

الشامل لنظام Docker دليل Scan AI Gaara

نظام الحاويات المتقدم مع Watchtower و Portainer

جدول المحتويات

1. [نظرة عامة](#)
2. [متطلبات النظام](#)
3. [التنسيق والإعداد](#)
4. [هيكل الحاويات](#)
5. [إدارة النظام](#)
6. [المراقبة والصيانة](#)
7. [استكشاف الأخطاء](#)

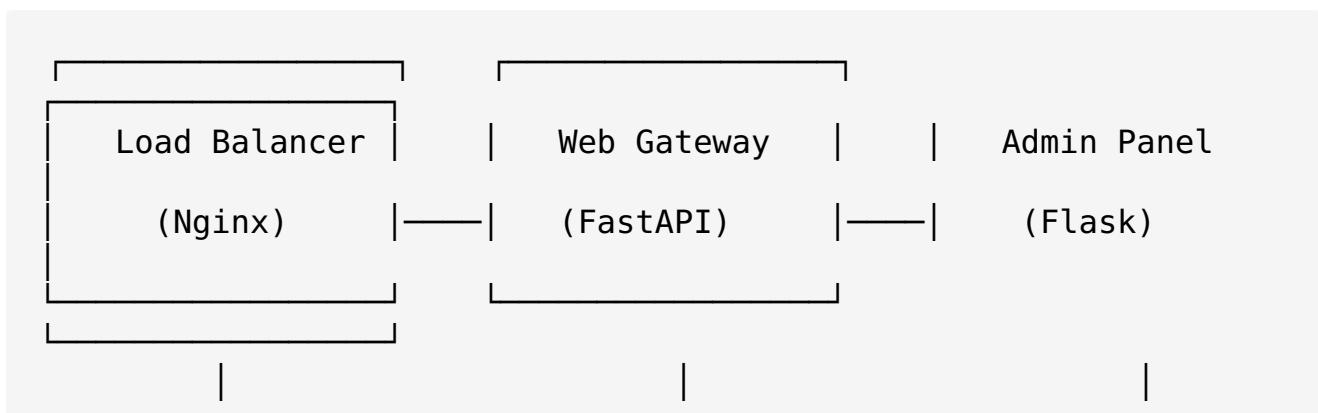
نظرة عامة

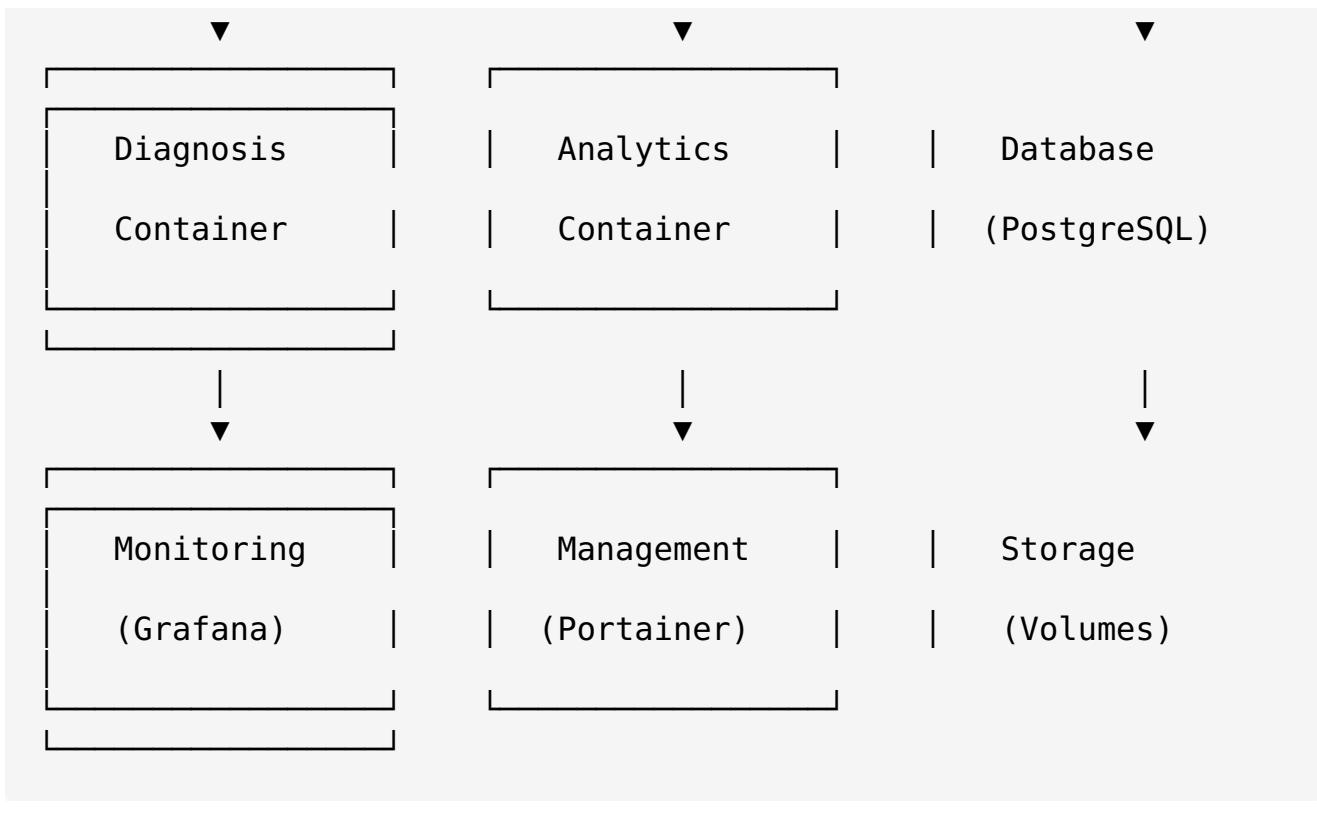
ليكون نظاماً شاملاً ومتقدماً يدعم Docker لمشروع Scan AI تم تصميم نظام:

الميزات الرئيسية

- تحسين حجم الحاويات والأداء: **بناء متعدد المراحل**
- للفحص والتشخيص والتحليلات: **حاويات منفصلة**
- مع **Watchtower و Portainer**: إدارة تلقائية
- مع **Grafana و Prometheus**: مراقبة متقدمة
- كوكيز عكسي Nginx مع: **أمان محسن**
- من موقع: **تصميم مستوحى MAGseeds.com**

المعمارية





متطلبات النظام

الحد الأدنى للمتطلبات

- أنيوحة 4 CPU: **المعالج**
- جيجابايت 8 RAM: **الذاكرة**
- جيجابايت مساحة فارغة 50: **التخزين**
- Ubuntu 20.04+ / CentOS 8+ / Docker Desktop: **نظام التشغيل**

المتطلبات الموصى بها

- أو أكثر CPU أنيوحة 8: **المعالج**
- أو أكثر RAM جيجابايت 16: **الذاكرة**
- جيجابايت 100 SSD: **التخزين**
- اتصال إنترنت مستقر: **الشبكة**

البرامج المطلوبة

```
# Docker Engine 20.10+
# Docker Compose 2.0+
# Git
# curl/wget
```

التثبيت والإعداد

1. تثبيت Docker

```
# Ubuntu/Debian
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
```

: إعادة تسجيل الدخول أو تشغيل #
newgrp docker

التحقق من التثبيت
docker --version
docker compose version

2. تحضير المشروع

استنساخ المشروع #
git clone <repository-url>
cd gaara-scan-ai

إنشاء ملف البيئة #
cp .env.example .env

تحرير متغيرات البيئة #
nano .env

3. إعداد متغيرات البيئة

```
# ملف .env
# =====
# إعدادات قاعدة البيانات
POSTGRES_DB=gaara_scan_ai
POSTGRES_USER=gaara_admin
POSTGRES_PASSWORD=your_secure_password_here
POSTGRES_HOST=postgres
POSTGRES_PORT=5432
```

إعدادات التطبيق
SECRET_KEY=your_secret_key_here
DEBUG=false
ENVIRONMENT=production

إعدادات الأمان
JWT_SECRET_KEY=your_jwt_secret_here

```
ENCRYPTION_KEY=your_encryption_key_here
```

إعدادات البريد الإلكتروني

```
SMTP_HOST=smtp.gmail.com
```

```
SMTP_PORT=587
```

```
SMTP_USER=your_email@gmail.com
```

```
SMTP_PASSWORD=your_app_password
```

إعدادات التخزين السحابي

```
AWS_ACCESS_KEY_ID=your_aws_key
```

```
AWS_SECRET_ACCESS_KEY=your_aws_secret
```

```
AWS_BUCKET_NAME=gaara-scan-ai-storage
```

إعدادات المراقبة

```
GRAFANA_ADMIN_PASSWORD=admin_password_here
```

```
PROMETHEUS_RETENTION=30d
```

إعدادات Portainer

```
PORTAINER_PASSWORD=portainer_password_here
```

4. بناء وتشغيل النظام.

بناء جميع الحاويات #

```
docker compose build
```

تشغيل النظام #

```
docker compose up -d
```

مراقبة السجلات #

```
docker compose logs -f
```

هيكل الحاويات

الحاويات الأساسية

1. التطبيق الرئيسي (gaara-main)

المنفذ: 8000

الوظيفة: واجهة برمجة التطبيقات الرئيسية #

```
# التقنيات: FastAPI, Python 3.11
```

```
# الموارد: 2GB RAM, 2 CPU cores
```

2. لوحة الإدارة (gaara-admin)

```
# المنافذ: 5000
# الوظيفة: واجهة إدارة النظام
# التقنيات: Flask, Jinja2
# الموارد: 1GB RAM, 1 CPU core
```

3. حاوية التشخيص (gaara-diagnosis)

```
# المنافذ: 8001
# الوظيفة: تشخيص الأمراض النباتية
# التقنيات: PyTorch, OpenCV
# الموارد: 4GB RAM, 2 CPU cores
```

4. حاوية التحليلات (gaara-analytics)

```
# المنافذ: 8002
# الوظيفة: تحليل البيانات الضخمة
# التقنيات: Pandas, Scikit-learn
# الموارد: 3GB RAM, 2 CPU cores
```

الحاويات المساعدة

5. قاعدة البيانات (postgres)

```
# المنافذ: 5432
# الوظيفة: تخزين البيانات الرئيسية
# الإصدار: PostgreSQL 15
# الموارد: 2GB RAM, 1 CPU core
```

6. التخزين المؤقت (redis)

```
# المنافذ: 6379
# الوظيفة: تخزين مؤقت وجلسات
# الإصدار: Redis 7
# الموارد: 512MB RAM
```

7. الوكيل العكسي (nginx)

```
# المنافذ: 80, 443
# الوظيفة: توزيع الأحمال والأمان
```

```
# الإصدار : Nginx 1.24
# الموارد: 256 MB RAM
```

حاويات المراقبة والإدارة

8. Grafana

```
# المنفذ: 3000
الوظيفة: مراقبة ولوحات تحكم
# الإصدار : Grafana 10
# الموارد: 512 MB RAM
```

9. Prometheus

```
# المنفذ: 9090
الوظيفة: جمع المقايس
# الإصدار : Prometheus 2.45
# الموارد: 1 GB RAM
```

10. Portainer

```
# المنفذ: 9000
الوظيفة: إدارة الحاويات
# الإصدار : Portainer CE 2.19
# الموارد: 256 MB RAM
```

11. Watchtower

```
الوظيفة: تحديث تلقائي للحاويات
# الجدولة: كل 24 ساعة
# الموارد: 128 MB RAM
```

إدارة النظام

الأساسية أوامر Docker Compose

تشغيل النظام

```
تشغيل جميع الخدمات
docker compose up -d
```

تشغيل خدمة محددة #
docker compose up -d gaara-main

تشغيل مع إعادة البناء #
docker compose up -d --build

إيقاف النظام

إيقاف جميع الخدمات #
docker compose down

إيقاف مع حذف البيانات #
docker compose down -v

إيقاف خدمة محددة #
docker compose stop gaara-main

مراقبة النظام

عرض حالة الخدمات #
docker compose ps

مراقبة السجلات #
docker compose logs -f

مراقبة سجلات خدمة محددة #
docker compose logs -f gaara-main

عرض استخدام الموارد #
docker stats

صيانة النظام

إعادة تشغيل خدمة #
docker compose restart gaara-main

تحديث الحاويات #
docker compose pull
docker compose up -d

تنظيف النظام #
docker system prune -a
docker volume prune

ادارة البيانات

النسخ الاحتياطي

نسخ احتياطي لقاعدة البيانات #

```
docker compose exec postgres pg_dump -U gaara_admin  
gaara_scan_ai > backup.sql
```

نسخ احتياطي للملفات المرفوعة #

```
docker run --rm -v gaara_uploads:/data -v $(pwd):/backup alpine  
tar czf /backup/uploads_backup.tar.gz -C /data .
```

نسخ احتياطي شامل #

```
./scripts/backup.sh
```

الاستعادة

استعادة قاعدة البيانات #

```
docker compose exec -T postgres psql -U gaara_admin  
gaara_scan_ai < backup.sql
```

استعادة الملفات #

```
docker run --rm -v gaara_uploads:/data -v $(pwd):/backup alpine  
tar xzf /backup/uploads_backup.tar.gz -C /data
```

استعادة شاملة #

```
./scripts/restore.sh backup_file.tar.gz
```

التحديثات التلقائية مع Watchtower

إعداد Watchtower

في docker-compose.yml

watchtower:

image: containrrr/watchtower:latest

volumes:

- /var/run/docker.sock:/var/run/docker.sock

environment:

- WATCHTOWER_CLEANUP=true

- WATCHTOWER_SCHEDULE=0 0 2 * * # كل يوم في الساعة 2

صباحاً

- WATCHTOWER_NOTIFICATIONS=email

- WATCHTOWER_NOTIFICATION_EMAIL_TO=admin@example.com

مراقبة التحديثات

```
# عرض سجلات Watchtower
docker compose logs watchtower

# تشغيل تحديث فوري
docker compose exec watchtower watchtower --run-once
```

المراقبة والصيانة

الوصول إلى لوحات التحكم

Portainer - إدارة الحاويات

URL: <http://localhost:9000>
المستخدم: admin
كلمة المرور من ملف .env.

: الميزات
- والصور الحاويات إدارة
- الموارد مراقبة
- والأحجام الشبكات إدارة
- السجلات عرض

Grafana - المراقبة والتحليلات

URL: <http://localhost:3000>
المستخدم: admin
كلمة المرور من ملف .env.

: المتابعة التحكم لوحات
- النظام على عام نظرة
- التطبيق أداء
- البيانات قاعدة استخدام
- الأمان مقاييس

Prometheus - جمع المقاييس

URL: <http://localhost:9090>

: المتابعة المقاييس
- والذاكرة CPU استخدام
- الاستجابة أوقات

- الطلبات عدد
- الخدمات حالة

مراقبة الصحة

فحص حالة الخدمات

```
فحص صحة جميع الخدمات #  
curl http://localhost/health
```

```
فحص خدمة محددة #  
curl http://localhost:8000/health  
curl http://localhost:5000/health
```

مراقبة الأداء

```
استخدام الموارد #  
docker stats --format "table {{.Container}}\t{{.CPUPerc}}\t{{.MemUsage}}\t{{.NetIO}}"
```

```
مساحة القرص #  
df -h  
docker system df
```

التنبيهات والإشعارات

إعداد التنبيهات في Grafana

```
{  
  "alert": {  
    "name": "High CPU Usage",  
    "conditions": [  
      {  
        "query": "cpu_usage",  
        "threshold": 80  
      }  
    ],  
    "notifications": [  
      {  
        "type": "email",  
        "addresses": ["admin@example.com"]  
      }  
    ]  
  }  
}
```

إعداد إشعارات Watchtower

```
# متغيرات البيئة في .env
WATCHTOWER_NOTIFICATIONS=email
WATCHTOWER_NOTIFICATION_EMAIL_FROM=watchtower@example.com
WATCHTOWER_NOTIFICATION_EMAIL_TO=admin@example.com
WATCHTOWER_NOTIFICATION_EMAIL_SERVER=smtp.gmail.com
WATCHTOWER_NOTIFICATION_EMAIL_SERVER_PORT=587
WATCHTOWER_NOTIFICATION_EMAIL_SERVER_USER=your_email@gmail.com
WATCHTOWER_NOTIFICATION_EMAIL_SERVER_PASSWORD=your_app_password
```

استكشاف الأخطاء

المشاكل الشائعة وحلولها

1. فشل في بناء الحاوية.

```
# المشكلة: خطأ في بناء الصورة
# الحل:
docker compose build --no-cache gaara-main
docker system prune -a
```

2. عدم الاتصال بقاعدة البيانات.

```
# المشكلة: Connection refused
# الحل:
docker compose logs postgres
docker compose restart postgres

# فحص الاتصال
docker compose exec gaara-main python -c "
import psycopg2
conn = psycopg2.connect(
    host='postgres',
    database='gaara_scan_ai',
    user='gaara_admin',
    password='your_password'
)
print('Connection successful!')"
```

نفاد مساحة القرص .3.

```
# : المشكلة No space left on device
# الحل:
docker system prune -a
docker volume prune
docker image prune -a

# حذف السجلات القديمة
sudo journalctl --vacuum-time=7d
```

بطء في الأداء .4.

```
# فحص استخدام الموارد
docker stats

# زيادة الموارد المخصصة
# تحرير docker-compose.yml:
deploy:
  resources:
    limits:
      memory: 4G
      cpus: '2'
```

مشاكل الشبكة .5.

```
# فحص الشبكات
docker network ls
docker network inspect gaara_network

# إعادة إنشاء الشبكة
docker compose down
docker network prune
docker compose up -d
```

سجلات التشخيص

جمع معلومات التشخيص

```
#!/bin/bash
# سكريبت جمع معلومات التشخيص

echo "==== معلومات النظام ===" > diagnosis.log
uname -a >> diagnosis.log
docker --version >> diagnosis.log
```

```
docker compose version >> diagnosis.log
echo "==== حاله الحاويات ===" >> diagnosis.log
docker compose ps >> diagnosis.log

echo "==== استخدام الموارد ===" >> diagnosis.log
docker stats --no-stream >> diagnosis.log

echo "==== سجلات الأخطاء ===" >> diagnosis.log
docker compose logs --tail=100 >> diagnosis.log

echo "==== مساحة القرص ===" >> diagnosis.log
df -h >> diagnosis.log
docker system df >> diagnosis.log

echo "تم حفظ معلومات التشخيص في diagnosis.log"
```

الدعم والمساعدة

قنوات الدعم

- في المشروع /docs: **التوثيق**
- GitHub Issues: **المشاكل**
- البريد الإلكتروني: support@garaa-scan-ai.com
- الدردشة: Discord/Slack

معلومات مفيدة للدعم

نظام التشغيل 2. Docker و Docker Compose عند طلب المساعدة، يرجى تضمين:
1. إصدار سجلات الأخطاء 5. خطوات إعادة إنتاج المشكلة 4. ملف docker-compose.yml 3. ملف

ملاحظات إضافية

أفضل الممارسات

- قم بعمل نسخ احتياطية يومية: **النسخ الاحتياطي المنتظم**
- راقب استخدام CPU والذاكرة: **مراقبة الموارد**
- حافظ على تحديث الحاويات: **تحديث الأمان**
- راجع السجلات بانتظام: **مراجعة السجلات**
- اخبر عملية الاستعادة دورياً: **اختبار الاستعادة**

الأمان

- استخدم كلمات مرور قوية
- في الإنتاج HTTPS فعال
- قم بتحديث الحاويات بانتظام
- راجع سجلات الأمان
- استخدم جدار حماية

الأداء

- خصص موارد كافية
 - للتخزين SSD استخدم
 - راقب استخدام الشبكة
 - حسن استعلامات قاعدة البيانات
 - استخدم التخزين المؤقت
-

نظام الذكاء الاصطناعي الزراعي المتقدم - Gaara Scan AI تم إنشاء هذا الدليل لنظام