

Overcommitment ratio dashboard sample.

Aakib Dawood

Objective: Create a dashboard to see the overcommitment ratios at the Datacenter, Cluster and Host level.

Step Overview:

1. Create supermetric.
2. Create Dashboard Widget for Datacenter.
3. Create Dashboard Widget for Cluster.
4. Create Dashboard Widget for Host.
5. Define Interactions

1. Create Supermetric:

First of all, why we need a supermetric in this case is because vROPS has an OOB metric for “Current overcommitment ratio” for Datacenter and a Cluster, but the same is not available for a host. Therefore, we need to create that metric before we can use it. For a host system we have the following metrics available that are applicable to our cause:

- a. Number of physical CPUs (Cores)
- b. vCPUs allocated on all consumers.
- c. vCPUs allocated on all powered on consumers.

The same can be found by visiting: Environment>Metrics>CPU.

For our case we will consider the “Number of physical CPUs” and “vCPUs allocated on all powered-on consumers”. The over commitment ratio would be the division of these two metrics as can be read from the super metric configuration:



Once the metric is created, it can now be used in a dashboard or a view.

2. Create Dashboard Widget for Datacenter.

For the overcommitment ratio for a datacenter we can use the OOB metric “Current overcommitment ratio”. We are going to use a scoreboard type for this widget. This type of widget displays the information in tiles and can also colour code the information based on the symptom or custom input.

The screenshot shows the 'Dashboard Configuration' interface for the 'Datacenter Overcommitment ratio' widget. The left sidebar lists the widget name and two sub-items: 'Cluster CPU overcommitment ratio' and 'Host CPU overcommitment ratio'. The main configuration area is titled 'Configuration' and includes the following settings:

- Refresh Content: ☐ On, ☒ Off
- Refresh Interval: 300 (seconds)
- Self Provider: ☒ On, ☐ Off
- Round Decimals: --Select--
- Box Columns: 1
- Layout Mode: ☐ Fixed Size, ☒ Fixed View
- Old metric values: ☐ Show, ☒ Hide
- Visual Theme: Gradient
- Max Scores Count: 100
- Show: Object Name x Metric Unit x

A preview section at the bottom right shows a green tile with a checkmark icon, the value '65.86', the unit 'Unit', and the label 'Object Name'.

The datacenter widget is created as a self provider so it does not have to wait for input from anyone or anywhere else.

The screenshot shows the 'Input Data' configuration section for the 'Datacenter Overcommitment ratio' widget. The left sidebar is the same as in the previous screenshot. The main configuration area is titled 'Input Data' and includes the following settings:

- Metrics: ☐ Metrics, ☐ Objects, ☒ All

Below the radio buttons is a text input field labeled 'Object name'.

The input data for this “self provider” widget is going to be all types and objects available in the environment. We shape the output by using the “output data” and “output filter” option.

The screenshot shows the 'Output Data' configuration section for the 'Datacenter Overcommitment ratio' widget. The left sidebar is the same as in the previous screenshots. The main configuration area is titled 'Output Data' and includes the following settings:

- Mode: -- Default Mode --

Below the mode selector is a table with the following columns: Object Type, Metric, Box Label, Unit, Color Method, Yellow, Orange, Red, and Link to. The table contains one row with the following data:

| Object Type | Metric | Box Label | Unit | Color Method | Yellow | Orange | Red | Link to |
|-------------|------------------------------|-----------|------|--------------|--------|--------|-----|---------|
| Datacenter | CPU Current Overcommit Ratio | | :1 | Custom | 2.1 | 3 | 4 | |

We selected the object as “Datacenter” and then selected the required metric which is “Current overcommitment ratio”. We have defined our own custom colour coding by defining the values for Yellow, orange and red.

No output filters were required.

3. Create Dashboard Widget for Cluster.

Just like we configured the widget for Datacenter, we will create another scoreboard widget for the cluster. However, this widget is supposed to take the selected datacenter as input and then display the overcommitment ratios for the clusters of that datacenter only. Thus, this widget won't be a self provider widget.

Cluster CPU overcommitment ratio (Scoreboard)

Configuration

Refresh Content: ☐ On ☒ Off

Refresh Interval: 300 (seconds)

Self Provider: ☒ On ☐ Off

Round Decimals: --Select--

Box Columns: 1

Layout Mode: ☐ Fixed Size ☒ Fixed View

Old metric values: ☐ Show ☒ Hide

Visual Theme: Gradient

Max Scores Count: 100

Show: Object Name x Metric Unit x

Preview:

Since this is not a self provider widget, we don't need to configure the input data but we need to define how the input (coming from another widget, in this case the datacenter widget) is used, basically defining the scope with respect to the incoming data.

Cluster CPU overcommitment ratio (Scoreboard)

Configuration

Input Data

Input Transformation

Relationship: ☐ Parent ☒ Children ☐ Self

Depth: 2

Lastly we need to configure the output data, i.e. select the metric to be displayed by the scoreboard tiles.

Output Data

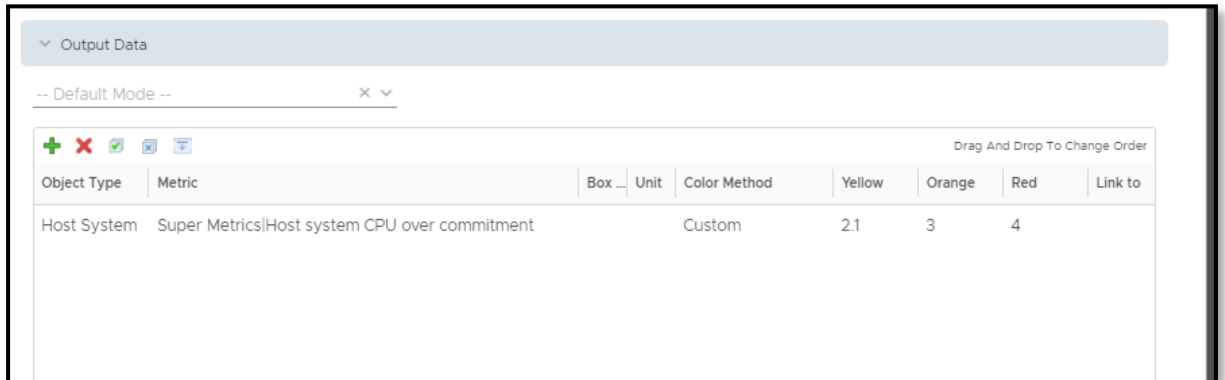
-- Default Mode --

Drag And Drop To Change Order

| Object Type | Metric | Box L | Unit | Color Method | Yellow | Orange | Red | Link to |
|--------------------------|------------------------------|-------|------|--------------|--------|--------|-----|---------|
| Cluster Compute Resource | CPU Current Overcommit Ratio | | .1 | Custom | 2.1 | 3 | 4 | |

4. Create Dashboard Widget for Host.

The widget for Host will be created the same as the widget for the cluster. The only difference being the metric. Since we don't have an OOB overcommitment metric for host and thus we created the super metric and we are going to use the super metric for this widget.

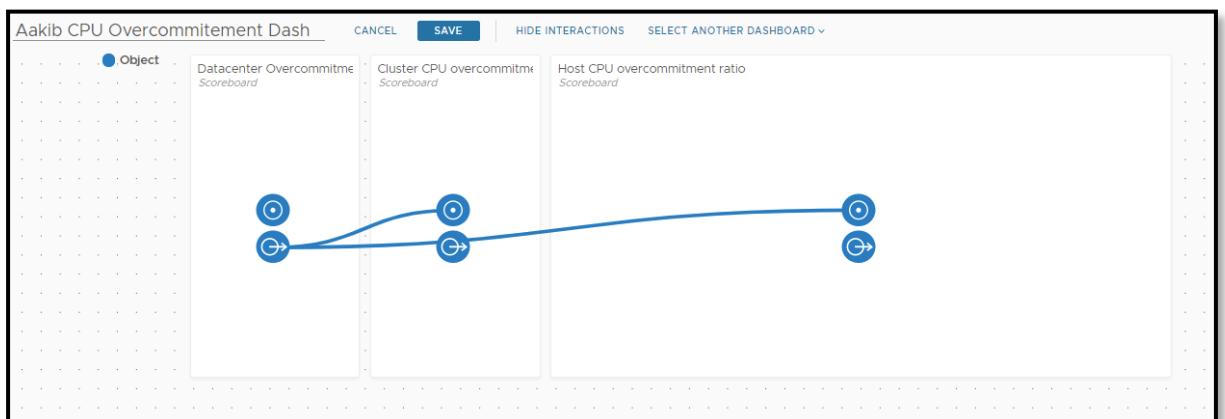


| Object Type | Metric | Box | Unit | Color Method | Yellow | Orange | Red | Link to |
|-------------|---|-----|------|--------------|--------|--------|-----|---------|
| Host System | Super Metrics Host system CPU over commitment | | | Custom | 2.1 | 3 | 4 | |

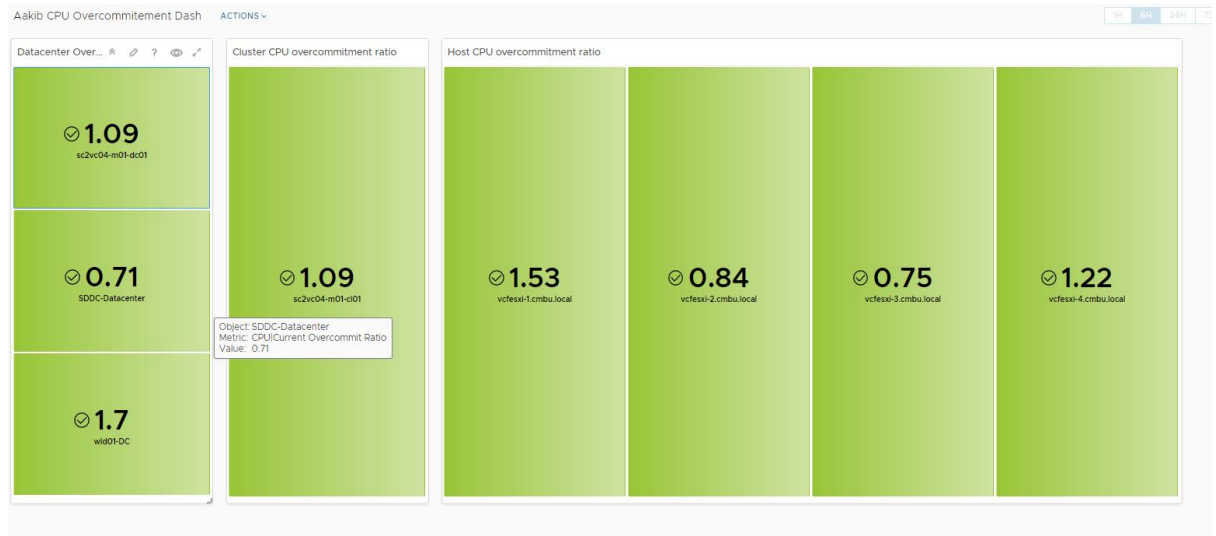
Rest everything would remain the same as the cluster widget.

5. Define Interactions

For our dashboard we have two widgets that are not self providers which means that they must get the input from another widget. The Datacenter widget acts as the main input for the other two dashboards. Note that we have not used the cluster widget as the provider for the host widget this is because the datacenter can have some hosts within a cluster and other hosts as standalone. If we choose the cluster widget as the provider for the host widget then we will not be able to consider the other hosts within the datacenter. Thus, when we click a tile on the datacenter widget, it serves as input for the cluster and host widget. The cluster and the host widget then automatically fetch the metric defined and display it as a tile (colour coded).



6. Result:



Similar dashboards can then be created for the memory overcommitment as well.