# ***Systemd: Создание Сервиса — Примеры***

Большинство дистрибутивов Linux в качестве менеджера системы и сервисов используют systemd.

systemctl является основной командой для управления сервисами в systemd.

В данной статье я покажу, как создать service-файл в systemd, который позволит управлять вашим сервисом с помощью команды systemctl, как без перезагрузки перезапустить systemd, чтобы он перечитал unit-файлы и как активировать ваш новый сервис.

Также я приведу описание наиболее важных опций используемых в service-файлах с примерами реальных service-файлов.

Создание Сервиса в Systemd

Создайте service-файл /etc/systemd/system/foo-daemon.service (замените foo-daemon на имя вашего сервиса):

$ sudo touch /etc/systemd/system/foo-daemon.service

$ sudo chmod 664 /etc/systemd/system/foo-daemon.service

Откройте файл foo-daemon.service и пропишите минимальные настройки, которые позволят управлять сервисом с помощью systemctl:

[Unit]

Description=Foo

[Service]

ExecStart=/usr/sbin/foo-daemon

[Install]

WantedBy=multi-user.target

**Путь К Демону:** Если вы не знаете путь к демону, попробуйте which foo-daemon.

После создания нового service-файла необходимо перезапустить systemd:

$ sudo systemctl daemon-reload

Теперь вы можете делать start, stop, restart и проверять status сервиса:

$ sudo systemctl start foo-daemon

$ sudo systemctl stop foo-daemon

$ sudo systemctl restart foo-daemon

$ systemctl status foo-daemon

Чтобы добавить сервис в автозагрузку, необходимо активировать его:

$ sudo systemctl enable foo-daemon

Чтобы проверить логи сервиса, выполните:

$ journalctl -u foo-daemon

Опции Service-файла в Systemd

Service-файла в systemd обычно состоит из трех секций.

Общие элементы конфигурации сервиса настраиваются в секциях [Unit] и [Install]

Параметры конфигурации конкретного сервиса настраиваются в секции [Service].

Важные Опции Секции [Unit]

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| Description | Краткое описание юнита. |
| Documentation | Список ссылок на документацию. |
| Before, After | Порядок запуска юнитов. |
| Requires | Если этот сервис активируется, перечисленные здесь юниты тоже будут активированы. Если один из перечисленных юнитов останавливается или падает, этот сервис тоже будет остановлен. |
| Wants | Устанавливает более слабые зависимости, чем Requires. Если один из перечисленных юнитов не может успешно запуститься, это не повлияет на запуск данного сервиса. Это рекомендуемый способ установления зависимостей. |
| Conflicts | Если установлено что данный сервис конфликтует с другим юнитом, то запуск последнего остановит этот сервис и наоборот. |

Список всех опций секции [Unit]:

$ man systemd.unit

Важные Опции Секции [Install]

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| Alias | Дополнительные имена сервиса разделенные пробелами. Большинство команд в systemctl, за исключением systemctl enable, могут использовать альтернативные имена сервисов. |
| RequiredBy, WantedBy | Данный сервис будет запущен при запуске перечисленных сервисов. Для более подробной информации смотрите описание опций Wants и Requires в секции [Unit]. |
| Also | Определяет список юнитов, которые также будут активированы или дезактивированы вместе с данным сервисом при выполнении команд systemctl enable или systemctl disable. |

Список всех опций секции [Install]:

$ man systemd.unit

Важные Опции Секции [Service]

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| Type | Настраивает тип запуска процесса. Один из: simple (по умолчанию) — запускает сервис мгновенно. Предполагается, что основной процесс сервиса задан в ExecStart. forking — считает сервис запущенным после того, как родительский процесс создает процесс-потомка, а сам завершится. oneshot — аналогичен типу simple, но предполагается, что процесс должен завершиться до того, как systemd начнет отслеживать состояния юнитов (удобно для скриптов, которые выполняют разовую работу и завершаются). Возможно вы также захотите использовать RemainAfterExit=yes, чтобы systemd продолжал считать сервис активным и после завершения процесса. dbus — аналогичен типу simple, но считает сервис запущенным после того, как основной процесс получает имя на шине D-Bus. notify — аналогичен типу simple, но считает сервис запущенным после того, как он отправляет systemd специальный сигнал. idle — аналогичен типу simple, но фактический запуск исполняемого файла сервиса откладывается, пока не будут выполнены все задачи. |
| ExecStart | Команды вместе с аргументами, которые будут выполнены при старте сервиса. Опция Type=oneshot позволяет указывать несколько команд, которые будут выполняться последовательно. Опции ExecStartPre и ExecStartPost могут задавать дополнительные команды, которые будут выполнены до или после ExecStart. |
| ExecStop | Команды, которые будут выполнены для остановки сервиса запущенного с помощью ExecStart. |
| ExecReload | Команды, которые будут выполнены чтобы сообщить сервису о необходимости перечитать конфигурационные файлы. |
| Restart | Если эта опция активирована, сервис будет перезапущен если процесс прекращен или достигнут timeout, за исключением случая нормальной остановки сервиса с помощью команды systemctl stop |
| RemainAfterExit | Если установлена в значение True, сервис будет считаться запущенным даже если сам процесс завершен. Полезен с Type=oneshot. Значение по умолчанию False. |

Список всех опций секции [Service]:

$ man systemd.service

Примеры Service-файлов в Systemd

[Unit]

Description=The NGINX HTTP and reverse proxy server

After=syslog.target network.target remote-fs.target nss-lookup.target

[Service]

Type=forking

PIDFile=/run/nginx.pid

ExecStartPre=/usr/sbin/nginx -t

ExecStart=/usr/sbin/nginx

ExecReload=/bin/kill -s HUP $MAINPID

ExecStop=/bin/kill -s QUIT $MAINPID

PrivateTmp=true

[Install]

WantedBy=multi-user.target

[Unit]

Description=The Apache HTTP Server

After=network.target remote-fs.target nss-lookup.target

[Service]

Type=notify

EnvironmentFile=/etc/sysconfig/httpd

ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND

ExecReload=/usr/sbin/httpd $OPTIONS -k graceful

ExecStop=/bin/kill -WINCH ${MAINPID}

KillSignal=SIGCONT

PrivateTmp=true

[Install]

WantedBy=multi-user.target

[Unit]

Description=Redis persistent key-value database

After=network.target

[Service]

ExecStart=/usr/bin/redis-server /etc/redis.conf --daemonize no

ExecStop=/usr/bin/redis-shutdown

User=redis

Group=redis

[Install]

WantedBy=multi-user.target

Больше примеров в документации по [systemd.service](https://www.freedesktop.org/software/systemd/man/systemd.service.html" \t "_blank) и [systemd.unit](https://www.freedesktop.org/software/systemd/man/systemd.unit.html" \t "_blank).