

## 配置网络

< Handbook/AMD64 | Installation

本页更新日期:Handbook/AMD64/Installation/Networking的翻译版本。 翻译工作已完成100%。

**其他语言:** Deutsch English Türkçe español français italiano polski português do Brasil čeština pyccкий gàidh 中文 (中国大陆) 日本語 한국어

<b>目录 (隐藏)</b>	<b>AMD64 手册</b>
<div> <div>1 自动网络检测配置</div> <div>1.1 使用DHCP</div> <div>1.2 测试网络</div> </div> <div> <div>2 获取接口信息</div> </div> <div> <div>3 Optional: Application specific configuration</div> <div>3.1 可选: 配置网络代理</div> <div>3.2 对 ADSL 使用 pppoe-setup</div> <div>3.3 使用 PPTP</div> <div>3.4 配置 WEP</div> </div> <div> <div>4 使用 net-setup</div> </div> <div> <div>5 互联网和 IP 基础</div> <div>5.1 接口和地址</div> <div>5.2 网络和 CIDR</div> <div>5.3 The Internet</div> <div>5.4 域名系统</div> </div> <div> <div>6 手动配置网络</div> <div>6.1 配置接口地址</div> <div>6.2 配置默认路由</div> <div>6.3 配置 DNS</div> </div>	<div> <div>安装</div> <div>关于安装</div> <div>选择安装媒介</div> <div>配置网络</div> <div>准备磁盘</div> <div>安装 stage3</div> <div>安装基础系统</div> <div>配置内核</div> <div>配置系统</div> <div>安装系统工具</div> <div>配置网络程序</div> <div>安装恢复</div> </div> <div> <div>使用 Gentoo</div> <div>Portage 介绍</div> <div>USE 标记</div> <div>Portage 功能特性</div> <div>inotify 系统</div> <div>环境变量</div> </div> <div> <div>使用 Portage</div> <div>文件和目录</div> <div>混合使用不同的软件分支</div> <div>额外的工具</div> <div>自定义软件包仓库</div> <div>高级特性</div> </div> <div> <div>配置网络</div> <div>网络</div> <div>高级配置</div> <div>模块化网络</div> <div>无线网络</div> <div>添加功能</div> <div>动态管理</div> </div>

### 自动网络检测配置

它能够自动检测到么？

如果系统接入到一个有 DHCP 服务器的以太网网络，那么很可能自动配置好网络。这样的话，安装媒介所包含的很多网络命令，比如 `ssh`, `scp`, `ping`, `ircs` 以及其他的，以及其他的一些命令，都可以立即工作。

### 使用DHCP

DHCP (动态主机配置协议) 使自动接受网络信息 (IP地址、掩码、广播地址、网关、名称服务器等) 变得容易。这只有在网络中有DHCP服务器 (或者如果ISP提供商提供一个DHCP服务) 时有用。要使一个网络接口自动接受信息，使用 `dhcpcd`：

DHCP requires that a server be running on the same *Layer 2 (Ethernet)* segment as the client requesting a *lease*. DHCP is often used on RFC1918 (*private*) networks, but is also used to acquire public IP information from ISPs.

**提示**

Official Gentoo *boot media* runs `dhcpcd` automatically at startup. This behavior can be disabled by adding the `nodhcp` argument to the *boot media* kernel commandline.

If it is not already running, `dhcpcd` can be started on `emip10` with:

```
root # dhcpcd emip10
```

一些网络管理员要求你使用DHCP服务器所提供的的主机名和域名。 这种情况下请用：

```
root # dhcpcd -HD emip10
```

如果要停止 `dhcpcd`，可以使用 `-x`：

```
root # dhcpcd -x
sending signal Term to pid 10831
waiting for pid 10831 to exit
```

**另请参阅**

使用 Dhcpcd

### 测试网络

A properly configured *default* route is a critical component of Internet connectivity, route configuration can be checked with:

```
root # ip route
default via 192.168.0.1 dev emip10
```

If no *default* route is defined, Internet connectivity is unavailable, and additional configuration is required.

Basic internet connectivity can be confirmed with a ping:

```
root # ping -c 3 1.1.1.1
```

**提示**

It's helpful to start by pinging a known IP address instead of a hostname. This can isolate DNS issues from basic internet connectivity issues.

Outbound HTTPS access and DNS resolution can be confirmed with:

```
root # curl --location gentoo.org --output /dev/null
```

如果这些都不工作，则本章节中其余的部分可跳读，直接跳到安装介绍的下一步骤 ([准备磁盘](#))。

If `curl` reports an error, but Internet-bound pings work, DNS may need configuration.

If Internet connectivity has not been established, first *interface information* should be verified, then:

- `net-setup` can be used to assist in network configuration.
- Application specific configuration may be required.
- Manual network configuration can be attempted.

### 获取接口信息

If networking doesn't work out of the box, additional steps must be taken to enable Internet connectivity. Generally, the first step is to enumerate host network interfaces. `ip` 命令，属于 `sys-apps/iproute2` 软件包，他可以用来查询和配置系统网络。

**link** 参数可以用来显示网络接口的信息：

```
root # ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4: emip10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:f2:ac:25:7a brd ff:ff:ff:ff:ff:ff
```

**address** 参数可以查询设备地址信息：

```
root # ip address
2: emip10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:f2:ac:25:7a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global emip10
        valid_lft forever preferred_lft forever
    inet6 fe80::e040:f2ff:feac:257a/64 scope link
        valid_lft forever preferred_lft forever
```

The output of this command contains information for each network interface on the system. Entries begin with the device index, followed by the device name: `emip10`.

**提示**

如果使用标准的 `ifconfig` 命令没有显示出口口，尝试使用带有 `-a` 选项的相同的命令。这个选项通过这个工具去显示系统检测到的所有的网络接口，不管他们是 `up`或`down`状态。如果 `ifconfig -a` 没有提供结果，则硬件有错误或者接口驱动没有加载到内核中。这些情况都通过本手册的[故障排除](#)需求支持。

为了保持一致性，本手册中假说主要的网络接口为 `emip10`。

因为名称变成了 `emip10`，所以系统的接口名称可能和旧的 `eth0` 命名规则不一样。近期的安装媒介上常规网络接口名字可能显示为 `em0`、`em1` 或 `emip50`。查看 `ifconfig` 输出中找到有你本地网络相关的IP地址的接口。

## Optional: Application specific configuration

The following methods are not generally required, but may be helpful in situations where additional configuration is required for internet connectivity.

### 可选: 配置网络代理

如果需要通过代理来访问互联网，那么必须为 Portage 定义代理信息。这样才能正确访问每个支持协议的代理。Portage 使用 `http_proxy`, `ftp_proxy` 和 `RSYNC_PROXY` 环境变量，通过 `wget` 和 `rsync` 的检索机制来下载软件包。

Certain text-mode web browsers such as **Links** can also make use of environment variables that define web proxy settings; in particular for the HTTPS access it also will require the `https_proxy` environment variable to be defined. While Portage will be influenced without passing extra run time parameters during invocation, **links** will require proxy settings to be set.

大多数情况下，只要将这个环境变量定义为代理服务主机名。在下面的例子中，假定代理服务器的主机是 `proxy.gentoo.org` 并且端口为 `8080`。

**附注**

下列命令中的 # 符号是注释，添加 # 符号只是为了清楚起见，在输入命令时不需要输入 # 符号。

定义一个 HTTP 代理 (用于 HTTP 和 HTTPS 流量)：

```
root # export http_proxy="http://proxy.gentoo.org:8080" # 同时应用到 Portage 和 Links
root # export https_proxy="http://proxy.gentoo.org:8080" # 只应用到 Links
```

如果代理需要验证，使用下面的语法来设置用户名和密码：

```
root # export http_proxy="http://username:password@proxy.gentoo.org:8080" # 同时应用到 Portage 和 Links
root # export https_proxy="http://username:password@proxy.gentoo.org:8080" # 只应用到 Links
```

使用下列参数启动可以为 **Links** 使用代理：

```
links -s links -http-proxy $http_proxy -https-proxy $https_proxy
```

为 Portage 和 **links** 定义 FTP 代理：

```
root # export ftp_proxy="ftp://proxy.gentoo.org:8080" # 同时应用到 Portage 和 Links
```

使用下列参数启动可以为 **Links** 使用 FTP 代理：

```
links -s links -ftp-proxy $ftp_proxy
```

为 Portage 定义 RSYNC 代理：

```
root # export RSYNC_PROXY="proxy.gentoo.org:8080" # 应用到 Portage; Links 不支持 rsync 代理
```

### 对ADSL使用pppoe-setup

If PPPoE is required for Internet access, the Gentoo *boot media* includes the `pppoe-setup` script to simplify **ppp** configuration.

安装时，`pppoe-setup` 会询问：

- 连接到 ADSL 调制解调器的以太网 接口名称。
- PPPoE 用户名和密码。
- DNS 服务器 IP。
- 是否需要拨号。

```
root # pppoe-setup
root # pppoe-start
```

In the event of failure, credentials in `/etc/ppp/pap-secrets` or `/etc/ppp/chap-secrets` should be verified. If credentials are correct, PPPoE Ethernet interface selection should be checked.

### 使用PPTP

如果需要 PPTP 支持，可以使用 `pptpClient`，但是需要在使用前提前配置。

编辑 `/etc/ppp/pap-secrets` 或者 `/etc/ppp/chap-secrets`，让他们包含正确的用户名和密码：

```
root # nano /etc/ppp/chap-secrets
```

如果需要，编辑 `/etc/ppp/options.pptp`：

```
root # nano /etc/ppp/options.pptp
```

配置完成后，运行 `pptp` (带有一些 `options.pptp`无法设置的选项) 来连接到服务器：

```
root # pptp -server ipv4 address
```

### 配置 WEP

**警告**

Do not use WEP unless it is the only option. WEP provides essentially no security over an open network.

**附注**

可能只有特定的架构支持 `wi` 命令。如果这个命令不可用，检查 `net-wireless/iw` 包是否可用于当前架构。除非安装 `net-wireless/iw` 包，否则 `w` 命令将一直不可用。

当使用一块无线 (802.11) 网卡，在继续之前需要先配置无线设置。要查看当前无线网卡的设置，你可以使用 `w`，运行 `w` 可能会显示如下：

```
root # iw dev wlan9s0 info
Interface wlan9s0
    txpower 3
    wdev 0x1
    addr 00:00:00:00:00:00
    type managed
    wiphy 0
    channel 11 (2462 MHz), width: 20 MHz (no HT), center1: 2462 MHz
    txpower 30.00 dBm
```

检查当前连接：

```
root # iw dev wlan9s0 link
Not connected.
```

或

```
root # iw dev wlan9s0 link
Connected to 00:00:00:00:00:00 (on wlan9s0)
SSID: GentooNode
freq: 2462
RX: 3279 bytes (25 packets)
TX: 1049 bytes (7 packets)
signal: -23 dBm
tx bitrate: 1.0 MBit/s
```

**附注**

一些无线网卡的设备名可能是 `wlan0`或`ra0`而不是 `wlan9s0`，运行 `ip link` 可以识别正确的设备名称。

对于大多数用户，只需要两个设置来连接，即ESSID (也称无线网络名称) 和可选的WEP密钥。

- 首先，确保接口处于活动状态：

**命令**

ip link set dev wlan9s0 up

- 连接到名为"GentooNode"的开放网络：

**命令**

iw dev wlan9s0 connect -w GentooNode

- 设置一个WEP密钥：使用 `d` 前缀：

**命令**

iw dev wlan9s0 connect -w GentooNode key 0:d:123412341234abcd

- 使用ASCII WEP密钥连接：

**命令**

iw dev wlan9s0 connect -w GentooNode key 0:some-password

**附注**

如果无线网络配置为WPA或WPA2，则需要使用**wpa\_supplicant**。关于为Gentoo Linux配置无线网络的更多信息，请阅读[Gentoo手册](#)中的无线网络章节。

使用 `w dev wlan9s0 link` 确认无线设置。如果无线已经工作，继续按下一章节 ([了解网络层](#)) 配置IP级别的网络选项或者使用前面描述的**net-setup**工具。

## 使用 net-setup

In cases where automatic network configuration is unsuccessful, the Gentoo *boot media* provides scripts to aid in network configuration. **net-setup** can be used to configure wireless network information and static IPs.

```
root # net-setup emip10
```

**提示**

`net-setup`将会询问关于网络环境的一些问题。当所有这些完成后，网络连接就应该工作，以前面的方式测试网络连接。如果测试通过，恭喜！跳过本章节剩余部分并继续[准备磁盘](#)。

**重要**

Network status should be tested after any configuration steps are taken. In the event that configuration scripts do not work, manual network configuration is required.

## 互联网和 IP 基础

If all of the above fails, the network must be configured manually. This is not particularly difficult, but should be done with consideration. This section serves to clarify terminology and introduce users to basic networking concepts pertaining to manually configuring an internet connection.

**提示**

Some CPE (Carrier Provided Equipment) combines the functions of a *router*, *access point*, *modem*, *DHCP server*, and *DNS server* into one unit. It's important to differentiate the functions of a device from the physical appliance.

### 接口和地址

Network *interfaces* are logical representations of network devices. An *interface* needs an *address* to communicate with other devices on the *network*. While only a single *address* is required, multiple addresses can be assigned to a single *interface*. This is especially useful for dual stack (IPv4 + IPv6) configurations.

For consistency, this primer will assume the interface `emip10` will be using the address `192.168.0.2`.

**重要**

IP addresses can be set arbitrarily. As a result, it's possible for multiple devices to use the same IP address, resulting in an *address conflict*. Address conflicts should be avoided by using DHCP or SLAAC.

**提示**

IPv6 typically uses StateLess Address AutoConfiguration (SLAAC) for address configuration. In most cases, manually setting IPv6 addresses is a bad practice. If a specific address suffix is preferred, *interface identification tokens* can be used.

## 网络和 CIDR

Once an address is chosen, how does a device know how to talk to other devices?

IP *addresses* are associated with *networks*. IP *networks* are contiguous groups of addresses.

Classless *Inter-Domain Routing* or *CIDR* notation is used to distinguish network sizes.

- The *CIDR* value, often notated starting with a *l*, represents the size of the network.
  - The formula *2<sup>l</sup> (32 - CIDR)* can be used to calculate network size.
  - Once network size is calculated, usable node count must be reduced by 2.
    - The first IP in a network is the *Network address*, and the last is typically the *Broadcast address*. These addresses are special and cannot be used by normal hosts.

**提示**

The most common *CIDR* values are */24*, and */32*, representing **254** nodes and a single node respectively.

A *CIDR* of */24* is the de-facto default network size. This corresponds to a subnet mask of `255.255.255.0`, where the last 8 bits are reserved for IP addresses for nodes on a network.

The notation `192.168.0./24` can be interpreted as:

- The address `192.168.0.2`
- On the *network* `192.168.0.0`
- With a size of `254 (28 (32 - 24) - 2)`
  - Usable IPs are in the range `192.168.0.1 - 192.168.0.254`
- With a *broadcast address* of `192.168.0.255`
  - In most cases, the last address on a network is used as the *broadcast address*, but this can be changed.

Using this configuration, a device should be able to communicate with any host on the same *network* (`192.168.0.0`).

## The Internet

Once a device is on a network, how does it know how to talk to devices on the Internet?

To communicate with devices outside of local *networks*, routing must be used. A *router* is simply a network device that forwards traffic for other devices. The term *default route* or *gateway* typically refers to whatever device on the current network is used for external network access.

**提示**

It's a standard practice to make the *gateway* the first or last IP on a network.

If an Internet-connected router is available at `192.168.0.1`, it can be used as the *default route*, granting Internet access.

To summarize:

- Interfaces must be configured with an *address* and *network information*, such as the *CIDR* value.
- Local network access is used to access a *router* on the same network.
- The *default route* is configured, so traffic destined for external networks is forwarded to the *gateway*, providing Internet access.

## 域名系统

Remembering IPs is hard. The *Domain Name System* was created to allow mapping between *Domain Names* and *IP addresses*.

Linux systems use `/etc/resolv.conf` to define *nameserver* to be used for *DNS resolution*.

**提示**

Many *routers* can also function as a DNS server, and using a local DNS server can augment privacy and speed up queries through caching.

Many ISPs run a DNS server that is generally advertised to the *gateway* over DHCP. Using a local DNS server tends to improve query latency, but most public DNS servers will return the same results, so server usage is largely based on preference.

## 手动配置网络

### 配置接口地址

**重要**

When manually configuring IP addresses, the local network topology must be considered. IP addresses can be set arbitrarily; conflicts may cause network disruption.

To configure `emip10` with the address `192.168.0.2` and CIDR `/24`:

```
root # ip address add 192.168.0.2/24 dev emip10
```

**提示**

The start of this command can be shortened to `ip a`.

### 配置默认路由

Configuring address and network information for an interface will configure **link** routes, allowing communication with that network segment:

```
root # ip route
192.168.0.0/24 dev emip10 proto kernel scope link src 192.168.0.2
```

**提示**

This command can be shortened to `ip r`.

The `default` route can be set to `192.168.0.1` with:

```
root # ip route add default via 192.168.0.1
```

## 配置 DNS

Nameserver info is typically acquired using DHCP, but can be set manually by adding `nameserver` entries to `/etc/resolv.conf`.

**警告**

If `dhcpcd` is running, changes to `/etc/resolv.conf` will not persist. Status can be checked with `ps -x | grep dhcpcd`.

`nano` is included in Gentoo *boot media* and can be used to edit `/etc/resolv.conf` with:

```
root # nano /etc/resolv.conf
```

Lines containing the keyword `nameserver` followed by a DNS server IP address are queried in order of definition:

```
文件 /etc/resolv.conf Use Quad9 DNS.
nameserver 9.9.9.9
nameserver 149.112.112.112
```

```
文件 /etc/resolv.conf Use Cloudflare DNS.
nameserver 1.1.1.1
nameserver 1.0.0.1
```

DNS status can be checked by pinging a domain name:

```
root # ping -c 3 gentoo.org
```

Once connectivity has been verified, continue with [Preparing the disks](#).

← 选择媒介

Home

准备磁盘 →