

Zhang-Lab 生信小课堂 第二一期

Applied Bioinformatics Club (ABC)

和趣求真  秉实生信

(张建伟生物信息学课题组 <https://zhang.hzau.edu.cn>)

在VSCode中远程编写和调试集群Linux程序

2025.04.11 二综C102 15:00 欢迎大家交流学习!

主讲人: 刘健

2025/04/11

目录

01 使用VSCode远程连接集群的优点

02 设置流程

03 配置c++运行环境

1.使用VSCode远程连接集群的优点

VSCode: 轻量级代码编辑器, 根据需要下载对应的插件

直接在windows界面上操作linux文件, vscode能轻松远程下载插件, 更加便捷。

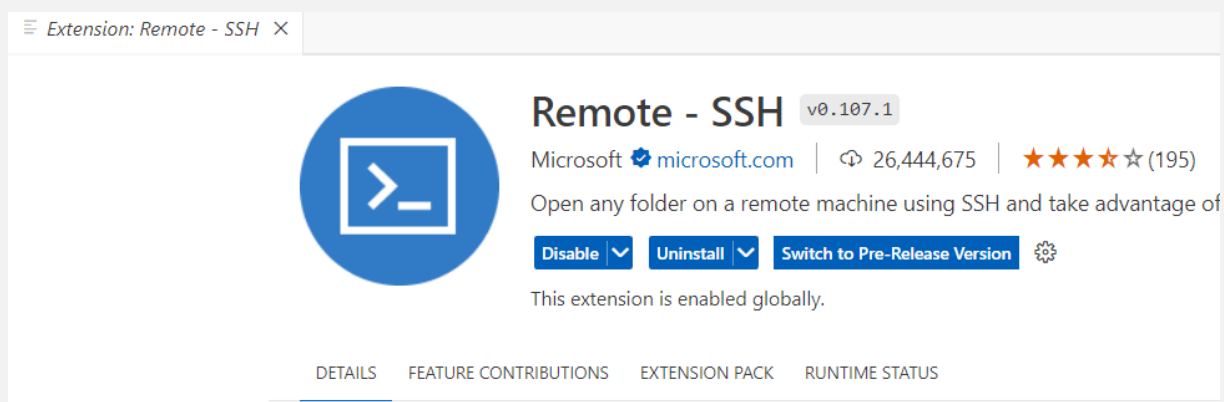
2. 设置流程

查看集群上Glibc(Linux系统的基础api)的版本：ldd --version

```
[jianliu@login02 ~]$ ldd --version
ldd (GNU libc) 2.17
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
[jianliu@login02 ~]$
```

根据集群上Glibc的版本选择VSCode的版本和Remote ssh插件的版本：

VSCode：1.58.2



2. 设置流程

The screenshot displays the Visual Studio Code interface with the Remote - SSH extension installed. The interface is annotated with red numbers and text to guide the user through the setup process.

1. The Extensions view is open, showing the list of installed extensions. The Remote - SSH extension is highlighted.

2. The search bar in the Extensions view is used to search for the Remote - SSH extension.

2. 搜索 Remote - SSH

3. 选择版本

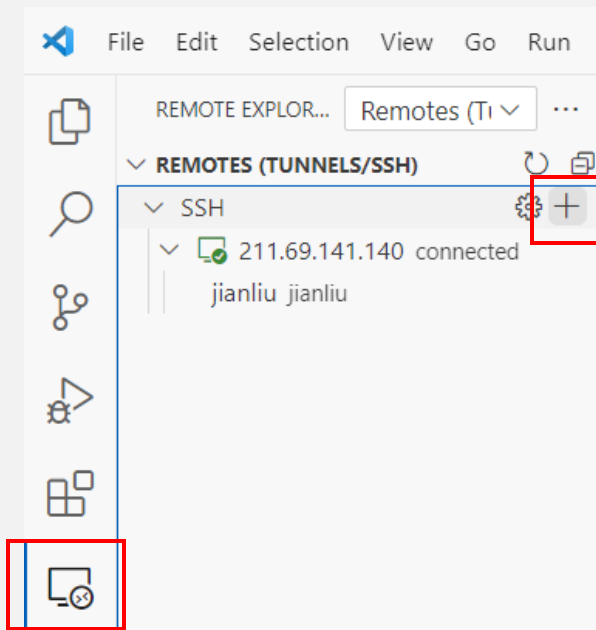
The Remote - SSH extension details page is shown, including the version number (v0.107.1) and the description: "Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set." The extension is currently disabled, and the user can choose to enable it, uninstall it, or switch to a pre-release version.

The terminal output shows the SSH connection process:

```
[13:19:41.123] Forwarding server listening on port 50691
[13:19:41.123] Waiting for ssh tunnel to be ready
[13:19:41.124] [Forwarding server port 50691] Got connection 0
[13:19:41.125] Tunneled port 43529 to local port 50691
[13:19:41.125] Resolved "ssh-remote+211.69.141.140" to "port 50691"
[13:19:41.133] -----
```

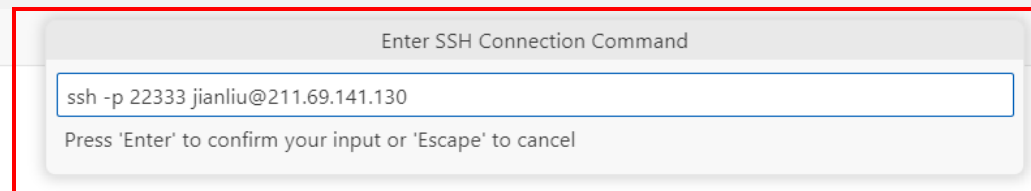
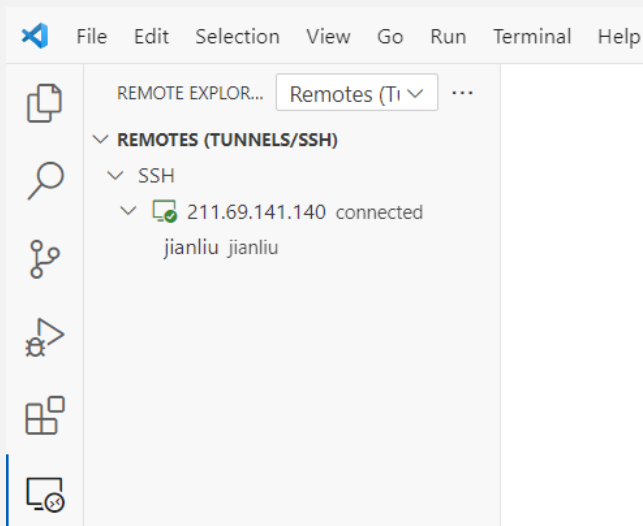
2. 设置流程

4.Remote explorer



5.New remote

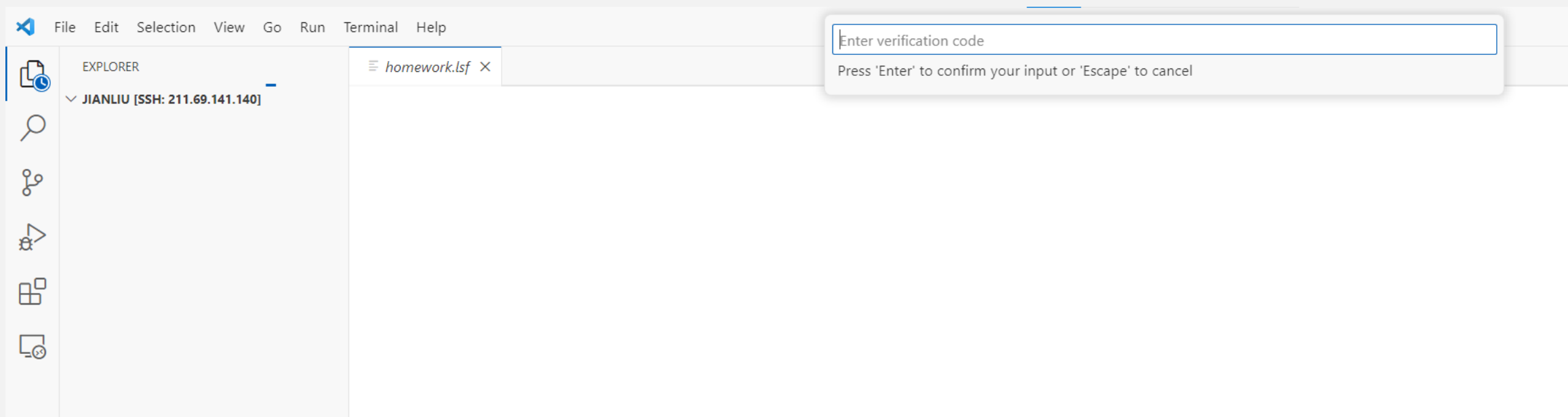
Remote -SSH插件默认会自动在集群安装VSCode server, 连接成功!



6.选择主机类型: linux, 输入端口号和主机地址

2. 设置流程

登陆界面：



2. 设置流程

集群文件列表，和
windows操作方式
一样，点击打开，
右键删除或新建

The screenshot displays the Visual Studio Code (VS Code) interface connected to a remote server via SSH. The interface is divided into three main panels:

- EXPLORER (Left Panel):** Shows a file tree for the user 'JIANLIU [SSH: 211.69.141.140]'. The tree includes directories like `__pycache__`, `.cache`, `.conda`, `.local`, `.lsbatch`, `.mozilla`, `.ssh`, `.vscode-server`, `ai_studio`, `GitPublish`, `lico_share_dir`, `Python-3.12.4`, `.bash_history`, `.bash_logout`, `.bash_profile`, `.bashrc`, `.google_authenticator`, `.kshrc`, `.lessht`, `.python_history`, `.viminfo`, `.Xauthority`, and various C++ source files like `aligningUnitAndItsLocation.cpp`, `allReadsAndItsID.cpp`, `f.h`, `function_in_assembly.cpp`, `function_in_Fastq_file.cpp`, and `globals.h`. The file `homework.lsf` is selected and highlighted.
- EDITOR (Center Panel):** Displays the content of the selected file `homework.lsf`. The file contains a list of job submission parameters for a batch system (likely PBS or similar):

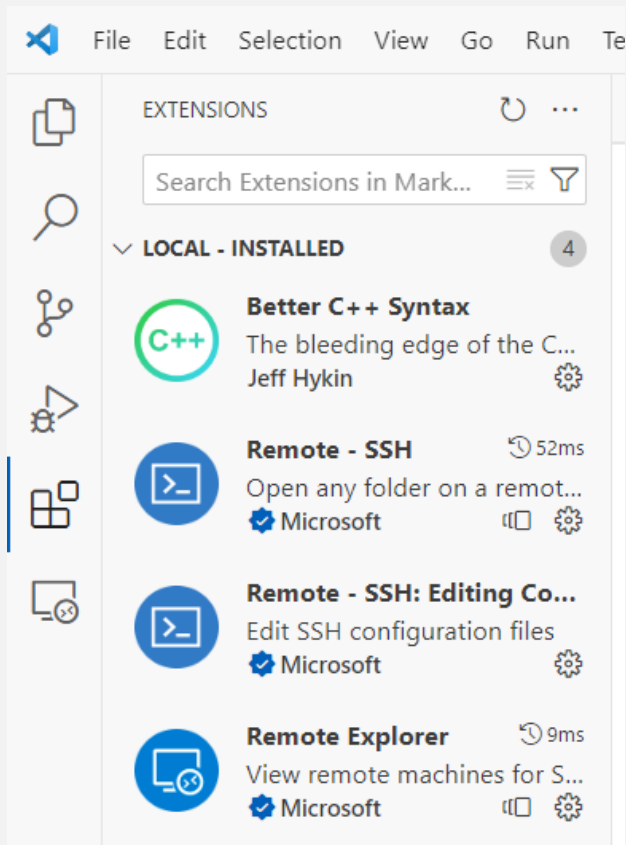
```
1 #BSUB -J blast
2 #BSUB -n 3
3 #BSUB -R span[hosts=1]
4 #BSUB -o %J.out
5 #BSUB -e %J.err
6 #BSUB -q normal
7
8 minimap2 -x ava-pb -c -t 8 SRR32655838.fasta SRR32655838.fasta > overlaps.paf
```
- TERMINAL (Bottom Panel):** Shows a command prompt session. The command `ls` has been entered, and the output lists the files and directories in the current directory, including `ai_studio`, `aligningUnitAndItsLocation.cpp`, `f.h`, `function_in_assembly.cpp`, `GitPublish`, `homework.lsf`, `main.cpp`, `Python-3.12.4.tgz`, `SRR13280199.fastq`, `SRR32655838.fasta`, and `vscode-server.tar.gz`.

Red annotations highlight key features:

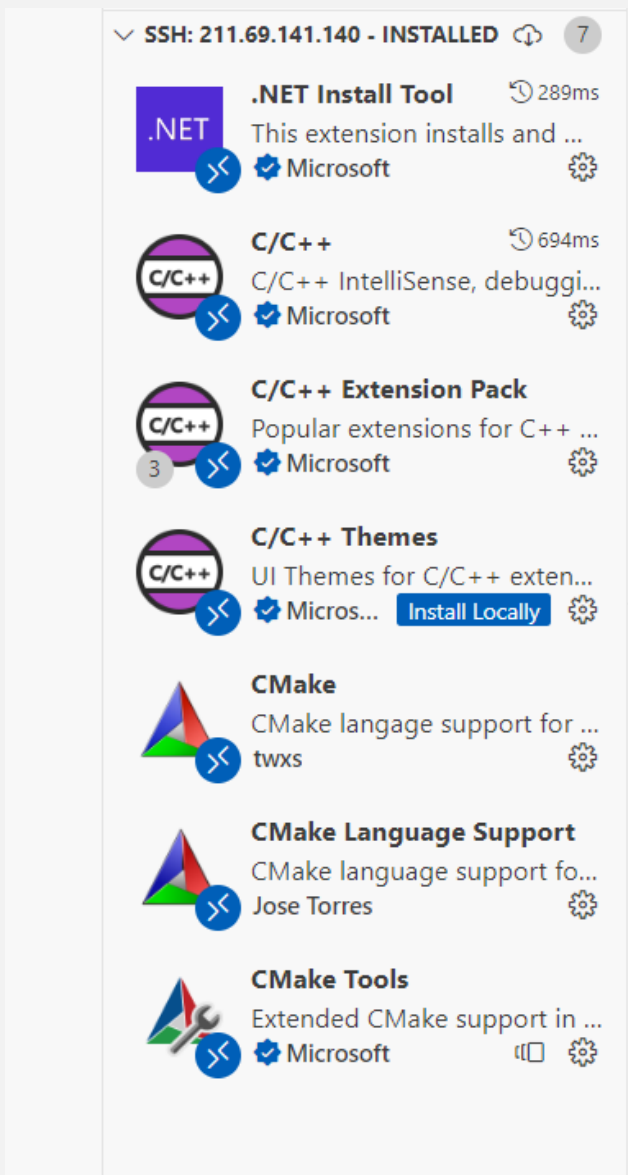
- A red box around the `.vscode-server` directory in the Explorer panel, with an arrow pointing to the text: "集群文件列表，和 windows操作方式 一样，点击打开， 右键删除或新建".
- A red box around the `homework.lsf` file in the Explorer panel, with an arrow pointing to the text: "可以编辑作业脚本".
- A red box around the `TERMINAL` tab in the bottom panel, with an arrow pointing to the text: "命令行终端，输入命令回车运行".

3.配置c++运行环境： 下载插件

本地：



集群上：



3. 配置c++运行环境：配置文件



名称不能更改

VSCode官方网址: <https://code.visualstudio.com/docs/debugtest/tasks>

tasks.json:

```
.vscode > {} tasks.json > [ ] tasks
1  {
2    "version": "2.0.0",
3    "tasks":
4      //可以为不同场景定义多个任务
5    [
6      {
7        "type": "cppbuild",
8        "label": "genomeAssemblyAlgorithm",
9        "command": "GCC", //使用GCC编译器
10       "args":
11         [
12           "-fdiagnostics-color=always",
13           "-g", // 生成调试信息
14           "${file}",
15           "-o", // 指定输出文件路径
16           "${fileDirname}/${fileBasenameNoExtension}" // 输出可执行文件路径
17         ],
18       "options":
19         {
20           "cwd": "${fileDirname}"
21         },
22       "problemMatcher":
23         [
24           "$gcc"
25         ],
26       "group":
27         {
28           "kind": "build",
29           "isDefault": true // 设为默认任务, 按 Ctrl+Shift+B 直接运行
30         },
31       "detail": "compiler: /bin/gcc"
32     }
33   ]
34 }
35
```

launch.json:

```
.vscode > {} launch.json > ...
1  {
2    "version": "0.2.0",
3    "configurations":
4    [
5      {
6        "name": "Remote C++ Debug",
7        "type": "cppdbg", // 调试器类型
8        "request": "launch", // 启动调试
9        "program": "${fileDirname}/${fileBasenameNoExtension}", // 可执行文件路径
10       "args": [], // 程序参数 (可选)
11       "stopAtEntry": false, // 是否在入口暂停
12       "cwd": "${workspaceFolder}", // 工作目录
13       "environment": [],
14       "externalConsole": false, // 使用 VSCode 内置终端
15       "MIMode": "gdb", // 调试器模式 (gdb 或 lldb)
16       "miDebuggerPath": "/usr/bin/gdb", // 远程机器的 gdb 路径
17       "setupCommands":
18         [
19           {
20             "description": "为 gdb 启用整齐打印",
21             "text": "-enable-pretty-printing",
22             "ignoreFailures": true
23           }
24         ],
25       "preLaunchTask": "genomeAssemblyAlgorithm" // 必须与 tasks.json 中的 label 一致, 确保调试前执行编译任务
26     }
27   ]
28 }
```

感谢观看！