Micro services testing

Micro service is the buzzing term in the current software development industry. Business is interested in converting as many applications as possible into the micro services architecture as it gets the applications work relatively faster.

How were the applications built before?

If taken a keen look at the legacy architectures of the applications or some of the current applications that are in use, they are pretty much built in a monolith fashion the entire code base i.e. various components of the application are smushed as a single package, hence the degree of coupling is more. This indeed results in the application to be working in a thread safe model and the communications and events happening in a synchronous way. This architecture delays in responding to the user as the resources would have been occupied by the other request.

Request response Data layer

---------- --------------

Application UI

Figure: Monolith architecture of an application

Controller

Database

Limitations of the monolithic architectural design:

* The code base is tightly coupled due which the degree of dependency is higher
* Since there is only one resource which serves the entire application slowness can be observed
* If at all one part of the system broke, entire system is affected and the whole application goes down until the issue is sorted out
* Triaging gets real tough for the developers, as the code base built as a whole unit.

Micro service architecture

The current industry is annoyed of the above limitations and started looking for a solution and thanks to micro services architecture as it has solutions for all the above problems. Let’s understand a bit more of this structure

* Development needs to identify the services and need to break them into smaller minces
* Every micro service component should be pointed to its own data base schema
* There should be a powerful orchestration service development to make sure the proper communication between one micro services to another micro service.

Below is the schematic diagram of micro service architecture

Cluster-3

Cluster-2

Cluster-1

Micro service-3

Micro service-2

UI<-->MS

Micro service-1

MS-DB| Figure: Micro service based architecture of an application

Advantages of the above design:

* Since the part of code base is split into smaller chunks, thus we make them independent.
* Since every micro service is responsible for its own job we see the application responds quite faster.
* Due to the one-to-one mapping of DB schema and micro service we don’t overload the database like we do in a single instance architecture
* Since these are independent, if any of the micro services breaks still the other portion of the application can operate.
* These micro services can talk to each other through APIs
* This architecture will help us to protect the service endpoints by implementing some JWKS authorizations if business needs.

This is a quick brief of micro services at a high level, understanding this is quite essential. This paves the path for testing them effectively at API level.

Thanks for the read, hope there would have been some take away from this crisp discourse ☺