

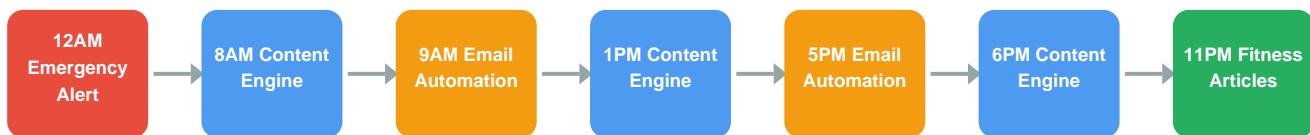
Social Media Empire

Complete Workflow Documentation

Generated: February 06, 2026

System Overview			
Active Workflows	9	Brands	3
Daily Pins	7	Python Modules	6+
Weekly Articles	5-10	APIs Used	8+
Supabase Tables	6+	GitHub Secrets	15+

DAILY SCHEDULE (PST)



WEEKLY SCHEDULE (PST)



Table of Contents

#	Workflow	Description	
1	Content Engine	3x daily pin generation + posting for all 3 brands	
2	Weekly Trend Discovery	Sunday night trend analysis + article generation	
3	Fitness & Deals Articles	Mon-Fri SEO article generation for fitover35.com	
4	System Health	Every 6h health checks + self-healing + alerts	
5	Email Automation	2x daily email sequences + subscriber sync	
6	Weekly Maintenance	Sunday morning weekly summary + email report	
7	Emergency Alert	Dead man's switch — alerts if content engine stops	
8	Auto Merge	Auto-merges claude/* branches to main	
9	ToolPilot Deploy	Deploys ToolPilot site to Netlify on push	

1. Content Engine

File: <.github/workflows/content-engine.yml> | Schedule: 3x daily (8AM, 1PM, 6PM PST) | Timeout: 30 min

The Content Engine is the heart of the system. It runs 3 times per day and generates 7 Pinterest pins across all 3 brands. Each run determines which brands to post for based on the time slot, then for each brand: generates unique pin content via Claude AI, fetches a relevant Pexels image, builds a UTM-tagged destination URL, and posts via the brand's designated platform (Late API or Make.com webhook).

Time Slot Distribution

Time (PST)	Slot Name	Brands Posted	Pins
8:00 AM	Morning	Fitness + Deals + Menopause	3
1:00 PM	Afternoon	Fitness + Deals	2
6:00 PM	Evening	Fitness + Menopause	2
		TOTAL PER DAY	7

Step-by-Step Flow



Posting Methods by Brand

Brand	Platform	API Method	Secrets Used
Fitness Made Easy	Late API	POST getlate.dev/api/v1/posts	LATE_API_KEY, LATE_API_KEY_3
Daily Deal Darling	Make.com S1	POST to webhook URL	MAKE_WEBHOOK DEALS
Menopause Planner	Make.com S2	POST to webhook URL	MAKE_WEBHOOK MENOPAUSE

Key Python Modules

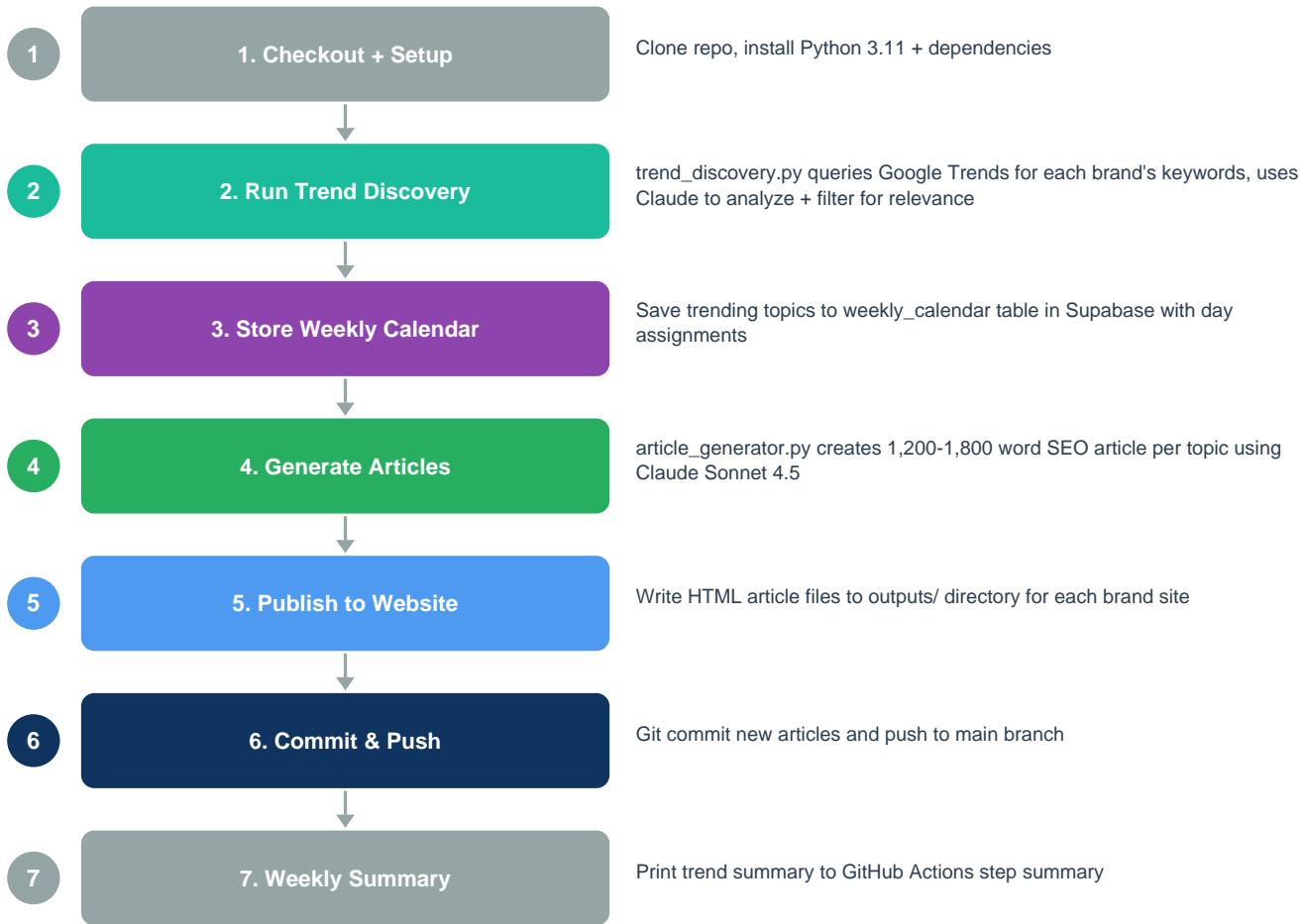
- **content_brain.py** — Claude Sonnet 4.5 generates pin title, description, hashtags, image search query. Checks content_history in Supabase to avoid repeating topics, angles, visual styles, and description openers. Calendar-aware: pulls from weekly_calendar if available.
- **image_selector.py** — Queries Pexels API with brand-specific guardrails. Each brand has allowed_themes and blocked_terms. Validates queries before sending. Tracks used images to prevent repeats.
- **supabase_client.py** — Wrapper for all database operations. Logs pins to content_history, errors to errors table, reads recent history for uniqueness checks.

2. Weekly Trend Discovery

File: [.github/workflows/weekly-discovery.yml](#) | Schedule: Sunday 10PM PST | Timeout: 45 min

Runs every Sunday night to discover what's trending in each brand's niche for the upcoming week. Uses Google Trends + Claude AI analysis to identify 5-7 trending topics per brand, then generates full SEO articles for each topic. Articles are committed to the repo and published to brand websites BEFORE the pins are posted during the week.

Step-by-Step Flow



Key Python Modules

- **trend_discovery.py** — Uses pytrends library to query Google Trends with brand-specific keywords (10 per brand). Claude analyzes raw trend data and selects 5-7 most relevant topics. Each brand has a relevance_filter to ensure on-niche results.
- **article_generator.py** — Claude Sonnet 4.5 writes full articles with: H1 title, meta description, 4-6 H2 sections, email capture CTA, 2-3 affiliate product recommendations. Loads affiliate_config.json for brand-specific product links.

Data Flow



3. Fitness & Deals Articles

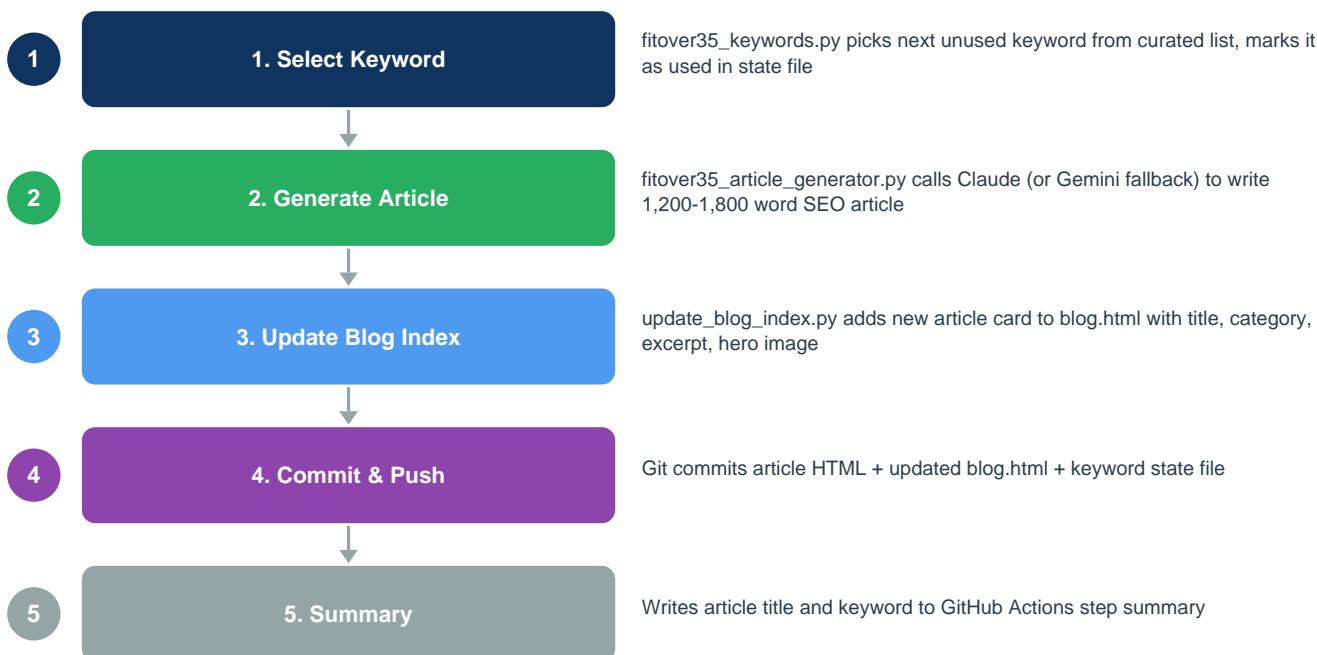
File: <.github/workflows/fitness-articles.yml> | Schedule: Mon-Fri 11PM PST | Timeout: 25 min

Generates one SEO article per weekday for fitover35.com. Selects a keyword from a curated list, generates a full article via Claude (with Gemini fallback), updates the blog index page, and commits everything to the repo. This is the primary SEO content engine — designed to build organic Google traffic over 3-6 months.

Two Parallel Jobs

Job	Runs When	What It Does
generate-fitover35-article	Every schedule run + manual (fitover35/both)	Placeholder — not yet implemented
generate-deals-article	Manual only (deals/both) + Mon/Wed/Fri	Placeholder — not yet implemented

FitOver35 Article Generation Steps



4. System Health

File: <.github/workflows/system-health.yml> | Schedule: Every 6 hours | Timeout: 15 min

The immune system of the platform. Runs every 6 hours to verify all API integrations are working, then conditionally triggers self-healing (retry failed pins, clean old errors) and alerts (email notification if system is unhealthy).

Three Chained Jobs



Health Check — APIs Tested

API	Test Method	Failure Impact
Anthropic (Claude)	Create test message with "ping"	DEGRADED — pins won't generate
Pexels	GET search endpoint	DEGRADED — no images for pins
Supabase	Initialize client connection	UNHEALTHY — no logging, no history
Late API	GET /accounts endpoint	DEGRADED — fitness pins won't post

Self-Healing Actions

- Retry failed pins** — Queries errors table for content_engine failures from last 24h, attempts to re-post
- Clean old errors** — Deletes resolved errors older than 30 days from Supabase
- Alert email** — Sends "UNHEALTHY" email via Resend to ALERT_EMAIL if Supabase is down

5. Email Automation

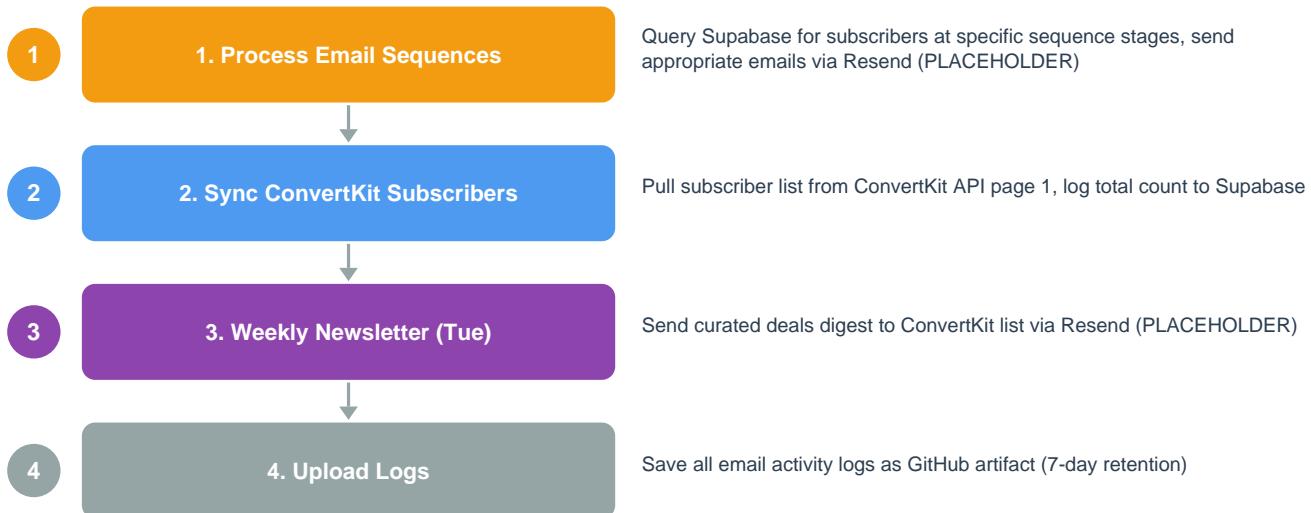
File: <.github/workflows/email-automation.yml> | Schedule: 2x daily (9AM, 5PM PST) | Timeout: 15 min

Handles email marketing operations: processes welcome sequences for new subscribers, syncs ConvertKit subscriber data to Supabase, and sends a weekly deals newsletter on Tuesdays. Note: Some steps are still placeholder implementations.

Two Jobs

Job	Schedule	Status	What It Does
email-automation	9AM + 5PM daily	Partial	Process sequences (placeholder) + sync ConvertKit subscribers
weekly-newsletter	Tuesdays 9AM only	Placeholder	Send weekly deals newsletter via ConvertKit

Email Automation Steps



Key Python Modules

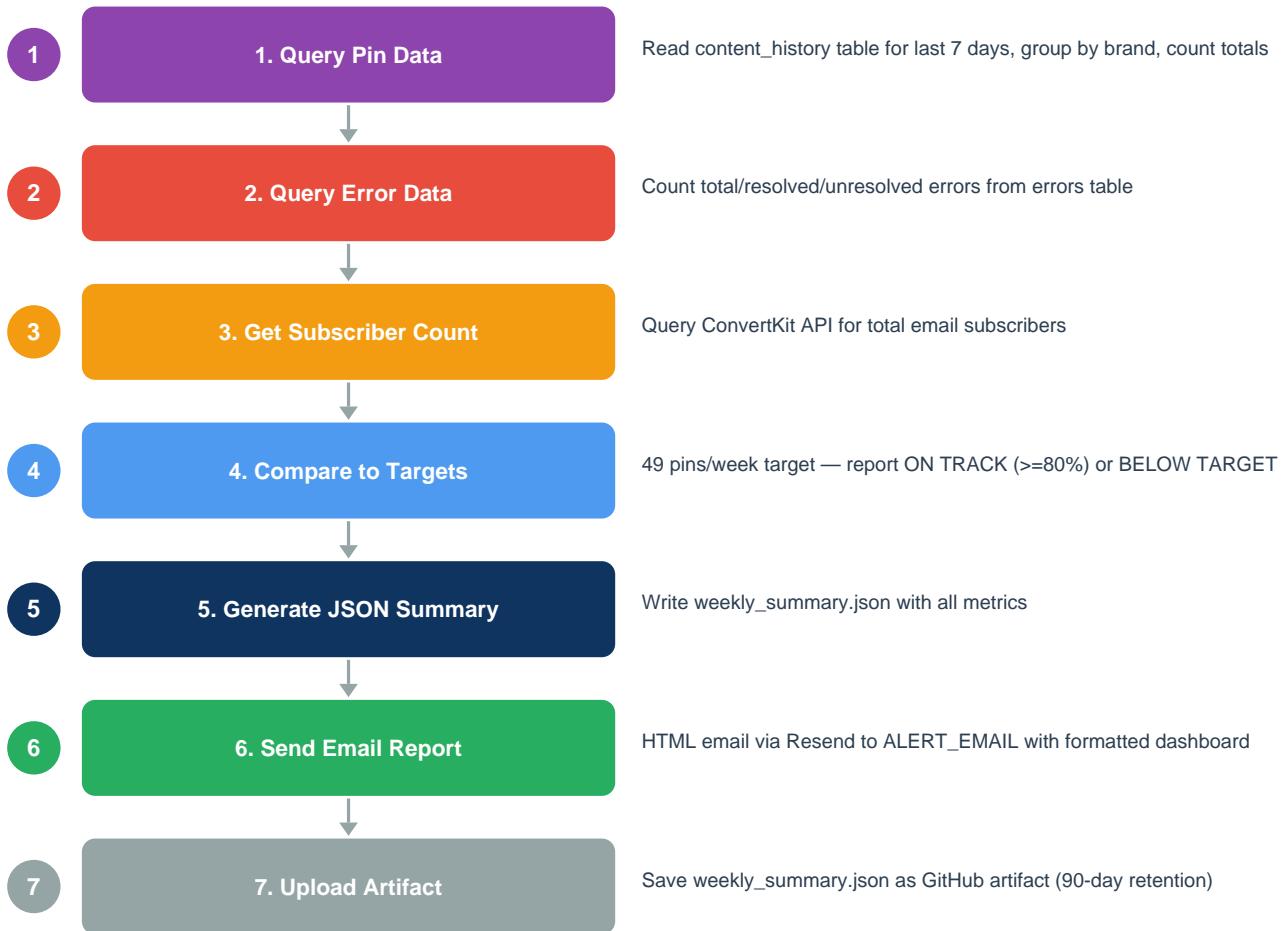
- `email_sender.py` — Resend API wrapper for sending transactional and marketing emails
- `convertkit_automation.py` — ConvertKitManager class for subscriber management, list operations, and tag management

6. Weekly Maintenance

File: <.github/workflows/weekly-maintenance.yml> | Schedule: Sunday 9AM PST | Timeout: 20 min

Generates a weekly performance summary and emails it to you. Aggregates all pin posting data, error counts, and subscriber metrics from the past 7 days. Compares against the target of 49 pins/week (7 pins/day x 7 days) and reports whether the system is on track.

Step-by-Step Flow



7. Emergency Alert (Dead Man's Switch)

File: <.github/workflows/emergency-alert.yml> | Schedule: Daily at Midnight PST | Timeout: 5 min

A dead man's switch that checks whether the Content Engine (the most critical workflow) has run successfully in the last 24 hours. If it hasn't, sends an emergency email alert. Uses a 26-hour threshold (2-hour buffer) to avoid false positives from schedule drift.

Decision Flow



How It Works

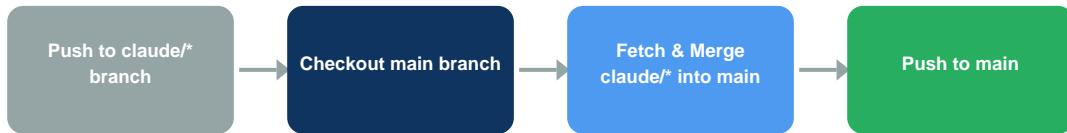
- Uses **gh api** to query the last successful content-engine.yml workflow run
- Calculates hours since that run using Unix epoch timestamps
- If > 26 hours (or no runs found), triggers alert via Resend email
- Alert includes direct link to GitHub Actions page for immediate debugging
- The `|| true` on the Python call ensures the workflow itself never fails

8. Auto Merge to Main

File: [.github/workflows/auto-merge.yml](#) | Trigger: Push to claude/* branches | No timeout

Automatically merges branches matching `claude/check-github-automation-*` or `claude/merge-to-main-*` into the main branch. This enables Claude Code agents to push changes to feature branches that get auto-merged without manual intervention.

Step-by-Step Flow



Details

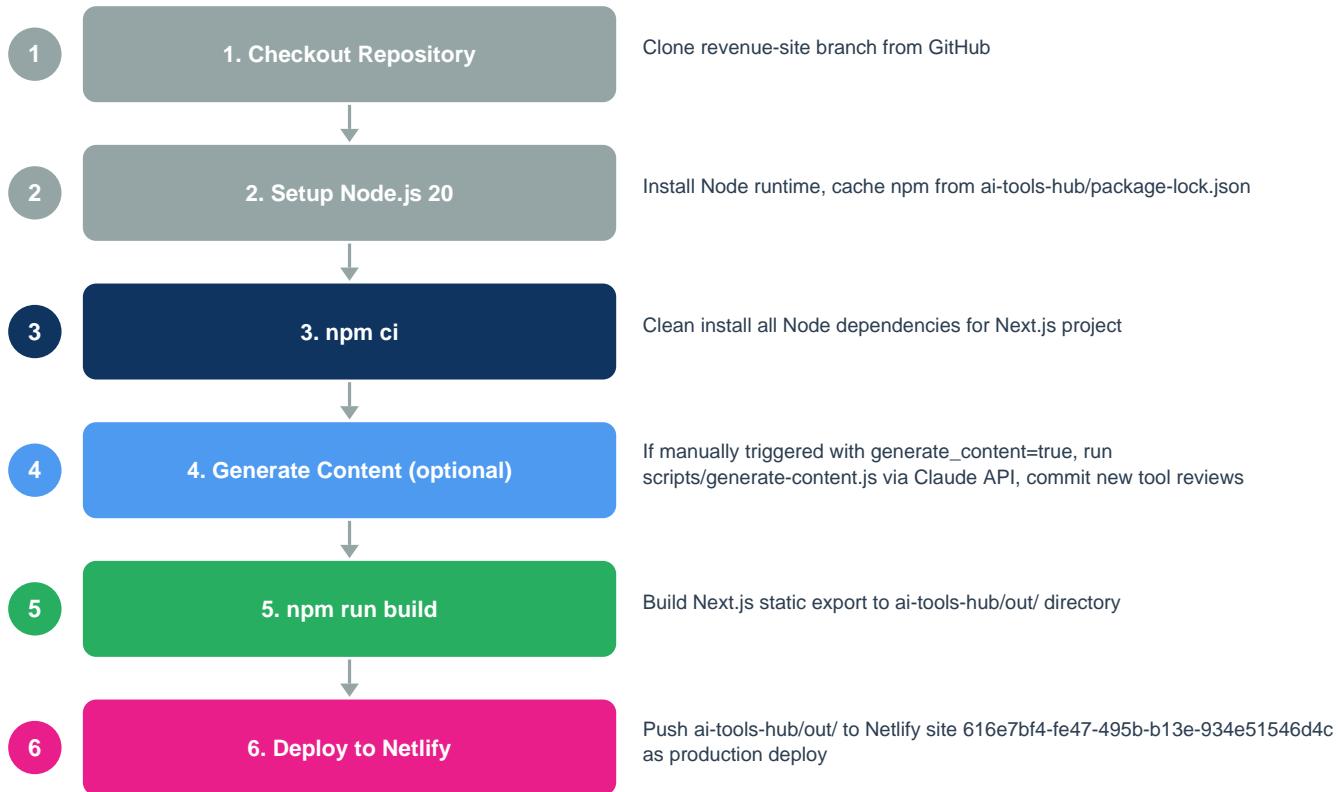
- Uses `fetch-depth: 0` for full git history (needed for clean merges)
- Commits as `github-actions[bot]` with standard bot email
- Merge uses `--no-edit` flag with custom merge commit message
- Requires `contents: write` and `pull-requests: write` permissions

9. ToolPilot Deploy

File: [.github/workflows/toolpilot-deploy.yml](#) | Trigger: Push to revenue-site branch (ai-tools-hub/**) | No timeout

Builds and deploys the ToolPilot AI tools directory website to Netlify. ToolPilot is a Next.js 14 static site with AI tool reviews, comparisons, and affiliate links. Deployed to [toolpilot-hub.netlify.app](#). Optionally generates new AI tool review content via Claude.

Step-by-Step Flow



Appendix: GitHub Secrets Reference

Secret Name	Used By	Purpose
ANTHROPIC_API_KEY	Content Engine, Discovery, Articles, Health	Claude AI content generation
PEXELS_API_KEY	Content Engine, Discovery, Articles, Health	Stock image search
SUPABASE_URL	All workflows except Auto Merge	Database connection URL
SUPABASE_KEY	All workflows except Auto Merge	Database authentication key
LATE_API_KEY	Content Engine, Health	Pinterest posting (Fitness brand) — primary key
LATE_API_KEY_3	Content Engine, Health	Pinterest posting — fallback key on 403
MAKE_WEBHOOK DEALS	Content Engine	Make.com Scenario 1 webhook (Daily Deal Darling)
MAKE_WEBHOOK_MENOPAUSE	Content Engine	Make.com Scenario 2 webhook (Menopause Planner)
PINTEREST_FITNESS_ACCOUNT_ID	Content Engine	Pinterest account ID for Late API
PINTEREST_FITNESS_BOARD_ID	Content Engine	Pinterest board ID for Late API
RESEND_API_KEY	Health, Email, Maintenance, Emergency	Transactional email sending
ALERT_EMAIL	Health, Maintenance, Emergency	Email address for system alerts
CONVERTKIT_API_KEY	Email, Maintenance, Health	Email marketing subscriber management
CONVERTKIT_API_SECRET	Email	ConvertKit API authentication
GEMINI_API_KEY	Articles (fallback)	Fallback AI if Anthropic key missing
CREATOMATE_API_KEY	Health (check only)	Video rendering (legacy, health check only)
NETLIFY_API_TOKEN	ToolPilot Deploy	Netlify deployment authentication

Appendix: Supabase Tables Used

Table Name	Used By	Purpose
content_history	Content Engine, Health, Maintenance	Tracks every pin posted (brand, title, image, status, timestamp)
errors	Content Engine, Health, Maintenance	Error logging with severity, type, resolution status
weekly_calendar	Weekly Discovery, Content Engine	Trending topics assigned to days of the week per brand
generated_articles	Weekly Discovery, Articles	Tracks generated articles with slugs, titles, publish status
pinterest_pins	Legacy (content_history replaces)	Original pin tracking table
email_sends	Email Automation	Tracks sent emails with recipient, subject, status