

The Cheapest Cloud Nobody Benchmarks

Cloudflare priced against AWS, Azure and Google Cloud; why AI-assisted coding makes low-code seat rent optional; and the workloads where none of this applies

David Soden

Independent analyst and practitioner, enterprise automation and applied AI
davidson.com/market-assessment

About this report

Every substantive claim in this report carries an evidence classification and, where applicable, a numbered source. Facts are separated from vendor claims, third-party estimates, the author's assessments, and untested hypotheses. Forward-looking sections use scenarios with observable tripwires rather than point forecasts. Stated assumptions are listed before the analysis begins.

PUBLISHED FOR PUBLIC READING | SHARE FREELY, UNMODIFIED, WITH ATTRIBUTION

© 2026 David Soden, davidsoden.com/market-assessment. All rights reserved. You may share this file complete and unmodified, and quote short passages with attribution. Republishing under another name, excerpting into a commercial deliverable, resale, and use as machine learning training data require written consent. Full terms appear under Rights and Permitted Use on the final page.

Market Assessment reports are published and commissioned at davidsoden.com/market-assessment. This document is analysis, not investment, legal, or procurement advice.

Scope and evidentiary standard

There are four companies running general-purpose compute at global scale and selling it by the request. Three of them get benchmarked against each other constantly. The fourth arrived from the content-delivery side rather than the compute side, prices on a structurally different model, and is largely absent from the comparison. This report prices it.

Two arguments run through the document. The first is a cost argument: Cloudflare does not bill outbound data transfer, and outbound data transfer is where the bill concentrates on anything that serves traffic to end users. The second is a tooling argument: configuring this platform means moving across a command-line tool, a set of AI agent integrations and an HTTP API, which is a barrier to a person learning it cold and an advantage to an AI assistant that has already read all three sets of documentation. Those two arguments together are what makes the third claim possible, which is that a team writing code with an assistant and deploying it here can decline to pay two separate rents: hyperscaler egress, and low-code platform seats.

Coverage runs from a five-page brochure site to enterprise workloads. Those are genuinely different analyses and the report keeps them separate rather than averaging them. The small-business case is the simplest and is treated first because it is the one where the answer is unambiguous. The enterprise case is where the interesting qualifications live.

Every platform claim is verifiable from public documentation or by running one command against the reader's own account. Nothing rests on a private engagement, a named client, or a case that cannot be published. Where a claim comes from practitioner experience rather than a citable source, it carries an ASSESSMENT label and says so in the sentence.

All pricing was captured on 22 August 2026 from each vendor's own current pricing page. A vendor pricing page is a primary source for what that vendor charges and those figures are labeled FACT. Vendor language about what the pricing means is labeled VENDOR CLAIM.

Comparison sites, hosting-review blogs and best-of listicles carry affiliate revenue on nearly every link; they are used here only to establish that a complaint recurs, never to quantify anything.

One methodological limitation deserves stating up front rather than in the limitations section, because it affects every comparison in the document. All hyperscaler figures here are list prices. Enterprise customers with committed-spend agreements routinely pay well below list, and no public source discloses those discounts. A list-to-list comparison therefore flatters Cloudflare, and the reader with an existing enterprise agreement should treat the gaps in this report as an upper bound rather than a quote.

The author has no commercial relationship with Cloudflare, Amazon, Microsoft, Google, Retool, or any other vendor named here. There is no reseller arrangement, no affiliate link, no referral fee and no sponsorship. That matters, because this is an argument a hosting reseller cannot credibly make and one a Cloudflare partner has every reason to overstate.

Label	Definition	How to treat it
FACT	Verifiable in the cited primary source: a filing, press release, official documentation, or standards body announcement.	Reliable as stated. Check the source if the exact wording matters.
VENDOR CLAIM	Reported by a vendor about its own business or product. Self-measured and not audited by any third party.	The vendor said it. Directionally useful, not a measurement. Definitions of the metric are usually undisclosed.
THIRD-PARTY ESTIMATE	A research firm forecast or survey result. Methodology is proprietary and generally not reproducible.	Use for direction and order of magnitude only. Do not build a model on it.
ASSESSMENT	The author's reasoned judgment from the cited evidence. Falsifiable, and the reasoning is shown so it can be argued with.	This is analysis, not reporting. Disagree with it where your evidence differs.
HYPOTHESIS	A proposition offered for testing. Not currently supported by sufficient evidence to assert.	Treat as a research question, not a conclusion. Each one names what would confirm or refute it.
SCENARIO	A structured future state with a stated subjective probability. The probability is the author's judgment, not a calculated figure.	Use the leading indicators attached to each, not the probability. The indicators are observable; the probability is not.

Assumptions this analysis rests on

Five premises are assumed rather than shown. Each names what fails if it is wrong.

First, that outbound traffic to end users is a material share of the workload. The entire cost argument rests on egress asymmetry. An internal data pipeline that never leaves the provider's

network has little egress, and for that workload Cloudflare's advantage largely evaporates and the comparison should be run on compute and storage alone.

Second, that list price is the relevant comparison for the reader. If the organization already holds a committed-spend agreement with a hyperscaler, the effective rate may be 40 to 70 percent below the figures used here, and the decision becomes a marginal-workload question rather than a platform question.

Third, that Cloudflare's no-egress-billing position holds. It is a published standing rate with no contractual protection behind it on self-serve plans. If Cloudflare introduces egress billing, the central argument of this report collapses and nothing else in it survives that change.

Fourth, that an AI coding assistant is available to the team and is supervised by someone who can read its output critically. The tooling argument in the second half assumes both. Unsupervised, the same assistant produces the silent failures catalogued later, and the configuration advantage inverts into a liability.

Fifth, that the workload fits inside a 128 MB isolate and bounded CPU time, or can be decomposed to fit. This is the hard technical ceiling on the platform. Where it cannot be met, Cloudflare Containers exist but change the pricing model, and the comparison has to be rerun.

The call

ASSESSMENT Cloudflare is the cheapest of the four clouds for anything that serves traffic to end users, and almost nobody benchmarks it because it arrived from the CDN side rather than the compute side. The saving is structural rather than promotional: no egress billing, against nine to twelve cents a gigabyte everywhere else. Pair that with an assistant writing the code and the per-seat rent a low-code platform charges becomes optional too. It is not cheaper for sustained compute, large-memory work, or anything needing a managed warehouse, and enterprise discount agreements narrow every gap here. What would overturn it: Cloudflare introducing egress billing, or the hyperscalers dropping theirs for ordinary outbound traffic rather than only for customers leaving.

My judgment on the evidence assembled here, nothing more. There is data I can't see and data I may have missed. New evidence can move me, including to the opposite position, and I reserve the right to move.

Executive summary

The comparison that decides most cloud bills is not compute price per millisecond. It is what the provider charges to move a gigabyte to a customer's browser, and on that line three of the four major providers charge between 8.7 and 12 cents while the fourth charges nothing at all. That asymmetry compounds every month and it is almost entirely absent from public cloud comparisons.

FINDING 1 Three providers charge for egress. One does not, and says so in writing.

FACT AWS charges 0.09 dollars per gigabyte for S3 data transfer out beyond the first 100 GB. Azure charges 0.087 dollars per gigabyte beyond 100 GB. Google Cloud Premium Tier charges 0.12 dollars per gibibyte to North America beyond the first gibibyte. Cloudflare's R2 documentation states that egressing directly from R2, including via the Workers API, S3 API and r2.dev domains, does not incur data transfer charges and is free. [\[8\]](#) [\[44\]](#) [\[45\]](#) [\[46\]](#)

FINDING 2 At ten terabytes a month the gap is roughly a thousand dollars, every month, for the same bytes.

FACT Ten terabytes of outbound traffic costs 913 dollars on AWS, 882 on Azure and 1,137 on Google Cloud at published rates. On Cloudflare it costs nothing. This is arithmetic on four vendor pricing pages, not a modeled estimate. [\[8\]](#) [\[44\]](#) [\[45\]](#) [\[46\]](#)

FINDING 3 Cloudflare is not cheaper on compute, and the honest comparison says so.

FACT AWS Lambda's free tier is one million requests and 400,000 GB-seconds per month, perpetual rather than introductory. Ten million monthly requests on Lambda with a small function costs roughly two dollars. The same volume on Cloudflare costs five, because five dollars is the floor on the Workers Paid plan. Cloudflare wins on data transfer, not on execution. [\[1\]](#) [\[47\]](#)

FINDING 4 The stateful and scheduled workloads I would previously have told you to leave alone are now serviceable.

FACT Durable Objects provide per-object state with an embedded SQLite database, priced at 0.15 dollars per million requests. Jurisdiction-restricted namespaces exist for the European Union, the United States and FedRAMP-Moderate data centers. Cron Triggers are supported on Workers. Containers exist for work that exceeds an isolate. The constraint that remains real is memory: 128 MB per isolate. [\[48\]](#) [\[49\]](#) [\[50\]](#) [\[51\]](#)

FINDING 5 The residency story is partial, and the distinction matters.

FACT Cloudflare documents jurisdictions as enforcing where a Durable Object runs and stores data, while location hints are explicitly described as best effort and not a guarantee. So object state can be pinned; general Workers execution is distributed by design. A workload with a hard residency obligation can be built here, but only deliberately. [\[50\]](#)

FINDING 6 Low-code platforms charge rent per person, and AI-generated code is what makes that rent optional.

FACT Retool's Business tier is 50 dollars per builder per month on annual billing plus 15 per internal user, with external users billed separately above 50. Power Apps Premium is 20 dollars per user per month, and additional Dataverse database capacity is 40 dollars per gigabyte per month. Cloudflare charges for usage, not for seats. [\[52\]](#) [\[53\]](#)

FINDING 7 Forty dollars per gigabyte-month is the number that should end the conversation.

FACT Microsoft's published Dataverse capacity add-on rate is 40 dollars per gigabyte per month. Cloudflare D1 is 0.75, Durable Objects SQLite storage is 0.20, and R2 is 0.015. The

same hundred gigabytes of application data costs 48,000 dollars a year in the first case and 18 in the last. [\[7\]](#) [\[8\]](#) [\[48\]](#) [\[53\]](#)

FINDING 8 Three configuration surfaces is a barrier to a person and an advantage to an assistant.

ASSESSMENT Only the command-line tool can deploy, because functions must be compiled first and no dashboard action performs that compilation. The AI agent integrations read account state and manage containers but not contents. The HTTP API fills the rest. A developer learning this cold loses a day to it. An assistant that has read all three sets of documentation does not, which is the actual reason the economics of this configuration work changed. [\[14\]](#) [\[15\]](#) [\[21\]](#)

FINDING 9 Every important failure mode on this platform is silent, which is precisely what unsupervised AI configuration produces.

FACT Dashboard drag-and-drop ships static files and no compiled functions, reporting success. A deploy without the right branch argument lands as a preview and reports success. Referrer-restricted API keys fail server-side with an error pointing at the front end. Function responses at extensionless paths are not edge-cached even when the headers ask for it. None of these produces a useful error anywhere. [\[12\]](#) [\[13\]](#) [\[14\]](#) [\[15\]](#) [\[11\]](#)

FINDING 10 The egress waivers announced in 2024 do not change any number in this report.

FACT Google, AWS and Microsoft each waived data transfer charges for customers leaving their platform during the first quarter of 2024, and from January 2027 the EU Data Act prohibits switching fees outright for EU customers. All of that concerns exit traffic. Ordinary outbound traffic to your own end users is billed exactly as before. [\[55\]](#)

FINDING 11 Cloudflare holds the certifications an enterprise procurement process asks for, and has a documented history of terminating customers over content.

FACT SOC 2 Type II, ISO 27001, ISO 27701 and PCI DSS Level 1 Service Provider assessment are all published. Cloudflare has also terminated the Daily Stormer in 2017, 8chan in 2019 and Kiwi Farms in 2022. The threshold applied has been extremely high, but the decision is Cloudflare's and is made under public pressure on a timescale of days. [\[54\]](#) [\[37\]](#) [\[38\]](#) [\[43\]](#)

FINDING 12 The product you would have picked two years ago is the one Cloudflare now tells you not to start on.

FACT The Pages documentation states that Workers supports most Pages use cases, offers a broader feature set, is Cloudflare's primary platform for building applications, and that new projects should start with Workers. The compatibility matrix confirms the direction: Cron Triggers, Queue Consumers, Rate Limiting, Tail Workers and Logpush are all Workers-only. [\[3\]](#) [\[20\]](#)

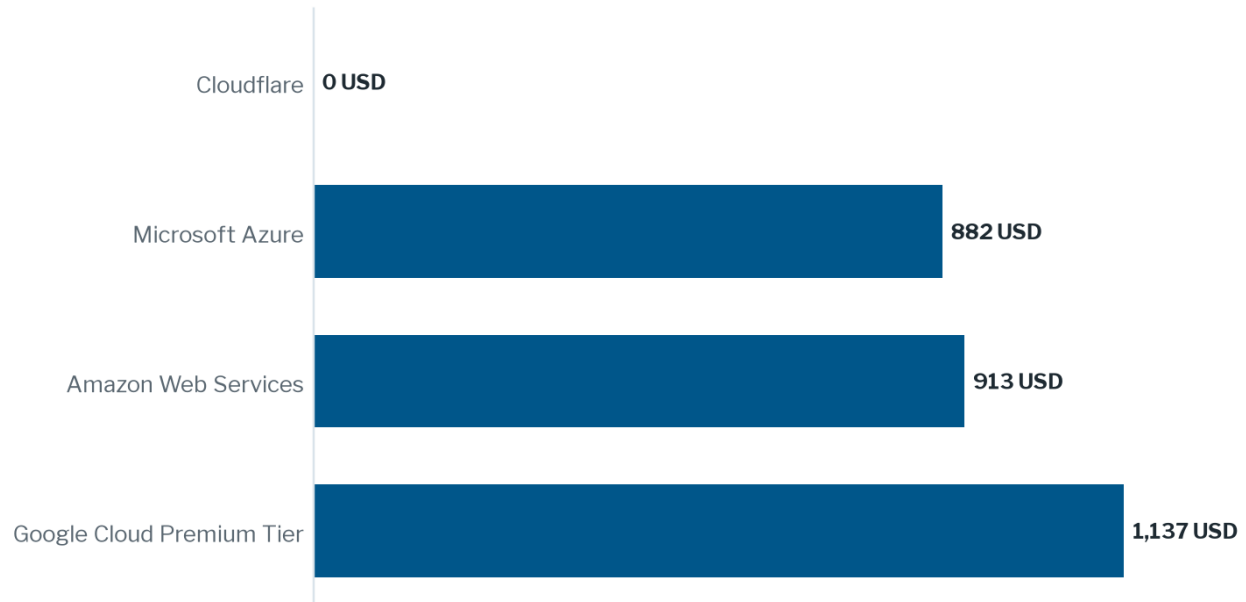
The structural difference: nobody bills you for bandwidth

Start here, because every other number in this report is downstream of it. The four providers do not merely price differently. One of them has removed an entire billing dimension that the other three treat as a primary revenue line.

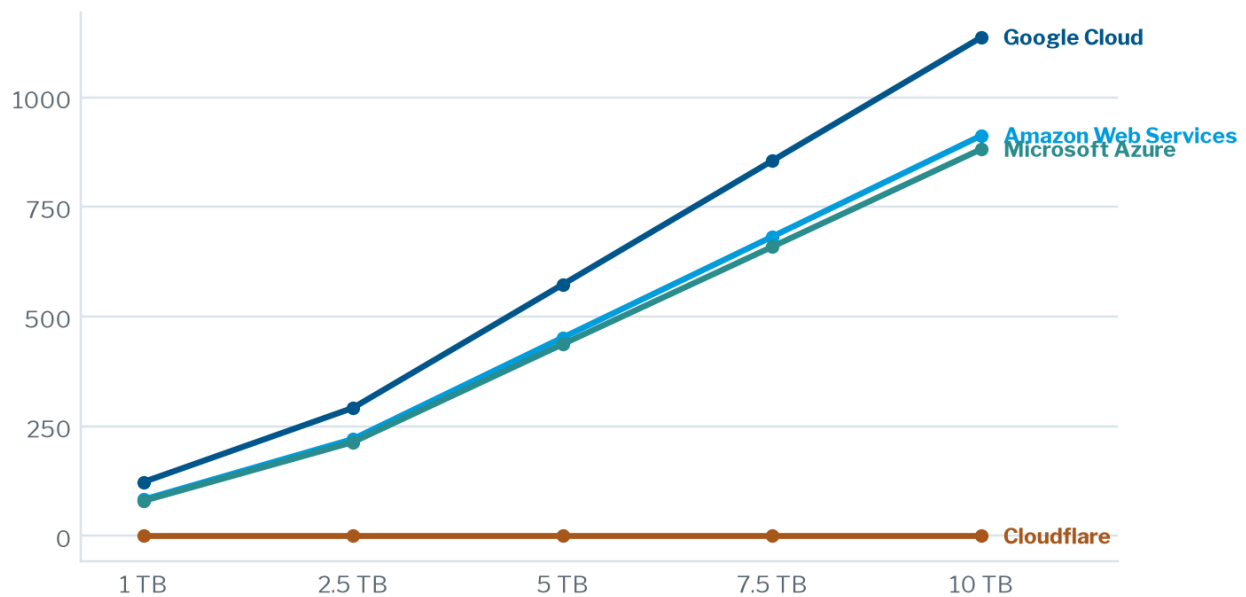
FACT Cloudflare's R2 documentation states that egressing directly from R2, including via the Workers API, S3 API and r2.dev domains, does not incur data transfer charges and is free. The Workers pricing page carries the corresponding statement for the compute platform: requests to static assets are free and unlimited, and there is no separate charge for bandwidth. [\[1\]](#) [\[8\]](#)

FACT AWS S3 provides the first 100 GB per month of data transfer out free, aggregated across services and regions, then charges 0.09 dollars per gigabyte from US East regions. Azure provides the first 100 GB per month free, then 0.087 dollars per gigabyte for the next 10 TB from North America and Europe, stepping down to 0.083, 0.07 and 0.05 at higher volumes. Google Cloud Premium Tier provides the first gibibyte free, then charges 0.12 dollars per gibibyte to North America up to one tebibyte, 0.11 to ten tebibytes, and 0.08 above that. [\[44\]](#) [\[45\]](#) [\[46\]](#)

Provider	Free per month	Rate after	Cost of 10 TB served
Cloudflare	Not applicable	No data transfer charge	0 USD
Microsoft Azure	100 GB	0.087 USD per GB to 10 TB	882 USD
Amazon Web Services	100 GB	0.09 USD per GB from US East	913 USD
Google Cloud	1 GiB	0.12 USD per GiB to 1 TiB, then 0.11	1,137 USD

FIGURE 1 Monthly cost of serving 10 TB, at published rates

FACT Arithmetic on four vendor pricing pages as at 22 August 2026, applying each provider's published free allowance and first-tier rate to 10,240 GB of outbound traffic to North America. List prices only; committed-spend agreements are not public and would reduce the three paid figures. [\[8\]](#) [\[44\]](#) [\[45\]](#) [\[46\]](#)

FIGURE 2 Monthly egress bill as traffic grows

FACT Computed from each provider's published tier structure, rounded to whole dollars. The Cloudflare line is flat at zero because there is no volume dimension to bill, which is the point: the other three curves are the shape of a cost that scales with your own success. [\[8\]](#) [\[44\]](#) [\[45\]](#) [\[46\]](#)

ASSESSMENT The commercial consequence is not that Cloudflare is a bit cheaper. It is that one provider has removed the line item that makes traffic growth financially frightening, and the other three have not. A marketing campaign that works, a product that gets written about, a file that gets shared more than expected: on three of these platforms each of those events arrives as an invoice.

FACT The 2024 egress waivers are frequently cited as having closed this gap. They did not. Google announced free network data transfer for departing customers in January 2024, AWS followed in March, and Microsoft completed the set the same month. The EU Data Act, in force since January 2024 and applying since September 2025, prohibits switching fees outright for EU customers from January 2027. Every one of those measures concerns data leaving the platform on exit. None of them touches the cost of serving your own users. ^[55]

ASSESSMENT That distinction is worth holding onto, because it reveals what the fees were actually for. Egress-on-exit was lock-in and regulators removed it. Egress-on-operation is revenue and nobody has removed it. Cloudflare's position is unusual precisely because it gave up the second one, not the first.

What Cloudflare actually is now

The platform is wider than its reputation. A reader whose mental model is a CDN with a DDoS product attached is working from roughly 2019. What follows is the current set of primitives, with the constraint that matters attached to each.

FACT Static assets. Build output is uploaded and served from the edge. Requests to static assets are free and unlimited on every plan including the free one, and a request qualifies as static when it does not invoke a Function. ^{[1] [2]}

FACT Workers. Server-side JavaScript or WebAssembly executing at the edge. The Workers Paid plan is five dollars a month including ten million requests and 30 million CPU milliseconds, with overage at 0.30 dollars per additional million requests and 0.02 dollars per additional million CPU milliseconds. The free plan allows 100,000 requests a day. ^[1]

FACT Workers KV. A low-latency key-value store read fresh on every request and editable without redeploying. Free plan: 100,000 reads, 1,000 writes, 1,000 deletes and 1,000 list operations per day with one gigabyte stored. Values are encrypted at rest with AES-256 in Galois/Counter Mode. ^{[4] [5] [6]}

FACT D1. A SQL database. Free plan allows five million rows read and 100,000 rows written per day with five gigabytes stored. The paid plan includes 25 billion rows read and 50 million written per month, then 0.001 dollars per million rows read and 0.75 dollars per gigabyte-month of storage. ^[7]

FACT R2. Object storage at 0.015 dollars per gigabyte-month standard, with Class A operations at 4.50 dollars per million and Class B at 0.36, and a free allowance of ten gigabyte-months plus one million Class A and ten million Class B operations. Egress is free. ^[8]

FACT Durable Objects. Single-instance stateful compute with an embedded SQLite database, addressed by name so that all requests for a given identity reach the same object. Priced at 0.15 dollars per million requests and 12.50 dollars per million GB-seconds beyond one million requests and 400,000 GB-seconds included monthly. SQLite storage is 0.20 dollars per gigabyte-month beyond five gigabytes, with 25 billion rows read included. ^[48]

FACT Containers. For work that will not fit in an isolate. The Workers Paid plan includes 25 GiB-hours of memory, 375 vCPU-minutes and 200 GB-hours of disk per month, with additional CPU at 0.000020 dollars per vCPU-second. Cloudflare documents that memory and disk are billed on provisioned resources for the selected instance type while CPU is billed on active usage only, and that instances sleep automatically after a timeout. ^[51]

FACT Queues. Message queues between Workers, with the first million operations a month included on the paid plan and 0.40 dollars per million after, where a write, a read and a delete each count as one operation. Queue consumers are a Workers feature and are unsupported on Pages. ^{[3] [57]}

ASSESSMENT Read that list against the objection that Cloudflare is not a real cloud. It has compute, a key-value store, a relational database, object storage, a stateful primitive with its own embedded SQL engine, and containers for the overflow. What it does not have is the fifty-service catalogue, and for a large class of applications the fifty-service catalogue was never the reason to be there.

How the pieces compose: cache, then refresh in the background

ASSESSMENT The pattern worth learning properly is the one that lets a site serve third-party data at edge speed without paying a round trip on every visit. A function reads a cached copy from KV and returns it immediately. If the copy is older than a threshold, the function also schedules a background refresh that fetches upstream and writes the new value back, using `waitUntil` on the request context so the response is not held waiting. ^[18]

ASSESSMENT The commercial consequence is that the upstream API's rate limit stops being the binding constraint. A feed refreshed every fifteen minutes makes 96 upstream calls a day whether the site serves a hundred visitors or a hundred thousand. That decouples traffic from vendor quota, which is frequently the thing that was forcing a more expensive plan somewhere else.

FACT One caveat that catches people. The Workers Cache API is not a substitute for KV in this pattern: Cloudflare documents that cache contents do not replicate across data centers, so a value cached in one location does not exist in another, and that stale-while-revalidate and stale-if-error directives are unsupported. KV is what gives a consistent view. ^[19]

Priced against the big three

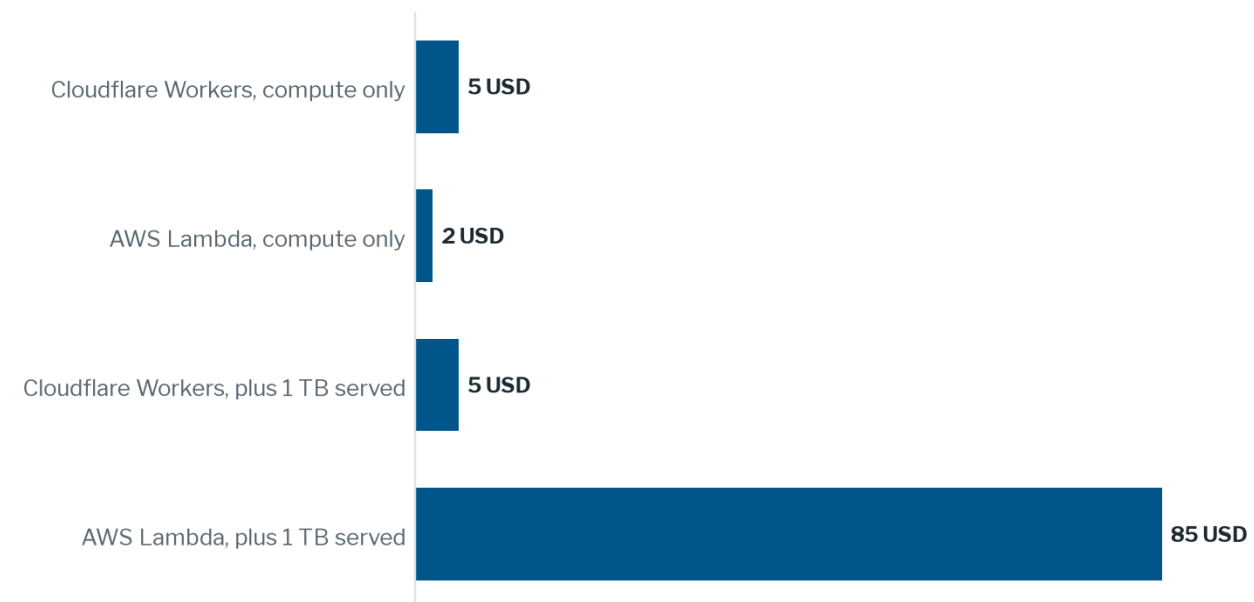
Egress is the headline and it is not the whole picture. Here is the rest, including the places where Cloudflare loses.

Compute, where Cloudflare loses

FACT AWS Lambda charges 0.20 dollars per million requests and 0.0000166667 dollars per GB-second on x86. The free tier is one million requests and 400,000 GB-seconds per month and AWS documents it as a perpetual monthly benefit rather than a twelve-month trial. Lambda Function URLs carry no separate charge. ^[47]

ASSESSMENT Run ten million monthly requests through a small function on each platform. On Lambda, nine million billable requests at 0.20 per million is 1.80 dollars, and a 128 MB function running 50 milliseconds consumes about 62,500 GB-seconds, which sits inside the 400,000 free. Total: under two dollars. On Cloudflare the same volume is five dollars, because five dollars is the Workers Paid floor. Lambda is cheaper. ^{[1] [47]}

FIGURE 3 Ten million monthly requests, compute only and then with traffic attached



ASSESSMENT The author's arithmetic on both vendors' published rates, rounded to whole dollars. Compute assumes a 128 MB function running 50 ms. The lower two bars add one terabyte of outbound traffic at each provider's published rate, which is where the comparison inverts. Excludes any managed service either function might call. ^{[1] [47]}

ASSESSMENT That chart is the whole argument. Compare execution and Cloudflare loses. Attach the traffic that execution exists to serve and the position reverses by a factor of seventeen. Any comparison that stops at the compute line reaches the wrong answer, and most published comparisons stop at the compute line.

Storage

FACT S3 Standard is 0.023 dollars per gigabyte-month in US East. R2 Standard is 0.015. On storage alone R2 is roughly a third cheaper, which is real but not decisive. The decisive difference is what happens when the objects are read by someone outside the provider's network. ^{[8] [44]}

ASSESSMENT One terabyte stored and ten terabytes served in a month costs about 936 dollars on S3, of which 913 is data transfer and 24 is storage. The same workload on R2 costs about 15 dollars, all of it storage. The storage line is a rounding error in both cases; the egress line is the entire bill in one of them. [\[8\]](#) [\[44\]](#)

FIGURE 4 One terabyte stored, ten terabytes served, per month



ASSESSMENT The author's arithmetic on both vendors' published rates as at 22 August 2026, rounded to whole dollars, excluding per-operation charges which are small at this volume. The two storage bars are nearly identical. The comparison is decided entirely by the two transfer bars beneath them. [\[8\]](#) [\[44\]](#)

State, and the verdict I got wrong

ASSESSMENT An earlier version of this analysis told readers that anything with meaningful server-side state, anything on a schedule, and anything with a residency obligation should stay where it was. That conflated the limits of one Cloudflare product, Pages, with the limits of the platform. It was wrong on all three counts and the correction is worth making explicitly rather than quietly.

FACT On state: Durable Objects are a stateful primitive by design. Each object is addressed by name, all requests for that name reach the same instance, and each instance has its own SQLite database. Twenty-five billion rows read a month are included on the paid plan. [\[48\]](#)

FACT On schedules: Cron Triggers are supported on Workers. They are unsupported on Pages, which is a product limitation stated in Cloudflare's own compatibility matrix, not a platform one. Scheduled invocations may run up to 15 minutes. [\[3\]](#) [\[49\]](#)

FACT On residency: Cloudflare documents jurisdiction-restricted Durable Object namespaces for eu, us and fedramp, describing them as creating objects that only run and store data within a

region to comply with local regulations. Location hints, by contrast, are documented as best effort and not a guarantee. ^[50]

ASSESSMENT The honest revised verdict: state and scheduling are solved problems on Workers. Residency is partially solved and requires deliberate design, because pinning an object's storage is not the same as pinning every execution path that touches it. A regulated workload can be built here but not accidentally, and anyone with a hard residency obligation should read the jurisdiction documentation themselves rather than take this paragraph for it. ^[50]

The compliance floor

FACT Cloudflare publishes SOC 2 Type II covering security, confidentiality and availability; ISO 27001 and ISO 27701; and annual assessment by a qualified security assessor as a PCI DSS Level 1 Merchant and Service Provider. Reports are downloadable from the account dashboard rather than the public site, which is standard practice. ^{[54] [56]}

ASSESSMENT This clears the bar most enterprise procurement processes actually set. What it does not do is match the breadth of authorizations the hyperscalers carry across government and healthcare verticals, and a FedRAMP-Moderate data center jurisdiction for Durable Objects is not the same thing as a FedRAMP-authorized platform. A reader whose deal depends on a specific authorization should verify it directly and not infer it from this list.

Where Cloudflare is not a competitor

This section is what makes the rest of the report believable. There are workloads where the answer is not close, and pretending otherwise would be the same overclaiming the platform's advocates already do.

FACT Memory is the hard ceiling. Cloudflare documents that each isolate can consume up to 128 MB, including the JavaScript heap and WebAssembly allocations. No plan raises it. A workload that needs to hold a large dataset in memory does not fit and cannot be made to fit by paying more. ^[49]

FACT CPU time is bounded. The free plan allows 10 milliseconds of CPU per request. The paid plan defaults to 30 seconds and permits up to five minutes. Cron Triggers, Queue Consumers and Durable Object alarms are capped at 15 minutes. Anything genuinely long-running belongs somewhere else, or in a Container. ^{[49] [51]}

ASSESSMENT There is no managed data warehouse. Nothing here corresponds to BigQuery, Redshift or Synapse. There is no managed model-training service and no general-purpose GPU fleet; Workers AI is inference against hosted models, which is a different product from renting accelerators by the hour. An analytics or machine-learning platform is not a candidate for this migration and no amount of egress saving changes that.

ASSESSMENT The service catalogue is narrow by design. If an application depends on a specific managed queue semantic, a particular identity provider integration, a hardware security module, or any of several hundred niche managed services, the equivalent either does not exist or has to

be assembled. That assembly is engineering time, and engineering time is the cost most migration business cases understate.

FACT Lambda beats Workers on compute-only cost at moderate volume, as established above, because Lambda's one million requests and 400,000 GB-seconds are free every month in perpetuity while Cloudflare's paid tier starts at five dollars. ^{[1] [47]}

FACT There is a contractual limit on media that a cost comparison should not skip. Cloudflare's service-specific terms state that Cloudflare offers specific paid services, naming the Developer Platform, Images and Stream, that a customer must use in order to serve video and other large files via the CDN, and that it reserves the right to limit accounts using the CDN without them to serve video or a disproportionate share of pictures, audio or other large files. Workers and R2 sit inside the Developer Platform, which is one of the named products, so a deliberately architected media workload is within the terms. A large file library served straight through the CDN is not. ^[24]

ASSESSMENT And the commercial reality that overrides several of the above: an organization with an existing enterprise agreement is comparing a discounted incumbent against a new vendor requiring its own security review, its own procurement cycle, and its own on-call runbook. The list-price gap has to be large enough to survive all of that. For egress-heavy workloads it usually is. For anything else, frequently not.

Workload	Verdict	Binding constraint
Marketing site, brochure site, docs	Move to Workers	None. This is the platform's strongest case
Media or file distribution at volume	Move to Workers and R2	Read the CDN clause in the service-specific terms first
API serving external consumers	Move	Egress asymmetry is largest here
Scheduled jobs and background refresh	Move to Workers, not Pages	Cron is Workers-only; 15 minute ceiling
Stateful sessions, collaborative state	Move, using Durable Objects	Design work, not a cost question
Regulated workload with residency rules	Possible, deliberately	Jurisdictions pin object storage, not all execution
Long-running batch processing	Containers or stay	128 MB isolate, bounded CPU
Data warehouse or BI workload	Stay	No equivalent product exists
Model training or GPU compute	Stay	No general-purpose accelerator fleet
Large in-memory application	Stay	128 MB per isolate, no paid upgrade
Internal pipeline with no egress	Run the numbers again	The egress advantage does not apply

FIGURE 5 Eleven workload classes, sorted by verdict

ASSESSMENT The author's classification of the workload table above. The counts are a summary of a judgment, not a measurement, and the classes are not equally weighted by how common they are. Presented to make the shape of the answer visible: most things move, and the things that do not, do not move for technical reasons rather than cost reasons. [\[49\]](#) [\[50\]](#) [\[51\]](#)

The second rent: low-code platform seats

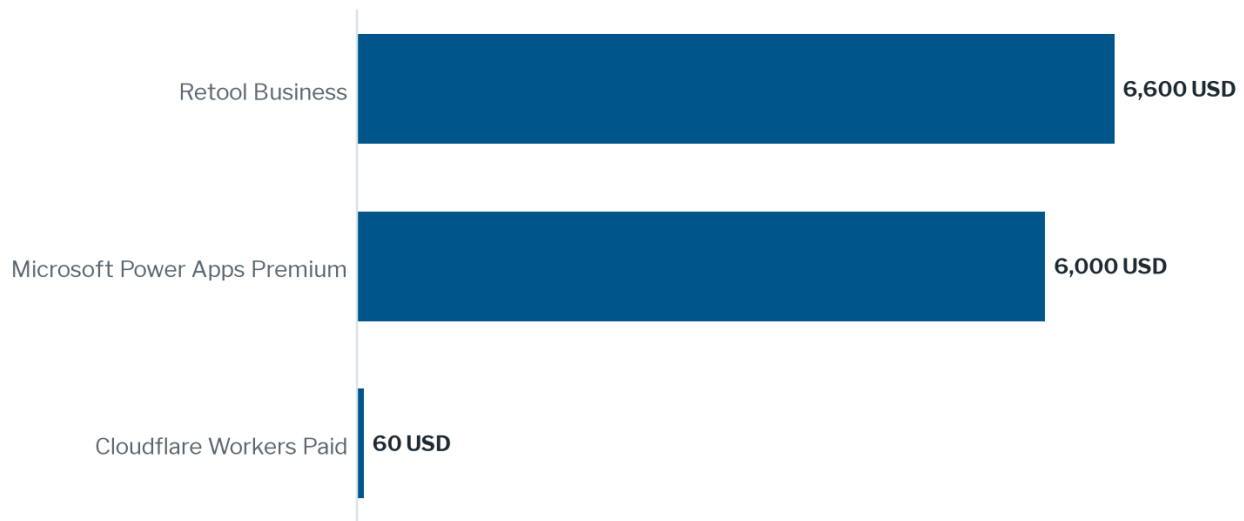
The cloud bill is one recurring charge an AI-assisted team can decline. The other is the low-code platform, and its pricing model is structurally worse in a way that has nothing to do with compute.

FACT Retool's published pricing is per seat with three user classes. On the Business tier, a builder is 50 dollars per month on annual billing or 65 monthly, and an internal user is 15 or 18. External users are free to 50, then 8 dollars per month annual for 51 to 250, 6 for 251 to 500, and 4 above 500. Enterprise pricing is quote-only. [\[52\]](#)

FACT Power Apps Premium is 20 dollars per user per month paid yearly, or 12 for organizations buying a minimum of 2,000 seats. The Premium licence includes 250 MB of database capacity and 2 GB of file storage per user. Additional Dataverse database capacity is 40 dollars per gigabyte per month paid yearly. [\[53\]](#)

ASSESSMENT Hold those two facts next to each other and the model becomes visible. You are billed for how many people touch the application and for how much data it holds, at rates set without reference to what the underlying compute or storage costs. Neither number has anything to do with the work the application does.

FIGURE 6 Annual platform cost for one internal application team of five builders and twenty users



ASSESSMENT The author's arithmetic on published list prices as at 22 August 2026. Retool: five builders at 50 USD and twenty internal users at 15 USD, annual billing. Power Apps: twenty-five users at 20 USD, paid yearly, before any Dataverse capacity add-on. Cloudflare: the five dollar Workers Paid floor, because seats are not a billing dimension. Excludes the engineering time to build the application, which is not equal across the three and is discussed in the text. [\[1\]](#) [\[52\]](#) [\[53\]](#)

ASSESSMENT That chart is not a fair fight and it should not be read as one. Retool and Power Apps ship an application builder, a permissions model, an audit trail, connectors and a hosted runtime. Cloudflare ships primitives. The 6,540 dollar gap buys you something, and until recently what it bought was the only realistic path to an internal tool for a team without spare engineering capacity.

ASSESSMENT The reason to revisit it now is that the thing being bought has changed. What a low-code platform sold was a visual canvas that let a non-specialist assemble an application without writing code. An AI coding assistant sells the same outcome by a different route, and the route it takes produces ordinary source code in a repository the customer owns.

FACT The platforms themselves are behaving as though this is true. Retool ships AI app generation that produces a working application from a plain-language description against connected data. Bubble ships AI generation as the entry path with the visual editor positioned as the place you go afterwards for control. Both are, in effect, putting a prompt in front of the canvas. [\[52\]](#) [\[58\]](#)

HYPOTHESIS The proposition, offered for testing rather than asserted: as AI generation becomes the primary authoring interface on low-code platforms, the visual canvas stops being the product and the remaining defensible value is the hosted runtime, the permissions model and the audit trail. If that holds, per-seat pricing for those three things looks increasingly hard to defend against a repository plus a five dollar compute plan. It would be confirmed by low-code vendors repricing away from seats toward usage, and refuted if seat-based revenue holds while AI generation is absorbed as a feature. Cloudflare's own pricing is already usage-based, which is not evidence for the hypothesis but does mean the alternative already exists.

ASSESSMENT The honest caveat, and it is a large one. Generated code is code, and code has to be reviewed, tested, secured and maintained by somebody. The low-code platform was absorbing some of that burden invisibly. A team that swaps a per-seat invoice for a repository has not eliminated a cost, it has converted a predictable operating expense into an engineering obligation, and the conversion is only favorable if the engineering capacity genuinely exists. Where it does not, the seat rent is buying something real.

ASSESSMENT Where the swap is clearly favorable: applications whose logic is simple and whose user count is large, because seat pricing punishes exactly that shape. Where it is clearly not: applications with complex permissions and a handful of users, which is what these platforms were built for and are still good at.

What AI-assisted coding actually changes

This is the part of the argument most likely to be overclaimed, so the limit comes before the benefit.

ASSESSMENT Configuring this platform means moving between three interfaces: the Wrangler command-line tool, the dashboard, and Cloudflare's HTTP API driven directly. Knowing which one a given step requires is the actual skill, and it is not written down in one place because no single document covers all three.

FACT Only the command-line tool can deploy. Functions are source files that must be compiled into a deployable worker before Cloudflare can run them, and Wrangler performs that compilation as part of the deploy. Cloudflare states directly that dashboard drag-and-drop deployments do not support compiling a functions folder. ^{[14] [15]}

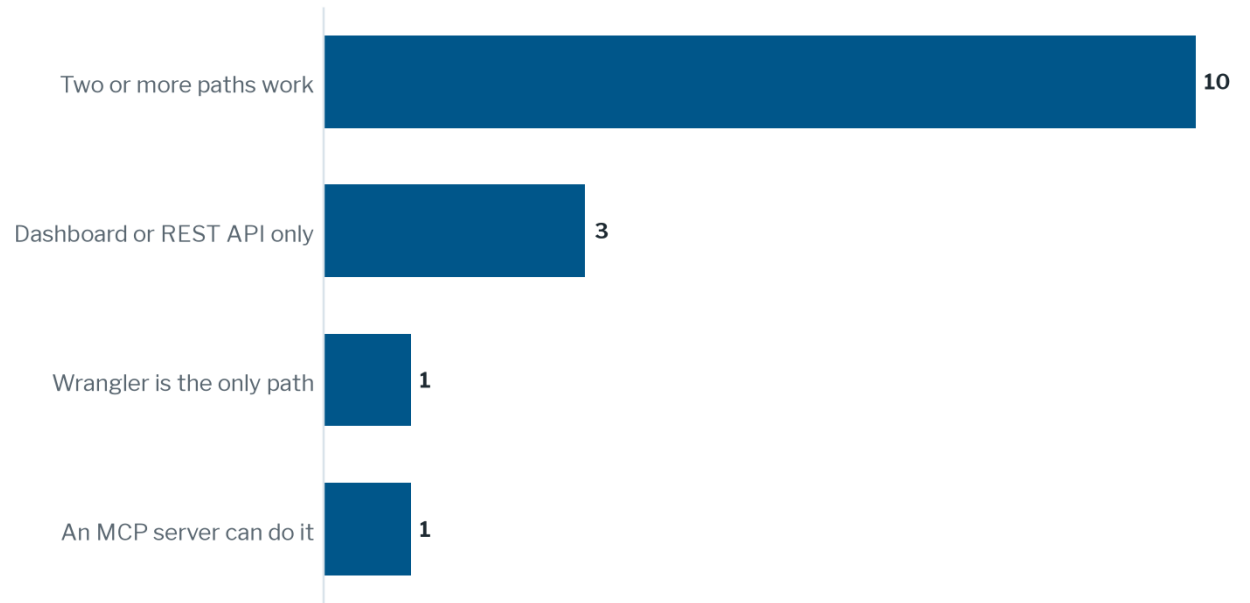
FACT The AI agent integrations are mostly read-only. Cloudflare publishes more than a dozen remote MCP servers covering documentation search, observability, audit logs, DNS analytics, Radar, AI Gateway, browser rendering and Logpush health. The Workers Bindings server is the one that changes account state: it can list, create, get, update and delete KV namespaces, manage D1 databases and run queries against them, manage R2 buckets, and list Workers and read their code. It has no tool for reading or writing an individual KV key-value pair, no tool for deploying a Worker, and no Pages tools at all. ^[21]

ASSESSMENT That gap is inverted from what an operator wants. Creating a namespace happens once. Editing a value inside it happens continuously, and it is exactly the operation you would most like to delegate. The tooling manages the container and not the contents. ^[21]

FACT The HTTP API fills the remainder. Writing a KV value is a PUT to `/accounts/{account_id}/storage/kv/namespaces/{namespace_id}/values/{key_name}`, a route Cloudflare moved from an older path and documented in a July 2026 changelog entry. Editing a project's bindings and environment variables, attaching a custom domain, reading DNS records and creating a Cache Rule are API or dashboard operations rather than Wrangler ones. ^[39]

Configuration step	Dashboard	Wrangler	REST API	MCP
Create the project	Yes	Yes	Yes	No
Upload static assets	Yes	Yes	Awkward	No
Deploy compiled functions	No	Yes	No	No
Set the production branch	No	At create	Yes	No
Create a KV namespace	Yes	Yes	Yes	Yes
Write a value into KV	Yes	Yes	Yes	No
Bind KV to the project	Yes	Config file	Yes	No
Set an environment variable	Yes	Config file	Yes	No
Set or rotate a secret	Yes	Yes	Yes	No
Attach a custom domain	Yes	No	Yes	No
Read or edit DNS records	Yes	No	Yes	No
Add a Cache Rule for a route	Yes	No	Yes	No
Add a scheduled job	Not on Pages	Yes, on Workers	Yes	No
Read deployment logs	Yes	Yes	Yes	Observability only

FIGURE 7 Fourteen setup steps, sorted by which tool can actually perform them



ASSESSMENT The author's classification of the steps in the preceding table, each verified against Cloudflare's published documentation for that tool. The first three categories are exclusive and sum to fourteen; the MCP count overlaps them and is shown for comparison. The uncomfortable number is the last one. [\[14\]](#) [\[15\]](#) [\[16\]](#) [\[17\]](#) [\[21\]](#) [\[39\]](#)

ASSESSMENT Now the argument. Three surfaces with no unified map is a barrier to a person, and the barrier is made of reading. Four documentation sites, several wrong attempts, and a day gone. It is not a barrier at all to an assistant that has already read all four, does not lose patience at step nine of fourteen, and will write the API call as readily as the CLI command. The economics of this specific work changed, and they changed for a narrow and identifiable reason: the correct answer was always written down publicly, and the failure was a lookup failure rather than a design failure.

ASSESSMENT This is the reconciliation of two things that look contradictory. The configuration surface is genuinely awkward, which is why the platform stayed niche. The same awkwardness costs an AI-assisted team almost nothing, which is why it is about to stop being niche. Both statements are true and they are true for the same reason.

ASSESSMENT The limit is judgment and it is severe. Left to choose its own path an assistant picks the plausible one, and on this platform the plausible path is frequently wrong. Upload the folder through the dashboard. Set the secret and expect it to take effect. Restrict the API key by referrer, because that is what the key's settings page suggests. Deploy without the branch flag. Every silent failure catalogued in the next section is a plausible-looking action that produces no error, which means an unsupervised assistant has no feedback signal telling it to try again.

HYPOTHESIS Offered for testing: on a platform where the common failure modes are silent, unsupervised AI configuration produces a higher rate of broken-but-passing deployments than either supervised AI configuration or manual configuration by an inexperienced developer, because the inexperienced human at least stops to check. Confirmed by a controlled comparison of deployment outcomes across those three conditions; refuted if silent-failure rates are comparable. The author is not aware of such a study and does not claim one exists.

ASSESSMENT The skill moved rather than disappeared. It used to be typing the commands. It is now knowing which command should be typed and recognizing when the output is lying to you. If a reader takes one operational habit from this report, take this one: after every deployment, request the function route rather than the homepage, and read the response and the cf-cache-status header. The homepage looks fine whether or not the deployment worked. ^[13]

Six documented behaviors that cost teams their first day

Not complaints. Documented properties of the platform, each verifiable, and knowing them is most of the difference between a working deployment and a broken one that reports success. Five of the six fail silently, which is what makes them expensive.

Behavior	Why it happens	What to do instead
Pages has no scheduled jobs	Cron Triggers are marked unsupported for Pages and supported for Workers in Cloudflare's compatibility matrix	Build on Workers, or deploy a second Worker alongside the Pages project

Behavior	Why it happens	What to do instead
SMTP mail is impractical	Cloudflare blocks outbound TCP on port 25 by default and the runtime has no Node mail stack	Send through an email provider's HTTPS API using fetch
Referrer-restricted API keys always fail	Server-side calls send no Referer header, so a browser-restricted key is rejected every time	Restrict the key by which API it may reach, and keep a separate referrer-restricted key for browser code
Function responses are not edge-cached	The CDN caches on file extension and does not cache HTML or JSON by default, so extensionless routes are skipped	Add an explicit Cache Rule, then verify with the cf-cache-status header
Drag-and-drop upload ships no functions	Cloudflare documents that dashboard drag-and-drop does not support compiling a functions folder	Deploy with the CLI, which performs the compilation the dashboard cannot
A deploy can silently land as a preview	Wrangler infers the branch from the local git repository and only the configured production branch deploys to production	Pass the branch argument explicitly and confirm the deployment against the production alias

FACT On scheduling: the compatibility matrix marks Cron Triggers, Queue Consumers, Rate Limiting, Tail Workers, Logpush, Source Maps, Email Workers, Image Resizing and Gradual Deployments as Workers-only, with Durable Objects listed as unsupported on Pages with a workaround. Every gap runs the same direction. ^[3]

FACT On mail: Cloudflare's TCP sockets documentation states that Workers cannot by default create outbound connections on port 25 to send email to SMTP servers, and blocks sockets to Cloudflare's own ranges, localhost and private addresses. ^[10]

ASSESSMENT A correction worth making rather than repeating a convenient simplification. It is often said that SMTP is impossible here. Port 25 is blocked outright, but 587 and 465 are reachable through the connect API and at least one open-source library exists to use them. The accurate statement is narrower: the runtime is not Node, there is no drop-in mail stack, an app password on port 25 cannot work, and anything on 587 depends on a third-party library plus a compatibility flag. An email provider's HTTPS API is the right answer, not the only possible one. ^{[10] [42]}

ASSESSMENT The rule that generalizes: this runtime is built around fetch. Any protocol that is not HTTPS is either unavailable or awkward enough that the vendor's HTTP API is the right move. Database drivers, queues and mail all follow the pattern. ^[10]

FACT On API keys: Google's documentation distinguishes browser restrictions, which list permitted HTTP referrers, from server restrictions, which list caller IP addresses, and states that only one type of client restriction can be set per key. A Cloudflare function sends no Referer header and has no stable outbound IP to allow-list, so the correct control is the API restriction, limiting which APIs the key can reach. ^[11]

FACT On caching: Cloudflare's cache documentation lists the extensions cached by default and states that the CDN does not cache HTML or JSON by default, with decisions made on file extension rather than MIME type. The `cf-cache-status` header reports the outcome, where DYNAMIC means Cloudflare determined at request time that the asset was not eligible for cache and went to origin without a lookup. ^{[12] [13]}

ASSESSMENT Any reader can check this in one command. Run `curl -sI` against a function route and read the `cf-cache-status` line. DYNAMIC means the function runs on every page view and every run draws on the request quota. HIT means the CDN is absorbing the traffic. That single header separates a site whose cost scales with traffic from one whose cost does not. ^[13]

FACT On deployment targets: the `branch` flag is optional, and Cloudflare documents that Wrangler infers the branch from the repository you are in and adds it implicitly. Only a deployment matching the configured production branch becomes production; everything else becomes a preview at a branch-specific subdomain. Deploying from a feature branch, or from a directory that is not a repository, produces a successful preview while the live site remains untouched. ^{[15] [16]}

Counter-arguments, taken seriously

Five objections deserve space rather than reassurance. One of them is disqualifying for some readers.

Lock-in, and what an exit costs

ASSESSMENT Static assets are not locked in at all: a directory of files deploys anywhere. Functions are genuinely coupled, using Cloudflare's routing convention and context object, and KV, D1, R2 and Durable Objects are reached through bindings that exist nowhere else. Porting a handful of small functions is hours per function, not a rewrite. Porting an application built around Durable Objects is a rewrite, because nothing else has that primitive with those semantics.

ASSESSMENT The heavier coupling is DNS and TLS, because leaving means another domain cutover with the same risk profile as the one that brought you in. Rate this as low lock-in for edge and static workloads, moderate for anything using D1 or R2, and high for anything built on Durable Objects.

FACT One asymmetry in Cloudflare's favor that is easy to miss: R2 is S3-API compatible, and egress from it is free. Getting data out of R2 costs nothing at any volume, which is the opposite of the position the hyperscalers held until regulators moved them in 2024. ^{[8] [55]}

Cloudflare terminates customers over content

FACT In August 2017 Cloudflare terminated the Daily Stormer. In August 2019 it terminated 8chan, with chief executive Matthew Prince writing that the site had proven itself lawless and that the lawlessness had caused multiple tragic deaths. In September 2022 it blocked Kiwi Farms, with Prince describing an unprecedented emergency and immediate threat to human life unlike anything the company had previously seen from a customer. ^{[37] [38] [43]}

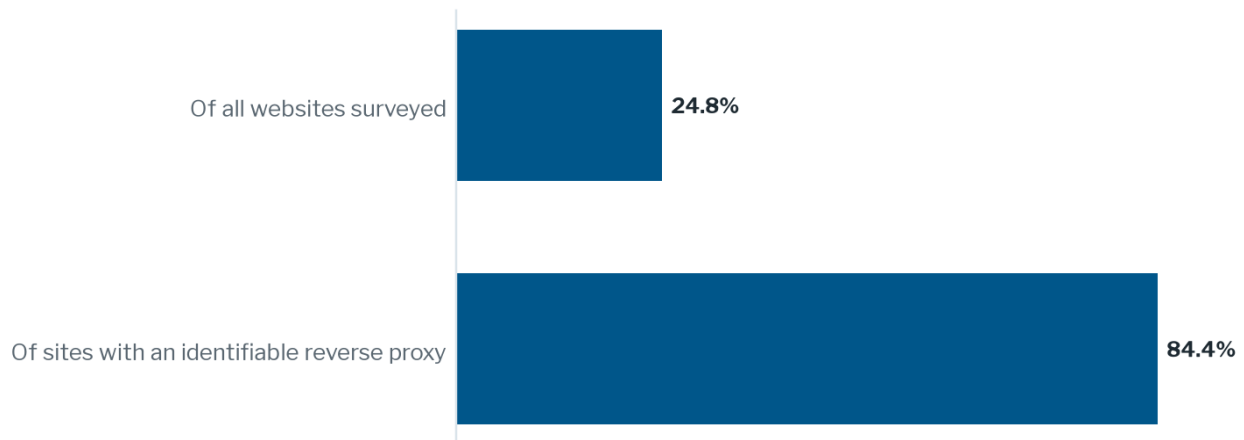
FACT Cloudflare's stated position, from a post by Prince and public policy chief Alissa Starzak, is that terminating security services for content its team personally finds disgusting and immoral would be the popular choice, but that in the long term such choices make it harder to protect content supporting oppressed and marginalized voices. The practice is a high bar, occasionally cleared. ^[38]

ASSESSMENT Do not soft-pedal this. The threshold actually applied is extremely high and no ordinary business is near it. But the decision is Cloudflare's, made under public pressure, on a timescale of days, and the company hosting the site is also resolving its DNS and terminating its TLS. Concentrating all three with a provider that has demonstrated willingness to make content judgments is a real risk for a reader whose subject matter is contested. Such readers should keep DNS at a different registrar and build output in a repository they control, which costs nothing and preserves an exit.

Concentration risk

FACT W3Techs reports Cloudflare in use by 24.8 percent of all websites and by 84.4 percent of websites whose reverse proxy it can identify, as at August 2026. W3Techs surveys the top ten million sites and identifies technology by fingerprint, so the denominator is not the whole web and the second figure excludes sites with no detectable proxy. ^[25]

FIGURE 8 Cloudflare's share of the web, as measured by W3Techs



FACT W3Techs figures for August 2026. Methodology matters: W3Techs surveys the top ten million sites by traffic rank and identifies technology by fingerprint, so the second figure describes the subset where a proxy is detectable rather than the whole web. Directional evidence of concentration, not a precise share. ^[25]

ASSESSMENT Moving one more workload onto Cloudflare makes systemic concentration marginally worse and makes that individual business no worse off than the quarter of the web already there. Both are true and they point different ways. A reader who finds the systemic argument persuasive should act on it where it can matter, which is not the hosting choice for one application.

Outages

FACT On 18 November 2025 Cloudflare's network began failing to deliver core traffic at 11:20 UTC. A database permissions change caused a Bot Management feature file to double in size, exceeding a memory limit and producing HTTP 500 errors across core CDN and security services. Core traffic was largely normal by 14:30 and the incident was resolved at 17:06. Cloudflare's postmortem calls it the company's worst outage since 2019. ^[22]

FACT On 12 June 2025 a two-and-a-half-hour outage was caused by a failure in the third-party cloud storage backend behind Workers KV, taking out Access, WARP, Workers AI and Stream. Cloudflare's postmortem states that while the trigger was a third-party vendor failure, the company is ultimately responsible for its chosen dependencies and how it architects around them, and committed to removing the single-provider dependency. ^[23]

ASSESSMENT Two multi-hour incidents in seven months is a real record and deserves stating without spin. It is also not obviously worse than the hyperscalers deliver, and the honest comparison is not outage against no outage but a published and credibly explained outage against an unexplained one on a support ticket. What a business with no fallback does during an incident is wait, and the plan should say so rather than pretend otherwise. Keep build output in version control, keep the domain registered elsewhere, keep a static status page on a different provider. Two hours of preparation, not an architecture.

Pages is no longer where the platform is going

FACT The Pages documentation states that Workers supports most Pages use cases and offers a broader feature set, that it is Cloudflare's primary platform for building applications, and that new projects should start with Workers. The compatibility matrix confirms it: with a handful of routing exceptions, every feature supported on one product and not the other is supported on Workers. ^{[3] [20]}

ASSESSMENT No deprecation date has been announced and existing projects keep working. Read it as maintenance mode rather than end of life. The practical consequence for this report: every cost argument here applies to Workers with static assets, which has the same free-and-unlimited static request treatment plus Cron Triggers. Start new work on Workers. Existing Pages projects have no urgent reason to move and should plan to eventually. ^{[1] [3] [20]}

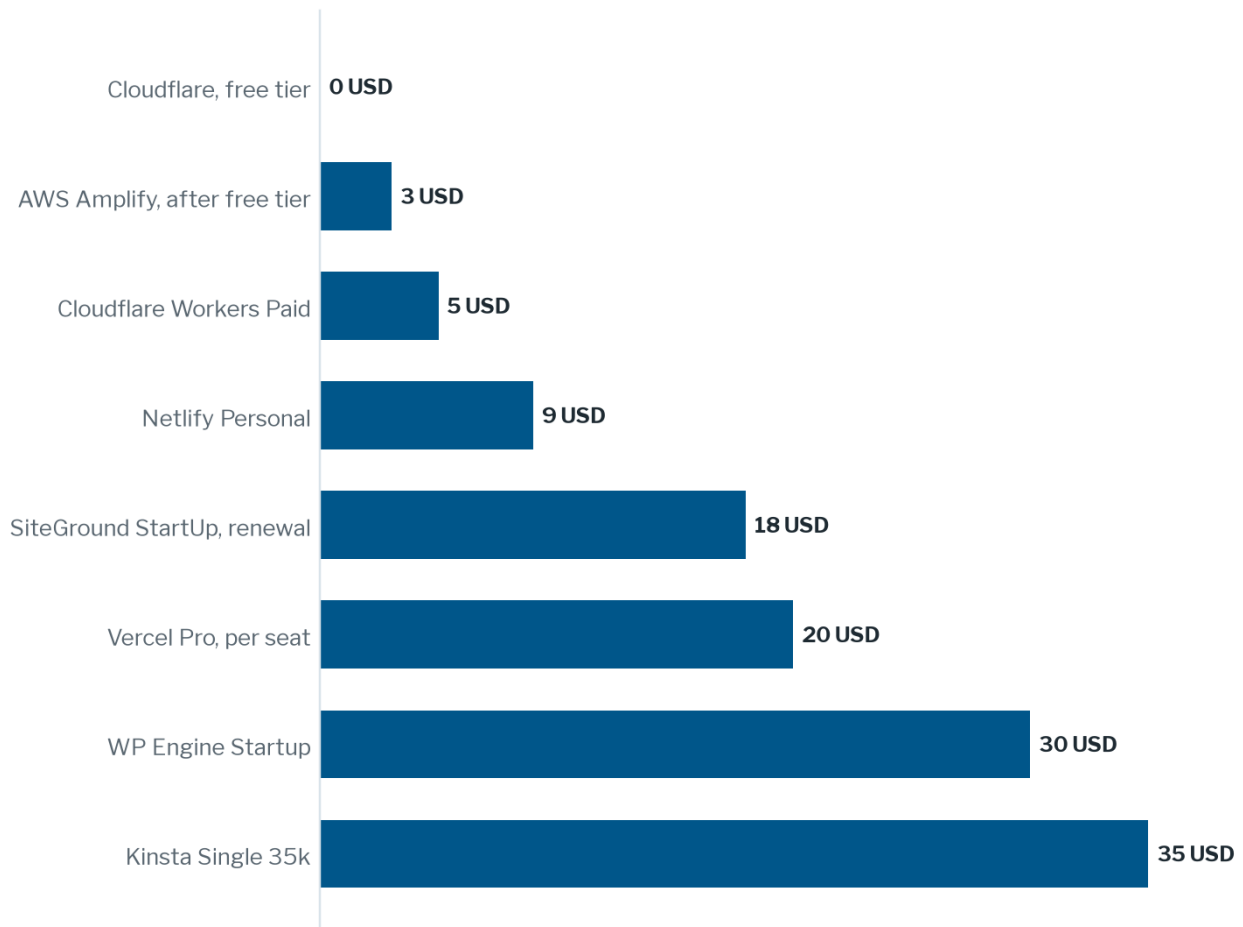
The small-business case, which is the simplest one

Everything above concerns workloads large enough to have an architecture. The clearest case for this platform is the smallest, and it is worth treating separately because the answer is unambiguous and the reader is different.

ASSESSMENT A great many business websites are brochure sites with two or three dynamic behaviors attached: a contact form, a feed from an external service, a newsletter signup. They are billed monthly as though they were applications. They are not applications. The static portion is served free and the dynamic remainder fits inside the free allowance with room to spare. ^{[1] [2]}

ASSESSMENT Take a representative site: 25,000 monthly page views, around forty static assets per view, three function-backed endpoints averaging 10,000 invocations a month between them, eight gigabytes of outbound traffic. Every static request is free. Ten thousand invocations against a 100,000-per-day allowance is roughly three percent of a single day's quota spread across a month. The eight gigabytes are not billed at all. ^{[1] [2]}

FIGURE 9 Monthly list price for the same brochure site



ASSESSMENT Published list prices as at 22 August 2026 applied to one site of 25,000 monthly page views, 10,000 monthly function calls and 8 GB of egress. The Amplify figure is the author's arithmetic on published per-GB and per-minute rates after the introductory free tier expires; every other figure is a plan price taken from the vendor. Excludes the domain, which all of them need and none include. ^{[1] [26] [27] [28] [29] [30] [31]}

FACT The structural ceilings are not reachable by accident at this size. A free-plan project may hold 20,000 files at up to 25 MiB each, with 500 builds a month and one build running at a time; paid plans raise the file count to 100,000. A five-page site with an image library is nowhere near any of these. ^[9]

FACT The cheapest conventional alternative is a small VPS, and it is the least complete. DigitalOcean's smallest Basic Droplet is 4 dollars a month for one vCPU and 512 MiB of memory

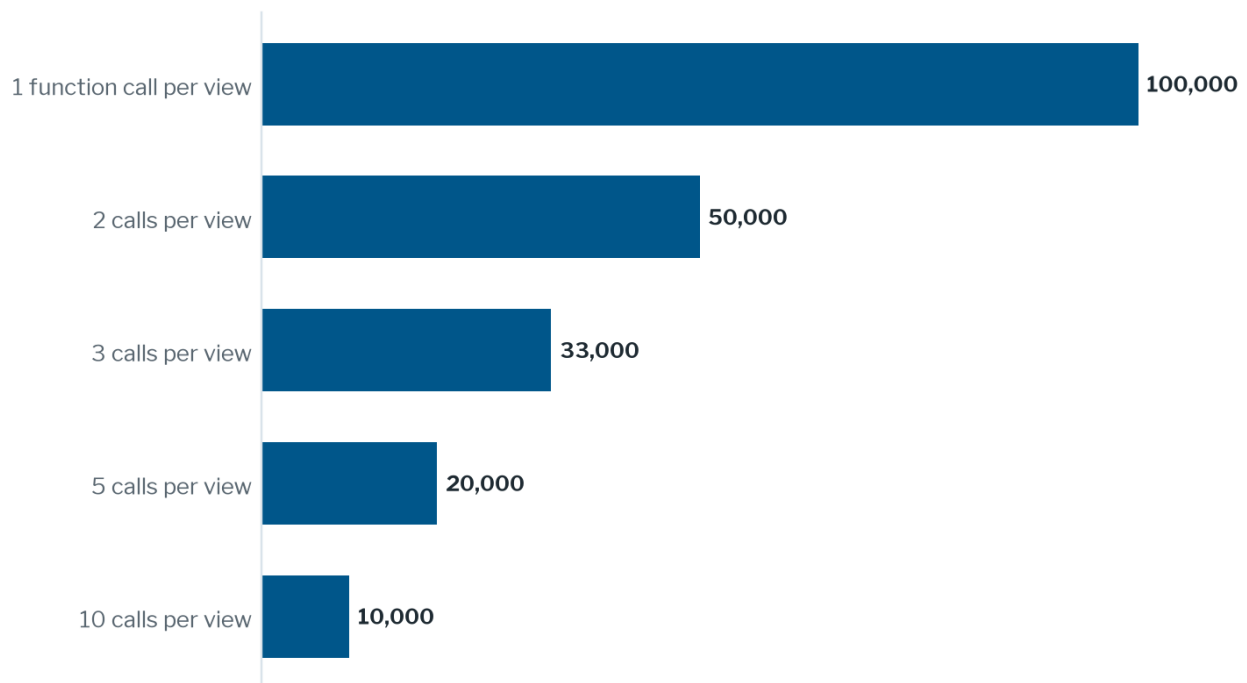
with 500 GiB of transfer; the 1 GiB tier is 6 dollars with a terabyte. Overage beyond the team's pooled allowance is one cent per gibibyte. That price buys a machine, not a running website: no TLS renewal, no patching, no monitoring, no backups. [\[32\]](#) [\[33\]](#)

FACT Shared hosting's headline price is a first-term promotion. SiteGround's own pricing page shows StartUp at 2.99 dollars promotional against 17.99 regular, GrowBig at 4.99 against 29.99, GoGeek at 7.99 against 44.99. The renewal price is the real price, roughly six times the number most buyers remember agreeing to. [\[31\]](#)

FACT Managed WordPress prices by traffic. WP Engine Startup is 30 dollars a month on annual billing for one site, 25,000 monthly visits and 75 GB of bandwidth. Kinsta's single-site plan starts at 35 dollars for 35,000 visits and 125 GB of CDN traffic. Both scale the bill with visit count, which means marketing success raises the hosting bill. [\[29\]](#) [\[30\]](#)

FACT Vercel's Hobby plan is free and Vercel states it is for personal, non-commercial use, which disqualifies it here. Pro is 20 dollars per seat including ten million edge requests and one terabyte of transfer, with overage at 15 cents per gigabyte. Netlify moved to credits: 300 a month free with a hard stop, bandwidth at 20 credits per gigabyte, both bandwidth and compute rates doubled in an April 2026 update. [\[26\]](#) [\[27\]](#) [\[41\]](#)

FIGURE 10 Daily page views that fit inside the free 100,000-request allowance



ASSESSMENT The author's arithmetic on Cloudflare's published free-plan limit of 100,000 Workers requests per day. Static asset requests are excluded because they are not metered, so a page with forty images and one API call counts as one request. Assumes no edge caching of function responses, which is the default and the conservative case. [\[1\]](#) [\[2\]](#)

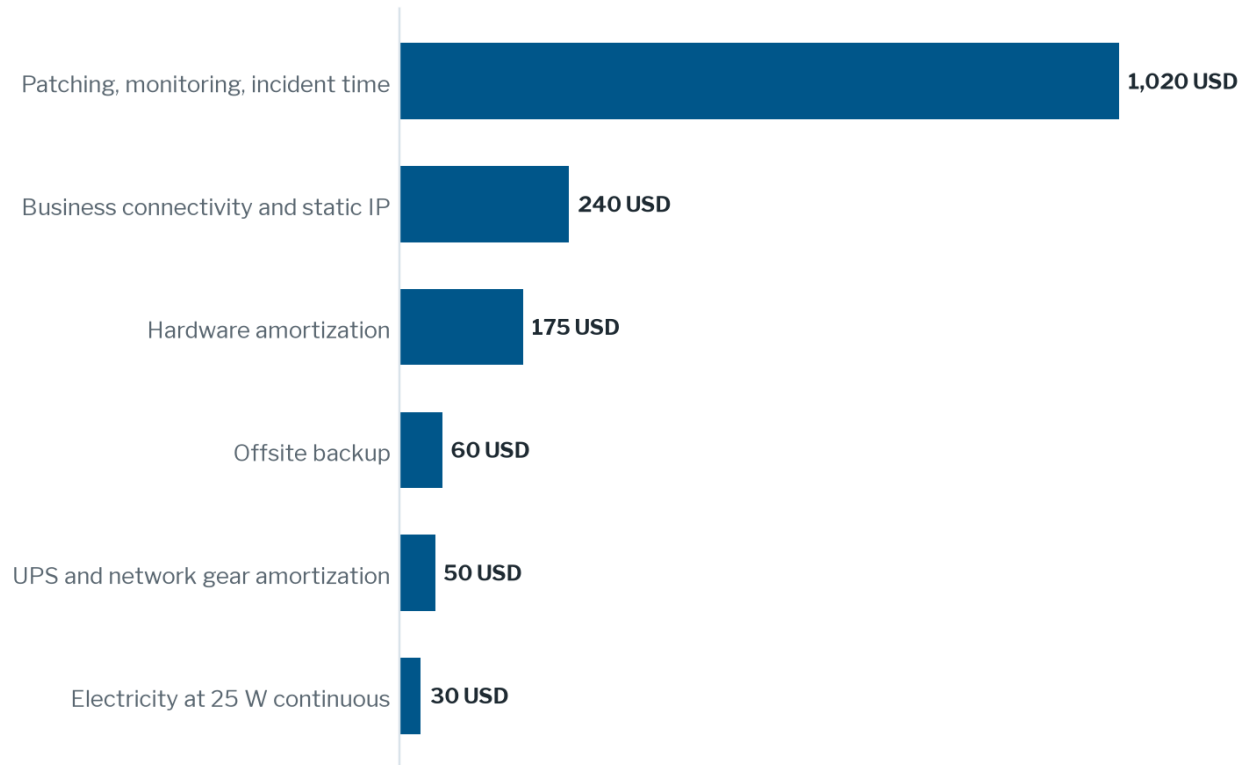
FACT The failure mode on the free tier is worth knowing. Exceeding 100,000 requests in a day does not generate a bill; it returns error 1027 and rate-limits the site until the quota resets at midnight UTC. Practitioner reports on Cloudflare's community forum describe hitting this without realizing Workers were in the request path. The site goes down rather than the invoice going up, which is a real tradeoff to choose deliberately. ^[40]

ASSESSMENT Contrast with the alternative failure mode. In February 2024 a developer received a bill of 104,000 dollars from Netlify for a static site after a sustained download event. It was reduced to 5,225 and then waived once the story spread. The point is not that Netlify behaved badly, because it eventually behaved well. The point is that the exposure existed and its resolution depended on the story going viral. ^{[35] [36]}

The self-hosted number nobody computes

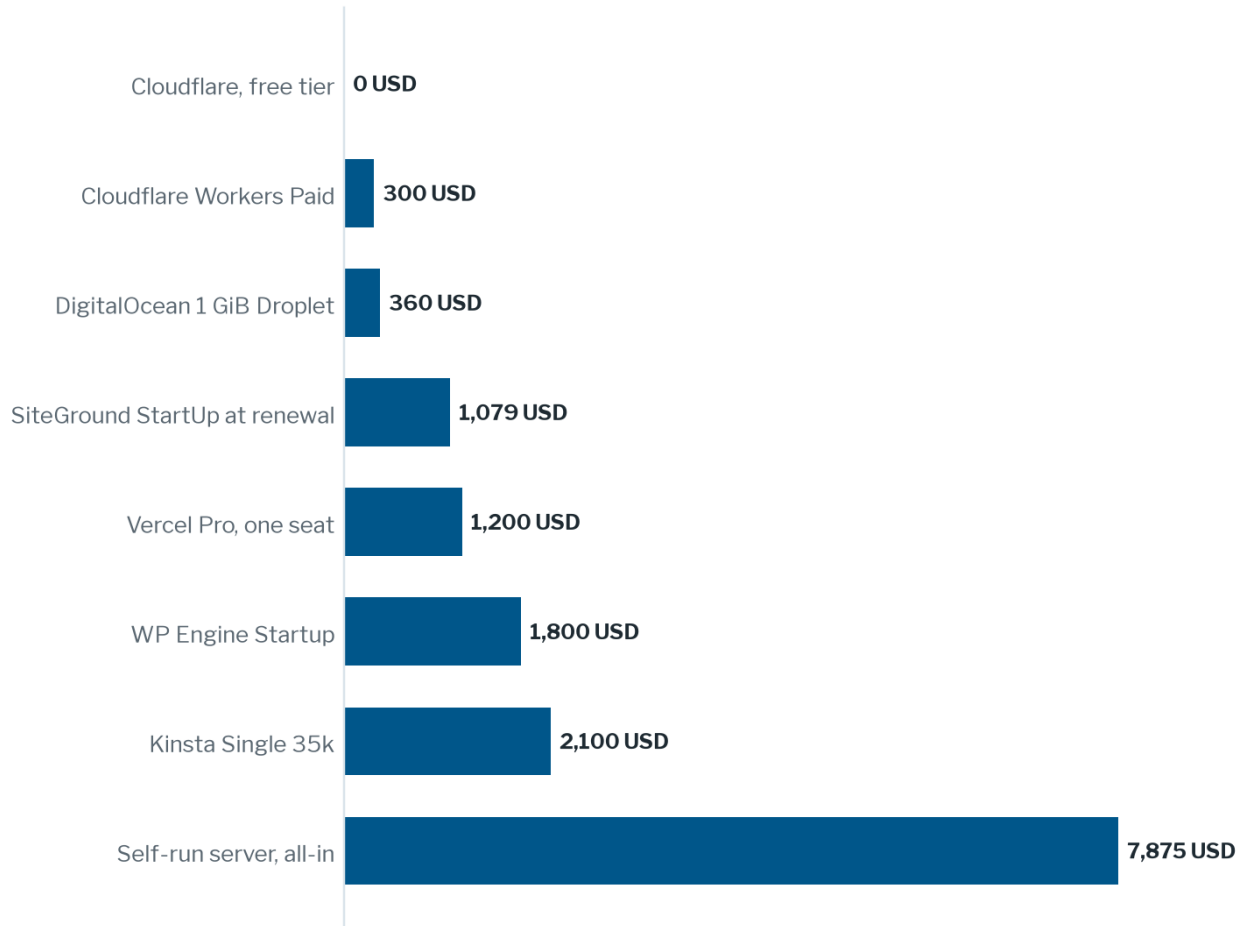
ASSESSMENT The comparison omitted from every hosting roundup is the true annual cost of running a small always-on server yourself, because its largest component never appears on a statement. The estimate below is the author's, built from published input prices, and it is deliberately conservative: a modest machine, low power draw, one hour of attention a month.

FACT The one input from an official statistical source: the US Energy Information Administration reports an average retail electricity price to commercial customers of 13.54 cents per kilowatt-hour for May 2026, the most recent month published. A machine drawing 25 watts continuously uses about 219 kilowatt-hours a year, roughly 30 dollars at that rate. ^[34]

FIGURE 11 Annual all-in cost of one self-run always-on web server

ASSESSMENT The author's estimate. Labor assumes 12 hours a year at a loaded rate of 85 USD, which is one hour a month and almost certainly low for a machine exposed to the internet. Hardware assumes a 700 USD small server over four years. Electricity uses the EIA commercial average for May 2026. Connectivity assumes a 20 USD monthly uplift for a static IP and business terms. Total 1,575 USD a year, of which 65 percent is time nobody invoices for. ^[34]

ASSESSMENT Twelve hours a year is the optimistic case and it still makes labor the largest line by a factor of four. A business that thinks a box in a closet is free is spending roughly 1,575 dollars a year to avoid a bill between zero and 360. That is before the outage it will eventually have, and before the fact that whoever set it up will one day leave.

FIGURE 12 Five-year cost of hosting one marketing site, at list price

ASSESSMENT Sixty months at each vendor's published rate as at 22 August 2026, holding prices flat, which none of them will. SiteGround uses the renewal rate, not the promotion. The DigitalOcean and self-run figures are not like-for-like: the Droplet price excludes all operational labor, and the self-run figure carries the previous chart over five years. [\[26\]](#) [\[29\]](#) [\[30\]](#) [\[31\]](#) [\[32\]](#) [\[34\]](#)

Implications

Seven audiences read this differently and blending them would serve none of them.

For founders and owners of small and mid-sized businesses

ASSESSMENT The question is not whether your website is cheap. It is what the hosting line item is paying for. If the site is a set of pages that change when you ask someone to change them, plus a contact form, you are paying an application price for a document: between 200 and 500 dollars a year, indefinitely, for capability the site does not use.

ASSESSMENT The saving is not life-changing and should not be sold as such. A few hundred dollars a year per site against a one-off cost of one to three days of developer time. Payback is roughly a year on a single site and much shorter across a portfolio. What makes it worth doing is not the money; it is that the money was buying nothing you can name.

For CTOs and engineering leaders

ASSESSMENT The limits you need to judge safety are short and documented. 128 MB per isolate, and no plan raises it. Ten milliseconds of CPU free, 30 seconds default and five minutes maximum on paid, 15 minutes for scheduled and queue work. The runtime is fetch-shaped and not Node. Cron, queue consumers and rate limiting are Workers features and not Pages features. Read the platform limits page and the compatibility matrix once and you will know within twenty minutes whether a workload qualifies. [\[3\]](#) [\[10\]](#) [\[49\]](#)

ASSESSMENT The risk to manage is not the platform, it is the silent failure surface, and your existing deployment checks will not catch it. Add one gate: after deploy, request a function route and assert on the response body, not a homepage health check. That single assertion catches the drag-and-drop failure, the preview-instead-of-production failure and the stale-secret failure at once.

ASSESSMENT On the build-versus-migrate question, the defensible position is neither wholesale migration nor dismissal. Move the egress-heavy edge of the estate, which is where the arithmetic is overwhelming, and leave the warehouse, the training jobs and the large-memory services alone. A team that tries to move everything will fail on the 128 MB ceiling and conclude the platform is a toy. A team that moves nothing keeps paying a thousand dollars a month for bytes.

ASSESSMENT Start new work on Workers rather than Pages. Cloudflare says so in its own documentation and the feature matrix confirms it. [\[3\]](#) [\[20\]](#)

For enterprise architects

ASSESSMENT Treat this as a fourth provider for a specific tier of the estate rather than a replacement for an incumbent. The compliance floor clears most procurement bars: SOC 2 Type II, ISO 27001, ISO 27701 and PCI DSS Level 1 assessment. Jurisdiction-restricted Durable Objects give a real residency mechanism for object state. Neither of those makes it a drop-in substitute for a hyperscaler holding a specific vertical authorization your contract names. [\[50\]](#) [\[54\]](#)

ASSESSMENT The architectural pattern that works: Cloudflare at the edge and for anything customer-facing, the incumbent for the data platform, connected deliberately rather than chattily. The failure pattern is a design that moves data back and forth across the boundary on every request, which converts an egress saving into an egress bill on the other provider's side of the line.

ASSESSMENT The number to take to a review is not the list-price gap in this report, because your effective hyperscaler rate is not list. It is the egress line on last month's actual invoice for customer-facing traffic, which is the only figure that survives contact with a procurement team that knows what it already pays.

For finance and operations leaders

ASSESSMENT Four questions separate load-bearing spend from residue. Does this workload hold data that changes when nobody deploys it? If not, the plan is priced for capability it does not use. What does the plan cost at renewal rather than at signup? SiteGround's own page shows a 2.99

promotional rate against 17.99 regular, a six-fold step nobody budgets. Is the bill indexed to traffic or to headcount? Both are incentives worth noticing. And what is the egress line, separately, on the cloud invoice? [\[29\]](#) [\[30\]](#) [\[31\]](#)

ASSESSMENT The two line items to scrutinize hardest are the ones nobody defends. Any server the organization owns: roughly 1,575 dollars a year on the estimate here, about 65 percent of it internal time appearing in no ledger. And any per-seat platform licence for an internal application, where the charge scales with how many colleagues use the tool rather than with what the tool does. [\[34\]](#) [\[52\]](#) [\[53\]](#)

For platform and low-code owners

ASSESSMENT If you own a Retool, Power Apps or similar estate, the exposure is not price, it is the pricing dimension. Per-seat charging is most painful exactly where internal tools succeed, which is broad adoption, and an AI-plus-repository alternative bills on usage instead. Audit the estate by cost per application rather than in total, because the aggregate number hides the handful of widely-adopted tools carrying most of the spend and offering the cleanest migration case.

ASSESSMENT Do not migrate the complicated ones. Applications with intricate permissions and a small user base are what these platforms are good at and the seat cost there is defensible. The candidates are simple-logic, many-user applications, which is also where a generated codebase is least risky to own.

For agencies and freelance developers who resell hosting

ASSESSMENT Recurring revenue from reselling hosting on brochure sites is exposed and the exposure is not gradual. The moment a client asks why the site costs 35 dollars a month when the underlying delivery is free, the conversation is about margin rather than service. The defensible answer is not to argue the platform is worse; it is to be the person who moved them and to bill for what actually requires judgment.

ASSESSMENT The work that survives is the work in the second half of this report: knowing which of three tools each step needs, catching the silent failures, owning the DNS cutover, being reachable when something breaks. That earns a retainer on its own terms and does not depend on marking up a commodity. An agency that repositions from reselling capacity to selling configuration and care comes out ahead. One that waits for the client to notice does not.

For developers who have never deployed serverless functions

ASSESSMENT Internalize one thing before your first deployment: the command-line tool is the only thing that can deploy, because functions must be compiled and no dashboard action does that. Second, pass the branch argument explicitly every time, even when it feels redundant. Third, after every deploy, curl the function route and read cf-cache-status. Those three habits skip most of a first day of confusion. [\[13\]](#) [\[14\]](#) [\[15\]](#)

ASSESSMENT Use an AI assistant and supervise it. It knows the documentation better than you do and it will still choose the dashboard upload if you let it, because that is the answer that looks right. Tell it which tool to use for each step, and check the output rather than the exit code.

Scenarios to 2028

No point forecast, because pricing is a vendor decision that can change on a Tuesday. Four structured futures with subjective probabilities and, more usefully, the earliest observable sign. The probabilities are the least reliable content here. The tripwires are the most useful.

SCENARIO A The egress gap holds and Cloudflare takes the edge tier 50 percent

Cloudflare keeps no-egress-billing, the hyperscalers keep charging for operational egress, and Cloudflare wins the customer-facing edge of enterprise estates without displacing anyone's data platform.

Consequence. The arithmetic in this report holds through 2028 and the migration case strengthens as estates grow.

Earliest visible sign. Hyperscaler pricing pages unchanged on operational egress through 2027 while Cloudflare ships more enterprise-tier controls, particularly around residency and observability.

SCENARIO B A hyperscaler drops operational egress to defend the tier 20 percent

One of the three cuts or eliminates charges for ordinary outbound traffic, not just exit traffic, to stop losing edge workloads.

Consequence. The central argument of this report weakens sharply and the comparison reverts to service breadth, where Cloudflare loses.

Earliest visible sign. Any change to a hyperscaler's internet egress rate card that is not framed as a switching or exit provision. Watch the rate, not the press release.

SCENARIO C Cloudflare introduces egress billing 15 percent

Cloudflare begins charging for data transfer in some form, including a generous one, or begins metering requests to static assets.

Consequence. The report's core claim collapses. Every comparison reverts to a per-request and per-gigabyte contest Cloudflare would probably still win but not overwhelmingly.

Earliest visible sign. Any change to the sentence stating that requests to static assets are free and unlimited, or to R2's statement that egress is free. These two lines are worth re-reading before acting on this document.

SCENARIO D Low-code vendors reprice away from seats 15 percent

Facing AI-generated alternatives, the major low-code platforms shift from per-seat to usage-based pricing.

Consequence. The second argument in this report loses most of its force, and the remaining case for these platforms as hosted runtimes with governance becomes considerably stronger.

Earliest visible sign. A major low-code vendor announcing usage-based or consumption pricing as the default rather than as an enterprise option, or quietly removing per-external-user charges.

Questions to ask about your own bill

None of these needs a vendor conversation or a consultant. All can be answered in an afternoon by someone with access to the invoice.

What is the egress line, on its own, for customer-facing traffic? Every hyperscaler invoice separates it and almost nobody reads it separately. This is the single number that decides whether the rest of this report is relevant to you.

What is the renewal rate on this plan, as opposed to the rate we signed at? Find it in the fine print on the vendor's pricing page, not on the invoice, because the invoice will not tell you what happens next year.

Is any bill here indexed to headcount rather than to work performed? Per-seat platform licences and per-visit hosting plans both charge you for success.

How many parts of this application actually run code on a server? Count them. On most business applications the answer is small, and each one is a few dozen lines.

If we run a server ourselves, who patches it, how often, and what is that person's loaded hourly cost? Multiply. The number is usually larger than the hosting bill it was meant to avoid.

For each managed service on the invoice, does a Cloudflare primitive cover it, or is this one of the genuine gaps? Warehouse, training and large-memory workloads are gaps. Most of the rest is not.

What happens to this workload during a four-hour provider incident, and what is the actual plan? If the answer is that we would wait, write that down. A documented decision to accept a risk is different from an undocumented failure to consider it.

The migration, honestly

Task	Typical hours	Principal risk
Produce static build output from the current site	2 to 8	A site with a live CMS is a rebuild, not an export; scope this before quoting
Port form handling to a function plus an email API	1 to 3	Spam handling must be rebuilt; the old host's protections do not travel
Port a third-party feed with cache-then-refresh	2 to 4	A wrong freshness threshold shows stale data or burns the upstream quota
Create namespaces, bindings, vars and secrets	1 to 2	Confusing secrets with KV, then wondering why a change did not take effect

Task	Typical hours	Principal risk
Port stateful behavior to Durable Objects	8 to 40	Genuine application work; the only task here that can become a project
First deployment and verification	1 to 2	Every silent failure lands here; verify the function route, not the homepage
Custom domain and DNS cutover	1 to 3	The only task that can take the live site down; do it last and off-hours
Cache Rules and cf-cache-status verification	1	Skipping it leaves the function running on every page view and the cost model wrong
Redirects, headers and 404 handling	1 to 3	Losing existing URLs costs search rankings, which costs more than hosting ever did

ASSESSMENT For a brochure site, ten to twenty-six hours, so one to three working days for a developer who knows the platform and materially longer for one who does not. At 85 dollars loaded that is 850 to 2,200 dollars against a saving of 200 to 500 a year on one site. Payback of one to four years, which is honest and unexciting. Across an agency portfolio of thirty sites, where the platform knowledge is acquired once and per-site work drops to a few hours, the arithmetic changes completely.

ASSESSMENT For an enterprise workload the migration hours are not the decision. The decision is the egress line, and if that line is four figures a month the migration pays for itself before the security review finishes. If it is two figures, do not start.

ASSESSMENT Do the domain cutover last and separately, after the new deployment is live and verified on its provider subdomain, with the old host still paid up for a month. That single sequencing decision removes most of the risk in the entire exercise.

What this document cannot answer

The boundaries are wide and should be read before anything here is acted on.

It cannot tell you what you actually pay. Every hyperscaler figure here is list price. Committed-spend agreements, enterprise discount programs, private pricing and credits are not public, and a reader with any of them has an effective rate this report cannot see. The gaps shown are upper bounds.

It contains no performance measurement. No latency was benchmarked, no page-speed comparison run, and no claim about anything being faster after migration appears anywhere. Cloudflare markets edge delivery on speed; that claim is not evaluated here and should not be inferred.

It does not evaluate reliability quantitatively. Two Cloudflare incidents are described because Cloudflare published postmortems for them. No comparable analysis of AWS, Azure or Google incident history was performed, so the outage section establishes that Cloudflare has outages

and explains them, and does not establish whether it has more or fewer than the alternatives. That asymmetry favors the incumbents and the reader should discount accordingly.

It says little about high-traffic application economics beyond the egress line. Request-level cost modeling at scale needs a real model with real traffic shapes, and the free-tier headroom chart is a demonstration of scale rather than a capacity plan.

On residency and compliance it establishes that mechanisms exist and not that they satisfy any particular obligation. Jurisdiction-restricted Durable Objects pin object storage. They do not constitute a residency guarantee for an entire application, and a FedRAMP-Moderate data center jurisdiction is not a FedRAMP platform authorization. Anyone with a hard obligation must verify it directly.

It does not assess support quality on any platform named. Decisions frequently turn on whether someone answers at 2am, and free tiers of developer platforms generally do not include anyone who will. That is a real cost and this report does not price it.

On the low-code argument, the evidence is thinner than on the cost argument and the report says so with a HYPOTHESIS label rather than a conclusion. Two vendors' pricing pages and the observation that both now lead with AI generation are not sufficient to establish that the category is repricing. That would need revenue disclosure by pricing dimension, which no private low-code vendor publishes.

It does not measure how often the silent failures described here reach production, because no such data exists publicly. Answering it would need a controlled study of deployment outcomes, which would be worth doing and which nobody appears to have done.

Pricing was captured on one day. Every vendor named changes prices, and several changed within the last eighteen months. Re-check every number before it drives a decision.

Half-life of this assessment

Decaying within six months: every specific price. Netlify's credit conversion rates, which moved in April 2026. The Cloudflare MCP server tool inventories, which are under active development and where the gap identified here is exactly the kind of thing that gets filled. Cloudflare Containers pricing, which is new enough to move.

Decaying within twelve months: the status of Cloudflare Pages as a supported product; the recommendation to start new projects on Workers is already published and whether it hardens into a deprecation notice is the thing to watch. The free-tier allowances, which are vendor decisions with no contractual protection. And the low-code pricing models, which are under the most competitive pressure of anything in this report.

Decaying within twenty-four months: the configuration friction in the second half. The three-surface split is a consequence of Pages predating current Workers tooling, and consolidation would remove much of it. If Cloudflare ships one tool handling deployment, bindings, domains and

cache rules together, the configuration-gap argument weakens considerably, though the judgment argument does not.

Architectural, and unlikely to change: that serving bytes costs a network operator close to nothing at these volumes, which is why one provider can give it away and why the others charging nine to twelve cents is a pricing decision rather than a cost recovery. That a brochure site is not an application. That the failure modes of a distributed serverless platform are quieter than those of a single server, because there is no log file on one machine to read. And that the scarce skill in infrastructure work is knowing which action to take rather than how to type it, which was true before AI assistants and is more true after.

Limitations

The author is an independent practitioner with no commercial relationship, reseller arrangement, affiliate link or sponsorship involving Cloudflare, Amazon, Microsoft, Google, Retool, Vercel, Netlify, WP Engine, Kinsta, SiteGround or DigitalOcean. No vendor reviewed this report before publication and none was asked to.

The single largest weakness is the list-price problem, stated in the scope section and repeated here because it affects every comparison: enterprise customers do not pay list, and no public source discloses what they do pay. A reader with a committed-spend agreement should treat every gap here as an upper bound.

The evidence base is heavily weighted toward vendor primary sources, because pricing and platform behavior are things vendors document and third parties mostly repeat. That is appropriate for what a vendor charges and what its platform does. It is a weakness anywhere the question is what the experience is like, and this report does not attempt that question.

The workload comparisons use synthetic profiles rather than measured traffic. Ten terabytes of egress, ten million requests, one terabyte stored: these are round numbers chosen to be legible, not observations of any real system. Real workloads have shapes that change the arithmetic, particularly around request concurrency and cache hit rates.

The self-hosting estimate is the softest number here. Its largest component is a labor assumption, twelve hours a year at 85 dollars. A reader who genuinely spends two hours would cut the total by more than half; one who has spent a weekend on a compromised server would double it. It is an order of magnitude and an argument that the number is not zero, which is the number most comparisons use.

The low-code section rests on two vendors' published pricing and on the observable fact that both now lead with AI generation. It does not include OutSystems, Mendix, Appian or ServiceNow, because none publishes a price list that can be cited primarily, and third-party figures for them were available but are commercially interested and were not used. The category conclusion is correspondingly labeled as a hypothesis.

The documented platform behaviors were verified against Cloudflare's and Google's published documentation, not reproduced independently in a fresh account for this report. Each names the page it came from so a reader can check both the documentation and their own deployment.

Practitioner complaints cited here come from self-selected forum posts and one widely circulated billing incident. People who have a bad experience post; people who do not, do not. These establish that a failure mode exists and is encountered, never how often.

Sources

- [1] Cloudflare. "Pricing." Cloudflare Workers documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [2] Cloudflare. "Pricing." Cloudflare Pages Functions documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [3] Cloudflare. "Compatibility matrix." Workers static assets documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [4] Cloudflare. "Limits." Workers KV documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [5] Cloudflare. "Pricing." Workers KV documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [6] Cloudflare. "Data security." Workers KV documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [7] Cloudflare. "Pricing." D1 documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [8] Cloudflare. "Pricing." R2 documentation, including the statement that egress is free. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [9] Cloudflare. "Limits." Cloudflare Pages documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [10] Cloudflare. "TCP sockets." Workers runtime API documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [11] Google. "Adding restrictions to API keys." Google Cloud API Keys documentation. Accessed 22 August 2026. *Vendor primary* docs.cloud.google.com
- [12] Cloudflare. "Default cache behavior." Cache documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [13] Cloudflare. "Cloudflare cache responses." Cache documentation, cf-cache-status values. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [14] Cloudflare. "Direct Upload." Cloudflare Pages documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [15] Cloudflare. "Pages commands." Wrangler documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [16] Cloudflare. "Configuration." Pages Functions Wrangler configuration documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [17] Cloudflare. "Bindings." Pages Functions documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [18] Cloudflare. "Routing" and "API reference." Pages Functions documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [19] Cloudflare. "Cache." Workers runtime API documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [20] Cloudflare. "Cloudflare Pages" overview, carrying the notice recommending Workers for new projects. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [21] Cloudflare. mcp-server-cloudflare repository, server list and Workers Bindings tool inventory. Accessed 22 August 2026. *Vendor primary* github.com
- [22] Cloudflare. "Cloudflare outage on November 18, 2025." Cloudflare Blog, 18 November 2025. *Vendor primary* blog.cloudflare.com

- [23] Cloudflare. "Cloudflare service outage June 12, 2025." Cloudflare Blog, 12 June 2025. *Vendor primary* blog.cloudflare.com
- [24] Cloudflare. "Service-specific terms: Application Services," CDN restrictions on video and large files. Accessed 22 August 2026. *Vendor primary* cloudflare.com
- [25] W3Techs. "Usage statistics and market share of Cloudflare, August 2026." Q-Success. Surveys the top ten million sites; fingerprint-based identification. *Third-party survey* w3techs.com
- [26] Vercel. "Pricing." Accessed 22 August 2026. *Vendor primary* vercel.com
- [27] Netlify. "Pricing." Accessed 22 August 2026. *Vendor primary* netlify.com
- [28] Amazon Web Services. "AWS Amplify Pricing." Accessed 22 August 2026. *Vendor primary* aws.amazon.com
- [29] WP Engine. "Plans and pricing." Accessed 22 August 2026. Prices on annual billing. *Vendor primary* wpengine.com
- [30] Kinsta. "Managed WordPress hosting plans." Accessed 22 August 2026. *Vendor primary* kinsta.com
- [31] SiteGround. "Web hosting," showing promotional and regular monthly rates. Accessed 22 August 2026. *Vendor primary* siteground.com
- [32] DigitalOcean. "Droplet pricing." Accessed 22 August 2026. *Vendor primary* digitalocean.com
- [33] DigitalOcean. "Bandwidth billing," overage rate and pooled transfer allowance. Accessed 22 August 2026. *Vendor primary* docs.digitalocean.com
- [34] US Energy Information Administration. "Electric Power Monthly," Table 5.3, average price of electricity by end-use sector, May 2026. Preliminary. *Government primary* eia.gov
- [35] "Netlify just sent me a \$104k bill for a simple static site." Hacker News discussion, February 2024. *Practitioner aggregate (self-selected)* news.ycombinator.com
- [36] Cybernews. "User got a \$104K bill from hosting provider: I thought it was a joke." 2024. *Independent press* cybernews.com
- [37] Cloudflare. "Why We Terminated Daily Stormer." Cloudflare Blog, 16 August 2017. *Vendor primary* blog.cloudflare.com
- [38] Cloudflare. "Blocking Kiwifarms." Cloudflare Blog, 3 September 2022. *Vendor primary* blog.cloudflare.com
- [39] Cloudflare. "KV legacy namespace routes deprecation." Cloudflare changelog, 15 July 2026. *Vendor primary* developers.cloudflare.com
- [40] Cloudflare Community forum threads on error 1027 and free-plan request limits, 2021 to 2026. Self-selected reports, used only to establish that the failure mode is encountered. *Practitioner aggregate (self-selected)* community.cloudflare.com
- [41] Netlify. "Pricing updates for Credit-based plans." Netlify changelog, 14 April 2026. *Vendor primary* netlify.com
- [42] zou-yu. worker-mailer, an SMTP library for Cloudflare Workers. GitHub. Cited to establish that ports 587 and 465 are reachable, not as a recommendation. *Practitioner, open source* github.com
- [43] Cloudflare. "Terminating Service for 8Chan." Cloudflare Blog, 5 August 2019. *Vendor primary* blog.cloudflare.com
- [44] Amazon Web Services. "Amazon S3 Pricing," storage and data transfer out rates. Accessed 22 August 2026. *Vendor primary* aws.amazon.com
- [45] Microsoft. "Bandwidth pricing," Azure data transfer out tiers. Accessed 22 August 2026. *Vendor primary* azure.microsoft.com
- [46] Google. "VPC network pricing," Premium Tier internet data transfer out rates by destination. Accessed 22 August 2026. *Vendor primary* cloud.google.com
- [47] Amazon Web Services. "AWS Lambda Pricing," x86 request and duration rates and free tier. Accessed 22 August 2026. *Vendor primary* aws.amazon.com
- [48] Cloudflare. "Pricing." Durable Objects documentation, including SQLite storage backend rates. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [49] Cloudflare. "Limits." Workers platform documentation, CPU time, memory and subrequest limits. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [50] Cloudflare. "Data location." Durable Objects documentation, jurisdictions and location hints. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [51] Cloudflare. "Pricing." Containers documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [52] Retool. "Pricing," per-builder, per-internal-user and external user rates. Accessed 22 August 2026. *Vendor primary* retool.com
- [53] Microsoft. "Power Apps pricing," per-user rates and Dataverse capacity add-on. Accessed 22 August 2026. *Vendor primary* microsoft.com

- [54] Cloudflare. "Cloudflare Certifications." Cloudflare Blog. Accessed 22 August 2026. *Vendor primary* blog.cloudflare.com
- [55] The Register. "Microsoft ends Azure egress fees for customers leaving," 13 March 2024. Used for the sequence and timing of the 2024 waivers by Google, AWS and Microsoft. *Independent press* theregister.com
- [56] Cloudflare. "Compliance documentation." Cloudflare Fundamentals documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [57] Cloudflare. "Pricing." Queues documentation. Accessed 22 August 2026. *Vendor primary* developers.cloudflare.com
- [58] Bubble. "Pricing." Accessed 22 August 2026. Cited for the positioning of AI generation relative to the visual editor, not for a price figure. *Vendor primary* bubble.io

Rights and permitted use

© 2026 David Soden, davidsoden.com/market-assessment. All rights reserved.

This document, including its structure, analysis, findings, evidence classification, and framing, is the original work of David Soden, published at davidsoden.com/market-assessment.

It is published for public reading. You are free to share the complete, unmodified file, to link to it, and to quote short passages in commentary, reporting, or discussion, provided the author and the site above are credited.

The following require prior written consent: republishing this work in whole or in substantial part under another name, masthead, or brand; excerpting or modifying it in a way that alters the analysis or removes the evidence classifications; incorporating any portion of it into a paid, commissioned, or client-facing deliverable; resale or distribution behind a paywall; and use of this document as training data for machine learning models.

Commissioned Market Assessment reports, commercial licensing, and briefings on this material are available.

This document is analysis, not investment, legal, or procurement advice. It was not commissioned, reviewed, or funded by any company named in it, and the author holds no position in any of them.

Market Assessment reports, commissioned research, and briefings: davidsoden.com/market-assessment