

- [Skip to main content](#)
- [Skip to footer](#)
- [Accessibility statement and help](#)



- [Hardware](#)

Low-cost, high-performance Raspberry Pi computers and accessories

[Buy Raspberry Pi 4](#) [Buy Raspberry Pi 400](#)

- [All products](#)
- [For industry](#)
- [Documentation](#)
- [Forums](#)

- [Software](#)

Everything you need to get started with your Raspberry Pi computer

[Our software](#)

- [Raspberry Pi OS](#)
- [Raspberry Pi Desktop](#)
- [Help](#)
- [Forums](#)

- [Books & magazines](#)

Books and magazines from Raspberry Pi Press

[Explore our titles](#)

- [The MagPi](#)
- [HackSpace](#)
- [Wireframe](#)
- [Custom PC](#)

- [Learn](#)

Free resources for young people to learn to code and become digital makers

[Learn at home](#)

- [Guided coding projects](#)
- [Learn at a Code Club](#)
- [Learn at a CoderDojo](#)

- [Teach](#)

Free training, resources, and guidance to help you teach computing with confidence

[Support for teachers](#)

- [Teach Computing](#)
- [Start a Code Club](#)
- [Online training courses](#)

- [Forums](#)

- [About us](#)

We work to put the power of computing and digital making into the hands of people all over the world

[About us](#)

- [Donate](#)
- [Support us](#)
- [Our supporters](#)
- [Blog](#)

DOCUMENTATION > LINUX > USAGE > SYSTEMD

systemd

In order to have a command or program run when the Pi boots, you can add it as a service. Once this is done, you can start/stop enable/disable from the linux prompt.

Creating a service

On your Pi, create a .service file for your service, for example:

myscript.service

```
[Unit]
Description=My service
After=network.target

[Service]
ExecStart=/usr/bin/python3 -u main.py
WorkingDirectory=/home/pi/myscript
StandardOutput=inherit
StandardError=inherit
Restart=always
User=pi

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

So in this instance, the service would run Python 3 from our working directory `/home/pi/myscript` which contains our python program to run `main.py`. But you are not limited to Python programs: simply change the ExecStart line to be the command to start any program/script that you want running from booting.

Copy this file into `/etc/systemd/system` as root, for example:

```
sudo cp myscript.service /etc/systemd/system/myscript.service
```

Once this has been copied, you can attempt to start the service using the following command:

```
sudo systemctl start myscript.service
```

Stop it using following command:

```
sudo systemctl stop myscript.service
```

When you are happy that this starts and stops your app, you can have it start automatically on reboot by using this command:

```
sudo systemctl enable myscript.service
```

The `systemctl` command can also be used to restart the service or disable it from boot up!


Some things to be aware of:


- The order in which things are started is based on their dependencies — this particular script should start fairly late in the boot process, after a network is available (see the After section).
- You can configure different dependencies and orders based on your requirements.

You can get more information from: `man systemctl` or here:
<https://fedoramagazine.org/what-is-an-init-system/>

VIEW/EDIT THIS PAGE ON GITHUB

READ OUR USAGE AND CONTRIBUTIONS POLICY





ABOUT US

- About us
- Our team
- Governance
- Safeguarding
- Research
- Jobs
- Contact us

SUPPORT

- Help
- Documentation
- Projects
- Training
- Software
- Forums
- FAQ

SIGN UP TO OUR NEWSLETTER

Your email here

SUBSCRIBE