


TOPICS 常见问题

远程开发常见问题

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

本文介绍了每个Visual Studio Code**远程开发**扩展的常见问题。有关设置和使用它们各自功能的更多详细信息，请参见SSH (/docs/remote/ssh)，Containers (/docs/remote/containers)和WSL (/docs/remote/wsl)文章。或者尝试逐步教程， (/docs/remote/remote-tutorials)以帮助您在远程环境中快速运行。

有关Visual Studio Codespaces (<https://aka.ms/vso>)的常见问题 (<https://aka.ms/vso-docs/faq>)，请参见服务文档中 (<https://aka.ms/vso-docs/faq>)的FAQ (<https://aka.ms/vso-docs/faq>)。

一般

什么是Visual Studio Code远程开发？

Visual Studio Code **远程开发扩展包** (<https://aka.ms/vscode-remote/download/extension>)允许您打开容器中，远程计算机上（通过SSH）或Windows Subsystem for Linux 中的任何文件夹，并利用VS Code的全部功能集。这意味着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），调试等，而无论您的代码位于或托管于何处。

与本地编辑相比，VS Code远程开发具有哪些优势？

远程开发的一些好处包括：

- 能够在不同于本地运行的操作系统上进行编辑，构建或调试。
- 能够在与目标部署环境匹配的环境中进行开发。
- 使用比本地计算机更大或更专业的硬件进行开发。
- 能够编辑存储在另一个位置（例如，云或客户站点）中的代码。
- 开发人员环境的沙盒化，可避免冲突，提高安全性并加快注册速度。

与使用网络共享或同步文件相比，VS Code远程开发提供了显著更好的性能以及对开发环境和工具的更好控制。

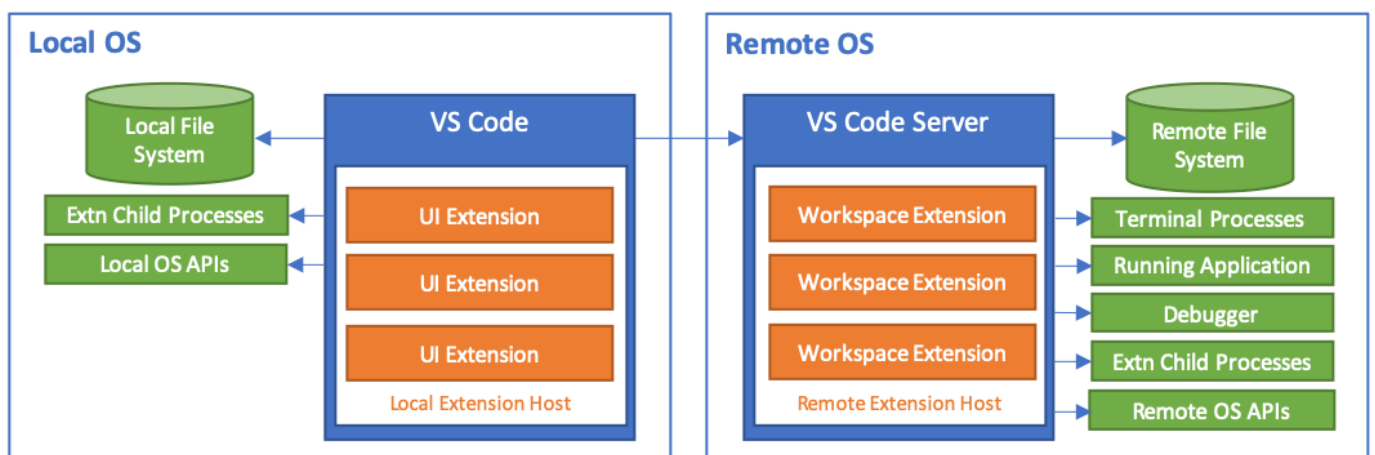
远程开发扩展与Visual Studio Codespaces有何关系？

Visual Studio Codespaces (<https://aka.ms/vso>)是一项服务，提供可从VS Code和新的基于浏览器的编辑器访问的托管的云托管开发环境。该服务还允许VS Code和基于浏览器的编辑器访问自托管环境（台式机或服务器），而无需SSH服务器甚至直接的网络路由。您可以在其文档中 (<https://aka.ms/vso-docs>)阅读有关Visual Studio Codespaces的更多信息。

尽管Remote Development和Visual Studio Codespaces扩展共享技术和功能，但是Remote Development扩展是单独发布的，并且可以独立于Visual Studio Codespaces运行（因此不需要Visual Studio Codespaces计划）。

远程开发扩展如何工作？

通过将某些命令的执行移至“远程服务器”，Visual Studio Code Remote Development允许本地VS Code安装与其他计算机（无论是虚拟机还是物理机）上的源代码和运行时环境透明地交互。该**VS代码服务器**由VS代码当您连接到远程端点，并且可以承载扩展迅速安装的直接交互与远程工作区，机器和文件系统。



有关扩展的更多详细信息，请参见支持远程开发 (/api/advanced-topics/remote-extensions)。

远程开发扩展如何确保对远程计算机，VM或容器的访问？

Visual Studio Code远程开发使用现有的众所周知的传输方式（例如安全外壳） (https://en.wikipedia.org/wiki/Secure_Shell)来验证和保护流量。除了这些众所周知的安全运输所使用的端口以外，无需公开开放任何端口。

注入的VS Code服务器以与登录计算机时相同的用户身份运行，从而确保未经授权，不会给VS Code及其扩展名带来不当的提升访问权限。该服务器由VS Code启动和停止，并且未连接到任何用户或全局登录或启动脚本中。VS Code管理着服务器的生命周期，因此您不必担心它是否正在运行。

VS Code Server可以单独安装或使用吗？

否。VS Code 服务器是远程开发扩展的组件，由 VS Code 客户端进行管理。当它连接到端点时，它将由 VS Code 自动安装和更新，如果单独安装，可能会很快过时。它不打算或许可通过其他客户端使用。

VS Code Server 的连接要求是什么？

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

- `update.code.visualstudio.com`
- `*.vo.mssecnd.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。

远程-容器扩展始终在本地下载并转移到容器中。

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或 `devcontainer.json` 安装扩展，则本地计算机和 VS Code Server 将需要出站 HTTPS（端口 443）访问权限：

- `marketplace.visualstudio.com`
- `vscode.blob.core.windows.net`
- `*.vo.mssecnd.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以下传输通道（取决于扩展名）完成的：

- SSH：经过身份验证的安全 SSH 隧道。
- 容器：Docker 配置的通信通道（通过 `docker exec`）。
- WSL：随机本地端口。

您可以在网络连接文章中（/docs/setup/network#_common-hostnames）找到 VS Code 本身需要访问的位置列表。

使用远程扩展时，为什么在 Docker 扩展中看不到本地容器？

默认情况下，Docker 扩展将远程运行。尽管在某些情况下这是明智的默认设置，但这意味着当 VS Code 连接到远程 SSH 主机，容器或 WSL 时，扩展名可能不会显示本地容器。

您可以使用以下解决方案之一来解决此问题：

- 打开一个新的本地窗口（**File > New Window**），并使用它来处理本地容器。
- 安装“远程-容器”（<https://aka.ms/vscode-remote/download/containers>）扩展并在需要查看本地容器的情况下使用“远程资源管理器”（/docs/remote/containers#_option-1-use-the-containers-remote-explorer）。
- **仅适用于远程 WSL**：使用 WSL 2（<https://docs.docker.com/docker-for-windows/wsl-tech-preview/>）的 Docker 技术预览版，（<https://docs.docker.com/docker-for-windows/wsl-tech-preview/>）或将 Docker Desktop 配置为在 WSL 1 中使用（<https://nickjanetakis.com/blog/setting-up-docker-for-windows-and-wsl-to-work-flawlessly>）。
- **远程-仅限容器**：转发 Docker 套接字并将 Docker CLI（<https://aka.ms/vscode-remote/samples/docker-from-docker>）（仅）安装（<https://aka.ms/vscode-remote/samples/docker-from-docker>）在容器中。
- 使用 `extensionKind` 属性（/docs/remote/containers#_advanced-forcing-an-extension-to-run-locally-or-remotely）将扩展名（/docs/remote/containers#_advanced-forcing-an-extension-to-run-locally-or-remotely）强制为 `ui`。但是，这将阻止某些命令起作用。

要使用远程开发，需要在主机上安装哪些 Linux 软件包或库？

远程开发需要内核 >= 3.10，glibc >= 2.17 和 libstdc++ >= 3.4.18。最近的基于 x86_64 基于 glibc 的发行版提供了最好的支持，但是确切的要求可能因发行版而异。

远程-容器和远程-WSL 支持基于 musl 的 Alpine Linux（<https://alpinelinux.org>），远程-SSH 中提供 ARMv7l（AArch32）/ ARMv8l（AArch64）。但是，某些扩展中的本机依赖性可能导致它们在非 x86_64 glibc 发行版上不起作用。请注意，仅在 VS Code Insiders 中（<https://code.visualstudio.com/insiders/>）提供实验性的 ARMv8l（AArch64）。

有关其他详细信息，请参见使用 Linux（</docs/remote/linux>）进行远程开发（</docs/remote/linux>）。

我可以安装单个扩展而不是扩展包吗？

是。该远程开发扩展包（<https://aka.ms/vscode-remote/download/extension>）提供了一个便捷的方式，他们被释放，为您访问所有最新的远程功能。但是，您始终可以从 Marketplace 或 VS Code Extensions 视图中安装各个扩展。

- 远程-SSH（<https://aka.ms/vscode-remote/download/ssh>）
- 远程-容器（<https://aka.ms/vscode-remote/download/containers>）
- 遥控器-WSL（<https://aka.ms/vscode-remote/download/wsl>）

遥控器-WSL

与使用 WSL 作为终端相比，扩展的优点是什么？

您可以将 WSL 视为运行在 Windows 上的 Linux 机器，您可以在其中安装 Linux 特定的框架/工具（例如 Python，Go，Rust 等），而不会影响 Windows 的设置。然后，您可以使用 VS Code 和 Remote-WSL 扩展来在 WSL 中安装的内容（与 Windows 上安装的内容隔离）的上下文中进行开发。

例如，您可以在WSL（编译器，调试器，lint等）中安装Go堆栈。如果仅在Windows上运行VS Code，则还必须在其中安装相同的Go堆栈，以获取诸如智能补全，调试和“转到定义”导航之类的功能。而且由于语言服务在Windows上运行，因此它们不知道WSL中包含什么。

的确可以在Windows的WSL中运行二进制文件，反之亦然，但是常规的VS Code扩展不知道如何执行此操作。这就是我们开始在WSL中支持调试的方式，但是很快意识到我们必须更新所有扩展才能了解WSL。

相反，我们决定让部分WS Code运行在WSL中，并让Windows上运行的UI与运行在WSL中的VS Code服务器进行对话。这就是Remote-WSL扩展启用的功能，Go扩展与其他Go工具（编译器，调试器，lints）一起在WSL中运行，而VS Code在Windows上运行。

通过这种方法，诸如智能完成之类的语言功能就可以与WSL中的功能配合使用，而无需在Windows上进行任何设置。您不必担心路径问题，也不必在Windows上设置不同版本的开发堆栈。如果要部署应用程序到Linux，则可以将WSL实例设置为看起来像运行时环境，同时在Windows上仍可获得丰富的编辑体验。

远程-容器

“开发容器定义”是否定义了应用程序的部署方式？

否。开发容器定义了一个环境，您可以在其中部署应用程序之前准备好进行部署。尽管部署和开发容器可能彼此相似，但是您可能不想在开发过程中使用的部署映像中包含工具。

该vscode-DEV-容器库 (<https://aka.ms/vscode-dev-containers>)包括一组用于一些通用开发环境中开发容器定义。如果您更喜欢使用备用容器构建或部署工作流，则也可以在 (/docs/remote/containers#_attaching-to-running-containers)不设置开发容器定义的情况下附加到正在运行的容器 (/docs/remote/containers#_attaching-to-running-containers)。

“开发容器定义”是否定义了应用程序的构建方式？喜欢Buildpacks吗？

号的Buildpacks (<https://buildpacks.io/>)概念的重点是通过一系列定义的步骤以源代码，并产生展开的容器的图像。开发容器是一个环境，您可以在其中构建应用程序之前先对其进行开发。因此，它们是互补的概念。

扩展作者

作为扩展作者，我需要做什么？

VS Code扩展API提取了本地/远程细节，因此大多数扩展无需修改即可使用。但是，给定扩展可以使用所需的任何节点模块或运行时，在某些情况下可能需要进行调整。我们建议您测试扩展程序（尤其是在容器中），以确保不需要更新。有关详细信息，请参见支持远程开发 (</api/advanced-topics/remote-extensions>)。

当用户远程连接时，扩展程序可以访问本地资源或API吗？

当VS Code连接到远程环境时，扩展被分类为UI或Workspace扩展。UI扩展在**本地扩展主机**中运行，可以提供UI或个性化功能（例如主题），并且可以访问本地文件或API。工作区扩展在带有工作区的**远程扩展主机**中运行，并且可以完全访问源代码，远程文件系统和远程API。尽管Workspace扩展不专注于UI定制，但它们也可以贡献资源管理器，视图和其他UI元素。

当用户安装扩展程序时，VS Code会尝试推断正确的位置并根据其类型进行安装。不需要像主题和其他UI自定义那样远程运行的扩展会自动安装在UI端。所有其他功能都功能最全，因此被视为Workspace扩展。但是，扩展程序作者也可以使用中的 `extensionKind` 属性覆盖此位置 `package.json`。

有关其他详细信息，请参见支持远程开发 (</api/advanced-topics/remote-extensions>)。

许可和隐私

位置

您可以在此处找到VS Code远程开发扩展的许可证：

- 远程SSH许可证 (<https://marketplace.visualstudio.com/items/ms-vscode-remote.remote-ssh/license>)
- 远程WSL许可证 (<https://marketplace.visualstudio.com/items/ms-vscode-remote.remote-wsl/license>)
- 远程容器许可证 (<https://marketplace.visualstudio.com/items/ms-vscode-remote.remote-containers/license>)

为什么远程开发扩展或其组件不是开源的？

Visual Studio Code远程开发扩展及其相关组件使用开放式计划，发布和功能请求流程 (<https://aka.ms/vscode-remote/feedback>)，但当前不是开源的。这些扩展共享源代码，这些源代码也用于完全托管的远程开发服务中，例如Visual Studio Codespaces (<https://aka.ms/vso>)及其相关扩展。鉴于这些服务也将支持其他专有产品（例如Visual Studio IDE），因此这些扩展可以在Microsoft预发行许可下使用，例如其他基于服务的跨产品扩展，例如Visual Studio IntelliCode (<https://marketplace.visualstudio.com/items/VisualStudioExptTeam.vscodeintellicode/license>)和Visual Studio Live Share (<https://marketplace.visualstudio.com/items/MS-vsliveshare.vsliveshare-pack/license>)。在预览期间。

有关更多信息，请参见Visual Studio代码和“代码-OSS”差异 (<https://github.com/microsoft/vscode/wiki/Differences-between-the-repository-and-Visual-Studio-Code>)和Microsoft扩展许可 (</docs/supporting/oss-extensions>)文章。

退出“预览”后，您是否需要为远程开发扩展付费？

不，他们将保持免费。将来，我们可能会提供其他“高级”开发人员服务，例如Visual Studio Codespaces (<https://aka.ms/vso>)，这些服务会提供其他功能，但是这些扩展将是免费的。

远程开发扩展的连接位置是否有限制？

您可以自由地将扩展用于个人或公司用途，以连接到自己的物理机，虚拟机或容器。这些可以在内部，在您自己的私有云或数据中心，在Azure或其他云/非云托管提供商中使用。您不能在扩展或其相关组件之上构建公共产品或服务（请参阅下一个问题）。

我可以使用VS Code远程开发扩展来构建自己的产品或服务吗？

您可以将扩展程序与自己的内部或私有服务一起使用。您不能在VS Code远程开发扩展或它们的相关组件（例如VS Code服务器）之上构建公共或商业服务。您不能创建其他扩展来扩展或操纵远程开发扩展。尽管许可证规定您可能不会“将软件作为独立或集成产品提供，也不能与您的任何应用程序组合以供他人使用”，但是您可以记录如何将扩展与服务结合使用。

我可以在自己的公共服务产品中重新打包或重用VS Code Server吗？

否。许可证规定您不得“将软件作为独立或集成产品提供，或与您的任何应用程序组合以供他人使用”，这意味着您不得在VS Code之上构建公共产品或服务服务器。

我有一个问题，我是否可以对X使用扩展名，我可以问谁？

如果您有任何疑问，请通过VSCode@microsoft.com (mailto:VSCode@microsoft.com)给我们发送电子邮件。

GDPR和VS Code远程开发

VS Code远程开发扩展遵循GDPR策略，就像Visual Studio Code本身一样。有关更多详细信息，请参见常规FAQ (/docs/supporting/faq#_gdpr-and-vs-code)。

问题或反馈 #

有问题或反馈吗？


- 请参阅提示和技巧 (/docs/remote/troubleshooting)。
- 搜索堆栈溢出 (https://stackoverflow.com/questions/tagged/vscode-remote)。
- 添加功能请求 (https://aka.ms/vscode-remote/feature-requests)或报告问题 (https://aka.ms/vscode-remote/issues/new)。

该文档对您有帮助吗？

☒ 是 ☐ 没有

6/10/2020

你好，西雅图。 按照@code (https://go.microsoft.com/fwlink/?LinkID=533687) Star 98,787

支持 (https://support.microsoft.com/en-us/getsupport?wf=0&tenant=ClassicCommercial&oaspsworkflow=start_1.0.0.0&locale=en-us&supportregion=en-us&pesid=16064&ccsid=636196895839595242)
隐私 (https://privacy.microsoft.com/en-us/privacystatement) 使用条款 (https://www.microsoft.com/en-us/legal/intellectualproperty/copyright/default.aspx) 执照 (License)
 Microsoft (https://www.microsoft.com)
©2020 Microsoft