

Increasing VMFS capacity with vRealize Orchestrator







I am working on adding some functionality to the FlashArray workflow package for vRealize Orchestrator and one of those features is automating the process to increase the capacity of a [VMFS volume](#). The FlashArray portion of that is pretty straight forward but what of the VMFS portion? Not much on the internet directly about this. Luckily using the vCenter SDK with vRO, this is pretty easy.

There is really only one piece of information that you need a user to specify, and that is the datastore that is to be re-sized. From that you can figure everything else out.

vRealize Orchestrator uses Javascript as a scripting language so all of this will be done with that. A **great** resource for the vSphere API specifically in context to vRO is <https://www.vroapi.com/>.

When you have a datastore passed into the workflow, it is stored in memory to be referenced by the Javascript as a custom vCenter Datastore object (information can be found [here](#)).

Looking at the spec for that object, you can see the various information it stores/references. One of those being the hosts that see the datastore. The expand operation is done by a ESXi host, so we need to choose one. For my purposes I just choose the first in that array of hosts.

 effectiveRole	 Number[]	Access rights the current session has to this er
 host	 VcDatastoreHostMount[]	Hosts attached to this datastore.
 id	 String	returns the id of this ManagedObject

We want the vcHostSystem object, but the host array is not an array of those, it is an array of keys that then contains (points to) those objects.

Plugins > VC 6.0.2 (vCenter Server) > VcDatastoreHostMount >

Class VcDatastoreHostMount

Host-specific datastore information.

Attributes

Name	Type	Description
 key	 VcHostSystem	The host associated with this datastore.
 mountInfo	 VcHostMountInfo	Host-specific information about the mount.

So each object in the array will have an attribute “key” which points to the vcHostSystem (the ESXi server)

and the mountinfo. We want the former. To store the first one in a variable I have my first line of Javascript as so:

```
[crayon-6427865cc25cb669695287/]
```

Replace 0 with 1 to get the second host, etc. etc.

So I stored it in a variable esxihost. If you need to rescan that host you can pass it to the vCenter vRO module for rescanning:

```
[crayon-6427865cc25d3939630129/]
```

Now you need to get the information for the new partition on the device. ESXi has a handy method for this called queryVmfsDatastoreExpandOptions. When run against a datastore object, it will return the options you have for expanding that datastore. Either creating a new partition and adding it as a new extent (no thank you!) or expanding the current partition to consume it (yes!). This will give you the partition information that is needed for the expansion operation. See some information about this method here:

<https://www.vroapi.com/Method/VC/6.0.2/VcHostDatastoreSystem/queryVmfsDatastoreExpandOptions>

All you need to do to run this is pass in the datastore object. queryVmfsDatastoreExpandOption is a method available from the "datastore subsystem" of the ESXi host. We could run this in one command from what we have like so:

```
[crayon-6427865cc25d4770580564/]
```

but for readability, I prefer to make this into two lines, by storing the datastoreSystem in a variable first and calling the method directly from it:

```
[crayon-6427865cc25d6768028287/]
```

I store the partition information in a variable called expandSpec. Looking at the spec for this method, you can see it returns the following:

Plugins > VC 6.0.2 (vCenter Server) > VcHostDatastoreSystem > queryVmfsDatastoreExpandOptions

Method queryVmfsDatastoreExpandOptions(VcDatastore datastore)

Queries for options for increasing the capacity of an existing VMFS datastore by expanding (increasing the size of

Returns

 VcVmfsDatastoreOption[]

Arguments

Name	Type	Description
datastore	 VcDatastore	The datastore to be expanded.

vcVmfsDatastoreOption. This is the custom object it returns that has all of the information we need. So how about running the expansion? We have all of the information we need. The expansion method is called

expandVmfsDatastore. Simple enough. We can see the SDK info here:

<https://www.vroapi.com/Method/VC/6.0.2/VcHostDatastoreSystem/expandVmfsDatastore>

It takes in two things, the datastore and something called vcVmfsDatastoreExpandSpec. So the second one is new, that isn't what is stored in my expandSpec variable, that is of type vcVmfsDatastoreOption. So how to we get vcVmfsDatastoreExpandSpec? Well it is part of vcVmfsDatastoreOption!

<https://www.vroapi.com/Class/VC/6.0.2/VcVmfsDatastoreOption>

Plugins > VC 6.0.2 (vCenter Server) >
VcVmfsDatastoreOption >

Search

Class VcVmfsDatastoreOption

VMFS datastore provisioning option that can be applied on a disk. VMFS datastores can be created or have their capacity increased using storage from disk. There are often multiple ways in which extents can be allocated on a disk. Each instance of this structure represents one of the possible options that can be applied to provisioning VMFS datastore storage. Only options that follow ESX Server best practice guidelines will be presented.

Attributes

Name	Type	Description
info	VcVmfsDatastoreBaseOption	Information about this VMFS datastore provisioning option. This structure describes the extent all
spec	VcVmfsDatastoreSpec	Specification to create or increase the capacity of a VMFS datastore. This property contains a confi

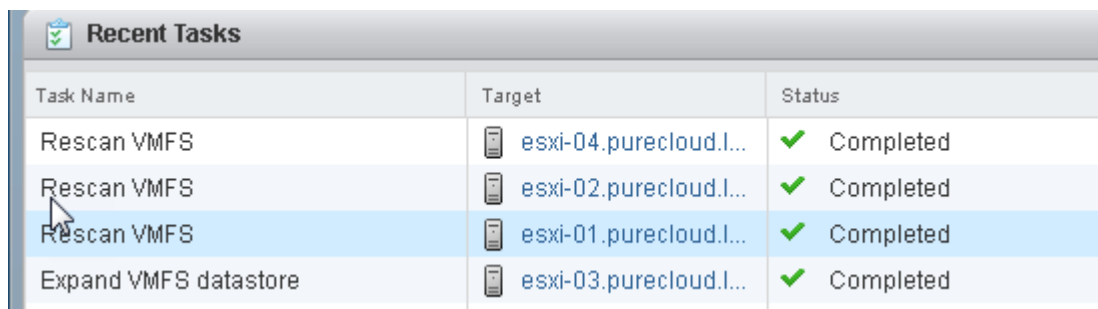
So our variable called expandSpec contains this! So instead of passing the whole thing we specify the right part. So expandSpec.spec, which is the name of the location where the object of type vcVmfsDatastoreExpandSpec is stored. We can re-use our variable hosting the storage system, because that is also where the expandVmfsDatastore method is called from.

[crayon-6427865cc25d7355042451/]

You will note that I had to call expandSpec as an array (using [0]). This is because it is an array! This is very easy to miss, because 99.99% of the time this will be an array with only one top-level object. So if you just use expandSpec.spec you will get "undefined" errors. Took me quite some time to catch.

That will expand our datastore. Done! If this host is in a vCenter cluster you will see it rescan the other hosts automatically to see the new VMFS size. If you have hosts elsewhere that see the datastore, you might want to add logic to rescan those too.

In vCenter you will see the "Expand VMFS Datastore" event complete.

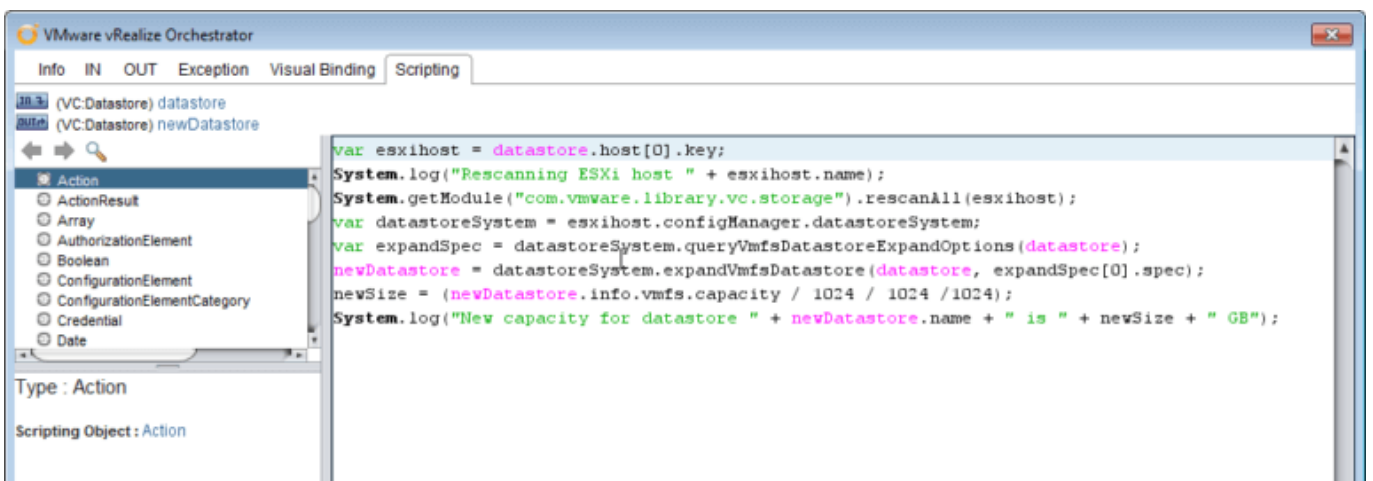
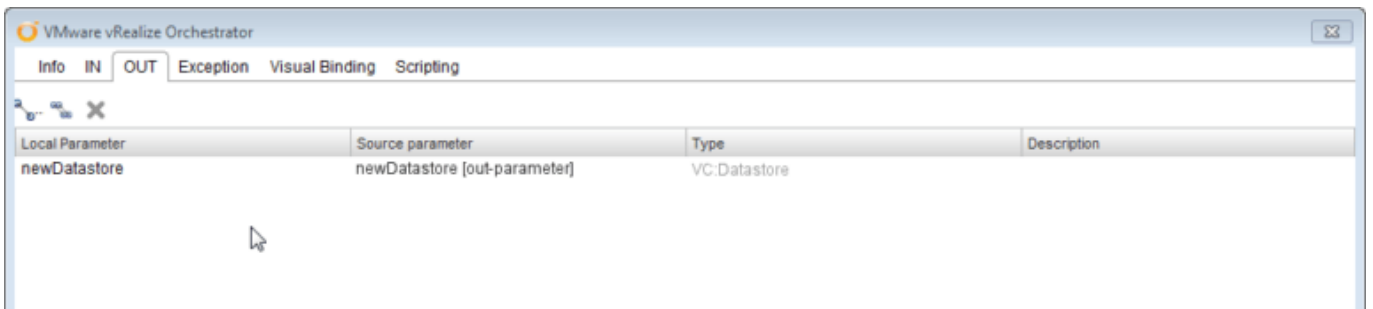
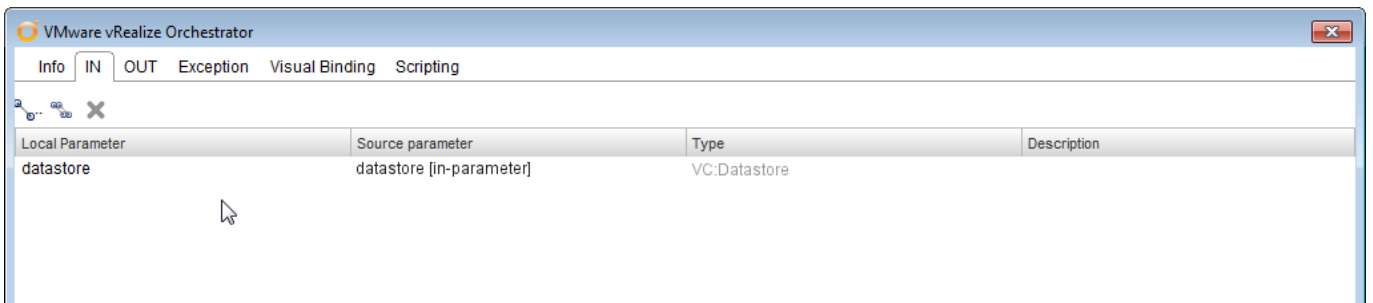


Task Name	Target	Status
Rescan VMFS	esxi-04.purecloud.l...	Completed
Rescan VMFS	esxi-02.purecloud.l...	Completed
Rescan VMFS	esxi-01.purecloud.l...	Completed
Expand VMFS datastore	esxi-03.purecloud.l...	Completed

All of the Javascript that I used to do this is here. Some extra logging mainly and I am outputting the new datastore object out in a variable named newDatastore. No error handling right now, to make it easier to understand.

[crayon-6427865cc25d8991361420/]

The workflow object looks like so. Only really three screens of changes. One input (the datastore), one output (the same datastore but with updated size) and the script:

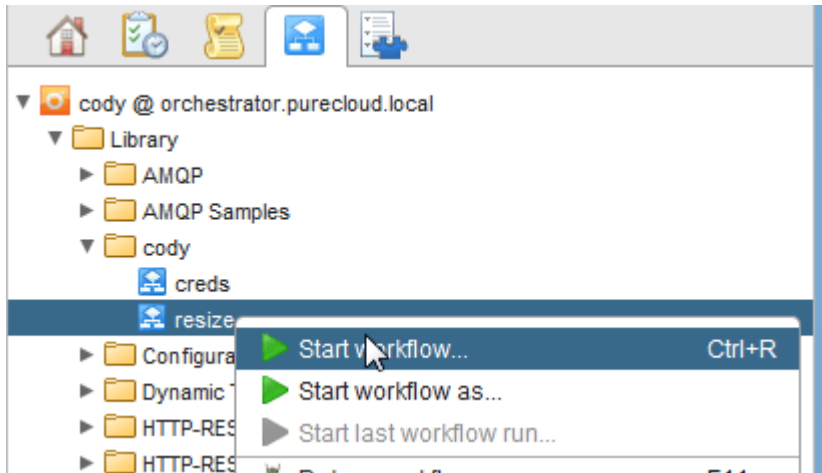


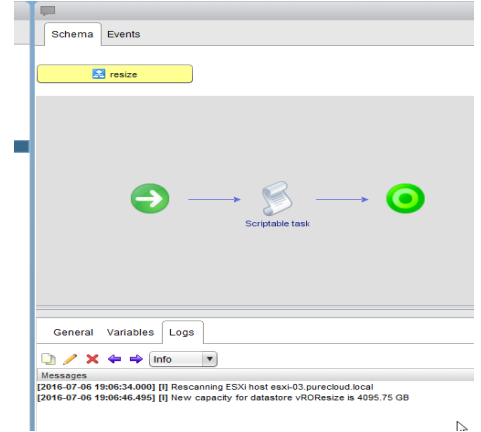
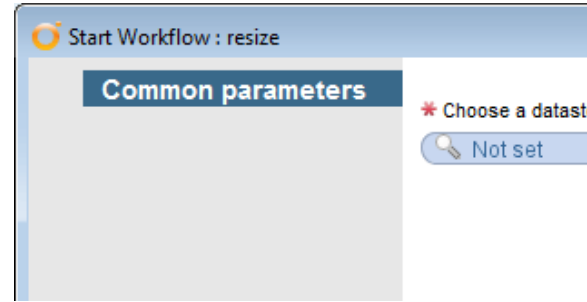
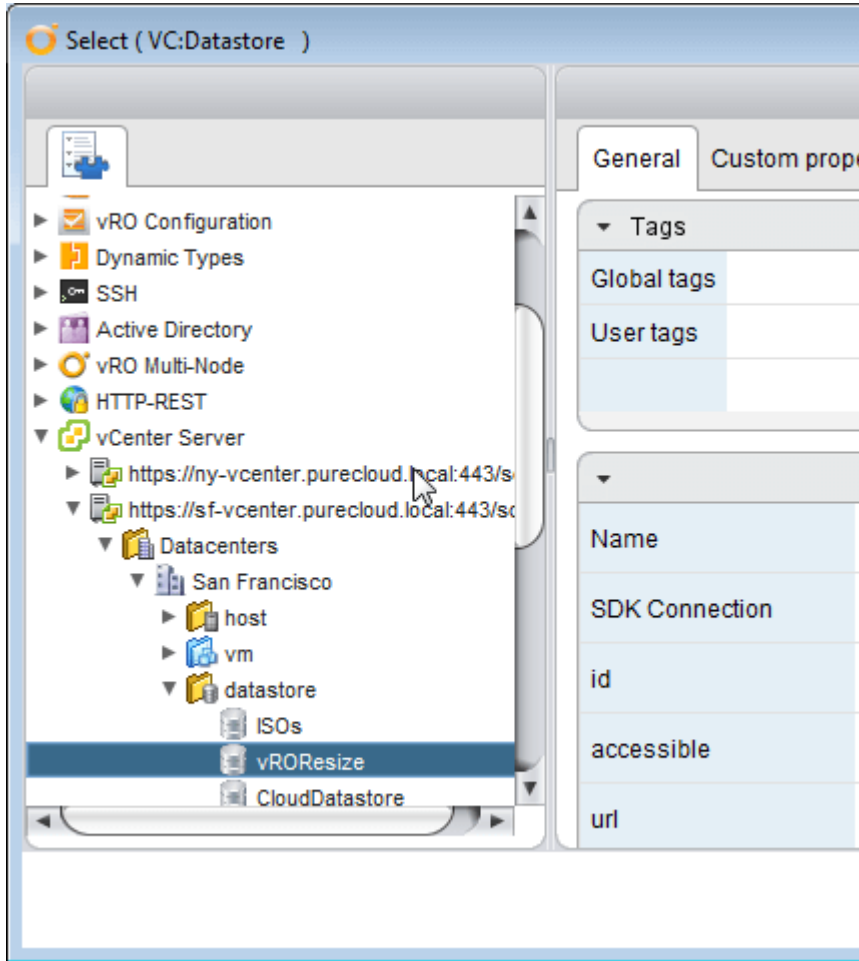
This workflow does the following:

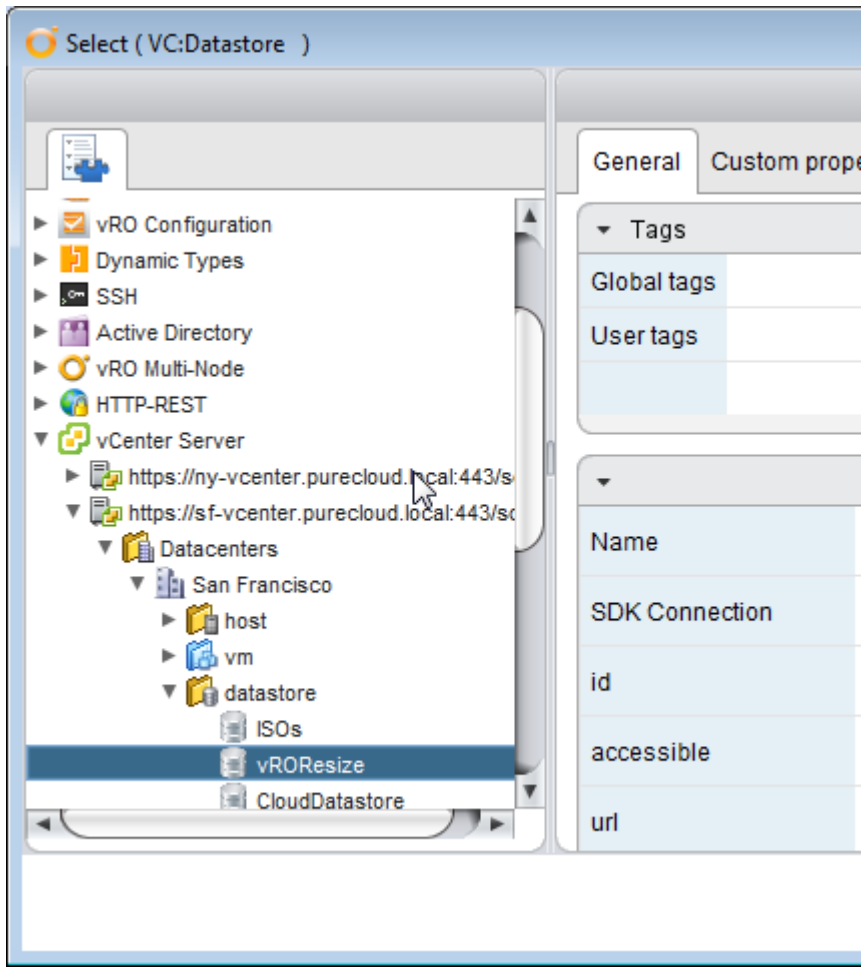
1. Takes in a datastore
2. Finds the new capacity information (this assumes the physical device has already been increased)

- in capacity)
3. Expands the VMFS partition to use it all
 4. Rescans the cluster for the host used to expand
 5. Returns the new datastore object. Logs the host used and the new size.

So if we were to run this:







Schema Events

resize



General Variables Logs

Info

Messages

[2016-07-06 19:06:34.000] [I] Rescanning ESXi host esxi-03.purecloud.local
[2016-07-06 19:06:46.495] [I] New capacity for datastore vROResize is 4095.75 GB