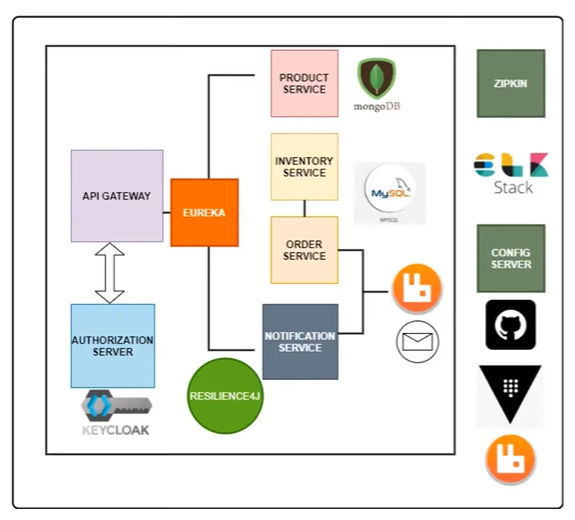
Online Shopping



Zipkin – Distributed Tracing

ELK – Logging

Resilience4J –

Ports-

Product-service : 9001

Lombok, Spring Web, Spring Data Mongo DB

1. Product Service Class – in Model Package

* Uses Mongo DB – configure that in Application.properties
* Model > Product – Add Lombok properties for Data/Builder/NoArgs/AllArgs constructor
* @Document(value="product") – MongoDB document linkage

1. ProductRepository interface – extends MongoDBRepository
2. ProductController –

@RequiredArgsConstructor – Creates argument constructor for the class

**private** **final** ProductRepository productRepository;

//Why is it not autowired

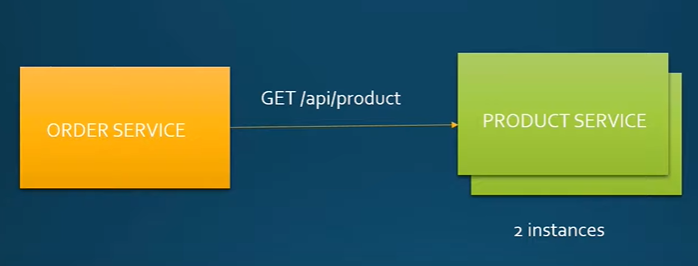
npx kill-port 8080 – to kill what’s running on 8080

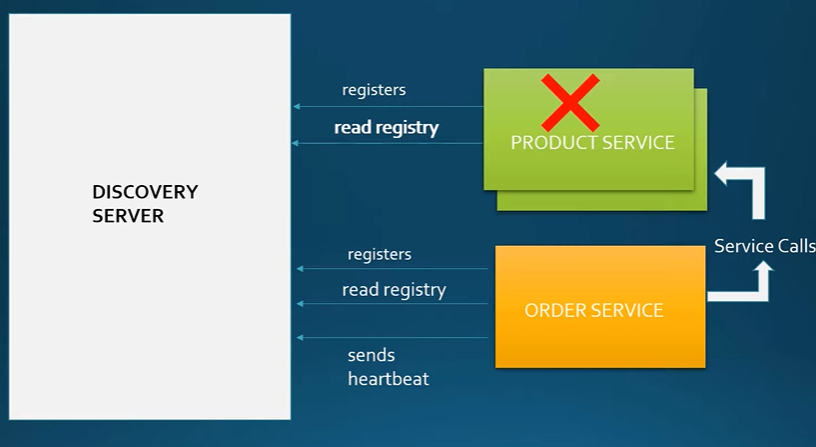
Order-Service : 9002

Lombok, SpringWeb, Spring Data JPA, MySQL Driver

create database `order-service`; - use backticks

Service Discovery : Automatic detection of all available services.



Service Registry – Process of services registering to discovery server

Services keep a local cache of registry. Do some client side load balancing in case of multiple instances

If no heartbeat – client is down

**Eureka-Service**

Eureka CServer Dependency

@EnableEurekaServer

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

Local copy of registry not needed

@EnableEurekaClient – For the clients

spring.application.name=order-service

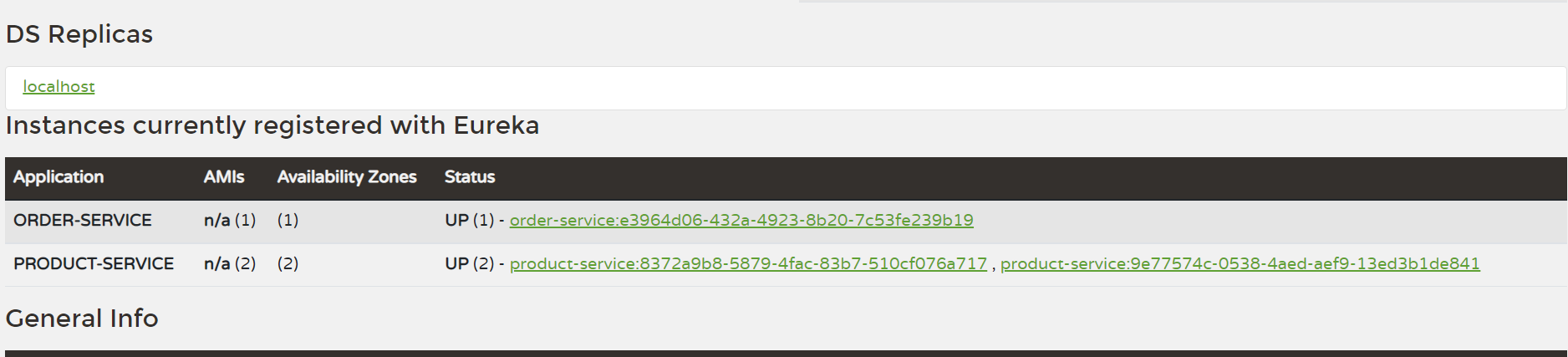
Eureka Client Dependency

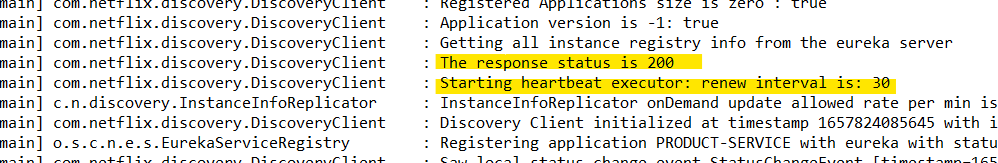
For Product Service:

server.port=0 – To assign a port number dynamically

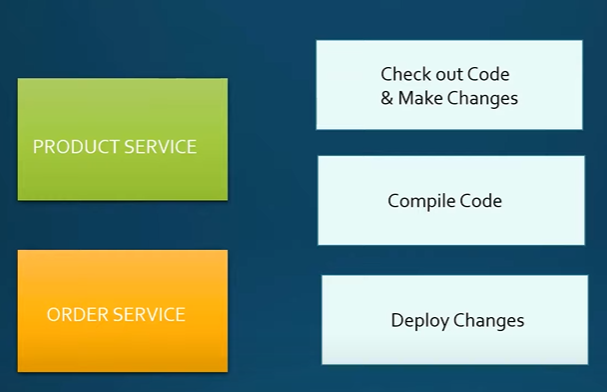
-We would be creating multiple instances for this

eureka.instance.instance-id=${spring.application.name}:${random.uuid}

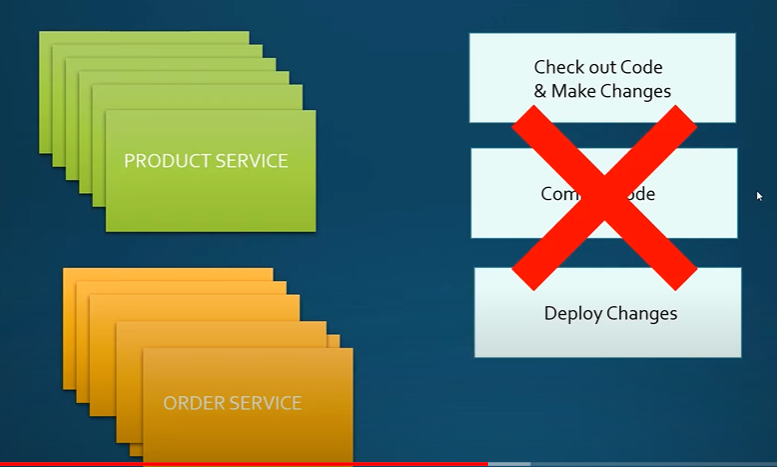




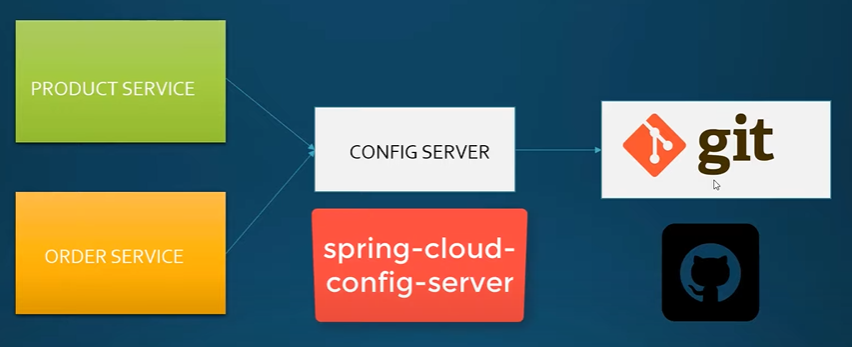
**Centralized Configuration**

****

**Config changes difficult for multiple instances**

****

**Dynamically load changes instead of restarting the instances.**

****

**Can store config in git repo and native files**

@EnableConfigServer

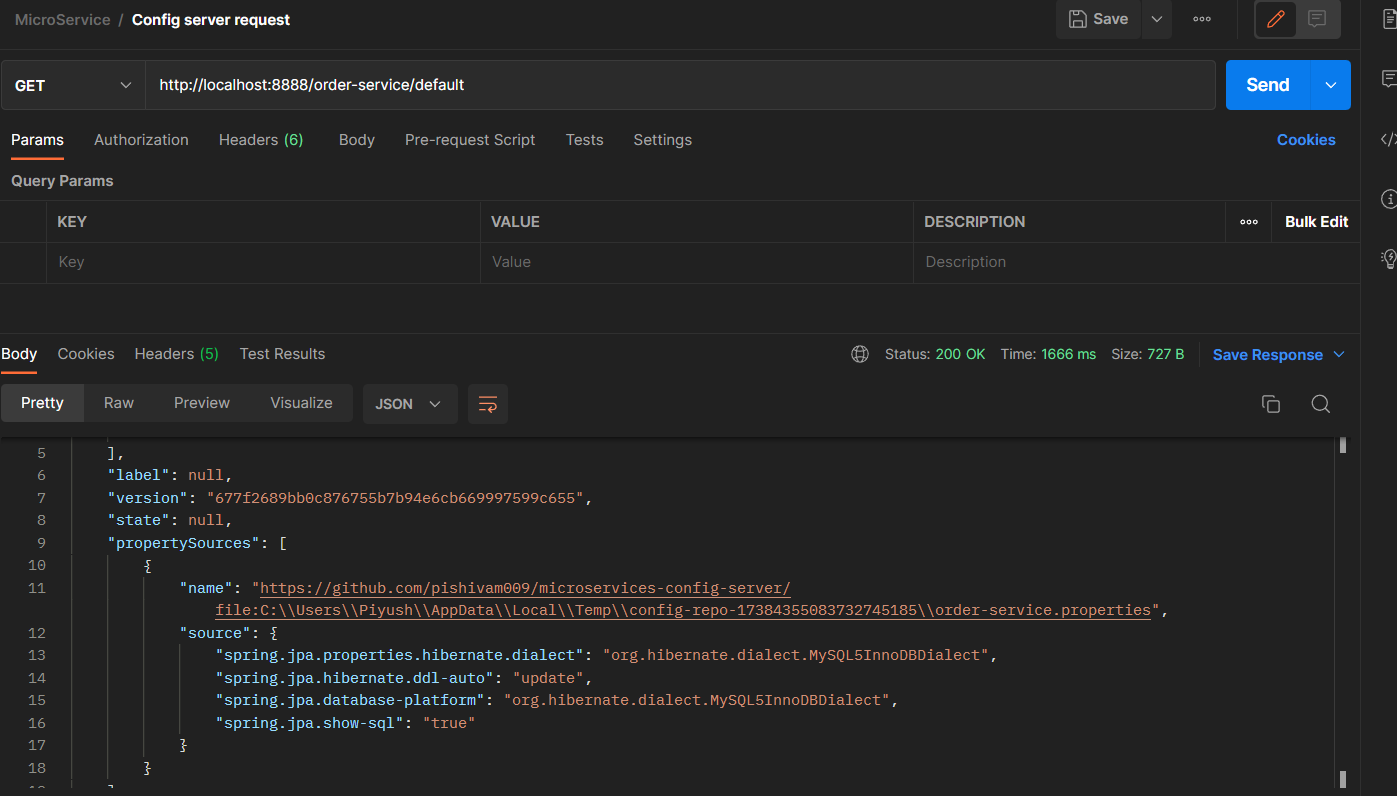
SpringCloud ConfigServer , Spring Actuator

Properties moved to new git repo

spring.cloud.config.server.git.uri=https://github.com/pishivam009/microservices-config-server

spring.cloud.config.server.git.clone-on-start=true

spring.cloud.config.server.git.default-label=master



Order-service -> Name of the properties file

Default- profile name

Prod>order-service.properties

Access > order-service/prod

Add Config-client dependency to Order Service and Product Service

Cloud-starter-bootstrap – legacy

New: <https://docs.spring.io/spring-cloud-config/docs/3.0.0/reference/html/#config-data-import>

Refresh Config

@RefreshScope - Beans annotated this way can be refreshed at runtime and any components that are using them will get a new instance on the next method call, fully initialized and injected with all dependencies.

Change test.name in product-service

Call /actuator/refresh

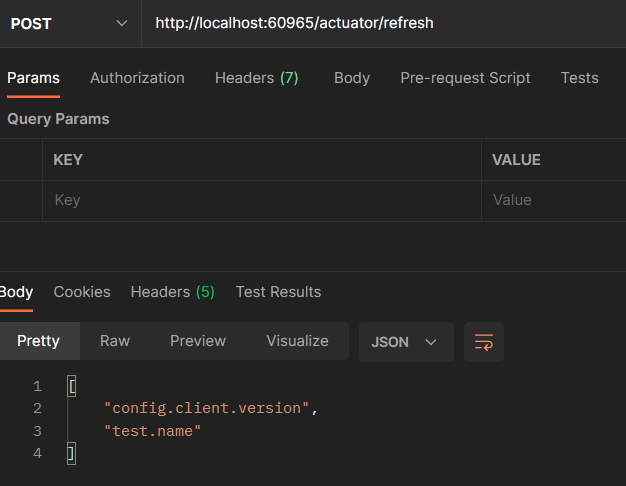
Add actuator dependency to product/Order service. That is needed to trigger config refresh

<https://stackoverflow.com/questions/61139281/spring-cloud-config-client-not-picking-values-from-config-server>

One Extra Space!!! – Stupid

Add management.endpoints.web.exposure.include=\* in product/order service for refresh

Test.name value is refreshed.



Store secrets in vault.

choco install vault

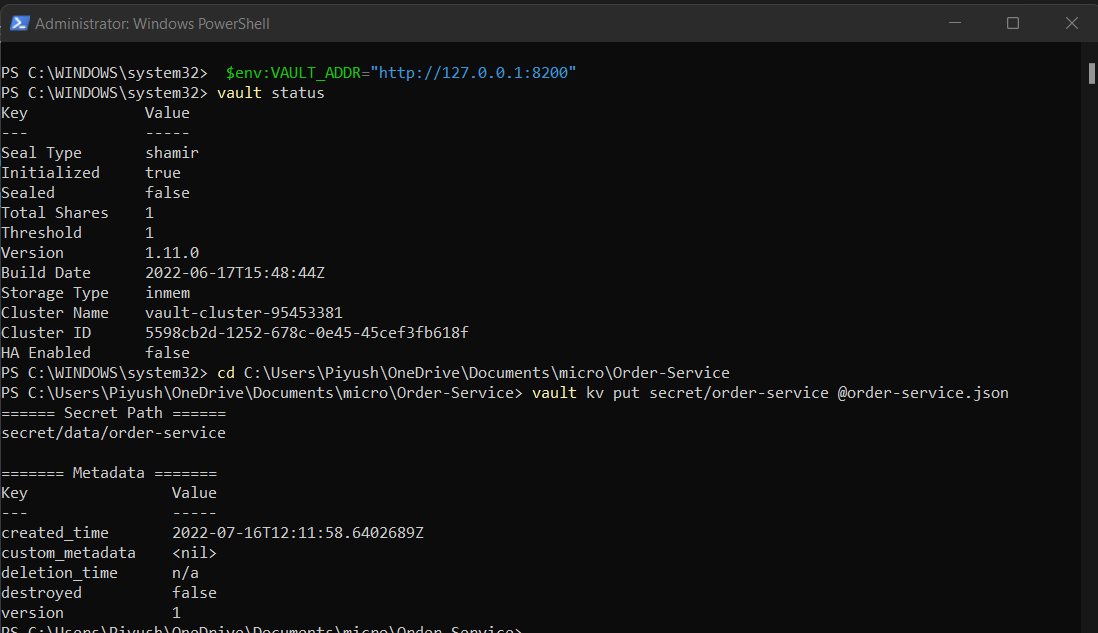
vault server –dev

set VAULT\_ADDR=http://127.0.0.1:8200

vault kv put secret/order-service @order-service.json

Having issues. Omitting for now. Vault can be integrated along with config server too.

https://www.youtube.com/watch?v=p65u4t26BBc



vault kv get secret/order-service

Same for product-service

Add spring cloud vault dependency