Application of Microservice Architecture on B2B Applications

(IBM Watson Customer Engagement)

by

Vipin Dhonkaria
(Roll No. 2015274)

Supervisor(s):

External
Atul A. Gohad
(IBM ISL, Bangalore)

Internal
Dr. Manish Kumar Bajpai
(PDPM IIITDM Jabalpur)

Computer Science and Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING JABALPUR

(15th June 2018 – 28th June 2018)

Introduction

Microservice Architecture

Microservice Architecture is an architectural style that structures an application as a collection of loosely coupled services, which implement business capabilities. In a microservices architecture, services are fine-grained and the protocols are lightweight. The benefit of decomposing an application into different smaller services is that it improves modularity and makes the application easier to understand, develop, test, and more resilient to architecture erosion.

Advantages:

- Frequent Releases can be handled easily.
- Enables continuous integration and delivery of software components.
- Better fault isolation; if one microservice fails, the others will continue to work and the entire system will continue to function rather than crashing down.

Disadvantages:

The individual development teams manage and make changes to their microservices independently, making it difficult to keep all of the pieces working together as a single unified application.

Deployment complexity. In production, there is also the operational complexity of deploying and managing a system comprised of many different service types.

IBM Bluemix

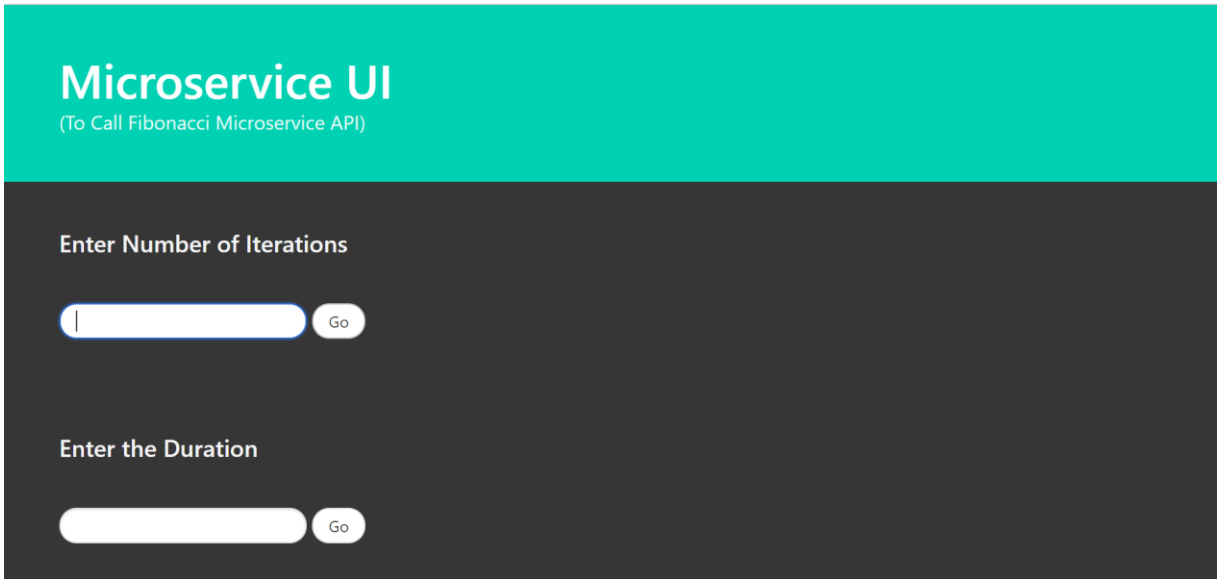
It is a cloud platform as a service (PaaS) developed by IBM. It supports several programming languages and services as well as integrated DevOps to build, run, deploy and manage applications on the cloud. Bluemix is based on CloudFoundry open technology and runs on SoftLayer infrastructure.

Bluemix is based on Cloud Foundry open technology and runs on SoftLayer infrastructure. Bluemix supports several programming languages including Java, Node.js, Go, PHP, Swift, Python, Ruby Sinatra, Ruby on Rails and can be extended to support other languages such as Scala through the use of buildpacks.

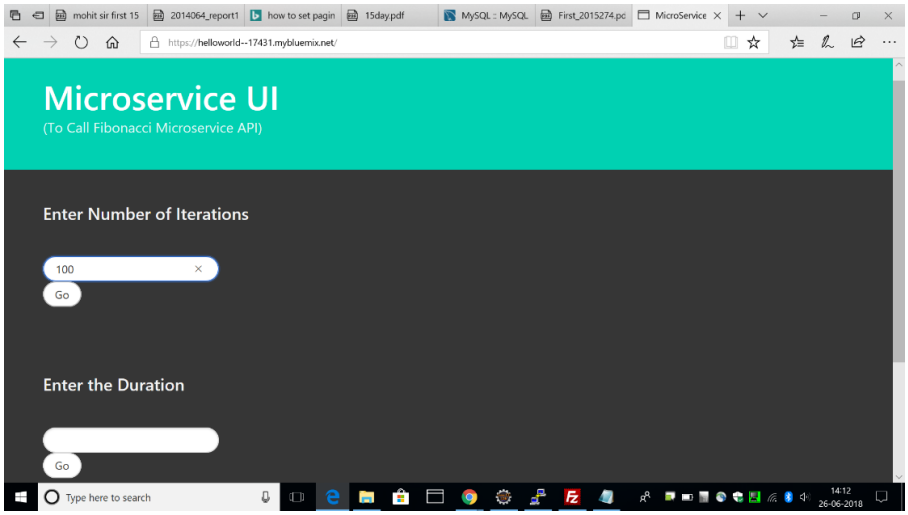
Report on the Present Investigation/Progress

(Progress during this 15-days period)

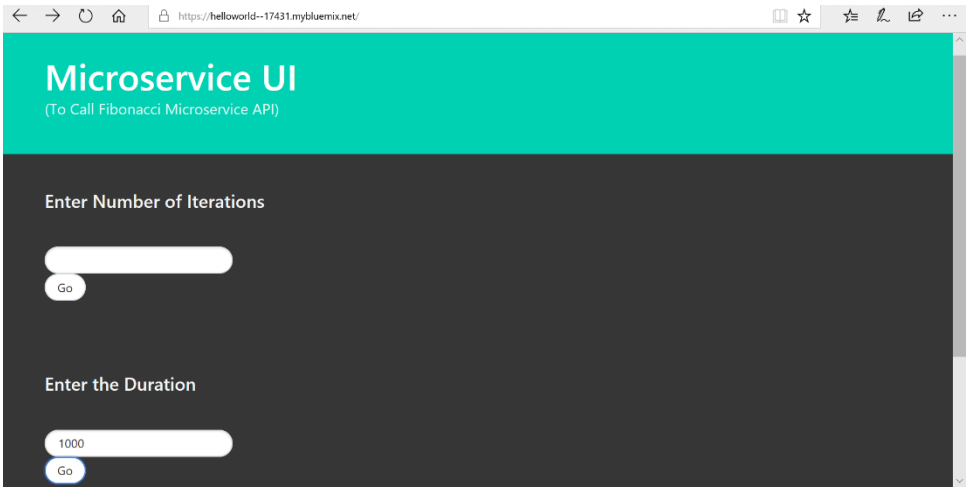
I have created one Dynamic Web Project on eclipse workspace and created an interface to call the microservice API. I have selected IBM Bluemix as a server over which this application is deployed. When I am creating the server IBM Bluemix takes my credentials and create the web application into my account. From this application I made a call to Fibonacci service API and display the output in json format.
So this make our work easy, simultaneously we can make changes on both the applications and deploy it on server and run it.

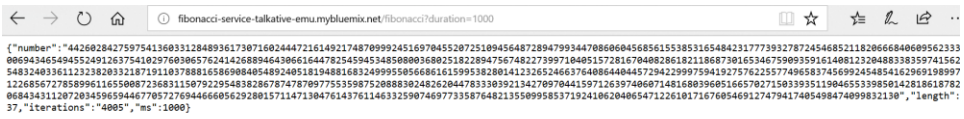


Entering the number of iterations will print the fibonacci series that number of times.



Entering the amount of duration will execute the fibonacci series for that amount of milliseconds.





Link of Deployed Application: http://helloworld--17431.mybluemix.net/

Results and Discussions

After deploying one application on IBM Bluemix server and make a call to microservice API using it, makes very clear that it is the best practise to decompose a large monolithic application to microservices.
We can make changes on multiple of services and can deploy them simultaneously.

Advantages of microservices Architecture:

- Frequent Releases can be handled easily.
- Enables continuous integration and delivery of software components.
- Better fault isolation; if one microservice fails, the others will continue to work and the entire system will continue to function rather than crashing down.
- Codes can be written in many languages and frameworks. Each microservice can be written in a different manner irrespective of other microservices.
- Each microservice has its own database eliminating the discrepancies of updating a common database from several simultaneous calls.

Conclusions

Microservice Architecture has some disadvantages too. It is very complex to design and an expensive process because it uses more resources. If we don't want to further expand your application then don't go for it but if you need frequent releases and want to maintain each module independently, then it is an amazing alternative to consider.

My target for the next 15 days is to build a war file and deploy it on IBM server. The war file is generated from java application and deploy it on IBM B2B sterling Integrator server.