

DAY16 微服务应用开发之持续交付

1 打卡任务

作业：

基于模板创建云上工程，并访问应用接口；更新代码后，执行流水线，访问新版本应用接口。

打卡：

访问新版应用接口，并截图。

2 准备工作

- 1、切换区域为“华东-上海二”。
- 2、确认选择的ServiceStage版本为“基础版”（可免费使用10个实例，**超过10个实例或选择的是专业版则会按小时进行计费**）。
- 3、已拥有DevCloud账号。
- 4、已拥有可运行的CCE集群、某一节点绑定弹性IP。

3 创建云上工程

本步骤可查阅ServiceStage用户指南，https://support.huaweicloud.com/usermanual-servicestage/servicestage_user_0011.html。

- 1、登录ServiceStage，选择 持续交付 > 工程。
- 2、单击“创建云上工程”，单击“基于模板创建”下的“创建云上工程”。
- 3、选择语言及框架
本次实践选择语言“Java 8”，选择框架“CSE-Java（SpringMVC）”。
- 4、选择部署系统
选择“云容器引擎CCE”
- 5、配置工程参数。

a. 输入项目名称。

b. 配置一个源码仓库

本次实践配置DevCloud“源码仓库”参数。

i. 在“代码源来源”中选择DevCloud，单击“绑定帐户”，输入密码(该密码是DevCloud仓库的密码)，获取DevCloud源码仓库授权。

ii. 选择默认命名空间Demo，并输入源码仓库名称：hellocloud。

c. 选择仓库组织（如果没有则通过“创建仓库组织”进行创建，创建成功后返回刷新进行选择即可）。

d. 选择部署集群：云容器引擎CCE。

e. 高级设置

i. 设置访问方式

勾选“把应用发布到主机网络上，用于website、LB等开放页面，接口给外部访问的场景。”

ii. 添加服务

1)设置服务名称

输入“hellocloud”

2)设置访问方式

“访问方式”选择“公网访问”

“访问类型”选择“弹性IP”

3)设置端口映射

容器端口输入“8080”

访问端口“30069”

* 项目名称

* 代码来源

DevCloud

GitHub

Gitee

Bitbucket

GitLab

已绑定账户:

命名空间 仓库名称

* 仓库组织 [创建仓库组织](#)

* 部署集群 [创建集群](#)

* 集群命名空间 [创建命名空间](#)

* 实例个数

高级设置 (资源配额, 设置访问方式) ^

资源配额

设置访问方式 ☒ 把应用发布到主机网络上,用于website、LB等开放页面,接口给外部访问的场景。

[添加服务](#)

内部域名访问地址	访问方式	容器端口	访问端口	协议	操作
hellocloud.default.svc.cluster.local:30069	公网访问>弹性IP:49.4.4.250	8080	30069	TCP	删除

- 6、单击“部署”，返回“持续交付 > 工程”查看已创建工程清单中查看刚创建的工程，“状态”栏显示正常图标，表示工程创建成功。

您可以创建基于微服务SDK的微服务工程并下载，快速进入业务代码开发阶段。 [如何创建微服务？](#)

云上工程 本地工程

[+ 创建云上工程](#)

名称	状态	创建时间	描述	操作
project-w6x7	正常	2019/03/02 17:11:08 GMT+08:00		源码地址 跳转到流水线 删除工程

4 访问云上工程应用

- 1、选择 持续交付 > 工程。
- 2、选择 [步骤3](#)创建的云上工程项目，单击“跳转到流水线”。
- 3、等待流水线运行完成。

微服务云应用平台

流水线 > 流水线详情

流水线名称: project-w6x7

状态: 成功

创建时间: 2019/03/27 09:44:49 GMT+08:00

最近一次执行时间: 2019/03/27 09:44:49 GMT+08:00

1 基本信息

流水线名称: project-w6x7

描述:

2 Build

build 构建系统 Assembling

成功 详情 2分钟前

3 Deploy

deploy 部署系统 CFE

成功 详情 刚刚

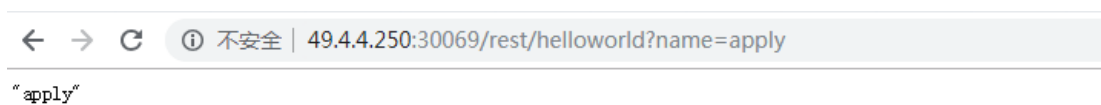
- 4、选择 应用管理 > 应用列表，找到对应的应用，查看状态为“运行中”，如下所示：

应用名称	类型	状态	部署系统	外部访问地址	实例总数	创建时间	操作
project-w6x7	容器应用	运行中	云容器引擎CCE	49.4.4.250:30069	1	2019/03/02 17:42:49 GMT+08:00	编辑 升级 更多

通过chrome浏览器访问应用接口（将下面的IP地址替换为上图中对应的外部访问

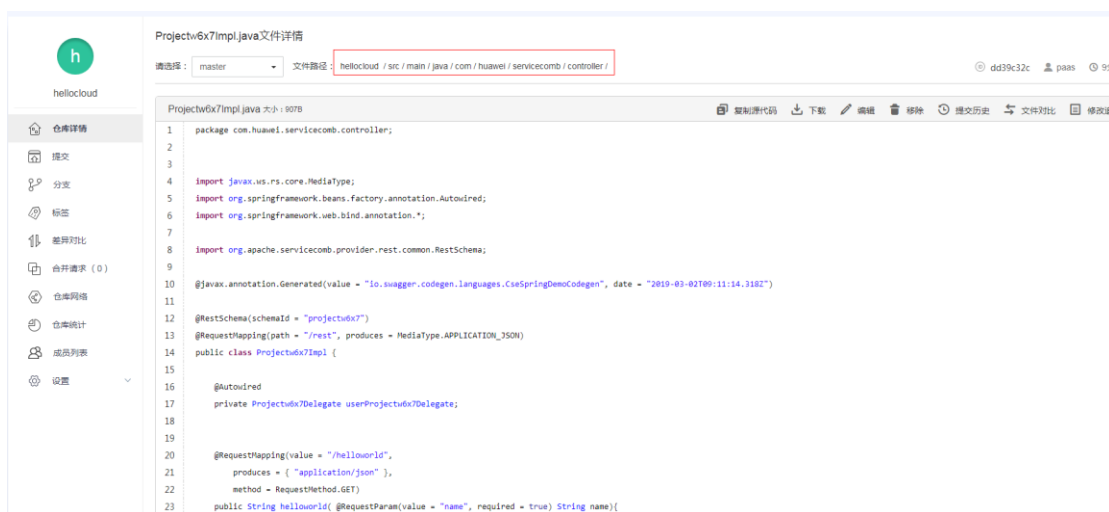
地址）：<http://49.4.4.250:30069/rest/helloworld?name=apply>

返回如下结果：



5 通过流水线升级应用

- 1、登录ServiceStage，选择 持续交付 > 工程。
- 2、选择步骤3创建的云上工程项目，单击“源码地址”，跳转到DevCloud源码仓库，“服务”-->“代码托管”，找到刚刚的代码仓库 Demo / hellocloud。
- 3、修改应用接口返回内容，



在hellocloud/src/main/java/com/huawei/servicecomb/controller/Projectw6x7Impl.java文件（注：名称Projectw6x7Impl会根据您输入的工程名有所不同）中将：

```
public class Projectw6x7Impl {

    @Autowired
    private Projectw6x7Delegate userProjectw6x7Delegate;

    @RequestMapping(value = "/helloworld",
        produces = { "application/json" },
        method = RequestMethod.GET)
    public String helloworld( @RequestParam(value = "name", required = true) String name){

        return userProjectw6x7Delegate.helloworld(name);
    }

}
```

中的 userProjectw6x7Delegate.helloworld(name) 在线编辑修改为：

userProjectw6x7Delegate.helloworld("HI: " + name);

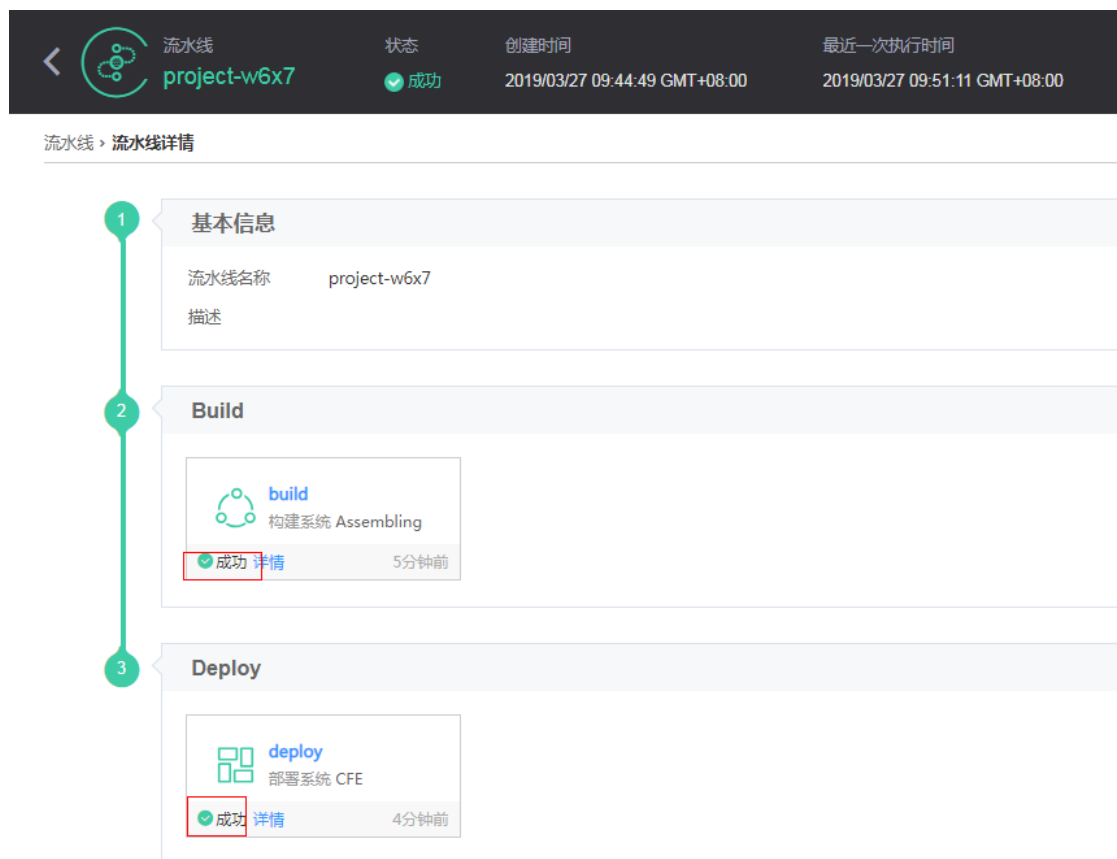
然后保存提交。

```

11
12 @RestSchema(schemaId = "projectw6x7")
13 @RequestMapping(path = "/rest", produces = MediaType.APPLICATION_JSON)
14 public class Projectw6x7Impl {
15
16     @Autowired
17     private Projectw6x7Delegate userProjectw6x7Delegate;
18
19
20     @RequestMapping(value = "/helloworld",
21         produces = { "application/json" },
22         method = RequestMethod.GET)
23     public String helloworld( @RequestParam(value = "name", required = true) String name){
24
25         return userProjectw6x7Delegate.helloworld("HI: " + name);
26     }
27
28 }
29

```

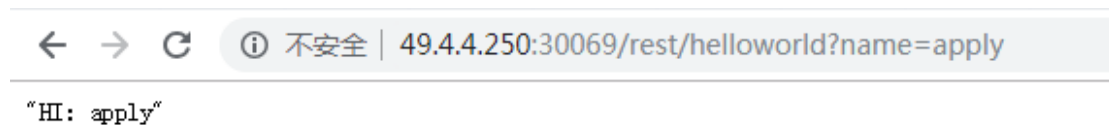
- 4、修改后在ServiceStage界面选择 持续交付 > 发布，选择上述流水线，单击“启动”



- 5、 等待流水线运行完成。
- 6、 点击 应用管理 > 应用列表，找到对应的应用，通过chrome浏览器访问应用接口（将下面的IP地址替换为应用实际对应的外部访问地址）：

<http://49.4.4.250:30069/rest/helloworld?name=apply>

返回如下结果:



6 打卡截图

- 1、通过浏览器访问应用接口: <http://49.4.4.250:30069/rest/helloworld?name=apply>

