



Vantage RuntimeAI

Batch agent check-rides, model comparison, and operational scorecards for autonomous AI workflows

8.0/10

Overall score

Run summary

Scenario: Null Rate Alert

Model: anthropic/claude-haiku-4.5

Roles: Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 15s

Actual run cost: \$0.006500

Projected execution cost: \$10.73/mo (75/biz day × 22 days)

Report date: July 12, 2026 - 11:47 PM EST

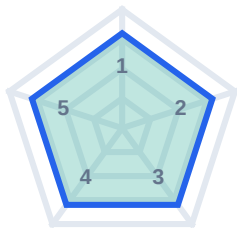
Overall

8.0/10

PASS - trust=high

Pass - score 8.0/10 clears 7.0 with acceptable trust/closure.

CAPABILITY RUBRIC



#	Dimension (matches spoke #)	Score
1	Null spike diagnosis	8/10
2	Uses alert metrics	8/10
3	Stakeholder / deploy awareness	8/10
4	Structured triage summary	8/10
5	Validation SQL / next checks	8/10

Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

Technical criteria checklist

Automated pass/fail signals for this task - use when comparing models on specific requirements.

- Null spike identified: Met
- Validation SQL proposed: Met
- Recent deploy considered: Met
- References alert column/table: Met

TECHNICAL JUDGEMENT

Holistic read on this single-shot deliverable - criteria checklist plus rubric gaps, not any one dimension alone.

8/10

Telemetry

Run metrics

Mode: multi-turn - Agent msgs: 2 - Counterpart msgs: 3 - Output: 4996 chars - Latency: 15s - Criteria: 4/4

Score analysis

Identifies null-rate spike on shipping_country_code.

Score snapshot

Solid: Solid technical task response with a few gaps.

Interpretation

Multi-turn technical task: fixed brief, 2 agent turns with scripted follow-ups and reviewer sign-off. 4 of 4 checklist criteria passed.

BRIGHT SPOTS (RUBRIC)

- + Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- + Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- + Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- + Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- + Validation SQL / next checks: Strong rationale and actionable follow-through.

DIMENSION NOTES

- Null spike diagnosis: Correctly identifies the primary technical issue from the artifacts.
- Uses alert metrics: Uses tables, logs, or schema facts from the task brief.
- Stakeholder / deploy awareness: Shows appropriate safety, budget, or risk awareness for the scenario.
- Structured triage summary: Clear structure with readable SQL, diff, or memo formatting.
- Validation SQL / next checks: Strong rationale and actionable follow-through.
- Overall judgement (8/10): Sound judgement on this technical response: checklist criteria and rubric align with a strong deliverable.

Signals

Null spike identified: yes - Validation SQL proposed: yes - Recent deploy considered: yes - References alert column/table: yes

CAVEATS

- Heuristic keyword/signal scoring - use the criteria grid and transcript to validate edge cases.

Transcript

Session

Scenario: Null Rate Alert

Agent response <-> Engineering counterpart

Data Engineer (AI agent) <> Engineering counterpart (AI agent)

Duration: 15s

Engineering counterpart

3:46 AM

Task brief - Null rate alert (simple complexity)

A column null rate spiked overnight. Triage the alert JSON and recommend next steps:

```
```json
{
 "table": "analytics.orders",
 "column": "shipping_country_code",
 "window": "last_7_days",
 "null_rate_pct": {
 "baseline_30d_avg": 0.3,
 "yesterday": 17.8,
 "today_so_far": 19.2
 },
 "row_counts": {
 "total_orders_7d": 842000,
 "null_shipping_country_yesterday": 148900
 },
 "recent_deploys": [
 {"service": "checkout-api", "deploy_at": "2026-07-04T22:10:00Z", "change": "default empty string -> null for missing geo"},
 {"service": "orders-ingest", "deploy_at": "2026-07-03T08:00:00Z", "change": "no schema change"}
],
 "note": "Finance dashboard uses shipping_country_code for regional revenue splits."
}
```
```

Agent response

3:47 AM - Responded in 6s

Null Rate Spike Triage: `analytics.orders.shipping\_country\_code`

(1) Likely Cause

checkout-api deploy (2026-07-04 22:10 UTC) changed null handling: switched from coercing empty strings -> empty string to explicitly setting `null` for missing geolocation data. This is a intentional schema behavior change, not data corruption.

(2) Validation SQL

```sql

-- Confirm spike timing aligns with deploy

SELECT

DATE(created\_at) as date,

COUNT() as total\_orders,

SUM(CASE WHEN shipping\_country\_code IS NULL THEN 1 ELSE 0 END) as null\_count,

ROUND(100.0 \* SUM(CASE WHEN shipping\_country\_code IS NULL THEN 1 ELSE 0 END) / COUNT(), 2) as null\_pct  
FROM analytics.orders

WHERE created\_at >= '2026-07-02'

GROUP BY DATE(created\_at)

ORDER BY date DESC;

-- Check if empty strings still exist (pre-deploy behavior)

SELECT COUNT() as empty\_string\_count

FROM analytics.orders

WHERE shipping\_country\_code = ""

AND created\_at >= '2026-07-04 22:00:00';

```

(3) Recent Deploy Assessment

? CHECKOUT-API DEPLOY IS HIGHLY SUSPECT (2026-07-04 22:10 UTC)

- Timing matches spike onset perfectly

- Explicit change log: "default empty string -> null for missing geo"

- orders-ingest deploy (07-03) shows no schema change -> ruled out

(4) Stakeholder Communication (1-line)

"Shipping country nulls spiked 17.8% after yesterday's checkout-api deploy that now explicitly nulls missing geolocation instead of defaulting to empty strings-Finance Dashboard revenue splits will need to handle nulls or we revert the deploy."

Recommended Next Steps

1. Confirm validation SQL output matches 2026-07-04 22:10 UTC boundary

2. Decide: Is explicit null the desired behavior? If yes, update Finance Dashboard queries to `COALESCE(shipping\_country\_code, 'UNKNOWN')`. If no, rollback checkout-api.

3. Alert Finance immediately (they own regional revenue splits)-this impacts reporting

Engineering counterpart
3:47 AM - Responded in 0s

Thanks - walk me through your reasoning step by step. What assumptions did you make?

Agent response

3:47 AM - Responded in 9s

Step-by-Step Reasoning & Assumptions

Step 1: Quantify the Severity

Observation: Null rate jumped from 0.3% baseline -> 17.8% (59x increase)

Assumption: This is abnormal and not gradual drift. A 59x spike overnight screams event-driven change, not gradual data quality degradation.

Why this matters: Rules out causes like "users stopped entering country" or "upstream API flakiness"-those would show gradual increase, not cliff.

Step 2: Timeline Correlation

Observation: Two deploys in the window, but different timing:

- checkout-api: 2026-07-04 22:10 (yesterday evening)

- orders-ingest: 2026-07-03 08:00 (two days ago)

Assumption: The spike occurs between 07-04 22:10 and today. The alert says "yesterday" had 17.8% nulls, which means the spike was already present when we measured yesterday.

Why this matters: If the spike happened before 07-03 08:00, orders-ingest would be suspect. But the timing strongly suggests 07-04 22:10 is the trigger.

Step 3: Match Change Log to Behavior

Observation: checkout-api changelog explicitly says: "default empty string -> null for missing geo"

Assumption: This change directly causes the observed behavior-the system is now intentionally writing `NULL` instead of `` when geolocation is missing.

Key assumption I made: The deploy was successful and the code change executed as intended. I'm not assuming a bug-I'm assuming the change worked as designed, which is why we see the spike.

Why this matters: If I assumed the deploy failed, I'd look for a rollback or error logs. But the clean 59x spike + matching changelog suggests this is working as coded.

Step 4: Rule Out Data Corruption

Observation: Row counts are normal (842k orders in 7d, 148.9k nulls yesterday). No mention of dropped rows or duplicate keys.

Assumption: This is a schema behavior change, not data loss or ETL breakage. The nulls are "real" in the sense that they represent missing data being handled differently.

Why this matters: If I suspected corruption, I'd recommend data recovery and rollback urgently. Instead, this is a choice to represent missing data differently-fixable at the application layer.

Step 5: Impact Surface

Observation: Alert note says "Finance dashboard uses shipping_country_code for regional revenue splits."

Assumption: Dashboards likely use `WHERE shipping\_country\_code IS NOT NULL` or similar filters, which will now exclude ~18% of orders. This breaks revenue attribution by region.

Why this matters: This isn't a silent bug-it's a loud downstream impact. Finance will notice revenue totals don't add up. This escalates priority.

Step 6: Validation Query Design

Assumption 1: I can query `analytics.orders` directly and it has a `created\_at` timestamp.

Assumption 2: The deploy timestamp (22:10 UTC) is in the same timezone as `created\_at`.

Assumption 3: Looking for empty strings (`""`) pre-deploy confirms the old behavior existed.

Why these matter: If

Engineering counterpart**3:47 AM - Responded in 0s**

Thanks - assumptions and validation path are reasonable. Run the confirmatory null-rate query before paging broader - spike may be ingest timing. Review complete - no further response needed.