

DevBlock Snitch Worker Manual

Heartbeat-loss monitoring, cooldown logic, and escalation failover

The snitch-worker is a deliberately small Cloudflare Worker that records heartbeats, checks them on a schedule, and escalates outages through Telegram with email failover if the main system stops reporting in.

DevBlock Snitch Worker Manual Snapshot

A quick structural reading of the project surface

The snitch-worker is a deliberately small Cloudflare Worker that records heartbeats, checks them on a schedule, and escalates outages through Telegram with email failover if the main system stops reporting in.

Identity

- This project is a safety net, not a general platform service.
- Its value comes from simplicity, isolation, and independence...
- It stores just enough state to know when heartbeats disa...

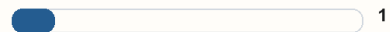
Stack

- Cloudflare Worker
- KV persistence
- Cron trigger
- Telegram alerting

Http Paths



Scheduled Jobs



Kv Accesses



Alert Paths



Cooldowns



DevBlock Snitch Worker Manual visual snapshot

What this project is for

The snitch-worker is a deliberately small Cloudflare Worker that records heartbeats, checks them on a schedule, and escalates outages through Telegram with email failover if the main system stops reporting in.

Primary audience: Use this when changing uptime monitoring, alerting thresholds, cooldowns, or heartbeat semantics.

How to identify it quickly

- This project is a safety net, not a general platform service.
- Its value comes from simplicity, isolation, and independence from the main application stack.
- It stores just enough state to know when heartbeats disappear and alerts need to fire.
- The worker's logic should remain easy to audit because it exists for failure scenarios.

Technology stack

- Cloudflare Worker
- KV persistence
- Cron trigger
- Telegram alerting
- Email failover

Structural counts

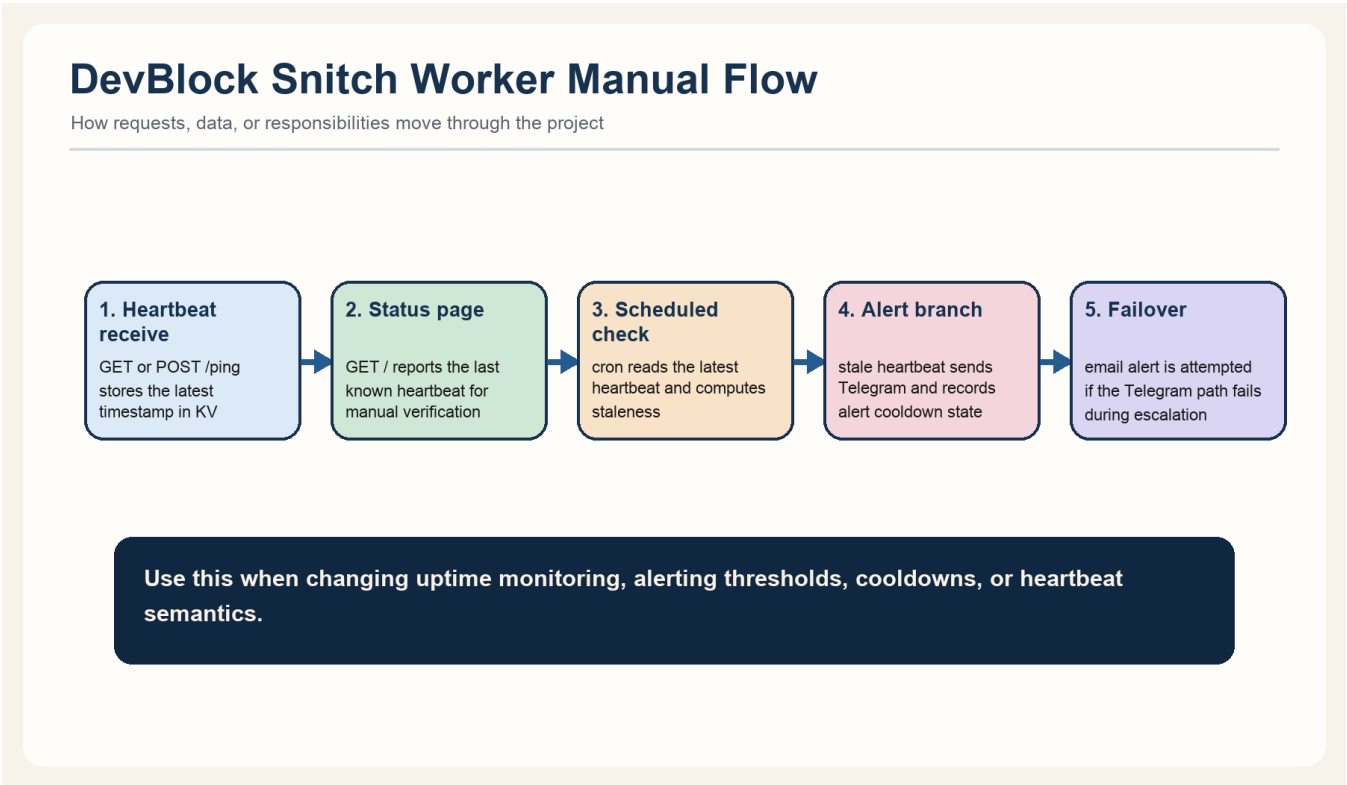
Metric	Count
Http Paths	2
Scheduled Jobs	1
Kv Accesses	8
Alert Paths	7
Cooldowns	2

Architecture reading

- The worker is intentionally tiny so it stays trustworthy when the larger system fails.
- Its cooldown logic is as important as the alert path because paging loops destroy signal quality.
- Independence from the main application stack is part of its architecture, not just a deployment convenience.
- Manual status inspection through ``/`` is useful because failure tooling must support human verification.

Core flow

This sequence is the shortest useful mental model for how the project behaves at runtime or during operator use.



DevBlock Snitch Worker Manual responsibility or data flow

Responsibility matrix

DevBlock Snitch Worker Manual Responsibility Matrix

What this project owns versus what it coordinates

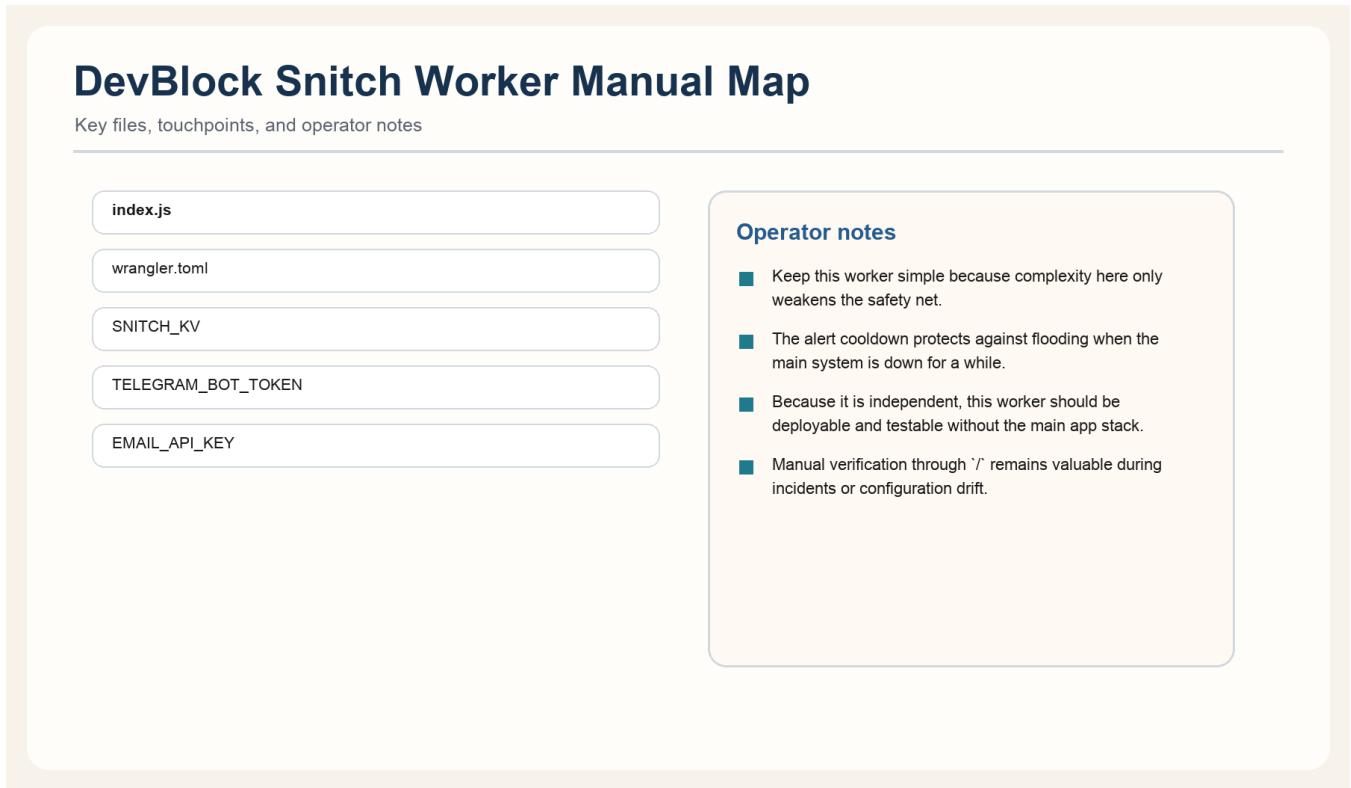
Area	Owns	Coordinates with	Failure mode
Product role	This project is a safety net, not a general platform service.	Its value comes from simplicity, isolation, and independence from the main application stack.	Alerting depends on external provider credentials and network reachability.
Main inputs	Heartbeat receive: GET or POST /ping stores the latest timestamp in KV	Status page: GET / reports the last known heartbeat for manual verification	If heartbeat semantics change in the main system, the snitch threshold must move with them.
Change surface	Read the full worker file before editing because its behavior is compact and tightly coupled.	Treat heartbeat timing, stale thresholds, and cooldown values as operational policy changes.	Minimal code is a strength here; turning it into a general monitoring service would weaken it.
Operator posture	Review stale-threshold and cooldown changes as incident-policy changes, not just	Test both the normal heartbeat path and the degraded failover path whenever alert logic changes.	Keep this worker simple because complexity here only weakens the safety net.

DevBlock Snitch Worker Manual ownership matrix

Key files to read first

If you are new to this project, read these files in order before making broad edits.

- index.js
- wrangler.toml
- SNITCH_KV
- TELEGRAM_BOT_TOKEN
- EMAIL_API_KEY



The image shows a screenshot of a manual orientation map for the DevBlock Snitch Worker. It features a title 'DevBlock Snitch Worker Manual Map' and a subtitle 'Key files, touchpoints, and operator notes'. On the left, there is a vertical list of five key files: index.js, wrangler.toml, SNITCH_KV, TELEGRAM_BOT_TOKEN, and EMAIL_API_KEY. On the right, there is a section titled 'Operator notes' containing four bullet points: 'Keep this worker simple because complexity here only weakens the safety net.', 'The alert cooldown protects against flooding when the main system is down for a while.', 'Because it is independent, this worker should be deployable and testable without the main app stack.', and 'Manual verification through `f` remains valuable during incidents or configuration drift.'

DevBlock Snitch Worker Manual orientation map

Change workflow

- Read the full worker file before editing because its behavior is compact and tightly coupled.
- Treat heartbeat timing, stale thresholds, and cooldown values as operational policy changes.
- If you change alert providers or secrets, preserve the primary-versus-failover simplicity.
- Avoid feature creep; the best improvement is usually clearer monitoring semantics, not more complexity.

Operator notes

- Keep this worker simple because complexity here only weakens the safety net.
- The alert cooldown protects against flooding when the main system is down for a while.
- Because it is independent, this worker should be deployable and testable without the main app stack.
- Manual verification through ``/`` remains valuable during incidents or configuration drift.

Validation posture

- Review stale-threshold and cooldown changes as incident-policy changes, not just code edits.
- Test both the normal heartbeat path and the degraded failover path whenever alert logic changes.
- Preserve manual verification output because operators need a quick truth source during incidents.

Current risks or watchpoints

- Alerting depends on external provider credentials and network reachability.
- If heartbeat semantics change in the main system, the snitch threshold must move with them.
- Minimal code is a strength here; turning it into a general monitoring service would weaken it.

Glossary

Term	Meaning
Cooldown	A quiet period after an alert during which repeated alerts are suppressed.
Failover	A backup notification path used when the primary alert channel fails.
Dead-man's snitch	A monitor that alerts when a regularly expected signal stops arriving.