High Dataflow Billing for a particular time

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Contents

Issue

Customer reports high billing for a particular date.

Root Cause

We should check what is the usage on that particular day. User is charged when activities are finished. If customer is charged more for specified period of time, check if for those days either more activities ran or ran for more time.

Resolution

Query to see how many runs finished on particular time window. You can change start/end dates as per your requirements and slice the data in different bins, like here it is sliced at a day level.

KustoLink

```
let subscription="Replace subscription id";
let startDate=datetime(2021-01-01 04:59:59.5927197);
let endDate=datetime(2021-01-08 04:59:59.5927197);
cluster('adfneu.kusto.windows.net').database("AzureDataFactory").ActivityRuns
| union cluster('adfcus.kusto.windows.net').database("AzureDataFactory").ActivityRuns
| where subscriptionId =~ subscription
| where TIMESTAMP between (startDate..endDate)
| where status !in ("Queued", "InProgress")
| where activityType == "ExecuteDataFlow"
| summarize count() by status, bin(TIMESTAMP, 1d)
```

In case you want to know the duration along with the when activities ended.

KustoLink

```
let subscription="Replace subscription id";
let startDate=datetime(2021-01-01 04:59:59.5927197);
let endDate=datetime(2021-01-08 04:59:59.5927197);
ActivityRuns
| where subscriptionId =~ subscription
| where TIMESTAMP between (startDate..endDate)
| where status !in ("Queued", "InProgress")
| where activityType == "ExecuteDataFlow"
| where errorCode != 4502 // Removing throttled dataflow, for which customer is not charged with Dataflow Mete
| where failureType != "SystemError" // Removing system error, for which customer is not charged.
| join (ActivityRuns
| where subscriptionId =~ subscription
| where TIMESTAMP between ((startDate-1d)..endDate)
| where status == "InProgress"
| where activityType == "ExecuteDataFlow"
) on activityRunId
| extend duration=TIMESTAMP - TIMESTAMP1, endTime=TIMESTAMP
| summarize count(), sum(duration) by bin(endTime, 1d)
```

In case you want to aggregate by different time period, or activity name, factory name, pipeline name etc, that can be done via changing above queries.

To get the how many cores and compute type customer is using, find example pipeline run id and run the following query.

```
AdfTraceEvent
| where TraceCorrelationId == "0a2640fd-a3b1-4981-9f5e-4f3c65a91861" // Replace your pipeline run id here
| where Message contains "computeType"
```

Following kind of message can point to which kind of computeType and coreCount customer is using. "*Returning Compute:DataflowComputeProperties= computeType:MemoryOptimized, coreCount:272, nodeCount:0, timeToLive:0, maxConcurrency:*" From above query and core count we can get the units with formula coreCount*durationInHours.

Additional Information:

- Icm Reference: N/A
- Author: Anudeep Sharma
- Reviewer: Anudeep Sharma
- Keywords:

How good have you found this content?