

好的，为了确保 VS Code 能够顺畅连接，我们需要创建一个针对 **VS Code Remote-SSH** 优化的 Dockerfile。

基于之前的经验，这个 Dockerfile 核心解决了以下痛点：

1. **屏蔽欢迎信息 (MOTD)**：防止 VS Code 握手超时。
2. **兼容 Fish Shell**：虽然安装了 Fish，但 SSH 默认 Shell 保持为 Bash，只在交互式登录时切换，确保 VS Code 兼容性。
3. **预装依赖**：安装 VS Code Server 运行所需的 `curl`、`git` 等基础工具。

## 推荐的 Dockerfile

将以下内容保存为 `Dockerfile`：

```
FROM ubuntu:22.04

# 1. 避免安装过程中的交互提示
ENV DEBIAN_FRONTEND=noninteractive

# 2. 安装基础工具、SSH 服务和 Fish Shell
# libatomic1 是 VS Code Server 有时需要的依赖
RUN apt-get update && apt-get install -y \
    openssh-server \
    curl \
    wget \
    git \
    vim \
    fish \
    locales \
    libatomic1 \
    && apt-get clean && rm -rf /var/lib/apt/lists/*

# 3. 配置 SSH 服务
RUN mkdir /var/run/sshd
# 设置 root 密码（请根据需要修改）
RUN echo 'root:my_secure_password' | chpasswd
# 允许 root 远程登录
RUN echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config
# 禁止 SSH DNS 解析，加快连接速度
RUN echo 'UseDNS no' >> /etc/ssh/sshd_config

# --- 4. 关键优化：解决 VS Code 连接超时问题 ---
# 创建 .hushlogin 文件，禁止 Ubuntu 的欢迎信息 (MOTD)
```

```

RUN touch /root/.hushlogin
# 禁止显示 "Last login"
RUN sed -i 's/^PrintLastLog yes/PrintLastLog no/' /etc/ssh/sshd_config
# 修改 PAM 配置，彻底屏蔽 motd 模块
RUN sed -i 's/^\s*session\s\+optional\s\+pam_motd.so/# &/' /etc/pam.d/ssh

# --- 5. 配置 Fish Shell (智能切换) ---
# 注意：我们不使用 chsh 修改 root 的默认 shell，因为这可能破坏 VS Code 的启动脚本。
# 我们通过修改 .bashrc，仅在"交互式登录"且"父进程不是 VS Code"时进入 fish。
RUN echo 'if [[ $- == *i* ]] && [[ "$TERM_PROGRAM" != "vscode" ]]; then e

# 配置 Locale (防止终端乱码)
RUN locale-gen en_US.UTF-8
ENV LANG en_US.UTF-8
ENV LANGUAGE en_US:en
ENV LC_ALL en_US.UTF-8

# 6. 配置工作目录
WORKDIR /workspace

EXPOSE 22

# 7. 启动 SSH 服务
CMD ["/usr/sbin/sshd", "-D"]

```

## 如何构建和使用

### 1. 构建镜像

```

# 如果使用 Docker
docker build -t vscode-dev-env .

# 如果使用 Podman
podman build -t vscode-dev-env .

```

### 2. 运行容器

启动容器并将 SSH 端口映射到宿主机的 2222 端口：

```

# Docker
docker run -d -p 2222:22 --name my-dev-container vscode-dev-env

```

```
# Podman
podman run -d -p 2222:22 --name my-dev-container vscode-dev-env
```

### 3. 清理旧指纹(重要!)

如果你之前用过 2222 端口，必须先清除旧的指纹，否则连接会报错。

在宿主机终端执行：

```
ssh-keygen -R [127.0.0.1]:2222
```

### 4. VS Code 连接配置

在 VS Code 中，打开 SSH 配置文件( ~/.ssh/config 或 C:\Users\...\ssh\config )，添加：

```
Host my-dev-container
    HostName 127.0.0.1
    User root
    Port 2222
    # 防止因为容器重建导致的 Host Key 报错
    StrictHostKeyChecking no
    UserKnownHostsFile /dev/null
```

然后点击 VS Code 左下角 -> Connect to Host -> 选择 my-dev-container -> 输入密码 my\_secure\_password。

## 为什么这样设计？

- **智能 Shell 切换**：Docker 文件里有一句 `if [[ ... ]] ... exec fish`。这意味着：
  - 当 VS Code 偷偷在后台连接时，它会看到标准的 Bash 环境，连接**不会超时**。
  - 当你手动打开 VS Code 的**集成终端**，或者用终端手动 SSH 进去时，它会自动帮你切换到 **Fish**，让你享受 Fish 的便利。这是两全其美的方案。
- **.hushlogin**：彻底消除了你之前遇到的“Welcome to Ubuntu”信息，保证 SSH 通道纯净。