# Table of Contents

Lab Overview - HOL-1921-06-CMP - vRealize Orchestrator - Advanced ............................. 2  
Lab Guidance .................................................................................................................. 3  
Module 1 - Creating Advanced vRealize Orchestrator Workflows (45 min) ................. 9  
  Introduction .................................................................................................................. 10  
  Prompting User Input In Workflows .......................................................................... 12  
  Using Conditional Execution and Loops ................................................................. 27  
  Bring it together: Sample Advanced Workflow ............................................. 42  
  Asynchronous Workflows ....................................................................................... 80  
  Conclusion ................................................................................................................. 90  
Module 2 - Integrating with REST APIs (45 min) ....................................................... 92  
  Introduction .............................................................................................................. 93  
  Configuring the vRealize Orchestrator Plug-in for REST ................................... 95  
  Invoking REST operations via vRealize Orchestrator ....................................... 106  
  Conclusion ............................................................................................................... 114  
Module 3 - Launching Workflows from Outside vRealize Orchestrator (45 min) ....... 116  
  Introduction ............................................................................................................. 117  
  Start a simple workflow ......................................................................................... 123  
  Start a workflow with complex inputs ................................................................. 134  
  Conclusion ............................................................................................................... 148  
Module 4 - Automating vRealize Automation with vRealize Orchestrator (45 min) ....... 150  
  Introduction ............................................................................................................ 151  
  Explore out-of-the-box workflows ....................................................................... 153  
  Create an automated onboarding process ......................................................... 166  
  Conclusion ............................................................................................................... 193  
Module 5 - Running In-Guest Operations with vRealize Orchestrator (45 min) .......... 195  
  Introduction ............................................................................................................. 196  
  Prepare the environment ....................................................................................... 197  
  Use Guest Operations ......................................................................................... 202  
  Use Guest Script Manager ................................................................................... 219  
  Clean Up the Environment .................................................................................. 250  
  Conclusion ............................................................................................................... 252  
Module 6 - Error Handling and Troubleshooting (45 min) ........................................... 254  
  Introduction ............................................................................................................. 255  
  Error handling and why is it important? ........................................................... 257  
  Adding Error Handling to workflows ................................................................. 261  
  Basic Workflow Troubleshooting ....................................................................... 297  
  Logging in Workflows ......................................................................................... 311  
  Conclusion ............................................................................................................... 320
Lab Overview - HOL-1921-06-CMP - vRealize Orchestrator - Advanced
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.

Use vRealize Orchestrator to perform advanced tasks such as integrating with REST APIs, perform actions in guest operating systems using VMware Tools and automate vRealize Automation tasks.

Lab Module List:

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
- **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.
- **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced) Discover the various ways to add error or exception handling to ease in troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

Lab Captains:

- **Module 1-6 - Benoit Serratrice, Staff Cloud Solutions Architect**
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 can not be saved. All your work must be done during the lab session. But you can click the **EXTEND** to increase your time. If you are at a VMware event, you can extend your lab time twice, for up to 30 minutes. Each click gives you an additional 15 minutes. Outside of VMware events, you can extend your lab time up to 9 hours and 30 minutes. Each click gives you an additional hour.
Alternate Methods of Keyboard Data Entry

During this module, you will input text into the Main Console. Besides directly typing it in, there are two very helpful methods of entering data which make it easier to enter complex data.

Click and Drag Lab Manual Content Into Console Active Window

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

Accessing the Online International Keyboard

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

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

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

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

Click on the @ key

1. Click on the "@ key".

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

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

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

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

This cosmetic issue has no effect on your lab.

Look at the lower right portion of the screen
Please check to see that your lab is finished all the startup routines and is ready for you to start. If you see anything other than "Ready", please wait a few minutes. If after 5 minutes you lab has not changed to "Ready", please ask for assistance.
Introduction

vRealize Orchestrator can be used to automate complex processes that require various steps, branching, and even looping.

In this module, conditional execution, loops, and asynchronous presentation are covered using real-life examples in relation with vRealize Automation. This module begins with details on how to implement these concepts in a workflow, and it is followed by a real-life example on how to request a vRealize Automation catalog item, wait for the provisioning to finish, and retrieve the newly created machine.

This type of use case is particularly useful for automated blueprint testing, or for provisioning load testing.

Note: Use drag and drop to enter code in scriptable tasks to avoid any mistakes.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: `vra-01a.corp.local:443`
2. User name: `administrator@vsphere.local`
3. Password: `VMware1!`
4. Click `Login`
Prompting User Input In Workflows

vRealize Orchestrator workflows allow you to validate input parameters. This way, the values have to match certain criteria before the workflow starts. In this lesson, basic validation types are covered including mandatory, min/max number, and dropdown.

Create a New Workflow

1. Make sure Design view is selected
2. Select Workflow tab
3. Right Click on HOL > 1921-06-CMP > Module 1 workflow folder
4. Click on New workflow
Name Workflow

1. Enter the name of the workflow: **Presentation Tutorial**
2. Click **OK**

Add Input Parameter: name

1. Select the **Inputs** tab
2. Click the **Add parameter** icon to add the first parameter to the list
3. Click the default parameter name **arg_in_0** to rename it
4. In the Choose attribute name... window, enter **name** for Attribute name
5. Click **Ok**
Add Input Parameter: list

1. Click the Add parameter icon to add the second parameter to the list
2. Click the default parameter name arg_in_0 to rename it
3. In the Choose attribute name... window, enter list for Attribute name
4. Click Ok
Add Input Parameter: num

1. Click the Add parameter icon to add the third parameter to the list
2. Click the default parameter name arg_in_0 to rename it
3. In the Choose attribute name... window, enter num for Attribute name
4. Click Ok
1. Click the **string** listing for the third parameter **num**
2. In the Select a type... window, select **number**
3. Click **Accept** to change the Type entry for this parameter to **number**
Add Property to name Parameter

1. Click on the Presentation tab
2. Select the (string) name name entry
3. In the pane on the right, click on the Add property... icon to add a property to this parameter

Make name Parameter Mandatory

1. In the Properties... window, select Mandatory input
2. Click Ok
Add List of Values to list Parameter

1. Select (string) list list entry on the left
2. Click the Add property... icon on the right (not shown)
3. In the Properties... window, select Predefined answers
4. Click Ok
Add Default Values

1. Click on the default value of Not set to open the Array of string window
2. Click the upper left corner of the Array of string window, and drag it down and to the right to resize it. Click the title bar of the window to move it up until the Cancel and Accept buttons are visible
3. Enter Small into the New value field
4. Click Insert value.

Repeat step 3 and for 4 with the following values: Medium and Large

5. Click Accept
Add Minimum Value for num Parameter

1. Click the **(number) num num** entry on the left
2. Click the **Add property...** icon on the right (not shown)
3. In the Properties... window, select **Minimum number value**
4. Click **Ok**
Add Maximum Value for num Parameter

1. Click the (number) num num entry on the left
2. Click the Add property... icon on the right (not shown)
3. In the Properties... window, select Maximum number value
4. Click Ok

These properties have default values of 0.0 for Minimum number value, and 100.0 for Maximum number value. These default values will suffice for this exercise.
Save and Close the Workflow

1. Click **Save and close** to close the workflow

Continue Despite Validation Errors

Unlike the previous modules, attempting to save and close this workflow will result in a Validation warning. This is because the 3 parameters defined in the previous steps are not actually used by this workflow. Click the **View details** button if you wish to see the specific information.
1. Since we know that we can proceed despite this warning, click **Continue anyway**

**Run the Workflow**

1. Click the **Start workflow...** icon
Observe Behavior for Mandatory Parameter

The name parameter has been designated as mandatory, as noted by the red * next to the parameter name. Attempting to skip the name parameter will generate an error.

1. Click the **name** text box, but do not enter any value
2. Click another field (such as the **num** text box) and observe the error message at the top of the parameter list
   - Notice that the Submit button is greyed out.
3. Enter a value in the name field, and the error message will be dismissed.
Observe Behavior for List of Values

The list parameter has a predefined set of values. For this reason, the list parameter on this screen is a dropdown menu of the values defined previously, and only these values may be chosen.

1. Choose a value for the list parameter from the dropdown menu
Observe Behavior for Numerical Range

The num parameter has both a minimum and a maximum defined value, with the minimum of 0.0 and the maximum of 100.0.

1. Enter a **number greater than 100.0** and observe the error message at the top of the parameter list.
2. Click on **Cancel** button

**To go Further**

Many more presentation properties are available, not limited to but including hide/show, dropdown list from Actions, and choices related to other fields. Feel free to take more time to explore all of the possibilities.
Using Conditional Execution and Loops

Conditional execution and loops are key constructs in vRealize Orchestrator. In a typical programming language, they are usually identified by a keyword such as for, while, if, then, else, and so on. In this lesson, the same constructs are used, but by dragging and dropping available workflow elements onto the schema of a workflow.

Create a New Workflow

1. Make sure Design view is selected
2. Select Workflow tab
3. Right Click on HOL > 1921-06-CMP > Module 1 workflow folder
4. Click on New workflow

Name New Workflow
1. Enter a name for the workflow: **My First Loop**
2. Click **Ok**

### Add Decision box

1. Select **Schema** tab
2. Select the **Generic** section from the list on the left
3. Click the **Decision** object, and drag it onto the arrow between the Start and End objects, noting the new branch added by the Decision object

A decision box is the equivalent of an IF, ELSE conditional branch. The green arrow is the path if the condition is true, and the red dotted arrow is the path if the condition is false.
Create New Decision Parameter

1. Make sure the **Decision** workflow element in the schema is **selected** (not shown)
2. Select the **Decision** tab in the lower left corner
3. Click **Not set (NULL)** to open the Chooser... window
4. In the new window, click **Create parameter/attribute in workflow**
1. In the Name field, enter **counter**
2. Change the Type to **number**
3. Enter a **0 (zero)** in **Value** field
4. Click **Ok**
Configure Decision

1. In the Decision tab in the lower pane, click the dropdown menu and change the value to **greater or equals**
2. Enter a value of **10** in the blank text field

The counter attribute defined in the previous step is used to defined how the Decision behave:

- If the counter value is **less than** 10, the path will be the **red dotted** one.
- If the value is **greater than or equal** to 10, the path will be the **green** one.

Add Increase counter Object

1. Select **Basic** category from the list on the left
2. Click on the **Increase counter** object and drag it into the workflow, making sure to drag it to the dotted red line under the Decision object
Bind Variables

1. Click the **Setup...** button in the upper right to proceed with the binding of the parameters of the element.
2. Click **Promote** to import all parameters from the Increase counter object to the workflow.

This automatically binds the attribute **counter** to the IN and the OUT parameters of the element.
Delete Workflow Branch

In order to make a loop, the second end of the workflow must be deleted. An End element can't be deleted by selecting it, so instead, the arrow preceding the element must be deleted.

1. Right-click on the blue arrow under the Increase counter object
2. Click on Delete

The End element should disappear with the arrow.
Add Scriptable Task

1. Select **Generic** category
2. Drag and drop **Scriptable task** element to **Increase counter** element in the schema, which will add the task to the right
3. Click the **Undock** icon to open the Scriptable task object in a new window

Undock Scriptable task configuration

1. Make sure the **Scriptable task** is selected
2. Click the **Undock** button

**Bind Parameter to Scriptable Task**

1. Click the **Visual Binding tab**
2. Drag and drop **counter** from **In Attributes** to **IN**
Add Script to Log Counter Value

1. Select the **Scripting** tab
2. Enter the following code:

   ```javascript
   System.log("Current counter: " + counter);
   ```
3. Click the **X** to close the window and return to the Schema pane
Add Next Item to Scriptable Task

1. Move the mouse over the **Scriptable task**, then over the **blue arrow** that appears along the right side of the object
2. Drag and drop the **blue arrow** to the **Decision** box

(Note: if necessary, click the dropdown menu above the schema to increase the zoom % from the present 50.)

The **loop** is created.

This visual representation is very similar to the following code in other languages:

```java
for(counter=0; counter <= 10; counter++){
    log("Current counter: " + counter);
}
```
Run the Workflow

1. Click on Run button
2. Click the Logs tab to view the output of the workflow execution
3. Adjust the size of the right pane if necessary.
4. Observe the log output showing the counter value increasing until it reaches 10, at which time the workflow ends
Create an Infinite Loop

Sometimes, an error in developing a workflow can lead to an infinite loop. Let's see how it can happen.

1. Click on the **Increase counter** object in the workflow schema
2. Select the **OUT** tab in the lower pane
3. Click the **counter** parameter to select it
4. Click on **Remove parameter** button
Run and cancel infinite loop

With the counter output parameter removed, the value of the counter is never updated. The condition in the decision will thus always be false (it is never greater or equal than 10), and this workflow will result in an infinite loop. To observe this, run the workflow.

1. Click Run to execute the workflow
2. Observe the log, which will continue to show new messages with a counter value remaining at 0
Cancel Workflow

This workflow will continue to run indefinitely. In order to stop the workflow execution, the only way is to cancel the workflow.

1. Click the **Cancel** icon
2. In the Cancel Workflow window, click the **Yes, cancel it** button

Save and close

1. Click on **Save and close** button

This exercise is complete. Out-of-the-box workflow elements allow very easy and visual conditional creation and execution sequences in any workflows. As always, with loops there is a risk of an infinite loop with an improperly configured workflow, but it can be easily detected and stopped by canceling the workflow manually.
Bring it together: Sample Advanced Workflow

Now let's take everything we've learned so far, and use it to create an advanced workflow within vRealize Orchestrator. This workflow provisions a vRealize Automation machine and waits for the provisioning task to complete. Then, it retrieves the Item.

Remember, Drag and Drop can be used to enter code from HOL manual to workflow Scriptable tasks.

Create a New Workflow

1. Make sure Design view is selected
2. Select Workflow tab
3. Right Click on HOL > 1921-06-CMP > Module 1 workflow folder
4. Click on New workflow
Name New Workflow

1. Enter "Create Machine and Wait" as workflow name
2. Click OK

Reset Schema Zoom

After completion of the previous exercise, the schema section may be set to a lower zoom level than usual. Before starting, the zoom can be can be changed.

1. Click on the Schema tab
2. Click on the dropdown menu in the top menu, and change the value to 100
Add an Existing Workflow

To begin, an existing out-of-the-box workflow is leveraged to considerably save time and effort.

1. Select **All Workflows** from the list on the left
2. Open the **administrator @ vra-01a.corp.local** menu (not shown) and navigate to **Library -> vRealize Automation -> Requests**
3. Drag and drop the **Request a catalog item with provisioning request** workflow onto the arrow between the start and end objects
4. Setup

---

---
Bind Parameters

1. Make sure the workflow element is selected
2. Click the Setup... button in the "Do you want to add the activity's parameters..." message that appears above the schema
3. For the request parameter under Promote Workflow Output Parameters, change the Mapping Type to Local variable
4. Click Promote

In this situation, the OUT parameter request is used inside the workflow, so it is configured as a Local variable instead of an output parameter.

Create a Loop to Wait

Now that the request has been configured to request a vRealize Automation catalog item, a loop is required to wait for the end of the provisioning of the machine. In addition, a timeout will be implemented to fail the workflow if the machine takes too much time to provision.
Add Decision Box

1. Click on the **Generic** menu item in the list on the left
2. Click on **Custom decision** and drag it onto the schema. Make sure to drag the Custom decision object onto the arrow to the right of the workflow, as shown

Name Custom Decision

1. Enter the name **Request done?** in the **Name** field
2. Set the **Success** to **End workflow**
3. Set the **Failure** to **End workflow**
1. Make sure the newly added Custom decision object is selected
2. In the lower pane, Make sure the Info tab is selected
3. For Name enter Request done?

**Bind Decision Inputs**

1. Select the IN tab in the lower pane (note that upon changing tabs, the Custom decision object name has changed)
2. Click the **Bind to workflow parameter/attribute** icon
3. In the Chooser... window that opens, click the checkbox next to request
4. Click Select

**Configure Decision**

1. [Diagram showing the configuration process]
2. [Code snippet: return if request.getExecutionStatus().value() == "STOPPED"]
1. Select **Scripting** tab  
2. Enter the following code:

```java
return (request.getExecutionStatus().value() == "STOPPED")
```

By setting this condition, the green path will be used only when the execution of the request is stopped (which means the provisioning has either succeeded or failed.)

**Add Sleep Box**

1. Select **Basic** category  
2. Drag and Drop **Sleep** workflow element on the **false** (red dotted line) of the decision box, below the Request done? custom decision
Configure Sleep Box

1. Click the **Setup...** button
2. Click on **Value** radio button
3. Enter value **20** (for 20 seconds)
4. Click on **Promote** button

This sleep box allows the workflow to wait 20 seconds between each request status check.
Add Increase Counter Element

1. Drag and Drop **Increase counter** workflow element after the Sleep element added previously (but do not click the Setup... button this time)

A counter is required to manage a timeout in the loop.
Set Source Parameter for Input

1. Make sure the workflow element **Increase counter**
2. Select **IN** tab
3. Click on **not set** for the unique Local Parameter "counter"
4. Click on the link **Create parameter/attribute in workflow** in the Chooser... window (not shown)
Create Attribute Value for Counter

1. Enter 0 (zero) in the Value field
2. Click on Ok button
Set Source Parameter for Output

1. Select **OUT** tab
2. Click on **not set** for the unique Local Parameter "counter"

Set Source Parameter for Output (Continued)
1. Select attribute **counter** in the list
2. Click on **Select** button

**Remove end of the workflow**

1. Right click on the **blue arrow** after the counter element
2. Click on **Delete**

Note: If clicking on the arrow is difficult, it is possible to change the zoom level using the dropdown at the top, as done at the start of this exercise.
Add Decision Element

1. Select **Generic** category
2. Drag and drop the **Decision** element on top of the **Increase counter** workflow element.
   The Decision element will be added to the right of the Increase counter element.
1. Make sure the **Decision** workflow element is selected
2. Select the **Info** tab in the lower pane
3. For Name, enter **no timeout?**
Configure Decision Box

1. Select Decision tab
2. Click on Not set (NULL) link
3. In the Chooser... window, select counter attribute from the list
4. Click on Select button

Set Timeout Threshold

1. Select smaller or equals from the dropdown
2. Enter **30** in the text box *(30 x 20sec sleep = 10min timeout)*

**Add Scriptable Task**

1. Select **Generic** category, if it is not already selected
2. Drag and drop the **Scriptable task** element onto the **Decision** workflow element. The new Scriptable task will appear below the Decision element.
Arrange Workflow Schema

1. Click and drag the **Scriptable task** element above the decision box. Note that the arrow follows the element as you drag it to the new location.

This is a best practice to re-arrange the workflow elements to make the overall workflow more readable.
Bind IN & OUT Parameters

1. Click on **Scriptable task**, if it is not already selected
2. Click the **Undock** icon in the lower right, to expand the lower pane into a new window
Bind IN & OUT Parameters (Continued)

1. Select Visual Binding tab
2. Drag and drop request attribute from In Attributes to IN
3. Drag and drop request attribute from Out Attributes to OUT
Add javascript

1. Select **Scripting** tab
2. Enter the following code (if drag and drop is used, make sure to keep the focus in the scripting tab all the time):

```javascript
//Refresh request
var host = vCACCAFEEntitiesFinder.getHostForEntity(request);
request = vCACCAFEEntitiesFinder.getCatalogItemRequest(host, request.id);

//Log state and execution status
System.log(request.getStateName() + "(" + request.getExecutionStatus().value() + ")");
```

3. Click the **X** to close the window and return to the schema pane

This step updates the *request* object to make sure that it contains an updated version with the latest status. Depending of the plugin used, this step may or may not be required. There is no golden rule, so the only way to know if this is needed or not is to test the behavior.
1. Move your mouse over **Scriptable task**, then on the **blue arrow**
   Drag and drop the **blue arrow** to the decision box "Request done?"

**Add Exception End**

1. Select **Generic** category, if it is not already selected
2. Scroll to find **Throw exception**
3. Drag and drop **Throw exception** to the decision box **no timeout?**
1. Make sure the Exception workflow element is selected (not shown)
2. Select Exception tab
3. Click on the link Not set next to Throw exception binding
4. In the first window that appears, click on the link Create parameter/attribute in workflow to open a second window
5. Click OK, as all of the defaults in this window will suffice
Add Scriptable Task for Error Message

1. Scroll to the top of the Generic category to find Scriptable task
2. Drag and drop Scriptable task to the red dotted arrow between the decision box and the exception
Add IN Parameters

1. In the lower pane, select the IN tab
2. Click on **Bind to workflow parameter/attribute** button
3. Mark the checkbox for request, sleepTime, counter attributes
4. Click **Select**
Add OUT Parameters

1. Select the OUT tab
2. Click on **Bind to workflow parameter/attribute** button
3. Mark the checkbox for **errorCode** attributes
4. Click **Select**

**Add javascript**

1. Select **Scripting** tab
2. Enter the following code:

```javascript
errorCode = "Request " + request.getRequestNumber() + ", reached the timeout of " + (counter * sleepTime) + " seconds."
```
errorCode = "Request " + request.getRequestNumber()
+ " reached the timeout of "
+ (counter * sleepTime) + " seconds."

Add Scriptable Task to Get Resource Item

1. Drag and drop an additional **Scriptable task** element onto the **arrow** between the decision box and the end of the workflow
Add IN Parameters

1. In the lower pane, select the **IN** tab
2. Click on **Bind to workflow parameter/attribute** button
3. Mark the checkbox for **request** attribute
4. Click **Select**
Add OUT Parameters

1. Select the OUT tab
2. Click on **Bind to workflow parameter/attribute** button
3. In the new window, click on **Create parameter/attribute in workflow** to open an additional window
4. Enter `resource` in the textbox **name**
5. Enter `CatalogResource` in the **Filter** textbox
6. Select `vCACAFE:CatalogResource` from the results
7. Select **Create workflow OUTPUT PARAMETER with the same name** radio button
8. Click **Ok**
1. Select **Scripting** tab
2. Enter the following code (if drag and drop is used, make sure to keep the focus in the scripting tab all the time):

   ```javascript
   //Get Consumer Resource Service (to query items)
   var host = vCACCAFEEntitiesFinder.getHostForEntity(request);
   var service = host.createCatalogClient().getCatalogConsumerResourceService();

   //Prepare Filter
   var filter = new Array();
   filter[0] = vCACCAFEFilterParam.equal("request/id", vCACCAFEFilterParam.string(request.getId()));
   filter[1] = vCACCAFEFilterParam.equal("resourceType", vCACCAFEFilterParam.string("Infrastructure.Virtual"));
   var query = vCACCAFEOdataQuery.query().addFilter(filter);

   // get items
   var items = service.getResourcesList(new vCACCAFEPageOdataRequest(query));
   resource = items[0];
   ```

This code allows the task to retrieve the newly provisioned item from the request.
Run the Workflow

1. Click the **Run** button to execute the workflow. This will open the Start Workflow window and prompt for input.

Enter Workflow Inputs

1. For Description, enter **automated creation**
2. Click on **Not set** under Catalog item to open a new window
Choose a vRealize Automation Catalog Item to Request

1. In the Select window, navigate to vRealize Automation -> vRA - DevMgr [https://vra-01a.corp.local] [vsphere.local] -> Catalog and Select CentOS
2. Click on Select to close this window
3. In the Start Workflow window, click on Submit (not shown) to begin the workflow execution
Monitor the Workflow

Before the workflow completes make sure you go to the next step. The workflow execution can take 5 to 15 minutes to complete (depending of the load on the HOL platform).

Observe how the schema is flowing, and the log messages as they appear in the Logs tab.
Look at the variables

Observe how variables are changing, especially the counter:

1. On the right, click on the **Variables** tab
2. View the **counter** variable, and its current value. Note the counter will increase as it cycles through each wait check.

Open Chrome Browser

Let’s take a look at the status of the VM creation process within vRA.

1. In the Windows Quick Launch task bar, click the **Google Chrome** icon to open the browser
Log In to vRealize Automation

1. Log in to vRealize Automation as **devmgr** with password **VMware1!**
2. Click **Sign in**

Observe Status of Catalog Request

1. Navigate to the **Requests** tab
2. Note the request for a CentOS item, with the description of automated creation as noted upon requesting in the workflow.

The request will show a Status of In Progress until provisioning of the machine is complete. When it is complete, the vRealize Automation request will show a Status of Successful or Failed.

**Look at the Final Variables**

![Image](image.png)

Return to vRealize Orchestrator, and observe the status of the completed workflow.

1. If necessary, expand the right pane to see the values of the variables.
2. Note that in the Variables tab, the resource parameter is now set to a value of the machine's name (this name was generated as part of the vRealize Automation blueprint requested by this workflow).
Look at the Log Again

The workflow log shows a final status of **Successful (STOPPED)**.

Destroy Machine in vRealize Automation

1. Switch to vRealize Automation in the Chrome browser (not shown,) and navigate to the **Items** tab
2. Select the line containing the **CentOS** item (do not click on the Name itself, or you will be taken to the Item Details screen)
   If nothing is shown, make sure the request was successful.
3. Click **Actions**
4. From the dropdown menu, select **Destroy**

**Destroy Machine in vRealize Automation (Continued)**

1. Click **Submit** to submit the destroy request
2. Click **OK** (not shown) when notified of successful request submission

Interacting with vRealize Automation is easy using out-of-the-box workflows and basic concepts, such as loops.

This allows an administrator to provide additional automation on top of the machine provisioning by creating a new machine, returning a reference to the newly created machine, and then executing additional actions with the machine such as configuration management, software installation, and more.

For more examples of additional options beyond machine creation, see the other vRealize Automation hands-on labs.
Asynchronous Workflows

Asynchronous workflows allow multiple tasks within a workflow to be launched without waiting for an individual task to complete! This is a good way to trigger multiple actions in parallel, or to track different parts of a process as separate workflow tokens.

Create a New Workflow

1. Make sure Design view is selected
2. Select Workflow tab
3. Right Click on HOL > 1921-06-CMP > Module 1 workflow folder
4. Click on New workflow
Name the New Workflow

1. Enter the name of the workflow: **Asynch**
2. Click **Ok**

Add Asynchronous Workflow

1. Click on the **Schema** tab
2. Select the **Generic** category from the list on the left, if it is not already selected
3. Scroll down until **Asynchronous workflow** element appears.
4. Click on **Asynchronous workflow** and drag it to the arrow between the start and end elements. Once this is done, a Search box will pop up.
Choose Workflow

1. Enter Create Machine and in the Search field
2. In the results, select the Create Machine and Wait workflow (created in the previous exercise) by double-clicking
Bind Parameters

1. Make sure the workflow element is selected (not shown)
2. Click the Setup... button in the upper right corner of the schema after choosing the Create Machine and Wait workflow
3. In the Promote Workflow Input/Output Parameters window, click Promote to accept all of the default values

Run the Workflow
1. Click **Run** to execute the workflow and open the Start Workflow window

### Enter Workflow Input

Notice that the input prompt is the same as the previous exercise. The same workflow is executed, but this time it is done from a parent workflow and in an asynchronous manner.

1. For Description, enter **automated creation**
2. Click **Not set** under Catalog item
Choose a vRealize Automation Catalog Item to Request

1. In the Select window, navigate to vRealize Automation -> vRA - DevMgr [https://vra-01a.corp.local] [vsphere.local] -> Catalog and Select CentOS
2. Click Select to close this window
3. In the Start Workflow window, click Submit (not shown) to begin the workflow execution

Save and Close

1. Click on Save and close to close this workflow
Observe Variables for Completed Workflow

1. In the Workflows list, expand the **Asynch** workflow to observe the recent run.
2. Note the **green checkmark** indicating that this workflow execution was successful. Click on the specific **Asynch** item to view the results.
3. Select the **Variables** tab on the right.
4. Observe the **wfToken** attribute; it is set to **Create Machine and Wait** and represent the workflow asynchronously executed.

Look at Other Workflow

1. Expand the **Create Machine and Wait** workflow to observe recent activity for that workflow.
2. Locate the most recent run of this workflow. Note that it is still running, while the **Asynch** workflow has already completed.

Open Chrome Browser

Let’s return to vRealize Automation and remove the newly created instance.
1. In the Windows Quick Launch task bar, click the **Google Chrome** icon to open the browser

**Log In to vRealize Automation**

1. Log in to vRealize Automation as **devmgr** with password **VMware1!**
2. Click **Sign in**
1. Navigate to the **Items** tab
2. Select the line containing the **CentOS** item (do not click on the Name itself, or you will be taken to the Item Details screen). Note that the item takes couple of minutes to appear in the item list, click the refresh button until it appears.
3. Click **Actions**
4. From the dropdown menu, select **Destroy**
1. Click **Submit** to submit the destroy request
2. Click **OK** (not shown) when notified of successful request submission

The exercise is complete. To go further, it is possible to use a loop to wait for the end of the asynchronous workflow by monitoring the workflow token state value.
Conclusion

vRealize Orchestrator can be used to automate complex processes that requires polling for status, launching multiple tasks at the same time (or asynchronously,) and using advanced input presentation. In conjunction with vRealize Automation, it can be very useful to automate processes around machine provisioning by using the power of vRealize Automation for the creation and the flexibility of Orchestrator for input parameters or third party integrations.

You've finished Module 1

Congratulations on completing Module 1.

If you are looking for additional information on vRO Coding Guidelines, try one of these:

- Go to http://bit.ly/2tKxJU6
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
• **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.

• **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced) Discover the various ways to add error or exception handling to ease in troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

**How to End Lab**

To end your lab click on the **END** button.
Module 2 - Integrating with REST APIs (45 min)
Introduction

According to Wikipedia, Representational State Transfer (REST) is an architectural style that defines a set of constraints and properties based on HTTP. Web Services that conform to the REST architectural style, or RESTful web services, provide interoperability between computer systems on the Internet. REST-compliant web services allow the requesting systems to access and manipulate textual representations of web resources by using a uniform and predefined set of stateless operations. Other kinds of web services, such as SOAP web services, expose their own arbitrary sets of operations.

vRealize Orchestrator provides a plugin to interact with REST web services easily and efficiently.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: **vra-01a.corp.local:443**
2. User name: **administrator@vsphere.local**
3. Password: **VMware1!**
4. Click **Login**

Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Configuring the vRealize Orchestrator Plug-in for REST

At Rainpole, they want to start using REST APIs as a standard for interfacing vRealize Orchestrator with third party systems that do not already have a plug in available. In an effort to learn more about this functionality, let's get familiar with vRealize Orchestrator REST plugin by using NSX Manager from a lab environment as a REST host and learn more about the configuration.

Note: NSX has its own plugin which is better and easier to use than the REST API. However for the purpose of this lab, this REST API will be studied.

Adding a REST Host

1. Select the Workflow tab
2. Navigate to Library > HTTP-REST > Configuration
   Select Add a REST host workflow
3. Click the Start workflow icon

Interacting with a NSX Manager via REST requires to add NSX Manager as a REST host in vRealize Orchestrator.
The REST host needs to be added only once.
1. Type **NSX** for the REST host name. This is the user friendly name that will appear in the vRealize Orchestrator inventory view.
2. Type **https://nsxmgr-01a.corp.local** for the URL. This is the URL to the NSX Manager in the lab environment.
3. Select **Yes** to accept the hosts certificate silently.
4. Click the **Next** button.
Host Authentication

There are several options for authentication, for NSX we need to use Basic. Take a moment to look through the other authentication method options.

1. Click the drop down menu and select **Basic** for the authentication type
2. Click the **Next** button
User Credentials

Select a session mode of either a Shared Session or Per User Session. In Per User Session, the credentials are used from the currently logged in user. In this case, Shared Session is the chosen option as every person using vRO should be able to execute a workflow against NSX manager API regardless of their own privileges in NSX.

1. Select **Shared Session** from the drop down menu
2. Type **admin** for the username
3. Type **VMware1!** for the password
4. Click the **Next** button
Proxy Settings

1. Add a REST host
   1a. Host Properties

2. Host Authentication
   2a
   2b. User credentials

3. Proxy Settings
   3a. Proxy Settings

4. Advanced
   4a. SSL

No proxy is needed leave the defaults.

1. Click the Submit button
Verify the Add a REST host workflow run was successful

The workflow token **Add a REST host** should have a **green checkmark** next to it, this shows that the workflow completed successfully.
Adding a REST operation

A REST operation is how you interact with a REST api. For example, you use a GET operation to make an inquiry or a POST operation to update data or a configuration.

1. Select the **Workflow** tab
2. Navigate to **Library > HTTP-REST > Configuration**
   - Select **Add a REST operation** workflow
3. Click the **Start workflow** icon
1. Click the **Not Set** link for the Parent host

This allows to select the NSX Manager REST host previously added.
Selecting the REST host

1. Browse to **HTTP-REST > NSX**
2. Click the **Select** button
The REST Operation will allow to add a NSX security tag.

1. Enter **NSX Add Security Tag** for the Name
2. Enter for the Template URL:

/api/2.0/services/securitytags/tag

Notice the URL is not a full URL. vRealize Orchestrator will concatenate the REST host URL and the template URL.

3. Select **POST** from the drop down for the HTTP method. Take a moment to look through the other HTTP methods available.
4. Enter **application/xml** for the Content type
5. Click the **Submit** button
Verify the Add a REST operation workflow run was successful

The workflow token **Add a REST operation** should have a **green checkmark** next to it, this shows that the workflow completed successfully.
Invoking REST operations via vRealize Orchestrator

NSX Manager from the lab has been added to vRealize Orchestrator as a REST host and one REST operation has been created. The REST operation will be invoked to interact with NSX REST API. By invoking the REST operation previously created, it will add a new NSX security tag to NSX.

Invoke a REST operation

1. Select the **Workflow** tab
2. Browse to **Library > HTTP-REST > Invoke a REST operation**
3. Click the **Start workflow** icon
Select the REST Operation

1. Click **Not Set**, to select the REST Operation

Select the REST Operation from the inventory
1. Browse to **HTTP-REST > NSX > NSX Add Security Tag**

Notice this is the REST operation created in the last part.

2. Click the **Select** button

### Selecting the REST Operation

1. Click the **Next** button

### Input Content

![Input Content Image]

1. **Content type**: `application/xml`

2. **Content**:

   ```xml
   <name>FinanceApp</name>
   <description>Component of the Finance Application</description>
   <extendedAttributes></extendedAttributes>
   </securityTag>
   ```

3. Click the **Submit** button
1. If not already present, enter **application/xml** for the Content type
2. Type the XML code into the Content window.

The code comes from the NSX API guide. The code is creating a NSX Security Tag called FinanceApp with the description of Component of the Finance Application.

```xml
<securityTag>
  <objectTypeName>SecurityTag</objectTypeName>
  <type><typeName>SecurityTag</typeName></type>
  <name>FinanceApp</name>
  <description>Component of the Finance Application</description>
</securityTag>
```

3. Click the **Submit** button
Verify the Invoke a REST operation workflow run was successful

The **Invoke a REST operation run** should have a **green checkmark** next to it, this shows that the workflow completed successfully.
Let's verify it in vCenter.

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

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

**Logging into the vSphere web client**

1. Click on the **Region A > RegionA vCenter** bookmark
2. Check the box for **Use Windows session authentication**
This will log you on with the administrator@corp.local credentials

3. Click the **Login** button

**Networking & Security**

1. Click the **Home** menu
2. Click **Networking & Security**

**Groups and Tags**
1. Click the **Groups and Tags** tab

## Security Tags

![Security Tags section with FinanceApp security tag highlighted]

1. Select **Security Tags** tab
2. **FinanceApp** security tag (created by the REST call) is listed
Conclusion

The REST web services are a good way to integrate with a lot of systems. With vRealize Orchestrator, the out-of-the-box REST plugin provides a quick and efficient solution to easily integrate with these systems.

You've finished Module 2

Congratulations on completing Module 2.

If you are looking for additional information on vRealize Orchestrator REST API plugin try one of these:

- Click on this [link](#)
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
- **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.
- **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced) Discover the various ways to add error or exception handling to ease in
troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

**How to End Lab**

To end your lab click on the **END** button.
Module 3 - Launching Workflows from Outside vRealize Orchestrator (45 min)
Introduction

vRealize Orchestrator has tight integration with a lot of VMware products but it can also be easily integrated with any other system by using its REST API.

In this exercise, Postman is used to demonstrate the few REST call needed to integrate with vRealize Orchestrator.

Launch Postman

Double click on the Postman icon on the desktop.
Be familiar with Postman

Postman is a powerful HTTP client for testing web services. Postman makes it easy to test, develop and document APIs by allowing users to quickly put together both simple and complex HTTP requests.

1. The HTTP method and the url can be specified in this form
2. Different aspect of the HTTP request can be configured such as Authentication, Headers and Body.
3. When the HTTP request has been sent, the response is available on the same page.
4. It is possible to store a list of HTTP requests in a Collection
5. Each HTTP Request can use variables that can be stored in an environment configuration.
Open Chrome Browser from Windows Quick Launch Task Bar

Let's take a look at the vRO API documentation.

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

vRO API documentation

1. Enter the URL


A swagger UI is provided to explore vRealize Orchestrator API.
Explore the API

1. Scroll down
2. Click on **workflow-controller** to expand the operations
   By clicking on one operation, it is possible to see the details of the input parameters and a sample of the response message.

vRealize Orchestrator API has multiple controllers, each controller allows to interact with a type of object. In the next step of this module, **workflow-controller** will be used to retrieve a workflow definition. **inventory-service-controller** will be used to retrieve objects from vRO inventory that can be used as input for a workflow. **execution-controller** is used to start a workflow and monitor its status.
Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.

Log In to vRealize Orchestrator

Use the following credentials

1. Host name: vra-01a.corp.local:443
2. User name: administrator@vsphere.local
3. Password: VMware1!
4. Click Login
Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Start a simple workflow

Let see how to start a simple workflow using vRealize Orchestrator API.

Find the workflow ID in vRO client

1. Select **Workflows** tab
2. Navigate to **HOL > 1921-06-CMP > Module 3**
   Select **Hello World** workflow
3. On the General tab in the right panel, make sure to **Scroll up** to the top to see the ID field.
4. **Copy** the workflow ID (Select the ID and Press **Ctrl+C**)

Switch to Postman

1. Click the **Postman** icon in the task bar
Get workflow details

1. **Click** on the collection button to show the sidebar
2. Navigate to **HOL-1921-06-CMP Module 3 > Hello World**
   - **Click** on **Get Hello World** request
3. **Click** somewhere else, outside of the sidebar, to make it disappear
Modify the URL

1. Click the URL textbox to make it expand
2. **Append** the workflow ID (it should have been copied from a previous step) to the URL: e73bad8-177f-499c-b478-bf5bcee770a1
3. Click on **Send** button
Analyze the response

On the **Response** pane (right part of Postman)

1. **Status: 200 OK** indicates that the request is successful
2. Make sure **Pretty** visualization is selected
3. **Scroll down** to the end of the response body
4. name value confirms that it is the right workflow
   there is no input parameter
   there is 1 output parameter
Prepare request to execute the workflow

On the **Request** pane (left part of Postman)

1. Change the HTTP method to **POST**
2. Click the **URL textbox** to make it expand
3. Append to the URL: `/executions`

Execute workflow

On the **Request** pane (left part of Postman)

1. Select **Body** tab (under the url bar)
2. Select **raw**
3. If not present, Enter the following in the body

```
{}
```
4. Click on **Send** button

vRealize Orchestrator API doesn't accept an empty body for a POST request, so, even if there are no inputs, it must be an empty JSON object.

**Analyze the response**

On the **Response** pane (right part of Postman)

1. Status: **202 Accepted** indicates that the workflow has been successfully started
2. Click the **Headers** tab
3. Look for the **Location** header and **Copy the link** (select the link and Press Ctrl + C)

The **Location** header contains a link to the workflow execution, this link allows to monitor the workflow status
Get workflow status

On the **Request** pane (left part of Postman)

1. Change the HTTP method to **GET**
2. Paste (Press **Ctrl + V**) the URL copied from the Location header (previous step)
3. Click on **Send** button
On the **Response** pane (right part of Postman)

1. Select **Body** tab
2. Make sure **Pretty** visualization is selected
3. **Scroll down** to the end of the response
4. The details of the workflow execution gives the following information:
   - **state: completed** -> workflow is done
   - **output-parameters**: message = Hello World
Understand Authentication in Postman

All requests were executed against vRealize Orchestrator using Basic Authentication. This authentication was configured at the Collection level.

1. **Click** the collection button to show the sidebar
2. **Click** the ... button of **HOL-1921-06-CMP Module 3** collection
3. **Click** the **Edit** button
Observe Authentication configuration

1. Select Authorization tab
2. Basic Auth is used for the type of authentication
3. Username and password used are the same as the one used to login into the vRealize Orchestrator client
4. Click Cancel to exit the window

Switch to vRealize Orchestrator Client

1. Click the VMware vRealize Orchestrator icon in the task bar
Verify in vRO client

1. Navigate to **HOL > 1921-06-CMP > Module 3**
   Select the last **Hello World** token
2. Select **Variables** tab
3. Check the value for the output parameter **message**

**Quick summary**

With only 3 calls, it is possible to:

- Get the details regarding a workflow, especially the expected inputs and outputs
- Start a workflow
- Monitor the workflow status
Start a workflow with complex inputs

Some workflow requires to choose an object from the vRO inventory. From an API standpoint, it requires an additional call to retrieve the vRO ID of this object before passing it to the workflow input.

Look at "Start virtual machine and wait" workflow

1. Enter "start vi" in the search box and Press Return
2. Double-Click on the workflow Start virtual machine and wait
3. Click on Close button
Get the ID

1. Select **General** tab
2. On the General tab in the right panel, make sure to **Scroll up** to the top to see the ID field.
3. Copy the ID (**Triple-Click** allows to get the whole ID, then **Ctrl + C** to copy)

Switch to Postman

1. Click the **Postman** icon in the task bar
Get workflow details

1. **Click** on the collection button to show the sidebar
2. Navigate to **HOL-1921-06-CMP Module 3 > Complex workflow**
   - Click on **Get Start VM wf** request
3. Click somewhere else, outside of the sidebar, to make it disappear
1. Click the **URL textbox** to make it expand
2. **Append** the workflow ID (from previous step) to the URL: 
   \[ BD808080808080808080808080808080CC280800122528313869552e41805bb \]
3. Click on **Send** button
1. Status: **200 OK** indicates that the request is successful
2. Scroll down to the end of the response body
3. In this case **input-parameters** are complex object (**VC:VirtualMachine**).
   To be able to start the workflow, it is required to get the object ID of the complex object.
Get VM object

1. Click on the collection button to show the sidebar
2. Navigate to **HOL-1921-06-CMP Module 3 > Complex workflow**
   Click on **Get VM object** request
3. Click somewhere else, outside of the sidebar, to make it disappear
The following URL is used to browse vRO inventory to find object where the type is VC:VirtualMachine and the name of the machine is "base-linux-cli"


- VC is the plugin namespace (found before ':' in the complex object type)
- VirtualMachine is the object type (found after ':' in the complex object type)
- conditions allows to narrow down the research to one machine by filtering by name

1. Click on Send button
Analyze the response

1. Status: **200 OK** indicates that the request is successful
2. **Scroll down**
3. **Find** the key/value pair where the key name is **sdkId**
   - Copy (Select and Ctrl + C) the value, it is used later to create a workflow execution.
Execute a workflow

1. Click on the collection button to show the sidebar
2. Navigate to **HOL-1921-06-CMP Module 3 > Complex workflow**
   - Click the **Execute Start VM wf request**
3. Click somewhere else, outside of the sidebar, to make it disappear
Execute workflow with complex input

1. Select **Body** tab
2. Insert the ID (Ctrl + V) retrieved in the previous step between the current blank ID values of "" to match the picture above

```
vcsa-01a.corp.local/vm-221
```

3. Click on **Send** button
Analyze the response

1. Status: **202 Accepted** indicates that the workflow has been successfully started.
2. Click on **Headers** tab.
3. The **Location** header contains a link to the execution, this link allows to monitor the workflow status.
   **Copy the link** (select the link and Press Ctrl + C)
Get workflow status

1. Change the HTTP method to **GET**
2. **Paste** the **URL** copied from the Location header (previous step)
3. Click on **Send** button
Analyze response

1. Select **Body** tab
2. **Scroll down** to find the workflow information (see picture above)
3. The details of the workflow execution gives the following information:
   - state: completed -> workflow is done
   - If the workflow is not completed, you can send this request multiple time to see the status change.

**Switch to vRealize Orchestrator Client**

1. Click the **VMware vRealize Orchestrator** icon in the task bar
Verify in vRO client

1. **Expand** Start virtual machine and wait
   **Select** the last token
2. Select **Variables** tab to check the inputs and attributes
3. Notice when the workflow completes, the task attribute has the value
   `VirtualMachine.powerOn`
Conclusion

On top of a lot of out-of-the-box capabilities, vRealize Orchestrator provides an easy to use REST API to integrate the orchestrator with any other products. vRealize Orchestrator is definitely a central tool to build around for every automation project.

You've finished Module 3

Congratulations on completing Module 3.

If you are looking for additional information on vRealize Orchestrator REST API try one of these:

- Click on this link
- Or use your smart device to scan the QRC Code.

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

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
- **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.
• **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced)
  Discover the various ways to add error or exception handling to ease in troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

**How to End Lab**

To end your lab click on the **END** button.
Module 4 - Automating vRealize Automation with vRealize Orchestrator (45 min)
Introduction

Rainpole company has to dedicate a lot of resources to maintain and manage their vRealize Automation implementation. Each time a new project is on boarded there are many tasks that need to be performed. These tasks include business group creation, creating reservations, building and assigning entitlements, etc...

It is very time consuming and it is also prone to human errors, delaying the on boarding and creating frustration for end-users.

Peter, the head of the Cloud Automation team, is thinking about automating this process. Unfortunately he has only a few employees, and they don't have time to spend on exploring vRealize Automation API to find out how to automate this process. They are, however, already using vRealize Orchestrator to integrate their CMDB and IPAM with virtual machine provisioning. Fortunately vRealize Orchestrator already contains a lot of out-of-the-box content to automate vRealize Automation configuration.

With their vRO knowledge and the existing content, the automation of the onboarding process should be quick!

In this module, explore what is possible to automate the configuration of vRealize Automation.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: `vra-01a.corp.local:443`
2. User name: `administrator@vsphere.local`
3. Password: `VMware1!`
4. Click `Login`

Select the Design View

1. Click the dropdown menu at the top of the client, and select the `Design` View
Explore out-of-the-box workflows

vRealize Orchestrator has numerous out-of-the-box workflows to easily manage vRealize Automation. They can be found in the vRO folder Library > vRealize Automation > Administration.

Create a business group

1. Navigate to Library > vRealize Automation > Administration > Business Groups
   Select workflow Create a business group
2. Click on Start a workflow... button
Select vRA Host

1. Click on **Not set** (not shown in the screenshot)
2. Expand **vRealize Automation**
3. Select **vRA - CloudAdmin**
4. Click on **Select** button
5. Click on **Next** button
Specify business group configuration

1. Enter the name for the new business group: **Development 2**
2. Enter the manager email: **devmgr@corp.local**
3. Click on **Not set** for the group manager role
4. Enter **devmgr@corp.local** in New value textbox
5. Click on **Insert value**
Accept the list of value

Move the window up to show the Accept button at the bottom

1. Click on Accept button
Submit the workflow

1. Click on **Submit** button
Create a new entitlement

1. Navigate to **Library > vRealize Automation > Administration > Entitlements**
   Select the workflow **Create an entitlement for subtenant**
2. Click on **Start a workflow...** button
Select vRA Host

1. Click on **Not set** for vRA CAFE host field (not shown in the screenshot)
2. Expand **vRealize Automation**
3. Select **vRA - CloudAdmin**
4. Click on **Select** button
Specify entitlement configuration

1. Enter the name for the entitlement: **Dev2-MyCommerce**
2. Click on **Not set** for **Business group**
Select business group

1. Navigate and Select vRA - CloudAdmin > Administration > Business Groups > Development 2
2. Click the Select button
Execute the workflow

1. Click the **Submit** button
2. (not shown) Make sure the workflow ends successfully (green checkmark)

Open Chrome Browser from Windows Quick Launch Task Bar

Let’s open vRealize Automation and see how the Business Group was added.

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.
Log In to vRealize Automation

1. Click on **vRealize Automation** bookmark
2. Enter **cloudadmin** for username
3. Enter **VMware1!** for password
4. Click the **Sign in** button
Verify new business group

1. Select the Administration tab
2. Navigate to Users & Groups > Business Groups
3. Click on Development 2 business group

Explore the business group configuration and verify it matches what was used in the workflow.

Verify Entitlement
1. Click on Administration parent menu to come back to the main administration menu
2. Navigate to Catalog Management > Entitlements
3. Observe the new entitlement Dev2-MyCommerce created through vRealize Orchestrator

**Explore the content available in vRealize Orchestrator**

![vRealize Automation Administration menu]

Let’s return to vRealize Orchestrator by click on VMware vRealize Orchestrator in the task bar (not shown).

Navigate to Library > vRealize Automation > Administration and observe all the pre-existing content that can help cloud administrator to better manage vRealize Automation every day. Note that some workflow folder might be already expanded.
Create an automated onboarding process

vRealize Orchestrator provides a lot of pre-existing content to manage vRealize Automation. However to provide a better value, these basic workflows need to be assembled together to provide meaningful operations for cloud administrators.

Creating a Business group or an entitlement is not enough to onboard new groups, it also requires services and users to be properly assigned as well as reservation created to be able to deploy a machine.

In this exercise, the final workflow will provide this end-to-end service for onboarding a new group of users to vRealize Automation through vRealize Orchestrator.

Create a new workflow

1. Navigate to HOL > 1921-06-CMP and select the folder Module 4
2. Click on New workflow button
3. In the pop-up window, Enter the name of the workflow: Onboard a new team
4. Click on Ok button
Create a Business Group

The first thing that needs to be created is a business group.

1. Select **Schema** tab
2. Select **All Workflows** tab
3. Navigate to **Library > vRealize Automation > Administration > Business Groups**
4. Drag and Drop the workflow **Create a business group** to the blue arrow in the schema
Setup inputs for business group

To make the on-boarding process quick and easy, the number of mandatory inputs will be reduced to a minimum. Each input that are not required will be set to null (Skip) or hardcoded.

Make sure the workflow element **Create a business group** is selected.

1. Click on **Setup...** button
2. Select **Skip** for the following input parameters:
   - description
   - defaultMachinePrefix
   - activeDirectoryContainer
   - support
   - sharedAccess
   - properties
3. Select **Local variable** for the single **output** parameter
4. Click on **Promote** button

### Create an Entitlement

1. Make sure **All Workflows** tab is selected
2. Navigate to **Library > vRealize Automation > Administration > Entitlements**
3. Drag and Drop the workflow **Create an entitlement for subtenant** to the blue arrow after the workflow element **Create a business group**
Undock visual binding

1. Select the workflow element **Create an entitlement for subtenant**
2. Click on the **Undock** button
Bind inputs

1. Select **Visual Binding** tab
2. Drag and Drop **host** from IN to **host** in In parameters
3. Drag and Drop **subtenant** from IN to **group** in In Attributes
4. Drag and Drop **entitlement** from OUT to the **white part** in Out Attributes (it creates a new attribute automatically)
5. Click **Ok** on the popup window (not shown in the screenshot)
6. **Close** the window

---

**Example Diagram:**
- ![Diagram of Visual Binding process](image-url)
Setup inputs for entitlement

1. Make sure the workflow element Create an entitlement for subtenant is selected (dark blue)
2. Click on Setup... button
3. Expand the window by drag and dropping lower right corner
4. Change the name of the 2nd parameter to entitlementName (in the first column)
   Select Value for this parameter
5. Select Skip for the following parameters: description, expirationDate
6. Select Value for the input shouldActivate
   Click on Input value and Select Yes
7. Click the Promote button
Assign all users

1. Make sure **All Workflows** tab is selected
2. Navigate to **Library > vRealize Automation > Administration > Entitlements**
3. Drag and Drop the workflow **Assign all users and groups to an entitlement** to the blue arrow after the workflow element **Create an entitlement**
Setup input value for all users

1. Make sure the workflow element **Assign all users and groups to an entitlement** is selected (dark blue)
2. Click the **Setup...** button
3. Click the **Promote** button
Assign Services

1. Make sure **All Workflows** tab is selected
2. Navigate to **Library > vRealize Automation > Administration > Entitlements**
3. Drag and Drop the workflow **Assign services to an entitlement** to the blue arrow after the workflow element **Assign all users and groups to an entitlement**
1. Make sure the workflow element **Assign services to an entitlement** is selected (dark blue)
2. Click the **Setup...** button
3. Select **Skip** for the following parameters:
   - **approvalPolicy**
4. Click the **Promote** button
Creating a reservation is not something provided out of the box, however in this lab, this very simple workflow is provided as a workaround. This workflow doesn't allow to setup all the aspect of a reservation but instead take an existing reservation as a "template" and create a copy assigned to a given business group.

1. Make sure **All Workflows** tab is selected
2. Navigate to **HOL > 1921-06-CMP > Module 4**
3. Drag and Drop the workflow **Create Reservation** to the blue arrow after the workflow element **Assign services to an entitlement**
Setup inputs for Reservation

1. Make sure the workflow element **Create Reservation** is selected (dark blue)
2. Click on **Setup...** button
3. Select **group** from the dropdown list for the first parameter
4. Rename the 2nd parameter to **reservationName** and Select **Value**
5. Rename the 3rd parameter to **reservationToDuplicate** and keep it as **Input**
6. Click the **Promote** button
Manage Naming

Reservation and Entitlement needs a name to be created. To create consistencies in the naming, each of these names will be generated based on the Business Group name.

1. Select Generic tab
2. Drag and drop a Scriptable task element at the very beginning of the workflow (just after the green arrow)

Expand visual binding

1. Select the Scriptable task element
2. Click on the Undock button
Bind input/outputs

1. Select **Visual Binding** tab
2. Drag and Drop **name** from In parameters to the white part of IN
3. Drag and Drop **reservationName** from Out Attributes to the blank part of OUT
4. Drag and Drop **entitlementName** from Out Attributes to the blank part of OUT
Assign names

1. Select the **Scripting** tab
2. **Enter** the following script

```java
entitlementName = name + "-MyCommerce";
reservationName = name + " Reservation";
```

3. **Close** the window
Save and close

1. Click the **Save and close** button

Run the workflow

1. Navigate to **Onboard a new team**
2. Click the play button to run the workflow
1. Make sure the workflow **On-Board a new team** is selected
2. Click the **Start workflow...** button

## Fill inputs

For this example, the QE team will be on boarded to vRA to be able to test the MyCommerce blueprint.

1. Enter the name for the new business group: **QE**
2. Click on **Not set** for the **group manager role**
3. Enter **qemgr@corp.local** in New value textbox
4. Click on **Insert value**
5. Move the window up
   - Click the **Accept** button (not shown)
Fill inputs

1. Enter the **Send manager emails to**: qemgr@corp.local
2. Click on **Not set** for the **User role**
3. Enter **qeuser@corp.local** in New value textbox
4. Click the **Insert value** button
5. Move the window up
   - Click the **Accept** button (not shown)
1. Click on **Not set** for **vRA Host** (not shown)
2. Expand **vRealize Automation**
3. Select **vRA - CloudAdmin**
4. Click the **Select** button
Fill Services to assign

1. Click on **Not set** for Services to assign to the entitlement
2. Click on **Insert value**
Select Services

1. Navigate to vRealize Automation > vRA - CloudAdmin > Administration > Services
   Select MyCommerce
2. Click the Add button
3. Click the Select button
Accept Service list

Move the window up to show the Accept button

1. Click the Accept button
Fill reservation name

1. Fill the reservation to duplicate name: Development Reservation
   This name reference an existing reservation and must be correct.
2. Click the Submit button
Verify the workflow ends well

1. Green checkmark visible
2. In the Logs tab, no errors in the logs

Now let's verify how it looks like in vRealize Automation.

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

Let's look at vRealize Automation to check if the onboarding process has been successful.

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.
Login to vRealize Automation

1. Click on **vRealize Automation** bookmark
2. Enter **qeuser** for username
3. Enter **VMware1!** for password
4. Click the Sign in button
1. Select the **Catalog** tab
2. The user is only allowed to request a catalog item from MyCommerce service
Conclusion

By automating the administrative aspects of vRealize Automation with vRealize Orchestrator, it is very easy to get the most out of vRealize Automation with the minimum time spent on operating the platform.

You've finished Module 4

Congratulations on completing Module 4.

If you are looking for additional information on vRealize Automation API, try one of these:

- Click on this link
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.
- **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced) Discover the various ways to add error or exception handling to ease in
troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

**How to End Lab**

To end your lab click on the **END** button.
Module 5 - Running In-Guest Operations with vRealize Orchestrator (45 min)
Introduction

Cloud administrators often perform operations within the guest operating system of the virtual machines they manage. They don't always have network connectivity and going through the vSphere console is very cumbersome to perform these tasks.

vRealize Orchestrator is here to help, with its native support of the vSphere Guest Operations and the Guest Script Manager, it is now quick and easy to perform operation within the guest of a virtual machine without requiring network connectivity or third party agent. It is free and out of the box.
Prepare the environment

In this module a small linux machine is going to be used to demonstrate guest operations interactions. A couple of steps are required to prepare the environment until the module can start.

Open Chrome Browser from Windows Quick Launch Task Bar

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.
Log In to vCenter Web UI

1. Click on **RegionA** bookmark folder
2. Click on **HTML5 Client** bookmark
3. Enter **administrator@vsphere.local** as user name
4. Enter **VMware1!** as password
5. Click on **Login** button
Select linux machine

1. Enter `linux` in the search bar
2. Click on link `base-linux-cli` in the result list

Start linux machine

1. Click on ACTIONS
2. Click on Power
3. Click on Power On

Wait a few seconds until VMware Tools changes from "Not running" to “Running” status.
Start putty

1. Click on **Putty** icon
2. Scroll down in the Saved Sessions list
3. Select **linux-base-01a.corp.local**
4. Click the **Open** button
Verify login is successful

The following screen should be visible, acknowledging a successful login.
Use Guest Operations

Basic guest operations can be used to perform operations inside the guest operating system such as running a script, uploading or downloading a file. As guest operations relies on VMware Tools to work, no network connectivity is required. In this part, the out-of-the-box workflows will be used.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: vra-01a.corp.local:443
2. User name: administrator@vsphere.local
3. Password: VMware1!
4. Click Login

Select the Design View

1. Click the dropdown menu at the top of the client, and select the Design View
Search for "Run program in guest" workflow

1. Enter "Run program in guest"
2. Press Return

Select "Run program in guest" workflow

1. Double-click on the workflow "Run program in guest"
2. Click on "Close" button
Run the workflow

1. Make sure workflow **Run program in guest** is selected
2. Click the **Start workflow...** button
Common parameters

The first step requires information regarding the machine. The credentials are guest credentials, they are required to let VMware Tools impersonate this user to run the command. This also adds a layer of security as you need to know vCenter credentials AND guest credentials to use this feature.

1. Enter **root** in the username field
2. Enter **VMware1!** in the Password field
3. Click on **Not set** link
1. Navigate to **vSphere vCenter Plug-in** > **VirtualCenter** > **Datacenters** > **RegionA01** > **vm**
2. Select **base-linux-cli**
3. Click on **Select** button
4. Click on **Next** button (not shown in the screenshot)
A bash command will be run, so the path where bash is needs to be inputed.

1. Enter the following path in Program path field:

/bin/bash

2. Click on Next button
Arguments

1. Enter the following in Arguments field:

```
-c "echo vRO rocks > /root/hol.txt"
```

2. Click on Next button
1. Click on **Next** button
Environment

1. Click on **Submit** button
Analyze the workflow run

1. Make sure the workflow run is selected (with the green checkmark)
2. (if needed) Drag and Drop the lower panel to expand it
3. Make sure the Variables tab is selected.
4. Observe the result attribute and save the value (it will be used in the next steps)

The result attribute is the unique process ID (PID) of the process created by guest operations.
Launch "Get processes from guest with logging" workflow

This workflow is the same as the one in Library > vCenter > Guest Operations > Processes > Get processes from guest, but for the purpose of this lab, it contains one more line of code to log the results in vRO logging panel.

1. **Scroll up** if necessary
2. Navigate to the workflow **HOL > Module 1921-06-CMP > Module 5 > Get processes from guest with logging**
3. Click on **Start workflow...** button
Enter username & password

1. Enter **root** in the username field
2. Enter **VMware1!** in the Password field
3. Click on **Not set** link
Select linux VM

1. Navigate to **vSphere vCenter Plug-in > VirtualCenter > Datacenters > RegionA01 > vm**
2. Select **base-linux-cli** virtual machine
3. Click on **Select** button
4. Click on **Submit** button (not shown in the screenshot)
Observe the logs

1. Make sure the workflow run (with the green checkmark) is selected
2. Click on Logs tab
3. Scroll up to see the beginning of the logs
4. process with the same pid should be found at the top

exitCode = 0 means the process has terminated successfully.

If there is no process with an exitCode: 0 at the beginning of the log, look at all the line in the logs. Even if the pid is not found, the lab can be pursued.
Check file created in guest os

1. Switch to putty terminal
2. Enter the following command

```
cat /root/hol.txt
```
3. Press **Return**

The message from vRO workflow appears on the screen!
Use Guest Script Manager

Steve, the Cloud Administrator is called by a developer who mistakenly broke the network configuration of his virtual machine. By using Guest Script Manager, Steve can easily reconfigure the virtual machine for the developer.

Guest Script Manager is a community package simplifying the use of vSphere Guest Operations in vRealize Automation. It can be found by following this link: https://communities.vmware.com/docs/DOC-25474

Note that ifconfig is used to change network configuration. This change is not persistent for the benefit of the lab. In real-life, the proper command should be run to obtain a persistent configuration.

Let's break the network

Switch to putty terminal (if it is not already the case)

1. **Enter** the following command:

   ```
   ifconfig eth0 1.1.1.1 netmask 255.255.255.0
   ```

2. **Press Return**

After couple of minute, Putty throw an error "Network error: Software caused connection abort". The error message can be closed.
Create a new script

Guest Script Manager requires the creation of a "script configuration" before being able to execute a command in the guest os. This allows for the user to specify different parameters as well as upload files (like script file).

1. Navigate to Guest Script Manager > Script Management
2. Select Add script configuration
3. Click the Start workflow... button
1. Enter the configuration name: **Fix Network**
2. Click on **Next** button
Script

1. Select **bash** in the dropdown list
2. Enter the following script content

   ```bash
   ifconfig eth0 {{ipAddress}} netmask 255.255.255.0
   route add default gw 192.168.120.1
   ```

3. Enter **30** for the timeout
4. Enter **5** for the refresh time
5. Click on **Submit** button

Note that the script contains a placeholder `{{ipAddress}}` that will be replaced dynamically during the execution of the script. This feature allows for the creation of modular script for better re-usability and scalability.
Run script in VM guest

1. Select workflow **Run script in VM guest**
2. Click on **Start workflow...** button
Select linux VM

1. Click on Not set for the VM field (not shown)
2. Navigate to vSphere vCenter Plug-in > VirtualCenter > Datacenters > RegionA01 > vm
3. Select base-linux-cli
4. Click on Select button
Enter inputs

1. Enter **root** for the VM guest username
2. Enter **VMware1!** for the VM guest password
3. Click on **Not set** for Script to run. This allows to select the Script Configuration previously created
4. **Double click** on **Fix Network**

Enter variables

1. Click on **Not set** link for Variables to replace in the script field
2. Click on **Insert value**
   (A resize of the new window might be required to see all the fields properly)
3. Enter `{{ipAddress}}` for `stringToReplace` field
4. Enter `192.168.120.101` for `replacingString`
5. Click the **Define** button

**Accept variable list**

![Image of Accept variable list window]

The window might need to be moved up to show the Accept button

1. Click on **Accept** button
Submit workflow

Verify all inputs are properly filled

1. Click on **Submit** button
Observe workflow run

Restart putty session
Switch to putty terminal (click on Putty in the taskbar, not show in the screenshot)

1. **Right-Click** on the title bar
2. Click on **Restart Session**

### Session is re-established

![Console output]

The session is successfully re-established, proving the network configuration has been restored.

### Run scripts on multiple machines

After running one script on one virtual machine, let see how to do the same on multiple machine at the same time. It can be very useful to perform IP change in case of DR or datacenter migration.
Create a New Workflow

1. Make sure the **Workflow** tab is selected
2. **Fold** the Guest Script Manager folder by clicking on the **arrow**.
3. Right Click on HOL > **1921-06-CMP > Module 5** workflow folder
4. Click on **New workflow**

**Name Workflow**

1. Enter the name of the workflow: **Update Network on multiple machines**
2. Click **OK**
Add "Start Workflows in parallel"

1. Select Schema tab
2. Enter "in parallel" in the search box
3. Drag and Drop the workflow **Start workflows in parallel** to the blue arrow
Configure input/outputs

1. Click on Setup... button,
   The window might need to be resized to see all the parameters
2. Select Value for both inputs
3. Select Local variable for the output
4. Click on Promote button

Make sure the workflow element in the design canvas is selected (not shown)
Add "Run script in VM guest"

1. Enter "run script in vm" in the search box
2. Drag and Drop the workflow **Run script in VM guest** before the existing workflow element
Setup inputs/outputs

Make sure the workflow element **Run script in VM guest** in the design canvas is selected (not shown)

1. Click the **Setup...** button
   The window might need to be resized to see all the parameters
2. Set the "Reset all bindings" to **Value**
3. Select **Input** for **vm** input
4. Select **Skip** for **scriptVariables** input
5. In the Output parameter section, Set the "Reset all bindings" to **Skip**
6. Click on **Promote** button
Delete workflow element

This workflow will be launched through the other workflow element "Start workflows in parallel" so it is not needed in this workflow. However it was helpful to use it in order to setup all the required attributes and inputs. By this way, it is more reliable than setting each attributes and input manually one by one.

1. Select the workflow element **Run script in VM guest**
2. Click on the **red cross** button

Configure workflow attribute

1. Select **General** tab
2. **Scroll down** to show attribute list
3. Click on **Not set** (not shown in the screenshot) for **wf** attribute and enter "**run script in vm**" in the search box
4. **Double-Click** on **Run script in VM guest** workflow
1. Enter **root** for **vmUsername**
2. Enter **VMware1!** for **vmPassword**
3. Click on **Not set** (not shown in the screenshot) for **scriptConfigurationResource** attribute and enter "**Fix**" in the search box
4. Double-Click on the resource element **Fix Network**
   (Make sure Fix Network appears in the attribute value)
Configure vm input as an array

1. Select **Inputs** tab
2. Click on the type VC:VirtualMachine for vm input
3. Select **Array Of**
4. Click on **Accept** button
Configure ipAddresses input parameter

1. Click on **Add parameter** button
2. Click on **string** type for the new input parameter
3. Select **Array Of**
4. Select **string** (if not already selected)
5. Click on **Accept** button
Rename ipAddresses input parameter

1. Click on arg_in_0
2. Enter ipAddresses
3. Click on Ok button
Add scriptable task

1. Select **Schema** tab
2. Select **Generic** tab
3. Drag and drop **Scriptable task** to the blue arrow before the workflow element

*Start workflows in parallel*
1. Select **IN** tab
2. Click on **Bind to workflow parameter/attribute** button
3. **Check** the checkbox for the following parameter/attribute
   - vm
   - ipAddresses
   - vmUsername
   - vmPassword
   - scriptConfigurationResource
4. Click on **Select** button
Configure OUT

1. Select **OUT** tab
2. Click on **Bind to workflow parameter/attribute** button
3. **Check** the checkbox for the following parameter/attribute **parameters**
4. Click on **Select** button
Configure Scripting

1. Click on **Scripting** tab
2. **Enter** the following script

```javascript
parameters = [];
for (var i = 0; i < vm.length; i++){
    var wfParams = new Properties();
    wfParams.put("vm", vm[i]);
    wfParams.put("vmUsername", vmUsername);
    wfParams.put("vmPassword", vmPassword);
    wfParams.put("scriptConfigurationResource", scriptConfigurationResource);
    wfParams.put("scriptVariables", {
        "stringToReplace": "{{ipAddress}}",
        "replacingString": ipAddress[i]
    });
    parameters.push(wfParams);
}
```

3. Click on **Save and close** button

This script creates a parameter array. Each element of the array is the list of inputs required for the workflow "Run a script in VM guest".

The variables **vmUsername**, **vmPassword**, **scriptConfigurationResource** are always the same for all workflow runs.
The variables \texttt{vm} (VC:VirtualMachine object) and \texttt{scriptVariables} differ from one workflow run to another.

Finally, note the specific formatting for the \texttt{scriptVariables} variable. It is because this variable has a type which is an Array of a composite type.

**Run the workflow**

1. Select \textbf{Update Network on multiple machines}
2. Click on \textbf{Start workflow...} button
Select machine list

1. Click on **Not set** for vm field
2. Click on **Insert value** for new value
Select linux VM

1. Expand **vSphere vCenter Plug-in > VirtualCenter > Datacenters > RegionA01 > vm**
2. Select **base-linux-cli**
3. Click on **Add** button
4. Click on **Select** button
5. Click on **Accept** button (not shown, the window might need to be moved to show the button)

Note that step 2 and 3 could be repeated to reconfigure networking on multiple machines at once.
Select IP Addresses

1. Click on **Not set** for ipAddresses
2. Enter the new IP address for the machine in the **New value** field: **192.168.120.202**
3. Click on **Insert value** button
4. Click on **Accept** button (not shown, the window might need to be moved to show the button)

Note that the number of IP addresses in the list must match the number of machine selected at the previous step.
Submit workflow

Make sure the number of VM provided matches the number of IP addresses. No check are performed in the workflow to verify this condition, in a production environment it is strongly recommended to perform such a verification.

1. Click on **Submit** button

Open Command Prompt terminal

1. Click the **Command Prompt** icon in the task bar
Ping the new IP

1. Enter the command and Press Return

```
ping 192.168.120.202
```

The new IP address is successfully responding.

2. Enter the command and Press Return

```
ping 192.168.120.101
```

The old IP address is timing out.
Clean Up the Environment

Cleanup is needed before proceeding with the lab

Login to vCenter Web UI

Switch to Chrome and If not already logged in:

1. Click on RegionA bookmark folder
2. Click on HTML5 Client bookmark
3. Enter administrator@vsphere.local as user name
4. Enter VMware1! as password
5. Click on Login button
Select linux machine

If the machine is not already selected.

1. Enter linux in the search bar
2. Click on link base-linux-cli in the result list

Stop linux machine

1. Click on ACTIONS
2. Click on Power
3. Click on Shut Down Guest OS
4. If a popup appears, Click the YES button
Conclusion

**vRealize Orchestrator is the perfect tool to get the most out of the vSphere Guest Operation capabilities. With its ability to easily perform parallel tasks, it is now easy and safe to perform seamless maintenance operations in any guest operating systems even without network connectivity or third-party agents.**

You've finished Module 5

Congratulations on completing Module 5.

If you are looking for additional information on Guest Script Manager try one of these:

- Click on this [link](#)
- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
- **Module 6 - Error Handling and Troubleshooting** (45 min) (Advanced) Discover the various ways to add error or exception handling to ease in
troubleshooting and more resilient workflows. Learn about and create exception binding to your workflows.

**How to End Lab**

To end your lab click on the **END** button.
Module 6 - Error Handling and Troubleshooting (45 min)
Introduction

Error handling and logging are very important concepts in vRealize Orchestrator for the creation of reliable workflows, to assist with workflow troubleshooting, and ultimately to make infrastructure more robust.

In this module, error handling, troubleshooting, and logging are covered. This lesson highlights how easy it is to catch errors, execute remediation actions, and finally to automate everything in a reliable fashion using vRealize Orchestrator.

Launch vRealize Orchestrator Client

Double click on the vRealize Orchestrator Client icon on the desktop.
Log In to vRealize Orchestrator

Use the following credentials

1. Host name: **vra-01a.corp.local:443**
2. User name: **administrator@vsphere.local**
3. Password: **VMware1!**
4. Click **Login**

Select the Design View

1. Click the dropdown menu at the top of the client, and select the **Design** View
Error handling and why is it important?

Exception handling catches any errors that occur when a schema element runs. Exception handling defines how the schema element behaves when the error occurs.

All elements in a workflow, except for decisions and start and end elements, contain a specific output parameter type that serves only for handling exceptions. If an element encounters an error during its run, it can send an error signal to an exception handler. Exception handlers catch the error and react according to the errors they receive. If the exception handlers you define cannot handle a certain error, you can bind an element's exception output parameter to an Exception element, which ends the workflow run in the failed state.

This is important because when your workflows do fail, exception handling allows them to fail as gracefully as possible and it provides meaningful error messages.

Open Chrome Browser from Windows Quick Launch Task Bar

Let’s open vRealize Automation and confirm Log Insight integration configuration.

1. Click on the Chrome Icon on the Windows Quick Launch Task Bar.
Log In to vRA Management

1. Click the **vra-01a mgmt** bookmark in HOL Admin bookmark folder
2. Enter **root** as user name
3. Enter **VMware1!** as password

Select Logs tab

1. Make sure **vRA Settings** tab is selected
2. Select **Logs** tab

**Update configuration**

1. Scroll down to find **Log Insight Agent Configuration** section
2. Enter **vrli-01a.corp.local** as host
3. Enter **9000** as Port
Save Configuration

1. Scroll up to the top
2. Click the **Save Settings** button
3. Click the **X** to close the browser.
4. Click the **Leave** button in the popup (not shown)
Adding Error Handling to workflows

To learn more about how to handle errors in vRealize Orchestrator workflows, let's build a basic workflow and add error handling to it.

Create a New Workflow

1. Right click the HOL > 1921-06-CMP > Module 6 workflow folder
2. Select New workflow

Name the Workflow

1. Type My Workflow Error Handling for the workflow name
2. Click the Ok button
Add the Create a simple virtual machine Workflow

1. Select the **Schema** tab
2. Select **All Workflows** tab
3. Navigate to Library > vCenter > Virtual Machine management > Basic > Create simple virtual machine
4. Drag and Drop the workflow **Create simple virtual machine** between the Start and End workflow elements
Add the Create a simple virtual machine Workflow

1. Click **Setup**
   By clicking setup, this is allowing vRealize Orchestrator to bind the inputs/outputs from Create simple virtual machine to the parent workflow.
   Look over the input and output parameters that are being added to the workflow.
2. Click **Promote**
Add the System Error Element to a Workflow

1. Select the **Log** tab
2. Drag and Drop the **System Error** Element on to the Create simple virtual machine workflow element.

By dragging the System Error Element onto the workflow, you are linking it to the workflow. Notice how a red dotted line is created from the workflow to the system error. Also, the System Error element is connected to a separate End workflow element. This shows that if the the workflow errors out, it will be logged and the workflow will end.
Setup the System Error Element inputs

1. Make sure the **System error** element is selected
2. Click the **Setup**... button
Setup the System Error Element inputs

1. Select the Value radio button for the **text** parameter
2. Click the **Input value** link for the text parameter
3. Type the following in the Enter Text field:

   The workflow has failed, please contact the support.

4. Click the **Ok** button
5. Click the **Promote** button

The text parameter allows you to supply the end user with more information when the workflow fails.
Move Parameters to Attributes

1. Select the **Inputs** tab
2. Select the **vmGuestOs** row (do not click on the parameter name itself, or you will open a new window to rename the parameter)
3. Click the **Move as attribute** icon
4. Repeat the above tasks for these additional parameters: **vmFolder, vmResourcePool, vmHost, vmNetwork, and vmDatastore**

The reason these parameters are moved to attributes is that they can be configured with preset values. The end users are not burdened with entering these values at provisioning time.
Configure vmGuestOS Attribute

1. Select the **General** tab
2. **Scroll** to the bottom
3. Click **Not Set** to set the value for **vmGuestOS**

Configure vmGuestOS Attribute

1. Enter **centos** in the Search field and press **Enter**
2. Double click **centos64Guest** to select it
Configuring vmFolder Attribute

1. Click **Not Set** to set the value for **vmFolder** (not shown)
3. Click the **Select** button
Configuring vmResourcePool Attribute

1. Click Not Set to set the value for vmResourcePool (not shown)
2. Navigate to vSphere vCenter Plug-in > https://vcsa-01a.corp.local:443/sdk > Datacenters > RegionA01 > host > RegionA01-COMP01 > Resources
3. Click the Select button
Configuring vmHost Attribute

1. Click **Not Set** to set the value for **vmHost** (not shown)
3. Click the **Select** button
Configuring vmNetwork Attribute

1. Click **Not Set** to set the value for **vmNetwork** (not shown)
2. Navigate to **vSphere vCenter Plug-in** > `https://vcsa-01a.corp.local:443/sdk` > **Datacenters** > **RegionA01** > **network** > **none**
   (since no connectivity is required for this exercise)
3. Click the **Select** button
Configuring `vmDatastore` Attribute

1. Click **Not Set** to set the value for `vmDatastore` (not shown)
2. Navigate to *vSphere vCenter Plug-in > https://vcsa-01a.corp.local:443/sdk* > *Datacenters > RegionA01 > datastore > RegionA01-ISCSI01-COMP01*
3. Click the **Select** button

Run the workflow
Let's do a test run of the workflow to see the effects of adding the System Error element.

1. Select the **Schema** Tab
2. Click the **Run Workflow** icon

### Provide the Inputs for the workflow

![Workflow Input Form]

1. Type **Finance** for the **Virtual machine name**
2. Type **5** for **Size of virtual disk in GB**
3. Type **512** for **Virtual machine's memory size in MB**
4. Type **.5** for **Number of virtual processors**

Note that .5 is used for the CPU count (which is invalid) to test the results of a failed workflow run with the added System Error element.
5. Select **Yes** for *Make disk thin-provisioned*
6. Click the **Submit** button

**Review the logs**

![Workflow Tokens, Events, Permissions](image)

1. Make sure the **Logs** tab is selected (on the right side)

Note that the workflow failed, and a few things were logged:

- The VMware provided error
- The additional user friendly text you provided

This is great, but now let's change an incorrect CPU count to 1 CPU if the user provided an invalid value, and proceed with the workflow.
This is a good example of error handling, and you will be building this into your workflow next.

2. **Click** to cross to close the right pane.
Add a Decision Element

1. Select **Generic** tab
2. Drag and drop the **Decision** Element on to the line between the Create simple virtual machine workflow and the System Error Element

By dragging the **Decision** Element onto the line it is linked to the other elements. The red dotted line is created from the workflow to a End Workflow element. The Decision Element is connected to the System Error element with a green line.
Cleanup the paths

If the Decision Element is false it needs to be connected to the System Error, so let's do some cleanup.

1. Right click the red dotted line between the Decision Element and a End Workflow Element
2. Select Delete

This will delete the red dotted line and the End Workflow Element that is connected to it.
Cleanup the paths

If the Decision Element is false it needs to be connected to the System Error, so let's do some cleanup.

1. Right click the **green line** between the **Decision** Element and the **System Error** Element
2. Select **Delete**
Wire the Decision Element

Finally you now can connect the red dotted line, or false to the System Error Element

1. Change the zoom to 75%
2. Hover over the Decision Element and drag the red arrow to the System Error Element
Configure the Decision Element

1. Click on the **Decision** Element to open the edit window in the lower pane
2. Select **Decision** tab
3. Click the **Not set (NULL)** link

Now you will want to give the decision some criteria to look for.

1. Click on the **Decision** Element to open the edit window in the lower pane
2. Select **Decision** tab
3. Click the **Not set (NULL)** link
Select decision criteria

1. Select the **errorCode** attribute
2. Click the **Select** button

This attribute is used by the decision element to make its decision.

Select decision criteria

1. Click the drop down and select **contains**
2. Enter **HOL-1921-06-CMP**

Now that the decision element has a string value to check, the condition can be configured:

1. Click the drop down and select **contains**
2. Enter **numCPUs** in the value field

The decision is set to look for the errorCode parameter that includes the string value "numCPUs".

**Move a parameter to an attribute**

1. Select the **Inputs** tab
2. Select the **vm Nb Of Cpus** parameter row
3. Click the **Move as attribute** icon

Input parameters are read-only and cannot be changed once an input value is given, attributes are read/write. The value of this parameter may need to be updated based on the outcome of the decision element.
Add an Input Parameter

Let's add an input parameter to have the end user provide the CPU count.

1. Click the **Add a parameter** icon
2. Click on **arg_in_0** to provide a name (not shown)
3. Type **vmCPU** for the Attribute name
4. Click the **Ok** button

**Change vmCPU description**

1. Double click on the Description field of the vmCPU parameter and type **Virtual Machine CPU count**
2. Click the **string** link for vmCPU parameter
Change vmCPU type

Parameter type needs to be changed to number because this value will be passed to `vmNbOfCpus` attribute, which is also of type number and type must match in vRO.

1. Select `number`
2. Click the `Accept` button

Add a Scriptable Task

1. Select the Schema tab
2. Select the Generic tab
3. Drag and Drop the Scriptable task Element on to the line between the Start workflow element and the Create simple virtual machine workflow element
This Scriptable task is used to pass the value from the vmCPU input parameter to the vmNbOfCpus attribute (used by the subsequent workflow element).

**Name the Scriptable Task**

1. Double click on the **Scriptable task**
2. Enter **Input CPU count** for the Name
   Press Return key

**Add an IN Parameter**
Add an IN Parameter

1. Select the IN tab
2. Click the **Bind a workflow parameter** button
3. Select the **vmCPU** parameter
4. Click the **Select** button

Add an OUT Parameter

1. Select the **OUT** tab
2. Click the **Bind a workflow parameter** button
3. Select the **vmNbOfCpus** parameter
4. Click the **Select** button

Add the script

1. **Scripting**
2. `vmNbOfCpus = vmCPU;`
1. Select the **Scripting** tab
2. Type the following in the scripting window

   ```
   vmNbOfCpus = vmCPU;
   ```

   This script will convert the `vmNbOfCpus` attribute to the value of the input parameter `vmCPU`.

**Add a Scriptable Task**

Let's add an additional Scriptable task to change the value of the `vmNbOfCpus` attribute to 1 when there is a failure.

1. Select **Generic** tab
2. Drag and Drop the **Scriptable task** Element on to the decision element

   It will be added to the schema with a green line that comes from the decision element.

This links the scriptable task to the decision element. When the criteria is met in the decision element, the workflow continues on to the scriptable task.
Name the Scriptable Task

The scriptable task is going to be renamed to make it more meaningful in the workflow.

1. Double click on the **Scriptable task**
2. Enter **Change CPU Count to 1** for the Name
3. Press Return key

Add an OUT Parameter

1. Make sure the scriptable task **Change CPU Count to 1** is selected (not shown)
2. Select the **OUT** tab
3. Click the **Bind a workflow parameter** button
4. Select the **vmNbOfCpus** parameter
5. Click the **Select** button

**Add the script**

1. Select the **Scripting** tab
2. Type the following in the scripting window

```
vmNbOfCpus = 1;
```

This script sets the value of vmNbOfCpus to 1.

**Link the Scriptable Task**

Link the "Change CPU Count to 1" Scriptable task to the "Create simple virtual machine workflow." This way, if there is an error due to an invalid CPU number, the value is changed to 1.
1. Hover over the scriptable task and drag the **blue arrow** from **Change CPU Count to 1** to the **Create a simple virtual machine** workflow

**Save and Close**

The schema should now look like that.

1. Click the **Save and close** button
Test the Workflow

1. Browse to **HOL > 1921-06-CMP > Module 6 > My Workflow Error Handling**, if it is not already selected
2. **Expand** the workflow to see the workflow runs
3. **Right click** on the last workflow run
4. Click the **Run again...** button
Provide Inputs

Most of the values should be pre-filled (from the previous run)

1. Enter \( .5 \) for the Virtual machine CPU count
2. Click the Submit button

The values are the same as at the beginning of the exercise. But this time there is some error handling to correct an invalid CPU count value. The workflow should proceed as expected and create the virtual machine.
Check the Logs

1. Make sure the last workflow run is selected
2. Make sure the **Schema** tab is selected
3. Click the **Logs** tab
4. Review the logs

In the logs, an error was thrown because there was an invalid CPU count, but the error handling task caught it and the workflow completed successfully.

Open Chrome Browser from Windows Quick Launch Task Bar

Let's check the virtual machine creation in the vCenter.

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.
Login into the vSphere Web Client

1. Click on the HTML5 Client bookmark in the RegionA folder
2. Check the box for Use Windows session authentication

This will log you on with the administrator@corp.local credentials

3. Click the Login button (This may take a minute)
Verify a Virtual Machine was Created

1. Locate the virtual machine **Finance** in the inventory
2. Verify the **CPU** entry showing 1 CPU allocated to the FinanceDB virtual machine
Delete the Virtual Machine

Since the exercise is done, the VM is deleted to clean the environment.

1. Select **Finance** virtual machine (if it is not already the case)
2. Click on **Actions** icon to show the dropdown list
3. Scroll down to see the full list of actions
4. Click on **Delete from Disk**
5. Click on **Yes** in the **Confirm Delete** dialog box (not shown)

Close the Chrome Browser

1. Click the **X** in the top right hand corner of the browser to close it
Basic Workflow Troubleshooting

vRealize Orchestrator provides various ways to investigate workflow failures. It can be done from the vRealize Orchestrator Client, or directly from the logs. In this lesson, troubleshooting from the Client is covered as well as very basic usage of Log Insight with the vRealize Orchestrator management pack.

Launch "workflow failure 1"

1. Make sure Design mode is selected
2. Make sure Workflows tab is selected
3. Expand HOL > 1921-06-CMP > Module 6
4. Select Workflow failure 1
5. Click Start workflow... icon
Observe the Failure

1. The workflow token from the failed run can be seen by the red x icon
2. Select Variables tab
3. Look at the Exception message

This is the first clue to understand what happened. Most of the time this message should be clear enough to have an understanding of what caused the workflow failure.
Find the Element that Failed

1. Make sure the **Schema** tab is selected
2. **Expand** the upper panel by **drag and dropping** the separator bar
3. Look at the **highlighted** item

The workflow element highlighted in red is where the exception happened.
Look at the Code

1. Double-click on the **Scriptable task** that was previously highlighted in red

Look at the code (continued)

1. The workflow **Workflow failure 1** should be now selected
2. Select **Scripting** tab
3. Look at the code, the first line is just a log, so the variable \textit{sizes} must be involved in the failure.

**Look at the Variable Value at Failure Time**

1. Click back on the \textit{workflow instance} that failed
2. Lower pane should be expanded by drag and dropping the separator bar
3. Select \textit{Variables} tab
4. Look at the variable \textit{sizes}

This variable, which is an array, has not been initialized (Not Set in value). It explains why the exception message says "Cannot call method "push" of null" - in this case, null is the array.
1. Select **Logs** tab
2. Select **Debug** from the dropdown
3. Logs before the exception can help narrow down the exception and the root cause. That’s why logging is very important. In the next exercise, we will review logging in more detail.
4. The exception message is written to the error log. It also helps to understand where this happened: In the workflow "Workflow failure 1", in the item 1, names "Scriptable task", at line 1.
5. It shows the full variable stack, so it is easy to see what the value of a variable was at the time of the exception.
Launch Workflow failure 2

1. Right-click on **Workflow failure 2**
2. Click on **Start workflow...**
Observe the failure

1. Select **Schema**
2. The workflow element where the exception occurred is highlighted
3. In this case, the workflow that failed is not the root workflow, so it is possible to see the workflow stack
Switch to Root Workflow

1. Click on Workflow failure 2 button, by this way, the schema and the variables in the lower panel will reflect this workflow.
2. Select the Variables tab
3. The workflow element in Workflow failure 2 which failed is highlighted
4. The variables are the variables from Workflow failure 2

This can be useful to track down the propagation of a bad value.
Switch Back to Child Workflow

1. Click on **Workflow failure 1** button
2. The variables are the variables from **Workflow failure 1**

This helps to understand how variables have flown from the root workflow to the child workflow.
Look at the Logs

1. Expand the lower pane by dragging and dropping the separator bar
2. Select **Logs** tab
3. This shows the variables for root workflow (Workflow failure 2)
4. This shows the variables for child workflow (Workflow failure 1), notice how this *Workflow failure 1* log entry is indented with *|*. This helps identify a failure in a called workflow.

In case of more workflows, all variables of all workflows are shown.

Open Chrome Browser from Windows Quick Launch Task Bar
Let's now take a look at Log Insight.

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar.

**Log In to Log Insight**

1. Click on **vRealize Log Insight** link
2. Enter credentials
   - Username: admin
   - Password: VMware1!
3. Click on **Login** button
Select vRO Dashboard

1. Click on **Dashboards** tab
2. Scroll down to find **VMware - Orchestrator - 7.0.1+** content pack
3. Expand **VMware - Orchestrator - 7.0.1+** content pack
4. Click on **Workflows failures** dashboard
5. Select **Last 24 hours of data** from the dropdown
1. Scroll down the dashboard
2. Look at the **FAILED workflows grouped by WORKFLOW NAME** widget
3. Close the browser (not shown)
Logging in Workflows

Despite multiple methods for workflow troubleshooting described in the previous exercise, logging is still very important in order to understand errors, and also to audit a workflow result.

vRealize has different way for logging, they are described in this exercise.

Create a New Workflow

1. Right click the HOL > 1921-06-CMP > Module 6 workflow folder
2. Select New workflow

Name New Workflow
1. Enter "Logging" as workflow name
2. Click OK

Add Scriptable Task

1. Select Schema
2. Select Generic category
3. Drag and Drop Scriptable task to the blue arrow
Undock Lower Panel

1. Select **Scriptable task** workflow element
2. Click on little window button
Add Script

1. Select **Scripting**
2. Enter the following code:

```javascript
System.log("INFO logging");
System.warn("WARN logging");
System.error("ERROR logging");
System.debug("DEBUG logging");

Server.log("INFO event");
Server.warn("WARN event");
Server.error("ERROR event");
```

3. Click on the **X** to close the window
Save and Close

1. Click on **Save and close** button (lower right)

Run the Workflow

1. Make sure the workflow **Logging** (just created) is selected
2. Click the **Start workflow...** button
Look at the Logs

1. Select Schema tab
2. Select Logs tab
3. Make sure the dropdown has Info selected
4. Look at the message

[Image of a window with tabs and dropdown menu]

Messages:
- [2018-07-24 02:53:34.898] [I] INFO logging
- [2018-07-24 02:53:34.894] [V] WARN logging
- [2018-07-24 02:53:34.898] [E] ERROR logging
Changing Log Level View

1. Click on the dropdown and select **Debug**

More log messages should appear!

**Message Details**

Each message in the log consists of the following items:

1. Time of the log execution, localized using current computer time
2. Flag indicating type of logs
   - I = INFO
   - W = WARNING
   - E = ERROR
   - D = DEBUG
3. Text decoration change depending of type of logs
   - black = INFO
   - black bold = WARNING
   - red bold = ERROR
   - black italic = DEBUG
Remember that these log messages have been written by this part of the code:

```java
System.log("INFO logging");
System.warn("WARN logging");
System.error("ERROR logging");
System.debug("DEBUG logging");
```

These logs can also be found in Server.log file in the vRealize Orchestrator instance. By default, DEBUG logs are not logged in this file.

In a vRealize Orchestrator cluster, logs are visible only when connected to the vRealize Orchestrator instance where the workflow instance actually ran.

Log Insight should be used to have a consolidated view at all times.

### Look at the Events

1. Select **Events** tab
2. Look at the message

### Events Details

Each event consists of the following information:
1. Message (what was written in the code)
2. Type of event
3. Time of event

Events are stored in database and therefore must be used carefully.

**Close vRealize Orchestrator**

1. Click the red X in the upper right corner to close the vRealize Orchestrator Client window
2. Click **Exit** when prompted
Conclusion

In addition to providing an easy way to develop workflows, third party integrations, and lifecycle extensibility to vRealize Automation, with vRealize Orchestrator it is easy to troubleshoot workflow failures and to include failure path in the workflow itself. This make every process automated through vRealize Orchestrator much more reliable.

You've finished Module 6

Congratulations on completing Module 6.

If you are looking for additional information on Debugging workflows, try one of these:

- Or use your smart device to scan the QRC Code.

Proceed to any module below which interests you most.

- **Module 1 - Creating Advanced vRealize Orchestrator Workflows** (45 min) (Advanced) Test your Orchestrator skills by in this module where you will create an advanced workflow. Learn about applying best practices to workflow development.
- **Module 2 - Integrating with REST APIs** (45 min) (Advanced) Learn how to unleash the power of vRealize Orchestrator by integrating it with REST APIs provided by external systems.
- **Module 3 - Launching Workflows from Outside vRealize Orchestrator** (45 min) (Advanced) Learn how you can call into vRealize Orchestrator to invoke the workflows you've created from external systems.
- **Module 4 - Automating vRealize Automation with vRealize Orchestrator** (45 min) (Advanced) vRealize Automation is an extremely powerful automation platform - but what if you want to help automate the automation? Learn how to orchestrate common tasks in vRealize Automation using vRealize Orchestrator.
• **Module 5 - Running In-Guest Operations with vRealize Orchestrator** (45 min) (Advanced) One of the most commonly used capabilities of vRealize Orchestrator is to execute tasks within a newly provisioned (or even previously existing) workload. In this module, you'll learn how to use this capability to take action within your provisioned objects.

### How to End Lab

To end your lab click on the **END** button.
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-1921-06-CMP

Version: 20181104-142657