knowledge15®
April 19–24, 2015 • Mandalay Bay • Las Vegas, NV

Lab Guide

Orchestration 301:
Design Your Own
Reusable Activities

Chris Maloy & Jake Burman

Default Login / Password:

admin / Knowledge15

itil / Knowledge15

employee / Knowledge15

This

Page

Intentionally

Left

Blank

Lab Overview

Automate everything and extend the usefulness of Orchestration by quickly creating reusable, drag-and-drop workflow activities. This session introduces you to the NEW Activity Designer, which gives you the ability to create reusable activities with little-to-no coding.

This is a game-changing feature for ServiceNow Orchestration and platform development. A series of labs show you how to create a new activity with just a few clicks. The labs cover basic to advanced activity creation and execution. Enhance service delivery, automate everything!

After these labs, you can work with Windows Triage activities that can be used to troubleshoot a system.

Use Case: Event Triage. Automatically triage the current state of a Windows system. If an Event Management system triggers an event, the first thing an operator typically does is see if the system is available (ping), then check the system properties to see if the CPU, Memory or Disk is high or out of space. Then they may investigate the services that are running.

- Lab 1: Create a Simple Workflow
- Lab 2: Create a PowerShell Activity to capture Windows CPU Utilization
- Lab 3: Create an Activity to Capture Windows Memory Utilization
- Lab 4: Create a Probe Activity (Shell Command) to Run Ping Against a Target Host, and Introduce Condition Results
- Lab 5: Create an Activity to Capture Windows Disk details, Introduce Advanced Scripting to Manipulate Outputs
- Lab 6: Create an Activity to Capture a List of Stopped Windows Services, Parse Data Using JSON
- Lab 7: Create a Triage Workflow to Audit a Target Windows Device for Current State Leveraging the Activities Created so Far
- (Optional) Lab 8: Create an Activity Leveraging a REST API (Google Maps integration)
- (Optional) Lab 9: Create a New Workflow That Leverages the Triage Workflow as a Subflow, then calls the Google Maps Activity to Calculate Dispatch Distance, and Creates an Incident with the Captured Details

®2015 ServiceNow, Inc. All rights reserved.

My First Workflow

This lab explains how to create a basic workflow, test its execution, and verify the Workflow logs. Later labs add to this workflow.

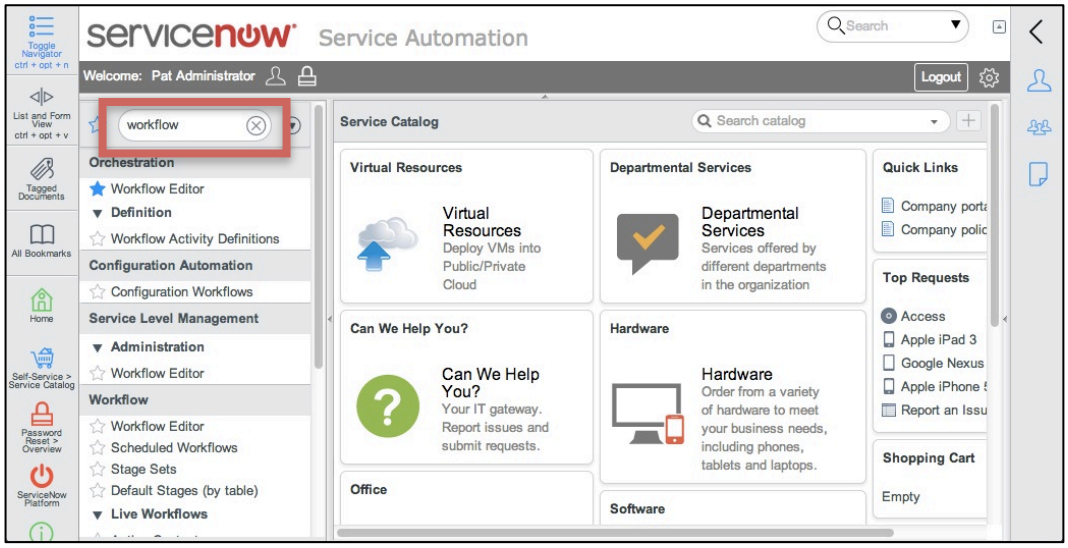
- Lab 1.1: Create a Workflow
- Lab 1.2: Test a Workflow
- Lab 1.3: Inspect the Workflow Log

Lab 1.0 Create a Workflow to Test New Activities

Lab 1.1: Create a Workflow

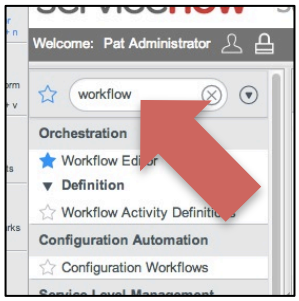
This lab explains how to create a simple workflow that adds a log entry to the workflow log.

1. Enter **workflow** in the **navigator** to locate the Orchestration Application.

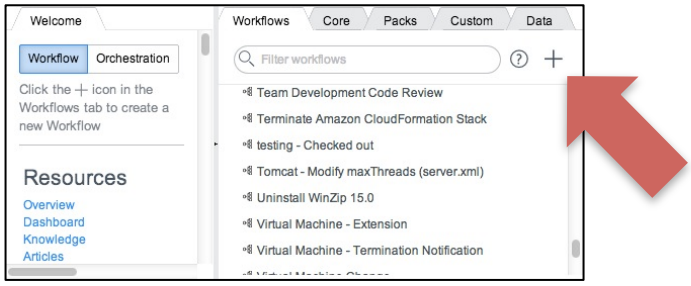


®2015 ServiceNow, Inc. All rights reserved.

2. Launch the **Workflow Editor** by clicking the menu option. A **new brows**er tab opens.



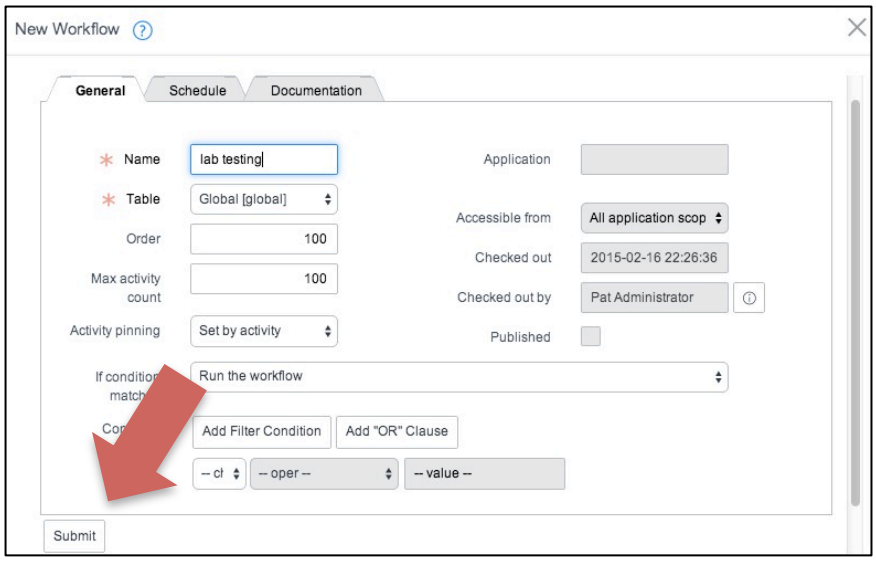
3. From the **Workflows** tab in the right panel, click the + plus to start a new Workflow. A new workflow popup appears.



®2015 ServiceNow, Inc. All rights reserved.

4. Enter the following **inputs**. Click **Submit**.

   Name: ***<your name>* lab testing**
   Table: **Global**

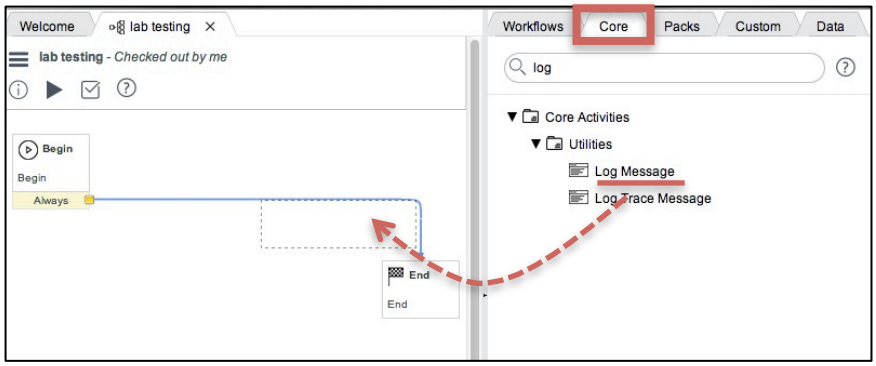


Note

Table defines which **workflows** can be executed from the various applications within the system.

5. Switch to the **Core** tab on the right panel and type **log** in the search field.

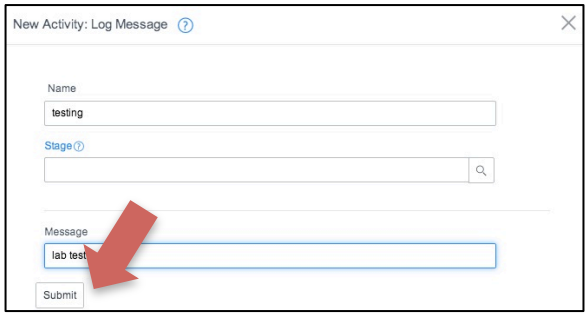
6. Drag and drop the **Core Activity > Utilities > Log Message** activity from the Core Activity panel to the connecting line between **Begin** and **End**. When the transition line appears blue, release the mouse (this automatically connects the transitions to the new activity).



©2015 ServiceNow, Inc. All rights reserved.

7. Enter the following **inputs** then click **Submit**.

   Name: **testing**
   Messag**e: lab test**





Tip

Click and drag the workflow steps to new locations on the canvas.

©2015 ServiceNow, Inc. All rights reserved.

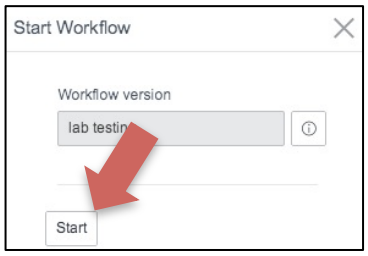
Lab 1.2: Test a Workflow

This lab explains how to test the new Workflow.

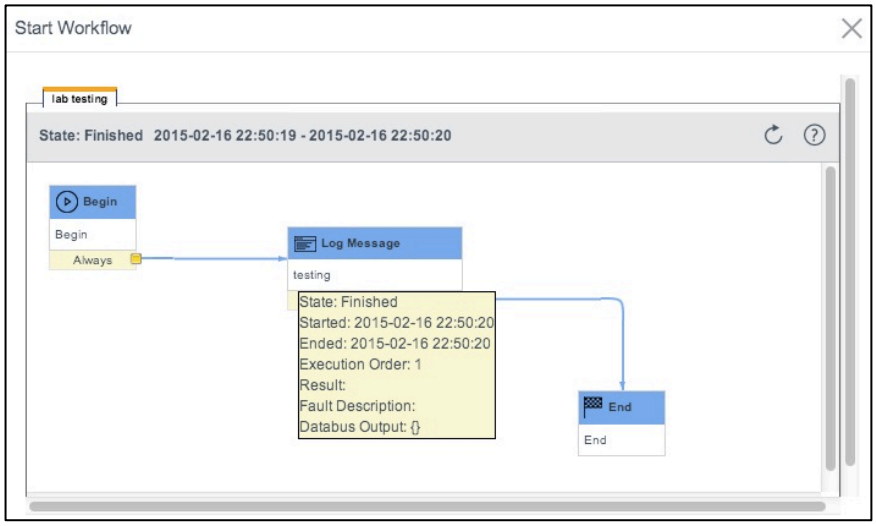
1. Click the **Play** icon on the top right of the Workflow canvas.



2. Click **Start** to activate the Workflow.



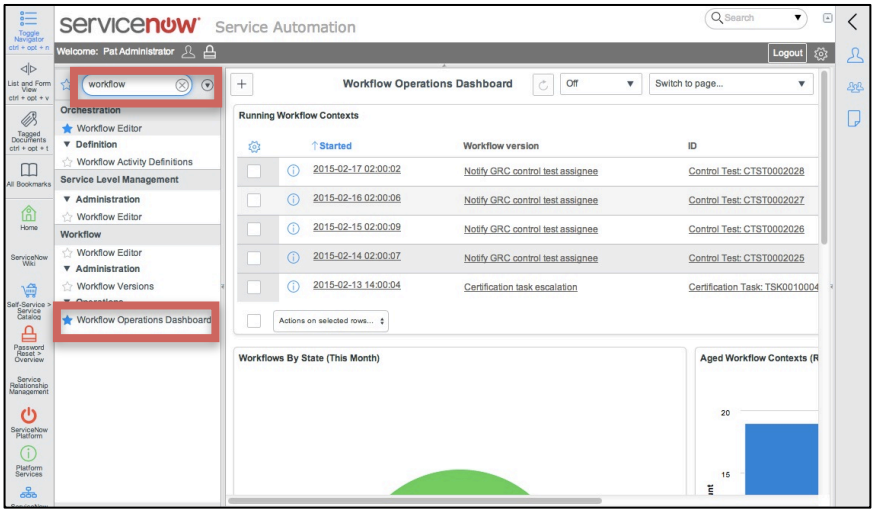
3. Mouse over the activities to inspect the start times, end times, and databus information. In later labs, additional workflow information and Orchestration results appear in this area. Click the ✕ on the top right to close the run window.



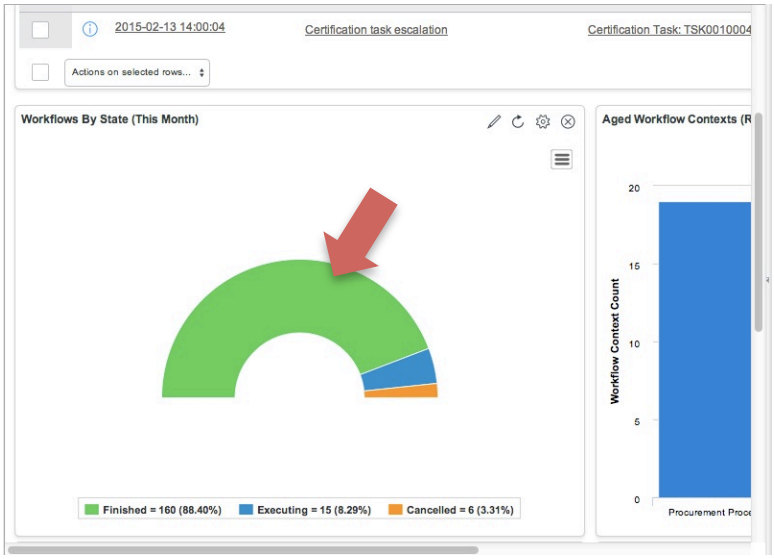
©2015 ServiceNow, Inc. All rights reserved.

Lab 1.3: Inspect the Workflow Log

This lab explains how to navigate to the Workflow logs, and view the last Workflow execution.

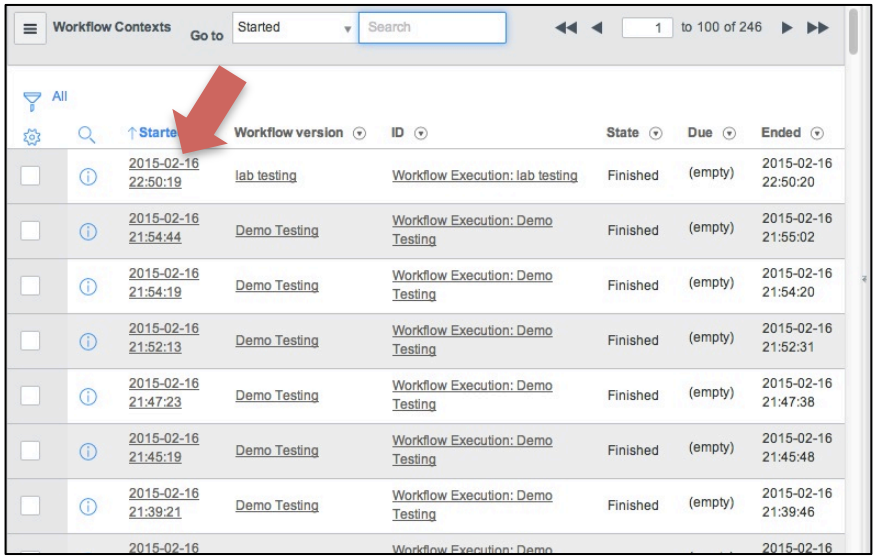


1. Navigate to **Workflow > Workflow Operators Dashboard**.

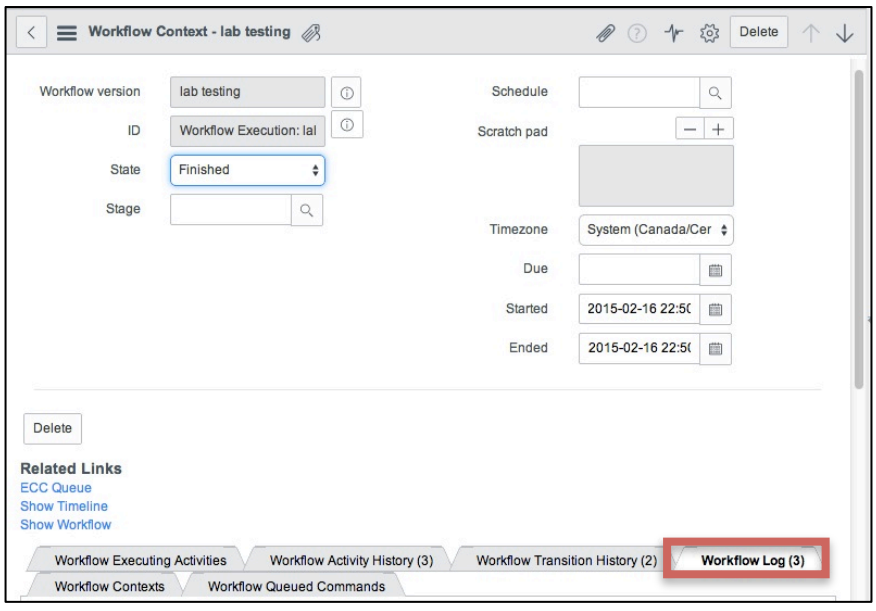


®2015 ServiceNow, Inc. All rights reserved.

2. Scroll to **Workflow By State (This Month)** and click the green **Finished** section of the graph. The Workflow Contexts list appears.

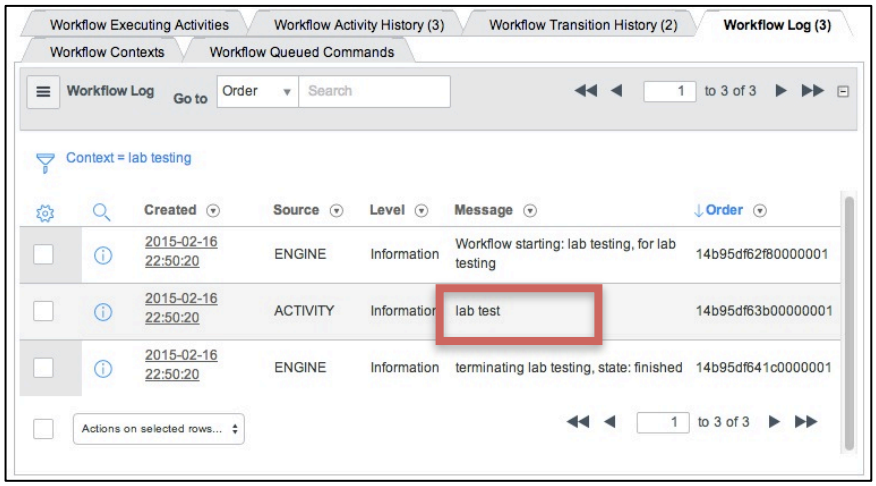


3. Sort by **Started** so that the most recent executions are at the top of the list window. Click the most recent entry, which shows the ID **Workflow Execution: lab testing**.



©2015 ServiceNow, Inc. All rights reserved.

4. Scroll down and click the **Workflow Log** tab. Locate the log message.



5. Locate the log message in the **Workflow Log**.

Later labs leverage the logging mechanism to perform testing and debugging of new orchestration activities.

Congratulations! You have successfully created a new Workflow. You have learned out to:

- Created a Workflow
- Executed a test of your new Workflow
- Inspected the Workflow execution log

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to create a new custom PowerShell activity to capture CPU Percent Used from a Windows host, using the Activity Designer. This lab covers the following:

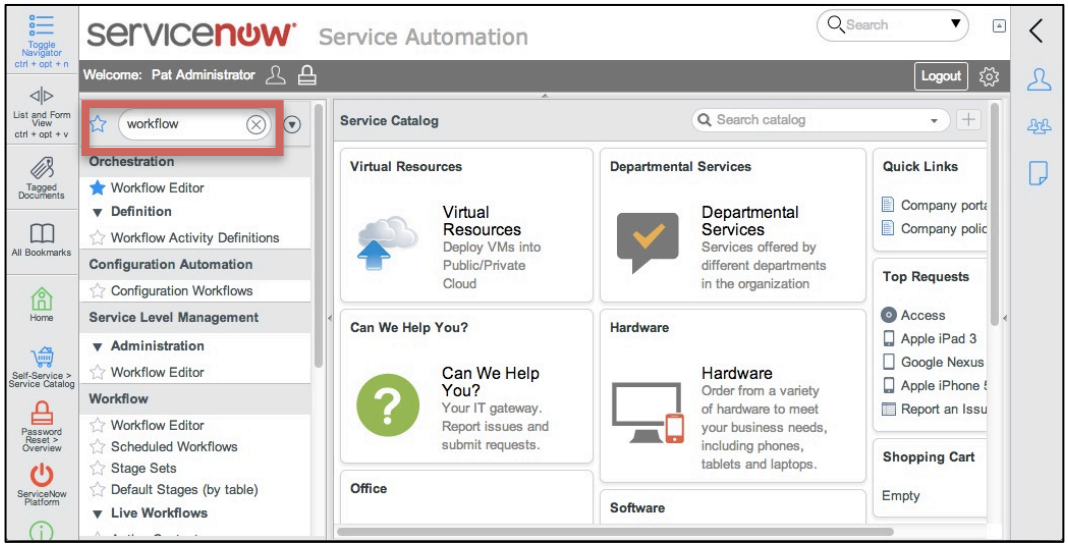
- Lab 2.1: Working with Inputs
- Lab 2.2: Working with Outputs

Lab 2.0 Create an Activity

Lab 2.1: Working with Inputs

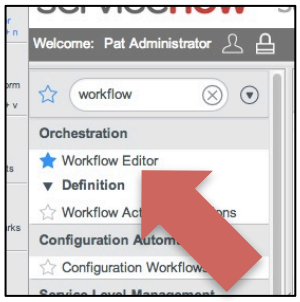
This lab explains how to create the new PowerShell activity with an input for the target host.

1. While logged into the lab instance, enter **workflow** in the **navigator** to locate the Orchestration Application.

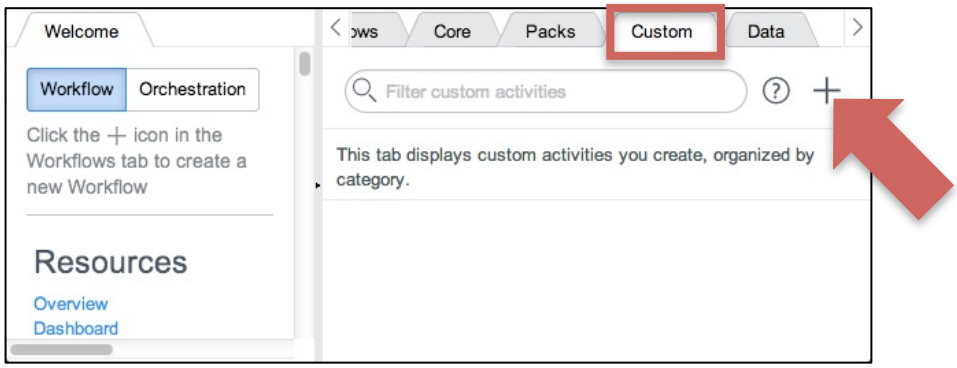


©2015 ServiceNow, Inc. All rights reserved.

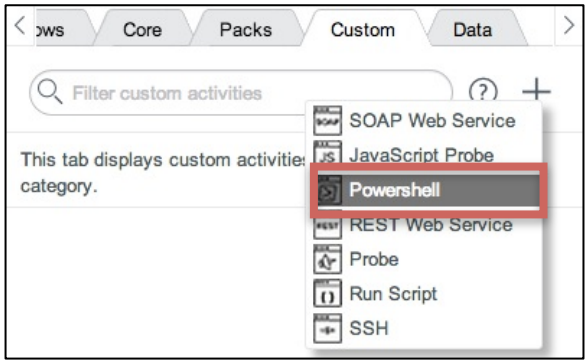
2. Launch the **Workflow Editor** by clicking on the menu option. A new browser tab opens.



3. Switch to the **Custom** tab in the right panel. Click the + plus to start a new Activity.



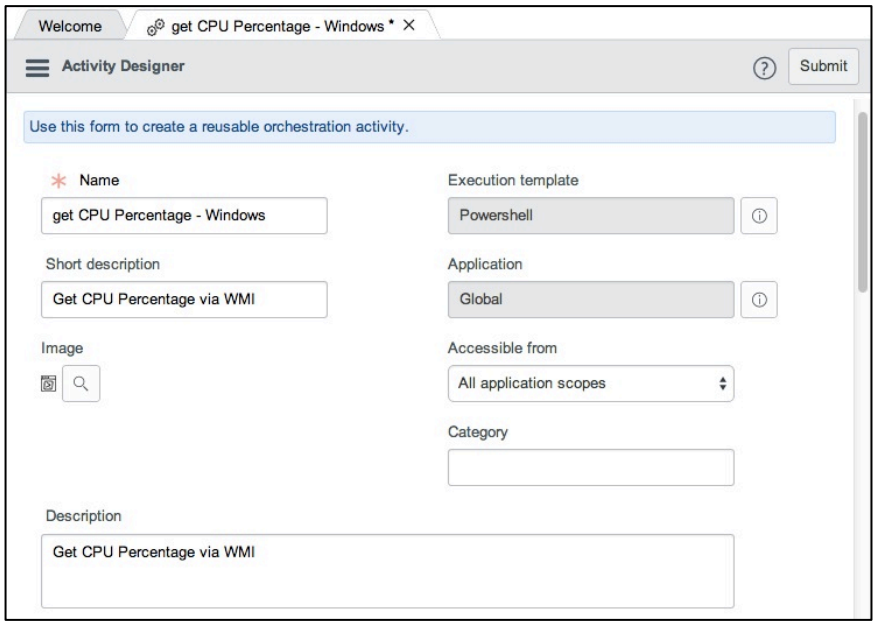
4. Select **Powershell** from the list. A new Activity panel appears.



©2015 ServiceNow, Inc. All rights reserved.

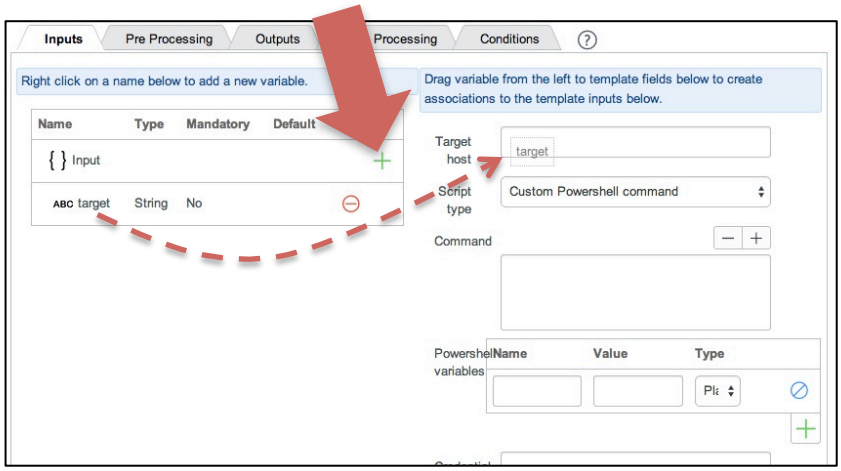
5. Enter the following **inputs**, and then scroll to input additional details.

   Name: **get CPU Percentage – Windows**
   Short description: **Get CPU Percentage via WMI**
   Description: **Get CPU Percentage via WMI**



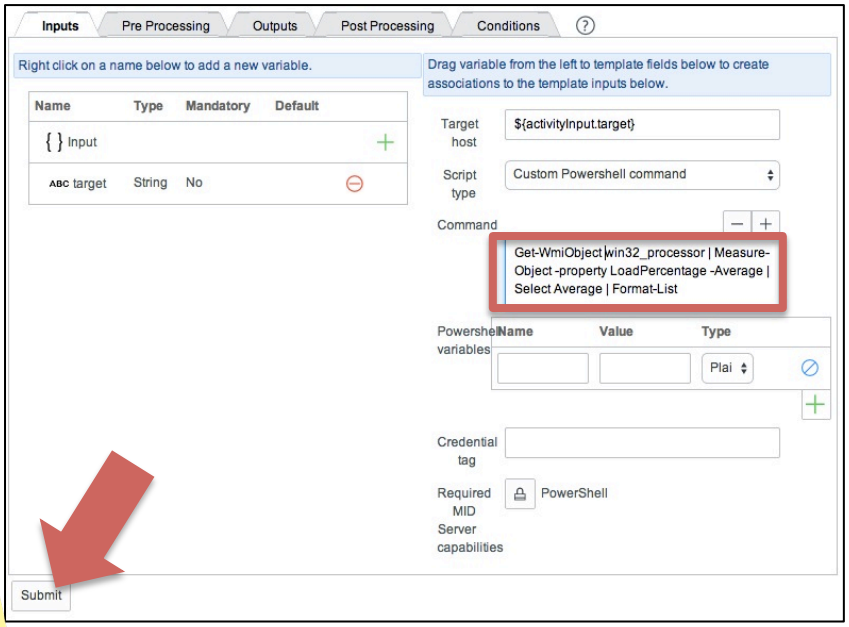
®2015 ServiceNow, Inc. All rights reserved.

6. Click the + **plus** on the **Inputs** tab to add a new input, enter ***target*** for the input name and press enter. Then, click and drag the new target input from the inputs on the left, to the **Target host** field on the right.



7. Enter the following command in the **Command** field. Click **Submit** to save the new Activity.

```
Get-WmiObject win32_processor | Measure-Object -property
LoadPercentage -Average | Select Average | Format-List
```



Tip

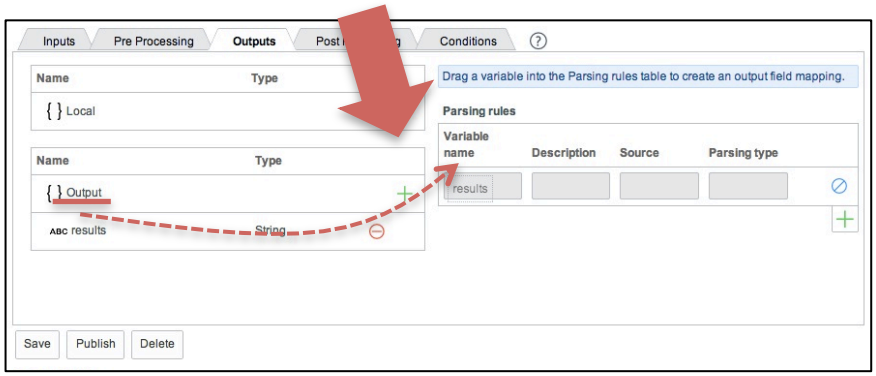
Inputs may be drag and dropped into other fields on the form.

®2015 ServiceNow, Inc. All rights reserved.

Lab 2.2: Working with Outputs

This lab explains how to add a result output to the get CPU Average activity

1. Click the **Outputs** tab. Click the + plus on the **Output** tab, to add a new Output, enter **result** for the output name and press **enter**. Click and drag the new result output from the outputs on the left, to the **Parsing rules** field on the right. A parsing rule pop-up window appears.



2. Choose **RegEx** from the **Parsing type** selection, the window expands to include a **RegEx** parsing condition.



©2015 ServiceNow, Inc. All rights reserved.

3. The left field Sample payload data, is a freeform text area for entering expected command results and testing the RegEx expression. The outcome of the get CPU Load PowerShell script is the text **`Average : %`**, where **%** is a numeric value.

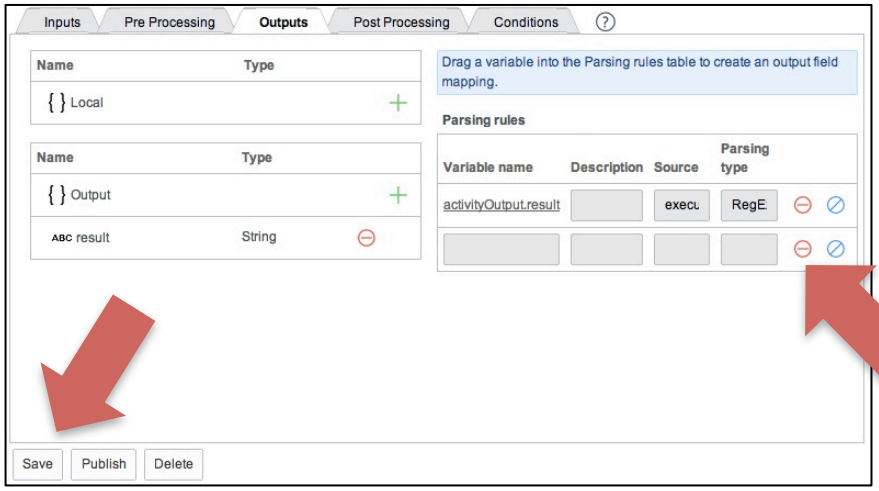
   Enter the following values, and then click **Test expression**. The numeric value you entered in the sample payload will appear in the Parsing result field. Then click **Submit**.
   Expression: **Average.*?: (.*)**
   Sample payload data: **Average : 1**
   RegEx breakdown

```
 Average.*?: (.*)
{        }_____________Find the word Average
 Average.*?: (.*)
       {   }_________Find all characters up to the next expression
 Average.*?: (.*)
         {   }_______Find the first colon :
 Average.*?: (.*)
           {  }       Find a space character
 Average.*?: (.*)
             {      }Match any character and include in a match group
```



®2015 ServiceNow, Inc. All rights reserved.

4. Click the red ⊖ minus to remove the extra parsing rule, and then click **Save** to commit your changes.



Note

When you drag and drop an output to the parsing area, it adds a new variable to the list, creating a blank extra.

Congratulations! You have completed creation of a PowerShell Activity. You have learned how to:

- Create a Custom Activity
- Work with Activity Inputs
- Work with Activity Outputs

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

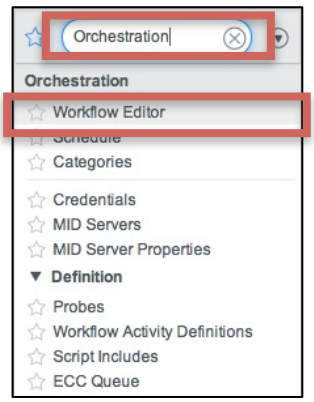
This lab explains how to create another PowerShell activity to capture current Memory usage from a Windows host, using the Activity Designer. This activity run through similar steps as the last lab. The two labs are used together in a subsequent workflow lab.

Lab 3.0 Create an Activity

Lab 3.1: Create a PowerShell Activity

This lab explains how to create the new PowerShell activity to capture current memory usage on a target host.

1. While logged into the lab instance, navigate to **Orchestration > Workflow Editor**.



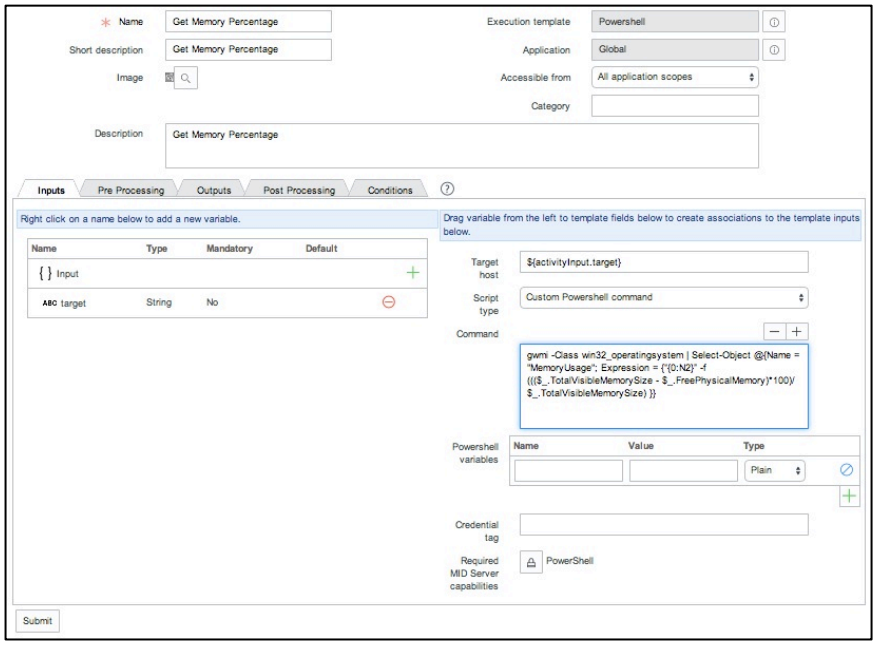
2. From the **Custom** tab in the right panel, click the + plus and create a new **PowerShell** Activity.

©2015 ServiceNow, Inc. All rights reserved.

3. Enter the following information in the **Activity Designer** form:

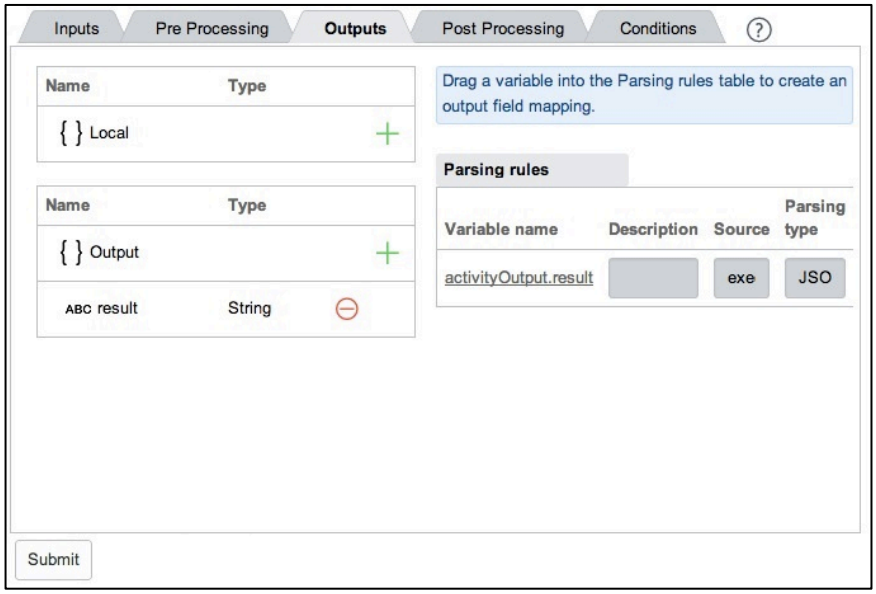
   Name: **get Memory Percentage - Windows**
   Short description: **Get Memory Percentage**
   Description: **Get Memory Percentage**
   Input variable: **name = target, Type = String, Mandatory = No, Default = blank**
   Target host: **${activityInput.target}**
   Command:

```
gwmi -Class win32_operatingsystem | Select-Object @{Name = "MemoryUsage";
Expression = {"{0:N2}" -f (($_.TotalVisibleMemorySize -
$_.FreePhysicalMemory)*100)/ $_.TotalVisibleMemorySize) }} | ConvertTo-JSON
```



®2015 ServiceNow, Inc. All rights reserved.

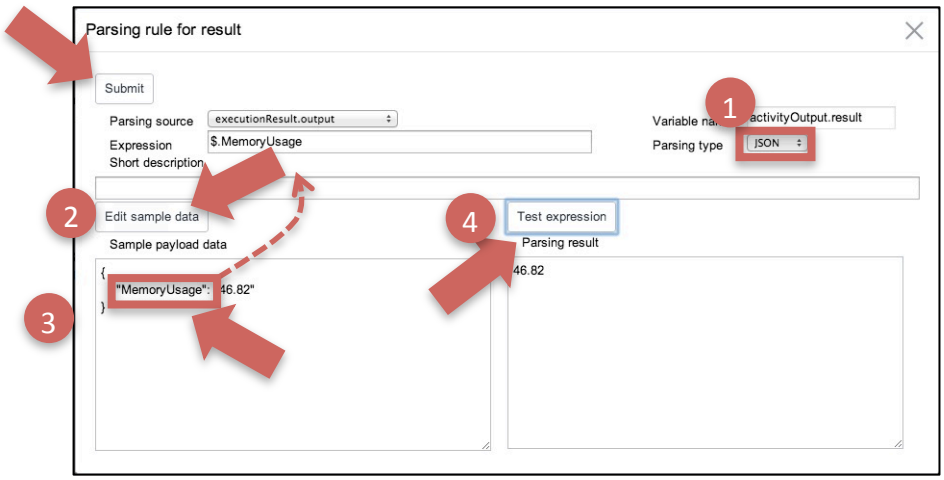
4. Switch to the **Outputs** tab and add an **Output** named **result**. Drag and drop the new result variable to the **Parsing rules**. A new Parsing rule screen appears.



®2015 ServiceNow, Inc. All rights reserved.

5. **(1)** *Select* the **Parsing Type *JSON***, *populate* the **Sample payload data** with the information below, **(2)** *click* **Parse sample data**, **(3)** *click* "**MemoryUsage**" within the sample data; the parsing Expression will automatically populate. *Click* **Test expression** to verify the sample output and *click* **Submit**:

   Parsing type: **JSON**
   Expression: **$.MemoryUsage**
   Sample payload data: **{"MemoryUsage":"36.67"}**



6. Click **Save** to commit the changes.

Congratulations! You have completed creation of an additional PowerShell Activity leveraging a JSON expression.

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

Lab 4.0 Create Your First Activity

This lab explains how to create a simple Probe activity (Shell command) that uses PING to get valuable information about a resource that needs to be triaged.

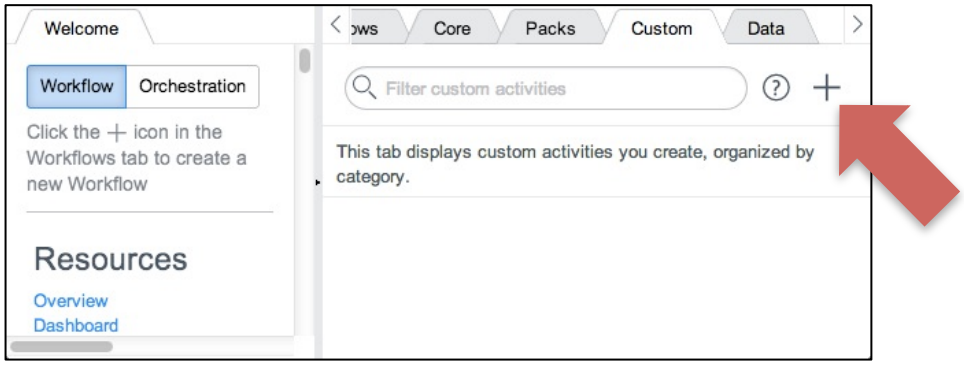
- Lab 4.1: Create a new Probe Activity
- Lab 4.2: Condition Results
- Lab 4.3: Lab Success Verification

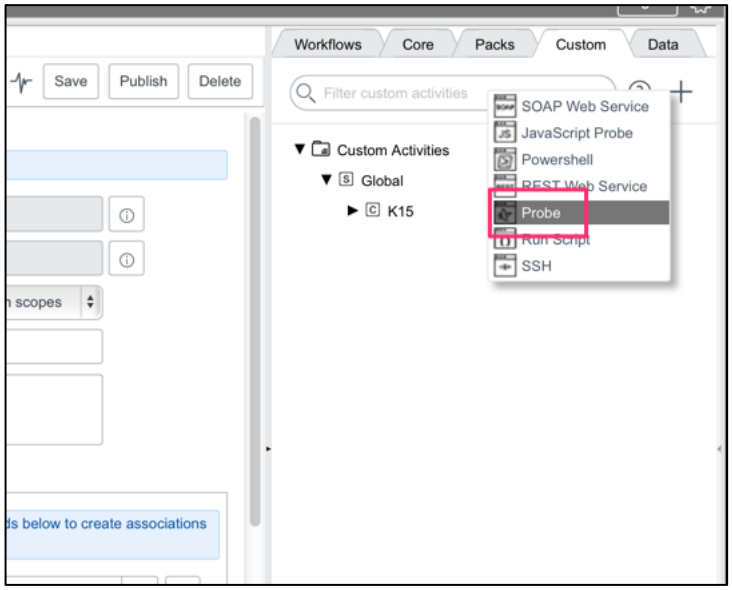
If you are unable to PING a particular Configuration Item in a triage use case, you cannot make any other request going forward. PING is a great place to start troubleshooting.

®2015 ServiceNow, Inc. All rights reserved.

Lab 4.1: Create a New Probe Activity

1. Within the Workflow Editor, switch to the **Custom** tab in the right panel, click the + plus to start a new **Probe** Activity.

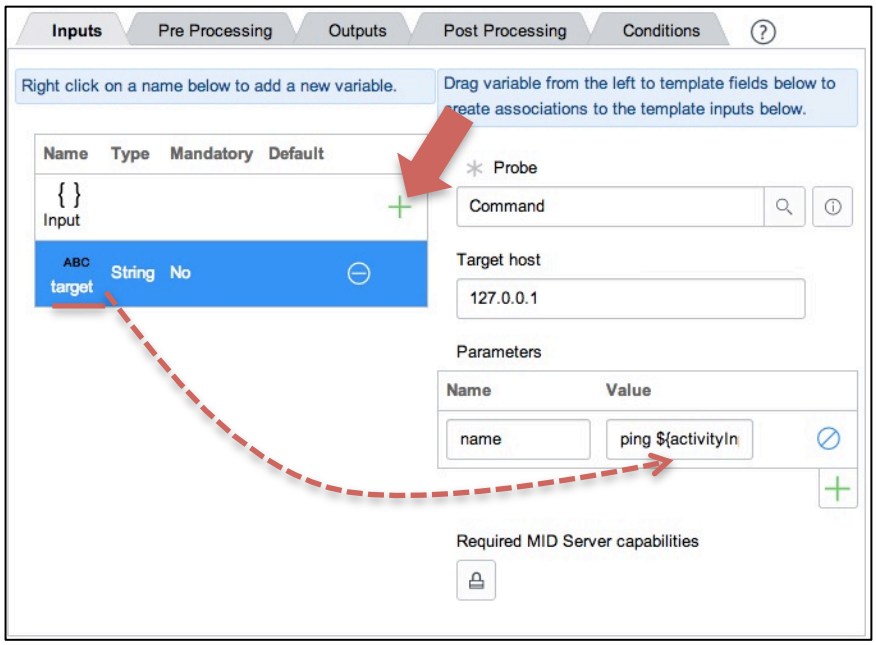




©2015 ServiceNow, Inc. All rights reserved.

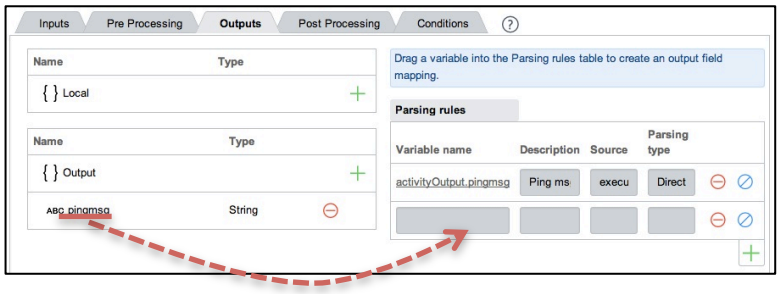
2. Enter the following information in the Activity form, then click + plus on the **Inputs** tab to add a new input. Enter **target** for the input name and press enter. Click and drag the new target input from the inputs on the left, to the **Parameters Value** field on the right.

   Name: **Ping**
   Short description: **Ping Activity**
   Application: **Global**
   Accessible from: **All application scopes**
   Description: **Whatever you want**
   Parameters: **name=name, value=ping *<drag your input value see image above>***



©2015 ServiceNow, Inc. All rights reserved.

3. Click the **Outputs** tab. Click the + plus on the **Output** section to add a new Output, enter **pingmsg** for the output name and press **enter**. Click and drag the new result output from the outputs on the left, to the **Parsing rules** field on the right. A parsing rule pop-up window appears.



When you drag and drop the output variable onto a Variable name in the parsing rules a wizard-like popup opens to help you parse the desired output. There are two ways of doing this. The simplest method is to get the raw XML from the ECC queue payload and assign that to the **pingmsg**.

4. Set **Parsing source** to **executionResult.payload** and **Parsing type** to **Direct**. Click **Submit**.

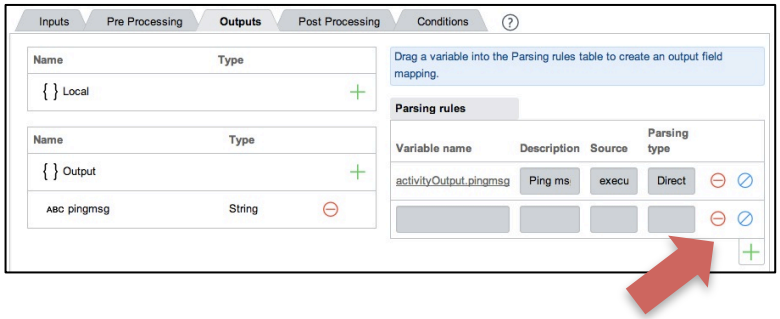


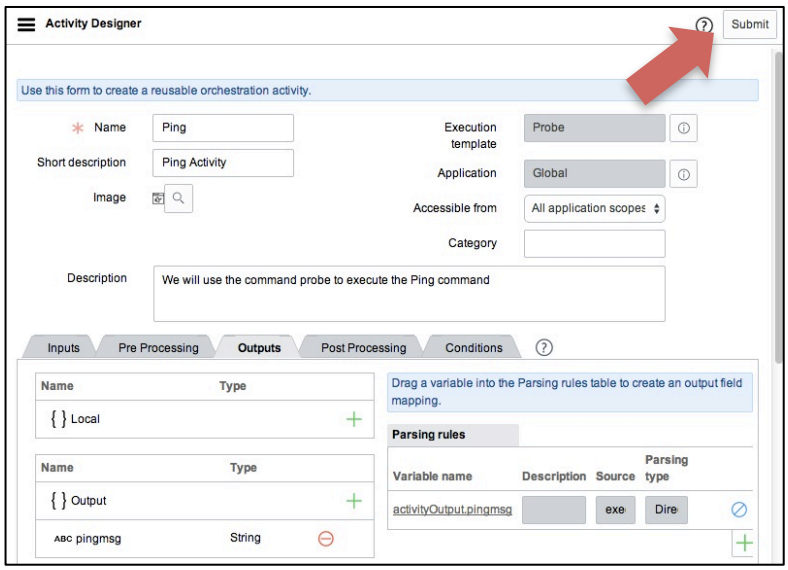
Note

When you drag and drop an output to the parsing area, it adds a new variable to the list, creating a blank extra.

©2015 ServiceNow, Inc. All rights reserved.

5. Click the red ⊖ minus to remove the extra parsing rule. Then click **Submit** to commit the changes.





®2015 ServiceNow, Inc. All rights reserved.

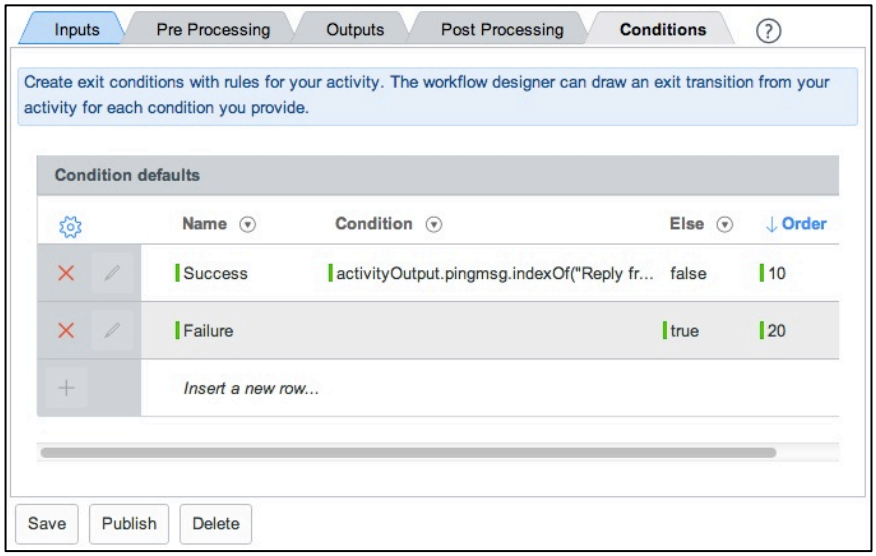
Lab 4.2: Condition Results

This lab explains how to create Condition responses to be used as transition paths in workflow.

The system evaluates the condition with the lowest order number first and stops when it finds the first true condition. The system delays evaluation of conditions with an **Else** value set to **true** until all conditions with an **Else** value set to **false** are evaluated. If the system does not find a true condition, it looks for a default condition in which the value for **Else** is **true** and takes this exit. You must have at least one default exit condition, or the activity hangs when the workflow runs.

1. Switch to the **Conditions** tab and enter the following conditions:

Name	2. Condition	Else	Order
Success	activityOutput.pingmsg.indexOf("Reply from") > -1	false	10
Failure		true	20



2. Click **Save** to commit the changes.

©2015 ServiceNow, Inc. All rights reserved.

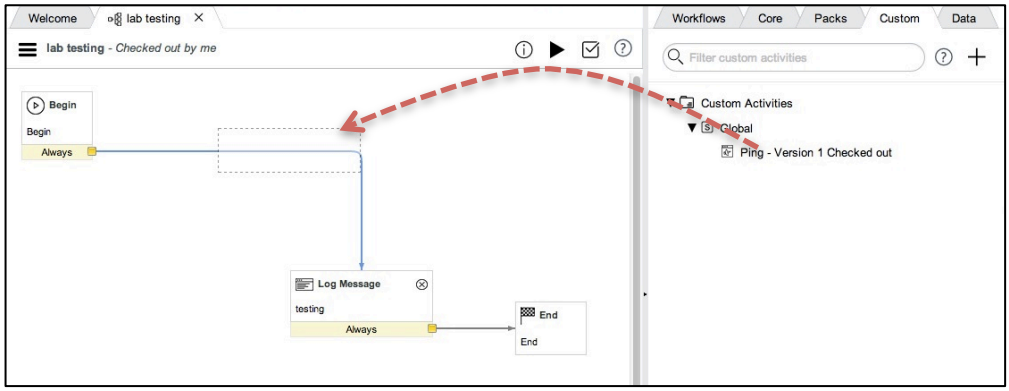
Lab 4.3: Lab Success Verification

This lab explains how to add the new activity to the test workflow.

1. Use the newly created activity in the test workflow or other workflows.
   Notice the version on it.

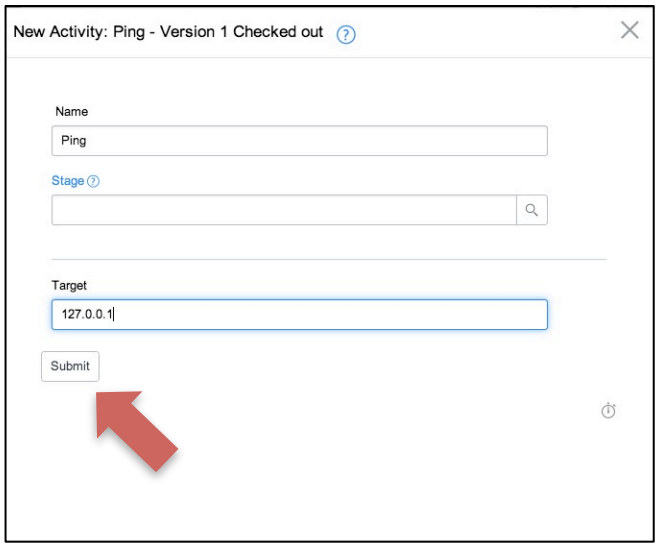
> When you drag and drop this onto your workflow use the **127.0.0.1** (or a valid IP Address) in the target field.

2. Open/Reopen the **lab testing** workflow and drag and drop the custom **Ping** Activity from the Custom Activity panel to the connecting line between **Begin** and **End**. When the transition line appears blue, release the mouse (this automatically connects the transitions to the new activity). A New Activity popup appears.

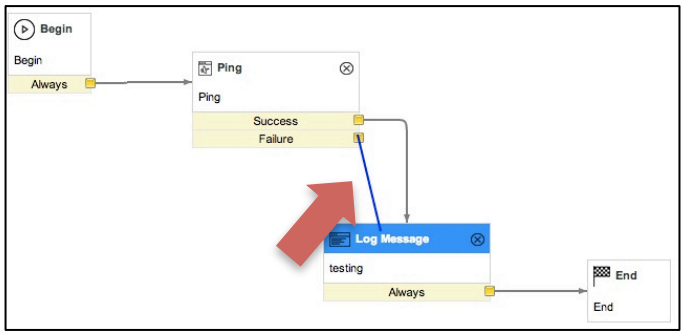


©2015 ServiceNow, Inc. All rights reserved.

3. Enter the following values and click **Submit**
   Name: **Ping**
   Target: **127.0.0.1**

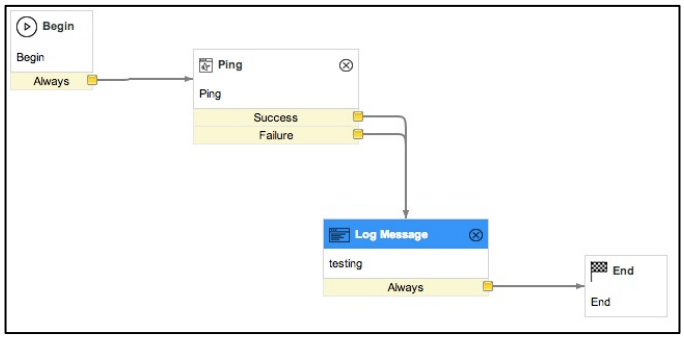


4. Drag the **Failure** response to the Log Message activity.

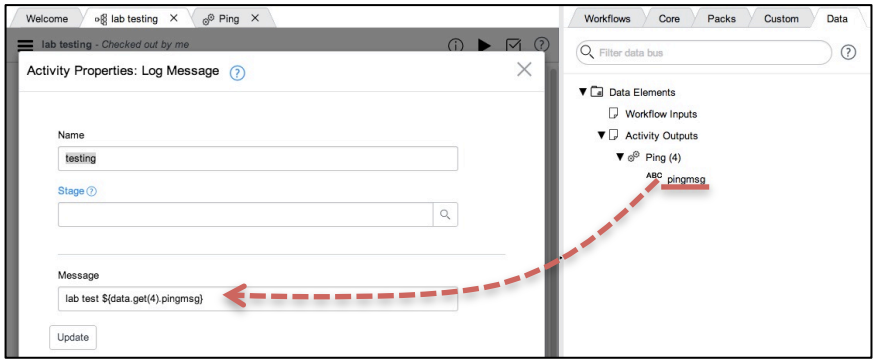


©2015 ServiceNow, Inc. All rights reserved.

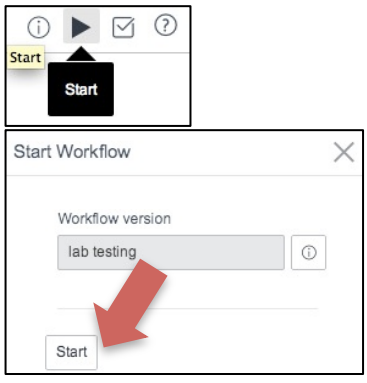
5. Double-click the **Log Message** activity. The Activity Properties appear.



6. Switch to the Data tab on the right; use the **Databus** during design time to drag and drop expected outputs into other transitioning activities. Drag and drop the **Activity Outputs > Ping > pingmsg** output into the **Message** statement.

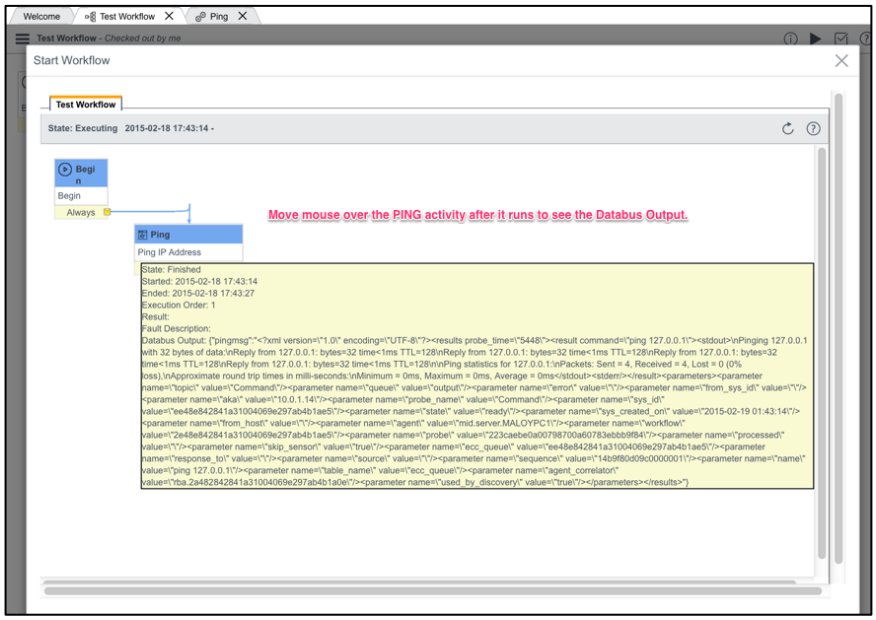


7. Click the **Play** icon on the top right of the Workflow canvas.



®2015 ServiceNow, Inc. All rights reserved.

8. The screen should look like the example when you run the workflow:



Congratulations! You have completed creation of a Probe Activity incorporating conditional results, and work with data outputs within a workflow. You have learned how to:

- Execute a shell command on a Mid Server
- Create conditional results based on outputs
- Leverage the databus to leverage Activity outputs within the workflow

®2015 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to create a PowerShell Activity to capture disk details on a target host. You pass input variables as parameters to the Activity command, work with additional outputs, and create condition results.

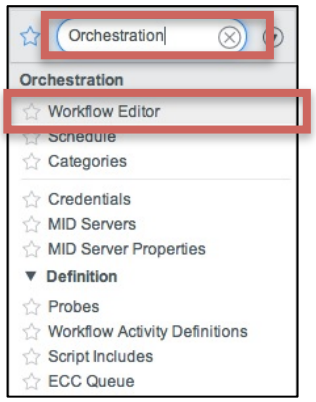
- Lab 5.1: Input Variables
- Lab 5.2: Output Variables
- Lab 5.3: Condition Results
- Lab 5.4: Advanced Script

Lab 5.0 Create an Activity

Lab 5.1: Input variables

This lab explains how to create the new PowerShell activity with an input for the target host.

1. While logged into the lab instance, navigate to **Orchestration > Workflow Editor**.



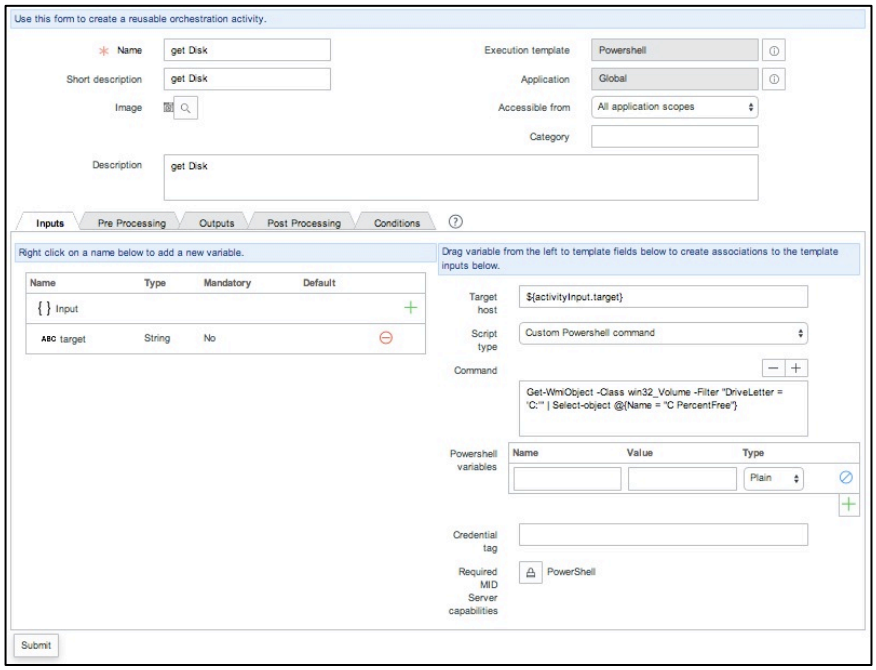
2. From the **Custom** tab in the right panel, click the + plus and create a new **PowerShell** Activity.

©2015 ServiceNow, Inc. All rights reserved.

3. Enter the following information in the **Activity Designer** form:

   Name: **get Disk - Windows**
   Short description: **Get Disk**
   Description: **Get Disk**
   Input variable**: name = target**, Type = **String, Manda**tory = **No, Defa**ult = **blank**
   Target host: ${**activityInput.t**arget}
   Command**:**

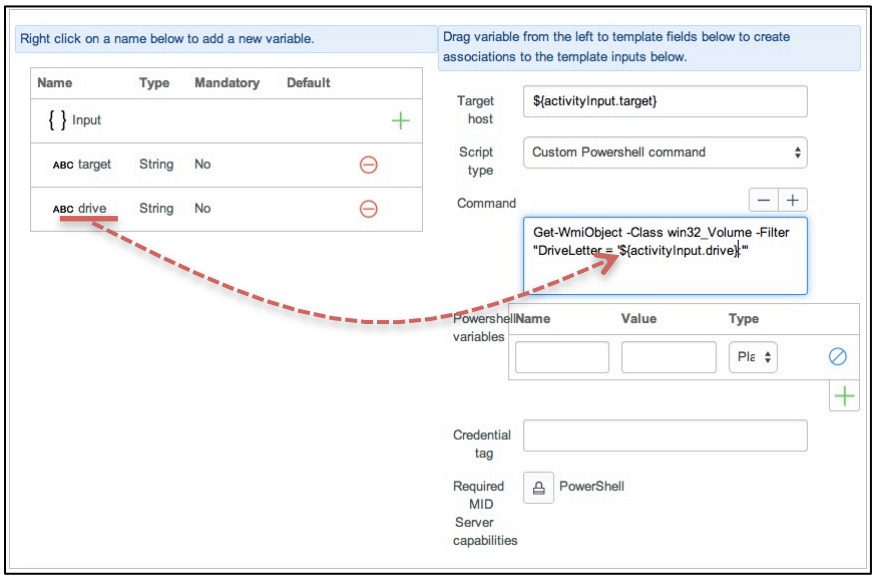
```
Get-WmiObject -Class win32_Volume -Filter "DriveLetter = 'C:'"
```



®2015 ServiceNow, Inc. All rights reserved.

4. Add a new input **drive**, and update the **command**, replacing the **C** drive letter with the variable syntax as follows:

```
Get-WmiObject -Class win32_Volume -Filter "DriveLetter =
'${activityInput.drive}:'"
```



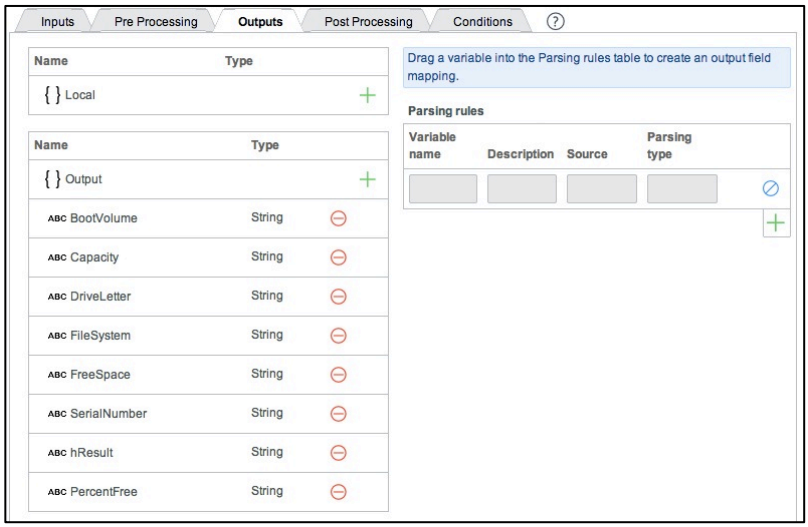
©2015 ServiceNow, Inc. All rights reserved.

Lab 5.2: Output Variables

This lab explains how to add outputs for the **win32_Volume WMI** class.

1. Switch to the **Outputs** tab and add the following **Output** variables**:**

BootVolume
Capacity
DriveLetter
FileSystem
FreeSpace
SerialNumber
hResult
errorMessage
PercentFree



Partial sample output for win32_Volume class:

```
__CLASS                    : Win32_Volume
__SUPERCLASS               : CIM_StorageVolume
__SERVER                   : WIN7VM
__NAMESPACE                : root\cimv2
Automount                  : True
BlockSize                  : 4096
BootVolume                 : True
Capacity                   : 32104284160
Caption                    : C:\
Compressed                 : False
Description                :
DriveLetter                : C:
FileSystem                 : NTFS
FreeSpace                  : 7038472192
IndexingEnabled            : True
Label                      :
MaximumFileNameLength      : 255
Name                       : C:\
PageFilePresent            : True
SerialNumber               : 1151123015
SystemName                 : WIN7VM
```

©2015 ServiceNow, Inc. All rights reserved.

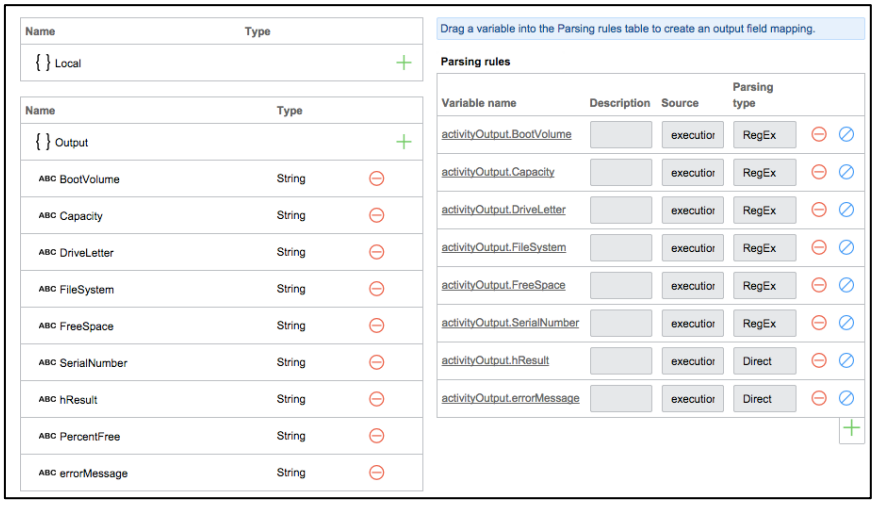
```
SystemVolume                    : False
```

2. Drag and drop each new result variable to the **Parsing rules**, except PercentFree. You'll set that during Task 3). A new Parsing rule screen appears. Complete the **Parsing rules** as follows:

Variable Name	Parsing Source	Parsing Type	Expression
activityOutput.BootVolume	executionResult.output	RegEx	BootVolume.*?: (.*)
activityOutput.Capacity	executionResult.output	RegEx	Capacity.*?: (.*)
activityOutput.DriveLetter	executionResult.output	RegEx	DriveLetter.*?: (.*)
activityOutput.FileSystem	executionResult.output	RegEx	FileSystem.*?: (.*)
activityOutput.FreeSpace	executionResult.output	RegEx	FreeSpace.*?: (.*)
activityOutput.SerialNumber	executionResult.output	RegEx	SerialNumber.*?: (.*)
activityOutput.errorMessage	executionResult.errorMessages	Direct	
activityOutput.hResult	executionResult.hResult	Direct	

Note

hResult is a special output containing the PowerShell exception codes (null if success), these can be useful to troubleshoot errors or control workflow.



©2015 ServiceNow, Inc. All rights reserved.

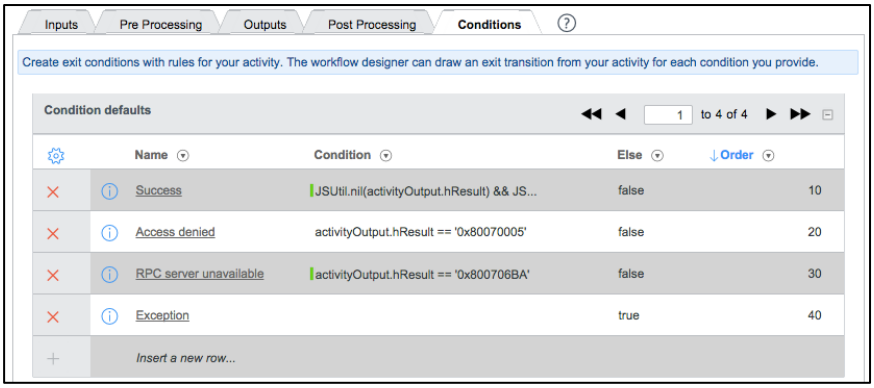
Lab 5.3: Condition Results

This lab explains how to create Condition responses to be used as transition paths in workflow.

The system evaluates the condition with the lowest order number first and stops when it finds the first true condition. The system delays evaluation of conditions with an **Else** value set to **true** until all conditions with an **Else** value set to **false** are evaluated. If the system does not find a true condition, it looks for a default condition in which the value for **Else** is **true** and takes this exit. You must have at least one default exit condition, or the activity hangs when the workflow runs.

1. Switch to the **Conditions** tab. Enter the following conditions:

Name	Condition	Else	Order
Success	JSUtil.nil(activityOutput.hResult) && JSUtil.nil(activityOutput.errorMessage)	false	10
Access denied	activityOutput.hResult == '0x80070005'	false	20
RPC server unavailable	activityOutput.hResult == '0x800706BA'	false	30
Exception		true	40



Bonus

Repeat Lab 5.3 for the activities you created during labs 2 and 3, be sure to also map the activityOutput.errorMessage in the Outputs tab.

©2015 ServiceNow, Inc. All rights reserved.

Lab 5.4: Advanced Script

This lab explains how to determine the **%free space** by leveraging outputs to perform a math expression in a post processing script.

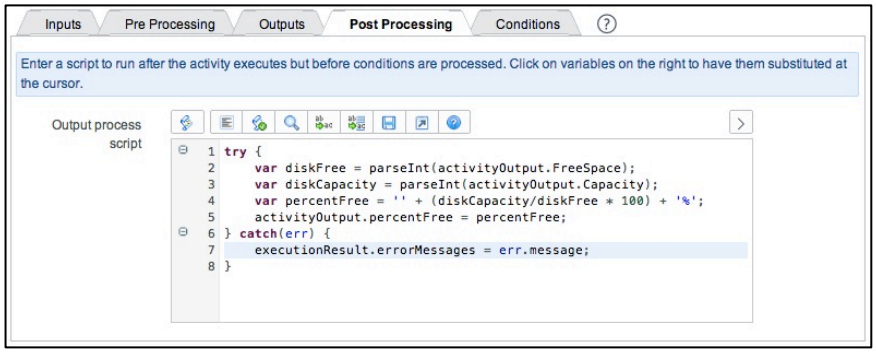
1. Switch to the **Post Processing** tab. Enter the following Output process script:

```
try {
    var diskFree = parseInt(activityOutput.FreeSpace);
    var diskCapacity = parseInt(activityOutput.Capacity);
    var percentFree = '' + (diskFree/diskCapacity * 100) + '%';
    activityOutput.PercentFree = percentFree;
} catch (err) {
    executionResult.errorMessages = err.message;
}
```

Note

Alternatively, to capture just the percentage disk free, you can use additional logic in the PowerShell script vs. the Post Processing script:

```
Get-WmiObject -Class win32_Volume -Filter "DriveLetter = 'C:'" | Select-
object @{Name = "C PercentFree"; Expression = {“{0:N2}” -f  (($_.FreeSpace
/ $_.Capacity)*100) } } | Format-List
```



2. Click **Submit** to commit the changes.

Congratulations! You have completed creation of a PowerShell Activity incorporating conditional results and a post processing script. You have learned how to:

- Incorporate Activity inputs to your Activity command
- Work with outputs
- Create Condition results based on Activity outputs
- Work with post processing script to perform extra logic

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

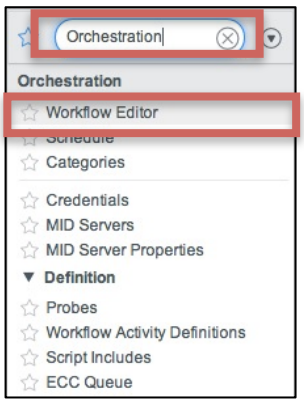
This lab explains how to create a PowerShell activity to capture current Windows services that are set to start Automatically, but are not currently running, using the Activity Designer. This activity used together with prior activities in a subsequent workflow lab.

Lab 6.0 Create an Activity

Lab 6.1: Create a PowerShell Activity

This lab explains how to create the new PowerShell activity to capture services that are not running on a target host.

1. While logged into the lab instance, navigate to **Orchestration > Workflow Editor**.



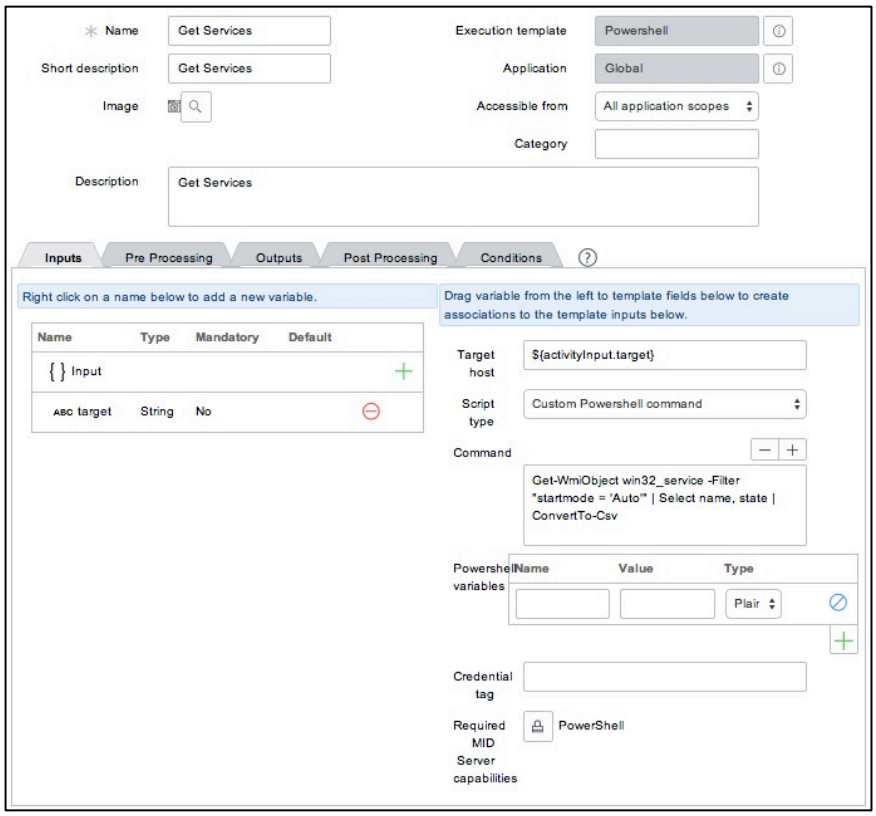
2. From the **Custom** tab in the right panel, click the + plus and create a new **PowerShell** Activity.

©2015 ServiceNow, Inc. All rights reserved.

3. Enter the following information in the **Activity Designer** form:

   Name: **Get Services - Windows**
   Short description: **Get Services**
   Description: **Get Services**
   Input variable**: name = target, T**ype = **String**, **Mandatory = No**, De**fault** = **blank**
   Target host: ${**activityInput.targ**et}
   Command**:**

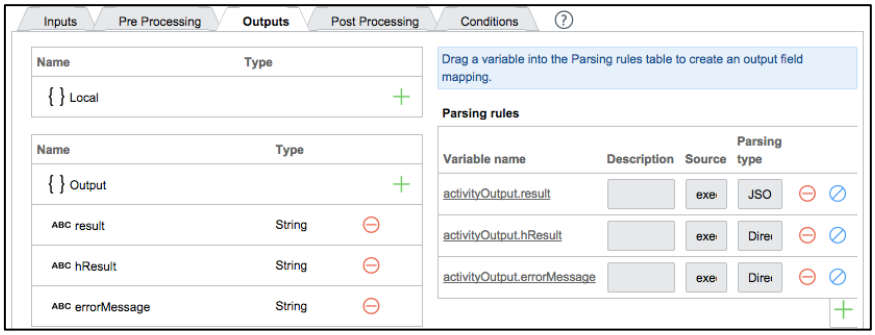
```
Get-WmiObject win32_service -Filter "startmode = 'Auto' AND state =
‘stopped’" | Select name, state
```



®2015 ServiceNow, Inc. All rights reserved.

4. Switch to the **Outputs** tab and add the following outputs and Parsing Rules:

Output	Type	Variable name	Source	Parsing type
hResult	String	activityOutput.hResult	executionResult.hResult	Direct
errorMessage	String	activityOutput.errorMessage	executionResult.errorMessages	Direct
Result	String	activityOutput.result	executionResult.output	Direct



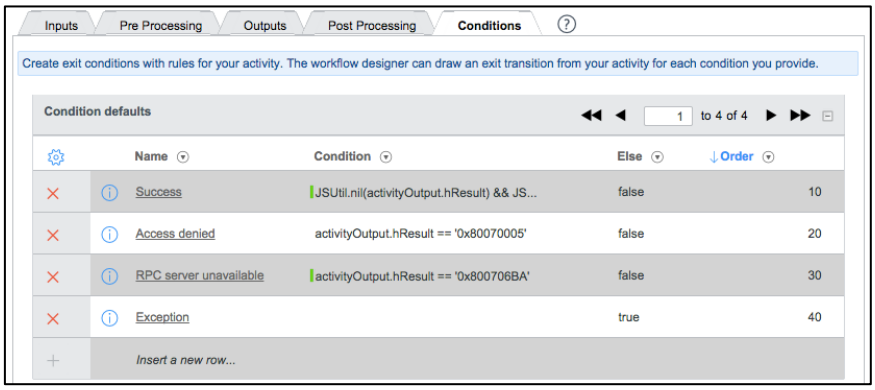
©2015 ServiceNow, Inc. All rights reserved.

5. Switch to the **Conditions** tab. Enter the following conditions:

Name	Condition	Else	Order
Success	JSUtil.nil(activityOutput.hResult) && JSUtil.nil(activityOutput.errorMessage)	false	10
Access denied	activityOutput.hResult == '0x80070005'	false	20
RPC server unavailable	activityOutput.hResult == '0x800706BA'	false	30
Exception		true	40

Tip

Repeat step 5 to improve the PowerShell activities created in labs 2 and 3.



6. Click **Submit** to commit your changes.

You can create a similar Activity to get the state of all services, then build a list for additional workflow to loop through failed services and perform a start service.

Congratulations! You have completed creation of an additional PowerShell Activity to capture failed Windows services.

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to create a new Triage workflow leveraging the activities you created. This workflow can be used by operators to test target servers or automatically by Event Management rules, to collect current state information at time of Event. This lab covers the following topics:

- Lab 7.1: Create a Triage Workflow
- Lab 7.2: Test Your Triage Workflow
- Lab 7.3: Create Flow Inputs and Outputs

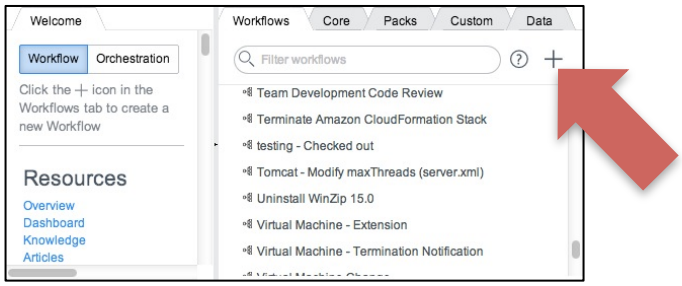
Lab 7.0 Create a Triage Workflow

Lab 7.1: Create a Triage Workflow

This lab explains how to wire together custom Activities, and log activity results in the workflow log.

1. While logged into the lab instance, navigate to **Orchestration > Workflow Editor**. A new browser tab opens.

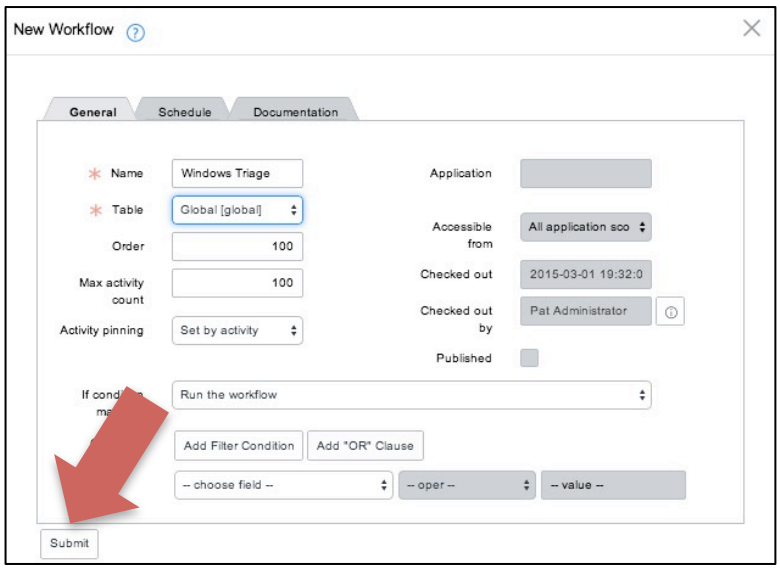
2. From the **Workflows** tab in the right panel, click the + plus to start a new Workflow. A new workflow popup appears.



©2015 ServiceNow, Inc. All rights reserved.

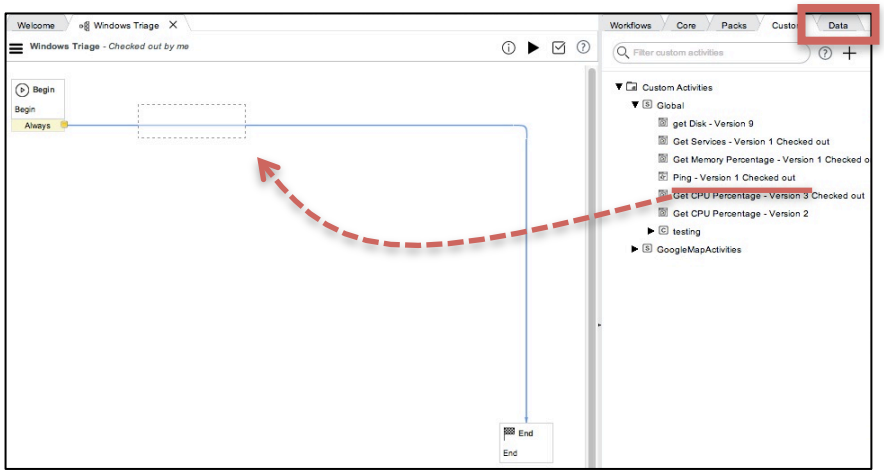
3. Enter the following **inputs**. Click **Submit**.

   Name: **<your name> Windows Triage**
   Table: **Global**



4. Switch to the **Custom** tab on the right panel to show the custom **Activities**.

5. Drag and drop the **Custom Activities > Global > Ping** Activity from the Custom Activity panel to the connecting line between **Begin** and **End**. When the transition line appears blue, release the mouse (this automatically connects the transitions to the new activity). A New Activity: Ping panel appears.



©2015 ServiceNow, Inc. All rights reserved.

6. Enter the following **inputs**. Click **Submit**.
   Name: **Ping**
   Target**: 127.0.0.1**



7. Drag and wire the other custom Activities as follows:

Get CPU Percentage

- Name: **Get CPU Utilization**
- Target: **127.0.0.1**

Get Memory Percentage

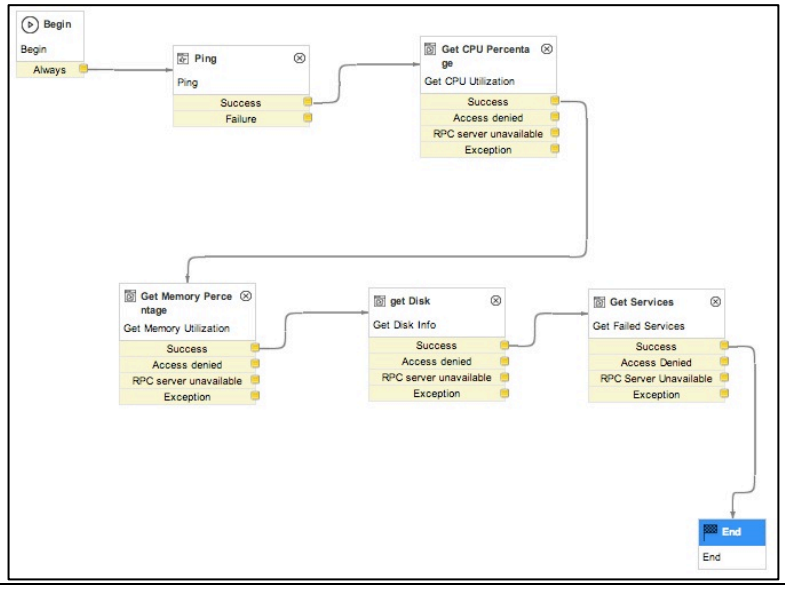
- Name: **Get Memory Utilization**
- Target: **127.0.0.1**

Get Disk Details

- Name: **Get Disk Info**
- Target: **127.0.0.1**
- Drive: **C**

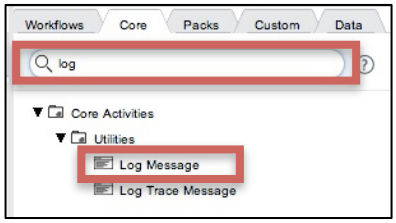
Get Services

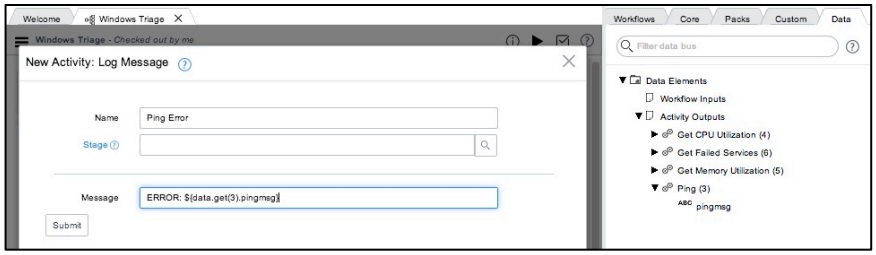
- Name: **Get Failed Services**
- Target: **127.0.0.1**



®2015 ServiceNow, Inc. All rights reserved.

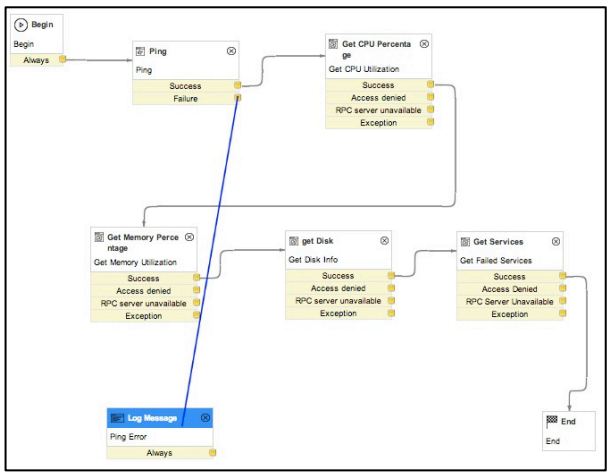
8. Switch to the **Core** tab, filter for **log**, and drag and wire the **Core Activities > Utilities > Log Message** to the canvas.





9. Enter Name**: Ping Error**, and drag the **Data, Data Elements > Activity Outputs > Ping > pingmsg** to the **Message** field.

10. Drag and wire the **Ping Failure** response to the **Log Message** activity.

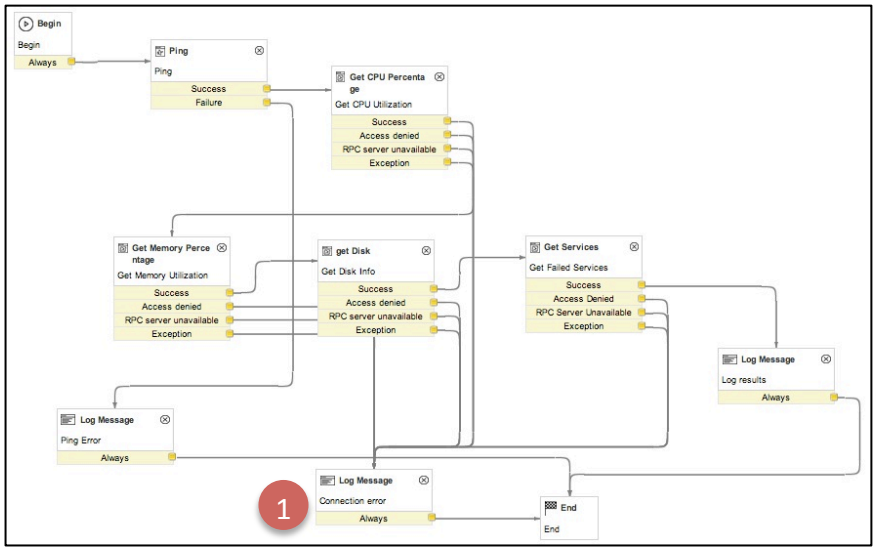


©2015 ServiceNow, Inc. All rights reserved.

11. Add two more Log Message activities, and drag and wire the canvas as depicted above.

(1) Log Message (wire all PowerShell errors to this activity)
Name: **Connection error**
Message: **Connection error: CPU hResult=${data.get(*n*).hResult} Memory hResu*l*t=${data.get(*n*).hResult} Disk hResult=${data.get(*n*).hResult} Services hRes*u*lt=${data.get(*n*).hResult}**
(where **n** aligns to the activity defined in the data tab, e.g., use drag & drop)

(2) Log Message (wire the final Get Services Success message to this activity)
Name: **Log Results**
Message: **CPU Utilization: ${data.get(*n*).result}% Memory Utilization: ${data.get(*n*).result}% Disk Free: ${data.get(*n*).PercentFree}% Failed Services: ${data.get(*n*).result}**

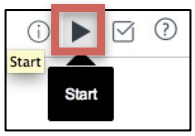


®2015 ServiceNow, Inc. All rights reserved.

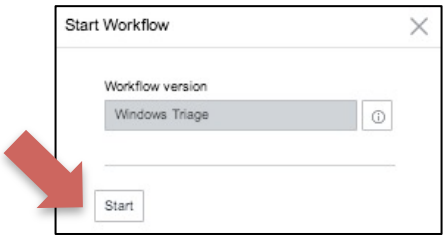
Lab 7.2: Test Your Triage Workflow

This lab explains how to execute the triage workflow and verify the log output.

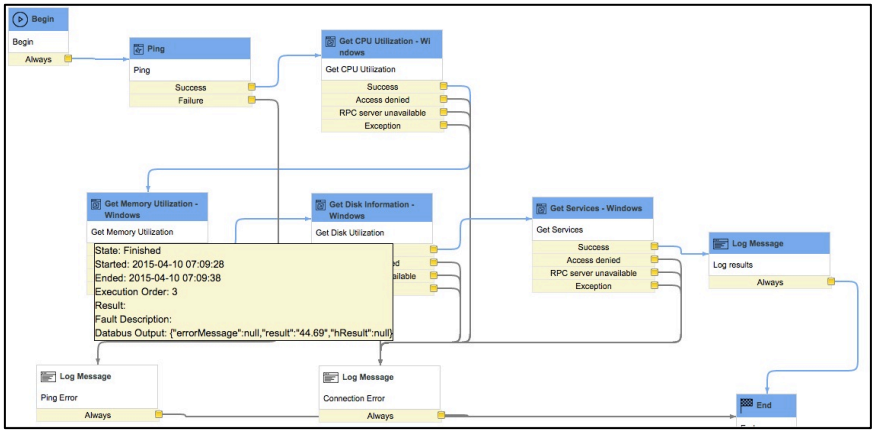
1. Click the **Play** icon on the top right of the Workflow canvas.



2. Click **Start** to start the Workflow.



3. Mouse over the activities to inspect the start times, end times, and databus information. In later labs, this contains additional workflow information and Orchestration results.
Click the ✕ on the top right to close the run window.

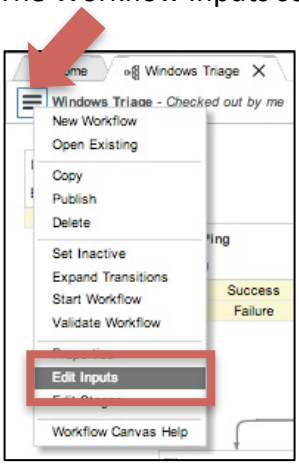


®2015 ServiceNow, Inc. All rights reserved.

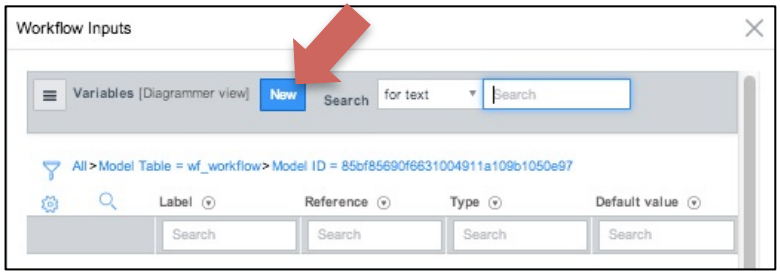
Lab 7.3: Create Flow Inputs and Outputs

This lab explains how to add a flow input so the triage flow may be used as a subflow within another workflow.

1. While editing the Windows Triage workflow, click the menu on the top left and choose **edit inputs**.
   The Workflow Inputs screen appears.



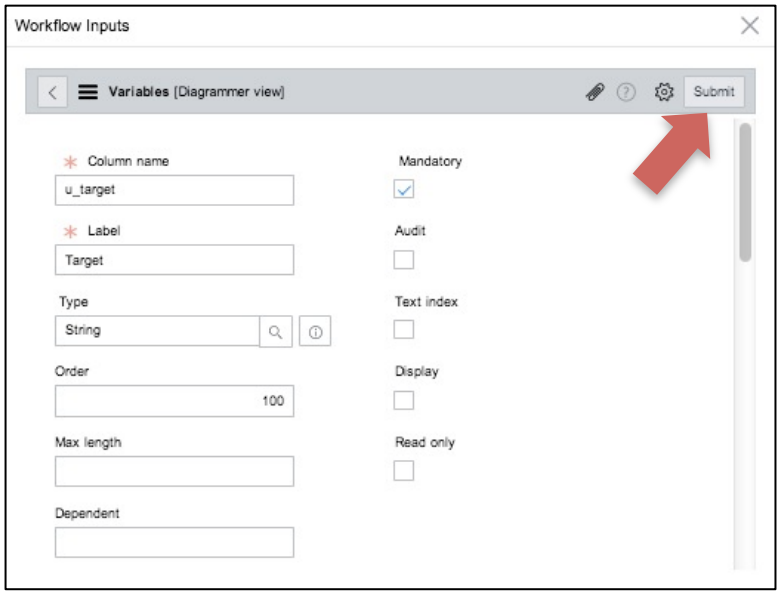
2. Click **New***;* a new variable screen appears.



®2015 ServiceNow, Inc. All rights reserved.

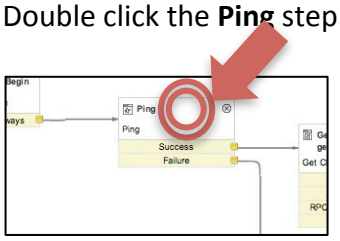
3. Enter the following inputs, then click **Submit**:

   Column name: **u_target**
   Label: **Target**
   Type: **String**
   Mandatory: **checked**



4. Click the ✕ on the top right to close the Workflow Inputs window, and return to the workflow.

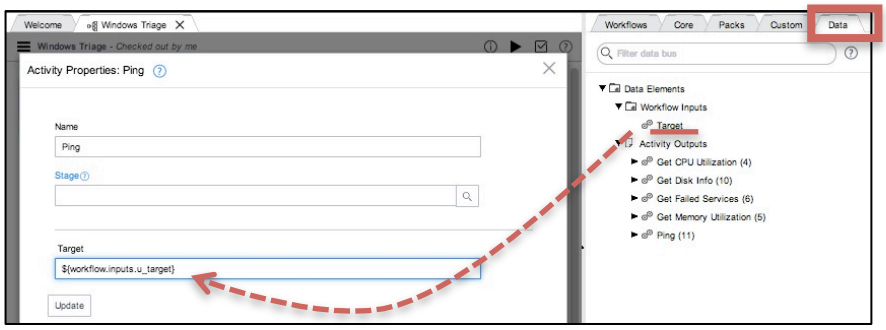
5. Double click the **Ping** step to update the target field with the new flow variable.



©2015 ServiceNow, Inc. All rights reserved.

6. Switch to the **Data** tab in the right tab, then drag and drop the **Data Element > Workflow Inputs > Target** to the ping **Target** input.

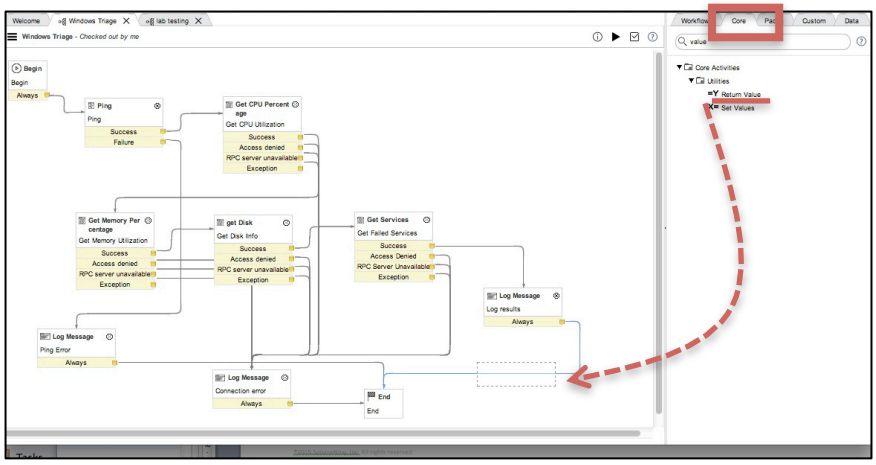
   Target: **${workflow.inputs.u_target}**



7. Repeat steps 5 & 6 for the other workflow activities:

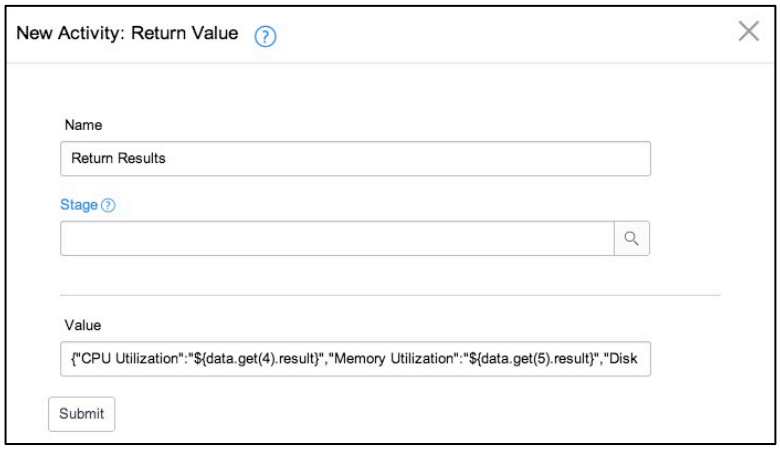
   Get CPU Utilization
   Get Memory Utilization
   Get Disk Info
   Get Failed Services

8. Filter the **Core** Activities, and **drag** the **=Y Return Value** to the workflow between the Log Message Activity, and the End.



©2015 ServiceNow, Inc. All rights reserved.

9. A New Activity: Return Value dialog appears.



10. Enter the following values, and click **Submit**:

    Name: **Return Results**
    Value:

```
{"CPU_Utilization":"${data.get(n).result}","Memory_Utilization":"${data.get(n).result}","Disk_Utilization":"${data.get(n).PercentFree}","Failed_Services":"${data.get(n).result}"}
```

    Where **n** is the proper output field from the Data tab

Note

Using the JSON format allows you to retrieve the field values programmatically in later steps/flows.

Congratulations! You have created a Windows Triage workflow incorporating custom Activities. You have learned how to:

- Wire together custom activities
- Leverage the databus to use outputs and inputs to other steps
- Create flow inputs and set flow outputs

©2015 ServiceNow, Inc. All rights reserved.

Lab Goal

Lab 8.0 Create REST Web Service Activity

This lab explains how to create a simple REST web service activity that calls the public Google Maps web service to get distance and driving duration between two locations.

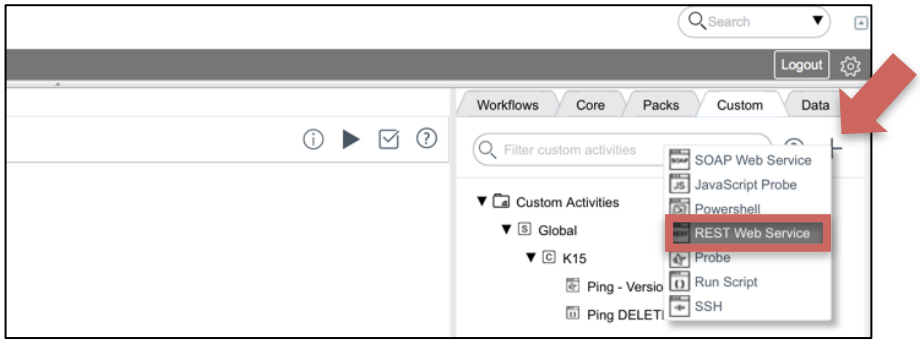
- Lab 8.1: Create a New REST Activity
- Lab 8.2: Lab Success Verification

This activity gives you insight on how to build web service activities, since so many integrations from the Cloud rely on web services.

Lab 8.1: Create a New REST Activity

This lab explains how to create a new Activity using the REST Web Service Activity Type.

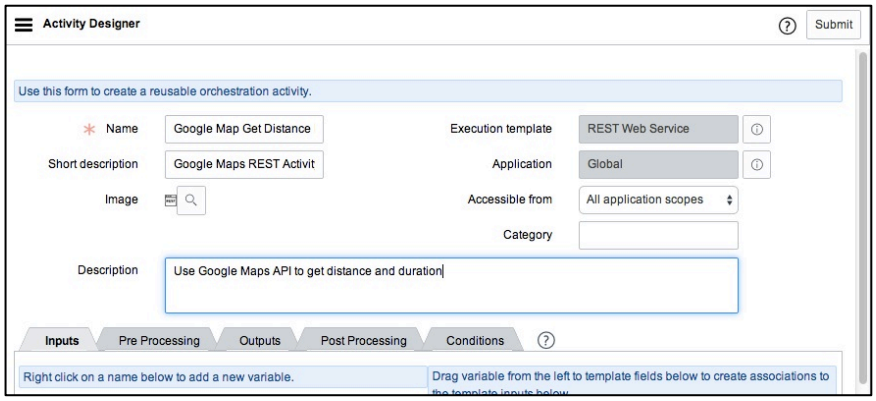
1. Click the + sign in the right corner to create a new **REST Web Service** Activity.



©2015 ServiceNow, Inc. All rights reserved.

2. Enter the following information in the Activity form:

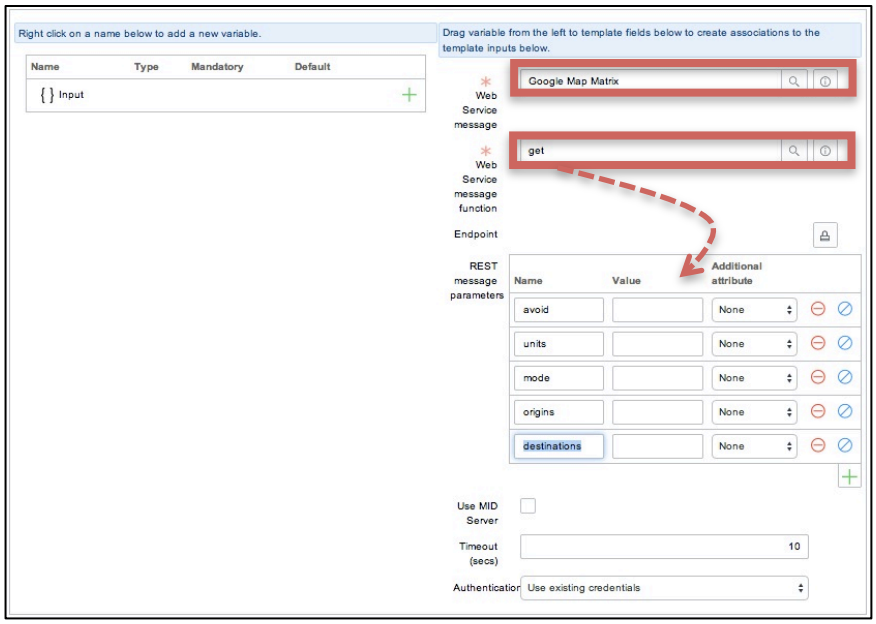
   Name: **Google Map Get Distance Duration**
   Short description: **Google Maps REST Activity**
   Application: **Global**
   Accessible from: **All application scopes**
   Description: **Whatever you want**



®2015 ServiceNow, Inc. All rights reserved.

3. Scroll down and fill in the **Inputs** tab details:

   Web Service message: **Google Map Matrix**
   Web Service message: **get**



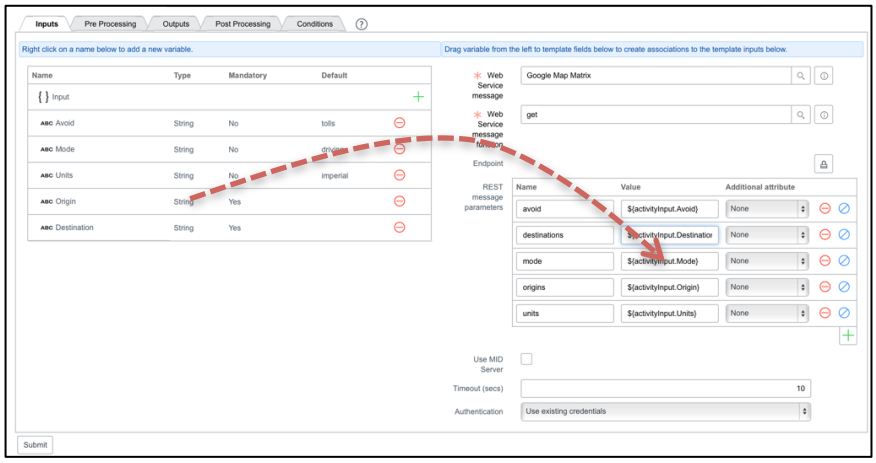
Note

As long as t**he REST message** is set up with test variables they auto populate when selected on this form. The REST message parameters automatically come in based on the selections.

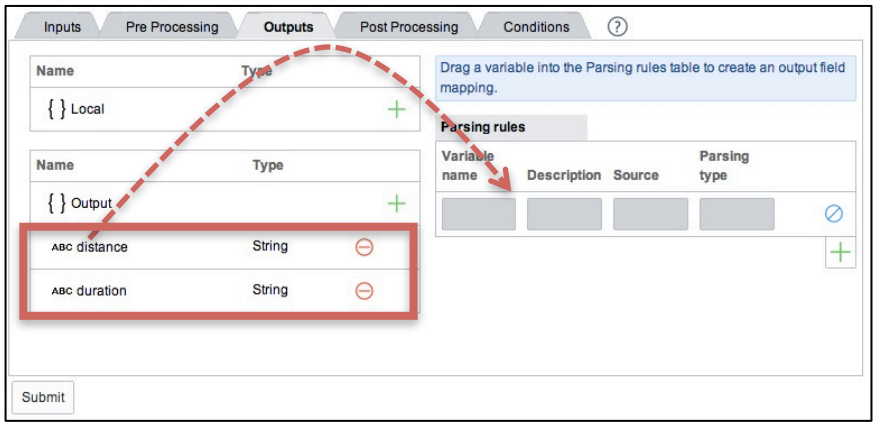
©2015 ServiceNow, Inc. All rights reserved.

4. Create the following Activity input variables for each REST input. Then drag and drop the input values into the REST message parameters field **Values**.

   Avoid
   Units
   Mode
   Origins
   Destinations



5. Switch to the **Output** tab and add the Output variables in the example. Then drag and drop to the **Parsing rules Variable name** field. A wizard opens to map the output.

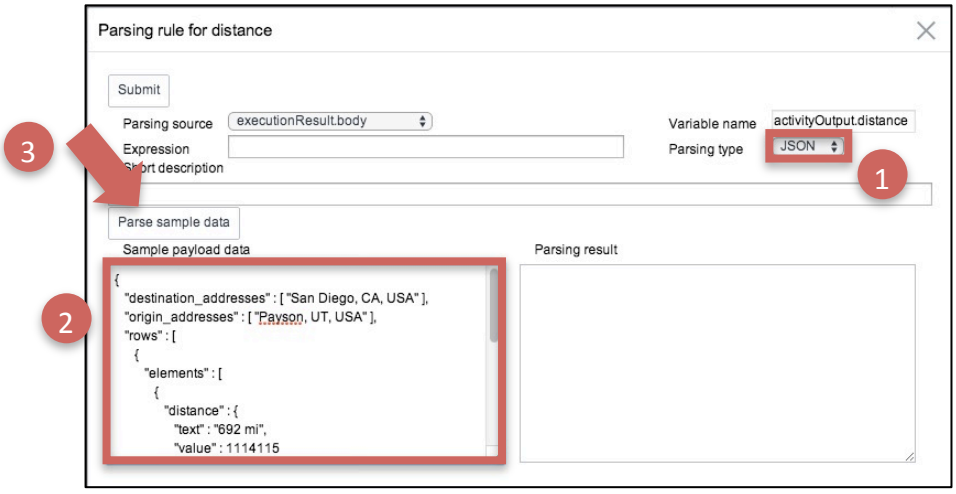


©2015 ServiceNow, Inc. All rights reserved.

6. Change the Parsing type to **(1) JSON**, enter the below **(2) Sample payload data**. Then click **Parse sample data**.

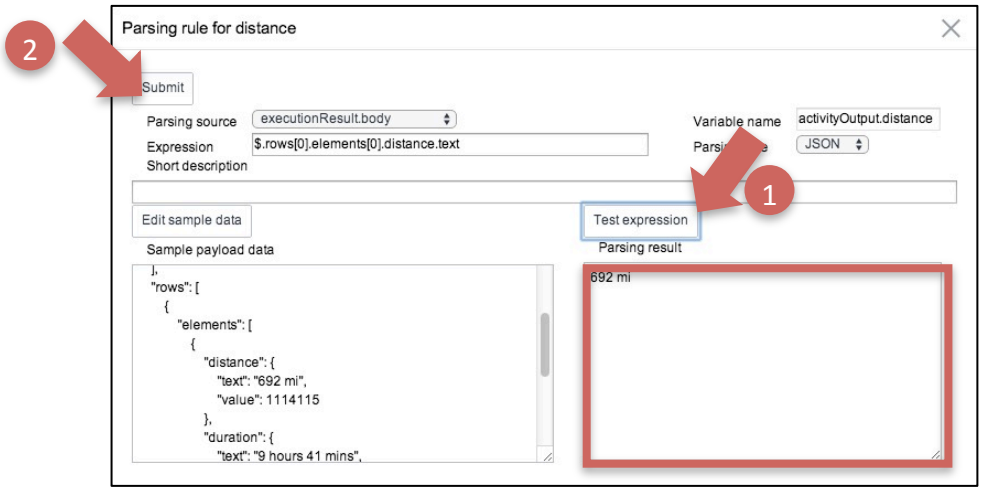
Sample **REST Response Data**


```
{
   "destination_addresses" : [ "San Diego, CA, USA" ],
   "origin_addresses" : [ "Payson, UT, USA" ],
   "rows" : [
      {
         "elements" : [
            {
               "distance" : {
                  "text" : "692 mi",
                  "value" : 1114115
               },
               "duration" : {
                  "text" : "9 hours 41 mins",
                  "value" : 34878
               },
               "status" : "OK"
            }
         ]
      }
   ],
   "status" : "OK"
}
```

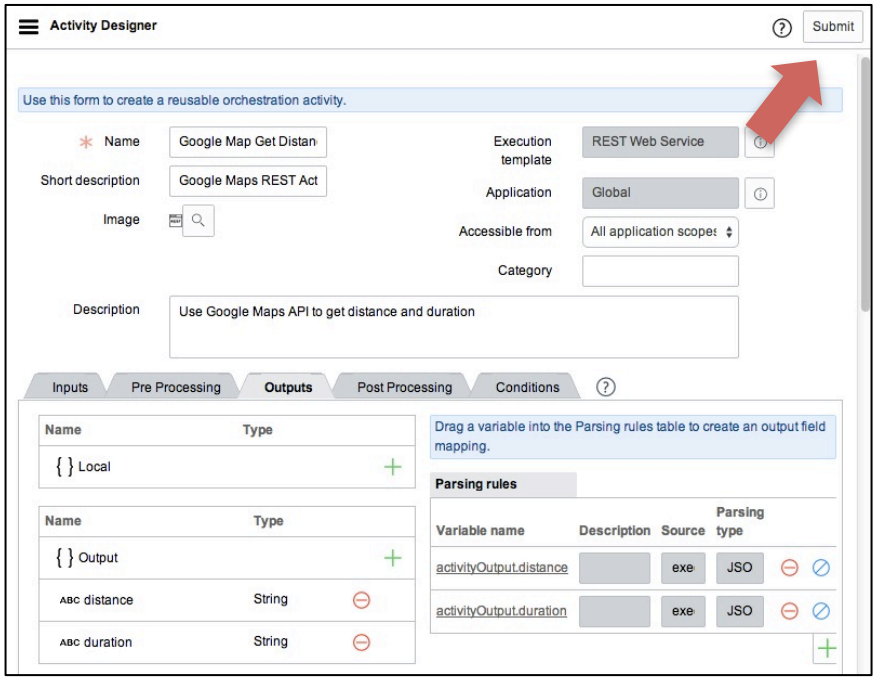
©2015 ServiceNow, Inc. All rights reserved.

7. Click **(1) Test expression** to verify the **Parsing result**, then click **(2) Submit**.



8. Repeat step **6 and 7** for duration.

9. Click **Submit** to save the newly created REST Activity.



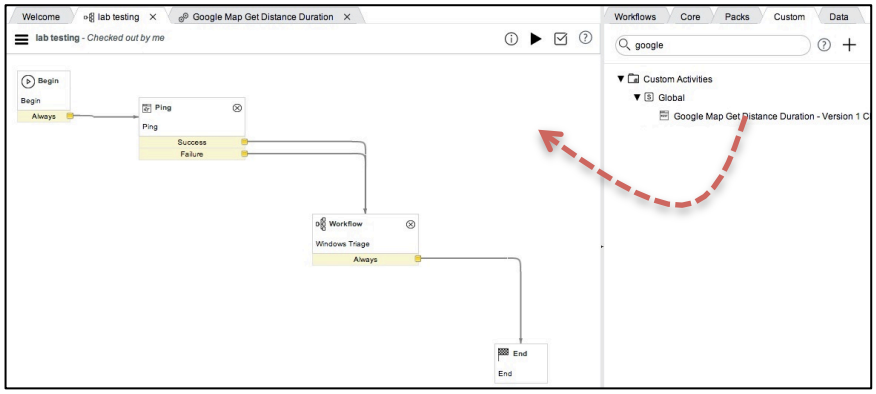
You now have a newly created activity you can reuse in your test workflow or other workflows. **Notice the version on it.**

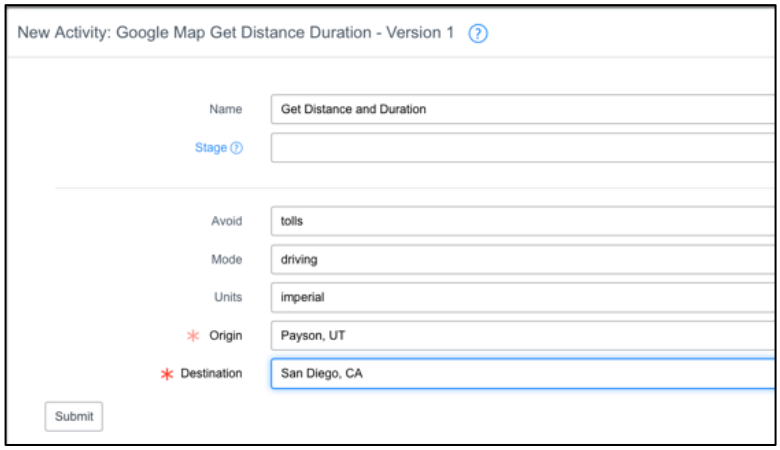
©2015 ServiceNow, Inc. All rights reserved.

Lab 8.2: Lab Success Verification

This lab explains how to update the lab testing workflow to verify the Google Maps API.

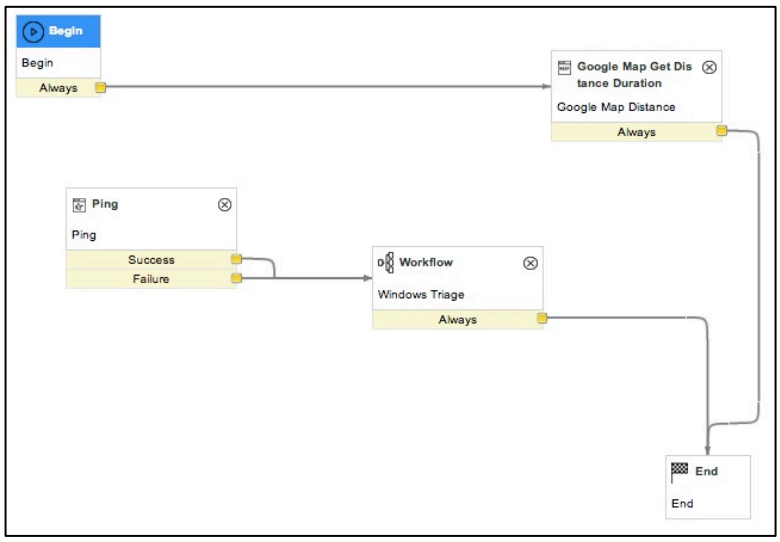
1. Drag and drop the Google Maps Activity to the testing workflow. When you use the activity, fill out the details you want (remember that the inputs can come from a variety of sources, catalog variables, the current object that executed the workflow, derived with scripts, etc.). Enter the testing details and click **Submit.**



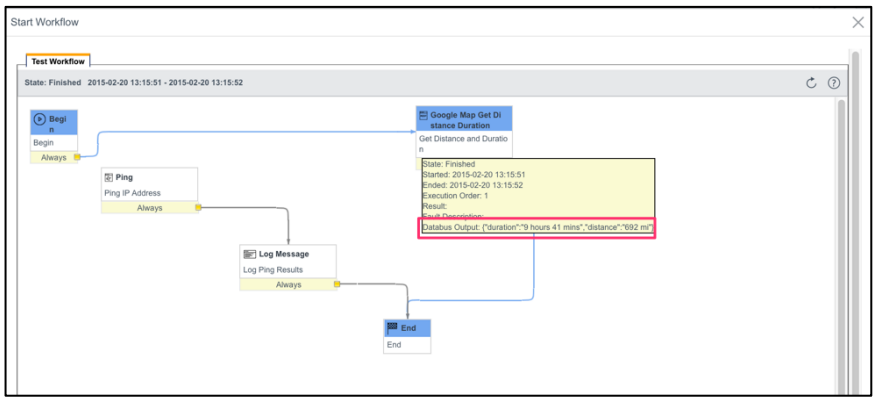


©2015 ServiceNow, Inc. All rights reserved.

2. The final workflow should be similar to the example. Run the workflow to test.



3. The screen should look similar to the example after you run the workflow.



®2015 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to create a new flow that is executed when an Incident is created.

- Lab 9.1: Call a subflow
- Lab 9.2: Lab Verification

Lab 9.0 Execute flow from Incident

Lab 9.1: Call a Subflow

This lab explains how to create a new flow that leverages the Windows Triage flow and the Google Maps Activity.

1. While logged into your lab instance, navigate to **Orchestration > *W*orkflow Editor**.
   A new browser tab opens.

2. From the **Workflows** tab in the right panel, click the + plus to start a new Workflow.
   A new workflow popup appears.

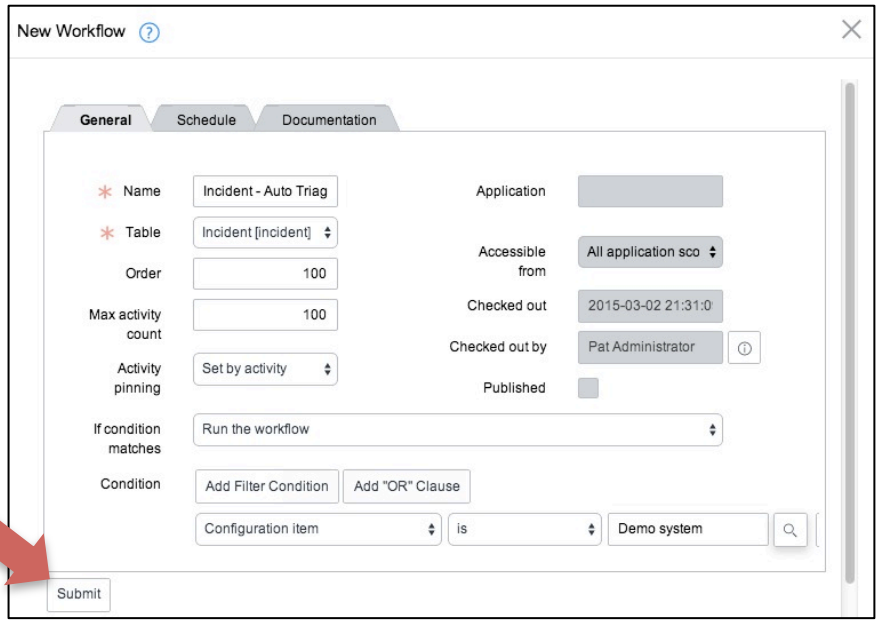
®2015 ServiceNow, Inc. All rights reserved.

3. Enter the following inputs and click **Submit**.

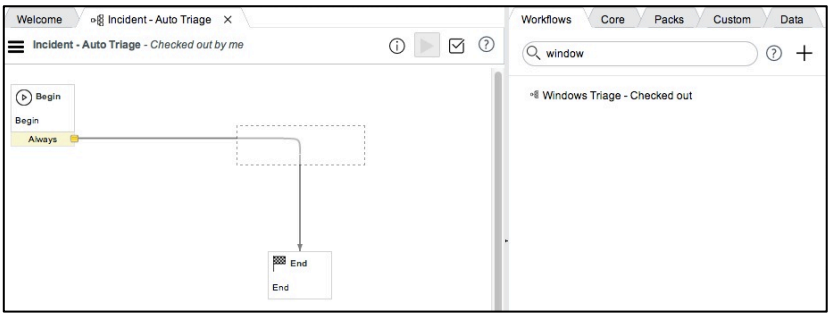
   Name: **Incident – Auto Triage**
   Table: **Incident [Incident]**
   Condition: **Configuration item is Demo system**

Note

The condition tells the workflow to execute when the Configuration item entered on the Incident record is Demo system. We read the Demo system CI address and pass the information to the Google Map Activity in a later step.



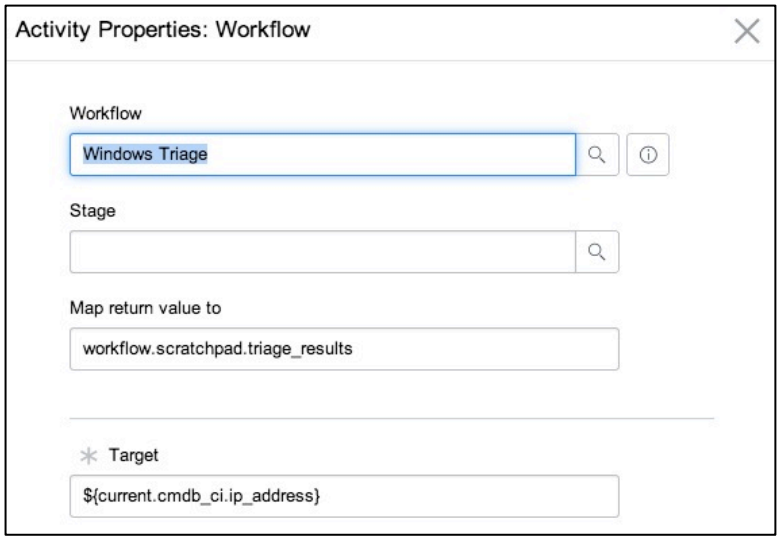
4. Locate the **Windows Triage** workflow from the **Workflows** navigator, and drag and drop it to the canvas. The Activity inputs appears.



©2015 ServiceNow, Inc. All rights reserved.

5. Enter the following inputs:

   Workflow: **Windows Triage**
   Map return value to: **workflow.scratchpad.triage_results**
   Target: **${current.cmdb_ci.ip_address}**



Note

The **Map return value to** field captures the results of the executed workflow. We configured the results to a JSON formatted string. This allows us to dot-walk the **triage_results** object for the various fields we captured during the triage activities. We could have formatted the output as freeform text as well, but in this lab, use JSON for extra logic.

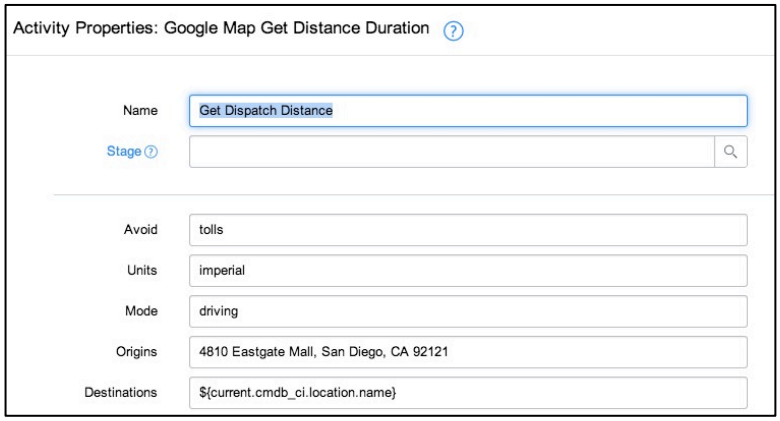
Note

The target value is using an expression to *dot walk* the current record (the Incident that executed the Workflow).

current.cmdb_ci.ip_address represents the Incident object
current.**cmdb_ci**.ip_address represents the Configuration Item field on the Incident
current.cmdb_ci.**ip_address** represents the IP Address field on the CI

©2015 ServiceNow, Inc. All rights reserved.

6. Locate the Google Map Get Distance Duration custom activity from the Custom Activities navigator, and drag and drop it to the workflow. A New Activity popup appears.



7. Enter the following inputs and click **Submit**:

   Name: **Get Dispatch Distance**
   Avoid: **tolls**
   Units: **imperial**
   Mode: **driving**
   Origins: **4810 Eastgate Mall, San Diego, CA 92121**
   Destination: **${current.cmdb_ci.location.name}**

Note

The 4810 address is a ServiceNow corporate office, where we'll dispatch a technician from. The $(current.cmdb_ci.location.name} is the address stored on the CI entered in the Incident. We'll use the Google Maps API to calculate the travel distance to the target repair location.

©2015 ServiceNow, Inc. All rights reserved.

8. From the **Core** Activity navigator, locate and drag and drop **Core Activities > Utilities > Run Script** to the workflow.



9. Leverage the Run Script method to read the Windows Triage results, and format a message to update the Incident:

   Enter the following inputs and click **Submit**.
   Name: **Format Message for Incident**
   Script:

```
//put the scratchpad triage_results JSON into a shorter variable
var triage_results = new JSONParser().parse(
workflow.scratchpad.triage_results);
//capture the distance returned by the Google Maps call
var distance = data.get(n).distance;
//create a incident_comments variable
workflow.scratchpad.incident_comments = "Distance for Dispatch: " + distance
+
    "CPU Utilization: " + triage_results.CPU_Utilization + "\n" +
    "Memory Utilization: " + triage_results.Memory_Utilization + "\n" +
    "Disk Utilization: " + triage_results.Disk_Utilization + "\n" +
    "Failed Services: " + triage_results.Failed_Services;
```

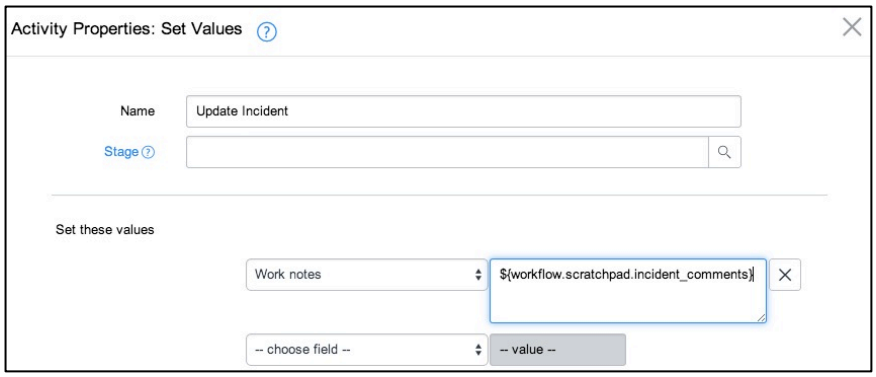
   Where **n** in data.get(n).distance is the output from the databus for the Google Maps step.

10. From the **Core** Activity navigator, locate and drag and drop **Core Activities > Utilities > Set Values** to the workflow.

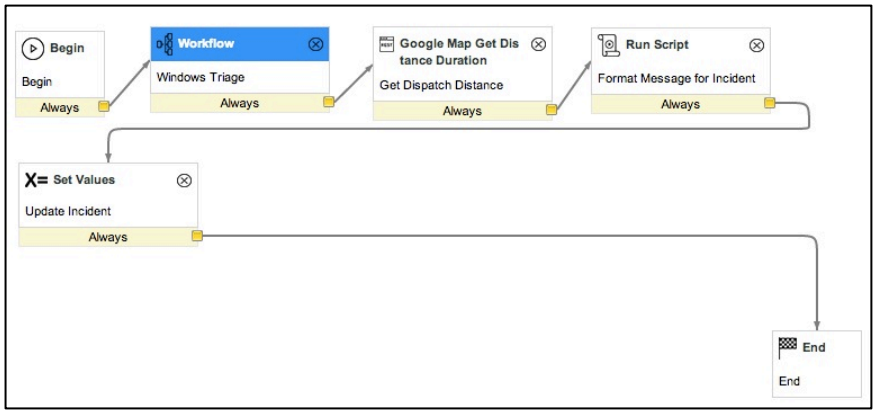
©2015 ServiceNow, Inc. All rights reserved.

11. Enter the following inputs and click **Submit**:

    Name: **Update Incident**
    Set these values: **Work notes = ${workflow.scratchpad.incident_comments}**

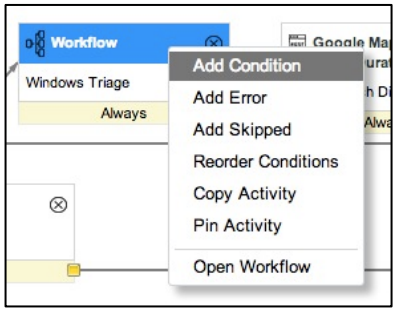


The workflow should look similar to the example.



©2015 ServiceNow, Inc. All rights reserved.

12. **Bonus**, add a condition to the Windows Triage step.
    Right-click the **Windows Triage** title bar, choose **Add Condition**



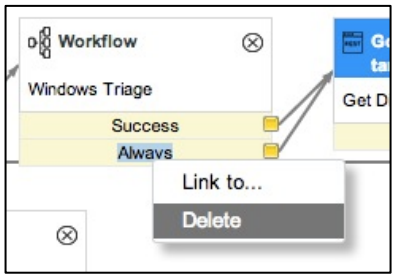
A **New Workflow Condition** form appears.

Bonus: Enter the following inputs:
Name: **Success**
Else: **Unchecked**
Condition: **JSUtil.notNil(workflow.scratchpad.triage_results) == true**



©2015 ServiceNow, Inc. All rights reserved.

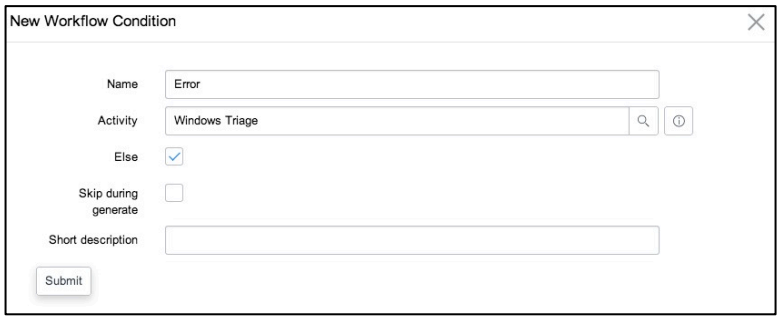
Bonus: Right click the **Always** result, and choose **Delete**.



Bonus: Add an Error condition with the following inputs, then wire as follows:

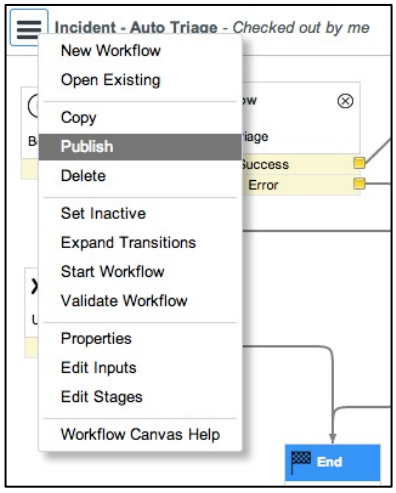
Name**: Error**

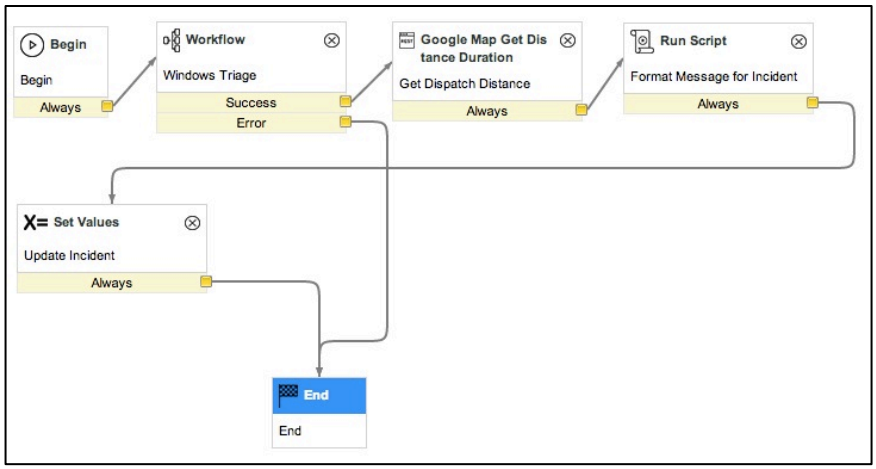
Else**: Checked**



®2015 ServiceNow, Inc. All rights reserved.

13. Publish the workflow so that it executes when an Incident meets the workflow conditions. Click the menu on the top left of the canvas, and select **Publish**.



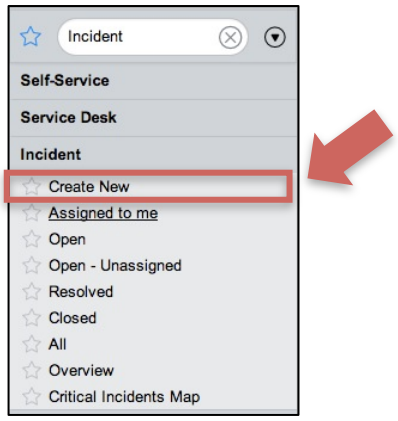


©2015 ServiceNow, Inc. All rights reserved.

Lab 9.2: Lab Verification

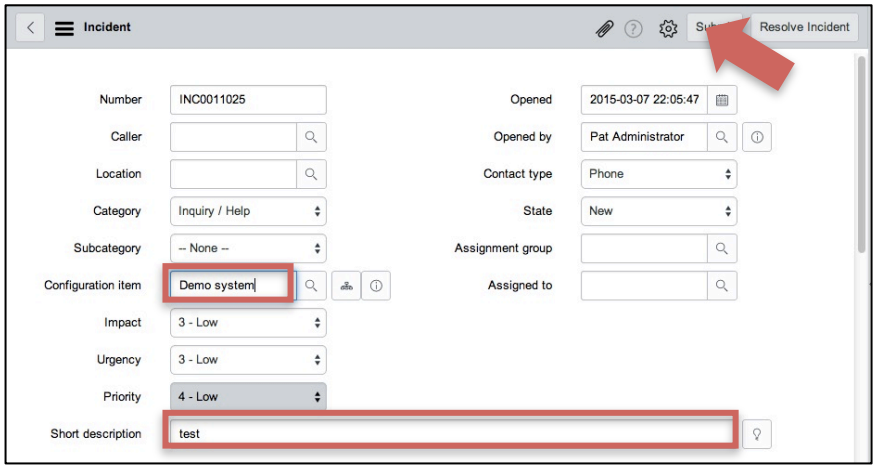
This lab explains how to create an Incident that triggers the above workflow.

1. Navigate to **Incident > Create New**.



2. Enter the following values and click **Submit**.

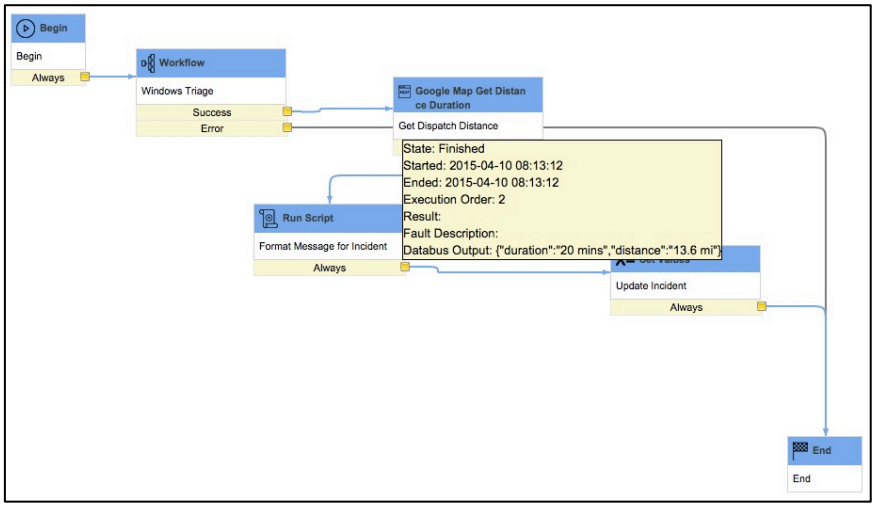
   Short Description: **Test**
   Configuration Item: **Demo system**

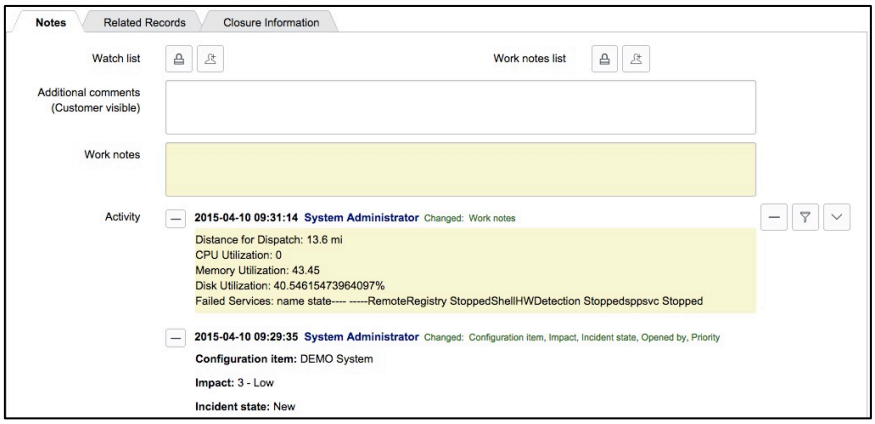


3. Navigate to **Workflow > All Contexts** and open the latest workflow execution **for Incident – Windows Triage**.

©2015 ServiceNow, Inc. All rights reserved.

4. Click the **Show Workflow** UI action and review the workflow execution.





Congratulations! You have completed a new workflow leveraging a subflow. You have learned how to:

- Call a subflow
- Leverage current record field values
- Update the current record with outputs from your activities

©2015 ServiceNow, Inc. All rights reserved.

Useful Links

Workflow Activity Designer
http://wiki.servicenow.com/index.php?title=Workflow_Activity_Designer

Activity Designer – PowerShell
http://wiki.servicenow.com/index.php?title=Activity_Designer_-_Powershell

Activity Designer – REST Web Services
http://wiki.servicenow.com/index.php?title=Activity_Designer_-_REST_Web_Service

Activity Designer – Probe
http://wiki.servicenow.com/index.php?title=Activity_Designer_-_Probe

Managing Activity Designer Variables
http://wiki.servicenow.com/index.php?title=Managing_Activity_Designer_Variables

Workflow Activity Parsing Rules
http://wiki.servicenow.com/index.php?title=Workflow_Activity_Parsing_Rules

Condition Activities
http://wiki.servicenow.com/index.php?title=Condition_Activities

Workflow Concepts
http://wiki.servicenow.com/index.php?title=Workflow_Concepts

Scripting with PowerShell
https://technet.microsoft.com/library/bb978526.aspx

Library of various Web APIs
http://www.programmableweb.com/category/all/apis?data_format=21190

©2015 ServiceNow, Inc. All rights reserved.