**Metric Extension – gv$active\_session\_history to csv**

December 2014

Primary Author: Karl Arao

|  |
| --- |
|  |

Table of Contents

[**I.** **Overview** 3](#_Toc405973422)

[**II.** **Create Metric Extension** 4](#_Toc405973423)

[**III.** **Deploy Metric Extension** 22](#_Toc405973424)

[**Appendix A: Extracting the Metric Extension data** 29](#_Toc405973425)

[**Appendix B: Graphing the Metric Extension data** 31](#_Toc405973426)

[**Appendix C: Uninstall Metric Extension** 39](#_Toc405973427)

|  |
| --- |
|  |

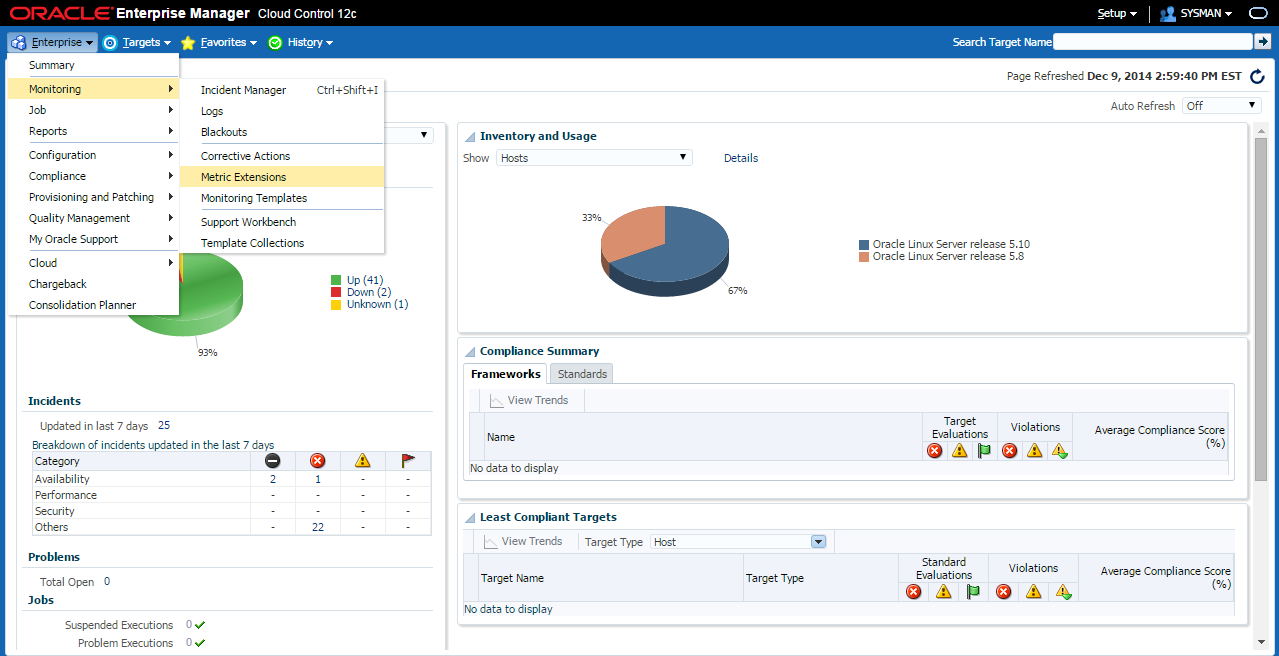
1. **Overview**

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:

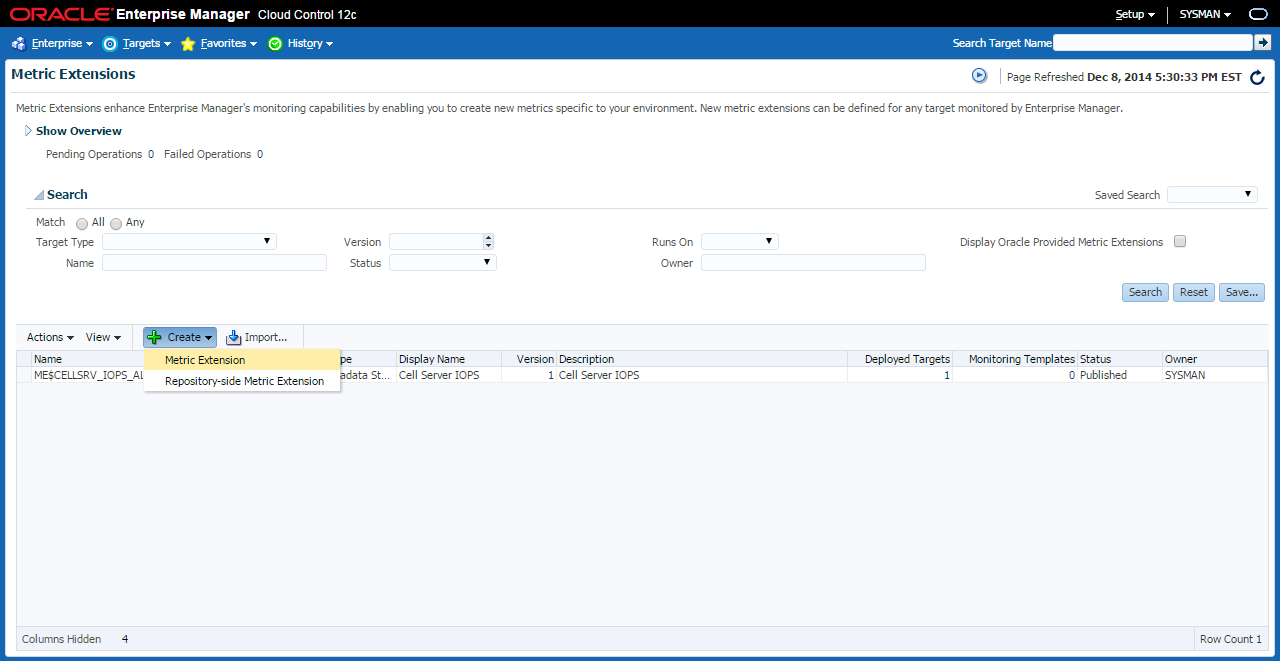
* OEM repository will be responsible of aging out the data (default: 32 days retention)
* No more overhead on the server filesystem
* Easier data access (though SQL\*Developer or SQL\*Plus)
* Easy to scale – deploy the Metric Extension to the new database and performance data will be automatically stored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

|  |
| --- |
|  |

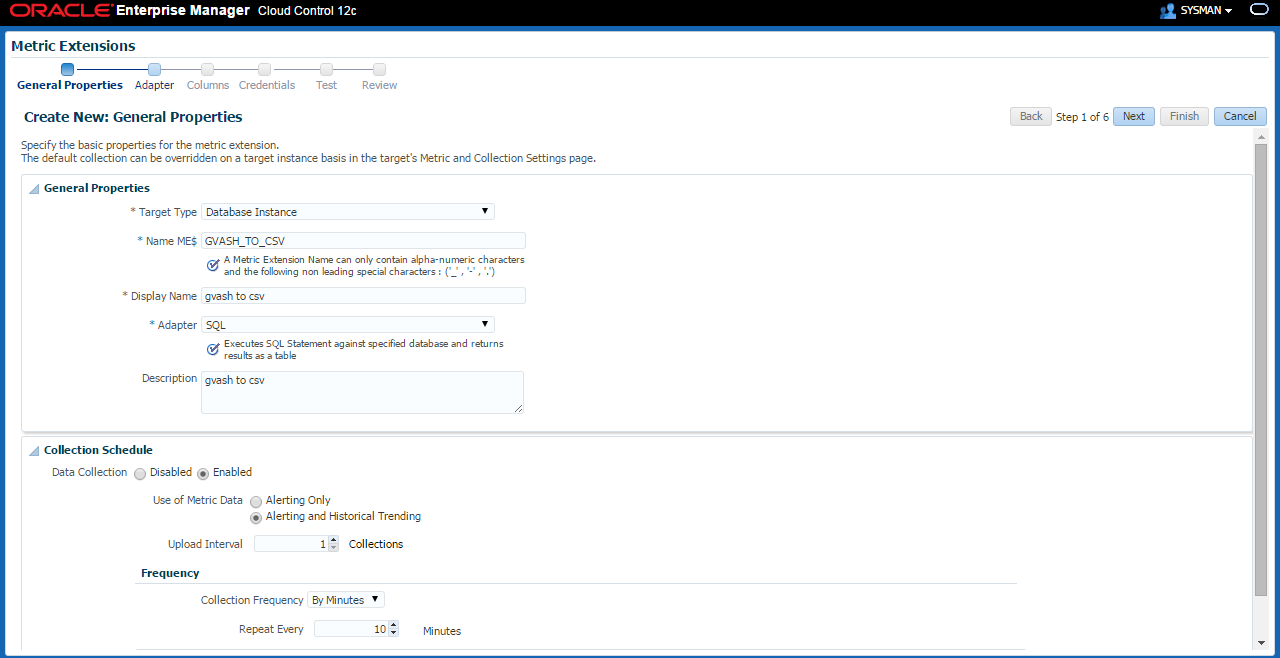
1. **Create Metric Extension**
2. On the OEM12c home page click on Monitoring -> Metric Extensions



1. Create -> Metric Extension

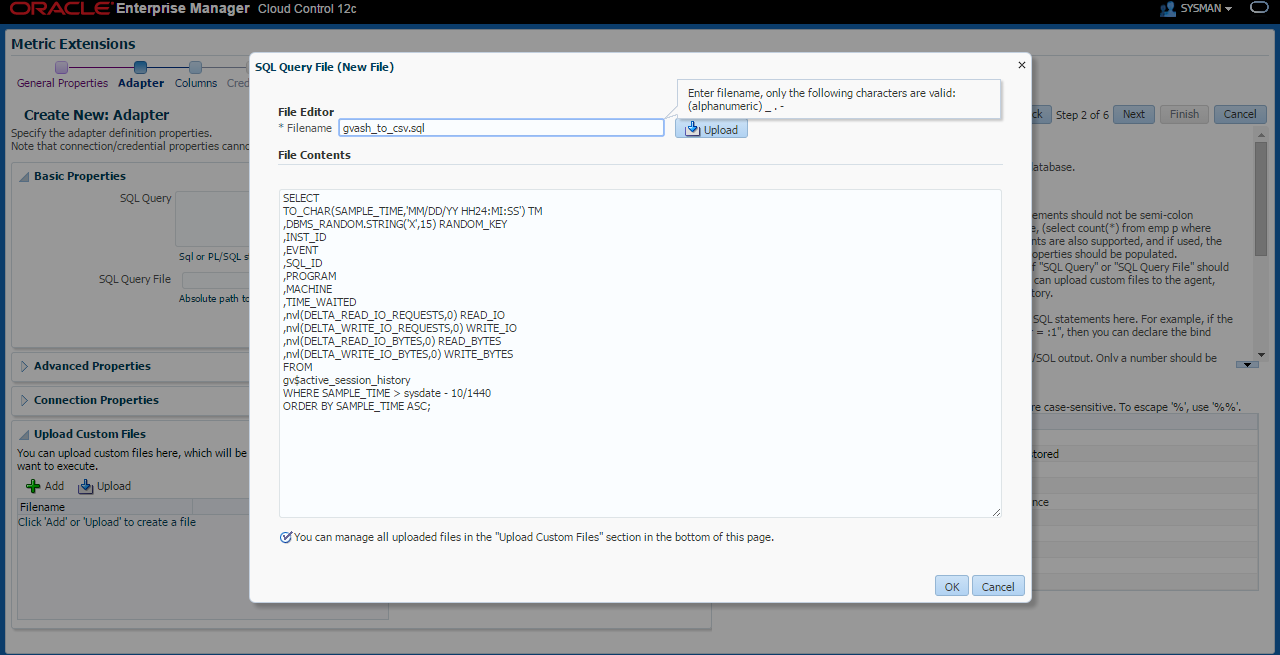


1. On General Properties and Collection Schedule
   1. Target Type: Database Instance
   2. Name ME$: GVASH\_TO\_CSV
   3. Display Name: gvash to csv
   4. Adapter: SQL
   5. Description: gvash to csv
   6. Repeat Every: change to 10 minutes



1. Click on the pencil -> upload the gvash\_to\_csv.sql script (below) -> OK

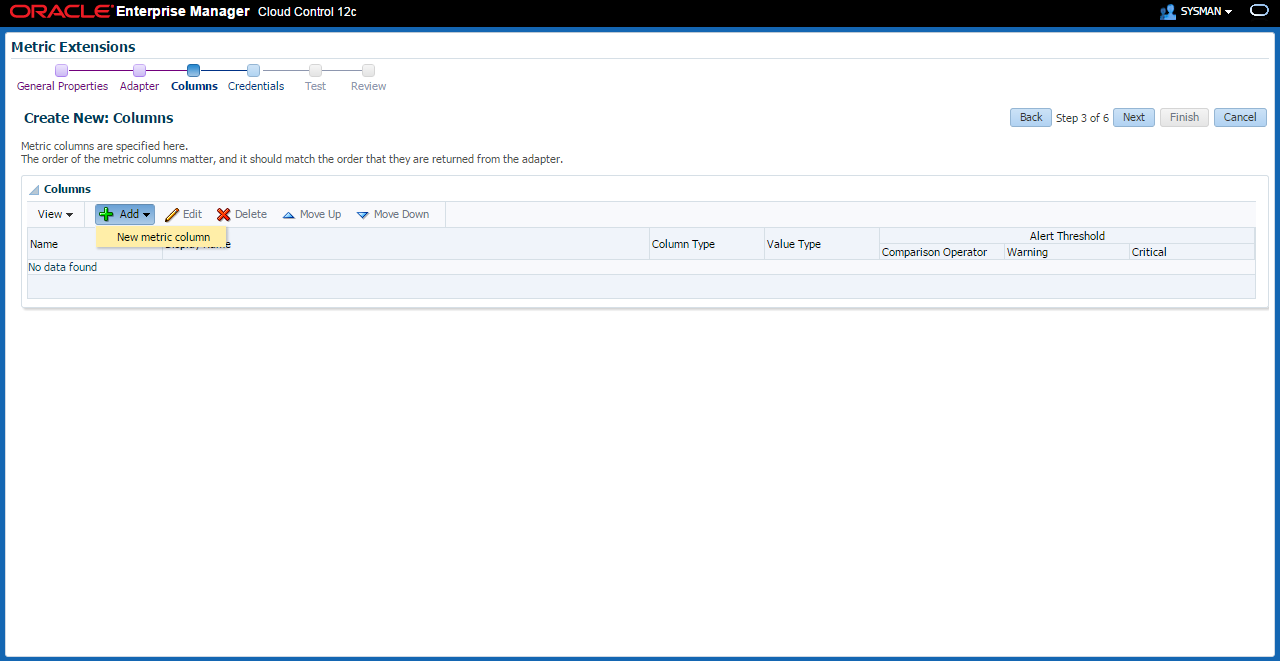




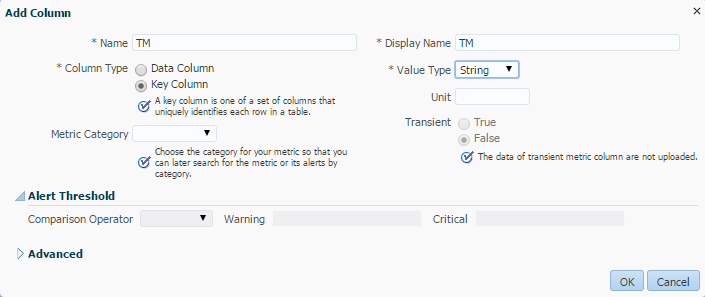
1. Next



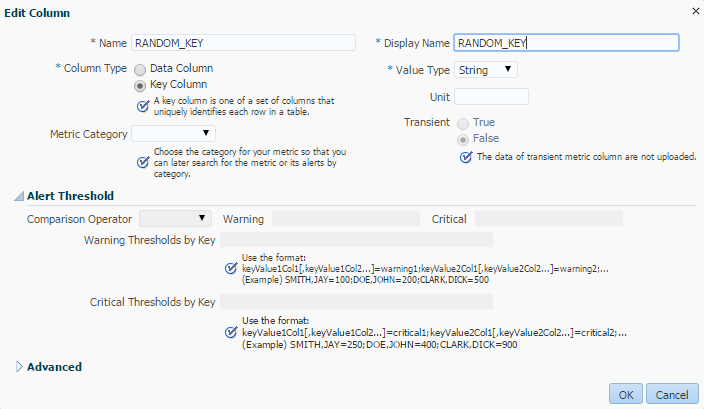
1. Add -> New Metric Column



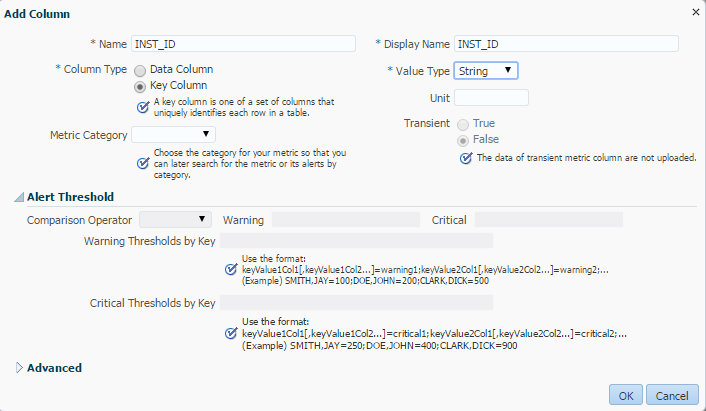
1. On Add Column: TM
   1. Name: TM
   2. Display Name: TM
   3. Column Type: Key Column
   4. Value Type: String



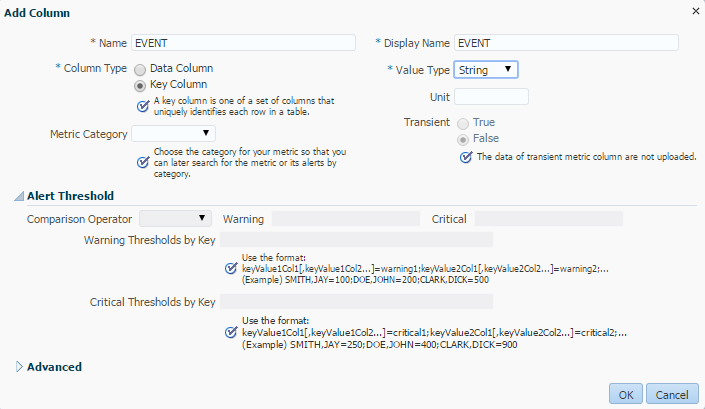
1. On Add Column: RANDOM\_KEY
   1. Name: RANDOM\_KEY
   2. Display Name: RANDOM\_KEY
   3. Column Type: Key Column
   4. Value Type: String



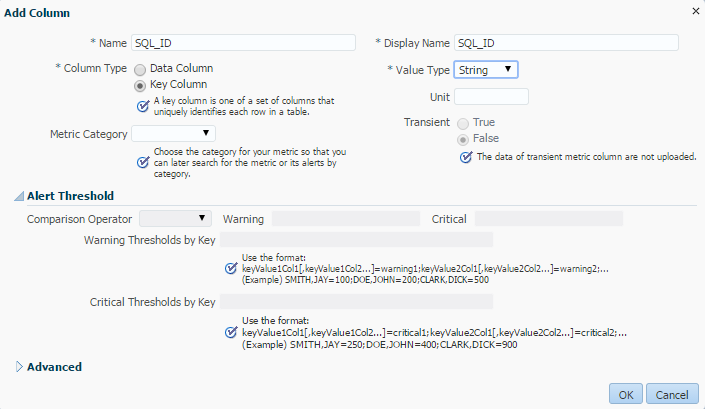
1. On Add Column: INST\_ID
   1. Name: INST\_ID
   2. Display Name: INST\_ID
   3. Column Type: Key Column
   4. Value Type: String



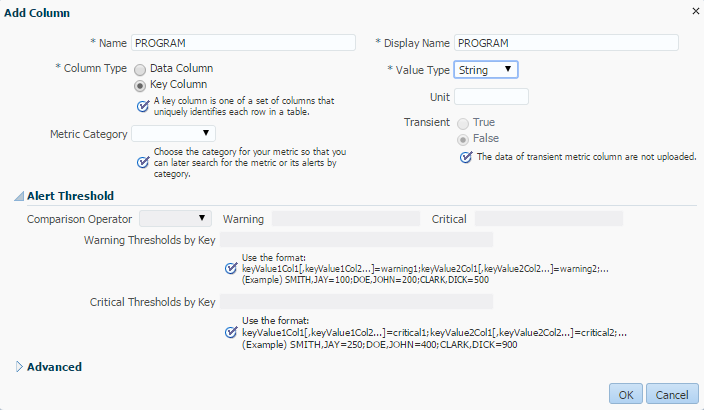
1. On Add Column: EVENT
   1. Name: EVENT
   2. Display Name: EVENT
   3. Column Type: Key Column
   4. Value Type: String



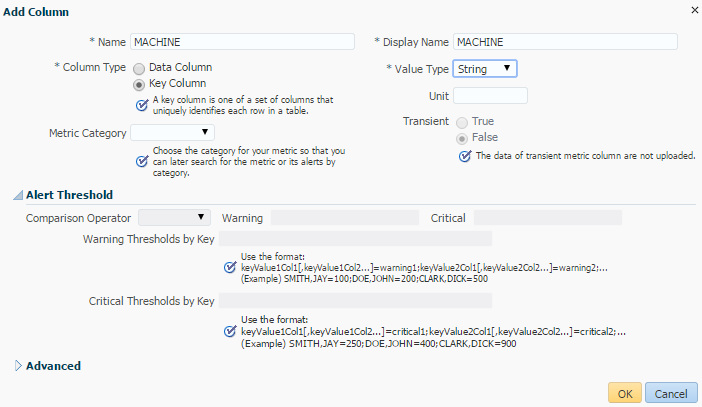
1. On Add Column: SQL\_ID
   1. Name: SQL\_ID
   2. Display Name: SQL\_ID
   3. Column Type: Key Column
   4. Value Type: String



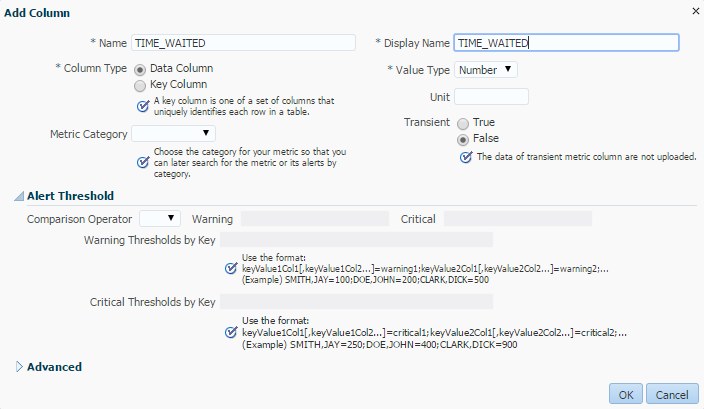
1. On Add Column: PROGRAM
   1. Name: PROGRAM
   2. Display Name: PROGRAM
   3. Column Type: Key Column
   4. Value Type: String



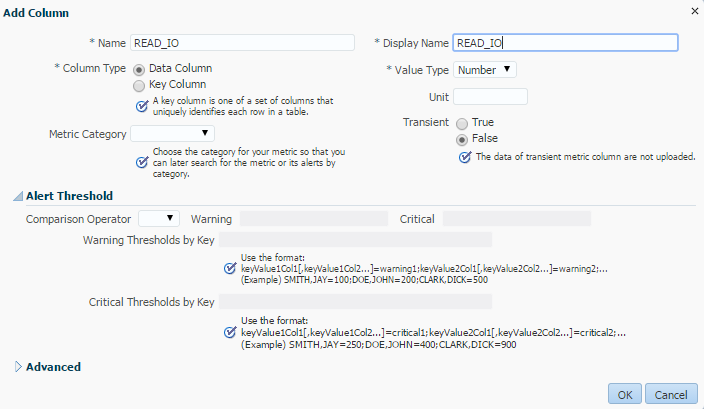
1. On Add Column: MACHINE
   1. Name: MACHINE
   2. Display Name: MACHINE
   3. Column Type: Key Column
   4. Value Type: String



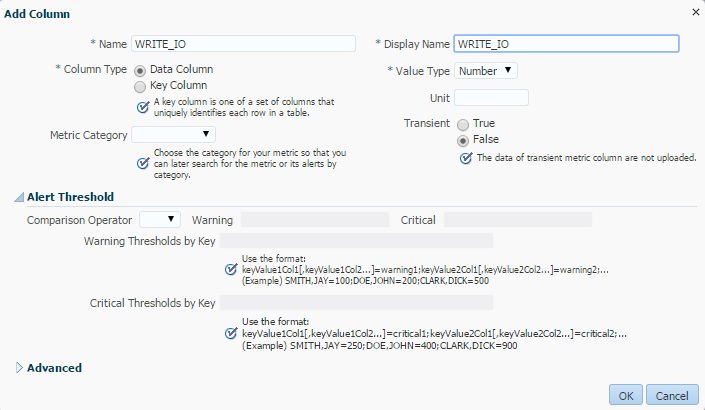
1. On Add Column: TIME\_WAITED
   1. Name: TIME\_WAITED
   2. Display Name: TIME\_WAITED
   3. Column Type: Data Column
   4. Value Type: Number



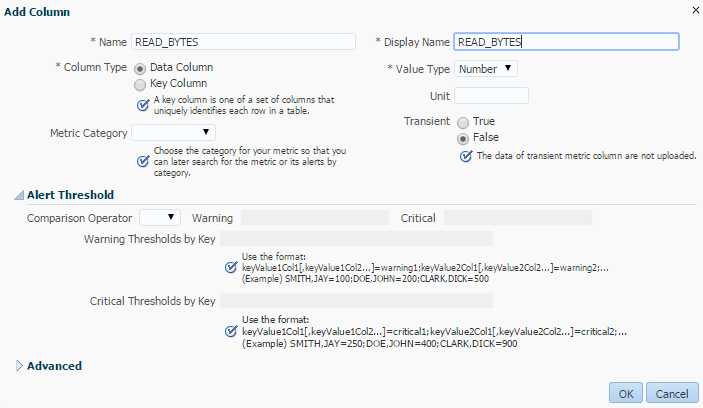
1. On Add Column: READ\_IO
   1. Name: READ\_IO
   2. Display Name: READ\_IO
   3. Column Type: Data Column
   4. Value Type: Number



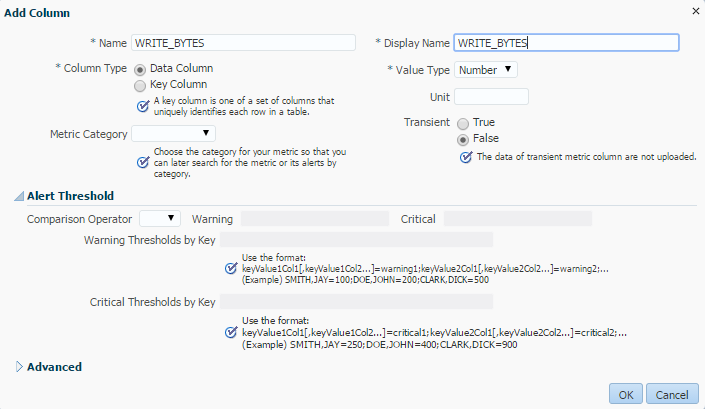
1. On Add Column: WRITE\_IO
   1. Name: WRITE\_IO
   2. Display Name: WRITE\_IO
   3. Column Type: Data Column
   4. Value Type: Number



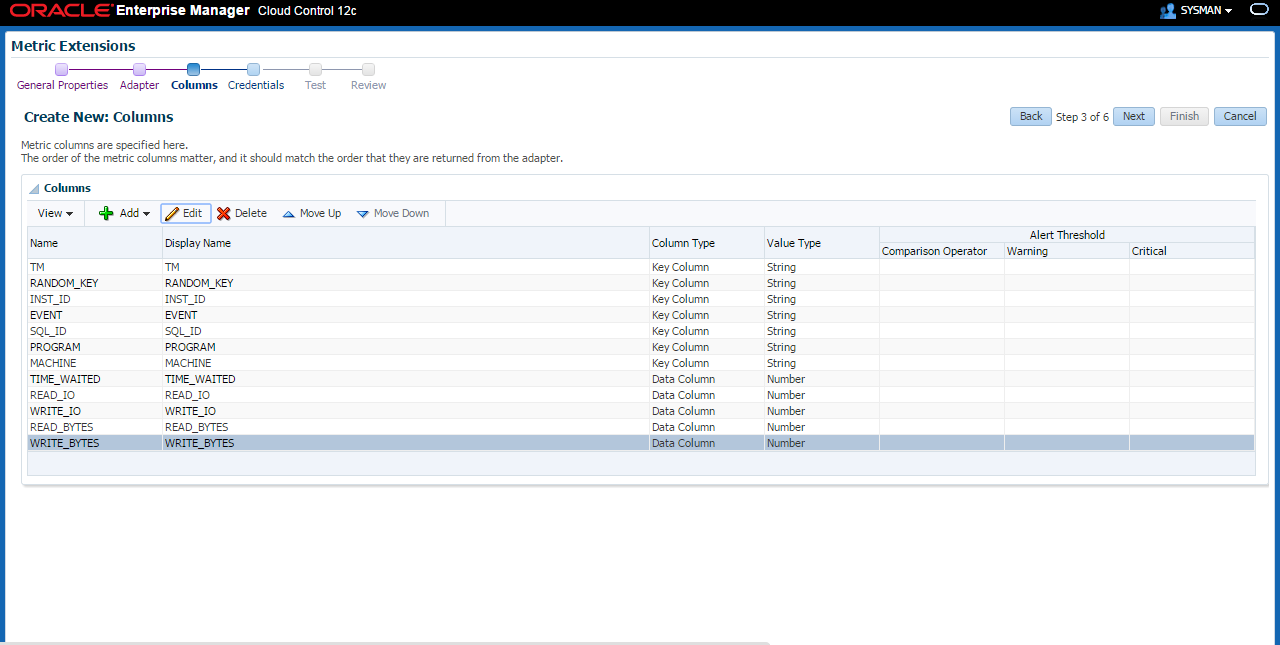
1. On Add Column: READ\_BYTES
   1. Name: READ\_BYTES
   2. Display Name: READ\_BYTES
   3. Column Type: Data Column
   4. Value Type: Number



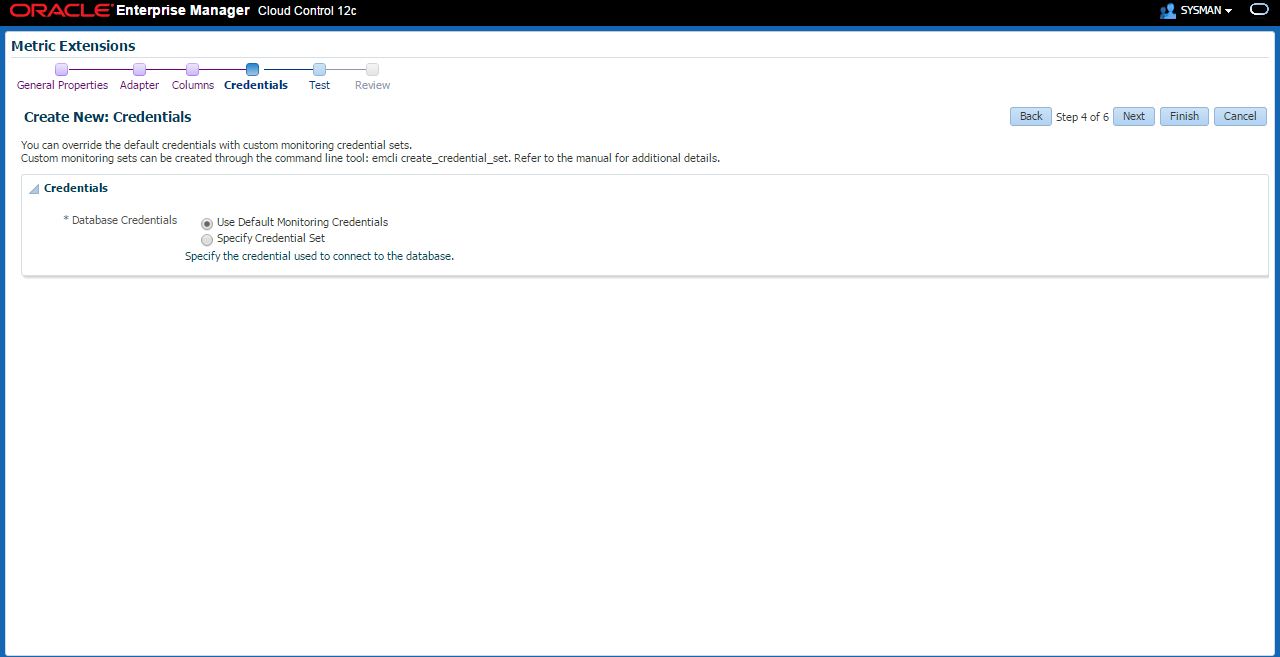
1. On Add Column: WRITE\_BYTES
   1. Name: WRITE\_BYTES
   2. Display Name: WRITE\_BYTES
   3. Column Type: Data Column
   4. Value Type: Number



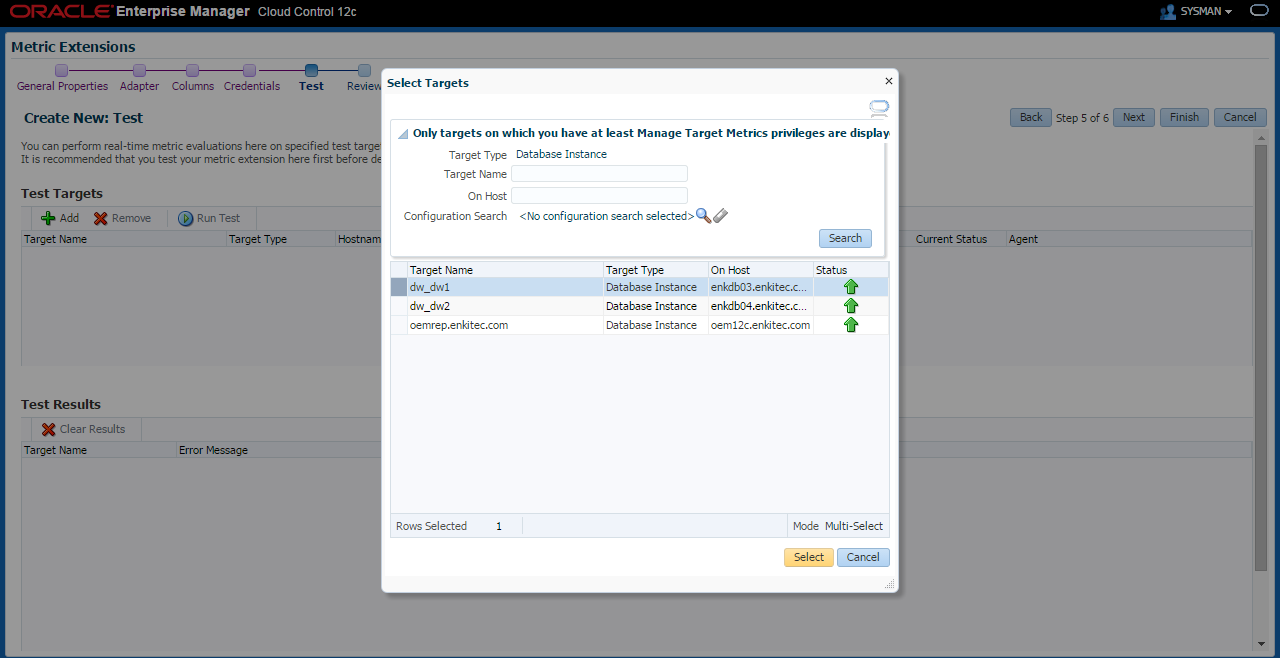
1. There should be a total of 12 columns created
   1. 7 key columns
   2. 5 data columns



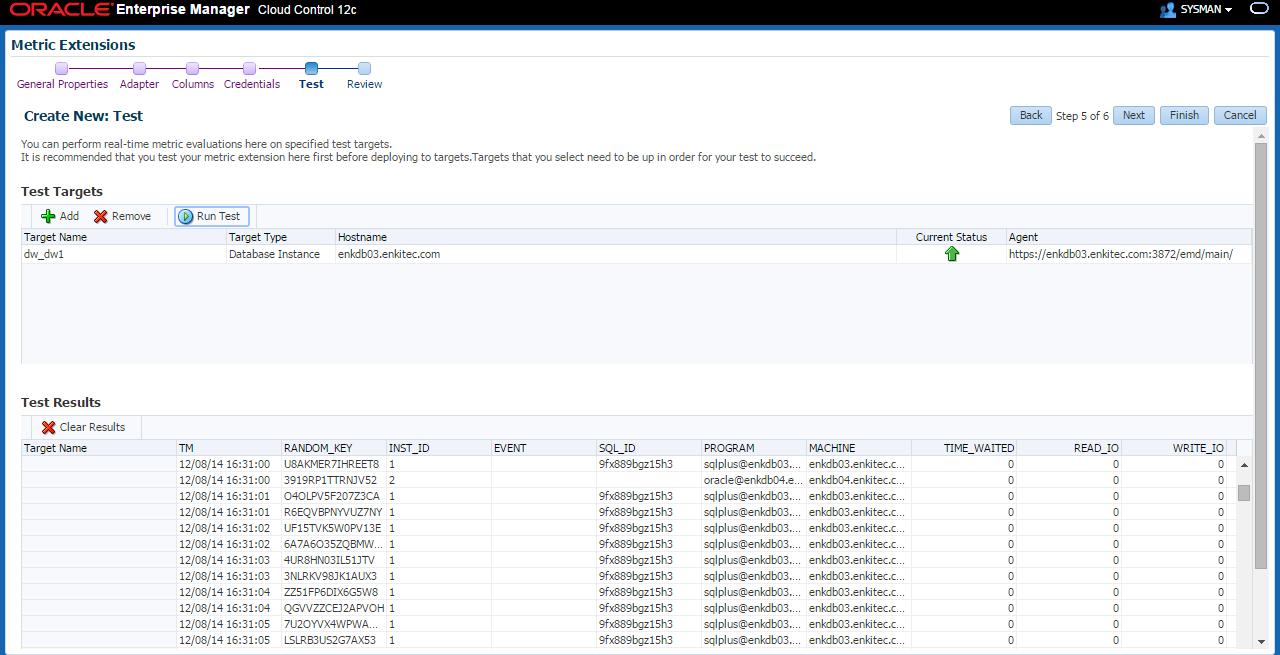
1. Use Default Monitoring Credentials -> Next



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



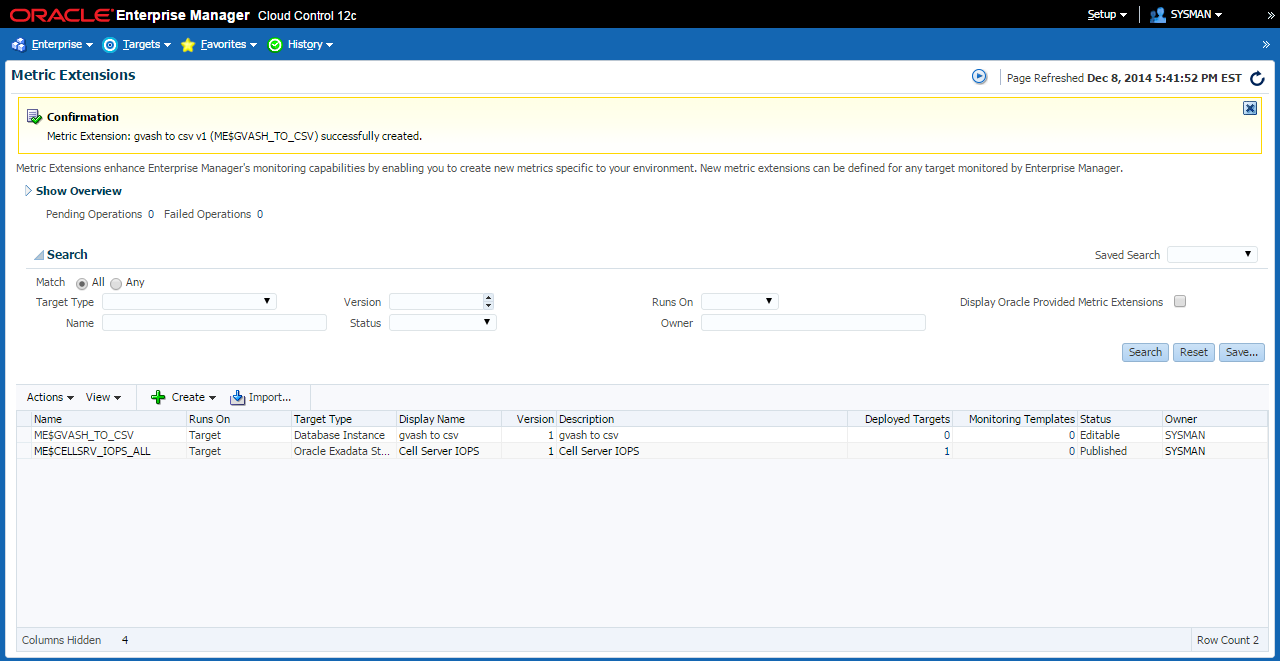
1. Click on Run Test -> Next



1. Review the details -> Finish



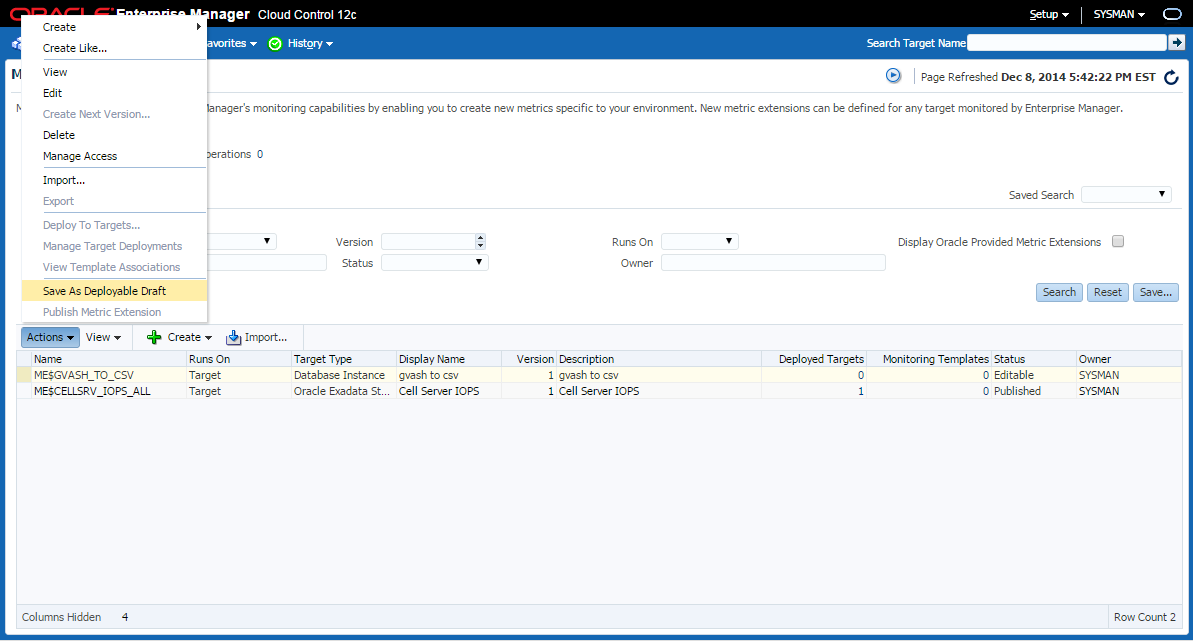
1. Metric Extension is created



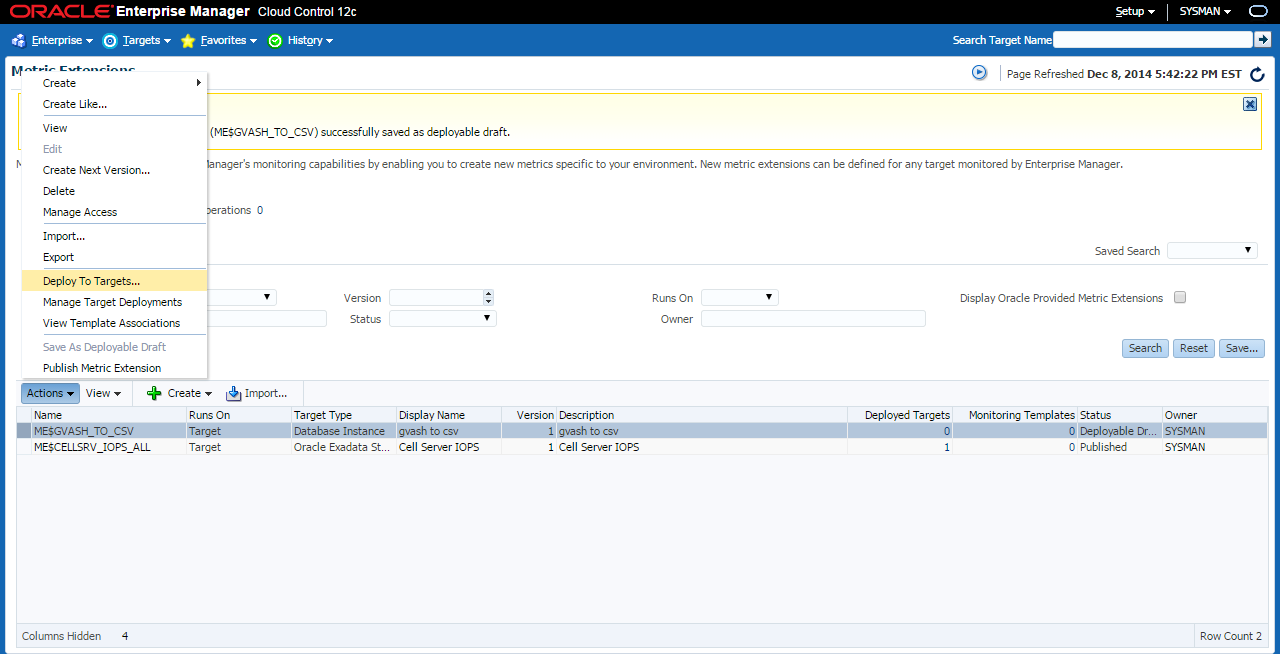
|  |
| --- |
|  |

1. **Deploy Metric Extension**
2. On the newly created Metric Extension -> Actions -> Save As Deployable Draft

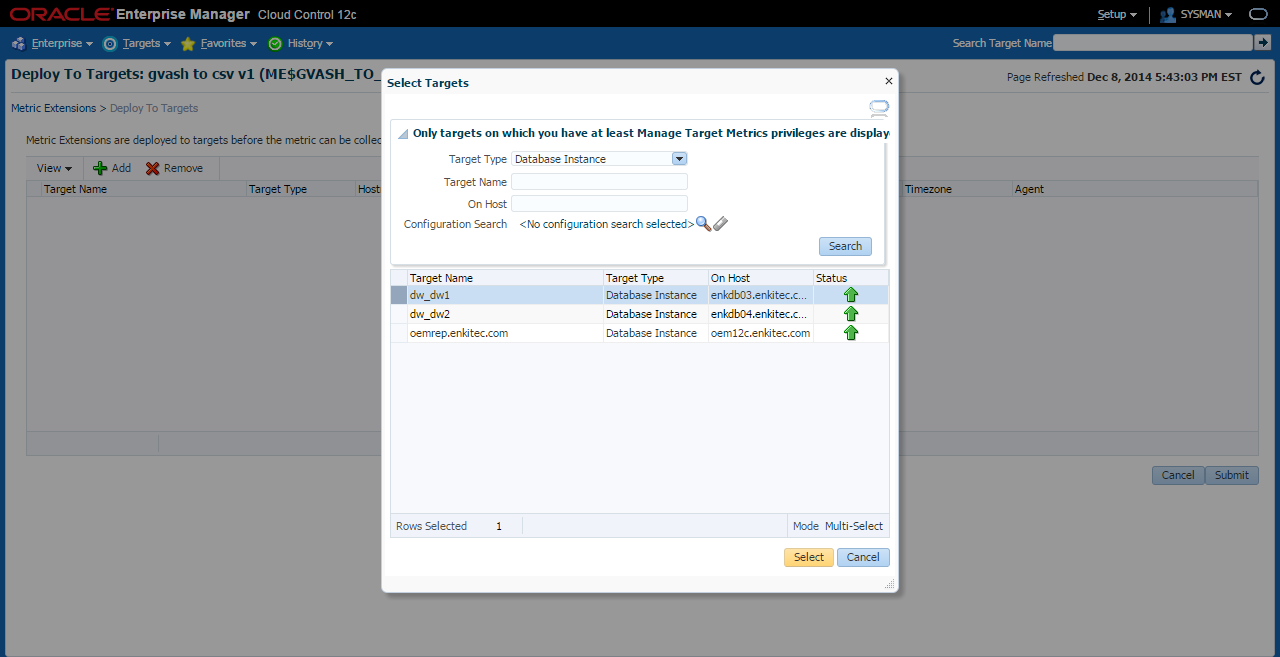
* Newly created Metric Extension has a status of “Editable”. It should be on “Deployable Draft” so you can assign targets to it. And then status of “Published” to start collecting data.



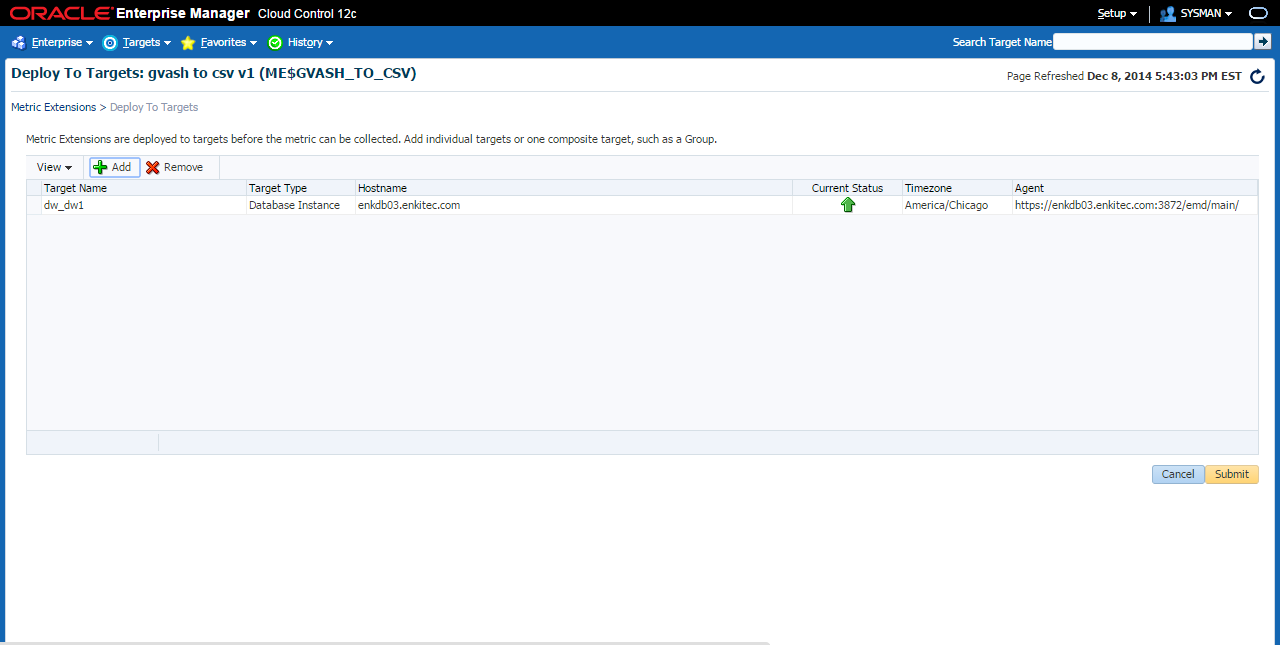
1. Actions -> Deploy To Targets



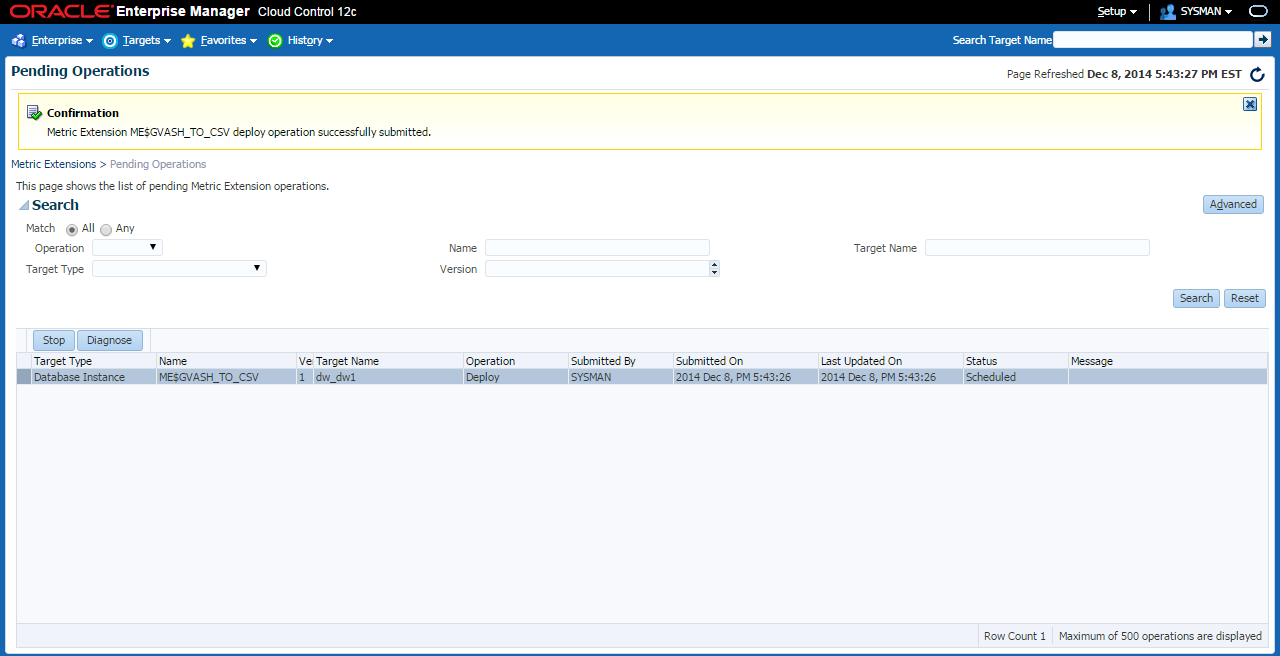
1. Select the 1st instance of any cluster database



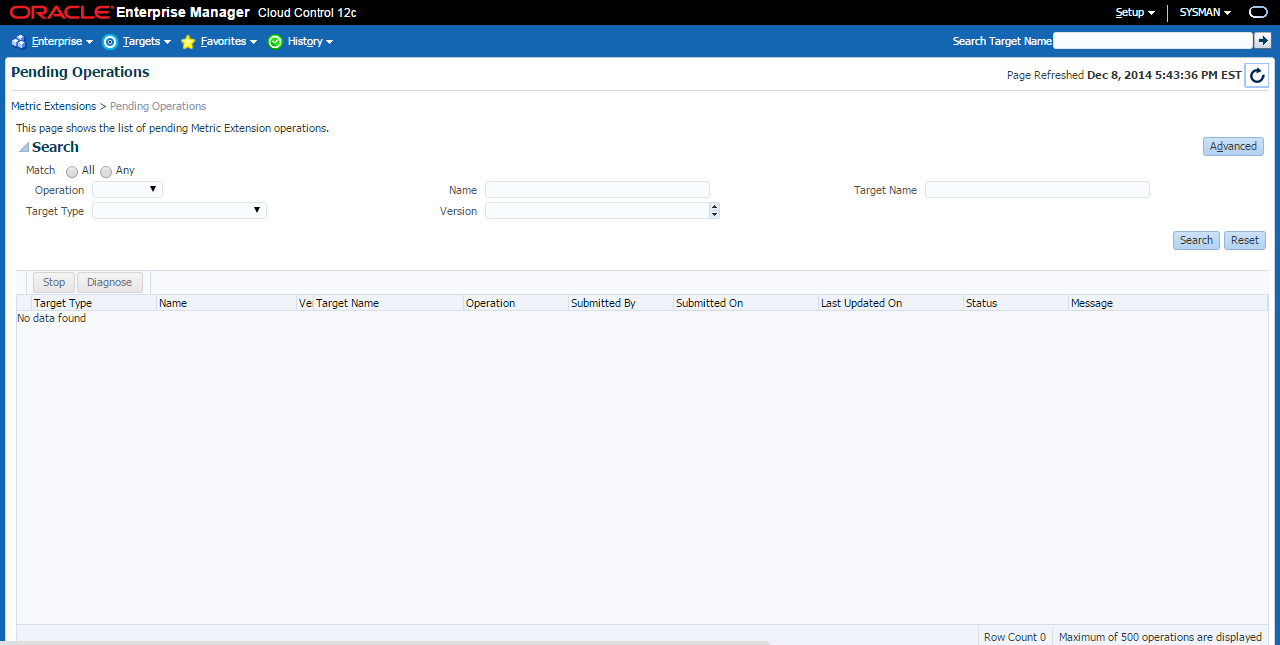
1. Submit



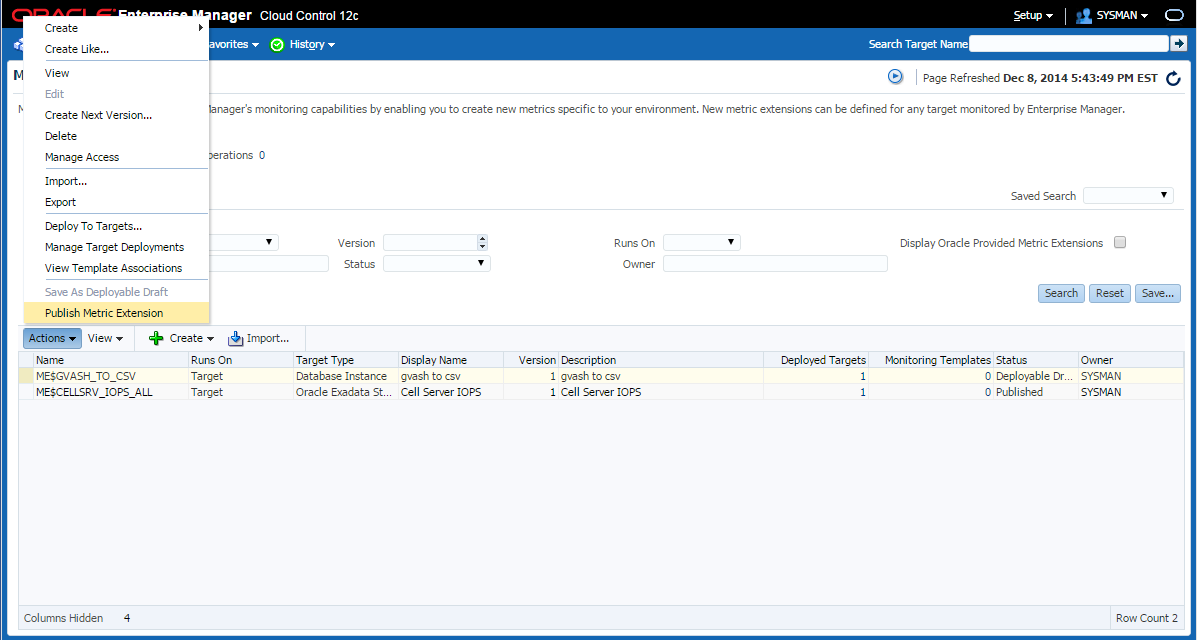
1. Hit “Refresh” (top right)



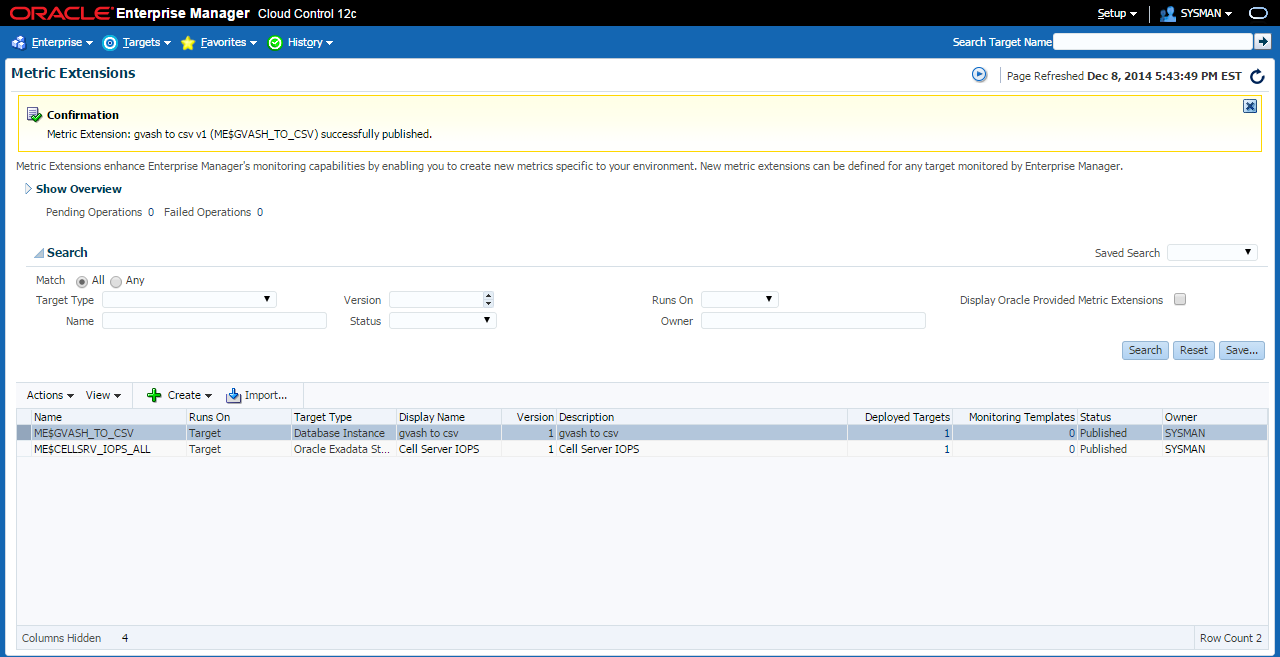
1. No more Pending Operations -> Click on Metric Extensions



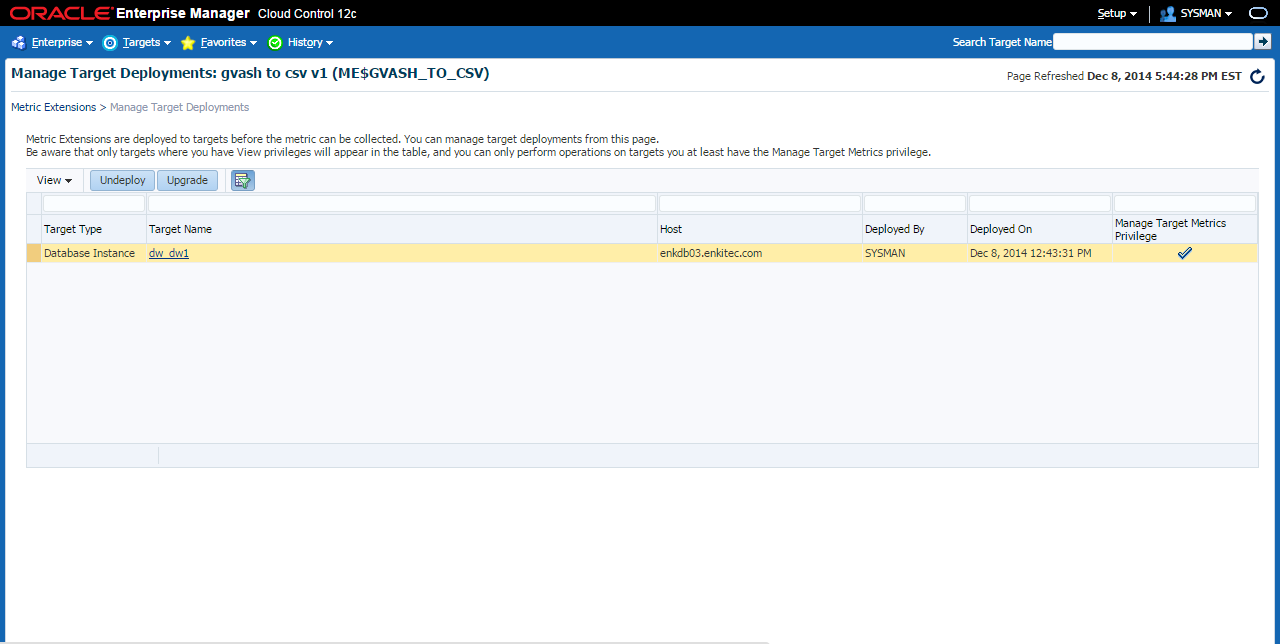
1. The Deployed Targets is now 1 -> go to Actions -> Publish Metric Extension (this will start the data collection)



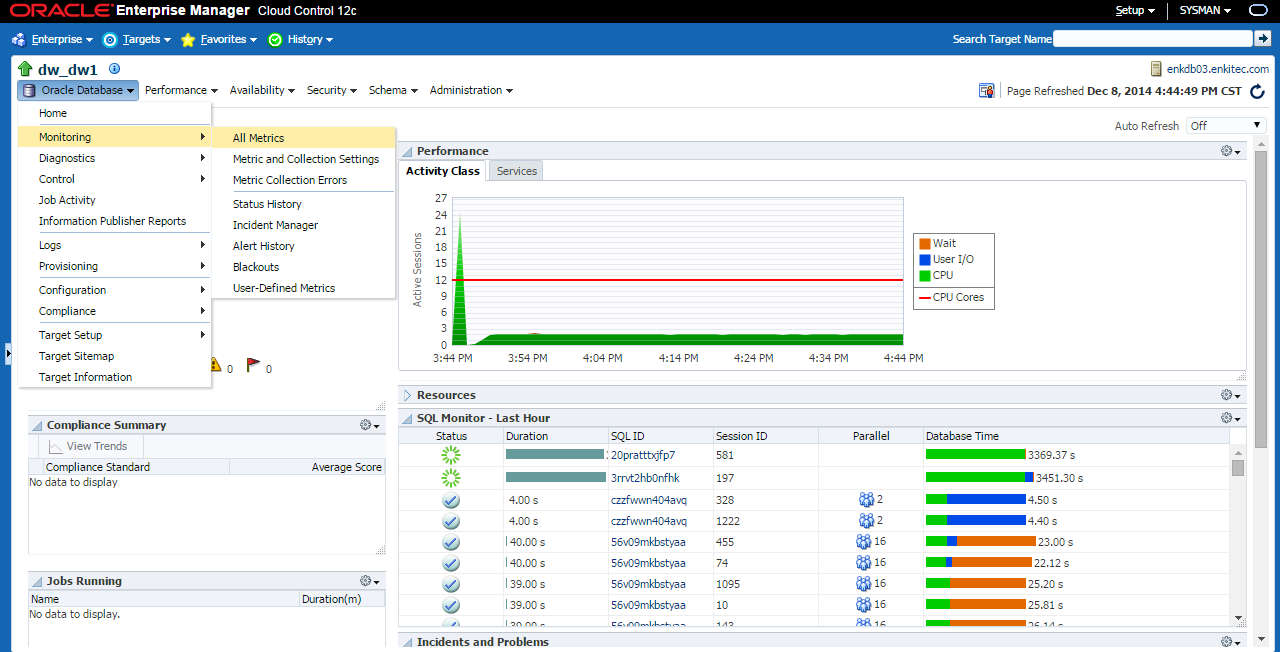
1. To check the data collection of the newly created Metric Extension -> Click on the number (1) on the Deployed Targets column, it’s also right next to the “gvash to csv” Description



1. Click on the Target Name

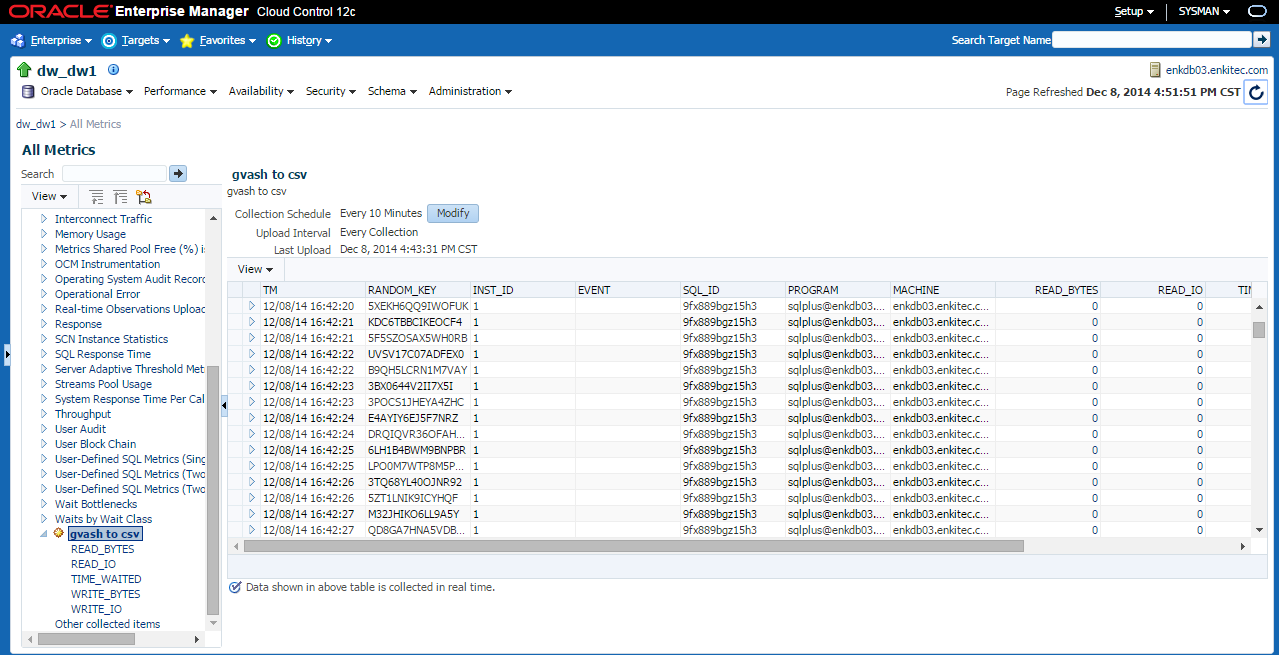


1. Monitoring -> All Metrics



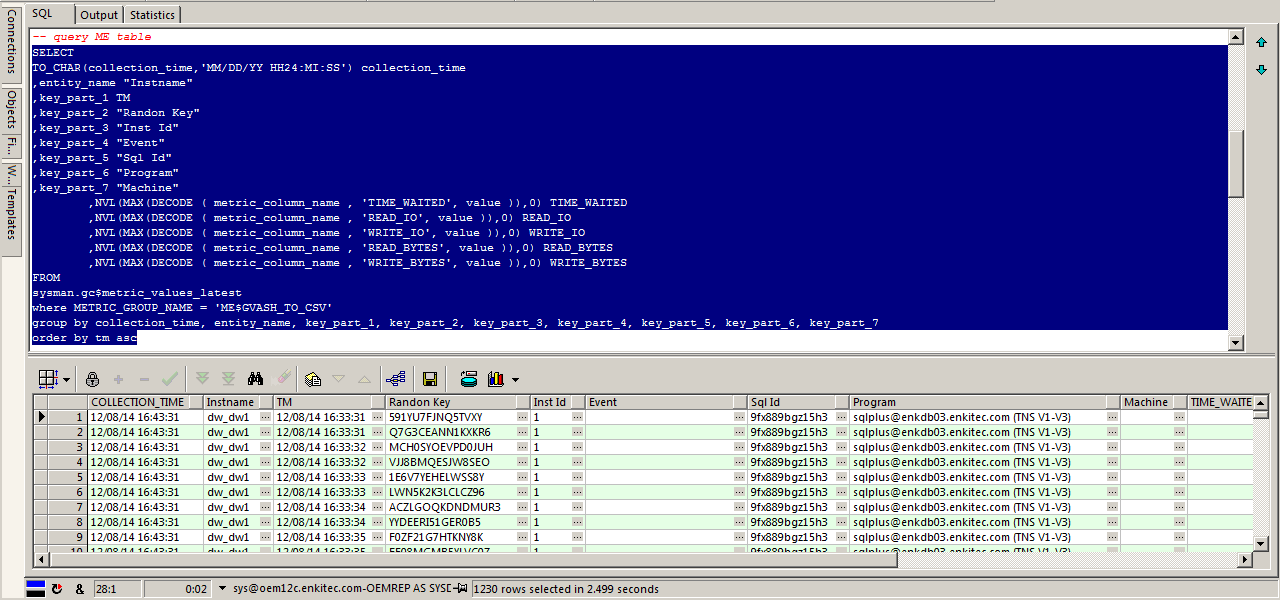
1. Scroll down -> click on “gvash to csv”

* 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.



1. 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**

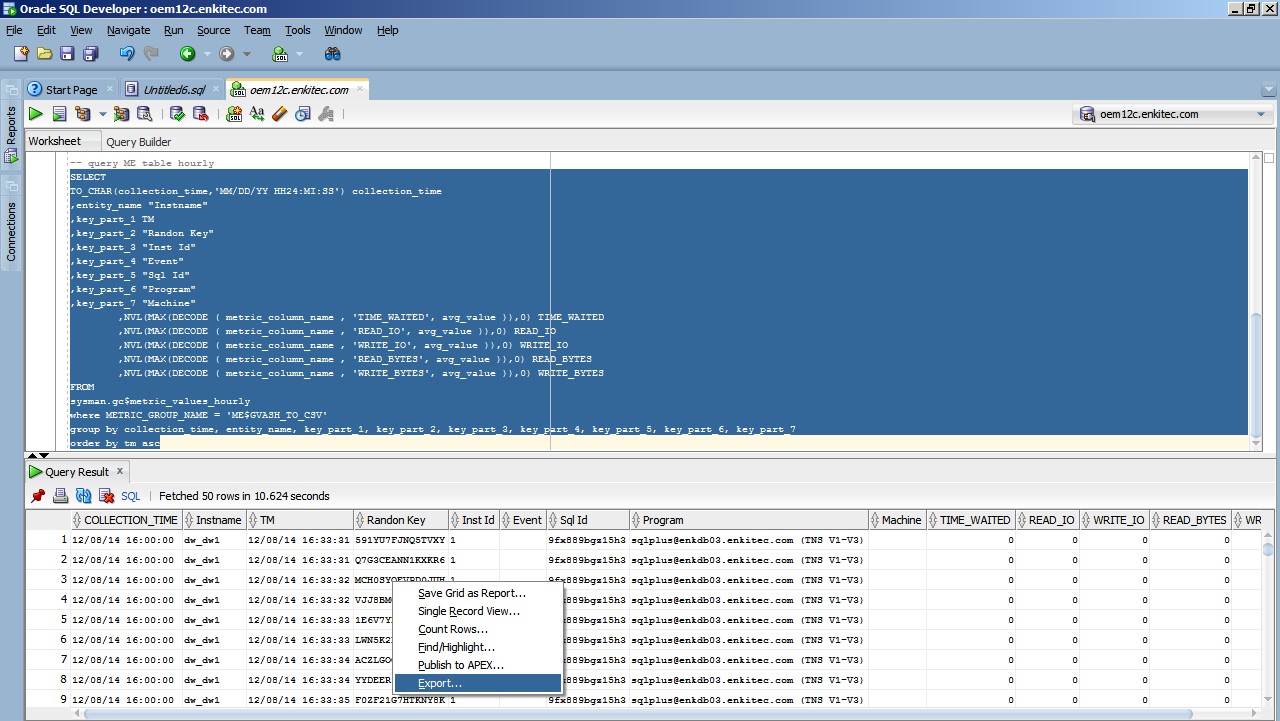
* “sysman.gc$metric\_values\_latest” shows the most recent collections

|  |
| --- |
| -- 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 |

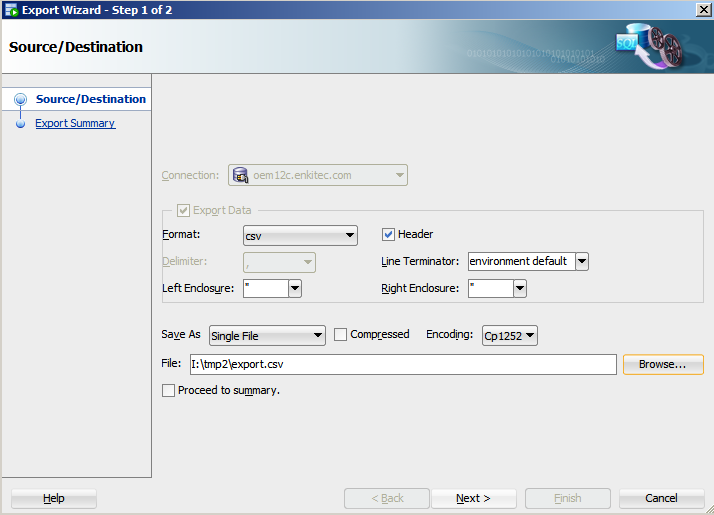
* “sysman.gc$metric\_values\_hourly” shows the hourly history view of the Metric Extension data

|  |
| --- |
| -- 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



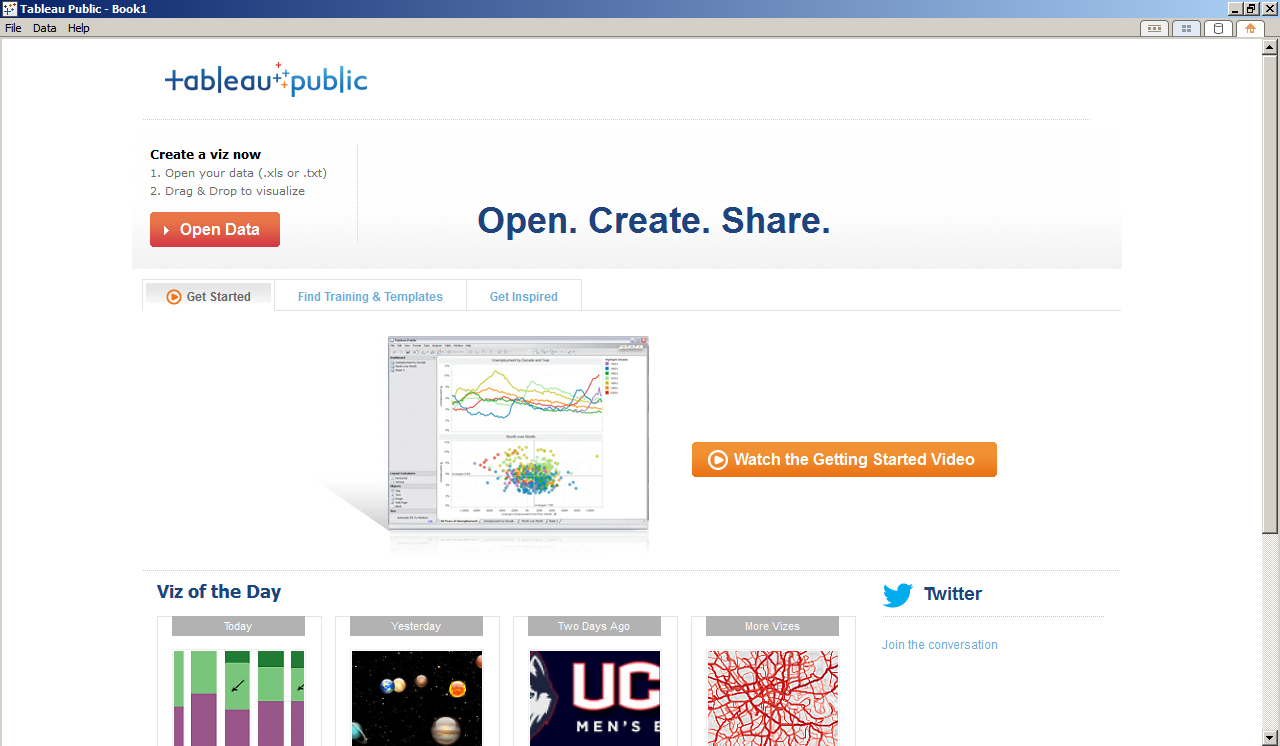
1. Select format as CSV -> Next -> Finish



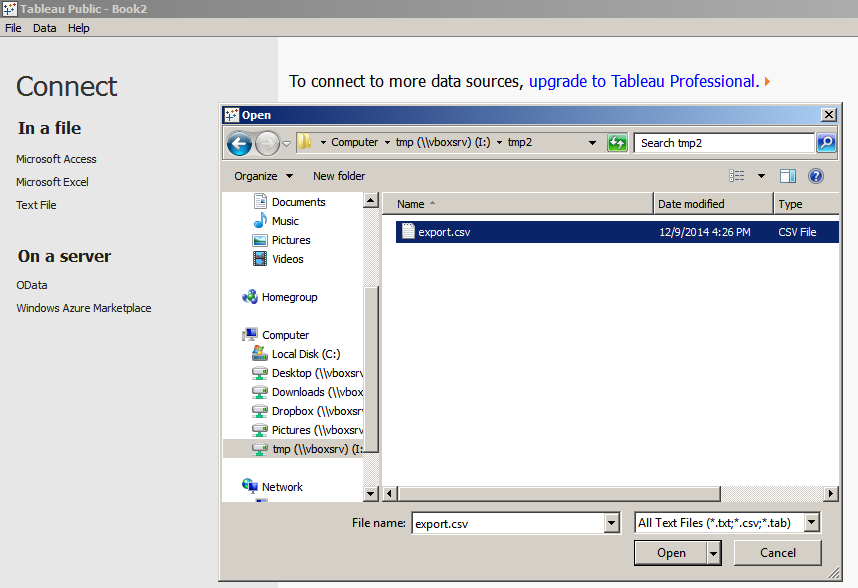
|  |
| --- |
|  |

**Appendix B: Graphing the Metric Extension data**

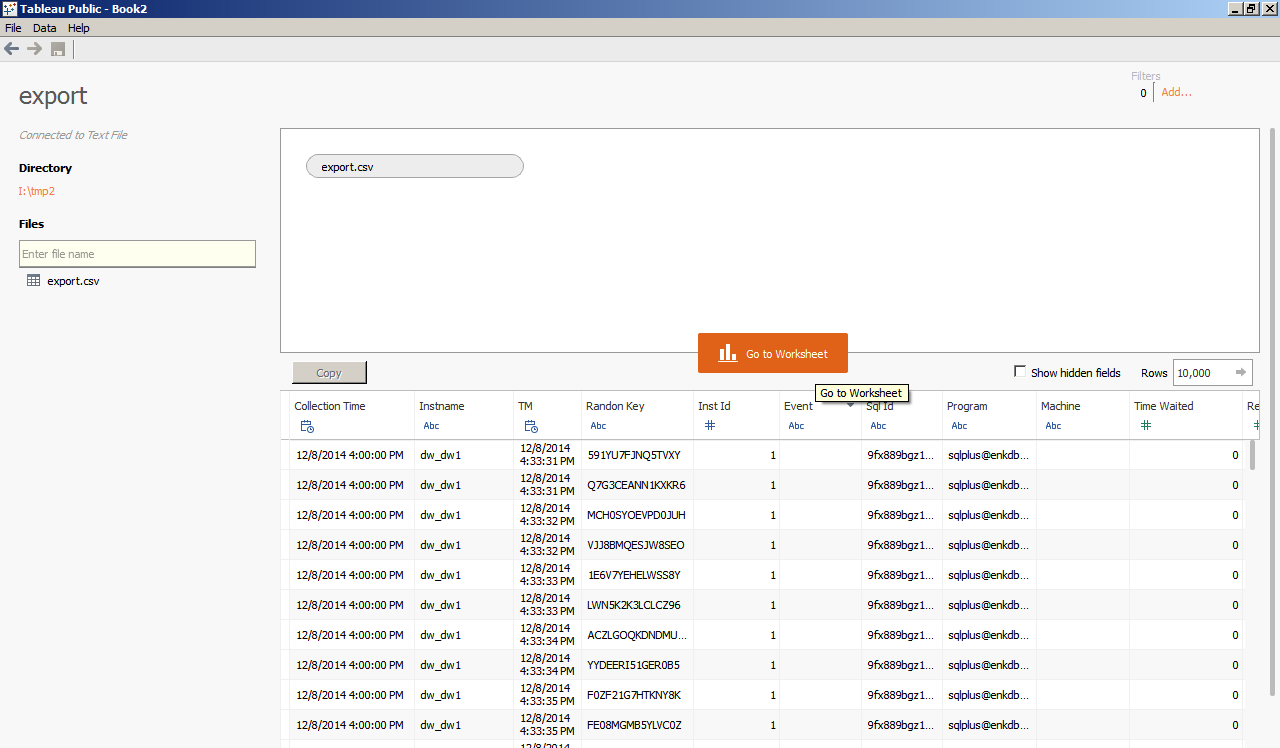
1. Download the tableau public at <http://www.tableausoftware.com/public/>
2. Click Open Data



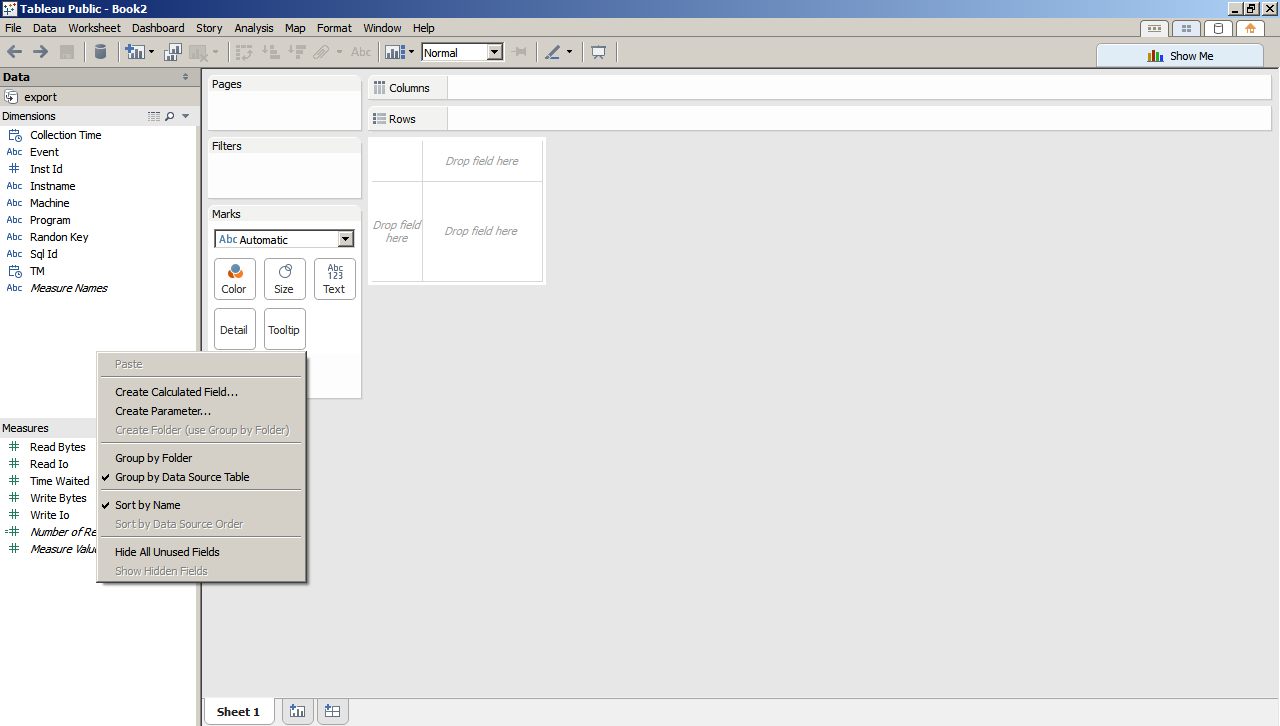
1. Click Text File -> Select the file



1. Click Go To Worksheet



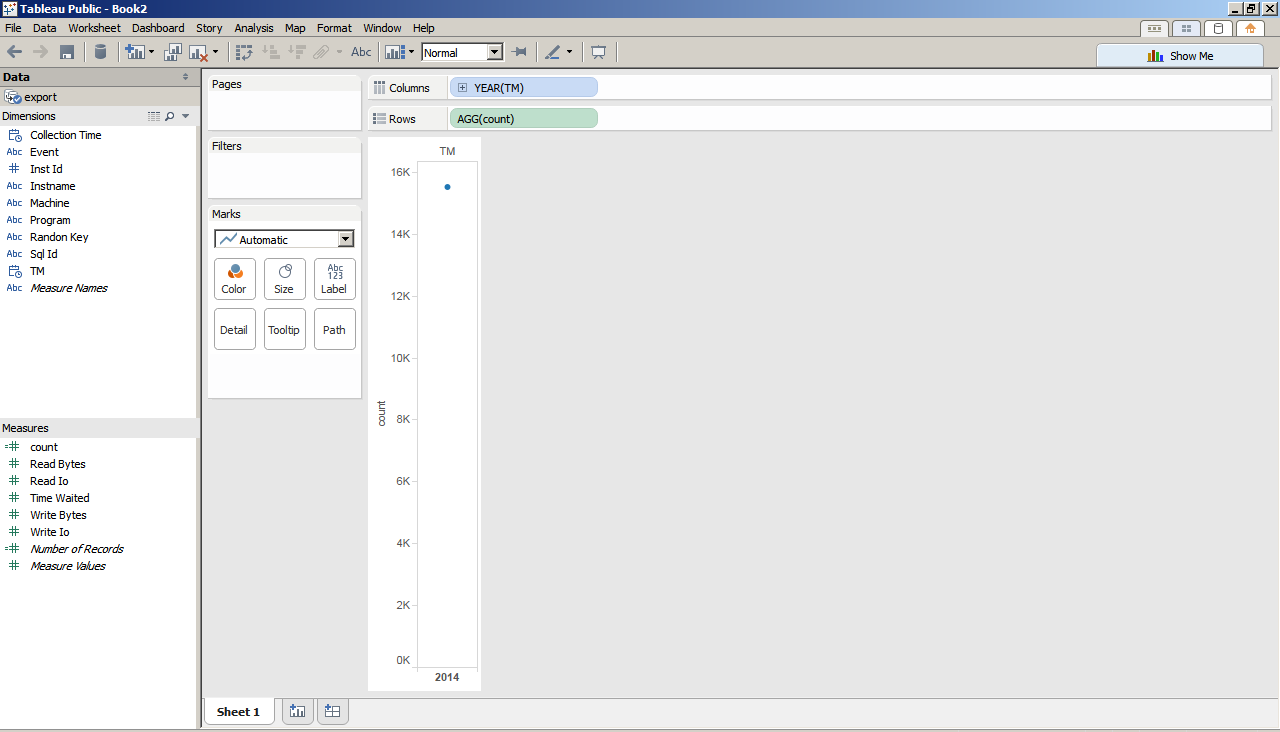
1. Right click on Measures section -> Create Calculated Field



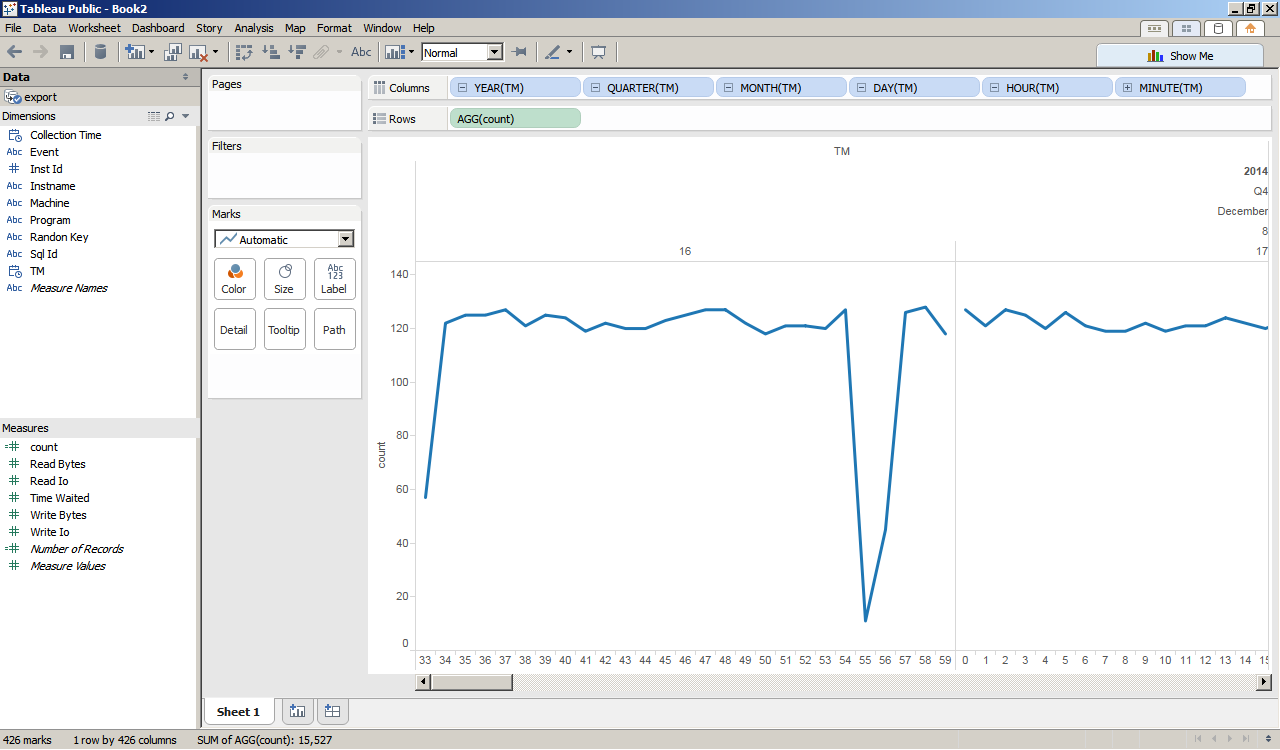
1. Create the “count” calculated field -> OK
   1. Name: count
   2. Formula: count(1)



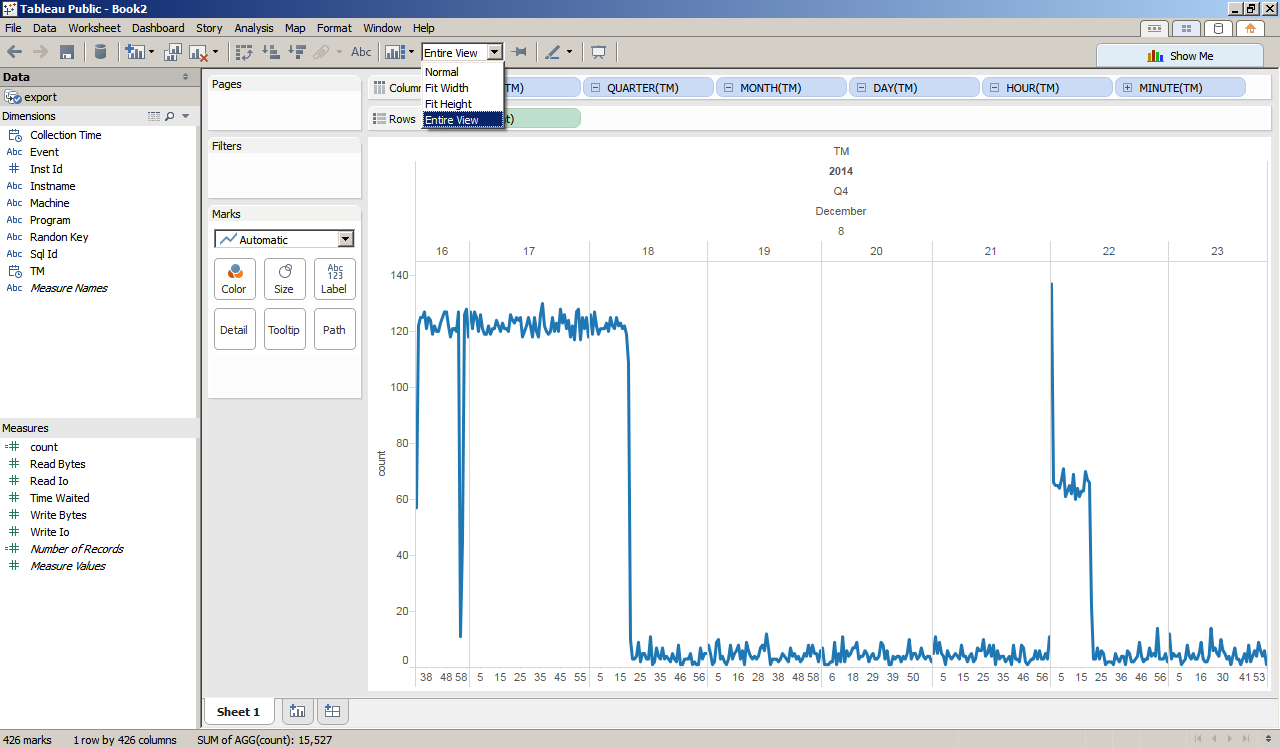
1. Drag the “TM” on the Dimensions section to the Columns. And then drag the “count” on the Measures section to the Rows



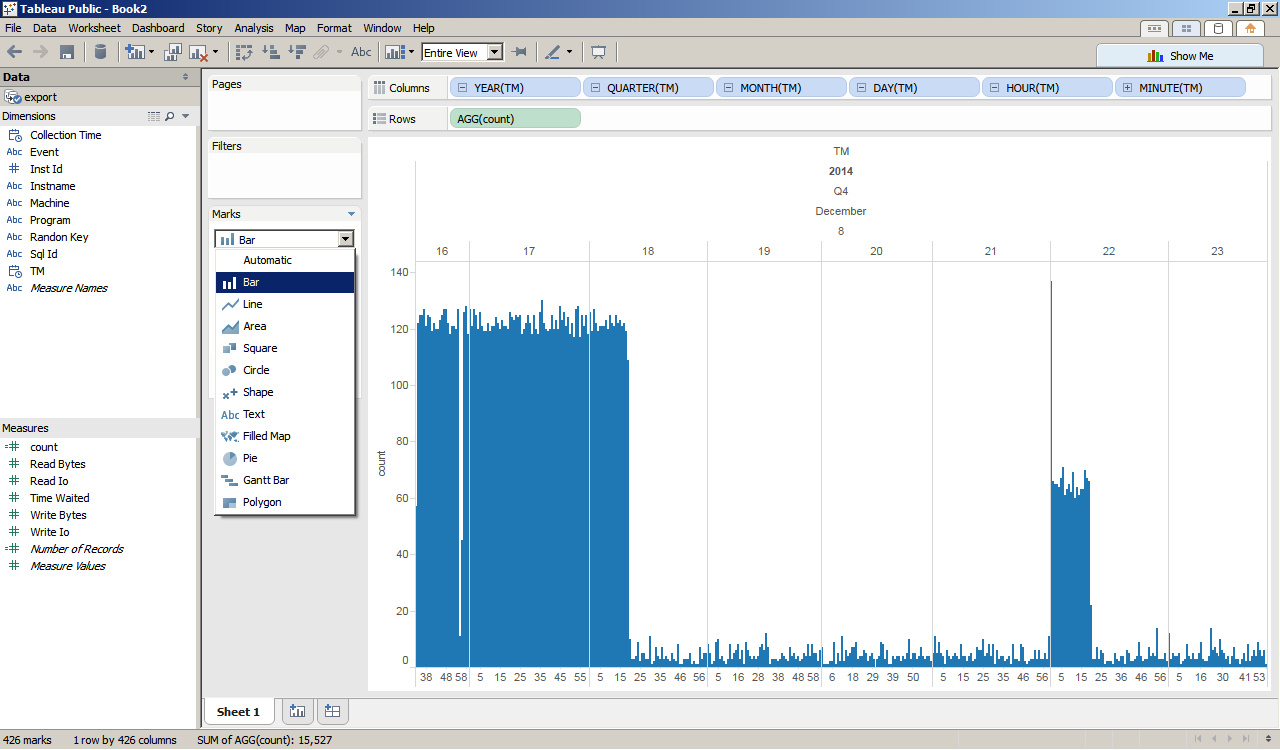
1. Click on the plus sign of the “Year(TM)” up to the “Minute(TM)” dimension



1. Change to “Entire View”



1. Change to “Bar” graph

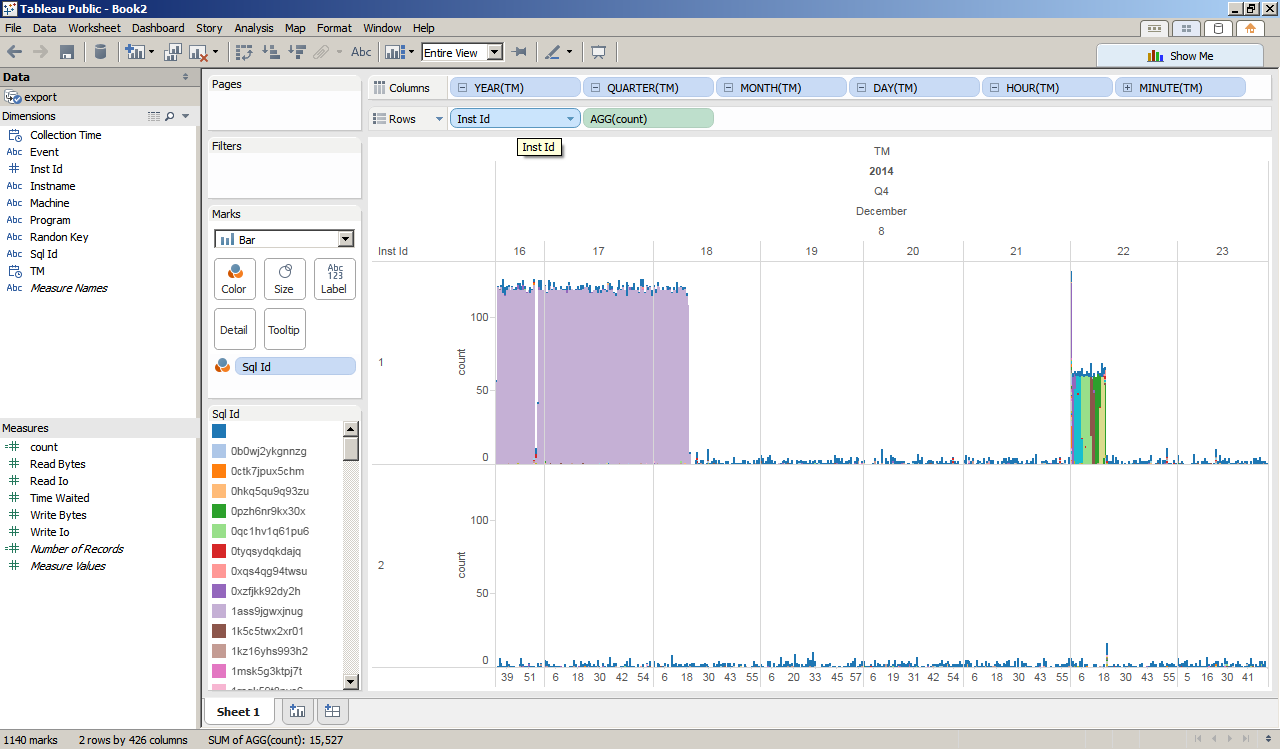


1. Drag the “Sql Id” on the Dimensions section to the Color



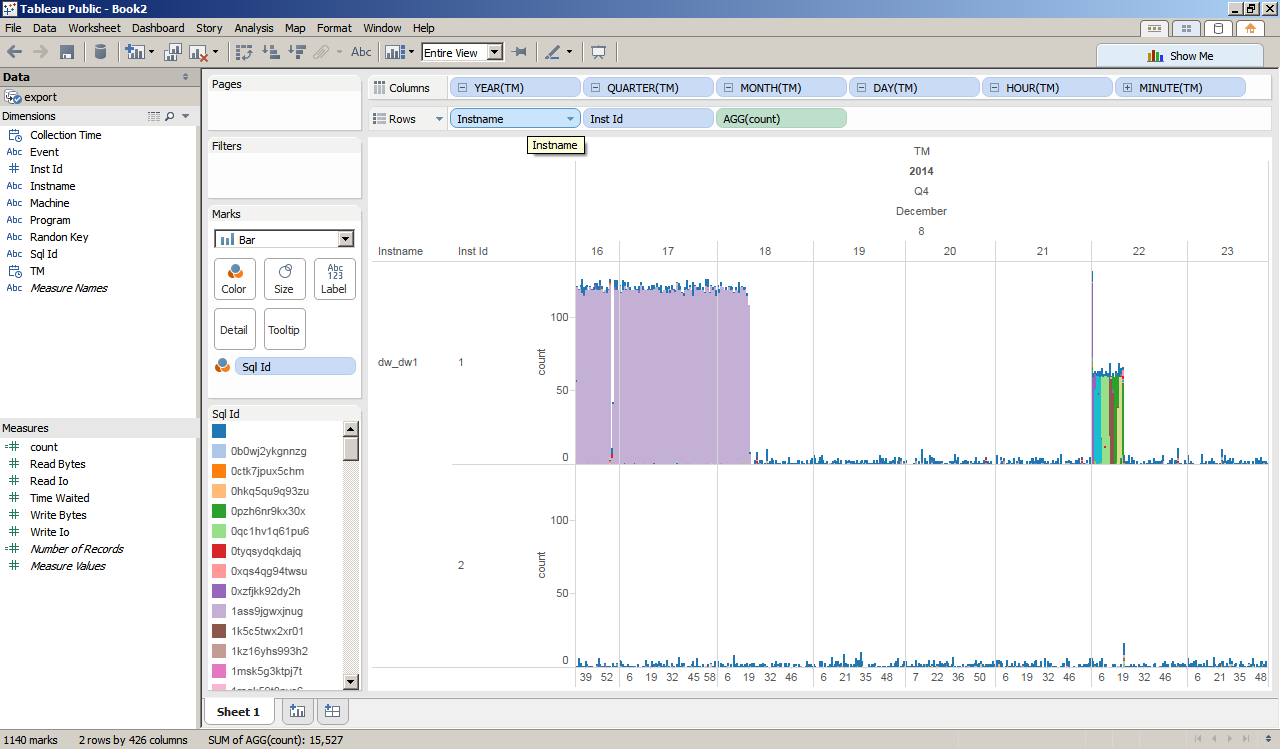
1. Drag the “Inst Id” on the Dimensions section to the Rows

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

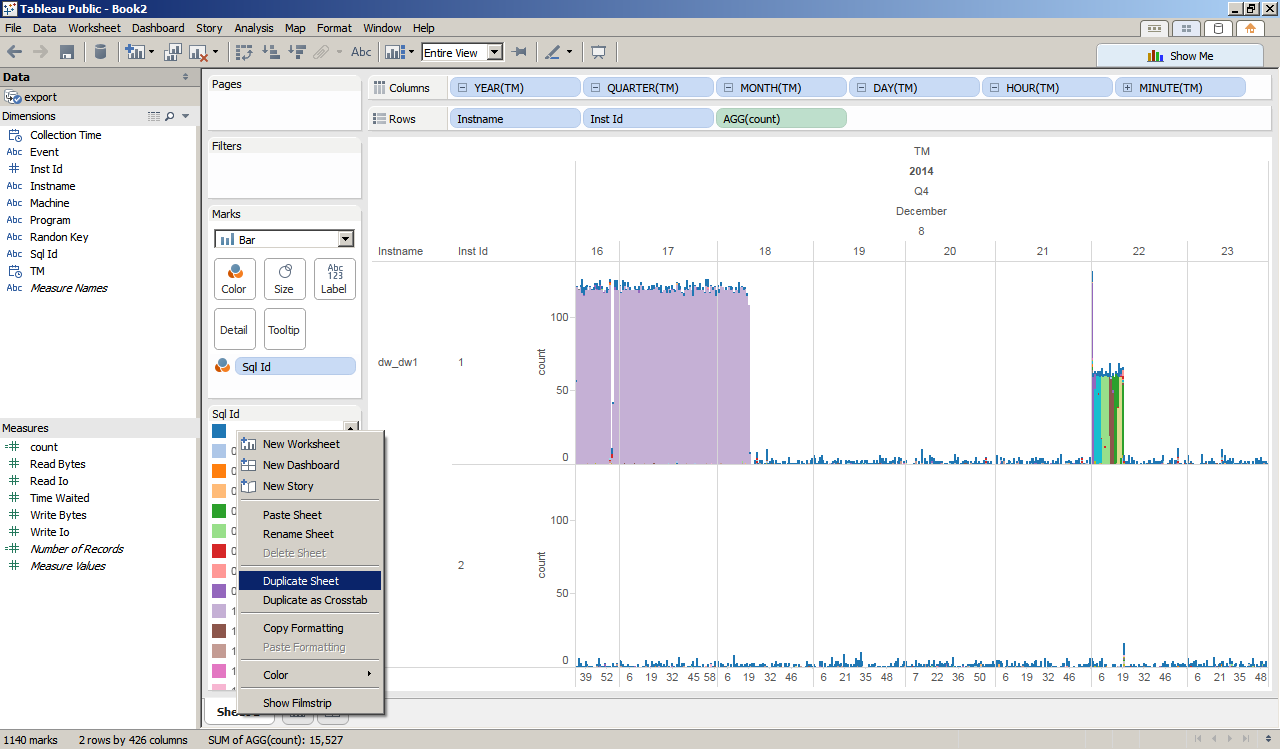


1. Drag the “Instname” on the Dimensions section to the Rows

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

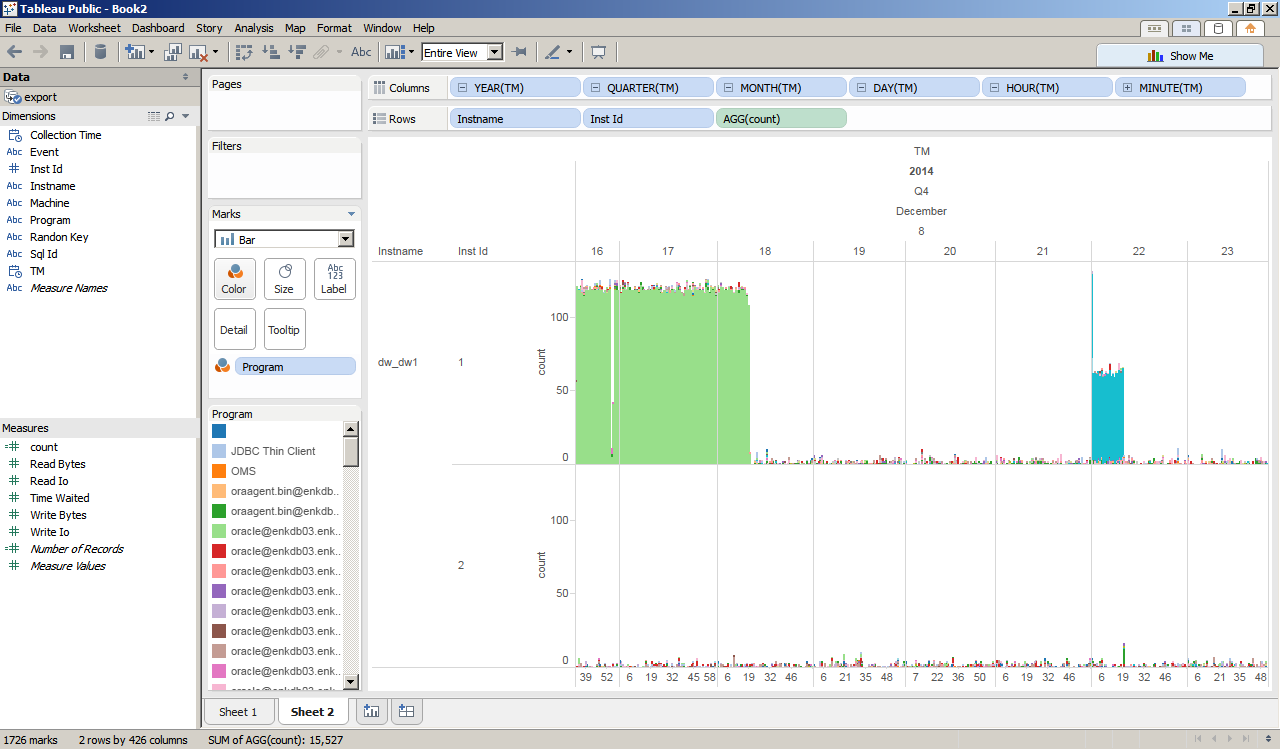


1. Duplicate the current sheet by right clicking on the tab -> Duplicate Sheet



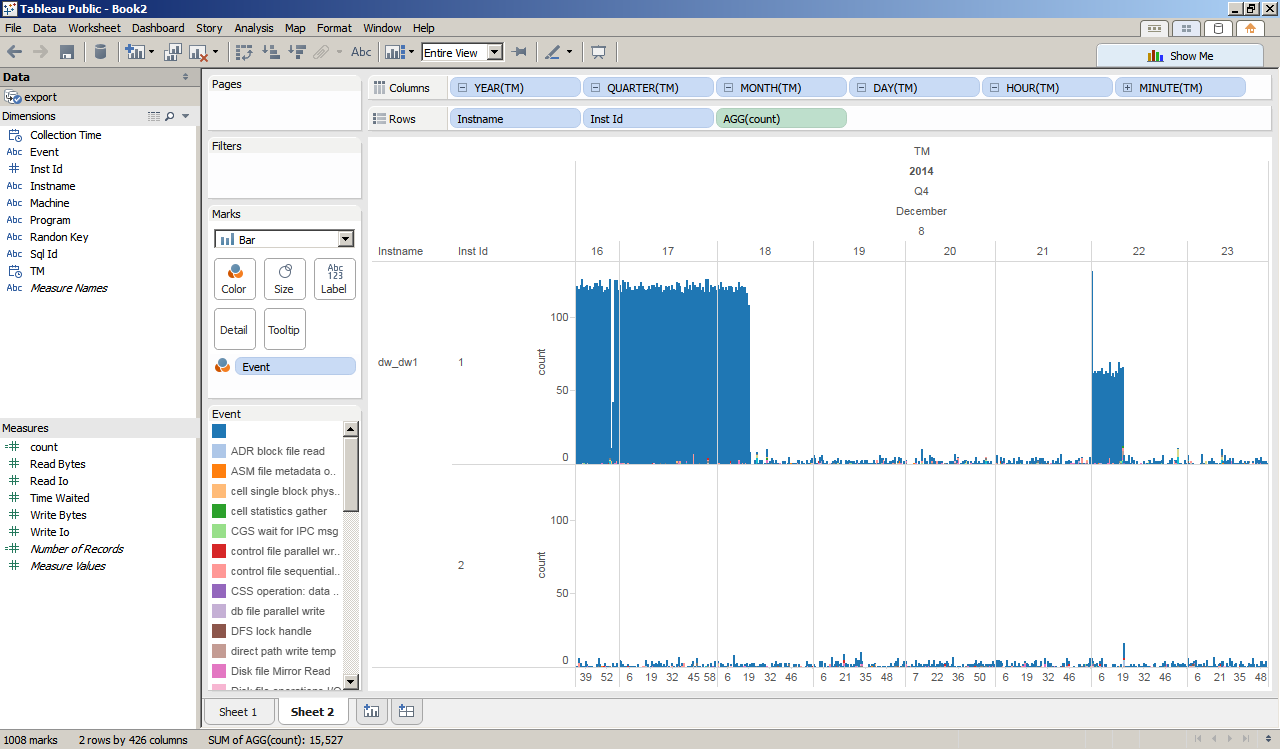
1. Drag the “Program” on the Dimensions section to the Rows

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



1. Drag the “Event” on the Dimensions section to the Rows

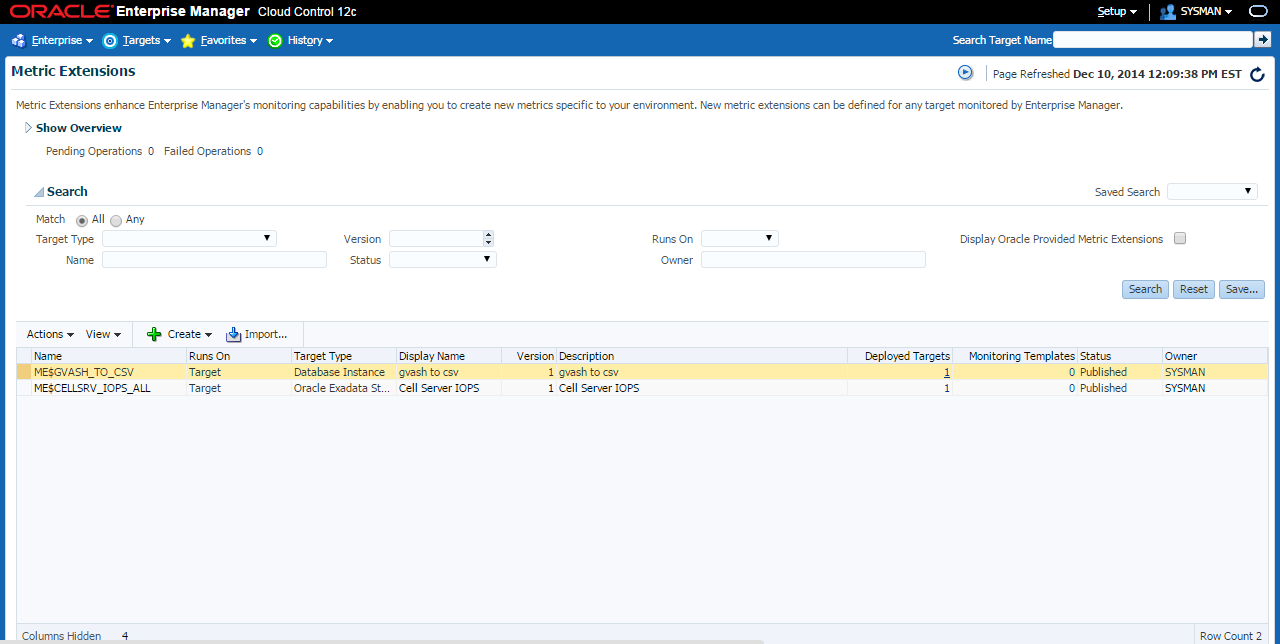
* This will show the distribution of the CPU and wait events across instance 1 and 2
* The “null” event pertains to CPU usage (shown below)



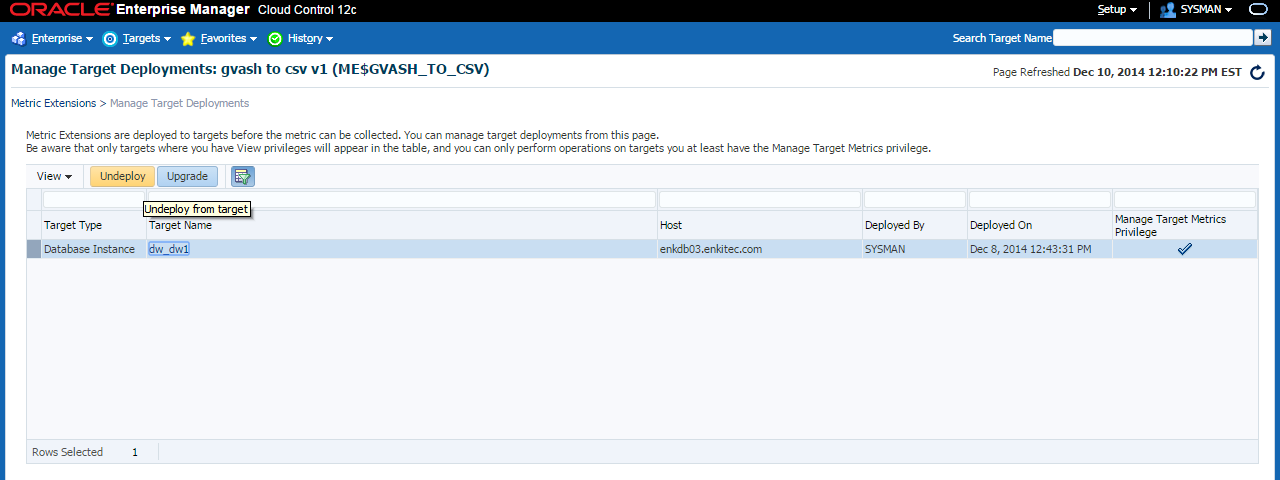
|  |
| --- |
|  |

**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



1. Select the target -> Click Undeploy



1. To delete/uninstall the Metric Extension -> Actions -> Delete

