

Debian

Debian, also known as Debian GNU/Linux, is a free and open source Linux distribution, developed by the Debian Project.

- [systemd: Creating a service](#)
- [Debian: Static IP Address Set Up](#)

systemd: Creating a service

Understanding and Managing Custom Services with systemd

systemd is the standard service and system manager for most Linux distributions, handling system startup, process management, and service control. If you're looking to automate or manage custom scripts or programs as services, systemd makes it easy to create and manage these services. Here's a comprehensive guide on creating, enabling, and managing custom services.

CHEATSHEET

1. **Create service file:** `/etc/systemd/system/my_custom_service.service`
2. **Reload systemd configuration:** `sudo systemctl daemon-reload`
3. **Enable service at startup:** `sudo systemctl enable my_custom_service.service`
4. **Start the service:** `sudo systemctl start my_custom_service.service`

DETAILS

Creating a Custom systemd Service

To create a custom service, start by creating a *service file*. This file will contain essential configurations and is typically stored in `/etc/systemd/system/`. Here's an example of a service file structure for running a custom script as a service:

```
[Unit]
Description=My Custom Service
After=network.target

[Service]
ExecStart=/usr/bin/my_script.sh
Type=simple

[Install]
WantedBy=multi-user.target
```

Breakdown of Service File Sections

- **[Unit]:** This section contains metadata about the service, such as its description and dependencies.
 - *Description:* Provides a brief overview of the service's purpose.
 - *After:* Specifies dependencies and ensures that `my_script.sh` runs only after the network is available.
 - **[Service]:** Defines how the service should be executed.
 - *ExecStart:* Specifies the path to the executable or script. In this example, the service runs `/usr/bin/my_script.sh`.
 - *Type:* The service type determines how systemd manages the process. Common types include:
 - `simple`: Default type, used when the process doesn't fork or exit quickly.
 - `forking`: Used if the process forks into the background.
 - **[Install]:** Determines how and when the service should be launched.
 - *WantedBy:* Specifies which target (runlevel) the service should start under. Setting this to `multi-user.target` means it will start when the system is in a multi-user, non-graphical environment.
-

Managing the Service with systemctl

Once you've created the service file, you'll need to use `systemctl`, the command-line tool for managing systemd services, to control it.

Steps to Enable and Start the Service

1. Load the Service Configuration

To ensure systemd reads your new service file, run the following command:

```
sudo systemctl daemon-reload
```

2. Enable the Service at Boot

This makes the service start automatically when the system boots:

```
sudo systemctl enable my_custom_service.service
```

3. Start the Service

Once enabled, start the service manually for immediate execution:

```
sudo systemctl start my_custom_service.service
```

Common systemctl Commands

- **Check Status:** See if the service is running and view recent log entries.

```
sudo systemctl status my_custom_service.service
```

- **Stop the Service:** Stop the service manually.

```
sudo systemctl stop my_custom_service.service
```

- **Restart the Service:** Restart the service to apply new changes.

```
sudo systemctl restart my_custom_service.service
```

- **Disable the Service:** Prevent the service from starting automatically at boot.

```
sudo systemctl disable my_custom_service.service
```

With these commands and an understanding of how to create a custom service file, you can effectively manage processes and tasks on your Linux system, enabling better automation and control.

Happy Me! ☺

Debian: Static IP Address Set Up

Debian uses different network management systems. In this article, I'll show you how to set a static IP address using all these methods.

? Find your Network Manager

To check which network management system your Debian is using, you have to check if some files exist that are used by either of your manager:

1. **networking:** `/etc/network/interfaces`
2. **dhcpcd:** `/etc/dhpcd.conf`
3. **systemd-networkd:** `/etc/systemd/network/*`

If neither of these files exist, your Debian uses **NetworkManager** to configure your network.

? Configure your Static IP

Firstly, you need to find the name of your local ethernet interface. To get this information, run the `ip a` command. The output should look something like this:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever

### ETHERNET INTERFACE

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether dc:a6:32:cc:b4:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global eth0
        valid_lft forever preferred_lft forever
```

```
inet6 2a02:a03f:a190:d700:dea6:32ff:fecc:b4ec/64 scope global dynamic mngtmpaddr
    valid_lft 86364sec preferred_lft 71964sec
inet6 fe80::dea6:32ff:fecc:b4ec/64 scope link
    valid_lft forever preferred_lft forever
```

###

```
3: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group
default qlen 1000
    link/ether dc:a6:32:cc:b4:ee brd ff:ff:ff:ff:ff:ff
```

Find the group with your IP address. In my case, it is the group `2:`. From here, the interface name is `eth0`

networking

To assign a static IP address using the `networking` service, edit the `/etc/network/interfaces` file:

```
auto eth0
iface eth0 inet static
    address 192.168.1.100
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameservers 8.8.8.8 1.1.1.1
```

Description line by line:

1. `auto eth0`: add the `eth0` interface to the list of interfaces that you want brought up at boot time.
2. `iface eth0 inet static`: tells Debian to set the IPv4 (`inet`) `eth0` interface in `static` mode.
3. `address`: the static address you want.
4. `netmask`: the netmask of your network.
5. `gateway`: the gateway of your network.
6. `dns-nameservers`: your DNS IP addresses separate by a space.

Next, brought `up` and `down` your network interface:

```
sudo ifdown eth0
sudo ifup eth0
```