

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 |
|----------------------------|--|
| <code>Description</code> | Краткое описание юнита. |
| <code>Documentation</code> | Список ссылок на документацию. |
| <code>Before, After</code> | Порядок запуска юнитов. |
| <code>Requires</code> | Если этот сервис активируется, перечисленные здесь юниты тоже будут активированы. Если один из перечисленных юнитов останавливается или падает, этот сервис тоже будет остановлен. |
| <code>Wants</code> | Устанавливает более слабые зависимости, чем <code>Requires</code> . Если один из перечисленных юнитов не может успешно запуститься, это не повлияет на запуск данного сервиса. Это рекомендуемый способ установления зависимостей. |
| <code>Conflicts</code> | 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 |
|---|--|
| <code>Alias</code> | Дополнительные имена сервиса разделенные пробелами. Большинство команд в <code>systemctl</code> , за исключением <code>systemctl enable</code> , могут использовать альтернативные имена сервисов. |
| <code>RequiredBy</code> , <code>WantedBy</code> | Данный сервис будет запущен при запуске перечисленных сервисов. Для более подробной информации смотрите описание опций <code>Wants</code> и <code>Requires</code> в секции <code>[Unit]</code> . |
| <code>Also</code> | Определяет список юнитов, которые также будут активированы или деактивированы вместе с данным сервисом при выполнении команд <code>systemctl enable</code> или <code>systemctl disable</code> . |

Список всех опций секции `[Install]`:

```
$ man systemd.unit
```

Важные Опции Секции `[Service]`

| Опция | Описание |
|-------------------------|---|
| <code>Type</code> | <p>Настраивает тип запуска процесса. Один из:</p> <ul style="list-style-type: none"> <code>simple</code> (по умолчанию) — запускает сервис мгновенно. Предполагается, что основной процесс сервиса задан в <code>ExecStart</code>. <code>forking</code> — считает сервис запущенным после того, как родительский процесс создает процесс-потомка, а сам завершится. <code>oneshot</code> — аналогичен типу <code>simple</code>, но предполагается, что процесс должен завершиться до того, как <code>systemd</code> начнет отслеживать состояния юнитов (удобно для скриптов, которые выполняют разовую работу и завершаются). <p>Возможно вы также захотите использовать <code>RemainAfterExit=yes</code>, чтобы <code>systemd</code> продолжал считать сервис активным и после завершения процесса.</p> <ul style="list-style-type: none"> <code>dbus</code> — аналогичен типу <code>simple</code>, но считает сервис запущенным после того, как основной процесс получает имя на шине D-Bus. <code>notify</code> — аналогичен типу <code>simple</code>, но считает сервис запущенным после того, как он отправляет <code>systemd</code> специальный сигнал. <code>idle</code> — аналогичен типу <code>simple</code>, но фактический запуск исполняемого файла сервиса откладывается, пока не будут выполнены все задачи. |
| <code>ExecStart</code> | <p>Команды вместе с аргументами, которые будут выполнены при старте сервиса. Опция <code>Type=oneshot</code> позволяет указывать несколько команд, которые будут выполняться последовательно.</p> <p>Опции <code>ExecStartPre</code> и <code>ExecStartPost</code> могут задавать дополнительные команды, которые будут выполнены до или после <code>ExecStart</code>.</p> |
| <code>ExecStop</code> | Команды, которые будут выполнены для остановки сервиса запущенного с помощью <code>ExecStart</code> . |
| <code>ExecReload</code> | 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 <code>systemctl stop</code> |
| RemainAfterExit | Если установлена в значение <code>True</code> , сервис будет считаться запущенным даже если сам процесс завершен. Полезен с <code>Type=oneshot</code> . Значение по умолчанию <code>False</code> . |

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](#) и [systemd.unit](#).