

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology in Information Technology

By

K SANKARI
(REG. NO. 39120102)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFAIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **K SANKARI (Reg. No: 39120102)** who carried out the project entitled "ISCSI" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. J. Jabez M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **K SANKARI** hereby declare that the project report entitled “**ISCSI**” done by me under the guidance of **Dr. J. JEBAZ** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information and Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean, School of Computing, Dr. R. Subhashini, M.E., Ph.D., Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J. Jebaz M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



Red Hat

Certificate of Participation

Sankari K

has attended and successfully completed

Red Hat Certified System Administrator, ON-LINE course

at Sathyabama Institute of Science and Technology

held from 1st Sep 2021 to 22nd Sep 2021




Authorised Signatory

S.No : 2021/ADV/11980

Date : 27th Oct 2021

ABSTRACT

iSCSI is a TCP/IP based protocol for sending SCSI commands over IP based networks. It allows you to connect block-based storage devices that act like normal hard drives or solid-state drives from a server to clients over the network. As a Storage Area Network (SAN) protocol, **iSCSI** helps you consolidate your storage devices in your local or remote data centers.

Because performance can degrade due to bandwidth congestion on shared networks, you should implement **iSCSI** on cabling that is independent of standard LAN traffic. Typically, **iSCSI** is paired with dedicated 10 Gigabit Ethernet or better networking to maximize performance.

SAN traffic is typically unencrypted to maximize performance. The cabling from the physical server to the storage is normally enclosed in the data center, and is ideally not directly connected to LAN networks. For WAN security, **iSCSI** administrators may use IPsec, a protocol suite for securing IP network traffic.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	i
	LIST OF FIGURES	ii
1	INTRODUCTION	9
	1.1 GENERAL	9
	1.2 MODULES	9
	1.3HARDWARE AND SOFTWARE REQUIREMENTS	10
2	AIM AND SCOPE	11
	2.1 AIM	11
	2.2 PROBLEM STATEMENT	11
	2.3 SCOPE	11
3	ALGORITHMS AND METHODS	12
	3.1 INTERNETWORKING BASIC	12
	3.2 TCP/IP	16
	3.3 ALGORITHM	28
	3.4 SUMMARY	29
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	30
	4.1 RESULT AND DISCUSSION	30
	4.2 PERFORMANCE	31
5	SUMMARY AND CONCLUSIONS	31
	5.1 REFERENCES	31

5.2 APPENDIX	32
A SCREENSHOTS	32
B SOURCE CODE	37

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.1	OSI REFERENCE MODEL LAYER	15
1.2	TCP/IP PROTOCOL SUITE	19
1.3	LAYER FUNCTIONS	20
1.4	TRANSPORT PROTOCOL LAYERS	21
1.5	TCP SEGMENT FORMAT	23
1.6	USER DATAGRAM PROTOCOL	24
1.7	UDP SEGMENT FORMAT	24

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Linux is a Unix-like, open source and community-developed operating system (OS) for computers, servers, mainframes, mobile devices and embedded devices. It is supported on almost every major computer platform, including x86, ARM and SPARC, making it one of the most widely supported operating systems.

LVM is a tool for logical volume management which includes allocating disks, striping, mirroring and resizing logical volumes. As the name implies, it can sort raw storage into logical volumes, making it easy to configure and use.

iSCSI is a TCP/IP based protocol for sending SCSI commands over IP based networks. It allows you to connect block-based storage devices that act like normal hard drives or solid-state drives from a server to clients over the network. As a Storage Area Network (SAN) protocol, **iSCSI** helps you consolidate your storage devices in your local or remote data centers.

1.2 MODULES

- Introduction to LVM
- Configuring and implementing LVM
- Installation and configuring Physical Volume
- Introduction to iSCSI
- Configuring and implementing iSCSI
- Backup from Linux to Linux

1.3 HARDWARE AND SOFTWARE REQUIREMENTS

1.3.1 Hardware Requirements

1. Intel machines with 128MB RAM/ 8GB HDD

1.3.2 Software Requirements

1. Redhat Enterprise Linux 8.0
2. LVM
3. ISCSI packages

CHAPTER 2

AIM AND SCOPE

2.1 AIM

The main objective of this project is to make a LVM to share the disk through ISCSI.

2.2 PROBLEM STATEMENT

The job is to create a Logical Volume Manager (LVM) in a storage area using ISCSI. Then share the LVM disk to your client machine. In that shared disk, we need to configure a partition

2.3 SCOPE

LVM makes easier to manage disk space. If a file system needs more space, it can be added to its logical volumes from the free spaces in its volume group and the file system can be re-sized as we wish. ISCSI is good alternative to fibre channel-based SANs. We can easily manage, mount and format ISCSI Volume under Linux.

CHAPTER 3

EXPERIMENTAL OR MATERIALS, METHODS AND ALGORITHMS USED

3.1 INTERNETWORKING BASIC

OSI REFERENCE MODEL: The OSI Model (Open Systems Interconnection Model) is a conceptual framework used to describe the functions of a networking system. The OSI model characterizes computing functions into a universal set of rules and requirements in order to support interoperability between different products and software.

THE LAYERED APPROACH: This approach breaks up the operating system into different layers. This allows implementers to change the inner workings, and increases modularity. As long as the external interface of the routines don't change, developers have more freedom to change the inner workings of the routines.

THE ADVANTAGES OF THE OSI MODEL ARE:

- It is a generic model and acts as a guidance tool to develop any network model.
- It is a layered model. Changes in one layer do not affect other layers, provided that the interfaces between the layers do not change drastically.
- It distinctly separates services, interfaces, and protocols. Hence, it is flexible in nature. Protocols in each layer can be replaced very conveniently depending upon the nature of the network.
- It supports both connection-oriented services and connectionless services.

THE OSI REFERENCE MODEL LAYERS

7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

1.1 OSI REFERENCE MODEL

THE 7 LAYERS ARE:

APPLICATION LAYER: An application layer is an abstraction layer that specifies the shared communications protocols and interface methods used by hosts in a communications network. An application layer abstraction is specified in both the Internet Protocol Suite (TCP/IP) and the OSI model.

PRESENTATION LAYER: Presentation Layer is the 6th layer in the Open System Interconnection (OSI) model. This layer is also known as Translation layer, as this layer serves as a data translator for the network. ... The main responsibility of this layer is to provide or define the data format and encryption.

SESSION LAYER: The Session Layer is the 5th layer in the Open System Interconnection (OSI) model. This layer allows users on different machines to establish active communications sessions between them. It is responsible for establishing, maintaining, synchronizing, terminating sessions between end-user applications.

TRANSPORT LAYER: In Computer Networking, the transport layer is a conceptual division of methods in the layered architecture of protocols in the network stack in the Internet Protocol Suite and the OSI model. The protocols of this layer provide host-to-host communication services for applications. It provides services such as connection-oriented communication, reliability, flow control, and multiplexing.

NETWORK LAYER: The network layer is the third level (Layer 3) of the Open Systems Interconnection Model (OSI Model) and the layer that provides data routing paths for network communication. Logical connection setup, data forwarding, routing and delivery error reporting are the network layer's primary responsibilities.

DATA LINK LAYER: The data link layer, or layer 2, is the second layer of the seven-layer OSI model of computer networking. This layer is the protocol layer that transfers data between nodes on a network segment across the physical layer.

PHYSICAL LAYER: In the seven-layer OSI model of computer networking, the physical layer or layer 1 is the first and lowest layer; The layer most closely associated with the physical connection between devices. This layer may be implemented by a PHY chip.

ETHERNET NETWORKING

Ethernet is a family of wired computer networking technologies commonly used in local area networks (LAN), metropolitan area networks (MAN) and wide area networks (WAN). Over time, Ethernet has largely replaced competing wired LAN technologies such as Token Ring, FDDI and ARCNET.

Ethernet at the Data Link Layer:

At the data link layer, Ethernet specifies what the data should look like, including the header and trailer. The protocol is defined by IEEE 802.3 and divides the data link layer into two sublayers: the Logical Link Control (LLC) sublayer and the Media Access Control (MAC) sublayer.

Ethernet Addressing:

In a LAN, each node is assigned a physical address, also known as a MAC/Ethernet address. This address is unique to each of the nodes on the LAN and is 6 bytes (48 bits) long, which is burned on the Ethernet card (also known as the network interface card). Ethernet is a byte-count protocol.

Ethernet Frame:

- An Ethernet frame is a piece of data along with the information that is required to transport and deliver that piece of data. In networking reference models, such as; OSI Seven Layers model and TCP/IP, the Ethernet frame is defined in the Data link layer.
- An Ethernet frame is preceded by a preamble and start frame delimiter (SFD), which are both part of the Ethernet packet at the physical layer. Each Ethernet frame starts with an Ethernet header, which contains destination and source MAC addresses as its first two fields. The middle section of the frame is payload data including any headers for other protocols carried in the frame.

The frame ends with a frame check sequence (FCS), which is a 32-bit cyclic redundancy check used to detect any in-transit corruption of data.

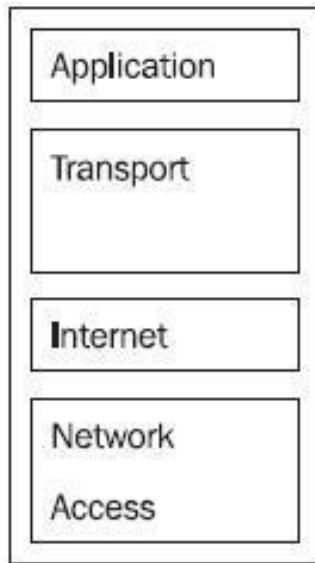
DATAENCAPSULATION:

Data encapsulation, also known as data hiding, is the mechanism whereby the implementation details of a class are kept hidden from the user. The user can only perform a restricted set of operations on the hidden members of the class by executing special functions commonly called methods.

3.2 TCP/IP

TCP/IP PROTOCOL SUITE:

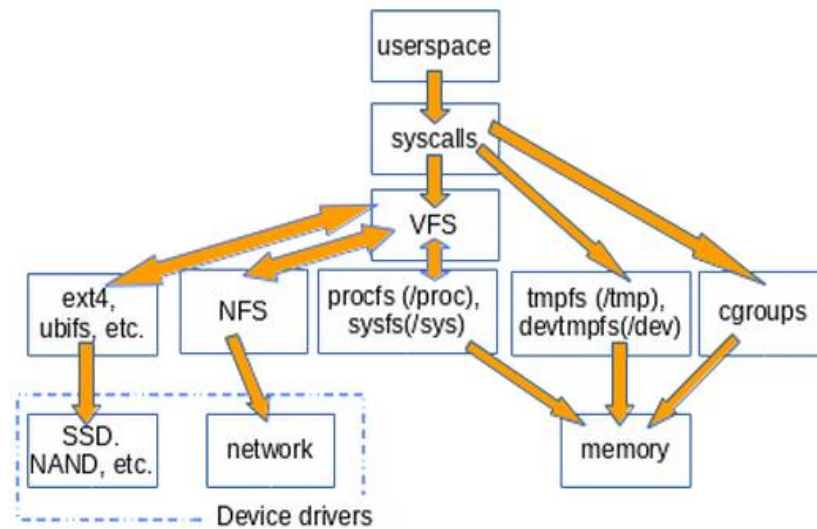
- TCP/IP (Transmission Control Protocol/Internet Protocol) is the most widespread networking protocol in use today, forming the base of many networks, including the Internet.
- TCP/IP is actually not just one protocol, but is a protocol suite.
- It is a group of protocols that work together and allow dissimilar systems to talk to each other through a common set of rules.



1.2 TCP/IP PROTOCOL

LAYER FUNCTIONS:

- To put the kernel in context, you can think of a Linux machine as having 3 layers: The hardware: **The physical machine**—the bottom or base of the system, made up of memory (RAM) and the processor or central processing unit (CPU), as well as input/output (I/O) devices such as storage, networking, and graphics.



1.3 LAYER FUNCTIONS

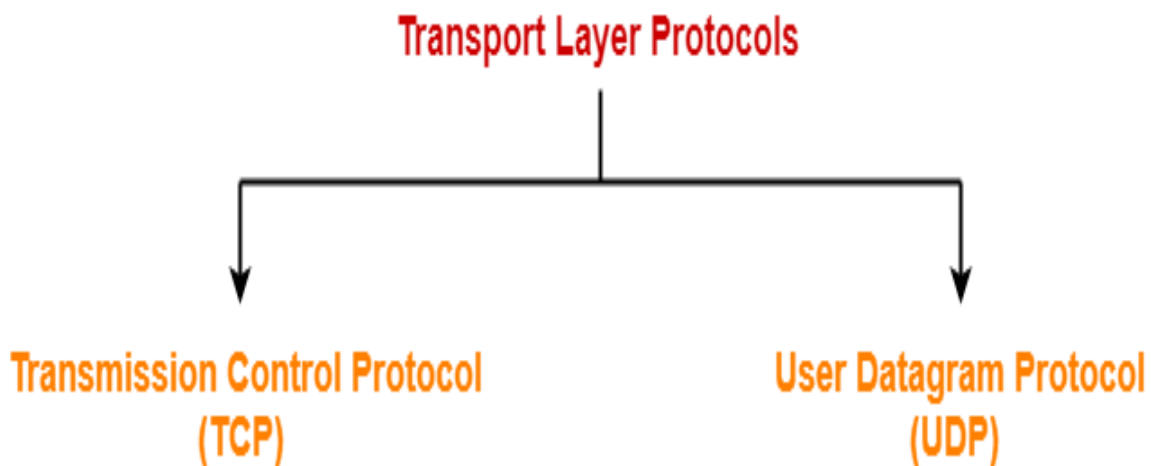
THE APPLICATION LAYER:

- The application layer is the top layer of both the seven-layer OSI model and the five-layer TCP/IP model.
- It provides services to connect application programs to communications protocols and to ensure effective communication with other application programs over a network.

These services include confirming that necessary communication resources exist, ensuring identification of the destination host, authenticating either or both the message sender and recipient, determining protocol rules at the application level and, ensuring agreement at both ends about error recovery procedures, data integrity and privacy.

THE TRANSPORT LAYER:

- The transport layer in TCP/IP consists of two major protocols. It contains a connection-oriented service reliable service otherwise known as a streaming service provided by the TCP protocol.
- In addition, TCP/IP includes an individual packet transmission service known as an *unreliable* or *datagram service*, which is provided by UDP.
- The function of the data link layer to provide an end-to-end connection-oriented service that includes error checking and recovery, one wonders why there is a need for a reliable delivery method at the transport layer.
- The reason why TCP/IP evolved this way is that it was designed to be carried over Ethernet and other LAN protocols, which don't provide a connection-oriented service at the link layer.



1.4 TRANSPORTATION LAYER PROTOCOLS

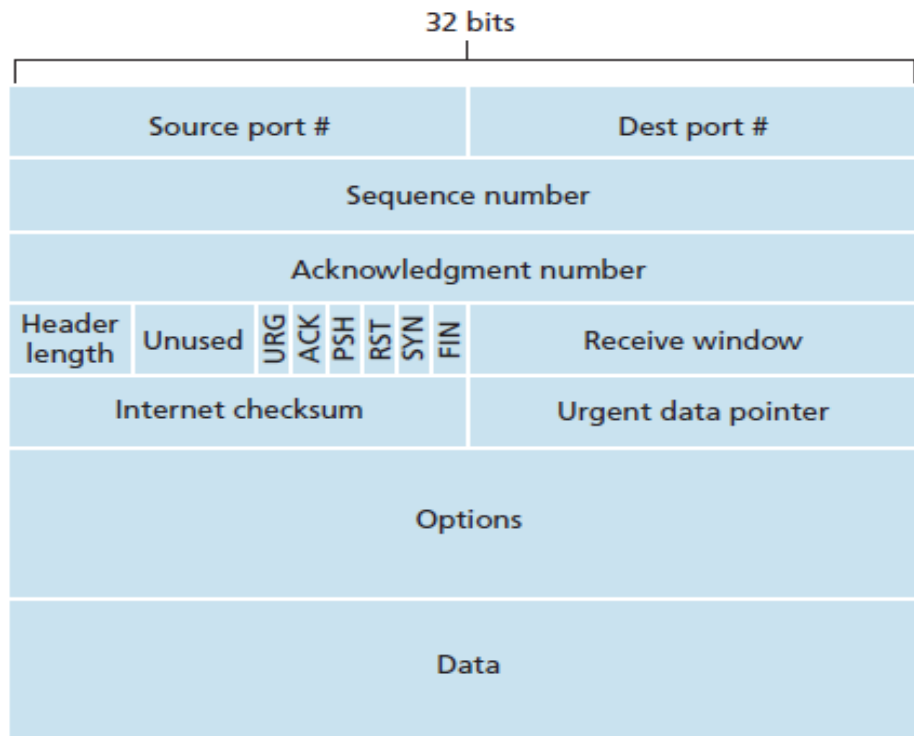
TRANSMISSION CONTROL PROTOCOL:

- The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite.

- It originated in the initial network implementation in which it complemented the Internet Protocol (IP).
- Therefore, the entire suite is commonly referred to as TCP/IP.
- TCP provides reliable, ordered, and error-checked delivery of a stream of octets (bytes) between applications running on hosts communicating via an IP network.

TCP SEGMENT FORMAT:

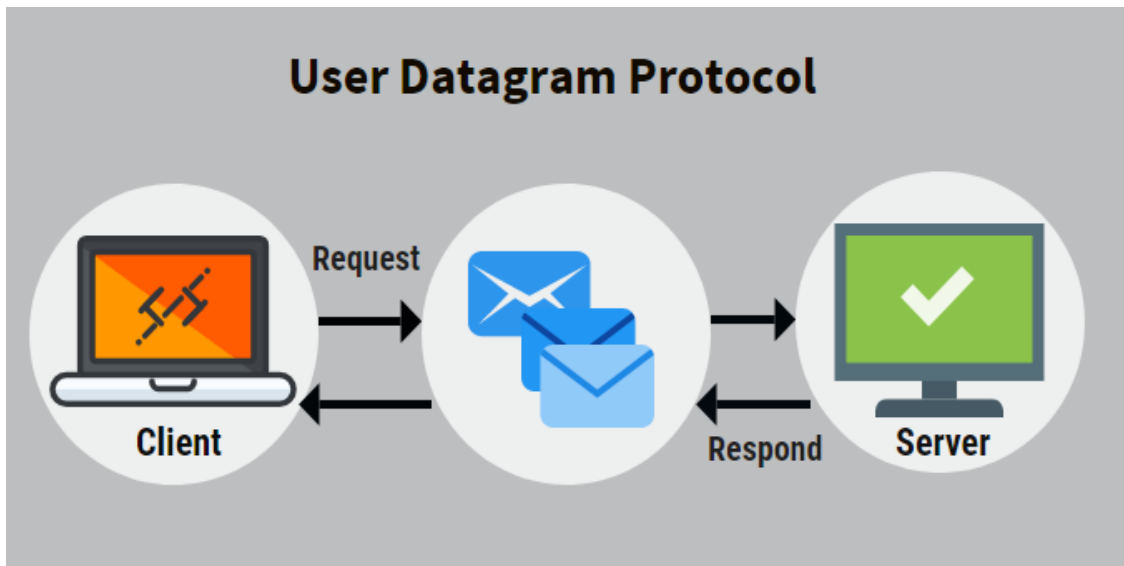
- The TCP segment consists of header fields and a data field. The data field contains a chunk of application data. The MSS (Maximum Segment Size) limits the maximum size of a segment's data field.
- When TCP sends a large file, such as an image as part of a web page, it typically breaks the file into chunks of size MSS (except for the last chunk, which will often be less than the MSS).



1.5 TCP SEGMENT FORMAT

USER DATAGRAM PROTOCOL:

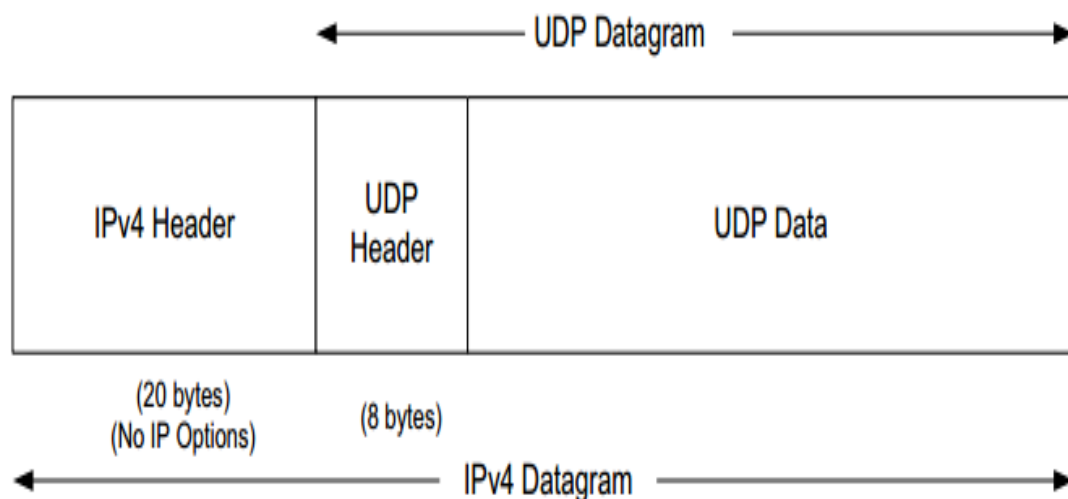
- User datagram protocol (UDP), one of the core protocols of the Internet protocol suite, is a connectionless, transport layer protocol that exchanges packets with minimal error recovery services and without guaranteed delivery or acknowledgment.
- It is a communications protocol that is primarily used to establish low-latency and loss-tolerating connections between applications on the internet.
- UDP speeds up transmissions by enabling the transfer of data before an agreement is provided by the receiving party.



1.6 USER DATAGRAM PROTOCOL 1

UDP SEGMENT FORMAT:

- UDP is short for User Datagram Protocol. It is the simplest transport layer protocol.
- It has been designed to send data packets over the Internet.
- It simply takes the datagram from the network layer, attaches its header and sends it to the user.



1.7 UDP SEGMENT FORMAT 1

PORT NUMBERS:

- A port is an application-specific or process-specific software construct serving as a communications endpoint and it is identified by its number such as TCP port number 80.
- It is used by TCP and UDP of the Internet Protocol Suite.
- A port number is a 16-bit unsigned integer, thus ranging from 0 to 65535.

LAYER FUNCTIONS

THE INTERNET LAYER:

- The Network Access Layer is the lowest layer of the TCP/IP protocol hierarchy.
- The protocols in this layer provide the means for the system to deliver data to the other devices on a directly attached network.
- It defines how to use the network to transmit an IP datagram.

INTERNET PROTOCOL:

- Linux and other Unix operating systems use the TCP/IP protocol. It is not a single network protocol, but a family of network protocols that offer various services.
- IP was developed based on an application used for military purposes and was

defined in its present form in an RFC in 1981. RFC stands for Request for Comments.

- IP (Internet Protocol) is where the insecure data transfer takes place.
- TCP, to a certain extent, is simply the upper layer for the IP platform serving to guarantee secure data transfer.
- The IP layer itself is, in turn, supported by the bottom layer, the hardware-dependent protocol, such as ethernet.

INTERNET CONTROL MESSAGE PROTOCOL:

- The Internet Control Message Protocol (ICMP) is the error-report mechanism for the IP layer, which also resides in the network layer.
- Though ICMP is based on IP, it doesn't make IP more reliable.
- Packets can be lost despite the use of ICMP, and IP or ICMP won't notice that packets are lost.
- Essentially, this is not a protocol for the end user, but a special control protocol that issues error reports and can control the behavior of machines participating in TCP/IP data transfer. In addition, a special echo mode is provided by ICMP that can be viewed using the program ping.

ADDRESS RESOLUTION PROTOCOL:

- The primary function of this protocol is to resolve the IP address of a system to its mac address, and hence it works between level 2(Data link layer) and level

3(Network layer).

- But at the Datalink layer, the addressing is MAC address (48 bit Physical Address), and this address is burned into the network card permanently.
- The purpose of Address Resolution Protocol (ARP) is to find out the MAC address of a device in your Local Area Network (LAN), for the corresponding IP address, which network application is trying to communicate.

REVERSE ADDRESS RESOLUTION PROTOCOL:

- The Reverse Address Resolution Protocol (RARP) is an obsolete computer communication protocol used by a client computer to request its Internet Protocol (IPv4) address from a computer network, when all it has available is its link layer or hardware address, such as a MAC address.
- RARP manipulates the kernel's RARP table in various ways. The primary options are clearing an address mapping entry and manually setting up one. For debugging purposes, the RARP program also allows a complete dump of the RARP table.

IP ADDRESSING

IP ADDRESSING:

- An internet protocol (IP) address allows computers to send and receive information. There are four types of IP addresses: public, private, static, and dynamic.
- An IP address allows information to be sent and received by the correct

parties, which means they can also be used to track down a user's physical location.

- It is a unique address assigned to each computing device in an IP network.
- ISP assigns IP Address to all the devices present on its network.
- Computing devices use IP Address to identify and communicate with other devices in the IP network.

IP TERMINOLOGY:

- The Internet Protocol (IP) is the heart of the Linux messaging system. While Linux (more or less) strictly adheres to the layering concept - and it is possible to use a different protocol (like ATM) - IP is almost always the nexus through which packets flow.
- The IP implementation of the network layer performs routing and forwarding as well as encapsulating data.

THE HIERARCHICAL IP ADDRESSING SCHEME:

- A hierarchical IP address design means that addresses are assigned in a hierarchical manner, rather than randomly.
- An IP address is made up of 32 bits of information. These bits are divided into four sections (sometimes called octets or quads) containing 1 byte (8 bits) each.

NETWORK ADDRESSING:

- A network address is any logical or physical address that uniquely distinguishes a network node or device over a computer or telecommunications network.
- It is a numeric/symbolic number or address that is assigned to any device that seeks access to or is part of a network.
- A network address is a key networking technology component that facilitates identifying a network node/device and reaching a device over a network.
- It has several forms, including the Internet Protocol (IP) address, media access control (MAC) address and host address.

CLASS A-

Class A comprises networks 1.0.0.0 through 127.0.0.0. The network number is contained in the first octet. This class provides for a 24-bit host part, allowing roughly 1.6 million hosts per network.

CLASS B-

Class B contains networks 128.0.0.0 through 191.255.0.0; the network number is in the first two octets. This class allows for 16,320 nets with 65,024 hosts each.

CLASS C-

Class C networks range from 192.0.0.0 through 223.255.255.0, with the network number contained in the first three octets. This class allows for nearly 2 million networks with up to 254 hosts.

3.3 ALGORITHM:

1. START
2. Login into Remote Desktop Connection using your IP address.
3. Open terminal and login into serverc.
4. Select or identify the correct disks to be used for LVM. (OR)
5. Type and go to the fdisk /dev/vdb and give the commands. Give the size of the storage of your wish.
6. Type part probe or udevadm settle to update your kernel.
7. Create a Physical Volumes (PV) on the disk.
8. Create the Volume Group (VG) on the Physical Volumes.
9. Create Logical Volumes (LV) on the Volume Group.
10. Create a filesystem for the logical volumes.
11. Install the targetcli package
12. Create a disk of the required size with a specific name. After creating a disk, that particular disk will be in a deactivated state, until the disk is created in a logical unit (LUN).
13. Create a server IQN address with a name used as an identifier. IQN is created in the iSCSI path of targetcli.
14. Create client IQN address with the same name.
15. Form a Logical Unit for the created disk. After creating LUN for the disk, it will change to an activated state.

16. In the path of portals, a default IP is created. Delete that IP and assign an IP for the target.
17. Save and exit from targetcli.
18. Start the firewall using the `systemctl` command and make the service permanent.
19. The configuration of the target is successful.
20. Install `iSCSI-initiator-utils` package to configure iSCSI initiator.
21. Discover the target using `iscsiadm` command.
22. Update the server IQN address in the `Initiatorname`.
23. Login to the target using `iscsiadm` command.
24. The configuration of the initiator is successful.
25. STOP

3.4 SUMMARY

Firstly, configure the iSCSI target on the server that will be exporting its storage over the network, and then set up an iSCSI initiator on a client system to connect to the target server. Once connected, the iSCSI initiator client is able to make use of the disks exported on the iSCSI target server as though they were locally attached. This includes performing operations such as partitioning or creating a file system and then mounting the disks and accessing them over the network to read and write data via iSCSI.

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE

4.1 RESULT AND DISCUSSION

```
Activities Terminal
Tue 23-29
root@serverc~
File Edit View Search Terminal Help
(root@serverc ~) # history
1 lsblk
2 fdisk /dev/vdb
3 udevadm settle
4 lsblk
5 pvcreate /dev/vdb1
6 pvdisplay
7 pvs
8 vgcreate -s 4 vgl /dev/vdb1
9 vgdisplay
10 vgs
11 lvcreate -L 16 -n lvi vgl
12 lvdisplay
13 lvs
14 lsblk
15 yum install targetcli
16 fdisk -l
17 targetcli
18 clear
19 history
20 clear
21 history
(root@serverc ~) #
```

```
Activities Terminal
192.168.0.38
root@serverd~
File Edit View Search Terminal Help
(root@serverd ~) # mkdir /test
(root@serverd ~) # mount /dev/sda1 /test
(root@serverd ~) # mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=396532k,nr_inodes=99133,mode=755)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,mode=755)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,seclabel,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,xattr,release_agent=/usr/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,net_cls,net_prio)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,freezer)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,devices)
cgroup on /sys/fs/cgroup/bklcio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,bklcio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,cpuset)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,cpu,cpuacct)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,hugetlb)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,rdma)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,pids)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,perf_event)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/vda1 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
rpc_pipefs on /var/lib/ntfs/rpc_pipefs type rpc_pipefs (rw,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=42,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=19835)
queue on /dev/queue type queue (rw,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime,seclabel)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=84096k,mode=700)
/dev/sda1 on /test type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
(root@serverd ~) # df -h
Filesystem Type Size Used Avail Use% Mounted on
devtmpfs devtmpfs 388M 0 388M 0% /dev
tmpfs tmpfs 411M 0 411M 0% /dev/shm
tmpfs tmpfs 411M 11M 400M 3% /run
tmpfs tmpfs 411M 0 411M 0% /sys/fs/cgroup
/dev/vda1 xfs 16G 1.5G 8.6G 15% /
tmpfs tmpfs 82M 0 82M 0% /run/user/0
/dev/sda1 xfs 495M 29M 466M 6% /test
(root@serverd ~) # lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 8:0 0 500M 0 disk
└─vda1 8:1 0 500M 0 part /test
vdb 252:0 0 16G 0 disk
└─vdb1 252:1 0 16G 0 part /
vdb 252:16 0 5G 0 disk
(root@serverd ~) # reboot -f
Rebooting.
```

4.2 PERFORMANCE

As I have expected there was no problem during the performing this project all the results are what I have expected. Finally, the test outputs came successfully.

CHAPTER 5

SUMMARY AND CONCLUSION

- LVM allows you to efficiently manage and utilize your physical disk space.
- This project explains a brief overview of logical volume management which is a very important framework of the Linux operating system. Also, these options fail to resolve the conflicts between the varying storage needs of multiple users. That is why LVM is considered as an essential component of the Linux based systems.
- iSCSI provides cheap connectivity as compared to FC. An existing network can be used for iSCSI implementation.
- As iSCSI is used for block storage, it is very fast.
- IT folks don't need a lot of training on iSCSI storage since it is very easy to understand and implement.
- iSCSI is an IP-based protocol. It leverages the interoperability benefits of TCP/IP and Ethernet.

5.1 REFERENCES

Here are few resources to learn more about the project.

- Linux-iSCSI project. <http://linux-iscsi.sourceforge.net>.
- Linux kernel sources version 2.4.23-pre5. <http://www.kernel.org>.
- Project: Intel iSCSI reference implementation.
<http://sourceforge.net/projects/intel-iscsi>.

- Project: iSCSI enterprise target.
<http://sourceforge.net/projects/iscsitarget>
- S. Aiken, D. Grunwald, and J. W. Andrew R. Pleszkun. A performance analysis of the iSCSI protocol. In 11th NASA Goddard, 20st IEEE Conference on Mass Storage Systems and Technologies (MSST2003), Apr. 2003.
- Redhat.com
- url: <https://www.redhat.com/en>

5.2 APPENDIX

A SCREENSHOTS

```

root@serverc:~# vgs
VG #PV #LV #SN Attr VSize VFree
vg1 1 0 0 wz--n- <2.00g <2.00g
root@serverc:~# lvcreate -L 1G -n lv1 vg1
Logical volume "lv1" created.
root@serverc:~# lvs
--- Logical volume ---
LV Path                /dev/vg1/lv1
LV Name                 lv1
VG Name                 vg1
LV UUID                 d35b1d-cyKF-opAU-NABL-sZMd-5Qyi-lidxUH
LV Write Access         read/write
LV Creation host, time serverc.lab.example.com, 2021-11-09 00:05:19 +0530
LV Status                available
# open                  0
LV Size                 1.00 GiB
Current LE              256
Segments                1
Allocation               inherit
Read ahead sectors      auto
  - currently set to    8192
Block device            253:0

root@serverc:~# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
lv1 vg1 -wi-a----- 1.00g

root@serverc:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda 252:0 0 100G 0 disk
├─vda1 252:1 0 100G 0 part /
├─vdb 252:16 0 5G 0 disk
├─vdb1 252:17 0 2G 0 part
├─vgi-lv1 253:0 0 1G 0 lvm
└─vdb2 252:18 0 2G 0 part
root@serverc:~#
root@serverc:~#

```

```
Activities Terminal
Tue 23/29
root@serverc~
File Edit View Search Terminal Help
[root@serverc ~]# history
 1 lsblk
 2 fdisk /dev/vdb
 3 udevadm settle
 4 lsblk
 5 pvcreate /dev/vdb1
 6 vgdisplay
 7 pvs
 8 vgcreate -s 4 vg1 /dev/vdb1
 9 vgdisplay
10 vgs
11 lvcreate -L 1G -n lvi vg1
12 lvdisplay
13 lvs
14 lsblk
15 yum install targetcli
16 fdisk -l
17 targetcli
18 clear
19 history
20 clear
21 history
[root@serverc ~]#
```

```
Activities Terminal
192.168.0.38
root@serverc~
File Edit View Search Terminal Help
Command (m for help): n
Partition type
  p primary (2 primary, 0 extended, 2 free)
  e extended (container for logical partitions)
Select (default p): p
Partition number (3,4, default 3):
First sector (8398656-10485759, default 8398656):
Last sector, +sectors or +size(K,M,G,T,P) (8398656-10485759, default 10485759): +2G
Error: out of range
Last sector, +sectors or +size(K,M,G,T,P) (8398656-10485759, default 10485759): +1G
Error: out of range
Last sector, +sectors or +size(K,M,G,T,P) (8398656-10485759, default 10485759): +900M
Created a new partition 3 of type 'Linux' and of size 900 MiB.
Command (m for help): t
Partition number (1-3, default 3): 1
Hex code (type l to list all codes): 8e
Changed type of partition 'Linux' to 'Linux LVM'.
Command (m for help): w
The partition table has been altered.
Syncing disks.
[root@serverc ~]# partprobe
[root@serverc ~]# fdisk -l
Disk /dev/vda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x16a1e957

Device            Boot  Start      End  Sectors  Size Id Type
/dev/vda1         *      2048 20971486 20969439   10G 83 Linux

Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb6abe8dd

Device            Boot  Start      End  Sectors  Size Id Type
/dev/vdb1         2048  4196351  4194304    2G 8e Linux LVM
/dev/vdb2        4196352  8398655  4194304    2G 8e Linux LVM
/dev/vdb3        8398656 10233855 1043200    900M 83 Linux

Disk /dev/mapper/vg1-lvi: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
```

```

root@serverc~
File Edit View Search Terminal Help
PV Name      /dev/vdb1
VG Name      vg1
PV Size      2.00 GiB / not usable 4.00 MiB
Allocatable  yes
PE Size      4.00 MiB
Total PE     511
Free PE      255
Allocated PE 256
PV UUID      T93YLL-wZt-9UDH-10m-3t40-mdiu-DY1B14

[root@serverc ~]# lvcreate -l 224 -n iscsi_target iscsi
Logical volume 'iscsi_target' created.
[root@serverc ~]# ls
LV          VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
iscsi_target iscsi   -wi-a----- 896.00m
lv1         vg1     -wi-a----- 1.00g
[root@serverc ~]# targetcli
targetcli shell version 2.1.fb49
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/backstores> ls
o- backstores ..... [Storage Objects: 0]
o- block ..... [Storage Objects: 0]
o- fileio ..... [Storage Objects: 0]
o- pscsi ..... [Storage Objects: 0]
o- ramdisk ..... [Storage Objects: 0]
/backstores> cd block
/backstores/block> create disk /dev/iscsi/iscsi_target
Created block storage object disk using /dev/iscsi/iscsi_target.
/backstores/block> ls
o- block ..... [Storage Objects: 1]
o- disk ..... [dev/iscsi/iscsi_target (896.0MiB) write-thru optimize]
o- alua ..... [ALUA Groups: 1]
o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
/backstores/block> cd ..
/backstores> cd ..
/ > ls
/ > Z
o- backstores ..... [Storage Objects: 1]
o- disk ..... [dev/iscsi/iscsi_target (896.0MiB) write-thru optimize]
o- alua ..... [ALUA Groups: 1]
o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
o- fileio ..... [Storage Objects: 0]
o- pscsi ..... [Storage Objects: 0]
o- ramdisk ..... [Storage Objects: 0]
o- iscsi ..... [Targets: 0]
o- loopback ..... [Targets: 0]
/ > cd iscsi
/iscsi> ls
o- iscsi ..... [Targets: 0]
/iscsi> create iqn.2021-11.com.example:server
Created target iqn.2021-11.com.example:server

```

```

root@serverc~
File Edit View Search Terminal Help
o- lun0 ..... [block/disk (/dev/iscsi/iscsi_target) (default to alua)]
o- portals ..... [Portals: 1]
o- 172.25.250.12:3260 ..... [TPGS: 1]
o- iqn.2021-11.com.example:server ..... [no-gen-acls, no-auth]
o- tpg1 ..... [ACLs: 1]
o- acls ..... [Mapped LUNs: 1]
o- lun0 ..... [LUNs: 1]
o- lun0 ..... [block/disk (/dev/iscsi/iscsi_target) (default to alua)]
o- portals ..... [Portals: 1]
o- 192.168.0.30:3260 ..... [no]
o- loopback ..... [Targets: 0]
/ > saveconfig
Configuration saved to /etc/target/saveconfig.json
/ > exit
Global pref auto save on: false
Exit in config saved in /etc/target/backup/.
Configuration saved to /etc/target/saveconfig.json
[root@serverc ~]# systemctl start targetcli
Failed to start targetcli.service: Unit targetcli.service not found.
[root@serverc ~]# systemctl start target
[root@serverc ~]# systemctl enable target
[root@serverc ~]# firewall-cmd --permanent --add-ports=3260/tcp
usage: firewall-cmd man page
[root@serverc ~]# firewall-cmd --reload
success
[root@serverc ~]# firewall-cmd --permanent --add-port=3260/tcp
warning: ALREADY ENABLED: 3260:tcp
success
[root@serverc ~]# firewall-cmd --reload
success
[root@serverc ~]# ifconfig
enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 172.25.250.12 netmask 255.255.255.0 broadcast 172.25.250.255
inet6 fe80::2177:5362:387:c217 prefixlen 64 scopeid 0x20<link>
ether 52:54:00:09:fa:0c txqueuelen 1000 (Ethernet)
RX packets 97896 bytes 14040988 (13.9 MiB)
RX errors 0 dropped 84589 overruns 0 frame 0
TX packets 8893 bytes 849206 (829.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 6 bytes 416 (416.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6 bytes 416 (416.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@serverc ~]#

```

```
File Edit View Search Terminal Help
192.168.0.38
root@serverd:~

iscsiadm: cannot make connection to 192.168.1.10: Network is unreachable
iscsiadm: cannot make connection to 192.168.1.10: Network is unreachable
iscsiadm: connection login retries (reopen max) 5 exceeded
iscsiadm: Could not perform SendTargets discovery: encountered connection failure
[root@serverd ~]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.250.12 --discover
192.168.0.38:3260,1 iqn.2021-11.com.example.server
172.25.250.12:3260,1 iqn.2021-11.com.example.server
[root@serverd ~]# man iscsiadm
[root@serverd ~]# Login, must use a node record id found by the discovery:
-bash: Login: command not found
[root@serverd ~]#
[root@serverd ~]# iscsiadm --mode node --targetname iqn.2021-11.com.example.server --portal 172.25.250.12 --login
Logging in to [iface: default, target: iqn.2021-11.com.example.server, portal: 172.25.250.12,3260] (multiple)
Login to [iface: default, target: iqn.2021-11.com.example.server, portal: 172.25.250.12,3260] successful.
[root@serverd ~]# fdisk -l
Disk /dev/vda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x16a1e057

Device      Boot Start      End  Sectors  Size Id Type
/dev/vda1   *    2048 20971486 20969439   10G 83 Linux

Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x16a1e057

Disk /dev/sda: 896 MiB, 939524096 bytes, 1835008 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@serverd ~]# mke2fs -t xfs /dev/vdb
mke2fs 1.44.3 (10-July-2018)
Creating filesystem with 1310720 4k blocks and 327680 inodes
Filesystem UUID: 52b1c4f2-0d5f-4e0b-88d0-3709eac02139
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[root@serverd ~]# fdisk -l
Disk /dev/vda: 10 GiB, 10737418240 bytes, 20971520 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x16a1e057
```

```
File Edit View Search Terminal Help
192.168.0.38
root@serverd:~

vda 252:0 0 100 0 disk
--vdal 252:1 0 100 0 part /
vdb 252:16 0 50 0 disk
[root@serverd ~]# iscsiadm --mode node --targetname iqn.2021-11.com.example.server --portal 172.25.250.12 --login
Logging in to [iface: default, target: iqn.2021-11.com.example.server, portal: 172.25.250.12,3260] (multiple)
Login to [iface: default, target: iqn.2021-11.com.example.server, portal: 172.25.250.12,3260] successful.
[root@serverd ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 896M 0 disk
vda 252:0 0 100 0 disk
--vdal 252:1 0 100 0 part /
vdb 252:16 0 50 0 disk
[root@serverd ~]# rht-vmctl status all
-bash: rht-vmctl: command not found
[root@serverd ~]# parted /dev/sda
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Error: /dev/sda: unrecognised disk label
Model: LTO-ORC disk (scsi)
Disk /dev/sda: 949M
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted) quit
[root@serverd ~]# udevadm settle
[root@serverd ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 896M 0 disk
vda 252:0 0 100 0 disk
--vdal 252:1 0 100 0 part /
vdb 252:16 0 50 0 disk
[root@serverd ~]# parted /dev/sda
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) help
align-check TYPE N check partition N for TYPE(min|opt) alignment
help [COMMAND] print general help, or help on COMMAND
mklabel, mktable LABEL-TYPE create a new disklabel (partition table)
mkpart PART-TYPE [FS-TYPE] START END make a partition
name NUMBER NAME name partition NUMBER as NAME
print [devices|free|list,all|NUMBER] display the partition table, available devices, free space, all found partitions, or a particular partition
quit exit program
rescue START END rescue a lost partition near START and END
resizepart NUMBER END resize partition NUMBER
rm NUMBER delete partition NUMBER
select DEVICE choose the device to edit
disk set FLAG STATE change the FLAG on selected device
disk toggle [FLAG] toggle the state of FLAG on selected device
set NUMBER FLAG STATE change the FLAG on partition NUMBER
toggle [NUMBER [FLAG]] toggle the state of FLAG on partition NUMBER
unit UNIT set the default unit to UNIT
version display the version number and copyright information of GNU Parted
```


B SOURCE CODE

SERVER SIDE

```
yum install targetcli
fdisk -l
fdisk /dev/vdb
partprobe
fdisk -l
pvcreate /dev/vdb1
pvdisplay
vgcreate lscsi /dev/vdb1
vgdisplay
lvcreate -l 767 -n iscsi_target lscsi
lvs
targetcli
ls
>cd backstores/
ls
cd block
create disk /dev/iscsi/iscsi_target
ls
cd ..
ls
cd iscsi
ls
create iqn.2021-11.com.example:server
ls
cd iqn.2021-11.com.example:server/
ls
```

```
cd tgp1/
ls
cd acls
create iqn.2021-11.com.example:client
ls
cd ..
ls
cd luns
ls
create /backstores/block/disk
ls
cd ..
ls
cd portals/
ls
delete ip_address=00.0.0.0 ip_port=3260
ls
create 172.25.250.12
ls
cd /
ls
saveconfig
exit
systemctl start target
systemctl enable target
firewall-cmd --permanent --add-port=3260/tcp
firewall-cmd --reload
```

CLIENT SIDE

```
yum install lscsi-initiator-utils
vim /etc/lscsi/initiatorname.iscsi
initiatorname= iqn.2021-11.com.example:client
wq
systemctl restart iscsid
systemctl enable iscsid
man iscsiadm
iscsiadm --mode discoverydb --type sendtargets --portal 172.25.250.12 --discover
man iscsiadm
iscsiadm --mode node --targetname iqn.2021-11.com.example:server --portal
172.25.250.12 --login
fdisk -l
mkfs.ext3 /dev/sda
fdisk -l
blkid
fdisk /etc/fstab
UUID="9f7cd92e-bcdd-4da0-4be9baec5ded" /mount ext3 default,_netdev 0 0
wq
mkdir /mount
iscsiadm --mode node --targetname iqn.2021-11.com.example:server --portal
172.25.250.12 --logout
fdisk -l
mount
reboot
```