

Cloudflare Workflows Lifecycle and CPU-Cost Controls: Documented Boundaries for Idle Time, Instance Deletion, and Workflow Deletion

Scaledex Research Labs

Abstract—Cloudflare’s documented Workflows CPU model distinguishes active processing from elapsed wall-clock time. Waiting for an API response, pausing with `step.sleep`, or otherwise being idle does not incur CPU time. Separately, waits on network requests, storage calls, and other general I/O do not count toward CPU time or Workflows compute consumption. These conclusions apply only to the documented CPU and, where expressly stated, Workflows-compute dimensions; they do not establish treatment under other billing, quota, request, storage, or duration meters. For lifecycle controls, documented Workflow-instance deletion frees storage, with storage-limit updates possibly taking a few minutes. Separately, Wrangler workflows delete deletes a Workflow and its own instances. The documentation examined does not specify comprehensive retained-data deletion or physical-erasure semantics.

Index Terms—Cloudflare Workflows, CPU accounting, deletion semantics, workflow lifecycle, evidence synthesis

I. Introduction

This paper addresses two lifecycle-control questions in Cloudflare Workflows documentation retrieved on 2026-09-01: how idle and waiting periods are treated for CPU accounting, and what documented deletion actions establish about Workflow instances, Workflows, and storage. It confines its conclusions to CPU time, Workflows compute consumption, named deletion channels, and stated storage-limit update behavior [1]–[5].

II. Methods

The study applied an evidence-bound documentation-review protocol. Search results were used for discovery only; candidate claims were admitted only when linked to exact retained passages from first-party Cloudflare documents. The candidate manuscript then underwent gap assessment, challenge retrieval, an independent entailment audit, developmental review, and block-level fact checking. Citation numbers and references were generated deterministically from block-to-claim-to-evidence-to-source bindings.

This synthesis interprets Cloudflare Workflows documentation retrieved on 2026-09-01. It preserves the distinction between CPU time and wall-clock time, between CPU and possible non-CPU meters, and between individual-instance deletion and Workflow-definition deletion [1]–[5].

III. Results

A. CPU Accounting During Idle and Waiting Periods

Cloudflare describes Workflow CPU time as active processing rather than elapsed wall-clock time. Under that model,

time spent waiting for an API response, paused through `step.sleep`, or otherwise idle does not incur CPU time [1], [2].

The documented boundary also covers waits on network requests, storage calls, and other general I/O: those waits do not count toward CPU time or Workflows compute consumption. The evidence therefore supports treating these periods as non-CPU time rather than treating a Workflow’s full wall-clock lifetime as CPU processing [2].

B. Deletion Channels and Documented Consequences

TABLE I
Deletion Channels and Documented Consequences [1], [3]–[5]

Documented action or channel	Documented consequence, storage implication and timing, and effects not established
Delete Workflow instances through the Workers API, Wrangler CLI, REST API, or Dashboard	<i>Consequence</i> : Deleting instances frees storage. <i>Storage and timing</i> : Storage-limit updates may take a few minutes. <i>Not established</i> : Complete deletion of retained execution state, event or history records, logs, Durable Object-backed data, every storage class, or physical-erasure semantics.
Run Wrangler workflows delete	<i>Consequence</i> : Deletes a Workflow and its own instances. <i>Storage and timing</i> : No separate storage consequence or timing statement is established here beyond the instance-deletion documentation. <i>Not established</i> : Handling of active executions, recovery, retention behavior, and complete retained-data deletion.
Delete a Workflow from the Dashboard or UI, or via API	<i>Consequence</i> : These deletion channels are documented as available. <i>Storage and timing</i> : No instance, storage, or storage-limit-update consequence is stated for these channels. <i>Not established</i> : Effects on instances, state, history, or storage; the Wrangler consequence is not established for these channels.

C. Instance Deletion and Storage Controls

Cloudflare documents that deleting Workflow instances through the Workers API, Wrangler CLI, REST API, or Dashboard frees storage. This is the documented storage effect of instance deletion [1].

Cloudflare also states that storage-limit updates may take a few minutes after documented instance-deletion actions. The stated delay applies to storage-limit updates; it is not a statement about physical erasure, billing or invoice updates, instance-state visibility, or synchronized updates across every storage category [1].

D. Workflow Deletion and Its Instances

The Wrangler reference states that `workflows delete` deletes a Workflow and its own instances. This is distinct from deleting individual Workflow instances through the named instance-deletion channels [1], [3], [4].

Cloudflare documentation also establishes that Workflows can be deleted from the Dashboard or UI and via API. For those Dashboard/UI and API channels, the documentation examined does not describe effects on instances, state, history, or storage. The consequence of deleting a Workflow and its own instances is documented for Wrangler; the documentation examined does not establish that consequence for the Dashboard/UI and API channels [3]–[5].

IV. Discussion

A. Engineering-Control Implications

- Model documented API-response waits, `step.sleep` pauses, and idle periods as non-CPU periods for Workflows CPU-time planning.
- Model documented network, storage, and general-I/O waits as non-CPU periods for Workflows CPU-time and Workflows compute-consumption planning.
- Use documented Workflow-instance deletion channels when the objective is to free storage, and allow for a possible few-minute delay before relying on the storage-limit update.
- Use `Wrangler workflows delete` when the documented objective is to delete a Workflow and its own instances.
- Obtain separate authoritative evidence before modeling non-CPU meters or treating either deletion action as complete removal of retained artifacts [1]–[5].

B. Limitations

The documentation establishes non-CPU treatment for API-response waits, `step.sleep` pauses, and other idle periods only for CPU time. It expressly establishes non-CPU and

non-Workflows-compute treatment for network, storage, and general-I/O waits. It does not establish treatment under other billing meters, quotas, request meters, storage meters, or duration-related measures [1], [2].

The documented Wrangler wording is limited to deletion of the Workflow and its own instances. It does not describe active-execution handling, recovery, retention behavior, or complete removal of retained artifacts [1], [3]–[5].

V. Conclusion

The documentation supports non-CPU CPU-time treatment for the specified API-response, sleep, and idle periods; non-CPU and non-Workflows-compute treatment for network, storage, and general-I/O waits; storage reclamation from instance deletion with a possible few-minute storage-limit update lag; and the Wrangler command’s deletion of a Workflow and its own instances. It does not support conclusions about other non-CPU meters, complete retained-data deletion, active-execution handling, recovery, physical erasure, or the retention behavior of deletion actions [1]–[5].

AI Assistance Disclosure

This report is an AI-assisted synthesis of retained public sources; it does not report original interviews or experiments.

References

- [1] Cloudflare, “Pricing - Cloudflare Workflows docs,” *Cloudflare Developers*. Accessed: Sep. 1, 2026. [Online]. Available: <https://developers.cloudflare.com/workflows/reference/pricing/>
- [2] Cloudflare, “Limits - Cloudflare Workflows docs,” *Cloudflare Developers*. Accessed: Sep. 1, 2026. [Online]. Available: <https://developers.cloudflare.com/workflows/reference/limits/>
- [3] Cloudflare, “Wrangler commands - Cloudflare Workflows docs,” *Cloudflare Developers*. Accessed: Sep. 1, 2026. [Online]. Available: <https://developers.cloudflare.com/workflows/reference/wrangler-commands/>
- [4] Cloudflare, “Workflows | Cloudflare API,” *Cloudflare Developers*. Accessed: Sep. 1, 2026. [Online]. Available: <https://developers.cloudflare.com/api/go/resources/workflows/>
- [5] Cloudflare, “Release notes - Cloudflare Workflows docs,” *Cloudflare Developers*. Accessed: Sep. 1, 2026. [Online]. Available: <https://developers.cloudflare.com/workflows/reference/changelog/>