2025FDCS部署指南

张威年 2025-06-12

目录

[一. 准备工作 2](#_Toc200652813)

[1. 2025FDCS源码获取 2](#_Toc200652814)

[2. 安装WSL 2](#_Toc200652815)

[3. 编辑器、编译器与编译环境 2](#_Toc200652816)

[4. 其他必要工具 2](#_Toc200652817)

[二. 在WSL2上部署 3](#_Toc200652818)

[1. Ubuntu换源 3](#_Toc200652819)

[2. pip换源 4](#_Toc200652820)

[3. 安装项目依赖 5](#_Toc200652821)

[4. 本地运行 5](#_Toc200652822)

[三. SSH连接远程服务器 8](#_Toc200652823)

[1. 安装SSH 8](#_Toc200652824)

[2. 通过SSH连接 8](#_Toc200652825)

[3. SSH config 8](#_Toc200652826)

## 准备工作

### 2025FDCS源码获取

在终端中进入存放代码的目录，执行如下命令：

* git clone https://github.com/ZhangWeinian/2025FDCS.git

此仓库开源，也可以复刻到自己的GitHub中，对代码的修改保持独立。

### 安装WSL

如需在Windows环境中本地调试代码，推荐使用WSL2+Ubuntu。关于WSL2的信息，请参考[Windows Subsystem for Linux 文档](https://learn.microsoft.com/zh-cn/windows/wsl/)；关于如何安装WSL2，请参考[如何使用 WSL 在 Windows 上安装 Linux](https://learn.microsoft.com/zh-cn/windows/wsl/install)。

### 编辑器、编译器与编译环境

无论是正式Ubuntu系统，还是WSL2，都推荐使用最新稳定版的[Python](https://www.python.org/downloads/)。可以使用任意文本编辑器修改代码，推荐使用[VSCode](https://code.visualstudio.com/)。VSCode既有Windows版本，也有Linux版本，还支持WSL（需安装相应插件），极为便利！

### 其他必要工具

一些基础的编程相关工具，git、ssh、gpg等等。

## 在WSL2上部署

为了确保项目在服务器正确工作，可以现在本地部署测试。考虑到大多数人使用Windows系统，推荐使用WSL2部署和测试。

先决条件：

* 已克隆代码在 ~/source/2025FDCS中；
* 已安装Python；
* 有合适的代码编辑器（如VSCode）。

下面开始操作：

### Ubuntu换源

默认Ubuntu源国内访问较慢，可以全局指定使用国内镜像，如[清华大学开源软件镜像站Ubuntu软件仓库](https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/)。

在命令行执行：

* sudo vim /etc/apt/sources.list.d/ubuntu.sources

删除其中所有原先的内容，如果有。删除前可以备份一份以防意外。拷贝下面的内容到其中：

Types: deb

URIs: https://mirrors.tuna.tsinghua.edu.cn/ubuntu

Suites: noble noble-updates noble-backports

Components: main restricted universe multiverse

Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释

# Types: deb-src

# URIs: https://mirrors.tuna.tsinghua.edu.cn/ubuntu

# Suites: noble noble-updates noble-backports

# Components: main restricted universe multiverse

# Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

# 以下安全更新软件源包含了官方源与镜像站配置，如有需要可自行修改注释切换

Types: deb

URIs: http://security.ubuntu.com/ubuntu/

Suites: noble-security

Components: main restricted universe multiverse

Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

# Types: deb-src

# URIs: http://security.ubuntu.com/ubuntu/

# Suites: noble-security

# Components: main restricted universe multiverse

# Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

# 预发布软件源，不建议启用

# Types: deb

# URIs: https://mirrors.tuna.tsinghua.edu.cn/ubuntu

# Suites: noble-proposed

# Components: main restricted universe multiverse

# Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

# # Types: deb-src

# # URIs: https://mirrors.tuna.tsinghua.edu.cn/ubuntu

# # Suites: noble-proposed

# # Components: main restricted universe multiverse

# # Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

保存即可。

### pip换源

默认pip源国内访问较慢，可以全局指定使用国内镜像，如[清华大学开源软件镜像站PyPI软件仓库](https://mirrors.tuna.tsinghua.edu.cn/help/pypi/)。注意，可能需要把命令中的python替换为python3。

在命令行执行：

* python -m pip install -i https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple --upgrade pip
* pip config set global.index-url https://mirrors.tuna.tsinghua.edu.cn/pypi/web/simple

### 安装项目依赖

进入Ubuntu控制台，按序执行以下操作。

1. 更新环境，在终端执行：

* sudo apt update

1. 安装MySQL，在终端执行：

* sudo apt install mysql-server

1. 安装nginx，在终端执行：

* sudo apt install nginx

1. 安装编译依赖：

* sudo apt install pkg-config libmysqlclient-dev build-essential python3-dev

1. 安装Python依赖（对于Ubuntu，推荐在[Python虚拟环境](https://docs.python.org/zh-cn/3/library/zipapp.html)中安装），激活Python虚拟环境并进入2025FDCS目录，然后在终端执行：

* pip install -r requirements.txt

1. 安装git，在终端执行：

* sudo apt-get install git

### 本地运行

1. 启动MySQL和nginx，在命令行执行：

* sudo systemctl start mysql
* sudo systemctl start nginx

1. MySQL配置
2. 设置root密码，移除匿名用户，禁止root远程访问等，在终端执行：
   * sudo mysql\_secure\_installation
3. 检查MySQL服务的状态，确保它正在运行：
   * sudo systemctl status mysql.service
4. 登录MySQL：
   * mysql -u root -p
5. 创建新用户并授权（也可以直接使用root用户）：
   * CREATE USER '用户名'@'localhost' IDENTIFIED BY '密码';
   * GRANT ALL PRIVILEGES ON \*.\* TO '用户名'@'localhost' WITH GRANT OPTION;
   * FLUSH PRIVILEGES;
6. 创建数据库（数据库名为FDCS），并设置UTF8编码：
   * CREATE DATABASE FDCS CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;
7. 修改代码中的设置

修改FDCS/setting.py文件的数据库配置，把 "USER" 和 "PASSWORD" 字段填入之前设置的'用户名'和'密码'：

文本

AI 生成的内容可能不正确。

1. 数据库表迁移，在命令行执行：

* python3 manage.py migrate

1. 静态文件收集，在命令行执行：

* python3 manage.py collectstatic

1. 启动项目，在命令行执行：

* python3 manage.py runserver 0.0.0.0:8080

在浏览器地址栏输入localhost:8080/login即可访问。

## SSH连接远程服务器

可以通过SSH连接本地Windows与远程Ubuntu服务器。

### 安装SSH

Windows中的SSH工具请参考[OpenSSH for Windows概述](https://learn.microsoft.com/zh-cn/windows-server/administration/openssh/openssh-overview)和[适用于Windows的OpenSSH入门](https://learn.microsoft.com/zh-cn/windows-server/administration/openssh/openssh_install_firstuse?tabs=powershell&pivots=windows-server-2022)。通过这两个文档，可以在Windows上正确安装OpenSSH for Windows，可以快一步了解[教程：Windows终端中的SSH](https://learn.microsoft.com/zh-cn/windows/terminal/tutorials/ssh?source=recommendations)从而在[Windows Terminal](https://learn.microsoft.com/zh-cn/windows/terminal/)快速启动指定的连接！

在命令行执行：

* ssh -V

以验证SSH工具是否正确安装。

### 通过SSH连接

先决条件：

* 拥有访问服务器的IP；
* SSH登录用户名；
* 身份验证方式：可能是创建服务器时设置的登录密码或密钥对的私钥。

1. 使用登陆密码连接

在命令行执行：

* ssh <用户名>@<IP>

按提示选择是否接受信任。若接受，输入登陆密码即可连接。

1. 使用密钥对登录

登陆时同时指定私钥文件，在命令行执行：

* ssh -i "<私钥文件路径>" <用户名>@<IP>

如果私钥设置了密码，输入该密码。如果没有，则直接登录成功。

### SSH config

通过配置SSH config文件，可以简化SSH连接流程。例如

Host \*

    AddKeysToAgent no

    ConnectTimeout 30

    ForwardAgent yes

    ServerAliveInterval 45

    ServerAliveCountMax 3

    PreferredAuthentications publickey

Host <登录别名>

    HostName <IP>

    User <用户名>

    # Port 22

    IdentityFile "<公钥文件路径>"

    IdentitiesOnly yes

则在命令行执行：

* ssh <登录别名>

即可连接，无需每次输入用户名和IP。