Sentrilite Cloud Cost Assessment Report

Generated: 2026-06-27 17:05:12 | Node: ip-172-31-25-143 | Region: us-east-1
Period: 2026-06-20 to 2026-06-27

Projected Monthly Cloud Bill Increase: $1,894 Go to Details

** Based on what changed from last period

About this report

This assessment turns "why did our cloud bill change?" into a clear, workload-level answer. Sentrilite collects deep system traces — network egress and compute usage — directly from your infrastructure and attributes spend to the exact namespace, pod, container, process, and destination responsible.

Sentrilite runs two independent data pipelines on every monitored node — one for network egress cost attribution, one for compute cost attribution. Both read directly from the Linux kernel with no application instrumentation required.

Contents

1. Cloud Cost Intelligence Report

Dashboard data — egress sorted by cost; compute by utilisation & right-sizing.

TOTAL BANDWIDTH / EGRESS COST

TOTAL BANDWIDTH	EST. TOTAL COST	CROSS-REGION	FLAGS
17710.08 GB	$1280.86	$7.32	cross-region, high-volume

Namespace	Pod	Container	Service	Provider	Region	Category	Bandwidth	Est. Cost	Flag
data-team	rag-pipeline-pro	main	python	UNKNO	—	Internet	5147.98	$463.3180	high-volume
marketing	content-generat	main	node	UNKNO	—	Internet	1652.73	$148.7461	high-volume
legal	contract-analyz	main	python	UNKNO	—	Internet	1347.66	$121.2894	high-volume
ml-platform	inference-api-pr	main	python	UNKNO	—	Internet	1213.47	$109.2122	high-volume
media	video-transcode	main	python	UNKNO	—	Internet	918.60 GB	$82.6739	high-volume
data-team	db-replication-9	main	postgres	GCP	us-central1	Internet	909.50 GB	$81.8551	cross-region
engineering	code-review-bot	main	python	UNKNO	—	Internet	907.82 GB	$81.7037	high-volume
ml-platform	llm-eval-runner-	main	python	UNKNO	—	Internet	771.89 GB	$69.4702	high-volume
analytics	etl-batch-8e2c5f	main	python	GCP	europe-west	Internet	613.30 GB	$55.1971	cross-region
support	ticket-response-	main	python	UNKNO	—	Internet	606.88 GB	$54.6196	high-volume
data-team	db-replication-9	main	postgres	AWS	us-west-2	Cross-region	365.80 GB	$7.3160	cross-region
analytics	warehouse-sync	main	python	AZURE	eastus2	Internet	60.68 GB	$5.4615	cross-region
data-team	db-replication-9	main	postgres	AWS	us-east-1	Intra-region	226.76 GB	$0.0000	high-volume
analytics	warehouse-sync	main	python	AWS	us-east-1	S3 same-rgn	462.21 GB	$0.0000	high-volume
analytics	etl-batch-8e2c5f	main	python	AWS	us-east-1	S3 same-rgn	305.29 GB	$0.0000	high-volume
data-team	rag-pipeline-pro	main	python	AWS	us-east-1	S3 same-rgn	302.18 GB	$0.0000	high-volume
data-team	rag-pipeline-pro	main	python	AWS	us-east-1	VPC Endpoint	150.76 GB	$0.0000	high-volume
frontend	web-app-prod-3	main	node	AWS	us-east-1	Internet	907.24 GB	$0.0000	high-volume
frontend	web-app-prod-3	main	node	AWS	us-east-1	NAT Gateway	459.95 GB	$0.0000	high-volume
ml-platform	inference-api-pr	main	python	AWS	us-east-1	S3 same-rgn	226.67 GB	$0.0000	high-volume
media	video-transcode	main	python	AWS	us-east-1	S3 same-rgn	152.69 GB	$0.0000	high-volume

COMPUTE — UTILISATION & RIGHT-SIZING

CPU WASTED	MEM WASTED	IDLE PODS	RIGHT-SIZE (<50% CPU)
71%	44%	3	12 / 15

Namespace	Pod	Container	Service	CPU Used	Mem Used	CPU% / Mem%	Status
ml-platform	model-training-staging-	main	python	0.004	328.9 MB	0.2% / 4.0%	Idle
batch	nightly-export-legacy-4	main	python	0.003	106.5 MB	0.3% / 5.2%	Idle
dev	sandbox-experiment-9f	main	python	0.006	46.7 MB	0.6% / 3.0%	Idle
marketing	content-generator-prod	main	node	0.079	844.9 MB	4.0% / 34.4%	Right-size
legal	contract-analyzer-3b9a	main	python	0.088	1.26 GB	4.4% / 45.1%	Right-size
analytics	warehouse-sync-a3f9c	main	python	0.104	1.41 GB	5.2% / 46.9%	Right-size
engineering	code-review-bot-7f8d9	main	python	0.126	1.52 GB	6.3% / 42.2%	Right-size
ml-platform	llm-eval-runner-3c9d1	main	python	0.143	1.14 GB	7.2% / 38.0%	Right-size

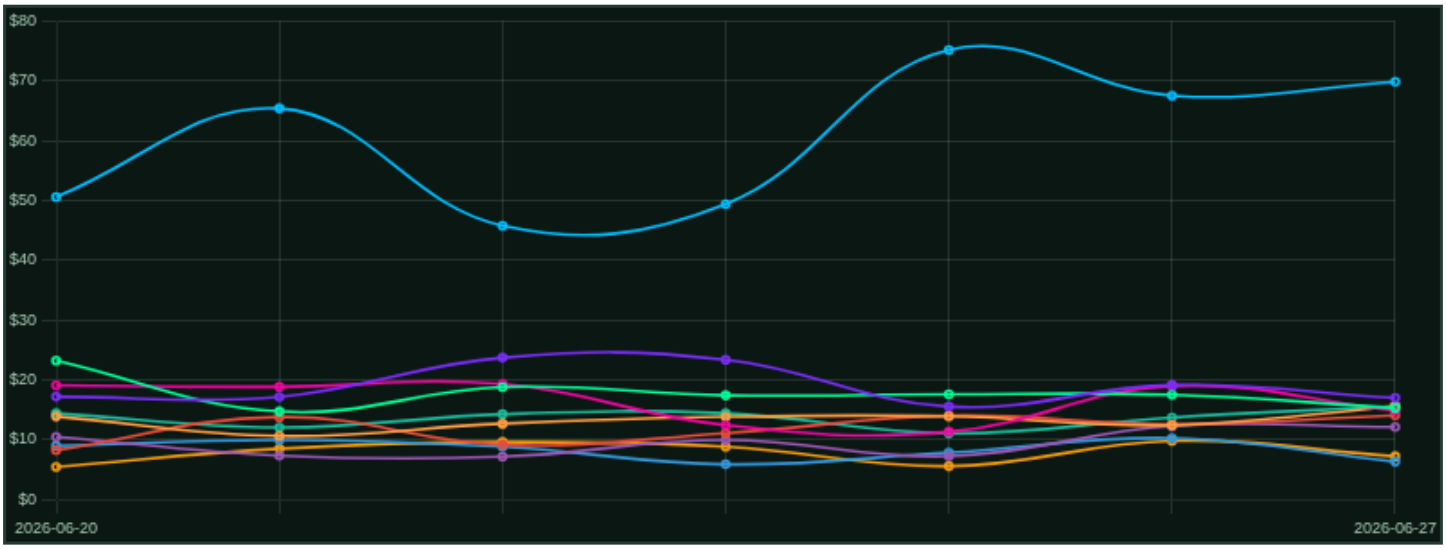
Namespace	Pod	Container	Service	CPU Used	Mem Used	CPU% / Mem%	Status
data-team	db-replication-9c4f8e-r	main	postgres	0.146	1.66 GB	7.3% / 33.2%	**Right-size**
support	ticket-response-drafter	main	python	0.165	939.2 MB	8.2% / 45.9%	**Right-size**
ml-platform	inference-api-prod-2e8	main	python	0.190	1.60 GB	9.5% / 39.9%	**Right-size**
media	video-transcoder-prod-	main	python	0.238	1.69 GB	11.9% / 42.2%	**Right-size**

2. Cloud Cost Visuals

Trend and breakdown charts — same colours as the dashboard Cost Visualizer.

EGRESS COST ANALYSIS

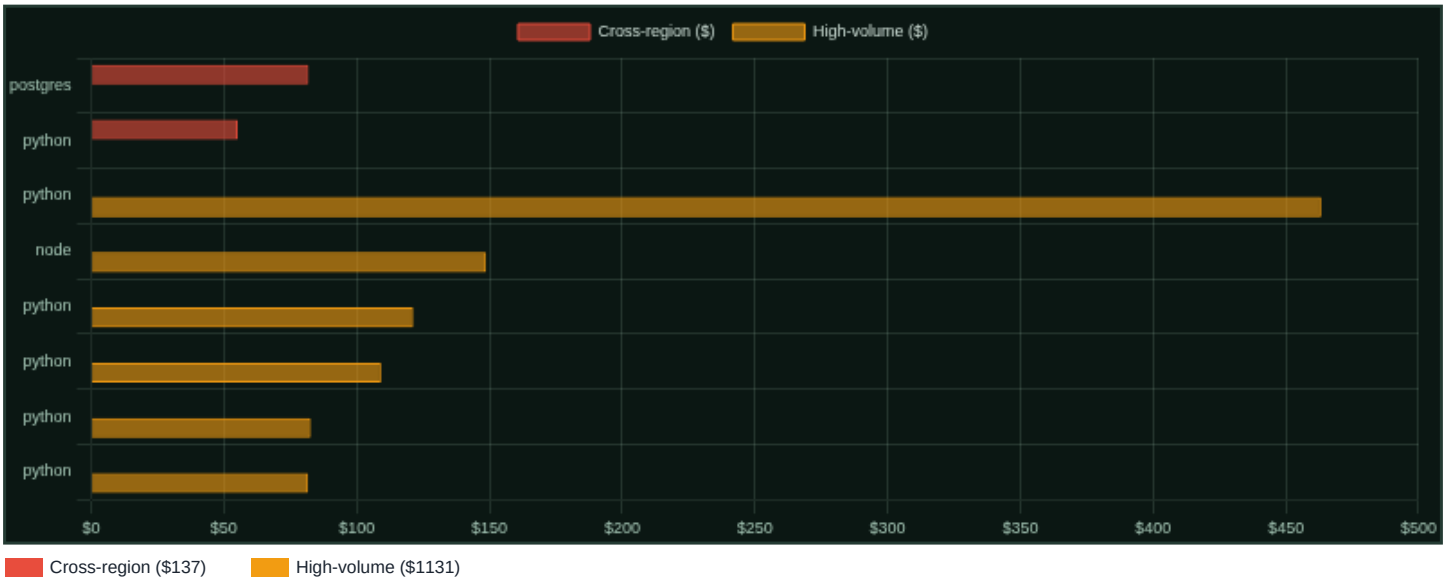
Egress Cost Over Time — Top 10 Services ($1,268)



Daily egress cost trend per service — identifies which workloads are driving spend.

EGRESS COST ANALYSIS

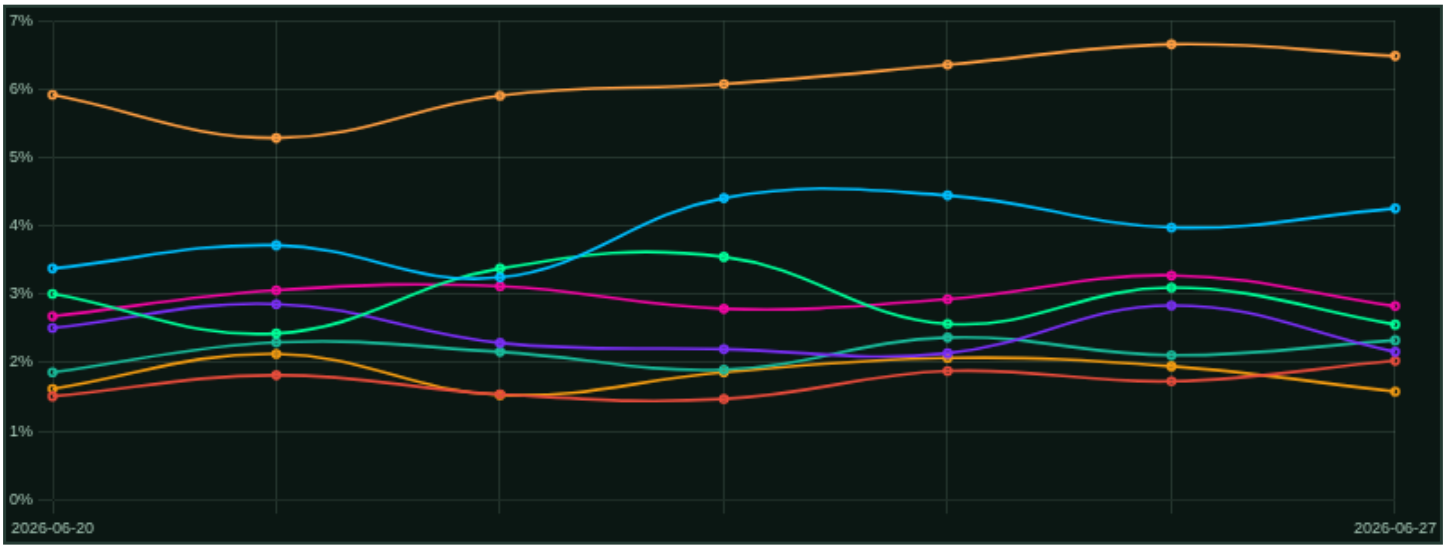
Flagged Services — Cross-Region & High-Volume ($1,268)



Red = cross-region transfers. Amber = high-volume flows beyond normal thresholds.

COMPUTE UTILISATION ANALYSIS

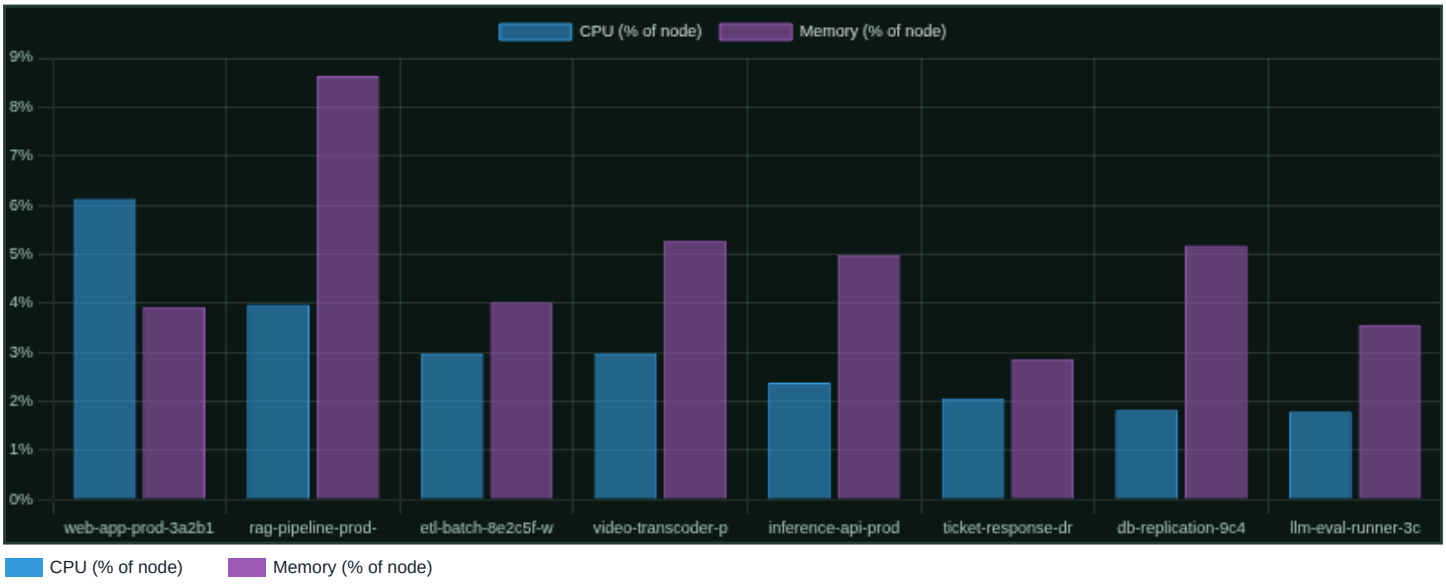
CPU Consumption Over Time — Top Workloads (% of node)



Each workload's share of node CPU over the period — taller lines consume more of the instance.

COMPUTE UTILISATION ANALYSIS

CPU & Memory Consumption — % of Node per Workload



Blue = CPU share of the instance. Purple = memory share. Higher bars are the heavier workloads to run — candidates to reschedule or right-size.

3. Cloud Cost Insights

Observations from Sentrilite — the same Cost Insights shown in the dashboard.

Period: 2026-06-20 to 2026-06-27 · Generated: 2026-06-27 18:56:06

CONCENTRATION

data-team/rag-pipeline-prod-7c4f8e-x9m2k/python drives 36% of egress cost ($463)

6318.10 GB sent+received this period. Concentrated spend is the fastest place to act.

Namespace data-team is 43% of egress cost ($552)

Worth confirming this is expected for that team or service.

data-team/rag-pipeline-prod-7c4f8e-x9m2k/python share of cluster egress grew 14pp (18% to 32%)

Cost grew $141 to $463 (+$323) · 6318.10 GB sent+received this period

Namespace data-team grew 13pp of cluster egress (28% to 40%)

Cost grew $203 to $552 (+$349)

CROSS-REGION

Cross-region / cross-cloud egress costs $150 this period

Cross-boundary traffic bills at a premium rate — verify these flows are intentional.

1 existing workload(s) started sending cross-region traffic this period

analytics/etl-batch-8e2c5f-w9n7v/python (692.01 GB cross-region, $55)

analytics/warehouse-sync-a3f9c2-h7n4p/python cross-region traffic dropped — saved $79

1054.87 GB (prior) to 68.24 GB (current) · likely remediation: read replica, caching, or same-region alternative

FOOTPRINT

Direct internet egress totals $1274 — the highest-rate category

Routing repeat traffic through a CDN, cache, or private link is usually cheaper.

12 high-volume workload(s) moved 50 GB or more this period

Largest: data-team/rag-pipeline-prod-7c4f8e-x9m2k/python (6318.10 GB). Unusually large flows can signal a cost spike or data moving where it need not.

Egress cost spans multiple providers

UNKNOWN 88% · GCP 11% · AWS 1% · AZURE 0%. Multi-provider egress often hides cross-cloud transfers billed at the highest tier.

Traffic touches 5 distinct regions

eastus2, europe-west1, us-central1, us-east-1, …. The more regions in play, the more cross-region transfer accumulates.

COMPUTE

About 71% of CPU and 44% of memory capacity is unused

3 idle pods this period — capacity you pay for but do not use.

12 workload(s) running under 50% CPU — right-sizing candidates

Lowering CPU/memory limits to match real usage recovers spend without changing throughput.

3 completely idle pods

Strong candidates to scale to zero, schedule off-hours, or decommission.

frontend/web-app-prod-3a2b1c-h9k4m/node CPU usage grew 2.2× period-over-period (0.23 to 0.49 cores avg)

Peak 0.74 cores · Sustained high CPU for 2 hours during this period. · Consider whether the limit needs to be raised or horizontal scaling (HPA) configured

marketing/content-generator-prod-5c8b6a-q9p3n/node CPU usage dropped 76% (0.33 to 0.08 cores avg)

Peak this period: 0.12 cores · Candidate for downsizing — current load is well below allocated capacity

ADOPTION

3 new workloads appeared this period, adding $261 in egress cost

ml-platform/inference-api-prod-2e8a4c-k7r3w/python (1626.77 GB, $109) · media/video-transcoder-prod-8a1f2c-q4r7t/python (1211.55 GB, $83) · ml-platform/llm-eval-runner-3c9d1b-w8k2p/python (871.50 GB, $69)

TRENDS

analytics/etl-batch-8e2c5f-w9n7v/python egress cost grew $55 (2.9× period-over-period)

360.83 GB to 1037.21 GB · prior cost <$0.01 to current cost $55

support/ticket-response-drafter-2b5c4d-z1m6t/python egress cost dropped $114 (3.1× less than prior period)

2111.57 GB to 680.41 GB · prior cost $169 to current cost $55

4. Summary of Findings & Recommendations

The headline numbers, and the actions most likely to reduce spend.

Projected Monthly Increase — Breakdown

Each workload's egress cost this period vs the period before, scaled to a month (×4). Increases add to the bill; savings (green) subtract. The lines sum to the figure on the cover.

Workload	Prior	Current	Change	Per month
data-team/rag-pipeline-prod-7c4f8e-x9m2k/python	$141	$463	+$323	**+$1,290**
ml-platform/inference-api-prod-2e8a4c-k7r3w/python	$0	$109	+$109	**+$437**
media/video-transcoder-prod-8a1f2c-q4r7t/python	$0	$83	+$83	**+$331**
ml-platform/llm-eval-runner-3c9d1b-w8k2p/python	$0	$69	+$69	**+$278**
analytics/etl-batch-8e2c5f-w9n7v/python	$0	$55	+$55	**+$221**
data-team/db-replication-9c4f8e-r5m2k/postgres	$62	$89	+$27	**+$108**
marketing/content-generator-prod-5c8b6a-q9p3n/node	$140	$149	+$9	**+$35**
engineering/code-review-bot-7f8d9c-xk4m2/python	$85	$82	-$3	**-$11**
legal/contract-analyzer-3b9a7d-l4k8h/python	$127	$121	-$6	**-$23**
analytics/warehouse-sync-a3f9c2-h7n4p/python	$84	$5	-$79	**-$315**
support/ticket-response-drafter-2b5c4d-z1m6t/python	$169	$55	-$114	**-$456**
Net projected monthly increase			**+$473**	**+$1,894**

Lines rounded to the nearest dollar; the net is exact and matches the cover.

Findings

- Total estimated egress cost for the period: $1280.86 across 17710.08 GB of bandwidth.
- Cross-region egress: $7.32 — billed at premium rates.
- Idle / wasted compute: about 71% of CPU and 44% of memory capacity went unused over the period (3 idle pods).
- Right-sizing: 12 of 15 workloads ran below 50% CPU utilisation and can be right-sized.
- Anomaly flags raised: cross-region, high-volume.

Recommended actions

1. Review cross-region transfers and co-locate the services involved, or route through cheaper paths, to cut premium egress.
2. Investigate high-volume flows — confirm they are necessary and not duplicating data movement.
3. Start with the top egress service ("python") — concentrated spend gives the biggest return per change.
4. Scale idle pods to zero on a schedule, or consolidate them, to stop paying for unused capacity.
5. Right-size the 12 over-provisioned workloads — lower CPU/memory limits to match real usage.
6. Re-run this assessment over a wider window to confirm trends and measure the impact of any changes.

All dollar amounts are list-price estimates intended to highlight optimization opportunities; actual cost varies with Reserved capacity, Savings Plans, and committed-use discounts.

5. Glossary: Calculation Details

How every number in this report is measured and priced.

This report combines two data streams Sentrilite collects continuously: network egress and compute usage.

EGRESS / BANDWIDTH COST

Data Collection	Network traffic is observed at the kernel level on every active network interface. Each outbound connection is tracked — remote IP, bytes transferred, and the originating process.
Sampling Window	Traffic is aggregated continuously and flushed every 60 seconds. The report sums all records in the selected date range.
Linux bare-metal	Traffic is attributed per process name as shown in top or ps. Each row represents one process communicating with one remote destination.
Kubernetes	Traffic is enriched with pod name, namespace, and container name from the Kubernetes API.
Provider & Region	Remote IPs are matched against cloud provider IP ranges to identify cross-region and cross-provider flows, which attract higher egress fees.
Cost Category	Each flow is classified into a pricing category — Intra-region, Cross-region, S3 (same/cross region), Internet, NAT Gateway, VPC Endpoint. The category determines the applied rate.
Cost Formula	Est. Cost = Bandwidth (GB) x Rate ($/GB). Rates loaded from cost_matrix.json. All dollar amounts are AWS list-price estimates — actual cost varies with Reserved capacity, Savings Plans, and EDP terms.
Flags	cross-region: traffic leaving the local cloud region. high-volume: unusually large data transfer — a potential cost spike or exfiltration signal.

COMPUTE — CPU & MEMORY UTILISATION

Data Collection	CPU and memory are read from the Linux kernel process filesystem (/proc) — the same source used by top and ps. No additional agents required.
CPU Sampling	Two CPU readings are taken 5 seconds apart per 60-second cycle. The tick difference over elapsed time gives a short-window utilisation, averaged across all cycles in the period.
Memory Sampling	Resident memory (RAM in use) is read once per 60-second cycle. The value shown is the average across all samples in the period.
Kubernetes	CPU% and Mem% are calculated against pod resource limits from the Kubernetes API — useful for spotting workloads whose limits are set far above actual usage.
% of Node	Each workload's share of the instance it runs on = resources used ÷ instance capacity (vCPUs for CPU, GB for memory), averaged across the period. This is what the compute charts show — which workloads are heaviest to run, and which could move to a smaller or larger instance.
Why no dollars	Compute analysis reports utilisation, not cost, on purpose: you already know the instance price. What's actionable is which workloads consume the node and whether the instance is the right size for them.
Idle / Wasted	Share of the node's CPU (or memory) capacity not consumed by any workload over the period. High waste means the instance is larger than the workloads need — reported as a percentage, not a dollar figure.
CPU% / Mem% Guide	Orange (0%): completely idle — paying for unused capacity. Yellow (<20%): low utilisation. Green (>=20%): reasonable utilisation.

COST INSIGHTS — CATEGORIES

Concentration	How much of egress spend is driven by a single workload or namespace. A few workloads usually dominate the bill — the fastest place to act, and the first thing to watch when one's share climbs.
Cross-Region	Traffic crossing cloud regions or providers, which bills at premium rates. Flags workloads that newly start sending cross-region, and confirms when that traffic has been reduced or removed.
Trends	Period-over-period cost movement — workloads whose egress spiked or dropped versus the prior period. The increases here are what drive the Projected Monthly Cloud Bill Increase on the cover.
Adoption	New workloads that appeared this period and the egress cost they add. Catches new spend the week it starts, instead of at month-end on the bill.
Footprint	The shape of egress overall — direct internet egress (highest rate), high-volume flows, the mix of providers, and how many regions traffic touches.

Compute	Node CPU/memory utilisation — idle and under-used workloads, and right-sizing candidates. Reported as % of node capacity, not dollars, since the instance price is already known.

PRICING DATA SOURCES

cost_matrix.json	Egress pricing: per-provider, per-region outbound rates in USD/GB by category. Example: AWS us-east-1 internet $0.09/GB, cross-region $0.02/GB, same-region S3 $0.00/GB.
List-Price Estimates	Egress dollar amounts are list-price estimates to help identify optimization opportunities. Actual cost varies with Reserved Instances, Savings Plans, committed-use discounts, and Enterprise Discount Programs. Compute is reported as % utilisation, not dollars.
Report Period	Egress costs are summed across the selected date range. CPU/memory values shown are the mean across all 60-second samples in the period.