boot contains the tomcat to get teh beneft for cloud sevice(PAS)

PAS- plateform as service

boot tomcat contains the application and will be deployed into cloud.

--------------------

spring boot CLI is the command line tool to work with spring boot instead of STS.

@configuration

@EnableAutoConfiguration

@Component scan

has been merged into the @SpringBootApplication

--------------------

CQRS- to be checked

--------------------

if we are using the @RestController, no need to add the @ResponseBody at method level

but in case of only @Controller it is required.

SRP-single responsbility person

Microservices-Monolithic

--------------------

service discovery is the main part to search the services

spring cloud is designe by spring io community

--------------------Spring cloud

service discovery

service Registration//use consul/edureka--wr all the microservices get registered

Load balancer//Ribon

Declarative rest Client

Circuit Breakup/trytrix

cloud security

API gateway//Zuul

--------------------

NETFLIX--good wiht amazon azure

polyglot supprt used for managing the mulitple discoverable service//check

--------------------

consul agent -dev // to start the consul agent from command promt

--------------------

consul memebers // to see all the client in cloud

--------------------

create spring start project.

create the aplicaiton.yml and put hte git uri

now run as spirng boot application and launch the below url

http://localhost:8080/configprops

-----

@SpringBootApplication

@EnableConfigServer

public class SpringsloudconfigserverApplication {

public static void main(String[] args) {

SpringApplication.run(SpringsloudconfigserverApplication.class, args);

}

}

------application.yml

---

spring:

cloud:

config:

server:

git:

uri: https://github.com/krishnamurthypradeep/ConfigData1

searchPaths: ConfigData1

--------------------

consul agent -dev //to start the consul client

--------------------

consule--to centralize the services

configserver --to make the centralize configuration so need to change fr every microservice. ms will consume and update the configuration

microservice--to write your service and register

@EnableDiscoveryClient--it will search and automatically register in consule

confg server also should be multiple instance

--------------------

Load baancer(software is apache,Nginx, HA Proxy or H/W F5,NSX,BigIp)

now load balacer is in s/w side as well.

--------------------

spring cloud is framework desigend for microservice based architecture.

it proive sevice discovery(consul/netfix),load balancer(mention above),communication b/w

service(restclient fin)

--------------------

for messagin use the clound bus

--------------------

@EnableConfigSErver--start the config server and enbale for auto register(user bootstrap.yml)

--------------------

check declarative restclient

--------------------

cleint side load balancer...(its application based LB)...it manage only incoming traffic as multiple instance is runnig

so it will decide that how to manage the incoming request with which instance.

spring cloud netflix Ribbon : it is the client side load blancer

reason could be

some may be unavailabe(faults)

some may be slower

soem may be further away than others

--------------------

check feign client

@EnableDiscoveryClient // for auto register service in cetralized place

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

--------------------

