

UNIVERSIDAD DE CASTILLA LA MANCHA  
ESCUELA SUPERIOR DE INFORMÁTICA

# PXE, TFPT E INSTALACIÓN DESATENDIDA

Gestión y Administración de Redes

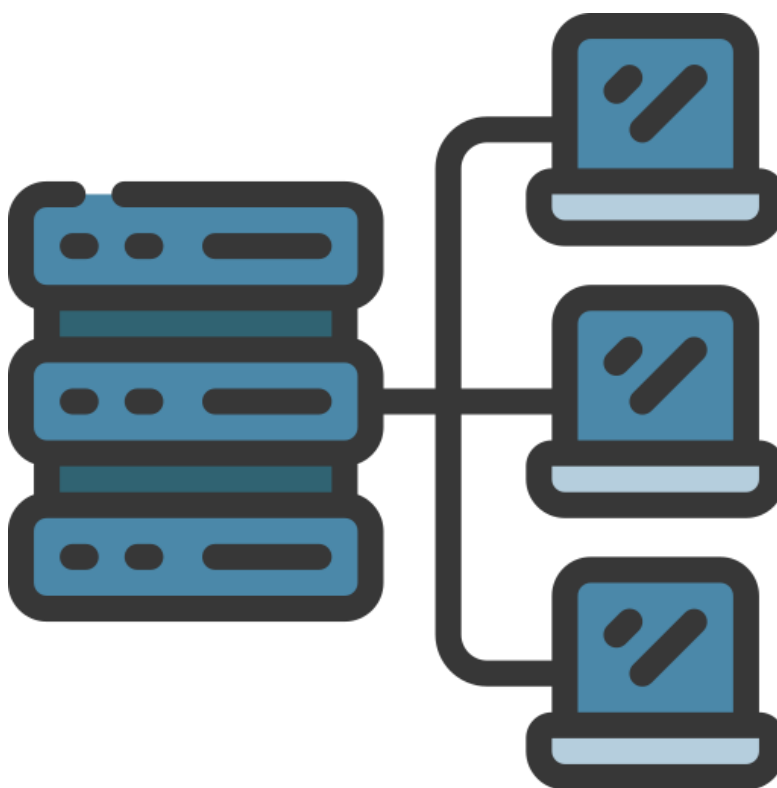
Alonso Villamayor Moreno

Víctor Ortega Gómez

[Alonso.Villamayor@alu.uclm](mailto:Alonso.Villamayor@alu.uclm.es)  
[.es](mailto:Victor.Ortega2@alu.uclm.es)

[Victor.Ortega2@alu.uclm.es](mailto:Victor.Ortega2@alu.uclm.es)

Curso 2022/2023



## Índice

Índice .....	2
Introducción .....	3
Requisitos del sistema .....	3
Licencias .....	3
Comparativa con Canonical MaaS .....	4
Comparativa con The Foreman .....	4
Parámetros de Entrada .....	4
Formas de acceder al servicio .....	4
Monitorización y Eventos.....	4
Métodos de instalación de los servicios para CentOS .....	5
Interacción con los servicios .....	8
Servidor PXE y TFTP redundantes.....	9
Configuración TCP/IP en un nodo cualquiera .....	9
Asignación de una IP determinada según la dirección MAC.....	11
Servir imagen de arranque a un nodo según la dirección MAC .....	11
Instalación desatendida según la dirección MAC.....	12
Referencias.....	14

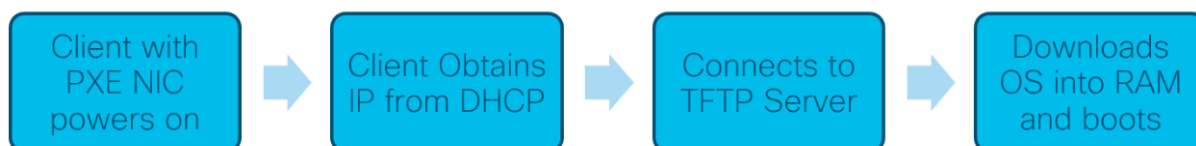
## Introducción

Si se va a diseñar e implementar una red desde cero, es posible que los nodos de la red necesiten un sistema operativo e ir uno por uno realizando la instalación no es el método ni más rápido ni más eficiente para llevar a cabo dicha tarea. Es por esto por lo que en las redes se puede hacer uso de PXE (Preboot eXecution Environment) y TFTP (Trivial File Transfer Protocol).

El concepto de PXE se creó durante los inicios de los protocolos BOOTP, DHCP y TFTP llegando a convertirse en un estándar UEFI en el año 2015. PXE es un entorno que nos permitirá arrancar e instalar el sistema operativo que se desee, siempre que este pueda ser iniciado a través de la red, de manera independiente de los dispositivos de almacenamiento de datos disponibles (como discos duros) o de los sistemas operativos instalados [1][2][3].

El protocolo PXE hace uso de una combinación de los protocolos DHCP y TFTP con pequeñas modificaciones en ambos. DHCP es utilizado para localizar el servidor de arranque apropiado y con TFTP se descarga el programa inicial de bootstrap y archivos adicionales.

Por otro lado, TFTP (Trivial File Transfer Protocol) es un protocolo cliente-servidor. Surgió a principios de los 80 y sirve para el intercambio de información y de archivos entre equipos y también para la regulación de esta transferencia. TFTP funciona a través de una conexión UDP [4].



## Requisitos del sistema

Para que PXE y TFTP puedan ser usados por los nodos de una red, el hardware de los dispositivos tiene que permitir el arranque mediante PXE. También la red debe de disponer de un servidor DHCP y otro TFTP (o cualquier otro que ponga a disposición un servicio de datos) que proporcione los datos que son necesarios para que los clientes puedan utilizarlo. Estos servidores pueden encontrarse en el mismo dispositivo [5][6].

## Licencias

Para la implementación de PXE existen posibilidades gratuitas como por ejemplo AOMEI PXE Boot para dispositivos Windows o también pxe-boot para dispositivos Linux, aunque estos traen por defecto un programa de arranque llamado pxelinux.0 que es proporcionado por el paquete de software syslinux [7][8].

Existen una gran variedad de programas que pueden ser utilizados para la implementación de un servidor TFTP tanto de pago como gratuitos. Algunos de este tipo de software gratuito que podemos encontrar sería AndFTP para dispositivos Android o Cyberduck para dispositivos Windows y macOS. Ejemplos de software de pago podrían ser FlashFXP para dispositivos Windows o SolarWinds que dispone de una versión gratuita pero la más completa es de pago. También existe software que nos permite implementar TFTP en Linux como el que viene por defecto u otros gratuitos y de código abierto como Tftpd32

[9][10][11].

## Comparativa con Canonical MaaS

MaaS (Metal as a Service) ofrece aprovisionamiento de estilo en la nube para servidores físicos. Es de código abierto y gratuito para usar, con soporte comercial disponible de Canonical. Por otro lado, el arranque PXE se refiere a la capacidad de arrancar una máquina a través de una tarjeta de interfaz de red (NIC). El arranque PXE requiere una tarjeta de interfaz de red que esté equipada con una API PXE que pueda ser accedida por el servidor que desea arrancar el dispositivo. También a diferencia de PXE, MaaS por defecto solo es capaz de instalar imágenes de Ubuntu Server y no entornos de escritorio [12][13].

## Comparativa con The Foreman

The Foreman es un proyecto de código abierto que es capaz de administrar los servidores durante su ciclo de vida, desde aprovisionamiento y configuración hasta la orquestación y supervisión. Además de interactuar con puppet, es capaz de integrar diferentes servicios, tales como DNS, DHCP, FreeIPA, SSH, TFTPd y PXE. Foreman proporciona una gestión completa de la configuración PXE de PXELinux, Grub, Grub2 y iPXE para una máxima flexibilidad en el arranque por red. Por esto podemos comprobar que The Foreman es capaz de hacer todo lo que PXE y TFTP hacen ya que en sus funcionalidades los engloba [14][15].

## Parámetros de Entrada

Parámetro	Valor por defecto	Significado	Observaciones
IP del servidor	192.168.10.100	Dirección IP que le será asignada al servidor.	La asignación se realizará de forma estática.
SO	CentOS 7	El sistema operativo que será utilizado.	
anaconda-ks.cfg		Archivo de configuración	
SELinux	Habilitado	Otorgará a los administradores más control sobre quién se puede acceder.	
Cortafuegos	Habilitado	Bloqueará accesos no autorizados.	

## Formas de acceder al servicio

Una vez montado el servidor PXE y TFTP los cuales, en nuestro caso son el mismo para ambos servicios y para DHCP, se podrá acceder a ellos simplemente accediendo a la URL [http://<ip\\_del\\_servidor>](http://<ip_del_servidor>). También a la hora de la instalación, los nodos son capaces de encontrar estos servicios gracias a DHCP, el cual les da la información que necesitan de cómo y dónde se encuentra el servidor al que necesitan acceder (en nuestro caso es el mismo como hemos indicado antes). Las conexiones que establecerán entre sí serán UDP.

## Monitorización y Eventos

Para la monitorización de PXE no existe un software específico con el que llevar a cabo esta tarea, pero sí se podría utilizar el software Wireshark con el que podemos analizar los paquetes de la red y comprobar así que las peticiones DHCP son contestadas de forma exitosa y que la transferencia y los paquetes enviados por TFTP a los nodos llegan también de forma exitosa suponiendo esto la correcta instalación de los sistemas operativos. Por otro lado, para monitorizar TFTP podemos acceder con un navegador a la URL del servidor <http://<servidor>:<puerto>>. Así podremos acceder a diferentes estadísticas o la configuración del propio servidor.

Respecto a los eventos, PXE no genera ningún tipo de reporte o notificación que indique ningún tipo de métrica, rendimiento, etc. TFTP y DHCP tampoco generan nada de lo mencionado anteriormente ya que son protocolos muy sencillos con funcionalidades muy básicas. No obstante, podemos observar el trabajo que realiza nuestro servidor consultando los mensajes de los en el archivo **/var/log/messages**.

## Métodos de instalación de los servicios para CentOS

Para la instalación del servicio PXE y TFTP en el sistema operativo CentOS (el cual ha sido usado para este trabajo) deberemos lo primero descargar los paquetes **dhcp**, **tftp-server**, **ftp server (vsftpd)** y **xinetd** lo cual se podrá hacer con el siguiente comando [16].

```
# yum install dhcp tftp tftp-server syslinux vsftpd xinetd
```

El siguiente paso será la configuración de DHCP para habilitar PXE. Para esto deberemos modificar el archivo DHCP que se crea al instalar el paquete **dhcp** para configurar este servicio. Este archivo se encontrará en **/etc/dhcp/dhcpd.conf**. Deberá quedar algo así:

```
[root@localhost ~]# cat /etc/dhcp/dhcpd.conf
allow booting;
allow bootp;
allow unknown-clients;
ddns-update-style interim;
ignore client-updates;
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.120 192.168.10.200;
    option routers 192.168.10.100;
    filename "pxelinux.0";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    next-server 192.168.10.100;
}
```

Tras configurar este archivo se deberá configurar el servidor TFTP modificando su archivo de configuración el cual se encuentra en la ruta **/etc/xinetd.d/tftp** de tal forma que el parametro *disable* = *yes* lo cambiemos a *disable* = *no*. Deberá quedar algo así:

```
GNU nano 2.3.1 Fichero: /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial file transfer \
# protocol. The tftp protocol is often used to boot diskless \
# workstations, download configuration files to network-aware printers, \
# and to start the installation process for some operating systems.
service tftp
{
    socket_type        = dgram
    protocol           = udp
    wait               = yes
    user               = root
    server              = /usr/sbin/in.tftpd
    server_args         = -s /var/lib/tftpboot
    disable             = no
    per_source          = 11
    cps                 = 100 2
    flags               = IPv4
}
```

Una vez modificados ambos archivos, deberemos copiar los archivos de arranque de red necesarios en la ruta **/var/lib/tftpboot/**. Para ello ejecutaremos los siguientes comandos.

```
[[email protected] ~]# cp -v /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot
[[email protected] ~]# cp -v /usr/share/syslinux/menu.c32 /var/lib/tftpboot
[[email protected] ~]# cp -v /usr/share/syslinux/memdisk /var/lib/tftpboot
[[email protected] ~]# cp -v /usr/share/syslinux/mboot.c32 /var/lib/tftpboot
[[email protected] ~]# cp -v /usr/share/syslinux/chain.c32 /var/lib/tftpboot
[[email protected] ~]#
[[email protected] ~]# mkdir /var/lib/tftpboot/pxelinux.cfg
[[email protected] ~]# mkdir /var/lib/tftpboot/networkboot
[[email protected] ~]#
```

Al acabar de ejecutar los comandos, continuaremos montando la ISO del sistema operativo y copiando su contenido en el servidor FTP local. Para esto ejecutaremos los siguientes comandos.

```
[[email protected] ~]# mount -o loop CentOS-7-x86_64-DVD-1511.iso /mnt/
mount: /dev/loop0 is write-protected, mounting read-only
[[email protected] ~]# cd /mnt/
[[email protected] mnt]# cp -av * /var/ftp/pub/
```

Y copiaremos el archivo Kernel (vmlinuz) y el archivo initrd de la ISO del sistema operativo a el directorio **/var/lib/tftpboot/networkboot/**

```
[[email protected] ~]# cp /mnt/images/pxeboot/vmlinuz /var/lib/tftpboot/networkboot/
[[email protected] ~]# cp /mnt/images/pxeboot/initrd.img /var/lib/tftpboot/networkboot/
[[email protected] ~]#
```

Cuando finalicemos con el archivo ISO podremos desmontarlo con el comando siguiente.

```
[[email protected] ~]# umount /mnt/
```

A continuación, deberemos crear los archivos Kickstart y PXE. Para ello deberemos crear primero la contraseña raíz en una cadena cifrada. Esto lo podremos hacer así.

```
[[email protected] ~]# openssl passwd -1 [[email protected]]#  
$1$e2wrcGGX$tZPQKPsXVhNmbiGg53MN4l  
[[email protected] ~]#
```

Para el archivo Kickstart, tendremos que crear un nuevo Kickstart en el directorio **/var/ftp/pub** y deberá llamarse *centos7.cfg*. Para esto podemos coger el archivo por defecto del sistema el cual se coloca en la ruta **/root** con el nombre *anaconda-ks.cfg* y modificarlo según lo necesitemos.

```
GNU nano 2.3.1          Fichero: /var/ftp/pub/centos8/anaconda-ks.cfg  
  
#version=DEVEL  
# System authorization information  
auth --enableshadow --passalgo=sha512  
# Use CDROM installation media  
#cdrom  
url --url="ftp://192.168.10.100/pub/centos8/"  
# Use graphical install  
graphical  
# Run the Setup Agent on first boot  
firstboot --enable  
  
# Keyboard layouts  
keyboard --vckeymap=es --xlayouts='es'  
# System language  
lang es_ES.UTF-8  
  
# Network information  
network --bootproto=dhcp --device=enp0s3 --ipv6=auto --activate  
network --hostname=localhost.localdomain  
  
# Root password  
rootpw --iscrypted $6$0BqWe2nk0fSF18KT$05C.LG2q?U6qUNUWpK1BDUR7QyMU70y2DbeCNhATFdUCUb/1pz5nSWLVE0eN$  
# System services  
services --disabled="chronyd"  
# System timezone  
timezone Europe/Madrid --isUtc --nontp  
# System bootloader configuration  
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda  
autopart --type=lvm  
# Partition clearing information  
clearpart --none --initlabel
```

Para el archivo PXE, deberemos configurar el archivo **/var/lib/tftpboot/pxelinux.cfg/default** y copiar los siguientes parámetros dentro.

```

GNU nano 2.3.1          Fichero: /var/lib/tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0
timeout 30
MENU TITLE CENTOS PXE MENU
LABEL Centos7_x86
MENU LABEL Centos7_x64
KERNEL /networkboot/vmlinuz
APPEND initrd=/networkboot/initrd.img inst.repo=ftp://192.168.10.100/pub/centos7_9 ks=ftp://192.168.10.100/pub/centos7_9
LABEL 23452345
KERNEL /networkboot/vmlinuz
APPEND initrd=/networkboot/initrd.img inst.repo=ftp://192.168.10.100/pub/centos8 ks=ftp://192.168.10.100/pub/centos8

```

Tras modificar y crear los anteriores archivos, lo siguiente que deberemos hacer es ejecutar y habilitar los servicios xinetd, dhcp y vsftpd. Esto se puede hacer con los siguientes comandos.

```

[[email protected] ~]# systemctl start xinetd
[[email protected] ~]# systemctl enable xinetd
[[email protected] ~]# systemctl start dhcpd.service
[[email protected] ~]# systemctl enable dhcpd.service
[[email protected] ~]# systemctl start vsftpd
[[email protected] ~]# systemctl enable vsftpd

```

En caso de que SELinux esté habilitado, deberemos configurar la regla de SELinux para el servidor ftp lo cual podremos llevar a cabo con el siguiente comando.

```

[[email protected] ~]# setsebool -P allow_ftp_full_access 1

```

A continuación, abriremos los puertos del Firewall.

```

[[email protected] ~]# firewall-cmd --add-service=ftp --permanent
success
[[email protected] ~]# firewall-cmd --add-service=dhcp --permanent
success
[[email protected] ~]# firewall-cmd --add-port=69/tcp --permanent
success
[[email protected] ~]# firewall-cmd --add-port=69/udp --permanent
success
[[email protected] ~]# firewall-cmd --add-port=4011/udp --permanent
success
[[email protected] ~]# firewall-cmd --reload
success
[[email protected] ~]#

```

Por último, quedará que los nodos que deben hacer uso de este servicio sean iniciados por PXE y todos los procesos y servicios estarán en funcionamiento.

## Interacción con los servicios



PXE [17]:

- Parar el servicio (Linux): `/etc/init.d/blpxe stop`
- Reiniciar el servicio (Linux): `/etc/init.d/blpxe restart`
- Para reiniciar el servicio en el sistema de la demo (CentOS7): `systemctl restart xinetd`
- Realizar cambios: Para realizar cambios en el servicio PXE lo único que debemos hacer es modificar su archivo de configuración y reiniciar el servicio.

TFTP [18]:

- Parar el servicio (Linux): `/etc/init.d/bltftp stop`
- Reiniciar el servicio (Linux): `/etc/init.d/bltftp restart`
- Para reiniciar el servicio en el sistema de la demo (Centos7): `systemctl restart tftp`
- Realizar cambios: Para realizar cambios en el servicio TFTP lo único que debemos hacer es modificar su archivo de configuración y reiniciar el servicio.

DHCP:

- Para reiniciar el servicio en el sistema de la demo (CentOS7): `systemctl restart dhcpd`

Adicionalmente podemos consultar el estado de los servicios para poder encontrar fallos en este mediante este comando.

```
systemctl status <servicio>
```

donde <servicio> puede ser cualquiera de los anteriores, por ejemplo, para DHCP sería el siguiente.

```
systemctl status dhcpd
```

## Servidor PXE y TFTP redundantes

Los servicios PXE y TFTP si soportan redundancia. Para introducir en nuestra red un servidor PXE redundante, lo que deberemos hacer es instalar el segundo servidor en otro nodo de la misma subred, lo configuramos de la misma forma que se configuró el servidor PXE principal. Una vez hecho, lo siguiente será la configuración del servidor TFTP, para ello deberemos editar el archivo **pxe.conf** en el segundo servidor y añadir las siguientes líneas de configuración [19].

```
use_config_file=true
```

Y estableceremos la misma configuración que para el servidor principal de TFTP excepto por la dirección IP la cual deberá ser la del nuevo servidor y el directorio base que tendrá que ser también la del nuevo servidor. A continuación, reiniciamos el servidor redundante y ya estaría disponible el segundo servidor para cuando las peticiones PXE y TFTP fallen en el primero.

## Configuración TCP/IP en un nodo cualquiera

Para la configuración de este apartado y de los siguientes vamos a tener en cuenta que hemos usado una distribución Red Hat de 64 bits, concretamente CentOS7 (en su versión 7.9.2009).

Para configurar TCP/IP en un nodo, lo que tenemos que hacer es acceder al archivo en la ruta `/etc/sysconfig/network-scripts/ifcfg-enp0s3` cabe destacar que `enp0s3` es el nombre de la interfaz de red de nuestro ejemplo, este puede ser cambiado en función de la interfaz. En este archivo de configuración podemos configurar la IP (pudendo ser estática, DHCP, etc), Gateway, etc. En este caso nuestro nodo servidor va a ser configurado como IP estática y tiene la siguiente configuración:

```
[root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
IPADDR=192.168.10.100
NETMASK=255.255.255.0
GATEWAY=192.168.10.100
NM_CONTROLLED=no
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="e61a0279-bfb2-470e-b7b6-7951188f32b3"
DEVICE="enp0s3"
ONBOOT="yes"
```

Una vez realizado esto, y como veremos en el siguiente punto, hemos configurado un servicio DHCP, por lo tanto la configuración de los nuevos nodos que se añadan a la red obtendrá su configuración TCP/IP gracias a este servicio.

Por último, en nuestro caso para aplicar los cambios hemos introducido el siguiente comando.

```
systemctl restart network
```

También podemos comprobar nuestra IP mediante el comando

```
ip a
```

```
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defa
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
    link/ether 08:00:27:6c:ff:79 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.100/24 brd 192.168.10.255 scope global noprefixroute enp0s
        valid_lft forever preferred_lft forever
    inet6 fe80::b782:4c5d:c150:ec02/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@localhost ~]#
```

## Asignación de una IP determinada según la dirección MAC

En este caso, primero vamos a explicar cómo hemos configurado el servicio DHCP, para ello hemos editado el archivo que se encuentra en la ruta **/etc/dhcp/dhcpd.conf**, en este archivo hemos configurado un pool de direcciones pertenecientes a nuestra red, en este caso la 192.168.10.0/24.

```
[root@localhost ~]# cat /etc/dhcp/dhcpd.conf
allow booting;
allow bootp;
allow unknown-clients;
ddns-update-style interim;
ignore client-updates;
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.120 192.168.10.200;
    option routers 192.168.10.100;
    filename "pxelinux.0";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    next-server 192.168.10.100;
}
```

Una vez realizado esto, podemos declarar host conocidos para asignarles una configuración en específico en función de su dirección MAC, como se muestra a continuación:

```
host C7 {
    hardware ethernet 08:00:27:E1:4F:E1;
    fixed-address 192.168.10.50;
    filename "pxelinux.0";
}
host C8 {
    hardware ethernet 08:00:27:70:FE:E1;
    fixed-address 192.168.10.75;
}
```

## Servir imagen de arranque a un nodo según la dirección MAC

En este caso, un paso previo para poder servir la imagen es tener la imagen montada en el directorio **/var/ftp/pub**, en este caso hemos creado dos directorios uno para cada imagen ISO montada.

```
[root@localhost ~]# ll /var/ftp/pub/
total 0
drwxr-xr-x. 9 root root 279 mar 11 17:43 centos7_9
drwxr-xr-x. 9 root root 279 mar 11 19:43 centos8
```

Una vez hecho esto, tenemos que añadir los archivos del directorio **/usr/share/syslinux** los archivos **chain.c32**, **mboot.c32**, **memdisk**, **menu.c32**, **pxelinux.0** al directorio en **/var/lib/tftpboot/**, además también debemos añadir los archivos **vmlinuz** y **initrd.img**, por una mayor organización estos archivos los hemos añadido dentro del directorio **networkboot**, ya por último creamos el directorio **pxelinux.cfg** (ES MUY IMPORTANTE QUE SEA CON ESTE NOMBRE), y dentro del mismo creamos un archivo del siguiente tipo según la MAC.

```

[root@localhost ~]# ll /var/lib/tftpboot/pxelinux.cfg/
total 8
-rw-r--r--. 1 root root 275 mar 12 12:14 01-08-00-27-e1-4f-e1
-rw-r--r--. 1 root root 457 mar 11 20:53 default
[root@localhost ~]# _

```

El archivo **01-08-00-27-e1-4f-e1** será el menú PXE que se arranque cuando la MAC **080027E14FE1** como se puede observar es necesario separar por “-” en lugar de “:” y que se debe añadir “01” antes de la dirección MAC.

El archivo de **01-08-00-27-e1-4f-e1** contiene lo siguiente:

```

GNU nano 2.3.1      Fichero: /var/lib/tftpboot/pxelinux.cfg/01-08-00-27-e1-4f-e1
default menu.c32
prompt 0
timeout 30
MENU TITLE CENTOSsPXE MENU
LABEL Centos7_x86
MENU LABEL Centos7_x64
KERNEL /networkboot/vmlinuz
APPEND initrd=/networkboot/initrd.img inst.repo=ftp://192.168.10.100/pub/centos8 ip=dhcp ks=ftp://1$
GNU nano 2.3.1      Fichero: /var/lib/tftpboot/pxelinux.cfg/01-08-00-27-e1-4f-e1
default menu.c32
prompt 0
timeout 30
MENU TITLE CENTOSsPXE MENU
LABEL Centos7_x86
MENU LABEL Centos7_x64
KERNEL /networkboot/vmlinuz
$tp://192.168.10.100/pub/centos8/anaconda-ks.cfg

```

En la opción **KERNEL** indicamos a la ruta al comprimido del kernel Linux desde el directorio **tftpboot**. Dentro del **APPEND** encontramos los siguientes elementos, el archivo **initrd.img** el cual es necesario para la instalación, ya que contiene un sistema de archivos que se va a utilizar en la instalación.

El enlace **inst.repo** indica donde se encuentra el sistema a instalar en la maquina cliente, en este caso le estamos indicando que instale los archivos de la ruta **/var/ftp/pub/centos8**, donde se encuentra montada la imagen del sistema operativo.

## Instalación desatendida según la dirección MAC

Para realizar una instalación desatendida, en el archivo del apartado anterior (**/var/lib/tftpboot/pxelinux.cfg/01-08-00-27-e1-4f-e1**), tenemos que añadir la opción **ks** (kickstart), indicándole la ruta al archivo, en este caso, se encuentra dentro de la ruta **/var/ftp/pub/centos8/anaconda-ks.cfg** (hemos añadido un archivo de este tipo en cada directorio con las ISOS montadas), este archivo de configuración indica como realizar el proceso de instalación desatendida, en nuestro caso tenemos lo siguiente.

```

GNU nano 2.3.1          Fichero: /var/ftp/pub/centos8/anaconda-ks.cfg
#version=DEVEL
# System authorization information
auth --enablesshadow --passalgo=sha512
# Use CDROM installation media
#cdrom
url --url="ftp://192.168.10.100/pub/centos8/"
# Use graphical install
graphical
# Run the Setup Agent on first boot
firstboot --enable

# Keyboard layouts
keyboard --vckeymap=es --xlayouts='es'
# System language
lang es_ES.UTF-8

# Network information
network --bootproto=dhcp --device=enp0s3 --ipv6=auto --activate
network --hostname=localhost.localdomain

# Root password
rootpw --iscrypted $6$0BqWe2nk0fSF18KT$05C.LG2q7U6qUNUWpK1BDUR7QyMU70y2DbeCNhATFdUCVb/1pz5nSWLVE0eN$
# System services
services --disabled="chronyd"
# System timezone
timezone Europe/Madrid --isUtc --nontp
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda
autopart --type=lvm
# Partition clearing information
clearpart --none --initlabel

```

```

GNU nano 2.3.1          Fichero: /var/ftp/pub/centos8/anaconda-ks.cfg

network --bootproto=dhcp --device=enp0s3 --ipv6=auto --activate
network --hostname=localhost.localdomain

# Root password
rootpw --iscrypted $6$0BqWe2nk0fSF18KT$05C.LG2q7U6qUNUWpK1BDUR7QyMU70y2DbeCNhATFdUCVb/1pz5nSWLVE0eN$
# System services
services --disabled="chronyd"
# System timezone
timezone Europe/Madrid --isUtc --nontp
# System bootloader configuration
bootloader --append=" crashkernel=auto" --location=mbr --boot-drive=sda
autopart --type=lvm
# Partition clearing information
clearpart --none --initlabel

%packages
@^minimal
@core
kexec-tools

%end

%addon com_redhat_kdump --enable --reserve-mb='auto'

%end

%anaconda
pwpolicy root --minlen=6 --minquality=1 --notstrict --nochanges --notempty
pwpolicy user --minlen=6 --minquality=1 --notstrict --nochanges --emptyok
pwpolicy luks --minlen=6 --minquality=1 --notstrict --nochanges --notempty
%end
%post

```

```
%post
useradd cliente
echo "12345" | passwd --stdin cliente
%end
reboot
```

Como podemos observar al final del archivo, se indica que se cree un usuario llamado cliente con la contraseña 12345. Además, uno de los aspectos relevantes de este archivo es el **%packages** ya que mediante este podremos añadir más paquetes a la maquina cliente que los básicos que se encuentren en la ISO en el caso de que fuesen necesarios.

## Añadir un nuevo nodo según la dirección MAC

Una vez realizado los dos pasos anteriores, para añadir un nodo a la red lo único que nos faltaría es indicarle como debe configurar TCP/IP para su funcionamiento, debido a que ya hemos montado un servicio DHCP, nos resulta fácil indicarle en el archivo **/var/lib/tftpboot/pxelinux.cfg/01-08-00-27-e1-4f-e1** que se inicie con DHCP con la opción **ip=dhcp** dentro del APPEND:

```
APPEND initrd=/networkboot/initrd.img inst.repo=ftp://192.168.10.100/pub/centos8 ip=dhcp ks=ftp://192.168.10.100/pub/centos8/anaconda-ks.cfg
```

## Referencias

- [1] [https://es.wikipedia.org/wiki/Entorno\\_de\\_ejecuci3n\\_de\\_prearranque](https://es.wikipedia.org/wiki/Entorno_de_ejecuci3n_de_prearranque)
- [2] <https://www.manageengine.com/products/os-deployer/pxe-preboot-execution-environment.html>
- [3] <https://www.minitool.com/es/respaldar-datos/inicio-pxe.html>
- [4] <https://www.redeszone.net/tutoriales/internet/protocolo-tftp-usos/>
- [5] <https://documentation.suse.com/es-es/sles/15-GA/html/SLES-all/cha-deployment-prep-pxe.html#:~:text=El%20hardware%20cliente%20debe%20admitir%20el%20arranque%20mediante,necesarios.%20PXE%20solo%20arranca%20un%20n3cleo%20e%20initrd.>
- [6] <https://learn.microsoft.com/es-es/windows/deployment/configure-a-pxe-server-to-load-windows-pe>
- [7] <https://www.ubackup.com/es/help/pxe-boot.html>
- [8] <https://www.codetd.com/es/article/11663216>
- [9] <http://www.lysesoft.com/products/andftp/>
- [10] <https://cyberduck.io>
- [11] <https://geekflare.com/es/tftp-servers-for-windows/>
- [12] <https://discourse.maas.io/t/about-maas-networks/5084>
- [13] [https://oa.upm.es/70543/1/TFG\\_PEDRO\\_FRANCISCO\\_REQUENA\\_DEL\\_AMO.pdf](https://oa.upm.es/70543/1/TFG_PEDRO_FRANCISCO_REQUENA_DEL_AMO.pdf)
- [14] <https://blog.ichasco.com/foreman-guia-definitiva/>
- [15] <https://www.theforeman.org/introduction.html>
- [16] <https://redessy.com/configure-el-servidor-de-instalacion-pxe-arranque-de-red-en-centos-7-x/>
- [17] <https://docs.bmc.com/docs/ServerAutomation/85/using/provisioning/managing-the-provisioning-environment/starting-and-stopping-a-pxe-server>
- [18] <https://docs.bmc.com/docs/ServerAutomation/85/using/provisioning/managing-the-provisioning-environment/starting-and-stopping-a-tftp-server>
- [19] <https://docs.bmc.com/docs/serverautomation/222/setting-up-the-pxe-server-for-high-availability-1094959561.html>