Spring cloud provides the tools that allow us to work with distributed systems that leverage microservice architecture

The key focus of Microservice architecture is to create independently deployable services that interface each other to work as distributed systems.

Microservices are quickly becoming the go-to choice for applications living in the cloud that need to service a substantial amount of traffic that comes from multiple platforms.

Microservices have introduced a new way for designing, building, and architecting software applications.

In order to understand microservices better, we will compare and contrast monolithic approach against the microservice approach.

Under a monolithic approach, entire application is built on single archive (war). In case of Java application, package it as war file.  We find the classes and libraries that make up the application.

Such applications are often built using layered approach.

Example: Reservation system for restaurant. May be we have several services.

Service1 – booking an application

Service2 – Providing table availability

Service3 – track customer visits

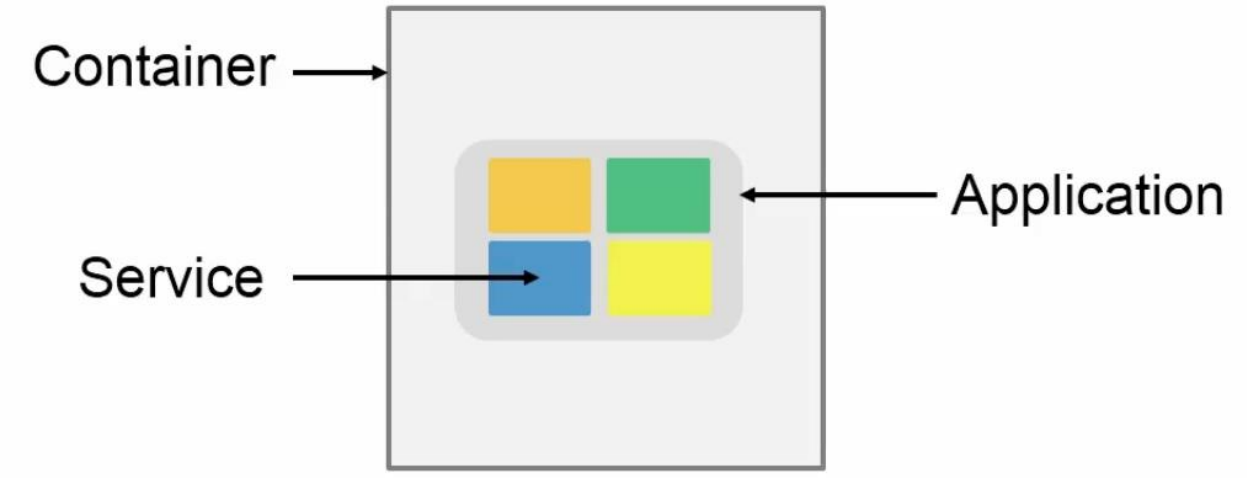


Figure 1

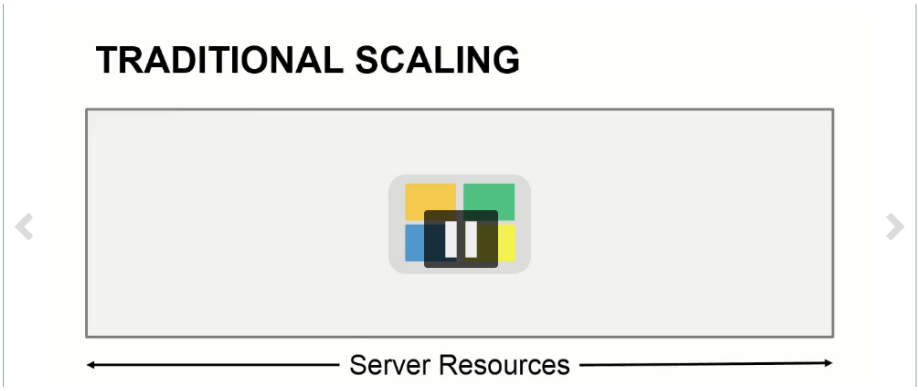
Q. What in case any changes is needed in booking application services?

A. In that case, any changes in the service results in entire archive to be rebuilt and redeployed to the container.

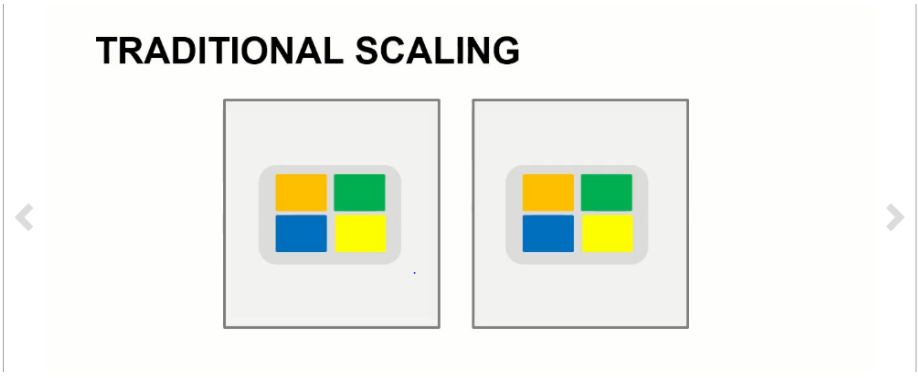
Q. What in case if server comes into substantial stress due to high request load?

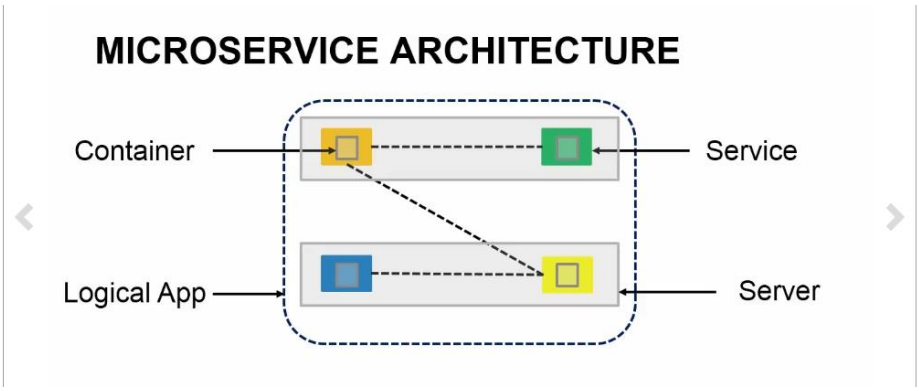
A. We need to scale out server at that point using some mechanism.

a. Scale vertically – meaning add more server resources such as more ram, more cpu.




b. Scale Horizontally – In this case, we would be adding another instance of the application and using that instance to handle some of the load.






Focus on particular domain

They communicate outside of the same process.

Exposed as API

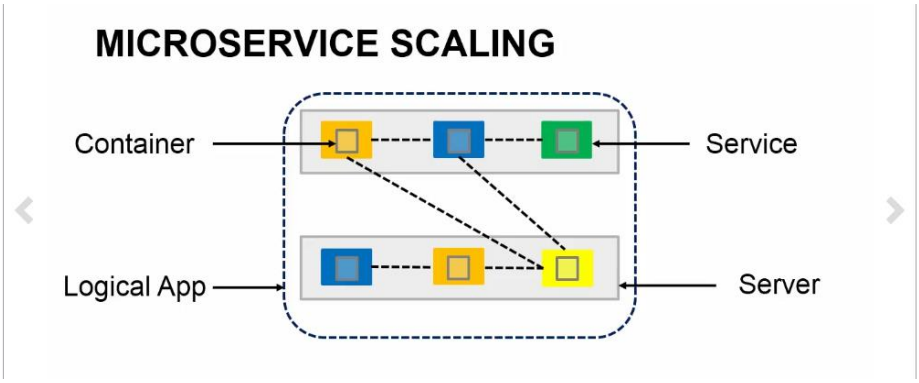
Application is made up of many services and communicating over a protocol such as HTTP.

Container is embedded in the service itself usually as a library

We are not build large applications that encompass the entire set of responsibilities.

Instead we are segmenting the responsibilities of the application as small services independently.

**Scaling the services**




**Definition of Microservices**

- A software design that focuses on integrating a group of lightweight and compartmentalized services over a communications layer to form a single application.

## CHARACTERISTICS

- Small independent services organized around business capabilities
- APIs expose service and its capabilities
- Services interact in different processes
- Data stores are focused and specific to the service
- Stateless
- Fault tolerant

## BENEFITS

- Compartmentalizes business capabilities
- Services are loosely coupled and modular
- Highly scalable and replaceable
- Independent service management
- Supports concurrent processing
- Allows for smaller teams and faster time to market
- Language agnostic

http://localhost:8080/autoconfig - shows all configurations done by boot

/beans – list of all beans

/configprops – about configuration proeprties

/info – generic information

/mappings

Endpoints.health.sensitive=false