# IBM i ILE RPG cloud integration sample with IBM Cloud

## Node-RED and IBM API Connect can extend IBM i ILE RPG application to the cloud

Yukihiro Minote
Naho Fujimura
Yohichi Nakamura

March 14, 2018
(First published May 02, 2017)

In this article, you can learn how to connect your ILE RPG programs from cloud applications such as Node-RED on IBM Cloud by using the REST web service call functionality of the integrated web services server.

## Introduction

Today, we have many cloud applications running on IBM® Cloud™ with systems of engagement (SOE) data stored in it. On the other hand, traditional mission-critical data such as financial data, are being tied up to our business activity and the real world. Such data that reside in a company's back end system (such as IBM i or IBM z®) is called systems of record (SOR) data. From the cloud application point of view, data integration between SOR and SOE is the key to success for future business. Moving forward to that future, many technical elements have been discussed and implemented on the internet.

IBM i has a powerful capability to handle business transactions and many customers who have developed such system are still running it in production. In addition, IBM i has provided many web access functions such as web service call through Report Program Generator (RPG) Program Call Markup Language (PCML) invocation. You can easily deploy your own web services on IBM i.

In this article, we focus on Representational State Transfer (REST) service call of ILE RPG from an IBM Cloud application. Combination of these technical elements opens the door to the future of IBM i.

## 1. Getting insights from cloud data and mission-critical application integration sample on IBM i

We can develop and deploy web or mobile native application quickly with cloud infrastructure, such as IBM Cloud. On IBM Cloud, middleware applications have been deployed and are ready to

use. In this article, we have selected Node-RED as the web application building tool. As you may know, Node-RED is the open source that is contributed by IBM. You can build web applications using a graphical user interface (GUI) with drawing tools and minimize application code writing. It is efficient for fast cycle development and deployment in today's business.

The server-side application method is the REST service call using IBM i web service with RPG PCML invocation. Integrated web services server on IBM i can handle this function. To use this function, you need to develop RPG programs that can be called by PCML. There is no need to write web service interface and REST interface code. Integrated web service server will automatically generate the code and definition for you. Node-RED itself can directly call the back-end web service within it. Node-RED application on IBM Cloud requests a REST service call and invokes IBM i RPG program running on your on-premises server. The connection between IBM Cloud and the on-premises IBM i server is done by IBM Secure Gateway that is also offered by IBM Cloud. In this article, you can see a real-world cloud example for IBM i, and learn design patterns for cloud and IBM i transaction integration that can be accomplished today.

## 2. Retrieving mission-critical data by calling IBM i REST service on integrated web services server

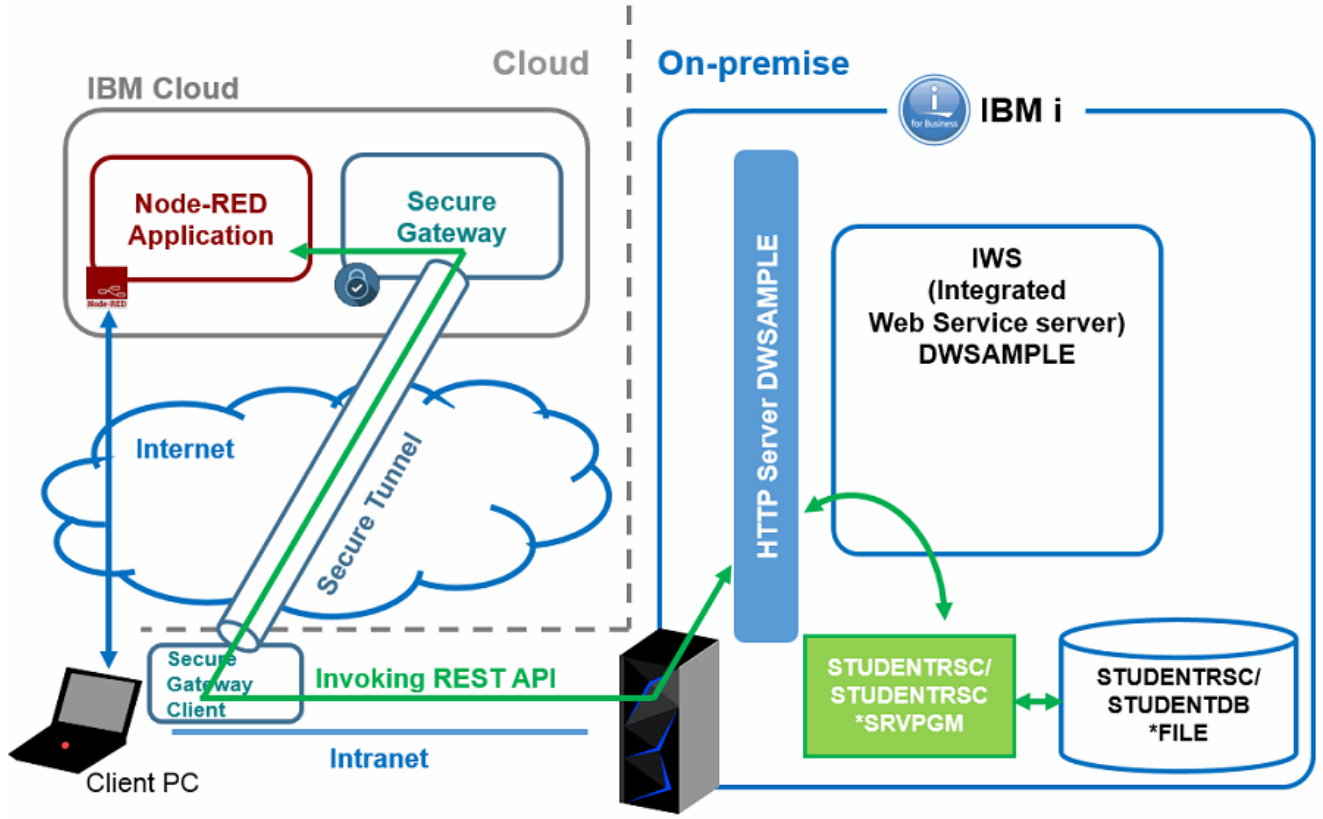
As mentioned earlier in this article, the integrated web services server has a function to deploy ILE programs and service programs as web services using SOAP or REST. ILE programming languages, such as RPG and COBOL, have been used in enterprise application for many years. By using an integrated web services server, you can still access your enterprise data using ILE programs that you are familiar with, and access from both 5250 applications and web applications at the same time. Further, by deploying it as REST web services, you can interchange data with IBM Cloud or SOE applications much more easily.

Here we will use the Student Registration Application, which is introduced in Building a REST service with integrated web services server for IBM i, Part 3, as a sample and guide you through connecting it with Node-RED application on IBM Cloud.

### Sample environment

Figure 1 shows the image of a sample environment setup.

**Figure 1. Sample environment**




Before guiding you the steps, you will need to have the *student information application*, written in ILE RPG, running on your on-premises IBM i server.

We will be deploying this application as an integrated web services REST service. To access this REST web service running on your on-premises server from IBM Cloud, you need to configure Secure Gateway.

Let us assume you need to deploy a sample *student search application* to your Node-RED on IBM Cloud. This *student search application* will call the REST-based web services on IBM i using HTTPS. This application implements REST call to the enterprise data and have a user interface to get input.

The following sections guide you how to:

1. Configure REST web service on an integrated web services server
2. Set up Secure Gateway
3. Call an integrated web services REST application programming interface (API) from a Node-RED application

## 3. REST web service configuration on integrated web services server

As mentioned earlier, we will be using the ILE RPG program explained in Building a REST service with integrated web services server for IBM i, Part 3. By following the steps in that article, you will

be able to configure the REST-based web services on IBM i. And to that configuration, we will be adding a few steps to enable the HTTPS call. Before configuring integrated web services to use HTTPS, you need to set up a local certificate authority and a *SYSTEM certificate store using a Digital Certificate Manager (DCM). To configure, see the following references:

- How to Create the Local Certificate Authority (CA) Store in DCM
- How to Create the *SYSTEM Store in DCM

## Configuring integrated web services HTTP server to use HTTPS

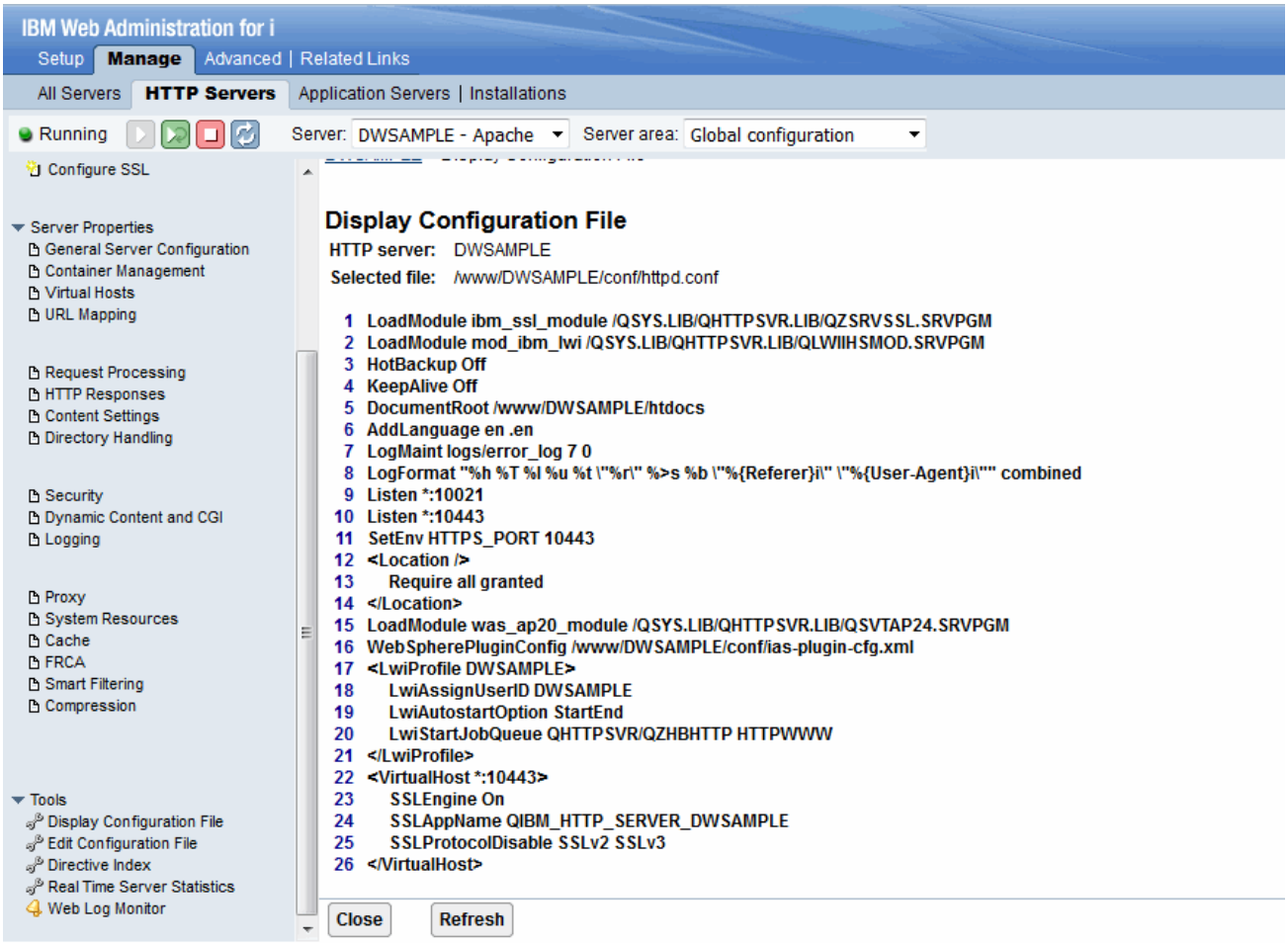
To configure your integrated web services server to use HTTPS connections, refer:

How To Enable an IBM Integrated Web Services (IWS) Server for Secure Socket Layer (SSL) / Transport Layer Security (TLS)

In this reference, disabling the non-SSL port while configuring the SSL port is recommended. But in our scenario, we will listen to both HTTP and HTTPS for verifying access. So, select **No, leave non-SSL port enabled while still configuring SSL port** under **Disable non-SSL port?** instead of **Yes** in the wizard.

After configuring the SSL port for your integrated web services server, you can check your configuration from **Display Configuration File** in the **Tools** section in the left navigation view of Web Administration for i (also known as HTTPAdmin). You can see that SSL setting will be added to your configuration file as shown in Figure 2.

## Figure 2. Configuration file of HTTPS enabled HTTP server




## Download certificate for HTTPS connection with Secure Gateway

In the next step, we will configure Secure Gateway to connect on-premises IBM i server from IBM Cloud. To enable the HTTPS connection, you first need to import the certificate that you created in the step, How To Enable an IBM Integrated Web Services (IWS) Server for Secure Socket Layer (SSL) / Transport Layer Security (TLS) into Secure Gateway.

Before importing the certificate, you must download the certificate from your browser. In our scenario, we used Mozilla Firefox for the web browser, but the steps and result might vary depending on the web browser you use.

1. From your browser, access the URL https://<*your-iws-system*>:<*your-iws-https-port*>/web/services/students and see that you can retrieve student information in the JSON format.
2. Right-click the page and click **View Page Info**.
3. In the **Page Info** window, click the **Security** tab. Under **Website Identity** click **ViewCertificate**.
4. Click the **Export** button on the **Details** tab of your **Certificate Viewer** window. Select **X.509 Certificate (PEM) (*.crt;*.pem)** as your file type and download the certificate to your PC. Now you are ready to connect with HTTPS using the certificate you have downloaded.

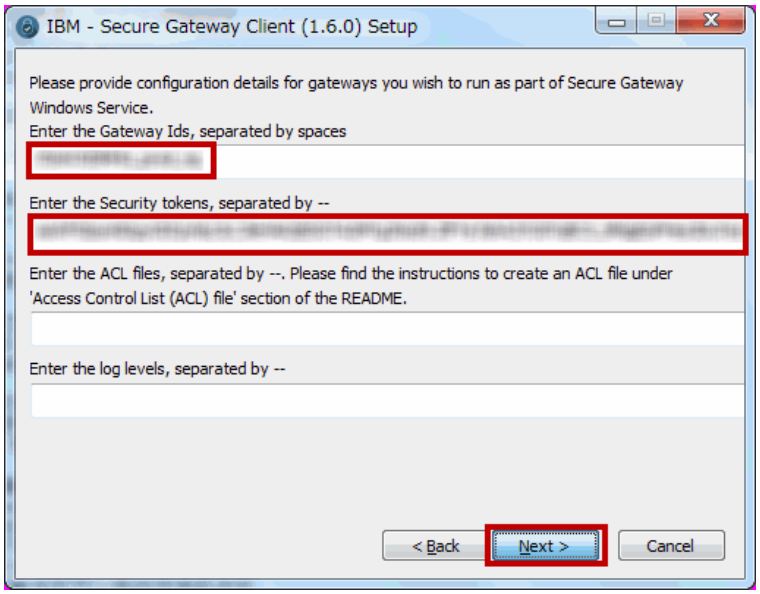
## 4. Setting up Secure Gateway

Now we will set up Secure Gateway on IBM Cloud. Secure Gateway is a service on IBM Cloud that will create a *secure* (as its service name) tunnel between IBM Cloud and the on-premises system. To use this service, you will also need to download Secure Gateway Client to your on-premises system.

In our scenario, we will configure Secure Gateway Client on your local PC that can access your on-premises IBM i server. Follow the steps to set up Secure Gateway.

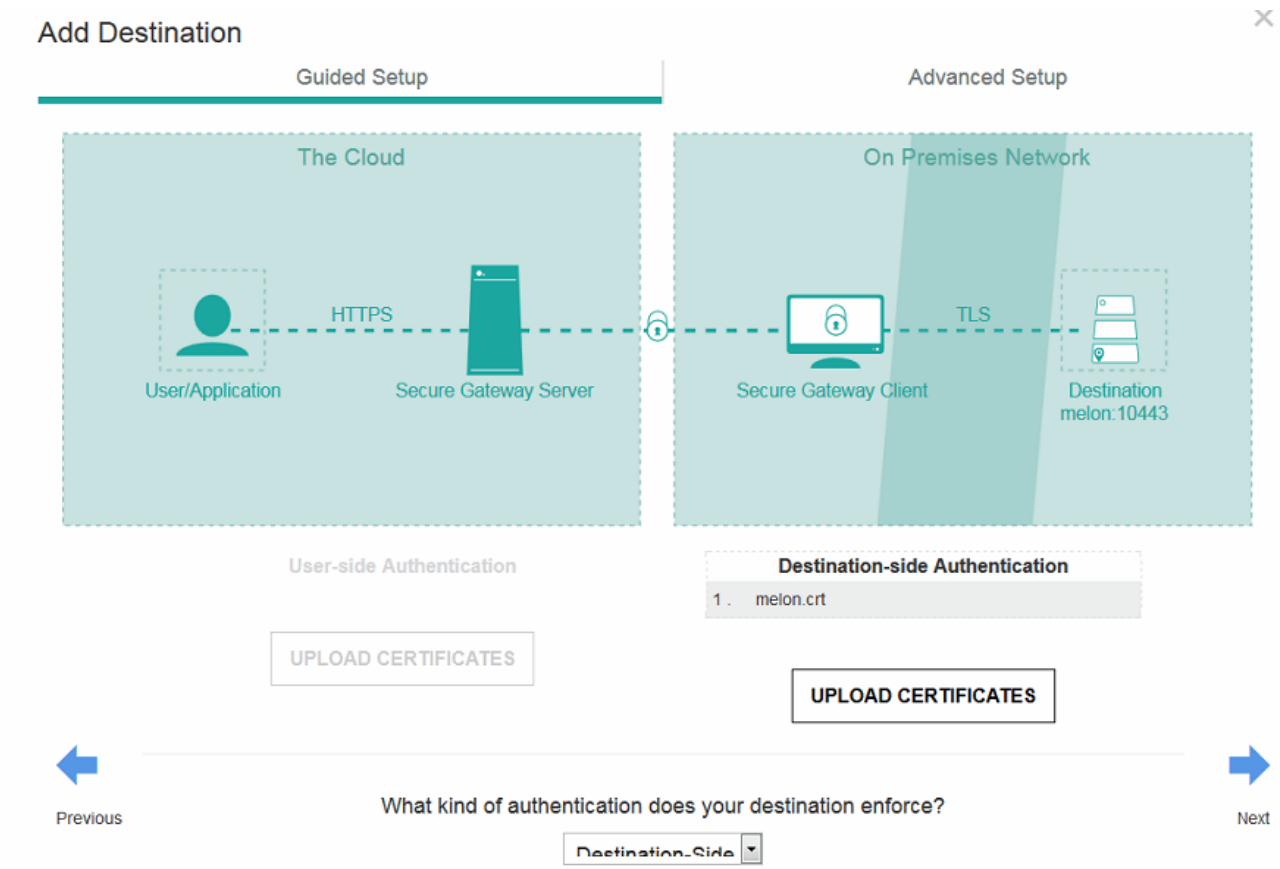
1. Access the IBM Cloud portal using your web browser. If you already have an account, click **Log In** and log in with your IBM ID. If this is your first time to use IBM Cloud, you can sign up for a 30-day trial by clicking **Sign Up**.
2. After logging in to IBM Cloud, you will see your dashboard open. To set up Secure Gateway, click **Catalog** on the top banner and click **Secure Gateway** under the **Integrate** category on the Catalog page.
3. At the bottom of the Secure Gateway page, click **Create** to create your own service. You will be setting up your gateway from here.
4. In the middle of the page, click **Add Gateway** and you will see the **Add Gateway** prompt. Enter your gateway service name and click **Add Gateway**.
5. Your gateway will be created and the dashboard for Secure Gateway will be shown. To set up Secure Gateway Client, scroll down to the bottom of the dashboard, click the **Clients** tab, and then click **Connect Client**.
6. The wizard asks how you would like to connect this new gateway. Because we are using a local Windows PC as our on-premises gateway client, select **IBM Installer** and click the **download** link next to Windows in the **Software Installers** list and save the installer package in your workstation.
7. Install Secure Gateway Client by running the EXE file that you have downloaded. In the wizard, you can choose whether you want to run your Secure Gateway Client as a Windows service. If you plan to use the Windows server as the Secure Gateway Client, running it as a Windows service is recommended (because we are installing it to a local PC in this scenario, and clear the **Please check this option if you would like the Secure Gateway Client to run as a service and restart automatically** check box).
8. Enter your Gateway ID and Security token on the next page of this wizard. You can find this information on the web page from which you downloaded the installer. You can copy the values from that web page and paste them into the wizard as shown in Figure 3. The access control list (ACL) and log level fields are left blank in this example. Click **Next**.

**Figure 3. Specifying Gateway ID and Security token values in the wizard**



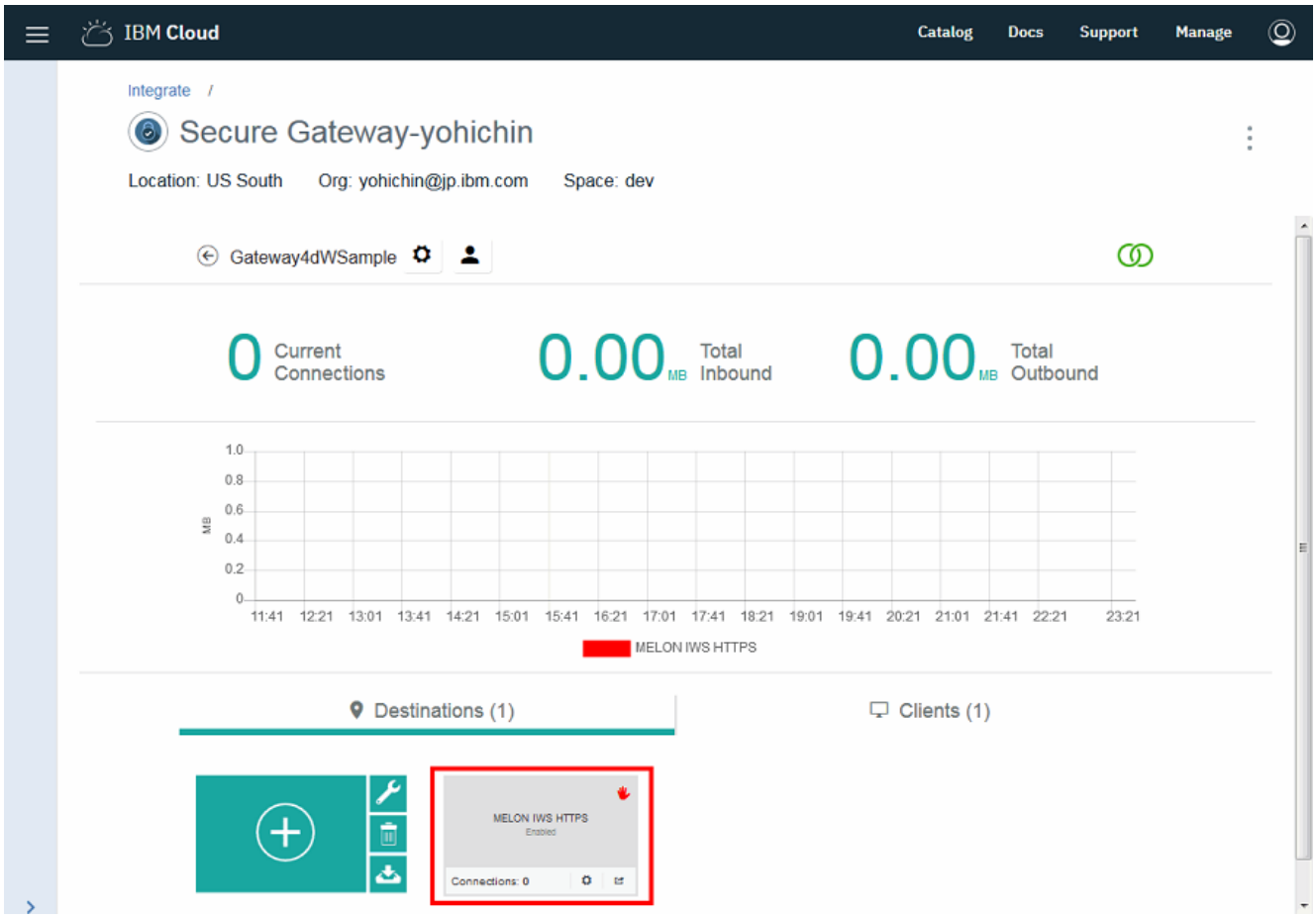

9. Select **Yes** on the next page to use the client GUI and then you can set password and port for the client. In this scenario, we retain the default values and click **Install**. Client installation begins.
10. When the status shows *Completed*, click **Close** to end the wizard. You have now installed the client.
11. Next, set up the gateway service on IBM Cloud. Close the web page from where you have downloaded the installer. On the **Destinations** tab, click the plus mark to add the destination.
12. The wizard to add the destination starts. For this scenario, REST-based web services on IBM i that we will be calling is running on the on-premises IBM i server, and therefore, select **On-Premise** for **Where is your resource located?** and click **Next**.
13. On the second page, enter the host name and port number for your integrated web services server that will be hosting your REST web service and click **Next**.
14. Specify the protocol that you will use to connect to the destination at the third page. For protocol access to your destination, you have a choice of **Mutual Auth HTTPS** for mutual authentication. But in this scenario, we select simple **HTTPS** and click **Next**.
15. We will be using authentication on the destination. So select **Destination-Side** and click **UPLOAD CERTIFICATES**.
16. In the upload window, select the certificate file (*.crt) that you have downloaded earlier and click **Open**. Check that your certificate appears as shown in Figure 4 and click **Next**.

**Figure 4. Specifying the certificate in the wizard**



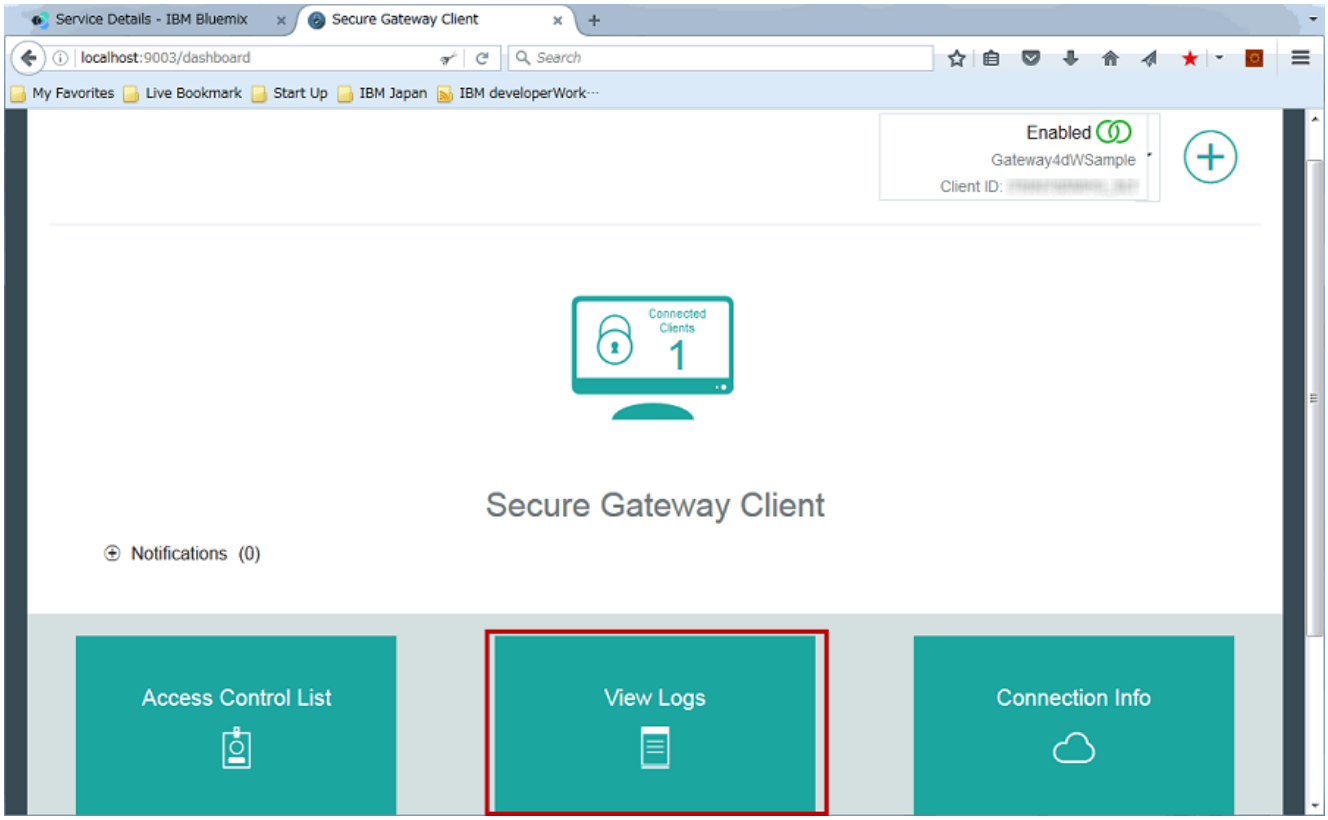

17. You can then specify the range of IP addresses to connect, but in this scenario, retain the default values and click **Next**. After entering a name for the destination, click **Add Destination**.
18. Notice that the destination you have just created is displayed under **Destinations** (as shown in Figure 5). You are now ready with Secure Gateway Server.

**Figure 5. Destination you have just created**



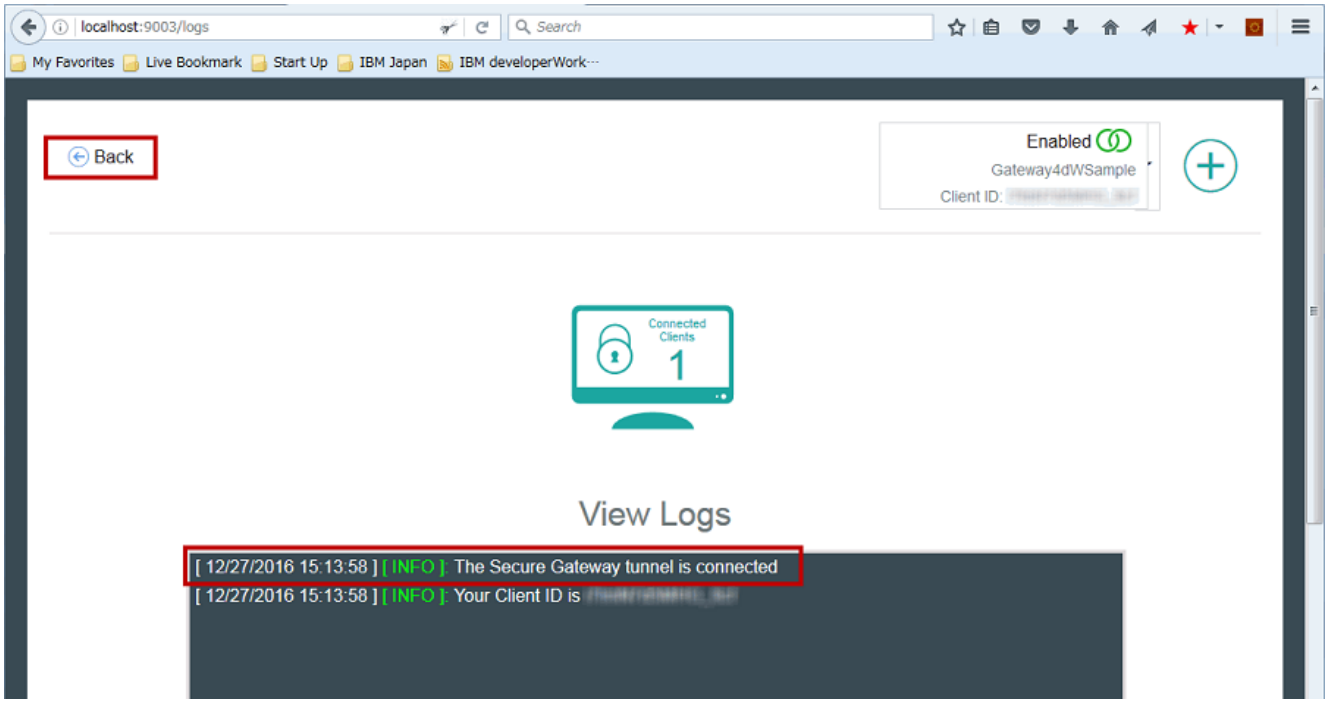

19. Let's see if you can access from your local PC. In your local PC, select **Start** -> **AllPrograms** -> **IBM** -> **SecureGateway Client** and click **Run it as an administrator.**
20. You will be asked if you are going to use the configuration file, *securegw_service.config*, when starting the Secure Gateway Client. Information about the gateway ID that you have entered while installing the client is written in your configuration file. Reply with **y** and press Enter.
21. The Secure Gateway Client GUI opens within your web browser. Click **View Logs** (as shown in Figure 6).

**Figure 6. View Secure Gateway logs**



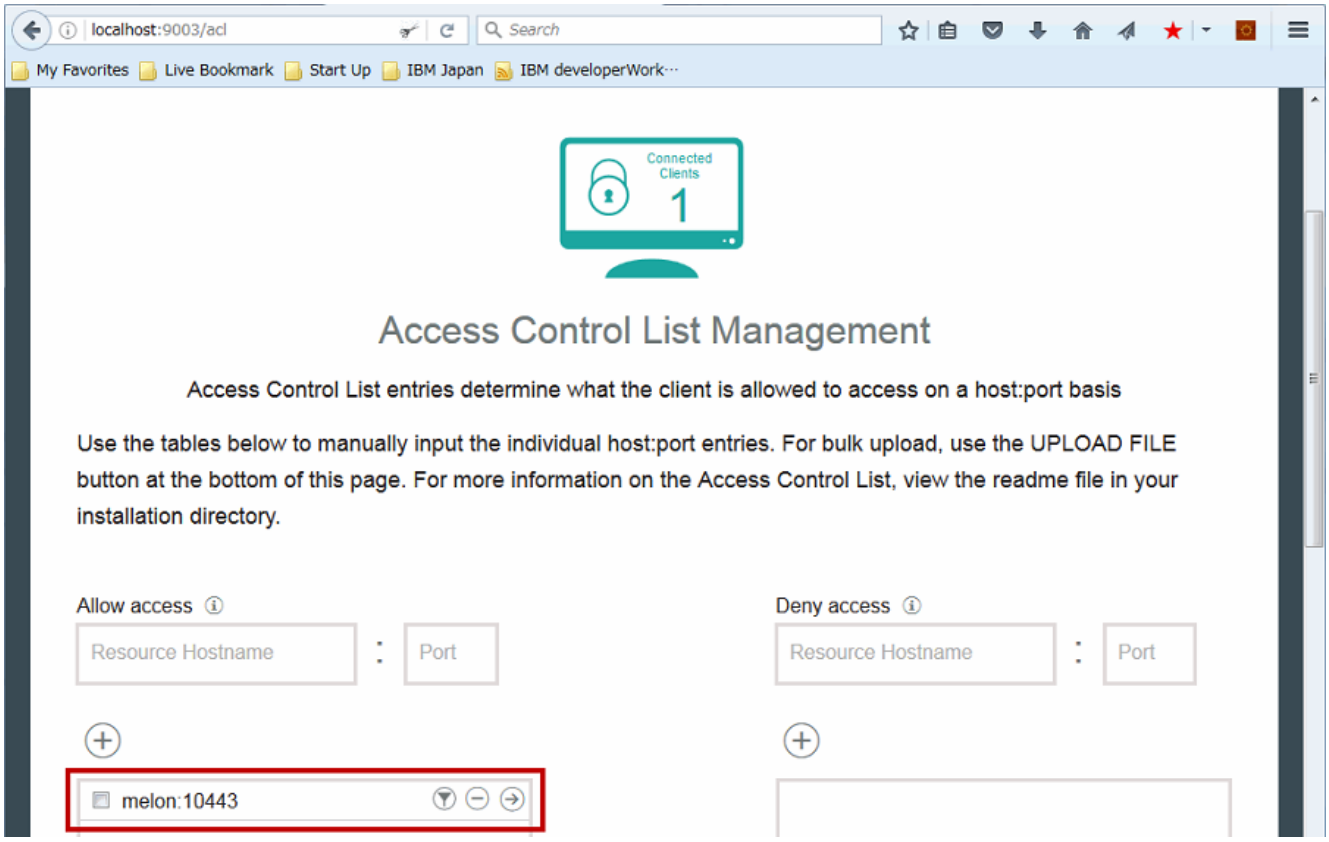

22. You can see in the log that the Secure Gateway tunnel is connected as shown in Figure 7. Click **Back** at the upper-left corner of the page.

### Figure 7. Confirming Secure Gateway logs



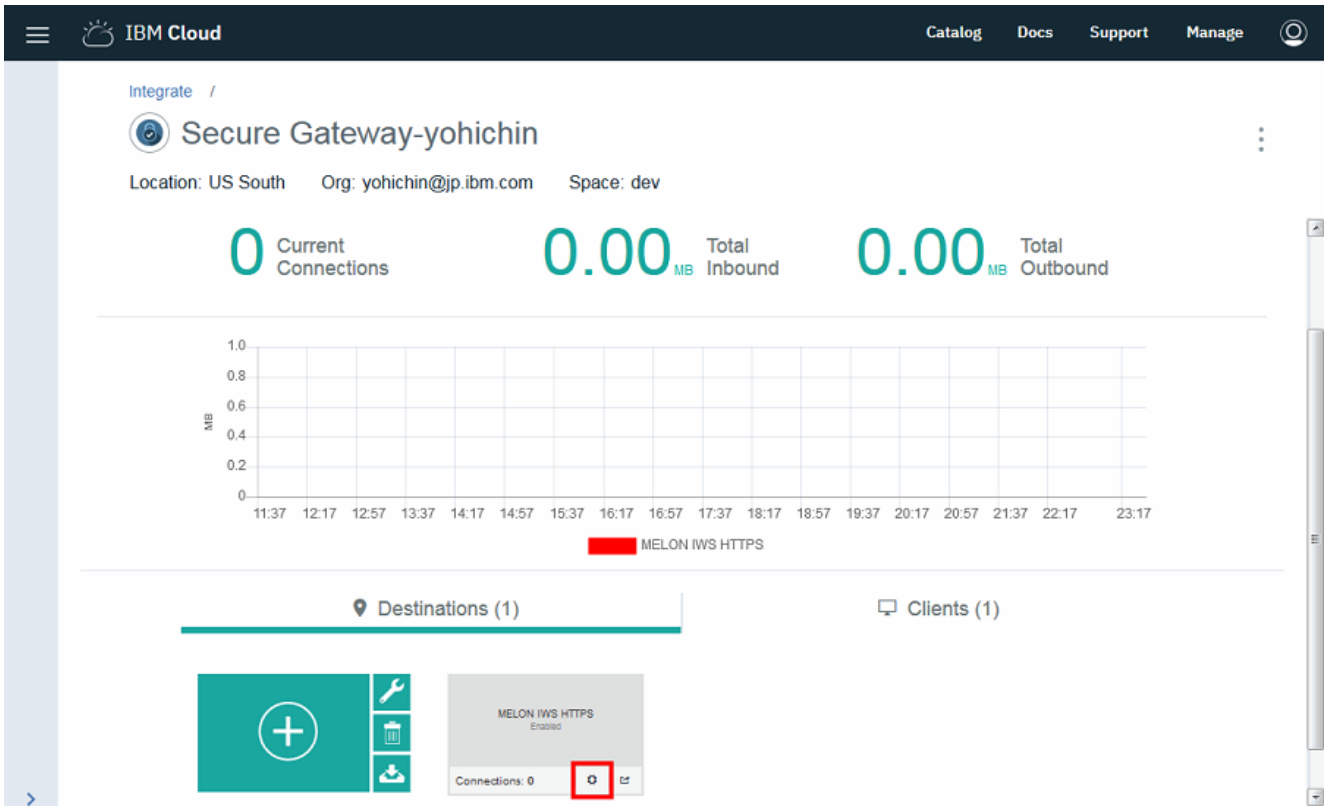

23. Add HTTPS access permission to the integrated web services server from Secure Gateway Client. Click **Access Control List** next to **View Logs** (refer to Figure 6).
24. Under **Allow access** on the **Access Control List Management** page, enter your integrated web services server host name and the HTTPS port number and click the plus sign (**+**) button.
25. Check that the information you have entered is listed under **Allow access**. Click **Back** on the top to return to the top page of Secure Gateway Client. This will end the setup on your Secure Gateway Client.

**Figure 8. Add Access Control List entry**



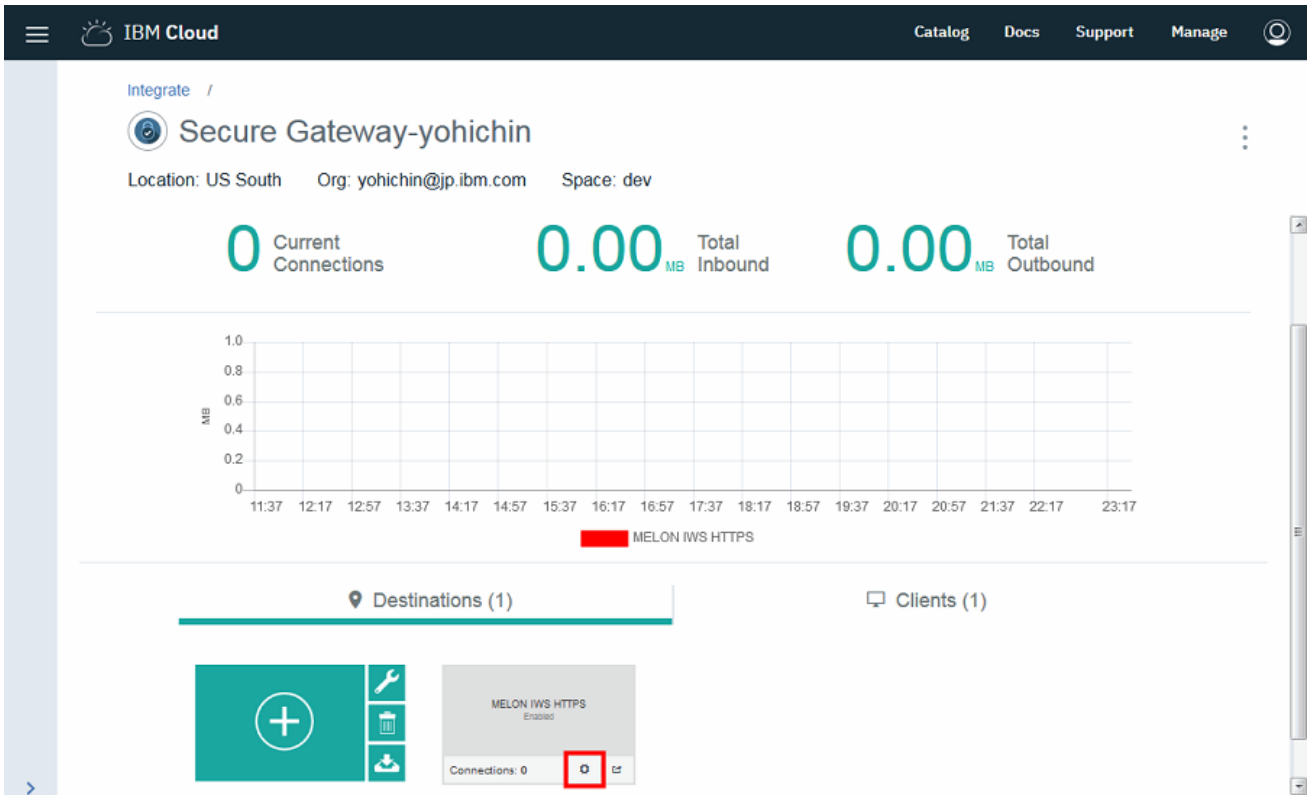

26. Move back to the Secure Gateway Service details page, and press F5 to refresh the page and click the gateway service that you have created.
27. You can see the list of destinations of your Secure Gateway. Click the **Settings** icon at the bottom of the page (refer to Figure 9) in your destination which describes the integrated web services HTTPS access.

**Figure 9. Settings icon of the Secure Gateway destination**




28. Here you can check the details of the destination you have added (as shown in Figure 10). **Cloud Host : Port** refers to the host name and port details that are accessed from IBM Cloud. Using this address, you can access the on-premises destination transparently using Secure Gateway. Make a note on this address because you will be using it later when deploying the Node-RED application.

**Figure 10. Cloud host name and port**




Now we have successfully connected Secure Gateway Server on IBM Cloud and Secure Gateway Client running on a local PC.

## 5. Calling integrated web services REST API from Node-RED application

Finally, we need to access the integrated web services API running on the on-premises IBM i platform from the Node-RED application residing on IBM Cloud through Secure Gateway.

### Configuring Node-RED

Perform the following steps to configure Node-RED on IBM Cloud:

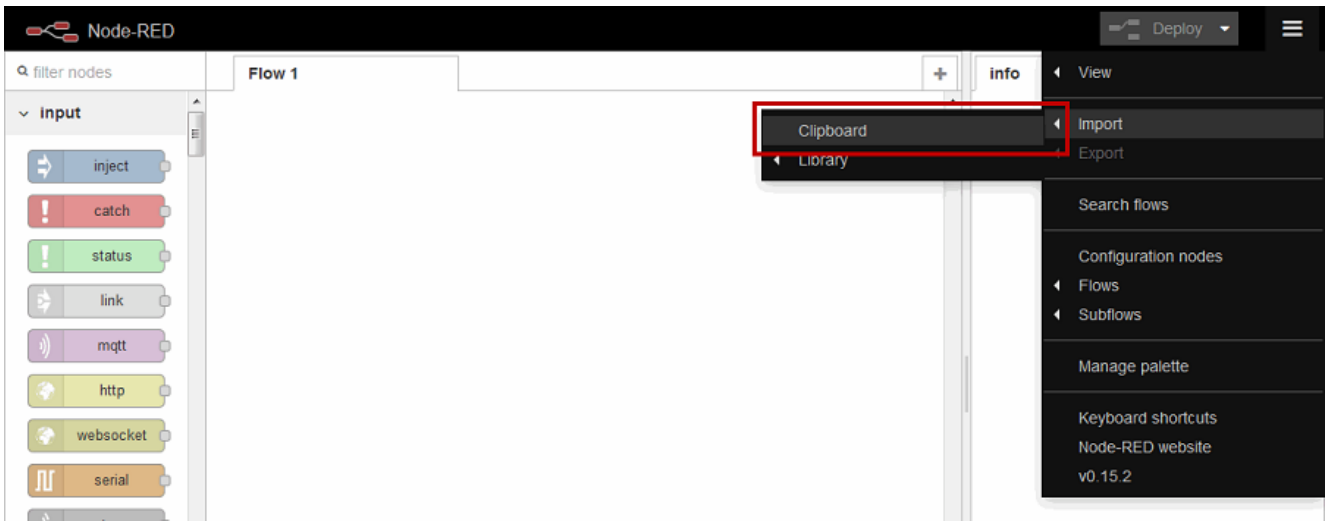
1. In the IBM Cloud dashboard, click **Catalog**.
2. In the **Search** field, enter Node-RED. Click **Node-RED Starter** under **Boilerplates**.
3. Scroll down the **Create a Cloud Foundry App** page and enter your app name and retain the default values for the other entries. The app name you entered will be the default host name to be accessed. You can change your host name if required. Then, click **Create**. Because the name of the application needs to be unique within IBM Cloud environment, you might get an error message. If so, enter a different app name and again click **Create**.
4. After your application is created, your new Node-RED application will get configured and started. Wait for a while or press F5 until you see your application status change to **Running**.

## Creating a sample application with Node-RED flow editor

Node-RED has a tool called Flow Editor using which you can drag nodes from palette to your canvas and connect them together. In this example, we will create a sample application using the Node-RED flow editor.

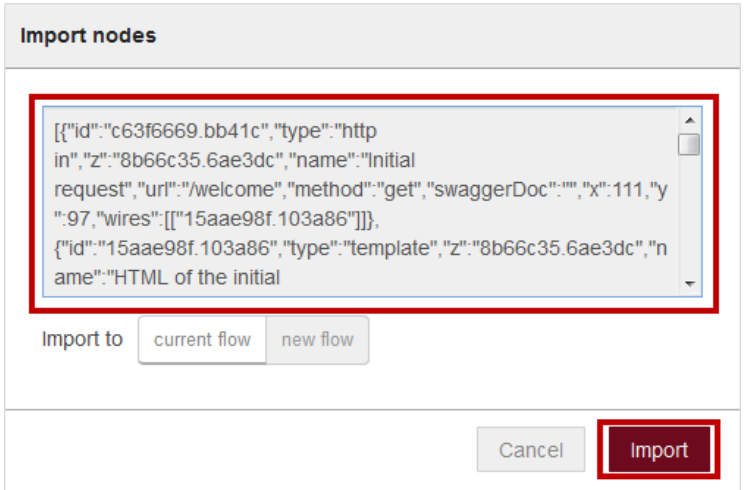
1. Click **Visit App URL** next to status *Running* on the Node-RED application page.
2. Click **Go to your Node-RED flow editor** on the Node-RED on the web page.
3. Flow Editor will start in your web browser. You can import and export the Node-RED flow written in the JSON format. You can download the attached *node-red-dw-sample-yohichin-flow.json* file that contains a sample application flow and import it. At the upper-right corner of the Flow Editor, click **Import** -> **Clipboard** (as shown in Figure 11).

### Figure 11. Import Node-RED flow JSON data – step 1



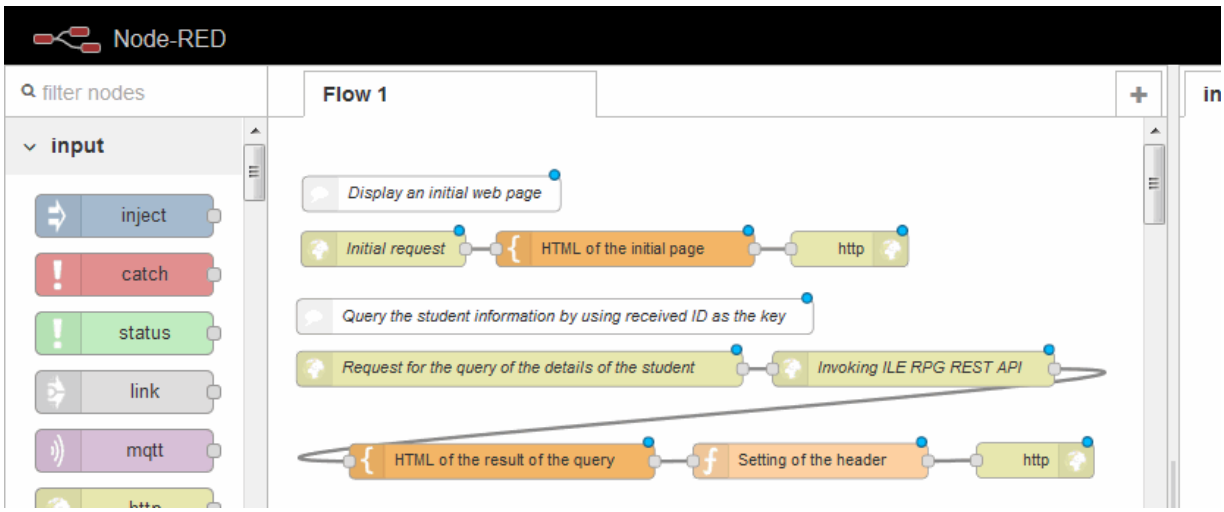

4. Copy all the scripts from the node-red-dw-sample-yohichin-flow.json file to the **Import Nodes** window and click **Import** (as shown in Figure 12).

### Figure 12. Import nodes



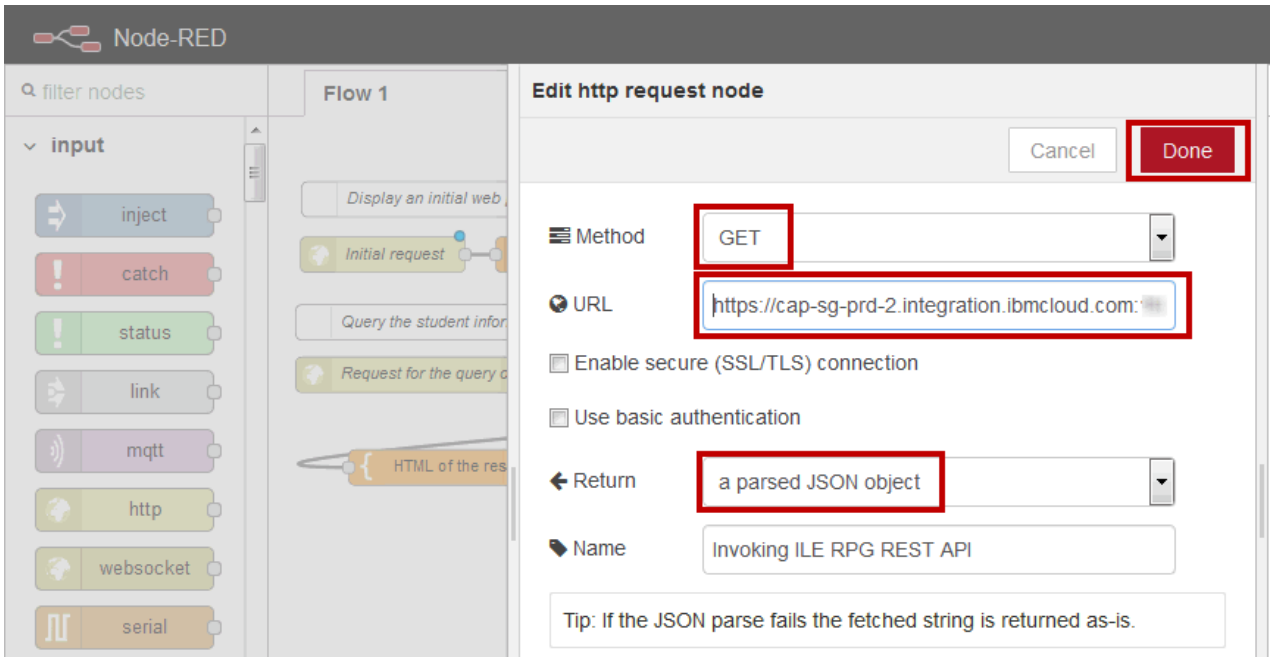
Notice that the imported flow includes two big flows. One is to display an initial web page by accessing https://<your-node-red-app-host-name>.mybluemix.net/welcome. Another is to retrieve student ID from the initial page and pass the data to the REST-based web services on IBM i through Secure Gateway. The REST web service queries the student information by using a received ID as the key and returns the result back to the web page.

### Figure 13. Flow of a sample application



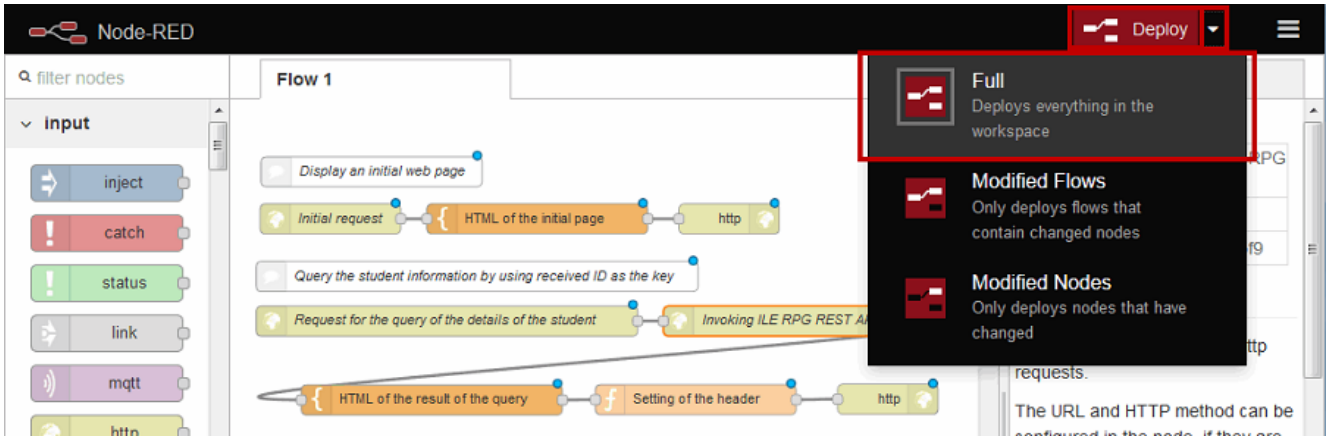

5. Double-click the **Invoking ILE RPG REST API** node, which is the logic to call integrated web services using REST through Secure Gateway (refer to Figure 13).
6. The **Edit http request node** page is displayed. In this scenario, we will query a single student ID by using the *getByID* REST API. So, select **GET** for the **Method** parameter. And for the **URL**, you need to use the host name and port number that you have noted down earlier when setting up Secure Gateway. Because you need to specify the student ID received from the web page, you need to construct your URL as https://<*sg-host-name*>:<*port-number*>/web/services/students/{{payload.studentID}} where <*sg-host-name*> and <*port number*> are the host name and port number for Secure Gateway Destination that you set up before. Lastly, for the **Return** parameter, select a parsed JSON object because the data retrieved from the REST API will be in the JSON format. After setting all the information needed, click **Done** as shown in Figure 14.

### Figure 14. Page for the created Node-RED application




7. To deploy the application to your runtime, click the down arrow next to **Deploy** at the upper-right corner and click **Full**. Then, click **Deploy** to deploy the application (as shown in Figure 15).

### Figure 15. Specify options for deploying the flow




8. When the deployment is complete, the message, **Successfully deployed**, is displayed on top of the page. Also, note that the blue dot that appeared next to each node is turned off.

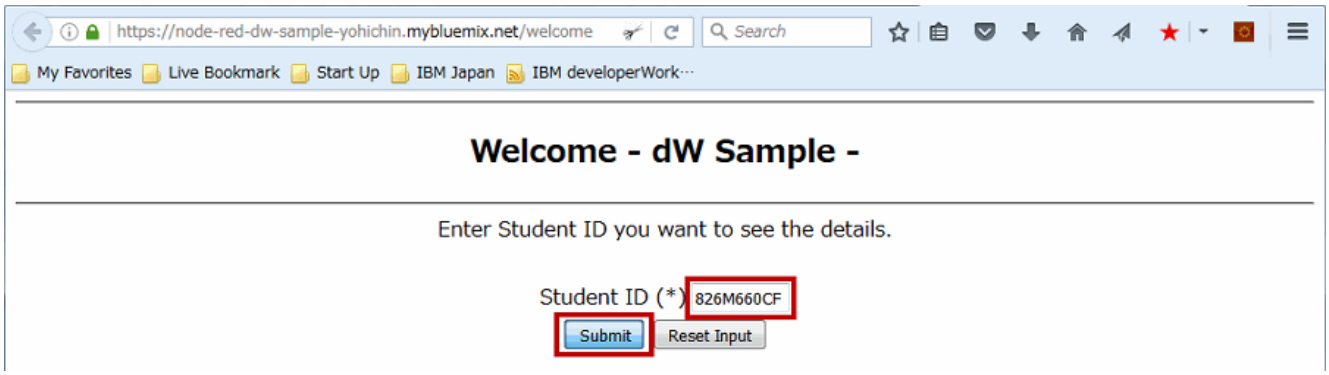
Now you are ready to access your application.

## Running the sample application

You need to perform the following steps to run the sample application:

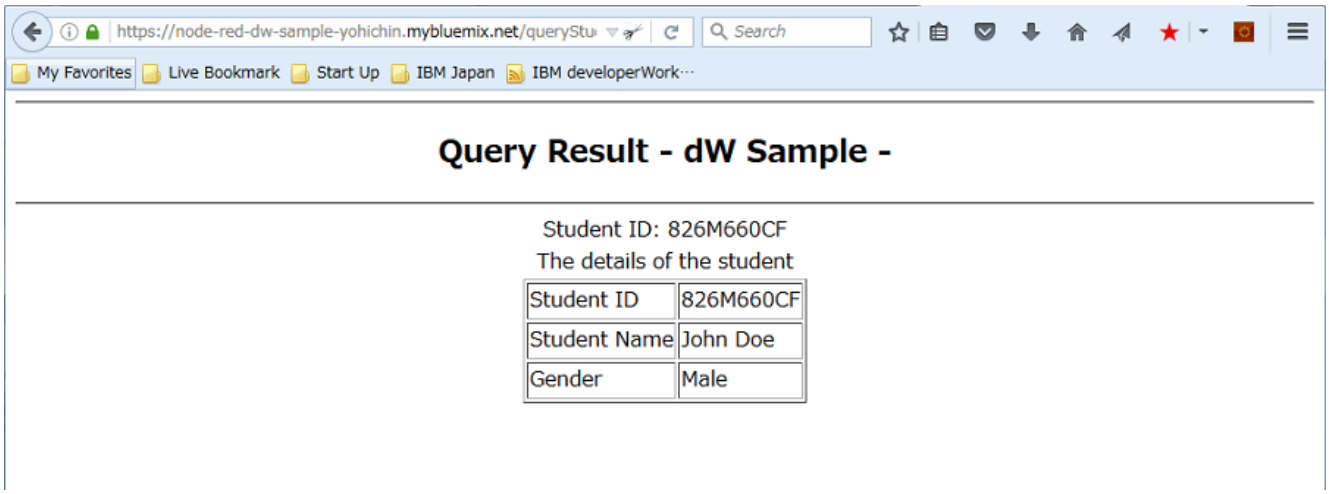
1. From the web browser, access the following and see that the initial page configured in Node-RED is displayed.
   https://<*your-node-red-app-host-name*>.mybluemix.net/welcome
   Replace *<your-node-red-app-host-name>* with your application name or the host name you specified when creating Node-RED.
2. Enter the student ID, for example, **826M660CF** in this scenario, and click **Submit** (as shown in Figure 16). On submission, the student ID that was entered on the initial page is passed to the getByID API configured on the REST-based web services on IBM i running on the on-premises server using Secure Gateway.

**Figure 16. Submit student ID**




3. Notice that the student information is returned for the specified student ID (refer to Figure 17).

**Figure 17. Result page of the sample Node-RED flow**

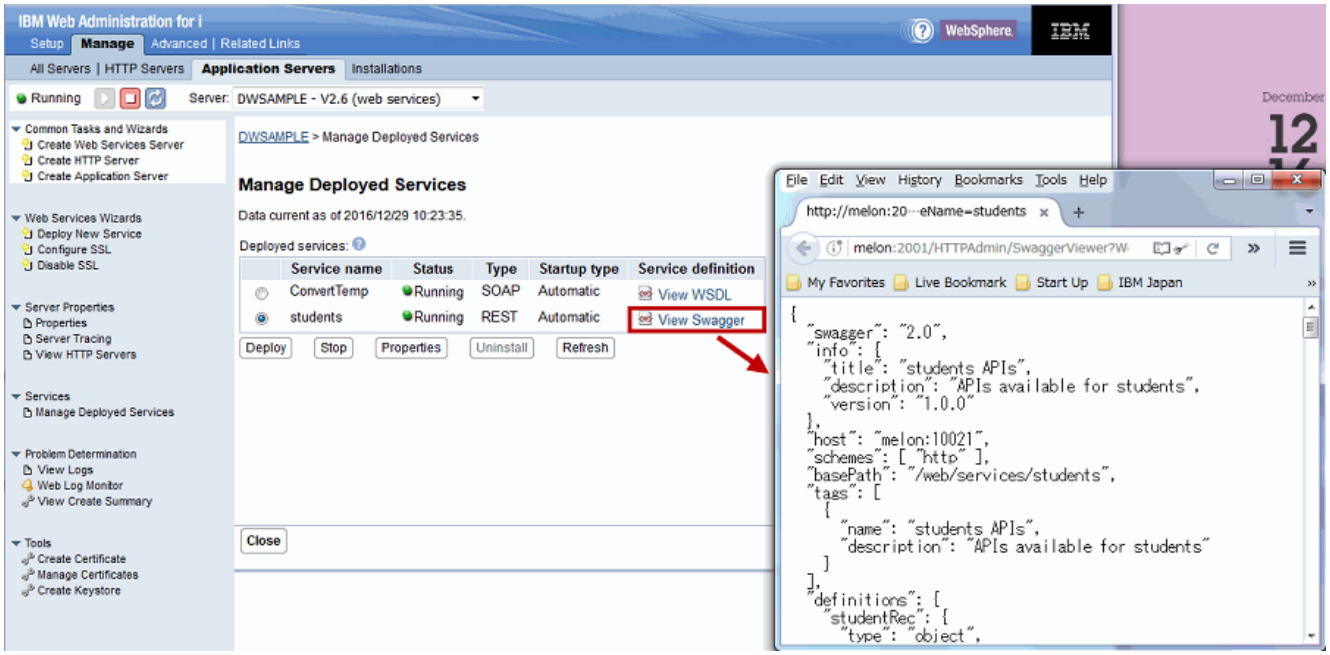



## Considerations and common pit falls

This article explained how easily you can access ILE RPG from an IBM Cloud application by using integrated web services. Node-RED applications on IBM Cloud is of course deployed on the internet world. By using Secure Gateway, you can retrieve mission-critical, on-premises IBM i system data from these applications easily. Against this advantage, you should consider how you can maintain security of your data. In this article, we used Secure Gateway and the HTTPS

protocol to access the ILE RPG REST web service on on-premises environment securely. HTTPS can maintain confidentiality of data in the route of connection. From the security point of view, you should also consider who can use ILE RPG REST web services you deployed to on-premises integrated web services server and implement access control. To implement this consideration to your REST web services, you can use IBM API Connect™. By using API Connect, you can set up security settings such as authentication to your REST web services. Similar to the Node-RED application, the API Connect service is also available on IBM Cloud. You can use API Connect by registering REST APIs of your ILE RPG REST web service. To register APIs to API Connect, we normally use a Swagger file. A Swagger file is a JSON file that defines the configuration of the API. Fortunately, the functionality for generating a Swagger file for ILE RPG REST web services is available by applying HTTP PTF Group Level 44 for i 7.1, Level 18 for i 7.2, and Level 5 for i 7.3 or later. This is a great enhancement in API Connect to use your ILE RPG REST web services because you can import ILE RPG REST web services into API Connect by using auto generated Swagger files.

**Figure 18. Generating Swagger file support on integrated web services server**




## Conclusion

This article explained IBM i and IBM Cloud capability of web or mobile native application deployment. We have shown you how to design process integration with Node-RED and how IBM i RPG can extend its power to the cloud. IBM i and IBM Cloud continues to move forward to this cloud era.

## Related topics

- Building a REST service with integrated web services server for IBM i, Part 3
- How to Create the Local Certificate Authority (CA) Store in DCM

- How to Create the *SYSTEM Store in DCM
- How To Enable an IBM Integrated Web Services (IWS) Server for Secure Socket Layer (SSL) / Transport Layer Security (TLS)

## Downloadable resources

| Description | Name | Size |
| --- | --- | --- |
| node-red-dw-sample-yohichin-flow | node-red-dw-sample-yohichin-flow.zip | 2 KB |