Microservices Central Configuration: Central configuration is defining properties in git not in loacal machine applicaiton

1. Work on Remote Repository or in git

1st step : create application.properties file in git >>> write all common properties of all the microservices (like database connection, password,username etc ) in application.properties file .

2nd Step: create properties file with same name as each individual microserivces like ( if we have PersonMS then we make PersonMS.properties file in git ) . And write properties specific to the microservices , meaning What kind of specific properties we want to PersonMS , write accordingly ).

1st step : Now make on configServer springBoot project and give annotation @EnableConfigServer with @SpringBootApplication.

2nd step : give uri of git in application.properties file of ConfigServer application .

server.port=8888

spring.cloud.config.server.git.uri= provide git uri as value

now this configServer application will pick the properties file from git .

Note : to check whether ConfigServer Picked the properties from git or not :

http://{host:port of ConfigServer}/application/default http://localhost:905/application/default

3rd step: now go to microservices and provide url configServer application in bootstrap.properties file of each microservice . because it will run before application.properties file and pick the properties from configServer .

spring.cloud.config.uri=http://localhost:905 ( give this url in microservices application file to pick the data from configserver that is picked from git by configServer)

4th :

Eureka server is not supposed to run in single instance since if only one instance running and that got failed then all the microservices got down . so we need to run microservices running in multiple port which is called cluster Eureka server :

For this we need to

1st :configure application.yml file in EurekaServer :

---

spring:

profiles: peer-1

application:

name: eureka

server:

port: 9001

eureka:

instance:

hostname: peer-1

client:

registerWithEureka: true

fetchRegistry: true

serviceUrl:

defaultZone: http://peer-2:9002/eureka/,http://peer-3:9003/eureka/

---

spring:

profiles: peer-2

application:

name: eureka-server-clustered

server:

port: 9002

eureka:

instance:

hostname: peer-2-server.com

client:

registerWithEureka: true

fetchRegistry: true

serviceUrl:

defaultZone: http://peer-1:9001/eureka/,http://peer-3:9003/eureka/

---

spring:

profiles: peer-3

application:

name: eureka-server-clustered

server:

port: 9003

eureka:

instance:

hostname: peer-3-server.com

client:

registerWithEureka: true

fetchRegistry: true

serviceUrl:

defaultZone: http://peer-1:9001/eureka/,http://peer-2:9002/eureka/

2nd: now go to git and add

eureka.client.service-url.defaultZone=http://Eur1:2222/eureka,http://Eur2:2223/eureka, <http://Eu3:2224/eureka>

3rd: done >> when configServer picks this from git then all microservices will pick the url for eurekaInstance from ConfigServer .

For this we need to configure our local machine then only eureka server runs in multiple port :

C:\Windows\System32\drivers\etc\hosts file and add the below configuration setup. This will allow you to access peer-1-server.com , peer-2-server.com and peer-3-server.com host names in your local machine.

127.0.0.1 peer-1-server.com  
127.0.0.1 peer-2-server.com  
127.0.0.1 peer-3-server.com

Q .How to make specific properties file in git for each single microsservices ?

Here : Each micro services pick the common properties from application.properties which is already picked from git by configServer application . And Specific properties will pick as microservices name matches to properties file created in git which was already picked by configServer application ( Meaning if PersonMS is microservces then it will pick only PersonMS.properties )

FINALLY:

ConfigServer will pick the properties from git by the help of git url configured in ConfgServer >>>

Then Each individual microservice will get properties from configserver by the help of configserverUrl in application properties file of microserivices >>> so whatever properties we provide in git all the properties will be used by microservices >> we can give url of ErukeaServer in git and microservice will pick it from configserver and do register to EurekaServer .