



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: deepseek/deepseek-v4-flash

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

Duration: 22s

Actual run cost: \$0.000616

Projected execution cost: \$1.02/mo (75/biz day × 22 days)

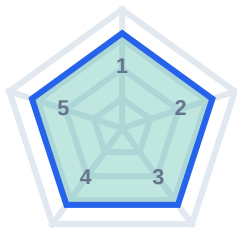
Report date: July 14, 2026 - 3:07 AM EST

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: 3568 chars - Latency: 22s - 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: 22s

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 12s

1. Likely Cause

The `checkout-api` deploy at 2026-07-04T22:10:00Z changed default behavior from empty string to `null` for missing geo data. This coincides with the null rate spike from 0.3% -> 17.8%. The `orders-ingest` deploy is unlikely suspect (no schema change).

2. Validation SQL

```
```sql
```

```
SELECT
```

```
 DATE(created_at) as order_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(), 1) as
null_rate_pct
```

```
FROM analytics.orders
```

```
WHERE created_at >= '2026-07-04' -- post-deploy
```

```
 AND created_at < CURRENT_DATE
```

```
GROUP BY 1
```

```
ORDER BY 1;
```

```
-- Check if nulls correlate with specific sources/categories
```

```
SELECT
```

```
 source_system,
```

```
 COUNT() as total,
```

```
 SUM(CASE WHEN shipping_country_code IS NULL THEN 1 ELSE 0 END) as null_count
```

```
FROM analytics.orders
```

```
WHERE created_at >= '2026-07-04T22:10:00Z'
```

```
GROUP BY 1
```

```
ORDER BY 2 DESC;
```

```
```
```

3. Deploy Suspect

Yes - the `checkout-api` deploy at 2026-07-04T22:10:00Z is highly suspect. The change "default empty string -> null for missing geo" perfectly explains the null rate spike. The `orders-ingest` deploy is not relevant.

4. Stakeholder Comms

"? The null rate spike in `shipping\_country\_code` is caused by last night's checkout-api deploy (22:10 UTC) changing missing geo data from empty string to null. We're rolling back the change and will backfill nulls to empty strings. Finance dashboard regional splits may be undercounting until fix is deployed (~30 min)."

Engineering counterpart

7:06 AM - Responded in 0s

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

Agent response

7:07 AM - Responded in 10s

Sure - here's the step-by-step reasoning I followed, along with the key assumptions.

Step 1 - Understand the baseline and spike

- Baseline 30?day average null rate: 0.3% -> that's a near-zero, stable rate.
- Yesterday's null rate: 17.8% -> massive, sudden jump.
- Today so far: 19.2% -> spike is persisting, not a one?day anomaly.
- Row count: 148,900 orders yesterday with a null `shipping\_country\_code` out of ~842,000 total over 7 days.

Implication: This is not a minor fluctuation. Something changed that causes many new orders to have a null country code instead of a non?null value (likely empty string previously).

Step 2 - Examine the deploys

| Service | Deploy Timestamp | Change Description |
|-----------------|----------------------|--|
| `orders-ingest` | 2026-07-03 08:00 UTC | "no schema change" |
| `checkout-api` | 2026-07-04 22:10 UTC | "default empty string -> null for missing geo" |

- The `orders-ingest` deploy is earlier (Jul 3) and explicitly says "no schema change" -> unlikely to cause a null spike that only appears on Jul 5 (yesterday).
- The `checkout-api` deploy is at Jul 4, 22:10 UTC - just before the spike day (Jul 5). The change directly says "default empty string -> null for missing geo".

If previously a missing country became an empty string `` (which is not null), now it becomes an actual SQL `NULL`. That perfectly explains the null rate jumping from 0.3% to ~18%.

Implication: The `checkout-api` deploy is the obvious culprit.

Step 3 - Validate with SQL

First query - confirm timing:

- Filter orders from the deploy time onward, group by day.
- Expected result: null rate immediately jumps from ~0% to near 20% from Jul 5 onward.

Second query - check correlation with source:

- If nulls are concentrated in orders from the checkout?api source (e.g., `source\_system = 'checkout'`), that further confirms the cause.
- Could also check `order\_channel` or `order\_type` if available.

Engineering counterpart

7:07 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.