

# Create Services in Alpine Linux

I began my development adventure, as everyone else, with simple CLI apps (mostly incredibly useless...), and my next step was with web development.

First with `Golang`, then `Node.js` and then others.

Little by little, I learned to create decent-looking websites and I wanted to deploy them in my homelab. First, I did it on my first dedicated server, a `Raspberry Pi 5 8Gb` and it worked great! But then I bought a `Dell Poweredge Rack Server` to have a server hosting VMs.

There, I wanted to deploy my websites (my Portfolio and other little projects that could be of interest), but I wanted to keep the VMs really simple and lightweight, so I chose `Alpine Linux`, a great Linux distribution that keep things quite essential. I knew of it because it's one of the most used images with Docker, so it explains its optimization.

But, what if the VM crashes or needs to be rebooted, or, simply, if I need to reboot the Server?

Then I would need to restart manually all webserver in the many VMs... a lot of hassle and possible issues.

The answer to that problematic is to **create a service that is started at boot time**.

In `systemd` distributions, you need to create a **service file**, but `Alpine Linux` uses `OpenRC` instead.

## Creating an `OpenRC` file

`OpenRC` files are located in `/etc/init.d/`, so first, you need to create a file there (the name of the file will be the name of the service):

```
sudo vim /etc/init.d/webserver
```

Then write your `OpenRC` file:

```
#!/sbin/openrc-run
```

```
# Service description
#
# This service starts the web server application
name=$RC_SVNAME
pidfile="/run/${RC_SVNAME}.pid"
# switch webserver to the name of your service in the two following lines (they are both optional configurations)
output_log="/var/log/webserver.log"
error_log="/var/log/webserver.err"

# Command needed to run the application
command="/path/to/your/webserver/executable"
# you can put all arguments below:
command_args="arg1 arg2 arg3"

# Command settings
command_background=true
# specify the user you wish to execute your webserver as:
command_user="www-data:www-data"

# Dependencies (most likely the network if it's a webserver, and there might be more if your application is more
demanding)
depend() {
    need net
}
```

Make the file executable, otherwise, you'll not be able to run anything and the start command will fail:

```
sudo chmod 755 /etc/init.d/webserver
```

### Warning!

You might need to give writing access to the user defined in the `command_user` line for the `output_log` and `error_log` files if you decide to specify them (they are optional)

## Using Environment variables

If you need to use environment variables in your `OpenRC`, you need to create a configuration file in `/etc/conf.d/` with the same name as the `OpenRC` file.

In it you can **export** all environment variables you might need, for example:

```
export WEBSERVER_PORT=4000
```

and then you can call it in the `OpenRC` file, for example:

```
command_args="port=${WEBSERVER_PORT}"
```

# Managing the **service** created

Consult that [Official documentation](#) from the Alpine Linux website if you wish to know more about it from the official page.

To manipulate your service, you can use those three commands (replace `webserver` with the name of the service):

```
rc-service webserver start
```

```
rc-service webserver status
```

```
rc-service webserver stop
```

To enable your service at the `default` level (when booting the system normally):

Replace `webserver` with the name of the service.

```
rc-update add webserver default
```

To remove your service from all levels:

```
rc-update delete webserver -a
```

With that, you should be able to create your services and execute them automatically at boot time in your `Alpine Linux` servers!

Happy coding!