Haramohan Sahu

This document talks about the basic Storage terminology

Basics Of Storage

Contents

[What is an aggregate? 3](#_Toc92665142)

[How is an aggregate created? 3](#_Toc92665143)

[HA CONSIDERATIONS 4](#_Toc92665144)

[EXAMPLES 4](#_Toc92665145)

[Understanding aggregate and lun: 8](#_Toc92665146)

[What is the use of aggregate? 10](#_Toc92665147)

[An aggregate is made of Raid Groups. 11](#_Toc92665148)

[**what happen when we give a volume directly to the user instead of LUN?** 12](#_Toc92665149)

[**What was the difference between these two?** 12](#_Toc92665150)

[Is it possible to give a volume directly in a san environment? 12](#_Toc92665151)

[**Aggregate raw space and volume-providing file system, then what about Lun?** 13](#_Toc92665152)

[**By this hierarchy what kind of function lun would provide?** 13](#_Toc92665153)

[From netapp perspective, volume will provide filesystem. 14](#_Toc92665154)

[Here the file system you are talking about is host side filesystem or WAFL ? 14](#_Toc92665155)

[What's the difference between a LUN and a Volume? 15](#_Toc92665156)

[What a FlexVol volume is 16](#_Toc92665157)

[What is a qtree? 16](#_Toc92665158)

[Is iSCSI a SAN or NAS? 17](#_Toc92665159)

[iSCSI LUNs and Targets 17](#_Toc92665160)

[SAN: 19](#_Toc92665161)

[Types of SAN 19](#_Toc92665162)

[SAN vs. NAS 20](#_Toc92665163)

[SAN Protocols 21](#_Toc92665164)

[NAS Protocols 21](#_Toc92665165)

[What is iSCSI? 21](#_Toc92665166)

[What is Fibre Channel? 22](#_Toc92665167)

[Logical unit number (LUN) 24](#_Toc92665168)

Basic Storage Document

# What is an aggregate?

An aggregate is a collection of disks (or partitions) arranged into one or more RAID groups.  It is the most basic storage object within ONTAP and is required to allow for the provisioning of space for connected hosts.

# How is an aggregate created?

* In 7-Mode, use the aggr create command with the desired parameters.
* In Clustered Data ONTAP, use the storage aggregate create command from the cluster shell with the desired parameters

Note that in order to create an aggregate on a given node, all of the required disks must be owned by the respective node.

The following commands are available in the aggr suite:

add mirror restrict undestroy

copy offline scrub verify

create online show\_space

destroy options split

media\_scrub rename status

## [HA CONSIDERATIONS](https://library.netapp.com/ecmdocs/ECMP1511537/html/man1/na_aggr.1.html" \l "toc5)

Aggregates on different nodes in an HA pair can have the same name. For example, both nodes in an HA pair can have an aggregate named aggr0.

However, having unique aggregate names in an HA pair makes it easier to migrate aggregates between the nodes in the HA pair.

## EXAMPLES

**aggr** **create** **aggr1** **-r** **10** **20**

Creates an aggregate named aggr1 with 20 disks. The RAID groups in this aggregate can contain up to 10 disks, so this new aggregate has two RAID groups. The node adds the current spare disks to the new aggregate, starting with the smallest disk.

**aggr** **create** **aggr1** **20@9**

Creates an aggregate named aggr1 with 20 9-GB disks. Because no RAID group size is specified, the default size (8 disks) is used. The newlycreated aggregate contains two RAID groups with 8 disks and a third group with four disks.

**aggr** **create** **aggr1** **-d** **8a.1** **8a.2** **8a.3**

Creates an aggregate named aggr1 with the specified three disks.

**aggr** **create** **aggr1** **10**  
**aggr** **options** **aggr1** **raidsize** **5**

The first command creates an aggregate named aggr1 with 10 disks which belong to one RAID group. The second command specifies that if any disks are subsequently added to this aggregate, they will not cause any current RAID group to have more than five disks. Each existing RAID group will continue to have 10 disks and no more disks will be added to that RAID group. When new RAID groups are created, they will have a maximum size of five disks.

**aggr** **show\_space** **-h** **ag1**

Displays the space usage of the aggregate `ag1' and scales the unit of space according to the size.

Aggregate 'ag1'

Total space WAFL reserve Snap reserve Usable space BSR NVLOG A-SIS

66GB 6797MB 611MB 59GB 65KB 8192

Space allocated to volumes in the aggregate

Volume Allocated Used Guarantee

vol1 14GB 11GB volume

vol2 8861MB 8871MB file

vol3 6161MB 6169MB none

vol4 26GB 25GB volume

vol1\_clone 1028MB 1028MB (offline)

Aggregate Allocated Used Avail

Total space 55GB 51GB 3494MB

Snap reserve 611MB 21MB 590MB

WAFL reserve 6797MB 5480KB 6792MB

**aggr** **status** **aggr1** **-r**

Displays the RAID information about aggregate aggr1. In the following example, we see that aggr1 is a RAID-DP aggregate protected by block checksums. It is online, and all disks are operating normally. The aggregate contains four disks -- two data disks, one parity disk, and one double-parity disk. Two disks are located on adapter 0b, and two on adapter 1b. The disk shelf and bay numbers for each disk are indicated. All four disks are 10,000 RPM Fibre Channel disks attached via disk channel A. The disk "Pool" attribute is displayed only if SyncMirror is licensed, which is not the case here (if SyncMirror was licensed, Pool would be either 0 or 1). The amount of disk space that is used by Data ONTAP ("Used") and is available on the disk ("Phys") is displayed in the rightmost columns.

Aggr aggr1 (online, raid\_dp) (block checksums)

Plex /aggr1/plex0 (online, normal, active)

RAID group /aggr1/plex0/rg0 (normal)

RAID Disk Device HA SHELF BAY CHAN Pool Type RPM Used (MB/blks) Phys (MB/blks)

--------- ------ ------------- ---- ---- ---- ----- -------------- --------------

dparity 0b.16 0b 1 0 FC:A - FCAL 10000 136000/278528000 137104/280790184

parity 1b.96 1b 6 0 FC:A - FCAL 10000 136000/278528000 139072/284820800

data 0b.17 0b 1 1 FC:A - FCAL 10000 136000/278528000 139072/284820800

data 1b.97 1b 6 1 FC:A - FCAL 10000 136000/278528000 139072/284820800

**aggr** **status** **aggr1** **-R**

Displays the disk information about aggregate **aggr1** as well as the corresponding VBN start and end addresses. In the following example, we see that aggr1 is an aggregate with 2 RAID groups. The first RAID group has 3 data disks and the second 1 data disk. By the VBN addresses given, we see that the data on the disks is laid out in the following order: 2c.00.3, 2c.00.4, 2c.00.6, 2c.00.9.

Aggr aggr1 (online, raid\_dp) (block checksums)

Plex /aggr1/plex0 (online, normal, active)

RAID group /aggr1/plex0/rg0 (normal)

RAID Disk Device Model Number Serial Number VBN Start VBN End

--------- ------ ------------ ------------- --------- -------

dparity 2c.00.0 X410\_HVIPC288A15 JTXT4HPJ - -

parity 2c.00.2 X410\_HVIPC288A15 JTXSZGWJ - -

data 2c.00.3 X410\_HVIPC288A15 JTXTEN2J 0 69626751

data 2c.00.4 X410\_HVIPC288A15 JTXTER3J 69626752 139253503

data 2c.00.6 X410\_HVIPC288A15 JTXSXXBJ 139253504 208880255

RAID group /aggr1/plex0/rg1 (normal)

RAID Disk Device Model Number Serial Number VBN Start VBN End

--------- ------ ------------ ------------- --------- -------

dparity 2c.00.7 X410\_HVIPC288A15 JTXTHW8J - -

parity 2c.00.8 X410\_HVIPC288A15 JTXTH4AJ - -

data 2c.00.9 X410\_HVIPC288A15 JTXTH6JJ 208880256 278507007

##### How do you view aggregate details?

There are several commands available to to see aggregate details.  The specific commands are determined by the type of information required.  The most common commands are as shown.

* To determine the physical size of an aggregate, use df -A <aggregate name>  to see the space usage for the aggregate in question.
* To see the physical disks that are in a given aggregate, the aggr status -r <aggregate name> command can be run with an output similar to :

###### 7-Mode:

**Example:**

controller> aggr status -r aggr0  
Aggregate aggr0 (online, raid\_dp) (block checksums)  
  Plex /aggr0/plex0 (online, normal, active)  
    RAID group /aggr0/plex0/rg0 (normal)  
  
      RAID Disk Device          HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Phys (MB/blks)  
      --------- ------          ------------- ---- ---- ---- ----- --------------    --------------  
      dparity   0a.16           0a    1   0   FC:A   -  FCAL 15000 136000/278528000  137104/280790184  
      parity    0a.17           0a    1   1   FC:A   -  FCAL 15000 136000/278528000  137104/280790184  
      data      0c.00.1         0c    0   1   SA:B   -  SAS  15000 136000/278528000  280104/573653840

To see specific aggregate details, use the following commands:

**7-Mode:**  
controller> aggr status -v <aggr name>

# Understanding aggregate and lun:

An aggregate is the physical storage. It is made up of one or more raid groups of disks.

A LUN is a logical representation of storage. It looks like a hard disk to the client. It looks like a file inside of a volume.

Raid groups are protected sets of disks.  Consisting of 1 or 2 parity, and 1 or more data disks.  We don't build raid groups, they are built behind the scene when you build an aggregate.  For example:

in a default configuration you are configured for RaidDP and a 16 disk raid group (assuming FC/SAS disks).  So, if i create a 16 disk aggregate i get 1 raid group.  if I create a 32 disk aggregate, i get 2 raid groups.  Raid groups can be adjusted in size.  For FC/SAS they can be anywhere from 3 to 28 disks, with 16 being the default.  You may be tempted to change the size so I have a quick/dirty summary of reasons.

Aggregates are collections of raid groups.  Consist of one or more Raid Groups.

* I like to think of aggregates as a big hard drive.  There are a lot of similarities in this.  When you buy a hard drive you need partition it and format it before it can be used.  Until then it’s basically raw space.  Well, that’s an aggregate.  It’s just raw space.
* A volume is analogous to a partition.  It’s where you can put data.  Think of the previous analogy.  An aggregate is the raw space (hard drive), the volume is the partition, it’s where you put the file system and data.  Some other similarities include the ability to have multiple volumes per aggregate, just like you can have multiple partitions per hard drive.  And you can grow and shrink volumes, just like you can grow and shrink partitions.
* A qtree is analogous to a subdirectory.  let’s continue the analogy.  Aggregate is hard drive, volume is partition, and qtree is subdirectory.
* Why use them? To sort data.  The same reason you use them on your personal PC.  There are 5 things you can do with a qtree you can't do with a directory and that’s why they aren't just called directories.
* Last but not least, LUNs.  It’s a logical representation of space, off your SAN.  But the normal question is when I use a LUN over a CIFS or NFS share/export.  I normally say it depends on the Application.
* Some applications need local storage, they just can't seem to write data into a NAS (think CIFS or NFS) share.  Microsoft Exchange and SQL are this way.  They require local hard drives.  So the question is, how do we take this network storage and make it look like an internal hard drive.  The answer is a LUN.  it takes a bit of logical space out of the aggregate (actually just a big file sitting in a volume or qtree) and it gets mounted up on the windows box, looking like an internal hard drive.  The file system makes normal SCSI commands against it.  The SCSI commands get encapsulated in FCP or iSCSI and are sent across the network to the SAN hardware where its converted back into SCSI commands then reinterpreted as WAFL read/writes.
* Some applications know how to use a NAS for storage (think Oracle over NFS, or ESX with NFS data stores) and they don't need LUNs.  They just need access to shared space and they can store their data in it.

# **What is the use of aggregate?**

Aggregates are the raw space in your storage system.  You take a bunch of individual disks and aggregate them together into aggregates.  But, an aggregate can't actually hold data, it’s just raw space.  You then layer on partitions, which in NetApp land are called volumes.  The volumes hold the data.

You make aggregates for various reasons.  For example:

* **Performance boundaries** - a disk can only be in one aggregate.  so each aggregate has its own discreet drives.  This lets us tune the performance of the aggregate by adding in however many spindles we need to achieve the type of performance we want.  This is kind of skewed by having Flash Cache cards and such, but it’s still roughly correct.

* **Shared Space boundary** - All volumes in an aggregate share the hard drives in that aggregate.  There is no way to prevent the volumes in an aggregate from mixing their data on the same drives.  I ran into a problem at one customer that, due to regulatory concerns, couldn't have data type A mixed with data type B.  The only way to achieve this is to have two aggregates.
* For volumes - you can't have a flexible volume without an aggregate.  Flex Vols are logical, Aggregates are physical.  You layer one or more flex vols on top (in side) of an aggregate.

# An aggregate is made of Raid Groups.

Let’s do a few examples using the command to make an aggregate  "aggr create aggr1 16".  if the default raid group size is 16, then the aggregate will have one raid group.  But, if i use the command "aggr create aggr1 32" now I have two full raid groups, but still only one aggregate.  So, the aggregate gets the performance benefit of 2 RGs worth of disks.  Notice we did NOT build a raid group.  Data ONTAP built the RG based on the default RG size.

# **what happen when we give a volume directly to the user instead of LUN?**

# **What was the difference between these two?**

Volumes are access via NAS protocols, CIFS/NFS

LUNS are accessed via SAN protocols, iSCSI/FCP/FCoE

In the end you can put data in a LUN, you can put data in a Volume.  It’s how you get there that’s the question.

# 

# Is it possible to give a volume directly in a san environment?

No

# **Aggregate raw space and volume-providing file system, then what about Lun?**

# **By this hierarchy what kind of function lun would provide?**

Luns are logical.  They go inside a volume, or in a qtree.

From a netapp perspective they are really just one big file sitting inside of a volume or qtree.

From a host perspective, they are like a volume, but use a different protocol to access them (purists will argue with that but I’m simplifying).  LUNs provide a file system, like Volumes provide a file system, the major difference is who controls the files system.

With a LUN the storage system can't see the file system, all it sees is one big file.  The host mounts the file system via one of the previously mentioned protocols and lays a file system down inside.  **The host then controls that file system**.

I normally determine LUN/Volume usage by looking at the Application.  Some apps won't work across a network, Microsoft SQL and Exchange are two examples of this.  They require local disks.  LUNs look like local disks.  Some applications work just fine across the network, using NFS, like Oracle.  In the end it’s normally the application that will determine whether you get your filesystem access through a LUN or a Volume.

Some things like Oracle or VMware can use either LUNs or NFS volumes, so with them its whatever you find easier or cheaper.  I'm an NFS bigot so I tend to push that but in the end either works.

# From netapp perspective, volume will provide filesystem.

# Here the file system you are talking about is host side filesystem or WAFL ?

The underlying filesystem is always WAFL in the volume.

When you share out a volume it looks like NTFS to a windows box, or it looks like a UNIX filesystem to a Unix box **but in the end its just WAFL in the volume**.

With a LUN it’s a bit different.

You first make a volume, then you put a LUN in the volume.  The volume has WAFL as the file system, the LUN looks like one big file in the volume.

You then connect to the storage system using a SAN protocol.  The big file we call a LUN is attached to the host via the SAN protocol and looks like a big hard drive.  The host then formats the hard drive with NTFS or whatever File system the unix box is using.  That file system is actually NTFS, or whatever.  It’s inside the LUN, which is big file inside of a Volume, which has WAFL as its file system.

# [What's the difference between a LUN and a Volume?](https://serverfault.com/questions/395517/whats-the-difference-between-a-lun-and-a-volume)

A LUN is a logical volume from the point of view of the storage. From the client point of view the LUN it is a disc volume that can be partitioned.

Volume is a generic term. It means a contiguous storage area. This means that you might need to partition it and that you might also need to create a filesystem. Some programs can work directly with a volume without having a partition or a filesystem. So by volume you can consider a LUN, a partition or even a file (loop-back mounted volume, DB volume), depending on the context.

C: and D: is, usually, a mounted disk partition. This means that the kernel expose to the programs the volume as a filesystem.

Oh, and you can mount the same filesystem in 2 places at once at the same time. E.g. having C: D:\mountpoint\ pointing to the same partition.

A LUN has a world-wide name, a Volume does not. Much the same thing otherwise.

# What a FlexVol volume is

A FlexVol volume is a data container associated with a Storage Virtual Machine (SVM) with FlexVol volumes. It gets its storage from a single associated aggregate, which it might share with other FlexVol volumes or Infinite Volumes. It can be used to contain files in a NAS environment, or LUNs in a SAN environment.

# What is a qtree?

A qtree is a logically defined file system that can exist as a special subdirectory of the root directory within an internal volume. You can create up to 4,995 qtrees per internal volume. There is no maximum for the storage system as a whole.

In general, qtrees are similar to internal volumes. However, they have the following key differences:

* Snapshot copies can be enabled or disabled for individual internal volumes but not for individual qtrees.
* Qtrees do not support space reservations or space guarantees.
* There are no restrictions on how much disk space can be used by the qtree or how many files can exist in the qtree.

# Is iSCSI a SAN or NAS?

iSCSI is a transport layer protocol that describes how Small Computer System Interface (SCSI) packets are transferred over a network. It works by transporting data between a server and a storage device.

The difference between iSCSI and NAS is that iSCSI is a data transport protocol where NAS is a [common way of connecting storage](https://www.promax.com/blog/deciding-on-san-vs-nas-for-your-data-storage) into a shared user network.

iSCSI is popular in the implementation of SAN systems because of their block level storage structure.

When data arrives at its destination the iSCSI protocol separates the SCSI commands so that the Operating System will see the storage as a local device and allow formatting as usual.

Many people use iSCSI because it uses Ethernet and requires less expensive and complex equipment to operate unlike Fibre Channel.

# iSCSI LUNs and Targets

Sharing block devices via iSCSI is a common way to make network-attached storage available. An iSCSI LUN is a logical unit of storage. In SoftNAS Cloud®, the basic storage LUN is a volume that is accessed as a block device. The block device volumes have a mount point in the Linux /dev/zvol filesystem because they are disk block devices.

For example, a storage pool naspool1 with volume name lun01 would be named /dev/zvol/naspool1/lun01 as its mount point. These device references are links to Linux block devices used to access the volume's raw data blocks via iSCSI.

iSCSI targets are used by iSCSI initiators to establish a network connection. The target serves up the LUNs, which are collections of disk blocks accessed via the iSCSI protocol over the network. A target can offer one or more LUNs to the iSCSI clients, who initiate a connection with the iSCSI server.

For example, VMware vSphere or Windows connects to the iSCSI server and retrieves a list of available targets. Then, for each target, the list of its published LUNs are available for use.

Note the default is Thick Provision, which reserves space for the LUN at time of creation. Alternatively, choose Thin Provision, which will create a "sparse" LUN that only consumes space as it is actually used.

# SAN:

A SAN is block-based storage, leveraging a high-speed architecture that connects servers to their logical disk units (LUNs). A LUN is a range of blocks provisioned from a pool of shared storage and presented to the server as a logical disk. The server partitions and formats those blocks—typically with a file system—so that it can store data on the LUN just as it would on local disk storage.

SANs make up about two-thirds of the total networked storage market. They are designed to remove single points of failure, making SANs highly available and resilient. A well-designed SAN can easily withstand multiple component or device failures.

# Types of SAN

The most common SAN protocols are:

* **Fibre Channel Protocol (FCP).** The most widely used SAN or block protocol, deployed in 70% to 80% of the total SAN market. FCP uses Fibre Channel transport protocols with embedded SCSI commands.
* **Internet Small Computer System Interface (iSCSI).** The next largest SAN or block protocol, with approximately 10% to 15% of the market. iSCSI encapsulates SCSI commands inside an Ethernet frame and then uses an IP Ethernet network for transport.
* **Fibre Channel over Ethernet (FCoE).** FCoE is less than 5% of the SAN market place. It is similar to iSCSI, since it encapsulates an FC frame inside an Ethernet datagram. Then like iSCSI, it uses an IP Ethernet network for transport.
* **Non-Volatile Memory Express over Fibre Channel (FC-NVMe).** [NVMe](https://www.netapp.com/data-storage/nvme" \o "SEO: nvme) is an interface protocol for accessing flash storage via a PCI Express (PCIe) bus. Unlike traditional all-flash architectures, which are limited to a single, serial command queue, NVMe supports tens of thousands of parallel queues, each with the ability to support tens of thousands of concurrent commands.

# SAN vs. NAS

Both SAN and network-attached storage (NAS) are methods of managing storage centrally and sharing that storage with multiple hosts (servers). However, NAS is Ethernet-based, while SAN can use Ethernet and Fibre Channel. In addition, while SAN focuses on high performance and low latency, NAS focuses on ease of use, manageability, scalability, and lower total cost of ownership (TCO). Unlike SAN, NAS storage controllers partition the storage and then own the file system. Effectively this makes a NAS server look like a Windows or UNIX/Linux server to the server consuming the storage.

# SAN Protocols

* Fibre Channel Protocol (FCP)
* Internet Small Computer System Interface (iSCSI)
* Fibre Channel over Ethernet (FCoE)
* Non-Volatile Memory Express over Fibre Channel (FC-NVMe)

# NAS Protocols

* **Common Internet File Services / Server Message Block (CIFS/SMB).** This is the protocol that Windows usually uses.
* **Network File System (NFS).** NFS was first developed for use with UNIX servers and is also a common Linux protocol.

# What is iSCSI?

The SNIA dictionary defines Internet Small Computer Systems Interface ([iSCSI](https://www.snia.org/education/online-dictionary/term/internet-small-computer-systems-interface)) as a transport [protocol](https://www.snia.org/education/online-dictionary/term/protocol) that provides for the [SCSI](https://www.snia.org/education/online-dictionary/term/scsi) protocol to be carried over a [TCP](https://www.snia.org/education/online-dictionary/term/tcp)-based [IP](https://www.snia.org/education/online-dictionary/term/ip) network, standardized by the [Internet Engineering Task Force](https://www.snia.org/education/online-dictionary/term/internet-engineering-task-force) and described in [RFC](https://www.snia.org/education/online-dictionary/term/rfc) 3720.

iSCSI is a block protocol for storage networking and runs the very common SCSI storage protocol across a network connection which is usually [Ethernet](https://www.snia.org/education/online-dictionary/term/ethernet).

iSCSI, like [Fibre Channel](https://www.snia.org/education/what-is-fibre-channel), can be used to create a [Storage Area Network](https://www.snia.org/education/storage_networking_primer/san/what_san) (SAN). iSCSI traffic can be run over a shared network or a dedicated storage network. However, iSCSI does not support file access [Network Attached Storage](https://www.snia.org/education/what-is-nas) (NAS) or [object storage](https://www.snia.org/education/online-dictionary/term/object-storage) access (they use different transport protocols).

There are multiple transports that can be used for iSCSI. The most common is [TCP/IP](https://www.snia.org/education/online-dictionary/term/tcp%252fip) over Ethernet, but Remote Direct Memory Access (RDMA) can also be used with iSER, which is iSCSI Extensions for RDMA. If using iSER, the transport is RoCE or [InfiniBand](https://www.snia.org/education/online-dictionary/term/infiniband) and the underlying network is Ethernet (for RoCE) or InfiniBand (for InfiniBand transport).

iSCSI offers good block storage performance along with low cost. It is also widely supported by all major operating systems and hypervisors and can run on standard network cards or specialized [Host Bus Adapters](https://www.snia.org/education/online-dictionary/term/host-bus-adapter) (HBAs). It is also supported by almost all enterprise storage arrays. For these reasons it has been popular for so-called “Tier 2” applications that require good, but not the best, block storage performance, and for storage that is shared by many hosts. It also is very popular among hyperscalers and large cloud service providers when they need a block storage solution that runs over Ethernet.

# What is Fibre Channel?

[Fibre Channel](https://www.snia.org/education/online-dictionary/term/fibre-channel) is a high-speed data transfer [protocol](https://www.snia.org/education/online-dictionary/term/protocol) that provides in-order, lossless delivery of raw block data. It is designed to connect general purpose computers, mainframes and supercomputers to storage devices. The technology primarily supports point-to-point (two devices directly connected to each other) though most common found in switched fabric (devices connected by [Fibre Channel switche](https://www.snia.org/education/online-dictionary/term/fibre-channel-switched)s) environments.

A [storage area network (SAN)](https://www.snia.org/education/storage_networking_primer/san/what_san) is a dedicated network used for storage connectivity between host servers and shared storage - typically shared [arrays](https://www.snia.org/education/online-dictionary/term/storage-array) that deliver [block-level](https://www.snia.org/education/online-dictionary/term/block) data storage.

Fibre Channel SANs are typically deployed for low latency applications best suited to block-based storage, such as databases used for high-speed online transactional processing (OLTP), such as those found in banking, online ticketing, and virtual environments. Fibre Channel typically runs on optical fiber cables within and between data centers but can also run on copper cabling.

Fibre Channel is a high-speed data transfer protocol that provides in-order, lossless delivery of raw block data to connect data storage to host servers. Fibre Channel fabrics can be extended over distance for [Disaster Recovery](https://www.snia.org/education/online-dictionary/term/disaster-recovery)and [Business Continuance](https://www.snia.org/education/online-dictionary/term/business-continuity) and most SANs are typically designed with redundant fabrics.

Begun in 1988, Fibre Channel is standardized in the T11 Technical Committee of the International Committee for [Information Technology Standards](https://www.snia.org/education/online-dictionary/term/incits) (INCITS), an [American National Standards Institute](https://www.snia.org/education/online-dictionary/term/american-national-standards-institute) (ANSI)-accredited standards committee.  The Fibre Channel Physical and Signaling Interface (FC-PH) was first published in 1994.

# Logical unit number (LUN)

A logical unit number (LUN) is a unique identifier for designating an individual or collection of physical or virtual storage devices that execute input/output (I/O) commands with a host computer, as defined by the Small System Computer Interface (SCSI) standard.

SCSI is a widely implemented I/O interconnect that can facilitate data exchange between servers and storage devices through transport protocols. Examples of transport protocols include Internet SCSI and Fibre Channel. An SCSI initiator in the host originates the I/O command sequence, which is then transmitted to a target endpoint or recipient storage device. A logical unit is an entity within the SCSI target that responds to the SCSI I/O command.

LUNs are used to identify subsets of data in a disk so that the computing devices using them can execute operations.