International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2062118

# Booting of Diskless Workstation

**Suriya Prasath. S[1], Shwetha Ravikumar[1], Pradnya Suryawanshi[1], Anindita Khade[2]**
[1]Department of Computer Engineering, SIES GST, University of Mumbai, Nerul, Maharashtra, India
[2]Assistant Professor, Department of Computer Engineering, SIES GST, University of Mumbai, Nerul, Maharashtra, India

## ABSTRACT

This paper focuses on proposing a system for centralization of the IT infrastructure by building upon the concept of Diskless Booting. The client stations are configured to boot over an existing network using the Network Boot. The Client Station lacks an actual Hard Drive, which is responsible for storing the boot files and other necessary files. It consists of other hardware excluding the Hard Drive such as, RAM, Processor, etc. The network consists of DHCP (Dynamic Host Configuration Protocol) server which assigns IP addresses to several client stations connected. The NIS (Network Information Service) server stores the login credentials of all the users, enabling the users to login from any client station without any restrictions. The NFS (Network File System) server makes the files available for the user, even on the absence of local secondary storage. Security is ensured by SHA (Security Hash Algorithm), which encrypts the sensitive data by creating the hash values. Individual ports and services can be kept secured by using the Firewall service. The open source Cockpit Tool provides a GUI with which the administrator can conveniently carry out several admin-oriented tasks, such as memory management, user management.

**Keywords:** Network Information Service, Dynamic Host Configuration Protocol, Network File System, Firewall,

## I. INTRODUCTION

Centralization of IT infrastructure helps to strengthen the organisation's technology resources. When implemented correctly, it simplifies both the administrative tasks and the security management. Data management can also be achieved easily by adopting this system. Multiple added benefits such as, Higher Productivity, Ease of Maintenance and Increased Level of Security can also be achieved.

The capabilities of the Diskless Workstation Booting can be enhanced further, by means of various functional modules and protocols. This concept simply utilizes the Remote Server for storage and yet runs all the applications on the Local Client Stations. Efficiency can be achieved by having full powered PCs and large number of client stations, consuming decent amount of CPU and RAM to run all their applications locally.

The client station requires Bootable PXE network card which queries the server (configured with both DHCP and TFTP server) to fetch its configuration and location of the necessary boot files. After booting the PXE image, the client boots the kernel image (with arguments specified) fetched via TFTP server. These arguments help the kernel to self-configure itself and also helps the specified path to get mounted via NFS share where / directory is located.

### A. Motivation

Points specified below encouraged us for development of this project:

1. Remarkable cutting of the initial capital and implementation cost.
2. Enhances the productivity and resource utilization.
3. Lowering of the power consumption and cooling requirements.
4. Ease the troubleshooting process and, refines the user convenience and administrative tasks.

## II. LITERATURE REVIEW

In the early 90's, Windows 3.x operating system was the first OS to support Network Boot (also referred as Netboot). Netboot enables the process of booting the computer from a network rather than a local drive. This method of booting can be used by routers, diskless work- stations, and centrally managed computers (also referred to as thin clients).

In late 1980s/early 1990s, network boot was utilized to cut down the expenses of a disk drive, as a decently sized hard disk drive (HDD) used to cost substantial amount, equalling the price of the CPU.

The modern-day personal computers usually provide an option to boot from the network, in their BIOS (Basic Input Output System) via the PXE (Pre-Execution Environment). Moreover, in 1998, Power-PC (G3, G5) Mac systems were designed to boot from their firmware to a network disk via Netboot. Old computers without Netboot firmware support, utilized a floppy disk or a flash drive consisting of the software to boot from the network.

Also, the evolution of the LINUX led to the procurement of network boot protocol as an open source project. In the upcoming years, the open source community actively took part in the development of various packages to implement the Network Boot. DBRL (Diskless Remote Boot in Linux) is one such tool developed by the Linux community. Hence this proves that, the core ideology of the diskless workstation booting was being preserved since the 90's.

## III. SYSTEM ANALYSIS

The overall system comprises of two types of workstations:

1. Client Station: Minimal systems with no primary storage devices attached to them.
2. Server Station: Systems consisting of various modules or packages such as DHCP, TFTP, NFS, NIS, etc.

A high capacity storage box is an essential part of this system architecture, replacing the conventional local storage-based setup. A network switch is the mainstay of the whole system to which all the client stations are connected. The entire architecture is safeguarded from the external threats by keeping the whole network under the inspection of the firewall. Security is further ensured by configuring the SELINUX to provide access controls for the individual files and folders. To preserve the core principle of consistency, ease of use, optimum performance, and interactive experience, Cockpit (an open source project) is used. The UI includes diverse functionalities such as login/logout, memory and storage management, etc.

### A. Overview of the Modules

1. DHCP: The DHCP Server dynamically assigns the IP address to the client stations demanding for it. When a client station sends a request to the DHCP server, it acknowledges the client station by allocating it a random IP from the list of available IP addresses.
2. NFS: This stores the root file system of all the users. It is utilized for mounting the local file systems of the respective client stations over the network.
3. TFTP: It stores the boot image file for the respective operating system. This boot image file is then used by the client stations to initiate the operating system.

4. NIS: This module is used for authentication purposes. The credentials entered during the login process is sent to the NIS server, and if a match is found, it returns an appropriate confirmation message and hence the login is approved.

### B. System Architecture

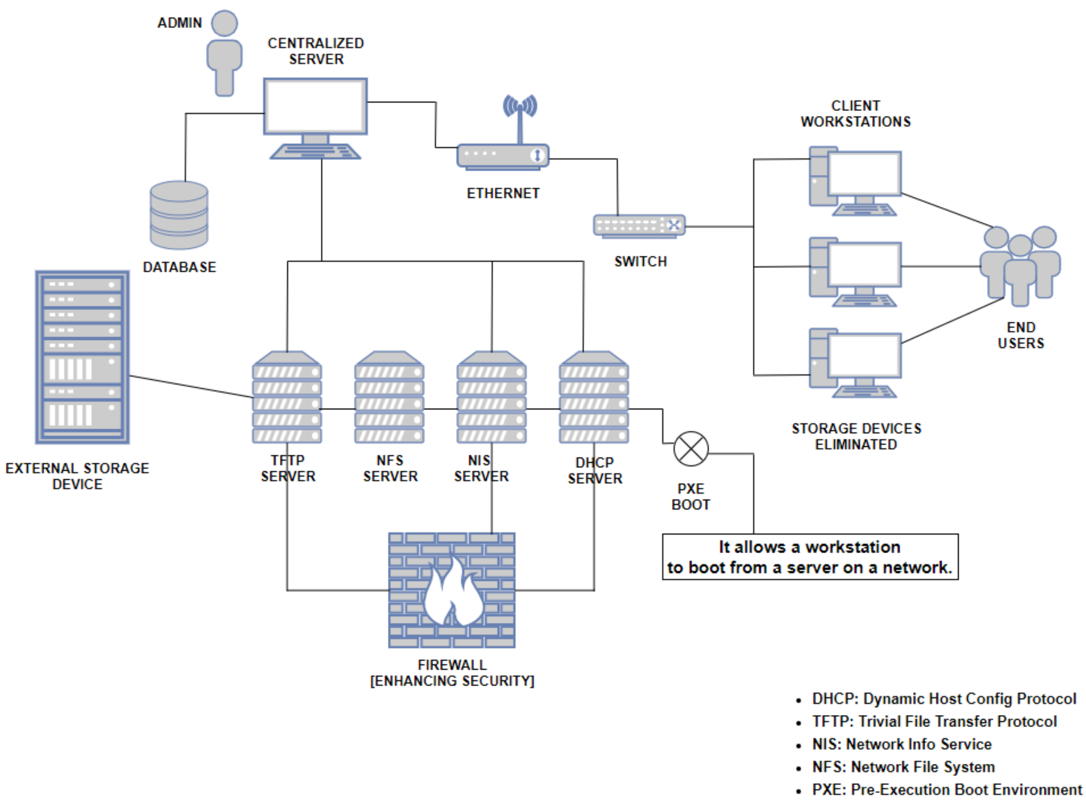


Fig: System Architecture

## IV. IMPLEMENTATION

### A. Operating System

We implemented this project using LINUX operating system due to its increased stability and compatibility with high availability tasks. Adding to it, LINUX serves a high-performance computation facility. Amongst various LINUX distributions, we chose CentOS (version 7), due to the fact that, it is a replica of RHEL (Red Hat Enterprise Linux), which is considered to be most widely used in the Corporate IT world. Also, it is very much appreciated for its high customizability, security and stability.

### B. OS files for the Client Machine

After the fresh installation of CentOS-7 on the server side, the necessary OS files for the client workstations can be copied by using the following command:

[root@dlp /]# cp -a /{b,e,h,l,m,o,ro,sb,sr,u,v}* /<destination path>/
[root@dlp /]# mkdir /<destination path>/{dev,proc,run,sys,tmp}

### C. TFTP Configuration

It stores the necessary boot files, which is utilized by the client stations to initiate the OS. At first, a new directory must be created at the / partition, with the name “tftpboot”. The tftpboot directory comprises of following files and directory:

1. initrd.img (file)
2. vmlinuz (file)
3. pxelinux.0 (file)
4. pxelinux.cfg (directory)

The initrd.img and vmlinuz can be obtained from “http://mirror.centos.org/”. The pxelinux.0, a syslinux derivative mainly used for achieving the network boot following the Intel PXE Specifications, can be copied from “/usr/share/syslinux” directory. Pxelinux.cfg is the directory within the tftpboot, which consists of the configuration file i.e., “default” in our case. The file “default” can be renamed with ‘IP’, ‘UUID’ or ‘MAC- ID’ of the respective client. The IP address must be mentioned in the uppercase hexadecimal values. Following commands must be fired for successfully configuring the TFTP module:

[root@dlp /]# cd /
[root@dlp /]# mkdir tftpboot
[root@dlp /]# chmod 777 tftpboot
(**Note:** After downloading the initrd.img and vmlinuz files from the above-mentioned source)
[root@dlp Downloads]# cp initrd.img /tftpboot/
[root@dlp Downloads]# cp vmlinuz /tftpboot/
[root@dlp /]# cp pxelinux.0 /usr/share/syslinux/ /tftpboot/
[root@dlp /]# cd /tftpboot
[root@dlp tftpboot]# mkdir pxelinux.cfg
[root@dlp tftpboot]# cd pxelinux.cfg
[root@dlp pxelinux.cfg]# vi default
(**Note:** default file configuration)
default cent
label cent
kernel vmlinuz
(**Note:** The below configuration must be written in a continuous stretch)
append initrd=initrd.img ip=dhcp root=/dev/nfs nfsroot=<IP address of server>:<OS file path for the client> selinux=0 rw

### D. DHCP Configuration

For configuring this module, we need to begin by installing the "dnsmasq" package. This package is the combination of both DNS and DHCP functionalities. It consumes less system resources and moreover it comprises of various features required to carry out network boot. This package supports modern network technologies such as, IPv6, BootP, PXE, TFTP etc. For configuring this module following commands must be fired:

[root@dlp /]# yum install dnsmasq* -y
[root@dlp /]# cd /etc
**(Note:** Taking backup of existing dnsmasq.conf file**)**
[root@dlp etc]# mv dnsmasq.conf dnsmasq.conf.backup
[root@dlp etc]# vi dnsmasq.conf
**(Note:** DHCP configuration**)**
log-dhcp
dhcp-authoritative
dhcp-range=192.168.29.0,192.168.29.255,24h
pxe-service=x86PC,"Network Boot",pxelinux
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=<path of tftpboot directory>
**(Note:** Exit the dnsmasq.conf file**)**
[root@dlp /]# systemctl restart dnsmasq.service
[root@dlp /]# systemctl start dnsmasq.service
[root@dlp /]# systemctl enable dnsmasq.service
[root@dlp /]# systemctl status dnsmasq.service

### E. NFS Configuration

This module is configured, by beginning with the installation of the "nfs-utils" package. As explained earlier, it is used for mounting the associated files and directories over the network, for loading the client machines. Moreover, it is also used for mounting the file systems of the respective clients via the network. Following are the important files that needs to be configured before initiating the diskless workstation:

1. /etc/exports
2. /etc/fstab

The "exports" file defines the exported files and the directory, whereas the "fstab" (must configure the client's fstab) file is used for mounting the NFS directory of the system.

**Configure the /etc/exports**

[root@dlp /]# yum install nfs-utils* -y
[root@dlp /]# cd /etc
**(Note:** Directories and files to be exported via network must be mentioned in "exports" file**)**
[root@dlp etc]# vi exports
/Diskless-OS/Centos-7/ *(rw,fsid=0,no_root_squash,no_subtree_check,async,insecure)
/Diskless-OS/Centos-7/home *(rw,sync)

(**Note:** "/Diskless-OS/Centos-7/" mentioned here is the OS file path for the client machines)

**Configure the /etc/fstab (for the client machines)**

[root@dlp /]# cd /Diskless-OS/Centos-7/

[root@dlp Centos-7]# cd etc

[root@dlp etc]# pwd

/Diskless-OS/Centos-7/etc

[root@dlp etc]# vi fstab

#

# /etc/fstab

# Created by anaconda on Mon Apr 13 12:38:36 2020

| | | | | |
|---|---|---|---|---|
| tmpfs | /dev/shm | tmpfs | defaults | 0 0 |
| devpts | /dev/pts | devpts | gid=5, mode=620 | 0 0 |
| sysfs | /sys | sysfs | defaults | 0 0 |
| proc | /proc | proc | defaults | 0 0 |

(**Note:** Exit the fstab file)

[root@dlp /]# systemctl restart nfs-server.service nfs.service

[root@dlp /]# systemctl start nfs-server.service nfs.service

[root@dlp /]# systemctl enable nfs-server.service nfs.service

[root@dlp /]# systemctl status nfs-server.service nfs.service

## F. NIS Configuration

This module needs to be configured both from the server side and the client side. Giving the server top priority, NIS to be configured from the server side, and then proceed further by configuring the NIS from the client side.

### 1. NIS (Server Side)

Before moving ahead with the configuration, we need to install the "ypserv" and "rpcbind" packages. The ypserv package ensures that, the NIS Database runs from the server side and, the rpcbind package, binds up the client stations with the server station, and vice-versa. Following are the commands for successfully configuring the NIS from server side:

(**Note:** Using nmtui specify the hostname of the server, in our case, hostname of server is dlp.srv.world and IP address of Server Station is 192.168.29.230)

[root@dlp /]# yum install ypserv* rpcbind* -y

[root@dlp /]# nisdomainname srv.world

[root@dlp /]# vi /etc/sysconfig/network

# Created by anaconda

NISDOMAIN=srv.world

(**Note:** Write and quite the network file)

[root@dlp /]# vi /var/yp/securenets

255.0.0.0 127.0.0.0

255.255.255.0 192.168.29.0

(**Note:** Write and quite the securenets file)

[root@dlp /]# vi /etc/hosts 192.168.29.230 dlp.srv.world

(**Note:** Write and quite the hosts file)

[root@dlp /]# systemctl start rpcbind ypserv ypxfrd yppasswdd [root@dlp /]# systemctl enable rpcbind ypserv ypxfrd yppasswdd

[root@dlp /]# /usr/lib64/yp/ypinit -m

At this point, we have to construct a list of the hosts which will run NIS servers. dlp.srv.world is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, type a <control D>.

next host to add: dlp.srv.world

next host to add:

The current list of NIS servers looks like this: dlp.srv.world

Is this correct? [y/n: y] y

We need a few minutes to build the databases...

Building /var/yp/srv.world/ypservers...

Running /var/yp/Makefile...

gmake[1]: Entering directory `/var/yp/srv.world'

Updating passwd.byname...

Updating passwd.byuid...

Updating group.byname...

Updating group.bygid...

Updating hosts.byname...

Updating hosts.byaddr...

Updating rpc.byname...

Updating rpc.bynumber...

Updating services.byname...

Updating services.byservicename...

Updating netid.byname...

Updating protocols.bynumber...

Updating protocols.byname...

Updating mail.aliases...

gmake[1]: Leaving directory `/var/yp/srv.world'

dlp.srv.world has been set up as a NIS master server.

Now you can run ypinit -s dlp.srv.world on all slave server.

[root@dlp /]# cd /var/yp

[root@dlp yp]# make

gmake[1]: Entering directory `/var/yp/srv.world'

Updating netid.byname...

gmake[1]: Leaving directory `/var/yp/srv.world'

## 2. NIS (Client Side)

For configuring the NIS on the client side, initially we need to install "ypbind" and "rpcbind" packages. The ypbind package is responsible for connecting the processes on a NIS client, to services on an NIS server. Following are the commands for successfully configuring the NIS from the client side:

**(Note:** Using nmtui specify the hostname of the client, in our case, hostname of client is www.srv.world)

[root@www /]# yum install ypbind* rpcbind * -y
[root@www /]# ypdomainname srv.world
[root@www /]# vi /etc/sysconfig/network #
#Created by anaconda
NISDOMAIN=srv.world

**(Note:** write and quite the network file)

[root@www /]# vi /etc/hosts
192.168.29.230 dlp.srv.world

**(Note:** Using "authconfig-tui" command, select "Use NIS" and "Use MD5 Passwords" options and proceed by selecting "Next". Now in NIS Settings, enter the appropriate Domain name and Server name)

**(Note:** Write and quite the hosts file)

[root@www /]# authconfig --enablenis --nisdomain=srv.world --nisserver=dlp.srv.world --ablemkhomedir --update
[root@www /]# systemctl start rpcbind ypbind
[root@www /]# systemctl enable rpcbind ypbind

## V. CONCLUSION

This system proposes an efficient environment to facilitate the centralization process, without compromising on the security and functionality. Administrator is given the top priority by including the GUI functionality via the open source cockpit project, to conveniently manage the client stations. Moreover, this system can be further scaled up with more added functionality.

## VI. REFERENCES


[1]. B. S. Sonawane and R.R. Deshmukh. 2018. International Journal of Innovations in Engineering and Technology. (April 2018) ISSN NO:2319-1058

[2]. August Anthony N. Balute and Dennis B.Gonzales. 2016. International Journal of Computer Applications. DOI:10.5120/ijca2016912281

[3]. Abhijeet Daga and Manish Ghumnani. 2015. Global Journal of engineering Science and Researchs. (June 2015). ISSN NO:2348-8034

[4]. Akshay Chavan and Swapnil Gujar. 2013. International Journal of Engineering Research and Technology. (May 2013). ISSN NO:2278-0181

[5]. Che-Yuan Tu and Wen-Chieh Kuo. 2009. IEEE/ACM International Conference on Grid Computing. (Oct 2009) ISSN NO:2152-1085.

[6]. Brian Zammit. "Linux Terminal Server Project: Server Configuration Guidelines"

[7]. Jim McQuillan. 2000. Annual Linux Showcase and Conference. (Oct 2000). Corpus ID:34076073



**Cite this article as** :

Suriya Prasath S, Shwetha Ravikumar, Pradnya Suryawanshi, Anindita Khade, "Booting of Diskless Workstation", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 2, pp. 398-405, March-April 2020. Available at doi : https://doi.org/10.32628/CSEIT2062118

Journal URL : http://ijsrcseit.com/CSEIT2062118