|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
|  | 最佳实践 | |  |
| 单品大数据产品解决方案 | |
| **文档版本** | **V1.0** |
| **发布日期** | **2021-04-13** |
| 华为网格系统---方案4-032.png | | | |

|  |  |
| --- | --- |
| 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](mailto:support@huawei.com) |
| 客户服务电话： | 4008302118 |

目 录

目录

[目 录 2](#_Toc74255060)

[1 使用前必读 3](#_Toc74255061)

[1.1 概述 3](#_Toc74255062)

[1.2 基本概念 3](#_Toc74255063)

[2 最佳实践方案架构 4](#_Toc74255064)

[2.1 适用场景 4](#_Toc74255065)

[2.2 方案架构 4](#_Toc74255066)

[2.3 方案优势 5](#_Toc74255067)

[3 单品大数据操作步骤 6](#_Toc74255068)

[3.1 云服务资源准备 6](#_Toc74255069)

[3.1.1 ECS云服务器创建 7](#_Toc74255070)

[3.1.2 VPC&安全组创建 8](#_Toc74255071)

[3.2 单品大数据应用安装部署 10](#_Toc74255072)

[3.2.1 单品大数据安装 10](#_Toc74255073)

# 使用前必读

## 概述

*众所周知，在任何系统的使用过程中，高性能和高可用都属于基本诉求。为了实现业务系统的高可用，两地三中心部署成为了高可用保障的基础方案，同时相对于双活方案，两地三中心的方案对业务的侵入性小，业务改造量小。华为云提供多种不同场景下的解决方案，在不改动或少改动企业现有IT系统的情况下，可快速实现数字转型。*

基本概念

* 账号

用户注册时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用他们进行日常管理工作。

* 用户

由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。

在[我的凭证](https://console.huaweicloud.com/iam/" \l "/myCredential" \o " )下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。

* 区域

指云资源所在的物理位置，同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户的要求，或满足不同地区的法律或其他要求。

* 可用区

一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。

* 项目

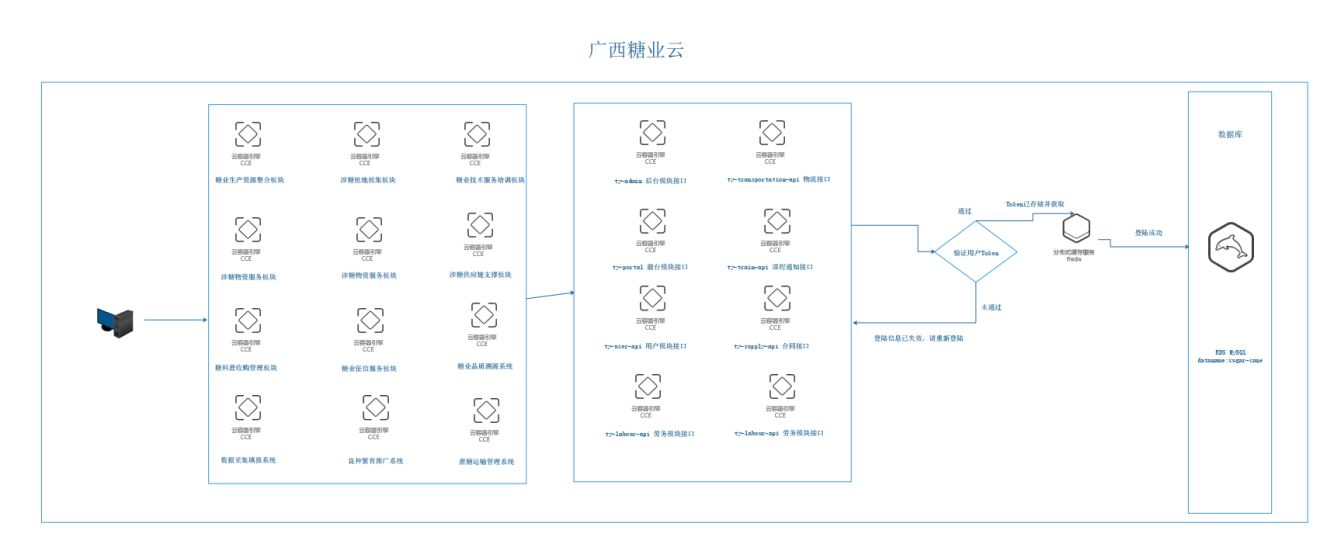
区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中创建资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

# 最佳实践方案架构

## 适用场景

*在传统运维中，自建或者IDC硬件资源不足或硬件资源过剩导致不必要的浪费，从而增加IT资源成本，而对于日常业务场景中，特别是电商业务难免会有促销活动从而导致某一个时刻高流量高并发，往前的资源过剩在高并发场景中显得不尽人意，而使用华为云，可以灵活使用弹性计算资源，按时按量收费，按需秒开，可以做到快速交付投入生产，使用完立即施放，大大增加企业选择性、灵活性，更重要的是解决了高并发场景的资源计算问题，直线型的降低成本，帮助企业更好上云，解决传统运维难题，提高效率。*

## 方案架构



方案架构说明：

*1.前端模块、后端应用解耦，单独拆分微服务部署，通过容器化的方式部署在CCE集群；*

*2.核心数据存储在RDS for MySQL 5.7，DCS Redis用于存放登陆信息及热点数据；*

*3.所有服务通过CCE中service类型中的负载均衡ELB绑定公网进行接入并作为外界访问系统入口；*

*6.运维人员通过实现准备好的镜像，直接控制CCE进行快速迭代部署，提高整体工作效率；*

## 方案优势

*1、前端使用nginx作为web服务器，每个前端模块与各模块之间解耦，独立运行，使用CCE容器方式部署；根据不同流量业务场景选择副本数，大大提高单一服务的灵活性，可以充分的利用各节点的硬件资源。又可以根据不同的服务做流量预估，从而更加一步提高资源利用率，提高系统高并发与降低低延迟。*

*2、后端使用springboot作为微服务改造，各个后端服务从原来的“大杂烩”按照功能需求拆分成多个单独的子模块，按照业务场景，通过CCE创建无状态服务进行自由的副本伸缩，提高硬件资源利用率。*

*3、核心数据库从传统的自建服务迁移到华为云所提供的云RDS MySQL 5.7版本，可以很直观的调节硬件资源配置，显著的提供数据库性能读写，运维人员无需关心数据库服务进程是否异常、或因为硬件资源故障导致整个服务器宕机，云资源可以自动的根据当前配置进行参数优化，达到最优状态，降低运维因参数配置不正确导致“负优化”，数据库帐号管理、库管理、用户授权、日常备份与恢复可通过华为云控制台直接进行操作，大幅度提高核心数据库的稳定与性能。*

*4、redis从传统的自建集群迁移到华为云所提供的云redis，根据业务场景要求选择节点数，自动集群化， 稳定性大大提高，运维人员只需关注资源是否足够使用，无需担心因为服务异常导致集群脑裂影响业务稳定性。*

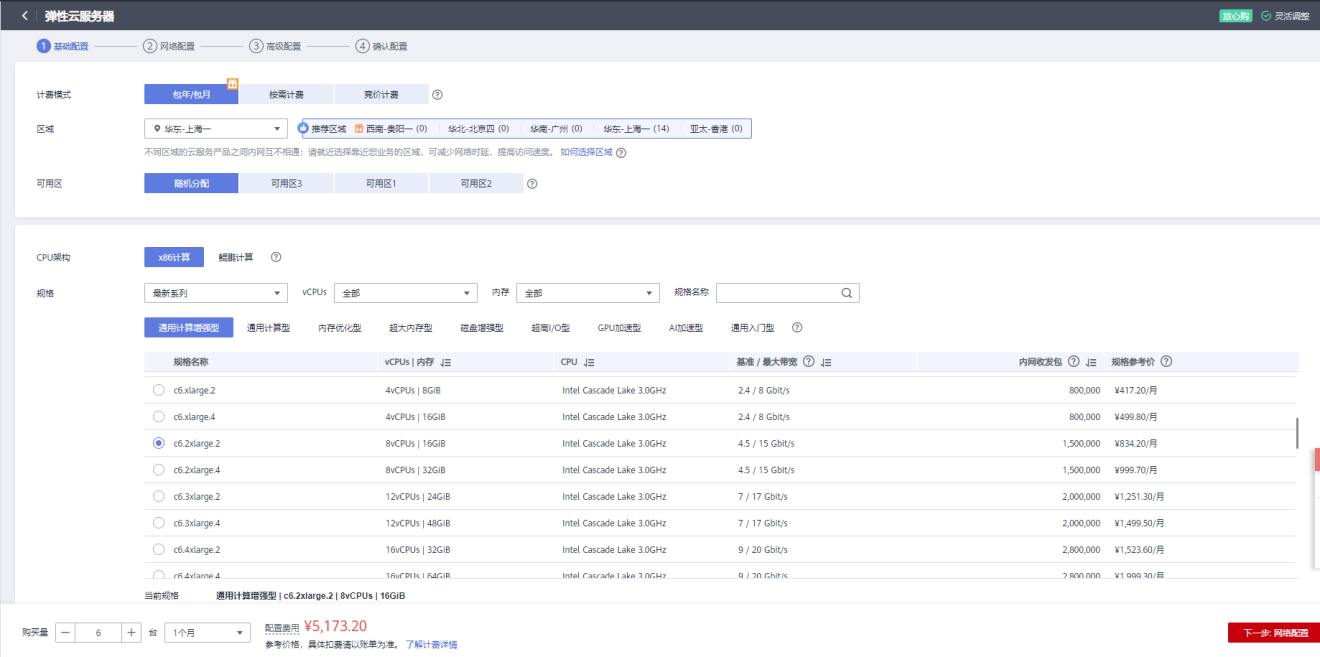
# 单品大数据操作步骤

## 云服务资源准备

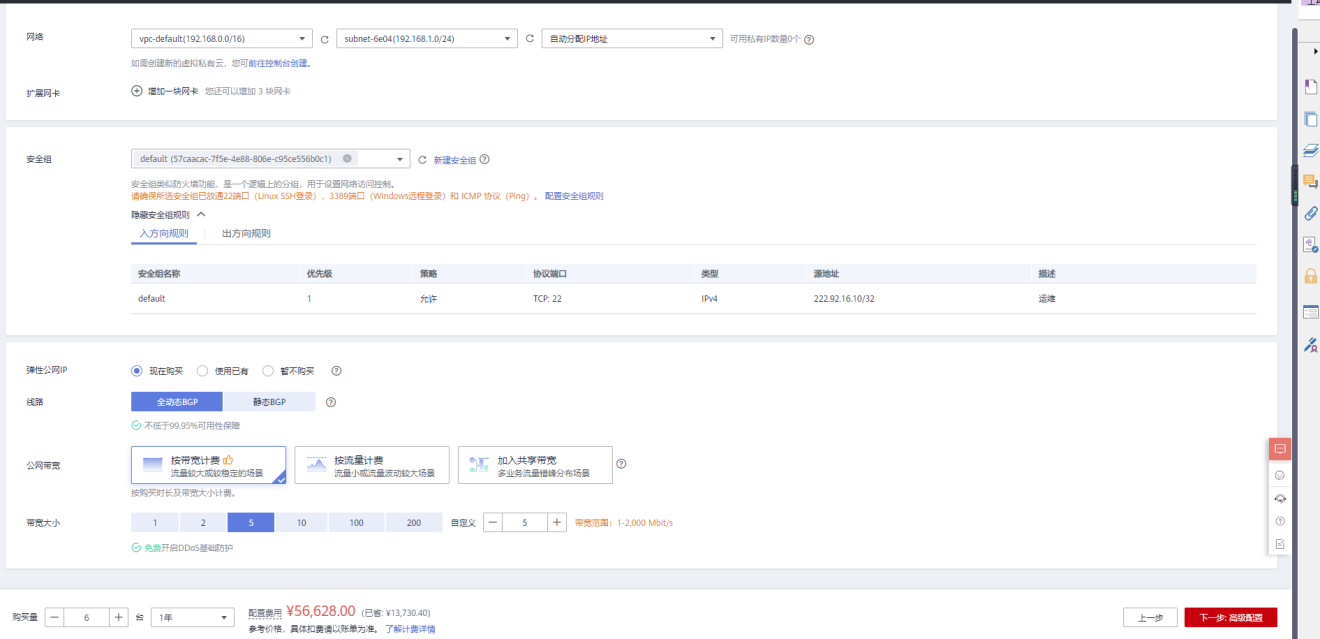
云服务资源清单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **资源名称** | **配置规格** | **命名示例** | **Region** | **数量** | **说明** |
| 1 | VPC | 默认 | vpc-default | 上海一、可用区一 | 1 | 所以资源使用同一个VPC，从而内网互通。 |
| 2 | 安全组 | 根据需要开通入方向端口 | x86安全组  [x86-bric-cce-node-4mpcd](https://console.huaweicloud.com/vpc/?agencyId=0c1e3a0f4400f4c61f4dc01a7c48dbb6&region=cn-east-3&locale=zh-cn" \l "/secGroups/SGDetail?instanceId=091d3ada-82f5-402d-97f0-969897191db9)  [x86-bric-cce-control-4mpcd](https://console.huaweicloud.com/vpc/?agencyId=0c1e3a0f4400f4c61f4dc01a7c48dbb6&region=cn-east-3&locale=zh-cn" \l "/secGroups/SGDetail?instanceId=5a8c8b24-cc99-4c1c-8b48-20642f81b741) | 上海一、可用区一 | 3 | 确保ECS、CCE、RDS、云Redis集群所在安全组可以互相访问。 |
| 3 | ECS | 通用计算增强型 | c6s.2xlarge.2 | 8vCPUs | 16GiB 6台 | nginx、应用A、应用B、应用C、应用D、应用E | 上海一、可用区一 | 6 |  |
| 4 | RDS | rds.mysql.n1.2xlarge.2.ha | 8 vCPUs | 16 GB | rds-f6c2 | 上海一、可用区一 | 1 |  |
| 5 | redis | 16 GB | 副本数：2 | x86系列 | 上海一、可用区一 | 1 |  |
| 6 | OBS | 5T | 默认 | 上海一、可用区一 | 1 |  |

### ECS云服务器创建



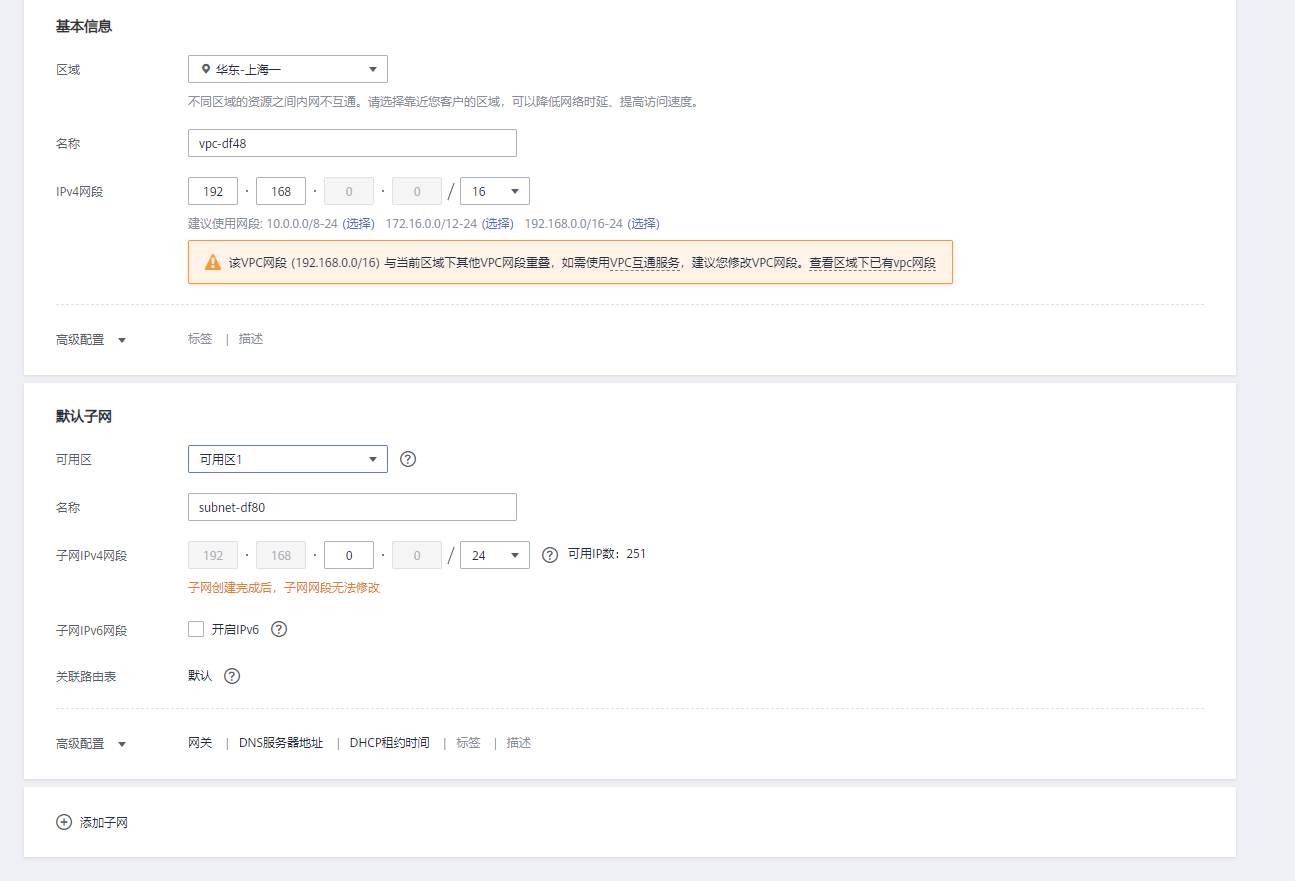




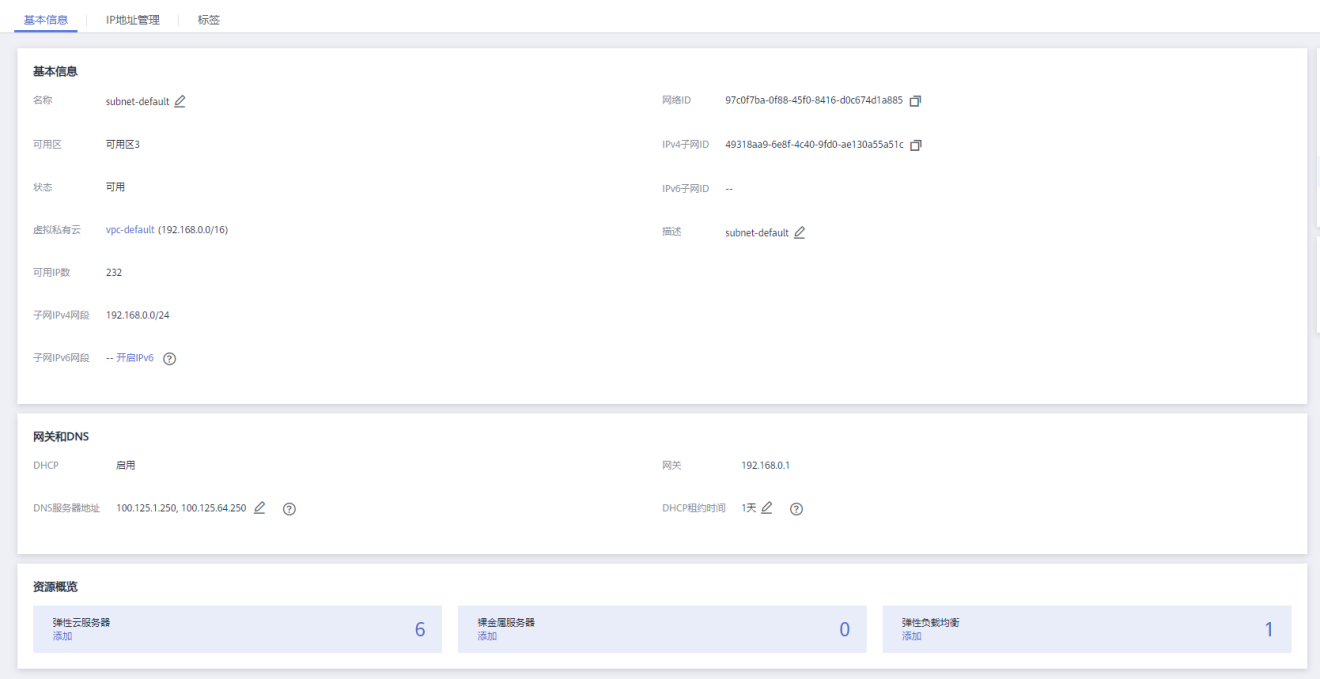
### *VPC&安全组创建*

*创建步骤如下：*

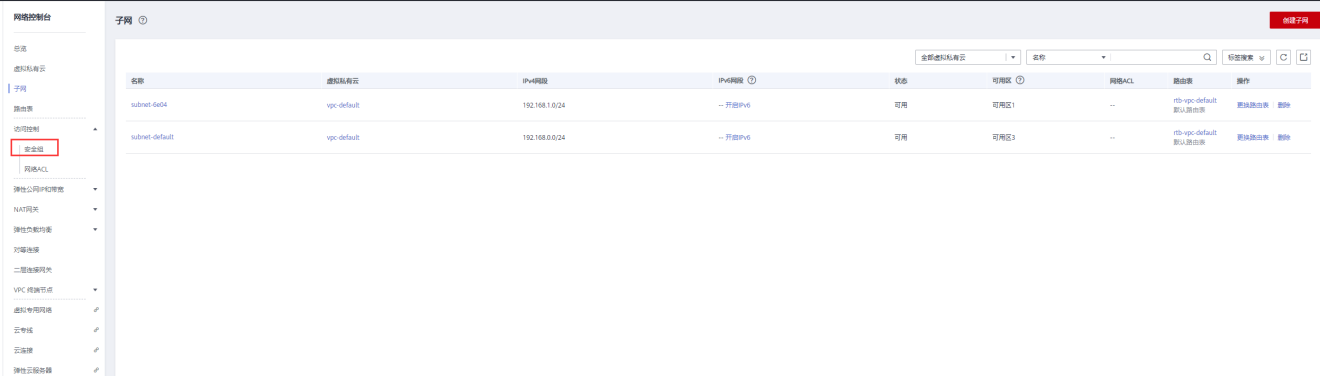
1. *登录华为云，进入[网络控制台](https://console.huaweicloud.com/vpc/?region=cn-south-1" \l "/vpcs)*
2. *在左侧菜单栏选择“虚拟私有云”，点击“创建虚拟私有云”*

* *自定义VPC名称，vpc-default*
* *自定义网段，192.168.0.0/16*
* *子网配置设置成192.168.0.0/24* 

1. *其他默认。点击“立即创建”，完成VPC的创建。可点击进入查看创建好的VPC详情。*



1. *在网络控制台左侧菜单栏选择“访问控制”-“安全组”，点击“创建安全组”，名称自定义，如：sg-smartcampus（这个安全组是为了后续给CDM连接数据源使用）*





1. *点击确定，完成安全组的创建*
2. *根据业务需求开通端口*

## 单品大数据应用安装部署

### *单品大数据安装*

事先准备好糖业生产资源整合模块、涉糖值地梳集板块、糖业技术培训模块、涉糖物资、涉糖、涉糖供应链、糖料蔗收购管理系统、糖业征信服务、糖业品质溯源、数据采集填报、良种繁育推广、蔗糖运输系统镜像

1、try\_planting\_api 安装

1.1、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

1.2、克隆try\_planting\_api源码

git clone –b dev <http://git.bric.com/java/sugar-guangxi.git>

1.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.6.3/bin/mvn package

1.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:1.8

COPY try\_planting\_api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/try\_planting\_api.jar","--spring.profiles.avtive=dev"]

1.5、打包成镜像

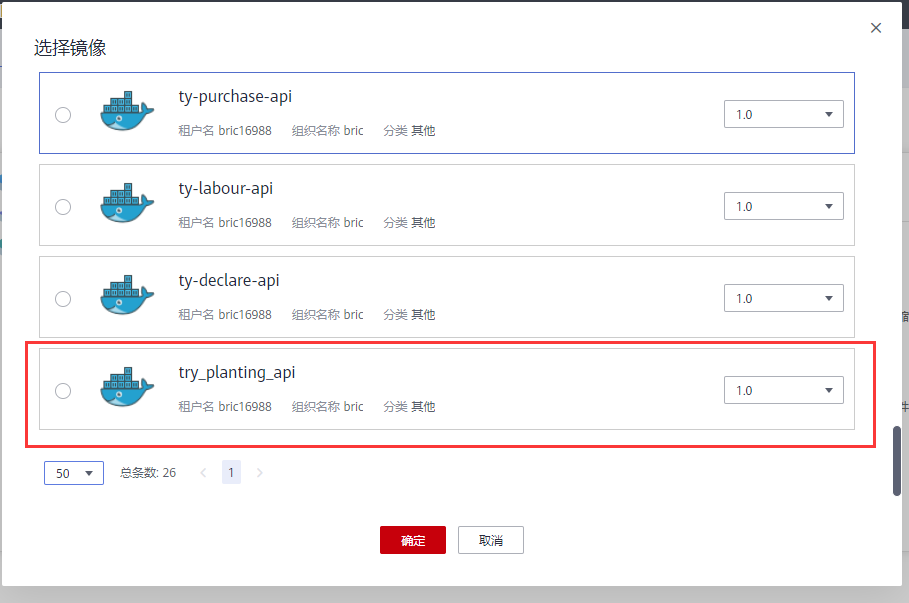
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/try\_planting\_api .

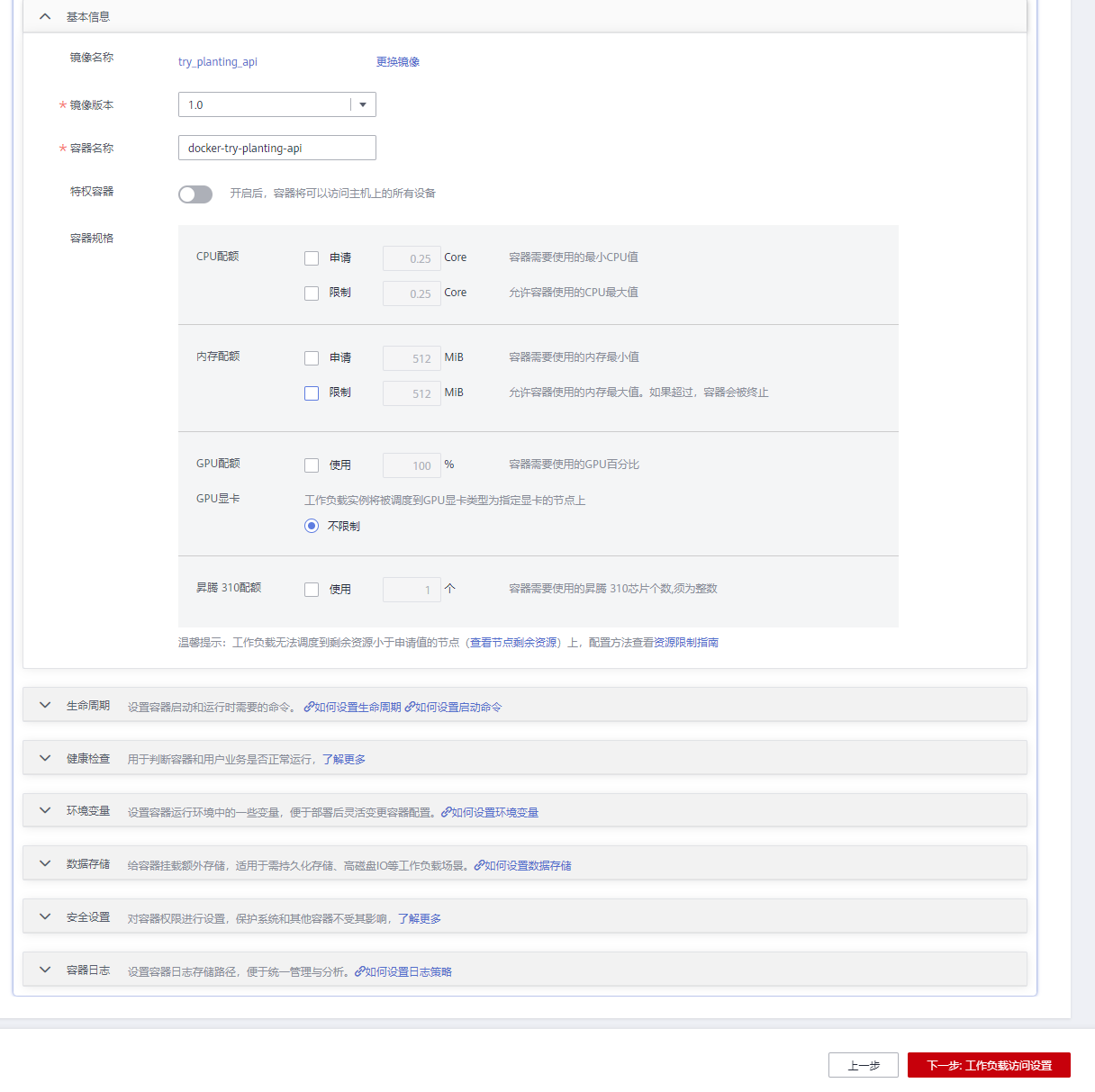
1.6、上传到华为镜像仓库

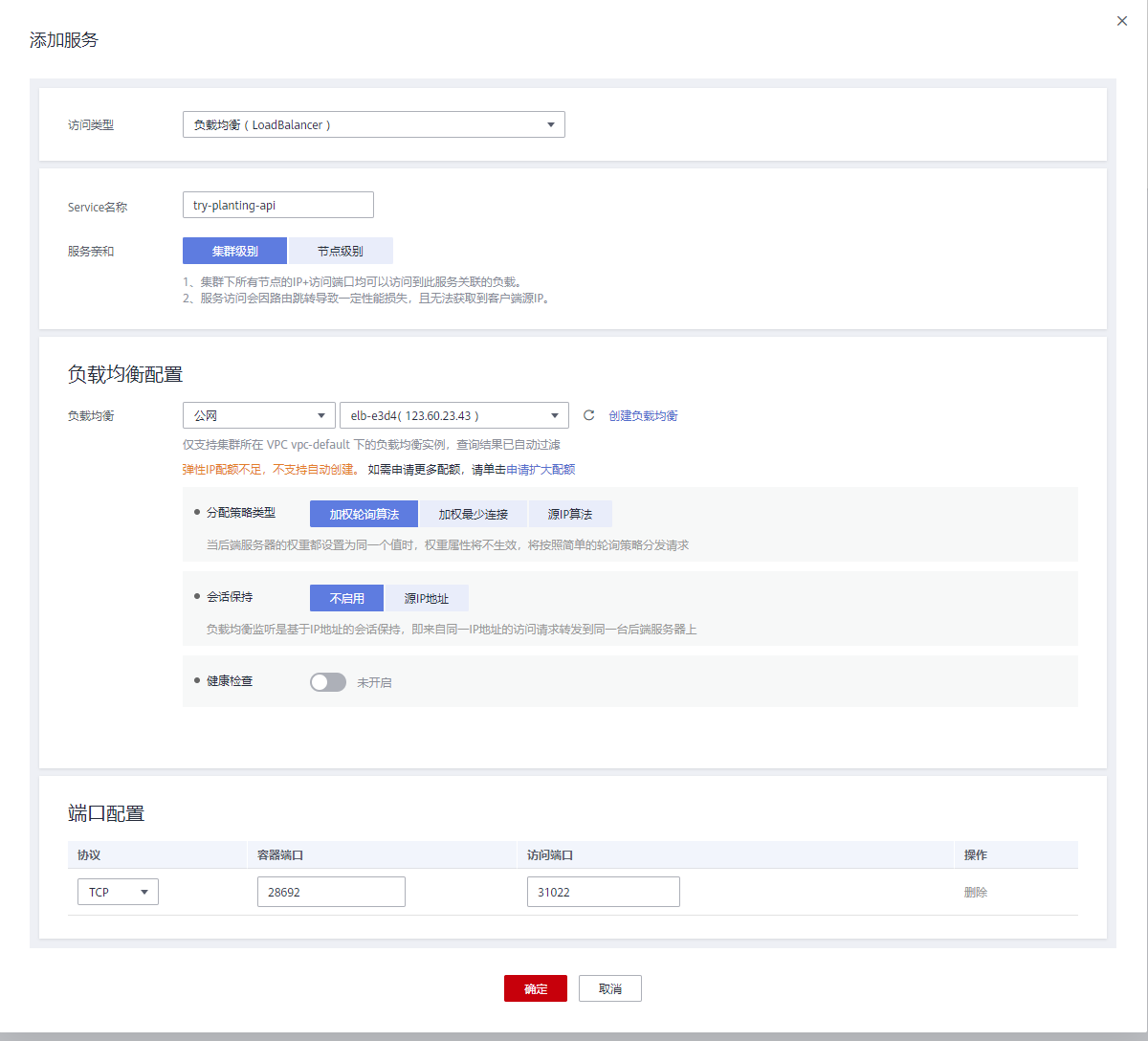
docker push swr.cn-east-3.myhuaweicloud.com/bric/try\_planting\_api

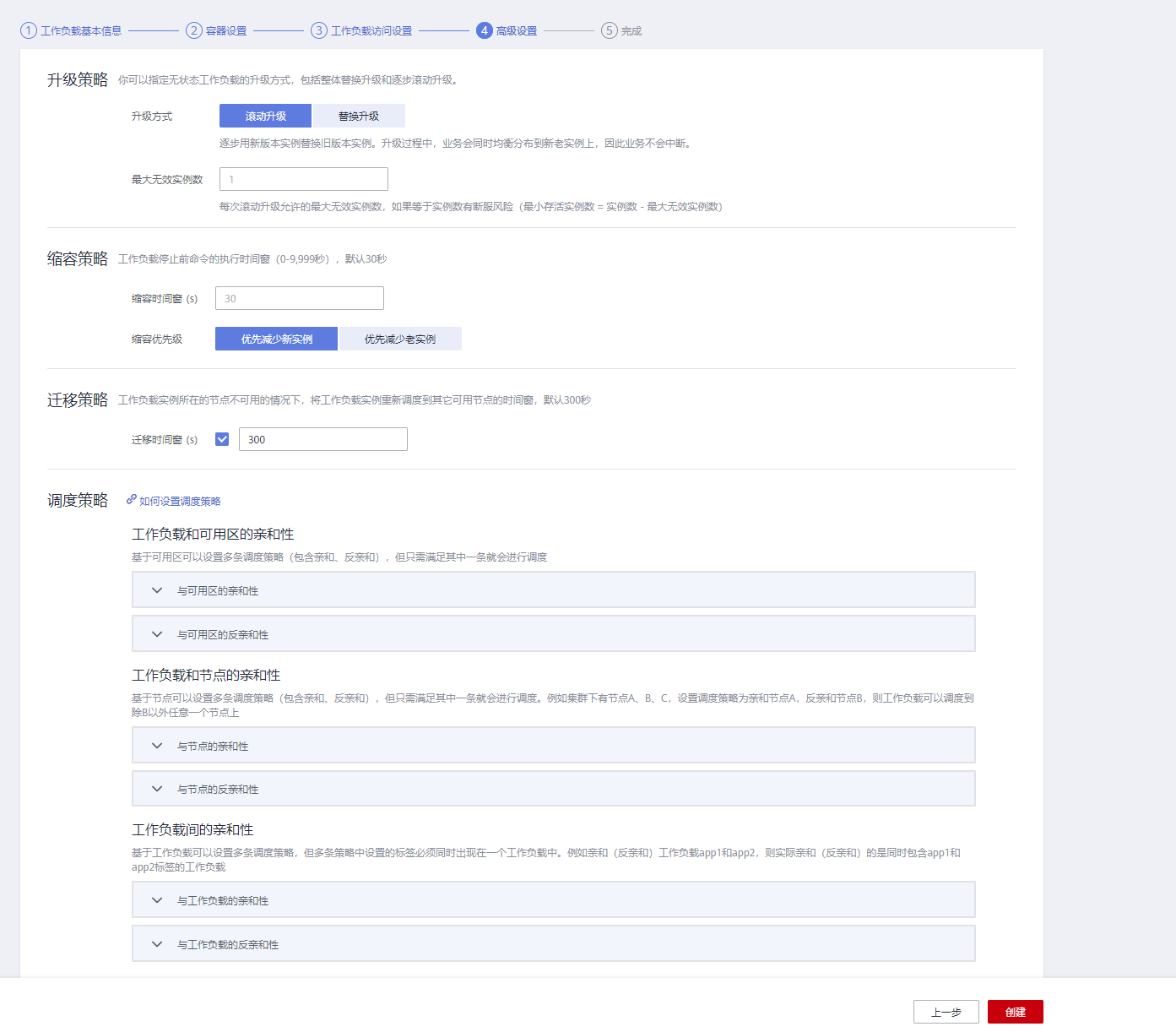
1.7、创建无状态工作负载













2、ty-portal 安装

2.2、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

2.2、克隆ty-portal源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

2.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.6.3/bin/mvn package

2.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:2.8

COPY ty-portal.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-portal.jar","--spring.profiles.avtive=dev"]

2.5、打包成镜像

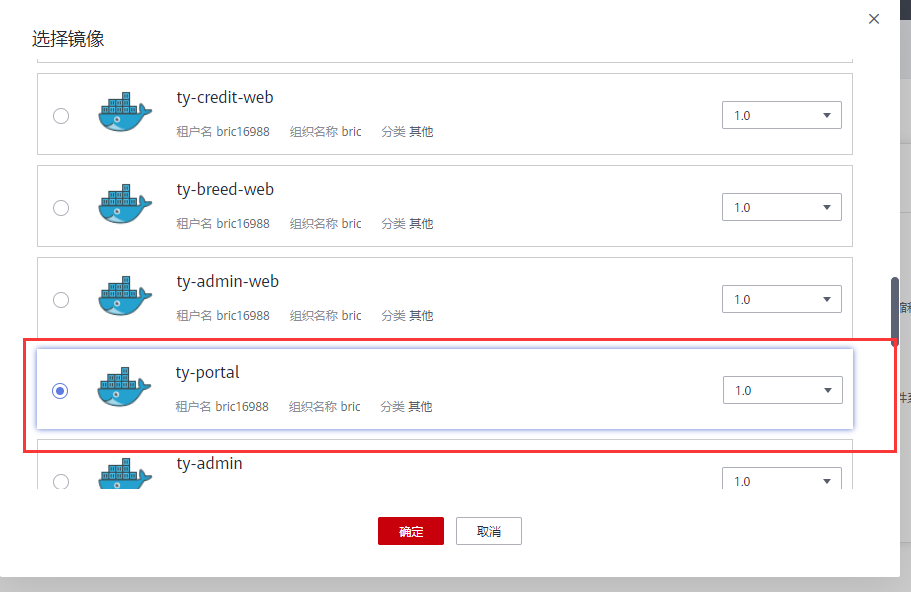
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-portal .

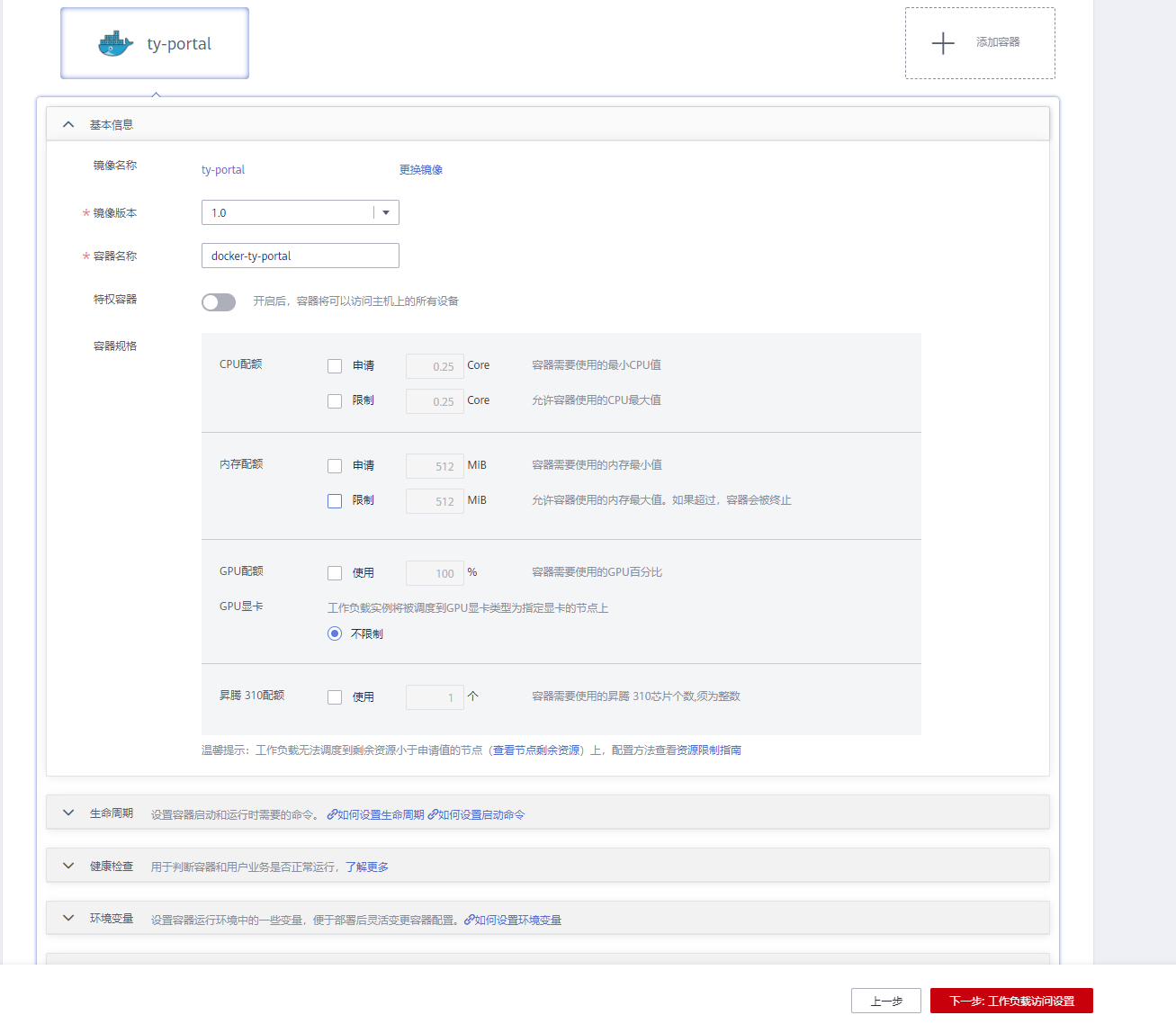
2.6、上传到华为镜像仓库

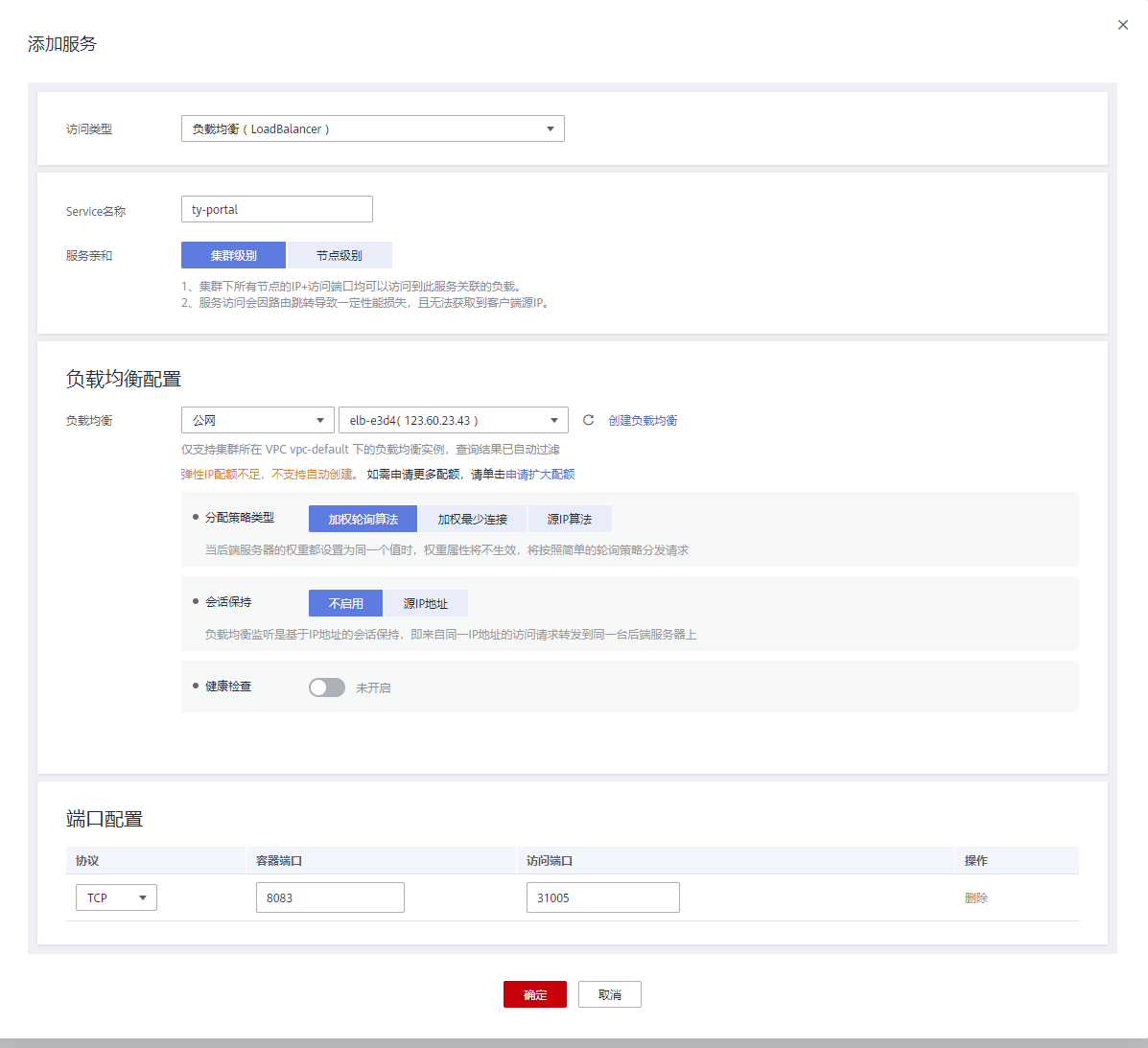
docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-portal

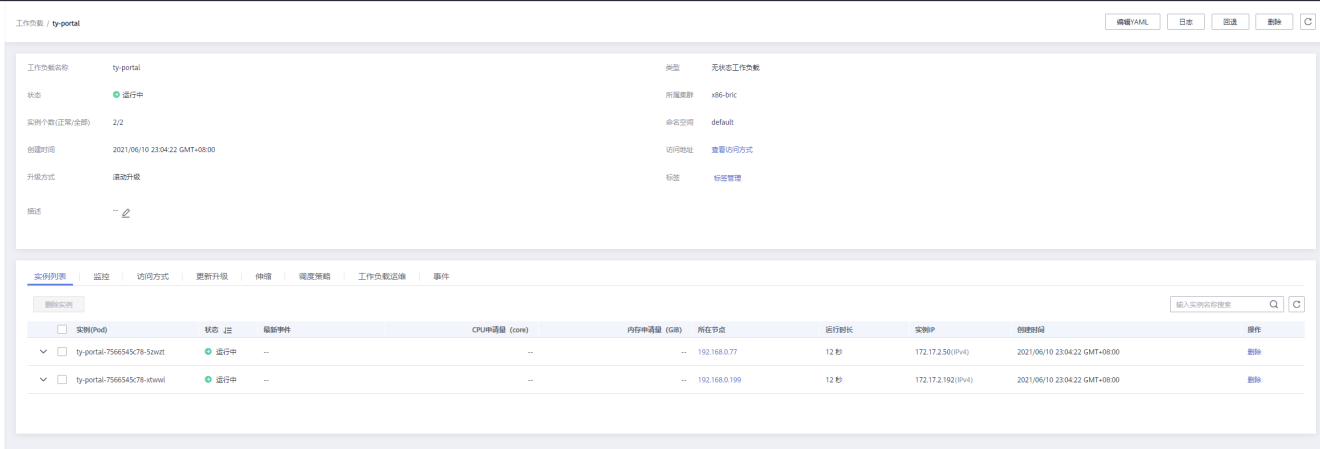
2.7、CCE创建无状态工作负载及访问



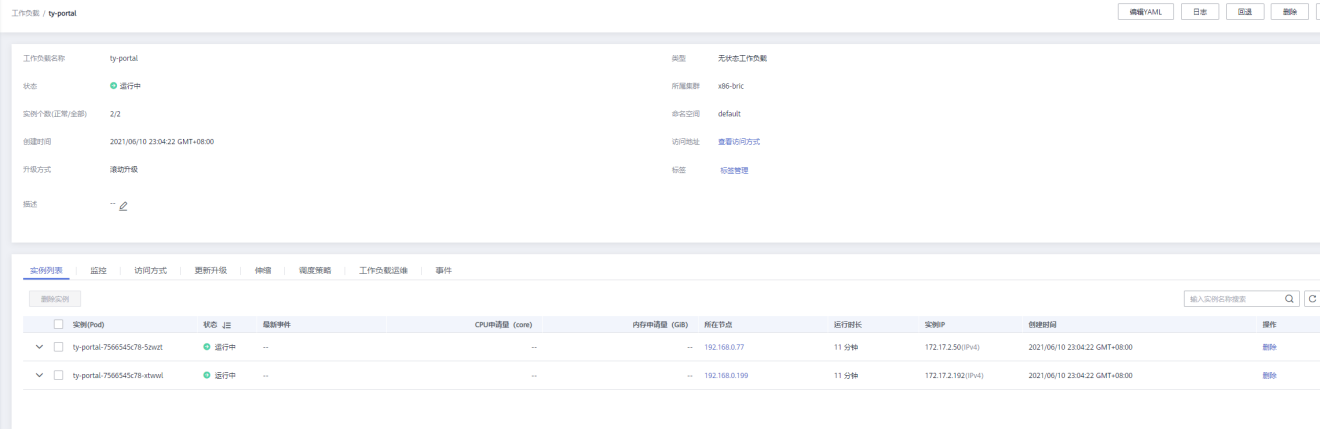












3、ty-admin 安装

3.3、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

3.3、克隆ty-admin源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

3.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.6.3/bin/mvn package

3.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:3.8

COPY ty-admin.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-admin.jar","--spring.profiles.avtive=dev"]

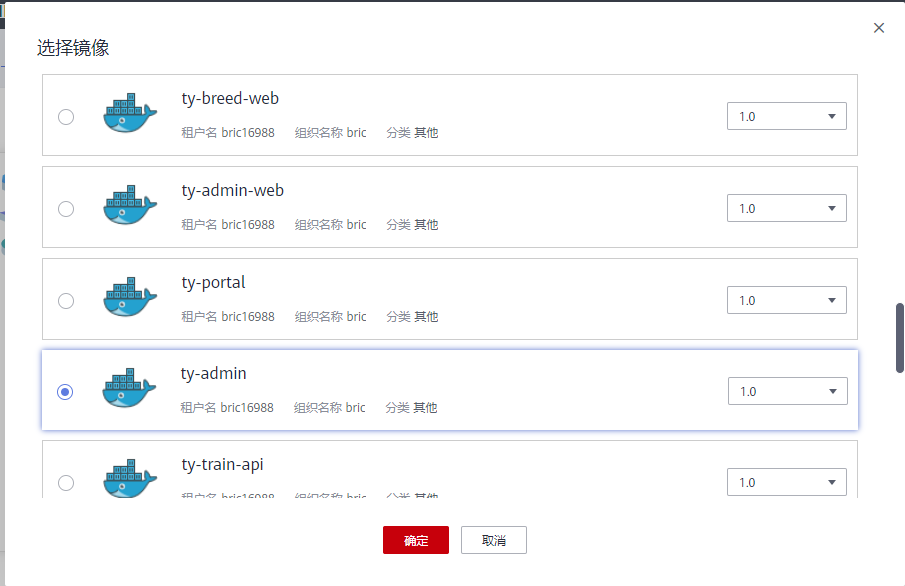
3.5、打包成镜像

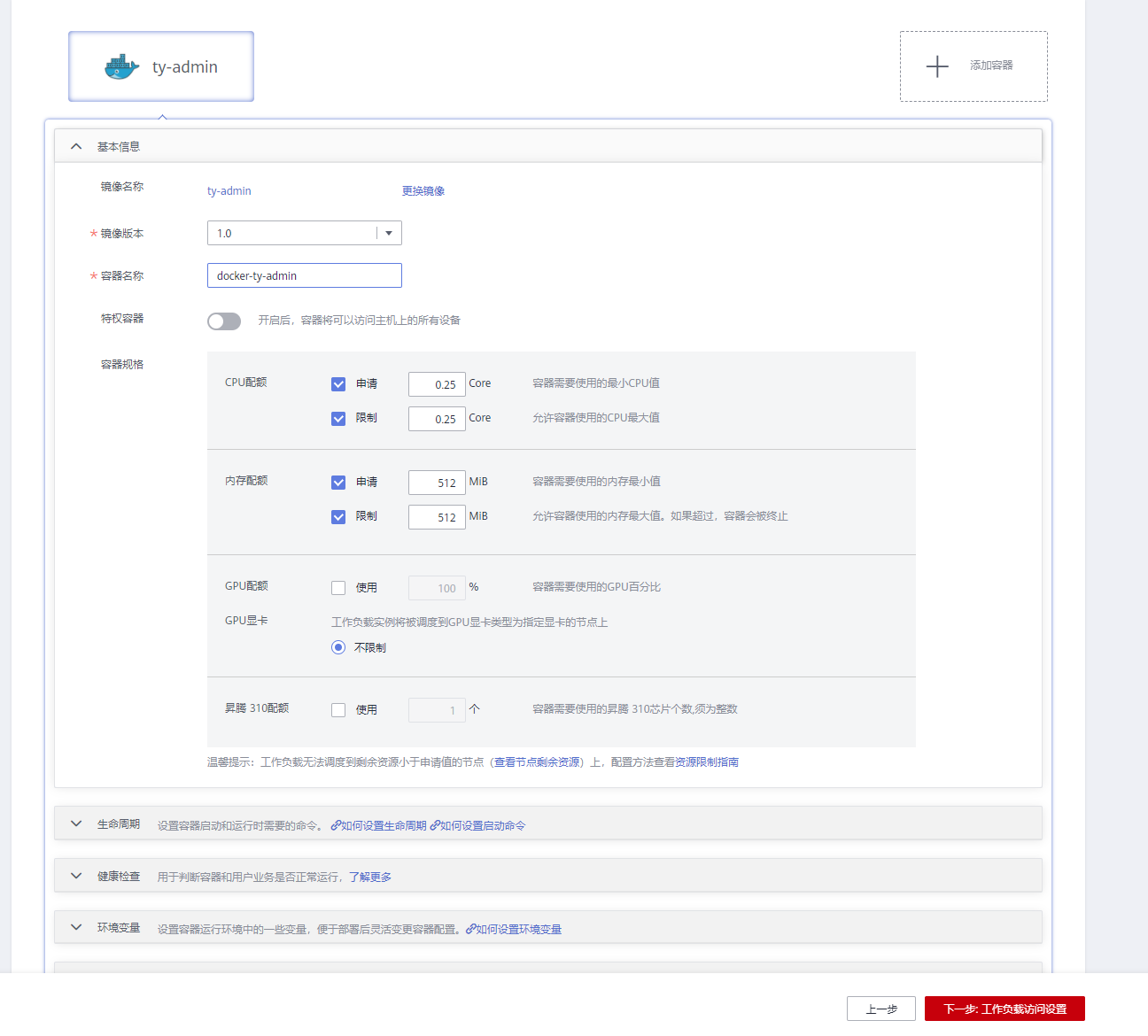
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-admin .

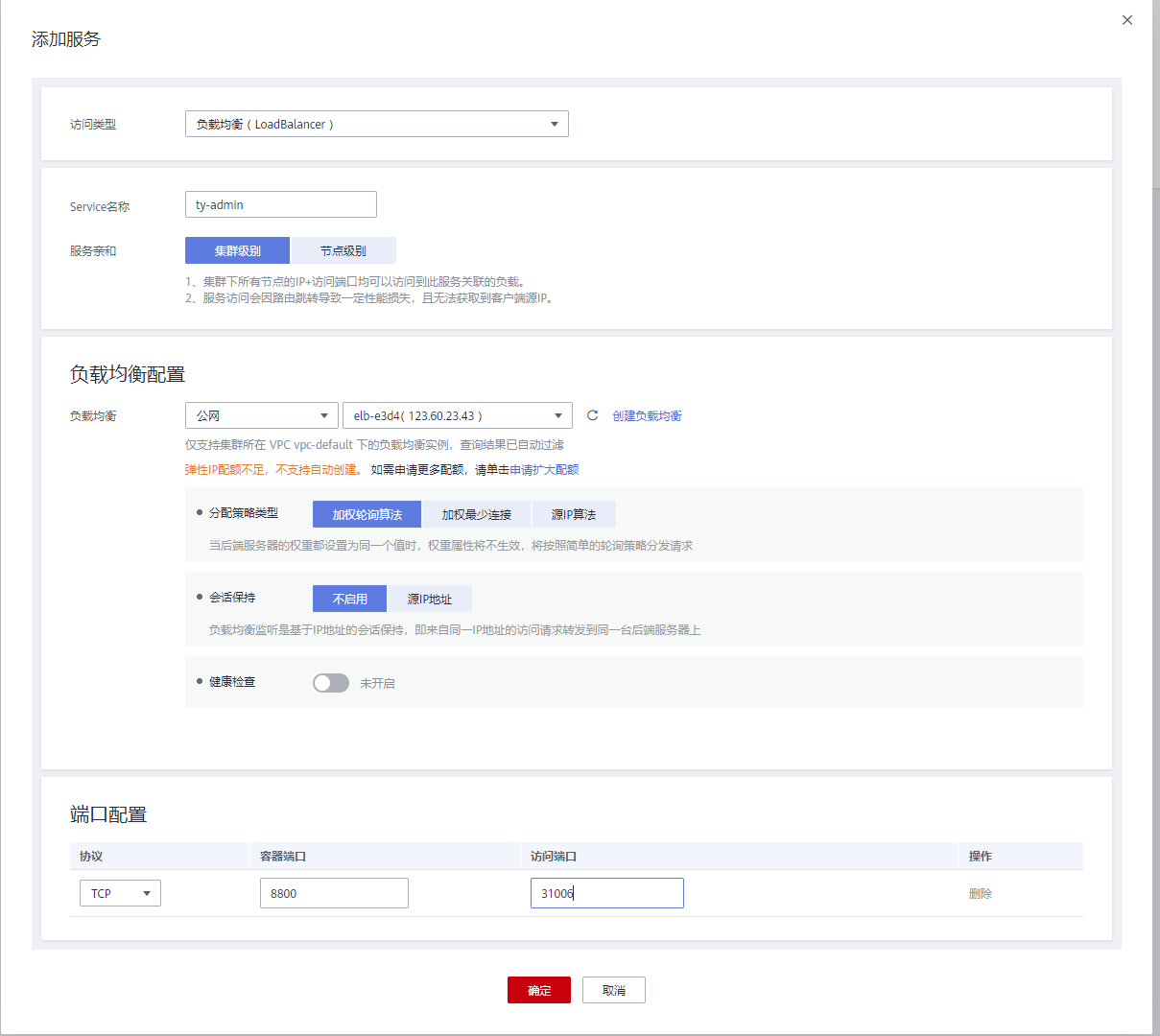
3.6、上传到华为镜像仓库

docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-admin

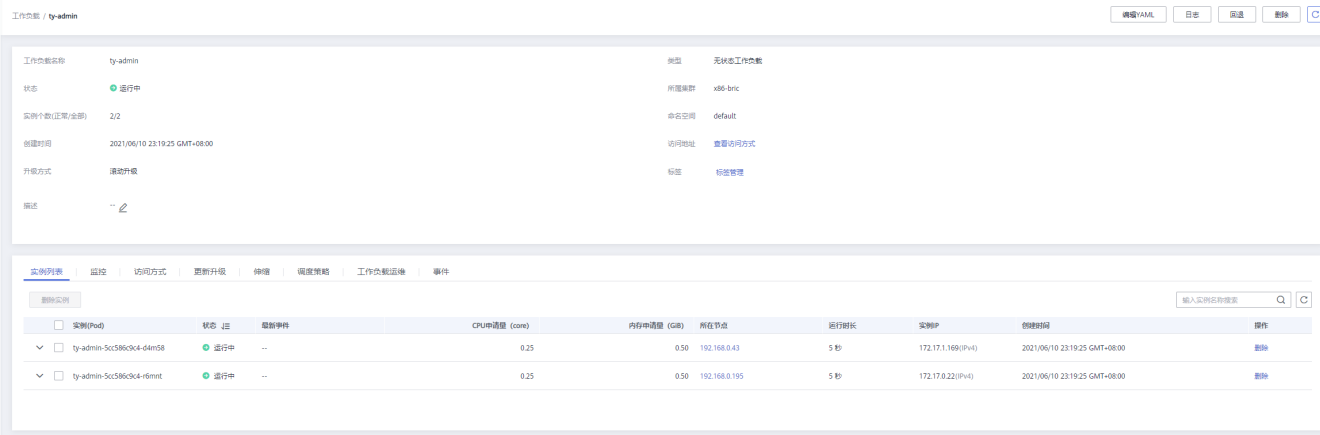




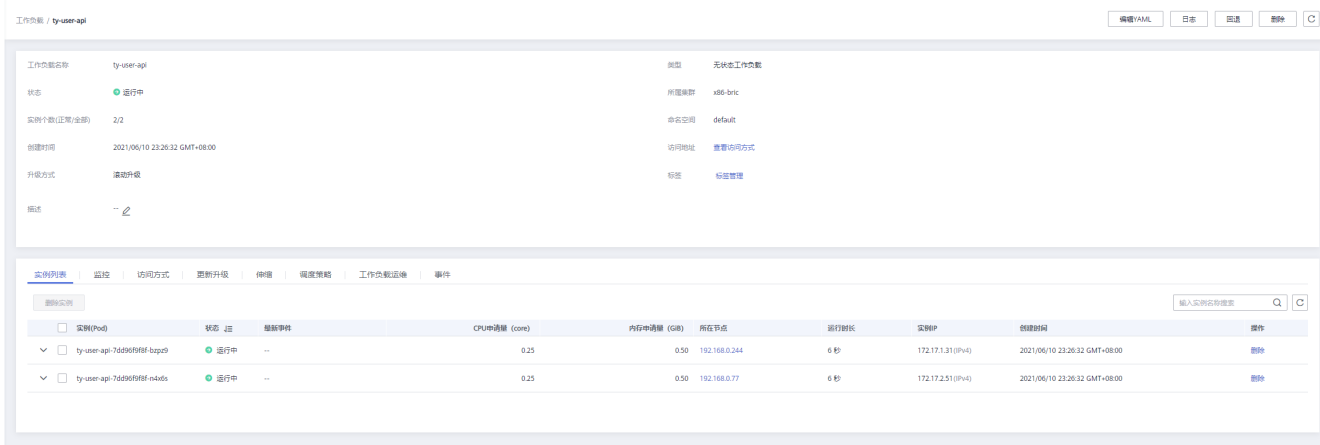












4、ty-user-api 安装

4.4、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

4.4、克隆ty-user-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

4.4、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-4.6.4/bin/mvn package

4.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:4.8

COPY ty-user-api.jar /bric/

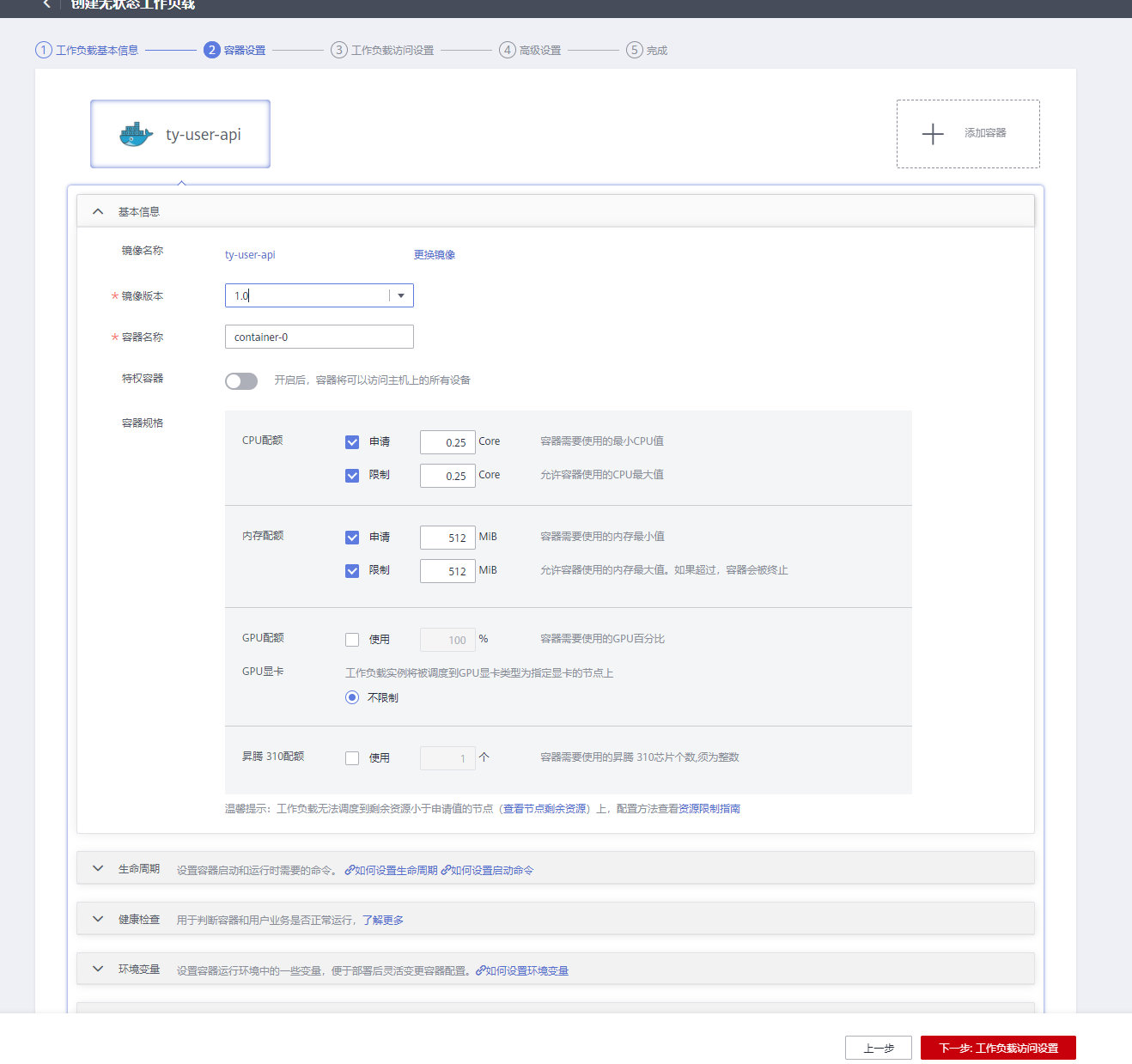
ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-user-api.jar","--spring.profiles.avtive=dev"]

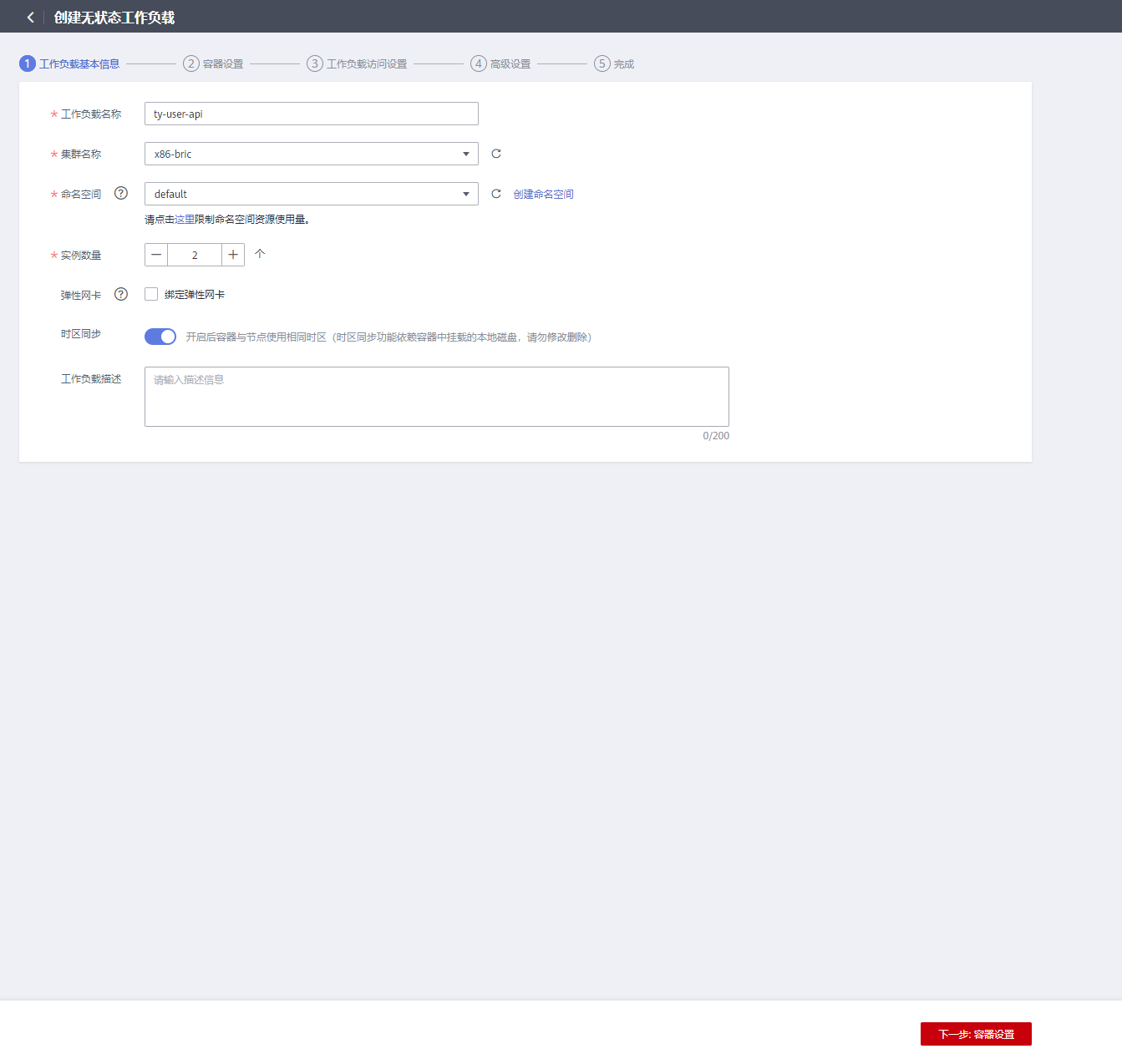
4.5、打包成镜像

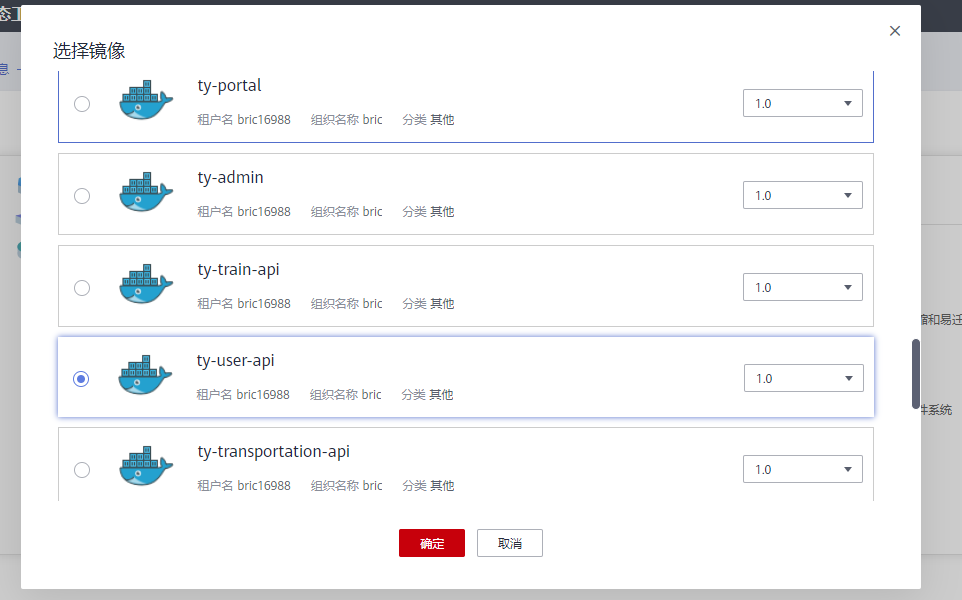
docker build –t swr.cn-east-4.myhuaweicloud.com/bric/ty-user-api .

4.6、上传到华为镜像仓库

docker push swr.cn-east-4.myhuaweicloud.com/bric/ty-user-api







5、ty-transportation-api 安装

5.5、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

5.5、克隆ty-transportation-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

5.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.6.3/bin/mvn package

5.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:5.8

COPY ty-transportation-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-transportation-api.jar","--spring.profiles.avtive=dev"]

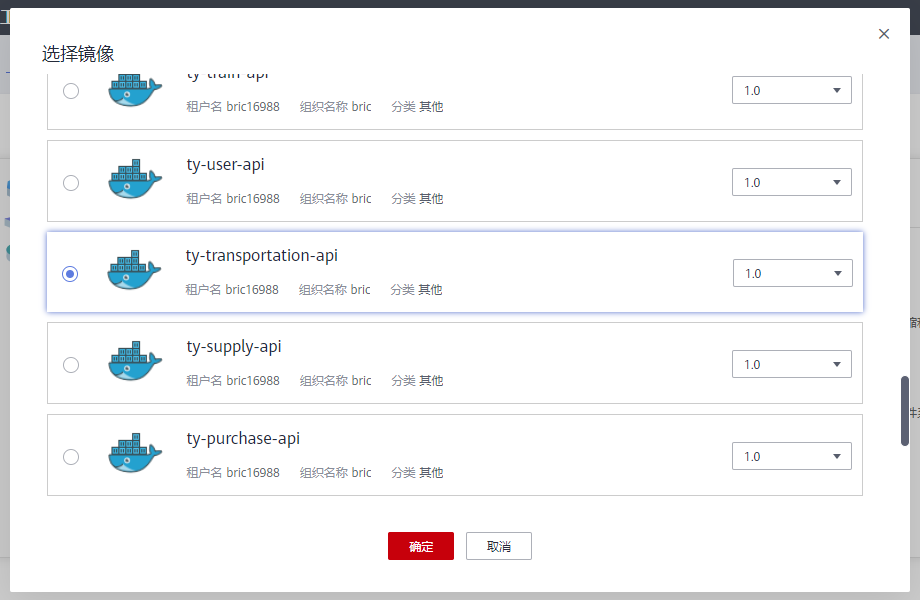
5.5、打包成镜像

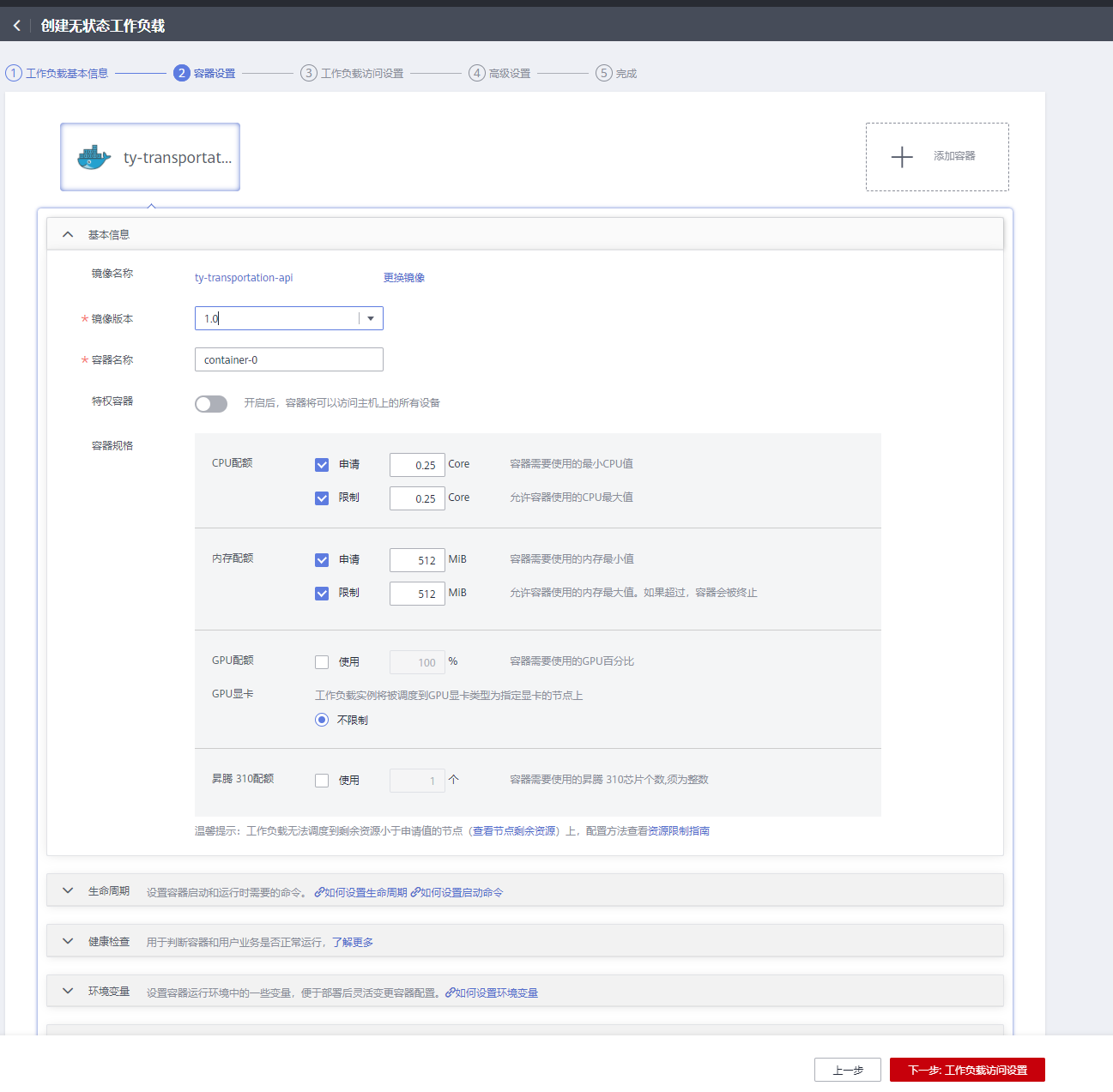
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-transportation-api .

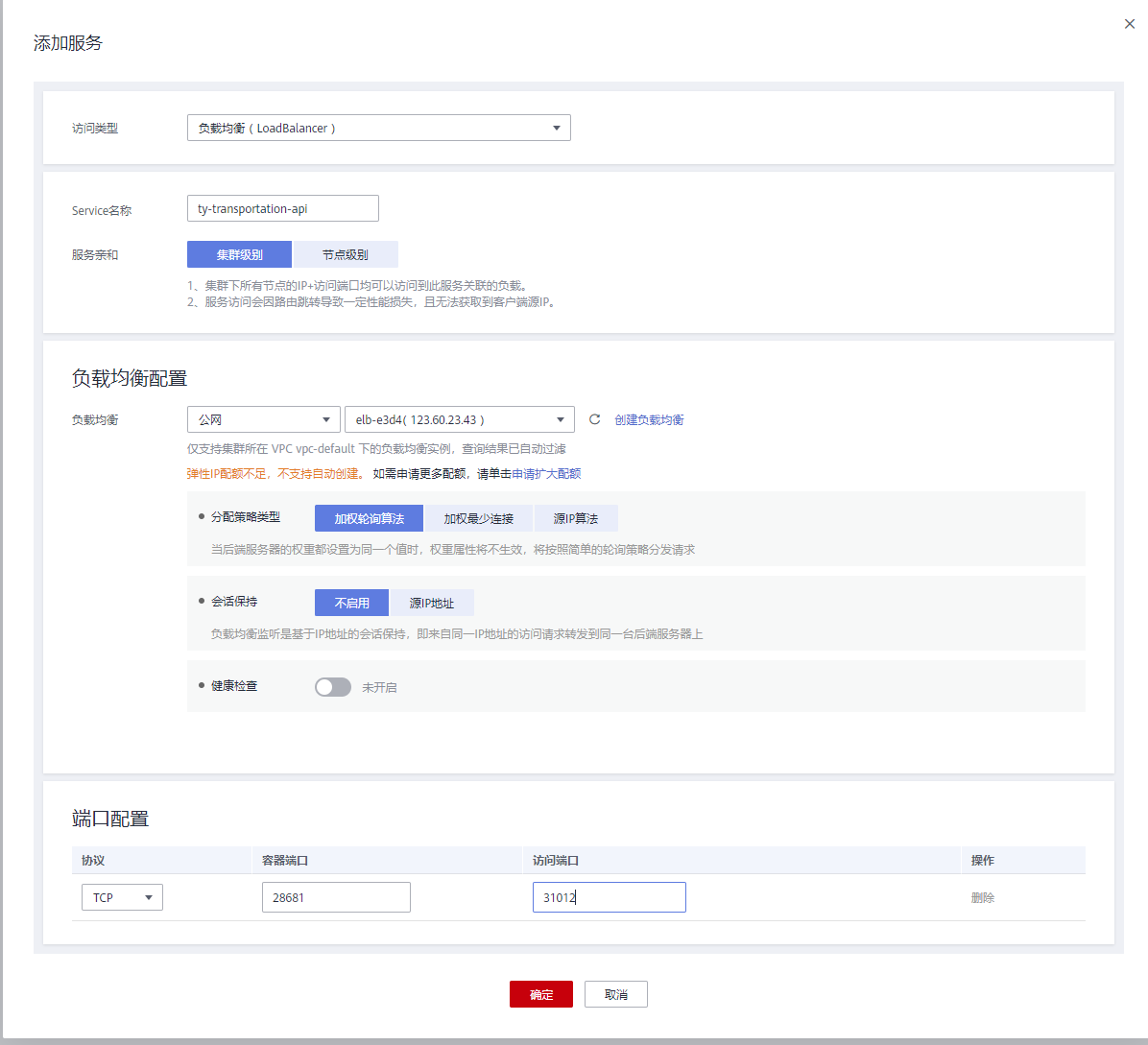
5.6、上传到华为镜像仓库

docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-transportation-api

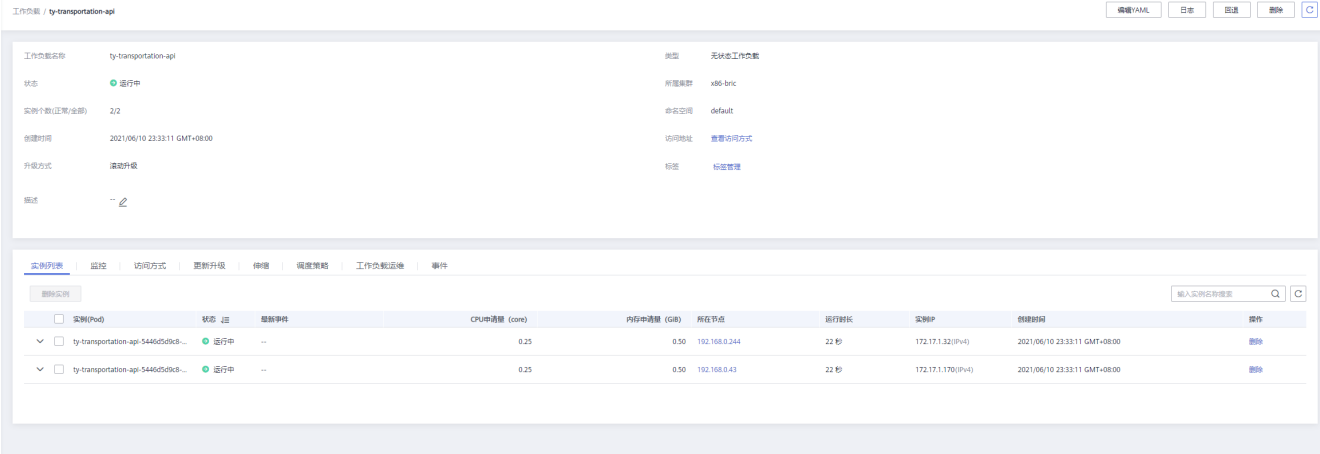
5.7、CCE创建无状态工作负载及访问











6、ty-train-api 安装

6.6、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

6.6、克隆ty-train-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

6.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.6.3/bin/mvn package

6.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:6.8

COPY ty-train-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-train-api.jar","--spring.profiles.avtive=dev"]

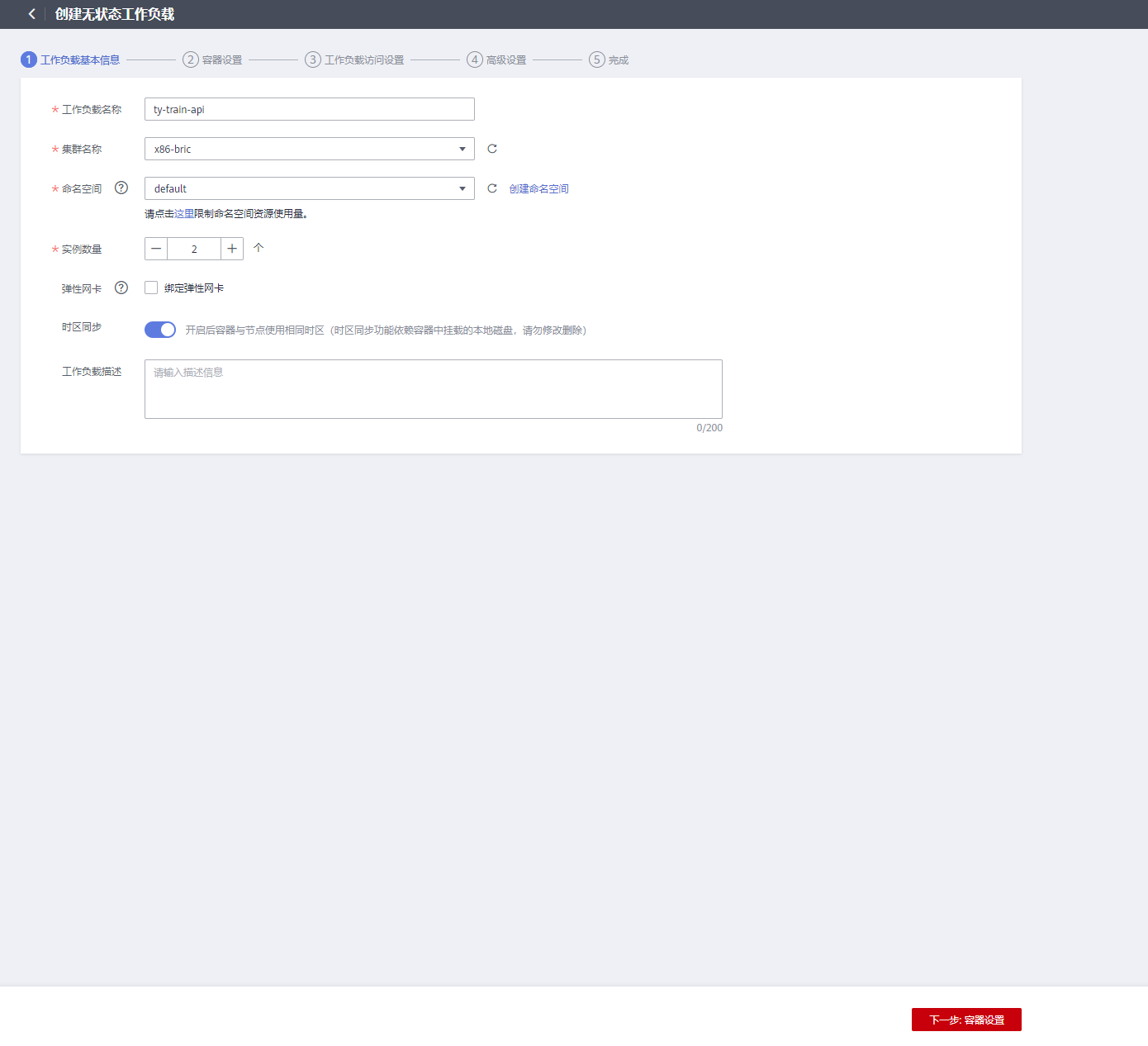
6.6、打包成镜像

docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-train-api .

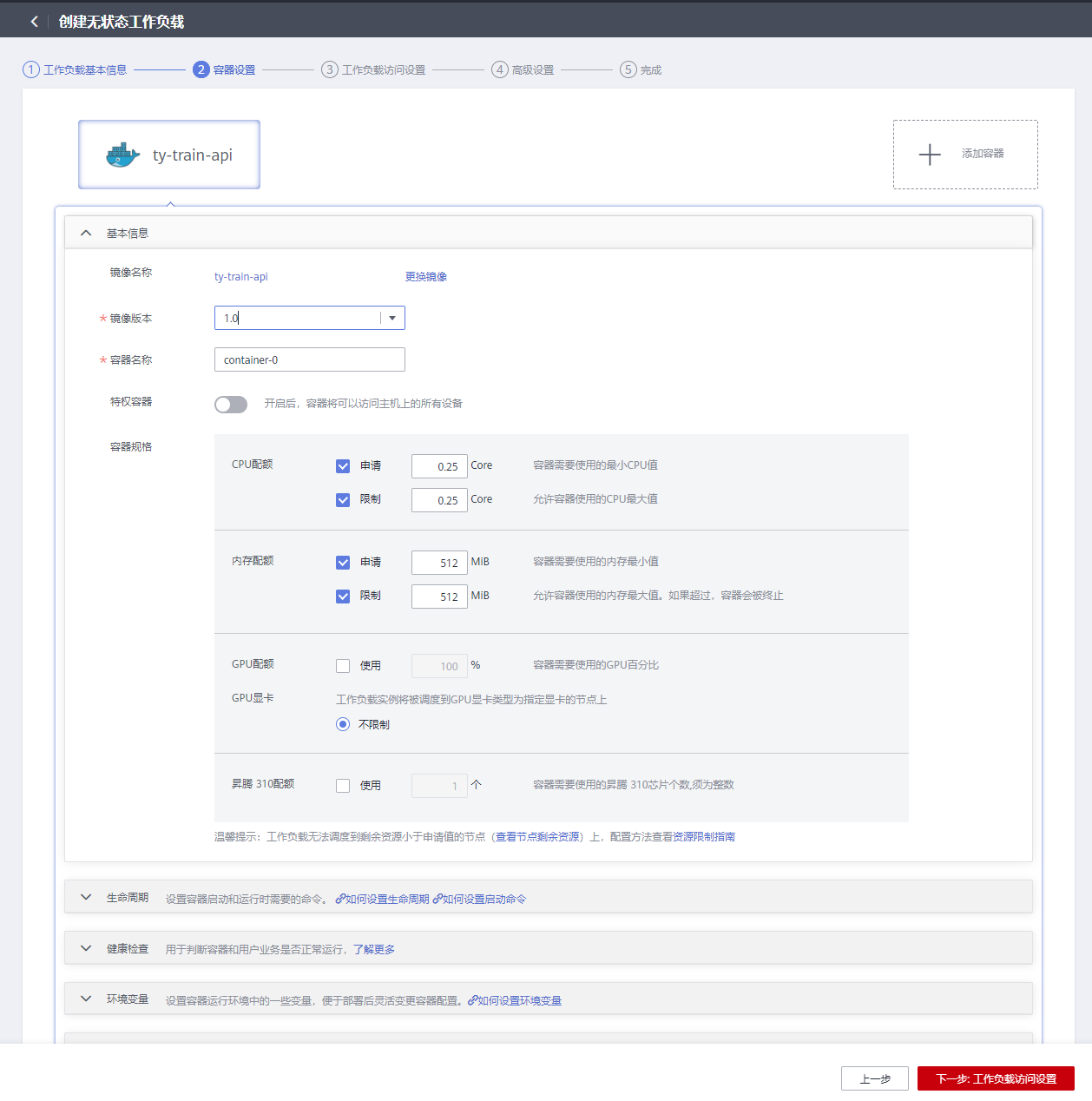
6.6、上传到华为镜像仓库

docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-train-api

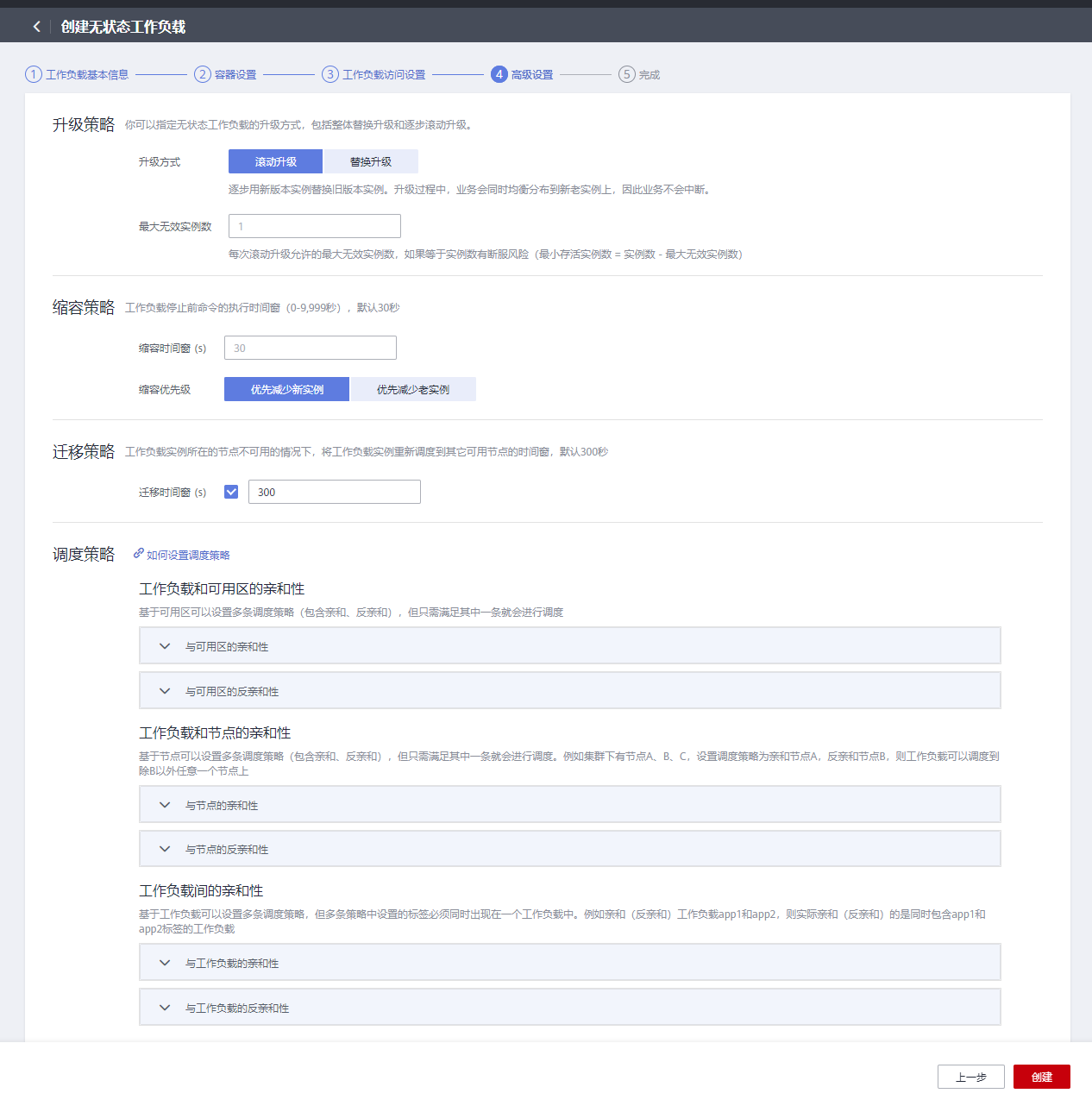
6.7、CCE创建无状态工作负载及访问

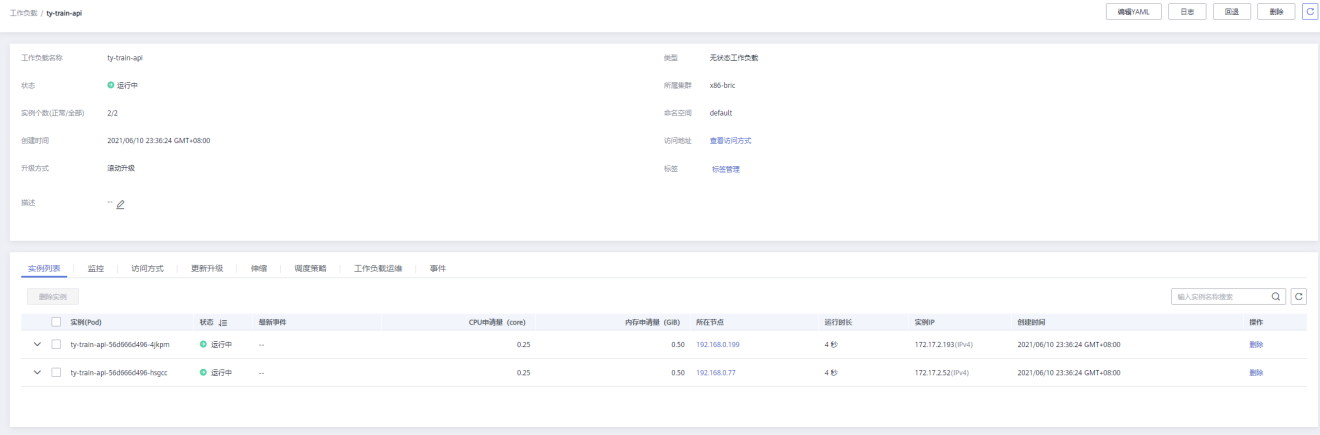












7、ty-supply-api 安装

7.7、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

7.7、克隆ty-supply-api源码

git clone ?b dev http://git.bric.com/java/sugar-guangxi.git

7.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.7.3/bin/mvn package

7.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/java:7.8

COPY ty-supply-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-supply-api.jar","--spring.profiles.avtive=dev"]

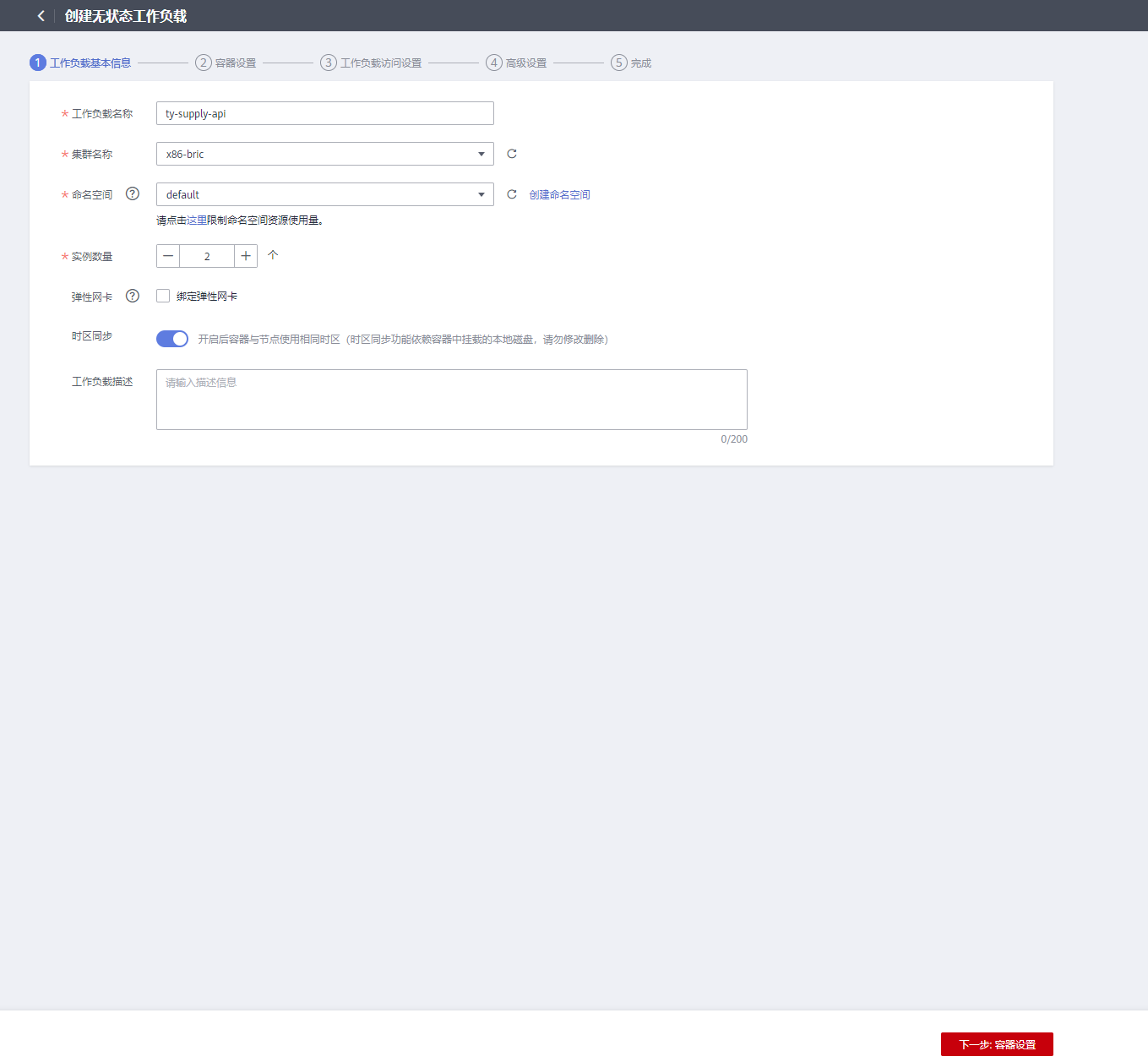
7.7、打包成镜像

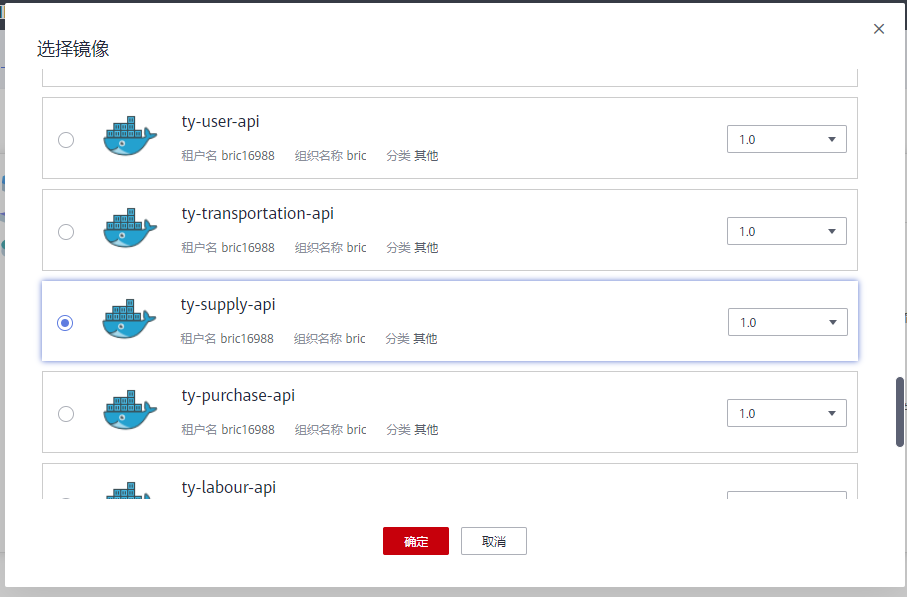
docker build ?t swr.cn-east-3.myhuaweicloud.com/bric/ty-supply-api .

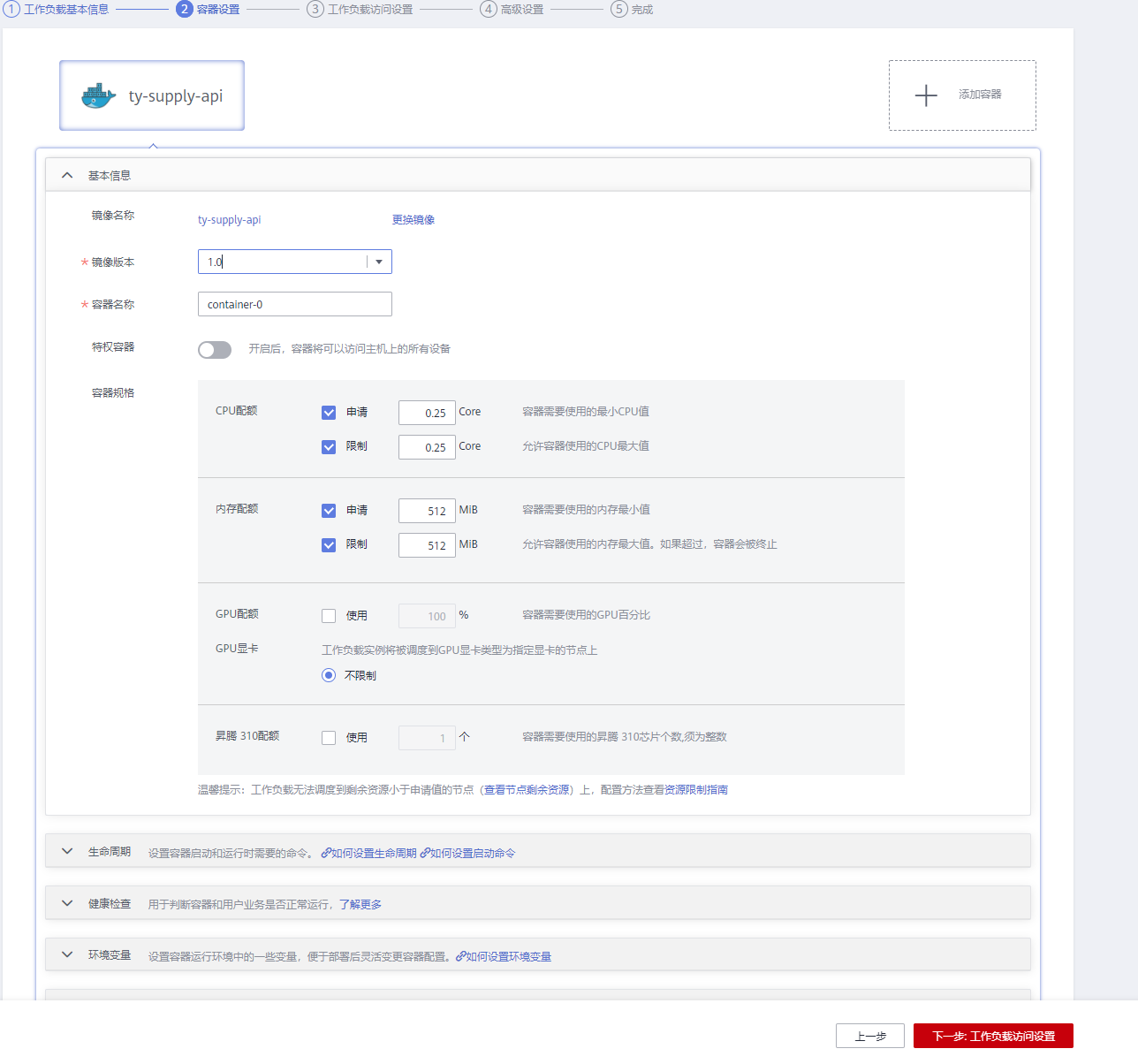
7.7、上传到华为镜像仓库

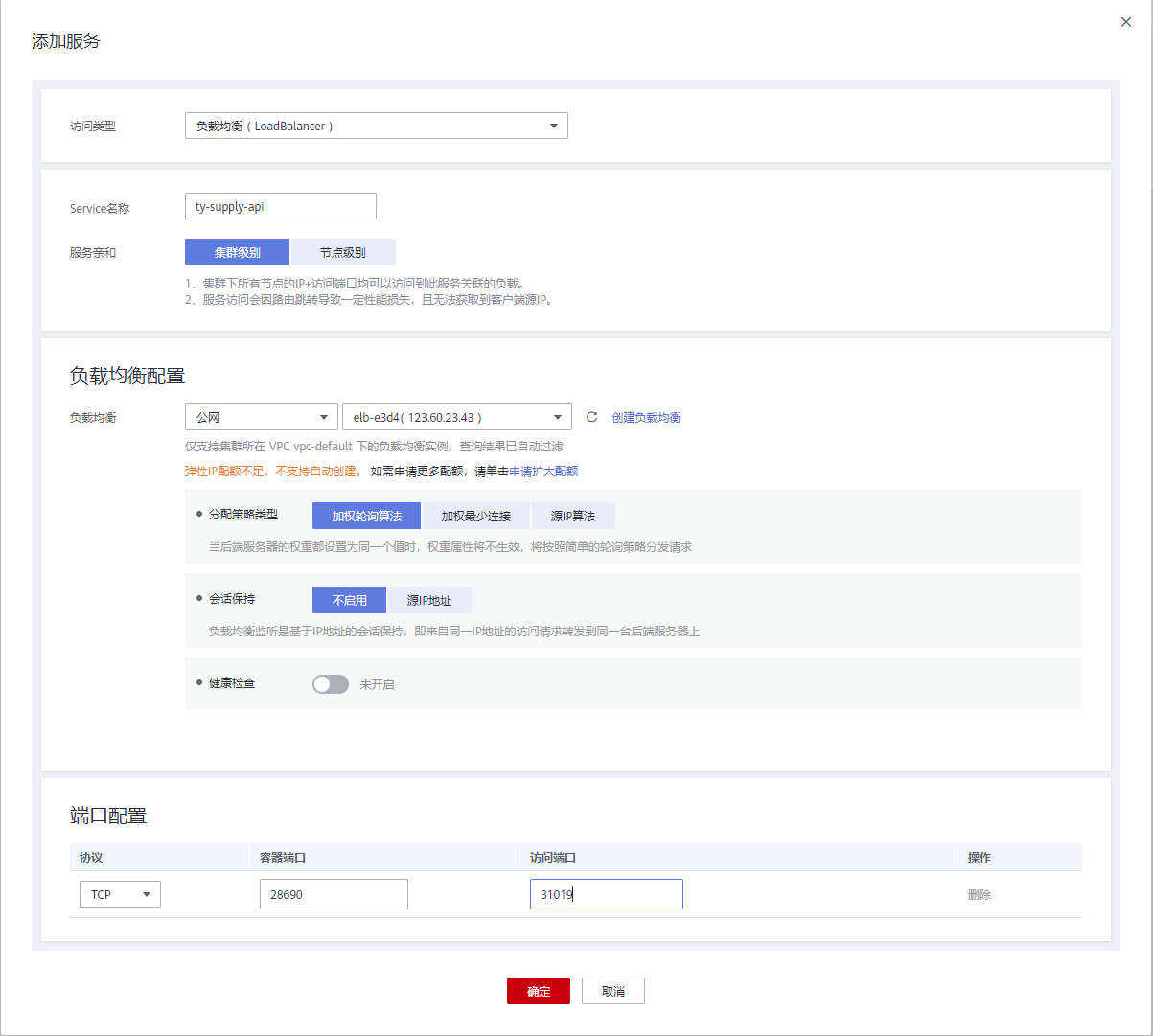
docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-supply-api

7.7、CCE创建无状态工作负载及访问

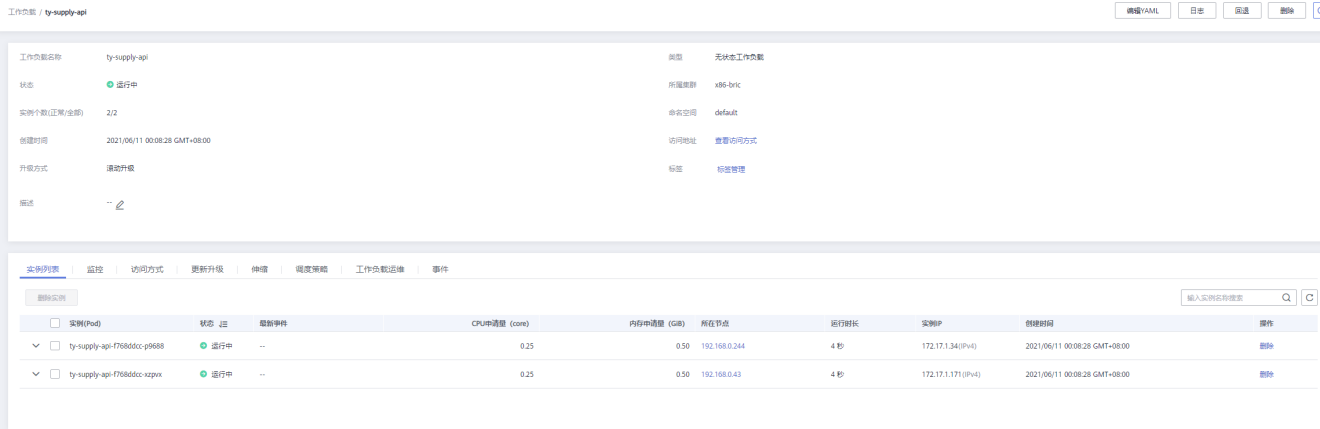












8、ty-purchase-api 安装

8.8、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

8.8、克隆ty-purchase-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

8.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.8.3/bin/mvn package

8.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos8\_linux/java:8.8

COPY ty-purchase-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+8","/bric/ty-purchase-api.jar","--spring.profiles.avtive=dev"]

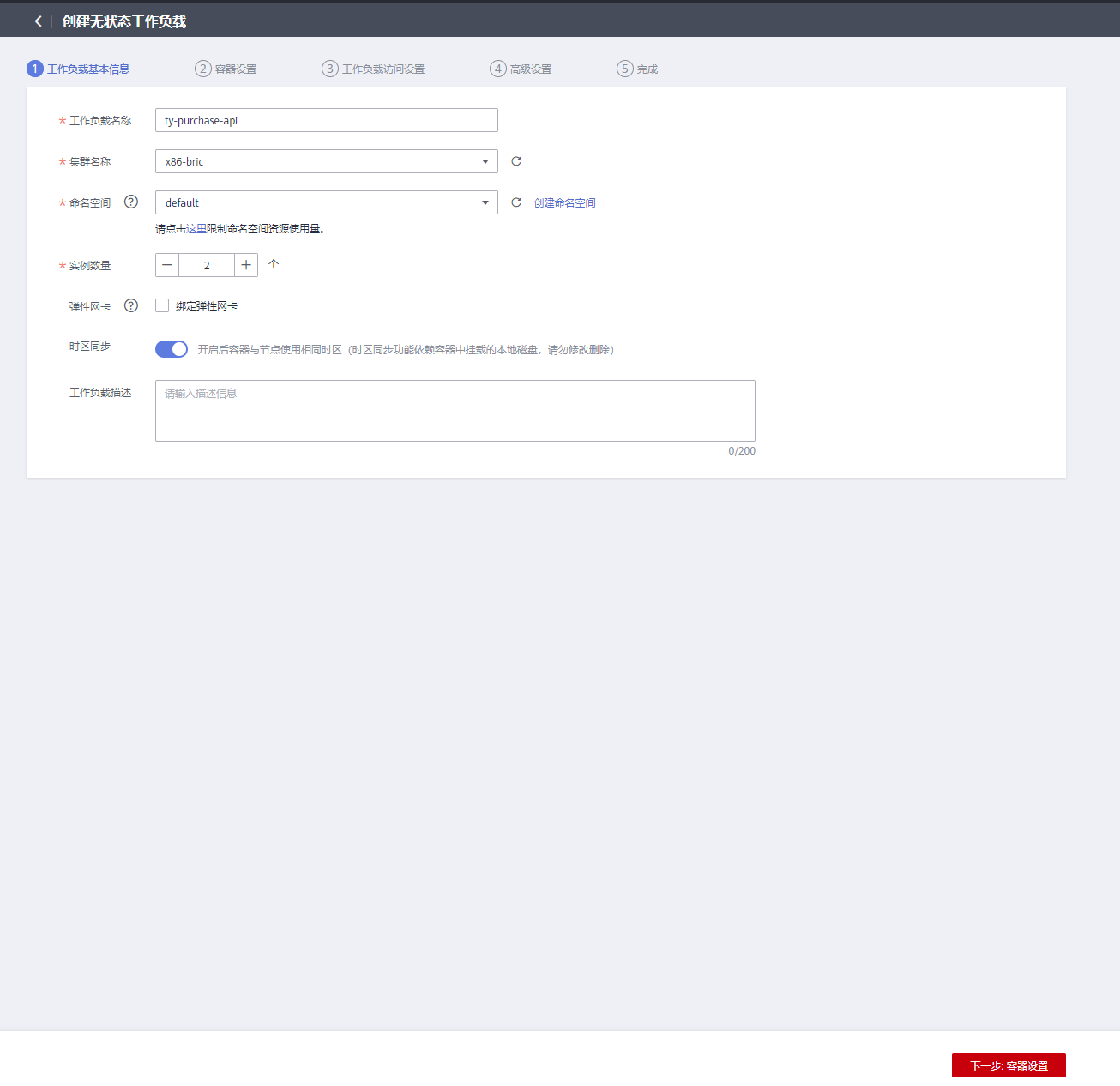
8.8、打包成镜像

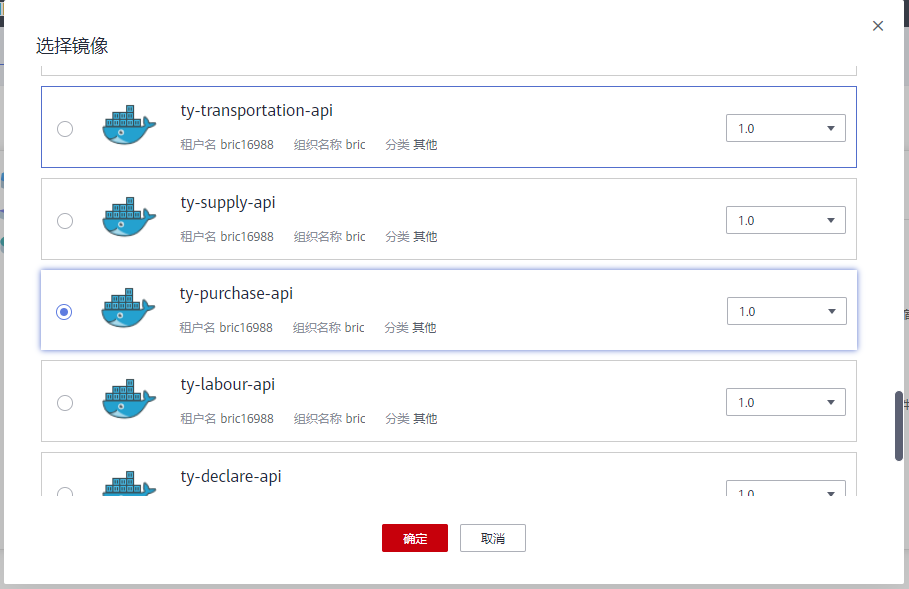
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-purchase-api .

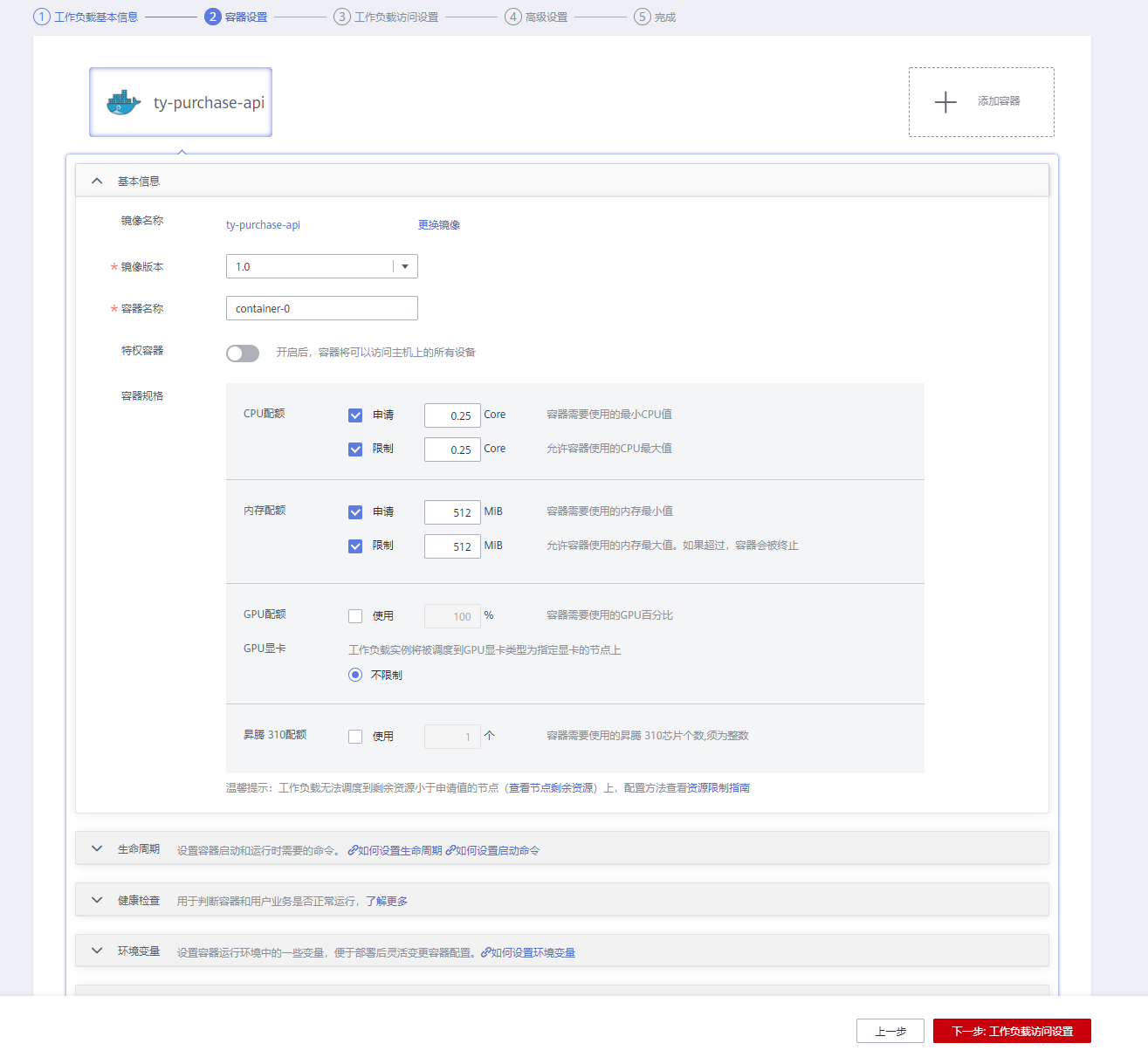
8.8、上传到华为镜像仓库

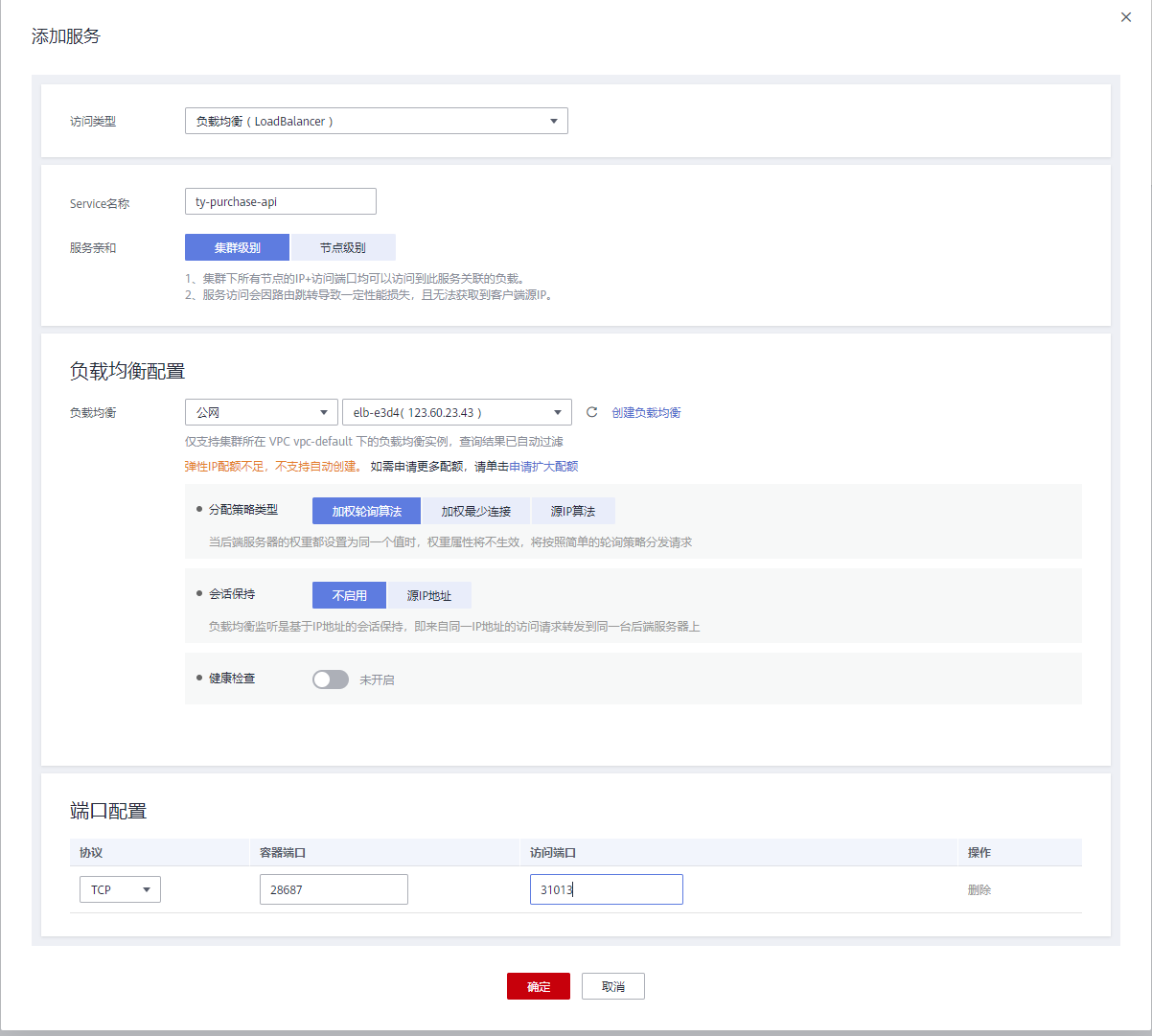
docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-purchase-api

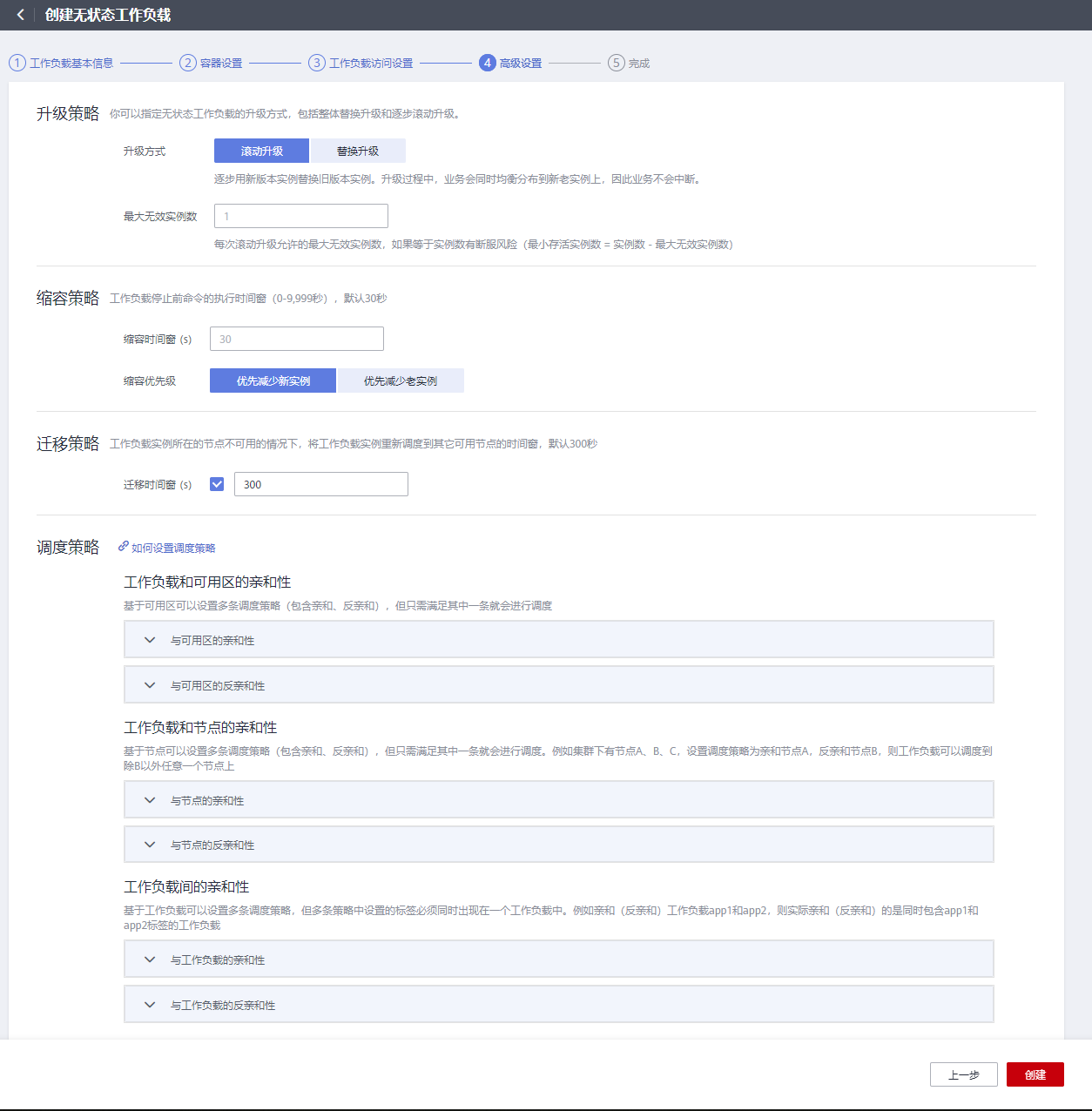
8.8、CCE创建无状态工作负载及访问

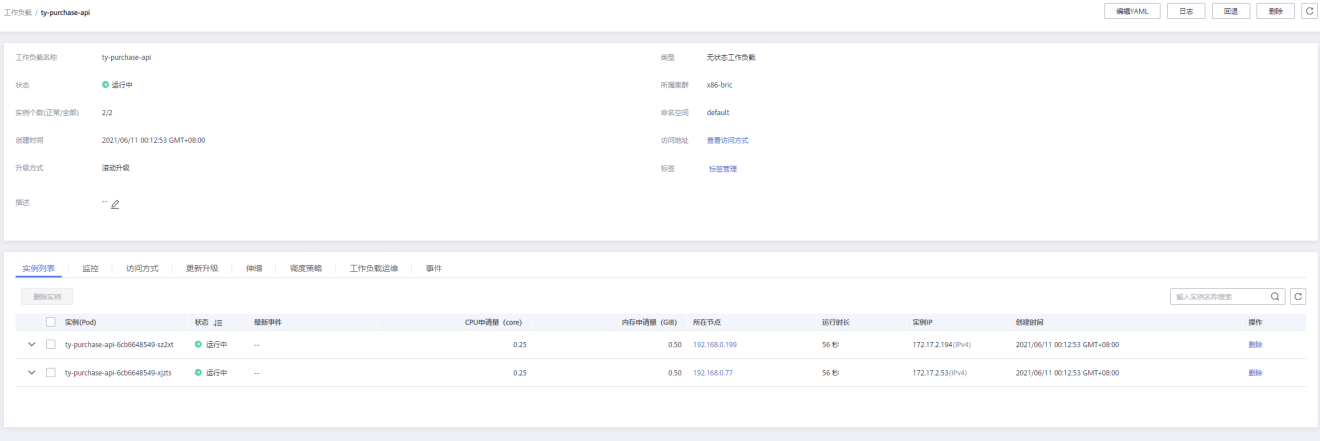












9、ty-labour-api 安装

9.9、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

9.9、克隆ty-labour-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

9.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.9.3/bin/mvn package

9.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos9\_linux/java:9.9

COPY ty-labour-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+9","/bric/ty-labour-api.jar","--spring.profiles.avtive=dev"]

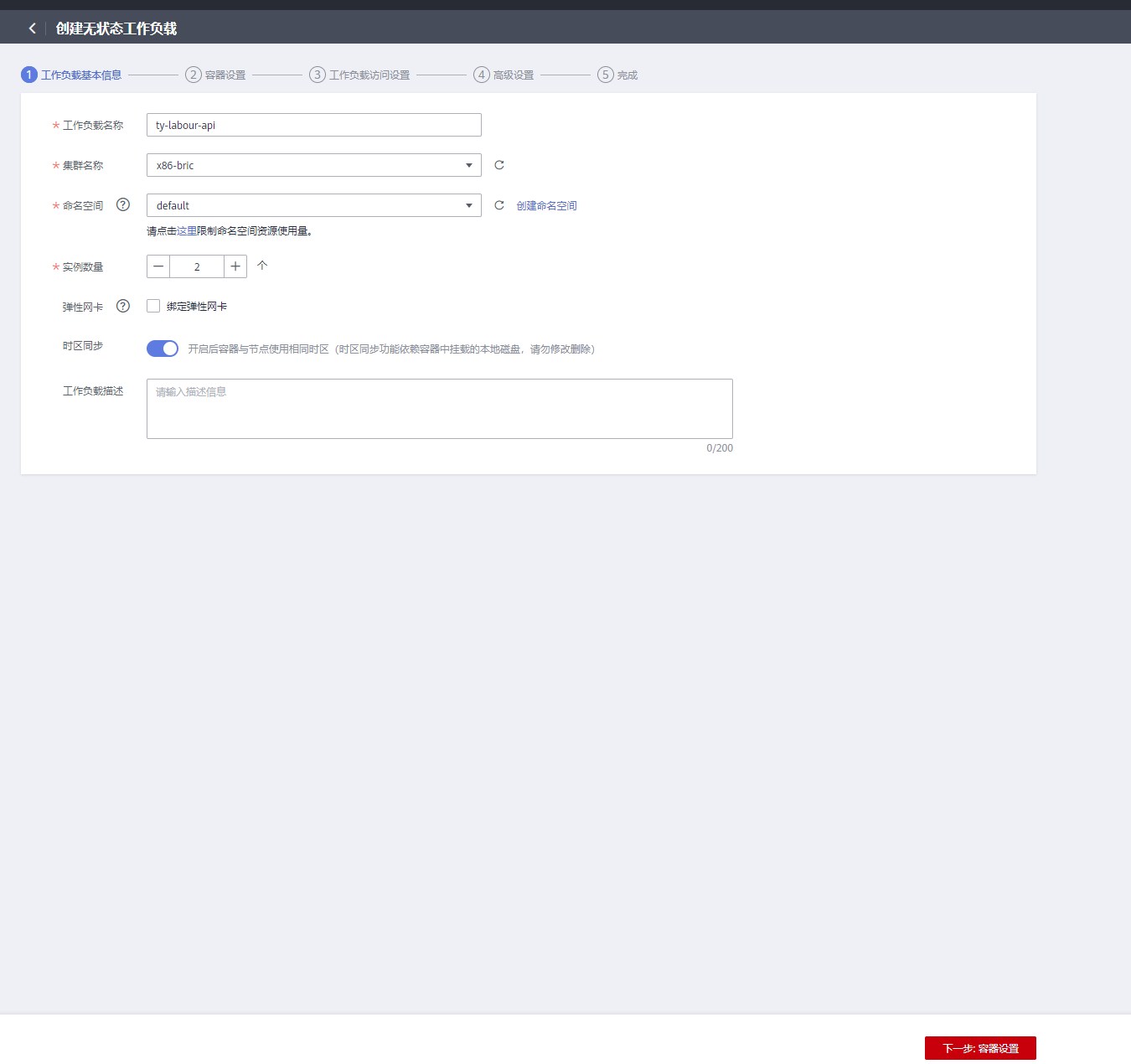
9.9、打包成镜像

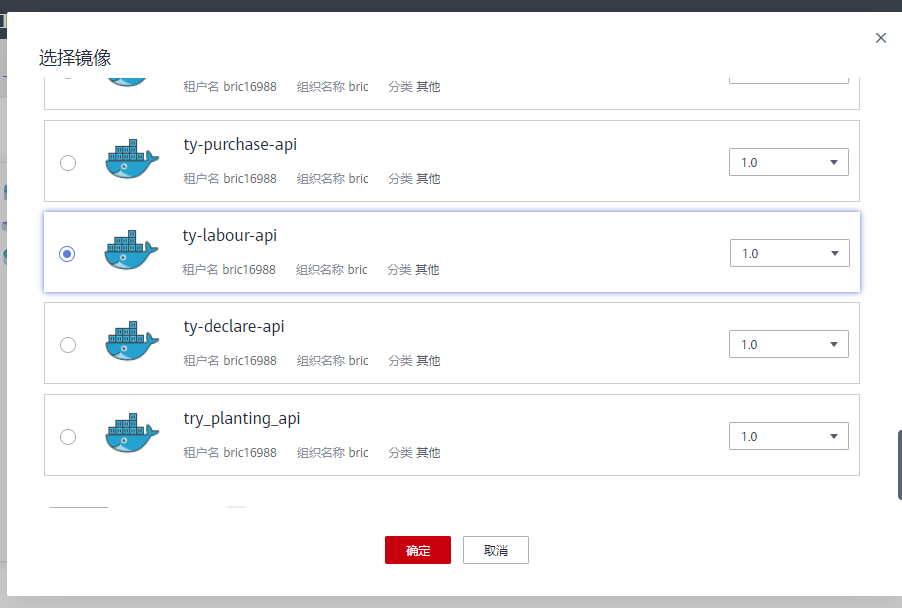
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-labour-api .

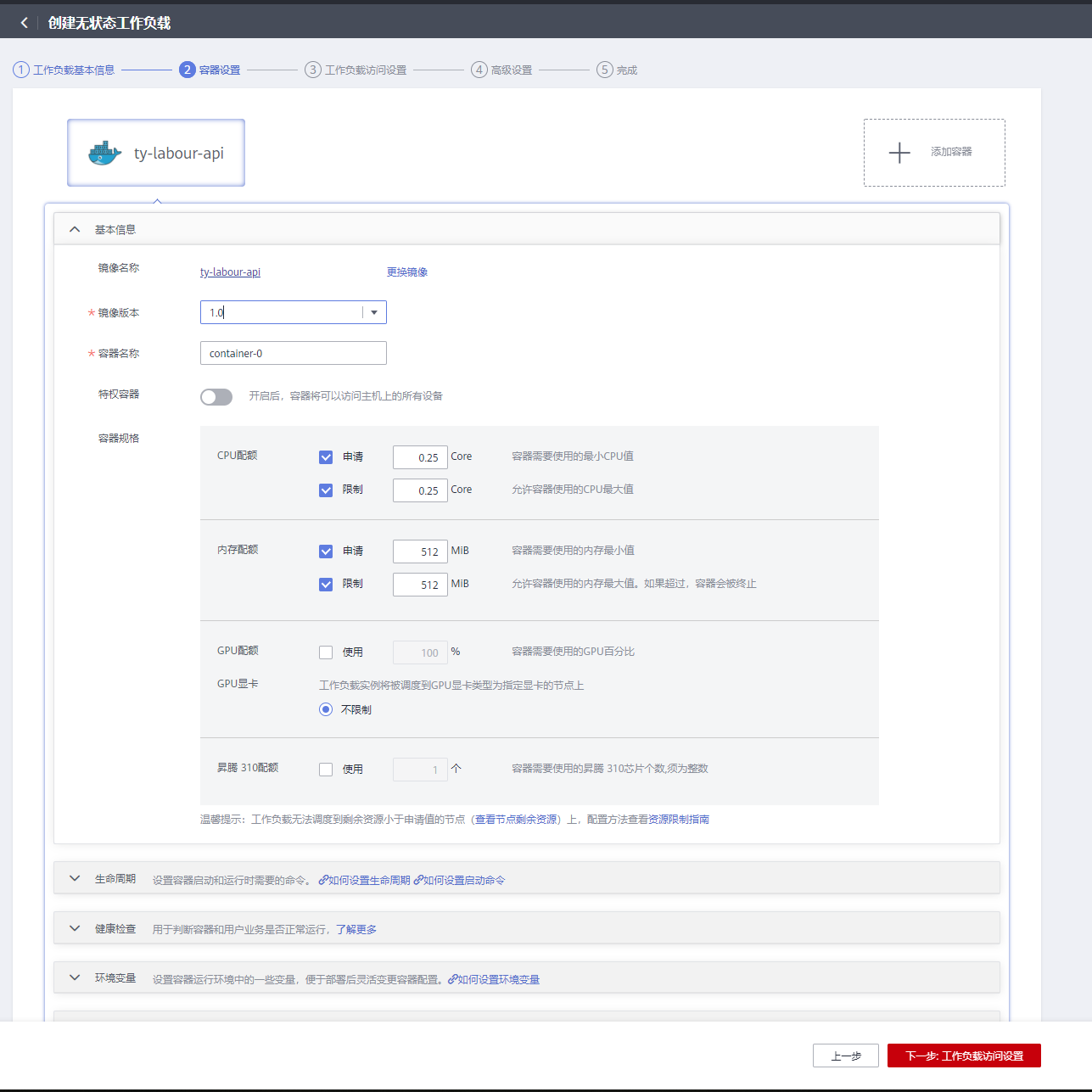
9.9、上传到华为镜像仓库

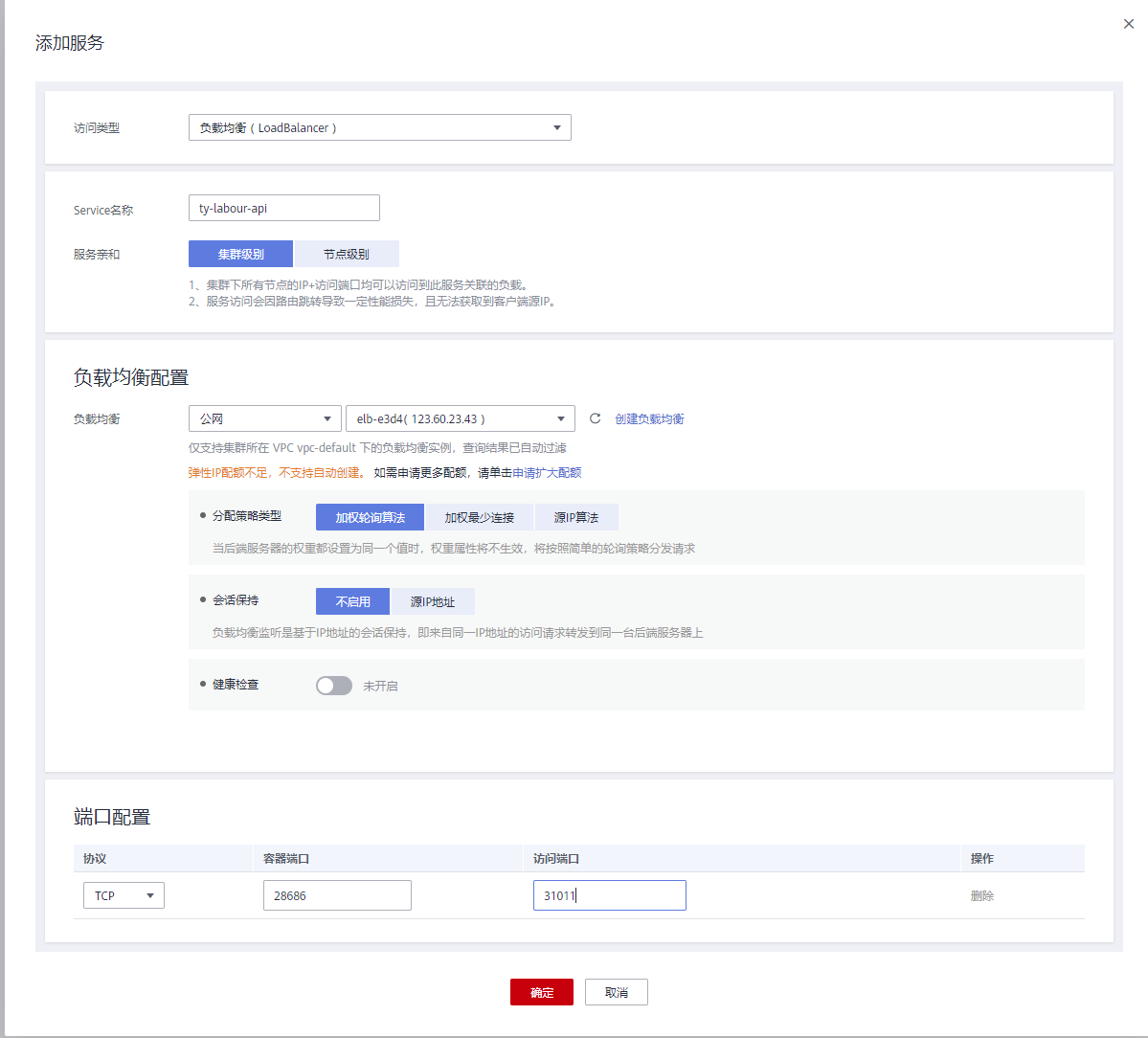
docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-labour-api

9.9、CCE创建无状态工作负载及访问

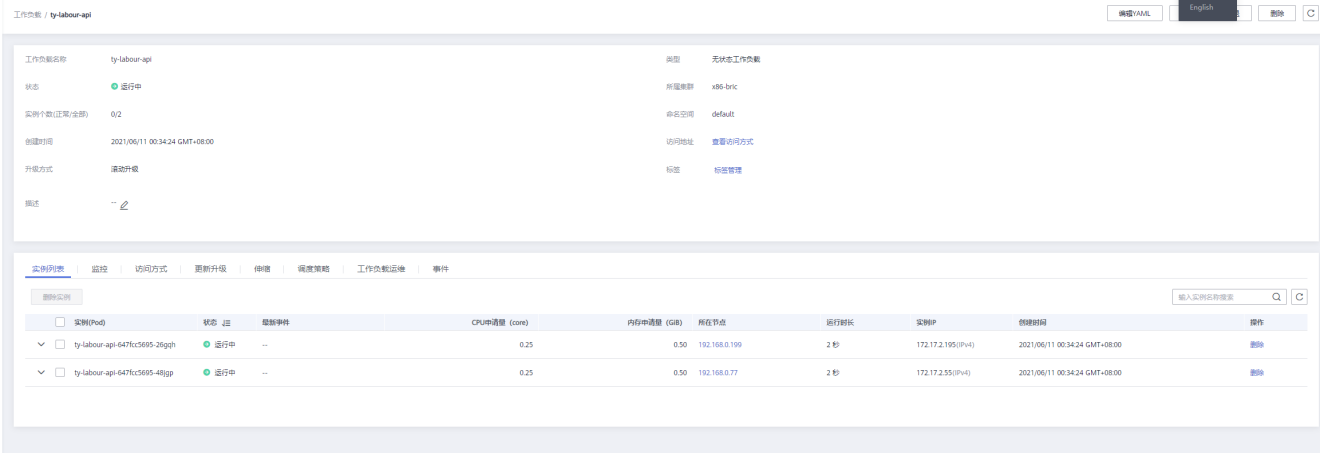












10、ty-declare-api 安装

10.10、在/tmp/目录中新建huawei目录

mkdir /tmp/huawei

10.10、克隆ty-declare-api源码

git clone –b dev http://git.bric.com/java/sugar-guangxi.git

10.3、检查配置文件无误后使用maven打包

/mnt/sda/usr/local/apache-maven-3.10.3/bin/mvn package

10.4、进入到包路径打包成镜像

cd /tmp/Huawei/ty\_planting\_api/target

创建Dockerfile文件内容如下

FROM registry.cn-shanghai.aliyuncs.com/centos10\_linux/java:10.10

COPY ty-declare-api.jar /bric/

ENTRYPOINT ["java","-jar","-Duser.timezone=GMT+10","/bric/ty-declare-api.jar","--spring.profiles.avtive=dev"]

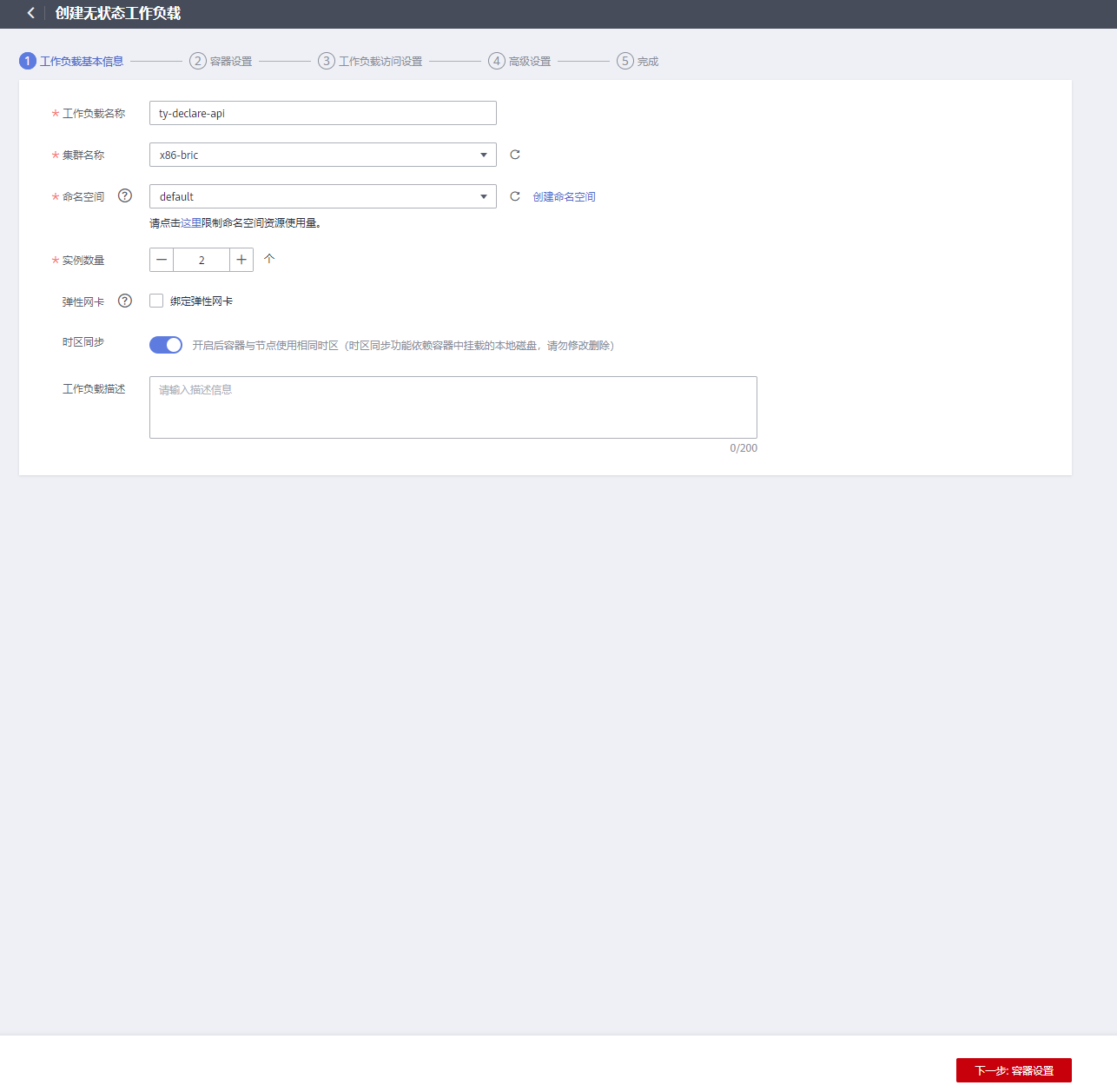
10.10、打包成镜像

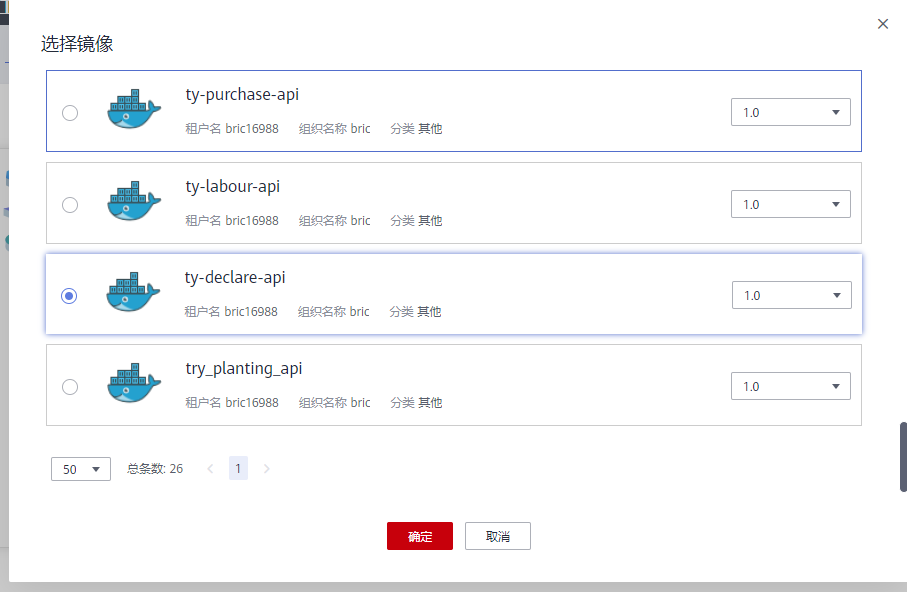
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-declare-api .

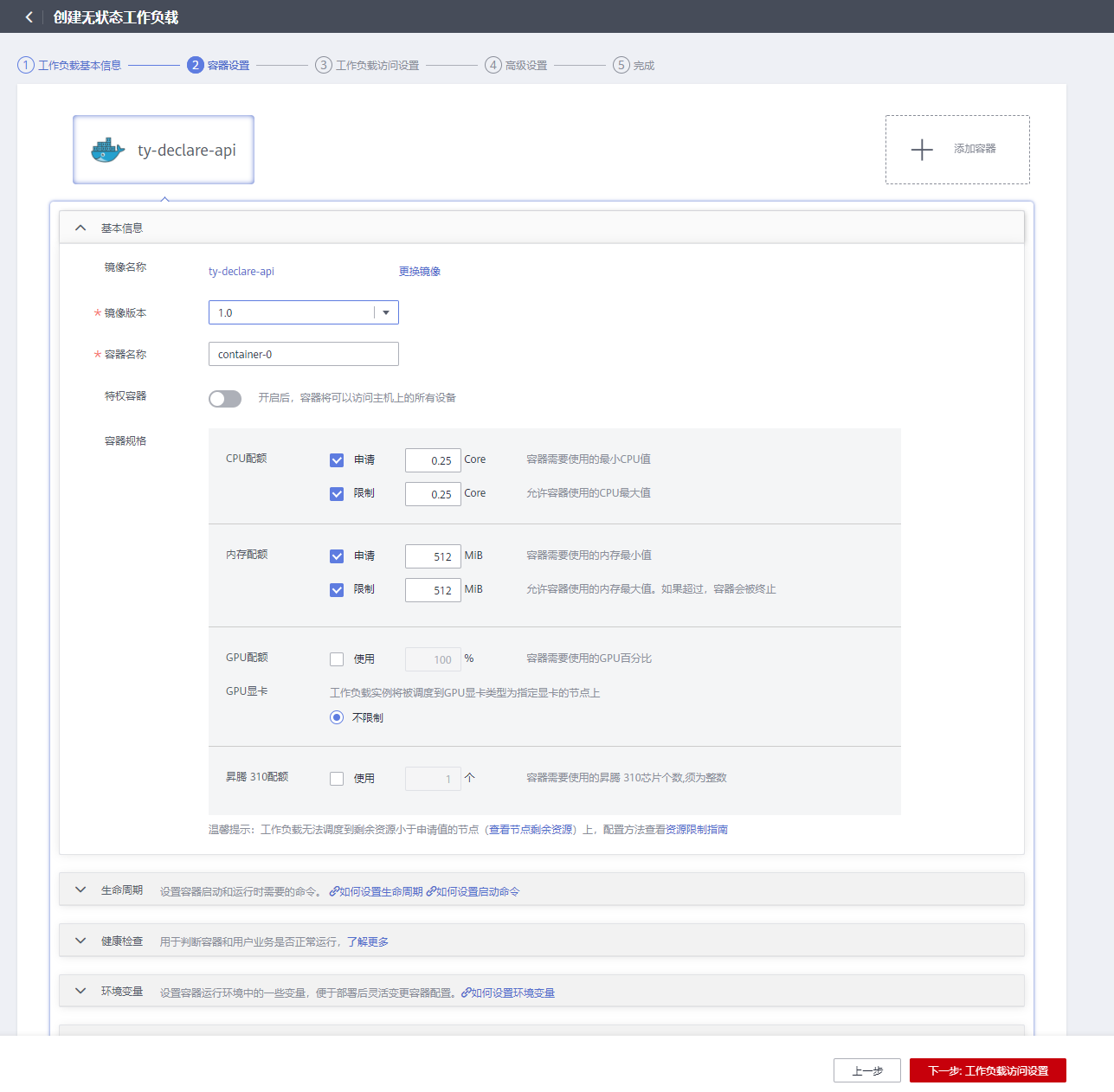
10.10、上传到华为镜像仓库

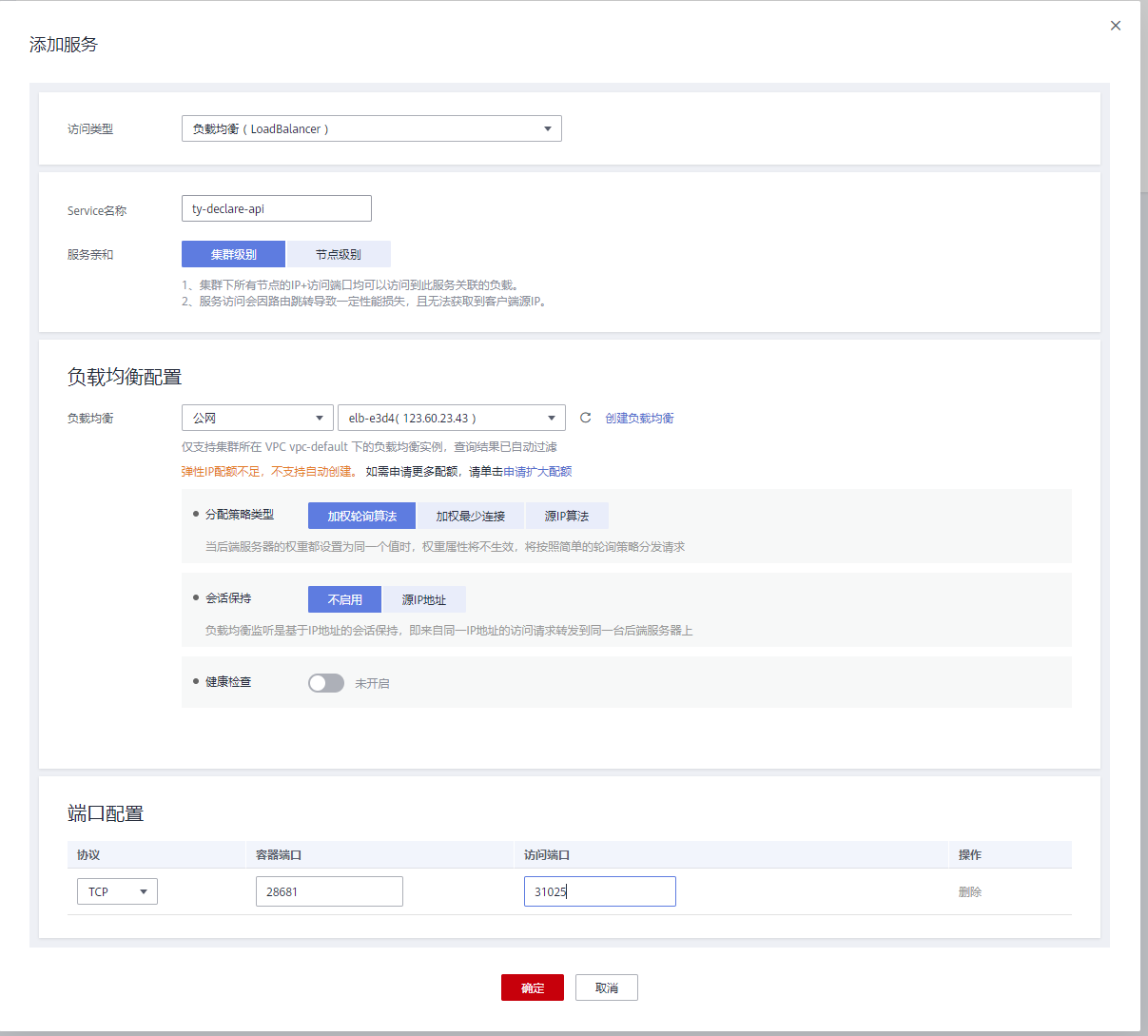
docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-declare-api

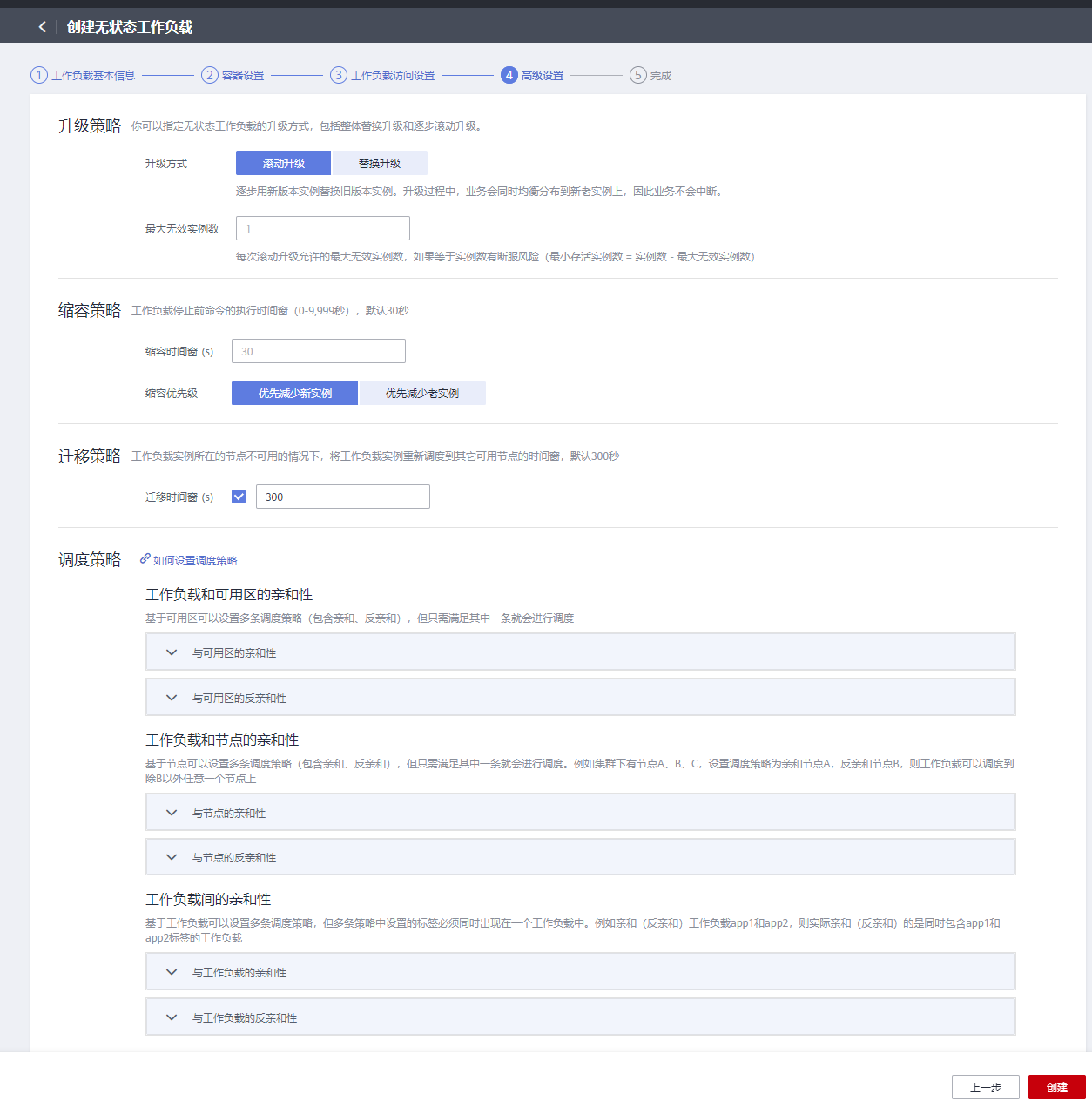
10.10、CCE创建无状态工作负载及访问

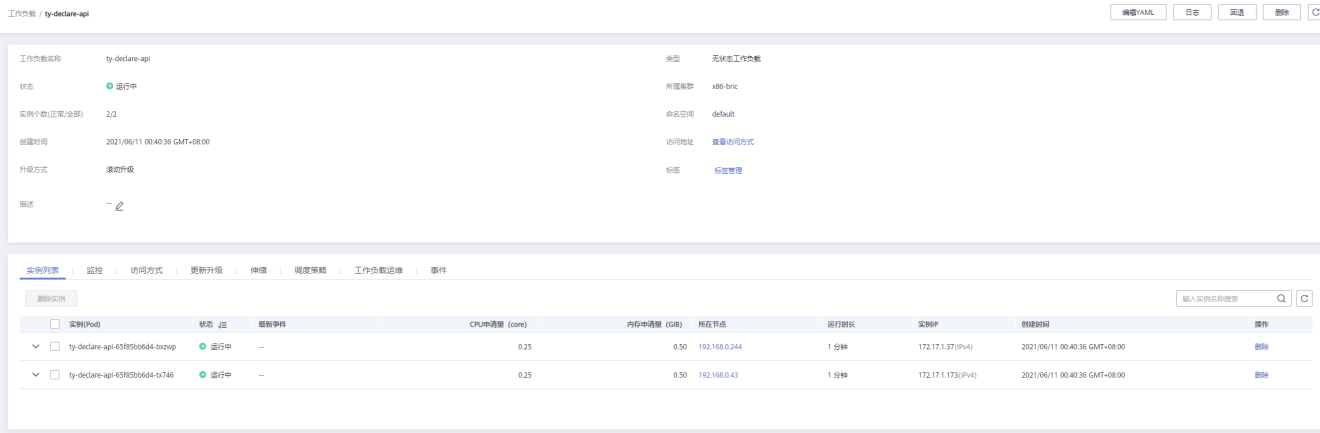












11.1、准备好ty-root-web镜像

11.2、制作Dockerfile

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/nginx:1.19

COPY nginx.conf /usr/local/nginx/conf/

ENTRYPOINT ["/usr/local/nginx/sbin/nginx","-g","daemon off;"]

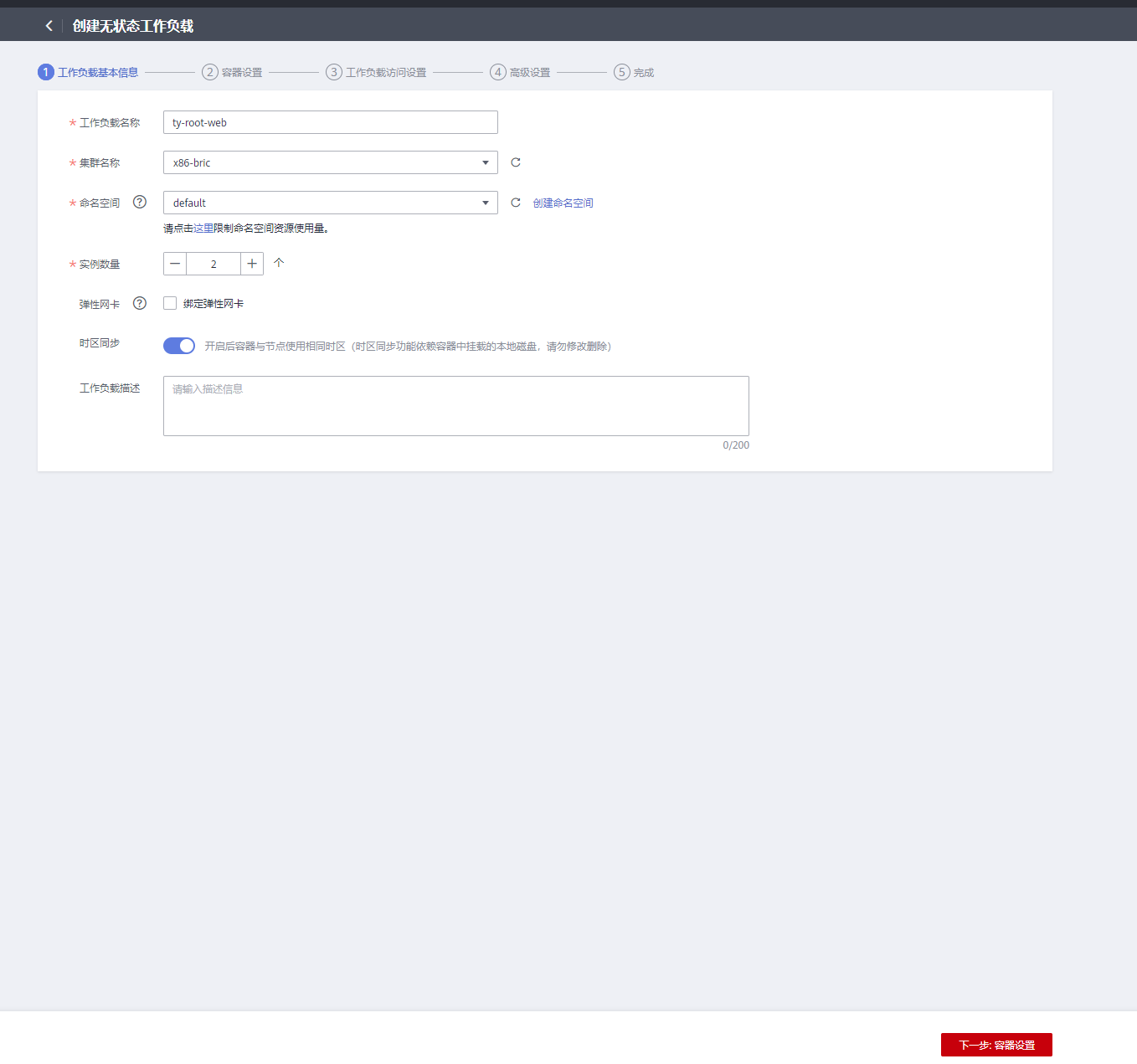
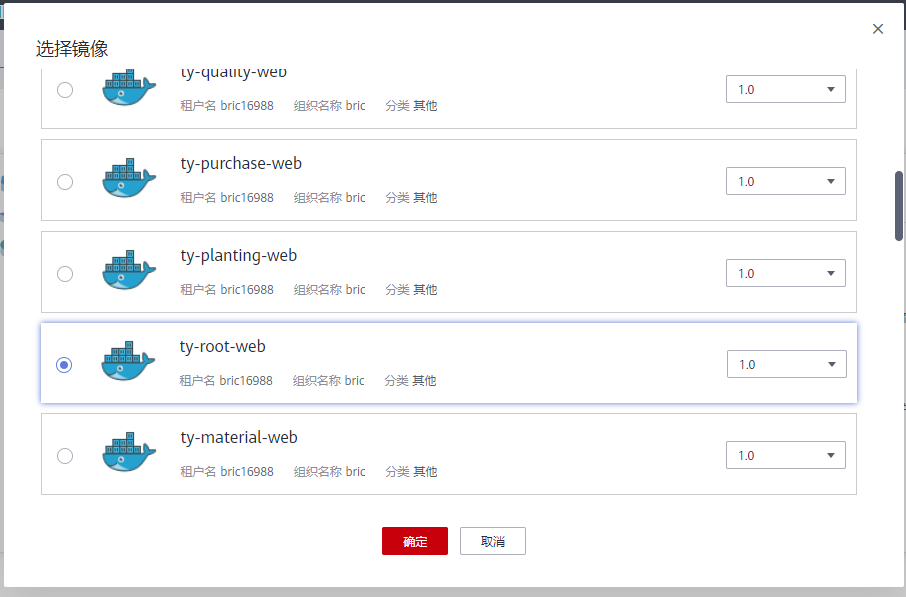
11.3、打包上传

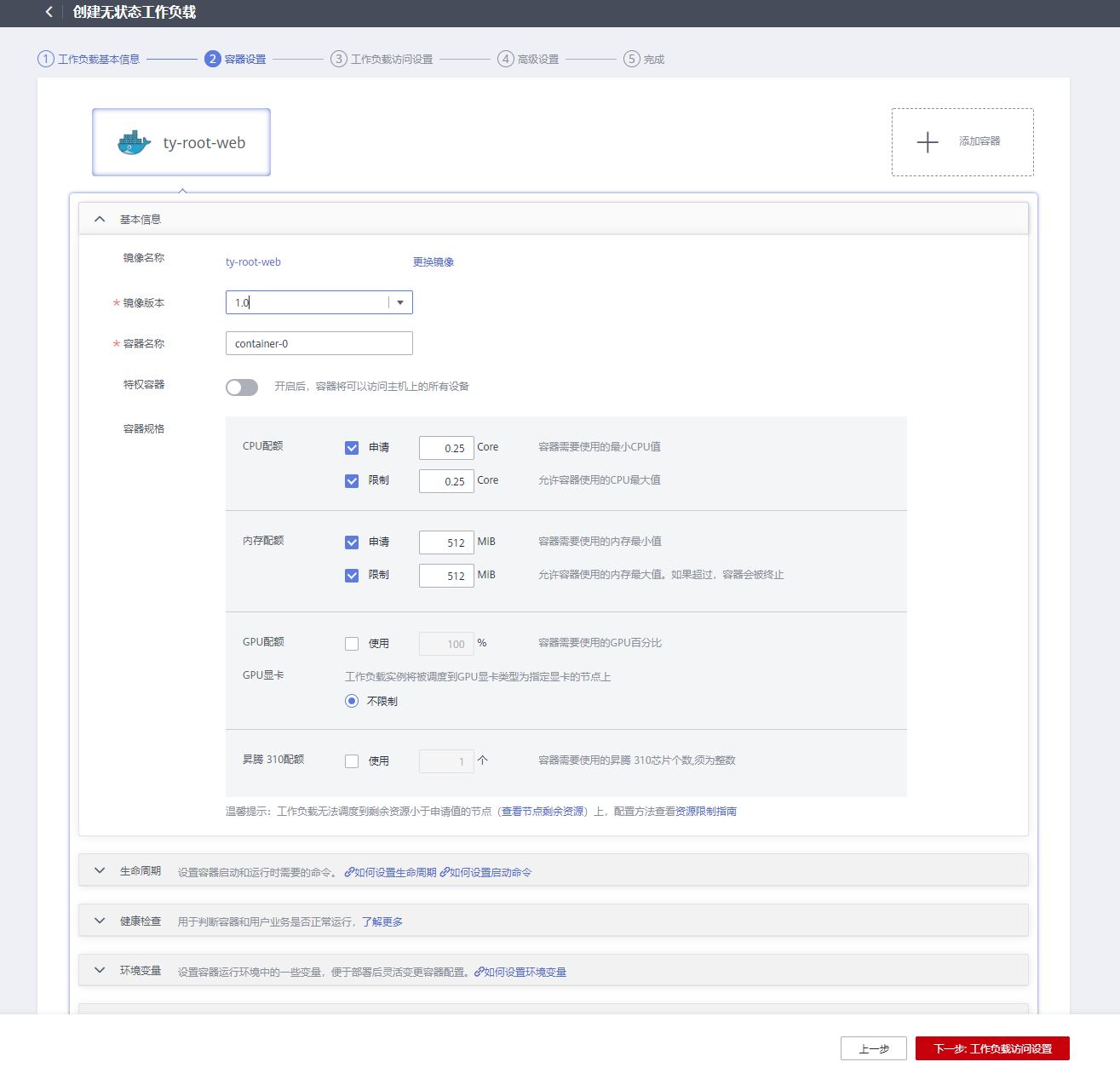
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-root-web .

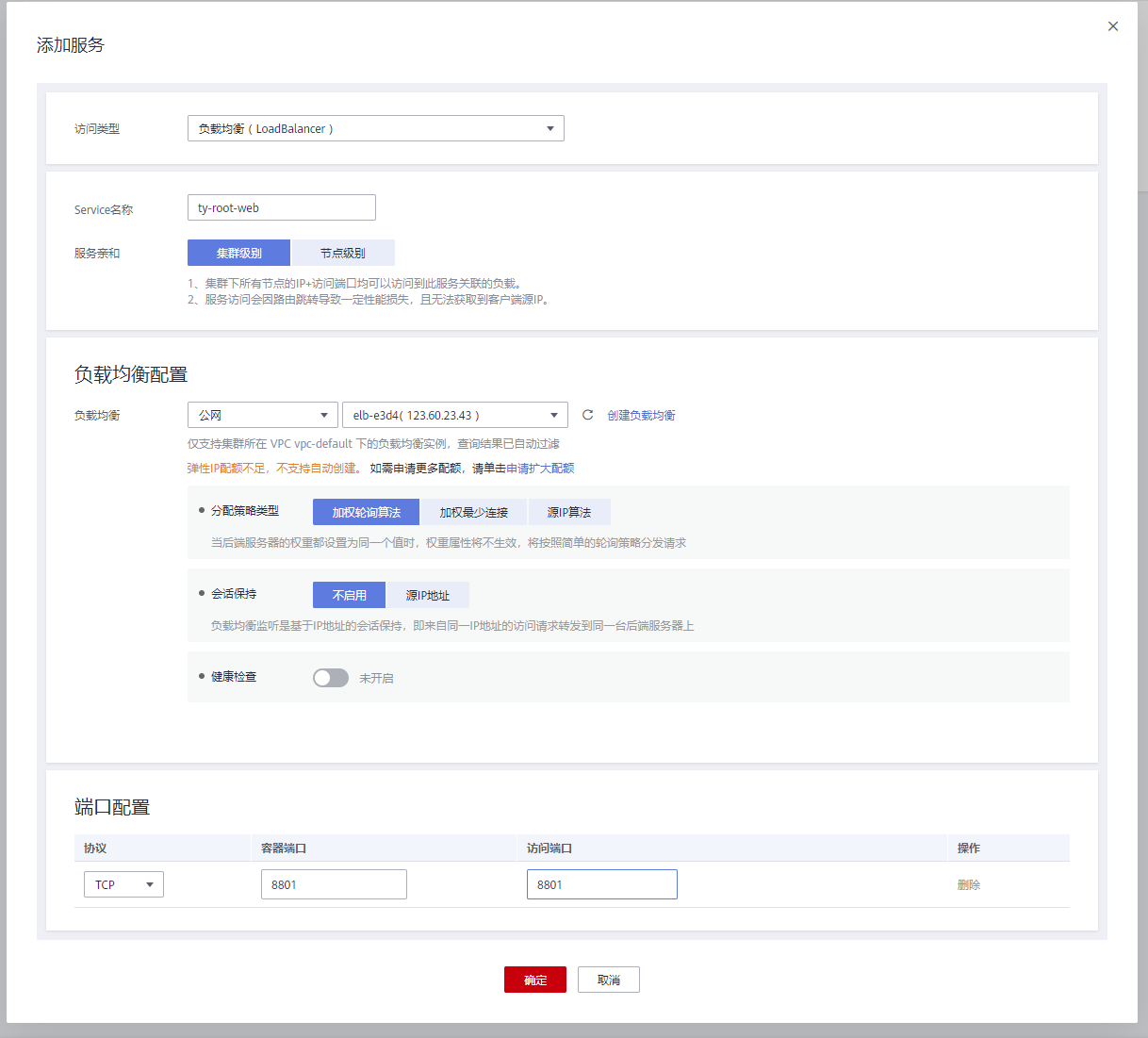
11.4、上传到华为镜像仓库

docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-root-web .

11.5、 CCE创建无状态工作负载及访问





12.1、准备好sugar-cloud镜像

12.2、制作Dockerfile

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/nginx:1.19

COPY nginx.conf /usr/local/nginx/conf/

ENTRYPOINT ["/usr/local/nginx/sbin/nginx","-g","daemon off;"]

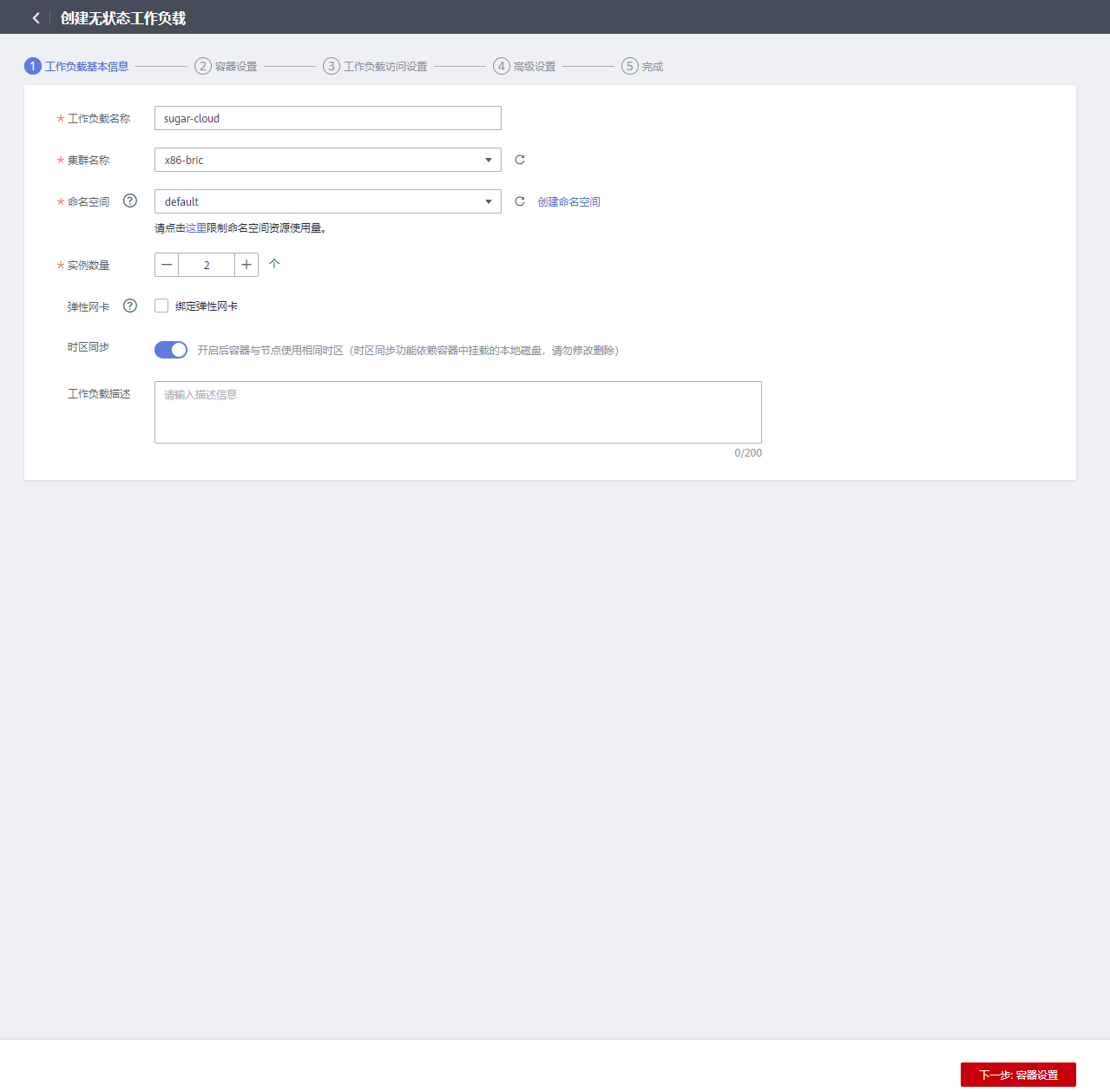
12.3、打包上传

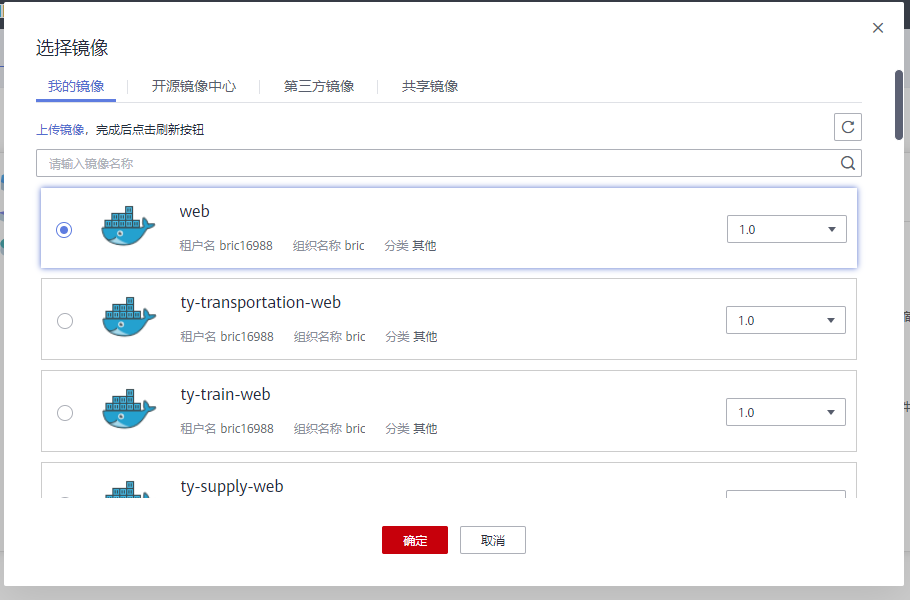
docker build –t swr.cn-east-3.myhuaweicloud.com/bric/sugar-cloud .

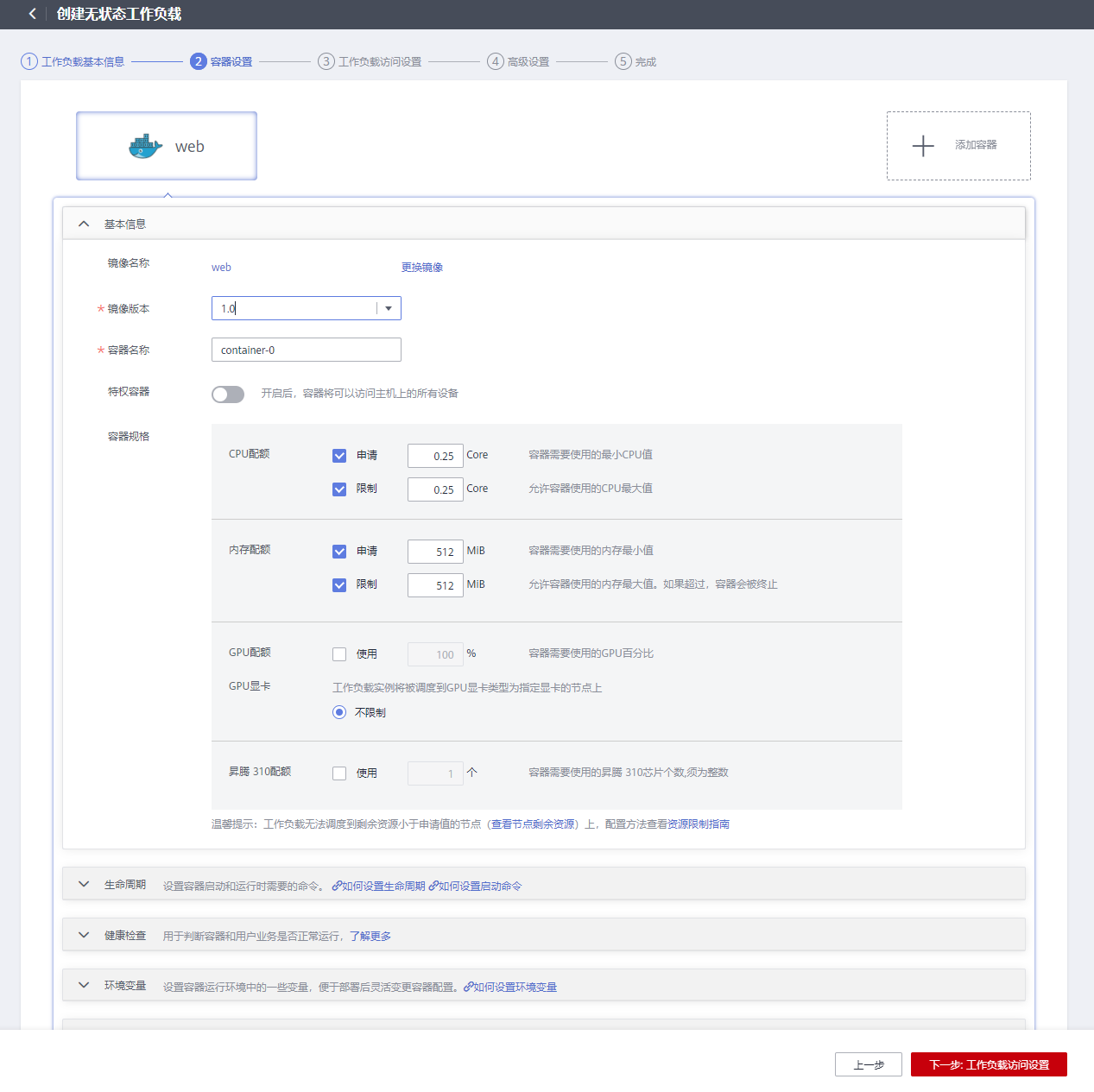
12.4、上传到华为镜像仓库

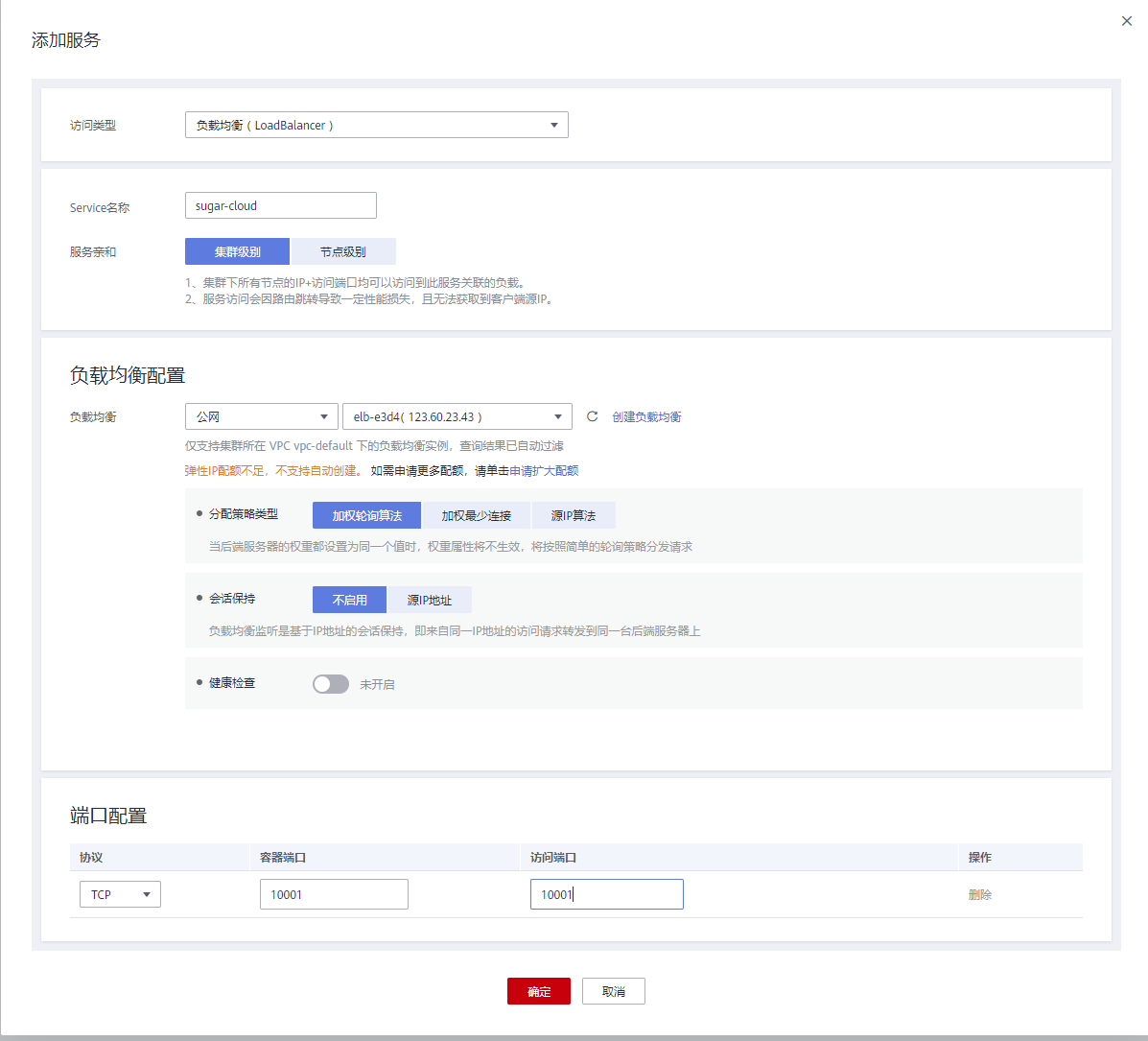
docker push swr.cn-east-3.myhuaweicloud.com/bric/sugar-cloud .

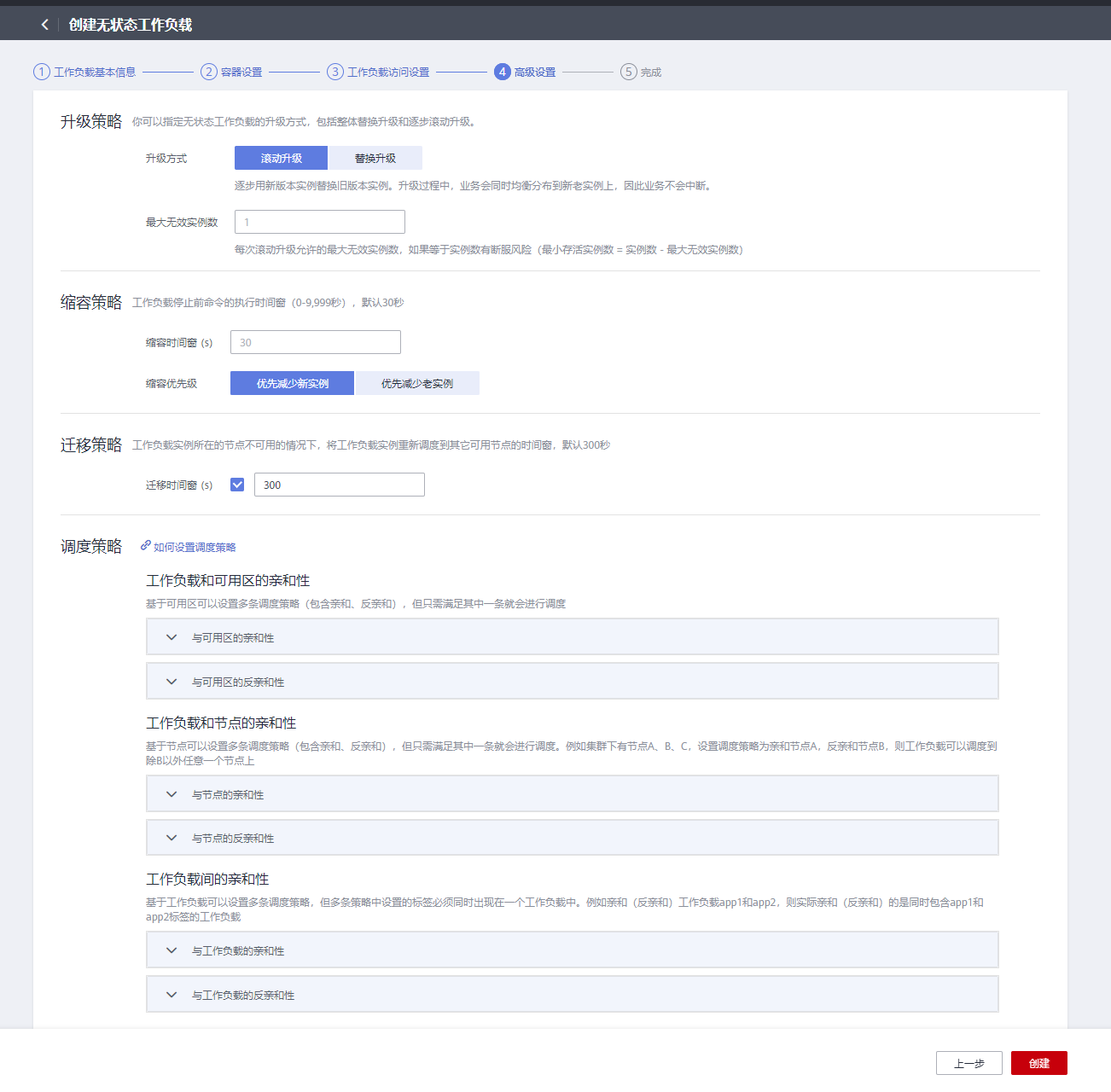
12.5、CCE创建无状态工作负载及访问











13.1、准备好ty-admin-web镜像

13.2、制作Dockerfile

FROM registry.cn-shanghai.aliyuncs.com/centos7\_linux/nginx:1.19

COPY nginx.conf /usr/local/nginx/conf/

ENTRYPOINT ["/usr/local/nginx/sbin/nginx","-g","daemon off;"]

13.3、打包上传

docker build –t swr.cn-east-3.myhuaweicloud.com/bric/ty-admin-web .

13.4、上传到华为镜像仓库

docker push swr.cn-east-3.myhuaweicloud.com/bric/ty-admin-web .

13.5、CCE创建无状态工作负载及访问

