Running a Node.js application using nvm as a systemd service

> This article was originally published at https://gist.github.com/joepie91/73ce30dd258296bd24af23e9c5f761aa.

> Hi there! Since this post was originally written, `nvm` has gained some new tools, and some people have suggested alternative (and potentially better) approaches for modern systems.
>
> Make sure to have a look at the comments on the original Gist, *before* following this guide!

Trickier than it seems.

1. Set up nvm

Let's assume that you've already created an unprivileged user named `myapp`. You should never run your Node.js applications as root!

Switch to the `myapp` user, and do the following:

1. `curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.31.0/install.sh | bash` (however, this will immediately run the nvm installer - you probably want to just download the `install.sh` manually, and inspect it before running it)
2. Install the latest stable Node.js version: `nvm install stable`

2. Prepare your application

Your package.json must specify a `start` script, that describes what to execute for your application. For example:

```
...
"scripts": {
    "start": "node app.js"
},
...
```

3. Service file

Save this as `/etc/systemd/system/my-application.service`:

```
[Unit]
Description=My Application

[Service]
EnvironmentFile=-/etc/default/my-application
ExecStart=/home/myapp/start.sh
WorkingDirectory=/home/myapp/my-application-directory
LimitNOFILE=4096
IgnoreSIGPIPE=false
KillMode=process
User=myapp

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

You'll want to change the `User`, `Description` and `ExecStart`/`WorkingDirectory` paths to reflect your application setup.

4. Startup script

Next, save this as `/home/myapp/start.sh` (adjusting the username in both the path *and* the script if necessary):

```
#!/bin/bash
. /home/myapp/.nvm/nvm.sh
npm start
```

This script is necessary, because we can't load nvm via the service file directly.

Make sure to make it executable:

```
chmod +x /home/myapp/start.sh
```

5. Enable and start your service

Replace `my-application` with whatever you've named your service file after, running the following **as root**:

1. `systemctl enable my-application`
2. `systemctl start my-application`

To verify whether your application started successfully (don't forget to `npm install` your dependencies!), run:

```
systemctl status my-application
```

... which will show you the last few lines of its output, whether it's currently running, and any errors that might have occurred.

Done!

Revision #1
Created 2024-12-11 01:06:03 UTC by joepie91
Updated 2024-12-11 18:44:19 UTC by joepie91