A PRACTICAL GUIDE

AI Monitoring with Splunk

How to track costs, token usage, quality and agent reliability

INSIDE

Contents

From the first question to AI monitoring in production

— Introduction 03

01 The AI monitoring challenge 06

02 What to measure 09

03 Splunk for AI monitoring 18

04 Getting started 22

AI is moving into production faster than ever

56%

of organisations have deployed or are rolling out their own AI agents

92%

saw machine data grow in the last year, led by AI agents and AI-assisted development

✦ More AI in production means more ways for it to go wrong quietly

Source: Splunk 2026 research

INTRODUCTION

Online is not the same as working

An AI application can be up and still fail in ways ordinary monitoring does not show

- **Slow answers**
- **Far more tokens than expected**
- **The wrong tool**
- **The same steps, repeated**
- **An unsupported answer**
- **No useful business result**

That is why AI monitoring needs a wider view than uptime, CPU and error rates

What AI monitoring means

Collecting and analysing evidence about how an AI application behaves, in development and in production.

> When something becomes slow, unreliable, expensive or inaccurate, the team can see what happened and where to look.

THE EVIDENCE

- Application logs
- Model calls
- Token usage
- Latency
- Errors
- Tool calls
- Retrieval steps
- Agent decisions
- Output evaluations
- Infrastructure signals
- Business outcomes

Not the same as AIOps

This guide is about monitoring AI workloads themselves: LLM applications, RAG systems, coding assistants and AI agents.

01

The AI monitoring challenge

Why AI needs a wider view than ordinary application monitoring

AI adds new ways to fail

STILL ESSENTIAL

Application performance monitoring

- ✓ Availability
- ✓ Error rates
- ✓ Response times
- ✓ Resource saturation

AI still runs on APIs, databases, containers, networks, queues and cloud services.

WHAT AI ADDS

Request succeeds	→	**Answer is irrelevant**
Every step completes	→	**Wrong tool chosen**
Documents return fast	→	**Context is weak**
Retries hide errors	→	**Latency and cost rise**
Pricier model, longer answers	→	**No better outcome**

01 · THE CHALLENGE

Six connected layers of visibility

1. **User & business journey**
   What the person wanted · Was the task completed
2. **Application**
   APIs · Services · Databases · Queues · User experience
3. **Agent**
   Planning · Handoffs · Retries · Tool selection · Stopping
4. **Model**
   Provider · Prompt version · Latency · Tokens · Finish reason
5. **Data & tools**
   Retrieval · Vector databases · Enterprise APIs · External services
6. **Infrastructure**
   Cloud · Containers · GPUs · Memory · Storage · Networks

One layer leaves gaps

Token use on a provider dashboard rarely shows which journey, feature, agent or failed workflow caused it.

WeAre

WeAre

02

What to measure

Cost, tokens, agents, quality and the infrastructure underneath

WeAre

Ten questions your AI monitoring should answer

DEMAND

How much work is the system receiving?

Requests · Runs · Active users

PERFORMANCE

Where is the user waiting?

Latency · First token · p50 and p95

RELIABILITY

Does the workflow finish reliably?

Errors · Timeouts · Retries · Escalations

TOKENS

What model consumption does each run create?

Input · Output · Cached · Reasoning

ESTIMATED COST

What is this consumption likely to cost?

Model rate × usage · Known estimates

AGENT BEHAVIOUR

What did the agent do?

Steps · Tool calls · Handoffs · Loops

QUALITY

Was the result good enough?

Correctness · Safety · Human acceptance

RETRIEVAL

Did the application receive useful context?

Search latency · Empty results · Relevance

INFRASTRUCTURE

Did the supporting platform cause the issue?

CPU · GPU · Memory · Queues

BUSINESS OUTCOME

Did the work create the intended result?

Completed tasks · Approved PRs · Resolved cases

WeAre

AI cost monitoring starts with tokens

INPUT

What the model reads

Prompt · Instructions · History · Retrieved context

OUTPUT

What the model writes

The response the user or agent receives

CACHED

What is reused

Repeated context served from the prompt cache

REASONING

How the model thinks

Billed reasoning activity, where the model reports it

Estimated model cost = **Input tokens × input rate** + **Output tokens × output rate** + **Cached, reasoning and other charges**

WHAT MOVES THE BILL

Traffic · Prompt length · Response length · Model choice · Agent steps · Retries · Caching

An estimate, not an invoice. Compare it with your provider and cloud invoices for financial reporting.

The hidden sources of AI cost



Token usage is only one part of the cost stack

Also below the surface: retrieval and embeddings, vector database storage, automated evaluations, data transfer and telemetry retention, and human review.

1. Repeated model calls
2. Third-party APIs and tools
3. GPUs, servers and containers
4. Failed runs, loops and retries

Connect AI spend to outcomes

WHERE DOES THE SPEND GO?

- ✓ **Useful work that creates accepted outcomes**
- Retries
- Oversized context
- Failed runs

QUESTIONS EVERY AI OWNER SHOULD ASK

Which product, team, model or agent drives the cost?

How much is useful work, and how much is waste?

What does each completed or accepted outcome cost?

Do the estimates match the invoices?

> Agree what success means first: an answer accepted without correction, a case resolved, a pull request approved.

WeAre

One request, many steps



RECORD

- Trace and run ID
- Agent and model version
- Tool status and duration
- Documents used
- Tokens
- Retries and handoffs
- Human decision
- Final result

WeAre

Measure the whole workflow

PERFORMANCE

Where is the user waiting?

Time to first token	Total response time
Model latency	Retrieval latency
Tool call latency	Queue time
p50, p95 and p99	Throughput

RELIABILITY

Does the workflow finish?

Completed runs	Accepted results
Tool failures and timeouts	Retries and fallbacks
Loops	Runs needing a person
Correction rounds	Failures by release

A retry can make a request look successful. Track the original failure and the recovery path.

It ran. But was it good?

DEFINE QUALITY FOR EACH USE CASE

- Correctness
- Relevance
- Groundedness
- Completeness
- Correct tool use
- Policy compliance
- Prompt injection
- Data leakage
- Human acceptance

There is no universal quality score

A support assistant, a coding agent and a document classifier each need their own tests.

COMBINE AUTOMATED AND HUMAN REVIEW

1. Build a representative test set
2. Define expected results and failure categories
3. Evaluate before release and on live traffic
4. Sample results for human review
5. Compare scores with human decisions
6. Track changes by prompt, model, agent and data

WeAre

Connect the signal to the AI journey

Model platforms

Amazon Bedrock · Azure OpenAI · GCP Vertex AI · OpenAI

GPUs and model serving

NVIDIA GPU · NVIDIA NIM · vLLM · Serving platforms

Vector databases

Pinecone · Weaviate · Milvus

Frameworks and gateways

LlamaIndex · AI gateways

CORRELATE

The AI journey

- Slow
- Failed
- Expensive
- Low quality

03

Splunk for AI monitoring

A gradual path, from run logs to dedicated AI capabilities

Which Splunk tool answers which question?

Also available: AI Security Monitoring with Cisco AI Defense, for AI-specific security risks

What happened in each run?

Structured run logs in Splunk Cloud

Runs · Failures · Duration · Tokens · Cost

What are tokens costing us?

AI Token and Cost Monitoring

Agents · Providers · Models · Tool calls

MAIN PLATFORM

Is the agent doing the right thing?

Splunk Agent Observability

Traces · Evaluations · Guardrails · Model comparison

Are coding agents paying off?

Tokenomics

Usage · Cost · Adoption · Productivity

Is the stack underneath healthy?

AI Infrastructure Monitoring

GPUs · Model serving · Vector DBs · Gateways

Is it observable before release?

OpenTelemetry and Observability Studio

Local workspace · VS Code · Open source

WeAre

Start where you are, grow from there

1 · START

Structured run logs in Splunk Cloud

Build on the logs and alerts you already have. See duration, failures, tokens and estimated cost for every run.

2 · DEEPEN

AI Token and Cost Monitoring

Token use and cost by agent, provider, model and tool call, with rates you can match to your prices.

AI Infrastructure Monitoring

GPUs, model serving, vector databases and gateways.

3 · EXPAND

Splunk Agent Observability

Trace agent workflows, connect quality, performance and cost, run evaluations and apply guardrails while agents run.

Tokenomics

See how teams use AI coding agents, what they cost and where they add most value.

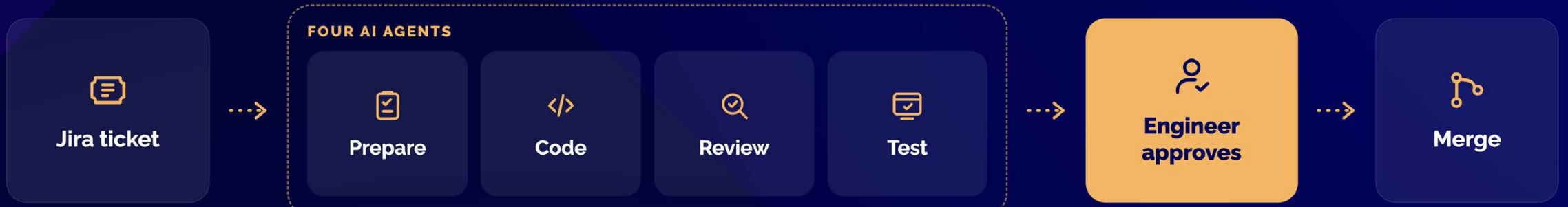
OpenTelemetry underneath · one open standard for metrics, traces and logs

Capabilities vary by deployment and subscription

WeAre

From Jira ticket to reviewed pull request

Read the case study →



3 Renta applications
Easy · Task · Tools

90 Minute time limit
per agent run

2 Correction rounds
then back to a person

Every run in Splunk Cloud
via AWS CloudWatch

- Task and product
- Duration
- Token usage
- Model cost
- Exit status
- Failures and timeouts

WeAre

AI monitoring, step by step

Start with one workload and prove the data helps a real decision

PLAN

1. One workload, one question
2. Define success and failure
3. Map the workload

→

BUILD

4. Decide the minimum data
5. Set privacy and access rules
6. Instrument and test

→

PROVE AND SCALE

7. Dashboards and alerts that drive action
8. Check costs and review results
9. Scale once the pilot delivers

04 · AVOID THESE

Common AI monitoring mistakes

⊗ The mistake ✓ What to do instead

Treating estimates as invoices

✓ Check against the invoice

Capturing every prompt by default

✓ Decide what you need first

Measuring tokens without outcomes

✓ Connect cost to results

Looking only at averages

✓ Check p95 latency and cost

Ignoring retries and loops

✓ Track the whole workflow

Starting with dashboards, not questions

✓ Define the decision first

Not matching Splunk to your setup

✓ Choose the right fit early

Alerting on every unusual value

✓ Alert on context and impact

WeAre

From first question to production

AI engineering | Software development | Atlassian solutions | Splunk observability

STAGE 1

Readiness review

One workload, one question, your data, privacy and the right Splunk setup

✓ **A focused plan and a clear decision**

STAGE 2

One-workload pilot

Instrument, connect to Splunk, test failure and cost scenarios, build first dashboards

✓ **Real evidence within weeks**

STAGE 3

Rollout and improvement

More workloads, better data quality, alerts, cost reviews and model comparisons

✓ **Monitoring that grows with your AI**

Every engagement starts with agreed scope, success measures and data boundaries

AI Engineering | Splunk Observability | Atlassian

Ready to see what your AI is really doing?

Let's start with one workload.

Talk to WeAre →

weare.fi