si es primera vez, hacer:

git clone https://github.com/javidep/curso-microservicios.git

## Escalar servicios con docker

|  |  |
| --- | --- |
| 1 | modificar docker-compose para escalar servicios  docker/common/docker-compose.yml |
| 2 | ejecutar nuevamente los servicios "dockerizados"  mvn clean package docker:build  docker-compose -f docker/common/docker-compose.yml up --scale licensingservice=2  verificar que no haya errores |
| 3 | verificar en que puertos están escuchando las instancias de los servicios.  Docker ps |
| 4 | ejecutar  Get http://localhost:xxxx/env  en cada puerto del servicio licensing |
|  | Refrescar propiedad en git |
| 5 | ejecutar en una de las instancias  post http://localhost:xxxx/refresh |
| 6 | verificar el cambio  Get http://localhost:xxxx/env |
|  | verificar la otra instancia  Get http://localhost:yyyy/env |

## configuracion service discovery

|  |  |
| --- | --- |
| 1 | Bajar fuentes  git pull  verificar que se hayan creado los objetos en  curso-microservicios/sesion3 |
| 2 | verificar instalacion de eclipse |
| 3 | verificar eurekasvr |
| 4 | construir imagenes docker  cd sesion3  mvn clean package docker:build |
| 6 | iniciar contenedores  docker-compose -f docker/common/docker-compose.yml up |
| 7 | Acceder a  http://localhost:8761/ |
| 8 | incluir organizationservice y registrarlo  modificar   * pom.xml padre .   descomentar organization  <modules>  ..  <module>organization-service</module>  ..  </modules>   * docker/common/docker-compose.yml   descomentar organization  ..  organizationservice:  image: johncarnell/tmx-organization-service:chapter4  ports:  - "8085:8085"  environment:  PROFILE: "default"  SERVER\_PORT: "8085"  CONFIGSERVER\_URI: "http://configserver:8888"  EUREKASERVER\_URI: "http://eurekaserver:8761/eureka/"  EUREKASERVER\_PORT: "8761"  CONFIGSERVER\_PORT: "8888"  DATABASESERVER\_PORT: "5432"  ENCRYPT\_KEY: "IMSYMMETRIC"  .. |
| 9 | Construir imagenes  mvn clean package docker:build |
| 10 | iniciar contenedores  docker-compose -f docker/common/docker-compose.yml up |
| 11 | ir a consola eureka  http://localhost:8761/ |
| 17 | registrar licensing y organization2 en eureka   * licensing-service/src/main/java/com/thoughtmechanix/licenses/Application.java   descomentar @EnableEurekaClient   * licensing-service/src/main/resources/application.yml   eureka:  instance:  preferIpAddress: true  client:  registerWithEureka: true  fetchRegistry: true  serviceUrl:  defaultZone: http://localhost:8761/eureka/   * licensing-service/pom.xml   <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-eureka</artifactId>  </dependency>   * docker/common/docker-compose.yml   organizationservice2:  image: johncarnell/tmx-organization-service:chapter4  ports:  - "8086:8086"  environment:  PROFILE: "default"  SERVER\_PORT: "8086"  CONFIGSERVER\_URI: "http://configserver:8888"  EUREKASERVER\_URI: "http://eurekaserver:8761/eureka/"  EUREKASERVER\_PORT: "8761"  CONFIGSERVER\_PORT: "8888"  DATABASESERVER\_PORT: "5432"  ENCRYPT\_KEY: "IMSYMMETRIC" |
| 18 | Desplegar cambios  mvn clean package docker:build  docker-compose -f docker/common/docker-compose.yml up |
| 19 | ir a consola eureka  http://localhost:8761 |

## comunicacion entre servicios

|  |  |
| --- | --- |
| 22 | consultar licensing usando cliente Service Discovery  get http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a/discovery |
| 25 | consultar licensing usando restclient  http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a/rest |
| 26 | Invocar multiples veces para verificar funcionamiento del balanceo de carga ribbon  http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a/rest |
|  | importar licensing a eclipse(como proyecto maven) |
| 27 | Modificar para usar Netflix feign   * licensing-service/src/main/java/com/thoughtmechanix/licenses/Application.java.   Agregar anotación @EnableFeignClients   * licensing-service/src/main/java/com/thoughtmechanix/licenses/services/LicenseService.java   modificar retrieveorginfo y agregar bean  …  @Autowired  OrganizationFeignClient organizationFeignClient;  …  case "feign":  System.out.println("I am using the feign client");  organization = organizationFeignClient.getOrganization(organizationId);  break;  … |
| 28 | desplegar cambios en licensing  mvn clean package docker:build -f licensing-service/pom.xml  docker-compose -f docker/common/docker-compose.yml up licensingservice |
| 29 | consultar licencia con informacion de organization usando feign  http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a/feign |

|  |  |
| --- | --- |
| Alta disponibilidad de config server | |
| 1 | Agregar nueva instancia de configserver   * docker/common/docker-compose.yml   agregar  configserver2:  image: johncarnell/tmx-confsvr:chapter4  ports:  - "8889:8889"  links:  - "eurekaserver:eurekaserver"  environment:  EUREKASERVER\_URI: "http://eurekaserver:8761/eureka/"  EUREKASERVER\_PORT: "8761"  ENCRYPT\_KEY: "IMSYMMETRIC" |
| 2 | Registrar confsvr a eureka  Modificar   * confsvr/src/main/resources/application.yml   agregar  …  eureka:  instance:  preferIpAddress: true  client:  registerWithEureka: true  fetchRegistry: true  serviceUrl:  defaultZone: http://localhost:8761/eureka/ …   * confsvr/src/main/java/com/thoughtmechanix/confsvr/ConfigServerApplication.java   agregar @ EnableEurekaClient   * licensing-service/src/main/resources/bootstrap.yml   agregar  cloud:  config:  …  discovery:  enabled: true   * licensing-service/src/main/docker/run.sh   + comentar líneas 16   + eliminar línea 25 |
| 3 | desplegar cambios  mvn clean package docker:build  docker-compose -f docker/common/docker-compose.yml up |

Hystrix

|  |  |
| --- | --- |
| 1 | agregar hystrix a licensing  modificar   * licensing-service/pom.xml   agregar dependencias  …  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-hystrix</artifactId>  </dependency>  <dependency>  <groupId>com.netflix.hystrix</groupId>  <artifactId>hystrix-javanica</artifactId>  <version>1.5.9</version>  </dependency>  …   * licensing-service/src/main/java/com/thoughtmechanix/licenses/Application.java   agregar @ EnableCircuitBreaker   * licensing-service/src/main/java/com/thoughtmechanix/licenses/services/LicenseService.java * agregar @HystrixCommand |
| 2 | desplegar cambios  mvn clean package docker:build  docker-compose -f docker/common/docker-compose.yml up |
|  | * http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/ * http://localhost:8080/v1/organizations/e254f8c-c442-4ebe-a82a-e2fc1d1ff78a/licenses/f3831f8c-c338-4ebe-a82a-e2fc1d1ff78a |