# Table of Contents

Lab Overview - HOL-1901-05-CMP - vRealize Operations and vRealize Log Insight - Advanced Topics .................................................................................................................. 3  
Lab Guidance .................................................................................................................. 4  
Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes) ........................................................................................................................ 10  
    Introduction .................................................................................................................. 11  
    Log in to the vRealize Operations Live Instance ........................................................... 12  
    Creating Custom Dashboards ...................................................................................... 16  
    Creating a New Dashboard .......................................................................................... 26  
    Conclusion ................................................................................................................... 47  
Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes) .............................................................................................................. 48  
    Introduction ................................................................................................................ 49  
    Log in to the vRealize Operations Live Instance ........................................................ 50  
    Create Simple View showing VM list with Metrics and Properties .......................... 54  
    Create a View with Variable data ............................................................................. 67  
    Create a View with Trends ....................................................................................... 81  
    Create a View with Distribution data ....................................................................... 88  
    Create Reports from Views and Dashboards ............................................................ 96  
    Conclusion ................................................................................................................ 104  
Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes) ............................................................................................................... 105  
    Introduction ............................................................................................................... 106  
    Log in to the vRealize Operations Live Instance ....................................................... 107  
    Recommended Actions - Using & Defining Alerts and Recommendations .............. 111  
    Conclusion ................................................................................................................ 134  
Module 4 - Super Metrics in vRealize Operations (45 minutes) .................................. 135  
    Introduction ............................................................................................................... 136  
    Log in to the vRealize Operations Live Instance ....................................................... 137  
    Create Some Super Metrics ..................................................................................... 141  
    Enable Super Metrics on Objects and View the Results ......................................... 167  
    Export and Import Super Metrics .......................................................................... 175  
    Conclusion ................................................................................................................ 179  
Module 5 - Using the vRealize Operations API (30 minutes) ....................................... 180  
    Introduction ............................................................................................................... 181  
    A tour through the REST API .................................................................................. 183  
    Working with resources and metrics ...................................................................... 193  
    Conclusion ................................................................................................................ 210  
Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes) ....................................................................................... 211  
    Introduction ............................................................................................................... 212  
    Log in to the vRealize Operations Live Instance ....................................................... 213  
    Self Driving Operations with vRealize Operations .................................................. 217
Conclusion........................................................................................................... 254
Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30
Minutes) .................................................................................................................. 255
  Introduction........................................................................................................ 256
  Log in to the vRealize Operations Live Instance........................................... 257
  Launching a workflow from vRealize Operations......................................... 261
  Conclusion........................................................................................................ 271
Module 8 - Users and Roles in vRealize Operations (30 minutes) ..................... 272
  Introduction........................................................................................................ 273
  Log in to the vRealize Operations Live Instance........................................... 274
  User Access and Preferences ........................................................................ 278
  Roles and Privileges ....................................................................................... 292
  Manage Dashboards ....................................................................................... 312
  Auditing Users and the Environment............................................................. 322
  Conclusion........................................................................................................ 328
Module 9 - PowerCLI for vRealize Operations (60 minutes) ................................. 330
  Introduction........................................................................................................ 331
  Basic Windows PowerShell usage and VMware PowerCLI for vRealize Operations
  cmdlets ............................................................................................................ 334
  Connecting to vRealize Operations with VMware PowerCLI........................ 340
  Using VMware PowerCLI to work with vRealize Operation Alerts ............. 341
  Using VMware PowerCLI to work with vRealize Operations Statistics........ 351
  Log in to the vRealize Operations Live Instance........................................... 356
  Using VMware PowerCLI to work with vRealize Operations Recommendations... 359
  Conclusion........................................................................................................ 368
Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator
  Workflows (45 minutes) ...................................................................................... 370
  Introduction........................................................................................................ 371
  Log in to vRealize Log Insight....................................................................... 372
  Self Healing DataCenter with vRealize Log Insight...................................... 377
  Conclusion........................................................................................................ 406
Lab Overview - HOL-1901-05-CMP - vRealize Operations and vRealize Log Insight - Advanced Topics
Lab Guidance

Note: It may 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.

In this lab we dive into some more of the advanced capabilities of Operations Manager. The individual modules attempt to use examples of these capabilities intended to give the user a good understanding of what advanced capabilities exist and how to get started using them. Advanced vROPS knowledge is not a prerequisite to get value from this lab. So don’t be afraid to jump in and get started with these capabilities. We hope it will just be the being of you gaining the skills that will help you get the most from your vROPS implementation.

Lab Modules:

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - RemEDIATE Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - RemEDIATE Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Lab Captains:

- **Modules 1,3,4** Michael Lapham  
  Cloud Management Specialist SE, USA - New Jersey
- **Module 2** Jack White  
  Cloud Management Specialist SE, USA - Ohio
• Modules 5, 6, 10  
  USA - San Diego  
  Tony Welsh  
  Cloud Management Specialist SE,

• Module 8  
  Minneapolis  
  Mandy Storbakken  
  Cloud Management Specialist SE, USA -

• Module 9  
  Sacramento  
  Jase Machado  
  Cloud Management Specialist SE, USA -

This lab manual can be downloaded from the Hands-on Labs Document site found here:

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:


Location of the Main Console

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 cannot 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 you lab has not changed to "Ready", please ask for assistance.
Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)
Introduction

If you are responsible for the health, availability, and performance of any of the IT resources used by your organization, you know that getting the right information about those resources at the right time is critical to meeting the heightened expectations of IT consumers.

As technologies like visualization and cloud have somewhat simplified using and supporting IT, they have also introduced additional complexities. Visibility, understanding of relationships, and automated analytics like those provided with vRealize Operations Manager (vROPS) are key in the successful management of these complex environments. How we consume and act on information must also fit within the operation models and skill sets of those using the tool. Dashboards are one of the most utilized parts of vROPS because they can help the user understand the environment and issues with simple views.

vROPS 6.7 comes pre-loaded with many Dashboards to get you going on day one, and Management Packs are available in the VMware Solution Exchange to be installed to extend the capabilities of vROPS. Additionally, if you own the Advanced or Enterprise editions, you can create your own custom Dashboards. Custom Dashboards are utilized when users have simple formatting preferences or wish to extend the capabilities by adding custom properties and metrics to new or existing Dashboards.

In this module, you will:

• Extend an existing out-of-the-box Dashboard by adding an additional view
• Create a simple Dashboard from scratch to help understand how views and widgets interact
• Configure widgets to display date in different ways
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance:** Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance:** Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs:** Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click [here](#) to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the **Chrome** icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the **vRealize Operations - Live Instance** link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
Creating Custom Dashboards

The vRealize Operations Manager out-of-the-box Dashboards are created by industry experts who have a deep understanding of vRealize Operations Manager as well as the characteristics and behavior of the underlying objects being managed. However, personalizing a Dashboard to fit a specific role or consolidate other information into a single view is a common use case.

To start, we will clone and make some simple changes to create a custom Overview Dashboard for our Rainpole administrators. For this example, we will clone the existing Operations Overview and add Recommended Actions.

Cloning the existing content to create a new/modified Dashboard is considered a best practice to ensure your custom content is not affected during an upgrade.

Extending an Existing Dashboard

To start, let's look at the existing Operations Overview Dashboard.

1. Click **Home**.
2. Click **Operations Overview**.
Operations Overview

Looking at the existing Dashboard, we see it provides a good understanding of the managed environment, including some basic performance and availability indicators. However, the Rainpole administrators would like to add a Scoreboard to this view to see some of these additional performance indicators. Following best practice, we will create and modify a new Dashboard by cloning this one.

Managing the Dashboard

![Dashboard Management Steps]

1. Click on the Dashboard tab.
2. Navigate to the Actions menu and select "Create Dashboard".
3. Choose "Manage Dashboard" to modify the existing Dashboard.
We need to access the Manage Dashboards screen to work with existing content.

1. Click **Dashboards**.
2. Click **Actions** to expand the list of actions.
3. Click on **Manage Dashboards** in the Actions list.

### Select Dashboard

![Manage Dashboards screenshot]

We can use a filter to zero in on the content we want to clone.

1. From **Manage Dashboards**, type **Operation** in the filter field and hit **Enter**.
Clone Dashboard

We clone the Dashboard so that our new content will not be overwritten during an upgrade.

1. Select the Operations Overview Dashboard and click on the clone icon.
2. Click Clone Dashboard.
3. In the Clone Dashboard dialog, enter My Operations Overview.
4. Click OK.

Locating the New Dashboard

1. Click Dashboards.
2. Click to expand **All Dashboards**.
3. Click **My Operations Dashboard**.

**Editing the New Dashboard**

You can edit Dashboards by selecting the Edit Dashboard Action in the existing Dashboard.

1. Click **Actions**.
2. Click **Edit Dashboard**.

**Configure Default Widget View**
Because of the screen resolution in our lab environment, we will collapse the widgets in the Dashboard template to make room on the screen for our new widgets. This will be saved as the default when viewing the Dashboard.

1. Hover over the **Top-15 VM Experiencing CPU Contention** widget and click the Up arrow to collapse.
2. Hover over the **Top-15 VM Experiencing Memory Contention** widget and click the Up arrow to collapse.
3. Hover over the **Top-15 VM Experiencing Disk Latency** widget and click the Up arrow to collapse.

### Extending an Existing Dashboard

![Dashboard screenshot]

We can now drag and drop the Scoreboard widget onto our new Dashboard. For now, we will just use the out-of-the-box Scoreboard setting and metrics. Later in this lab, we will modify a Scoreboard to use a custom set of metrics.

1. Type **score** in the filter.
2. Click and drag the **Scoreboard** widget onto the Dashboard.
Re-size the widget

1. Click and drag the **Scoreboard** widget to the left column.
2. Click the bottom corner and stretch the **Scoreboard** widget to fill 3 columns.

Apply Interaction
Now that we have added a new Scoreboard widget, we need to setup up the widget integration make sure that the widget will populate with information about the Datacenter selected from the **Datacenter (DC)** widget.

1. Click on **Widget Interactions** in the scroll-down menu.
2. Choose **Select a Datacenter (DC)** to provide the **Scoreboard**.
3. Click **Apply Interactions**.
4. Click **SAVE**.

**Widget Interactions**

If you configured interactions between widgets at the Dashboard level, you can then select one or more objects in the providing widget to filter the data that appear in the receiving widget. This allows you to focus on data related to an object.

When you configure Widget Interactions, you specify the providing widget for the receiving widget. For some widgets, you can define two providing widgets, each of which can be used to filter data in the receiving widget.

**Self Provider**

Indicates whether the objects with data appearing in the widget are defined in the widget, or provided by another widget.

- On: You define the objects for which data appear in the widget
- Off: You configure other widgets to provide the objects to the widget, using the Dashboard Widget Interactions options
1. Click **Dashboards**.
2. Click **All Dashboards** to expand.
3. Click **My Operations Dashboard**.
Review Your New Dashboard

We created this simple example Dashboard to demonstrate how existing content can be enhanced. Note your changes to the new Dashboard and how the view is set with the Top-N widgets collapsed until you expand them.
Creating a New Dashboard

Now, we will create a Dashboard from scratch that will be the base we will expand on in the remainder of this lab.

Creating a New Dashboard

If you are starting with a blank Dashboard, you can the select the action from the Getting Started section.

1. Click **Dashboards**.
2. Click **Actions** to expand.
3. Click **Create Dashboard**.
Name the Dashboard

To begin, our new Dashboard must have a name.

1. In the **Dashboard Configuration** section, name the Dashboard **My Compute Favorites**.

Selecting Widgets

---
1. Click on **Widget List**.
2. Enter **object** in the filter. (Note: Update the filter to locate the additional widgets.)
3. Drag and drop the **Object List, Object Relationship, Metric Picker, Metric Chart, and Recommended Action** widgets onto the canvas.
4. Re-size **Recommended Action** to fill the bottom of the screen.

**Filter by Object Type**

By filtering on an object type in the Object list, the user will be presented with just those types of objects to select. For this example, we will focus at the cluster level.

1. Click the **Edit** icon on the Object List widget.
2. Rename it to **Clusters**.
3. Set Mode = **Self**.
4. Expand **Object Types**.
5. Select **Cluster Compute Resource**.
6. **Click SAVE**.
1. Click on **Widget Interaction**.
2. Set Interactions as **Clusters >> Object Relationship, Object Relationship >> Metric Picker, Object Relationship (select object) & Metric Picker (select metric) >> Metric Chart**.
3. Click **Apply Interactions**.
4. Click **SAVE**.
Open New Dashboard

1. Click **Dashboards**.
2. Click **My Compute Favorites**.

Review New Dashboard (continued)
1. Use the **Compute Resourcing Picker** to select different objects to verify your Dashboard updates.

Note how you can select metrics for any of the objects you choose from the Object Laureateship view.

Now, we will improve our basic Dashboard using some of the advanced widgets that became available with the latest 6.7 vRealize Operations Manager release.

**Widget Enhancements**

![Diagram of widget enhancements](image)

In this section, we will clone a Dashboard and modify the Heat Map widget to demonstrate of the new settings that are available.

1. Click **Dashboards**.
2. Click **Actions**.
3. Click **Manage Dashboards**.
4. Type `datastore` in filter and hit **Enter**.
5. Click on **Datastore Usage Overview**.
6. Click the **Clone** icon to copy the Dashboard.
7. Enter **My Datastore Usage**.
8. Click **OK**.
Focus on Groups

A couple of simple changes can help you focus on specific areas of interest.

1. Click on **My Datastore Usage** Dashboard.
2. Click the **Pencil** icon to edit the **Storage Demand from all VMs** widget.
3. Click the drop-down for **Group BY** and select **Host System**.
4. Click to check **Focus on Groups**.
5. Click **SAVE**.

Next, we will create our own Scoreboard using a Custom Metric Configuration.
Linking Dashboards

1. Click **Edit Dashboard**.
Adding Dashboard Navigation

Using Dashboard navigation allows us to access our new Dashboard in context from another Dashboard.

1. Click on **Dashboard Navigation**.
2. Click on **My Datastore Usage** for destination on Cluster widget.
3. Click to select **All Widgets**.
4. Click **SAVE**.
Custom Metric Configurations

Custom Metric Configurations allow us to define a set of metrics we can use when creating our custom Dashboards.

1. Click **Administration**.
2. Click **Configuration** to expand.
3. Click **Metric Configurations** to select.
4. Click **ReskndMetric**.
5. Click the + icon to add a new configuration.
6. Enter **My Scoreboard**.
7. Click **OK**.
Select Metrics

The wizard is used to generate the required xml.

1. Click `<AdapterKind>`.
2. Click `vCenter Adapter`.
3. Click `GO`.

![Diagram showing steps to select metrics using the wizard]
Metric Configurations

You are now able to select a resource type based on the selected adapter. For our Compute Scoreboard, we will select the Cluster Compute Resource.

1. Click `<ResourceKind>`.
2. Click Cluster Compute Resource.
3. Click GO.
Pick Metrics

1. Click <Metrics>.
2. Click to expand Capacity Analytics Generated and select Capacity Remaining Percentage, >CPU>Demand Capacity Remaining, & Time Remaining, >Disk Space >Demand Capacity Remaining, & Time Remaining and >Memory>Demand Capacity Remaining, & Time Remaining
3. Click SAVE.

My Scoreboard Dashboard
Now, we will create a simple Dashboard with our Scoreboard metrics.

1. Click **Dashboards**.
2. Click **Create Dashboard**.

**Edit Dashboard**

1. Enter **My Scoreboard** in **Name** field.
Adding Widget

1. Drag the **Scoreboard** widget onto the Dashboard.
1. Select **My Scoreboard** from Metric Configuration menu.
2. Click **SAVE**.
1. Stretch the widget to fill the Dashboard.
2. Click **SAVE**.
Linking Dashboards

Now, we will return to our My Computer Favorites Dashboard and link or new Scoreboard using Dashboard Navigation. This allows us to create our custom workflow.

1. Click **Dashboards**.
2. Click on **My Compute Favorites**.
1. Click **Edit Dashboard**.
Adding Dashboard Navigation

By applying navigation to the Dashboard template, we can navigate in context between Dashboards.

1. Click on **My Scoreboard** for destination on Cluster widget.
2. Click to select **All Widgets**.
3. Click **Apply Navigations**.
4. Click **SAVE**.
Review Scoreboard

1. Click the **Open** icon.
2. Click **Navigate**.
3. Click **My Scoreboard**.

Scoreboard

Note: The selected metrics in the Metric Configuration are displayed for the cluster selected from the previous Dashboard.
Conclusion

In this lab, we customized existing Dashboards, created new Dashboards, updated widgets, created custom metric schema, and added navigation. This is just the start of what can be done with vROPS Dashboards, so be sure to check out some of the Blogs at blogs.vmware.com for additional ideas for custom Dashboards of your own.

You've Finished Module 1

Congratulations on completing Module 1.

If you are looking for additional information on vRealize Operations, visit https://www.vmware.com/products/vrealize-suite.html.

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow the links below.

- Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)
- Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)
- Module 4 - Super Metrics in vRealize Operations (45 minutes)
- Module 5 - Using the vRealize Operations API (30 minutes)
- Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)
- Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)
- Module 8 - Users and Roles in vRealize Operations (30 minutes)
- Module 9 - PowerCLI for vRealize Operations (60 minutes)
- Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)

Or, if you want to end your lab:

1. Click on the **END** button at the top of the page.
Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)
Introduction

This Module contains the following lessons:

• Create Customized View
• Simple View showing VM list with Metrics and Properties
• Create a View with Variable data
• Create a View with Trends
• Create a View with Distribution data
• Create Reports from Views and Dashboards
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance**: Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance**: Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs**: Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the **Chrome** icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the **vRealize Operations - Live Instance** link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
Create Simple View showing VM list with Metrics and Properties

In this lesson, we will create a view. A view can be used in dashboards and reports. A view is also viewable as its own content in the Details section of the vRealize Operations interface.

The view for this lesson is a starting point and intended to be a simple example to create. It will contain some basic metrics and properties for virtual machines.

Go to Environment

1. Click on 'Environment'

Hosts and Clusters
1. Click on 'vSphere Hosts and clusters'

Expand vSphere World

1. Expand vSphere World
2. Select 'vcsa-01a.corp.local'
3. Click on 'more...' 

Create a View

1. Click on 'Details'
2. Click the Green Plus Sign to create a new view.

**View Name**

The view creation wizard starts. Create a view with the following:

**Name and Description**

1. Enter: "Demo - Simple List of VMs with Metrics and Properties"
1. Click on **Presentation**
2. Select: **List**
1. Click on Subjects
2. Enter 'Virtual Machine' (Begin Typing and the list will populate with matched options. Click on Virtual Machine)
1. Click on **Data**:  
2. Switch from Metrics to Properties
1. Expand Summary and scroll down
2. Double-click on 'Parent Cluster' (drag and drop the data to the center will work also)
3. Double-click on 'Parent Host'
4. Double-click on 'Datastore(s)'
Data - Metrics

We've been working with Virtual Machine Properties, now we need to select Virtual Machine metrics.

1. Switch from Properties to Metrics
Select Metrics

1. Expand Configuration (Single-click the chevron ‘>’ to expand)
2. Expand Hardware
3. Double-click ‘Number of CPUs (vCPUs)’

Memory Metrics

Scroll down and Expand Memory
1. Expand the Memory Category

**Total Capacity**

1. Scroll down and Double-click on 'Total Capacity (KB)'

**Summary**

1. In the center of the screen, Click on Summary
2. Click on the green plus sign to create a summary
Aggregation

1. Change aggregation to 'Sum'
2. Click 'Show Advanced Settings'

Clear All

1. Deselect all by clearing the check box next to 'Data'

Clicking the box will toggle 'Select all' and 'De-select All'.

Make sure your screen matches the image. Nothing should be selected at this point.
Selected Sum

Scroll down to find the following:

1. Select ‘Configuration|Hardware|Number of CPU(s) (vCPUs)’
2. Select ‘Memory|Total Capacity (KB)’
3. Click Save

Viewing the data

After clicking Save you will be in the view area again. The data we just selected will be displayed.
You should see the three properties and two metrics we selected. At this point, your view is created and saved.

**View the Sums**

1. Scroll to the bottom of the results to see the summary for the total vCPUs and Total Memory.

The sum is for all of the Virtual Machines contained in the view.

Because we used Virtual Machines as our subject matter, the view can be utilized for a single VM or anything that contains Virtual Machines like Hosts, Groups, Clusters, Datacenters, Applications, etc.

Feel free to navigate to a Host or any object that contains virtual machines to see the flexibility of a View.

**Lesson End**

This completes the Simple View creation. In the next lesson, we will show how to create a view with variable data.
Create a View with Variable data

In this lesson, we are going to create a custom view. The view will concentrate on Virtual Machine data but can be applied to any resource collected in vRealize Operations.

Views can be used within reports and dashboards. They also allow vRealize Operation Users to see data within vRealize Operations.

Go to Environment

1. Click on Environment
Hosts and Clusters

1. Expand **vSphere Environment**
2. Click on **vSphere Hosts and clusters**

Select a vCenter Server

1. Expand **vSphere World**
2. Select vCenter Server **vcsa-01a.corp.local**
3. Click on 'more...'
Create a View

1. Click on **Details**
2. Click the **Green Plus Sign** to create a new view.
View Name

Create a view with the following data:

Section 1. Name

1. Enter **Demo - Variable Data**
View Presentation

1. Click on 2. Presentation
2. Click on List
Subjects - Virtual Machine

1. Click on **3. Subjects**
2. Begin typing **Virtual Machine** (Start typing Virtual and the list will show available resources that match. Click on Virtual Machine).
Data Metrics

1. Click on 4. Data
2. Expand CPU
3. Double-Click Demand (%)
4. Double-Click Demand (%) -- It will be in the list two times.

Metric Transformation

1. After Demand (%) is in the list a second time, select the second instance with a single click.
2. For **Metric label**: Enter "CPU Max %" (This will be our column header name in the final view)
   - The screenshot was taken after the changes were made to the Metric label. The Data column will reflect what you type into the Metric label.

3. For **Transformation**, Select **Maximum** in the drop down list

**Note:** You may need to manipulate the screen by scrolling down in the configuration area.

**CPU Ready**

1. Put **Ready (%)** into the data window with a double-click (or drag and drop)
2. Change **Transformation: Metric Correlation**
You may have to scroll down to see the metric correlation area. There will be a link to select the correlated metric.

1. Click **SELECT**
Correlated Metric

In the pop-up window:

1. Expand CPU
2. Select **Demand (%)**
3. Click **OK**

With this correlation, we are going to see the value of CPU Ready (%) when the CPU Demand (%) is at a maximum.
Time Settings

In the center of the screen:

1. Click on **Time Settings**.

Number of Days

1. Set the **Relative Date Range** to the **Last 30 Days**.
Date Range

In the center of the screen:

1. Click on **Filter** to create a virtual machine filter.

Scroll down

1. If you feel limited by screen space, scroll down in the filter view to display the area to define the filter.
Switch from Metric to Properties.

1. Select **Properties**:
2. Expand **Summary** and Expand **Runtime**, Double-click '**Power State**'.
3. Select: **is**
   - Use the tab key to move into the property value field. You should see the available options of **Powered On** and **Powered Off**.
4. Select: **Powered On**
5. Click **SAVE**
We now have a view that shows us the last CPU Demand collected for each Powered ON Virtual Machine. We also show the Maximum CPU Demand as a percentage for the last 30 days. The last value in our view shows us what the Ready % was when the demand was at maximum during the same 30 day period.

I believe this is a very powerful feature of the product. While we are showing the ready % when the CPU is highly demanded, you may wish to see what disk latency looks like when network transmissions are high. You can correlate any two metrics that are being collected in vRealize Operations.

Lesson End

This completes this lesson. In the next lesson, we will create a view with trended data.
Create a View with Trends

In this lesson, we continue the concept of creating custom views. This time, we will create a view with data that is trended over a period of time.

Go to Environment

1. Click on 'Environment'

Hosts and Clusters

1. Click on 'vSphere Hosts and clusters'
Select a vCenter Server

1. Expand **vSphere World**
2. Select vCenter Server `vcsa-01a.corp.local`
3. Click on 'more...'

Create a View

1. Click on **Details**
2. Click the Green Plus Sign '+' to create a new view.
View Name

1. **Name**: "Demo - Trend View"

Presentation Style

1. Click on **Presentation**
2. Select **Trend**
3. When selecting a Trend presentation, notice the option to set the maximum plot lines. The default is 25. Since we are using Virtual Machines as our subject, if we select a cluster that has 25 or more virtual machines, the view we are creating will attempt to display 25 trend lines. It can be a crowded view. Be aware of that constraint when you are selecting the metrics to trend. You can always create multiple views to display different metrics trended.

**NOTE:** Be aware that properties cannot be trended, only metrics.

**View Subjects**

1. Click on 3. Subjects,
2. Start typing 'Virt' and a filtered list will appear. Click on 'Virtual Machine.'
View Data

1. Click on **4. Data**:  
2. Expand **Storage**  
3. Double-click **Read Latency (ms)** (You may drag and drop the metric as well)

Transform the data
1. Once the metric is in the view, single-click on it and change the following:
2. Uncheck 'Trend of the historical data'
3. Uncheck 'Forecast data for the next'
4. In the center of the window, click on **Time Settings**.

**Time Range**

![Image of Time Settings window]

1. Change the **Relative Date Range** to **Last 30 Days**.
2. Click **SAVE**

**Summary**

You now have a view that shows selected virtual machines read latency trended over the last 30 days. While we unchecked the forecast data option, leaving it checked would have trended the forecast of the selected metrics for up to a year.
Lesson End

You have completed this lesson. The next lesson will show how to create a view with distribution data.
Create a View with Distribution data

If you've completed the previous lessons in this module, we have created various views. In this lesson, we continue creating custom views with the Distribution view. The distribution view gives us the ability to create pie charts based on data from selected object type.

Go to Environment

1. Click on 'Environment'

Hosts and Clusters

1. Click on 'vSphere Hosts and clusters'
Select a vCenter Server

1. Expand **vSphere World**
2. Select vCenter Server '**vcsa-01a.corp.local**'
3. Click on '**more...**'

Create a View
1. Click on 'Details'
2. Click the Green Plus Sign '+' to create a new view.

1. Enter the view name 'Demo - Distribution View'

View Presentation

1. Click on '2. Presentation'
2. Select 'Distribution'
3. Scroll down
4. Visualization **Pie Chart**

**Distribution Type**

![Distribution View](image)

1. Scroll down Further
2. Distribution Type: **Discrete Distribution**
View Subjects

1. Click on '3. Subjects'
2. Begin typing 'Virt'. The filter will display resources that match what is typed. Click on Virtual Machine.
Select Properties

1. Select '4. Data'
2. Switch from Metric to Properties
Select a Metric

1. Click on 4. Data
2. Expand Summary
3. Expand Guest Operating System
4. Double-click on 'Tools Version'
5. Click SAVE

View Results
You will now have a distribution of the VMware Tools version in the environment.

Lesson End

You have completed this lesson. In the next lesson, we take you through the process to put views and dashboards into reports.
Create Reports from Views and Dashboards

In this lesson, we show how to create custom reports using views and dashboards.

Go to Environment

1. Click on 'Environment'

Hosts and Clusters

1. Click on 'vSphere Hosts and Clusters'
Select a vCenter Server

1. Expand **vSphere World**
2. Select vCenter Server `'vcsa-01a.corp.local'`
3. Click on `'more...'`

Create a Report

1. Click on `'Reports'`
2. Click the Green Plus Sign `'+'` to create a new report.
**Report Name**

1. Name: Demo - Report from Views and Dashboard

**Data Type**

1. Select '2. Views and Dashboards'
2. Data Type: Views
3. All Filters: Size
Selected Views

With the filter set to 'Size', we only see views that have the term 'size' in their name.

1. Double-click undersized virtual machines
2. Double-click oversized virtual machines

Data Type Dashboard

1. Change Data Type to **Dashboards**
2. Quick Filter: **Cost**
Select Dashboard

1. Double click on 'Assess Cost'

Report Format

1. Select '3. Formats'
2. Since we are including a dashboard as part of this report, uncheck CSV. A dashboard will not convert to CSV (comma separated values).
Layout Options

We have the ability to include the following:

- Cover Page
- Table of contents
- Footer

For a Cover Page:

- Can contain an image up to 5 MB.
- The default report size is 8.5 inches by 11 inches. The image is resized to fit the report front page.

Table of Contents

- Provides a list of the template parts, organized in the order of their appearance in the report

Footer

- Includes the date when the report is created, a note that the report is created by VMware vRealize Operations Manager, and page number.

We won't make any changes here. The defaults will be used.
Portrait or Landscape

Each view and dashboard can be oriented to portrait or landscape mode. For dashboards in a report, landscape will likely be a better choice to simulate the aspect ratio of a monitor. Some dashboards require scrolling. When a dashboard is too large to be displayed on the screen, it will not fit into a report very well either.

1. Make sure Access Cost is set to Landscape. Click the layout icon.

Save

1. Click SAVE

Run the report
1. Click ‘Run Template' from the icons in the toolbar above the report names.

**Generated Reports**

1. Click on Generated Reports

**Select PDF**

1. The report will be available as a PDF for viewing

Clicking the PDF icon will open the report. Feel free to open the PDF to see the results.

**Lesson End**

You have completed the last lesson in this module. You should now have an understanding in creating new views. You also now have the tools to create reports from any view or dashboard.
Conclusion

In this module you explored a few approaches for creating new and customizing views and reports in vRealize Operations.

You've finished module 2

Congratulations on completing module 2.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,

1. Click on the **END** button at the top of the page.
Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)
Introduction

vRealize Operations Manager (vROPS) Alerts are similar to rules used for years in monitoring critical IT resources. However, previous rules-based systems tended to be static and difficult to build, deploy, and maintain. vROPS Alerts leverage vROPS analytics and pre-defined content to provide a dynamic, effective, and scalable approach for identifying and resolving issues in your environment.

Alert Definitions consist of the following components that raise alerts, provide recommendations, and take automated actions to resolve the issues:

Symptoms

Symptoms define conditions that trigger if a condition becomes true; they are based on metrics or super metrics, message events, or fault events. A symptom set combines one or more symptom definitions by applying an Any or All condition that can trigger the alert.

Recommendations

Recommendations are the remediation options provided to resolve the issues. Recommendations are provided by domain experts and can be extended to include tribal knowledge, local procedures, etc.

Actions

Actions are accessible several ways in vROPS. They can link to recommendations for the user to execute after review, or be fully automated to execute when the alert is triggered.

This Lab

Upon completing this lab, you will be able to:

- Build a custom Alert Definition
- Simulate issues in the environment to demonstrate how to customize the alerts
- Utilize the different ways alerts can be used based on the critical nature or other characteristics of the monitored infrastructure
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance:** Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance:** Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs:** Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the Chrome icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the vRealize Operations - Live Instance link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
Recommended Actions - Using & Defining Alerts and Recommendations

First, we need to create a Symptom Definition. Symptom Definitions enable the vRealize Operations Manager to identify problems with objects in your environment, and then trigger alerts when conditions qualify as problems. In this scenario, the condition to monitor is the high CPU workload on the virtual machine "Photon-01a.corp.local." Creating one or more of the symptoms enables them to be added to an Alert Definition. When a symptom is triggered, vRealize Operations Manager will issue an alert.

Creating Custom Alerts

![Diagram of vRealize Operations Manager interface]

Alert, Symptom and remediation definitions are all managed under the Alert tab.

1. Click the **Alerts** tab.
2. Click to expand the **Alert Settings**.
3. Click **Symptom Definitions**.
4. Click the + icon to create a new Symptom Definition.
Select Object

1. Expand the Adapter list.
2. Click vCenter Adapter.
CPU Usage Definition

Configure the Symptom Definition with the following parameters.

1. Type a metric filter of `CPU|USAGE`.
2. Click to expand the resulting metrics.
3. Double-click `Usage (%)`.
4. Type `High CPU` for the symptom name.
5. Set the definition to `Critical`.
6. Type `95` as the value the symptom must exceed to be triggered.
7. Click to expand the `Advanced` section.
8. Modify the `Wait Cycle` to `1`. The Wait Cycle field shows that the trigger condition should remain true for this number of collection cycles before the symptom is triggered. This means that the symptom is triggered in the same collection cycle when the condition became true.
9. Modify the `Cancel Cycle` to `2`. The Cancel Cycle field shows that the symptom after the trigger condition is false for this number of collection cycles, after which the symptom is canceled. This means that the symptom is canceled in the same cycle when the condition becomes false.
10. Click `SAVE`.

HOL-1901-05-CMP
Alert Definitions are a combination of symptoms and recommendations you can use to identify problem areas in your environment and generate alerts.

To create Alert Definitions:

1. Click **Alert Definitions**.
2. Click the + icon to create a new Alert Definition.
Alert Name & Object

1. Type **High CPU Alert** for the alert name and description.
2. Click **Base Object Type**.
3. Scroll down and click **vCenter Adapter**.
4. Scroll down and click **Virtual Machine**.
Alert Impact

Alert Impact settings and their definitions are shown below. These settings determine how your alert will be classified and triggered. Note: The default settings will be used in this scenario.

- The **Impact** field will categorize the alert as a health, risk, or efficiency problem
- The **Criticality** field shows how serious the problem is

**For Criticality, you can select one of the following values:**

- Info (informational purposes only; does not affect badge colors)
- Warning (lowest level; displays yellow)
- Immediate (medium level; displays orange)
- Critical (highest level; displays red)
- The **Alert Type** and **Subtype** fields can be used to classify the alert; an example would be using these fields' information to route the alert to the appropriate personnel in your organization

**Finally, choose settings for your cycle, which are data collection intervals.**

- **Wait Cycle** indicates how many cycles should pass before triggering the alert
- **Cancel Cycle** indicates how many cycles without symptoms should pass before the alert is canceled
Add Symptom Definition

1. Type `high cpu` and press the Enter key to filter the Symptom Definitions to what we created in the previous step.
2. Drag High CPU to the Symptom Definition section on the workspace as shown.
3. Click Add Recommendations.

Add Recommendation

Now, we will define a new Recommendation for our custom alert based on our organization's policies.

1. Click the + icon to create a new Recommendation.
Custom Recommendation

For Production Virtual Machines, please assess the trend and add CPU Resources if trend is high.
All development machines are shut down and the developer is notified.

1. Paste the text shown above into the Recommendation Text area.
2. Click Actions.
3. Click Power Off VM.
4. Click SAVE.
Save New Alert Definition

1. Type production in the filter and hit Enter.
2. Drag your custom recommendation onto the Alert Definition.
3. Click SAVE to save the new Alert Definition.

Alert List

Verify that the Alert exists.

1. Type High CPU and then press the Enter key to reduce the Alert Definition list.
PuTTY to Photon-OS

1. Click on the PuTTY icon in System tray.

Start PuTTY Session

1. Click on Photon-01a.corp.local.
2. Click Open to start the PuTTY session.
Run CPU Load

We will now redirect dev/zero to dev/null to generate CPU load so that we can see the impact on the VM in vROPS.

1. Type `cat /dev/zero > /dev/null` and press the Enter key to start the CPU load.

Search for VM

1. Type `photon` to search for objects beginning with "photon".
2. Click Photon-01a to go to the Summary page of this VM.
Set up the CPU and Memory graphs by completing the following.

1. Click **All Metrics**.
2. Click to expand the **CPU** metric section.
3. Double-click **CPU/Usage (%)** to create a chart.
4. Click to expand the **Memory** metric section.
5. Double-click **Memory/Usage (%)** to create a chart for memory usage.
6. Click the `^` symbol to open the relationship window. Note: This is helpful when you want to see when an alert is triggered.
7. In our example, the color of the VM Health badge now changes from green to red. However, the color can be green, yellow, orange, or red, depending on the severity of the alert.
1. Click the **Refresh** icon after a couple of minutes. The graph will begin to show the increase in CPU usage on the Metric chart. Once CPU usage is above 95%, an alert should be generated.

2. Click **Summary** to move to the Summary page.

**Summary Page**
As shown here, the Health Status of the VM has changed to critical, and an alert has triggered regarding the high CPU usage. Note: You may see additional alerts for this VM as there are other alerts active within our environment.

1. Click **High CPU Alert** in the Alert column for more details about the alert.

**Alert Details**

From the **Symptoms** tab, we can easily view the activity that triggered the alert, the usage percentage, and the threshold.

1. Click on **Alerts**.
2. Click on **Symptoms**.
3. Click on **High CPU**.
4. Verify that the **custom action** is available.

We will now stop the load so that we can complete some additional configuration to enable automation.
Stop CPU Load

Closing the PuTTY session will end the CPU load, and the alert will clear.

1. Click the X in the upper-right corner to close the PuTTY session.

Fully Automated Actions

Here, we will create a custom policy for test VMs to enable the system to act based on the VM's policy assignment. In this case, we will power off test VMs that spike usage to prevent them from causing resource constraints in the virtual environment. By using the HOL Default Policy, all settings in that policy will be applied if they are not explicitly set in our new policy.

1. Click Administration.
2. Click Policies.
3. Click **Policy Library**.
4. Click **HOL Default Policy** to highlight.
5. Click the + icon to create a new policy.

**Name the Policy**

1. Enter **HOL Test Policy** in the Name field.
The policy allows us to set the action to be run at the time of an alert. In this case, we are adding resources, so it is important to know that the OS must support hot add for the changes to take effect. The critical VM in our lab does, so the change should take effect and the alert should clear. In cases where the OS does not support the change, the action would run but not take effect until the system is rebooted.

1. Click **Alert / Symptom Definitions**.
2. Type **High** in filter and hit **Enter**.
3. Under the **Automate** tab, select **High CPU Alert** and then click on **Local** in the drop-down list. Note that the action we assign in the alert is linked in the policy.
4. Click **SAVE**.
Add VM to Test Group

We will now create a new group for test VMs and apply our Test VM Policy to it. In this lab, we only have one test VM, but we will be able to configure the group to add additional machines dynamically and apply our policy.

1. Click on **Environment**.
2. Click the + icon to create a new group.
Define the New Group

1. Enter **Test Virtual Machines** in the **Name** field.
2. Select **Function** for group type.
3. Select **HOL Test Policy**.
4. Click the check box to select **Keep group membership up to date**.
5. Select **Virtual Machine** under the vCenter adapter for object type.
6. Under **Object Name**, select **contains** and then enter `pho` for the selection criteria.
7. Click **PREVIEW** (make sure the VM shows up).
8. Click **CLOSE**.
9. Click **OK**.
Check Policy

Verify our critical VM as the newly assigned policy.

1. Click the search glass and enter photon-01a.
2. Click photon-01a.
3. Verify that HOL Test Policy is assigned.

Open PuTTY Session

1. Click on PuTTY icon in System tray to view the PuTTY Configuration options.
Open PuTTY Session

1. Click on Photon-01a.corp.local.
2. Click Open to start the PuTTY session.

Run CPU Load

Redirect dev/zero to dev/null to generate CPU load again; this will trigger the alert and show how it behaves with the new policy.
1. Type `cat /dev/zero > /dev/null` and press the **Enter** key to start the CPU load.

**Alert**

Let the load populate for a few minutes, and then check the alerts from the Alert screen.

1. Click **Alerts**.
2. Click **All Alerts**.
3. Click the **High CPU Alert** and verify when it was triggered.

We looked at the alert previously, so now we will check the status of the action.

**Recent Task List**
1. Click the **Power Off VM** task. Additional details regarding the completed operations are displayed below.

Let's take a look at the VM in the vSphere Client.

**vSphere Client**

To log into vCenter:

1. Click the **HTML5 Client** link in the browser ribbon.
2. Click on **Use Windows session authentication**.
3. Click **Login**.

**VM Status**

The Summary screen will update automatically after a few minutes, showing the health is once again green and that the alert has cleared.
Conclusion

Self-driving operations by VMware vRealize® Operations™ automates and simplifies IT operations management and provides unified visibility from applications to infrastructure across physical, virtual and cloud environments. We hope in this module you learned how Intelligent Remediation helps predict, prevent and even take actions to resolve issues upon detection. VMware vRealize® Operations™ allows for faster troubleshoot with actionable insights correlating metrics and logs and unified visibility from applications to infrastructure.

You've finished module 3

Congratulations on completing module 3.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,

1. Click on the END button at the top of the page.
Module 4 - Super Metrics in vRealize Operations (45 minutes)
Introduction

A super metric in vRealize Operations is a mathematical formula that contains one or more metrics. It is a custom metric that you design to help track combinations of metrics, either from a single object or from multiple objects. If a single metric does not inform you about the behavior of your environment, you can define a super metric.

After you define it, you assign the super metric to one or more object types. This action calculates the super metric for the objects in that object type and simplifies the metrics display. For example, you define a super metric that calculates the average CPU usage on all virtual machines, and you assign it to an ESXi host. The average CPU usage on all virtual machines on that host is reported as a super metric for the host.

Because super metric formulas can be complex, plan your super metric before you build it. The key to creating a useful super metric is knowing your own enterprise and data.

When creating or modifying super metrics, you might find it helpful to open a new browser tab or window open to the All Metrics tab in vRealize Operations for an object of the type you are creating a super metric for. You can then switch back and forth between the tabs to make sure you are using the desired metrics in your super metric definition.

Super metrics can reference an object above or below the object in the hierarchy where it will be applied. For example a super metric that is defined for hosts can use metrics from VMs on that host or the parent cluster or datacenter for that host.

With super metrics you can

- Apply functions to a metric, for example, what the average is of a metric such as execution time of queries on a database, you can do that with one operation (avg()).
- Perform rollups. For instance, if you are interested in three volumes on a NetApp storage appliance not reaching 80%, you can do that by creating the average of the sum of all three ‘volume space occupied’ metrics.

In this module we will be learning more about super metrics by creating some examples in our lab environment.
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- Live Instance: Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- Historical Instance: Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- Blue Medora Management Packs: Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the Chrome icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the **vRealize Operations - Live Instance** link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER:  hol

PASSWORD:  VMware1!

1. Click Sign in
Create Some Super Metrics

We will create three different super metrics in this lesson, each more complex than the previous. While you will learn the mechanics of how to create some super metrics and will see some of the available functions, be aware that super metrics while very powerful can also be a bit complex. This lesson will get you started but you can go a lot deeper into super metrics than we will delve here.

Create a Super Metric Using Metrics from Multiple Objects

For our first example, we will create a super metric that performs some mathematical functions using existing metrics from two different objects to calculate a percentage.

Let's suppose you want to know the under-subscription or over-subscription percentage of allocated datastore space (the sum of all allocated virtual disks on a datastore) versus the size of the datastore. This is not a metric that is calculated by default in vRealize Operations so you are going to have to create a new (super) metric to calculate that information. This can be a helpful thing to know.

Navigate to the Super Metric Page and Open the Editor for a New Super Metric

To create a new super metric:

1. Click Administration
2. Click > Configuration to expand the section
3. Click Super Metrics
4. Click + to create a new super metric

The Super Metric Editor Window

Note the parts of the Manage Super Metric window.

1. Super metric. Use the toolbar selections to build and display your super metric formula.
   - Functions. Mathematical functions that operate on a single object or group of objects.
   - Operators. Mathematical symbols to enclose or insert between functions.
   - This Object. Assigns the super metric to the object selected in the Object pane and displays this in the formula instead of a long description for the object.
   - Show Formula Description. Shows the formula in a textual format.
   - Visualize Super Metric. Shows the super metric in a graph. Look at the graph so that you can verify that vRealize Operations Manager is calculating the super metric for the target objects that you selected.
   - Name. The name you give to the super metric.

2. Objects pane. Displays the list of objects collecting metrics. Use this list to select the object with the metrics to measure. If an object type is selected, only objects of the selected type are listed. Column headings help you to identify the object.
3. Object Types Pane. Use this list to select the object type with the metrics to measure. The object type selection affects the list of objects, metrics, and attribute types displayed.

- Adapter Type. Shows the object types for the adapter selected.
- Filter. Shows the object types with the filter words.

4. Metrics pane. Displays the list of available metrics for the object or object type selection. Use this list to select the metrics to add to the formula.

5. Attribute Types Pane. Displays the list of attribute types for the object or object type selection. Use this list to select the metrics for the attribute type to add to the formula.

Name the Super Metric and Begin Building

Let's get started.

1. Type the name **Datastore Subscription Percentage**
2. In the filter field, type **datastore** and press Enter
3. Click to select the Datastore object type
4. In the Metrics pane, scroll down
5. Click >**Disk Space** to expand the metric category
Add the Provisioned Space Metric

We will be applying this super metric to datastores in our environment. The super metric editor provides a shortcut method of referring to the object where the super metric will be applied - the This Object button. Whenever you are creating a super metric and need to use a metric in the formula from the object where the super metric will be applied, use the This Object button. In the next section we will see how you can also reference other objects when creating a super metric and the distinction should become more clear at that point.

1. Click the **THIS** button
2. Double-click the **Provisioned Space (GB)** metric to add it to the super metric definition
3. Type the `/` key on your keyboard (forward slash for the divided-by mathematical function). You can also click the Operators drop-down list to see all available operators and select one from there.
4. Verify: `_${this, metric=diskspace|provisioned}_/`
Add the Total Capacity Metric

Now we will add the denominator of the mathematical equation - the datastore's total capacity.

In the Metrics pane, scroll down

1. Click the **THIS** button
2. Hover your mouse over the **Total Capacity (GB)** metric. Note that for many metrics, tool-tips will provide additional information about the metrics. Double-click the **Total Capacity (GB)** metric to add it to the super metric definition
3. Verify: `${this, metric=diskspace|provisioned}/${this, metric=diskspace|capacity}`

Finish the super metric definition
At this point we have a ratio of provisioned space to total capacity. To make it into a percentage, we just have to multiply it by 100. The super metric editor does not always follow strict order of operations rules so it's a good idea to use parenthesis to ensure that the calculations are performed as expected.

Place your cursor at the beginning of the super metric definition and type the ( key on your keyboard for the opening parenthesis

1. Place your cursor at the end of the super metric definition and type )*100 on your keyboard
2. Click the blue **Show Formula Description** button to see a more readable form of the super metric definition.
3. Verify: `(${this, metric=diskspace|provisioned}/${this, metric=diskspace|capacity})*100`

### Visualize Your Super Metric

Within the super metric editor, you can visualize how your super metric would have been calculated over the past several data points (default is six hours but can be changed using the calendar icon above the graph).

1. Click the **Visualize Super Metric** button (small graph icon)
2. The list of objects to visualize the metric for is based the object type selected in the Object Types pane. You should still have the Datastore object type selected
3. The Objects pane will list all instances of the selected object type. In our case we only have one datastore. Click the **RegionA01-ISCSI01-COMP01** datastore object.

4. Note the time series data points and graph showing what your super metric value would have been over the past six hours.

5. You are done. Click **SAVE**

### Assign the Super Metric to an Object Type

The final step within the super metrics area of vRealize Operations is to make the super metric available to one or more object types. In our case we want to make this super metric available for use with datastores.

Note that there is one final step that must be done before the super metric will actually be calculated for objects and that is to enable the calculation in a policy. We will cover enabling super metrics in policies near the end of this lesson.

1. Make sure that **Object Types** is underlined.
2. Click the line to select your new Datastore Subscription Percentage super metric.
3. Click the '+' icon to add an object type.
Select the Object Type

1. Type **datast** (or some portion of the object type)
2. Click **Datastore** under the vCenter Adapter category to select that object type

3. Click **SELECT** to save the selection
Verify the Object Type

Note that your new metric is now available for the Datastore object type.

You have created a metric to determine the subscription percentage and made it available for use with datastore object types.

Create a Super Metric Based on a Looping Function

In this example we will get a bit more advanced by using a looping function in our super metric. A looping function works on more than one value. Suppose you want to know something like the average CPU utilization of all VMs on a host. Or the maximum or minimum memory utilization across all VMs in a cluster. A looping function allows you to collect from all descendent VMs in this case instead of working with a single object.

Looping functions can return either a single data point or a set of data points depending on the syntax of the function arguments. We will learn more about that in a bit.

vRealize Operations has the following looping functions available for use in super metrics:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg</td>
<td>Average of the collected values</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>combine</td>
<td>Combines all the values of the metrics of the included objects in a single metric timeline</td>
</tr>
<tr>
<td>count</td>
<td>Number of values collected</td>
</tr>
<tr>
<td>max</td>
<td>Maximum value of the collected values</td>
</tr>
<tr>
<td>min</td>
<td>Minimum value of the collected values</td>
</tr>
<tr>
<td>sum</td>
<td>Total of the collected values</td>
</tr>
</tbody>
</table>

Let's create a super metric to find the maximum CPU usage percent across all VMs running in a cluster (or host or datacenter).

**Add a New Super Metric**

Let's get started.

1. Click the + to add a new super metric
Add the VM CPU Usage (%) Attribute

The metric we are interested in here is the CPU usage % of VMs. Let's add that metric to our super metric definition.

1. Type the metric name **Maximum VM CPU Usage (%)**
2. In the filter field start typing the object type with the metric you are interested in, virtual mach and press Enter
3. Click Virtual Machine in the Object Type pane to select it. You will now see metrics and attributes for that object type in the lower panes
4. In the Attribute Types filter field, type usage to filter the attribute list below to only show attributes with "usage" in their name
5. Click > CPU to expand the category
6. Double-click on Usage (%) to add that attribute to your metric definition
7. Verify: `{adaptertype=VMWARE, objecttype=VirtualMachine, attribute=cpu|usage_average, depth=1}`

Note that in this case we didn't use the This Object button because the super metric is referencing a VM metric but will be applied to a different object type (cluster, host or datacenter)
A Note About Metrics, Attributes and Specificity in Super Metrics

Regarding metrics and attributes, an attribute is metadata that describes the metric for the adapter to collect from the object. A metric is an instance of an attribute. The argument syntax (that you build by double-clicking attributes or metrics in the editor or by typing directly) defines the desired result.

For example, CPU usage is an attribute of a virtual machine object. If a virtual machine has multiple CPUs, the CPU usage for each CPU is a metric instance. If a virtual machine has one CPU, then the function for the attribute or the metric return the same result.

Super metric arguments can return a single data point when referring to a specific metric on a specific object (like when using the This Metric button) but often you won’t want to be so specific in your super metric calculation and most arguments will actually return a set of data points. If you want your super metric to return a single data point but the argument returns a set of data points, you can apply a looping function to find a single-value match from set of data points.

Think about having an set of numbers. If you take the average or maximum or minimum value for the set or if you count the number of values collected, the result is a single number. This is what looping functions will do.

In our case, we have so far configured this super metric to return a set of data points, each one representing the CPU utilization percent of for each of VMs that we are evaluating. If this super metric were going to be applied to a VM then we would have used the This Object button like we did with the datastore metrics in our previous example. But since we want to look at the metric for all VMs that are running on a host
or in a cluster or datacenter, the way our syntax is written it will return multiple data points.

Let's check our current argument syntax.

1. Click the **Visualize Super Metric** button
2. Notice the red exclamation mark to the right of the metric definition? **Hover** your mouse over the exclamation mark and see the error: **Cannot convert aggregated results to number**. This is an indication that your argument is returning a set of data points and can't be used in its current state.

### Convert the Super metric From a Set of Points to a Single Point

So how do we get our desired single data point from the set of data points? We use a looping function to match the one data point of interest. In our case, we are looking for the **maximum** CPU usage percent from all VMs on the host or in the cluster or datacenter.

Let's find the maximum value.

1. Place your cursor at the beginning of the formula definition and type `max(`
2. Note that you can also click the drop-down to see and insert available functions into your metric definition but you will still have to type your parenthesis
3. Place your cursor at the end of the formula definition and type `)`
4. Verify: \[ \text{max} \{ \text{adapterType=VMWARE, objectType=VirtualMachine, attribute=cpu|usage_average, depth=1} \} \]

5. Click the **Visualize Super Metric** button twice to de-select and then select it again. Note that we do not get an error this time.

6. Click any VM in the Objects pane

7. Note the resulting super metric visualization for your formula against that VM object

Why is it zero? As we have been discussing, this super metric isn't valid for a single VM. It needs to loop through a set of VMs in order to return a value.

### Visualize Metric for a Host

Let's visualize our metric for host.

1. Clear the Object Types filter field then type **host** and press the Enter key
2. Select **Host System** in the Object Types pane
3. Select any host in the Objects pane to visualize your super metric for that host
4. Note the results

Your resulting visualization will be different that what's show here but you should get a time-based graph with values. Each metric point represents the CPU utilization of the whichever VM on this host had the highest usage percentage at that time.
Visualize Metric for a Cluster

Let's check the metric for one of our clusters.

1. Clear the Object Types filter field then type **cluster** and press the Enter key
2. Select **Cluster Compute Resource** in the Object Types pane
3. Select any cluster in the Objects pane to visualize your super metric for that host
4. Note the results

Why is the value zero? It worked for our hosts but not for our clusters. The answer lies in the **depth** parameter in the super metric syntax.
Set the Super Metric Depth Parameter

The depth parameter in the super metric definition specifies how far down (or up) the object hierarchy to look for the specified object type (in our case VirtualMachine). So the default depth=1 works for a host because in the vSphere Hosts and Clusters hierarchy, a VM is one level below a host. But since a VM is two levels below a cluster in that hierarchy, the super metric returns a zero.

To be clear, the vSphere Hosts and Clusters hierarchy from top-down is: vSphere World -> vCenter -> datacenter -> cluster -> host -> VM.

So if we make depth equal to 3, this super metric can be applied at the host (one level above a VM), cluster (two levels above a VM) or datacenter (three levels above a VM). Note that a depth parameter of \( n \) will look at each levels in the hierarchy from one to \( n \) and will stop when it finds a match for the object type (virtual machine in our case). This will allow us to use our super metric with hosts, clusters and datacenters if we set the dept to 3. However, if you tried to apply this super metric with the depth=3 to a vCenter object, that would be four levels above the VMs and therefore would not return any results.

1. Click to place your cursor on your metric definition and use your keyboard to change the depth= value from 1 to 3
2. Verify: \[ \text{max}\{\text{adaptertype=VMWARE, objecttype=VirtualMachine, attribute=cpu|usage_average, depth=3}\} \]
3. Click on the cluster that is not highlighted in the Objects pane to show the visualization against that cluster.
4. Note the results. We have values again so it works.
5. Click **SAVE** to save your new super metric.

Note: If you want to look up in the hierarchy instead of down, you specify the depth value as a negative number.

**Assign the Super Metric to an Object Type**

1. Click the line to select your new **Maximum VM CPU Usage (%)** super metric.
2. Click the + icon to add an object type.

**Select the Object Type**
1. Type **cluster compute** (or some portion of the object type)
2. Click **Cluster Compute Resource** under the vCenter Adapter to select that object type

3. Click **SELECT** to save the selection

**Repeat this process** two more times to add Host System and Datacenter to the Object Types that will be have this super metric available for use.

**Verify the Object Types**

1. If you don't see all three object types, go back one step and add the additional object types.

As noted above, by setting the depth=3 in the syntax for this super metric, this super metric will work for any objects up to three levels above the subject object type (VM) so it makes sense to make this super metric available for all of those object types.
Create a Super Metric With Advanced Operators

Super metric definitions can include conditional logic and looping functionality as well.

The **where** clause can be used to point to a different metric of the same object, such as

```
where = "metric_group|my_metric > 0"
```

A conditional if-then-else operator can be used to return different values depending on a condition. The format for a conditional operation is:

```
expression_condition ? expression_if_true : expression_if_false
```

The expressions in the if?then:else operation can contain other operators including other if?then:else operators. The result of the conditional expression is converted to a number and any value not equal to zero is assumed as true.

In this example, let's assume we want to know whether or not the CPU usage percent is between our acceptable range of 3% and 90%. If it is within that range then our metric should be set to 100 to indicate a good state. Otherwise the value should be set to zero. We will use what we have learned about using a looping function to extract a single data point from a set of data points and combine that with our if?then:else operators to get our resulting value of either 0 or 100.

Add a New Super Metric

Let's create our final super metric for the lesson.

1. Click the + to add a new super metric
Add the VM CPU Usage (%) Metric

1. Type the metric name **VM CPU Usage In Target Range**
2. In the filter field start typing the object type with the metric you are interested in **virtual mach** and press Enter
3. Click **Virtual Machine** in the Object Type pane to select it. You will now see metrics and attributes for that object type in the lower panes
4. In the Metrics filter field, type **usage** to filter the attribute list below to only show attributes with "usage" in their name
5. Click > CPU to expand the category
6. Click the **THIS** button because we will be applying the metric to VMs
7. Double-click on **Usage (%)** to add that metric to your metric definition
8. Verify: \( \{\text{this, metric=cpu|usage_average}\} \)
Let's see what the value of our metric is so far. Right now we are just looking at the CPU utilization percent.

1. Click the **Visualize Super Metric** button (small graph icon)
2. Click a VM in the Objects list
3. You should see a graph of that metric over the past six hours. Note that if your lab pod was started up less than six hours ago you will see zero values for that time frame.
4. Click on different VMs in the Objects pane to see if you can find a VM that has had CPU utilization values below and above 3% during this time period.

If you don't find a VM with values both below and above 3% that's fine. You will find some that are all below or all above 3%.

So how do we modify our metric definition to add logic to identify when the value is greater than 3%?
1. Click to place your cursor at the end of your super metric definition and type $>3$
2. Verify: $\{\text{this, metric=cpu|usage\_average}\} > 3$
3. Click on some VMs in the Objects pane
4. The visualization will now show that this metric will be zero when the condition is false (utilization is not greater than 3) and one when the condition is true (utilization is greater than three).
5. Great! Now let's add more logic.
Extending the Logic

We will now use the logical AND operator and copy then modify our existing expression to create an express that is true when the CPU utilization is above 3% AND below 90%. In other words, it is in our target range.

1. Click and drag across the entire metric definition to select all of it
2. Right-click and choose Copy to copy the text to the clipboard
3. Place your cursor at the end of the metric definition and type && (this is the logical AND operator).
4. Place your cursor at the end of the metric definition then right-click and choose Paste to append your initial metric definition to the end
5. Delete the >3 and replace it with <90
6. Verify: `${this, metric=cpu|usage_average}>3 && ${this, metric=cpu|usage_average}<90`
7. Click on some VMs in the Objects pane
8. The visualization will now show that this metric will be one when the utilization value is between 3 and 90. It will be zero otherwise.

This could be enough to indicate whether or not the metric is within the target range. However, remember from the beginning of this section that we wanted the metric to be either zero or 100 depending on whether it was not or was within the target range. We could just multiply the existing definition by 100 and that would meet the objective.
Add Conditional Logic

Let's use our if-then-else (or if?then:else) logic here. If the existing definition evaluates to 1 (true) then make the value 100. Otherwise, make it zero.

1. Place your cursor at the beginning of the expression and type an open parenthesis (.
2. Place your cursor at the end of the expression and type a close parenthesis ).
3. After that, type \(100:0\) That will be evaluated as if the preceding expression is true then return 100, otherwise return 0.
4. Your final super metric definition should be: \((${this, metric=cpu|usage_average}>3 \&\& ${this, metric=cpu|usage_average}<90) ? 100 : 0\)
5. Notice in the visualization graph that your metric is now either zero or 100 for any given data point.
6. Click **SAVE** to save your super metric.
Assign the Super Metric to an Object Type

1. Click the line to select your new VM CPU Usage in Target Range super metric
2. Click the + icon to add an object type

Select the Object Type
1. Type *virtual mach* (or some portion of the object type)
2. Click **Virtual Machine** under vCenter Adapter to select that object type

![Select Object Type](image)

3. Click **SELECT** to save the selection

**Verify the Object Type**

![Super Metrics](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Formula Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM CPU Usage in Target Range</td>
<td>(This Resource: cp/ui/usage_average)&gt;10</td>
</tr>
<tr>
<td>Children Badge Health</td>
<td>sum(DCC Health: BadgeHealth)-sum</td>
</tr>
<tr>
<td>Database Subscription Percentage</td>
<td>(This Resource: diskSpace/provisioned)/</td>
</tr>
<tr>
<td>Number of Children</td>
<td>count(DCC Health: BadgeHealth)+cti</td>
</tr>
<tr>
<td>Maximum VM CPU Usage (%)</td>
<td>max(Virtual Machine: CPUUsage)</td>
</tr>
</tbody>
</table>

Note that your new metric is now available for the Virtual Machine object type
Enable Super Metrics on Objects and View the Results

We just created three different super metrics. Hopefully you learned a bit about the power of super metrics and have some ideas about how you can use them in your vRealize Operations environment. There are a lot of functions and operators that we didn’t get a chance to look at but you can find out more about them in the product documentation and find some of the many blog entries on the topic.

Before we can use our super metrics, they first need to be enabled in one or more policies. We will take a look at how to do that next.

Policies are a set of rules that you define in order to specify how vRealize Operations analyzes and displays information about the objects in your environment. You can create, modify, and administer policies to determine how vRealize Operations displays data in dashboards, views, and reports. One section in the policies dictates whether or not metrics are collected from the endpoints (like vCenter) and whether or not super metrics that have been created and made available are calculated for an object type. You can create create groups of objects in vRealize Operations (custom groups) and then apply different policies to different custom groups. That means that any super metrics you create can be calculated for all or any subset of your environment.

In this lesson we will modify the policy that is being used for all objects in the lab environment (the HOL Default Policy) to enable the calculation of your super metrics.

Open The Policy Editor

1. Click Policies in the Navigation pane
2. Click the Policy Library tab
3. Double-click the HOL Default Policy to edit it
1. Click **Collect Metrics and Properties**
2. This section of the policy lists all of the metrics, properties and super metrics (attributes) that are defined in vRealize Operations. You can see the attribute type, which adapter type they are defined by, the object type they apply to, the state (whether or not they are enabled), whether they are defined as a key performance indicator (KPI) and whether dynamic thresholds (DT) are calculated for them.

Note that a discussion about KPIs and DTs is outside of the scope of this lab module.

### Hide Unnecessary Columns

1. Select all columns except for **Type** and **State**.
2. Click **OK** to apply the changes.
Due to the relatively low resolution of the user interface in this lab environment, it's easier to continue if we hide some columns in the table that we won't need.

Click the icon in the lower-left corner of the table to display the Show Columns window

1. Click to de-select the **Adapter Type**, **KPI** and **DT** columns
2. Click **OK** to save your selections

**Filter To Show Only the Super Metrics**

![Attribute Filter](image)

1. Click the **Attribute Type** to expand the selection filter
2. Click to uncheck the **Metric** and **Property** boxes

The attribute list will now only show super metrics
Super Metrics in This Instance

This is a list of all super metrics that are defined in this vRealize Operations instance. You will see two metrics that were created in the environment prior to the work you did and the three that you created earlier. Notice that for each super metric that you created, you will see a line for each of the object types that you enabled the metric for plus "All Object Types". Take your Maximum VM CPU Usage (%) super metric for example.

1. This metric is available for All Object Types, clusters, datacenters and hosts.
2. The state for all of your super metric lines shows disabled and indicates that the state is inherited

A policy can be configured in a hierarchical manner that allows some or all of the settings to be inherited from its parent policy. Any super metric that you create will be disabled at the root policy and will have to be explicitly enabled in any policy hierarchy tree where you want to use it.

IMPORTANT: It is very rare for users to want to enable a super metric for All Object Types. Think about it - all of the super metrics that we created used metrics from specific object types if you enable them for all object types, vRealize Operations will try (and fail) to calculate them for objects where it would make no sense to do so. For example in this small lab environment there are 67 different object types including things like AIX, cache disk (for vSAN), business group, vRealize Business for Cloud cluster, etc. Be very cautious about enabling metrics for all objects because usually that creates a large number of unintended metrics in the system that won't have values because the calculations are not applicable.
Enable Your Super Metrics In The Policy

To enable your super metrics in this policy:

1. Click the **Object Type** column header to sort the table on that column
2. While holding down the Ctrl key on your keyboard, click each of the super metric entries that is NOT for All Object Types
3. Click **Actions** to expand the menu
4. Click **State** to expand the menu
5. Click **Enable** to enable the selected metrics
Verify the State of the Metric Collections

1. Note that the collection state for the super metrics are now all set to Local. Local means that the setting is overriding the inherited state in this policy.
2. Click **SAVE** to save changes to the policy

Search For the Datacenter Object

Let’s go look at an object that we have enabled a super metric calculation for. We will check the RegionA01 datacenter object.
1. Click the magnifying glass to open the search box (not shown with the box open in this screen shot)
2. Type region. The search list will populate with partial text matches for the word.
3. Click RegionA01 in the Datacenter section.

**View the Super Metric Graph**

![Graph Image]

Note that it will take two collection/analytics cycles after saving the policy in vRealize Operations before you will see calculated values here. The first cycle after saving the policy will create the super metric attribute on the object. On the second and subsequent cycles, the numeric value will be calculated and stored.

The default collection/analytics cycle in vRealize Operations is 5 minutes. However, we have set the frequency to run every one minute in this lab environment so it will take a minute or so before you even see the Super Metric category and another minute before you start seeing any values on the graph.

1. Click the All Metrics tab for the RegionA01 datastore.
2. Click >All Metrics to expand the section.
3. Click >Super Metric to expand the section (this should appear within one minute).
4. Double-click your super metric name Maximum VM CPU Usage (%) to display the graph.
5. Click the graph refresh button to update the graph until you are satisfied that the metric is being calculated every minute.
Note: You can click and drag across the current time on the graph to zoom in from the default timeline of the last 7 days.

Feel free to check VM, host and cluster objects to see that your other super metrics are also being calculated.
Export and Import Super Metrics

So far you have created some super metrics and enabled them for calculation in a policy then verified that the metrics are being calculated. The final topic in this module explores how to export and import super metrics. You might create some super metrics in a development instance of vRealize Operations and then import the metrics into a production instance. Or maybe you want to share them with a colleague who works for a different company. Perhaps you just want to back them up somewhere. In any case, it's easy to move super metrics between instances.

Navigate Back to The Super Metrics Page

Let's return to the super metrics page.

1. Click Administration
2. Click > Configuration to expand the section
3. Click Super Metrics
4. Select on or more super metrics that you want to export
5. Click the gear icon to expand the menu
6. Click Export Selected Super Metric

Note that this is also where you would import super metric(s) from a saved file.
Choose a Location and Save the Export

A Save As dialog box will open.

1. Click **Desktop** as a location to save the exported file
2. Click **Save**

Now we have a single json file on the desktop that contains everything necessary to import back into another vRealize Operations 6.7 instance.
Open the Exported File

Let's take a look at the format of the export file.

1. Click the Windows Explorer icon in the taskbar
2. Click Desktop to show files in that location
3. Right-click the supermetric.json file
4. Click Edit with Notepad++ to open the file in that application
View the Exported File

Here you can see what the json-formatted super metric export file looks like.
Conclusion

That's it. You have created some supermetrics, enabled them in a policy and seen the results of the metric calculations. Finally, you saw how to import and export super metrics for portability or back up reasons.

You've finished module 4

Congratulations on completing module 4.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,

1. Click on the **END** button at the top of the page.
Module 5 - Using the vRealize Operations API (30 minutes)
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 programmatically 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 the Chrome Browser

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

How the vRealize Operations API Works

You use a Web browser to communicate with the vRealize Operations Manager analytics engine, either through the product user interface or through API calls.

The adapter instance collects data from objects in your monitored environment. The vRealize Operations Manager analytics engine processes the data and displays the complete model in the graphical interface.

The API is most useful when there is a need to automate a well-defined workflow such as repeating the same tasks to configure access control for new vRealize Operations Manager users. The API is also useful when performing queries on the vRealize Operations Manager data repository, such as retrieving data for particular assets in your monitored environment.
virtual environment. In addition, you can use the API to extract all data from the vRealize Operations Manager data repository and load it into a separate analytics system.

**Say hello to the vRealize Operations REST API!**

![Image of vRealize Operations API interface]

Let's dive straight into the vRealize Operations 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. Open a new tab in the browser and type `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.
Let's have a look at the API!

2. 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.

**Browse the API functions**

![API functions screenshot]

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 `enumerateAdapterInstances`

You should see something similar to the screen shot above. **You may have to scroll the right-hand pane to see 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.

3. Click **Show** to see a sample response.
Sample Response

When you click "Show" you are presented with a sample of what the output would be if you made the actual request.

Making our first vRealize Operations API call

REST API documentation includes request, response types and API endpoints.
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 `enumerateAdapterInstances`, we'd have to use the URL https://vrops-01a.corp.local/suite-api/api/adapters

In the browser:

1. Open a new tab and type [https://vrops-01a.corp.local/suite-api/api/adapters](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 **Sign In** 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.

1. 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 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.

**End Point Adapter Output**

1. If you continue to scroll down the output you will see additional adapters. In the example above, the Endpoint Adapter is the next adapter listed.

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

To make it easier to find a specific topic such as the API calls for users let's use the Find feature of the browser.

1. For the Chrome browser, click the dots in the upper right hand corner.
2. Click **Find**
3. When the search bar appears, type **User** to search for all instances of user in the API list.
Getting List of Users

1. Scroll down until you find the getUsers API.
2. Click **getUsers**
3. In the browser address bar type: `https://vrops-01a.corp.local/suite-api/api/auth/users`
Analyzing the Results

1. Scroll the results until you find the username "hol@corp.local". Using this method, you can look up details about a particular local user. These details can then be used to modify user properties.
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, 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. If you receive a notice about application updates click the **X in the upper right corner** to close the notice.

Postman Setup

Check the environment setup by:
1. Verify the environment is configured for `vrops-01a.corp.local` as shown in the image above.
2. Verify `Collections` is selected.
3. Click the folder `vRealize Operations 6` to expand the contents.
4. Click the folder `Resources` to expand its contents.

**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. Click **Get VirtualMachine Res by Name** in the Resources folder.
2. Click **Params** to modify the VM name.
3. Enter `app-01a` for the value of the Key field.
4. Click on "Send"
5. The results are a list of all resources pertaining to the VM `app-01a` and can be found in the Body window as shown. You will need to scroll the window to see all the results.
Notice the results are 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.

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 "Virtual Machine".

1. Click on the link.
2. Click Send
If your results contain Unauthorized or no valid credentials you need to set the query to Basic credentials.

1. Click the **V** to view a list of alternative authentications.
2. Click **Basic Auth**
3. Check **Show Password** and verify the Username is **admin** and the password is **VMware1!**
4. Click "Send" to send the request again.
1. Click the **Magnifying Glass** to start a Search.
2. Type **relationships** in the search bar as shown. Scroll down the body until you find the first link pertaining to relationships.

## 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 a 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 screenshot 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.

Let's create a new resource.

1. Click on Create Resource in the collection.
2. Click on the Body tab as indicated in the screenshot 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.
Let's open the Live Instance of vRealize Operations 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.

**Login to vRealize Operations Live Instance**

To review the payload that was created with the Postman tool we go to the "Inventory Explorer" in vRealize Operations

1. Click **Administration**
2. Click the V to expand the **Configuration** section.
3. Click **Inventory Explorer**. This lists all of the objects that exist in Operations.
4. Type **power** to filter the objects.
5. As the image shows there are 3 new objects created: Main 110V supply, Power Monitor, and Power Supply

**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 Operations, we will execute a simple PowerShell script.

1. Click the Windows Start key
2. Click Windows PowerShell
3. Click Run ISE as Administrator
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. Click **Add metrics** in the Resources folder.
2. 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.
3. 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.
4. Click the **Create Resource** tab from 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. In the body of Create Resource tab

1. 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.
2. Click **Add Metrics** to return to that tab
Let's complete the API call and send it to vRealize Operations.

1. 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.
2. Click **Send**.

**NOTE:** The POST was successful if there are no error message returned.
Search For the New Object in vRealize Operations

In the vRealize Operations UI. Click the magnifying glass icon to open the search bar.

1. Type `main` to search for the newly created resource.
2. Click `Main 110V Supply`

**Metric Chart**

1. Click All Metrics
2. Click `V` to expand the All metrics section.
3. Click `V` to expand the voltages section
4. Double-Click **output** to add the metric to the chart.
5. Click the **Calendar icon** to change the time range.
6. Click the **V** to see the list of ranges.
7. Click **Last hour**
8. Click **GO** to update the metric chart.

Note: There currently is only one data point for the metric. If you want to add additional metrics follow these additional steps.

**Add additional data points**

To add more data points you must redo the following steps. If you need more detail just refer back to the previous pages.

1. Cut and Paste a new timestamp for each data point. This is generated by running the Power Shell script you ran earlier.
2. Edit the 20 to another value. This is the voltage reading and will be reflected in the graph.
3. Click **Send**

Add a few more data points by entering different values for the timestamp and for the voltage. Remember to click **Send** each time.
View the Metric Graph

Switch back to vRealize Operations.

1. Click the **Refresh icon** to update the chart with the new values. Note: It may take a couple minutes for the data to appear.
Conclusion

In this module you explored the APIs offered with vRealize Operations Manager. This module was to introduce you to how to access the API's and how it can be useful when performing queries on the vRealize Operations Manager data repository, such as retrieving data for particular assets in your virtual environment.

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.

You've finished module 5

Congratulations on completing module 5.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

• Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)
• Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)
• Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)
• Module 4 - Super Metrics in vRealize Operations (45 minutes)
• Module 6 - RemEDIATE Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)
• Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)
• Module 8 - Users and Roles in vRealize Operations (30 minutes)
• Module 9 - PowerCLI for vRealize Operations (60 minutes)
• Module 10 - RemEDIATE Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)

Or if you want to end your lab,

1. Click on the END button at the top of the page.
Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)
Introduction

Would you like to extend vRealize Operations alerts as a mechanism for automatically fixing the identified issue? These are possible with the integration of vRealize Operations and vRealize Orchestrator. The lab module will take you through an example of the self healing datacenter. For this lesson we will manually stop a web service and then using the resulting alert in vRealize Operations a workflow will be launched in vRealize Orchestrator to restart the service.
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance**: Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance**: Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs**: Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the **Chrome** icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the vRealize Operations - Live Instance link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
Self Driving Operations with vRealize Operations

Both vRealize Log Insight and vRealize Operations support webhooks. This module will walk you through configuring webhooks for vRealize Operations and fixing a database service which has stopped running. Webhooks allows integration with other solution that has a REST API available. Ticketing, notification, chat and other capabilities can be leveraged by vRealize Intelligent Operations. Extending that capability can make your datacenter a self-healing datacenter!

If you are a user of vRealize Operations, you know that it can monitor your infrastructure, server OS, applications and more. But as you know, monitoring is only part of the answer. Wouldn’t it be much better to have vRealize Operations attempt some simple fixes before giving up and calling for human intervention?

Using Webhooks to Automate Remediation

Currently, vRealize Operations alerting can trigger notifications via email, SNMP trap or REST notification. The REST notification plugin allows you to interact with REST APIs. The screen above shows the different methods you can find in vRealize Operations.
The truth is the REST Notification Plugin is not very robust and does not provide a lot of control over format of the REST call. You plug in the URL and some credentials and vRealize Operations will try a PUT and POST against that URL. Also the plugin will append the alert ID onto the end of the URL.

The image above shows adding a REST Notification Plugin instance in vR Ops. Note the limited options available; there is nowhere to format the request body or control the method used.
REST API

This means that unless your REST API is expecting the request in the way vR Ops would like to send it – it will fail.

For example, Orchestrator has a robust REST API. To launch a workflow via that API, take a look at the image above.

Enter Webhook Shims!

One way is to translate the REST call from vRealize Operations into a REST call that can be consumed by Orchestrator. Enter the [Webhook Shims for vRealize Operations](#). This diagram shows how it works:

Webhook Shims enables self-healing with vRealize Operations through an Orchestrator shim. The Orchestrator shim takes the simplistic input from vR Ops REST notification and rebuilds the request in the format expected by Orchestrator.
Understanding The Lab Environment

From the Home screen when you first logged in.

1. Click **Environment** to navigate to the vRealize Operations Manager Environment Overview.
2. Click **vSphere Hosts and Clusters**
Expand vSphere World

1. Click **vSphere World** and continue to expand the tree until you can view the VM's running on the esx hosts as shown in the picture. This lab will use VM's web-01a, app-01a, db-01a and photon-01a.
Cluster and VM View

3 of the virtual machines represent a 3 tier application. As you can see the web-01a and app-01a are running on esx-01a host. The database server db-01a is running on esx-03a.
3-Tier Application

The components that make up the application are the following.

**Web App Components - Default Configuration**

- **web-01a** - 192.168.120.30
  - This machine is a reverse proxy, listening on port 80
  - All traffic is proxied to http://app-01a.corp.local:8080/
  - The self-signed SSL certificate uses the name webapp.corp.local

- **app-01a** - 192.168.120.20
  - This machine runs a web server, listening on port 8080
  - The “application” configured here makes a connection to db-01a on port 3306 (MySQL)

- **db-01a** - 192.168.120.10
  - This machine is a database server, listening on port 3306
- It runs MariaDB (a non-Oracle fork of MySQL), with a single database.

**The Web Page for the 3-Tier Application**

Let's view the contents of our application. Open a new tab in the browser.

1. Click **HOL - Multi-Tier Application**. The web page should return results from the database server with employee details.
2. Click **Add New Employee** to enter a new record.
Create a New Record

1. Click Submit once you fill out the form to add a entry to the database.

Photon-OS-Container-Host

1. Click photon-01a to see the summary page of the VM.
The photon-01a is the virtual machine which contains a docker container that includes the webhook shim run environment. The webhook container will be listening for REST notifications sent from vRealize Operations Alert then reformatting the information and calling the vRealize Orchestrator API. vRealize Orchestrator will then run a workflow called "Restart Service from Alert" to restart the service.

**db-01a Virtual Machine**

![Virtual Machine Screen](image)

1. Click the **db-01a** virtual machine. The summary page is shown.
2. As you can see in section 2, currently there are no alerts for the db-01a VM and so the health status is green. You will see an alert appear once the MariaDB database service is stopped. Let's verify that the process is running in the db-01a console.

**Verify The MariaDB Service Status**

![PuTTY Icon](image)

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 db-01a VM

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

1. Click the db-01a entry shown in the list of saved sessions.
2. Click on **Load**
3. Click **Open** to start the putty session for db-01a.
Verify the MariaDB Process is Running

Once the console is open. We can check the process is running by starting a script.

1. Enter `ls` to view the contents of the directory. There are some scripts you will be using for this lab.
2. Enter `./check_running_service` to verify the database service is running as shown in the image above.
3. Click `-` to minimize the putty session. You will come back to this window later in the lab.

Start another Putty Session

1. Right-Click on `root@db-01a` putty session. This will bring up a menu.
2. Click on PuTTY to start a new session.
PuTTY session for Photon-01a

1. Using the Scroll bar to find photon-01a.corp.local
2. Click photon-01a.corp.local
3. Click Load
4. Click Open to start the putty session for the photon-01a.corp.local VM.
Start Docker Container

In the console session we need to start the container where the webhook shim 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.

1. Enter `ls` to see the directory contents. Notice there are 2 scripts. **edit-docker-shim.sh** and **start_webhook.sh**.

2. Enter `./start_webhook.sh`. This command will start the docker container called "webhook". It only takes a couple seconds to start the container. The result is shown as a listener on port "5001".
Verify That the Webhook Shims Are Running

To test the webhook shims and that the listener is working, complete the following steps.

1. Open a new tab in the browser.
2. Click the bookmark or enter `photon-os-container-host:5001` as the URL. The listener of the webhook container responds with documentation of all the shims available.
3. 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. You can close the webhook tab since you are done with it.
Webhook Response

1. Notice there a couple more lines displayed from the listener when requested the webhook documentation in the last step.

Explore the Workflow in vRealize Orchestrator

1. Lets go take a look at the workflow that will be triggered. Double-click the vRO Workflow icon on the Desktop to launch the client.
vRealize Orchestrator Login

The Host name and the User name should already be filled in.

1. Enter **VMware1!** for the password.
2. Click **Login**
Viewing the Workflows

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

**Restart Service from Alert Workflow**

1. Click the **Webhook Shim Sample** folder to expand the contents.
2. Click the **Restart Service from Alert** workflow to see the contents of the workflow.
3. Notice the workflow ID? This is the key to connecting the vRealize Operations alert to running the correct workflow in VRO. You will see where this is used later in the lab.
When the MariaDB service is stopped on db-01, a vRealize Operations Alert is generated. The Alert is sent to the webhook shim which then sends it to vRealize Orchestrator. For the VRO workflow to successfully restart the MariaDB service it needs 3 pieces of information:

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

1. Click the **Schema** tab to see the steps comprise the workflow.
2. Click the **Zoom** pull down menu and change it you cannot see the workflow steps. Usually 75% works well.
3. Slide the **Scroll Bar** on the bottom of the schema window to see all the steps for the workflow. Notice the last step (not shown) of this workflow is a callout to the vROPS Start MariaDB workflow. This is the workflow which actually executes the restart service.
4. Click **Get Alert from ID**
5. This area outlines the API calls that are used to obtain the 3 pieces of information (Alert ID, Resource Id, Resource properties).
6. Click the **vROPS Start MariaDB** step. This is the second workflow that VRO runs.
vROPS Start MariaDB Workflow

1. Click the vROPS Start MariaDB workflow.
2. Click the Schema tab to see the steps.
3. Click Execute SSH Command
4. Click the IN tab to see the parameters that are passed to the SSH command.
5. This area shows the hostname which was resolved from the 3 attributes in the first workflow and then a hostname was sent to this workflow.

Minimize VRO. We will take a look at it after the workflow has completed. Now back to vRealize Operations.
vRealize Operations Configuration

Return to the vRealize Operations UI in the browser. Let’s review the configuration of the Outbound settings for messages for alerts.

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 **VRO Restart Service from Alert**
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 of the dialog box** so you can see the entire URL string as shown above.

2. Notice the URL is pointing to photon-01a where the webhook container is running and listening for outbound messages from vRealize Operations.

3. The other thing to notice is the hex string which ends in 0015. This is the ID of the "Restart Service from Alert" workflow in vRealize Orchestrator.

4. Click "CANCEL" since there are no changes needed.

Search for db-01a

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

1. Click the "**Search icon**" found in the upper right corner of vRealize Operations.
Search for db-01a VM

1. In the Search window begin typing **db-01a**. vRealize Operations will show objects containing that string.
2. Click **db-01a** located under the Virtual Machine heading. You may have to scroll down the list for the Virtual Machine.
3. **Note:** This is the **Refresh** icon and will be helpful for updating the dashboard once the service has been stopped.
The db-01a.corp.local Summary

This is the summary page of the db-01a.corp.local VM that is being monitored by the Endpoint Operations agent. The "children" of this operating system object are the Endpoint Operations Agent.
1. Click **Alert** to review the alert definition.
2. Click > to expand Alert Settings.
3. Click **Alert Definitions** A long list of predefined alert definitions will be displayed.
4. Filter the list by entering `down` and **Enter on the keyboard**. Notice the filter is set to "Name: down" and shows the alerts with down in the name.
5. Click the alert **A Monitored Service is Down** After clicking the alert notice additional details about the alert are shown in the lower section.
6. Click the **Pencil icon** to view the symptom definition of the alert.
Alert Definition

1. Click 3. Alert Impact. Notice that the alert will reflect a Health Badge changing from green to red when the MariaDB service goes down.
2. Click CANCEL after reviewing the alert since there are no changes needed.

To learn more about creating new alerts, enroll in lab HOL-1901-04-CMP Module 2.

Alert Notifications
Now to create a notification setting that instructs what vRealize Operations will do with the alert when it is triggered.

1. Click **Notification Settings**
2. Click the + icon to create a new notification.

### Create Alert Notification

1. Enter **MariaDB Down Notification** for the name of the rule. A notification rule instructs vRealize Operations what to do when the alert defined earlier is triggered.
2. For Method, click the V for the drop down list and click **Rest Notification Plugin** as shown.
3. And then click V for the drop down list and click instance **VRO Restart Service from Alert** for our Method of notification.
4. Click the V for the drop down list of the Notification trigger and select **Alert Definition**.
5. Finally click **SELECT AN ALERT DEFINITION**
1. Similar to before, enter "down" for the filter to reduce the list to alert definitions with "down" in the name.
2. Click **A Monitored Service is Down**
3. Click **SELECT**
1. Review the settings for the notification and click "SAVE"
Arrange the Screen

Before stopping the MariaDB, let's arrange your screen similar to above. This will allow you to see what is happening in the environment.

1. Click **BACK** to return to the db-01a VM Summary page. Notice the db-01a is still showing a Green health state.
2. Click the **photon-01a.corp.local** putty session that is minimized. You may want to drag the edges of the putty session to change the size so you can still see the db-01a Green health status.
3. Click the **db-01a.corp.local** putty session that is minimized. Again you may want to move and resize the session.
Script to Stop MariaDB

1. Type `./mariadb_stop.sh` in the db-01a.crop.local putty session to start the script which will stop the service.

Service Not Running

The result of the script is showing that the service is inactive. Also the script contains the watch command which will update every 2 seconds to show the status of the MariaDB service. Let's look at the 3-Tier application.
3-Tier App Error

1. Go back to the HOL-Multi-Tier tab on the browser.
2. Click the browser Refresh icon.

You can see the web page no longer returns the inventory of the database server since the database service is down.

Now click the vRealize Operations tab in the browser, then the photon-01a console session and then the db-01a console session to rearrange the screen so you can view everything happening.

Webhook Output
1. When the Alert is triggered the webhook listener responds to the alert which is shown in the photon-01a session. When you see output similar to what is shown you should refresh the vRealize Operations Summary page.

2. Click the **Refresh icon** in vRealize Operations. The Health badge is now Red representing a down service.

### Alert

View the vRealize Operations Manager UI. You may need to Click the **Refresh** icon 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.

1. Once the Health badge is red you will see alert is displayed indicating the service is down.
Health State is Returned to Normal

Once the alert shows in Operations it doesn't take long before the workflow is started and service is restored. If you review all of your windows you notice everything is indicating that the environment is healthy again.

1. Once VRO has completed the workflow, the db-01a console session shows **MariaDB is active** again.
2. If you click the **Refresh icon** in vRealize Operations the db-01a Health status will again be green. Also notice the Alert is cleared from the Summary page as shown in the image.
vRealize Orchestrator Client

Let's look at the vRealize Orchestrator window.

1. Click V to expand the contents of 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 a Workflow Run Event.
3. Click the Logs tab as shown. You will see the output of the workflow including the command that was executed to restart the service.
Clean up Environment

This concludes this module but there is some cleanup that needs to be completed.

Switch to the photon-01a putty session.

1. Enter Ctrl-C to get back to command prompt and then at the command prompt enter `docker stop webhook`
2. Enter `docker ps` to check that there are no running containers.
3. Close the db-01a putty session
4. Close the vRealize Orchestrator Client.

Clean Up Environment Continued
Also the Notification that you setup should be deleted. Switch to the vRealize Operations tab in the browser.

1. Click **Alerts**
2. Click **Notification Settings**
3. Click **MariaDB Down Notification**
4. Click **X** to delete the notification.

That's It.
Conclusion

In this module you extended vRealize Operations alerts as a mechanism for automatically fixing a identified issue. The lab module took through an example of the self healing datacenter by manually stopping a database service and then using the resulting alert in vRealize Operations to launched in vRealize Orchestrator automatically to restart the service.

You've finished module 6

Congratulations on completing the module.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)
- Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)
- Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)
- Module 4 - Super Metrics in vRealize Operations (45 minutes)
- Module 5 - Using the vRealize Operations API (30 minutes)
- Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)
- Module 8 - Users and Roles in vRealize Operations (30 minutes)
- Module 9 - PowerCLI for vRealize Operations (60 minutes)
- Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)

Or if you want to end your lab,

1. Click on the END button at the top of the page.
Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)
Introduction

The management pack for vRealize Orchestrator (vRO) provides access to the vRealize Orchestrator workflows for remediation actions and an ability to launch workflows directly from the vRealize Operations Manager (vROps) user interface.

In this module we are going to take a look at the integration between vRealize Operations Manager and vRealize Orchestrator. We will also execute a simple workflow.
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance**: Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance**: Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs**: Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click to skip ahead.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

1. Click the **Chrome** icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the **vRealize Operations - Live Instance** link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
Launching a workflow from vRealize Operations

In this lesson, we are going to walk through executing a vRealize Orchestrator (vRO) workflow directly from vRealize Operations (vROps).

Management pack for vRealize Orchestrator offers an out of the box self-healing for various VMs, Hosts and Datastore issues while providing an ability to add some customized remediations with ease. End result is reduced complexity and faster time to resolution. Integration of vRO capabilities with intelligent operations of vROps enables you to map the vROps alerts with the remediation workflows in vRO and automagically resolve operational issues.

Locating the workflows

Viewing the workflows

1. Expand vro-01a and com.vmware.library.vcenter

![Diagram showing vRealize Operations Manager interface with workflow locations marked]

Page 261
2. Expand the navigation pane to show the names of the workflows

Here you can see the extensive list of automated tasks you can kick off from here.

**Execute a workflow**

1. Select **Create host folder**

This workflow is new with the latest vRO management pack.

**Actions**

1. Click on **Actions** drop down
2. Select **Execute Workflow Action**
Workflow Window

1. The parameters needed to execute the workflow.
2. Clicking the drop down will reveal the vcenter server here: vcsa-01a.corp.local

Next we are going to find what inputs are needed to execute this particular workflow within vRO

Open vRealize Orchestrator client

1. On your desktop double click the vROWorkflow shortcut

This will launch the thick client for vRealize Orchestrator
Log into vRO

1. Enter password  VMware1!
2. Click Login
Finding the workflow you want

1. In the search field type: **create host folder**
2. Make sure the workflow is selected
3. Click on **Go to selection**
4. Click **Close**

This will not close your window, but it will open the folder and expose the workflow behind your window. This is just to show how to quickly find a workflow within vRO.
1. Click on the **Inventory tab**
2. Expand the **vSphere vCenter Plug-in** as shown

Added bonus: If you open the vCenter client in Chrome you will not see this folder. It's only exposed in the vCenter MOB, Browse Datastore, and here in vRO. At least for our purposes.
Execute the workflow

1. Enter: *host, EasterEgg*
2. Click *Begin Action*
3. Optional, click on The Task ID to see status

Go back to your open window for Execute WorkFlow Action in vRealize Operations. Now that we know what is needed for the required parameters lets enter them here.
View the file in the vCenter server

1. Click on the HTML5 Client shortcut in Chrome
2. Select Use Windows session authentication
3. Click Login

Seeing is believing

1. Select Host and Cluster
2. There is your new folder!

This was a very simple example of a new workflow available in vROPs.
Example of new workflows

1. Click on the **Environment** tab on the home page of vROps
2. Then select **vSphere Hosts and Clusters**
3. Expand the drop downs until you expose the VMs that live in the first host **esx-01a.corp.local**
Comparing the difference

1. Select the host **esx-01a.corp.local**
2. Select **Actions** drop down
3. All of the blue icons are added with the vRealize Orchestrator management pack

Without the management pack you would not have this functionality within vROps. This could save time and effort to complete some of these actions.

**Congratulations you have completed this lesson!**

Integration of vRO capabilities with intelligent operations of vROps enables you to map the vROps alerts with the remediation workflows in vRO and automagically resolve the operational issues.
Conclusion

The management pack for vRealize Orchestrator (vRO) provides access to the vRealize Orchestrator workflows for remediation actions and an ability to launch workflows directly from the vRealize Operations Manager (vROps) user interface.

In this module we are going to take a look at the integration between vRealize Operations Manager and vRealize Orchestrator. We will also execute a simple workflow.

You've finished module 7

Congratulations on completing the module.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,

1. Click on the **END** button at the top of the page.
Module 8 - Users and Roles in vRealize Operations (30 minutes)
Introduction

In this module we will take a deeper look at the part that users and roles play in vRealize Operations. We will look at the built-in role based access controls, and how you can create additional roles with extremely granular control. We look at how to grant access to objects within the environment, and will also review dashboard sharing between users and groups.
Log in to the vRealize Operations Live Instance

This lab environment is running three different instances of vRealize Operations and one instance of vRealize Log Insight. We have the different vRealize Operations instances in order to be able to work through different use cases that have unique requirements. The lab instances are as follows:

- **Live Instance**: Connected to the small running vSphere environment in the lab. There isn't a large inventory of objects in this instance but it allows us to interact with vCenter.
- **Historical Instance**: Running a 30-minute time loop of data that was captured in the past. This instance has a much larger inventory of objects but since it is not currently connected to a vCenter, we can't perform any actions here.
- **Blue Medora Management Packs**: Also running in historical mode, this instance has a large number of management packs from Blue Medora that allow us to see information from adjacent infrastructure (storage and physical servers) as well as operating system and application information.

In this lesson we will be using the Live Instance of vRealize Operations.

If you are already logged into the live instance of vRealize Operations, click **to skip ahead**.

**Open the Chrome Browser from Windows Quick Launch Task Bar**

1. Click the **Chrome** icon on the Windows Quick Launch Task Bar
Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the vRealize Operations - Live Instance link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
User Access and Preferences

To ensure security of the objects in your vRealize Operations Manager instance, as a system administrator you can manage all aspects of user access control. You create user accounts, assign each user to be a member of one or more user groups, and assign roles to each user or user group to set their privileges.

One of the more requested features of vRealize Operations Manager is how to create user specific content, for example a dashboard for leadership, where the user can see that content but no other information. This module will walk you through assigning content to a user, using the example of a user or group specific dashboard. We will create a user that has access to a single dashboard.

User Access Control

You can authenticate users in vRealize Operations Manager in several ways:

- Create local user accounts in vRealize Operations Manager.
- Use VMware vCenter Server users.
- Add an authentication source to authenticate imported users and user group information that resides on another machine.
  - Use LDAP to import users or user groups from an LDAP server.
  - Create a single sign-on source and import users and user groups from a single sign-on server.

Most customers use VMware Identity Manager - this is the preferred single sign-on (SSO) source for VMware solutions, and enables simple SSO configuration and management between the vRealize solutions.

VMware Identity Manager

vIDM is a service that extends on-premises directory infrastructure to provide a seamless Single Sign-On (SSO) experience. It is supported for all vRealize solutions, as well as many other VMware and non-VMware products.

VMware Identity Manager does not replace Active Directory, it integrates with it. Microsoft Active Directory integration will be configured in VMware Identity Manager instead of in the individual products.

Why not go directly to AD? Active Directory is an identity provider. vIDM is an identity service, which can connect to multiple identity providers, including AD.
Authentication Sources

To configure or view authentication sources within vRealize Operations Manager:

1. Click on **Administration**, in the title bar
2. Select **Authentication Sources** (under Access)
3. Highlight **VMware Identity Manager** in the list of configured Authentication Sources in the main pane
4. Click on the **pencil** icon to edit or view the source

Additional sources can be added here, but this lab has only one - a vIDM identity source.

Configuration Options
Here you can see the configuration settings that have been set up for the vIDM identity source in the lab. Further configuration is done on the vIDM appliance itself, including the AD or LDAP sources and users, groups and application entitlements. Once you have reviewed the options, cancel out of this screen:

1. Click **Cancel**

### Access Control

Once the identity source is added, users and groups from that source can be granted access to vRealize Operation Manager. Let’s take a quick look at this configuration:

1. Click on **Administration**, in the title bar
2. Select **Access Control** (under Access)
3. Note the **hol@corp.local** user

hol@corp.local is an Active Directory user that has access to vRealize Operations Manager via the following configurations:

- corp.local is added as an Identity source in vIDM
- hol is in an AD group called vRealize Users@corp.local
- The group and users were imported into vIDM
- The user (or group) was imported from vIDM into vRealize Operations Manager, and assigned groups and permissions

### Group-Based Security

vRealize Operations Manager provides user group-based security. With group-based security, you control the access privileges for each user group. You add users to user
groups, and assign access privileges to user groups. For example, one user group might be able to view only the objects integrity levels, and another user group might be able to configure objects. This simplifies privilege management significantly.

You must have privileges to access specific features in the vRealize Operations Manager user interface. The roles associated with your user account determine the features you can access and the actions you can perform. Roles are covered in the next lesson.

**User Preferences**

Some settings are per user, and can be modified by the user at any time. These are display preferences, such as colors for the display and health chart, the number of metrics and groups to display, and whether to synchronize system time with the host machine. You configure the user preferences on the top toolbar.

Open the preferences menu:

1. Select the **User** icon in the top right-hand title bar menu.
2. Select **Preferences**.
Change Preferences

Review the Preferences that can be changed on a per user basis, including Font, Time and other options.

1. Change the **Color Scheme** from Light to Dark.
2. **Save** changes.

One of the most requested features in vRealize Operations 6.7 was the return of the dark theme. Change it here, and if you like it keep it for the rest of the lab. Otherwise, you can switch back anytime.

*Note: You will only be able to change Preferences if you are in the Live instance. You will not be able to make any changes if you are logged into the Historical View Mode (HVM) instance of vRealize Operations Manager in this lab.*

**Lab Work**

In this module we are going to create a user with access to a single dashboard and no other area in the tool. Let's pretend we have a summer intern, who's role it will be to check the Optimize Performance dashboard each day for under-sized or over-sized virtual machines so that he can contact those users and request that they right-size their systems.
Add User "intern1"

Here we will take the first step by creating the user. For simplicity sake, we will create a local user:

1. Click on Administration, in the title bar
2. Select Access Control (under Access)
3. Verify you are on the User Accounts tab in the main pane
4. Click on the plus (+) icon, to add a user

For the sake of simplicity in the lab, we are creating a local user. However, in your environment you will likely be working with Active Directory users and the process of assigning permissions is the same.
Create the new user with the following properties:

1. User Details:
   - User Name: **intern1**
   - Password: **VMware1!**
   - First Name: **Summer**
   - Last Name: **Intern**
   - Email Address: **intern@acme.com**
   - Description: **Summer intern working on rightsizing project**
   - Do not disable or require password at next login

2. Click **Next**
Review Roles

The objects tab is where we would assign roles and objects to the group. However, we are not going to do this until the next lesson. For now, finish adding the user:

1. Switch to the **Objects** tab.
2. Click on the **Select Role** dropdown to see the list of available roles. Do not select any at this time.
3. Click **Finish**

Move to the next step.
User Warning

You will get a warning message that this user will not be able to login as it has no object permissions. Proceed anyway by clicking Yes.

We will assign the appropriate role and object permissions in the next lesson.

Add Group "interns"

Now we will create a User Group. Again, for simplicity sake, we will create a local group:
1. Click on **Administration**, in the title bar
2. Select **Access Controls** (under Access)
3. Select the **User Groups** tab in the main pane
4. Click on the plus (+) icon, to add a user

**Group Details**

Create the new user with the following properties:

1. **Group Details:**
   - Group Name: **interns**
   - Description: You can leave blank

2. Click **Next**
From the **Members** tab, add the **intern1** user to the interns group:

1. Verify you are on the **Members** tab
2. Click the checkbox beside the **intern1** user, to add to this group
Group Objects

The objects tab is where we would assign roles and objects to the group. However, we are not going to do this until the next lesson. For now, finish adding the group.

1. Click on the **Objects** tab and review the options, but make no changes
2. Click **Finish**
Group Warning

You will get an warning message that users in this group will not be able to login as it has no object permissions. Proceed anyway by clicking **Yes**.

We will assign the appropriate role and object permissions in the next lesson.

Login as "intern1"

For the sake of time, I will not have you attempt to login as **intern1** just yet. However, if you were to try you would see the error shown here: User does not have any roles assigned.
This is because although we created the user and group in this lesson, we have not assigned a role yet, so this user currently has no permissions within vRealize Operations Manager.

**Conclusion**

In this lesson we touched on the difference authentication sources that are supported in vRealize Operations Manager, and the Preferences that a user can set themselves. We also created a new user and new group, and added the user to the group.

In your environment, you will likely create the group and role before creating the user. We are doing it slightly out of order in this lab to clearly illustrate the relationships between users, groups and roles.

In the next lesson, to illustrate the power of role based access in vRealize Operations Manager, we will look at Roles and how to assign permissions to users. We will also look at Dashboard Management, and how to share dashboards with users.
Roles and Privileges

You must have privileges to access specific features in the vRealize Operations Manager user interface. The roles associated with your user account determine the features you can access and the actions you can perform.

vRealize Operations Manager provides several predefined roles to assign privileges to users. You can also create your own roles.

Each predefined role includes a set of privileges for users to perform create, read, update, or delete actions on components such as dashboards, reports, administration, capacity, policies, problems, symptoms, alerts, user account management, and adapters.

Predefined Roles

There are several pre-defined roles in vRealize Operations Manager. In addition, you can create your own roles. Roles are an efficient way to configure a standard set of privileges to a user or group of users.

Predefined Roles:

- Super user administrator
- Administrator
- ReadOnly
- PowerUser
- PowerUserMinusRemediation
- ContentAdmin
- GeneralUser-1 through GeneralUser-4

For more detailed information on these roles and the privileges assigned to them, refer to the documentation for vRealize Operations Manager.
View Roles

Navigate to the Roles configuration tab under Access Control:

1. Click on Administration, in the title bar
2. Select Access Control (under Access)
3. Select the Roles tab
4. Highlight the role for ContentAdmin

Can you find the details for this role, where you can see the permissions and assigned users? Don't worry if you can't, we will take a look in the next step.
Now let's look at the PowerUser role:

1. Highlight the PowerUser role in the list.
2. Review the Details pane on the bottom of the screen. Verify that it is showing details for PowerUser.
3. Look at the Permissions assigned to this role.

Notice how this user has some, but not all, permissions assigned. Click on the chevrons to the left of the permission grouping to see the individually assigned permissions. You will also browse the permissions in the next step, if the current window is too small.

You can assign individual permissions, or groups of permissions. Let's take a look at the permissions that are available:

1. Click on the pencil icon
View/Assign Permissions

1. Click on **Expand All**
2. Scroll through the list of permissions available

*Take some time to look at the permission groupings and the permissions available in each group. Access control is extremely granular in vRealize Operations Manager.*

3. Cancel out of this screen once you have reviewed the groupings.

Assigning Permissions to Objects

We have looked at how roles define **what** a user or group of users can do within vRealize Operations Manager. Now we are going to see how we can define **where** they can do it, i.e. which objects they have access to.

This is configured when the role is assigned to the user or user group. We will see it in action in the next step when we create our own role.
Lab Work

In the last lesson, User Access and Privileges, we created the user *intern1*, the group *interns*, and added the user to the group. Now we are going to assign him to a role where he can only view dashboards within vRealize Operations.

Create New Role

First we create a role and assign permissions to it. Make sure you are still on the Roles tab in Access control, then:

1. Click on the plus (+) sign to add a role

Configure Role

1. Name: *Intern Role*, Description: *Role with access to dashboards*
2. Click OK

An easy way to create roles is to select an existing role that is close to what you need, and select clone.
Edit New Role

Your new role should be visible in the list of roles now. To see the Users, Groups and Permissions assigned to a role, highlight the role. The information in the bottom pane "Details for Role" will change to the context of the the highlighted role. Let's assign permissions to your new role.

1. Highlight the **Intern Role** from the list of configured roles
2. Verify that the **Details** pane changes to reflect the details for that role
3. Click on the **pencil** icon by permissions. There should be no permissions assigned to this role initially.
Login Interactively

First, we must configure this role to be able to log into vRealize Operations Manager interactively. This permission is under **Administration > Support > Login Interactively**.

1. Expand **Administration**
2. Check the box for **Login Interactively**
3. Collapse All. Do you see the line by "Administration" to indicate one or more sub-permissions are checked?

Stay in this pane for the next step.
Now, we must configure this role to be able to see dashboards. This permissions is under **Dashboard > Dashboard Management > View Dashboards List**.

1. Expand **Dashboards**
2. Expand **Dashboard Management**
3. Check the box for **View Dashboards List**
4. Collapse All

Stay in this pane for the next step.
Next, we must allow this role to render views, or the user will not be able to see widgets. This permission is under **Dashboards > Views Management > Render**.

1. Expand **Dashboards**
2. Expand **Views Management**
3. Check the box for **Render**
4. Check the box for **View Dashboards Page**
5. Collapse All

Stay in this pane for the next step.
Finally, we must allow this role to have access to the environment, in this case the vCenter Adapter Inventory Tree. This permission is under Environment > Inventory Trees > vCenter Adapter.

1. Expand Environment
2. Expand Inventory Trees
3. Check the box for vCenter Adapter
4. Click Update.
Review Updates to Role Permissions

1. Verify that there are now permissions assigned to the role.

Overview of Updates

It may seem like there were a lot of steps here, but in reality we assigned only 5 permissions:

Login Interactively:

- Administration > Login Interactively

View Dashboards:

- Dashboards > Dashboard Management > View Dashboards List
- Dashboards > Views Management > Render
- Dashboards > View Dashboards Page

Search inventory tree:

- Environment > Inventory trees > vCenter Adapter
Assign Role to Group

We are now going to assign the new role to the group that we created in lesson one:

1. Return to the User Groups tab
2. Highlight the interns group
3. Click on the pencil to edit
Group Information

Verify that you are editing the correct group. We are not going to make and changes to the user information, so we can click Next to move to Assign Members and Permissions:

1. Click **Next**
We are not going to make any changes to the Members of this group, but you can verify that intern1 is checked. Select the Objects tab to assign the new role:

1. Click **Objects**
Assign Role

Now it all comes together! On the Objects tab, we are going to assign the role (the what) and objects (the where) to the user:

1. Select the Intern Role from the dropdown
2. Check the box Assign this role to the group
3. Select Adapter Instance from the Object Hierarchy
4. Select vCenter Adapter from Objects
Assign Hosts and Clusters

Still the Objects tab, scroll down to assign access to **vSphere Hosts and Clusters**:

1. Check the box **vSphere Hosts and Clusters**
2. Click **Finish**
Verify Permissions

While still on the User Groups tab, select the interns user and verify that the Intern Role was applied and that the group now has permissions to objects:

1. Verify you are on the **User Groups** tab
2. Highlight the **interns** group
3. Select the **Permissions** tab from the **Details** pane
4. Verify that the **interns** group now has Permissions associated, inherited from the **Intern Role**.
Log out as hol@corp.local

Log out as the hol user so we can test the inter1 user and Role:

1. Select the person icon.
2. Select the Log Out option.

Log in as intern1

Log in as the new local user intern1:
1. Select **Local Users** from the authentication source drop down
2. Enter the credentials for intern1:
   - User Name: **intern1**
   - Password: **VMware1!**
3. Click on **LOG IN**

Is the view for this user different? What is missing?

### Dashboard only access

If we configured the role correctly, the intern1 user should only see dashboards - the Getting Started dashboard, Dashboards in the title bar, and the dashboards menu.

This may be exactly how we want to set up access for some operational users. However, for our intern we don't want to just restrict access to dashboards, we want to restrict it to one particular dashboard. In the next lesson we will make a couple of additional changes to see how to accomplish that.
Log out intern1

Log out as the intern1 user:

1. Select the **person** icon.
2. Select the **Log Out** option.

**Conclusion**

In this lesson we discovered how granular we can get with the vRealize Operations Manager controls, using roles to determine which content and objects a user has access to.

In the next lesson we will look at dashboard sharing, and how it can be used to target specific content at user groups.
Manage Dashboards

You can share a dashboard or dashboard template with one or more user groups. When you share a dashboard, it becomes available to all of the users in the group that you select. This is very useful for creating custom content for specific groups.

Most out-of-the-box dashboards are shared to the Everyone group. To share custom dashboards to all users, they can be shared to the Everyone group. To restrict access to dashboards, they should be removed from the Everyone group.

*Note: Some content packs will share their dashboards to the Everyone group on install. If you do decide to limit content to certain users, you will need to keep track of what you expect to be shared and to whom, and verify those settings after adding additional content or upgrading.*

Lab Work

We are now going to log in as the local admin to edit dashboard sharing. A vanilla install may have all dashboards shared to the Everyone group. We want to get more granular than that, so we need to make some changes.

Log in admin user

![Login screen](image-url)
1. Verify the **Local Users** authentication source is still selected
2. Enter the credentials for intern1:
   - User Name: **admin**
   - Password: **VMware1!**
3. Click on **LOG IN**

### Manage Dashboards

Select the **Dashboards** tab and then select the **Actions** dropdown, and the **Manage Dashboards** menu item:

1. Select **Dashboards**.
2. Click the **Actions** down chevron to open the options.
3. Select **Manage Dashboards**.
From the **Manage Dashboards** content pane, click on the **Actions** icon and select **Share Dashboards**:

1. Click on the **Actions** icon
2. Select **Share Dashboards**
By default, dashboards are shared to everyone, and users cannot be removed from the everyone group. In order to limit the dashboards that certain users see, you will need to give some thought to how you plan to organize dashboards. Here, we are simply going to unshare all dashboards to the Everyone group, and share one dashboard to the interns group:

1. Select the **Everyone** group.
2. Check that the **Share Dashboards Tabs** view is showing.
   - There are 2 views - **Dashboard Tabs** and **Dashboard Templates**
   - You switch between them by clicking on the small icon to the right of the text in (2)
3. Highlight all of the shared dashboards
4. Click the **Unshare** icon.
Verify Everyone Group

Verify that Dashboards Assigned number for the Everyone group is 0. Notice the small red triangle in the corner by the 0. This means there is an unsaved change. Now we need to assign a dashboard to the interns group:

1. **Verify** Everyone has 0 Assigned Dashboards.
2. Highlight the interns group, and verify that Dashboards Assigned is also 0.
3. Make sure you are still toggled to the Share Dashboard Tabs view.
4. Click on the small Deselect All icon to show the full list of dashboards.
1. Switch to page 2 of the listed dashboards, and scroll down about halfway until you see the **Optimize Performance** dashboard.
2. Highlight the **Optimize Performance** dashboard.
3. Drag the **highlighted dashboard** into the **interns** group, and check that the **Dashboards Assigned** number is now **1**.
4. Click **Save**.

That's it! You have now configured our intern to only have access to one dashboard. Let's verify!
Log out as admin user

Log out as the admin user so we can test shared dashboards:

1. Select the person icon.
2. Select the Log Out option.

Log in as intern1

Log back in as intern1:
1. Select **Local Users** from the authentication source drop down.
2. Enter the credentials for intern1:
   - User Name: **intern1**
   - Password: **VMware1!**
3. Click on **LOG IN**.

**Validate intern1 user**

![Dashboard Image]

Hmmm, we do not see only the dashboard that we expected to see. This is because the **SDDC Health Dashboards** cannot be unshared by simply removing from the Everyone group. Depending on how VMware or the third party configured the management pack, there may be additional steps needed to unshare content. In this case, there is a KB that shows how to prevent the display of default dashboards in vRealize Operations 6.x:

[https://kb.vmware.com/s/article/2133879](https://kb.vmware.com/s/article/2133879)

For this lab, we will simply uncheck the SDDC Health Dashboards in the next step to hide them from view.

1. Select **All Dashboards**
Select Dashboards

Unselect the SDDC Health Dashboards and select the Optimize dashboards to get the expected view:

1. Select the checkbox next to **Optimize**
2. Unselect the checkbox next to **SDDC Health Dashboards**

*Note that this is a workaround for this lab that simply hides the SDDC Health Dashboards from the navigation pane. To remove access all together you will need to follow the steps outlined in the KB article in the previous step.*

What we expected to see

Congratulations! If we followed the configuration steps correctly, the intern user should now only be able to see the Optimize Performance dashboard, for a list of undersized and oversized virtual machines. We are in the live instance of vRealize Operations.
Manager so there won't be a lot to see here, but hopefully it is enough to show you the power of user controls in vRealize Operations Manager.

Note: In this lab you are using a live instance of vRealize Operations. Depending on how long the lab has been running, there may or may not be any Oversized or Undersized VMs listed.

Log out intern1

Log out as the intern1 user:

1. Select the person icon.
2. Select the Log Out option.

Conclusion

In this lesson we touched on dashboard sharing - how to share or unshare dashboards to a user group. Most customers use the sharing feature to share new, custom created content with groups of users. We are also starting to see more administrators using the feature to restrict content to users. We looked at the caveat with some custom created content, and how there may be additional steps needed to remove access to that content.

In the next lesson, we will look at auditing users and the environment.
Auditing Users and the Environment

At times you might need to provide documentation as evidence of the sequence of activities that took place in your vRealize Operations Manager environment. Auditing allows you to view the users, objects, and information that is collected. To meet audit requirements, such as for business critical applications that contain sensitive data that must be protected, you can generate reports on the activities of your users, the privileges assigned to users to access objects, and the counts of objects and applications in your environment.

Auditing reports provide traceability of the objects and users in your environment.

Log in as hol user

First we need to make sure we are logged back in as the hol user. If you haven't already, log out the intern1 user and log back in as the hol user:

1. Select **VMware Identity Manager** from the authentication source drop down.
2. Click on **Redirect**.

First we need to make sure we are logged back in as the hol user. If you haven't already, log out the intern1 user and log back in as the hol user:
Login hol

If you are still in the same browser, your SSO session will still be valid and you may go straight back in as the hol user. Otherwise, log back in:

1. Enter the credentials for intern1:
   - User Name: hol
   - Password: VMware1!

2. Click on Sign In.

Audit Reports

There are 4 preconfigured audit reports in vRealize Operations Manager to provide documentation to support traceability of objects and users in your environment:

1. User Activity Report
2. Users Permissions Audit
3. System Audit
4. System Component Audit
They can all be found under Administration > History > Audit. We are going to look at each of these reports in this lesson.

User Activity Audit

The User Activity Audit report shows user related logging activity such as login, actions run, changes made and log out.

1. Select the Administration tab in the title bar.
2. Select History > Audit from the navigation pane.
3. Select the User Activity Audit tab.
4. We will look at the report options here in the options bar next step.

Report Options

The options available from the options bar (from left to right) are:

- Download - download report to PDF or XLS
• Configure - Output log to external syslog server
• Data range - Configure the start and end dates for the report
• Starting Line - Truncate the report to start at the specified line
• Number of Lines - Truncate the report to the specified number of lines

Filter

You can filter the log entries by various fields, including User Name, User ID and Auth Source. The filter is in the top right of the window.

User Permissions Audit

The User Permissions Audit report shows permissions assigned to a user.

1. Select the Administration tab in the title bar
2. Select **History > Audit** from the navigation pane

3. Select the **User Permissions Audit** tab

The only option for this report is to download it, to PDF or XLS format.

The report will show the following information about a user:

- Username
- Role
- Access Group(s)
- Access Right(s)

Scroll down to see the bottom of the report to see the user the we created in the last lesson, **intern1**. Do the permissions look correct?

**System Audit**

![Image of vRealize Operations Manager interface with选中System Audit tab]

The **System Audit** report shows object types, metrics, super metrics, applications and custom groups in your environment, including counts of each. This report can help you to understand the scale of your environment.

1. Select the **Administration** tab in the title bar
2. Select **History > Audit** from the navigation pane
3. Select the **System Audit** tab

The only option for this report is to download it, to PDF or XLS format.
System Component Audit

The **System Component Audit** report shows every component installed in the system, including version information.

1. Select the **Administration** tab in the title bar
2. Select **History > Audit** from the navigation pane
3. Select the **User Permissions Audit** tab

The only option for this report is to download it, again to PDF or XLS format.

### Conclusion

Is this lesson we looked at the different audit reports included with vRealize Operations Manager, and looked at them in the context of our new user. These reports provide an easy way to provide documentation of activities that have taken place in your environment and user permission levels. Auditing reports provide traceability of the objects and users in your environment.

This is the last lesson in this module.
Conclusion

In this module we walked through the Users and Roles in vRealize Operations. Access Control is an important part of a robust operational environment, and you should now be comfortable using users, groups, roles and object permissions to make sure users only have access to the content and objects needed.

We then looked at Dashboard Management - how to control which users see which dashboards.

We also reviewed how to audit these controls and provide documented evidence of user actions and configured permissions.

You've finished module 8

Congratulations on completing the module.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,
1. Click on the **END** button at the top of the page.
Module 9 - PowerCLI for vRealize Operations (60 minutes)
Introduction

VMware PowerCLI contains modules of cmdlets based on Microsoft PowerShell for automating vSphere, VMware Site Recovery Manager, vSphere Automation SDK, vRealize Automation, 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.

The module contains the following lessons:

- Lesson 1: Basic Windows PowerShell usage and VMware PowerCLI for vRealize Operations cmdlets
- Lesson 2: Connecting to vRealize Operations with VMware PowerCLI
- Lesson 3: Using VMware PowerCLI to work with vRealize Operations Alerts
- Lesson 4: Using VMware PowerCLI to work with vRealize Operations Recommendations
- Lesson 5: Using VMware PowerCLI to work with vRealize Operations Statistics
- Lesson 6: Conclusion

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 file, 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.

- **PowerShell ISE** - The PowerShell Integrated Scripting Environment (ISE) is a host application for Windows PowerShell. The ISE provides a Windows-based GUI for PowerShell script development. Primary ISE features include multiline editing, selective execution, and storing functions/variables.
Launching VMware Modules

Before executing commands in a script or interacting at the PowerShell command prompt, we must load the VMware modules. The new preferred method to import these modules is to simply execute a 'Connect-' command. By connecting to a VMware Infrastructure (VI) server, PowerShell now understands to automatically import the relevant PowerCLI modules for that VI provider. Please see the following video that details this process of configuring PowerShell for module download from your preferred PowerShell repo.

https://youtu.be/r2vi78DWhLQ

Optional areas to execute commands

When executing PowerCLI commands, the user has a number of options of where they might want to execute the command(s) to affect the environment. The user may opt to execute a single command in an ad-hoc PowerShell window, use the ISE to execute commands so that variables may be stored, or they may choose to execute commands in an automated approach using scripts. Before scripts can be automated, its important to first test and validate all commands in your environment to ensure proper compatibility and functionality. In this module we will help you through various steps of testing PowerCLI commands in both the native PowerShell window and with the ISE. We will only use the ISE when we expect to store a value in a variable that we plan to use in a subsequent command.
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:

   ```powershell
   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.

Import the VMware Modules

To import the vROps PowerCLI module:

1. Type ```Import-Module -Name "C:\Program Files\WindowsPowerShell\Modules\VMware.VimAutomation.vROps"``` and hit enter.

Now that the module has imported, the VMware PowerShell commands can now be executed at the PowerShell command prompt.
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. Type `Get-Help Get-VM` and hit enter.

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.
Checking the version information for the vRealize Operations module

1. Type `Get-Module -Name VMware.VimAutomation.vROps` and hit enter.
2. Check the module version information in the output.

Checking the available cmdlets in the vRealize Operations module

2. Check all the available cmdlets within the module.

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.
Listing the available cmdlets

Let's explore what we can do with VMware PowerCLI. PowerCLI's snap-ins provide more than 500 cmdlets for managing vSphere, SRM, vRA, and vROps. You can view the available PowerCLI cmdlets by typing "Get-VICommand". 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. Type `Get-VICommand *VM` and hit enter.
2. As a result, you will see all the cmdlets containing the word "VM"

Please note, 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.
Executing VMware PowerCLI cmdlets in Windows PowerShell command prompt or the ISE

During this lab you will be executing VMware PowerCLI cmdlets using both the basic PowerShell command prompt and also the PowerShell ISE Script Pane. The advantage of using the ISE Script Pane is the ability to store variables and run multiple commands, this is an important capability when testing and validating a script for automation purposes. When using the ISE, 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.
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, also known as vROps "HVM" mode.

Connect to vRealize Operations server

1. Using the PowerShell command prompt, type `Connect-OMServer -Server vrops-hvm -User admin -Password VMware1!` and hit enter.
2. Check for any errors 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. Type `Get-OMAlert -Status Active -Criticality Critical -Impact Health` and hit enter.
2. Check the list of active critical alerts.
Listing existing alert types and subtypes

If you take a detailed look on the previous command result you will see that there are columns for **Type** and **Subtype** which can be used as input parameters for the **Get-OMAlert** cmdlet (when developing a script), 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. Type `Get-OMAlertType` and hit enter.
2. From the list of alert types in the output 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. Type `Get-OMAlertSubType -AlertType 'Virtualization/Hypervisor Alerts'` and hit enter.
2. From the list of alert subtypes in the output 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. Type `Get-OMAlertDefinition -Type 'Network Alerts' -SubType Configuration` and hit enter.
2. From the list of alert definitions in the output you can see several alert definitions for the `Configuration` subtype of `Network Alerts` type.

Showing detailed information about an alert definition
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. Type `Get-OMAlertDefinition -Name 'There is an MTU*' | Format-List` and hit enter.
2. In the output you can see all details about the specific alert definition related to MTU configuration issues.

Please Note: The values for AdapterKind and ResourceKind properties 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. Type `Get-OMAlertDefinition -ResourceKind VmwareDistributedVirtualSwitch` and hit enter.
2. In the output 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.

For the remaining steps in this lesson we would like to demonstrate the ability to store variables similarly as done while developing a script. To do this you will need to find the previously opened PowerShell ISE window and bring it into focus. You can minimize the PowerShell command prompt we have been using so far.

**Starting Windows PowerShell ISE**

Now that we've explored some of the base functionality of the PowerShell command prompt, let's launch the PowerShell ISE. The Windows PowerShell ISE will provide an alternative area to run commands for the purpose of script editing.
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.

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.
Connect to vROps "HVM" mode instance

To connect to vROps HVM instance:

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. From the output in the **Command Pane**, we can see that we are now connected.
Searching for a specific VM to use as parameter for the alert variable

Once the ISE window is in focus, let's do a search for all resources that have 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; this command may take a little longer to execute as it retrieves a large amount of information.
3. From the listed alerts in the Command Pane, let's choose a specific resource to use as a parameter for our script. We can use the scroll bar to view all the results.
4. In this example let's plan to use the "t-win2k12" object which is a VM for our next step. You may need to scroll down about 2/3 of the output to find this server.
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**

Using the **Set-OMAlert** cmdlet we 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.
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 = Set-OMAlert $alert -TakeOwnership -Confirm:$false
   $alert = Set-OMAlert $alert -SuspendMinutes 60
   $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**.

In the **Command Pane**, check the information about ownership and suspension of the alert. You will need to scroll up to look at the initial command output.

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.

Please Note: 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.
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 the 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 (HVM) 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.

Connect to vRealize Operations server

1. In the Script Pane type "Connect-OMServer -Server vrops-01a -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.

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 the remaining steps in 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); In the next line of the Script Pane type "Get-OMStatKey -Name cpu* -Resource $vmresource" 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 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 $vmresource" in the first line and hit **ENTER**; In the next line of the **Script Pane** type "$statkey | Format-List" 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 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 Operatios integration can be a huge help.
Log in to the vRealize Operations Live Instance

In this lesson we will log into vRealize Operations Live Instance in preparation for the next lesson. Once you log in successfully following the steps below, please minimize the browser.

Open the Chrome Browser from Windows Quick Launch Task Bar

If your browser isn't already open, launch Google Chrome

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

Open the vRealize Operations - Live Instance Tab

The browser home page has links to the different instances of vRealize Operations that are running in the lab.

1. Click the vRealize Operations - Live Instance link to open the UI in a new browser tab
Log in to vRealize Operations

vRealize Operations is integrated with VMware Identity Manager which we will use for user authentication in this lab.

VMware Identity Manager should be pre-selected as the identity source. However, if it is not you will choose it.

Click the drop-down arrow

1. Select **VMware Identity Manager**
2. Click **REDIRECT** to take you to the user login page
VMware Identity Manager Login

The user and password information should already be filled out. However, if needed type them in.

USER: hol

PASSWORD: VMware1!

1. Click Sign in
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
- How to apply recommendations for a specific VM
- List the current alerts of the VM after applying the recommendation

Connect with Putty

To assist with the process of identifying and interacting with Operations Manager alerts via Powershell, we must first generate an alert that we can interact with. For this purpose we have staged a CPU load script on the server 'web-01a'. We will connect to this server using Putty and issue the commands to start the load script.
1. Launch Putty

We will now select the saved session information to connect to 'web-01a'.

1. Scroll down to the bottom
2. Select 'web-01a.corp.local'
3. Select 'Open'
Generating a CPU Load Alert

We have written a script and stored it on the root partition of the CentOS 7 server web-01a. This script is ready to be executed and will generate load until the user presses enter to end the script.

Now that we are connected, let's execute the script cpu_load.sh on web-01a to generate an alert.

1. Enter the command `./cpu_load.sh` to execute the script.

Listing specific alerts for a chosen VM

1. Enter the command `Get-OnAlert -Status Active -Criticality Critical` to list specific alerts for a chosen VM.
After generating load on a server, let's navigate back to the ISE and execute commands to search for all resource objects that have **Active** alerts impacting on **Health** and with **Critical** criticality. We are expecting to find our web server with high CPU load.

1. In the **Script Pane** type `Get-OMAlert -Status Active -Criticality Critical`
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**, notice our server "**web-01a**" listed as a resource in an active alert.

If for some reason you got to this part very fast and enough time has not elapsed for the data collection, you may need to wait a few moments and re-execute the command before the 'VM CPU High Web Utilization' alert is triggered. You are doing so well!

**Listing recommendations for a specific alert**

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

1. In the **Script Pane** type `Get-OMRecommendation -Alert 'VM CPU High Web Utilization' | 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 more CPU capacity.
Applying recommendations for a specific VM

Navigate to vROps Alerts

We will now go to the vRealize Operation Manager console to view the alert we have generated in the previous attempt. Please open the browser window previously launched in a previous lesson, it should already be logged into vROps Live Mode. If not, please relaunch the browser and connect to the vROps Live Mode instance. Then perform the following steps:

1. In the upper right hand corner, please select the magnifying glass and type in web-01a.
2. When the search results come back, select the web-01a icon with the mouse

View Alerts Details and Recommendation(s)

From the details of web-01a, lets navigate to the Alerts area, expand the alerts, inspect our specific CPU alert
1. Select Alerts.
2. Expand the list of Alerts.
3. Click on the Alert to get details.

**Lets apply the recommendation to add more CPU capacity to web-01a to alleviate the high CPU alert**

From the details of web-01a, lets navigate to the Alerts area, note the available recommendation, Run the recommendation.

1. Notice the available Recommendation(s).
2. Select **Run Action**.
Confirm CPU Changes

Let's increment the amount of CPUs by 1.

1. Enter the number 2 for the count for "New CPU"
2. Press the button for **BEGIN ACTION**

**Confirmation of task generation**

Let's review the confirmation. We have options to monitor the task(s) related to the action item.
1. Select the **Recent Task** link
2. Also close the dialogue box

### Reviewing Recent Tasks

On the Recent Tasks view we can see our task execution and its progress. You may refresh this screen as needed to see updates on Status and Completed Time.

**List the current alerts of the VM after applying the recommendation**

Now that we applied the recommendation for the "web-01a" VM by increasing its CPU, let's check if that particular alert is gone by issuing the related powershell commands. Please note that it may take up to 1 min for the alert to clear.
Check for alerts

Let's search again for all resource objects that have Active alerts impacting on Health and with Critical criticality. We are expecting to NOT find our web server with high CPU load.

1. In the **Script Pane** type `Get-OMAlert -Status Active -Criticality Critical`
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**, notice our server "**web-01a" is no longer listed as a resource in an active alert.

This completes this lesson and module. We hope that you were able to learn some new skills around script writing and automation of vROps alerts, definitions, and recommendations.
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

You've finished module 9

Congratulations on completing the module.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

You may proceed to the next module by advancing to the next page. If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)**

Or if you want to end your lab,
1. Click on the **END** button at the top of the page.
Module 10 - Remediate Issues Using vRealize Log Insight and vRealize Orchestrator Workflows (45 minutes)
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 database service and then using the resulting alert in vRealize Log Insight to launch a workflow in vRealize Orchestrator via the webhook shim to restart the service.
Log in to vRealize Log Insight

This module uses vRealize Log Insight

Open the Chrome Browser from Windows Quick Launch Task Bar

Now let's start the lab module.

1. If Chrome is not currently open, click the "Chrome" icon on the Windows Quick Launch Task Bar.

Open a vRealize Log Insight Tab

1. The default page will be the HOL-1901 Lab Links page. Click the "vRealize Log Insight" link.
Login to vRealize Log Insight

vRealize Log Insight is integrated with VMware Identity Manager which we will use for this lab.

One of the major reasons for the change to vIDM is not only the reduced footprint and ability to easily scale, it’s also the added functionality. vIDM now brings a slew of new features, including:

- OTB 3rd party SAML Token Support
- OTB Smart Card Support
- Multi-factor authentication
- Login Auditing
- Major Scalability Improvements
- HA support (configured by wizard)

In addition, the vIDM identity stack supports many authentication protocols and methods (some of which were previously not supported):

- Username/password
- Kerberos
- SAML Authentication
- Smart Card / Certificate
- RSA SecurID
- RADIUS
- RSA Adaptive Authentication

1. VMware Identity Manager should be pre-selected however if needed click the drop down as shown and select "VMware Identity Manager"
2. Click "LOGIN VIA SSO" to take you to the user login page.
VMware Identity Manager Login

The user and password information should already be pre-selected, however if needed the user and password are:

USER: hol

PASSWORD: VMware1!

1. Click "Sign in"
vRealize Log Insight Dashboard Page

When you first log in to vRealize Log Insight you are presented with the General Overview dashboard.

Set Chrome Browser Zoom Level

The lab environment has a default resolution of 1280 x 800 with a Zoom setting of 80%. This minimizes the need for extensive scrolling within the vRealize Operations user interface. If needed, you can adjust the zoom level in Chrome by:

1. Open the Chrome drop down menu as shown.
2. Set the desired zoom level. Typically 80-90% is sufficient to provide adequate screen space for vRealize Operations in the lab environment. Also making use of the full-screen option is recommended.
Self Healing DataCenter with vRealize Log Insight

Both vRealize Log Insight and vRealize Operations support webhooks. This module will walk you through configuring webhooks for vRealize Log Insight and fixing a database service which has stopped running. Webhooks allows integration with other solutions that has a REST API available. Ticketing, notification, chat and other capabilities can be leveraged by vRealize Log Insight. Extending that capability can make your datacenter a self-healing datacenter!

Wouldn’t it be great to have a vRealize Log Insight alert on particular logs and trigger a workflow to fix the issue automatically? This module will show that capability.

REST API

This means is that unless your REST API is expecting the request in the way vR Ops would like to send it – it will fail.

For example, Orchestrator has a robust REST API. To launch a workflow via that API, the inputs have to match the image above.
Enter Webhook Shims!

One way is to translate the REST call from vRealize Log Insight into a REST call that can be consumed by Orchestrator. Enter the [Webhook Shims for vRealize Log Insight](#)! This diagram shows how it works:

Webhook Shims enables self-healing with vRealize Log Insight through an Orchestrator shim. The Orchestrator shim takes the simplistic input from vRealize Log Insight REST notification and rebuilds the request in the format expected by Orchestrator.
3-Tier Application

This simple application provides an environment that uses 3 machines which communicate in order to provide access to data via a web page. It is nothing terribly fancy, but allows for demonstration of various communication and security options between distributed components of an application within a vPod.

The base package consists of 3 base CentOS 6.6, 64-bit machines. The initial IP addresses are indicated here.

**Web App Components - Default Configuration**

- **web-01a** - 192.168.120.30
  - This machine is a reverse proxy, listening on port 80
  - All traffic is proxied to http://app-01a.corp.local:8080/
  - The self-signed SSL certificate uses the name webapp.corp.local

- **app-01a** - 192.168.120.20
  - This machine runs a web server, listening on port 80
The “application” configured here makes a connection to db-01a on port 3306 (MySQL)

- **db-01a** - 192.168.120.10

- This machine is a database server, listening on port 3306
- It runs MariaDB (a non-Oracle fork of MySQL), with a single database, vcndsb

### Understanding The Lab Environment

The next few pages will help you understand the components of this lab.

### Photon-OS-Container-Host

The **Photon-OS-Container-Host** is the virtual machine which contains a python script (the webhook shim) which will be listening for REST notifications sent from vRealize Operations Alert then reformatting the information and calling the vRealize Orchestrator API. vRealize Orchestrator will then run a workflow called “Restart Service from Alert” to restart the service.
db-01a Virtual Machine

The summary page is shown for the db-01a server. This shows you that there is one OS process being monitoring using the Endpoint Operations. In this example we are monitoring the MySQL process running on this database server.

Another PuTTY session

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 db-01a VM

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

1. Click the **db-01a** entry shown in the list of saved sessions.
2. Click on **Load**
3. Click **Open** to start the putty session for db-01a.
Verify the MariaDB Process is Running

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

1. Type `./check_running_service` and click Enter. Verify that the database 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@db-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-01a.corp.local**
2. Click **photon-01a.corp.local**
3. Click **Load**
4. Click **Open** to start the putty session for the photon-01a.
Start Docker Container

In the console session we need to start the container where the webhook shim 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.

1. Enter `ls` to see the directory contents. Notice there are 2 scripts. "edit-docker-shim.sh" and "start_webhook.sh".
2. Enter `./start_webhook.sh` This command will start the docker container called "webhook". It only takes a couple seconds to start the container. The result is shown as a listener on port "5001".

```bash
 ls  # Directory contents

 1. ls

 2. ./start_webhook.sh

2018-06-30 22:53:45,342 INFO Please navigate to the below URL for the available routes
2018-06-30 22:53:45,350 INFO * Running on http://0.0.0.0:5001/ (Press CTRL+C to quit)
```
Verify That the Webhook Shims Are Running

To test the webhook shims and that the listener is working, complete the following steps.

1. Open a new tab in the browser.
2. Click the bookmark or enter `photon-os-container-host:5001` as the URL. The listener of the webhook container responds with documentation of all the shims available.
3. 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. You can close the webhook tab since you are done with it.
Webhook Response

1. The result from the previous step is shown above. This shows the listeners response.

Explore the Workflow in vRealize Orchestrator

1. Lets go take a look at the workflow that will be triggered. Double-click vRealize Orchestrator on the Desktop to launch the client.
vRealize Orchestrator Login

1. Enter **VMware1!** for the password.
2. Click **Login**
Viewing the Workflows

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

Restart Service from Alert Workflow

1. Click the v to expand the Webhook Shim Sample folder.
2. Click the Log Insight Start MariaDB workflow to view the details of the workflow.
3. The is ID of the workflow and will be used later for create the alert which triggers this workflow.

**Schema**

![Schema Diagram]

When the MariaDB service is stopped on db-01, a Alert is generated within Log Insight. The Alert is sent to the webhook shim which then sends it to vRO. There are several attributes in the workflow to restart the service.

1. Slide the **Scroll Bar** on the bottom of the General tab to see the attributes of the workflow.
2. Notice there are attributes for user/password, command to run, host among others.
Log Insight

1. Click the **Log Insight** tab in the browser.
2. Click **Interactive Analytics** to review the log messages.

Interactive Analytics

Currently Log Insight is displaying all the logs it has received for the last 5 minutes. Your environment may have a different number but still too many logs to review. Let's see what logs have been received from our database server db-01a.

1. Enter **db-01a** as shown.
2. Click the **Search** icon to filter the logs to just db-01a.

**db-01a Logs**

1. Now the logs are reduced significantly. In our example shown there are 214 events, however in our example of a service going down we want to see a log message for a specific query and create an alert from that query.

**Create Hostname Filter**

1. Click **5 minutes of data** on the drop down menu for timeframe.
2. **Clear the search field so it is blank.**
3. Click **+ ADD FILTER** to setup a filter.
Add Hostname Filter

1. Click text and a property dropdown menu will appear. Find hostname in the list either by scrolling the list or begin entering host and select hostname.
2. Change the other fields so that contains and db-01a is set as the filter.
3. Click + Add Filter found under the filter you just created.

Text Filter

1. Create a second filter which is text contains: Stopping MariaDB
2. Change the timeframe from 5 minutes to All time to verify you have the filters set correctly. You should get a list of old events as shown.
3. Now change the timeframe from All time back to 5 minutes of data to create the alert. The number of events will be zero until the service is stopped.
Create Alert from Query

1. Verify the **2 filters** and the timeframe is **5 minutes of data** is set in your lab as shown above.
2. Click the **Bell icon**
3. Click **Create Alert from Query**

Create a New Alert from Query

1. Name: **MariaDB Service is Down**
2. Recommendation: *Don’t worry about it! The service is being restarted automatically*
3. Notify: Email address(es) separated by commas
   - Email: [http://photon-01e.5001/endpoint/vro/d03293e2-d082-4530-8396-9c533de7f38d](http://photon-01e.5001/endpoint/vro/d03293e2-d082-4530-8396-9c533de7f38d)
4. Send to vRealize Operations Manager
5. **Raise an alert**
   - On occurrence of: **When an event is seen for the first time in the last 10 minutes**
1. Enter **MariaDB Service is Down** for the name of the Alert
2. Recommendation is optional but here is one possibility just for fun.

   Disclaimer: Don’t worry about it! The service is being restarted automatically!

3. Click the **email checkbox to clear it and check the webhook to enable it** as shown. Also enter the following for the Webhook URL:
5. Click the Radio button as shown, select **Custom** from the dropdown menu and enter **10** minutes.
6. Click **SAVE** to save the alert.

### Save Shutdown Query

1. Click the **Add current query to favorites icon** as shown. It will helpful to save our current query for later.
2. For the query **Name** enter: **Shutdown Query for MariaDB on db-01a**
3. Click **SAVE**

### Successful Restart Filter

1. [Image of Successful Restart Filter]
2. [Image of Successful Restart Filter]
1. Modify the second filter by entering **Starting MariaDB**
2. Change the timeframe to "**All time**" to retrieve old events and test the filters.

Change the timeframe back to "**5 minutes of data**" before moving to the next step.

### Save Success Query

1. Click the "Add current query to favorites icon" as shown.
2. For the query Name enter: **Successful Start of MariaDB**
3. Click "**SAVE**"

### Check Favorite Queries

1. Click the **Favorite Queries icon**. A dialog box will appear with the 2 saved queries in the list.
2. Click **Shutdown Query for MariaDB on db-01a**. Now to create a dashboard.
Adding Shutdown Query to Dashboard

1. Click Add current query to dashboard icon.
2. Type Shutdown Query for MariaDB on db-01a as the Name.
3. Click New Dashboard
4. Type My Services as the new dashboard name.
5. Click SAVE

Query List Widget Type

1. Click Chart and click Query List to set the widget type.
**Query List**

1. Click **New Query List** to create a new list.
2. Type **Save Queries**
3. Click **SAVE**
4. Click **ADD**

**Success Favorite**

1. Click the favorite icon.
2. Type **Successful start of MapReduce**
3. Click **ADD**
1. Click the **Favorites icon**
2. Click the **Successful Start of MariaDB** favorite.
3. Click the **Add current query to dashboard** icon.

**Query List Widget Type**

1. For the name type: **Successful Start of MariaDB**
2. Verify the other fields are filled out as shown. Modify them if needed.
3. Click "**ADD**".
**MariaDB Event Chart**

1. Click favorite query **Shutdown Query for MariaDB on db-01a** to change back to that filter. The fields begin filtered are highlighted above.
2. Click the **Add current query to dashboard** icon.
3. For the Name type: **MariaDB Shutdown Event**
4. Change Widget Type: **Chart**
5. Click **ADD** to add another widget to the dashboard.

**Dashboard**

1. Switch to the Dashboard screen by clicking **Dashboards**
2. You may need to expand the **My Dashboards** and select **My Services** to see the dashboard you built.
3. Let's add a couple more widgets to this dashboard. Click **Interactive Analytics**
Success Filter setting

1. Click the **Favorites icon**
2. Click the **Successful Start of MariaDB** favorite query.
3. Click the **Add current query to dashboard** icon.

VRO Workflow Field Widget

1. For the name of the widget type: **MariaDB Start Events** and verify the other fields are set as shown.
2. Click **ADD** to add to your dashboard.

**Completed Log Insight Dashboard**

1. Click **Dashboards** to switch to the dashboard view. There is no data listed right now because no event has occurred yet in the last 5 minutes.
2. Click the **Enter Presentation Mode (enables auto refresh mode)** so the dashboard is updated as events occur.
Arrange Screen

If you arrange your screen similar to what is shown you will see the events occur. You may want to resize the PuTTy windows as shown.

Script to Stop MariaDB
1. Type `./mariadb_stop.sh` to begin the shutdown script.

### Service Not Running

The db-01a putty session shows the MariaDB service is not active. The watch command will update this every 2 seconds and will show when the service has begun.

### Multi-Tier App Error

1. Click the **HOL-Multi-Tier** tab on the browser.
2. Click the **Browser Refresh** icon.

You can see the web page no longer returns the inventory of the database server since the database service is down.
Events

There are several items to observe when the database service goes down and then also when the service comes backup.

1. When the Log Insight Alert is triggered the webhook listener responds to the alert which is shown in the photon-01a session.
2. The db-01a session shows the service has been restarted after a few minutes.
3. Log Insight events are recorded in the widgets when the database service goes down and then when successfully restarted.
4. Hover over the area of the Saved Queries Widget. When the icon appears as shown click the "Run each query to check for results" icon.

Interactive Analytics Results

The Interactive Analytics returns the results of the query. This concludes this module.
Conclusion

This module you learned about Webhook Shims and took you through an example of a self-healing datacenter. This lesson the database service was stopped and then using vRealize Log Insight alert to automatically launch a vRealize Orchestrator workflow via the webhook shim to restart the service.

You've finished module 10

Congratulations on completing the module.

If you are looking for additional information on vRealize Operations, you can start here: https://www.vmware.com/products/vrealize-suite.html

If you want to jump to a particular module, follow one of the links below.

- **Module 1 - Creating and Modifying Dashboards in vRealize Operations (45 minutes)**
- **Module 2 - Creating and Modifying Views and Reports in vRealize Operations (60 minutes)**
- **Module 3 - Symptoms, Recommendations and Alerts in vRealize Operations (45 minutes)**
- **Module 4 - Super Metrics in vRealize Operations (45 minutes)**
- **Module 5 - Using the vRealize Operations API (30 minutes)**
- **Module 6 - Remediate Issues Using vRealize Operations and vRealize Orchestrator Workflows (45 minutes)**
- **Module 7 - vRealize Orchestrator Management Pack for vRealize Operations (30 Minutes)**
- **Module 8 - Users and Roles in vRealize Operations (30 minutes)**
- **Module 9 - PowerCLI for vRealize Operations (60 minutes)**

Or if you want to end your lab,

1. Click on the **END** button at the top of the page.
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-1901-05-CMP

Version: 20190914-210709