

[UEFI Practice] HttpBoot

UEFI Development ...

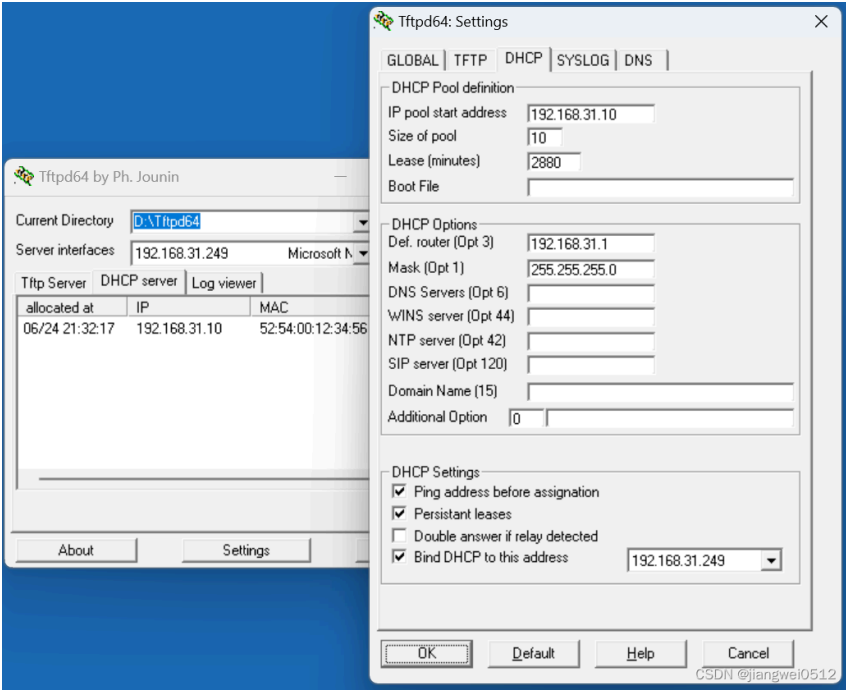
This column includes this content

136 articles

Subscribe to our column

Environment Configuration

1. First, download the tftpd tool, which can be downloaded from phjounin/tftpd64/Downloads — Bitbucket . It is recommended not to install it on the C drive because you may not be able to modify its configuration.
2. Configure the DHCP service of the tftpd tool:



Note that the IP address here needs to match the actual network card IP.

3. Download Apache from Apache VS17 binaries and modules download (apachelounge.com). As above, it is recommended not to install it on the C drive.
4. To configure Apache, you need to modify the conf\httpd.conf file in the installation directory. The modification points are as follows:

bash

1 Define SRVROOT "F:\httpd\Apache24"

2 ServerName www.example.com:80

AI generated projects

登录复制

5. After the configuration is successful, run the following command to check whether the configuration is successful:

bash

1 D:\Apache24>bin\httpd.exe -t

2 Syntax OK

3

4 D:\Apache24>

AI generated projects

登录复制

A return value of " Syntax OK " indicates that the configuration is successful.

6. Further install Apache, the corresponding command:

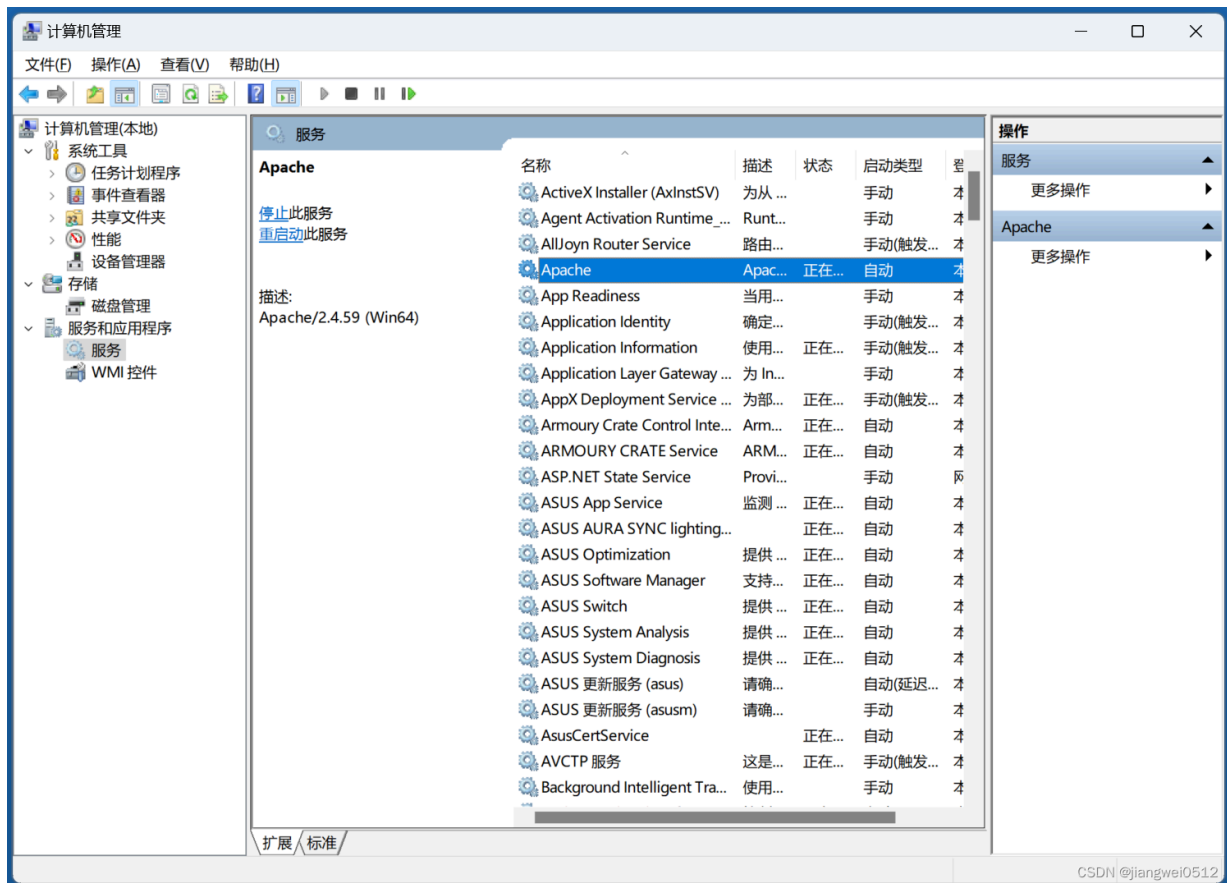
bash

1 bin\httpd.exe -k install -n Apache

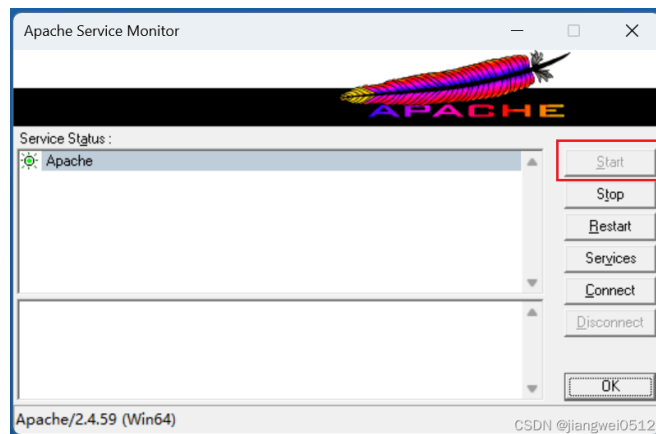
AI generated projects

登录复制

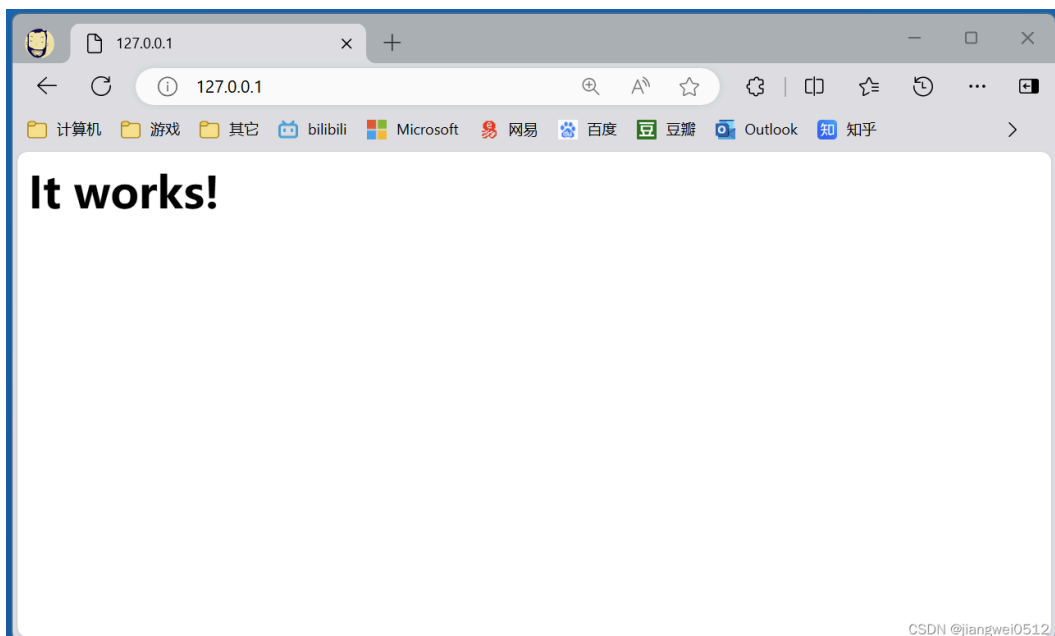
7. After that, you can see Apache in Windows Services:



8. To use the Apache server later, use bin/ApacheMonitor.exe in the installation directory. It is a graphical interface. Open it as follows (you can find it in the taskbar icon. The following figure shows that the Apache server has been opened):



After opening, you can access it through the browser:



If " **It works!** " is displayed, it means that the Apache service is working properly. Note that the address here is 127.0.0.1, because it is accessed locally. If it is **accessed remotely** , the actual IP address of the device where Apache is located can be used, for example:

22:13 



It works!

大小

192.168.31.249

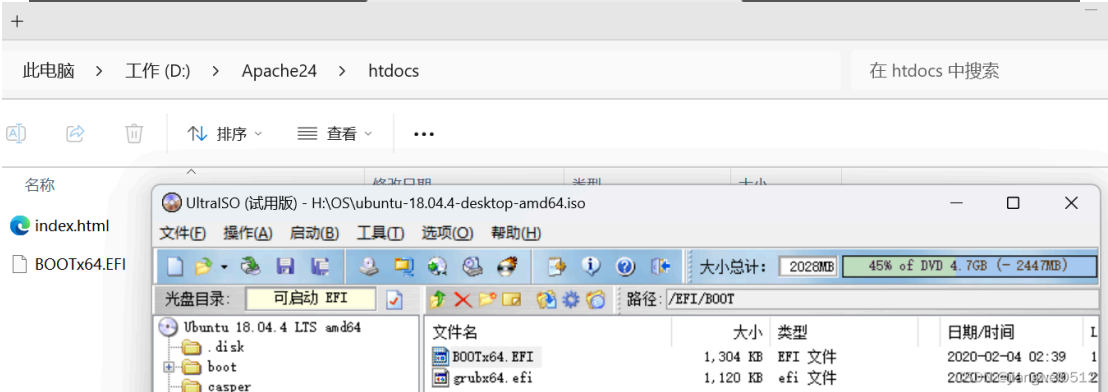


At this point, the HttpBoot environment is ready.

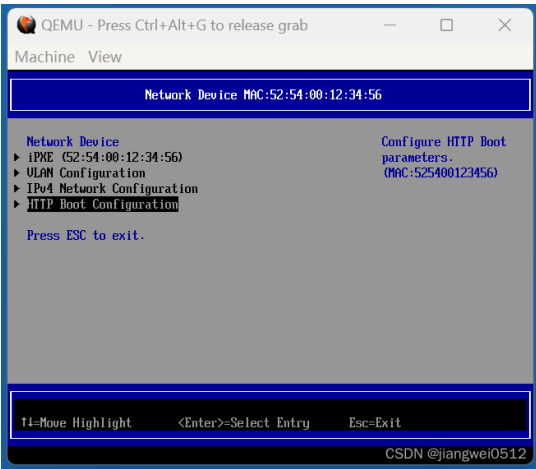
The main function of the above tools is to provide DHCP and HTTP services. The former is used to assign IP addresses and the latter is used to download the Boot loader.

use

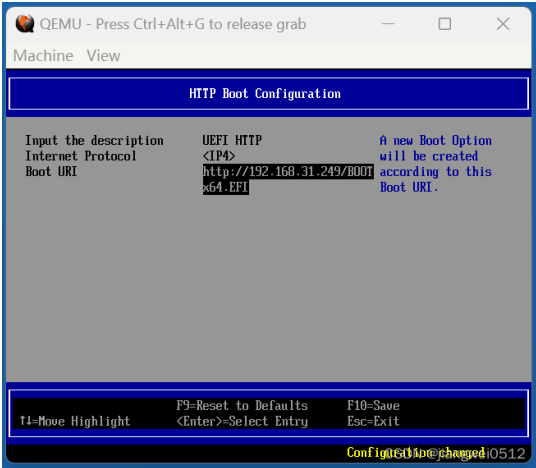
1. First, get the BootLoader. Here, get a bootx64.efi from the Ubuntu installation package and put it in the Apache specific directory:



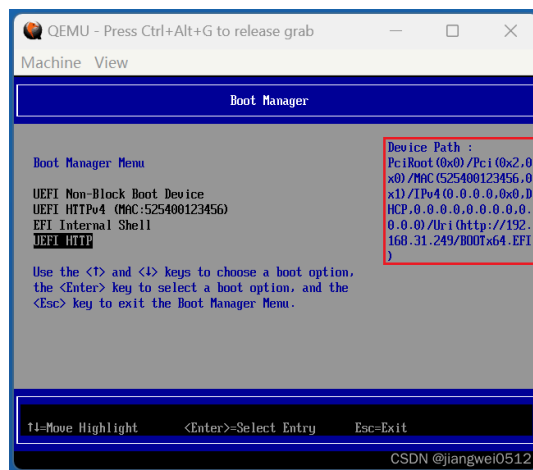
2. To compile a BIOS that supports the HttpBoot function, you can download the source code from edk2-beni: Used to learn and verify UEFI BIOS. (gitee.com) NETWORK_HTTP_BOOT_ENABLE and open the compilation macro to enable HttpBoot.
3. Start BIOS, enter Setup, select "Device Manager" -> "Network Device List", and select the network card that needs to be started by HttpBoot:



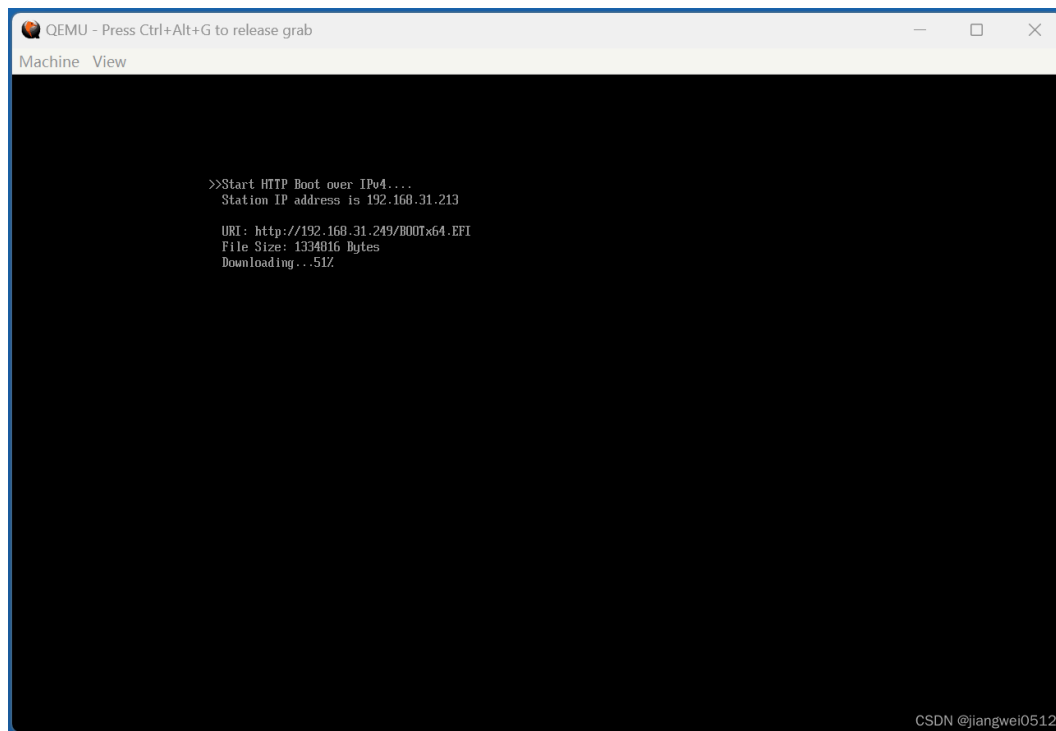
Select HTTP Boot Configuration and enter the URI:



After that, save the configuration and return to the BootManager interface, where you can see the HttpBoot just configured:



To enable this configuration item, execute the following:



You can see that bootx64.efi is downloaded and eventually executed, which means that HttpBoot has been successfully started. However, the subsequent operation still depends on the configuration of bootx64.efi, which is not the focus here.