Application of Microservice Architecture to B2B Processes

(IBM Watson Customer Engagement)

by

Arpit Jain

(2015047)

Supervisor(s):

External
Mr. Atul A. Gohad
(IBM ISL, Bangalore)

Internal
Dr. Aparajita Ojha
(PDPM IIITDM Jabalpur)

Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR

(15st June 2018 – 29th June 2018)

Introduction

The International Business Machines Corporation (IBM) is an American multinational technology company headquartered in Armonk, New York, United States, with operations in over 170 countries. IBM manufactures and markets computer hardware, middleware and software, and provides hosting and consulting services in areas ranging from mainframe computers to nanotechnology.

IBM aims to bring Businesses closer and smarter than ever with the help of their state of the art enterprise software product called B2B Sterling Integrator. IBM B2B Integrator helps companies integrate complex B2B (Business to Business) / EDI (Electronic Data Exchange) processes with their partner communities. It provides a single, flexible B2B platform that supports most communication protocols, helps secure your B2B network and data and achieve high availability operations. The offering enables companies to reduce costs by consolidating on a single B2B platform and helps automate B2B processes across enterprises while providing governance and visibility over those processes.

IBM aims to transform the B2B Sterling product into Microservice architecture. The application is extremely large and monolithic. The making of the product is in process since the last 8 years and is still growing. Primary aim of my entire Six-month PBI period is to convert 2-3 components of B2B Sterling Integrator into microservices and connect with the main application.

Microservice Architecture

Microservices are an Engineering Approach focused on decomposing applications into single function modules and well-defined interfaces which are independently deployed and operated by small teams who own the entire lifecycle of service.

Microservices accelerate delivery by minimizing communication and coordination between people while reducing the scope of change. Microservices can be developed in any language and they communicate with each other using Language Neutral APIs Like REST.

Microservice Architecture is a special design pattern of Service-oriented Architecture(SOA). In this type of service architecture, all the processes will communicate with each other with the smallest granularity to implement a big system or service. It deals with system of services that communicate over a network.

Report on the Present Investigation and Progress

(Progress during this 15-days Period)

Creation of Demo Microservices application and its deployment on cloud

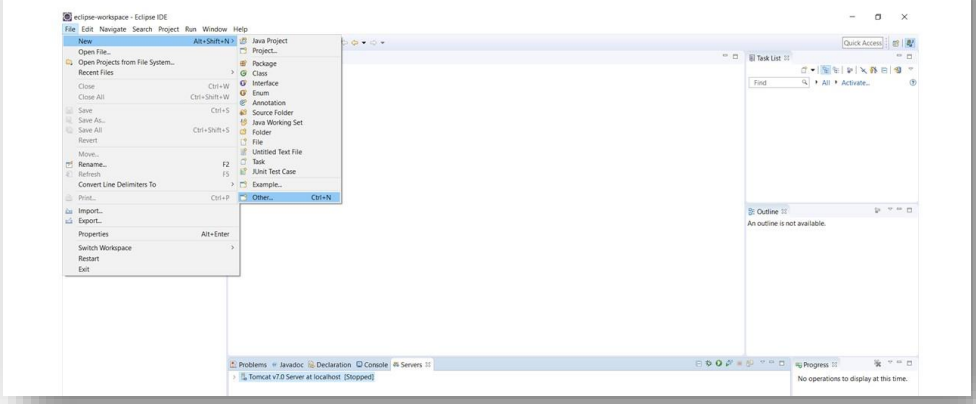
To demonstrate the usage of Microservices architecture for better application design, I implemented a two-microservice application. One microservice acted as a Backend API Fibonacci microservice which was called by another microservice which acted as an UI.

- **The UI Microservice [Code Link]**

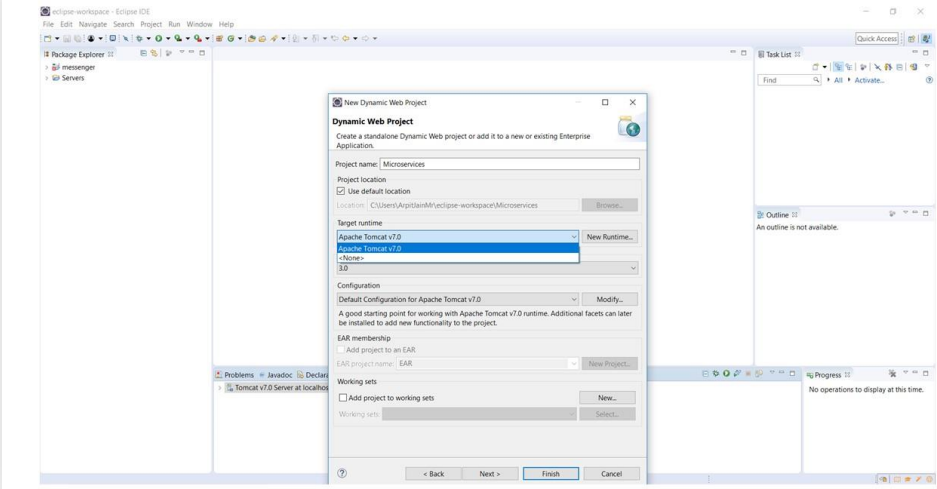
This Microservice is based on Java Liberty framework. This is just an User interface to the Second Backend Fibonacci service. The User can supply two parameters – Duration and Number of Iterations to the second service. Depending upon these two parameters, the second microservice API returns a JSON object.

Following is the process of creation of this Microservice-

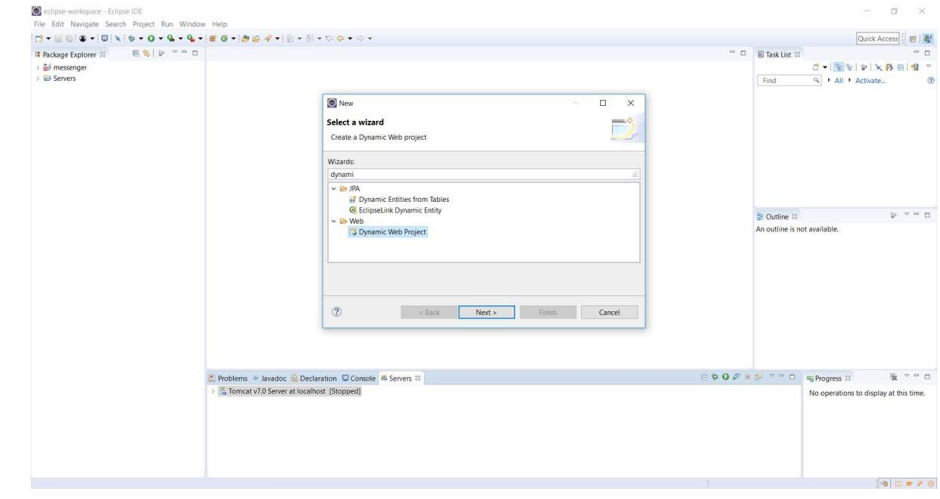
Step 1 : Open an eclipse EE workspace
Step 2: Create a new Dynamic web project



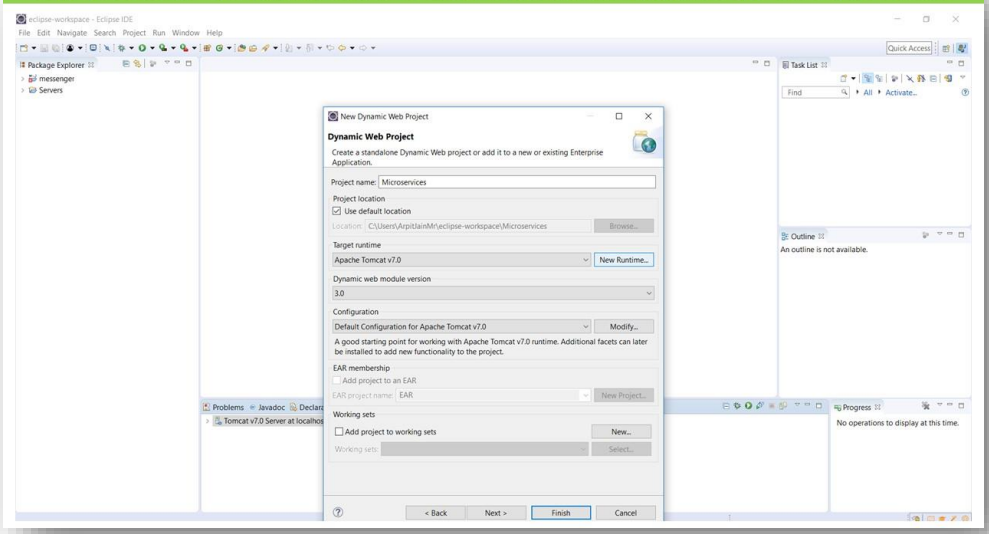
Step 3: Search IBM Bluemix in target runtime



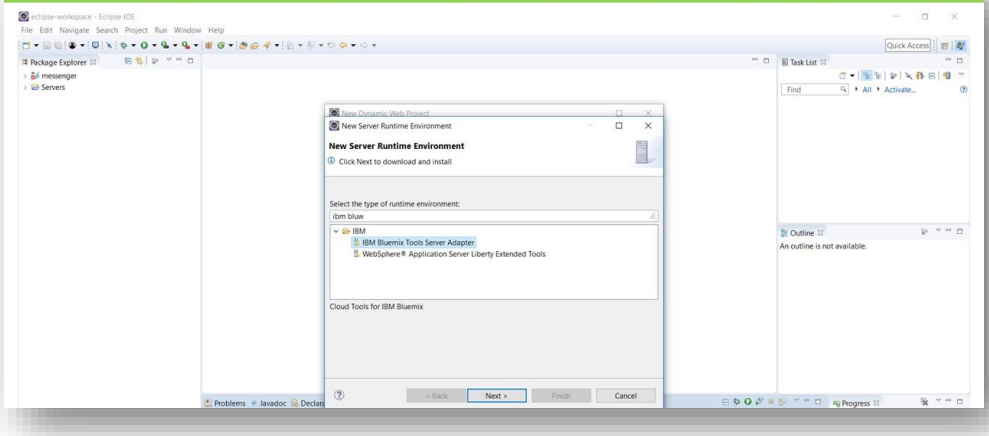
Step 4: Select Dynamic web project and hit next.



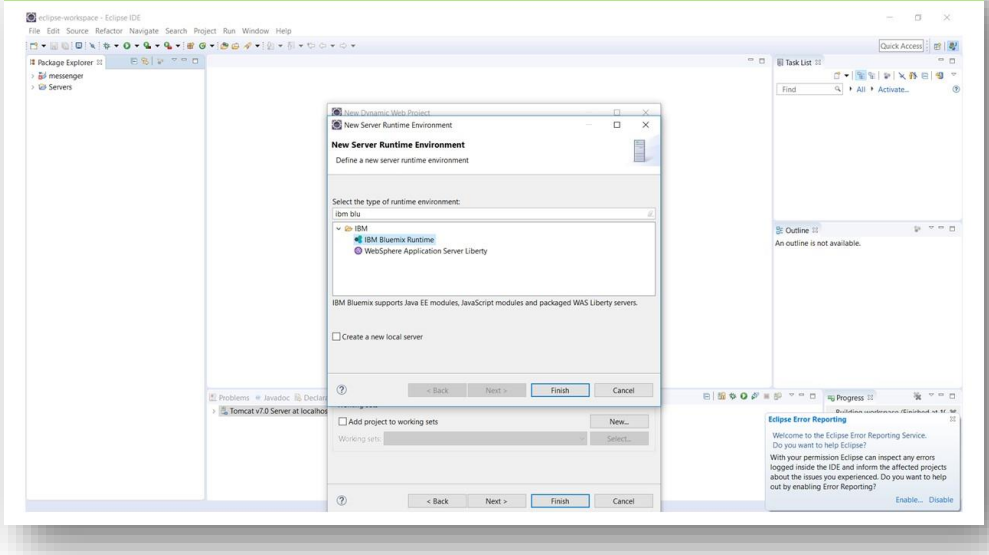
Step 5: Click on New Runtime button to install IBM Bluemix runtime if you don't have Bluemix



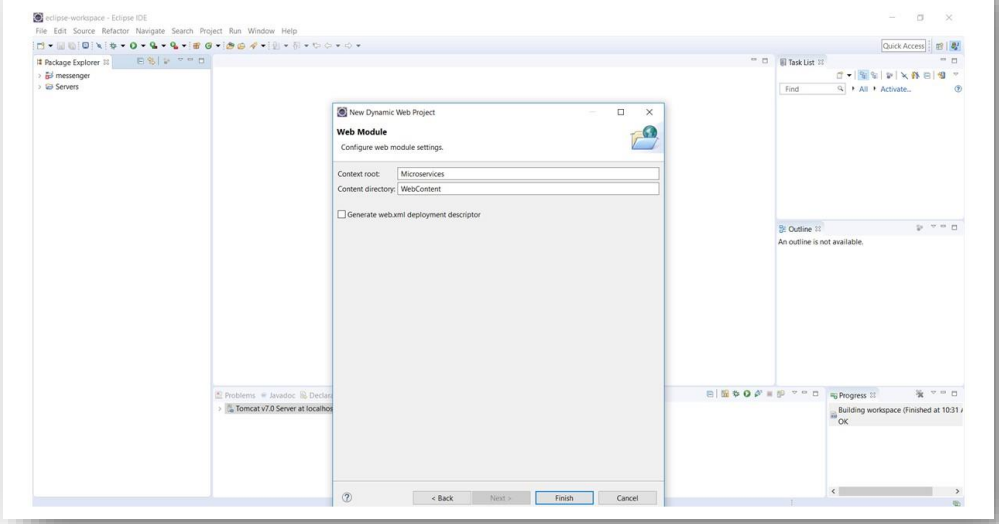
Step 6: Search IBM Bluemix runtime and select IBM Bluemix tool adapter



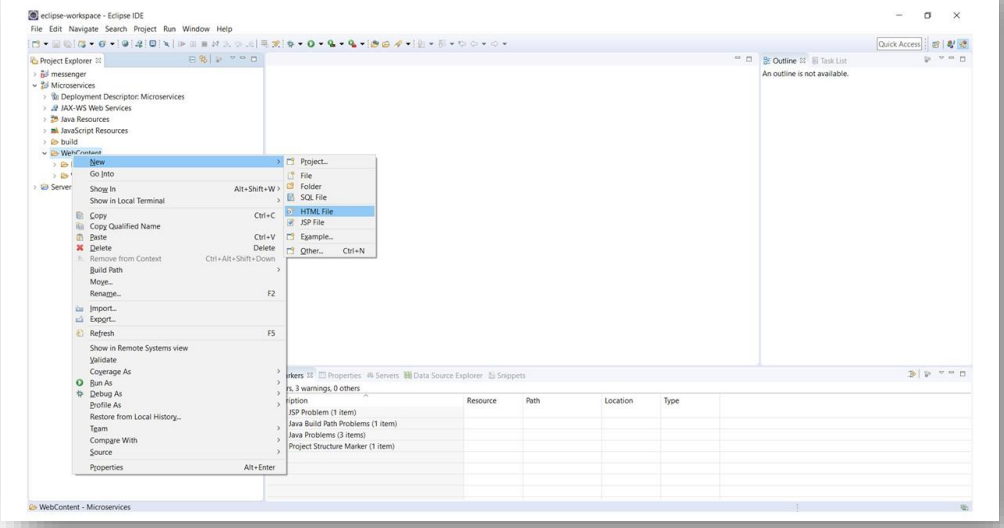
Step 7: Select IBM Bluemix Runtime and hit Finish.



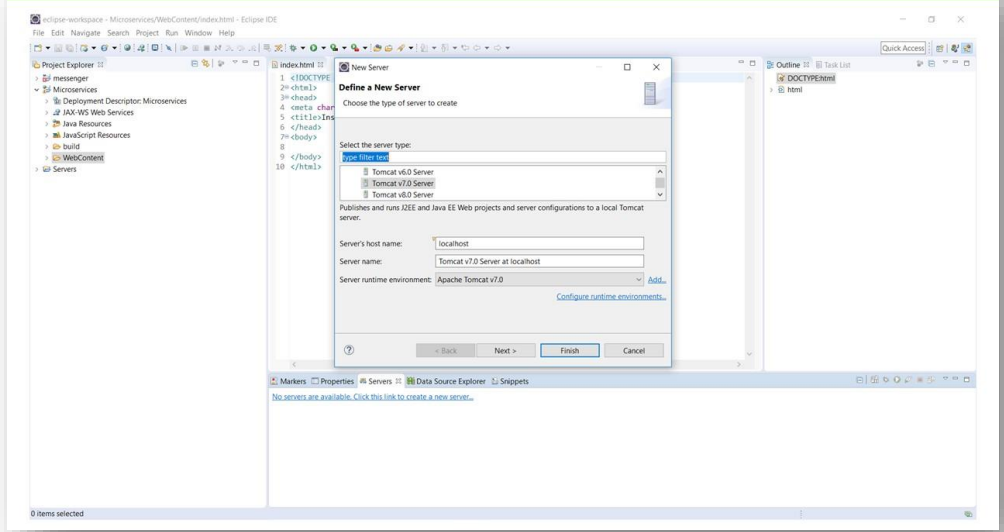
Step 8: Select the check box.



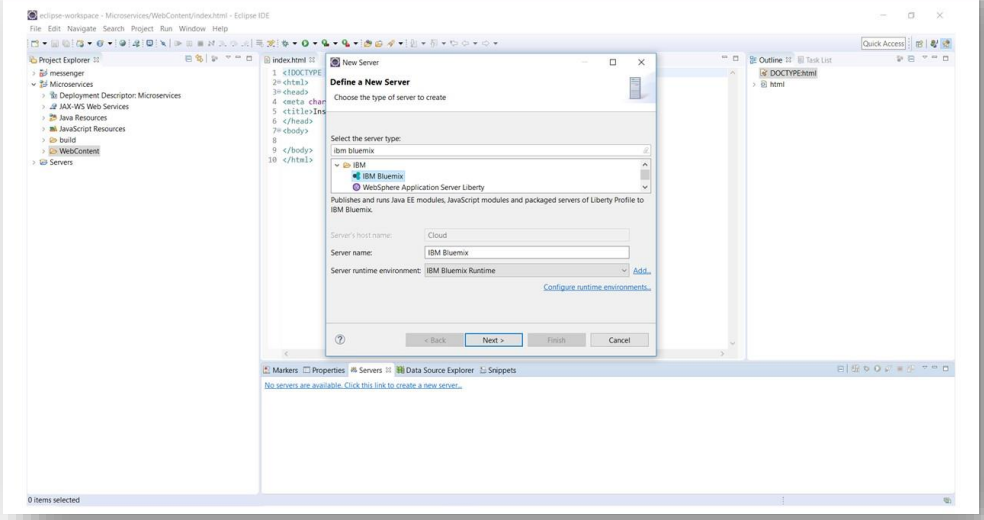
Step 9: Create a new HTML file in Web Content folder



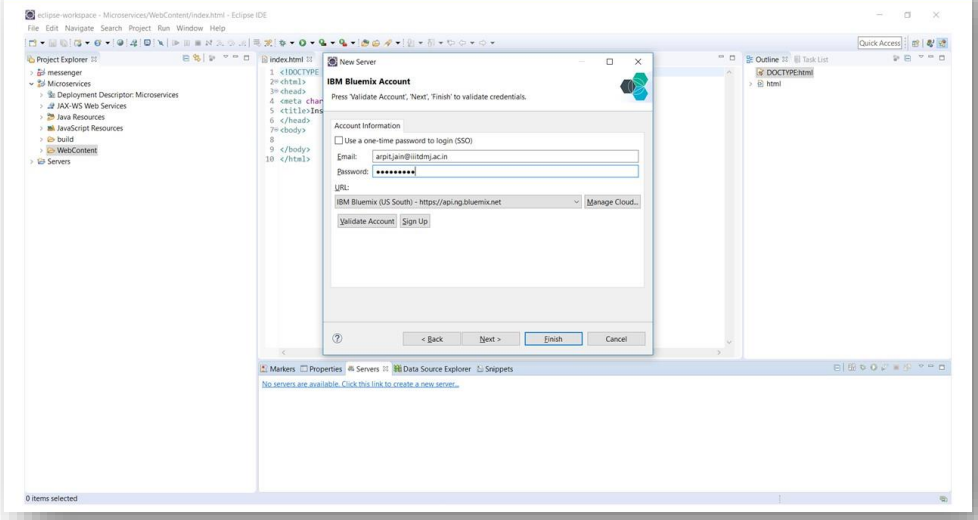
Step 10: Create a new server i.e. IBM Bluemix



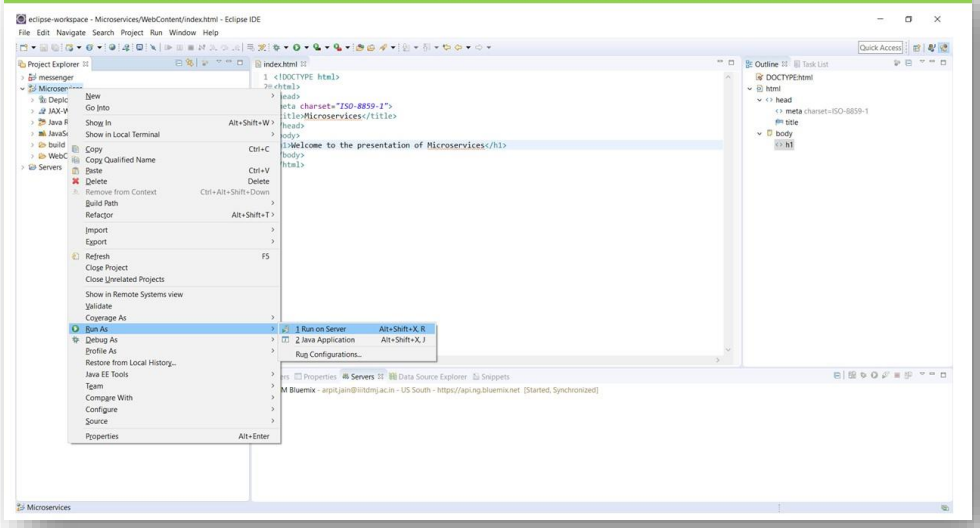
Step 11: Select IBM Bluemix as server.



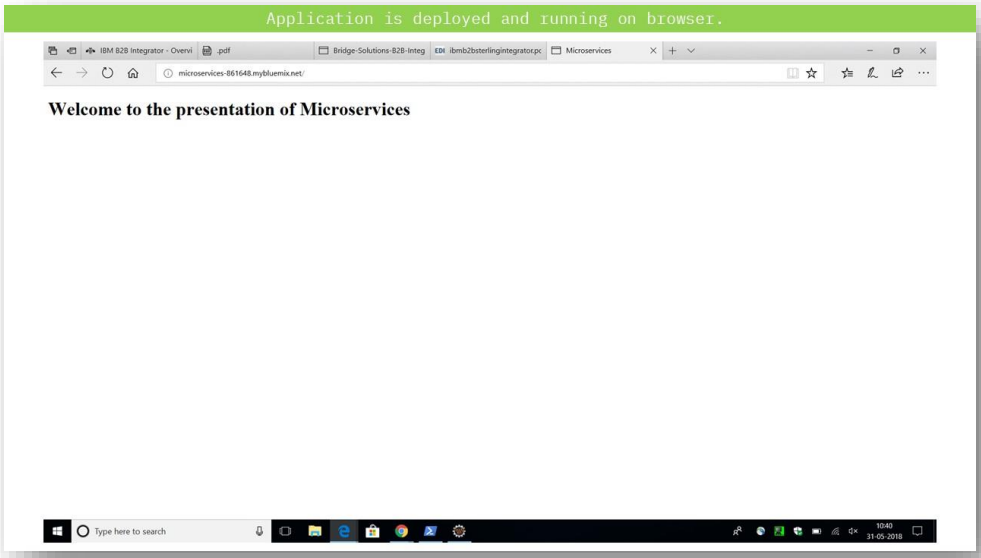
Step 12: Enter IBM Bluemix Account credentials.



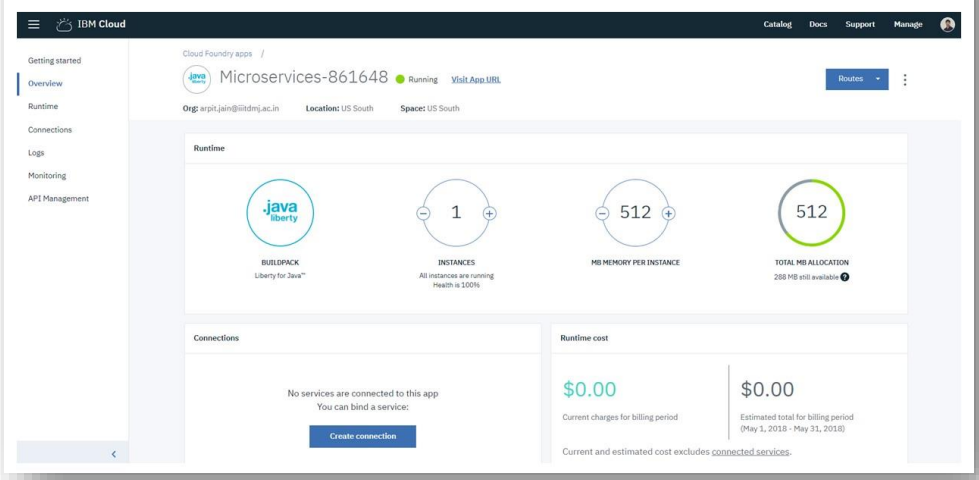
Step 13: Right click on application them select Run as- Run on server.



Application is deployed and running on browser.



Application is running on the IBM Bluemix server.

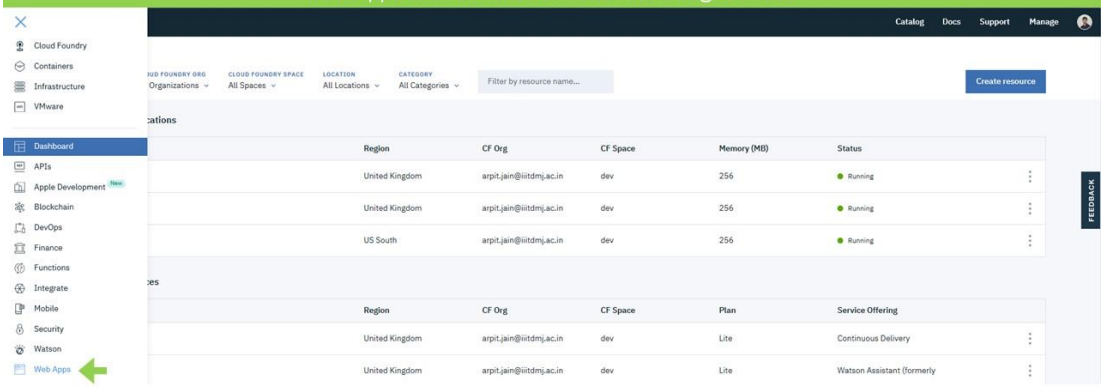


- **The Backend Fibonacci Microservice [Code Link]**

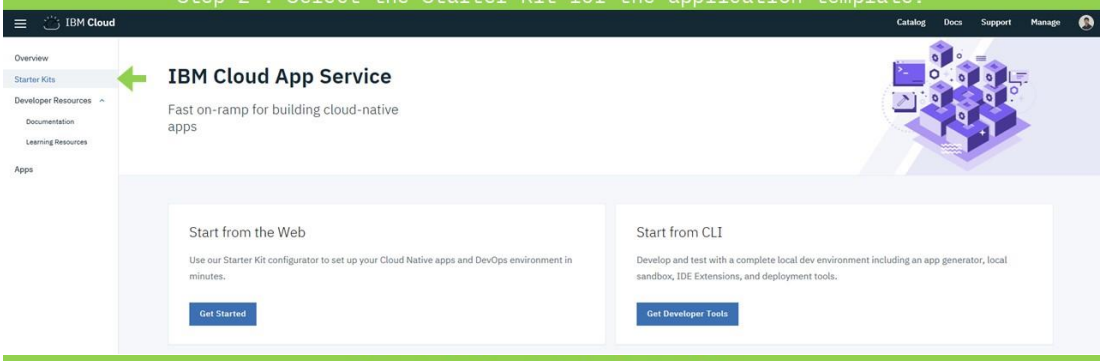
This Microservice is based on node.js framework. This is just a backend API that accepts either of the two parameters – Duration or Iterations through REST API calls. On Accepting the input parameters from the URL, this service returns the JSON object which contains the Fibonacci number generated using the parameters supplied.

Following is the process of creation of this Microservice-

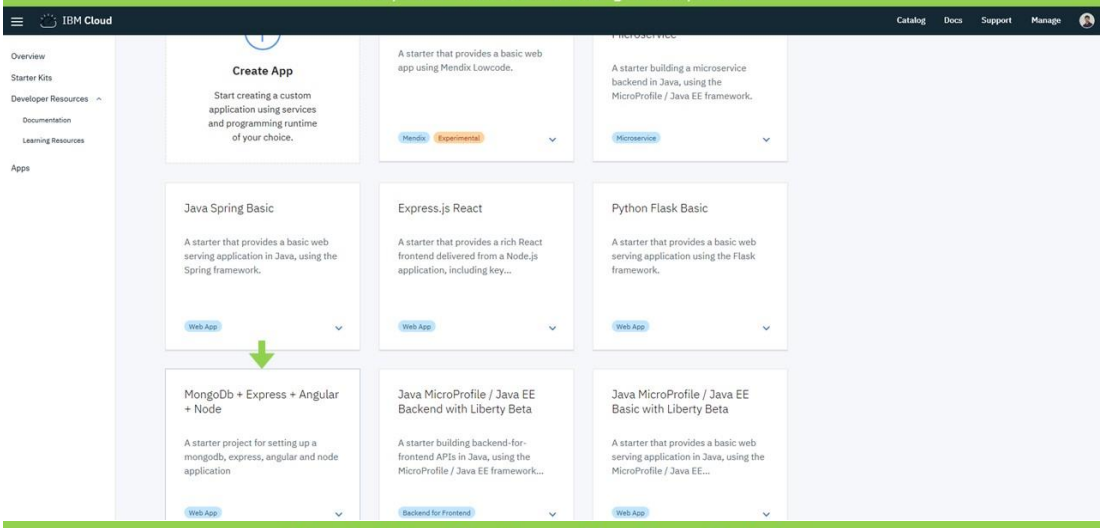
Step 1 : When you are on the IBM Cloud Dashboard, Go to the Web Apps section from the Hamburger Menu



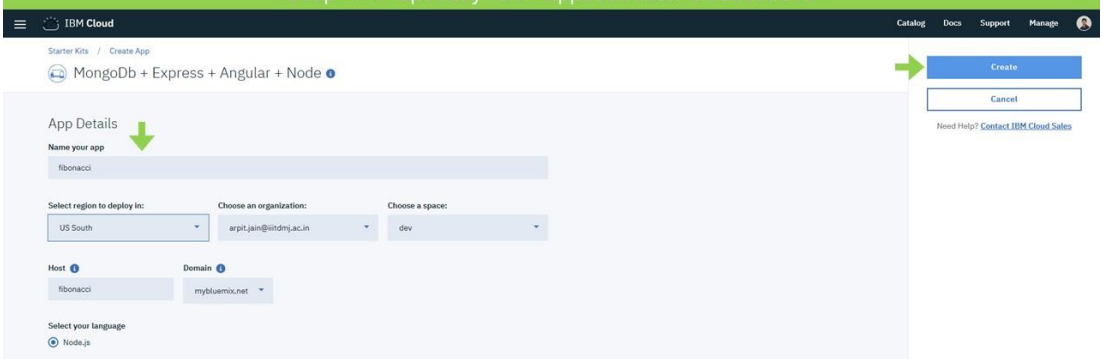
Step 2 : Select the Starter Kit for the application template.



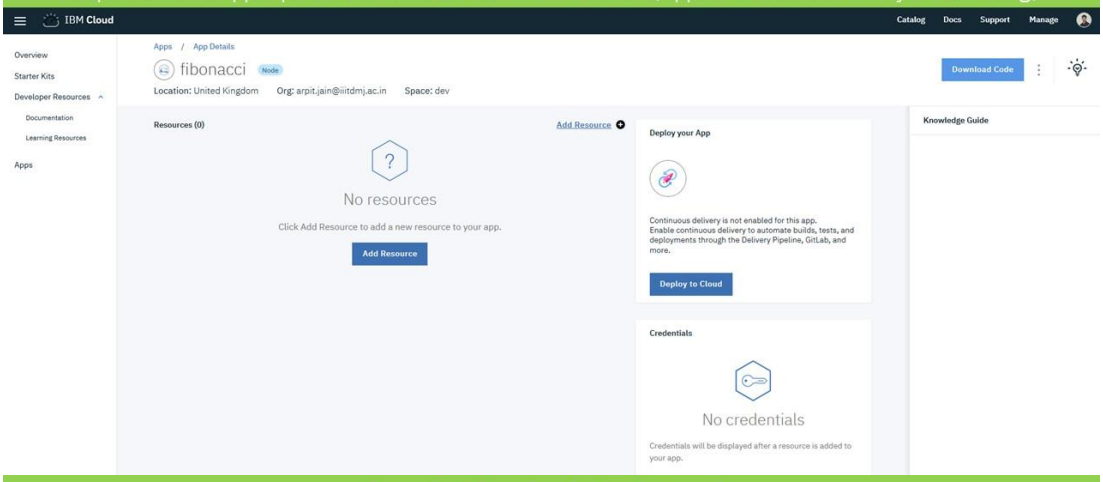
Step 3 : Select Node.js template



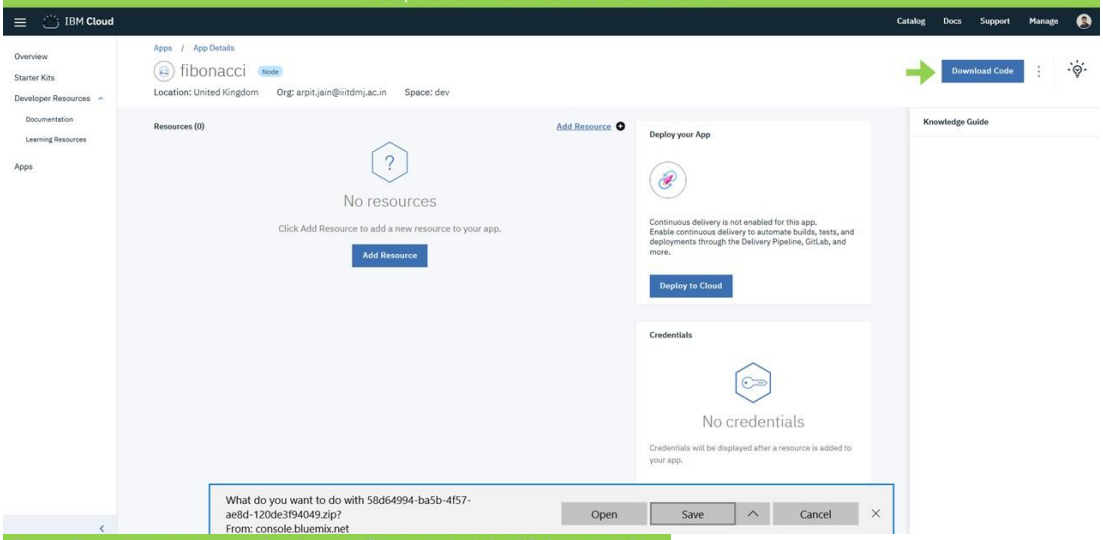
Step 4 : Specify the application attributes



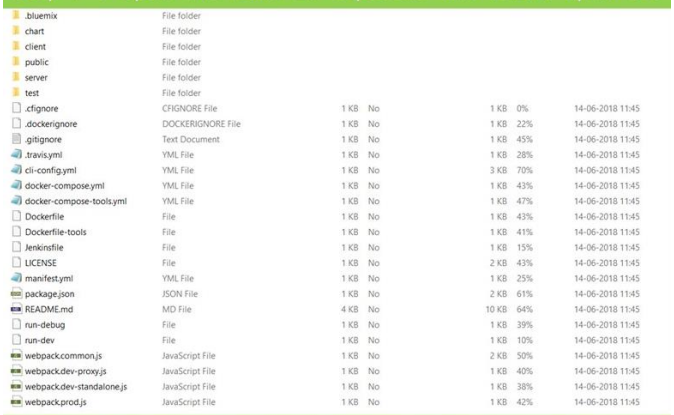
Step 5 : The App space is created on the cloud. (Application is not yet running)



Step 6 : Download starter code

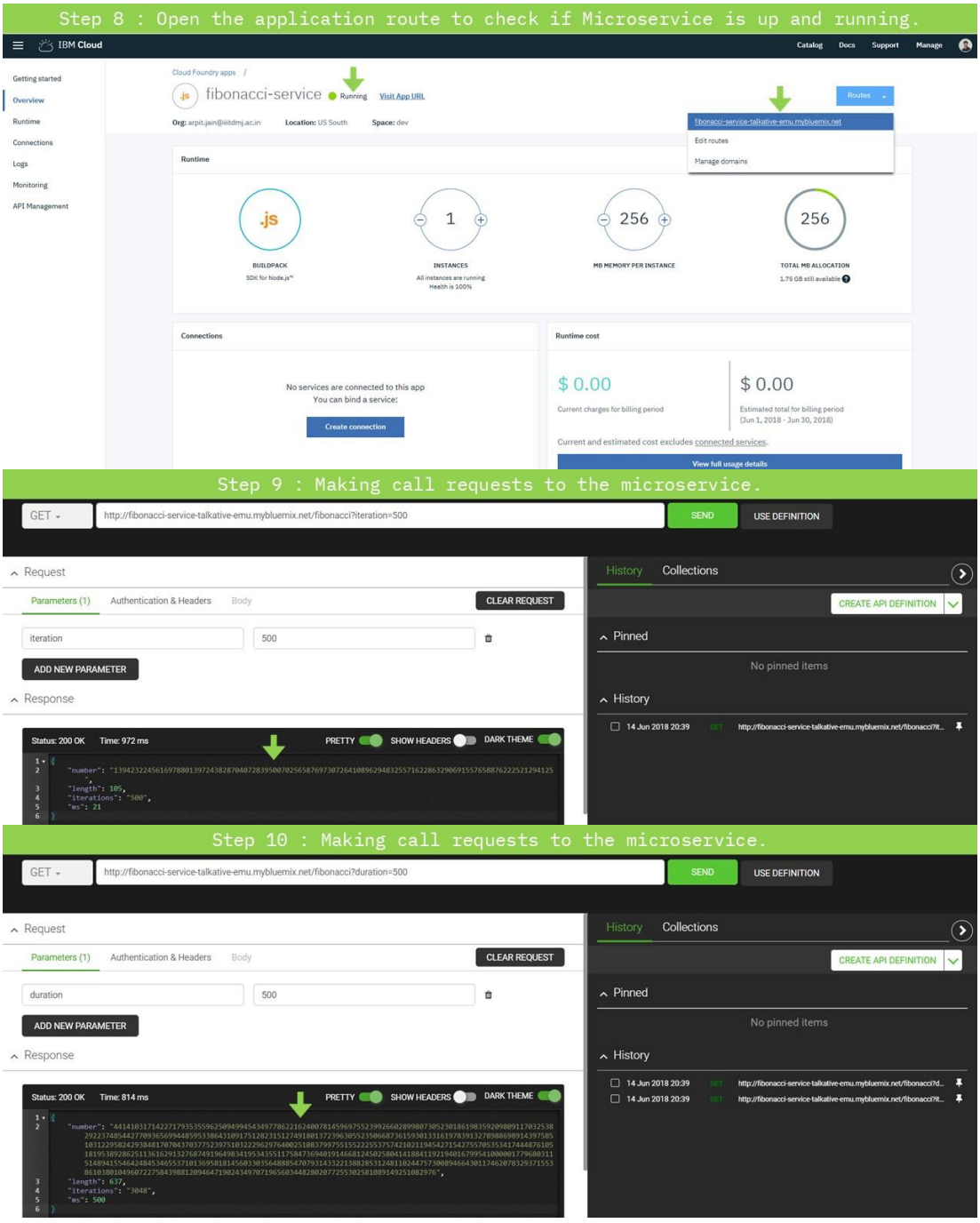


Step 7 : Open the code in explorer and build upon it.



Code the Fibonacci Microservice using this starter template.

After writing the code for the Fibonacci application, the microservice was pushed to the IBM Cloud using the Bluemix Command Line Interface.



Access request to B2B Sterling GitHub repository

To work on and understand the B2B Sterling Integrator product, access to the code repositories on which the product is currently stored is granted to me. Now my task for the next 15 days is to have a look at the code and understand it so that I can convert some parts of it to microservice architecture.

Research Work

Apart from the above demo application deployment on the IBM cloud, I also studied the instruction manual of B2Bi Software and understood the structure of a Business process specification in XML.

Results and Discussions

Microservices are an effective and fault-tolerant manner of designing and developing an application. B2Bi is a large monolithic application which needs to be divided into microservices. By developing and deploying a demo microservice application on the cloud, it's clear that microservice architecture of an application development is vastly scalable and dynamic. Depending upon the requirement, the individual microservice could be replicated without the need to duplicate the entire application.

Conclusions

During these 15 days, I completed the demo microservice application development and deployment on the IBM Cloud. Also, I got the access request to the GitHub repository of the codebase of the B2Bi product.

Live demo of the application can be seen at this **Link.** (Unfortunately, the application is unstable on the Chrome Browser due to JavaScript issues. Any other browser like Microsoft Edge and Mozilla Firefox are fine.)

Next Target

My target for the next 15 days is to understand the B2Bi Application code, viewing the deployed code/services on the servers and start researching on the development of Java JSP Dynamic web application development.