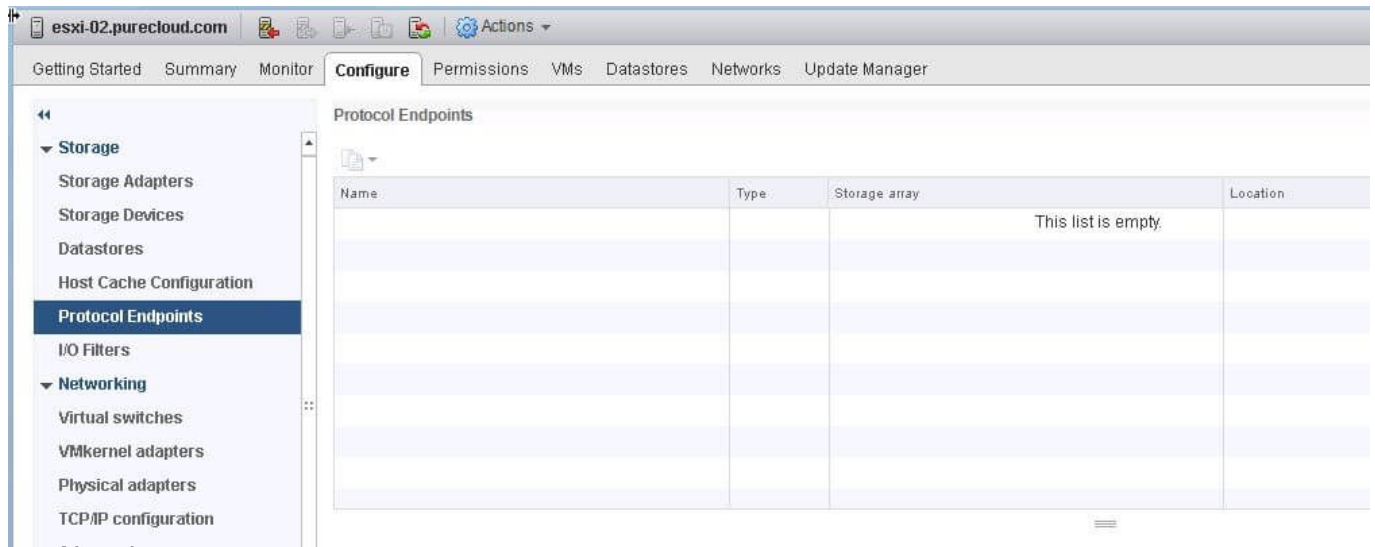


# Virtual Volumes: VVol Bindings Explained



This change is necessitated for two reasons:

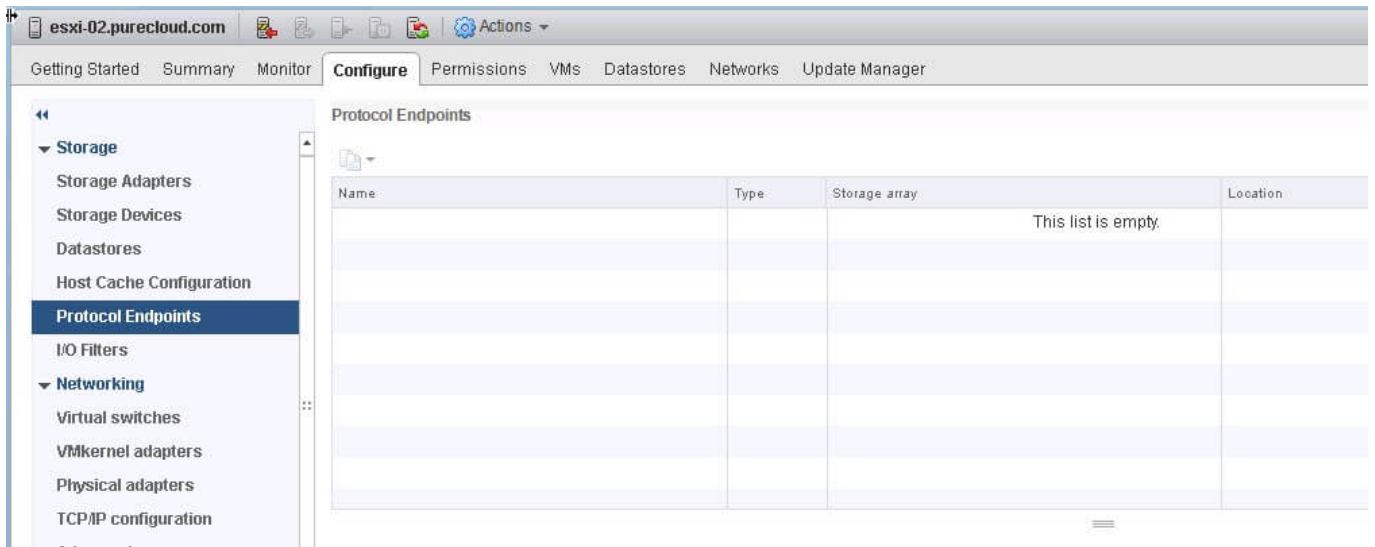
- **Scale.** Traditional ESXi SCSI limits how many SCSI devices can be seen at once. 256 in 6.0 and earlier and 512 in 6.5. This is still not enough when every virtual disk its own volume.
- **Performance.** Virtual Volumes are provisioned and de-provisioned and moved and accessed constantly. If every time one of these operations occurred a SC SI rescan was required, we would see rescan storms unlike this world has ever witnessed.

So VMware changed how this is done.

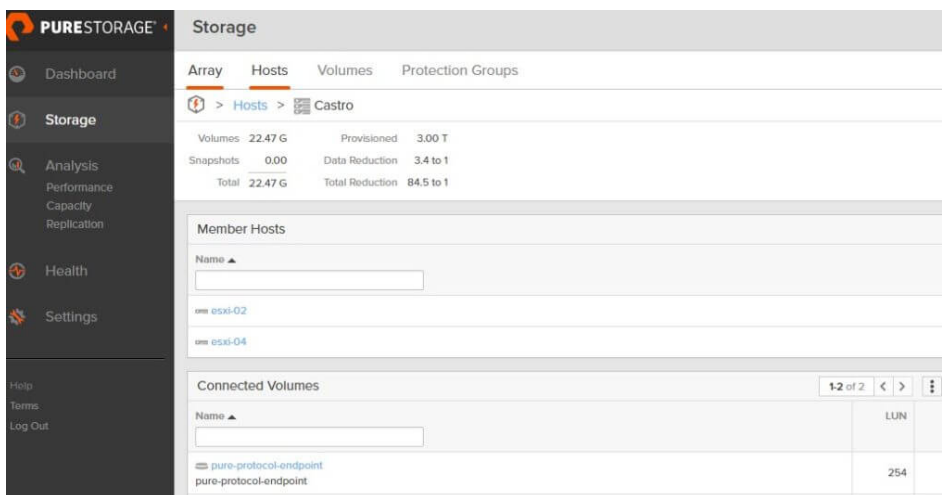
## Protocol Endpoints

A fancy name for a simple thing. This is a special device on an array that has special VPD information that says that it is a protocol endpoint. This is of zero (or minimal) capacity and is the only volume you need to manually provision to an ESXi host. Connect it on your array, (however that is done on your array) and then rescan the ESXi hosts.

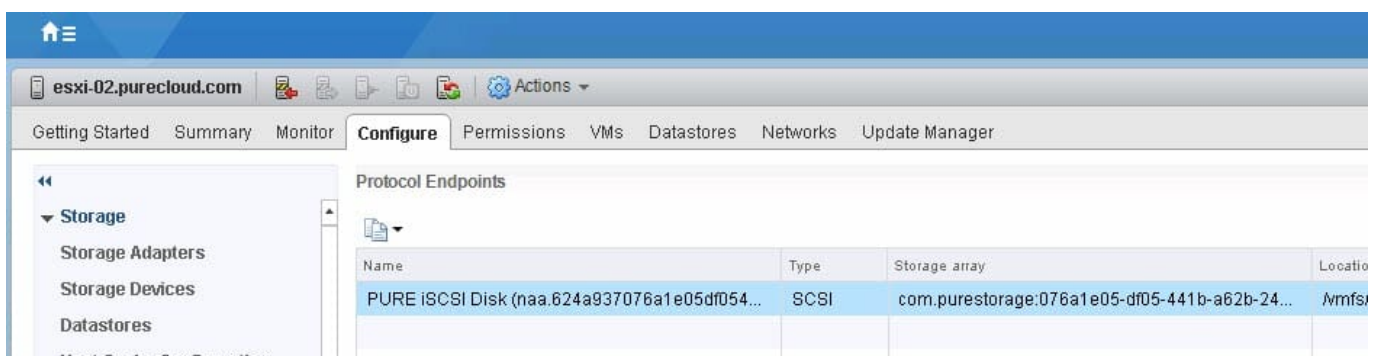
Below I have an ESXi host, which currently has no protocol endpoints



Then from my array I add my PE:



Then rescan your ESXi host. Now, you might notice that even though the PE is presented it does not show up as a PE in the “protocol endpoints” host view in the GUI until a storage container (a VVol datastore) is actually mounted and using it.



Even though the vSphere Web Client doesn't report the PEs that are currently un-used, ESXi does actually see it as a PE if you use the CLI:

[crayon-64278cd1a5af9173548549/]

Frankly, I am not sure if this is a Web Client feature or a bug. It is useful I guess in the sense that it shows the PEs that are actually in use by that particular host. So my guess is the former option.

# WVol Bindings

So now that we have a protocol endpoint, how are volumes connected? Well when a virtual machine is created or a new virtual disk is added to a VM (that is configured to be a virtual volume), one or more virtual volumes are created. Every VM has a config VVol (holds the configuration information), and every virtual disk is one virtual volume.

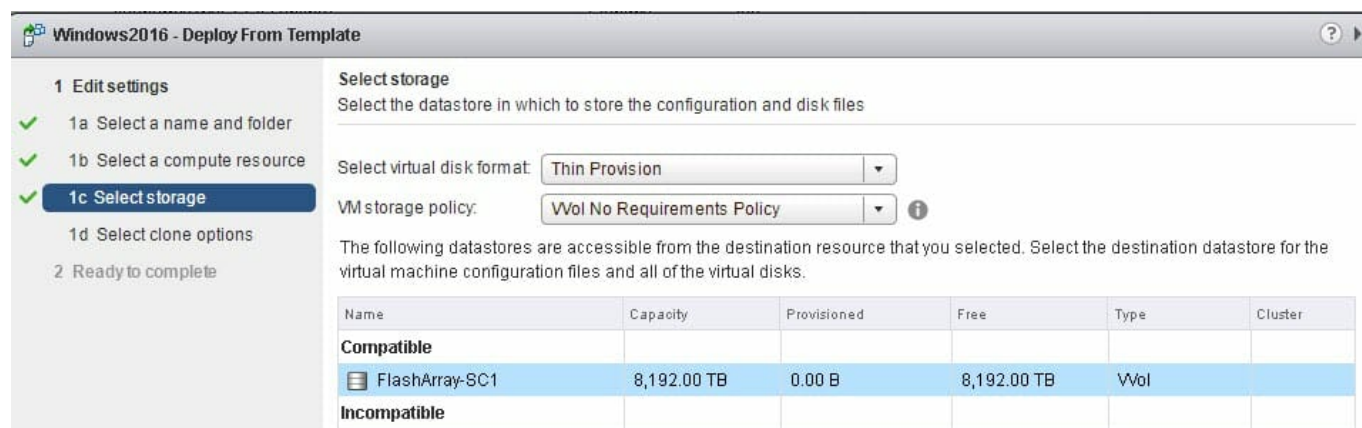
When you create a new virtual volume and it needs to be accessed by a VM, the VVol is bound to a protocol endpoint. What “bound” means is that it is presented to a host as a sub-lun through that protocol endpoint. A sub-lun is a unique connection of a VVol through a PE to a host—in other words, even if a PE is shared, it does not mean that all hosts that see the PE can see the VVol.

A VVol binding goes through the PE, so if the PE is LUN ID 255 for a given host and the VVol is sub-lun 10, the VVol LUN ID for that VVol for that host is 255:10.

The important thing to note is that when VVols are created they are not automatically connected to hosts and stay that way—they are connected only when needed, and disconnected automatically when they no longer are. This is one of the nice points of flexibility/automation that comes with VVols. So the question is, when are they bound?

## Creating a VM

At this point though there are no VVols. So let's provision a virtual machine.



The screenshot shows the 'Deploy From Template' wizard for Windows 2016. The '1c Select storage' step is active. The 'Select storage' section shows the 'FlashArray-SC1' datastore selected. The 'Select virtual disk format' is set to 'Thin Provision' and the 'VM storage policy' is 'VVol No Requirements Policy'. A table lists available datastores:

Name	Capacity	Provisioned	Free	Type	Cluster
<b>Compatible</b>					
FlashArray-SC1	8,192.00 TB	0.00 B	8,192.00 TB	VVol	
<b>Incompatible</b>					

This will then go and create the VM and the respective VVols on the array. This VM has only one virtual disk, so only one data VVol will be created. Every new VM also gets a config VVol when the VM is created, which holds the configuration information of the VM itself (virtual hardware etc). I will go into more detail on what a config VVol actually is in a later blog post.

**PURE STORAGE** Storage

Dashboard | Storage | Analysis | Health | Settings

Array | Hosts | **Volumes** | Protection Groups

> Volumes > vvol-myVVolVM-vg

Volumes	24.16 G	Provisioned	19.20 T
Snapshots	420.04 M	Data Reduction	2.6 to 1
Total	24.57 G	Total Reduction	>100 to 1

**Volumes**

Name ▲

- vvol-myVVolVM-vg/Config-d967b3d9
- vvol-myVVolVM-vg/Data-e7b81434

VVols do get bound when the VM is created, but only shortly.

As we can see in the audit log, my two VVols (one config, one data) were connected briefly and then disconnected:

Flag	Time	User	Command	Subcommand	Name	Arguments
in	2017-09-28 11:10	root	purehost	disconnect	esxi-02	--vol vvol-myVVolVM-vg/Data-05c7795
in	2017-09-28 11:10	root	purehost	disconnect	esxi-02	--vol vvol-myVVolVM-vg/Config-03ca033a
in	2017-09-28 11:09	root	purevol	connect		--protocol-endpoint pure-protocol-endpoint --host esxi-02 vvol-myVVolVM-vg/Data-05c7795
in	2017-09-28 11:09	root	purevol	connect		--protocol-endpoint pure-protocol-endpoint --host esxi-02 vvol-myVVolVM-vg/Config-03ca033a

The config VVol has no remaining binding once the VM has been created.

**Storage**

Array | Hosts | **Volumes** | Protection Groups

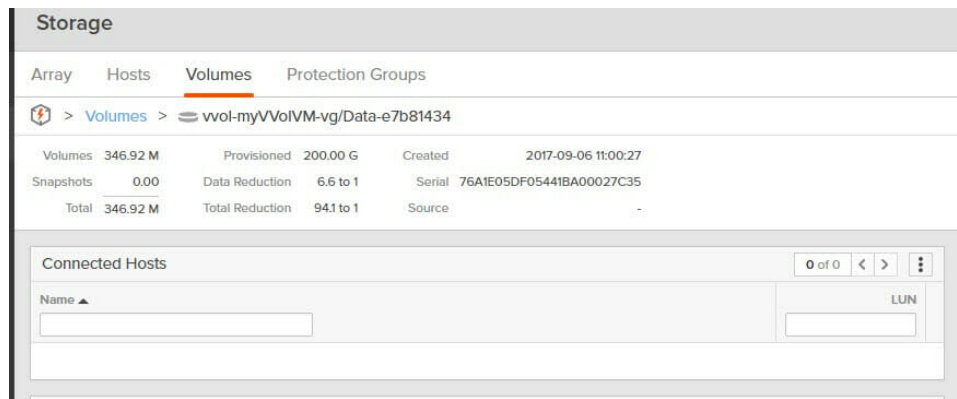
> Volumes > vvol-myVVolVM-vg/Config-d967b3d9

Volumes	0.00	Provisioned	4.00 G	Created	2017-09-06 11:00:26
Snapshots	0.00	Data Reduction	7.6 to 1	Serial	76A1E05DF05441BA00027C34
Total	0.00	Total Reduction	>100 to 1	Source	-

**Connected Hosts** 0 of 0

Name ▲

LUN



Neither does the data VVol.

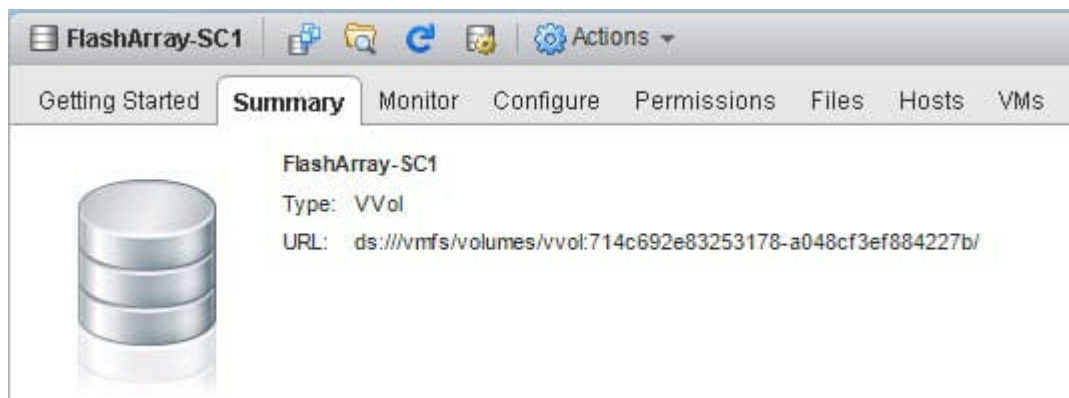
This makes sense, neither objects need to be read by any host at this point. This will stay this way unless you do something to the VM or one of its objects need to be accessed.

# Reading the Config VVol

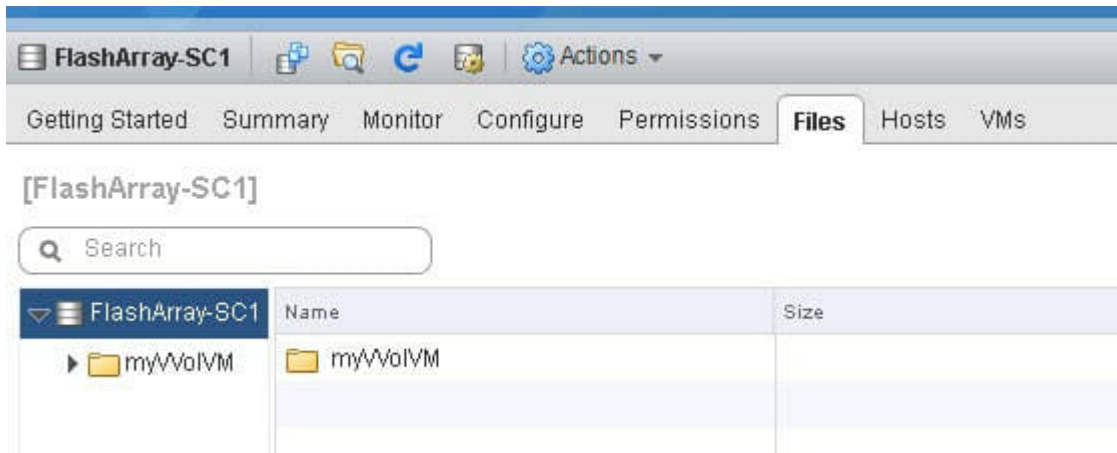
Well, what about if you need to look at the config information of a VM even when it isn't powered-on? What happens then? Well first off, a config VVol is basically a mini-VMFS that is hidden directly from view. A config VVol is associated with a VVol datastore and can then be read when you navigate the VVol datastore. Just like VMFS. The main difference is that a VVol datastore is not a physical object, instead it is just an abstraction of VVol pointers and config VVol file systems.

## Datastore browser

So, I have a VVol datastore:



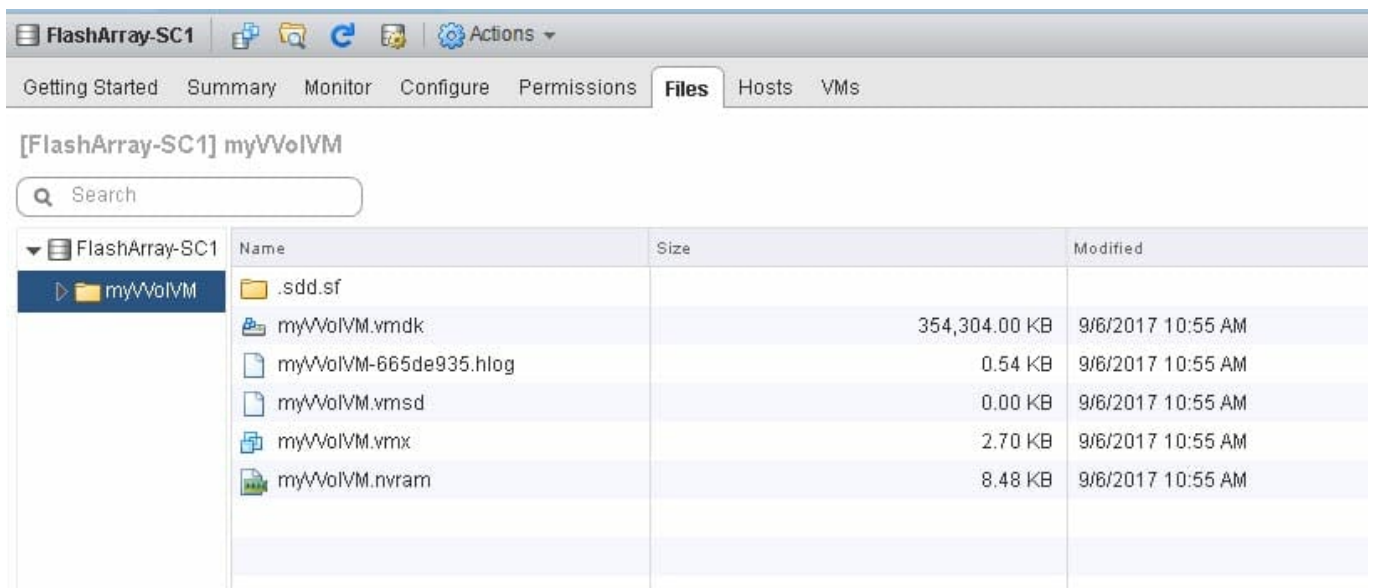
If I click on files, it shows all of my VM folders (I just have one).



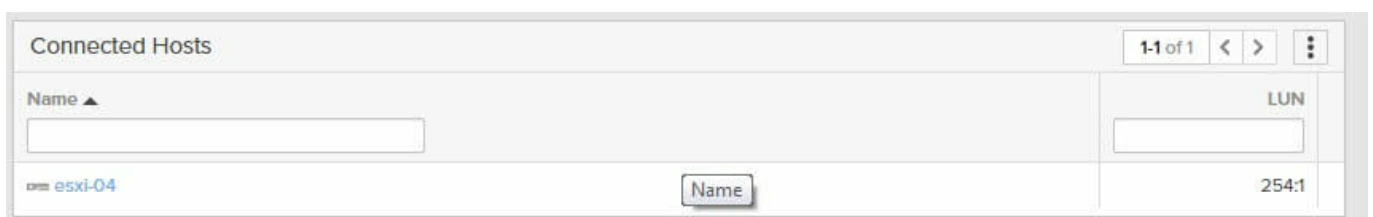
A folder is essentially a reference to a config VVol. The config VVol is still unbound.



If I click on the folder:

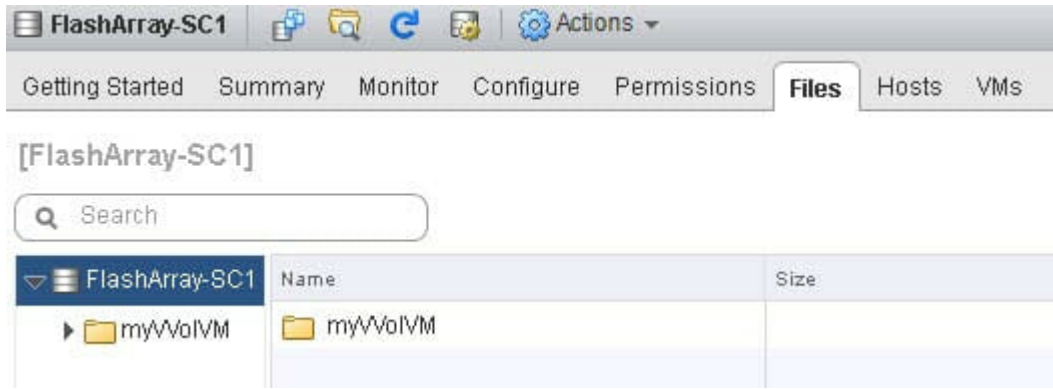


It is now bound:



It is only bound to one host, so vCenter chooses a host to actually leverage to query the actual VVol datastore, this is no different than with VMFS conceptually. As soon as you back out of it, and there is no

“handle” open to the VVol datastore the binding is removed. The config VVol is no longer needed.



We can prove this out with the array audit logs:

Time	User	Command	Subcommand	Name	Arguments
2017-09-06 04:20	root	purehost	disconnect	esxi-04	--vol vvol-myVVolVM.vg/Config-d967b3d9
2017-09-06 04:19	root	purevol	connect		--protocol-endpoint pure-protocol-endpoint --host esxi-04 vvol-myVVolVM.vg/Config-d967b3d9

A binding through the protocol endpoint and then an unbind as soon as I closed the file browser (well almost immediately, within it seems 15-30 seconds).

## Shell change directory

A similar, but slightly different behavior can be exhibited if you SSH into an ESXi host and navigate to the VVol datastore and cd into the folder.

When I SSH into an ESXi host and then CD into /vmfs/volumes/<name of VVol datastore>, nothing is bound:

```
[root@esxi-04:/vmfs/volumes] cd FlashArray-SC1/  
[root@esxi-04:/vmfs/volumes/vvol:714c692e83253178-a048cf3ef884227b]
```

Name	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	

But if a run “ls” I will see all of my config VVols get bound:

```
[root@esxi-04:/vmfs/volumes/vvol:714c692e83253178-a048cf3ef884227b] ls  
myVVolVM rfc4122.eefbc60c-c093-4eaf-a5  
cfc4122.d88b29d0-c8b1-4f8c-aac9-2b2b0affb554 testvm2  
[root@esxi-04:/vmfs/volumes/vvol:714c692e83253178-a048cf3ef884227b]
```

Volumes 22.05 G      Provisioned 3.01 T  
 Snapshots 0.00      Data Reduction 3.6 to 1  
 Total 22.05 G      Total Reduction 90.9 to 1

Connected Volumes			
Name ▲	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	
vvol-myVVolVM-vg/Config-03ca033a	False	254:2	
vvol-testvm2-vg/Config-2b9ca8f6	False	254:1	

No data VVols. But after a short time, the config VVols are unbound automatically.

Connected Volumes			
Name ▲	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	

So it seems this behavior is slightly different between vCenter browse datastore and with shell. Browse datastore causes no bindings when you list the highest level of the VVol datastore directory, but shell binds all config VVols then unbinds them shortly after.

If I CD into a VM directory that config VVol is bound and stays that way until I CD back out.

Pretty dynamic—really nice as it makes sure only things are presented that are actually needed.

## Edit VM settings

If I log into vCenter and click edit VM settings when the VM is off, I see that no bindings occur. This is due to the fact that vCenter doesn't need to access the VMX file, it looks at what is in memory.

Though, if I make a change I will see a bind then an unbind. This is because you made a change and it needs to commit that change to the VMX file on the config VVol. It uses the host that the VM is registered to for the binding operations.

Command	Subcommand	Name	Arguments
purehost	disconnect	esxi-02	--vol vvol-myVVolVM-vg/Config-03ca033a
purevol	connect		--protocol-endpoint pure-protocol-endpoint --host esxi-02 vvol-myVVolVM-vg/Config-03ca033a



# Power-On a VM

The next step is powering a VM on. What gets bound?

Well everything. The config VVol, the data VVols and also a swap VVol gets created which is bound to the host that runs it.

Powered-off, no bindings:

Hosts > Castro > esxi-02

Volumes 22.05 G      Provisioned 3.00 T  
Snapshots 0.00      Data Reduction 3.6 to 1  
Total 22.05 G      Total Reduction 90.7 to 1

Name	Shared	LUN
pure-protocol-endpoint pure-protocol-endpoint	True	254
Template	True	253

vmware vSphere Web Client

myVVolVM Summary

Guest OS: Microsoft Windows Server 2012 (64-bit)  
Compatibility: ESXi 6.5 and later (VM version 13)  
VMware Tools: Not running, version:10272 (Current)  
DNS Name:  
IP Addresses:  
Host: esxi-02.purecloud.com

Power-on on ESXi-02:

All of the VMs VVols are bound to ESXi-02:

Name	Shared	LUN
pure-protocol-endpoint pure-protocol-endpoint	True	254
Template	True	253
vvol-myVVolVM-vg/Config-03ca033a	False	254:1
vvol-myVVolVM-vg/Data-05c77a5	False	254:3
vvol-myVVolVM-vg/Swap-16be3fd7	False	254:2

When you power-off the VM, all of the bindings go away (and the swap VVol is deleted entirely).

# vMotion

So what about if you move hosts?

Currently my VM is bound to esxi-04 (and running on it).

The screenshot shows the vSphere Web Client interface for host esxi-04. It displays storage statistics and a list of connected volumes.

Name	Shared	LUN
pure-protocol-endpoint	True	254
Template	True	253
wvol-myVVVoIVM-vg/Config-03ca033a	False	254:1
wvol-myVVVoIVM-vg/Data-05c7f7a5	False	254:3
wvol-myVVVoIVM-vg/Swap-5abd709c	False	254:4

The screenshot shows the vSphere Web Client interface for VM myVVVoIVM. The VM is powered on and running on host esxi-04.purecloud.com.

Property	Value
Guest OS	Microsoft Windows Server 2012 (64-bit)
Compatibility	ESXi 6.5 and later (VM version 13)
VMware Tools	Running, version:10272 (Current)
DNS Name	WIN-CBNUAKBNLS
IP Addresses	
Host	esxi-04.purecloud.com

Now if I choose esxi-02 and vMotion it, you can see the bindings appear on esxi-02 now:

The screenshot shows the vSphere Web Client interface with a filter applied to hosts. The 'Hosts' tab is selected, and the list shows two hosts: esxi-02.purecloud.com and esxi-04.purecloud.com, both belonging to the Castro cluster.

Name	Cluster
esxi-02.purecloud.com	Castro
esxi-04.purecloud.com	Castro

Volumes 22.57 G Provisioned 3.21 T  
 Snapshots 0.00 Data Reduction 3.9 to 1  
 Total 22.57 G Total Reduction 89.7 to 1

Connected Volumes			
Name ▲	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	
vvol-myVVolVM-vg/Config-03ca033a	False	254:1	
vvol-myVVolVM-vg/Data-05c7f7a5	False	254:3	
vvol-myVVolVM-vg/Swap-9fad31ef	False	254:2	

The config VVol is bound explicitly at the end of the vMotion operation from the original host:

> Hosts > Castro > esxi-04

Volumes 22.56 G Provisioned 3.21 T  
 Snapshots 0.00 Data Reduction 3.9 to 1  
 Total 22.56 G Total Reduction 89.6 to 1

Connected Volumes			
Name ▲	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	
vvol-myVVolVM-vg/Data-05c7f7a5	False	254:3	
vvol-myVVolVM-vg/Swap-5abd709c	False	254:4	

Then shortly thereafter, the data and swap VVols are unbound too from the original host:

Connected Volumes			
Name ▲	Shared	LUN	
pure-protocol-endpoint pure-protocol-endpoint	True	254	
Template	True	253	

# Conclusion

All of this is really behind the scenes stuff, which is nice for users. You don't need to worry about provisioning and de-provisioning or rescans etc. Simple, dynamic, automatic. Good stuff. I will probably revisit this topic as I dig into more advanced scenarios.