

คู่มือระบบสั่ง RUN การตรวจสอบสถานะระบบและแจ้งเตือนอัตโนมัติ

LINE & TELEGRAM • CRON-JOB.ORG + WINDOWS TASK SCHEDULER + CLI

ระบบสั่ง RUN การตรวจสอบสถานะระบบและแจ้งเตือนอัตโนมัติ LINE & TELEGRAM – CRON-JOB.ORG • WINDOWS TASK SCHEDULER • CLI

ระบบสั่งรับการตรวจสอบสถานะและ แจ้งเตือนอัตโนมัติ LINE & Telegram (ส่วน 6b. Setting and Tester การแจ้งเตือนอัตโนมัติ LINE & Telegram) ให้ทำงานตลอด 24 ชั่วโมงโดยไม่ต้องเปิดหน้าต่างทิ้งไว้ – ใช้ตัวกระตุ้น 3 อันที่ทำงานร่วมกันแต่ไม่ส่งซ้ำ เพราะทุกอันเรียก engine ตรวจสอบสถานะด้วยตัวกัน

การตั้งค่า (cron-job.org – ฟรี ไม่ต้องเปิดหน้า Dashboard ค้างไว้ตลอดบนคอมพิวเตอร์)

บริการจองตามเวลาแบบ free tier – เรียก endpoint ขึ้น C ของระบบจาก cloud ทุก 5 นาที

Job name	GNSS-AlertCheck		
Method / ทุก	GET • 5 นาที (300 วินาที)		
URL ที่เรียก (พร้อม key)	https://project12569.theeiot.com/EE6717..ert-cron.php (ใช้ค่าเริ่มต้น)		
แก้ URL ที่ใช้จริง (กรอกค่าใหม่ หรือกรอกเปล่า = ใช้ค่าเริ่มต้น)	https://project12569.theeiot.com/EE67172310228/api/v1/alert-cron.php	บันทึกค่าใหม่	เคลียร์ (ใช้ค่าเริ่มต้น)
Login เพื่อจัดการ	cron-job.org (บัญชีที่ส่งไว้ตอนสมัคร) → Jobs → Dashboard		
Key ที่ใช้	ไม่ต้องใช้ key อีกต่อไป (X-Admin-Key ถูกถอดออก) – ตั้ง URL เดิมจากช่องบนสุด (ถ้า URL เดิมมี ?key=... ให้ลบส่วนนั้นทิ้งได้เลย)		

ระบบสั่ง RUN การตรวจสอบสถานะระบบและแจ้งเตือนอัตโนมัติ LINE & TELEGRAM – CRON-JOB.ORG • WINDOWS TASK SCHEDULER • CLI

ระบบสั่งรับการตรวจสอบสถานะและ แจ้งเตือนอัตโนมัติ LINE & Telegram (ส่วน 6b. Setting and Tester การแจ้งเตือนอัตโนมัติ LINE & Telegram) ให้ทำงานตลอด 24 ชั่วโมงโดยไม่ต้องเปิดหน้าต่างทิ้งไว้ – ใช้ตัวกระตุ้น 3 อันที่ทำงานร่วมกันแต่ไม่ส่งซ้ำ เพราะทุกอันเรียก engine ตรวจสอบสถานะด้วยตัวกัน

การตั้งค่า (cron-job.org – ฟรี ไม่ต้องเปิดหน้า Dashboard ค้างไว้ตลอดบนคอมพิวเตอร์)

บริการจองตามเวลาแบบ free tier – เรียก endpoint ขึ้น C ของระบบจาก cloud ทุก 5 นาที

Job name	GNSS-AlertCheck		
Method / ทุก	GET • 5 นาที (300 วินาที)		
URL ที่เรียก (พร้อม key)	https://project12569.theeiot.com/EE6717..ert-cron.php		
แก้ URL ที่ใช้จริง (กรอกค่าใหม่ หรือกรอกเปล่า = ใช้ค่าเริ่มต้น)	https://project12569.theeiot.com/EE67172310228/api/v1/alert-cron.php	บันทึกค่าใหม่	เคลียร์ (ใช้ค่าเริ่มต้น)
Login เพื่อจัดการ	cron-job.org (บัญชีที่ส่งไว้ตอนสมัคร) → Jobs → Dashboard		
Key ที่ใช้	ไม่ต้องใช้ key อีกต่อไป (X-Admin-Key ถูกถอดออก) – ตั้ง URL เดิมจากช่องบนสุด (ถ้า URL เดิมมี ?key=... ให้ลบส่วนนั้นทิ้งได้เลย)		

✓ ขอบเขต

เอกสารฉบับนี้ปรับปรุงใหม่ให้ตรงกับหน้า RUN ล่าสุด โดยเน้นการตั้งค่า cron-job.org สำหรับเรียก endpoint ตรวจสอบสถานะและแจ้งเตือนทุก 5 นาที พร้อมอธิบายการเปลี่ยน URL และการใช้งานแบบไม่ต้องเปิด Dashboard ค้างไว้

สถานะสำคัญของเวอร์ชันนี้: ไม่ต้องใช้ X-Admin-Key แยกอีกต่อไป ให้ใช้ URL ที่ระบบแสดงในช่อง “URL ที่เรียก” เป็นค่าที่ cron-job.org เรียกโดยตรง

1. ภาพรวมการทำงานของระบบ RUN

ระบบ RUN มีหน้าที่กระตุ้น engine ฝั่ง Server ให้ตรวจสอบสถานะล่าสุดและประมวลผลการแจ้งเตือน LINE และ Telegram แบบอัตโนมัติ โดยไม่จำเป็นต้องเปิดหน้าเว็บ Dashboard ค้างไว้ตลอดเวลา

ระบบสั่ง RUN การตรวจสอบสถานะระบบและแจ้งเตือนอัตโนมัติ LINE & TELEGRAM – CRON-JOB.ORG • WINDOWS TASK SCHEDULER • CLI
ระบบสั่งรับการตรวจสอบสถานะและ แจ้งเตือนอัตโนมัติ LINE & Telegram (ส่วน 6b. Setting and Tester การแจ้งเตือนอัตโนมัติ LINE & Telegram) ให้ทำงานตลอด 24 ชั่วโมงโดยไม่ต้องเปิดหน้าเว็บทิ้งไว้ – ใช้ตัวกระตุ้น 3 ชั้นที่ทำงานร่วมกันแต่
ไม่ซ้ำ เพราะทุกชั้นเรียก engine ตรวจสอบสถานะด้วย

✓ หลักการทำงาน

หน้าเว็บระบุว่าระบบใช้ตัวกระตุ้น 3 ชั้น ได้แก่ cron-job.org, Windows Task Scheduler และ CLI ซึ่งเรียก engine ตรวจสอบสถานะตัวเดียวกัน เพื่อให้ระบบแจ้งเตือนทำงานได้ต่อเนื่อง 24 ชั่วโมง

รายการ	คำ / ความหมาย
cron-job.org	ตัวเลือกหลักบน Cloud แบบ free tier เรียก endpoint ทุก 5 นาที
Windows Task Scheduler	ตัวกระตุ้นจากคอมพิวเตอร์ Windows ตามรอบเวลาที่กำหนด
CLI	ใช้สั่ง RUN จาก command line เมื่อทดสอบหรือดูแลระบบ
Dashboard	ไม่ต้องเปิดค้างไว้ ระบบทำงานผ่าน endpoint ฝั่ง Server

i หมายเหตุ

ภาพที่แนบแสดงรายละเอียดการตั้งค่า cron-job.org เท่านั้น ดังนั้นฉบับนี้จะลงรายละเอียดขั้นตอนของ cron-job.org เป็นหลัก ส่วน Windows Task Scheduler และ CLI ระบุไว้ในภาพรวมโดยไม่เดาค่าที่ไม่ได้แสดงในภาพ

2. ค่าที่ต้องใช้จากหน้า RUN - cron-job.org

การตั้งค่า (cron-job.org – ฟรี ไม่ต้องเปิดหน้า Dashboard ดังไว้ตลอดบนคอมพิวเตอร์)
 บริการจองตามเวลาแบบ free tier – เรียก endpoint ขึ้น C ของระบบจาก cloud ทุก 5 นาที

Job name: GNSS-AlertCheck

Method / ทุก: GET • 5 นาที (300 วินาที)

URL ที่เรียก (พร้อม key): <https://project12569.theeiot.com/EE6717-ert-cron.php> (ใช้ทำระบบอัตโนมัติ)

แก้ URL ที่อ้างอิง (กรอกค่าใหม่ หรือกรอกเปล่า = ใช้ค่าเริ่มต้น): <https://project12569.theeiot.com/EE67172310228/api/v1/alert-cron.php> [บันทึกค่าใหม่](#) [ลบค่าเริ่มต้น](#)

Login เพื่อจัดการ: cron-job.org (บัญชีที่ตั้งไว้ตอนสมัคร) → Jobs → Dashboard

Key ที่ใช้: ไม่ต้องใช้ key อีกต่อไป (X-Admin-Key ถูกถอดออก) – ตั้ง URL เริ่มจากช่องบนสุด (ถ้า URL เริ่มมี ?key= ให้ลบส่วนนี้ทิ้งได้เลย)

ค่าที่ปรากฏในหน้า RUN ปัจจุบันมีดังนี้

รายการ	ค่า / ความหมาย
Job name	GNSS-AlertCheck
Method / ทุก	GET • 5 นาที (300 วินาที)
URL ที่เรียก	ใช้ URL เต็มจากช่องบนสุดของระบบ (ช่องนี้อ่านอย่างเดียว)
URL ค่าเริ่มต้น	https://project12569.theeiot.com/EE67172310228/api/v1/alert-cron.php
Login เพื่อจัดการ	cron-job.org (บัญชีที่ตั้งไว้ตอนสมัคร) → Jobs → Dashboard
Key ที่ใช้	ไม่ต้องใช้ key อีกต่อไป; X-Admin-Key ถูกถอดออก

→ จุดที่ต้องยึดเป็นหลัก

ตอนตั้งค่าใน cron-job.org ให้คัดลอก URL จากช่อง “URL ที่เรียก (พร้อม key)” บนหน้า RUN เพราะช่องนี้คือ URL ที่ระบบประกอบให้พร้อมใช้งานจริง

3. วิธีตั้งค่า cron-job.org ตามหน้า RUN ปัจจุบัน

1. เข้า cron-job.org ด้วยบัญชีที่ใช้ตั้งระบบ แล้วไปที่ Jobs → Dashboard
2. สร้าง Job ใหม่หรือเปิด Job เดิมสำหรับระบบ GNSS โดยตั้งชื่อเป็น GNSS-AlertCheck
3. กำหนด Method เป็น GET
4. กำหนดรอบการเรียกเป็นทุก 5 นาที เพื่อให้ตรงกับค่าที่หน้า RUN แสดง
5. คัดลอก URL จากช่อง “URL ที่เรียก (พร้อม key)” บนหน้า RUN แล้ววางเป็น URL ของ Job
6. ไม่ต้องเพิ่ม X-Admin-Key หรือ Header Key แยก เพราะเวอร์ชันปัจจุบันถอด X-Admin-Key ออกแล้ว
7. บันทึก Job แล้วทดสอบเรียก endpoint หนึ่งครั้ง ก่อนปล่อยให้ทำงานตามรอบอัตโนมัติ
8. เมื่อบันทึกสำเร็จ ไม่ต้องเปิดหน้า Dashboard ของระบบ GNSS ค้างไว้ cron-job.org จะเรียก endpoint จาก Cloud ตามรอบเวลาเอง

✓ ค่าหลักที่ต้องตรง

Job = GNSS-AlertCheck | Method = GET | Schedule = ทุก 5 นาที | URL = URL เต็มจากหน้า RUN

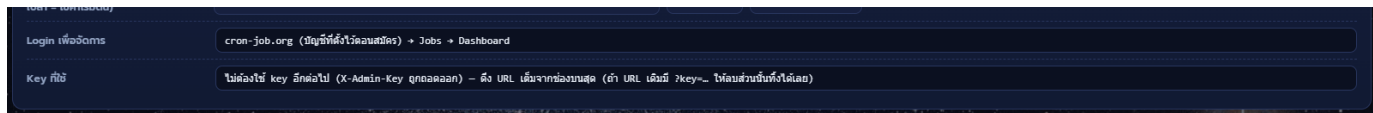
3.1 URL ที่ใช้จริง

หากไม่ได้แก้ค่าเอง ระบบมี URL ค่าเริ่มต้นตามที่หน้า RUN แสดงไว้ หากระบบสร้าง URL เต็มที่มีข้อมูลเพิ่มเติมในช่อง “URL ที่เรียก” ให้ใช้ค่าจากช่องบนสุดนั้นเป็นตัวจริงสำหรับ cron-job.org

4. การเปลี่ยน URL และการใช้ค่าเริ่มต้น

หน้า RUN มีช่อง “แก้ไข URL ที่ใช้จริง” สำหรับเปลี่ยน endpoint โดยไม่ต้องแก้ค่าใน cron-job.org แบบสุ่มหรือคาดเดา

1. หากต้องการใช้ endpoint ใหม่ ให้กรอก URL ลงในช่อง “แก้ไข URL ที่ใช้จริง” แล้วกด “บันทึกค่าใหม่”
2. หากเว้นช่องว่าง ระบบจะกลับไปใช้ค่าเริ่มต้นของโปรเจกต์
3. ปุ่ม “เคลียร์ (ใช้ค่าเริ่มต้น)” ใช้ล้างค่าที่บันทึกเองและกลับไปใช้ URL ค่าเริ่มต้น
4. หลังเปลี่ยน URL ให้ตรวจสอบช่อง “URL ที่เรียก” อีกครั้ง แล้วคัดลอกค่าที่แสดงไปอัปเดต Job ใน cron-job.org



4.1 การใช้ Key - เวอร์ชันปัจจุบัน

รายการ	ค่า / ความหมาย
X-Admin-Key	ไม่ต้องใช้ - ถูกถอดออกจาก flow ปัจจุบัน
Header เพิ่มเติม	ไม่ต้องเพิ่ม Key แยกสำหรับ cron-job.org ตามหน้าที่แนบ
URL เดิมมี ?key=...	ให้ลบส่วน ?key=... ออกได้
URL ที่ควรใช้	ดึง URL เดิมจากช่องบนสุดของหน้า RUN

! สำคัญ

คำแนะนำเรื่อง Cron Key / X-Admin-Key ในคู่มือเดิมไม่สอดคล้องกับหน้าปัจจุบันแล้ว จึงถูกตัดออกจากฉบับแก้ไขนี้

5. การทดสอบและยืนยันว่า Cron ทำงาน

หลังสร้างหรือแก้ไข Job แล้ว ควรทดสอบ endpoint ก่อนใช้งานต่อเนื่อง เพื่อแยกปัญหา URL, HTTPS และ Server ออกจากปัญหาการแจ้งเตือน

1. ใช้ฟังก์ชัน Test Run ของ cron-job.org หรือสั่ง Run Job หนึ่งครั้ง
2. ตรวจสอบว่า request เป็น GET และ URL ตรงกับช่อง “URL ที่เรียก” ในหน้า RUN
3. ผลที่คาดหวังคือ Server ตอบ HTTP 200 OK
4. จากนั้นเปิด History ของ Job และตรวจสอบว่ารอบอัตโนมัติทำงานทุกประมาณ 5 นาทีอย่างต่อเนื่อง
5. หาก HTTP 200 แต่ไม่เกิดการแจ้งเตือน LINE/Telegram ให้ตรวจสอบสถานะ/เงื่อนไข Alert ฟังก์ชัน เพราะการตอบ 200 หมายถึง endpoint ถูกเรียกสำเร็จ ไม่ได้หมายความว่าทุกครั้งต้องมี Alert

รายการ	ค่า / ความหมาย
Method	GET
รอบเวลา	ทุก 5 นาที (300 วินาที)
HTTP ที่คาดหวัง	200 OK
History	ควรมี execution ใหม่ตามรอบเวลา
Dashboard	ไม่ต้องเปิดค้างไว้

i การทดสอบไม่ควรใส่ Key เอง

หาก Job เดิมเคยตั้ง Header เช่น X-Admin-Key หรือใช้ URL ที่มี ?key=... ให้ปรับให้ตรงกับหน้า RUN ปัจจุบันก่อนทดสอบ

6. Troubleshooting เฉพาะส่วน RUN / cron-job.org

อาการ	จุดที่ตรวจ	แนวทางแก้
Job ไม่ทำงานทุก 5 นาที	การตั้งค่า/Server	ตรวจว่า Job เปิดใช้งานและตั้ง schedule เป็น 5 นาทีจริง
HTTP 404	การตั้งค่า/Server	ตรวจ path ของ alert-cron.php และคัดลอก URL จากช่องบนสุดใหม่
HTTP 401 / 403	การตั้งค่า/Server	ตรวจว่า Job เดิมยังส่ง X-Admin-Key หรือ Key เก่าที่ไม่ใช่แล้วหรือไม่ และตรวจสอบสิทธิ์ฝั่ง Server
HTTP 5xx	การตั้งค่า/Server	ปัญหาอยู่ที่ Server/PHP ให้ตรวจ log ของ endpoint
URL เดิมมี ?key=...	การตั้งค่า/Server	ลบส่วน ?key=... และใช้ URL ตามหน้า RUN ปัจจุบัน
เปลี่ยน URL แล้ว Cron ยังเรียกค่าเก่า	การตั้งค่า/Server	บันทึก URL ในหน้า RUN แล้วอัปเดต URL ของ Job ใน cron-job.org ให้ตรงกัน
HTTP 200 แต่ไม่มีข้อความแจ้งเตือน	การตั้งค่า/Server	ตรวจว่า LINE/Telegram เปิดใช้งานและเงื่อนไข Alert มีเหตุให้แจ้งหรือไม่
เปิด Dashboard แล้วจึงทำงาน	การตั้งค่า/Server	การทำงานที่ถูกต้องไม่ต้องพึ่งหน้า Dashboard; cron-job.org ต้องเรียก endpoint ได้จาก Internet โดยตรง

→ ลำดับตรวจที่แนะนำ

ตรวจ 1) URL → 2) Method GET → 3) รอบ 5 นาที → 4) ไม่มี Key เก่า → 5) Test Run/HTTP status → 6) History → 7) เงื่อนไข Alert และ LINE/Telegram

7. Checklist ฉบับใช้งานจริง

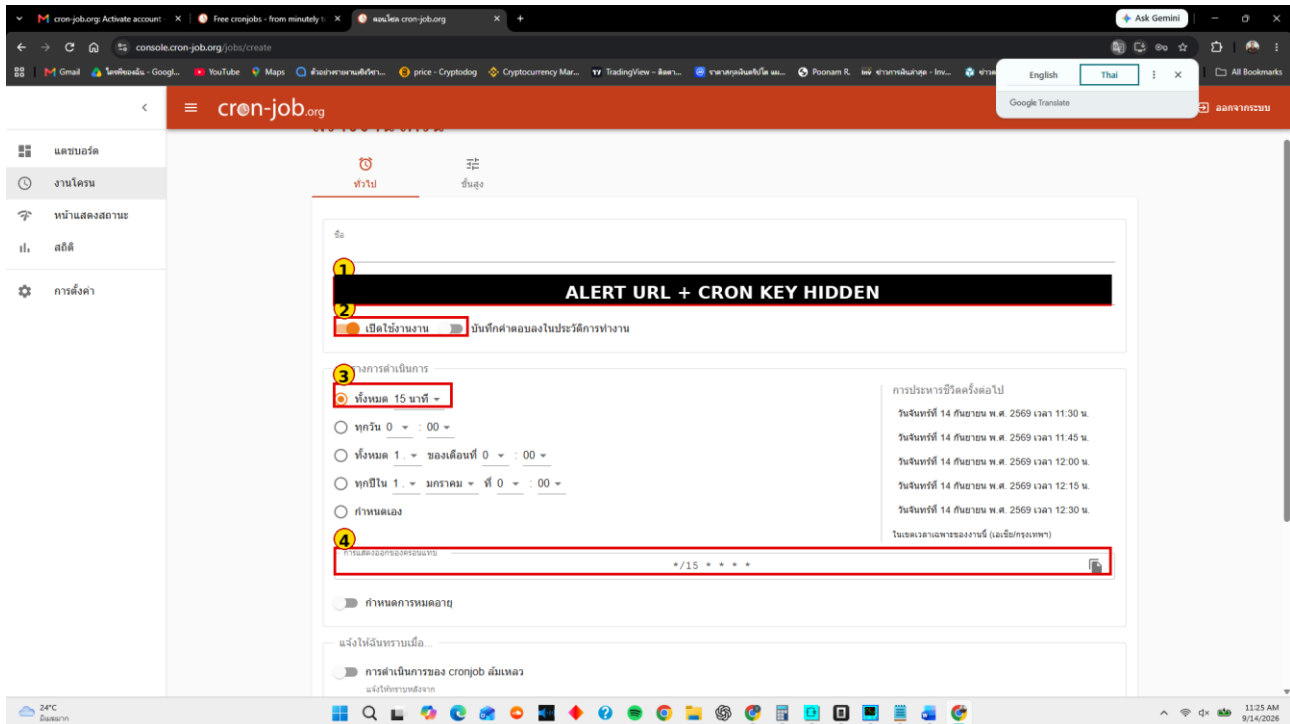
รายการ	ค่า / ความหมาย
หน้า RUN เปิดได้	ตรวจเห็นส่วน cron-job.org และค่าปัจจุบัน
Job name	GNSS-AlertCheck
Method	GET
Schedule	ทุก 5 นาที (300 วินาที)
URL	คัดลอกจาก “URL ที่เรียก” บนหน้า RUN
X-Admin-Key	ไม่ใช้งาน
URL ?key=...	ไม่ใช่; ลบจาก Job เดิมหากยังมี
Test Run	HTTP 200 OK
History	มีรอบอัตโนมัติต่อเนื่อง
Dashboard	ไม่ต้องเปิดค้างไว้
LINE / Telegram	เปิดใช้งานและ Configured ตามหน้าตั้งค่าแจ้งเตือน

✓ ผลลัพธ์สุดท้าย

เมื่อค่าข้างต้นครบ cron-job.org จะเรียก endpoint ฟังก์ชัน Server ทุก 5 นาที และ engine จะตรวจสอบสถานะ/แจ้งเตือนแจ้งเตือนโดยอัตโนมัติ แม้ไม่มีผู้ใช้เปิดหน้าเว็บค้างไว้

8. การสร้าง Cron Job สำหรับตรวจสอบระบบ Alert

ต้องมี Scheduler ที่เรียกว่า alert-cron.php เป็นระยะ เพื่อให้ Server ตรวจสอบข้อมูลล่าสุดและแจ้งเตือน Alert แม้ไม่มีผู้ใช้เปิดหน้าเว็บ



ภาพที่ 11 หน้าสร้าง Cron Job ในช่วงแรกที่ยังตั้งทุก 15 นาที

กรอบ 1: URL ของ alert-cron.php โดย Secret Key ถูกปิดบัง

กรอบ 2: เปิดใช้งาน Job

กรอบ 3: รอบเดิม Every 15 minutes ซึ่งภายหลังปรับเป็น 5 นาที

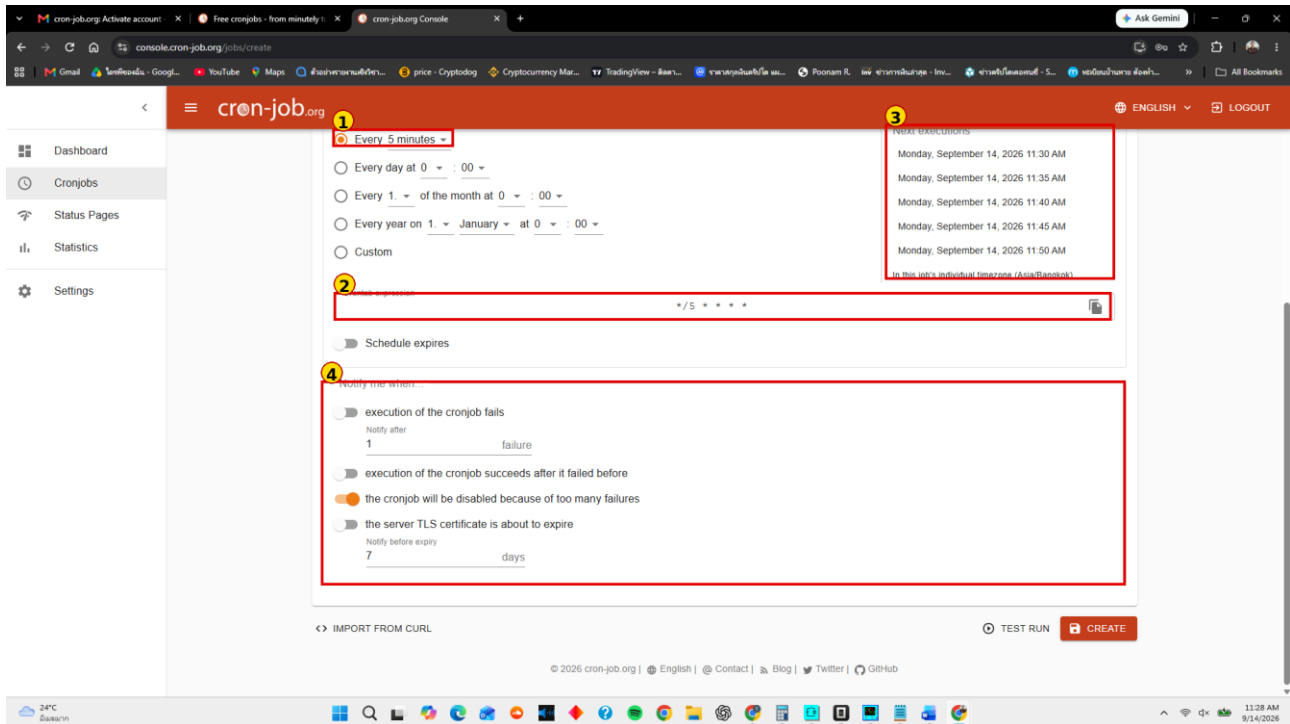
กรอบ 4: Crontab expression ของรอบเดิม

รอบ 15 นาทีใช้งานได้ แต่ทำให้การตรวจเหตุผิดปกติช้าได้ถึงประมาณ 15 นาที ขณะที่ ESP32 สรุปรู Measurement เป็นรอบ 5 นาที จึงปรับ Cron ให้สอดคล้องเป็นทุก 5 นาที

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

รายการ	ค่า/ความหมาย
Alert URL	[HIDDEN]
HTTP method	GET
Enabled	เปิด
Schedule expires	ปิด

9. ตั้ง Cron Job ให้ทำงานทุก 5 นาที



ภาพที่ 12 ค่าตารางเวลาที่ใช้งานจริง

กรอบ 1: เลือก Every 5 minutes

กรอบ 2: Cron expression = `* / 5 * * * *`

กรอบ 3: Next executions แสดงรอบ 11:30, 11:35, 11:40, ... และ Timezone Asia/Bangkok

กรอบ 4: ส่วน Notify me when... ใช้เปิดแจ้งเตือน failure/recovery/TLS ตามต้องการ

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

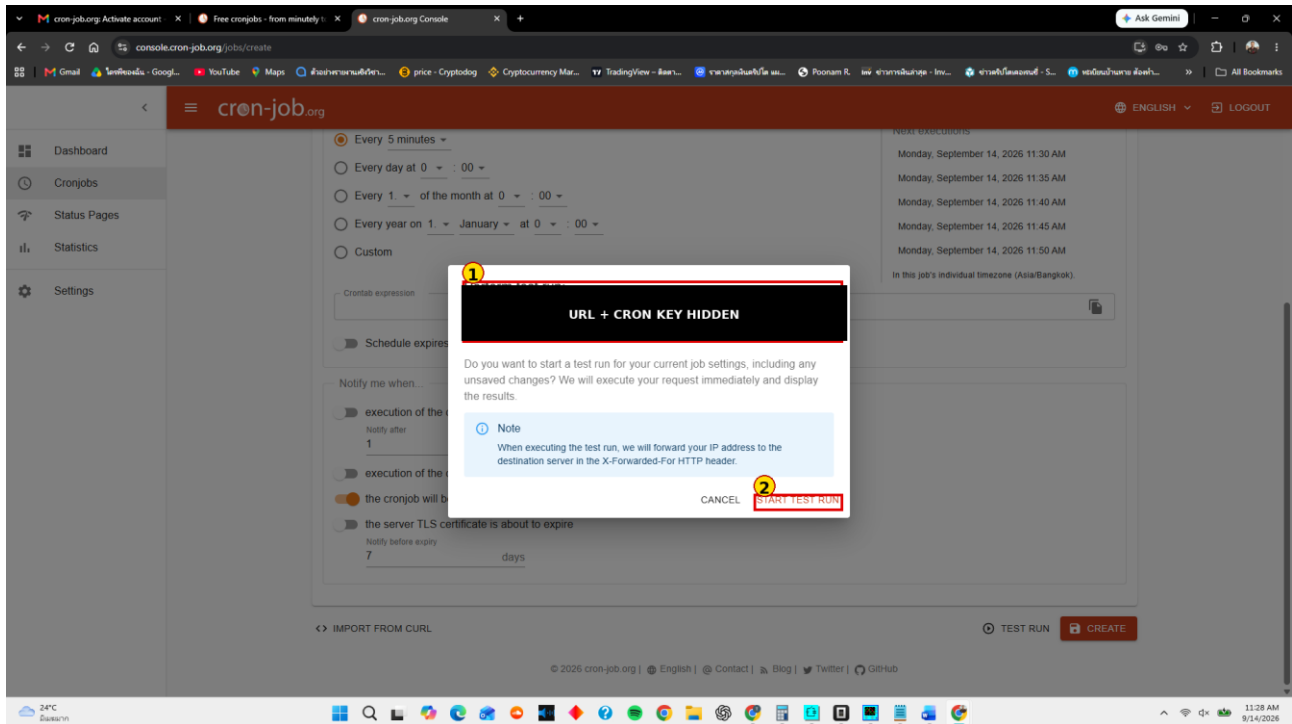
รายการ	ค่า/ความหมาย
Schedule	Every 5 minutes
Cron expression	<code>* / 5 * * * *</code>
Timezone	Asia/Bangkok
ตัวอย่างรอบ	11:30, 11:35, 11:40, 11:45, 11:50, ...
Schedule expires	Off

การแจ้งเตือนที่แนะนำใน cron-job.org ได้แก่ แจ้งเมื่อ execution ล้มเหลว, แจ้งเมื่อกลับมาสำเร็จหลังเคยล้มเหลว และแจ้งเมื่อ TLS certificate ใกล้หมดอายุ

เหตุผลที่เลือก 5 นาที: ให้ Scheduler ตรวจสอบ Alert ใน cadence เดียวกับ Measurement interval 300 วินาที ทำให้ตรวจสอบข้อมูลใหม่ได้เร็วโดยไม่เรียก Server เกินความจำเป็น

10. ทดสอบ Cron ก่อนสร้างจริงด้วย TEST RUN

ก่อนกด CREATE ให้กด TEST RUN เพื่อทดสอบ URL, HTTPS และการตอบกลับของ Server ทันที โดยไม่ต้องรอถึงรอบเวลา



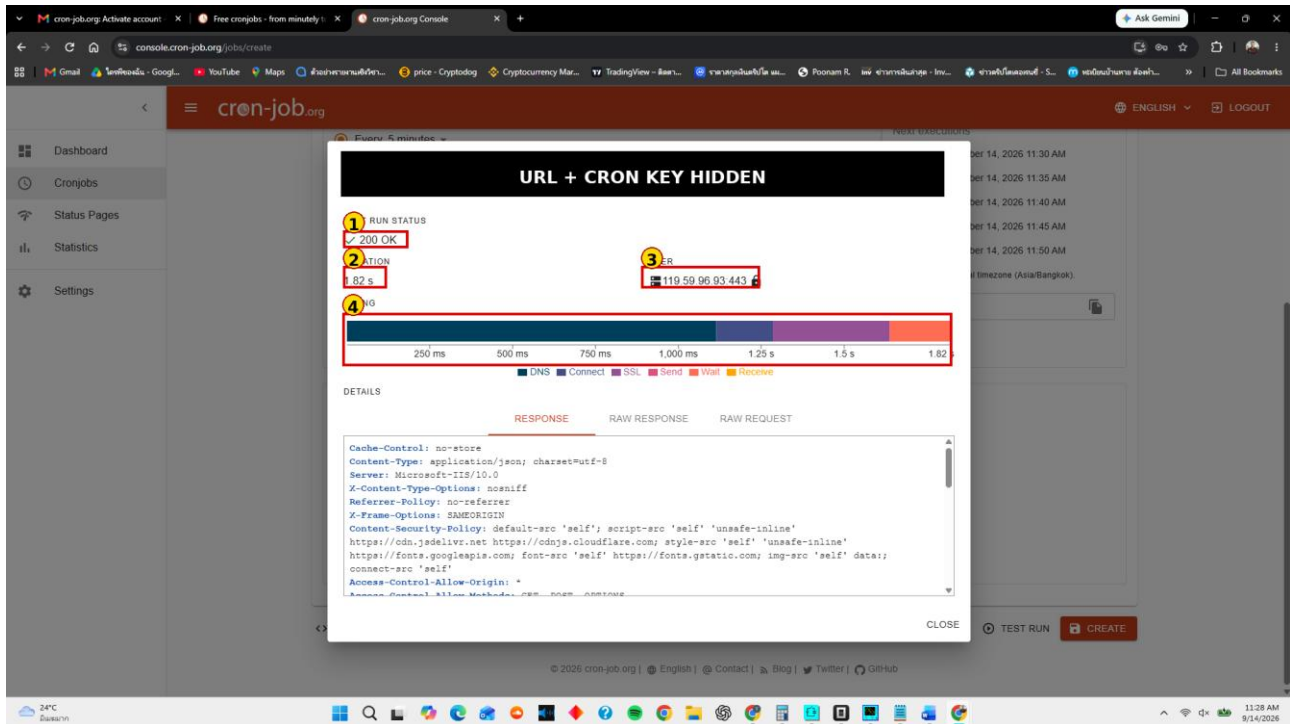
ภาพที่ 13 หน้าต่างยืนยันการทำ Test Run

กรอบ 1: cron-job.org แสดง URL ที่จะเรียก (Secret Key ถูกปิดบัง)

กรอบ 2: กด START TEST RUN เพื่อส่งคำขอทันที

ข้อความ X-Forwarded-For: cron-job.org แจ้งว่าจะส่ง IP ของผู้ทดสอบไปใน HTTP header X-Forwarded-For ซึ่งเป็นข้อมูลประกอบของ request ไม่ใช่ Error

11. อ่านผล Test Run: HTTP 200 OK



ภาพที่ 14 ผล Test Run สำเร็จ

กรอบ 1: TEST RUN STATUS = 200 OK

กรอบ 2: Duration ประมาณ 1.82 s

กรอบ 3: Peer ปลายทางเชื่อมต่อผ่านพอร์ต 443

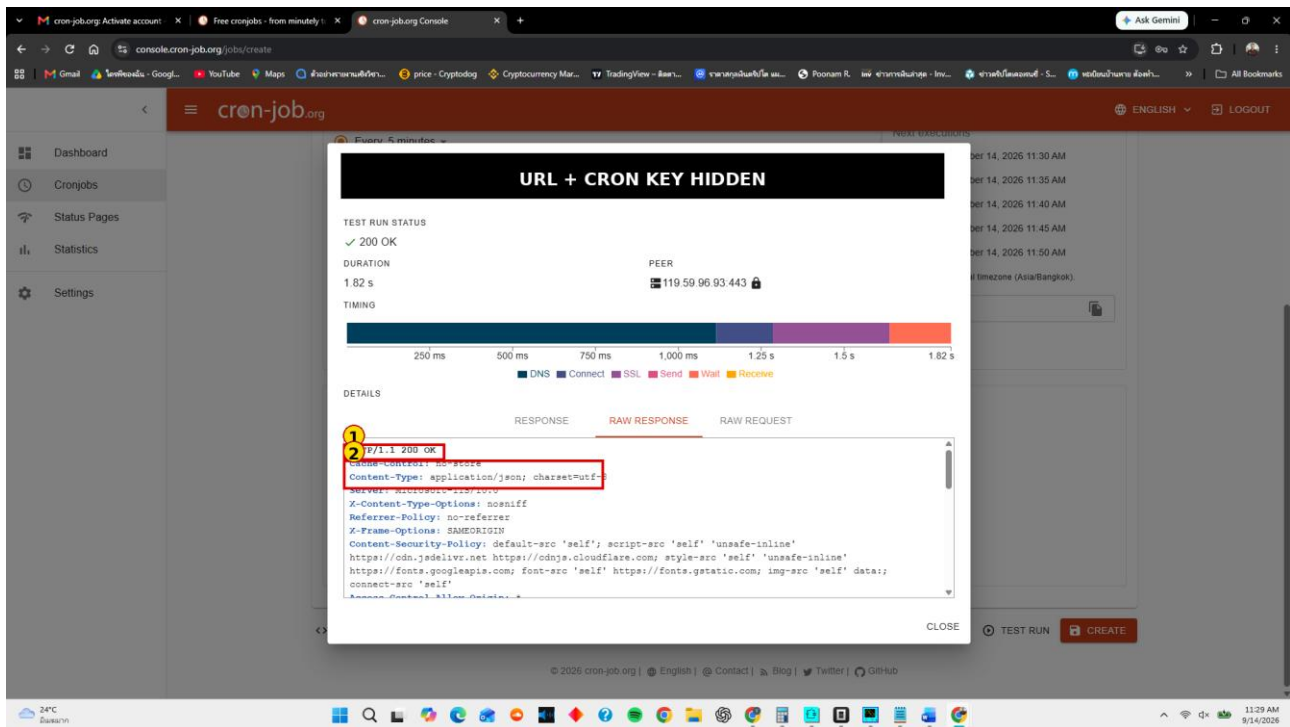
กรอบ 4: Timing bar แสดง DNS, Connect, SSL, Send, Wait และ Receive

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

รายการ	ค่า/ความหมาย
HTTP status	200 OK
Duration	ประมาณ 1.82 วินาที ในรอบตัวอย่าง
Peer port	443
ความหมาย	cron-job.org ติดต่อ Server และเรียก endpoint ได้สำเร็จ

HTTP 200 ยืนยันระดับ Transport/Application response ว่า endpoint ตอบกลับ แต่เพื่อยืนยัน Logic ภายใน PHP ควรอ่าน JSON body ต่อด้วย

12. ตรวจ RAW RESPONSE



ภาพที่ 15 Header ที่ Server ตอบกลับ

กรอบ 1: บรรทัด HTTP/1.1 200 OK

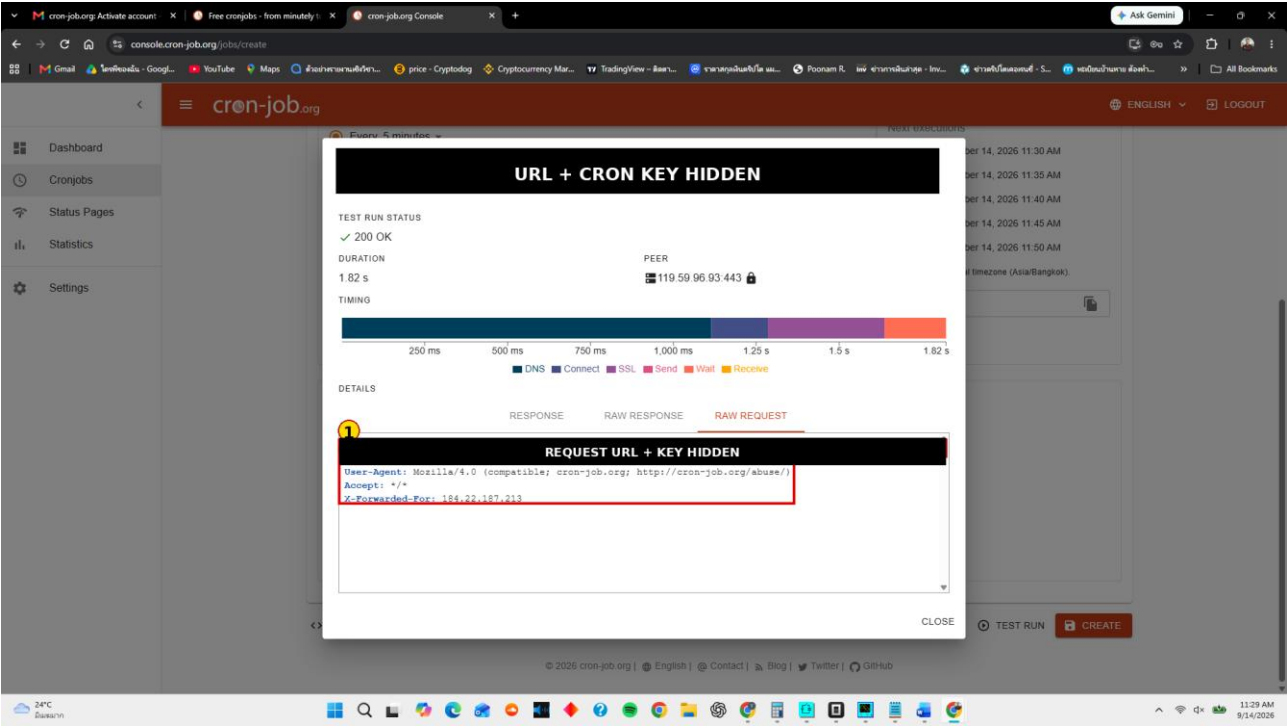
กรอบ 2: Header สำคัญ เช่น Content-Type และ Server

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

รายการ	ค่า/ความหมาย
HTTP/1.1	200 OK
Content-Type	application/json; charset=utf-8
Server	Microsoft-IIS/10.0
Cache-Control	no-store

จาก Header นี้สามารถยืนยันได้ว่า Server ตอบเป็น JSON และการเชื่อมต่อกับ Web Server ทำงานปกติ หาก Cron ได้ 200 แต่ระบบไม่แจ้งเตือน ให้เลื่อนดู JSON body ของ alert-cron.php เพื่อเช็คจำนวน record ที่ตรวจและจำนวน alert ที่สร้าง

13. ตรวจ RAW REQUEST



ภาพที่ 16 Request ที่ cron-job.org ส่งไปยัง Server

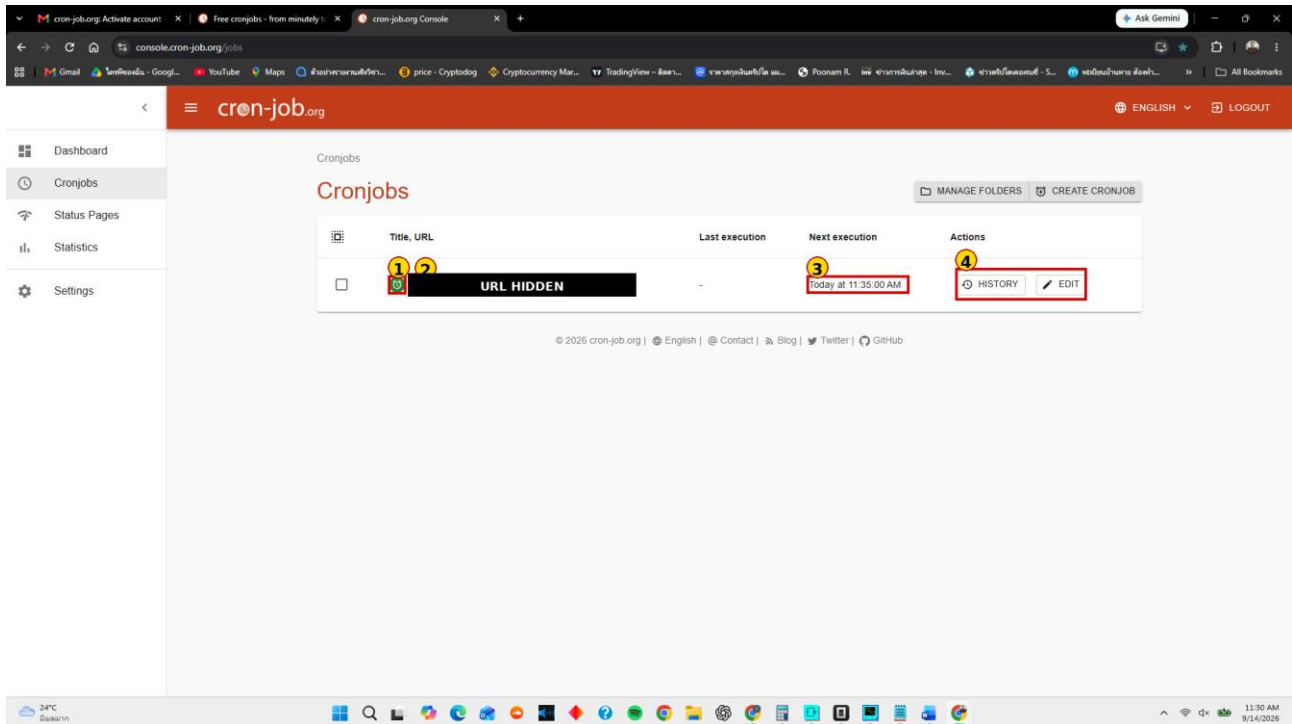
- กรอบ 1: GET path ของ alert-cron.php โดย Secret Key ถูกปิดบัง
- กรอบ 2: Host, User-Agent และ X-Forwarded-For ที่ถูกส่งมากับ Request

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

รายการ	ค่า/ความหมาย
Method	GET
Path	[HIDDEN]
Host	[HIDDEN]
Header	X-Forwarded-For มีค่า IP ของต้นทางทดสอบ

ด้านความปลอดภัย: Secret Key ไม่ควรใส่ในภาพ/รายงานที่เผยแพร่สาธารณะ หากเคยเผยแพร่ Key จริงควร Rotate เป็นค่าใหม่ แล้วแก้ URL ใน cron-job.org ให้ตรงกัน

14. กด CREATE และตรวจว่า Cron Job เปิดใช้งานจริง



ภาพที่ 17 Cron Job หลังสร้างสำเร็จ

กรอบ 1: ไอคอนสีเขียวแสดงว่า Job เปิดใช้งาน

กรอบ 2: URL ของ Job แสดงในรายการ ที่ต้องนำค่านี้ไปกรอกบนหน้า Dashboard

กรอบ 3: Next execution แสดงเวลารอบถัดไป

กรอบ 4: ปุ่ม HISTORY ใช้ดูผลแต่ละรอบ และ EDIT ใช้แก้ไขค่า

ค่าที่ใช้/ค่าที่ต้องตรวจสอบ

รายการ	ค่า/ความหมาย
Job status	Enabled / Active
Next execution	ควรขยับไปทุก 5 นาที
Last execution	ก่อนรอบแรกอาจเป็น “-”; หลังทำงานแล้วต้องมีเวลา
History	ควรเห็น HTTP 200 ต่อเนื่อง

หลังจากเวลา Next execution ให้ Refresh หน้า Cronjobs แล้วกด HISTORY เพื่อตรวจว่ารอบอัตโนมัติ (ไม่ใช่ Test Run) ทำงานได้จริง