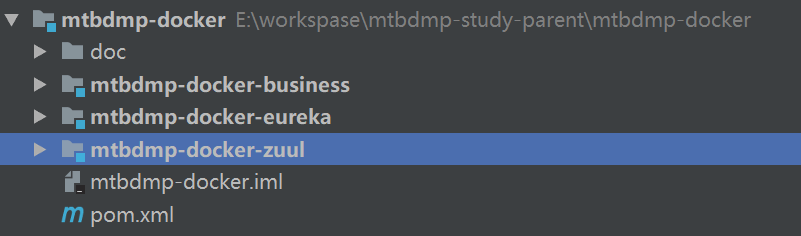
使用Docker部署

Spring Cloud微服务

# 手动将Spring Cloud 微服务打包镜像

## 准备阶段

我们准备了一个Eureka项目(服务注册发现中心)mtbdmp-docker-eureka，业务项目(模拟业务) mtbdmp-docker-business，zuul（网关路由）mtbdmp-docker-zuul,项目结构如下：



|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <modelVersion>4.0.0</modelVersion>  <groupId>com.cmaotai.docker</groupId>  <artifactId>mtbdmp-docker</artifactId>  <packaging>pom</packaging>  <version>1.0-SNAPSHOT</version>  <modules>  <module>mtbdmp-docker-eureka</module>  <module>mtbdmp-docker-zuul</module>  <module>mtbdmp-docker-business</module>  </modules>   <properties>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  <java.version>1.8</java.version>  <docker.image.prefix>mtbdmp</docker.image.prefix><!--配置镜像仓库的属性-->  </properties>   <!-- 引入spring boot的依赖 -->  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>2.0.1.RELEASE</version>  </parent>   <dependencies>  <!-- 加入spring-boot 关于web的依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  </dependencies>   <dependencyManagement>  <dependencies>  <!-- 统一依赖管理 -->  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-dependencies</artifactId>  <version>Finchley.RELEASE</version>  <type>pom</type>  <scope>import</scope>  </dependency>  </dependencies>  </dependencyManagement>   <!-- 添加spring-boot的maven插件 -->  <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  </plugin>  </plugins>  </build>  <repositories>  <repository>  <id>spring-snapshots</id>  <name>Spring Snapshots</name>  <url>https://repo.spring.io/libs-snapshot</url>  <snapshots>  <enabled>true</enabled>  </snapshots>  </repository>  </repositories>  </project> |

下面先分别来看一下每个模块中的配置文件和启动类

### mtbdmp-docker-eureka

配置文件application.yml：

|  |
| --- |
| **spring**:  **application**:  **name**: eureka7001 **server**:  **port**: 7001 **eureka**:  **client**:  **register-with-eureka**: **false** *#false:不作为一个客户端注册到注册中心* **fetch-registry**: **false** *#意思是不作为微服务注册到eureka中，默认为true  #为true时，可以启动，但报异常：Cannot execute request on any known server* **service-url**:  **defaultZone**: http://192.168.3.233:7001/eureka *#默认注册到8761端口，不写的话会报错* **instance**:  **prefer-ip-address**: **true** *#表示微服务显示IP* |

启动类：

|  |
| --- |
| package com.cmaotai.mtbdmp.docker;  import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;  @SpringBootApplication @EnableEurekaServer public class EurekaApplication {  public static void main(String[] args) {  SpringApplication.*run*(EurekaApplication.class,args);  } } |

pom.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <parent>  <artifactId>mtbdmp-docker</artifactId>  <groupId>com.cmaotai.docker</groupId>  <version>1.0-SNAPSHOT</version>  </parent>  <modelVersion>4.0.0</modelVersion>   <artifactId>mtbdmp-docker-eureka</artifactId>   <dependencies>  <!-- 加入spring-boot 关于web的依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <!--将此服务注册到eureka-->  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  </dependency>  </dependencies>  </project> |

### mtbdmp-docker-business

配置文件application.yml：

|  |
| --- |
| **spring**:  **application**:  **name**: buiness8001 *#微服务调用时的名字* **server**:  **port**: 8001 **eureka**:  **client**:  **service-url**:  **defaultZone**: http://eureka7001:7001/eureka/ *#Eureka的地址* **instance**:  **instance-id**: business8001 *#注册到eureka中的名字* **prefer-ip-address**: **true** *#表示微服务显示IP* |

这里的defaultZone访问url，决于部署docker容器时的 **--link** 参数， **--link** 可以让两个容器之间互相通信。

启动类：

|  |
| --- |
| package com.cmaotai.mtbdmp.docker;   import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.cloud.netflix.eureka.EnableEurekaClient;  @SpringBootApplication @EnableEurekaClient public class BusinessApplication {   public static void main(String[] args) {  SpringApplication.*run*(BusinessApplication.class,args);  } } |

pom.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">   <modelVersion>4.0.0</modelVersion>  <version>1.0-SNAPSHOT</version>  <artifactId>mtbdmp-docker-business</artifactId>  <packaging>jar</packaging>   <parent>  <artifactId>mtbdmp-docker</artifactId>  <groupId>com.cmaotai.docker</groupId>  <version>1.0-SNAPSHOT</version>  </parent>  <dependencies>  <!-- 加入spring-boot 关于web的依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-eureka-server</artifactId>  <version>RELEASE</version>  </dependency>   </dependencies>   <dependencyManagement>  <dependencies>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-dependencies</artifactId>  <version>Finchley.RELEASE</version>  <type>pom</type>  <scope>import</scope>  </dependency>  </dependencies>  </dependencyManagement>  </project> |

业务类：

|  |
| --- |
| package com.cmaotai.mtbdmp.docker.controller;  import org.springframework.web.bind.annotation.PathVariable; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController; @RestController public class BusinessController {   @RequestMapping("/business/{info}")  public String test(@PathVariable(value="info") String info) throws Exception {  return info;  } } |

### mtbdmp-docker-zuul

配置文件application.yml：

|  |
| --- |
| **spring**:  **application**:  **name**: zuul9001 *#微服务调用时的名字* **server**:  **port**: 9001 **eureka**:  **client**:  **service-url**:  **defaultZone**: http://eureka7001:7001/eureka/ *#Eureka的地址* **instance**:  **instance-id**: zuul9001 *#注册到eureka中的名字* **prefer-ip-address**: **true** *#表示微服务显示IP* **zuul**:  **ignored-services**: "\*" *#表示忽略所有微服务的名字，通过下面匹配的路径去访问* **prefix**: /api *#为zuul设置一个公共的前缀* **routes**:  **test**:  **path**: /business/\*\* *#表示拦截/api/business/的所有请求* **service-id**: buiness8001 *#对应business微服务中的spring.application.name ,这句很重要，不配置或者配置错误，通过zuul访问服务提供者将失败* |

这里的defaultZone访问url，决于部署docker容器时的 **--link** 参数， **--link** 可以让两个容器之间互相通信。

启动类：

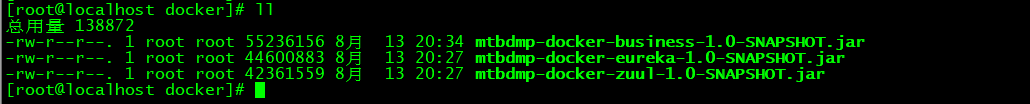
|  |
| --- |
| package com.cmaotai.mtbdmp.docker;  import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.cloud.client.discovery.EnableDiscoveryClient; import org.springframework.cloud.netflix.eureka.EnableEurekaClient; import org.springframework.cloud.netflix.zuul.EnableZuulProxy;  @SpringBootApplication @EnableEurekaClient @EnableZuulProxy @EnableDiscoveryClient public class ZuulApplication {   public static void main(String[] args) {  SpringApplication.*run*(ZuulApplication.class,args);  } } |

pom.xml

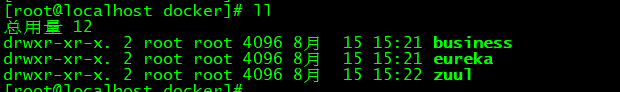
|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  <parent>  <artifactId>mtbdmp-docker</artifactId>  <groupId>com.cmaotai.docker</groupId>  <version>1.0-SNAPSHOT</version>  </parent>  <modelVersion>4.0.0</modelVersion>   <artifactId>mtbdmp-docker-zuul</artifactId>   <dependencies>  <!-- 加入spring-boot 关于web的依赖 -->  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <!--将此服务注册到eureka-->  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-eureka</artifactId>  <version>1.3.4.RELEASE</version>  </dependency>  <dependency>  <groupId>org.springframework.cloud</groupId>  <artifactId>spring-cloud-starter-netflix-zuul</artifactId>  <version>2.0.0.RELEASE</version>  </dependency>  </dependencies>  </project> |

## 上传服务器，编写Dockerfile

将三个微服务打成jar包，并上传到服务器。



创建三个文件夹，将以上三个jar包移动到对应的文件夹



接着在对应的文件夹下编写他们的Dockerfile

### Eureka

|  |
| --- |
| **FROM webapp:1.0**  **VOLUME /tmp**  **ADD mtbdmp-docker-eureka-1.0-SNAPSHOT.jar app.jar**  **EXPOSE 7001**  **ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]** |

### business

|  |
| --- |
| **FROM webapp:1.0**  **VOLUME /tmp**  **ADD mtbdmp-docker-business-1.0-SNAPSHOT.jar app.jar**  **EXPOSE 8001**  **ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]** |

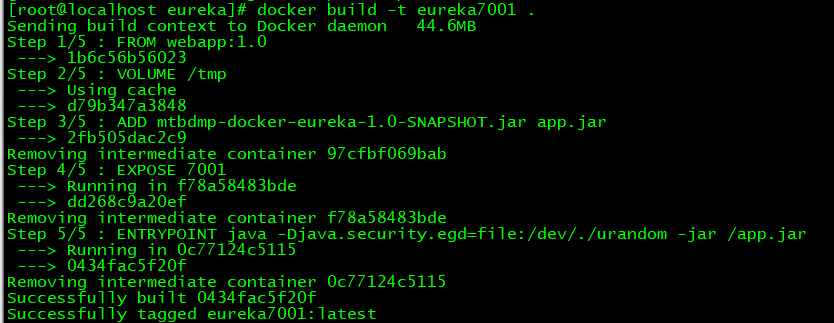
### Zuul

|  |
| --- |
| **FROM webapp:1.0**  **VOLUME /tmp**  **ADD mtbdmp-docker-zuul-1.0-SNAPSHOT.jar app.jar**  **EXPOSE 9001**  **ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]** |

## 分别将三个jar打成镜像

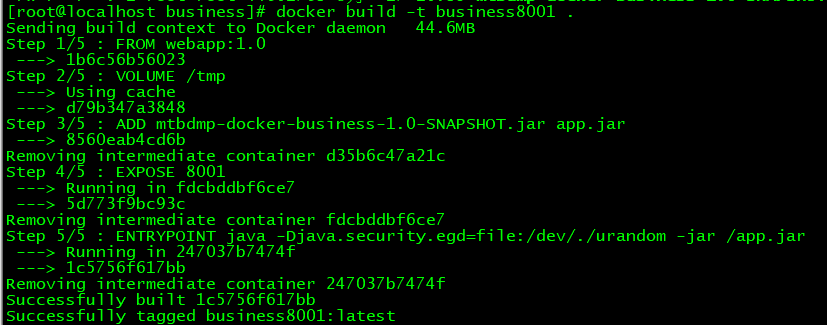
### Eureka

|  |
| --- |
| **[root@localhost eureka]# docker build -t eureka7001 .** |



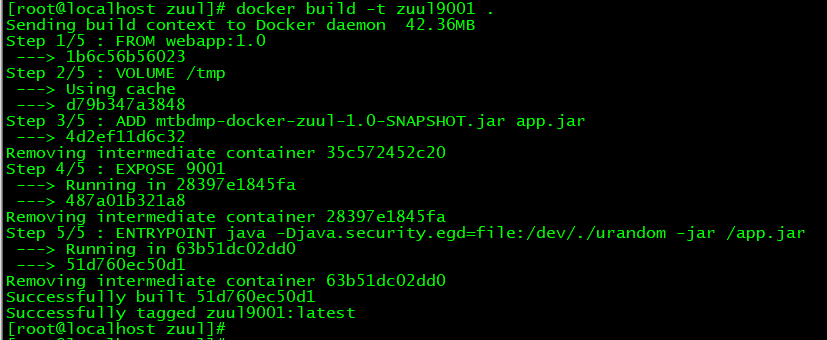
### business

|  |
| --- |
| [root@localhost business]# docker build -t business8001 . |

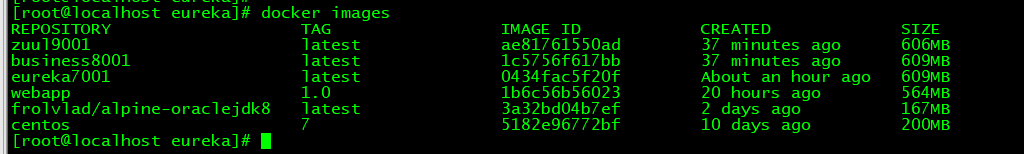


### zuul

|  |
| --- |
| [root@localhost zuul]# docker build -t zuul9001 . |



## 接着运行三个镜像



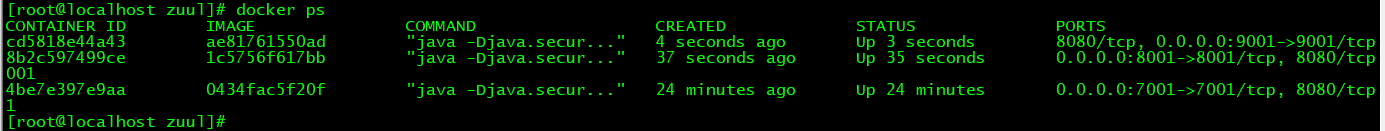
### 运行Eureka容器

|  |
| --- |
| [root@localhost zuul]# docker run -d -p 7001:7001 --name eureka7001 0434fac5f20f |

### 运行Business、Zuul容器

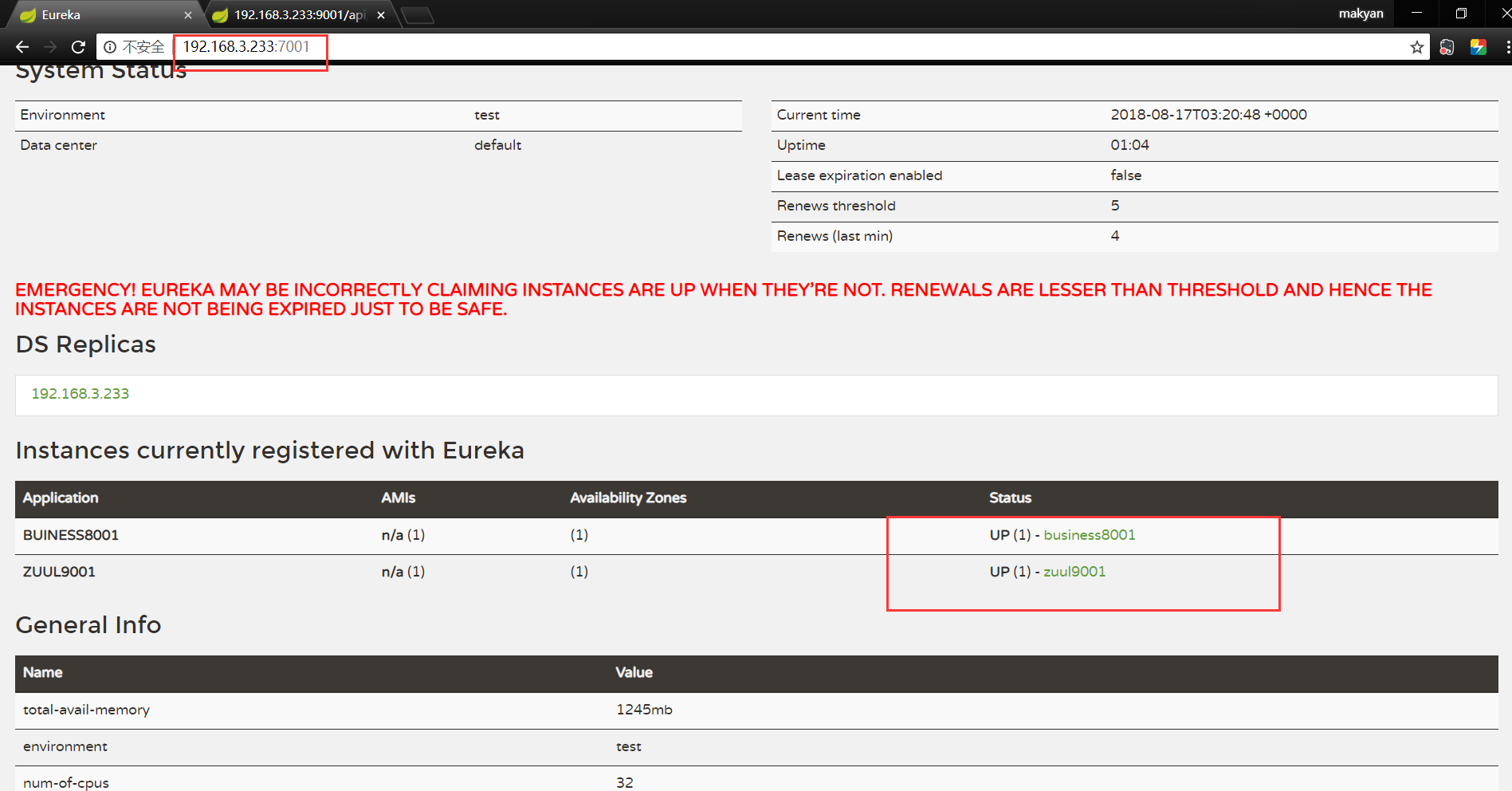
|  |
| --- |
| [root@localhost zuul]# docker run -d -p 8001:8001 --name business8001 --link eureka7001 1c5756f617bb |

|  |
| --- |
| [root@localhost zuul]# docker run -d -p 9001:9001 --name zuul9001 --link eureka7001 ae81761550ad |



现在，三个镜像都已经运行起来并且已经映射了端口号。

测试，先访问服务注册中心



通过Zuul网关防伪Business业务

