

Автоматически перезапуск сервисов в Linux

Интернет · 14.01.2026 · PDF-версия статьи с сайта cms.dolinais.ru

Сервис в «Линуксе» — это программа, которая работает в фоновом режиме: Apache, MySQL, боты

Как понять, что сервис «упал»

Сайт на VPS перестал открываться. Программа, подключающаяся к базе, выдаёт ошибку. Telegram-бот молчит. Обычно мы узнаём об этом от клиентов. Лучше — узнать первым и сделать так, чтобы сервис сам восстанавливался.

Чтобы не полагаться на жалобы пользователей, можно внедрить простую систему мониторинга. Вот несколько способов узнать о проблеме вовремя.

Проверка статуса через systemctl. Выполните команду:

`systemctl status имя_сервиса` Это позволит увидеть текущее состояние службы.

Журнал systemd. Используйте: `journalctl -u имя_сервиса --since "1 hour ago"` Это поможет узнать, были ли ошибки или аварийные остановки.

Пинг до порта или HTTP-запрос. Для веб-серверов и ботов можно настроить **curl** или **netcat**: `curl -s http://localhost:порт` или `nc -z localhost порт` Если ответа нет, сервис, скорее всего, «упал».

В популярных версиях Linux (Ubuntu, Debian, CentOS, Fedora и др.) основным механизмом запуска и управления службами стал **systemd**.

```
sudo systemctl start имя_сервиса    # Запустить вручную
```

```
sudo systemctl stop имя_сервиса     # Остановить
```

```
sudo systemctl status имя_сервиса   # Проверить статус
```

Ещё **systemd** может перезапустить сервис сам, если тот перестал работать. Главное — правильно его настроить.

Если вы не знаете, как называется нужный сервис, выполните: `systemctl list-units --type=service`

Или просто начните вводить **systemctl status** и нажмите **Tab** — автодополнение покажет доступные варианты.

Автоматический перезапуск службы через systemd

Представим, что на сервере есть нужный нам сервис — назовём его **myservice**. Его автозапуск в случае

остановки можно настроить через встроенную систему **systemd**.

Необходимо открыть unit-файл. Пример: `sudo nano /etc/systemd/system/myservice.service`

Найти его можно, введя: `systemctl cat myservice` Она покажет путь и содержимое unit-файла для выбранного сервиса.

Если unit-файла нет

Иногда бывает, программа, запущенная вручную работает, но **systemctl** про неё ничего не знает. Значит, что **systemd** не управляет этим действием. В этой ситуации можно создать свой unit-файл.

Допустим, у нас есть файл **/home/user/myscript.sh** со скриптом. Тогда вводим: `sudo nano /etc/systemd/system/myscript.service` И вставляем:

```
[Unit]
```

```
Description=Мой сервис
```

```
[Service]
```

```
ExecStart=/bin/bash /home/user/myscript.sh
```

```
Restart=on-failure
```

```
RestartSec=5
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Сохраняем, затем:

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable myscript
```

```
sudo systemctl start myscript
```

Теперь **systemd** отвечает за запуск и работу этой программы, как и за другие службы в системе. Всё — она стала полноценной «службой».

Добавляем параметры для автозапуска

В файле ищем раздел **[Service]** — именно туда нужно добавить строки, отвечающие за автоматический

перезапуск:

```
Restart=on-failure
```

```
RestartSec=5
```

Что они делают:

- **Restart=on-failure** — говорит системе перезапустить сервис, если тот завершился с ошибкой;
- **RestartSec=5** — даёт системе команду подождать 5 секунд перед повторным запуском.

Важно: эти строки должны быть именно в блоке [Service], а не в [Unit] или [Install]. Если поставить их не туда — ничего работать не будет.

Сохраняем изменения и обновляем systemd.

После того как вы добавили параметры, сохраняем файл и перезапускаем systemd, чтобы он учёл изменения:

```
sudo systemctl daemon-reload
```

Включаем сервис и проверяем перезапуск

Введите команды:

```
sudo systemctl enable myservice
```

```
sudo systemctl restart myservice
```

Проверим, как система поведёт себя, если сервис вдруг завершится с ошибкой. Принудительно «завалим» его командой:

```
sudo kill -9 $(pidof myservice)
```

Теперь ждём несколько секунд и проверяем статус:

```
systemctl status myservice
```

Чтобы убедиться, что перезапуск действительно происходил, можно посмотреть журнал: `journalctl -u myservice --since "10 minutes ago"`

Альтернатива systemd: простой скрипт с cron

Если **systemd** по какой-то причине не справляется с автоматическим перезапуском сервиса, можно сделать подстраховку. Особенно, если сервис нестабилен или надо проверить, что он точно работает. Создадим инструмент, который следит за состоянием сервиса и в случае сбоя сам его перезапускает. А затем подключим его к cron — встроенному в Linux планировщику заданий.

Создаём файл с инструкцией

Откройте терминал. В домашней папке (или где вам удобно) сделаем новый файл с названием, например: `pano check_service.sh`

Вставляем в него этот код:

```
#!/bin/bash

SERVICE=myservice

if ! systemctl is-active --quiet $SERVICE

then

echo "$(date): $SERVICE не работает, перезапускаем..." >>

/var/log/service_monitor.log

systemctl restart $SERVICE

fi
```

Пояснение:

- **SERVICE=myservice** — вместо **myservice** напишите название своего сервиса (например, `nginx`, `apache2`, `ssh` и т.д.);
- скрипт проверяет, работает ли сервис. Если он неактивен — пишет об этом в журнал `/var/log/service_monitor.log` и перезапускает его.

Делаем скрипт исполняемым

Чтобы Linux разрешил запускать этот файл как программу, выполним:

```
chmod +x check_service.sh
```

Подключаем через cron

Теперь можно настроить автоматический запуск этой проверки, например, каждые 5 минут.

Открываем планировщик заданий `cron`: `crontab -e`

Если откроется редактор, то выбирайте `pano`, он самый простой.

В самый конец файла добавьте строку:

```
* /5 * * * * /полный/путь/к/check_service.sh
```

Узнать, где находится файл, можно с помощью команды `pwd` в той папке, где он лежит. Например, если в домашней папке:

```
/home/username/check_service.sh
```

И тогда строка в `crontab` будет выглядеть так:

```
* /5 * * * * /home/username/check_service.sh
```

Теперь каждые 5 минут система будет запускать проверочный скрипт. Если сервис «упал», программа его перезапустит — и об этом будет запись в логе.

Убедитесь, что демон `cron` запущен: `sudo systemctl status cron`

Если он остановлен, введите: `sudo systemctl start cron`

Источник: <http://taskchat.ru/avtomaticheskii-perezapusk-servisov-v-linux>