

SWIFT TLC (Tape Library Connector) Quick Start Guide

Index of Contents

| | |
|--|---|
| Scope of this document | 1 |
| Before you start..... | 1 |
| Setup preparation..... | 2 |
| All in one SWIFT TLC single node setup | 3 |
| SWIFT TLC multi node setup..... | 7 |
| Uninstall SWIFT TLC | 8 |

Scope of this document

The scope of this document is to provide a quick start installation guide to make evaluation of this software as easy as possible. To be able to follow this document you need to have some Linux administration knowledge (e.g. create and mount volume, monitor Linux processes, edit and verify Linux Service configuration files).

This document describes just how to setup a running system. It does not describe the architecture of the SWIFT TLC Software. If you are looking for more information about the architecture of SWIFT TLC please read the SWIFT TLC Technical Overview document.

The first part will describe the setup of the minimum environment required to use this software on a single server system. All required SWIFT and TLC services will be installed and configured on one single server. After the setup you will have a running OpenStack SWIFT Object Storage storing all objects on tape.

At the end of the document we will provide some advice to integrate SWIFT TLC in an existing SWIFT cluster to run a SWIFT Cluster with Disk Storage Nodes and Tape Storage nodes altogether.

Before you start

The SWIFT TLC software is still under development in an early stage. Although the described functionality is completely implemented and tested already there are still a lot of improvements necessary and unexpected errors including data corruption can occur occasionally any time. So please consider this for your intended application. We offer this software as is and any usage of this software is on your own risk.

The SWIFT TLC software is intended to run on CentOS 7 and OpenStack Juno release. Other Linux Distributions and OpenStack release are possible in general but require modified installer scripts first.

To be able to setup our SWIFT TLC software there is at least one physical server and one tape library necessary.

The Setup is divided in three parts:

- **TLC package (tlc.0.9.x.tar.gz):** This package contains the majority of our software. It mainly will install a Virtual File System (VFS) which allows reading and writing of files directly to a tape library. Our TLC VFS has quite some SWIFT specific implementation features inside but it will be completely transparent to SWIFT. All standard SWIFT services (like Auditing, Replication) are fully supported by this TLC software.
- **SWIFT installer package (CentOS-SWIFT.0.9.3.tar.gz):** To simplify the setup of the required OpenStack SWIFT services we provide an installer tool which could install all of them automatically and configure SWIFT and TLC to work together right out of the box. This installer allows as well supporting a manual setup of a SWIFT Cluster. This installer is optionally. Alternatively you could use other SWIFT deployment tools like RDO/packstack as well.
- **TLC Tape Auditor package (SWIFT-Tape-Auditor.0.9.3.tar.gz):** This package contains a specific extension for the Standard SWIFT Auditor to support a tape specific object auditing.

Each package contains its own installer script (install.sh). Beside a few specific installation options every install support at least to install (./install.sh) and to uninstall (./install.sh uninstall) this package.

The TLC package must be installed prior the installation of TLC Tape Auditor package

The minimum hardware requirements to evaluate the SWIFT TLC software are:

- 1 server system (recommended Xeon E5, minimum 8GB RAM, no hardware RAID controller required)
- 1 SAS 6Gb HBA with tape support (preferred with LSI 92xxx chip, no RAID support, needs to be released for LTFS support), HBA needs external SAS connector
- 1 Tape Library (preferred HP, IBM, DELL, Fujitsu) with minimum 1 LTO tape drive (recommended 2 LTO6 SAS tape drives), supported tape drive vendors: HP, IBM
- Server should have at least 1.5TB local disk storage (SSD Flash disks recommended e.g. 3 x 512GB) available for TLC Cache
- Server should have 2 network interfaces

Setup preparation

- Install CentOS 7 and select CentOS the Workstation profile for setup
- Create a storage volume for TLC Metadata Cache and consider 10GB for each 100000 objects (recommended size 60GB as RAID1 on SSD storage), format with XFS and mount TLC Metadata Cache volume to folder /opt/VS/vsCache/meta
- Create a storage volume for TLC Data Cache with minimum size of 1TB (recommended size 1,5TB as RAID0 on SSD storage), format with XFS and mount TLC Data Cache volume to folder /opt/VS/vsCache/diskCache
- Verify that the TLC cache volumes are mounted correctly and are formatted with XFS

```
[root@tlc01 ~]# mount | grep vs
/dev/md126 on /opt/VS/vsCache/diskCache type xfs
(rw,relatime,attr2,inode64,sunit=1024,swidth=3072,noquota)
/dev/md127 on /opt/VS/vsCache/meta type xfs
(rw,relatime,attr2,inode64,sunit=1024,swidth=2048,noquota)
```

- Attach the tape library to an external SAS connector of the server (every tape drive needs to be connected the server) and add a few tapes to the library (recommended to start just with a few tapes only, you can add more tapes later any time)
- Verify that the tape library is connected right to the TLC server. This can be verified by running command “*lsscsi -g*”. The output needs to list a Media Changer Device (mediumx) and each tape drive (tape). (If the lsscsi command is not available it could be installed with the command “*yum -y install lsscsi*”)

```
[root@tlc01 ~]# lsscsi -g
[1:0:0:0]    tape      HP          Ultrium 5-SCSI   Z58B   /dev/st0   /dev/sg8
[1:0:0:1]    mediumx   BDT        FlexStor II      4.90   /dev/sch0  /dev/sg9
[1:0:1:0]    tape      HP          Ultrium 5-SCSI   Z58B   /dev/st1   /dev/sg10
```

- Configure Network Time Synchronization (NTP) to ensure that ALL SWIFT servers use the same time settings.
- Ensure that the yum installer can connect to the internet. If an internet proxy is required add the proxy server to the yum configuration file.

```
[root@tlc01 ~]# cat /etc/yum.conf | grep proxy
proxy=http://172.16.56.2:3128/
```

All in one SWIFT TLC single node setup

The simplest way to evaluate the SWIFT TLC software is to use our auto installer. This installer will install and configure our software and all required OpenStack components on one sever almost automatically.

All stated Linux commands should be executed as user root! So please login as user root or change existing user to root “*sudo su root*”

1. Install the TLC software by executing these steps:
 - copy the package *tlc.0.9.x.tar.gz* to one local folder and unpack the files with command “*tar xvf tlc.0.9.x.tar.gz*”
 - Change to folder TLC and start the installer with the parameter install “*./install.sh install*”
 - Follow the instructions on the screen.
 - Be careful to select the right tape vendor for the installed tape drives in your tape library. Possible options are HP or IBM

```
#####
#####
#####
WELCOME

TO ValueStor INSTALLATION WIZARD

You are

about to 'fresh install' VStor

#####
#####
#####
[INFO] Checking if any tape mounting.
[INFO] No tape is mounted.
[QUESTION] Which vendor of LTFs Edition to be installed?
0):  IBM LTFs SE Edition
1):  HP LTFs SE Edition
```

Input 'C' to cancel the installation.(0/1/C):

- If the TLC cache volumes are mounted to the right mount points and yum can access the internet repositories everything else will be done automatically.

```
[INFO] You are selecting to install the 'HP LTFS SE' Edition.  
[SUCCESS] Found an XFS volume was mounted at /opt/VS/vsCache/diskCache, will  
use this directory as the diskCache  
[SUCCESS] Found an XFS volume was mounted at /opt/VS/vsCache/meta, will use  
this directory as the metaCache
```

- If installation went well you will see message telling that you are ready to start the TLC software now.

```
[SUCCESS] CONGRATULATIONS!! The Fresh Installation has been completely done!  
Installed version: '0.9.3'  
[INFO] Please check everything ready then start the VStor service with  
command: '/etc/init.d/vs start'
```

- The installer will configure TLC to startup after every reboot automatically.
- Start the TLC software manually with the command “/etc/init.d/vs start”. You can execute the command “/etc/init.d/vs status” any time to verify if the TLC software is running
- The TLC software starts to inventory all tapes from the tape library. It will also format ALL tapes from the library with the LTFS format. This happens in the background so you can start using the TLC software instantly and there is no need to wait until all tapes got formatted. Nevertheless as it can take a few minutes just to format one tape it could take easily several hours to format all tapes initially with LTFS if you have many tapes inserted into the library already. So to start quickly we recommend inserting just up to 3 tapes for the beginning.
- Wait 5mins after starting the TLC services and verify that TLC is running well by executing the commands “df -h | grep vs”. It should list the TLC VFS volume (/srv/node/vsnode) with a capacity of several TB. The shown VFS capacity is the capacity of the sum of all currently LTFS formatted tapes.

```
[root@tlc01 TLC]# df -h | grep vs  
/dev/md126      60G   33M   60G   1% /opt/VS/vsCache/meta  
/dev/md127      1.4T   34M   1.4T   1% /opt/VS/vsCache/diskCache  
vfsclient       7.9T    0   7.9T   0% /srv/node/vsnode
```

2. Install and configure OpenStack Swift services

- After installing the TLC package the next step is to install and configure the required OpenStack SWIFT components. This can be done by using our SWIFT installer package
- Copy the package CentOS7-SWIFT.0.9.3.tar.gz to a local folder and unpack it “tar xvf CentOS7-SWIFT.0.9.3.tar.gz”
- Start the installer by executing command “./install.sh” from folder CentOS7-SWIFT.
- After starting the installer you need to select the installation option first. To perform the SWIFT All in one installation select **Option 0**

```
[root@dsn01 auto]# ./install.sh  
[QUESTION] Please Select SWIFT Installer Option:  
1): SWIFT All in one  
2): SWIFT PROXY + Keystone  
3): SWIFT Storage Node  
4): Uninstall  
Input 'C' to cancel and exit.(1/2/3/4/C):
```

- As a next step you need to enter the IP address of your server.

```
[QUESTION] Please enter 'IP address' for tlc01:
[INFO] This IP address will be used in several configuration files: '1.
/etc/rsyncd.conf; 2. /etc/swift/proxy-server.conf; 3.
/etc/swift/account,container,object-server.conf'
(IP address): 172.16.56.5
```

- The installer will install all OpenStack SWIFT service from the current Juno release, the Keystone Service for authentication, the OpenStack Dashboard Horizon and all services which are required to run these services (e.g. MariaDB, RabbitMQ, memcache, Python Clients ...).
- The installer will build an initial SWIFT ring file and add the TLC VFS volume
- After the installer has finished successfully the server is set up as a standalone SWIFT Object Storage which uses Tape as the storage backend. It should run out of the box. No further configuration is necessary to run the SWIFT Object Storage in default mode.
- To start the SWIFT server enter “swift-init all start”
- As a next step you should perform a quick functional test that the system is up and running and uses the TLC to write all objects to tape.

- Execute the script /root/admin-openrc.sh to set a few environment variables which enable that you could use the SWIFT CLI tool to access the SWIFT OS with the right credentials. The user credentials are setup as default user= admin, password=admin

- The script admin-openrc.sh will set the following variables

```
export OS_USERNAME=admin
export OS_TENANT_NAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL=http://<your_server_ip_address>:5000/v2.0/
export OS_REGION_NAME=RegionOne
```

- Run the command “swift stat” to verify that you can access to the SWIFT server. You should see an output like this.

```
[root@RWCLD01]# swift stat
Account:
AUTH_7b0e36c644484ald9a81055ca6898b29
Containers: 0
Objects: 0
Bytes: 0
Containers in policy "policy-0": 0
Objects in policy "policy-0": 0
Bytes in policy "policy-0": 0
X-Timestamp: 1429187966.02466
X-Trans-Id: txf76f195167434fde98701-00552fbd7c
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

- Upload a first test file using the SWIFT CLI by executing these commands to verify that the file got uploaded

```
[root@RWCLD01]# dd if=/dev/urandom of=test.bin bs=1M count=20
iflag=fullblock
20+0 records in
20+0 records out
20971520 bytes (21 MB) copied, 1.42911 s, 14.7 MB/s
[root@RWCLD01]# swift upload test test.bin
test.bin
```

```
[root@RWCLD01]# swift stat test test.bin
Account: AUTH_7b0e36c644484a1d9a81055ca6898b29
Container: test
Object: test.bin
Content Type: application/octet-stream
Content Length: 20971520
Last Modified: Thu, 16 Apr 2015 13:58:26 GMT
ETag: 911d786b99a1b7feb505d4da53d36a50
Meta Mtime: 1429192676.009388
Accept-Ranges: bytes
X-Timestamp: 1429192705.01800
X-Trans-Id: txf8b7ee9479d74513850aa-00552fc038
```

- v. Optionally: Wait approximately 10mins and verify that the file got written to tape as well. Run this command to verify it. It should list one line starting with ltfs: which indicates that one tape is currently used by the TLC software (size will differ). The tape inventory process needs to be finished before the first objects get written to tape. Each tape gets formatted with LTFS so you should consider 5min for each tape. This means if you have 10 tapes the inventory take approx. 50min and you need to wait 50mins before you can check if the objects got written to tape.

```
[root@tlc01 ~]# df -h | grep ltfs
ltfs:/dev/st0 1.4T 20G 1.3T 2% /opt/Vs/vsMounts/N00169L5
```

- vi. As a next test you should download the uploaded file back to your system and check the md5 checksum to verify that the source file and downloaded file are identical. To run this test you could execute the following commands:

```
[root@RWCLD01 ~(swift)]# mv test.bin test.bin.src
[root@RWCLD01 ~(swift)]# swift download test test.bin
test.bin [auth 0.207s, headers 0.264s, total 0.464s, 81.451 MB/s]
[root@RWCLD01 ~(swift)]# md5sum test.bin.src test.bin
911d786b99a1b7feb505d4da53d36a50 test.bin.src
911d786b99a1b7feb505d4da53d36a50 test.bin
```

- vii. Alternatively you can use the OpenStack Dashboard Horizon as a GUI to access the SWIFT object storage. Just open a web browser and login at “http://<SWIFT node>/dashboard” as user: admin with password: admin

3. Install and configure SWIFT TLC tape auditor

- The normal SWIFT auditor service will continually read all objects to check if they are still consistent. This would result in that the Auditor would be permanently trying reading individual objects back from tape. Because of the nature of tape storage as a sequential storage device this would not work well for Tape and results in a horrible performance drop. To avoid this issue we developed a specific tape auditor tool for SWIFT. Please refer to the separate SWIFT Tape Auditor Overview document if you are interested in more background information about this tool.
- The package Tape Auditor (SWIFT-Tape-Auditor.0.9.3.tar.gz) contains our specific Tape Auditor for SWIFT. Copy it to a local folder and unpack it with “tar xvf SWIFT-Tape-Auditor.0.9.3.tar.gz”. Change to the folder SWIFT-Tape-Auditor and start the installer script “./install.sh install”
- The installer will ask to select the right tape drive vendor.

```
[QUESTION] Which platform of TapeVerify Tool to be installed?
0): TapeVerify Tool for IBM platform
1): TapeVerify Tool for HP platform
```

```
Input 'C' to cancel the installation.(0/1/C): 1
[INFO] You are selecting to install TapeVerify Tool for HP platform.
```

- The installer will install and configure the auditor, adds a Python component (/usr/lib/python2.7/site-packages/swift_vs_backend-0.1-py2.7.egg) to the Python SWIFT Python repository and modifies the standard SWIFT Object Auditor python source file (/usr/lib/python2.7/site-packages/swift/obj/auditor.py) to use our auditor for auditing files in conjunction with our TLC software.
- The tool will run out of the box. There are no further configuration settings necessary.
- After the installation of this tool the installation and configuration of the complete TLC software is completed and you can start to use the SWIFT TLC Object Storage like every other SWIFT Cluster without any functional limitations.

SWIFT TLC multi node setup

- If you want to add a TLC node to an existing SWIFT cluster or if you want to use an existing SWIFT proxy server it will become necessary to increase one timeout setting (node_timeout) in the proxy server configuration file (<swiftproxy>:/etc/swift/proxy-server.conf) to prevent communication timeout issue between the SWIFT proxy server and the TLC node. The read response time from a TLC node can be significantly longer than from a disk storage node, because of the nature of tape as a sequential device. The timeout value should be increased to 300s.

```
[app:proxy-server]
...
allow_account_management = true
account_autocreate = true
node_timeout = 300
```

- The described TLC single node setup is primarily intended a quick easy installation which takes just a few minutes to begin with evaluating of our TLC software
- Nevertheless the real power of SWIFT comes from building large clusters of storage nodes
- As the SWIFT TLC is totally transparent to SWIFT it can be easily integrated in any existing clusters regardless of the used storage back-ends of the existing nodes.
- Using SWIFT storage policies you can easily control which data shall reside on the TLC nodes and which does not.
- It is possible to add several TLC nodes to one SWIFT cluster
- To simply building up a multi node SWIFT cluster with TLC nodes our SWIFT installer allows two different installation options beside the single node all in one installation option. After starting our SWIFT installer script you are able to select:
 - SWIFT Storage Node
 - will just install the SWIFT Storage Services (e.g. Object, Account, Container, Replicator, Auditor)
 - user need to take care of creating and deploying the ring files
 - the SWIFT Storage Node services are mandatory to be able to run the TLC Software on that node
 - user need to take care of increasing the node_timeout setting for the SWIFT proxy server
 - SWIFT Proxy nodes
 - allows to install quickly minimum setup to run a SWIFT Proxy server including Keystone for user authentication and Horizon as dashboard

- does not install any other SWIFT Storage Service (like Object, Account, Container, Replicator, Auditor)
- if user wants to add these SWIFT Storage services to this nodes as well, he can run this script again and select the SWIFT storage node installation option that time
- user need to take care of creating and deploying the ring files
- user need to take care of increasing the node_timeout setting for the SWIFT proxy server

Uninstall SWIFT TLC

- Please follow these steps in this sequence to uninstall the SWIFT TLC packages
 1. Stop the SWIFT services: *"swift-init all stop"*
 2. Stop the TLC server: *"/etc/init.d/vs stop"*
 3. Uninstall the Tape Auditor: *"cd SWIFT-Tape-Auditor/; ./install.sh uninstall"*
 4. Uninstall the TLC: *"cd TLC/; ./install.sh uninstall"*
 5. Uninstall the SWIFT service: *"cd CentOS7-SWIFT/; ./install.sh"* and select *Option 4) Uninstall*

```
[root@dsn01 auto]# ./install.sh
[QUESTION] Please Select SWIFT Installer Option:
1): SWIFT All in one
2): SWIFT PROXY + Keystone
3): SWIFT Storage Node
4): Uninstall
Input 'C' to cancel and exit.(1/2/3/4/C):
```