Cloud Computing - Mini Project Report
Breaking Down Monoliths
April 2023

Submitted By:
Name : Aryan Puranik
SRN : PES1UG20CS081
VI Semester Section B
PES University

Short Description and Scope of the Project

- This problem statement involves breaking down a monolithic application to a microservice architecture.
- We were given a flask app which performs arithmetic calculations.
- That flask application needs to be changed into microservices.
- The given flask application contains all the functions(addition,subtraction) in one container.
- So, if the container is down the whole application won't work.

- So, if we divide that application into different containers, then if a container is down, the whole application won't be unavailable.
- Suppose addition container is down, only that functionality won't work, rest all functionalities work normally.
- This is the advantage of breaking a monolithic application into microservices.

Methodology

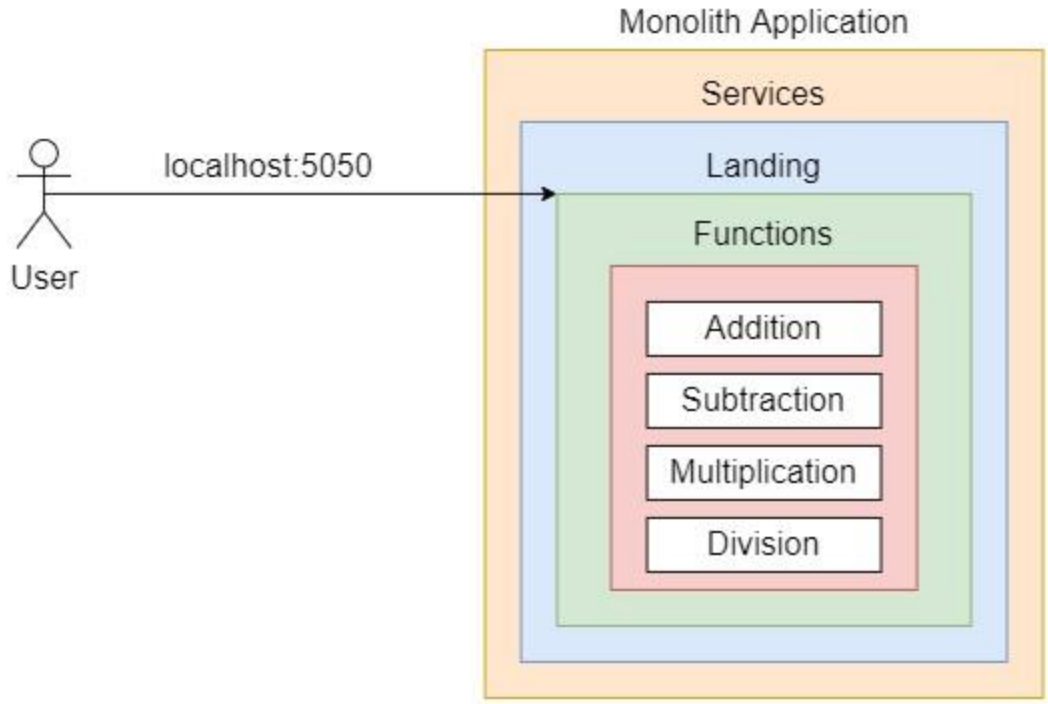
- As the given application is a flask application, we used the Resource class from flask_restful library to divide it into different containers.
- First there were some changes to be done for the flask applicationto run.
- After those errors were fixed, we started breaking it.
- For each functionality we created separate directories(containers) and the functionality will be present in that container.
- Each container has its own docker file.
- The landing page routes the application based on functionality using resource class.
- Then we wrote the docker-compose file which executes the required docker files based on the functionalities.

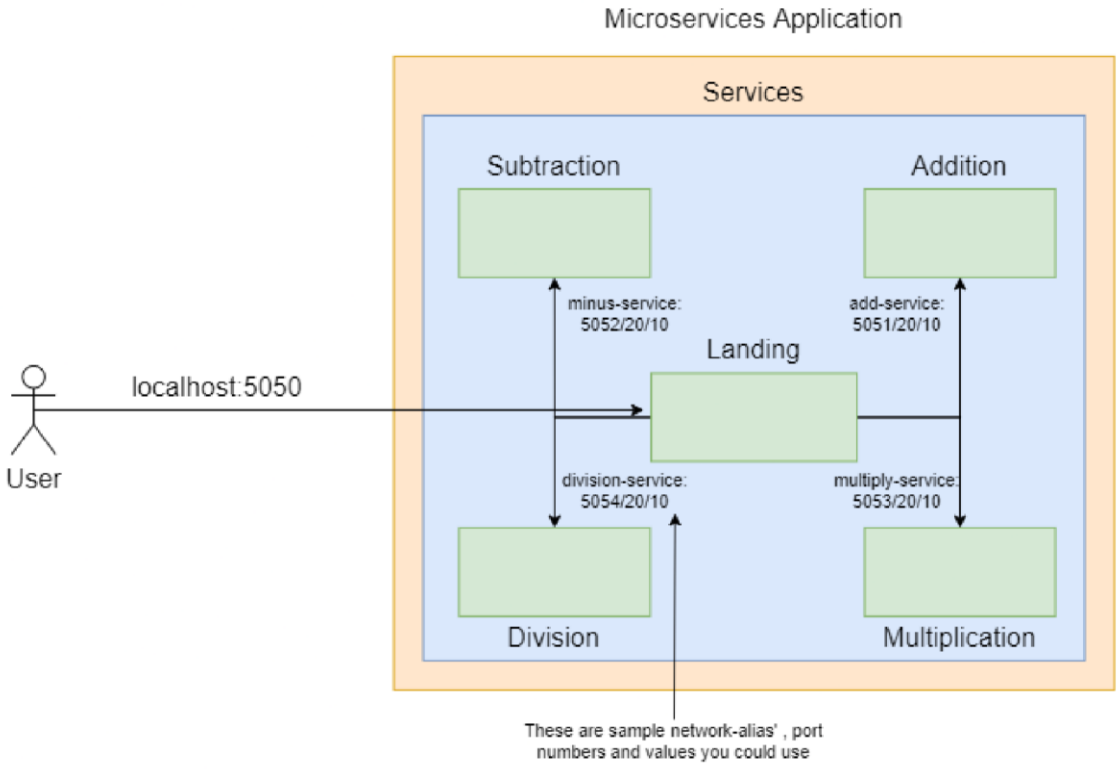
Testing

We have tested the application with different values in every functionality.

Results and Conclusions

Finally we were able to convert the monolithic application into a microservice architecture.


Monolith Application
Services
Landing
Functions
Addition
Subtraction
Multiplication
Division
localhost:5050
User


Microservices Application
Services
Subtraction
Addition
minus-service:
5052/20/10
add-service:
5051/20/10
Landing
localhost:5050
User
division-service:
5054/20/10
multiply-service:
5053/20/10
Division
Multiplication
These are sample network-alias' , port
numbers and values you could use