



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: amazon/nova-micro-v1

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

Duration: 6s

Actual run cost: \$0.000203

Projected execution cost: \$0.3350/mo (75/biz day × 22 days)

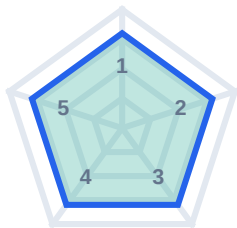
Report date: July 14, 2026 - 3:06 AM EST

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: 5101 chars - Latency: 6s - 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: 6s

Engineering counterpart

7:06 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

7:06 AM - Responded in 2s

Triage Summary for Null Rate Spike in `analytics.orders.shipping\_country\_code`

1. Likely Cause of the Spike:

- The most likely cause for the spike in null rate for `shipping\_country\_code` is the recent deploy from `checkout-api` on `2026-07-04`. This deploy changed the behavior from storing an empty string for missing geo to using `null`, which could significantly increase the null rate if there are many records without a shipping country.

2. Validation SQL to Confirm:

To confirm the spike, we can run the following SQL to compare the count of null entries before and after the deploy:

```
```sql
```

```
WITH pre_deploy AS (
 SELECT COUNT() AS total_orders, COUNT() FILTER (WHERE shipping_country_code IS NULL) AS null_count
 FROM analytics.orders
 WHERE DATE(order_date) BETWEEN '2026-07-02' AND '2026-07-03'
```

```
),
```

```
post_deploy AS (
 SELECT COUNT() AS total_orders, COUNT() FILTER (WHERE shipping_country_code IS NULL) AS null_count
 FROM analytics.orders
 WHERE DATE(order_date) BETWEEN '2026-07-04' AND '2026-07-05'
```

```
)
```

```
SELECT
 p.total_orders AS total_orders_pre,
 p.null_count AS null_count_pre,
 d.total_orders AS total_orders_post,
 d.null_count AS null_count_post
FROM pre_deploy p, post_deploy d;
```

```
```
```

3. Whether a Recent Deploy is Suspect:

- Yes, the recent deploy on `2026-07-04` to `checkout-api` where the behavior changed from storing an empty string to `null` for missing geo is highly suspect. Given the spike in null rates, it's likely that this deploy is directly responsible.

4. One-Line Stakeholder Comms:

"Confirmed spike in null rate for orders table's shipping_country_code likely due to recent checkout-api deploy changing missing geo handling; investigate further."

If you need any more specifics or further investigation steps, please let me know!

Engineering counterpart

7:06 AM - Responded in 0s

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

Agent response

7:06 AM - Responded in 4s

Sure, let's break down the steps and assumptions to understand the likely cause of the spike and validate it thoroughly.

Step-by-Step Reasoning:

1. Initial Analysis:

- From the alert JSON, the null rate of the `shipping\_country\_code` column spiked from an average of 0.3% to 17.8% yesterday and further increased to 19.2% today.
- The significant increase suggests a potential systemic issue or a changed data behavior.

2. Key Information:

- Recent Deploys:
 - `checkout-api`: Changed default to `null` for missing geo on `2026-07-04`.
 - `orders-ingest`: No schema changes on `2026-07-03`.

3. Assumptions:

- The `orders` table is sufficiently populated such that small changes in handling of missing data could significantly alter null rates.
- The `checkout-api` deploy would affect the data quality for shipments as it's responsible for handling geo information.

4. Examine the Recent Deploy:

- The `checkout-api` deploy changed how missing geographic information is recorded. Previously, missing information might have been represented as an empty string (`""`), which is not `NULL`. Now, missing geo data is recorded as `NULL`.

5. Impact Assessment:

- If there were many orders missing shipping country information, this change results in many more entries being recorded as `NULL`, thereby spiking the null rate.

Validation:

To confirm the hypothesis:

1. SQL Query:

- Write a SQL query to compare data before and after the `checkout-api` deployment to see if there's a significant jump in the number of `NULL` values.

```
```sql
WITH pre_deploy AS (
 SELECT COUNT() AS total_orders, COUNT() FILTER (WHERE shipping_country_code IS NULL) AS null_count
 FROM analytics.orders
 WHERE DATE(order_date) BETWEEN '2026-07-02' AND '2026-07-03' -- One day before the deploy
),
post_deploy AS (
 SELECT COUNT() AS total_orders, COUNT() FILTER (WHERE shipping_country_code IS NULL) AS null_count
 FROM analytics.orders
 WHERE DATE(order_date) BETWEEN '2026-07-04' AND '2026-07-05' -- One day after the deploy
)
SELECT
 p.total_orders AS total_orders_pre,
 p.null_count AS null_count_pre,
 d.total_orders AS total_orders_post,
 d.null_count AS null_count_post
FROM pre_deploy p, post_deploy d;
```
```

Observations After Running the SQL Query:

- If there's a significant increase in `null\_count` after the deploy compared to before, it confirms that the spike is likely due to the change in `checkout-api`.

Conclusion:

- Based on the assumption that the new behavior of recording missing geo data as `NULL` would significantly impact the `NULL` rate and the confirmation through SQL, we can conclude that the recent deployment to `checkout-api` is most likely responsible for the spike.

- One-Line Stakeholder Comms: "Confirmed spike in null rate for orders table's shipping_country_code likely due to recent checkout-api deploy changing missing geo handling; investigate further."

This thorough reasoning aligns with both the data provided and logical conclusions based on observed changes and typical deployment impacts.

Engineering counterpart**7:06 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.