

حزمة التنصيب الشاملة - منصة سهول



نظرة عامة

هذا الدليل يحتوي على جميع الأكواد والскريبتات المطلوبة لتنصيب منصة سهول على سيرفر محلي . (HCI)

المحتويات

1. [스크립ت التنصيب التلقائي](#)
 2. [ملفات الإعداد](#)
 3. [스크립تات الصيانة](#)
 4. [دليل التشغيل](#)
-

스크립ت التنصيب التلقائي

1. سكريبت التنصيب الرئيسي

احفظ هذا الملف باسم `install.sh`

```

#!/bin/bash

=====
# منصة سهول - سكريبت التنصيب الشامل
# الإصدار: 2.2.0
# التاريخ: 03-11-2025
=====

set -e # إيقاف عند أي خطأ

# الألوان للطباعة
RED='\033[0;31m'
GREEN='\033[0;32m'
YELLOW='\033[1;33m'
BLUE='\033[0;34m'
NC='\033[0m' # No Color

# دالة الطباعة الملونة
print_info() {
    echo -e "${BLUE}[INFO]${NC} $1"
}

print_success() {
    echo -e "${GREEN}[SUCCESS]${NC} $1"
}

print_warning() {
    echo -e "${YELLOW}[WARNING]${NC} $1"
}

print_error() {
    echo -e "${RED}[ERROR]${NC} $1"
}

# التحقق من صلاحيات المستخدم
check_root() {
    if [[ $EUID -ne 0 ]]; then
        print_error يجب تشغيل هذا السكريبت بـ${color}root${color}
        print_info استخدم: sudo bash install.sh
        exit 1
    fi
}

# رسالة الترحيب
welcome_message() {

```

```

clear
echo " منصة شهول - سكريبت التنصيب التلقائي "
echo " الإصدار 2.2.0 "
echo ""
echo " منصة شهول"
echo ""
echo " سيتم تنصيب المكونات التالية:"
echo " - Node.js 22.x"
echo " - MySQL 8.x"
echo " - Nginx"
echo " - Redis"
echo " - PM2"
echo " - منصة شهول"
echo ""
read -p "هل ت يريد المتابعة؟ (y/n): " -n 1 -r
echo
if [[ ! $REPLY =~ ^[Yy]$ ]]; then
    print_warning "تم إلغاء التنصيب"
    exit 0
fi
}

# تحديث النظام
update_system() {
    print_info "...تحديث النظام"
    apt update && apt upgrade -y
    apt install -y curl wget git build-essential ufw
    print_success "تم تحديث النظام بنجاح"
}

# تثبيت Node.js
install_nodejs() {
    print_info "تثبيت Node.js 22.x..."

    if command -v node >& /dev/null; then
        NODE_VERSION=$(node --version)
        print_warning "Node.js موجود بالفعل: $NODE_VERSION"
        read -p "هل تريد إعادة التثبيت؟ (y/n): " -n 1 -r
        echo
        if [[ ! $REPLY =~ ^[Yy]$ ]]; then
            return
        fi
    fi
}

```

```

curl -fsSL https://deb.nodesource.com/setup_22.x | bash -
apt install -y nodejs

# تثبيت pnpm
npm install -g pnpm

print_success "تم تثبيت Node.js $(node --version)"
print_success "تم تثبيت pnpm $(pnpm --version)"
}

# تثبيت MySQL
install_mysql() {
    print_info "تم تثبيت MySQL 8.x..."

    if command -v mysql &> /dev/null; then
        print_warning "MySQL موجود بالفعل"
        return
    fi

    apt install -y mysql-server
    systemctl start mysql
    systemctl enable mysql

    print_success "نجاح تثبيت MySQL"
}

إعداد قاعدة البيانات # إعداد قاعدة البيانات
print_info "... إعداد قاعدة البيانات"

read -p "اتركها فارغة للتنطيط) : " MYSQL_ROOT_PASS

if [ -n "$MYSQL_ROOT_PASS" ]; then
    mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY '$MYSQL_ROOT_PASS';"
fi

read -p "أدخل كلمة مرور قاعدة بيانات sahool: " SAHOOOL_DB_PASS

mysql -e "CREATE DATABASE IF NOT EXISTS sahool CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;"
mysql -e "CREATE USER IF NOT EXISTS 'sahool_user'@'localhost'
IDENTIFIED BY '$SAHOOOL_DB_PASS';"
mysql -e "GRANT ALL PRIVILEGES ON sahool.* TO
'sahool_user'@'localhost';"
mysql -e "FLUSH PRIVILEGES;"

# حفظ كلمة المرور للاستخدام لاحقاً

```

```

echo "$SAHOOOL_DB_PASS" > /tmp/sahool_db_pass

print_success "تم إعداد قاعدة البيانات بنجاح"
}

# تثبيت Redis
install_redis() {
print_info "تثبيت Redis..."

apt install -y redis-server
systemctl start redis-server
systemctl enable redis-server

print_success "Redis تم تثبيت بنجاح"
}

# تثبيت Nginx
install_nginx() {
print_info "تثبيت Nginx..."

apt install -y nginx
systemctl start nginx
systemctl enable nginx

print_success "Nginx تم تثبيت بنجاح"
}

# تثبيت PM2
install_pm2() {
print_info "تثبيت PM2..."

npm install -g pm2

print_success "PM2 تم تثبيت بنجاح"
}

# إعداد Firewall
setup_firewall() {
print_info "إعداد Firewall..."

ufw allow 22/tcp    # SSH
ufw allow 80/tcp    # HTTP
ufw allow 443/tcp   # HTTPS
ufw allow 3000/tcp  # Application

echo "y" | ufw enable
}

```

```

        print_success "بنجاح تم إعداد Firewall"
    }

# تحميل المشروع
download_project() {
    print_info "تحميل منصة سهول..."
    APP_DIR="/opt/sahool-platform"

    if [ -d "$APP_DIR" ]; then
        print_warning "المشروع موجود بالفعل في $APP_DIR"
        read -p "هل تريد حذفه وإعادة التحميل؟ (y/n): " -n 1 -r
        echo
        if [[ $REPLY =~ ^[Yy]$ ]]; then
            rm -rf "$APP_DIR"
        else
            return
        fi
    fi

    mkdir -p "$APP_DIR"
    cd "$APP_DIR"

    git clone https://github.com/kafaat/sahool-platform.git .
    print_success "تم تحميل المشروع بنجاح"
}

# إعداد ملف البيئة
setup_env() {
    print_info "إعداد ملف البيئة..."
    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    قراءة كلمة مرور قاعدة البيانات #
    if [ -f /tmp/sahool_db_pass ]; then
        DB_PASS=$(cat /tmp/sahool_db_pass)
        rm /tmp/sahool_db_pass
    else
        read -p "أدخل كلمة مرور قاعدة البيانات": DB_PASS
    fi

    # عشوائي توقيت JWT Secret
    JWT_SECRET=$(openssl rand -base64 32)
}

```

```

# إنشاء ملف .env
cat > .env << EOF

# Database
DATABASE_URL=mysql://sahool_user:${DB_PASS}@localhost:3306/sahool

# Security
JWT_SECRET=${JWT_SECRET}

# Application
NODE_ENV=production
PORT=3000

# Owner Info
OWNER_OPEN_ID=admin
OWNER_NAME=مدير النظام

# App Settings
VITE_APP_TITLE=سهوول
VITE_APP_LOGO=/logo.png

# Redis
REDIS_URL=redis://localhost:6379

# APIs (يمكن إضافتها لاحقاً)
OPENWEATHER_API_KEY=
IQAIR_API_KEY=
NREL_API_KEY=
SENTINEL_HUB_CLIENT_ID=
SENTINEL_HUB_CLIENT_SECRET=
EOF

    print_success "بنجاح . تم إنشاء ملف env."
}

# تثبيت حزم المشروع
install_dependencies() {
    print_info "تثبيت حزم المشروع... (قد يستغرق 10-5 دقائق)"
    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm install --frozen-lockfile

    print_success "تم تثبيت الحزم بنجاح"
}

```

```
# بناء المشروع
build_project() {
    print_info "...بناء المشروع"
    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm run build

    print_success "تم بناء المشروع بنجاح"
}

# إعداد قاعدة البيانات
setup_database() {
    print_info "...إعداد جداول قاعدة البيانات"
    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    pnpm db:push

    print_success "تم إعداد قاعدة البيانات بنجاح"
}

# إعداد PM2
setup_pm2() {
    print_info "...إعداد PM2..."
    APP_DIR="/opt/sahool-platform"
    cd "$APP_DIR"

    # إيقاف التطبيق إذا كان يعمل
    pm2 delete sahool-platform 2>/dev/null || true

    # تشغيل التطبيق
    pm2 start dist/index.js --name sahool-platform
    pm2 save
    pm2 startup systemd -u root --hp /root

    print_success "تم إعداد PM2 بنجاح"
}

# إعداد Nginx
setup_nginx() {
    print_info "...إعداد Nginx..."
}
```

```

# السيرفر IP الحصول على
SERVER_IP=$(hostname -I | awk '{print $1}')

read -p "ادخل اسم النطاق أو المخطىء) Enter IP للاستخدام: $SERVER_IP": " DOMAIN

if [ -z "$DOMAIN" ]; then
    DOMAIN=$SERVER_IP
fi

# إعداد ملف إنشاء Nginx
cat > /etc/nginx/sites-available/sahool << EOF
upstream sahool_backend {
    server 127.0.0.1:3000;
    keepalive 64;
}

server {
    listen 80;
    server_name ${DOMAIN};

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Logging
    access_log /var/log/nginx/sahool-access.log;
    error_log /var/log/nginx/sahool-error.log;

    # Client Body Size
    client_max_body_size 50M;

    # Proxy Settings
    location / {
        proxy_pass http://sahool_backend;
        proxy_http_version 1.1;

        proxy_set_header Upgrade \$http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host \$host;
        proxy_set_header X-Real-IP \$remote_addr;
        proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto \$scheme;

        proxy_cache_bypass \$http_upgrade;
    }
}
EOF

```

```

        proxy_buffering off;
    }

# Static Files
location /assets {
    alias /opt/sahool-platform/dist/public/assets;
    expires 1y;
    add_header Cache-Control "public, immutable";
}
}

EOF

# تفعيل الاعداد
ln -sf /etc/nginx/sites-available/sahool /etc/nginx/sites-enabled/
rm -f /etc/nginx/sites-enabled/default

# اختبار وإعادة تشغيل Nginx
nginx -t
systemctl restart nginx

print_success "نجاح إعداد Nginx تم"

# حفظ النطاق لل استخدام لاحقاً
echo "$DOMAIN" > /tmp/sahool_domain
}

# إعداد SSL ( اختياري )
setup_ssl() {
    print_info "إعداد SSL/TLS..."

    read -p "هل تريد تثبيت شهادة SSL؟ (y/n): " -n 1 -r
    echo

    if [[ ! $REPLY =~ ^[Yy]$ ]]; then
        return
    fi

    DOMAIN=$(cat /tmp/sahool_domain 2>/dev/null || echo "")

    if [ -z "$DOMAIN" ]; then
        read -p "أدخل اسم النطاق: " DOMAIN
    fi

    # التحقق من نوع الشهادة
    echo "اختر نوع الشهادة"
    echo "1) Let's Encrypt (النطاقات العامة)"

```

```

echo "شهادة ذاتية التوقيع (الشبكة المحلية) (2"
read -p "(1/2): " SSL_CHOICE

if [ "$SSL_CHOICE" == "1" ]; then
    # Let's Encrypt
    apt install -y certbot python3-certbot-nginx
    certbot --nginx -d "$DOMAIN" --non-interactive --agree-tos --
register-unsafe-without-email
else
    # شهادة ذاتية
    mkdir -p /etc/nginx/ssl
    openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
        -keyout /etc/nginx/ssl/sahoool.key \
        -out /etc/nginx/ssl/sahoool.crt \
        -subj "/C=SA/ST=Riyadh/L=Riyadh/O=Sahoool/CN=$DOMAIN"

    # تجديد إعداد Nginx
    cat > /etc/nginx/sites-available/sahoool << EOF
upstream saahoool_backend {
    server 127.0.0.1:3000;
    keepalive 64;
}

server {
    listen 443 ssl http2;
    server_name ${DOMAIN};

    ssl_certificate /etc/nginx/ssl/sahoool.crt;
    ssl_certificate_key /etc/nginx/ssl/sahoool.key;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers HIGH:!aNULL:!MD5;

    # Security Headers
    add_header X-Frame-Options "SAMEORIGIN" always;
    add_header X-Content-Type-Options "nosniff" always;
    add_header X-XSS-Protection "1; mode=block" always;

    # Logging
    access_log /var/log/nginx/sahoool-access.log;
    error_log /var/log/nginx/sahoool-error.log;

    # Client Body Size
    client_max_body_size 50M;

    # Proxy Settings

```

```

location / {
    proxy_pass http://sahoool_backend;
    proxy_http_version 1.1;

    proxy_set_header Upgrade \$http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host \$host;
    proxy_set_header X-Real-IP \$remote_addr;
    proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto \$scheme;

    proxy_cache_bypass \$http_upgrade;
    proxy_buffering off;
}

# Static Files
location /assets {
    alias /opt/sahoool-platform/dist/public/assets;
    expires 1y;
    add_header Cache-Control "public, immutable";
}
}

server {
    listen 80;
    server_name \$DOMAIN;
    return 301 https://\$server_name\$request_uri;
}
EOF

nginx -t
systemctl restart nginx
fi

print_success "نجاح تم إعداد SSL"
rm -f /tmp/sahoool_domain
}

#إعداد النسخ الاحتياطي
setup_backup() {
    print_info "...إعداد النسخ الاحتياطي التلقائي"
    mkdir -p /backup/sahoool

    #إنشاء سكريبت النسخ الاحتياطي
    cat > /opt/sahoool-platform/backup.sh << 'EOF'

```

```

#!/bin/bash
BACKUP_DIR="/backup/sahoool"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR

# نسخ قاعدة البيانات
DB_PASS=$(grep DATABASE_URL /opt/sahoool-platform/.env | cut -d':' -f3 | cut -d'@' -f1)
mysqldump -u sahool_user -p"$DB_PASS" sahool > $BACKUP_DIR/db_$DATE.sql

# مخطط النسخة
gzip $BACKUP_DIR/db_$DATE.sql

# حذف النسخ القديمة (أكثر من 7 أيام)
find $BACKUP_DIR -name "*.sql.gz" -mtime +7 -delete

echo "Backup completed: $DATE"
EOF

chmod +x /opt/sahoool-platform/backup.sh

# إضافة إلى cron
(crontab -l 2>/dev/null; echo "0 2 * * * /opt/sahoool-platform/backup.sh
>> /var/log/sahoool-backup.log 2>&1") | crontab -
}

print_success "تم إعداد النسخ الاحتياطي التلقائي (يومياً الساعة 2 صباحاً)"

# التحقق من التنصيب
verify_installation() {
    print_info "...التحقق من التنصيب"

    # التتحقق من الخدمات
    SERVICES=("mysql" "redis-server" "nginx" "pm2")

    for service in "${SERVICES[@]}"; do
        if systemctl is-active --quiet "$service" 2>/dev/null || pm2 list | grep -q "sahoool-platform" 2>/dev/null; then
            print_success "$service يعمل بنجاح"
        else
            print_warning "$service قد لا يعمل بشكل صحيح"
        fi
    done
}

# التتحقق من التطبيق

```

```

sleep 3
if curl -s http://localhost:3000 > /dev/null; then
    print_success "التطبيق يعمل بنجاح"
else
    print_warning "التطبيق قد لا يعمل بشكل صحيح"
fi
}

# رسالة النهاية
final_message() {
    DOMAIN=$(hostname -I | awk '{print $1}')

    echo ""
    echo "██████████"
    echo "||"
    echo "||           تم التنصيب بنجاح 🎉"
    echo "||"
    echo "██████████"
    echo ""

    print_success "!منصة سهول جاهزة للاستخدام"
    echo ""

    echo "معلومات الوصول:"
    echo "  - HTTP: http://$DOMAIN"
    echo "  - HTTPS: https://$DOMAIN إذا تم تفعيل (SSL)"
    echo ""

    echo "الأوامر المفيدة"
    echo "  - عرض حالة التطبيق: pm2 status"
    echo "  - عرض اللوجات: pm2 logs sahool-platform"
    echo "  - إعادة التشغيل: pm2 restart sahool-platform"
    echo "  - إيقاف التطبيق: pm2 stop sahool-platform"
    echo ""

    echo "مسارات مهمة:"
    echo "  - المشروع: /opt/sahool-platform"
    echo "  - النسخ الاحتياطي: /backup/sahool"
    echo "  - لوجات Nginx: /var/log/nginx/sahool-*.log"
    echo ""

    print_info "لمزيد من المعلومات: https://github.com/kafaat/sahool-platform"
    echo ""
}

=====
# البرنامج الرئيسي
=====

main() {
    check_root
}

```

```
welcome_message

update_system
install_nodejs
install_mysql
install_redis
install_nginx
install_pm2
setup_firewall

download_project
setup_env
install_dependencies
build_project
setup_database

setup_pm2
setup_nginx
setup_ssl
setup_backup

verify_installation
final_message
}

# تشغيل البرنامج الرئيسي
main

exit 0
```

ملفات الاعداد الإضافية

2. ملف (PM2 بديل لـ systemd service)

احفظ باسم : sahool.service

[Unit]

```
Description=Sahool Platform
Documentation=https://github.com/kafaat/sahool-platform
After=network.target mysql.service redis.service
```

[Service]

```
Type=simple
User=root
WorkingDirectory=/opt/sahool-platform
Environment=NODE_ENV=production
ExecStart=/usr/bin/node /opt/sahool-platform/dist/index.js
Restart=on-failure
RestartSec=10
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=sahool-platform
```

[Install]

```
WantedBy=multi-user.target
```

التشييت:

```
sudo cp sahool.service /etc/systemd/system/
sudo systemctl daemon-reload
sudo systemctl enable sahool
sudo systemctl start sahool
```

3. سكريبت التحديث

: update.sh : احفظ باسم

```

#!/bin/bash

=====
# سكريبت تتحديث منصة سهول
=====

set -e

APP_DIR="/opt/sahool-platform"
BACKUP_DIR="/backup/sahool/updates"

echo "⚡ ... بدء عملية التحديث"

# إنشاء نسخة احتياطية
echo "📦 ... إنشاء نسخة احتياطية"
mkdir -p "$BACKUP_DIR"
DATE=$(date +%Y%m%d_%H%M%S)

# نسخ قاعدة البيانات
DB_PASS=$(grep DATABASE_URL "$APP_DIR/.env" | cut -d':' -f3 | cut -d'@' -f1)
mysqldump -u sahool_user -p"$DB_PASS" sahool | gzip >
"$BACKUP_DIR/db_before_update_$DATE.sql.gz"

# نسخ الملفات
tar -czf "$BACKUP_DIR/app_before_update_$DATE.tar.gz" -C /opt sahool-
platform

echo "✅ تم إنشاء النسخة الاحتياطية"

# إيقاف التطبيق
echo "🚧 ... إيقاف التطبيق"
pm2 stop sahool-platform

# تتحديث الكود
echo "⬇️ تتحديث الكود من GitHub...""
cd "$APP_DIR"
git pull origin master

# تثبيت الحزم الجديدة
echo "📦 ... تثبيت الحزم"
pnpm install --frozen-lockfile

# بناء المشروع
echo "🔨 ... بناء المشروع"

```

```
pnpm run build

# تحديث قاعدة البيانات
echo "Updating database..." > /dev/null
pnpm db:push

# إعادة تشغيل التطبيق
echo "...Restarting application..." > /dev/null
pm2 restart sahool-platform

# التحقق من الحالة
sleep 3
pm2 status sahool-platform

echo "✓ Success!" > /dev/null
echo "Logs: pm2 logs sahool-platform"
```

4. سكريبت المراقبة

: monitor.sh احفظ باسم

```

#!/bin/bash

#=====#
# سكريبت مراقبة منصة سهول
#=====#

APP_DIR="/opt/sahool-platform"

# الألوان
GREEN='\033[0;32m'
RED='\033[0;31m'
YELLOW='\033[1;33m'
NC='\033[0m'

echo "_____|"
echo "||           منصة سهول - لوحة المراقبة          ||"
echo "||_____|"
echo ""

# حالة الخدمات
echo "📊: حالة الخدمات"
echo
""

services=("mysql" "redis-server" "nginx")
for service in "${services[@]}"; do
    if systemctl is-active --quiet "$service"; then
        echo -e "  ✓ $service: ${GREEN}يُعمل${NC}"
    else
        echo -e "  ✗ $service: ${RED}متوقف${NC}"
    fi
done

# حالة التطبيق
if pm2 list | grep -q "sahool-platform.*online"; then
    echo -e "  ✓ sahool-platform: ${GREEN}يُعمل${NC}"
else
    echo -e "  ✗ sahool-platform: ${RED}متوقف${NC}"
fi

echo ""

# استخدام الموارد
echo "💻: استخدام الموارد"
echo
"
```

```

"-----



# CPU
CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print $2}' | cut -d'%' -f1)
echo "  CPU: ${CPU_USAGE}%



# RAM
MEM_TOTAL=$(free -m | awk 'NR==2{print $2}')
MEM_USED=$(free -m | awk 'NR==2{print $3}')
MEM_PERCENT=$(awk "BEGIN {printf \"%.1f\", ($MEM_USED/$MEM_TOTAL)*100}")
echo "  RAM: ${MEM_USED}MB / ${MEM_TOTAL}MB (${MEM_PERCENT}%)



# Disk
DISK_USAGE=$(df -h / | awk 'NR==2{print $5}')
echo "  Disk: ${DISK_USAGE} مسخنداه



echo ""



# قاعدة البيانات
echo "  قاعدة البيانات"
echo "-----



DB_PASS=$(grep DATABASE_URL "${APP_DIR}/.env" | cut -d':' -f3 | cut -d'@' -f1)
DB_SIZE=$(mysql -u sahool_user -p"${DB_PASS}" -e "SELECT
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size (MB)' FROM
information_schema.tables WHERE table_schema='sahool';" -sN)
echo "  حجم قاعدة البيانات: ${DB_SIZE} MB



TABLE_COUNT=$(mysql -u sahool_user -p"${DB_PASS}" -e "SELECT COUNT(*) FROM
information_schema.tables WHERE table_schema='sahool';" -sN)
echo "  عدد الجداول: ${TABLE_COUNT}



echo ""



# آخر 5 سطور من اللوگات
echo "  آخر اللوگات"
echo "-----



pm2 logs sahool-platform --lines 5 --nostream



echo ""
echo "  لعرض اللوگات الكاملة: pm2 logs sahool-platform"
"-----"

```

دليل التشغيل السريع



خطوات التنصيب:

```
# تحميل سكريبت التنصيب 1.  
wget https://raw.githubusercontent.com/kafaat/sahoool-  
platform/master/install.sh  
  
# إعطاء صلاحيات التنفيذ 2.  
chmod +x install.sh  
  
# تشغيل السكريبت 3.  
sudo bash install.sh
```

بعد التنصيب:

```
# عرض حالة التطبيق  
pm2 status  
  
# عرض اللوجات  
pm2 logs sahool-platform  
  
# إعادة التشغيل  
pm2 restart sahool-platform  
  
# المراقبة  
bash monitor.sh  
  
# التحديث  
bash update.sh
```

الدعم

للمزيد من المعلومات:

GitHub: <https://github.com/kafaat/sahoool-platform> •

• التوثيق: https://github.com/kafaat/sahool-platform/blob/master/API_DOCUMENTATION.md

تم إنشاء هذا الملف بواسطة فريق منصة سهول