# AI重构DevOps课程环境准备指南

## 📋 课程环境概述

本课程将通过实战演示的方式，带领学员体验AI如何重构DevOps全流程。为确保课程效果，请学员提前准备以下开发环境和工具。

## 🌐 网络环境要求

### 科学上网配置（强烈建议）

为什么需要科学上网？

* GitHub Copilot 需要稳定的海外网络连接
* 访问最新的AI模型API（Claude、OpenAI等）
* 获取国外技术文档和最佳实践资源
* 保证实时在线演示的流畅性

推荐工具：

* 🔧 Clash for Windows / V2rayN （Windows）
* 🔧 ClashX / V2rayU （macOS）
* 🔧 Qv2ray （Linux）

配置建议：

语言: bash

# 测试网络连通性

ping google.com

curl -I https://api.github.com

# 检查GitHub访问速度

git clone https://github.com/microsoft/vscode.git --depth=1

### 备选方案（无法科学上网的学员）

* 使用国内AI服务（如阿里云、腾讯云的AI接口）
* 提前下载必要的软件包和依赖
* 课程现场提供网络热点共享

## 🛠️ 核心开发工具链

### 1. Git & GitHub 配置

#### 1.1 Git 安装与配置

语言: bash

# Windows（使用Git for Windows）

https://git-scm.com/download/win

# macOS（使用Homebrew）

brew install git

# Linux（Ubuntu/Debian）

sudo apt update && sudo apt install git

# 基础配置

git config --global user.name "你的姓名"

git config --global user.email "你的邮箱@example.com"

git config --global init.defaultBranch main

#### 1.2 GitHub 账户准备

* 📝 注册GitHub账户：https://github.com
* 🔐 生成SSH密钥对：

语言: bash

ssh-keygen -t ed25519 -C "你的邮箱@example.com"

cat ~/.ssh/id\_ed25519.pub

# 将公钥添加到GitHub Settings -> SSH and GPG keys

* 🧪 测试连接：

语言: bash

ssh -T git@github.com

# 应该看到："Hi username! You've successfully authenticated..."

#### 1.3 GitHub 仓库准备

创建以下仓库作为课程练习使用：

语言: bash

# 1. 课程主仓库

git clone https://github.com/你的用户名/ai-devops-workshop.git

cd ai-devops-workshop

# 2. 初始化项目结构

mkdir -p {src,docs,tests,scripts,configs}

echo "# AI DevOps Workshop" > README.md

git add . && git commit -m "Initial project setup"

git push origin main

### 2. VS Code + GitHub Copilot 开发环境

#### 2.1 VS Code 安装

语言: bash

# 官方下载

https://code.visualstudio.com/

# Windows（Chocolatey）

choco install vscode

# macOS（Homebrew）

brew install --cask visual-studio-code

# Linux（Snap）

sudo snap install code --classic

#### 2.2 GitHub Copilot 配置

前置要求：

* ✅ 有效的GitHub账户
* 💳 GitHub Copilot订阅（个人版$10/月，学生免费）
* 🌐 稳定的国际网络连接

安装步骤：

1. 打开VS Code
2. 安装扩展：

* GitHub Copilot
* GitHub Copilot Chat

1. 登录GitHub账户：Ctrl+Shift+P → GitHub Copilot: Sign In
2. 验证安装：创建Python文件，输入注释测试自动补全

#### 2.3 必备 VS Code 扩展

语言: json

// 推荐扩展列表（.vscode/extensions.json）

{

"recommendations": [

"github.copilot",

"github.copilot-chat",

"ms-python.python",

"ms-python.pylint",

"ms-python.black-formatter",

"ms-vscode.vscode-git-graph",

"eamodio.gitlens",

"ms-vscode.vscode-json",

"redhat.vscode-yaml",

"ms-vscode-remote.remote-containers",

"ms-azuretools.vscode-docker"

]

}

#### 2.4 VS Code 配置优化

语言: json

// .vscode/settings.json

{

"python.defaultInterpreterPath": "python",

"python.linting.enabled": true,

"python.linting.pylintEnabled": true,

"python.formatting.provider": "black",

"editor.formatOnSave": true,

"editor.codeActionsOnSave": {

"source.organizeImports": true

},

"github.copilot.enable": {

"\*": true,

"yaml": true,

"plaintext": false,

"markdown": true

},

"git.autofetch": true,

"git.enableSmartCommit": true

}

## 🐍 Python 开发环境

### 3.1 Python 安装

#### Windows

语言: powershell

# 方法1：官方安装包

https://www.python.org/downloads/windows/

# 选择Python 3.9+版本，勾选"Add Python to PATH"

# 方法2：Microsoft Store

# 搜索"Python 3.11"直接安装

# 方法3：Chocolatey

choco install python --version=3.11.0

#### macOS

语言: bash

# 方法1：Homebrew（推荐）

brew install python@3.11

# 方法2：pyenv（版本管理）

brew install pyenv

pyenv install 3.11.0

pyenv global 3.11.0

#### Linux

语言: bash

# Ubuntu/Debian

sudo apt update

sudo apt install python3.11 python3.11-pip python3.11-venv

# CentOS/RHEL

sudo yum install python3.11 python3.11-pip

### 3.2 Python 环境验证

语言: bash

# 检查Python版本

python --version

# 应该显示：Python 3.9+

# 检查pip版本

pip --version

# 检查虚拟环境创建

python -m venv test\_env

source test\_env/bin/activate # Linux/macOS

# 或

test\_env\Scripts\activate # Windows

# 测试包安装

pip install requests

python -c "import requests; print('Python环境正常')"

### 3.3 课程所需Python包

语言: bash

# 创建课程虚拟环境

python -m venv ai-devops-env

# 激活环境

source ai-devops-env/bin/activate # Linux/macOS

ai-devops-env\Scripts\activate # Windows

# 安装课程依赖

pip install --upgrade pip

pip install -r requirements.txt

requirements.txt 内容：

语言: txt

# Web开发框架

fastapi==0.104.1

uvicorn==0.24.0

requests==2.31.0

# 数据处理

pandas==2.1.3

numpy==1.24.4

# 测试框架

pytest==7.4.3

pytest-asyncio==0.21.1

# 代码质量

black==23.10.1

pylint==3.0.2

mypy==1.7.0

# DevOps工具

docker==6.1.3

kubernetes==28.1.0

# AI/ML工具

openai==1.3.7

anthropic==0.7.7

# 监控工具

prometheus-client==0.19.0

## 🔧 辅助开发工具

### 4.1 Docker 环境

语言: bash

# Windows：Docker Desktop

https://www.docker.com/products/docker-desktop/

# macOS：Docker Desktop

brew install --cask docker

# Linux：Docker CE

curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

sudo usermod -aG docker $USER

# 验证安装

docker --version

docker run hello-world

### 4.2 数据库工具

语言: bash

# SQLite（轻量级，课程演示用）

# Python内置，无需额外安装

# PostgreSQL客户端（可选）

pip install psycopg2-binary

# Redis客户端（缓存演示）

pip install redis

### 4.3 API测试工具

* 📦 Postman：https://www.postman.com/downloads/
* 🔧 HTTPie：pip install httpie
* ⚡ curl：系统通常预装

## 📊 课程演示项目

### 5.1 示例项目：航空票务管理系统

语言: bash

# 克隆课程项目模板

git clone https://github.com/instructor/aviation-booking-system.git

cd aviation-booking-system

# 查看项目结构

tree .

#

# ├── src/

# │ ├── api/ # FastAPI接口

# │ ├── models/ # 数据模型

# │ ├── services/ # 业务逻辑

# │ └── utils/ # 工具函数

# ├── tests/ # 测试用例

# ├── docs/ # 项目文档

# ├── scripts/ # 部署脚本

# ├── .github/ # CI/CD配置

# │ └── workflows/

# ├── docker-compose.yml

# ├── Dockerfile

# ├── requirements.txt

# └── README.md

### 5.2 本地开发环境启动

语言: bash

# 安装依赖

pip install -r requirements.txt

# 启动开发服务器

uvicorn src.main:app --reload --host 0.0.0.0 --port 8000

# 验证服务

curl http://localhost:8000/health

# 返回：{"status": "healthy", "timestamp": "..."}

# 访问API文档

open http://localhost:8000/docs

## ✅ 环境检验清单

### 课程前自检项目

请在课程开始前完成以下检验：

#### 基础环境检查

* [ ] 🌐 能够访问GitHub且网络稳定
* [ ] 📁 Git配置正确，可以push/pull代码
* [ ] 💻 VS Code正常运行
* [ ] 🤖 GitHub Copilot可以正常提供代码建议
* [ ] 🐍 Python 3.9+环境可用
* [ ] 📦 能够安装Python包

#### 功能验证

* [ ] 🔧 创建并激活Python虚拟环境
* [ ] 🚀 FastAPI示例项目能够启动
* [ ] 🧪 pytest测试命令可以执行
* [ ] 🐳 Docker可以运行容器（如果已安装）
* [ ] 📊 可以访问GitHub仓库并创建Issue/PR

#### 课程项目准备

* [ ] 📂 已fork课程示例项目到个人GitHub
* [ ] 🔄 本地可以运行航空票务系统demo
* [ ] 📝 VS Code中Copilot Chat能够响应问题
* [ ] 🎯 已阅读课程大纲，了解学习目标

## 🆘 常见问题解决

### Q1: GitHub Copilot无法激活

问题现象：VS Code中看不到Copilot建议

解决方案：

语言: bash

# 1. 检查订阅状态

https://github.com/settings/copilot

# 2. 重新登录

Ctrl+Shift+P -> "GitHub Copilot: Sign Out"

Ctrl+Shift+P -> "GitHub Copilot: Sign In"

# 3. 重启VS Code

### Q2: Python包安装失败

问题现象：pip install报错或超时

解决方案：

语言: bash

# 使用国内镜像源

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ package-name

# 永久配置

pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple/

### Q3: Git push失败

问题现象：Permission denied或Authentication failed

解决方案：

语言: bash

# 检查SSH配置

ssh -T git@github.com

# 重新配置SSH密钥

ssh-keygen -t ed25519 -C "your-email@example.com"

# 将~/.ssh/id\_ed25519.pub内容添加到GitHub

### Q4: 网络连接问题

问题现象：无法访问GitHub或AI服务

解决方案：

* 确认科学上网工具正常工作
* 检查代理设置：

语言: bash

git config --global http.proxy http://127.0.0.1:7890

git config --global https.proxy http://127.0.0.1:7890

## 📞 技术支持

### 课前支持渠道

* 📧 邮件支持：course-support@example.com
* 💬 QQ群：123456789（AI-DevOps学习群）
* 📱 微信群：扫描二维码加入
* 🎥 视频教程：环境配置演示视频

### 应急预案

如果个别学员环境有问题，课程现场提供：

* 🔄 Cloud IDE：GitHub Codespaces备用环境
* 💻 虚拟机镜像：预配置完整环境
* 🤝 结对编程：与环境正常的学员组队

---

## 🎯 课程预期效果

完成环境准备后，您将能够：

* ⚡ 快速体验AI辅助编程的强大能力
* 🔄 实践完整的DevOps工作流程
* 🛠️ 掌握现代开发工具链的使用方法
* 📈 构建个人AI+DevOps技能体系

记住：良好的环境准备是成功学习的第一步！

最后更新：2025年8月17日

版本：v1.0