Wickramasinghe T.W.M.K.L IT19243290
IT4090 – Cloud Computing
Year 4, Semester 2 Assignment 1

Introduction

Monolithic architecture is all features of a software reside in a single code base and deployed as a single file. If any code updates are required, then those updates can't be accommodated independently. Developer has to use the same code base, make the required code changes and then re-deploy the updated code. So even if a single change is required, the whole code base is touched and re-deployed.

Monolithic scaling issues and how to cover these issues in Microservices

In monolithic architecture we can't scale individual components. Scalability in monolithic architecture is difficult to achieve due to the size and scale of the structure. This option is hard to upgrade. Symmetric scalability can harm systems, especially those created with a lot of non-functional speed and scalability needs. This may result in the sharing of system resources between services and resources that need to be scalable and those that do not. Systems typically require asymmetric scalability, where the helper or non-core features may have lower hardware or memory requirements and the core features or key business logic services may have higher scalability requirements.

In Microservice architecture, AKA microservices, are a specific method of designing software systems to structure a single application as a collection of loosely coupled services. Applications tend to begin as a monolithic architecture (more on that below), and over time grow into a set of interconnected microservices. The main idea behind microservices is that some types of applications are easier to build and maintain when they are broken down into many small pieces that work together. Though the architecture has increased complexity, microservices still offer many advantages over the monolithic structure.

Because microservices are made of much smaller components, they can take up fewer resources and therefore more easily scale to meet increasing demand of that specific component. As a result of their isolation, microservices can properly function even during large changes in size and volume, making it ideal for enterprises dealing with a wide range of platforms and devices.

Serverless model and Function as a Service (Faas)

Serverless model also known as function as a service (FaaS), is a cloud computing code execution model in which the cloud provider fully manages starting and stopping of a function's container platform as a service (PaaS) as necessary to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request.

A cloud computing approach called function as a service (FaaS) enables clients to create apps and deploy functions while only being charged when the capability is used. FaaS, often known as serverless computing, is frequently used to launch microservices.

Serverless server-less execution of modular chunks of code on the edge is possible using function-as-a-service (FaaS). In reaction to an event, such as a user clicking on an element in a web application, FaaS enables developers to write and change a piece of code. Is Microservices always the way to go and Monolithic architecture never to be used? No, the Reason, it is much easier to add new features to a microservice application than a monolithic one. Easier understanding. Split up into smaller and simpler components, a microservice application is easier to understand and manage

References

[1] https://microservices.io/

[2] https://ieeexplore.ieee.org/abstract/document/8529465

[3] https://www.techtarget.com/searchitoperations/definition/function-as-a-service-FaaS#:~:text=Function%20as%20a%20service%20(FaaS)%20is%20a%20cloud%20computing%20model,referred%20to%20as%20serverless%20computing.

[4] https://www.oreilly.com/library/view/production-ready-microservices/9781491965962/ch04.html#:~:text=There%20are%20several%20approaches%20one,traffic%20level%20grows%20over%20time.

[5] https://articles.microservices.com/monolithic-vs-microservices-architecture-5c4848858f59