低延迟部署方案：直接运行Node.js服务器

为了获得最低延迟，我们将跳过Nginx反向代理，直接部署Node.js服务。这样可以减少数据传输路径中的中间环节。

#### 第一步：准备云服务器

购买云服务器：

建议选择离你物理位置较近的服务器区域减少网络延迟

配置：至少1核2G内存

带宽：建议10Mbps以上，带宽越高延迟越低

开放端口：

在云服务器控制台开放3000端口（用于WebSocket和HTTP服务）

开放22端口（SSH访问）

#### 第二步：连接服务器并安装环境

# SSH连接到服务器

ssh 用户名@服务器IP

# 更新系统

sudo apt update && sudo apt upgrade -y

# 安装Node.js和npm

curl -fsSL https://deb.nodesource.com/setup\_16.x | sudo -E bash -

sudo apt install -y nodejs

# 检查安装

node -v

npm -v

# 安装Git(用于拉取代码)

sudo apt install git -y

#### 第三步：上传代码到服务器

# 创建目录

mkdir -p /home/用户名/cursor-ESP8266

# 方法一：使用Git拉取

git clone 仓库地址 /home/用户名/cursor-ESP8266

# 方法二：直接SCP上传(在本地执行)

scp -r /本地路径/cursor-ESP8266/\* 用户名@服务器IP:/home/用户名/cursor-ESP8266/

#### 第四步：安装项目依赖

cd /home/用户名/cursor-ESP8266

npm install

#### 第五步：配置服务器地址

修改服务器代码以确保它直接监听3000端口并对外部开放：

# 打开服务器文件

nano /home/用户名/cursor-ESP8266/server/server.js

// 替换或确认监听配置

server.listen(3000, '0.0.0.0', () => {

console.log('服务器运行在 http://0.0.0.0:3000');

});

#### 第六步：启动服务器进程

为了保持服务持续运行，使用PM2：

# 安装PM2

sudo npm install pm2 -g

# 进入服务器目录

cd /home/用户名/cursor-ESP8266/server

# 启动服务

pm2 start server.js --name "cursor-ESP8266-server"

# 设置开机自启

pm2 startup

pm2 save

#### 第七步：确保3000端口已开放

# 如果使用本地防火墙

sudo ufw allow 3000/tcp

sudo ufw allow 22/tcp

sudo ufw enable

# 检查防火墙状态

sudo ufw status

#### 第八步：更新ESP8266固件

修改ESP8266代码中的服务器地址和端口：

// 将服务器地址改为你的云服务器IP，端口改为3000

const char\* serverAddress = "你的服务器IP";

const int serverPort = 3000;

重新编译并上传固件到ESP8266设备。

#### 第十步：测试连接

访问 http://服务器IP:3000 测试网页界面

打开ESP8266设备，测试是否能连接到服务器

检查控制延迟是否满意

**这种直接部署方式可以最大限度减少延迟，但请注意，它牺牲了安全性和可扩展性。**