Integrating NetBackup 7.7.x with SwiftStack Object Storage
Table of Contents

Introduction

Architecture
  - Replicas
  - Erasure Coding
  - Multiple Regions
  - Account / Container Services
  - Storage Unit Sizing

Certification
  - SwiftStack - Support with Windows / Linux
  - SwiftStack - Support with UNIX

Prerequisites
  - Cloud Configuration Package
    - Installation
    - Verification Output
  - Device Mapping Package
    - Installation
    - Verification Output
  - Enabling Swift3 Middleware

Configuring Storage Units
  - Cloud Storage Wizard
  - Defining Storage Server Credentials
  - Creating Disk Pools
  - Creating a Storage Unit from Disk Pool Wizard
  - Creating a Storage Unit
  - Creating Storage Unit Groups

Tuning
  - Backup Concurrency
  - Media Servers
  - Master Servers

Migrating from existing storage pools to SwiftStack
  - Duplicating Backups
  - Increasing Duplicate Job Size
Introduction

For storage administrators looking for a more cost-efficient, reliable, scalable, and easier to use solutions for backup, archiving and recovery, object storage is an excellent destination target. Traditionally, backup applications needed disk storage to emulate tape or be locally mounted to the media server. Veritas NetBackup version 7.7.1 or later can now natively utilize object storage using the S3 API. This type of storage offers several attractive capabilities for these workloads, including:

- Nearly unlimited scalability of capacity without reducing performance. No more running into hard capacity limitations of a storage silo, forcing you to recycle your backup data and/or manage it across many namespaces.
- Automatic disaster recovery protection by replicating data offsite. Cluster nodes can exist in multiple physical locations, so you do not need to worry about hauling media to the mountain anymore.
- Cost-efficient hardware infrastructure based on standard servers, Ethernet-based networking, and affordable disk drives rather than on more expensive proprietary solutions. That’s the advantage of software-defined storage.

This runbook will walk you through the specific steps you need to take to optimally configure Veritas NetBackup to utilize a SwiftStack object storage cluster for nearline backup. If you have questions when designing, testing, or deploying SwiftStack with NetBackup, please feel free to contact us, as we are here to help.
Architecting a SwiftStack storage cloud is easy and reference hardware examples are available. In this case, SwiftStack will be used as a direct target for NetBackup Media servers. SwiftStack storage can be deployed alongside your existing backup targets or replace existing targets such as NAS, DAS, PBBAs or even tape.

Starting systems consist of a minimum of three nodes to ensure data availability and durability. Your overall performance can also improve, as all nodes in the cluster can be directly addressed for both backups and restores. SwiftStack is a scale out storage system and all nodes share a common namespace. Having a common namespace eliminates the need for silos that comes with direct
attached storage, SANs or filers that don’t scale. Each node in a SwiftStack cluster is active and performance scales linearly as more storage servers are added.

While not addressed in this runbook, SwiftStack clusters also have the ability to span multiple sites or regions. Unlike typical storage systems, nodes at other sites are active, share the same namespace, and handle cross site replication of data automatically. This makes SwiftStack unlike any other solution on the market with its ability to scale out without boundaries and automatically protect your vital business data from a major disaster.

**Replicas**

With less than 5 nodes, SwiftStack data will always be protected using replication. Replication happens automatically. Using replicas with the 3 node solution referenced above will allow SwiftStack survive the loss of an entire node or multiple drive failures.

A replica policy means that every piece of data given to the object storage is copied multiple times and distributed across many nodes. While replicas increases the amount of raw storage a cluster must have, it has a number of important benefits.

1. Increased availability of data even if network split brain
2. Fastest read / write performance
3. Geographic distribution of data with highly latent networking
4. Lowest time to first byte

**Erasure Coding**

As an alternative to having full replicas of each object, erasure coding is another data protection method that is more space efficient. While similar to RAID, erasure coding protects at an object level, where RAID protects a collections of drives, whereas the loss of more than one drive can result in a complete loss of the collections or RAID group.

Erasure coding is a feature included with SwiftStack, but not necessarily a good selection for NetBackup. Erasure coding requires a minimum of 5 nodes to be enabled so it is more suited for large installations of over 1PB of data. In addition Erasure Coded data can not be replicated between multiple regions/sites if that is required. Lastly, Erasure Coding incurs a performance penalty when used with NetBackup’s 1MB objects, cutting performance in half.
Multiple Regions

Regions are geographically distant areas connected by networking. Replicating data across multiple regions is the ultimate solution in disaster recovery. SwiftStack is able to scale out, storing copies of data in all regions or just selected regions. Data placement within regions is controlled by storage policies and replication happens automatically without the need for schedules, snapshots or other complexity. Storage policies are defined by the storage administrator, but selectable by users and applications on a per bucket basis.

All regions are active, able to distribute data and consume data with no region needing to be read only. The same namespace is common to all regions. This allows media servers in all regions to perform backups and restores locally, thus reducing complexity. A backup done in region 1 can be restored in region 2 by a different media server. This media server will address the local SwiftStack storage and the restore will proceed. If all of the data has not replicated over the wire, the local region will pull the data as needed.

Account / Container Services

Account / Container Services in SwiftStack provide listing information for containers (also known as buckets). Each container has its own database to keep track of object listings. Container performance drops as large numbers of objects are stored in a single container. For this reason SwiftStack recommends assigning backup policies to Storage Unit Groups or to separate Storage Units if use of Accelerator is required.

Account / Container Services should always be assigned to SSDs and not to HDDs for performance.

Storage Unit Sizing

As noted in the previous header the amount of data stored in a Storage Unit should be limited to ensure Container performance. The recommended amount of data to be stored per Storage Unit is 50TB or less. The number of objects in a Container running on SSD will still be quite performant at 50TB.

Calculating number of objects in a storage unit

Since NetBackup objects are 1MB in size, 50TB of data equates to just over 50 million objects. As noted in Account / Container Services, performance drops with the increase in number of objects in a container. With 50 million objects in a container, that container has been tested to ingest data at 200MB/sec.
Certification

SwiftStack Object Storage is tested and certified by Vertias for NetBackup versions 7.7.1 and above. The official Hardware Compatibility Lists (HCL) for NetBackup can be found at the following link: https://www.veritas.com/support/en_US/article.TECH59978

**SwiftStack - Support with Windows / Linux**

<table>
<thead>
<tr>
<th>Cloud Storage Solution</th>
<th>NetBackup Version Containing Plug-in or Vendor-Supplied Plug-in Version</th>
<th>Supported OpenStorage Functionality</th>
<th>Windows Server 2008 x64</th>
<th>Windows Server 2012 x64</th>
<th>Red Hat Enterprise Linux on x86-64</th>
<th>SUSE Linux Enterprise Server on x86-64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Storage [1][2][3]</td>
<td>NetBackup 7.7.1 and later</td>
<td>Accelerator, Accel_VMware, KMS, Opt_Synth</td>
<td>Enterprise Edition</td>
<td>Datacenter</td>
<td>6, 7</td>
<td>11, 12</td>
</tr>
</tbody>
</table>

1. Creating a disk volume via the NetBackup Admin Console or CLI is not supported with this solution. Create the bucket outside of NetBackup using the cloud provider's interface.
2. Requires Cloud Configuration Package version 2.1.2 or later.
3. This is an S3 API-compatible solution.

**SwiftStack - Support with UNIX**

<table>
<thead>
<tr>
<th>Cloud Storage Solution</th>
<th>NetBackup Version Containing Plug-in or Vendor-Supplied Plug-in Version</th>
<th>Supported OpenStorage Functionality</th>
<th>AIX on POWER</th>
<th>HP-UX on IA64</th>
<th>Solaris on SPARC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Storage [1][2][3]</td>
<td>NetBackup 7.7.1 and later</td>
<td>Accelerator, Accel_VMware, KMS, Opt_Synth</td>
<td>6.1, 7.1</td>
<td>11.31</td>
<td>10, 11</td>
</tr>
</tbody>
</table>

1. Creating a disk volume via the NetBackup Admin Console or CLI is not supported with this solution. Create the bucket outside of NetBackup using the cloud provider's interface.
2. Requires Cloud Configuration Package version 2.1.2 or later.
3. This is an S3 API-compatible solution.
Prerequisites

As noted by the Veritas HCL, NetBackup versions 7.7.1 & 7.7.2 require:

- Cloud Configuration Package version 2.1.2 or later
- Device Mapping File Package 1.128 or later
- Swift3 Middleware enabled for SwiftStack

Cloud Configuration Package

At the time of this writing the current Cloud Configuration Package was 2.1.3 and can be found at the following site:


The most up to date package can always be found in the NetBackup HCL referenced in the previous section.

Installation

The following needs to be performed on each Master Server:

1. Copy the existing CloudProvider.xml and CloudInstance.xml file from the following location to an alternate location.

   Default file location in NetBackup:
   In Windows version 7.7.1 - C:\Program Files\Veritas\NetBackup\bin\ost-plugins
   In Windows version 7.7.2 and later - C:\Program Files\Veritas\NetBackup\db\cloud

   In Unix / Linux version 7.7.1 - /usr/openv/lib/ost-plugins
   In Unix / Linux version 7.7.2 and later - /usr/openv/netbackup/db/cloud

   Note: You can revert this file, if needed.

2. Download and extract the zip file package to a temporary folder.

   This will create two files in the temporary location:
   * Readme.txt (this file)
* CloudProvider.xml

3. Stop all NetBackup services.

Windows:

```
<NB_INSTALL_PATH>\NetBackup\bin\bpdown -f -v
```

Unix / Linux:

```
/etc/init.d/netbackup stop
```

4. Copy the new CloudProvider.xml file from the temporary location in step 2, to the location mentioned in step 1.

5. Start all NetBackup services.

Windows:

```
<NB_INSTALL_PATH>\NetBackup\bin\bpup -f -v
```

Unix / Linux:

```
/etc/init.d/netbackup start
```

6. Run the following command. If the command outputs a list of cloud providers, the new CloudProvider.xml was updated successfully. If the command does not output a list of cloud providers or outputs an error message, refer to the troubleshooting section below.

Windows:

```
<NB_INSTALL_PATH>\NetBackup\bin\admincmd\csconfig cldprovider -l
```

Unix / Linux:

```
<NB_INSTALL_PATH>/netbackup/bin/admincmd/csconfig cldprovider -l
```

**Verification Output**

Successful installation of Cloud Package will include SwiftStack as a provider in the listing as seen below.
Device Mapping Package

In order for advanced features like NetBackup Accelerator to function properly, the Device Mapper v1.128 or greater must be installed.

At the time of this writing, the current published version is v1.129 and can be found at:

Linux/UNIX: http://www.veritas.com/docs/000025759
Windows: http://www.veritas.com/docs/000025758

The most up to date package can always be found in the NetBackup HCL referenced in the previous section.
Installation

Note: For the commands in step 4 and later when using Windows, bring up a command window (Start -> Run -> Type "cmd", hit enter), and enter the required commands in that command window.

1. Download and extract the new mappings file package to a temporary folder. This will create three files in the temporary location:
   - Readme.txt
   - external_types.txt
   - external_robotics.txt

2. Copy the external_types.txt file from the temporary location to this location on the Master Server or EMM Server:

   Windows -> C:\Program Files\VERITAS\NetBackup\var\global\n   Unix / Linux -> /usr/openv/var/global
   (For NetBackup High Availability environments, copy the file to the shared disk.)

3. Copy the external_robotics.txt file from the temporary location to this location on the master server, EMM Server, each media server that controls a robot, and each media server from which robot inventories will be run:

   Windows -> C:\Program Files\VERITAS\NetBackup\var\global\n   Unix / Linux -> /usr/openv/var/global
   (For NetBackup High Availability environments, copy the file to the shared disk.)

4. Update the NetBackup Enterprise Media Manager database with the new device mappings version. This only needs to be done once and must be run from the Master Server or the EMM Server. Use the command format below that corresponds to the installed version of NetBackup:

   Windows -> C:\Program Files\VERITAS\Volmgr\bin\tpext -loadEMM
   Unix / Linux -> /usr/openv/volmgr/bin/tpext -loadEMM

5. Restart Device Manager (ltid) on each Media Server.
6. Verify that the version that is now stored in the Enterprise Media Manager database is the same as what is in the file stored on the Media Server:

Windows -> C:\Program Files\VERITAS\Volmgr\bin\tpext -get_dev_mappings_ver
Unix / Linux -> /usr/openv/volmgr/bin/tpext -get_dev_mappings_ver

Verification Output

C:\Program Files\Veritas\Volmgr\bin>tpext -get_dev_mappings_ver
device mappings version in the EMM database is 1.124
device mappings version from the local file is 1.124
Local device mappings file is up-to-date

C:\Program Files\Veritas\Volmgr\bin>tpext -loadEMM

# Restart Service LTID

C:\Program Files\Veritas\Volmgr\bin>tpext -get_dev_mappings_ver
device mappings version in the EMM database is 1.129
device mappings version from the local file is 1.129
Local device mappings file is up-to-date

Enabling Swift3 Middleware

S3 emulation must be enabled within SwiftStack for NetBackup to be able to access the storage buckets. Follow this guide to enable the Swift3 middleware.

1. Choose Manage for the cluster that S3 API emulation needs to be enabled.
2. Click the **Middleware** icon on the left panel.

3. Click on the **Swift3 -- S3 Emulation Layer** towards the bottom of the page.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swauth</td>
<td>inactive</td>
</tr>
<tr>
<td><strong>Swift3 -- S3 Emulation Layer</strong></td>
<td>inactive</td>
</tr>
<tr>
<td>Swift3 Keystone Integration</td>
<td>inactive</td>
</tr>
</tbody>
</table>

4. Click the **Enabled** checkbox at the top of the page. Scroll to the bottom of the page and Click Submit at the bottom of the page. Other S3 configurables can be left at default settings.

5. Swift3 should now show as enabled.
6. The changes will now need to be pushed to all nodes. Click the RED, “Click Here to Deploy” button at the top of the page.

7. Click the **Deploy Config to Swift Nodes** button to push the configuration.

8. The progress will progress from Pending to Job Finished. The system is now ready to accept S3 API requests from NetBackup.
Pending Jobs
Config Deploy in progress...

Job Finished
Config Deploy finished successfully.
Configuring Storage Units

NetBackup stores data in storage units which can be created in the three step process described below.

Cloud Storage Wizard

Netbackup has a wizard that will lead the administrator through the same three steps as defined below. This can be used to setup the first Storage Unit, but subsequent Storage Units should be added by defining additional Disk Pools.

Defining Storage Server Credentials

In order to start using SwiftStack Object Storage, the credentials for the user account to be used to store data must be defined. Multiple accounts can be added to NetBackup if needed.

1. Expand **Media and Device Management**.
2. Expand **Credentials**.
3. Right click on **Storage Servers**.
4. Click on **New Cloud Storage Server**.

Expand **Media and Device Management**.

Expand **Credentials**.

Right click on **Storage Servers**.

Click on **New Cloud Storage Server**.
2. Click **Next** on the first screen. On the “Add Storage Server” screen either find **SwiftStack** at the bottom of the list or type **SwiftStack** into the search field. Click on **SwiftStack** then click the **Next** button.

3. If this is the first time adding SwiftStack Storage the Service host selection will be empty. Click **Add Cloud Storage** to add your SwiftStack cluster.
4. On the Add Cloud Storage Screen enter in the FQDN or IP address of the SwiftStack cluster in the Service host field. If a load balancer is used, this should be the address of the load balancer.

Storage server name is a user defined descriptor of the cluster. By default it will fill in with my-[service host], however this can be changed as the field is a descriptor.

HTTP port, HTTPS port, and Endpoint access style should not be changed in a default configuration.

Region Settings should not be used and left blank.

Click Ok when finished editing settings.
5. After adding the service host, the account and secret key must be entered. **Access key ID** is the SwiftStack account name which can be found by logging into the SwiftStack controller and clicking on User & Accounts. The **Secret access key** can be found by clicking the “Show S3 API Key” button. Note: Multiple accounts can be used by rerunning this process, selecting the Service Host, entering a new Storage Server Name, Access, and Secret Key.

Review the **Advanced Settings** before moving on to the next screen.
6. In Advanced Server Configuration the Use SSL determines whether data transported to SwiftStack via HTTP or HTTPS. If using HTTPS click both Use SSL and the Data Transfer radial button. If the SwiftStack cluster is configured only for HTTP access, remove the checkmark from Use SSL.

Optionally: If a proxy is required that information can be inputted in this screen.

HTTP Headers should not be used and remain blank.

Click Ok to return to the previous screen.

7. Once all settings are complete click Next to move to the Encryption selection screen. Setup of NetBackup Encryption is beyond the scope of this manual, however, NetBackup AES-256 encryption is done server side and supported by SwiftStack.

8. Click Next on the summary screen. NetBackup will now test the credentials and configure the cloud storage credentials.

After success, the option to Close or continue on to creation of Disk Pools by clicking Next is given. Clicking Next will continue the wizard process.
Creating Disk Pools

Disk Pools map NetBackup storage to a bucket or container in SwiftStack. If a storage policy other than the system default is desired, the NetBackup administrator should precreate said buckets. A disk pool can then be mapped to an existing bucket, and the storage policy governing data replication, erasure coding, disk tier or geolocation will be obeyed.
1. Expand “Media and Device Management”
   Expand “Devices”
   Right click “Disk Pools”
   Click on “New Disk Pool”

2. Select “Cloud Storage (swiftstack_raw)”
   Click “Next” button

3. If multiple SwiftStack clusters or Accounts have been defined select the desired Storage Server
4. If the bucket for backups has been preconfigured skip to step 6
Otherwise, click “Add New Volume” to create a bucket.
Note: If a pre-created bucket is not listed ensure it follows NetBackup naming conventions

5. In this step one or more buckets can be created. Input a bucket name ensuring it meets with
the NetBackup naming convention. Click add until all buckets are listed in the “Cloud Volume
Name” listing. Then Click “Create”
6. Select only one bucket, buckets can not be aggravated at the Disk Pool level. And do not use a single bucket for multiple Disk Pools or Storage Units. Click “Next” to continue.

7. Name the Disk Pool and click continue. The high and low watermarks do not matter as each Disk Pool is listed at 8 Petabytes of available storage. Click “Next” to continue.
8. NetBackup will now create the Disk Pool. Once completed the option to continue on to creating a storage unit for this disk pool is presented. It is recommended to continue on to this step.

Creating a Storage Unit from Disk Pool Wizard
Continuing from the Disk Pool Wizard, the creation of a storage unit is straightforward and simple. The screens vary slightly from the manual process. If the manual process is desired, please continue with the next section - Creating a Storage Unit

1. Continuing from the Disk Pool Wizard, a Storage Unit Name of automatically populated. This can be altered as desired, however the default is to append “-stu” to the Disk Pool Name. Media servers allowed to use this disk pool can be chosen.  
   Maximum concurrent jobs is defaulted to 1. Object Storage handles hundreds of connections easily. This number should be increased to support a number of simultaneous backup streams. Starting with 100 will allow proper testing.  
   Maximum fragment size has no effect on object storage and should be ignored.  
   Clicking “Next” will complete the wizard.

Creating a Storage Unit

A Storage Unit is the base unit for which backups are targeted. While it is possible to point multiple storage units to the same cloud disk pool, this is not recommended. A storage unit should be mapped to a disk pool on a one to one basis.

1. Expand NetBackup Management
Expand Storage
Right Click Storage Unit
Click New Storage Unit
2. Enter a Storage Unit Name
   Select Disk Type of “Cloud Storage (swiftstack_raw)”
   Select the Disk Pool to be used
   Increase Maximum concurrent jobs, start with 10, max at 100
   Maximum fragment size has no affect on object storage and should be ignored.
   Click “Ok” to finish.
Creating Storage Unit Groups

Creating Storage Unit Groups allows for the pooling of Storage Units into a single larger structure. While a Storage Unit list in NetBackup as having 8 PetaBytes of storage space in actuality the amount of data that should be stored in a single storage unit is much smaller. Please see Storage Unit Sizing under Architecture for further details.

While grouping Storage Units increases the amount of storage available to a backup policy the trade-off is that NetBackup Accelerator is not compatible with Storage Unit Groups configured in a round robin fashion. If Accelerator is to be used those policies must be connected to an ungrouped storage unit. For more on Accelerator please see Accelerator under Architecture.

1. Expand NetBackup Management
   Expand Storage
   Right Click Storage Unit Groups
   Click “New Storage Unit Group”
2. Give the Storage Unit Group a name.
   Select multiple Storage Units to comprise the group.
   Select Round Robin so that the storage units are filled evenly
   Note: Prioritized & Failover make little sense for use in a SwiftStack cluster
   Click “OK” to complete group creation.
Tuning

Performance of backups is governed by the speed of the object storage system, the number and capability of media servers, network speed, and a number of other factors. Following best practices for NetBackup and SwiftStack will optimize performance.

Backup Concurrency

Object Storage excels with an increase of simultaneous backup streams. Where traditional filers drop dramatically in performance with each connection, object storage can scale linearly with proper equipment and load balancing.

A single backup job may proceed at 20MB/s, since NetBackup uses only a single connection per backup job. A second backup job will also progress at 20MB/s, but the aggregate for the two jobs is 40MB/s. In order to achieve minimal backup windows, many jobs should run concurrently. Fifty simultaneous jobs on a single media server can max a 10Gb network link. Just be sure to back off if primary storage is affecting application performance during backups.

In order to allow a media server to run more than one concurrent jobs, please make the configuration changes in the next section.

Media Servers

In order for media servers to support multiple backup streams the following changes must be configure. Otherwise media servers will restrict throughput to a single backup by default.

The following must be completed for each media server that intends to backup to a cloud storage unit.

1. Expand NetBackup Management
   Expand Host Properties
   Expand Media Servers
   Right click on the desired Media Server
   Select Properties
2. Select **Scalable Storage** from the left panel
   
   Increase the Total available bandwidth to the full network bandwidth of the media server. In our case with 10Gb networking, the value was adjusted from 102400 (default) to 10240000. Increase the **Maximum concurrent jobs** from 1 to 10, the max should be 100. This will allow the media server to take full advantage of the concurrency of object storage. Tune this setting as needed to meet the ability of the media server to handle client backup streams.
   
   Click “Ok” when finished
Master Servers

In order for media servers to support multiple backup streams from within a job the following changes must be configure. This will allow an agent to backup multiple drives within a target machine at the same time. Reducing the time per agent and improving throughput.

1. Expand *NetBackup Management*
   
   Expand *Host Properties*
   
   Click on *Master Server*
   
   Right Click Master server, choose *Properties*
2. Select *Global Attributes* from the left panel
Increase *Maximum jobs per client*
Start with 5 and increase up to a max of 10
Migrating from existing storage pools to SwiftStack

Migrating data from existing existing Storage Units to new Cloud based Storage Units is a simple process. This can be useful in a number of scenarios:

- Aging data from expensive disk pools to economical Cloud Storage
- Duplicating data from Cloud Storage to Tape for DR compliance
- Restoring data from Offsite tape to Cloud Storage

All of the above scenarios will follow the same process outlined below:

Duplicating Backups

The following assumes that the backup is currently loaded into the NetBackup Catalog. If this is not the case please follow the article below to import an expired backup back into the catalog

Import Expired Media to Catalog

1. Expand **NetBackup Management**
   Click on Catalog

2. Select **Action** "Duplicate"
   Select the current location of the backups under **Media**
   Select the **Date Range**
   Select **Search Now**
   Results will populate below

3. Highlight the desired backups to Duplicate
   Right Click and Select “Duplicate”
4. If only one copy of the data is desired, leave Copies at 1
   Check the checkbox of Copy 1 if it is to be made the new Primary Copy
   Choose the Cloud Storage Unit to store the backup in
   If desired, change the Retention time
   Preserve Multiplexing is not required. It is for tape to tape transfers.

5. Confirm your request to proceed.
   Click “OK”
6. The job can be monitored in the Results Tab or the Activity Monitor

7. Once the job is complete, the original backup can age out according to the original retention policy or can be expired immediately. To expire immediately:
   Select the backups to Expire
   Right Click the backups
   Click on “Expire”
   Select “Yes” on the next screen
Increasing Duplicate Job Size

By default NetBackup limits the size of a duplication job to 100GB. This may be too small for most duplication jobs involving LTO tape. To change this value perform the following:

1. Expand NetBackup Management
   Expand Host Properties
   Click on Master Server
   Right Click Master server, choose Properties
2. In the Left Panel select **SLP Parameters**
   Change the **Maximum size per duplication job** from 100GB to the required value for the environment