Korrel8r

Signal Correlation for Kubernetes and Beyond

Korrel8r Project

Version v0.3.1_dev_10, 2023-10-23

Table of Contents

***Korrel8r is a correlation engine for observability signals and observable** resources.*

It can correlate across multiple domains, diverse signal types, inconsistent labelling and varied data stores.

NOTE	Get the latest code from the Korrel8r project. This guide is available online in HTML and PDF ebook format.

Overview

Korrel8r is a correlation engine for *observability signals* and *observed resources*. A *signal* is a unit of observation data such as a log record, a trace record, a metric time-series, or an alert. A *resource* is something to be observed, such as a Kubernetes Pod or Deployment.

The goal of korrel8r is to express relationships between *objects*(resources or signals) in different *domains* as *rules*, and to automatically navigate the resulting graph of rules and objects to find related data.

For example: an alert is raised on a cluster console. Diagnosing the problem, requires logs from the containers that caused the alert. Korrel8rs can automatically follow the chain of relationships `alert → deployment → pod → logs` and display the related logs directly.

Given a *start* object (e.g. an Alert in a cluster) and a *goal* (e.g. *find related logs*) the engine can search for a chain of rules leading from start to goal data. The engine can also display the *neighbourhood* of all related data around some object of interest.

The rules capture expert knowledge about troubleshooting in an executable form. The engine aims to provide common rule-base that can be re-used in many settings: as a service, in consoles, in command line tools, or offline data-processing systems.

The goals of this project include:

- Encode domain knowledge from SREs and other experts as re-usable rules.
- Automate navigation from symptoms to data that helps diagnose causes.
- Relate diverse data that is held in different types of storage with different schema.
- Reduce multiple-step manual procedures to fewer clicks or queries.
- Help tools that gather and analyze diagnostic data to focus on relevant information.

There is a short video demo

Key Concepts

Object

An instance of a signal or resource.

Domain

A family of related objects (signals and/or resources) with common storage and representation. Examples: `k8s` (Kubernetes resources), `alert` (Prometheus alerts), `metric` (Prometheus metrics), `log` (Loki logs), `trace` (OpenTelemetry traces)
FIXME: xref domain ref section

Store

A source of stored objects from a single Domain. Examples: Loki `log`, Prometheus `alert`, Kubernetes API server `k8s`.

Query

A Query selects a set of objects from a store. The structure of a query depends on the domain of the store.

Class

A subset of objects in a Domain with a common schema (the same field definitions). Classes are named <class-name>.<domain-name>
Examples: `Pod.k8s`, `DaemonSet.k8s` `audit.log`

Rule

A Rule applies to an object of a *start* class, and generates a query for a *goal* class. The start and goal can be in different domains (e.g. `Pod.k8s` → `application.log`) A rule definition is a *template* that constructs a query for the *goal class* using the native field names of an object of the *start class*. Rules provide the bridge between domains with inconsistent labels and schema.

Conflicting Vocabularies

Different signal and object domains may use different vocabularies to identify the same things. For example:

- `k8s.pod.name` (trace)
- `pod` or `pod_name` (metric)
- `kubernetes.pod_name` (log)

The correlation problem would be simpler if there was a single vocabulary to describe signal attributes. The Open Telemetry Project aims to create such a standard vocabulary, but in the short to medium term we will have to contend with mixed-format signals. Korrel8r expresses rules in the native vocabulary of each domain and allowing rules to cross domains.

Examples of Objects

A Kubernetes cluster generates many types of *observable signal*, including:

Signal Type	Description
Metrics	Counts and measurements of system behaviour.
Alerts	Rules that fire when metrics cross important thresholds.
Logs	Application, infrastructure and audit logs from Pods and cluster nodes.
Kubernetes Events	Describe significant events in a cluster.
Traces	Nested execution spans describing distributed requests.
Network Events	TCP and IP level network information.

A cluster also contains *resources* that are not usually considered ``signals'', but that can be correlated with signals and other resources:

Resource Type	Description
k8s resources	Spec and status information.
Run books	Problem solving guides associated with Alerts.
k8s probes	Information about resource state.

Korrel8r does not distinguish between signals and resources, and refers to both as `objects`.

Appendix A: REST API for korrel8r

Informations

Version

v1alpha1

Contact

https://github.com/korrel8r/korrel8r

Content negotiation

URI Schemes

- http

Consumes

- application/json

Produces

- application/json

All endpoints

configuration

Method	URI	Name	Summary
GET	/api/v1alpha1/domains	get domains	List all configured domains and stores.
GET	/api/v1alpha1/domains/{domain}/classes	get domains domain classes	Get class names and descriptions for the domain.

search

Method	URI	Name	Summary
POST	/api/v1alpha1/graphs/goals	post graphs goals	Create a correlation graph from start objects to goal queries.

Method	URI	Name	Summary
POST	/api/v1alpha1/graphs/neighbours	post graphs neighbours	Create a correlation graph of neighbours of a start object to a given depth.
POST	/api/v1alpha1/lists/goals	post lists goals	Generate a list of goal nodes related to a starting point.

Paths

[get-domains]# List all configured domains and stores. (*GetDomains*)

```
GET /api/v1alpha1/domains
```

All responses

Code	Status	Description	Has headers	Schema
200	OK	OK		schema

Responses

[get-domains-200]# 200 - OK

Status: OK

======= [get-domains-200-schema]# Schema

[][APIDomain](#api-domain)

[get-domains-domain-classes]# Get class names and descriptions for the domain. (*GetDomainsDomainClasses*)

```
GET /api/v1alpha1/domains/{domain}/classes
```

Parameters

Name	Source	Type	Go type	Separator	Required	Default	Description
domain	path	string	string		☐		Domain to get classes from.

All responses

Code	Status	Description	Has headers	Schema
200	OK	OK		schema

Responses

[get-domains-domain-classes-200]# 200 - OK

Status: OK

======= [get-domains-domain-classes-200-schema]# Schema

APIClasses

[post-graphs-goals]# Create a correlation graph from start objects to goal queries. (*PostGraphsGoals*)

```
POST /api/v1alpha1/graphs/goals
```

Parameters

Name	Source	Type	Go type	Separator	Required	Default	Description
withRules	query	boolean	bool				include rules in graph edges
start	body	APIGoalsRequest	models.APIGoalsRequest		✓		search from start to goal classes

All responses

Code	Status	Description	Has headers	Schema
200	OK	OK		schema

Responses

[post-graphs-goals-200]# 200 - OK

Status: OK

======= [post-graphs-goals-200-schema]# Schema

APIGraph

[post-graphs-neighbours]# Create a correlation graph of neighbours of a start object to a given depth. (*PostGraphsNeighbours*)

```
POST /api/v1alpha1/graphs/neighbours
```

Parameters

Name	Source	Type	Go type	Separator	Required	Default	Description
withRules	query	boolean	bool				include rules in graph edges
start	body	APINeighboursRequest	models.APINeighboursRequest		✓		search from neighbours

All responses

Code	Status	Description	Has headers	Schema
200	OK	OK		schema

Responses

[post-graphs-neighbours-200]# 200 - OK

Status: OK

======= [post-graphs-neighbours-200-schema]# Schema

APIGraph

[post-lists-goals]# Generate a list of goal nodes related to a starting point. (*PostListsGoals*)

```
POST /api/v1alpha1/lists/goals
```

Parameters

Name	Source	Type	Go type	Separator	Required	Default	Description
start	body	APIGoalsRequest	models.APIGoalsRequest		✓		search from start to goal classes

All responses

Code	Status	Description	Has headers	Schema
200	OK	OK		schema

Responses

[post-lists-goals-200]# 200 - OK

Status: OK

======= [post-lists-goals-200-schema]# Schema

[][APINode](#api-node)

Models

[api-classes]# api.Classes

> Classes maps class names to a short description.

APIClasses

[api-domain]# api.Domain

> Domain configuration information.

Properties

Name	Type	Go type	Required	Default	Description	Example
errors	[]string	`[]string`				
name	string	`string`				
stores	[][Korrel8rStoreConfig](#korrel8r-store-config)	`[]Korrel8rStoreConfig`				

[api-edge]# api.Edge

Properties

Name	Type	Go type	Required	Default	Description	Example
goal	string	`string`			Goal is the class name of the goal node.	`class.domain`
rules	[][APIRule](#api-rule)	`[]*APIRule`			Rules is the set of rules followed along this edge (optional).	

Name	Type	Go type	Required	Default	Description	Example
start	string	`string`			Start is the class name of the start node.	

[api-goals-request]# api.GoalsRequest

> Starting point for a goals search.

Properties

Name	Type	Go type	Required	Default	Description	Example
goals	[]string	`[]string`			Goal classes for correlation.	`["class.domain"]`
start	APIGoalsRequest	`APIGoalsRequest`			Start of correlation search.	

[api-graph]# api.Graph

> Graph resulting from a correlation search.

Properties

Name	Type	Go type	Required	Default	Description	Example
edges	[][APIEdge](#api-edge)	`[]*APIEdge`				
nodes	[][APINode](#api-node)	`[]*APINode`				

[api-neighbours-request]# api.NeighboursRequest

> Starting point for a neighbours search.

Properties

Name	Type	Go type	Required	Default	Description	Example
depth	integer	`int64`			Max depth of neighbours graph.	
start	APINeighboursRequest	`APINeighboursRequest`			Start of correlation search.	

[api-node]# api.Node

Properties

Name	Type	Go type	Required	Default	Description	Example
class	string	`string`			Class is the full name of the class in ``CLASS.DOMAIN'' form.	`class.domain`
count	integer	`int64`			Count of results found for this class, after de-duplication.	
queries	[][APIQueryCount](#api-query-count)	`[]*APIQueryCount`			Queries yielding results for this class.	

[api-query-count]# api.QueryCount

> Query run during a correlation with a count of results found.

Properties

Name	Type	Go type	Required	Default	Description	Example
count	integer	`int64`			Count of results or -1 if the query was not executed.	
query	interface\{}	`interface{}`			Query for correlation data.	

[api-rule]# api.Rule

Properties

Name	Type	Go type	Required	Default	Description	Example
name	string	`string`			Name is an optional descriptive name.	
queries	[][APIQueryCount](#api-query-count)	`[]*APIQueryCount`			Queries generated while following this rule.	

[api-start]# api.Start

> Starting point for correlation.

Properties

Name	Type	Go type	Required	Default	Description	Example
class	string	`string`			Class of starting objects	`class.domain`
objects	interface\{}	`interface{}`			Objects in JSON form	
queries	interface\{}	`interface{}`			Queries for starting objects	

[korrel8r-store-config]# korrel8r.StoreConfig

Korrel8rStoreConfig