

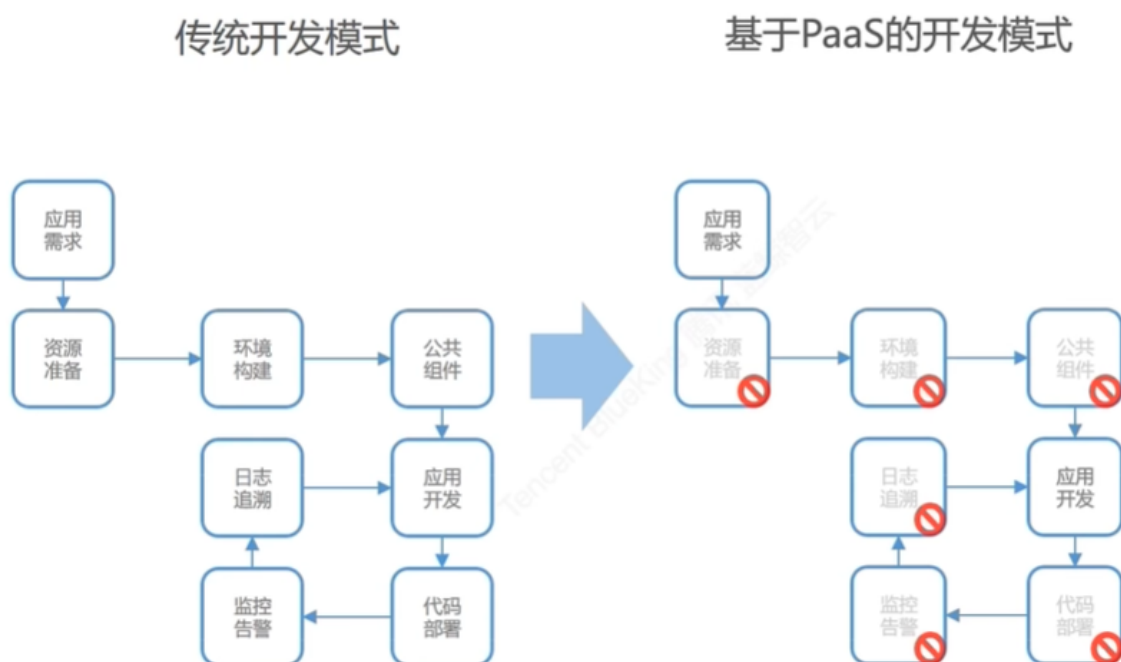
蓝鲸平台部署手册

蓝鲸智云是一套基于PaaS的技术解决方案，提供了完善的前后端开发框架、调度引擎、公共组件等模块，帮助业务的技术人员快速构建低成本、免运维的支撑工具和运营系统。

PaaS(Platform as a Service)

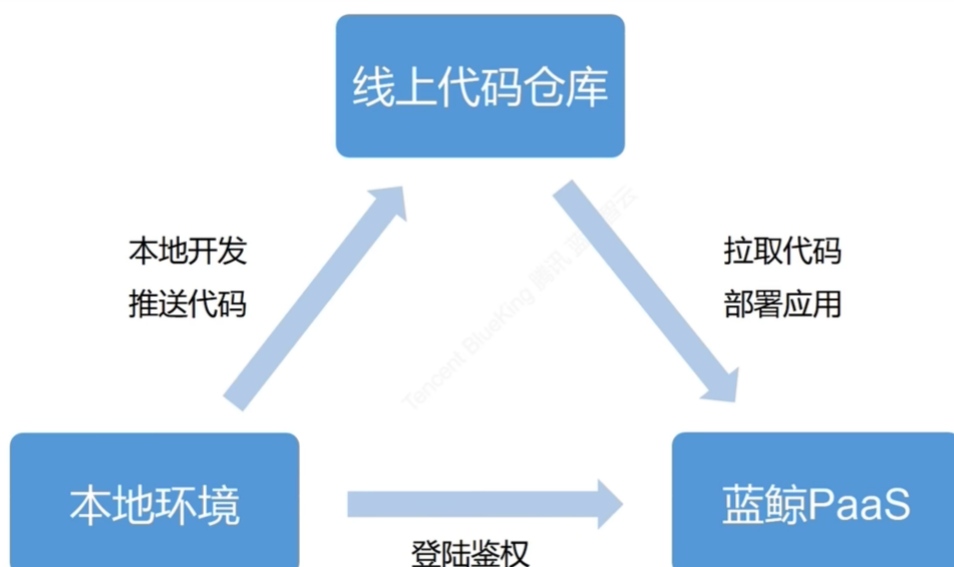
PaaS平台提供软件部署的功能，抽象掉硬件和操作系统的细节，开发者只需要关注自己的业务逻辑，无需关心底层的配置和实现。

开发模式对比



基于PaaS来开发SaaS应用

SaaS开发流程



作业目的

体验如何在蓝鲸云平台上部署一个应用。

作业流程

1. 安装本地开发环境
2. 代码仓库：申请账号并创建一个项目
3. 在蓝鲸平台上创建一个SaaS应用
4. 下载蓝鲸SaaS开发框架3.0，安装框架所需依赖包
5. 本地创建数据库，修改框架本地运行配置
6. 配置Host：127.0.0.1 appdev.paas-edu.bktencent.com
7. 本地运行框架并访问 <http://appdev.paas-edu.bktencent.com:8000>
8. 增加一个helloworld页面，要求可以通过/helloworld/访问到
9. 把代码推送到仓库，分别部署到测试环境和正式环境

1. 安装本地开发环境

开发环境

- Python 3.6.7
- Mysql 5.7
- Git
- Pycharm

Python 安装

下载地址：<https://www.python.org/downloads/release/python-367/>

可以参考网上的安装教程。

Mysql 安装

下载地址：<https://dev.mysql.com/downloads/mysql/5.7.html>

可以参考：<http://www.jb51.net/article/23876.htm>，教程可能有点老，根据自己的情况适当配置。

Git 安装

下载地址：<https://gitforwindows.org/>

Pycharm安装

下载地址：<https://www.jetbrains.com/pycharm/download/>

2. 配置代码仓库

以gitee为例：<https://gitee.com/>

2.1 创建gitee账号（自行创建）

2.2 创建项目仓库

点击在右上方加号里的新建仓库的选项：



参考以下模板创建仓库：

新建仓库

在其他网站已经有仓库了吗？[点击导入](#)

仓库名称 * ✓

bkdemo1

归属



路径 * ✓

/ bkdemo1

仓库地址：https://gitee.com/[username]/bkdemo1

仓库介绍

11/200

应用部署到蓝鲸平台示例

☐ 开源（所有人可见）?

☒ 私有（仅仓库成员可见）

☒ 初始化仓库（设置语言、.gitignore、开源许可证）

选择语言

Python

添加 .gitignore

Python

添加开源许可证

许可证向导

请选择开源许可证

☐ 设置模板（添加 Readme、Issue、Pull Request 模板文件）

☐ 选择分支模型（仓库创建后将根据所选模型创建分支）

创建

2.3 本地仓库与线上仓库同步

在本地打开Git后配置全局用户名和邮箱：

```
git config --global user.name "xxx"
git config --global user.email "xxx@xxx"
```

查看配置：

```
git config --list
```

如下图所示：

```
abc@DESKTOP-E444806 MINGW64 ~
$ git config --global user.name "I"

abc@DESKTOP-E444806 MINGW64 ~
$ git config --global user.email "I"

abc@DESKTOP-E444806 MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=D:/git/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
credential.https://gitee.com.provider=generic
user.name=I
user.email=I
```

在本地创建一个用于与线上仓库同步的目录并在Git上跳转到该目录下：

```
abc@DESKTOP-E444806 MINGW64 /d/ta_course/bkdemo1
$ |
```

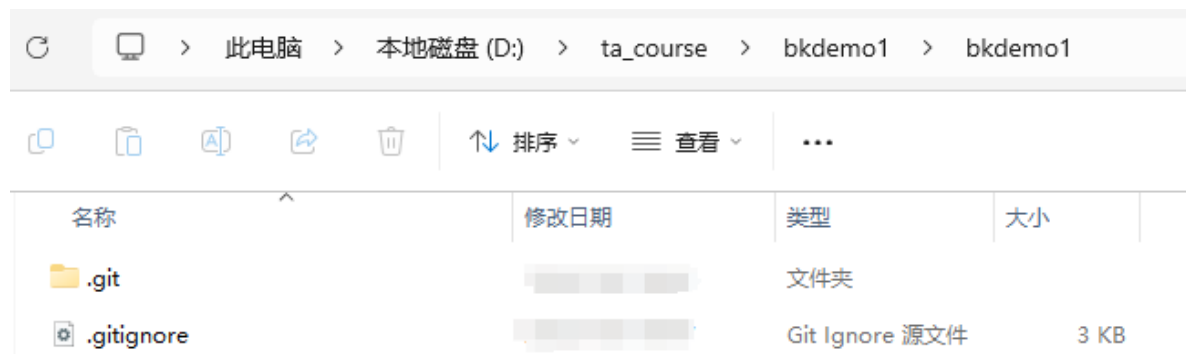
在gitee上复制线上仓库的地址：



使用Git来克隆线上仓库的文件到本地仓库中：

```
git clone xxx
```

成功后可以看到本地仓库已经克隆线上仓库的文件：



3. 在蓝鲸平台上创建SaaS应用

访问<https://ce.bktencent.com/>并登录:



点击开发者中心:



在开发中心页面上点击创建应用后的页面：

创建应用

普通应用

云原生应用

外链应用

① 平台为该类应用提供应用引擎、增强服务、云 API 权限、应用市场等功能；适用于自主基于PaaS平台开发SaaS的场景。

代码来源

代码仓库

蓝鲸可视化开发平台

S-mart 应用

基本信息

应用 ID *

请输入 3-16 字符的小写字母、数字、连字符(-)，以小写字母开头

应用的唯一标识，创建后不可修改

应用名称 *

由汉字、英文字母、数字组成，长度小于 20 个字符

应用模板

模板来源

蓝鲸开发框架

蓝鲸插件

Python开发框架

NodeJS开发框架

蓝鲸应用开发框架（Django3.x）

Python + Django3.x，集成用户鉴权、日志处理等基础功能模块

应用创建模板参考：

创建应用

```

graph LR
    A[普通应用] --- B[云原生应用]
    A --- C[外链应用]
  
```

① 平台为该类应用提供应用引擎、增强服务、云 API 权限、应用市场等功能；适用于自主基于PaaS平台开发SaaS的场景。

代码来源

代码仓库

蓝鲸可视化开发平台

S-mart 应用

基本信息

应用 ID *	<input type="text" value="sysu-bkdemo1"/>
应用的唯一标识，创建后不可修改	
应用名称 *	<input type="text" value="蓝鲸平台部署示例"/>

应用模板

模板来源

蓝鲸开发框架

蓝鲸插件

Python开发框架

NodeJS开发框架

☒ 蓝鲸应用开发框架 (Django3.x)

Python + Django3.x, 集成用户鉴权、日志处理等基础功能模块

☐ Django3.x 框架

Python + Django3.x

源码管理

代码源


Git 代码库


SVN 代码库

源代码地址 *

https://gitee.com/1024720328/bkdemo1.git

代码仓库地址

部署目录

请输入

Git账号 *

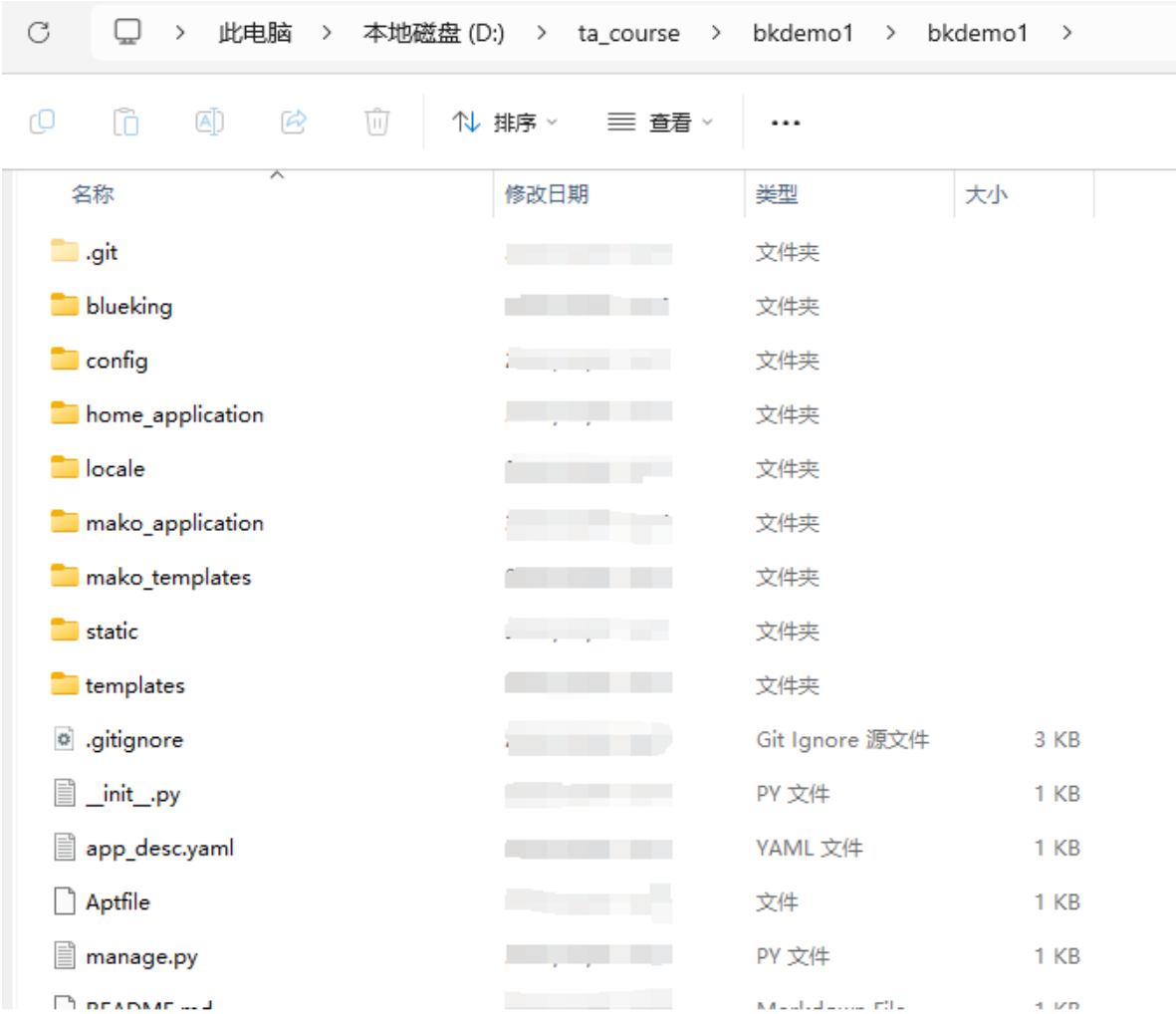
密码 *

4. 下载蓝鲸平台开发框架代码

在开发者中心的应用开发里的模块管理中下载模板代码：



把压缩包的文件全部解压到本地仓库的项目目录下：

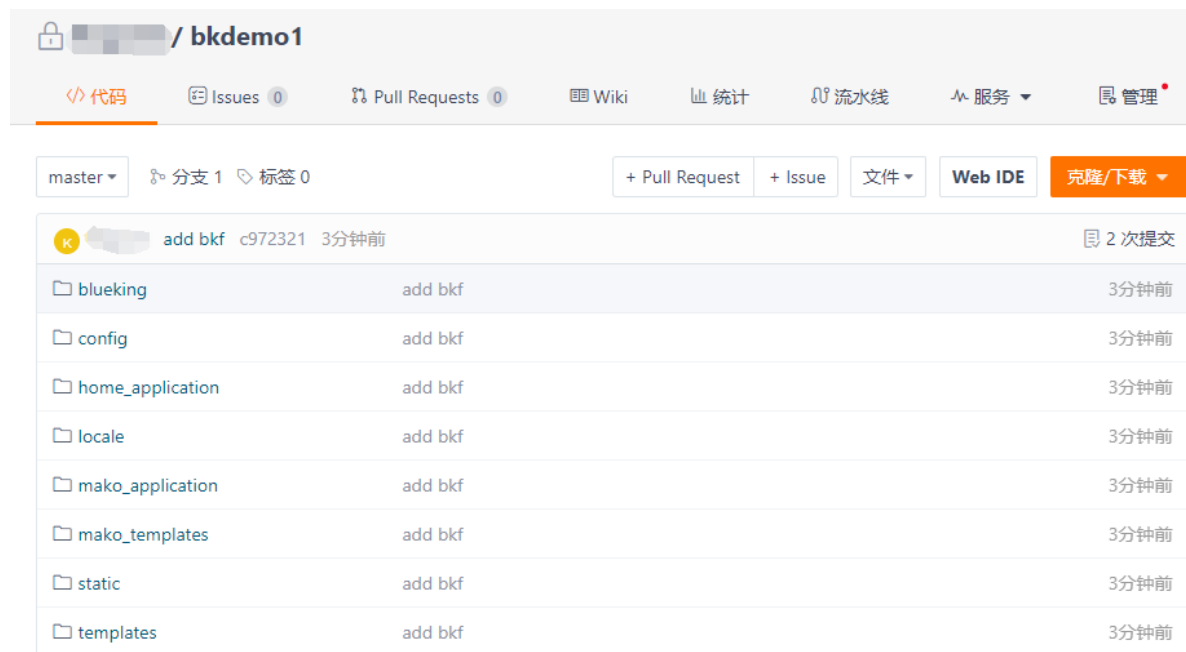


在Git上把本地仓库更新的文件同步到线上仓库上，在项目目录下进行操作：

```
abc@DESKTOP-E444806 MINGW64 /d/ta_course/bkdemo1/bkdemo1 (master)
```

```
git add .
git commit -m "add bkf"
git push
```


成功后在线上仓库能看到更新后的文件：



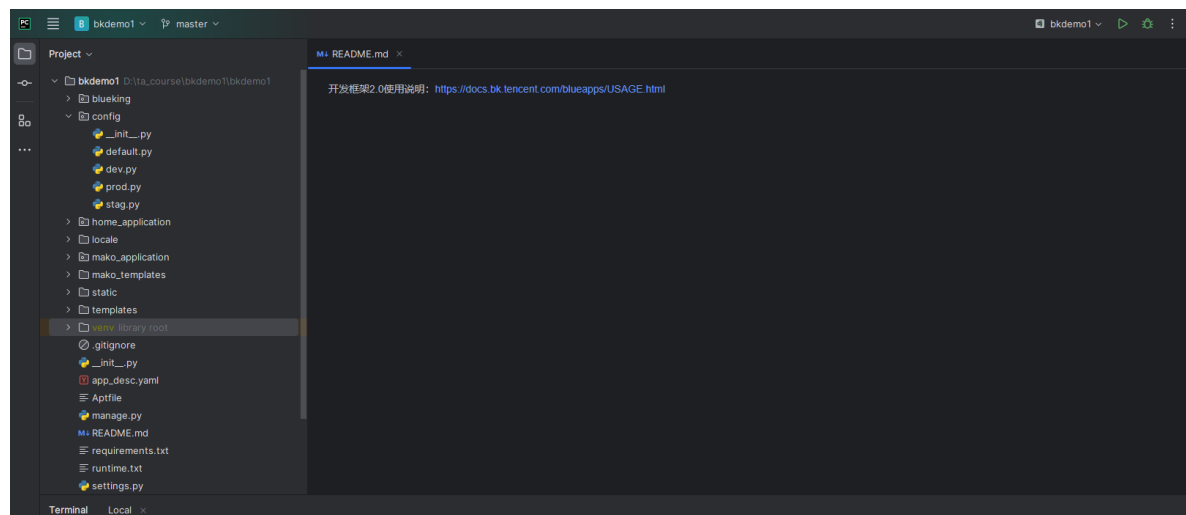
安装依赖包，使用cmd或者Git都可以：

```
pip install -r requirements.txt
```

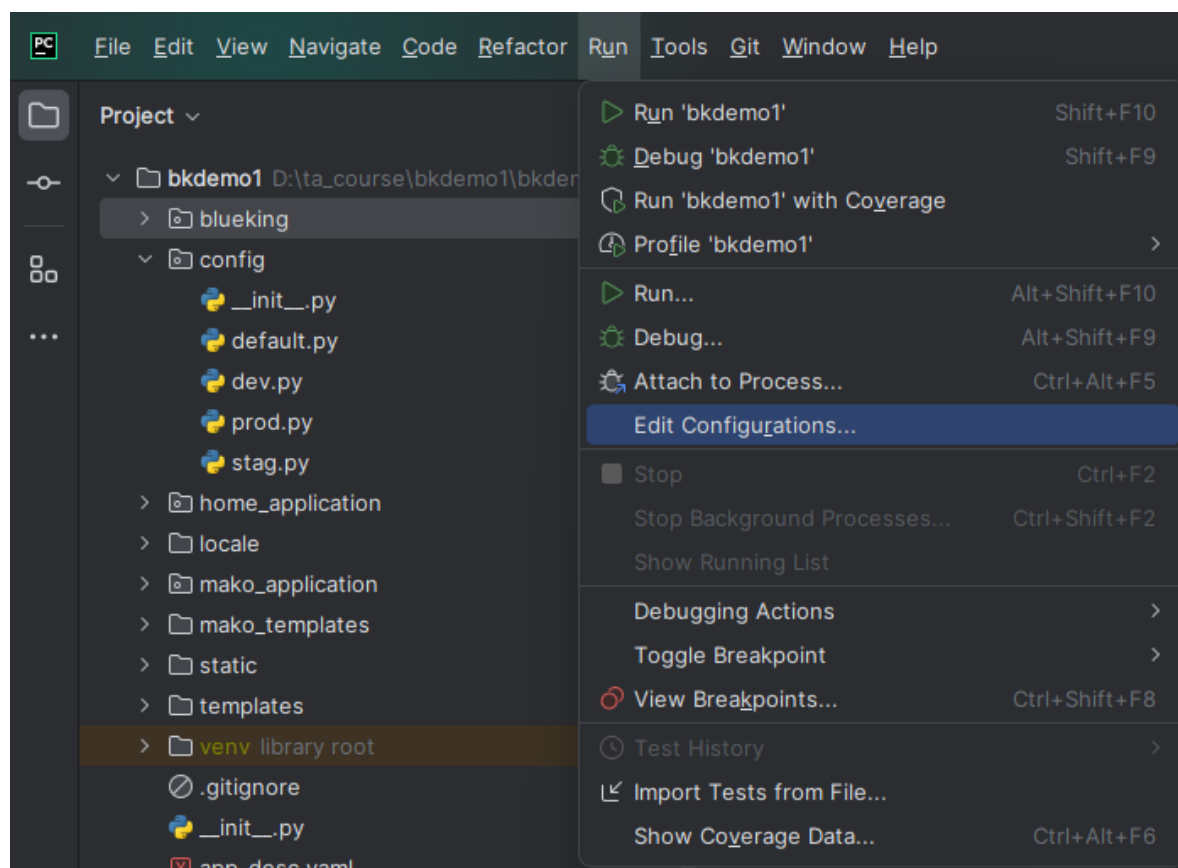
5. 本地创建数据库，修改框架本地运行配置

5.1 使用Pycharm进行数据库创建和修改配置

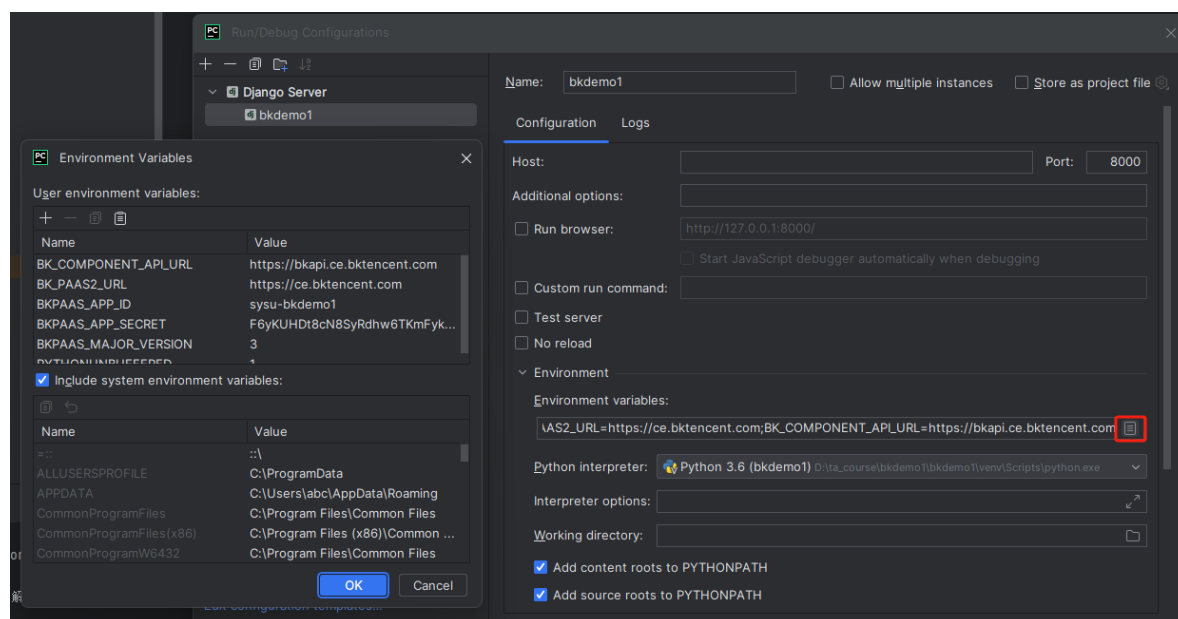
使用Pycharm打开项目目录后会提示创建虚拟环境，点击ok即可，界面应该和下图类似：



在Run中点击Edit Configurations选项修改配置：



点击修改环境变量：



添加变量：

```
BK_COMPONENT_API_URL=https://bkapi.ce.bktencent.com
BK_PAAS2_URL=https://ce.bktencent.com
BKPAAS_APP_ID='xxx'
BKPAAS_APP_SECRET='xxx'
BKPAAS_MAJOR_VERSION=3
```

BKPAAS_APP_ID和BKPAAS_APP_SECRET的值可以在蓝鲸平台的开发者中心的应用开发里的基本信息中查看，用来让蓝鲸平台鉴权：

蓝鲸开发者中心

首页应用开发API 网关服务

蓝鲸平台部署...
sysu-bkdemo1

概览

应用引擎

增强服务

模块管理

云 API 管理

应用推广

基本设置

基本设置

成员管理

基本信息

基本信息

管理员、开发者和运营者可以修改应用名称等基本信息

应用logo

更换图片

支持jpg、png等图片格式，图片尺寸为72*72px，不大于2MB。

应用名称

蓝鲸平台部署示例

创建时间

应用描述文件

已启用应用描述文件 app_desc.yaml，可在文件中定义环境变量，服务发现等。[文档：什么是应用描述文件](#)

鉴权信息

在调用蓝鲸云 API 时需要提供应用鉴权信息。使用方法请参考：[API调用指引](#)

bk_app_code

sysu-bkdemo1

bk_app_secret

F6yKUHDt8cN8SyRdhw6TKmFyk

删除应用

只有管理员才能删除应用，请在删除前与应用其他成员沟通

删除应用

该操作无法撤回

复制刚配置好的环境变量：

Run/Debug Configurations

Django Server

bkdemo1

Name: bkdemo1

Allow multiple instances

Store as project file

Configuration

Logs

Host:

Port: 8000

Additional options:

Run browser:

http://127.0.0.1:8000/

Start JavaScript debugger automatically when debugging

Custom run command:

Test server

No reload

Environment

Environment variables:

HDt8cN8SyRdhw6TKmFyk9DvP60W2Xk;BKPAAS_MAJOR_VERSION=3;PYTHONUNBUFFERED=1

Python interpreter: Python 3.6 (bkdemo1) D:\ta_course\bkdemo1\bkdemo1\venv\Scripts\python.exe

Interpreter options:

Working directory:

Add content roots to PYTHONPATH

Add source roots to PYTHONPATH

Edit configuration templates...

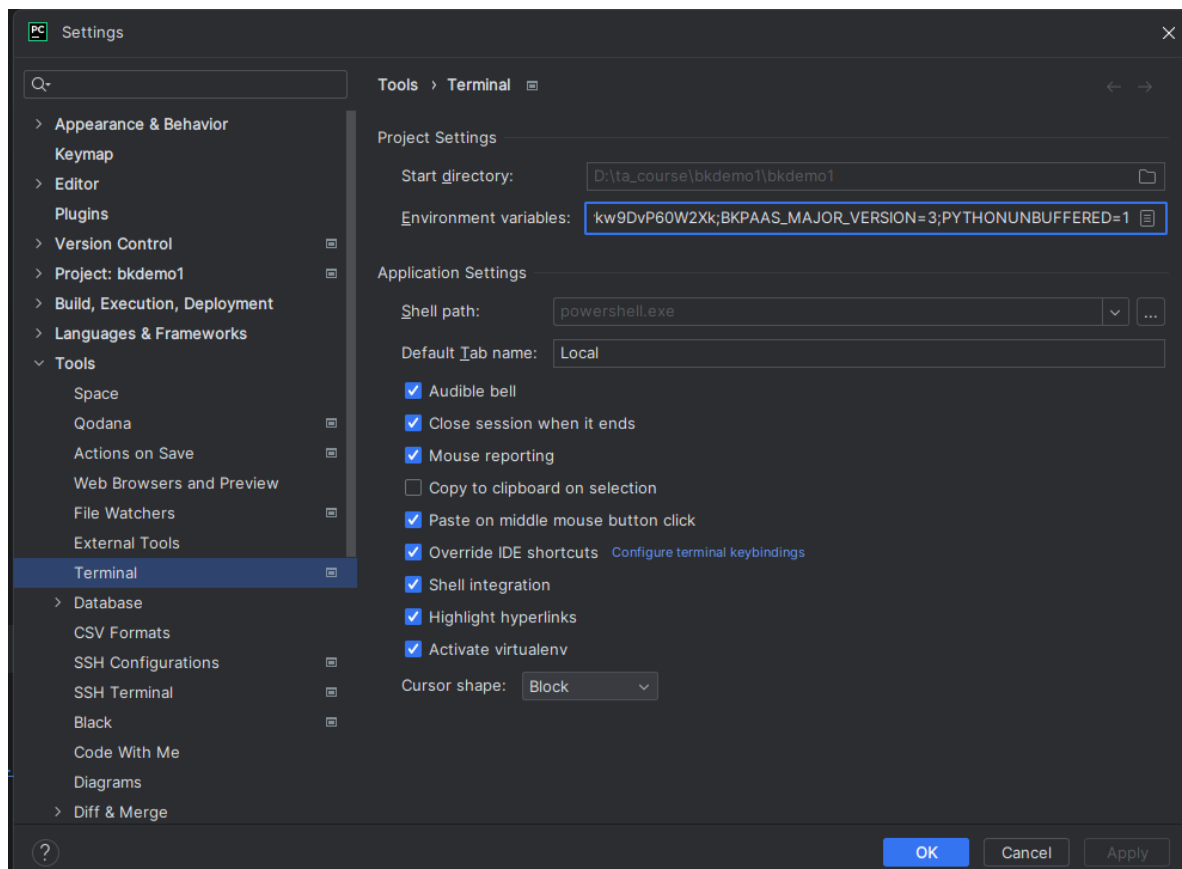
Run

OK

Cancel

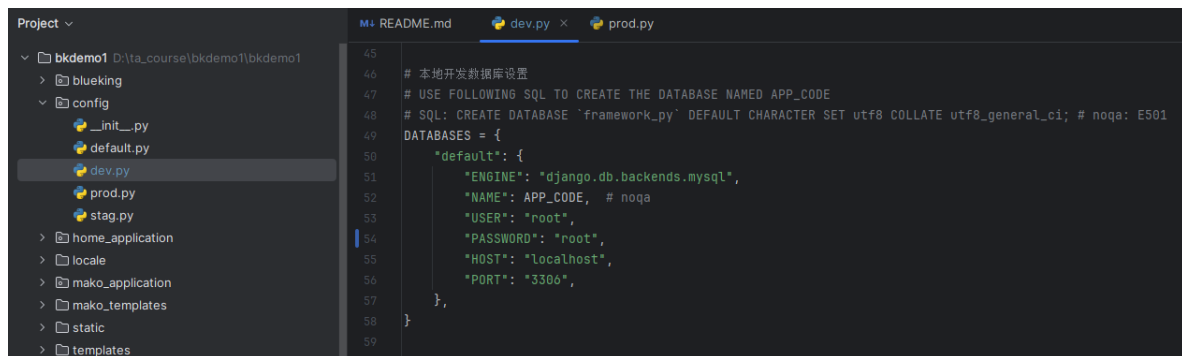
Apply

在Settings中的Terminal里的环境变量里粘贴：



重启Pycharm。

在config文件夹的dev.py文件中添加连接mysql的密码（mysql密码可以参考网上来设置更改）：



5.2 创建数据库

在mysql控制台执行下面指令，APP_CODE为前面用来鉴权的bk_app_code：

```
CREATE DATABASE `{APP_CODE}` default charset utf8 COLLATE utf8_general_ci;
```

添加后查看：

```
mysql> show databases;
+-----+
| Database |
+-----+
|          |
|          |
|          |
|          |
|          |
| sysu-bkdemo1 |
+-----+
7 rows in set (0.01 sec)
```

在Pycharm命令行中执行下面指令后添加数据库表信息：

```
python manage.py migrate
```

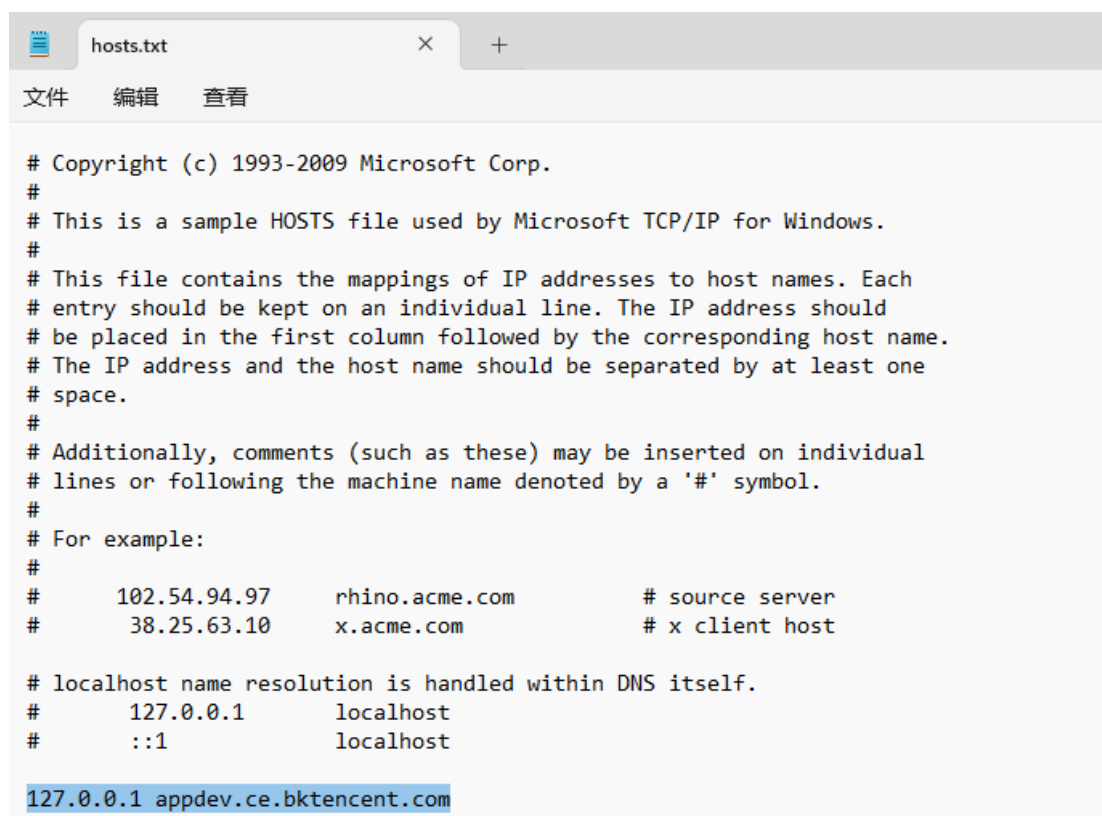
```
mysql> use sysu-bkdemo1
Database changed
mysql> show tables;
+-----+
| Tables_in_sysu-bkdemo1 |
+-----+
| account_cache |
| account_user |
| account_user_groups |
| account_user_property |
| account_user_user_permissions |
| account_verifyinfo |
| auth_group |
| auth_group_permissions |
| auth_permission |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_session |
| django_site |
+-----+
14 rows in set (0.00 sec)
```

6. 配置host信息

在C:/Windows/System32/drivers/etc目录下有个hosts文件，用来配置本地DNS信息，添加项：

```
127.0.0.1 appdev.ce.bktencent.com
```

如下图所示，可以把hosts文件后缀更改为txt来修改：



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10       x.acme.com               # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#       ::1              localhost
127.0.0.1 appdev.ce.bktencent.com
```

7. 运行框架

在Pycharm命令行中输入下面指令运行框架：

```
python manage.py runserver appdev.ce.bktencent.com:8000
```

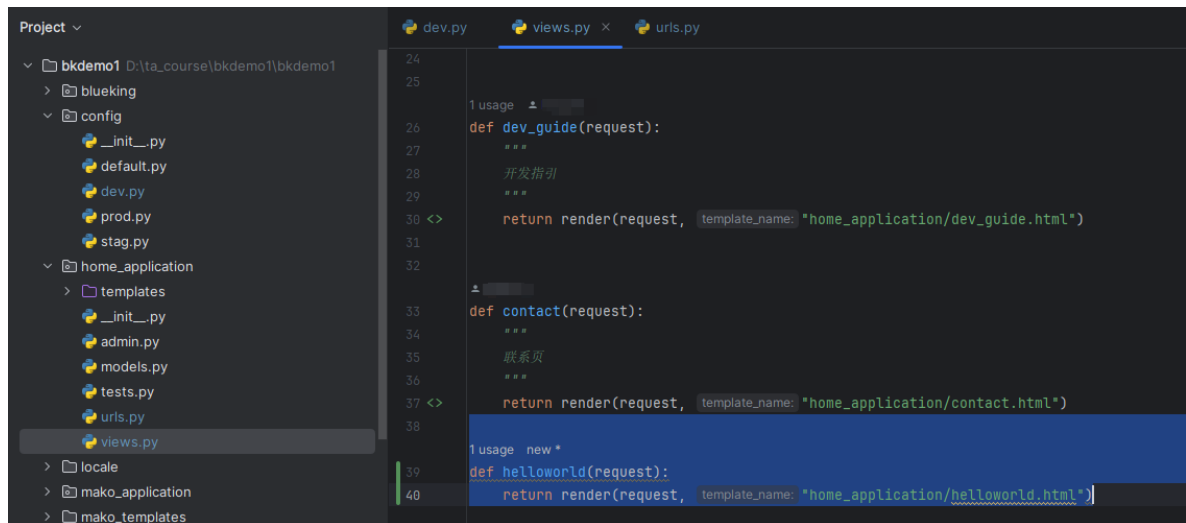
访问<http://appdev.ce.bktencent.com:8000/>并且登录，可以看到以下页面就成功：



8. 编写helloworld页面并访问

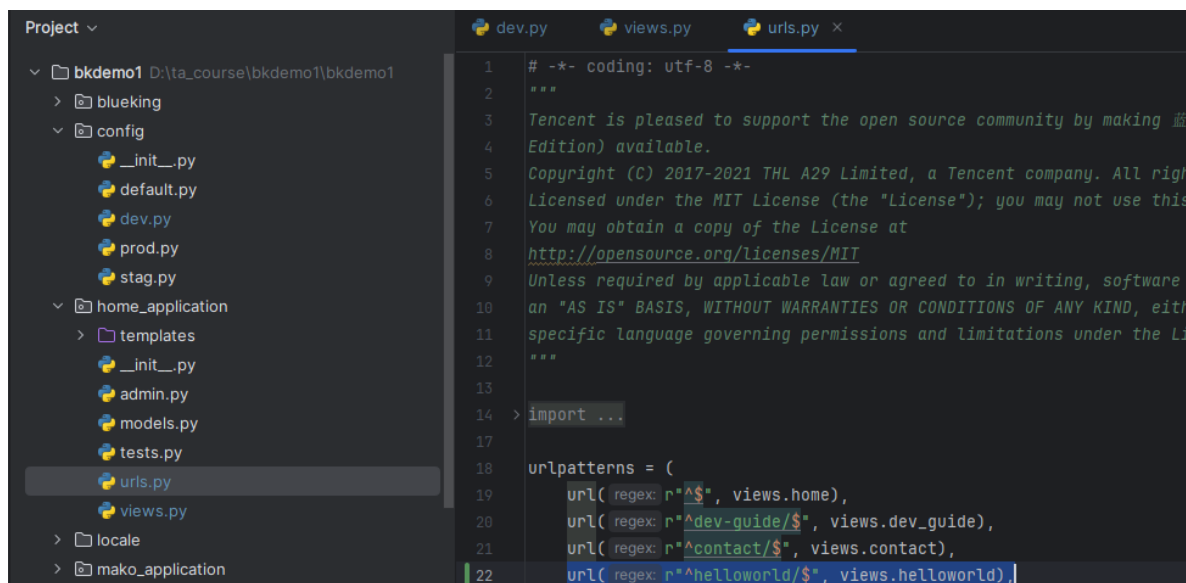
在项目目录下的home_application/views.py文件中添加：

```
def helloworld(request):  
    return render(request, "home_application/helloworld.html")
```



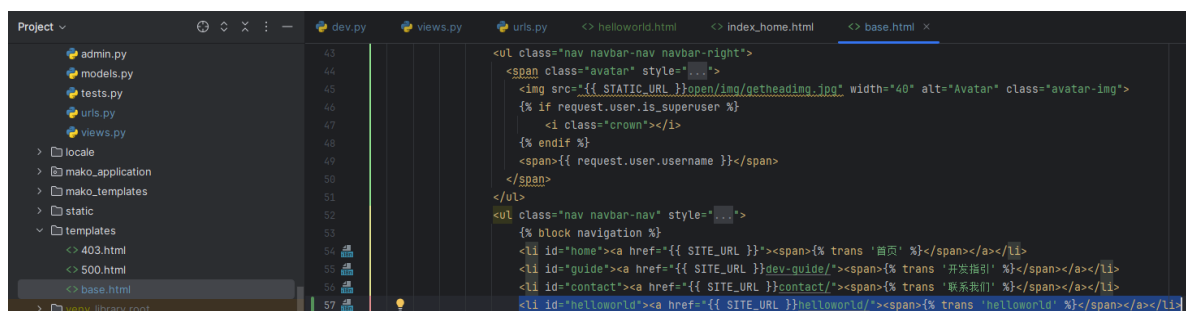
在项目目录下的home_application/urls.py文件的urlpatterns里添加：

```
url(r"^helloworld/$", views.helloworld),
```

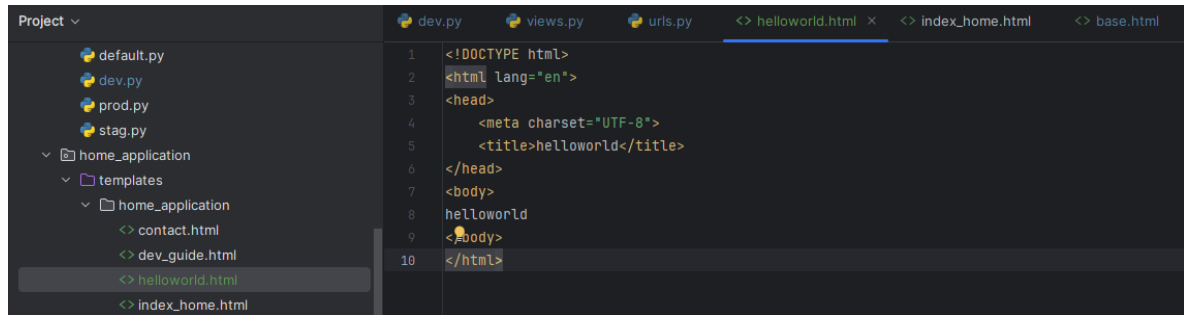


在项目目录下的templates/base.html文件里添加：

```
<li id="helloworld"><a href="{{ SITE_URL }}helloworld/"><span>{% trans  
'helloworld' %}</span></a></li>
```



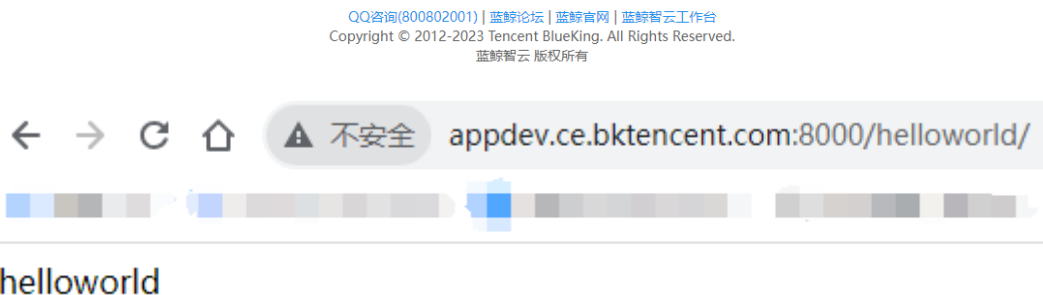
在项目目录下的home_application/templates/home_application创建helloworld.html文件：



```
1 <!DOCTYPE html>
2 <html lang='en'>
3 <head>
4   <meta charset='UTF-8'>
5   <title>helloworld</title>
6 </head>
7 <body>
8   helloworld
9 </body>
10</html>
```

重新运行框架：

```
python manage.py runserver appdev.ce.bktencent.com:8000
```

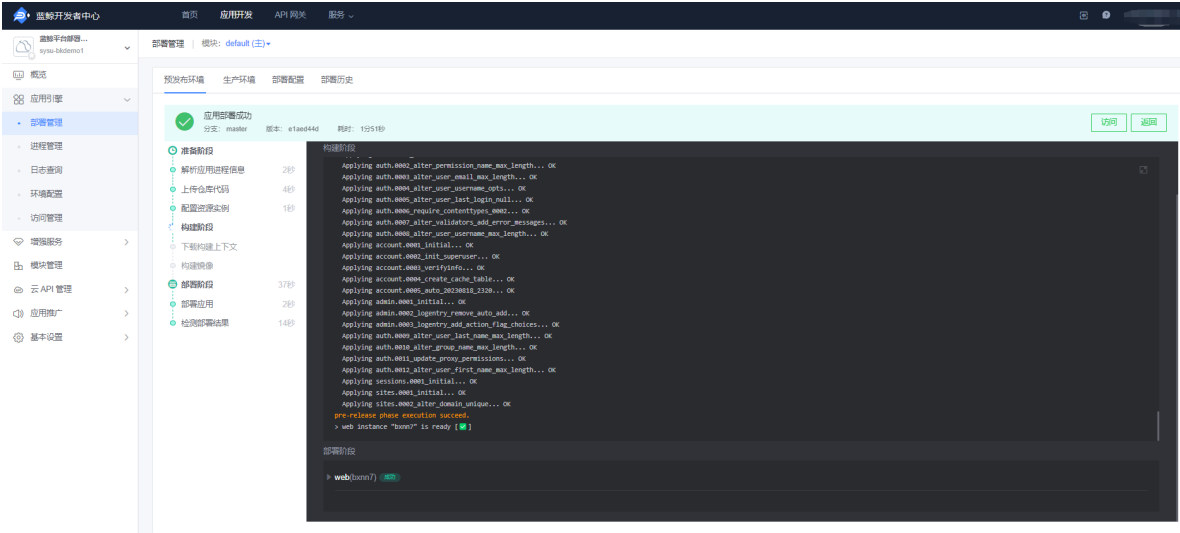


9. 把代码推送到仓库，分别部署到测试环境和正式环境

在Git中的项目目录下输入：

```
git add .
git commit -m "add helloworld"
git push
```


在蓝鲸平台的开发者中心的应用开发里的部署管理中，在预发布环境下点击部署，部署成功如下图所示：



生产环境部署也类似，最终访问页面：



作业提交方式

提交如下图所示的截图，右上角会显示自己账号，欢迎同学们把遇到的问题和建议写到作业里：



参考

视频教程：<https://www.bilibili.com/video/BV1bB4y1G7gW/>

文档教程：<https://bk.tencent.com/docs/markdown/ZH/PaaS/DevelopTools/SaaSGuide/DevBeginner/Windows.md>