

# 结业实践（第一阶段）

本章为微服务课程最终的结业实践，采用目前应用场景较多的 SpringCloud 微服务模式，来搭建一个电商网站的后台管理系统。在开始实践前，请先接受作业，获取实践项目源码。

## 接受实践作业

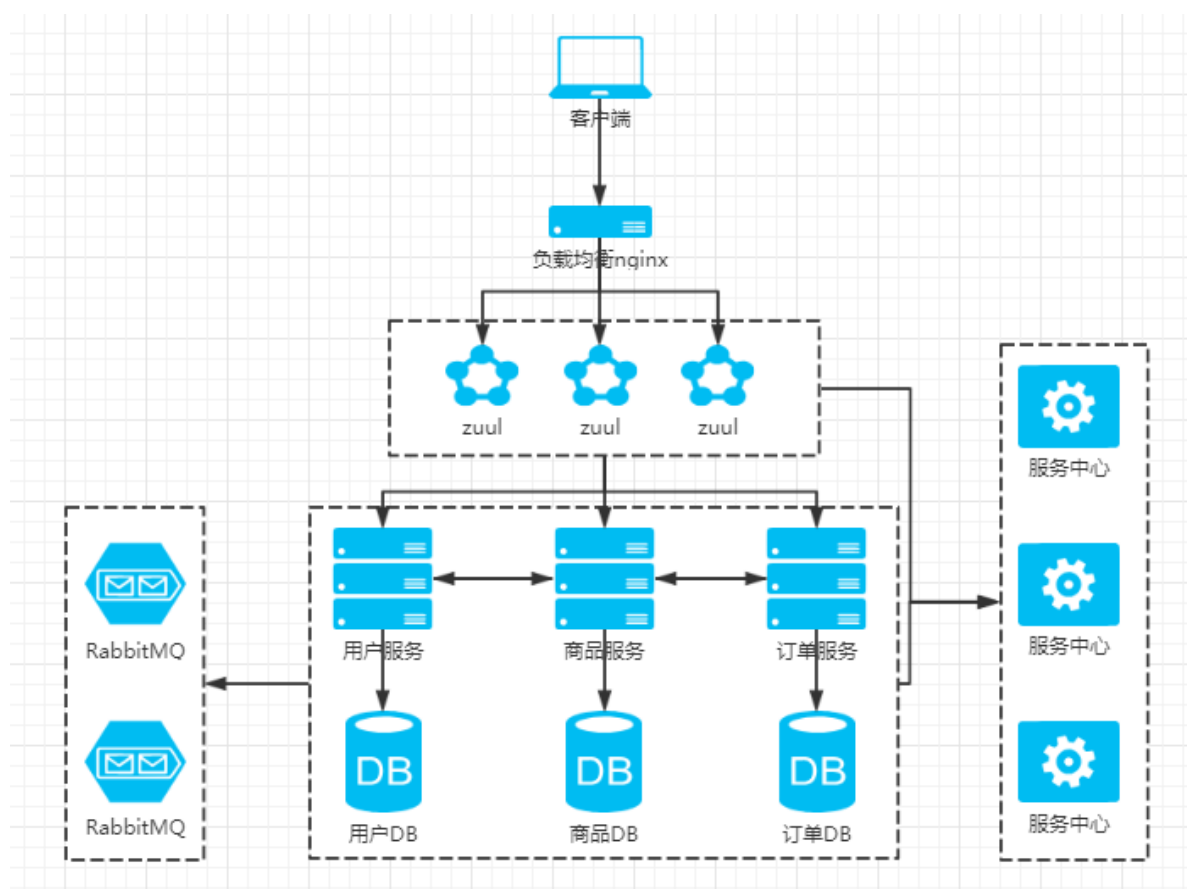
[点此接受实践作业](#)

## 实践概述

### 1. 实践项目简介

电商网站的后台管理系统提供了单点与统一认证的服务，除了本系统自用外，如企业其他子系统也可以接入到这个认证系统中，有用户认证、用户角色、用户管理、等诸多功能，还有仓库管理服务，可对电商系统的仓库进行更新与维护，如价格、库存等，还有订单服务，自动下单、手动下单、订单维护查看等，通过这个系统可满足小范围的电商应用，也可在此基础上进行二次开发。

## 2. 系统架构图



## 3. 技术选型

Springboot、jpa、mysql、rabbitmq、nginx、eureka、hystrix、feign、zuul、Docker

软件代码管理：[华为云 DevCloud](#)

软件编译工具：[华为云 DevCloud](#)

软件部署工具：[华为云 DevCloud](#)、[华为云 ServiceStage](#)

## 4. 实践准备

- [接受实践作业](#)，获取实践项目源码
- 开通云容器引擎 CCE，[进入活动页](#)，领取免费虚拟机集群

# 数据库&RabbitMQ 应用创建

## 1. 创建数据库应用

集群创建成功后，需要先创建一些本实践中部分微服务需要依赖的数据库

### 操作方法：

右键新标签页打开 [ServiceStage 控制台](#)

访问：[ServiceStage 控制台](#) > 应用上线 > 应用管理 > 创建应用

需要依次创建 user、store、bill 服务需要的数据库：mysql-user，mysql-store，mysql-bill

以 mysql-user 为例，点击创建应用后，按照下述流程创建：

应用类型：容器应用

应用名称：mysql-user

实例数量：1 个

选择镜像：Docker 官方镜像>mysql 镜像



镜像版本：5.6

高级配置：环境变量>添加环境变量：变量名称- `MYSQL_ROOT_PASSWORD`，引用值-自定义

高级设置（生命周期，健康检查，环境变量，数据存储，容器日志） ^

生命周期：设置容器启动和运行时需要命令 ▼

健康检查：用于判断容器和内部业务是否正常运行 ▼

环境变量：设置容器运行环境中的一些变量，便于部署后灵活变更容器配置 ^

环境变量设置 添加环境变量

| 类型   | 变量名称             | 变量/变量引用 | 操作 |
|------|------------------|---------|----|
| 手动添加 | MYSQL_ROOT_PASSW | 123456  | 删除 |

下一步

添加服务，设置 mysql 访问方式：

## 添加服务

\* 服务名称

mysql-user

访问方式

☐ 集群内访问 ☐ VPC内网访问 ☒ 公网访问

提供支持TCP/UDP协议的Internet访问入口，包含弹性IP方式。

\* 访问类型

弹性IP

\* 端口映射

| 协议  | 容器端口 | 访问端口       |
|-----|------|------------|
| TCP | 3306 | 指定端口 30036 |

点击创建应用，并等待应用创建完成。

按照上述步骤，重复创建 mysql-store 和 mysql-bill 应用

mysql-store 的端口号为：3306、30037，mysql-bill 的端口号为 3306、30038（访问端口可自定义，注意不要重复）

全部创建成功后，可在应用管理界面查看：

  
应用上线

应用管理 **NEW!**

应用编排  
任务管理  
配置中心

您可以创建自己的应用，并且可以根据需要对应用进行部署、配置、监控、扩容、升级、卸载、服务发

+ 创建应用

删除应用 移至应用组 集群：cce-d... 已选择命名空间

| <input type="checkbox"/> | 应用名称        | 类型    | 状态   | 外部访问地址             | 实例个数（正常... |
|--------------------------|-------------|-------|------|--------------------|------------|
| <input type="checkbox"/> | mysql-bill  | 容器... | 运... | 117.78.27.44:30038 | 1/1        |
| <input type="checkbox"/> | mysql-store | 容器... | 运... | 117.78.27.44:30037 | 1/1        |
| <input type="checkbox"/> | mysql-user  | 容器... | 运... | 117.78.27.44:30036 | 1/1        |

## 2. 初始化数据库

三个 mysql 应用创建成功后，使用 mysql 客户端链接数据库，进行数据初始化

用户：root

密码：123456

端口：30036（教程中的 mysql-user 端口号为 30036，mysql-store 端口号为 30037，mysql-bill 端口号为 30038）

链接成功之后，执行以下命令创建 gift-store 的数据库实例（以 mysql-user 为例）：

```
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;  
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;  
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;  
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;  
CREATE DATABASE /*!32312 IF NOT EXISTS*/`gift-user` /*!40100 DEFAULT CHARACTER SET utf8 */;
```

依次为 mysql-store、mysql-bill 创建实例，对应上述指令中的 gift-user 替换为 gift-store、gift-bill

## 3. 创建 RabbitMQ 应用

返回 [ServiceStage 控制台](#)，按上述应用创建步骤，继续创建 RabbitMQ 应用

应用类型：容器应用

应用名称：rabbitmq

实例数量：1 个

选择镜像：Docker 官方镜像>rabbitmq 镜像



镜像版本：3.6-management

高级配置：环境变量>添加环境变量：变量名称- RABBITMQ\_DEFAULT\_USER ,

RABBITMQ\_DEFAULT\_PASS , 引用值-自定义



下一步

添加服务，设置 rabbitmq 访问方式：

## 添加服务

\* 服务名称

rabbitmq

访问方式

☒ 集群内访问

☐ VPC内网访问

☐ 公网访问

提供支持TCP/UDP协议的被集群内其他服务访问的入口。

\* 端口映射

协议

容器端口

访问端口

TCP

5672

5672

点击创建应用，并等待应用创建完成。

## gift-server 服务构建部署

接受作业进入 DevCloud 实践项目中后，点击左侧菜单：构建&发布 > 编译，本实践提供已创建好的编译任务。

微服务驾云攻略-搭... ▼ ♥

仪表盘

工作 ▼

代码 ▼

构建&发布 ▲

流水线

编译

| 名称          | 创建者         | 最近构建时间           | 健康度   | 构建状态 | 操作 |
|-------------|-------------|------------------|-------|------|----|
| gift-server | grandmaster | 2018-09-18 16:59 | ★★★★★ | 成功   | ▶  |
| gift-user   | grandmaster | 2018-09-18 16:59 | ★★★★★ | 成功   | ▶  |
| gift-store  | grandmaster | 2018-09-18 16:59 | ★★★★★ | 成功   | ▶  |
| gift-bill   | grandmaster | 2018-09-18 16:59 | ★★★★★ | 成功   | ▶  |
| gift-ui     | grandmaster | 2018-09-18 16:59 | ★★★☆☆ | 成功   | ▶  |
| gift-zuul   | grandmaster | 2018-09-18 16:59 | ★★★★☆ | 成功   | ▶  |

### 1. 修改编译任务

在执行之前请先完成任务修改：

点击任务 gift-server 左侧的 “✎” 图标，将 **组织** 改为当前账号下的镜像组织名称。

镜像组织创建流程：

右击新标签页打开：[容器镜像服务](#)

点击 立即使用 ，访问：总览 > 创建组织，输入 组织名称 ，点击确定。

返回 DevCloud，修改镜像组织名称后点击保存。

### 编译构建任务

**gift-server**  
Gift-MicroServi... master

Maven构建

制作镜像并推送到SWR仓库

+

每个任务都有一个目录作为工作空间：  
1. 工作空间会挂载到每个镜像，镜像内把内容放到工作空间，才不会丢失  
2. 需要使用路径的时候，都是相对于工作空间的路径

**制作镜像并推送到SWR仓库**  
通过Dockerfile制作镜像并推送到SWR仓库

环境选择： 推荐镜像

镜像类型： docker docker18.03

\* 仓库地址 100.125.0.198:20202

\* 组织 microservice

\* 镜像名字 gift-server

\* 镜像标签 1.0.0

Dockerfile路径 ./gift-server/Dockerfile

## 2. 执行编译任务

任务编辑完成后，点击 **开始构建** ，等待构建任务执行完毕。

构建任务执行成功后，会将构建成功的镜像文件 “gift-sever” 上传至华为云 SWR，可访问：

[SWR 控制台](#) > 我的镜像 ，验证镜像是否上传成功。



容器镜像服务

总览

我的镜像

镜像构建

镜像资源

组织管理

体验馆

自有镜像 他人共享

删除

| <input type="checkbox"/> | 镜像名称        | 所属组织         |
|--------------------------|-------------|--------------|
| <input type="checkbox"/> | gift-zuul   | microservice |
| <input type="checkbox"/> | gift-bill   | microservice |
| <input type="checkbox"/> | gift-store  | microservice |
| <input type="checkbox"/> | gift-user   | microservice |
| <input type="checkbox"/> | gift-server | microservice |

### 3. 执行部署任务

访问项目左侧菜单：**构建&发布>部署**，点击新建任务，选择非模板任务。

按如下信息，配置任务详情：

基本信息：

1 基本信息

\* 名称：

gift-server

描述 ?

部署设置：

2 部署设置

\* 部署类型：

Shell部署(Linux)

容器部署

Ansible部署

\* 所在集群：

cce-demo1522483688767

\* 命名空间：

default

创建命名空间

其中所在集群可选择账号下已开通的集群资源（可在活动页面内领取免费资源）

应用配置：

3 应用配置

\* 实例： ?  1  个

☐ 启用java应用监控(可提供java应用调用链，拓扑等监控能力。仅java镜像生效)，请确认集群已安装ICAgent组件

\* 容器： 

容器 X  
container001

+ 添加容器

镜像： 

microservice/gift-server:1.0.0

[上传镜像](#) [构建镜像](#)

版本： 

1.0.0

高级配置

其中镜像 点击添加镜像，选择上一步构建成功的镜像文件：gift-sever

点开高级配置，内存和 CPU 按照集群实际使用情况填写

高级配置 ^

内存： ? 

1

 G

CPU： ? 

0.5

 核

环境变量：

| 变量名称                 | 变量值                  | 操作                       |
|----------------------|----------------------|--------------------------|
| <input type="text"/> | <input type="text"/> | <div><div></div> +</div> |

存储： ?

| 主机路径                            | 容器路径                            | 操作                       |
|---------------------------------|---------------------------------|--------------------------|
| <div>e.g. /var/lib/docker</div> | <div>e.g. /home/docu/file</div> | <div><div></div> +</div> |

设置访问方式：

#### 4 设置访问方式

发布为服务：☒ 是 ☐ 否

服务参数：

| 访问名称        | 访问类型   | 容器端口 | 访问端口      | 协议 | 操作   |
|-------------|--------|------|-----------|----|------|
| gift-server | 集群虚拟IP | 8000 | 指定端口 8000 | TC | 🗑️ + |

全部内容填写完毕后，点击确认，并执行部署任务，等待部署成功。

#### 4. 获取 gift-server 服务的访问地址

我们需要获取 gift-server 服务的集群内访问地址，以便修改其他服务的配置文件。

gift-server 服务部署成功后，进入 [ServiceStage 控制台](#)

访问应用上线>应用管理，可查看已在运行状态的 gift-server 服务：



应用上线

应用管理 NEW!

应用编排

任务管理

配置中心

您可以创建自己的应用，并且可以根据需要对应用进行部署、配置、监控、扩容、

[+ 创建应用](#)

🗑️ 删除应用

🔄 移至应用组

集群：cce-d... ▼

| <input type="checkbox"/> | 应用名称        | 类型    | 状态     | 外部访问地址             |
|--------------------------|-------------|-------|--------|--------------------|
| <input type="checkbox"/> | gift-server | 容器... | ➡ 运... | --                 |
| <input type="checkbox"/> | rabbitmq    | 容器... | ➡ 运... | --                 |
| <input type="checkbox"/> | mysql-bill  | 容器... | ➡ 运... | 117.78.27.44:30038 |
| <input type="checkbox"/> | mysql-store | 容器... | ➡ 运... | 117.78.27.44:30037 |
| <input type="checkbox"/> | mysql-user  | 容器... | ➡ 运... | 117.78.27.44:30036 |

点击任务名称>访问方式>添加服务，复制并保存 内部域名访问地址

- 概览
- 实例列表
- 访问方式
- 更新升级
- 伸缩
- 事件
- 运行日志
- 阈值告警

TCP/UDP路由配置 支持TCP/UDP四层负载均衡

+ 添加服务

| 内部域名访问地址                                   | 访问地址                                                                                                  | 访问方式  |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------|-------|
| gift-server.default.svc.cluster.local:8000 |  10.247.177.174:8000 | 集群内访问 |

至此，第一阶段的实践课程已结束，实践的后半部分所使用的集群规格需要 4C16G 以上，请在原有低规格集群基础上完成扩容或新购 4C16G 集群进入第二阶段结业实践。