

systemd timer

If systemd is installed on the system, a systemd timer could be an alternative to a cronjob.

This approach requires two files: `myappcron.service` and `myappcron.timer`. Create these two files in `/etc/systemd/system/`.

`myappcron.service` should look like this:

```
[Unit]
Description=myapp cron.php job

[Service]
User=www-data
ExecStart=/usr/bin/php -f /var/www/myapp/cron.php

[Install]
WantedBy=basic.target
```

Replace the user `www-data` with the user of your http server and `/var/www/myapp/cron.php` with the location of `cron.php` in your nextcloud directory.

`myappcron.timer` should look like this:

```
[Unit]
Description=Run myapp cron.php every 5 minutes

[Timer]
OnBootSec=5min
OnUnitActiveSec=5min
Unit=myappcron.service

[Install]
WantedBy=timers.target
```

The important parts in the timer-unit are `OnBootSec` and `OnUnitActiveSec`. `OnBootSec` will start the timer 5 minutes after boot, otherwise you would have to start it manually after every boot. `OnUnitActiveSec` will set a 5 minute timer after the service-unit was last activated.

Now all that is left is to start and enable the timer by running this command:

```
# systemctl enable --now myappcron.timer
```

When the option `--now` is used with `enable`, the resp. unit will also be started.

Last update: **2019/11/20 08:28**