Notes

Micro service level 1 : Communication and Service Discovery

How do you know that it is spring boot project?

By name of main application.

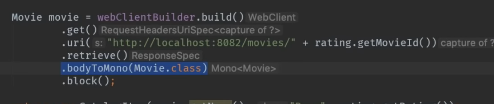
Configure port number?

In application.property add server.port = port\_number



Using WebClient to make API calls

bodyToMono: it means it will give object , it is a reactive way of saying you will get object but not right away(Asynchronous)



15- why you should avoid returning lists in APIs – spring boot microservices

Always return a object t make an API backward compatible.

2nd reason:

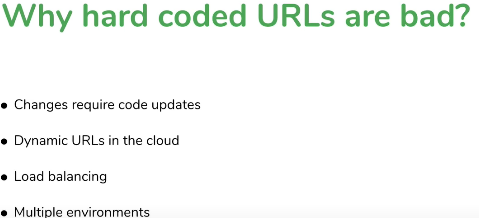
16- Rest template is thread safe

When to use the webclient and when to use rest template ?

Webclient is asynchronous but that supports both

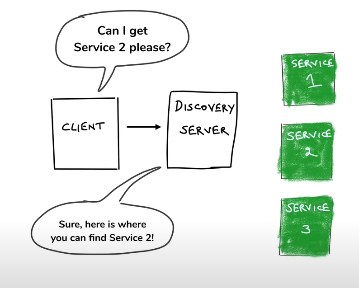
Security can be maintained using either https call or authenticator.

17- Understanding Service Discovery

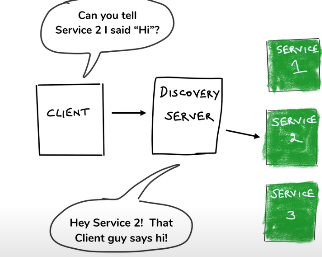


So service discovery comes in the picture.

It discovers all the services which is stored somewhere.



However it requires an extra hop



So difference between first and second picture is that in first one there is extra hop but not in second one.

Spring Cloud uses Client Side discovery

18. Introducing Eureka

Technology to implement the service discovery is Eureka

Netflix OSS have all the library like Eureka, Ribbon, Hysterix, Zuul and it is open source.

Spring Boot /Spring Cloud / Abstraction Layer

19) like spring boot we also create the eureka server

**There are 2 dependency -> whatever application you use to do the service discovery and tell that I am here -> eureka client dependency otherwise what ever application you want use eureka server.**

**So** for this we can have eureka server.

@EnableEurekaServer

If you want to use xml and other features you need to add add the jaxp dependency because in java9 onwards there are no xml and other files.

Eureka also provides an UI. Nevertheless we have to add the client discovery as well.

Every Eureka server is Eureka client and hence they register. So you have to add the property as false



@EnableEurekaCLient add at client level anyway it does not required in the latest versions.

21 Discovering services through Eureka

Use the load balanced annotation. Rest template get the url but it is not really a url rather than hint of the url.

# 22 Doing client side load balancing

# 

# What makes sure is that service is not down it sends I am alive means to send heart beats

# But what happens if service discovery downs , the cache will do the work

# Recap: started multiple microservices -> communicated among each other using hard coded url-> used eureka client to make service discovery and on both side.

# Level2- Microservice springBoot Fault Tolerance and Resilience

# Understand challenges with availability

# Making microservices resilient and fault tolerant

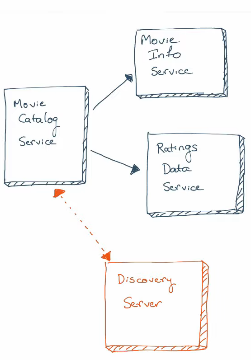
# What is Fault tolerance?

Given an application if there is an fault then how much the system can tolerate it. Is called ….

What is Resilience?

How many fault system can tolerate before it can brought down also where it can bounce back.

The Story so far



Calling an External API call

Movie db

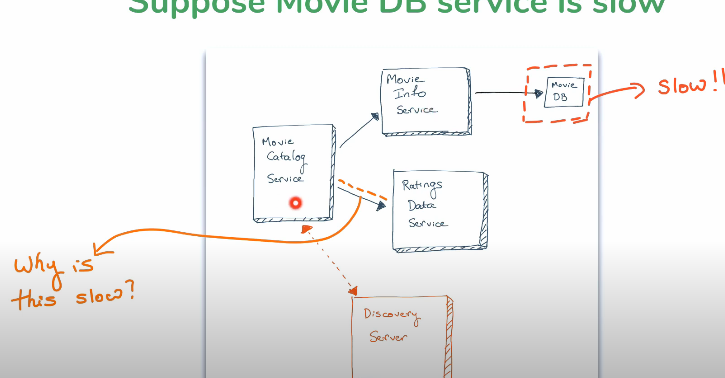
<https://github.com/koushikkothagal/spring-boot-microservices-workshop/commit/8b01c6d153f7b6aee49fa91a5b1c62c1a0c09408>

5) (Issues with Microservices)

what if a microservices goes down instance gets down 🡪 one solution is to have a multiple instances (and it follows round robin algorithm)

2) A microservices instance is slow:

Micro service being slow is a big issues because it makes slow to other instances slow as well

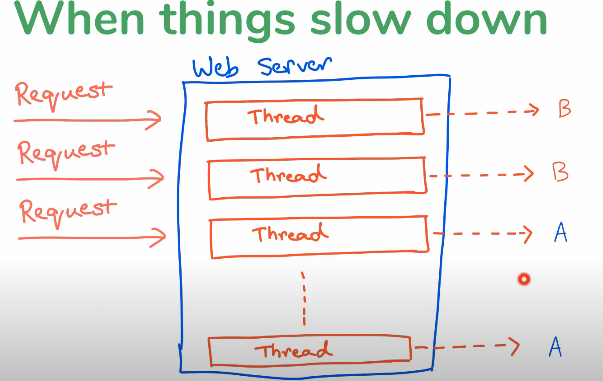


Reason of why is this slow is 🡪 thread

7) How Threads work in a web Server

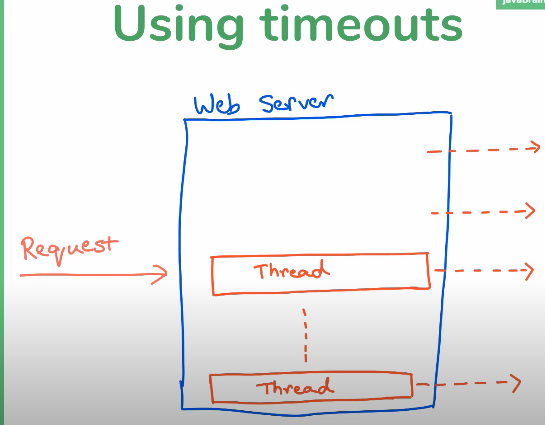
On each request web server creates the thread and it gives the response and gets freed up..

Similarly when instance is slow on each each request new thread is created but thread is still processing and it is active so multiple request makes multiple threads active and it consumes a lot of resources



Let’s there are multiple request for instance B and it is slow and instance A b is fast there are less request as well , but because of multiple request at B and limit of maximum thread is achived then even though Instance A is fast it has to wait for a thread to be freed and hence it makes it slow even though it was free.

So How to solve this problem : 🡪 TimeOUt



Setting timeout using restemplate

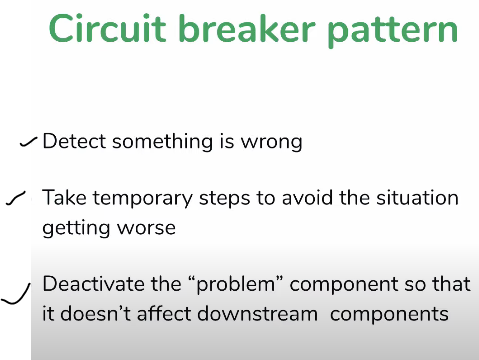


Is this solves the problem completely?

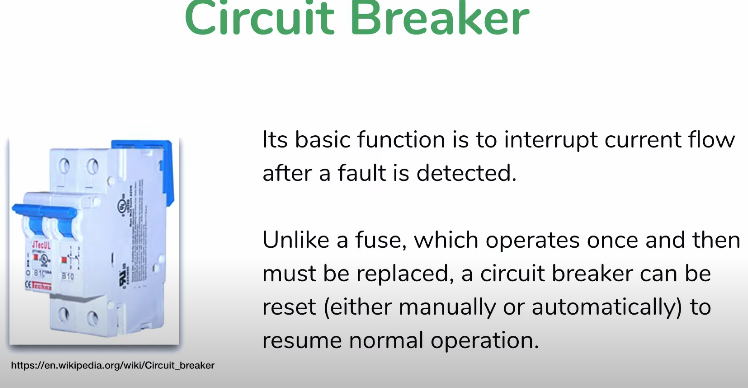
No it solves the problem when request comes in slow frequency 🡪 there are issues for shorter period say 3 but but …… same it makes a larger at some time ,

Say timeout is 3 secs but there are 3(frequency of request is larger than 🡨 time out ) more request in next 3 seconds so eventually there will be again a lot of threads that makes it consume all and same problem again.

11 Understanding the circuit breaker point.(one solution)



Example of circuit breaker

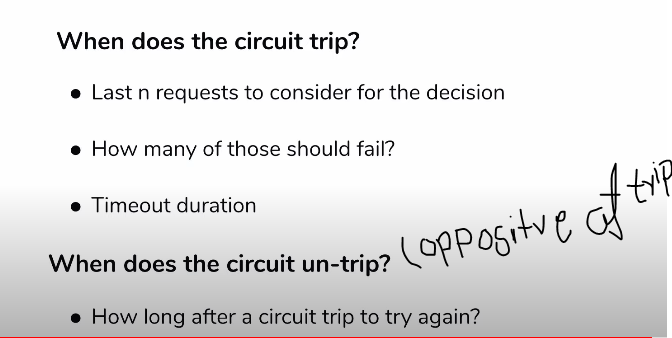


12) Circuit breaker Parameter

Before talking about what to do after circuit break lets understand when the circuit break like for a current,

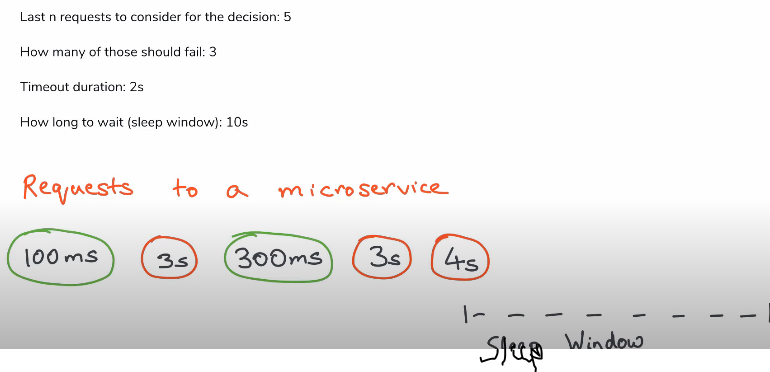
So circuit breaker parameters in microservices

We can assume one way let’s look last three request if all of them are timeout I will beak the circuit but that is not always possible , what if it runs alternatively like one is successful other is timeout so you will never be able to break the circuit.



Untrip -> bounce back

Example



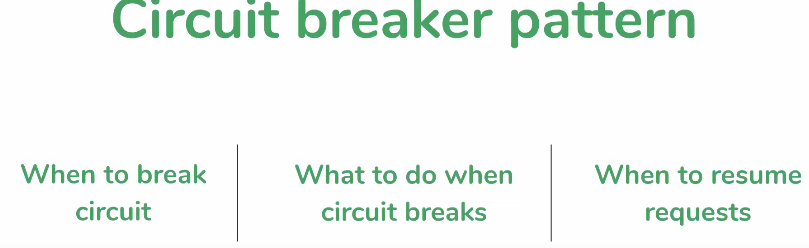
13) What to do when a circuit breaks

* Throw an error (not recommended)
* A fallback default response(partially recommended)
* Save previous responses(cache) and use that when possible (strongly recommended)

14) Circuit breaker benefits

Why we want circuit

1. Failing fast
2. Fallback functionality
3. Automatic recovery



(to implement the circuit we need concurrency programming and thread programming ) instead of implementing manually we can use Hystrix framework

15) what is Hystrix

Open source library originally created by Netflix

Implements circuit breaker pattern so you don’t have to

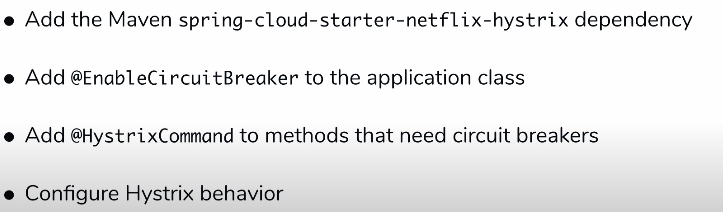
Give it the configuration params and it does the work

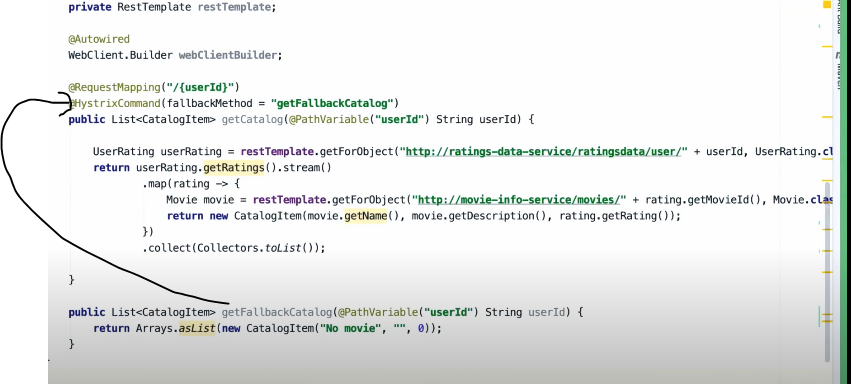
Works Well with spring boot

Even Netflix doesn’t use it directly they use it with spring cloud but bad news is this is maintained only and they are not being developed

, so there are are patterns like adapt fault tolerance

16) Adding Hystrix to a Spring boot microservice





Its always better to write a hard coded value in a fallback method because you don’t want to call another url which also needs a fallaback this will become never ending loop

How does this work?

We just did few annotation and it worked but how it really work?

It wrap ups in a proxy class and it calls an actual method. And it constantly checks if everything is going correct.

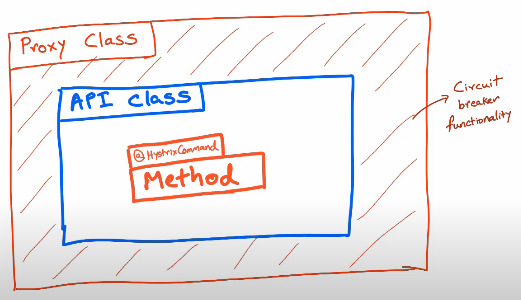
18) problem with Hystrix proxy

When two url is called in one method or method is within the class it will create a problem in fall back.

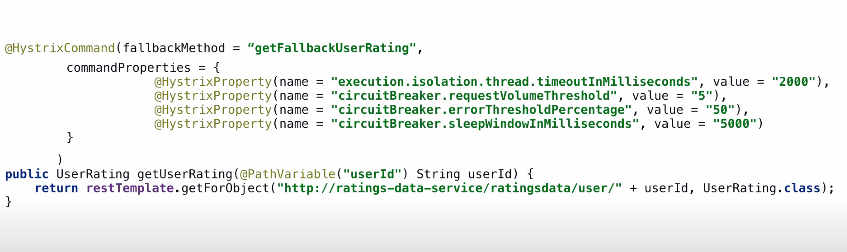
So to get out of this problem we should implement it in a different method basically it should have different bean.

19) refactoring the code at granular level.

We also solve a problem when one service calls another service by writing different method and fallback for each.



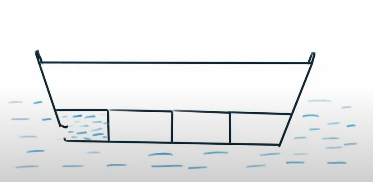
20) Configuring Hystix parameters



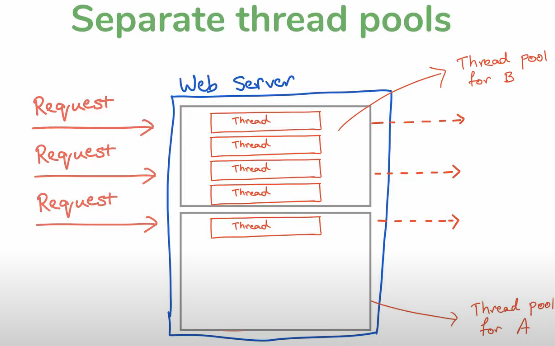
21) Hystrix Dashboard

22) Bulk head Pattern

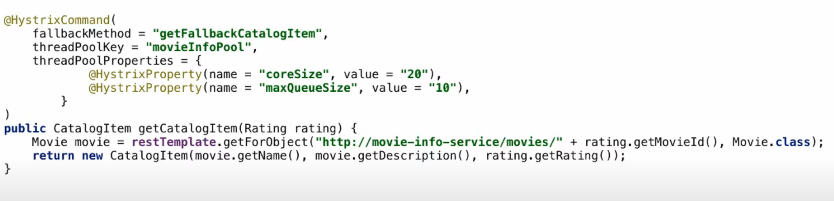
Third way of handling circuit breaker. The names comes with ship building architecture where there are chances of of being hole in the ship so it is built with water compartments.



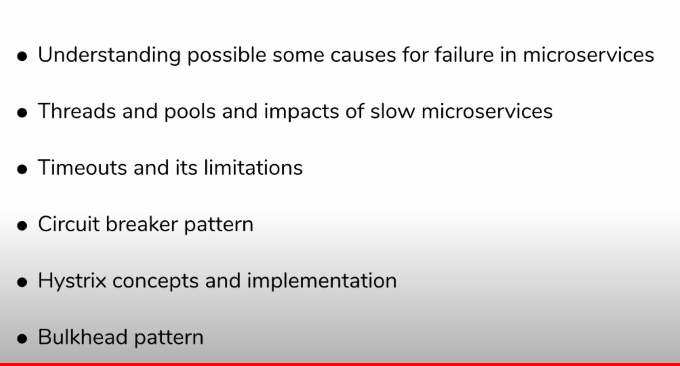
So same concept we can also apply in microservices be creating separate thread pools so none of them will share so we can configure the thread per instance.



Configure



Summary



Level 3 🡪 spring boot configuration and management



2. Microservice configuration what and why.

Configuration is essential for any production database and keep it aside from jar.

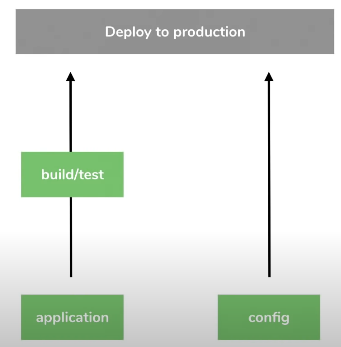
Example of configuration

Database connections , Credentials , feature flags

Business Logic Configuration parameters(mostly handled by database)

Scenario testing

Spring boot configuration



This can be achieved by configuration management.

Config files: properties,yaml,json

Why not to have in jar

Multiple microservices , multiple instances of one microservice

Goals

Externalized

Environment specific

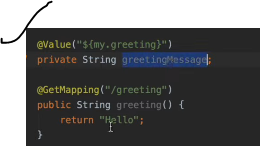
Consistent

Version history

Real -time management

3) Using property file config with spring boot.

@value



To refer the value of one property in another property inside property file



4) Accessing External Properties in Spring boot

so if we want to change anything in the application.properites file then we need to deploy the jar again even though there is not a single line of code.

* We can overwrite using external application property.

5) @Value annotation tricks you should know

@value(“we can assign this value directly to this variable or we can also read from property”)

Private String variable;

@Value(“${my.property : default value}”)

Private String variable;

@Value(“${my.list.values}”)//my.list.values=one,two,three

Private List<String> listofvalue;

\*\*it also works with key values pair

@Value(“#{${dbValues}}”) // it is going to evaluate the db values string which is declared in spring boot

Private Map<String,String> dbValues;

6)Configuration properties

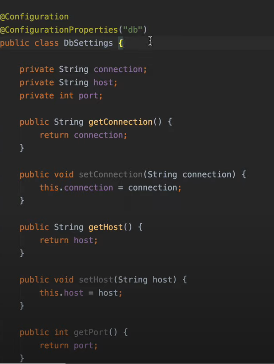
Instead of applying properties one by one we can generate as bean

Let’s say we have these bunch of properties in property file.



Create a class with dB Setting, all values which matches with db. Wil be read by particular string.

Add the @Configuration so that it will know it has to create a bean.



Reason : we get to group to all values and bundle as a on jar,

It has spring bean : so we can autowire this anywhere and use it at multiple places

Spring boot Actuator: it is exposes one end point which tell where is all the property.

Management.endpoints.web.exposre.include = \*(expose everything)

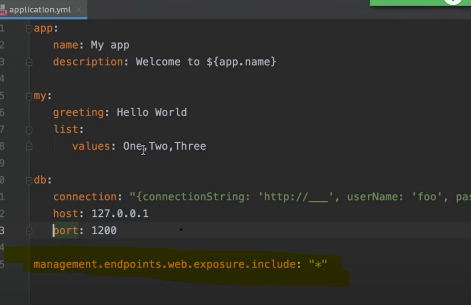
7) what is Yaml and what is the difference.

Using Yaml for config files.

It uses key value pair , real benefit of using yaml is in nested structure.

We can also use both types in yaml (instead of intending we can use directly if not required)

Have consistent intend form.



8) Spring profiles

Spring profiles help to assign the property as the per the environment

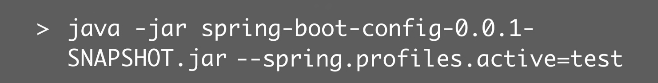
Application-prfilename.extension(property/yaml)

Default properties is always active irrespective of

Test,acc,prod

If both are active the test will overwrite the default(with combination of other data)

While deploying we can tell deploy using this profile



Selecting beans by profile:

We can instantiate the bean based on environment or profile based instantiation.

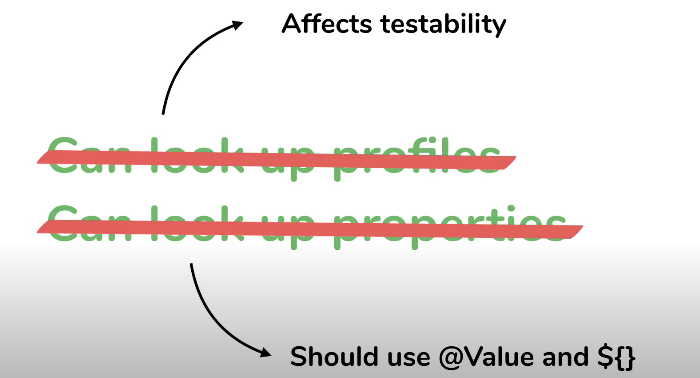


9) Using environment object(what is environment object)

Envirnoment is a bean you can inject and load

We can lookup the active profiles

Create an end point with /envdetails



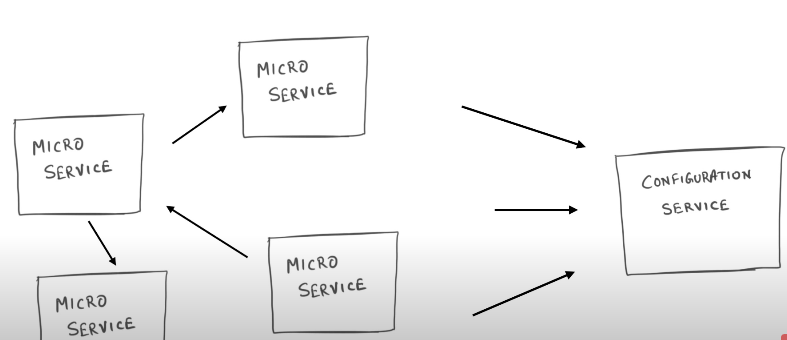
This is something which we should avoid using.

10) Spring Cloud Config Server,

From single service to multiple microservices

Eco system of microservices application so the goal is to make to it consistent

* Consistency in configuration : important with microservices



Config as a Service(one source which provides the configs to all the microservices)

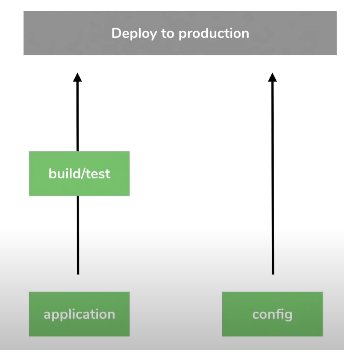
Options :

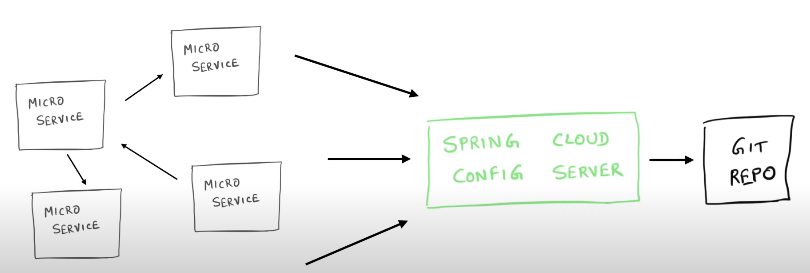
Apache Zookeeper

Etcd- distributed key -value store

Hashicorp consul

Spring cloud configuration server





11) Setup Spring Cloud Config server

* Create new spring boot project for spring-cloud-config-server
* Add at least one dependency config server

What’s the url for cloud config.

Localhost/filename/profile

Spring cloud config server will take the values from git and other microservices can talk with spring cloud config server

12)Creating Spring Cloud Config client(which will connect to spring cloud config server

1.. add the dependency of springcloud starter config server

2.. add in the management

3. add the url of cloud config server in application.properties

There is onr problem:

We want to externalize the port numbers but all the microservices should have different port number

Solution: create a property file per microservices and as soon as particular microservices called the respective file will be used

Spring.application.name: spring-boot-config

13) Refreshing Properties run time(dynamic config with spring boot)

Spring cloud always looks to git repo hence it is up to date. But microservices doesn’t up to date. Because to update it takes a resources . so we have to restart the microservices but it is not good idea.

So have a

* spring actuator dependency(it provides the end point which when called) so there is an point which can be called for the refreshed
* Marks the classes, beans that needs to be refreshed
* Add the @RefreshScope(in the bean to get the property file)
* Post the request as refresh end point

14) Configuration Strategies

Spring security can be used to secure any spring boot project

The Twelve-Factor APP –website