VIXOR-RLB-001

VIXOR ENGINEERING BIBLE

إنجيل عمليات الإطلاق

عملية إطلاق منظمة — من الكود إلى المستخدم النهائي

جدول المحتويات

نظرة عامة على عملية الإطلاق

البنية التحتية للإطلاق

مراحل الإطلاق

إدارة البيانات في الإطلاق

المهام المجدولة والإنتاج

المراقبة والتنبيهات

التراجع والإصلاح

قائمة فحص الإطلاق

SEC-01

نظرة عامة على عملية الإطلاق

يعتمد مشروع فيكسور على منصة فيركل كبيئة نشر رئيسية، حيث يتم تحويل الكود المصدري إلى تطبيق ويب متكامل يعمل ع
السحابة الحديثة. تم تصميم خط أنابيب الإطلاق وفق منهجية منظمة تضمن انتقال الكود من مرحلة التطوير إلى الإنتاج بسلاس
الحفاظ على استقرار المنصة وتجربة المستخدم النهائي.

يتكون خط أنابيب الإطلاق من خمس مراحل أساسية متسلسلة: **الكود** حيث يُكتب ويُراجع عبر طلبات الدمج، ثم **الاختبار** الذي ي
شاملة من الاختبارات الآلية للتحقق من صحة الوظائف والبنية، يليه **البناء** عبر أداة فايت مع إطار عمل تان ستاك لإنشاء ملفات الإ
ثم **النشر** إلى بيئة فيركل الإنتاجية، وأخيراً **التحقق** الذي يؤكد سلامة النشر عبر فحوصات شاملة.

استراتيجية الفروع

فرع `main` هو الفرع الوحيد المعتمد للإنتاج. كل دمج في هذا الفرع يُشغّل تلقائياً عملية النشر. الفروع الأخرى مخصصة للتطوير والاختبار فقط

يستخدم المشروع مدير الحزم **pnpm** بإصدار `9.15.0` لضمان تثبيت سريع وموحد للتبعيات. يتم تشغيل المشروع على
`nodejs22.x` مع إعداد مسبق `nitro` محسّن لمنصة فيركل. هذه المجموعة تضمن أداءً عالياً واستقراراً في بيئة الإنتاج، م
مع ميزات الجافاسكربت الحديثة.

SEC-02

البنية التحتية للإطلاق

تُعرَّف البنية التحتية لفيكسور من خلال ملف `vercel.json` الذي يحتوي على جميع الإعدادات اللازمة لعملية النشر على
يتضمن هذا الملف تهيئة الإعداد المسبق كـ `vercel` مع بيئة تشغيل `nodejs22.x` ، مما يضمن التوافق الأمثل مع متطل
وتوفير أحدث ميزات محرك فايت.

ملف vercel.json

يحدد هذا الملف إعدادات البناء والنشر including المهام المجدولة عبر خاصية `crons` . تحتوي هذه الخاصية على تعريفات نقا
يجب استدعاؤها بشكل دوري، مثل توليد الإشارات وفحص التنبيهات. كما يحدد الملف إعدادات التوجيه والرؤوس الأمنية و
الطلبات.

متغيرات البيئة

يعتمد المشروع على مجموعة من متغيرات البيئة الحساسة التي يتم تعيينها عبر لوحة تحكم فيركل. تشمل هذه المتغيرات: `URL`
للاتصال بقاعدة البيانات، `SUPABASE_SERVICE_KEY` للمصادقة مع خدمة سوبابيز، `TELEGRAM_BOT_TOKEN` لإرسال
تيليجرام، وغيرها من المفاتيح الخاصة بكل وحدة وظيفية.

تحذير أمني

لا يتم تخزين أي متغيرات بيئة في الكود المصدري أو في ملفات الإعداد. جميعها تُدار حصرياً عبر واجهة فيركل مع تفعيل التشفير الكامل.

SEC-03

مراحل الإطلاق

قائمة فحص ما قبل الإطلاق

قبل البدء بعملية النشر، يجب استكمال عدة متطلبات أساسية: التأكد من اكتمال جميع مراجعات الكود، نجاح جميع الاختبارات
تحضير ملفات الترحيل الجديدة إن وجدت، تحديث متغيرات البيئة إذا لزم الأمر، والتأكد من عدم وجود تعارضات في فرع `main`

مرحلة البناء

يتم تشغيل أمر `vite build` مع إطار عمل تان ستاك الذي يوفر بنية SSR متقدمة. يقوم البناء بتجميع جميع المكونات والتو
ملفات جافاسكربت وCSS مُحسّنة. يشمل البناء أيضاً إنشاء ملفات الخادم اللازمة لتقديم المحتوى من الجانب الخادم عبر بيئة نيترو

مرحلة النشر

يقوم فيركل تلقائياً بنشر الناتج على شبكة التوزيع العالمية. يتم توزيع التطبيق على نقاط حضور متعددة لضمان سرعة الوصول من
العالم. يستغرق النشر عادةً أقل من دقيقتين.

التحقق بعد النشر

بعد اكتمال النشر يتم تنفيذ فحص صحي عبر `/api/health` للتأكد من استجابة الخادم، واختبار دخان شامل يتضمن ف
الرئيسية ونقاط API الأساسية واتصال قاعدة البيانات ووظائف المصادقة. أي فشل في هذه الفحوصات يستدعي إطلاق إجراء التر

SEC-04

إدارة البيانات في الإطلاق

تُعد إدارة الترحيلات قاعدة بيانات سوبابيز من أهم مراحل عملية الإطلاق. يعتمد المشروع على نظام ترحيل منظّم يضمن تتبع ج
على البنية التحتية للبيانات والقدرة على التراجع عنها عند الحاجة.

عملية الترحيل

يتم تنفيذ الترحيلات عبر نقطتي نهاية: نقطة `/api/migrate` في التطبيق المُنشر لعمليات الترحيل الآلية، وأداة سطر أوامر
للعمليات المحلية. يفضل تنفيذ الترحيلات محلياً أولاً والتحقق من نتائجها قبل دمجها في الإنتاج.

اصطلاح التسمية

تُسمّى ملفات الترحيل وفق صيغة زمنية دقيقة: `YYYYMMDDHHMMSS_description.sql` . يضمن هذا الاصطلاح ترتيب الت
تلقائياً ويسهّل تتبع التغييرات زمنياً. مثال: `20250115143000_add_signals_table.sql` .

استراتيجية التراجع عن الترحيل

يجب أن يرافق كل ملف ترحيل ملف تراجع معكوس يُنفّذ في حال فشل النشر. لا تُستخدم أوامر الحذف المباشر، بل يُعتمد على ترحيل إضاف
الحالة السابقة بشكل آمن دون فقدان بيانات غير مقصود.

يتم تخزين جميع ملفات الترحيل في مجلد مخصص ضمن المشروع وتُخضع لمراجعة الكود كأي تغيير آخر. لا يُسمح بتنفيذ ترحيلا
قاعدة بيانات الإنتاج دون المرور بنقطة الترحيل الرسمية.

SEC-05

المهام المجدولة والإنتاج

تستفيد فيكسور من نظام المهام المجدولة المدمج في منصة فيركل لتنفيذ عمليات دورية ضرورية لعمل التطبيق. يتم تعريف ه
ملف `vercel.json` ضمن قسم `crons` ، وتعمل بوصفها طلبات HTTP مؤقتة تستدعي نقاط نهاية محددة في التطبيق.

توليد الإشارات — `api/generate-signals/`

تُنفَّذ يومياً عند منتصف الليل بتوقيت UTC. تقوم بتحليل بيانات العملات الميمية على شبكة سولانا وتوليد إشارات تداول جديدة بناءً على نماذج الذكاء الاصطناعي.

فحص التنبيهات — `api/check-alerts/`

تُنفَّذ يومياً عند الساعة 00:30 بتوقيت UTC. تفحص شروط التنبيهات المُعدّة مسبقاً وتُرسل الإشعارات عبر بوت تيليجرام تفعّل أي شرط.

في بيئة الإنتاج، يقوم فيركل تلقائياً باستدعاء هذه النقاط في الأوقات المحددة دون الحاجة لأي إعداد إضافي. تمر الطلبات عبر التحتية للتطبيق مع توفير رأس خاص `Authorization` للتأكد من أن الطلب قادم من فيركل وليس من مصدر خارجي. يجب هذا الرأس في كل نقطة نهاية مجدولة.

ملاحظة حول التوقيت

تعمل جميع المهام المجدولة بتوقيت UTC. عند التخطيط لتغيير الجدول الزمني، يجب مراعاة الفرق مع توقيت المستخدمين المستهدفين و على جودة الإشارات المُنتجة.

SEC-06

المراقبة والتنبيهات

تشكّل المراقبة المستمرة خط الدفاع الأول ضد المشاكل في بيئة الإنتاج. توفر فيكسور طبقات متعددة من المراقبة تشمل فحو ومقاييس الأداء وتتبع الأخطاء، مما يضمن اكتشاف أي خلل والاستجابة له بأسرع وقت ممكن.

فحص الصحة — `api/health/`

نقطة نهاية مخصصة تُرجع حالة النظام الشاملة. تفحص هذه النقطة اتصال قاعدة البيانات سوبابيز، حالة خدمة تيليجرام، استجابة تطبيقات سولانا، والذاكرة المتاحة. تُرجع استجابة بصيغة JSON تحتوي على حالة كل مكون وعلامة زمنية.

مقاييس الأداء — `api/metrics/`

تُقدّم هذه النقطة مقاييس بصيغة بروميثيوس القياسية يمكن استهلاكها من خلال أنظمة المراقبة الخارجية. تشمل المقاييس: الواردة، زمن الاستجابة لكل نقطة نهاية، معدلات الخطأ، وحالة الاتصال بالخدمات الخارجية.

تتبع الأخطاء عبر سنتري

يتم ربط المشروع بخدمة سنتري لتتبع الأخطاء في الوقت الفعلي. تلتقط سنتري جميع الاستثناءات غير المعالجة وتُصنّفها حسب ا التنبيهات عبر تيليجرام عند رصد أخطاء حرجة تتطلب تدخلاً فورياً من فريق التطوير.

ما يجب مراقبته

زمن استجابة نقاط API، معدل الأخطاء (هدف أقل من 1%)، حالة اتصال قاعدة البيانات، نجاح المهام المجدولة، واستهلاك الموارد على فيرك

SEC-07

التراجع والإصلاح

رغم اتخاذ جميع الاحتياطات، قد تظهر مشاكل غير متوقعة بعد النشر. لذلك وُضعت إجراءات تراجع واضحة ومُجرّبة تضمن ال مستقرة في أقل وقت ممكن، مع حماية بيانات المستخدمين وسلامة النظام.

تراجع نشر فيركل

يوفر فيركل إمكانية التراجع إلى أي نشر سابق بنقرة واحدة من لوحة التحكم أو عبر واجهة سطر الأوامر. يحتفظ فيركل بسجل كامل ل النشر مع معرّف فريد لكل عملية. في حال اكتشاف مشكلة حرجة، يتم تنفيذ: تحديد آخر نشر سليم، التراجع إليه، ثم إجراء فحص للتأكد من استعادة الخدمة.

تراجع ترحيل قاعدة البيانات

إذا تضمن النشر ترحيلات جديدة وأُكتشف خلل فيها، يتم تنفيذ ملف التراجع المُعدّ مسبقاً لإرجاع البنية إلى حالتها السابقة. يجب التراجع بعناية قبل تنفيذه والتأكد من عدم تأثيره على البيانات الحالية.

تدفق الاستجابة للحوادث

الاكتشاف: تنبيه سنتري أو فشل فحص الصحة.
التقييم: تحديد شدة المشكلة ونطاق تأثيرها.
الاحتواء: تراجع النشر فوراً إذا كانت المشكلة حرجة.
التحليل: مراجعة السجلات وتحديد السبب الجذري.
الإصلاح: تطبيق الحل واختباره محلياً ثم إعادة النشر.

SEC-08

قائمة فحص الإطلاق

القائمة التالية تمثل الفحص الشامل والنهائي الذي يجب استكماله قبل كل عملية إطلاق إلى بيئة الإنتاج. تغطي هذه القائمة جميع من جودة الكود إلى التحقق النهائي، وتُعد المعيار الذهبي لضمان نشر ناجح وخالٍ من المشاكل.

#	البند	الفئة	الح
1	جميع مراجعات الكود مكتملة ومعتمدة	جودة الكود	☐
2	لا توجد تعارضات في فرع `main`	جودة الكود	☐
3	اختبارات الوحدة ناجحة محلياً	الاختبارات	☐
4	اختبارات التكامل ناجحة	الاختبارات	☐
5	لا توجد تحذيرات TypeScript جديدة	جودة الكود	☐
6	ترحيلات قاعدة البيانات مُجهّزة ومراجعة	البيانات	☐
7	ترحيلات التراجع مُجهّزة لكل ترحيل جديد	البيانات	☐
8	تنفيذ الترحيلات محلياً بنجاح	البيانات	☐
9	`SUPABASE_URL` مُحدّث في فيركل	متغيرات البيئة	☐
10	`TELEGRAM_BOT_TOKEN` صالح ويعمل	متغيرات البيئة	☐
11	جميع المفاتيح السرية محدّثة	متغيرات البيئة	☐
12	بناء `vite build` ناجح بدون أخطاء	البناء	☐
13	حجم البناء ضمن الحدود المقبولة	البناء	☐
14	إعداد `vercel.json` صحيح	البنية التحتية	☐
15	المهام المجدولة (`crons`) مُعرّفة	البنية التحتية	☐
16	سِنتري مُفعّل ويرسل التنبيهات	المراقبة	☐
17	فحص الصحة `/api/health` يستجيب	المراقبة	☐
18	مقاييس `/api/metrics` تعمل	المراقبة	☐

#	البند	الفئة	الح
19	الصفحة الرئيسية تُحمَّل بنجاح	التحقق	☐
20	المصادقة تعمل بشكل صحيح	التحقق	☐
21	اتصال قاعدة البيانات سليم	التحقق	☐
22	بوت تيليجرام يستقبل ويرسل الرسائل	التحقق	☐
23	إجراء التراجع مُحدّد وجاهز	الطوارئ	☐
24	فريق التطوير مُبلَّغ بوقت النشر	التواصل	☐

وثيقة رسمية لمنظومة فيكسور الهندسية

VIXOR — AI-Powered Trading Terminal