Metric Extension – gv$active_session_history to csv

December 2014

Primary Author: Karl Arao

Table of Contents

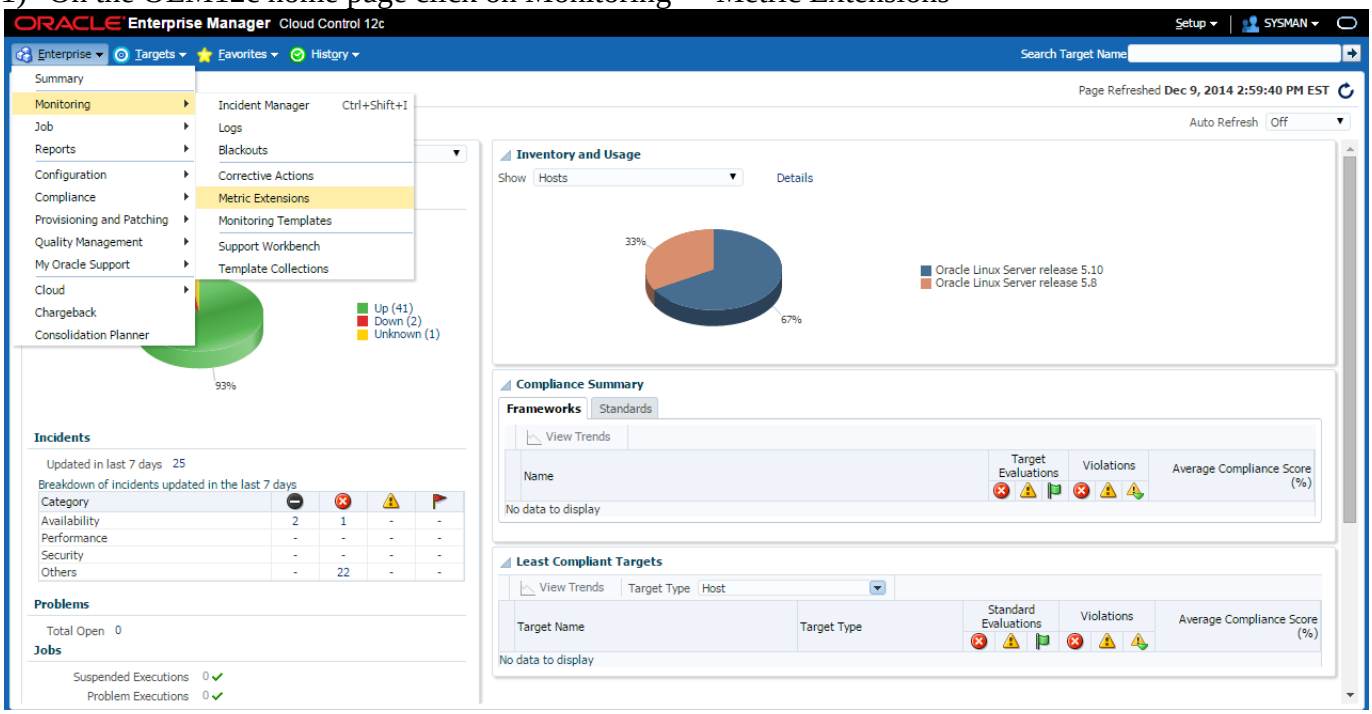
I. Overview

The Active Data Guard (ADG) can only store historical information in memory because of its “read only” state. Unlike the primary database where all the in memory performance data gets flushed periodically to Automatic Workload Repository (AWR) for later reporting, in ADG all data is gone once the buffer gets full and recycled for newer data. One workaround we did is to dump the Active Session History (ASH) data to filesystem and later on analyzed with a desktop analytics tool. This document shows how to pull the ASH data across the ADG databases through a Metric Extension and store it inside the Oracle Enterprise Manager (OEM) repository. The following are the advantages of doing this:

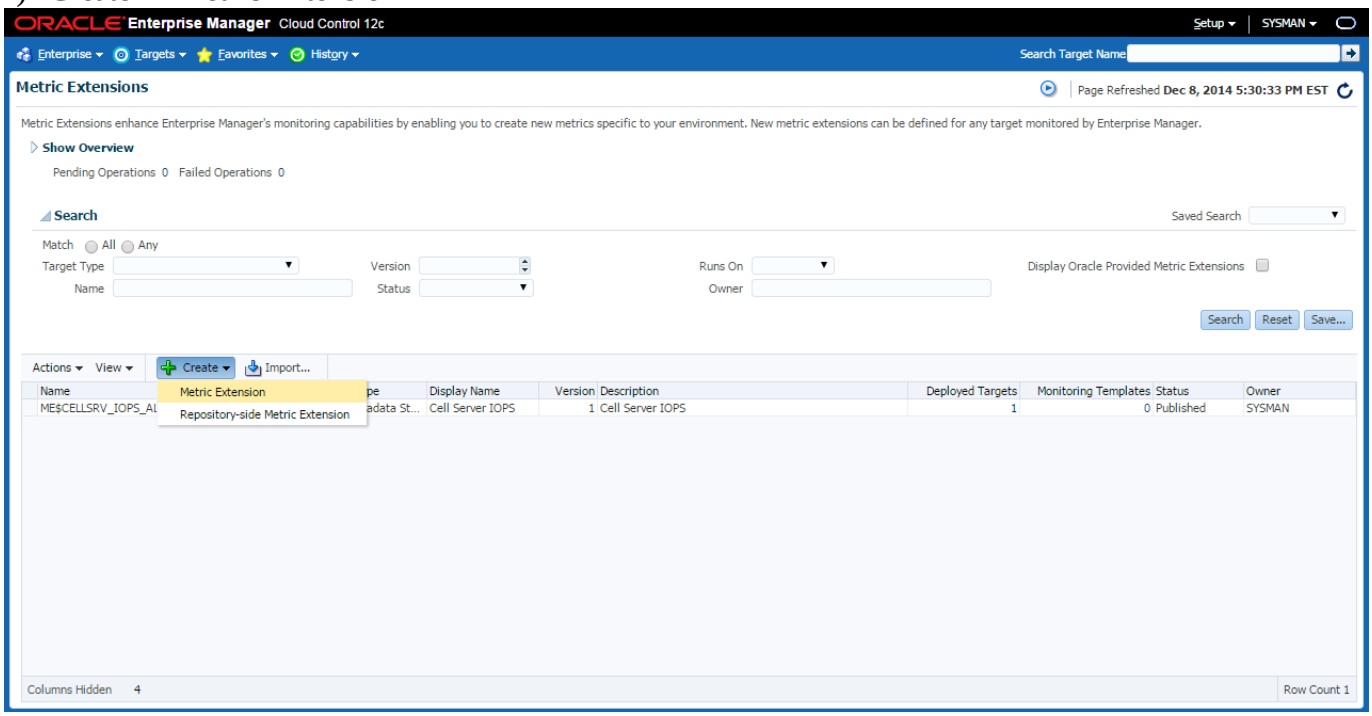
- OEM repository will be responsible of aging out the data (default: 32 days retention)
- No more overhead on the server filesystem
- Easier data access (though SQL*Developer or SQL*Plus)
- Easy to scale – deploy the Metric Extension to the new database and performance data will be automatically stored and purged

II. Create Metric Extension

1) On the OEM12c home page click on Monitoring -> Metric Extensions

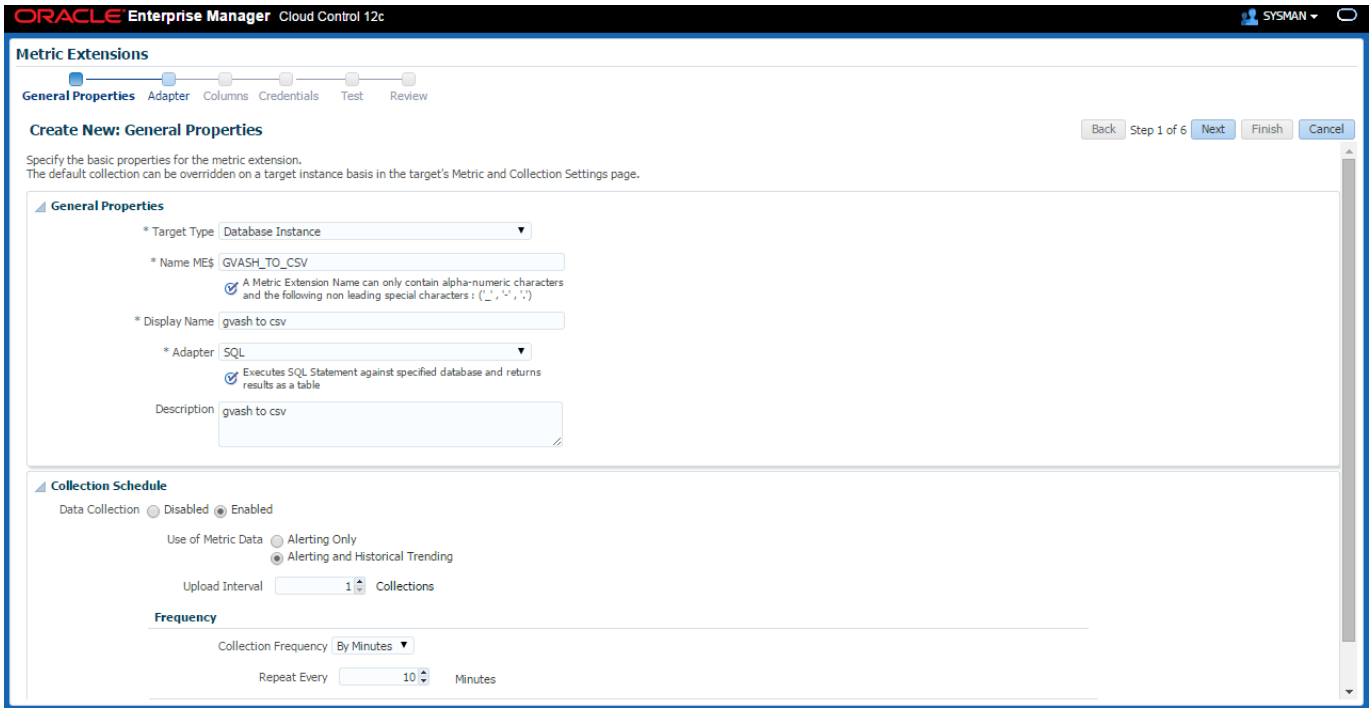


2) Create -> Metric Extension

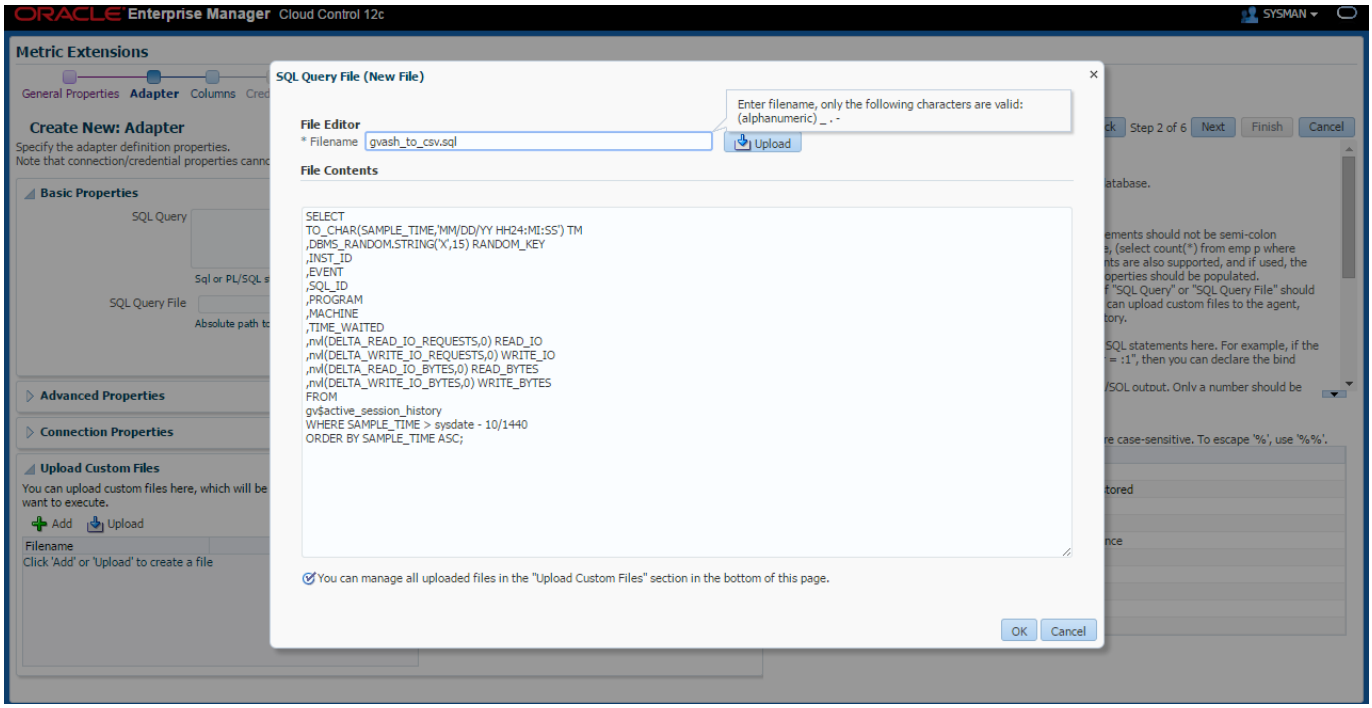


3) On General Properties and Collection Schedule
   a. Target Type: Database Instance
   b. Name ME$: GVASH_TO_CSV
   c. Display Name: gvash to csv
   d. Adapter: SQL

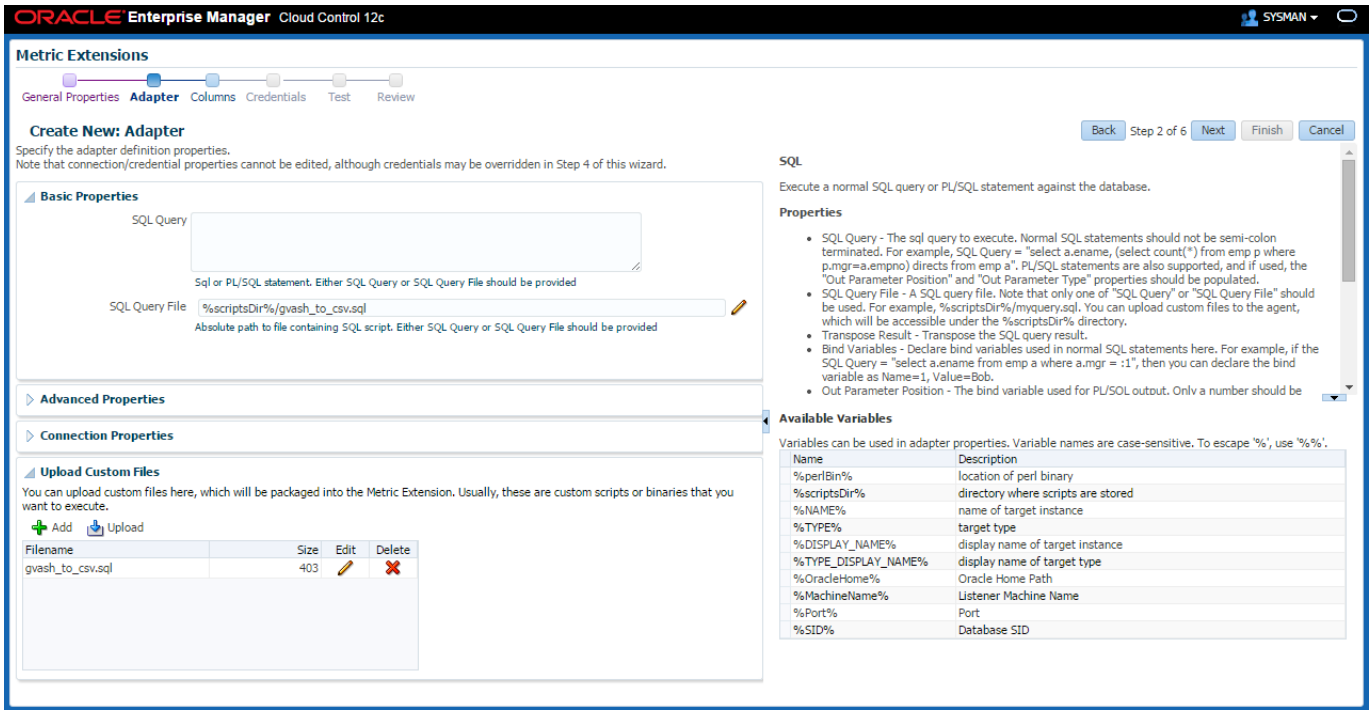
e. Description: gvash to csv
f. Repeat Every: change to 10 minutes



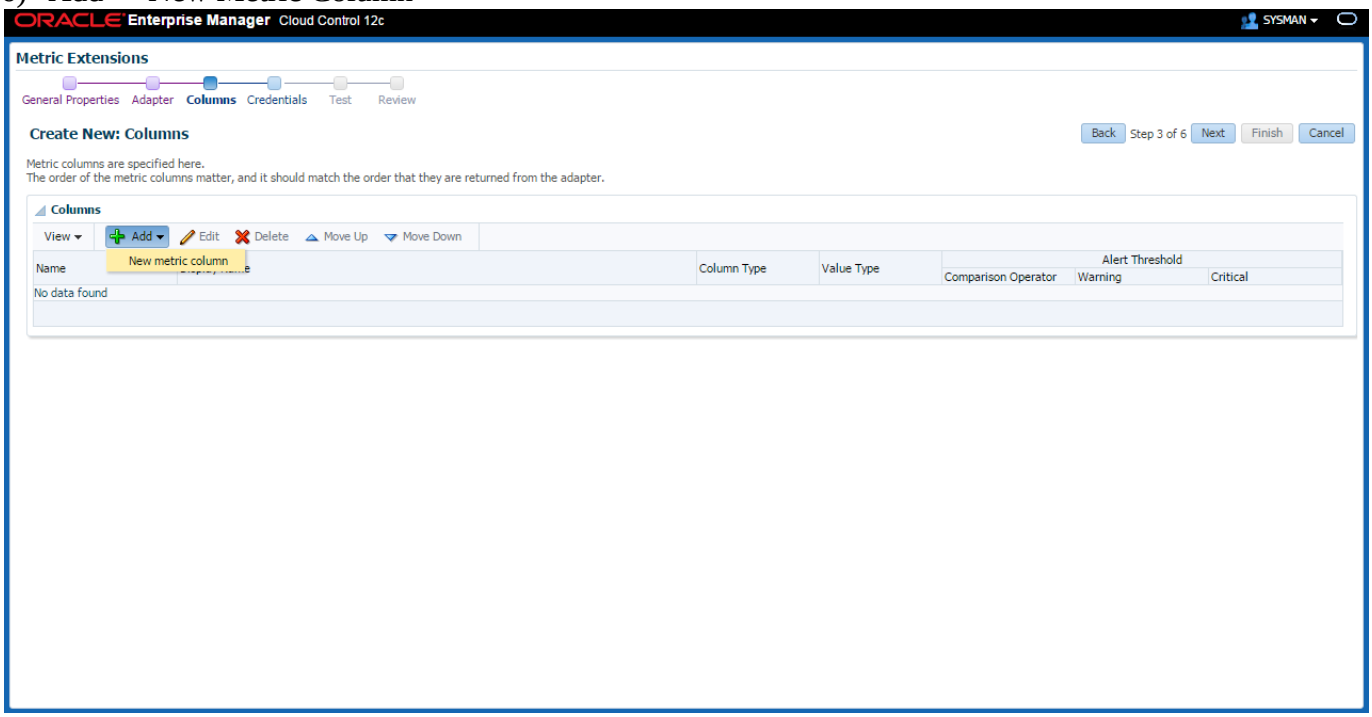
4) Click on the pencil -> upload the gvash_to_csv.sql script (below) -> OK



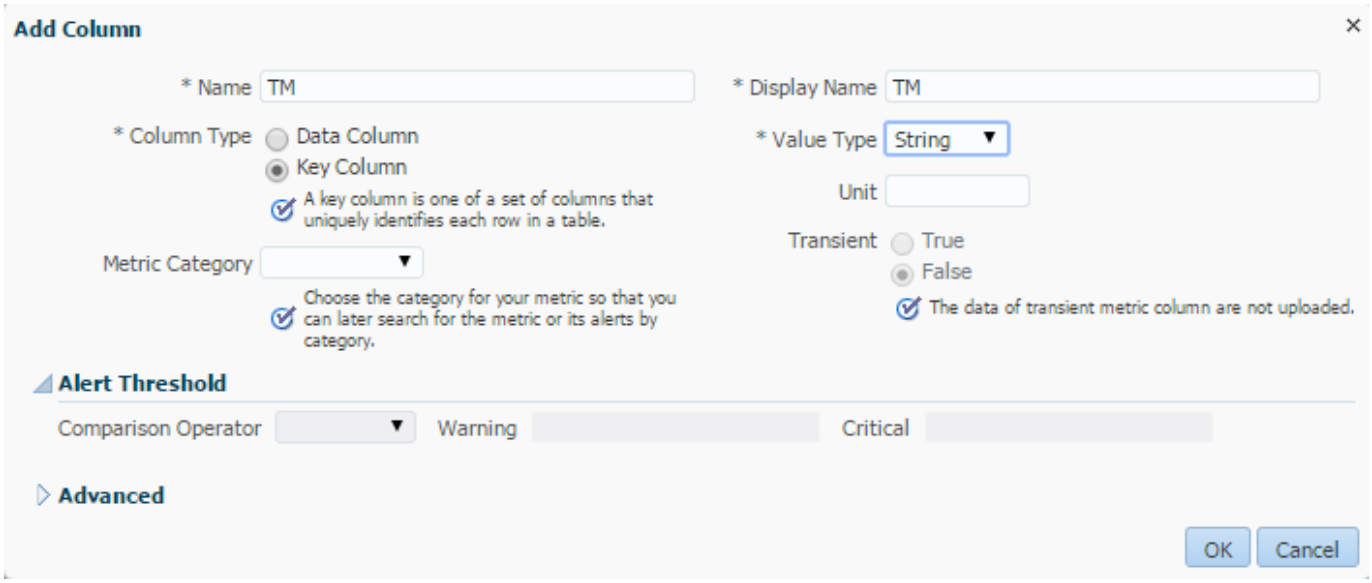
5) Next



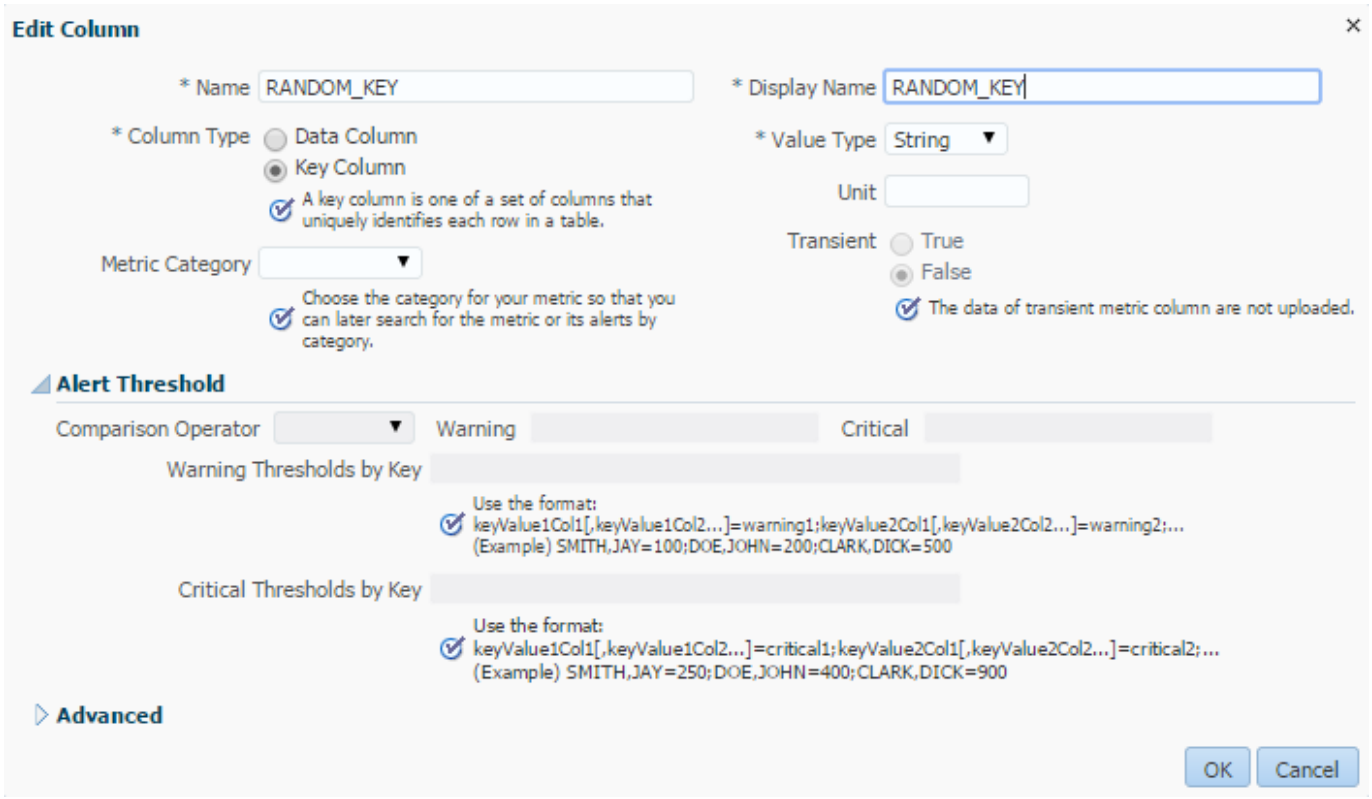
6) Add -> New Metric Column



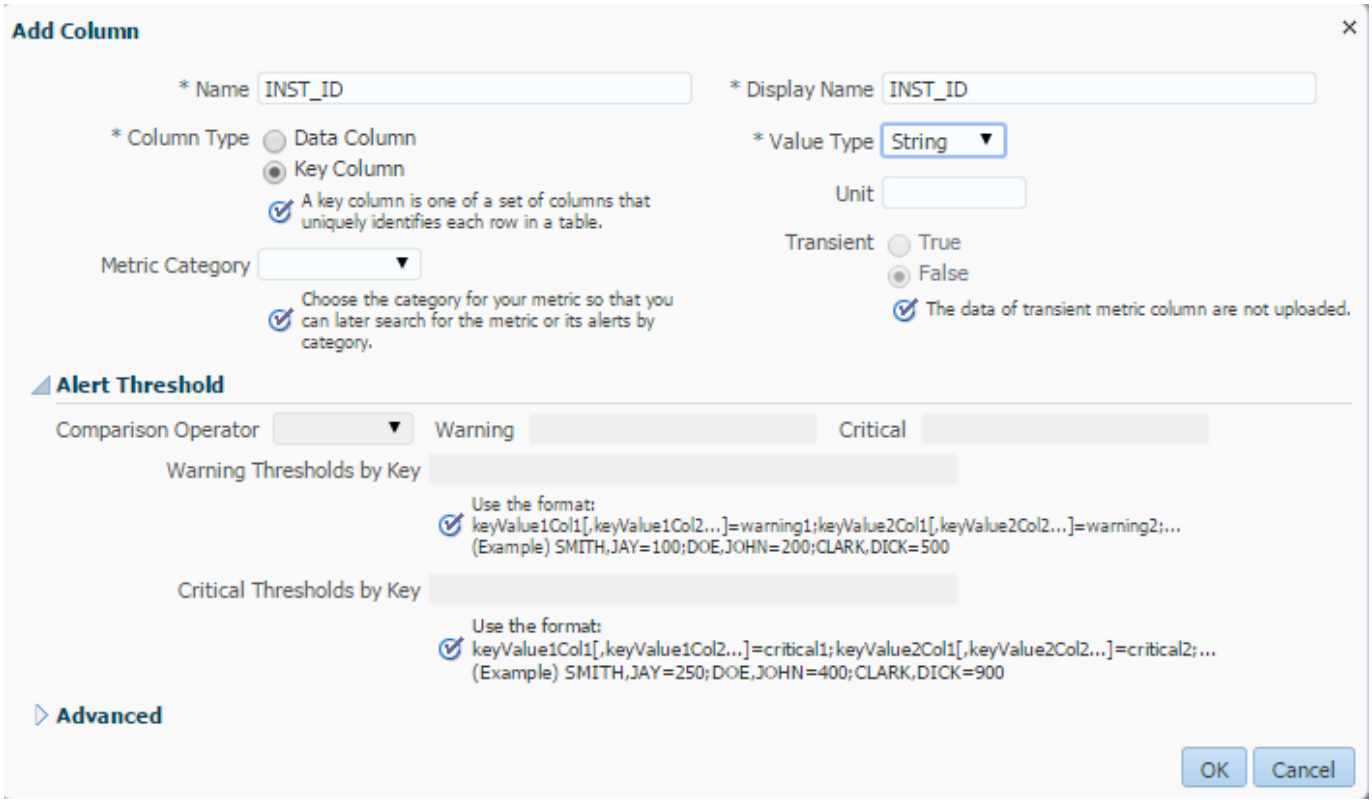
7) On Add Column: TM
   a. Name: TM
   b. Display Name: TM
   c. Column Type: Key Column
   d. Value Type: String



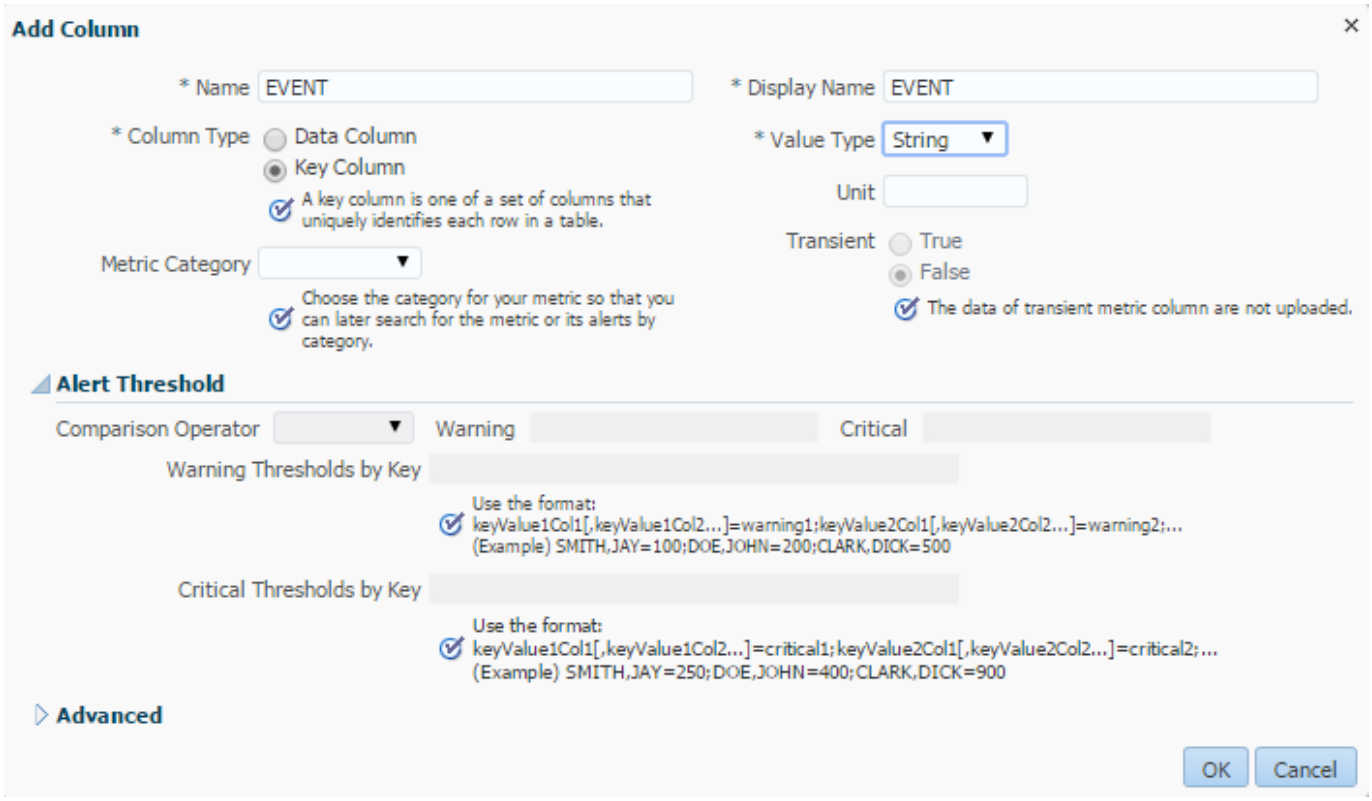
8) On Add Column: RANDOM_KEY
   a. Name: RANDOM_KEY
   b. Display Name: RANDOM_KEY
   c. Column Type: Key Column
   d. Value Type: String



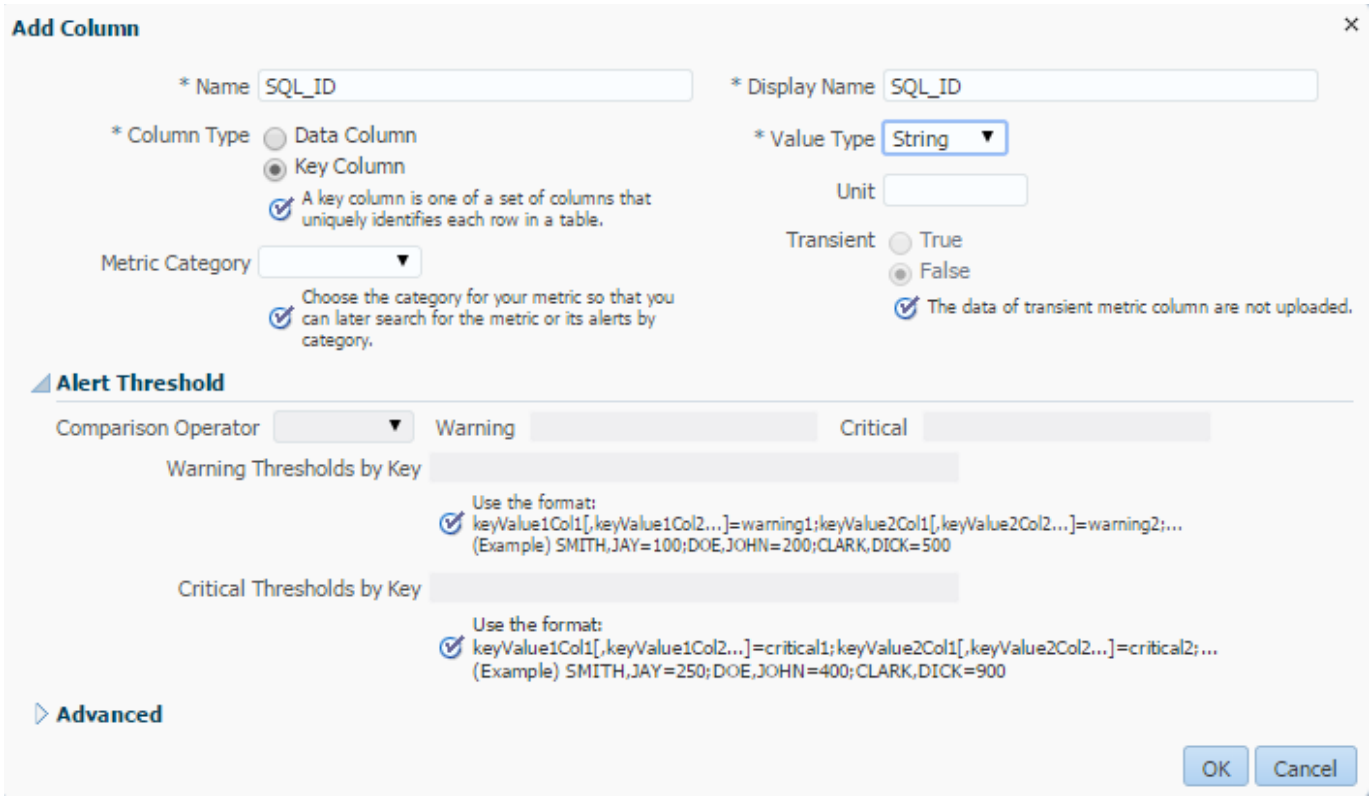
9) On Add Column: INST_ID
   a. Name: INST_ID
   b. Display Name: INST_ID
   c. Column Type: Key Column
   d. Value Type: String



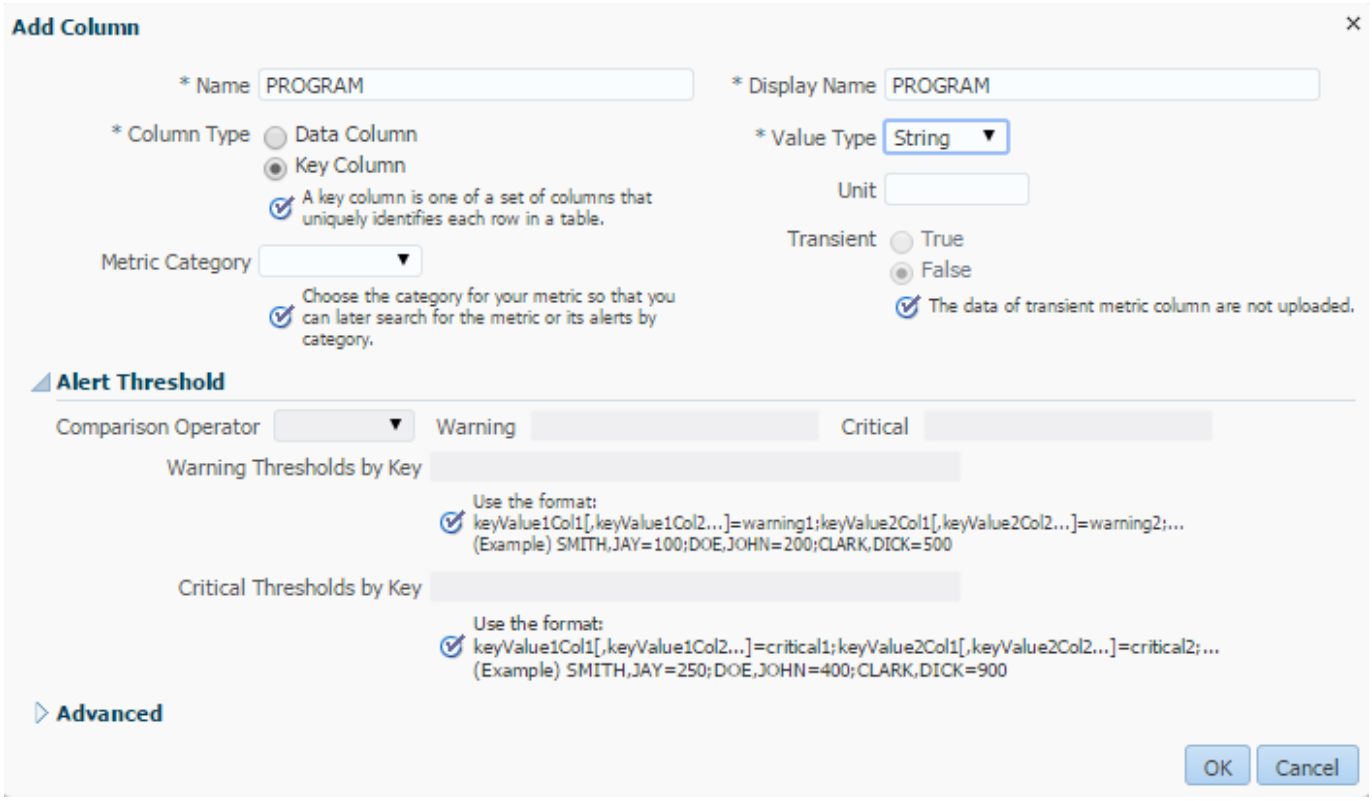
10) On Add Column: EVENT
    a. Name: EVENT
    b. Display Name: EVENT
    c. Column Type: Key Column
    d. Value Type: String



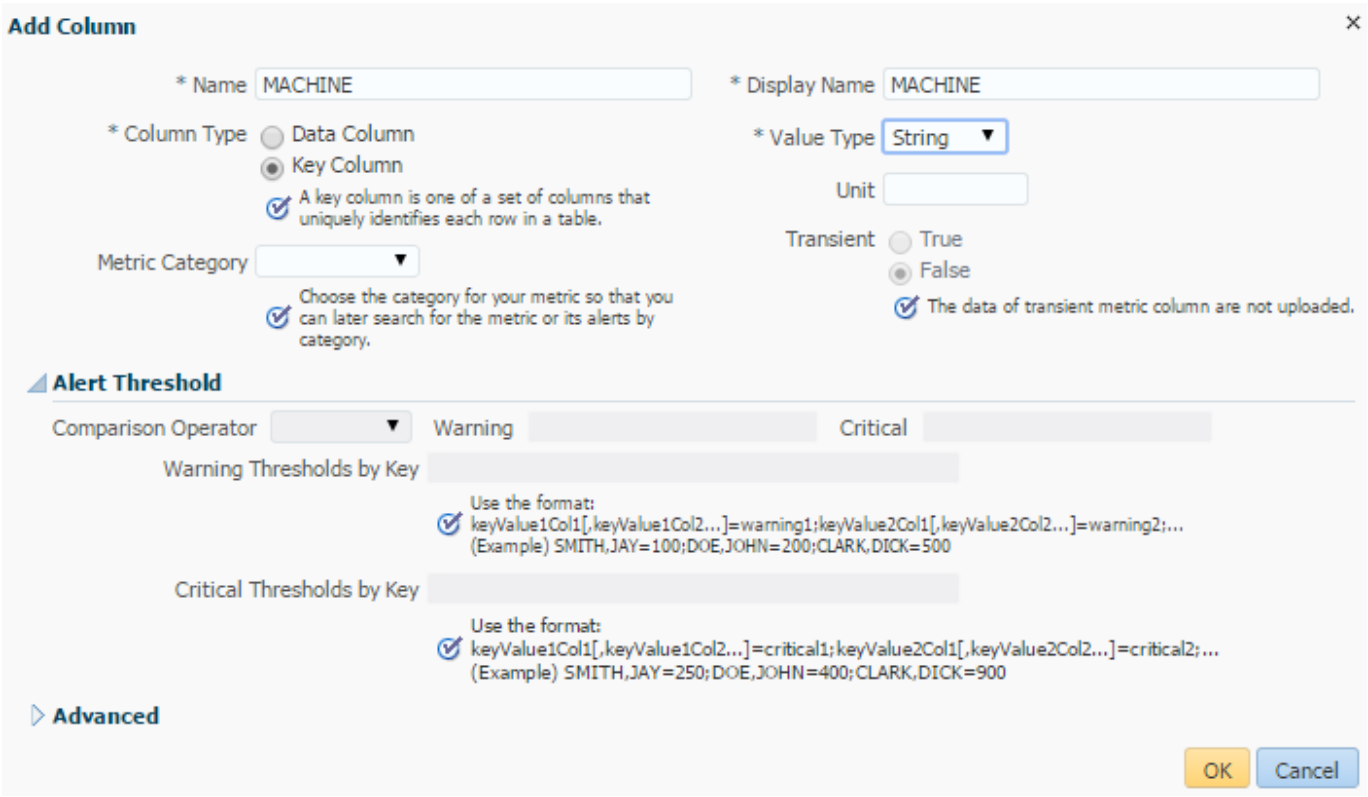
11) On Add Column: SQL_ID
    a. Name: SQL_ID
    b. Display Name: SQL_ID
    c. Column Type: Key Column
    d. Value Type: String



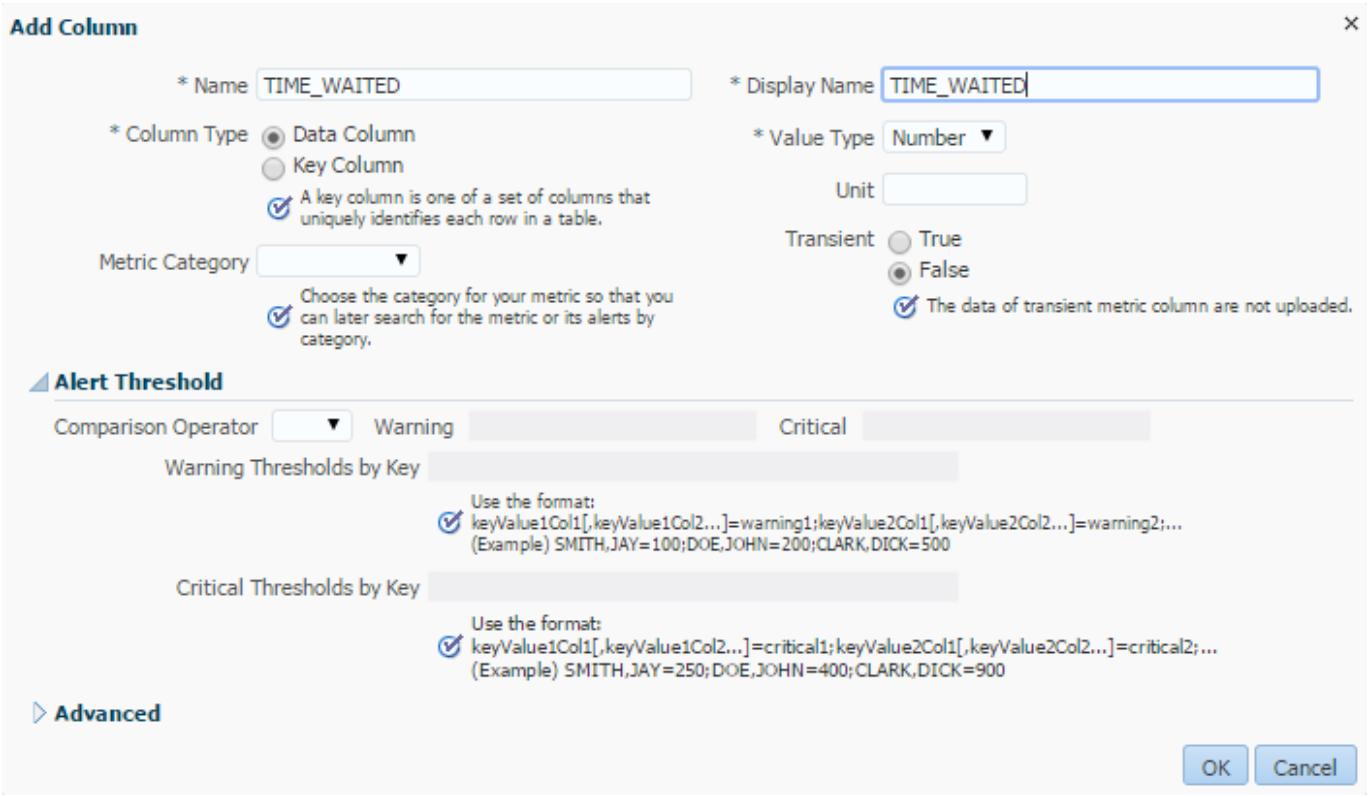
12) On Add Column: PROGRAM
    a. Name: PROGRAM
    b. Display Name: PROGRAM
    c. Column Type: Key Column
    d. Value Type: String



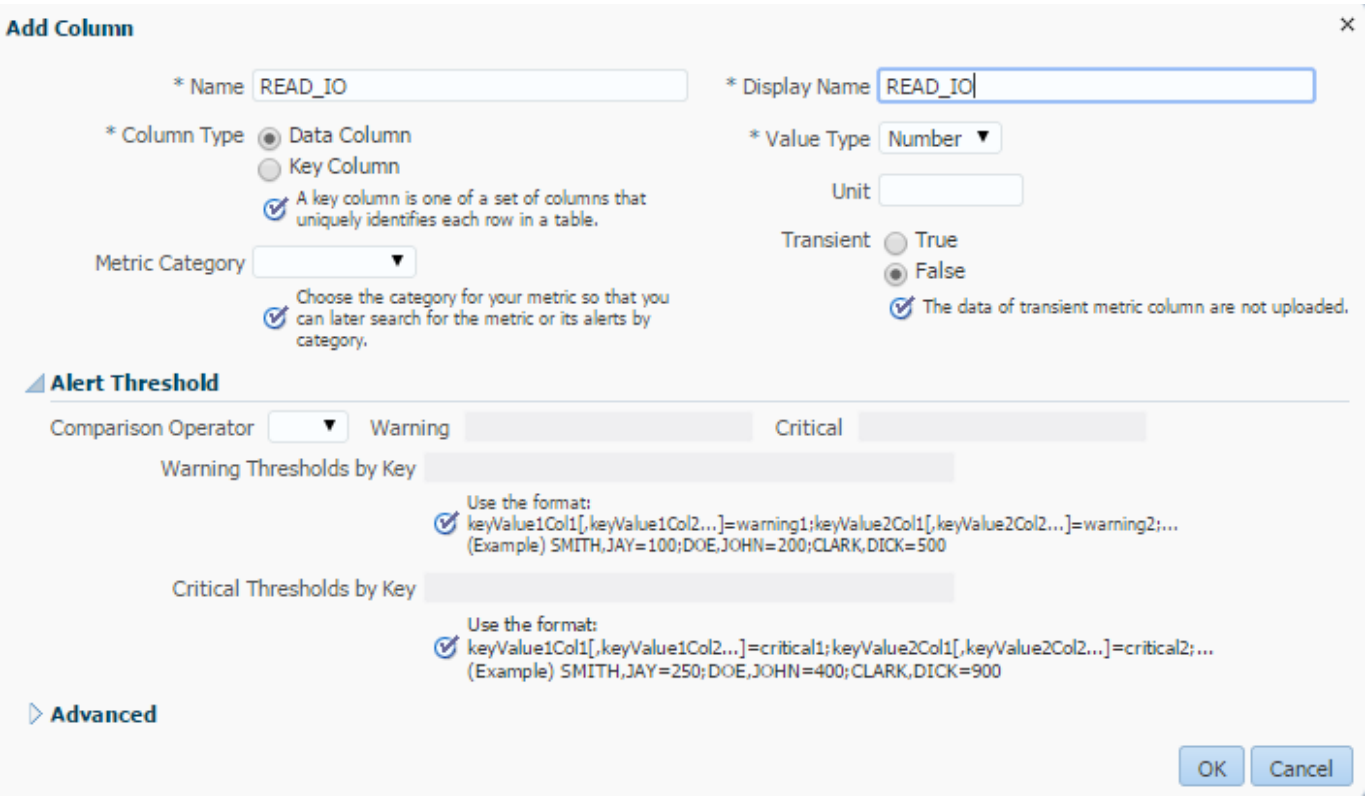
13) On Add Column: MACHINE
    a. Name: MACHINE
    b. Display Name: MACHINE
    c. Column Type: Key Column
    d. Value Type: String



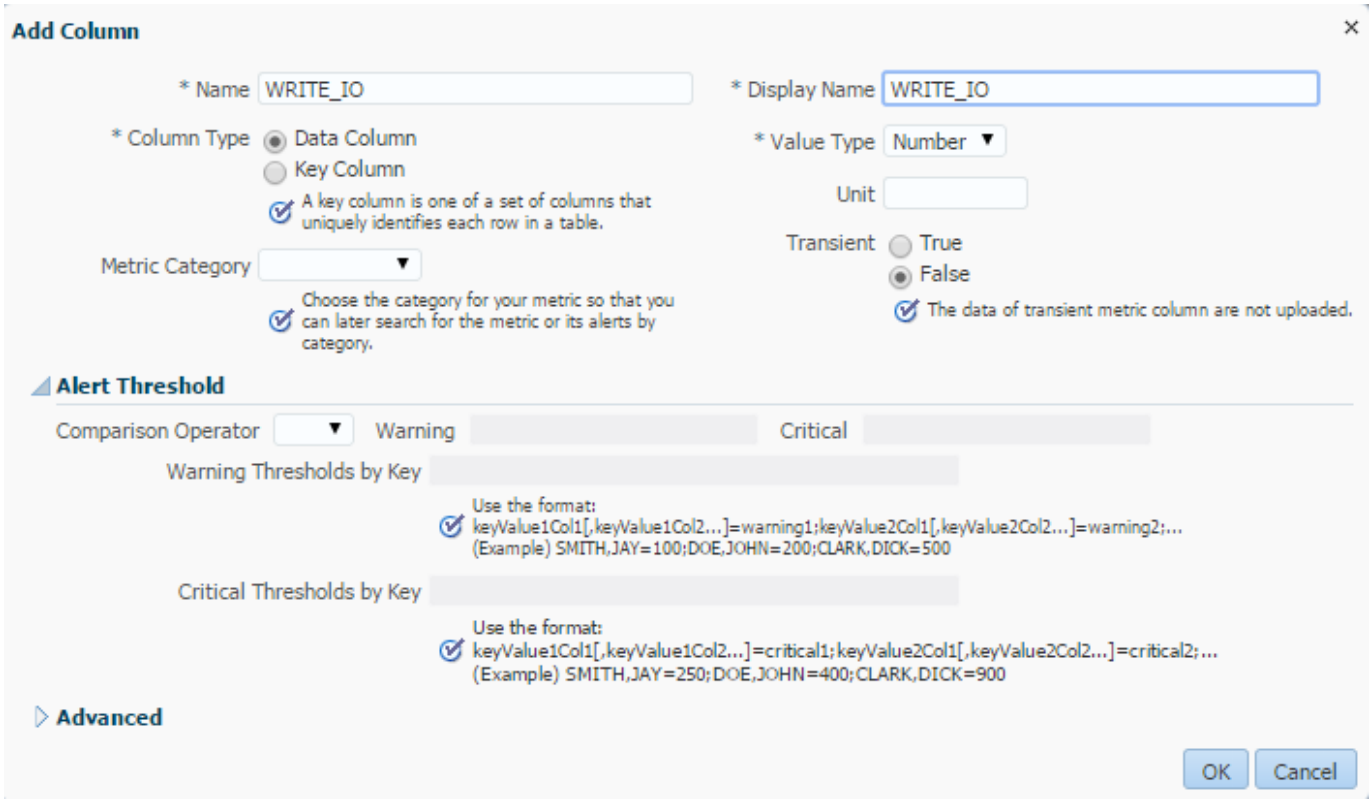
14) On Add Column: TIME_WAITED
   a. Name: TIME_WAITED
   b. Display Name: TIME_WAITED
   c. Column Type: Data Column
   d. Value Type: Number



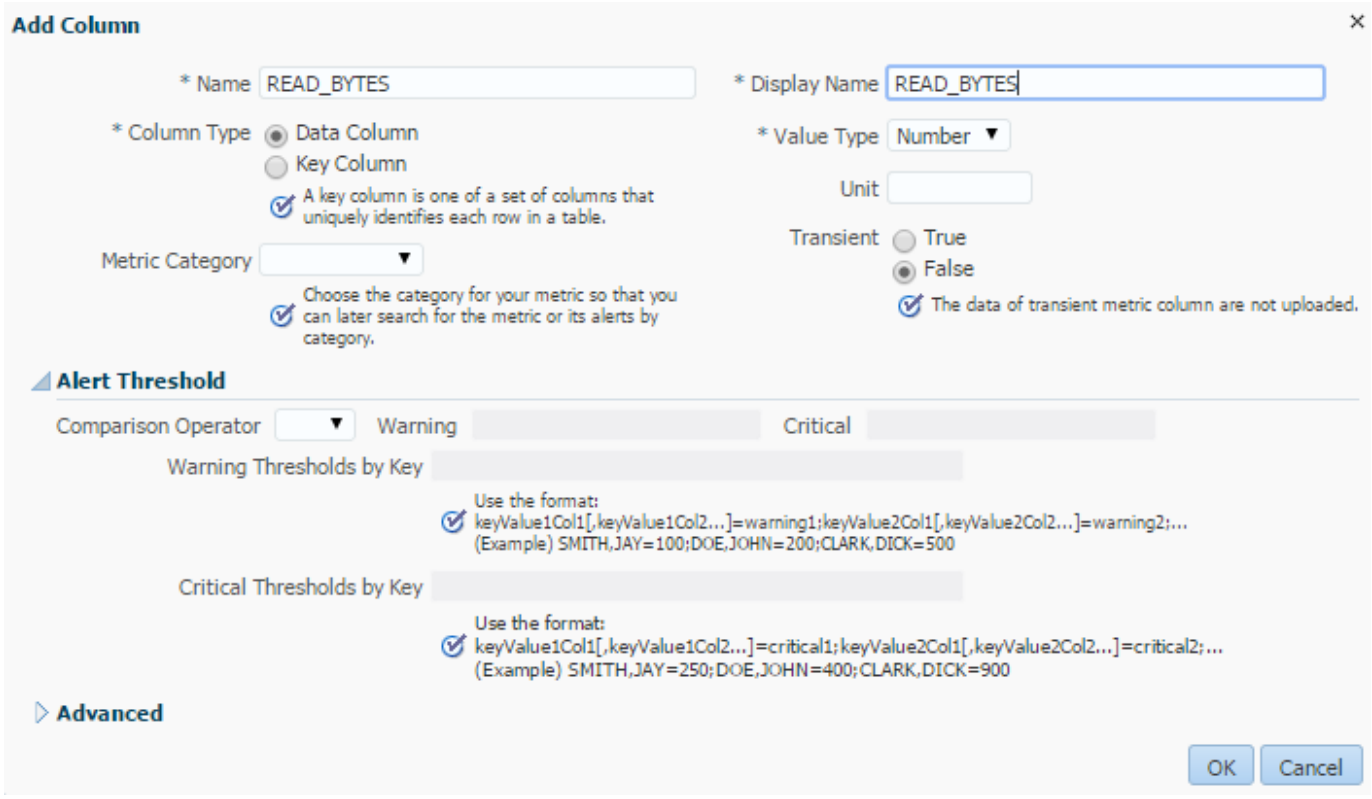
15) On Add Column: READ_IO
    a. Name: READ_IO
    b. Display Name: READ_IO
    c. Column Type: Data Column
    d. Value Type: Number



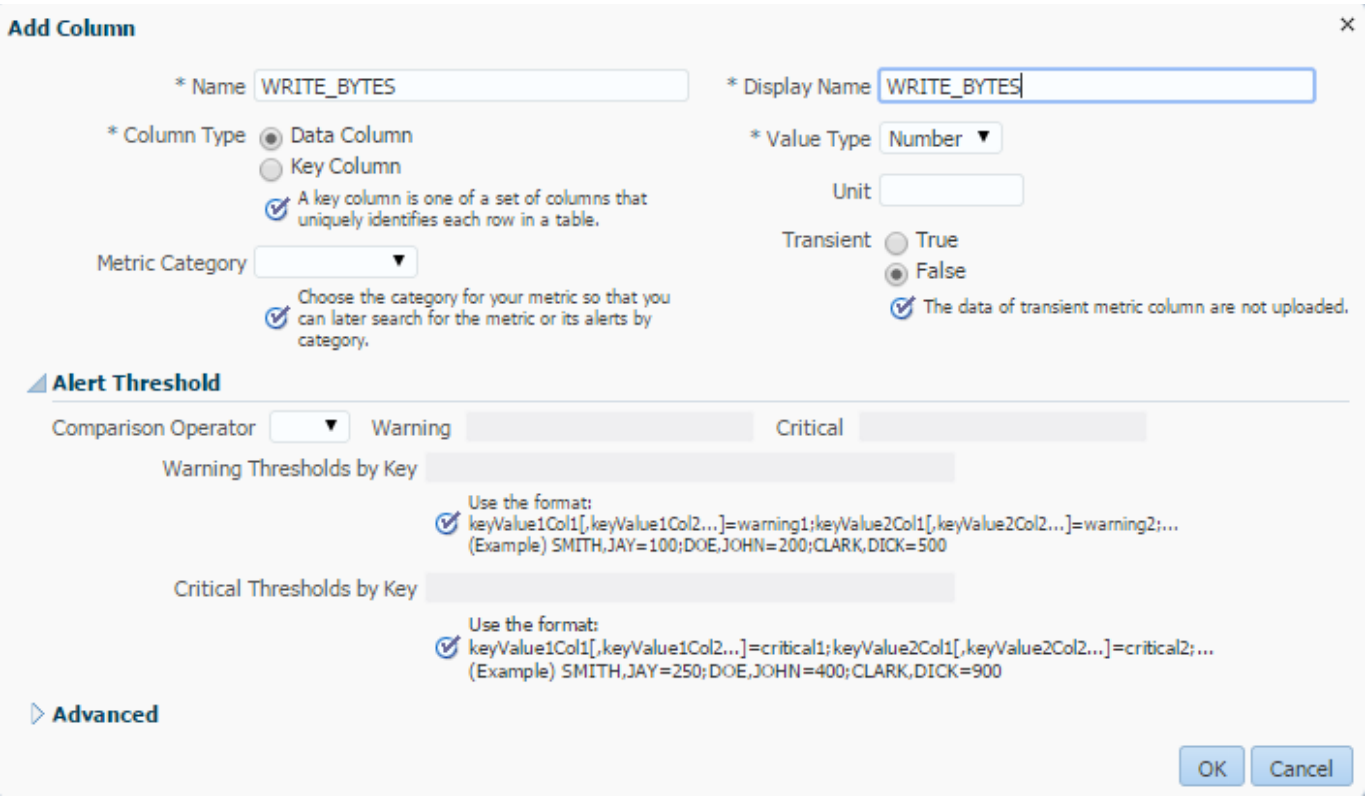
16) On Add Column: WRITE_IO
    a. Name: WRITE_IO
    b. Display Name: WRITE_IO
    c. Column Type: Data Column
    d. Value Type: Number



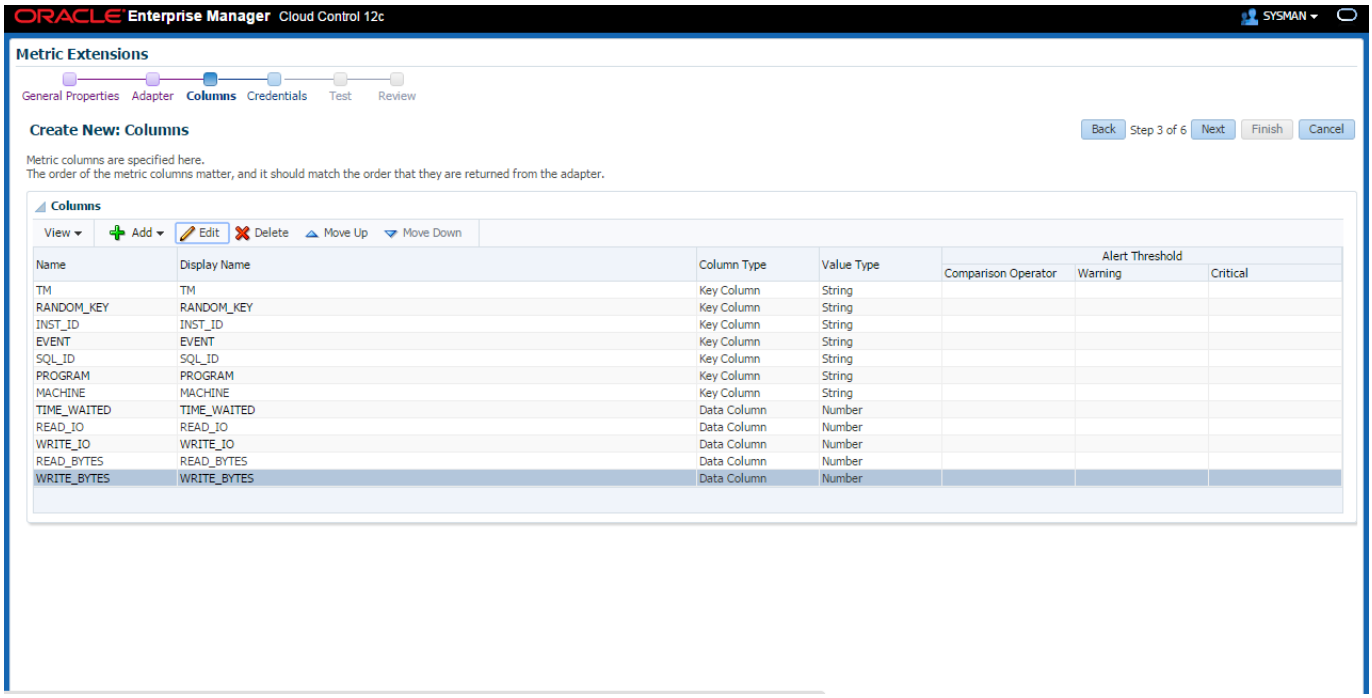
17) On Add Column: READ_BYTES
    a. Name: READ_BYTES
    b. Display Name: READ_BYTES
    c. Column Type: Data Column
    d. Value Type: Number



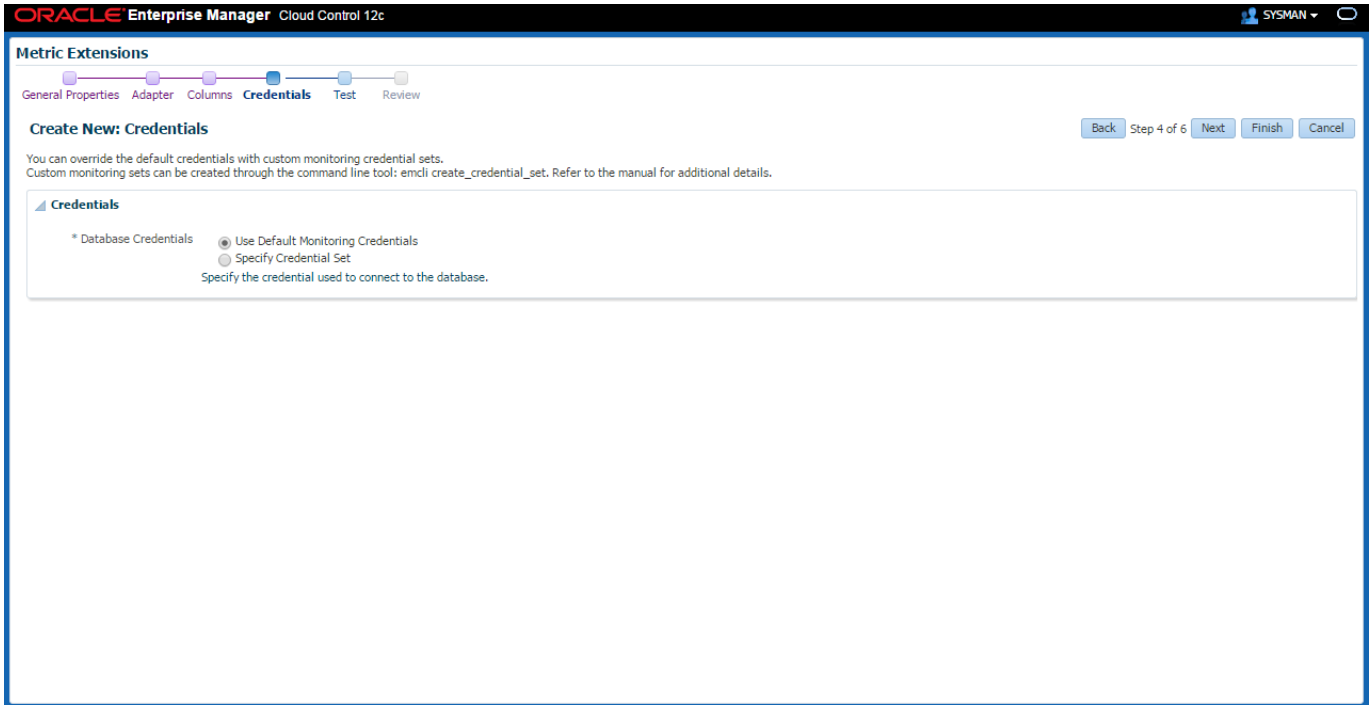
18) On Add Column: WRITE_BYTES
    a. Name: WRITE_BYTES
    b. Display Name: WRITE_BYTES
    c. Column Type: Data Column
    d. Value Type: Number



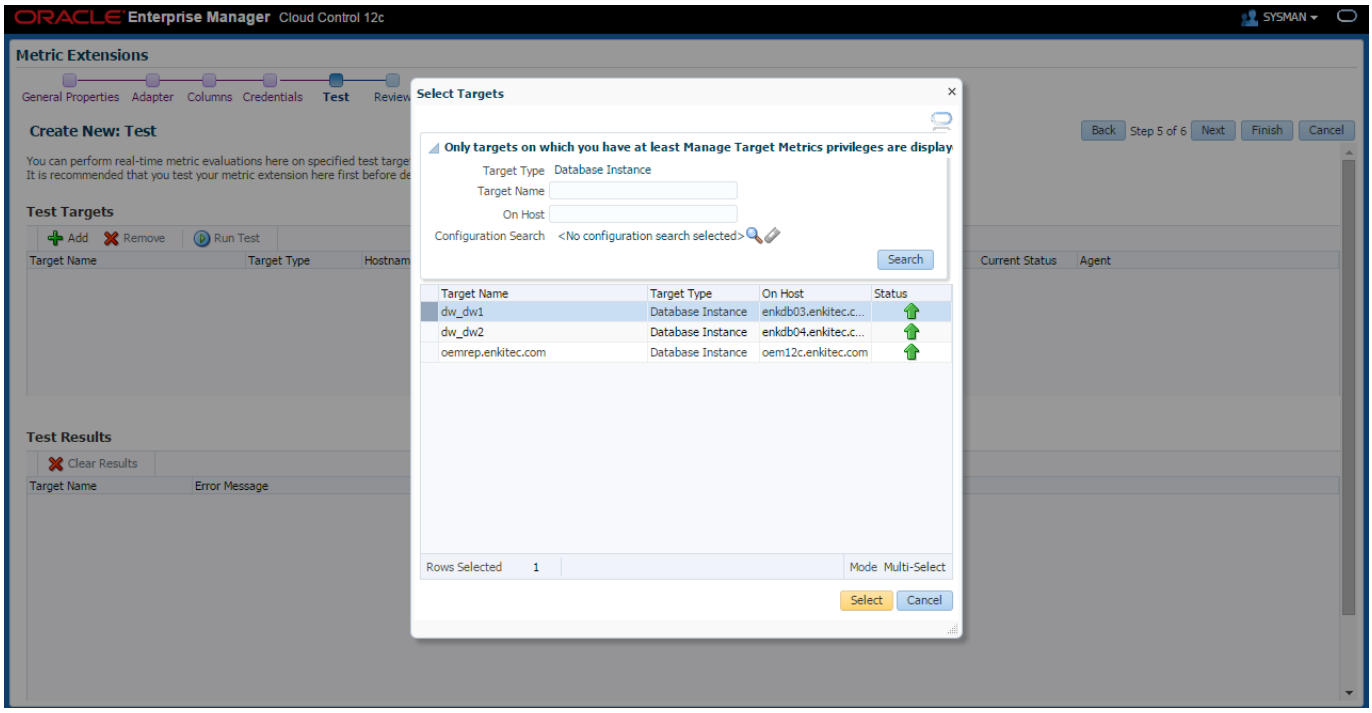
19) There should be a total of 12 columns created
    a. 7 key columns
    b. 5 data columns



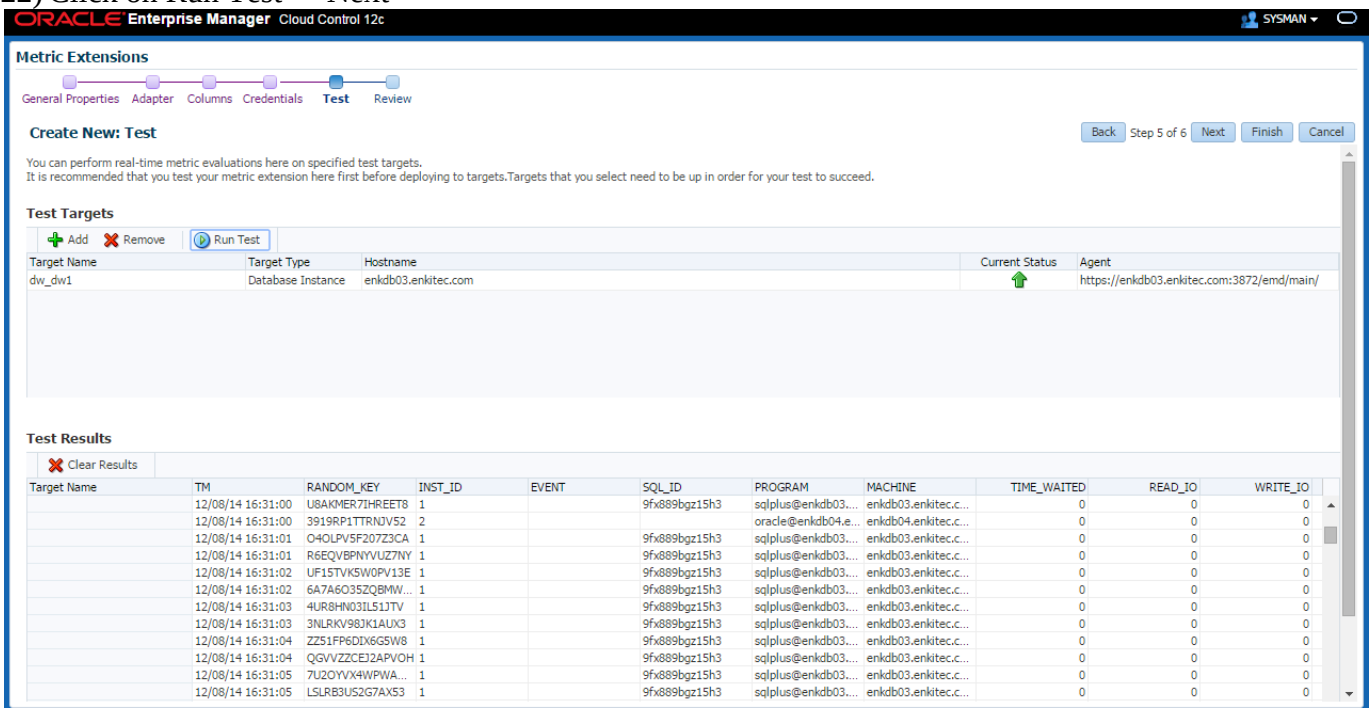
20) Use Default Monitoring Credentials -> Next



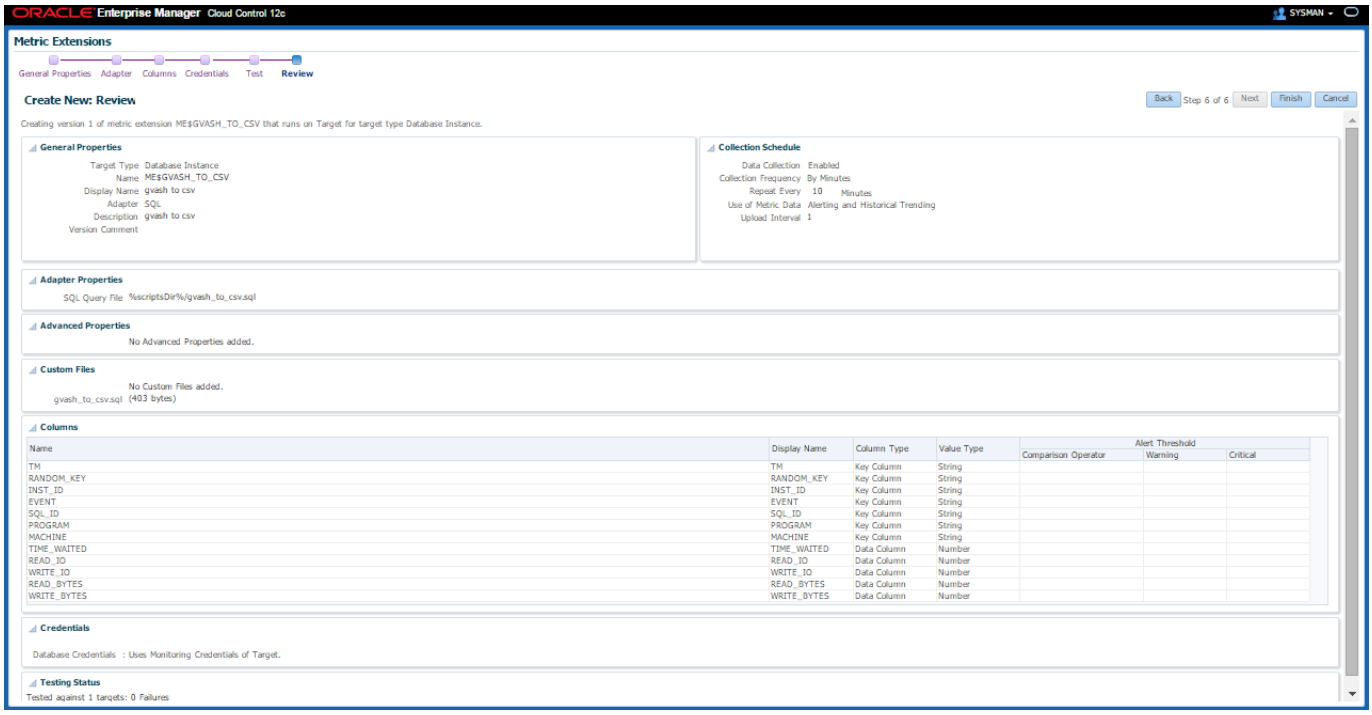
21) Add Test Target -> Select the 1st instance of any cluster database



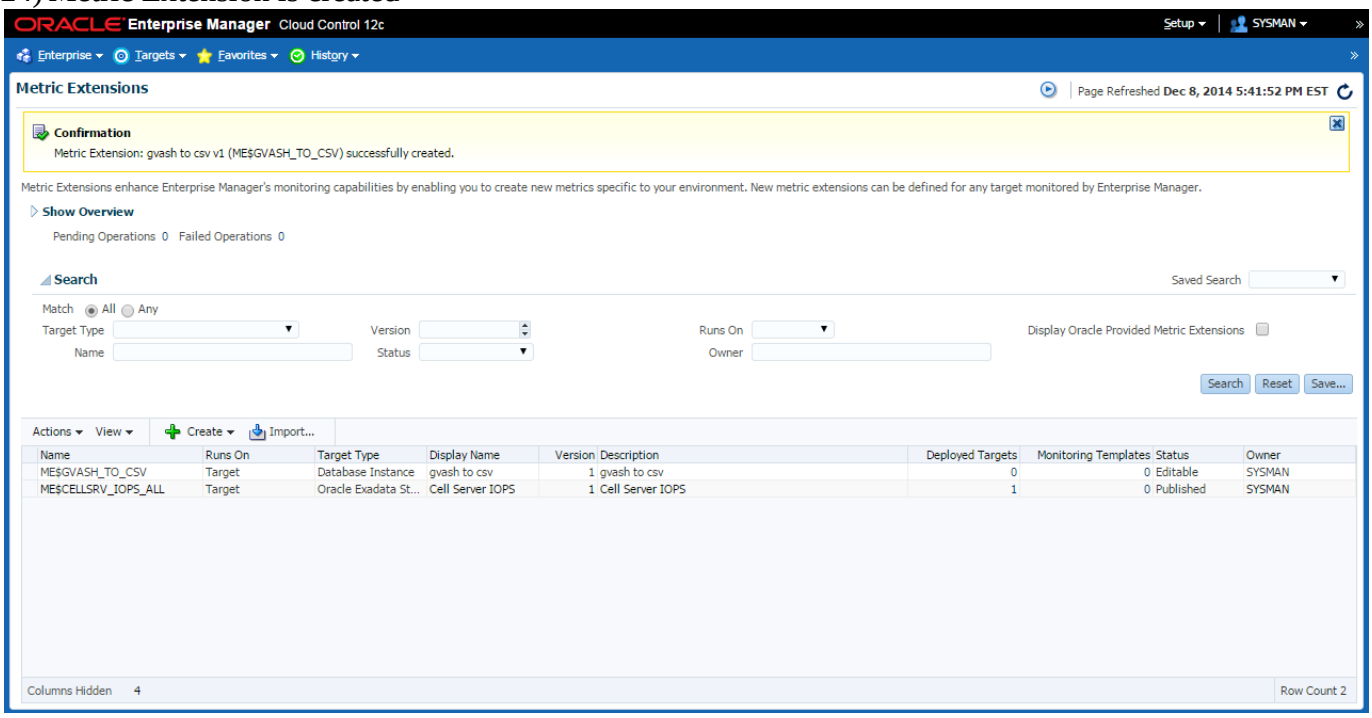
22) Click on Run Test -> Next



23) Review the details -> Finish

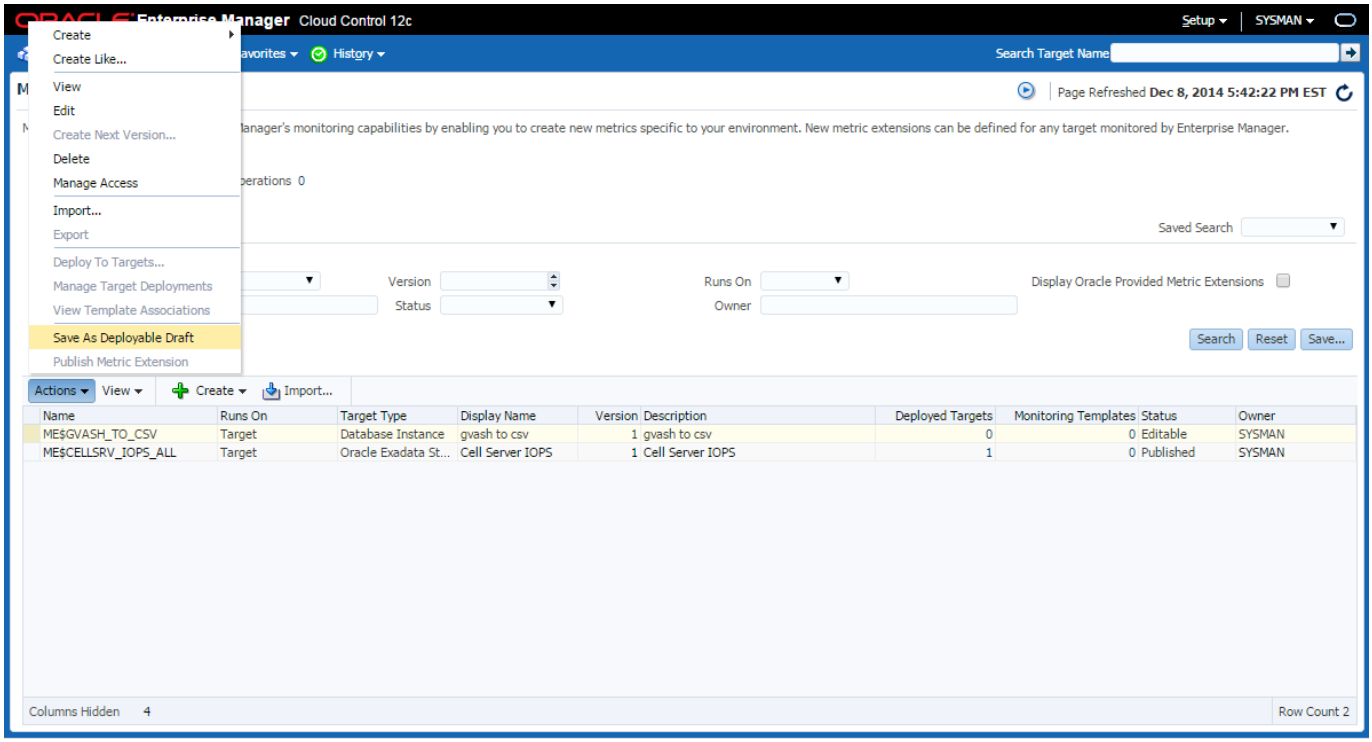


24) Metric Extension is created

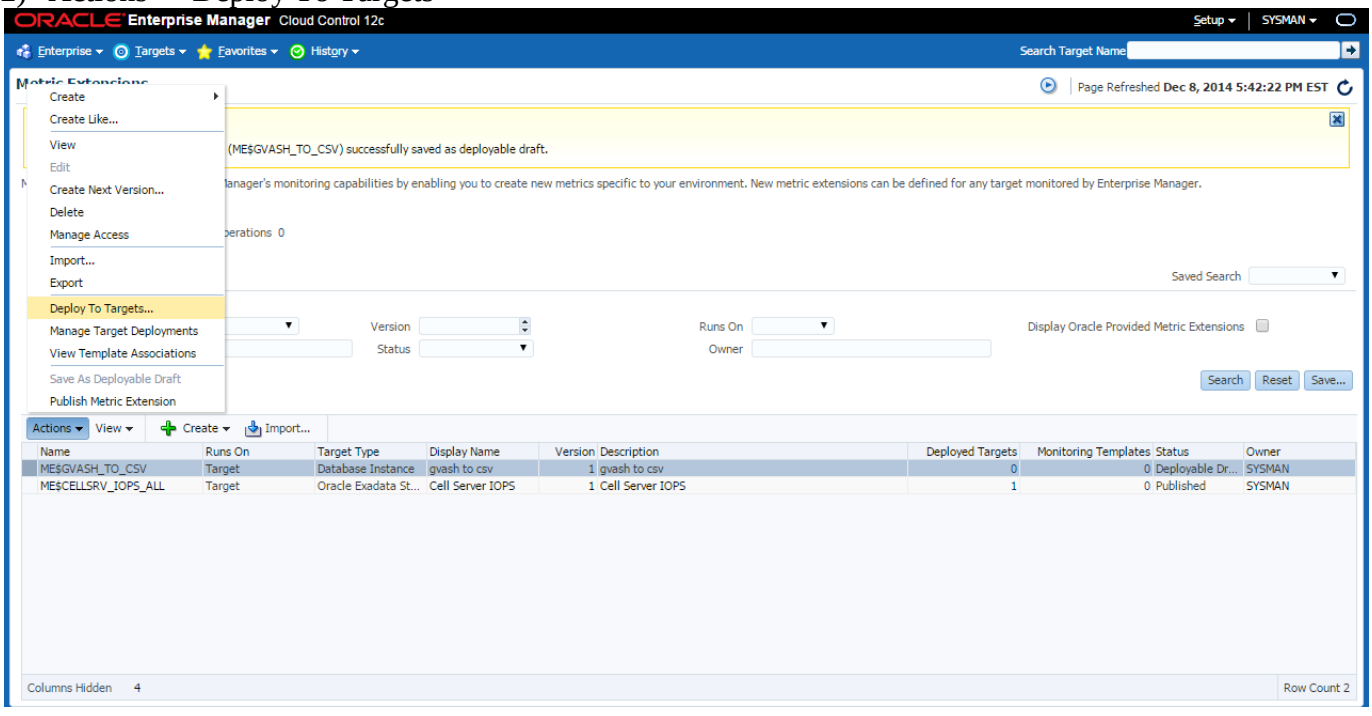


III. Deploy Metric Extension

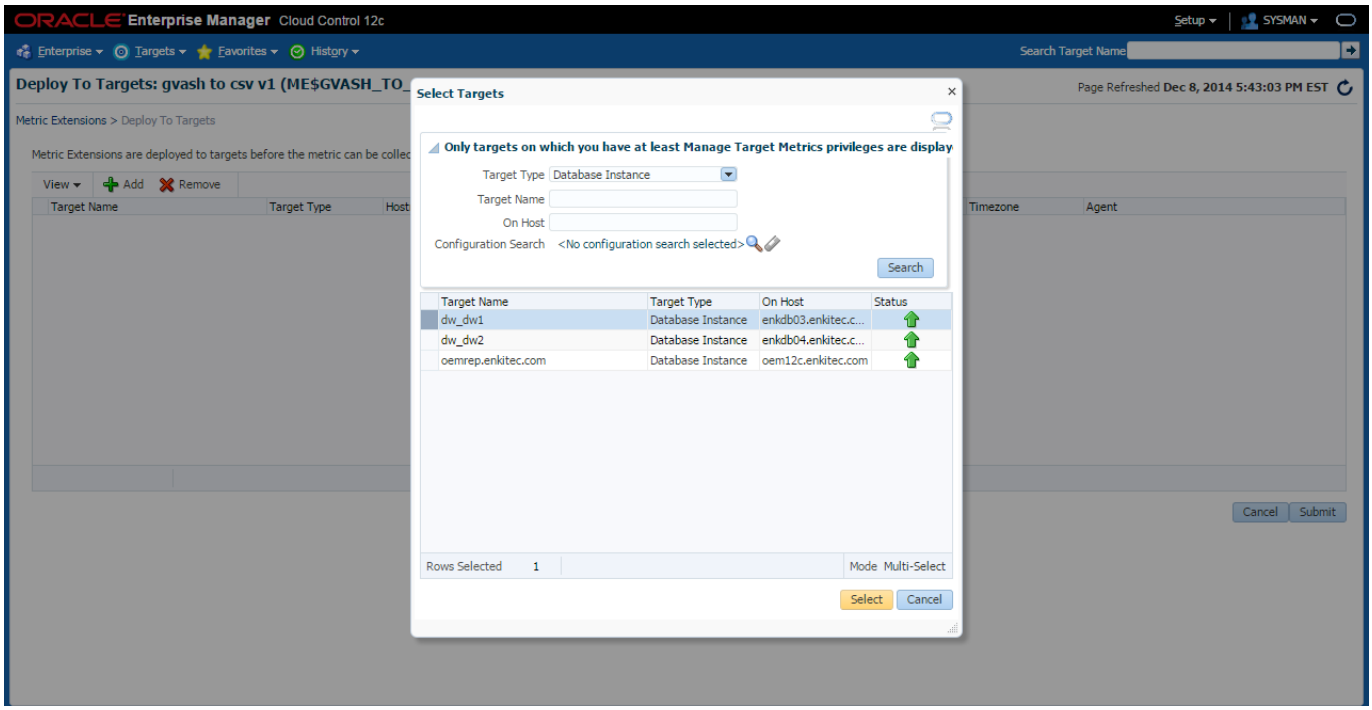
1) On the newly created Metric Extension -> Actions -> Save As Deployable Draft
   - Newly created Metric Extension has a status of "Editable". It should be on "Deployable Draft" so you can assign targets to it. And then status of "Published" to start collecting data.



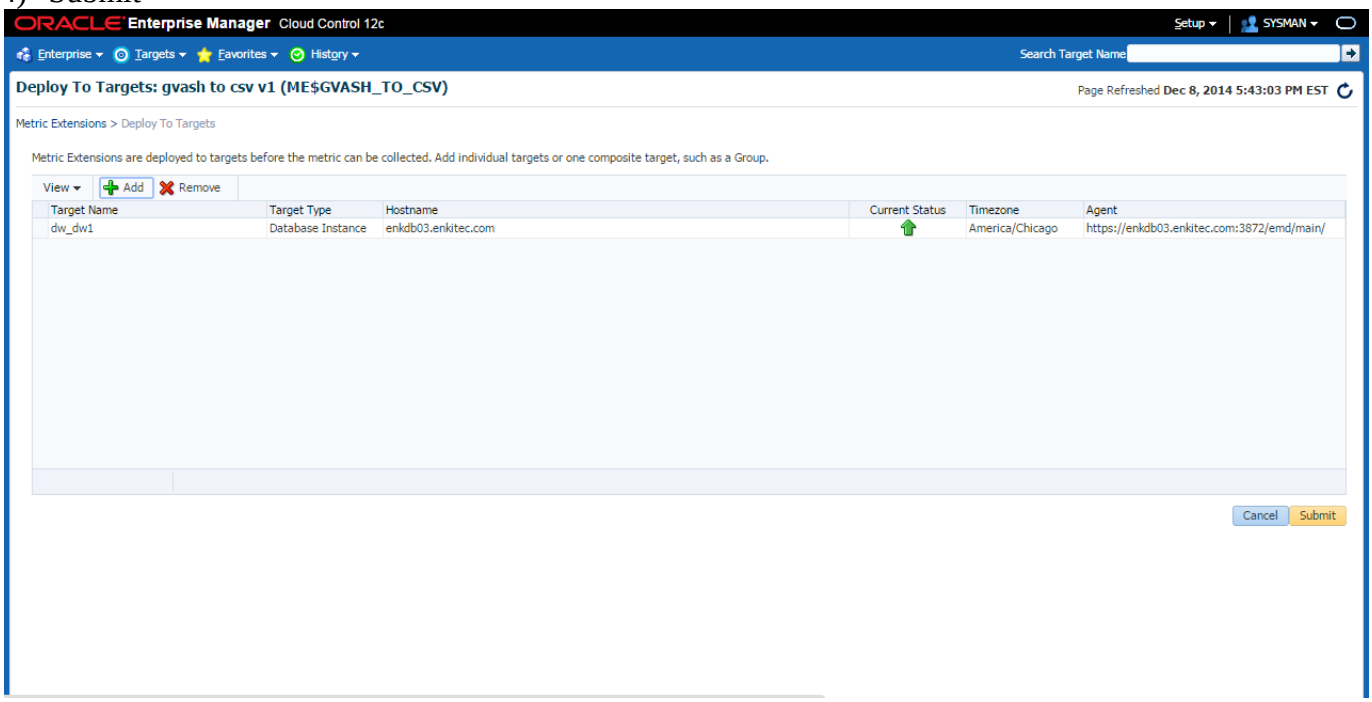
2) Actions -> Deploy To Targets



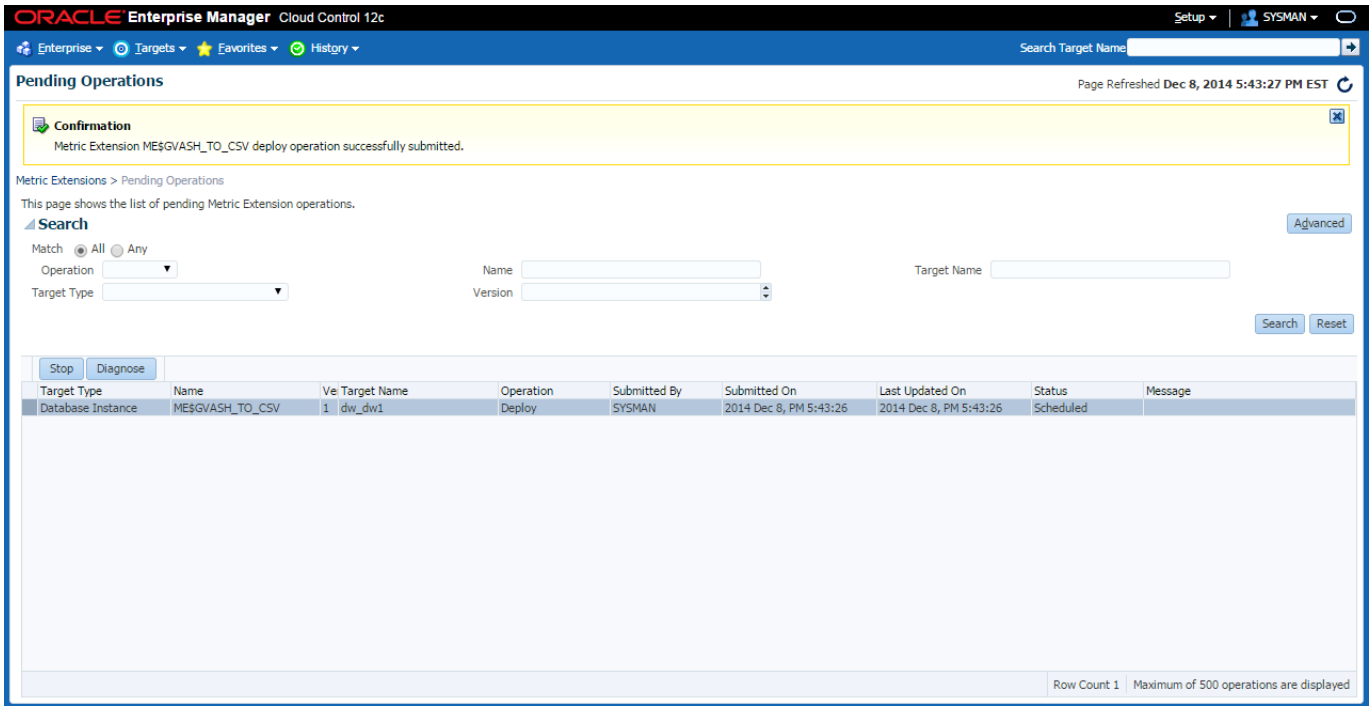
3) Select the 1st instance of any cluster database



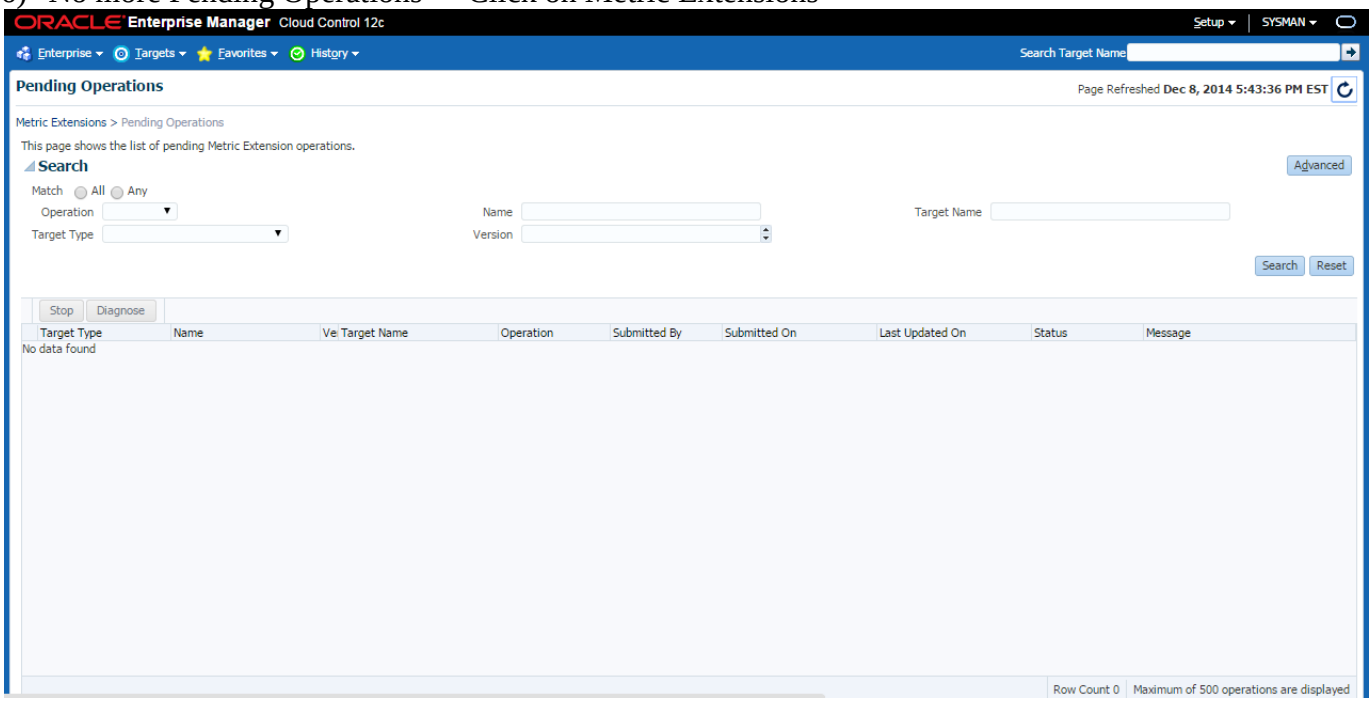
4) Submit



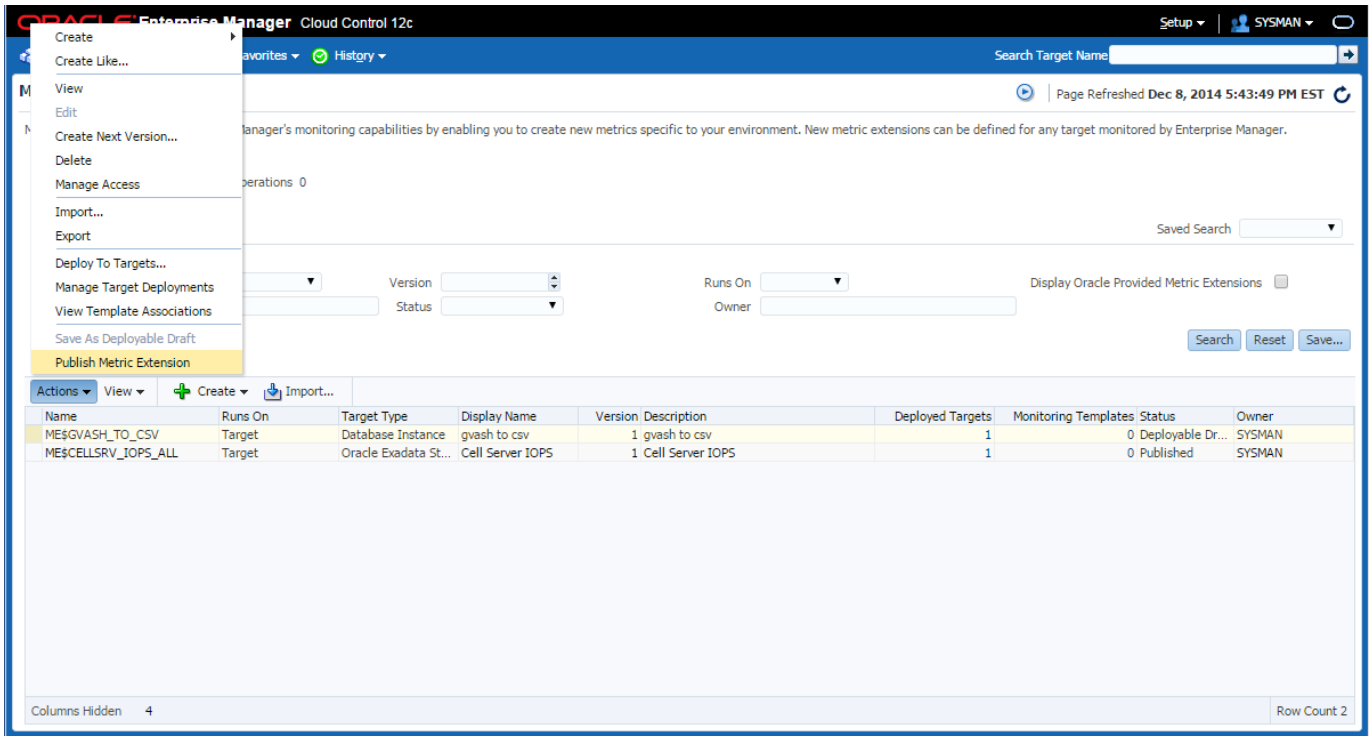
5) Hit “Refresh” (top right)



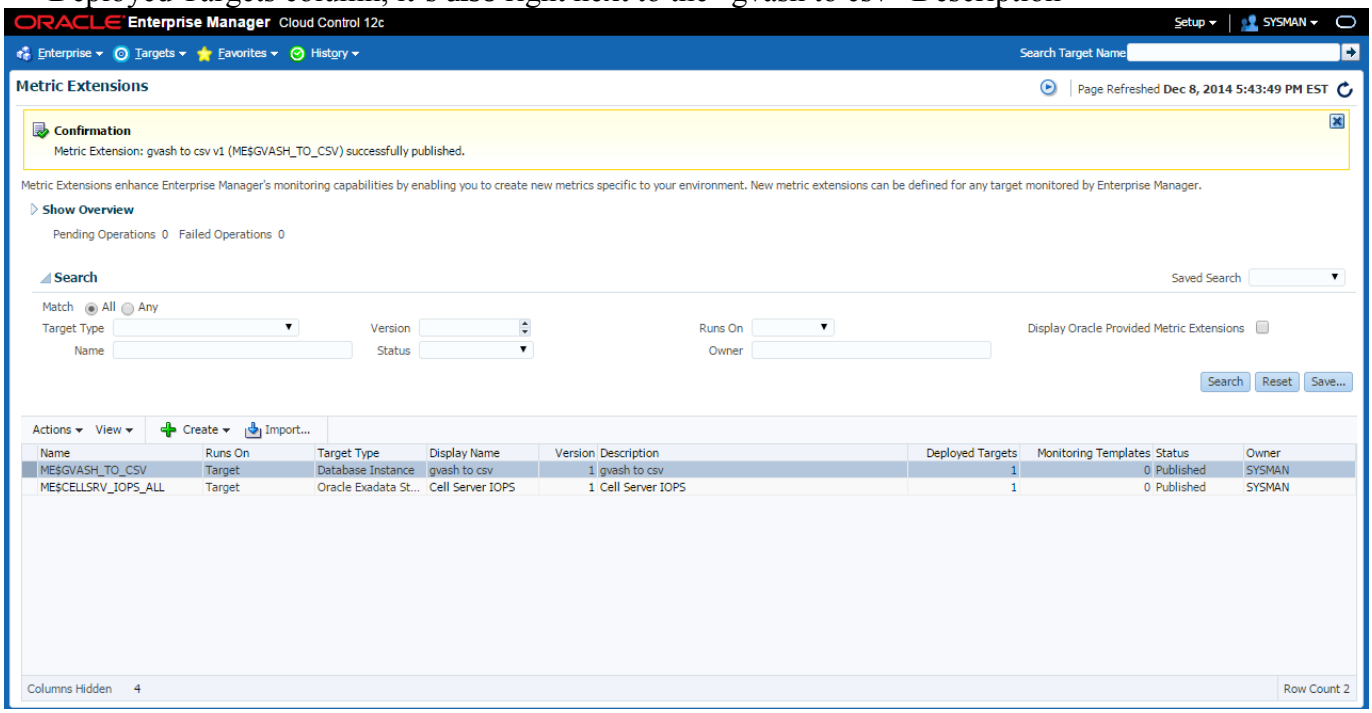
6) No more Pending Operations -> Click on Metric Extensions



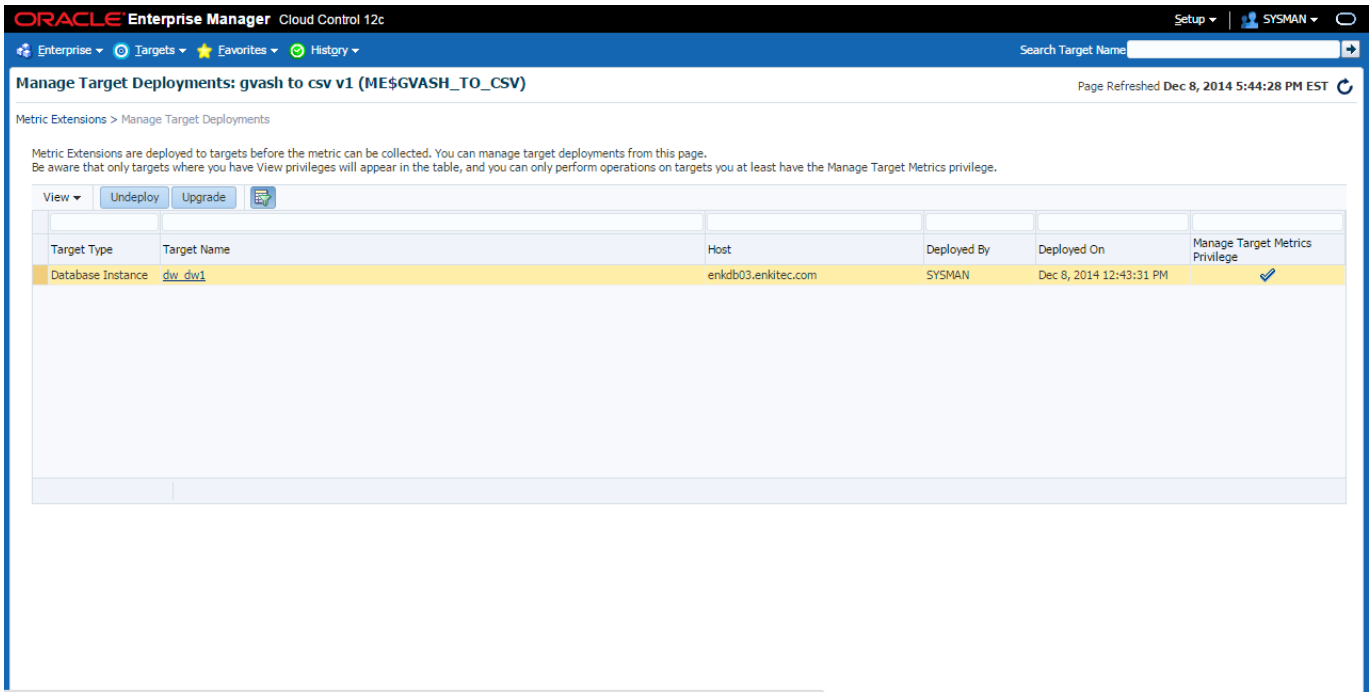
7) The Deployed Targets is now 1 -> go to Actions -> Publish Metric Extension (this will start the data collection)



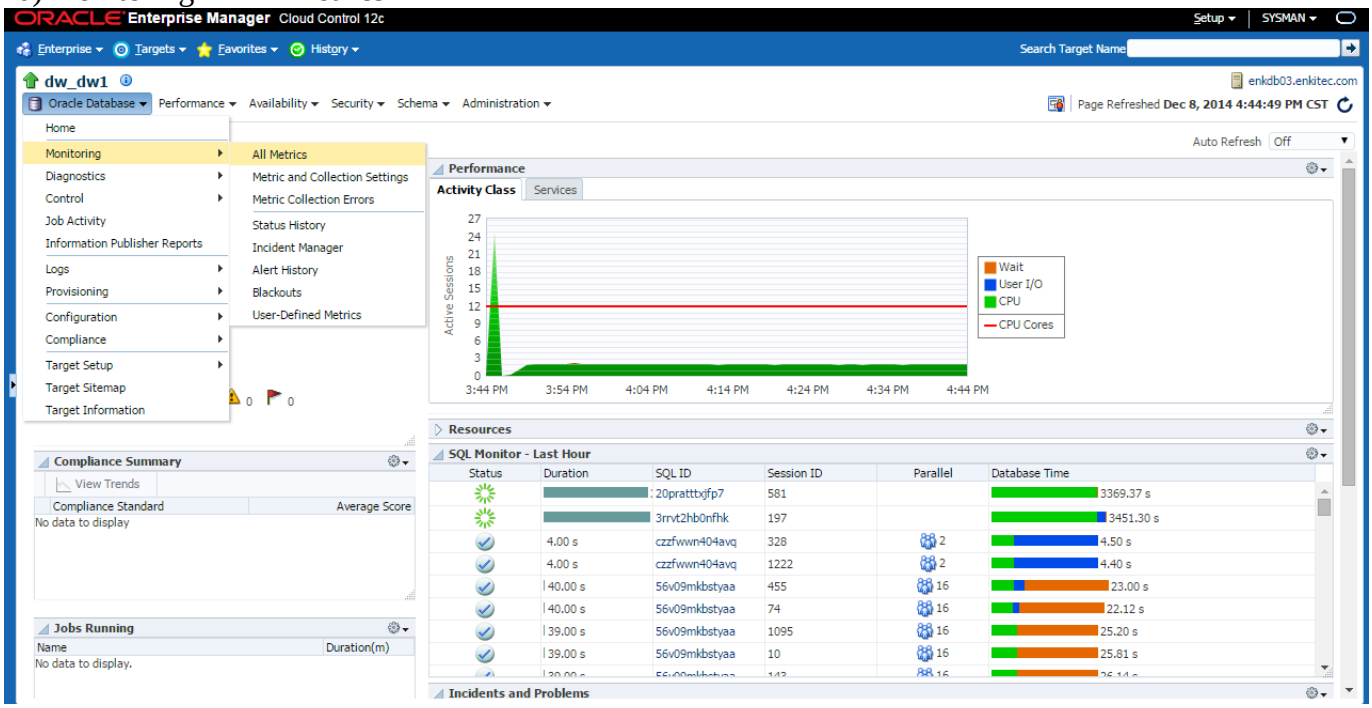
8) To check the data collection of the newly created Metric Extension -> Click on the number (1) on the Deployed Targets column, it's also right next to the "gvash to csv" Description



9) Click on the Target Name

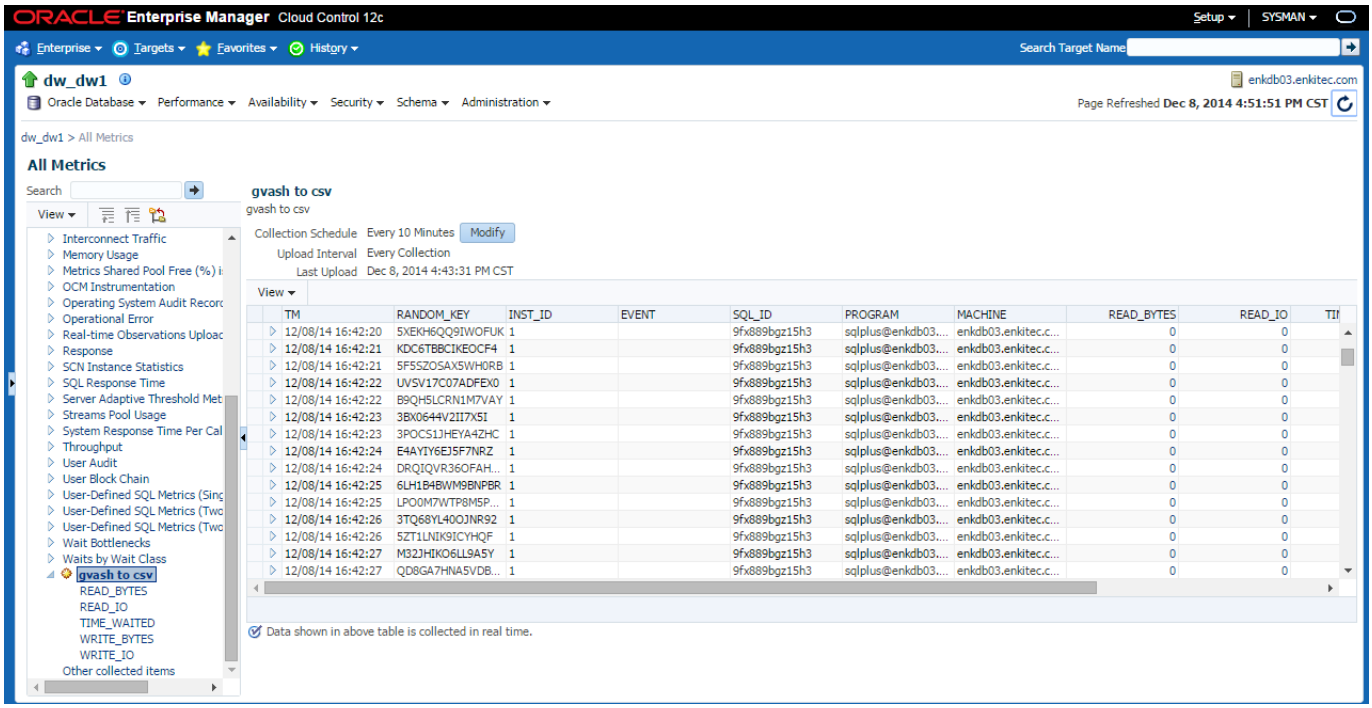


10) Monitoring -> All Metrics



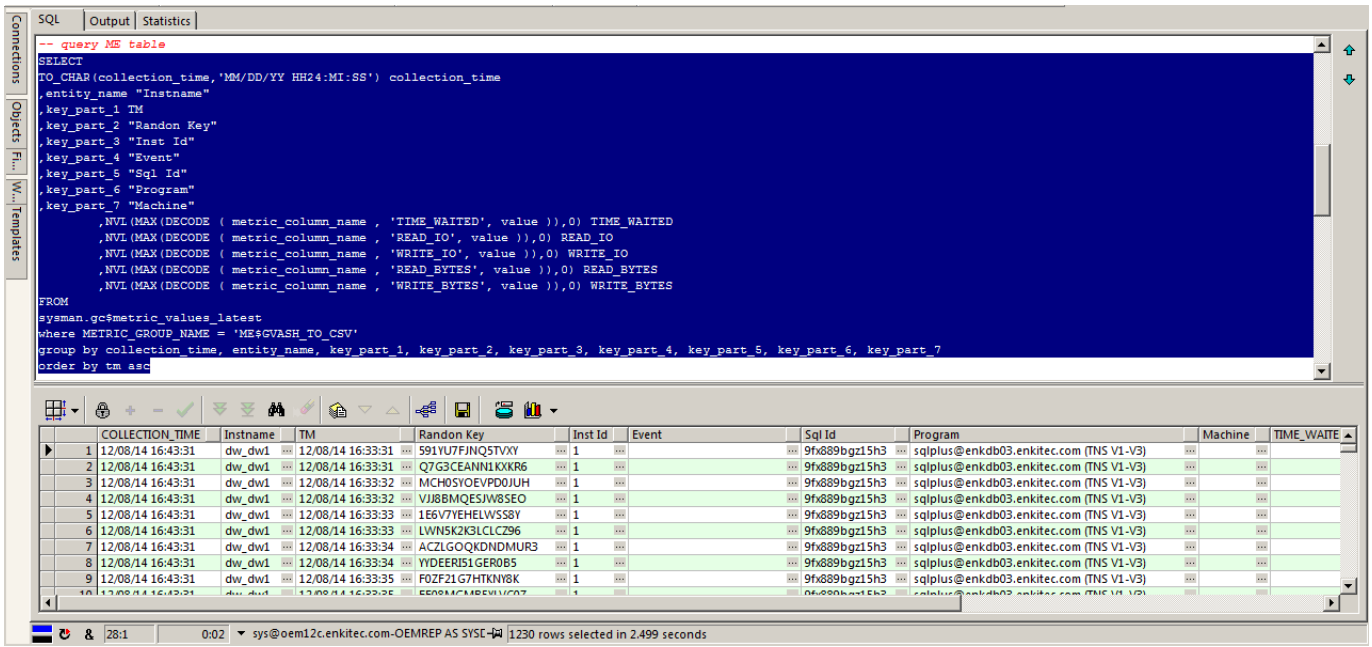
11) Scroll down -> click on “gvash to csv”

- It should show a timestamp of “Last Upload”, else wait for a couple of minutes. Once you see the timestamp you can check the data on the OEM12c repository through SQL*Plus.



12) Check the latest collection by querying the SYSMAN.GC$METRIC_VALUES views

- The view "sysman.gc$metric_values_latest" shows the most recent collections
- The view "sysman.gc$metric_values_hourly" shows the hourly history view of the Metric Extension data



Check the next section below "Appendix A: Extracting the Metric Extension data" section on how to extract the data

Appendix A: Extracting the Metric Extension data

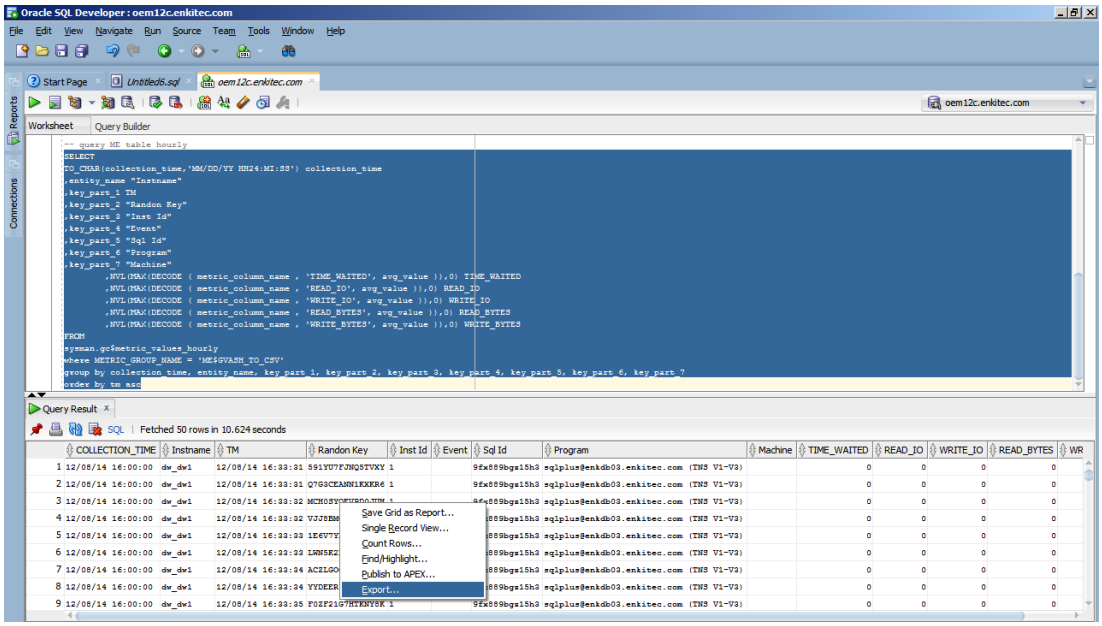
- "sysman.gc$metric_values_latest" shows the most recent collections

```
-- query ME table latest
SELECT
TO_CHAR(collection_time,'MM/DD/YY HH24:MI:SS') collection_time
,entity_name "Instname"
,key_part_1 TM
,key part 2 "Randon Key"
,key_part_3 "Inst Id"
,key_part_4 "Event"
,key_part_5 "Sql Id"
,key_part_6 "Program"
,key_part_7 "Machine"
        ,NVL(MAX(DECODE ( metric column name , 'TIME WAITED', value )),0) TIME WAITED
        ,NVL(MAX(DECODE ( metric column name , 'READ IO', value )),0) READ IO
        ,NVL(MAX(DECODE ( metric column name , 'WRITE IO', value )),0) WRITE IO
        ,NVL(MAX(DECODE ( metric_column_name , 'READ_BYTES', value )),0) READ_BYTES
        ,NVL(MAX(DECODE ( metric_column_name , 'WRITE_BYTES', value )),0) WRITE_BYTES
FROM
sysman.gc$metric values latest
where METRIC GROUP NAME = 'ME$GVASH TO CSV'
group by collection_time, entity_name, key_part_1, key_part_2, key_part_3, key_part_4, key_part_5, key_part_6, key_part_7
order by tm asc
```

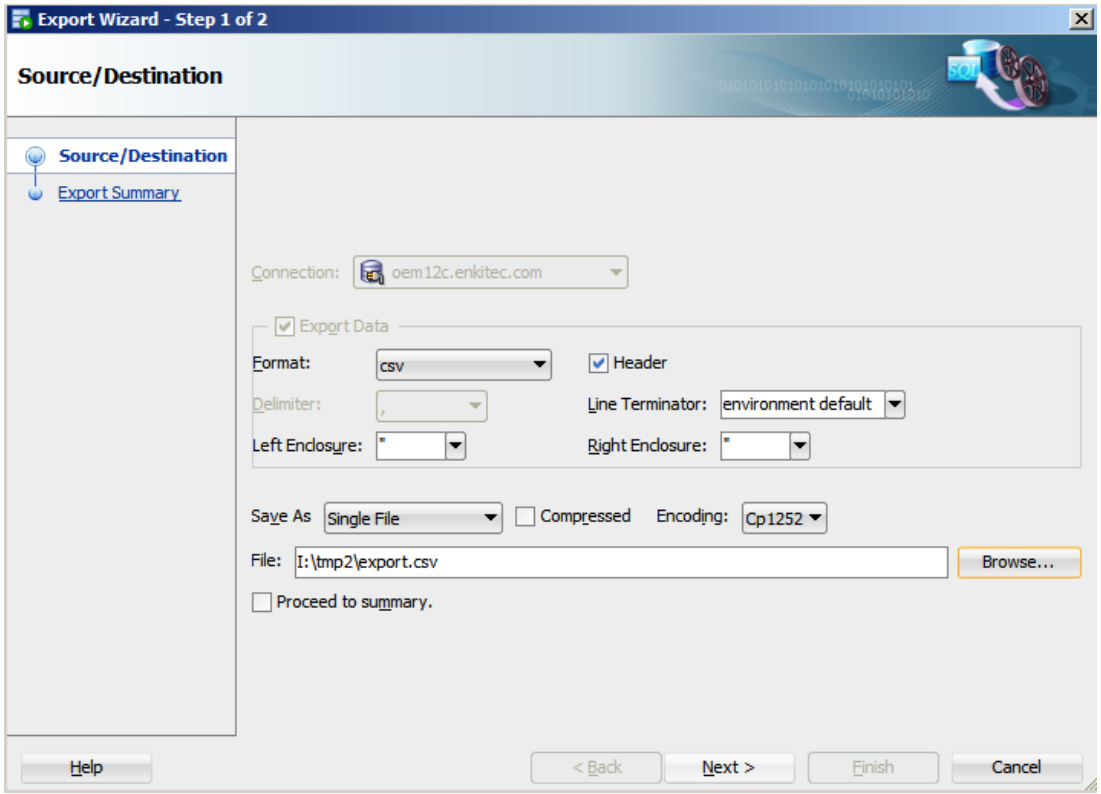
- "sysman.gc$metric_values_hourly" shows the hourly history view of the Metric Extension data

```
-- query ME table hourly
SELECT
TO_CHAR(collection_time,'MM/DD/YY HH24:MI:SS') collection_time
,entity_name "Instname"
,key_part_1 TM
,key_part_2 "Randon Key"
,key part 3 "Inst Id"
,key part 4 "Event"
,key part 5 "Sql Id"
,key_part_6 "Program"
,key_part_7 "Machine"
        ,NVL(MAX(DECODE ( metric_column_name , 'TIME_WAITED', avg_value )),0) TIME_WAITED
        ,NVL(MAX(DECODE ( metric_column_name , 'READ_IO', avg_value )),0) READ_IO
        ,NVL(MAX(DECODE ( metric_column_name , 'WRITE_IO', avg_value )),0) WRITE_IO
        ,NVL(MAX(DECODE ( metric_column_name , 'READ_BYTES', avg_value )),0) READ_BYTES
        ,NVL(MAX(DECODE ( metric_column_name , 'WRITE_BYTES', avg_value )),0) WRITE_BYTES
FROM
sysman.gc$metric_values_hourly
where METRIC_GROUP_NAME = 'ME$GVASH_TO_CSV'
group by collection time, entity name, key part 1, key part 2, key part 3, key part 4, key part 5, key part 6, key part 7
order by tm asc
```

1) On SQL*Developer copy the "sysman.gc$metric_values_hourly" SQL -> hit the "Run Statement" -> right click on the grid -> Export



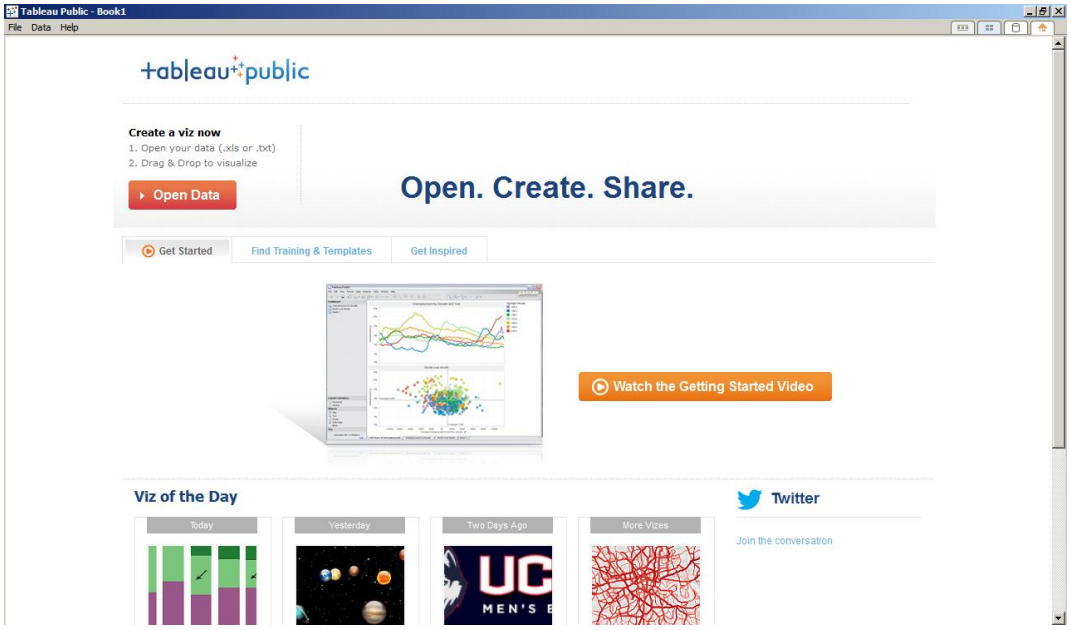
2) Select format as CSV -> Next -> Finish



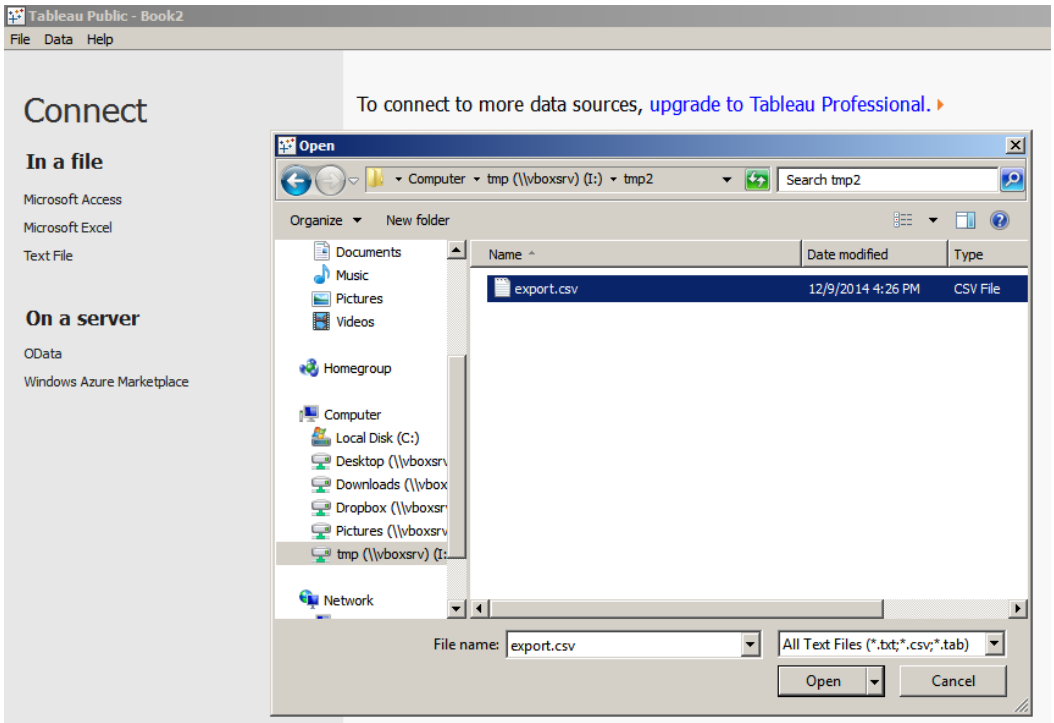
Appendix B: Graphing the Metric Extension data

1) Download the tableau public at http://www.tableausoftware.com/public/

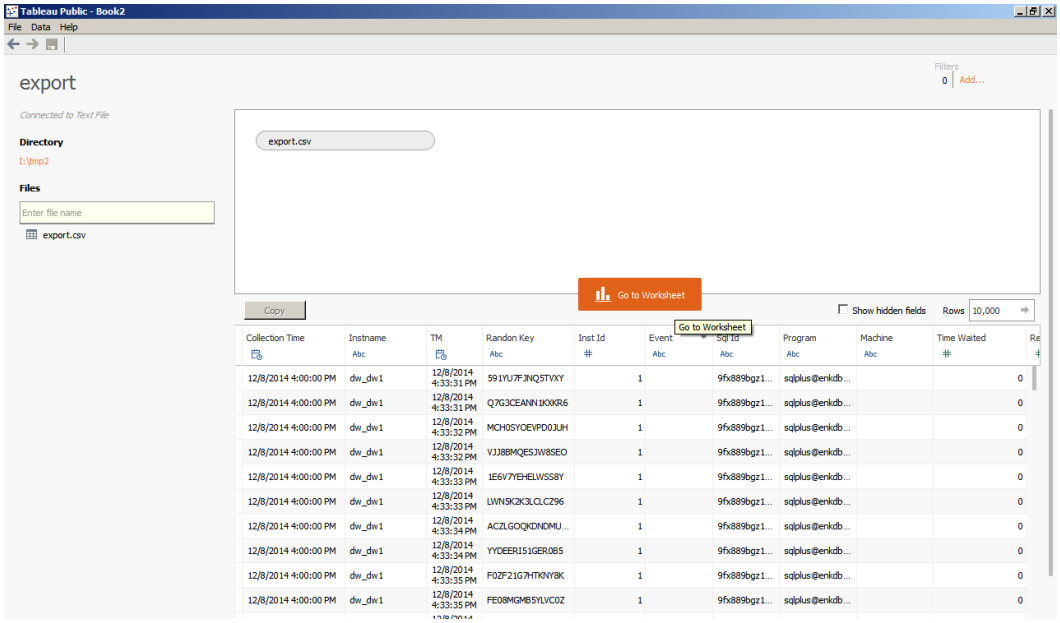
2) Click Open Data



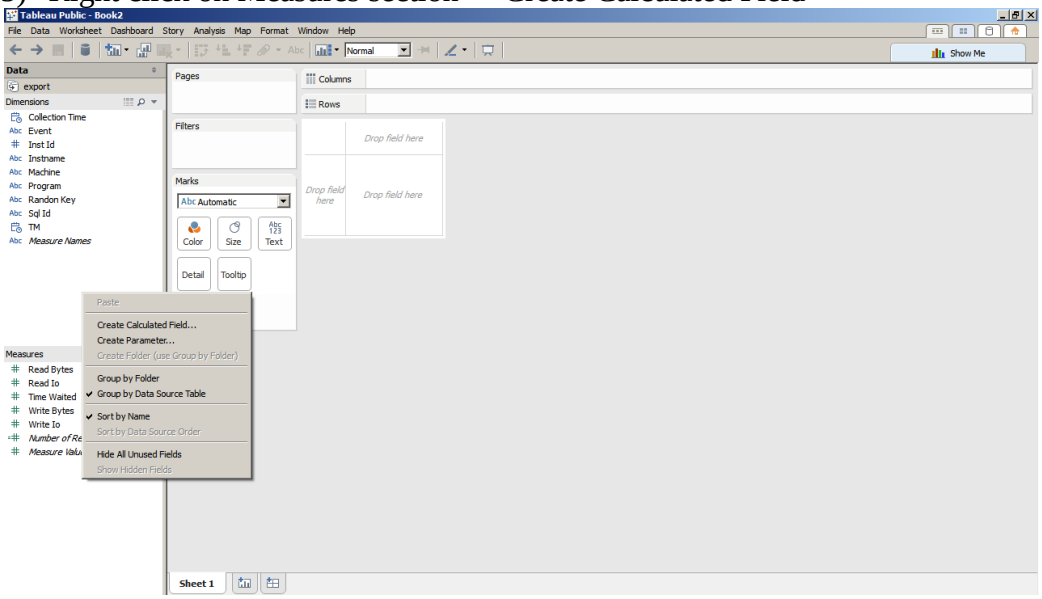
3) Click Text File -> Select the file



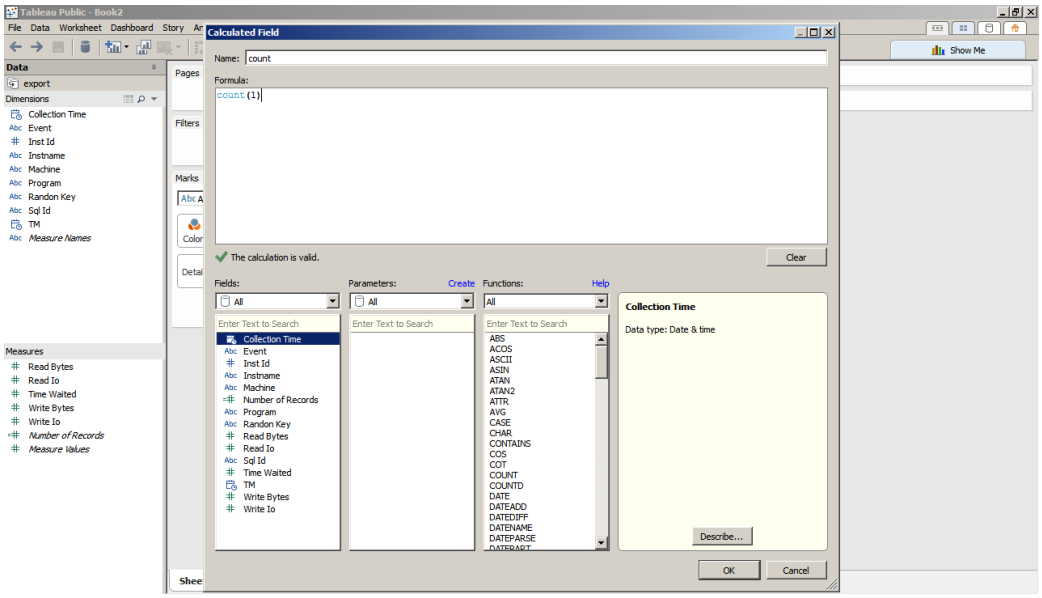
4) Click Go To Worksheet



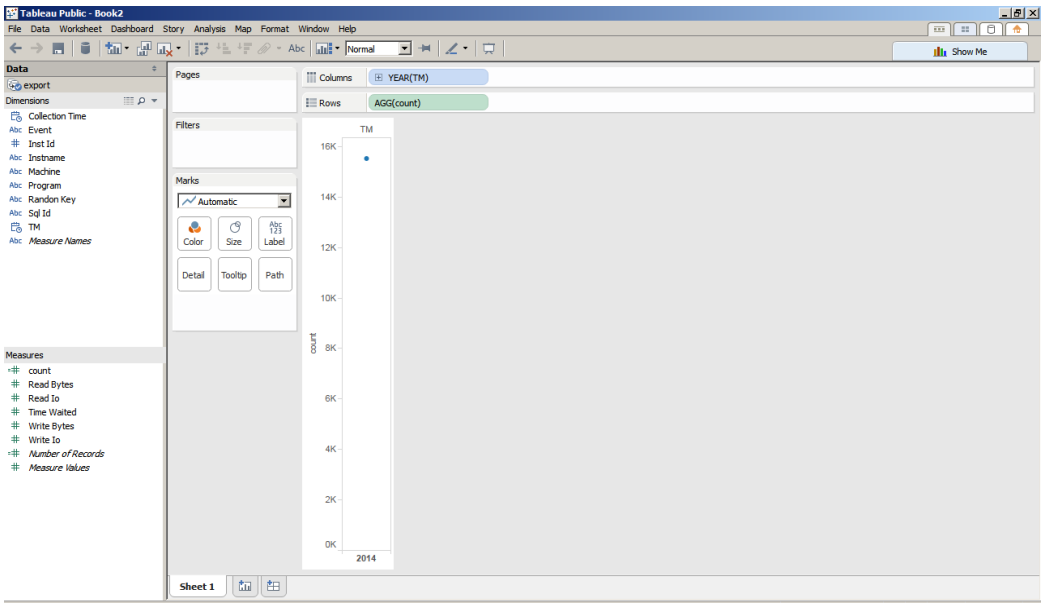
5) Right click on Measures section -> Create Calculated Field



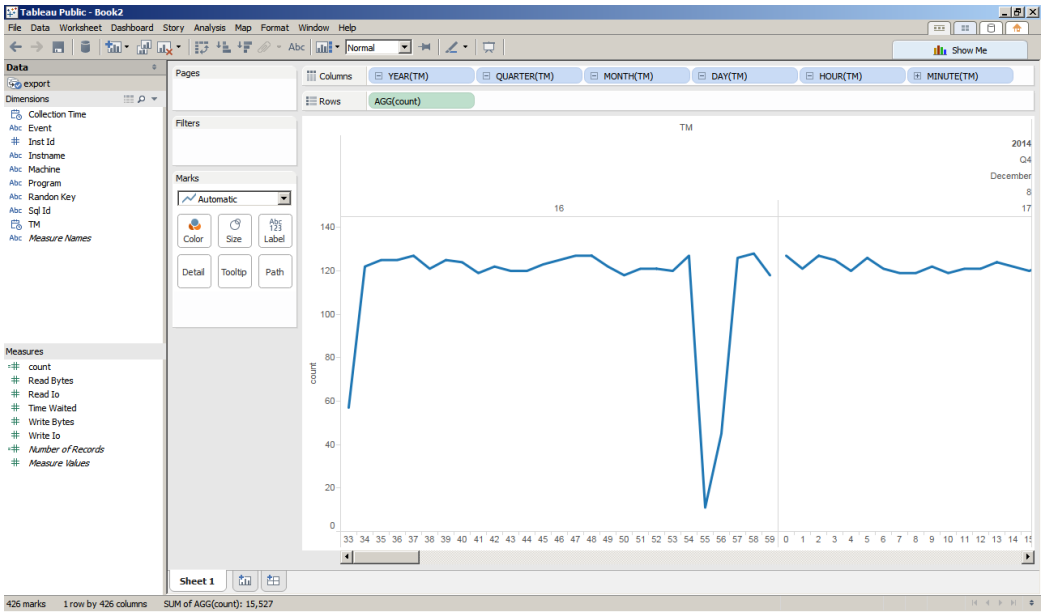
6) Create the “count” calculated field -> OK
   a. Name: count
   b. Formula: count(1)



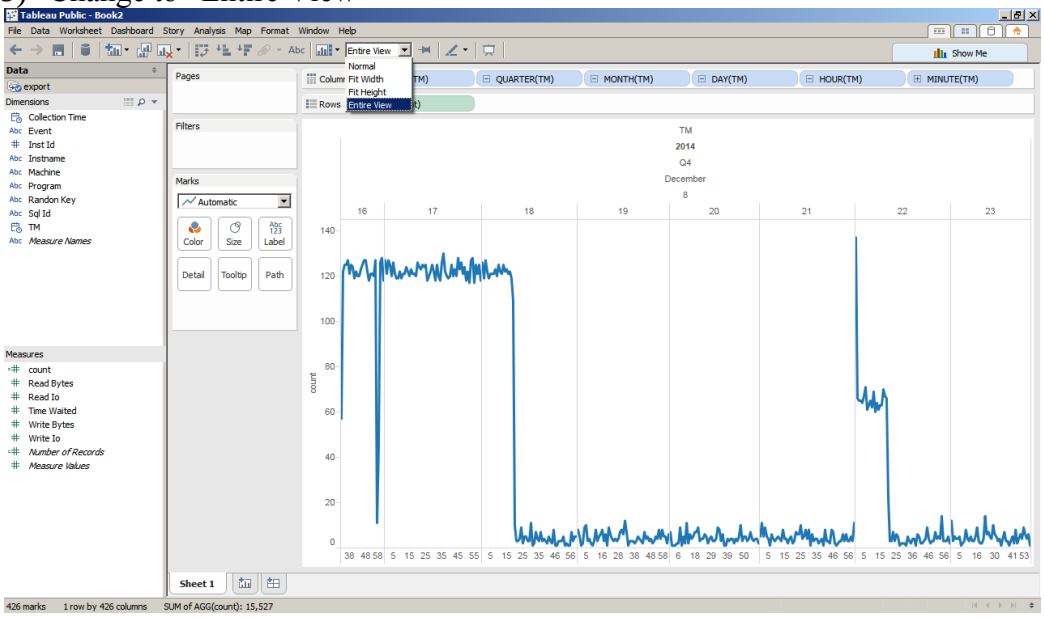
7) Drag the “TM” on the Dimensions section to the Columns. And then drag the “count” on the Measures section to the Rows



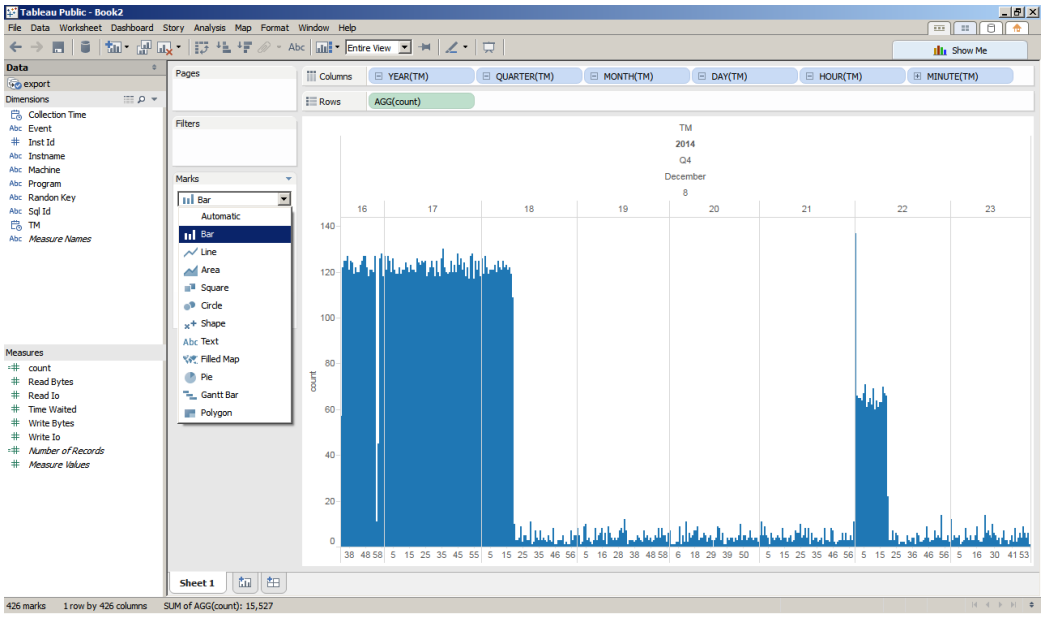
8) Click on the plus sign of the “Year(TM)” up to the “Minute(TM)” dimension



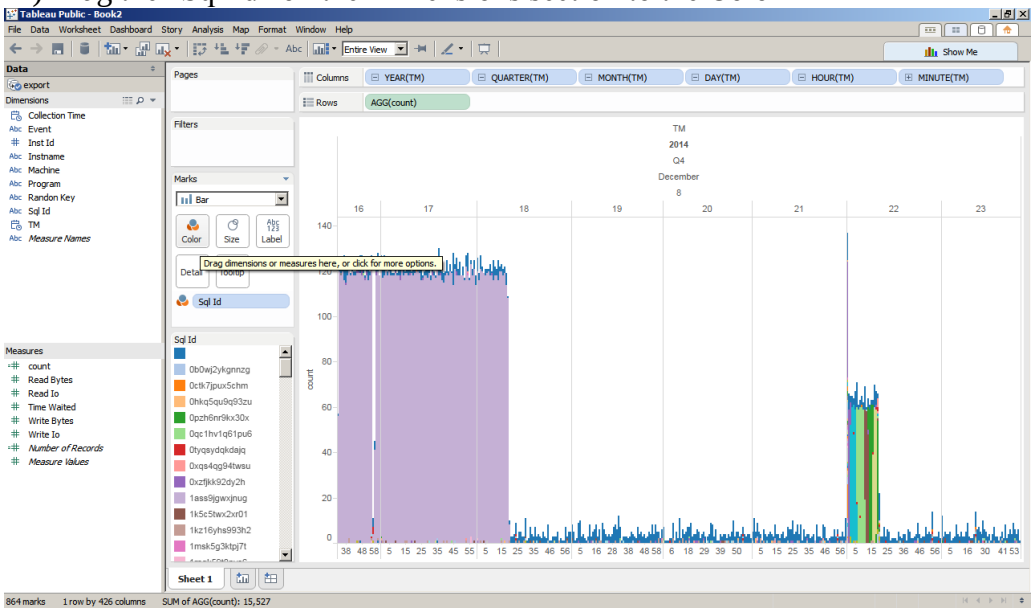
9) Change to “Entire View”



10) Change to “Bar” graph

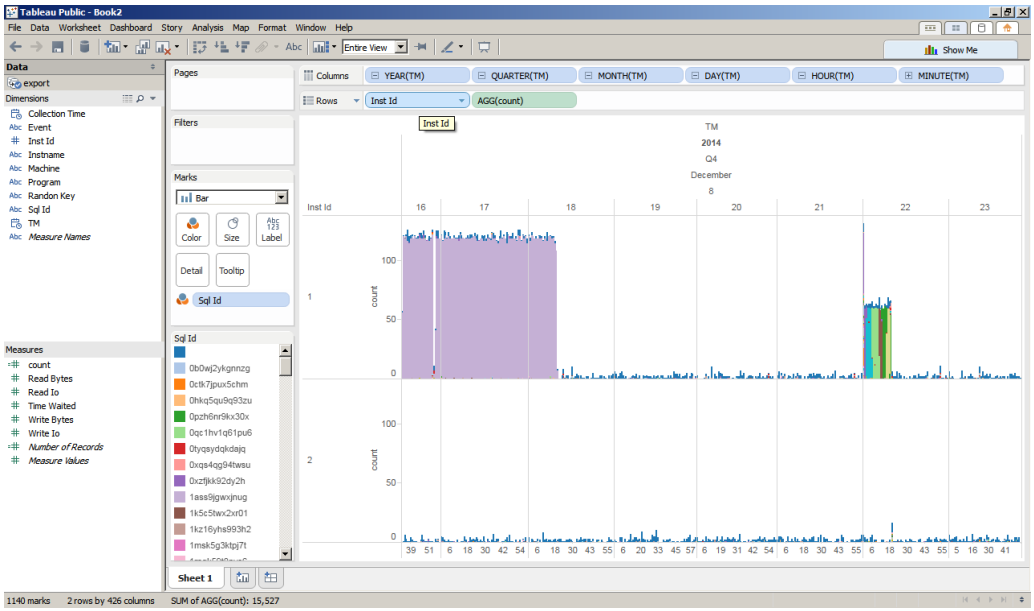


11) Drag the “Sql Id” on the Dimensions section to the Color



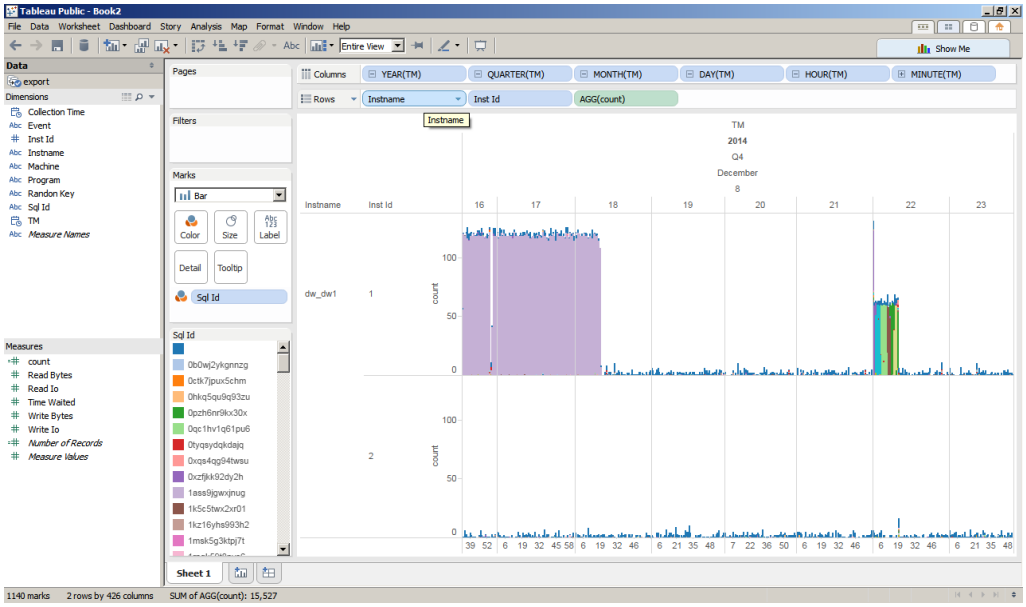
12) Drag the “Inst Id” on the Dimensions section to the Rows

- This will show the distribution of load across instance 1 and 2

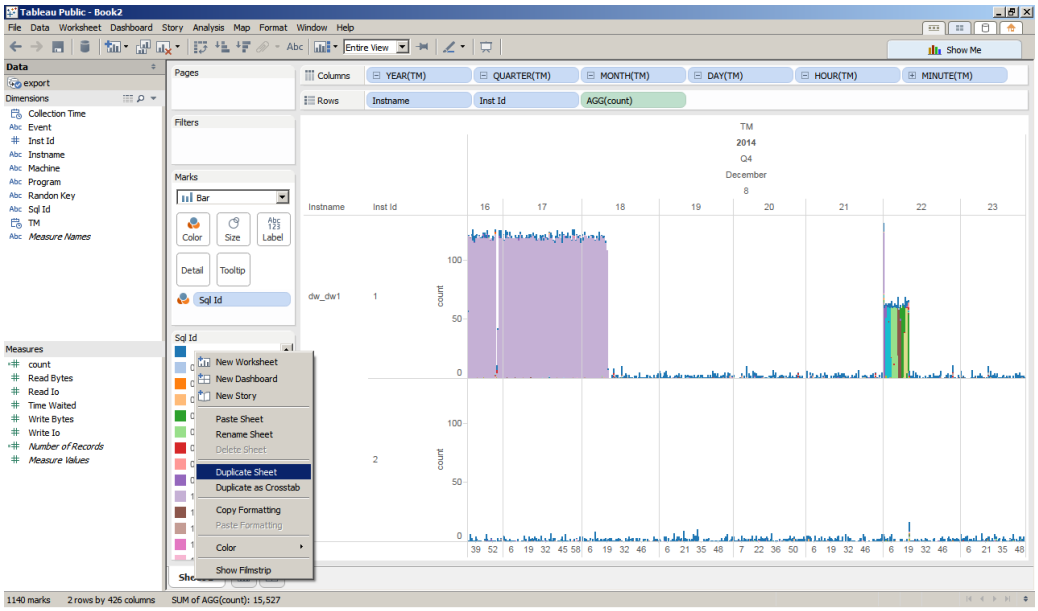


13) Drag the “Instname” on the Dimensions section to the Rows

- This will show the distribution of the databases across instance 1 and 2

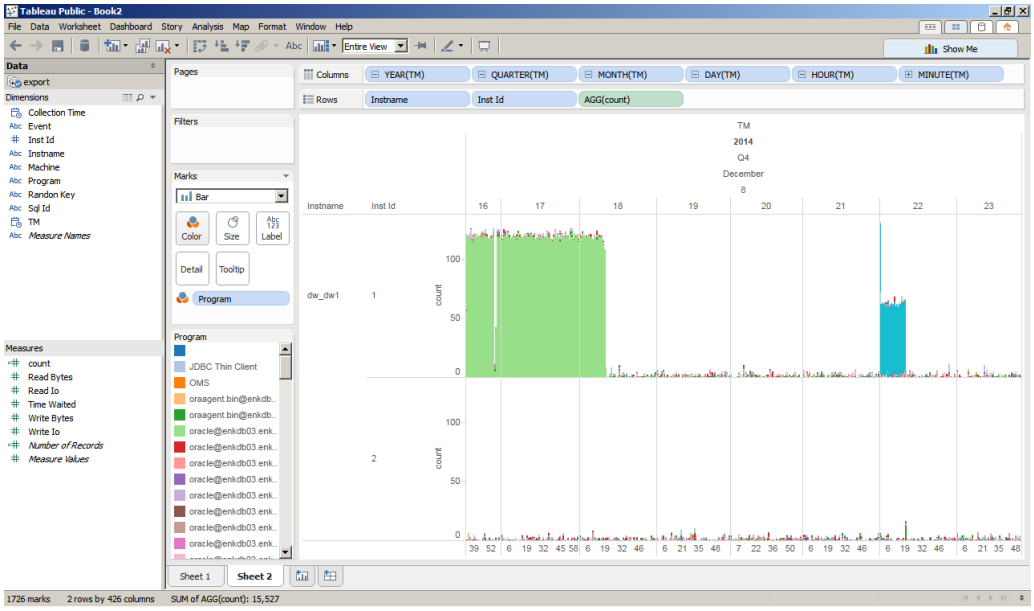


14) Duplicate the current sheet by right clicking on the tab -> Duplicate Sheet



15) Drag the “Program” on the Dimensions section to the Rows

- This will show the distribution of the programs across instance 1 and 2



16) Drag the “Event” on the Dimensions section to the Rows

- This will show the distribution of the CPU and wait events across instance 1 and 2
- The “null” event pertains to CPU usage (shown below)

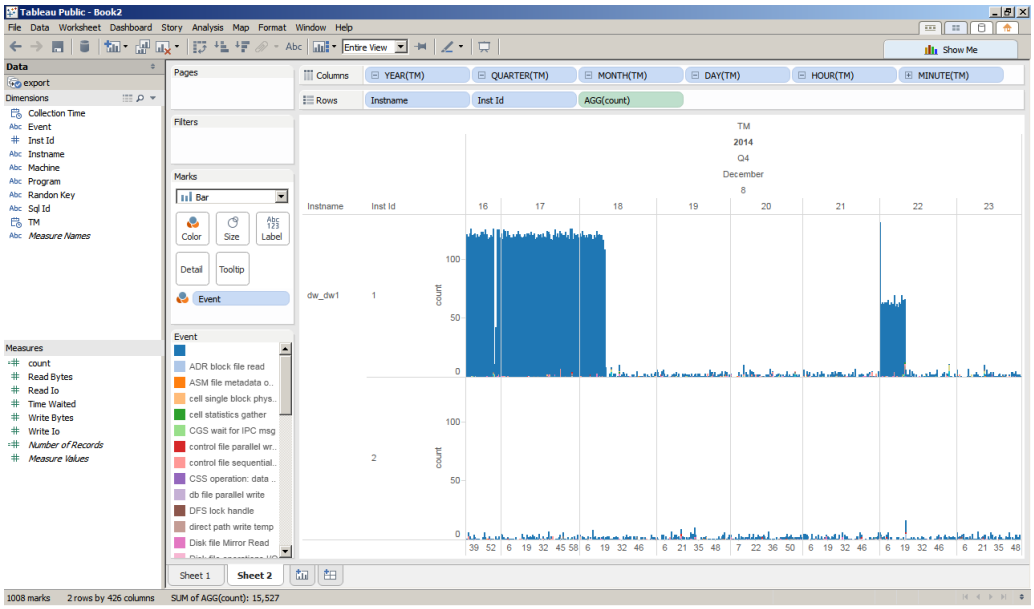
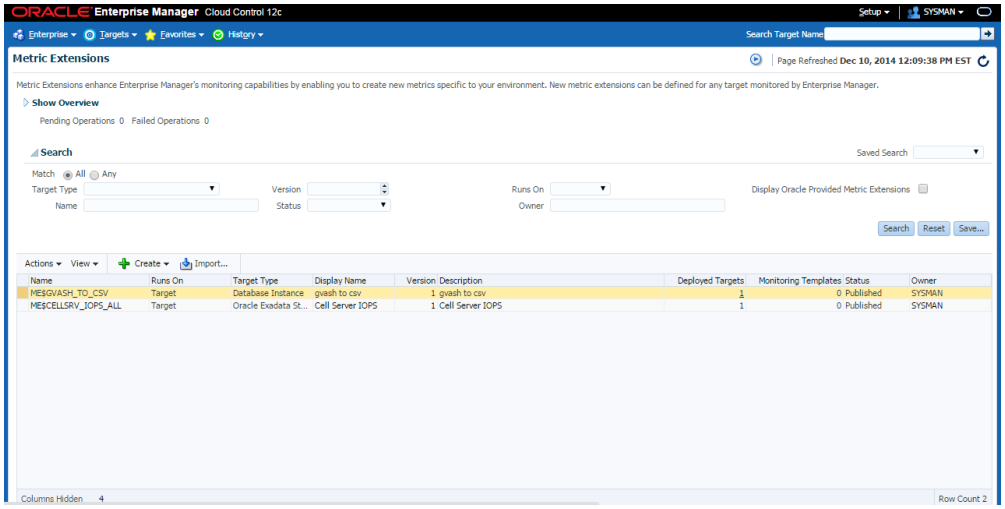


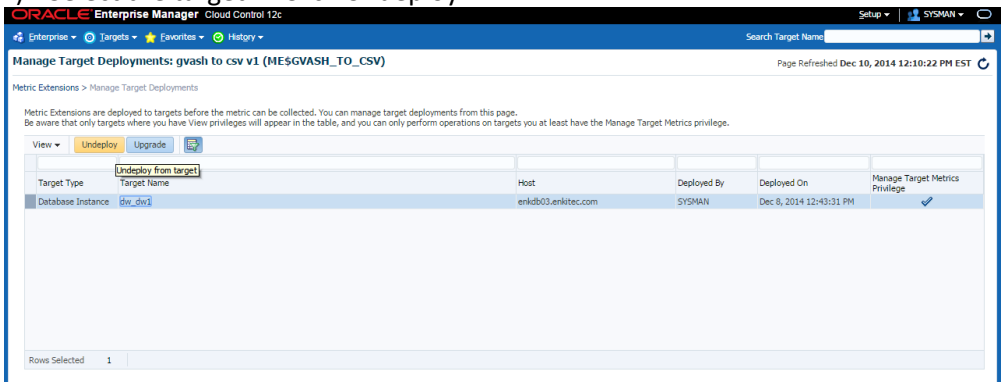
Tableau Public - Book2
File Data Worksheet Dashboard Story Analysis Map Format Window Help
Show Me
Data
export
Dimensions
Collection Time
Event
Inst Id
Instname
Machine
Program
Randon Key
Sql Id
TM
Measure Names
Measures
count
Read Bytes
Read Io
Time Waited
Write Bytes
Write Io
Number of Records
Measure Values
Pages
Filters
Marks
Bar
Color
Size
Label
Detail
Tooltip
Event
ADR block file read
ASM file metadata o..
cell single block phys..
cell statistics gather
CGS wait for IPC msg
control file parallel wr..
control file sequential..
CSS operation: data ..
db file parallel write
DFS lock handle
direct path write temp
Disk file Mirror Read
Columns
YEAR(TM)
QUARTER(TM)
MONTH(TM)
DAY(TM)
HOUR(TM)
MINUTE(TM)
Rows
Instname
Inst Id
AGG(count)
TM
2014
Q4
December
8
Instname
Inst Id
16
17
18
19
20
21
22
23
dw_dw1
1
2
count
100
50
0
39 52 6 19 32 45 58 6 19 32 46 6 21 35 48 7 22 36 50 6 19 32 46 6 19 32 46 6 21 35 48
Sheet 1
Sheet 2
1008 marks
2 rows by 426 columns
SUM of AGG(count): 15,527

Appendix C: Uninstall Metric Extension

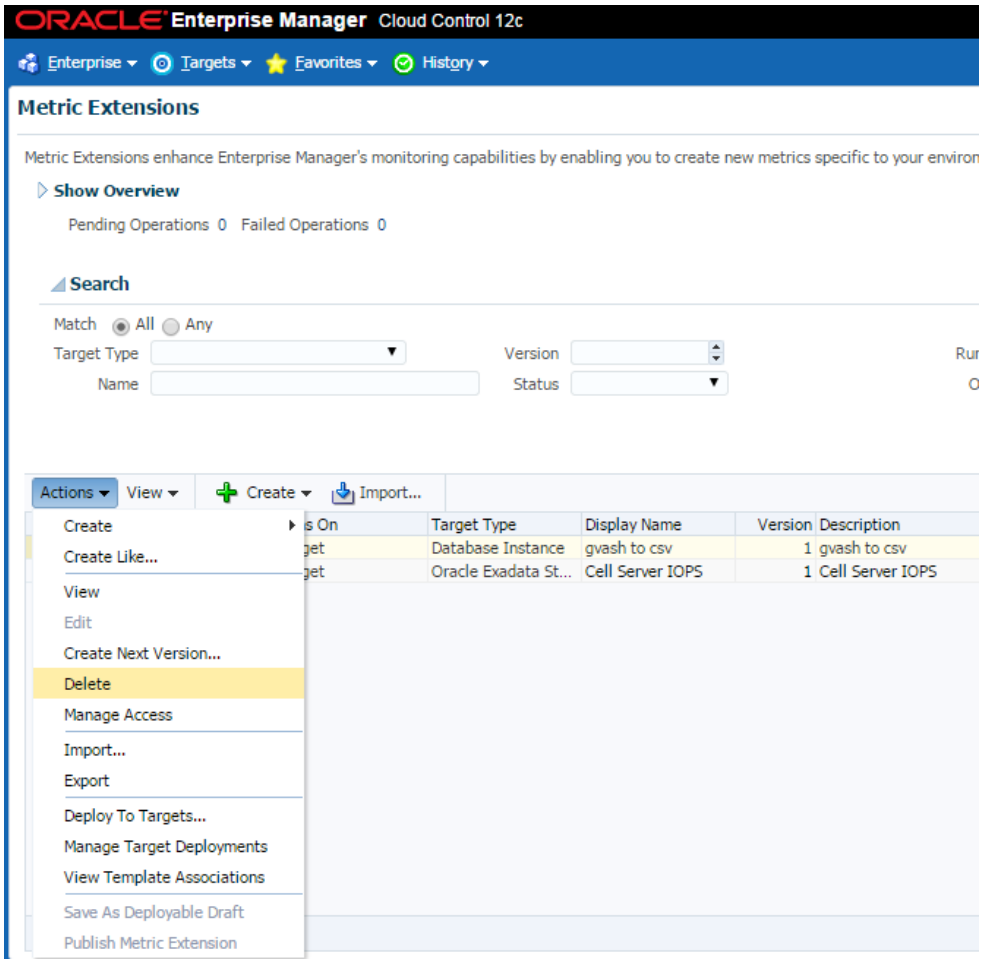
1) To stop the data collection of the Metric Extension -> Click on the number (1) on the Deployed Targets column, it's also right next to the "gvash to csv" Description



2) Select the target -> Click Undeploy



3) To delete/uninstall the Metric Extension -> Actions -> Delete


ORACLE Enterprise Manager Cloud Control 12c
Enterprise
Targets
Favorites
History
Metric Extensions
Metric Extensions enhance Enterprise Manager's monitoring capabilities by enabling you to create new metrics specific to your environ
Show Overview
Pending Operations 0 Failed Operations 0
Search
Match All Any
Target Type
Version
Name
Status
Actions
View
Create
Import...
Create
Create Like...
View
Edit
Create Next Version...
Delete
Manage Access
Import...
Export
Deploy To Targets...
Manage Target Deployments
View Template Associations
Save As Deployable Draft
Publish Metric Extension
Target Type
Display Name
Version
Description
Database Instance
gvash to csv
1
gvash to csv
Oracle Exadata St...
Cell Server IOPS
1
Cell Server IOPS