AMAZON CONNECT · AI AGENT METRICS

Go-live scorecard for AI agents on Amazon Connect

The eight native metrics AWS added in April 2026, sorted into the two that gate a release, the four that diagnose it, and the ones that only prove the plumbing works.

ExecuteCX
executecx.com
September 2026

In brief. If you had to defend a go-live on two numbers, use **Goal Success Rate** (did the agent resolve the issue) and **AI Tool Utilization Accuracy** (did it call tools correctly where they write to systems of record). The first tells you whether the agent is failing; the second tells you whether it is failing dangerously. Both are LLM-judged and refresh every 24 hours, so give every change a 48-hour window before you read it.

STARTING THRESHOLDS: DEFAULTS TO OPEN THE CONVERSATION, NOT AWS BENCHMARKS

METRIC	GATE	WHY THIS BAR	YOURS
Goal Success Rate GOAL_SUCCESS_RATE · 0–1 · LLM-judged	**≥ 0.85**	Below this, human agents inherit too much rework. The one judgment score a non-technical stakeholder can read unaided.	☐
AI Tool Utilization Accuracy AI_TOOL_UTILIZATION_ACCURACY · derived	**≥ 0.95**	Wherever tools write to a system of record. A wrong tool call is a data defect that surfaces days later.	☐
Faithfulness Score FAITHFULNESS_SCORE · 0–1 · LLM-judged	**≥ 0.95**	In regulated contexts. A hallucinated policy detail on a recorded line is a compliance event.	☐
Completeness Score COMPLETENESS_SCORE · 0–1 · LLM-judged	**≥ 0.90**	Below this, multi-part requests come back as repeat contacts.	☐
Avg AI Tool Invocation Latency AVG_AI_TOOL_INVOCATION_LATENCY · mean, not p95	**< 1,500 ms**	On voice a slow tool call is silence. The mean hides the tail. Compute p95 from invocation-level data.	☐
Minimum evaluation volume sessions scored before you decide	**≈ 300**	At 0.85 observed, 300 sessions is about ±4 points of sampling error. Judge-model bias does not shrink with volume.	☐

TIER 1 · GATE THE RELEASE

Goal Success Rate · Tool Utilization Accuracy

Is the agent doing its job, and is it safe where it touches your systems. Everything else explains how it fails, not whether.

TIER 2 · DIAGNOSE

Faithfulness · Completeness · Tool Selection · Tool Parameter

Low Faithfulness, good Completeness: fluent but ungrounded. Low Selection, high Parameter: fix tool descriptions first. Readings are hypotheses.

TIER 3 · HYGIENE

Invocation success · latency · turns · knowledge refs

Proof the plumbing ran. A 99.8% invocation success rate is compatible with resolving nothing. Alert on them; never report them as quality.

OPERATING RULES

- **One change per 48 hours.** Evaluation scores refresh every 24 hours, feedback counts about every 6. Stack changes and you cannot attribute the movement.
- **Proportions, not call share.** A 0.78 Goal Success Rate is the share of agent sessions resolved, not the share of your call volume handled.
- **Alert on relative drops via GetMetricDataV2,** not dashboard floors, and route the alert to whoever owns the prompt.
- **Metric → trace → transcript.** The metric names the dimension, the trace names the conversation, the transcript names the fix. Traces take up to 30 minutes and need Automated Interaction Logs on.
- **Intent is not a native dimension.** An intent-level number means separate agents per use case or contact attributes joined downstream.

THE ONE THAT GETS MISUSED

Handoff rate. Driving it down is trivial: make the agent reluctant to escalate. You hit the target and strand every caller the agent cannot help. Read it only next to Goal Success Rate: handoffs down with goal success holding is progress; handoffs down with goal success falling is a containment illusion that resurfaces as repeat contacts.

NOT COVERED BY THE EIGHT

- Tone, brand fit and required disclosures: use custom generative-AI evaluations (May 2026).
- Cost per resolved contact, and contacts that never created an agent session.

Full write-up: executecx.com/post/amazon-connect-ai-agent-metrics · Definitions from the Amazon Connect Administrator Guide and the April 2026 AWS announcement. Thresholds are ExecuteCX operating defaults; calibrate per use case.