Importing a VVol Snapshot with PowerCLI

Here’s a quick walkthrough of importing a VVol snapshot with PowerCLI.

First, enjoy a poorly photoshopped Back To the Future reference:

First, some background. In a VVol world, when you use VMware to take a snapshot of a VM, it doesn't create that delta VMDK that you are used to. Instead, it actually creates array-based snapshots. Here is a great post from Cormac that explains this concept.

So, creating and restoring from array based snapshots can now be managed inside of vCenter. But what if you want to take that snapshot and present it to another VM? A test/dev scenario of a database for instance?

Well there is an easy way to do this. There is a method called importUnmanagedSnapshot under the virtualDiskManager in vCenter that allows you to do this.

This method takes in three parameters:

1. **Virtual disk path**. So datastore in brackets [mydatastore] and then the path folder/folder/mydisk.vmdk
2. **Datacenter.** If you are connecting to a single ESXi host this is optional, but otherwise you need to specify this. The SDK call needs to MoRef (managed object reference) of the datacenter
3. **VVol ID.** vCenter needs to know the actual VVol you want to import and you specify this via the VVol ID, which is a unique identifier for a given VVol.

So first, how doe call this from PowerCLI. As far as I know, there is no built-in cmdlet to do this, so we need to use the trusty get-view. This method is under the virtual disk manager, so once you connect to your vCenter, you need to pull that in like so:
[crayon-642790d3f196e306383433/]
Then you can actually called the method. But first we need to populate the parameters. The first is you virtual disk path. This needs to be a virtual disk that does not exist. What this method does is basically:

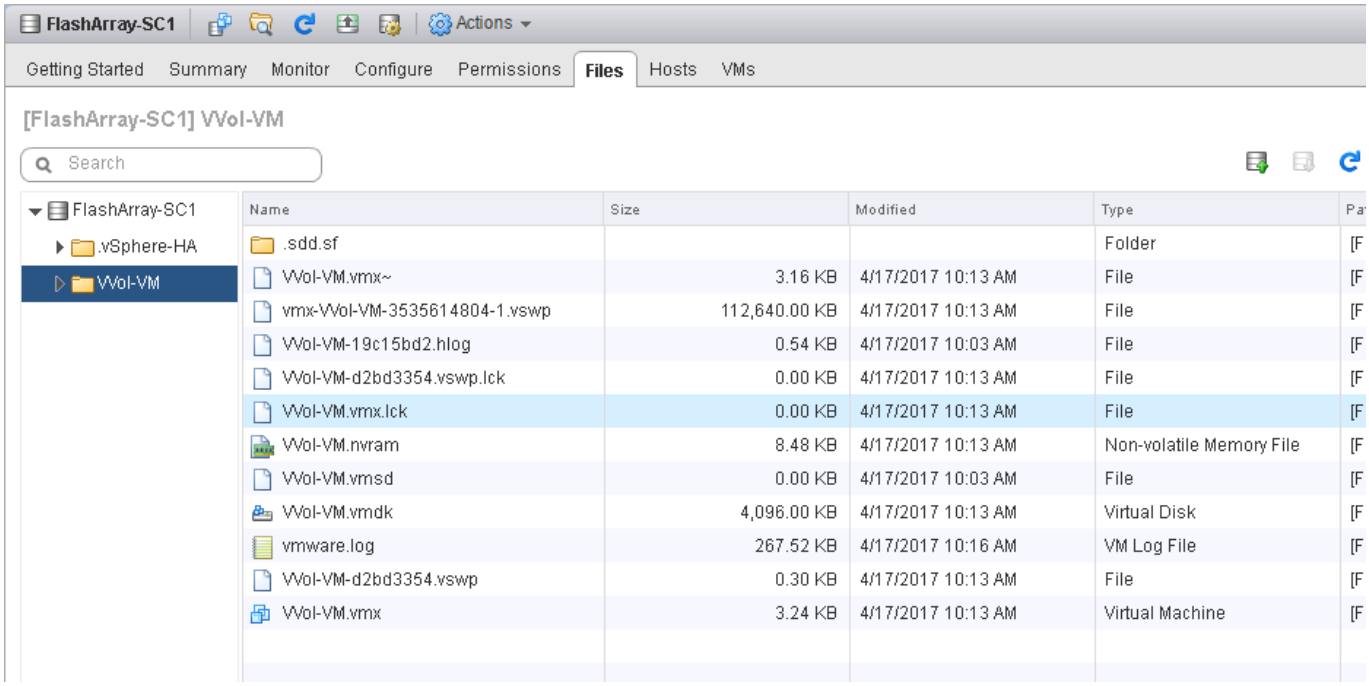
- Imports that VVol into the VASA database
- Binds the VVol to the appropriate Protocol Endpoint
- Creates a new VMDK pointer file which references that VVol

This then allows you to add that VMDK which contains the point-in-time of that VVol to a VM.

So back to the disk path. A disk path looks like so:

[mydatastore] folder/folder/mydisk.vmdk

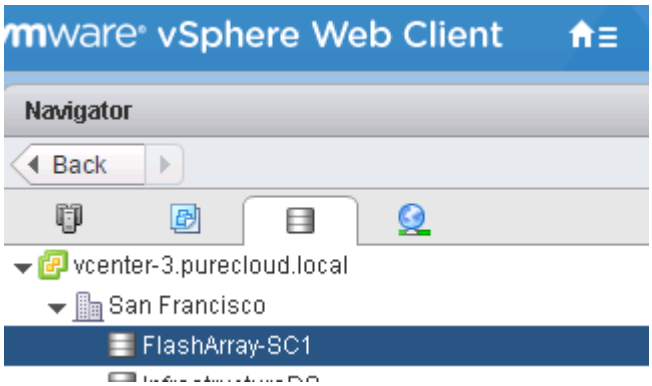
In the brackets is the VVol datastore (a storage container) and the rest are folders and the virtual disk file name. The VMDK should not exist yet–otherwise it will fail. You can put it on any storage container from the array this VVol is on. But it has to be a storage container, it cannot be VMFS, for instance. Furthermore it can be in any folder in there, but ideally you probably want to put it in the folder of the VM you plan on adding it to. I am going to add it to a VM called VVol-VM on a storage container called FlashArray-SC1:



The vmdk filename can be anything that is just unique in that directory, so I will just called it "importedvvol.vmdk". Making my path:

[FlashArray-SC1] VVol-VM/importedvvol.vmdk

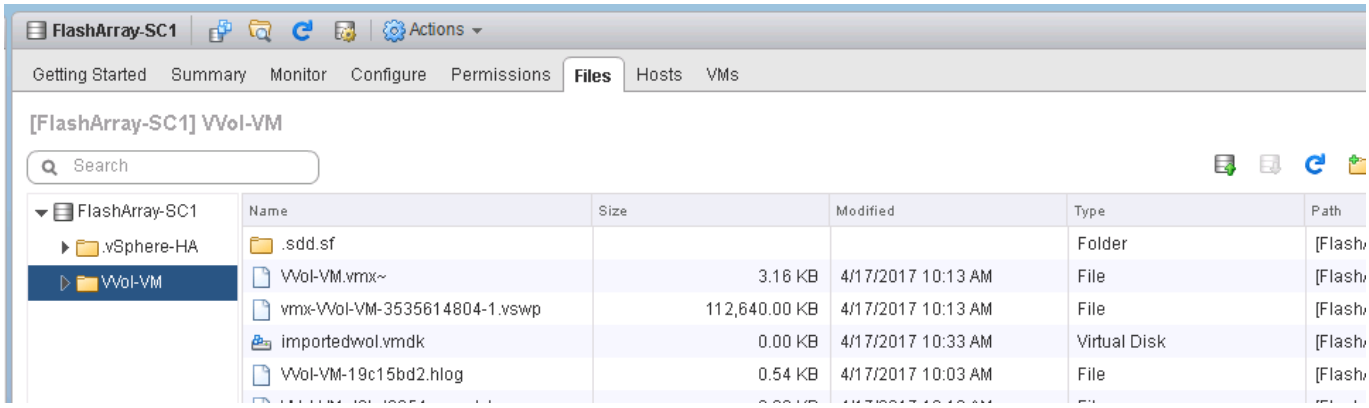
So I will store that in a new variable $vvolpath:
[crayon-642790d3f1977818022517/]
Great. Now the next step is my datacenter. This can be easily pulled in via the cmdlet, get-datacenter <name>.



Mine is called “San Francisco” so I will run:
[crayon-642790d3f1978535210834/]
Lastly is my VVolID. I will get into the particulars for the FlashArray at a later date (still gathering feedback from beta so things might change by GA so not worth documenting just yet), but refer to your vendor on how to do that. My VVol ID is “rfc4122.bebdbe84-64f5-4c4b-9f50-13998ece64dd”. Which I have stored in a variable called $uuid.

One last thing to note, I do not enter the entire $dc object in for the datacenter, it only wants the MoRef, so that needs to be specified. So instead it will be:
[crayon-642790d3f197c204477389/]
So now I can make the call! Parameters in order: vvol path, datacenter MoRef, then VVol ID.
[crayon-642790d3f197d241591042/]

```
PowerCLI C:\> $vvolpath = "[FlashArray-SC1] VVol-VM/importedvvol.vmdk"
PowerCLI C:\> $virtualDiskManager.ImportUnmanagedSnapshot($vvolpath,$dc.ExtensionData.MoRef,$uuid)
PowerCLI C:\>
```



Now I can add the vmdk to a VM via PowerCLI:
[crayon-642790d3f197e977216926/]
Or of course through the GUI or whatever.

Look for a lot more VVol-content to come!