**华为云PaaS平台微服务治理**

**1 课程介绍**

**1.1 课程概述**

本课程以华为ServiceComb微服务框架为基础，讲解微服务引擎CSE的开发方法、SpringCloud项目接入CSE的流

程，以及在华为云PaaS平台治理微服务的策略。

**1.2 快速体验**

**1.2.1 功能介绍**

本章节使用华为云官网提供的weathermap（天气预报案例）来初步体验华为云治理微服务的过程，通过本次体验

对华为云PaaS平台治理微服务有个感官的认识，并且明确本次课程的目标。

1、案例功能描述如下：

下图是weathermap的操作界面，操作流程如下：

a、进入天气预报界面，默认显示深圳的天气情况，左侧显示当天的天气情况，右侧显示未来天气预报。

b、输入城市名称（全拼），点击搜索，显示搜索城市的天气情况。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、weathermap的微服务结构如下：

各服务功能职责如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

Weathermapweb：前端界面服务，提供天气信息展示及查询界面。

Fusionweather：汇聚服务，前端请求查询天气统一Fusionweather服务，它相当于一个网关。

Weather：当前天气查询，前端请求汇聚服务查询天气，汇聚服务请求Weather查询天气。

Forecast：预报天气，查询未来天气信息，同Weather服务，汇聚服务请求Forecast查询未来天气信息。

**1.2.2 环境准备**

**1.2.2.1 开发环境**

1、JDK版本要求1.8以上。

2、Maven版本要求3.3.0以上。

本教程安装3.3.9版本。

3、配置maven setting文件以获取SDK依赖。

proﬁles中增加如下配置。

<profile>

<id>nexusProfile</id>

<repositories>

<repository>

<id>cse1</id>

<url>http://maven.huaweicse.com/nexus/content/groups/public/</url>

</repository>

</repositories>

</profile>

新增activeProﬁles配置。

<activeProfiles>

<activeProfile>nexusProfile</activeProfile>

</activeProfiles>

4、IDEA配置

设置编码格式都为 utf-8。

打开IDEA，选择 File -> Settings -> Editor -> File Encoding 把 project Encoding 和 default encoding for

properties ﬁles 改为 utf-8。

设置maven 配置

打开IDEA，选择 File -> Settings -> Build，Execution,Deployment -> Bulid Tools -> Maven 注意配置 Maven

home directory 和 User settings ﬁle

5、安装NodeJS

NodeJS版本要求4.8.4以上。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

安装成功查询版本，CMD进入命令模式，执行 node -v

**1.2.2.2 华为云环境**

微服务会连接华为云上的CSE服务中心、配置中心、仪表盘，本节配置华为云账号。

1、注册华为云账号

官网地址：https://www.huaweicloud.com/

2、AK/SK访问密钥

微服务连接CSE服务中心时，需要提供AK/SK秘钥信息，否则无法通过认证。

访问密钥即AK/SK（Access Key ID/Secret Access Key），表示一组密钥对，用于验证调用API发起请求的访问者

身份，与密码的功能相似，需要使用成对的AK/SK进行加密签名确保请求的机密性、完整性和请求双方身份的正确

性。用户可以在“我的凭证”中生成并管理访问密钥。

获取步骤：

1. 单击用户名，在下拉列表中单击“基本信息”。

2. 在基本信息页面单击“管理我的凭证”。

3. 单击“管理访问密钥”。

4. 单击“新增访问密钥”，进入“新增访问密钥”页面。

5. 输入当前用户的登录密码。

6. 通过邮箱或者手机进行验证，输入对应的验证码。

7. 下载credentials.csv文件，文件内容包括了 AK和SK。

**1.2.3 启动服务**

本节启动weathermap应用，体验weathermap的功能。

1、下载weathermap

https://registry.cn-north-

1.huaweicloud.com/swr/v2/domains/hwcse/namespaces/hwcse/repositories/default/packages/weathermap

/versions/1.0.0/ﬁle\_paths/weathermap-1.0.0.zip

2、解压weathermap下载包（解压路径不允许包含中文字符）

3、进入weathermap解压路径，打开“credentials”文件配置AK/SK

4、运行startup\_all.bat脚本启动所有微服务

5、停止服务，在Windows系统下停止服务需要手动关闭进程。

6、服务启动成功后，进入管理控制台查看“微服务管理 > 服务目录”页面，选择“ weathermap”应用，可以看到

weathermap的微服务自动注册到了云平台。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

访问http://localhost:3000/进入界面：

**1.2.4 服务治理**

进入“微服务管理 > 服务治理”页面：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**1.2.4.1 服务监控**

进入“微服务管理 > 仪表盘”界面监控服务的运行状态：

1）下图是服务监控示例图

2）下图是weathermap的各服务监控状态

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**1.2.4.2 服务降级**

降级是容错的一种特殊形式，当出现服务吞吐量巨大，资源不够用等情况，我们可使用降级机制关掉部分不重要、

性能较差的服务，避免占用资源，以保证主体业务功能可正常使用。

1. 进入前台查看“微服务管理 > 服务治理”页面，单击fusionweather微服务进入治理界面。

2. 选择“降级”页签，单击“新增”。

1. “降级对象”选择forecast微服务的所有方法。

2. “降级策略”选择“开启”。

3. 单击“确定”保存更改。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

3. 重新访问http://localhost:3000/，可以看到预报天气不再展示。

服务降级是服务治理的一种策略，常用的策略还有容错、熔断、限流、错误注入等策略。

**1.2.5 小结**

通过快速体验了解到，华为云PaaS平台提供微服务注册、微服务治理完整的解决方案，其实华为云PaaS平台提供

了微服务开发、注册、测试、治理、运维等一站式的解决方案。

**1.3 课程目标**

本课程包括以下三个部分的内容：

1、使用微服务引擎CSE（以下简称CSE）开发微服务的方法，讲解Apache ServiceComb项目接入CSE的方法、

SpringCloud项目接入CSE的方法

2、掌握微服务在华为云上的部署流程及方法。

3、深入理解微服务治理，掌握华为云PaaS平台微服务治理的常用策略。

4、华为service mesher微服务开发。

本课程以传智播客JavaEE学科实战项目“学成在线”作为课程案例，通过本课程的学习最终实现“学成在线”项目在华

为云进行部署、微服务治理，详细的学习目标如下：

1、掌握微服务引擎CSE的开发方法

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、掌握SpringCloud架构重构ServiceComb的流程

3、掌握ServiceComb接入CSE的流程

4、掌握微服务云容器引擎CCE的使用方法

包括：微服务应用生命周期管理，部署、启动、停止、升级、灰度发布、回滚、删除

5、理解微服务治理的常用策略

6、掌握微服务调用链跟踪方法

7、掌握微服务监控方法

包括：应用级监控、故障告警、日志分析

8、掌握弹性伸缩方法

**2 ServiceComb回顾**

为了更好的学习本课程，本节带你回顾ServiceComb的开发流程，如果想更加详细的学习Apache ServiceComb开

发方法请参考本系列的ServiceComb课程。

**2.1 什么是ServiceComb**

ServiceComb是华为2017年开源的微服务框架，ServiceComb在华为内部的实践中沉淀了丰富的企业级应用开发

经验，该项目已于2017年12月进入Apache孵化器。

Apache ServiceComb官网地址在http://servicecomb.incubator.apache.org/cn/

ServiceComb相比SpringCloud的优势有什么？

1、ServiceComb支持的通信协议比SpringCloud多

ServiceComb支持多种通信协议, Rest、Highway（RPC）等

SpringCloud仅支持Rest。

2、相比SpringCloud的Rest协议，Highway（RPC）协议性能更高，Highway是基于二进制的序列化方式传输数

据，采用二进制编码的系统的性能远高于采用文本的HTTP协议。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

3、ServiceComb的商业版本CSE相比SpringCloud不仅提供了微服务开发框架，还提供了微服务云部署，管理、治

理等一站式解决方案。

**2.2 注册中心**

服务中心是微服务框架中的重要组件，用于服务元数据以及服务实例元数据的管理和处理注册、发现。服务中心与

微服务提供/消费者的逻辑关系下图所示：

1、安装注册中心

提供windows、linux版本安装

http://apache.org/dyn/closer.cgi/incubator/servicecomb/incubator-servicecomb-service-

center/1.0.0-m1/apache-servicecomb-incubating-service-center-1.0.0-m1-windows-amd64.tar.gz

http://apache.org/dyn/closer.cgi/incubator/servicecomb/incubator-servicecomb-service-center/1.0.0-

m1/apache-servicecomb-incubating-service-center-1.0.0-m1-linux-amd64.tar.gz

本教程开发环境使用windows版本的注册中心。

2、启动方式

提供可执行文件启动，以Docker的方式运行

本教程开发环境使用启动可执行文件，进入安装程序的解压目录：

start-service-center.bat 注册中心

start-frontend.bat 注册中心前端

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

默认http://127.0.0.1:30103

**2.3 服务提供方**

**2.3.1 需求分析**

本案例接口调用关系如下图：

1、客户端请求helloworld-consumer服务，传入参数name

2、helloworld-consumer服务请求helloworld-provider得到student

3、helloworld-consumer服务将student响应给客户端

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**2.3.2 创建SpringBoot工程**

本案例基于SpringBoot进行构建，ServiceComb提供spring-boot-starter-provider、spring-boot-starter-

transport等与springboot的整合包。

导入本案例工程结构，如下：

工程介绍如下：

1、helloworld-model 模型工程

管理所用到模型类，如student等。

2、helloworld-api 接口工程

管理所有的服务接口。

3、helloworld-provider服务提供方

定义接口实现提供方。

4、helloworld-consumer服务消费方

服务接口调用方。

**2.3.3 引入依赖**

1、java-chassis-dependencies依赖管理

向微服务工程添加 ServiceComb的JAVA SDK包依赖，为了控制依赖版本在父工程添加 依赖管理，如下：

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>java‐chassis‐dependencies</artifactId>

<version>${servicecomb.java‐chassis.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

完整的父工程pom.xml如下：

<?xml version="1.0" encoding="UTF‐8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

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>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter‐parent</artifactId>

<version>1.5.9.RELEASE</version>

</parent>

<groupId>com.itheima</groupId>

<artifactId>test‐servicecomb</artifactId>

<packaging>pom</packaging>

<version>1.0‐SNAPSHOT</version>

<properties>

<java.version>1.8</java.version>

<tomcat.version>8.5.28</tomcat.version>

<spring‐boot.version>1.5.9.RELEASE</spring‐boot.version>

<springframework.version>4.3.13.RELEASE</springframework.version>

<mybatis‐spring‐boot.version>1.3.1</mybatis‐spring‐boot.version>

<mybatis.version>3.4.5</mybatis.version>

<druid.version>1.1.6</druid.version>

<mysql‐connector‐java.version>5.1.45</mysql‐connector‐java.version>

<commons‐io.version>2.6</commons‐io.version>

<org.apache.commons.io.version>1.3.2</org.apache.commons.io.version>

<commons‐fileupload.version>1.3.3</commons‐fileupload.version>

<commons‐codec.version>1.10</commons‐codec.version>

<commons‐lang3.version>3.6</commons‐lang3.version>

<okhttp.version>3.9.1</okhttp.version>

<lombok.version>1.16.16</lombok.version>

<mysql‐connector‐java.version>5.1.40</mysql‐connector‐java.version>

<servicecomb.java‐chassis.version>1.0.0‐m2</servicecomb.java‐chassis.version>

</properties>

<!‐‐test‐‐>

<dependencyManagement>

<dependencies>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>java‐chassis‐dependencies</artifactId>

<version>${servicecomb.java‐chassis.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql‐connector‐java</artifactId>

<version>${mysql‐connector‐java.version}</version>

</dependency>

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis‐spring‐boot‐starter</artifactId>

<version>${mybatis‐spring‐boot.version}</version>

</dependency>

<dependency>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<groupId>com.github.pagehelper</groupId>

<artifactId>pagehelper‐spring‐boot‐starter</artifactId>

<version>1.2.4</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid</artifactId>

<version>${druid.version}</version>

</dependency>

<dependency>

<groupId>commons‐io</groupId>

<artifactId>commons‐io</artifactId>

<version>${commons‐io.version}</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons‐io</artifactId>

<version>${org.apache.commons.io.version}</version>

</dependency>

<dependency>

<groupId>commons‐fileupload</groupId>

<artifactId>commons‐fileupload</artifactId>

<version>${commons‐fileupload.version}</version>

</dependency>

<dependency>

<groupId>commons‐codec</groupId>

<artifactId>commons‐codec</artifactId>

<version>${commons‐codec.version}</version>

</dependency>

<dependency>

<groupId>org.apache.commons</groupId>

<artifactId>commons‐lang3</artifactId>

<version>${commons‐lang3.version}</version>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>${lombok.version}</version>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>${fastjson.version}</version>

</dependency>

</dependencies>

</dependencyManagement>

<build>

<finalName>${project.artifactId}</finalName>

<plugins>

<!‐‐ java编译插件 ‐‐>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven‐compiler‐plugin</artifactId>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<configuration>

<source>1.8</source>

<target>1.8</target>

<encoding>UTF‐8</encoding>

</configuration>

</plugin>

</plugins>

</build>

</project>

2、向helloworld-provider添加依赖

<dependencies>

<dependency>

<groupId>com.itheima</groupId>

<artifactId>helloworld‐api</artifactId>

<version>1.0‐SNAPSHOT</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

</dependencies>

其中引入spring-boot-starter-provider表示启用java chassis的核心功能，用于“JAVA应用方式”和”Web开发方式“开

发。

**2.3.4 接口定义**

ServiceComb基于OpenAPI规范定义服务契约，契约即接口契约是服务端和消费端之间的接口定义，ServiceComb

支持Swagger注解方式定义接口契约。

（OpenAPI是Apache下的项目，它定义一种用来描述API格式或API定义的语言来规范RESTful服务开发过程，

Swagger是基于OpenAPI规范定义接口的套工具集）

1、接口定义

在helloworld-api工程定义接口：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

public interface HelloWorldInterface {

Student hello(String name);

}

在helloworld-provider定义接口，接口定义采用springmvc方式，定义接口需指定schemaId，schemaId必须保证

微服务范围内唯一。

@RestSchema(schemaId="helloworld")

@RequestMapping("/")

public class HelloWorldController implements HelloWorldInterface {

@GetMapping(path = "hello")

@Override

public Student hello(String name) {

Student student = new Student();

student.setName(name);

student.setAddress("北京");

return student;

}

}

注意：

1. 通过RestSchema替换RestController

2. 需要显示声明@RequestMapping

**2.3.5 服务描述配置**

在src/main/resources下定义microservice.yaml：

APPLICATION\_ID: helloworldproject

service\_description:

name: helloworld‐provider

version: 1.0.0

servicecomb:

rest:

address: 0.0.0.0:8080

service:

registry:

address: http://127.0.0.1:30100

配置说明：

APPLICATION\_ID：项目名/应用名

service\_description.name：微服务名

version： 微服务版本

servicecomb.rest.address:

0.0.0.0:8080：接口对外暴露的端口

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

ervicecomb.rest.service.registry.address:

http://127.0.0.1:30100

**2.3.6 启动类**

@EnableServiceComb

@SpringBootApplication

public class HelloWorldProviderApplication {

public static void main(String[] args) {

SpringApplication.run(HelloWorldProviderApplication.class);

}

}

@EnableServiceComb：启用java.chassis核心功能。

@SpringBootApplication：springBoot提供的注解，实现包扫描、自动配置。

**2.3.7 测试**

启动成功，观察服务在注册中心是否注册成功

在浏览器输入http://localhost:8080/hello?name=%E9%BB%91%E9%A9%AC

输出如下：

{"name":"黑马","age":0,"address":"北京","birthday":null}

**2.4 服务消费方**

**2.4.1 引入依赖**

在helloworld-consumer工程引入如下依赖：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

完整的代码如下：

<?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">

<parent>

<artifactId>test‐servicecomb</artifactId>

<groupId>com.itheima</groupId>

<version>1.0‐SNAPSHOT</version>

</parent>

<modelVersion>4.0.0</modelVersion>

<artifactId>helloworld‐consumer</artifactId>

<dependencies>

<dependency>

<groupId>com.itheima</groupId>

<artifactId>helloworld‐api</artifactId>

<version>1.0‐SNAPSHOT</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

</dependencies>

</project>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**2.4.2 服务调用**

ServiceComb提供RestTemplate、AsynRestTemplate、透明RPC方式调用服务接口。

RestTemplate：

RestTemplate是Spring提供的RESTful访问接口，ServiceComb提供该接口的实现类用于服务的调用。

AsynRestTemplate：

AsyncRestTemplate 开发方式允许用户异步的进行服务调用。具体的业务流程和 restTemplate 类似，只是这里以

异步的形式进行服务的调用。

透明RPC方式：

透明RPC开发模式允许用户通过简单的java interface像本地调用一样进行服务调用。透明RPC模式之所以说透明，

是因为它与服务接口使用的什么协议无关，与服务接口的开发方式无法（使用SpringMVC方式、RPC/Jax-RS），所

以在实例开发中更推荐使用透明RPC方式。

采用透明PRC方式向服务调用controller中注入接口代理对象：

@RpcReference(microserviceName="helloworld‐provider",schemaId="helloworld")

HelloWorldInterface helloWorldInterface;

此代理对象将从注册中心查找helloworld服务提供方地址，进行调用。

完整的服务调用代码如下：

@RestSchema(schemaId="helloworldConsumer")

@RequestMapping("/")

public class HelloWorldConsumerController {

@RpcReference(microserviceName="helloworld‐provider",schemaId="helloworld")

HelloWorldInterface helloWorldInterface;

@GetMapping(path = "request")

public Student request(String name){

return helloWorldInterface.hello(name);

}

}

**2.4.3 服务描述配置**

内容格式与服务提供方大致相同，内容如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

APPLICATION\_ID: helloworldproject

service\_description:

name: helloworld‐consumer

version: 1.0.0

servicecomb:

rest:

address: 0.0.0.0:8081

service:

registry:

address: http://127.0.0.1:30100

**2.4.4 启动类**

@EnableServiceComb

@SpringBootApplication

public class HelloWorldConsumerApplication {

public static void main(String[] args) {

SpringApplication.run(HelloWorldConsumerApplication.class);

}

}

**2.4.5 测试**

客户端请求服务消费方，服务消费方请求服务提供方。

浏览器请求：http://localhost:8081/request?name=%E9%BB%91%E9%A9%AC

http://localhost:8081/request

输出：

{"name":"黑马","age":0,"address":"北京","birthday":null}

**3 微服务引擎CSE**

**3.1 华为云PaaS平台介绍**

华为云PaaS平台以华为FusionStage2.0项目为基础，基于Docker、Kubernetes等主流开源容器项目，提供了开

发、测试、部署、运维的全流程自动化流水线能力，提供先进的微服务框架和丰富的服务组件，致力于让开发者专

注于代码开发本身，提升开发效率，支持丰富的应用生态，推动企业更好的云化转型。

什么是PaaS？

PaaS是云计算的三种模式之一，三种模式如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

IaaS: Infrastructure-as-a-Service（基础设施即服务），也叫Hardware-as-a-Service，将计算机硬件资源打包对

外提供服务，比如云主机、云存储等。

PaaS: Platform-as-a-Service（平台即服务），也叫中间件服务，比如：MySQL数据库服务、ElasticSearch搜索服

务等。 SaaS: Software-as-a-Service（软件即服务） ，提供具体应用软件服务，比如：CRM系统，电商平台等。

什么是FusionStage2.0？

华为云称为FusionCloud，它包括三部分：FusionStage（PaaS服务），FusionInsight(华为大数据平台)、

FusionSphere(IaaS服务)，FusionStage作为华为PaaS平台的项目其版本已经升级到2.0。

什么是Docker？

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中，然后发布

到任何流行的Linux机器上，也可以实现虚拟化，容器是完全使用沙箱机制，相互之间不会有任何接口。微服务由

于数量众多，其推荐的部署方式业界流行的做法就是Docker技术，使用Docker可以快速批量部署微服务，提高运

维效率。

什么是Kubernetes？

Kubernetes是一个开源的，用于管理云平台中多个主机上的容器化的应用，使用Kubernetes的目标是让Docker容

器化部署更加简单、智能和高效。

华为云PaaS平台支持哪些微服务开发框架？

华为云PaaS平台不光提供云容器部署、微服务治理的功能，而且提供微服务开发、测试、部署、运维一站式解决方

案，支持Dubbo、SpringCloud等技术开发的微服务接入PaaS平台，也支持老系统的接入，并且对于Apache

ServiceComb开的微服务应用更是零修改代码接入。

华为云PaaS平台提供微服务引擎CSE框架，此框架在ServiceComb框架的基础上另外提供更多的商业功能支持。本

教程会讲解SpringCloud项目接入CSE的方法。

**3.2 微服务引擎CSE介绍**

微服务引擎CSE(Cloud Service Engine) 是华为云PaaS家族提供的企业级微服务应用管理平台，包含微服务

SDK（用于开发）、服务注册、配置和治理中心，帮助用户实现微服务应用的快速构建、实时监控和高可用保障。

兼容主流开源生态，不绑定特定开发框架和平台。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

和

CSE具备开发模式门槛低、通用性强的改造优势，并支持契约优先，服务接口统一规范管理，微服务之间的调用非

常方便，根据业务的现实痛点将其后端拆分为多个微服务，并实施集群部署，这一切全部基于CSE微服务框架开发

并统一管理。

产品功能如下：

微服务开发框架

打包了微服务注册、发现、通信和治理等基础能力，开箱即用。

微服务治理中心

提供微服务负载均衡、限流、降级、熔断、容错等治理能力。

微服务安全管控

提供认证鉴权、黑白名单等能力，保障微服务访问安全。

微服务灰度发布

支持使用接口任意参数（例如用户群组、用户类别、用户所属区域等等）定义微服务灰度发布规则。

分布式事务管理

提供最终一致性（TCC）和强一致性（WSAT）事务管理框架。

非侵入式微服务接入

提供Service Mesh服务，可实现非侵入式接入已有微服务。

统一配置中心

支持微服务配置项的发布、变更和通知。

微服务仪表盘

提供微服务实例和接口级吞吐量、时延和成功率的实时监控仪表盘。

ServiceComb与CSE有什么不同？

1、 CSE是ServiceComb的商业版本，CSE的大部分组件来自于开源的ServiceComb，ServiceComb是开源的，

CSE没有开源但可以免费使用。

2、 使用ServiceComb开发的微服务不依赖华为云，使用CSE开发的微服务则依赖华为云。

为什么要用微服务引擎CSE？

在云上使用CSE，用户无需关心自己业务之外的微服务基础设施，保证系统整体稳定性。同时微服务部署，管理、

治理全部开箱即用，还可以获得开发时期的指导和帮助，以及后续技术发展的最新信息。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**3.3 ServiceComb项目接入CSE**

**3.3.1 准备华为云环境**

参考“快速体验”章节的“华为云环境”章节内容，注册云账号、获取AK/SK密钥。

**3.3.2 服务提供方**

本案例使用ServiceComb回顾章节的helloworld案例进行改造，ServiceComb接入CSE只需要修改依赖及部分配

置，无需修改代码。

**3.3.2.1 引入依赖**

接入CSE平台需要在服务提供方及消费方引入 cse-solution-service-engine依赖，为了控制依赖版本在父工程进行

依赖管理如下：

<dependencyManagement>

<dependencies>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐dependency</artifactId>

<version>2.3.20</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

在服务提供方添加依赖：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

</dependency>

服务提供方完整的pom.xml 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>

<parent>

<groupId>com.itheima</groupId>

<artifactId>test‐servicecomb</artifactId>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<version>1.0‐SNAPSHOT</version>

</parent>

<artifactId>helloworld‐provider</artifactId>

<dependencies>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>com.itheima</groupId>

<artifactId>helloworld‐api</artifactId>

<version>1.0‐SNAPSHOT</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

</dependencies>

</project>

**3.3.2.2 服务描述配置**

注册中心更改为公有云地址，并且配置AK/SK。

APPLICATION\_ID: helloworld

service\_description:

name: helloworld‐provider

version: 1.0.0

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com

instance:

watch: false

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

credentials:

accessKey: CMPEUCGEDINQKGWBGHVB

secretKey: HEYSZL5zTw6HLfPEOxzGnWpbzLwHWAylUc99vMtd

akskCustomCipher: default

rest:

address: 0.0.0.0:8080

**3.3.2.3 启动服务**

启动服务后，登录云平台：微服务引擎CSE-->微服务管理--->服务目录

**3.3.2.4 测试**

在浏览器输入http://localhost:8080/hello?name=%E9%BB%91%E9%A9%AC

输出如下：

{"name":"黑马","age":0,"address":"北京","birthday":null}

**3.3.3 服务消费方**

**3.3.3.1 引入依赖**

同提供提供方。

**3.3.3.2 服务描述配置**

同提供提供方。

**3.3.3.3 启动服务**

启动消费者服务

**3.3.3.4 测试**

客户端请求服务消费方，服务消费方请求服务提供方。

浏览器请求：http://localhost:8081/request?name=%E9%BB%91%E9%A9%AC

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

http://localhost:8081/request

输出：

{"name":"黑马","age":0,"address":"北京","birthday":null}

**3.4 小结**

本章节重点讲解了ServiceComb项目接入CSE的方法，全新的CSE项目开发可以按照ServiceComb开发流程进行，

并且引用CSE依赖、配置公有云注册中心地址等，详细的流程如下：

1、注册云账号

2、引入依赖

添加依赖管理如下：

<dependencyManagement>

<dependencies>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐dependency</artifactId>

<version>2.3.20</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

添加依赖如下：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

</dependency>

3、接口定义

同serviceComb开发。

4、服务描述配置

配置公有云注册中心地址。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4 学成在线项目接入CSE**

学成在线项目是黑马程序员JavaEE大型综合实战项目，本章节实现学成在线项目接入CSE，本项目采用

SpringCloud框架开发，本章节系统讲解SpringCloud项目接入CSE的开发步骤。

**4.1 项目介绍**

**4.1.1 技术架构**

学成在线采用当前流行的前后端分离架构开发，前端采用vue.js构建，服务端采用Spring Cloud微服务架构，系统

分为用户层、CDN、负载均衡、前端UI、微服务层、数据层、接口层及DevOps等部分组成，技术架构图如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.1.2 业务流程**

本次接入CSE以学成在线项目中“在线播放视频”业务流程为案例进行讲解，业务流程业务如下：

1、进入门户

2、搜索课程

点击“课程”进入课程搜索页面，搜索课程

3、进入课程学习页面

点击进入课程详情页面，点击“马上学习”

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.1.3 项目部署**

1、安装MySQL、导入“资料”--》“学成在线”--》MySQL目录下的脚本。

2、安装MongoDB，导入“资料”--》“学成在线”--》mongodb目录下的脚本。

3、安装ElasticSearch，参考“资料”--》“学成在线”--》elasticsearch下的脚本使用postman导入初始化数据。

4、启动以下微服务

xc-govern-center：注册中心

xc-govern-gateway： 网关

xc-service-portalview：数据视图服务

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

xc-service-search： 搜索服务

xc-service-learning： 学习服务

5、前端工程

解压“资料”--》“学成在线”--》xc-ui-pc-learning.zip，此为学习中心前端工程

解压“资料”--》“学成在线”--》xc-ui-pc-portal.zip，此为搜索前端工程

使用webstorm分别打开两个前端工程。

分别在两个目录下运行npm install或cnpm install

运行 npm run dev 启动前端工程。

解压门户工程，“资料”--》“学成在线”--》xc-ui-pc-static-portal.zip

解压静态资源，“资料”--》“学成在线”--》static.zip

解压视频资源，“资料”--》“学成在线”--》video.zip

安装nginx，并参考“资料”--》“学成在线”--》nginx.conf配置nginx.conf

配置nginx.conf注意以下虚拟主机的配置：

90端口为视频播放

91端口为静态资源

xuechengcloud.java.itcast.cn为门户

xcucentercloud.java.itcast.cn 为学习中心

启动Nginx。

运行工程，需要在hosts文件中配置域名

127.0.0.1 xuechengcloud.java.itcast.cn 127.0.0.1 xcucentercloud.java.itcast.cn 127.0.0.1

xcsystemcloud.java.itcast.cn 127.0.0.1 xcteachercloud.java.itcast.cn 127.0.0.1 xcimgcloud.java.itcast.cn

127.0.0.1 xcvideocloud.java.itcast.cn

浏览：http://xuechengcloud.java.itcast.cn

**4.1.4 微服务技术栈**

在“在线播放视频”业务流程中包括数据视图服务、在线学习服务、搜索等微服务，交互关系图如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

交互流程如下：

1、前端通过网关调用xc-service-search搜索课程

2、前端通过网关调用xc-service-learning获取学习权限，拥有学习权限则得到视频播放地址

3、xc-service-learning校验学习权限，有权限则请求xc-service-portalview查询视频播放地址

4、学习服务、数据视图服务、搜索服务向Eureka服务注册中心上报微服务信息。

微服务基于Spring Boot构建，技术栈如下：

SpringBoot、SpringCloud、Swagger、Spring Data Mongodb、MyBatis、Spring Data Jpa、

com.alibaba.druid、Ribbon、Feign Client、Hystrix

**4.1.5 原始代码结构**

项目接入CSE需要修改项目代码，在修改代码之前我们先了解一下原始接口的代码结构。

**4.1.5.1 查询课程视图接口**

学习中心前端请求搜索服务查询课程信息接口

1、接口定义

项目统一在xc-service-api工程中定义接口，在搜索服务包下定义此接口。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

@Api(value = "课程搜索",description = "课程搜索")

public interface EsCourseControllerApi {

final String API\_PRE = "/search/course";

@GetMapping(API\_PRE+"/list/{page}/{size}")

@ApiOperation("课程搜索")

public

Map list(@PathVariable("page") int page,

@PathVariable("size") int size,

CourseSearchParam courseSearchParam) throws IOException;

@GetMapping(value=API\_PRE+"/getall/{id}")

@ApiOperation("根据id查询课程信息")

public Map<String,CoursePub> getall(@PathVariable("id") String id) throws IOException;

...

}

2、接口实现

在xc-service-search工程定义接口实现。

@RestController

public class EsCourseController implements EsCourseControllerApi {

@Autowired

EsCourseService esCourseService;

@Override

public

Map list(@PathVariable("page") int page, @PathVariable("size") int size,

CourseSearchParam courseSearchParam) throws IOException {

String result = esCourseService.list(page,size,courseSearchParam);

return JSON.parseObject(result, Map.class);

}

@Override

public Map<String, CoursePub> getall(@PathVariable("id") String id) throws IOException {

return esCourseService.getall(id);

}

...

}

**4.1.5.2 获取学习地址接口**

学习服务提供获取学习地址接口，学习中心前端请求学习服务获取获取学习地址。

1、接口定义

在xc-service-api工程下的学习服务包下定义此接口。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

@RequestMapping("/learning")

@Api(value = "录播课程学习管理",description = "录播课程学习管理")

public interface CourseLearningControllerApi {

@GetMapping("/getmedia/{courseId}/{teachplanId}")

@ApiOperation("获取课程学习地址")

public GetMediaResult getmedia(@PathVariable String courseId,@PathVariable String

teachplanId);

...

}

2、接口实现

@RestController

public class CourseLearningController extends BaseController implements

CourseLearningControllerApi {

@Autowired

LearningService learningService;

@Override

public GetMediaResult getmedia(@PathVariable String courseId, @PathVariable String

teachplanId) {

//获取课程学习地址

GetMediaResult media = learningService.getMedia(courseId, teachplanId);

return media;

}

...

}

在learningService采用远程调用，这里使用SpringCloud提供的FeignClient，在学习服务中定义FeignClient：

@FeignClient(value = XcServiceList.XC\_SERVICE\_PORTALVIEW)

public interface PortalViewClient extends ViewCourseControllerApi {

}

learningService的getMedia方法代码如下：

//获取课程学习地址

public GetMediaResult getMedia(String courseId, String teachplanId) {

//校验学生的学习权限...

String mediaFileId = null;

//远程调用视图服务获取课程媒资信息

ViewCourseMedia viewCourseMedia = portalViewClient.findMediaById(teachplanId);

if (viewCourseMedia == null) {

ExceptionCast.cast(LearningCode.LEARNING\_GETMEDIA\_ERROR);

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

}

//视频播放地址

String mediaUrl = viewCourseMedia.getMediaUrl();

if (StringUtils.isEmpty(mediaUrl)) {

ExceptionCast.cast(LearningCode.LEARNING\_GETMEDIA\_NOTFOUND);

}

//响应结果对象

GetMediaResult getMediaResult = new GetMediaResult(CommonCode.SUCCESS, mediaUrl);

return getMediaResult;

}

**4.1.5.3 获取视频播放地址**

数据视图服务提供获取视频播放地址接口，学习服务在验证学生有学习权限后请求数据视图服务获取视频播放地

址。

1、接口定义

@Api(value = "课程数据视图服务",description = "门户课程数据视图管理接口")

public interface ViewCourseControllerApi {

final String API\_PRE = "/portalview/course";

@GetMapping(API\_PRE+"/getmedia/{id}")

@ApiOperation("通过教学计划id查询课程媒资视图")

public ViewCourseMedia findMediaById(@PathVariable("id") String id) ;

...

}

2、接口实现

@RestController

public class ViewCourseController implements ViewCourseControllerApi{

@Override

public ViewCourseMedia findMediaById(@PathVariable("id") String id) {

ViewCourseMedia viewCourseMedia = viewCourseService.findMediaById(id);

return viewCourseMedia;

}

...

}

**4.2 SpringCloud项目接入CSE**

本章节讲解SpringCloud项目（学成在线）接入CSE的步骤和方法，以接口为单元进行接入。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.2.1 查询课程视图接口**

**4.2.1.1 引入依赖**

接入CSE平台需要在服务提供方及消费方引入 cse-solution-service-engine依赖，为了控制依赖版本在父工程进行

依赖管理如下：

<dependencyManagement>

<dependencies>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐dependency</artifactId>

<version>2.3.20</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

完整的父工程pom.xml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>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter‐parent</artifactId>

<version>1.5.9.RELEASE</version>

</parent>

<groupId>com.xuecheng</groupId>

<artifactId>xc‐framework‐parent</artifactId>

<packaging>pom</packaging>

<version>1.0‐SNAPSHOT</version>

<properties>

<java.version>1.8</java.version>

<tomcat.version>8.5.28</tomcat.version>

<spring‐boot.version>1.5.9.RELEASE</spring‐boot.version>

<springframework.version>4.3.13.RELEASE</springframework.version>

<mybatis‐spring‐boot.version>1.3.1</mybatis‐spring‐boot.version>

<mybatis.version>3.4.5</mybatis.version>

<druid.version>1.1.6</druid.version>

<mysql‐connector‐java.version>5.1.45</mysql‐connector‐java.version>

<commons‐io.version>2.6</commons‐io.version>

<org.apache.commons.io.version>1.3.2</org.apache.commons.io.version>

<commons‐fileupload.version>1.3.3</commons‐fileupload.version>

<commons‐codec.version>1.10</commons‐codec.version>

<commons‐lang3.version>3.6</commons‐lang3.version>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<okhttp.version>3.9.1</okhttp.version>

<feign‐okhttp.version>8.18.0</feign‐okhttp.version>

<lombok.version>1.16.16</lombok.version>

<springfox‐swagger.version>2.7.0</springfox‐swagger.version>

<fastjson.version>1.2.30</fastjson.version>

<fastdfs‐client‐java.version>1.27.0.0</fastdfs‐client‐java.version>

<mysql‐connector‐java.version>5.1.40</mysql‐connector‐java.version>

<elasticsearch.version>6.2.1</elasticsearch.version>

<guava.version>24.0‐jre</guava.version>

<cse.version>2.3.30</cse.version>

</properties>

<!‐‐test‐‐>

<dependencyManagement>

<dependencies>

<!‐‐<dependency>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring‐cloud‐dependencies</artifactId>

<version>Dalston.SR5</version>

<type>pom</type>

<scope>import</scope>

</dependency>‐‐>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐dependency</artifactId>

<version>${cse.version}</version>

<type>pom</type>

<scope>import</scope>

</dependency>

...

在xc-service-search工程添加依赖：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.2.1.2 屏蔽本项目Swagger**

serviceComb提供Swagger方式生成接口契约，为防止本项目生成Swagger契约与ServiceComb冲突现将本项目的

xc-service-api工程中的conﬁg/Swagger2Conﬁguration.java类上的注解屏蔽：

//@Configuration

//@EnableSwagger2

public class Swagger2Configuration {

...

**4.2.1.3 接口定义**

修改xc-service-course工程的com.xuecheng.search.web.controller.EsCourseController

针对query参数，ServiceComb推荐使用基本类型，课程搜索列表接口修改如下：

public

Map<String,String> list(@PathVariable("page") int page, @PathVariable("size") int size,

@RequestParam("keyword") String keyword,

@RequestParam("mt") String mt,

@RequestParam("st") String st,

@RequestParam("grade") String grade) throws IOException;

接口实现代码如下：

屏蔽@RestController注解

//@RestController

@RequestMapping("/search/course")

@RestSchema(schemaId = XcServiceList.XC\_SERVICE\_SEARCH)

public class EsCourseController implements EsCourseControllerApi {

@Autowired

EsCourseService esCourseService;

@GetMapping(value="/list/{page}/{size}",produces="application/json;charset=UTF‐8")

@Override

public

Map<String,String> list(@PathVariable("page") int page, @PathVariable("size") int

size,

@RequestParam("keyword") String keyword,

@RequestParam("mt") String mt,

@RequestParam("st") String st,

@RequestParam("grade") String grade

) throws IOException {

CourseSearchParam courseSearchParam = new CourseSearchParam();

courseSearchParam.setKeyword(keyword);

courseSearchParam.setSt(st);

courseSearchParam.setMt(mt);

courseSearchParam.setGrade(grade);

String result = esCourseService.list(page,size,courseSearchParam);

return JSON.parseObject(result, Map.class);

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

}

@GetMapping(value="/getbase/{ids}")

@Override

public Map<String,CourseInfo> getbase(@PathVariable("ids") String ids) throws IOException {

return esCourseService.getbase(ids);

}

@GetMapping(value="/getall/{id}")

@Override

public Map<String, CoursePub> getall(@PathVariable("id") String id) throws IOException {

return esCourseService.getall(id);

}

}

**4.2.1.4 服务描述配置**

在src/main/resources配置microservice.yaml：

APPLICATION\_ID: xc‐edu‐cloud1.0

service\_description:

name: xc‐service‐search

version: 1.0.0

properties:

allowCrossApp: false

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com

instance:

watch: false

credentials:

accessKey: CMPEUCGEDINQKGWBGHVB

secretKey: HEYSZL5zTw6HLfPEOxzGnWpbzLwHWAylUc99vMtd

akskCustomCipher: default

rest:

address: 0.0.0.0:40100

**4.2.1.5 定义服务启动入口**

删除：

@EnableDiscoveryClient @EnableFeignClients

添加：

@EnableServiceComb

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

代码如下：

@EnableServiceComb

@SpringBootApplication

public class SearchApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(SearchApplication.class, args);

}

}

**4.2.1.6 启动服务**

启动搜索服务成功。

启动服务后，登录云平台：微服务引擎CSE-->微服务管理--->服务目录，查看搜索服务已经向公有云注册中心注册

成功，并且查看接口契约是否正确。

接口契约：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

浏览器输入搜索地址进行测试：

**4.2.2 获取视频播放地址接口**

由于xc-service-learning要调用xc-service-portalview服务的获取视频播放地址接口，所以先改造此接口。

参考“查询课程视图接口”修改xc-service-portalview的获取视频播放地址接口

**4.2.2.1屏蔽eureka**

1、屏蔽spring-cloud-starter-eureka依赖

2、屏蔽spring-cloud-starter-eureka配置

**4.2.2.2 添加依赖**

向xc-service-portalview的pom.xml中添加如下依赖：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

**4.2.2.3 接口定义**

参考“查询课程视图接口”修改接口

//@RestController

@RequestMapping("/portalview/course")

@RestSchema(schemaId = XcServiceList.XC\_SERVICE\_PORTALVIEW)

public class ViewCourseController implements ViewCourseControllerApi{

@Autowired

ViewCourseService viewCourseService;

@GetMapping(API\_PRE+"/get/{id}")

@Override

public ViewCourse findById(@PathVariable("id") String id) {

ViewCourse viewCourse = viewCourseService.findCourseById(id);

return viewCourse;

}

@GetMapping(API\_PRE+"/getmedia/{id}")

@Override

public ViewCourseMedia findMediaById(@PathVariable("id") String id) {

ViewCourseMedia viewCourseMedia = viewCourseService.findMediaById(id);

return viewCourseMedia;

}

}

**4.2.2.4 服务描述配置**

参考 “查询课程视图接口”配置microservice.yaml

APPLICATION\_ID: xc‐edu‐cloud1.0

service\_description:

name: xc‐service‐portalview

version: 1.0.0

properties:

allowCrossApp: false

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

instance:

watch: false

credentials:

accessKey: CMPEUCGEDINQKGWBGHVB

secretKey: HEYSZL5zTw6HLfPEOxzGnWpbzLwHWAylUc99vMtd

akskCustomCipher: default

rest:

address: 0.0.0.0:40200

**4.2.2.5 定义服务启动入口**

删除：

@EnableDiscoveryClient @EnableFeignClients

添加：

@EnableServiceComb

代码如下：

//@EnableDiscoveryClient

@EnableServiceComb

@SpringBootApplication(exclude=

{DataSourceAutoConfiguration.class,HibernateJpaAutoConfiguration.class})

public class PortalviewApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(PortalviewApplication.class, args);

}

}

**4.2.2.6 启动服务**

启动服务，登录云平台查看服务目录及接口契约：

**4.2.3 获取学习地址接口**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.2.3.1 需求分析**

获取学习地址接口的功能内容如下：

1、接收到前端请求校验学生的学习权限。

2、如果具有学习权限则远程调用“数据视图服务”获取视频播放地址

3、将视频播放地址返回给前端。

本接口改造除了参考“查询课程视图接口”修改接口定义之外，还必须将原有的FeignClient改为ServiceComb的远程

调用方式，本项目使用ServiceComb提供的透明RPC方式实现远程调用。

**4.2.3.2 远程调用接口定义**

在xc-service-api工程定义xc-service-portalview接口门面，方便其它服务引用接口并调用。

1、在xc-service-api工程添加依赖

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

2、在portalview目录下定义PortalViewApiFacade接口门面

使用serviceComb提供的@RpcReference 注解，指定微服务名称及schemaId。

@Component

@Getter

public class PortalViewApiFacade {

@RpcReference(microserviceName= XcServiceList.XC\_SERVICE\_PORTALVIEW, schemaId=

XcServiceList.XC\_SERVICE\_PORTALVIEW)

ViewCourseControllerApi viewCourseControllerApi;

}

**4.2.3.3 添加依赖**

向xc-service-learning的pom.xml中添加如下依赖：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring‐jdbc</artifactId>

<scope>compile</scope>

</dependency>

**4.2.3.4 接口定义**

//@RestController

@RequestMapping("/learning")

@RestSchema(schemaId = XcServiceList.XC\_SERVICE\_LEARNING)

public class CourseLearningController extends BaseController implements

CourseLearningControllerApi {

@Autowired

LearningService learningService;

@GetMapping("/getmedia/{courseId}/{teachplanId}")

@Override

public GetMediaResult getmedia(@PathVariable String courseId, @PathVariable String

teachplanId) {

//获取课程学习地址

GetMediaResult media = learningService.getMedia(courseId, teachplanId);

return media;

}

...

learningService的getMedia方法修改如下：

注入PortalViewApiFacade。

代码如下：

@Service

public class LearningService {

private static final Logger LOGGER = LoggerFactory.getLogger(LearningService.class);

//

//

@Autowired

PortalViewClient portalViewClient;

@Autowired

CourseClient courseClient;

@Autowired

PortalViewApiFacade portalViewApiFacade;

@Autowired

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

XcLearningCourseRepository xcLearningCourseRepository;

//获取课程学习地址

public GetMediaResult getMedia(String courseId, String teachplanId) {

//校验学生的学习权限...

String mediaFileId = null;

//远程调用视图服务获取课程媒资信息

//

ViewCourseMedia viewCourseMedia = portalViewClient.findMediaById(teachplanId);

ViewCourseMedia viewCourseMedia =

portalViewApiFacade.getViewCourseControllerApi().findMediaById(teachplanId);

if (viewCourseMedia == null) {

ExceptionCast.cast(LearningCode.LEARNING\_GETMEDIA\_ERROR);

}

//视频播放地址

String mediaUrl = viewCourseMedia.getMediaUrl();

if (StringUtils.isEmpty(mediaUrl)) {

ExceptionCast.cast(LearningCode.LEARNING\_GETMEDIA\_NOTFOUND);

}

//响应结果对象

GetMediaResult getMediaResult = new GetMediaResult(CommonCode.SUCCESS, mediaUrl);

return getMediaResult;

}

**4.2.3.5 服务描述配置**

参考前边章节，略。

**4.2.3.6 定义服务启动入口**

参考前边章节，略。

**4.2.3.7 测试**

启动服务观察云平台服务目录：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

在浏览器请求：

http://localhost:40600/learning/getmedia/402885816243d2dd016243f24c030002/8a7e867a6459d24c01645

9e7cac90002

**4.2.4 网关**

**4.2.4.1 Edge Service**

在SpringCloud中Zuul作为网关提供路由、过虑等功能。在ServiceComb中Edge Service提供JAVA网关服务。

Edge Service作为整个微服务系统对外的接口，向最终用户提供服务，接收RESTful请求，转发给内部微服务。

Edge Service以开发框架的形式提供，开发者可以非常简单的搭建一个Edge Service服务，通过简单的配置就可以

定义路由转发规则。同时Edge Service支持强大的扩展能力，服务映射、请求解析、加密解密、鉴权等逻辑都可以

通过扩展实现。

**4.2.4.2 新建Edge Service工程**

创建Spring Boot工程，创建包：com.xuecheng.govern.gateway

引入如下依赖：

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>edge‐core</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

完整的pom.xml如下：

<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>

<parent>

<artifactId>xc‐framework‐parent</artifactId>

<groupId>com.xuecheng</groupId>

<version>1.0‐SNAPSHOT</version>

<relativePath>../xc‐framework‐parent/pom.xml</relativePath>

</parent>

<artifactId>xc‐govern‐edgeService</artifactId>

<properties>

<project.build.sourceEncoding>UTF‐8</project.build.sourceEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>com.huawei.paas.cse</groupId>

<artifactId>cse‐solution‐service‐engine</artifactId>

<exclusions>

<exclusion>

<groupId>org.slf4j</groupId>

<artifactId>slf4j‐log4j12</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate‐validator</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>spring‐boot‐starter‐provider</artifactId>

</dependency>

<dependency>

<groupId>org.apache.servicecomb</groupId>

<artifactId>edge‐core</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter</artifactId>

</dependency>

</dependencies>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐maven‐plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**4.2.4.3 服务描述配置**

网关也相当与一个微服务，最终会向服务中心注册，服务描述配置如下：

在src/main/resource下配置application.yml，内容如下：

spring:

application:

name: xc‐edu‐edgeService

server:

port: 50201

# 应用名称

APPLICATION\_ID: xc‐edu‐cloud1.0

# 微服务名称和版本号

service\_description:

name: ${spring.application.name}

version: 1.0.0

# 服务中心和配置中心的地址

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com

instance:

watch: false

config:

client:

serverUri: https://cse.cn‐north‐1.myhuaweicloud.com

refreshMode: 1

refresh\_interval: 15000

# 对外发布的地址，端口号使用server.port

rest:

address: 0.0.0.0:${server.port}

# AK/SK认证信息，需要修改为自己的AK/SK

credentials:

accessKey: 9AKR38RXJG48XMDDIUUA

secretKey: 2MEIV4VhNzRausA3tLOsN1s7nWw66SSp9hVXBJrl

akskCustomCipher: default

**4.2.4.4 启动类**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

定义启动类：

@SpringBootApplication

@EnableServiceComb

public class EdgeApplication {

public static void main(String[] args) throws Exception {

SpringApplication.run(EdgeApplication.class, args);

}

}

启动EdgeService工程，观察服务注册中心是否注册成功

**4.2.4.5 配置路由**

网关的一个作用就是路由，在application.yml中配置路由，如下：

cse:

# 路由配置

http:

dispatcher:

edge:

default:

enabled: true

prefix: api

withVersion: false

prefixSegmentCount: 2

url:

enabled: true

mappings:

xc‐service‐search:

prefixSegmentCount: 2

path: "/openapi/search/.\*"

withVersion: false

microserviceName: xc‐service‐search

#

versionRule: 1.0.0‐2.0.0

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

说明：

上边的路由配置包括两部分，一部分为默认路由，一部分是根据具体url配置的路由。

1、默认路由：

edge:

default:

enabled: true

prefix: api

withVersion: false

prefixSegmentCount: 2

默认路由是根据 “/preﬁx/微服务名称/\*来匹配” 请求url，如果withVersion设置为true则根据

“/preﬁx/微服务名称/版本号/\*" 来匹配请求。

根据上边的配置举例：请求/api/xc-service-search/search/course/list/1/2 网关将请求转发到xc-service-search微

服务，请求该微服务的“/search/course/list/1/2 ” URL。

preﬁxSegmentCount表示前缀段的个数，由于使用“/preﬁx/微服务名称 ”来匹配请求url所以设置为2。

启动Edge Service服务。

请求：http://localhost:50201/api/xc-service-search/search/course/list/1/2

通过Edge Service 请求搜索服务。

2、根据具体url配置的路由

url:

enabled: true

mappings:

xc‐service‐search:

#其中一个mapping名称，所有的mapping名称不允许重复

prefixSegmentCount: 2

path: "/openapi/search/.\*"

withVersion: false

microserviceName: xc‐service‐search

#versionRule: 1.0.0‐2.0.0

请求 "/openapi/search/"将转发到微服务xc-service-search。

microserviceName：微服务名称

versionRule：版本匹配规则，常用于灰度发布，上边versionRule: 1.0.0-2.0.0表示该mapping只用于1.0.0到2.0.0

版本。

启动Edge Service服务。

请求：http://localhost:50201/openapi/service/search/course/list/1/2

通过Edge Service 请求搜索服务。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**4.2.4.6 过虑器**

通过过虑器可以校验请求的合法性，Edge Service提供过虑器定义方式。

定义过虑器如下：

package com.xuecheng.govern.gateway.filter;

import org.apache.commons.lang3.StringUtils;

import org.apache.servicecomb.common.rest.filter.HttpServerFilter;

import org.apache.servicecomb.core.Invocation;

import org.apache.servicecomb.foundation.vertx.http.HttpServletRequestEx;

import org.apache.servicecomb.provider.springmvc.reference.RestTemplateBuilder;

import org.apache.servicecomb.swagger.invocation.Response;

import org.apache.servicecomb.swagger.invocation.exception.InvocationException;

import org.springframework.web.client.RestTemplate;

import javax.ws.rs.core.Response.Status;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import static org.springframework.http.HttpHeaders.AUTHORIZATION;

public class AuthenticationFilter implements HttpServerFilter {

private final RestTemplate template = RestTemplateBuilder.create();

private static final Set<String> NOT\_REQUIRED\_VERIFICATION\_SERVICE\_NAMES = new HashSet<>(

Arrays.asList("xc‐service‐search", "xc‐service‐portalview", "xc‐service‐learning"));

@Override

public int getOrder() {

return 0;

}

@Override

public Response afterReceiveRequest(Invocation invocation, HttpServletRequestEx

httpServletRequestEx) {

if (isInvocationNeedValidate(invocation.getMicroserviceName())) {

String token = httpServletRequestEx.getHeader(AUTHORIZATION);

if (StringUtils.isNotEmpty(token)) {

//校验token...

} else {

return Response.failResp(

new InvocationException(Status.UNAUTHORIZED, "authentication failed, missing

AUTHORIZATION header"));

}

}

return null;

}

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

private boolean isInvocationNeedValidate(String serviceName) {

for (String service\_name : NOT\_REQUIRED\_VERIFICATION\_SERVICE\_NAMES) {

if (serviceName.startsWith(service\_name)) {

return false;

}

}

return true;

}

}

配置过虑器：

过虑器配置采用Java SPI（Service Provider Interfaces）的方式来配置，Java SPI是通过java.util.ServiceLoader实

例化META-INF/services下定义的接口类（和Spring的配置类似）配置如下：

在src/main/resource下定义META-INF/services目录。

由于上边定义的过虑器实现了HttpServerFilter接口，在META-INF/services目录下定义文件，文件名称为

HttpServerFilter的全限定名，如下：

文件内容是上边定义的过虑器的全限定名，如下：

com.xuecheng.govern.gateway.filter.AuthenticationFilter

重新启动Edge Service工程，测试。

**4.2.5 集成测试**

启动portalview、learning、search、edge Service服务，测试整个在线学习流程，下图集成测试结果，可以正常

获取视频地址并在前端播放。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5 云容器引擎CCE**

**5.1 CCE介绍**

云容器引擎CCE（Cloud Container Engine）提供高可靠高性能的企业级容器应用管理服务，支持Kubernetes社

区原生应用和工具，简化云上自动化容器运行环境搭建。通过云容器引擎可以快速高效的将微服务部署在云端。

**5.1.1 功能列表**

CCE涵盖了应用全生命周期管理、DevOps持续交付、应用调度、弹性伸缩等众多功能，详细如下：

1、一站式部署和运维 一键创建Kubernetes集群，自动化部署、运维容器应用，整个生命周期都在容器服务内一站

式完成。

2、高可用业务不中断 集群控制面HA和跨AZ高可用，容器应用优雅伸缩，安全下线，保障业务不掉线。

3、丰富的应用调度策略 支持多种亲和反亲和调度策略，用户可方便地根据应用特点在高性能和高可靠中找到平衡

点。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

4、灵活的弹性伸缩策略 支持按性能、按时间、按周期的弹性伸缩策略，用户可自由组合策略以应对多变的突发浪

涌。

5、兼容原生Kubernetes和Docker 紧跟社区，3个月同步最新版本，兼容原生API和命令行。

6、可定制化的DevOps能力 支持Java/Python/Go/Node.JS/Vert.x等语言代码开发一键式生成镜像能力。

7、IaaS资源深度整合 直接使用华为云ECS/BMS、网络(VPC/EIP)、存储（EVS/OBS/SFS）能力。

**5.1.2 产品优势**

CCE基于业界主流的Docker和Kubernetes开源技术，并进行了大量的商用增强，在系统可靠性、高性能、开源社

区的兼容性等多个方面具有独特的优势。

简单易用 自动化创建容器集群，一站式部署/运维容器应用，一键式滚动升级。

高性能 自研高性能容器网络，秒级自动弹性伸缩，支持高性能裸金属容器私有集群。

企业级 集群控制面HA和跨AZ高可用，容器应用优雅伸缩，安全下线，保障业务不掉线。

开放性 全兼容Kubernetes/Docker社区原生版本，CNCF首批认证的Kubernetes服务提供商，社区的主要贡献者。

**5.2 创建集群**

**5.2.1 什么是集群**

集群用于运行Docker应用，集群是应用运行所需要的云资源组合，关联了若干云服务器节点，即一个集群可以包

括若干结点，每个结点对应一个云服务器，在云服务器上运行了多个Docker容器，在Docker容器中运行着具体的

应用服务，下图是集群的结构图：

在默认情况下，每个用户最多能够创建5个集群。如果您想创建更多的容器集群，请拨打热线号码或发送电子邮件

到客户服务邮箱。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.2.2 注册云账号**

华为云注册账号 ：

1. 打开浏览器。

2. 登录华为云网站**www.huaweicloud.com**

3. 单击“注册”。

4. 在注册页面，根据提示信息完成注册。

注册成功后，系统会自动跳转至您的个人信息界面。

**5.2.3 集群基本设置**

1、使用云账号登录云平台

2、选择“服务列表“--》“云容器引擎CCE”

创建Kubernetes集群 ：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.2.4 创建虚拟私有云**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

选择虚拟私有云

**5.2.4 创建节点**

**5.2.4.1 付费类型设置**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.2.4.2 规则设置**

设置节点规格，本教程使用通用计算增强型服务器，8核16G内存。

**5.2.4.3 购买数量**

初次测试建议先购买一个节点。

**5.2.4.4 网络设置**

设置节点的网络

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.2.4.5 磁盘设置**

**5.2.4.6 登录设置**

设置登录方式：

1、采用密码登录

2、采用密钥对登录

本教程采用密钥对登录。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

创建密钥对成功会自动下载密钥文件（上图为KeyPair-32d5.pem）到本地，此文件需要保存好，用于远程登录。

**5.2.4.7 提交订单创建集群**

集群设置完毕，确定无误，提交订单。

等待6--10分钟集群创建成功：

**5.2.5 查询集群**

集群创建成功，可以通过查询集群功能查询：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.2.6 查询结点**

点击查询结点，查询集群所包括的结点数。

**5.2.7 弹性服务器**

弹性云服务器是由CPU、内存、镜像、云硬盘组成的一种可随时获取、弹性可扩展的计算服务器，同时它结合虚拟

私有云、虚拟防火墙、数据多副本保存等能力，为您打造一个高效、可靠、安全的计算环境，确保您的服务持久稳

定运行。在云容器引擎中一个节点就是具有多个云硬盘的一台弹性云服务器。

点击“服务列表”--》计算--》弹性云服务器

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

上图显示的弹性IP为49.4.87.154为创建集群时自动创建的公网IP地址，此IP绑定了弹性服务器结点。

**5.2.7 远程登录**

使用49.4.87.154公网IP地址即可登录弹性服务器。

使用SSH Client登录，设置登录参数：

连接登录：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

选择 Public Key，浏览选择 密钥文件。

登录成功：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.3 CCE快速体验**

**5.3.1 体验流程**

本章节以在集群中部署nginx服务为示例，体验华为云工作负载的创建过程。

完整的流程如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.3.2 设置环境**

第一步设置网络环境，“创建虚拟私有云”，用于安全网络隔离。此步骤在上一章节“创建集群”中已完成。

**5.3.3 创建集群**

第二步创建集群，此步骤在上一章节“创建集群”中已完成。

**5.3.4 创建无状态工作负载**

工作负载分为无状态工作负载和有状态工作负载，有状态工作负载是服务在运行过程中会持久化数据，nginx属于

无状态工作负载。

基于“nginx镜像”创建无状态工作负载，快速搭建nginx web服务器。

**5.3.4.1 填写基本信息**

1 登录CCE控制台，选择左侧导航栏的“工作负载-无状态（Deployment）”，单击“创建无状态工作负载”。

**5.3.4.2 添加容器**

单击“添加容器”，选择nginx镜像，单击“确定”，所有参数保持默认，单击“下一步”。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

选择镜像

填写容器参数

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.3.4.3 添加访问方式**

添加访问方式，通过公网访问nginx。

添加成功：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.3.4.4 创建工作负载成功**

**5.3.4.5 查询工作负载**

查询已创建的工作负载：

**5.3.4.6 测试**

通过工作负载的外部访问地址访问Nginx：http://49.5.87.154:31666

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.3.5 弹性伸缩**

**5.3.5.1 配置策略**

弹性伸缩（Auto Scaling）可根据用户的业务需求和预设策略，自动调整计算资源，使云服务器数量自动随业务负

载增长而增加，随业务负载降低而减少，保证业务平稳健康运行。

一个例子：按性能、按时间、按周期”的弹性伸缩策略，满足不同业务场景需求。此处以定时策略为例。

登录CCE控制台，选择”工作负载> 无状态（Deployment）”，单击nginx，进入详情页面。配置弹性伸缩策略。

点击“点击添加伸缩策略”，配置定时策略，可在系统负载较高的时候增加实例，可在系统负载低的时候减少负载。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

策略添加成功，一个是在9点43分增加实例，一个是在9点45分减少实例。

**5.3.5.2 伸缩验证**

1、在9点43分时云平台会自动增加一个nginx实例

登录服务器观察nginx容器实例，共有3个

2、在9点45分时云平台会自动减少一个nginx实例

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

登录服务器观察nginx容器实例，共有2个

**5.3.5.3 其它策略**

除了配置定时策略还可以配置告警策略、周期策略，下边介绍一下告警策略。

上图在指标中显示了很多的系统资源使用指标，比如可以根据内存利用率配置策略，当内存利用率大于80%时减少

一个实例，配置如下：

**5.3.6 关闭集群**

当不进行测试时建议将集群休眠，并将服务关机，因为节点和工作负载运行过程中会产生费用。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

集群休眠：

服务器关机：

**5.4 容器镜像服务**

容器镜像服务提供安全可靠的Docker容器镜像管理功能，支撑客户的容器化应用部署。

**5.4.1 镜像使用流程**

1、创建组织

组织的作用是对镜像进行分类管理，创建的每个组织对应一个公司、部门或个人，将其拥有的镜像集中在该组织

下。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、镜像获取

通过以下三种方式获取镜像：

1） 上传自有镜像

将本地服务器的镜像上传到镜像仓库。

2）源码构建镜像

从代码到镜像的一键式构建。可通过基于dockerﬁle标准的自动构建功能，敏捷高效地打通快速容器化服务的全部

环节。

3）使用镜像资源

从DockerHub官方拉取镜像到云平台。

3、应用部署

使用CCE容器引擎创建工作负载，参见快速体验。

4、更新镜像

当镜像更新后自动更新容器镜像。

**5.4.2 创建组织**

点击组织管理--》创建组织

组织创建成功，查询组织，当前已创建三个组织，如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**5.4.3 上传自有镜像**

这里介绍上传自有镜像，点击“我的镜像”--》“客户端上传”

操作步骤：

1、本地服务器安装Docker1.11.2版本以上

2、使用root登录本地服务器

3、复制上图 Step 2的docker login指令

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

4、在本地服务，执行docker login指令

5、将本地镜像上传到云平台镜像仓库

本测试我们先从公网拉取一个镜像到本地，再将该本地镜像上传到华为云平台镜像仓库

1）拉取一个测试镜像

docker pull swr.cn‐north‐1.myhuaweicloud.com/root/swr‐demo‐2048:latest

拉取成功，查询该镜像已存在本地docker镜像仓库

2）修改镜像的组织名，以便推送到个人组织内

docker tag swr.cn‐north‐1.myhuaweicloud.com/root/swr‐demo‐2048:latest swr.cn‐north‐

1.myhuaweicloud.com/xc‐edu/swr‐demo‐2048:latest

再次查询镜像信息：

docker image

3 ) 上传镜像

docker push swr.cn‐north‐1.myhuaweicloud.com/xc‐edu/swr‐demo‐2048:latest

4）上传成功，查询云平台镜像仓库

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6 学成在线项目部署**

**6.1 部署规划**

本章节会将学成在线项目前端、服务端等部署至华为云平台，最终实现在线点播视频的业务流程。

项目集群的整个结构图如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

容器说明：

1、前端容器包括：门户、搜索、学习服务。

门户使用Nginx、搜索采用Nuxt.js、学习服务采用Nginx

2、服务容器包括：EdgeService网关、portalview数据视图服务、search搜索服务、learning学习服务、

ElasticSearch全文检索服务。

3、数据层包括：MySQL数据库和MongoDB数据库。

**6.2 percona部署**

Percona Server为 MySQL 数据库服务器进行了改进,在功能和性能上较 MySQL 有着很显著的提升。本项目使用

Percona 作为MySQL数据库。

**6.2.1 镜像获取**

percona镜像使用DockerHub官方提供的。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.2.2 创建工作负载**

创建有状态工作负载，如下图：

**6.2.2.1 填写工作负载信息**

填写工作负载名称、选择实例数量等信息。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.2.2.2 选择镜像**

为工作负载添加容器，点击“添加容器”

2、选择镜像

3、填写镜像版本

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.2.2.3 填写容器规格**

注意：cpu和内存最小限制，尽量设置小一些，累计最小资源配置超过服务器资源配置则不允许再添加容器。

**6.2.2.4 添加docker数据卷**

percona的数据文件存储在宿主机中，下边设置宿主机数据文件的路径：

**磁盘源**

**挂载路径**

/usr/share/eduproject/data/mysql-data

/var/lib/mysql

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

点击确定，数据卷添加成功

**6.2.2.5 配置环境变量**

这里通过环境变量设置percona的密码，环境变量如下：

**变量名称**

**变量/变量引用**

MYSQL\_ROOT\_PASSWORD

mysql

**6.2.2.6 工作负载访问设置**

设置实例前访问的端口。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

添加外网访问方式

设置成功如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.2.2.7 创建成功**

有状态工作负载创建成功，通过有状态工作负载查询：

点击工作负载名称，在下图中点击 “访问方式”，查询工作负载的访问方式。

**6.2.2.8 通过外网访问**

上边创建的percona工作负载可以通过外网访问（ 49.4.87.154:30542）.

使用MySQL客户端连接

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

连接成功，查询数据库及表信息：

**6.2.2.9 导入数据**

导入资料文件夹下的"mysql"目录下的sql脚本，导入成功如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.3 MongoDB部署**

**6.3.1 镜像获取**

percona镜像使用DockerHub官方提供的。

**6.3.2 创建工作负载**

创建有状态工作负载，如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.3.2.1 填写工作负载信息**

**6.3.2.2 选择镜像**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.3.2.3 填写容器规格**

**6.3.2.4 添加docker数据卷**

**磁盘源**

**挂载路径**

/usr/share/eduproject/data/mongodb-data

/data/db

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.3.2.5 工作负载访问设置**

**6.3.2.6 创建成功**

**6.3.2.7 通过外网访问**

这里使用Studio 3T for MongoDB连接mongodb数据库，新建连接：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

连接成功：

**6.3.2.8 导入数据**

导入资料文件夹下 “mongodb\xc\_portalview”目录的脚本，导入成功如下：

**6.4 ElasticSearch部署**

**6.4.1 上传镜像**

ElasticSearch镜像在DockerHub中不存在，从ElasticSearch官方拉取镜像到本地服务器，再上传到云平台镜像仓

库。

1、登录本地服务器

2、从ElasticSearch官方拉取镜像到本地服务器

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

docker pull docker.elastic.co/elasticsearch/elasticsearch-platinum:6.2.1

3、上传镜像到云平台镜像仓库

docker tag docker.elastic.co/elasticsearch/elasticsearch-platinum:6.2.1 swr.cn-north-

1.myhuaweicloud.com/xc-edu/elasticsearch-platinum:6.2.1 docker push swr.cn-north-

1.myhuaweicloud.com/xc-edu/elasticsearch-platinum:6.2.1

**6.4.2 创建工作负载**

1、创建有状态工作负载

容器规格

2、数据卷

elasticsearch数据卷如下：

**磁盘源**

**挂载路径**

/usr/share/eduproject/elasticsearch/conﬁg/elasticsearch.yml

/usr/share/eduproject/elasticsearch/conﬁg/jvm.options

/usr/share/eduproject/elasticsearch/data

/usr/share/eduproject/elasticsearch/plugins/ik

/usr/share/elasticsearch/conﬁg/elasticsearch.yml

/usr/share/elasticsearch/conﬁg/jvm.options

/usr/share/elasticsearch/data

/usr/share/elasticsearch/plugins/ik

配置成功如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

具体的/conﬁg/elasticsearch.yml、/conﬁg/jvm.options、/plugins/ik等内容参见：资料文件夹下的“学成在线”--》

elasticsearch目录下的内容。

需要在服务器安装 yum install lrzsz -y，并将elasticsearch目录中的内容上传到服务。

需要在服务器安装 yum install -y unzip zip 将上传的zip包解压。

3、访问方式

**6.4.3 错误处理**

1、java.nio.ﬁle.AccessDeniedException: /usr/share/elasticsearch/data/nodes

进入弹性服务 的/usr/share/eduproject/elasticsearch目录

执行 chmod 777 \* 修改目录权限

2、max virtual memory areas vm.max\_map\_count [65530] is too low,

进入宿主机：

1、切换到root用户

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、修改配置sysctl.conf

vi /etc/sysctl.conf 在文件最下边添加配置： vm.max\_map\_count=655360 保存文件

3、执行命令sysctl -p使修改生效

4、 重启elasticsearch容器

**6.4.4 远程连接**

在本地使用elasticsearch-head（见资料文件夹下的elasticsearch目录）连接远程的elasticsearch服务

**6.4.5 初化数据**

参考“资料”--》“学成在线”--》elasticsearch下的脚本，使用postman导入初始化数据。

详细如下：

1、创建索引库

通过elasticsearch-head创建索引库 xc\_course，1个分片，0个副本。

2、使用postman工具创建映射，post请求：http://49.4.87.154:30968/xc\_course/doc/\_mapping

**注意：49.4.87.154:30968是访问ElasticSearch的外网地址，根据自己的环境进行修改。**

内容如下：

{

"properties" : {

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

"description" : {

"analyzer" : "ik\_max\_word",

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"grade" : {

"type" : "keyword"

},

"id" : {

"type" : "keyword"

},

"mt" : {

"type" : "keyword"

},

"name" : {

"analyzer" : "ik\_max\_word",

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"users" : {

"index" : false,

"type" : "text"

},

"charge" : {

"type" : "keyword"

},

"valid" : {

"type" : "keyword"

},

"pic" : {

"index" : false,

"type" : "keyword"

},

"qq" : {

"index" : false,

"type" : "keyword"

},

"price" : {

"type" : "float"

},

"price\_old" : {

"type" : "float"

},

"st" : {

"type" : "keyword"

},

"status" : {

"type" : "keyword"

},

"studymodel" : {

"type" : "keyword"

},

"teachmode" : {

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

"type" : "keyword"

},

"teachplan" : {

"analyzer" : "ik\_max\_word",

"search\_analyzer": "ik\_smart",

"type" : "text"

},

"expires" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

},

"pub\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

},

"start\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

},

"end\_time" : {

"type" : "date",

"format": "yyyy‐MM‐dd HH:mm:ss"

}

}

}

使用postman向elasticsearch添加初始数据

Post http://49.4.87.154:30968/xc\_course/doc/4028858162e0bc0a0162e0bfdf1a0000

{

"pub\_time":"2018‐07‐02 19:12:55",

"expires":null,

"mt":"1‐3",

"description":"课程大纲： 第一章： Python 介绍、基础语法、流程控制 第二章：Python 数据类型、字符编码学习

第三章： Python 文件操作、函数、装饰器第四章： Python 学用模块、开发规范学习",

"@timestamp":"2018‐07‐02T11:13:00.274Z",

"grade":"200002",

"charge":"203001",

"start\_time":null,

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

"timestamp":"2018‐07‐02T11:12:56.000Z",

"qq":"45323453",

"valid":"204001",

"end\_time":null,

"name":"Python开发入门",

"users":"零基础，多次从入门到放弃的小白，想要深入学习python的同学",

"id":"4028858162e0bc0a0162e0bfdf1a0000",

"teachplan":"{\"children\":[{\"children\":

[{\"grade\":\"3\",\"id\":\"8a7e867a645aa7c801645aad506a0001\",\"pname\":\"Python\u4e0e\u5176\u4ed

6\u8bed\u8a00\u7684\u533a\u522b\"}],\"grade\":\"2\",\"id\":\"8a7e867a645aa7c801645aace1f20000\",\

"pname\":\"Python\u4ecb\u7ecd\u3001\u57fa\u7840\u8bed\u6cd5\u3001\u6d41\u7a0b\u63a7\u5236\"},

{\"children\":

[{\"grade\":\"3\",\"id\":\"8a7e867a645aa7c801645aae02090003\",\"pname\":\"\u4e8c\u8fdb\u5236\u8f6

c\u6362\"},

{\"grade\":\"3\",\"id\":\"8a7e867a645aa7c801645aae5c4b0004\",\"pname\":\"ASCII\u7801\u4e0e\u4e8c\

u8fdb\u5236\"}],\"grade\":\"2\",\"id\":\"8a7e867a645aa7c801645aadc16c0002\",\"pname\":\"Python\u6

570\u636e\u7c7b\u578b\u3001\u5b57\u7b26\u7f16\u7801\u5b66\u4e60\"},{\"children\":

[{\"grade\":\"3\",\"id\":\"8a7e867a645aa7c801645aaf0cbd0006\",\"pname\":\"\u51fd\u6570‐

\u51fd\u6570\u8fdb\u9636‐

\u95ed\u5305\"}],\"grade\":\"2\",\"id\":\"8a7e867a645aa7c801645aaed7dc0005\",\"pname\":\"Python\u

6587\u4ef6\u64cd\u4f5c\u3001\u51fd\u6570\u3001\u88c5\u9970\u5668\"}],\"grade\":\"1\",\"id\":\"402

8858162e0bc0a0162e0bfdf2b0001\",\"pname\":\"\u4eba\u5de5\u667a\u80fd+python\"}",

"st":"1‐3‐6",

"@version":"1",

"price":0.009999999776482582,

"pic":"group1/M00/00/00/rBEAa1s96cOAaglmAAAfBQbuVNA131.jpg",

"studymodel":"201001",

"teachmode":null,

"price\_old":null

}

post

http://49.4.87.154:30968/xc\_course/doc/4028e58161bd22e60161bd23672a0001

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

{

"pub\_time":"2018‐07‐02 19:18:37",

"expires":null,

"mt":"1‐1",

"description":"如果你之前已经习惯了用jQuery操作DOM，学习Vue.js时请先抛开手动操作DOM的思维，因为Vue.js是

数据驱动的，你无需手动操作DOM。 它通过一些特殊的HTML语法，将DOM和数据绑定起来。 一旦你创建了绑定，DOM将和

数据保持同步，每当变更了数据，DOM也会相应地更新。",

"@timestamp":"2018‐07‐02T11:19:00.203Z",

"grade":"200002",

"charge":"203001",

"start\_time":null,

"timestamp":"2018‐07‐02T11:18:38.000Z",

"qq":"4324322",

"valid":"204002",

"end\_time":null,

"name":"vue2.0高级教程",

"users":"有一定的css，html，和js基础开发者，前端开发，传统web开发。",

"id":"4028e58161bd22e60161bd23672a0001",

"teachplan":"{\"children\":[{\"children\":

[{\"grade\":\"3\",\"id\":\"4028e58161bd269f0161bd2778750001\",\"pname\":\"第一节 vue基础\"},

{\"grade\":\"3\",\"id\":\"4028e58161bd269f0161bd27d7c50002\",\"pname\":\"第二节属性和事件、模板、交

互、案例\"}],\"grade\":\"2\",\"id\":\"4028e58161bd269f0161bd270a340000\",\"pname\":\"Vuejs第一讲

\"},{\"children\":[{\"grade\":\"3\",\"id\":\"4028e58161bd269f0161bd2877740005\",\"pname\":\"第一节

计算属性的使用、vue实例的简单方法\"},

{\"grade\":\"3\",\"id\":\"4028e58161bd269f0161bd293df90006\",\"pname\":\"第二节自定义过滤器、自定义

指令\"}],\"grade\":\"2\",\"id\":\"4028e58161bd269f0161bd281bde0003\",\"pname\":\"Vuejs第二讲\"},

{\"children\":[],\"grade\":\"2\",\"id\":\"4028e58161bd269f0161bd284bad0004\",\"pname\":\"Vuejs第三

讲\"}],\"grade\":\"1\",\"id\":\"4028e58161bd22e60161bd2366fb0000\",\"pname\":\"Javascript之

VueJS\"}",

"st":"1‐1‐9",

"@version":"1",

"price":0.009999999776482582,

"pic":"group1/M00/00/00/rBEAa1s‐JPyATixdAABhNYbB2oM761.jpg",

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

"studymodel":"201001",

"teachmode":null,

"price\_old":null

}

post

http://49.4.87.154:30968/xc\_course/doc/402885816243d2dd016243f24c030002

{

"pub\_time":"2018‐04‐25 19:11:35",

"expires":null,

"mt":"1‐3",

"description":"Bos物流项目2.0是传智播客推出的高级项目课程，课程包括了项目概述、系统管理、用户管理、运单管

理、统计分析等模块，深入浅出的讲解了当前的流行技术。",

"@timestamp":"2018‐07‐02T11:26:00.058Z",

"grade":"200001",

"charge":"203001",

"start\_time":null,

"timestamp":"2018‐07‐03T10:28:58.000Z",

"qq":null,

"valid":"204002",

"end\_time":null,

"name":"Bos物流项目2.0",

"users":"本项目适应于具有一定java基础的开发人员。",

"id":"402885816243d2dd016243f24c030002",

"teachplan":"{\"children\":[{\"children\":

[{\"grade\":\"3\",\"id\":\"297e02f7639af61a01639afd3a7b0000\",\"pname\":\"业务需求分析\"},

{\"grade\":\"3\",\"id\":\"8a7e867a6459d24c016459e507260000\",\"pname\":\"订单业务分析

\"}],\"grade\":\"2\",\"id\":\"402885816243d2dd016243f24c040003\",\"pname\":\"项目概述\"},

{\"children\":

[{\"grade\":\"3\",\"id\":\"8a7e867a6459d24c016459e745410001\",\"pname\":\"ElasticSearch下载安装

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

\"},{\"grade\":\"3\",\"id\":\"8a7e867a6459d24c016459e7cac90002\",\"pname\":\"ElasticSearch入门案例

\"}],\"grade\":\"2\",\"id\":\"1011\",\"pname\":\"运单管理

\"}],\"grade\":\"1\",\"id\":\"1012\",\"pname\":\"Bos物流项目课程\"}",

"st":"1‐3‐2",

"@version":"1",

"price":98,

"pic":"group1/M00/00/00/rBEAa1s‐JKqAGRsQAAJx5ZjNDEM811.jpg",

"studymodel":"201001",

"teachmode":null,

"price\_old":199

}

Post http://49.4.87.154:30968/xc\_course\_media/doc/40281f8164553706016455390e140001

{

"courseid": "297e7c7c62b8aa9d0162b8ab56ba0001",

"media\_id": "595e4d0fca23387ca9b5ded130630909",

"media\_url": "5/9/595e4d0fca23387ca9b5ded130630909/hls/595e4d0fca23387ca9b5ded130630909.m3u8",

"teachplan\_id": "40281f8164553706016455390e140001",

"timestamp": "2018‐07‐07T15:51:59.000Z",

"media\_fileoriginalname": "01Vue基础‐ MVVM模式讲解.avi"

}

Post http://49.4.87.154:30968/xc\_course\_media/doc/40281f81645537060164553a783f0005

{

"courseid": "297e7c7c62b8aa9d0162b8ab56ba0001",

"media\_id": "53ac4cca3ddf386c21f4f1cbb4dc9876",

"media\_url": "5/3/53ac4cca3ddf386c21f4f1cbb4dc9876/hls/53ac4cca3ddf386c21f4f1cbb4dc9876.m3u8",

"teachplan\_id": "40281f81645537060164553a783f0005",

"timestamp": "2018‐07‐07T15:51:59.000Z",

"media\_fileoriginalname": "3.avi"

}

Post http://49.4.87.154:30968/xc\_course\_media/doc/40281f81645537060164553a29d80004

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

{

"courseid": "297e7c7c62b8aa9d0162b8ab56ba0001",

"media\_id": "809694a6a974c35e3a36f36850837d7c",

"media\_url": "8/0/809694a6a974c35e3a36f36850837d7c/hls/809694a6a974c35e3a36f36850837d7c.m3u8",

"teachplan\_id": "40281f81645537060164553a29d80004",

"timestamp": "2018‐07‐07T15:51:59.000Z",

"media\_fileoriginalname": "1.avi"

}

Post http://49.4.87.154:30968/xc\_course\_media/doc/40281f81645537060164553972790002

{

"courseid": "297e7c7c62b8aa9d0162b8ab56ba0001",

"media\_id": "1cfabb64792eaf2a0d9d72d237e07256",

"media\_url": "1/c/1cfabb64792eaf2a0d9d72d237e07256/hls/1cfabb64792eaf2a0d9d72d237e07256.m3u8",

"teachplan\_id": "40281f81645537060164553972790002",

"timestamp": "2018‐07‐07T15:51:59.000Z",

"media\_fileoriginalname": "03Vue基础 ‐ 系统指令v‐on注册事件.avi"

}

**6.5 微服务部署**

**6.5.1 上传镜像**

**6.5.1.1 上传流程**

微服务镜像需要在本地服务器使用Dockerﬁle文件制作，在本地服务器创建镜像文件，再使用docker login向云平

台镜像仓库上传，流程如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

1、软件工程师在本地进行软件开发，修改代码

2、软件工程师将代码提交到git仓库（GitLab或GitHub等）

3、在本地服务器从git仓库获取工程代码

4、使用maven插件进行工程构建，并创建docker镜像

5、执行docker tag修改镜像的组织名，以便推送到个人组织内

6、执行docker push 将本地服务器上的镜像推送到云平台镜像仓库。

**6.5.1.2 本地服务器环境**

本地服务器通常在公司局域网部署，本地服务器需具有项目工程构建的环境，如下：

1、jdk1.8

2、maven3.3.0以上

3、Git

4、Docker1.11.2版本以上

**6.5.1.3 构建脚本**

为了方便镜像构建，特编写shell脚本，下边以xc-service-portalview为例，在xc-service-portalview工程根目录下

定义image\_build.sh脚本文件，如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

image\_build.sh的内容如下：

#!/bin/bash

#删除原有xc‐service‐portalview镜像

docker rmi xc‐service‐portalview:1.0‐SNAPSHOT

docker rmi swr.cn‐north‐1.myhuaweicloud.com/xc‐edu/xc‐service‐portalview:$1

#将parent、model、utils、common基础工程打包，上传到maven本地仓库

mvn ‐f ../xc‐framework‐parent/pom.xml install

mvn ‐f ../xc‐framework‐model/pom.xml install

mvn ‐f ../xc‐framework‐utils/pom.xml install

mvn ‐f ../xc‐framework‐common/pom.xml install

#使用maven插件构建docker镜像

mvn ‐f pom\_docker.xml clean package ‐DskipTests docker:build

#修改镜像的组织名，以便推送到个人组织内，注意这里xc‐edu‐cloud‐01是新添加的组织名

docker tag xc‐service‐portalview:1.0‐SNAPSHOT swr.cn‐north‐1.myhuaweicloud.com/xc‐edu‐cloud‐

01/xc‐service‐portalview:$1

#执行docker push 将本地服务器上的镜像推送到云平台镜像仓库

docker push swr.cn‐north‐1.myhuaweicloud.com/xc‐edu‐cloud‐01/xc‐service‐portalview:$1

**6.5.1.4 Dockerﬁle**

上边构建脚本中 使用maven插件执行docker:build构建镜像，此构建需要编写Dockerﬁle文件

在src/main/resouces下添加Dockerﬁle文件

文件内容如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

FROM java:8

ENV ARTIFACTID xc‐service‐portalview

ENV ARTIFACTVERSION 1.0‐SNAPSHOT

ENV HOME\_PATH /home

WORKDIR $HOME\_PATH

ADD /$ARTIFACTID‐$ARTIFACTVERSION.jar $HOME\_PATH/$ARTIFACTID.jar

ENTRYPOINT ["java", "‐jar", "xc‐service‐portalview.jar"]

**6.5.1.5 配置maven插件**

新增pom\_docker.xml文件：

文件内容是在原有pom.xml的基础上增加build标签，如下定义了docker-maven-plugin插件创建镜像的参数：

<build>

<finalName>${project.artifactId}‐${project.version}</finalName>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐maven‐plugin</artifactId>

</plugin>

<plugin>

<groupId>com.spotify</groupId>

<artifactId>docker‐maven‐plugin</artifactId>

<version>1.0.0</version>

<!‐‐docker镜像相关的配置信息‐‐>

<configuration>

<!‐‐镜像名，这里用工程名‐‐>

<imageName>${project.artifactId}‐${project.version}</imageName>

<!‐‐Dockerfile文件所在目录‐‐>

<dockerDirectory>${project.basedir}/src/main/resources</dockerDirectory>

<!‐‐TAG,这里用工程版本号‐‐>

<imageTags>

<imageTag>${project.version}</imageTag>

</imageTags>

<imageName>${project.artifactId}:${project.version}</imageName>

<!‐‐构建镜像的配置信息‐‐>

<resources>

<resource>

<targetPath>/</targetPath>

<directory>${project.build.directory}</directory>

<include>${project.artifactId}‐${project.version}.jar</include>

</resource>

</resources>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

</configuration>

</plugin>

</plugins>

</build>

**6.5.1.6 执行构建**

1、登录本地服务器

2、git clone 将git仓库的工程代码下载到本地服务器

3、进入xc-service-portalview目录，执行sh image\_build.sh

执行成功，镜像上传到云平台镜像仓库

登录云平台，查询镜像仓库：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.5.2 Portalview部署**

创建无状态工作负载：

**6.5.2.1 填写工作负载信息**

**6.5.2.2 选择镜像**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.5.2.3 填写容器规格**

**6.5.2.4 配置MongoDB连接地址**

xc-service-portalview需要连接mongodb数据库，在src/main/resources/application.yml配置文件中，配置如

下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

1、首先配置mongodb集群内部的访问方式

添加成功，如下mongodb在集群内容的访问地址为：10.247.102.74:27017

2、在创建xc-service-portalview工作负载中配置环境变量

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.4.2.5 工作负载访问设置**

为了方便在外网测试xc-service-portalview接口，这里配置xc-service-portalview可以外网访问

**6.5.2.6 创建成功**

观察运行日志，直接通过图形界面查看日志：

观察服务目录是否注册成功：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

并且可以查看服务的接口契约：

**6.5.2.7 测试**

浏览器请求：http://49.4.87.154:30984/portalview/course/get/4028e581617f945f01617f9dabc40000

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.5.2.8 自动升级**

本节实现代码修改后如何将修改后更新到docker容器中，云平台镜像仓库提供触发器功能，当镜像修改后更新相应

容器的镜像。

1、创建触发器

创建成功：

修改代码再次上传影响流程如下：

1、修改代码。

2、重新创建镜像并上传。

注意：镜像的版本号要增加。

为了方便构建镜像将脚本更改为可以传参数：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

#!/bin/bash

docker rmi xc‐service‐portalview:1.0‐SNAPSHOT

docker rmi swr.cn‐north‐1.myhuaweicloud.com/xc‐edu/xc‐service‐portalview:$1

mvn ‐f xc‐framework‐parent/pom.xml install

mvn ‐f xc‐framework‐model/pom.xml install

mvn ‐f xc‐framework‐utils/pom.xml install

mvn ‐f xc‐framework‐common/pom.xml install

mvn ‐f xc‐service‐portalview/pom\_docker.xml clean package ‐DskipTests docker:build

docker tag xc‐service‐portalview:1.0‐SNAPSHOT swr.cn‐north‐1.myhuaweicloud.com/xc‐edu‐cloud‐

01/xc‐service‐portalview:$1

docker push swr.cn‐north‐1.myhuaweicloud.com/xc‐edu‐cloud‐01/xc‐service‐portalview:$1

在执行镜像构建脚本时可以将版本号作为参数传入，一个例子：

sh image\_build.sh 1.0.5‐SHAPSHOT

3、观察工作负载实例，当镜像更新后自动重启实例

**6.5.3 Search部署**

1、xc-service-search微服务在部署时必须是配置ElasticSearch的地址。

查看xc-service-search工程的application.yml配置文件,ElasticSearch地址配置如下

ElasticSearch地址需要查看ElasticSearch容器的访问方式，取集群内访问方式即可：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

在创建Search容器时需要配置环境变量如下：

名称：ES\_HOSTLIST

值：10.247.141.21:9200

2、配置成功，查看访问方式

3、测试

Get http://49.4.87.154:32476/search/course/list/1/2

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.5.4 Learning部署**

xc-service-learning微服务在部署时主要是配置mysql数据库的地址。

1、查看xc-service-learning的application.yml配置文件，如下这里配置了mysql的连接url：

2、查看percona的访问方式

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

3、在部署xc-service-learning时需要配置环境变量如下：

名称：MYSQL\_URL

值：jdbc:mysql://10.247.183.70:3306/xc\_learning?characterEncoding=utf-8

4、配置成功，查看访问方式

5、测试

获取课程计划的视频播放地址：

http://49.4.87.154:31576/learning/getmedia/402885816243d2dd016243f24c030002/297e02f7639af61a0163

9afd3a7b0000

**6.5.6 网关部署**

参考前边的微服务部署流程，向云平台部署EdgeService。

**6.6 前端部署**

**6.6.1 门户**

**6.6.1.1 镜像获取**

从dockerHub中获取Nginx 1.14.0镜像

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.6.1.2 创建工作负载**

1、创建无状态工作负载，名称为nginx-portal。

配置数据卷如下：

‐v/usr/share/eduproject/nginx‐portal/log/:/var/log/nginx \

‐v/usr/share/eduproject/nginx‐portal/nginx.conf:/etc/nginx/nginx.conf \

‐v/usr/share/eduproject/nginx‐portal/xc\_portal\_static/:/etc/nginx/html \

2、根据上边数据卷的设置需要配置nginx.conf及上传程代码到指定目录：

将“资料”--》学成在线\前端\nginx-portal\xc-ui-pc-static-portal.zip 上传到云服务器。

将xc-ui-pc-static-portal.zip内容解压到/usr/share/eduproject/nginx-portal/xc\_portal\_static下边。

将“资料”--》学成在线\前端\nginx-portal\nginx.conf 上传到云服务器。

将nginx.conf 拷贝到/usr/share/eduproject/nginx-portal/目录下，覆盖原有nginx.conf。

将“资料”--》学成在线\static.zip和“资料”--》学成在线\video.zip上传到云服务器。

将static.zip中的内容解压到/usr/share/eduproject/nginx-portal/xc\_portal\_static目录下。

将video.zip中的内容解压到/usr/share/eduproject/nginx-portal/xc\_portal\_static目录下。

**6.6.1.3 访问**

1、申请弹性负载均衡

由于门户通过外网IP的80端口访问，这里需要申请弹性负载均衡及外网IP地址。

2、配置nginx.conf

弹性负载均衡及外网IP申请成功，比如这里申请的外网IP地址为49.4.93.176，需要配置nginx.conf：

修改/usr/share/eduproject/nginx-portal/nginx.conf

修改 名称为xuechengcloud.java.itcast.cn的虚拟主机，添加server\_name 49.4.93.176。

3、配置hosts

由于系统通过域名来访问，为了方便测试，这里通过修改本地hosts文件来实现域名访问，此方法测试通过后可通

过域名提供商修改公网DNS，指定域名所对应的公网IP（这里是上边申请的弹性负载均衡所对应的公网IP）

修改本地hosts文件，将域名指向弹性负载均衡IP。

49.4.93.176 xuechengcloud.java.itcast.cn 49.4.93.176 xcucentercloud.java.itcast.cn 49.4.93.176

xcsystemcloud.java.itcast.cn 49.4.93.176 xcteachercloud.java.itcast.cn 49.4.93.176 xcimgcloud.java.itcast.cn

49.4.93.176 xcvideocloud.java.itcast.cn

修改完毕重启nginx-portal。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

访问：http://xuechengcloud.java.itcast.cn

**6.6.2 学习中心**

**6.6.2.1 镜像获取**

从dockerHub中获取Nginx 1.14.0镜像

**6.6.2.2 创建工作负载**

1、创建无状态工作负载，名称为nginx-ucenter。

配置数据卷如下：

‐v /usr/share/eduproject/nginx‐ucenter/log/:/var/log/nginx \

‐v /usr/share/eduproject/nginx‐ucenter/nginx.conf:/etc/nginx/nginx.conf \

‐v /usr/share/eduproject/nginx‐ucenter/xc‐ui‐pc‐learning/:/etc/nginx/html \

2、根据上边数据卷的设置需要配置nginx.conf及上传程代码到指定目录：

将“资料”--》学成在线\前端\nginx-ucenter\打包\xc-ui-pc-leanring.zip 上传到云服务器。

将xc-ui-pc-leanring.zip的内容解压到/usr/share/eduproject/nginx-ucenter/xc-ui-pc-learning目录下。

将“资料”--》学成在线\前端\nginx-ucenter\nginx.conf上传到云服务器。

将nginx.conf拷贝到/usr/share/eduproject/nginx-ucenter/下。

**6.6.2.3 访问**

学习中心前端通过门户nginx代理访问，所以无需配置弹性负载均衡。

在门户的nginx.conf配置学习中心虚拟主机：

#前端ucenter

upstream ucenter\_server\_pool{

server 10.247.7.229:13000 weight=10;

}

#学成网用户中心

server {

listen

80;

server\_name xcucentercloud.java.itcast.cn;

#个人中心

location / {

proxy\_pass http://ucenter\_server\_pool;

}

...

重启学习中心和门户实例，访问下边的地址：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

http://xcucentercloud.java.itcast.cn/#/learning/402885816243d2dd016243f24c030002/0

**6.6.3 搜索**

**6.6.3.1 创建镜像**

编写 Dockerﬁle：

FROM node:9.4.0

MAINTAINER mrt

ENV NODE\_ENV=production

ENV HOST 0.0.0.0

RUN mkdir ‐p /app

COPY . /app

WORKDIR /app

# Expose the app port

COPY ./app‐entrypoint.sh /

RUN chmod +x /app‐entrypoint.sh

EXPOSE 10000

RUN npm config set registry https://registry.npm.taobao.org

RUN npm config set user 0

RUN npm config set unsafe‐perm true

RUN npm install cnpm ‐g

RUN cnpm install nuxt@1.3.0 ‐g

ENTRYPOINT ["/app‐entrypoint.sh"]

在Dockerﬁle的当前目录编写app-entrypoint.sh脚本，内容如下：

#!/bin/bash

cd /app

nuxt build

npm start

在Dockerﬁle的当前目录：docker build -t nuxt1.3 . 创建镜像

docker tag nuxt1.3:latest swr.cn-north-1.myhuaweicloud.com/xc-edu-cloud-01/nuxt-search:latest

docker push swr.cn-north-1.myhuaweicloud.com/xc-edu-cloud-01/nuxt-search:latest

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.6.3.2 创建工作负载**

创建无状态工作负载：

添加数据卷：

将搜索前端工程目录所在路径挂载到磁盘，方便代码的升级。

**磁盘源**

**挂载路径**

/usr/share/eduproject/nuxt-search

/app

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

将“学成在线\前端\nuxt-search”下的xc-ui-pc-portal.zip和node\_modules.tar上传代码

到/usr/share/eduproject/nuxt-search。

解压xc-ui-pc-portal.zip和node\_modules.tar，/usr/share/eduproject/nuxt-search目录内容如下：

**6.6.3.4 访问**

搜索前端通过门户nginx代理访问，所以无需配置弹性负载均衡。

在门户的nginx.conf配置搜索代理路径：

#前端搜索

upstream dynamic\_portal\_server\_pool{

server 10.247.141.109:10000 weight=10;

}

#学成网门户

server {

listen

80;

server\_name

......

49.4.93.176 xuechengcloud.java.itcast.cn;

#前端门户课程搜索

location ^~ /course/search {

proxy\_pass http://dynamic\_portal\_server\_pool;

}

......

访问：http://xuechengcloud.java.itcast.cn/course/search

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**6.6.4 前端与微服务集成**

**6.6.4.1 集成说明**

按照学成在线微服务的架构，前端通过网关请求访问微服务，如下图所示：

1、前端通过门户Nginx访问网关

2、网关将请求转发到具体的微服务。

**6.6.4.2 集成搜索服务**

1、添加网关内网访问方式

由于前端通过门户nginx内网转发请求到网关，这里配置网关的内网访问方式：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、在门户Nginx配置搜索转发地址

#后台api网关

upstream api\_server\_pool{

#weight是配置权重，根据权重值进行负载

server 10.247.171.174:50201 weight=10;

}

#学成网门户

server {

listen

80;

server\_name 49.4.84.18 xuechengcloud.java.itcast.cn;

......

#后端搜索服务

location ^~ /openapi/search/ {

proxy\_pass http://api\_server\_pool/openapi/search/search/;

proxy\_set\_header Host $host;

proxy\_set\_header X‐Real‐IP $remote\_addr;

proxy\_set\_header X‐Forwarded‐For $proxy\_add\_x\_forwarded\_for;

}

......

3、重启门户Nginx

访问：http://xuechengcloud.java.itcast.cn/course/search

**6.6.4.3 集成学习服务**

1、添加网关内网访问方式

前边集成搜索服务章节中已配置。

2、在门户Nginx的学习中心虚拟主机中配置学习服务、搜索及数据视图的转发地址

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

#学成网用户中心

server {

listen

80;

server\_name xcucentercloud.java.itcast.cn;

#学习服务

location ^~ /api/learning/ {

proxy\_pass http://api\_server\_pool/api/xc‐service‐learning/learning/;

}

#搜索服务

location /openapi/search/ {

proxy\_pass http://api\_server\_pool/openapi/search/search/;

}

#数据视图查询服务

location ^~ /openapi/portalview/ {

proxy\_pass http://api\_server\_pool/openapi/portalview/portalview/;

}

......

**6.7 小结**

学成在线部署云容器引擎总结：

1、镜像获取

1）私有镜像

2）从DockerHub获取镜像

3）从源代码构建镜像

2、创建工作负载

包括有状态工作负载、无状态工作负载。

3、配置访问方式

内网访问、外网访问，其中外网访问必须添加弹性负载均衡方可走外网IP的80端口。

4、更新升级

注意：构建镜像的版本要自增。

选择新版本的镜像执行升级。

**7 微服务治理**

**7.1 微服务治理介绍**

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**7.1.1 什么是微服务治理**

微服务开发上线后由于数量众多，并且微服务之间调用关系复杂，需要对微服务进行统一治理，微服务治理提供了

熔断、容错、限流、降级等高级服务治理能力，最大限度保障服务的可用性。

**7.1.2 云平台如何微服务治理**

在学习ServiceComb时采用微服务治理都需要我们在配置文件中配置治理策略，而采用云平台的方式进行微服务

治理只需要在微服务中配置少量的必要配置，具体的治理参数通过云平台配置，微服务也会把自已的信息上报给云

平台，并且微服务通过配置中心获取治理参数，从而实现通过云平台对微服务进行治理。

如何学习微服务治理：

1、测试阶段 可以在本地开发环境运行微服务，接入CSE平台，在云平台配置微服务治理的参数，观察微服务治理

的效果。

2、生产阶段将微服务部署到云平台，并接入CSE平台。

**7.2 准备工作**

**7.2.1 连接配置中心**

微服务要通过云平台治理需要连接服务注册中心和配置中心，在云平台设置微服务治理参数，通过配置中心将配置

信息下发到微服务。

在微服务中配置服务注册中心和配置中心：

在microservice.yaml中配置：

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com

instance:

watch: false

config:

client:

serverUri: https://cse.cn‐north‐1.myhuaweicloud.com

refreshMode: 1 #默认为定时拉取

refresh\_interval: 30 #客户端主动从配置中心拉取配置的周期 单位秒

**7.2.2 配置Handler**

服务治理主要涉及熔断、容错、限流、负载均衡等。

需要增加下面治理相关的handler。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

cse:

handler:

chain:

Provider:

default: bizkeeper‐provider,qps‐flowcontrol‐provider

Consumer:

default: bizkeeper‐consumer,loadbalance,qps‐flowcontrol‐consumer

每个Handler就是独立的处理模块，CSE提供如下的Handler：

**handler-loadbalance**

handler-bizkeeper

handler-tracing

handler-tcc

**负载均衡模块。提供各种路由策略和配置方法。一般客户端使用。**

和服务治理相关的功能，比如隔离、熔断、容错。

调用链跟踪模块，对接监控系统，吐出打点数据。

提供TCC事务开发管理能力。

**7.3 负载均衡策略**

**7.3.1 策略介绍**

当出现访问量和流量较大，一台服务器无法负载的情况下，我们可以通过设置负载均衡的方式将流量分发到多个服

务器均衡处理，从而优化响应时长，防止服务器过载。

可以通过新增规则配置负载均衡策略，设置参数支持轮询、随机、响应时间权值、会话保持等多种负载均衡策略。

**7.3.2 策略实现**

设置负载均衡：

1、进入服务目录

2、单击需要治理的微服务，进入微服务治理配置页面。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

3、在微服务治理配置页面，单击“负载均衡”，展开负载均衡详情页。

4、单击“新增”，进入负载均衡策略配置。先选择需要治理的微服务，再选择合理的负载均衡策略，请参见下表。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**策**

**略**

**名**

**策略说明**

轮

询

支持按照服务实例的位置信息顺序路由。

提供服务实例随机路由。

随

机

响

应

时

间

权

值

提供最小活跃数（时延）的权重路由，支持业务处理慢的服务实例接收较少的请求，防止系统停止响

应。这种负载均衡策略适合请求量少且稳定的应用。

会

话

粘

滞

会话粘滞是负载均衡器上的一种机制，在设定的会话保持时间内，会保证同一用户相关联的访问请求

会被分配到同一实例上。会话保持时间：会话保持的限制时间，0-86400，单位为秒。失败次数阈

值：访问失败次数，0-10。当微服务访问下属实例的失败次数和会话保持时间超过设定的值时，微服

务不再访问该实例。

**7.4 限流策略**

**7.4.1 策略介绍**

限流主要解决微服务之间的流量分配问题，保证微服务在自己的资源池运行，互不影响。当限流对象对当前服务实

例的每秒请求数量超过设定的值，当前服务实例不再接受该对象的请求。

常用的检测方法是请求超时、流量过大等，设置参数包括限流对象、QPS阈值等。

**7.4.2 策略实现**

使用云平台服务治理功能，通过图形界面完成限流设置，如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**7.5 容错策略**

**7.5.1 策略介绍**

容错是服务调用者访问服务实例，服务实例出现异常时的一种处理策略，出现异常后按照定义的策略进行重试或访

问新的服务实例。一旦发生异常，服务会根据容错机制来进行尝试重新访问或直接返回失败。

**7.5.2 策略实现**

设置容错：

1. 单击需要治理的微服务，进入微服务治理配置页面。

2. 服务治理配置页面，单击“容错”，展开容错详情页。

3. 单击“新增”，进入容错策略配置页面。

4. 在容错策略配置页面选择合理的策略，容错策略配置项如下表所示。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**配置项**

**配置项说明**

容错对象

该应用依赖的应用或方法，下拉菜单可直接选择。

开启：向容错对象发起请求时发生错误的处理策略，开启后，会根据选择

的处理策略处理请求。关闭：关闭容错策略，即使请求失败也会等到超时

后，再返回失败结果。

是否开启容错

容错策略说明：当“是否

开启容错”配置项设置

为“开启”时配置

Failover：在不同服务器上重新尝试建立连接。

同上

同上

Failfast：不再重新尝试建立连接，即请求失败时会立即返回失败结果。

Failback：在同一个服务器上重新尝试建立连接。

custom：同一个服务器上尝试重新建立连接的次数，取值范围0-9的整

数。在不同服务器上尝试建立连接的次数，取值范围0-9的整数。

同上

测试：

1、先设置负载均衡

2、断掉一个服务，测试发现远程连接

3、容错设置为Failback或Failover

**7.6 降级策略**

**7.6.1 策略介绍**

降级是容错的一种特殊形式，当出现服务吞吐量巨大，资源不够用等情况，我们可使用降级机制关掉部分不重要、

性能较差的服务，避免占用资源，以保证主体业务功能可正常使用。

**7.6.2 策略实现**

设置降级：

1. 单击需要治理的微服务，进入微服务治理配置页面。

2. 在服务治理配置页面，单击“降级”，展开降级详情页。

3. 单击“新增”，进入降级策略配置页面。

4. 在降级策略配置页面选择合理的策略，降级策略配置项如下表所示。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**配置项**

**配置项说明**

降级对象

降级策略

选择需要降级的服务。

开启：开启降级关闭：关闭降级

**7.7 熔断策略**

**7.7.1 策略介绍**

当我们发现由于某些原因导致服务出现了过载现象，为避免造成整个系统故障，可采用熔断来进行保护。熔断在服

务请求处理出现异常时产生作用。进入熔断状态后，hystrix会认为被请求的服务已经无法处理请求，在第一时间截

断请求直接返回错误给调用者。hystrix每隔一段时间会尝试访问后端服务，如果服务恢复正常，会退出熔断状态，

恢复正常的请求访问。

**7.7.2 策略实现**

设置熔断：

1. 单击需要治理的微服务，进入微服务治理配置页面。

2. 在服务治理配置页面，单击“熔断”，展开熔断详情页。

3. 单击“新增”，进入熔断策略配置页面。

4. 在熔断策略配置页面选择合理的策略，熔断策略配置项如下表所示。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

1、手动熔断

设置手动相当于实现降级功能。

2、取消熔断相关于删除熔断

3、自动熔断

测试流程为：设置失败率10、窗口请求数为10，熔断时间窗为2分钟，断掉服务，触发熔断，此时再启动服务也需

要等待2分钟才可恢复

**7.8 错误注入策略**

**7.8.1 策略介绍**

错误注入策略用于测试微服务的容错能力，可以让用户知道，当出现延时或错误时，系统是否能够正常运行。

错误注入通过延时、错误等方式，供用户测试微服务的容错能力。

**7.8.2 策略实现**

需要在消费微服务中配置hander： fault-injection-consumer

例如：

handler:

chain:

Provider:

default: bizkeeper‐provider,qps‐flowcontrol‐provider

Consumer:

default: bizkeeper‐consumer,loadbalance,qps‐flowcontrol‐consumer,fault‐injection‐consumer

测试：

通过错误注入来测试熔断功能，思路如下：

1、通过注入触发消费方请求延时。

向网关设置错误注入，如下图，表示网关请求portalview出现3600ms延时。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2、向网关设置消费超时时间，当错误注入触发延迟，由于网关消费超时时间小于上图设置的3600ms，则请求失

败。

cse:

isolation:

Consumer:

timeout:

enabled: true

timeoutInMilliseconds: 2000 #消费者超时时间

3、设置网关熔断策略，如下图，表示熔断后将停止90s时间。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

4、熔断触发后将错误注入取消，并观察90s后恢复正常。

**8 Mesher**

**8.1 微服务落地困难**

随着互联网、移动互联网的发展，微服务在国内从2015年兴起，2017/2018微服务热度很高，因为它解决了企业

的实际问题，企业将一个系统的业务处理拆分成一个一个微小的服务，服务与服务之间的耦合度非常低，大大提高

了系统开发的生产力，借助云计算的能力，可自动化管理微服务开发、容器化部署、监控，并完在服务容错、限

流、降低等微服务治理。

当前很多传统企业慢慢向互联网技术转型，其中微服务就是技术转型的核心，微服务在落地时遇到了很大的困难！

微服务落地是什么意思？

微服务落地就是企业的软件系统采用微服务技术构建，开发成为云原生应用。

什么是云原生（Cloud Native）应用？

CNCF是一个开源Linux基金会（https://www.cncf.io/），它致力于推进云端原生应用和服务的开发，由CNCF孵化

的项目目前达到20多个，其中包括当前热门的Kubernetes、Prometheus等；CNCF的成员包括：google、IBM、

英特尔、阿里巴巴、华为、思科、Docker、eBay、红帽等一百多家。

CNCF定义的云原生应用包括以下特征：

1、面向微服务：面向微服务开发、测试、运维。

2、容器化封装：在容器中运行应用程序，并作为应用程序部署的独立单元，最终应用程序运行在公有云或私有云

中。

3、自动化管理：使用自动化的管理平台对容器部署、运维进行统一调度，提高系统的资源利用率，降低运维成

本。

传统企业微服务落地困难的原因有哪些？

1、技术门槛高

当前流行的微服务框架的技术门槛过高，如SpringCloud，SpringCloud出于名门望族的Spring家族，如果要使用

Spring Cloud构建微服务需要掌握Zuul、Hystrix、Ribbon、Feign、Eureka、Zipkin、Turbine等组件，并且需要

深入学习每个组件的特性，对于一个新手来说需要投入的学习成本是很高的。

2、缺人才

传统企业中缺乏互联网技术人才，比如：他们对微服务、容器等知识匮乏，这就是微服务落地难的最大问题。

3、旧系统改造困难

对于老系统改造为微服务方式存在问题，老系统开发语言多种多样，如果采用统一的微服务框架重新开发则成本巨

大。

面对以上问题企业如何构建微服务系统？

1、对员工进行技术培训。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

成本高。

2、挖人

互联网公司做事的态度，财大气粗，你呢？

3、外包

将软件项目外包给第三方的微服务开发团队。这个项目做完，下一个项目呢？

**8.2 Service Mesh**

**8.2.1 什么是Service Mesh**

要彻底解决微服务落地的问题需要转换思路，Service Mesh给出了答案,Service Mesh翻译为“服务网格”，它是一

种微服务开发的理念。

微服务开发要支持代码0侵入则可大大降低了微服务开发的难度，提高微服务落地的速度，即不必使用统一的微服

务框架进行开发，对于一些老系统不论开发语言是什么0代码侵入则可以改为微服务。

Service Mesh采用Sidecar模式，由一系列轻量级的网络代理组成，Sidecar翻译为边车模式，可以想象为在摩托车

旁边加装座椅，如下图：

加装了坐骑此摩托车的功能就增强了，开车的人专心开车，坐车的人可以专心照相，欣赏风景。

Sidecar在Service Mesh中如何应用？

下边是一个服务消费方采用service mesh的例子：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

Service Mesh与Service A处于同一个Pod，Pod是Kubernetes中的概念，pod是一个容器环境下的“逻辑主机”，上

图说明Service Mesh与Service A处于同一个主机，Service Mesh与Service A通过127.0.0.1本地地址通信。

上图Service A请求Service B的流程如下：

1、service A请求Service mesh A。

2、Service mesh A请求Service mesh B。

3、Service mesh A请求Service mesh C。

下边是一个服务提供方采用service mesh的例子：

下边是一个服务消费方请求服务提供方都采用service mesh的例子：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

上图Service A请求Service B的流程如下：

1、service A请求Service mesh A。

2、Service mesh A请求Service mesh B。

3、Service mesh B 请求 Service B。

Service Mesh作为TCP/IP之上的一个抽象层，微服务只负责业务逻辑，Sidecar来负责微服务的通信、容错、限

流、降低等功能，Sidecar就是把微服务层面的功能抽取了出来，应用程序之间通过Service Mesh进行通信，如下

图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

Service Mesh翻译为“服务网格”，Willian Morgan（Linkerd的CEO）对Service Mesh的定义是：Service Mesh是

一个基础设施层，用于处理服务间通信。云原生应用有着复杂的服务拓扑，Service Mesh保证请求可以在这些拓扑

中可靠地传输。在实际应用当中，Service Mesh 通常是由一系列轻量级的网络代理组成的，它们与应用程序部署

在一起，但应用程序不需要知道它们的存在。

当该Sidecar在微服务中大量部署时，这些 sidecar 节点自然就形成了一个网格：

为什么要使用Service Mesh：

1、作为基础设施层，抽取微服务通信、治理功能。

2、老系统0代码侵入改造为微服务。

3、Service Mesh负责微服务通信、治理功能，这样微服务专注业务开发。

**8.2.2 Service Mesh发展**

业界将Spring Cloud称为第一代微服务技术，即侵入式微服务框架，将Service Mesh称为下一代的微服务技术。

Service Mesh起源于2016年，2017年越来越多的企业开始知道Service Mesh，业界将2018年称为Service Mesh元

年，预示着Service Mesh将从2018年开始发展。

Service Mesh发展至今已有如下开源项目：

Linkerd 、Envoy、 Istio 、Conduit、华为mesher、新浪Motan Mesh。

**8.2 Mesher介绍**

Mesher是华为公司自研的Service Mesh的具体实现，它是一个轻量的代理服务，以Sidecar的模式与微服务一起

运行，使用Mesher可以实现异构系统快速改为微服务，成为云原生应用。

Mesher 属于微服务引擎CSE的一个组件，Mesher在CSE中的结构图如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

Mesher是L7层协议代理，Mesher以Sidecar模式运行在应用所在的Pod内，与Pod共享网络与存储：

1. Pod中的应用使用Mesher作为http代理，可以自动发现其他服务。

2. Mesher会代替Pod中的应用向注册中心注册应用相关信息，以便让其他应用发现。

发起一次网络请求的过程中存在微服务消费者和提供者，场景如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

1、场景一：

仅消费者使用Mesher作为Sidecar。

提供者需要自己实现服务注册发现，或者使用Java或GO chassis开发框架，否则无法发现AccountService。

应用间的网络请求如下：

Store web -> Mesher -> Account service

2、场景二：

消费者提供者均使用Mesher作为Sidecar。

此场景无需再使用微服务开发框架。

应用间的网络请求如下：

Store web -> Mesher -> Mesher -> Account service

3、场景三：

仅提供者使用Mesher作为SideCar

消费者采用CSE微服务框架开发，通过mesher请求服务提供者。

应用间的网络请求如下：

Store web -> Mesher -> Account service

Mesher性能如何？

Mesher与Istio、Linkerd性能对比如下：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

与普通服务的性能对比：

**使用前**

1749

2.8ms

50%

5

**使用后**

1496

3.34ms

100%

5

TPS

Latency

CPU

Concurrency

为什么使用了Mesher比普通服务的性能低还这么流行？

相比微服务开发、运维的成本来说这点性能损耗是可以接受的！

**8.3 快速体验**

**8.3.1 功能介绍**

将weathermap的web前端weathermapweb工程接入CSE，weathermapweb前端基于Node.js运行，没有采用

CSE开发框架，如果要将weathermapweb接入CSE可以使用mesher来实现，通过mesher接入CSE后

weathermapweb就可以调用CSE中的微服务，如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

weathermapweb通过mesher向CSE服务注册中心进行注册，并通过mesher调用服务中心的微服务。

**8.3.2 环境准备**

下载mesher，进入应用开发--》微服务开发--》工具下载界面，下载mesher

本教程使用1.4.12版本，链接如下：

https://cse-bucket.obs.myhwclouds.com/Mesher/mesher-1.4.12-windows-amd64.zip

1. 解压Mesher安装包。

2. 配置服务监听和服务发布地址。

打开conf/chassis.yaml文件，修改地址。

listenAddress配置为：{本机实际IP地址}:30101

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

‐‐‐

cse:

service:

registry:

address: https://cse.cn‐north‐1.myhuaweicloud.com # uri of service center

scope: full #set full to be able to discover other app's service

watch: false # set if you want to watch instance change event

autoIPIndex: true # set to true if u want to resolve source IP to microservice

api:

version:

config:

client:

serverUri: https://cse.cn‐north‐1.myhuaweicloud.com #uri of config center

refreshMode: 1 # 1: only pull config. 2: watch config change events

refreshInterval: 30 # unit is second

autodiscovery: false

api:

version:

protocols:

http:

listenAddress: 192.168.1.104:30101

3、配置AK/SK。

打开conf/auth.yaml文件进行配置AK/SK

## Huawei Public Cloud ak/sk

cse:

credentials:

accessKey: 9AKR38RXJG48XMDDIUUA

secretKey: 2MEIV4VhNzRausA3tLOsN1s7nWw66SSp9hVXBJrl

project:

akskCustomCipher: default #used to decrypt sk when it is encrypted

4、打开conf目录下的microservice.yaml文件，修改用于描述微服务的信息，配置如下：

APPLICATION\_ID修改为weathermap。

service\_description.name修改为weathermapweb，该名字将作为Mesh注册到weathermap应用中的服务

名。

其他内容不作修改。

APPLICATION\_ID: weathermap

service\_description:

name: weathermapweb

version: 0.0.1

properties:

allowCrossApp: true

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

5、启动Mesher的start.bat。

6、进入前台“应用开发 > 微服务管理 > 服务目录”页面查看，可以看到名为weathermapweb微服务，其“所属应

用”为weathermap。

**8.3.3 weathermapweb接入CSE**

1、修改weathermap-1.0.0/weathermapweb/routes/目录下的weathermapweb.js文件。

修改端口号为Mesh监听端口，即30101。

将原来的IP:Port替换为fusionweather服务名，通过mesher请求微服务通过服务名进行请求。

其他内容不作修改。

示例如下：

router.get('/', function (req, res, next) {

var proxy\_host = process.env.HTTP\_PROXY\_HOST || '127.0.0.1';

var proxy\_port = process.env.HTTP\_PROXY\_PORT || '30101';

console.log(proxy\_host + ':' + proxy\_port);

console.log(req.query.city + ", " + req.query.type);

var opt = {

host: proxy\_host,

port: proxy\_port,

method: 'GET',

path: 'http://fusionweather/fusionweather/show?city=' + req.query.city,

headers: {}

};

2、在weathermapweb目录下启动startup.bat。

3、访问http://localhost:3000/

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

4、在“应用开发 > 微服务管理 > 服务治理”页面也可以看到weathermapweb与Fusionweather之间的调用关系。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**8.4 mesher研究**

**8.4.1 mesher作为消费方**

**8.4.1.1 方法介绍**

本章节测试mesher的场景一，即mesher作为消费者调用其它微服务（提供者），如下图：

mesher作为消费者调用其它微服务的流程如下：

1、consumer将请求发给mesher，请求地址为 “http://微服务名称/url”

2、mesher从服务注册中心查找provider微服务信息

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

3、mesher调用微服务provider

4、mesher将调用provider的结果返回给consumer

mesher作为消费者调用其它微服务的实现步骤如下：

1、在消费方设置http代理

通过代理发起Http请求,代理地址为mesher的地址（本机地址）

2、通过服务名消费服务

将原来的通过IP:Port请求方式改为 通过服务名请求。

**8.4.1.2 服务提供方**

服务提供方使用学成在线 的portalview服务。

项目介绍略。

**8.4.1.3 服务消费方**

本章节编写一个消费客户端用于通过mesher请求微服务，本客户端使用springBoot开发，不使用微服务框架，

创建服务消费方工程：

1、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>

<groupId>com.xuecheng</groupId>

<artifactId>xc‐meshertest01‐consumer</artifactId>

<version>1.0‐SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐starter‐web</artifactId>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>fastjson</artifactId>

<version>1.2.30</version>

</dependency>

</dependencies>

<dependencyManagement>

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring‐boot‐dependencies</artifactId>

<version>1.5.9.RELEASE</version>

<type>pom</type>

<scope>import</scope>

</dependency>

</dependencies>

</dependencyManagement>

</project>

2、application.yml

server:

port: ${PORT:40000}

context‐path: /

spring:

application:

name: xc‐meshertest01‐consumer

3、启动类

package com.xuecheng.meshertest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;

import org.springframework.http.client.SimpleClientHttpRequestFactory;

import org.springframework.web.client.RestTemplate;

import java.net.InetSocketAddress;

import java.net.Proxy;

import java.net.SocketAddress;

@SpringBootApplication

public class ConsumerMain {

public static void main(String[] args) {

SpringApplication.run(ConsumerMain.class, args);

}

}

4、controller

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

定义消费方法

package com.xuecheng.meshertest.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import org.springframework.web.client.RestTemplate;

@RestController

public class ConsumerController {

@GetMapping("/query")

public Object query(@RequestParam("id") String id) {

return null;

}

}

**8.4.1.4 mesher配置**

1、配置服务监听和服务发布地址。

配置方法同 快速体验。

2、配置AK/SK。

配置方法同 快速体验。

3、打开conf目录下的microservice.yaml文件，修改用于描述微服务的信息，配置如下：

APPLICATION\_ID: xc‐edu‐cloud1.0

service\_description:

name: xc‐meshertest01‐consumer

version: 0.0.1

properties:

allowCrossApp: true

**8.4.1.5 通过mesher消费**

目前服务消费方只能通过IP和端口的方式调用CSE中的微服务，其实是没有通过服务注册中心取微服务，消费过程

无法参与微服务治理，通过mesher进行消费则可以以服务名方式调用微服务，并且可以使用CSE微服务治理的功

能。

1、配置Http代理

本项目使用RestTemplate 发起Http请求，基于RestTemplate配置http代理如下：

在启动类中定义RestTemplate

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

@Bean

public RestTemplate restTemplate(SimpleClientHttpRequestFactory httpClientFactory) {

return new RestTemplate(httpClientFactory);

}

@Bean

public SimpleClientHttpRequestFactory httpClientFactory() {

SimpleClientHttpRequestFactory httpRequestFactory = new SimpleClientHttpRequestFactory();

SocketAddress address = new InetSocketAddress("127.0.0.1", 30101);

Proxy proxy = new Proxy(Proxy.Type.HTTP, address);

httpRequestFactory.setProxy(proxy);

return httpRequestFactory;

}

2、通过服务名消费服务

@RestController

public class ConsumerController {

@Autowired

RestTemplate restTemplate;

@GetMapping("/query")

public Object query(@RequestParam("id") String id) {

String url = "http://xc‐service‐portalview/portalview/course/get/"+id;

Object body = restTemplate.getForEntity(url,Object.class).getBody();

return body;

}

}

**8.4.1.6 测试**

通过postman发起get请求：http://localhost:40000/query?id=4028e581617f945f01617f9dabc40000

**8.4.2 mesher作为提供方**

**8.4.2.1 方法介绍**

本章节测试mesher的场景三，即mesher作为提供者被其它微服务调用，如下图：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

mesher作为提供者被其它微服务调用的流程如下：

注意：这里的consumer使用CSE框架开发。

1、consumer请求 “cse://微服务名称/url”

2、consumer从服务注册中心查找provider微服务信息，找到了mesher的地址

3、consumer调用mesher

4、mesher调用provider

5、mesher将调用provider的结果返回给consumer

mesher作为提供者被其它微服务调用的步骤如下：

1、consumer是CSE微服务，调用微服务请求“cse://微服务名称/url”地址

2、mesher监听本地服务提供方的地址，设置环境变量 SPECIFIC\_ADDR=127.0.0.1:本地服务端口

3、编写本地服务提供方的契约

**8.4.2.2 服务提供方**

服务提供方未使用微服务框架，需要使用mesher接入CSE。

服务提供方代码在上一章节的服务消费方工程中编写，创建ProviderController如下：

@RestController

@RequestMapping("/")

public class ProviderController {

@GetMapping(value="/hello",produces = "application/json;charset=utf‐8")

public String query(@RequestParam("id") String id) {

return "{\"id\":\""+id+"\"}";

}

}

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**8.4.2.3 服务消费方**

服务消费方使用学成在线的portalview工程，编写一个测试方法如下：

@RequestMapping("/portalview/consumer")

@RestSchema(schemaId = "consumer")

public class ConsumerController {

private final RestTemplate restTemplate = RestTemplateBuilder.create();

@GetMapping("/get/{id}")

public String consumer(@PathVariable("id") String id) {

String url = "cse://xc‐meshertest01‐consumer/hello?id="+id;

final Object result = restTemplate.exchange(url,

HttpMethod.GET,null,Object.class).getBody();

return result.toString();

}

}

**8.4.2.4 mesher配置**

1、mesher监听本地服务提供方的地址，在mesher目录下的“start.bat”中设置环境变量

SPECIFIC\_ADDR=127.0.0.1:40000

@echo off

SET SPECIFIC\_ADDR=127.0.0.1:40000

start /B /WAIT mesher.exe

pause

2、配置微服务的信息

打开conf目录下的microservice.yaml文件，修改用于描述微服务的信息，配置如下：

APPLICATION\_ID: xc‐edu‐cloud1.0

service\_description:

name: xc‐meshertest01‐consumer

version: 0.0.1

properties:

allowCrossApp: true

3、编写契约

mesher作为服务提供方需要为服务提供方编写契约，契约格式采用OpenApi规范，下边是/hello的契约：

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

swagger: '2.0'

info:

title: consumer

version: 1.0.0

basePath: /

produces:

‐ application/json

paths:

/hello:

get:

operationId: "hello"

produces:

‐ "application/json"

parameters:

‐ name: "id"

in: "query"

required: false

type: "string"

responses:

200:

description: "response of 200"

schema:

type: "object"

将契约内容以mesher代理的微服务名称命名，并保存在mesher目录下的 conf/微服务名/schema/下。

比如：本教程设置的mesher代理的微服务名称为 xc-meshertest01-consumer，则此契约文件保存在：

onf\xc-meshertest01-consumer\schema下，契约文件名为：xc-meshertest01-consumer.yaml

**8.4.2.5 通过mesher消费**

由于本例子消费方采用CSE框架实现微服务，所以可以直接采用CSE调用微服务方法，如下：

@RequestMapping("/portalview/consumer")

@RestSchema(schemaId = "consumer")

public class ConsumerController {

private final RestTemplate restTemplate = RestTemplateBuilder.create();

@GetMapping("/get/{id}")

public String consumer(@PathVariable("id") String id) {

String url = "cse://xc‐meshertest01‐consumer/hello?id="+id;

final Object result = restTemplate.exchange(url,

HttpMethod.GET,null,Object.class).getBody();

return result.toString();

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

}

}

**8.4.2.6 测试**

请求portalview的消费方法：

get:http://localhost:40200/portalview/consumer/get/fd

**8.5 学成在线使用mesher**

**8.5.1 需求分析**

本章节使用mesher接入学成在线令牌系统，此令牌系统使用Java开发，没有采用CSE开发，此系统基于Spring

Security开发考虑改造微服务框架的成本决定采用mesher接入认证系统，认证系统与其它微服务的关系如下：

1、微服务介绍

xc-edu-ucenter-auth：认证系统，采用serviceComb开发。

xc-edu-ucenter-security: 令牌系统，非微服务框架开发，使用mesher接入cse。

xc-edu-ucenter：用户中心服务，提供用户信息管理、查询功能。

2、认证流程

1）用户登录,输入账号和密码

2）请求xc-edu-ucenter-auth，校验账号和密码

3）xc-edu-ucenter-auth请求xc-edu-ucenter-security申请令牌

4）xc-edu-ucenter-security请求xc-edu-ucenter查询用户信息。

3、mesher接入分析

根据上边的流程涉及到mesher交互的业务如下：

1）xc-edu-ucenter-auth请求xc-edu-ucenter-security申请令牌

微服务xc-edu-ucenter-auth是通过mesher请求xc-edu-ucenter-security

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

2）xc-edu-ucenter-security请求xc-edu-ucenter查询用户信息。

xc-edu-ucenter-security是通过mesher请求微服务xc-edu-ucenter。

4、目标

通过本次xc-edu-ucenter-security接入CSE，可以学习到mesher作为服务消费方及服务提供方的开发方法。

**8.5.2 导入工程**

解压“资料”-》“学成在线”-》xc-edu-ucenter.zip，将解压得到的xc-edu-ucenter-auth、xc-edu-ucenter-security、

xc-edu-ucenter三个工程导入IDEA，导入成功如下图：

**8.5.3 mesher配置**

本次测试在本地运行mesher，将服务向公网云平台服务中心注册。

**8.5.3.1 配置微服务信息**

xc-edu-ucenter-security通过mesher向服务中心注册为微服务。

修改mesher目录下的"conf/microservice.yaml"文件

APPLICATION\_ID: xc‐edu‐cloud1.0

service\_description:

name: xc‐service‐security

version: 1.0.0

properties:

allowCrossApp: true

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**8.5.3.2 配置认证信息**

配置ak/sk，作业向公网服务中心注册的凭证。

修改mesher目录下的"conf/auth.yaml"文件。注意：ak/sk需与自己账号匹配。

## Huawei Public Cloud ak/sk

cse:

credentials:

accessKey: 9AKR38RXJG48XMDDIUUA

secretKey: 2MEIV4VhNzRausA3tLOsN1s7nWw66SSp9hVXBJrl

project:

akskCustomCipher: default #used to decrypt sk when it is encrypted

**8.5.3.3 对接服务中心和配置中心**

修改mesher目录下的"conf/chassis.yaml"文件

‐‐‐

cse:

protocols:

http:

listenAddress: 192.168.1.104:30101

service:

registry:

# address: http://127.0.0.1:30100 # uri of service center

address: https://cse.cn‐north‐1.myhuaweicloud.com # uri of service center

scope: full #set full to be able to discover other app's service

watch: false # set if you want to watch instance change event

autoIPIndex: false # set to true if u want to resolve source IP to microservice

config:

client:

serverUri: https://cse.cn‐north‐1.myhuaweicloud.com #uri of config center

refreshMode: 1 # 1: only pull config.

refreshInterval: 30 # unit is second

**8.5.4 查询用户信息**

**8.5.4.1 方法介绍**

本章节实现xc-edu-ucenter-security是通过mesher请求微服务xc-edu-ucenter查询用户信息，即mesher作为服

务消费方的开发方法。

通过mesher调用微服务的方法如下：

1、设置http代理

xc-edu-ucenter-security发起的请求全部通过mesher，代理地址为mesher的地址（本机地址）

2、通过服务名请求

将原来的通过IP:Port请求方式改为 通过服务名请求。

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

**8.5.4.2 设置http代理**

xc-edu-ucenter-security是通过RestTemplate发起http请求，这里向RestTemplate对象设置http代理地址，方法

如下：

在SpringBoot启动类中定义RestTemplate

@Bean

public RestTemplate restTemplate(SimpleClientHttpRequestFactory httpClientFactory) {

return new RestTemplate(httpClientFactory);

}

@Bean

public SimpleClientHttpRequestFactory httpClientFactory() {

SimpleClientHttpRequestFactory httpRequestFactory = new

SimpleClientHttpRequestFactory();

SocketAddress address = new InetSocketAddress("127.0.0.1", 30101);

Proxy proxy = new Proxy(Proxy.Type.HTTP, address);

httpRequestFactory.setProxy(proxy);

return httpRequestFactory;

}

**8.5.4.3 通过服务名请求**

修改原来的UserDetailsServiceImpl类，使用RestTemplate通过mesher代理请求xc-service-ucenter，代码如下：

@Service

public class UserDetailsServiceImpl implements UserDetailsService {

@Autowired

RestTemplate restTemplate;

//

//

@Autowired

UserService userService;

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

if (StringUtils.isEmpty(username)) {

return null;

}

Map<String,String> map = null;

String path = "http://xc‐service‐ucenter/ucenter/getuserext?username="+username;

try {

Object result = restTemplate.exchange(path, HttpMethod.GET,

null, Object.class).getBody();

map = (Map<String, String>) result;

} catch (Exception e) {

e.printStackTrace();

return null;

}

if(map == null){

return null;

}

XcUserExt userext = new XcUserExt();

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

/\*\*

{

"id": "49",

"username": "itcast",

"password": "$2a$10$TJ4TmCdK.X4wv/tCqHW14.w70U3CC33CeVncD3SLmyMXMknstqKRe",

"salt": null,

"name": "test02",

"utype": "101002",

"birthday": null,

"userpic": null,

"sex": "1",

"email": "",

"phone": "12345",

"status": "1",

"createTime": "2018‐03‐07T08:27:47.000Z",

"updateTime": null,

\*/

userext.setId(map.get("id"));

userext.setName(map.get("name"));

userext.setUsername(map.get("username"));

userext.setPassword(map.get("password"));

XcUserExt userext = userService.getUserExt(username);

if(userext == null){

//

return null;

}

//取出正确密码（hash值）

String password = userext.getPassword();

//这里暂时使用静态密码

//

String password ="123";

//用户权限，这里暂时使用静态数据，最终会从数据库读取

//从数据库获取权限

List<XcMenu> permissions = userext.getPermissions();

if(permissions == null){

permissions = new ArrayList<>();

}

List<String> user\_permission = new ArrayList<>();

permissions.forEach(item‐> user\_permission.add(item.getCode()));

user\_permission.add("course\_get\_baseinfo");

user\_permission.add("course\_find\_pic");

String user\_permission\_string = StringUtils.join(user\_permission.toArray(), ",");

UserJwt userDetails = new UserJwt(username,

password,

//

//

AuthorityUtils.commaSeparatedStringToAuthorityList(user\_permission\_string));

userDetails.setId(userext.getId());

userDetails.setUtype(userext.getUtype());//用户类型

userDetails.setCompanyId(userext.getCompanyId());//所属企业

userDetails.setName(userext.getName());//用户名称

userDetails.setUserpic(userext.getUserpic());//用户头像

/\* UserDetails userDetails = new

org.springframework.security.core.userdetails.User(username,

password,

AuthorityUtils.commaSeparatedStringToAuthorityList(""));\*/

//

AuthorityUtils.createAuthorityList("course\_get\_baseinfo","course\_get\_list"));

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

return userDetails;

}

}

**8.5.4.4 测试**

直接请求xc-edu-ucenter-security申请令牌的Url进行测试，观察通过mesher请求是否正常。

posthttp://localhost:40401/oauth/token

参数如下：

grant\_type：密码模式授权填写password

username：账号

password：密码

并且此链接需要使用 http Basic认证。

**8.5.5 申请令牌**

**8.5.5.1 方法介绍**

本章节实现微服务xc-edu-ucenter-auth是通过mesher请求xc-edu-ucenter-security申请令牌，即mesher作为服

务提供方的开发方法。

mesher作为提供者被其它微服务调用的方法如下：

1、xc-edu-ucenter-auth调用微服务请求“cse://xc-edu-security/url”地址

2、mesher监听本地服务提供方的地址，设置环境变量 SPECIFIC\_ADDR=127.0.0.1:40401

3、编写契约

**8.5.5.2 服务消费方**

xc-edu-ucenter-auth调用微服务的方法修改为请求“cse://xc-edu-ucenter-security/url”地址

修改：com.xuecheng.auth.service.AuthService的public AuthToken applyToken(String username,String

password,String clientId,String clientSecret)方法：

......

String path = "cse://xc‐service‐security/oauth/token";

Map<String,String> map = null;

Object obj = null;

RestTemplate restTemplate = RestTemplateBuilder.create();

try {

Object result = restTemplate.exchange(path, HttpMethod.POST,

new HttpEntity<MultiValueMap<String, String>>(formData, headers),

Object.class).getBody();

System.out.println(result);

map = (Map<String, String>) result;

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

} catch (Exception e) {

LOGGER.error("request oauth\_token\_password error: {}",e.getMessage());

e.printStackTrace();

ExceptionCast.cast(AuthCode.AUTH\_LOGIN\_ERROR);

}

......

**8.5.5.3 mesher配置**

1、mesher监听本地服务提供方的地址，在mesher目录下的“start.bat”中设置环境变量

SPECIFIC\_ADDR=127.0.0.1:40401

@echo off

SET SPECIFIC\_ADDR=127.0.0.1:40401

start /B /WAIT mesher.exe

pause

2、编写契约

mesher作为服务提供方需要为服务提供方编写契约，契约格式采用OpenApi规范，下边是/oauth/token的契约：

swagger: '2.0'

info:

title: hello

version: 1.0.0

basePath: /

produces:

‐ application/json

paths:

/oauth/token:

post:

operationId: "postAccessToken"

produces:

‐ "application/json"

parameters:

‐ name: "grant\_type"

in: "formData"

required: false

type: "string"

‐ name: "username"

in: "formData"

required: false

type: "string"

‐ name: "password"

in: "formData"

required: false

type: "string"

responses:

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090

200:

description: "response of 200"

schema:

type: "object"

**8.5.5.4 测试**

请求xc-edu-ucenter-auth的认证方法：Post http://localhost:40400/auth/userlogin

参数：

username：账号

password：密码

北京市昌平区建材城西路金燕龙办公楼一层 电话：400-618-9090