

应用编排服务

最佳实践)

文档版本 01
发布日期 2023-05-05



版权所有 © 华为技术有限公司 2023。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目录

1 应用编排	1
1.1 应用编排+容器技术实现快速部署 Magento 电商网站.....	1

1 应用编排

1.1 应用编排+容器技术实现快速部署 Magento 电商网站

通过**Magento电商应用**模板，您可以快速部署一个Magento电子商务网站容器应用。Magento电子商务网站包含一个前端组件，和一个MySQL数据库。本章节将指导您修改“Magento电商应用”公共模板，基于应用编排和容器技术实现快速部署Magento电商网站，修改后的模板支持在创建堆栈时对容器使用的CPU和内存进行申请与限制。

在此演练中，您将完成以下步骤：

1. **了解基本概念**：在操作之前，建议您先了解本次演练用到的一些基本概念。
2. **修改公共模板**：修改“Magento电商应用”公共模板，增加magento前台应用的CPU和内存的申请与限制参数。
3. **创建Magento电商网站应用堆栈**：通过修改后的模板创建容器应用。最终将创建的应用部署到集群中。
4. **清除资源**：为了防止您为不需要的服务付费，建议您删除堆栈，清理资源。

前提条件

请确保已有华为云帐号，且在部署容器前，需确保至少已包含一个可用容器集群和一个2C4G的节点。

您可登录**CCE控制台**，单击“资源管理 > 节点管理”，查看到节点状态为“可用”，弹性IP已绑定。获取该弹性IP值。

图 1-1 查看是否存在可用节点



了解基本概念

修改模板之前，请了解一下**Magento**电商应用模板的基本构成。

```
# 应用模板所基于的类型定义版本
tosca_definitions_version: huaweicloud_tosca_version_1_0
# 输入参数定义
inputs:
  app-name:
    default: magento
    description: '应用名称##Application name'
    label: magento
  magento-EIP:
    description: 'magento服务对外暴露访问地址##External access address of the Magento service'
    label: magento
  magento-EPORT:
    default: 32080
    description: 'magento服务对外监听端口##External listening port of the Magento service'
    label: magento
    type: integer
  ... ..
# 映射表定义
mappings:
  region_map: #定义不同region镜像和规格的映射
  cn-east-2:
    magento-image: '10.125.17.64:20202/aos-samples/magento:1.9.1.0'
    mysql-image: '10.125.17.64:20202/aos-samples/mysql:latest'
  ... ..
#应用拓扑定义
node_templates:
  magento: #元素名称
    metadata:
      Designer:
        id: e66e332a-3466-4638-9896-f7d2e93a1ae3
    properties: #元素属性
      k8sManifest:
        apiVersion: apps/v1
        kind: Deployment
      metadata:
        labels:
          app:
            get_input: app-name
        name:
          get_input: app-name
      ... ..
    requirements: #元素依赖
      - dependency:
          node: mysql-service
      - dependency:
          node: mysql-conf
      - dependency:
          node: magento-config
    type: HuaweiCloud.CCE.Deployment #元素类型
  ... ..
# 输出参数定义
outputs:
  ingress-admin_password:
    description: Password of super user.
    value: magentorocks1
  magento-addr:
    description: Access URL for magento service.
    value:
      concat:
        - 'http://'
        - get_input: magento-EIP
        - ':'
        - get_input: magento-EPORT
  magento-admin_username:
```

```
description: Super user name.  
value: admin
```

模板的各属性解释如下：

- **inputs**

可选项，用于定义基于模板创建堆栈时的可变部分。一个模板最多支持定义60个inputs输入参数。每个input输入参数，均需定义唯一的名称，以便在使用时通过get_input内置函数获取到具体的值。如果重复定义了input，后面定义的将会覆盖之前定义的。

作用范围：node_templates与outputs，即输入参数可以在node_templates的属性及outputs的value中进行传参。

inputs格式：

```
<输入参数名称>:  
type: <类型>  
default: <默认值>  
constraints: <约束>  
description: <描述>  
label: <标签>  
invisible: <是否回显>
```

- **mappings**

可选项，用于定义映射表。在基于模板创建堆栈时，可以根据输入的变量信息，通过get_in_map方法提取特定变量对应的内容。一个模板最多支持定义10个映射。

mappings格式：

```
<映射名称>:  
<映射对象名称>:  
  <映射对象属性名称>: <映射对象属性值>  
  <映射对象属性名称>: <映射对象属性值>  
  ...  
  ...  
  ...
```

- **node_templates**

必填项，用于定义该模板中编排的元素对象集合，其中所有对象均为元素。一个元素可以是一个应用、一个云服务资源。

node_templates格式：

```
<元素名称>:  
type: <元素类型>  
properties: <元素属性>  
requirements: <元素依赖>  
condition: <条件名称>
```

- 元素名称：命名必须唯一，由小写字母、数字和中划线“-”组成，长度为[1, 48]。
- 元素类型：用于指定编排对象的具体类型，来自元素类型列表。
- 元素属性：属性信息基于元素类型展开，每种元素类型会有其相应的属性。属性的变量可以来自inputs获取，也可以通过get_attribute得到运行时的信息。当一个元素没有需要特别声明的属性时，可以不定义properties。
- 元素依赖：可选字段，如果元素与其他元素之间没有关系，则不需要定义。元素的依赖关系来自所定义的元素类型。针对具体的类型可以定义相关的依赖。
- 条件名称：可选字段，用于控制node_templates中的元素是否创建。具体方法详见conditions说明。

- **outputs**

可选项，用于定义模板生成堆栈运行时的输出参数。每个输出参数都需要定义唯一的名称。

outputs格式:

```
<输出参数名称>:  
description: <描述>  
value: <值>
```

- `tosca_definitions_version`
必填项，用于标识模板的版本信息

关于模板的更多信息请参阅[模板参考](#)。

修改公共模板

为了节约容器运行的资源及防止容器超载，可对容器使用的CPU和内存进行申请与限制。本节通过编排公共模板，增加magento前台应用和MySQL数据库应用的CPU和内存的申请与限制参数。

- 申请：表示容器运行所需最少资源。
- 限制：如果担心容器超载，导致系统故障，建议设置容器配额的最高限制，确保容器资源不会超额。

步骤1 登录AOS控制台，选择左侧导航栏的“模板市场”，单击“公共模板”。

步骤2 在“行业场景模板”下，单击**Magento电商应用**，查看模板详情。

图 1-2 选择公共模板



在模板详情中，展示了该模板的概述，以及模板图示。magento应用组中包含了一个magento前台应用和MySQL数据库应用。且magento依赖于MySQL应用，需要将数据存储在MySQL中。

步骤3 单击“复制模板”，复制模板文件到“我的模板”。

图 1-3 复制模板



步骤4 修改“模板名称”，本例为“my-business-app”，单击“确定”，系统跳转到“my-business-app”模板详情页面。

图 1-4 修改模板名称



步骤5 单击操作列的“下载模板”，根据业务需要修改模板信息。以下以增加magento前台应用的CPU和内存的申请与限制参数为例编排模板。

```
name: magento-container
resources:
  requests:
    cpu:
      get_input: requestMagentoCPU
    memory:
      get_input: requestMagentoMemory
  limits:
    cpu:
      get_input: limitMagentoCPU
    memory:
      get_input: limitMagentoMemory
ports:
  - containerPort: 80
    protocol: TCP
```

各属性解释如下：

- resources: 定义容器资源规格。
- requests: 给容器分配的资源配额。
- limits: 容器能够使用的资源上限。
- cpu: 容器使用的CPU需求。
- memory: 容器使用的Memory需求。
- get_input: 用于获取模板文件中inputs区域中定义的输入参数的值。

在inputs部分添加输入参数的描述和标签。

```
inputs:
  app-name:
    default: magento
    description: 应用名称
    label: magento
  mysql-service-name:
    default: magento-mysql
    description: MySQL数据库服务的名称
    label: mysql
  requestMagentoCPU:
    description: MagentoCPU申请
    label: magento
  requestMagentoMemory:
    description: Magento内存申请
    label: magento
  limitMagentoCPU:
    description: MagentoCPU限制
    label: magento
  limitMagentoMemory:
    description: Magento内存限制
    label: magento
  magento-EIP:
    description: magento服务对外暴露访问地址
    label: magento
```

各属性解释如下:

- inputs: 定义基于模板创建堆栈时的可变部分。
- requestMagentoCPU: 定义输入参数Magento应用的CPU申请。
- requestMagentoMemory: 定义输入参数Magento应用的内存申请。
- limitMagentoCPU: 定义输入参数Magento应用的CPU限制。
- limitMagentoMemory: 定义输入参数Magento应用的内存限制。
- description: 参数描述信息。
- label: 参数的标签, 此处定义的标签可在创建堆栈时进行分类展示。

步骤6 保存本地模板文件。

步骤7 在“my-business-app”模板详情页面, 单击“新增版本”, 版本号设置为2.0, 上传本地修改后的模板文件。

---结束

创建 Magento 电商网站应用堆栈

修改后的模板支持在创建堆栈时设置资源限制, 可以分别为magento前台应用和MySQL数据库应用的CPU和内存进行申请与限制。

- CPU配额:

表 1-1 CPU 配额说明

参数	说明
CPU申请	容器使用的最小CPU需求，作为容器调度时资源分配的判断依据。但申请不限制容器的最大可使用CPU。
CPU限制	容器能使用的CPU最大值。

建议配制方法：节点的实际可用分配CPU量 \geq 当前实例所有容器CPU限制值之和 \geq 当前实例所有容器CPU申请值之和，节点的实际可用分配CPU量请在**CCE控制台**“资源管理 > 节点管理”中对应节点的“可用CPU (Core)”列下查看。

- 内存配额：

表 1-2 内存配额说明

参数	说明
内存申请	容器使用的最小内存需求，作为容器调度时资源分配的判断依据。
内存限制	容器能使用的内存最大值。当内存使用率超出设置的内存限制值时，该实例可能会被重启进而影响应用的正常使用。

建议配制方法：节点的实际可用分配内存量 \geq 当前实例所有容器内存限制值之和 \geq 当前实例所有容器内存申请值之和，节点的实际可用分配内存量请在**CCE控制台**“资源管理 > 节点管理”中对应节点的“可用内存 (GB)”列下查看。

步骤1 在左侧导航栏中，选择“我的模板”。在模板列表中可查看到保存成功的模板**my-business-app**。

单击my-business-app模板后的“创建堆栈”。

步骤2 输入堆栈信息。

- 堆栈名称：自定义堆栈名称，例如**my-magento**。
- 描述：可不填写。
- 选择集群：选择已创建的集群。
- 命名空间：保持默认default。
- 在magento页签中配置magento应用输入参数，如**图1-5**。下图红框中为**修改公共模板节**中新增参数。

图 1-5 magento 应用输入参数

配置

标签: magento mysql

参数名称	类型	参数值	参数描述
app-name	string	<input type="text" value="magento"/>	应用名称
limitMagentoCPU	string	<input type="text"/>	MagentoCPU限制
limitMagentoMemory	string	<input type="text"/>	Magento内存限制
magento-EIP	string	<input type="text"/>	magento服务对外暴露访问地址
magento-EPORT	integer	<input type="text" value="32080"/>	magento服务对外监听端口
requestMagentoCPU	string	<input type="text"/>	MagentoCPU申请
requestMagentoMemory	string	<input type="text"/>	Magento内存申请

表 1-3 配置 magento 应用输入参数

参数	说明	参数值
limitMagentoCPU	通过修改公共模板增加的输入参数，Magento应用的CPU限制。	根据应用实际情况填写，数值后无需加上单位，默认单位为核，例如：1。
limitMagentoMemory	通过修改公共模板增加的输入参数，Magento应用的内存限制。	根据应用实际情况填写，数值后加上单位M，例如：2048M。
magento-EIP	节点的弹性IP	请从 前提条件 中获取弹性IP数值，例如10.0.0.0。
magento-EPORT	节点端口	请输入30000-32767之间的整数，请保证集群内唯一。可保持默认32080。
requestMagentoCPU	通过修改公共模板增加的输入参数，Magento应用的CPU申请。	根据应用实际情况填写，数值后无需加上单位，例如：0.5。
requestMagentoMemory	通过修改公共模板增加的输入参数，Magento应用的内存申请。	根据应用实际情况填写，数值后加上单位M，例如：1024M。

步骤3 单击“下一步”，查看堆栈信息，确认无误后，单击“创建堆栈”。

可查看到当前堆栈为“创建中”，预计需要5分钟创建成功。

步骤4 待堆栈创建成功，单击“堆栈详情”，可查看到堆栈状态为“正常”，“堆栈元素”中已存在六个云服务。

图 1-6 堆栈创建成功

堆栈元素 | 输出参数 | 输入参数 | 告警 | 事件

元素名称	类型	资源名称	健康状态	规格	操作状态
magento	CCE.Deployment	magento	健康	类型 Deployment	● 创建成功
magento-config	CCE.ConfigMap	magento-config-9da7900a	健康	类型 ConfigMap	● 创建成功
magento-service	CCE.Service	magento	健康	类型 Service	● 创建成功
mysql	CCE.Deployment	magento-mysql	健康	类型 Deployment	● 创建成功
mysql-conf	CCE.Secret	magento-mysql	健康	名称 密钥类型 magento-mysql Opaque	● 创建成功
mysql-service	CCE.Service	magento-mysql	健康	类型 Service	● 创建成功

步骤5 在“输出参数”页签中，可查看到三个输出参数。

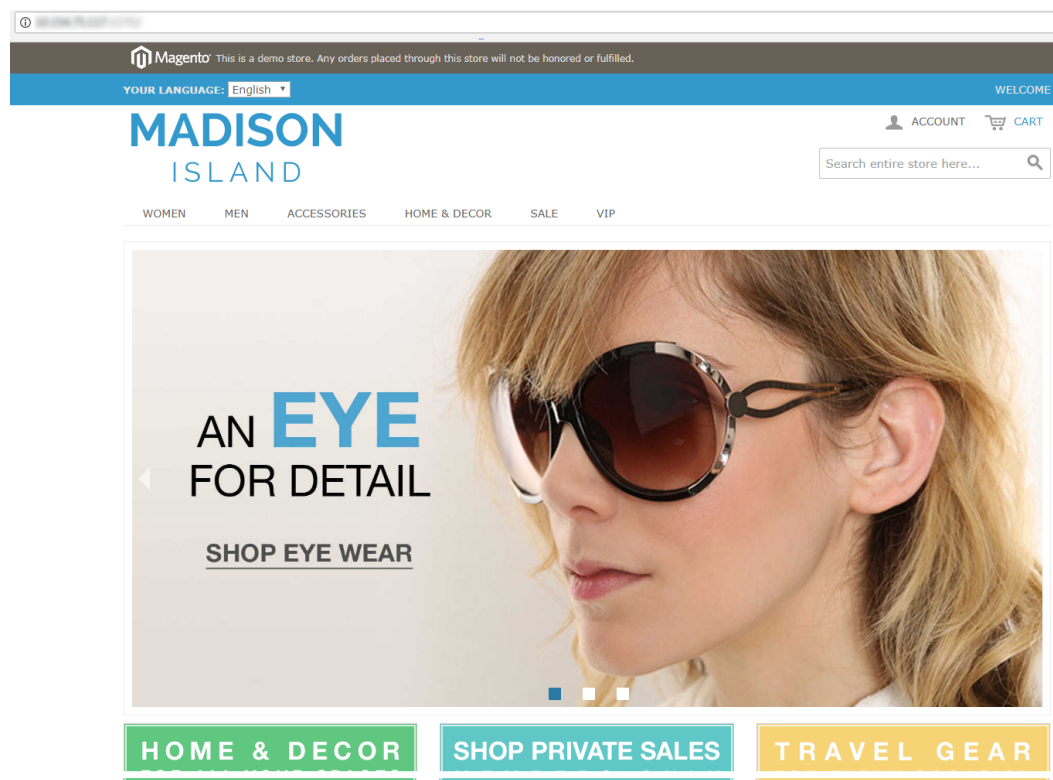
图 1-7 输出参数

堆栈元素 | 输出参数 | 输入参数 | 告警 | 事件

输出项	输出值	描述
ingress-admin_password	magentorocks1	Password of super user.
magento-addr	http://[redacted]:32080	Access URL for magento service.
magento-admin_username	admin	Super user name.

步骤6 在浏览器中输入输出参数中获取的“magento-addr”，例如http://10.0.0.0:32080。可访问magento应用。

图 1-8 访问 magento 应用



----结束

清除资源

为了防止您为不需要的服务付费，建议您删除堆栈，清理资源。

步骤1 登录AOS控制台。

步骤2 在左侧导航栏中，单击“我的堆栈”。

步骤3 勾选已创建成功的堆栈，单击“删除堆栈”，根据界面提示删除堆栈。

----结束