



区块链服务

最佳实践

文档版本 01

发布日期 2018-03-30

华为技术有限公司



版权所有 © 华为技术有限公司 2018。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默示的声明或保证。

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：<http://www.huawei.com>

客户服务邮箱：support@huawei.com

客户服务电话：4008302118

目 录

1 快速部署区块链服务	1
1.1 简介	1
1.2 环境准备	1
1.3 订购及部署	2
2 链代码开发及可视化管理	5
2.1 简介	5
2.2 链代码开发	5
2.3 可视化管理链代码	7
3 通过 CCE 部署的线上应用对接链代码	10
3.1 简介	10
3.2 应用程序开发配置和构建	10
3.3 应用程序线上部署	12
3.4 访问应用并调试业务	18

1 快速部署区块链服务

- 1.1 简介
- 1.2 环境准备
- 1.3 订购及部署

1.1 简介

本系列最佳实践是以基于区块链身份共享的银行II类账户跨行开户为例，介绍华为云区块链服务的使用。

关于示例

- 业务场景：
基于A银行的I类账户以及已有的KYC信息背书，免KYC过程开通另外一个B银行的II类账户。
- 主要诉求：
用户身份等信息需要加密，避免暴力破解；提供基于身份信息的快速查询；
- 关键挑战：
 - 安全隐私：银行不希望把客户隐私信息泄露给其它银行。
 - 高性能检索：提供类传统数据库的检索性能(毫秒级响应)，保持一致用户体验。
 - 良好接口：提供简单方便的接口，便于应用开发。

关于本文

本文介绍如何在华为云上为以上示例业务快速部署一个区块链服务。

在名为“testwangole9i”的集群中创建部署一个名为“test”的区块链服务，包含三个节点组织，并使用快速拜占庭容错共识算法。

1.2 环境准备

由于华为云区块链服务是基于容器所构建的集群进行部署的，区块链服务中的Peer节点、共识节点以及链代码均需要运行在容器中，同时为使外网能够正常访问，且保证

数据不丢失，在开始部署区块链服务之前，需要先完成相应的环境准备工作，包括：创建容器集群、绑定弹性IP、创建网络存储。

创建集群

区块链服务是基于容器所构建的集群进行部署的。首先需要在CCE版本中创建好虚拟机集群。

如何创建虚拟机集群，请参考[创建虚拟机集群](#)。本实践中我们已创建好名为“testwangole9i”的集群。



注意

- 创建集群时请务必在CCE服务（建议使用CCE 1.9及以上版本）页面中创建，不能在ServerStage服务内创建。
- 创建集群的时候可一并完成绑定弹性IP。您需要给集群中每个虚拟机分别绑定一个弹性IP。
- 创建集群时，您需要最少添加2个节点且每个节点规格为8U16G及以上，否则可能会导致服务部署失败。

绑定弹性 IP

若您创建集群时未绑定弹性IP，则可以通过申请弹性IP并将弹性IP绑定到弹性云服务器上，实现弹性云服务器访公网的目的。

如何绑定弹性IP，请参见[绑定弹性IP](#)。

创建网络存储

使用网络存储来保存区块链服务的相关数据。如何创建网络存储，请参见CCE中的[创建文件存储](#)。

1.3 订购及部署

订购区块链服务时需要您为区块链服务配置基本参数和网络节点，以便快速完成区块链服务的创建及部署。

前提条件

已完成环境准备。

操作步骤

步骤1 登录区块链服务管理控制台。

步骤2 单击页面右上角的“购买区块链服务”。

 购买区块链服务

步骤3 在购买页面填写相关参数，见下表参数值示例。

表 1-1 参数列表

参数类型	参数名称	参数值示例	说明
基本信息	区块链服务名称	test	支持中英文字符及数字，长度为4-24个字符。
	区块链类型	私有链	可选择创建私有链或者联盟链。 私有链：仅本租户内部使用的区块链服务。 联盟链：可以邀请其他租户一起参与的区块链服务。
	容器集群	testwangole9i	选择区块链服务部署在哪个容器集群上。一个集群容器只允许部署一个区块链服务。此处我们选择在环境准备环节创建的集群。
	网络存储	-	选择已创建好的网络存储来保存区块链服务的相关数据。
	节点组织	添加三个节点组织，分别为： xxx1、节点数量为2，xxx2、节点数量为2，xxx3、节点数量为2。	为区块链服务添加Peer节点组织，每个区块链服务最少添加一个，最多添加五个。每个节点组织的节点数量为1-10，即每个节点组织的实例数。
	共识策略	快速拜占庭容错共识算法	提供的共识策略有：测试策略、快速拜占庭容错共识算法、Kafka(CFT)。
	安全机制	ECDSA	支持2种安全加密机制，包括ECDSA、国密算法。
	版本信息	1.6.0	版本。
	链代码管理初始密码	Test1234_	设置登录链代码管理页面的初始密码。
	确认密码	Test1234_	确认密码。
	共识节点数量	4	即orderer节点的数量。
通道配置	添加通道	添加名为“testchannel”的通道，并将刚才创建的三个节点组织添加进此通道。	点击“添加通道”为区块链服务添加通道，用于节点组织之间的通信。完成后再为每个通道添加节点组织。 说明 如果您在部署服务时未配置通道，可在 通道管理 页面创建通道并进行通道与服务的绑定。

参数类型	参数名称	参数值示例	说明
-	购买量	1个月	BCS计费方式为包周期方式。 根据需要选择购买量。

步骤4 配置完成后，单击“立即购买”。完成购买后，进入服务汇总列表。

查看组织状态若为“未知”，请等待1-2分钟后刷新，等待状态变为“运行”后，表示区块链服务部署完成。

----结束

2 链代码开发及可视化管理

2.1 简介

2.2 链代码开发

2.3 可视化管理链代码

2.1 简介

链代码也称为智能合约，实质上是控制区块链网络中的不同实体或相关方如何相互交互或交易的业务逻辑。

本文以基于区块链身份共享的银行II类账户跨行开户的链代码为例，讲解如何开发链代码，以及链代码的安装和实例化。

2.2 链代码开发

链代码将业务网络交易封装在代码中，最终在一个 Docker 容器内运行。目前华为云区块链服务暂时支持Golang语言编写代码。链代码开发包括如下几个步骤：

操作步骤

步骤1 将shim包导入您的链代码中。



说明

shim包提供了一些 API，以便您的链代码与底层区块链网络交互来访问状态变量、交易上下文、调用方证书和属性，并调用其他链代码和执行其他操作。

示例如下图：

```
package main

import (
    "fmt"
    "crypto/sha256"
    "github.com/hyperledger/fabric/core/chaincode/shim"
    sc "github.com/hyperledger/fabric/protos/peer"
)
```


步骤2 编写main函数。**说明**

任何 Go 程序的起点都是 main 函数，因此该函数被用于引导/启动链代码。当对等节点部署其链代码实例时，就会执行 main 函数。

示例如下图：

```
// The main function is only relevant in unit test mode. Only included here for completeness.
func main() {

    // Create a new Smart Contract
    err := shim.Start(new(SmartContract))
    if err != nil {
        fmt.Printf("Error creating new Smart Contract: %s", err)
    }
}
```

步骤3 实现Init方法。**说明**

Init方法在链代码首次部署到区块链网络时调用，将由部署自己的链代码实例的每个对等节点执行。此方法可用于任何与初始化、引导或设置相关的任务。

示例如下图：

```
func (s *SmartContract) Init(APIstub shim.ChaincodeStubInterface) shim.Response {
    fmt.Println("Initing chaincode")
    _, args := APIstub.GetFunctionAndParameters()

    if len(args) > 10 {
        return shim.Error("Incorrect number of arguments. Expecting 1")
    }

    return shim.Success(nil)
}
```

步骤4 实现Invoke方法。**说明**

只要修改区块链的状态，就会调用 Invoke 方法。简言之，所有创建、更新和删除操作都应封装在 Invoke 方法内。因为此方法将修改区块链的状态，所以区块链 Fabric 代码会自动创建一个交易上下文，以便此方法在其中执行。对此方法的所有调用都会在区块链上记录为交易，这些交易最终被写入区块中。

示例如下图：

```
func (s *SmartContract) Invoke(APIstub shim.ChaincodeStubInterface) shim.Response {

    // Retrieve the requested Smart Contract function and arguments
    function, args := APIstub.GetFunctionAndParameters()

    // Route to the appropriate handler function to interact with the ledger appropriately
    if function == "creditAccountInfo" {
        return s.creditAccountInfo(APIstub, args)
    } else if function == "authAccount" {
        return s.authAccount(APIstub, args)
    }

    return shim.Error("Invalid Smart Contract function name.")
}
```

链代码开发的更多信息，可参阅hyperledger-fabric的[链代码教程](#)。

----结束

示例代码

这里提供完整的示例链代码文件[fabbankid.go](#)，您可直接使用体验。更多内容可参考[区块链服务开发指南](#)。

2.3 可视化管理链代码

链代码安装在peer节点上，然后在通道上进行实例化。所有通道成员都需要在将运行此链代码的每个peer节点上安装链代码，且只需在一个peer节点上进行链代码实例化。如需使用相同的链代码，通道成员必须在链代码安装期间为链代码提供相同的名称和版本。如下我们以fabbankid.go链代码为例介绍如何可视化安装及实例化。

安装链代码

步骤1 登录区块链服务管理控制台。

步骤2 单击区块链服务列表“操作”列的“链代码管理”。

步骤3 跳转至链代码管理登录页面后，输入用户名及密码登录系统。



用户名：admin，初始登录密码为您在部署区块链服务时设置的密码。

步骤4 在右上角过滤框中选择xxx1组织的peer-0节点，单击“安装链代码”，进入链代码安装配置界面，输入如下信息后，单击“确认”。

安装链代码

- * 链代码名称: fabbank
- * 链代码版本: 1.0
- * 链代码SHA256摘要 @: b72d5c9b1df48e7e77246124a07c8827bd92
- * 链代码文件: 添加文件 fabbankid.zip

确认 取消



- SHA256摘要可在linux环境中执行命令: sha256sum fabbankid.zip后生成。
- 链代码zip包可点击如下链接下载，[fabbankid.zip](#)。

安装成功，如下截图。



----结束

实例化链代码

步骤1 在链代码管理界面，单击操作列的“实例化”，进入链代码实例化配置界面，配置信息如下图。



说明

链代码函数名和参数与链代码强相关，需谨慎填写

步骤2 单击“确定”，开始实例化，可能持续1-2分钟。

实例化成功，如下截图。



----结束

安装其他组织链代码

上文我们详细介绍了xxx1组织的链代码安装及实例化。由于我们创建的区块链服务中包含三个组织，需要对其余的xxx2和xxx3组织的peer-0节点分别进行链代码安装。

操作步骤类似，这里不再介绍，请参考上文。

3 通过 CCE 部署的线上应用对接链代码

3.1 简介

3.2 应用程序开发配置和构建

3.3 应用程序线上部署

3.4 访问应用并调试业务

3.1 简介

本文以基于区块链身份共享的银行II类账户跨行开户为例，介绍应用程序如何对接链代码并通过CCE部署上线，最终完成银行II类账户开户业务。

3.2 应用程序开发配置和构建

您可以使用 Go 开发应用程序，并利用 [Hyperledger Fabric SDK Go](#) 中的可用 API 来调用链代码，以在区块链网络中完成事务处理。

服务组件证书配置

目前支持两种证书：管理员证书和用户证书。创建通道、加入通道、更新通道、安装链代码、实例化链代码、升级链代码和删除链代码需要使用管理员证书，交易和查询推荐使用用户证书。

下载证书：

应用程序开发人员需要到区块链服务管理页面下载对应服务的证书，如下图：

区块链服务管理

删除 所有状态 (4) 输入

服务名称	服务状态	容器集群	共识策略	创建时间	操作
fortest	运行	fortest	测试策略(SOLO)	2018-03-06T11:21:25+08:00	更新版本 输入弹性IP地址
test40	运行	testbiling	测试策略(SOLO)	2018-02-24T20:14:43+08:00	更新版本 输入弹性IP地址
test	运行	testwangole9i	快速拜占庭容错共识算法	2018-03-06T15:59:43+08:00	更新版本 链代码管理

组织名称	组织状态	组织类型	实例数量	操作
test-orderer	运行	共识	4	下载管理员证书
xxx1	运行	节点	2	下载管理员证书 下载用户证书
xxx2	运行	节点	2	下载管理员证书 下载用户证书
xxx3	运行	节点	2	下载管理员证书 下载用户证书

构建证书目录:

建议下载证书前预先构建好目录结构。

bank-account-demo/api-server/src/api-server/conf/crypto-config 的结构, 如下:

crypto-config

```

├── ordererOrganizations
|   ├── users
├── peerOrganizations
|   ├── users

```

Orderer 和 Peers 对应证书解压放置到对应的目录中, 如: orderer 证书解压放置在 ordererOrganization/users 下 ordererOrganization/users/Admin@6193bdab9fd94f51e453f65cbf86e75b8e697e10.orderer-6193bdab9fd94f51e453f65cbf86e75b8e697e10.default.svc.cluster.local, peer证书解压放在 peerOrganizations/users 下 peerOrganizations/users/User1@d459afc8331c61db9ecd0b38099873dbd3b4c402.peer-d459afc8331c61db9ecd0b38099873dbd3b4c402.default.svc.cluster.local, 多个peer证书要放在同级目录 peerOrganizations/users 下。如果只进行交易只需要下载 orderer 管理员证书和各Peer组织的用户证书即可。

用户构建的应用程序需要各角色证书与 Fabric 组件进行通信, 需要规划好crypto-config 文件路径, 并将各路径信息同步到 SDK 配置文件, 以bank-account-demo为例, 编译时我们默认将证书拷贝放置在/opt/gopath/src/github.com/hyperledger/api-server/conf/crypto-config。

Fabric SDK 配置

Fabric SDK 下载完成后, 需按照fabric-sdk-go-master/test/fixtures/config/config_test.yaml 的配置要求配置通道、组织、各角色 MSP 路径和 TLS 证书路径。可参考 bank-account-demo/api-server/src/api-server/conf/bohai.yaml。

示例配置:

config_test.yaml

bohai.yaml

应用程序开发

用户开发人员可在自己的应用程序中通过 Fabric 的 API 来调用链代码，执行事务处理。可参考 bank-account-demo/api-server/src/api-server/controllers/transaction/transaction.go。

示例代码：[transaction.go](#)

如下为“基于区块链身份共享的银行II类账户跨行开户”示例的服务端及前端程序包，供您下载使用。

[api-server.rar](#) [portal.rar](#)

应用程序配置

配置文件 bank-account-demo/api-server/src/api-server/conf/app.conf 中链代码（参数为 chaincode_id）及通道名称（参数为 channel_id）的值，需与安装链代码时一致。

配置示例：[app.conf](#)

应用程序构建

应用服务端构建：

进入 bank-account-demo/api-server/build 目录执行命令：bash -x build.sh，生成应用程序服务端镜像 api-server.tar.gz。如下

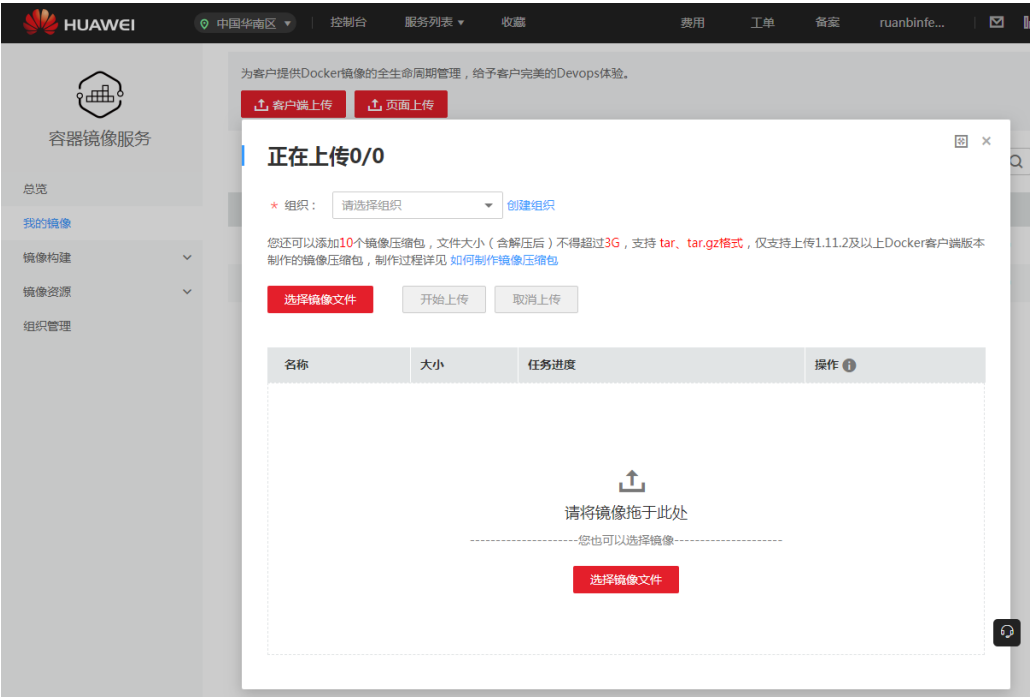
```
root@HGH1000050619:~/bank-account-demo/api-server/build# bash -x build.sh
root@HGH1000050619:~/bank-account-demo/api-server/build# ls ../release/
api-server  api-server.tar.gz
```

3.3 应用程序线上部署

部署服务端

上传镜像：

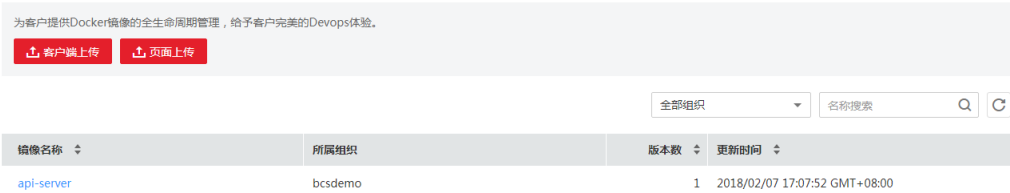
步骤1 进入[软件仓库](#)，点击“页面上传”，弹出对话框如下：



步骤2 当没有组织时需要创建组织，已存在则选择组织，点击“选择镜像文件”选择 `api-server.tar.gz`，然后点击“上传”。



上传镜像成功，如图：



----结束

部署应用：

步骤1 进入[应用管理](#)，选择自己所在的 CCE 集群，并开始创建应用。



步骤2 填写基本信息

填写基本信息 添加容器 应用访问设置（可选） 高级设置（可选）

1 2 3 4

* 应用名称: demo-chaincode-api-server

* 容器集群: testwangole9i

应用组: 默认

* 实例数量: 2

应用描述: 请输入描述信息

步骤3 添加容器

选择镜像并设置启动参数



The screenshot shows the configuration page for a container named 'demo-chaincode-api-server-con'. The image is 'api-server' and the version is 'latest'. The container specification is set to '1X' (0.25 GiB memory, 1 Core CPU). Under '高级设置' (Advanced Settings), the '生命周期' (Lifecycle) tab is active, showing the '启动' (Start) button and the start command: '/opt/gopath/src/github.com/hyperledger/api-server/api-server'.

容器名称: demo-chaincode-api-server-container

启动命令: /opt/gopath/src/github.com/hyperledger/api-server/api-server

步骤4 应用访问设置

The screenshot shows the '添加访问方式' (Add Access Method) dialog box. The internal access domain is 'demo-chaincode-api-ser', and the access method is '外部访问' (External Access). The access type is '弹性IP' (Elastic IP), the container port is '8080', the access port is '指定端口' (Specified Port) '31432', and the protocol is 'TCP'.

- 内部访问域名: demo-chaincode-api-server
- 访问方式: 外部访问
- 访问类型: 弹性IP
- 容器端口: 8080
- 指定端口: 31432
- 协议: TCP

步骤5 高级设置



为调试方便可选择节点亲和，配置完成之后，点击“创建”，成功后返回应用列表。

----结束

部署前端

上传镜像：

上传应用程序前端镜像demo-chaincode-portal.tar.gz，可参考api-server.tar.gz的上传过程。

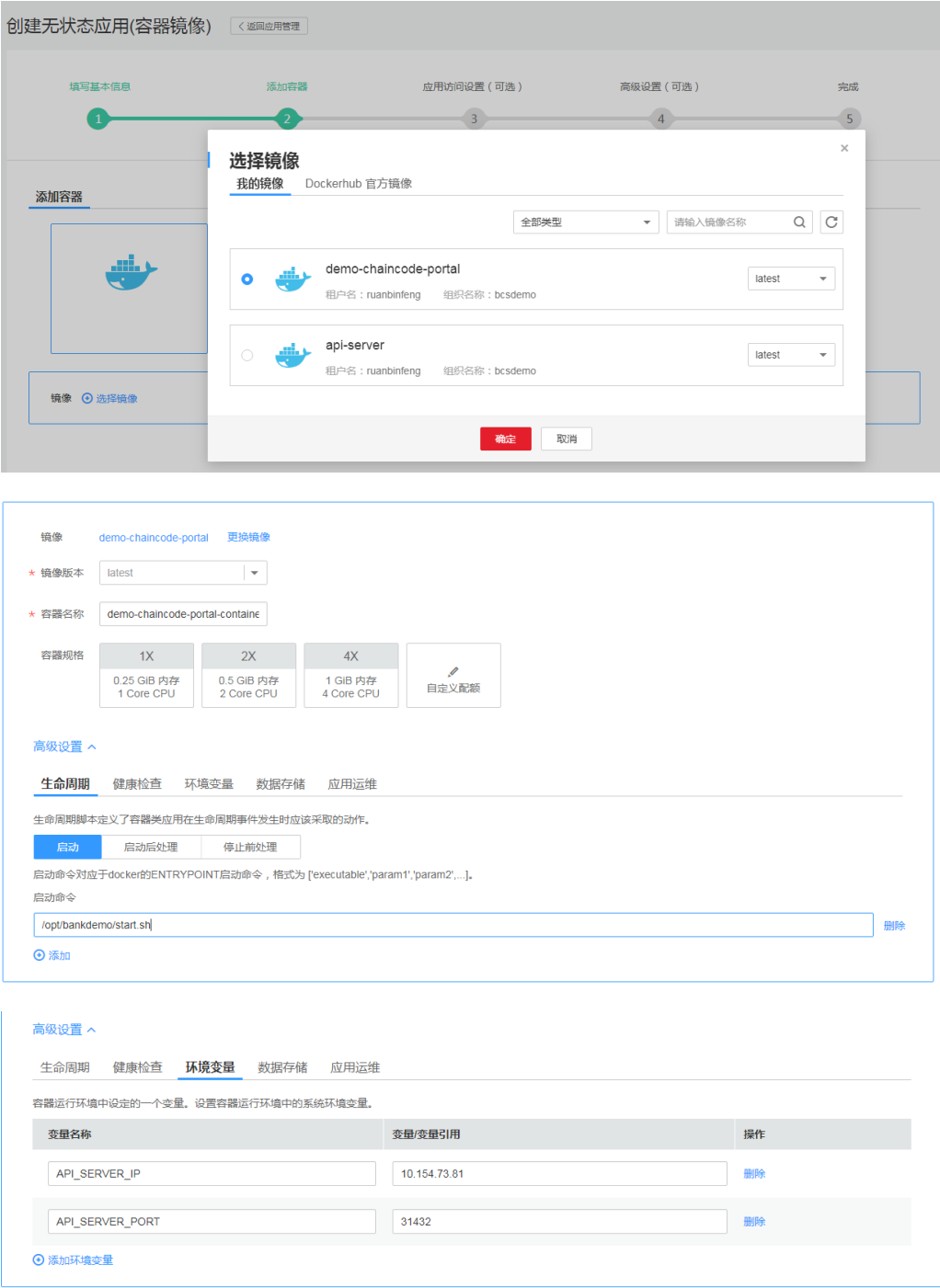
部署应用：

步骤1 填写基本信息



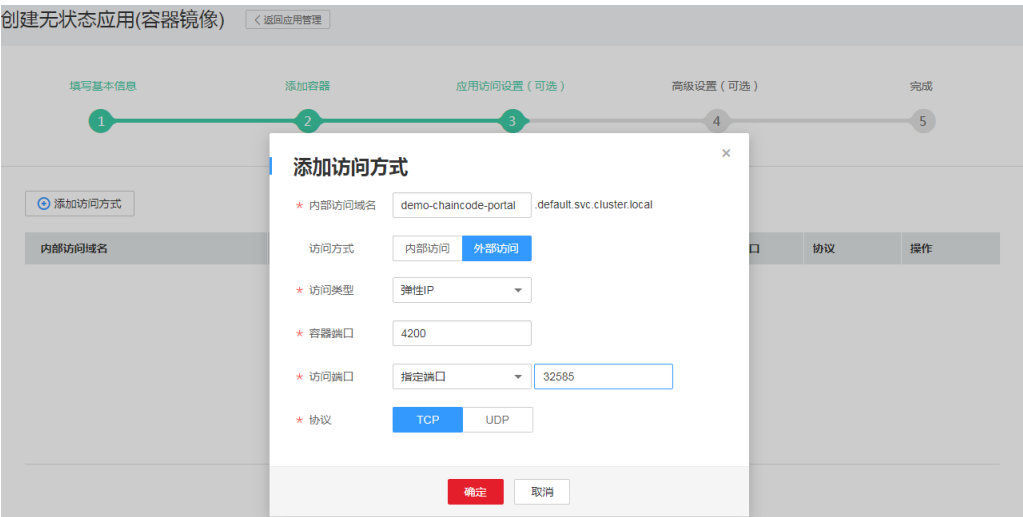
步骤2 添加容器

选择镜像并设置启动参数及环境变量



IP需要设置为集群 EIP， 端口设置为应用程序服务端端口

步骤3 应用访问设置



步骤4 高级设置

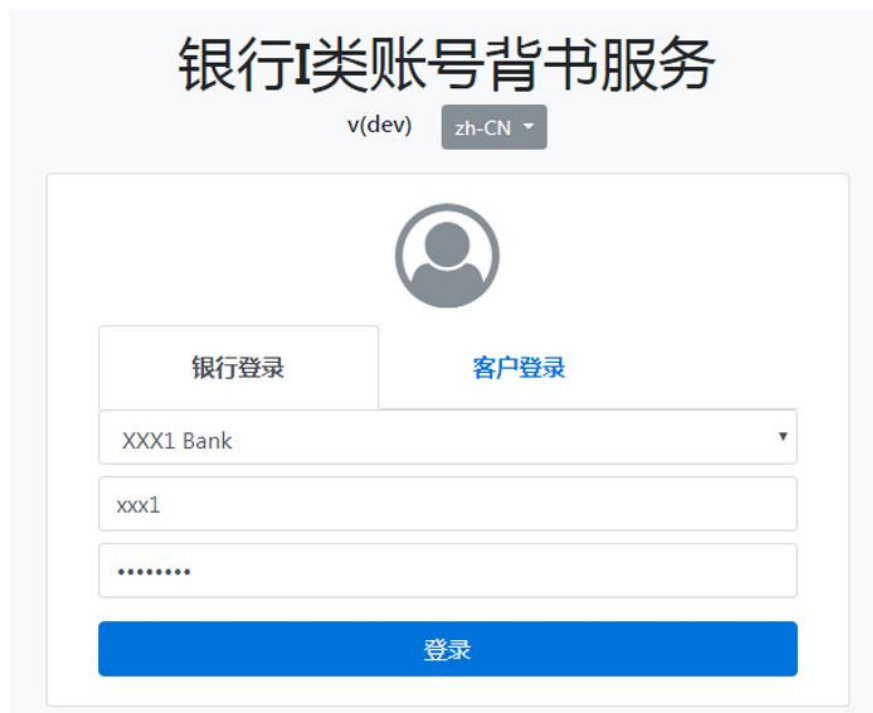


为调试方便可选择节点亲和，配置完成之后，点击“创建”，成功后返回应用列表。
----结束

3.4 访问应用并调试业务

生成客户信息

demo-chaincode-api-server 和 demo-chaincode-portal 成功运行后，通过 <http://EIP:nodePort>（需更改 EIP 及 portal 外网访问端口）访问 portal。示例如下：



The image shows a web interface for '银行I类账号背书服务' (Bank I-type account verification service). At the top, there's a title and a version/language selector showing 'v(dev)' and 'zh-CN'. Below is a user profile icon. There are two tabs: '银行登录' (Bank Login) and '客户登录' (Customer Login). Under '银行登录', there's a dropdown menu showing 'XXX1 Bank', a text input for 'xxx1', a password input with dots, and a blue '登录' (Login) button.

选择银行登录模式，银行：XXX1 Bank，用户名：xxx1，密码：password。登录成功后，上传客户信息文件**test.csv**，生成客户信息。

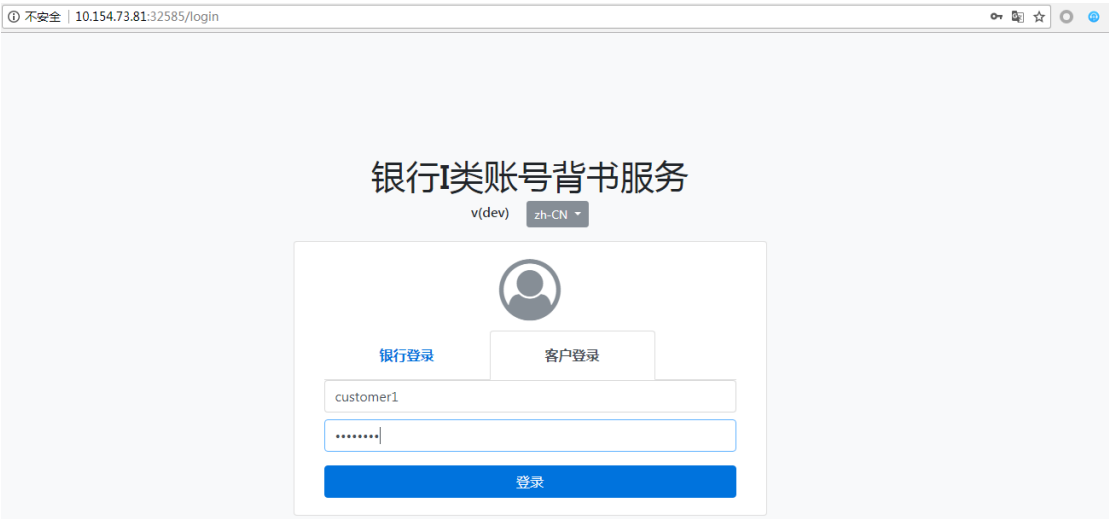


The image shows a web interface for uploading a CSV file. At the top, there's a header '银行I类账号背书服务' and a sub-header '快捷实操演示银行'. Below is a section for 'XXX1 Bank'. On the left, there's a box titled '请上传客户列表' (Please upload customer list) with a blue '上传CSV文件' (Upload CSV file) button and a dashed box containing 'test.csv'. Below this are '清除' (Clear) and '提交' (Submit) buttons. On the right, there's a table titled '请上传CSV文件中包含以下信息' (Please upload CSV file containing the following information) with the following fields:

用户名 -->	user_name
用户身份证 # -->	user_id_card#
用户帐号 -->	user_account
手机号码 -->	mobile_number

申请 II 类账户

退出系统后重新登录，选择客户登录模式，用户名：customer1，密码：password。



登录成功后，申请二类账户，输入信息后，点击“提交”，申请成功。

