
Karrot Documentation

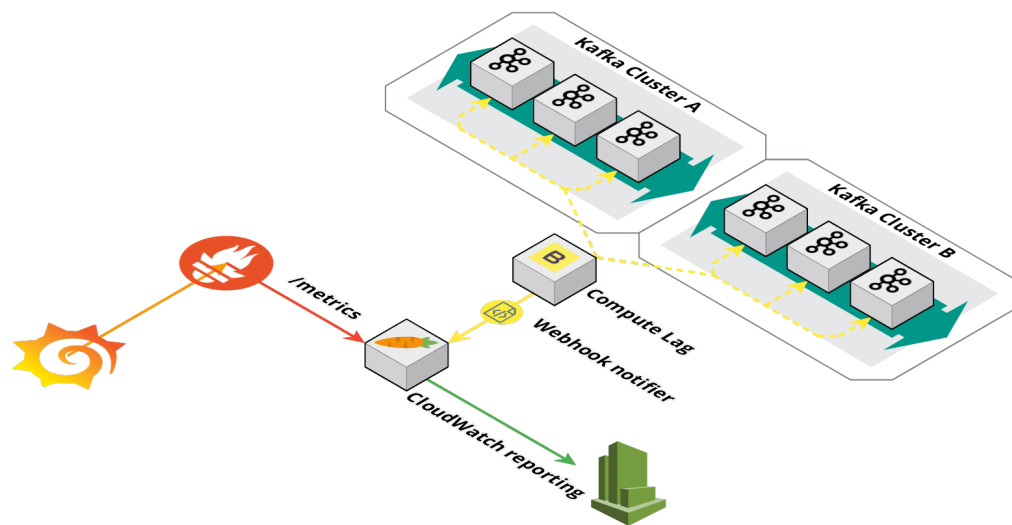
Release 0.0.1

GumGum MLE Team

Oct 07, 2020

1	Application diagram	1
1.1	Burrow	1
1.2	Heartbeat	2
1.3	Config	2
1.4	Reporters	2
2	Indices and tables	5
	Python Module Index	7
	Index	9

Application diagram



1.1 Burrow

synopsis Burrow Blueprint

Main application endpoint called by Burrow

```
karrot.burrow.controllers.event_handler(event)
```

For each enabled reporter, call the `process(event)` function.

If `prometheus` is enabled, it tracks the reporter update.

Parameters `event` (*str*) – A valid Burrow Json event

`karrot.burrow.controllers.webhook_handler()`

Process an incoming event from Burrow and call `event_handler()` to process it.

Parameters `event` (*str*) – A valid Burrow Json event POSTed to this endpoint

1.2 Heartbeat

synopsis Heartbeat Blueprint

Heartbeat endpoint of the application

`karrot.heartbeat.controllers.health()`

Returns a simple JSON string when the application is healthy.

Returns `json` – A JSON with the following format:

```
“{“status”: “success”, “msg”: “Burrow-reporter is healthy”, “time”: “<datetime.now()>”}“
```

1.3 Config

synopsis Configuration Blueprint

Configuration endpoint for the application

`karrot.config.controllers.display()`

Returns a simple JSON string with the current application configuration.

Returns `json` – A JSON with the following format:

```
{"status": "success", "settings": "{...}"}
```

1.4 Reporters

Reporters called once a webhook has been received from Burrow

1.4.1 Cloudwatch

class `karrot.reporters.cloudwatch.models.CloudwatchReporter` (*name*)

Bases: `karrot.reporters.models.Reporter`

process (*event*)

Refresh the object with the event details. A child class should call this parent function and then implement the proper processing steps for the event.

Parameters `event` (*str*) – A

stats (*reporter*)

Report stats of an other reporter. Useful for Prometheus to collect stats about other reporters.

This method must be implemented in child classes.

1.4.2 Prometheus

`karrot.reporters.prometheus.controllers.metrics()`

class `karrot.reporters.prometheus.models.PrometheusReporter` (*name*)

Bases: `karrot.reporters.models.Reporter`

process (*event*)

Refresh the object with the event details. A child class should call this parent function and then implement the proper processing steps for the event.

Parameters *event* (*str*) – A

stats (*reporter*)

Increment the number of events processed by the reporter.

synopsis Reporter Blueprint

Reporter endpoint of the application

`karrot.reporters.controllers.display` (*reporter*)

Returns a simple JSON with the current application reporters.

Parameters *reporter* (*str*) – If provided returns details about this reporter only

Returns *json* – A JSON with the following format:

```
{"reporters": ["reporter1", "reporter2"]}
```

class `karrot.reporters.models.Reporter` (*name*)

Bases: `object`

process (*event*)

Refresh the object with the event details. A child class should call this parent function and then implement the proper processing steps for the event.

Parameters *event* (*str*) – A

stats (*reporter*)

Report stats of an other reporter. Useful for Prometheus to collect stats about other reporters.

This method must be implemented in child classes.

class `karrot.reporters.factory.ReporterFactory`

Bases: `object`

ReporterFactory used to create a Reporter object based on it's name.

static get (*reporter*)

Return the proper notifier object based on the name.

Params *str reporter* The type of reporter to create

Returns *Reporter* – A reporter matching *reporter*

Raises *ValueError* – When the requested reporter is not found.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

k

- `karrot.burrow.controllers`, 1
- `karrot.config.controllers`, 2
- `karrot.heartbeat.controllers`, 2
- `karrot.reporters.cloudwatch.models`, 2
- `karrot.reporters.controllers`, 3
- `karrot.reporters.factory`, 3
- `karrot.reporters.models`, 3
- `karrot.reporters.prometheus.controllers`, 3
- `karrot.reporters.prometheus.models`, 3

C

CloudwatchReporter (class in *karrot.reporters.cloudwatch.models*), 2

D

display() (in module *karrot.config.controllers*), 2

display() (in module *karrot.reporters.controllers*), 3

E

event_handler() (in module *karrot.burrow.controllers*), 1

G

get() (*karrot.reporters.factory.ReporterFactory* static method), 3

H

health() (in module *karrot.heartbeat.controllers*), 2

K

karrot.burrow.controllers (module), 1

karrot.config.controllers (module), 2

karrot.heartbeat.controllers (module), 2

karrot.reporters.cloudwatch.models (module), 2

karrot.reporters.controllers (module), 3

karrot.reporters.factory (module), 3

karrot.reporters.models (module), 3

karrot.reporters.prometheus.controllers (module), 3

karrot.reporters.prometheus.models (module), 3

M

metrics() (in module *karrot.reporters.prometheus.controllers*), 3

P

process() (*karrot.reporters.cloudwatch.models.CloudwatchReporter* method), 2

process() (*karrot.reporters.models.Reporter* method), 3

process() (*karrot.reporters.prometheus.models.PrometheusReporter* method), 3

PrometheusReporter (class in *karrot.reporters.prometheus.models*), 3

R

Reporter (class in *karrot.reporters.models*), 3

ReporterFactory (class in *karrot.reporters.factory*), 3

S

stats() (*karrot.reporters.cloudwatch.models.CloudwatchReporter* method), 2

stats() (*karrot.reporters.models.Reporter* method), 3

stats() (*karrot.reporters.prometheus.models.PrometheusReporter* method), 3

W

webhook_handler() (in module *karrot.burrow.controllers*), 2