


TOPICS 总览 ▼

使用容器

 (https://github.com/Microsoft/vscode-docs/blob/master/docs/containers/overview.md)

该泊坞窗 (<https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker>) 扩展可以很容易地构建，管理和部署从Visual Studio代码集装箱的应用。

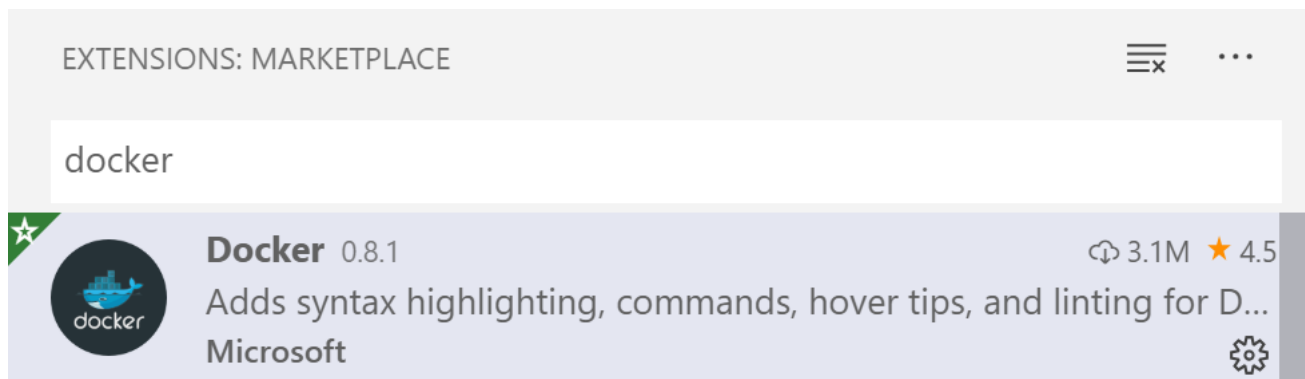
该页面概述了Docker扩展功能。使用侧面菜单可了解有关感兴趣主题的更多信息。如果您刚刚开始Docker开发，请先阅读Docker应用程序开发 (<https://docs.docker.com/develop/>) 以了解Docker的关键概念。

安装

(<https://docs.docker.com/install/>) 在您的机器上安装Docker (<https://docs.docker.com/install/>) 并将其添加到系统路径。

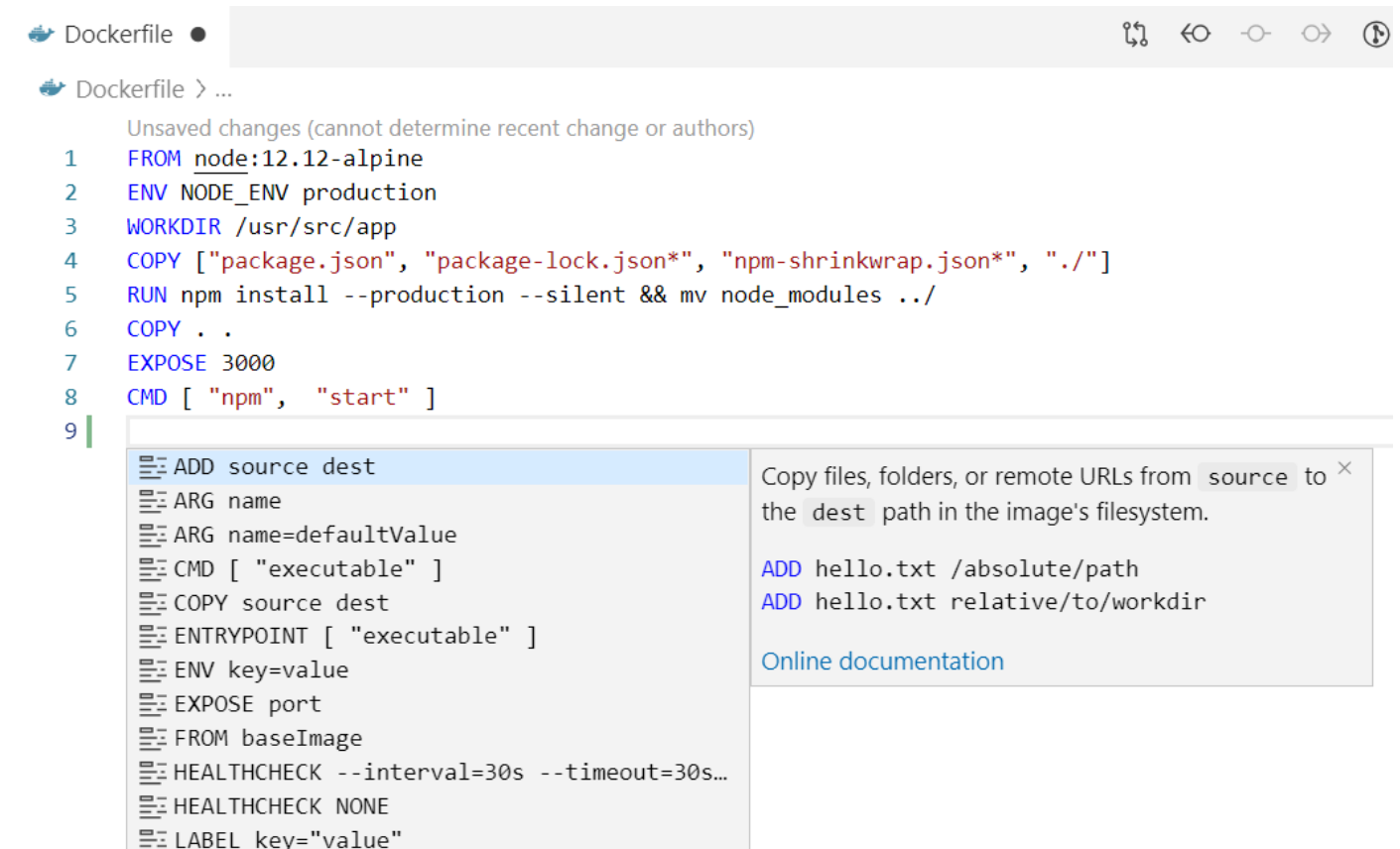
在Linux上，还应该为 (<https://docs.docker.com/install/linux/linux-postinstall/#manage-docker-as-a-non-root-user>) 将用于运行VS Code 的非root用户帐户启用Docker CLI (<https://docs.docker.com/install/linux/linux-postinstall/#manage-docker-as-a-non-root-user>)。

要安装扩展，请打开“扩展”视图（Ctrl + Shift + X），搜索 docker 以过滤结果，然后选择Microsoft编写的Docker扩展。



编辑Docker文件

您可以在编辑 Dockerfile 和 docker-compose.yml 文件时获得IntelliSense，并提供常用命令的完成功能和语法帮助。



此外，您可以使用“问题”面板（Ctrl + Shift + M）查看 Dockerfile 和 docker-compose.yml 文件的常见错误。

生成Docker文件

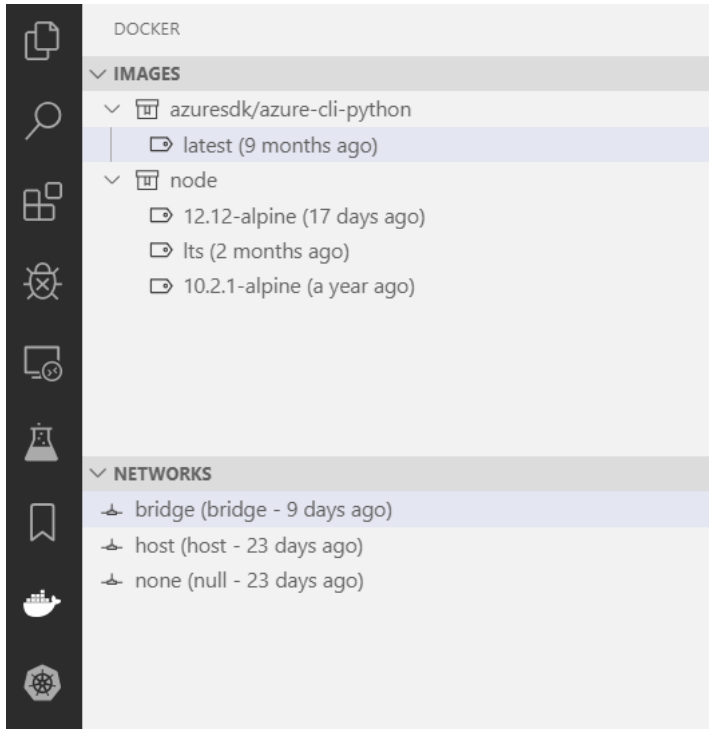
您可以通过打开命令面板（Ctrl + Shift + P）并使用**Docker将Docker文件添加到工作区：将Docker Files添加到工作区**命令。该命令将生成 Dockerfile 和 .dockerignore 文件，并将它们添加到您的工作区中。该命令还将查询您是否还需要添加Docker Compose文件。这是可选的。

该扩展可以识别使用最流行的开发语言（C#，Node.js，Python，Ruby，Go和Java）的工作区，并相应地自定义生成的Docker文件。

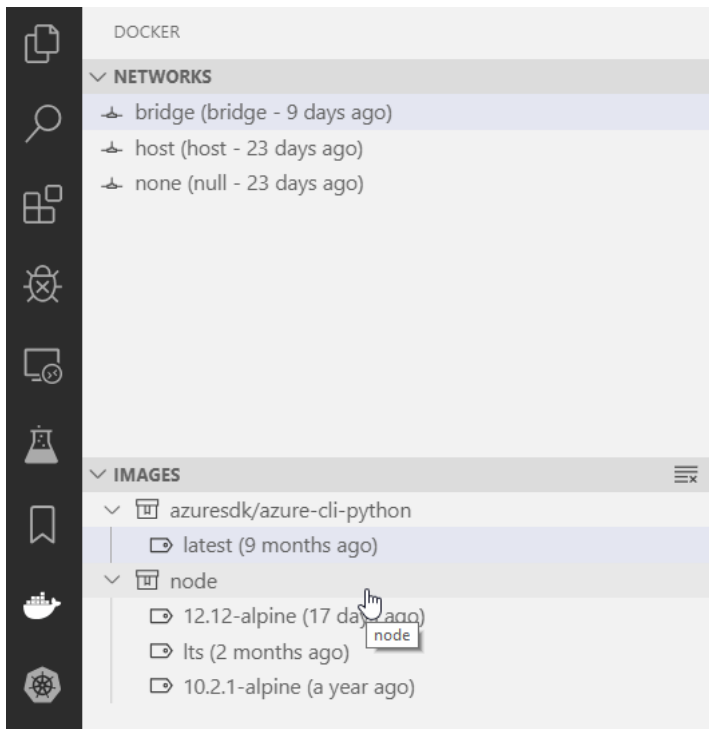
Docker视图

Docker扩展为VS Code提供了Docker视图。Docker视图使您可以检查和管理Docker资产：容器，映像，卷，网络和容器注册表。如果安装了Azure帐户扩展，则也可以浏览Azure容器注册表。

右键单击菜单可访问每种资产的常用命令。

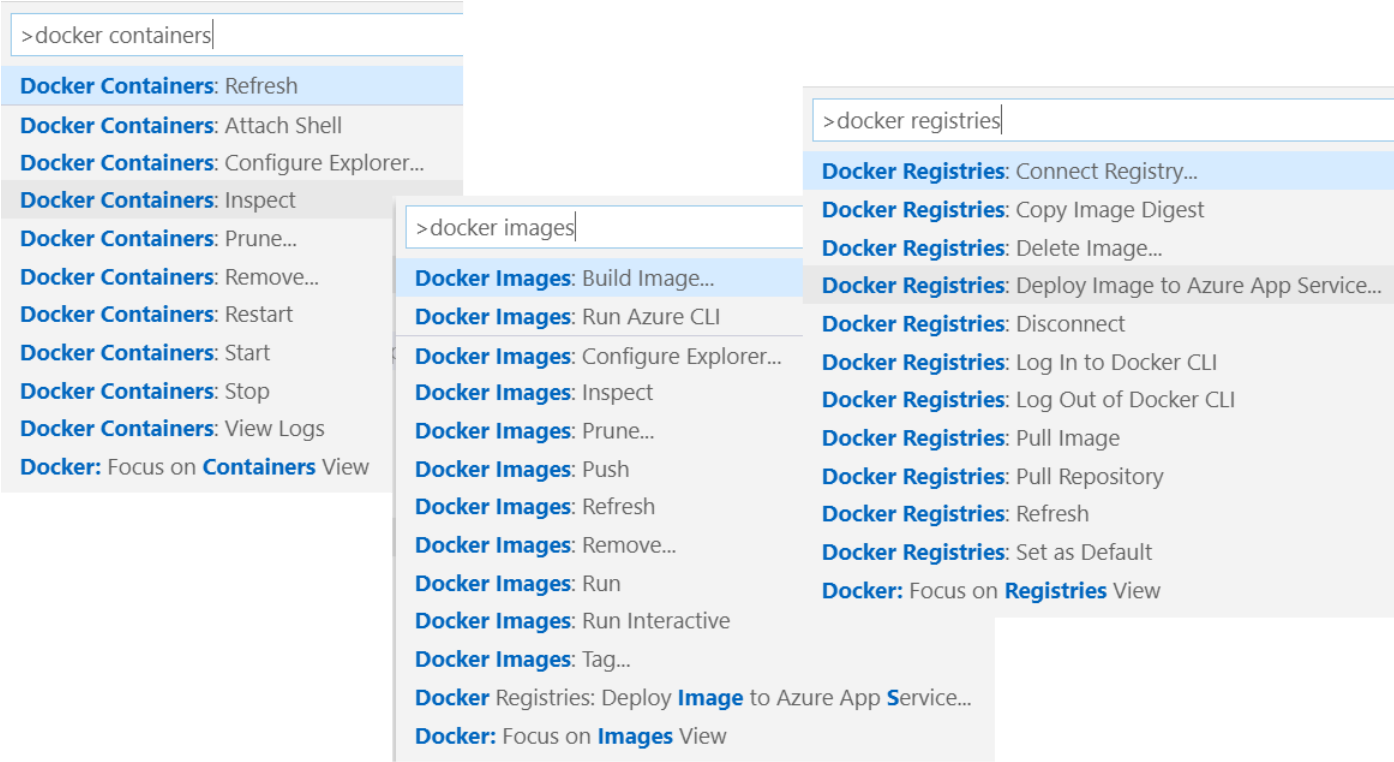


您可以通过使用鼠标上下拖动Docker视图窗格来重新排列Docker视图窗格，并使用上下文菜单隐藏或显示它们。



Docker命令

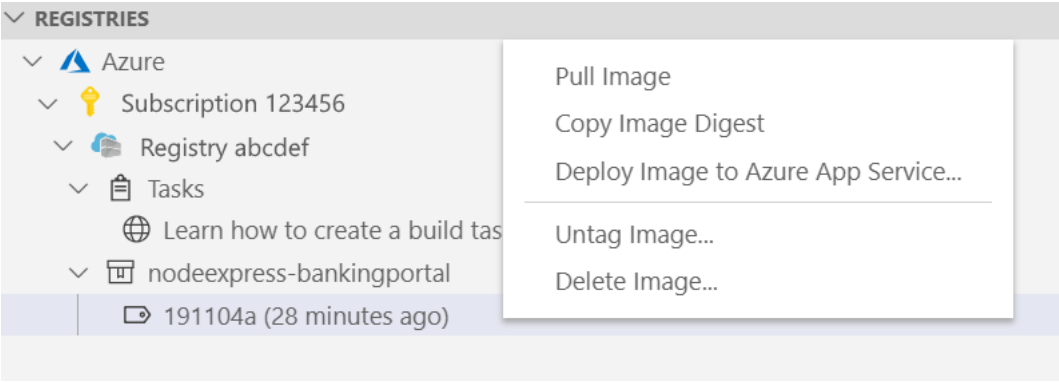
许多最常见的Docker命令都内置在命令面板中：



您可以运行Docker命令来管理映像 (<https://docs.docker.com/engine/reference/commandline/image/>)，网络 (<https://docs.docker.com/engine/reference/commandline/network/>)，卷 (<https://docs.docker.com/engine/reference/commandline/volume/>)，映像注册表 (<https://docs.docker.com/engine/reference/commandline/push/>)和Docker Compose (<https://docs.docker.com/compose/reference/overview/>)。另外，**Docker: Prune System**命令将删除停止的容器，悬空的映像以及未使用的网络和卷。

使用图像注册表

您可以显示内容并通过Azure容器注册表 (<https://docs.microsoft.com/azure/container-registry/>)，Docker Hub (<https://hub.docker.com/>)，GitLab (<https://gitlab.com/>)等来推/拉/删除图像：



Azure容器注册表中的图像可以直接从VS Code部署到Azure App Service；请参阅将映像部署到Azure App Service (</docs/containers/app-service>)页面。有关如何向注册表进行身份验证和使用的 (</docs/containers/quickstart-container-registries>)更多信息，请参见使用容器注册表 (</docs/containers/quickstart-container-registries>)页面。

调试容器内运行的服务

您可以调试在容器内运行的使用.NET (C #) 和Node.js构建的服务。该扩展程序提供了自定义任务，这些任务有助于在调试器下启动服务以及将调试器附加到正在运行的服务实例。有关更多信息，请参见调试容器应用程序 (</docs/containers/debug-common>) 和扩展属性和任务 (</docs/containers/reference>)页面。

Azure CLI集成

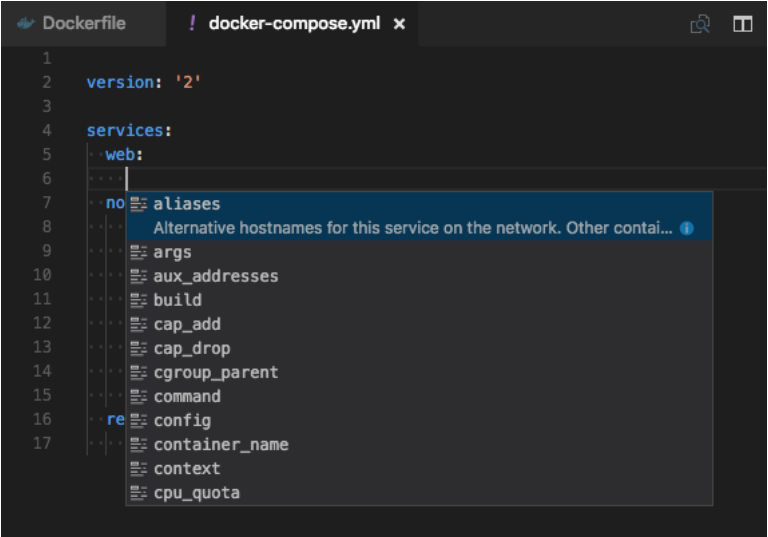
您可以使用**Docker映像**在基于Linux的独立容器中启动Azure CLI（命令行界面）：**运行Azure CLI命令**。这允许在隔离的环境中访问完整的Azure CLI命令集。有关可用命令的更多信息，请参见Azure CLI入门 (<https://docs.microsoft.com/cli/azure/get-started-with-azure-cli?view=azure-cli-latest#sign-in>)页面。

Docker撰写

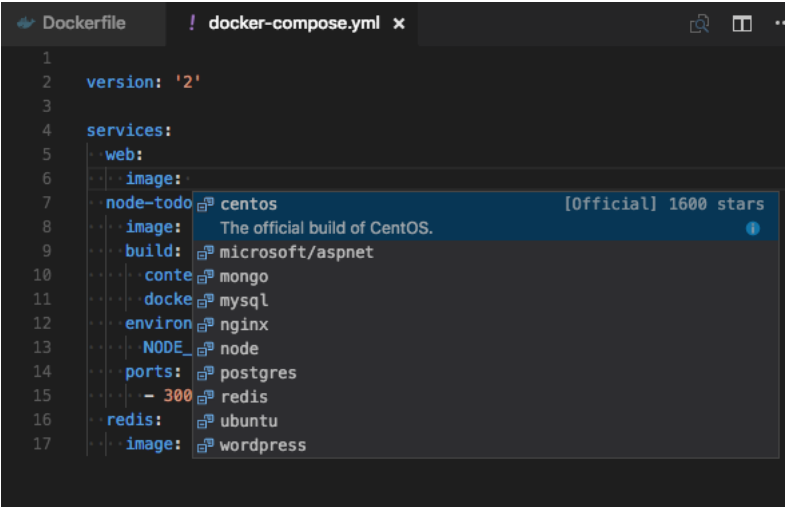
Docker Compose (<https://docs.docker.com/compose/>)允许您使用Docker定义和运行多容器应用程序。您可以使用名为的文件来定义这些容器的外观 `docker-compose.yml`。

Visual Studio Code的创作经验 `docker-compose.yml` 也非常丰富，它为有效的Docker compose指令提供了IntelliSense，它将查询Docker Hub以获取公共Docker映像上的元数据。

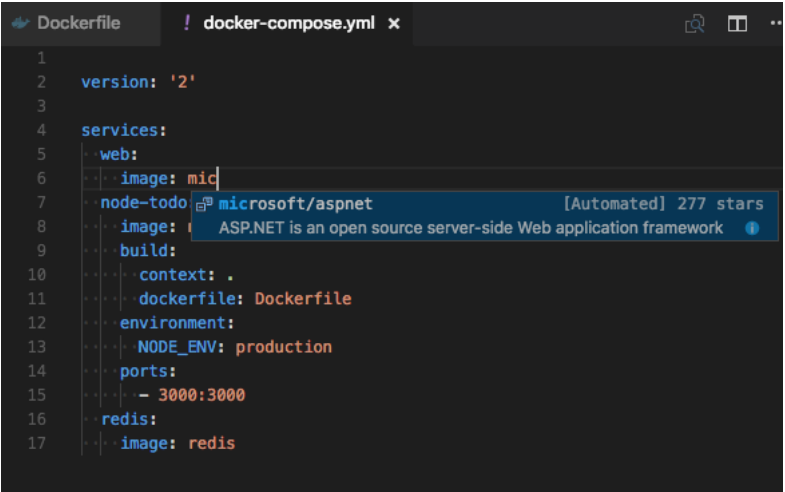
- 在工作区中创建一个新文件，名为 `docker-compose.yml`
- 定义一个名为 `web`：
- 在第二行中，通过按 `Ctrl + Space` 来打开IntelliSense，以查看所有有效compose指令的列表。



4. 对于 image 指令，您可以再次按 Ctrl + Space 键，VS Code将在Docker Hub索引中查询公共映像。



VS Code将首先显示热门图片列表以及元数据（例如星数和描述）。如果继续输入，VS Code将在Docker Hub索引中查询匹配的图像，包括搜索公共配置文件。例如，搜索“Microsoft”将显示所有公共Microsoft映像。



下一步

继续阅读以了解更多信息

- 选择您的开发环境 (/docs/containers/choosing-dev-environment)
- 在容器中构建并运行Node.js应用 (/docs/containers/quickstart-node)
- 在容器中构建并运行.NET Core应用 (/docs/containers/quickstart-aspnet-core)
- 在Docker容器中调试应用 (/docs/containers/debug-common)

该文档对您有帮助吗？

是 没有