

How to set up proper start/stop services

15. September 2014 — [5 Comments](#)

I've noticed that there are a lot of faulty or non properly working **start/stop services** out there in the wild. That's why I decided to build and explain some example services for the most common distributions and operating systems (like Windows).

In this case, I assume that our example service is Python's `SimpleHTTPServer` which starts a web server on port 80 and serves the current working directory. The server can be easily executed by:

```
$ python -m SimpleHTTPServer 8000
```

(Caution: this command will probably serve the root directory of your machine. Make sure to stop the service again!)

Debian and Ubuntu (sysvinit)

1. Create an user for the desired service
2. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

3. Copy the following script (as root) to `/etc/init.d/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/debian -O /etc/init.d/example
```

4. Adjust the variables:

```
sudo vi /etc/init.d/example
```

5. Make sure the script is executable:

```
chmod +x /etc/init.d/example
```

6. Enable the daemon with:

```
update-rc.d example defaults
```

7. Start the service with:

```
service example start
```

Ubuntu (upstart)

1. Create an user for the desired service
2. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

3. Copy the following script to `/etc/init/`:

```
sudo wget https://raw.githubusercontent.com/frdmn/service-daemons/master/ubuntu -O /etc/init/example.conf
```

4. Adjust the variables:

```
sudo vi /etc/init/example.conf
```

5. Start the service with:

```
sudo start example
```

CentOS 6 (sysvinit)

1. Create an user for the desired service
2. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

3. Copy the following script (as root) to `/etc/init.d/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/centos -O /etc/init.d/example
```

4. Adjust the variables (as root):

```
vi /etc/init.d/example
```

5. Make sure the script is marked as executable:

```
chmod +x /etc/init.d/example
```

6. Enable the config in in runlevels 2, 3, 4, and 5:

```
chkconfig example on
```

7. Start the service with:

```
service example start
```

Arch Linux (systemd)

1. Create an user for the desired service
2. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

3. Copy the following script (as root) to `/etc/systemd/system/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/arch -O /etc/systemd/system/example.service
```

4. Adjust the variables (as root):

```
/etc/systemd/system/example.service
```

5. Make sure the script is executable:

```
chmod +x /etc/systemd/system/example.service
```

6. Enable the script on boot with:

```
systemctl enable example
```

7. To start the script:

```
systemctl start example
```

Gentoo (runscript)

1. Create an user for the desired service
2. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

3. Copy the following script (as root) to `/etc/init.d/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/gentoo-script -O /etc/init.d/example
```

4. Copy the following configuration file (as root) to `/etc/conf.d/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/gentoo-conf -O /etc/conf.d/example
```

5. Make those files executable:

```
chmod +x /etc/init.d/example /etc/conf.d/example
```

6. Load the daemon:

```
rc-update add etherpad-lite default
```

7. Start the daemon with:

```
rc-service etherpad-lite start
```

Mac OS X (launchd)

Note: For some reasons the SimpleHTTPServer python module didn't work via `launchd`, so I included this tiny web server python script in [the repo](https://raw.githubusercontent.com/frdmn/service-daemons/master/macosx-httpd.py) [https://raw.githubusercontent.com/frdmn/service-daemons/master/macosx-httpd.py] .

1. Create an user for the desired service
2. Create a log folder for the service:

```
mkdir /var/log/example
```

3. Ensure the created user has full access to the binary you want to set up:

```
/usr/bin/python
```

4. Copy the following script (as root) to `/Library/LaunchDaemons/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/macosx -O /Library/LaunchDaemons/mn.frd.example.plist
```

5. Copy the python script (as root) to `/tmp/`:

```
wget https://raw.githubusercontent.com/frdmn/service-daemons/master/macosx-httpd.py -O /tmp/httpd.py
```

6. Launch the daemon with:

```
sudo launchctl load /Library/LaunchDaemons/mn.frd.example.plist
```

Windows (nssm)

1. Download and install `nssm`:
[non sucking service manager](http://nssm.cc/) [http://nssm.cc/]
2. Move it into your %PATH%
3. Open an administrative terminal window
4. Load the service: `nssm install "C:/Python27/python" -m SimpleHTTPServer 8000`
5. Reboot your machine

Please let me know in case you found some typos or errors above. You can also fork the repo and send a [Pull Request](#) via [GitHub](https://github.com/frdmn/service-daemons) [https://github.com/frdmn/service-daemons] if you have some improvements for the start/stop daemons.