Librato Provider

The Librato provider is used to interact with the resources supported by Librato. The provider needs to be configured with the proper credentials before it can be used.

Use the navigation to the left to read about the available resources.

Example Usage

```
# Configure the Librato provider
provider "librato" {
  email = "ops@company.com"
  token = "${var.librato_token}"
}

# Create a new space
resource "librato_space" "default" {
  # ...
}
```

Argument Reference

The following arguments are supported:

- `token` - (Required) Librato API token. It must be provided, but it can also be sourced from the `LIBRATO_TOKEN` environment variable.
- `email` - (Required) Librato email address. It must be provided, but it can also be sourced from the `LIBRATO_EMAIL` environment variable.

librato_alert

Provides a Librato Alert resource. This can be used to create and manage alerts on Librato.

Example Usage

```
# Create a new Librato alert
resource "librato_alert" "myalert" {
  name        = "MyAlert"
  description = "A Test Alert"
  services    = ["${librato_service.myservice.id}"]

  condition {
    type        = "above"
    threshold   = 10
    metric_name = "librato.cpu.percent.idle"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the alert.
- `description` - (Required) Description of the alert.
- `active` - whether the alert is active (can be triggered). Defaults to true.
- `rearm_seconds` - minimum amount of time between sending alert notifications, in seconds.
- `services` - list of notification service IDs.
- `condition` - A trigger condition for the alert. Conditions documented below.
- `attributes` - A hash of additional attribtues for the alert. Attributes documented below.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the alert.
- `name` - The name of the alert.
- `description` - (Required) Description of the alert.
- `active` - whether the alert is active (can be triggered). Defaults to true.
- `rearm_seconds` - minimum amount of time between sending alert notifications, in seconds.
- `services` - list of notification service IDs.

- `condition` - A trigger condition for the alert. Conditions documented below.

Conditions (`condition`) support the following:

- `type` - The type of condition. Must be one of `above`, `below` or `absent`.
- `metric_name`- The name of the metric this alert condition applies to.
- `source`- A source expression which identifies which sources for the given metric to monitor.
- `detect_reset` - boolean: toggles the method used to calculate the delta from the previous sample when the summary_function is `derivative`.
- `duration` - number of seconds condition must be true to fire the alert (required for type `absent`).
- `threshold` - float: measurements over this number will fire the alert (only for `above` or `below`).
- `summary_function` - Indicates which statistic of an aggregated measurement to alert on. ((only for `above` or `below`).

Attributes (`attributes`) support the following:

- `runbook_url` - a URL for the runbook to be followed when this alert is firing. Used in the Librato UI if set.

librato_metric

Provides a Librato Metric resource. This can be used to create and manage metrics on Librato.

Example Usage

```
# Create a new Librato metric
resource "librato_metric" "mymetric" {
    name = "MyMetric"
    type = "counter"
    description = "A Test Metric"
    attributes {
      display_stacked = true
    }
}
```

Argument Reference

The following arguments are supported:

- `type` - (Required) The type of metric to create (gauge, counter, or composite).
- `name` - (Required) The unique identifier of the metric.
- `display_name` - The name which will be used for the metric when viewing the Metrics website.
- `description` - Text that can be used to explain precisely what the metric is measuring.
- `period` - Number of seconds that is the standard reporting period of the metric.
- `attributes` - The attributes hash configures specific components of a metric's visualization.
- `composite` - The definition of the composite metric.

Attributes Reference

The following attributes are exported:

- `name` - The identifier for the metric.
- `display_name` - The name which will be used for the metric when viewing the Metrics website.
- `type` - The type of metric to create (gauge, counter, or composite).
- `description` - Text that describes precisely what the metric is measuring.
- `period` - Number of seconds that is the standard reporting period of the metric. Setting the period enables Metrics to detect abnormal interruptions in reporting and aids in analytics. For gauge metrics that have service-side aggregation enabled, this option will define the period that aggregation occurs on.
- `source_lag` -

- `composite` - The composite definition. Only used when type is composite.

Attributes (`attributes`) support the following:

- `color` - Sets a default color to prefer when visually rendering the metric. Must be a seven character string that represents the hex code of the color e.g. #52D74C.
- `display_max` - If a metric has a known theoretical maximum value, set display_max so that visualizations can provide perspective of the current values relative to the maximum value.
- `display_min` - If a metric has a known theoretical minimum value, set display_min so that visualizations can provide perspective of the current values relative to the minimum value.
- `display_units_long` - A string that identifies the unit of measurement e.g. Microseconds. Typically the long form of display_units_short and used in visualizations e.g. the Y-axis label on a graph.
- `display_units_short` - A terse (usually abbreviated) string that identifies the unit of measurement e.g. uS (Microseconds). Typically the short form of display_units_long and used in visualizations e.g. the tooltip for a point on a graph.
- `display_stacked` - A boolean value indicating whether or not multiple metric streams should be aggregated in a visualization (e.g. stacked graphs). By default counters have display_stacked enabled while gauges have it disabled.
- `summarize_function` - Determines how to calculate values when rolling up from raw values to higher resolution intervals. Must be one of: ‘average’, 'sum’, 'count’, 'min’, 'max’. If summarize_function is not set the behavior defaults to average.

If the values of the measurements to be rolled up are: 2, 10, 5:

- average: 5.67
- sum: 17
- count: 3
- min: 2
- max: 10
- `aggregate` - Enable service-side aggregation for this metric. When enabled, measurements sent using the same tag set will be aggregated into single measurements on an interval defined by the period of the metric. If there is no period defined for the metric then all measurements will be aggregated on a 60-second interval.

This option takes a value of true or false. If this option is not set for a metric it will default to false.

librato_service

Provides a Librato Service resource. This can be used to create and manage notification services on Librato.

Example Usage

```
# Create a new Librato service
resource "librato_service" "email" {
  title = "Email the admins"
  type  = "mail"

  settings = <<EOF
{
  "addresses": "admin@example.com"
}
EOF
}
```

Argument Reference

The following arguments are supported. Please check the relevant documentation (https://github.com/librato/librato-services/tree/master/services) for each type of alert.

- `type` - (Required) The type of notificaion.
- `title` - (Required) The alert title.
- `settings` - (Required) a JSON hash of settings specific to the alert type.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the alert.
- `type` - The type of notificaion.
- `title` - The alert title.
- `settings` - a JSON hash of settings specific to the alert type.

librato_space

Provides a Librato Space resource. This can be used to create and manage spaces on Librato.

Example Usage

```
# Create a new Librato space
resource "librato_space" "default" {
  name = "My New Space"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the space.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the space.

- `name` - The name of the space.

librato_space_chart

Provides a Librato Space Chart resource. This can be used to create and manage charts in Librato Spaces.

Example Usage

```
# Create a new Librato space
resource "librato_space" "my_space" {
  name = "My New Space"
}

# Create a new chart
resource "librato_space_chart" "server_temperature" {
  name     = "Server Temperature"
  space_id = "${librato_space.my_space.id}"

  stream {
    metric = "server_temp"
    source = "app1"
  }

  stream {
    metric           = "environmental_temp"
    source           = "*"
    group_function   = "breakout"
    summary_function = "average"
  }

  stream {
    metric         = "server_temp"
    source         = "%"
    group_function = "average"
  }
}
```

Argument Reference

The following arguments are supported:

- `space_id` - (Required) The ID of the space this chart should be in.
- `name` - (Required) The title of the chart when it is displayed.
- `type` - (Optional) Indicates the type of chart. Must be one of line or stacked (default to line).
- `min` - (Optional) The minimum display value of the chart's Y-axis.
- `max` - (Optional) The maximum display value of the chart's Y-axis.
- `label` - (Optional) The Y-axis label.
- `related_space` - (Optional) The ID of another space to which this chart is related.
- `stream` - (Optional) Nested block describing a metric to use for data in the chart. The structure of this block is described below.

The `stream` block supports:

- `metric` - (Required) The name of the metric. May not be specified if `composite` is specified.
- `source` - (Required) The name of a source, or `*` to include all sources. This field will also accept specific wildcard entries. For example us-west-*-app will match us-west-21-app but not us-west-12-db. Use % to specify a dynamic source that will be provided after the instrument or dashboard has loaded, or in the URL. May not be specified if `composite` is specified.
- `group_function` - (Required) How to process the results when multiple sources will be returned. Value must be one of average, sum, breakout. If average or sum, a single line will be drawn representing the average or sum (respectively) of all sources. If the group_function is breakout, a separate line will be drawn for each source. If this property is not supplied, the behavior will default to average. May not be specified if `composite` is specified.
- `composite` - (Required) A composite metric query string to execute when this stream is displayed. May not be specified if `metric`, `source` or `group_function` is specified.
- `summary_function` - (Optional) When visualizing complex measurements or a rolled-up measurement, this allows you to choose which statistic to use. Defaults to "average". Valid options are: "max", "min", "average", "sum" or "count".
- `name` - (Optional) A display name to use for the stream when generating the tooltip.
- `color` - (Optional) Sets a color to use when rendering the stream. Must be a seven character string that represents the hex code of the color e.g. "#52D74C".
- `units_short` - (Optional) Unit value string to use as the tooltip label.
- `units_long` - (Optional) String value to set as they Y-axis label. All streams that share the same units_long value will be plotted on the same Y-axis.
- `min` - (Optional) Theoretical minimum Y-axis value.
- `max` - (Optional) Theoretical maximum Y-axis value.
- `transform_function` - (Optional) Linear formula to run on each measurement prior to visualization.
- `period` - (Optional) An integer value of seconds that defines the period this stream reports at. This aids in the display of the stream and allows the period to be used in stream display transforms.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the chart.
- `space_id` - The ID of the space this chart should be in.
- `title` - The title of the chart when it is displayed.