systemctl — ??????????

Systemctl — утилита для управления **systemd** — системой инициализации и менеджером служб в Linux. Она отвечает за запуск и контроль демонов, таймеров, точек монтирования и других компонентов, описываемых *unit-файлами*.

??????

systemd управляет набором объектов: сервисами, сокетами, таймерами, монтированиями и т.д. Каждый объект описывается unit-файлом; `systemctl` — основной интерфейс для взаимодействия с этими unit'ами и с самой системой.

??? ????? unit-?????

Unit-файл — это конфигурационный файл, который описывает, как именно systemd должен управлять конкретным компонентом (сервисом, сокетом, таймером и т.д.). Обычно имеют расширение `.service`, `.socket`, `.timer`, `.mount`, `.target` и т.д.

Где хранятся:

- `/usr/lib/systemd/system/` — unit-файлы поставщика (пакетов).
- `/etc/systemd/system/` — локальные или переопределяющие unit-файлы администратора.

Пример простого unit-файла (`nginx.service`):

```
Description=A high performance web server
After=network.target
[Service]
ExecStart=/usr/sbin/nginx -g 'daemon off;'
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s QUIT $MAINPID
Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Пояснение полей:

- `[Unit]` — метаинформация и зависимости (например, `After=`).
- `[Service]` — команды запуска/остановки, поведение при сбое (`Restart`), пользователь и окружение.
- `[Install]` — как unit включается в target'ы (автозапуск).

??????? ??????????

???????? ??????? ???????

Команда показывает состояние (active/failed), PID, путь к unit-файлу и последние строки логов:

```
sudo systemctl status nginx
```

?????? / ????????? / ??????????

Запуск вручную (не влияет на автозапуск при следующей загрузке):

```
sudo systemctl start nginx
```

Остановка:

```
sudo systemctl stop nginx
```

Перезапуск (полностью остановит, затем запустит):

```
sudo systemctl restart nginx
```

Перезапустить только если сервис уже запущен:

```
sudo systemctl try-restart nginx
```

Когда использовать: перезапуск обязателен при изменении конфигурации демона, но перед этим убедитесь, что unit-файл и конфигурация корректны.

???????????? ????????

enable / disable

`enable` создаёт символьные ссылки unit'а в каталогах target'ов — сервис запустится при загрузке системы. `disable` убирает эти ссылки, но не мешает ручному запуску.

```
sudo systemctl enable nginx
sudo systemctl disable nginx
systemctl is-enabled nginx
```

Важно: `enable` не запускает сервис немедленно — для этого используйте `start` или `restart`.

???????????? (mask / unmask)

Иногда нужно гарантированно запретить запуск сервиса — даже случайный или зависимый запуск. Для этого используется `mask`, который перенаправляет unit на `/dev/null`.

```
sudo systemctl mask nginx    # сервис нельзя будет запустить ни вручную, ни зависимостями
sudo systemctl unmask nginx  # снять маску
```

Когда применять `mask`:

- Нужно полностью заблокировать конфликтующий сервис (например, запрещаем start сетевого демона, если используем другой).
- Убрать возможность случайного запуска устаревших, небезопасных демонов (telnet, rsh и т.п.).
- Тестирование: временно запрещаем сервис, чтобы проверить поведение зависимостей.

?????? ? ??????????? unit-??????

После добавления/изменения unit-файла **systemd** нужно пересканировать. Это делается командой:

```
sudo systemctl daemon-reload
```

Дальше обычно перезапускают сервис:

```
sudo systemctl restart имя_сервиса
```

Если забыть `daemon-reload`, systemd будет использовать старую конфигурацию, даже если файл на диске изменён.

Targets (?????? runlevels)

В systemd понятие runlevel заменено на *targets* — наборы unit'ов, которые запускаются вместе.

- `multi-user.target` — многопользовательский режим без графики (аналог runlevel 3).
- `graphical.target` — графический режим (аналог runlevel 5).
- `rescue.target` — аварийный режим.

```
systemctl get-default               # посмотреть текущую цель
sudo systemctl set-default multi-user.target  # установить дефолтную цель
sudo systemctl isolate multi-user.target     # переключиться сразу
```

????: journalctl

systemd использует журнал (journald). Для просмотра логов сервиса применяют:

```
journalctl -u nginx         # все логи для unit
journalctl -u nginx -f      # следить в реальном времени
```

Полезно сочетать с временными фильтрами: `--since`, `--until`, и с `-p` (уровень приоритета).

???????? ??????? ? ??????

```
systemctl list-units --type=service        # активные службы
systemctl list-unit-files --type=service     # все файлы unit (включая disabled)
systemctl is-active ssh                       # быстро узнать, активен ли сервис
systemctl list-dependencies имя_сервиса       # показать зависимости
```

???????? ?????? ? ?? ???????

1) ??????? unit-????, ?? ????????? ?? ???????????

```
sudo systemctl daemon-reload
sudo systemctl restart имя_сервиса
```

Обязательная последовательность: `daemon-reload` → перезапуск/старт.

2) `Failed to start ...`

Проверяем статус и логи:

```
sudo systemctl status имя_сервиса
journalctl -xeu имя_сервиса
```

Частые причины: ошибка в пути в `ExecStart`, неверные права, конфликт портов, отсутствующие зависимости.

3) ?????? ??????????? ???????, ?? ?? ???????? ??? ????????

```
systemctl is-enabled имя_сервиса
sudo systemctl enable имя_сервиса
```

Проверьте также, не мешают ли target'ы или маскировка, и нет ли ошибок в `[Install]` блока unit-файла (например, `WantedBy=`).

4) "Unit not found"

Скорее всего unit-файл отсутствует или не установлен:

```
systemctl list-unit-files | grep имя_сервиса
```

Возможные причины: пакет не установлен, unit называется иначе, или unit расположен в нестандартном каталоге.

5) Unit ??????? ??? `masked`

```
systemctl is-enabled имя_сервиса   # покажет "masked"
sudo systemctl unmask имя_сервиса
```

???? unit-?????? (??????)

- `.service` — сервис/демон.
- `.socket` — сокет-активация; может запускать .service при подключении.
- `.timer` — периодические/отложенные задания (альтернатива cron).
- `.mount` — описание точки монтирования.
- `.target` — агрегатор unit'ов (аналог runlevel).

???????? ???????????? (best practices)

- Редактируйте unit-файлы в `/etc/systemd/system/` (локальные переопределения не затрутся при обновлении пакета).
- После правок — всегда `daemon-reload`, затем `restart` или `try-restart`.
- Используйте `Restart=on-failure` аккуратно; логируйте причины падений (через `StandardOutput`/`StandardError` или journald).
- Для временной блокировки сервиса применяйте `mask`. Для обычного отключения — `disable`.

Revision #1
Created 5 October 2025 12:08:02 by Admin
Updated 5 October 2025 12:11:23 by Admin