Allocation Trends API

Allocation Trends API

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

Path Parameters

Name	Type	Description
window*	string	
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: `cluster`, `namespace`, `controllerKind`, `controller`, `service`, `node`, `pod`, `label:<name>`, and `annotation:<name>`. Also accepts comma-separated lists for multi-aggregation, like `namespace,label:app`.
accumulate	boolean	When set to `false`, returns daily time series data vs. cumulative data. Default is `true`.
external	string	If `true`, include external costs in each allocation. Default is `false`.
idle	boolean	If `true`, include idle cost (i.e. the cost of the un-allocated assets) as its own allocation. Default is `true`.
filterNamespaces	string	Comma-separated list of namespaces to match; e.g. `namespace-one,namespace-two` will return results from only those two namespaces.
shareIdle	boolean	If true, idle cost is allocated proportionally across all non-idle allocations, per-resource. That is, idle

Name	Type	Description
		CPU cost is shared with each non-idle allocation's CPU cost, according to the percentage of the total CPU cost represented. Default is false.
shareNamespaces	string	Comma-separated list of namespaces to share; e.g. `kube-system, kubecost` will share the costs of those two namespaces with the remaining non-idle, unshared allocations.
shareLabels	string	Comma-separated list of labels to share; e.g. `env:staging, app:test` will share the costs of those two label values with the remaining non-idle, unshared allocations.
shareCost	float	Floating-point value representing a monthly cost to share with the remaining non-idle, unshared allocations; e.g. `30.42` ($1.00/day == $30.42/month) for the query `yesterday` (1 day) will split and distribute exactly $1.00 across the allocations. Default is `0.0.`
shareTenancyCosts	boolean	If `true`, share the cost of cluster overhead assets such as cluster management costs and node attached volumes across tenants of those resources. Results are added to the sharedCost field. Both cluster management and attached volumes are shared by cluster. Default is `true`.
idleByNode	boolean	If `true`, idle allocations are created on a per node basis, which will result in different values when shared and more idle allocations when split. Default is `false`.
splitIdle	boolean	If `true`, and `shareIdle == false`, idle allocations are created on a per cluster or per node basis rather than being aggregated into a single idle allocation. Default is `false`.
reconcile	boolean	If `true`, pulls data from the Assets cache and corrects prices of Allocations according to their related Assets. The corrections from this process are stored in each cost categories cost adjustment

Name	Type	Description
		field. If the integration with your cloud provider's billing data has been set up, this will result in the most accurate costs for Allocations. Default is `true`.

200: OK

```
{
    "code": 200,
    "data": {
        "step": ,
        "sets": [
            {
                "allocationTrends": {
                    "aggregator": {
                        "trends": {
                            "costs": {
                                "totalCost": {
                                    "relativeChange": {
                                        "isInfinite": false,
                                        "isNaN": false,
                                        "value":
                                    }
                                }
                            }
                        }
                    },
                },
                "window": {
                    "start": "",
                    "end": ""
                }
            }
        ]

    }
}
```

Calculating trend value

The Allocation 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 the previous window, 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 allocation data must be available and sampled in order to provide an accurate value.

The equation for calculating `value` is: `value=current/previous - 1`

Receiving a positive `value` means your more recent `totalCost` has increased compared to the previous window. 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 `-0.27...`, meaning spending has decreased in the current window by roughly 27% from the previous window.

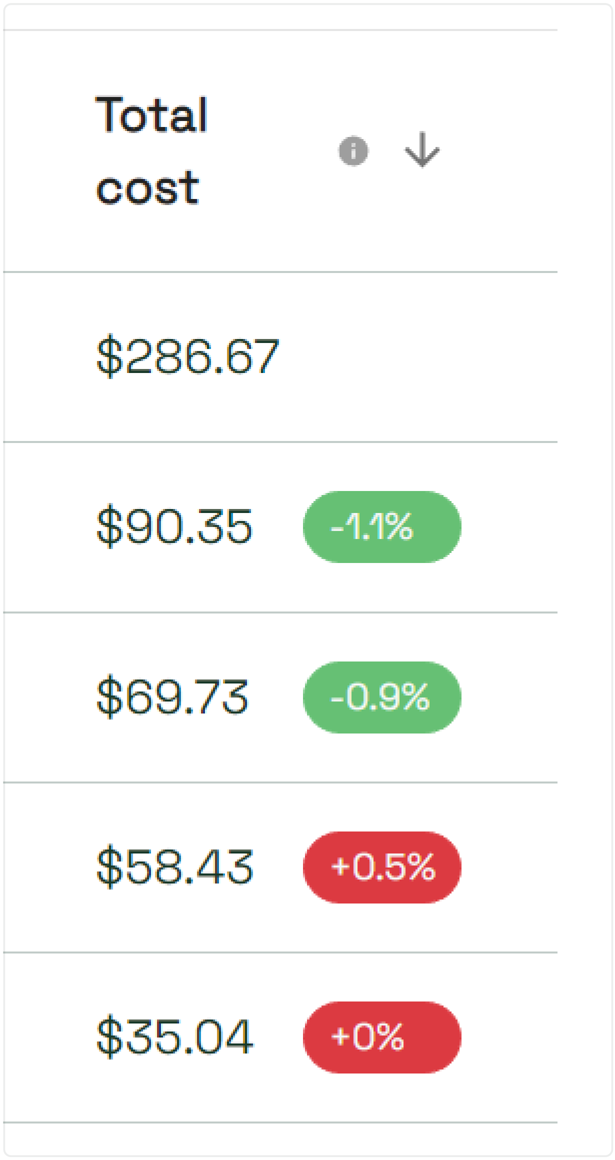
```
                  "allocationTrends": {
                      "aggregator": {
                          "trends": {
                              "costs": {
                                  "totalCost": {
                                      "relativeChange": {
                                          "isInfinite": false,
                                          "isNaN": false,
                                          "value": -0.2786223889995989
                                      }
                                  }
                              }
                          }
                      },
```

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=namespace`, the user should provide a comma-separated list of all namespaces they wish to see trend values for in this category. In this case, they should provide `names=kubecost,aggregator,kube-system...` to receive a list of trend values for all provided namespaces as ordered. If the user does not provide a value for `aggregate`, they must still use the names parameter to list all line items requested.

Viewing trend percentages in the UI

Trend values are converted into percentages in the Kubecost Allocations 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.



Total cost	
$286.67	
$90.35	-1.1%
$69.73	-0.9%
$58.43	+0.5%
$35.04	+0%

Total cost column

> Requests with large time intervals for `window` can result in an error. The recommended maximum interval for `window` is 7 days. A failed response will show a *N/A* inside a gray bubble in the UI with no percentage returned.

The Trends API does not currently support cost comparisons besides `totalCost`.