Microservices old

A diagram of a computer

Description automatically generated

We have users with dpets .Hystrix dashboard shows us how many services are working and failings. WE use Config Server to hold config details of all microservices in git repo using cloud config server. Zipkin and sleuth for distributed log tracing .

Lets create dept service

Create packages entity , controller , services ,repository

*@Entity  
@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class* Department {  
 *@Id  
 @GeneratedValue*(strategy = GenerationType.AUTO)  
 *private* Long departmentId;  
 *private* String departmentName;  
 *private* String departmentAddress;  
 *private* String departmentCode;  
  
  
}

Reporsitry

*@Repository  
public interface* DepartmentRepository *extends* JpaRepository<Department, Long> {  
}

Service

*@Service  
public class* DepartmentService {  
 *@Autowired  
 private* DepartmentRepository departmentRepository;  
  
}

Service

*@Service  
public class* DepartmentService {  
 *@Autowired  
 private* DepartmentRepository departmentRepository;  
  
}

Controller

*@RestController  
@RequestMapping*("/departments")  
*@Slf4j  
public class* DepartmentController {  
 *@Autowired  
 private* DepartmentService departmentService;  
  
 *@PostMapping*("/")  
 *public* Department saveDepartment(*@RequestBody* Department department){  
 log.info("inside saveDept");  
 *return* departmentService.saveDepartment(department);  
 }  
}

Repo after adding method swil be like below , pls check spelling of findByDepartmentId for repo

*@Repository  
public interface* DepartmentRepository *extends* JpaRepository<Department, Long> {  
 Department findByDepartmentId(Long departmentId);  
}

Keep addin methods and then go to yaml to add configuration.Use postman to test it.

Create service as above steps .

Here in controller we need to create a wrapper object for User along with dept also .sWe don’t need to ceat dept class again here in User service .We creat package VO and create dpet object

*@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class* Department {  
 *private* Long departmentId;  
 *private* String departmentName;  
 *private* String departmentAddress;  
 *private* String departmentCode;  
}

We createmwrapper class RepsonseTempalteVO in VO package.Here data is for getter setter , and all args and no args for contructors

*@Data  
@AllArgsConstructor  
@NoArgsConstructor  
public class* ResponseTemplateVO {  
 *private* User user;  
 *private* Department departmennt;  
}

In ocntrolle r add this methid

*@GetMapping*("/{id}")  
*public* ResponseTemplateVO getUserWithDepartment(*@PathVariable*("id") Long userId){  
 log.info("Inside getUserWithDepartment of UserController");  
 *return* userService.getUserWithDepartment(userId);  
}

Now in service , to get dept for user id , we need a Rest Template to call dept service.Creat rsettrempalte method in main controller class

*@SpringBootApplication  
public class* UserServiceApplication {  
  
 *public static void* main(String[] args) {  
 SpringApplication.run(UserServiceApplication.*class*, args);  
 }  
  
 *@Bean  
 private* RestTemplate restTemplate(){  
 *return new* RestTemplate();  
 }  
  
  
}

And autowried it in sercive to use it

*@Service  
@Slf4j  
public class* UserService {  
 *@Autowired  
 private* UserRepository userRepository;  
  
 *@Autowired  
 private* RestTemplate restTemplate;  
 *public* User saveUser(User user) {  
 log.info("Inside saveUser of UserService");  
 *return* userRepository.save(user);  
 }  
  
 *public* ResponseTemplateVO getUserWithDepartment(Long userId) {  
 log.info("Inside getUserWithDepartment of UserService");  
 ResponseTemplateVO vo = *new* ResponseTemplateVO();  
 User user = userRepository.findByUserId(userId);  
 Department department = restTemplate.getForObject("http://localhost:9001/departments/" +user.getDepartmentId(), Department.*class*);  
 vo.setUser(user);  
 vo.setDepartmennt(department);  
 *return* vo;  
 }  
}

Use postman to test departmtnest and useress post requests.

To manage lot of microservices , we use Service Registry to maintain it to see status updates , ports, url etc.

Now we create service Regsitry, we creat proejt in spring intlzr , we add Eureka Server- Spring cloud discovery.

We make sure we resiter-with-eureka as false in propoeties.uyaml file.

Now we van use service names in url and make rest Teampalte bean to be loadBalanced

Department department =  
 restTemplate.getForObject("http://DEPARTMENT-SERVICE/departments/" + user.getDepartmentId()  
 ,Department.*class*);

*public class* UserServiceApplication {  
  
 *public static void* main(String[] args) {  
 SpringApplication.run(UserServiceApplication.*class*, args);  
 }  
  
 *@Bean  
 @LoadBalanced  
 public* RestTemplate restTemplate() {  
 *return new* RestTemplate();  
 }  
}

No lets create api-gateway ie cloud-gateway project so request scome through 9191 port added in application.yaml file.

Now lets implement Hysterix circruit breaker .If one of services is not running then it will run the fallback method to notify the user

We add Hysterix dependency to Cloudgateway in pom and and enable Hystric in main method

*@SpringBootApplication  
@EnableEurekaClient  
@EnableHystrix  
public class* CloudGatewayApplication {  
  
 *public static void* main(String[] args) {  
 SpringApplication.run(CloudGatewayApplication.*class*, args);  
 }  
  
}

We create fallback controller here

*@RestController  
public class* FallBackMethodController {  
  
 *@GetMapping*("/userServiceFallBack")  
 *public* String userServiceFallBackMethod() {  
 *return* "User Service is taking longer than Expected." +  
 " Please try again later";  
 }  
  
 *@GetMapping*("/departmentServiceFallBack")  
 *public* String departmentServiceFallBackMethod() {  
 *return* "Department Service is taking longer than Expected." +  
 " Please try again later";  
 }  
}

No we add congiuration to call these methos in application.yml file in cloudgateway project

filters:  
 - name: CircuitBreaker  
 args:  
 name: USER-SERVICE  
 fallbackuri: forward:/userServiceFallBack

whoel file is like below now

spring:  
 application:  
 name: API-GATEWAY  
 cloud:  
 gateway:  
 routes:  
 - id: USER-SERVICE  
 uri: lb://USER-SERVICE  
 predicates:  
 - Path=/users/\*\*  
 filters:  
 - name: CircuitBreaker  
 args:  
 name: USER-SERVICE  
 fallbackuri: forward:/userServiceFallBack  
 - id: DEPARTMENT-SERVICE  
 uri: lb://DEPARTMENT-SERVICE  
 predicates:  
 - Path=/departments/\*\*  
 filters:  
 - name: CircuitBreaker  
 args:  
 name: DEPARTMENT-SERVICE  
 fallbackuri: forward:/departmentServiceFallBack

and add hysteric config for 4seconds and add configuration for actuator to enable hystrix strram to use it in hysrix dahsboad

hystrix:  
 command:  
 fallbackcmd:  
 execution:  
 isolation:  
 thread:  
 timeoutInMilliseconds: 4000

management:  
 endpoints:  
 web:  
 exposure:  
 include: hystrix.stream

Now create hytstrix dashboard project in spring inzlr add Eureka client and hystrix dabsarod.

Create application.yml file

server:  
 port: 9295  
  
spring:  
 application:  
 name: HYSTRIX-DASHBOARD  
  
  
hystrix:  
 dashboard:  
 proxy-stream-allow-list: "\*"

Enable hystrc dashboard in main class

*@SpringBootApplication  
@EnableHystrixDashboard  
@EnableEurekaClient  
public class* HystrixDashboardApplication {  
  
 *public static void* main(String[] args) {  
 SpringApplication.run(HystrixDashboardApplication.*class*, args);  
 }  
  
}

You can see it on eureka localhost:8761 and to go to Hydtrix dasboad use local host:9295/hystrix.

Order of starting services is api gateway , services, hystrix, .

We need to see if hysterix stream is working on not

Lolcahost:9191/actuator/hystrix.stream to see data pings.

Use this url tand pasteit in dashboard and click monitor .Use postman and you can see graph .

Now we creat one config server and get all our configurations from conig server in github etc.All services to that repo to fetch configuration.Add uereka client , config server in spring intlzr.Create yml file

server:  
 port: 9296  
  
spring:  
 application:  
 name: CONFIG-SERVER  
 cloud:  
 config:  
 server:  
 git:  
 uri: https://github.com/shabbirdwd53/config-server  
 clone-on-start: *true*

Enable eurekaclient, enableconfigserver in main class

*@SpringBootApplication  
@EnableEurekaClient  
@EnableConfigServer  
public class* CloudConfigServerApplication {  
  
 *public static void* main(String[] args) {  
 SpringApplication.run(CloudConfigServerApplication.*class*, args);  
 }  
  
}

We creat github repo to store all congiuration there.

Create config0server-springboot2 project and creat a application.yml file

eureka:  
 client:  
 register-with-eureka: *true* fetch-registry: *true* service-url:  
 defaultZone: http://localhost:8761/eureka/  
 instance:  
 hostname: localhost

and add its uri in application .yml in project service file in IDE

Now we add spring-clod-starter-config dependency in dept service, user service,service resgistry , cloud gatweway , hystrix dashboard service.

Now we add configuration in yml files so services can talk to config server.THer is one thing that we use application.yml is used for application context, for getting all cloud config ther is usually bootstrap.yml.

For getting all cloud config from github , there is bootstrap.yml fiel

spring:  
 cloud:  
 config:  
 enabled: *true* uri: http://localhost:9296

above is for dpet service , we add same in user service , cloud gateway , hytsrix dahsbaord, cloud config service we have directly, no need to add in service registry not needed there .

Now lets start all services and test them . First registry , ocnfig server , api gateway , other services in this order.Got to eureka you see this

A screenshot of a computer

Description automatically generated

Test in postman the usersr and dept.reqeusts

Trace all of requeests we use Zipkin and sleuth distributed logging . log tracing by Trace Id which is unique for a request and Span Id which changes according to according to microservice.

We need Zipkin gerver and implement zipkin client in all microservices. We implement using docker.We run docker and go to locajost:9411

WE take depencey zipkin client and Slueth from spring intlzr , got dept service pom file , user service , now we point both to zipkin server url in applicaiton.yml files

server:  
 port: 9001  
  
spring:  
 application:  
 name: DEPARTMENT-SERVICE  
 zipkin:  
 base-url: http://127.0.0.1:9411/

Go to postmamn , run request again go tot zipkin server at localhost:9411 to see logs tracing in UI.

In console , we see like this

INFO [DEPARTMENT-SERVICE,7976adf1c99a0005,7976adf1c99a0005,true] 19477 --- [nio-9001-exec-8] c.d.d.service.DepartmentService : I

If 7976adf1c99a0005 is unique then its dept servie or else other service , that ho we can if its one requrest or multiple ones

A screenshot of a computer

Description automatically generated