

Cloud Insights Example Tenant – Example Guide 03

Note: this is an accompanying doc to the Cloud Insights Customer Example Tenant that you will have access to during your Cloud Insights trial. This example tenant is a live tenant that you can explore right away to see how Cloud Insights works and where you can go with Cloud Insights.

While the example tenant example environment might not match your own, it will give you an idea what an active Cloud Insights tenant is like. This is a very simple environment example, though you'll be able to accomplish much more!

Troubleshooting

Along with knowing what you have and where you have it and identifying where you are needlessly wasting resources, having the ability to find where trouble is occurring or about to occur, is a critical ability that Cloud Insights enables for you.

The following sequence will walk you through using a few different methods to identify and determine root causes to common issues in performance and capacity. You will see the note widget on the right-hand side of the Getting Started page.

Troubleshooting - Quickly Identifying Where Issues Are

Along with knowing what you have and where you have it, and identifying where you are overspending or needlessly wasting resources, having the ability to find where trouble is occurring or about to occur is a critical ability that Cloud Insights enables for you.

[What Policy Violations Exist in my Environment?](#)

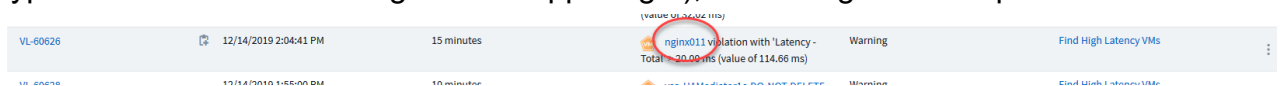
[Where Are My Shared Application Resources Impacted?](#)

[When is my NetApp Storage Node Headroom Not Meeting Optimal Utilization?](#)

- You can start the examples by clicking on this link in the widget: "What Policy Violations Exist in my Environment? (Dashboard)"

What Policy Violations Exist in my Environment?

- The dashboard will appear showing Alert lists for events that have exceeded thresholds of several Performance Policies:
 - o Find High Latency VMs
 - o Find High Latency FlexVols
 - o Find FlexVols Low On Capacity
- Let's use the first Alert for an example of examining a virtual machine latency issue. Scroll the first box of **Alerts for High Latency VMs** till you see either the VM name **nginx011** or the vm name **mongoDB05b** in the Description column (or type it in the filter of the widget in the upper right), both are good examples...



VL-60626	12/14/2019 2:04:41 PM	15 minutes	nginx011 violation with 'latency - Total: 30.00 ms (value of 114.66 ms)	Warning	Find High Latency VMs
----------	-----------------------	------------	---	---------	-----------------------

- Here's a quick explanation of the value of each column entry:
 - o the VL ID number is a unique code related to this particular incident and is a hotlink to the object that violates the threshold (we'll get to the policy in just a moment)
 - o The Time is when the threshold was first breached and the Duration how long the breach lasted (which is important as longer breaches likely have more impact)
 - o The Description includes the object that breached the threshold of the policy (in this case the vm **nginx011** or **mongoDB05b**) as well as the particular threshold exceeded and the maximum value it reached during the breach
 - o The Severity is a tag set when the policy is created noting how important the breach should be considered.
 - o And finally, the Policy name (and link to the policy)
 - o Clicking on the Policy link allows you to review the criteria for the threshold, in this case...

Edit Policy



Policy Name

Find High Latency VMs

Apply to Objects of Type

Virtual Machine

With Annotation

No Value

Annotation Value

Value

Apply After a Window of

5 minutes

With Severity

Warning

Email Recipients

Email will be sent to global recipient list.

Create alert if any of the following are true:

Latency - Total

>

20

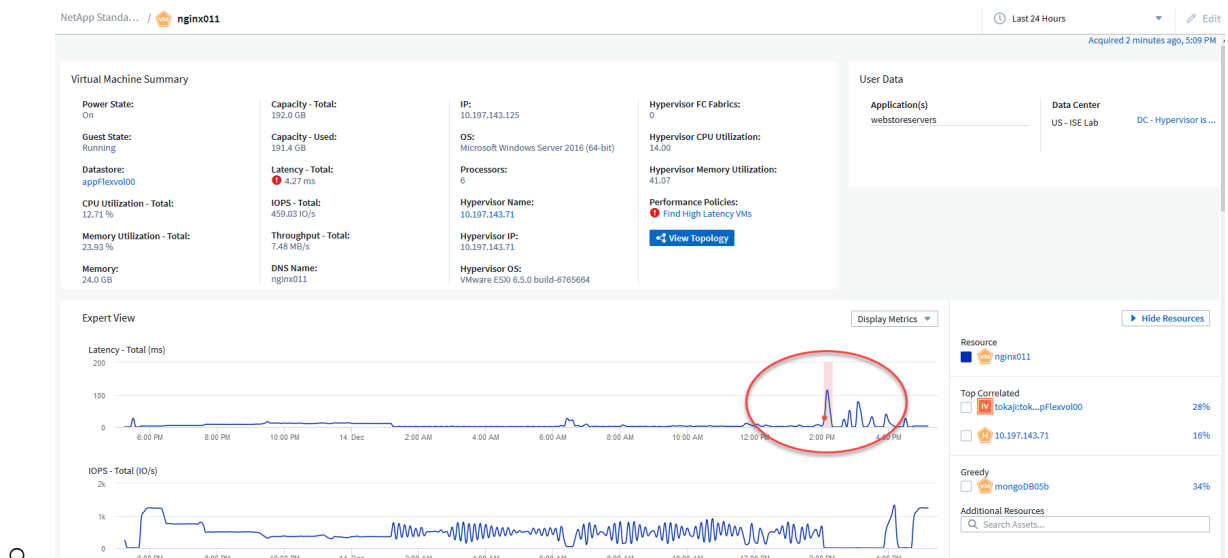
ms

☐ Stop processing further policies if alert is generated

Close

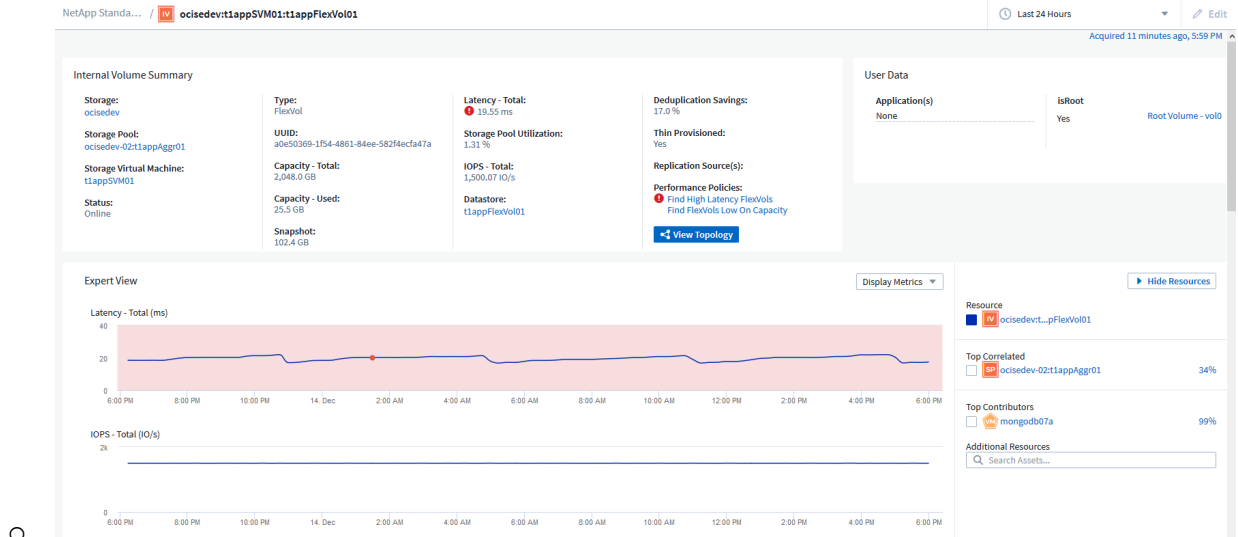
-
- ...this is a policy for monitoring VM's which in this case are not annotated or labeled in more general way to associate them with others (they could be, allowing this policy to only watch those with a particular metadata tag for example of a certain department or business unit), and an alert will not be applied until the threshold at the bottom of the dialog box has been in breach for a certain period of time, in this case 5 minutes. This is important as a momentary latency spike might not be disruptive, but a longer period of latency would likely be
- Though this policy only has one criteria (latency > 20ms), several could be evaluated simultaneously by adding further criteria such as IOPS (for instance if you would like this to apply only when there's an active workload), CPU utilization, etc.

- To find more information about the object in breach of the latency threshold, click on the name **nginx011** or **mongoDB05b**, under the description, or click on the VL ID number at the beginning of the line
 - o You're now shown the "landing page" (or general overview) for the object, displaying a summary description of the object as well as latency over time with the moment the breach occurred highlighted in red in the timeline, and associated and correlated resources such as volumes, hosts and other VM's are shown to the right, some having impact on the object we are investigating:

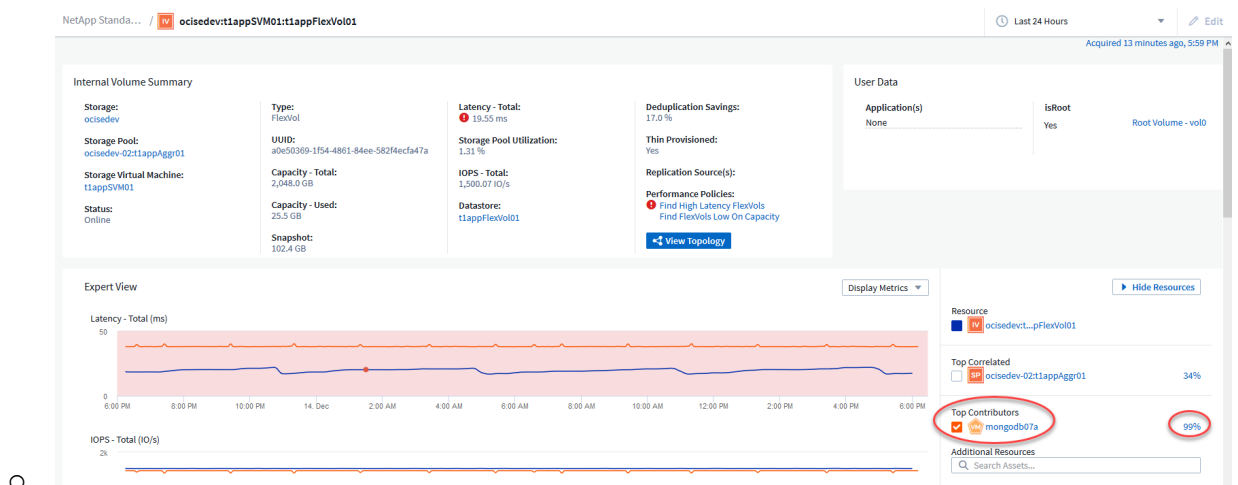


- o Note that the correlated resources (correlation are assets that have performance behavior that is similar in pattern to others) can help identify contribution as well as impact in the cases where contention between objects like VM's on the same volume may have
- o Note that Greedy/Degraded evaluation is not always related to the violation you are tracking, depending on the time frame you are viewing. In this case there is a Greedy VM identified in the 24 hour time span, but if you narrow your time to around the latency event, you may see the correlated Greedy VM disappear; it was not causing this issue at that time. Again, it depends on the time frame visible in the view.
- o For example, sometimes in this example the VMs showing latency and tripping the threshold has just finished a long workload and the momentary latency showed up at the end of the job and did not repeat.
- Let's return to the Violation list and look at more examples of how violations and alerts can help address, and also avoid upcoming issues

- From the second Violations section on the dashboard, **Alerts for High Latency FlexVols**, scroll down in description till you find the volume name **t1appFlexVol01** (or type the first few letters of the name in the widget filter)
- Choose that name in the description and you'll see this landing page:



- The flexVol has been in constant latency, and to the right is a correlated VM likely driving all the load and resulting latency, which you can see if you select to view it and its correlation percentage seen below



Contribution

mongodb07a contributes 99% to ocisedev:t...pFlexVol01

- In this particular case there is a QOS policy limiting the IOPS of the flexVol and the VM has hit the limit and the flexVol and the VM are now suffering high latency
- Back to the Violations dashboard and let's look at another example, this time in alerting the customer when running low on capacity
- Go to the third section, **Alerts for FlexVols Running Low on Capacity**

- Look for the flexVol name **appFlexvol00** in the list (or type the first few letters of the name in the widget filter), and choose the Policy name that has tripped the Alert, the policy in this case looks like this:

Edit Policy
X

Policy Name Find FlexVols Low On Capacity	Apply to Objects of Type Internal Volume
With Annotation No Value	Annotation Value Value
Apply After a Window of First Occurrence	With Severity Warning

Email Recipients
Email will be sent to global recipient list.

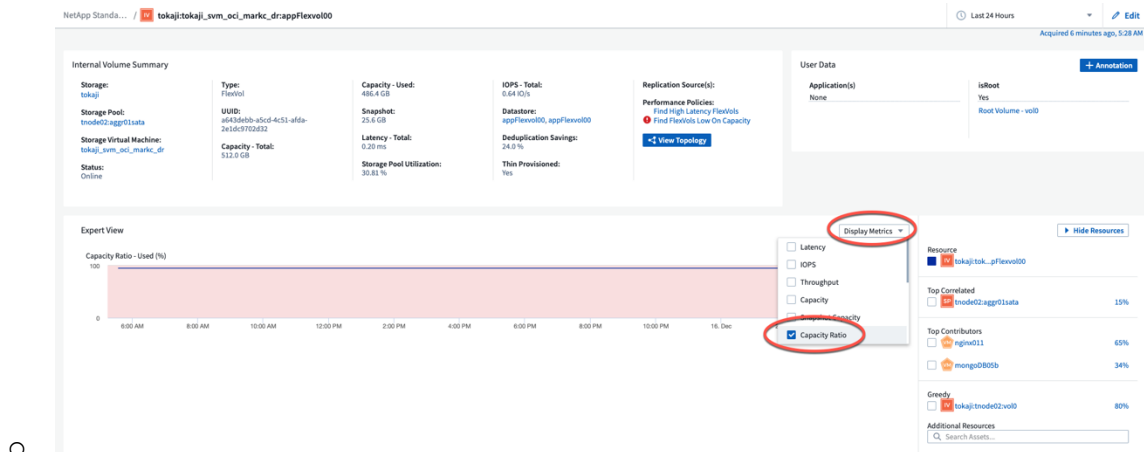
Create alert if any of the following are true:

Capacity Ratio - Used	>	90	%
-----------------------	---	----	---

☐ Stop processing further policies if alert is generated

Close

- It is watching for flexVols reaching 90% or greater (and the first occurrence of it as you don't need 5 minutes of evaluation to know you are running low on space)
 - o Click the name of the flexVol **appFlexvol00** from under the Description column, and you will see its landing page looks something like this (in this example turn off the IOPS and Latency display under "Display Metrics" and turn on Capacity Ratio to see the Capacity timeline):



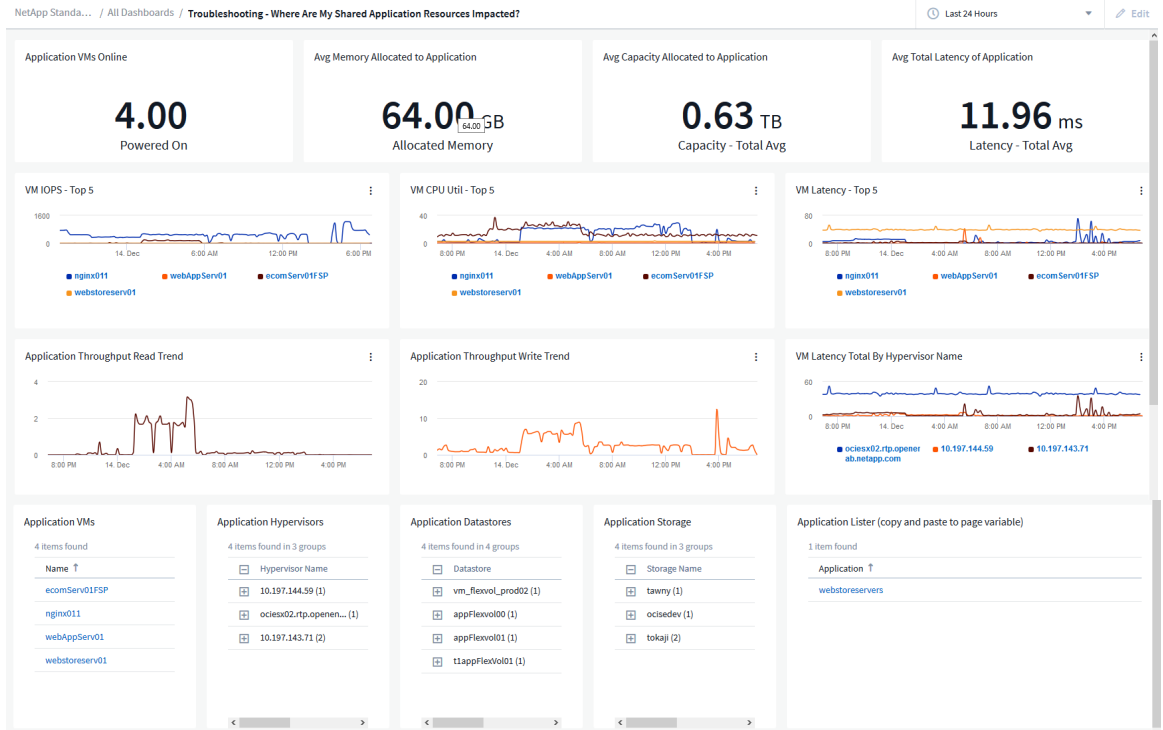
○

Sunday 12/15/2019 6:34:33 PM
tokaji:tokaji_svm_oci_markc_dr:appFlexvol00: 95.00 %

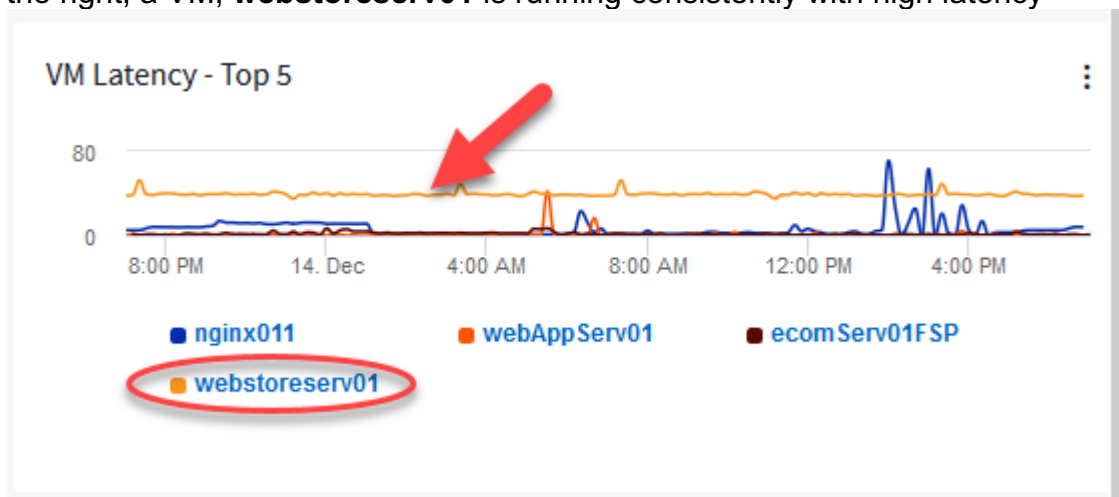
- Your screen may or may not look exactly like this one, though you should have a red indicator of threshold breach someplace in your Capacity Ratio timeline, and the statistic box above should be visible anywhere you place your mouse pointer on the red points on the timeline
- This volume is very near being out of capacity and should be expanded or data removed from it to resolve this issue
- In all of the three examples above, Violations widgets and Performance Policies were used to track results of a particular policy and to alert the user through a Violations dashboard and/or an email message notifying the recipient of the threshold breach, with the info seen on the Violation widget line.

Where Are My Shared Application Resources Impacted?

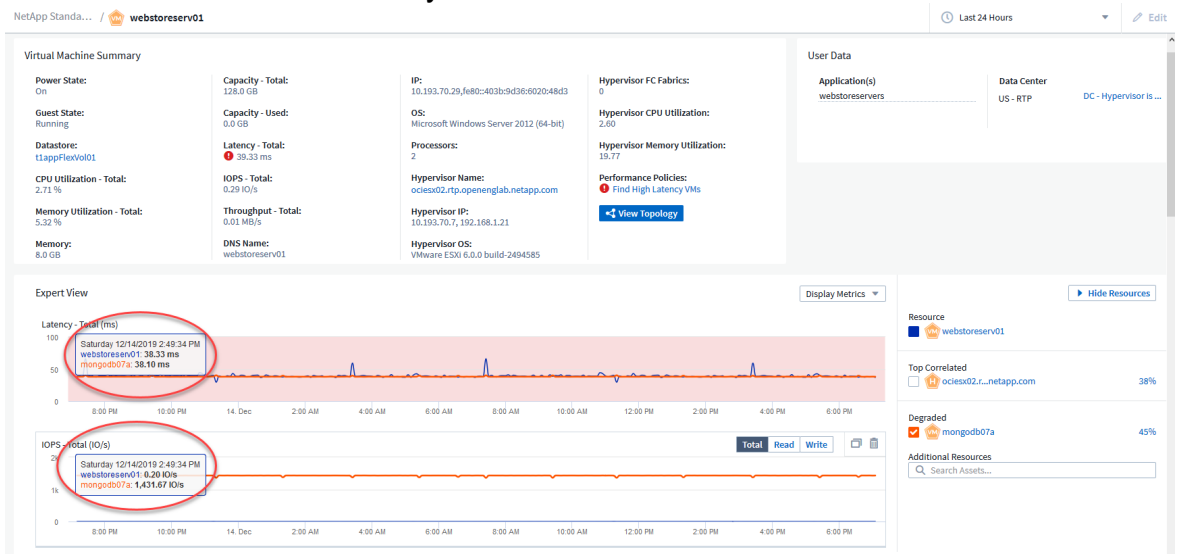
- Another method of determining where issues may exist and understanding what is affecting the VM or volume showing poor performance, is using a dashboard giving you visibility to all the shared resources in related groups of objects.
- Applications (like a tag or annotation) in Cloud Insights are associated assets such as VMs running the same application.
- In this next example, choose the second link under the Troubleshooting heading on the home page, **Where Are My Shared Application Resources Impacted?** which will display the following dashboard:



- This application dashboard tracks 4 VM web application servers in different locations, sharing hypervisor and storage resources with other VM's at those sites
- In situations like these, it helps to be able to monitor capacity and performance across the application servers and quickly narrow down where issues occur and take the proper action to keep services running and performing optimally
- Looking over the resources used by the application helps the customer understand what is needed to support the application too. The memory, CPU and disk metrics are available to be reviewed easily and tables at the bottom of the dash easily display the hierarchy of resources from VM to datastore to hypervisor to storage
- One metric stands out on this page, in the **VM Latency – Top 5** widget on the right, a VM, **webstoreserv01** is running consistently with high latency



- Choose the name from the legend to go to its query page (this shows the filter the widget is using to determine the vm), then click on the name again to go to its landing page, to understand more about what the root cause might be
- On this landing page for the web server, we see a **mongoDB07a** server sharing the same resources as the webserver, and running high IOPS which is starving the shared infrastructure
- It is interesting to note the **mongoDB07a** server is showing Degraded instead of Greedy, as its latency is nearly equal to the webserver. What is really happening is that both VMs are asking too much of the flexVol they are on and both exhibit latency in this case



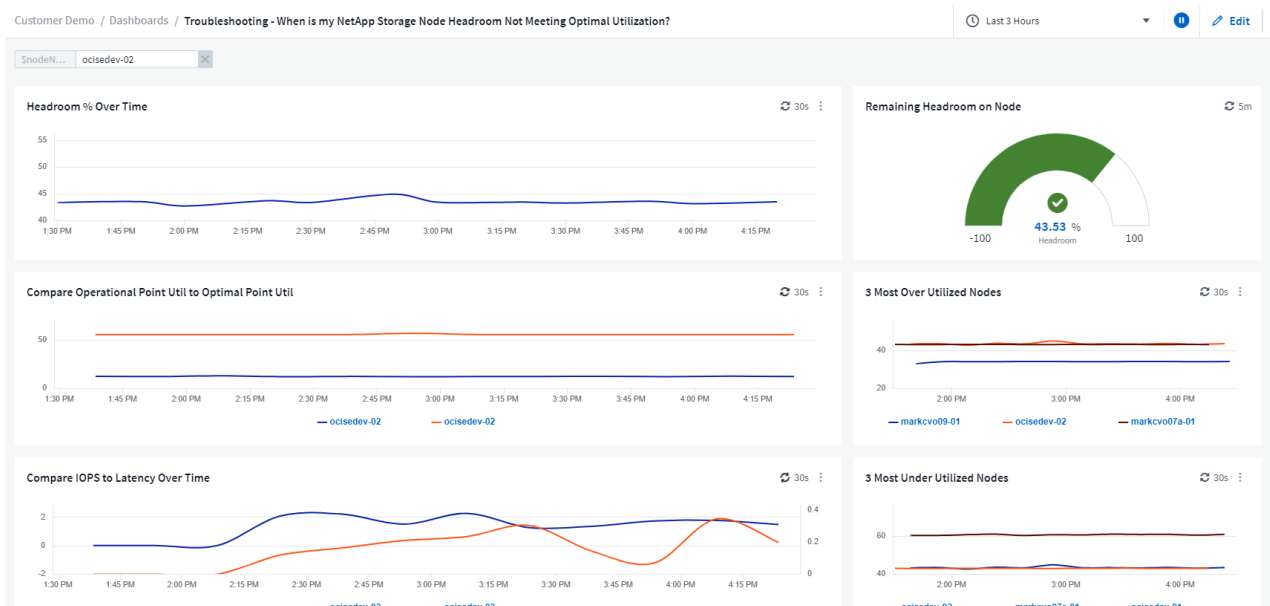
- Further down the landing page, we see the shared flexVol **t1appFlexVol01** that both vm's are contending for IOPS on:

Name	Datstore	Capacity - Ratio - Used (%)	Capacity - Used (GB)	Capacity - Total (GB)	Latency - Total (ms)	IOPS - Total (IO/s)	Throughput - Total (MB/s)
ociesdevt1appSV01t1appFlex...	t1appFlexVol01	1.24	25.49	2,048.00	19.55	1,500.10	13.78

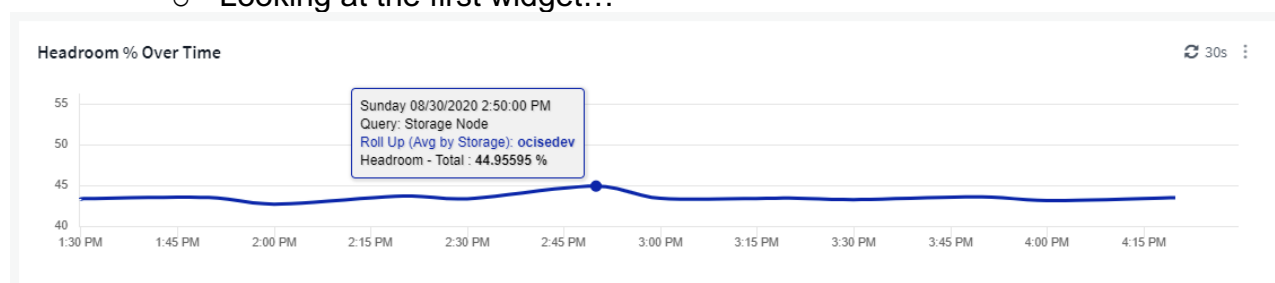
- This example has showed that Cloud Insights provides you the ability to watch over and investigate root cause in multiple assets showing issues in multiple locations in a straightforward and logical manner, not possible with multiple device managers or other tools that do not associate shared resources

When is my NetApp Storage Node Headroom Not Meeting Optimal Utilization?

- From the Home page in the tenant, let's look at another example of how to determine whether issues are caused by over-utilization of storage nodes.
- For this example, choose the link When is my NetApp Storage Node Headroom Not Meeting Optimal Utilization? from the Troubleshooting section on the right of the Home Page



- NetApp ONTAP 9.2 and later has the ability to provide a metric called Optimal Point, which is the utilization point at which the storage node is at its best performance, neither under nor over utilized.
- Cloud Insights can use this metric in dashboards to provide a timeline and current state of storage node “headroom” to provide the customer with knowledge of when their storage nodes are running out of performance, become overutilized and begin to perform badly.
- Looking at the first widget...




- ...this is simply a widget subtracting storage node Utilization Total from Optimal Point Utilization, shows the remaining headroom (> 0) that the storage node currently has. When this value dips below 0, the storage node is overutilized and performance suffers. When it is far above 0, the storage node is underutilized.
- The widget just below, breaks out the two separate metrics Optimal Point (in orange) and Utilization Total (in blue) to see how they track differently based on how they are calculated. You'll note that this helps your customer understand that running storage nodes at 100% utilization is far from getting the best performance from them.
- Using inventive combinations of tools like this in Cloud Insights helps you in solving performance issues better with a clearer idea of how you are managing your asset's performance and effective services to your customers.

Side Note on Optimal Point:

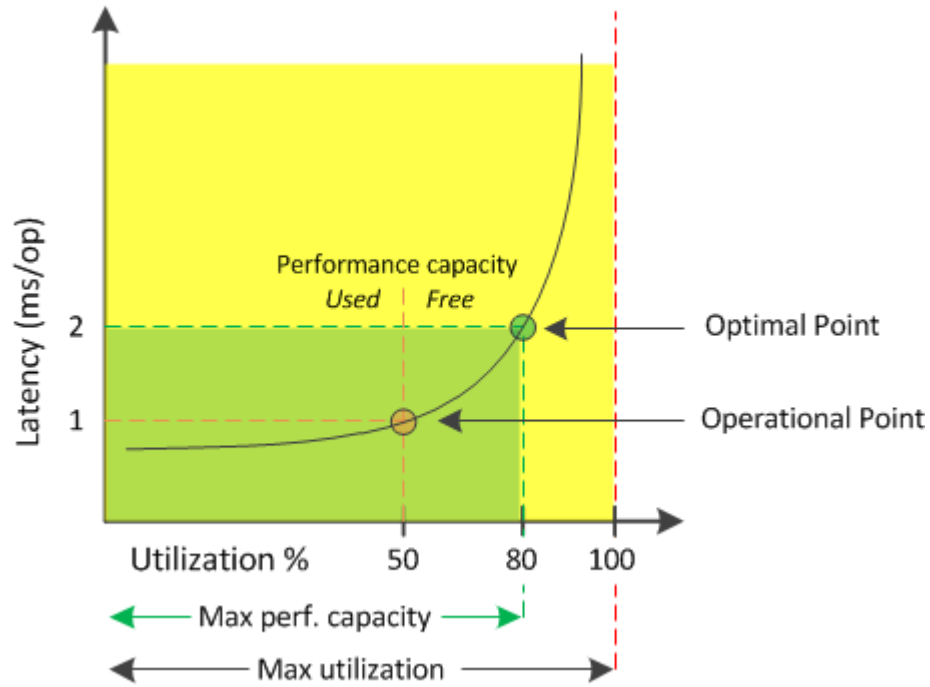
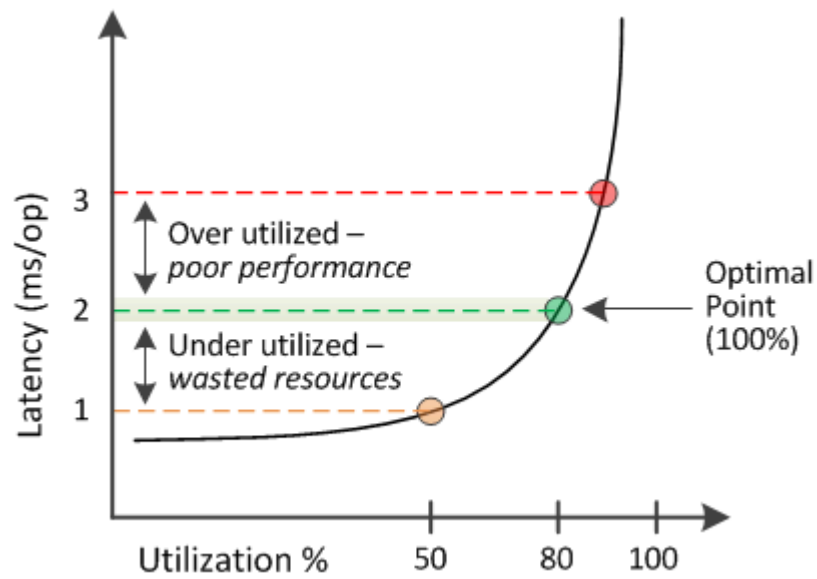
- Here is where we get optimal (point) from. ONTAP 9.2 and later provides us a value called optimal point, which is different for each array and changes based on workload and conditions. This link has a good article to the concepts:
- <https://community.netapp.com/t5/ONTAP-Recipes/ONTAP-Recipes-Easily-identify-remaining-performance-capacity/td-p/133353>
- ...and here is the main concept from that link, and below it two graphs showing the function of these two points:

Counter	Value
-----	-----
ewma_hourly	-
current_ops	4376
current_latency	37719
current_utilization	86
optimal_point_ops	2573
optimal_point_latency	3589
optimal_point_utilization	72
optimal_point_confidence_factor	1



You can compute the available performance capacity for a node by subtracting the optimal_point_utilization counter from the current_utilization counter. In this example, the utilization capacity for CPU_node1 is -14% (72%-86%), which suggests that the CPU has been overutilized on average for the past hour.

- o I believe you can think of the space between the operational point and optimal point as headroom, with the best use of resources being operational and optimal being the same point:



- Note that there is online documentation available to the customer by clicking the [Help->Documentation](#) link from the Cloud Insights menu:

