

Arranque en red mediante PXE

Jorge Manuel Montes Ot

I.E.S Medina Azahara

2º Administración de Sistemas Informáticos en Red

(Victor Zamora)

Índice

1.Definición del problema	3
1.1 Análisis empresarial	3
1.2 Objetivos empresariales.	3
1.3 Objetivos específicos.	3
2.Antecedentes	3
2.1 Serva	3
2.2 Windows WDS (Windows Deployment Services)	4
2.3 Tiny PXE Server	4
2.4 Mirantis	5
3.Propuesta de solución	5
3.1 Limitaciones	5
3.2 Alcance	6
4 Presupuesto	6
4.1 Recursos materiales	6
4.2 Recursos personales	7
5.Temporalización	7
6.Memoria Técnica	9
6.1 Resumen	9
6.2 Marco teórico	9
1. PXE:	9
2. DHCP (Dynamic Host Configuration Protocol):	9
3. TFTP (Trivial File Transfer Protocol):	10
4. NFS	10
6.3 Especificaciones de implementación	11
•Instalación del sistema operativo.	11
•Configuración de red.	11
•Creación de directorios:	12
•Instalación de paquetes necesarios para el arranque en red:	12
•Clonación del repositorio lpxe:	12
•Configuración del código IPXE:	13
•Configuración de servicios	13
7.Conclusiones	16
7.1 Debilidades	16
7.2 Amenazas	16
7.3 Fortalezas	16
7.4 Oportunidades	16
7.5 Ampliaciones futuras	17
7.6 Escalabilidad	17
8.Referencias	17

1. Definición del problema

1.1 Análisis empresarial

En entornos empresariales o instituciones con múltiples computadoras, es necesario instalar, actualizar y administrar los sistemas operativos de manera eficiente. Sin embargo, el proceso tradicional de instalación manual en cada computadora puede resultar tedioso, lento y propenso a errores. Esto puede llevar a una gestión ineficiente de los sistemas operativos y a una mayor carga de trabajo para los administradores de sistemas, suponiendo un gasto innecesario de recursos, tiempo y dinero.

1.2 Objetivos empresariales.

Como objetivo de este proyecto facilitaremos el proceso de instalación y administración de sistemas en una red. En lugar de tener que instalar y configurar manualmente los sistemas operativos de cada máquina individualmente, Además, facilitaremos la recuperación de sistemas ya que si un sistema operativo falla o se corrompe, el servidor de arranque en red puede ser usado para reiniciar la máquina a través de la red, eliminando la necesidad de medios físicos de arranque. Esto no solo facilita la recuperación de sistemas, sino que también puede reducir el tiempo de inactividad, proporcionando una mayor disponibilidad.

1.3 Objetivos específicos.

El objetivo de este proyecto será la configuración de un servidor de arranque en red PXE que nos permita cargar isos o imágenes tanto del propio servidor como de servidores web que nosotros configuremos debidamente, permitiéndonos realizar una instalación de cualquier iso o recuperación de imagen seleccionadas de manera sencilla en los equipos.

2. Antecedentes

Existen varios software que se utilizan para la configuración del arranque en red como son Serva, Windows WDS, Tiny PXE , Mirantis.

A continuación describiré cada una de estas soluciones y cuales son sus puntos fuertes y débiles.

2.1 Serva

Serva es un software para windows que hace de servidor de arranque en red pudiendo configurar desde su interfaz gráfica el dhcp, tftp y todos los protocolos necesarios para la

implementación de un servidor que nos permite gestionar el arranque en red de nuestros equipos.

Ventajas:

Fácil de configurar y usar.

Soporta una amplia variedad de protocolos de red.

Compatible con una gran cantidad de sistemas operativos.

Desventajas:

La versión gratuita tiene funcionalidades limitadas.

El soporte técnico puede ser limitado.

2.2 Windows WDS (Windows Deployment Services)

WDS es una tecnología de Microsoft que permite a los administradores de TI desplegar sistemas operativos de Windows a través de la red. WDS es especialmente útil en grandes entornos empresariales donde se requiere una instalación o actualización masiva de sistemas operativos.

Ventajas:

Integración con otras tecnologías de Microsoft.

Permite el despliegue de imágenes personalizadas.

Soporta tanto la instalación desatendida como la instalación basada en imágenes.

Desventajas:

Limitado a sistemas operativos Windows.

Requiere una comprensión avanzada de las tecnologías de red y de Microsoft para su configuración y mantenimiento.

2.3 Tiny PXE Server

Tiny PXE Server es una utilidad de software gratuita y de código abierto que permite a los usuarios arrancar sistemas operativos a través de una red. Aunque su funcionalidad es limitada en comparación con otros servicios de arranque en red, su tamaño pequeño y su facilidad de uso lo hacen útil para entornos más pequeños o para tareas específicas.

Ventajas:

Libre y de código abierto.

Pequeño y ligero, lo que lo hace ideal para entornos con recursos limitados.

Desventajas:

Funcionalidad limitada en comparación con otras soluciones.

Puede requerir conocimientos técnicos avanzados para su configuración y uso.

2.4 Mirantis

Mirantis es una plataforma de software que proporciona servicios de infraestructura en la nube. Uno de sus componentes es el Mirantis Secure Registry, que puede utilizarse para almacenar y distribuir imágenes de sistemas operativos a través de una red.

Ventajas:

Proporciona una solución integral para la gestión de la infraestructura en la nube.

Fuerte seguridad y control de acceso.

Desventajas:

Es una solución más costosa en comparación con otras alternativas.

Puede ser excesivo para entornos más pequeños o para organizaciones que sólo necesitan servicios de arranque en red.

3.Propuesta de solución

Después de la investigación de las distintas soluciones para la creación de un servidor de arranque me he decantado por la realización de este proyecto en ubuntu y configurar un servidor PXE con la configuración individual de cada uno de los protocolos implicados en el arranque en red. Esto supone una serie de ventajas y desventajas.

Como punto a favor para esta elección ha sido la flexibilidad que nos va a brindar el realizar el servidor de arranque en red de esta manera. Esto te permite adaptar la configuración a tus necesidades exactas, lo que puede ser particularmente útil en entornos con requerimientos específicos y dando una gran escalabilidad al proyecto.(Necesidades de red, almacenamiento, software, etc.).

En contra podemos encontrarnos un aumento en la complejidad, debido a que requiere una comprensión sólida de las redes, los sistemas operativos y los protocolos relevantes y mayor costo de mantenimiento ya que la configuración manual de un servidor puede requerir más mantenimiento que una solución "todo en uno" como hemos mencionado anteriormente.

3.1 Limitaciones

A la hora de realizar el proyecto he encontrado limitaciones sobre todo a la hora de la creación de las imágenes para luego recuperarlas en clonezilla ya que necesitamos una gran cantidad de almacenamiento libre.

También el tema de realizar las pruebas en un entorno virtualizado complica más las pruebas debido a muchos fallos que conlleva la virtualización nivel de red y con la bios legacy.

También se puede comentar la falta de documentación fiable ya que encuentras muchas fuentes pero o están muy desactualizadas o no dejan claro lo que se está realizando aportando simplemente código.

3.2 Alcance

Este proyecto se puede llevar a cabo en cualquier empresa o institución donde se requiera del formateo y configuración de equipos informáticos o el uso de diversos sistemas operativos periódicamente.

Algunos ejemplos del alcance de este proyecto pueden ser:

- Empresas de Tic en las que debido a la gran infraestructura que posee y a los múltiples equipos se necesitan de mantenimientos y despliegues software, con el servidor de arranque en red se podría centralizar estos asegurando una mayor eficacia para el equipo de mantenimiento.
- Instituciones educativas en las que se requiere gestionar y desplegar sistemas operativos y software en las computadoras de los laboratorios de informática.

Al final cualquier organización/institución que posea varias computadoras puede aprovechar las ventajas del servidor de arranque en red proporcionando mayor eficiencia en el mantenimiento y la restauración de los equipos.

4 Presupuesto

En este apartado abordaremos los distintos requisitos, tanto materiales como personales , para la elaboración del proyecto.

4.1 Recursos materiales

- Servidor: Dependiendo de las necesidades de la organización, pueden necesitar un servidor dedicado o utilizar un servidor existente. Un servidor dedicado de gama media puede costar entre 500 y 2000.

Aquí dejo una estimación de un posible servidor

Componentes	Especificación	Coste Aproximado
Procesador	Intel Xeon E-2224G	200
Memoria Ram	16 GB DDR4 ECC	100
Almacenamiento	2 x 1 TB SSD	120(60 cada uno)
Placa base	Compatible con CPU y RAM	200
Fuente de alimentación	Unidad de 500W	100
Caja de servidor	Un chasis adecuado	100
Tarjeta de Red		50
Sistema Operativo	Ubuntu Server	0
Total		870

- Red: Necesitarás un switch o un router que soporte el protocolo PXE y sea capaz de manejar el tráfico de red del arranque en red. Los costos pueden variar desde 100 para un dispositivo básico hasta 1000 para un equipo más avanzado.
- Software: Si decides usar Ubuntu y otras soluciones de código abierto, el software puede ser gratuito. Sin embargo, si requieres software comercial o licencias especiales, los costos pueden incrementarse.

En cuanto a recursos materiales el coste final es variable dependiendo de las necesidades requeridas de la organización.No es lo mismo querer implementarlo desde 0 y necesitar comprar un router, servidor, cableado etc, implementarse en un servidor existente El costo aproximado del material puede llegar a ser 950 euros.

4.2 Recursos personales

Consultor IT: Para la instalación y configuración inicial, podrías contratar a un consultor IT. Esto podría costar entre 500 y 2000 euros, dependiendo de la complejidad de la instalación y la tarifa del consultor.

Administrador de Sistemas: El mantenimiento continuo del servidor requerirá un administrador de sistemas. En función de las responsabilidades y del mercado local, podrías considerar un coste anual de entre 30.000 y 50.000 euros.

Técnico de Soporte: Para la resolución de problemas y la asistencia a los usuarios, podrías necesitar un técnico de soporte. De nuevo, los costos pueden variar, pero podrías considerar un coste anual de entre 20.000 y 30.000 euros.

Por lo tanto, los costos totales de los recursos humanos para el primer año podrían variar entre 50.500 y 82.000 euros, incluyendo la instalación y configuración inicial y el primer año de salarios para el administrador de sistemas y el técnico de soporte. En años posteriores, sin tener en cuenta posibles aumentos salariales y otros factores, podrías esperar pagar entre 50.000 y 80.000 euros por el mantenimiento del servidor.

5. Temporalización

El siguiente cronograma proporciona una estimación del tiempo necesario para instalar y configurar un servidor de arranque en red (PXE) en Ubuntu. Este proceso implica una serie de etapas, cada una de las cuales requiere su propio conjunto de tareas y su propio tiempo para completarse. Incluimos la instalación del sistema operativo, la configuración de la red, la configuración de los servicios DHCP y TFTP, la preparación de las imágenes de arranque en red, y finalmente, las pruebas y la resolución de problemas.

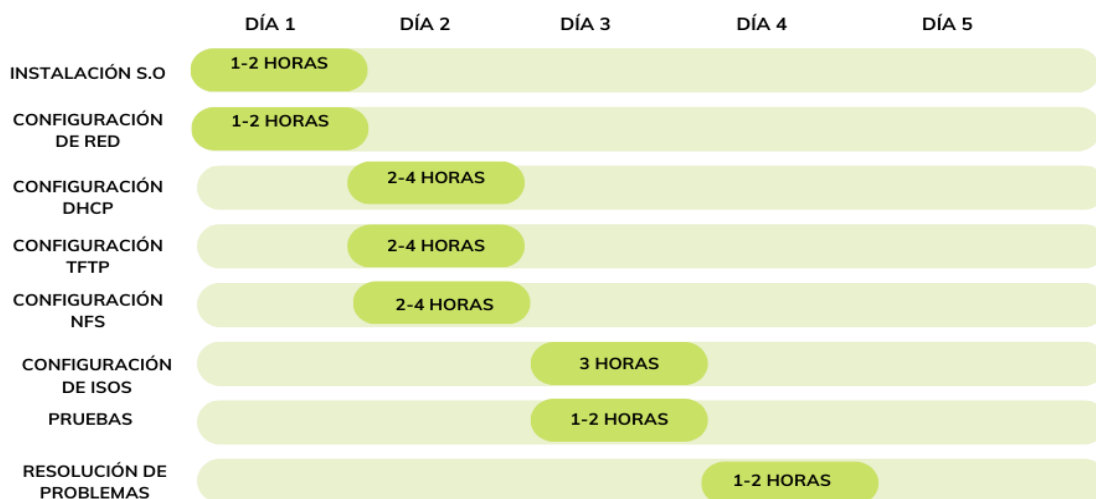
Proceso	Tiempo Aproximado
Instalación del Sistema Operativo (Ubuntu)	1-2 Horas
Configuración de la red (IP estática, etc.)	1-2 Horas
Instalación y configuración de DHCP	2-4 Horas
Instalación y configuración de TFTP	2-4 Horas
Instalación y configuración de NFS (si se usa para el almacenamiento en red)	2 Horas
Preparación de las imágenes de arranque en red	3 Horas
Pruebas y resolución de problemas	2-4 Horas
Total	13 - 21 Horas

Por lo tanto, si todo va bien, podrías esperar completar la instalación y configuración en 3-4 días de trabajo.(Todo depende de la complejidad de la red en la que se vaya a instalar o la cantidad de isos a cargar en el servidor)

En este diagrama ha supuesto una red básica en la que montaremos dos isos(ubuntu 22.04 y clonezilla live server)



Diagrama Gantt



6.Memoria Técnica

En este apartado vamos a proceder a dar toda la información necesaria para el desarrollo y realización del proyecto

6.1 Resumen

Este proyecto como se ha mencionado anteriormente se puede configurar en cualquier institución o empresa pero en este caso voy a tomar de referencia para el desarrollo de los procedimientos nuestra clase del instituto.Por tanto la red en la que trabajaremos es la 192.168.8.0/24

Instalaremos un ubuntu Server en el que configuraremos la red, el dhcp, tftp, nfs y los demás servicios necesarios para la configuración de pxe (Preboot Execution Environment)

6.2 Marco teórico

Definiciones de conceptos que debemos de tener en cuenta para la comprensión del proyecto:

1. PXE:

El Preboot eXecution Environment (PXE) es un estándar que permite a una computadora arrancar un sistema operativo a través de una red, sin la necesidad de tener almacenamiento local. Este estándar está implementado en el firmware de la tarjeta de red de la computadora o en la placa base, y permite que la máquina busque y cargue un sistema operativo desde un servidor en la red.

2. DHCP (Dynamic Host Configuration Protocol):

DHCP, que significa Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host), es un protocolo de red que permite a los servidores asignar automáticamente una dirección IP y otros detalles de la red a los dispositivos cada vez que se conectan a la red. En el contexto del PXE, el DHCP juega un papel muy importante ya que permite al dispositivo cliente obtener la dirección IP del servidor PXE. De esta manera, el cliente sabe a qué servidor contactar para obtener la imagen de arranque. Adicionalmente, la respuesta del servidor DHCP también proporciona opciones especiales PXE que dirigen al cliente a descargar la imagen de arranque utilizando TFTP.

El funcionamiento de DHCP se basa en un modelo cliente-servidor y utiliza un proceso de cuatro pasos conocido como DORA (Discover, Offer, Request, Acknowledge):

- Discover: Cuando un dispositivo se conecta a una red, envía un mensaje de "descubrimiento" DHCP a la red para buscar un servidor DHCP disponible.
- Offer: Los servidores DHCP que reciben este mensaje responden con un mensaje de "oferta" DHCP, que incluye una dirección IP que el servidor ha reservado para el dispositivo.
- Request: El dispositivo cliente selecciona una de las ofertas y envía un mensaje de "solicitud" DHCP al servidor correspondiente para solicitar la dirección IP ofrecida.
- Acknowledge: El servidor DHCP confirma la asignación de la dirección IP al dispositivo con un mensaje de "reconocimiento" DHCP. El dispositivo puede ahora usar esta dirección IP para comunicarse en la red.

3. TFTP (Trivial File Transfer Protocol):

Una vez que el cliente ha obtenido la dirección IP del servidor PXE mediante DHCP, necesita descargar la imagen de arranque del servidor. Para hacer esto, utiliza TFTP, que es un protocolo simplificado para la transferencia de archivos a través de la red. A diferencia de FTP, TFTP es menos complejo y no requiere autenticación, lo que lo hace ideal para la transferencia de imágenes de arranque en un entorno PXE.

TFTP utiliza el protocolo UDP para la transferencia de archivos, a diferencia de FTP que usa TCP. No tiene la capacidad de listar directorios o autenticar usuarios, simplemente transfiere archivos de un punto a otro. Su operación se realiza a través de dos tipos de paquetes: solicitudes de lectura/escritura y paquetes de datos.

Una transferencia de archivos TFTP comienza con una solicitud de lectura o escritura enviada por el cliente. Esta solicitud incluye el nombre del archivo a transferir y el modo de transferencia (normalmente "binario" o "texto"). Si la solicitud es aceptada, el servidor envía el archivo en bloques de 512 bytes, cada uno de los cuales debe ser reconocido por el cliente antes de que se envíe el siguiente

4. NFS

NFS, o Network File System, es un protocolo que permite a los usuarios de computadoras en una red acceder a archivos compartidos en esa red como si estuvieran en el disco duro de su propio equipo. Fue desarrollado por Sun Microsystems en la década de 1980 y se utiliza comúnmente en sistemas de archivos distribuidos en entornos Unix y Linux.

El servicio NFS se puede utilizar para alojar las imágenes de los sistemas operativos o los datos que se deben cargar en los equipos cliente durante el proceso de arranque en red. Por ejemplo, un servidor podría tener una imagen de un sistema operativo alojada en un directorio compartido de NFS. Luego, los equipos cliente podrían utilizar un gestor de arranque de red, como iPXE, para cargar esa imagen del sistema operativo desde el servidor NFS y arrancar el sistema operativo.

5. iPXE

iPXE es un gestor de arranque de red de código abierto. Puede ser utilizado para arrancar sistemas operativos o herramientas de diagnóstico a través de la red sin necesidad de tenerlos instalados en el almacenamiento local de una máquina.

Desde su repositorio de github podemos descargarnos su código y con unas pequeñas modificaciones podemos adaptarlo a nuestra red.

6.3 Especificaciones de implementación

Para la instalación y configuración del proyecto debemos de seguir una serie de procedimientos que describiremos a continuación:

La configuración del servidor de arranque se puede dividir en varias etapas;

•Instalación del sistema operativo.

En esta etapa procederemos a la instalación del ubuntu server sobre el que crearemos la infraestructura y configuraremos los servicios necesarios para el arranque en red.

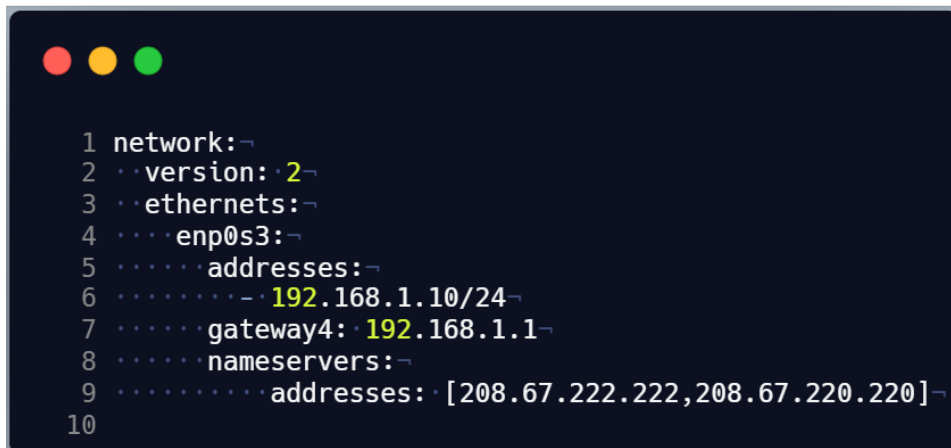
•Configuración de red.

Primero debemos asegurarnos de que en nuestra red permite el arranque por red, para ello procederemos a configurar el router, en caso necesario, ya que se puede dar el caso en el que este no esté configurado para permitir las conexiones por los puertos necesarios.(DHCP:67-68,TFTP:69,NFS:2049).Además, algunos router vienen con la opción BOOTP/DHCP desactivada lo que no permitiría este tipo de conexiones

Una vez comprobado esto procederemos a asignarle una ip estática a nuestro ubuntu server(Debemos de escoger una ip con la que sepamos que no vamos a tener problemas de solapamiento yo uso la 192.168.1.10)

Para dar una misma ip yo uso netplan el cual es una utilidad de redes de linux con la que mediante ficheros de tipo yaml podemos configurar la red en el arranque del sistema.

Nos vamos al directorio donde se almacenan dichos archivos de configuración `cd /etc/netplan/` y creamos un archivo .yaml (debemos de tener en cuenta que lee los archivos en orden alfabético, por tanto, si el archivo que creamos debe ser el primero en leerse . Recomiendo hacer un .bck del que viene por defecto y modificarlo)

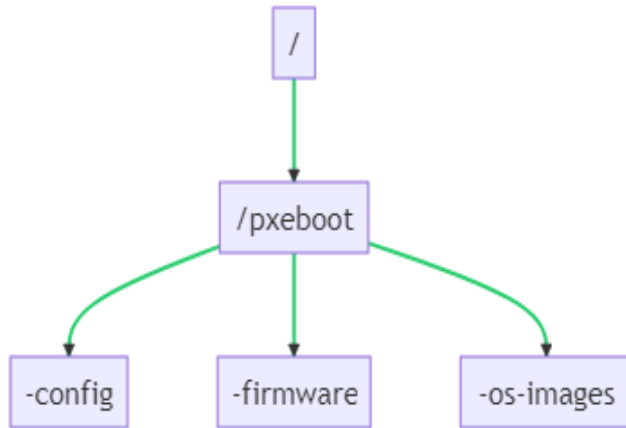


```
1 network:
2   version: 2
3   ethernets:
4     enp0s3:
5       addresses:
6         - 192.168.1.10/24
7       gateway4: 192.168.1.1
8       nameservers:
9         addresses: [208.67.222.222,208.67.220.220]
10
```

Como podemos ver en la imagen configuramos la interfaz de red `enp0s3` para obtener una dirección ip estática; 192.168.1.10 y le proporcionamos las direcciones DNS que debe de consultar.

•Creación de directorios:

Creamos la estructura de directorios que contendrán los distintos archivos de configuración del PXE y los archivos de arranque que serán utilizados por los distintos equipos para el arranque en red.



El diagrama muestra la estructura de directorios de "/pxeboot", que contiene tres subdirectorios:

1. "config": Este directorio contiene los archivos de configuración del IPXE.
2. "firmware": Este directorio contiene los archivos firmware del IPXE.
3. "os-images": En este directorio se deben crear subdirectorios con las distintas ISOS.

•Instalación de paquetes necesarios para el arranque en red:

Antes de seguir nos aseguraremos de que nuestro equipo tiene todos los repositorios actualizados mediante un ***sudo apt update*** e instalamos los siguientes paquetes *build-essential*, *liblzma-dev*, *isolinux*, *git*

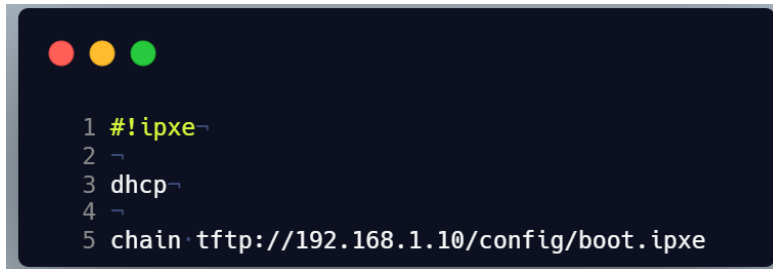
- **build-essential**: Contiene una serie de paquetes que incluyen utilidades para la compilación de código fuente y la herramienta make para automatizar esta compilación. Será necesario para la compilación del código del IPXE
- **Isolinux**: Es un cargador de arranque que contiene los archivos necesarios para que un sistema operativo pueda comenzar a funcionar
- **git**: El repositorio git se usará para la clonación del repositorio de IPXE con el que obtendremos los script necesarios para el arranque.

•Clonación del repositorio lpxe:

Nos posicionamos en el directorio donde realizaremos la clonación del repositorio (en mi caso descargas `cd ~/Descargas`) y ejecutando el siguiente comando podremos descargarnos el repositorio de github **`git clone https://github.com/ipxe/ipxe.git`**

•Configuración del código IPXE:

Una vez descargado el código del IPXE debemos de hacer unos pequeños cambios para que este se adapte a nuestra red para ello nos posicionamos en el directorio `cd ipxe/src` donde crearemos un archivo con extensión .ipxe (recomiendo `bootconfig.ipxe`)

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The terminal displays a configuration script for IPXE with line numbers 1 through 5. The script includes a comment, a DHCP configuration, and a chain command to load a boot file from a TFTP server.

```
1 #! ipxe
2
3 dhcp
4
5 chain tftp://192.168.1.10/config/boot.ipxe
```

Como se puede ver en la primera línea indicamos que el script debe de ser interpretado por IPXE y así el sistema puede interpretarlo correctamente.

La segunda línea indica que se use el servicio dhcp para la configuración de la red de los demás equipos al arrancar.

Por último con `chain tftp://ip_servidor/config/boot.ipxe` Indicamos la ruta del archivo que cogen los distintos equipos para el arranque en red.

Una vez tenemos el archivo debidamente configurado acorde a nuestras necesidades debemos de compilar el código usando `$ make bin/ipxe.pxe bin/undionly.kpxe bin/undionly.kkpxe bin-x86_64-efi/ipxe.efi EMBED=bootconfig.ipxe`

Con este comando construimos varias versiones de IPXE para obtener mayor compatibilidad con los distintos equipos con los que trabajaremos y con `EMBED=bootconfig.ipxe` indicamos que se debe insertar el script `bootconfig.ipxe` en las imágenes de iPXE que se están construyendo.

•Configuración de servicios

Usaremos dnsmasq que es una utilidad que nos proporciona servicios de infraestructura de red como DNS, DHCP y TFTP y así centralizamos la gestión de todos estos servicios simplificando la configuración de estos.

Para su instalación usaremos el comando `sudo apt install dnsmasq -y`

Primero antes de configurar los servicios realizaremos una copia de los archivos de configuración de este `$ sudo mv -v /etc/dnsmasq.conf /etc/dnsmasq.conf.backup`

lo abrimos y editamos con el siguiente contenido

```

1 interface=enp0s3 #nombre de interfaz de red
2 bind-interfaces
3 domain=linuxhint.local
4 dhcp-authoritative
5
6 dhcp-range=enp0s3,192.168.1.3,192.168.1.9,255.255.255.0,8h #rangos ip y concesion
7
8 dhcp-option=3,192.168.1.1
9
10 dhcp-option=option:dns-server,208.67.222.222 #dns
11
12 dhcp-option=option:dns-server,208.67.220.220
13
14 enable-tftp
15 tftp-root=/pxeboot
16 # boot config for BIOS systems
17
18 dhcp-match=set:bios-x86,option:client-arch,0
19
20 dhcp-boot=tag:bios-x86,firmware/ipxe.pxe
21
22 # boot config for UEFI systems
23
24 dhcp-match=set:efi-x86_64,option:client-arch,7
25
26 dhcp-match=set:efi-x86_64,option:client-arch,9
27
28 dhcp-boot=tag:efi-x86_64,firmware/ipxe.efi
29

```

Primero estableceremos cuál de las interfaces de red se encargará de dar servicio.

bind-interfaces: Este parámetro le indica a Dnsmasq que se una solo a la interfaz de red especificada anteriormente, en lugar de a todas las interfaces.

domain=linuxhint.local:

Aquí configuramos el nombre del dominio local para la resolución de DNS.

dhcp-range: define qué interfaz de red se encarga del dhcp, da un rango de direcciones ip que puede asignar a los equipos de la red, se establece la máscara de subred y por último se da el tiempo de concesión de la ip

dhcp-option=option:router,192.168.8.2: Este parámetro establece la opción de DHCP para el router.

dhcp-option=option:dns-server,208.67.222.222 indicamos los servidores dns que proporcionará por defecto el servidor dhcp(opendns)

enable-tftp y tftp-root=/pxeboot: Estos parámetros habilitan el servidor TFTP de Dnsmasq y establecen el directorio raíz TFTP

la opción **dhcp-match:** etiqueta a los equipos BIOS u EFI

dhcp-boot:según la etiqueta se elige la imagen de arranque que deben coger

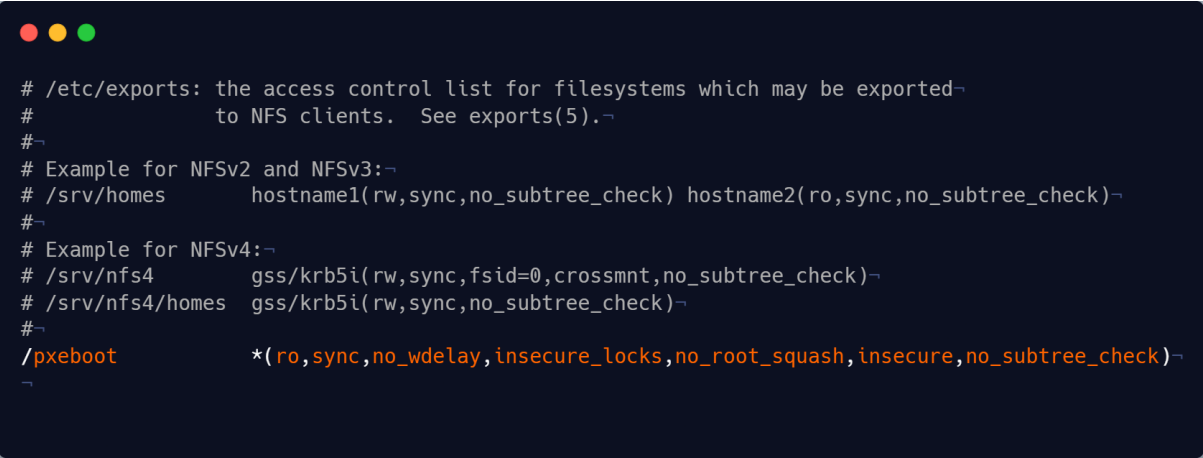
Con esta configuración debería ser suficiente para la configuración de los 3 servicios
Reiniciamos el servicio con `$ sudo systemctl restart dnsmasq`

Y procedemos a configurar NFS, un protocolo que nos permite compartir y acceder a archivos dentro de una misma red.

Para su instalación usamos el siguiente comando:

\$ sudo apt install nfs-kernel-server

Abrimos el archivo de configuración de nfs `sudo nano /etc/exports`
y añadimos al final del archivo la ruta de los archivos del pxe



```
# /etc/exports: the access control list for filesystems which may be exported-
#               to NFS clients.  See exports(5).-
#-
# Example for NFSv2 and NFSv3:-
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)-
#-
# Example for NFSv4:-
# /srv/nfs4       gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)-
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)-
#-
/pxeboot          *(ro,sync,no_wdelay,insecure_locks,no_root_squash,insecure,no_subtree_check)-
-
```

En la primera línea especificamos el directorio que queremos compartir y a continuación con el asterisco indicamos que pueden acceder todos los usuarios de la red, podríamos especificar nosotros una serie de host

- ro: Siglas de "solo lectura". Que indica que solo se tendrá permisos de lectura
- sync: esto significa que los cambios deben escribirse en el disco antes de que se complete la solicitud. Esto es más lento que asíncrono pero es más seguro en caso de un bloqueo.
- no_wdelay: esto significa que las solicitudes de escritura se procesan tan pronto como se reciben. Normalmente, las solicitudes de escritura pequeñas se retrasan durante un breve período de tiempo para ver si llega alguna otra solicitud de escritura que se pueda procesar en conjunto.
- insecure_locks: Esto permite bloquear solicitudes de puertos sin privilegios (puertos por encima de 1024).
- no_root_squash: esto permite que los clientes se conecten como root. Normalmente, por razones de seguridad, las solicitudes del usuario raíz se asignan al usuario de forma predeterminada.
- insecure: esto permite solicitudes de puertos no privilegiados.
- no_subtree_check: esto deshabilita la verificación de subárboles, lo que puede mejorar la confiabilidad en ciertas situaciones en las que los directorios se mueven o desmontan mientras se exportan.

Una vez configurado el archivo correctamente procedemos a aplicar los cambios realizados mediante el comando **sudo exportfs -av**

•Configuración de ISO:

Descargamos la iso que queramos configurar en nuestro servidor de arranque(en mi caso haré el ejemplo con la iso de ubuntu 22.04)

Primero nos posicionamos en el directorio donde queramos descargar la iso y descargamos la iso de su página oficial con el comando **wget** <https://releases.ubuntu.com/jammy/ubuntu-22.04.2-desktop-amd64.iso>.

La montamos en el directorio /mnt **sudo mount -o loop ~/Downloads/ubuntu-22.04-desktop-amd64.iso /mnt**

Creamos una subcarpeta en el directorio del pxe destinado para ello como mencionamos anteriormente y copiamos los archivos de la iso previamente montados **sudo rsync -avz /mnt/ /pxeboot/os-images/ubuntu-22.04-desktop-amd64/**

Por último configuramos el archivo de arranque del pxe en donde especificaremos la ruta de los archivos de arranque, para ello creamos el archivo de arranque en el directorio de configuración del pxe **sudo nano /pxeboot/config/boot.ipxe** e incertamos la siguiente configuración.

```
1 $ sudo nano /pxeboot/config/boot.ipxe
2
3 #!ipxe
4
5 set server_ip 192.168.8.10
6
7 set root_path /pxeboot
8
9 menu Select an OS to boot
10
11 item ubuntu-22.04-desktop-amd64 Install Ubuntu Desktop 22.04 LTS
12 item clonezilla-live-3.1.0-22-amd64 Run clonezilla live
13 choose --default exit --timeout 10000 option && goto ${option}
14
15
16 :ubuntu-22.04-desktop-amd64
17
18 set os_root os-images/ubuntu-22.04-desktop-amd64
19
20 kernel tftp://${server_ip}/${os_root}/casper/vmlinuz
21
22 initrd tftp://${server_ip}/${os_root}/casper/initrd
23
24 imgars vmlinuz initrd=initrd boot=casper maybe-ubiquity netboot=nfs ip=dhcp
   nfsroot=${server_ip}>
25 boot
26
27 :clonezilla-live-3.1.0-22-amd64
28 set os_root os-images/clonezilla-live-3.1.0-22-amd64
29 kernel tftp://${server_ip}/${os_root}/live/vmlinuz
30 initrd tftp://${server_ip}/${os_root}/live/initrd.img
31 imgars vmlinuz boot=live config noswap nolocales edd=on nomodeset ocs_live-run="ocs-live-
   genera>
32
33 boot || got
34 return go(f, seed, [])
35 }
```

En este archivo indicamos la dirección ip de nuestro servidor y establecemos la ruta raíz de nuestro servidor de arranque(en este caso /pxeboot)

En la sección menú Select an OS to boot... establecemos un menú que el usuario puede ver y seleccionar durante el arranque PXE. Hay dos opciones disponibles para el usuario, 'ubuntu-22.04-desktop-amd64' y 'clonezilla-live-3-1-0-22-amd64'.

Indicando la etiqueta ubuntu-22.04-desktop-amd64 indicamos la ruta de los archivos necesarios para el arranque en red.

vmlinuz: Este es el nombre que se le da al kernel de Linux comprimido. Es la pieza principal del sistema operativo y tiene la tarea de administrar los recursos de la computadora y permitir que los diferentes programas interactúen con el hardware del sistema. El archivo vmlinuz se carga en la memoria y se ejecuta al arrancar la computadora.

initrd: (que significa "disco RAM inicial"): Es una imagen temporal del sistema de archivos que se carga en la memoria durante el proceso de arranque, antes de que el sistema de archivos real esté montado.

7.Conclusiones

La implementación de un servidor de arranque en red (PXE) puede ser una herramienta valiosa en entornos empresariales e institucionales. Aumenta la eficiencia, centraliza la gestión del sistema operativo y permite un mantenimiento más fácil de las máquinas clientes.

Es una manera sencilla de mejorar el rendimiento y la eficiencia de cualquier organización o institución que posea equipos informáticos.

7.1 Debilidades

Se puede considerar la dependencia a la red ya que si la red cae, los sistemas no podrán arrancar. Además, si el servidor PXE se cae por alguna otra razón como quedarse sin luz o cualquier otro problema, ninguna máquina cliente podrá arrancar.

7.2 Amenazas

En caso de un ataque a la red, si alguien consiguiera comprometer la seguridad de la red y conectarse al servidor podría comprometer y afectar a los equipos conectados a este.

7.3 Fortalezas

La capacidad para administrar centralizadamente las imágenes del sistema operativo, lo que puede ahorrar tiempo y recursos. Además, la capacidad de arrancar equipos sin un disco duro local puede ser útil en una variedad de aplicaciones, como en diagnóstico y reparación de sistemas, entre otros.

7.4 Oportunidades

Un servidor PXE puede proporcionar una oportunidad para aprender más sobre la gestión de redes y la administración de sistemas. Además, puede ser una base para explorar tecnologías más avanzadas, como la virtualización y la computación en la nube.(Dockers y proxmox) u otros servicios como NFS, apache que implementar en el servidor y mejorarlo.

7.5 Ampliaciones futuras

Las futuras ampliaciones podrían incluir la incorporación de más servicios en el servidor, como NFS para proporcionar acceso a archivos o un servidor HTTP para ofrecer servicios web que puedan ser utilizados por el mismo servidor pxe. También podrías considerar la implementación de medidas de seguridad adicionales, como la autenticación y el cifrado.

7.6 Escalabilidad

El servidor es escalable en la medida en que el servidor pueda manejar la carga de red y la capacidad de almacenamiento necesaria para las imágenes de arranque. Para entornos más grandes, podrías considerar la implementación de múltiples servidores PXE para balancear la carga y proporcionar redundancia.

8.Referencias

1. UsuarioDebian. (22 DE SEPTIEMBRE DE 2019). Servidor PXE en Debian. Recuperado el 12 de junio de 2023, <https://usuariodebian.blogspot.com/2019/09/servidor-pxe-en-debian.html>
2. Alfonso Pastor –10 de octubre de 2014- Clonezilla en la red privada de un aula mediante arranque PXE. Recuperado el 12 de junio de 2023, de <https://2tazasdelinux.blogspot.com/search?q=pxe>
3. LinuxQuestions.org.-pxe-03-16-2008- boot problem no root-path_, de <https://www.linuxquestions.org/questions/linux-networking-3/pxe-boot-problem-no-root-path-628510/>
4. Ken Brobeck- PXE Server -Modificado: 07-Oct-2019 Por Jingchen Zhou <https://www.slac.stanford.edu/grp/cd/soft/unix/PXEServer.html>
5. Ian D. Foster- 18 de Abril de 2010 -Building a PXE Server, <https://lanrat.com/building-a-pxe-server/>
6. FourtyTwo-Jan 27, 2020-ZFS boot stuck at (initramfs), no root device specified,<https://forum.proxmox.com/threads/zfs-boot-stuck-at-initramfs-no-root-device-specified.64147/>
7. Installing Debian using network booting de <https://wiki.debian.org/PXEBootInstall>
8. Miguel, 04/02/2013, Cómo montar un servidor PXE <https://www.quintoblog.com/2013/02/04/como-montar-un-servidor-pxe/>

9. Shahriar Shovon ,2019,Configuring PXE Network Boot Server on Ubuntu 22.04 LTS
https://linuxhint.com/pxe_boot_ubuntu_server/
10. <https://ipxe.org/>