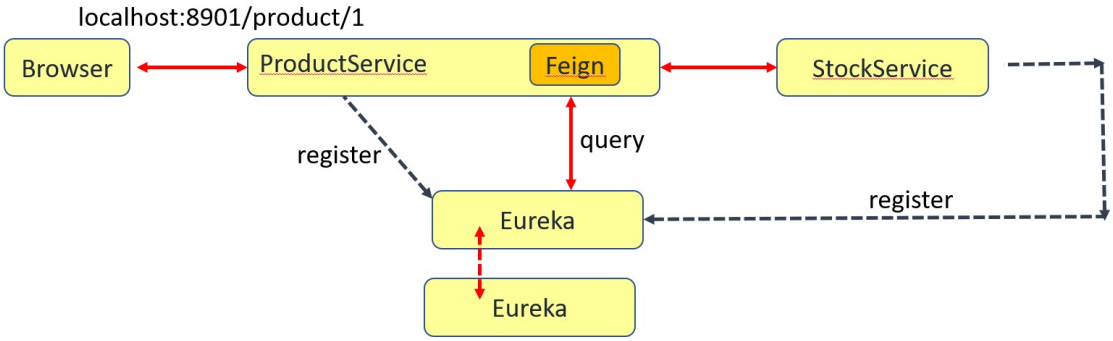
Lab 8. Eureka cluster, load balancing, API gateway

Part 1: Eureka cluster

Write 2 Eureka services that work in a cluster of replicas.
Give both the StockService and the ProductService the URL of both Eureka services.

Start Eureka Service 1
Start Eureka Service 2
Check the eureka dashboard if both services are replica of each other.
Start the StockService
Check if the StockService is registered in both eureka instances
Start the ProductService
Check if the ProductService is registered in both eureka instances
Check if everything works



Now kill one Eureka Service instance.
Check if everything still works

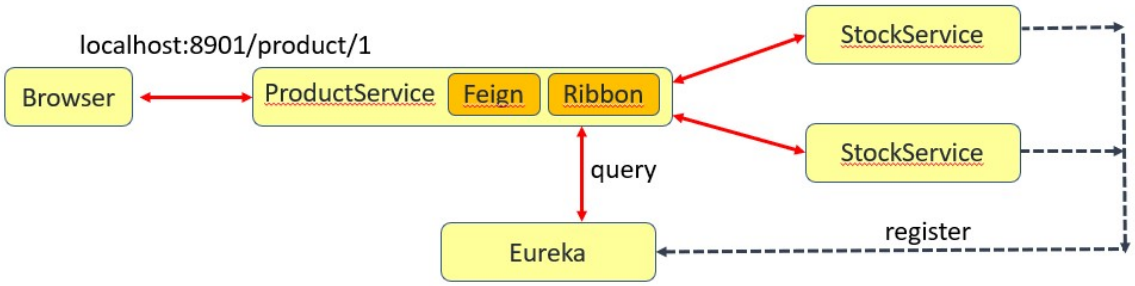
Part 2: load balancing with ribbon

Copy and paste the StockService, and make sure the 2nd stockservice runs at a different port.
Let the 2nd stockservice return a different number that the 1st stocksevice.

First run the EurekaServer and check the dashboard if it works
Then run the first StockService and see in the Eureka dashboard if it is registered.
Then run the second StockService and see in the Eureka dashboard if it is registered.
Then run the ProductService and see in the Eureka dashboard if it is registered.

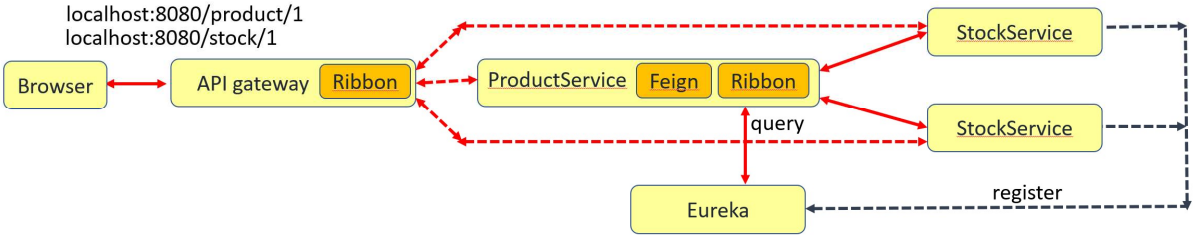
Then if you call the ProductService, it should load balance the call to the StockService.
Then stop one StockService. Check in the Eureka dashboard if it is removed. (This can take up to 30 seconds)
Now if you call the ProductService, it should call to the StockService that is still running.

Then start the stopped StockService again. Check in the Eureka dashboard if it registered again.
Now if you call the ProductService, it should load balance the call to the StockService again.



Part 3: API gateway

Write an API gateway in front of the ProductService and StockService that the browser can call both services via the API gateway



Part 4

A problem with the API gateway is that it is a single point of failure. Describe how you would solve this problem?

Part 5

A problem with a microservice architecture is that is difficult to keep track of the business processes that run on the microservice architecture. Describe how you could solve this problem?

Part 5

A problem with a microservice architecture is that is difficult to keep the interfaces of the different microservices in sync with each other. Describe how you could solve this problem?

What to hand in?

1. A zip file containing all services for part 1, 2 and 3
2. A PDF for part 4, 5 and 6
3. Write a readme.txt file with the following statement and sign with your name:

I hereby declare that this submission is my own original work and to the best of my knowledge it contains no materials previously published or written by another person. I am aware that submitting solutions that are not my own work will result in an NC of the course.

[your name as signature]