Security with ClickHouse

Building for the long term

Name	Evan Johnson
Date	Sept 17, 2024
Email	evan@runreveal.com
Twitter	@ejcx_

$(whoami)

2015-17	First Security Engineer at Cloudflare
2017-18	First Security Hire at Segment
2018 - 2022	Cloudflare, Sr Director of Security Engineering
2023 - 20XX	Cofounder CEO of RunReveal

runreveal

Who knows what a SIEM is?

Security people have a lot of logs

Security people have a lot of logs

CDR

SIEM

ITDR

XDR

MDR

Security people want to

- Search for threats
- Keep their data for > 1 yr
- Not think about “data problems”

splunk>®

elastic

snowflake

Google
Big Query

ClickHouse

splunk>®

elastic

LogRhythm®

exabeam®

ArcSight™
An HP Company

snowflake

Microsoft
Sentinel

Google
Big Query

ClickHouse

ALIEN VAULT®

Chronicle

QRadar

Easy to start

elastic

snowflake

Google
Big Query

Enterprise Ready

splunk>

elastic

???? snowflake ????

Best Option

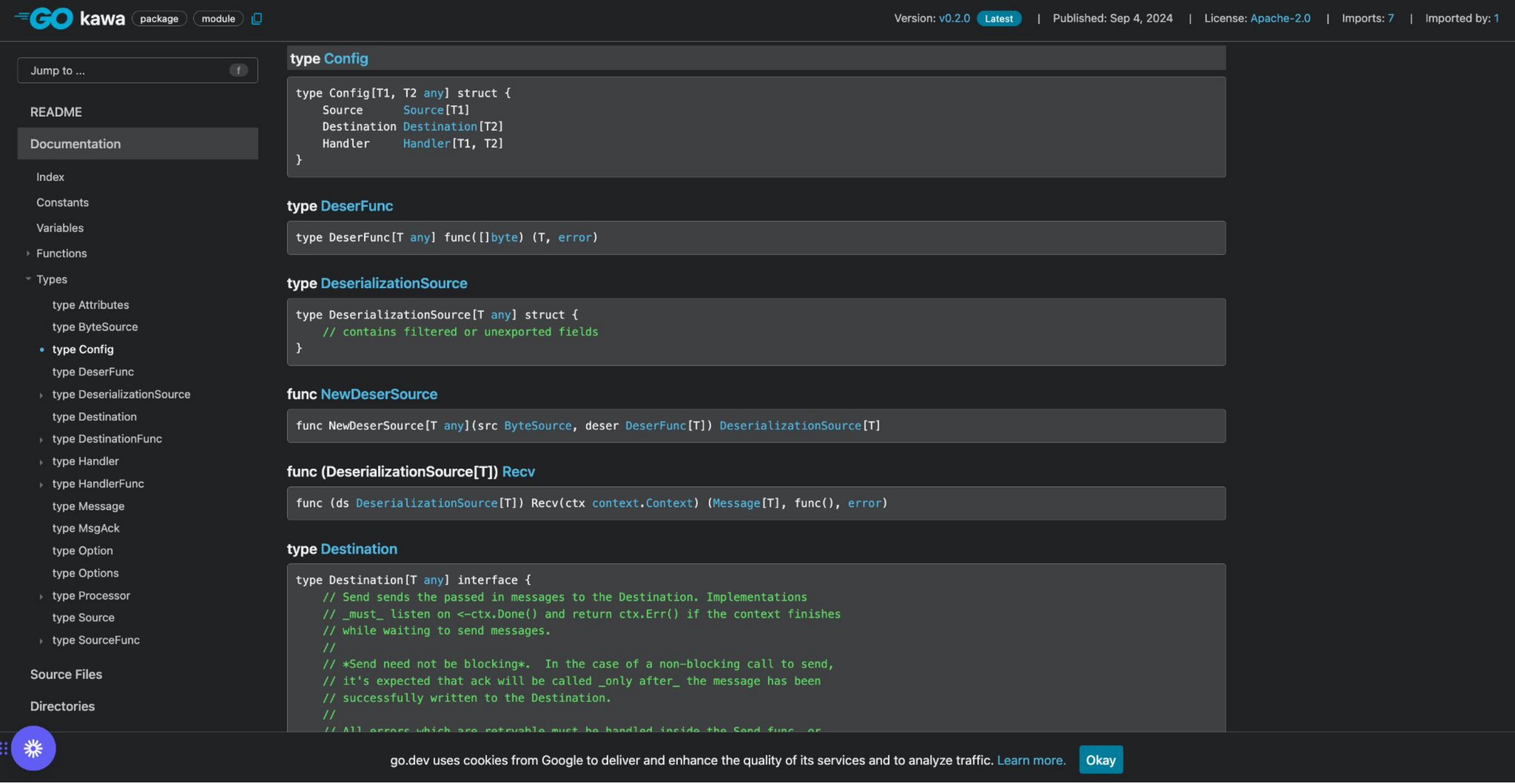
If you don’t want to think about data problems

You actually want speed, reliability, good ux, cost-efficiency, technical support, clean data, etc etc.

GO + ClickHouse

= runreveal

rrq



GO kawa package module
Version: v0.2.0 Latest | Published: Sep 4, 2024 | License: Apache-2.0 | Imports: 7 | Imported by: 1
Jump to ... f

- README
- Documentation
  - Index
  - Constants
  - Variables
  - Functions
  - Types
    - type Attributes
    - type ByteSource
    - type Config
    - type DeserFunc
    - type DeserializationSource
    - type Destination
    - type DestinationFunc
    - type Handler
    - type HandlerFunc
    - type Message
    - type MsgAck
    - type Option
    - type Options
    - type Processor
    - type Source
    - type SourceFunc
- Source Files
- Directories

type Config

```
type Config[T1, T2 any] struct {
    Source      Source[T1]
    Destination Destination[T2]
    Handler     Handler[T1, T2]
}
```

type DeserFunc

```
type DeserFunc[T any] func([]byte) (T, error)
```

type DeserializationSource

```
type DeserializationSource[T any] struct {
    // contains filtered or unexported fields
}
```

func NewDeserSource

```
func NewDeserSource[T any](src ByteSource, deser DeserFunc[T]) DeserializationSource[T]
```

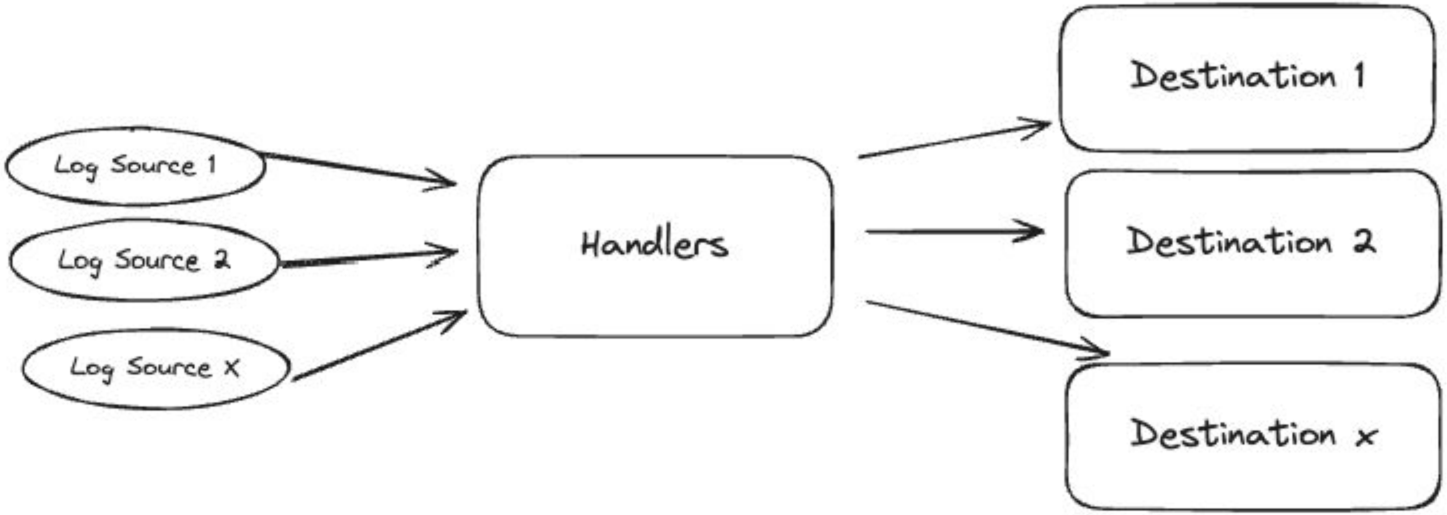
func (DeserializationSource[T]) Recv

```
func (ds DeserializationSource[T]) Recv(ctx context.Context) (Message[T], func(), error)
```

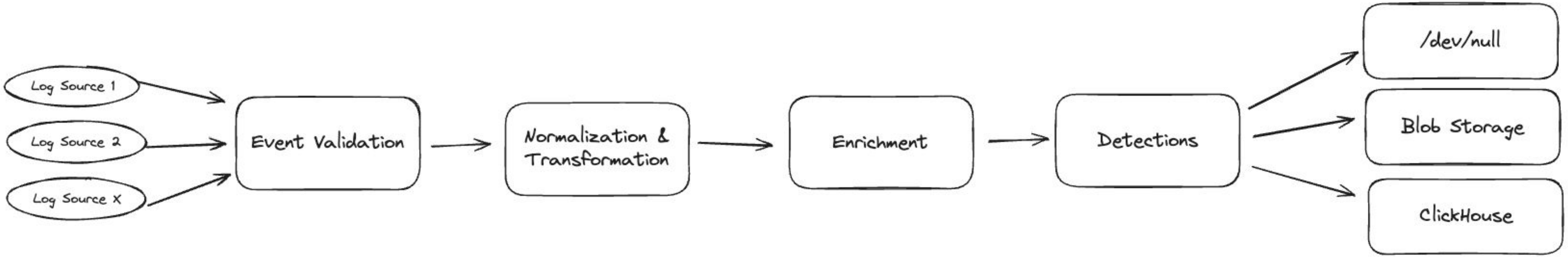
type Destination

```
type Destination[T any] interface {
    // Send sends the passed in messages to the Destination. Implementations
    // _must_ listen on <-ctx.Done() and return ctx.Err() if the context finishes
    // while waiting to send messages.
    //
    // *Send need not be blocking*.  In the case of a non-blocking call to send,
    // it's expected that ack will be called _only after_ the message has been
    // successfully written to the Destination.
    //
    // All errors which are retryable must be handled inside the Send func, or
```

https://github.com/runreveal/kawa



https://github.com/runreveal/kawa



Log Source 1

Log Source 2

Log Source X

Event Validation

Normalization & Transformation

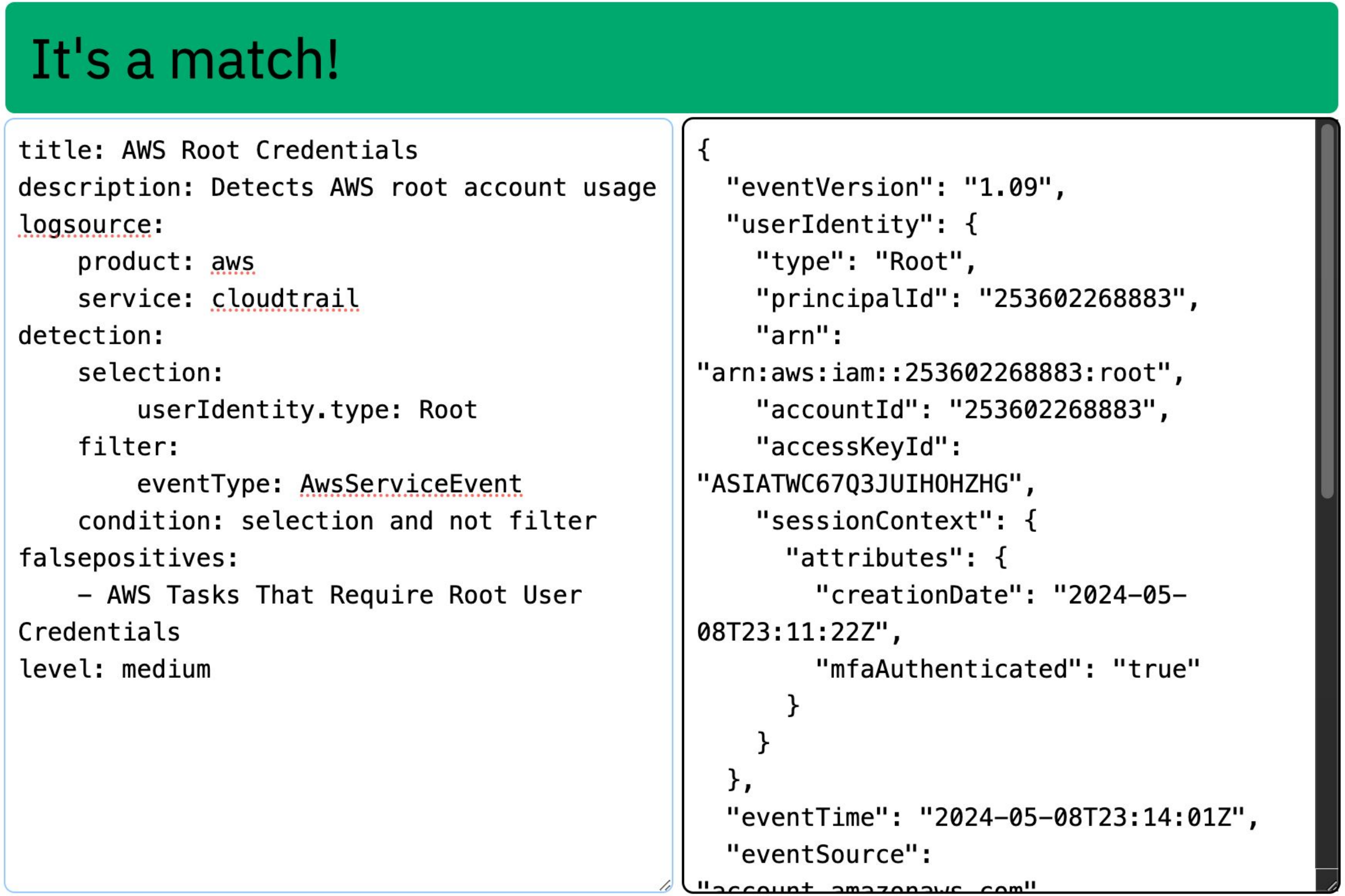
Enrichment

Detections

/dev/null

Blob Storage

ClickHouse



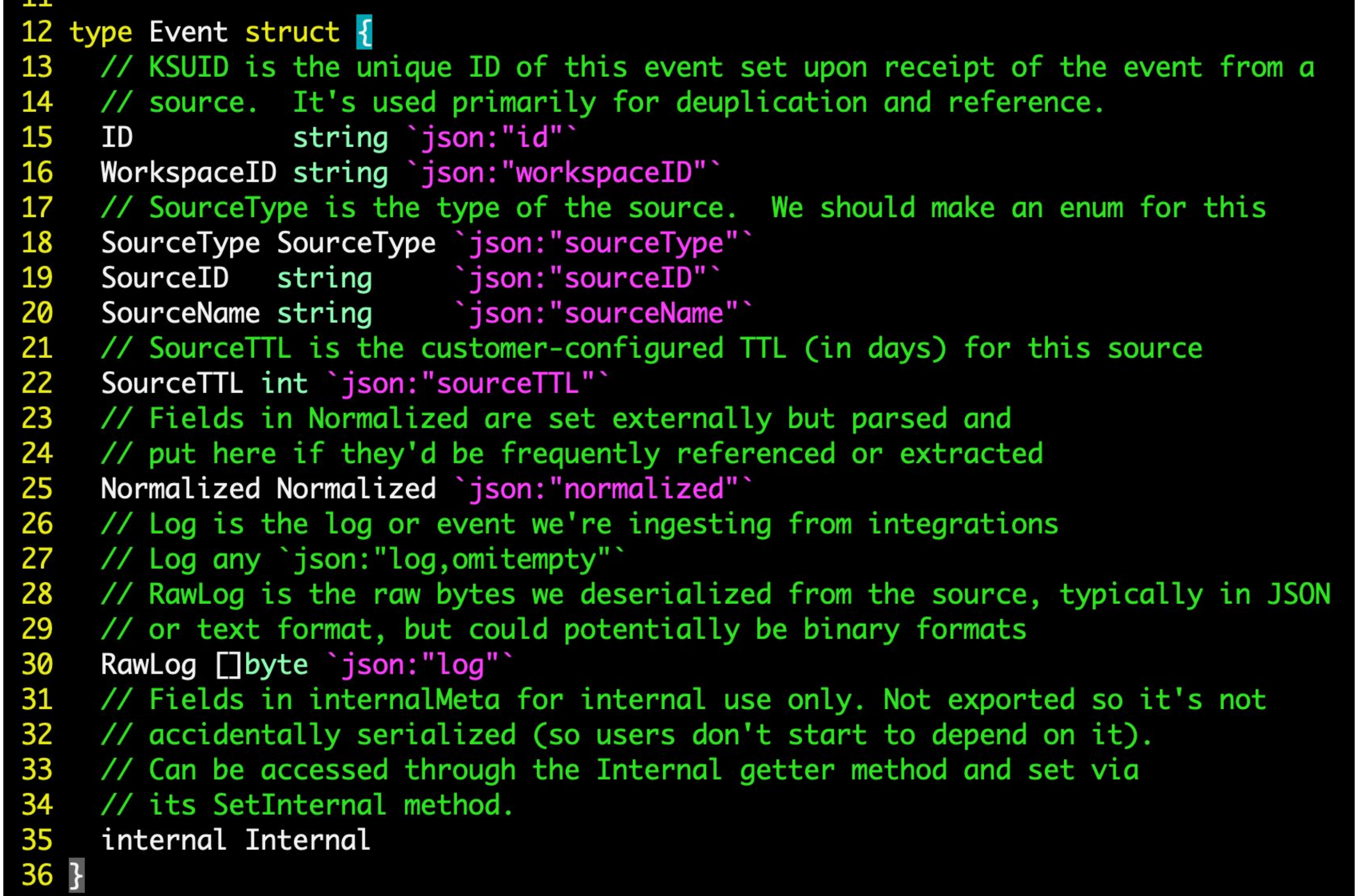
It's a match!

```yaml
title: AWS Root Credentials
description: Detects AWS root account usage
logsource:
    product: aws
    service: cloudtrail
detection:
    selection:
        userIdentity.type: Root
    filter:
        eventType: AwsServiceEvent
    condition: selection and not filter
falsepositives:
    - AWS Tasks That Require Root User
Credentials
level: medium
```

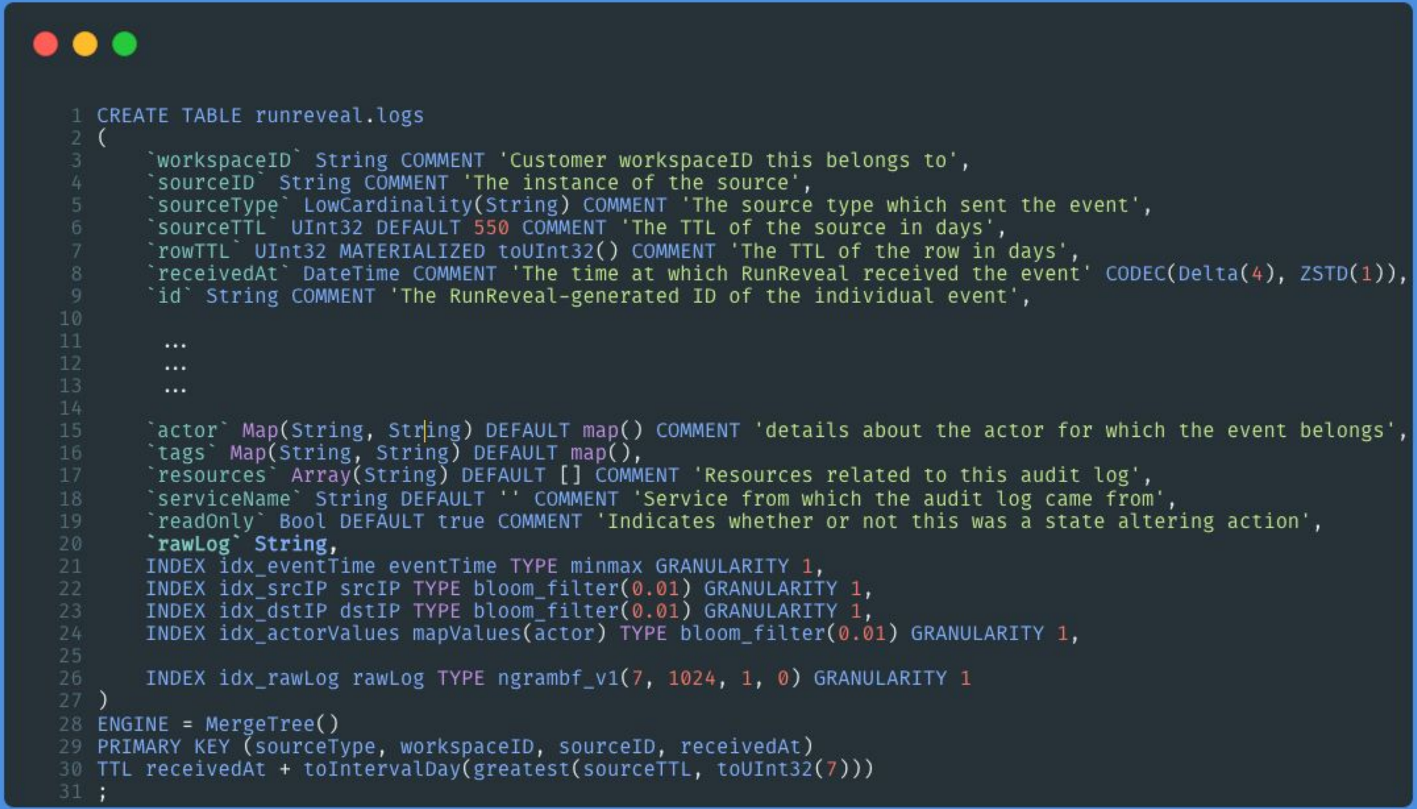
```json
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "Root",
    "principalId": "253602268883",
    "arn":
"arn:aws:iam::253602268883:root",
    "accountId": "253602268883",
    "accessKeyId":
"ASIATWC67Q3JUIHOHZHG",
    "sessionContext": {
      "attributes": {
        "creationDate": "2024-05-
08T23:11:22Z",
        "mfaAuthenticated": "true"
      }
    }
  },
  "eventTime": "2024-05-08T23:14:01Z",
  "eventSource":
"account.amazonaws.com"
```

https://github.com/runreveal/sigmalite

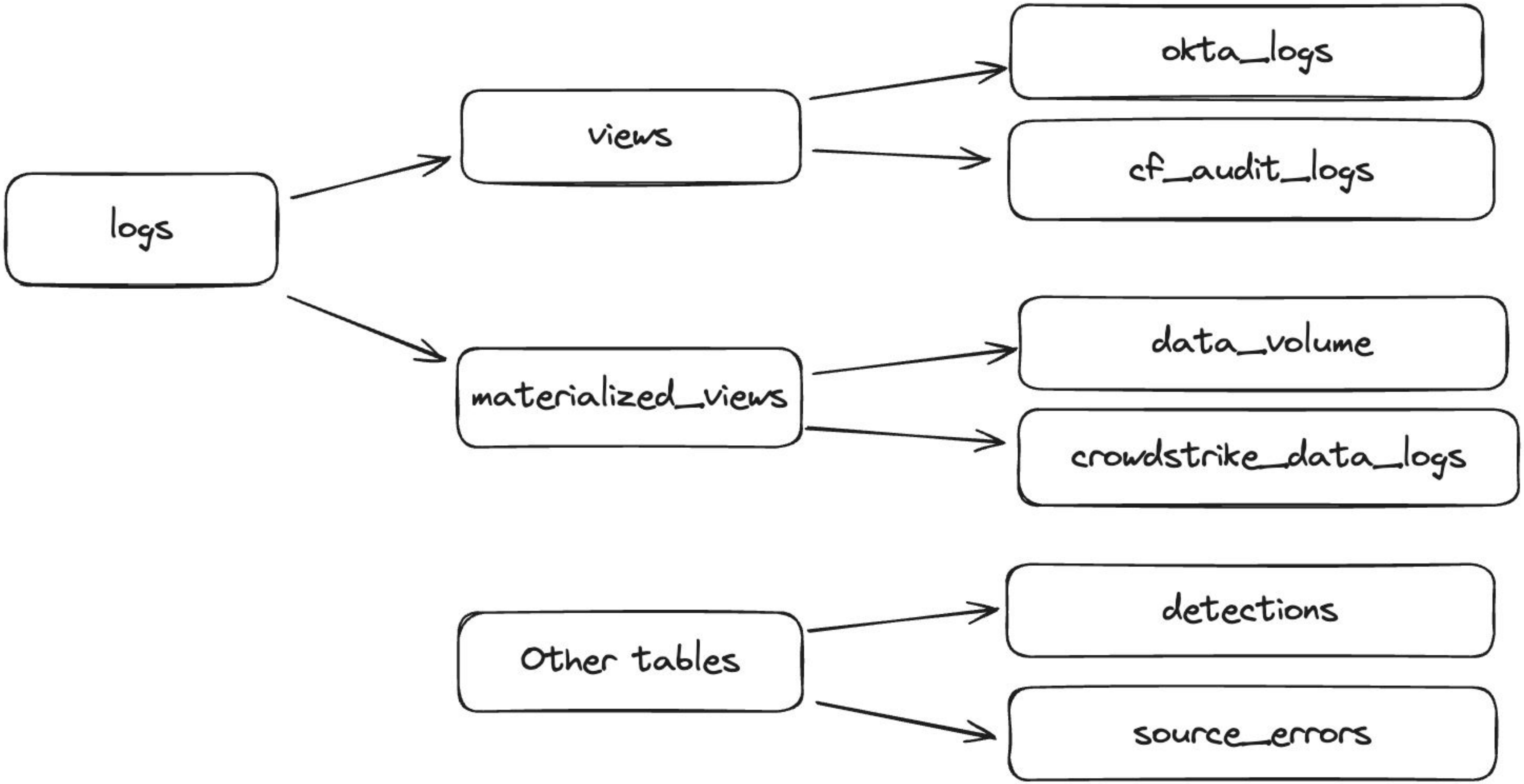
clickhouse

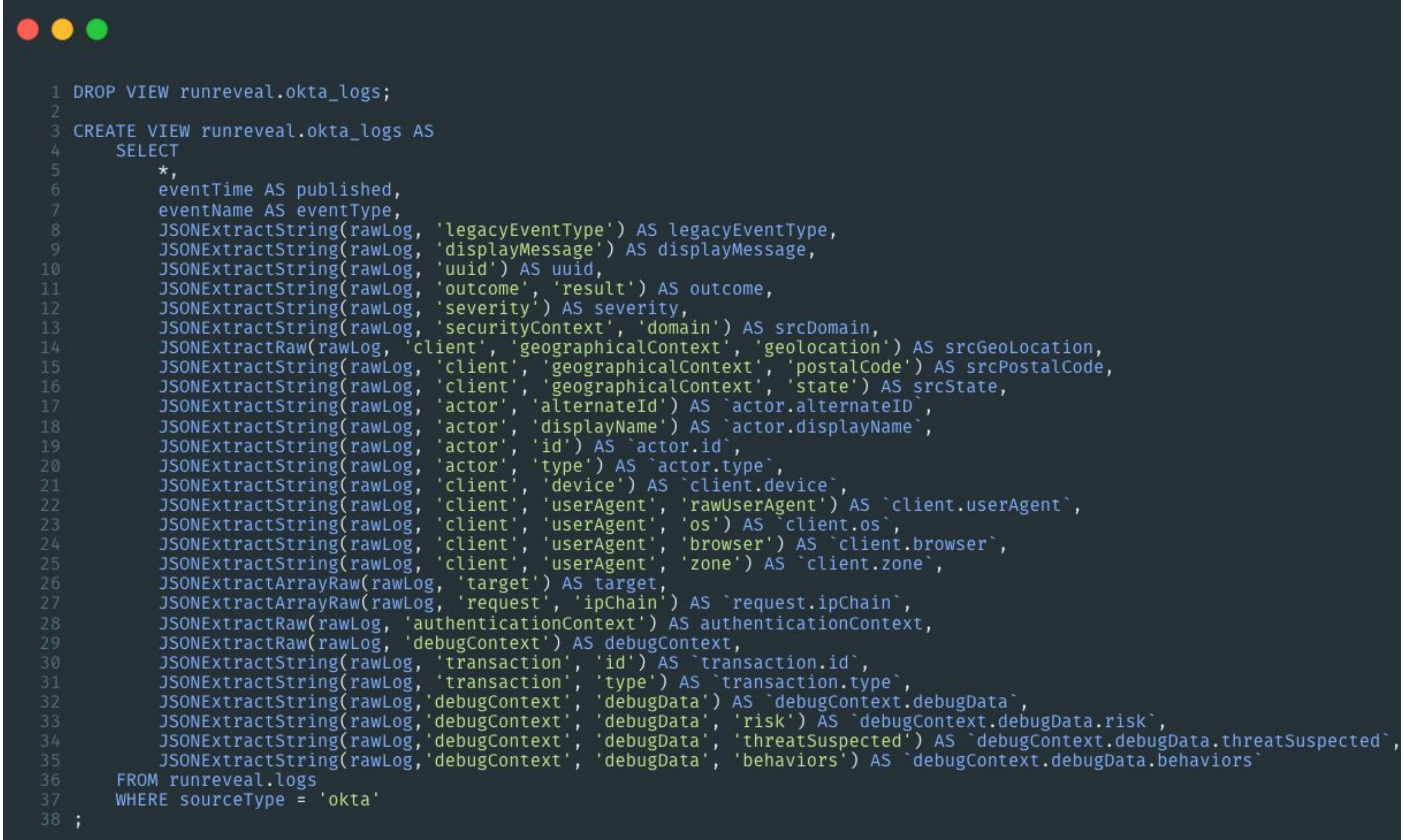


```go
type Event struct {
  // KSUID is the unique ID of this event set upon receipt of the event from a
  // source.  It's used primarily for deuplication and reference.
  ID          string `json:"id"`
  WorkspaceID string `json:"workspaceID"`
  // SourceType is the type of the source.  We should make an enum for this
  SourceType SourceType `json:"sourceType"`
  SourceID   string     `json:"sourceID"`
  SourceName string     `json:"sourceName"`
  // SourceTTL is the customer-configured TTL (in days) for this source
  SourceTTL int `json:"sourceTTL"`
  // Fields in Normalized are set externally but parsed and
  // put here if they'd be frequently referenced or extracted
  Normalized Normalized `json:"normalized"`
  // Log is the log or event we're ingesting from integrations
  // Log any `json:"log,omitempty"`
  // RawLog is the raw bytes we deserialized from the source, typically in JSON
  // or text format, but could potentially be binary formats
  RawLog []byte `json:"log"`
  // Fields in internalMeta for internal use only. Not exported so it's not
  // accidentally serialized (so users don't start to depend on it).
  // Can be accessed through the Internal getter method and set via
  // its SetInternal method.
  internal Internal
}
```

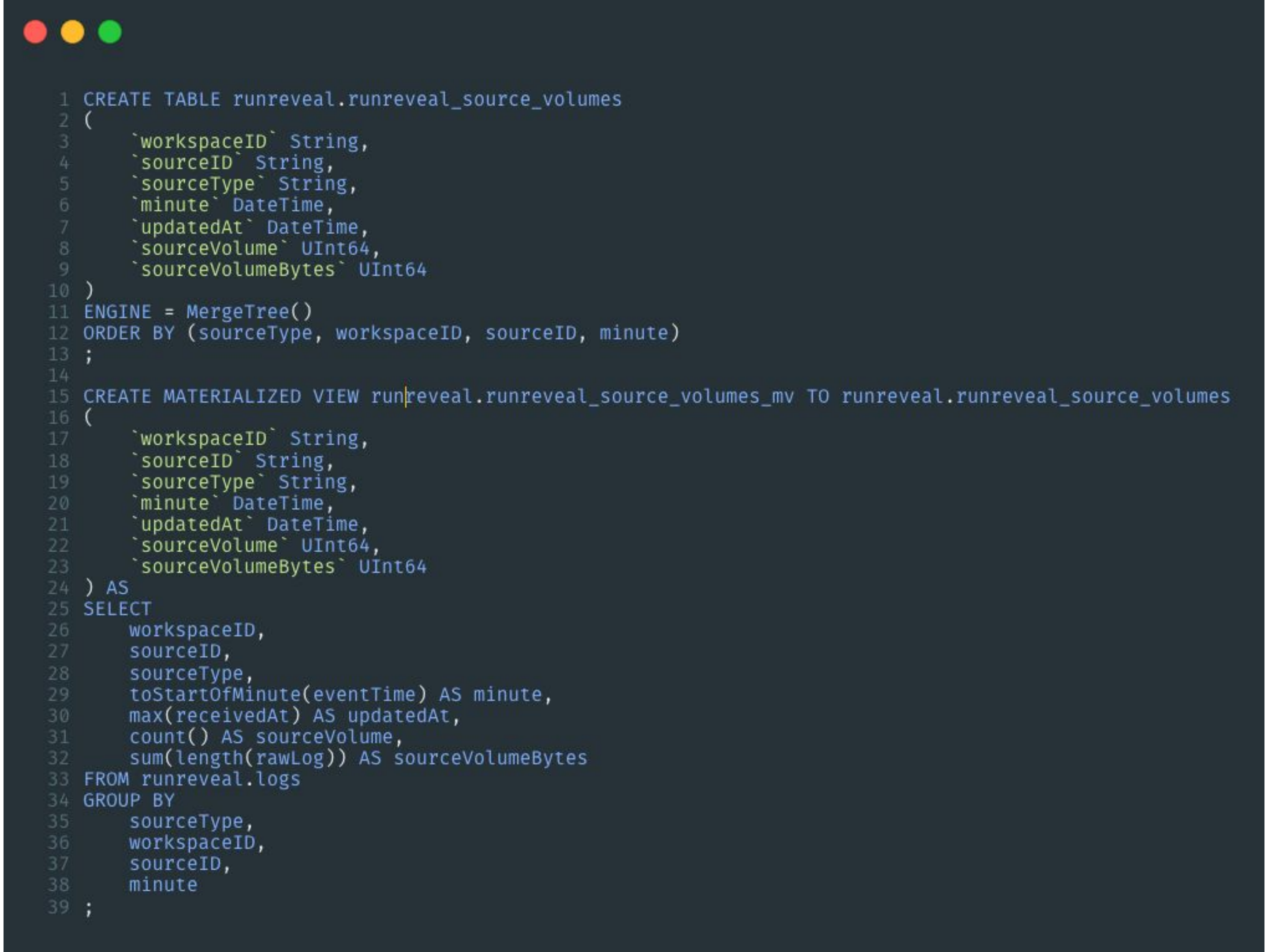


```
CREATE TABLE runreveal.logs
(
    `workspaceID` String COMMENT 'Customer workspaceID this belongs to',
    `sourceID` String COMMENT 'The instance of the source',
    `sourceType` LowCardinality(String) COMMENT 'The source type which sent the event',
    `sourceTTL` UInt32 DEFAULT 550 COMMENT 'The TTL of the source in days',
    `rowTTL` UInt32 MATERIALIZED toUInt32() COMMENT 'The TTL of the row in days',
    `receivedAt` DateTime COMMENT 'The time at which RunReveal received the event' CODEC(Delta(4), ZSTD(1)),
    `id` String COMMENT 'The RunReveal-generated ID of the individual event',

     ...
     ...
     ...

    `actor` Map(String, String) DEFAULT map() COMMENT 'details about the actor for which the event belongs',
    `tags` Map(String, String) DEFAULT map(),
    `resources` Array(String) DEFAULT [] COMMENT 'Resources related to this audit log',
    `serviceName` String DEFAULT '' COMMENT 'Service from which the audit log came from',
    `readOnly` Bool DEFAULT true COMMENT 'Indicates whether or not this was a state altering action',
    `rawLog` String,
    INDEX idx_eventTime eventTime TYPE minmax GRANULARITY 1,
    INDEX idx_srcIP srcIP TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_dstIP dstIP TYPE bloom_filter(0.01) GRANULARITY 1,
    INDEX idx_actorValues mapValues(actor) TYPE bloom_filter(0.01) GRANULARITY 1,

    INDEX idx_rawLog rawLog TYPE ngrambf_v1(7, 1024, 1, 0) GRANULARITY 1
)
ENGINE = MergeTree()
PRIMARY KEY (sourceType, workspaceID, sourceID, receivedAt)
TTL receivedAt + toIntervalDay(greatest(sourceTTL, toUInt32(7)))
;
```

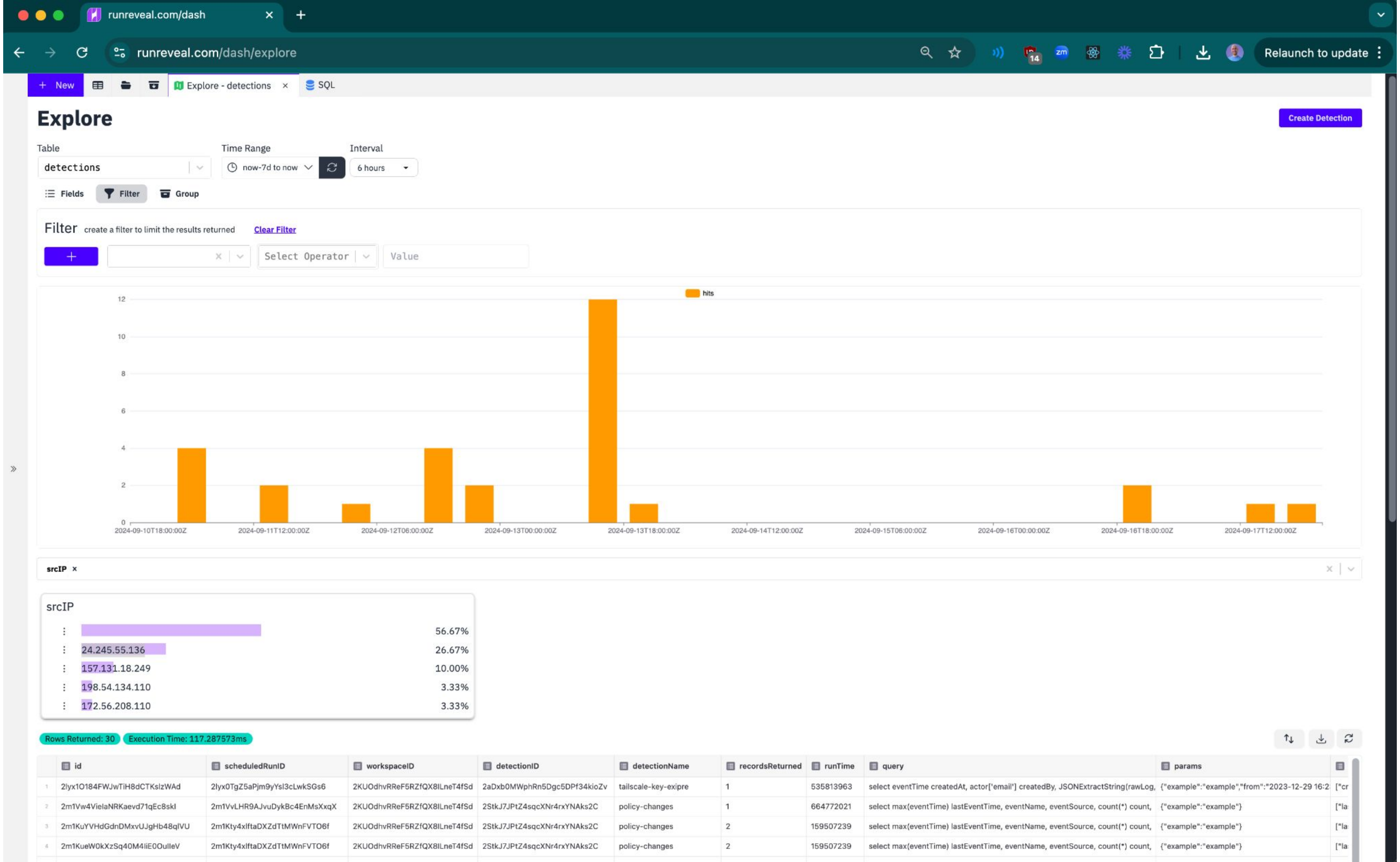

logs
views
materialized_views
Other tables
okta_logs
cf_audit_logs
data_volume
crowdstrike_data_logs
detections
source_errors



```sql
DROP VIEW runreveal.okta_logs;

CREATE VIEW runreveal.okta_logs AS
    SELECT
        *,
        eventTime AS published,
        eventName AS eventType,
        JSONExtractString(rawLog, 'legacyEventType') AS legacyEventType,
        JSONExtractString(rawLog, 'displayMessage') AS displayMessage,
        JSONExtractString(rawLog, 'uuid') AS uuid,
        JSONExtractString(rawLog, 'outcome', 'result') AS outcome,
        JSONExtractString(rawLog, 'severity') AS severity,
        JSONExtractString(rawLog, 'securityContext', 'domain') AS srcDomain,
        JSONExtractRaw(rawLog, 'client', 'geographicalContext', 'geolocation') AS srcGeoLocation,
        JSONExtractString(rawLog, 'client', 'geographicalContext', 'postalCode') AS srcPostalCode,
        JSONExtractString(rawLog, 'client', 'geographicalContext', 'state') AS srcState,
        JSONExtractString(rawLog, 'actor', 'alternateId') AS `actor.alternateID`,
        JSONExtractString(rawLog, 'actor', 'displayName') AS `actor.displayName`,
        JSONExtractString(rawLog, 'actor', 'id') AS `actor.id`,
        JSONExtractString(rawLog, 'actor', 'type') AS `actor.type`,
        JSONExtractString(rawLog, 'client', 'device') AS `client.device`,
        JSONExtractString(rawLog, 'client', 'userAgent', 'rawUserAgent') AS `client.userAgent`,
        JSONExtractString(rawLog, 'client', 'userAgent', 'os') AS `client.os`,
        JSONExtractString(rawLog, 'client', 'userAgent', 'browser') AS `client.browser`,
        JSONExtractString(rawLog, 'client', 'userAgent', 'zone') AS `client.zone`,
        JSONExtractArrayRaw(rawLog, 'target') AS target,
        JSONExtractArrayRaw(rawLog, 'request', 'ipChain') AS `request.ipChain`,
        JSONExtractRaw(rawLog, 'authenticationContext') AS authenticationContext,
        JSONExtractRaw(rawLog, 'debugContext') AS debugContext,
        JSONExtractString(rawLog, 'transaction', 'id') AS `transaction.id`,
        JSONExtractString(rawLog, 'transaction', 'type') AS `transaction.type`,
        JSONExtractString(rawLog,'debugContext', 'debugData') AS `debugContext.debugData`,
        JSONExtractString(rawLog,'debugContext', 'debugData', 'risk') AS `debugContext.debugData.risk`,
        JSONExtractString(rawLog,'debugContext', 'debugData', 'threatSuspected') AS `debugContext.debugData.threatSuspected`,
        JSONExtractString(rawLog,'debugContext', 'debugData', 'behaviors') AS `debugContext.debugData.behaviors`
    FROM runreveal.logs
    WHERE sourceType = 'okta'
;
```



```sql
CREATE TABLE runreveal.runreveal_source_volumes
(
    `workspaceID` String,
    `sourceID` String,
    `sourceType` String,
    `minute` DateTime,
    `updatedAt` DateTime,
    `sourceVolume` UInt64,
    `sourceVolumeBytes` UInt64
)
ENGINE = MergeTree()
ORDER BY (sourceType, workspaceID, sourceID, minute)
;

CREATE MATERIALIZED VIEW runreveal.runreveal_source_volumes_mv TO runreveal.runreveal_source_volumes
(
    `workspaceID` String,
    `sourceID` String,
    `sourceType` String,
    `minute` DateTime,
    `updatedAt` DateTime,
    `sourceVolume` UInt64,
    `sourceVolumeBytes` UInt64
) AS
SELECT
    workspaceID,
    sourceID,
    sourceType,
    toStartOfMinute(eventTime) AS minute,
    max(receivedAt) AS updatedAt,
    count() AS sourceVolume,
    sum(length(rawLog)) AS sourceVolumeBytes
FROM runreveal.logs
GROUP BY
    sourceType,
    workspaceID,
    sourceID,
    minute
;
```



Explore

Table: detections

Time Range: now-7d to now

Interval: 6 hours

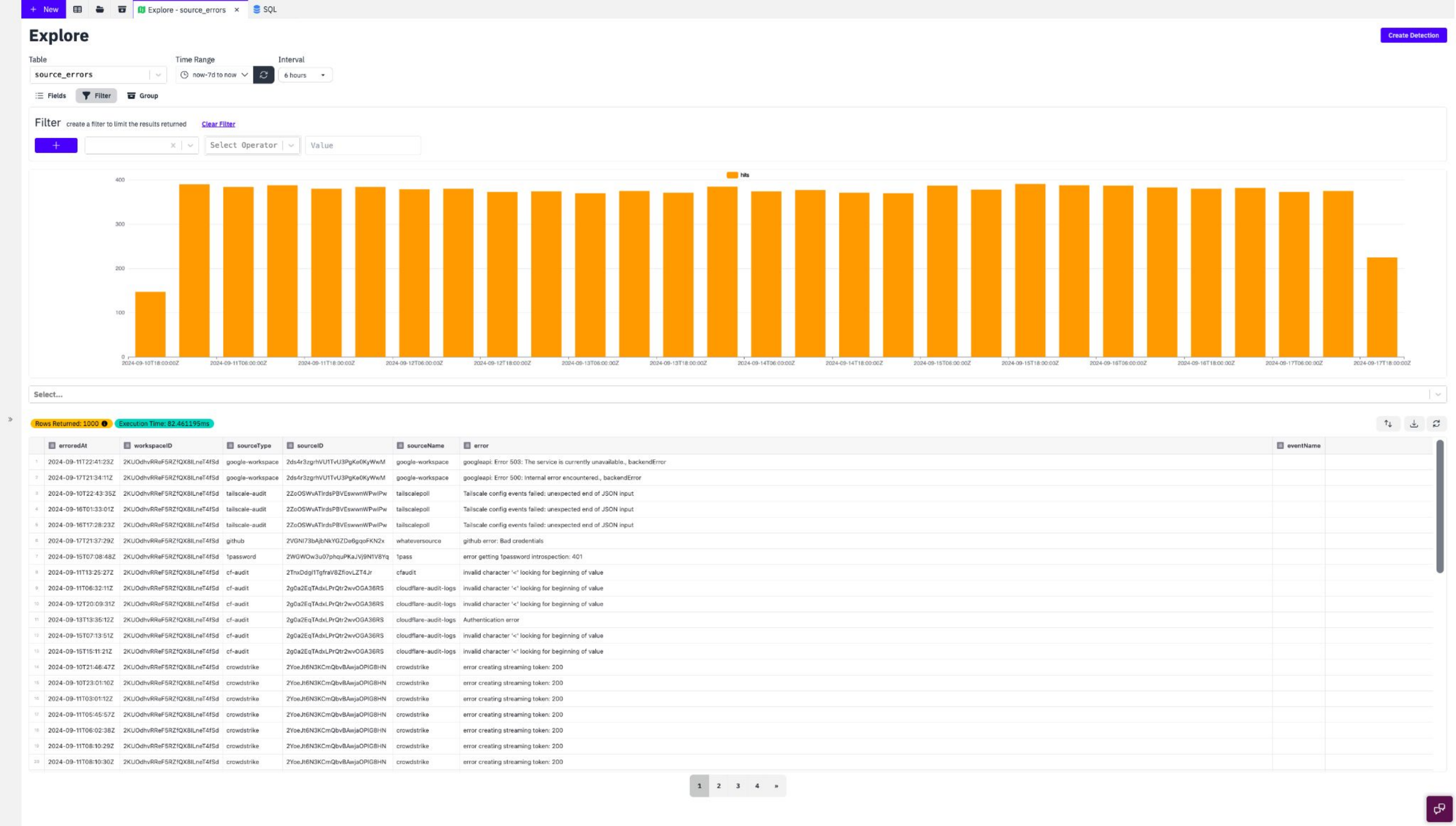
Filter create a filter to limit the results returned Clear Filter

srcIP

	56.67%
24.245.55.136	26.67%
157.131.18.249	10.00%
198.54.134.110	3.33%
172.56.208.110	3.33%

Rows Returned: 30 Execution Time: 117.287573ms

	id	scheduledRunID	workspaceID	detectionID	detectionName	recordsReturned	runTime	query	params	
1	2lyx1O184FWJwTiH8dCTKslzWAd	2lyx0TgZ5aPjm9yYsl3cLwkSGs6	2KUOdhvRReF5RZfQX8ILneT4fSd	2aDxb0MWphRn5Dgc5DPf34kioZv	tailscale-key-exipre	1	535813963	select eventTime createdAt, actor['email'] createdBy, JSONExtractString(rawLog,	{"example":"example","from":"2023-12-29 16:2	["cr
2	2m1Vw4VielaNRKaevd71qEc8skl	2m1VvLHR9AJvuDykBc4EnMsXxqX	2KUOdhvRReF5RZfQX8ILneT4fSd	2StkJ7JPtZ4sqcXNr4rxYNAks2C	policy-changes	1	664772021	select max(eventTime) lastEventTime, eventName, eventSource, count(*) count,	{"example":"example"}	["la
3	2m1KuYVHdGdnDMxvUJgHb48qlVU	2m1Kty4xlftaDXZdTtMWnFVTO6f	2KUOdhvRReF5RZfQX8ILneT4fSd	2StkJ7JPtZ4sqcXNr4rxYNAks2C	policy-changes	2	159507239	select max(eventTime) lastEventTime, eventName, eventSource, count(*) count,	{"example":"example"}	["la
4	2m1KueW0kXzSq40M4iiE0OulleV	2m1Kty4xlftaDXZdTtMWnFVTO6f	2KUOdhvRReF5RZfQX8ILneT4fSd	2StkJ7JPtZ4sqcXNr4rxYNAks2C	policy-changes	2	159507239	select max(eventTime) lastEventTime, eventName, eventSource, count(*) count,	{"example":"example"}	["la



Explore

Table: source_errors

Time Range: now-7d to now

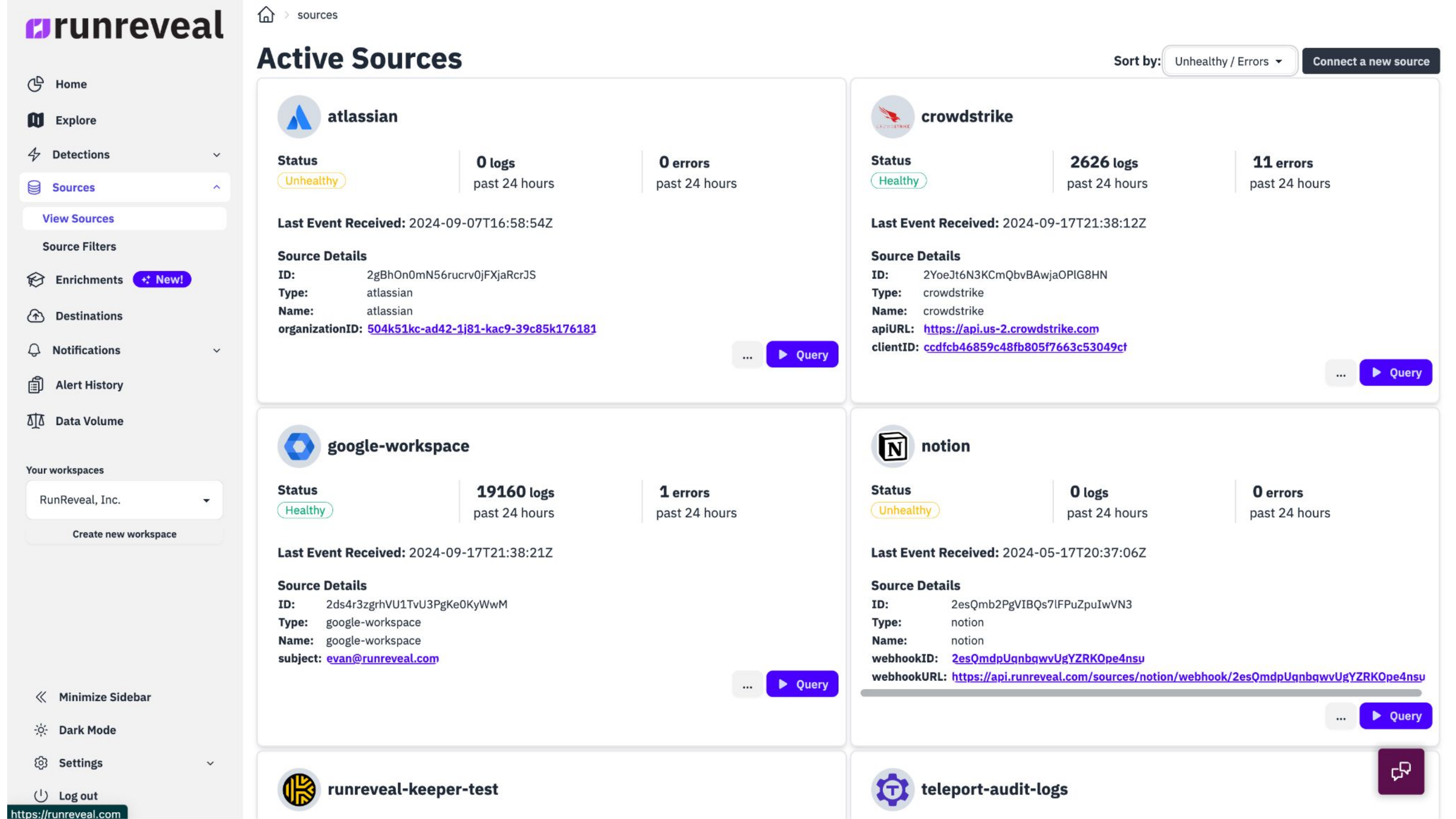
Interval: 6 hours

Filter create a filter to limit the results returned

Rows Returned: 1000

Execution Time: 82.461195ms

	erroredAt	workspaceID	sourceType	sourceID	sourceName	error	eventName
1	2024-09-11T22:41:23Z	2KUOdhvRReF5RZfQX8ILneT4fSd	google-workspace	2ds4r3zgrhVU1TvU3PgKe0KyWwM	google-workspace	googleapi: Error 503: The service is currently unavailable., backendError	
2	2024-09-17T21:34:11Z	2KUOdhvRReF5RZfQX8ILneT4fSd	google-workspace	2ds4r3zgrhVU1TvU3PgKe0KyWwM	google-workspace	googleapi: Error 500: Internal error encountered., backendError	
3	2024-09-10T22:43:35Z	2KUOdhvRReF5RZfQX8ILneT4fSd	tailscale-audit	2ZoOSWvATlrdsPBVEswwnWPwlPw	tailscalepoll	Tailscale config events failed: unexpected end of JSON input	
4	2024-09-16T01:33:01Z	2KUOdhvRReF5RZfQX8ILneT4fSd	tailscale-audit	2ZoOSWvATlrdsPBVEswwnWPwlPw	tailscalepoll	Tailscale config events failed: unexpected end of JSON input	
5	2024-09-16T17:28:23Z	2KUOdhvRReF5RZfQX8ILneT4fSd	tailscale-audit	2ZoOSWvATlrdsPBVEswwnWPwlPw	tailscalepoll	Tailscale config events failed: unexpected end of JSON input	
6	2024-09-17T21:37:29Z	2KUOdhvRReF5RZfQX8ILneT4fSd	github	2VGNI73bAjbNkYGZDe6gqoFKN2x	whateversource	github error: Bad credentials	
7	2024-09-15T07:08:48Z	2KUOdhvRReF5RZfQX8ILneT4fSd	1password	2WGWOw3u07phquPKaJVj9N1V8Yq	1pass	error getting 1password introspection: 401	
8	2024-09-11T13:25:27Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2TnxDdgl1TgfraV8ZfiovLZT4Jr	cfaudit	invalid character '<' looking for beginning of value	
9	2024-09-11T06:32:11Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2g0a2EqTAdxLPrQtr2wvOGA36RS	cloudflare-audit-logs	invalid character '<' looking for beginning of value	
10	2024-09-12T20:09:31Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2g0a2EqTAdxLPrQtr2wvOGA36RS	cloudflare-audit-logs	invalid character '<' looking for beginning of value	
11	2024-09-13T13:35:12Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2g0a2EqTAdxLPrQtr2wvOGA36RS	cloudflare-audit-logs	Authentication error	
12	2024-09-15T07:13:51Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2g0a2EqTAdxLPrQtr2wvOGA36RS	cloudflare-audit-logs	invalid character '<' looking for beginning of value	
13	2024-09-15T15:11:21Z	2KUOdhvRReF5RZfQX8ILneT4fSd	cf-audit	2g0a2EqTAdxLPrQtr2wvOGA36RS	cloudflare-audit-logs	invalid character '<' looking for beginning of value	
14	2024-09-10T21:46:47Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
15	2024-09-10T23:01:10Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
16	2024-09-11T03:01:12Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
17	2024-09-11T05:45:57Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
18	2024-09-11T06:02:38Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
19	2024-09-11T08:10:29Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	
20	2024-09-11T08:10:30Z	2KUOdhvRReF5RZfQX8ILneT4fSd	crowdstrike	2YoeJt6N3KCmQbvBAwjaOPIG8HN	crowdstrike	error creating streaming token: 200	



Active Sources

atlassian

Status Unhealthy | **0** logs past 24 hours | **0** errors past 24 hours

Last Event Received: 2024-09-07T16:58:54Z

Source Details

ID: 2gBhOn0mN56rucrv0jFXjaRcrJS
Type: atlassian
Name: atlassian
organizationID: 504k51kc-ad42-1j81-kac9-39c85k176181

crowdstrike

Status Healthy | **2626** logs past 24 hours | **11** errors past 24 hours

Last Event Received: 2024-09-17T21:38:12Z

Source Details

ID: 2YoeJt6N3KCmQbvBAwjaOPlG8HN
Type: crowdstrike
Name: crowdstrike
apiURL: https://api.us-2.crowdstrike.com
clientID: ccdfcb46859c48fb805f7663c53049ct

google-workspace

Status Healthy | **19160** logs past 24 hours | **1** errors past 24 hours

Last Event Received: 2024-09-17T21:38:21Z

Source Details

ID: 2ds4r3zgrhVU1TvU3PgKe0KyWwM
Type: google-workspace
Name: google-workspace
subject: evan@runreveal.com

notion

Status Unhealthy | **0** logs past 24 hours | **0** errors past 24 hours

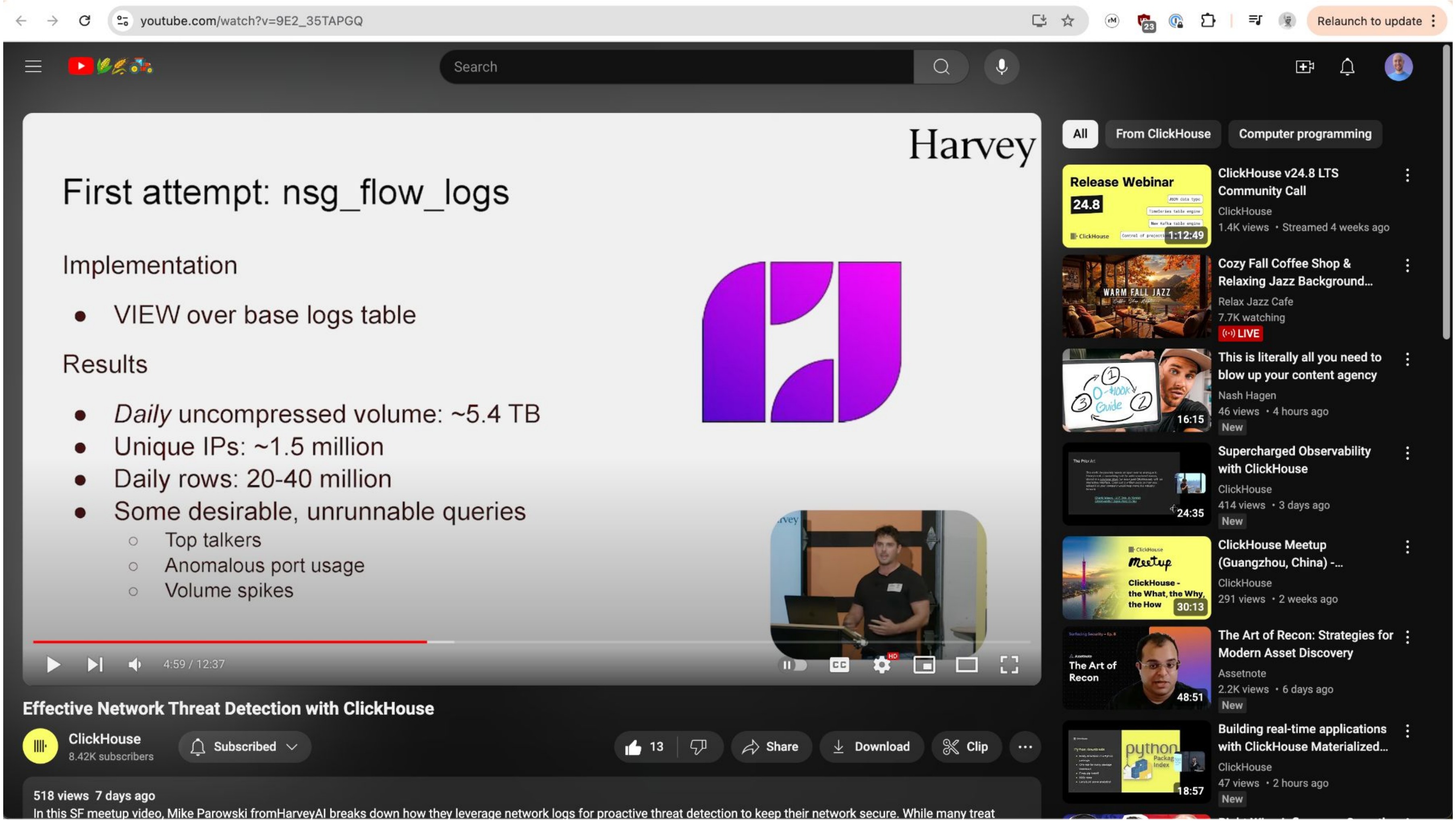
Last Event Received: 2024-05-17T20:37:06Z

Source Details

ID: 2esQmb2PgVIBQs7lFPuZpuIwVN3
Type: notion
Name: notion
webhookID: 2esQmdpUqnbqwvUgYZRKOpe4nsu
webhookURL: https://api.runreveal.com/sources/notion/webhook/2esQmdpUqnbqwvUgYZRKOpe4nsu

runreveal-keeper-test

teleport-audit-logs



Harvey

First attempt: nsg_flow_logs

Implementation

- VIEW over base logs table

Results

- *Daily* uncompressed volume: ~5.4 TB
- Unique IPs: ~1.5 million
- Daily rows: 20-40 million
- Some desirable, unrunnable queries
  - Top talkers
  - Anomalous port usage
  - Volume spikes

Effective Network Threat Detection with ClickHouse

ClickHouse
8.42K subscribers

518 views 7 days ago
In this SF meetup video, Mike Parowski fromHarveyAI breaks down how they leverage network logs for proactive threat detection to keep their network secure. While many treat

ClickHouse v24.8 LTS Community Call
ClickHouse
1.4K views • Streamed 4 weeks ago

Cozy Fall Coffee Shop & Relaxing Jazz Background...
Relax Jazz Cafe
7.7K watching
LIVE

This is literally all you need to blow up your content agency
Nash Hagen
46 views • 4 hours ago
New

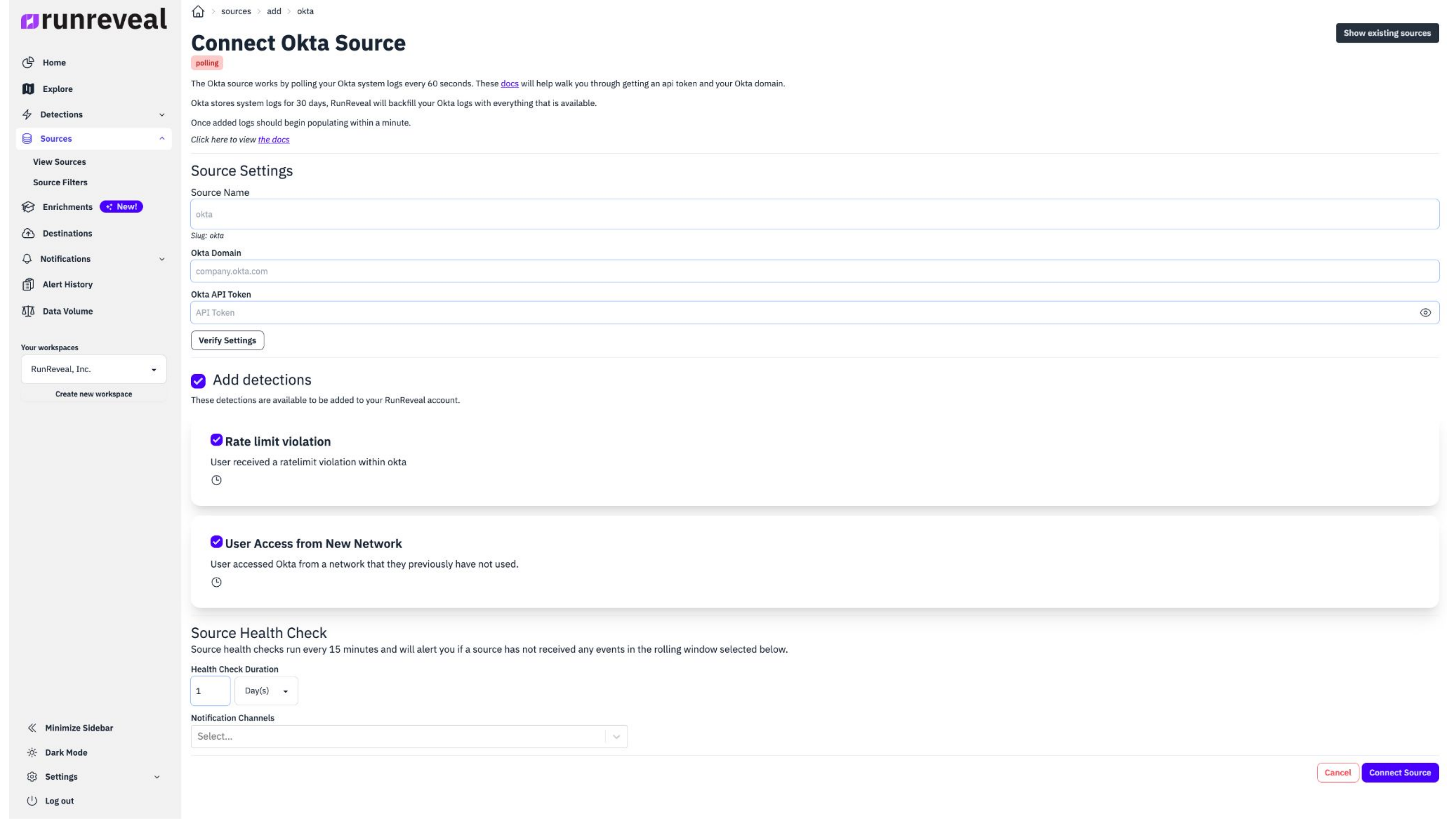
Supercharged Observability with ClickHouse
ClickHouse
414 views • 3 days ago
New

ClickHouse Meetup (Guangzhou, China) -...
ClickHouse
291 views • 2 weeks ago

The Art of Recon: Strategies for Modern Asset Discovery
Assetnote
2.2K views • 6 days ago
New

Building real-time applications with ClickHouse Materialized...
ClickHouse
47 views • 2 hours ago
New

rrsch



Connect Okta Source

polling

The Okta source works by polling your Okta system logs every 60 seconds. These docs will help walk you through getting an api token and your Okta domain.

Okta stores system logs for 30 days, RunReveal will backfill your Okta logs with everything that is available.

Once added logs should begin populating within a minute.

Click here to view the docs

Source Settings

Source Name

okta

Slug: okta

Okta Domain

company.okta.com

Okta API Token

API Token

☑ Add detections

These detections are available to be added to your RunReveal account.

☑ Rate limit violation

User received a ratelimit violation within okta

☑ User Access from New Network

User accessed Okta from a network that they previously have not used.

Source Health Check

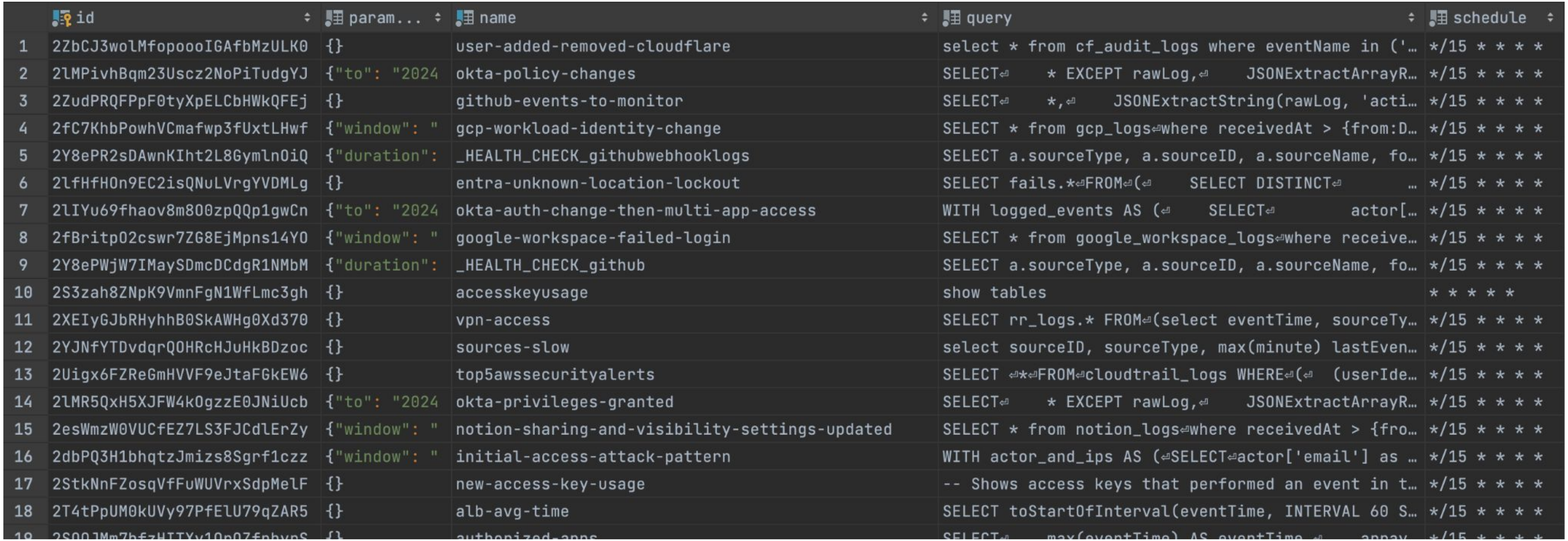
Source health checks run every 15 minutes and will alert you if a source has not received any events in the rolling window selected below.

Health Check Duration

1 Day(s)

Notification Channels

Select...



	id	param...	name	query	schedule
1	2ZbCJ3wolMfopoooIGAfbMzULK0	{}	user-added-removed-cloudflare	select * from cf_audit_logs where eventName in ('…	*/15 * * * *
2	2lMPivhBqm23Uscz2NoPiTudgYJ	{"to": "2024	okta-policy-changes	SELECT⏎ * EXCEPT rawLog,⏎ JSONExtractArrayR…	*/15 * * * *
3	2ZudPRQFPpF0tyXpELCbHWkQFEj	{}	github-events-to-monitor	SELECT⏎ *,⏎ JSONExtractString(rawLog, 'acti…	*/15 * * * *
4	2fC7KhbPowhVCmafwp3fUxtLHwf	{"window": "	gcp-workload-identity-change	SELECT * from gcp_logs⏎where receivedAt > {from:D…	*/15 * * * *
5	2Y8ePR2sDAwnKIht2L8GymlnOiQ	{"duration":	_HEALTH_CHECK_githubwebhooklogs	SELECT a.sourceType, a.sourceID, a.sourceName, fo…	*/15 * * * *
6	2lfHfHOn9EC2isQNuLVrgYVDMLg	{}	entra-unknown-location-lockout	SELECT fails.*⏎FROM⏎(⏎ SELECT DISTINCT⏎ …	*/15 * * * *
7	2lIYu69fhaov8m8O0zpQQp1gwCn	{"to": "2024	okta-auth-change-then-multi-app-access	WITH logged_events AS (⏎ SELECT⏎ actor[…	*/15 * * * *
8	2fBritpO2cswr7ZG8EjMpns14YO	{"window": "	google-workspace-failed-login	SELECT * from google_workspace_logs⏎where receive…	*/15 * * * *
9	2Y8ePWjW7IMaySDmcDCdgR1NMbM	{"duration":	_HEALTH_CHECK_github	SELECT a.sourceType, a.sourceID, a.sourceName, fo…	*/15 * * * *
10	2S3zah8ZNpK9VmnFgN1WfLmc3gh	{}	accesskeyusage	show tables	* * * * *
11	2XEIyGJbRHyhhB0SkAWHg0Xd370	{}	vpn-access	SELECT rr_logs.* FROM⏎(select eventTime, sourceTy…	*/15 * * * *
12	2YJNfYTDvdqrQOHRcHJuHkBDzoc	{}	sources-slow	select sourceID, sourceType, max(minute) lastEven…	*/15 * * * *
13	2Uigx6FZReGmHVVF9eJtaFGkEW6	{}	top5awssecurityalerts	SELECT ⏎*⏎FROM⏎cloudtrail_logs WHERE⏎(⏎ (userIde…	*/15 * * * *
14	2lMR5QxH5XJFW4kOgzzE0JNiUcb	{"to": "2024	okta-privileges-granted	SELECT⏎ * EXCEPT rawLog,⏎ JSONExtractArrayR…	*/15 * * * *
15	2esWmzW0VUCfEZ7LS3FJCdlErZy	{"window": "	notion-sharing-and-visibility-settings-updated	SELECT * from notion_logs⏎where receivedAt > {fro…	*/15 * * * *
16	2dbPQ3H1bhqtzJmizs8Sgrf1czz	{"window": "	initial-access-attack-pattern	WITH actor_and_ips AS (⏎SELECT⏎actor['email'] as …	*/15 * * * *
17	2StkNnFZosqVfFuWUVrxSdpMelF	{}	new-access-key-usage	-- Shows access keys that performed an event in t…	*/15 * * * *
18	2T4tPpUM0kUVy97PfElU79qZAR5	{}	alb-avg-time	SELECT toStartOfInterval(eventTime, INTERVAL 60 S…	*/15 * * * *
19	2SOQJMm7bfzHITXy1Op0ZfnbvpS	{}	authorized-apps	SELECT⏎ max(eventTime) AS eventTime,⏎ array…	*/15 * * * *

What are the takeaways?

- ClickHouse is great to build around
- Lots of use cases + opportunities
-