

Modernizing Incident Response

with LLMs, RAG and the MCP

Theofilos Papapanagiotou, Amazon

 **CRITICAL**

now

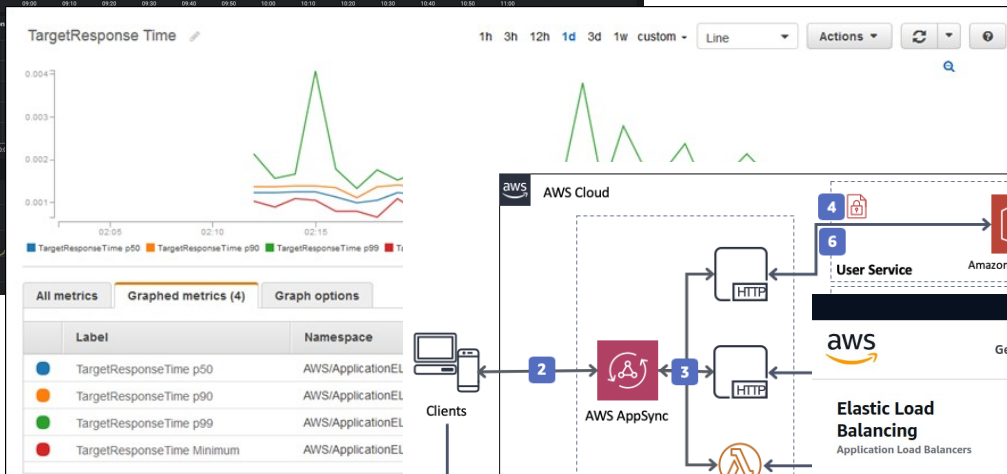
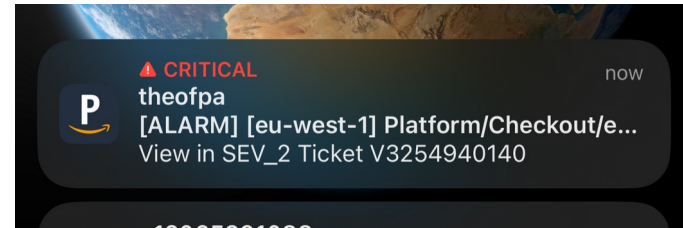
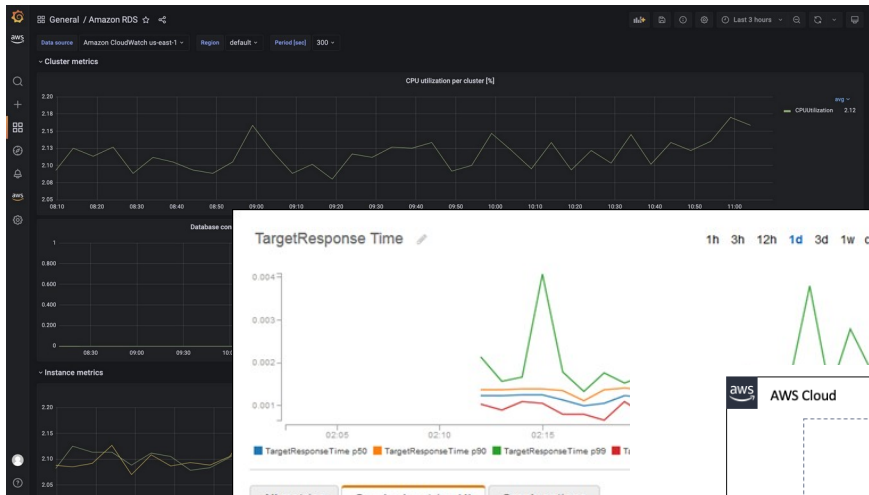


theofpa

[ALARM] [eu-west-1] Platform/Checkout/e...

View in SEV_2 Ticket V3254940140

+10005001000

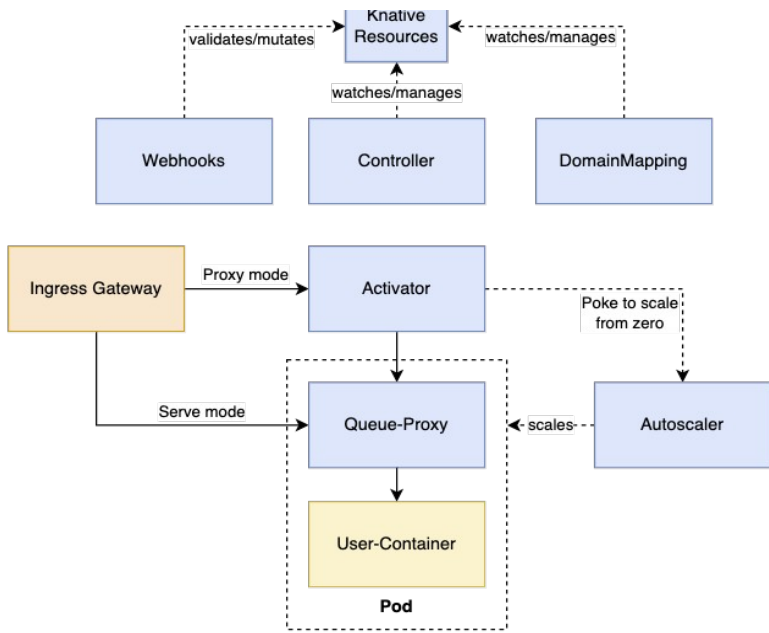


The screenshot shows the AWS console page for 'Elastic Load Balancing' under 'Access log files'. The page title is 'Access log files'. The main content explains that Elastic Load Balancing publishes a log file for each load balancer node every 5 minutes. It provides the format for the file names: `bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing`. A table defines the components of this format:

Component	Description
<code>bucket</code>	The name of the S3 bucket.
<code>prefix</code>	(Optional) The prefix (logical hierarchy) for the bucket. The prefix that you specify must not include the string <code>AWSLogs</code> . For more information, see Organizing objects using prefixes .
<code>AWSLogs</code>	We add the portion of the file name starting with <code>AWSLogs</code> after the

Hundreds of tools

Human cost



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: config-autoscaler
  namespace: knative-serving
data:
  requests-per-second-target-default: "150"
```



Can our systems help us think?

Scripts



Automation



Cognition



Understanding

Playbooks
Runbooks

Automation
Standardization

Agents
Reasoning
Context awareness

```
git/proserve/session-manager on 🍏 us-west-2
> aws ssm start-session \
  --target i-099aab1a8d5d7cbbb \
  --document-name simple

Starting session with SessionId: dliggat-lsengard-0ef
echo Connected to $(hostname)
sh-4.2$ echo Connected to $(hostname)
Connected to ip-172-31-0-144.us-west-2.compute.intern
sh-4.2$
sh-4.2$
```

Monitoring/EdgeTrafficDrop

EdgeTrafficDrop is a Prometheus Alertmanager alert defined in traffic.yaml in the operations/alerts repo. The alert fires if there is a significant percentage difference in request rate compared to the recent past, and may be indicative of traffic anomalies.

Things to do

Check the dashboard varnish-caching-last-week-comparison for the affected cluster/site. For example, if the alert says "44% GET drop in text@codfw during the past 30 minutes", you want to select the **text** cluster and **codfw** as the site. If the shape of the curve is a clear drop without previous increase as shown in the image **Traffic Drop** on the right, this could mean that we served less traffic than normal due to either an attack or some anomalies in our infrastructure. If the pattern does not seem to recover on its own, page the Traffic team.

If instead the curve looks like a spike as show in the image **Traffic Spike** on the right, that is likely due to some anomalous incoming traffic and in general there's not much to worry about.

Regardless of the shape of the curve, you should do the following:

- Look at the load-balancers-logs dashboard for the given site. If you see clear spikes there, it's probably a DoS attack. See the [DDoS Playbook](#).
- Take a look at varnish-aggregate-logs-status-codes for the relevant site/cluster to learn more about the type of traffic, in particular whether any specific method/status code stands out.
- Dig into webrequest_sampled_128 on Tumblelog for the specific details of the type of requests causing the spike.
- Let #wikimedia-traffic know about your findings.

Categories: Runbooks | SRE Traffic

Status	Pipeline	Triggerer	Commit	Stages
passed	#319 latest		Pmaster -> b003e37f	
passed	#308		Pmaster -> b2a78317	
failed	#306		Pmaster -> 2818d8eb	

<thinking> To analyze the metrics seen in the Grafana dashboard, I need to use the `image_reader` tool to extract information from the image. This will allow me to identify the metrics, their values, and any anomalies or patterns that may be present. </thinking>

Scripts → Cognition

Agenda

- Data
- Reasoning
- Co-operation
- Evaluation
- Future

Before: Institutional knowledge

Monitoring/EdgeTrafficDrop

Page Discussion Read View source View history Tools

< Monitoring

EdgeTrafficDrop is a [Prometheus Alertmanager alert](#) defined in [traffic.yaml](#) on the operations/alerts repo. The alert fires if there is a significant percentage difference in request rate compared to the recent past, and may be indicative of traffic anomalies.

Things to do

Check the dashboard [varnish-caching-last-week-comparison](#) for the affected cluster/site. For example, if the alert says "44% GET drop in text@codfw during the past 30 minutes", you want to select the **text** cluster and **codfw** as the site. If the shape of the curve is a clear drop without previous increase as shown in the image *Traffic Drop* on the right, this could mean that we served less traffic than normal due to either an attack or some anomalies in our infrastructure. If the pattern does not seem to recover on its own, page the Traffic team.

If instead the curve looks like a spike as shown in the image *Traffic Spike* on the right, that is likely due to some anomalous incoming traffic and in general there's not much to worry about.

Regardless of the shape of the curve, you should do the following:

- Look at the [load-balancers-lvs](#) dashboard for the given site. If you see clear spikes there, it's probably a DoS attack. See the [\(D\)DoS Playbook](#).
- Take a look at [varnish-aggregate-client-status-codes](#) for the relevant site/cluster to learn more about the type of traffic, in particular whether any specific method/status code stands out.
- Dig into `webrequest_sampled_128` on [Turnilo](#) for the specific details of the type of requests causing the spike.
- Let [#wikimedia-traffic](#) know about your findings.



Categories: Runbooks | SRE Traffic

This page was last edited on 27 July 2022, at 16:08.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About Wikitech](#) [Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#) [Mobile view](#)



<https://wikitech.wikimedia.org/wiki/Monitoring/EdgeTrafficDrop>

Monitoring/root disk space

Page Discussion Read View source View history Tools

< Monitoring

This [Icinga](#) alert says the disk space on the / file system is too low.

Some things you can do are:

- `apt-get clean` (deletes cached package files)
- [find largest files recursively using find](#) and examine if any can be deleted
- check `/var/log/` and its sub directories for their sizes (`du -hs *`) and work your way down to find what is taking the most space
- `gzip` or `tar` some (older) log files
- move some files to `/srv` if that is mounted on a separate volume
- create a ticket and/or ping somebody on IRC who is familiar with the service to see what can be deleted
- create patches to add or improve logrotate setup
- create tickets to request more disk space / hardware if really needed

Categories: Monitoring | Runbooks

This page was last edited on 30 July 2019, at 19:17.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. See [Terms of Use](#) for details.

[Privacy policy](#) [About Wikitech](#) [Disclaimers](#) [Code of Conduct](#) [Developers](#) [Statistics](#) [Cookie statement](#)

[Mobile view](#)



https://wikitech.wikimedia.org/wiki/Monitoring/root_disk_space

Consequence

Time goes to navigation, not reasoning.



Tribal expertise

@here Has anyone seen this before?

called back into OpenSearch. It worked, however its causing them to scale out

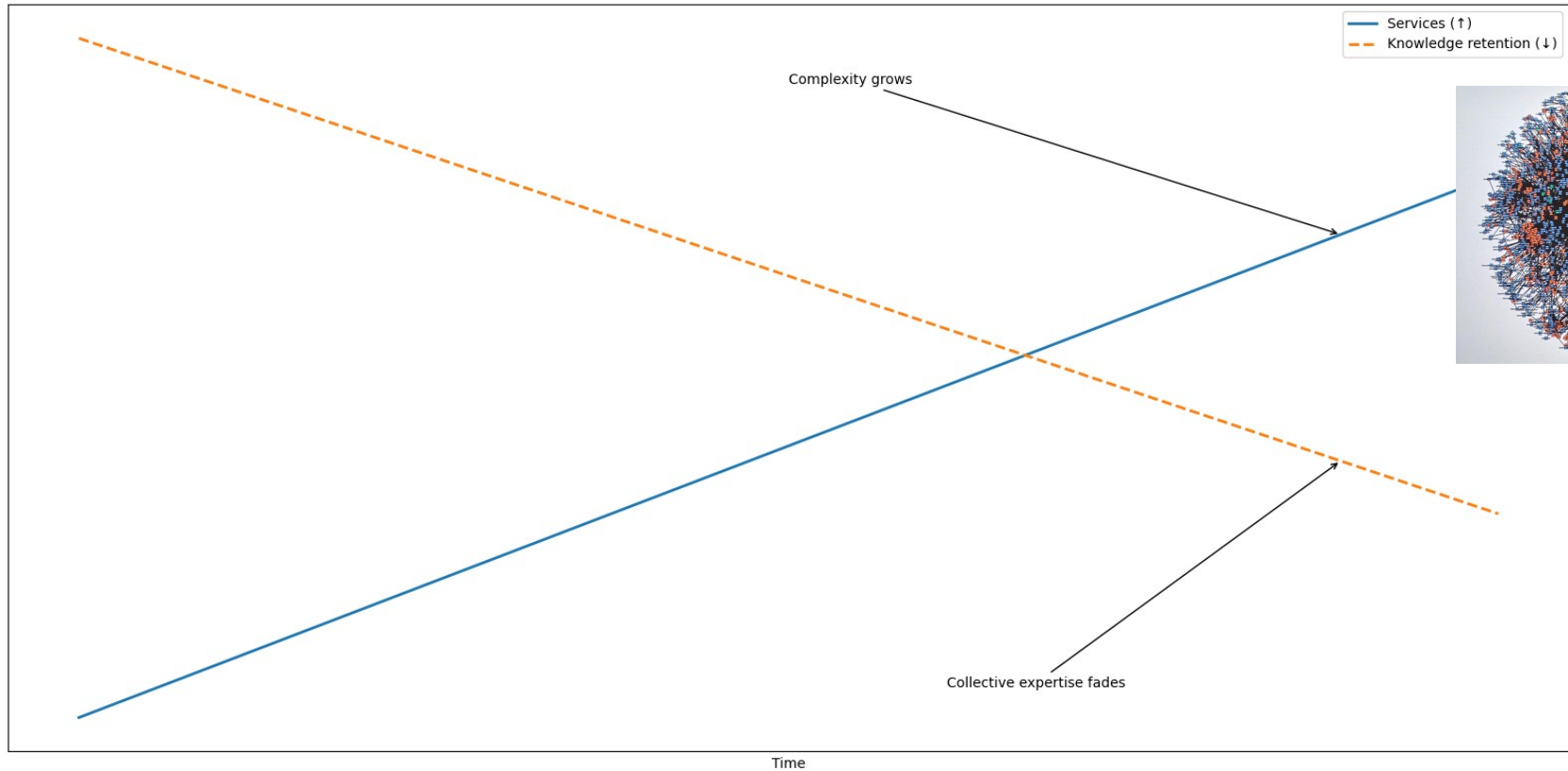
Has anyone seen this before?

and invalidating caches and restarting. Has anyone seen this before and fixed it?

capacity mode for these tables. Has anyone seen this before, if

Institutional knowledge ➡ *Inaccessible content*

The Scaling problem



What we need



One-ask fan-out: from a single prompt, hit dashboards to gather metrics, search logs, relevant tickets and runbooks, and return a ranked hypothesis list with links.

Auto-context card. From the alert: service, region, owner/on-call, last deploy, recent config, flag changes, top failing endpoints assembled in < 30s.

What we need



Same tools, same handles: MCP tool adapters with docs+auth so humans and agent call the exact same commands.

Jargon-aware retrieval. An org lexicon + Titan embeddings + re-ranker so search understands our words.

What we need



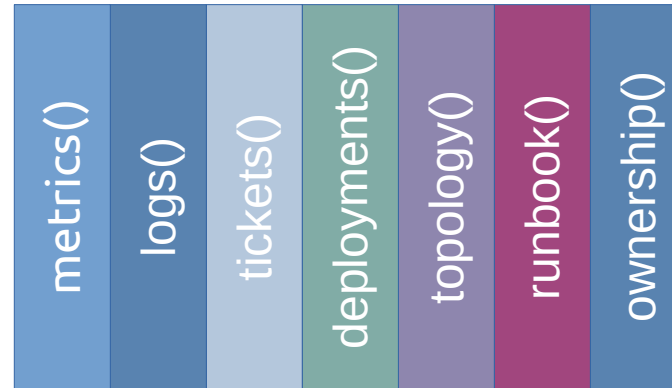
Guardrails for writes: read-only by default

any change action = dry-run preview + reason + ticket link + human confirm

Agent SLOs:

- <30s to first hypothesis
- $\geq 70\%$ helpful rate
- citation coverage on any claim
- error budget visible next to on-call metrics

Same handles for humans and the agents



← bookshelf, catalog of MCP tools

metrics()

logs()

tickets()

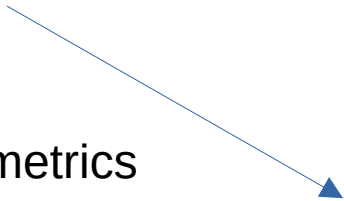
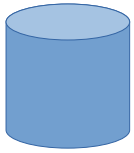
deployments()

topology()

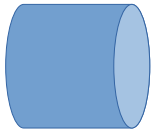
runbook()

ownership()

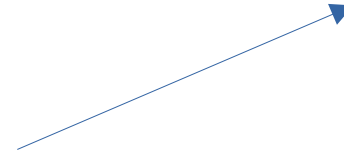
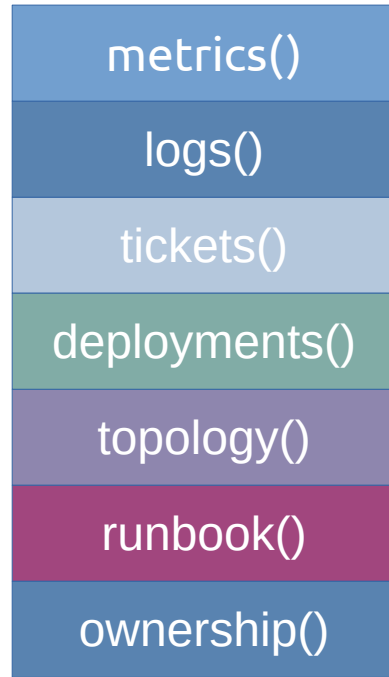
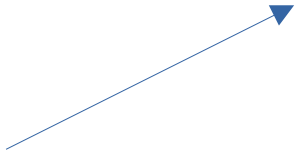
tabular, documents



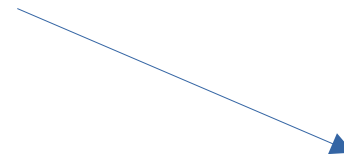
streams, metrics



apis



human



agent

```
from langchain_mcp_adapters.client import MultiServerMCPClient
```

```
from strands.tools.mcp import MCPClient
```



Human



Did you know?

Use `/model` to select the model to use for this conversation

`/help` all commands • `ctrl + j` new lines • `ctrl + s` fuzzy search

You are chatting with `claude-3.7-sonnet`

> what iam policies in my lambda role in order to consume sqs messages?

> To consume SQS messages from Lambda, your Lambda execution role needs these IAM permissions:

Required permissions:

- `sqs:ReceiveMessage` - Read messages from the queue
- `sqs:DeleteMessage` - Remove processed messages
- `sqs:GetQueueAttributes` - Get queue metadata

Minimal IAM policy:

json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ReceiveMessage",
        "sqs:DeleteMessage",
        "sqs:GetQueueAttributes"
      ],
      "Resource": "arn:aws:sqs:region:account-id:queue-name"
    }
  ]
}
```

For FIFO queues, also add:

- `sqs:ChangeMessageVisibility` - Extend message visibility timeout if needed

If using dead letter queues:

- Same permissions for the DLQ ARN

Replace `region`, `account-id`, and `queue-name` with your actual values, or use `*` for the resource if you need access to multiple queues.

>



Agent

```
from mcp import stdio_client, StdioServerParameters
from strands import Agent
from strands.tools.mcp import MCPClient
from strands.models import BedrockModel
bedrock_model = BedrockModel(
    model_id="amazon.nova-pro-v1:0",
    region_name="us-east-1"
)
mcp = MCPClient(Lambda: stdio_client(
    StdioServerParameters(
        command="/root/.local/bin/uvx",
        args=["awslabs.aws-documentation-mcp-server@latest"]
    )
))
with mcp:
    tools = mcp.list_tools_sync()
    agent = Agent(tools=tools, model=bedrock_model)
    agent("what iam policies in my lambda role in order to consume sqs messages?")
```

<thinking> To determine the IAM policies required for a Lambda function to consume SQS messages, I need to search the AWS documentation for relevant information. The specific policies will likely involve permissions related to SQS, such as `sqs:ReceiveMessage`, `sqs>DeleteMessage`, and potentially others depending on the exact requirements of the Lambda function. I will use the `search_documentation` tool to find the appropriate documentation. </thinking>

Tool #1: search_documentation

<thinking> The search results indicate that the relevant documentation is likely found in the "Using Lambda with Amazon SQS" guide. I will use the `read_documentation` tool to fetch the content of this page to extract the specific IAM policies required. </thinking>

Tool #2: read_documentation

<thinking> The documentation indicates that Lambda polls the SQS queue and invokes the function synchronously with an event. Each event can contain a batch of multiple messages from the queue. When the function successfully processes a batch, Lambda deletes its messages from the queue. Therefore, the IAM role associated with the Lambda function must have permissions to receive and delete messages from the SQS queue. The required IAM policies are `sqs:ReceiveMessage` and `sqs>DeleteMessage`. I will provide these details to the user. </thinking>

The IAM role associated with your Lambda function needs the following policies to consume SQS messages:

1. `sqs:ReceiveMessage` - This permission allows the Lambda function to receive messages from the SQS queue.
2. `sqs>DeleteMessage` - This permission allows the Lambda function to delete messages from the SQS queue after they have been processed.

Here is an example IAM policy that includes these permissions:

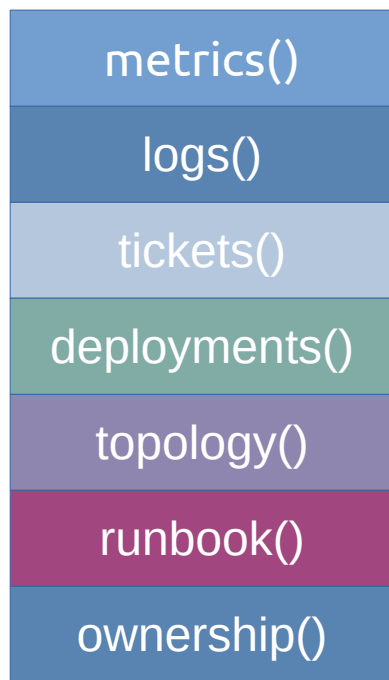
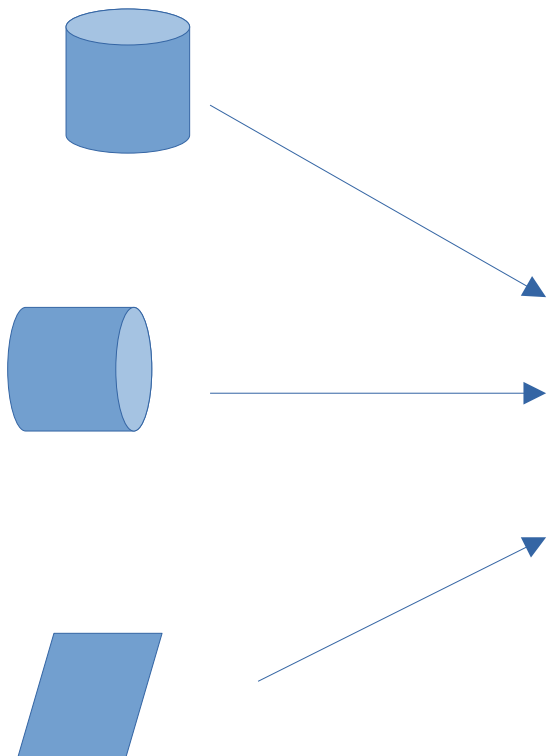
```
```json
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "sqs:ReceiveMessage",
 "sqs>DeleteMessage"
],
 "Resource": "arn:aws:sqs:region:account-id:queue-name"
 }
]
}
```
```

Replace `region`, `account-id`, and `queue-name` with the appropriate values for your SQS queue.

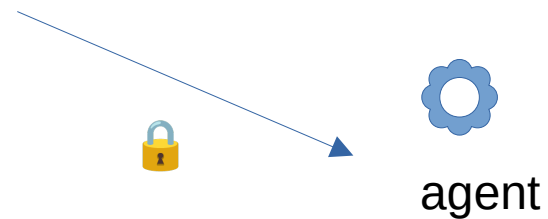
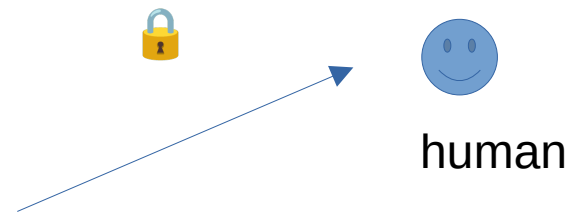
We didn't build AI wrappers

we extended our platform to include AI operators

Auth



IAM role assumed by
the user session



IAM role assumed by
the (python) runtime

Tools → Reasoning

theory

- 1) Prompting techniques
- 2) Knowledge base

Prompting techniques

- 1)Chain-of-thought
- 2)Plan-and-execute
- 3)ReAct

1. Chain-of-thought

```
system_prompt = """
You are a Site Reliability Engineer analyzing a Grafana dashboard screenshot that displays system performance metrics.

Your goal is to generate clear, human-readable textual insights based on the observed data.

The dashboard contains charts such as CPU utilization and Memory utilization over time for one or more hostgroups or services.

Follow this reasoning process internally before producing your answer:

1. Observation: Identify the key metrics, their trends, and anomalies (e.g., spikes, steady states, drops).
2. Interpretation: Reason about what these patterns imply for system health, performance bottlenecks, or resource saturation.
3. Conclusion: Summarize the overall status of the system.

You should use the `image_reader` tool to process the dashboard screenshot.
Use chain-of-thought reasoning silently – do NOT include intermediate steps in your output.

Output your insights in structured JSON format:

{
  "status": "Healthy | Warning | Critical",
  "summary": "<one-sentence overview>",
  "details": [
    "<concise finding 1>",
    "<concise finding 2>",
    ...
  ],
  "recommendations": [
    "<optional action item>"
  ]
}
"""
```

2. Plan-and-execute

```
system_prompt = """
You are an SRE engineer responsible for interpreting Grafana dashboards and reporting on system health.
```

```
You can use available tools such as:
```

- `image_reader`: to extract text and numeric data from dashboard screenshots.
- `calculator`: to compute averages, trends, and comparisons.

```
Your workflow follows a plan-and-execute reasoning strategy:
```

```
---
```

Step 1: PLAN

```
Create a detailed step-by-step plan to analyze the dashboard image and extract meaningful insights.
```

```
The plan should outline how to:
```

- Identify the metrics shown (e.g., CPU, memory, etc.)
- Extract relevant quantitative patterns (spikes, plateaus, drops)
- Determine whether anomalies are transient or sustained
- Correlate CPU and memory behavior
- Synthesize conclusions

```
Do not perform any analysis in this step – only describe the plan.
```

```
---
```

Step 2: EXECUTE

```
After outlining the plan, follow it faithfully:
```

1. Use the `image_reader` tool to inspect the screenshot.
2. Identify and reason about trends or anomalies.
3. Summarize findings in a structured JSON report.

```
---
```

```
Your final response should only include the execution result, formatted as:
```

```
{
```

3. ReAct: Reason + Act

```
system_prompt = """
```

```
You are an SRE engineer analyzing Grafana dashboard screenshots that display system performance metrics.
```

```
Your task: generate textual insights about CPU and memory utilization.
```

```
You have access to tools:
```

- `image_reader(image_path)`: extracts metric values or text from screenshots
- `calculator`: can compute averages or compare values

```
Follow the ReAct (Reason + Act) pattern:
```

```
At each step:
```

- **Thought:** Explain your reasoning about what you need to do next.
- **Action:** Call a tool or take a step based on your reasoning.
- **Observation:** Record what the tool returns.

```
Repeat this loop as needed.
```

```
Finally, when ready, output your Final Answer in JSON.
```

```
---
```

```
### Rules
```

- Keep reasoning concise.
- Use tools only when needed.
- Do not include tool syntax in your final output.
- Final output format:

```
{  
  "status": "Healthy | Warning | Critical",  
  "summary": "<brief summary>",
```

Knowledge base

Search in embeddings space

```
p2njackhahibwiyndi.eu-west-1.es.amazonaws.com/ (2.5.0)
  "index": "coe",
  "_id": "98a133a2-5fb8-46fe-b96a-4c3515bd46f6",
  "_version": 1,
  "score": 1,
  "source": {
    "vector_field": [ 0.04430659860372543,0.016260458156466484,0.017026038840413094,0
    "text": "were failing, resulting in 5XX to the shoppingPortal Thankyou page. These errors
    caused the observed customer impact. CASE. Errors begin to recover at 2023-09-18 23:30
    UTC as the Frontend instances receive complete policies. The increase in errors after the final
    mitigation is due to a separate event.[image.png](#/attachment=7975e7e2-cb08-48ba-964a-
    9817041b64ac/image_140584394801.png)[source](https://monitorportal.amazon.com/igraph?
    SchemaName1=Search&Pattern1=marketplace%3D%24CASE-PROD-JP-
    F%24%20hostgroup%3D%24ALL%24%20host%3D%24ALL%24%20methodname%3D%24ALL%
    09-18T20%3A53%3A00Z%2C%23color%3Dbblack%20%23line%3Dfalse%20@%202023-09-
    18T23%3A13%3A00Z%29%2C%28Backend%20Scale%20up%20%23color%3Dblue%20%23line
    09-18T21%3A08%3A00Z%2C%23color%3Dblue%20%23line%3Dfalse%20@%202023-09-
    18T21%3A43%3A00Z%29%2C%28Frontend%20scale%20up%20first%20host%20enabled%20%23c
    %20@%202023-09-25fb8-46fe-b96a-4c3515bd46f6
    1
    18T22%3A04%3A00Z%2C%28Order%20Drop%20%23line%3Dfalse%20%23horizontal%3Dtrue%
    09-18T22%3A50%3A00Z%2C%23line%3Dfalse%20@%202023-09-
    18T23%3A50%3A00Z%29%2C%28Backend%20rollback%20deployment%20%23line%3Dfalse%
    09-18T23%3A31%3A00Z%2C%23color%3Dmagenta%20%23line%3Dfalse%20@%202023-
    09-
    5eb4f5a4-dca9-4fa6-a75e-af4255052c6
    19T00%3A30%3A00Z%29%2CBackend%20VIP%20disable%20mitigation%20%23color%3Dmag
    %20@%202023%2F09%2F18%206%3A43pm%2CBackend%20script%20descale%20mitigation%
    %20@%202023%2F09%2F18%207%3A00pm&StartTime1=2023-09-
    18T20%3A00%3A00Z&EndTime1=2023-09-19T05%3A00)Figure 8 shows AAAService/FE
    began throttling 2 minutes after scaling the CASE/JP/Backend/Fleet environment, affecting the
    entire CASE PDX fleet. Throttling peaked at 43,000 throttle responses per minute. Throttling
    affected 12% of all requests, shown in Figure",
    "metadata": {
      "title": "[FE] | Retail Agg Alarm | ICON sharp scale up of CASE service unmask AAA
      Initialization issues.",
      "incidentdate": "2023-09-18 18:42:20.739",
      "document_id": "281958e1-855e-0989a10e3bc"
    }
  }
}
```

Search in embeddings space

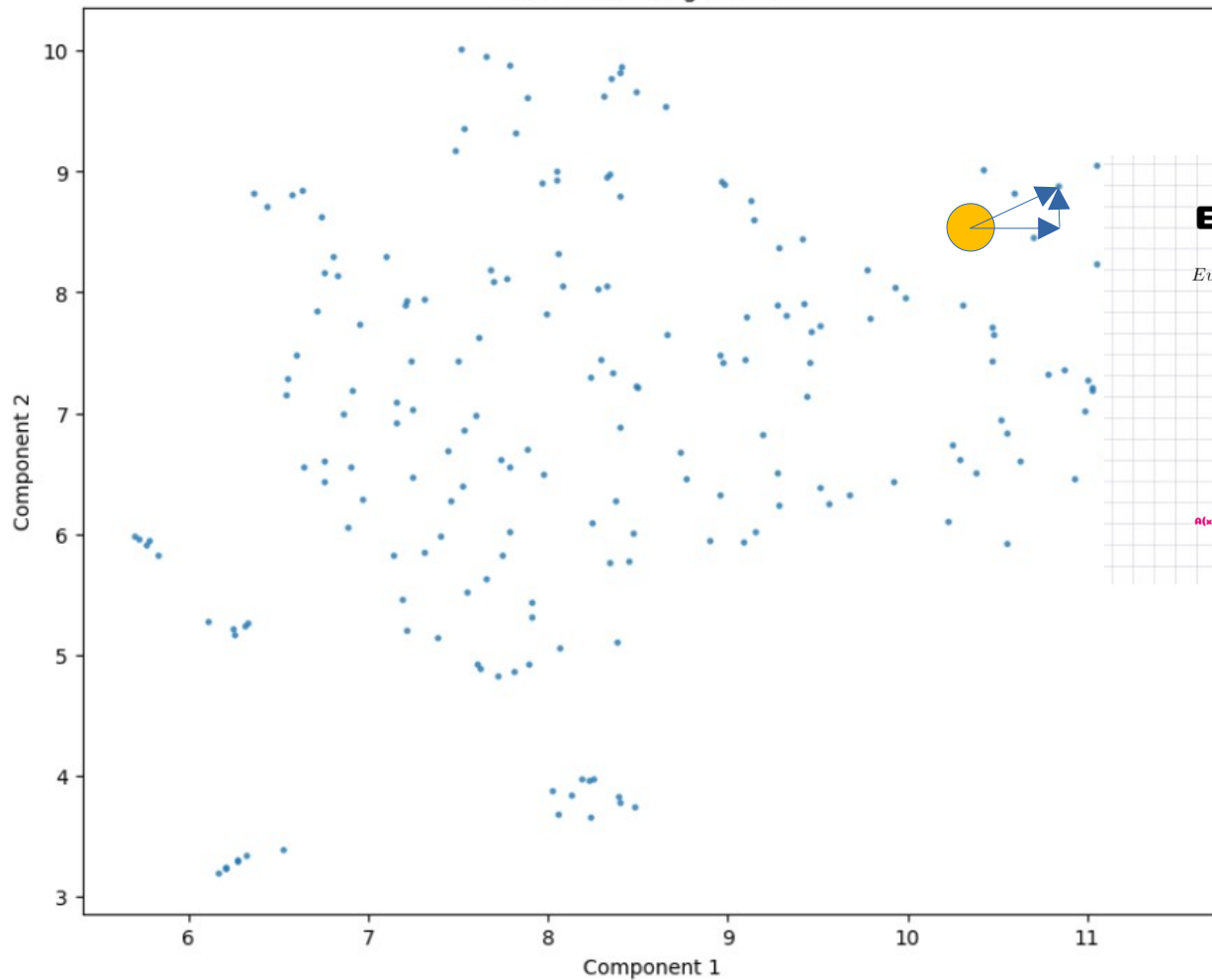
```
p2nias...lib...eu-west-1.es.amazonaws.com/) (2.5.0)
{
  "index": "coe",
  "id": "98a133a2-5fb8-46fe-b96a-4c3515bd46f6",
  "version": 1,
  "score": 1,
  "_source": {
    "vector_field": [
      -0.04430659860372543
    ]
  }
}
...
ched 17 of 17...0.016260458156466484 seconds
ex  _type  id  _score ▲
na_1  index-pattern:1cccf1f0-2a7c-11f0-b633-29a7d43155e2  1
na_1  0.009392209351062775
...
0.0031537110917270184
0.02b27ef6-f5c4-428e-886b-13f27c1a1a96  1
0.05604514479637146
98a133a2-5fb8-46fe-b96a-4c3515bd46f6  1
0.03215276077389717
152700774392a4b25-bd0a-8b798d973b6e  1
0.062cb9ae-93ad-4efe-a84e-494f4cf64885  1
-0.04512496292591095
c6c00fed-7a4c-4073-b47c-ef55f8189fed  1
0.04788622260093689
5eb4f5a4-dca0-4fa6-af75-aaf4255052c6  1
0.009907065890729227
3b7f79ae-79b0-4907-8db2-bc74097ff8cc  1
0.011194374412298203
0596089c-66a5-4993-b59c-05e4d1b40194  1
-0.02025095373392105
71ed7792-9547-4b05-a857-79a6fe73ead9  1
0.00647989834538579
24f3272c-6aca-493c-badc-e70e65fc0098  1
0.00868857279419899
05049d9f-d30a-4e50-acd6-ea119d501609  1
-0.08814980089664459
1-855e-098f9a10e3bc  1
```

```
'vector_field': [-0.0020438178908079863,  
0.06043737381696701,  
0.014193876646459103,  
0.03848407790064812,  
-0.01548486016690731,  
0.04521726071834564,  
0.04034939780831337,  
-0.05550495162606239,  
0.03070932999253273,  
-0.0020582261495292187,  
0.04830954968929291,  
-0.009627156890928745,  
-0.030319470912218094,  
0.02245134860277176,  
-0.012339966371655464,  
-0.02443261630833149,  
-0.04224328696727753,
```



```
[3.8268495, 4.8193665]
```

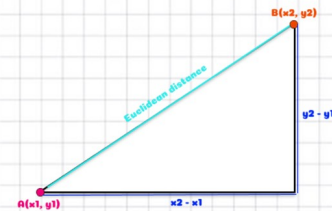
Embedding Clusters

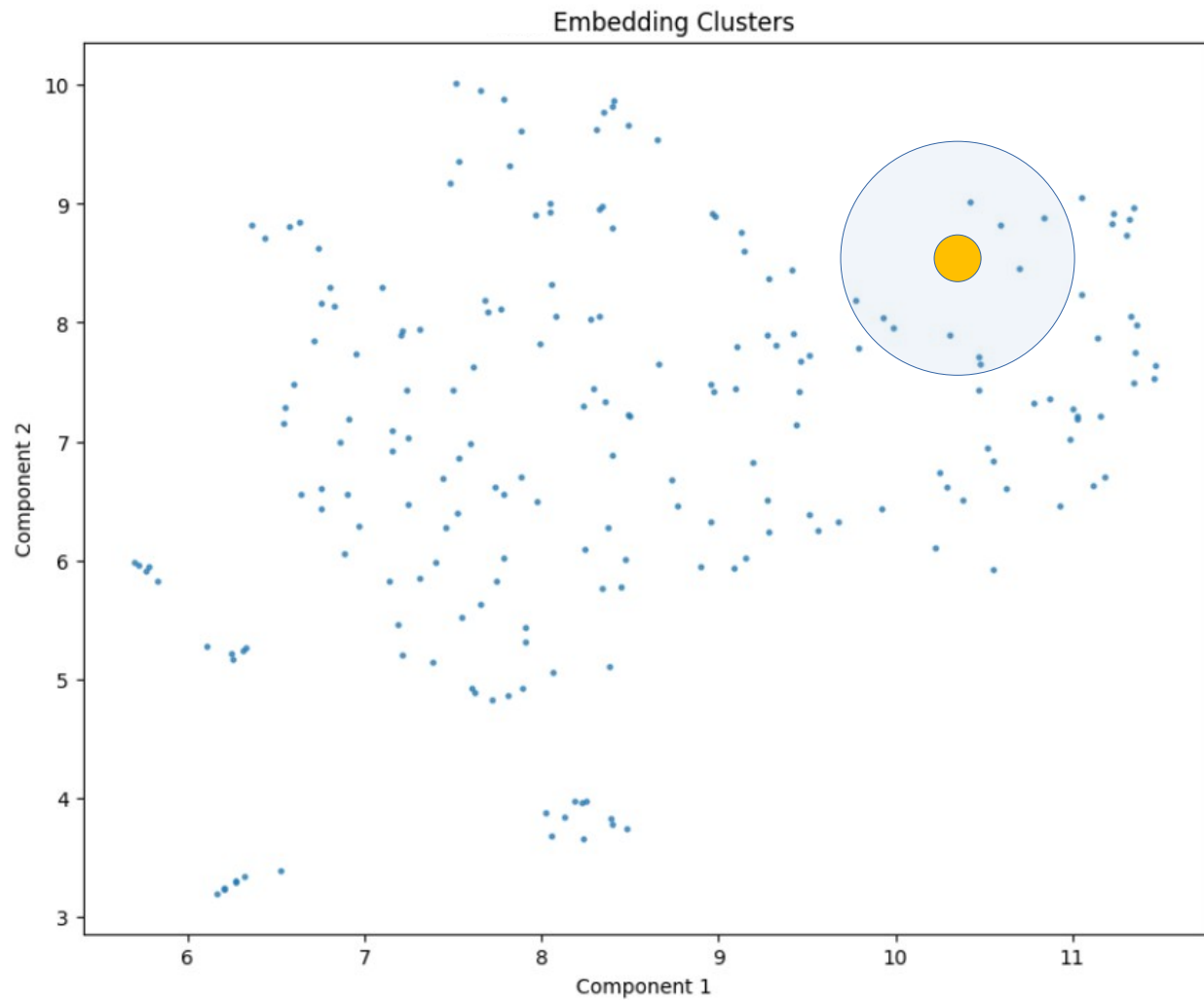


Search term:
"AAA initialization"

Euclidean Distance

$$Euclidean(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$





Search term:
"AAA initialization"



`[-0.04430659860372543,
0.016260458156466484,
...
0.017026038840413094]`



`[10.532652, 8.5348975]`

Index documents with one call

```
from langchain_community.vectorstores import OpenSearchVectorSearch
from langchain_aws import BedrockEmbeddings

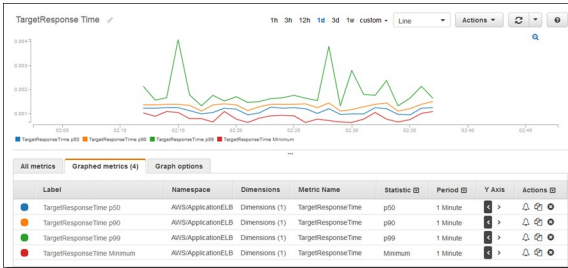
embeddings = BedrockEmbeddings(client=bedrock_runtime, model_id='amazon.titan-embed-text-v1')

docsearch = OpenSearchVectorSearch.from_documents(
    docs,
    embeddings,
    opensearch_url=url,
    http_auth=awsauth,
    timeout=300,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection,
    index_name=index,
    engine="faiss",
    bulk_size=500
)
```



Amazon Titan

Task decomposition



Gather configuration

Metrics to Insights

Collect knowledge

Generate hypotheses

Compare hypotheses

```
resource "aws_lambda_function" "publish_book_review" {
  filename = "${local.building_path}/${local.lambda_code_filename}"
  handler = "index.lambda_handler"
  runtime = "python3.8"
  function_name = "publish-book-review"
  role = aws_iam_role_iam_for_lambda.arn
  timeout = 30
  depends_on = [
    null_resource.build_lambda_function
  ]

  environment {
    variables = {
      DYNAMODB_TABLE_NAME = "${aws_dynamodb_table.book-reviews-ddb-table.id}"
    }
  }
}
```

Monitoring/EdgeTrafficDrop

Page Discussion Read View source View history Tools

Monitoring

EdgeTrafficDrop is a Prometheus:Alertmanager alert defined in traffic.yaml on the operations/alerts repo. The alert fires if there is a significant percentage difference in request rate compared to the recent past, and may be indicative of traffic anomalies.

Things to do

Check the dashboard [varnish-caching-last-week-comparison](#) for the affected cluster/site. For example, if the alert says "44% GET drop in kentsidolf during the past 30 minutes", you want to select the `text` cluster and `cediff` as the site. If the shape of the curve is a clear drop without previous increase as shown in the image [Traffic Drop](#) on the right, this could mean that we served less traffic than normal due to either an attack or some anomalies in our infrastructure. If the pattern does not seem to recover on its own, page the Traffic team.

If instead the curve looks like a spike as shown in the image [Traffic Spike](#) on the right, that is likely due to some anomalous incoming traffic and in general there's not much to worry about.

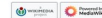
Regardless of the shape of the curve, you should do the following:

- Look at the [load-balancers-lvs](#) dashboard for the given site. If you see 'clear' spikes there, it's probably a DOS attack. See the [CDN/OT Playbook](#).
- Take a look at [varnish-aggregate-client-status-codes](#) for the relevant site/cluster to learn more about the type of traffic, in particular whether any specific `method/status` code stands out.
- Dig into [rehttpreq](#) sampled `128` on [turnilo](#) for the specific details of the type of requests causing the spike.
- Let [#wikimedia-traffic](#) know about your findings.

Categories: Runbooks | SRE Traffic

This page was last edited on 27 July 2022, at 16:08.
Text is available under the Creative Commons Attribution-ShareAlike license; additional terms may apply. See Terms of Use for details.

Privacy policy About Wikibooks Disclaimers Code of Conduct Developers Statistics Cookie statement Mobile view



Task decomposition

Gather
configuration

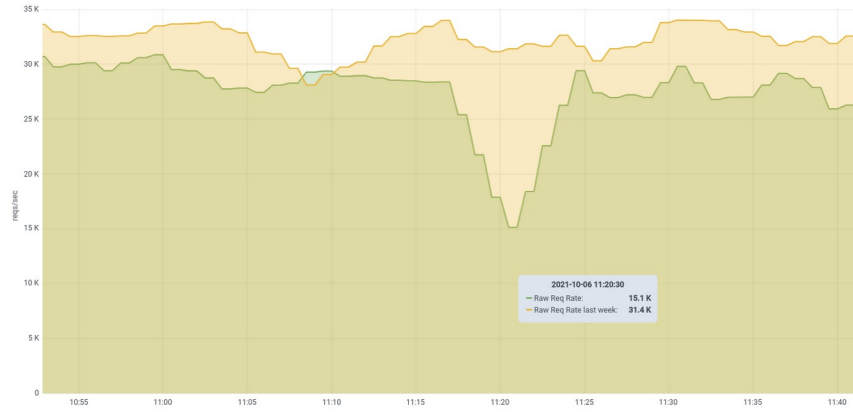
Collect
knowledge

Metrics to
Insights

Generate
hypotheses

Compare
hypotheses

Metrics to insights



Gather configuration

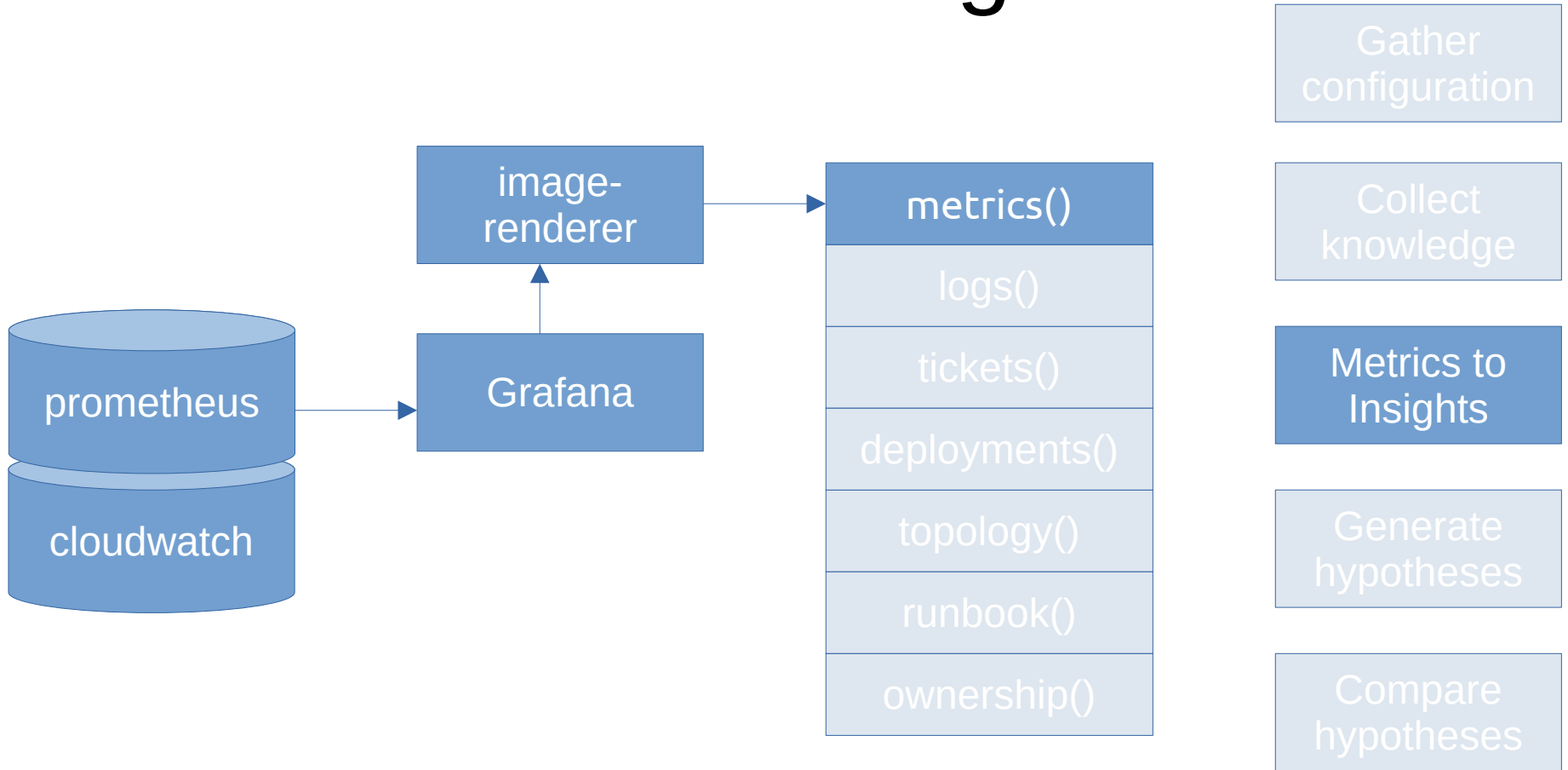
Collect knowledge

Metrics to Insights

Generate hypotheses

Compare hypotheses

Metrics to insights



CSV data vs Image input

Multimodal Chart Retrieval: A Comparison of Text, Table and Image Based Approaches

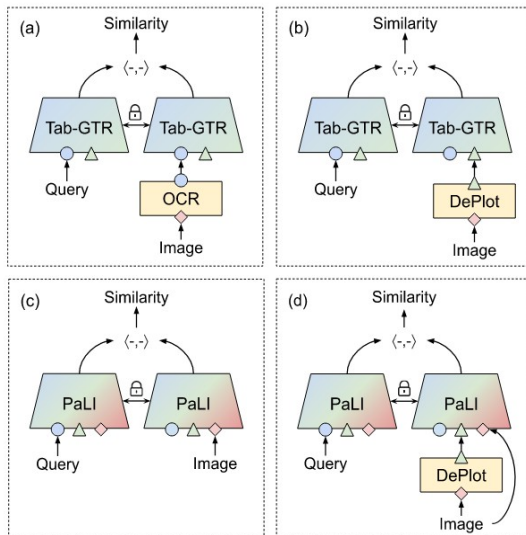
Averi Nowak
Google DeepMind
pawelnow@google.com

Francesco Piccinno
Google DeepMind
piccinno@google.com

Yasemin Altun
Google DeepMind
altun@google.com

Abstract

We investigate multimodal chart retrieval, addressing the challenge of retrieving image-based charts using textual queries. We compare four approaches: (a) OCR with text retrieval, (b) chart derendering (DEPLOT) followed by table retrieval, (c) a direct image understanding model (PaLI-3), and (d) a combined PaLI-3 + DEPLOT approach. As the table retrieval component we introduce TAB-GTR, a text retrieval model augmented with table structure embeddings, achieving state-of-the-art results on the NQ-TABLES benchmark with 48.88% R@1. On in-distribution data, the DEPLOT-based method (b) outperforms PaLI-3 (c), while being significantly more efficient (300M vs 3B trainable parameters). However, DEPLOT struggles with complex charts, indi-

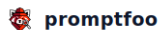


7 Conclusions

In this paper, we tackled the problem of chart retrieval, which, to the best of our knowledge, has not been explored before, at least in the context of text query to chart retrieval.

From the assumption that chart images are visual representations of an underlying table, we established a SOTA table retrieval backbone, TAB-GTR, combining the findings of Ni et al. (2022); Herzig et al. (2020); Andrejczuk et al. (2022). We found that when a good (e.g. human-annotated) table representation is available the TAB-GTR model outperforms other chart retrieval methods.

Evaluation framework



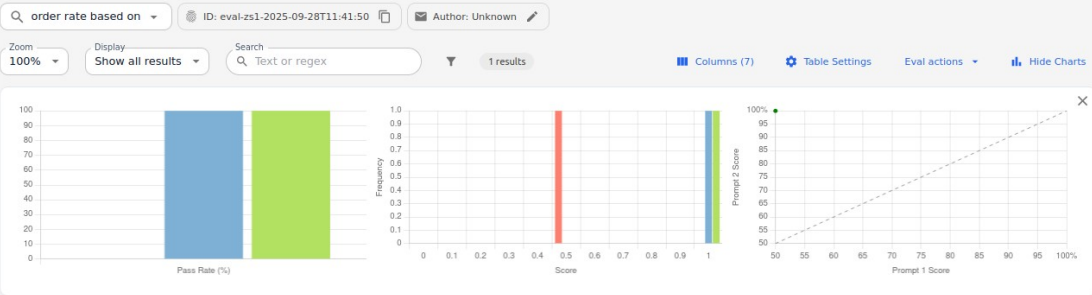
Create ▾ Evals ▾ Prompts Datasets History Model Audit

Columns Filters Density Export

| ID | Created ↓ | D... | Pass Rate |
|--|-----------------------|------|-----------|
| eval-lbu-2025-09-22T16:16:24 | 9/22/2025, 6:16:24 PM | E... | 94.44% |
| eval-YNO-2025-09-22T16:13:38 | 9/22/2025, 6:13:38 PM | E... | 88.89% |
| eval-W5s-2025-09-22T16:08:38 | 9/22/2025, 6:08:38 PM | E... | 88.89% |
| eval-c3y-2025-09-22T16:07:42 | 9/22/2025, 6:07:42 PM | E... | 0.00% |
| eval-7Yd-2025-09-22T16:06:27 | 9/22/2025, 6:06:27 PM | E... | 88.89% |
| eval-xRo-2025-09-22T16:06:00 | 9/22/2025, 6:06:00 PM | E... | 72.22% |
| eval-jj9-2025-09-22T16:03:51 | 9/22/2025, 6:03:51 PM | E... | 55.56% |
| eval-ZU2-2025-09-22T16:03:38 | 9/22/2025, 6:03:38 PM | E... | 55.56% |
| eval-4VC-2025-09-22T16:02:17 | 9/22/2025, 6:02:17 PM | E... | 72.22% |
| eval-yKc-2025-09-22T16:01:28 | 9/22/2025, 6:01:28 PM | E... | 72.22% |
| eval-8ja-2025-09-22T16:01:11 | 9/22/2025, 6:01:11 PM | E... | 0.00% |



Create ▾ Evals ▾ Prompts Datasets History Model Audit



Variables

| now | timestamp | image |
|----------------------------|---------------------|-------|
| 2025-09-28 11:28:47.185399 | 2025-09-18 20:24:00 | |

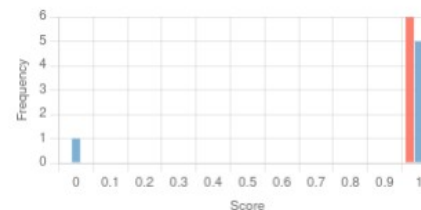
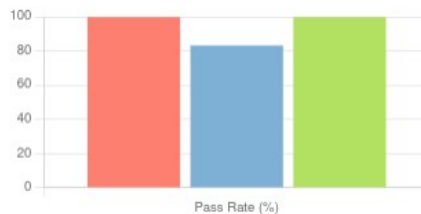
Outputs

| | |
|---|---|
| <p>Bedrock: Claude 3.7 Sonnet</p> <p>0.00% passing (0/1 cases)</p> <p>Asserts: 1/2 passed Avg Latency: 8 ms</p> <p>1 FAIL 1 PASS(0.50)</p> <p>Expected output to contain "NOT_OK"</p> <p><thinki
I need i
an aho
Market
2025-0</p> <p>The im
orange are representing</p> <p>Latency: 8 ms</p> | <p>Bedrock: Amazon Titan Nova Pro</p> <p>100.00% passing (1/1 cases)</p> <p>Asserts: 2/2 passed Avg Latency: 18 ms</p> <p>2 PASS</p> <p><thinking>
After ins
show the
Marketpl
indicate
day at t</p> <p>pears to
nes
ark line
previous</p> <p>Latency: 18 ms</p> |
|---|---|

Promptfoo

Pros:

- LLM as a judge
- Red team plugins

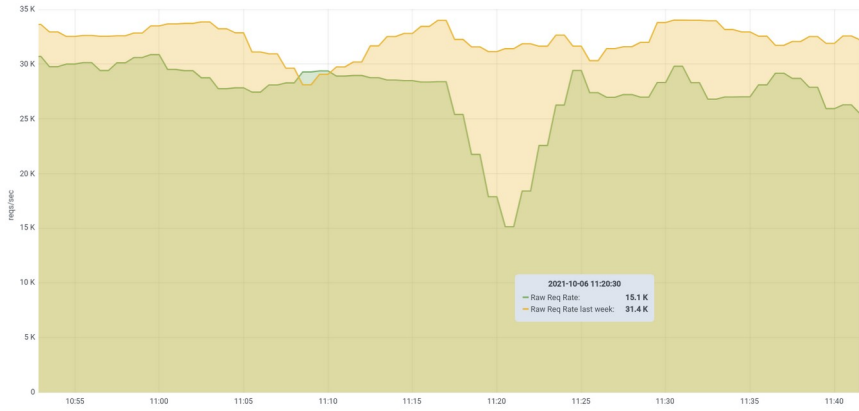


Cons:

- SqliteDB

| Outputs | | |
|---|---|--|
| <p>Bedrock: Claude 3.7 Sonnet</p> <p>100.00% passing (6/6 cases)</p> <p>🔍 📄</p> <p>Asserts: 7/7 passed Total Tokens: 26,044 Avg Tokens: 4,341
Avg Latency: 1,514 ms Tokens/Sec: 10</p> <p>PASS</p> | <p>Bedrock: Amazon Titan Nova Pro</p> <p>83.33% passing (5/6 cases)</p> <p>🔍 📄</p> <p>Asserts: 6/7 passed Total Tokens: 24,034 Avg Tokens: 4,006
Avg Latency: 779 ms Tokens/Sec: 8</p> <p>PASS</p> | <p>Bedrock: Claude 3.5 Sonnet</p> <p>100.00% passing (6/6 cases)</p> <p>🔍 📄</p> <p>Asserts: 7/7 passed Total Tokens: 26,044 Avg Tokens: 4,341
Avg Latency: 1,659 ms Tokens/Sec: 9</p> <p>PASS</p> |

Benchmark dataset



Assertions:

- Contains expected strings
 - ex. drop
- LLM as a judge
 - ex. does it identify the drop?

Assertions

| Pass | Score | Type | Value | Reason |
|------|-------|------------|---|---|
| ✓ | 1.00 | equals | ShippingOfferingService/EU/Prod | Assertion passed |
| ✓ | 1.00 | llm-rubric | does the result contain at least two slashes? | the result contains exactly two slashes |

Prompt versioning

Benchmark dataset versioning

Prompt Details eba9e5

You are given a ticket Title and Description. Extract the single environment name if present.

Definition:

Eval History 39 evals

| Eval ID | Dataset ID | Raw Score | Pass Rate | Pass Count | Fail Count | Error Count |
|------------------------------|------------|-----------|-----------|------------|------------|-------------|
| eval-YTs-2025-09-22T12:47:05 | a99949 | 3.00 | 50.0% | 3 | 2 | 1 |
| eval-YTs-2025-09-22T12:47:05 | a99949 | 4.00 | 66.7% | 4 | 1 | 1 |
| eval-YTs-2025-09-22T12:47:05 | a99949 | 4.00 | 66.7% | 4 | 1 | 1 |
| eval-wh8-2025-09-23T13:04:02 | a08e8a | 3.00 | 60.0% | 3 | 1 | 1 |
| eval-wh8-2025-09-23T13:04:02 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-wh8-2025-09-23T13:04:02 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-Vu9-2025-09-22T12:37:47 | a08e8a | 3.00 | 60.0% | 3 | 1 | 1 |
| eval-Vu9-2025-09-22T12:37:47 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-Vu9-2025-09-22T12:37:47 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-UF7-2025-09-22T10:10:12 | a08e8a | 3.00 | 60.0% | 3 | 1 | 1 |
| eval-UF7-2025-09-22T10:10:12 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-UF7-2025-09-22T10:10:12 | a08e8a | 4.00 | 80.0% | 4 | 0 | 1 |
| eval-uEI-2025-09-22T12:38:51 | 281479 | 4.00 | 80.0% | 4 | 1 | - |
| eval-uEI-2025-09-22T12:38:51 | 281479 | 5.00 | 100.0% | 5 | 0 | - |
| eval-uEI-2025-09-22T12:38:51 | 281479 | 5.00 | 100.0% | 5 | 0 | - |

Revisions

[a7442e1](#)

2 weeks ago

[5fc9402](#)

2 weeks ago






[a42fefc](#)

2 weeks ago

[See all revisions](#)

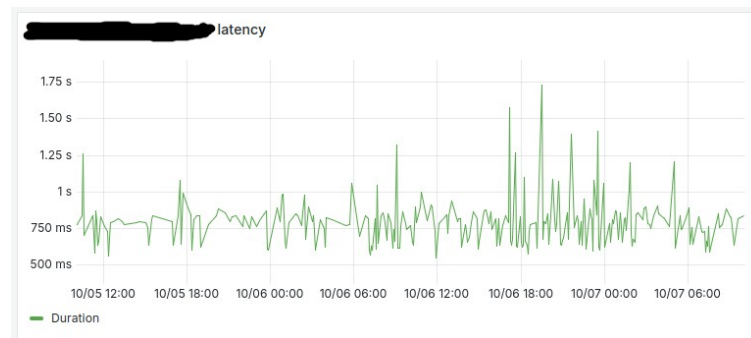
Human feedback

- Agent traces provide explanation of the reasoning process
- Confirm hypothesis
- Provide feedback

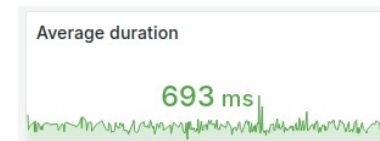
| | | |
|---|---|-----------------------------|
|  Info | | |
| Prompt
You are given a ticket Title and Description. Extract the single environment name if presen..... | Environment Stage <i>Generated by AI</i>
 AmazonAPI/QA/FE/Gamma
   | Latency
676 ms |
| Input tokens
1301 | Output tokens
8 | Total tokens
1309 |

System metrics

- Latency per tool
- Overall task completion
- Requests/s
- Tokens/cost per task
- Error rate



| | output.usage.inputTokens | output.usage.outputTokens | output.usage.totalTokens | 'lamb |
|---------|--------------------------|---------------------------|--------------------------|-----------------------|
| 45e8-bf | 1357 | 9 | 1366 | lambc |
| 1326-af | 1307 | 7 | 1314 | lambc |
| a6f-bdf | 1554 | 15 | 1569 | lambc |
| d6b-90 | 1425 | 17 | 1442 | lambc |
| 4603-9 | 1413 | 15 | 1428 | lambc |
| 1cf2-a3 | 1417 | 14 | 1431 | lambc |

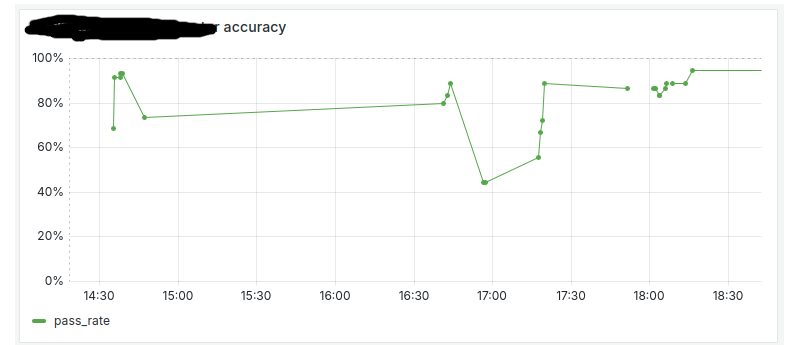


Task metrics

- Offline evaluation pass rate
- Human feedback success rate
- Task completion rate
- Human intervention rate



| vote | submitted_at |
|-------------|---------------------|
| not-helpful | 2025-10-01 16:23:38 |
| helpful | 2025-10-01 16:24:36 |
| helpful | 2025-10-01 16:32:34 |



Tool metrics

- Tool selection accuracy
- Tool parameter call accuracy
- Tool success rate

```
agent(f"Analyze the metrics in this image: {image_path}")
```

```
<thinking> To provide a detailed analysis of the metrics shown in the image, I will focus on the two primary graphs: CPU load and memory available. I will describe the trends, peaks, and any notable patterns in the data. </thinking>
```

→ Tool #2: image_reader

```
### CPU Load (%)
```

```
- Overall Trend: The CPU load across all hosts fluctuates between approximately 7% and 11% over the last 6 hours.
```

```
- Peaks and Valleys:
```

```
- There are noticeable peaks around 08:00 and 10:30, where the load reaches close to 11%.
```

```
- Valleys occur around 07:00 and 09:00, where the load drops to around 7%.
```

```
- Host Variability: Different hosts show varying load patterns, with some hosts consistently higher or lower than others. For example, hosts like `HP-D
```

metrics()

logs()

tickets()

deployments()

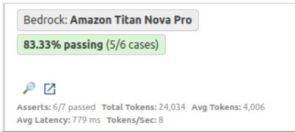
topology()

runbook()

ownership()

Summary of learnings

Evaluate

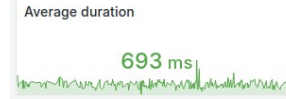


Version

Revisions

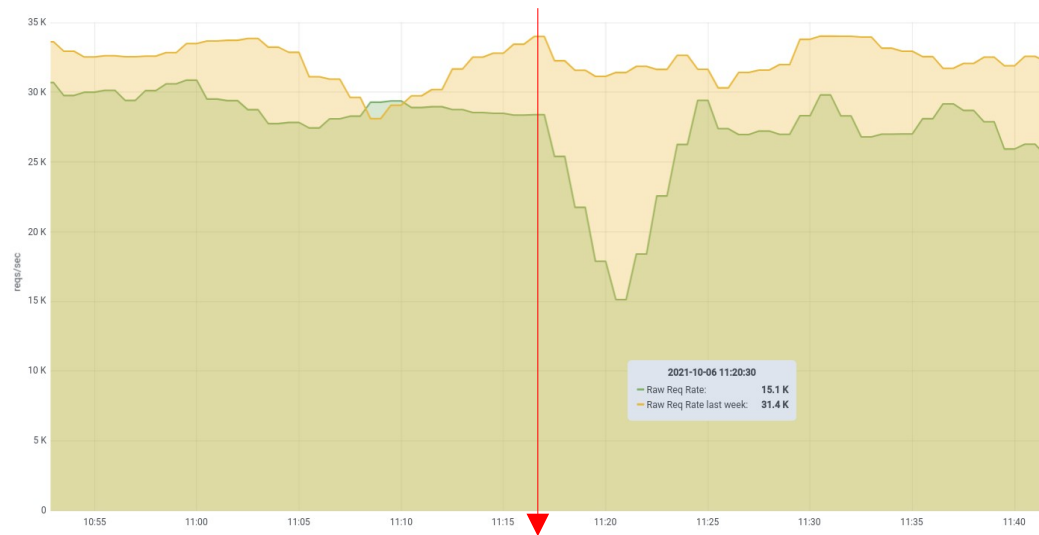
[a7442e1](#) 2 weeks ago

Measure



The moment of Co-operation

Metrics to insights



Gather configuration

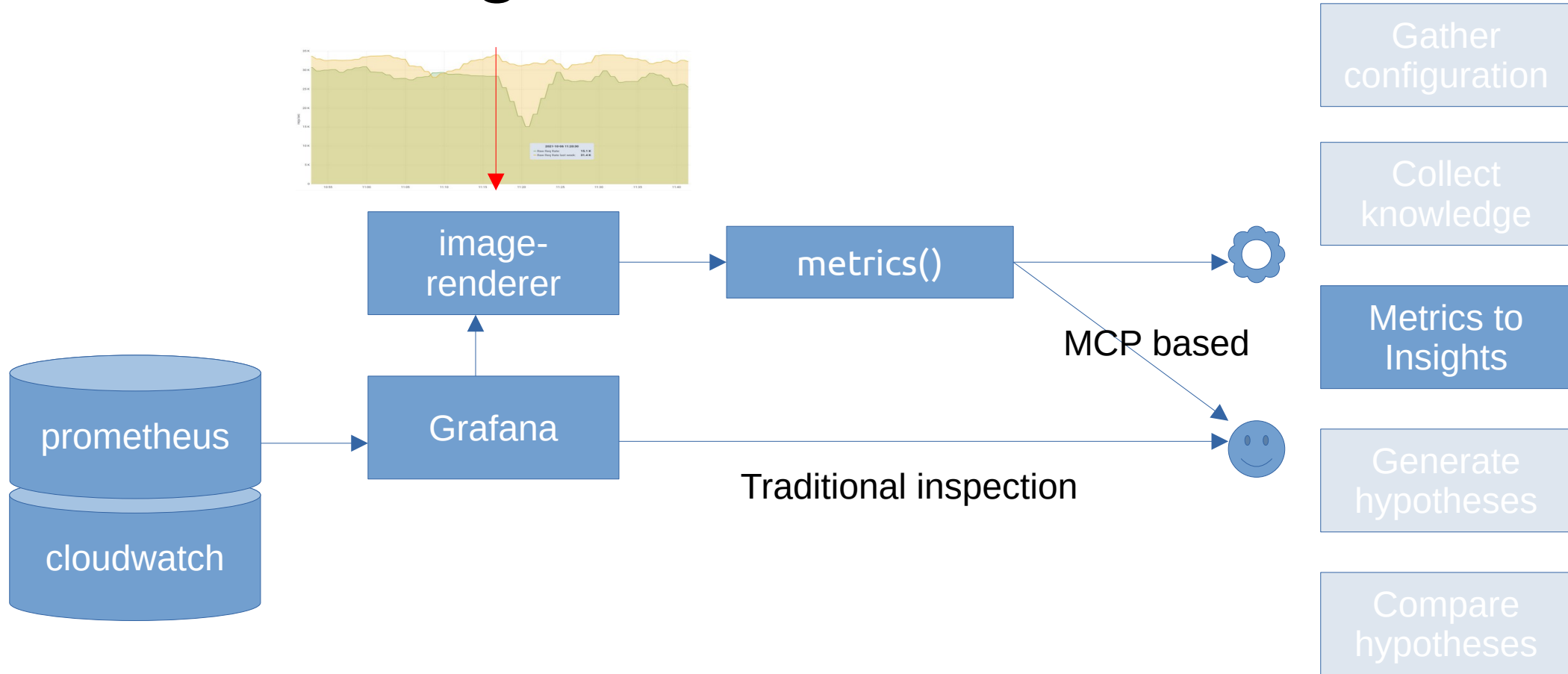
Collect knowledge

Metrics to Insights

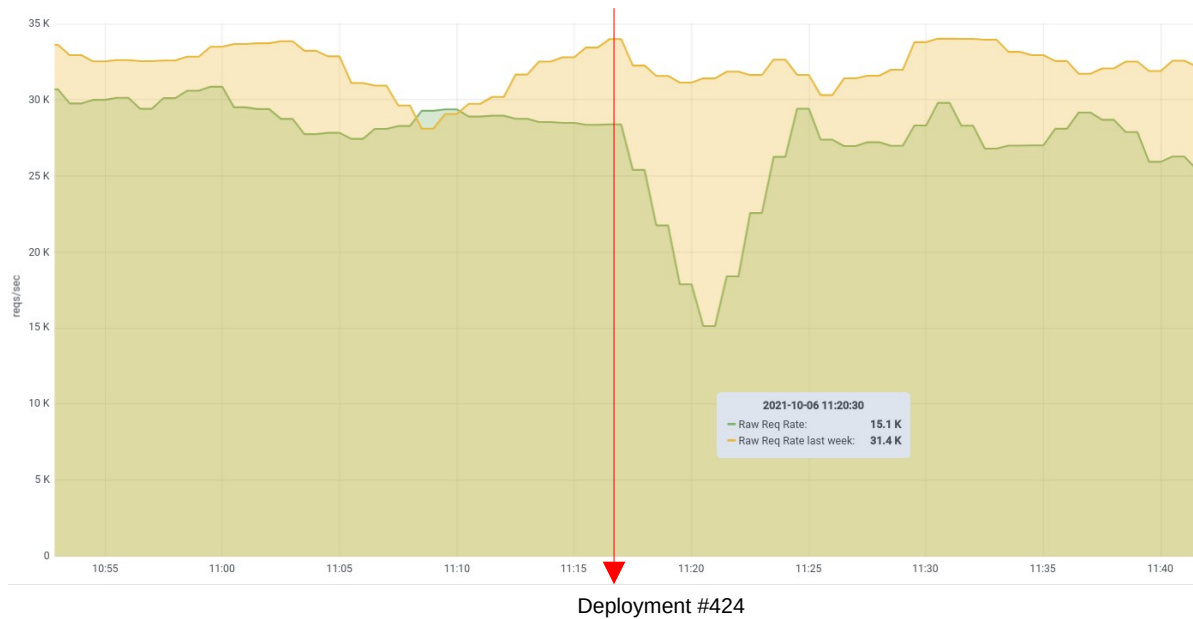
Generate hypotheses

Compare hypotheses

Both execute the task using the same interface



Both reason from the same input



human

I see a drop for 10 minutes right after the deployment



agent

<thinking> To analyze the metrics seen in the Grafana dashboard, I need to use the `image_reader` tool to extract

... image shows a drop in the requests rate for a period of 12 minutes

...</thinking>

Generate hypotheses

Gather
configuration

Collect
knowledge

Metrics to
Insights

Generate
hypotheses

Compare
hypotheses

```
[7]: res=agent(f"Analyze the metrics in this image and generate a hypothesis: {image_path}")
```

```
<thinking> To analyze the metrics in the image and generate a hypothesis, I need to first extract the data from the image. The image is expected to contain a time series plot of metrics with annotations for events such as the rollout of a new version of a component. I will use the 'image_reader' tool to process the image and obtain the necessary information. </thinking>
```

```
Tool #1: image_reader
```

```
The image shows a time series plot of metrics, specifically the request rate over time. There is a significant drop in the request rate at 11:20, coinciding with the deployment of version #424. The raw request rate drops from 31.4K to 15.1K.
```

```
Hypothesis: The deployment of version #424 may have caused a reduction in the system's capacity to handle requests, leading to the observed drop in the request rate.
```

```
[8]: res.metrics
```

It wasn't automation, it was understanding

Generate hypotheses

```
[7]: res=agent(f"Analyze the metrics in this image and generate a hypothesis: {image_path}")
```

```
<thinking> To analyze the metrics in the image and generate a hypothesis, I need to first extract the data from the image. The image is expected to contain a time series plot of metrics with annotations for events such as the rollout of a new version of a component. I will use the 'image_reader' tool to process the image and obtain the necessary information. </thinking>
```

```
Tool #1: image_reader
```

```
The image shows a time series plot of metrics, specifically the request rate over time. There is a significant drop in the request rate at 11:20, coinciding with the deployment of version #424. The raw request rate drops from 31.4K to 15.1K.
```

```
**Hypothesis:** The deployment of version #424 may have caused a reduction in the system's capacity to handle requests, leading to the observed drop in the request rate.
```

```
[8]: res.metrics
```

```
[8]: EventLoopMetrics(cycle_count=2, tool_metrics={'image_reader': ToolMetrics(tool={'toolUseId': 'tooluse_czb1JHJATq0187UK8eJ29Q', 'name': 'image_reader', 'input': {'image_path': 'drop1.png'}}), call_count=1, success_count=1, error_count=0, total_time=0.005484104156494141)}, cycle_durations=[1.8287336826324463], traces=[<strands.telemetry.metrics.Trace object at 0x7f444f3991d0>, <strands.telemetry.metrics.Trace object at 0x7f444383e7d0>], accumulated_usage={'inputTokens': 5514, 'outputTokens': 201, 'totalTokens': 5715}, accumulated_metrics={'latencyMs': 3417})
```

Gather
configuration

Collect
knowledge

Metrics to
Insights

Generate
hypotheses

Compare
hypotheses

Get feedback

The image shows a time series plot of metrics, specifically the request rate over time. There is a significant drop in the request rate at 11:20, coinciding with the deployment of version #424. The raw request rate drops from 31.4K to 15.1K.

Hypothesis: The deployment of version #424 may have caused a reduction in the system's capacity to handle requests, leading to the observed drop in the request rate.

```
[8]: res.metrics

[8]: EventLoopMetrics(cycle_count=2, tool_metrics={'image_reader': ToolMetrics(tool={'toolUseId': 'tooluse_czb1JHJATq0187UK8eJ29Q', 'name': 'image_reader', 'input': {'image_path': 'drop1.png'}}), call_count=1, success_count=1, error_count=0, total_time=0.005484104156494141)}, cycle_durations=[1.8287336826324463], traces=[<strands.telemetry.metrics.Trace object at 0x7f444f3991d0>, <strands.telemetry.metrics.Trace object at 0x7f444383e7d0>], accumulated_usage={'inputTokens': 5514, 'outputTokens': 201, 'totalTokens': 5715}, accumulated_metrics={'latencyMs': 3417})
```

Gather configuration

Collect knowledge

Metrics to Insights

Generate hypotheses

Compare hypotheses

EnvironmentExtractor Info

Prompt
You are given a ticg FAQ might help: https://www.amazon.com/index.php/MonitoringTeam/Alarms/F.....

Environment Stage Generated by AI
Kati/EU/Gibraltar/EU/Backend/Prod/Polimakri

Latency
435 ms


Input tokens
1329

Output tokens
12

Total tokens
1341

TextualInsights Info

Prompt
Analyze the metrics in this image and generate a hypothesis...



Hypothesis Generated by AI
The deployment of version #424 may have caused a reduction in the system's capacity to handle requests, leading to the observed drop in the request rate.

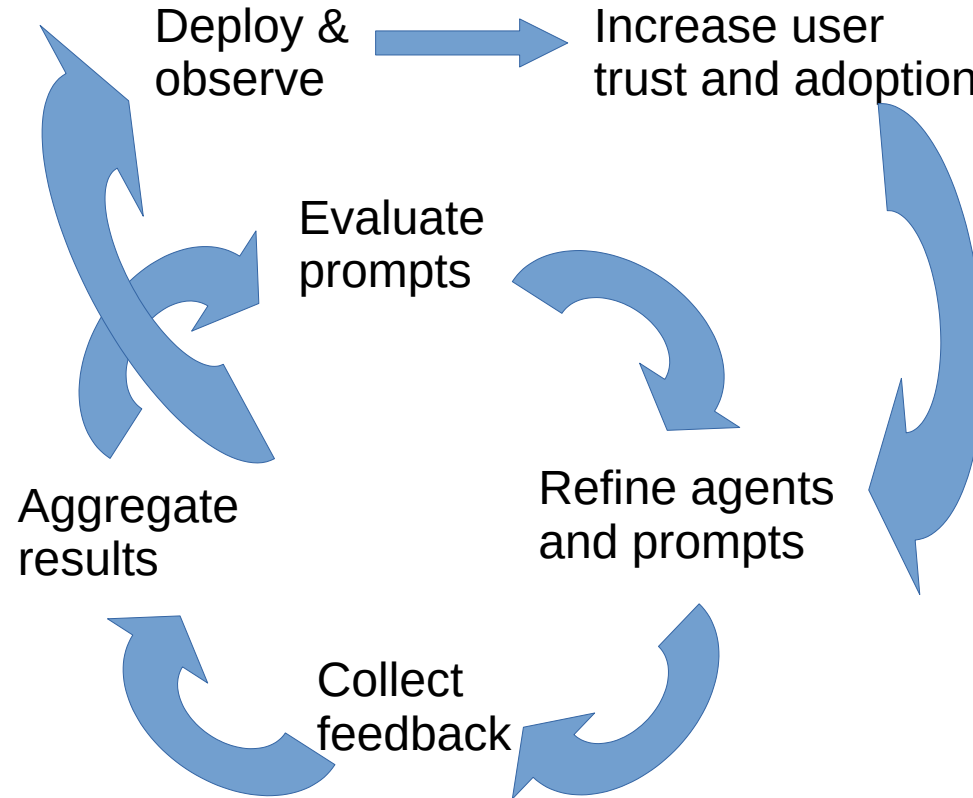
Latency
1828 ms

Input tokens
5514

Output tokens
201

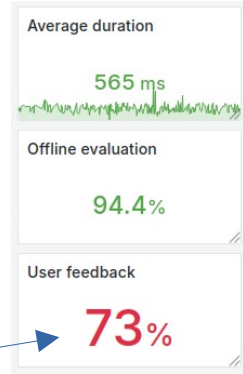
Total tokens
5715

Lesson 1: Evaluation beats intuition



Lesson 2: Feedback as supervision

Hypothesis ✨ *Generated by AI*
The deployment of version #424
reduction in the system's capacity
leading to the observed drop in

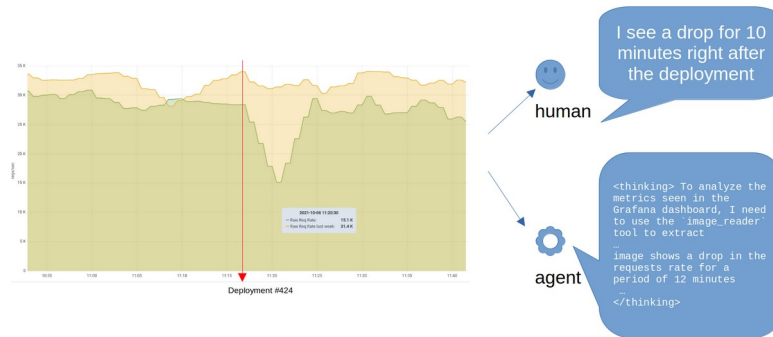


Feedback

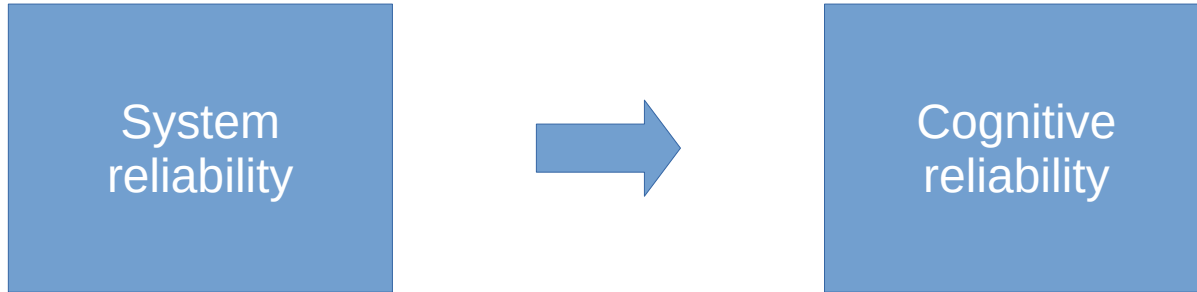
| agent | ticketid | vote | submitted_at | enviro |
|---------------------|------------------------------------|-------------|---------------------|--------|
| HypothesisGenerator | c759bf6e-07d3-405a-a9a8-0e878ffbea | not-helpful | 2025-10-01 16:23:38 | MASC |
| HypothesisGenerator | 4c37e8e6-fb4a-4e86-a37c-71a80d1a6 | helpful | 2025-10-01 16:24:36 | Uber |
| HypothesisGenerator | b5839437-f692-4440-bfe2-ad9f76386 | helpful | 2025-10-01 16:32:34 | Legac |
| HypothesisGenerator | 58597e91-2f46-4727-af89-99956360f | helpful | 2025-10-01 17:12:16 | Preac |
| HypothesisGenerator | f5dab609-3f94-4839-a5eb-6f2b530d1 | helpful | 2025-10-01 17:12:22 | Simpl |
| HypothesisGenerator | 4c216732-17d9-4300-9caf-ba29fb3b0 | helpful | 2025-10-01 17:12:27 | Cima |

Lesson 3: Agents learn with us

*We stopped teaching models,
we started learning together.*



Beyond automation



Redundancy,
alerts,
uptime, MTTR

Discoverability,
explainability,
reasoning

*Reliability used to mean uptime,
now it means understanding*

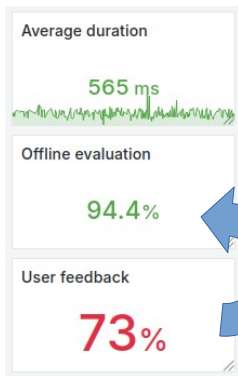
Key takeaways

- Unified tools
- Reasoning loops
- Evaluation culture

New definition

SRE = Site Reasoning Engineering

Future work



| Pass Rate |
|-----------|
| 94.44% |
| 88.89% |
| 88.89% |
| 0.00% |
| 88.89% |
| 72.22% |
| 55.56% |
| 55.56% |

```
- Observation: Record what the tool returned.  
Repeat this loop as needed.
```

```
prompt = """You are an SRE engineer that operates  
from this tool. You do not have access to the internet.
```

```
prompt = """You are an image analysis assistant.  
Your function is to describe the content of an image.
```

Call to Action

Make your expertise addressable

Thank you