Storage Services - Volumes (Block Services)

[PDF generated June 03 2025. For all recent updates please see the Nutanix Bible releases notes located at https://nutanixbible.com/release_notes.html. Disclaimer: Downloaded PDFs may not always contain the latest information.]

The Nutanix Volumes feature (previously know as Acropolis Volumes) exposes back-end DSF storage to external consumers (guest OS, physical hosts, containers, etc.) via iSCSI.

This allows any operating system to access DSF and leverage its storage capabilities. In this deployment scenario, the OS is talking directly to Nutanix bypassing any hypervisor.

Core use-cases for Volumes:

- Shared Disks
  - Oracle RAC, Microsoft Failover Clustering, etc.

- Disks as first-class entities
  - Where execution contexts are ephemeral and data is critical
  - Containers, OpenStack, etc.

- Guest-initiated iSCSI
  - Bare-metal consumers
  - Exchange on vSphere (for Microsoft Support)

Qualified Operating Systems

The solution is iSCSI spec compliant, the qualified operating systems are just those of which have been validated by QA.

- Microsoft Windows Server 2008 R2, 2012 R2
- Redhat Enterprise Linux 6.0+

Volumes Constructs

The following entities compose Volumes:

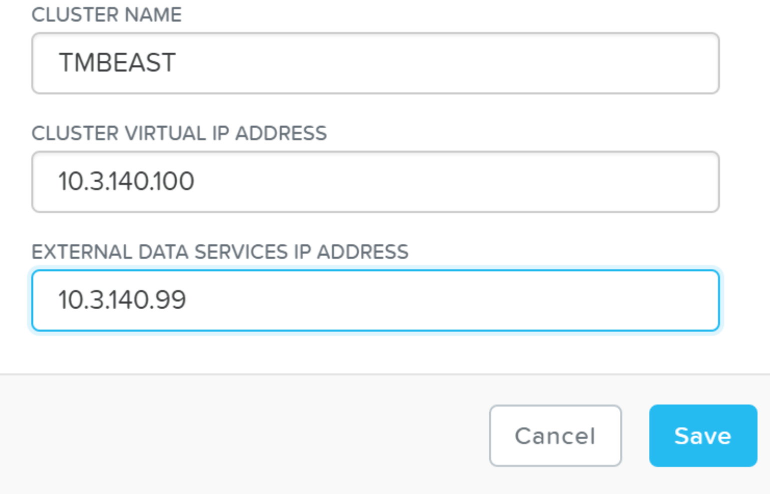
- **Data Services IP:** Cluster wide IP address used for iSCSI login requests (Introduced in 4.7)
- **Volume Group:** iSCSI target and group of disk devices allowing for centralized management, snapshotting, and policy application
- **Disk(s):** Storage devices in the Volume Group (seen as LUNs for the iSCSI target)
- **Attachment:** Allowing a specified initiator IQN access to the volume group
- **Secret(s):** Secret used for CHAP/Mutual CHAP authentication

NOTE: On the backend, a VG's disk is just a vDisk on DSF.

Pre-Requisites

Before we get to configuration, we need to configure the Data Services IP which will act as our central discovery / login portal.

We'll set this on the 'Cluster Details' page (Gear Icon -> Cluster Details):



CLUSTER NAME

TMBEAST

CLUSTER VIRTUAL IP ADDRESS

10.3.140.100

EXTERNAL DATA SERVICES IP ADDRESS

10.3.140.99

Volumes - Data Services IP

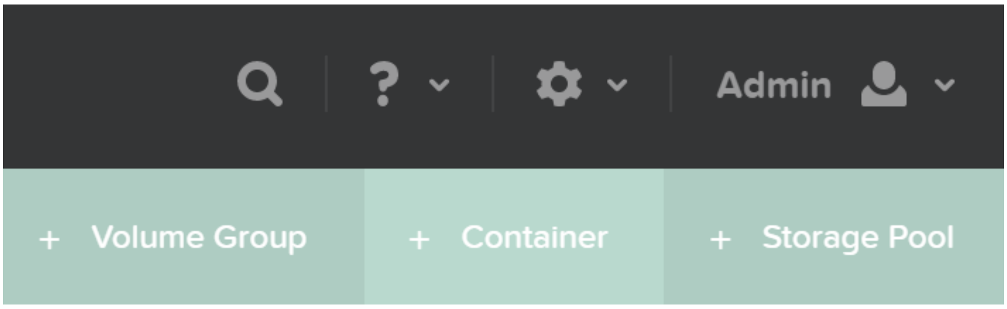
This can also be set via NCLI / API:

```
ncli cluster edit-params external-data-
services-ip-address=DATASERVICESIPADDRESS
```

Target Creation

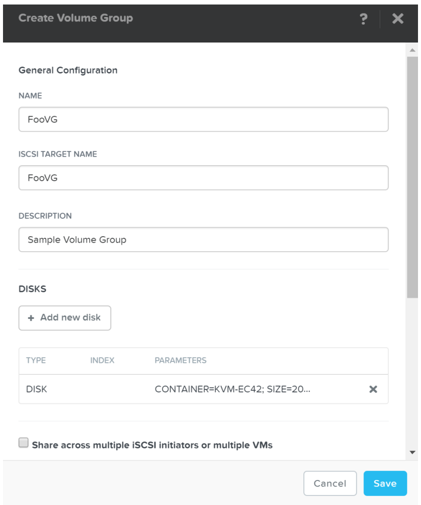
To use Volumes, the first thing we'll do is create a 'Volume Group' which is the iSCSI target.

From the 'Storage' page click on '+ Volume Group' on the right hand corner:



Volumes - Add Volume Group

This will launch a menu where we'll specify the VG details:



Create Volume Group

General Configuration

NAME

FooVG

ISCSI TARGET NAME

FooVG

DESCRIPTION

Sample Volume Group

DISKS

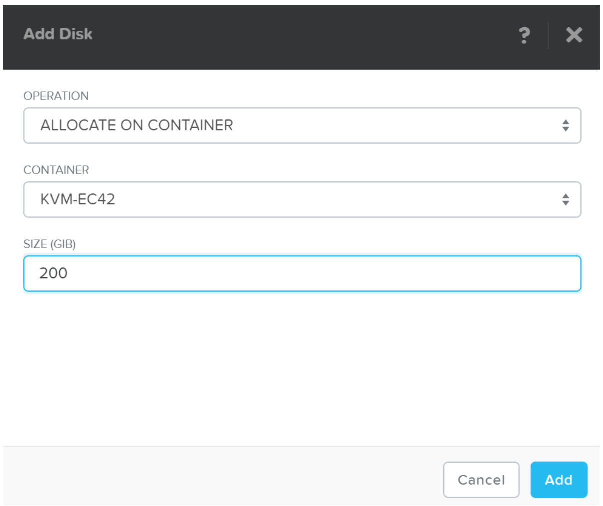
TYPE	INDEX	PARAMETERS
DISK		CONTAINER=KVM-EC42; SIZE=20...

□ Share across multiple iSCSI initiators or multiple VMs

Volumes - Add VG Details

Next we'll click on '+ Add new disk' to add any disk(s) to the target (visible as LUNs):

A menu will appear allowing us to select the target container and size of the disk:



Add Disk

OPERATION: ALLOCATE ON CONTAINER

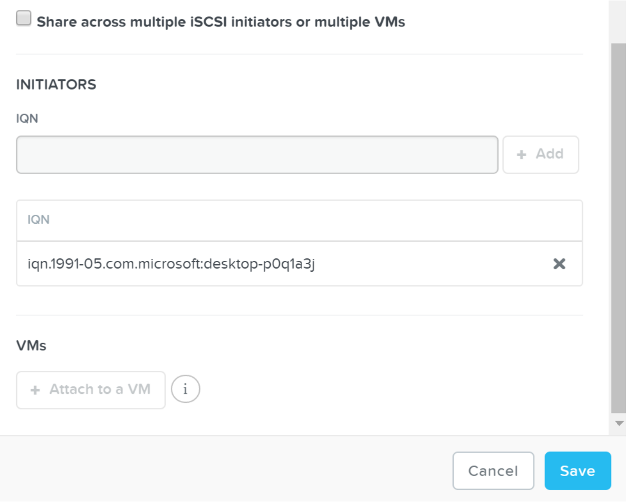
CONTAINER: KVM-EC42

SIZE (GIB): 200

Volumes - Add Disk

Click 'Add' and repeat this for however many disks you'd like to add.

Once we've specified the details and added disk(s) we'll attach the Volume Group to a VM or Initiator IQN. This will allow the VM to access the iSCSI target (requests from an unknown initiator are rejected):



□ Share across multiple iSCSI initiators or multiple VMs

INITIATORS

IQN

IQN
iqn.1991-05.com.microsoft:desktop-p0q1a3j

VMs

Volumes - Initiator IQN / VM

Click 'Save' and the Volume Group configuration is complete!

This can all be done via ACLI / API as well:

Create VG

```
vg.create VGName
```

Add disk(s) to VG

```
Vg.disk_create VGName container=CTRName create_size=Disk size, e.g. 500G
```

Attach initiator IQN to VG

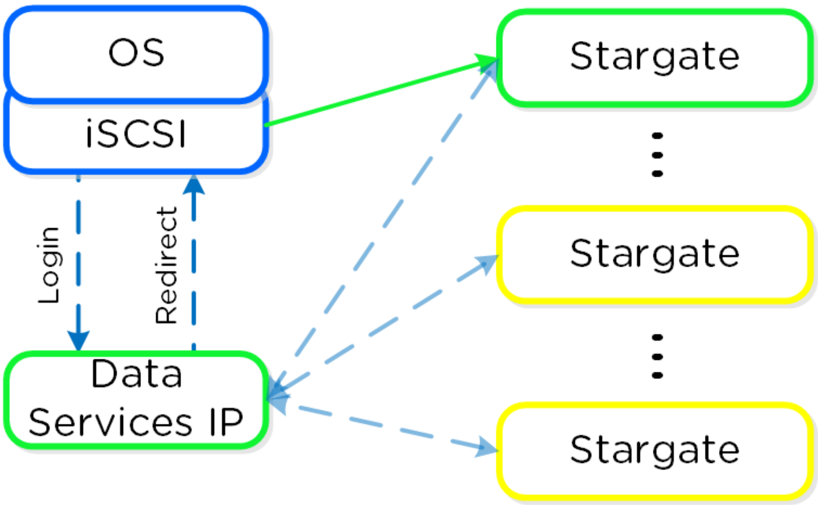
```
Vg.attach_external VGName InitiatorIQN
```

Path High-Availability (HA)

As mentioned previously, the Data Services IP is leveraged for discovery. This allows for a single address that can be leveraged without the need of knowing individual CVM IP addresses.

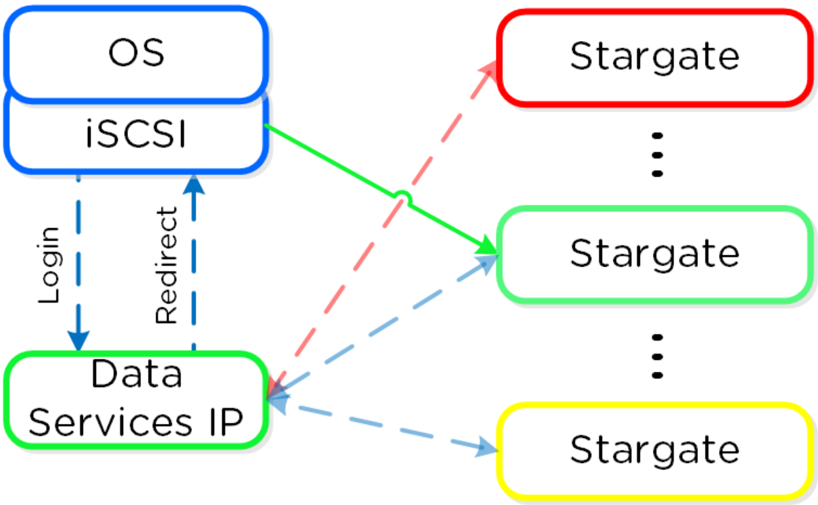
The Data Services IP will be assigned to the current iSCSI leader. In the event that fails, a new iSCSI leader will become elected and assigned the Data Services IP. This ensures the discovery portal will always remain available.

The iSCSI initiator is configured with the Data Services IP as the iSCSI target portal. Upon a login request, the platform will perform an iSCSI login redirect to a healthy Stargate.



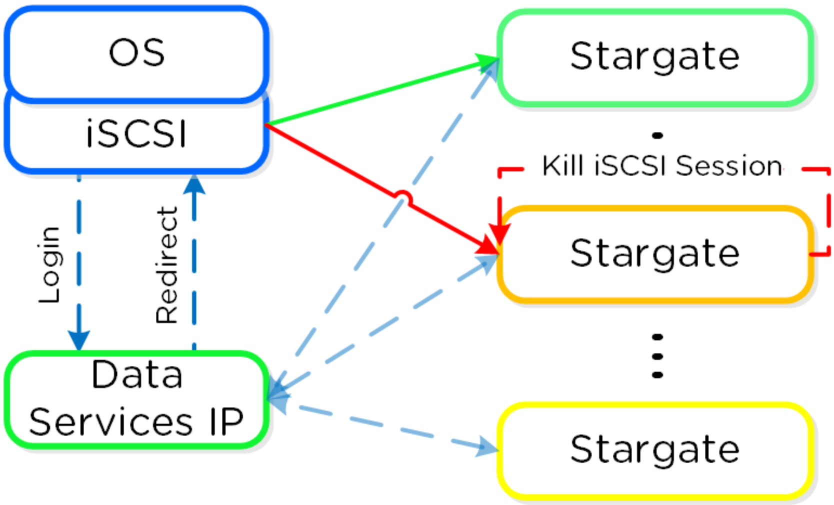
Volumes - Login Redirect

In the event where the active (affined) Stargate goes down, the initiator retries the iSCSI login to the Data Services IP, which will then redirect to another healthy Stargate.



Volumes - Failure Handling

If the affined Stargate comes back up and is stable, the currently active Stargate will quiesce I/O and kill the active iSCSI session(s). When the initiator re-attempts the iSCSI login, the Data Services IP will redirect it to the affined Stargate.



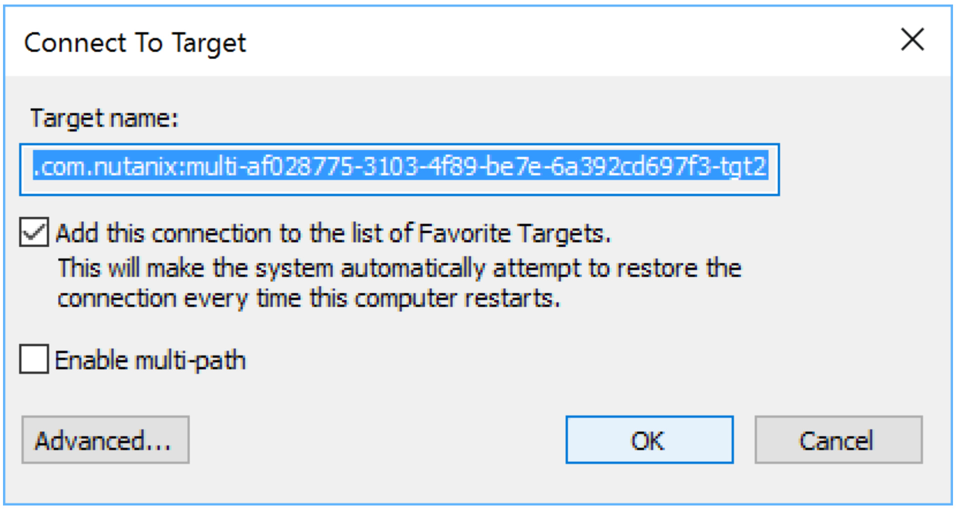
Volumes - Failback

Health Monitoring and Defaults

Stargate health is monitored using Zookeeper for Volumes, using the exact same mechanism as DSF.

For failback, the default interval is 120 seconds. This means once the affined Stargate is healthy for 2 or more minutes, we will quiesce and close the session. Forcing another login back to the affined Stargate.

Given this mechanism, client side multipathing (MPIO) is no longer necessary for path HA. When connecting to a target, there's now no need to check 'Enable multi-path' (which enables MPIO):



Connect To Target

Target name:

.com.nutanix:multi-af028775-3103-4f89-be7e-6a392cd697f3-tgt2

☑ Add this connection to the list of Favorite Targets.
This will make the system automatically attempt to restore the connection every time this computer restarts.

☐ Enable multi-path

Volumes - No MPIO

Multi-Pathing

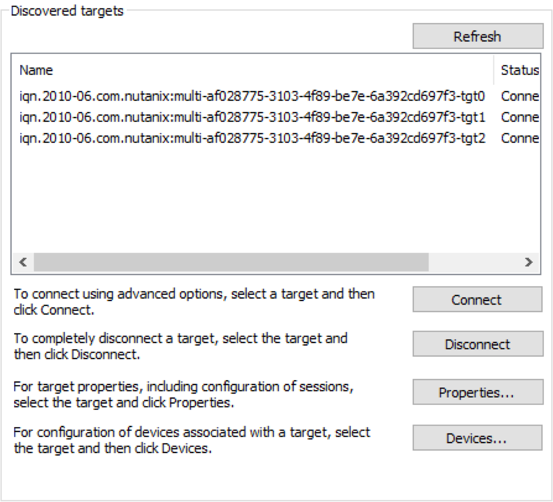
The iSCSI protocol spec mandates a single iSCSI session (TCP connection) per target, between initiator and target. This means there is a 1:1 relationship between a Stargate and a target.

As of 4.7, 32 (default) virtual targets will be automatically created per attached initiator and assigned to each disk device added to the volume group (VG). This provides an iSCSI target per disk device. Previously this would have been handled by creating multiple VGs with a single disk each.

When looking at the VG details in ACLI/API you can see the 32 virtual targets created for each attachment:

```
attachment_list {
    external_initiator_name: "iqn.1991-05.com.microsoft:desktop-foo"
    target_params {
        num_virtual_targets: 32
    }
}
```

Here we've created a sample VG with 3 disks devices added to it. When performing a discovery on my client we can see an individual target for each disk device (with a suffix in the format of '-tgt[int]'):



Discovered targets

Name	Status
iqn.2010-06.com.nutanix:multi-af028775-3103-4f89-be7e-6a392cd697f3-tgt0	Conne
iqn.2010-06.com.nutanix:multi-af028775-3103-4f89-be7e-6a392cd697f3-tgt1	Conne
iqn.2010-06.com.nutanix:multi-af028775-3103-4f89-be7e-6a392cd697f3-tgt2	Conne

To connect using advanced options, select a target and then click Connect.

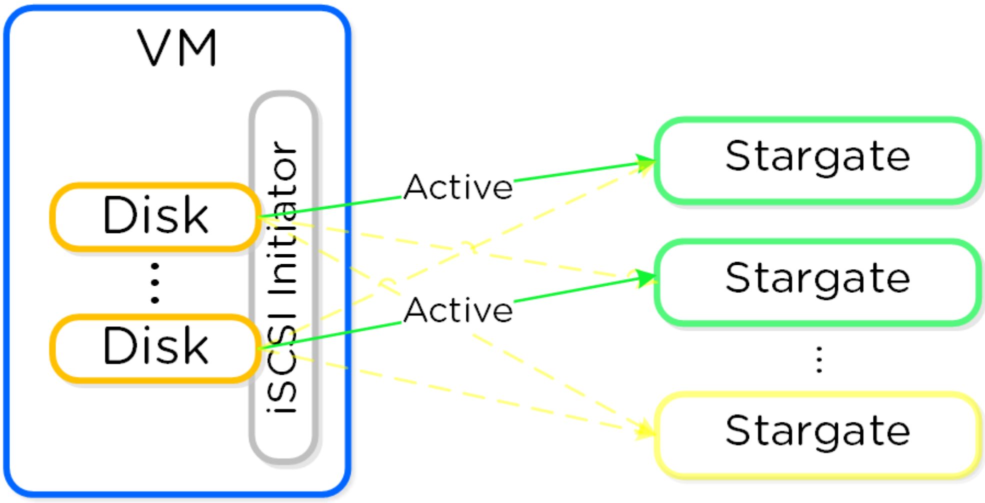
To completely disconnect a target, select the target and then click Disconnect.

For target properties, including configuration of sessions, select the target and click Properties.

For configuration of devices associated with a target, select the target and then click Devices.

Volumes - Virtual Target

This allows each disk device to have its own iSCSI session and the ability for these sessions to be hosted across multiple Stargates, increasing scalability and performance:



Volumes - Multi-Path

Load balancing occurs during iSCSI session establishment (iSCSI login), for each target.

Active Path(s)

You can view the active Stargate(s) hosting the virtual target(s) with the following command (will display CVM IP for hosting Stargate):

```
# Windows
Get-NetTCPConnection -State Established -RemotePort 3205
```

```
# Linux
iscsiadm -m session -P 1
```

As of 4.7 a simple hash function is used to distribute targets across cluster nodes. In 5.0 this is integrated with the Dynamic Scheduler which will re-balance sessions if necesary. We will continue to look at the algorithm and optimize as necessary. It is also possible to set a preferred node which will be used as long as it is in a healthy state.

SCSI UNMAP (TRIM)

Volumes supports the SCSI UNMAP (TRIM) command in the SCSI T10 specification. This command is used to reclaim space from deleted blocks.