

Upskill with StackOps Series

AI Assistants

Nisham Muhd Kovallil, Solution Consultant, StackOps

Upskill with StackOps Series

Smarter Troubleshooting with StackOps AI Assistant

Nisham MUHD Kovvalil,
Solutions Consultant, StackOps



AI-Driven Observability

Evolution of Observability



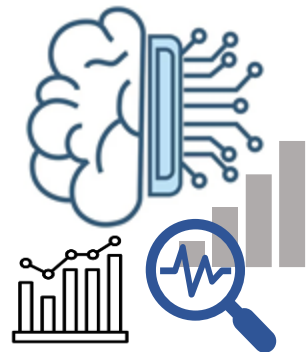
Monitoring

- What is broken
- Known - Unknown
- Siloed tools
- Predefined set of Metrics



Observability

- Why it's broken
- End-to-end Visibility
- Performance insights
- Unified Monitoring



AI-powered Observability

- Proactive Monitoring
- Intelligent Correlation
- Smart Remediation

Contents

AI Basics-Good to Know	7
How AI Assistant Works	8
AI Assistant in Action: Use Cases	9
<i>AI Driven Troubleshooting & RCA</i>	10
<i>Query Creation: ESQL</i>	11
<i>Dashboarding Building Using AI Assistant</i>	12
<i>Security: Attack Discovery</i>	13
<i>Enrich data using LLM</i>	14
AI Connector Setup	16
Knowledge Base	19
Anonymization	22
Coming soon in StackOps Serverless	25

AI Basics

Tokens: "Currency of AI"

- Text split into smaller chunks
- Basis for model billing costs
- Define strict input/output length limits

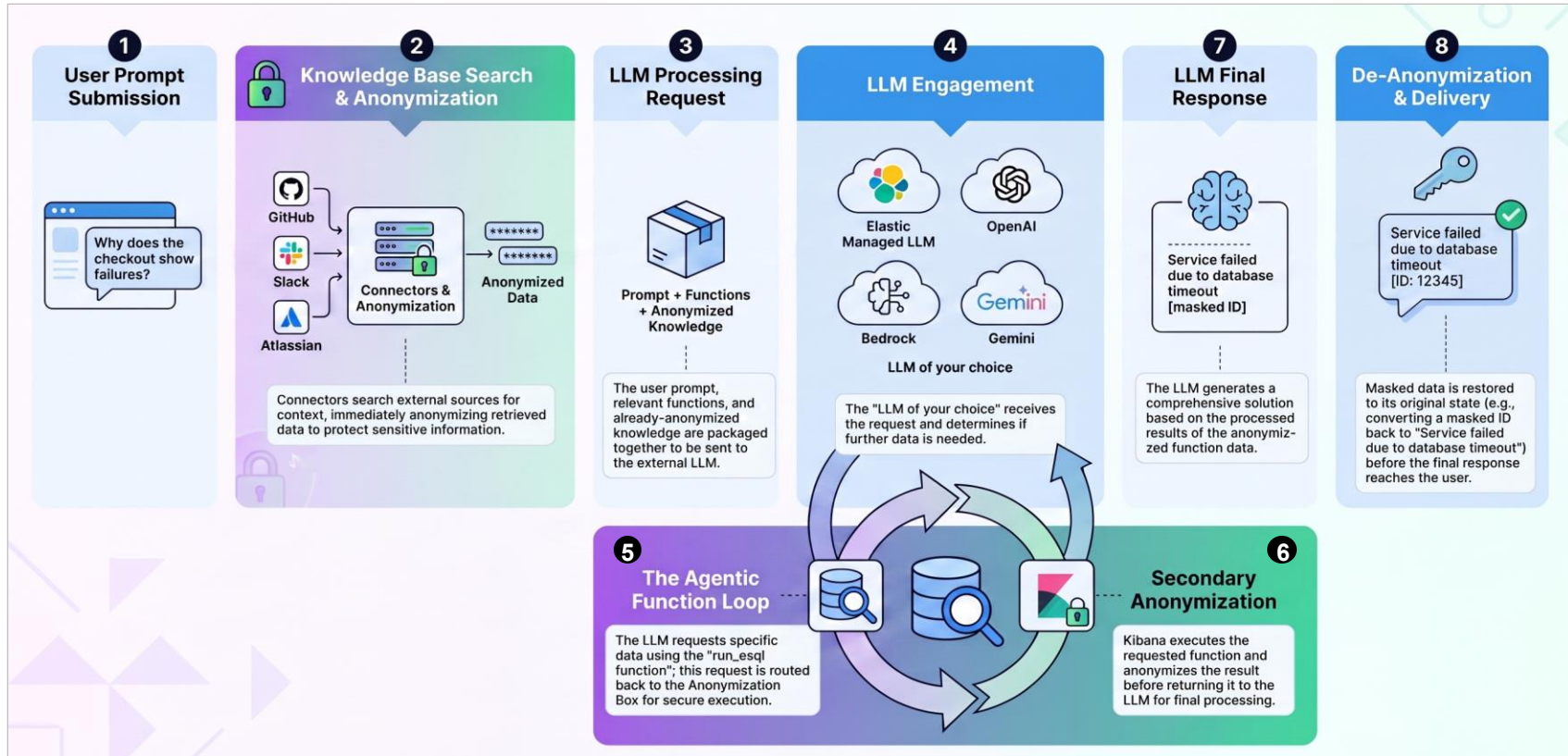
Context Window: "Short-term Memory"

- Amount of information (tokens) AI can handle per session
- Includes history & retrieved data
- Limits the depth of conversation

Rate Limits: "The Traffic Cop"

- Prevents over-utilization risks
- Managed by tokens or requests
 - RPM (Request per min)
 - TPM (Tokens per min)
- Critical for budgeting and performance

How AI Assistant Works?



AI Assistant in Action: Use Cases

AI Driven Troubleshooting & RCA

Long hours on RCA?

"Reduce hours of complex troubleshooting to minutes with AI-guided root cause analysis"

Siloed Infra & App data?

"AI Assistant automatically correlates distributed traces, logs and infrastructure metrics"

Unfamiliar with errors?

"Instantly decode complex stack-traces and cryptic error messages into plain English."


Shopping App SLO Violation Analysis
Private


Elastic Assistant

Now I have the full picture. Here's the complete investigation:


Payment SLO — Deep Dive Investigation

SLO Summary

Metric	Value
Status	VIOLATED
SLI Type	APM Transaction Error Rate
SLI Value	0.26% success (99.74% of transactions are failing!)
Target	99.95%
Error Budget Remaining	-1993.7% (massively exhausted)
Service	payment
Total Events (7d rolling)	37,186 transactions
Good Events	Only 98 out of 37,186


Root Cause Identified

Every single failure in the payment service is the **same error**:

```
Payment request failed. Invalid token. app.loyalty.level=gold
```

- Error Type:** Error
- Failing Transaction:** grpc.oteldemo.PaymentService/Charge
- Failure Rate:** ~99.7% of all payment charge requests are failing
- Error Count (last hour):** 141 errors / 111 failed transactions

What's Happening

- The **Payment Service** is rejecting charge requests with an **"Invalid token"** error specifically for customers with **loyalty level = gold**.
- This appears to be an **authentication/authorization token issue** — the payment token being passed for gold-tier loyalty customers is invalid or expired.
- The transaction duration is very low (~400-450µs), meaning the service is **failing fast** — it's rejecting requests immediately rather than timing out.

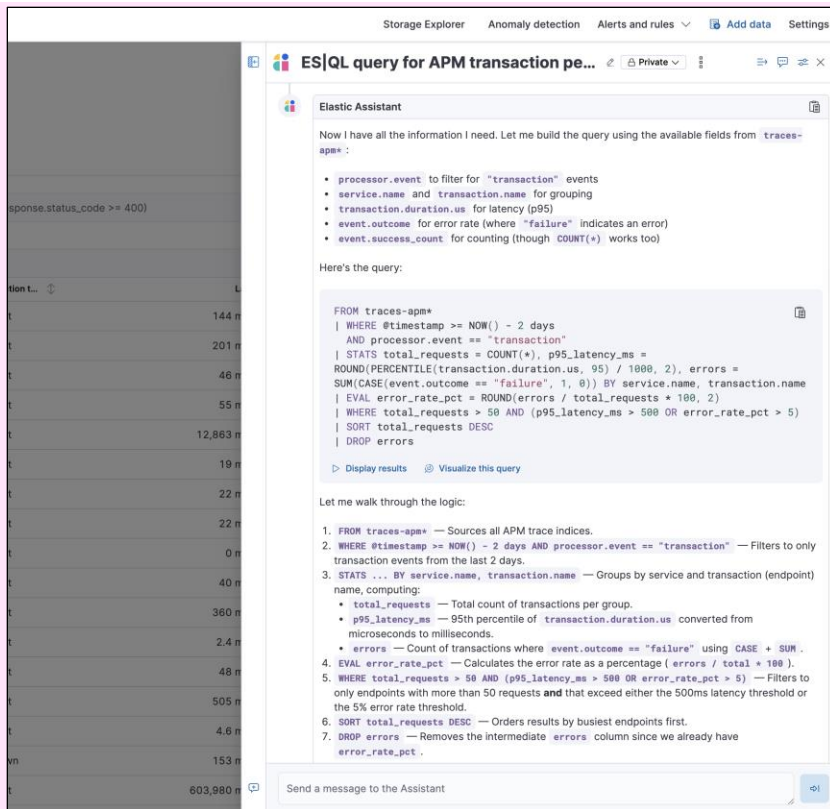
Query Creation: ESQL

**Complex Syntax;
Holding you back from Analysis?**

**"Generate advanced, highly optimized
ESQL queries using simple natural
language prompts."**

**Struggling to optimize
searches**

**"Ask the AI Assistant to refine and
speed up your custom queries for
maximum performance."**



The screenshot shows the ES|QL query interface for APM transaction performance. The left pane displays a complex query with various filters and aggregations. The right pane shows the 'Elastic Assistant' interface, which provides a natural language explanation of the query logic.

ES|QL query for APM transaction performance:

```
FROM traces-apm*
| WHERE @timestamp >= NOW() - 2 days
  AND processor.event == "transaction"
| STATS total_requests = COUNT(*), p95_latency_ms =
  ROUND(PERCENTILE(transaction.duration.us, 95) / 1000, 2), errors =
  SUM(CASE(event.outcome == "failure", 1, 0)) BY service.name, transaction.name
| EVAL error_rate_pct = ROUND(errors / total_requests * 100, 2)
| WHERE total_requests > 50 AND (p95_latency_ms > 500 OR error_rate_pct > 5)
| SORT total_requests DESC
| DROP errors
```

Elastic Assistant explanation:

Now I have all the information I need. Let me build the query using the available fields from `traces-apm*`:

- `processor.event` to filter for "transaction" events
- `service.name` and `transaction.name` for grouping
- `transaction.duration.us` for latency (p95)
- `event.outcome` for error rate (where "failure" indicates an error)
- `event.success_count` for counting (though `COUNT(*)` works too)

Here's the query:

```
FROM traces-apm*
| WHERE @timestamp >= NOW() - 2 days
  AND processor.event == "transaction"
| STATS total_requests = COUNT(*), p95_latency_ms =
  ROUND(PERCENTILE(transaction.duration.us, 95) / 1000, 2), errors =
  SUM(CASE(event.outcome == "failure", 1, 0)) BY service.name, transaction.name
| EVAL error_rate_pct = ROUND(errors / total_requests * 100, 2)
| WHERE total_requests > 50 AND (p95_latency_ms > 500 OR error_rate_pct > 5)
| SORT total_requests DESC
| DROP errors
```

Let me walk through the logic:

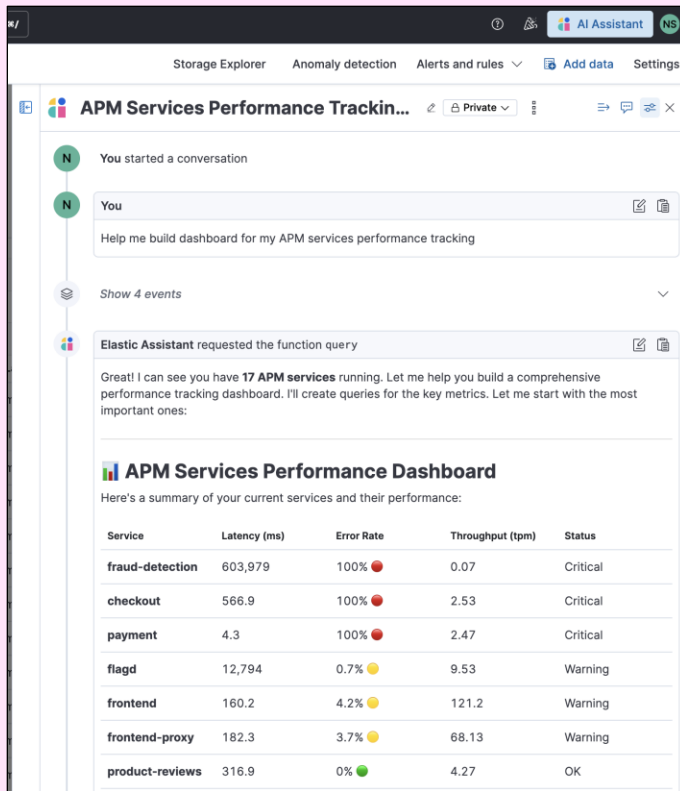
- `FROM traces-apm*` — Sources all APM trace indices.
- `WHERE @timestamp >= NOW() - 2 days AND processor.event == "transaction"` — Filters to only transaction events from the last 2 days.
- `STATS ... BY service.name, transaction.name` — Groups by service and transaction (endpoint) name, computing:
 - `total_requests` — Total count of transactions per group.
 - `p95_latency_ms` — 95th percentile of `transaction.duration.us` converted from microseconds to milliseconds.
 - `errors` — Count of transactions where `event.outcome == "failure"` using `CASE + SUM`.
- `EVAL error_rate_pct` — Calculates the error rate as a percentage (`errors / total * 100`).
- `WHERE total_requests > 50 AND (p95_latency_ms > 500 OR error_rate_pct > 5)` — Filters to only endpoints with more than 50 requests and that exceed either the 500ms latency threshold or the 5% error rate threshold.
- `SORT total_requests DESC` — Orders results by busiest endpoints first.
- `DROP errors` — Removes the intermediate `errors` column since we already have `error_rate_pct`.

Dashboard Building

Tedious Dashboard Building?

"Skip manual work.
Ask the AI Assistant to build perfect visualizations for your data"

"Help me build dashboard for my APM services performance tracking.."



APM Services Performance Tracking

You started a conversation

You: Help me build dashboard for my APM services performance tracking

Show 4 events

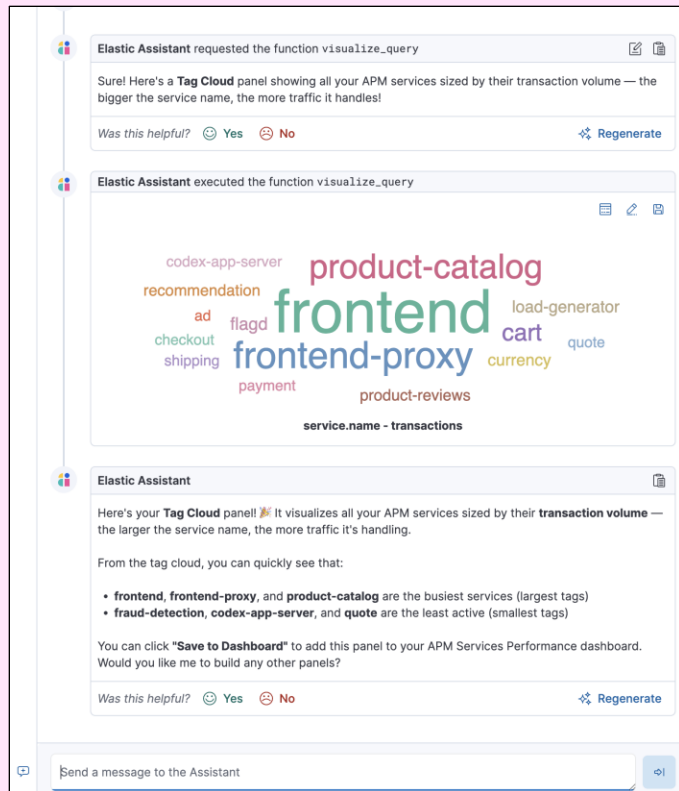
Elastic Assistant requested the function query

Great! I can see you have **17 APM services** running. Let me help you build a comprehensive performance tracking dashboard. I'll create queries for the key metrics. Let me start with the most important ones:

APM Services Performance Dashboard

Here's a summary of your current services and their performance:

Service	Latency (ms)	Error Rate	Throughput (tpm)	Status
fraud-detection	603,979	100% ●	0.07	Critical
checkout	566.9	100% ●	2.53	Critical
payment	4.3	100% ●	2.47	Critical
flagd	12,794	0.7% ●	9.53	Warning
frontend	160.2	4.2% ●	121.2	Warning
frontend-proxy	182.3	3.7% ●	68.13	Warning
product-reviews	316.9	0% ●	4.27	OK



Elastic Assistant requested the function visualize_query

Sure! Here's a **Tag Cloud** panel showing all your APM services sized by their transaction volume — the bigger the service name, the more traffic it handles!

Was this helpful? 😊 Yes 😞 No [Regenerate](#)

Elastic Assistant executed the function visualize_query

product-catalog
frontend
frontend-proxy
service.name - transactions
codex-app-server
recommendation
ad
flagd
checkout
shipping
payment
load-generator
cart
quote
currency
product-reviews

Elastic Assistant

Here's your **Tag Cloud** panel! 🏷️ It visualizes all your APM services sized by their **transaction volume** — the larger the service name, the more traffic it's handling.

From the tag cloud, you can quickly see that:

- **frontend**, **frontend-proxy**, and **product-catalog** are the busiest services (largest tags)
- **fraud-detection**, **codex-app-server**, and **quote** are the least active (smallest tags)

You can click **"Save to Dashboard"** to add this panel to your APM Services Performance dashboard. Would you like me to build any other panels?

Was this helpful? 😊 Yes 😞 No [Regenerate](#)

Send a message to the Assistant

AI Driven Attack Discovery

Collapse Alert Noise into one Story

"Connect the dots across fragment alerts to reveal the full narrative of potential breach."

Blind spots in Security?

"Accelerate attack discovery and investigate security incidents with AI context"

Attack discovery

Run
Schedule

Search...
Visibility 2
Status 2
Connector
Last 24 hours
Refresh

1 discovery | 13 alerts
Show anonymized values

AWS IAM Manipulation and Log Tampering
Open
Not shared

Jun 16, 2026 @ 16:10:10+13 Created by: rishant_mishra_kovvali_from.secmatindra@tech.gov.sg

Attack chain:
Alerts: 13
Take action

Actor: 9d6e5c3e-c5a6-4ce0-b19c-ce5cc397f528 conducted AWS IAM and CloudTrail attacks from 104.30.161.22
View in AI Assistant

Attack discovery Alerts

Summary

9d6e5c3e-c5a6-4ce0-b19c-ce5cc397f528 from 104.30.161.22 performed a full IAM role manipulation lifecycle followed by CloudTrail log tampering to cover their tracks.

Details

- Discovery/Reconnaissance (07:21-07:23 UTC): 9d6e5c3e-c5a6-4ce0-b19c-ce5cc397f528 from 104.30.161.22 began probing us-east-1 with multiple unauthorized API calls (access denied), indicating enumeration of available permissions and services.
- IAM Role Manipulation (07:23-07:35 UTC): The actor executed a sustained campaign of IAM role modifications across us-east-1, performing at least 6 role modification operations — likely attaching permissive policies, modifying trust relationships, or removing permission boundaries to enable privilege escalation.
- Backdoor Creation (07:31 UTC): 9d6e5c3e-c5a6-4ce0-b19c-ce5cc397f528 created a new IAM role (T1136.003 – Cloud Account), establishing a persistent backdoor or shadow admin identity within the AWS environment.
- Evidence Cleanup (07:35 UTC): The actor deleted the IAM role, suggesting deliberate cleanup of artifacts created during the attack to hinder forensic investigation and evade detection.
- Defense Evasion – Log Tampering (07:43 UTC): 9d6e5c3e-c5a6-4ce0-b19c-ce5cc397f528 updated CloudTrail trail configuration in ap-southeast-1, consistent with impairing cloud logging (T1562.008) to reduce visibility into prior and future actions. All activity originated from 104.30.161.22 with successful outcomes.

Attack Chain

Reconnaissance
Resource Development
Initial Access
Execution
Persistence
Privilege Escalation
Defense Evasion
Credential Access
Discovery
Lateral Movement
Collection
Command & Con

View in AI Assistant
Investigate in Timeline

Rows per page: 10

13

Enrich data using LLM

Summarization

"Automatically translate complex, multiline error stack traces into simple, clear explanations using LLM to save triage time."

Risk Summary

"Evaluate the incoming data using the LLM to provide a clear risk summary, determining whether the event may lead to potential threat"

Remediation Suggestions

"Analyze the error stack trace to automatically generate mitigation steps and enrich the data directly"

elastic

Find apps, content, and more.

<

AI Connectors

Centralized Access : Securely manage API keys and credentials for multiple LLM providers in one place.

Model Agnostic : Seamless switching between models/providers via Kibana.

Elastic Managed LLM

- Pre-configured LLM, ready-to-use
- Hosted in Singapore
- Pay only for the token used

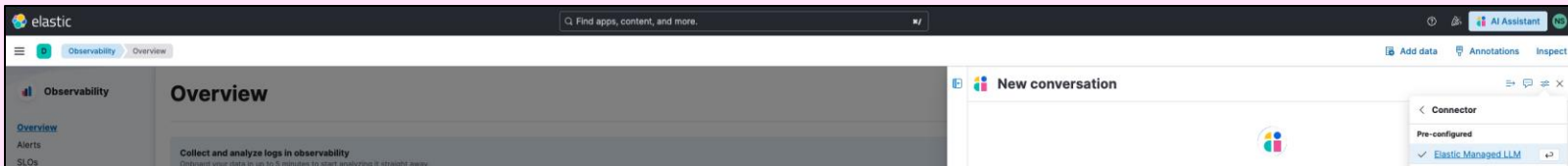
Bring your own LLM

- OpenAI
- Google Gemini
- Amazon Bedrock
- AI Connectors

AI Connector Setup

Elastic Managed LLM

- Out-of-the-box AI capabilities without requiring external subscriptions – **no setup needed, ready to use.**
- Hosted in Singapore, powered by Anthropic Claude Sonnet 4.6
- Pay only for the **tokens you use** ([refer docs for pricing details](#))



AI Connector Setup

Bring your own LLM

- Integrate AI assistant with your large language model (LLM) provider through connectors.
 - **OpenAI connector** for OpenAI or Azure OpenAI Service.
 - **Amazon Bedrock** connector for Amazon Bedrock, specifically for the Claude models.
 - **Google Gemini** connector for Google Gemini.

Connectors

[Documentation](#)[Create connector](#)

Connect third-party software with your alerting data.



OpenAI

Send a request to an OpenAI or Azure OpenAI service.



Amazon Bedrock

Send a request to Amazon Bedrock.



Google Gemini

Send a request to Google Gemini.

Knowledge Base

Bridging the Context Gap

Standard LLMs lack information about your specific environment. Knowledge Base fills this gap using RAG (Retrieval Augmented Generation)

- **Proprietary Context** -- Access internal playbooks and infrastructure details.
- **Up-to-Date Info** -- Sync with latest company or product documentation.
- **Reduced Hallucination** -- Responses are grounded in provided facts.

Knowledge Base

Setting Up Your Knowledge Base

Requirement	Specification
ML Node	Recommend to have ML node with min 4GB
Models	ELSER or E5 (Available as part of ML nodes)

Manual Configuration

Quickly **add entry** using **Kibana UI**. Simply copy-paste **markdown runbooks, text instructions, or policy on-call rosters** directly into the UI.



Connected Ecosystems

Use Elastic's content connectors to sync information seamlessly from **Atlassian** , **Slack**, **GitHub** etc. for KB by leveraging the **semantic_text** data type.



20

Anonymization

Privacy First: Anonymize your data

Why It's critical:

Protects unintended data exposure before it leaves your env for external LLMs.

Compliance: Meet strict industry-specific regulations by preventing unintended data from leaving your network.

Provider Trust: Prevent third-party LLM providers from training models on your proprietary infrastructure details.

Anonymization

Anonymization Workflow

1. Prompt

"User give prompt via Kibana"

2. Anonymize

"Inference logic strips unintended data based on the anonymization settings"

3. LLM Evaluation

"The anonymized prompt is passed to LLM for core logical evaluation"

4. De-Anonymize

"Elastic populate the original values back into the AI Assistant UI in Kibana"

Anonymization

Anonymization Settings

AI Assistants Observability

Anonymization Settings

[technical preview] List of anonymization rules

- **type:** "ner" or "regex"
- **entityClass:** (regex type only) eg: EMAIL, URL, IP
- **pattern:** (regex type only) the regular-expression string to match
- **modelId:** (ner type only) ID of the NER (Named Entity Recognition) model to use
- **enabled:** boolean flag to turn the rule on or off
- **timeoutSeconds:** (ner type only) maximum seconds *per inference request* before timing out (multiple requests may be issued during a single chat interaction)

Simulate function calling

ai:anonymizationSettings

```
{
  "rules": [
    {
      "entityClass": "EMAIL",
      "type": "RegExp",
      "pattern": "([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,})",
      "enabled": false
    },
    {
      "type": "NER",
      "modelId": "elastic__distilbert-base-uncased-finetuned-conll03-english",
      "enabled": false,
      "allowedEntityClasses": [
        "PER",
        "ORG",
        "LOC"
      ],
      "timeoutSeconds": 30
    }
  ]
}
```

observability:aiAssistantSimulatedFunctionCalling

Coming Soon in Serverless...

Elastic AI Capabilities: Serverless

Elastic Agent Builder

Agent Builder is Elastic's AI conversational platform for creating agents that answer questions and act (From AI answers to AI actions – using Tools & Skills)

Custom Instructions

Define your agent's ultimate objective, boundaries, tone of voice, and default LLM behavior in plain language.

Modular Skills

Equip agents with predefined "expert packages" to let them handle specific tasks.

Tool Catalogs

Grant your agent's capability to interact with internal data, run automated tasks or communicate to external services securely.

Guardrails & Safety

Enforce data anonymization, apply system limits, filter unintended data queries etc.

Tools are the mechanism that connects agent reasoning to data

- Built-in tools – Local index search using run_esql, fetch apm details, visualize etc.
- Custom tools – Integrate external tools like GitHub, Atlassian, Slack etc.

Elastic AI Enablement: Serverless

Elastic Workflows & MCP Servers

An automation engine build into Elastic platform

Mitigate Alert Fatigue: "Automated immediate responses, deduplication and sorting to reduce tedious manual triages for teams"

Eliminate Repetition: "Avoid human error or manual, recurring tasks and automate consistently."

Unify Tool Stack: "Eliminate the need to procure, configure and maintain external tools for automation."

MCP Server

- Integrate external data into your AI agents to expand their context and problem-solving capabilities.
- Securely connect external AI to your Elastic data using standardized MCP endpoint.

AI Can Make Mistakes, Review Before You Act !!!

Let's hear from you!

Thank you.