

مشروع البنية السحابية والتطبيقات المتكاملة

Complete DevOps Infrastructure & Automation Project

ملخص تنفيذي

تم إنشاء مشروع متكامل يجمع بين:

- آمن وخوادم وقاعدة بيانات VPC متكاملة مع AWS بنية (Task Group A): البنية السحابية.
- ومراقبة شاملة GitHub Actions نشر تلقائي مع CI/CD (Task Group B) والأتمتة

ما تم إنجازه

Task Group A: البنية الأساسية

البنية الشبكية.

- مع شبكات فرعية عامة وخاصة 10.0.0.0/16
- شبكة عامة EC2 instances 10.0.1.0/24
- شبكة خاصة RDS Database 10.0.2.0/24
- Internet Gateway: لاتصال بالإنترنت
- Security Groups: مُعدّة بشكل آمن مع أقل صلاحيات مطلوبة

خوادم التطبيقات (EC2)

- Frontend Server (Uptime Kuma)
 - OS: Ubuntu 22.04
 - Type: t2.micro (1 vCPU, 1GB RAM)
 - Storage: 8GB SSD
 - Application: Docker + Uptime Kuma على port 3001
 - IP: Elastic IP (ثابت)
- Backend Server (Laravel)
 - OS: Ubuntu 22.04
 - Type: t2.micro (1 vCPU, 1GB RAM)
 - Storage: 8GB SSD
 - Application: PHP 8.1 + Laravel + Nginx على port 80
 - IP: Elastic IP (ثابت)

3. قاعدة البيانات (RDS)

- **Engine:** MySQL Community 8.0.35
- **Instance Type:** db.t3.micro (الخطة الأقل سعراً)
- **Storage:** 20GB
- **Location:** Private Subnet (آمن، غير متاح من الإنترنت)
- **Database Name:** obeliondb
- **Backups:** 7 أيام retention
- **Access:** فقط من EC2 instances

الأمان والإدارة 4.

- **IAM Roles:** التقارير إلى EC2 للسماح لـ CloudWatch
- **Security Groups:** تحديد دقيق للوصول
- **Elastic IPs:** ثابتة للخوادم
- **Auto-configuration:** User Data scripts لتنشيط وتشغيل التطبيقات

Task Group B: CI/CD ↴ والمراقبة

1. GitHub Actions لـ Frontend (Uptime Kuma)

الإجراء:

```
أي Push إلى main branch →
└─ Checkout Code
└─ Build Docker Image
└─ Push إلى Container Registry
└─ SSH إلى Frontend EC2
└─ Pull و Run Image
└─ Health Check
```

الميزات:

- بناء تلقائي لـ Docker images
- نشر فوري بدون توقف الخدمة
- فحص صحة التطبيق بعد النشر

2. GitHub Actions لـ Backend (Laravel)

الإجراء:

```
أي Push إلى main branch →
└─ Checkout Code
└─ Setup PHP 8.1
```

```
|--- Install Composer Dependencies
|--- SSH إلى Backend EC2
|--- git pull (سحب التحديثات)
|--- php artisan migrate (تطبيق تغييرات قاعدة البيانات)
|--- Clear Cache
|--- Set Permissions
|--- Restart Services
```

الميزات:

- نشر آمن Database Migrations
- تطبيق تغييرات المخطط تلقائياً
- لتطبيق التحديثات Cache تنظيف إلـ

3. المراقبة والتنبيهات

CloudWatch Alarms:

- Frontend CPU > 50% → Email Alert
- Backend CPU > 50% → Email Alert
- RDS CPU > 50% → Email Alert
- Disk Space > 80% → Email Alert

SNS Integration:

- Topic لـ Email Notifications
- Subscription للبريد الإلكتروني
- تنبيهات فورية للمشاكل

الملفات المُنشأة

ملفات Terraform (Infrastructure as Code):

1. terraform-main.tf (البنية الأساسية - VPC, IGW, Security Groups)
2. terraform-ec2.tf - خوادم EC2 مع User Data
3. terraform-rds.tf - قاعدة بيانات MySQL
4. terraform-monitoring.tf - CloudWatch Alarms و SNS
5. terraform-variables.tf - المتغيرات القابلة للتخصيص

ملفات GitHub Actions:

1. github-actions-frontend.yml - Workflow لـ Frontend
2. github-actions-backend.yml - Workflow لـ Backend

ملفات التوثيق:

1. [documentation.md](#) - توثيق شامل (عربي + إنجليزي)
2. [quickstart.md](#) - دليل البدء السريع
3. [summary.md](#) - ملخص تنفيذي
4. [README.md](#) - الفهرس الرئيسي
5. [github-secrets-guide.md](#) - دليل إعداد Secrets

أمثلة إضافية:

1. [Dockerfile.frontend-example](#) - مثال Dockerfile
2. [docker-compose.yml.example](#) - مثال docker-compose

المخططات المعمارية:

1. [aws-vpc-arch.png](#) - مخطط البنية الأساسية
2. [devops-full-arch.png](#) - مخطط معماري شامل
3. [DevOps-Architecture Dashboard](#) - واجهة تفاعلية

المتطلبات التقنية

للتطوير:

- Terraform >= 1.0
- AWS CLI >= 2.0
- SSH Key لـ EC2
- Git و GitHub Account

الموارد المنشأة:

- 1x VPC (10.0.0.0/16)
- 2x Subnets (Public + Private)
- 1x Internet Gateway
- 2x EC2 t2.micro
- 1x RDS MySQL db.t3.micro
- 2x Security Groups
- 2x Elastic IPs
- 4x CloudWatch Alarms
- 1x SNS Topic
- 1x IAM Role

تقدير التكاليف

الشهري (Approximate):

الخدمة	السعر
EC2 t2.micro x2	\$17/شهر
RDS db.t3.micro	\$10/شهر
Data Transfer	\$5/شهر
الإجمالي	~\$32/شهر

ملاحظة: مجاني للسنة الأولى إذا كنت في AWS Free Tier

خطوات الاستخدام

1. التحضير (15 دقيقة)

- تثبيت Terraform و AWS CLI
- إعداد AWS Credentials

2. النشر (30 دقيقة)

- نسخ ملفات Terraform
- إنشاء terraform.tfvars
- تشغيل terraform apply

3. CI/CD (20 دقيقة | إعداد)

- إنشاء SSH Keys
- إضافة Secrets إلى GitHub
- نسخ Workflow files

4. الاختبار (10 دقائق)

- اختبار Frontend
- اختبار Backend
- اختبار GitHub Actions

الوقت الإجمالي: ~75 دقيقة

الميزات المُنجزة ✓

البنية الأساسية:

- [x] آمن مع شبكات منفصلة VPC
- [x] في شبكة عامة Frontend و Backend
- قاعدة بيانات في شبكة خاصة [x]
- [x] Security Groups مُحددة
- [x] IAM Roles مُعدّة

CI/CD:

- [x] GitHub Actions لـ Frontend
- [x] GitHub Actions لـ Backend
- [x] Automatic Deployments
- [x] Database Migrations التلقائية
- [x] Health Checks

المراقبة:

- [x] CloudWatch Monitoring
- [x] CPU Utilization Alerts
- [x] Disk Space Alerts
- [x] SNS Email Notifications
- [x] Real-time Metrics

التوثيق:

- [x] Terraform Code
- [x] GitHub Actions Workflows
- [x] Comprehensive Documentation
- [x] Quick Start Guide
- [x] Architecture Diagrams
- [x] Security Guide

الخطوات التالية (اختبارية) ▶

للبيئة الإنتاجية:

1. **Load Balancer:** إضافة Application Load Balancer
2. **Auto Scaling:** Auto Scaling Groups لـ EC2
3. **SSL/TLS:** AWS Certificate Manager
4. **CDN:** CloudFront للتوزيع العالمي
5. **Backup:** تحسين سياسات Backup
6. **Logging:** CloudWatch Logs و CloudTrail
7. **Cost Optimization:** Reserved Instances

للتطوير:

1. **Testing:** إضافة Automated Tests
2. **Staging:** بيئة Staging منفصلة
3. **GitOps:** Terraform Cloud/Enterprise
4. **Secret Management:** AWS Secrets Manager
5. **Infrastructure Audit:** CloudTrail و Config

الدعم والموارد

الوثائق:

- [Terraform Documentation](#)
- [AWS Documentation](#)
- [GitHub Actions](#)
- [Laravel Documentation](#)

الأدوات:

- AWS Console: <https://console.aws.amazon.com>
- GitHub: <https://github.com>
- Terraform Cloud: <https://app.terraform.io>

المساعدة:

- AWS Support (اعتماداً على الخطة)
- GitHub Community
- Stack Overflow
- DevOps Forums

اعتبارات الأمان

مُنفّذ:

- ✓ VPC معزول
- ✓ Security Groups محدودة
- ✓ IAM Least Privilege
- ✓ SSH بـ Key Authentication
- ✓ RDS في شبكة خاصة

الوصيات:

- ⚠ IP محدد لـ SSH (0.0.0.0/0) استخدم
- ⚠ فغل Encryption لـ RDS
- ⚠ AWS Secrets Manager استخدم
- ⚠ Regular Backups قم بـ
- ⚠ Disaster Recovery اختبر

الملخص

تم إنشاء مشروع متكمّل يشمل:

- معزول مع تقسيم منطقي آمن VPC: **البنية السحابية المتقدمة**.
- مع قاعدة بيانات Frontend و Backend: **التطبيقات الجاهزة**.
- CI/CD الكامل: نشر تلقائي وأمن مع GitHub Actions
- CloudWatch Alarms و Email Notifications: **المراقبة الشاملة**.
- التوثيق الاحترافي:** شامل وسهل الفهم (عربي + إنجليزي).

الخلاصة*

هذا المشروع يوفر:

- بنية جاهزة للإنتاج
- أتمتة كاملة للنشر
- مراقبة شاملة
- توثيق احترافي
- أمان محسّن

الحالة: جاهز للاستخدام الفوري والنشر على LinkedIn

آخر تحديث: 23 نوفمبر 2025

النسخة: 1.0

**

1. <https://registry.terraform.io/modules/terraform-aws-modules/rds/aws/1.24.0>
2. https://www.reddit.com/r/aws/comments/1jv07uc/how_to_set_up_aws sns_to_trigger_alerts_for_high/
3. <https://towardsdatascience.com/end-to-end-aws-rds-setup-with-bastion-host-using-terraform/>
4. <https://zeet.co/blog/terraform-aws-rds>
5. <https://github.com/cloudposse/terraform-aws-rds>
6. <https://www.easydeploy.io/blog/launching-ec2-instance-and-aurora-rds-instance/>
7. <https://dev.to/fabiancdng/automatic-deployment-using-docker-and-github-actions-16fb>
8. <https://notes.kodekloud.com/docs/AWS-CloudWatch/Anatomy-of-Alarms/Demo-Creating-and-Configuring-Alarms-with-alerting>
9. <https://developer.hashicorp.com/terraform/tutorials/aws/aws-rds>
10. <https://runcloud.io/blog/setup-docker-github-actions-ci-cd>