Если вы видите что-то необычное, просто сообщите мне.

Запуск java как сервиса через systemd

Предоставим у вас есть jar файл и вам нужно его запустить как сервис. Так же есть необходимость запускать его автоматически когда система перезагружается.

Убунту имеет встроенный механизм для создания сервисом в ручном режиме, запуск во время загрузи системы и останавливать и запускать как сервис. В этой статье мы сделаем простую сервис обертку для вашего jar файла, который можно будет запускать как сервис. Начнем.

1) Создадим сервис

```
sudo vim /etc/systemd/system/my-webapp.service
```

Копируем следующее содержание в файл `/etc/systemd/system/my-webapp.service`:

```
[Unit]
Description=My Webapp Java REST Service
[Service]
User=ubuntu
# Конфигурационный файл приложения application.properties должен быть тут:

# Замените на вашу рабочую папку
WorkingDirectory=/home/ubuntu/workspace

# Путь к экзешнику.
# Экзешник это bash скрипт, который вызывает jar файл
ExecStart=/home/ubuntu/workspace/my-webapp
```

```
SuccessExitStatus=143
TimeoutStopSec=10
Restart=on-failure
RestartSec=5

[Install]
WantedBy=multi-user.target
```

2) Создаём Bash скрипт для вызова сервисов.

Это bash скрипт, который вызывает JAR файл: my-webapp

```
#!/bin/sh
sudo /usr/bin/java -jar my-webapp-1.0-SNAPSHOT.jar server config.yml
```

Не забудем выдать скрипту права на исполнение:

```
sudo od u+x my-webapp
```

3) Запуск сервиса

```
sudo systemctl daemon-reload
sudo systemctl enable my-webapp.service
sudo systemctl start my-webapp
sudo systemctl status my-webapp
```

4) Настройка логирования

Первый запуск:

```
sudo journalctl --unit=my-webapp
```

Чтобы увидеть логи в реальном времени используйте опцию `-f`.

Если вы хотите обрезать логи, используйте `-n <# количество линий>`, чтобы увидеть нужное количество строк лога:

```
sudo journalctl -f -n 1000 -u my-webapp
```

Для отображения части логов используйте ключ `-f`:

```
sudo journalctl -f -u my-webapp
```

Остановите сервис с помощью команды:

```
sudo systemctl stop my-webapp
```

Revision #4
Created 2021-04-18 10:32:34 UTC by gasick
Updated 2022-11-02 07:24:30 UTC by gasick