

FISCO BCOS即拆即用的“Python区块链盒子”

原创 汤会枫 FISCO BCOS开源社区 2月21日



汤会枫

FISCO BCOS开源社区贡献者

— AUTHOR — 作者 —

作者为Python开发者，将FISCO BCOS Python SDK封装成镜像，以提升配置环境的速度，提高易用性。

FISCO BCOS

前言

作为一个 Python 开发者，我一直想通过 Python 去了解区块链。机缘巧合下，在 2019 年一次开源年会上，我接触并加入了 FISCO BCOS 开源社区，此后，一直用课余时间琢磨 FISCO BCOS 的 Python-SDK。

在配置环境的时候，我耗费了一些时间，因此也萌生了将整个框架封装成一个 docker 镜像的想法，我为它取名“Python 区块链盒子”，就像 Minecraft 的“工作台”一样，可以提升配置环境的速度，提高易用性。有了这样的想法，我便开始利用课余时间着手去编写 Dockerfile。

```
RUN python:3.7-alpine
```

```
LABEL 99kies 1290017556@qq.com https://github.com/99kies
```

```
RUN apk update
```

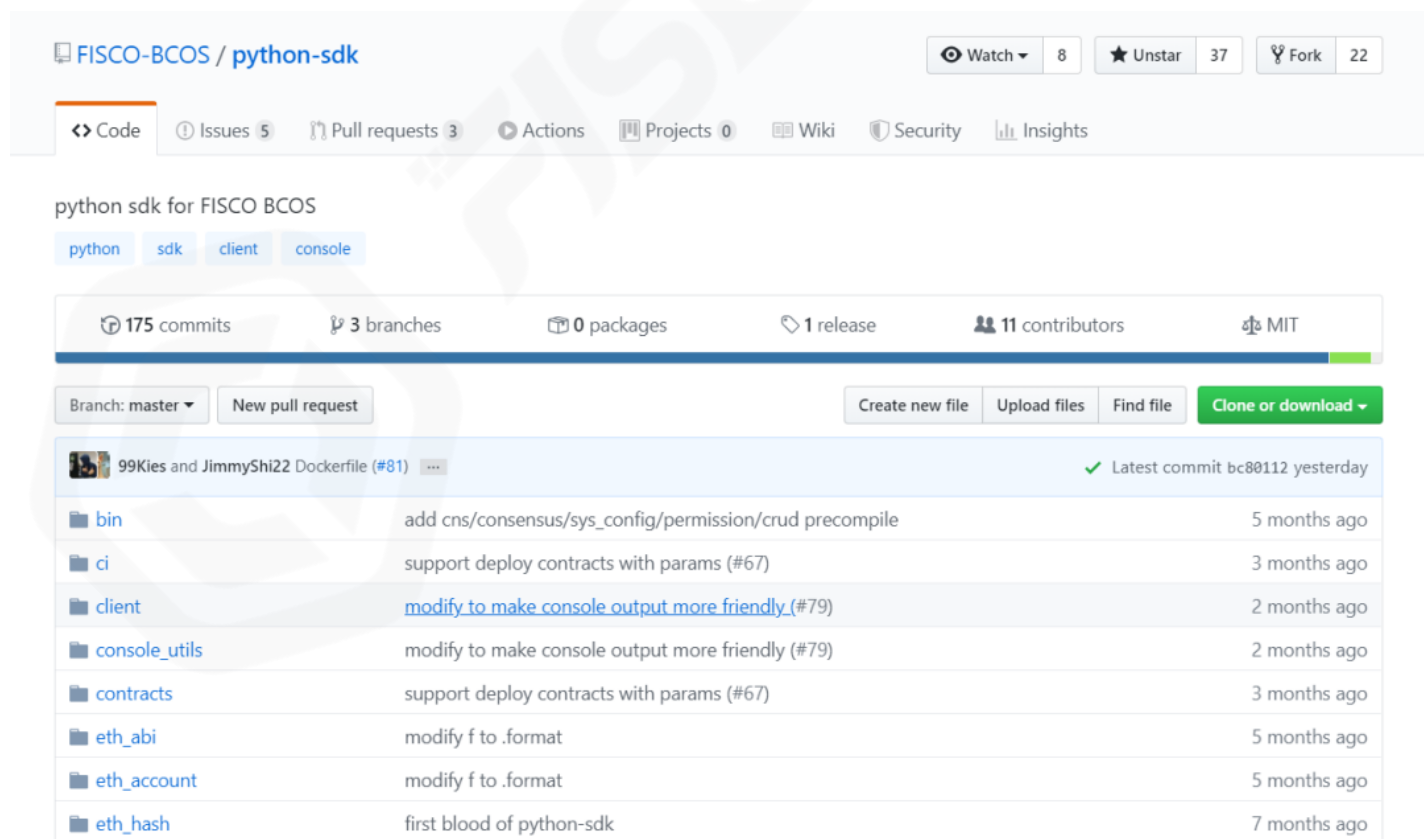
```
CMD ["bash"]
```

我把搭建好的 docker 镜像分享给周围的同学体验，“即拆即用”的特性反响很好👉，大家不会因为环境配置难而退怯，能像Minecraft里的Steve一样，放下“工作台”就可以制造出一堆有用的工具出来。

什么是“Python 区块链盒子”？

//////////

在回答这个问题之前，先来了解一下 Python-SDK。这是由 FISCO BCOS 开源，帮助开发者运用 Python 语言开发区块链应用的组件。既然是通过 Python 语言开发，我相信它将有持续不断的生命力。



FISCO-BCOS / python-sdk

Watch 8 Unstar 37 Fork 22

Code Issues 5 Pull requests 3 Actions Projects 0 Wiki Security Insights

python sdk for FISCO BCOS

python sdk client console

175 commits 3 branches 0 packages 1 release 11 contributors MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

99Kies and JimmyShi22 Dockerfile (#81) Latest commit bc80112 yesterday

bin	add cns/consensus/sys_config/permission/crud precompile	5 months ago
ci	support deploy contracts with params (#67)	3 months ago
client	modify to make console output more friendly (#79)	2 months ago
console_utils	modify to make console output more friendly (#79)	2 months ago
contracts	support deploy contracts with params (#67)	3 months ago
eth_abi	modify f to .format	5 months ago
eth_account	modify f to .format	5 months ago
eth_hash	first blood of python-sdk	7 months ago

Python 的阅读难度相对较低，特别方便学生、初学者通过 Python-SDK 去了解、学习区块

链。

可尝试按以下环境要求安装搭建。

- Python 环境：python 3.6.3, 3.7.x
- FISCO BCOS 节点：请参考 FISCO BCOS 安装搭建

https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/installation.html

关于 Python-SDK，可点击参考 FISCO BCOS 团队推出的开发教程。

[《麻雀虽小五脏俱全 | 从 Python-SDK 谈谈 FISCO BCOS 多语言 SDK》](#)

[《Python-SDK 的前世今生》](#)

“Python 区块链盒子”相当于将配置好的 Python-SDK 和部署好的区块链打包成一个包裹，类似于将一个大房子整理成一辆房车。通过这种方式，用户无需关注环境配置问题，可减少部署时间——不用一分钟就能获得镜像，方便开发者快速上手使用，便于自动化运维。

你也可以尝试将这个服务和 JenKins 结合，进一步优化运维过程。

“Python 区块链盒子”可以当成 Python 区块链开发的工作台，一个现拆现用的工具箱，而且可以保证开发环境的整洁，最重要的是它很轻，就像 Minecraft 里的工作台。





图片来源于网络

只要你有一台计算机，就可以随时随地打开这个工具箱。安装工具箱的过程也变成了享受：只需一行代码就能上手直接研究开发，不必过多关注环境配置的繁杂，能节省大量时间，解放双手。

获得并运行“Python 区块链盒子”

可以通过以下部署来获得这个“盒子”。

```
1 docker run -it -p 20200:20200 --name python_sdk fiscoorg/playground:py
```

进入容器后，首先需要启动节点，启动节点后就可以“食用”了。在这个过程中，请注意不要使用 sh。

```
1 bash /root/fisco/nodes/127.0.0.1/start_all.sh
```

随后，在/python_sdk 中进行调试，调试过程中的小贴士：console.py 添加了 tab 自动补全。

```
1 # 查看 SDK 使用方法
2 ./console.py usage
3
4 # 获取节点版本
5 ./console.py getNodeVersion
```

```
1 bash-5.0# ./console.py getNodeVersion
2
3 INFO >> user input : ['getNodeVersion']
4 INFO >> getNodeVersion >> { "Build Time": "20190923 13:22:09", "Build
```

完成这些之后就成功了，前前后后，相当于在几秒内就搭建了一个开源的区块链框架。你可以把自己的合约放到/python-sdk/contracts 中，更多关于 Python-SDK 的使用方法可以参考以下链接。

<https://github.com/FISCO-BCOS/python-sdk>

这个即拆即用的“区块链盒子”对于要使用 Python 来开发区块链应用或者学习区块链的开发者很有帮助。开发者可以通过调用/python-sdk/client 中的函数使用 ./console.py 命令行并与盒子中正在运行的区块链进行交互。

下面将会以 Python-Flask 开发作为例子，实现调用 HelloWorld 合约的功能。

- step1 进入容器

```
1 docker run -it -p 20200:20200 -p 80:80 --name flask_web fiscoorg/playg
```

- step2 启动节点

```
1 bash /root/fisco/nodes/127.0.0.1/start_all.sh
```

- step3 部署 HelloWorld 合约

/python-sdk/contract 下存放了 HelloWorld.sol，可以直接使用这个合约进行测试。

首先，查看 HelloWorld.sol 合约的内容。

```
1 pragma solidity ^0.4.24;
2
3 contract HelloWorld{
4     string name;
5
6     constructor() public{
7         name = "Hello, World!";
8     }
9
10    function get() constant public returns(string){
11        return name;
12    }
13
14    function set(string n) public{
15        name = n;
16    }
17 }
```

可以通过 get 接口返回字符串 name 的值，通过 set 更新 name 的值。有 Solidity 编程基础的开发者还可以自己定制合约内容。

然后，就可以部署 HelloWorld 合约。

```

1 $ ./console.py deploy HelloWorld save
2
3 INFO >> user input : ['deploy', 'HelloWorld', 'save']
4
5 backup [contracts/HelloWorld.abi] to [contracts/HelloWorld.abi.201908
6 backup [contracts/HelloWorld.bin] to [contracts/HelloWorld.bin.201908
7 INFO >> compile with solc compiler
8 deploy result for [HelloWorld] is:
9 {
10     "blockHash": "0x3912605dde5f7358fee40a85a8b97ba6493848eae7766a8c31
11     "blockNumber": "0x1",
12     "contractAddress": "0x2d1c577e41809453c50e7e5c3f57d06f3cdd90ce",
13     "from": "0x95198b93705e394a916579e048c8a32ddfb900f7",
14     "gasUsed": "0x44ab3",
15     "input": "0x6080604052...省略若干行...c6f2c20576f726c642100000000000
16     "logs": [],
17     "logsBloom": "0x000...省略若干行...0000",
18     "output": "0x",
19     "status": "0x0",
20     "to": "0x0000000000000000000000000000000000000000000000000000000",
21     "transactionHash": "0xb291e9ca38b53c897340256b851764fa68a86f2a53cb
22     "transactionIndex": "0x0"
23 }
24 on block : 1,address: 0x2d1c577e41809453c50e7e5c3f57d06f3cdd90ce
25 address save to file: bin/contract.ini

```

完成后可得到 HelloWorld 合约部署的地址，通过这个地址调用函数接口。

- step4 在/python-sdk 文件夹下编辑 app.py

```

1 $ vi app.py
2
3 # -*- coding:utf-8 -*-
4 from client.common import transaction_common
5 from flask import Flask, request
6 from jinja2 import escape
7
8 app = Flask(__name__)
9
10 tx_client = transaction_common.TransactionCommon("0x2d1c577e41809453c
11 # tx_client = transaction_common.Transaction_Common("地址","合约路径","
12
13 @app.route('/')
14 def index():
15     '''
16     调用 get 接口, 获得 HelloWorld 的字符串
17     '''
18     new_str = request.args.get('new_str')
19     # 通过 new_str 的值更新 HelloWorld 的字符串
20     if new_str:
21         tx_client.send_transaction_getReceipt("set", (new_str,))
22         # 若 new_str 不为空则更新字符串
23     return escape(str(tx_client.call_and_decode("get")))
24
25 if __name__ == '__main__':
26     app.run(host="0.0.0.0", port=80)

```

- step5 安装 app.py 依赖/运行程序

```

1 pip install flask
2 python app.py

```

这样就通过 Python-Flask 框架实现了调用 HelloWorld 合约，通过调用 get 接口查看字符串，调用 set 接口更新字符串。

FISCO BCOS 的 Python-SDK 十分适合像我这样的学生党或者初学者去研究和了解区块链技术。十分期待更多开发者参与其中，用它搭建出更多有趣好玩的开源项目。

Dockerfile 地址请参考文末，最近我也会对其做一些更新，提升其操作易用性，最新操作手册和消息将在GitHub上发布，欢迎大家关注。

更多Python Demo请参考：

https://fisco-bcos-documentation.readthedocs.io/zh_CN/latest/docs/sdk/python_sdk/demo.html

3. 提交 pr 体验

//////////

关于如何在 FISCO BCOS 中提交 pr 可点击参考开源社区整理的内容，这里不多做说明。

《揭秘 FISCO BCOS 开源项目开发协作》

我想分享一下个人提交 pr 的体验，整个过程既新奇又有趣。

萌生“区块链盒子”的 idea 之后，我很快就编写了第一个 Dockerfile，随即就提交了 pr。很快社区的小哥哥石翔就回复了我，刚开始我以为是国外友人，就一直用 "Poor English" 和他交流。

他十分欢迎我的pr，还不断给我👍点赞和支持，他不但解决了许多在配置过程中的问题，还时不时分享一些小故事鼓励我。

在合并我的pr过程中，十分感谢团队的热情帮助😊，及时向我提出review意见，并细心给我介绍需要添加的内容功能，这让我充分感受到 FISCO BCOS 社区和睦共处、互帮互助的氛围。

目前，“Python 区块链盒子”还有一些有待优化的地方，例如，每次启动一个容器都需要手动操作，启动节点的条件是-->机器暴露所需端口，默认是 20200, 8045, 30300。

后期可能会考虑对默认启动节点的功能进行优化，添加数据卷，以及优化容器大小，使 fiscoorg/playground:python_sdk 更精致。

如果你有不错的优化想法，欢迎提交 pr，来一次有趣新奇的 pr 体验😊

参考链接

Dockerfile:

<https://github.com/FISCO-BCOS/python-sdk/blob/master/Dockerfile>

如对这个"盒子"有更多的问题或建议，欢迎加入FISCO BCOS技术交流群，与作者深入探讨。

进群请长按二维码添加小助手

微信ID: **FISCOBCOS010**



..... **FISCO BCOS**

FISCO BCOS的代码完全开源且免费

下载地址↓↓↓

<https://github.com/FISCO-BCOS/FISCO-BCOS>



FISCO BCOS



长按二维码关注
下载最新区块链应用案例集

