

搭建微信公众号后台

消息回复

实验指导手册

版本:2.5



华为技术有限公司

目录

背景介绍	3
实验目的	3
注意事项	3
1 实验简介.....	4
1.1 实验介绍	4
1.2 实验架构	4
1.3 实验流程	5
2 实验环境准备	6
2.1 注册微信公众号.....	6
2.2 进入控制台（华为云）	10
2.3 创建虚拟私有云 VPC.....	11
2.4 创建弹性云服务器 ECS	14
3 安装基础软件	19
3.1 登陆弹性云服务器.....	19
3.2 升级默认 Python 版本.....	20
3.3 升级默认 pip 版本	23
3.4 安装 Web.py 框架	24
3.5 安装 WinSCP	24
4 上传代码&服务启动	26
4.1 本地编写代码文件.....	26
4.1.1 新建 main.py 文件	26
4.1.2 新建 handle.py 文件	27
4.1.3 新建 receive.py 文件.....	30
4.1.4 新建 templates 文件夹.....	32
4.2 WinSCP 上传代码文件	33
4.3 启动对应服务	34
5 启用开发者模式&实验验证	35



5.1 启用开发者模式.....	35
5.2 微信公众号验证实验结果.....	36
6 资源释放.....	38
6.1 实验介绍.....	38
6.2 删除弹性云服务器 ECS.....	38
6.3 删除虚拟私有云.....	38

背景介绍

Cloud2.0 时代，企业更加关注应用上云，华为云强大的技术优势能为企业用户打造专业、快速、稳定、安全的项目开发、部署、运营支撑平台，并以完善快速的售后服务、端云协同的生态支撑，助力项目的实施成功。

微信公众号是开发者或商家在微信公众平台上申请的应用账号。通过公众号，商家可以在微信平台上实现和特定群体的文字、图片、语音、视频的全方位沟通、互动，形成一种主流的线上线下微信互动营销方式。

本次实验通过弹性云服务器 ECS、弹性公网 IP 搭建微信公众号处理后台，使用 Python 语言编写对应的微信消息处理逻辑代码，接收从微信服务端转发过来的消息，并返回处理结果给最终用户，体验基于华为公有云服务不一样的感受。

实验目的

通过本实验，您将能够：

- 理解弹性云服务器与镜像以及弹性公网 IP 的关系及作用
- 掌握在华为公有云服务器上软件安装下载以及版本升级的操作方法
- 掌握 WinSCP 工具的正确使用方法
- 利用弹性公网 IP 以及 token 成功对接弹性云服务器

注意事项

实验资源一旦购买就开始计费，请合理安排时间进行实验，并注意以下几点：

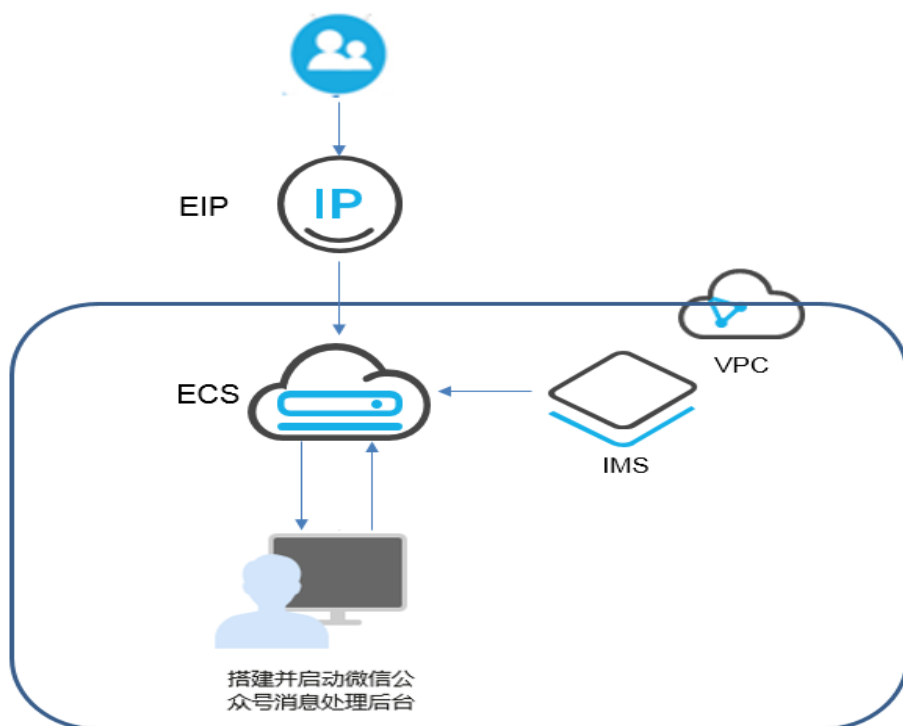
- 实验需注册微信公众号，需要审批 1-2 天，建议没有公众号的学员获取公众号后再开始实验
- 实验中所涉及到的相关软件工具，可以在实验中提供的链接中获取，也可以从以下 OBS 存储桶链接中直接获取：
(WinSCP)
<https://weirengzhengnew.obs.myhuaweicloud.com:443/weixingongzhonghao/WinSCP-5.15.1-Setup.exe>
(Putty)
<https://weirengzhengnew.obs.myhuaweicloud.com:443/weixingongzhonghao/PUTTY.rar>
- 本实验预计 2 小时完成，实验结束后请一定释放资源，并确认资源彻底删除；
- 若实验中途离开或中断，建议释放实验资源，否则将会按照购买的资源继续计费；
- 资源释放步骤请根据实验指导进行。

1 实验简介

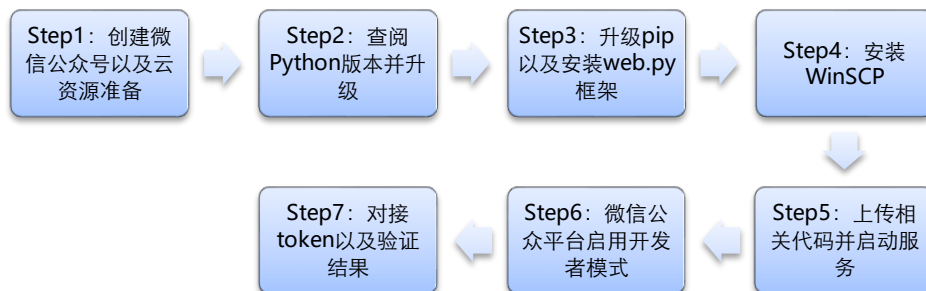
1.1 实验介绍

使用 Python+Web.py 组合完成微信公众号开发，在弹性云服务器 ECS 上部署微信公众号后台，配置开发者模式，通过弹性公网 IP 以及 token 成功对接弹性云服务器 ECS。

1.2 实验架构



1.3 实验流程



2 实验环境准备

在进行实验之前，需提前配置好以下资源，实验及资源准备均在“华北-北京四”区域下进行：

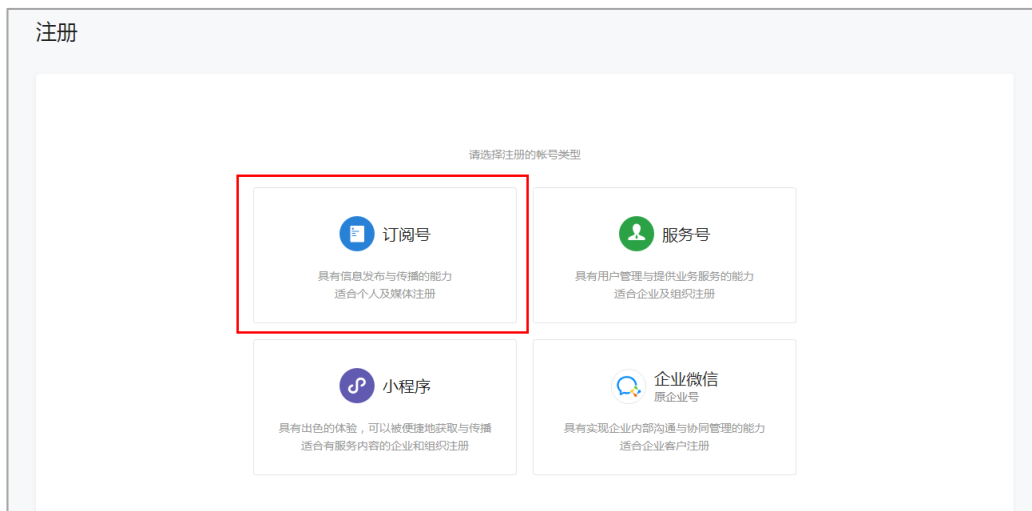
- 已成功在微信公众平台注册微信公众号(注册微信公众号审批需要 1-2 天，故为顺利完成实验，请学员提前申请好微信公众号)。
- 已成功注册华为云官网并通过认证。
- 创建虚拟私有云 VPC 及其子网，并创建安全组添加安全组规则。

2.1 注册微信公众号

步骤 1 打开浏览器，输入微信公众平台网站平台 <https://mp.weixin.qq.com/>，在上方点击“立即注册”



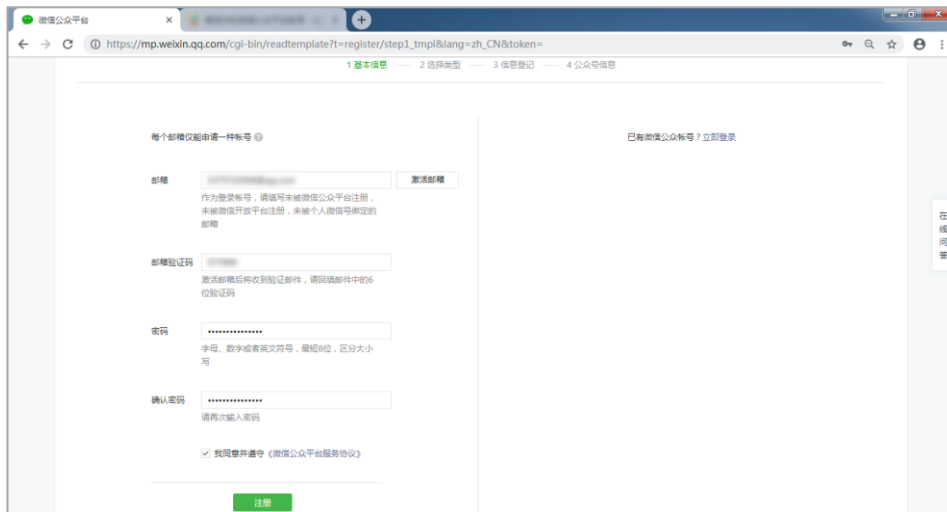
步骤 2 在弹出的注册界面中，选择“订阅号”，并点击。



步骤 3 选择“订阅号”后，按照腾讯公众号指引流程进行引导注册，先填写“基本信息”。

要求：

- 邮箱是未被微信公众平台注册，未被微信开放平台注册，未被个人微信号绑定的邮箱。
- 密码要求是字母、数字或者英文符号，最短 8 位，区分大小写。
- 密码和确认密码要求一致。
- 且必须要勾选“我同意并遵守《微信公众平台服务协议》”。



步骤 4 填写好“基本信息”之后，点击注册。来到“选择类型”界面，注册地默认即可。点击确定。在弹出的界面中，选择“订阅号”，点击“选择并继续”。

1 基本信息 — 2 选择类型 — 3 信息登记 — 4 公众号信息

请选择企业注册地，暂只支持以下国家和地区企业类型申请帐号

中国大陆

确定

1 基本信息 — 2 选择类型 — 3 信息登记 — 4 公众号信息

订阅号

为媒体和个人提供一种新的信息传播方式，构建与读者之间更好的沟通与管理模式。

适用于个人和组织

群发消息	1条/天
消息显示位置	订阅号列表
基础消息接口/自定义菜单	有
高级接口能力	无
微信支付	无
了解详情	

选择并继续 >

服务号

给企业和组织提供更强大的业务服务与用户管理能力，帮助企业快速实现全新的公众号服务平台。

不适用于个人

群发消息	4条/月
消息显示位置	会话列表
基础消息接口/自定义菜单	有
高级接口能力	有
微信支付	可申请
了解详情	

选择并继续 >

企业微信

为企业提供专业的通讯工具。丰富的办公应用与API，助力企业高效沟通与办公。

粉丝关注需验证身份且关注有上限

群发消息	无限制
消息显示位置	会话列表
基础消息接口/自定义菜单	有
高级接口能力	有
了解详情	

选择并继续 >

步骤 5 选择好之后，来到“信息登记页面”，进行对应信息的登记和选择（“主体信息”选择个人）。

微信公众平台

gh_10a629fc5270

1 基本信息 2 选择类型 3 信息登记 4 公众号信息

用户信息登记

微信公众平台致力于打造真实、合法、有效的互联网平台。为了更好的保障你和广大微信用户的合法权益，请你认真填写以下登记信息。

用户信息登记审核通过前：

1. 你可以依法享有本微信公众平台帐号所产生的权利和收益；
2. 你将对本微信公众平台帐号的所有行为承担全部法律责任；
3. 你的注册信息将在法律允许的范围内向微信用户展示；
4. 人民法院、检察院、公安机关等有权机关依法调取你的注册信息等。

个人可注册1个帐号，个体工商户、企业、其他组织可注册2个帐号，政府和媒体可注册50个帐号。
请确认你的微信公众平台主体类型属于政府、企业、其他组织或个人，并请选择对应的类别进行信息登记。
点击查看微信公众平台信息登记指引。

帐号类型 订阅号

如何选择主体类型？

政府

媒体

企业

其他组织

个人

个人类型包括：由自然人注册和运营的公众帐号。
帐号能力：个人类型暂不支持微信认证、微信支付及高级接口能力。

主体信息登记

身份证姓名

陈

信息审核通过后身份证姓名不可修改；如果名字包含分隔号“”，请勿省略。

身份证号码

33900

请输入您的身份证号码。

管理员身份验证

返回二维码

管理员信息登记

管理员手机

182

发送验证码

请输入你的手机号码，一个手机号码只能注册5个公众帐号。

短信验证码

171585

无法接收验证码？

请输入手机短信接收到的6位验证码

上一步

继续

关于腾讯 | 服务协议 | 运营中心 | 帮助中心 | 腾讯客服 | 侵权投诉 | 反馈账号weixingongzhong

Copyright © 2012-2019 Tencent. All Rights Reserved.

步骤 6 信息登记完成之后，按照指引进行下一步操作。以上四步完成之后，大家就会看到自己的公众号相关信息。

微信公众平台

cloud7168

订阅号 | 未认证

首页

功能

小程序

管理

自动回复

自定义菜单

投票管理

页面模板

赞赏功能

原创管理

添加功能插件

消息管理

用户管理

素材管理

春节期间小程序小游戏审核提醒通知

2019-01-07 更多

帐号整体情况

0 新消息

新增人数

总用户数

新建群发

开始写一篇文案的写作吧

下载订阅号助手

2.2 进入控制台（华为云）

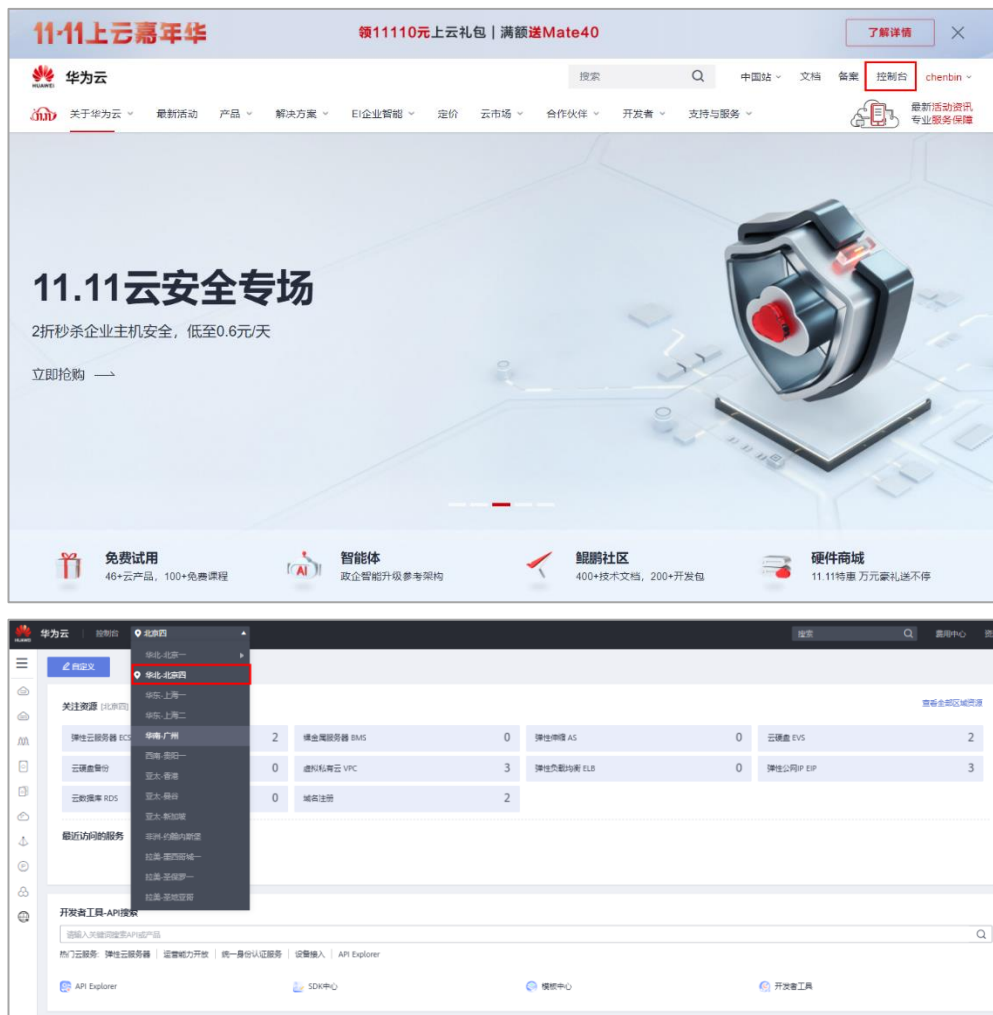
步骤 1 打开浏览器，输入华为云网站地址 <https://www.huaweicloud.com/>，在上方菜单栏中点击“登录”。




步骤 2 在登录界面输入账户名、密码，点击“登录”。

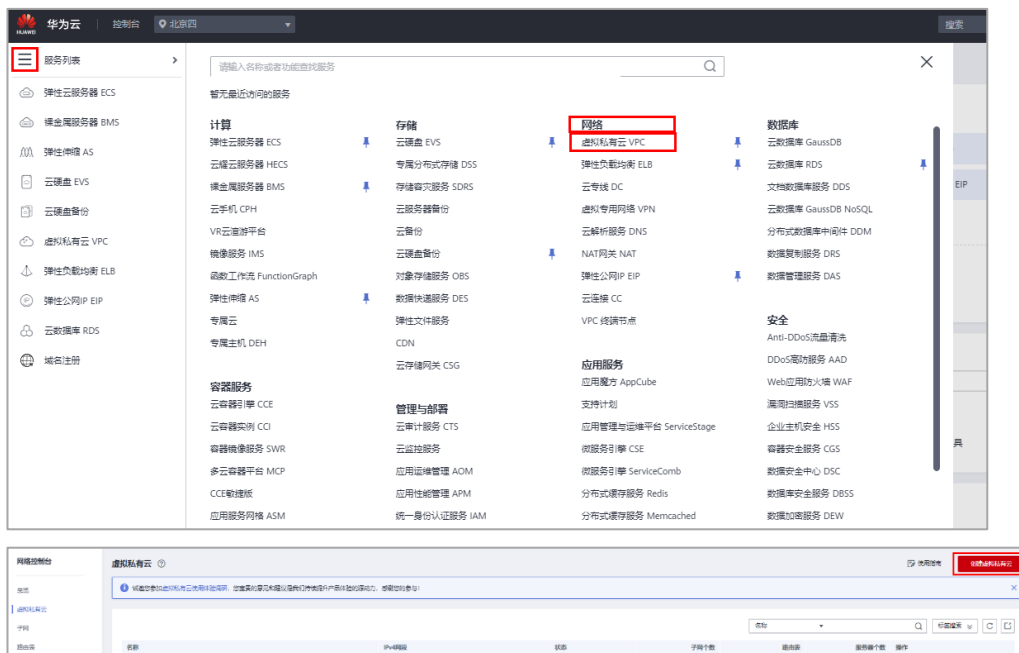


步骤 3 进入华为云控制台，更改当前区域为“华北-北京四”。以保证后续实验都在该区域下进行。



2.3 创建虚拟私有云 VPC

步骤 1 登陆华为云控制台，点击左侧图标  找到“网络>虚拟私有云 VPC”，进入网络控制台，点击“创建虚拟私有云”进入创建界面。



基本信息：

- 当前区域：华北-北京四
- 名称：vpc-shiyan
- 网段：192.168.0.0/16

基本信息

区域

华北-北京四

名称

vpc-shiyan

IPv4网段

192 · 168 · 0 · 0 / 16

高级配置

标签

不同区域的资源之间网内互通。请选择靠近您客户的区域，可以降低网络时延、提高访问速度。

建议使用网段 10.0.0.0/8-24 (选择) 172.16.0.0/12-24 (选择) 192.168.0.0/16-24 (选择)

该VPC网段 (192.168.0.0/16) 与当前区域下其他VPC网段重叠，如需使用VPC互通服务，建议您修改VPC网段。查看区域下已有vpc网段

默认子网：

- 可用区：可用区 1
- 名称：subnet -shiyan
- 子网 IPv4 网段：192.168.0.0/24
- 高级配置：默认配置（其余选项均保持默认即可）

默认子网

可用区: 可用区1

名称: subnet-shiyan

子网IPv4网段: 192.168.0.0 / 24 可用IP数: 251
子网创建完成后, 子网网段无法修改

子网IPv6网段: ☐ 开启IPv6

关联路由表: 默认

高级配置 | 网关 | DNS服务器地址 | DHCP租约时间 | 标签

步骤 2 点击“立即创建”，即可看到如下界面。

网络控制台

虚拟私有云

创建

子网

名称	IPv4网段	状态	子网个数	路由表	关联子网个数	操作
vpc-shiyan	192.168.0.0/16	可用	1	1	0	编辑 删除
subnet-shiyan	192.168.0.0/24	可用	1	1	0	编辑 删除

步骤 3 在网络控制台左侧菜单栏中点击“访问控制>安全组”进入安全组页面，点击需要修改的安全组“Sys-default”（当然也可以新建一个安全组，添加以下对应规则，并在创建云服务器时使用该安全组）。在“入方向规则”中点击“添加规则”。

< Sys-default

基本信息 | 入方向规则 | 出方向规则 | 关联实例

添加规则 | 快速添加规则 | 删除 | 一键放通 | 入方向规则: 15 教我设置

第一条规则如下：

- 协议：TCP
- 端口：22
- 其余保持默认

添加入方向规则 教我设置

安全组入方向规则为白名单（允许），放通入方向网络流量。

安全组 Sys-default

如您要添加多条规则，建议单击导入规则以进行批量导入。

协议端口	类型	源地址	描述	操作
TCP	IPv4	IP地址 0.0.0.0/0		复制 删除
22				

增加1条规则

确定 取消

第二条规则如下：

- 协议/应用：TCP
- 端口：80
- 其余保持默认



添加方向规则 教我设置

安全组 Sys-default

安全组入方向规则为白名单（允许），放通入方向网络流量。

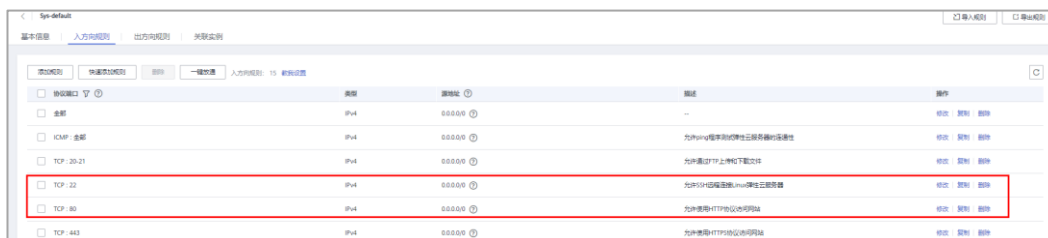
如需要添加多条规则，建议单击导入规则以进行批量导入。

协议端口	类型	源地址	描述	操作
TCP	IPv4	IP地址		复制 删除
80		0.0.0.0/0		

+ 增加1条规则


确定 取消

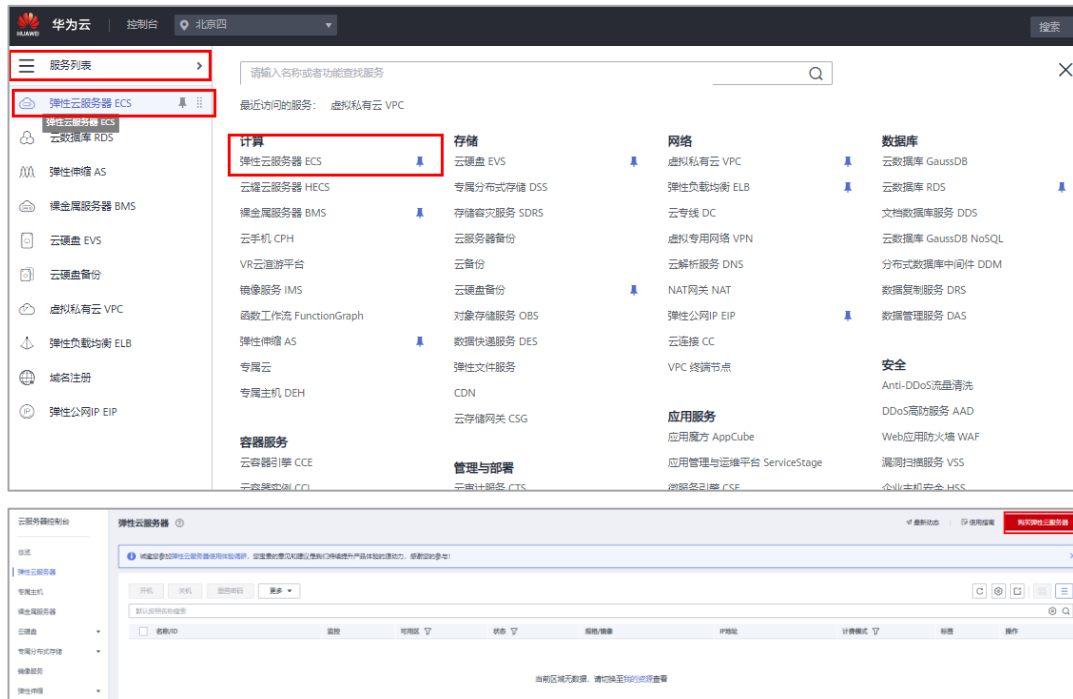
步骤 4 此时可以在该界面中查看到已添加成功的两条安全组规则。



协议端口	类型	源地址	描述	操作
全部	IPv4	0.0.0.0/0	--	修改 删除
ICMP - 全部	IPv4	0.0.0.0/0	为华为云服务的弹性云服务器提供连通性	修改 删除
TCP - 20-21	IPv4	0.0.0.0/0	为通过20-21上传和下载文件	修改 删除
TCP - 22	IPv4	0.0.0.0/0	为通过SSH登录云服务器	修改 删除
TCP - 80	IPv4	0.0.0.0/0	为通过HTTP协议访问网站	修改 删除
TCP - 443	IPv4	0.0.0.0/0	为通过HTTPS协议访问网站	修改 删除

2.4 创建弹性云服务器 ECS

步骤 1 登陆华为云控制台，点击 下的“计算>弹性云服务器 ECS”，进入云服务器控制台，点击购买弹性云服务器。



步骤 2 设置云服务器参数

填写待创建弹性云服务器的“基本信息”。

- 计费方式：按需付费
- 区域：华北-北京四
- 可用区：可用区 1
- 规格：通用计算型，2 个 vCPU，4GB 内存
- 镜像：公共镜像，CentOS7.4
- 磁盘：系统盘，高 IO，40GB

基础配置

网络配置

高级配置

确认配置

计费模式

包年/包月

按量计费

竞价计费

区域

华北-北京四

华东-上海一

华南-广州一

西南-贵阳一

华北-北京四

华东-上海一

华南-广州一

西南-贵阳一

可用区

可用区1

可用区2

可用区3

可用区7

CPU架构

x86计算

ARM计算

规格

最新系列

vCPU

全部

内存

全部

规格名称

通用计算型

通用计算型

内存优化型

磁盘优化型

超算VIO型

GPU加速型

AI加速型

通用入门型

规格名称	vCPU	内存	CPU	基频 / 最大睿频	内网收发包	规格参考价
x86.small.1	1vCPU	1GB	Intel Cascade Lake 2.6GHz	0.1/0.8 GHz/s	100,000	¥0.07/小时
x86.medium.2	1vCPU	2GB	Intel Cascade Lake 2.6GHz	0.1/0.8 GHz/s	100,000	¥0.17/小时
x86.medium.4	1vCPU	4GB	Intel Cascade Lake 2.6GHz	0.1/0.8 GHz/s	100,000	¥0.34/小时
x86.large.2	2vCPU	4GB	Intel Cascade Lake 2.6GHz	0.2/1.5 GHz/s	150,000	¥0.36/小时
x86.large.4	2vCPU	8GB	Intel Cascade Lake 2.6GHz	0.2/1.5 GHz/s	150,000	¥0.68/小时
x86.xlarge.2	4vCPU	8GB	Intel Cascade Lake 2.6GHz	0.35/2 GHz/s	250,000	¥0.73/小时
x86.xlarge.4	4vCPU	16GB	Intel Cascade Lake 2.6GHz	0.35/2 GHz/s	250,000	¥1.37/小时
x86.2xlarge.2	8vCPU	16GB	Intel Cascade Lake 2.6GHz	0.75/3 GHz/s	500,000	¥1.46/小时

当前规格 通用计算型 | x86.large.2 | 2vCPU | 4GB

镜像

公共镜像

私有镜像

共享镜像

市场镜像

CentOS

CentOS 7.4 64bit(AVG)

主机安全

开通主机安全

基础版(免费增值)

新磁盘

高IO

40

GB IOPS上限1,440, IOPS峰值上限5,000

购买量

1

台

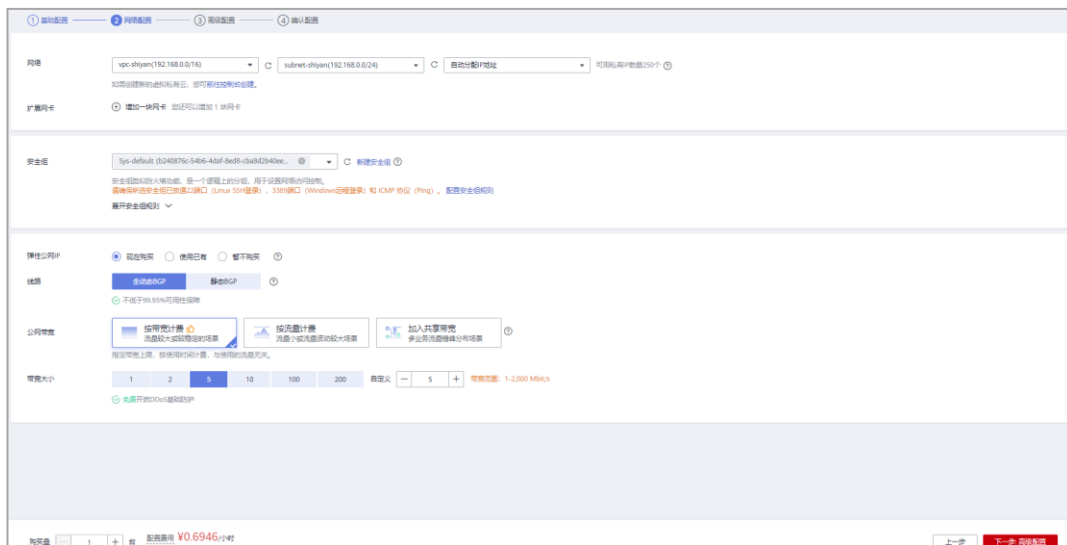
配置费用 ¥0.3796/小时

参考价格, 具体价格请以账单为准, 了解计费详情

下一步: 网络配置

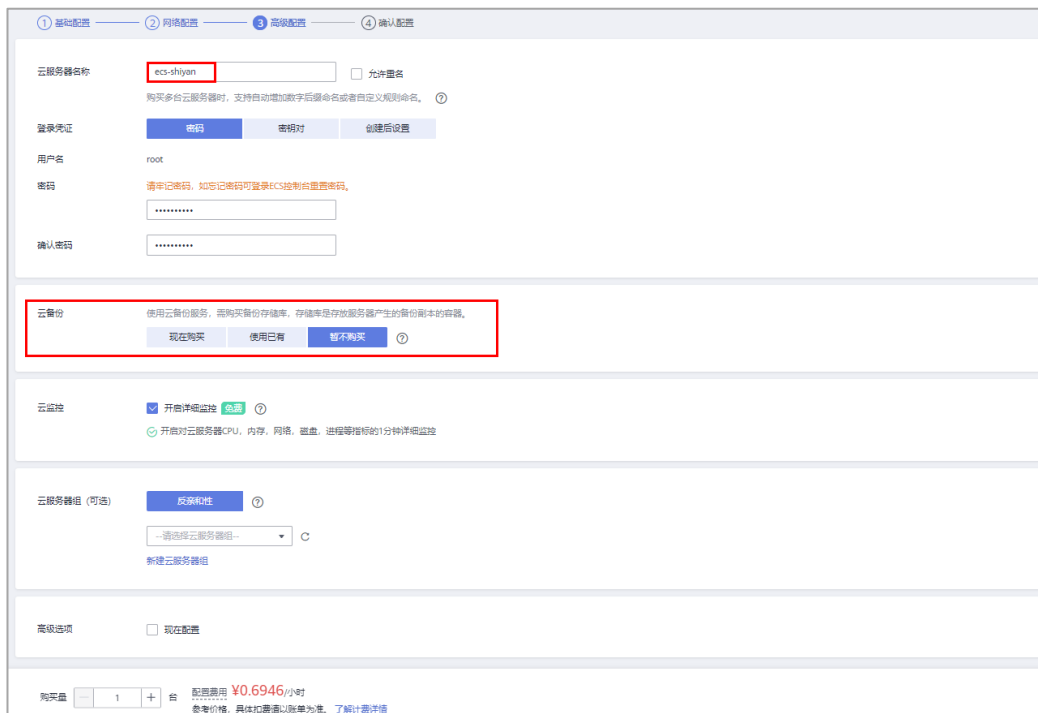
填写“网络”相关信息。

- 虚拟私有云：选择已创建好的 VPC，如 vpc-shiyan
- 安全组：选择已添加过安全组规则的安全组，如 Sys-default
- 弹性 IP：现在购买
- 规格：全动态 BGP
- 计费模式：按带宽计费
- 宽带：5M

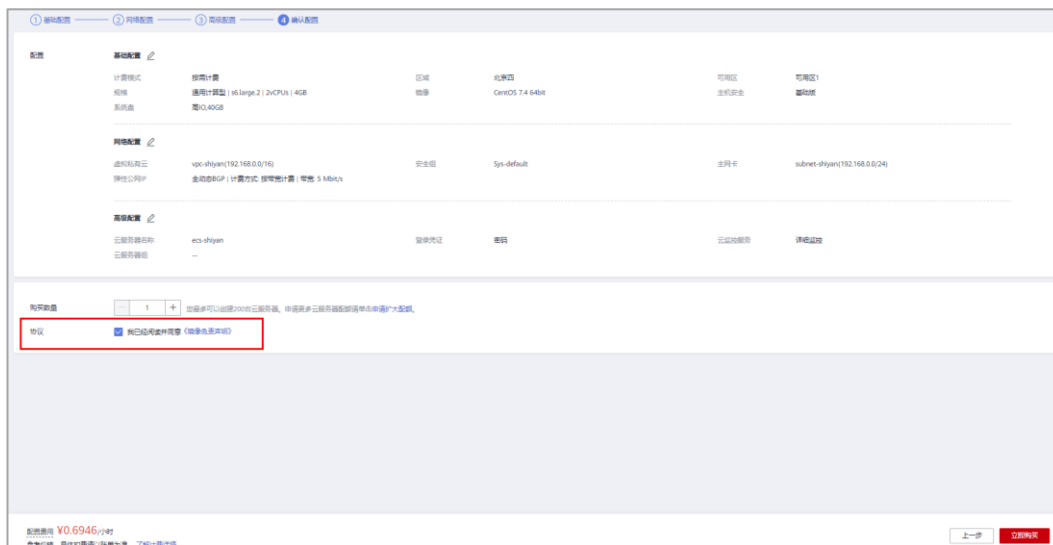
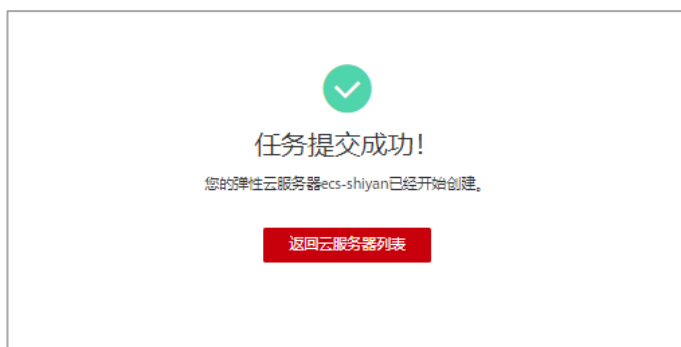


填写“高级配置”相关信息。

- 云服务器名称：如 ecs-shiyant
- 登录凭证：公有云提供两种弹性云服务器鉴权方式。此处我们选择密码登录方式。密码：自定，如 Huawei@123
- 云备份：暂不购买
- 云监控：可选，默认即可
- 云服务器组：可选，默认即可
- 高级配置：不勾选



步骤 3 点击“下一步 确认配置”，确认无误后，勾选“我已经阅读并同意《镜像免责声明》”，点击“立即购买”

步骤 4 返回云服务器控制台，等待片刻后，可看到已创建好的弹性云服务器。

弹性云服务器

诚邀您参加弹性云服务器使用体验调研，您宝贵的意见和建议是我们持续提升产品体验的源动力，感谢您的参与！

开机 关机 重置密码 更多

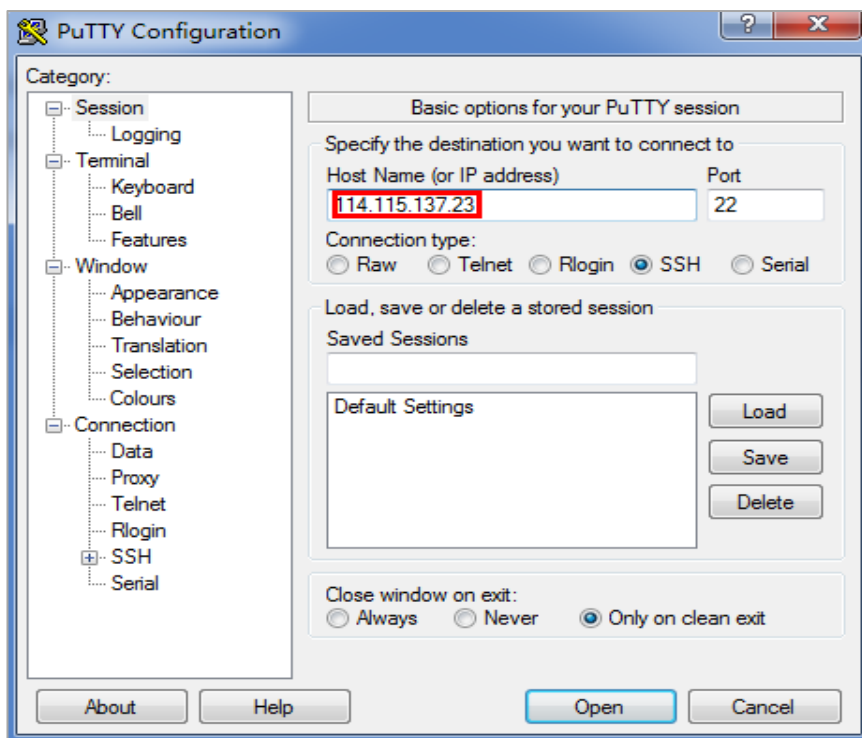
默认按照名称搜索

名称/ID	监控	可用区	状态	规格/镜像
ecs-shiyan 2ad05f24-70b7-4fcf-9923-b4e6994723f3		可用区1	运行中	2vCPUs 4GB s6.large.2 CentOS 7.4 64bit

3 安装基础软件

3.1 登陆弹性云服务器

步骤 1 使用 Putty 软件（若无 Putty 软件，可以参考去 <https://www.chiark.greenend.org.uk/~sgtatham/putty/> 进行 Putty 软件的下载安装）登录弹性云服务器（在“Host Name”中输入弹性云服务器的弹性 IP，点击“Open”即可）



步骤 2 输入在创建时就已设置好的弹性云服务器的用户名和密码，就能够成功登录到弹性云服务器。


```
[root@ecs-shiyan ~]# python --version
Python 2.7.5
[root@ecs-shiyan ~]# wget https://www.python.org/ftp/python/3.6.0/Python-3.6.0a1.tar.xz
--2019-03-04 15:14:26-- https://www.python.org/ftp/python/3.6.0/Python-3.6.0a1.tar.xz
Resolving www.python.org (www.python.org)... 151.101.228.223, 2a04:4e42:36::223
Connecting to www.python.org (www.python.org)|151.101.228.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15328032 (15M) [application/octet-stream]
Saving to: 欵窠ython-3.6.0a1.tar.xz欵?

100%[=====>] 15,328,032  1.77MB/s   in 96s

2019-03-04 15:16:02 (155 KB/s) - 欵窠ython-3.6.0a1.tar.xz欵? saved [15328032/15328032]
```

步骤 3 解压安装包，建议使用如下命令：tar xvf Python-3.6.0a1.tar.xz

```
Python-3.6.0a1/Tools/unicode/gencodec.py
Python-3.6.0a1/Tools/unicode/genwincodec.py
Python-3.6.0a1/Tools/unicode/genwincodecs.bat
Python-3.6.0a1/Tools/unicode/listcodecs.py
Python-3.6.0a1/Tools/unicode/makeunicodedata.py
Python-3.6.0a1/Tools/unicode/mkstringprep.py
Python-3.6.0a1/Tools/unicode/python-mappings/
Python-3.6.0a1/Tools/unicode/python-mappings/CP1140.TXT
Python-3.6.0a1/Tools/unicode/python-mappings/CP273.TXT
Python-3.6.0a1/Tools/unicode/python-mappings/KOI8-U.TXT
Python-3.6.0a1/Tools/unicode/python-mappings/TIS-620.TXT
Python-3.6.0a1/Tools/unittestgui/
Python-3.6.0a1/Tools/unittestgui/README.txt
Python-3.6.0a1/Tools/unittestgui/unittestgui.py
Python-3.6.0a1/aclocal.m4
Python-3.6.0a1/config.guess
Python-3.6.0a1/config.sub
Python-3.6.0a1/configure
Python-3.6.0a1/configure.ac
Python-3.6.0a1/install-sh
Python-3.6.0a1/pyconfig.h.in
Python-3.6.0a1/setup.py
[root@ecs-shiyan ~]#
```

步骤 4 执行命令：cd Python-3.6.0a1

```
Python-3.6.0a1/config.guess
Python-3.6.0a1/config.sub
Python-3.6.0a1/configure
Python-3.6.0a1/configure.ac
Python-3.6.0a1/install-sh
Python-3.6.0a1/pyconfig.h.in
Python-3.6.0a1/setup.py
[root@ecs-shiyan ~]# cd Python-3.6.0a1
[root@ecs-shiyan Python-3.6.0a1]#
```

步骤 5 执行命令：./configure

```
[root@ecs-shiyan ~]# cd Python-3.6.0a1
[root@ecs-shiyan Python-3.6.0a1]# ./configure
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for --enable-universalsdk... no
checking for --with-universal-archs... no
checking MACHDEP... linux
checking for --without-gcc... no
checking for --with-icc... no
checking for gcc... no
checking for cc... no
checking for cl.exe... no
configure: error: in `/root/Python-3.6.0a1':
configure: error: no acceptable C compiler found in $PATH
See 'config.log' for more details
[root@ecs-shiyan Python-3.6.0a1]#
```

注：如果出现“configure: error: no acceptable C compiler found in \$PATH”异常提示，是因为未安装合适的编译器。解决方法：安装/升级 gcc 及其他依赖的包，建议

使用命令: `sudo yum install gcc-c++` (随后执行步骤 6 和步骤 7, 若步骤 5 无异常提示, 则忽略步骤 6 和步骤 7 即可)

步骤 6 安装/升级 gcc 及其他依赖的包, 建议使用命令: `sudo yum install gcc-c++`。并在随后提示安装包是否 OK 时, 输入 y 并回车。出现如下提示说明依赖的包安装成功。

```
Verifying : glibc-2.17-196.el7.x86_64 17/19
Verifying : libstdc++-4.8.5-16.el7.x86_64 18/19
Verifying : glibc-common-2.17-196.el7.x86_64 19/19

Installed:
gcc-c++.x86_64 0:4.8.5-36.el7

Dependency Installed:
cpp.x86_64 0:4.8.5-36.el7
gcc.x86_64 0:4.8.5-36.el7
glibc-devel.x86_64 0:2.17-260.el7_6.3
glibc-headers.x86_64 0:2.17-260.el7_6.3
kernel-headers.x86_64 0:3.10.0-957.5.1.el7
libmpc.x86_64 0:1.0.1-3.el7
libstdc++-devel.x86_64 0:4.8.5-36.el7
mpfr.x86_64 0:3.1.1-4.el7

Dependency Updated:
glibc.x86_64 0:2.17-260.el7_6.3 glibc-common.x86_64 0:2.17-260.el7_6.3
libgcc.x86_64 0:4.8.5-36.el7 libgomp.x86_64 0:4.8.5-36.el7
libstdc++.x86_64 0:4.8.5-36.el7

Complete!
[root@ecs-shiyan Python-3.6.0a1]#
```

步骤 7 再重新执行 `./configure` 命令。

```
checking for broken mbstowcs... no
checking for --with-computed-gotos... no value specified
checking whether gcc -pthread supports computed gotos... yes
checking for build directories... done
checking for -O2... yes
checking for glibc _FORTIFY_SOURCE/memmove bug... no
checking for gcc ipa-pure-const bug... no
checking for stdatomic.h... no
checking for GCC >= 4.7 __atomic builtins... yes
checking for ensurepip... upgrade
checking if the dirent structure of a d_type field... yes
checking for the Linux getrandom() syscall... yes
checking for the getrandom() function... no
configure: creating ./config.status
config.status: creating Makefile.pre
config.status: creating Modules/Setup.config
config.status: creating Misc/python.pc
config.status: creating Misc/python-config.sh
config.status: creating Modules/ld_so_aix
config.status: creating pyconfig.h
creating Modules/Setup
creating Modules/Setup.local
creating Makefile
[root@ecs-shiyan Python-3.6.0a1]#
```

步骤 8 执行命令: `make && make install`

```
rm -f /usr/local/bin/idle3
(cd /usr/local/bin; ln -s idle3.6 idle3)
rm -f /usr/local/bin/pydoc3
(cd /usr/local/bin; ln -s pydoc3.6 pydoc3)
rm -f /usr/local/bin/2to3
(cd /usr/local/bin; ln -s 2to3-3.6 2to3)
rm -f /usr/local/bin/pyvenv
(cd /usr/local/bin; ln -s pyvenv-3.6 pyvenv)
if test "x" != "x" ; then \
    rm -f /usr/local/bin/python3-32; \
    (cd /usr/local/bin; ln -s python3.6-32 python3-32) \
fi
rm -f /usr/local/share/man/man1/python3.1
(cd /usr/local/share/man/man1; ln -s python3.6.1 python3.1)
if test "xupgrade" != "xno" ; then \
    case upgrade in \
        upgrade) ensurepip="--upgrade" ;; \
        install|*) ensurepip="" ;; \
    esac; \
    ./python -E -m ensurepip \
        $ensurepip --root=/ ; \
fi
Ignoring ensurepip failure: pip 8.1.1 requires SSL/TLS
[root@ecs-shiyan Python-3.6.0a1]#
```

注：以上界面为执行成功的界面。但提示 pip 错误，原因是我的系统中少了 openssl-devel 包，可以先忽略。

步骤 9 查看 Python3 版本，使用命令：python3 --version，可以查看到此时的版本为 Python 3.6.0a1

```
[root@ecs-shiyan Python-3.6.0a1]# python3 --version
Python 3.6.0a1
```

3.3 升级默认 pip 版本

pip 是通用的 Python 包管理工具。提供了对 Python 包的查找、下载、安装、卸载功能。Python3 安装成功后自带 pip3，但版本比较老，建议升级到 pip 最新版本。同时前面安装 python3 提示 “Ignoring ensurepip failure: pip 8.1.1 requires SSL/TLS” 错误，导致 pip 未成功安装，所以需要重新安装 pip。

步骤 1 安装 openssl-devel 包，建议使用命令：yum install openssl-devel -y


```

Dependency Installed:
keyutils-libs-devel.x86_64 0:1.5.8-3.el7      krb5-devel.x86_64 0:1.15.1-37.el7_6
libcom_err-devel.x86_64 0:1.42.9-13.el7      libkadm5.x86_64 0:1.15.1-37.el7_6
libselinux-devel.x86_64 0:2.5-14.1.el7        libsepol-devel.x86_64 0:2.5-10.el7
libverto-devel.x86_64 0:0.2.5-4.el7          pcre-devel.x86_64 0:8.32-17.el7
zlib-devel.x86_64 0:1.2.7-18.el7

Dependency Updated:
e2fsprogs.x86_64 0:1.42.9-13.el7
e2fsprogs-libs.x86_64 0:1.42.9-13.el7
krb5-libs.x86_64 0:1.15.1-37.el7_6
libcom_err.x86_64 0:1.42.9-13.el7
libselinux.x86_64 0:2.5-14.1.el7
libselinux-python.x86_64 0:2.5-14.1.el7
libselinux-utils.x86_64 0:2.5-14.1.el7
libsepol.x86_64 0:2.5-10.el7
libss.x86_64 0:1.42.9-13.el7
openssl.x86_64 1:1.0.2k-16.el7
openssl-libs.x86_64 1:1.0.2k-16.el7
zlib.x86_64 0:1.2.7-18.el7

Complete!
[root@ecs-shiyang Python-3.6.0a1]#

```

步骤 2 执行命令：make && make install。出现如下提示说明安装 pip 安装成功。

```

Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-8.1.1 setuptools-20.10.1
[root@ecs-shiyang Python-3.6.0a1]#

```

步骤 3 升级 pip3，使用命令：pip3 install --upgrade pip。出现如下提示说明升级 pip 到最新版本了。

```

Successfully uninstalled pip-8.1.1
Successfully installed pip-19.0.3
[root@ecs-shiyang Python-3.6.0a1]#

```

3.4 安装 Web.py 框架

Web.py 官方教程地址：<http://webpy.org/>，建议使用如下命令安装 web.py：pip3 install web.py==0.40.dev0。出现如下提示说明安装成功。

```

Installing collected packages: web.py
Running setup.py install for web.py ... done
Successfully installed web.py-0.40.dev0
[root@ecs-shiyang Python-3.6.0a1]#

```

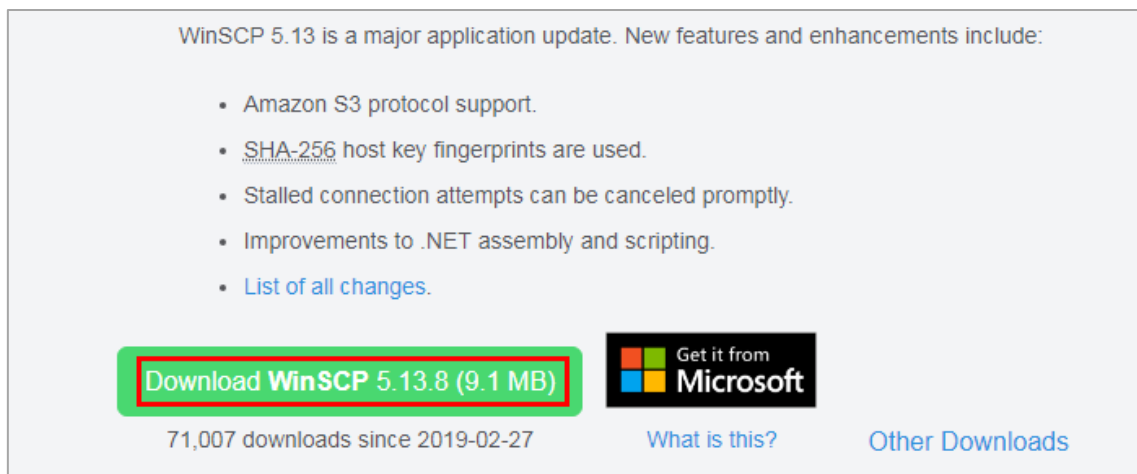
3.5 安装 WinSCP

通常情况下，我们在本地 Windows 操作系统上编辑代码，完成后再上传至 ECS 上（CentOS Linux 系统）。WinSCP 是一个 Windows 环境下使用的 SSH 的开源图形化 SFTP 客户端，同时支持 SCP 协议。它的主要功能是在本地与远程计算机间安全地复制文件，并且可以直接编辑文件。

步骤 1 打开浏览器，输入 WinSCP 安装链接：<https://winscp.net/eng/docs/lang:chs>，点击“下载”。



步骤 2 点击“WinSCP 下载界面（英文）”，进行对应的下载，下载好后，自行执行安装即可。



4 上传代码&服务启动

4.1 本地编写代码文件

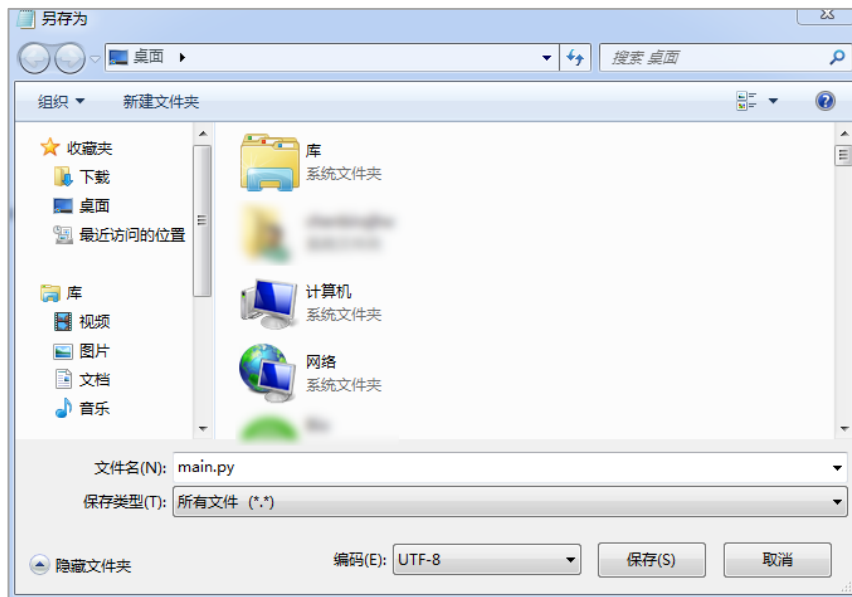
4.1.1 新建 main.py 文件

步骤 1 在本地新建一个记事本文件，输入代码如下：

```
# -*- coding: utf-8 -*-  
# filename: main.py  
import web  
from handle import Handle  
  
urls = (  
    '/wx', 'Handle',  
)  
  
if __name__ == '__main__':  
    app = web.application(urls, globals())  
    app.run()
```

步骤 2 将该记事本文件另存为，格式如下：

- 文件名：main.py
- 保存类型：所有文件
- 编码：UTF-8



4.1.2 新建 handle.py 文件

步骤 1 在本地新建一个记事本文件，输入代码如下：

```
# -*- coding: utf-8 -*-
```

```
# filename: handle.py
```

```
import hashlib
```

```
import web
```

```
import receive
```

```
import time
```

```
import os
```

```
class Handle(object):
```

```
    def __init__(self):
```

```
        self.app_root = os.path.dirname(__file__)
```

```
        self.templates_root = os.path.join(self.app_root, 'templates')
```

```
        self.render = web.template.render(self.templates_root)
```

```
    def GET(self):
```

```
try:
    data = web.input()
    if len(data) == 0:
        return "hello, this is handle view"
    signature = data.signature
    timestamp = data.timestamp
    nonce = data.nonce
    echostr = data.echostr
    token = "weixin"

    list = [token, timestamp, nonce]
    list.sort()
    s = list[0] + list[1] + list[2]
    hashcode = hashlib.sha1(s.encode('utf-8')).hexdigest()
    print( "handle/GET func: hashcode, signature: ", hashcode, signature)
    if hashcode == signature:
        return echostr
    else:
        return echostr
except (Exception) as Argument:
    return Argument
```

```
def POST(self):
```

```
    try:
        webData = web.data()
        print("Handle Post webdata is:\n", webData)
        #打印消息体日志
        recMsg = receive.parse_xml(webData)

        if isinstance(recMsg, receive.Msg) and recMsg.MsgType == 'text':
```

```

toUser = recMsg.FromUserName
fromUser = recMsg.ToUserName
content = "欢迎关注! " + str(recMsg.Content)
print('Reply message info:\n')
print('toUser =', toUser)
print('fromUser = ', fromUser)
print('content = ', content)
return self.render.reply_text(toUser, fromUser, int(time.time()),
content)

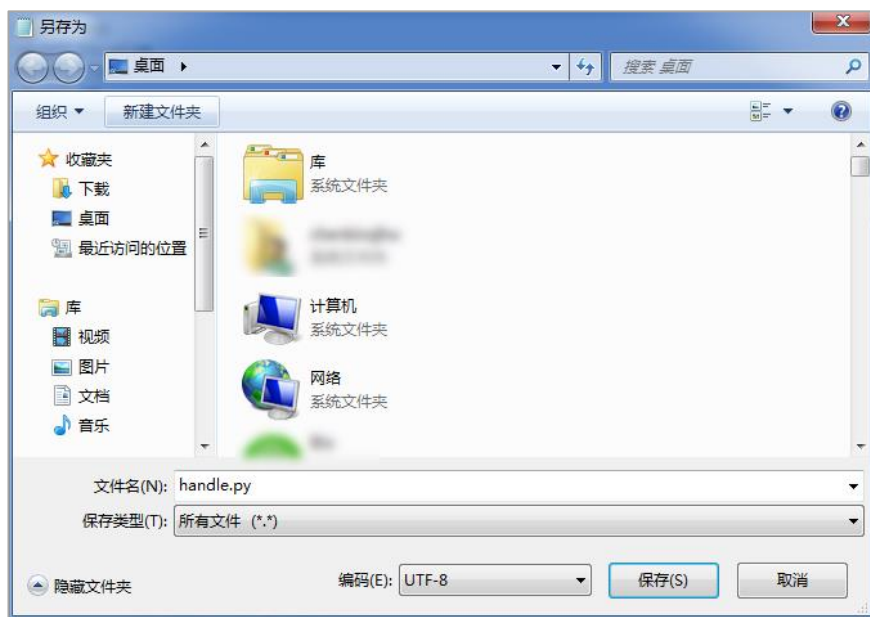
else:
    print("不支持的消息类型: ",recMsg.MsgType)
    return "success"

except (Exception) as Argument:
    return Argument

```

步骤 2 将该记事本文件另存为，格式如下：

- 文件名：handle.py
- 保存类型：所有文件
- 编码：UTF-8



4.1.3 新建 receive.py 文件

步骤 1 在本地新建一个记事本文件，输入代码如下：

```
# -*- coding: utf-8 -*-  
  
# filename: receive.py  
  
import xml.etree.ElementTree as ET  
  
def parse_xml(web_data):  
    if len(web_data) == 0:  
        return None  
  
    xmlData = ET.fromstring(web_data)  
    msg_type = xmlData.find('MsgType').text  
    if msg_type == 'text':  
        return TextMsg(xmlData)  
    elif msg_type == 'image':  
        return ImageMsg(xmlData)  
    elif msg_type == 'location':  
        return LocationMsg(xmlData)  
    elif msg_type == 'event':  
        return EventMsg(xmlData)  
  
class Event(object):  
    def __init__(self, xmlData):  
        self.ToUserName = xmlData.find('ToUserName').text  
        self.FromUserName = xmlData.find('FromUserName').text  
        self.CreateTime = xmlData.find('CreateTime').text  
        self.MsgType = xmlData.find('MsgType').text  
        self.Eventkey = xmlData.find('EventKey').text  
  
class Msg(object):
```

```
def __init__(self, xmlData):
    self.ToUserName = xmlData.find('ToUserName').text
    self.FromUserName = xmlData.find('FromUserName').text
    self.CreateTime = xmlData.find('CreateTime').text
    self.MsgType = xmlData.find('MsgType').text
    self.MsgId = xmlData.find('MsgId').text

class TextMsg(Msg):
    def __init__(self, xmlData):
        Msg.__init__(self, xmlData)
        self.Content = xmlData.find('Content').text

class ImageMsg(Msg):
    def __init__(self, xmlData):
        Msg.__init__(self, xmlData)
        self.PicUrl = xmlData.find('PicUrl').text
        self.MediaId = xmlData.find('MediaId').text

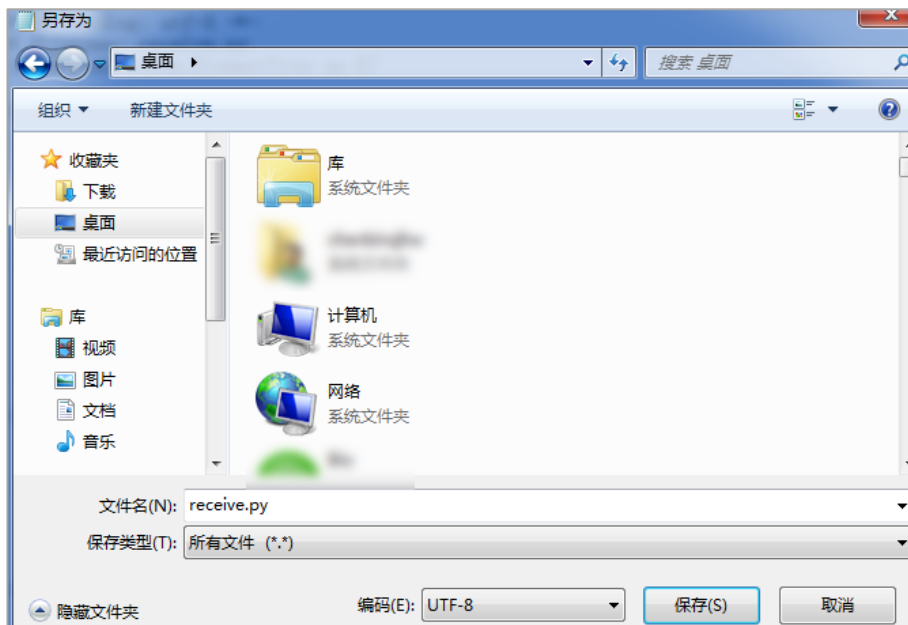
class LocationMsg(Msg):
    def __init__(self, xmlData):
        Msg.__init__(self, xmlData)
        self.Location_X = xmlData.find('Location_X').text
        self.Location_Y = xmlData.find('Location_Y').text

class EventMsg(Msg):
    def __init__(self, xmlData):
        Event.__init__(self, xmlData)
        self.Event = xmlData.find('Event').text
```

步骤 2 将该记事本文件另存为，格式如下：

- 文件名：receive.py

- 保存类型：所有文件
- 编码：UTF-8



4.1.4 新建 templates 文件夹

步骤 1 新建 templates 文件夹，在文件夹下新建记事本文件，输入代码如下：

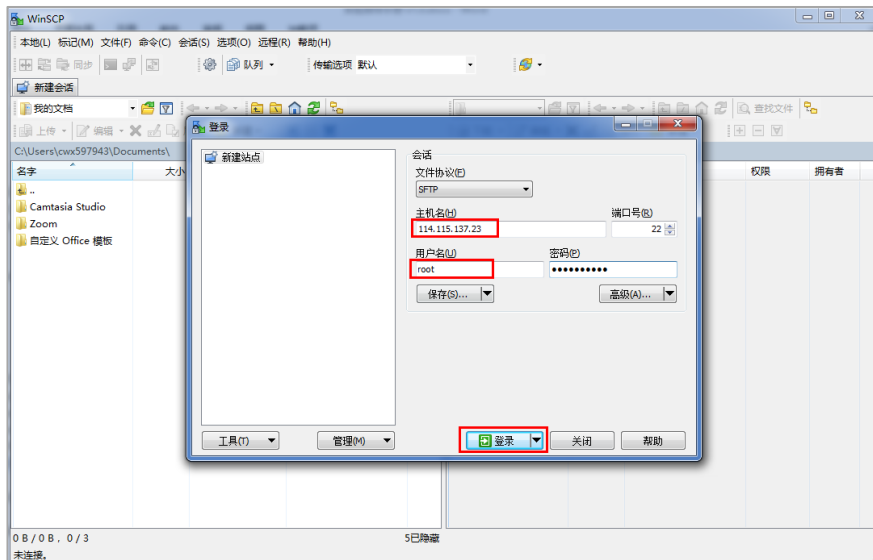
```
$def with (toUser,fromUser,createTime,content)
<xml>
<ToUserName><![CDATA[$toUser]]></ToUserName>
<FromUserName><![CDATA[$fromUser]]></FromUserName>
<CreateTime>$createTime</CreateTime>
<MsgType><![CDATA[text]]></MsgType>
<Content><![CDATA[$content]]></Content>
</xml>
```

步骤 2 将该记事本文件另存为，格式如下：

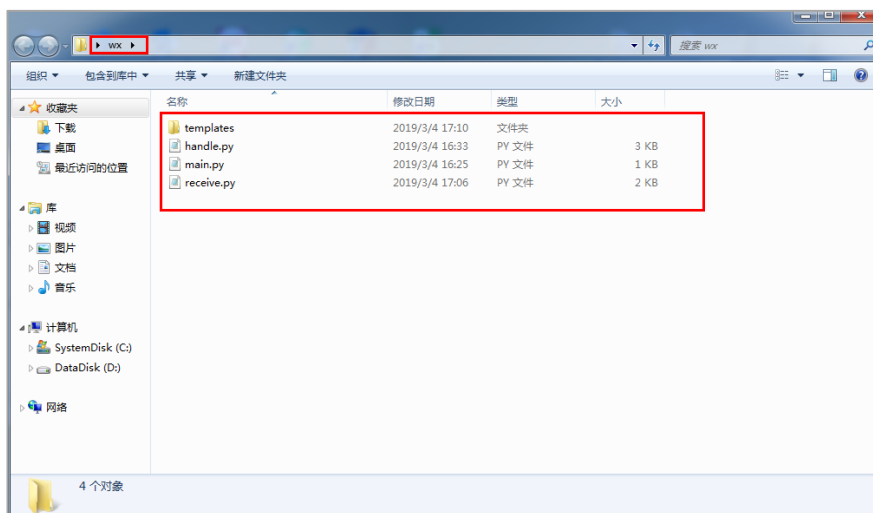
- 文件名：reply_text.xml
- 保存类型：所有文件
- 编码：UTF-8

4.2 WinSCP 上传代码文件

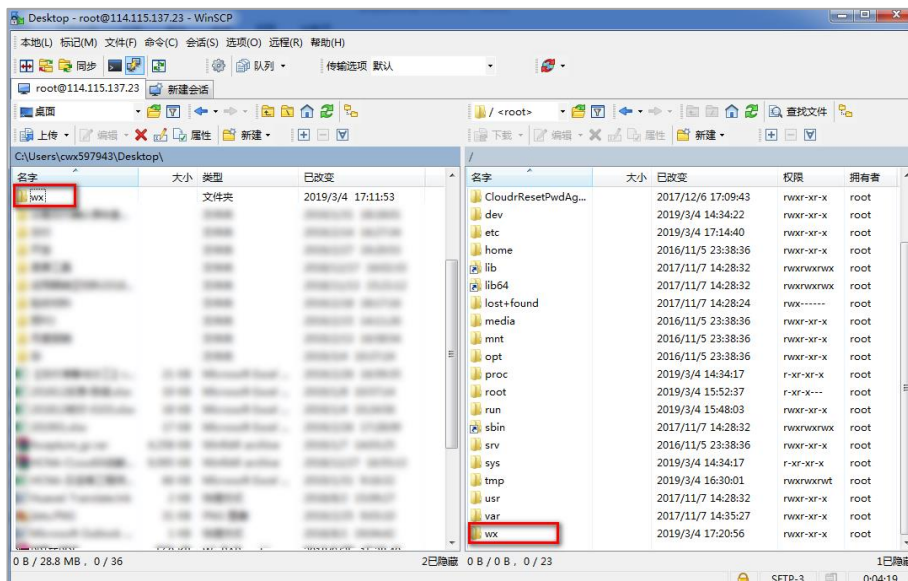
步骤 1 打开 WinSCP，在界面中输入弹性云主机的主机名、用户名以及密码，确认输入无误后，点击登录。



步骤 2 在本地创建一个名称为 wx 的文件夹，将上述代码文件拖入到该文件夹中。

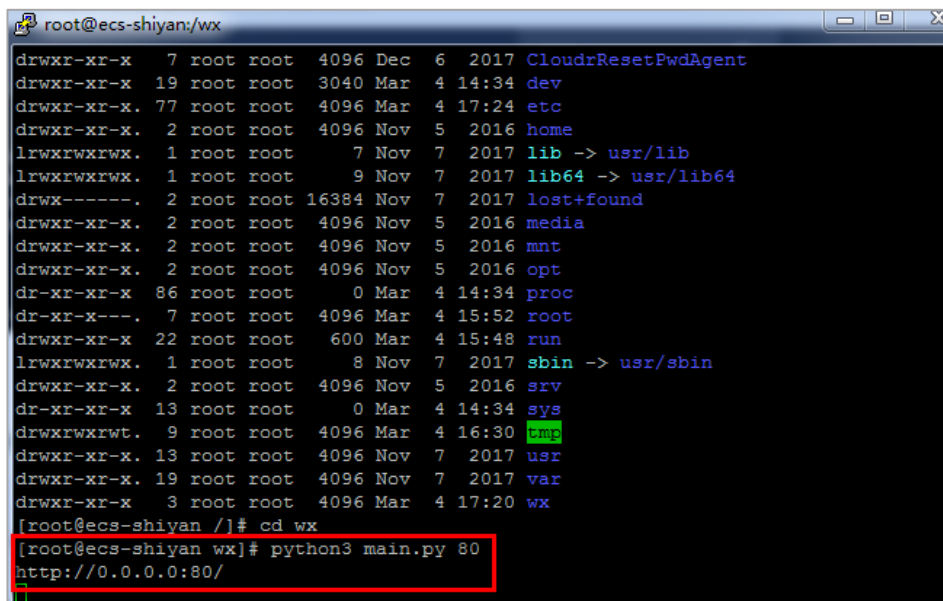


步骤 3 用 WinSCP 工具将 wx 文件夹上传到云服务器的根目录中。



4.3 启动对应服务

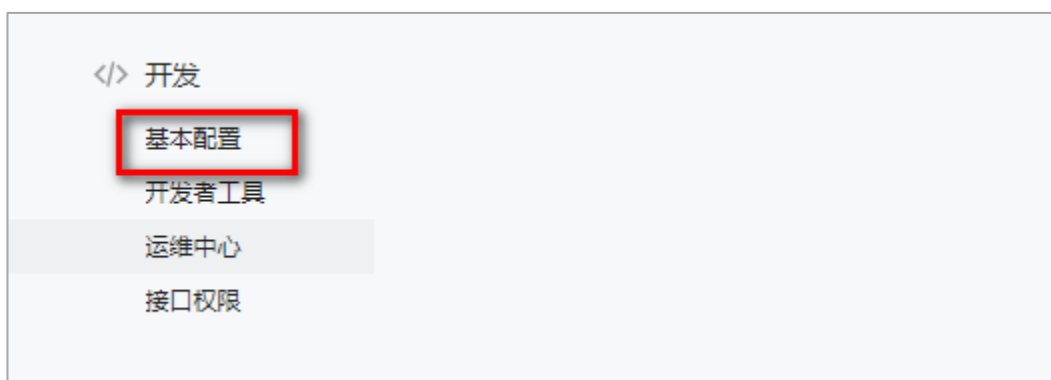
使用如下命令启动服务：python3 main.py 80。启动成功如图所示。



5 启用开发者模式&实验验证

5.1 启用开发者模式

步骤 1 登录微信公众平台，选择“开发 > 基本配置”，单击“修改配置”。



步骤 2 填写配置信息，单击“提交”。

URL：填写 ECS 上配置的弹性公网 IP，此处应该写：http://114.115.137.23/wx。

Token：需要与 handle.py 中对应 token 取值完全一致，此处为 weixin。

EncodingAESKey：随机生成。

消息加解密方式：此为示例，选择简单的“明文模式”。

基本配置

基本配置 / 填写服务器配置

请填写接口配置信息，此信息需要你拥有自己的服务器资源。
填写的URL需要正确响应微信发送的Token验证，请阅读接入指南。

URL
必须以http://或https://开头，分别支持80端口和443端口。

Token
必须为英文或数字，长度为3-32字符。
[什么是Token？](#)

EncodingAESKey
消息加密密钥由43位字符组成，可随机修改，字符范围为A-Z，a-z，0-9。
[什么是EncodingAESKey？](#)

消息加解密方式 请根据业务需要，选择消息加解密类型，启用后将立即生效

- ☒ 明文模式
明文模式下，不使用消息体加解密功能，安全系数较低
- ☐ 兼容模式
兼容模式下，明文、密文将共存，方便开发者调试和维护
- ☐ 安全模式（推荐）
安全模式下，消息包为纯密文，需要开发者加密和解密，安全系数高

步骤 3 验证 token 成功，单击“启用”。

注：如果 token 验证失败，请检查 Token 配置与 handle.py 中 GET 消息处理代码是否一致。

服务器配置(已启用)

服务器地址(URL)

令牌(Token)

消息加解密密钥①
(EncodingAESKey)

消息加解密方式 ☒ 明文模式

已绑定的微信开放平台帐号

未绑定帐号

5.2 微信公众号验证实验结果

使用微信关注公众号，任意发送一条文本消息，看是否能够收到回复。如能收到回复则表明系统处理正常。



6 资源释放

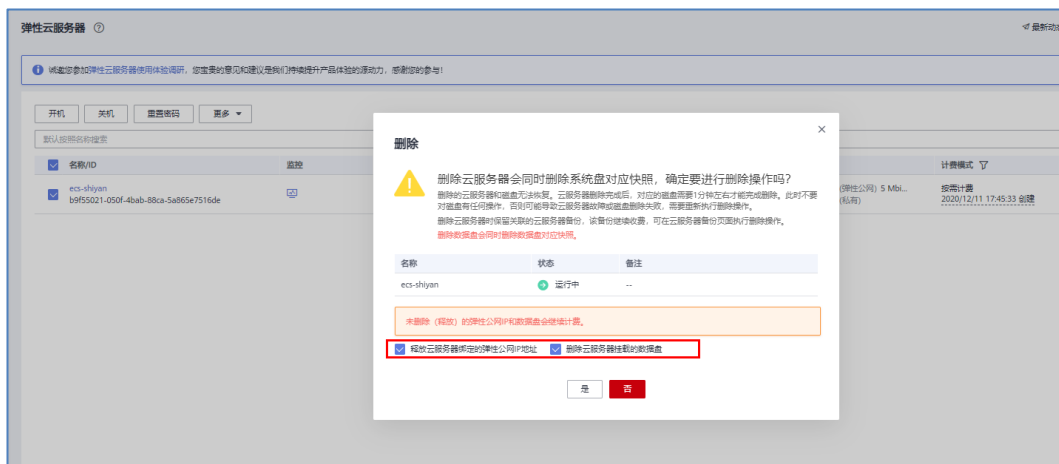
6.1 实验介绍

在完成所有实验之后，需手动释放服务所占用的资源，包括：

- 弹性云服务器 ECS 以及弹性公网 IP
- 虚拟私有云 VPC

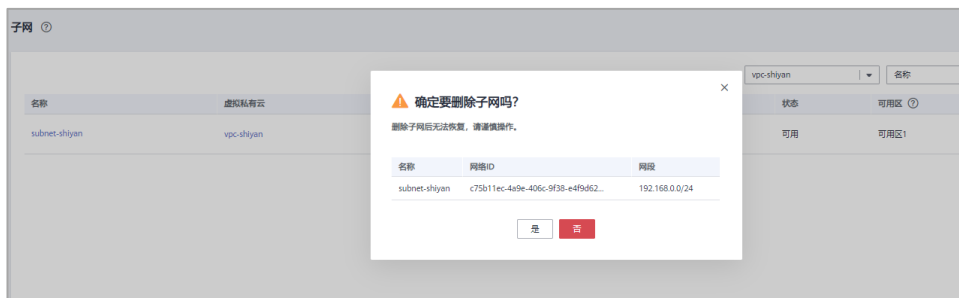
6.2 删除弹性云服务器 ECS

步骤 1 登陆华为云控制台，进入弹性云服务器界面，点击“更多”按钮，在展开的选项中点击“删除”（在删除时，需要勾选删除弹性公网 IP 和数据盘，以免删除不完整带来不必要的费用）。



6.3 删除虚拟私有云

步骤 1 登陆华为云控制台，找到自己创建的虚拟私有云 VPC，点击该 VPC，删除该 VPC 的子网。



步骤 2 删除该 VPC。



至此，资源已全部删除完毕。