USER GUIDE v1.2

Last Updated: Dec 2022

Cyral Standard Dashboard: Datadog

TABLE OF CONTENTS

I. Introduction

This guide accompanies the Cyral Standard Dashboard for Datadog. This dashboard ingests Data Activity Logs from all sidecars registered in your Control Plane. It includes a collection of pre-configured tables and graphs that display patterns in the ways users and applications access your data.

The Cyral Log Detail View accompanies the Standard Dashboard file. This detail view includes a set if pre-configured filters on your raw log data that complement the visualizations in the dashboard, and that make it easy to analyze your data.

II. Pre-Requisites

First
Configure the Datadog Integration in your Control Plane using the instructions provided in Cyral Docs: Send Cyral logs to Datadog.

Second (optional)
Many tables and visualizations in the dashboard require SSO User or Group information to display data correctly. To enable this, your IdP provider should be integrated with the Control Plane, and users should be using Cyral connection information to access your database(s).

III. Dashboard Setup Instructions

To setup the dashboard, you will need to complete four tasks in Datadog:

A. Create a Cyral Pipeline to be used for various mappings
B. Processor Creation
C. Create the facets needed for proper dashboard operation
D. Install JSON file

A. Create a Cyral Pipeline

1. Login to your Datadog
2. Navigate to Logs -> Configuration -> Pipelines
3. Click the `New Pipeline` button
4. Add the following information on the `Create Pipeline` screen
    - Filter: `source:cyral-*-wire`
    - Name: `CyralPipeline`

5. Click the `Create` button.
6. Once the `CyralPipeline` pipeline has been created, create the processors listed in the `Processor Creation` section)

B. Processor Creation

For the Cyral dashboards to function as expected, we need to create a few processors in the pipeline. This section outlines the processors that are required along with the order in which they should be created.

1. GeoIP Parser : IP Geo Mapper
    - Processor Type : GeoIP Parser
    - source IP attribute : client.host
    - target attribute path : network.client.geoip
    - Name : IP Geo Mapper

2. Remapper : Map Client IP
    - Processor Type : Remapper
    - attribute(s) or tag key to remap : client.host
    - target attribute or tag key : network.client.ip
    - Preserve source attribute : Enabled
    - Override on conflict : Disabled
    - Force attribute type : String
    - Name : Map Client IP

3. String Builder Processor : %{request.statementType}_%{repo.type} - in attribute lookup.categoryKey
    - Processor Type : String Builder Processor
    - target attribute path : lookup.categoryKey
    - target attribute value : %{request.statementType}_%{repo.type}
    - Replace missing attribute by an empty string : Enabled
    - Name : %{request.statementType}_%{repo.type} - in attribute lookup.categoryKey

4. Lookup Processor : Category Lookup
    - Processor Type : Lookup Processor
    - lookup mapping : Manual Mapping (Upload the ./lookups/CategoryLookupTable.csv from this directory)
    - Fallback value : Unknown
    - source attribute : lookup.categoryKey
    - target attribute path : lookup.categoryActivity
    - Name : Category Lookup

5. Lookup Processor : Term Lookup
    - Processor Type : Lookup Processor
    - lookup mapping : Manual Mapping (Upload the ./lookups/TermLookupTable.csv from this directory)
    - Fallback value : Unknown
    - source attribute : lookup.categoryKey
    - target attribute path : lookup.TermActivity
    - Name : Term Lookup

6. Lookup Processor : Privileged Command Lookup
    - Processor Type : Lookup Processor
    - lookup mapping : Manual Mapping (This should contain at least the line Privileged Commands,True. If you would like to include additional categories as "privileged", then list those here one per line)
    - source attribute : lookup.categoryActivity
    - target attribute path : lookup.priviledgedCmd
    - Name : Privileged Command Lookup
    - Some additional categories that could be used for this Manual Mapping are:
        - Access Changes
        - Data Changes
        - Data Reads
        - Schema Changes

To consider additional categories as `Privileged` simply add them to the lookup, (The example below adds both `Privileged Commands` and `Access Changes` as "privileged" commands) `Privileged Commands,True Access Changes,True`

C. Create Facets

For the dashboards to function properly, you will need to create the following facets from the Cyral logs:

Path	Display Name	Type	Group	Notes
@activityTypes	activityTypes	String	Cyral	
@client.applicationName	client.applicationName	String	Cyral	
@repo.name	repo.name	String	Cyral	
@sidecar.name	sidecar.name	String	Cyral	
@identity.repoUser	identity.repoUser	String	Cyral	

@identity.endUser	identity.endUser	String	Cyral	
@identity.group	identity.group	String	Cyral	
@client.connectionId	client.connectionId	String	Cyral	
@response.bytes	response.bytes	Integer	Cyral	
@response.isError	response.isError	Boolean	Cyral	
@response.executionTimeNanos	response.executionTimeNanos	Integer	Cyral	
@repo.type	repo.type	String	Cyral	
@lookup.categoryActivity	lookup.categoryActivity	String	Cyral	
@lookup.TermActivity	lookup.TermActivity	String	Cyral	
@lookup.priviledgedCmd	lookup.priviledgedCmd	String	Cyral	
@action	action	String	Cyral	This will only be available if you have integrated Cyral Activity Logs
@subject.user	subject.user	String	Cyral	This will only be available if you have integrated Cyral Activity Logs
@details.repoName	details.repoName	String	Cyral	This will only be available if you have integrated Cyral Activity Logs

D. JSON File Installation

1. While logged into Datadog, navigate to Dashboards -> New Dashboard

2. On the Create a Dashboard screen, enter any Dashboard Name (this will get overwritten) and click the New Dashboard button

3. On the resulting screen, click the gear icon in the upper right corner of the screen

4. From the gear menu, click on Import dashboard JSON.

5. Import the ./dashboards/Cyral-DataActivityLogsStandardDashboard.json file from this repo

6. At the Paste dashboard JSON prompt, click the Yes, Replace button

IV. Guide to Graphs and Tables

SYSTEM SUMMARY

Chart or Table Name	Notes
Number of Repositories	Conveys how many repositories contributed to data visible in the dashboard. This number does not necessarily reflect the number of repositories registered in your Control Plane.
Number of Sidecars	Conveys how many sidecars contributed to data visible in the dashboard. This number does not necessarily reflect the number of sidecars registered in your Control Plane.
Number of Registered Repository Accounts	Conveys how many individual repository accounts contributed to data visible in the dashboard. This number does not necessarily reflect the number of database accounts registered in your Control Plane.
Total Connections In Period	The Cyral Standard Dashboard uses the Datadog out-of-the-box configuration to calculate this field. The value is calculated differently depending on what value you have for the Timeframe filter.
Connection Activity by Geography	Shows IP address location of all connections to all repositories (unless filters are applied).

Number of Active Users	Number of unique users captured in the logs. This value reflects individual users who logged in via BI Tools (i.e. Looker, Tableau) if Service Account Resolution has been configured.
Number of Active SSO Groups	Number of unique SSO groups captured in the logs. This value reflects the SSO Groups of individual users who logged in via BI Tools (i.e. Looker, Tableau) if Service Account Resolution has been configured.
Number of Users in each SSO Group	Conveys number of users from each SSO Group that accessed one or more repositories during the time range specified by the dashboard filters.
Repository Connections by SSO Group	Conveys number of unique connections from users--classified by SSO Group—for the time range specified by the dashboard filters.
Recent Access Approvals	Displays all approvals; results are paginated after the 10 most recent approvals.
Approvals by Approver	Displays all approvals; results are paginated after the 10 most recent approvals.
Approvals by Repository	Displays all approvals; results are paginated after the 10 most recent approvals.
Repository Connections	Displays total number of connections and total number of queries to each repository across the timeframe specified. Average Query Size in Bytes calculated based on timeframe filter applied.

DATA ACTIVITY

Chart or Table Name	Notes
Repository Traffic Across Time (Queries)	Displays trends in total number of queries submitted to each repository. These queries may have been from individual users or applications.
Repository Traffic Across Time (Connections)	Displays trends in total number of connections to each repository. These connections could have been made by individual users or applications.
Activity Category by SSO Group	Displays which query types are used most frequently by various SSO Groups. See **Appendix A** for details on the specific queries grouped into the "Activity Categories" listed.
Trends in User Activity	Displays trends in query usage across various repos.

SECURITY ACTIVITY

Chart or Table Name	Notes
Privileged Commands by Users	See Appendix A for a list of queries categorized as a "privileged command."
Privileged Commands Trends	Conveys how many users from various SSO groups executed privileged commands/queries.
Access Changes by Users	See Appendix A for a list of queries categorized as an "Access Change."
Access Changes Trends	Conveys how many users from various SSO groups executed access change commands/queries.
Suspicious Activity	Describes frequency and types of Suspicious Activity happening in each of your repositories. Suspicious Activity includes: • Port Scans

	• Full Table Scans • Authentication Failures • Cyral Policy Triggers
Suspicious Activity Trends	Conveys frequency of Suspicious Activity for each of your repositories.

REPOSITORY PERFORMANCE

Chart or Table Name	Notes
Summary	Displays statistics based on the timeframe assigned in your global filters. Duration metrics are calculated based on the difference between first query seen and last query seen for a given connection.
Query Error Rates by Repository	Displays the count and percentage of queries with errors.
Trend of Errors Over Time	Displays trends in number of errors across time for each repository.
Activity Categories with Highest Error Rates	Displays the number of queries with errors across the timeframe specified in your global filters.
Slowest Queries	Displays top ten slowest queries. Note: The Data Table/Field only populates if a query was for data registered in your Data Map.

VI. Appendix A – Data Activity Logs Taxonomy

Cyral maps relevant query language from all logs coming from your various repositories to these central Activity Terms. These terms are then grouped into Activity Categories for ease of analysis and visualization in your dashboards. If you'd like to see the complete mapping between Activity Terms and database query statements, please contact Customer Support.

Activity Category	Activity Term	Notes
Data Reads	Select Data	*Example:* SELECT (PostgreSQL) and FIND (MongoDB) both map to the term Select Data.
	Select Data Bulk	
	Select Data	
	Export Data Bulk	
	Analyze Statistics	
	Explain	
Data Changes	Insert Data	
	Update Data	
	Delete Data	
	Merge Data	
	Copy Data	
	Truncate Data	
View Changes	Create View	
	Alter View	
	Drop View	
Schema Changes	Create Table	
	Alter Table	
	Delete Table	
	Rename Table	
	Create Schema	
	Modify Schema	
	Delete Schema	
	Annotate Schema	
	Create Collection	*Not all Activity Terms are relevant to every repository type.*

	Alter Collection	
	Delete Collection	
	Create Bucket	
	Delete Bucket	
Query Flow Operation	Begin Transaction	
	Cursor Operation	
	Variable Declaration	
	Clear Session State	
	Execute Stored Procedure	
	Listen	
	Notify	
	Prepare Statement	
	Release Savepoint	
	Rollback Transaction	
	Commit Transaction	
	Savepoint	
Repo Changes	Create Database	
	Update Database	
	Annotate Database	
	Delete Database	
Access Changes	Create User Account	
	Modify User Account	
	Delete User Account	
	Create Group	
	Modify Group	
	Delete Group	
	Create Role	
	Modify Role	
	Delete Role	
	Modify Access	*Not relevant for PG repositories*
	Grant Access	
	Revoke Access	
Privileged Commands	Modify Log	
	System Change	
	Functionality Change	
	Alter Trigger	
	Create Trigger	

	Delete Trigger	
	Backup	
	Restore	*Not relevant for PG repositories*
	Create Access Method	
	binlog	*Not relevant for PG repositories*
	Flush	*Not relevant for PG repositories*
	Kill	*Not relevant for PG repositories*
	Deadlock	*Not relevant for PG repositories*
	Load Index Into Cache	*Not relevant for PG repositories*
	Reset	*Not relevant for PG repositories*
	Reset Persist	*Not relevant for PG repositories*
	Startup	*Not relevant for PG repositories*
	Restart	*Not relevant for PG repositories*
	Shutdown	*Not relevant for PG repositories*