

VSCode Python 调试经验分享

存储研发部 五处
梁鑫辉

2021年3月

一、背景介绍

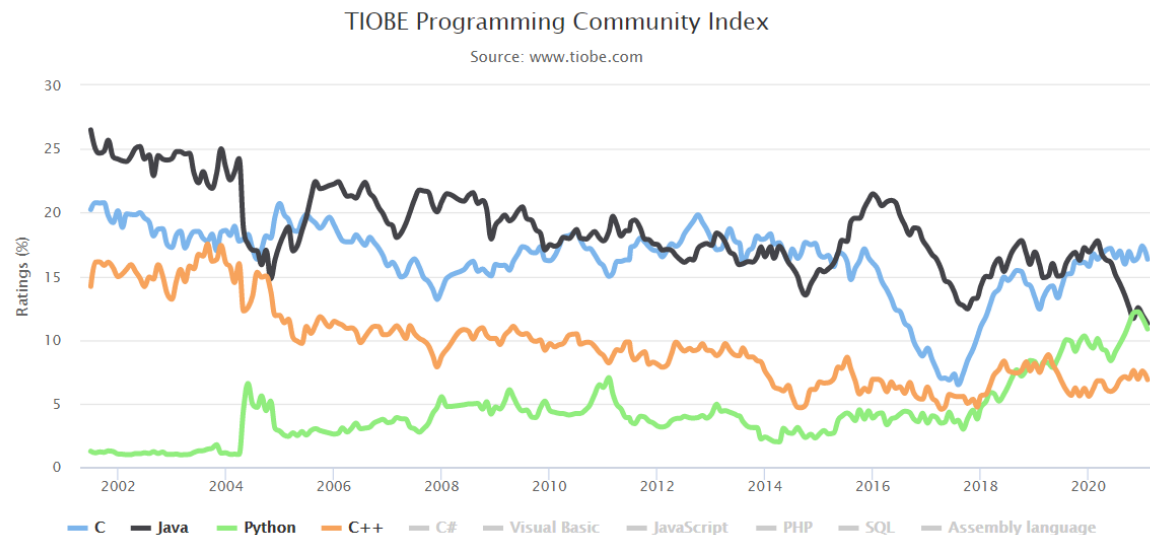
二、VSCode Python 调试

三、离线开发环境部署

四、特殊场景调试

选题背景 – Python 越来越流行

- Python 越来越流行

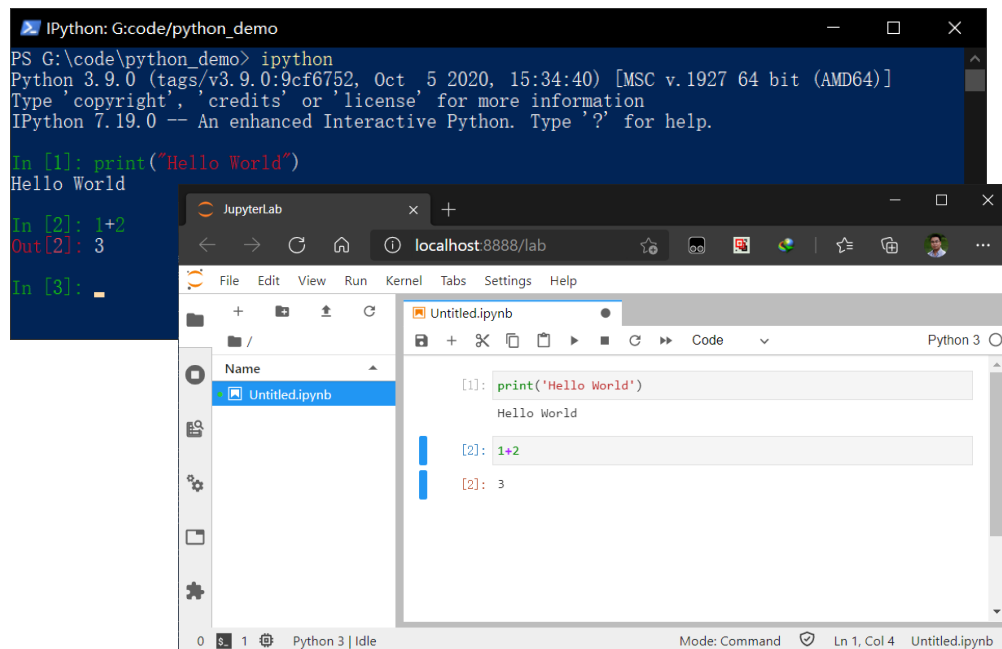


<https://www.tiobe.com/tiobe-index/>

- 存储五处（管理软件）使用 Python 的项目
 - InView / AI Ops 预研：磁盘故障预测、性能容量预测、SSD寿命预测
 - AS13000管理软件：CLI
 - 主机插件：OpenStack 存储插件、VMWare 存储插件
 -

Python 主流开发环境

- **纯文本编辑：**
 - 常用工具：vi/vim, 记事本, notepad++
 - 调试方式：日志 / gdb
 - 适用场景：通用
- **交互式环境：**
 - 常用工具：
 - Python 命令行：python
 - 命令行交互式环境：ipython
 - **Web交互式环境：Jupyter Notebook/ Lab**
- **IDE 集成开发环境：**
 - 常用工具：**PyCharm / VSCode**
 - 调试方式：断点调试
 - 适用场景：**代码量大 / 线上环境**



The image shows two overlapping windows. The top window is a terminal titled 'IPython: G:\code\python_demo'. It displays the IPython prompt and the execution of 'ipython', showing the Python version (3.9.0) and IPython version (7.19.0). It then shows the execution of 'print("Hello World")' resulting in 'Hello World', and '1+2' resulting in '3'. The bottom window is a JupyterLab interface titled 'JupyterLab' with a browser address bar showing 'localhost:8888/lab'. It displays a file browser on the left with 'Untitled.ipynb' selected. The main area shows the code from the IPython terminal, with the output 'Hello World' and '3' displayed below the code cells.



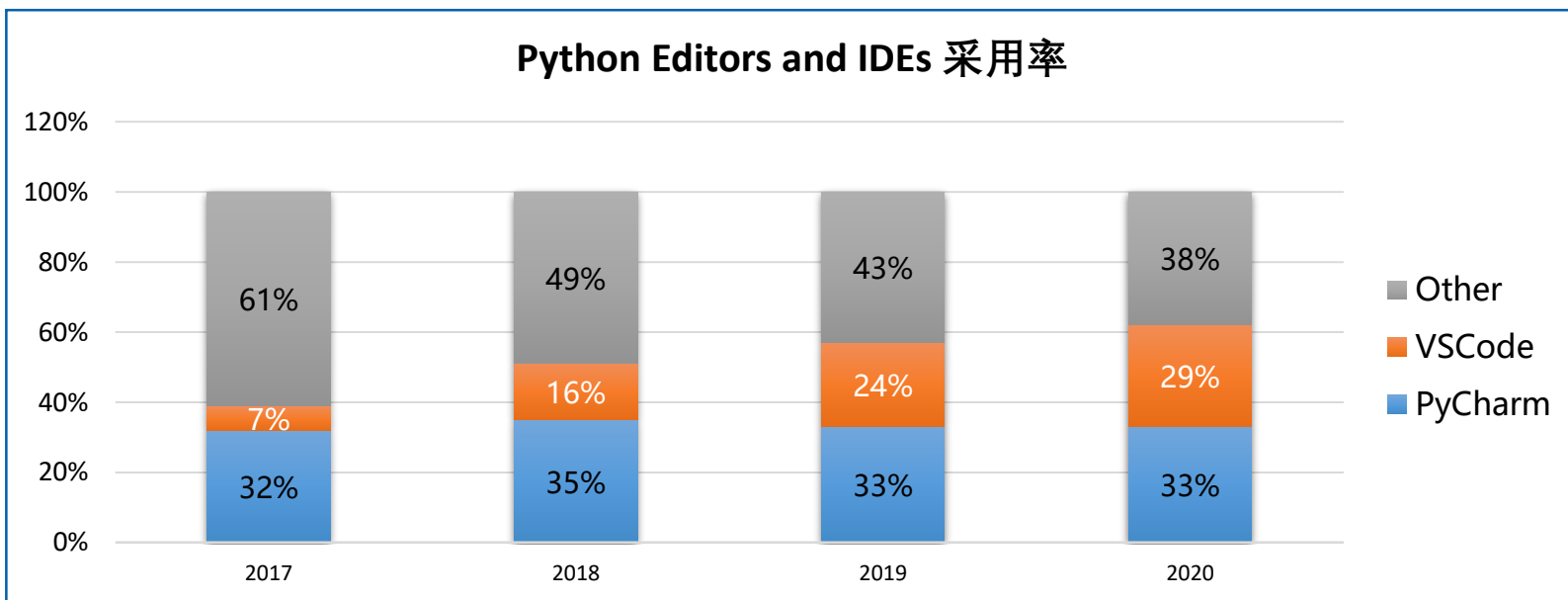
PyCharm



VSCode

Python IDE 对比

	PyCharm	VSCode
发布者	JetBrains	微软
授权	商业软件，高级特性收费	开源软件，免费
上手难度	低，开箱即用	稍复杂，需要简单配置
资源开销	大	少



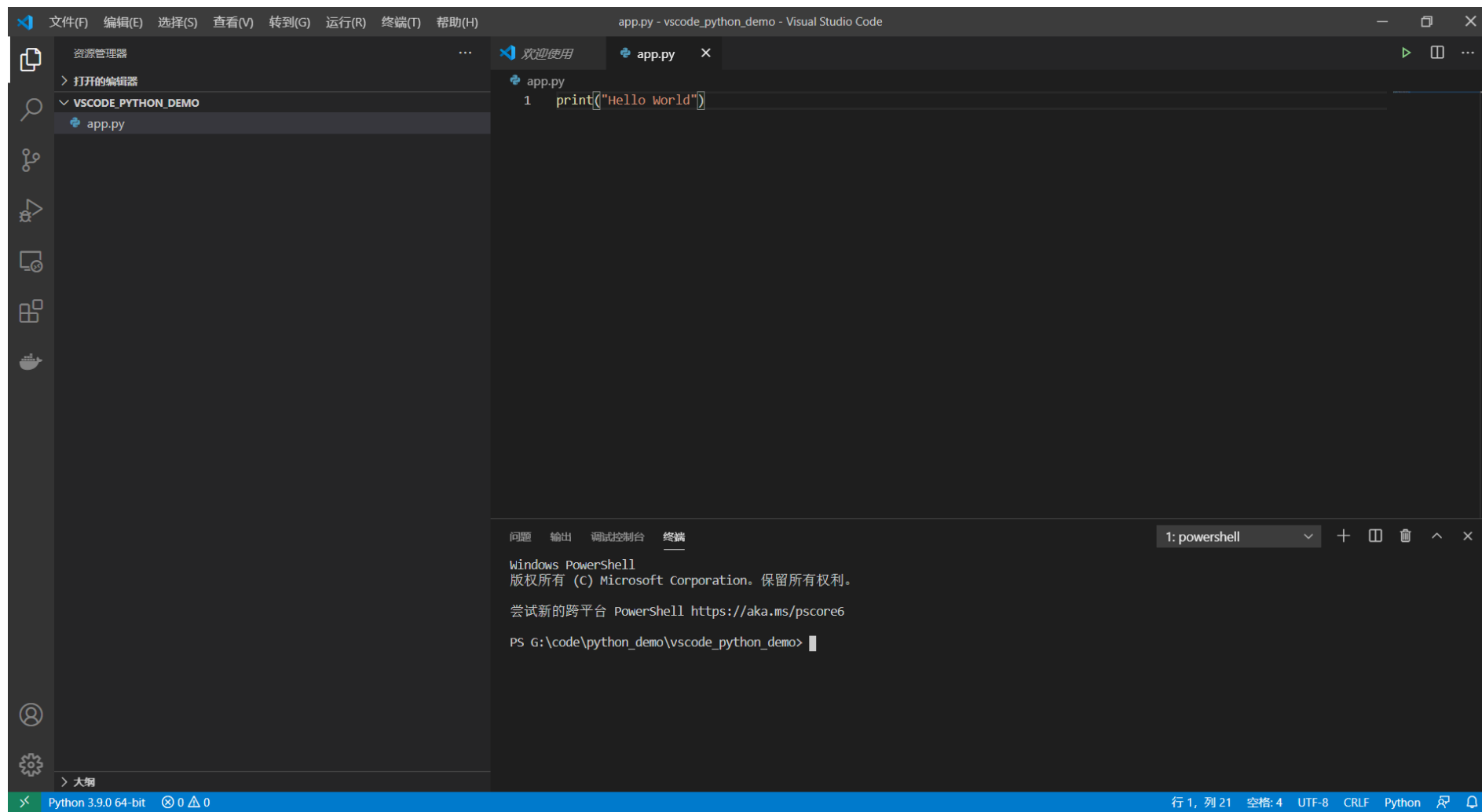
数据来源: <https://www.jetbrains.com/lp/python-developers-survey-2020>

VS Code 工作面板

- 资源管理器 →
- 全局搜索/替换 →
- 代码管理 →
- 调试 →
- 远程主机 →
- 扩展 →



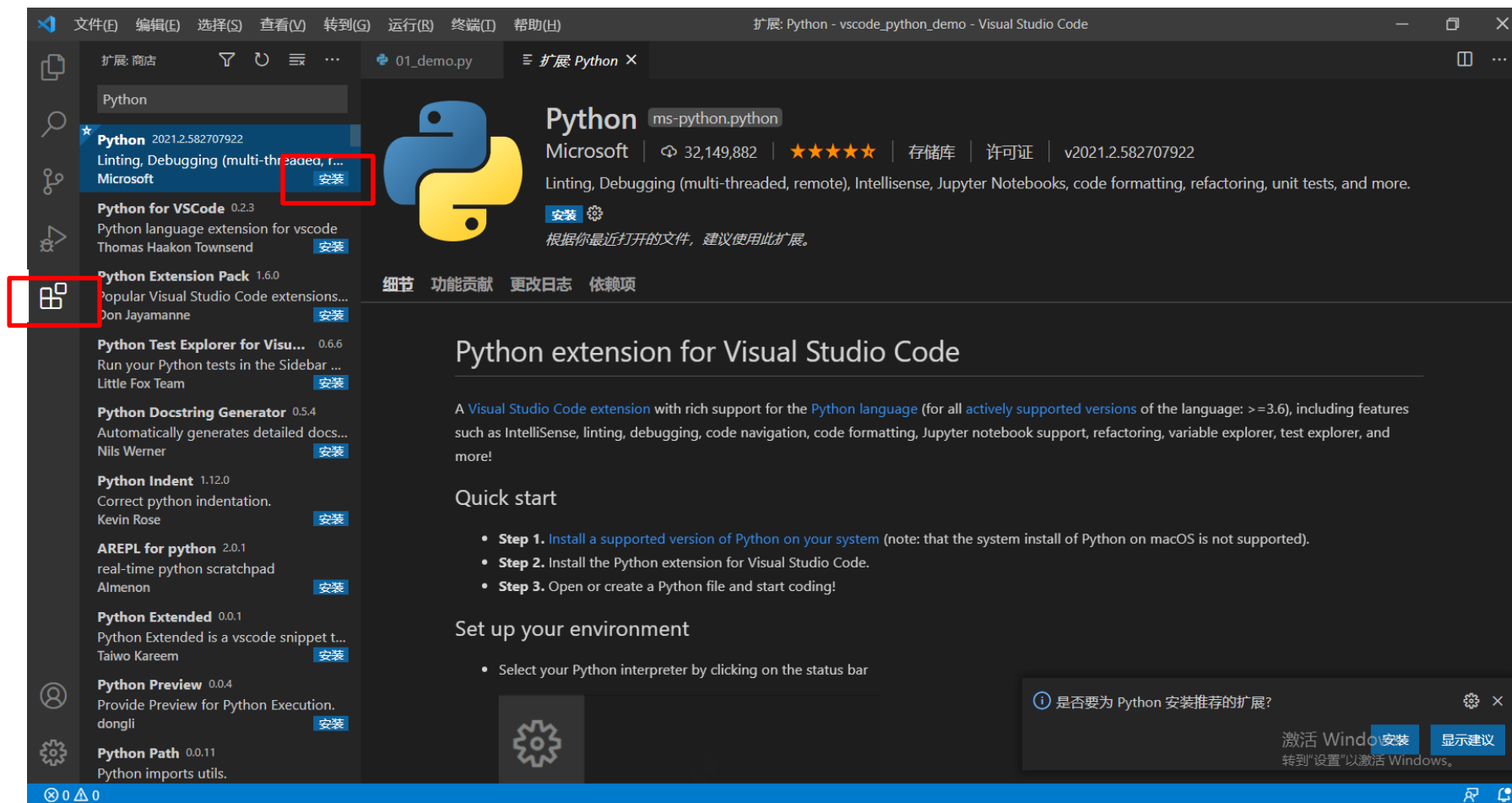
VSCode



下载地址: <https://code.visualstudio.com/>

VSCode 插件安装

- 丰富的插件库
- 官方支持各种主流编程语言
 - C/C++
 - Java
 - Python
 - Go
 - PHP
 -



一、背景介绍

二、VSCode Python 调试

三、离线开发环境部署

四、特殊场景调试

VSCode Python 调试 – 目录

- **调试场景**
 - 常见调试场景
 - 远程调试场景特点
- **VSCode Python 调试配置**
- **VSCode Python 调试实战**
 - 本地代码调试 – Demo展示
 - 远程代码调试
 - 原理介绍
 - 环境准备
 - Demo展示
 - 本地+远程联调
 - 原理介绍
 - 环境准备
 - Demo展示

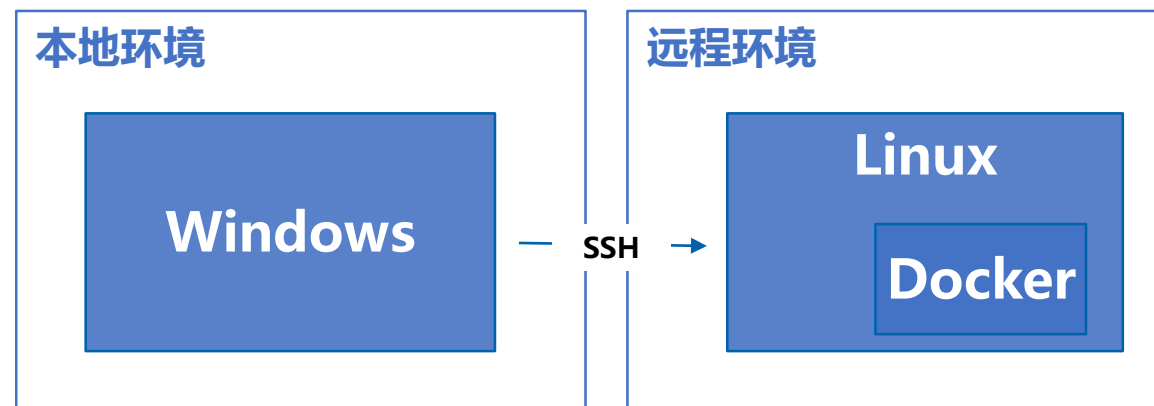
调试场景：常见调试场景

调试场景	典型场景	调试配置	依赖软件包	备注
本地代码调试	简单Python脚本。 开发环境与运行环境操作系统基本一致， 无操作系统或硬件依赖。	Python 文件 PID 连接	无	无
远程代码调试	源码在Linux远程环境管理。 或者调试、观察已在运行的项目。	Python 文件 PID 连接	VSCode-SSH 插件 Python 插件	VSCode 特有方案 适合 瘦终端 + 云端 环境 优点： 开发体验好，接近本地调试 缺点：离线部署稍麻烦
本地+远程联调	源码在Windows端，并同步到Linux远程 环境，程序运行在Linux端。	远程调试	Debugpy SFTP 插件	通用方案，PyCharm 远程调试也是类似原理 优点：远程服务端依赖少 缺点：需要维护代码同步；程序启动不方便；Debug启动操作步骤较多

调试场景：远程调试场景特点

需要远程开发的场景：

- 开发主机与运行环境差异大
 - OS 不同 (Windows / Linux)
 - 基础库不同 (dll/so, python版本)
- 有些问题只能在远程服务器上调试
 - 硬件关联信息 (服务器IPMI、物理硬盘SMART等)
 - 复杂网络环境 (万兆以太网、FC存储网等)
- 临时调试远程正在运行的程序

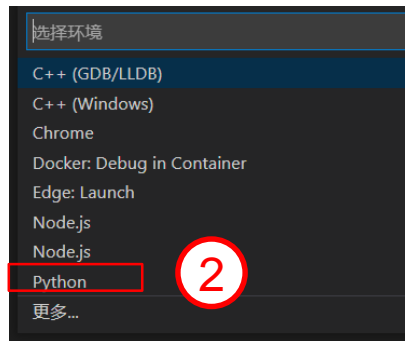


VSCode Python 调试配置

① 调试面板, 创建launch.json文件



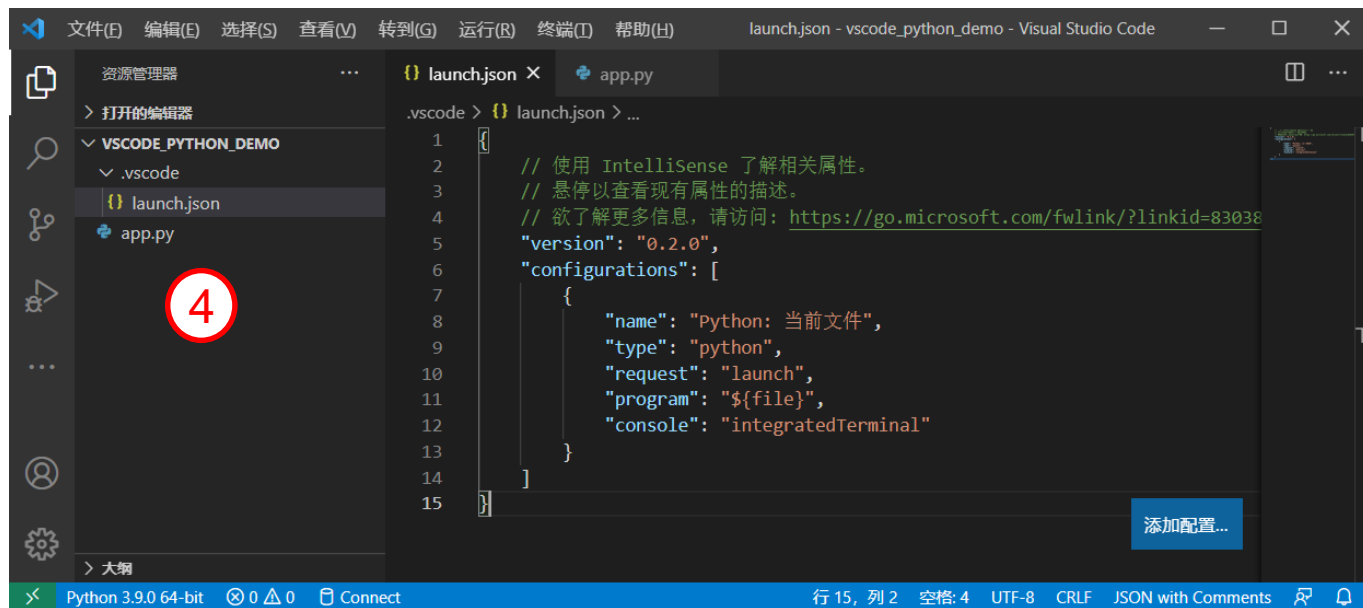
② 选择 环境 - Python



③ 选择调试配置



④ 修改调试配置



常用调试配置:

- **Python文件**
 - 重新启动一个python脚本, 开始调试
- **PID连接**
 - 连接到本地Python进程 (需要gdb)
- **远程调试**
 - 连接到运行中的Python调试网络端口

官方文档: <https://code.visualstudio.com/docs/python/debugging>

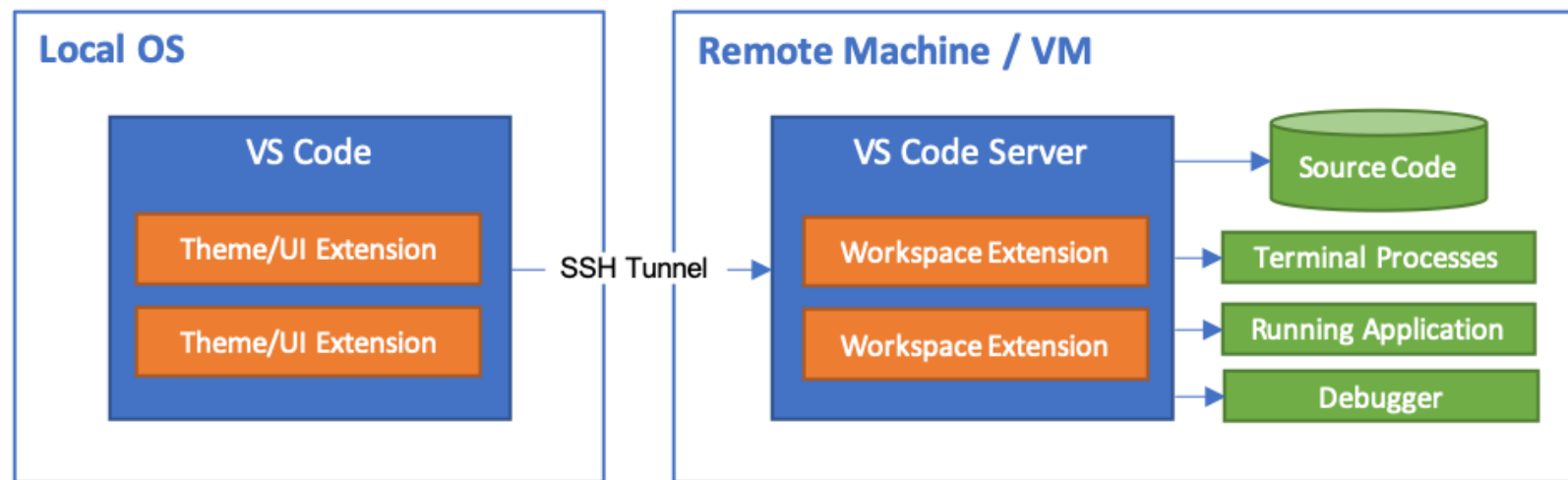
调试实战1：本地代码调试 – Demo展示



Demo

本地代码调试

调试实战2：远程代码调试 – 原理介绍



本地

- VS Code (仅作为UI界面)
- 不需要有源代码
- 扩展插件
- SSH Client
 - 建议使用 Git for Windows 内嵌ssh
 - Windows 10 的 SSH Client 兼容性不好

远程主机

- VS Code Server (真正的运行和调试环境)
- 需要有源代码
- 扩展插件
- SSH Server
- Python 环境
 - Python / Conda / virtualenv

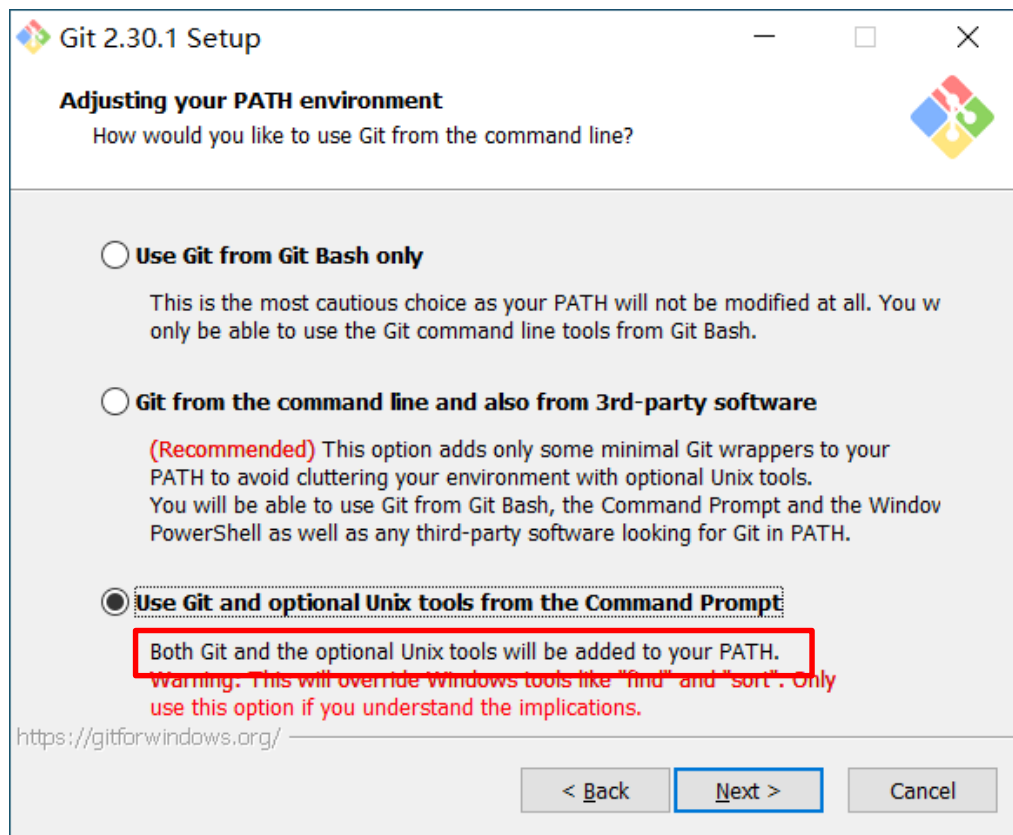
<https://code.visualstudio.com/docs/remote/remote-overview>

(不只是Python, C++、Java、Go、Python、PHP 等都是通用的)

调试实战2： 远程代码调试 – 环境准备 - SSH免密登录

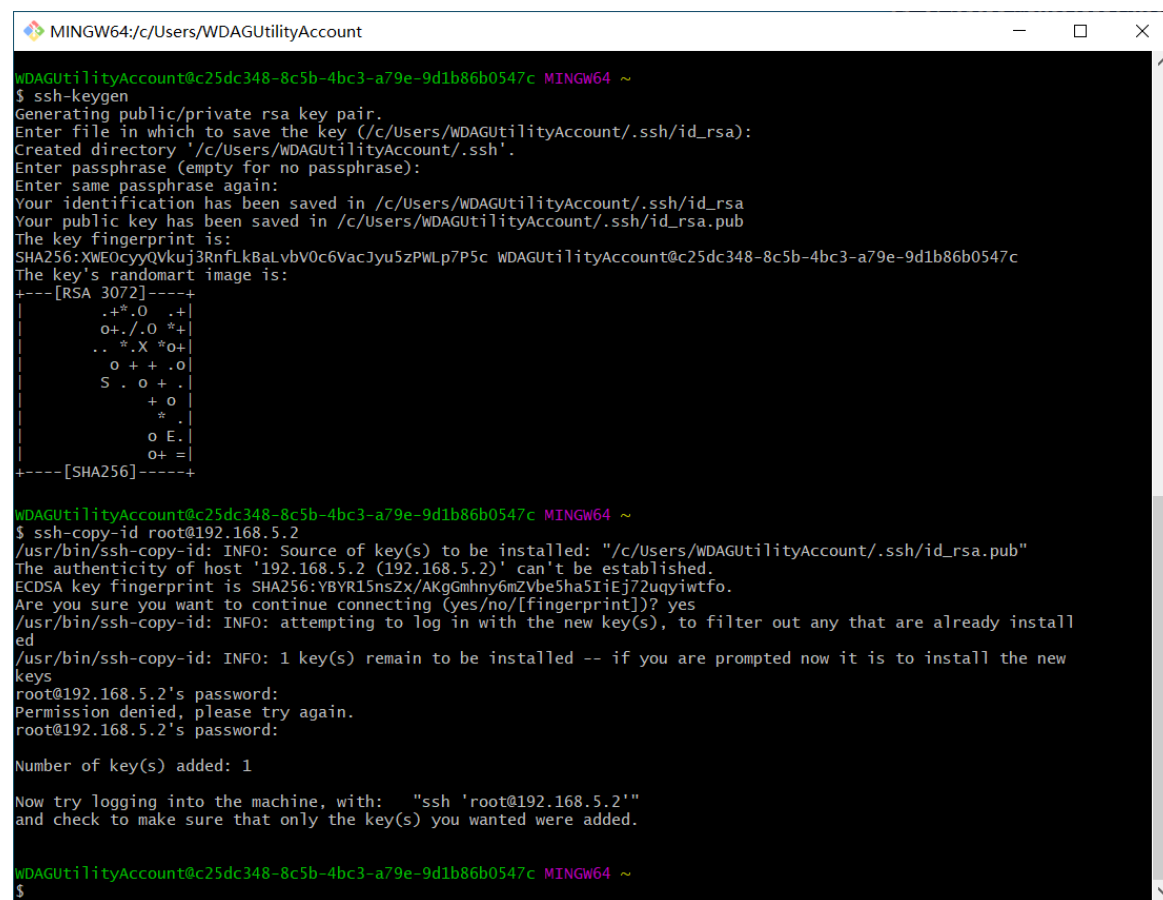
安装 ssh客户端

- Git for windows (推荐)
确保ssh在系统环境变量中，可直接在命令行执行



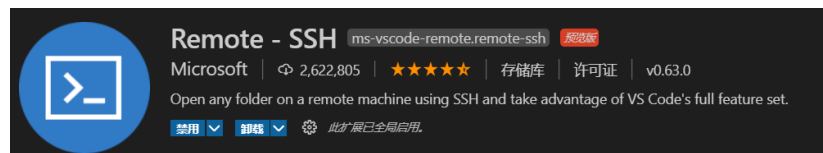
配置免密登录 (可选, 推荐)

- 生成Windows主机密钥
 - ssh-keygen
- 添加密钥到远程主机
 - 方法1: ssh-copy-id
 - 方法2: 手工复制到服务器 /root/.ssh/authorized_keys

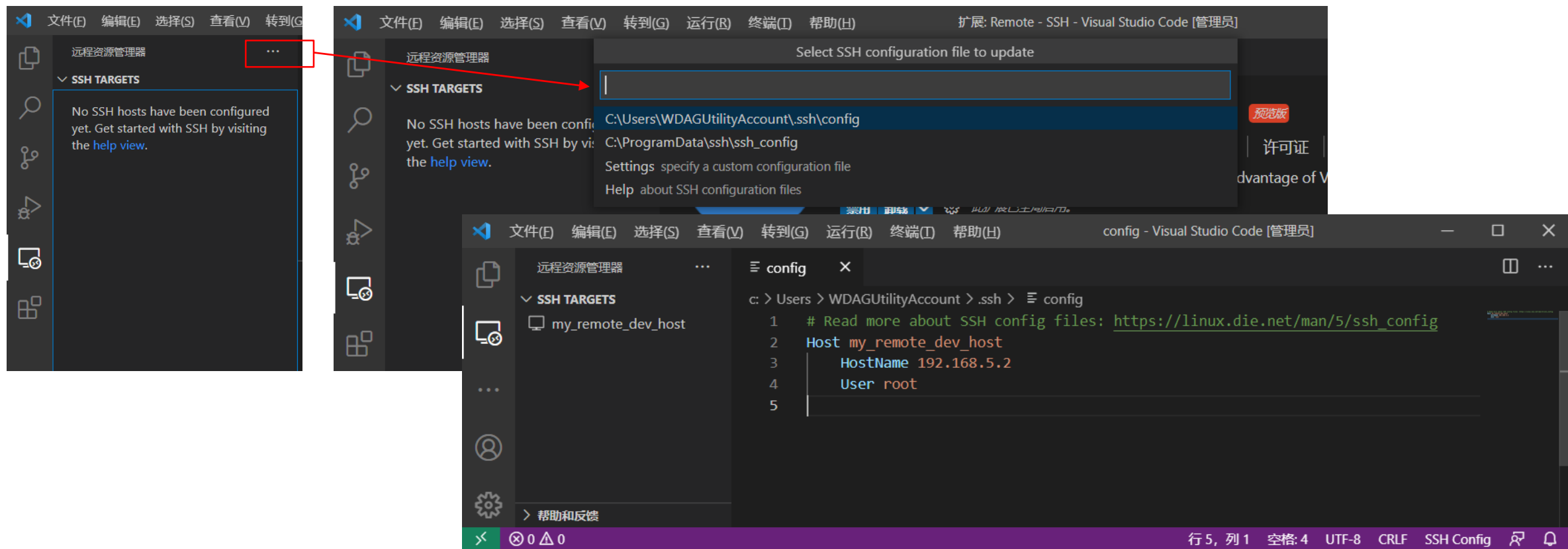


调试实战2： 远程代码调试 – 环境准备 – 添加 Remote 主机


① 安装 Remote-SSH 插件

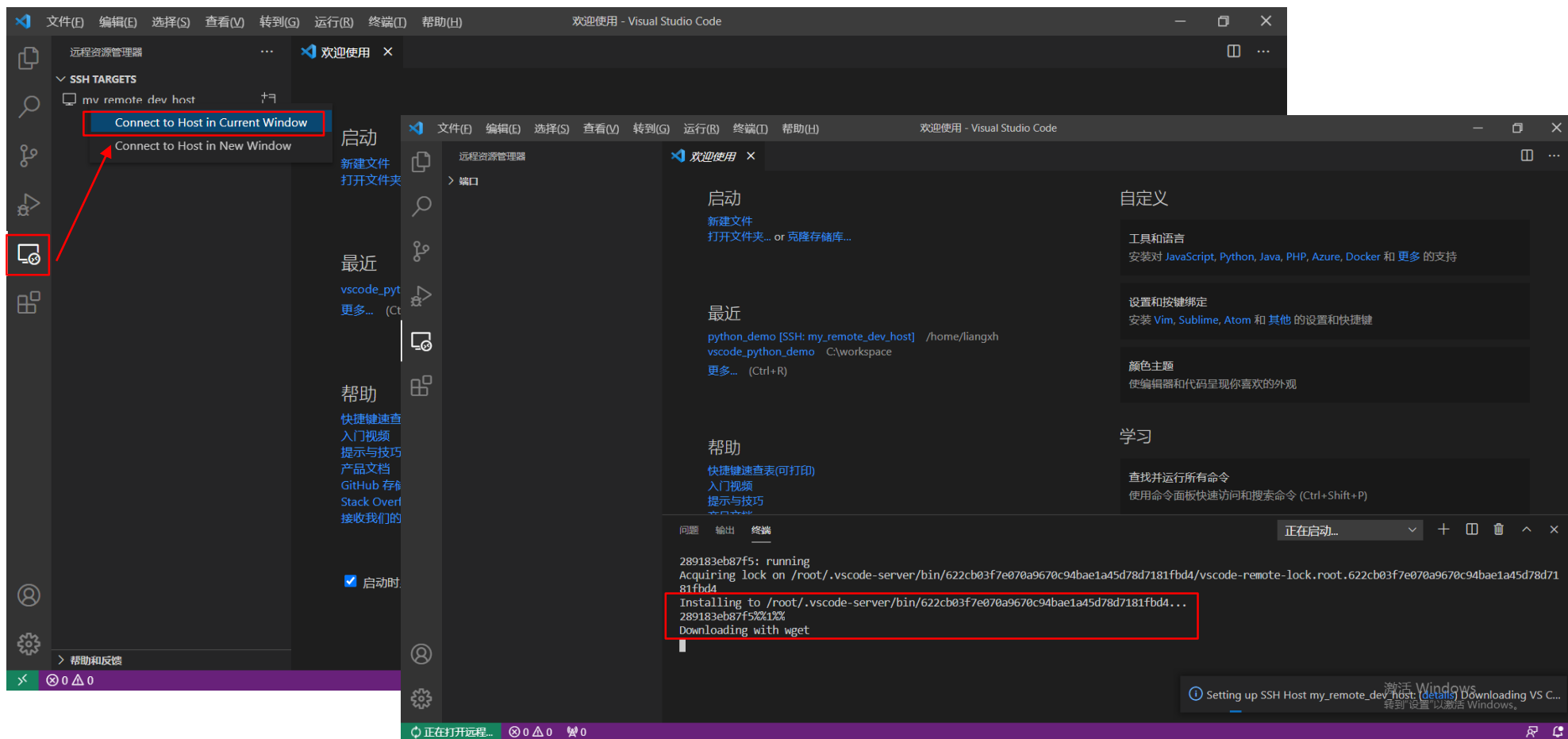


② 配置远程主机



调试实战2： 远程代码调试 – 环境准备 – 连接到远程主机

远程资源管理器面板 ，右键点击添加的主机，进行连接




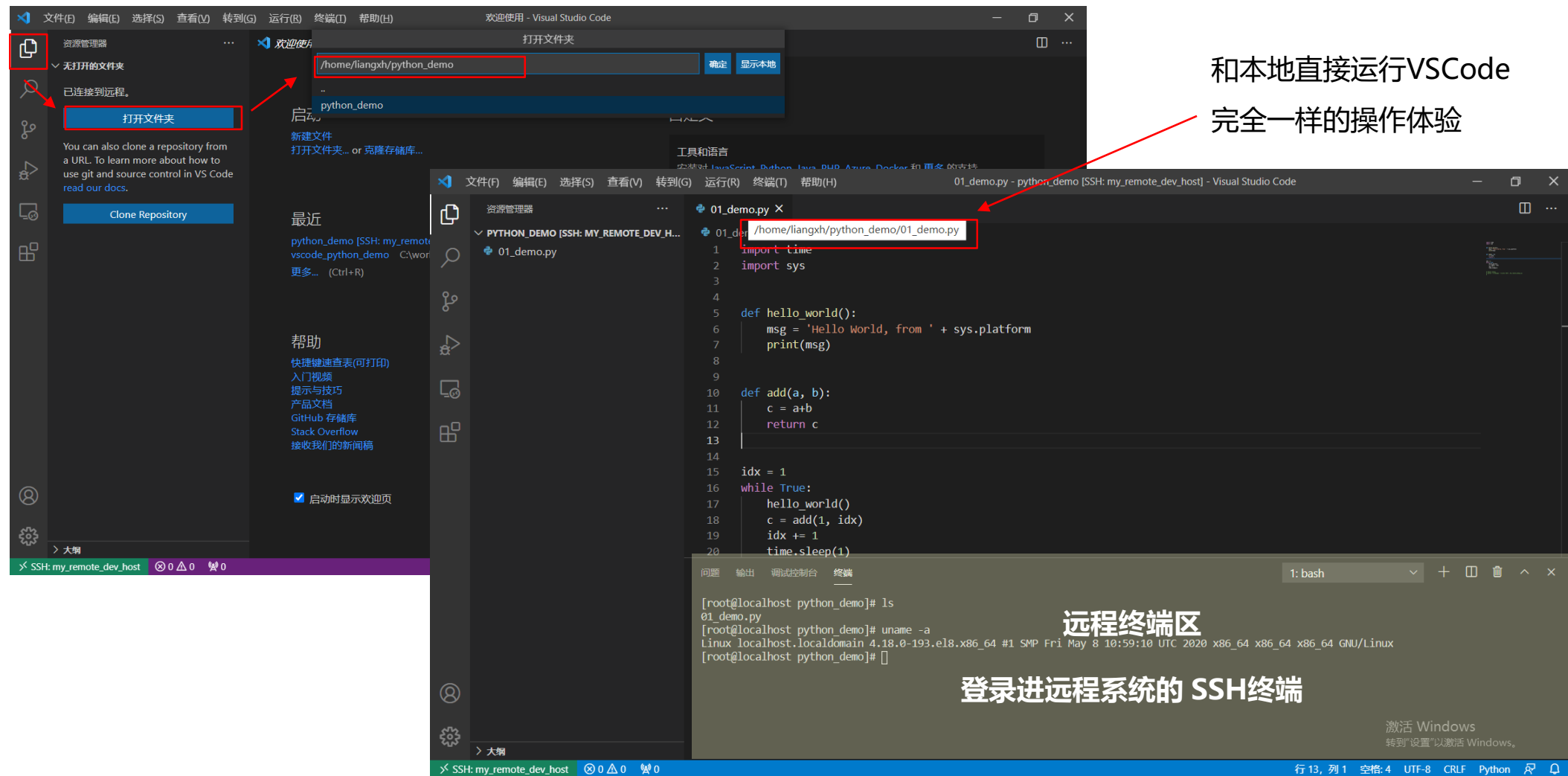
第一次连接时

联网环境会自动在远程主机安装 VSCode Server

离线环境安装参照离线部署章节

调试实战2： 远程代码调试 – 环境准备 - 打开远程目录

资源管理器面板  , 打开文件夹, 填写远程主机文件夹地址



和本地直接运行VSCode 完全一样的操作体验

资源管理器

无打开的文件夹

已连接到远程。

打开文件夹

Clone Repository

最近

python_demo [SSH: my_remote_dev_host]

vscode_python_demo C:\work

更多... (Ctrl+R)

帮助

快速键速查表(可打印)

入门视频

提示与技巧

产品文档

GitHub 存储库

Stack Overflow

接收我们的新闻稿

☒ 启动时显示欢迎页

01_demo.py X

/home/liangxh/python_demo/01_demo.py

```
1 import time
2 import sys
3
4
5 def hello_world():
6     msg = 'Hello World, from ' + sys.platform
7     print(msg)
8
9
10 def add(a, b):
11     c = a+b
12     return c
13
14
15 idx = 1
16 while True:
17     hello_world()
18     c = add(1, idx)
19     idx += 1
20     time.sleep(1)
```

远程终端区

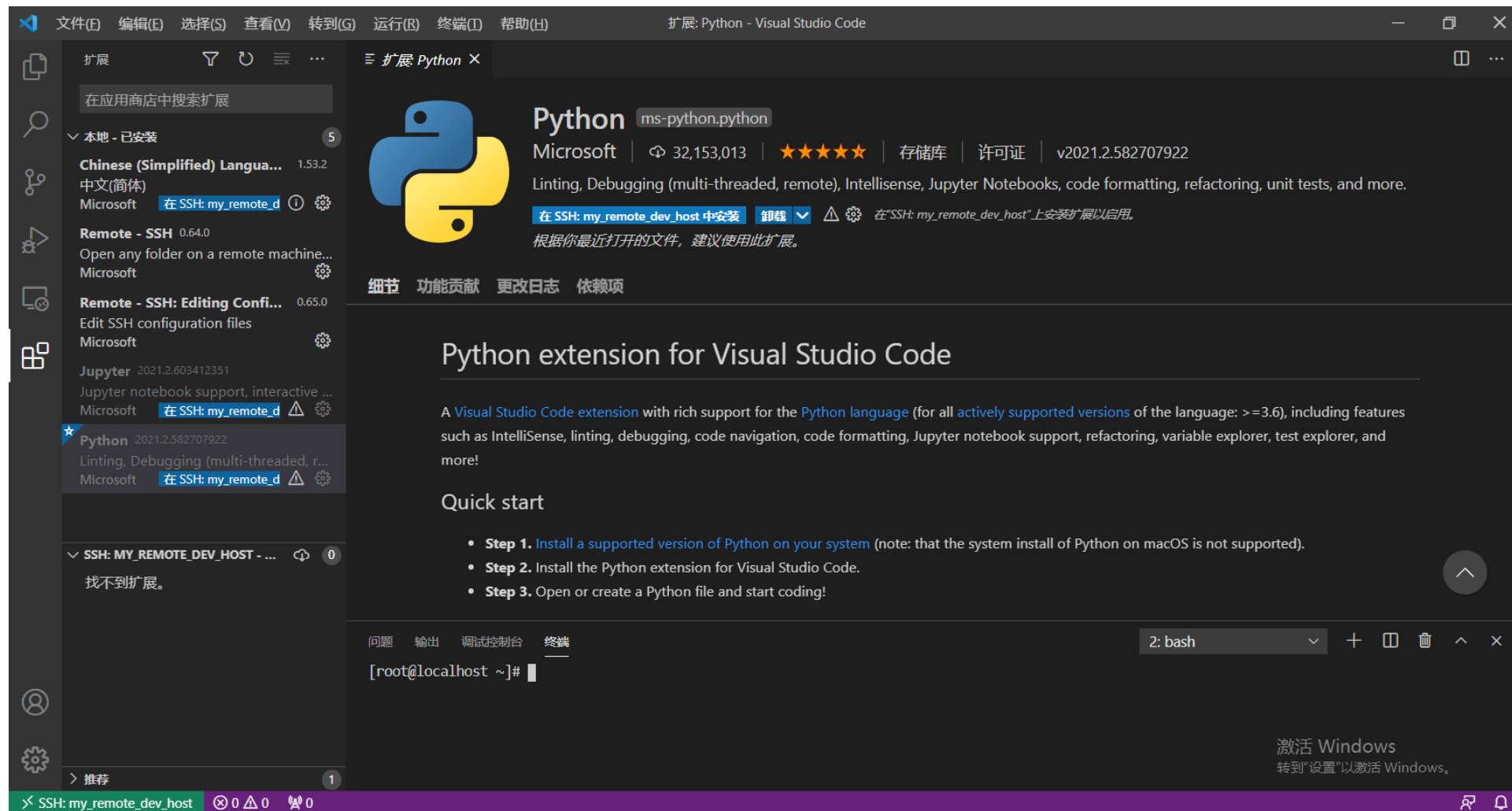
登录进远程系统的 SSH终端

```
[root@localhost python_demo]# ls
01_demo.py
[root@localhost python_demo]# uname -a
Linux localhost.localdomain 4.18.0-193.el8.x86_64 #1 SMP Fri May 8 10:59:10 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[root@localhost python_demo]#
```

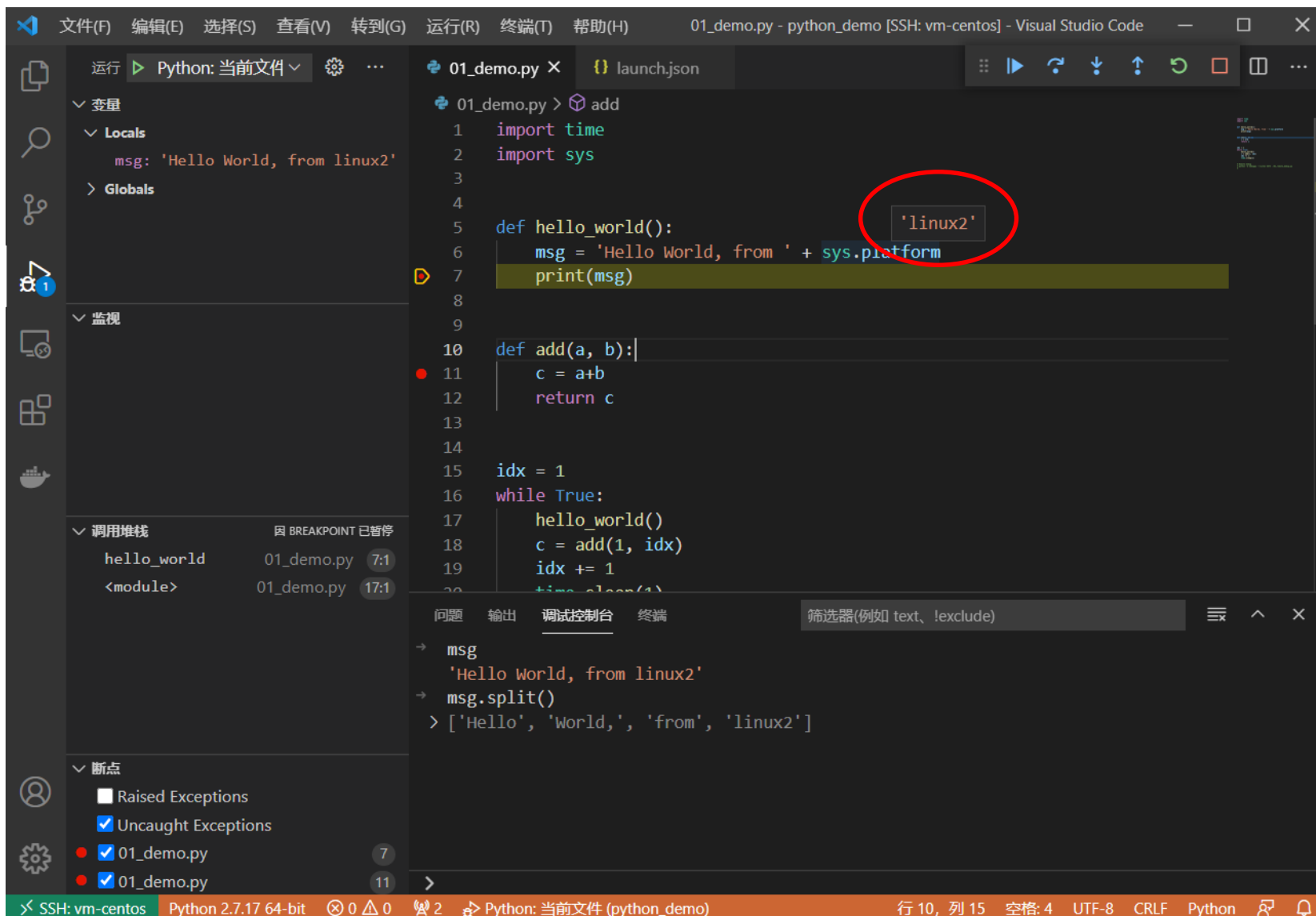
行 13, 列 1 空格: 4 UTF-8 CRLF Python

调试实战2： 远程代码调试 – 环境准备 – 安装插件到远程主机

直接点击扩展插件旁边的 “在 SSH:xxxx 中安装” ， 将相关插件推送到远程主机。



调试实战3： 远程代码调试 – Demo展示



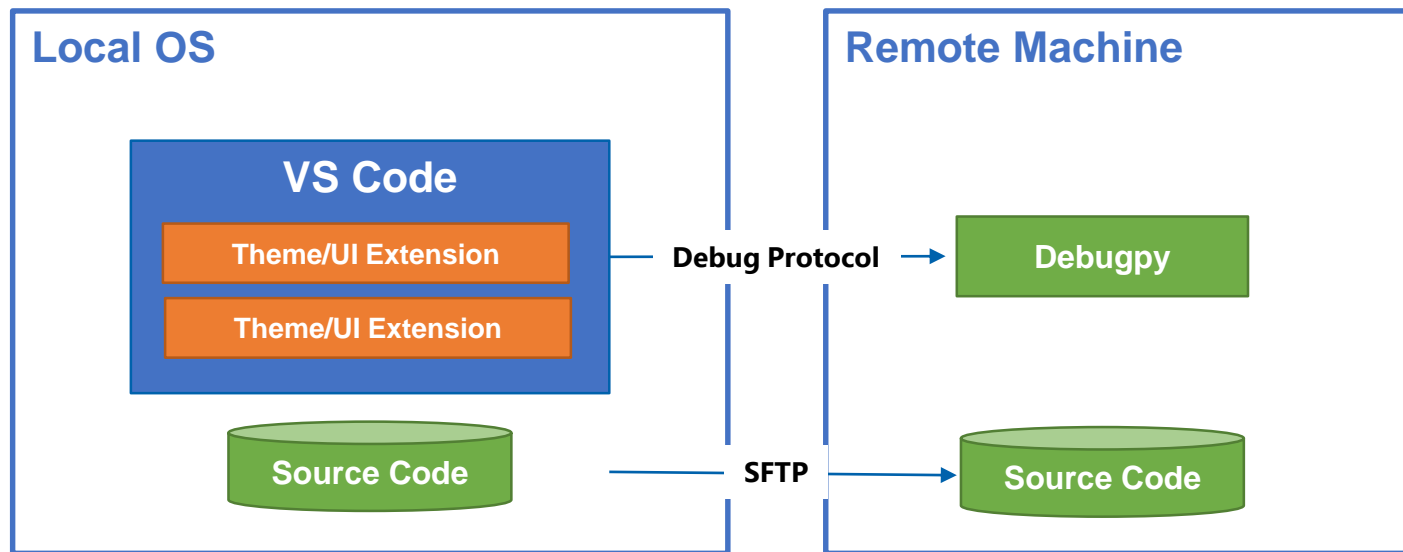
VSCode Python 调试配置
与同本地调试相同

Demo

远程代码调试

- 示例代码
- AS13000 命令行
 - Python文件 (`icfs -s`)
 - PID连接 (已运行服务)

调试实战3：本地+远程联调 – 原理介绍



本地

- VS Code
- Python 扩展插件
- SFTP 插件

远程主机

- Debugpy
- SSH Server

调试实战3：本地+远程联调 – 环境准备 – 远程服务器 Debugpy

远程服务器

安装 vscode Python 调试适配器

方案1：安装包

`pip install debugpy`

方案2：拷贝 debugpy 目录 (任选一种)

- `.../site-packages/debugpy`
- `.../site-packages/xxx.pth`
- `$PYTHONPATH / debugpy`

debugpy - a debugger for Python

Azure Pipelines succeeded license MIT pypi v1.2.1 python 2.7 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9

Coverage

OS	Coverage
Windows	coverage 83%
Linux	coverage 83%
Mac	coverage 82%

This debugger implements the Debug Adapter Protocol: [debugProtocol.json](#)

debugpy CLI Usage

Debugging a script file

To run a script file with debugging enabled, but without waiting for the client to attach (i.e. code starts executing immediately):

```
-m debugpy --listen localhost:5678 myfile.py
```

To wait until the client attaches before running your code, use the `--wait-for-client` switch.

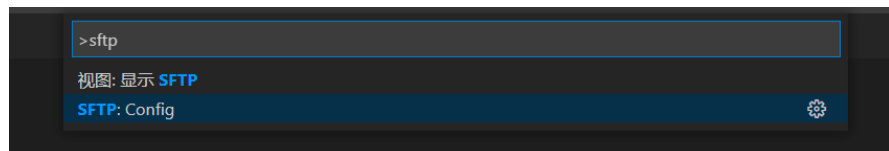
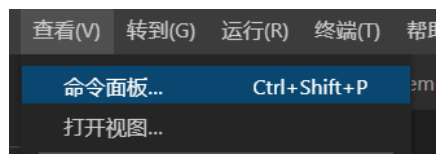
```
-m debugpy --listen localhost:5678 --wait-for-client myfile.py
```

- 微软官方模块
- 兼容多个python版本

<https://github.com/microsoft/debugpy/>

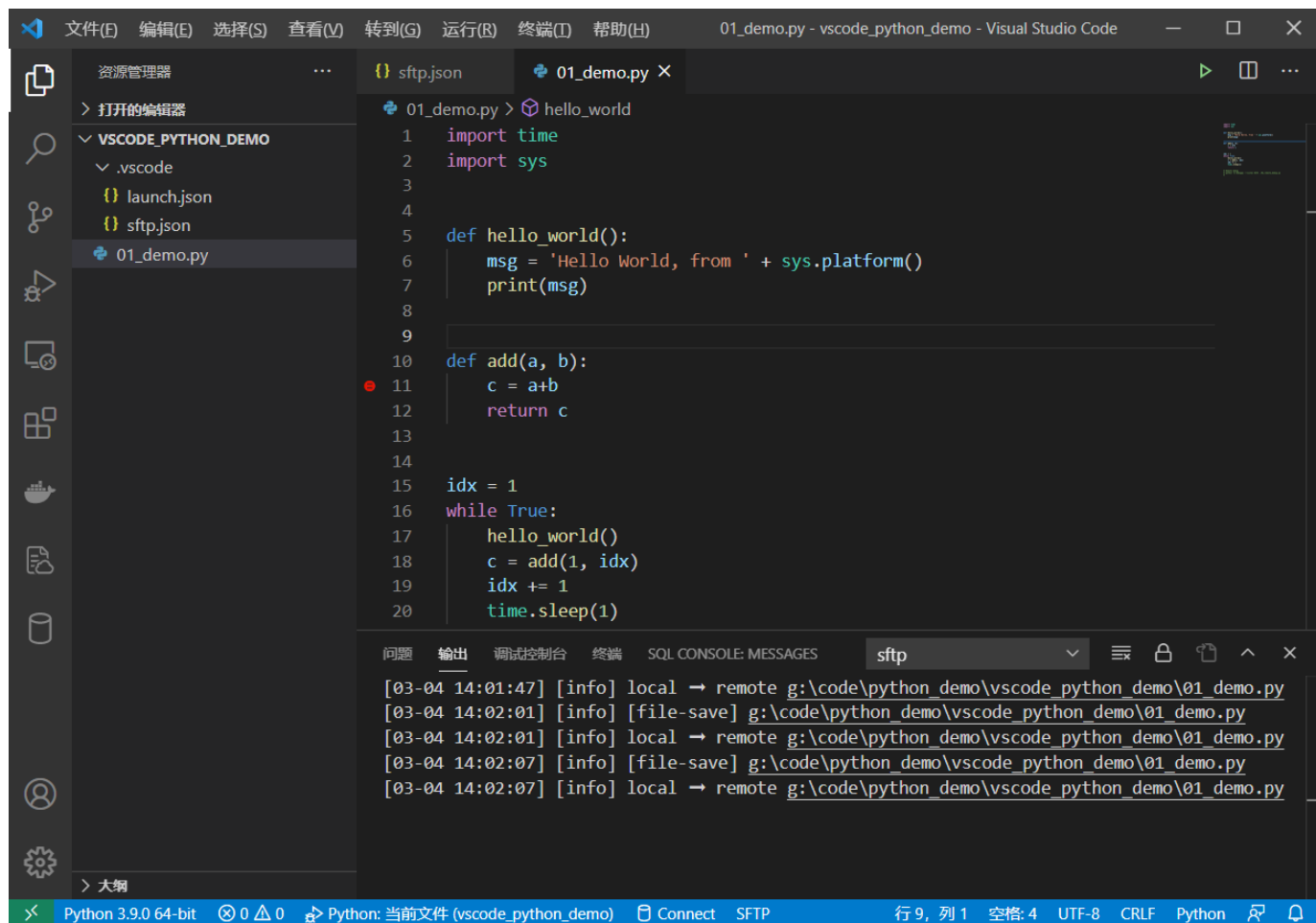
调试实战3：本地+远程联调 – 环境准备 – SFTP 同步插件

1. 创建 SFTP 配置文件



2. 编辑 SFTP 配置文件 (.vscode/sftp.json)

```
{
  "name": "My Server",
  "host": "192.168.5.2",
  "protocol": "sftp",
  "port": 22,
  "username": "root",
  "password": "123456",
  "remotePath": "/home/liangxh/python_demo/",
  "uploadOnSave": true
}
```



- 默认在文件保存时，可自动上传到远端。
也可右键点击文件或目录，手工上传。

调试实战3： 本地+远程联调 – Demo展示

```
hyper-v_centos - root@localhost:/home/liangxh/python_demo - Xshell 6 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root:***...
1 hyper-v_centos x +
[root@localhost python_demo]# python3 -m debugpy --listen 0.0.0.0:5678 --wait-for-client 01_demo.py

01_demo.py x
01_demo.py > hello_world
1 import time
2 import sys
3
4
5 def hello_world():
6     msg = 'Hello World, from linux'
7     print(msg)
8
9
10 def add(a, b):
11     c = a+b
12     return c
13
14
15 idx = 1
16 while True:
17     hello_world()
18     c = add(1, idx)
19     idx += 1
20     time.sleep(1)
21
22
23 # Remote debug
24 # python -m debugpy --listen 5678 ./01_demo.py
25 #
26

launch.json x
.vscode > launch.json > JSON 语言功能 > [ ] configurations
4 // 欲了解更多信息, 请访问: https://go.microsoft.com/fwlink
5 "version": "0.2.0",
6 "configurations": [
7     {
8         "name": "Python: 远程连接",
9         "type": "python",
10        "request": "attach",
11        "connect": {
12            "host": "192.168.5.2",
13            "port": 5678
14        },
15        "pathMappings": [
16            {
17                "localRoot": "${workspaceFolder}",
18                "remoteRoot": "/home/liangxh/python_demo/"
19            }
20        ]
21    }
22 ]
23 }
```

ssh://root@192.168.5.2:22

Python 3.9.0 64-bit 0 0 Python: 远程连接 (vscode_python_demo) Connect SFTP 行 8, 列 1 空格: 4 UTF-8 CRLF Python

添加配置...

Demo

本地+远程联调

一、背景介绍

二、VSCode Python 调试

三、离线开发环境部署

四、特殊场景调试

远程调试离线开发环境 搭建主要步骤

远程调试环境离线部署

Windows 本地主机

1. Git for Windows 安装 (ssh客户端)
2. VSCode 安装
3. VSCode 扩展插件

Linux 远程服务器

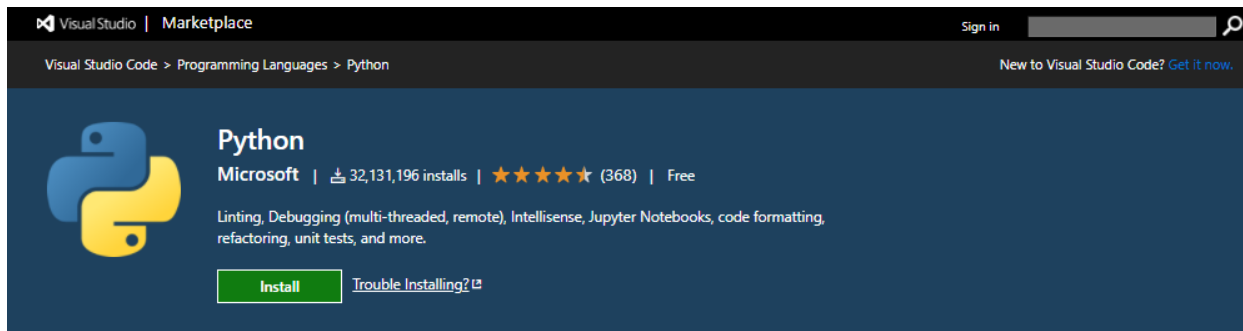
1. VSCode Server (手动部署)
2. VSCode 扩展插件 (通过VSCode推送)

VS Code 扩展插件

相关插件	本地调试	远程调试	本地+远程联调
 <p>Python <small>ms-python.python</small> Microsoft 30,405,486 ★★★★★ 存储库 许可证 v2021.1.502429796 Linting, Debugging (multi-threaded, remote), Intellisense, Jupyter Notebooks, code formatting, refactoring, unit tests, and more. [安装] [更新] [文档] [此扩展已全局启用]</p> <p>Python扩展基础插件</p>	●	●	●
 <p>Jupyter <small>ms-tools.jupyter</small> Microsoft 8,030,293 ★★★★★ 存储库 许可证 v2020.12.414227025 Jupyter notebook support, interactive programming and computing that supports Intellisense, debugging and more. [安装] [更新] [文档] [此扩展已全局启用]</p> <p>Jupyter Notebook 插件，新版Python依赖此插件</p>	●	●	●
 <p>Pylance <small>ms-python.vscode-pylance</small> <small>Active</small> Microsoft 870,269 ★★★★★ 存储库 许可证 v2021.1.2 A performant, feature-rich language server for Python in VS Code [安装] [更新] [文档] [此扩展已全局启用]</p> <p>Pylance 更智能的代码检查和语法提示插件</p>	● 可选	● 可选	● 可选
 <p>Remote - SSH <small>ms-vscode-remote.remote-ssh</small> <small>Active</small> Microsoft 2,622,805 ★★★★★ 存储库 许可证 v0.63.0 Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set. [安装] [更新] [文档] [此扩展已全局启用]</p> <p>SSH 远程连接插件</p>		●	
 <p>SFTP <small>liximomo.sftp</small> liximomo 777,565 ★★★★★ 存储库 许可证 v1.12.9 SFTP/FTP sync [安装] [更新] [文档] [此扩展已全局启用]</p> <p>SFTP/FTP 代码同步插件</p>			●

VS Code 扩展插件 – 下载与安装

<https://marketplace.visualstudio.com/VSCode>



[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported versions of the language: >=3.6), including features such as IntelliSense, linting, debugging, code navigation, code formatting, Jupyter notebook support, refactoring, variable explorer, test explorer, and more!

Quick start

- Step 1. Install a supported version of Python on your system (note: that the system install of Python on macOS is not supported).
- Step 2. Install the Python extension for Visual Studio Code.
- Step 3. Open or create a Python file and start coding!

Set up your environment

- Select your Python interpreter by clicking on the status bar
- Configure the debugger through the Debug Activity Bar

Categories

Programming Languages Linters Debuggers
Formatters Extension Packs Data Science
Machine Learning Notebooks

Tags

debuggers django ini jinja json keybindings
linters multi-root ready pip requirements
pip-requirements python requirements.txt toml
unittest yami

Resources

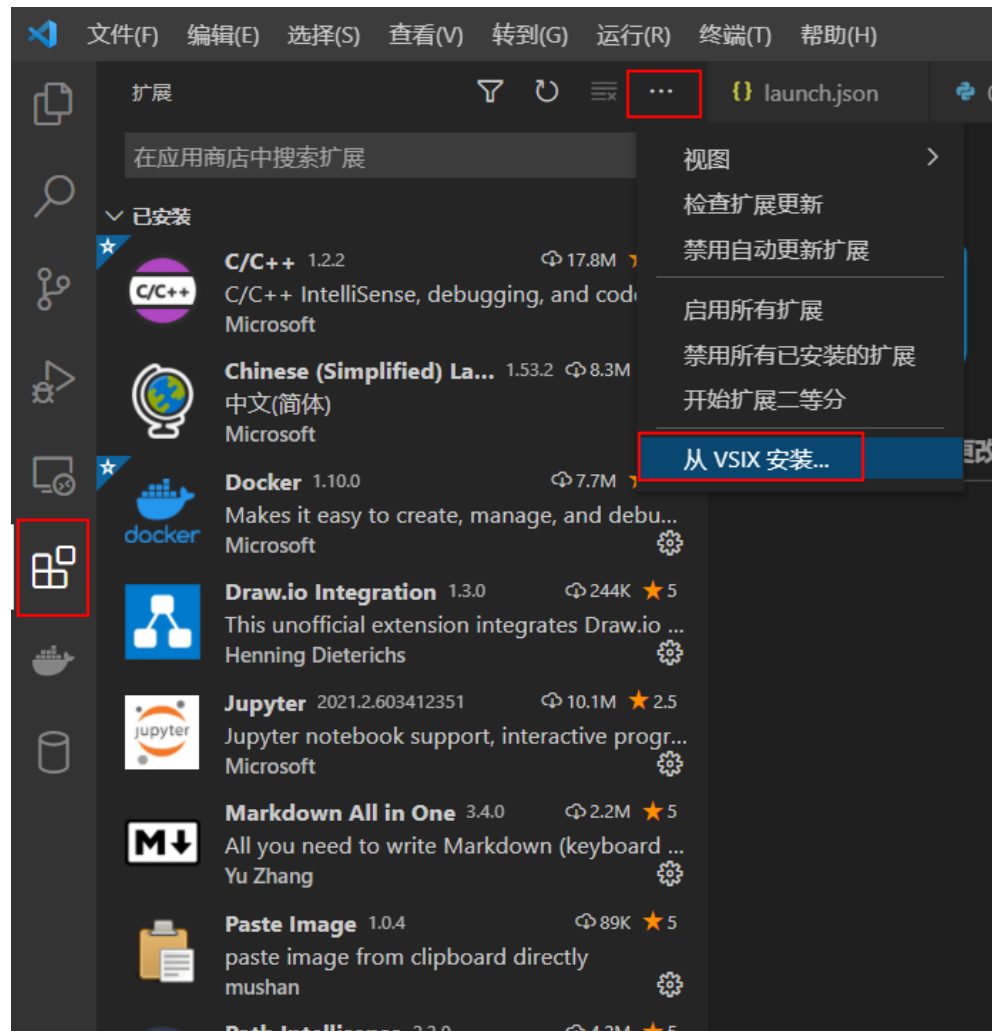
Issues
Repository
Homepage
License
Changelog
[Download Extension](#)

Project Details

Microsoft/vscode-python
Last Commit: 4 hours ago
17 Pull Requests
924 Open Issues

More Info

Version 2021.2.582707922
Released on 2016/1/19 下午11:03:11

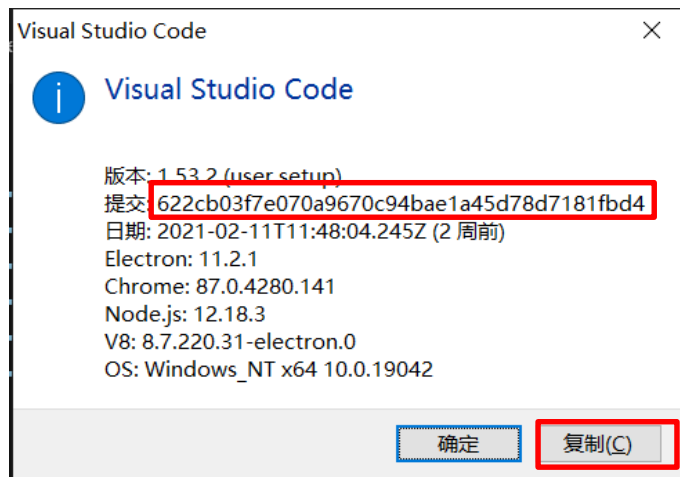


VS Code Server 离线下载

操作步骤:

1. 获取 commit_id (不同版本的VSCode 对应不同的 VSCode Server)
2. 下载 VSCode Server
地址: [https://update.code.visualstudio.com/commit:\\${commit_id}/server-linux-x64/stable](https://update.code.visualstudio.com/commit:${commit_id}/server-linux-x64/stable)
3. 上传 vscode-server-linux-x64.tar.gz 到远程服务器
4. 解压 vscode-server-linux-x64.tar.gz 到 ~/.vscode-server/bin/\${commit_id} , 不包含 vscode-server-linux-x64 目录
`tar zxvf /tmp/vscode-server-linux-x64.tar.gz -C ~/.vscode-server/bin/${commit_id} --strip 1`

commit_id 获取方式: VSCode / 帮助 / 关于



服务器解压后示例:

```
[root@node241 bin]# pwd
/root/.vscode-server/bin
[root@node241 bin]# ls
622cb03f7e070a9670c94bae1a45d78d7181fbd4
[root@node241 bin]# ls -l 622cb03f7e070a9670c94bae1a45d78d7181fbd4/
total 47572
drwxr-xr-x  3 root root      33 Feb 11 19:49 bin
drwxr-xr-x 34 root root    4096 Feb 11 19:49 extensions
-rw-r--r--  1 root root   13380 Feb 11 19:48 LICENSE
-rwxr-xr-x  1 root root 48646656 Feb 11 19:49 node
drwxr-xr-x 66 root root    4096 Feb 11 19:49 node_modules
drwxr-xr-x  3 root root     151 Feb 11 19:49 out
-rw-r--r--  1 root root    1077 Feb 11 19:48 package.json
-rw-r--r--  1 root root   29960 Feb 11 19:48 product.json
-rwxr-xr-x  1 root root     222 Feb 11 19:48 server.sh
-rw-r--r--  1 root root        0 Mar  3 13:41 vscode-remote-lock.root.622cb03f7e070a9670c94bae1a45d78d7181fbd4
```

参考链接:

<https://stackoverflow.com/questions/56671520/how-can-i-install-vscode-server-in-linux-offline>

一、背景介绍

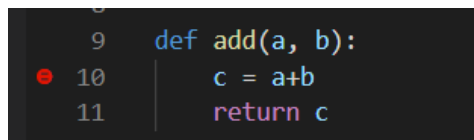
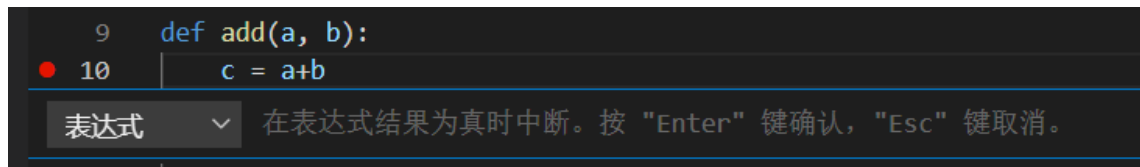
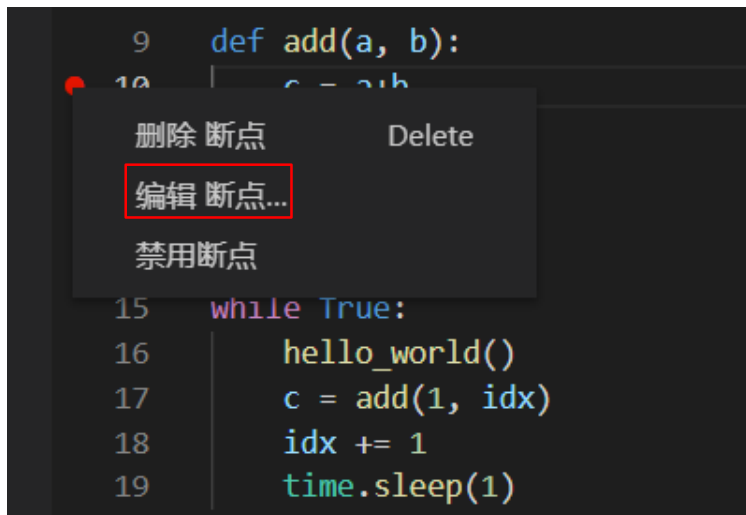
二、VSCode Python 调试

三、离线开发环境部署

四、特殊场景调试

条件断点

应用场景：在一个循环中，处理到某一条数据、某个文件时触发。



- 右键点击断点图标 (小红点)
- 编辑断点
- 满足条件时触发 (可以是任意返回 bool 的表达式)

调试端口被防火墙拦截 – 利用 SSH 隧道端口转发

添加步骤 (以 xshell 为例) :

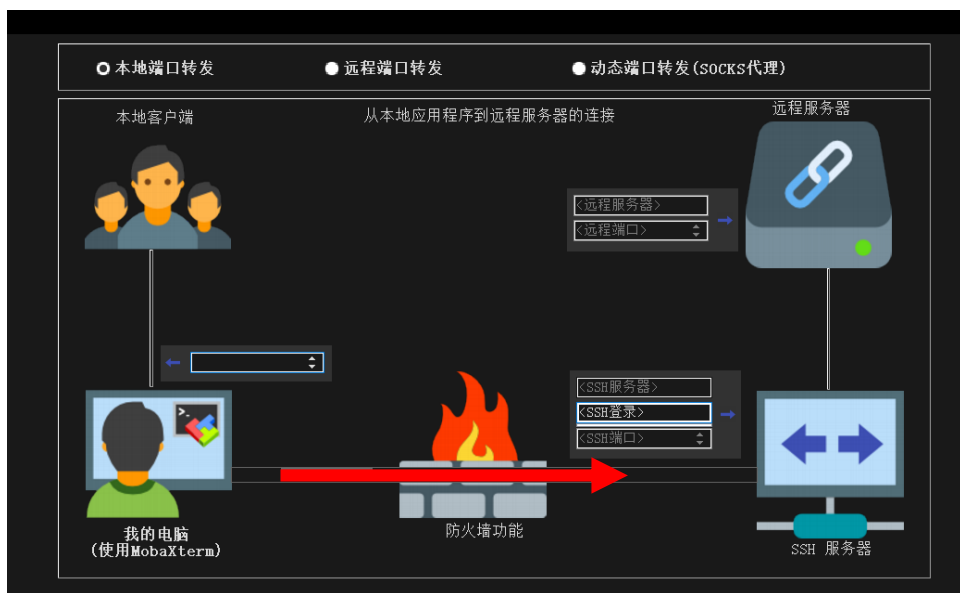
1. Xshell 通过ssh 登录远程主机 (默认22端口)
2. Xshell 添加 SSH隧道 端口转发规则 (**xshell 连接属性**, 或者 **隧道窗格**)

类型: 本地拨出

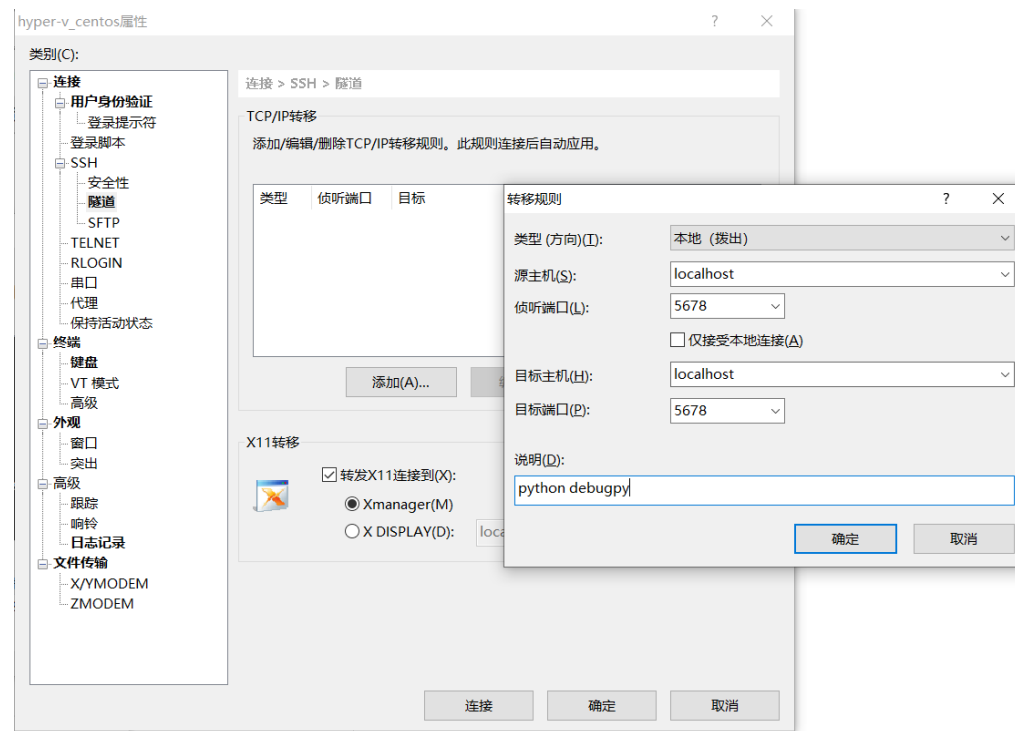
指定源主机 (本地开发环境) 监听IP (网卡) 和端口

指定目标主机IP (不一定是远程服务器自身, 跳转机到别处也可以) 和端口

3. 访问本地端口即相当于访问远程主机端口



原理示意 (来源: mobaxterm)



GEvent (OpenStack)

关键步骤:

1. 安装或拷贝 debugpy 模块
2. 在原有启动服务中增加环境变量
 1. PYTHONPATH, 用于指定自定义模块搜索路径 (可选)
 2. GEVENT_SUPPORT=True, 支持Gevent调试
3. 修改 ExecStart启动参数, 以 debugpy 启动原有脚本

```
xxx.service
```

```
[Service]
```

```
# 自定义 Python模块搜索路径, debugpy 所在目录 (如果已安装 debugpy包, 可省略)
```

```
Environment=PYTHONPATH=/opt/my_python_lib/
```

```
# GEVENT_SUPPORT 开关, 多线程相关
```

```
Environment=GEVENT_SUPPORT=True
```

```
ExecStart=/usr/bin/python2 -m debugpy --listen 0.0.0.0 1234 /usr/bin/cinder-volume -config-file ...
```

开发阶段：

使用虚拟机环境进行调试，虚拟机环境和 Docker 镜像保持一致。

- 操作系统一致
- Python版本一致
- requirements.txt 一致

特殊情况：

Docker 内启动 debugpy，开放远程调试端口。

谢谢大家!



inspur 浪潮