

部署

最佳实践

文档版本 02
发布日期 2024-07-01



版权所有 © 华为云计算技术有限公司 2024。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云计算技术有限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，华为云计算技术有限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。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 通过代理主机在内网部署应用.....	1
2 基于 Nginx 实现应用的灰度发布.....	6
3 基于 Kubernetes Nginx-Ingress 实现应用的灰度发布.....	27
4 通过自托管资源池部署应用至云下 IDC.....	32
5 通过自托管资源池实现跨 Region 虚拟机部署.....	35
6 HE2E DevOps 实践：部署应用.....	39
6.1 方案概述.....	39
6.2 部署应用至 CCE.....	39
6.3 部署应用至 ECS.....	44

1 通过代理主机在内网部署应用

应用场景

通过代理主机在内网部署应用，能有效控制内外网通信，增强数据安全与网络性能，同时满足合规性要求；这一策略广泛应用于资源访问控制、数据中心间安全通信、内容缓存加速、环境隔离、安全审计及敏感数据处理等多个关键场景。

方案架构

采用squid代理服务中的Internet正向代理功能，在代理主机上指定目标主机的地址和端口，实现目标主机的公网访问。

想了解squid更多相关知识，请前往[squid官网](#)。下面以Linux系统机器进行演示。



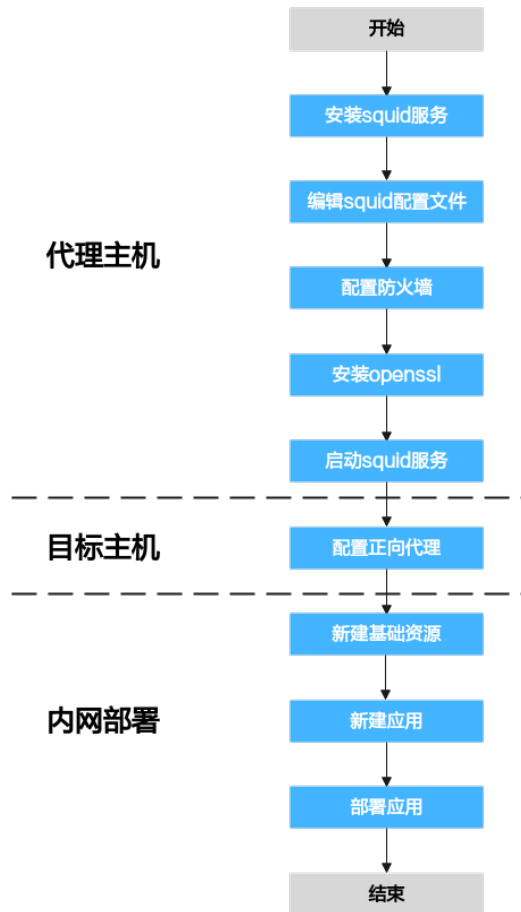
前提条件

- 一台绑定公网IP的“代理主机-B”，若无主机可参考[申请ECS](#)。
- 一台未绑定公网IP的“目标主机-A”。
- “代理主机-B”与“目标主机-A”可通过内网进行互相访问。

操作流程

本节介绍如何将应用通过代理主机部署到内网的主机或服务器。

图 1-1 操作流程示意图



步骤1 安装squid代理服务。

进入“代理主机-B”的命令行工具，执行以下命令：

```
yum install squid -y
```

若回显最后为“**Complete**”，可继续执行以下命令：

```
yum install iptables-services
```

回复“**Y**”，回显最后为“**Complete**”，即为安装完成。

步骤2 编辑squid配置文件。

1. 进入“代理主机-B”的命令行工具，执行以下命令：

```
vim /etc/squid/squid.conf
```

```
# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/24          # RFC1918 possible internal network
acl localnet src 172.16.0.0/16       # RFC1918 possible internal network
acl localnet src 192.168.0.0/16      # RFC1918 possible internal network
acl localnet src 10.0.0.0/8          # RFC 4193 local private network range
acl localnet src ::0/0               # RFC 4291 link-local (directly plugged) machines
acl SSL_ports port 443
acl Safe_ports port 80               # http
```

2. 在上图红框位置添加以下命令：

```
acl local src 主机的内网IP/24
```

3. 按“**ESC**”键，输入“**: wq**”，退出并保存。

步骤3 配置“代理主机-B”防火墙。

进入“代理主机-B”的命令行工具，依次执行以下命令：

```
systemctl stop firewalld.service
systemctl disable firewalld.service
yum install iptables-services iptables-devel -y
systemctl enable iptables.service
systemctl start iptables.service
iptables -I INPUT 1 -s 主机的内网IP/24 -p tcp --dport 3128 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 3128 -j DROP
```

📖 说明

其中倒数第二行的IP需配置为“目标主机-A”的内网IP网段或IP，3128是squid的代理端口。

步骤4 安装openssl。

进入“代理主机-B”的命令行工具，执行以下命令：

```
yum install openssl
```

回显最后为“**Complete**”，即为安装完成。

步骤5 启动squid代理服务。

进入“代理主机-B”的命令行工具，执行以下命令：

```
systemctl start squid //启动squid代理服务
systemctl status squid //检查squid状态
```

```
Oct 25 19:59:51 systemd[1]: Starting Squid caching proxy...
Oct 25 19:59:51 squid[8386]: 2022/10/25 19:59:51 acIIParseIpData: WARNING: Netmask masks away part of ...59/24
Oct 25 19:59:51 squid[8389]: Squid Parent: will start 1 kids
Oct 25 19:59:51 squid[8389]: Squid Parent: (squid-1) process 8311 started
Oct 25 19:59:51 systemd[1]: Started Squid caching proxy.
Hint: Some lines were ellipsized, use -l to show in full.
```

步骤6 配置正向代理。

进入“目标主机-A”的命令行工具，执行以下命令：

```
echo "export http_proxy=http://代理机的内网IP:3128" >>/etc/profile
echo "export https_proxy=http://代理机的内网IP:3128" >>/etc/profile
echo "export http_proxy=http://代理机的内网IP:3128" >>~/.bashrc
echo "export https_proxy=http://代理机的内网IP:3128" >>~/.bashrc
echo "export http_proxy=http://代理机的内网IP:3128" >>~/.bash_profile
echo "export https_proxy=http://代理机的内网IP:3128" >>~/.bash_profile
source /etc/profile
source ~/.bashrc
source ~/.bash_profile
```

步骤7 新建基础资源。

1. 单击“**首页**”可以查看已创建的所有项目。进入目标项目下，单击菜单“**设置 > 通用设置 > 基础资源管理**”，默认进入“**主机集群**”页面。
2. 单击“**新建主机集群**”，填写以下信息后，单击“**保存**”，完成主机集群的创建。

参数项	是否必填	说明
集群名称	是	请输入自定义的主机集群名称。
操作系统	是	根据即将添加主机的操作系统，选择“ Linux ”。
主机连通方式	是	选择代理模式。

参数项	是否必填	说明
执行主机	是	资源池是部署软件包时执行部署命令的物理环境的集合，本场景执行主机采用 官方资源池 。
描述	否	请输入对主机集群的描述。

- 单击“**添加或导入主机**”，“**选择添加方式**”选择“**通过IP手动添加**”，填写以下信息后，单击“**确定**”，完成代理主机的创建。


表 1-1 Linux 系统代理主机参数说明

参数项	是否必填	说明
主机名	是	请输入自定义的代理主机名称，示例：代理主机-B。
IP	是	请输入“ 代理主机-B ”所绑定的公网IP地址。
操作系统	是	不可更改，默认为当前主机集群的操作系统。
认证方式	是	本场景使用 密码 进行认证，输入的是“ 代理主机-B ”的用户名及密码。
SSH端口	是	推荐使用22端口。

- 单击“**添加或导入主机**”，“**选择添加方式**”选择“**通过IP手动添加**”，填写以下信息后，单击“**确定**”，完成目标主机的创建。

表 1-2 Linux 系统目标主机参数说明

参数项	是否必填	说明
主机名	是	请输入自定义的目标主机名称，示例：目标主机-A。
代理主机	是	请选择“ 代理主机-B ”为无法连接公网的目标主机进行网络代理。
IP	是	请输入“ 目标主机-A ”的私有IP地址。
操作系统	是	不可更改，默认为当前主机集群的操作系统。
认证方式	是	本场景使用 密码 进行认证，输入的是“ 目标主机-A ”的用户名及密码。
SSH端口	是	推荐使用22端口。


- 单击某个主机操作列的  图标，即可启动该主机进行连通性验证操作。

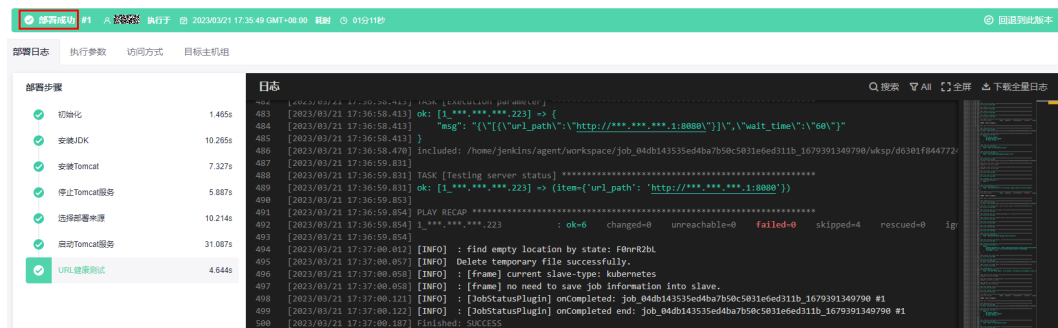
步骤8 新建应用。

- 登录软件开发生产线首页，单击目标项目名称，进入项目首页。
- 单击菜单“**持续交付 > 部署**”，进入项目下部署服务界面。
- 单击“**新建应用**”，进入“**基本信息**”页面，可根据需要修改名称、描述、执行主机等基本信息。

- 完成应用基本信息的编辑后，单击“下一步”，进入选择部署模板页面，选择“空白模板”，单击“确定”。
- 进入“部署步骤”页面，在右侧步骤列表中，单击目标步骤的“添加”键，可将该部署步骤添加到左侧的步骤编排区中。
- 进入“环境管理”页面，单击“新建环境”，输入环境基本信息，单击“保存”完成新建。
- 单击“导入主机”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，将“代理主机-B”、“目标主机-A”导入环境中。

步骤9 部署应用，详细操作可参考用户指南>应用管理。

- 应用列表中选择待部署应用，单击  图标，即可部署该应用。
- 部署完成后，单击应用名称，单击目标部署记录，应用状态栏的颜色变成绿色，并显示“部署成功”，则表示此次部署应用成功。



----结束

2 基于 Nginx 实现应用的灰度发布

应用场景

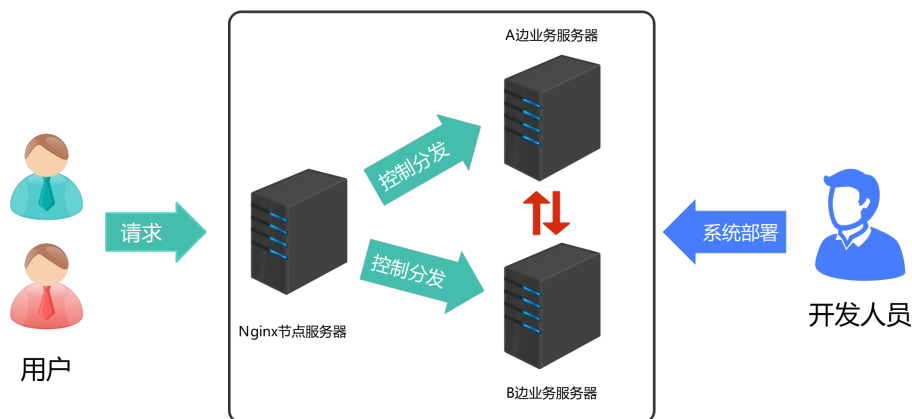
在您进行新系统升级时，会遇到停服部署或者无法灰度验证的情况，本实践采用基于 Nginx 负载均衡机制，在不影响业务正常运行的前提下，实现系统的平滑升级。

方案架构

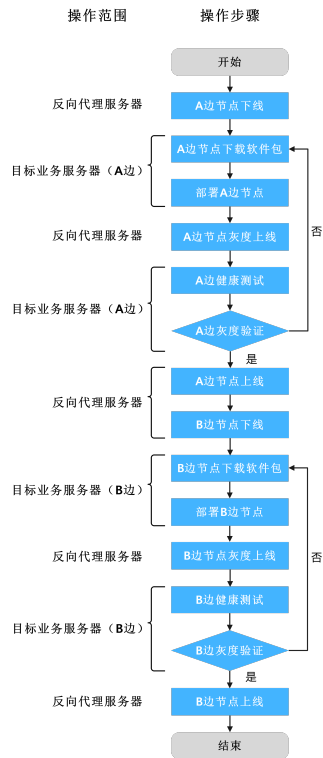
系统升级时，若采用蓝绿部署方式，开发人员需进行如下操作：

1. 先将A边服务器（原蓝环境）下线，同时访问流量将全部切分到B边服务器，此时对A边服务器进行升级操作。
2. A边服务器升级完毕后，将A边服务器设为灰度测试环境，由测试人员对A边服务器进行灰度验证。
3. 灰度验证完毕且功能正常后，A边业务服务器（绿环境）正式上线，并将所有流量切分到A边业务服务器，此时蓝绿部署完毕。
4. 在服务运行过程中，如果A边服务器出现紧急情况，执行蓝绿倒换实现业务快速恢复。

图 2-1 灰度发布原理



若采用金丝雀灰度部署方式，再重复前面的操作，将B边服务器进行升级，并完成灰度测试及正式上线，此时就完成新系统的灰度发布。



前提条件

- 已有可用项目，如果没有，请先[新建项目](#)。
- 具有创建应用的权限，参考[权限管理](#)。
- 已有目标业务服务器：**A_test**与**B_test**，且业务服务器已有应用服务正在运行。
- 已有反向代理服务器：**Gray_release**。
- 已有灰度验证主机，该主机代表灰度测试人员。

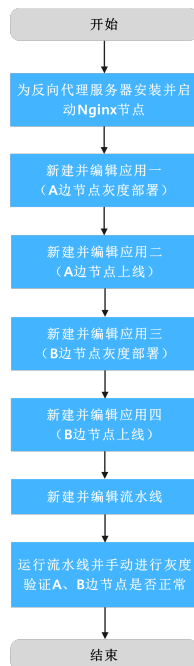
📖 说明

要确保各服务器之间可以相互通信，例如：将所有服务器加到同一VPC下。

操作流程

本文介绍基于Nginx负载均衡机制，实现主机部署场景的蓝绿发布和灰度发布能力，关于Nginx更多信息请参考[Nginx官网](#)。

图 2-2 操作流程示意图



步骤1 （可选）为反向代理服务器安装并启动Nginx节点。

📖 说明

若您的反向代理服务器已安装并启动Nginx节点，则本步骤跳过。

1. 新建基础资源。
 - a. 进入软件开发生产线首页，单击目标项目名称，进入目标项目下，单击菜单“设置 > 通用设置 > 基础资源管理”，默认进入“主机集群”页面。
 - b. 单击“新建主机集群”，填写集群名称、操作系统、使用代理机接入、执行主机及描述等基本信息后，单击“保存”，完成主机集群的创建。
 - c. 单击“添加或导入主机”，选择“通过IP手动添加”，填写主机名（例如：**A_test**、**B_test**、**Gray_release**）、IP、用户名、密码/密钥、ssh端口信息后，单击“确定”，重复操作完成三台目标主机的创建，并单击“连通性验证”进行连通性验证操作。
2. 新建应用。
 - a. 单击菜单“持续交付 > 部署”进入到部署服务页面。
 - b. 单击“新建应用”，进入“基本信息”页面，可根据需要修改名称、描述、执行主机等基本信息。
 - c. 完成应用基本信息的编辑后，单击“下一步”，进入选择部署模板页面。
 - d. 选择“空白模板”，单击“确定”，进入“部署步骤”页签。
3. 编辑应用。
 - a. 切换到“环境管理”页签，添加并编辑环境。
 - 单击“新建环境”，填写环境名，例如：**反向代理服务器组**，选择与服务器对应的操作系统，填写描述信息等。
 - 单击“保存”，完成环境的创建。


- 单击“**导入主机**”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，单击目标主机操作列的图标，即可将该主机导入环境中。
- b. 切换到“**部署步骤**”页签，添加并编辑以下步骤。
 - 添加“**安装Nginx**”步骤，并修改为下表参数（以Linux系统为例）。

表 2-1 参数填写说明

参数	填写说明
环境	选择“反向代理服务器组”。
nginx版本	选择目标版本。示例：nginx-1.14.2。
nginx安装路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。

- 添加“**启动/停止Nginx**”步骤，并修改为下表参数（以Linux系统为例）。


表 2-2 参数填写说明

参数	填写说明
环境	选择“反向代理服务器组”。
操作类型	选择“启动（nginx）”。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。

- c. 单击“**保存并部署**”，开始部署应用。
4. 部署应用。
- 部署完成后，应用状态栏的颜色变成绿色，并显示“**部署成功**”，则表示此次部署应用成功。
- 若应用状态栏的颜色变成红色，并显示“**部署失败**”，则表示此次部署应用失败，请单击“**查看解决方案**”链接查看解决方法。

步骤2 新建并编辑应用一（A边节点灰度部署）。

1. 新建应用。
 - a. 单击菜单“**持续交付 > 部署**”进入到部署服务页面。
 - b. 单击“**新建应用**”，进入“**基本信息**”页面，可根据需要修改名称、描述、执行主机等基本信息。
 - c. 完成应用基本信息的编辑后，单击“**下一步**”，进入选择部署模板页面。
 - d. 选择“**通用部署**”模板，单击“**确定**”。
2. 编辑应用。
 - a. 切换到“**环境管理**”页签，添加并编辑环境。

- 单击“**新建环境**”，填写环境名，例如：**反向代理服务器组**，选择与服务器对应的操作系统，填写描述信息等。
 - 单击“**保存**”，完成环境的创建。
 - 单击“**导入主机**”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，单击目标主机操作列的图标，即可将该主机导入环境中。
 - 重复上面操作步骤，完成新建“**目标业务服务器组_A边节点**”并添加“**A_test**”服务器的操作。
- b. 切换到“**参数设置**”页签，新增如下参数。

名称	类型	默认值	私密参数	运行前设置	描述
app_name	字符串	test	<input type="checkbox"/>	<input type="checkbox"/>	应用名称，用于获取进程id并...
service_port	字符串	3000	<input type="checkbox"/>	<input type="checkbox"/>	应用端口

- c. 切换到“**部署步骤**”页签，添加并编辑以下步骤。
- 添加“**启动/停止Nginx**”步骤，并修改为下表参数（以Linux系统为例）。

表 2-3 参数填写说明

参数	填写说明
步骤显示名称	示例：A边节点下线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 A边节点下线-代码示例 。

- 编辑“**下载软件包**”步骤，并修改为下表参数（以Linux系统为例）。

表 2-4 参数填写说明

参数	填写说明
步骤显示名称	示例：A边节点下载软件包。

参数	填写说明
选择源类型	选择源类型。示例：制品仓库。
环境	选择目标环境。示例：目标业务服务器组_A边节点。
选择软件包	选择制品仓库中需要部署的软件包。
下载到主机的部署目录	软件包下载到目标主机的路径。示例：/usr/local/。

- 编辑“执行部署脚本”步骤，并修改为下表参数（以Linux系统为例）。

表 2-5 参数填写说明

参数	填写说明
步骤显示名称	示例：部署A边节点。
环境	选择目标环境。示例：目标业务服务器组_A边节点。
shell命令	填写待执行命令。示例：见附录中 部署节点 。

- 添加“启动/停止Nginx”步骤，并修改为下表参数（以Linux系统为例）。

表 2-6 参数填写说明

参数	填写说明
步骤显示名称	示例：A边节点灰度上线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 A边节点灰度上线-代码示例 。

- 编辑“健康测试”步骤，并修改为下表参数（以Linux系统为例）。

表 2-7 参数填写说明

参数	填写说明
步骤显示名称	示例：A边健康测试。
环境	选择目标环境。示例：目标业务服务器组_A边节点。
重试次数	失败后重试次数，超过预设的重试次数，服务未启动会导致URL检测失败。示例：1。
间隔时间	每次重试测试的间隔时间（s）。示例：60。
测试路径	URL健康测试，支持添加多个。示例：http://127.0.0.1:3000（应用服务的IP及端口）。

- 单击“保存”，应用创建完成。



步骤3 新建并编辑应用二（A边节点上线）。

- 新建应用。
 - 单击“新建应用”，进入“基本信息”页面，可根据需要修改名称、描述、执行主机等基本信息。
 - 完成应用基本信息的编辑后，单击“下一步”，进入选择部署模板页面。
 - 选择“空白模板”，单击“确定”。
- 编辑应用。
 - 切换到“环境管理”页签，添加并编辑环境。
 - 单击“新建环境”，填写环境名，例如：反向代理服务器组，选择与服务器对应的操作系统，填写描述信息等。


- 单击“保存”，完成环境的创建。
 - 单击“导入主机”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，单击目标主机操作列的图标，即可将该主机导入环境中。
- b. 切换到“部署步骤”页签，添加并编辑以下步骤。
添加“启动/停止Nginx”，并修改为下表参数（以Linux系统为例）。

表 2-8 参数填写说明

参数	填写说明
步骤显示名称	示例：A边节点上线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 节点上线-代码示例 。

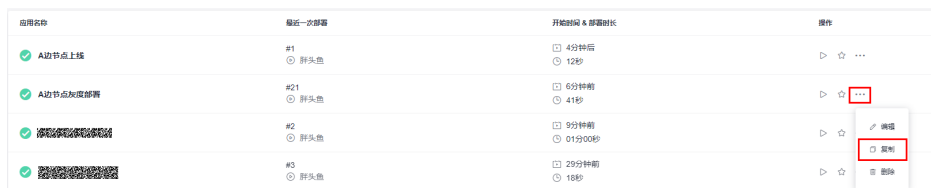
3. 单击“保存”，应用创建完成。



步骤4 复制并编辑应用一，完成应用三的创建（B边节点灰度部署）。

1. 复制应用。

单击图标，单击“复制”，完成应用的复制。



2. 编辑应用。

- a. 切换到“环境管理”页签，添加并编辑环境。


- 单击“**新建环境**”，填写环境名，例如：**反向代理服务器组**，选择与服务器对应的操作系统，填写描述信息等。
 - 单击“**保存**”，完成环境的创建。
 - 单击“**导入主机**”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，单击目标主机操作列的图标，即可将该主机导入环境中。
 - 重复上面操作步骤，完成新建“**目标业务服务器组_B边节点**”并添加“**B_test**”服务器的操作。
- b. 切换到“**部署步骤**”页签，添加并编辑以下步骤。
- 编辑“**A边节点下线**”步骤，并修改为下表参数（以Linux系统为例）。

表 2-9 参数填写说明

参数	填写说明
步骤显示名称	示例：B边节点下线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 B边节点下线-代码示例 。

- 编辑“**A边节点下载软件包**”步骤，并修改为下表参数（以Linux系统为例）。

表 2-10 参数填写说明

参数	填写说明
步骤显示名称	示例：B边节点下载软件包。
选择源类型	选择源类型。示例：制品仓库。
环境	选择目标环境。示例：目标业务服务器组_B边节点。
选择软件包	选择制品仓库中需要部署的软件包。

参数	填写说明
下载到主机的部署目录	软件包下载到目标主机的路径。示例：/usr/local/。

- 编辑“部署A边节点”步骤，并修改为下表参数（以Linux系统为例）。

表 2-11 参数填写说明

参数	填写说明
步骤显示名称	示例：部署B边节点。
环境	选择目标环境。示例：目标业务服务器组_B边节点。
shell命令	填写待执行命令。示例：见附录中 部署节点 。

- 编辑“A边节点灰度上线”步骤，并修改为下表参数（以Linux系统为例）。

表 2-12 参数填写说明

参数	填写说明
步骤显示名称	示例：B边节点灰度上线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 B边节点灰度上线-代码示例 。

- 编辑“A边健康测试”步骤，并修改为下表参数（以Linux系统为例）。

表 2-13 参数填写说明

参数	填写说明
步骤显示名称	示例：B边健康测试。
环境	选择目标环境。示例：目标业务服务器组_B边节点。
重试次数	失败后重试次数，超过预设的重试次数，服务未启动会导致URL检测失败。示例：1。
间隔时间	每次重试测试的间隔时间（s）。示例：60。
测试路径	URL健康测试，支持添加多个。示例：http://127.0.0.1:3000（应用服务的IP及端口）。

3. 单击“保存”，应用创建完成。




步骤5 复制并编辑应用二，完成应用四的创建（B边节点上线）。

1. 复制应用。

单击  图标，单击“复制”，完成应用的复制。

2. 编辑应用。

- a. 切换到“环境管理”页签，添加并编辑环境。

- 单击“新建环境”，填写环境名，例如：反向代理服务器组，选择与服务器对应的操作系统，填写描述信息等。
- 单击“保存”，完成环境的创建。
- 单击“导入主机”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，单击目标主机操作列的  图标，即可将该主机导入环境中。

- b. 切换到“部署步骤”页签，添加并编辑以下步骤。

编辑“**A边节点上线**”，并修改为下表参数（以Linux系统为例）。

表 2-14 参数填写说明

参数	填写说明
步骤显示名称	示例：B边节点上线。
环境	选择目标环境。示例：反向代理服务器组。
操作类型	选择操作类型。示例：重载配置文件（reload）。
Nginx安装的路径	目标环境中Nginx服务的安装路径。示例：/usr/local/nginx。
是否在操作之前修改配置文件	勾选。
要修改的Nginx配置文件路径	目标主机中Nginx配置文件所在路径。示例：/usr/local/nginx/conf/nginx.conf。
配置文件备份路径	目标主机中原Nginx配置文件备份目标路径。示例：/usr/local/nginx/conf/nginx_bak.conf。
配置文件内容	新配置文件内容。示例：见附录中 节点上线-代码示例 。

- 单击“**保存**”，应用创建完成。



步骤6 新建并编辑流水线。

- 新建流水线。
 - 单击菜单“**持续交付 > 流水线**”进入到流水线服务页面。
 - 单击“**新建流水线**”，选择“**所属项目**”，输入“**名称**”，流水线源选择“**暂不选择**”，单击“**下一步**”。
 - 模板选择“**空模板**”，单击“**确定**”。




- 编辑流水线阶段任务一（**A边节点灰度部署**）。
 - 单击 ，将弹框中的参数修改为如下参数并单击“**确定**”。

表 2-15 参数填写说明

参数	填写说明
名称	示例：A边节点灰度部署。
总是运行	选择“否”。


- 单击  图标，在弹框中选择准入类型为“自动”，单击“确定”。
- 单击“新建任务”，单击“部署”，选择“Deploy部署”，单击“添加”，将弹框中的参数修改为如下参数并单击“确定”。

表 2-16 参数填写说明

参数	填写说明
名称	示例：A边节点灰度部署。
请选择需要调用的任务	选择“A边节点灰度部署”。
关联构建任务	暂不选择。

3. 新建并编辑流水线阶段任务二（A边节点上线）。



- 单击  图标，单击 ，将弹框中的参数修改为如下参数并单击“确定”。

表 2-17 参数填写说明

参数	填写说明
名称	示例：A边节点上线。
总是运行	选择“否”。


- 单击  图标，在弹框中选择准入类型为“自动”，单击“确定”。
- 单击“新建任务”，单击“通用 > 人工审核 > 添加”，将弹框中的参数修改为如下参数并单击“确定”。

表 2-18 参数填写说明

参数	填写说明
名称	示例：A边节点灰度验证。
审核人	选择业务验证人员。
审核模式	选择“全部人审核”。

参数	填写说明
超时处理	选择“审核失败，流水线终止执行”。
审核时长	示例：4小时。
审核说明	选填。


- 如下图所示，单击  图标，单击“部署”，选择“Deploy部署”，单击“添加”，将弹框中的参数修改为如下参数并单击“确定”。



表 2-19 参数填写说明

参数	填写说明
名称	示例：A边节点上线。
请选择需要调用的任务	选择“A边节点上线”。
关联构建任务	暂不选择。

4. 编辑流水线阶段任务三（B边节点灰度部署）。




- 单击  图标，单击  ，将弹框中的参数修改为如下参数并单击“确定”。

表 2-20 参数填写说明

参数	填写说明
名称	示例：B边节点灰度部署。
总是运行	选择“否”。

- 单击  图标，在弹框中选择准入类型为“自动”，单击“确定”。

- 单击“新建任务”，单击“部署”，选择“Deploy部署”，单击“添加”，将弹框中的参数修改为如下参数并单击“确定”。

表 2-21 参数填写说明

参数	填写说明
名称	示例：B边节点灰度部署。
请选择需要调用的任务	选择“B边节点灰度部署”。
关联构建任务	暂不选择。

5. 新建并编辑流水线阶段任务四（B边节点上线）。

- 单击  图标，单击  ，将弹框中的参数修改为如下参数并单击“确定”。

表 2-22 参数填写说明

参数	填写说明
名称	示例：B边节点上线。
总是运行	选择“否”。


- 单击  图标，在弹框中选择准入类型为“自动”，单击“确定”。
- 单击“新建任务”，单击“通用 > 人工审核 > 添加”，将弹框中的参数修改为如下参数并单击“确定”。

表 2-23 参数填写说明

参数	填写说明
名称	示例：B边节点灰度验证。
审核人	选择业务验证人员。
审核模式	选择“全部人审核”。
超时处理	选择“审核失败，流水线终止执行”。
审核时长	示例：4小时。
审核说明	选填。


- 如下图所示，单击  图标，单击“部署”，选择“Deploy部署”，单击“添加”，将弹框中的参数修改为如下参数并单击“确定”。



表 2-24 参数填写说明

参数	填写说明
名称	示例：B边节点上线。
请选择需要调用的任务	选择“B边节点上线”。
关联构建任务	暂不选择。

6. 完成以上操作后，单击“保存并运行”，开始运行流水线任务。



步骤7 执行流水线并手动进行灰度验证A、B边节点是否正常。

当流水线执行到上线A边节点或上线B边节点时，流水线暂停执行，需要灰度用户手动验证灰度环境下的A边、B边业务服务器是否正常，业务正常后再继续运行流水线。

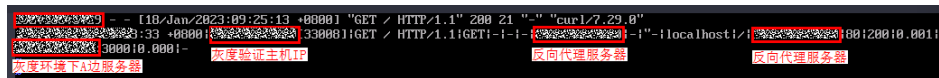


灰度用户可使用“curl”命令验证灰度环境系统是否正常。

```
curl http://反向代理服务器IP:Nginx端口
```

说明

查看灰度用户是否进入到目标灰度环境服务器，可登录反向代理服务器，进入logs/access.log路径查看日志。



----结束

附录

- A边节点下线-代码示例

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local|$remote_addr|$remote_port|$request|$request_method|
$content_length|
$content_type|$http_referer|$host|$http_x_forwarded_for|
$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #访问日志：存放路径，日志级别。
    error_log logs/error.log; #错误日志：存放路径。
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        # 下面填入A主机IP及应用服务端口
        #server X.X.X.X; #A节点下线
        # 下面填入B主机IP及应用服务端口
        server X.X.X.X;
    }
    upstream portal_test {
```

```
# 下面填入A主机IP及应用服务端口
server X.X.X.X;
# 下面填入B主机IP及应用服务端口
server X.X.X.X;
}

server {
    listen    XXX;#填入Nginx端口
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #下面填入灰度验证机器IP
        #if ( $remote_addr ~* "X.X.X.X" ) {
        #    set $backend $test;
        #}
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
}
```

- **部署节点**

```
# 获取应用进程id
pid=`ps -ef | grep app_name | grep -v grep | awk '{print $2}'`
if [ -z "$pid" ];
then
    echo "[app_name pid is not exist.]"
else
    echo "app_name pid: $pid "
    # 停止该进程
    kill -15 $pid
fi
# 重新启动应用，可通过执行部署脚本和部署命令两种方式启动应用，如下：
# 方式一：通过执行部署脚本启动应用
# sh startup.sh
# 方式二：通过执行命令启动应用，建议采用nohup后台启动方式
# nohup java -jar /usr/local/app/SpringbootDemo.jar &
```

- **A边节点灰度上线-代码示例**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local|$remote_addr|$remote_port|$request|$request_method|
$content_length|
$content_type|$http_referer|$host|$http_x_forwarded_for|
$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #访问日志：存放路径，日志级别。
    error_log logs/error.log; #错误日志：存放路径。
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        # 下面填入A主机IP及应用服务端口
        #server X.X.X.X; #A节点下线
        # 下面填入B主机IP及应用服务端口
        server X.X.X.X;
    }
    upstream portal_test {
        # 下面填入A主机IP及应用服务端口
        server X.X.X.X; #A节点灰度上线
        # 下面填入B主机IP及应用服务端口
```

```
#server X.X.X:X;
}

server {
    listen    XXX;#填入Nginx端口
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #下面填入灰度验证机器IP
        if ( $remote_addr ~* "X.X.X.X") {
            set $backend $test;
        }
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
}
```

- **B边节点下线-代码示例**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port] $request $request_method $content_length'
        '$content_type|$http_referer|$host|$http_x_forwarded_for|'
        '$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|'
        '$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #访问日志: 存放路径, 日志级别。
    error_log logs/error.log; #错误日志: 存放路径。
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        # 下面填入A主机IP及应用服务端口
        server X.X.X.X;
        # 下面填入B主机IP及应用服务端口
        #server X.X.X.X; #B节点下线
    }
    upstream portal_test {
        # 下面填入A主机IP及应用服务端口
        server X.X.X.X;
        # 下面填入B主机IP及应用服务端口
        server X.X.X.X;
    }
}

server {
    listen    XXX;#填入Nginx端口
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #下面填入灰度验证机器IP
        #if ( $remote_addr ~* "X.X.X.X") {
        # set $backend $test;
        #}
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
```

- **B边节点灰度上线-代码示例**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|
$content_length|
$content_type|$http_referer|$host|$http_x_forwarded_for|
$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #访问日志: 存放路径, 日志级别。
    error_log logs/error.log; #错误日志: 存放路径。
    sendfile on;
    keepalive_timeout 65;
    upstream portal {
        # 下面填入A主机IP及应用服务端口
        server X.X.X.X;
        # 下面填入B主机IP及应用服务端口
        #server X.X.X.X; #B节点下线
    }
    upstream portal_test {
        # 下面填入A主机IP及应用服务端口
        #server X.X.X.X;
        # 下面填入B主机IP及应用服务端口
        server X.X.X.X; #B节点灰度上线
    }

    server {
        listen XXX;#填入Nginx端口
        server_name localhost;

        location / {
            set $backend portal;
            set $test portal_test;
            #下面填入灰度验证机器IP
            if ( $remote_addr ~* "X.X.X.X" ) {
                set $backend $test;
            }
            proxy_pass https://$backend;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root html;
        }
    }
}
```

- **节点上线-代码示例**

```
worker_processes 1;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    log_format main '$time_local$remote_addr[$remote_port]$request$request_method|
$content_length|
$content_type|$http_referer|$host|$http_x_forwarded_for|
$http_true_client_ip|$server_name|$request_uri|$server_addr|$server_port|
$status|$request_time|$upstream_addr|$upstream_response_time|$cookie_domain_tag';
    access_log logs/access.log main; #访问日志: 存放路径, 日志级别。
    error_log logs/error.log; #错误日志: 存放路径。
    sendfile on;
    keepalive_timeout 65;
```

```
upstream portal {
    # 下面填入A主机IP及应用服务端口
    server X.X.X.X;
    # 下面填入B主机IP及应用服务端口
    server X.X.X.X;
}
upstream portal_test {
    # 下面填入A主机IP及应用服务端口
    server X.X.X.X;
    # 下面填入B主机IP及应用服务端口
    server X.X.X.X;
}

server {
    listen    XXX;#填入Nginx端口
    server_name localhost;

    location / {
        set $backend portal;
        set $test portal_test;
        #下面填入灰度验证机器IP
        #if ( $remote_addr ~* "X.X.X.X") {
        #    set $backend $test;
        #}
        proxy_pass https://$backend;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root html;
    }
}
}
```

3 基于 Kubernetes Nginx-Ingress 实现应用的灰度发布

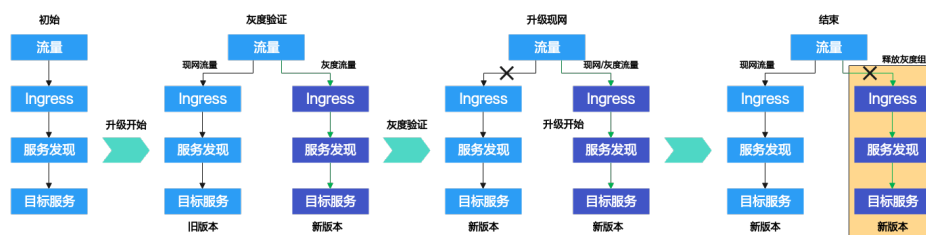
应用场景

本实践基于Kubernetes原生的特性实现灰度发布。在您进行新系统升级时，会遇到停服部署或者无法灰度验证的情况，本实践采用基于Kubernetes原生的Service特性，在不影响业务正常运行的前提下，实现系统的平滑升级。

方案架构

系统升级时，开发人员第一次部署应用会创建出一组灰度负载，此时灰度负载中的系统版本为新版本，此时Service将部分流量转发至灰度负载上，由测试人员在灰度负载中进行版本验证。版本验证结束后，开发人员开始第二次部署应用升级现网服务，此时Service将全部流量转发至灰度负载上并升级现网服务，现网服务升级到最新版本后，Service将全部流量转发回现网负载并释放灰度负载，完成新系统的发布。

图 3-1 灰度发布原理



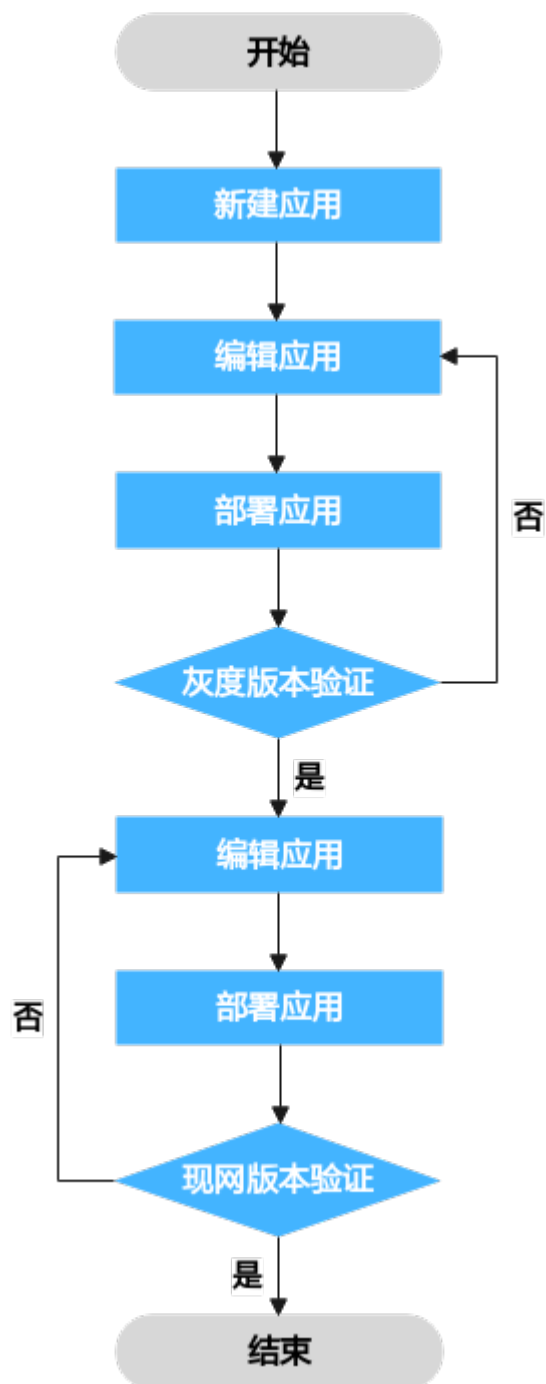
前提条件

- 已有可用项目，如果没有，请先[新建项目](#)。
- 具有创建应用的权限，参考[权限管理](#)。
- 服务包含以下资源，并定义为v1版本：
 - 已有CCE集群，示例：cce-demo；
 - CCE集群中已创建无状态工作负载，示例：deployment-doc；
 - CCE集群中已创建服务，示例：service-doc；
 - CCE集群中已创建路由，示例：ingress-doc；

- CCE集群中已安装nginx-ingress插件。

操作流程

图 3-2 操作流程示意图



步骤1 新建应用。

1. 进入软件开发生产线平台，单击目标项目名称进入项目。
2. 单击“持续交付 > 部署”，单击“新建应用”，进入“基本信息”页面。

3. 可根据需要修改以下基本信息。

参数项	说明
名称	必填。应用的名称。示例： Kubernetes_Nginx-Ingress灰度部署
所属项目	默认。该应用的归属项目。
描述	可选。对应用的描述。示例： 无
执行主机	可选。资源池是部署软件包时执行部署命令的物理环境的集合，您可以使用华为云托管的 官方资源池 ，您也可以将自有的服务器作为 自托管资源池 托管到华为云，托管方式请参考 自托管资源池场景 。示例： 官方资源池
来自流水线	可选。开启后，只能通过流水线驱动执行，不能单独执行。

4. 完成应用基本信息的编辑后，单击“下一步”，进入选择部署模板页面，选择“空白模板”，单击“确定”。

步骤2 编辑应用。

在“部署步骤”页签中添加“Kubernetes Nginx-Ingress灰度发布(CCE集群)”步骤并修改为如下参数：

表 3-1 参数说明

参数项	说明	示例
步骤显示名称	步骤添加后在部署步骤编排区显示的名称。	默认
部署租户	<ul style="list-style-type: none">当前租户：表示将软件包部署到当前租户的CCE集群中发布。选择“当前租户”，当前租户需要有CCE集群操作权限，如果没有CCE集群操作权限，可以通过下方的IAM授权选择已有权限的授权用户进行部署。其他租户：表示通过IAM授权方式将软件包部署到其他租户的CCE集群中发布。选择“其他租户”，则必须选择授权租户进行CCE集群部署。	当前租户
IAM授权	如果当前用户无操作权限执行接口时，可通过IAM提权获取到其他用户的临时AK/SK去执行CCE接口。	不勾选
区域	选择要部署的区域。	默认
集群名称	选择CCE云容器引擎中申请的K8S集群。	cce-ldf
命名空间名称	选择CCE云容器引擎中K8S集群的命名空间。	default
工作负载名称	选择需要部署的无状态工作负载。	deployment-doc

参数项	说明	示例
访问方式名称	选择与目标工作负载所绑定的service服务名称。	service-doc
路由名称	选择与目标service服务所绑定的路由名称。	ingress-doc
容器名称	选择需要部署的容器实例名。	container-1
镜像名称	选择需要部署的镜像。	默认
镜像版本	选择需要部署的镜像版本。	v2
开启灰度配置	灰度策略： <ul style="list-style-type: none"> ● Header Header-Key：持输入自定义Header的键。 Header-Value：支持输入自定义Header的值，支持字符串及正则表达式，正则表达式格式为：<code>^...\$</code>。 灰度流量权重（%）：支持流量自定义配置。 ● Cookie Cookie：支持输入自定义Cookie内容。 灰度流量权重（%）：支持流量自定义配置。 说明 Header与Cookie所输入的字符串长度最大为500。	勾选 灰度策略： Header Header-Key： foo Header-Value： bar 灰度流量权重 （%）：30

步骤3 部署应用（创建灰度版本）。

单击“保存并部署”，部署该应用。此时，部署服务已为您在CCE集群中创建了以下灰度版本资源并将30%的现网流量引向灰度负载：

- 工作负载：deployment-doc-v2，镜像为V2版本

图 3-3 增加镜像版本为 V2 的工作负载



- 服务：service-doc-v2
- 路由：ingress-doc-v2

说明

此时，测试人员可以在Header中添加一条数据（Key取值为foo，Value取值为bar）到灰度负载中验证最新版本。

步骤4 编辑应用（部署最新版本）。

进入步骤1中创建的应用，修改以下参数：

表 3-2 参数说明

参数项	示例
开启灰度配置	不勾选

步骤5 部署应用（部署最新版本）。

单击“保存并部署”，部署该应用。此时，部署服务已为您在CCE集群中删除了以下灰度环境资源并将V1版本镜像替换为V2版本镜像：

- 工作负载： deployment-doc-v2
- 服务： service-doc-v2
- 路由： ingress-doc-v2

图 3-4 镜像版本升级为 V2



说明

此时，您可以到现网查看系统是否为升级后的最新版本。

----结束

4 通过自托管资源池部署应用至云下 IDC

应用场景

企业能够在保留自有数据中心基础设施的同时，利用云服务的灵活性和扩展性，实现混合云环境下的应用部署与管理；此外，这种部署模式支持复杂的工作负载迁移，促进资源优化及业务连续性，特别是在需要高性能计算或数据驻留合规的场景中。

方案架构

采用自托管资源池功能，通过自定义执行机连接CodeArts和云下IDC，实现云下IDC的主机或容器集群的部署。

下面以部署至云下IDC主机进行演示。



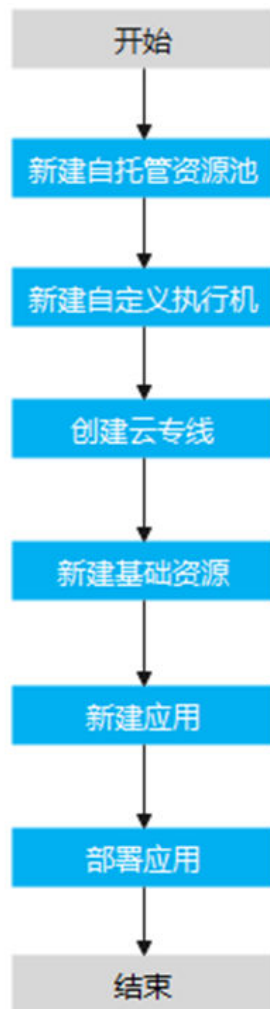
前提条件

- 一台具备访问公网能力的“自定义执行机-A”，若无自定义执行机可参考“[申请ECS](#)”和“[申请EIP](#)”。
- 一台具备访问公网能力的“云下目标主机-B”。
- “自定义执行机-A”与“云下目标主机-B”可通过内网互相访问，“云下目标主机-B”需允许“自定义执行机-A”访问22端口。

操作流程

本节介绍如何将应用通过自托管资源池部署到云下IDC的主机或容器集群。

图 4-1 操作流程示意图



步骤1 新建自托管资源池

1. 登录CodeArts首页，在导航栏中单击用户名。
2. 选择“租户设置”。
3. 单击“资源池管理 > 资源池”，进入“资源池管理”页面，单击“新建资源池”，输入资源池名称，资源池类型选择“LINUX”，单击“保存”，完成资源池配置。

步骤2 新建自定义执行机，详情请参考“新建代理”。

步骤3 创建云专线DC，打通云上和云下资源的网络连接，请参考“创建云专线”。


步骤4 新建基础资源

1. 进入目标项目下，单击菜单“设置 > 通用设置 > 基础资源管理”，默认进入“主机集群”页面。
2. 单击“新建主机集群”，填写以下信息后，单击“保存”，完成主机集群的创建。

参数项	是否必填	说明
集群名称	是	请输入自定义的主机集群名称。
操作系统	是	根据即将添加主机的操作系统，选择“Linux”。
主机连通方式	是	选择直连模式
执行主机	是	资源池是部署软件包时执行部署命令的物理环境的集合，本场景执行主机采用自托管资源池，选择步骤1中新建的自托管资源池。
描述	否	请输入对主机集群的描述。

- 单击“添加或导入主机”，填写以下信息后，单击“确定”，完成目标主机的创建。

参数项	是否必填	说明
主机名	是	请输入自定义的目标主机名称，示例：目标主机B。
IP	是	请输入“目标主机-B”的私有IP地址。
操作系统	是	不可更改，默认为当前主机集群的操作系统。
认证方式	是	本场景使用密码进行认证，输入的是“目标主机-B”的用户名及密码。
SSH端口	是	推荐使用22端口。

- 单击某个主机操作列的  图标，即可启动该主机进行连通性验证操作。

步骤5 新建应用。

- 登录软件开发生产线首页，单击目标项目名称，进入项目首页。
- 单击菜单“持续交付 > 部署”，进入项目下部署服务界面。
- 单击“新建应用”，进入“基本信息”页面，可根据需要修改应用名称、描述、执行主机等基本信息，执行主机采用自托管资源池，选择上述操作中新建的自托管资源池。
- 完成应用基本信息的编辑后，单击“下一步”，进入选择部署模板页面，选择“空白模板”，单击“确定”。
- 进入“部署步骤”页面，在右侧步骤列表中，单击目标步骤的“添加”键，可将该部署步骤添加到左侧的步骤编排区中。
- 进入“环境管理”页面，单击“新建环境”，输入环境基本信息，单击“保存”完成新建。
- 单击“导入主机”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，将“目标主机-B”导入环境中

步骤6 部署应用，详细操作可参考部署应用。

----结束

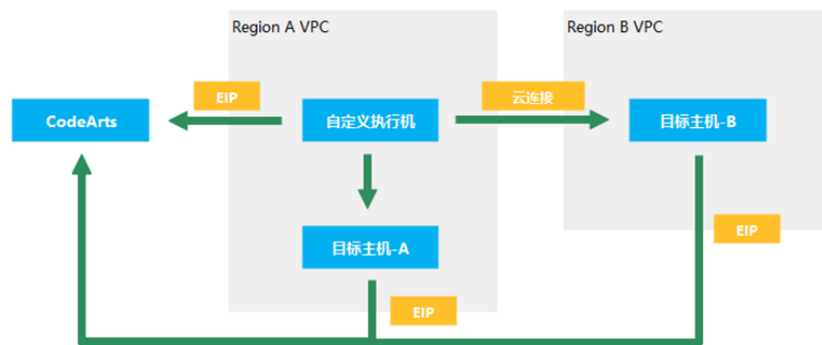
5 通过自托管资源池实现跨 Region 虚拟机部署

应用场景

通过自托管资源池实现跨Region虚拟机部署，企业可以统一管理分布在不同地区的资源，优化应用性能，减少数据冗余并增强灾备能力，确保业务连续性和全球用户低延迟访问。

方案架构

采用VPC Peering功能，将用户自托管资源池主机与其他Region的目标主机的网络连接，实现使用同一套代理执行机，部署至多Region的华为云主机。



前提条件

- 一台具备访问公网能力的“自定义执行机-A”，若无自定义执行机可参考[申请ECS](#)和“[申请EIP](#)”。
- 一台具备访问公网能力的“目标主机-B”。
- “自定义执行机-A”与“目标主机-B”在不同的Region。

操作步骤

本节介绍如何将应用通过自托管资源池部署到华为云其他Region的主机。

图 5-1 操作流程示意图



步骤1 新建自托管资源池

1. 登录CodeArts首页，在导航栏中单击用户名。
2. 选择“租户设置”。
3. 单击“资源池管理 > 资源池”，进入“资源池管理”页面，单击“新建资源池”，输入资源池名称，资源池类型选择“LINUX”，单击“保存”，完成资源池配置。

步骤2 新建自定义执行机，详情请参考“[新建代理](#)”。

步骤3 创建云连接，打通自托管资源池所在VPC与目标主机-B所在VPC的网络连接，请参考“[跨区域VPC互通](#)”。

步骤4 配置目标主机安全组。

1. 进入控制台，在页面左上角单击“服务列表 > 计算 > 弹性云服务器ECS”，进入“弹性云服务器ECS”服务。
2. 通过名称搜索找到“目标主机-B”。
3. 单击“目标主机-B”名称，进入主机详情页面。
4. 单击“安全组”，切换到安全组标签页，单击“配置规则”，在“目标主机-B”安全组的入方向规则中，添加规则，允许“自定义执行机”的IP网段访问22端口。


步骤5 新建基础资源。

1. 进入目标项目下，单击菜单“设置 > 通用设置 > 基础资源管理”，默认进入“主机集群”页面。
2. 单击“新建主机集群”，填写以下信息后，单击“保存”，完成主机集群的创建。

参数项	是否必填	说明
集群名称	是	请输入自定义的主机集群名称。
操作系统	是	根据即将添加主机的操作系统，选择“Linux”。
主机连通方式	是	选择直连模式
执行主机	是	资源池是部署软件包时执行部署命令的物理环境的集合，本场景执行主机采用自托管资源池，选择步骤1中新建的自托管资源池。
描述	否	请输入对主机集群的描述。

3. 单击“添加或导入主机”，填写以下信息后，单击“确定”，完成目标主机的创建。

参数项	是否必填	说明
主机名	是	请输入自定义的目标主机名称，示例：目标主机B。
IP	是	请输入“目标主机-B”的私有IP地址。
操作系统	是	不可更改，默认为当前主机集群的操作系统。
认证方式	是	本场景使用密码进行认证，输入的是“目标主机-B”的用户名及密码。
SSH端口	是	推荐使用22端口。

4. 单击某个主机操作列的  图标，即可启动该主机进行连通性验证操作。

步骤6 新建应用。

1. 登录软件开发生产线首页，单击目标项目名称，进入项目首页。
2. 单击菜单“持续交付 > 部署”，进入项目下部署服务界面。

3. 单击“**新建应用**”，进入“**基本信息**”页面，可根据需要修改应用名称、描述、执行主机等基本信息，执行主机采用自托管资源池，选择上述操作中新建的自托管资源池。
4. 完成应用基本信息的编辑后，单击“**下一步**”，进入选择部署模板页面，选择“**空白模板**”，单击“**确定**”。
5. 进入“**部署步骤**”页面，在右侧步骤列表中，单击目标步骤的“**添加**”键，可将该部署步骤添加到左侧的步骤编排区中。
6. 进入“**环境管理**”页面，单击“**新建环境**”，输入环境基本信息，单击“**保存**”完成新建。
7. 单击“**导入主机**”，系统已自动筛选符合当前环境的所有集群，在弹框中选择目标主机集群，将“**目标主机-B**”导入环境中。

步骤7 部署应用，详细操作可参考[部署应用](#)。

----**结束**

6 HE2E DevOps 实践：部署应用

6.1 方案概述

本文以“DevOps全流程示例项目”为例，介绍如何部署应用至CCE与ECS。

开展实践前，需要完成[编译构建](#)。

样例项目中预置了以下3个部署应用。

其中，第一个用于CCE部署，第二、三个用于ECS部署。

表 6-1 预置应用

预置应用	应用说明
phoenix-cd-cce	部署至CCE流程对应的应用。
phoenix-sample-standalone	部署至ECS流程对应的应用。
phoenix-sample-predeploy	向ECS中安装依赖工具操作对应的应用。

6.2 部署应用至 CCE

购买并配置云容器引擎

本节中使用的是云容器引擎CCE。

通过控制台可[购买CCE集群](#)。

其中集群及节点的必要配置建议参照[表6-2](#)与[表6-3](#)，表中未涉及的可根据实际情况选择。

表 6-2 CCE 集群购买配置

配置分类	配置项	配置建议
基础配置	计费模式	选择“按需计费”。
	集群版本	根据需要选择，建议选择最新版本。
网络配置	容器网络模型	选择“容器隧道网络”。
	虚拟私有云	选择已有的虚拟私有云，如果列表中没有合适的选项，单击“新建虚拟私有云”完成创建。
	子网	选择已有的子网，如果列表中没有合适的选项，单击“新建子网”完成创建。
	容器网段	勾选“自动设置网段”。

表 6-3 节点配置

配置分类	配置项	配置建议
计算配置	计费模式	选择“按需计费”。
	节点类型	选择“弹性云服务器-虚拟机”。
	节点规格	选择2核8G及以上规格即可。
	操作系统	选择公共镜像中的Euler镜像。
	节点名称	输入自定义名称。
	登录方式	选择“密码”。
	密码	输入自定义密码。
网络配置	节点IP	选择“自动分配”。
	弹性公网IP	选择“自动创建”。

（可选）调整 yaml 文件配置

如果CCE集群版本高于v1.15（不包括v1.15），需调整代码仓库中的yaml文件，使其适配CCE集群版本。

步骤1 进入“凤凰商城”项目，单击导航“代码 > 代码托管”，选择代码仓库“phoenix-sample”。

步骤2 更新文件“kompose/db-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到“spec”代码段，添加以下代码行。

```
selector:  
  matchLabels:  
    io.kompose.service: db
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

图 6-1 更新文件

```
1  apiVersion: apps/v1  
2  kind: Deployment  
3  metadata:  
4    annotations:  
5      kompose.cmd: kompose convert --file docker-stack-k8s.yml  
6      kompose.version: 1.11.0 (39ad614)  
7    creationTimestamp: null  
8    labels:  
9      io.kompose.service: db  
10   name: db  
11  spec:  
12   replicas: 1  
13   selector:  
14     matchLabels:  
15       io.kompose.service: db  
16   strategy:  
17     type: Recreate  
18   template:  
19     metadata:  
20       creationTimestamp: null  
21       labels:  
22         io.kompose.service: db  
23     spec:  
24       containers:  
25         - image: docker-server/docker-org/postgres:9.4  
26           name: db  
27           command: [ "/bin/bash", "-c", "--" ]  
28           args: [ "while true; do sleep 30; done;" ]  
29           restartPolicy: Always  
30           imagePullSecrets:  
31             - name: default-secret  
32           restartPolicy: Always  
33   status: {}
```

步骤3 更新文件“kompose/redis-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”，在代码段中添加以下代码行。

```
selector:  
  matchLabels:  
    io.kompose.service: redis
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤4 更新文件“kompose/result-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”代码段，添加以下代码行。

```
selector:  
  matchLabels:  
    io.kompose.service: result
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤5 更新文件“kompose/vote-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次出现的“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: vote
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤6 更新文件“kompose/worker-deployment.yaml”。

- 将第1行中“extensions/v1beta1”修改为“apps/v1”。
- 找到文件中第一次的“spec”代码段，添加以下代码行。

```
selector:
  matchLabels:
    io.kompose.service: worker
```

- 找到“imagePullSecrets”代码段，将“regcred”修改为“default-secret”。

步骤7 单击导航“持续交付 > 编译构建”，执行任务“phoenix-sample-ci”。

----结束

配置并执行应用

步骤1 配置应用。

1. 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，页面中显示样例项目内置的应用。
2. 找到应用“phoenix-cd-cce”。单击图标**，选择“编辑”。
3. 选择“部署步骤”页签，在每个步骤中完成以下配置。

表 6-4 配置部署步骤

配置项	配置建议
集群名称	选择在购买云容器引擎时设置的集群名称。
命名空间名称	本文中選擇“default”。

4. 选择“参数设置”页签，配置以下参数。

表 6-5 参数设置

参数名称	参数值
ci_task_name	输入“phoenix-sample-ci”。
version	输入任务“phoenix-sample-ci”的参数“version”值。


5. 单击“保存”，完成应用的编辑。


步骤2 单击导航“控制台”，通过服务列表进入云容器引擎服务。

找到目标集群，单击集群名称进入总览页。

在导航中单击“工作负载”，选择“无状态负载”页签，确认列表中无记录。

如果列表中有记录，则勾选全部记录，单击“批量删除”，并勾选全部资源释放选项，单击“是”，将列表记录清空。

步骤3 返回应用列表页面，单击应用“phoenix-cd-cce”所在行的 ，在弹框中单击“确定”，启动部署。

当页面中显示  时，表示部署成功。如果部署失败，请根据失败步骤信息与日志中的报错信息排查。

步骤4 验证部署结果。

1. 进入云容器引擎服务。
2. 找到目标集群，单击集群名称进入总览页，选择“无状态负载”页签。
页面中显示5条记录，状态均为“运行中”。
3. 单击“vote”进入详情页，在“访问方式”页签中单击“更多 > 更新”。
参照表6-6配置参数，单击“确定”。

表 6-6 更新服务

参数名称	参数值
服务亲和	选择“集群级别”。
负载均衡器	<ul style="list-style-type: none">- 选择“共享型 > 自动创建”。- 实例名称：输入“phoenix”。- 弹性公网IP：选择“自动创建”。 <p>说明 如果账号下已有负载均衡器，可选择“共享型 > 使用已有”，并选择已存在的负载均衡器名称。</p>
端口配置	<ul style="list-style-type: none">- 容器端口：80- 服务端口：5000



4. 更新成功，返回列表中，当列表中显示  phoenix  时，鼠标悬停在该负载均衡器名称处，在弹窗中复制公网地址。

图 6-2 复制访问地址



5. 打开新的浏览器页面，在地址栏中输入“http://IP:5000”（其中，IP为[步骤4.4](#)记录的公网地址），页面中显示商城主页。
6. 返回“无状态负载”页面，参照[步骤4.3](#)更新“result”（其中，负载均衡器选择[步骤4.3](#)已创建的“phoenix”，服务端口输入“5001”）。
创建成功后，在新的浏览器页面中输入“http://IP:5001”，页面中显示商城仪表盘。

----结束

释放资源

为了避免不必要的费用产生，完成实践后可参考[删除集群](#)删除CCE集群。
若完成部署后还需要创建流水线，可在完成流水线实践后再删除集群。

6.3 部署应用至 ECS

购买并配置 ECS

本节使用的是ECS，您也可以使用自己的Linux主机（Ubuntu 16.04操作系统）。

步骤1 购买弹性云服务器。

购买时的必要配置参照下表，表中未列出的配置可根据实际情况选择。

表 6-7 弹性云服务器购买配置

配置分类	配置项	配置建议
基础配置	计费模式	选择“按需计费”。
	CPU架构	选择“x86计算”。
	规格	选择2核8G或以上规格。
	镜像	选择“公共镜像 > Ubuntu > Ubuntu 16.04 Server 64bit”。

配置分类	配置项	配置建议
网络配置	弹性公网IP	选择“现在购买”。
	公网带宽	选择“按带宽计费”。
高级配置	登录凭证	选择“密码”。
	密码	输入自定义密码。

步骤2 配置安全组规则。

样例项目的验证需要用到端口5000与5001，因此添加一条允许访问5000以及5001端口的入方向规则。

操作步骤如下：

1. 登录ECS页面，在列表中找到步骤[步骤1](#)中购买的ECS，单击服务器名称。
2. 选择“安全组”页签，参考[配置安全组规则](#)添加一条协议为TCP、端口为5000-5001的入方向规则。

---结束

添加目标主机至项目

部署应用到ECS之前，需要先将目标主机添加到项目基础资源中。

步骤1 进入“凤凰商城”项目，单击导航栏“设置 > 通用设置 > 基础资源管理”。

步骤2 单击“新建主机集群”，配置以下信息，单击“保存”。

表 6-8 新建主机集群

配置项	配置建议
集群名称	输入“host-group”。
操作系统	选择“Linux”。
主机联通方式	选择“直连模式”。
执行主机	选择“官方资源池”。

步骤3 单击“添加或导入主机”，配置以下信息，单击“确定”。

表 6-9 添加主机

配置项	配置建议
选择添加方式	选择“通过IP手动添加”。
主机名	建议与在购买ECS时设置的名称保持一致。
IP	输入在购买ECS时生成的弹性IP。

配置项	配置建议
认证方式	选择“密码”。
用户名	输入“root”。
密码	输入在购买ECS时设置的密码。
ssh端口	输入“22”。

步骤4 页面显示一条主机记录，当“连通性验证”列的值显示为“成功”，表示主机添加完成。

如果主机添加失败，请根据失败详情排查主机配置。

---结束

在 ECS 中安装依赖工具

样例程序的运行需要Docker及Docker-Compose环境，需要将依赖环境安装到目标ECS中。

步骤1 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，在列表中找到应用“phoenix-sample-predeploy”。

步骤2 单击**，在下列表中选择“编辑”，进入编辑页面。

步骤3 选择“环境管理”页签，配置主机环境。

1. 单击“新建环境”，配置以下信息，单击“保存”。

表 6-10 新建环境

配置项	值
环境名称	输入“phoenix-env”。
资源类型	选择“主机”。
操作系统	选择“Linux”。

📖 说明

若无新建环境权限，请联系管理员通过应用的“权限管理”页面添加权限。

2. 单击“导入主机”，在弹框中选择已配置好的主机集群与主机，单击“导入”。

3. 页面提示导入成功，关闭此窗口。

步骤4 在“部署步骤”页签，编辑应用的步骤。

在步骤“安装Docker”中，在环境下拉列表中选择“phoenix-env”。如果页面显示弹框“是否将后续步骤的环境也修改为phoenix-env？”，单击“确定”。

步骤5 单击“保存并部署”，启动部署任务。

当出现页面提示“部署成功”时，表示任务执行成功。

步骤6 登录弹性云服务器，检测依赖工具是否安装成功。

- 执行命令 `docker -v`，查看 Docker 镜像版本。
- 执行命令 `docker-compose -v`，查看 Docker-Compose 版本。

如果出现类似图6-3所示的回显，说明安装成功。

图 6-3 查看 Docker 及 Docker-Compose 版本

```
root@ecs-he2e:~# docker -v
Docker version 18.09.0, build 4d60db4
root@ecs-he2e:~# docker-compose -v
docker-compose version 1.17.1, build 6d101fb
root@ecs-he2e:~#
```

----结束

配置并执行应用

部署时需要将 ECS 配置在应用的环境列表中，并将构建任务“phoenix-sample-ci”设置为部署来源。

步骤1 进入“凤凰商城”项目，单击导航“持续交付 > 部署”，在列表中找到应用“phoenix-sample-standalone”。

步骤2 单击“**”，在下列表中选择“编辑”，进入编辑页面。

步骤3 选择“环境管理”页签，配置主机环境。

1. 单击“新建环境”，配置以下信息，单击“保存”。

表 6-11 新建环境

配置项	值
环境名称	输入“phoenix-env”。
资源类型	选择“主机”。
操作系统	选择“Linux”。

2. 单击“导入主机”，在弹框中选择已配置好的主机集群与主机，单击“导入”。
3. 页面提示导入成功，关闭此窗口。

步骤4 在“部署步骤”页签，编辑应用的步骤。

1. 在步骤“选择部署来源”中，参照表6-12设置部署来源。

表 6-12 部署来源配置

配置项	配置建议
选择源类型	选择“构建任务”。
请选择构建任务	选择“phoenix-sample-ci”。

配置项	配置建议
环境	选择“phoenix-env”。 如果页面显示弹框“是否将后续步骤的环境也修改为phoenix-env? ”，单击“确定”。

2. 最后两个步骤“解压文件”与“执行shell命令”保持默认配置即可。

步骤5 选择“参数设置”页签，根据容器镜像服务登录指令填写参数。

登录指令通过控制台获取。

步骤6 单击“保存并部署”，启动部署。

当页面显示“部署成功”时，表示部署成功。如果部署失败，请根据失败步骤信息与日志中的报错信息排查。

步骤7 验证部署结果。

1. 打开浏览器，输入“http://IP:5000”，其中IP为ECS的弹性公网IP地址，页面中显示商城主页。
2. 输入“http://IP:5001”，其中ECS的弹性公网IP地址，页面中显示商城仪表盘。

----结束

释放资源

为了避免不必要的费用产生，完成实践后可参考[如何删除、重启弹性云服务器?](#) 删除ECS。

若完成部署后还需要创建流水线，可在完成流水线实践后再删除ECS。