Cloud Cost Trends API

Cloud Cost Trends API

`GET` `http://<your-kubecost-address>/model/cloudCost/view/trends`

Analyzes change in cloud costs relative to a previous window of the same size

Path Parameters

Name	Type	Description
window*	string	Duration of time over which to query. Compares cost usage of window to cost usage window of equal size directly preceding it. Accepts multiple different formats of time (see this Using the `window` parameter section for more info)).
names*	string	Determines order sequence of queried items via comma-separated list. Dependent on the value of `aggregate` to list items. See more below.
aggregate	string	Field by which to aggregate the results. Accepts: `invoiceEntityID` , `accountID` , `provider` , `service` , and `label:<name>` . Supports multi-aggregation using comma-separated lists, such as `aggregate=accountID,service` . When no value is provided, the query will aggregate by individual items.
accumulate	boolean	When set to `false` , returns daily time series data vs. cumulative data. Default is `true` .
CostMetricName e	string	Determines which cloud cost metric type will be returned. Acceptable values are `AmortizedNetCost` , `InvoicedCost` , `ListCost` , and `NetCost` . Default is `AmortizedNetCost` .
filter	string	Filter your results by a particular category or value. For example, when to only see

Name	Type	Description
		trends in AWS spend, set this parameter to `filter=provider:"AWS"`. Supports Kubecost's [advanced filtering](#) language.

200: OK

```
{
    "code": 200,
    "data": {
        "trends": {
            "": {
                "isInfinite": false,
                "isNaN": false,
                "value": 0.000
            }
        },
        "window": {
            "start": "",
            "end": ""
        },
        "comparisonWindow": {
            "start": "",
            "end": ""
        }
    }
}
```

Calculating trend value

The Trends API determines changes in resource cost usage over time based on the interval set `window` parameter and provides that information via the schema field `value`. Cost usage for the current `window` sampled will be compared with `comparisonWindow`, the window directly before the current window of the same size interval. For example, for `window=3d`, Kubecost will output cost usage for the past three days compared to cost usage of the three days before the start of the window. This means a total of six days of cloud cost data are sampled in order to provide an accurate value.

The equation for calculating `value` is: `value=window/comparisonWindow - 1`

Receiving a positive `value` means spending has increased in the current `window` when compared to `comparisonWindow`. A negative `value` means spending has decreased.

> ⚠ It's important to recognize when a resource is not detected to exist in the previous window. This is designated by the field `IsInfinite=true`, which means the allocation could not be determined to exist. Otherwise, the cause of an unexpected or major trend change could be misattributed. The field `isNaN`, meaning not a number, refers to if the `value` is unreal. If so, `isNan` should return `true`, which means there was an error during calculation. Both fields should return `false` during a successful query.

In the example output below, `value` is expressed as `-0.147`, meaning spending has decreased for `project-123` by roughly 14.7%.

```
"trends": {
    "project-123": {
        "isInfinite": false,
        "isNaN": false,
        "value": -0.1472170691451784
    }
},
"window": {
    "start": "2023-11-29T00:00:00Z",
    "end": "2023-12-06T00:00:00Z"
},
"comparisonWindow": {
    "start": "2023-11-22T00:00:00Z",
    "end": "2023-11-29T00:00:00Z"
}
```

Trend values are converted into percentages in the Kubecost Cloud Costs Explorer page, calculated based on your current query. Trends will be presented in the rightmost column, next to your Total cost. The `window` parameter is determined by your selected date range in the top right of the page. The default is Last 7 days (`window=7d`). The equation `value*100` is used to provide percentages.

Using the `names` parameter

`names` is a mandatory parameter which determines the sequence of items returned, based on whatever the query is aggregating by. For example, when using `aggregate=provider`, the user should provide a comma-separated list of all providers they wish to see trend values for in this category. In this case, they should provide `names=AWS,GCP,Azure` to receive a list of trend values for all three providers. If the user does not provide a value for `aggregate`, they must still use the names parameter to list all line items requested.