VECTRA®

Vectra Detect: Google SecOps SIEM Integration - User Guide

V 1.0.0

Contents

Overview

Vectra Platform

Vectra Detect is a cybersecurity platform designed to identify and respond to advanced cyber threats using AI-driven network detection and response (NDR). It continuously analyzes network traffic to detect suspicious behavior and potential breaches, providing real-time threat intelligence and actionable insights. By leveraging machine learning, Vectra Detect helps organizations proactively detect, investigate, and mitigate threats, enhancing security posture and reducing the time to respond. It is particularly effective in identifying hidden, sophisticated attacks that traditional security solutions may miss.

Google SecOps

Google SecOps is a cybersecurity telemetry platform for threat hunting, and threat intelligence and is part of the Google Cloud Platform. Google SecOps stores log events it receives in two formats: either as the original raw log or structured Unified Data Model (UDM) log. There are two critical elements to consider for parsing, Unified Data Model (UDM) which defines the schema for parsing, and Configuration Based Normalizers (CBN) which describes how log data is transformed to the UDM schema.

Looker Dashboards for Vectra Detect

This integration aims to enable seamless ingestion, parsing, and visualization of Vectra network intelligence data within Google SecOps SIEM. This integration will allow Google SecOps SIEM to receive real-time detections, hosts, accounts, health, audit, and lockdown data from Vectra using Vectra API , enriching the SIEM’s threat detection and response capabilities with comprehensive network data.

Release Notes

V1.0.0

- Provided the parser that processes data ingested from the Vectra platform and converts it into the Google SecOps UDM data model.
- Provided below dashboards for visualization
  - Entities
  - Detection
  - Audit
  - Health
  - Lockdown
  - Match

App Installation & Configuration

Pre-Requisites

- Vectra Platform
- Google SecOps
- Looker Instance

Configure Vectra Detect to Export Logs in JSON Format

The newly developed Vectra parser is specifically optimized to support logs ingested in **JSON format**. This allows for accurate parsing and proper mapping of fields to the **Unified Data Model** (UDM)—which powers dashboards, analytics, and threat detection capabilities within Google SecOps.

Although the parser includes fallback support for non-JSON formats (such as CEF) to accommodate existing customer configurations, these formats are **not officially supported going forward**. Logs in unsupported formats may result in **incomplete parsing**, limited field extraction, and **degraded dashboard and analytics functionality**.

To ensure the best possible experience with the Vectra integration for Google SecOps, we **strongly recommend configuring your log export to use the JSON format**.

Follow the steps below to apply the configuration change.

1. In the Vectra Detect UI, navigate to, **Settings** > **Notifications**.
2. Scroll to the Syslog configuration section at the bottom of the page and click **Edit**.
3. Configure the following:
    a. **IP** address of your Syslog server
    b. **Port** number and **Protocol**
4. Under the **Format** dropdown, select **JSON**.
5. Choose the log types you wish to export.
6. Click **Save** to apply the changes.

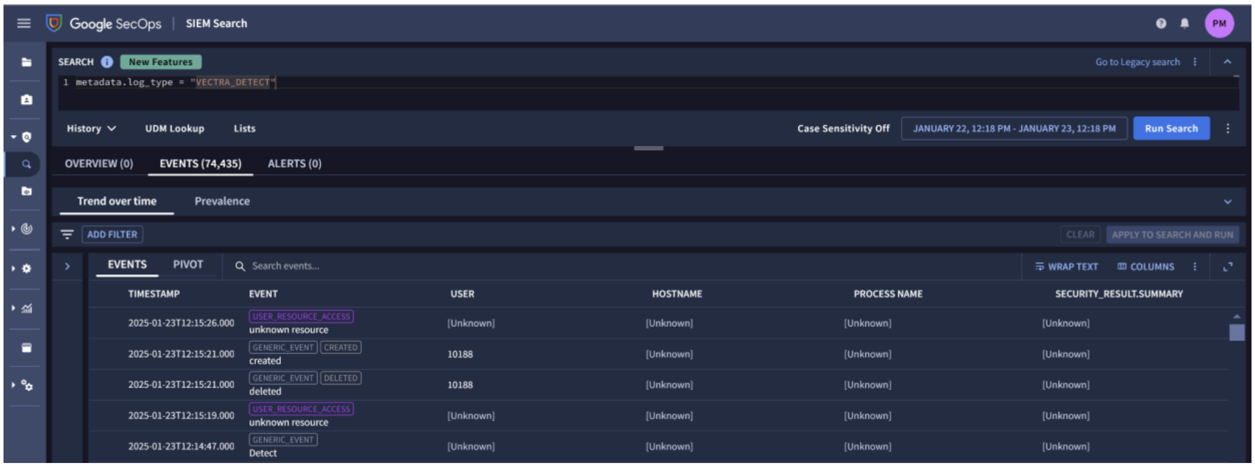
Vectra Google SecOps Forwarder

1. Setup Google Security Operations Forwarder
   a. Users must first install and configure the Google Security Operations forwarder for **Vectra Detect** log type in their environment.
   b. Refer to the below guides for detailed setup instructions.
      i. Forwarder Configuration from UI.
      ii. Install and Configure the Forwarder.
2. Configure Vectra for Syslog Forwarding
   a. Once the Google SecOps Syslog forwarder is configured, Vectra Administrator users can enable Vectra to send the host and Account scoring information, detection details, and audit logs over syslog to external collectors for further storage and analysis.

Refer to the Vectra Syslog Guide for step-by-step configuration details.

View Events in Google SecOps

1. Log in to Google SecOps:
   a. Open a web browser and navigate to the Google SecOps instance URL. For example: https://test.backstory.chronicle.security/
   b. Replace test with your actual Google SecOps instance name.
2. Access SIEM Search:
   a. From the top left corner of the Google SecOps console, select the "Investigation" option.
   b. Within the Investigation section, choose "SIEM Search".
3. Filter Events by Log Type:
   a. In the SIEM Search interface, locate the "UDM Search" section.
   b. Apply a filter for the metadata field "log_type". Set the filter value to metadata.log_type="VECTRA_DETECT".
4. View Vectra Events:
   a. The SIEM Search results will display Vectra events within the "Events" section.



```
metadata.log_type = "VECTRA_DETECT"
```

Case Sensitivity Off

JANUARY 22, 12:18 PM - JANUARY 23, 12:18 PM

TIMESTAMP	EVENT	USER	HOSTNAME	PROCESS NAME	SECURITY_RESULT.SUMMARY
2025-01-23T12:15:26.000	USER_RESOURCE_ACCESS unknown resource	[Unknown]	[Unknown]	[Unknown]	[Unknown]
2025-01-23T12:15:21.000	GENERIC_EVENT CREATED created	10188	[Unknown]	[Unknown]	[Unknown]
2025-01-23T12:15:21.000	GENERIC_EVENT DELETED deleted	10188	[Unknown]	[Unknown]	[Unknown]
2025-01-23T12:15:19.000	USER_RESOURCE_ACCESS unknown resource	[Unknown]	[Unknown]	[Unknown]	[Unknown]
2025-01-23T12:14:47.000	GENERIC_EVENT Detect	[Unknown]	[Unknown]	[Unknown]	[Unknown]

Looker Installation & Configuration

Pre-Requisites

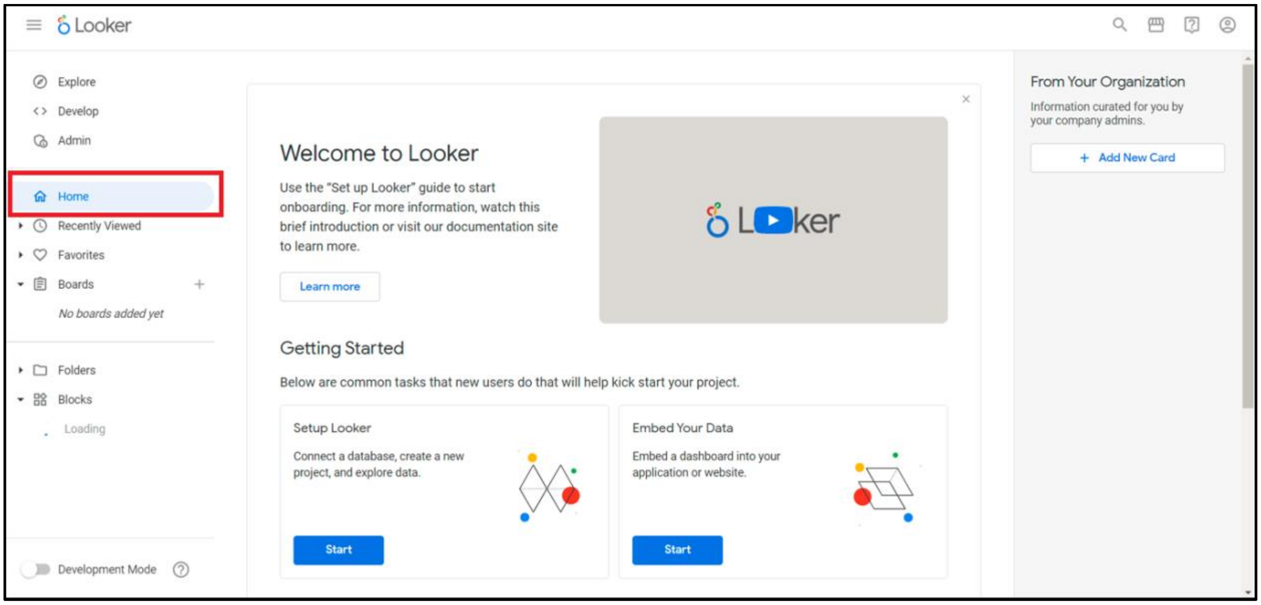
- Billing Project ID, Dataset name, and Service account file of BigQuery that stores Google SecOps data for database connection in Looker.
- BigQuery Export feature needs to be enabled for your Google SecOps tenant. (Reach out to your Google SecOps representative to set this up.)

User Permissions

- Admin Role User - to create database connections and install blocks from the marketplace.

Create a connection to Google SecOps in Looker

1. To create a connection to Google SecOps, first open the Looker instance and navigate to the Home page.



Welcome to Looker

Use the "Set up Looker" guide to start onboarding. For more information, watch this brief introduction or visit our documentation site to learn more.

Getting Started

Below are common tasks that new users do that will help kick start your project.

Setup Looker

Connect a database, create a new project, and explore data.

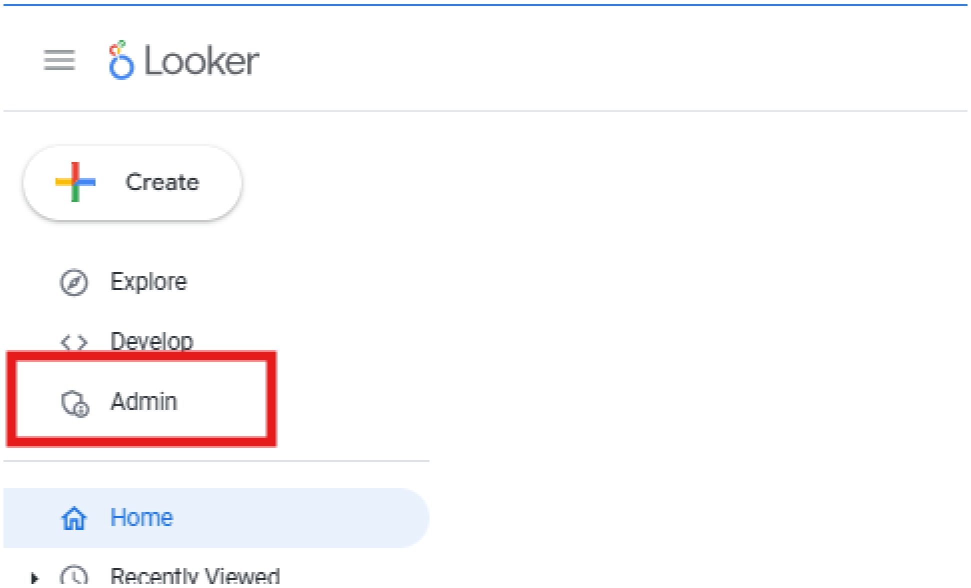
Embed Your Data

Embed a dashboard into your application or website.

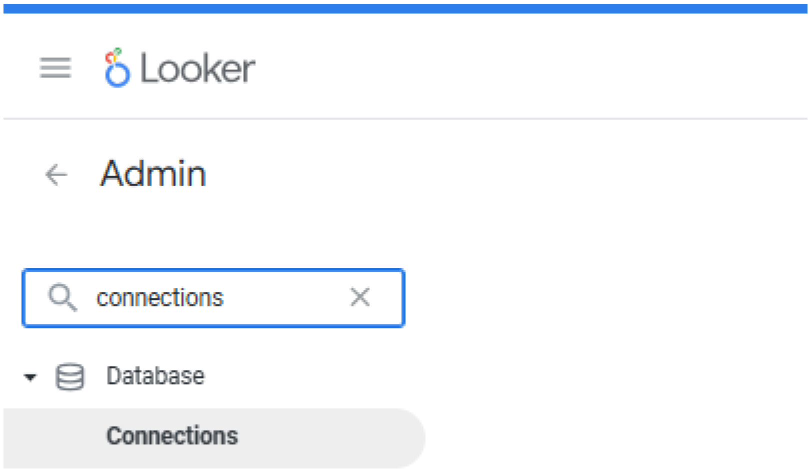
From Your Organization

Information curated for you by your company admins.

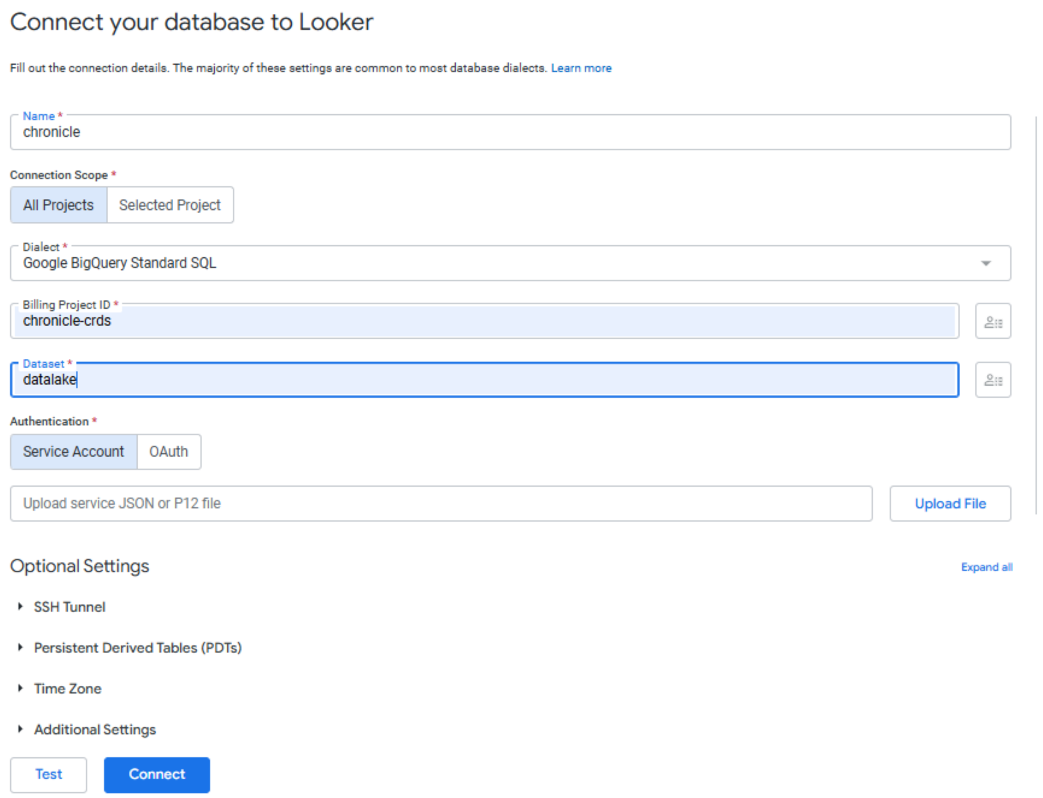
2. Now click on the **Admin** from the main menu (in left panel)



3. Now type **Connections** in the search, once the Connections option appears click on it to see the connection page.



4. Now click on the **Add connection**(Add Connection) to create a new connection and name it as **chronicle**.
5. Select **Google BigQuery Standard SQL** in the Dialect. Now several new fields will appear.
6. Enter Billing Project ID field. Example: “**chronicle-crds**” here, where Chronicle data is present.
7. Enter the **datalake** in the Dataset name.



Connect your database to Looker

Fill out the connection details. The majority of these settings are common to most database dialects. Learn more

Name * chronicle

Connection Scope * All Projects | Selected Project

Dialect * Google BigQuery Standard SQL

Billing Project ID * chronicle-crds

Dataset * datalake

Authentication * Service Account | OAuth

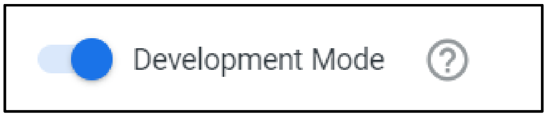
Upload service JSON or P12 file

Optional Settings

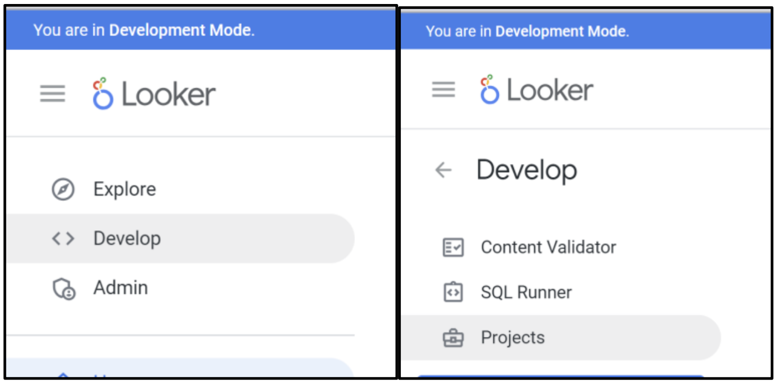
8. To configure authentication, select the service account method and upload your Chronicle service account file.
9. In the optional settings, set both the timestamps (Database timestamp and query timestamp) as UTC (the time fields shown in dashboards will be populated accordingly).
10. Click on Test to check the connectivity of Looker with Google Chronicle database.
11. Click on the Connect button (Connect) to complete the connection setup. Looker is now connected to the Google Chronicle database.

Get the Block from GitHub Repository

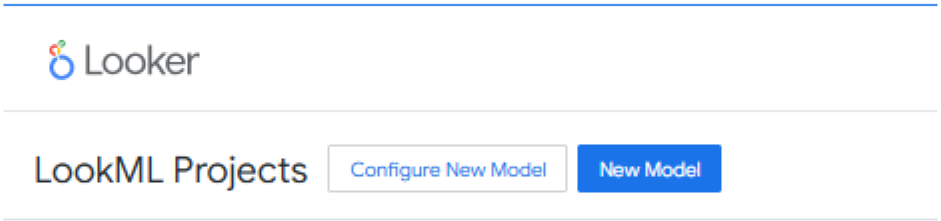
1. Go to Vectra looker dashboard github repository and fork it. Make sure to uncheck the option to fork only the **vectra_detect_dashboards** branch.
2. Go to Looker and turn on "Development Mode" from the sidebar panel.



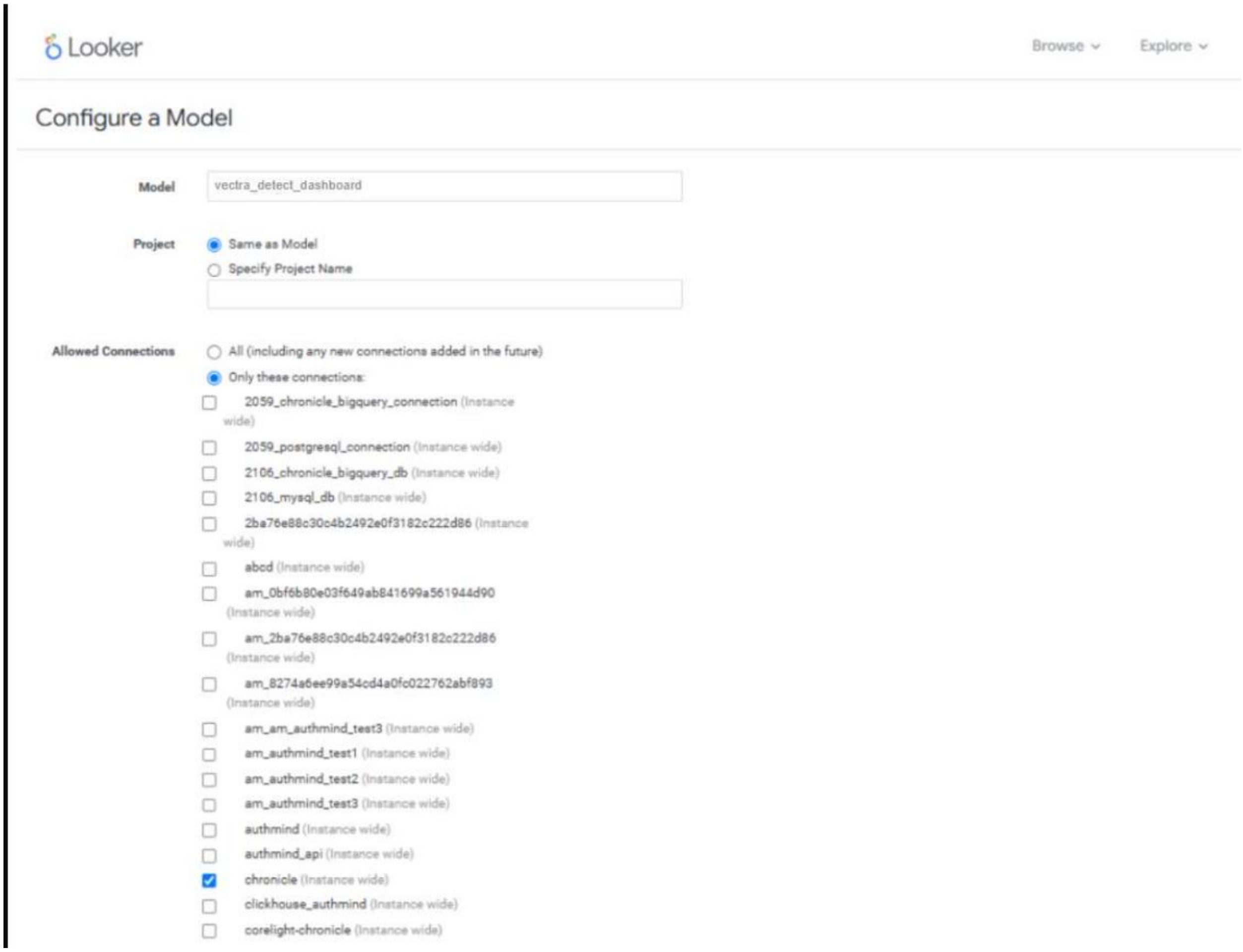
3. Select Projects from the Develop menu.



4. From the LookML Projects page, select **Configure New Model** to open the model configuration page.



5. Enter the model name as **vectra_detect_dashboards** and keep the **Same as Model** option selected.



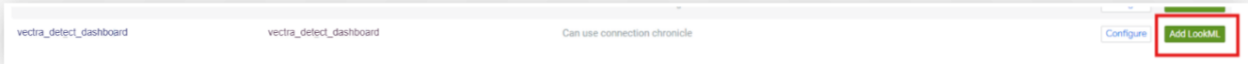
Configure a Model

Model: vectra_detect_dashboard

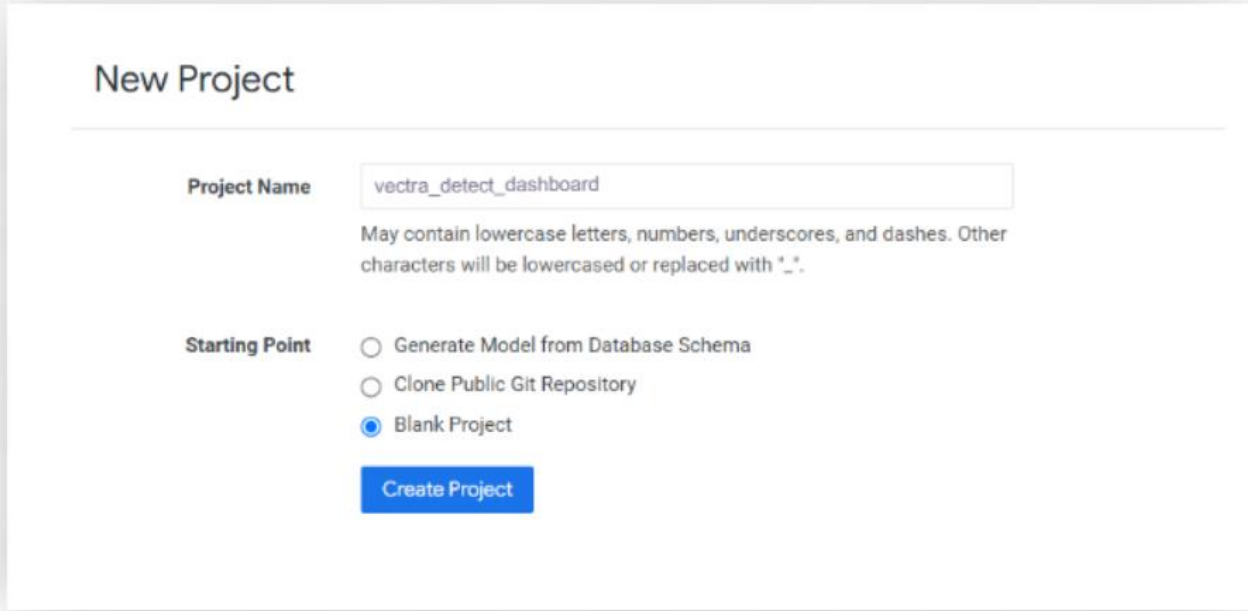
Project:
● Same as Model
○ Specify Project Name

Allowed Connections:
○ All (including any new connections added in the future)
● Only these connections:
□ 2059_chronicle_bigquery_connection (Instance wide)
□ 2059_postgresql_connection (Instance wide)
□ 2106_chronicle_bigquery_db (Instance wide)
□ 2106_mysql_db (Instance wide)
□ 2ba76e88c30c4b2492e0f3182c222d86 (Instance wide)
□ abcd (Instance wide)
□ am_0bf6b80e03f649ab841699a561944d90 (Instance wide)
□ am_2ba76e88c30c4b2492e0f3182c222d86 (Instance wide)
□ am_8274a6ee99a54cd4a0fc022762abf893 (Instance wide)
□ am_am_authmind_test3 (Instance wide)
□ am_authmind_test1 (Instance wide)
□ am_authmind_test2 (Instance wide)
□ am_authmind_test3 (Instance wide)
□ authmind (Instance wide)
□ authmind_api (Instance wide)
☑ chronicle (Instance wide)
□ clickhouse_authmind (Instance wide)
□ corelight-chronicle (Instance wide)

6. Click on **Save** to save the configuration.
7. After saving the configuration you'll be redirected to the **Projects** page and you'll find your project in the **Pending Project** section.
8. Click on **Add LookML** to configure your project and add the lookML file in it.


vectra_detect_dashboard   vectra_detect_dashboard   Can use connection chronicle

9. On the New Project page, configure these options for your new project:
   Project Name: Give project name **vectra_detect_dashboards**.
   Starting Point: Select Blank Project.
   Click on Create Project. The project will be created and opened in the Looker IDE.



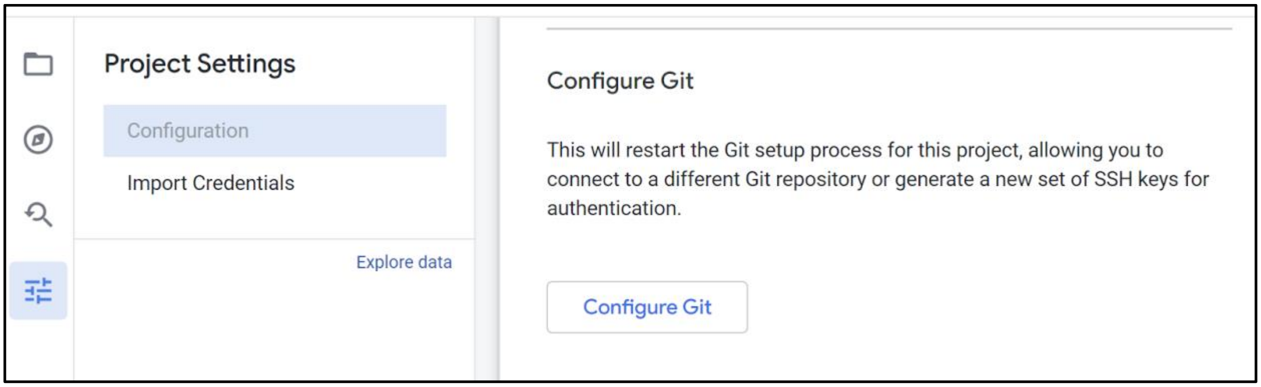
New Project

Project Name: vectra_detect_dashboard

May contain lowercase letters, numbers, underscores, and dashes. Other characters will be lowercased or replaced with "_".

Starting Point:
○ Generate Model from Database Schema
○ Clone Public Git Repository
● Blank Project

10. Click on the Settings icon from the navigation bar, and open the Configure Git page by selecting the Configure Git button.

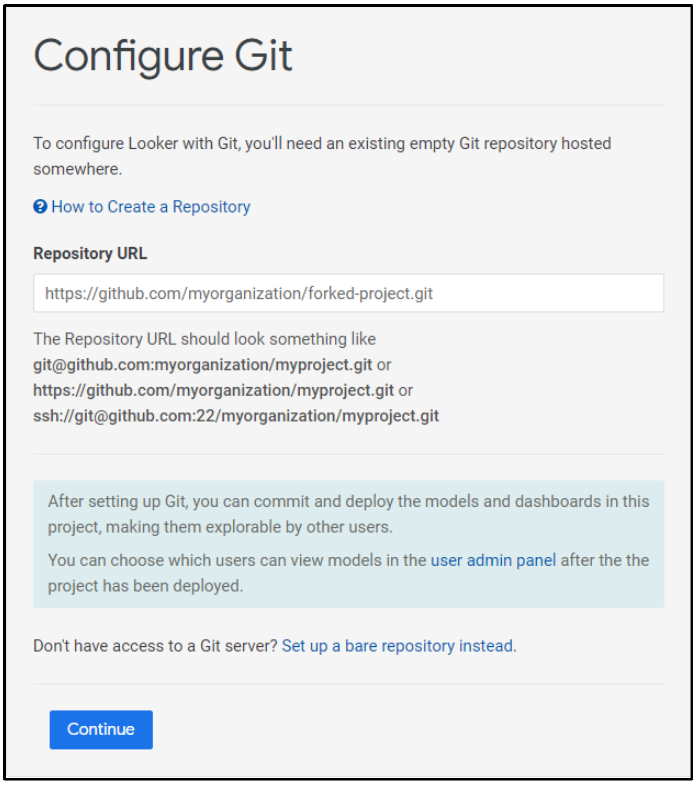


Configure Git

This will restart the Git setup process for this project, allowing you to connect to a different Git repository or generate a new set of SSH keys for authentication.

11. In Looker's Configure Git section, paste the URL of the forked Vectra Looker Dashboard Git Repository in the Repository URL field, then select Continue.

    e.g. https://github.com/<your_username>/looker-dashboards.git



Configure Git

To configure Looker with Git, you'll need an existing empty Git repository hosted somewhere.

How to Create a Repository

Repository URL

https://github.com/myorganization/forked-project.git

The Repository URL should look something like **git@github.com:myorganization/myproject.git** or **https://github.com/myorganization/myproject.git** or **ssh://git@github.com:22/myorganization/myproject.git**

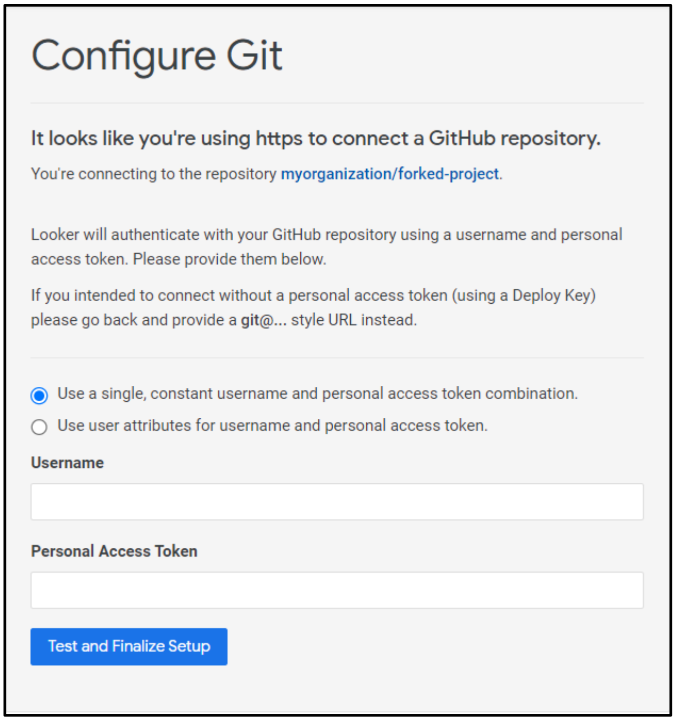
After setting up Git, you can commit and deploy the models and dashboards in this project, making them explorable by other users.

You can choose which users can view models in the user admin panel after the the project has been deployed.

Don't have access to a Git server? Set up a bare repository instead.

12. Enter the github username and Personal Access Token, then click “Test and Finalize Setup”.

Note: Make sure the **Personal Access Token (PAT)** you created from your github repository has **Read/Write** permissions of the repository.



Configure Git

It looks like you're using https to connect a GitHub repository.

You're connecting to the repository myorganization/forked-project.

Looker will authenticate with your GitHub repository using a username and personal access token. Please provide them below.

If you intended to connect without a personal access token (using a Deploy Key) please go back and provide a **git@...** style URL instead.

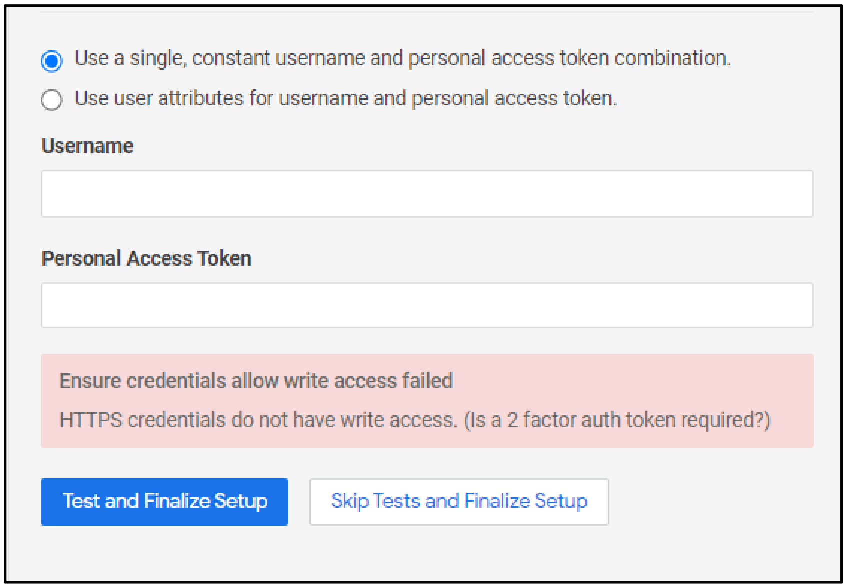
● Use a single, constant username and personal access token combination.

○ Use user attributes for username and personal access token.

Username

Personal Access Token

13. If you get an error like “Ensure credential allow write access failed”, just enter the username and token again and click “Skip Tests and Finalize Setup”.



● Use a single, constant username and personal access token combination.

○ Use user attributes for username and personal access token.

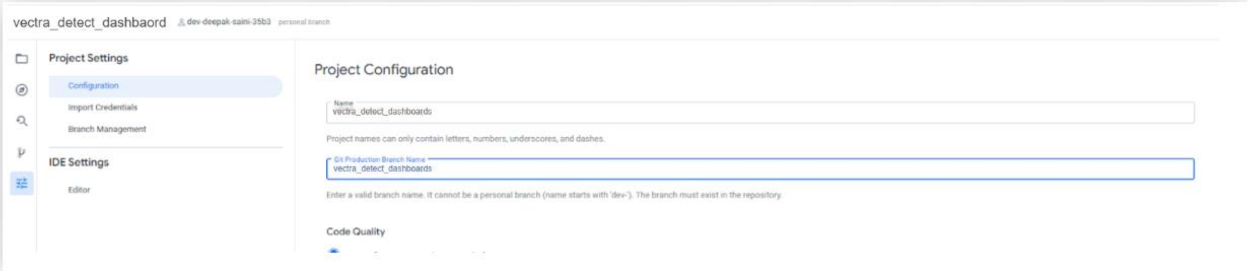
Username

Personal Access Token

Ensure credentials allow write access failed

HTTPS credentials do not have write access. (Is a 2 factor auth token required?)

14. Once the git is configured, open the project settings and change the production branch to **vectra_detect_dashboards**



vectra_detect_dashbaord dev-deepak-saini-35b3 personal branch

Project Configuration

Name: vectra_detect_dashboards

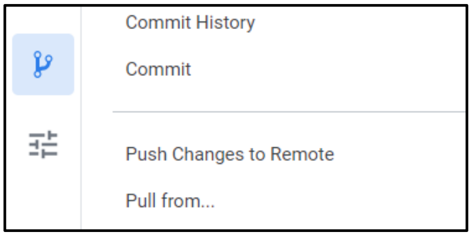
Project names can only contain letters, numbers, underscores, and dashes.

Git Production Branch Name: vectra_detect_dashboards

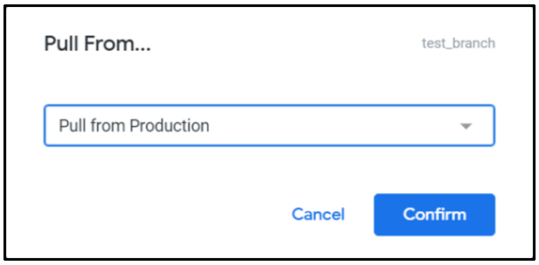
Enter a valid branch name. It cannot be a personal branch (name starts with 'dev-'). The branch must exist in the repository.

15. Now, you should be able to see the code in your project from the **vectra_detect_dashboards** branch. If not then,

    a. In the ‘Git Actions’ tab from the left side, click on the “Pull from…” option.

    b. Select the “Pull From Production” option and click on the Confirm button.

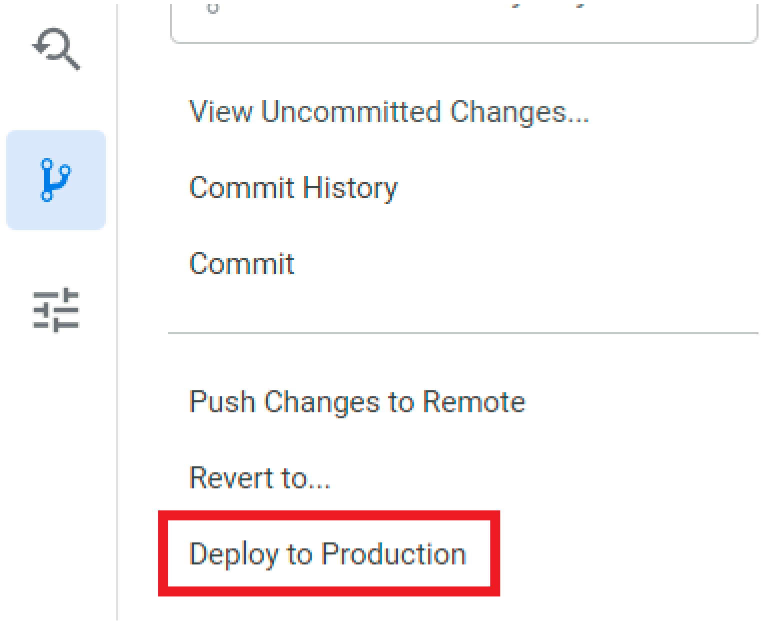
    Pull From...

    test_branch

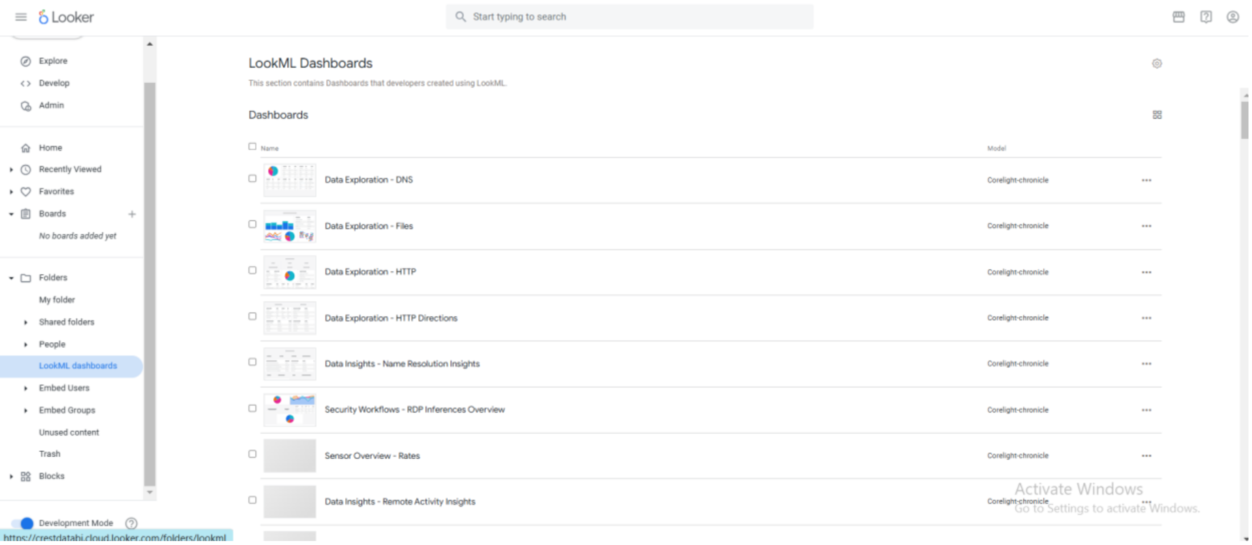
    Pull from Production

16. Any changes can be committed and pushed to the dev branch by clicking on **Validate LookML** and once the files are validated click on **Commit and Push changes**.
17. After the above steps, In the Git Actions, click on the “Deploy to Production” or you can press “Deploy to Production” from the top right corner.

    Note: ‘Deploy to Production’ will push code to the production branch that is set in the project settings. By default, it will be the **vectra_detect_dashboards** branch. If you don’t want to push code to the ‘main’ branch, then create your own branch and set it to ‘Git Production Branch Name’ in project settings. Then click on Deploy to Production.



18. Now you can turn off the development mode in order to see the LookML dashboards.
19. On the Homepage of your Looker instance, navigate to the "LookML dashboards" tab under the "Folders" tab to access and view all the dashboards.



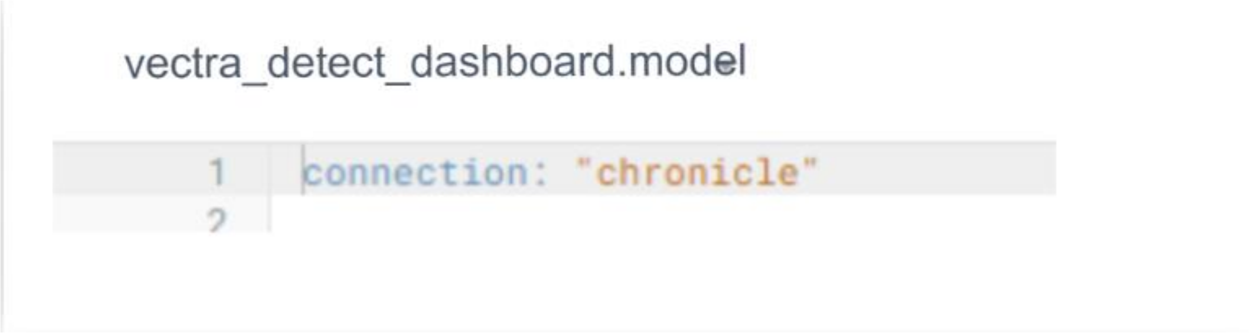
LookML Dashboards

This section contains Dashboards that developers created using LookML.

Dashboards

Name	Model
Data Exploration - DNS	Corelight-chronicle
Data Exploration - Files	Corelight-chronicle
Data Exploration - HTTP	Corelight-chronicle
Data Exploration - HTTP Directions	Corelight-chronicle
Data Insights - Name Resolution Insights	Corelight-chronicle
Security Workflows - RDP Inferences Overview	Corelight-chronicle
Sensor Overview - Rates	Corelight-chronicle
Data Insights - Remote Activity Insights	Corelight-chronicle

20. The connection name defined at the top of the **vectra_detect_dashboards.model** file must match the connection name created earlier. If the user has named it chronicle, no changes are necessary. Otherwise, the connection field needs to be updated with the correct connection name.



vectra_detect_dashboard.model

```
connection: "chronicle"
```

Dashboards

Entities Dashboard

This dashboard displays data for the "Scoring" log type. The table panel presents details about each incident, including their most recent update status.

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Scoring.

Timerange

- This filter updated the panel based on the time range selected in it. **Default:** Last 7 days.

Entity Type

- Filters the Panels according to the selected entity type **i.e.** account or host. **Default**: all.

Severity

- Filters the Panels according to the selected priority. It has the following values Low, Medium, High, Critical and Unknown **Default**: all

Data Source

- Filters the Panels according to the selected data source type. It has the following values AWS, O365, M365, SAML and Network **Default**: all.

Assignment

- Filters the Panels according to the selected entity type **i.e.** Assigned or Unassigned. **Default**: all.

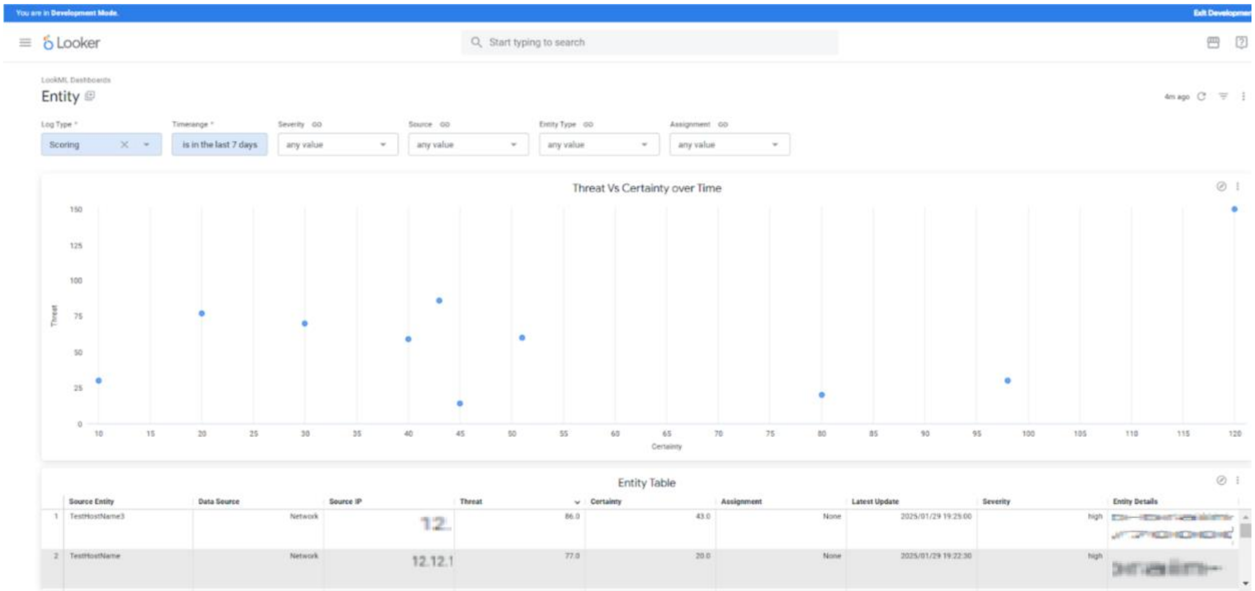
2. Panel description as per label

Threat Vs Certainty over Time

- This panel displays the data point on the scatter chart based on threat and certainty of an incident, which were collected from the Vectra Detect platform sent to Google SecOps.

Entities Table

- The table panel displays the incidents based on their most recent updates, which were collected from the Vectra Detect platform sent to Google SecOps.



Log Type *: Scoring
Timerange *: is in the last 7 days
Severity: any value
Source: any value
Entity Type: any value
Assignment: any value

	Source Entity	Data Source	Source IP	Threat	Certainty	Assignment	Latest Update	Severity	Entity Details
1	TestHostName3	Network	12.	86.0	43.0	None	2025/01/29 19:25:00	high	[redacted]
2	TestHostName	Network	12.12.1	77.0	20.0	None	2025/01/29 19:22:30	high	[redacted]

Detection Dashboard

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Detection.

Timerange

- This filter updates the panel based on the time range selected in it. **Default:** Last 7 days.

Source

- Filters the Panels according to the selected data source type. Source filter have the following values AWS, O365, M365, SAML and Network **Default**: all.

Behaviour

- Filters the Panels according to the selected behaviour. **Default**: all.

Detection Category

- Filters the Panels according to the selected category. **Default**: all.

Entity Type

- Filters the Panels according to the selected entity type **i.e.** account or host. **Default**: all.

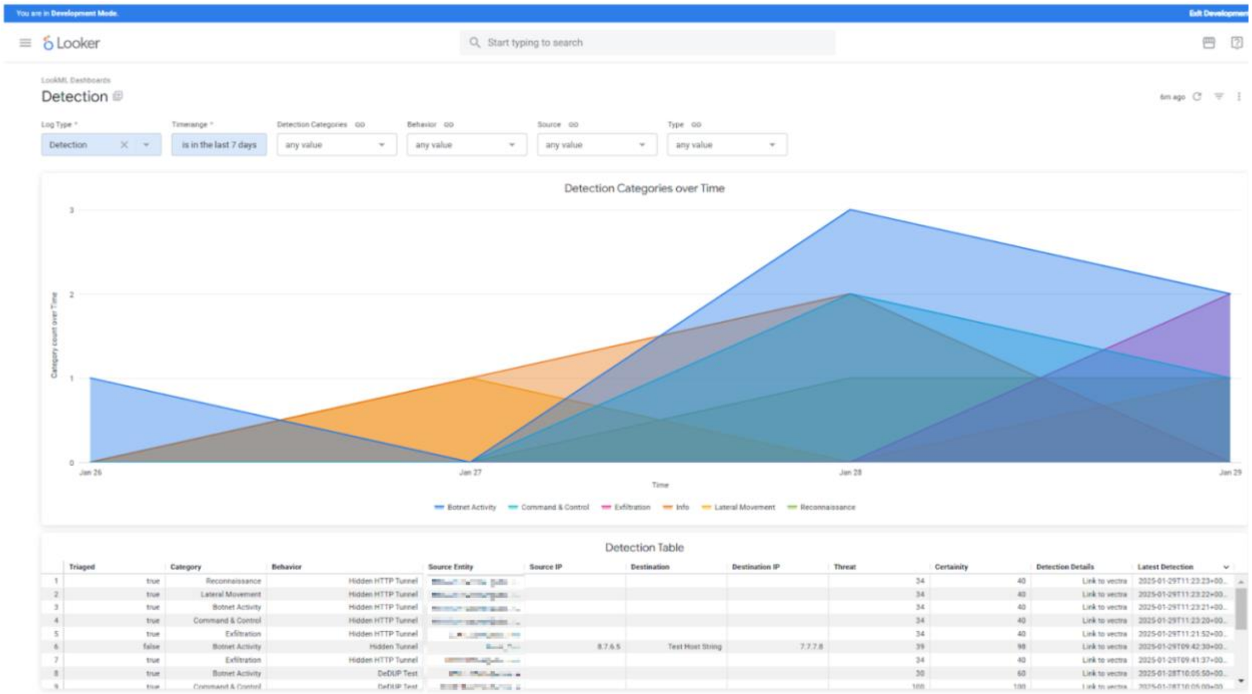
2. Panel description as per label

Detection Categories over Time

- This displays an area chart for the count of the categories over time of Detection log.

Detection List

- The table panel displays the incidents of detection logs based on their most recent updates, which were collected from the Vectra Detect platform sent to Google SecOps.



Detection

Log Type *: Detection
Timerange *: is in the last 7 days
Detection Categories: any value
Behavior: any value
Source: any value
Type: any value

Detection Table

	Triaged	Category	Behavior	Source Entity	Source IP	Destination	Destination IP	Threat	Certainity	Detection Details	Latest Detection
1	true	Reconnaissance	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T11:23:23+00...
2	true	Lateral Movement	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T11:23:22+00...
3	true	Botnet Activity	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T11:23:21+00...
4	true	Command & Control	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T11:23:20+00...
5	true	Exfiltration	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T11:21:52+00...
6	false	Botnet Activity	Hidden Tunnel	[redacted]	8.7.6.5	Test Host String	7.7.7.8	39	98	Link to vectra	2025-01-29T09:42:30+00...
7	true	Exfiltration	Hidden HTTP Tunnel	[redacted]				34	40	Link to vectra	2025-01-29T09:41:37+00...
8	true	Botnet Activity	DeDUP Test	[redacted]				30	60	Link to vectra	2025-01-28T10:05:50+00...
9	true	Command & Control	DeDUP Test	[redacted]				100	100	Link to vectra	2025-01-28T10:05:00+00...

Audit Dashboard

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Audit.

Timerange

- This filter updated the panel based on the time range selected in it. **Default:** Last 7 days.

User

- Filters the Panels according to the selected user. **Default**: all.

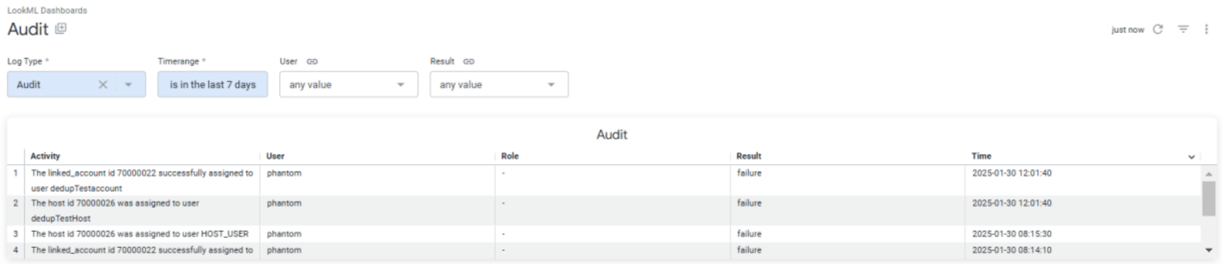
Result

- Filters the Panels according to the selected result. **Default**: all.

2. Panel description as per label

Audit Table

- The table panel displays the incidents of audit logs based on their most recent updates, which were collected from the Vectra Detect platform sent to Google SecOps.



Log Type *: Audit

Timerange *: is in the last 7 days

User
any value

Result: any value

Audit

	Activity	User	Role	Result	Time
1	The linked_account id 70000022 successfully assigned to user dedupTestaccount	phantom	-	failure	2025-01-30 12:01:40
2	The host id 70000026 was assigned to user dedupTestHost	phantom	-	failure	2025-01-30 12:01:40
3	The host id 70000026 was assigned to user HOST_USER	phantom	-	failure	2025-01-30 08:15:30
4	The linked_account id 70000022 successfully assigned to	phantom	-	failure	2025-01-30 08:14:10

Lockdown Dashboard

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Lockdown.

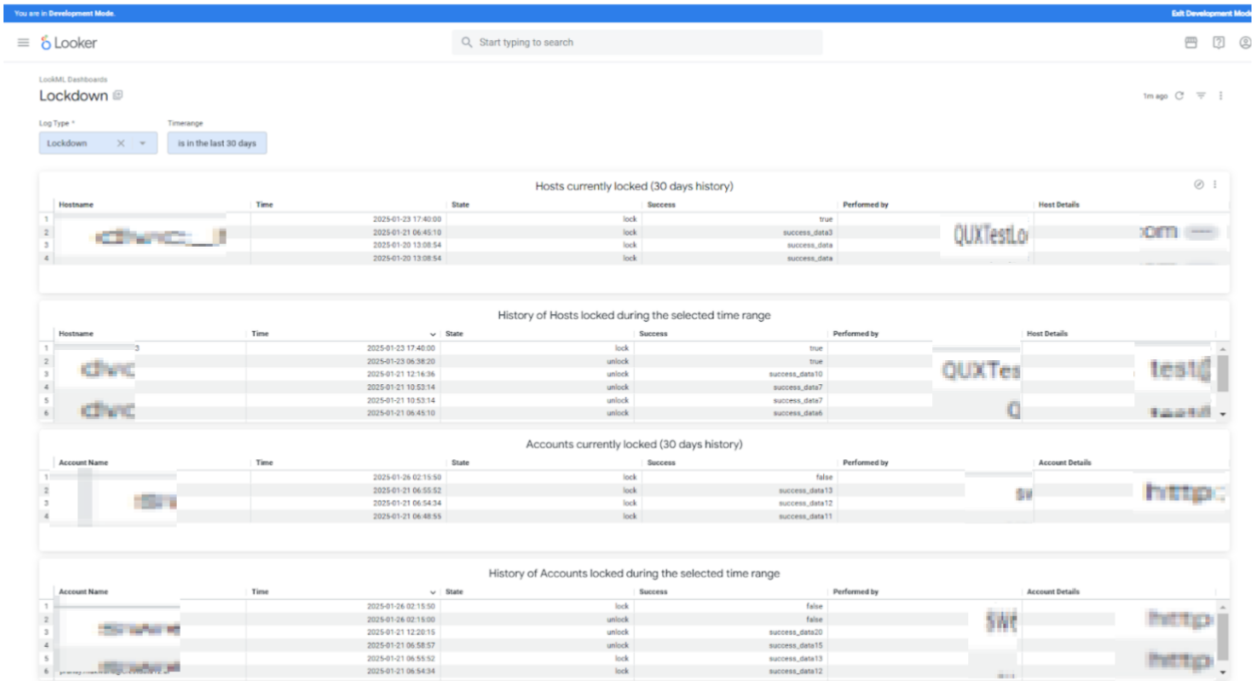
Timerange

- This filter updated the panel based on the time range selected in it. **Default:** Last 7 days.

2. Panel description as per label

Lockdown

- The table panel displays the incidents of lockdown logs based on their most recent updates, which were collected from the Vectra Detect platform sent to Google SecOps. It shows whether the incident is still locked or not.



Hosts currently locked (30 days history)

	Hostname	Time	State	Success	Performed by	Host Details
1	[redacted]	2025-01-23 17:40:00	lock	true	[redacted]	[redacted]
2		2025-01-21 06:45:10	lock	success_data3		
3		2025-01-20 13:08:54	lock	success_data		
4		2025-01-20 13:08:54	lock	success_data		

History of Hosts locked during the selected time range

	Hostname	Time	State	Success	Performed by	Host Details
1	[redacted]	2025-01-23 17:40:00	lock	true	[redacted]	[redacted]
2		2025-01-23 06:38:20	unlock	true		
3		2025-01-21 12:16:36	unlock	success_data10		
4		2025-01-21 10:53:14	unlock	success_data7		
5	[redacted]	2025-01-21 10:53:14	unlock	success_data7		[redacted]
6		2025-01-21 06:45:10	unlock	success_data6	[redacted]	

Accounts currently locked (30 days history)

	Account Name	Time	State	Success	Performed by	Account Details
1	[redacted]	2025-01-26 02:15:50	lock	false	[redacted]	[redacted]
2		2025-01-21 06:55:52	lock	success_data13		
3		2025-01-21 06:54:34	lock	success_data12		
4		2025-01-21 06:48:55	lock	success_data11		

History of Accounts locked during the selected time range

	Account Name	Time	State	Success	Performed by	Account Details
1	[redacted]	2025-01-26 02:15:50	lock	false	[redacted]	[redacted]
2		2025-01-26 02:15:00	unlock	false		
3		2025-01-21 12:20:15	unlock	success_data20		
4		2025-01-21 06:58:57	unlock	success_data15		
5	[redacted]	2025-01-21 06:55:52	lock	success_data13		[redacted]
6		2025-01-21 06:54:34	lock	success_data12	[redacted]	

Health Dashboard

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Health.

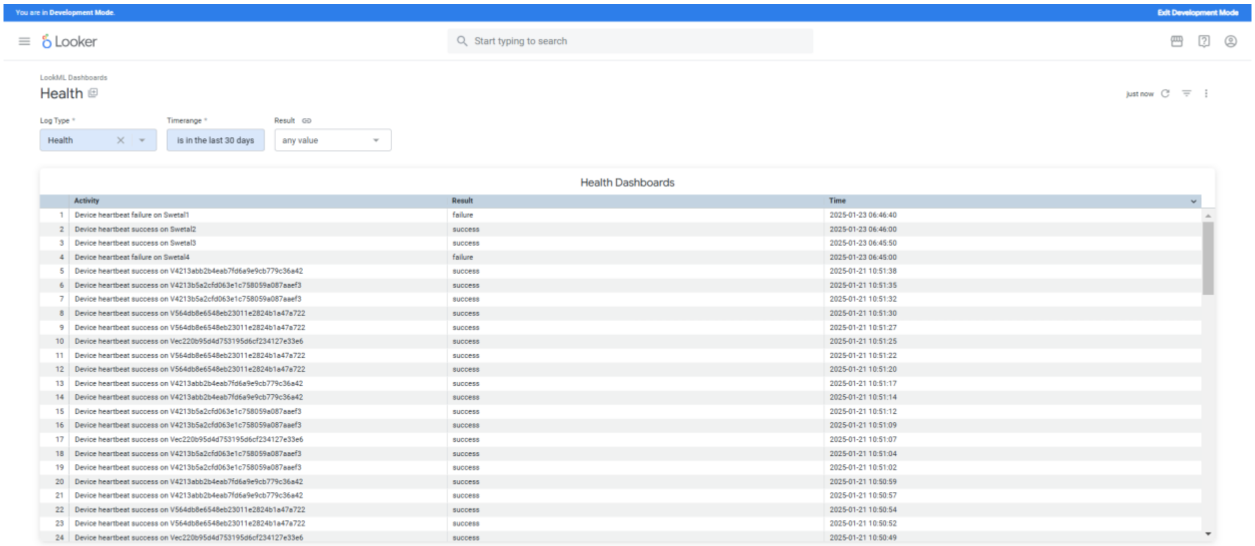
Timerange

- This filter updated the panel based on the time range selected in it. **Default:** Last 7 days.

2. Panel description as per label

Health Dashboard

- This panel displays the alerts/incidents of log type health.



Health

Log Type * Health

Timerange * is in the last 30 days

Result any value

Health Dashboards

	Activity	Result	Time
1	Device heartbeat failure on Swetal1	failure	2025-01-23 06:46:40
2	Device heartbeat success on Swetal2	success	2025-01-23 06:46:00
3	Device heartbeat success on Swetal3	success	2025-01-23 06:45:50
4	Device heartbeat failure on Swetal4	failure	2025-01-23 06:45:00
5	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:38
6	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:35
7	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:32
8	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:30
9	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:27
10	Device heartbeat success on Vec220b95d4d753195d6cf234127e33e6	success	2025-01-21 10:51:25
11	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:22
12	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:20
13	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:17
14	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:14
15	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:12
16	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:09
17	Device heartbeat success on Vec220b95d4d753195d6cf234127e33e6	success	2025-01-21 10:51:07
18	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:04
19	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:02
20	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:50:59
21	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:50:57
22	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:50:54
23	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:50:52
24	Device heartbeat success on Vec220b95d4d753195d6cf234127e33e6	success	2025-01-21 10:50:49

Match Dashboard

1. Filters description as per label

Log type

- This filter contains a single value for the dashboard, and other filters will update based on the selection made in the **Log Type** filter. **Default**: Match.

Timerange

- This filter updated the panel based on the time range selected in it. **Default:** Last 7 days.

Direction

- Filters the Panels according to the selected entity type **i.e.** Server to Client or Client to Server. **Default**: all.

Top Protocols

- This filter updated the panel based on the Protocol selected in it

Source IP

- This filter updated the panel based on the Source IP selected in it

Destination IP

- This filter updated the panel based on the Destination IP selected in it.

2. Panel description as per label

Top Signature

- This panel displays a **Pie chart** of **signatures** based on the top 10 highest counts.

Top Sources

- This panel displays a **Pie chart** of **Source IP** based on the top 10 highest counts.

Top Destinations

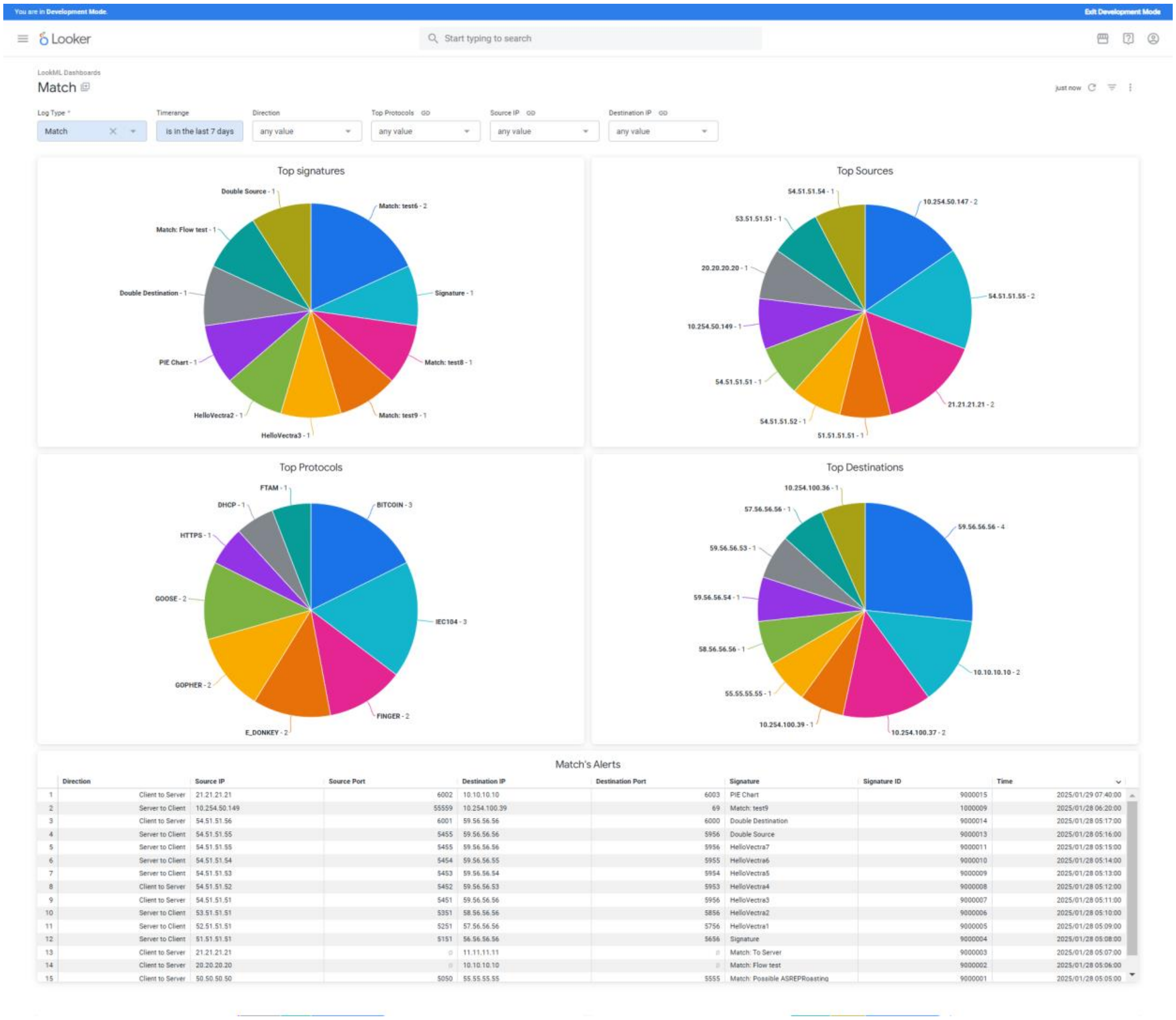
- This panel displays a **Pie chart** of **Destination IP** based on the top 10 highest counts.

Top Protocols

- This panel displays a **Pie chart** of **Protocols** based on the top 10 highest counts.

Match’s Alerts

- This panel displays the alerts of Match log type.



Match's Alerts

	Direction	Source IP	Source Port	Destination IP	Destination Port	Signature	Signature ID	Time
1	Client to Server	21.21.21.21	6002	10.10.10.10	6003	PIE Chart	9000015	2025/01/29 07:40:00
2	Server to Client	10.254.50.149	55559	10.254.100.39	69	Match: test9	1000009	2025/01/28 06:20:00
3	Client to Server	54.51.51.56	6001	59.56.56.56	6000	Double Destination	9000014	2025/01/28 05:17:00
4	Server to Client	54.51.51.55	5455	59.56.56.56	5956	Double Source	9000013	2025/01/28 05:16:00
5	Server to Client	54.51.51.55	5455	59.56.56.56	5956	HelloVectra7	9000011	2025/01/28 05:15:00
6	Server to Client	54.51.51.54	5454	59.56.56.55	5955	HelloVectra6	9000010	2025/01/28 05:14:00
7	Server to Client	54.51.51.53	5453	59.56.56.54	5954	HelloVectra5	9000009	2025/01/28 05:13:00
8	Client to Server	54.51.51.52	5452	59.56.56.53	5953	HelloVectra4	9000008	2025/01/28 05:12:00
9	Client to Server	54.51.51.51	5451	59.56.56.56	5956	HelloVectra3	9000007	2025/01/28 05:11:00
10	Server to Client	53.51.51.51	5351	58.56.56.56	5856	HelloVectra2	9000006	2025/01/28 05:10:00
11	Server to Client	52.51.51.51	5251	57.56.56.56	5756	HelloVectra1	9000005	2025/01/28 05:09:00
12	Server to Client	51.51.51.51	5151	56.56.56.56	5656	Signature	9000004	2025/01/28 05:08:00
13	Client to Server	21.21.21.21	∅	11.11.11.11	∅	Match: To Server	9000003	2025/01/28 05:07:00
14	Client to Server	20.20.20.20	∅	10.10.10.10	∅	Match: Flow test	9000002	2025/01/28 05:06:00
15	Client to Server	50.50.50.50	5050	55.55.55.55	5555	Match: Possible ASREPRoasting	9000001	2025/01/28 05:05:00

Notes

- If an optional environment variable is not provided during the Cloud Function deployment, default values will be used, and data collection will start accordingly.
- The chunk limit for data collection is set to 100 to minimize data duplication in case of errors during ingestion, as the Google SecOps Ingestion API processes data in chunks of 100.
- We recommend setting the timeout in the RUNTIME variable to the maximum value (3600) to prevent the Cloud Function from terminating during data collection.
- The dashboard displays data in a tabular format and will show only the top 100 incidents, sorted by the selected field in the table.
- The Data Source Type will no longer include static values like Network, Microsoft 365 / Azure AD, and AWS. Instead, it will feature AWS, M365, O365, SAML, and Network, and filter data accordingly.

Limitations

- If the user does not specify the required environment variable while configuring the Cloud Function, the script deployment will fail.
- CBN parser will only be able to parse the Vectra Detect events.
- We suggest using the second generation of Cloud Function. The first generation of Cloud Function has a maximum execution time of 9 minutes and the second generation of Cloud Function has a maximum execution time of 60 minutes. If the execution time of the Cloud Function exceeds timeout then there are chances that the complete data won’t be ingested in the Google SecOps.
- The rate limit for a Vectra account depends on the user's subscription. Based on this API rate limit, the integration will be able to collect data and ingest into Google SecOps. Once the API rate limit is exceeded, data collection will only resume when the limit is reset after a specific interval.
- Looker doesn’t support API calls to fetch live data for populating dashboards.

- We do not have any marketplace for Google SecOps integration, so as a part of deliverables, we use to push our integration code to google public repo which undergoes many checks from google side.
- Common search filters can't be implemented in looker.
- Pagination is not supported in looker.
- Looker loads 5000 rows at a time, so charts are populated with the data of the latest 5000 rows based on the time range/value selected in the Time filter.
- Pagination in tabular visualization is not supported in Looker.
- Looker will only show data from the past 180 days, but this can vary as per the retention policy configured in BigQuery.
- According to the query time zone selected by the user in connection with the Google SecOps database, the Looker dashboards would be reflected according to the configured timezone.
- Looker doesn't support API calls to fetch live data for populating dashboards.
- The looker dashboard does not display data in the drill down table when there are too many records to be displayed.

Troubleshooting

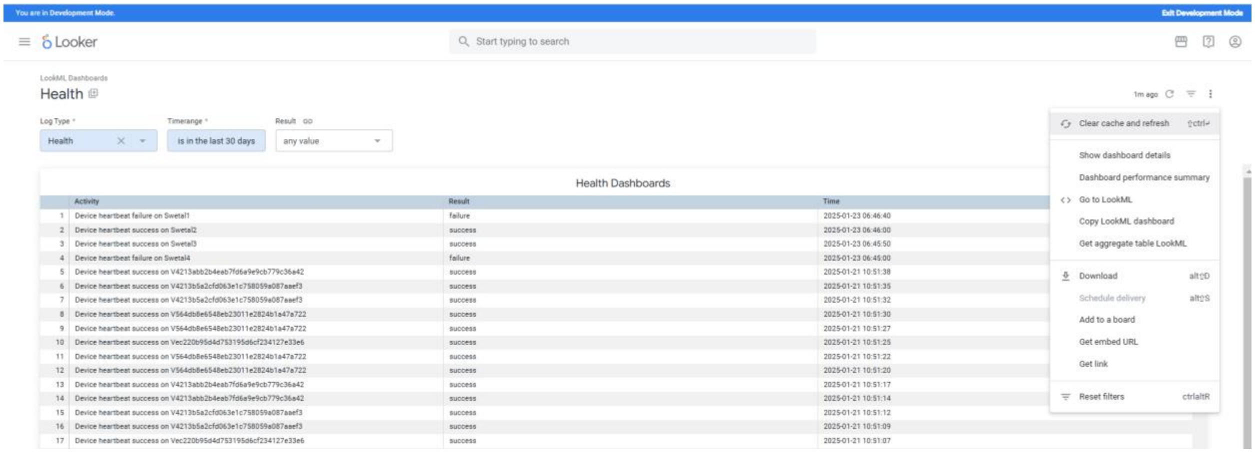
This section describes the common issues that might happen during the deployment or the running of the app and the steps to resolve the issues.

1. GCloud logs can be used for troubleshooting.
    a. Log in to the "https://console.cloud.google.com/" using valid credentials.
    b. Navigate to 'Cloud functions' and click on the deployed function where you can find the logs module.
    c. Logs can be filtered using severity.

    Currently, this logs feature is disabled by the google team in some GCP projects. We are currently checking with the google team regarding this.
2. If you test the cloud function immediately after deploying it on gcloud, It might be possible that the cloud function will not work as expected. To resolve this, wait for a few seconds and then test it.
3. If the cloud function stops its execution because memory exceeds the limit, reconfigure the cloud function’s memory configuration and increase the memory limit.
4. The Looker dashboard takes data from the cache and does not display the latest events.

    Solution:

    1. Click on the three dots present on the rightmost side of the dashboard.
    2. Click on the Clear cache and refresh.



Health

Log Type *: Health

Timerange *: is in the last 30 days

Result: any value

Health Dashboards

	Activity	Result	Time
1	Device heartbeat failure on Swetal1	failure	2025-01-23 06:46:40
2	Device heartbeat success on Swetal2	success	2025-01-23 06:46:00
3	Device heartbeat success on Swetal3	success	2025-01-23 06:45:50
4	Device heartbeat failure on Swetal4	failure	2025-01-23 06:45:00
5	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:38
6	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:35
7	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:32
8	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:30
9	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:27
10	Device heartbeat success on Vec220b95d4d753195d6cf234127e33e6	success	2025-01-21 10:51:25
11	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:22
12	Device heartbeat success on V564db8e6548eb23011e2824b1a47a722	success	2025-01-21 10:51:20
13	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:17
14	Device heartbeat success on V4213abb2b4eab7fd6a9e9cb779c36a42	success	2025-01-21 10:51:14
15	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:12
16	Device heartbeat success on V4213b5a2cfd063e1c758059a087aaef3	success	2025-01-21 10:51:09
17	Device heartbeat success on Vec220b95d4d753195d6cf234127e33e6	success	2025-01-21 10:51:07

5. The data is not displayed on the dashboard -

   This could be a problem with the data source as the database connection might be wrongly configured.

6. If desired events are not showing in the visualization -

   Make sure that the filters in the dashboard are configured correctly. If the filters are too restrictive, they may be preventing the dashboard from displaying any Data.

7. The dashboard may be slow to load or unresponsive - This could be due to a problem with the data source being unavailable or having too much data, the query that is being used, or the way that the dashboard is being rendered.

References

- Looker
- Looker Marketplace