

DEBIX: Network boot instructions

Step 1. NFS server settings (virtual machine: username wei)

1. Install NFS services: `sudo apt install nfs-kernel-server rpcbind`
2. Create an nfs service folder : `udo mkdir /home/wei/nfs`

```
wei@wei:~$ sudo mkdir nfs
[sudo] password for wei:
wei@wei:~$ ls
Desktop  Downloads  Music  Pictures  poky-master.zip  snap  Templates
Documents  linux      nfs     poky      Public          STM32  Videos
```

3. Configure NFS server: `sudo vi /etc/exports`
Add this line at the end: `/home/wei/nfs *(rw,sync,no_root_squash)`

```
# /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)
#
/home/wei/nfs/rootfs *(rw,sync,no_root_squash)
```

4. restart NFS server: `sudo /etc/init.d/nfs-kernel-server restart`

```
wei@wei:~$ sudo /etc/init.d/nfs-kernel-server restart
Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
wei@wei:~$
```

Step 2. Store the debix system kernel image in the nfs directory (Image) 、 device tree (.dtb) and the unzipped root filesystem.

```
wei@host:~/nfs$ ls
bin  dev  home  imx8mp-evk.dtb  log_lock.pid  main.c  mnt  proc  run  snap  swapfile  tmp  usr
boot  etc  Image  lib            lost+found    media  opt  root  sbin  srv  sys      unit_tests  var
```

Step 3. Enter the uboot of the Debix system and configure the uboot environment variables

Configure bootargs:

```
Setenv bootargs 'console=ttyMXC1,115200 root=/dev/nfs rw nfsroot=192.168.1.13:/home/wei/debix/nfs/rootfs,proto=tcp ip=192.168.1.138:192.168.1.13:192.168.1.1:255.255.255.0::ens33:off'
```

Configure bootcmd (This setting allows Debian to boot from the network)

```
setenv bootcmd 'nfs 40480000 192.168.1.13:/home/wei/debix/nfs/rootfs/Image;
nfs 43000000 192.168.1.13:/home/wei/debix/nfs/rootfs/imx8mp-evk.dtb; booti
40480000 - 43000000'
```

Execute the boot command, the board will load the Image and dtb files from the network.

```
File transfer via NFS from server 192.168.1.13; our IP address is 192.168.1.138
Filename '/home/wei/debix/nfs/rootfs/Image'.
Load address: 0x40480000
Loading: #####
#####
#####
```

```
File transfer via NFS from server 192.168.1.13; our IP address is 192.168.1.138
Filename '/home/wei/debix/nfs/rootfs/imx8mp-evk.dtb'.
Load address: 0x43000000
Loading: #####
done
Bytes transferred = 61298 (ef72 hex)
Moving Image from 0x40480000 to 0x40600000, end=42440000
## Flattened Device Tree blob at 43000000
   Booting using the fdt blob at 0x43000000
   Using Device Tree in place at 0000000043000000, end 0000000043011f71
```

```

Starting Time & Date Service...
[ OK ] Started Disk Manager.
[ OK ] Started Time & Date Service.
[ OK ] Finished Wait until snapd is fully seeded.
[ OK ] Started Dispatcher daemon for systemd-networkd.

Ubuntu 20.04.3 LTS imx8mpevk ttyxc1

imx8mpevk login: debix
Password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.10.72 aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Wed Jul 27 09:29:49 UTC 2022 on ttyxc1
debix@imx8mpevk:~$ ls
Desktop  Downloads  Pictures  Templates  gstshark_2022-07-27_09:28:05
Documents Music      Public    Videos
debix@imx8mpevk:~$ █

```

Here, Debix has successfully boot up from the network

Alternative method: It is also possible to use TFTP to make Debix load the kernel image and device tree from the network:

1. Enter uboot to set network related environment variables:

environment variable	Description
ipaddr	Dev board IP address, can be obtained with dhcp command from the router
ethaddr	Dev board mac address, must be configured
gatewayip	IP of gateway
netmask	subnet mask
serverip	Server IP address, the IP of Ubuntu host

```

setenv ipaddr 192.168.1.111
setenv ethaddr b8:ae:1d:01:00:00
setenv gatewayip 192.168.1.1
setenv netmask 255.255.255.0
setenv serverip 192.168.1.13
saveenv

```

2. Configure bootargs environment variables (The root filesystem of the SD card is still used here, only the kernel image and device tree are obtained from the network):

```
setenv bootargs 'console=ttyxc1,115200 root=/dev/mmcbk1p2 rootwait rw'
```

3. Set the bootcmd environment variable:

```
setenv bootcmd 'tftp 40480000 Image; tftp 43000000 imx8mp-evk.dtb; booti 40480000 - 43000000'
```

4. Execute the boot command, the system will obtain files from the virtual machine and load the system.