

UNVARNISHED REVIEWS

MongoDB Atlas vs. Azure Cosmos DB vs. SingleStore: The RU Forecasting Problem

Market Position: Big Picture

MongoDB Atlas, Azure Cosmos DB, and SingleStore are three of the most-compared platforms for organizations moving beyond a traditional relational database, whether toward flexible document schemas, globally distributed multi-model data, or unified transactional-and-analytical (HTAP) workloads. All three are credible, well-reviewed platforms serving distinctly different technical priorities: MongoDB Atlas leads on document-model maturity and the largest ecosystem of the three, Azure Cosmos DB leads on native multi-region distribution within the Azure ecosystem, and SingleStore leads on combining real-time analytics with transactional SQL in a single system.

The finding that should change how any of the three gets budgeted: Azure Cosmos DB's Request Unit (RU) billing model is repeatedly documented, across independent benchmarks rather than a single source, as producing meaningfully higher real-world costs than MongoDB Atlas's more predictable instance-based pricing at comparable workloads. This isn't a one-off finding, the multiplier varies by scenario and scale, but the direction is consistent across every independent comparison this report reviewed.

Platform Ratings at a Glance

Platform	G2 Rating	Reviews (G2)	Note
MongoDB Atlas	4.5 / 5	~370	Largest review base of the three; skews small-business (57.8%)
SingleStore	4.5 / 5	~118	Rated identically to MongoDB Atlas
Azure Cosmos DB	4.2 / 5	~68	Smallest review base; enterprise-skewed (42.4%)

MongoDB Atlas and SingleStore rate identically in user ratings, with Cosmos DB trailing both. Reviewer-cited pain points diverge in a telling way: MongoDB Atlas reviews mention "expensive" and "unclear pricing" alongside strong usability scores, while Cosmos DB reviews cite similar cost concerns without the offsetting usability advantage, consistent with the RU model's documented forecasting difficulty.

MongoDB Atlas: The Predictability Baseline

MongoDB Atlas prices primarily by instance-based dedicated clusters: M0 is free forever (512 MB storage, no backup, on AWS, Azure, or Google Cloud), scaling through M10 at approximately \$57/month up to M60 at approximately \$2,847/month. Enterprise Advanced, MongoDB's self-managed offering, is quote-based, with independent estimates placing it around \$10,000-\$30,000 per server annually. Add-ons include dedicated search nodes from \$0.12/hour, Stream Processing from \$0.06/hour, Data Federation and Online Archive at \$5/TB queried, backups at \$0.14/GB/month, and paid support starting at \$49/month. Autoscaling is included at no additional charge, a real, documented differentiator from Cosmos DB's autoscaling pricing below.

Vendr's transaction data confirms buyers committing to longer terms or prepaying annually typically achieve 15%-30% discounts compared to month-to-month billing, with the deepest discounts reserved for 2-3 year prepaid agreements on larger deployments.

Azure Cosmos DB: Native Multi-Region, With a Documented Cost Premium

Cosmos DB prices by Request Units (RUs), a normalized measure of the compute, memory, and I/O consumed by a database operation, provisioned at approximately \$0.008/hour per 100 RU/s, or consumption-based (Serverless) at roughly \$0.25 per million RUs. Storage runs approximately \$0.25/GB-month, with separate transactional and analytical storage options. A single 1KB document read costs roughly 1 RU; a 1KB write costs roughly 5 RUs; complex, unindexed queries can consume 100+ RUs, meaning real cost depends heavily on query patterns that are difficult to estimate before a workload is actually running.

Independent benchmarking is consistent on the resulting cost gap. One detailed 2023 benchmark comparing equivalent throughput found MongoDB Atlas at \$1,900/month against Cosmos DB above \$9,000/month for comparable performance. More recent 2026 scenario modeling shows a narrower but still real gap: A 100GB, 10,000 operations/second workload runs approximately \$570/month on MongoDB Atlas (M30) against \$1,000-\$1,500/month on Cosmos DB, roughly 2.2x at the midpoint. Cosmos DB's own autoscaling feature carries a documented 1.5x premium over its dedicated standard pricing, on top of the base RU cost, a real, additional line item MongoDB Atlas's included autoscaling doesn't carry.

SingleStore: HTAP Positioning, With a Real Multi-AZ Cost Doubling

SingleStore uses credit-based, usage-driven pricing across three editions: Shared (free, shared infrastructure, a way to test workloads before committing budget), Standard at \$0.99 per credit-hour, and Enterprise at \$1.49 per credit-hour, which adds disaster recovery, audit logging, and encryption key management. Workspace sizes range from S-00 (2 vCPU, 16 GB memory, 0.25 credits/hour) up to S-20 (160 vCPU, 1,280 GB memory, 20 credits/hour). Storage runs approximately \$0.023/GB-month at the lowest regional tier. Reserved pricing (prepaying for capacity) saves roughly 25% off on-demand rates.

The structural cost point worth understanding before comparing SingleStore's per-credit rate to MongoDB's or Cosmos DB's: Multi-AZ deployment, required for SingleStore's 99.99% SLA tier, effectively doubles compute cost, since it means running two availability zones' worth of compute rather than one. A single-AZ quote at the 99.9% SLA tier understates what production-grade redundancy costs by roughly 2x, a pattern this site has now documented across multiple distributed database platforms in this series. Minimum-viable-production topology is the detail a headline per-unit rate omits.

Pricing (July 2026)

Platform	Billing Unit	Entry Tier	Representative Mid-Tier Rate
MongoDB Atlas	Instance-based (dedicated cluster)	Free (M0, 512MB)	~\$570/month (M30)
Azure Cosmos DB	Request Units (RU) per 100 RU/s, or Serverless per million RUs	Free tier available	~\$0.008/hour per 100 RU/s (provisioned)
SingleStore	Compute credit-hour + storage	Free (Shared tier)	\$0.99/credit-hour (Standard)

TCO Comparison: 100GB Workload, 10,000 Operations/Second, 1 Month

This scenario uses independently modeled figures at matched scale for MongoDB Atlas and Cosmos DB; SingleStore is shown separately since its HTAP architecture serves a different workload profile than a pure document-store comparison, and its figure includes the real Multi-AZ cost for production redundancy.

Platform	Configuration	Monthly Cost
MongoDB Atlas	M30 dedicated cluster, single AZ	~\$570
Azure Cosmos DB	Provisioned, comparable throughput	~\$1,000-\$1,500
SingleStore	S-1 workspace (8 vCPU), reserved, single AZ	~\$2,168
SingleStore	S-1 workspace, reserved, Multi-AZ (99.99% SLA)	~\$4,336

Cosmos DB runs roughly 2.2x MongoDB Atlas's cost at this matched scale, consistent with the wider pattern documented across independent benchmarks. SingleStore isn't directly comparable to the other two at this specific scenario since it's priced and architected for HTAP workloads, not pure document storage, but its own Multi-AZ requirement doubles the single-AZ figure, the same structural pattern this database series has now found at CockroachDB, YugabyteDB, TiDB, and Google Cloud Spanner: Minimum viable production redundancy costs meaningfully more than the headline rate implies.

Costs the Pricing Table Misses

- **Cosmos DB's RU consumption is difficult to forecast before a workload runs.** Query complexity, indexing, and access patterns all affect RU cost per operation; budget a real testing phase against representative production queries before committing to a provisioned RU/s tier.
- **Cosmos DB's autoscaling carries a documented 1.5x premium over dedicated standard pricing.** MongoDB Atlas includes autoscaling at no additional charge, a real, quantified difference in what "included" actually means at each vendor.
- **SingleStore's Multi-AZ requirement for its top SLA tier effectively doubles compute cost.** A quote at the 99.9% single-AZ tier does not represent true production-grade availability; model the Multi-AZ rate explicitly if 99.99% uptime is a real requirement.
- **MongoDB Atlas Enterprise Advanced (self-managed) pricing is not published.** Independent estimates of \$10,000-\$30,000/server/year should be confirmed with MongoDB before budgeting a self-managed deployment.
- **None of the three include cross-region data transfer or egress in their base compute or RU rate.** Multi-region deployments, especially relevant for Cosmos DB's core positioning, should model network costs as a distinct line item.

Complexity Comparison

Pricing model differences trace back to real architectural differences in how each platform scales and how much of that scaling process a database team has to manage directly.

Dimension	MongoDB Atlas	Azure Cosmos DB	SingleStore
Implementation	Widely used managed service with mature tooling; independent reviews cite "ease of use" for core deployment, though initial configuration for sharding and production topology is documented as more involved than the headline simplicity suggests	Scaling is not instantaneous by default: Microsoft's own comparison documentation states Atlas-equivalent workloads scale automatically only after a day of workload analysis, while Cosmos DB's autoscaling responds immediately; this is a real architectural difference, not just a pricing one	Independent reviews document ongoing challenges with documentation depth and SQL feature completeness for teams new to the platform, consistent with its smaller market presence relative to the other two
Admin dependency	Manual sharding configuration is required for horizontal scaling, per Microsoft's own comparison documentation; this is a real, ongoing administrative responsibility MongoDB's ease-of-use reputation doesn't fully capture	Lower ongoing administrative burden for scaling specifically, since autoscaling is automatic and immediate, but administration of the RU-based cost model itself becomes its own ongoing discipline, tuning queries to control cost is a real, recurring task	Independent reviews specifically flag administration and monitoring as areas needing improvement, alongside migration tooling
Learning curve	Real for teams coming from relational databases; the document model and the absence of native joins is a documented adjustment period, though the concepts themselves are well-established and broadly taught	Steeper than MongoDB Atlas per multiple independent sources; understanding the Request Unit cost model, choosing a consistency level, and Azure-specific tooling all add real onboarding time beyond core database concepts	Documented as the steepest of the three for new users, per independent reviews citing both documentation gaps and the added complexity of its HTAP (hybrid transactional/analytical) architecture, which is a different mental model from either MongoDB's or Cosmos DB's
Ecosystem and community support	Largest community and ecosystem of the three, multi-cloud (AWS, Azure, GCP), the most third-party tooling and hiring pool	Strong within the Azure ecosystem specifically, but locked to Azure; documented as a real constraint for organizations not already committed to that cloud	Smallest community support of the three per independent comparison data, a real factor when troubleshooting or hiring for the platform

The practical takeaway: MongoDB Atlas's reputation for ease of use holds for getting started, but real administrative work (sharding, capacity planning) still exists once a deployment scales. Cosmos DB's automatic scaling reduces one class of operational burden while adding another, the ongoing discipline of managing RU consumption and cost. SingleStore's HTAP architecture is solving a real problem, unifying transactional and analytical workloads, but that architectural ambition comes with the steepest learning curve and thinnest ecosystem of the three, a cost worth weighing against how much a given workload actually needs that unification.

The Decision Framework

Choose MongoDB Atlas if: Document-model flexibility and predictable, instance-based cost forecasting matter more than native multi-region distribution, and you want autoscaling included rather than priced as a premium.

Choose Azure Cosmos DB if: You're Azure-committed and need native, low-latency multi-region distribution as a core requirement, and can invest in workload-specific RU forecasting before committing to a provisioned tier, since the documented cost premium over MongoDB is real but the underlying capability may justify it for truly global applications.

Choose SingleStore if: Your workload needs unified transactional and real-time analytical queries on the same data, avoiding a separate ETL pipeline to a data warehouse, and you can budget for the real Multi-AZ cost if 99.99% availability is a production requirement, not just a nice-to-have.

Everyone: Model your real workload against each platform's specific billing unit, instances for MongoDB, RUs for Cosmos DB, compute credits for SingleStore, before comparing headline rates. The most consistent, best-documented risk across all three is that a quote based on assumed or best-case usage patterns understates real cost once a production workload is

running.

The Bottom Line

MongoDB Atlas, Azure Cosmos DB, and SingleStore are all credible, well-reviewed platforms, each built for a distinctly different priority: document-model maturity and pricing predictability at MongoDB, native multi-region distribution at Cosmos DB, and unified transactional-analytical workloads at SingleStore. The real finding this comparison surfaces is that Cosmos DB's RU-based billing model carries a consistent, independently documented cost premium over MongoDB Atlas at comparable scale, and that SingleStore's real production cost, once Multi-AZ redundancy is included, is meaningfully higher than its single-AZ headline rate. Model the unit that actually drives your bill, not the one that's easiest to quote, before comparing any two of these three.