


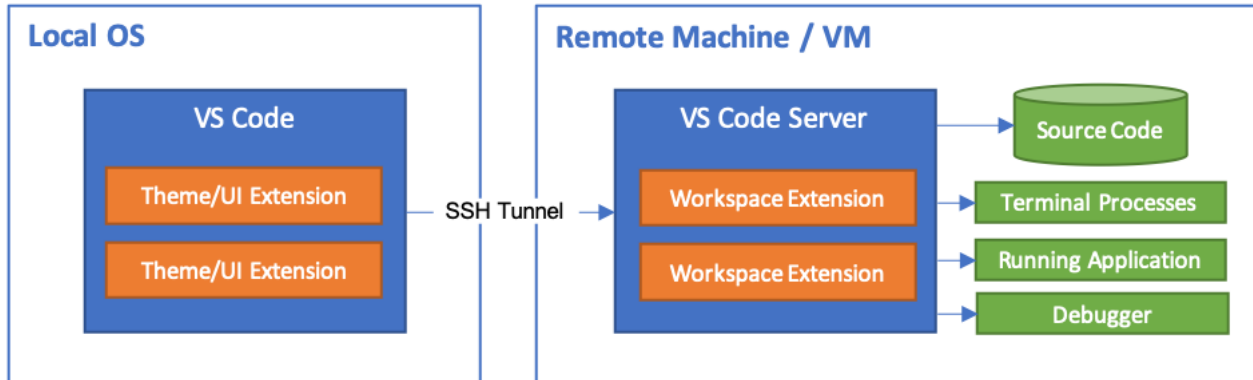
TOPICS SSH协议 ▼

使用SSH进行远程开发

 (<https://github.com/Microsoft/vscode-docs/blob/master/docs/remote/ssh.md>)

在Visual Studio代码远程- SSH扩展允许你打开任何远程计算机，虚拟机或容器上的远程文件夹与正在运行的SSH服务器，并充分利用VS代码的功能集。连接到服务器后，您可以与远程文件系统上任何位置的文件和文件夹进行交互。

由于扩展名直接在远程计算机上运行命令和其他扩展名，因此无需在本地计算机上使用源代码即可获得这些好处。



这使VS Code可以提供本地质量的开发体验 - 包括完整的IntelliSense（完成功能），代码导航和调试- 不管代码托管在何处。

入门

系统要求

本地： 还必须安装 受支持的OpenSSH兼容SSH客户端 (/docs/remote/troubleshooting#_installing-a-supported-ssh-client)。

远程SSH主机： 位于以下位置的正在运行的SSH服务器 (/docs/remote/troubleshooting#_installing-a-supported-ssh-server)：

- x86_64 Debian 8以上版本， Ubuntu 16.04以上版本， CentOS / RHEL 7以上版本。
- ARMv7l (AArch32) Raspbian Stretch / 9 + (32位) 。
- ARMv8l (AArch64) Ubuntu 18.04+ (64位) 。
- 使用官方OpenSSH Server的 (https://docs.microsoft.com/windows-server/administration/openssh/openssh_install_firstuse) Windows 10 / Server 2016/2019 (1803+) 。
- 启用 (<https://support.apple.com/guide/mac-help/allow-a-remote-computer-to-access-your-mac-mchlp1066/mac>) 了远程登录的 (<https://support.apple.com/guide/mac-help/allow-a-remote-computer-to-access-your-mac-mchlp1066/mac>) macOS 10.14+ (Mojave) SSH主机 。

glibc 如果具有必要的先决条件，则其他基于x86_64， ARMv7l (AArch32) 和ARMv8l (AArch64) 的Linux发行版应该可以使用。有关启动和运行社区支持的发行版 (/docs/remote/linux)的信息先决条件和技巧，请参阅《使用Linux (/docs/remote/linux)进行远程开发》 — (/docs/remote/linux)文。

虽然提供了ARMv7l (AArch32) 和ARMv8l (AArch64) 支持，但是由于扩展中使用了x86本机代码，因此这些设备上安装的某些扩展可能无法正常工作。

安装

要开始使用，您需要：

1. 如果尚不存在与OpenSSH兼容的SSH客户端，请 (/docs/remote/troubleshooting#_installing-a-supported-ssh-client)安装该客户端 (/docs/remote/troubleshooting#_installing-a-supported-ssh-client)。
2. 安装Visual Studio Code (<https://code.visualstudio.com/>)或Visual Studio Code Insiders (<https://code.visualstudio.com/insiders/>)。
3. 安装远程开发扩展包 (<https://aka.ms/vscode-remote/download/extension>)。

SSH主机设置

1. 如果您没有设置SSH主机，请按照Linux (/docs/remote/troubleshooting#_installing-a-supported-ssh-server)， Windows 10 / Server (1803+) (https://docs.microsoft.com/windows-server/administration/openssh/openssh_install_firstuse)或macOS (<https://support.apple.com/guide/mac-help/allow-a-remote-computer-to-access-your-mac-mchlp1066/mac>) SSH主机的说明进行操作，或者在Azure上 (<https://docs.microsoft.com/azure/virtual-machines/linux/quick-create-portal?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json>)创建VM (<https://docs.microsoft.com/azure/virtual-machines/linux/quick-create-portal?toc=%2fazure%2fvirtual-machines%2flinux%2ftoc.json>)。
2. [可选]虽然支持基于密码的身份验证，但建议为主机设置基于密钥的身份验证。有关详细信息，请参见“技巧和窍门” (/docs/remote/troubleshooting#_configuring-key-based-authentication)文章。

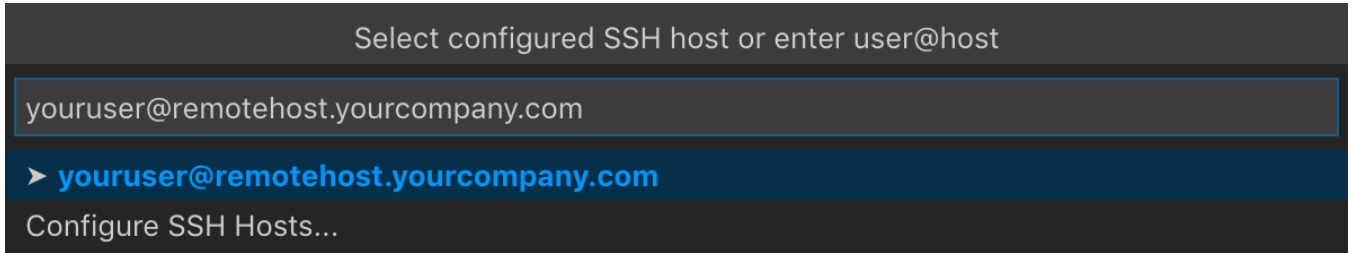
连接到远程主机

要首次连接到远程主机，请按照下列步骤操作：

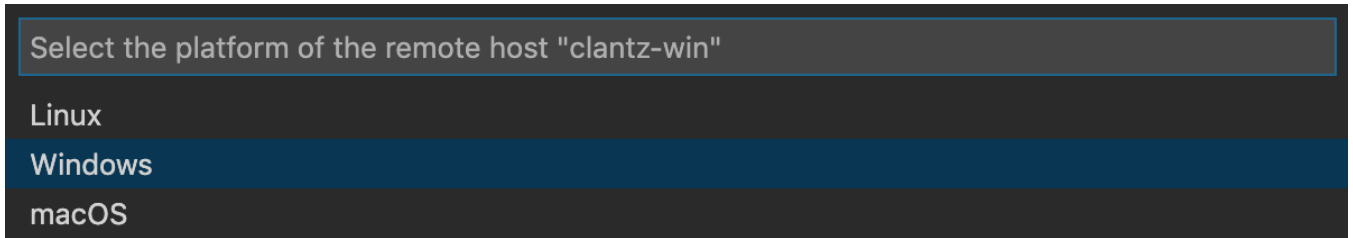
1. 通过在终端/ PowerShell窗口中运行以下命令（ user@hostname 如果适当进行替换）来验证您可以连接到SSH主机。

```
ssh user@hostname
# Or for Windows when using a domain / AAD account
ssh user@domain@hostname
```

2. 在VS Code中, 从命令面板 (F1) 中选择**Remote-SSH: Connect to Host ...**, 并使用与步骤1 相同的命令。 user@hostname



3. 如果VS Code无法自动检测到您要连接的服务器的类型, 则会要求您手动选择类型。



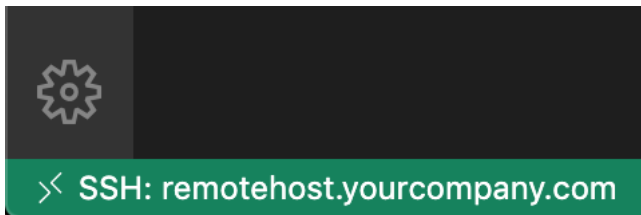
选择平台后, 它将存储在该属性下的VS Code设置中 (/docs/getstarted/settings), remote.SSH.remotePlatform 因此您可以随时对其进行更改。

4. 稍后, VS Code将连接到SSH服务器并进行自我设置。VS Code将使用进度通知使您保持最新状态, 并且可以在 Remote - SSH 输出通道中看到详细的日志。

提示: 连接挂起或失败? 请参阅故障排除提示 (/docs/remote/troubleshooting#_troubleshooting-hanging-or-failing-connections)以获取有关解决常见问题的信息。

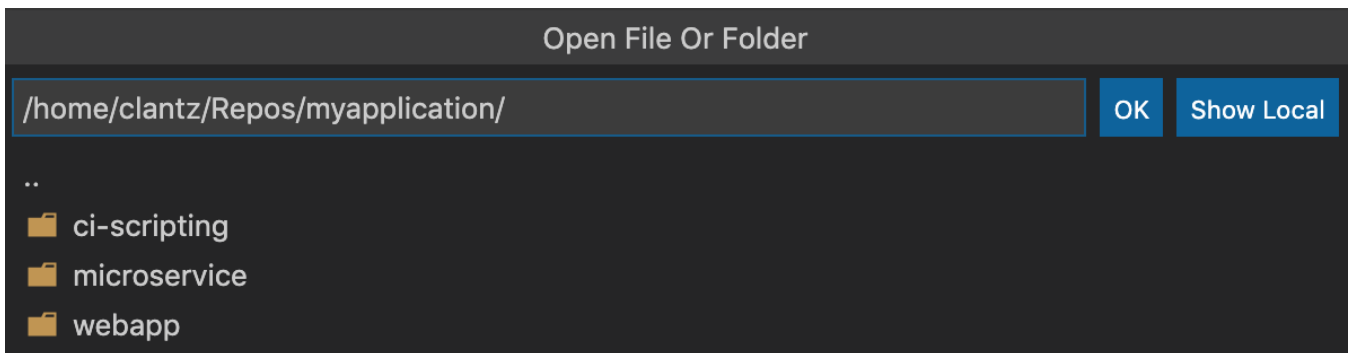
如果看到有关SSH文件权限的错误, 请参阅“修复SSH文件权限错误”部分 (/docs/remote/troubleshooting#_fixing-ssh-file-permission-errors)。

5. 连接后, 您将进入一个空窗口。您始终可以参考状态栏来查看连接到的主机。



在连接状态下, 单击状态栏项目将提供远程命令列表。

6. 然后, 您可以使用**文件>打开...**或**文件>打开工作区...**来打开远程计算机上的任何文件夹或工作区, 就像在本地一样!



从这里, 安装要连接到主机时要使用的所有扩展, 然后开始编辑!

注意: 在ARMv7l / ARMv8l glibc SSH主机上, 由于扩展程序内的x86编译本机代码, 某些扩展程序可能无法使用。

断开与远程主机的连接

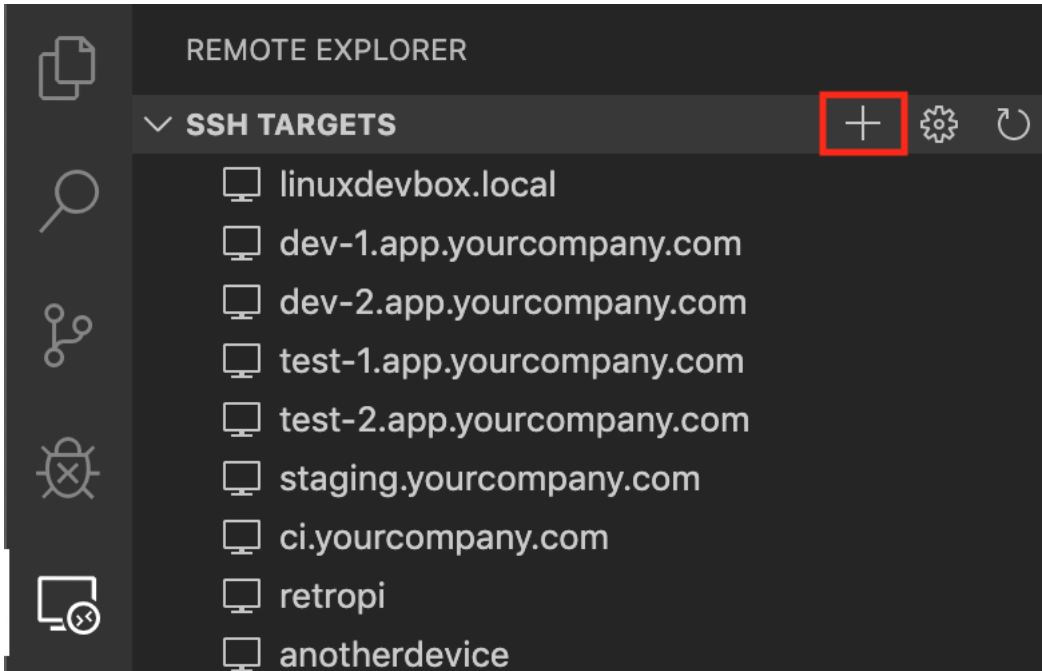
要在完成远程主机上的文件编辑后**关闭连接**, 请选择“**文件**”>“**关闭远程连接**”以断开与主机的**连接**。默认配置不包括此命令的键盘快捷键。您也可以直接退出VS Code以关闭远程连接。

记住主机和高级设置

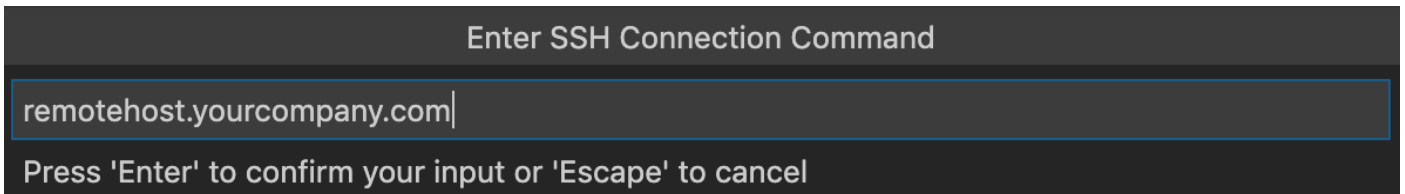
如果您有一组经常使用的主机, 或者需要使用一些其他选项连接到主机, 则可以将它们添加到遵循SSH配置文件格式 (http://man7.org/linux/man-pages/man5/ssh_config.5.html)的本地文件中。

为了简化设置, 该扩展程序可以指导您添加主机, 而无需手工编辑此文件。

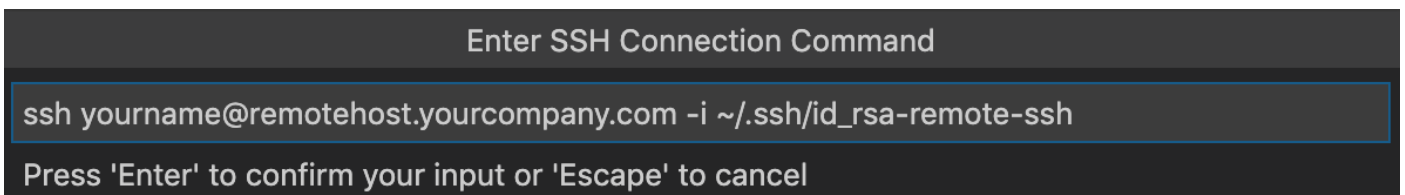
首先从命令面板（ F1 ）中选择“ 远程SSH： 添加新的SSH主机...”，或者在活动栏中的“ SSH 远程资源管理器”中单击“ 添加新”图标。



然后，将要求您输入SSH连接信息。您可以输入主机名：



或用于 ssh 从命令行连接到主机的完整命令：



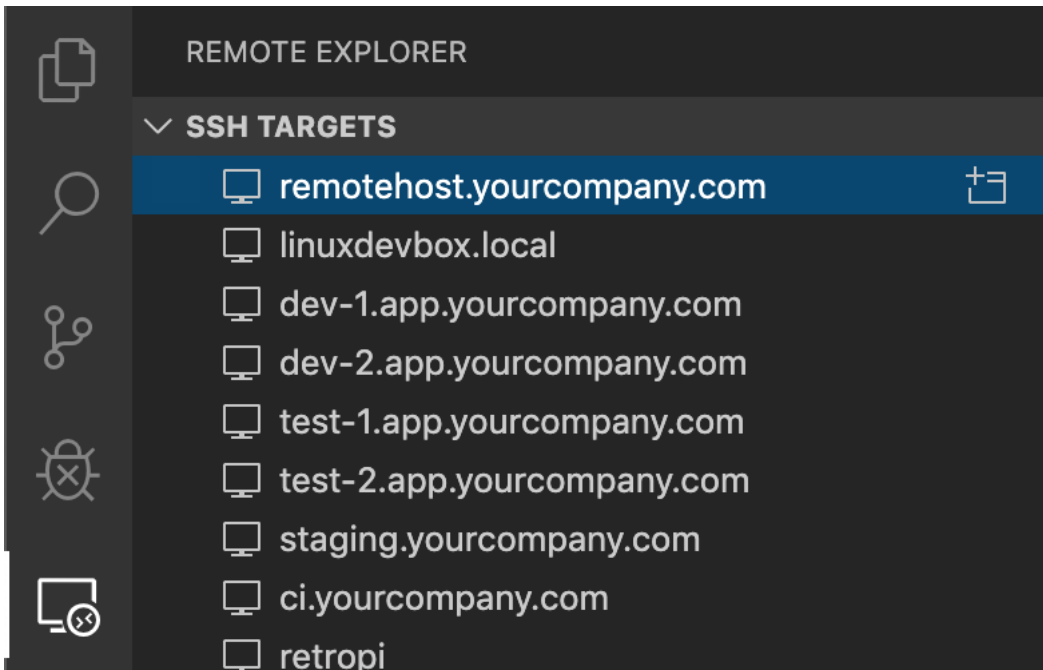
最后，系统会要求您选择要使用的配置文件。如果要使用与列出的配置文件不同的配置文件，也可以“remote.SSH.configFile”在用户 settings.json 文件中设置属性。该扩展程序将处理其余部分！

例如， ssh -i ~/.ssh/id_rsa-remote-ssh yourname@remotehost.yourcompany.com 在输入框中输入将生成以下条目：

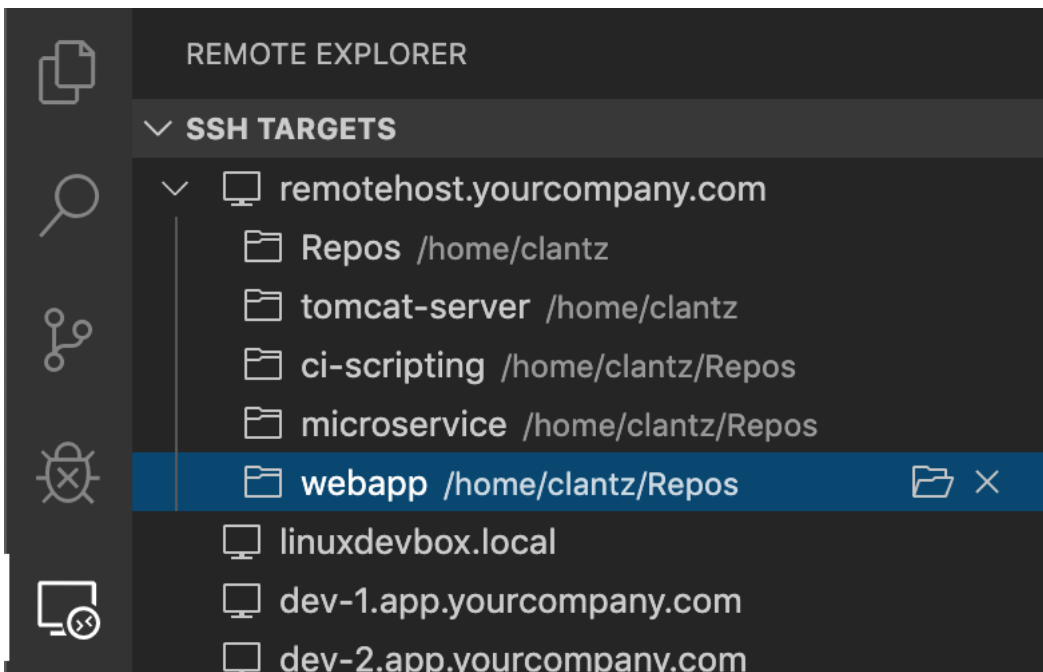
```
Host remotehost.yourcompany.com
  User yourname
  HostName another-host-fqdn-or-ip-goes-here
  IdentityFile ~/.ssh/id_rsa-remote-ssh
```

有关生成此处显示的密钥的详细信息，请参见提示和技巧 (/docs/remote/troubleshooting#_improving-your-security-with-a-dedicated-key)。您可以使用SSH配置文件格式 (http://man7.org/linux/man-pages/man5/ssh_config.5.html) 支持的任何内容手动编辑此文件，因此这只是一个示例。

从现在开始，当您从命令面板（ F1 ）或“ 远程资源管理器”的“ SSH目标”部分中选择“ 远程SSH： 连接到主机...”时，该主机将出现在主机列表中。



在**远程浏览器**允许你同时打开远程主机或文件夹以前打开直接打开一个新的空窗口。展开主机，然后单击要在主机上**打开的文件夹**旁边的“**打开文件夹**”图标。

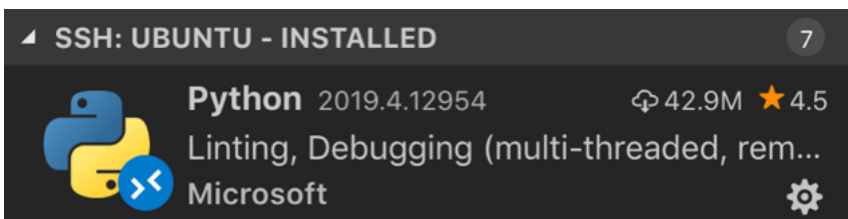


管理扩展

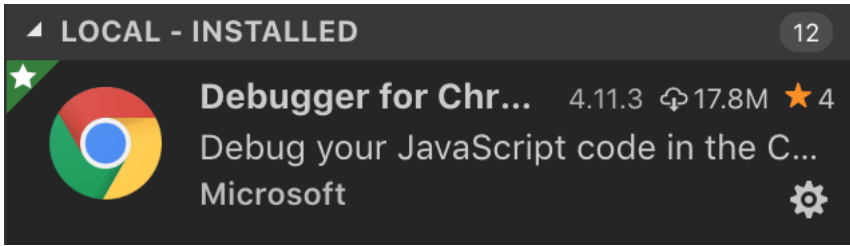
VS Code在以下两个位置之一运行扩展：在UI /客户端本地，或在SSH主机远程。虽然会在本地安装影响VS Code UI的扩展（例如主题和代码片段），但大多数扩展将驻留在SSH主机上。这样可以确保您拥有流畅的体验，并允许您从本地计算机在SSH主机上安装给定工作区的任何所需扩展。这样，您就可以从另一台配有扩展程序的计算机上准确地取回中断的地方。

如果您从“扩展”视图安装扩展，它将自动安装在正确的位置。安装后，您可以根据类别分组确定扩展的安装位置。

远程SSH主机将有一个类别：

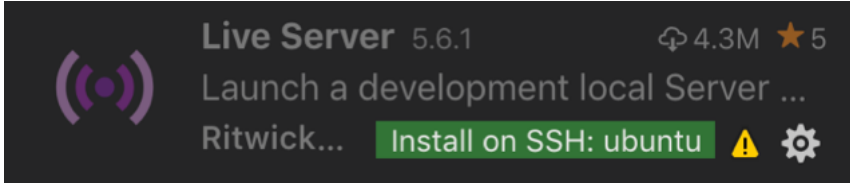


还有**本地安装**类别：



注意：如果您是扩展程序作者，并且发现扩展程序无法正常工作或安装在错误的位置，请参阅支持远程开发 (/api/advanced-topics/remote-extensions)以获取详细信息。

实际上需要远程运行的本地扩展将在“**本地安装**”类别中显示为灰色并禁用。选择**安装**在您的远程主机上安装扩展。



您还可以通过以下方法在SSH主机上安装所有本地安装的扩展，方法是：转到“扩展”视图，然后选择“**本地-已安装**”标题栏右侧的云按钮，选择“**在SSH中安装本地扩展：[主机名]**”。这将显示一个下拉列表，您可以在其中选择要在SSH主机上安装的本地安装扩展。

“始终安装”扩展

如果您希望始终在任何SSH主机上安装扩展，则可以使用中的 `remote.SSH.defaultExtensions` 属性指定哪些扩展 `settings.json`。例如，如果要安装GitLens (<https://marketplace.visualstudio.com/items?itemName=eamodio.gitlens>)和资源监视器 (<https://marketplace.visualstudio.com/items?itemName=mutantdino.resourcemonitor>)扩展，请如下指定其扩展ID：

```
"remote.SSH.defaultExtensions": [
  "eamodio.gitlens",
  "mutantdino.resourcemonitor"
]
```

高级：强制扩展在本地/远程运行

扩展通常经过设计和测试，可以在本地或远程运行，不能同时运行。但是，如果扩展支持，则可以强制其在 `settings.json` 文件中的特定位置运行。

例如，以下设置将强制Docker扩展在本地运行，而Debugger for Chrome扩展将在远程运行而不是默认设置：

```
"remote.extensionKind": {
  "ms-azuretools.vscode-docker": [ "ui" ],
  "msjsdiag.debugger-for-chrome": [ "workspace" ]
}
```

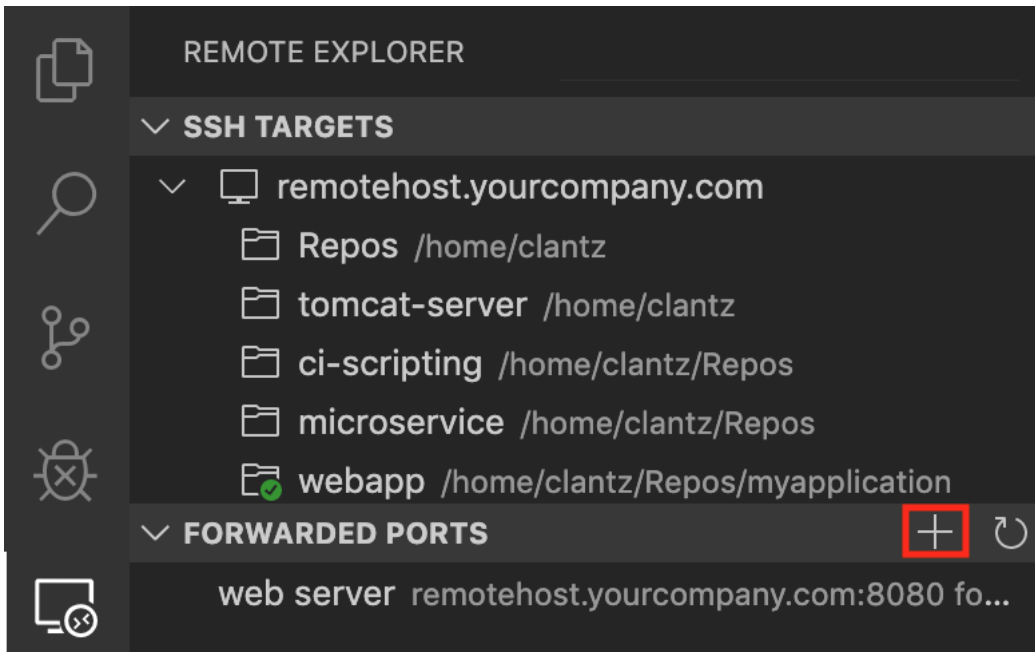
的值“ui”，而不是“workspace”将迫使扩展到在本地UI/客户端，而不是运行。通常，除非扩展文档中另有说明，否则应仅将其用于测试，因为它**可能会破坏扩展**。有关详细信息，请参见支持远程开发 (/api/advanced-topics/remote-extensions)的文章。

转发端口/创建SSH隧道

有时在开发时，您可能需要访问未公开公开的远程计算机上的端口。有两种方法可以使用SSH隧道 (<https://www.ssh.com/ssh/tunneling/example>)将所需的远程端口“转发”到本地计算机。

临时转发端口

连接到主机后，如果要在会话期间**临时转发**新端口，请在命令面板（F1）中选择“**转发端口**”，或在选择后单击**远程资源管理器**中的“**转发新端口**”图标。从活动栏中。



系统会要求您输入您要转发的端口，然后给它起一个名字。

Enter the port on "remotehost.yourcompany.com" that should be forwarded:

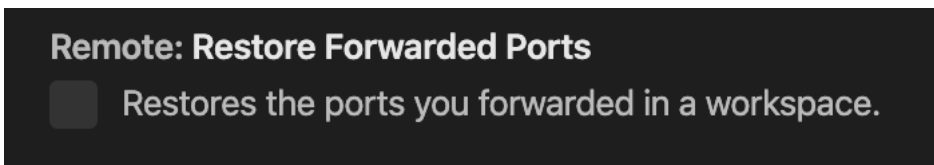
9000|

Press 'Enter' to confirm your input or 'Escape' to cancel

一条通知将告诉您应该用于访问远程端口的本地主机端口。例如，如果您转发了侦听端口3000的HTTP服务器，则该通知可能会告诉您该映射已映射到本地主机上的端口4123，因为3000已被使用。然后，您可以使用http://localhost:4123 (http://localhost:4123)连接到此远程HTTP服务器。

如果以后需要访问此信息，可在Remote Explorer的**转发端口**部分中找到。

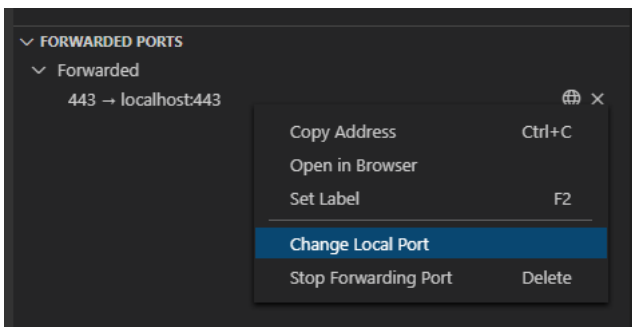
如果您希望VS Code记住已转发的任何端口，请在“设置”编辑器（Ctrl + ,）中选中“**远程：还原转发的端口**”或在中设置。`"remote.restoreForwardedPorts": true` settings.json



更改隧道上的本地端口

如果您希望隧道的本地端口与远程服务器的本地端口不同，则可以通过“**转发的端口**”面板进行更改。

右键单击要修改的隧道，然后在上下文菜单中选择“**更改本地端口**”。



始终转发端口

If you have ports that you **always want to forward**, you can use the `LocalForward` directive in the same SSH config file you use to remember hosts and advanced settings.

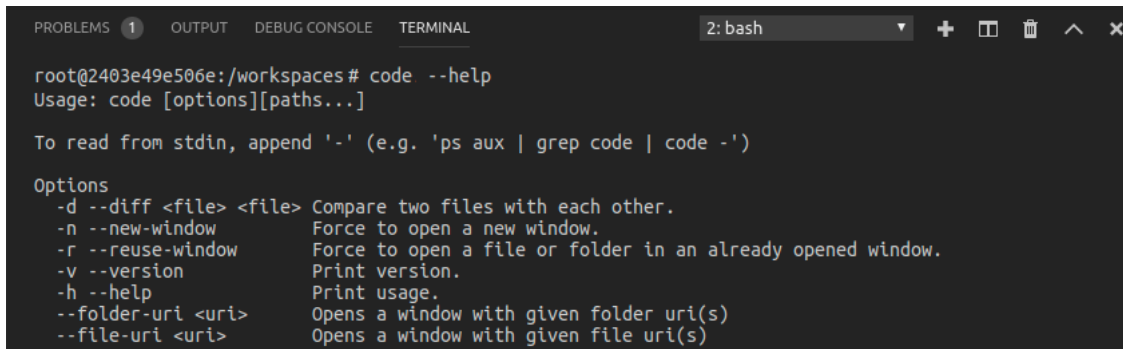
For example, if you wanted to forward ports 3000 and 27017, you could update the file as follows:

```
Host remote-linux-machine
  User myuser
  HostName remote-linux-machine.mydomain
  LocalForward 127.0.0.1:3000 127.0.0.1:3000
  LocalForward 127.0.0.1:27017 127.0.0.1:27017
```

Opening a terminal on a remote host

Opening a terminal on the remote host from VS Code is simple. Once connected, **any terminal window** you open in VS Code (**Terminal > New Terminal**) will automatically run on the remote host rather than locally.

You can also use the `code` command line from this same terminal window to perform a number of operations such as opening a new file or folder on the remote host. Type `code --help` to see all the options available from the command line.



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 2: bash
root@2403e49e506e:/workspaces# code --help
Usage: code [options][paths...]

To read from stdin, append '-' (e.g. 'ps aux | grep code | code -')

Options
-d --diff <file> <file> Compare two files with each other.
-n --new-window Force to open a new window.
-r --reuse-window Force to open a file or folder in an already opened window.
-v --version Print version.
-h --help Print usage.
--folder-uri <uri> Opens a window with given folder uri(s)
--file-uri <uri> Opens a window with given file uri(s)
```

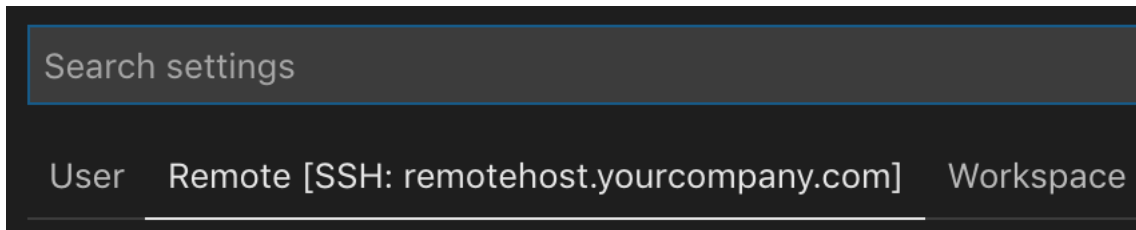
Debugging on the SSH host

Once you are connected to a remote host, you can use VS Code's debugger in the same way you would when running the application locally. For example, if you select a launch configuration in `launch.json` and start debugging (F5), the application will start on remote host and attach the debugger to it.

See the debugging (/docs/editor/debugging) documentation for details on configuring VS Code's debugging features in `.vscode/launch.json`.

SSH host-specific settings

当您连接到SSH主机时，也会重复使用VS Code的本地用户设置。虽然这可以使您的用户体验保持一致，但是您可能希望在本地图算机和每个主机之间更改某些设置。幸运的是，一旦连接到主机，您还可以通过运行“**偏好设置**”来设置特定于主机的设置：从命令面板（F1）**打开“远程设置”**命令，或者在“设置”编辑器中的“**远程**”选项卡上进行选择。每当您连接到主机时，这些设置将覆盖您已有的所有本地设置。



使用本地工具

Remote-SSH扩展不直接支持同步源代码或将本地工具与远程主机上的内容一起使用。但是，有两种方法可以使用适用于大多数Linux主机的通用工具进行此操作。具体来说，您可以：

1. 使用SSHFS挂载远程文件系统 (/docs/remote/troubleshooting#_using-sshfs-to-access-files-on-your-remote-host)。
2. 使用同步文件到/从远程主机到本地计算机 rsync (/docs/remote/troubleshooting#_using-rsync-to-maintain-a-local-copy-of-your-source-code)。

SSHFS是最方便的选项，不需要任何文件同步。但是，性能将比通过VS Code显著降低，因此最好用于单个文件编辑以及上载/下载内容。如果您需要使用一次可批量读取/写入多个文件的应用程序（例如本地源代码控制工具），则rsync是一个更好的选择。

已知限制

远程-SSH限制

- 强烈建议使用基于密钥的身份验证。为备用身份验证方法 (/docs/remote/troubleshooting#_enabling-alternate-ssh-authentication-methods)输入的密码和其他令牌不会保存。
- 不支持Alpine Linux和基于非glibc的Linux SSH主机。
- 较旧的（受社区支持）Linux发行版需要变通办法来安装所需的必备软件 (/docs/remote/linux)。
- Windows不支持PuTTY。
- 如果您使用SSH克隆Git存储库，并且您的SSH密钥具有密码短语，则在远程运行时VS Code的提取和同步功能可能会挂起。使用不带密码短语的SSH密钥，使用HTTPS克隆或从命令行运行 `git push` 以解决此问题。
- 本地代理设置不会在远程主机上重用，这会阻止扩展工作，除非在远程主机上配置了适当的代理信息（例如，具有适当代理信息的全局变量 `HTTP_PROXY` 或 `HTTPS_PROXY` 环境变量）。
- 有关与SSH相关的活动问题的列表，(https://aka.ms/vscode-remote/ssh/issues)请参见此处 (https://aka.ms/vscode-remote/ssh/issues)。

Docker扩展限制

尽管Docker扩展既可以在本地也可以在远程运行，但是如果已在本地安装，则必须先在本地图算机卸载SSH主机，才能将其安装在远程SSH主机上。我们将在将来的VS Code版本中解决此问题。

扩展限制

许多扩展无需修改即可在远程SSH主机上运行。但是，在某些情况下，某些功能可能需要更改。如果遇到扩展问题，则总结了一些常见问题和解决方案 (/docs/remote/troubleshooting#_extension-tips)，您可以在报告问题时向扩展作者提及。

另外，由于仅支持x86_64的扩展中的本机模块或运行时，某些安装在ARMv7I (AArch32) / ARMv8I (AArch64) 设备上的扩展可能无法正常工作。在这些情况下，扩展将需要通过编译/包括用于ARMv7I / ARMv8I的二进制文件来选择支持这些平台。

常见问题

如何在...上设置SSH客户端？

有关详细信息，请参见安装受支持的SSH客户端 (/docs/remote/troubleshooting#_installing-a-supported-ssh-client)。

如何在...上设置SSH服务器？

有关为主机设置SSH服务器 (/docs/remote/troubleshooting#_installing-a-supported-ssh-server)的详细信息，请参见安装受支持的SSH服务器 (/docs/remote/troubleshooting#_installing-a-supported-ssh-server)。

我可以不使用其他/其他身份验证机制（例如密码）登录SSH服务器吗？ #

是的，应该提示您自动输入令牌或密码。但是，密码不会保存，因此使用基于密钥的身份验证 (/docs/remote/troubleshooting#_configuring-key-based-authentication)通常更为方便。

如何解决有关“权限错误”的SSH错误？

有关解决这些类型的错误的详细信息，请参见修复SSH文件权限 (/docs/remote/troubleshooting#_fixing-ssh-file-permission-errors)错误。

需要在远程SSH主机上安装哪些Linux软件包/库？

大多数Linux发行版都不需要其他依赖项安装步骤。对于SSH，Linux主机需要安装Bash (/bin/bash)，，或者安装Bash ()，tar 并且某些简化的发行版中可能缺少这些实用程序。远程开发还需要内核> = 3.10，glibc> = 2.17，libstdc ++> = 3.4.18。当前仅支持基于glibc的发行版，因此扩展不支持Alpine Linux (https://alpinelinux.org)。curl wget (https://alpinelinux.org)

有关详细信息，请参见Linux先决条件 (/docs/remote/linux)。

在远程计算机/ VM上运行时，VS Code Server对连接的要求是什么？

VS Code Server的安装要求您的本地计算机具有与以下设备的出站HTTPS（端口443）连接：

- update.code.visualstudio.com
- vscode.blob.core.windows.net
- *.vo.msecnd.net (Azure CDN)

默认情况下，“远程-SSH”将尝试在远程主机上进行下载，但如果启用 remote.SSH.allowLocalServerDownload，则扩展名将退回到本地下载VS Code Server并在建立连接后进行远程传输。

您可以使用“扩展：从VSIX安装...”命令在没有Internet连接的情况下手动安装扩展，但是如果使用扩展面板安装扩展，则本地计算机和VS Code Server服务器将需要出站HTTPS（端口443）访问权限：

- marketplace.visualstudio.com
- *.vo.msecnd.net (Azure CDN)
- *.gallerycdn.vsassets.io (Azure CDN)

最后，某些扩展程序（例如C#）从 download.microsoft.com 或下载辅助依赖项 download.visualstudio.microsoft.com。其他（例如Visual Studio Live Share (https://docs.microsoft.com/visualstudio/liveshare/reference/connectivity#requirements-for-connection-modes)）可能具有其他连接要求。如果遇到麻烦，请查阅扩展的文档以获取详细信息。

服务器和VS Code客户端之间的所有其他通信都是通过经过身份验证的安全SSH隧道完成的。

我可以在远程SSH主机上的源代码上使用本地工具吗？

是。通常，这样做是使用SSHFS (/docs/remote/troubleshooting#_using-sshfs-to-access-files-on-your-remote-host)或者通过使用 rsync (/docs/remote/troubleshooting#_using-rsync-to-maintain-a-local-copy-of-your-source-code)得到您的本地机器上的文件的副本。SSHFS挂载远程文件系统是需要编辑单个文件或浏览源树并且不需要使用任何同步步骤的情况的理想选择。但是，使用诸如批量管理文件的源代码控制工具之类的方法并不理想。在这种情况下，这种 rsync 方法更好，因为您可以在本地计算机上获得远程源代码的完整副本。有关详细信息，请参见提示和技巧 (/docs/remote/troubleshooting#_using-sshfs-to-access-files-on-your-remote-host)。

仅对远程主机具有SFTP / FTP文件系统访问权限（无外壳访问权限）时，可以使用VS Code吗？

一些云平台仅为开发人员提供远程文件系统访问，而不提供直接的Shell访问。VS Code远程开发在设计时就没有考虑此用例，因为它否定了性能和用户体验优势。

但是，通常可以通过将诸如SFTP的 (https://marketplace.visualstudio.com/items?itemName=liximomo.sftp)扩展与Node.js (/docs/nodejs/nodejs-debugging#_remote-debugging)，Python (/docs/python/debugging#_remote-debugging)，C# (https://marketplace.visualstudio.com/items?itemName=ms-dotnettools.csharp)或其他的远程调试功能结合使用来处理此用例。

作为扩展作者，我需要做什么？