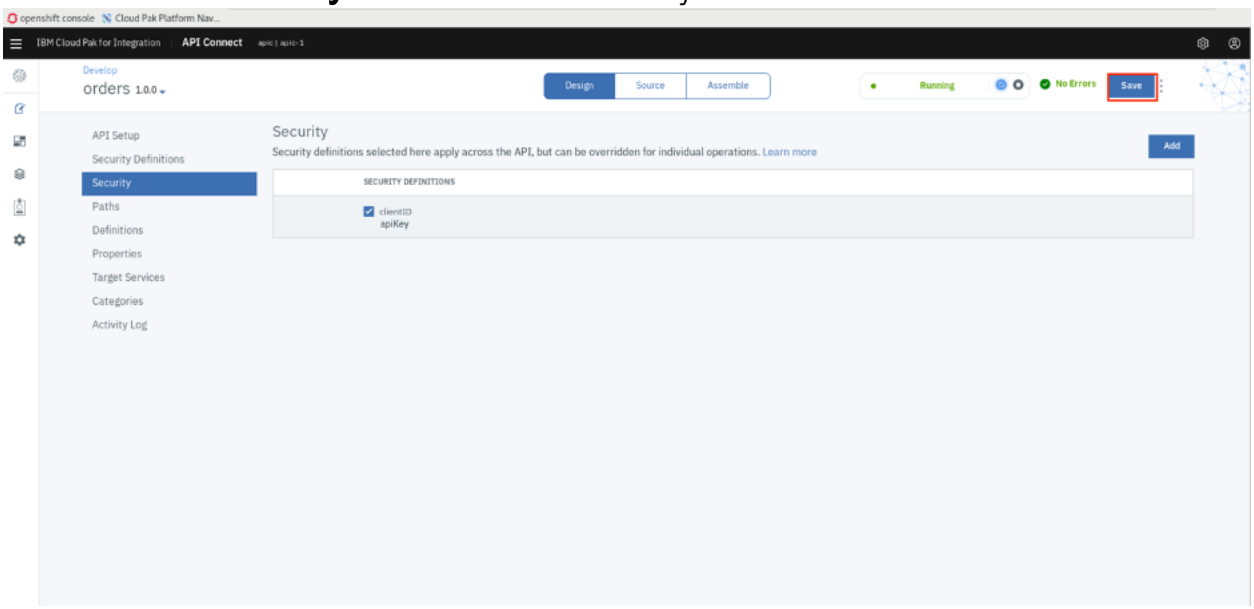
16. In the Design page, click **Properties** and the click **target-url** link, to enter the target gateway.



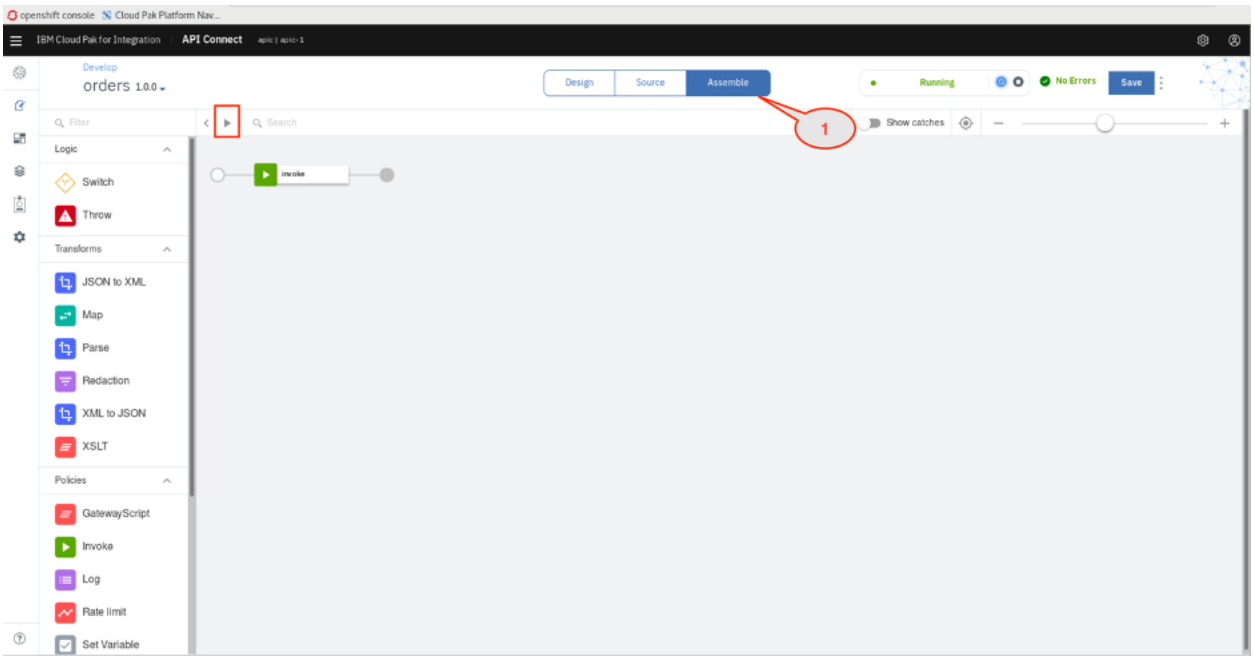
17. Enter as Name: **target-url** and Default Value (optional): http://orders-http-ace.apps.demo.ibmdte.net .



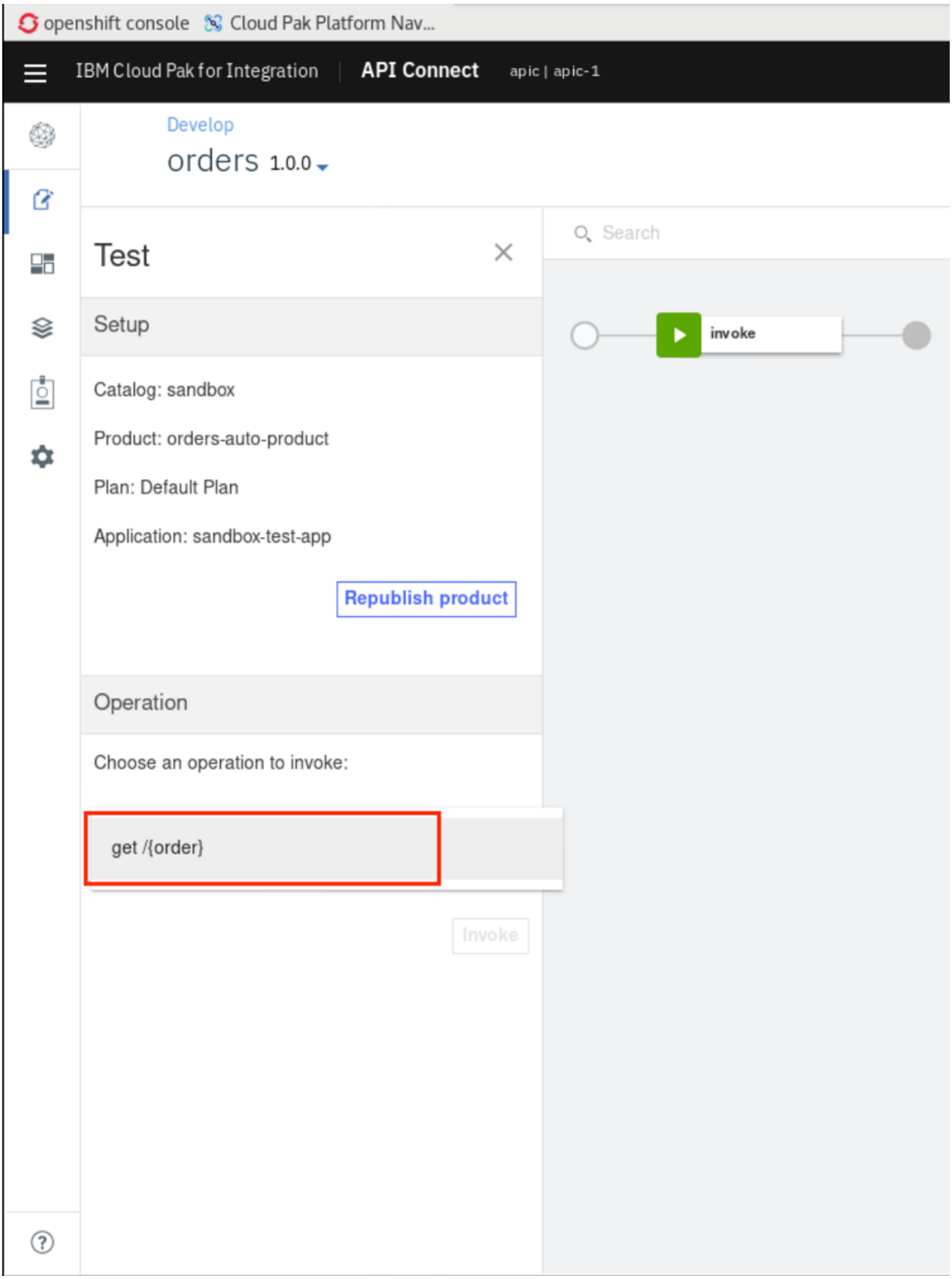
18. Go to **Security**. Check the Security Definitions click **Save .**



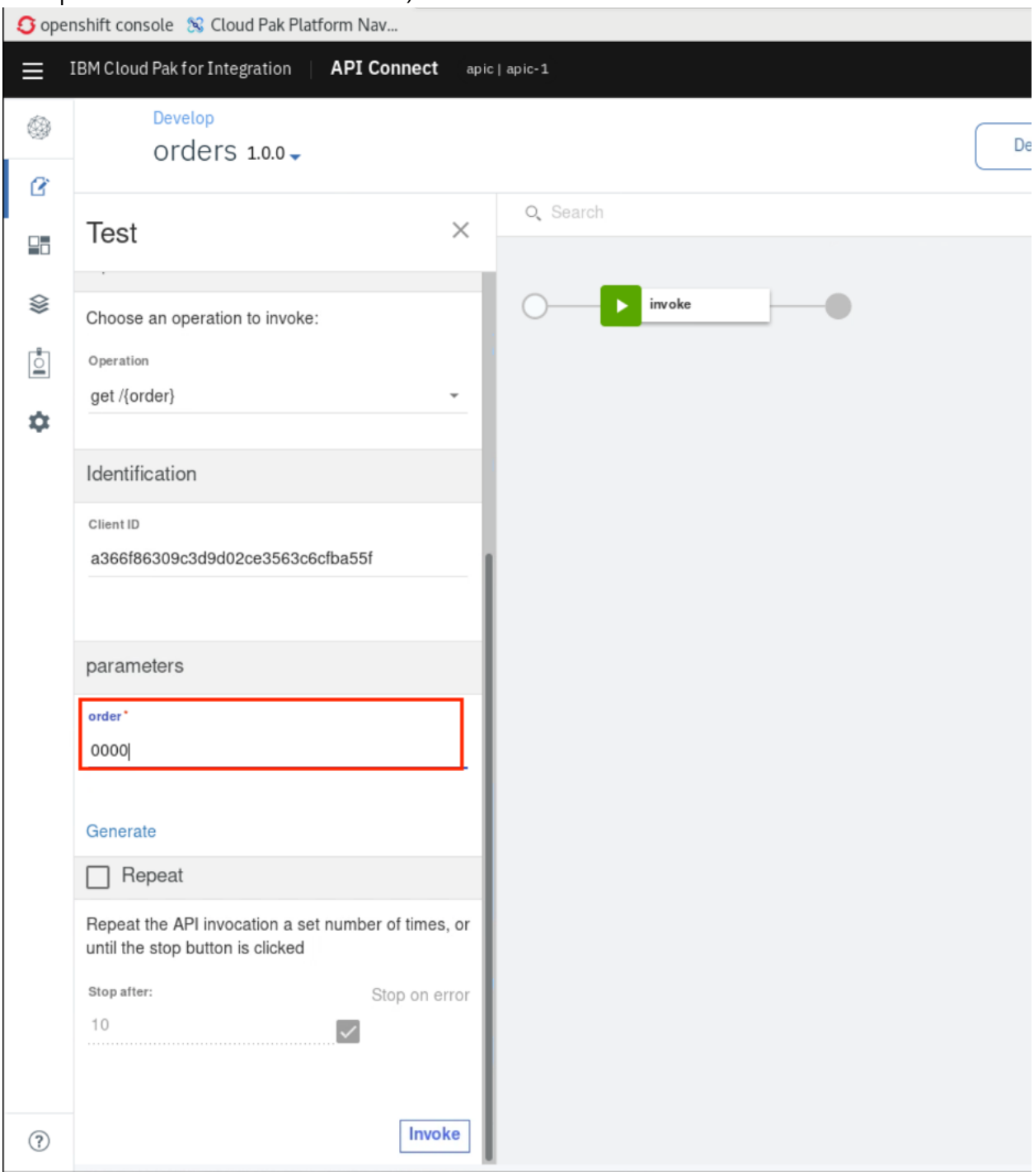
19. To test the API, Click **Assemble**, then click the **Test button .**



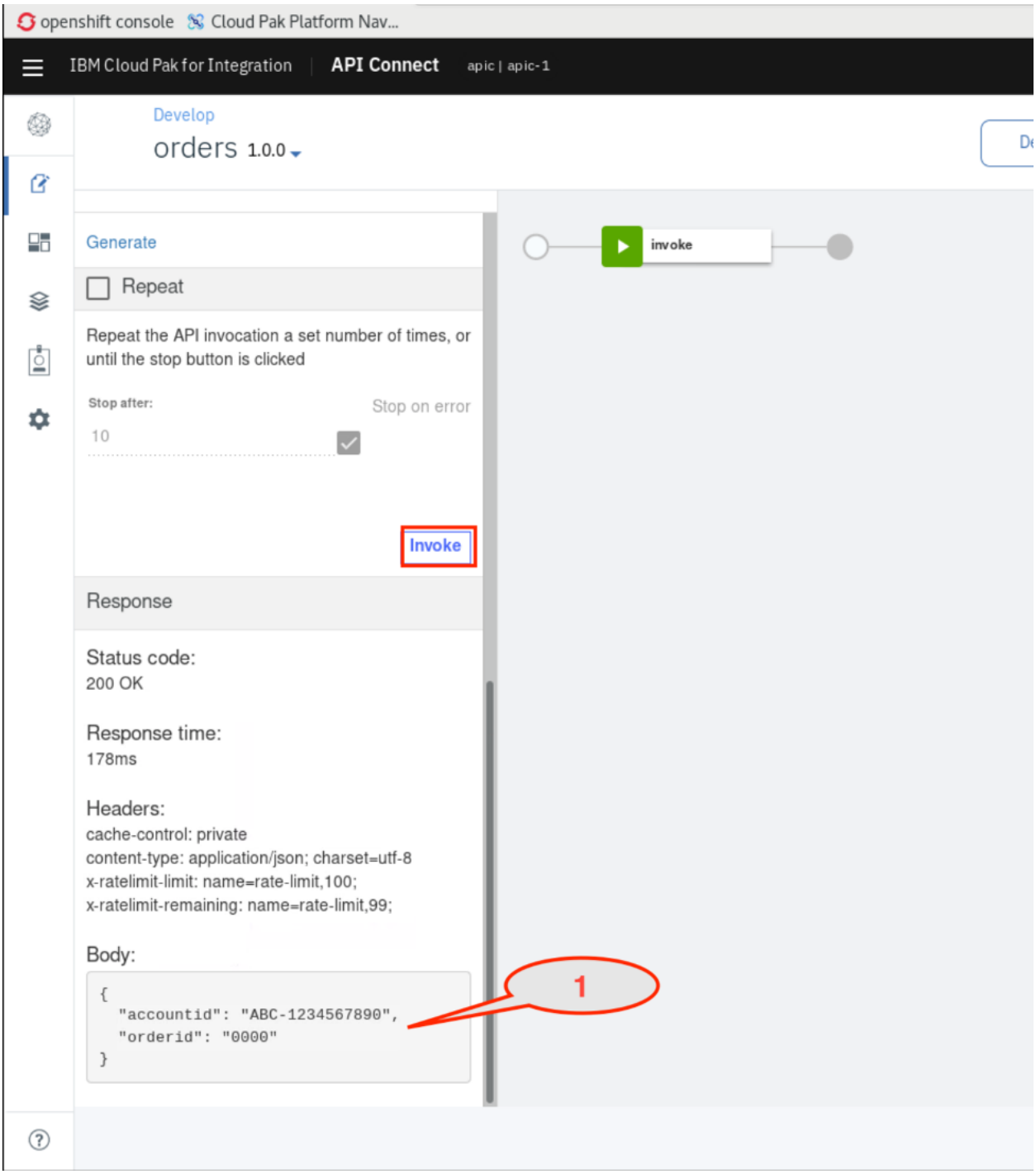
20. In the Test window, click the **operation** and choose **get/(order).** After the API is published, the Catalog, Product, Plan, and Application are displayed under setup. The default plan has a rate limit of 100 calls per hour.


openshift console
Cloud Pak Platform Nav...
IBM Cloud Pak for Integration
API Connect
apic | apic-1
Develop
orders 1.0.0
Test
Search
Setup
Catalog: sandbox
Product: orders-auto-product
Plan: Default Plan
Application: sandbox-test-app
Republish product
invoke
Operation
Choose an operation to invoke:
get /{order}
Invoke

21. The clientId fields auto-populate with the test application values. Under parameters under **order**, enter a value **0000** .



22. Click **Invoke**. Scroll down and see Body and check the results. You see a status code: 200 created with a response body containing the results details.



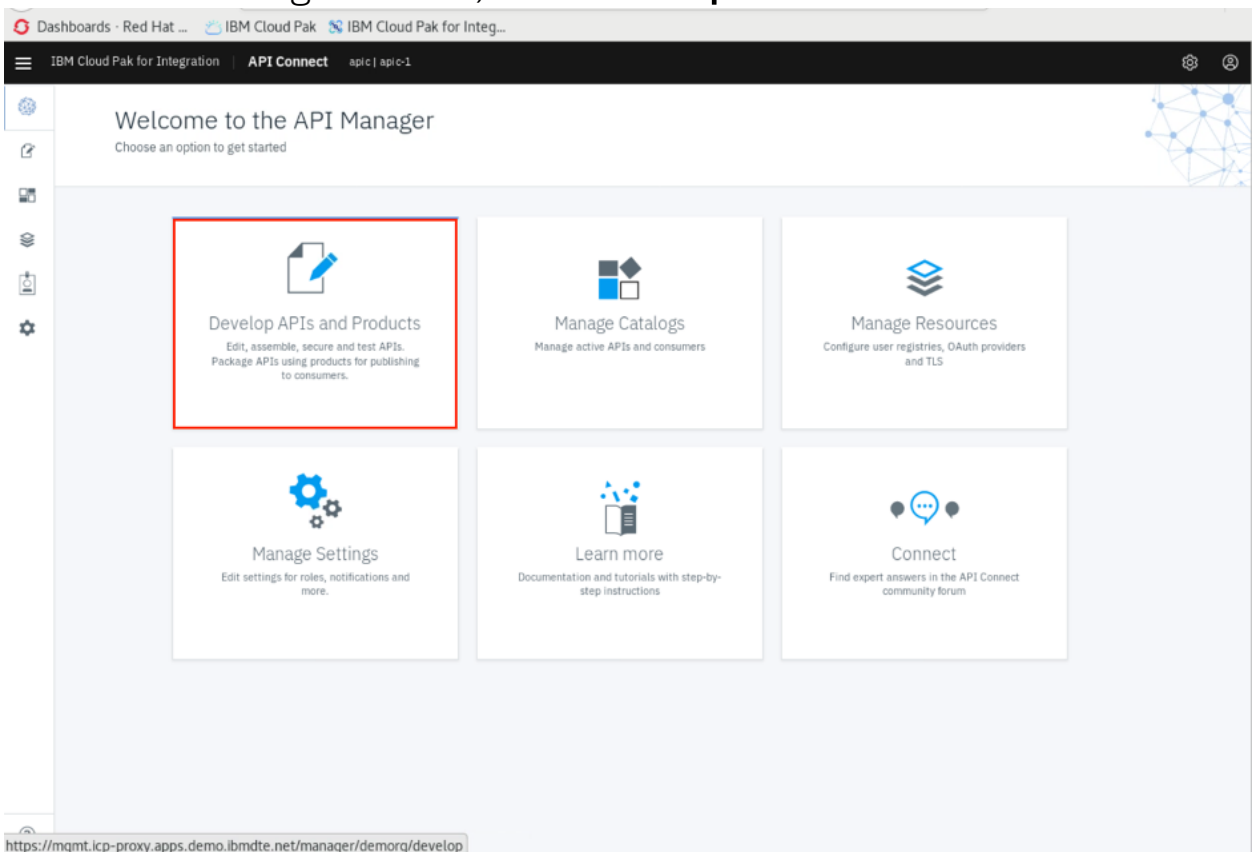
Task 7 - Share the API

Now that you've built, secured, published, and tested your API, the last step is to add it to the Asset Repository. With the Asset Repository, your organization can store, manage, and share all of your integration assets in one central location. Sharing assets in this way increases

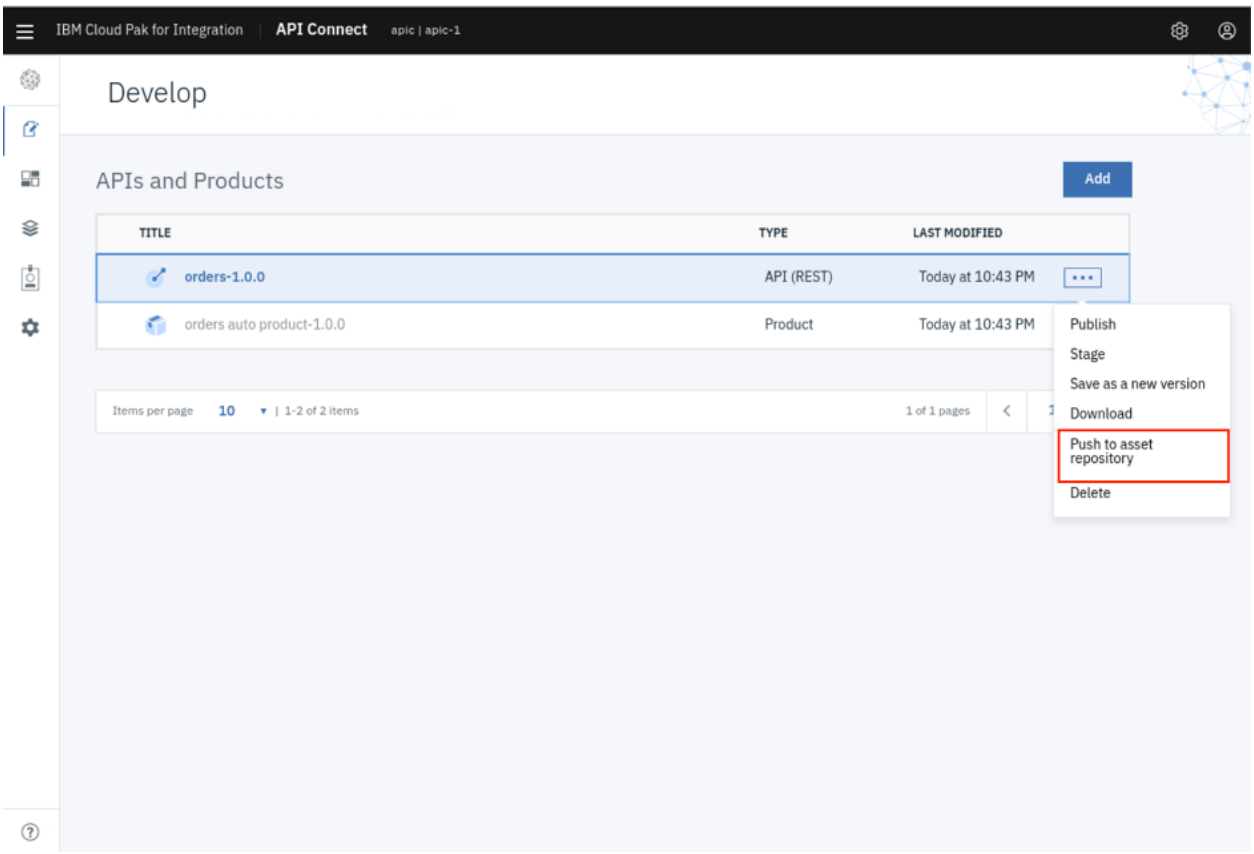
collaboration between teams, avoids unnecessary duplication and boosts productivity.

1. To push our new API to Asset Repository we must return to the API Manager. From the Cloud Pak for Integration home page, choose **apic-1** .

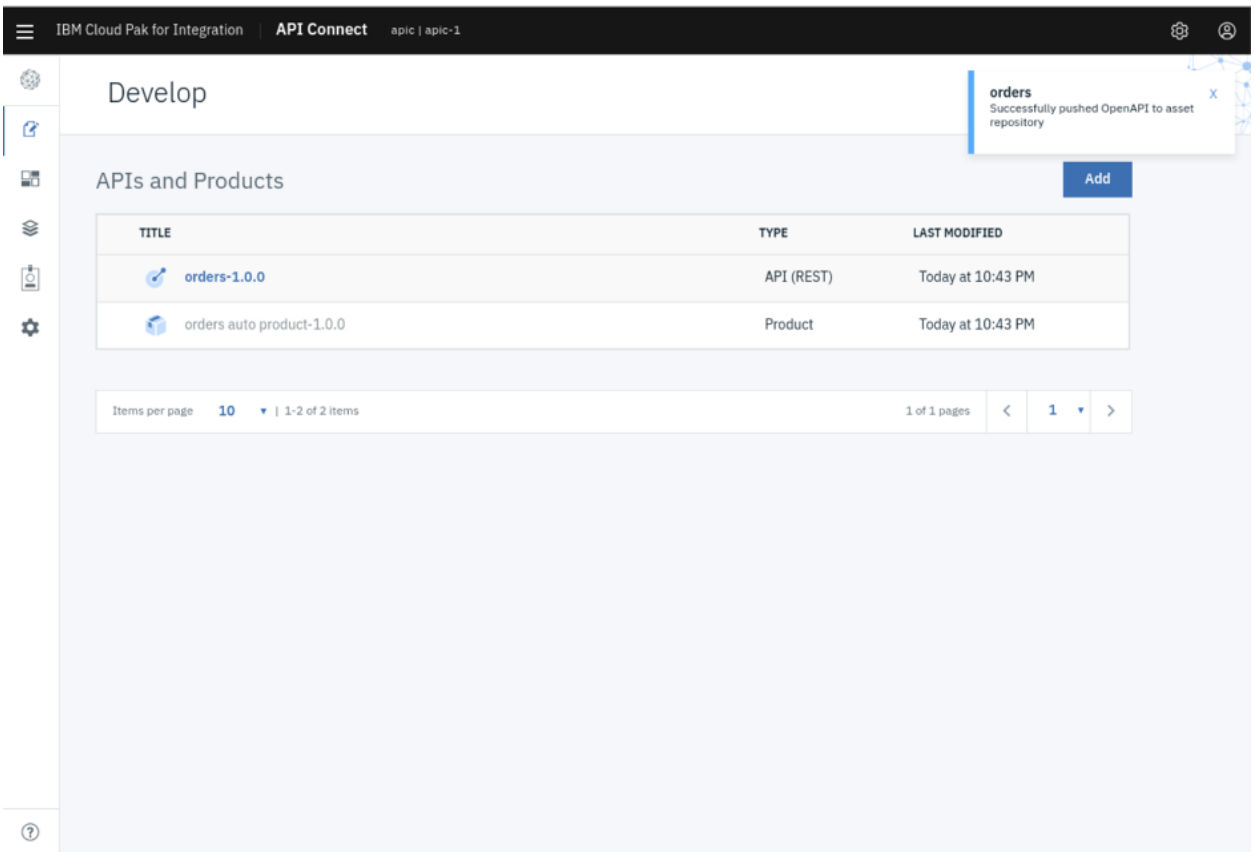
2. In the API Manager screen, click **Develop APIs and Products.**



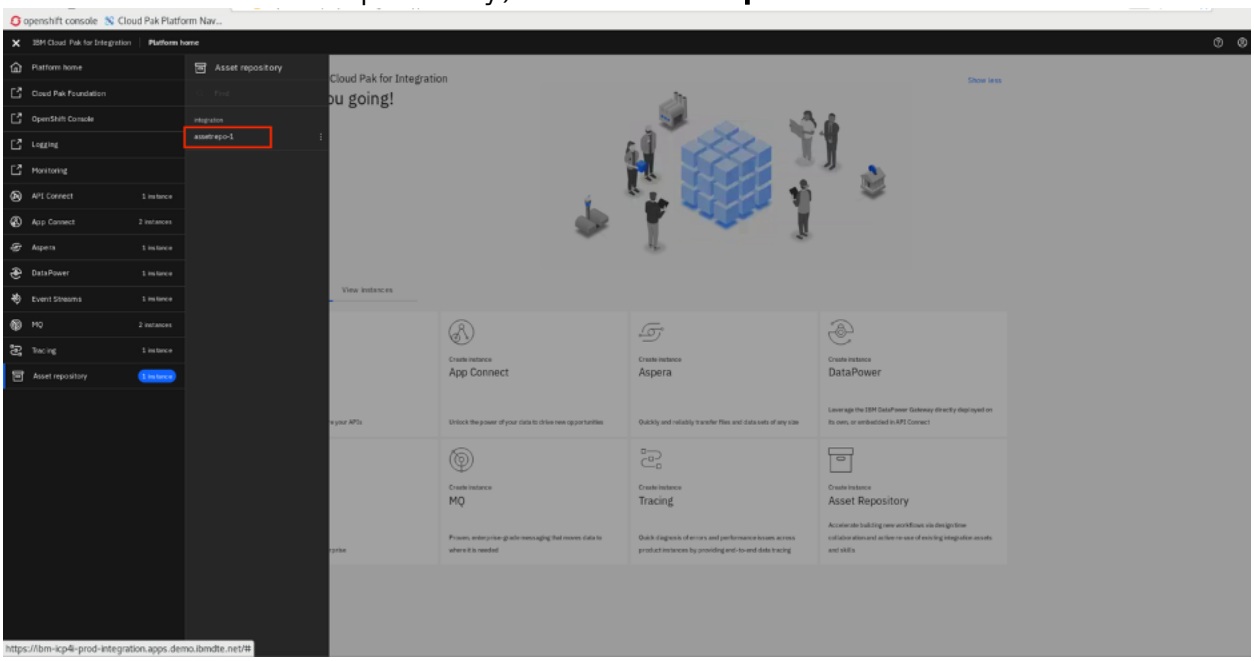
3. Returning to the Develop Screen, select the menu next to your API and from the drop-down, click **Push to asset repository**.



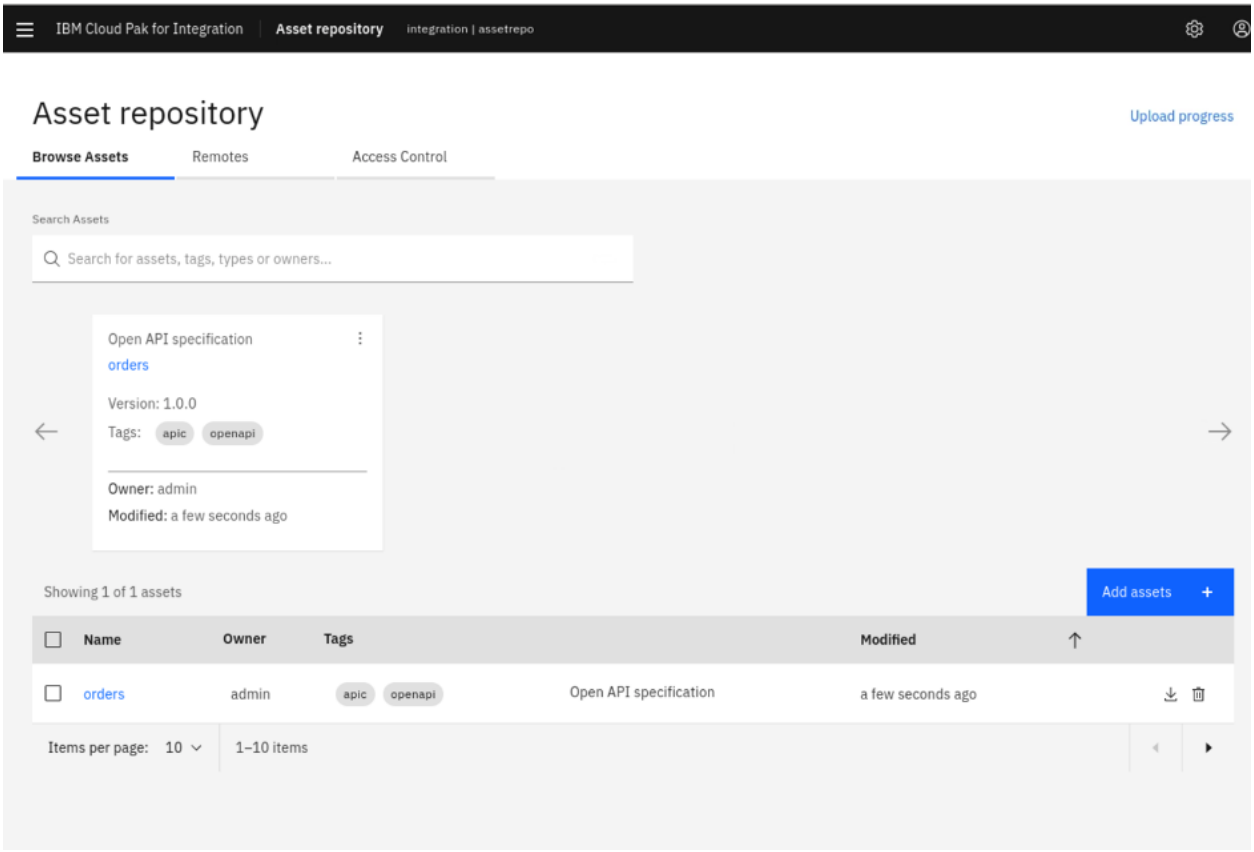
4. Once the upload completes, you see a success dialogue at top-right.



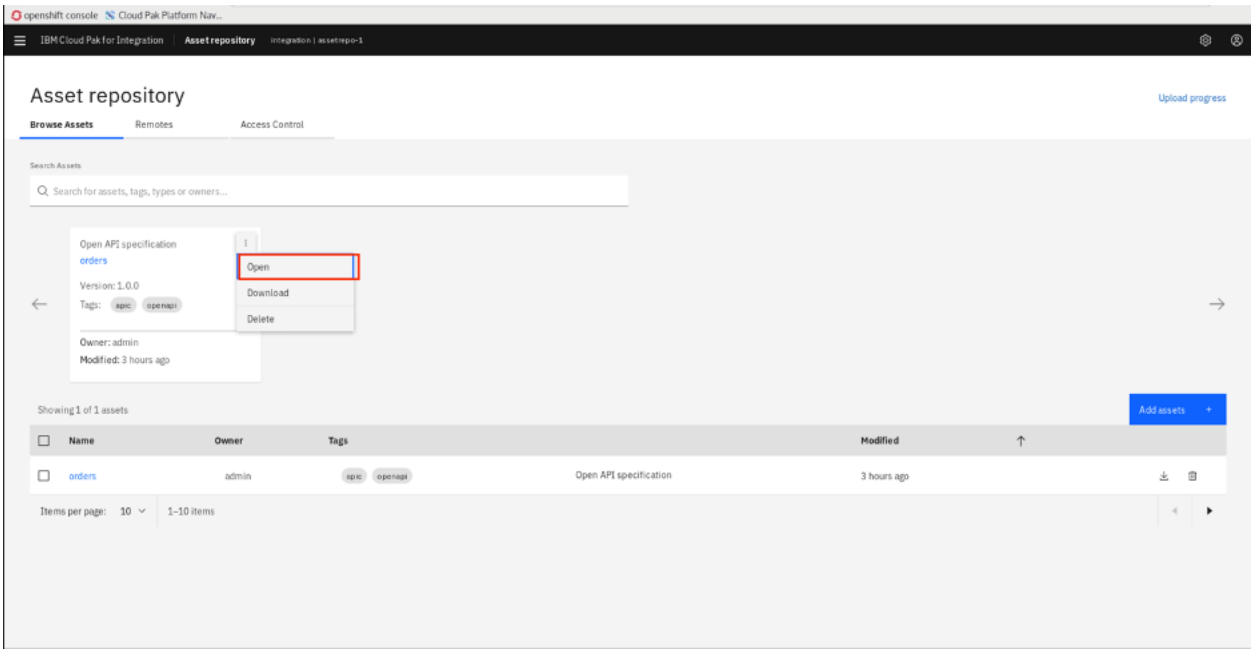
5. To access the Asset Repository, click **assetrepo**-1 .



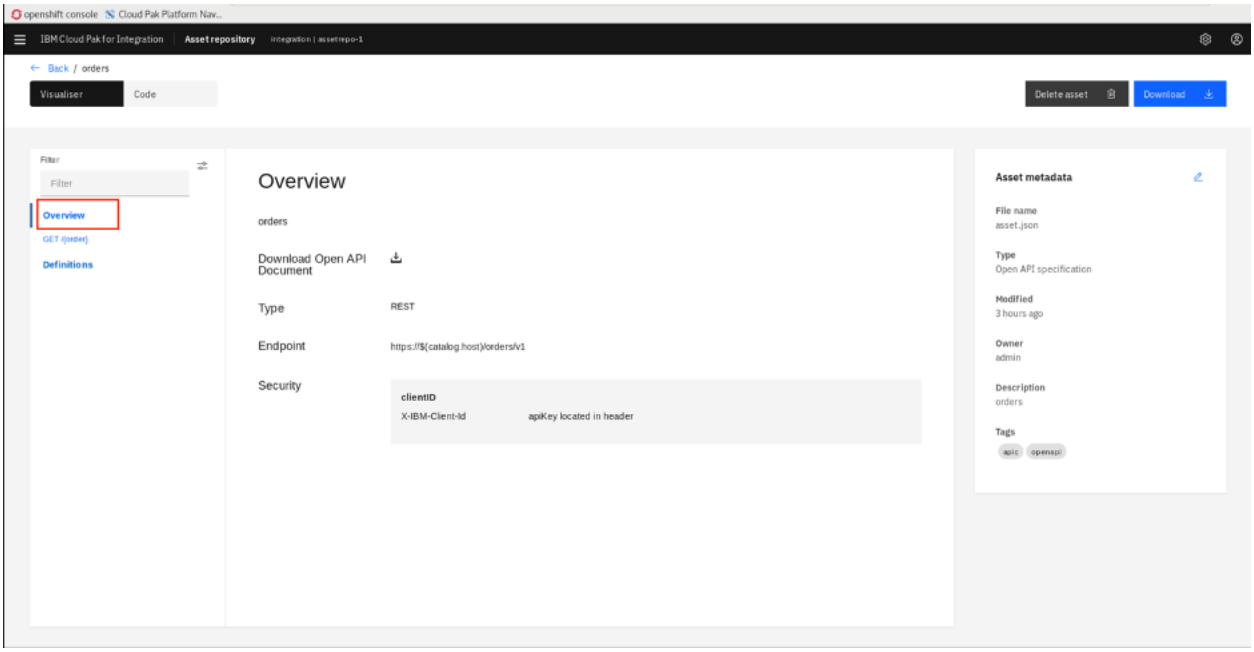
6. You see the orders asset you pushed from API Connect in the previous tasks.



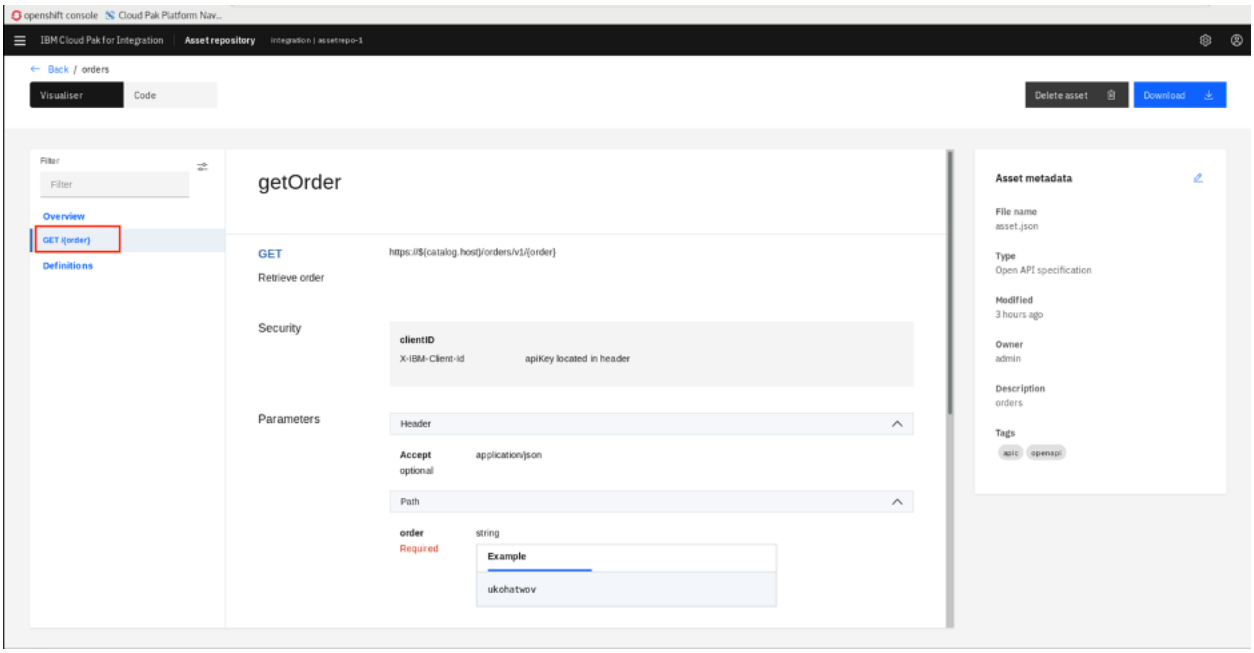
7. Click the ellipsis and choose **Open** from the drop-down menu.to check orders API.



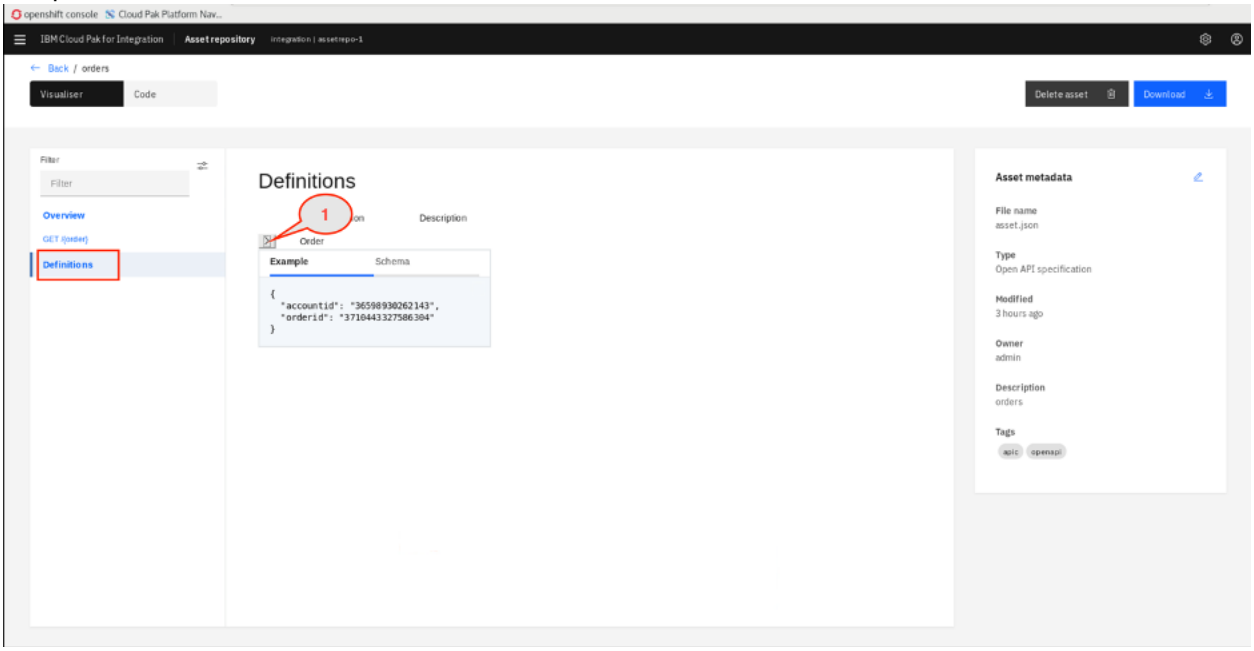
8. You check the API overview.



9. Click **Get/{order}** link you see the API parameters.



10. Click **Definitions** and then click the Arrow (**Order**). You see an example of results.

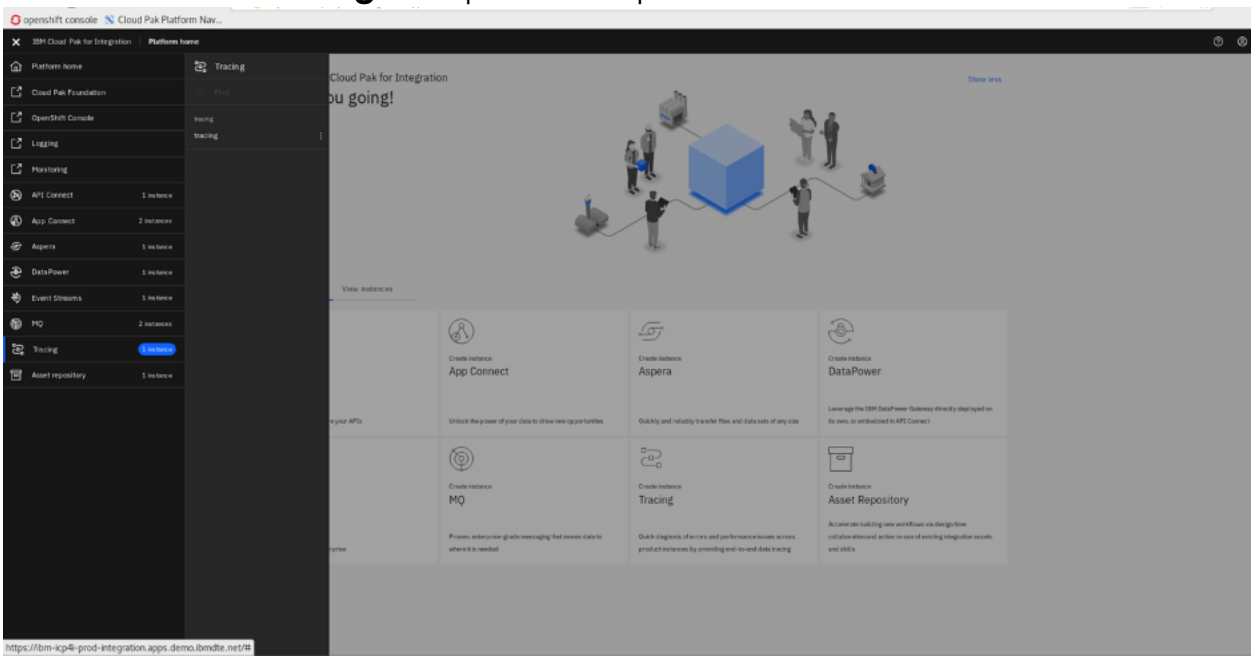


You've successfully added a review. Now your teammates know that this asset is reusable and reliable. Additional information about the asset is available in the sidebar including when the file was created, a description that explains the purpose and use, and any relevant tags

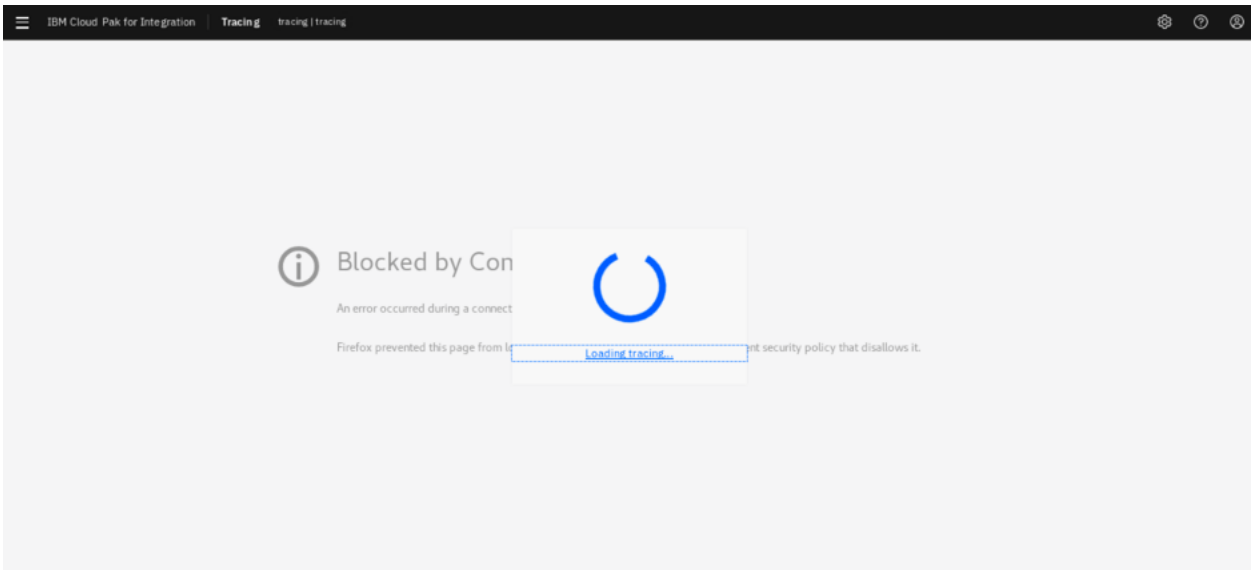
Task 8 - Using Operations Dashboard (tracing)

Cloud Pak for Integration - Operations Dashboard Add-on is based on Jaeger open source project and the OpenTracing standard to monitor and troubleshoot microservices-based distributed systems. Operations Dashboard can distinguish call paths and latencies. DevOps personnel, developers, and performance engineers now have one tool to visualize throughput and latency across integration components that run on Cloud Pak for Integration. Cloud Pak for Integration - Operations Dashboard Add-on is designed to help organizations that need to meet and ensure maximum service availability and react quickly to any variations in their systems.

1. Go to the IBM Pak Cloud Integration main page select **View events** and click **tracing** to open the Operations Dashboard instance.

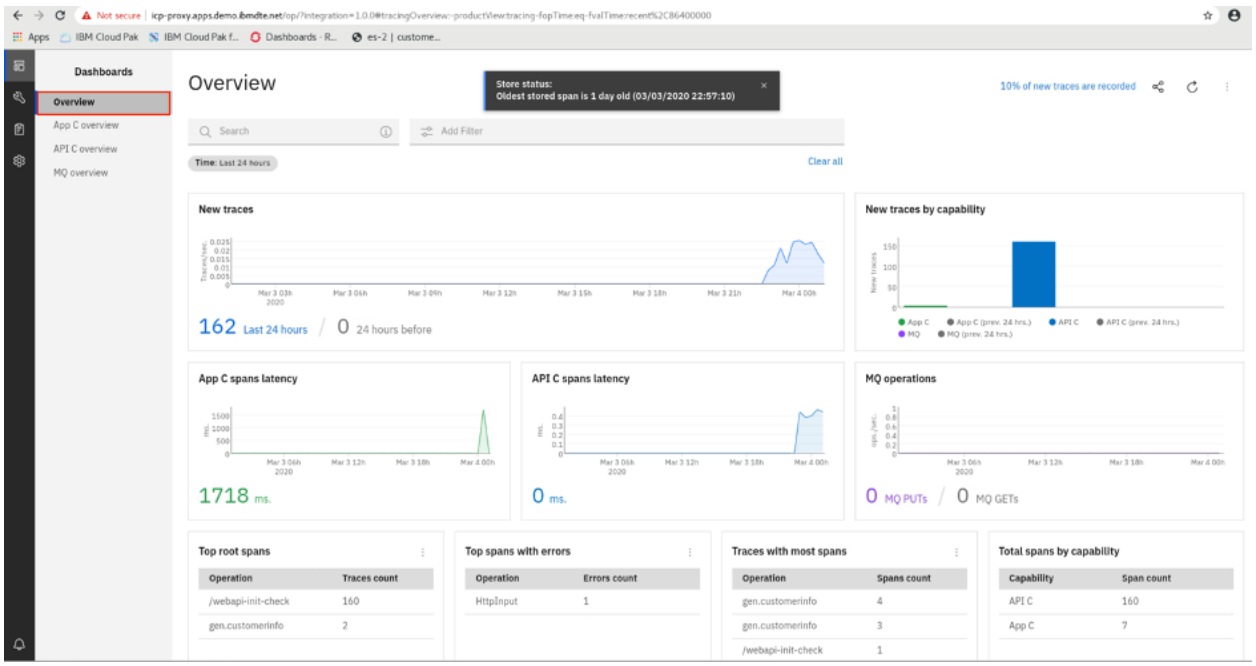


Note: You might see this page. Click the loading tracing to open tracing page.

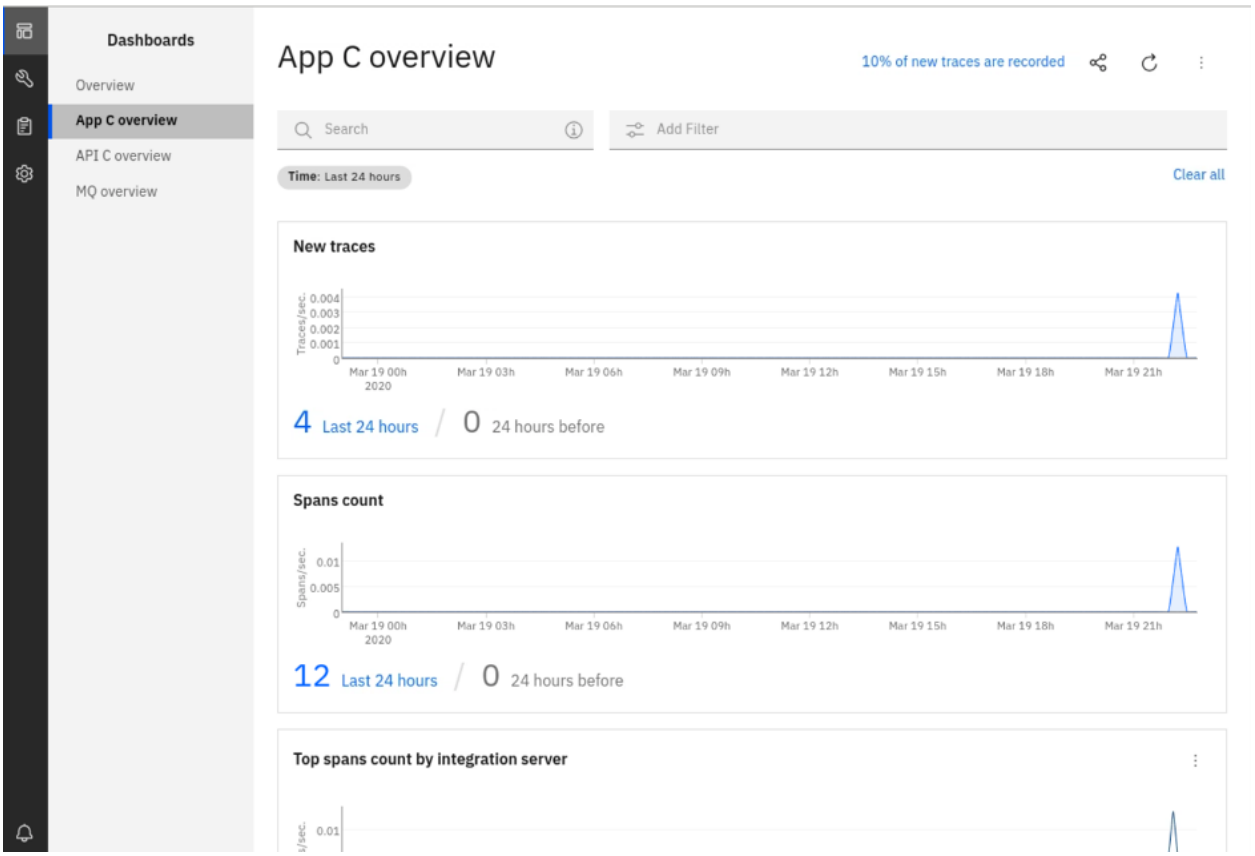


2. In the Tracing page, check the Overview page. You see all products that you can use this tool: APIC ,APP Connect and MQ. (more tracing products will add in the future releases).

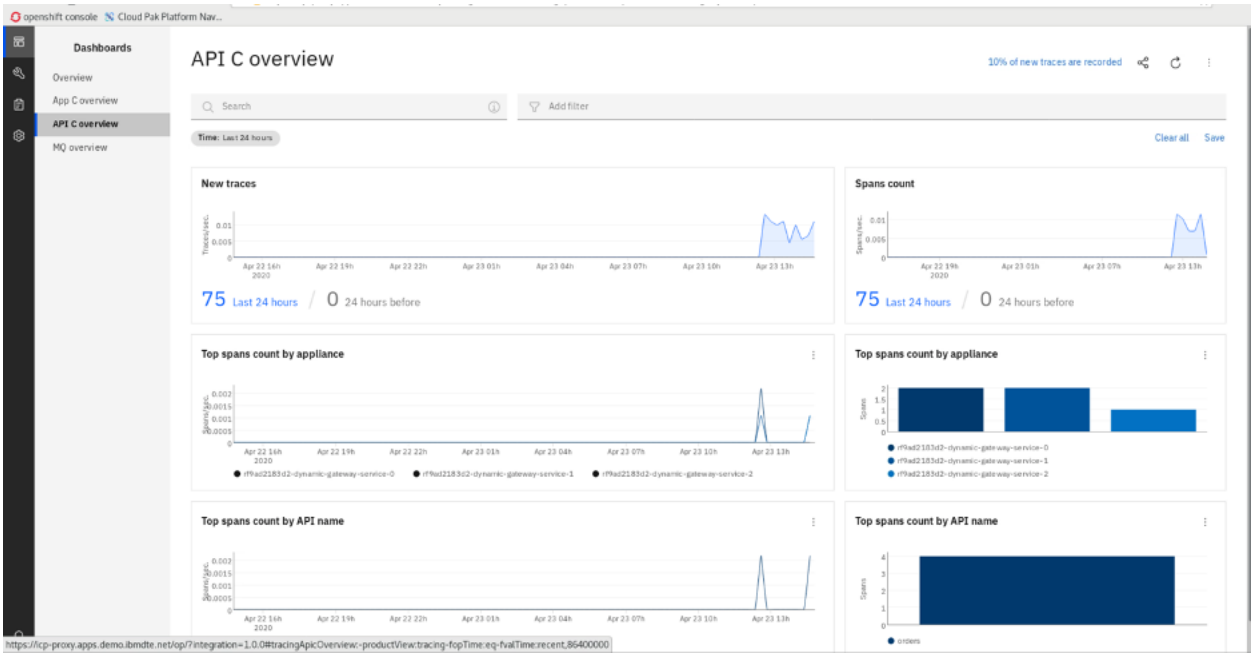
.



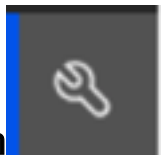
3. You can monitor each product separately. Click **App C overview**.

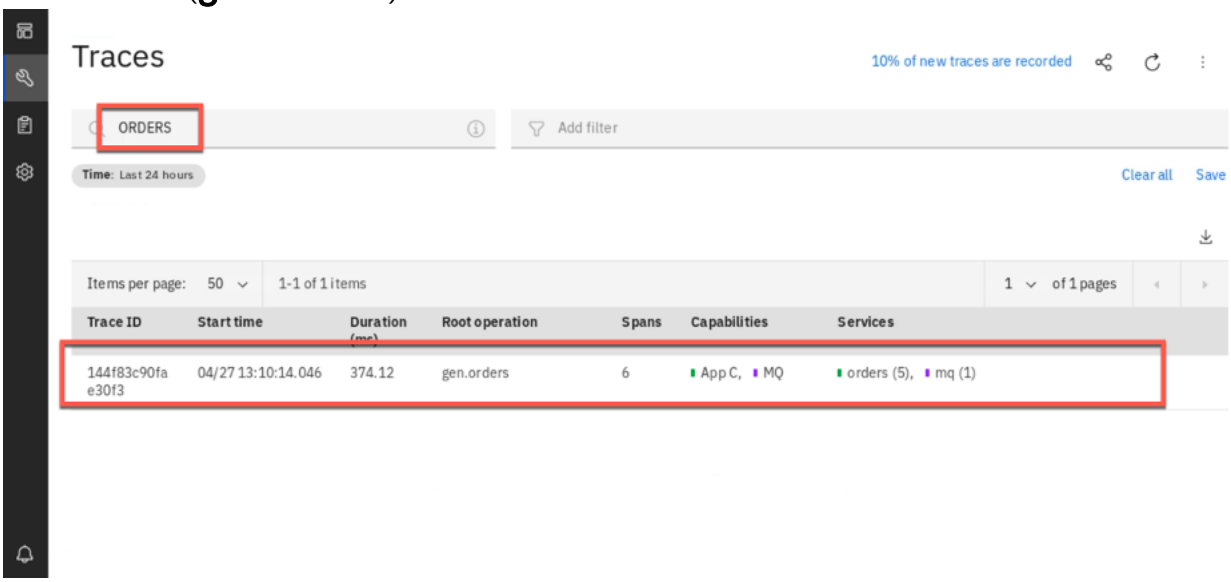


4. See API Connect **overview**.

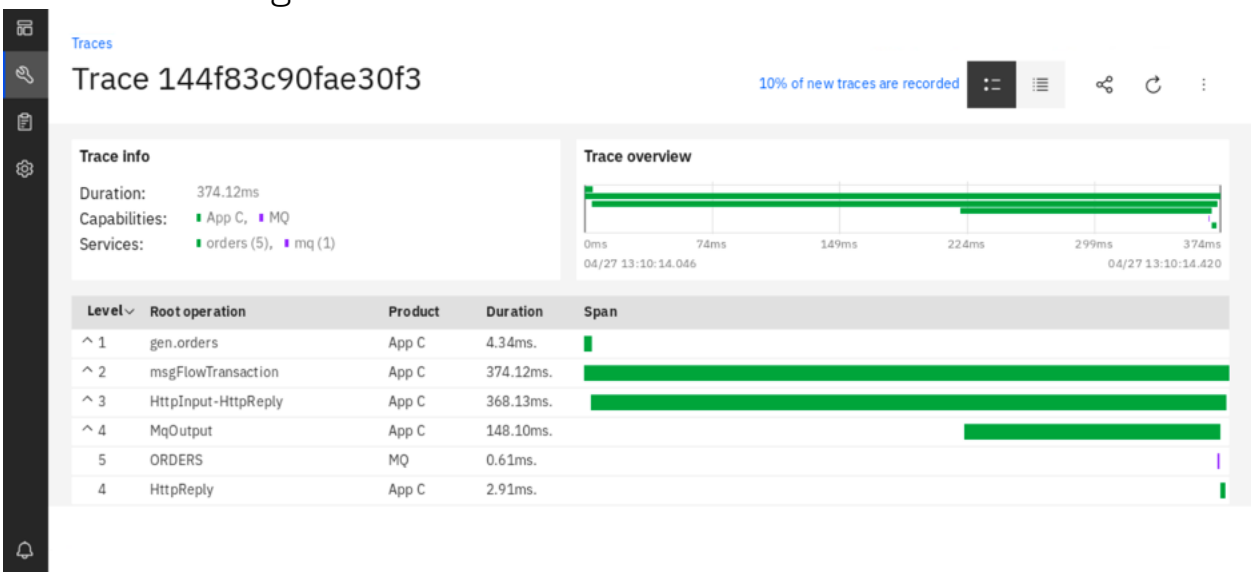


5. Operations Dashboard generated a list of tracing. Select a line to analyze the trace of MQ App Connect Enterprise, and API Connect. select **traces**


icon the menu on the left. Select the line that you want to see the trace.)] Enter the name of App Connect server name: **ORDERS** and click the line (**gen.orders**) .



6. You see the tracing chart .



Summary

You have successfully completed this lab. In this lab you learned how to:

- Deploy a back-end integration to containers that are readily available as a scalable web service.

- Secure access to the back-end integration by creating a secure, governed API using the OpenAPI definition of the integration.
- Use Operations Dashboard to tracing MQ, APIC and APPC

  Now that you've made your back-end integrations ready for external distribution, your developer community is able to access the APIs via a developer portal. The developer portal is included in the platform and provides a full-featured experience to onboard and nurture your API consumers. To try out more labs, go to Cloud Pak for Integration Demos. For more information about Cloud Pak for Integration, go to https://www.ibm.com/cloud/cloud-pak-for-integration.