



# **Creating custom dashboards**

## **OnCommand Insight**

NetApp

September 19, 2024

# Table of Contents

- Creating custom dashboards ..... 1
  - Custom Dashboard concepts ..... 2
  - Custom Dashboard: Virtual Machine Performance ..... 14
  - Example storage node utilization dashboard with variables ..... 19
  - Best Practices for Dashboards and Widgets ..... 30

# Creating custom dashboards

OnCommand Insight 7.3 includes enhanced custom dashboard capabilities to give users an operational view of the data that is important to them and provide a single-stop view of that data.

OnCommand Insight provides users the flexibility to create operational views infrastructure data across IT platforms, by allowing you to create custom dashboards with a variety of widgets, each of which provides extensive flexibility in displaying and charting your data. In this How-To, we will create an example dashboard to highlight VM Performance.

This How-To should serve as an example only and does not cover every possible scenario. The concepts and steps herein can be used to create your own custom dashboards to highlight the data specific to your particular needs.

## Overview

You create a custom dashboard by either of the following methods:

- **Dashboards > +New dashboard**
- **Dashboards > Show all dashboards** and click **+Add**

The New Dashboard screen has several controls:

- **Time selector:** allows you to view dashboard data for a range of time from 3 hours up to 90 days by use of the custom date range selector. You can choose to override this global time range in individual widgets.
- **Edit button:** Selecting “On” will enable Edit mode, which allows you to make changes to the dashboard. New dashboards open in Edit mode by default.
- **Save button:** Allows you to save, rename or delete the dashboard.
- **Variable button:** Variables can be added to dashboards. Changing the variable updates all of your widgets at once. For more information on variables, see [Custom Dashboard concepts](#)
- **Widget button,** which allows you to add any number of tables, charts, or other widgets to the dashboard.

Widgets can be resized and relocated to different positions within the dashboard, to give you the best view of your data according to your current needs.

## Widget types

You can choose from the following widget types:

**Table** widget: A table displaying data according to filters and columns you choose. Table data can be combined in groups that can be collapsed and expanded.

**Line, Spline, Area, Stacked Area** charts: These are time-series chart widgets on which you can display performance and other data over time.

**Single value** widget: A widget allowing you to display a single value that can be derived either directly from a counter or calculated using a query or expression. For example, you can display the sum of total IOPS for all storage in your environment as a single value at the top of your dashboard.

**Bar** chart: A chart to display top or bottom 5, 10, 20, or 50 values.

**Box Plot** chart: A plot of the min, max, median, and the range between lower and upper quartile of data in a single chart.

**Scatter Plot** chart: Plots related data as points, for example, IOPS and latency. In this example, you would quickly see assets with high latency and low IOPS.

Additionally, there are a number of legacy widgets that you can choose. In the **Widgets** drop-down, select **Show more...** to see these widgets.

## Custom Dashboard concepts

Custom dashboards and widgets allow great flexibility in how data is displayed. Here are some concepts to help you get the most from your custom dashboards. Each concept is explained in greater detail in the following sections.

### Variables

Variables allow you to change the data displayed in some or all widgets on a dashboard at once. By setting each widget to use a common variable, changes made in one place cause the data displayed in each widget to update automatically.

### Multiple queries and/or expressions

Each time series widget (line, spline, area, or stacked area charts) can have up to five queries and/or expressions to determine what data to display, allowing you to compare different sets of data on a single chart. For example, you can have a line chart showing IOPS for both Storage and VM's, or a single chart comparing Throughput and Latency for all Storage Pools.

### Rollup and Grouping

Data displayed in each widget is rolled up from the data points collected. You can choose to roll up this data in one of several ways:

- Avg: rolls up data as the average of the underlying data
- Max: rolls up data to the maximum of the underlying data
- Min: rolls up data to the minimum of the underlying data
- Sum: rolls up data as the sum of the underlying data

By default, all of the underlying data is rolled up into a single entry (All) on the chart or table. You can choose to roll up data for a specific attribute instead, such as Data Center or Tier, to distribute the underlying data into desired groups. Your widget will display data for only the attributes you select.

You can group data in a table widget according to the attribute you choose. For example, you might choose to group your table by Data Center. Groups can be expanded or collapsed at will. Performance data in a table is rolled up in the group header according to the rollup method you set in the widget (average, max, min, or sum).

Table widgets can be sorted by any column, and columns can be moved or resized as needed.

### Top / Bottom

Use this to limit the result set in chart widgets, to select whether to display the top N results in your widget, or the bottom N results. You can choose this option when data is either not rolled up or is rolled up by a specific attribute.

## Override dashboard time

By default, most widgets you add to a dashboard show data according to the dashboard's time range setting (3h, 24h, 3d, 7d, 30d or custom range). However, you can override this time setting in individual widgets to force them to show data in a specific time context, regardless of the dashboard's time setting.

These concepts are explained in greater detail in the following section.

## Dashboard variables

Dashboard variables allow you to filter data across multiple widgets on a dashboard quickly and easily.

### Before you begin

This example requires the **City** annotation (also called City attribute) to be set on multiple storage assets.

For best results, set different cities on different storages.

### About this task

Variables provide a quick and simple way of filtering the data shown in some or all of the widgets on a custom dashboard. The following steps will guide you to creating widgets that use variables, and show you how to use them on your dashboard.

### Steps

1. Log in to Insight as a user with administrative permissions
2. Click on **Dashboards** > **+New Dashboard**.
3. Before adding widgets, we want to define the variables we will use to filter the dashboard data. Click on the **Variable** button.

The list of attributes is displayed.

4. Let's say we want to set the dashboard to filter based on City. Select the **City** attribute from the list.

The \$city variable field is created and added to the dashboard.

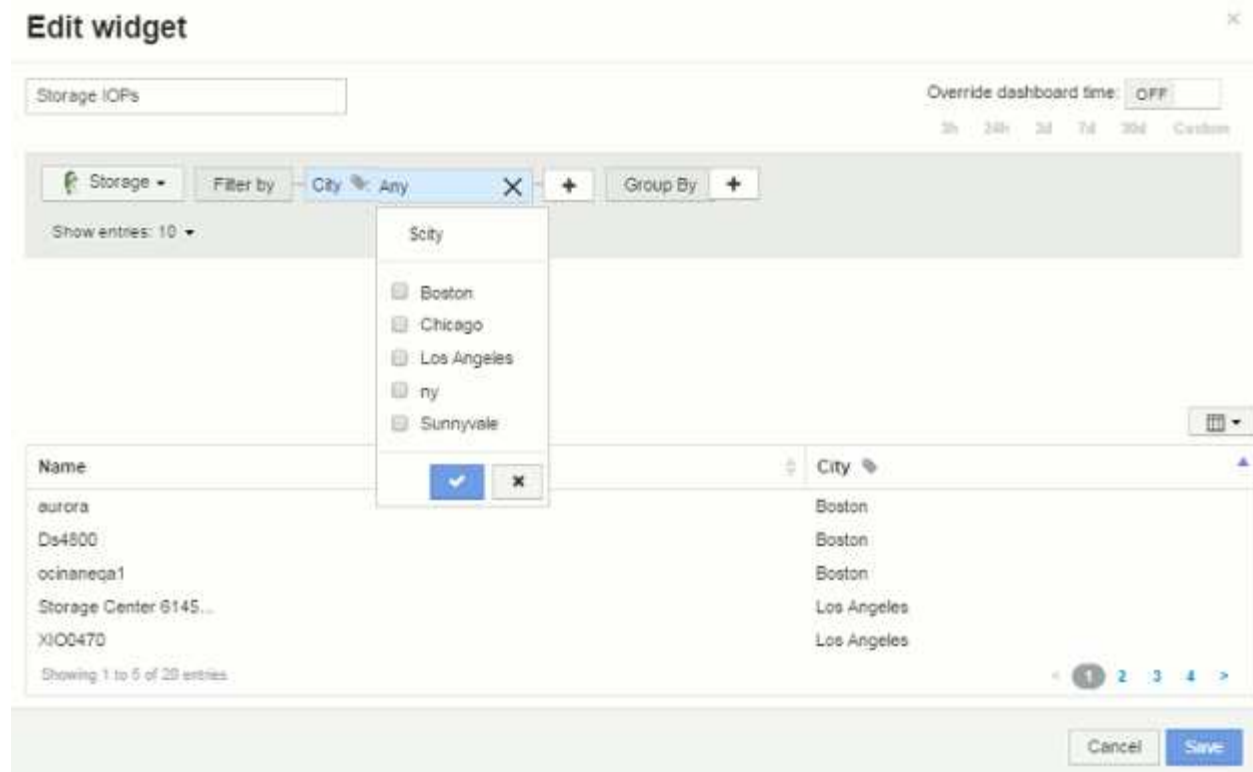
5. Next, we must tell our widgets to use this variable. The simplest way to illustrate this is to add a table widget showing the City column. Click on the **Widget** button and select the **Table** widget.

6. First, add the City field to the table by selecting it from the column picker  button.

City is a list-type attribute, so it contains a list of previously-defined choices. You may also choose text, boolean, or date-type attributes.

7. Next, click the **Filter by +** button and choose **City**.
8. Click **Any** to view the possible filter choices for City. Notice that the list now includes "\$city" at the top, in addition to any previously-available choices. Select "\$city" to use this dashboard variable.

The "\$city" choice only appears here if it was defined previously on the main dashboard page. If the variable was not previously defined, only the existing choices for the filter will be shown. Only variables that are applicable to the selected attribute type will be displayed in the drop-down for that filter.



9. **Save** the widget.
10. On the dashboard page, click on **Any** next to the \$city variable, and select the city or cities you want to see.

Your table widget updates to show only the cities you selected. You can change the values in the \$city variable at will, and all widgets on your dashboard that are set to use the \$city variable will refresh automatically to show only data for the values you selected.

11. Be sure to **Save** your dashboard when you have it configured as you want it.

### More on dashboard variables

Dashboard variables come in several types, can be used across different fields, and must follow rules for naming. These concepts are explained here.

#### Variable types

A variable can be one the following types:

**Text:** Alphanumeric string. This is the default variable type.

**Numerical:** a number or range of numbers.

**Boolean:** Use for fields with values of True/False, Yes/No, 0/1, etc. For the boolean variable, the choices are *Yes*, *No*, *None*, *Any*.

**Date:** A date or range of dates.

## “Generic” variables

You can set a generic or universal variable by clicking the **Variable** button and selecting one of the types listed above. These types are always shown at the top of the drop-down list. The variable is given a default name, for example “\$var1”, and is not tied to a specific annotation or attribute.

Configuring a generic variable allows you to use that variable in widgets to filter for *any* field of that type. For example, if you have a table widget showing *Name*, *Alias*, and *Vendor* (which are all text-type attributes), and “\$var1” is a text-type variable, you can set filters for each of those fields in the widget to use the \$var1 variable. You can set other widgets to use \$var1 for those or any text fields.

On your dashboard page, setting \$var1 to a value (for example “NetApp”) will filter *all* of those fields in *all* widgets that are set to use that variable. In this way, you can update multiple widgets at once to highlight dashboard data you choose at will.

Because generic variables can be used for any field of that type, you can change the name of a generic variable without changing its functionality.



All variables are treated as “generic” variables, even those you create for a specific attribute, because all configured variables of a type are shown when you set a filter for any attributes or annotations of that type. However, best practice is to create a generic variable when you will use it to filter for a value across multiple fields, as in the *Name/Alias/Vendor* example above.

## Variable naming

Variables names:

- Must always be prefixed with a “\$”. This is added automatically when you configure a variable.
- Cannot contain any special characters; only the letters a-z and the digits 0-9 are allowed.
- Cannot be longer than 20 characters, including the “\$” symbol.
- Are not case-sensitive: \$CityName and \$cityname are the same variable.
- Cannot be the same as an existing variable name.
- Cannot be only the “\$” symbol.

## Widgets that use variables

Variables can be used with the following widgets:

- Area Chart
- Bar Chart
- Box Plot Chart
- Line Chart
- Scatter Plot Chart
- Single Value Widget
- Spline Chart
- Stacked Area Chart
- Table Widget

## Displaying widget legends

Widgets in dashboards can be displayed with or without legends.

Legends in widgets can be turned on or off on a dashboard by either of two methods:

1. When creating or editing the widget itself, check the Legends checkbox and save the widget.
2. With the dashboard in Edit mode, click the Options button on the widget and check the Legends checkbox in the menu.

As you edit and change the data displayed in the widget, the legend for that widget is updated dynamically.

When legends are displayed, if the landing page of the asset indicated by the legend can be navigated to, the legend will display as a link to that asset page.

## Dashboard widget queries and filters

The Query in a dashboard widget is a powerful tool for managing the display of your data. Here are some things to note about widget queries.

Some widgets can have up to five queries. Each query will plot its own set of lines or graphs in the widget. Setting rollup, grouping, top/bottom results, etc. on one query does not affect any other queries for the widget.

You can click on the eye icon to temporarily hide a query. The widget display updates automatically when you hide or show a query. This allows you to check your displayed data for individual queries as you build your widget.

The following widget types can have multiple queries:

- Area chart
- Stacked area chart
- Line chart
- Spline chart
- Single value widget

The remaining widget types can have only a single query:

- Table
- Bar chart
- Box plot
- Scatter plot

## Filtering in dashboard queries

You can filter using any of the following to refine your search in any **text field** in the query:

- An asterisk enables you to search for everything. For example, `vol*rhel` displays all resources that start with “vol” and end with “rhel”.
- The question mark enables you to search for a specific number of characters. For example, `BOS-PRD??-s12` displays BOS-PRD12-S12, BOS-PRD13-S12, and so on.



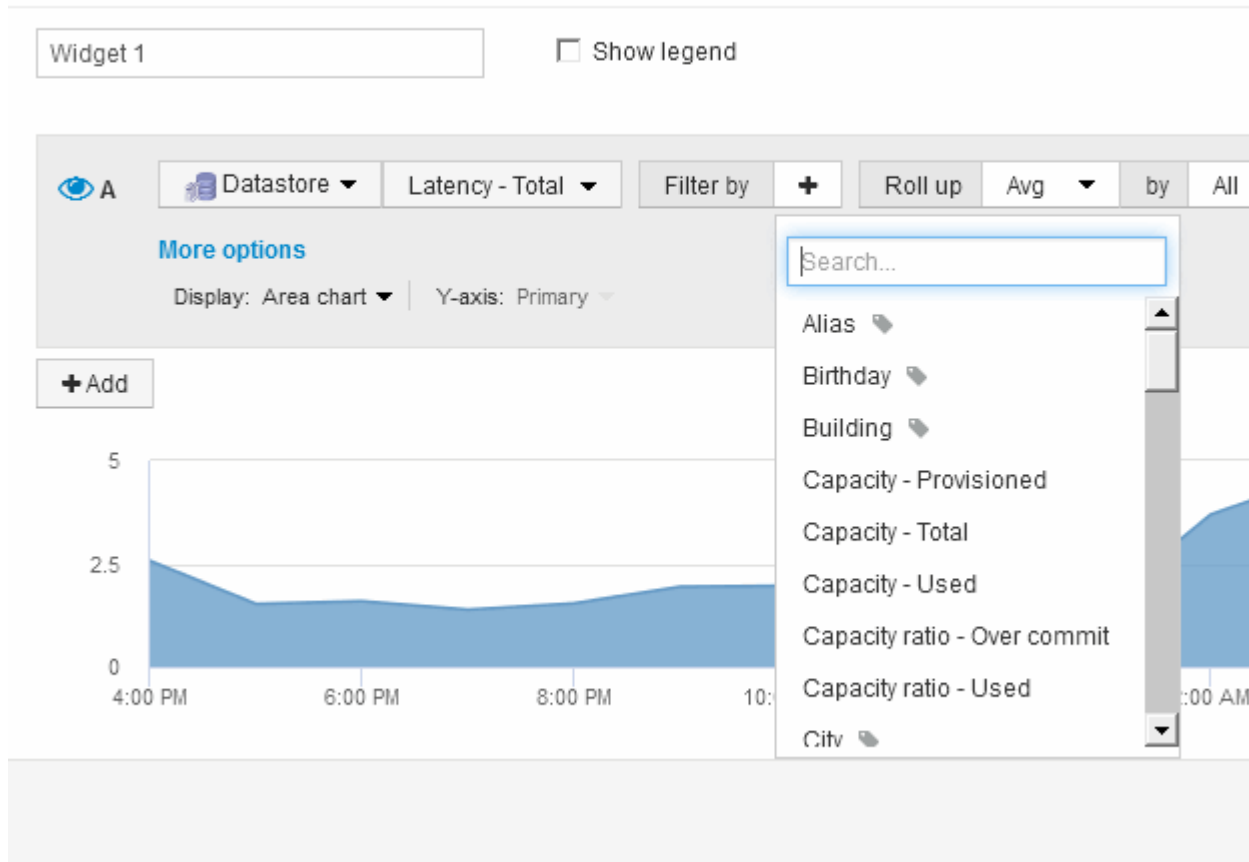
- The OR operator enables you to specify multiple entities. For example, `FAS2240 OR CX600 OR FAS3270` finds multiple storage models.
- The NOT operator allows you to exclude text from the search results. For example, `NOT EMC*` finds everything that does not start with “EMC”. You can use `NOT *` to display fields that contain null values.

If you enclose a filter string in double quotes, Insight treats everything between the first and last quote as an exact match. Any special characters or operators inside the quotes will be treated as literals. For example, filtering for “\*” will return results that are a literal asterisk; the asterisk will not be treated as a wildcard in this case. The operators AND, OR, and NOT will also be treated as literal strings when enclosed in double quotes.

## Identifying objects returned by queries and filters

The objects returned by queries and filters look similar to those shown in the following illustration. Objects with 'tags' assigned to them are annotations while the objects without tags are performance counters or object attributes.

## Edit widget



## Roll up and Aggregation

Data displayed in dashboard widgets is rolled up from acquired data points, allowing flexibility and conciseness in your dashboards.

Data displayed in each widget is rolled up from the underlying data points collected during acquisition. For example, if you have a line chart widget showing Storage IOPS over time, you might want to see a separate line for each of your data centers, for a quick comparison. You can choose to roll up this data in one of several

ways:

- **Avg**: displays each line as the *average* of the underlying data.
- **Max**: displays each line as the *maximum* of the underlying data.
- **Min**: displays each line as the *minimum* of the underlying data.
- **Sum**: displays each line as the *sum* of the underlying data.

To do this, in your widget's query, first choose an asset type (for example, *Storage*) and metric (such as *IOPS - Total*). For **Roll up**, choose a rollup method (such as *Avg*) and select an attribute or annotation by which to roll up the data (for example, *Data Center*). The widget updates automatically and shows a line for each of your data centers.

You can also choose to roll up *all* of the underlying data into the chart or table. In this case, you will get a single line for each query in the widget, which will show the average, min, max or sum of the chosen metric for all of the underlying assets.

If you have set a filter for the query, the data that is rolled up is based on the filtered data.

Note that when you choose to roll up a widget by any field (for example, *Model*), you will still need to **Filter by** that field in order to properly display the data for that field properly on the chart or table.

**Aggregating data:** You can further align your time-series charts (line, area, etc.) by aggregating data points into minute, hour, or day buckets before that data is subsequently rolled up by attribute (if chosen). You can choose to aggregate data points according to their Avg, Max, Min, or Sum, or by the Last data point collected during the chosen interval. To choose an aggregation method, click on **More options** in the widget's query section.

The minimum allowed interval is ten minutes. A small interval combined with a long time range may result in a "Aggregation interval resulted in too many data points." warning. You might see this if you have a small interval and increase the dashboard time frame to 7 days. In this case, Insight will temporarily increase the aggregation interval to 1 hour until you select a smaller time frame.

You can also aggregate data in bar chart widget and single-value widget.

Most asset counters aggregate to *Avg* by default. Some counters aggregate to *Max*, *Min*, or *Sum* by default. For example, port errors aggregate to *Sum* by default, where storage IOPS aggregate to *Avg*.

## Showing top/bottom results in dashboard widgets

In a chart widget on a custom dashboard, you can show either the Top or Bottom results for rolled up data, and choose the number of results shown. In a table widget, you can select the number of rows displayed and sort by any column.

### Chart widget top/bottom

In a chart widget, when you choose to rollup data by a specific attribute, you have the option of viewing either the top N or bottom N results. Note that you cannot choose the top or bottom results when you choose to rollup by *all* attributes.

You can choose which results to display by choosing either **Top** or **Bottom** in the query's **Show** field, and selecting a value from the list provided.

## Table widget show entries

In a table widget, you can select the number of results shown in the table results. You can choose from 5, 10, 20, or 50 results. You are not given the option to choose top or bottom results because the table allows you to sort ascending or descending by any column on demand.

You can choose the number of results to show in the table on the dashboard by selecting a value from the query's **Show entries** field.

Note that the more results you choose to display, the taller your widget will be when you save it to the dashboard. You will not be able to resize the widget smaller than the number of rows displayed.

## Grouping in table widgets

Data in a table widget can be grouped by any available attribute, allowing you to see an overview of your data, and to drill-down into it for more detail. Metrics in the table are rolled up for easy viewing in each collapsed row.

Table widgets allow you to group your data based on the attributes you set. For example, you might want your table to show Total Storage IOPS grouped by the data centers in which those storages live. Or you might want to display a table of Virtual machines grouped according to the hypervisor that hosts them. From the list, you can expand each group to view the assets in that group.

Grouping is only available in the **Table** widget type.

## Performance data roll up

If you include a column for performance data (for example, *IOPS - Total*) in a table widget, when you choose to group the data you can then choose a roll up method for that column. The default roll up method is to display the *average* of the underlying data in the group row. You can also choose to display the *sum*, *minimum*, or *maximum* of the data..


## Grouping example (with rollup explained)

Table widgets allow you to group data for easier display.

### About this task

In this example, we will create a table widget showing all VMs grouped by Data Center.

### Steps

1. Create or open a dashboard, and add a **Table** widget.
2. Select **Virtual Machine** as the asset type for this widget.
3. Click on the Column Selector  and choose *Hypervisor name* and *IOPS - Total*.

Those columns are now displayed in the table.

4. Let's disregard any VM's with no IOPS, and include only VMs that have total IOPS greater than 1. Click the **Filter by +** button and select **IOPS - Total**. Click on **Any**, and in the **from** field, type 1. Leave the **to** field empty. Click the check button to apply the filter.

The table now shows all VMs with Total IOPS greater than or equal to 1. Notice that there is no grouping in

the table. All VMs are shown.

5. Click the **Group by +** button.

Because **All** is selected as the grouping method by default, all VMs are moved into a single group named “All”.

6. Above the *IOPS - Total* column is now a **Roll up** option. The default roll up method is *Avg*. This means that the number shown for the group is the average of all the Total IOPS reported for each VM inside the group. You can choose to roll this column up by *Avg*, *Sum*, *Min* or *Max*. Each column that you display that contains performance metrics can be rolled up individually.
7. Click **All** and select **Hypervisor name**.

The VM list is now grouped by Hypervisor. You can expand each hypervisor to view the VMs hosted by it.

**Edit widget**

Table - Grouping Example

Override dashboard time: OFF

Virtual Machine Filter by: IOPS - Total (IOPS) >= 5 Group By: Hypervisor name

Show entries: 5

		Roll up: Avg	
<input type="checkbox"/> Hypervisor name	Name	Hypervisor name	IOPS - Total (IOPS)
• hv-72-001.nane.neta... (3)		hv-72-001.nane.neta...	8.68
• hv-72-002.nane.neta... (4)		hv-72-002.nane.neta...	12.34
	vsa-5-vo	hv-72-002.nane.neta...	14.77
	ns5	hv-72-002.nane.neta...	7.01
	ns6	hv-72-002.nane.neta...	6.94

37 items found in 35 groups

Cancel Save

8. Click **Save** to save the table to the dashboard. You can resize the widget.
9. Click **Save** to save the dashboard.

## Overriding dashboard time for individual widgets

You can override the main dashboard time frame setting in individual widgets. These widgets will display data based on their set time frame, not the dashboard time frame.

To override the dashboard time and force a widget to use a its own time frame, in the widget's edit mode set the **Override dashboard time** to **On**, and select a time frame for the widget. **Save** the widget to the dashboard.

The widget will display its data according to the time frame set, regardless of the time frame you select on the dashboard itself.

The time frame you set for one widget will not affect any other widgets on the dashboard.

## Primary and Secondary axis explained

The secondary axis makes it easier to view data from two different sets of values that use different units of measurement.

### About this task

Different metrics use different units of measurements for the data they report in a chart. For example, when looking at IOPS, the unit of measurement is the number of I/O operations per second of time (IO/s), while Latency is purely a measure of time (milliseconds, microseconds, seconds, etc.). When charting both metrics on a single line chart using a single set of values for the Y-Axis, the latency numbers (typically a handful of milliseconds) are charted on the same scale with the IOPS (typically numbering in the thousands), and the latency line gets lost at that scale.

But it is possible to chart both sets of data on a single meaningful graph, by setting one unit of measurement on the primary (left-side) Y-axis, and the other unit of measurement on the secondary (right-side) Y-axis. Each metric is charted at its own scale.

### Steps

1. Create or open a dashboard. Add a **line chart**, **spline chart**, **area chart** or **stacked area chart** widget to the dashboard.
2. Select an asset type (for example **Storage**) and choose **IOPS - Total** for your first metric. Set any filters you like, and choose a roll-up method if desired.

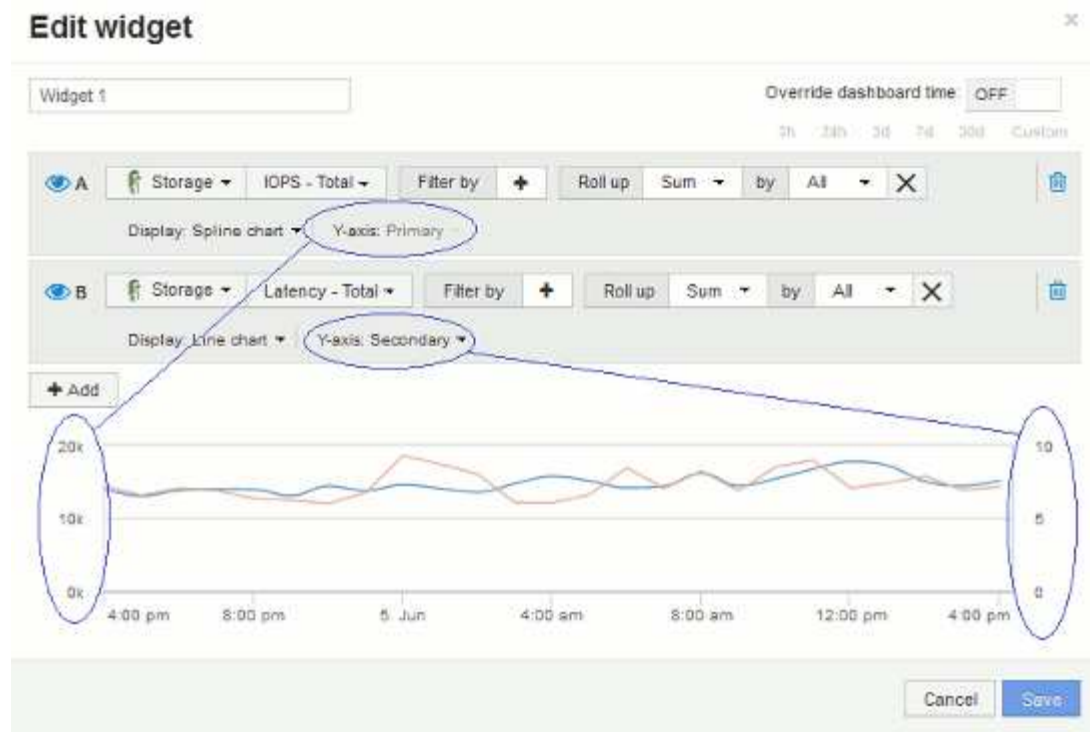
The IOPS line is displayed on the chart, with its scale shown on the left.

3. Click **+Add** to add a second line to the chart. For this line, choose **Latency - Total** for the metric.

Notice that the line is displayed flat at the bottom of the chart. This is because it is being drawn at the same scale as the IOPS line.

4. In the Latency query, select **Y-Axis: Secondary**.

The Latency line is now drawn at its own scale, which is displayed on the right side of the chart.



## Expressions in dashboard widgets

Expressions in time series widgets allow you to display data based on calculations with metrics of your choosing.

In a dashboard, any time series widget (line, spline, area, stacked area) allows you to build expressions from metrics you choose, and show the result of those expressions in a single graph. The following examples use expressions to solve specific problems. In the first example, we want to show Read IOPS as a percentage of Total IOPS for all storage assets in our environment. The second example gives us visibility into the "system" or "overhead" IOPS that occur in our environment—those IOPS that are not from reading or writing data.

### Expressions example: Read IOPS percentage

Using expressions, you can view metrics by alternate means, such as percentage of total.

#### About this task

In this example we want to show Read IOPS as a percentage of Total IOPS. You can think of this as the following formula:

- Read Percentage = (Read IOPS / Total IOPS) x 100

This data can be shown in a line graph on your dashboard. To do this, follow these steps:

#### Steps

1. Create a new dashboard, or open an existing dashboard in **edit mode**.
2. Add a widget to the dashboard. Choose **Area chart**.

The widget opens in edit mode. By default, a query is displayed showing **IOPS - Total** for **Storage** assets.

If desired, select a different asset type.

3. Click the **Convert to Expression** button.

The current query is converted to Expression mode. Notice that you cannot change the asset type while in Expression mode. Notice that while you are in Expression mode, the button changes to **Revert to Query**. Click this if you wish to switch back to Query mode at any time. Be aware that switching between modes will reset fields to their defaults.

For now, stay in **Expression** mode.

4. The **IOPS - Total** metric is now in the alphabetic variable field "a". In the "b" variable field, click **Select** and choose **IOPS - Read**.

You can add up to a total of five alphabetic variables for your expression by clicking the **+** button following the variable fields. For our Read Percentage example, we only need Total IOPS ("a") and Read IOPS ("b").

5. In the **Expression** field, you use the letters corresponding to each variable to build your expression. We know that *Read Percentage = (Read IOPS / Total IOPS) x 100*, so we would write this expression as: (b / a) \* 100
6. The **Label** field identifies the expression. Change the label to "Read Percentage", or something equally meaningful for you.
7. Change the **Units** field to "%" or "Percent".

The chart displays the IOPS Read percentage over time for the chosen storage devices. If desired, you can set a filter, or choose a different rollup method. Be aware that if you select **Sum** as the rollup method, all percentage values are added together, which potentially may go higher than 100%.

8. Click **Save** to save the chart to your dashboard.

You can also use expressions in **Line chart**, **Spline chart**, or **Stacked Area chart** widgets.

### Expressions example: "System" I/O

Expressions give you the freedom to chart data that can be calculated from other metrics.

#### About this task

Example 2: OnCommand Insight acquires many metrics from data sources. Among them are read, write, and total IOPS. However, the total number of IOPS reported by acquisition sometimes includes "system" IOPS, which are those IO operations that are not a direct part of data reading or writing. This system I/O can also be thought of as "overhead" I/O, necessary for proper system operation but not directly related to data operations.

To show these system I/Os, you can subtract read and write IOPS from the total IOPS reported from acquisition. The formula might look like this:

- System IOPS = Total IOPS - (Read IOPS + Write IOPS)

This data can then be shown in a line graph on your dashboard. To do this, follow these steps:

#### Steps

1. Create a new dashboard, or open an existing dashboard in **edit mode**.

2. Add a widget to the dashboard. Choose **Line chart**.

The widget opens in edit mode. By default, a query is displayed showing **IOPS - Total** for **Storage** assets. If desired, select a different asset type.

3. Click the button to create a copy of the query.

A duplicate of the query is added below the original.

4. In the second query, click the **Convert to Expression** button.

The current query is converted to Expression mode. Click **Revert to Query** if you wish to switch back to Query mode at any time. Be aware that switching between modes will reset fields to their defaults.

For now, stay in **Expression** mode.

5. The **IOPS - Total** metric is now in the alphabetic variable field “a”. Click on **IOPS - Total** and change it to **IOPS - Read**.
6. In the “b” variable field, click **Select** and choose **IOPS - Write**.
7. In the **Expression** field, you use the letters corresponding to each variable to build your expression. We would write our expression simply as:  $a + b$ . In the **Display** section, choose **Area chart** for this expression.
8. The **Label** field identifies the expression. Change the label to “System IOPS”, or something equally meaningful for you.

The chart displays the total IOPS as a line chart, with an area chart showing the combination of read and write IOPS below that. The gap between the two shows the IOPS that are not directly related to data read or write operations.

9. Click **Save** to save the chart to your dashboard.

## Custom Dashboard: Virtual Machine Performance

OnCommand Insight’s custom dashboards and widgets help provide operational views into inventory and performance trends.

### About this task

There are many challenges facing IT operations today. Administrators are being asked to do more with less, and having full visibility into your dynamic data centers is a must. In this example, we will show you how to create a custom dashboard with widgets that give you operational insights into the virtual machine performance in your environment. By following this example, and creating widgets to target your own specific needs, you will be able to visualize backend storage performance compared to frontend virtual machine (VM) performance, or view VM latency versus I/O demand.

Custom dashboards allow you to prioritize efforts and identify resource availability. You can respond to the ebb and flow of workloads and minimize the time to detect and remediate emerging issues. Custom dashboards allow you the flexibility to create prioritized views into business-critical infrastructure, and are useful for identifying performance availability across multi-vendor technologies.

Here we will create a Virtual Machine Performance dashboard containing the following:



- a table listing VM names and performance data
- a chart comparing VM Latency to Storage Latency
- a chart showing Read, Write and Total IOPS for VMs
- a chart showing Max Throughput for your VMs

This is just a basic example. You can customize your dashboard to highlight and compare any performance data you choose to target for your own operational best practices.

## Steps

1. Log in to Insight as a user with administrative permissions.
2. From the **Dashboards** menu, select **+New dashboard**.

The New dashboard page opens.

3. Let's give our dashboard a meaningful name. Click **Save**. In the **Name** field, enter a unique name for the dashboard, for example "VM Performance by Application".
4. Click **Save** to save the dashboard with the new name.
5. Let's start adding our widgets. If necessary, slide the **Edit** switch to "On" to enable Edit mode.
6. Click the **Widget** button and select **Table widget** to add a new table widget to the dashboard.

The Edit Widget dialog opens. The default name is "Widget 1" and the default data displayed is for all storages in your environment.

**Edit widget**

Widget 5

Override dashboard time: ☐ OFF

3h 24h 3d 7d 30d Custom

Storage Filter by + Group By +

Show entries: 5


Name	Vendor
3070-a,3070-b	NetApp
APM000934007420000	EMC
Ds4800	NetApp
FNM00142500950	EMC
Storage Center 6145...	Dell

18 items found


Cancel Save

7. We can customize this widget. In the Name field, delete "Widget 1" and enter "Virtual Machine Performance table".
8. Click the asset type drop-down and change **Storage** to **Virtual Machine**.

The table data changes to show all virtual machines in your environment. For now, the table only shows the VM names. Let's add a few columns to the table.

9. Click the \*Columns\*  button and select *Data Center*, *Storage name*, and *IOPS - Total*. You can also try typing the name into the search to quickly display the desired field(s).

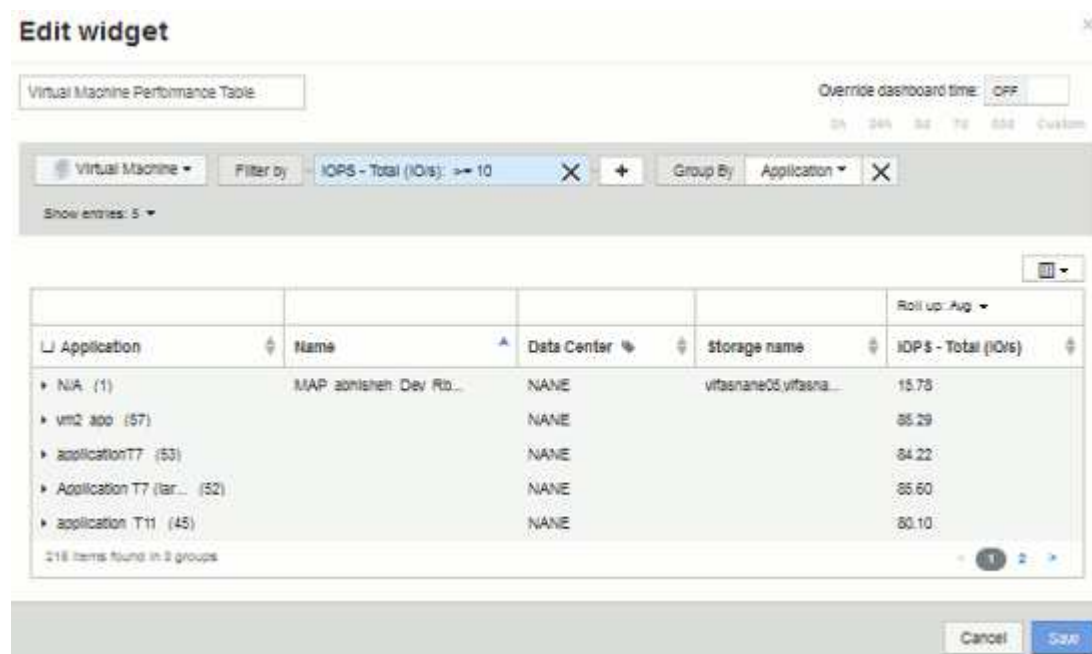
These columns are now displayed in the table. You can sort the table by any of these columns. Note that the columns are displayed in the order in which they were added to the widget.

10. For this exercise we will exclude VMs that are not actively in use, so let's filter out anything with less than 10 total IOPS. Click the "+" button next to **Filter by** and select *IOPS - Total (IO/s)*. Click on **Any** and enter "10" in the **from** field. Leave the **to** field empty. Click the  button to save the filter.

The table now shows only VMs with 10 or more total IOPS.

11. We can further collapse the table by grouping results. Click the "+" button next to **Group by** and select a field to group by, such as *Application* or *Cluster*. Grouping is automatically applied.

The table rows are now grouped according to your setting. You can expand and collapse the groups as needed. Grouped rows show rolled up data for each of the columns. Some columns allow you to choose the roll up method for that column.



Application	Name	Data Center	Storage name	IOPS - Total (IO/s)
N/A (1)	MAP admin Dev Rb...	N/A	vfasname05.vfasna...	15.78
vm2 app (57)		N/A		85.29
applicationT7 (53)		N/A		84.22
Application T7 (lar... (52)		N/A		85.60
application T11 (45)		N/A		80.10

12. When you have customized the table widget to your satisfaction, click the **Save** button.

The table widget is saved to the dashboard.

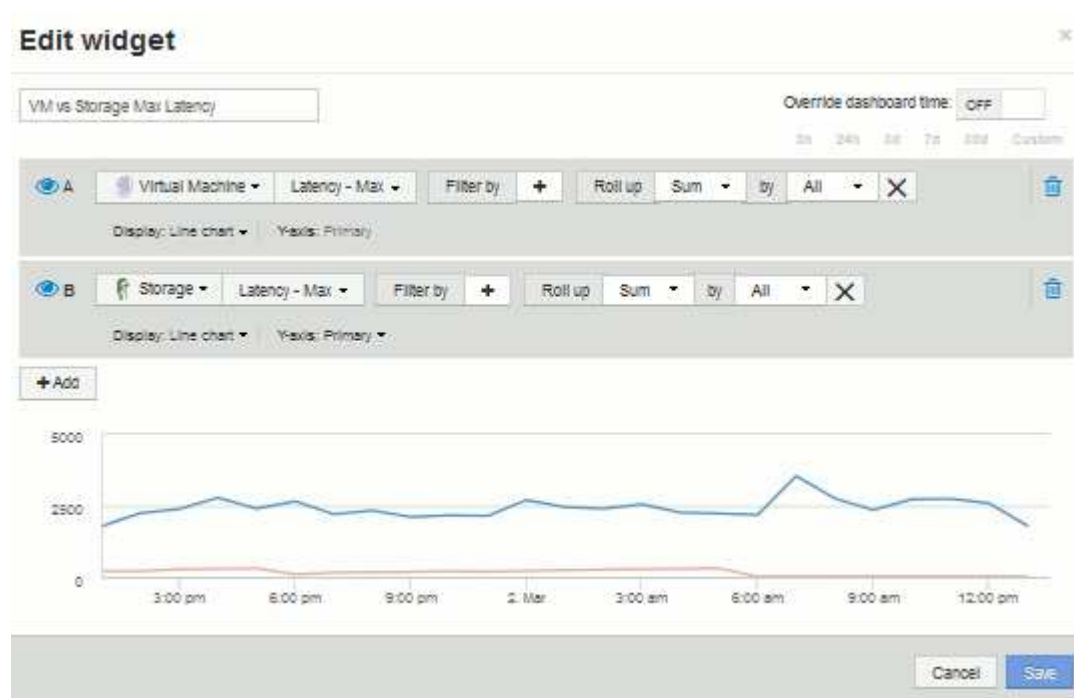
13. You can resize the widget on the dashboard by dragging the lower-right corner. Make the widget wider to show all the columns clearly. Click **Save** to save the current dashboard.
14. Next we will add some charts to show our VM Performance. Let's create a line chart comparing VM latency with Storage latency.
15. If necessary, slide the **Edit** switch to "On" to enable Edit mode.
16. Click the **Widget** button and select **Line Chart** to add a new line chart widget to the dashboard.

The Edit Widget dialog opens. Click the **Name** field and name this widget "VM vs Storage Max Latency"

17. Select **Virtual Machine** and choose **Latency - Max**. Set any filters you wish, or leave **Filter by** empty. For

**Roll up**, choose “Sum” by “All”. Display this data as a**Line Chart**, and leave Y-Axis as **Primary**.

18. Click the **+Add** button to add a second data line. For this line, select **Storage** and **Latency - Max**. Set any filters you wish, or leave **Filter by** empty. For **Roll up**, choose “Sum” by “All”. Display this data as a**Line Chart**, and leave Y-Axis as **Primary**.

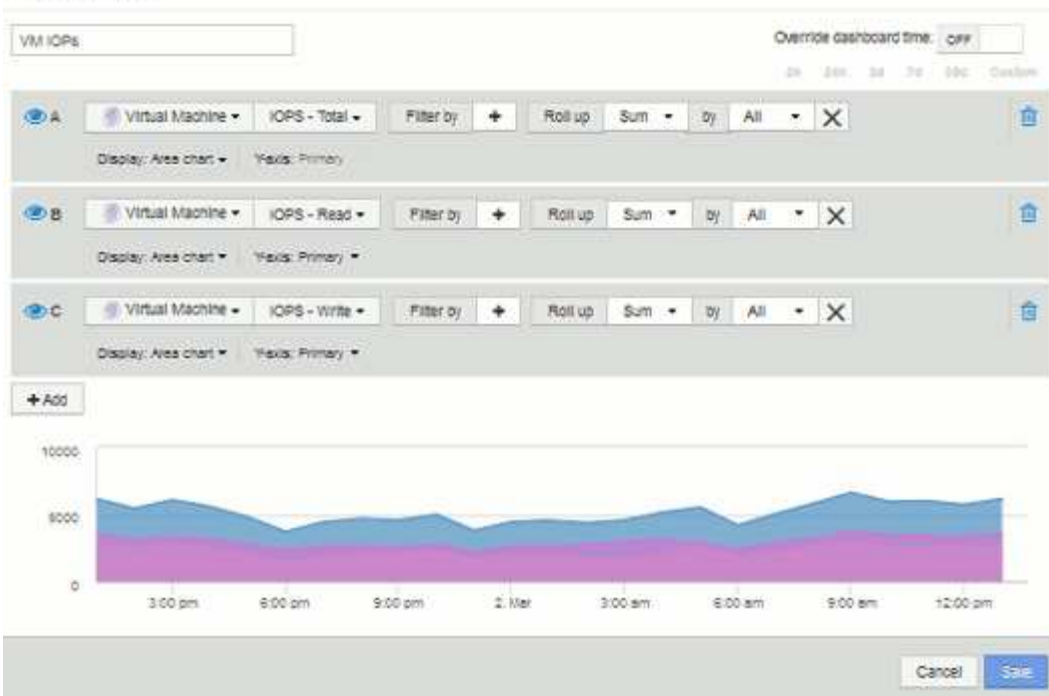


19. Click **Save** to add this widget to the dashboard.
20. Next we will add a chart showing VM Read, Write and Total IOPS in a single chart.
21. Click the **Widget** button and select **Area Chart** to add a new area chart widget to the dashboard.

The Edit Widget dialog opens. Click the **Name** field and name this widget “VM IOPS”

22. Select **Virtual Machine** and choose **IOPS - Total**. Set any filters you wish, or leave **Filter by** empty. for **Roll up**, choose “Sum” by “All”. Display this data as a**Area Chart**, and leave Y-Axis as **Primary**.
23. Click the +Add button to add a second data line. For this line, select **Virtual Machine** and choose **IOPS - Read**. Leave Y-Axis as **Primary**.
24. Click the +Add button to add a third data line. For this line, select **Virtual Machine** and choose **IOPS - Write**. Leave Y-Axis as **Primary**.

## Edit widget



25. Click **Save** to add this widget to the dashboard.

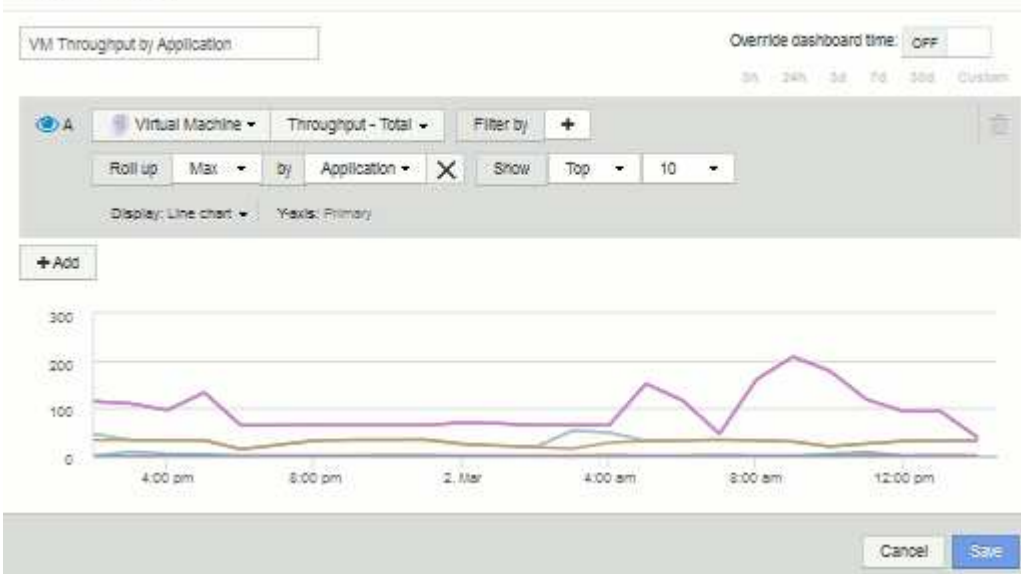
26. Next we will add a chart showing VM Throughput for each Application associated with the VM. We will use the Roll Up feature for this.

27. Click the **Widget** button and select **Line Chart** to add a new line chart widget to the dashboard.

The Edit Widget dialog opens. Click the **Name** field and name this widget “VM Throughput by Application”

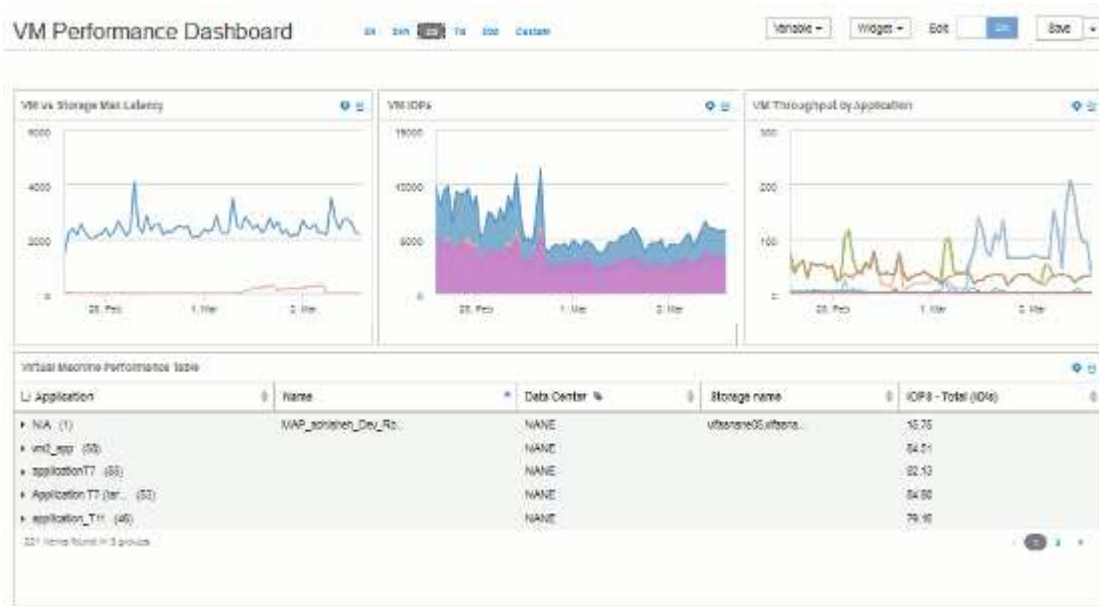
28. Select **Virtual Machine** and choose **Throughput - Total**. Set any filters you wish, or leave **Filter by** empty. For **Roll up**, choose “Max” and select by “Application” or “Name”. Show the **Top 10** applications. Display this data as a **Line Chart**, and leave Y-Axis as **Primary**.

## Edit widget



29. Click **Save** to add this widget to the dashboard.
30. You can move widgets by holding down the mouse button anywhere in the top of the widget and dragging to a new location. You can resize widgets by dragging the lower-right corner. Be sure to **Save** the dashboard after you make your changes.

Your final VM Performance Dashboard will look like this:



## Example storage node utilization dashboard with variables

Create a custom dashboard for Storage Analysis which has variables for storage, storage pool, node, tier, utilization and latency.

### Before you begin

Familiarity with dashboards in Insight is recommended but not required.

### About this task

The following procedure will create a custom Storage Analysis Overview dashboard which uses variables for storage, storage pool, node, tier, utilization and latency. Variables in the example below will be used to filter the displayed assets or metrics across one or many widgets available on the dashboard. The widgets that use these variables as filters will be updated with filtered content on-demand according to the values entered in the variable fields on the dashboard, allowing you to quickly filter multiple charts and graphs to drill down to a specific area of interest.

By following the steps in this example, you will create a dashboard like the following. You can change these widgets, or add any number of additional widgets, to highlight any data you choose.



## Steps

1. Create a new dashboard, and name it "Analysis: Storage Overview", or something equally descriptive.
2. Click on the **Variable** drop-down and select **Text** variable type. By default, the variable is named `$var1`. Click on `$var1` to edit the name, and change it to `$storage`, then click the check mark to save the variable. Repeat to create text variables for `$node`, `$pool`, and `$volume`.
3. Repeat the above process to create **Number**-type variables named `$utilization` and `$latency`.
4. Click on the **Variable** drop-down and search for the `Tier` annotation. Select it to create a variable named `$tier`.

You can add variables at any time, however it is easier to create them up front and therefore make them available to all widgets as you create them.

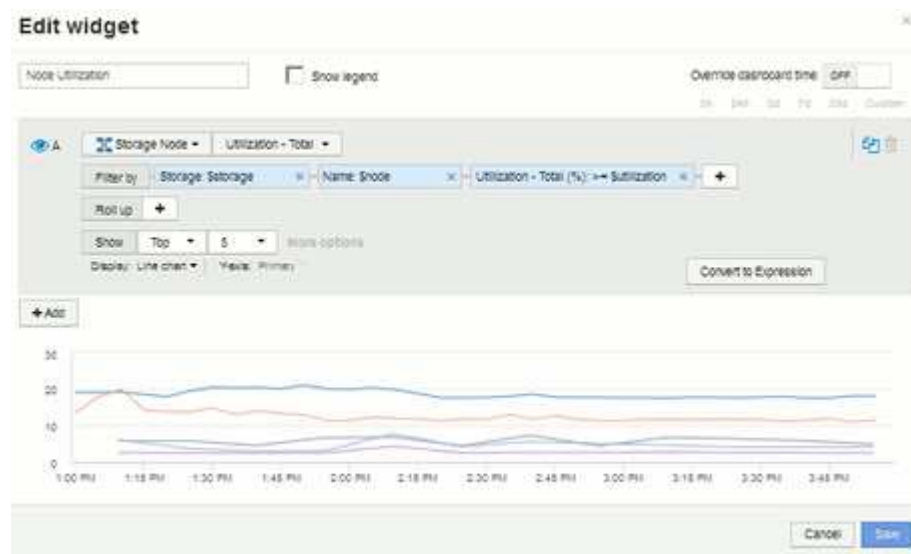


5. Add a widget by clicking on the **Widget** drop-down and selecting either a **line chart** or **area chart** widget. Name the widget “Node Utilization”. Click on the **Storage** asset type and change it to **Storage Node**. Select **Utilization - Total** for the chart data.
6. Click on the **Filter by +** button to add a filter. Search for and select **Storage**, then click on **Any** and select the *\$storage* variable.
7. Click the **+** button to add another filter for **Name**. Set the variable to *\$node*.

Different variables can be assigned to the annotation name filter. Use the name/variable pair at the lowest level depending on the object in the widget. For example:

- You can assign the *\$node* variable to the **Name** filter for a Node-focused widget.
  - You can assign the *\$pool* variable to the **Name** filter for a Pool-focused widget.
8. Click the **+** button to add another filter for **Utilization - Total (%)**. Set the variable to *>= \$utilization*.
  9. Click the **X** after the **Roll up** field to collapse the field.
  10. Select **Show Top 5** and click **Save** to save the widget and return to your Dashboard.

Your widget should look something like this:



11. Add another line or area chart widget to your dashboard. Select **Storage Node** as the asset type and **Latency - Total** as the metric to chart.
12. Click on the **Filter by +** button to add filters for **Storage: \$storage** and **Name: \$node**.
13. Add a filter for **Latency - Total** and select the *\$latency* variable.
14. Name the widget “Node Latency” and save it.
15. You can add supporting tables to show more details for the charts you created, for example, Max or Avg node utilization. Add a **Table widget** to the dashboard and select **Storage Node** as the asset type, and create filters for **Storage: \$storage**, **Name: \$node**, and **Utilization - Total: \$utilization**.
16. Add columns to the table for **Utilization - Max**, **Utilization - Total**, or any other desired columns.
17. Name the widget “Node Peak and Avg Utilization” and save it.







- You can use the variables to focus on specific assets in your dashboard. As you enter values into the variable fields, your widgets update automatically to reflect those variables. For example, by entering "15" in the \$utilization variable field, the widgets using that variable update to show only assets with total utilization  $\geq 15\%$ .

Node utilization widget showing top 5 of all nodes:



Node utilization widget showing nodes with 15% or greater utilization:



3. Keep in mind the following when creating your widgets:

- The **\$tier** variable will only impact resources that are annotated with the **Tier** annotation.
- Not all filters will impact all widgets, depending on whether the widget is designed to accept the variable(s) specified.
- Number variables are applied as “greater than or equal to” the value specified. Note that any variable can be used as a filter on any widget at any level in a storage hierarchy, as long as the variable is valid for the asset against which the widget is running. As you move down from a Node level to Storage Pool to a Volume widget, more variables are present for assignment as filters. For example, at a Storage Node level widget, the *Storage* and *Name* variables can be assigned as filters. At a Storage Pool level, *Storage*, *Nodes*, *Storage Pools* and *Name* are all available. Assign your variables as appropriate and use the *\$name* variable at the lowest level in the stack. Doing this will allow your *\$name* variable to filter on the actual name of the asset against which the widget is running.

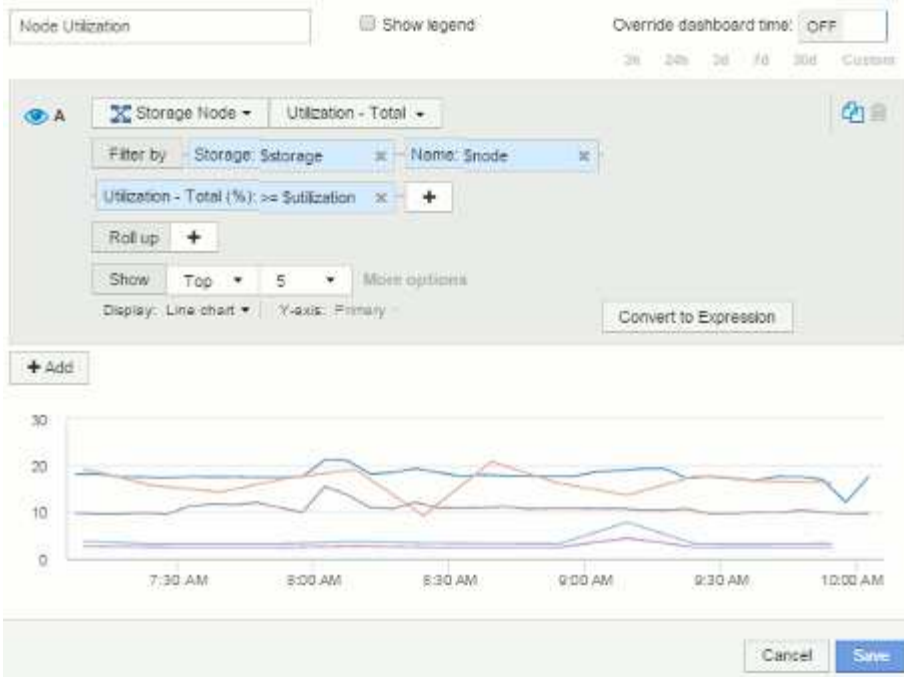
## Node dashboard example widget settings

Widget settings for the node dashboard with variables example.

Following are the settings for each of the widgets in the storage node dashboard example.

Node utilization:

## Edit widget



## Edit widget

Node Peak and Avg Utilization

Override dashboard time: OFF

3h 24h 3d 7d 30d Custom

Storage Node

Filter by Storage: \$storage Name: \$node Utilization - Total (%): >= \$utilization

Group by

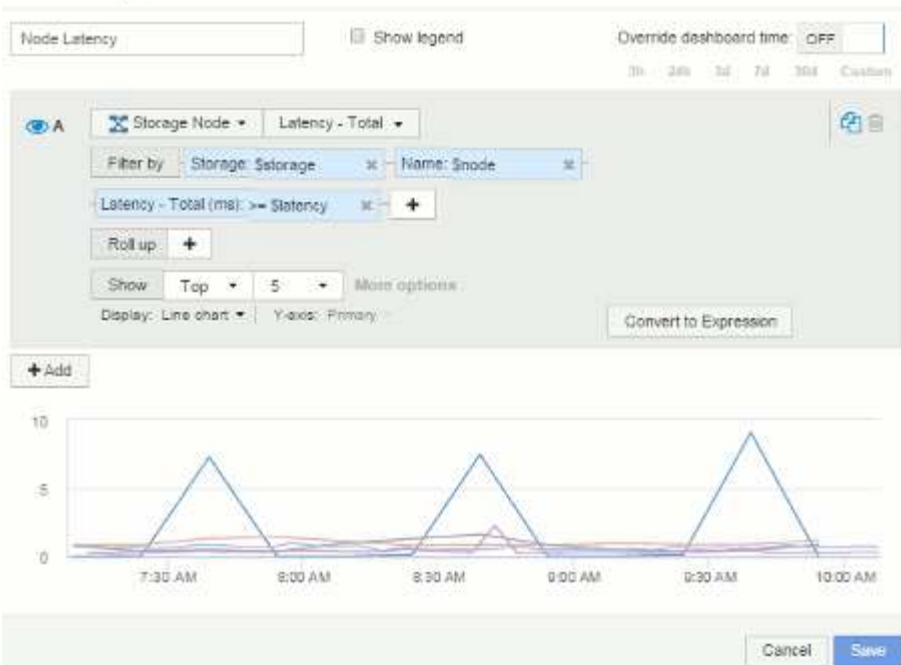
Name	Utilization - Max (%)	Utilization - Total (%)
3070-a	76.79	21.57
3070-b	76.79	21.57
vifasane01	54.83	18.55
vifasane02	32.50	6.06
aurora3	29.27	12.88

53 items found

Cancel Save

Node latency:

## Edit widget



## Edit widget

Node Peak and Avg Latency

Override dashboard time: OFF

3h 30m 3d 7d 30d Custom

Storage Node

Filter by: Storage: \$storage Name: \$node Latency - Total (ms) >= \$latency

Group by: +

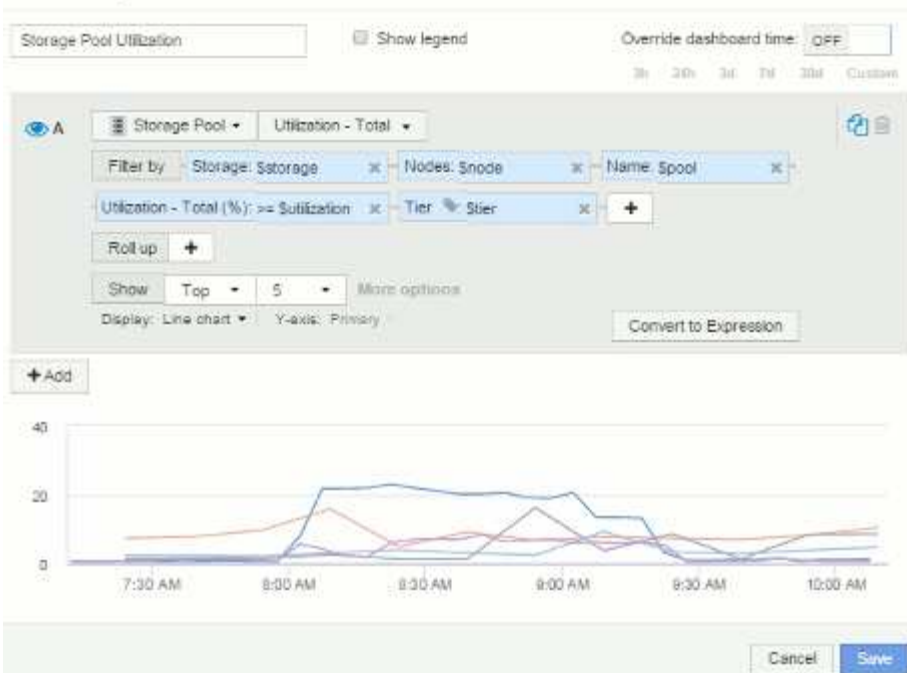
Name	Latency - Max (ms)	Latency - Total (ms)
vfasname04	9.05	7.70
vfasname05	2.25	0.41
vfasname02	1.62	0.90
vfasname01	1.42	1.03
vfasname06	0.97	0.64

8 items found

Cancel Save

Storage pool utilization:

## Edit widget



## Edit widget

Storage Pool Peak and Avg Utilization

Override dashboard time:

3h 24h 3d 7d 30d Custom

Storage Pool

Filter by: Storage: \$storage x Nodes: \$node x Name: \$pool x

Utilization - Total (%) >= Utilization x Tier: \$tier x +

Group by: +

Table view

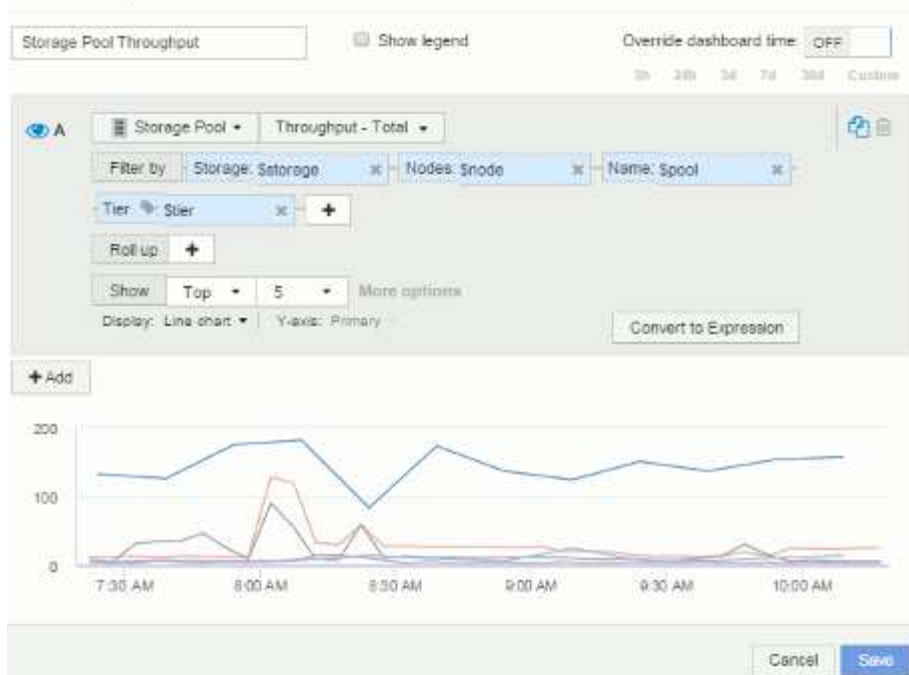
Name	Utilization - Max (%)	Utilization - Total (%)
vfasname01:aggr1	15.85	8.52
vfasname01:vfasna...	16.19	4.71
vfasname02:aggr2	9.28	3.65
vfasname02:vfasna...	4.96	1.63
vfasname03:aggr3	1.04	0.68

14 items found

Cancel Save

Storage pool throughput:

## Edit widget



## Edit widget

Storage Pool Peak and Avg Throughput

Override dashboard time: OFF

3h 24h 3d 7d 30d Custom

Storage Pool

Filter by: Storage: \$storage Nodes: \$node Name: \$pool

Tier: \$tier

Group by: +

Name	Throughput - Max (MB/s)	Throughput - Total (MB/s)
vfasname01:aggr1	181.17	143.62
vfasname06:aggr1	127.19	26.75
vfasname05:aggr1	89.83	18.20
vfasname02:aggr2	24.57	9.70
vfasname05:aggr_opm1	14.61	4.75

14 items found

Cancel Save

Volume latency:

Edit widget

Volume Latency

Show legend

Override dashboard time: OFF

3h 24h 3d 7d 30d Custom

Volume

Latency - Total

Filter by

Storage: \$storage

Nodes: \$node

Storage pools: \$pool

Name: \$volume

Tier: \$tier

Roll up: +

Show: Top 5

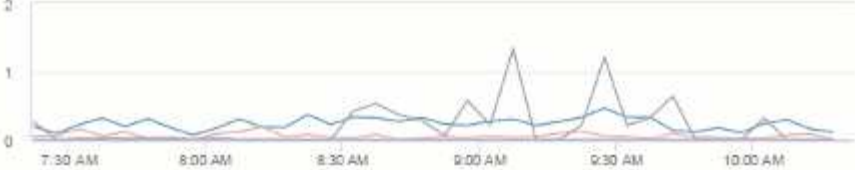
More options

Display: Line chart

Y-axis: Primary

Convert to Expression

+Add



Cancel

Save

Edit widget

Volume Peak and Avg Latency

Override dashboard time: OFF

3h 24h 3d 7d 30d Custom

Volume

Filter by

Storage: \$storage

Nodes: \$node

Storage pools: \$pool

Name: \$volume

Latency - Total (ms): >= Latency

Tier: \$tier

Group by: +

Name	Latency - Max (ms)	Latency - Total (ms)
vifasname05 /vol/bo...	0.00	0.00
vifasname05 /vol/bo...	0.19	0.06
vifasname05 /vol/bo...	0.00	0.00
vifasname05 /vol/bo...	0.00	0.00
vifasname05 /vol/bo...	0.00	0.00

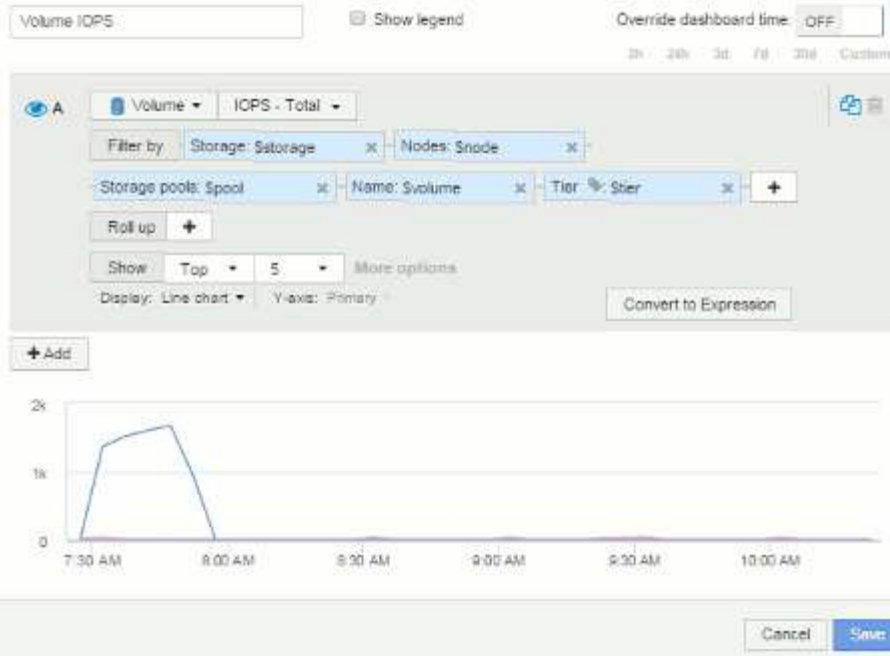
51 items found

Cancel

Save

Volume IOPS:

## Edit widget



## Edit widget

Volume Peak and Avg IOPS

Override dashboard time: OFF

3h 24h 3d 7d 30d Custom

Volume

Filter by: Storage: Sstorage x Nodes: Snode x Storage pools: Spool x

Name: Svolume x Tier: Stier x +

Group by +

Name	IOPS - Max (IO/s)	IOPS - Total (IO/s)
vfasname05/vol/vl...	1,889.31	198.97
vfasname05/vol/vl...	50.03	19.18
vfasname05/vol/bo...	1.51	1.11
vfasname05/vol/bo...	0.00	0.00
vfasname06/vol/bo...	0.00	0.00

31 items found

Cancel Save

## Best Practices for Dashboards and Widgets

Tips and tricks to help you get the most out of the powerful features of dashboards and widgets.

### Best Practice: finding the right metric

OnCommand Insight acquires counters and metrics using names that sometimes differ from data source to data source.



When searching for the right metric or counter for your dashboard widget, keep in mind that the metric you want could be under a different name from the one you are thinking of. While drop-down lists in OnCommand Insight are usually alphabetical, sometimes a term may not show up in the list where you think it should. For example, terms like "raw capacity" and "used capacity" do not appear together in most lists.

Best practice: Use the search feature in fields such as **Filter by** or places like the column selector  to find what you are looking for. For example, searching for "cap" will show all metrics with "capacity" in their names, no matter where it occurs. You can then easily select the metrics you want from that short list.

Here are a few alternative phrases you can try when searching for metrics:

When you want to find:	Try also searching for:
CPU	Processor
Capacity	Used capacityRaw capacity  Provisioned capacity  Storage pools capacity  <other asset type> capacity  Written capacity
Disk Speed	Lowest disk speedLeast performing disk type
Host	HypervisorHosts
Hypervisor	HostIs hypervisor
Microcode	Firmware
Name	AliasHypervisor name  Storage name  <other asset type> name  Simple name  Resource name  Fabric Alias

Read / Write	Partial R/W Pending writes  IOPS - Write  Written capacity  Latency - Read  Cache utilization - read
Virtual Machine	VMIs virtual

This is not a comprehensive list. These are examples of possible search terms only.

## Best Practice: finding the right assets

The Insight assets you can reference in widget filters and searches vary from asset type to asset type.

In dashboards, the asset type around which you are building your widget determines the other asset type counters for which you can filter or add a column. Keep the following in mind when building your widget:

This asset type / counter:	Can be filtered for under these assets:
Virtual Machine	VMDK
Datastore(s)	Internal Volume VMDK  Virtual Machine  Volume
Hypervisor	Virtual Machine
Is hypervisor	Host
Host(s)	Internal Volume Volume
Cluster	Host Virtual Machine
Fabric	Port

This is not a comprehensive list.

Best practice: If you are filtering for a particular asset type that does not appear in the list, try building your query around an alternate asset type.

## Scatterplot Example: knowing your axis

Changing the order of counters in a scatterplot widget changes the axes on which the data is displayed.

### About this task

This example will create a scatter plot that will allow you to see under-performing VMs that have high latency compared to low IOPS.

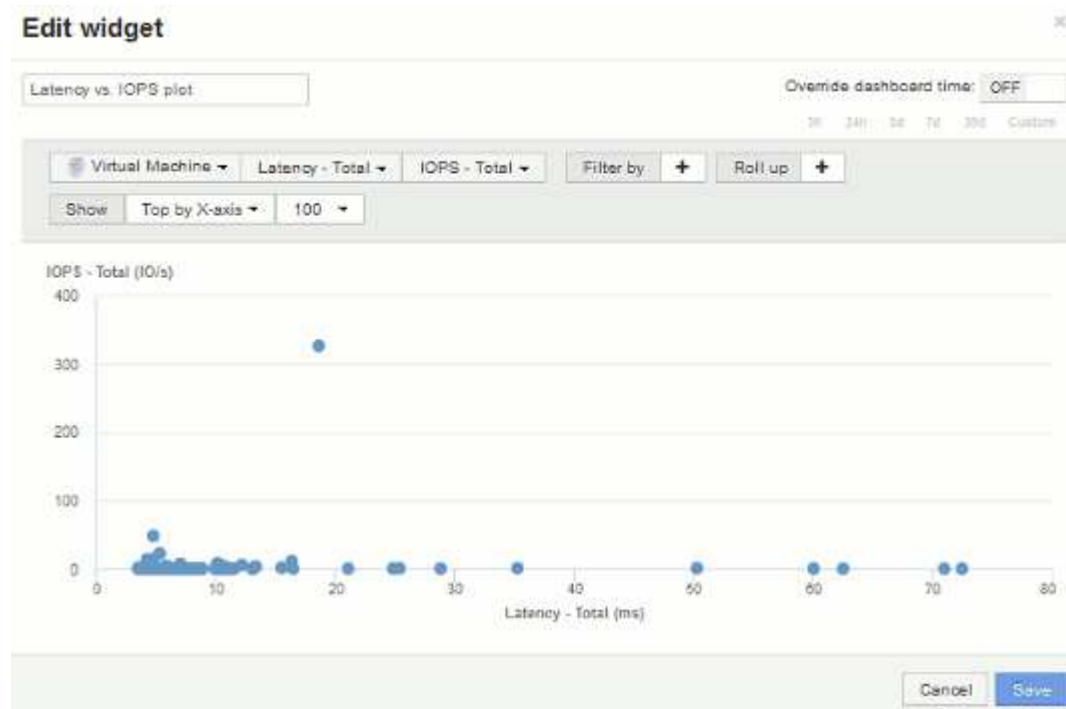
### Steps

1. Create or open a dashboard in edit mode and add a **Scatter Plot Chart** widget.
2. Select an asset type, for example, **Virtual Machine**.
3. Select the first counter you wish to plot. For this example, select **Latency - Total**.

*Latency - Total* is charted along the X-axis of the chart.

4. Select the second counter you wish to plot. For this example, select **IOPS - Total**.

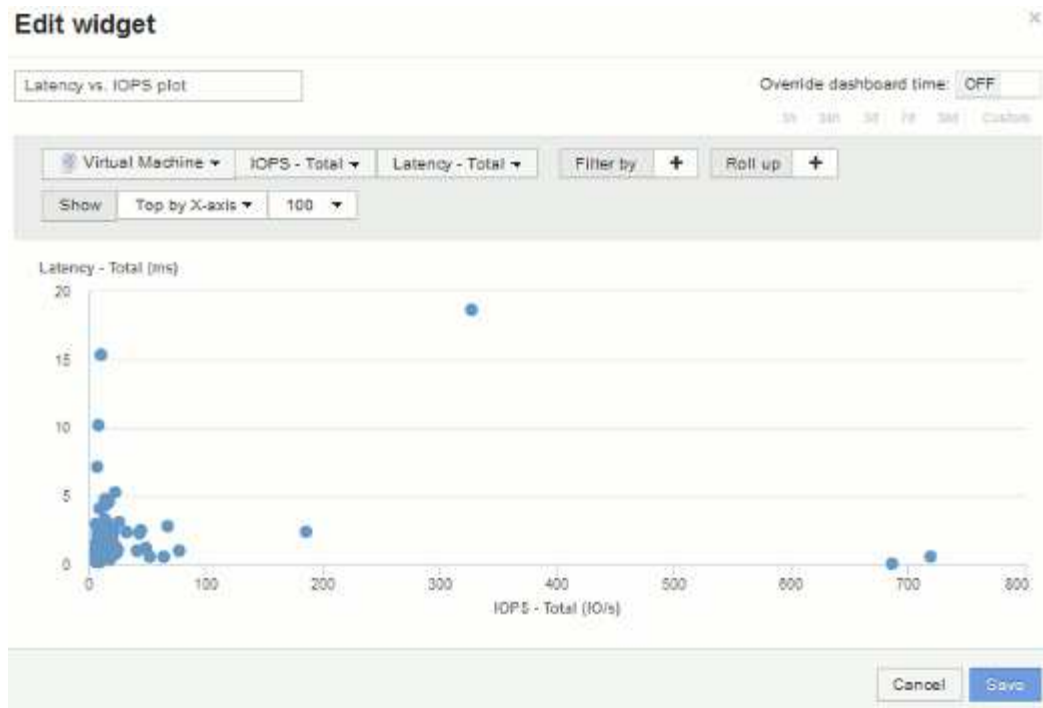
*IOPS - Total* is charted along the Y-axis in the chart. VMs with higher latency display on the right side of the chart. Only the top 100 highest-latency VMs are displayed, because the **Top by X-axis** setting is current.



5. Now reverse the order of the counters by setting the first counter to **IOPS - Total** and the second to **Latency - Total**.

*latency- Total* is now charted along the Y-axis in the chart, and *IOPS - Total* along the X-axis. VMs with higher IOPS now display on the right side of the chart.

Note that because we haven't changed the **Top by X-Axis** setting, the widget now displays the top 100 highest-IOPS VMs, since this is what is currently plotted along the X-axis.



6. You can choose for the chart to display the Top N by X-axis, Top N by Y-axis, Bottom N by X-axis, or Bottom N by Y-axis. In our final example, the chart is displaying the Top 100 VMs that have the highest *total IOPS*. If we change it to Top by Y-axis, the chart will once again display the top 100 VMs that have the highest *total latency*.

Note that in a scatterplot chart, you can click on a point to open the asset page for that resource.

## Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

## Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.