

AIS Demo Example

How to demo the value of IBM Application Integration Suite – setup instructions, demo scenarios and demo script.

Contents

[Set up demo environment 3](#_Toc479261544)

[Install Docker app 3](#_Toc479261545)

[1.1 Setup Docker environment 3](#_Toc479261546)

[Configure Bluemix Secure Gateway 5](#_Toc479261547)

[Start API Connect in Bluemix 6](#_Toc479261548)

[Setup App Connect Professional 7](#_Toc479261549)

[Deploy the IBM Integration Bus BAR file to the Integration Server 8](#_Toc479261550)

[AIS Demo Scenarios 10](#_Toc479261551)

[Scenerio1 – Update Salesforce Account info using a spreadsheet 11](#_Toc479261552)

[Scenerio2 – Create a REST Service in IIB and Push API to API Connect in Bluemix 12](#_Toc479261553)

[Scenario 3 – Mange, monitor and test APIs in API Connect 14](#_Toc479261554)

[Conclusion 17](#_Toc479261555)

# **Set up demo environment**

The demo uses Docker containers (for IIB and Secure Gateway) and App Connect SaaS and API Connect in Bluemix. The demo is loosely based on the redbook – An Architectural and Practical Guide to IBM Hybrid Integration Platform - <http://www.redbooks.ibm.com/abstracts/sg248351.html>. This document describes basic instructions on setting up the environment and demo script, primarily for Macbook but the same can can be setup on Windows systems also.

## Install Docker app

If you do not already have Docker installed on your system, download Docker App from the following link:

For Mac

<https://docs.docker.com/docker-for-mac/install/>

For Windows

<https://docs.docker.com/docker-for-windows/install/>

The demo was tested using Docker version 1.13 and also the latest version 17.03.1-ce, build c6d412e as of April, 2017.

## Setup Docker environment

Create a local directory for the demo artifacts, like for example, mkdir ~/AISDemo; cd ~/AISDemo

Clone the GitHub repository using the command:

git clone https://github.com/sg248351/scenario1

Accept Licensing agreement before starting the containers

export LICENSE=accept

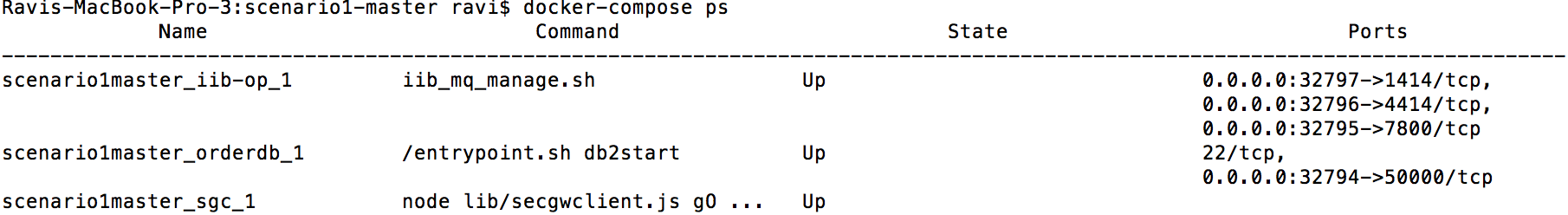
Create and start all containers that are required for this scenario (this will take several minutes)

docker-compose up –d

Verify that the containers have started successfully by running docker-compose ps

docker-compose ps

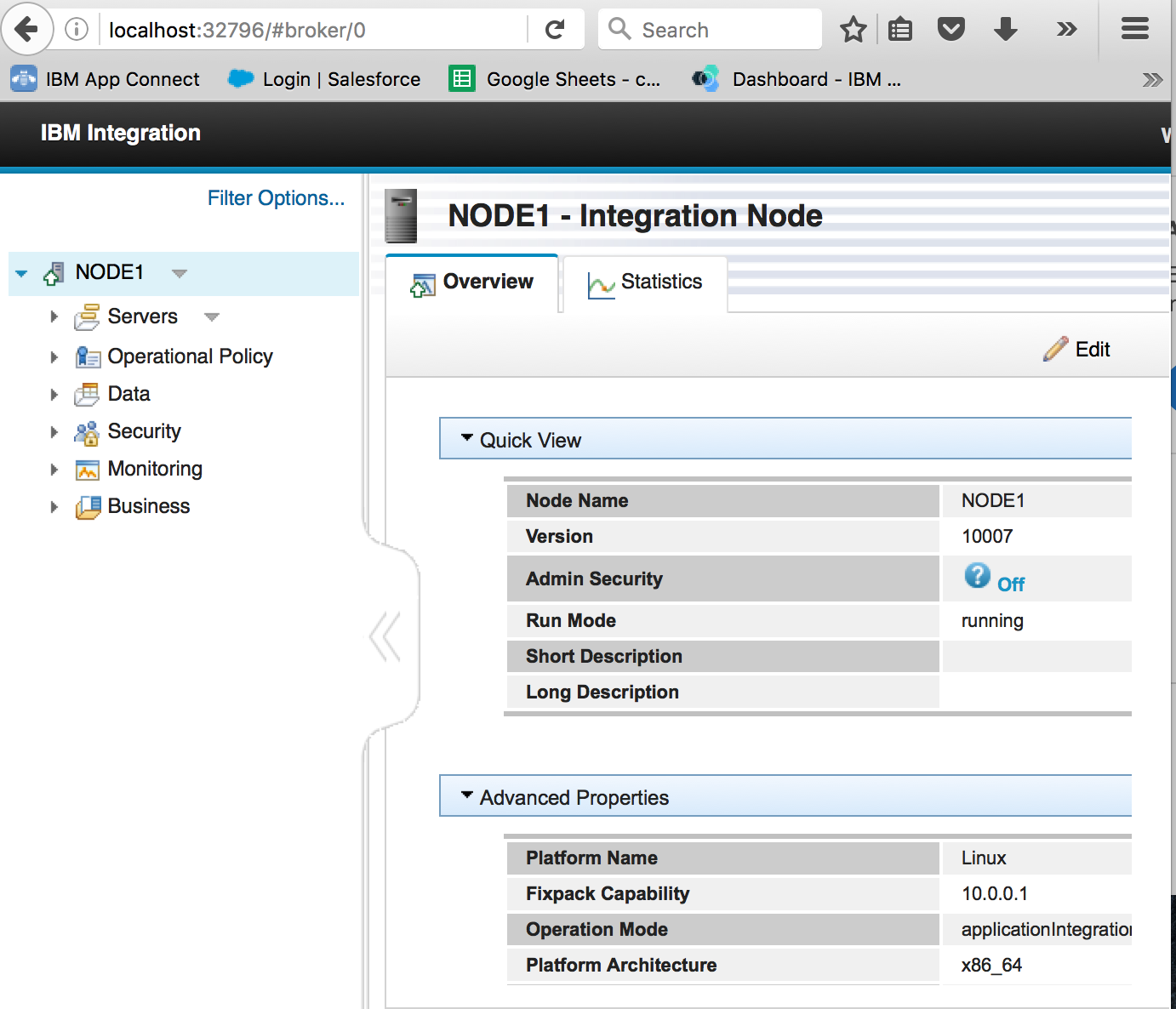
Make sure 4 Docker containers are running and their current state is “up”.



To verify IBM Integration Bus is running, make a note port mapping for 4414 for the Container name “scenario1master\_iib-op\_1” and open the web console on that port. In my case, the port mapping points to 32796.

http://localhost:32796/

The web console results should look like below:

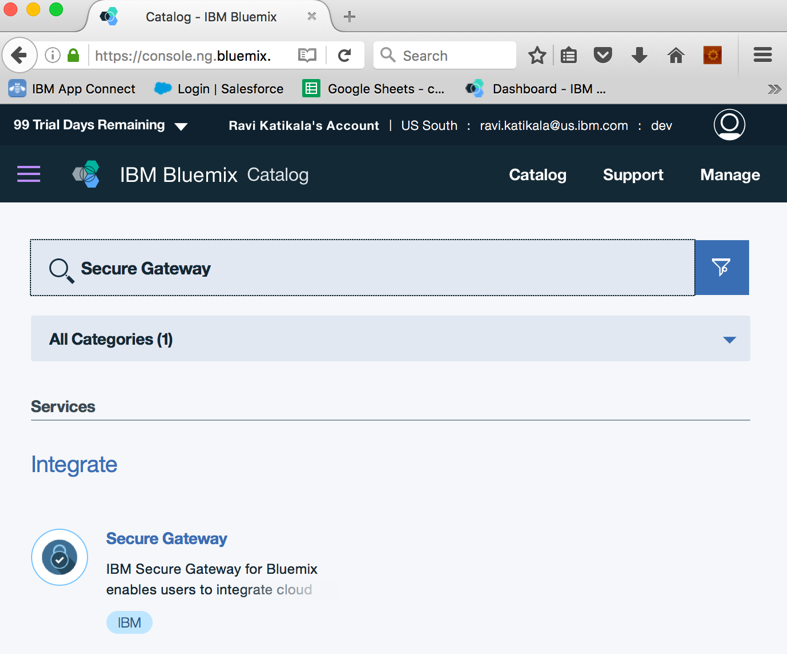


## Configure Bluemix Secure Gateway

Log in to Bluemix using IBM ID and create a secure gateway instance

<https://new-console.ng.bluemix.net/#overview>

Select Integrate then the Plus icon and Secure Gateway service to create a new service.



Add a new gateway, you can name it as AIS Demo or something else.

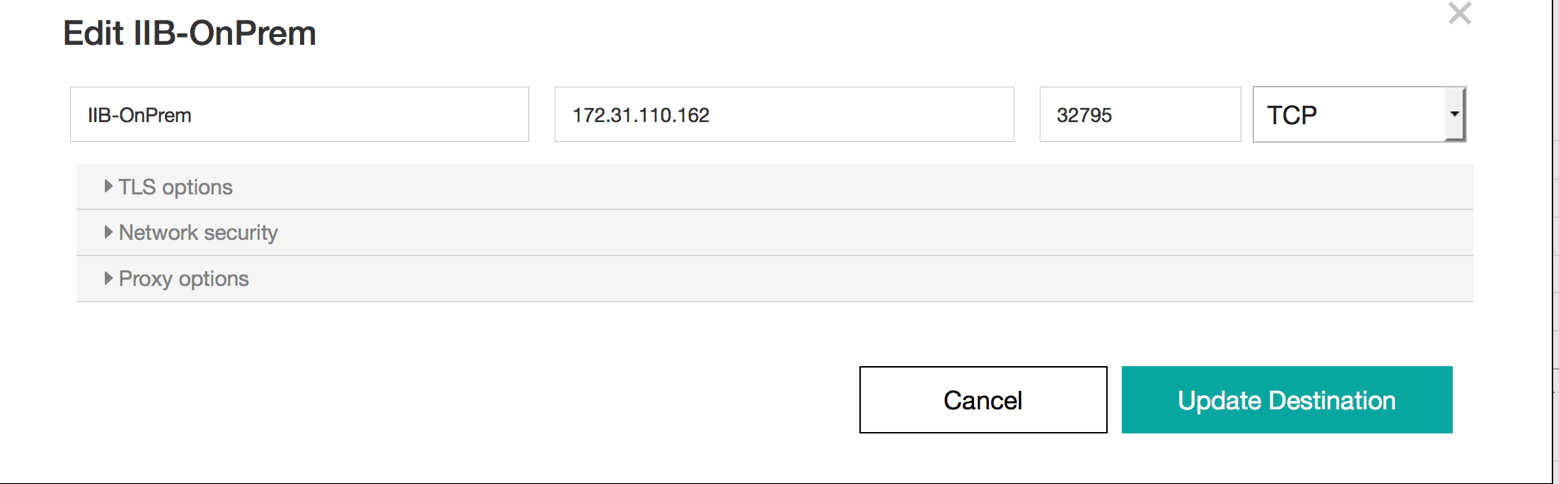
Click the Cogwheel icon in the secure gateway definition to display the settings for this gateway, and make a note of Security Token and Gateway ID. You need this information for the next step:



From the commands line set the environment variables SCI\_GWID and SCI\_SECTOKEN and restart Security Gateway container as follows (use the codes from your container)



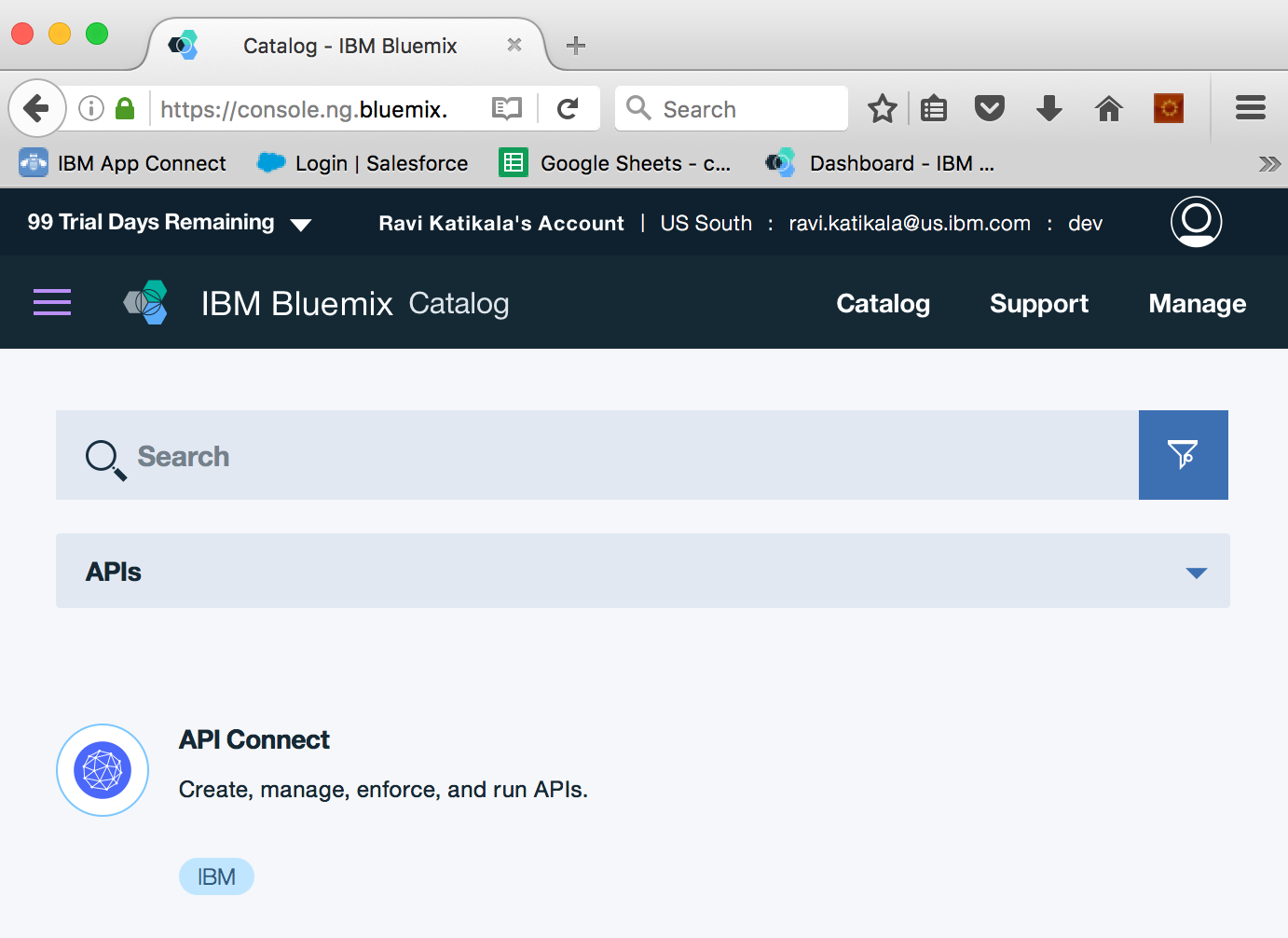
Add Destination for IIB\_On Prem using the IP and port number that corresponds to your local system (use ifconfig or ipconfig to find the IP address for your system, port mapping for container name scenario1master\_iib-op\_1 and port number 7800 is needed for port field in Security Gateway.



## Start API Connect in Bluemix

Log in to Bluemix using IBM ID and create an instance of API Connect Gateway (if you do not already have one). There is nothing more to be done in API Connect at this time as API will be pushed from IIB.

<https://new-console.ng.bluemix.net/#overview>



## Setup App Connect Professional

Create a trial account for App Connect Professional from the following link (if you do not already have one).

<https://appconnect.ibmcloud.com/professional/>

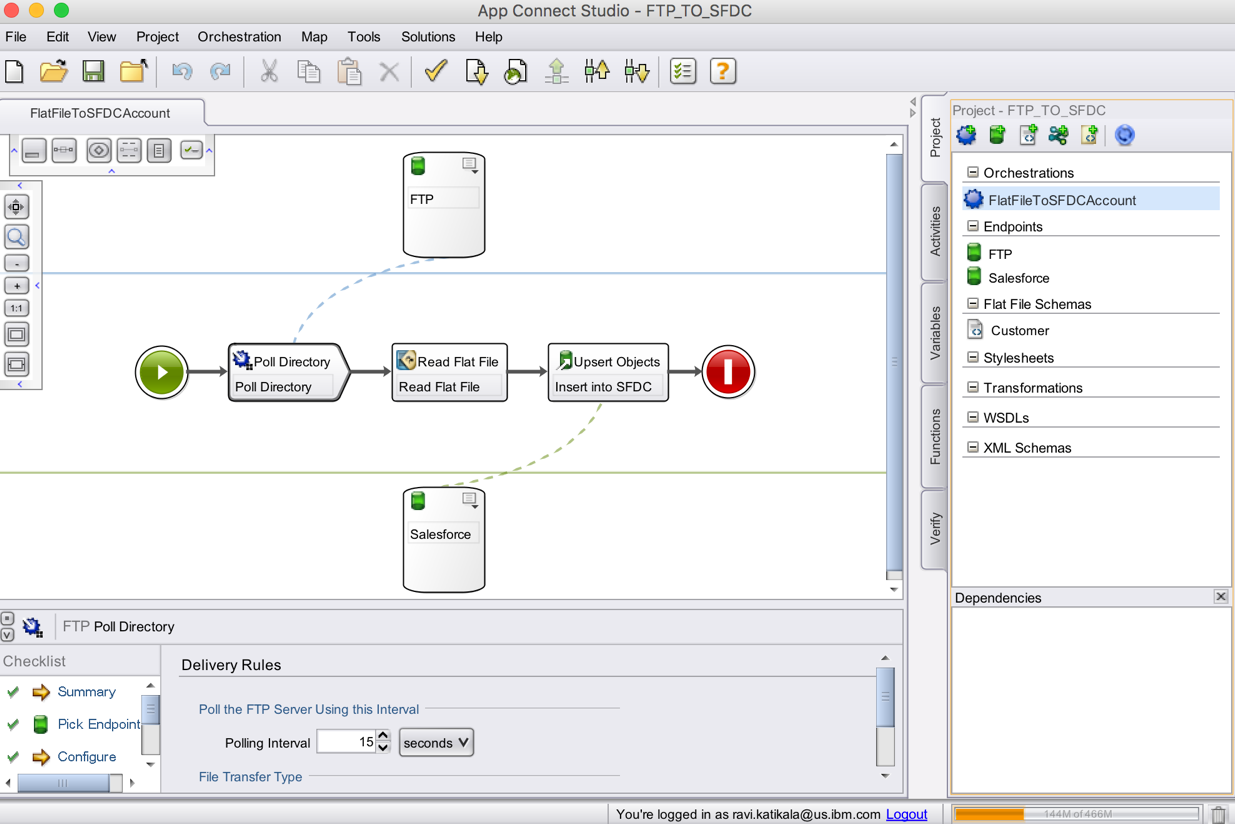
Open AppConnect Studio (either locally or from Web Console by selecting option “Project Designer”).

Clone AppConnect sample from Github

git clone https://github.com/rkatikal/Demos

Open Project Local and specify “FTP\_TO\_SFDC.sp3” file in Demos folder

A simple orchestration as shown below reads a CSV file from an FTP server and updates Salesforce Account object information.



Select FTP and Salesforce end points and update them with your login credentials.

Test connectivity to Saleforce and FTP Server after making the changes.

## Deploy the IBM Integration Bus BAR file to the Integration Server

Look for the SCENARIO1.bar in the project folder.

Before deploying the bar file, set Salesforce.com credentials using the following commands

docker exec -it scenario1\_iib-op\_1 /bin/bash

mqsisetdbparms NODE1 -n salesforce::sf1 -u <UserId> -p <Password> -c

<ClientIdentity> -s <ClientSecret>

The following article explains how to retrieve client id and secret for you login.

<https://help.salesforce.com/articleView?id=000205876&type=1>

Restart integration node

mqsistop NODE1

mqsistart NODE1

4. Deploy the bar file using the web console ( see Step 1.2 on how to bring web console).



The system is now ready for demo!

# **AIS Demo Scenarios**

AIS Demo Scenario consists of 3 demo scenes each of which applies to a diiferent persona in the Integration Community.

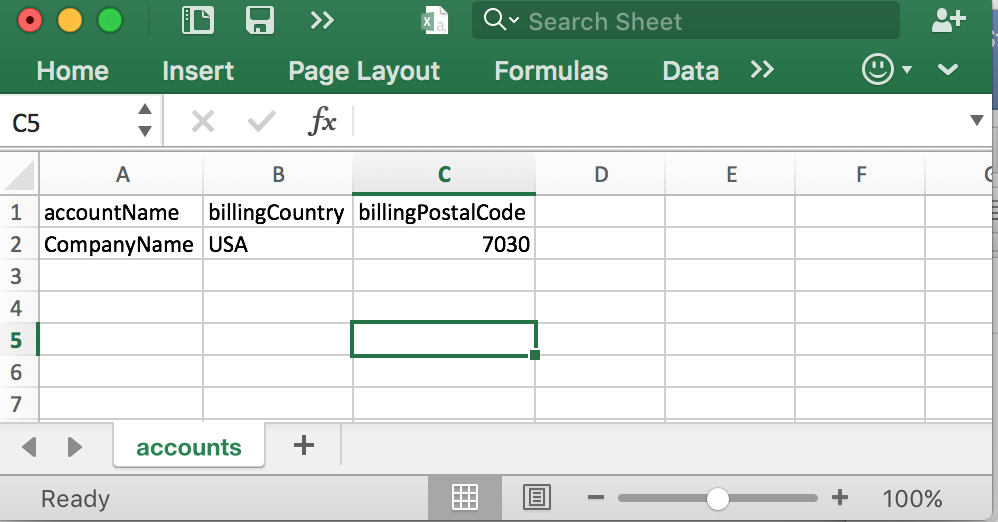


The 3 demo scenes are as follows:

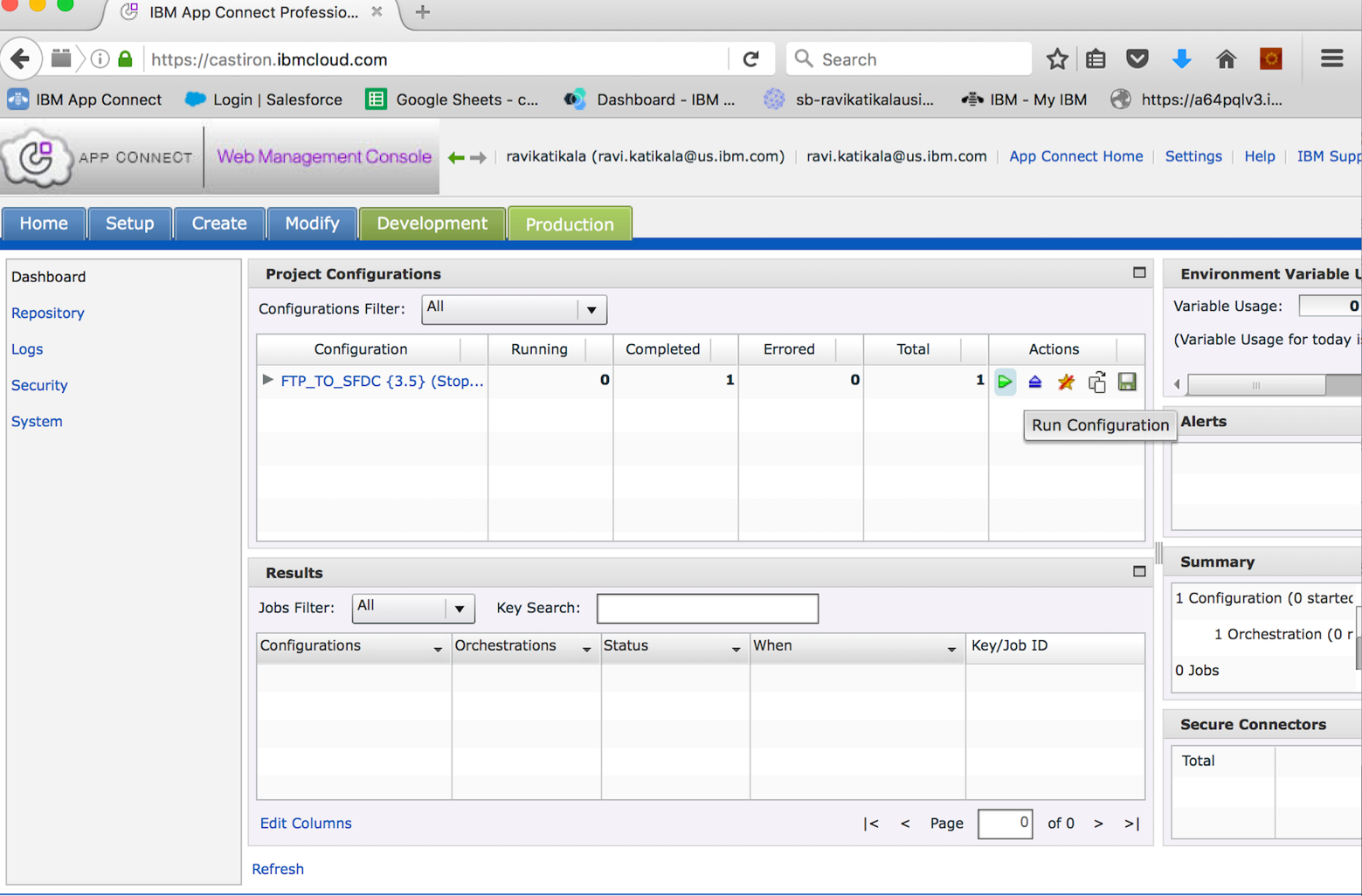
* Scenario-1 Automatically update Salesfoce when Cassie updates Account Information in a spreadsheet
  1. Cassie updates an Excel spreadsheet
  2. Uploads the spreadsheet to an FTP Server
  3. AppConnect Professional orchestration automatically updates Salesforce.com records
* Scenario-2 Create an Order Service by combining data from multiple Systems of Records (DB2 and Salesforce)
  1. Isaac creates a REST API in IBM Integration Bus
  2. REST API implementation includes a message flow optimized for backend processing
  3. Publishes API to APIConnect in Bluemix with a click of a button
* Scenario-3 Manage, monitor, secure and productize APIs
  1. Andre defines security, governance, and access levels for APIs
  2. Creates a Product for Order processing
  3. Makes the Product available in Developer Portal

## Scenerio1 – Update Salesforce Account info using a spreadsheet

Create a file called accounts.csv with the following contents (You my change the data values fro company name, billing country and postal code).



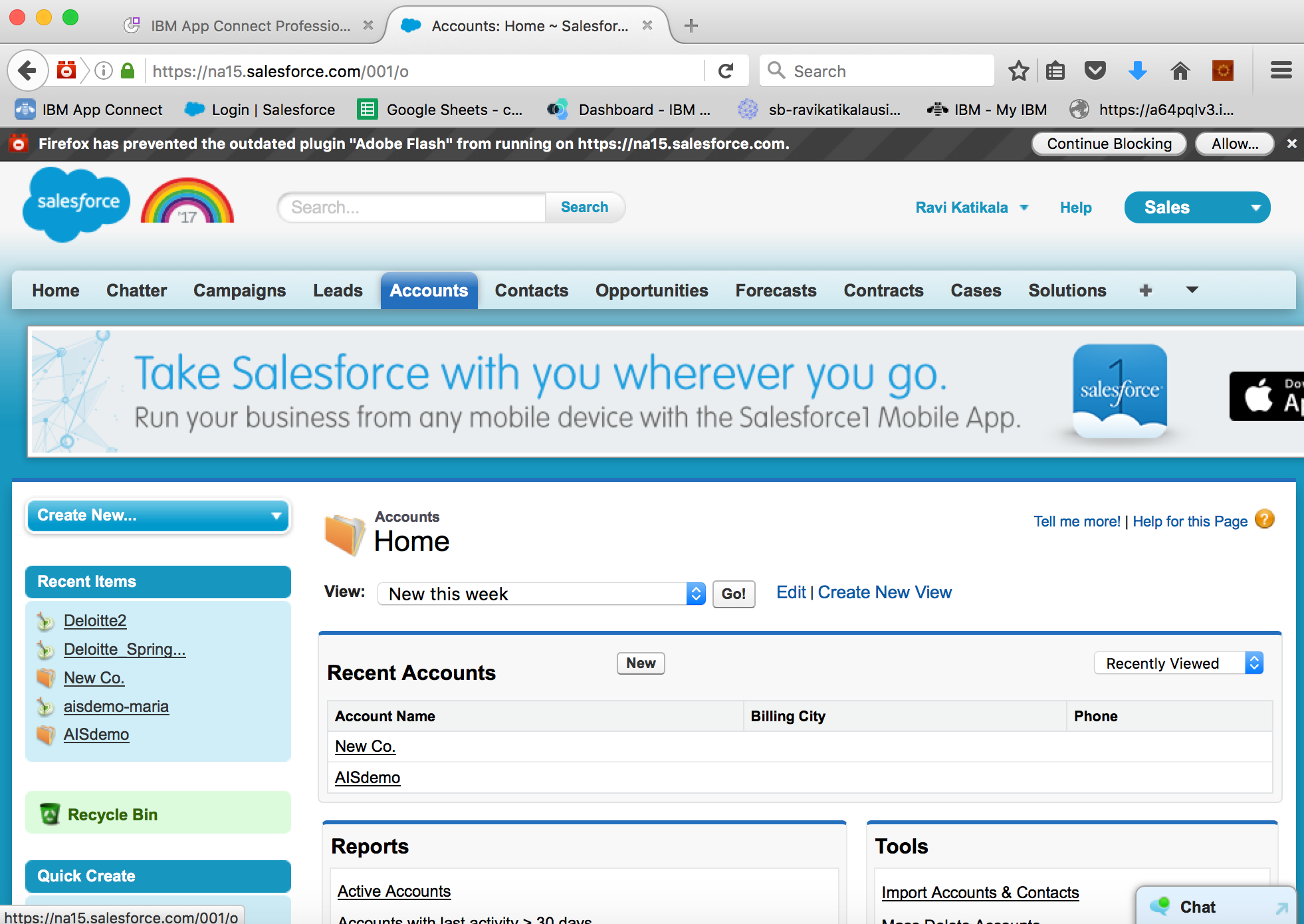
Make sure the orchestration is deployed in App Connect Professional and it is running.



Upload accounts.csv file to the FTP Server “Accounts” folder.

Depending on the Directory poll interval (typically set to 15 secs) a new account will be created in Salesforce.com.

Log in to Salesforce.com and show that a new account is created by the company name “New Co” or whatever name you gave in accounts.csv file.



This completes the demo scenarios1. The contents created in Salesforce.com will be later used in other scenarios.

## Scenerio2 – Create a REST Service in IIB and Push API to API Connect in Bluemix

To demo creation of REST Service and to walk through flow details you will need IIB Integration toolkit that is included as part of IBM Integration Bus for Developers. It can be downloaded from - <https://developer.ibm.com/integration/docs/ibm-integration-bus/get-started/get-started-with-ibm-integration-bus-for-developers/>

Load the project files from iib\_scenario1.zip

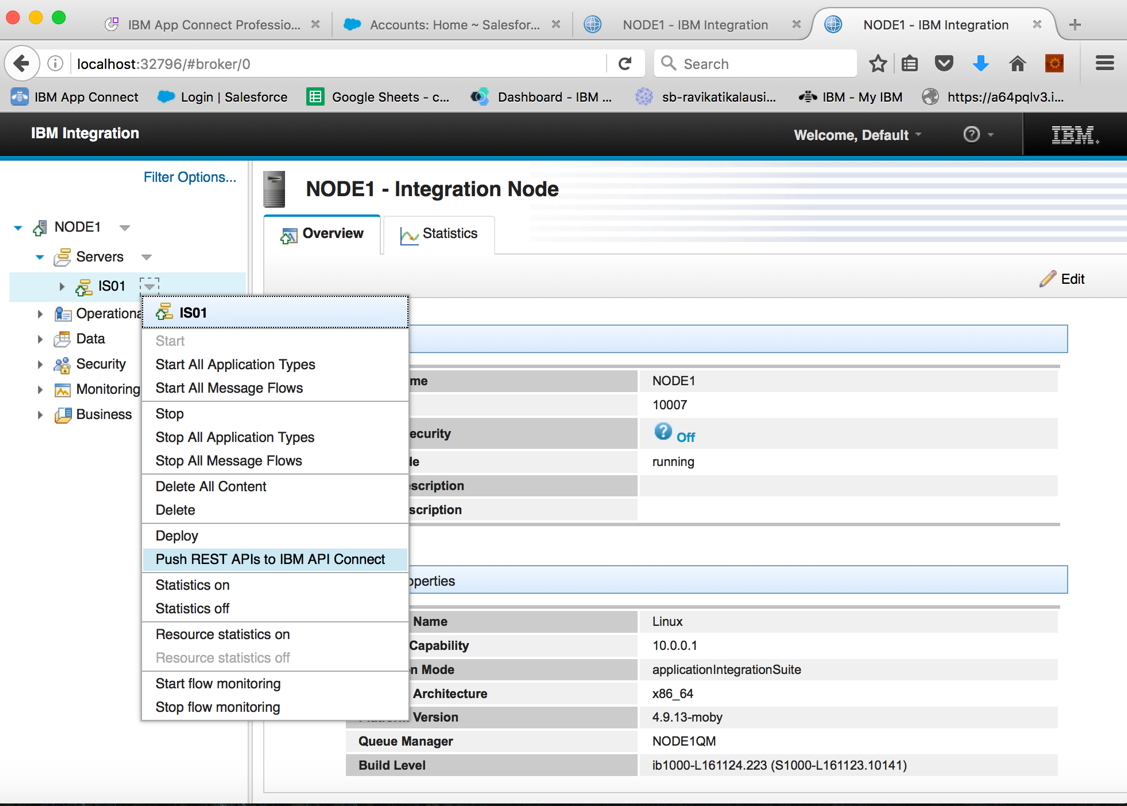
Open up Orders REST service project and walk through the steps

The flow of particular interest is the following:



This flow demonstrates aggregation of content from Salesforce and a Caralog Service (Database).

Demonstrate how to publish REST API to API Connect from toolkit and also IIB web console.

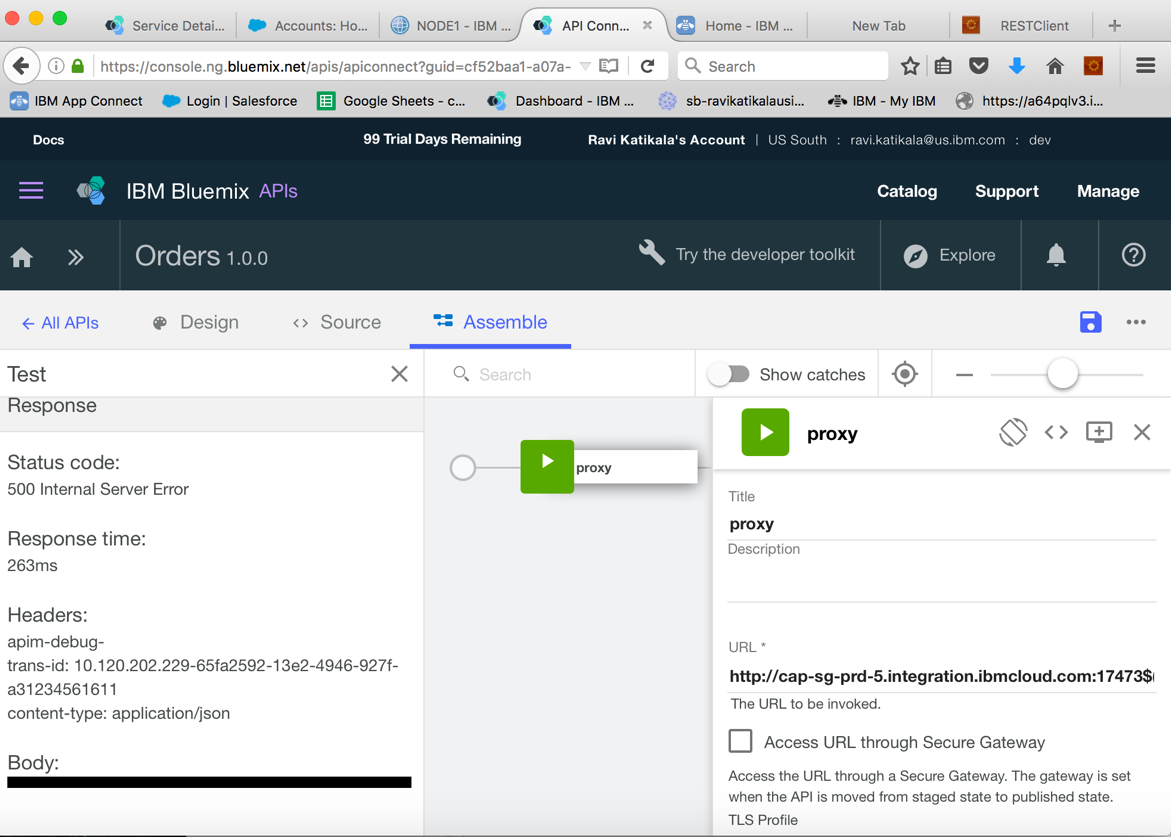


Log in to API Connect on Bluemix and show that a new service “Orders” is created and ready to be deployed.

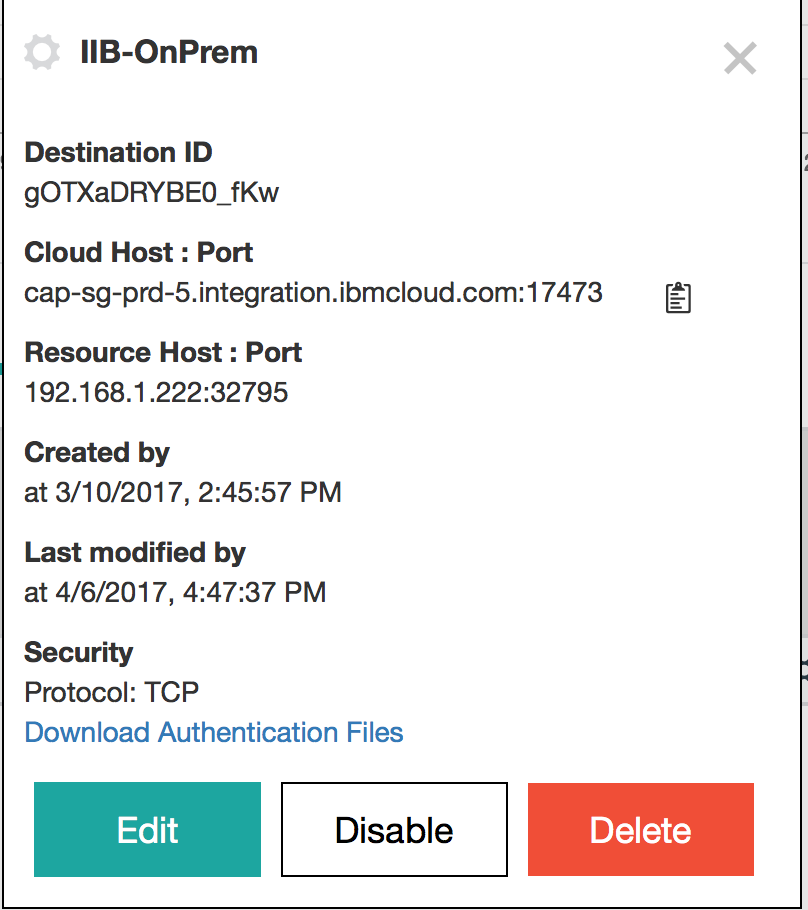
## Scenario 3 – Mange, monitor and test APIs in API Connect

Walk through API Connect features including the concept of Products, Rate limiting, subscription, life cycle management, developer portal.

The only configuration change needed to “Orders” that is published from IIB is to correct the service proxy end point. This should point to Service gateway Destination URI instead of IIB On-prem URI.

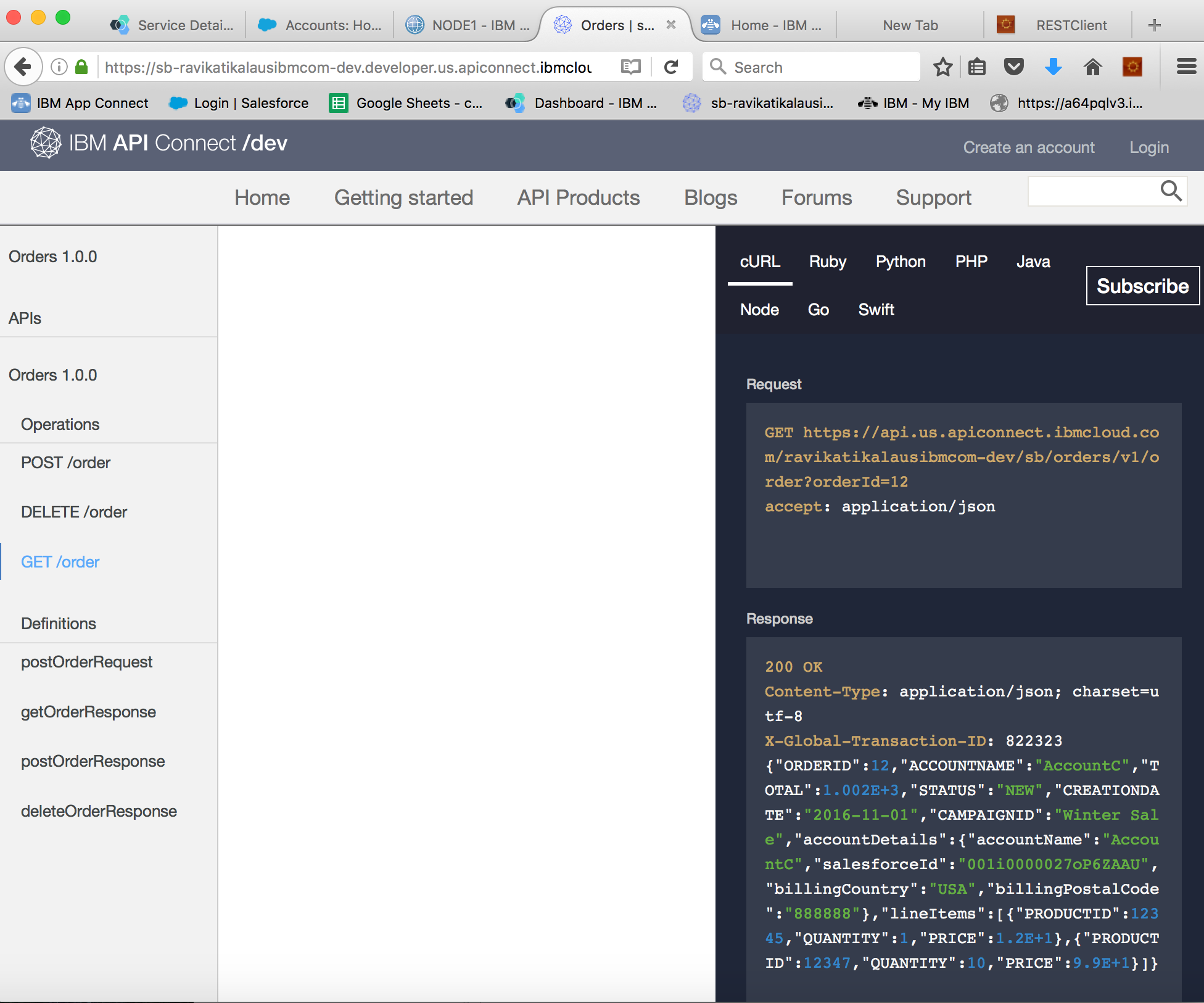


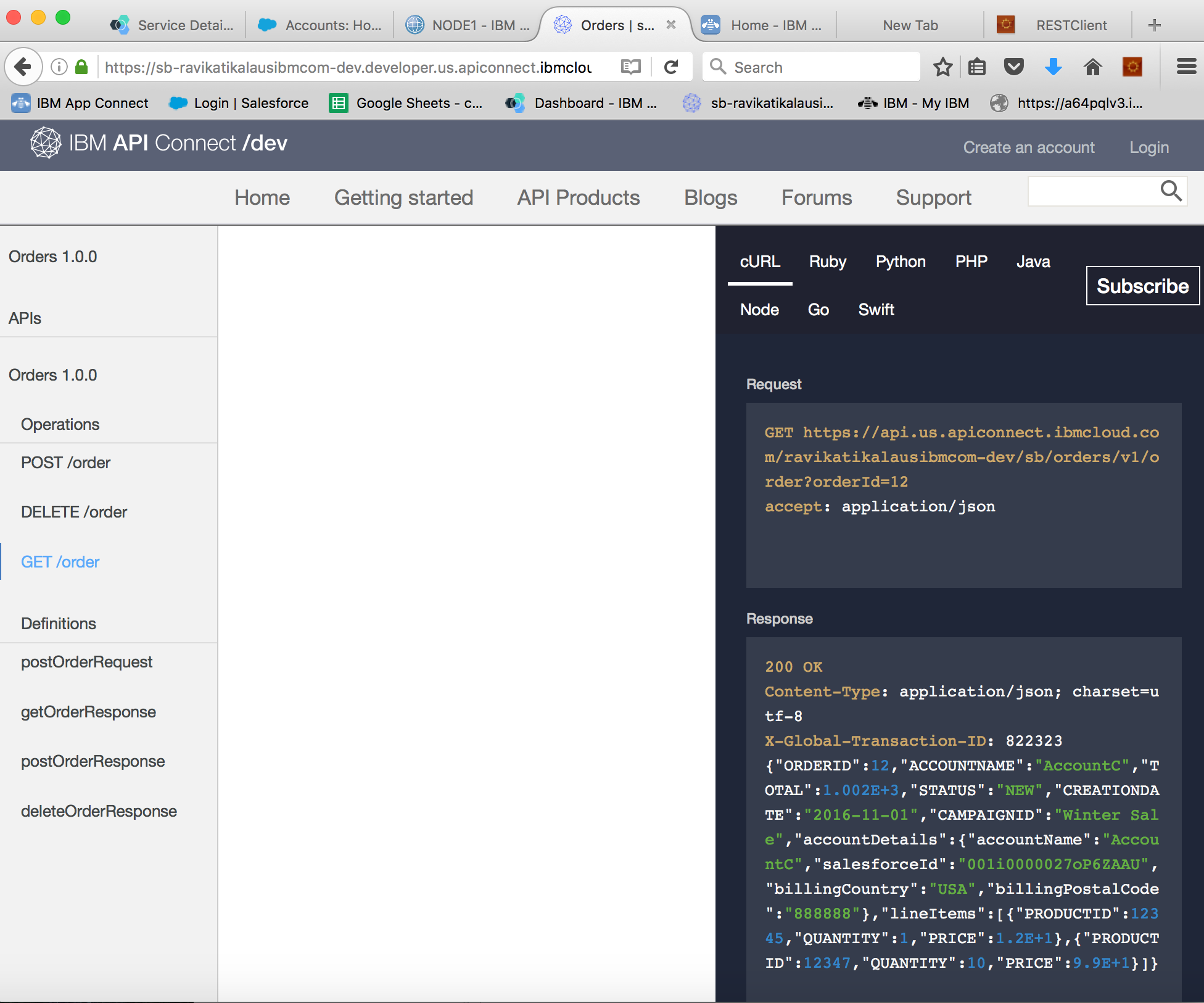
Copy cloud the host name and port number from the Gateway destination properties:



Walk through Developer Portal and the subscription process. Test Orders service (POST orders and GET orders) using Developer Portal.

The results from GET Orders will show that REST API in API Connect displays aggregated content from Salesforce and CatalogService previously defined in IIB and App Connect.





## Conclusion

IBM Application Integration Suite addresses the needs of Advanced Integration Specialist, API Developers and Citizen Integrators. The product components are integrated and work seamlessly.