

# systemd Timer

As a modern `systemd` replacement for `chron`, the following is an example of `oneshot` **Unit** triggered periodically by a **Timer**.

Reference: <https://www.buggycoder.com/network-backups-using-rsync/>

Some people dislike `systemd`, but on a modern linux distro, it's the best way to keep everything as "stock" and "plain vanilla" as possible.

Example `oneshot` **Unit** file:

```
/etc/systemd/system/<example-unit.service>
```

```
[Unit]
Description= <Example Unit Name>

#target requirements (may vary)
Requires=network.target
After=network.target

[Service]
Type=oneshot

#optional settings
Nice=19
StandardOutput=journal
IOSchedulingClass=best-effort
IOSchedulingPriority=5

ExecStart= <command> or </absolute/path/to/exec>

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

Example **Timer** file to execute the **Unit**:

```
/etc/systemd/system/<example-unit.timer>
```

```
[Unit]
Description= <Example Timer Name>
Requires=<example-unit>.service

[Timer]
OnCalendar=daily
Unit=<example-unit>.service

[Install]
WantedBy=timers.target
```

Enable the **Unit** and **Timer**:

```
systemctl daemon-reload

systemctl enable <example-unit>.service
systemctl enable <example-unit>.timer

systemctl start <example-unit>.service
systemctl start <example-unit>.timer

# Ensure all is well
journalctl -f -u <example-unit>.service
```

---

Revision #1

Created 4 August 2023 13:37:07 by giw

Updated 4 August 2023 14:02:29 by giw