

PowerAutomation 部署资源汇总



主要部署链接

GitHub 仓库

官方仓库: <https://github.com/alexchuang650730/aicore0615>

快速部署命令

一键部署 (推荐)

```
# 下载并运行快速部署脚本
curl -fsSL https://raw.githubusercontent.com/alexchuang650730/aicore0615/main/quick_deploy.sh | bash
```

Docker部署

```
# 下载并运行Docker部署脚本
curl -fsSL https://raw.githubusercontent.com/alexchuang650730/aicore0615/main/docker_deploy.sh | bash
```

手动部署

```
# 克隆仓库
git clone https://github.com/alexchuang650730/aicore0615.git
cd aicore0615

# 安装依赖
pip install -r requirements.txt

# 启动系统
cd test && python main.py
```

部署资源清单

核心组件

组件	GitHub链接	说明
SmartUI MCP	链接	智能用户界面
Enhanced Workflow MCP	链接	增强工作流
测试框架	链接	完整测试系统
MCP协调器	链接	中央协调器

配置文件

文件	路径	用途
测试配置	/test/config/test_config.yaml	测试框架配置
调度配置	/test/config/schedule_config.yaml	定时任务配置
MCP配置	/mcp/adapter/*/config.toml	各MCP组件配置

部署脚本

脚本	文件名	用途
快速部署	quick_deploy.sh	一键部署到生产环境
Docker部署	docker_deploy.sh	容器化部署
开发环境	dev-setup.sh	开发环境搭建

系统要求

最低要求

- **操作系统:** Ubuntu 20.04+ / CentOS 8+ / macOS 12+
- **Python:** 3.11.0+
- **内存:** 4GB RAM
- **磁盘:** 10GB 可用空间

- **网络:** 稳定的互联网连接

推荐配置

- **操作系统:** Ubuntu 22.04 LTS
- **Python:** 3.11.0+
- **内存:** 8GB+ RAM
- **磁盘:** 50GB+ SSD
- **CPU:** 4核心+

快速开始

方法1: 一键部署（最简单）

```
# 下载并执行部署脚本
curl -fsSL https://raw.githubusercontent.com/alexchuang650730/aicore0615/main/quick_deploy.sh | bash

# 访问应用
open http://localhost
```

方法2: Docker部署（推荐）

```
# 下载并执行Docker部署脚本
curl -fsSL https://raw.githubusercontent.com/alexchuang650730/aicore0615/main/docker_deploy.sh | bash

# 检查状态
docker-compose ps

# 访问应用
open http://localhost
```

方法3: 手动部署（开发者）

```
# 1. 克隆代码
git clone https://github.com/alexchuang650730/aicore0615.git
cd aicore0615

# 2. 安装依赖
pip install -r requirements.txt

# 3. 启动测试框架
```

```
cd test
python main.py

# 4. 启动SmartUI MCP
cd ../mcp/adapter/smartui_mcp
python cli.py start

# 5. 验证部署
python cli.py status
```

配置管理

环境变量

```
export POWERAUTOMATION_ENV=production
export DATABASE_URL=postgresql://user:pass@localhost:5432/
powerautomation
export REDIS_URL=redis://localhost:6379/0
export SECRET_KEY=your-secret-key-here
```

配置文件示例

```
# config/production.yaml
database:
  url: ${DATABASE_URL}
  pool_size: 20

cache:
  redis_url: ${REDIS_URL}
  default_timeout: 3600

security:
  secret_key: ${SECRET_KEY}
  jwt_expiry: 3600
```

验证部署

健康检查

```
# 检查系统健康
curl http://localhost/health

# 检查组件状态
```

```
curl http://localhost/api/v1/status

# 检查SmartUI MCP
cd mcp/adapter/smartui_mcp && python cli.py status

# 检查测试框架
cd test && python cli.py status
```

功能测试

```
# 运行完整测试
cd test && python cli.py run --type comprehensive

# 运行演示脚本
python powerautomation_demo.py

# 检查测试报告
ls test/reports/
```

容器化部署

Docker Compose

```
version: '3.8'
services:
  powerautomation:
    image: powerautomation/powerautomation:latest
    ports:
      - "8000:8000"
    environment:
      - DATABASE_URL=postgresql://user:pass@db:5432/
powerautomation
      - REDIS_URL=redis://redis:6379/0
    depends_on:
      - db
      - redis
```

Kubernetes

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: powerautomation
spec:
```

```
replicas: 3
selector:
  matchLabels:
    app: powerautomation
template:
  spec:
    containers:
      - name: powerautomation
        image: powerautomation/powerautomation:latest
        ports:
          - containerPort: 8000
```

安全配置

SSL/TLS

```
# 安装SSL证书
sudo certbot --nginx -d your-domain.com

# 配置HTTPS重定向
sudo nano /etc/nginx/sites-available/powerautomation
```

防火墙

```
# 配置UFW防火墙
sudo ufw allow 22/tcp    # SSH
sudo ufw allow 80/tcp    # HTTP
sudo ufw allow 443/tcp   # HTTPS
sudo ufw enable
```

监控和日志

监控配置

```
# monitoring/prometheus.yml
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'powerautomation'
    static_configs:
      - targets: ['localhost:8000']
```

日志管理

```
# 查看应用日志
tail -f /var/log/powerautomation/app.log

# 查看系统日志
sudo journalctl -u powerautomation -f

# 查看Docker日志
docker-compose logs -f powerautomation
```

故障排除

常见问题

```
# 端口占用
sudo lsof -i :8000
sudo kill -9 <PID>

# 权限问题
sudo chown -R $USER:$USER /opt/powerautomation
chmod +x scripts/*.sh

# 依赖问题
pip install --upgrade pip
pip install -r requirements.txt --force-reinstall
```

诊断脚本

```
# 系统诊断
./scripts/diagnose.sh

# 性能检查
./scripts/performance_check.sh

# 网络测试
./scripts/network_test.sh
```

支持资源

官方资源

- **GitHub:** <https://github.com/alexchuang650730/aicore0615>
- **Issues:** <https://github.com/alexchuang650730/aicore0615/issues>
- **Wiki:** <https://github.com/alexchuang650730/aicore0615/wiki>
- **Releases:** <https://github.com/alexchuang650730/aicore0615/releases>

社区支持







- **讨论区:** <https://github.com/alexchuang650730/aicore0615/discussions>
- **Stack Overflow:** 标签 `powerautomation`
- **Reddit:** [r/PowerAutomation](https://www.reddit.com/r/PowerAutomation)

商业支持

- **技术支持:** support@powerautomation.io
- **企业服务:** enterprise@powerautomation.io
- **培训服务:** training@powerautomation.io

更新日志

v1.0.0 (2025-06-17)

-  完整的SmartUI MCP实现
-  Enhanced Workflow MCP功能
-  端到端测试框架
-  一键部署脚本
-  Docker容器化支持
-  完整的文档和指南

最后更新: 2025年6月17日

版本: 1.0.0

维护者: PowerAutomation团队