

SUSE Enterprise Storage™ 3 and iSCSI

	page
Table of Contents	
Executive Summary.....	2
Introduction	2
The Ceph Storage System	2
iSCSI Block Storage.....	3
lrbd.....	4
SUSE Enterprise Storage 3/lrbd Deployment Considerations.....	5
Installing and Configuring SUSE Enterprise Storage 3 and lrbd	6
Creating RBD Images.....	6
Connecting to lrbd-Managed Targets.....	8
Conclusion.....	19

Executive Summary

SUSE Enterprise Storage™ is a distributed storage management product from SUSE® that builds on the distributed, highly-available, open-source Ceph storage system.

Included with SUSE Enterprise Storage 3 is a facility that opens Ceph storage connectivity to heterogeneous clients, such as Microsoft Windows and VMware vSphere, through the iSCSI protocol. Multipath iSCSI access enables availability and scalability for these clients, and the standardized iSCSI protocol also provides an additional layer of security isolation between clients and the SUSE Enterprise Storage 3 cluster. The configuration facility is named “lrbd” and ships as part of SUSE Enterprise Storage 3. Like Ceph, lrbd is an open-source project, which builds on proven technology such as the Linux kernel and the Linux iSCSI target. Using lrbd, Ceph storage administrators can define thin-provisioned, replicated, highly-available volumes supporting read-only snapshots, read-write clones, and automatic resizing with Ceph RADOS Block Device (RBD). Administrators can then export volumes either via a single lrbd gateway host, or via multiple gateway hosts supporting multipath failover. Linux, Windows, and VMware hosts can connect to volumes using the iSCSI protocol, which makes them available like any other SCSI block device. This means SUSE Enterprise Storage 3 customers can effectively run a complete block-storage infrastructure subsystem on Ceph that provides all features and benefits of a conventional SAN while reducing vendor lock-in, enabling future growth and reducing cost.

Introduction

This white paper provides an overview of the following technologies:

- *The Ceph distributed storage system*
- *The iSCSI network block-storage protocol, its Linux kernel implementation, and the command-line utility associated with this implementation, targetcli*

- *lrbd, a program creating targetcli configurations based on Ceph RBD images and metadata stored in the Ceph RADOS object store*

All these technologies are provided in a unified, streamlined fashion in SUSE Enterprise Storage 3.

The Ceph Storage System

Ceph is a distributed, highly available, software-driven open-source storage system. It was originally conceived at the University of California, Santa Cruz (UCSC) and has been in continuous development since 2004. Ceph is software-defined storage (SDS) technology. Instead of being tied to specialized hardware designed for a specific purpose like conventional computer storage technology, it relies on commodity hardware, an open-source implementation, open protocols and an open application programming interface (API) to provide highly scalable storage capacity to clients.

Ceph is extremely versatile. Applications can use Ceph storage directly, but can also use intermediate abstraction layers providing a distributed POSIX file system, a RESTful object store akin to OpenStack Swift and Amazon S3, and a virtual block device. All of Ceph’s server-side components run on the Linux operating system in userspace. The same is true for the majority of client-side components; however a limited subset of these do run in the Linux kernel.

Reliable Distributed Autonomic Storage (RADOS)

The Reliable Distributed Autonomic Object Store (RADOS) is a core design component of Ceph and sets it apart from other comparable storage solutions. RADOS is highly distributed, scalable, self-healing, auto-balancing, and highly available. Any chunk of data in RADOS is an object (of arbitrary size), and clients can localize any object computationally, using an algorithm known as Controlled Replication Under Scalable Hashing (CRUSH). RADOS omits the notion of a centralized data look-up service, which is a typical single point of failure and scalability choke point in conventional storage systems. RADOS objects also support the notion of arbitrary key/value pairs known as “extended attributes” (xattrs), and any object may have zero or more extended attributes.

The physical distribution of objects in RADOS is based on a configurable set of policies called the CRUSH map. Using the CRUSH map, a storage administrator can model the physical storage nodes in the Ceph cluster in a hierarchical fashion, thereby defining an arbitrarily complex storage topology. For example, individual data storage locations can be hierarchically organized along the lines of physical nodes, racks, aisles, rooms and data centers.

At the storage administrator’s discretion, RADOS as a whole can be broken down into logical subdivisions known as “pools.” It is then possible to define replication and availability policies by pool. An administrator can define that one specific pool holds three copies of each RADOS object, with each replica being in a different rack. An administrator can also define that another pool holds just two copies of each object, but with each replica being in a different data center.

RADOS Block Devices

Block devices are a simple and ubiquitous storage abstraction and are, in and of themselves, very simple in nature. Block devices have a defined length (the maximum capacity they can store) and a sector size (the smallest increment in which they can read and write data). As a result, there are defined addresses for every location on the device, known as the “sector offset.” The input/output (I/O) interface on a block device is very simple. Either a memory buffer of a given length (a multiple of the sector size) is being written onto the device at a given location, or data at a given location is being read into a memory buffer.

RADOS Block Devices (RBDs) are a very versatile client-side abstraction layer on top of RADOS, which enable users to interact with the storage cluster as if it is conventional block storage. With RBD, the user defines images (volumes), and can then use the available RBD client libraries to interact with these devices as if they were regular block devices. What makes RBDs special is that every block read-and-write operation automatically translates into a RADOS object operation. This is transparent to the user and means that RBDs seamlessly “inherit” RADOS’ distribution, replication, scalability and high availability.

iSCSI Block Storage

iSCSI is an implementation of the Small Computer System Interface (SCSI) command set using the Internet Protocol (IP), specified in RFC 3720. iSCSI is implemented as a service in which a client (the initiator) talks to a server (the target) via a session on TCP port 3260. An iSCSI target’s IP address and port are called an “iSCSI portal,” where a target can be exposed through one or more portals. The combination of a target and one or more portals is called “the target portal group” (TPG).

The underlying data link layer protocol for iSCSI is commonly Ethernet. More specifically, modern iSCSI infrastructures use 10 Gigabit Ethernet or faster networks for optimal throughput. Ten Gigabit Ethernet connectivity between the iSCSI gateway and the back-end Ceph cluster is strongly recommended.

The Linux Kernel iSCSI Target

The Linux kernel iSCSI target was originally named “LIO” for linux-iscsi.org, the project’s original domain and website. For some time, no fewer than four competing iSCSI target implementations were available for the Linux platform, but LIO ultimately prevailed as the single iSCSI reference target. The mainline kernel code for LIO uses the simple but somewhat ambiguous name “target,” distinguishing between “target core” and a variety of front-end and back-end target modules.

The most commonly used front-end module is arguably iSCSI. However, LIO also supports Fibre Channel (FC), Fibre Channel over Ethernet (FCoE) and several other front-end protocols. At this time, only the iSCSI protocol has been validated with SUSE Enterprise Storage 3.

The most frequently used target back-end module is one that is capable of simply re-exporting any available block device on the target host. This module is named “iblock.” However, and importantly for SUSE Enterprise Storage, LIO also has an RBD-specific back-end module supporting parallelized multipath I/O access to RBD images.

iSCSI Initiators

LINUX

The standard initiator for the Linux platform is open-iscsi. This launches a daemon, iscsid, which the user can then use to discover iSCSI targets on any given portal, log in to targets and map iSCSI volumes. iscsid communicates with the SCSI mid-layer to create in-kernel block devices that the kernel can then treat like any other SCSI block device on the system. The open-iscsi initiator can be deployed in conjunction with the Device Mapper Multipath (dm-multipath) facility to provide a highly available iSCSI block device.

MICROSOFT WINDOWS AND HYPER-V

The default iSCSI initiator for the Microsoft Windows operating system is the Microsoft iSCSI initiator. The iSCSI service can

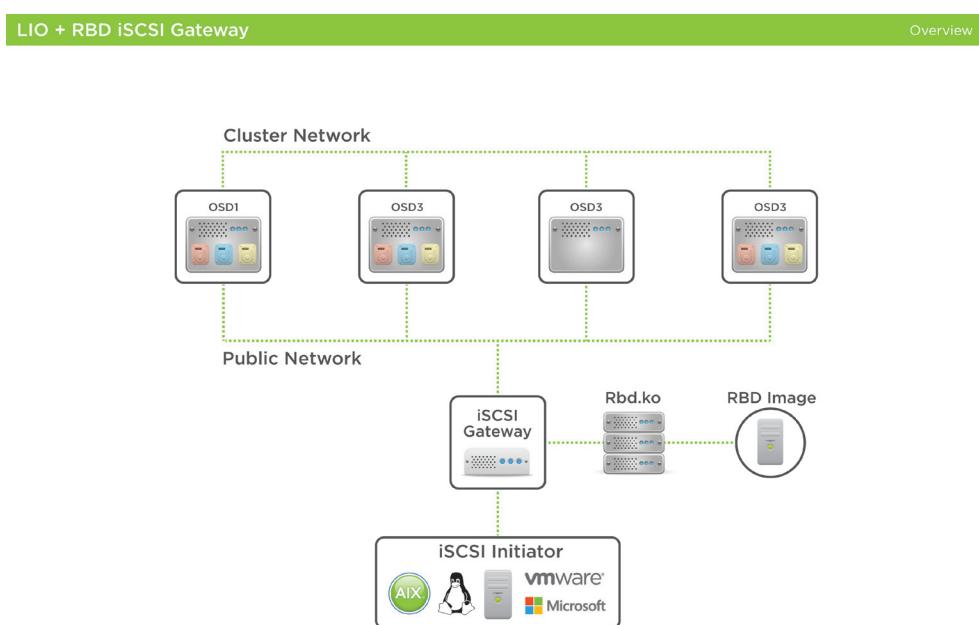
be configured via a graphical user interface (GUI) and supports multipath I/O for high availability.

VMWARE

The default iSCSI initiator for VMware vSphere and ESX is the VMware ESX software iSCSI initiator, vmkiscsi. Once enabled, it can be configured either from the vSphere client or using the vmkiscsi-tool command. You can then format storage volumes connected through the vSphere iSCSI storage adapter with VMFS and use them like any other VM storage device. The VMware initiator also supports multipath I/O for high availability.

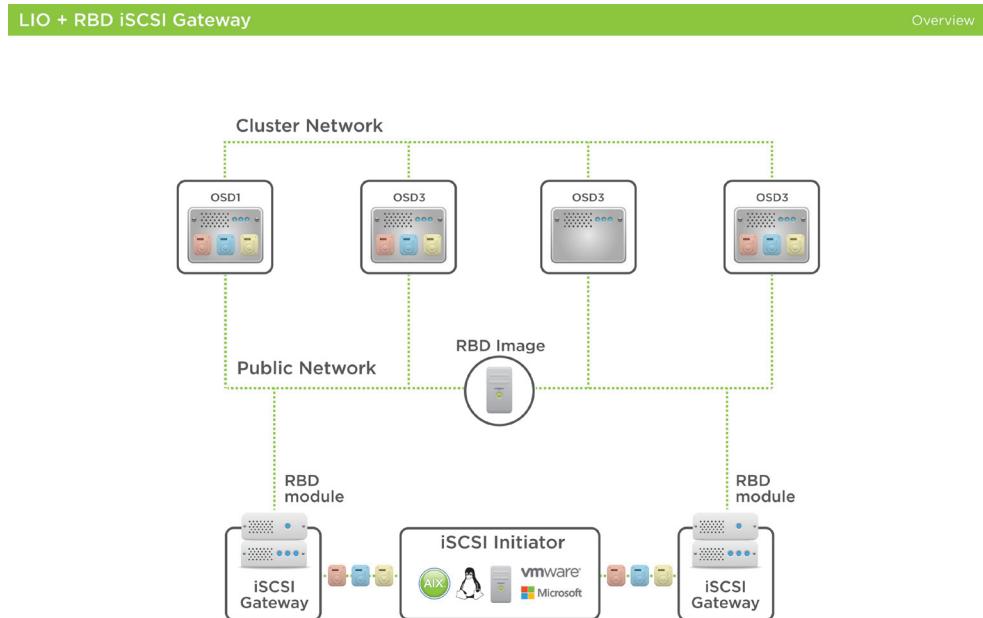
Irbd

Irbd combines the benefits of RADOS Block Devices with the ubiquitous versatility of iSCSI. By employing Irbd on an iSCSI target host (known as the “Irbd gateway”), any application that needs to make use of block storage can benefit from Ceph, even if it does not speak any Ceph client protocol. Instead, users can use iSCSI or any other target front-end protocol to connect to an LIO target, which (automatically and transparently) translates all target I/O to RBD storage operations.



lrbd is inherently highly available and supports multipath operations. Thus, downstream initiator hosts can use multiple iSCSI gateways for both high availability and scalability. When communicating with an iSCSI configuration with more than

one gateway, initiators may load-balance iSCSI requests across multiple gateways. In the event of a gateway failing, being temporarily unreachable or being disabled for maintenance, I/O will transparently continue via another gateway.



SUSE Enterprise Storage 3/lrbd Deployment Considerations

A minimum configuration of SUSE Enterprise Storage 3 with lrbd consists of the following components:

- A Ceph storage cluster. The Ceph cluster consists of a minimum of four physical servers hosting at least eight object storage daemons (OSDs) each. In such a configuration, three OSD nodes also double as a monitor (MON) host.
- An iSCSI target server running the LIO iSCSI target, configured via lrbd.
- An iSCSI initiator host, running open-iscsi (Linux), the Microsoft iSCSI Initiator (Microsoft Windows), or any other compatible iSCSI initiator implementation.

A recommended production configuration of SUSE Enterprise Storage 3 with lrbd consists of:

- A Ceph storage cluster. A production Ceph cluster consists of any number of (typically more than ten) OSD nodes, "each typically running 10-12 object storage daemons (OSDs), with no fewer than three dedicated MON hosts."
- Several iSCSI target servers running the LIO iSCSI target, configured via lrbd. For iSCSI failover and load-balancing, these servers must run a kernel supporting the target_core_rbd module. This also requires that the target servers run at least the version 3.12.48-52.27.1 of the kernel-default package. Updates packages are available from the SUSE Linux Enterprise Server maintenance channel.
- Any number of iSCSI initiator hosts running open-iscsi (Linux), the Microsoft iSCSI Initiator (Microsoft Windows) or any other compatible iSCSI initiator implementation.

Installing and Configuring SUSE Enterprise Storage 3 and Lrbd

Adding the SUSE Enterprise Storage Installation Media

Use YaST® to configure SUSE Enterprise Storage as an add-on product using a local ISO.

The recommended method would be to utilize the Subscription Management Tool (SMT) from SUSE. This tool can be obtained as an add-on to SUSE Linux Enterprise Server and is found here: www.suse.com/documentation/smt11/

After SMT is installed and your server(s) are connected to it and the proper repositories are mirrored (including SUSE Enterprise Storage 3), going to YaST → Add-on Products → Add- → Extensions and Modules from Registration Server... → SUSE Enterprise Storage 3 x86_64 will allow installation of SUSE Enterprise Storage 3 in a way that ensures proper repositories are configured to include the most recent updates and allow for future updates to be easily propagated.

Deploying a Ceph Cluster

Once your SUSE Enterprise Storage 3 repository has been configured, proceed with deploying your Ceph cluster as outlined in: www.suse.com/documentation/ses-2/book_storage_admin/data/cha_ceph_install.html

Installing the ceph_iscsi Pattern

On your designated iSCSI target server nodes, install the [ceph_iscsi pattern](#). Doing so will automatically install lrbd, the necessary Ceph binaries and libraries, and the targetcli command-line tool:

```
# zypper in -t pattern ceph_iscsi
```

Repeat this step on any node that you want to act as a fail-over or load-balancing target server node.

Creating RBD Images

RBD images are created in the Ceph store and subsequently exported to iSCSI. We recommend that you use a dedicated

RADOS pool for this purpose. You can create a volume from any host that is able to connect to your storage cluster using the Ceph rbd command-line utility. This requires the client to have at least a minimal ceph.conf configuration file and appropriate CephX authentication credentials. To create a new volume for subsequent export via iSCSI, use the rbd create command, specifying the volume size in megabytes. For example, in order to create a 100GB volume named "testvol" in the pool named "iscsi," run:

```
# rbd --pool iscsi create --size=102400 testvol
```

The above command creates an RBD "format 1" volume. In order to enable layered snapshots and cloning, you may want to create a "format 2" volume instead:

```
# rbd --pool iscsi create --image-format 2 --size=102400 testvol
```

To always enable format 2 on all newly created images, add the following configuration options to /etc/ceph/ceph.conf:

```
[client]
rbd default image format = 2
```

Once this option is set, every invocation of rbd create becomes functionally equivalent to rbd create --image-format 2 and always creates a volume in the new image format. Note that this is a client-side option that must be set on any host you use to create a new volume. It is not sufficient to set this on your Ceph servers.

Exporting RBD Images via iSCSI

To export RBD images via iSCSI, use the lrbd utility. Lrbd allows you to create, review and modify the iSCSI target configuration, which uses a JSON format.

In order to edit the configuration, use lrbd -e or lrbd --edit. This command will invoke the default editor, as defined by the EDITOR environment variable. You may override this behavior by setting the -E option in addition to -e.

Below is an example configuration for:

1. Two iSCSI gateway hosts named “iscsi1.example.com” and “iscsi2.example.com”
2. Defining a single iSCSI target with an iSCSI Qualified Name (IQN) of “iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol”

3. With a single iSCSI Logical Unit (LU)
4. Backed by an RBD image named “testvol” in the RADOS pool “rbd”,
5. And exporting the target via two portals named “east” and “west”

```
{  
    "auth": [  
        {  
            "target": "iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol",  
            "authentication": "none"  
        }  
    ],  
    "targets": [  
        {  
            "target": "iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol",  
            "hosts": [  
                {  
                    "host": "iscsi1.example.com",  
                    "portal": "east"  
                },  
                {  
                    "host": "iscsi2.example.com",  
                    "portal": "west"  
                }  
            ]  
        }  
    ],  
    "portals": [  
        {  
            "name": "east",  
            "addresses": [  
                "192.168.124.104"  
            ]  
        },  
        {  
            "name": "west",  
            "addresses": [  
                "192.168.124.105"  
            ]  
        }  
    ]  
}
```

```

"pools": [
    {
        "pool": "rbd",
        "gateways": [
            {
                "target": "iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol",
                "tpg": [
                    {
                        "image": "testvol"
                    }
                ]
            }
        ]
    }
]
}

```

Note that whenever you refer to a hostname in the configuration, this hostname must match the iSCSI gateway's “uname -n” command output.

The edited JSON is stored in the extended attributes (xattrs) of a single RADOS object per pool. This object is available to the gateway hosts where the JSON is edited, and all gateway hosts connected to the same Ceph cluster. No configuration information is stored locally on the lrbd gateway.

To activate the configuration, store it in the Ceph cluster, and do one of the following things (as root):

- Run the “lrbd” command (without additional options) from the command line
- Restart the “lrbd” service with “service lrbd restart”

The lrbd “service” does not operate any background daemon. Instead, it simply invokes the lrbd command. This type of service is known as a “one-shot” service.

You should also enable lrbd to auto-configure on system startup. To do so, run the “systemctl enable lrbd” command.

The configuration above reflects a simple, one-gateway setup. Lrbd configuration can be much more complex and powerful. The lrbd RPM package comes with an extensive set of configuration examples, which you may refer to by checking the contents of the /usr/share/doc/packages/lrbd/samples directory after installation. The samples are also available from: <https://github.com/SUSE/lrbd/tree/master/samples>

Connecting to Lrbd-Managed Targets

Linux (open-iscsi)

Connecting to Lrbd-backed iSCSI targets with open-iscsi is a two-step process. First, the initiator must discover the iSCSI targets available on the gateway host; then he or she must log in and map the available logical units (LUs).

Both steps require that the open-iscsi daemon is running. The way you start the open-iscsi daemon depends on your Linux distribution:

- On SUSE Linux Enterprise Server and Red Hat Enterprise Linux hosts, run “service iscsid start”
- On Debian and Ubuntu hosts, run “service open-iscsi start”

If your initiator host runs SUSE Linux Enterprise Server, refer to www.suse.com/documentation/sles-12/stor_admin/data/sec_iscsi_initiator.html or www.suse.com/documentation/sles11/stor_admin/data/sec_inst_system_iscsi_initiator.html for details on how to connect to an iSCSI target.

For any other Linux distribution supporting open-iscsi, proceed to discover targets on your lrbd gateway (this example uses iscsi1.example.com as the portal address; for multipath access repeat these steps with iscsi2.example.com):

```
# iscsiadadm -m discovery -t sendtargets -p iscsi1.example.com  
192.168.124.104:3260,1 iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol
```

Then, log in to the portal. If the login completes successfully, any RBD-backed logical units on the portal will immediately become available on the system SCSI bus:

```
# iscsiadadm -m node -p iscsi1.example.com --login  
Logging in to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol, portal:  
192.168.124.104,3260] (multiple)  
Login to [iface: default, target: iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol, portal:  
192.168.124.104,3260] successful.
```

Repeat this process for other portal IP addresses or hosts.

If your system has the lsscsi utility installed, you use it to enumerate available SCSI devices on your system:

```
# lsscsi  
[8:0:0:0]    disk    SUSE      RBD          4.0    /dev/sde  
[9:0:0:0]    disk    SUSE      RBD          4.0    /dev/sdf
```

In a multipath configuration (where two connected iSCSI devices represent one and the same LU), you can also examine the multipath device state with the multipath utility:

```
# multipath -ll  
360014050cf9dcfcb2603933ac3298dca dm-9 SUSE,RBD  
size=49G features='0' hwhandler='0' wp=rw  
|-- policy='service-time 0' prio=1 status=active  
| `-- 8:0:0:0 sde 8:64 active ready running  
`-- policy='service-time 0' prio=1 status=enabled  
   `-- 9:0:0:0 sdf 8:80 active ready running
```

You can now use this multipath device as you would any block device. For example, you can use the device as a Physical Volume for Linux Logical Volume Management (LVM), or you can simply create a file system on it. The example below demonstrates how to create an XFS file system on the newly connected multipath iSCSI volume:

```
# mkfs -t xfs /dev/mapper/360014050cf9dcfcb2603933ac3298dca
log stripe unit (4194304 bytes) is too large (maximum is 256KiB)
log stripe unit adjusted to 32KiB
meta-data=/dev/mapper/360014050cf9dcfcb2603933ac3298dca isize=256    agcount=17, agsize=799744 blks
          =                      sectsz=512  attr=2, projid32bit=1
          =                      crc=0      finobt=0
data       =                      bsize=4096   blocks=12800000, imaxpct=25
          =                      sunit=1024   swidth=1024 blks
naming     =version 2           bsize=4096   ascii-ci=0 ftype=0
log        =internal log       bsize=4096   blocks=6256, version=2
          =                      sectsz=512   sunit=8 blks, lazy-count=1
realtime   =none                extsz=4096   blocks=0, rtextents=0
```

Note that because XFS is a non-clustered file system, you may only ever mount it on a single iSCSI initiator node at any given time.

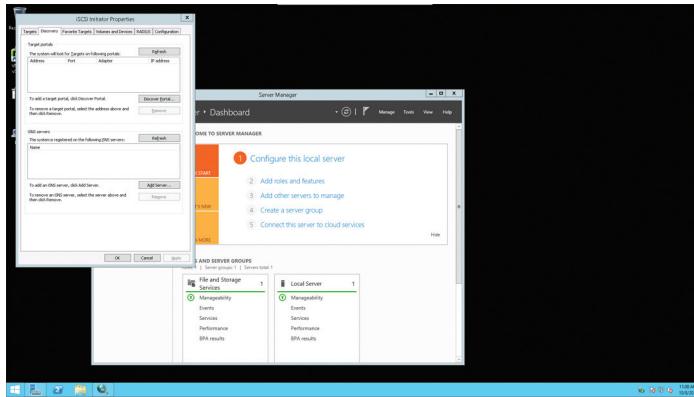
If at any time you want to discontinue using the iSCSI LUs associated with a particular target, run the following command:

```
# iscsiadm -m node -p iscsil.example.com --logout
Logging out of session [sid: 18, iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol, portal:
192.168.124.104,3260]
Logout of [sid: 18, target: iqn.2003-01.org.linux-iscsi.iscsi.x86:testvol, portal: 192.168.124.104,3260]
successful.
```

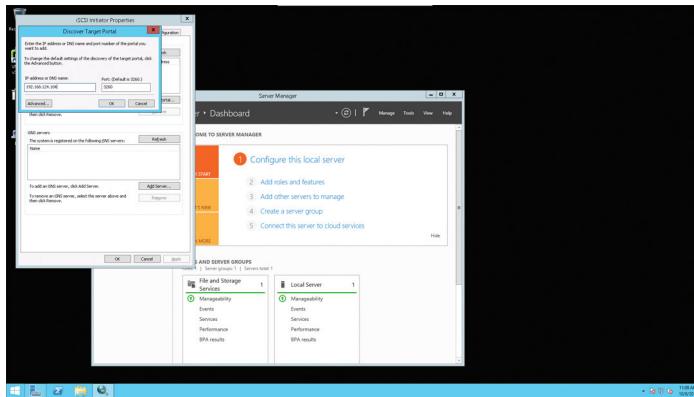
As with discovery and login, you must repeat the logout steps for all portal IP addresses or hostnames.

Microsoft Windows (Microsoft iSCSI Initiator)

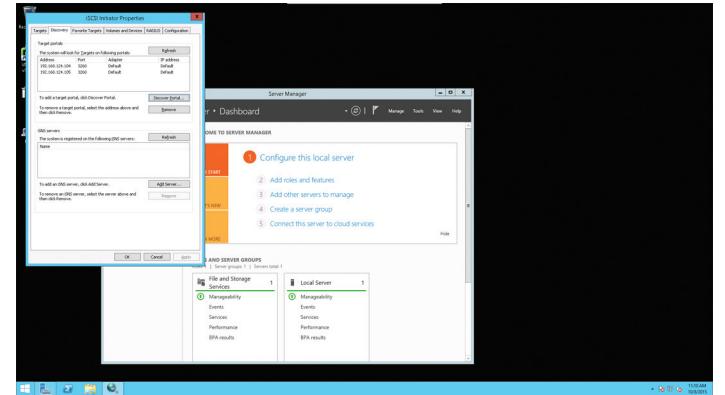
To connect to a SUSE Enterprise Storage iSCSI target from a Windows 2012 server, open Windows Server Manager. From the Dashboard, select Tools → iSCSI Initiator. The iSCSI Initiator Properties dialog appears. Select the Discovery tab:



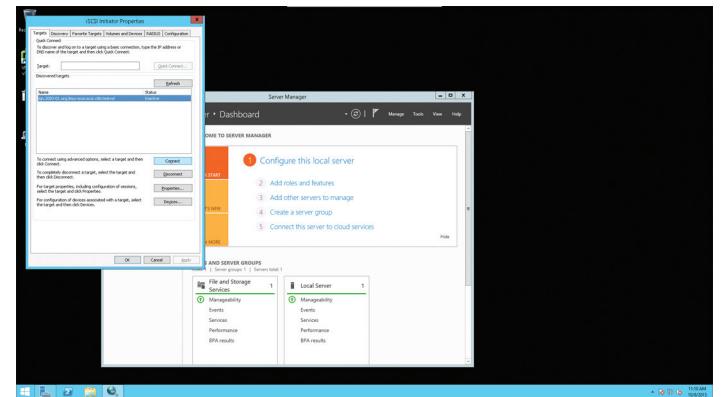
In the Discover Target Portal dialog, enter the target's hostname or IP address in the Target field and click OK:



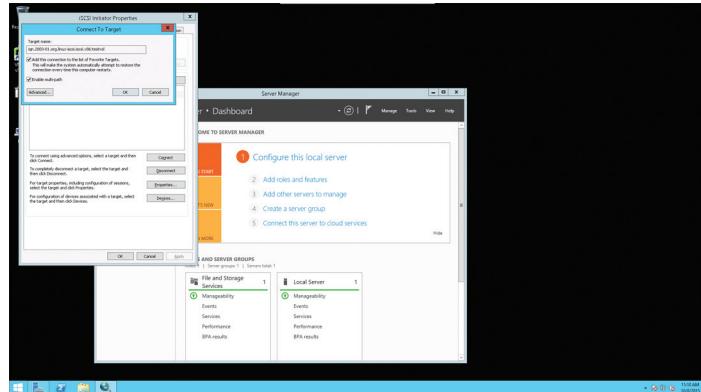
Repeat this process for all other gateway hostnames or IP addresses. Once completed, review the Target Portals list:



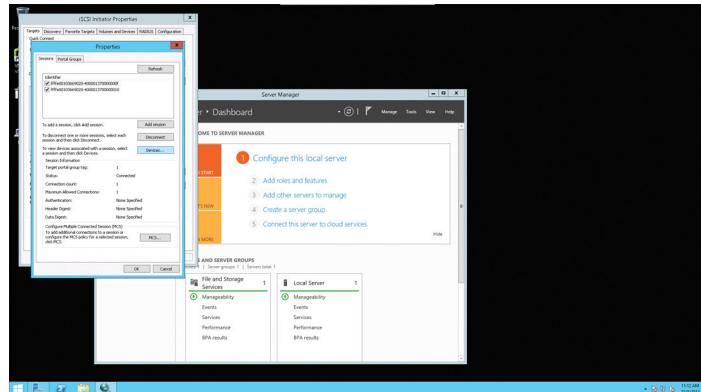
Next, switch to the Targets tab and review your discovered target(s).



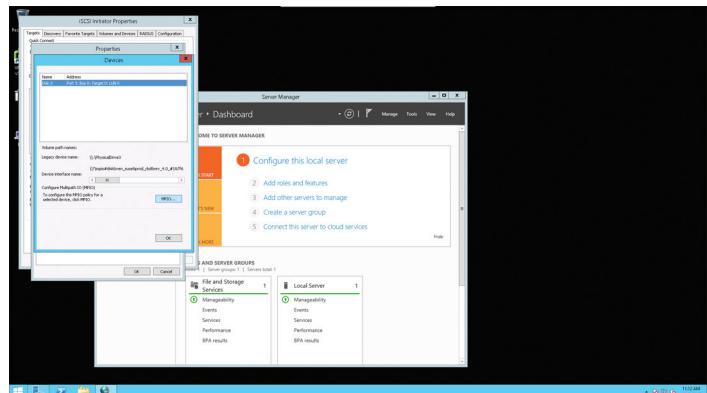
Click “Connect” in the Targets tab. The Connect To Target dialog appears. Select the “Enable multi-path” checkbox to enable multipath I/O (MPIO); then click OK:



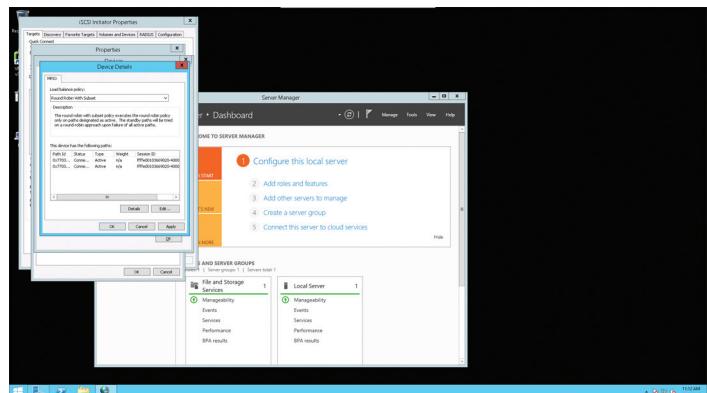
Once the Connect to Target dialog closes, select Properties to review the target's properties:



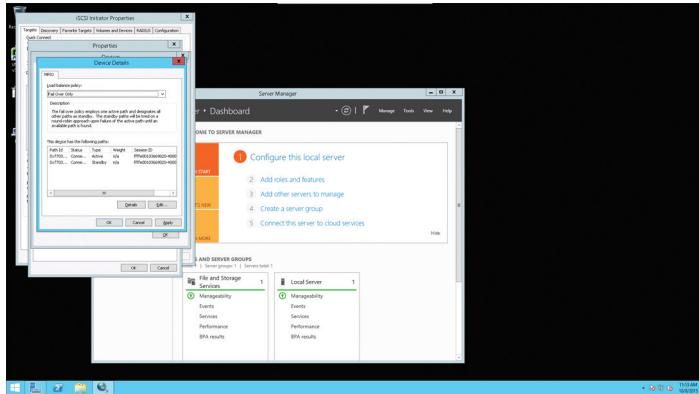
Select “Devices”:



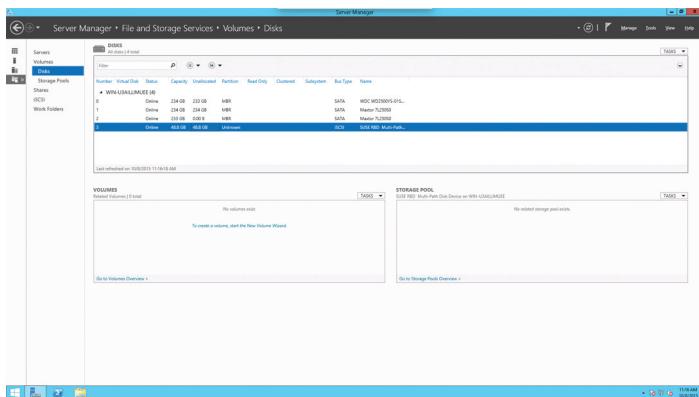
Now click on “MPIO” to review the multipath I/O configuration:



The default Load Balance policy is Round Robin With Subset. If you prefer a pure fail-over configuration, change it to Fail Over Only:

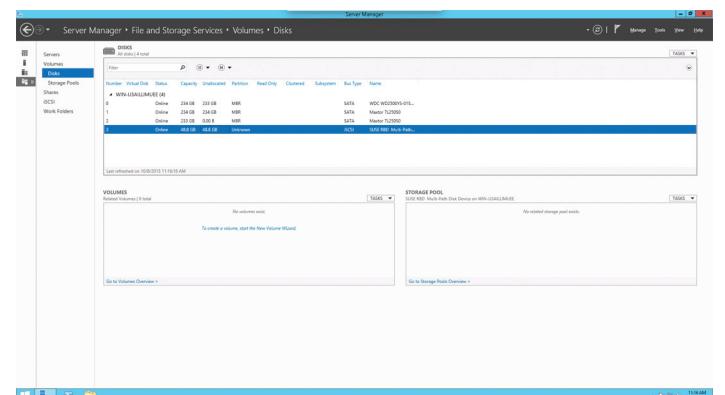


This concludes the iSCSI initiator configuration. The iSCSI volumes are now available just like any other SCSI devices and may be initialized for use as volumes and drives. Click OK to close the iSCSI Initiator Properties dialog, and proceed with the File and Storage Services role from the Server Manager dashboard.

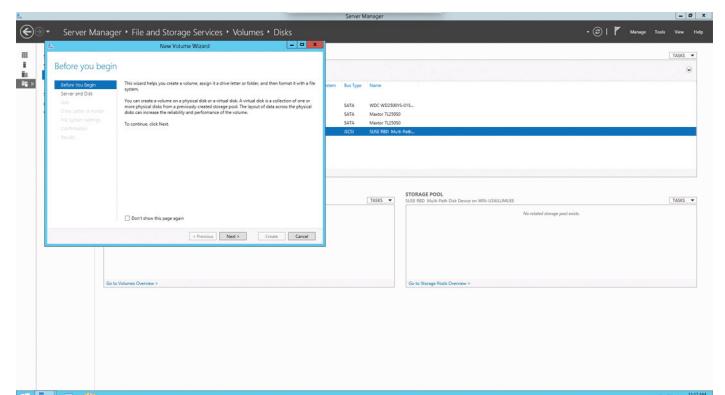


Observe the newly connected volume. It identifies as SUSE RBD SCSI Multi-Path Drive on the iSCSI bus and is initially marked with an Offline status and a partition table type of Unknown.

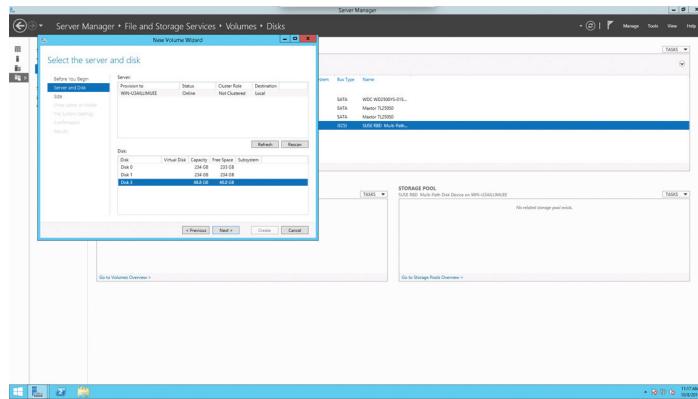
If the new volume does not appear immediately, select Rescan Storage from the Tasks drop-down to rescan the iSCSI bus. Right-click on the iSCSI volume and select New Volume from the context menu:



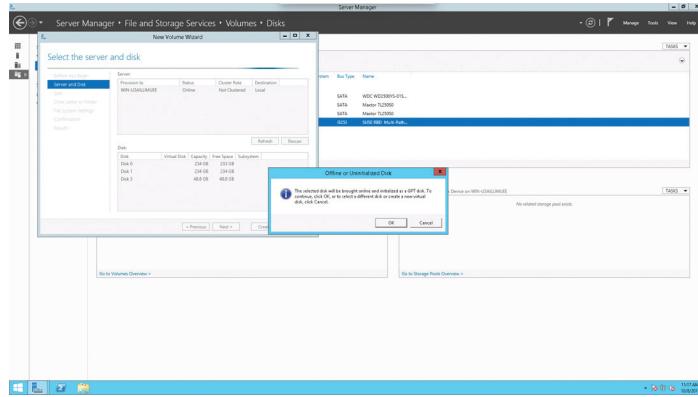
The New Volume wizard appears. Click Next to begin.



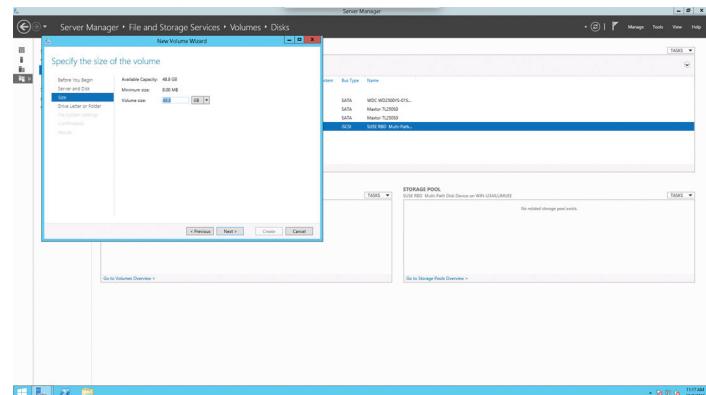
Highlight the newly connected iSCSI volume and click Next:



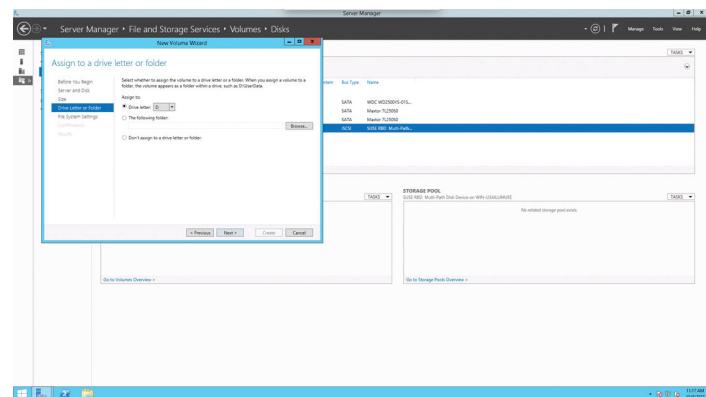
Initially, the device is empty and does not contain a partition table. When prompted, confirm the dialog indicating that the volume will be initialized with a GPT partition table:



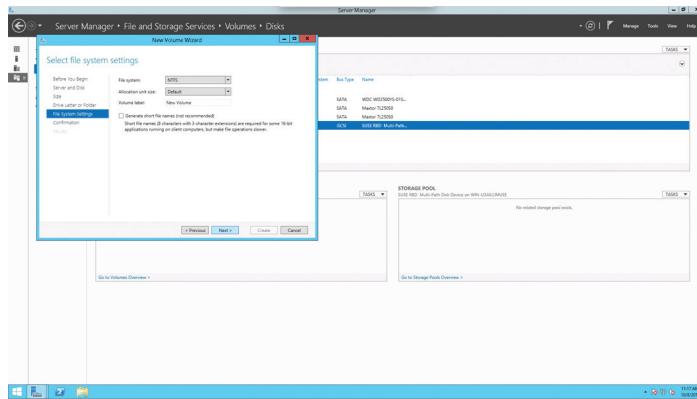
Next, select the volume size. Typically, you would use the device's full capacity:



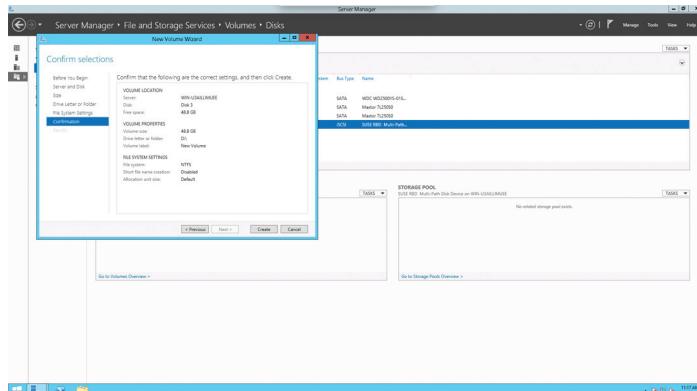
Then, assign a drive letter or folder name where the newly created volume will become available:



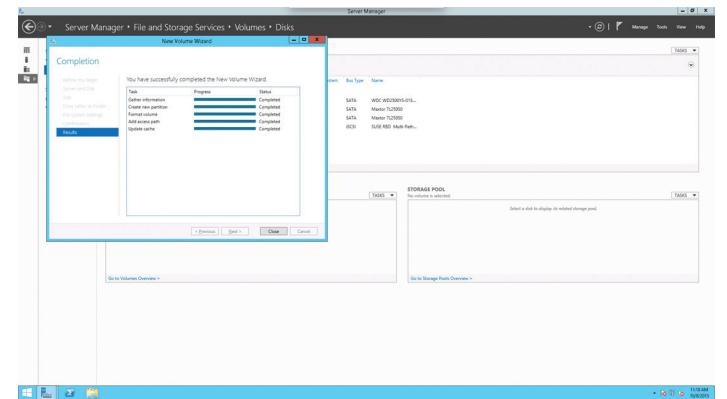
Select a file system to create on the new volume:



Finally, confirm your selections and click the Create button to finish creating the volume:



When the process finishes, review the results, then click Close to conclude the drive initialization:

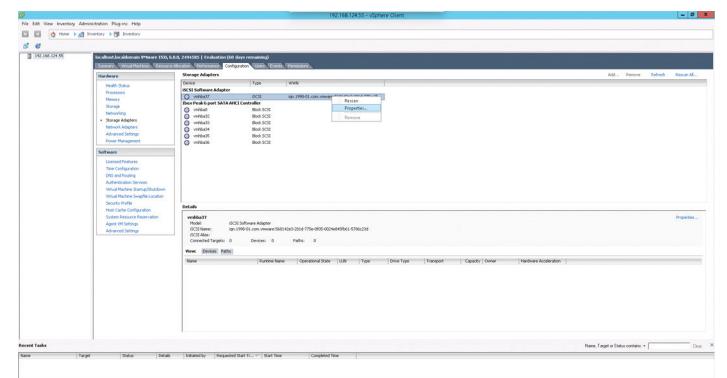


Once initialization completes, the volume (and its NTFS file system) becomes available just like a newly initialized local drive.

VMware

To connect to iRbd-managed iSCSI volumes you need a configured iSCSI software adapter. If no such adapter is available in your vSphere configuration, create one by selecting Configuration → Storage Adapters → Add → iSCSI Software initiator.

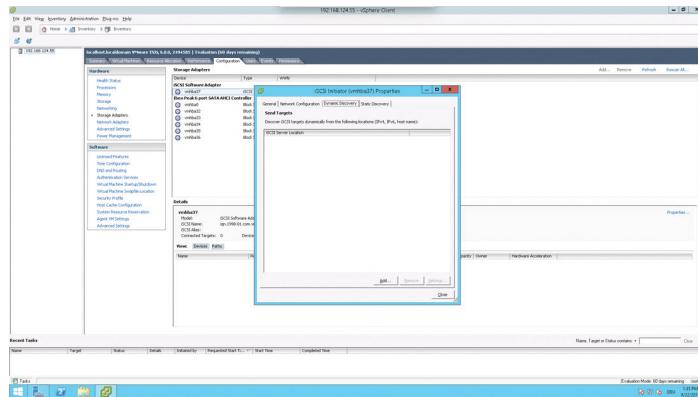
Once available, select the adapter's properties by right-clicking on the adapter and selecting Properties from the context menu:



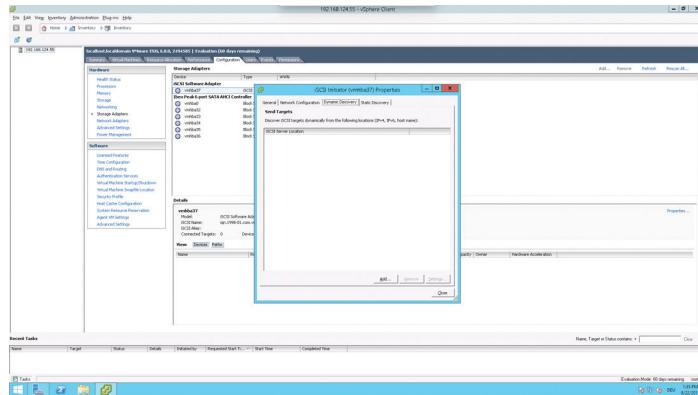
Storage White Paper

SUSE Enterprise Storage 3 and iSCSI

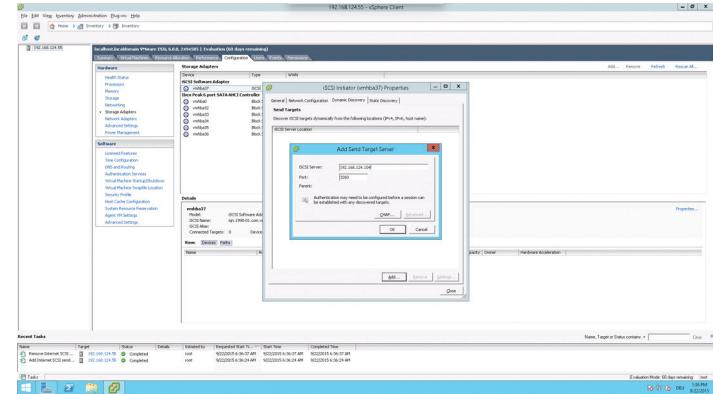
In the iSCSI Software Initiator dialog, click the Configure button:



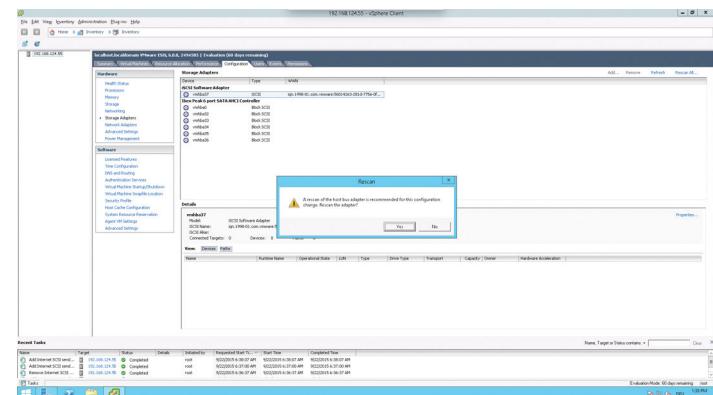
Go to the Dynamic Discovery tab and select Add.



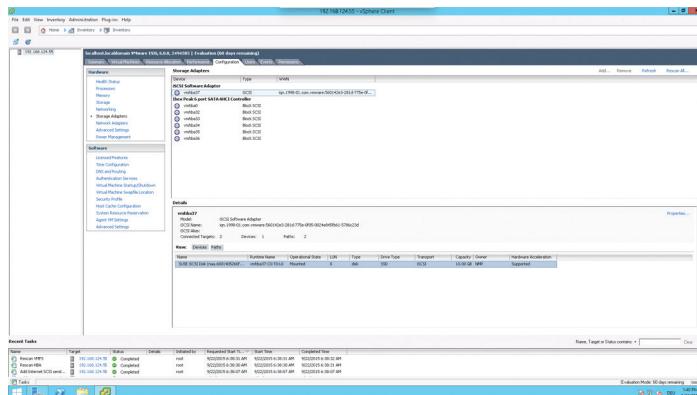
Enter the IP address or hostname of your lrbd iSCSI gateway. If you run multiple iSCSI gateways in a failover configuration, repeat this step for as many gateways as you operate.



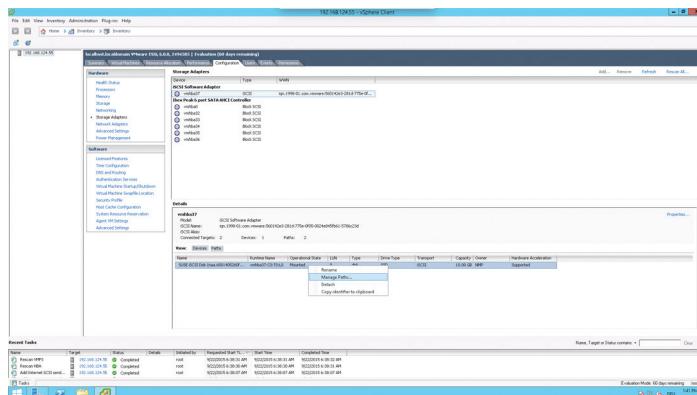
Once you have entered all iSCSI gateways, click Yes in the dialog box to initiate a rescan of the iSCSI adapter:



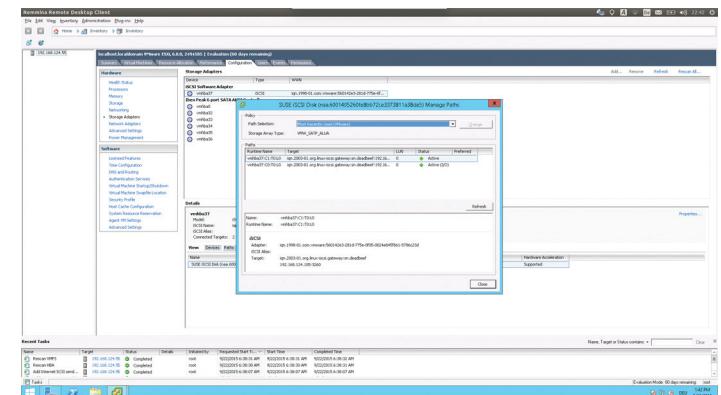
Once the rescan completes, the new iSCSI device appears below the Storage Adapters list in the Details pane:



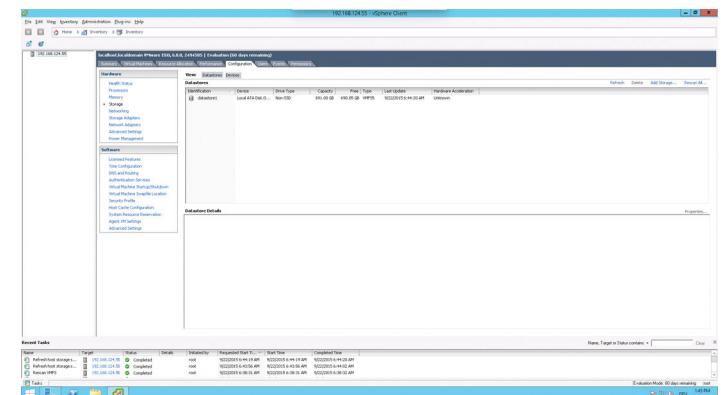
For multipath devices, you can now right-click on the adapter and select Manage Paths from the context menu:



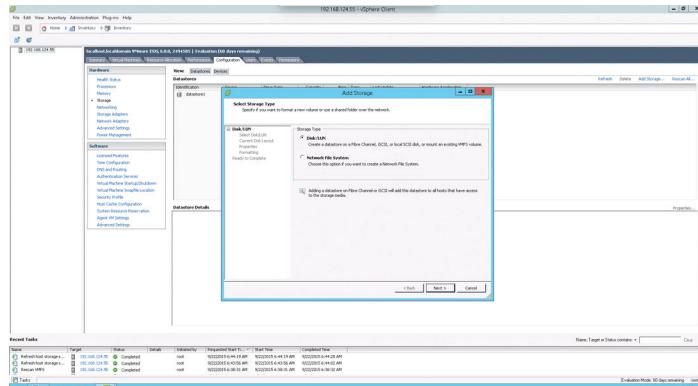
You should now see all paths with a green light under Status. One of your paths should be marked “Active (I/O)” and all others simply “Active”:



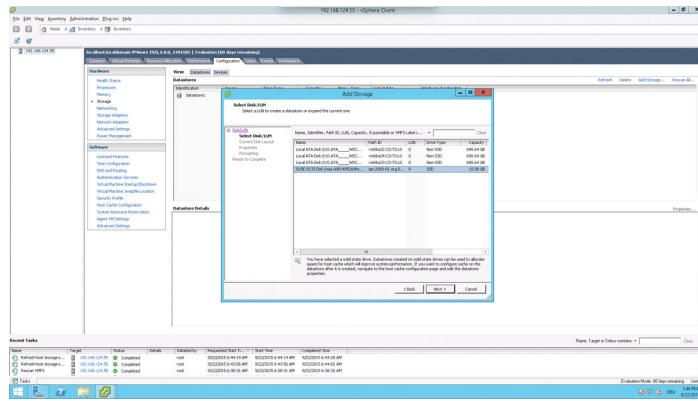
You can now switch from Storage Adapters to the item labeled simply “Storage”:



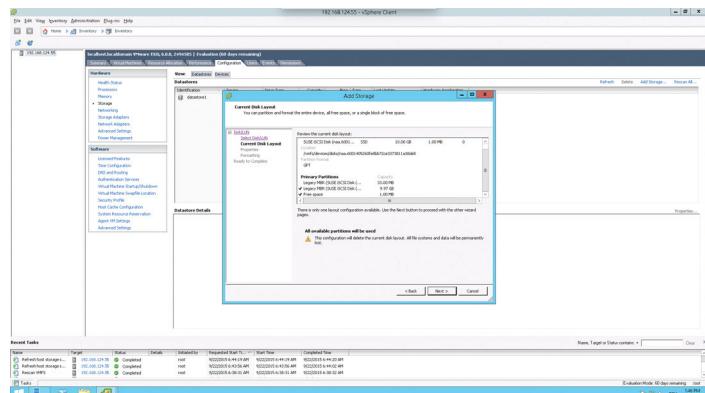
Select Add Storage in the top-right corner of the pane to bring up the Add Storage dialog. Then, select Disk/LUN and click Next:



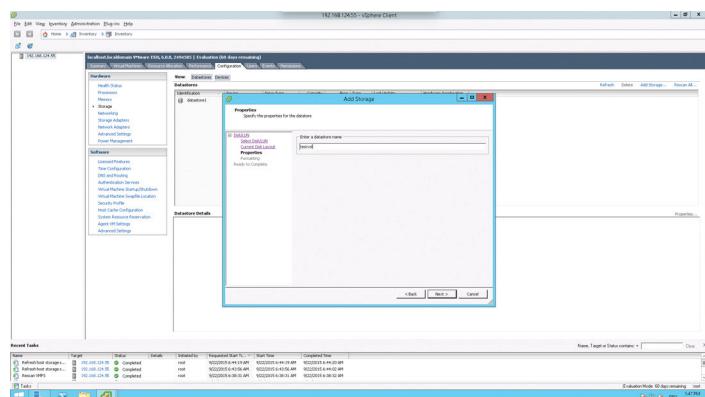
The newly added iSCSI device appears in the Select Disk/LUN list. Select it; then click Next to proceed:



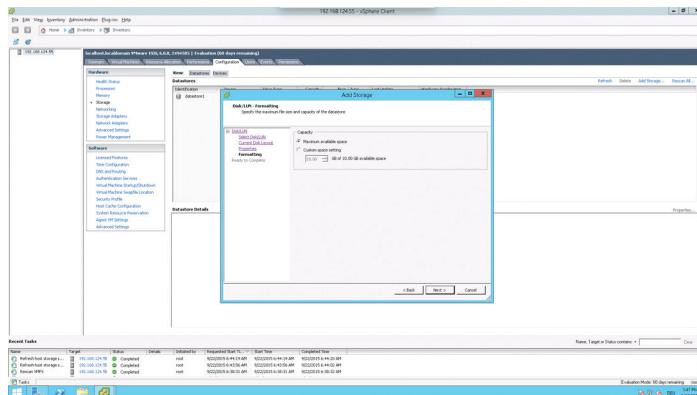
Click Next to accept the default disk layout:



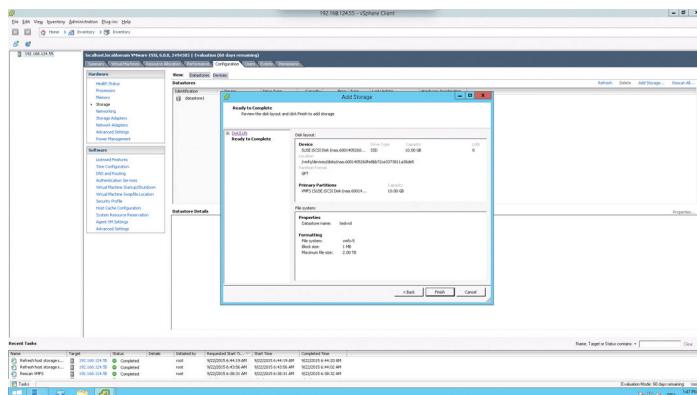
In the Properties pane, assign a name to the new datastore. Then, click Next.



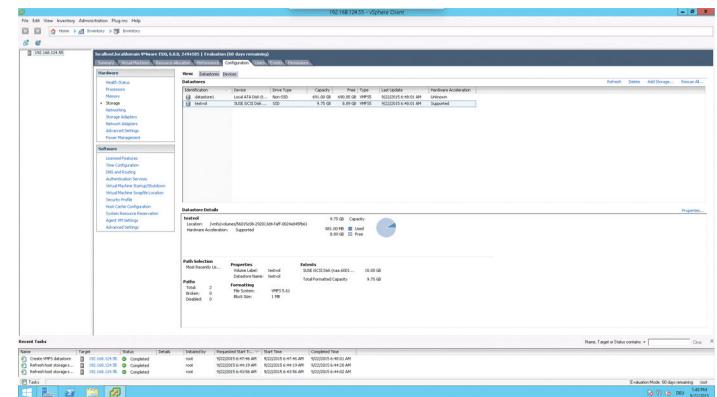
Accept the default setting to use the volume's entire space for the datastore, or select Custom Space Setting for a smaller datastore:



Finally, click Finish to complete the datastore creation:



The new datastore now appears in the datastore list, and you can select it to retrieve details. You are now able to use the lrbd-backed iSCSI volume like any other vSphere datastore.



Conclusion

lrbd is a key component of SUSE Enterprise Storage 3 that enables access to distributed, highly available block storage from any server or client capable of speaking the iSCSI protocol. By using lrbd on one or more iSCSI gateway hosts, Ceph RBD images become available as Logical Units (LUs) associated with iSCSI targets, which can be accessed in an optionally load-balanced, highly available fashion.

Since all of lrbd's configuration is stored in the Ceph RADOS object store, lrbd gateway hosts are inherently without persistent state and thus can be replaced, augmented, or reduced at will. As a result, SUSE Enterprise Storage 3 enables SUSE customers to run a truly distributed, highly-available, resilient, and self-healing enterprise storage technology on commodity hardware and an entirely open-source platform.

Disclaimer: The findings detailed in this whitepaper are developed using SUSE's long-term technical expertise and to the best of SUSE's knowledge and abilities, but on an "as is" basis and without warranty of any kind, to the extent permitted by applicable law.



Contact your local SUSE Solutions Provider,
or call SUSE at:

1 800 796 3700 U.S./Canada
1 801 861 4500 Worldwide

SUSE
Maxfeldstrasse 5
90409 Nuremberg
Germany

www.suse.com