

rpi-os 实验环境使用手册

欢迎使用 rpi-os 云实验环境。本学期的所有服务器资源由华为赞助提供。我们一共有三台服务器，地址为 `huawei0{1,2,3}.soaringlab.top`。目前只启用了一台服务器。如果后续服务器需求增加，我们可能会启用更多的服务器。

登录

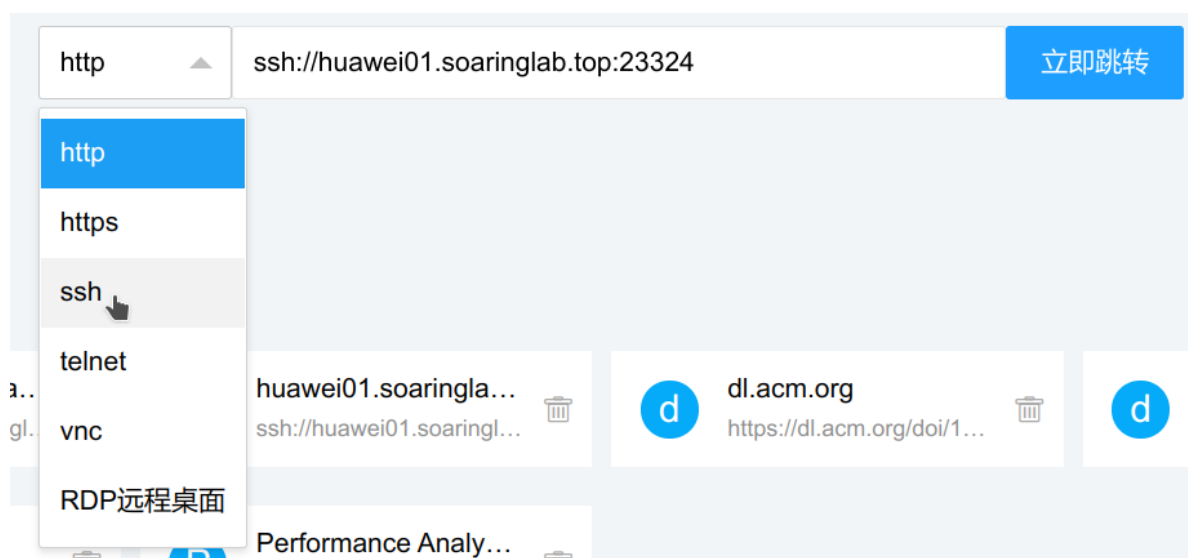
我们会给每个人分配一个容器作为实验环境。所有容器都在同一台服务器上，IP 地址是相同的，因此我们会用不同的端口号来区分不同的容器。

如果你在使用校园网，可以用 `ssh` 登录你的容器：

```
$ ssh -p 23324 root@huawei01.soaringlab.top
```

`-p 23324` 表示登录到端口号为 23324 的容器。所有容器都以 root 用户身份登录，因此登录之后你可干任何事情。

如果你无法连接到校园网，可以考虑使用 VPN 或者 webvpn.fudan.edu.cn。WebVPN 中，可以在地址栏里面输入 `ssh://huawei01.soaringlab.top:[你的端口号]`，也可以在左侧手动选择 ssh 协议。然后使用密钥登录 root 帐号。



基本配置

使用 ssh 登录后，默认是在 `~` 目录下（`/root` 目录）。使用 `ls` 命令可以看到我们预先准备好的文件和目录：

```
$ ls
README.md fudan-net-auth.sh share/ data/
```

`fudan-net-auth.sh` 用于网络认证。执行后，会要求你输入帐号（学号）和密码。登录成功后就可以访问互联网。

`share` 目录只读。有时候我们会向所有同学分发新的文件，就会放到这个目录下。

`data` 目录挂载在 4.4T 的硬盘上面。一般情况下你应该把你的文件和数据都放在这个目录下面。

同步仓库

我们建议你把仓库放在 `/root/data` 目录下。

同步仓库有多种方法：

- 使用 `git clone/pull/fetch`。
 - 这种方法比较通用。如果你的个人仓库是公开的，那么可以直接 `clone`。否则，你可能需要在容器下自己配置一个新的密钥。
- 使用 `rsync`。
 - 直接用 `rsync` 传输整个仓库：`rsync -avzP -e "ssh -p [你的端口号]" repo root@huawei01.soaringlab.top:/root/data`。
 - 只传输仓库的 `.git` 文件夹，这样可以减少传输量：`rsync -avzP -e "ssh -p [你的端口号]" repo/.git`
`root@huawei01.soaringlab.top:/root/data/repo`。然后在容器里面的 `repo` 目录下使用 `git reset --hard HEAD` 命令恢复所有文件。
 - 把整个仓库打包成压缩包上传：懂的都懂。

以上是往服务器上传文件的方法。从服务器下载文件也是类似的。

警告：我们强烈建议不要使用 Windows 环境（WSL 除外）操作实验仓库。因为 Windows 环境下存在不支持软链接、保留文件名等问题，很有可能在编译的时候出现奇怪的问题。

编辑、编译、调试

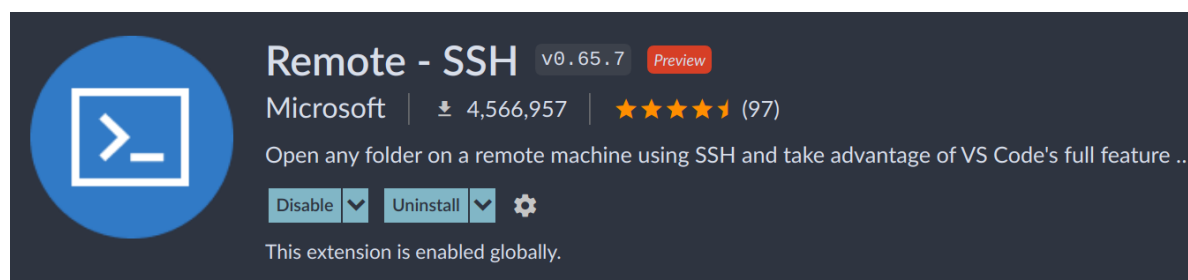
“Edit” - “Compile” - “Debug” 是编程开发的经典步骤。接下来将一一介绍。

编辑

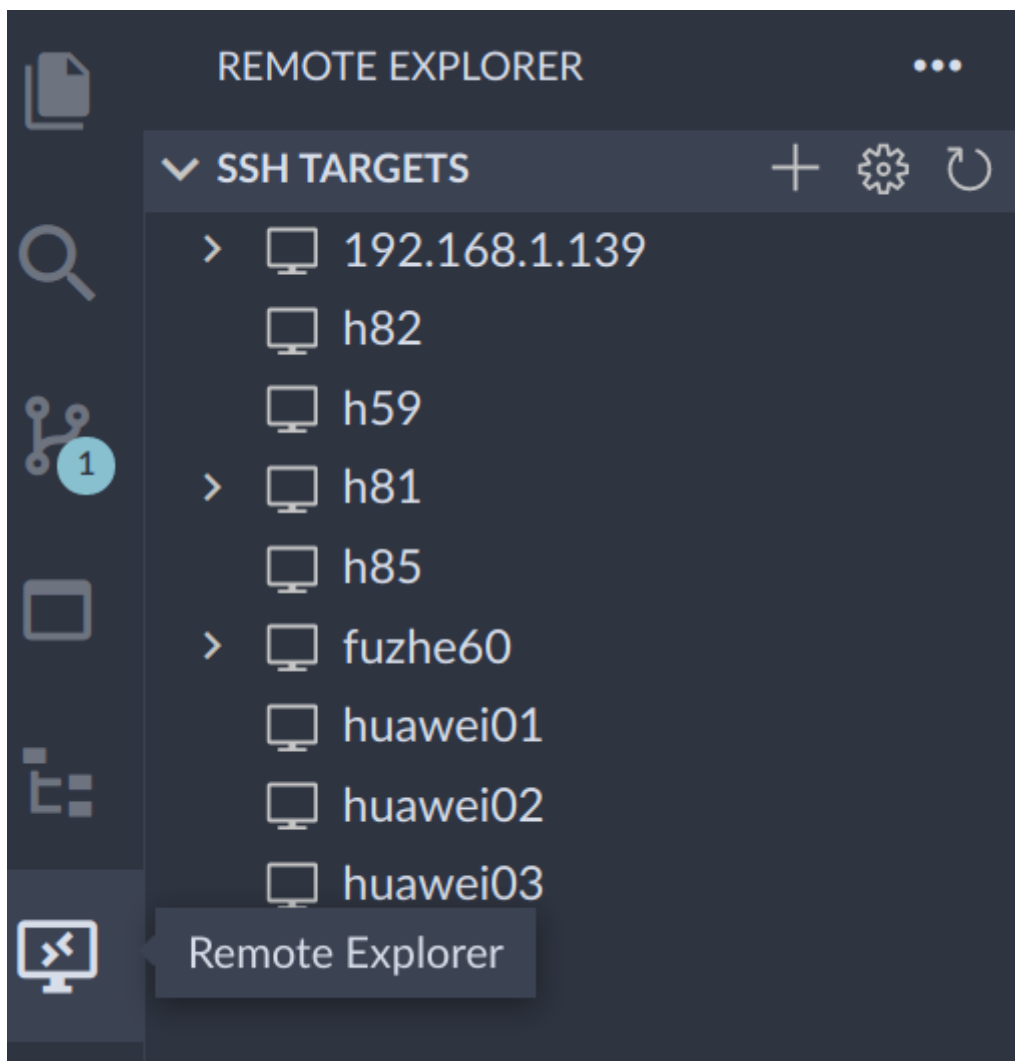
你可以选择直接在服务器上编辑代码。实验环境预装了 `nano` 和 `vim` 编辑器。如果你有需要，你可以使用 `apt` 安装其它的编译器，如 `emacs`。

此外，实验环境预装了 `tmux`，方便各位分屏工作。

你也可以选择在本地编辑完代码后再同步到服务器上。同步的方法在前面已经介绍。这里我们推荐你使用 Visual Studio Code 远程连接到服务器。只需要安装下图所示的插件即可：



安装后，在左侧面板可以找到“Remote Explorer”：



连接后就可以像编辑本地文件一样编辑服务器上的文件了。

编译

服务器上编译和在本地编译没有区别。请参考实验仓库里面的介绍。

注意：因为服务器本身就是 AArch64 架构，因此在服务器上编译的时候用的不是交叉编译器。一般情况下你不需要担心这个事情，因为我们的 CMake 脚本会自动检测编译环境。

调试

服务器上调试和在本地调试没有区别。请参考实验仓库里面的介绍。

调试时需要启动两个终端。你可以：

- 使用 `tmux` 分屏。
- 开启两个 `ssh` 会话。

注意：因为服务器本身就是 AArch64 架构，因此服务器上不会使用 `gdb-multiarch`，而是使用原始的 `gdb`。我们的 CMake 脚本和 `make debug` 命令会自动处理这件事情。

后记

实验环境本质上就是个 Ubuntu 20.04 的容器。本手册主要面向没有远程连接 Linux 环境的经验的同学。相信熟悉 Linux 的同学一定能找到适合自己的使用方法。

警告：服务器属于公共资源，仅供课程学习、交流使用。请勿滥用服务器资源，确保各位能有良好的使用体验。