<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab Overview - HOL-1801-04-CMP - vRealize Operations - Advanced Topics</td>
<td>2</td>
</tr>
<tr>
<td>Lab Guidance</td>
<td>3</td>
</tr>
<tr>
<td>Module 1 - Operating system and application monitoring with Endpoint</td>
<td>10</td>
</tr>
<tr>
<td>Operations (30 minutes)</td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>11</td>
</tr>
<tr>
<td>Exploring Endpoint Operations</td>
<td>13</td>
</tr>
<tr>
<td>Installing the Endpoint Operations Agent</td>
<td>24</td>
</tr>
<tr>
<td>Building a Troubleshooting Dashboard</td>
<td>40</td>
</tr>
<tr>
<td>Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>Module 2 - Creating Custom Dashboards and Super Metrics (30 minutes)</td>
<td>51</td>
</tr>
<tr>
<td>Introduction</td>
<td>52</td>
</tr>
<tr>
<td>Use case: Creating a Summary Metric</td>
<td>53</td>
</tr>
<tr>
<td>Use case: VM Utilization Index</td>
<td>78</td>
</tr>
<tr>
<td>Conclusion</td>
<td>105</td>
</tr>
<tr>
<td>Module 3 - Managing policies (30 Minutes)</td>
<td>106</td>
</tr>
<tr>
<td>Introduction</td>
<td>107</td>
</tr>
<tr>
<td>What are Policies and why are they important?</td>
<td>108</td>
</tr>
<tr>
<td>Refining the Default Policy</td>
<td>112</td>
</tr>
<tr>
<td>Adding new Policies and Groups</td>
<td>142</td>
</tr>
<tr>
<td>Module 4 - Working with views and reports (30 mins)</td>
<td>158</td>
</tr>
<tr>
<td>Introduction</td>
<td>159</td>
</tr>
<tr>
<td>Working with Views in vRealize Operations Manager</td>
<td>160</td>
</tr>
<tr>
<td>Creating a Report from our Views</td>
<td>210</td>
</tr>
<tr>
<td>Module 5 - Using the API for inbound integration (30 mins)</td>
<td>222</td>
</tr>
<tr>
<td>Introduction</td>
<td>223</td>
</tr>
<tr>
<td>A tour through the REST API</td>
<td>225</td>
</tr>
<tr>
<td>Working with resources and metrics</td>
<td>230</td>
</tr>
<tr>
<td>Conclusion</td>
<td>249</td>
</tr>
<tr>
<td>Module 6 - Using Webhook shims for outbound integration (30 mins)</td>
<td>251</td>
</tr>
<tr>
<td>Introduction</td>
<td>252</td>
</tr>
<tr>
<td>Self Healing DataCenter Example</td>
<td>253</td>
</tr>
<tr>
<td>Conclusion</td>
<td>275</td>
</tr>
<tr>
<td>Module 7 - Using VMware PowerCLI for vRealize Operations (45 mins)</td>
<td>277</td>
</tr>
<tr>
<td>Introduction to VMware PowerCLI</td>
<td>278</td>
</tr>
<tr>
<td>Basic Windows PowerShell usage and VMware PowerCLI for vRealize</td>
<td>278</td>
</tr>
<tr>
<td>Operations cmdlets</td>
<td></td>
</tr>
<tr>
<td>Connecting to vRealize Operations with VMware PowerCLI</td>
<td>280</td>
</tr>
<tr>
<td>Using VMware PowerCLI to work with vRealize Operation Alerts</td>
<td>290</td>
</tr>
<tr>
<td>Using VMware PowerCLI to work with vRealize Operations Recommendations</td>
<td>292</td>
</tr>
<tr>
<td>Using VMware PowerCLI to work with vRealize Operations Statistics</td>
<td>302</td>
</tr>
<tr>
<td>Conclusion</td>
<td>318</td>
</tr>
<tr>
<td></td>
<td>322</td>
</tr>
</tbody>
</table>
Lab Overview - HOL-1801-04-CMP - vRealize Operations - Advanced Topics
Lab Guidance

Note: It will take more than 90 minutes to complete this lab. You should expect to only finish 2-3 of the modules during your time. The modules are independent of each other so you can start at the beginning of any module and proceed from there. You can use the Table of Contents to access any module of your choosing.

The Table of Contents can be accessed in the upper right-hand corner of the Lab Manual.

The lab introduces advanced vRealize Operations topics such as crafting your Policies, creating Custom Dashboards and creating custom Views and Reports. We will also introduce the user to End Point Operations for operating system and application monitoring, working with the vRealize Operations API for Inbound integrations, webhook shims for outbound integration and using PowerCLI with vRealize Operations.

Lab Module List:

• **Module 1 - Operating system and application monitoring with Endpoint Operations** (60 minutes) (Advanced) Explore the End Point Operations (EPOps) adapter in vRealize Operations by installing a small agent inside the guest OS that allows you to expose OS-specific metrics, such as OS CPU usage, memory usage and swapping.

• **Module 2 - Creating custom dashboards** (30 minutes) (Advanced) Experiment with creating a custom Dashboard using super metrics. Super Metrics are metrics that aren't directly collected from an underlying collector, but calculated from other metrics. This type of metric is sometimes also known as a synthetic metric or a derived metric.

• **Module 3 - Managing policies** (30 minutes) (Advanced) Introduction to Policies in vRealize Operations. Understanding how policies work and what they do is critical in ensuring vRealize Operations operates in your environment in the way you want it to.

• **Module 4 - Working with views and reports** (30 minutes) (Advanced) Introduction to Views and Reports in vRealize Operations by looking at the various options for creating Views and how they can be built into Report Templates.

• **Module 5 - Using the API for inbound integration** (30 minutes) (Advanced) In this module we will explore the APIs and integration tools offered with vRealize Operations Manager. We will discuss and demonstrate how to export data, as well as how to create custom agents and automate administrative tasks.

• **Module 6 - Using webhook shims for outbound integration** (30 minutes) (Advanced) This lab module will take you through a example of a self healing datacenter. For this lesson we will be stopping a web service and through the use of vRealize Operations, webhook using REST API and Orchrestrator, the service will be started automatically.
• **Module 7 - Using PowerCLI for vRealize Operations** (30 minutes) (Advanced) This module introduces PowerCLI which contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vCloud Director, vCloud Air, vSphere Update Manager, vRealize Operations Manager, and VMware Horizon administration. VMware PowerCLI provides a PowerShell interface to the VMware product APIs.

**Lab Captains:**

- Module 1 - Peter Kieren, Senior Systems Engineer, Canada
- Module 2 - Peter Kieren, Senior Systems Engineer, Canada
- Module 3 - Tony Okwechime, Senior Systems Engineer, United States
- Module 4 - Tony Okwechime, Senior Systems Engineer, United States
- Module 5 - Tony Welsh, Staff Systems Engineer, United States
- Module 6 - Tony Welsh, Staff Systems Engineer, United States
- Module 7 - Tiago Baeta, Senior Systems Engineer, Brazil

This lab manual can be downloaded from the Hands-on Labs Document site found here: [http://docs.hol.vmware.com](http://docs.hol.vmware.com)

This lab may be available in other languages. To set your language preference and have a localized manual deployed with your lab, you may utilize this document to help guide you through the process:

1. The area in the RED box contains the Main Console. The Lab Manual is on the tab to the Right of the Main Console.

2. A particular lab may have additional consoles found on separate tabs in the upper left. You will be directed to open another specific console if needed.

3. Your lab starts with 90 minutes on the timer. The lab can not be saved. All your work must be done during the lab session. But you can click the **EXTEND** to increase your time. If you are at a VMware event, you can extend your lab time twice, for up to 30 minutes. Each click gives you an additional 15 minutes. Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.

### Alternate Methods of Keyboard Data Entry

During this module, you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.
Click and Drag Lab Manual Content Into Console Active Window

You can also click and drag text and Command Line Interface (CLI) commands directly from the Lab Manual into the active window in the Main Console.

Accessing the Online International Keyboard

You can also use the Online International Keyboard found in the Main Console.

1. Click on the Keyboard Icon found on the Windows Quick Launch Task Bar.
Click once in active console window

In this example, you will use the Online Keyboard to enter the "@" sign used in email addresses. The "@" sign is Shift-2 on US keyboard layouts.

1. Click once in the active console window.
2. Click on the **Shift** key.

**Click on the @ key**

1. Click on the "@ key".

Notice the @ sign entered in the active console window.
Activation Prompt or Watermark

When you first start your lab, you may notice a watermark on the desktop indicating that Windows is not activated.

One of the major benefits of virtualization is that virtual machines can be moved and run on any platform. The Hands-on Labs utilizes this benefit and we are able to run the labs out of multiple datacenters. However, these datacenters may not have identical processors, which triggers a Microsoft activation check through the Internet.

Rest assured, VMware and the Hands-on Labs are in full compliance with Microsoft licensing requirements. The lab that you are using is a self-contained pod and does not have full access to the Internet, which is required for Windows to verify the activation. Without full access to the Internet, this automated process fails and you see this watermark.

This cosmetic issue has no effect on your lab.

Look at the lower right portion of the screen
Please check to see that your lab is finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes your lab has not changed to "Ready", please ask for assistance.
Module 1 - Operating system and application monitoring with Endpoint Operations (30 minutes)
Introduction

In this module, we will explore the End Point Operations (EPOps) adapter in vRealize Operations.

As you may already know, the vCenter adapter for vRealize Operations allows you to perform detailed monitoring of all layers of your virtualization infrastructure including virtual machines, hosts, clusters, datastores and virtual networking. While it also offers some limited insight into the guest, such as disk space utilization, it lacks some of the operating system specific metrics.

This is where the End Point Operations adapter comes into play. The End Point Operations adapter works by utilizing a Java agent inside the guest OS that allows you to expose OS-specific metrics, such as OS CPU usage, memory usage and swapping.

Another very important feature of End Point Operations is that it doesn't have to run on a virtual machine. You can install the agent on physical hardware (as long at the OS is supported) and bring in physical machine performance metrics into the vRealize Operations single pane of glass.

How it works

vRealize Operations collects information about your virtual machines by querying vCenter via the API. When you install the Endpoint Operations adapter, you augment the data collected through vCenter with metrics from inside the OS.

This is done by installing an agent on the machine (virtual or physical). This Java agent probes the OS and hardware and sends data back to vRealize Operations. For operating systems running in a vSphere virtual machine, vRealize Operations links the VM object...
to the operating system object so you can see them and build dashboards and alerts based on metrics and discovered properties from the vSphere layer as well as the OS layer.
Exploring Endpoint Operations

In this chapter, we are going to look closer at what the Endpoint Operations adapter offers and how to use it.

Check for Lab Readiness

Before we start the next section, please check to ensure the lab is ready. If it is not, you may need to wait a few minutes until the status shows the Lab is ready.

Log Into the Live Instance of vRealize Operations Manager

The lab environment includes three instance of vRealize Operations Manager. Two of the instances are configured to run in a special mode that allows them to replay historically collected data. We will be using a live instance of vRealize Operations for this lab module.

Launch the Firefox Browser

Launch the Firefox browser:

1. Click the Firefox icon in the Windows taskbar.
Open the vRealize Operations UI

1. Click the **Live vRealize Operations** shortcut or **vRealize Operations Manager - Live Instance** hyperlink to open the vRealize Operations login page.
Select the Authentication Source

You will see different choices for an authentication source to log in to vRealize Operations. In our lab, we have configured the VMware Identity Manager to use Active Directory for authentication and have configured vRealize Operations Manager, vRealize Log Insight and vRealize Business for Cloud to all use the VMware Identity Manager as an authentication source. This allows for a single sign on experience with all three of these vRealize products.

1. Click on the drop-down menu to expand it.
2. Select **VMware Identity Manager** as the authentication source.
3. Click on the **Redirect** Button.
Confirm the Credentials

The following account information should already be filled in. If not, type or select the following information (this is the Active Directory account we will be logging in as):

User name: hol
Password: VMware1!
Domain: corp.local

1. Verify that the information is already filled in.
2. Click the Sign in button.
Finding an Operating System Object

Let's find an operating system object that has an Endpoint Operations agent in it.

1. Click on the Magnifying Glass to expand the search field at the top right corner, type **web-01a**.

A list of vRealize Operations objects with names containing the search string will show up in the list. For this example, they are in order:

- **EP Ops Agent** - this is the Endpoint Operations agent itself. You can find availability and performance metrics for the agent here.
- **Linux** - this is the operating system object. Here you will find the information from the OS layer that has been collected by the Endpoint Operations object.
- **MultiProcess** - in this case, the Endpoint Operations agent was configured to also collect information specific to the web service (httpd / Apache web service) that is running inside of the OS. Processes are not automatically created as objects but one process object was configured in this lab environment.
- **Virtual Machine** - this is the VM object. Here you will find information collected from vCenter.

2. From the list, select the Linux **web-01a.corp.local** as shown in the screen shot.
You will be taken to the Summary page for the operating system object. Remember that everything here has been collected from or calculated based on information from the OS by the Endpoint Operations agent.

**Locating the OS Relationships and Metrics**

1. Click the **All Metrics** Tab.
2. Click the **Show Object Relationship** Down Arrow.

**Viewing the OS Relationships and Metrics**
The top panel is a relationship tree showing the operating system object and it's parents and children.

1. This illustrates how vRealize Operations Manager automatically builds and maintains the relationship between the vCenter representation of the virtual machine and its OS.
2. Expand the **All Metrics -> UTILIZATION** categories and double-click on **Percent Used Memory (%)**.

A graph of the memory usages from the perspective of the OS should show up to the right of the list of metrics.

If you don't see a graph on the right, then double click on the Percent Used Memory (%) Metric again

### Adding Process Monitoring

In addition to the standard OS metrics, such as CPU and Memory utilization, you can also monitor specific processes running on a machine. This will give you availability and utilization on a per-process level.

The processes you wish to monitor are selected using a simple query language. You can search using process name, program path, pid file, pid and a number of other criteria. In our example, we're going to look at the VM Tools process. We do that by matching anything that contains the string "vmtoolsd" in the process name. The "ct" operator in "State.Name.ct" stands for "contains". If you want an exact match, you would use "eq" (for "equals").
To add the process monitor:

1. Click the Actions drop-down.
2. Select Monitor OS Object
3. Select Monitor Processes. A dialog box should pop up.

Create a Process Monitor

We want to monitor the VM Tools process running in the Linux OS of the web-01a.corp.local VM. The process monitor query will look for a process whose name contains the string "vmtoolsd".

1. In the Display Name field, enter VM Tools.
2. In the process.query field, enter State.Name.ct=vmtoolsd
3. Click OK.

The UI will update.
The New Process Monitor

You will need to open the Show Object Relationship section again. You should now see:

1. The new VM Tools process monitor object that you just defined.

Note that the health indicator for the process monitor object will be gray initially. This is because the agent hasn't yet performed its first data collection on the object.
Viewing Metrics For a Process

Since it will take a few minutes for the metrics to be collected for the VM Tools process monitor we just set up, let's look at one that already exists in the lab environment, the Apache httpd process monitor.

1. Click on the **Web Service Process** icon object to bring focus to that object.
2. You can tell which object is "in focus" because the object will have a circle around it.
3. Now the metrics and properties in the lower panel will be for the Web Service Process.
To see metrics specific to this process:

1. In the metrics list on the left, expand the **All Metrics -> UTILIZATION** group and double-click on **Cpu Usage (%)**. You will see the graph on the right-hand side.

The graph you're looking at shows the CPU utilization of that httpd Apache web server service. You can also monitor other metrics or the process availability from here.

**Conclusion**

You should now be able to utilize End Point Operations to access operating system level metrics and process-specific metrics from inside an operating system. Any of these metrics can be used in dashboards, symptom and alert definitions, views and reports within vRealize Operations.
Installing the Endpoint Operations Agent

In this module we will Install the Endpoint Operations Agent on a Windows 10 Virtual Machine. This will show you the installation process and how to confirm that the agent is running inside of the virtual machine. When you have a workload that you want to monitor at a more granular level from vRealize Operations, the Endpoint Operations Agent will allow access to view Process and Alerts inside the virtual machine.

The Endpoint Operations agent is available for a variety of Windows and Linux platforms. It can be installed either with its own copy of a Java Runtime Environment (JRE) or a smaller package can be downloaded and installed and then configured to use a compatible JRE environment that is already existing on the server that will be monitored.
Connecting to the Target Server

We will be installing on a Windows 10 VM that is in our lab pod. In order to do so, we first need to connect to the Windows server via a Remote Desktop Connection session.
1. Click the **Windows Start Menu** Icon
2. Click the **Remote Desktop Connection** shortcut.
3. In the Remote Desktop Connection window, verify that it is set to connect to `win-10.corp.local`.
4. Click on the **Connect** button to launch the RDP Session.

### Installing the End Point Operations Agent on Windows - Continued

An Endpoint Operations installer has been placed on the desktop of the Windows server. This agent package includes a JRE that will be installed in the Endpoint Operations directory alongside the Java agent code.

### Launch the Installer

![Image showing the Windows desktop with the Endpoint Operations installer highlighted]

On the Windows 10 desktop:

1. Double-click on the Endpoint Operations installer to launch it.

The installation wizard will start.
Installation Wizard

1. Click Next to continue the installation.
You enter the location of the vRealize Operations Manager cluster. It is a best practice to use a fully qualified domain name of the cluster and not the IP address of an individual node.

1. In the Server Address Field enter: vrops-01a.corp.local Leave the Secure Port at the default of 443
2. Click Next to continue the installation.
Log In to the vRealize Operations Admin UI

For security purposes, the agent needs to have the SSL certificate thumbprint from the vRealize Operations cluster. This keeps unauthorized parties from trying to connect to the agent. The vRealize Operations cluster thumbprint can be found on the Admin console of vRelize Operations Manager.

1. Launch Internet Explorer from the taskbar on the Win 10 remote desktop session
2. Type the Admin UI of vRealize Operations in the address bar: https://vrops-01a.corp.local/admin - note that this is NOT the address of the page that is currently open.
3. Type in the following credentials:

   User name: admin
   
   Password: VMware1!
   
4. Click the Log In button

---

**Copy The Certificate Thumbprint**

1. Launch Firefox and go to the following address:

   https://vrops-01a.corp.local/admin
Login with the following Username: **admin**  Password: **VMware1!** (Not Shown)

1. Click on the **Certificate Icon** in the upper right hand corner of the web page.
2. Find the following information for the Certificate Thumbprint:
   

3. Highlight the Certificate Thumbprint and Copy the information using **CTRL-C**
4. Click Cancel to close the Operations Manager Certificate Information Webpage.

**Paste The Certificate Thumbprint**

1. Paste the Certificate Thumbprint using **CTRL-V** into the Certificate thumbprint field in the installation wizard.
2. Click **Next** to Continue the installation.
**Enter Server Credentials**

Here you enter credentials that have sufficient privileges in vRealize Operations to perform the collections. See the product documentation for more details. Here we are just using our local admin account.

1. **User Name:** admin
2. **Password:** VMware1!
3. Click **Next** to continue the installation.
Accept the License Agreement

1. Click on the **I accept the agreement** Radio Button
2. Click **Next** to continue the installation.

Agent File Location

- **Select Destination Location**
- Where should End Point Operations Manager Agent be installed?

- Setup will install End Point Operations Manager Agent into the following folder.

- To continue, click Next. If you would like to select a different folder, click Browse.

- At least 215.1 MB of free disk space is required.
1. Accept the default File Location and Click Next to continue.

**Agent Installing**

![Agent Installing](image)

The agent will now install.
Installation Wizard Complete

1. Click on the Finish Button to complete the installation. Congratulations, you have now installed the End Point Operations Manager Agent on a Windows Device!

You can review the Readme.txt file that will launch after you click the Finish Button. When you are done with the review, close the text file and move on to the next step.

Checking the Status of the Agent
Open Windows Services

1. Click on the **Windows Start Menu** and Enter `Services.msc` in the search bar.
2. Click on Services Desktop App to launch the Services Console.
Find the EndPoint Operations Manager Service

1. Scroll Down to the **EndPoint Operations Management Agent** service
2. Verify that the agent service is **Running**.

Close the RDP Window

We will now check and see the status of the agent within the vRealize Operations Manager Console.

1. Close the RDP Session and return to the Main Console.
Search For win-10 Objects

Now we can verify that the agent has been registered in vRealize Operations.

In vRealize Operations Manager UI:

1. In the search bar type `win-10` to see a list of all objects that contain that string.
2. Notice that the list includes an EP Ops Agent object and a Windows object from the agent we just installed.
3. Click the `Win-10.corp.local` Window object link.

View Endpoint Operations Objects in vRealize Operations Manager

Here you can see the:
1. Agent object
2. The Windows OS object
3. And notice that the OS object has been connected to the win-10 virtual machine object as well.

Congratulations, you have now successfully installed the Endpoint Operations agent on a Windows 10 virtual machine and verified that the agent is sending data to vRealize Operations Manager.

Conclusion

In this chapter, we installed and configured the **End Point Operations Agent** inside a Windows VM. With this agent installed, you can now monitor items within the OS from within the vRealize Operations Manager Console. This expands the functionality that is available to administrators.

There is also a Linux version of the End Point Operations Agent available for download from VMware.
Building a Troubleshooting Dashboard

In this module we will build a troubleshooting dashboard within vRealize Operations Manager that will use custom process information that is pulled from the Endpoint Operations Agent. This will give easy access to process information for diagnosing and resolving issues quickly and easily.

Logging in to vRealize Operations

Open the Firefox web browser from the taskbar
1. Click the "Live vRealize Operations" shortcut or the vRealize Operations Manager - Live Instance hyperlink as shown above.
1. When the login page appears, Select the VMware Identity Manager authentication method from the dropdown menu.
2. Click on the Redirect Button.

1. Accept the default username: **HOL** and password: **VMware1!!**.
2. Click on the **Sign In** button to continue.

If during your session should time out while you're taking the lab, simply repeat steps the log in steps to re-authenticate into the vRealize Operations Console.
Creating a New Dashboard

1. Navigate to the Dashboard section.
2. Click on "Create Dashboard".
1. Click on the **Dashboards** button
2. Click on the **Actions** menu item and select **Create Dashboard**. An editor window will pop up.

![New Dashboard]

1. In the Name field, enter **Endpoint Operations**.

**Configuring the Health Widget**

![Configuring the Health Widget]
1. Click the **Widget List** tile on the left hand side.
2. On the left side, locate the **Health Chart** widget, click and hold down the mouse button and drag it onto the right hand side as shown in the screenshot. Release the mouse button.
3. Click the **Edit** button (looks like a small pencil) on the top right corner of the widget. A widget editor will pop up.

![Screenshot of Health Chart Editor](image)

1. Leave Health Chart as the name or fill in your own.
2. Click the Self Provider Radio Button to **On**.
3. In the Order By dropdown, select **Name**.
4. In the list of object tags (lower portion of the window), expand **Object Types**, scroll down to **Virtual Machine** and select it.
5. Click **Save**. You should now see a widget containing a list of virtual machines colored by their health status.
Configuring the Object Relationships Widget

1. Click the **Widget List** tile on the left hand side.
2. On the left side, locate the **Object Relationship** widget, click and hold down the mouse button and drag it to the right of the **Health Chart** as shown in the screenshot. Release the mouse button.

Configuring the Metric Picker & Metric Chart

1. Click the **Widget List** tile on the left hand side.
2. On the left side, locate the **Metric Picker** widget, click and hold down the mouse button and drag it below the **Health Chart** as shown in the screenshot. Release the mouse button.
3. On the left side, locate the **Metric Chart** widget, click and hold down the mouse button and drag it to the right of the **Health Chart** as shown in the screenshot. Release the mouse button.

**Configuring the Widget Interactions**

As you may have noticed, all widgets except the Health Chart are blank. That's because we haven't decided how the widgets should interact yet. Let's set it up as follows: What you select in the Health Chart determines what's shown in the Object Relationship and what you click in the Object Relationship determines what you see in the metric picker and metric chart. In other words, the Health Chart allows you to focus on a virtual machine and the Object Relationships allows you to pick an object related to that virtual machine.

On the left hand side, select the Widget Interaction tile. You should see five dropdowns (You will have to scroll to see the bottom two) . Set them as follows (top to bottom):

1. **--- Provider List ---**  --> **Health Chart (Not Shown)**
2. **Health Chart**  --> **Object Relationship**
3. **Object Relationship**  --> **Metric Picker**
4. **Object Relationship**  --> **Metric Chart**
5. **Metric Picker**  --> **Metric Chart**
6. Click **Apply Interactions**
7. Click **Save** on the lower right hand corner of the dashboard editor. The browser window will reload and show the new dashboard.

Don't navigate away from the current view! You will need it in the next exercise!
Testing the Dashboard

If not already open, click on the **Endpoint Operations Dashboard**.

1. On the health chart, click on the graph for **app-01a**. The rest of the widgets should populate.
2. Let's drill down into the OS! In the Object Relation widget, click the OS node marked **app-01a.corp.local**.
3. In the metric picker, expand the UTILIZATION node and double-click **Cpu Usage (%)**. You should now see a graph on the right-hand side.
Drilling Deeper

Finally, we're going to drill into the OS instance and look at some of the processes running.

1. Double-click on the OS node marked `web-01a.corp.local`. This should expand it and show some of the processes running on the machine.
2. Click on the **Web Service Process**. This should update the metric picker.
3. Expand the **UTILIZATION** node and double-click **Cpu Usage (%)**. You should see a graph of CPU utilization.
4. (Optional) Experiment with various time ranges and metrics as we have shown in previous steps.

Conclusion

In this chapter, we developed a sophisticated Dashboard to help troubleshoot VM's and the process/applications running on them. You should now feel comfortable building out a custom dashboard, with different time ranges. We also showed how the interactions with the different Dashboard components can help you isolate information that will be useful in troubleshooting situations.
Conclusion

You've finished Module 1!

Congratulations on finishing the Operating system and Application Monitoring with Endpoint Operations

In this module, we have explored the Endpoint Operations functionality of vRealize Operations. We have covered the following main topics:

• Overview of the Endpoint Operations
• Installation of the Endpoint Operations Agent on a Windows 10 machine
• Drill-down into processes
• Design of a simple troubleshooting dashboard

Additional Materials

If you are looking for additional information on Endpoint Operations:

• Click on this link

Proceed to any module below which interests you most. [Add any custom/optional information for your lab manual.]

• **Module 2 - Building Custom Dashboards** (30 minutes) (Advanced)
• **Module 3 - Managing Policies** (30 minutes) (Advanced)
• **Module 4 - Working with Views and Reports** (30 minutes) (Advanced)
• **Module 5 - Using the API for Inbound Integration** (30 minutes) (Advanced)
• **Module 6 - Using Webhook Shims for Outbound Integration** (30 minutes) (Advanced)
• **Module 7 - Using PowerCLI for vRealize Operations** (30 minutes) (Advanced)

How to End Your Lab

If you wish to conclude your lab at this time click on the **END** button. This will terminate your lab and all progress. Do this only if you wish to **NOT** proceed with the other modules.
Module 2 - Creating Custom Dashboards and Super Metrics (30 minutes)
Introduction

In this module, we will discuss and experiment with creating a custom dashboard using super metrics. Super Metrics are metrics that aren't directly collected from an underlying collector, but calculated from other metrics. This type of metric is sometimes also known as a synthetic metric or a derived metric.

Super metrics have many different uses. For example, you can use them to combine existing metrics on a resource. Let's say you have a memory consumed metric and a total memory metric. You want to combine these two and keep track of it as a percentage. You can easily create a super metric for that. Another example is rolling up metrics of multiple resources into a single one. You may have a group of hosts that aren't in the same cluster where you want to keep track of the total remaining capacity. This can be accomplished by creating a group or hosts using a Custom Group and then summing up the remaining capacity of the members of that group.

In this lab module, we will create a super metric and use it along with collected metrics in a custom dashboard.

By completing this module you should have an idea on how to use different types of dashboards to bring pertinent information in an easily consumable format.
Use case: Creating a Summary Metric

In this chapter, we will examine the case where we're looking to summarize or roll up a metric from child objects. Specifically, we are going to look at how we can calculate the CPU demand average for virtual machines on a host. This can be an important way to aggregate and blend statistics.

Each virtual machine has a CPU Demand % metric that indicates how much CPU that virtual machine is demanding in relation to its entitlement. So if we are entitled to, say 1000MHz of CPU and we're currently running at 900MHz, our Demand% would be 90%. This metric is useful for determining how close VM is to exhausting its allocated resources.

Taking the average of the Demand % of the virtual machines running on a host could be useful as a wellness measure of the VM population. A high average would indicate that the virtual machines on that host are close to running out of CPU capacity. Notice that this number will be different from the Demand % of the host itself. A big difference between the VM averages and the host Demand % could indicate that the virtual machines have resource restrictions that are too tight for them to take advantage of the capacity of the host.

Enough theory for now! Let's go build this super metric!
Log Into The Live vRealize Operations Instance

Open the Firefox web browser.

1. Click the "Live vRealize Operations" shortcut as shown above. You can select either the link in the toolbar or the hyperlink. Both go to the same login page.
Choose the Authentication Source

1. When the login page appears, make sure the VMware Identity Manager is selected as authentication source.
2. Click on the Redirect button to be redirected to the VMware Identity Manager login.
Log in to vRealize Operations

1. Leave Username: hol and Password: VMware1! as the defaults.
2. Click on the Sign In button.
Create and Enable a Super Metric

1. In the vRealize Operations window, click the Administration tab.
2. Click the Configuration menu section.
3. Click Super Metrics to open the super metrics page.

Create a New Super Metric

1. Click the + icon to create a new super metric.
Name and Object Type

We need to name the super metric and choose the object type that has the metric attribute(s) that we are interested in. In this case, we will be working with the CPU demand for VMs.

1. In the Name field, type **Average VM CPU Demand (%)**. This will be the name of the metric when using it in dashboard, reports, alerts, etc.
2. In the formula field, type `avg(` to begin an average calculation. Be sure to include the opening parenthesis! (Alternatively, if you're not sure of the name or spelling of a function, you may use the Functions dropdown).
3. In the Adapter Type dropdown, select **vCenter Adapter**.
4. From the list right below the Adapter Type, scroll down and select **Virtual Machine**.

We need to name the super metric and choose the object type that has the metric attribute(s) that we are interested in. In this case, we will be working with the CPU demand for VMs.

1. In the Name field, type **Average VM CPU Demand (%)**. This will be the name of the metric when using it in dashboard, reports, alerts, etc.
2. In the formula field, type `avg(` to begin an average calculation. Be sure to include the opening parenthesis! (Alternatively, if you're not sure of the name or spelling of a function, you may use the Functions dropdown).
3. In the Adapter Type dropdown, select **vCenter Adapter**.
4. From the list right below the Adapter Type, scroll down and select **Virtual Machine**.
1. In the Metrics list (lower left-hand side), expand the CPU metric category, scroll down and double-click Demand (%). The formula field at the top of the window should now show the string `avg(${adaptertype=VMWARE, objecttype=VirtualMachine, metric=cpu|demandPct, depth=1})`

2. Click to place the cursor at the end of the formula and type `)` to add the end parenthesis. The string should read `avg(${adaptertype=VMWARE, objecttype=VirtualMachine, metric=cpu|demandPct, depth=1})`.

If you don't see the `avg(${adaptertype=VMWARE, objecttype=VirtualMachine, metric=cpu|demandPct, depth=1})`, ensure you Double Click on the Demand (%) Metric.
Test the super metric

Before we save our super metric, it's a good idea to test it to make sure it behaves the way we expected. The super metric editor provides a mechanism to do this.

1. On the top toolbar, click the **Visualize Super Metric** button (looks like a small line diagram).
2. Click the **Reset Interaction** button (looks like two sheets of paper) on top of the list of objects (see screenshot).
3. In the Filter field on top of the objects list (see screenshot), type `esx` and press Enter. A list of objects containing that string will show up in the list below.
4. Select any of the hosts to see a graph of the calculated super metric value (average of the CPU demands for all VMs running on that host).
5. You should now see a graph in the lower portion of the window that represents the value of the super metric.
6. Click the **Save** to save the super metric definition and exit the editor.
Assign the Super Metric to an Object Type

Next, we need to determine for which object types the super metric will be available. This is useful when you have a more generic super metric that applies to multiple object types, such as Windows and Linux machines. In our case, we want it to appear on Host objects only.

1. Make sure the **Average VM CPU Demand (%)** item is selected in the list of super metrics. Then, direct your attention to the tabs on the lower part of the screen.
2. Select the **Object Types** tab.
3. Click the green **plus sign** (+).

---

**Average VM CPU Demand (%)**

- **Formula Description**: `avg(Virtual Machine: CPU/Demand)`
Select the Object Type

1. In the pop-up dialog, select adapter type **vCenter Adapter** and object type **Host System**.
2. Click **Select** (not shown). The type **Host System** should show up under the Object Types tab.

Enable The Super Metric In a Policy

Before we can use our new super metric, we need to enable it in one or more policies. Policies determine, among many other things, what metrics are collected and which super metrics are calculated for each object type. Let's enable the **Average VM CPU Demand (%)** super metric that we just created our lab environment's policy!

1. Click the **Policies** menu item on the left hand menu.
Edit the Lab Policy

1. Select the Policy Library tab.
2. Click on Hands on Lab Policy to select it.
3. Click the Edit icon (looks like a pencil).

A policy editor will open.
**Editing the policy**

1. Click on the **Collect Metrics and Properties** tile.
2. In the search field in the top right corner, type `average vm cpu` and press Enter.

You will see two entries that match the search string. The first (Object Type = All Object Types) would enable super metric for all objects in the system. This is definitely not something that you would typically want to do.

Note: If you don't see the search field in the top right corner, your screen resolution may be too low to display the full header. Click the small icon that looks like a double arrow on the top right corner of the list and the search field will pop up.
Enabling Metric Collection

Here we want to enable the super metric to be calculated for all host objects that are governed by this policy. The columns indicate the following:

- **State** - whether this attribute (super metric) will be calculated for the object type (hosts)
- **KPI** - whether to set this attribute as a key performance indicator
- **DT** - whether to calculate dynamic thresholds (range of expected values)

For each column and attribute, you can allow the value to be inherited from the parent policy (this is the default for super metrics) or you can specify that for this policy (Local) explicitly calculate (√) or explicitly not calculate the metric.

1. On the line where the object type is **Host System**, set down the State dropdown to √ **Local**.
2. Click **Save** (not shown).

The Dropdown will take a bit of time to open in our Hands on Labs environment. Follow this procedure to change the value:

- Click once on the 'Inherited' text and wait for the cursor to blink on that line.
- Click on the down arrow next to 'Inherited' and wait for the drop-down menu to appear.
- Click on the √ **Yes** entry in the list and wait.
Build a Simple Dashboard using the Average VM CPU Demand Super Metric

Let's create a new dashboard that uses our new super metric.

1. In the vRealize Operations window, click the **Dashboards** tab.
2. Under the **Actions** drop-down, select **Create Dashboard**.

The dashboard editor window will open.

**Free Up Space In The Wizard**

Many wizards in vRealize Operations provide information on the wizard as a fly-out panel. In order to create more working space within the wizard, close the "What is a Dashboard?" panel.
1. Click the double right arrows to collapse the pane.

**Name the Dashboard**

![New Dashboard Interface](image)

Provide a name for the new dashboard.

1. Type in a name for the dashboard: **VM CPU Utilization Summary**.

You can optionally provide a description for the dashboard as well.
Add a Heatmap Widget to the Dashboard

We will add a couple of widgets to our new dashboard.

1. Select the **Widget List** tile.
2. Scroll down to the **Heatmap** widget.
3. Click on the widget and hold down the mouse button. Drag it to the design canvas on the right and release the mouse button. This should result in an unconfigured widget frame on the design canvas.
Add a Top-N Widget to the Dashboard

1. Scroll down to the Top-N widget.
2. Click on the widget and hold down the mouse button. Drag it to the design canvas to a spot below the Heatmap widget and release the mouse button. This should result in an unconfigured widget frame on the design canvas.
Resize the Widgets

To resize the widgets on the canvas,

1. Hover your cursor over the lower-right corner of the Heatmap widget. The corner will darken to indicate that the widget can be resized. Click the corner then
2. Drag it to the right and release the mouse.
3. Scroll down and repeat the resize action on the Top-N widget

Edit the Heatmap Widget

To open the widget editor for the heatmap, do the following:

1. On the top right corner of the heatmap widget, click the edit icon (looks like a small pencil)

A widget editor will pop up.
Configure the Heatmap Name and Group By Object

1. In the Title field, delete the exiting text and type **Average VM CPU Demand**.
2. In the Name field, type **Default**. Heat map widgets can have multiple configurations but in this case we will only configure one.
3. In the Group By field, type **Cluster Com**.
4. A list of suggestions should drop down. Select **Cluster Compute Resource**. This will group hosts by cluster in the heatmap.
Select the Heatmap Metrics

1. In the Object Type field, type Host Sys. A list of suggestions should drop down. Select Host System.
2. Click on the Size By field. A list of metrics should pop up.
3. Scroll down and expand the Super Metric category and select the super metric that we created earlier: Average VM CPU Demand (%).
4. Repeat the above two steps for the Color By field.

Heatmap widgets can be configured to draw the relative box sizes based on one metric value and change the color of each box based on another metric. Or you can choose to have both the color and the size of each box depend on the same metric. In this case, we are setting both attributes based on the super metric that we created earlier.
Set the Color Thresholds

You can let vRealize Operations automatically calculate the Min. Value and Max. Value that will correspond to the color scale that you choose. If you do this, the smallest metric value will become the Min. Value and the largest will become the Max. Value.

In this case, we want to specify the minimum (green) and maximum (red) thresholds.

1. In the Min Value field, type 0.
2. In the Max Value field, type 10. (Normally, the limit for red would probably be more like 90%, but since we’re running a lab with mostly idle VM, we’re setting this limit artificially low to simulate VMs that are running near capacity)
3. Click the Save button. You should now see a heatmap with a few tiles in colors ranging from green to yellow.

Edit the Top-N Widget

You can now edit the Top-N Widget.
1. Click the **Edit** icon (the pencil at the top of the Top-N widget when you hover over the widget).

The Edit Top-N page will open.

**Configure the Name and Time Range**

![Edit Top-N configuration diagram]

When a user clicks on one of the Heatmap widget boxes representing the hosts in our environment, we will want this Top-N widget to display the CPU utilization for all of the VMs running on the selected host in descending order.

1. In the Title field, delete the exiting text and type **Top VM CPU Utilization**.
2. From the Period Length Range dropdown, select **Current Value**.

**Finish the Top-N Widget Configuration**

![Finish Top-N configuration diagram]

1. On the bottom set of tabs, select the **Metric** tab.
2. From the Adapter Type dropdown, select **vCenter Adapter**.
3. Scroll down and select **Virtual Machine** from the list below the Adapter Type.
4. In the Metric list on the right, scroll down, expand the **CPU** section and select **Demand (%)**.
5. Scroll down and click **Save**.

Notice that the widget won't populate. This is because we haven't told it how to get its values and we turned the Self Provider option off. What we want to achieve here is a drill-down, so that a user can click a host in the heatmap and get details about it in the Top-N widget. The next step will set up that interaction.

**Configure the Widget Interactions**

1. On the left-hand side of the editor window, click on the **Widget Interactions** tile.
2. On the lower dropdown (with an arrow pointing to Top VM CPU Utilization), select **Average VM CPU Demand**.
3. Click on **Apply Interactions**.

This widget interaction allows the Top-N list to receive inputs from the Heatmap. In other words, when you click on a host in the heatmap, you will see the top VMs in terms of CPU demand on that host.
Save the Dashboard

Finally, we save the dashboard!

1. Click the Save button in the lower right corner.

The editing window will close and the browser will automatically reload to show the final version of the dashboard.

Examine the Dashboard
1. Hover over the box representing the esx-02a.corp.local host and don't move your mouse. A box will pop up. In this example, the average of the CPU utilization for all VMs on this host is 3.3% (your numbers may differ).
2. Click the box representing the esx-02a.corp.local host.
3. This will trigger the dashboard interaction so that the Top VM CPU Utilization widget shows a list of all VMs on that host and their CPU utilization percent.

Here is the final version of the dashboard! Let's walk through it.

At the top, we have a heatmap that shows on average how much of their entitled CPU the virtual machines are demanding on each host. The "hotter" the color, the closer to full CPU utilization the virtual machines are on average for that host.

If we click on a box in the heatmap representing a host, the lower portion of the dashboard will show us the list of virtual machines on that host and how much CPU utilization each VM has. This is very useful for troubleshooting why we're seeing hotspots among the hosts and could give us important information on how to optimize resource utilization.

**Conclusion**

You have now completed creating a Super Metric Driven Dashboard. vRealize Operations Manager 6.6 includes many new Dashboards that you can use as template for customization. With the number of objects that you can manage or monitor, the possibilities for dashboard creation are endless.
Use case: VM Utilization Index

A Little Bit of Math

\[ I = \frac{2}{\frac{1}{\text{cpu}} + \frac{1}{\text{memory}}} \]

Taking the harmonic mean of the CPU and memory utilization results in the equation above. This will give us a good indication of utilization and has the advantage of being more "fair" when the numbers are very different in size.

Using this formula, a utilization index near 0% means virtually no utilization and an index near 100% means that both memory and CPU are fully utilized or "maxed out". The only problem with this formula is that it doesn't work when either CPU or memory is zero, but that shouldn't happen on a powered-on VM, so we can ignore that problem in this particular case.
Logging in to vRealize Operations

Open the Firefox browser according to the instructions in the lab intro.
1. Click on the "vRealize Operations" shortcut or hyperlink as shown above.
1. When the login page appears, Select the VMware Identity Manager option from the drop down menu.
2. Click on the **Redirect** button to go to the Identity Manager Login.

1. Accept the default user: **HOL** and password: **VMware1**!
2. Click on **Sign In**

If during your session should time out while you're taking the lab, simply repeat the previous steps to log in again.
Lab Guidance

As we have a resolution of 1280*800 in the lab environment, the screen real estate can be a little small for the vRealize Operation Manager window. To help with this, we can use the customize menu in Firefox to zoom out the content. A setting of 80% will allow you to see the full menu bar in the window.

Opening the Super Metric Editor

1. Click on the Administration tab.
2. Select Configuration.
3. Click on Super Metrics.
1. In the vRealize Operations window, click the **Administration** Menu Item on the top bar.
2. Click the **Configuration** sub menu.
3. A new menu opens up. Click the **Super Metrics** item.

**Creating the Super Metric**

![Super Metric creation interface](image)

```plaintext
Creating the Super Metric
```

```plaintext
Super Metrics

- New Super Metric

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Children Badge Health</td>
<td>(sum(SDDC Health: Badge Health) + sum(WOps Health: Badge Health)) + sum(...</td>
</tr>
<tr>
<td>Number of Children</td>
<td>count(SDDC Health: Badge Health) + count(WOps Health: Badge Health) + count(...</td>
</tr>
</tbody>
</table>

```
1. Click the plus sign to create a new Super Metric.

1. In the Name field, enter `utilizationIndex`. This will be the key for the metric when using it in dashboard, reports, alerts, etc.
Creating the Formula

![Diagram of Manage Super Metric interface with options for functions and operators, and a highlighted object with name utilizationindex.](image-url)
1. On the top toolbar, select the **THIS OBJECT** button and make sure it lights up. This means that the next metric we select is going to be for the specific object type we select, rather than for its children.
1. Click on the Adapter Type Dropdown Menu.
2. Select the **vCenter Adapter** under adapter types. This should load a list of object types related to vSphere.
1. In the list of object types, scroll down and select **Virtual Machine**.
2. In the list of Metrics expand the **CPU** metric category.
1. scroll down and double-click on **Demand (%)**.
2. The formula in the top section of the window should update to read `(${this, metric=cpu|demandPct}`.
1. Select the **THIS OBJECT** button again.
2. In the list of Metrics, expand the **Memory** metric category.

1. Scroll down to Usage (%) and Double Click it.
2. The formula should now read $\{this, metric=cpu|demandPct\}$\{this, metric=mem|usage_average\}.

If it doesn't look like the above formula, erase it using backspace and repeat these steps again.
Editing the Formula

In the previous step, we added the metrics to the formula, but the mathematical operators are missing. Let’s add the math we discussed and see what we get!

1. Click the formula field and edit the formula to read
   \[ \frac{2}{1 / \{\text{this, metric=cpu|demandPct}\} + 1 / \{\text{this, metric=mem|usage_average}\}} \]
   The spaces are optional and added for readability only. As you can see, this is equivalent to the utilization index formula we discussed.
2. Click the Show Formula Description button (looks like a blue square with an arrow in it). You should now see a more “friendly” rendition of the formula.
3. Click the Visualize Super Metric button (looks like a small line graph).
4. In the list of resources, select web-02a. You should now see a graph at the bottom of the screen. This shows you in real time what the result of the super metric calculation is. You should see a graph somewhat similar to above. (Optional) click on other virtual machines in the list of resources and see how the graph changes.
5. Click Save to save and close the editor.
Assigning the super metric to an object type

Next, we need to determine on which types of object the super metric should appear. This is useful when you have a more generic super metric that applies to multiple object types, such as Windows and Linux machines. In our case, we want it to appear on Virtual Machine objects only.
1. Make sure the `utilizationIndex` item is selected.
2. Select the **Object Types** tab.
3. Click the plus sign (+)

![Select Object Type]

1. In the pop-up dialog, click on the drop down menu for adapter type
2. Click on the **vCenter Adapter** selection.

![Select Object Type]

1. Scroll down and find the object type **Virtual Machine**. Click on it to select it.
2. Click the **Select** button. The type Virtual Machine should show up under the **Object Types** tab.
Enabling the Super Metric

Before we can use our new super metric, we need to enable it on our policy. Policies determine, among many other things, what metrics are collected and which super metrics are available. Let’s enable the utilizationIndex in our default policy!

1. In the vRealize Operations window, click the **Administration** menu item in the top menu bar.
2. Click the **Policies** menu item.
3. Select the **Policy Library** tab.
4. Click on **Hands on Lab Policy**.
5. Click on **Edit** (the icon that looks like a small pencil). A policy editor should pop up.
1. Click on the Collect Metrics and Properties tile.
2. In the search field in the top right corner, type `utilizationIndex` and press Enter. You should now see two lines.
3. Select the Second Super Metric that has the adapter type of vCenter Adapter.
Enabling the super metric

1. On the lines where the object type is Virtual Machine, set down the State dropdown to √ Local, KPI to √ Yes and DT to √ Yes.

2. Click Save.

Note: The dropdowns can take some time in the lab environment. Please be patient.
Testing the Metric

1. In the search field in the upper right-hand corner, type `app-01a` followed by Enter. The monitoring screen for that resource should load.

2. Click the **All Metrics** sub tab.

3. In the list of metrics, expand the **Super Metrics** metric category.

4. Double-click on the **utilizationIndex** metric. A line diagram should appear.

5. Click on the calendar icon (see above)

6. In the time range dropdown, select **Last 6 hours**. You should see one or a couple of data points.

7. Click the **Go** button.

8. It may take a few minutes for your first data point to appear. Click **Refresh** (the green circle above the diagram) until a data point appears.
A utilization index like the one we just designed is great for comparing workloads to each other. Let's use it for a Top-N-list so we can see which workloads are closest to their resource limits!

1. In the vRealize Operations window, click the **Home** button (the button that looks like a small house)
2. Under the **Actions** menu, select **Create Dashboard**. A dashboard editor window should pop up.
3.
Editing the Dashboard

1. In the Dashboard Configuration Section, name the Dashboard: **VM Utilization**

1. Select the **Widget List** tile.
2. Scroll down to the **Top-N** widget.
3. Left-click on the widget and hold down the mouse button. Drag it to the design canvas on the right and release the mouse button.
4. This should result in an empty widget frame on the design canvas.

**Editing the Top-N Widget**

![Edit Top-N Widget](image)

Let's configure the Top-N widget and point it to our newly created super metric!

Click the Edit button (the small pencil on the top left of the new widget). A new edit window should pop up. (Not Shown)
1. In the Title field, type **Top Resource Utilization**.
2. Check the **On** button for Self Provider.
3. From the Period Length dropdown, select **Current Value**.
4. On the bottom set of tabs, select the **Metric** tab.
5. From the Adapter Type dropdown, select **vCenter Adapter**.

1. Select **Virtual Machine** from the list below the Adapter Type.
2. In the Metric list on the right, scroll down, expand the **Super Metric** node and select **utilizationIndex**.
3. Click **Save**.
Examining the New Dashboard

If everything worked out, you should see a dashboard that looks similar to the one above. You will likely see the NSX-controllers relatively high on this list, since they are the only nodes actually doing any real work. Let’s finish by saving the dashboard.

1. Click the **Save** button in the bottom right corner.
Opening the Dashboard from the Menu

1. From the top menu, select **Dashboards**.
2. Select **VM Utilization**. You should now see the full-size version of your dashboard.

(Optional) Add CPU and Memory Top-N lists
As an optional exercise, you may add the CPU and Memory demand to the dashboard for comparison.

1. From Action menu, select Edit dashboard.
2. Drag and drop Top-N as you did in the previous steps, but instead of selecting the metric Super Metrics/utilizationIndex, select **CPU/Demand (%)** and **Memory/Effective Demand (%)**. You should get a dashboard that looks similar to above.

**Conclusion**

In this use case example, showed how to calculate a number that tells us how heavily a VM is using its compute resources, i.e. CPU and memory. While we could look at the metrics individually, we would like a way to compare workloads using a single number. Such a number is commonly known as a "Utilization Index" and uses some kind of mathematical function to combine CPU and memory.

In this exercise, we used a harmonic mean, which tends to work better when the input numbers are of different orders magnitudes, and setup a dashboard to show the environments utilization. Using formulas like this to show how different metrics can be blended together can give a holistic picture of VM Health and Performance.

You can read more about how harmonic means operate here: [https://en.wikipedia.org/wiki/Harmonic_mean](https://en.wikipedia.org/wiki/Harmonic_mean)
Conclusion

In this module, you learned how to create two different types of super metrics and how to visualize them in a dashboard. We first created a "utilization index" on a per-VM basis that seeks to provide an indication of the overall compute resource utilization of the VM. Next, we went on to create a summary metric, i.e. a metric acting on the children of an object and rolling up some metrics. In our case, we used it to calculate and visualize a CPU demand average per host.

You've finished Module 2: Creating Custom Dashboards

Proceed to any module below which interests you most.

- **Module 1 - Operating System and Application Monitoring with Endpoint Operations** (30 minutes) (Advanced)
- **Module 3 - Managing Policies** (30 minutes) (Advanced)
- **Module 4 - Working with Views and Reports** (30 minutes) (Advanced)
- **Module 5 - Using the API for Inbound Integration** (30 minutes) (Advanced)
- **Module 6 - Using Webhook Shims for Outbound Integration** (30 minutes) (Advanced)
- **Module 7 - Using PowerCLI for vRealize Operations** (30 minutes) (Advanced)

How to End Your Lab

If you wish to conclude your lab at this time click on the END button. This will terminate your lab and all progress. Do this only if you wish to NOT proceed with the other modules.
Module 3 - Managing policies (30 Minutes)
Introduction

This first module will introduce you to Policies in vRealize Operations. Understanding how policies work and what they do is critical in ensuring vRealize Operations operates in your environment in the way you want it to.

We will cover the following topics:

1. What are Policies and why are they important?
2. Initial configuration and the Default Policy
3. Changes you should be considering to the Default Policy
4. Adding new Policies and Groups

The module should take about 30 minutes to complete.

Let's get started...
What are Policies and why are they important?

Let's start by looking at what vRealize Operations Policies are and why they are so important.

Policies in vRealize Operations

![Policy Library]

How you manage your IT systems will vary quite significantly depending on the system itself. For example:

- **Production IT** - in production, you may want some compute resources dedicated to some of your systems. You may decide not to overcommit memory, or, you may want to use thick provisioned eager-zeroed disk. You may want alerts to tell you whenever there are any issues with your systems, regardless of the level of criticality.

- **Test/Development IT** - in test or development parts of your infrastructure you may be more relaxed. You may use thin provisioned disks, and you may be happy with, perhaps significant amounts of, CPU and/or memory overcommit.

Although vRealize Operations has significant analytic capabilities, and can 'self-learn' how your IT systems should be operating, you still need to define some of the fundamental ways that it will operate to account for these differences in your environment.

That is why we have Policies in vRealize Operations.

Before we look at how you change Policies, lets look at the main terms and definitions you need to know about:
Base Settings

- when you install a Management Pack, the objects being managed by that pack have their Base Settings determined. This can be considered the 'out of the box' policy for those object types.

For example, out of the box, vRealize Operations considers Virtual Machine snapshots to be 'old', for waste reclamation purposes, at 180 days. This is configured in the Base Settings for the Virtual Machine Object type.

You cannot make changes to the Base Settings themselves. To make changes you need to create and apply Policies.

Default Policy

- when you install vRealize Operations for the first time, you will typically run the initial settings wizard which creates the Default Policy. The choices you make during installation will probably mean changes from the Base Settings need to be made. These changes are set in the Default Policy.

The Default Policy is then applied to all objects discovered in vRealize Operations.
You always need to have a Default Policy in vRealize Operations. If you don't need to make changes from Base Settings, you still need the Default Policy, it just won't have any changes defined.

**Additional Policies**

You can optionally define additional policies. In the above example we have defined an additional **Production Policy** for our production systems. You'll notice in the screenshot it is nested below the Default Policy. We will cover nesting later in this module.

**Groups**

When using Additional Policies you apply them to **Groups** of objects. In the above example, we have grouped our Production Systems together.
Now that we know the main policy constructs, let's go ahead and dive into the lab!
Refining the Default Policy

First let’s take a look at the Default Policy and the sort of changes you should consider to match the Policy to your environment and operational imperatives.

Open Firefox Browser from Windows Quick Launch Task Bar

1. Click on the Firefox icon on the Windows Quick Launch Task Bar.

HOL-1801 Lab Links Page

Once the firefox browser has been opened, the HOL-1801 Lab Links page should be loaded.
If not, please type the link below to load the HOL-1801 Lab Links Page.

http://192.168.110.10

Open vRealize Operations

1. Click on the **vRealize Operations Manager - Live Instance** link.

Select the VMware Identity Manager Authentication Option
1. Be sure to select the **VMware Identity Manager** authentication option from the drop down menu if it is not already selected.
2. Click the **REDIRECT** button

---

**Login to vRealize Operations**

1. Ensure that the displayed domain is **corp.local**. If a different domain is displayed, you can click on the **Change to different domain** link and then select the **corp.local** domain.
2. **Login** to vRealize Operations Manager with the following credentials. It is likely that the credentials have been auto populated.

<table>
<thead>
<tr>
<th>User name: hol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password: VMware1!</td>
</tr>
</tbody>
</table>

2. Click the **Sign in** button.
A note about using vRealize Operations in Hands on Labs

Unfortunately the resolution we need to use in Hands on Labs does not always allow you to complete some tasks or see enough of the screen. You may, therefore, need to adjust the browser zoom (resolution) on occasion. To do this:

1. Click on the **Open Menu** icon in Firefox.
2. Use the **Minus** icon to reduce the Zoom (to 90% in this case) and the **Plus** icon to increase it back as needed.

Go ahead and practice this now! You will need to use this skill during the lab :-) Set the zoom to **90%**.
Initial monitoring goals

Please answer the following list of questions to create a new default policy or click Next to accept the existing default policy.

Which objects do you want to be alerted on in your environment?
- Infrastructure objects except for Virtual Machines
- Virtual Machines only
- All vSphere objects

Which type of alerts do you want to enable? (Select all that apply)
- Health alerts that usually require immediate attention.
- Risk alerts indicating that you should look into any problems in the near future.
- Efficiency alerts indicating that you can reclaim resources.

Overcommit CPU in your environment?
- Overcommitting allows you to consolidate more VMs and get more out of your resources.
- Yes
- No

Configure Memory Capacity based on?
- True Demand - Most aggressive
- Memory Consumed - vSphere Default
- Do not Overcommit - Most Conservative

Enable vSphere Hardening Guide Alerts?
- Yes
- No

Note: this screenshot is from the deployment wizard - in Hands on Labs we have already deployed vRealize Operations so you won't see this step in your lab.

The screenshot shows the typical choices most people will make when deploying vRealize Operations. Specifying whether you do, or do not, want to Overcommit CPU, and, how you want to configure Memory Capacity, will make key changes to the Default Policy that we will cover next.
Navigate to Administration

Click on **Administration** to bring up the administration panel.
Navigate to Policies

1. Click on Policies to open the Policies panel
2. There is only one Policy in effect in this lab environment, the "Hands on Lab Policy"
3. Notice the D - this shows that this is the Default Policy in place
Edit the Default Policy

Now we are going to make some changes to the Default Policy we have in place.

1. Click on the **Policy Library** tab
2. Click on **Hands on Lab Policy** to select it
3. Click on the **pencil** icon to edit it
The main changes to Default Policy to consider

If you were to use the Define Initial Monitoring Goals wizard during installation, the 'out of the box' Default Policy we are looking at will be close to your requirements, but, there would be some key changes you’d need to make. They are mainly related to the Analysis Settings section of the policy.

1 - The Analysis Settings section always opens first when you edit a policy. Lets explore this screen...

2 - Object Types - every Object Type will have its own policy elements that you can define. The first Object Type, Cluster Compute Resource is open in the right hand panel.

3 - Policy element 'locks'. Notice the locks on the right hand side of the panel. The unlocked icons show policy elements that have had changes from their Base Settings made. The locked ones are the ones with no changes made. In the screenshot we can see that the Workload section has had changes made, but that the Anomalies section below it has not. Notice Anomalies is also greyed out.

4 - Generally speaking, the area where you will make the most changes to policy will be the Capacity Remaining, Time Remaining policy element. You will see some changes have already been made as it is unlocked. Remember the Initial Monitoring
Goals screenshot where you could define CPU and Memory overcommit? That is where these changes will have been made.

The screen you are looking at is the policy elements for just the Cluster Compute Resource Object Type. Each and every Object Type has similar policy elements that can, and should, be considered for change.

Let's now look at the policy elements in turn and, as we get to the ones we need to change, we'll make the necessary changes.

**Workload**

![Workload](image)

1 - Click on the arrow next to Workload to open the section.

Most Object Types are made up of a number of different 'resources' such as CPU, Memory, Disk and Network. **Workload** is a measure of what percentage of a resource is being demanded at a particular point in time.

For example, a Virtual Machine may be configured with 4GB RAM. **The memory workload for that VM would be the amount of RAM actually being used at a given point in time.**

The use of each resource can also be measured in different ways and those ways will vary depending on the Object Type.

For example, consider the Virtual Machine with 4GB memory. **‘Memory Demand’ would be the number of pages currently used by the VM. ‘Memory Consumed’ would be memory allocated to that VM by vSphere.** Both are valid ways of measuring workload - for most use cases, 'Memory Consumed' is usually the better metric to consider for workload.

You can see there is a huge amount of granularity open to you. For the Cluster Compute Object Type, the default three checked resource types are the best way to measure Cluster workload.
As you look at other Object Types, you will see different options for Workload. You will probably find the defaults are appropriate for most use cases.

**Workload Thresholds**

1 - Now look at the **Workload Score Threshold** graphic.

Remember, we are in the *Analysis Settings* section. Analysis, refers to all the Badges that an object has. This graphic refers to the score values that will change the colour of the badge and affect the criticality of related alerts. If you hover over them you will see the values. You can drag the coloured icons left and right to change these values.

You will see a similar graphic for every Analysis Settings policy element. The out of the box values are usually fine. You would, perhaps, change them if you ran a particular part of your infrastructure 'hotter'. For example you may consider a critical alert in these cases should be triggered at 99% Workload instead of the default of 90% Workload.

2 - We aren't going to make any changes to this section so click on the *arrow* again to collapse it.

**Anomalies and Faults**
The only changes you can make to Anomalies and Faults are to their badge score thresholds. If you do need to make changes, you will need to click on the lock icon to unlock them. In the above screen you will see we have unlocked the Anomalies section.

Click on the lock in your environment to unlock it, then, optionally click again to lock it.

It's unlikely you will ever need to make changes to these policy elements.

**Capacity Remaining, Time Remaining**

This is the Policy Element you will most often need to make changes to.

1. Click on the arrow to expand the section
2. Scroll down so you can see all the Checked items as seen on the next page
There are a lot of options! You will see there are 6 areas, or *resource dimensions*, to consider for a Cluster Compute Object Type:

- Memory
- CPU
- Network I/O
- Datastore I/O
- vSphere Configuration Limit
- Disk Space

Also, the key resource dimensions of Memory, CPU and Disk Space have 2 or 3 options - Demand, Allocation or Consumed.

These are, arguably, the most important policy changes you need to consider. We will look at each in turn. Before we do this, we first need to look at Capacity Buffers and Overcommit settings.
You can optionally turn on Network and Datastore I/O but given they are based on observed, as opposed to configured, values, they are probably best measured elsewhere, for example in the Storage or Network Management packs. The default of leaving them off is generally best.

vSphere Configuration Limit can be left on but will rarely have an impact on any of the calculations.

**Capacity Buffers**

Notice all the options have a 'Capacity Buffer' percentage. This can be thought of as a sort of 'synthetic' buffer for the administrator. As a default it is always set at 10% for every resource dimension.

**How does it work?**

Consider a 4-host cluster, each host has 100GB RAM installed (rare, but it makes the calculations easy!!)

We have a total of 400GB RAM available. If you use a capacity buffer of 10%, it masks 10% of the capacity from the capacity analysis calculation. It means you only have 360GB of **USEABLE** capacity and it will calculate your percentage usage based on this **useable** capacity.

If you are using conservative policies, such as Allocation, then a buffer is probably not needed. For a Demand based policy, having the 10% safety net for if you drive up utilization a bit far, is probably useful.
Overcommit

Now look at the Overcommit column. If you choose an 'Allocated' capacity measurement you can optionally add an overcommitment to your resources.

Consider your 4-host cluster, again with the 10% buffer. We were left with 360GB useable capacity. If you added an overcommit of, say 20% that increases the useable capacity value. 20% of 360GB is 72GB so you now have $360\text{GB} + 72\text{GB} = 432\text{GB}$ of Useable capacity when it comes to calculating capacity.

Memory

Let's look at what the three memory options are:

- **Memory Demand** - a measure of how many memory pages are active
- **Memory Consumed** - how much memory is shown as 'allocated' in vSphere. For a given VM this could be thought of as the 'high water mark' for used memory.
- **Memory Allocation** - how much memory has been actually allocated to the Virtual Machine Containers.

Remember - we are looking at a Cluster Compute object at the moment so this will be the total quantity of demanded, consumed and allocated memory for the Virtual Machines on the cluster.
Which box(s) should you 'check'?

It really depends on the environment and your attitude toward risk

- Memory Allocation - most appropriate for production environments where you currently don't overcommit memory resources
- Memory Demand - test or development environments where it doesn't matter too much if you become starved of resources on occasion
- Memory Consumed - also appropriate for production environments - probably the best middle ground

So, the default of just ticking Memory Consumed is fine, however, it's probably worth additionally selecting Memory Demand. This will highlight to you, in the Capacity analysis dashboards, how much memory is actually being used. You can start to close the gap between demand and consumed/allocated through rightsizing and eliminating waste.

If you select more than one model, then the capacity analysis will report on the most constrained model.

As you can see, there are many options for capacity analysis of memory.

**Selections for this lab**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Capacity Buffer %</th>
<th>Overcommit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Demand</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Memory Consumed</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Memory Allocation</td>
<td>10%</td>
<td>No</td>
</tr>
</tbody>
</table>

Check the **Memory Demand** box to add this measurement. We want to consider Memory Demand and Memory Consumed in our environment.

**CPU settings**

<table>
<thead>
<tr>
<th>Resource</th>
<th>Capacity Buffer %</th>
<th>Overcommit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Demand</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>CPU Allocation</td>
<td>10%</td>
<td>No</td>
</tr>
</tbody>
</table>

For CPU you have two options

- **CPU Demand** - how much CPU is being demanded as a percentage of the CPU that is available
• **CPU allocation** - how many vCPUs have been provisioned vs. how many physical CPU cores are available.

You should almost always stick to the default of CPU demand. vSphere is so good at time slicing CPU that your ratio of vCPU to CPU is just not relevant in the vast majority of cases. Probably the only time you would select this is if you have a system that absolutely must have a 1 to 1 mapping of vCPU to pCPU.

**vCPU to pCPU blog**

If you are interested in learning more about the relevance of vCPU to pCPU ratios, then this blog article describes it well!


**Disk Space settings**

<table>
<thead>
<tr>
<th>Disk Space Demand</th>
<th>10%</th>
<th>Not Applicable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space Allocation</td>
<td>10%</td>
<td>Yes, by percentage</td>
</tr>
</tbody>
</table>

For disk space you also have two options:

**Disk Space Demand** - use this if you use thin provisioning

**Disk Space Allocation** - use if you are using thick provisioning

*(note: if you use thick provisioning, Disk Space Demand will generally reflect allocated disk! To that end, you could probably still just use Disk Space Demand)*

**Why do we need to choose a different option if we are thin provisioning?**
Consider a cluster with 100 VMs, each with a 20 GB disk. You know they only need, and are using, 5GB (on average,) so you are using thin provisioning. You have 1000GB of disk.

With the allocation model, the capacity analysis will calculate you need 100 x 20GB = 2000GB and tell you that you are out of capacity as you only have 1000GB. It assumes that would want to know that you don't have enough to support your workload, as you have told it you need the allocated space.

The demand model will look at what you are actually using (5GB x 100 = 500GB) and tell you that you are at 50% of capacity. Plenty of disk space left!

**Disk selections for this lab**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk Space Allocation</td>
<td>10%</td>
</tr>
<tr>
<td>Disk Space Demand</td>
<td></td>
</tr>
</tbody>
</table>

Uncheck the Disk Space Allocation option. We'll leave the 10% buffer on for Disk Space Demand.

**Additional Settings**

- **High Availability** - you want Capacity Analysis to factor in your spare capacity reserved for HA. For example, with a 4-host cluster you may configure such that it can cater for 1 host to fail. It will mask out one of the hosts (25% of your capacity) and show USEABLE capacity as 75%
- **Peak Consideration** - you usually want to consider peak workloads - particularly in production environments.
- **Committed Projects** - if you are using the *vRealize Operations Capacity Projects* feature you probably want to include your capacity projects in the capacity analysis

*(You may need to scroll down to see the additional settings)*

The default additional settings are generally appropriate:
• **Capacity Calculation** - you usually want to base analysis on the current hardware you have, as opposed to a trend of the amount of hardware you have had over time.

• **Provisioning Time Buffer** - the **Time Remaining** calculations will add this value to the calculation to give you time to deploy resources. For example, if it calculates you will run out of capacity in 200 days, it will display 170 days in the UI if your buffer is set to the default of 30 days.

### Bringing it all together

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Demand</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Memory Consumed</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Memory Allocation</td>
<td>10%</td>
<td>No</td>
</tr>
<tr>
<td>CPU Demand</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>CPU Allocation</td>
<td>10%</td>
<td>No</td>
</tr>
<tr>
<td>Network I/O Data Transmit Rate</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Network I/O Data Receive Rate</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Network I/O Usage Rate</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Datastore I/O Outstanding I/O Requests</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Datastore I/O Reads per second</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Datastore I/O Writes per second</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Datastore I/O Read Rate</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Datastore I/O Write Rate</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>vSphere Configuration Limit</td>
<td>Not Applicable</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Disk Space Demand</td>
<td>10%</td>
<td>Not Applicable</td>
</tr>
<tr>
<td>Disk Space Allocation</td>
<td>10%</td>
<td>No</td>
</tr>
</tbody>
</table>

**Additional settings that affect time and capacity remaining calculations**

- **High Availability**
  - Use High Availability (HA) Configuration (Uses more capacity)

- **Peak Consideration**
  - Use Stress to account for spikes and peaks (May result in higher rightsize)

- **Committed Projects**
  - Affects capacity and time remaining scores

- **Capacity Calculation**
  - Current
  - Trend

- **Provisioning Time Buffer**
  - 30 days

In summary, there is a LOT to consider! Hopefully you have set the values for the Cluster Compute Resource object type as per the screenshot. The two changes made:

- Added the **Memory Demand** option so you can compare this value with Memory Consumed in the UI
- Removed the **Disk Space Allocation** option - you should almost always do this
Lets quickly look at the other policy elements

**Stress**

1. Scroll down and click on the arrow to expand the Stress section.
2. The default values are typically fine. Take a look at the panels on the right hand side for more detail on how Stress is calculated.

**Compliance, Reclaimable Capacity and Density**

The only setting you can change for Compliance is the badge threshold values.

**Click** on the arrow to expand the Reclaimable Capacity section. The default values are generally fine -
• **Flag as oversized at less than 25%** - this says that vRealize Operations will only consider an object oversized if it is 4x, or more, the size that vRealize Operations calculates it should be. This goes a long way to ensuring the objects flagged oversized really are much too large! There may be times you want to reduce this to 50% - this would flag an object as 'oversized' when it is twice the recommended size.

• **Flag as idle** - this considers the amount of time an object should be idle before it is flagged - a more conservative policy may up this value to, say, 97%

• Flag as **powered off** - this considers the amount of time an object should be powered off before it is flagged - a more conservative policy may up this value to, say, 97%

The only setting you can change for Density is the badge threshold values.

### Time Range

Finally, **scroll down to Time Range**

• **Track Usage** - this gives you the option to only apply analysis during, say business hours

• **Data Range** - the out of the box default is 30 days, and is generally fine for most environments. For this lab, we have increased the value to 180 days.

• **Maintenance Schedule** - if you have a regular Maintenance Schedule, you can add this to the policy to ensure it doesn't impact the calculations.

For this lab, there is no need to make any changes to the settings.

That concludes our view of the *Analysis* section of the *Default Policy*. So far we have only looked at the *Cluster Compute Resource* Object Type. You need to consider all the other Object Types:

• vCenter Server
• Custom Datacenter
• Datacenter
• Host System
• Datastore
• Virtual Machine

For the purposes of this lab, we are not going to look at them all in detail as it will take too long (and may become tiresome!) - we will show a couple of other things you need to check for though.

**Collapse the Cluster Compute Resource policy elements**

Scroll back to the top and click on the double arrow to collapse the Cluster Compute Resource section.

**Other Container Object Types**

The following Object Types have the same settings to change as the Cluster Compute Resource we just looked at:

- vCenter Server
- Custom Datacenter
- Datacenter
- Host System
The vCenter Server

1 - Click on the **double arrow** to expand the **vCenter Server** section

2 - Click on the **arrow** to expand the Capacity Remaining/Time Remaining section

You will see we have to consider the same things and make the same changes as we did with the Cluster Compute Resource. Notably, you will see if we scroll down that we need to uncheck the Disk Space Allocation option

For the purposes of this lab, there is no need to go and make the changes - feel free to explore the settings and options

3 - when you are done, click on the **double arrow** to collapse the vCenter Server section
1 - Now click on the **double arrow** to expand the **Datastore** section

2 - Click on the **arrow** to expand the **Capacity Remaining/Time Remaining** section

3 - Notice there are much fewer options - Datastores don’t have memory or CPU so these settings are not available to change. You still need to turn off **Disk Space Allocation** if you are using thin provisioning. Go ahead and **uncheck** that box.

4 - When you are done, click on the **double arrow** to collapse the Datastore section
Virtual Machine Policy

Edit Monitoring Policy

Now we need to change the Virtual Machine policy. If you notice that it is not available in the right hand pane, that is because no changes to its Base Settings have been made yet, so we need to add it. If you do see that the Virtual Machine policy is already available, you can skip ahead and click the double arrow to expand the policy.

1. Click on the arrow to expand the Show Changes for menu
2. Scroll down until you locate the vCenter Adapter - Virtual Machine object type
3. Click on the vCenter Adapter - Virtual Machine to select it
Add Virtual Machine

1. Click on the **Show object type** icon
2. The **Virtual Machine** Object Type will appear in the right hand panel
3. Click on the **double arrow** to expand the section

**Reclaimable Capacity**

For the Virtual Machine object type, there aren't really any Capacity Remaining changes to make. You do, however, want to make changes to the Reclaimable Capacity section, so scroll down and click on the Lock icon to unlock the section if you need to.
Idle machines and snapshots

1. Click on the arrow to expand the section
2. Notice the values for **Idle level**. These are the values that vRealize Operations will look for when considering a VM to be 'idle'. In some cases you may need to alter these (for example, you may have a background virus scan that means your idle VMs are actually consuming a bit more CPU)
3. The most common modification here would be to change the time at which snapshots are considered 'old'. The default of 180 days is far too long - most people would select a lower date - perhaps in the region of 3-7 days. Here we have already set it to **2 days** so you do not need to make any changes.
Alert Definitions

There is one more area of policy to look at in this section of this lab. Two sections we won’t look at are:

4. Workload Automation - This defines how workload placement and balancing will be used.

5. Collect Metrics and Properties - this is where you decide which particular Metrics and Policies you want to collect for each adapter type. You will certainly need to consider these if you are using End Point Operations.

The area we do want to look at is ‘Alert/Symptom Definitions’. Your policies are where you enable and disable the Alert Definitions for your environment. Most alerts are enabled by default - let’s look at how we can enable an additional alert in the Default Policy.

Click on the arrow to expand the section 6. Alert/Symptom Definitions

Add the vSphere Hardening Guide alert

First, we need to find the alert we are interested in. Using the filter box is the best way.

1. Click on the double arrow to bring up the filter box
2. In the box, type the word Hardening and hit return
1. Notice the current state is inherited. The Hardening Guide Alert has not been enabled in this policy.
2. If you hover over the alert you can see the full description. Click on the **ESXi Host is violating VMware vSphere Hardening Guide** alert to select it.

**Enable the alert**

1. On the Alert Definitions menu bar, click on **Actions**
2. Then **State**
3. Then click on **Enable** to enable the alert
Save your policy

1. Done! You can see the alert is now **enabled**. **Local** indicates that the alert is enabled within this policy.
2. Finally click on **Save** to save your amended Default Policy

You have successfully changed the Default Policy. Now let's look at how you can create additional policies and groups for specific objects.
Adding new Policies and Groups

In this lesson we will look at how you can create additional policies for specific parts of your environment, and apply them to groups of objects.

Setting the scene

1. Click on the Home icon to get back to the vRealize Operations home screen.
2. Click the Risk badge to display the Risk Status information.
Notice the **ESXi Hardening Guide alerts**

We activated this alert in the **Hands on Lab** policy remember?

We have 3 hosts in the lab and the alert has triggered on all the hosts - let's imagine we have a cluster of hosts that is less important, and on which we don't want to trigger our hardening alerts. To do this, we need to group together those objects and then apply a new policy to them.
Browse the environment

1. Click on the **Environment** tab
2. Click on **vSphere Hosts and Clusters** to view them
Expand the hierarchy

Click on the arrows next to vSphere World, then the vCenter, Datacenter and Cluster arrows to expand the view.

Even though we only have one cluster, we are going to create a group for the RegionA01-COMP01 cluster which will contain its Hosts and Virtual Machines.

Create a group

1. Click on the Environment tab again
2. Now click on the green plus icon to add a group
Naming the group

1. Give the group a name, for example **RegionA01-COMP01 Cluster**
2. Click on the arrow to expand the Group Type dropdown. Select the group type **Function** *(these group types can just be considered as folders for vRealize Operations custom groups)*
3. Check the **Keep group membership up to date** box as we want the group to be dynamic. This means if new members are added they will automatically join the group and have the policy applied.

Define membership criteria

The members of this group are going to be Hosts and Virtual Machines. Let's do the Hosts first...
1. In the 'Select the Object Type that matches...' drop down box type **host**
2. A list of matching Object Types will appear - click on **Host System** to select it

**Set the criteria**

### New group

**Name**
RegionA01-COMP01 Cluster

<table>
<thead>
<tr>
<th>Group Type</th>
<th>Function</th>
<th>Policy</th>
<th>☑ Keep group membership up to date</th>
</tr>
</thead>
</table>

**Define membership criteria**

1. Click on the arrow to display the membership criteria options. Then select the **Relationship** option.
Define the criteria

1 - Now we need to define the membership criteria. In order from left to right set the drop downs to:

- **Relationship**
- **Descendant of**
- **Is**
- **RegionA01-COMP01** (note as you type, a list of matching objects appears that you can select from)

You can leave the navigation tree blank

2 - Now click on 'Add another criteria set'
1. This time, in the 'Select the Object Type that matches...' drop down box type **Virtual Machine**
2. Again, we need to define the membership criteria. In order from left to right set the drop downs to:
   
   - **Relationship**
   - **Descendant of**
   - **Is**
   - **RegionA01-COMP01**
Preview the results

1. Click on the **Preview** button to preview the results, You will see the relevant Hosts and Virtual Machines
2. Click on **Close** to close the preview pane
3. Click on **OK** to save the group
Navigate to Policies

Now we need to navigate back to Policies to create our new policy - we will then apply this policy to the group we just created.

1. Click on the Administration tab
2. Select Policies
3. Select the Policy Library tab
4. Click on the Hands on Lab Policy - we are going to make our new policy a child of this policy. This means it will inherit all the values that we set earlier.
5. Click on the green plus icon to add the new policy
Give the Policy a name

1. Give the policy a name - for example **Exclude Hardening Guidelines**
2. Navigate to **6. Alert / Symptom Definitions** by clicking on the arrow
Filter for Hardening Guidelines

Now we need to find the Hardening Guidelines alerts. In the box, type **hardening**, then hit **return**

**Select the alert**

1. Notice that the alert is currently active (designated by the tick) and that is because it is Inherited from the parent policy.
2. Select the alert (it is highlighted blue when selected)
Change the state

Now we will disable the alerts.

1. Select the **Actions** drop down
2. Select **State**
3. Click on **Disable**

Review the changes
1. Notice the state has changed to a *disabled* icon and *Local* - this means it is disabled through the local policy.

2. Next we need to assign this policy to the group we just created. Click on the **Arrow** icon next to *8. Apply Policy to Groups*

### Add the group

<table>
<thead>
<tr>
<th>Name</th>
<th>Apply To Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>vSAN Datastores</td>
<td></td>
</tr>
<tr>
<td>Product Licensing</td>
<td></td>
</tr>
<tr>
<td><strong>RegionA01-COMP01 Cluster</strong></td>
<td></td>
</tr>
<tr>
<td>Non vSAN Datastores</td>
<td></td>
</tr>
<tr>
<td>Unlicensed Group</td>
<td></td>
</tr>
</tbody>
</table>

1. Check the box next to the **RegionA01-COMP01 Cluster** group we created
2. Click on **Save** to save your new policy
Review the results

Let's have a look at how this has changed one of our ESXi hosts.

1. Click the search icon in the top right hand corner, start typing the name of a server `esx-`
2. A list will start populating - click on `esx-02a.corp.local` - that was one of the hosts with the new policy we applied.

Look for the alert

1. In the top left hand corner you will see the panel context has changed to `esx-02a.corp.local`
2. In the top right hand corner you will see the new policy that is in effect - *Exclude Hardening Guidelines*
3. In the Alerts section you should no longer see the hardening guideline alert.
4. If you do see the alert still there wait a minute or so then click on the refresh icon - the alert will disappear quite quickly!
Summary

in this module we have looked at Policies. We looked at the Default Policy and some of the changes you should make so that capacity calculations are made correctly.

We then looked at how you can create a new policy to determine how particular parts of your infrastructure can be set to alert differently.

That concludes this module. You can now move onto any of the other Modules in this lab.
Module 4 - Working with views and reports (30 mins)
Introduction

This module will introduce you to Views and Reports in vRealize Operations.

We will look at the various options you have for creating Views and how they can be built into Report Templates.

This module will take about 30 minutes to complete.
Working with Views in vRealize Operations Manager

Your vRealize Operations Manager solution contains a huge amount of data. If you consider that every single object has all its metrics and properties collected every five minutes and stored for, by default, 6 months - there is a lot of data available to you.

vRealize Operations Manager creates additional metrics and properties for you to consume. For example, all the badge scores are available as metrics and stored with granularity.

Views

There are different ways you can access and consume this data:

- **Dashboards** - Module 2 of this lab cover creating custom dashboards.
- **Views** - a 'view' of data in a particular context.
- **Reports** - Reports are made up of Views and, optionally, Dashboards.
- **API** - you can programatically access the data through the API - this is looked at in Module 6 of this lab.

Lets start by looking at Views...
Open Firefox Browser from Windows Quick Launch Task Bar

1. Click on the **Firefox** icon on the Windows Quick Launch Task Bar.

HOL-1801 Lab Links Page

Once the firefox has been opened, the HOL-1801 Lab Links page should be loaded.

If not, please copy the link below to load the HOL-1801 Lab Links Page.

http://localhost or click the Home button in the browser.
Open vRealize Operations

1. Click on the vRealize Operations Manager - Live Instance link.

Select the VMware Identity Manager Authentication Option

1. Be sure to select the VMware Identity Manager authentication option from the drop down menu if it is not already selected.
2. Click the REDIRECT button

Login to vRealize Operations

1. Ensure that the displayed domain is corp.local. If a different domain is displayed, you can click on the Change to different domain link and then select the corp.local domain.

2. Login to vRealize Operations Manager with the following credentials. It is likely that the credentials have been auto populated.

   - User name: hol
   - Password: VMware1!

2. Click the Sign in button.
Unfortunately, the resolution we need to use in the Hands on Labs does not always allow you to complete some tasks or see enough of the screen. You may, therefore, need to adjust the browser zoom (resolution) on occasion. To do this:

1. Click on the **Open Menu** icon in Firefox.
2. Use the **minus** icon to reduce the Zoom (to 90% in this case) and the **plus** icon to increase it back as needed

Go ahead and practice this now! You will need to use this skill during the lab :-) Set the zoom to 90%.
Navigating to Views

Views are part of the Content in vRealize Operations Manager. Out of the box, the Management Packs will install a number of Views for you to use. To navigate to Views:

1. Click on the Dashboards menu
2. Click on the Views menu item
1. Clicking on the **Views** menu item will bring up the list of available Views in the right hand panel.

2. You will see there are already over 250 Views that have been installed with the Management Packs in place.

**Creating Your Own Views**
Let's look at how we can create some new Views of our own. In this first Scenario, we're going to create a View that looks at the NTP configuration of our environment.

Click on the green **plus** icon to create this new View.

### Name the View

1. Give the View a name, for example - **Hands on Lab NTP View**
2. Click on "**2. Presentation**"
1. This is going to be a list of properties, so click on the **List** menu option to select it.
2. Notice we get a configuration option when we click on List. We will leave the default of 50 items per page. This defines how the View will be presented in the UI.
3. Click on "3. Subjects" to move to the next section

Select Host System

1. This is where we select the Object Type(s) we want to use for this View. We want to look at the NTP configuration of our Hosts so type *host* in the Subjects box.
2. After a second or so, a list of options will appear - select the **Host System** option.
3. Now lets move the next section - click on "4. Data"
Selecting Properties

The NTP configuration data we are looking at is made up of Properties. Click on Metrics in the drop down and select Properties to change the selection tree to the Properties one.

Search for NTP
There are a lot of Properties available for Host Systems. It’s often easier to use a filter to find the ones you are looking for.

Type **NTP** in the filter box and hit **return**

**Open the Tree**

![Diagram of a tree structure with expandable nodes labeled 'Configuration', 'Security', 'Service', and 'NTP Daemon'.](image)

Click on the **arrow** icons next to **Configuration**, **Security**, **Service** and **NTP Daemon** to expand the tree so we can see the properties we are looking for.
Drag the NTP Server Across the Pane

Hands on Lab NTP View - New View

Now we need to add the properties to the Data box.

Click on **NTP Server**, and holding the mouse button down, drag the property into the Data box and release the mouse button.
Rename the Label

Let's give this property a meaningful name - type **NTP Source** in the *Metric Label* box.

Drag the Other Properties Over

Now let's drag the other two Properties into the Data Box:

- Drag the **Policy** property and give it a Metric Label of **Automatic Start?**
- Drag the **Running** property and give it a Metric Label of **Service Running?**
Change Visibility

Hands on Lab NTP View - New View

1. Click on "5. Visibility"
2. Check the Compliance box so we can add this View to the Compliance tab within Analytics.
3. Click on Save to save the View
Navigating to vSphere Hosts and Clusters by:

1. Click on the **Environment** menu option
2. Click on **vSphere Hosts and Clusters**
The vSphere World Context

The view we just built was to look at the NTP configuration of hosts in our environment. The context of this view can be any group or hierarchy of objects that contain Host Systems (the object type defined in vRealize Operations Manager for ESXi hosts).

1. Click on **vSphere World** as this will be our context for this view - all the ESXi hosts that exist in vRealize Operations Manager. By expanding the vSphere World hierarchy you can look at hosts in a specific vCenter or cluster, for example.
2. Click on **more** to reveal the hidden advanced tabs.
Examine the vSphere World Context

1. Click on the **Details** tab to bring up the list of Views for this context
2. Type **hands on lab** in the filter field and hit **return** to find the view you just created
3. Here we see the results - *All 3 hosts have the correct NTP source configured. In addition they all have the NTP service running and set to start automatically.*

Your view looks good!
The Compliance Tab

Do you remember we set the Visibility for this view? Let's go and look at the dashboard we enabled it on.

1. Click on **Analysis**
2. Click on **Compliance** *(you may need to scroll to the right using the arrow)*
3. Notice we have the **Hands on Lab NTP View** in the Further Analysis section - if you click on it, it will take you to the screen in the previous section.
A Metric View

Now let's construct a view to look at some metric data.

1. Click on the **Dashboard** tab and then **Views**
2. Click on the green **Plus** icon to create a new View.
Name the View

1. Give the View a name, for example - **Hands on Lab CPU Ready View**
2. Click on "**2. Presentation**"
Select List View

1. This is going to be a List of properties, so click on the **List** menu option to select it.
2. Notice we get a configuration option when we click on List. We will leave the default of 50 items per page. This defines how the View will be presented in the UI.
3. Click on "**3. Subjects**" to move to the next section.
Select Virtual Machine under the vCenter Adapter

Hands on Lab CPU Ready View - New View

1. This is where we select the Object Type(s) we want to use for this View. We want to look at the *CPU Ready time* of our Virtual Machines so type `virtual` in the *Subjects* box.
2. After a second or so, a list of options will appear - select the **Virtual Machine** option.
3. Now let’s move to the next section - click on "4. Data".
Find the Metric

There could be hundreds of metrics to choose from. The easiest way to find the one you are looking for is to use the filter option. Type `ready` in the filter box and hit `return`. 
Drag the Ready Metric

1. Click on the **Arrow** next to CPU to expand the tree. Now we need to drag the *Ready (%)* metric to the data panel. We are going to do this three times since we want to look at the metric in three different ways.
2. Click on **Ready (%)** and **holding** the mouse key down, **drag** the metric to to the data panel. **Repeat** two more times until you have three CPU | Ready (%) items in the box (in the screenshot you can see we are dragging the third instance)
Current CPU Ready

For the first instance, we want to get the current CPU Ready data - all we have to do is rename the metric for our view.

1. In the **Metric label** box, type **Current CPU Ready (%)**.
2. Notice the field name changes as you type.

Max CPU Ready

1. **Click** on the 2nd instance of the metric
2. Change the Metric Label to **Max CPU Ready (%)**
3. Now we want to transform the data - click on **Last** in the Transformation drop down to view the options

_Last means the 'last value of the metric collected - i.e. the current value - that is the default transformation we used for the first instance of this metric_
Select the Maximum Transformation

Move your mouse up and click on **Maximum** to select this option.

**Average CPU Ready**

1. **Click** on the 3rd metric instance to select it.
2. Change the Metric label to **Average CPU Ready (%)**
3. Change the transformation to **Average**
4. Now we need to change the time range for our transformed values. We'd like to see the Maximum and Average CPU Ready for the last 90 days. Click on **Time Settings**.
Relative Date Range

1. Change the value in relative date range to **90 Days**.
2. Next we are going to add a Summary row - click on the **Summary** button to bring up the summary panel.

Add a Summary Row

1. Click on the green **Plus** icon to add the summary row.
2. Change the **Summary title** to **Maximum**.
3. In the **Aggregation** drop down, change the option to **Maximum**.
Preview the View

We've done quite a lot with this View, we should check to see if it does what we want before we save it!

Click on Select preview source
Select vSphere World

1. Click on **vSphere World** to select it
2. Click on **OK** to save
Save the View

Click on **Save** to save your view.
Navigate to the View

To navigate to vSphere Hosts and Clusters, do the following.

1. Click on the **Environment** tab
2. Click on **vSphere Hosts and Clusters**
Viewing the View!

The view we just built was to look at the CPU Ready metric of virtual machines in our environment. The context of this view can be any group or hierarchy of objects that contain virtual machines (the object type defined in vRealize Operations Manager for virtual machines)

1. Click on **vSphere World** as this will be our context for this view - all the virtual machines that exist in vRealize Operations Manager. By expanding the vSphere World hierarchy you can look at virtual machines in a specific vCenter or cluster, for example.
2. Click on **more** to reveal the hidden advanced tabs.
Our context for this view will be vSphere World

1. Click on the **Details** tab to bring up the list of Views for this context.
2. The **hands on lab** filter should still be there and the new view we created will be selected.
3. **Scroll down** so you can see the **Summary** line we created.
4. The summary line shows the maximum value for each of the columns.
Creating a Trend View

Let's create a *Trend* view to look at the CPU ready trend. Click on the green **Plus** icon.
1. Give the view a name, **Hands on Lab - CPU Ready (%) Trend**
2. Click on "2. Presentation"
This time select the **Trend** view type. We will leave the default of 25 plot lines.

Click on "**3. Subjects**"
This is where we select the Object Type(s) we want to use for this View. We want to look at the CPU Ready time for our Virtual Machines so type virtual in the Subjects box.

2. After a second or so, a list of options will appear, select the Virtual Machine option.
3. Now let's move the next section - click on "4. Data"
Find the Metric

There could be hundreds of metrics to choose from. The easiest way to find the one you are looking for is to use the filter. Type **ready** in the filter box and hit **enter**.
Drag the Ready Metric

Hands on Lab - CPU Ready (%) Trend - New View

1. Click on the **arrow** sign next to **CPU** to expand the tree. Now we need to drag the **Ready (%)** metric to the data panel.
2. Click on **Ready (%)** and holding the mouse key down, **drag** the metric to to the data panel.
3. The default for this view is to trend the previous 7 days. Depending on when we are taking this lab we may need to go back further. Since this lab was built in June/July 2017, this is where there is CPU Ready Data that we created! So, click on the **Time Settings** tab to change out time range as required.
Change the Date Range

Now change the Relative Date Range depending on when you are taking this lab. If you are taking the lab in June 2017, you do not have to make any changes and can use the default settings.

If you are taking the lab after June 2017, follow the instructions below to select the proper date range.

Change the drop down to Months

Change the Last number to reflect the time elapsed since June 2017 as follows:

- If taking this lab in July 2017 change the value to 2 Months (as in the screenshot)
- If taking this lab in August 2017 change the value to 3 Months
- September 2017 - 4 months
- October 2017 - 5 months
- November 2017 - 6 months
- December 2017 - 7 months
- January 2018 - 8 months
- February 2018 - 9 months
- March 2018 - 10 months
- April 2018 - 11 months
- May 2018 - 12 months
- June 2018 - 13 months
- July 2018 - 14 months
Save the View

Click on **Save** to save the view.
The Trend View for virtual machines in vSphere World

1. The view can be seen and displays the **CPU Ready (%) trend** for all virtual machines in the environment. This is because the selected object is vSphere World.
2. Click on the **left** of the **timeline** to display the trend information in list format.
3. Click on the **right** of the **timeline** to display a 5 day trend forecast for the virtual machines in the environment.

The Trend View for an individual virtual machine
1. Click the search icon at the top right of the screen and type **web-02a** in the search box
2. Click to select the virtual machine **web-02a**

**The Trend View**

1. Click on the **Hands on Lab - CPU Ready (%) Trend** view.
2. The view now displays the trend for **web-02a** and on the right hand side you should see a short dotted line representing the next 5 day's trend. Likely this will be flat since the collected previous CPU Ready Time can clearly be seen as very temporary peaks.
Navigate back to vSphere World Context

Click on **vSphere World** to change the context back

**Distribution Views**

Click on the **X** to clear the Hands on lab filter
OS distribution

There are a huge amount of distribution views already created; let’s look at one.

1. Change the filter to **OS name** and hit return. You will see the Virtual Machine Guest OS Name pie chart. This represents the proportion of each OS in the environment.

2. An easy way to create your own content is to clone the existing content and change it. Click on the **Clone View** icon.
## Rename the view

### Hands on Lab - Oversized Virtual Machines

<table>
<thead>
<tr>
<th>1. Name and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> Hands on Lab - Oversized Virtual Machines</td>
</tr>
<tr>
<td><strong>Description:</strong> Distribution of oversized Virtual Machines</td>
</tr>
</tbody>
</table>

1. Provide a new name and description - **Hands on Lab - Oversized Virtual Machines**
2. We are just going to change the data element so click on **"4. Data"**
Delete the Existing Metric

In the data panel, click on **Remove** to remove the existing metric.

Find and Drag the Oversized Metric

1. In the filter box, enter **oversized** and hit return.
2. Click on the **arrow** icon to expand the **Summary** section
3. Drag the **Is Oversized** metric to the data panel

**Remove the Operating System metric and change the Label**

1. Click **Remove** to **delete** the **Operating System** metric as the distribution view supports only one metric
2. Change the Metric label to **Oversized VMs**
3. Click on **Save** to save the new view.

**Clear the Filter**

Click on the **X** to clear the OS name filter
Add **Hands on Lab** to the filter field, here you can see your new distribution view!

![Filter Screenshot]

**Add the Hands on Lab Filter**
Creating a Report from our Views

Now that we have built our Views, let's go ahead and create a Report.

Creating the Report

1. Click on the **Dashboards** tab
2. Click on **Reports**
3. Click on the green **plus** icon to create a new Report Template
Name the Report Template

Hands on Lab Report - New Temp

1. Give the Report Template a name - Hands on Lab Report
2. Click on 2. Views and Dashboards
Find the views

1. Type **hands on lab** in the filter box and hit **return** to filter
2. Notice the Data Type drop down set to views - you can also add dashboards - we won't do this in this lab

Drag the NTP View
Holding the mouse button down, click on **Hands on Lab NTP View** and drag it into the **Views and Dashboards** panel. Release the mouse button.

**Drag the CPU Ready View**

Holding the mouse button down, click on **Hands on Lab CPU Ready View** and drag it into the Views and Dashboards panel **below** the NTP view. Release the mouse button.

*(Take care to release the mouse button in the small area beneath the NTP view - the screen resolution makes this area a little small)*
## Collapse the first CPU Ready View

### Hands on Lab Report - New Template

Click on the **double up** arrows to collapse the CPU Ready report - this will make it easier to complete the next step.
Drag the Oversized Virtual Machine vView

Holding the mouse button down, click on **Oversized Virtual Machine View** and drag it into the **Views and Dashboards** panel below the CPU Ready view. Release the mouse button.

*(Take care to release the mouse button in the small area beneath the CPU Ready view - the screen resolution makes area a little small)*
Check Formats

Hands on Lab Report - New Temp

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Name and Description</td>
<td>✔</td>
</tr>
<tr>
<td>2. Views and Dashboards</td>
<td>✔</td>
</tr>
<tr>
<td>3. Formats</td>
<td>✔</td>
</tr>
<tr>
<td>✔ PDF</td>
<td>Good for sharing with people who need to print it out.</td>
</tr>
<tr>
<td>✔ CSV</td>
<td>Good for exporting data to be used by other data analysis applications.</td>
</tr>
</tbody>
</table>

Click on 3. Formats - notice both PDF and CSV are enabled.

Add options

Hands on Lab Report - New Temp

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Name and Description</td>
<td>✔</td>
</tr>
<tr>
<td>2. Views and Dashboards</td>
<td>✔</td>
</tr>
<tr>
<td>3. Formats</td>
<td>✔</td>
</tr>
<tr>
<td>4. Layout Options</td>
<td>✔</td>
</tr>
<tr>
<td>✔ Cover Page</td>
<td></td>
</tr>
<tr>
<td>✔ Table of contents</td>
<td></td>
</tr>
<tr>
<td>✔ Footer</td>
<td></td>
</tr>
</tbody>
</table>

Click on 4. Layout Options
Check the three boxes for **Cover Page**, **Table of contents** and **Footer**

### Save the Report Template

![Save button]

Click on **Save** to save the Report Template

### Run the Report

![Image of report templates]

Your new report will be highlighted in blue.

Click on the **Run template icon** to run the report
Select the object

1. We want to run this against the vSphere World - click on vSphere World to select it
2. Click on OK to run the report
Review the report

Click on **Generated reports** to review the report

**Refresh if needed**

The report may take a minute or two to process - click on the **refresh** button to refresh the screen

**Download the report**
Click on the PDF icon to download the report

Open the report

1. Select **Open with** in the dialog box
2. Click on **Ok** to view the report
Browse the report

The report will open in the default PDF reader application. You can scroll up and down to see the results of the Views you have created.

This concludes this module - we hope you enjoyed it!
Module 5 - Using the API for inbound integration (30 mins)
Introduction

In this module we will explore the APIs and integration tools offered with vRealize Operations Manager. We will discuss and demonstrate how to export data, as well as how to create custom agents and automate administrative tasks.

Intended audience: This is an advanced topic. Although the lab has been designed so that anyone with a basic knowledge of vRealize Operations can follow the instructions and successfully complete it, some knowledge of REST, JSON and basic script programming will be very beneficial to fully taking advantage of the content.

What is a REST API?

A REST API is a method of programatically interacting with a piece of software. It uses the HTTP or HTTPS protocol and mimics the way a web browser accesses pages on the web. The underlying programming model is based on the concept of a document, very similar to how you would view pages with your web browser. Each document has a unique URL and can be retrieved using a web browser simply by entering that URL into the address field. The payload is typically XML or JSON.

However, the HTTP/HTTPS protocol also allows you to perform updates to documents. Each time you access a URL, you do so using what's known as a "method". When you load a web page, your web browser uses the "GET" method. But GET is only one of a handful if methods available. The most important ones are these:

- GET - Retrieves a document
- POST - Creates a new document
- PUT - Updates a document
- DELETE - Deletes a document

But what is a "document" when we are not talking about web pages, but programmatic interactions with some software? Typically, API designers map the underlying objects in their software to documents. In vRealize Operations, for example, documents can be VMs, adapter instances and user records. Using the verbs described above, we can now treat these as documents and retrieve and manipulate them.

How REST APIs are used in vRealize Operations

So what can we do using REST APIs in vRealize Operations? Much of the functionality that you can perform from within the vRealize Operations UI can also be done via the API. Some functions (like formatting email notifications) can be done via the API but have no corresponding UI functionality. Of course, you can use the APIs to retrieve information about your resources as well as the full set of metrics. Since we can do both read and write operations, we can also use the API to create new resources and to post metrics for new or existing resources. But it doesn't stop there. A lot of the
administrative functionality is also exposed through the API. For example, you can start and stop adapters as well as manipulating users and groups. Throughout this lab, you are going to see examples of many of these interactions in this lab module.
A tour through the REST API

Let's take a look at the vRealize Operations REST API.

**Start Firefox Browser**

1. Click the "Firefox icon" in the System tray to start the browser window.

**Say hello to the REST API!**

Let's dive straight into the API and see what it has to offer! A nice thing about it is that it is completely contained within the vRealize Operations virtual appliance. There's no additional software to download or install and all the documentation is included.

To access the API, simply follow these steps:
1. Type [https://vrops-01a.corp.local/suite-api/](https://vrops-01a.corp.local/suite-api/) in the address field and press the "Enter key".

Note: Rather than typing you can drag and drop the link to the browser address field.

You should see a page similar to the screen shot above.

From this page you can browse the documentation or download language bindings. Although the REST API is very easy to work with, language bindings make it even easier, since they hide all the details of the protocol and API and let you interact using local function calls instead. Currently, language bindings are available for Java and Python. Later in this lab, we are going to explore the Python bindings in more detail.

**An overview of the API components**

Let's have a look at the API!

1. Locate the link labeled "Click to view our complete REST API documentation" and click on it.

A page with a list of API functions on the left-hand side should open up.
You are now looking at the full API. On the left hand side you can see the sections of the API (such as /api/adapters) along with all the functions they expose.

1. Scroll the left-hand list to the /api/adapters section.
2. Click on the "enumerateAdapterInstances"

You should see something similar to the screen shot above. **You may have to scroll the right-hand pane upwards a little to get to the "GET /api/adapters" section.**

As you can see, the "enumerateAdapterInstances" function maps to the HTTP operation of "GET /api/instances". In other words, if you ask for that document, you should get a list of adapter instances in our vRealize Operations instance.

If you like, you can click around in the list of API functions to explore what you can do or you can move straight ahead to the next step where we're going to try to call the "enumerateAdapterInstances" function.
Making our first API call

Let's try it out! As we mentioned in the previous step, enumerateAdapterInstances maps to the GET /api/adapters HTTP interaction and should give us a list of adapters in our system. The easiest way to try this out is to go to a web browser and type in the URL. The API has the root URL of https://hostname/suite-api, so to access the enumerateAdaptersInstances, we'd have to use the URL https://vrops-01a.corp.local/suite-api/api/adapters

In the browser:

1. Type https://vrops-01a.corp.local/suite-api/api/adapters in the address field and press the "Enter key".

Note: Rather than typing you can drag and drop the link to the browser address field.

2. A login window will pop up. Enter "admin" as username and "VMware1!" as password.
3. Click "OK" to login.
Analyzing the results

The default content type for the API is XML, so that’s what we’re seeing here. If you set a Content-Type header of "application/json" you would get the results as JSON. Let’s have a quick look at what we’re seeing here.

Your output may look different from the screenshot, but you are looking at a list of adapters in vRealize Operations. In our output, the first ones are the Log Insight Adapter and a vRealize Cluster Node adapter pointing to a ClusterID. You will also see some attributes of the adapters, such as the timestamp for the last heartbeat and the last collection.
Working with resources and metrics

Let's take a look at how to use the REST API to work with resources (objects) and metrics.

Let's face it, the reason you're taking this lab is probably because you want to export or import metrics of some sort. So let's get to that!

If you took some time to browse the documentation for the API in the earlier exercise, you probably noticed a very large section of API functions under the `/api/resources` sections. That's the place to go for functions dealing with resources, attributes and metrics. Let's take a quick look at the basics!

**Understanding the resources API**

The objects under management in vRealize Operations are called "resources". This includes virtual machines, hosts, clusters, datastores and any object you put under management using third-party tools, such as application servers and databases. Each resource has a few common properties:

- A name. This is the human readable name of the managed object, such as "vm-123", "production-cluster-1" or "InvoiceDB".
- A unique ID. This is a UUID generated by the system and used as a static and stable identifier of an object. This is what is typically used for fetching resources and linking them together.
- A resource identifier. This is a set of "primary keys" uniquely identifying an object. A resource identifier is different from the unique ID in that it is generated outside of vRealize Operations and typically has some meaning in the monitored environment. For example, a resource identifier for a vSphere VM has a MOID and a BIOS id, which not only can be used to identify the object in vRealize Operations but also at its origin (vCenter).
- (Optional) a set of properties. These are fairly static properties of an object, such as OS-version, CPU-version, serial numbers etc.
- (Optional) a set of metrics. These are the dynamic properties of an object, such as current CPU utilization, current memory utilization and latencies. They are always stored as a time series of double-precision floating point numbers.

In addition to this, resources also have a collection of links that allows you to navigate and get more information about an object. For example, you can use links to list the related resources or drill deeper into metrics.
Starting Postman

For the next few exercises, we are going to use a tool called Postman, which is a powerful HTTP client for testing web services. Let's start by starting up Postman on our desktop.

1. Double-click the "Postman" icon on the desktop

After a short while, the Postman application should open.
Looking up a resource

Let's look up a resource and find out some information about it! For this we're going to use a name-based query. As you will see, you can do this simply by adding a query string to the end of the URL.

In Postman:

1. On the left-hand side, there's a list with two headers labeled "History" and "Collections". Select "Collections". A Postman collection is a stored list of API calls that you can create or modify. In this case, we have provided some API calls for you in the vRealize Operations collection.
2. Click on "Lookup Resource" in the list of REST operations. Examine the URL shown on the right hand side. Notice the "?name=app-01" query at the end. This returns a list of all resources containing the "app-01a" string. In our lab environment, that will be a Virtual Machine object, a Linux object and an EP Ops Agent object.
3. Click on "Send" to the right of the URL.

Examine the result. First of all, notice that the result is now expressed using JSON rather than XML. This is because we're sending a Content-Type header containing "application/json" as part of the API call that was stored in the collection.
Understanding links

Links allow you to drill deeper into a resource and get, for example, additional properties, metrics and related objects. Let's follow a link that gives us the names of all the metrics recorded by this object.

1. In the result you obtained in the previous step, scroll down a bit until you see the "links" section. This section allows you to navigate to get further details about a resource. In the "links" section there is a URL for the "statKeysOfResource". We can use the search tool to find the correct URL.
Search Links

1. Click the "Magnifying Glass" to start a Search.
2. Type "statKeysOfResource" in the search bar as shown.
3. Click the Link above the statKeysOfResource name. Note: The link in your lab may be different than what is shown above.
Authentication settings

1. Click on the **Authorization** tab.
2. In the **Type** dropdown, select Basic Auth.
3. Click the **Update Request** button.
4. Click on the **Headers** tab.

Note that the Authorization key has been added to the header along with the password stored for this saved environment.
Fix the URL and Set The Content Type

In our Postman instance, we have a saved environment that includes a {{server}} variable. That variable string is https://vrops-01a.corp.local/suite-api/api. Since the link that we clicked on in the previous step already includes the "/suite-api/api" portion of the URL, we need to delete it from this URL before we send the request. We could have configured the {{server}} variable to only include the server name and then we would not have had to edit the URL here.

1. Highlight the /suite-api/api text in the URL and press the Delete key to remove it from the URL string. Make sure that the URL now begins: {{server}}/resources/...
2. Add the following Keys and Values (you can start typing the text then click the appropriate Key or Value from the resulting list):
   - Content-Type, application/json
   - Accept, application/json
3. Click Send.
4. You should now see a list of metric keys that are available for this resource.

Creating a new resource

So far, we've discussed how to get various information about a resource. But how can we create a new resource?

Let's imagine the following scenario: In our data center, we have large central power supplies that supply various voltages to our equipment. We're interested in building custom adapter for vRealize Operations that feeds metrics about these power supplies, such as voltages and currents. To do this, we'd have to introduce a brand new resource type we call "Power Supply" and programmatically create instances of it. Let's give it a try.

We need three pieces of information:
• The name of the adapter type this object comes from. Since there's no standard Power Supply monitoring in vRealize Operations, we have to make up our own. Let's call it "PowerMonitor".
• The name of the resource type. In other words, a string the determines what kind of object this is. Let's call that "PowerSupply".
• The name of the resource itself. This is intended to identify a specific instance of a resource. Let's call ours "Main 110V Supply".

Our resource is simple enough that it can be uniquely identified by those items and no additional identifiers are needed. So let's go ahead and create it!

**Create the Resource in Postman**

Let's create a new resource.

1. Click on **Create Resource** in the collection.
2. Click on the **Body** tab as indicated in the screen shot above. This shows you the payload we're sending with the API call. You'll also notice that the request method is POST, which typically indicates we're creating something new based on the payload. Notice how we're simply sending in the adapter type, resource type and resource name.
3. Click the **Send** button.
4. The lower section of the window should now be populated with the full details of the new resource we created.

(Optional) If you like, you may open the vRealize Operations web UI and do a search for the "Main 110V supply" resource. This should show you a resource with no metrics. It may take a couple minutes before the newly created resource can be found in vRealize Operations.

A couple of interesting things happened, aside from creating the resource. A brand new resource type called "Power Supply" was also created. This type will now be available for dashboarding, alerting, super metrics and everything else you can do with a resource type. The system also automatically created the "PowerMonitor" adapter type for us.

DO NOT close this Create resource tab in Postman.

**Adding metrics to a resource**

Let's continue building our power supply monitoring solution! So far we've successfully created a resource, but it's not a very interesting resource, since it lacks metrics. So we need to figure out a way to post metrics to a resource. Luckily, the REST API makes this very easy for us. All we have to do is to identify the adapter type and ID of the object we want to post metrics for and do a HTTP POST to its URL, along with a payload message. Let's have a look at the payload format for adding metrics:

```
{
    "stat-content": [ {
        "timestamps": [ 1465477762000 ],
        "statKey": "voltages|output",
        "data": [ 12.0 ]
    }
    ]
}
```

This happens to be the JSON version of a payload. All you need to know about JSON is that the symbol "{" starts a new record and "[" starts a new array. So we have a complex type called stat-content. That's simply the container for all our metrics. Next, you'll see that it has an array of records inside of it. Each record represents a metric and its values. The `statKey` attribute contains the name of the metric and `data` and `timestamp` represent the value of the metric and when that value was recorded. You'll also notice that `data` and `timestamps` are themselves arrays, meaning that you can send multiple readings on a time series with a single call. The `timestamp` value is what's known as a UNIX-timestamp which is simply the number of milliseconds since 1/
1/1970 12:00AM UTC. Most programming languages provides an easy way to obtain such a timestamp.

Let's try this in our lab!

First we need to generate the current timestamp for our metric.

**Start the Powershell ISE**

In order to get the current time in the format expected by vRealize Operaitions, we will execute a simple PowerShell script.

1. Click the Windows Start key and move your mouse over *Windows PowerShell*.
2. Click *Run ISE as Administrator*.

![Screenshot of PowerShell ISE](image)

---

**Note:** The screenshot shows the Windows PowerShell ISE interface, which is used to run PowerShell scripts. To execute the script, you need to switch to the ISE by selecting it from the Start menu and then run it as an administrator. This ensures that you have the necessary permissions to access the system's time and date properties.
Get The Current Time

To get the current timestamp enter the script in the PowerShell Window. This script gets the current epoch date/time on the local server in seconds, gets the offset of the current server from UTC because the vRealize Operations server is set to the UTC time zone, then combines them to create the timestamp in the proper format for generating the metric.

Rather than typing the text you can highlight the text and drag it to the PowerShell input section.

1. Type in (or select here and drag to) the upper pane:

```powershell
$here = [int][double]:Parse((Get-Date -UFormat %s)) * 1000
$offset = [int][double]:Parse((Get-Date -UFormat %Z)) * 3600 * 1000
$here - $offset
```

2. Click the Run Script icon to execute your simple script.
3. Highlight the resulting timestamp value and use the CTRL-C keyboard combination to copy the timestamp to the Windows clipboard. Be sure to get all the numbers.
Build the POST API Call

Go to the Postman window you left open in the previous steps.

1. Make sure Collections is selected as shown above.
2. Click on Add metrics under Collections.
3. Click on the Body tab on the right-hand side as shown in the screen shot. Note that this saved API call contains an old timestamp and a resource ID that won't correspond to the new resource that you just created.
4. Locate the timestamp section of the payload as shown in the screen shot. Use the backspace key to erase all the numbers on the line under timestamp. Use the Ctrl-V keyboard combination to paste the number from the clipboard that you generated in the previous step.
Get the Resource ID For the New Power Supply Object

We need to get the ID of our newly created power supply object.

1. Return to the "Create resources" tab in Postman where you created the "Main 110V supply" resource. Click the Create resource tab.
2. Find the resource ID for that newly created resource from the SELF link in the response body of the API post. Select the ID and use the CTRL-C keyboard combination to copy the string to the Windows clipboard. Be sure to get all the characters.
Complete the API Post

Let's complete the API call and send it to vRealize Operations.

1. Return to the "Add metric" tab to finish building the API call.
2. Select the old resource ID from the POST command, delete it and then use the **CTRL-V** keyboard combination to paste in the new resource ID string that you just copied to the Windows clipboard.
3. Click **Send**.

If the call returned an empty result without any error messages, your Post was successful.

Log In To vRealize Operations Manager UI

If you are already logged into the vRealize Operations Manger from a previous module, skip ahead to [here](#).

Otherwise proceed to launch and log into vRealize Operations Manager.

Open Firefox Browser from Windows Quick Launch Task Bar

1. Click on the **Firefox** icon on the Windows Quick Launch Task Bar.
Once the firefox has been opened, the HOL-1801 Lab Links page should be loaded. If not, please copy the link below to load the HOL-1801 Lab Links Page.

http://localhost or click the Home button in the browser.

Open vRealize Operations Manager UI

1. Click on the vRealize Operations Manager - Live Instance link.
Select the VMware Identity Manager Authentication Option

1. Be sure to select the **VMware Identity Manager** authentication option from the drop down menu if it is not already selected.
2. Click the **REDIRECT** button
Login to vRealize Operations

1. Ensure that the displayed domain is corp.local. If a different domain is displayed, you can click on the Change to different domain link and then select the corp.local domain.

2. Login to vRealize Operations Manager with the following credentials. It is likely that the credentials have been auto populated.

   - User name: hol
   - Password: VMware1!

2. Click the Sign in button.

Verify That the Object and Metric Values Were Created

Now that we have created a new object (the power supply) and added a metric to it using the API, let's verify everything.
Search For the New Object

![Image of the vRealize Operations UI with search bar open]

Return to the vRealize Operations UI.

1. Click the magnifying glass icon to open the search bar.

Search For The Power Supply Object

![Image of the vRealize Operations UI with search bar open and power supply object selected]

1. Type **Main 110** to search for that string.  
2. Click **Main 110v supply** to select the object.

A summary page will be displayed for the resource.
View the Metric Graph

1. Click the **All Metrics** tab.
2. Click "v" to expand the All Metrics category.
3. Click "v" to expand the voltages metric category (we created this when we added the metric value using the API).
4. Double-Click the **output** metric to see a graph of the metric values.
5. Note the dot representing the single metric value of 12 volts that we just added using the API.

Lesson Conclusion

In this lesson we used Postman to explore some of the available actions in the vRealize Operations API, created a new resource and added a single metric value to the resource using the API and then verified the creation of the resource and metric in the vRealize Operations UI.
Conclusion

In this module we have explored the APIs and integration tools offered with vRealize Operations Manager. We discussed and demonstrate how to export data, as well as how to create custom agents and automate administrative tasks.

You've finished Module 5

Congratulations on completing Module 5.

If you are looking for additional information on [subject], try one of these:

- Click on this link
- Or go to http://tinyurl.com/nxqxca
- Or use your smart device to scan the QR Code.

Proceed to any module below which interests you most. [Add any custom/optional information for your lab manual.]

- **Module 1 - Operating system and application monitoring with Endpoint Operations** (60 minutes) (Advanced) Explore the End Point Operations (EPOps) adapter in vRealize Operations by installing a small agent inside the guest OS that allows you to expose OS-specific metrics, such as OS CPU usage, memory usage and swapping.
- **Module 2 - Creating custom dashboards** (30 minutes) (Advanced) Experiment with creating a custom Dashboard using super metrics. Super Metrics are metrics that aren't directly collected from an underlying collector, but calculated from other metrics. This type of metric is sometimes also known as a synthetic metric or a derived metric.
- **Module 3 - Managing policies** (30 minutes) (Advanced) Introduction to Policies in vRealize Operations. Understanding how policies work and what they do is critical in ensuring vRealize Operations operates in your environment in the way you want it to.
- **Module 4 - Working with views and reports** (30 minutes) (Advanced) Introduction to Views and Reports in vRealize Operations by looking at the
various options for creating Views and how they can be built into Report Templates.

- **Module 6 - Using webhook shims for outbound integration** (30 minutes) (Advanced) This lab module will take you through an example of a self healing datacenter. For this lesson we will be stopping a web service and through the use of vRealize Operations, webhook using REST API and Orchestrator, the service will be started automatically.

- **Module 7 - Using PowerCLI for vRealize Operations** (30 minutes) (Advanced) This module introduces PowerCLI which contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vCloud Director, vCloud Air, vSphere Update Manager, vRealize Operations Manager, and VMware Horizon administration. VMware PowerCLI provides a PowerShell interface to the VMware product APIs.

**How to End Lab**

If you wish to conclude your lab at this time click on the **END** button. This will terminate your lab and all progress. Do this only if you wish to **NOT** proceed with the other modules.
Module 6 - Using Webhook shims for outbound integration (30 mins)
Introduction

Would you like to extend your vRealize Log Insight and vRealize Operations alerts to be used to automatically remediate the identified issue? Or perhaps open a ticket in ServiceNow. How about sending notifications to ChatOps? These and more are possible with Webhook Shims, an open source solution to empower VMware Intelligent Operations with practically unlimited functionality.

This lab module will take you through an example of a self healing datacenter. For this lesson we will manually stop a web service and then using the resulting alert in vRealize Operations, launch a workflow in vRealize Orchestrator via the webhook shim to restart the service.
Self Healing DataCenter Example

We will start by looking

Launch and Log In to vRealize Operations Manager

Start Firefox Browser

1. Click the Firefox icon in the task manager to open the browser.

vRealize Operations Lab Links

1. Click the link "vRealize Operations Manager - Live Instance"
1. If VMware Identity Manager shown here is not the default, you can change it by clicking the "v"
2. Click "REDIRECT" to take you to the VMware Identity Manager Login screen.
Sign in to Identity Manager

Notice the user "HOL" for the domain "corp.local" is already entered.

1. Click **Sign In**

Gain Screen Space in vRealize Operations

1. Open the Firefox menu found in the upper right hand corner.
2. Click the "-" twice to set the display to 80%. This will provide you more viewing space while still allowing you to read the text.
Understanding The Lab Environment

1. Click **Environment** to navigate to the vRealize Operations Manager Environment Overview.

Expand Hosts and Clusters

1. Click the **"vSphere Hosts and Clusters"** to view the environment for the Hosts and Clusters.
Environment of Hosts and VM's

The virtual machines for this lesson are the "Photo-OS-Container-Host" and the "web-01a". To see the hierarchy for these VM's you need to click as shown to expand the tree.

1. Click the "v" symbols to expand the tree until you see the virtual machines that are running on the host "esx-01a.corp.local"

Photon-OS-Container-Host
1. Click the **Photon-OS-Container-Host**. This virtual machine contains a python script (the webhook shim) which will be listening for REST notifications from vRealize Operations Alerts then reformattting the information and calling the vRealize Orchestrator API. vRealize Orchestrator will then run a workflow to restart the service.

**web-01a Virtual Machine**

1. Click the "**web-01a**" virtual machine. The summary page is shown.
2. Click the "**MultiProcess (1)**" tab. This shows you that there is one OS process being monitoring using the Endpoint Operations. In this example we are monitoring the httpd process running on this web server.
3. Currently there are no alerts for the httpd process and the health status is green because the process is running. Let's verify that the process is running in the VM's console.

**Verify The Web Service Status**

Launch PuTTy in order to be able to SSH into the Linux servers.

1. In the taskbar, click the PuTTy **icon**.
Launch a PuTTY Session to the web-01a VM

To start a putty session to the web-01a VM:

1. Using the scroll bar, scroll down to the bottom of the list of saved sessions.
2. Double-click on web-01a.corp.local to launch the session.

Verify http process is running

Once the console is open. We can check the process is running by entering:

1. Type `service httpd status` and click Enter. Verify that the httpd service is running.
2. Click the "-" to minimize the window since we will need this session later in the lesson.

**Start the Webhook Shims**

1. Right-Click on "root@web-01a" putty session. This will bring up a menu.
2. Click on "PuTTY" to start a new session.

**PuTTY session for Photon-OS-Container**

1. Using the **Scroll bar** find "photon-os-container-host.corp.local"
2. Double-Click "**photon-os-container-host.corp.local**" to open a console session.
Start Docker Container

In the console session we need to start the container where the python script resides. These components will be listening for vRealize Operations alert REST notifications then translating the alert message and making an API call to vRealize Orchestrator to start the remediation workflow.

Note that this container with the webhook shims can be downloaded from https://github.com/vmw-loginsight/webhook-shims.

1. Type `docker start webhook` This command starts the container called webhooks which contains the python script.
2. Type `docker ps -a` This command will return the results of the containers running on the Photon-OS-Container VM. The only container running is the "webhook" container shown at the end of the line.
3. Type `docker exec -it webhook /bin/bash` This command will open a bash shell in the container. You are now logged into the console of the container running within the VM.
4. Type `python runserver.py` to start the listening script.
Verify That the Webhook Shims Are Running

To test that the webhook shims and the listener is working is to open a browser and enter the name of the photon VM and the port.

1. Click the "+" to open a new tab in the browser.
2. Type `photon-os-container-host:5001` The listener of the webhook container responds with documentation of all the shims available. If you would like to see which shims are available scroll down the documentation page. We will be using the "vro()" shim in this lab to call a vRealize Orchestrator workflow.
3. Minimize the Firefox browser.

Explore the Workflow in vRealize Orchestrator

1. Double-click the **vRealize Orchestrator shortcut** on the Desktop to launch the client.
vRealize Orchestrator Login

1. The username and password are both "vcoadmin".
2. Click "Login"
Viewing Workflows

1. Click the **Workflows icon** to view the workflows for this lesson.

**Restart Service from Alert Workflow**

1. Expand the folder "**Webhook Shim Sample**" folder.
2. Click the workflow "**Restart Service from Alert**" to see more detail.
3. Notice the long ID of the workflow ends in "0015". This long number is used in vRealize Operations as part of an Outbound Instance configuration. We will get to that in a couple steps.

**Schema**

When the httpd service is stopped on web-01, a vRealize Operations Alert is generated. The Alert is sent to the webhook shim which then sends it to vRO. For the VRO workflow to successfully restart the httpd service it needs 3 pieces of information:

- The Alert ID
- The Resource Id
- The Resource Properties

1. Click the "Schema" tab to see what steps comprise the workflow.
2. Slide the Scroll Bar to the right to see all the steps for the workflow. The last step of this workflow is a callout to the "SSH Restart httpd" workflow
1. Click the "SSH Restart httpd" workflow. The schema is displayed showing the steps which will be run.
vRealize Operations Configuration

Return to the vRealize Operaations UI in the Firefox browser.

1. Click "Administration"
2. Click on "Management" to expand the menu
3. Click on "Outbound Settings" to view the instances that have been created.
4. Click on "vRealize Orchestrator Remediation"
5. Click the "Pencil icon" to edit the instance.
vRealize Orchestrator Remediation

1. Move your mouse to the right hand edge of the dialog box then click and drag the side so you can see the entire URL string.
2. Notice the URL is pointing to the Photon-OS-Container. Remember this is listening for outbound messages from vRealize Operations.
3. The other thing to notice is the hex string which ends in 0015 is the ID of the "Restart Service from Alert" workflow in vRealize Orchestrator.
4. Click "CANCEL" since there are no changes needed.

Search

To get back to the web-01a summary page lets use the search tool.

1. Click the Search icon found in the upper right corner of vRealize Operations.
1. In the Search window begin typing `web-01a`. vRealize Operations will show objects containing that string.
2. Click `web-01a.corp.local` located under the Linux heading. You may have to scroll down the list for the Linux objects.
3. Note: This is the refresh icon and will be helpful for updating the dashboard once the service has been stopped.

**Web-01a.corp.local Summary**

This is the summary page of the web-01a.corp.local Linux operating system that is being monitored by the Endpoint Operations agent. The "children" of this operating system object are the Endpoint Operations Agent and MultiProcess. The MultiProcess
children will be any services that are configured to be monitored by the Endpoint Operations agent.

1. Click **MultiProcess (1)** to see how many processes are being monitored in this operating system - in this case one.

**MultiProcess screen**

1. It takes about one minute in our lab environment for the alert to appear in the dashboard when it gets triggered. It will appear in the area highlighted above. Refreshing the screen is sometimes helpful. Remember this is the icon in the upper right hand corner.

**Alert Trigger and Remediation**

Return to the consoles stop the httpd service to trigger the alert and monitor the progress of the webhook shim process.

1. Click the photon-os-container... PuTTY session.
2. Click the root@web-01a... PuTTY session.
Stop httpd Service

If you arrange your windows similar to what is shown you can stop the httpd service, see the photon-os-container and web-01a dashboard alert area.

1. Type `service httpd stop` in the web-01a console session to stop the web server service.
Alert

View the vRealize Operations Manager UI. Click the refresh icon every periodically until you see that the alert has triggered. It can take a minute or two for the alert to trigger because vRealize Operations alerts are evaluated on a periodic cycle. The default evaluation cycle is once every five minutes. In the lab environment, we have configured the cycle for once every minute.
When the alert triggers, it will send the Rest notification to the webhook shim.

1. Notice that photon-os-container-host console is showing it received a alert message regarding HTTPD. This input is then sent to vRealize Orchestrator.

vRealize Orchestrator Client

Switch to the vRealize Orchestrator window.
1. Click to Expand the "**Restart Service from Alert**". Notice there is a green check for a operation involving the workflow "Restart Service from Alert". The green checkmark indicates a successful run of the workflow.

2. Click the "**Events**" tab. Notice there are three elements executed with within the workflow.

### Web-01a.corp.local Dashboard

![Web-01a.corp.local Dashboard](image)

Switch back to the vRealize Operations session and refresh the UI again.

Notice that Summary page of "web-01a.corp.local" is once again showing a Green Health status because the web service is once again running. If the alert still shows as active, continue to refresh the UI periodically until the alert is gone.

### HTTP Service for web-01a

```bash
[root@web-01a ~]# service httpd status
httpd (pid 4451) is running...
[root@web-01a ~]#
```

To further verify, switch to the `web-01a` console session.

1. Type `**service httpd status**` and notice the service is running again.
Conclusion

In this module you went through an example of a self healing datacenter. This lesson stopped a web service and through the use of vRealize Operations, webhook shims using the REST API and vRealize Orchestrator, the service was restarted automatically.

You've finished Module 6

Congratulations on completing Module 6.

Proceed to any other module which interests you most.

- **Module 1 - Operating system and application monitoring with Endpoint Operations** (60 minutes) (Advanced) Explore the End Point Operations (EPOps) adapter in vRealize Operations by installing a small agent inside the guest OS that allows you to expose OS-specific metrics, such as OS CPU usage, memory usage and swapping.
- **Module 2 - Creating custom dashboards** (30 minutes) (Advanced) Experiment with creating a custom Dashboard using super metrics. Super Metrics are metrics that aren’t directly collected from an underlying collector, but calculated from other metrics. This type of metric is sometimes also known as a synthetic metric or a derived metric.
- **Module 3 - Managing policies** (30 minutes) (Advanced) Introduction to Policies in vRealize Operations. Understanding how policies work and what they do is critical in ensuring vRealize Operations operates in your environment in the way you want it to.
- **Module 4 - Working with views and reports** (30 minutes) (Advanced) Introduction to Views and Reports in vRealize Operations by looking at the various options for creating Views and how they can be built into Report Templates.
- **Module 5 - Using the API for inbound integration** (30 minutes) (Advanced) In this module we will explore the APIs and integration tools offered with vRealize Operations Manager. We will discuss and demonstrate how to export data, as well as how to create custom agents and automate administrative tasks.
- **Module 6 - Using webhook shims for outbound integration** (30 minutes) (Advanced) This lab module will take you through a example of a self healing...
datacenter. For this lesson we will be stopping a web service and through the use of vRealize Operations, webhook using REST API and Orchestrator, the service will be started automatically.

- **Module 7 - Using PowerCLI for vRealize Operations** (30 minutes)  
  (Advanced) This module introduces PowerCLI which contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vCloud Director, vCloud Air, vSphere Update Manager, vRealize Operations Manager, and VMware Horizon administration. VMware PowerCLI provides a PowerShell interface to the VMware product APIs.

### How to End Lab

If you wish to conclude your lab at this time click on the END button. This will terminate your lab and all progress. Do this only if you wish to NOT proceed with the other modules.
Module 7 - Using VMware PowerCLI for vRealize Operations (45 mins)
Introduction to VMware PowerCLI

VMware PowerCLI contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vCloud Director, vCloud Air, vSphere Update Manager, vRealize Operations Manager, and VMware Horizon administration. VMware PowerCLI provides a PowerShell interface to the VMware product APIs.

- PowerCLI is based on Microsoft PowerShell and uses the PowerShell basic syntax and concepts.
- PowerCLI cmdlets are created to automate VMware environments administration and to introduce some specific features in addition to the PowerShell concepts.

Microsoft PowerShell Basics

PowerCLI is based on Microsoft PowerShell and uses the PowerShell basic syntax and concepts.

Microsoft PowerShell is both a command-line and scripting environment, designed for Windows. It uses the .NET object model and provides administrators with system administration and automation capabilities. To work with PowerShell, you run commands, named cmdlets.

- **PowerShell Command-Line Syntax** - PowerShell cmdlets use a consistent verb-noun structure, where the verb represents the action and the noun represents the object to operate on.
- **PowerShell Pipelines** - A pipeline is a series of commands separated by the pipe operator `|`.
- **PowerShell Wildcards** - PowerShell has a number of pattern-matching operators named wildcards that you can use to substitute one or more characters in a string, or substitute the complete string.
- **PowerShell Common Parameters** - The Windows PowerShell engine retains a set of parameter names, referred to as common parameters. All PowerShell cmdlets, including the PowerCLI cmdlets, support them.

PowerCLI Concepts

PowerCLI cmdlets are created to automate VMware environments administration and to introduce some specific features in addition to the PowerShell concepts.

- **PowerCLI Modules** - VMware PowerCLI 6.5.1 consists of multiple modules that you can install and use according to your needs and environments.
- **Interoperability Between the PowerCLI and vCloud Director PowerCLI Modules** - With the RelatedObject parameter of PowerCLI cmdlets, you can retrieve vSphere inventory objects from cloud resources. This interoperability
between the PowerCLI and vCloud Director PowerCLI modules expands cloud administration, automation, reporting, and troubleshooting options for provider administrators.

- **Selecting Objects in PowerCLI** - In PowerCLI, you can pass strings and wildcards to all parameters that take inventory objects, datastores, OSCustomizationSpec objects, and VIserver objects as arguments. This PowerCLI approach is named Object-by-Name (OBN) selection.

- **Providing Login Credentials** - When you provide login credentials in the command prompt or in a script, a PowerShell limitation might prevent PowerCLI from processing non-alphanumeric characters correctly. To prevent login problems, escape the non-alphanumeric characters in your credentials.

- **Running PowerCLI Cmdlets Asynchronously** - By default, PowerCLI cmdlets return an output only after completion of the requested tasks. If you want a cmdlet to return to the command line immediately, without waiting for the tasks to complete, you can use the RunAsync parameter.

- **Managing Default Server Connections** - By default, PowerCLI and PowerCLI cmdlets run on the vCenter Server systems or vCloud Director servers you are connected to, if no target servers can be determined from the provided parameters.

- **Customization Specification Objects in PowerCLI** - PowerCLI provides two types of objects for customization specification: persistent and nonpersistent.

- **Using ESXCLI with PowerCLI** - PowerCLI provides you the capability to use ESXCLI through its console.

- **PowerCLI Inventory Provider** - The Inventory Provider is designed to expose an unfiltered inventory view of the inventory items from a server.

- **PowerCLI Datastore Provider** - The Datastore Provider is designed to provide access to the contents of one or more datastores.

- **PowerCLI About Articles** - You can learn more about some PowerCLI concepts and features from the built-in help articles named about articles. You can access them through a running PowerCLI process.
Basic Windows PowerShell usage and VMware PowerCLI for vRealize Operations cmdlets

In this lesson you will learn:

- The basics of Windows PowerShell and VMware PowerCLI
- How to start the Windows PowerShell command prompt and Windows PowerShell ISE and get basic version information
- How to use Windows PowerShell ISE interface
- How to execute VMware PowerCLI cmdlets and get help on using them
- What can be done with the VMware PowerCLI module for vRealize Operations

Starting Windows PowerShell and checking VMware PowerCLI version information

In vSphere 6.5 you can install PowerCLI by running a Windows PowerShell command. You can install all official modules with a single command, or install modules individually.

The PowerCLI modules are available on the PowerShell Gallery Web site. When you run Install-Module from the Windows PowerShell prompt, the command downloads and installs the specified module. For a list of available PowerCLI modules, see the PowerShell Gallery Web site.

There is no need to follow this procedure since it was already done in the lab for you.

1. Click on the Windows PowerShell icon on your desktop taskbar
Check PowerCLI Version

Note: The modules have already been installed for you in the lab environment so you DO NOT NEED TO COMPLETE THIS STEP.

1. Type the following cmdlet to check the VMware PowerCLI version information:

   Get-PowerCLIVersion

Ignore the warning about deprecated cmdlet. Since we just want to know the version of the VMware PowerCLI framework this command should be just fine.

All of the Windows PowerShell commands and parameters can be auto-completed with the TAB key. Just start typing the first letters of the command and/or parameter and press the TAB key for auto-completion.
Starting Windows PowerShell ISE

As an alternative to the simple command prompt you can use Windows PowerShell ISE script editor, which provides a better user experience.

1. Type in `ise` and hit the **enter key** at the Windows PowerShell command prompt to launch ISE script editor.
Using Windows PowerShell ISE

Once the Windows PowerShell ISE is launched you will see two panes by default, the **Script Pane** on the top and the **Command Pane** on the bottom. You can choose whether to show the **Command Add-on window** or not.

1. The **Script Pane** is for viewing/editing script files in the text form;
2. The **Command Pane** which is the actual Windows PowerShell command prompt is for running individual cmdlets and displaying their output;
3. Click on the **Show Command Add-on** icon to display it;
4. In the **Command Add-on** window you can search for a specific module, their commands and their respective parameters.
Executing VMware PowerCLI cmdlets in Windows PowerShell ISE

During this lab you will be executing VMware PowerCLI cmdlets using Windows PowerShell ISE Script Pane. The advantage of using the Script Pane is the ability to run multiple commands at the same time since it's a script! A script can be executed by pressing the F5 key or by clicking on "Run Script (F5)" icon. Feel free to use the Command Pane and the ENTER key instead but remember that with this method you will be able to run only one command at time. Before executing a new script make sure the previous one completed successfully (you will see "Completed" status at the bottom of the Windows PowerShell ISE interface).

Listing the available cmdlets

![Image of PowerShell ISE interface with cmdlet listing]
Let's explore what we can do with VMware PowerCLI. PowerCLI's snapins provide more than 500 cmdlets for managing vSphere, vCloud Air, SRM, vROps, and VUM. You can view the available PowerCLI cmdlets by typing "Get-VICommand"

You can change how the panes are displayed in the Windows PowerShell ISE interface. You can resize the panes as well as hide the Command Add-on window to make the interface looks a bit cleaner. Feel free to customize it as much as you want until you are comfortable with.

This will list all PowerCLI commands. As the list is quite large, you may want to narrow it down to something more specific, for example to list the commands related to VMs:

1. In the **Script Pane** type "Get-VICommand *VM".
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script.
3. As a result, you will see all the cmdlets containing the word "VM" listed in the **Command Pane**.

All of the Windows PowerShell Commands and Parameters can be auto-completed with the TAB key. Just start typing the first letters of the command and/or parameter and press the TAB key for auto-completion.
Getting help on using VMware PowerCLI cmdlets

It is possible to get help on how to use any cmdlet and their respective parameters in Windows PowerShell using a cmdlet called "Get-Help". Since our focus is on VMware PowerCLI let's do this using the cmdlet "Get-VM" as parameter.

1. In the **Script Pane** type "Get-Help Get-VM";
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script (If you receive a prompt about downloading the last available documentation choose "No" because we don't have internet connection in the labs);
3. You will see the help content for this particular cmdlet in the **Command Pane**.
What can be done with the VMware PowerCLI module for vRealize Operations

VMware PowerCLI includes a module for interacting with vRealize Operations. Let's see what are the available cmdlets to manage a Realize Operations environment.

Import the vROps PowerCLI Module

To import the vROps PowerCLI module:

1. Type `Import-Module -Name "C:\Program Files\WindowsPowerShell\Modules\VMware.VimAutomation.vROps"` in the PowerShell command prompt.
2. Click on **Run Script (F5)**.
Checking the version information for the vRealize Operations module

1. In the **Script Pane** type "Get-Module -Name VMware.VimAutomation.vROps";
2. Click on "**Run Script (F5)" icon or press the F5 key to execute the script;
3. Check the module version information in the **Command Pane**.
Checking the available cmdlets in the vRealize Operations module

1. In the **Script Pane** type "Get-Command -Module VMware.VimAutomation.vROps";
2. Click on "Run Script (F5)" icon or press the **F5** key to execute the script;
3. Check all the available cmdlets within the module in the **Command Pane**.

You can see that there is a function called "**Get-vROpsCommand**" in the listed cmdlets. That function has the same effect as the previous command and simplifies the listing of available vROps cmdlets so you can use it at anytime without having to write the whole syntax over and over.
Connecting to vRealize Operations with VMware PowerCLI

In this lesson you will learn how to connect to a vRealize Operations server using VMware PowerCLI.

You will be using a vRealize Operations server with a database containing historical data on it.

Connect to vRealize Operations server

```
Connect-Observer -Server vrops-hwm -User admin -Password VMware!
```

![Image of PowerShell interface with Connect-Observer command executed]

1. Connect-Observer command
2. Enter server name
3. Enter user and password

Example output:
```
Name    | User | AuthSource
-------|------|-----------
vrops-hwm| admin|----------
```
vRealize Operations server

Server: vrops-hvm
User: admin
Password: VMware1!

1. In the **Script Pane** type "**Connect-OMServer** -**Server** vrops-hvm -**User** admin -**Password** VMware1!";
2. Click on "**Run Script (F5)**" icon or press the **F5** key to execute the script;
3. Check for any errors in the **Command Pane** to see if the connection was successful.
Using VMware PowerCLI to work with vRealize Operation Alerts

In this lesson you will learn:

- How to list the active critical alerts that are impacting Health
- How to list the existing alert types and subtypes
- How to list the available alert definitions for a specific type and subtype
- How to show detailed information about a specific alert definition
- How to show alert definitions from a specific resource kind
- How to store an alert instance into a PowerShell variable to explore other details of the alert
- How to take ownership and suspend an alert stored in a PowerShell variable

Listing active critical alerts impacting health

1. In the **Script Pane** type "Get-OMAlert -Status Active -Criticality Critical -Impact Health";
2. Click on **"Run Script (F5)"** icon or press the **F5** key to execute the script;
3. Check the list of active critical alerts in the **Command Pane**.
If you take a good look on the previous script result you will see that there are columns for **Type** and **Subtype** which can be used as input parameters for the `Get-OMAlert` cmdlet, yet there are cmdlets provided for those specific parameters (**Get-OMAlertType** and **Get-OMAlertSubtype**). Using those cmdlets without input parameters returns a list of all valid types and subtypes on the server.

1. In the **Script Pane** type "**Get-OMAlertType**";
2. Click on "**Run Script (F5)**" icon or press the **F5** key to execute the script;
3. From the list of alert types in the **Command Pane** you can see in the column "**Name**" that there are five of them: **Application Alerts**, **Virtualization/Hypervisor Alerts**, **Hardware (OSI) Alerts**, **Storage Alerts** and **Network Alerts**.
List the available subtypes of a specific alert type

Let's take the alert type Virtualization/Hypervisor Alerts from the previous step to list their available subtypes.

1. In the Script Pane type "Get-OMAlertSubType -AlertType 'Virtualization/Hypervisor Alerts'";
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
3. From the list of alert subtypes in the Command Pane you can see in the column "Name" that there are also five of them: Availability, Performance, Capacity, Compliance and Configuration.
Listing alert definitions for a specific type and subtype

In addition to type and subtype, you can retrieve alerts using the AlertDefinition parameter and the cmdlet Get-OMAlertDefinition can be used to find the available alert definitions in the system. You can filter the output to show alert definitions of a given alert type and subtype.

Let's take the alert type Network Alerts and alert subtype Configuration from previous steps to list their alert definitions.

1. In the Script Pane type "Get-OMAlertDefinition -Type 'Network Alerts' -SubType Configuration";
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
3. From the list of alert definition in the Command Pane you can see in the column "i" several definitions for the Configuration subtype of Network Alerts type.
Alert definitions contain a lot of information that may be helpful. Here we going to show the output of the cmdlet using the `-Name` parameter with a wildcard. Let's take the first definition on the previous list “There is an MTU..” for this example.

1. In the **Script Pane** type 
   ```powershell
   Get-OMAlertDefinition -Name 'There is an MTU*' | Format-List;
   ```
2. Click on "Run Script (F5)" icon or press the **F5** key to execute the script;
3. You can see in the **Command Pane** all details about this alert definition.

Note the values for **AdapterKind** and **ResourceKind** properties. These can be used as input parameters as well and that is what we going to do next.
List alert definitions from a specific resource kind

Let's take the resource kind listed on the previous step, "VmwareDistributedVirtualSwitch".

1. In the **Script Pane** type "Get-OMAlertDefinition -ResourceKind VmwareDistributedVirtualSwitch";
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
3. In the **Command Pane** you can see all the alert definitions for "Distributed Virtual Switch" object as well as their type, subtype, criticality, impact, wait cycle and cancel cycle.
Searching for a specific VM to use as parameter for the alert variable

It is possible to pull a single alert instance into a variable so we can explore other details of the alert. Information on the status, event times and control state are available as well as other useful information.

First, let’s search for all resources that has **Active** alerts impacting on **Risk** and choose one of them to pass as a parameter for our PowerShell script.

1. In the **Script Pane** type "Get-OMAlert -Impact Risk -Status Active;"
2. Click on "Run Script (F5)" icon or press the **F5** key to execute the script;
3. From the listed alerts in the **Command Pane**, choose a specific resource to use as a parameter for our script. In this example let's choose "t-win2k12" object which is a VM.

![Image of PowerShell interface showing script execution and command pane selection]

It is possible to pull a single alert instance into a variable so we can explore other details of the alert. Information on the status, event times and control state are available as well as other useful information.

First, let’s search for all resources that has **Active** alerts impacting on **Risk** and choose one of them to pass as a parameter for our PowerShell script.

1. In the **Script Pane** type "Get-OMAlert -Impact Risk -Status Active;"
2. Click on "Run Script (F5)" icon or press the **F5** key to execute the script;
3. From the listed alerts in the **Command Pane**, choose a specific resource to use as a parameter for our script. In this example let's choose "t-win2k12" object which is a VM.
Store an alert instance into a PowerShell variable to explore other details of the alert

Now let's write a script that stores the active alert impacting on risk for "t-win2k12" object and then show its details in formatted list.

1. In the **Script Pane** type

```powershell
$alert = Get-OMAlert -Resource 't-win2k12' -Impact Risk -Status Active
$alert | Format-List
```

2. Click on **"Run Script (F5)"** icon or press the **F5** key to execute the script;
3. Check the details of the alert in the **Command Pane**.

Note that we used the **"Format-List"** parameter after a pipe to indicate to PowerShell that we want to see the list of all properties of this particular alert. Try an execute the
same script without the "**Format-List**" parameter and you will see that it will return only the main properties of the alert.

**Taking ownership and suspending an alert stored in a PowerShell variable**

![Image of PowerShell ISE with Set-OMAlert cmdlet]

Using the **Set-OMAlert** cmdlet I can take or release ownership of an alert, suspend an alert for a period of time (in minutes) or cancel the alert. For example, let's take ownership and suspend the alert we stored in "$alert" variable from the previous step.

We are assuming you still have your Windows PowerShell ISE opened and that the variable from the previous step is still valid. If the variable is not valid anymore re-run the previous script.

1. In the **Script Pane** type "**$alert = Set-OMAlert $alert -TakeOwnership -Confirm:$false**" and hit **ENTER**;
2. In the next line of the Script Pane type "$alert = Set-OMAlert $alert - SuspendMinutes 60" and hit ENTER;
3. In the next line of the Script Pane type "$alert | Format-List";
4. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
5. In the Command Pane, check the information about ownership and suspension of the alert.

A couple of things to note for this cmdlet is that the ownership for the TakeOwnership property assigns the currently connected user as the owner. Also, the example above shows the optional "-Confirm:$false" parameter to skip any confirmation prompt but there is also a WhatIf parameter to display the changes that would be made but not commit them.
Using VMware PowerCLI to work with vRealize Operations Recommendations

In this lesson you will learn:

• Listing specific alerts for a chosen VM
• How to list recommendations for a specific alert stored in a PowerShell variable
• How to apply recommendations for a specific VM
• List the current alerts of the VM after applying the recommendation
• How to retrieve metric data (or stats) within a specific period of time

In this lesson we are going to apply some recommendations on a specific VM. Until now we were only working with the cmdlets from vReralize Operations module, since we are going to work with PowerCLI VI module cmdlets to perform operations on a VM, in addition to connecting to a vRealize Operations server we will also need to connect to a vCenter server.

The vRealize Operations we were using was the one with historical data and for this reason, the vCenter Servers connected to it do not exist in our lab environment, therefore we are going to switch to the live instance of vRealize Operations.

Listing specific alerts for a chosen VM

Get the specific alert and retrieve the chosen VM name

As we did in the previous lesson, we are going to store an alert instance into a variable in order to manipulate its properties.

First, let’s search for all resource objects that has **Active** alerts impacting on **Health** and with **Immediate** criticality. We are going to use that resource object to pass as a parameter to our script in the next step.
1. In the **Script Pane** type "**Get-OMAlert** -**Impact Health** -**Status Active** -**Criticality Immediate**";
2. Click on "**Run Script (F5)**" icon or press the **F5** key to execute the script;
3. From the listed alerts in the **Command Pane**, choose a specific resource to use as a parameter for our script. In this example let's choose "**web-01a**" object which is a VM.

![PowerShell window showing script execution](image)

### Storing the specific alert for the chosen VM in a PowerShell variable

Now let's write a script that stores the active alert impacting on health for "**web-01a**" object and then show its details in a formatted list.

1. In the **Script Pane** type "$alert = Get-OMAlert -Resource 'web-01a' -Impact Health -Status Active -Criticality Immediate" in the first line and hit **ENTER**;
2. In the next line of the **Script Pane** type "$alert | Format-List";
3. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
4. Check the details of the alert in the Command Pane.

Note that we used the "Format-List" parameter after a pipe to indicate to PowerShell that we want to see the list of all properties of this particular alert. Try an execute the same script without the "Format-List" parameter and you will see that it will return only the main properties of the alert.

Listing recommendations for a specific alert stored in a PowerShell variable

As part of an alert which we have previously discussed, vRealize Operations provides recommendations that guide you to possible solutions. These recommendations are available as well via the Get-OMRecommendation cmdlet.

1. In the Script Pane type "Get-OMRecommendation -Alert $alert | Format-List";
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
3. From the Command Pane you can see that the recommendation is to add a new virtual hard disk to the virtual machine or to expand the existing one.

Applying recommendations for a specific VM
Getting the VM resource object from the alert properties and storing in a PowerShell variable

We can apply the recommendation we got from the previous step - add a new virtual hard disk - from PowerCLI but to do this we need the VM object. Happily, the vRealize Operations resource object is provided as a property of the alert. We simply need to retrieve it and here we see it contains the Managed Object Reference ID also know as MOID or MoRefID that uniquely identifies an object inside the virtual infrastructure.

1. In the Script Pane type "$vmresource = $alert.Resource" in the first line and hit ENTER;
2. In the next line of the Script Pane type "$vmresource | Format-List";
3. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
4. Check the details of resource object in the Command Pane and look for the MOID of the VM.

We can apply the recommendation we got from the previous step - add a new virtual hard disk - from PowerCLI but to do this we need the VM object. Happily, the vRealize Operations resource object is provided as a property of the alert. We simply need to retrieve it and here we see it contains the Managed Object Reference ID also know as MOID or MoRefID that uniquely identifies an object inside the virtual infrastructure.
Getting the actual VM object from the resource and storing in a PowerShell variable

Even better, you can simply use the **Get-VM** cmdlet from PowerCLI VI module with the vRealize Operations resource to retrieve the VM object. We can also use the **Get-HardDisk** cmdlet from PowerCLI VI module to list the VM disks and choose the one we want to expand (in this case there is only one but it could have more).

1. In the **Script Pane** type "**$vmobj = Get-VM $vmresource**" in the first line and hit **ENTER**;
2. In the next line of the **Script Pane** type "**Get-HardDisk -VM $vmobj | Format-List**" and hit **ENTER**;
3. Click on **"Run Script (F5)"** icon or press the **F5** key to execute the script;
4. Check the details of resource object in the **Command Pane** and look for the name and capacity of the disk.
Note the we have two properties for capacity, one in **Gigabytes** and the other one in **Kilobytes**. To make things easier we are going to use the capacity in Gigabytes.

**Getting the hard disk from the VM object and storing in a PowerShell variable**

Now that we have the name of the disk we want to expand we need to create a script to store the disk in a variable so we can expand it using the **Set-HardDisk** cmdlet also from the PowerCLI VI module. We are going to expand it from 2.5GB to 5GB.

1. In the **Script Pane** type "\$vmhd = Get-HardDisk -VM $vmobj -Name 'Hard disk 1'" and hit ENTER;
2. In the next line of the **Script Pane** type "Set-HardDisk -HardDisk $vmhd -CapacityGB 5 -Confirm:$false" and hit ENTER;
3. In the next line of the **Script Pane** re-type "\$vmhd = Get-HardDisk -VM $vmobj -Name 'Hard disk 1'" so we can refresh the variable with the new value and hit ENTER;
4. In the next line of the Script Pane type "$vmhd | Format-List";
5. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
6. Check the details of the disk in the Command Pane and look how the capacity has changed from 2.5GB to 5GB.

Expanding the disk inside the Guest OS

Connecting to the VM via SSH

In the previous steps we expanded the "web-01a" VM hard disk to apply a vRealize Operations recommendation from an alert but that changed only the size of the disk from the perspective of the virtual machine hardware. In order to have the alert gone we need to expand the disk inside the virtual machine to reflect the change in the guest OS level.

1. Click on the putty icon in the desktop taskbar;
2. Type the **web-01a** in the Host Name field of the **Putty** client;
3. Click on the **Open** button. If you get the **POTENTIAL SECURITY BREACH!** warn from **Putty** just ignore it and click **Yes**.

Rebooting the VM to reflect the new disk size

---

1. Once in the login prompt of "**web-01a**" VM type **root** as the username and hit **ENTER**. Since the lab uses SSH public key to authenticate in the VM you will not have to provide any password;
2. At the command prompt type **"fdisk -l"** to show the existent disk/partitions and their respective sizes;
3. Note that the disk still reports the old size of **2.5GB**. Let's reboot the server so the OS can re-scan their disk and reflects the newest size we set from previous steps;
4. At the command prompt type **"reboot"** and hit **ENTER**. Close the dead **Putty** window and wait for a few minutes until the server comes back. Once the server is back repeat the procedure to re-open the **Putty** and SSHd into "**web-01a**" VM.
Re-check the new disk size

1. Back in the login prompt of "web-01a" VM type root as the username and hit ENTER;
2. On the command prompt type "`fdisk -l`" again to re-check the existent disk/partitions and their respective sizes;
3. Note that the disk now shows the correct size of 5GB;
4. We can also note that this machine uses a Linux LVM type of partition for its root partition with its logical volume group named as "VolGroup-lv_root".
5. As we can see, the size of this LVM partition approximately 2GB and that is because we only extended the physical disk.
6. In order to expand the partition with the capacity we have added we need to know what is the hexadecimal code for Linux LVM partition type. We can see that this code is 8e.
Creating a partition to accommodate the new unallocated capacity

1. At the command prompt type "df -h" and hit ENTER;
2. As you can see the disk information is showing that our initial setup only has the one 2GB disk currently, which is under the logical volume named "/dev/mapper/VolGroup-lv_root". Note that "/dev/mapper/VolGroup-lv_root" is the volume currently made up from "/dev/sda1". This is what we will be expanding with the new disk;
3. First we need create a new primary partition on "/dev/sda/" to make use of the new expanded disk space. Note that we do not have 4 primary partitions already in place but only 3, making this method possible. On the command prompt type "fdisk /dev/sda" and hit ENTER;
4. Type "n" to create a new partition;
5. Type "p" to define it as a primary one;
6. Since we already have 3 primary partitions, `fdisk` chooses the number 4 automatically for this new primary partition;

7. Just press **ENTER** twice as by default the first and last cylinders of the unallocated space should be correct. After this the partition is then ready;

8. Type **"t"** to change the partition type;

9. Type **"4"** to select our new partition;

10. Type **"8e"** to select the **LVM** **linux** type. You will see that the partition type has now changed.

11. Type **"w"** to write the changed table to disk and exit from `fdisk` utility;

12. As you can see, there is a warning reporting that the **"Device or resource is busy"** and that the system still uses the **old partition table**. This is an expected behavior and to use the new partition table we need to reboot the VM;

13. At the command prompt type **"reboot"** and hit **ENTER**. Close the dead Putty window and wait for a few minutes until the server comes back. Once the server is back repeat the procedure to re-open the **Putty** and **SShd into "web-01a" VM**.

### Check the new partition size

1. At the command prompt type **"fdisk =l"** and hit **ENTER**;

![fdisk output](image)
2. That is all for partitioning, we now have a new partition on "/dev/sda4" which is making use of the previously unallocated disk space from the expanded disk.
3. Note that the LVM group still have the same size. This is because we just created a partition to allocate the new added capacity. In the next steps we're going to create the physical volume and add it to the logical LVM group.

**Creating the physical volume**

We are going to use the `pvcreate` command to create a physical volume to be used by the logical volume manager (LVM). In this case the physical volume will be our new "/dev/sda4" partition.

1. At the command prompt type `pvcreate /dev/sda4` and hit ENTER;
2. At the command prompt type `vgdisplay` and hit ENTER to confirm the name and size of the current volume group;
3. At the command prompt type `vgextend VolGroup /dev/sda4` to extend the "VolGroup" volume group by adding in the physical volume of "/dev/sda4" which we created using the `pvcreate` command earlier;
4. At the command prompt type `pvscan` to scan all disks for physical volumes and hit ENTER. This should confirm the original "/dev/sda2" and "/dev/sda3" partitions and the newly created physical volume "/dev/sda4";
Expanding the logical volume

Next we need to expand the logical volume (rather than the physical volume) which basically means we will be taking our original logical volume and extending it over our new partition/physical volume of "/dev/sda4".

1. At the command prompt type "lvdisplay" and hit ENTER to confirm the name of the current volume group;
2. At the command prompt type "lvextend /dev/VolGroup/lv_root /dev/sda4" and hit ENTER to extend the logical volume using the name obtained above;
3. Finally, at the command prompt type "resize2fs /dev/VolGroup/lv_root" and hit ENTER to resize the file system so that it can take advantage of this additional space;
4. At the command prompt type "df -h" and confirm that the total available disk space has been increased to approximately 4.3GB.

You will notice that the original disk size was 2.5G but the root LVM have approximately 2GB due to space allocated to other partitions like swap. After expanding the volume it will have approximately 4.3GB instead of 5GB for the same reason.

List the current alerts of the VM after applying the recommendation

Now that we applied the recommendation for the "web-01a" VM by expanding its disk and presenting the new space to the guest OS, let's check if that particular alert is gone by using the same variable we used to store the alerts.

Re-create the alert PowerShell variable for the chosen VM
Since the specific alert for the "web-01a" VM is expected to be gone, we need to re-create the alert variable in order to refresh it with the new value.

1. In the **Script Pane** type "$alert = Get-OMAlert -Resource 'web-01a' -Impact Health -Status Active -Criticality Immediate" in the first line and hit **ENTER**;
2. In the next line of the **Script Pane** type "$alert | Format-List";
3. Click on "**Run Script (F5)"** icon or press the **F5** key to execute the script;
4. The **Command Pane** should return no results since the alert was gone after the recommendation was applied.
Using VMware PowerCLI to work with vRealize Operations Statistics

In this lesson you will learn

- How to retrieve metric data (or statistics) from vRealize Operations using VMware PowerCLI

Until now we were only working with the cmdlets from vRealize Operations module, since we are going to work with PowerCLI VI module cmdlets to perform operations on a VM, in addition to connecting to a vRealize Operations server we will also need to connect to a vCenter server.

The vRealize Operations we were using was the one with historical data and for this reason, the vCenter Servers connected to it do not exist in our lab environment, therefore we are going to switch to the live instance of vRealize Operations.

Retrieving metric data from vRealize Operations using VMware PowerCLI

Many times customers will ask if they can export metric data from vRealize Operations for usage in other analytical tools or reports. While there are other methods, the PowerCLI module offers a really elegant way to extract that data.

The cmdlet Get-OMStat will provide the metric data output but it is useful to review the cmdlet Get-OMStatKey first. vRealize Operations stores hundreds of metrics for CPU, memory, disk, networking and other items. Each of these metrics is contained in a construct called statKeys. To retrieve these statKeys programmatically you need to use the cmdlet Get-OMStatKey.
Storing VM resource object in a PowerShell variable and listing its CPU metrics

For this lesson we will still be using the VM named "web-01a".

1. In the **Script Pane** type "$vmresource = "web-01a"" in the first line and hit \ENTER\ (note that the VM name must be between double quotes since it is a string);
2. In the next line of the Script Pane type "Get-OMStatKey -Name cpu* -Resource $vmresource" and hit \ENTER\;
3. Click on "Run Script (F5)" icon or press the \F5\ key to execute the script;
4. Look at the **Command Pane** and you should see a lot of CPU related metrics.
Storing a specific metric in a PowerShell variable and listing its details

1. In the Script Pane type "$statkey = Get-OMStatKey -Name "cpu|workload" -Resource $vmresource" in the first line and hit ENTER;
2. In the next line of the Script Pane type "$statkey | Format-List" and hit ENTER;
3. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
4. Look at the Command Pane and you should see the details about the "cpu|workload" metric.
Getting statistics on the previous metrics for the VM based on a period of time

In this example we are going to list the "cpu|workload" metric average by minute for the last hour. Since this is a live instance of the vRealize Operations we don't have much data to work with a broader time range.

1. In the Script Pane type "Get-OMStat -Resource $vmresource -Key $statkey -From ([DateTime]::Now).AddHours(-1) -IntervalType Minutes -IntervalCount 1 -RollupType Avg" in the first line and hit ENTER;
2. Click on "Run Script (F5)" icon or press the F5 key to execute the script;
3. Look at the Command Pane and you should see the "cpu|workload" metric average by minute for the last hour for the "web-01a" VM.

There is a lot more capability than we have seen here, but hopefully this gives you a good start. For customers who have deep expertise in PowerShell and PowerCLI the vRealize Operations integration can be a huge help.
Conclusion

In this module we learned the basics of the Windows PowerShell and VMware PowerCLI and also:

- How to connect to a vRealize Operations instance using VMware PowerCLI
- How to use VMware PowerCLI to work with vRealize Operations alerts
- How to use VMware PowerCLI to work with vRealize Operations recommendations
- How to use VMware PowerCLI to work with vRealize Operations statistics

Congratulation on completing "Using VMware PowerCLI for vRealize Operations".

If you are looking for additional information on "Using VMware PowerCLI for vRealize Operations", try one of these:

- Click on this link
- Or go to https://tinyurl.com/y735p7lr
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most. [Add any custom/optional information for your lab manual.]

- **Module 1 - Operating system and application monitoring with Endpoint Operations** (60 minutes) (Advanced)
- **Module 2 - Creating custom dashboards** (30 minutes) (Advanced)
- **Module 3 - Managing policies** (30 Minutes) (Advanced)
- **Module 4 - Working with views and reports** (30 mins) (Advanced)
- **Module 5 - Using the API for inbound integration** (30 mins) (Advanced)
- **Module 6 - Using Webhook shims for outbound integration** (30 mins) (Advanced)
How to End Lab

If you wish to conclude your lab at this time click on the END button. This will terminate your lab and all progress. Do this only if you wish to NOT proceed with the other modules.
Conclusion

Thank you for participating in the VMware Hands-on Labs. Be sure to visit http://hol.vmware.com/ to continue your lab experience online.

Lab SKU: HOL-1801-04-CMP

Version: 20181024-190102