

Installation and initial Configuration

1. Install required Packages

```
sudo apt update -yy
sudo apt upgrade -yy
sudo apt install dhcpd isc-dhcp-server netplan.io tftpd-hpa nginx binutils
```

2. Configure DHCP Server

2.1 Configure DHCPd

Edit Configuration File `/etc/dhcp/dhcpd.conf`

```
# --- GLOBAL I-PXE DEFINITIONS ---
option space ipxe;
option ipxe-encap-opts code 175 = encapsulate ipxe;
option ipxe.priority code 1 = signed integer 8;
option ipxe.keep-san code 8 = unsigned integer 8;
option ipxe.no-pxedhcp code 176 = unsigned integer 8;

# Define Architecture Option
option architecture-type code 93 = unsigned integer 16;

# Standard Options
option domain-name "pxe.example.local";
option domain-name-servers 8.8.8.8;
default-lease-time 600;
max-lease-time 7200;
ddns-update-style none;
authoritative;
```

```
# --- SUBNET CONFIGURATION ---
subnet 172.16.128.0 netmask 255.255.255.0 {
    range 172.16.128.100 172.16.128.199;
    option routers 172.16.128.254;
    next-server 172.16.128.254;

    # If the client sends Option 175, it IS 100% iPXE.
    if exists ipxe-encap-opts {

        # SEND SCRIPT VIA HTTP
        filename "http://172.16.128.254/boot.ipxe";

    } else {

        # SEND BINARY VIA TFTP
        filename "undionly.kpxe";
        if option architecture-type = 00:07 { filename "ipxe.efi"; }
        if option architecture-type = 00:09 { filename "ipxe.efi"; }
        if option architecture-type = 00:06 { filename "ipxe.efi"; }
    }
}
```

2.2 Configure isc-dhcp-server

Edit Configuration File `/etc/default/isc-dhcp-server`

```
# Path to Configuration File from Step 2.1
DHCPD4_CONF=/etc/dhcp/dhcpd.conf

# Space-delimited List of Interfaces where the DHCP Server should listen on
# Example for multiple Interfaces: "eth0 eth1 wifi1"
INTERFACESv4="eth0"
```

3. Configure Netplan

Netplan is used to configure the local Interfaces on the Device which hosts the DHCP/iPXE Server

3.1 Disable Conflicting Services

The Default Network Configuration Services might interfere with Netplan

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl stop dhcpcd
sudo systemctl disable dhcpcd
```

3.2 Enable networkd

```
sudo systemctl enable systemd-networkd
sudo systemctl start systemd-networkd
```

3.3 Edit Configuration File `/etc/netplan/network.yaml`

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses:
        - 172.16.128.254/24
      nameservers:
        addresses:
          - 172.16.128.254
      ignore-carrier: true

  wifis:
    wlan0:
      dhcp4: true
      access-points:
        "your-wifi":
          password: "your-wifi-password"
```

With this File, the Interface `eth0` gets a static IP `172.16.128.254` and the Wifi interface `wlan0` is set to dhcp with the specified Wifi `your-wifi` and Password `your-wifi-password`

3.4 Update Permissions for `/etc/netplan/network.yaml`

```
sudo chmod 600 /etc/netplan/network.yaml
```

3.5 Apply Configuration

```
sudo netplan generate  
sudo netplan apply
```

3.6 Restart isc-dhcp-server

After configuring netplan, we need to restart isc-dhcp-server

```
sudo systemctl restart isc-dhcp-server  
sudo systemctl status isc-dhcp-server
```

4. Configure TFTP Server

The TFTP Server will serve the Bootloader

4.1 Ensure default Configuration File

Edit Configuration File `/etc/default/tftp-hpa`

```
TFTP_USERNAME="tftp"  
TFTP_DIRECTORY="/srv/tftp"  
TFTP_ADDRESS=":69"  
TFTP_OPTIONS="--secure --create"
```

4.2 Restart TFTP Server

```
sudo systemctl restart tftpd-hpa
```

5. Configure NGINX

NGINX will serve the actual OS Images, Drivers, Firmware and Configuration Files

5.1 Edit Configuration File

Open `/etc/nginx/sites-available/default`

```
server {  
  
    listen 80;  
    root /var/www/html;  
  
    server_name _;  
  
    location / {  
        autoindex on;  
        try_files $uri $uri/ =405;  
    }  
}
```

5.2 Restart NGINX

```
sudo systemctl restart nginx
```

6. Bootloader and Scripts

6.1 Download Bootloader(s)

```
cd /srv/tftp  
  
sudo wget http://boot.ipxe.org/undionly.kpxe  
sudo wget http://boot.ipxe.org/ipxe.efi
```

6.2 Setup Signed Bootloader Shim

We need this because otherwise Secure Boot will not allow booting from it

```
cd /srv/tftp  
  
sudo wget "http://ftp.debian.org/debian/pool/main/s/shim-signed/shim-signed_1.44~1+deb12u1+15.8-1~deb12u1_amd64.deb"  
  
sudo wget "http://ftp.debian.org/debian/pool/main/s/shim-helpers-amd64-signed/shim-helpers-amd64-signed_1+15.8+1~deb12u1_amd64.deb"
```

```
sudo ar x shim-signed*amd64.deb data.tar.xz
sudo tar -xf data.tar.xz ./usr/lib/shim/shimx64.efi.signed
sudo mv ./usr/lib/shim/shimx64.efi.signed ./shimx64.efi

sudo ar x shim-helpers*amd64.deb data.tar.xz
sudo tar -xf data.tar.xz ./usr/lib/shim/mmx64.efi.signed
sudo mv ./usr/lib/shim/mmx64.efi.signed ./mmx64.efi

sudo rm *.deb *.tar.xz data.tar.xz
sudo rm -r ./usr
sudo mv ipxe.efi grubx64.efi
```

6.3 Create Boot Script

Edit/Create File `/var/www/html/boot.ipxe`

```
#!ipxe

# --- Global Settings ---
set timeout 3600000
set menu-default shell # Default to shell

# --- The Menu ---
:start
menu MueTec iPXE Server
item --gap -- -----
item shell      Drop to iPXE Shell
item reboot     Reboot Computer
item --gap -- -----
item netinfo    Show Network Info
choose --timeout ${timeout} --default ${menu-default} selected || goto shell

# --- Actions ---
goto ${selected}

:shell
echo Dropping to shell...
shell
goto start
```

:reboot

reboot

:netinfo

ifstat

echo Press any key to return...

prompt

goto start

Revision #25

Created 3 December 2025 13:34:32 by Oliver Karger

Updated 4 December 2025 14:52:31 by Oliver Karger