

How To Deploy Node.js Applications Using Systemd and Nginx

Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=274>

This tutorial uses a simple Node.js application, but is applicable to most, if not all, others as well (be they Ruby, Python, etc). For PHP web applications, it is recommended to use a more specialized LAMP or LEMP stack instead.

Commands will be provided for both Fedora and Arch, do keep a lookout for which is which to avoid misconfiguration and/or confusion. When not indicated, the command is the same for both systems. It is also recommended you read through the entire tutorial before attempting it step-by-step, so as to get an idea of what it entails and whether it is appropriate for your situation.

System Preliminaries

—
• **A server with systemd.** Arch Linux and Fedora server instance are configured thus by default, but systemd can be installed on other distributions as well; refer to yours' documentation: [Ubuntu](#), [Debian](#).

•
Nginx, to be used as a reverse-proxy http and websocket server.

•
Git, to install nvm, and to pull your application if using git.

•
Root access. It is also possible to login as a normal user and *sudo all commands*, or to *su -or sudo su - to a root prompt*.

Install packages

—
Arch:

```
# pacman -Sy  
# pacman -S nginx git
```

Fedora:

```
# yum install nginx git
```

Application Preliminaries

—
These are settings you can customise to your liking, but they have to be decided upon and set before

starting.

User

The application will run in its own separate user account. Pick a name, it should relate to the application to make it easy to remember and maintain. Here, `srv-node-sample` is used.

```
# useradd -mrU srv-node-sample
```

Port

To avoid conflicts, pick a high port. Here, "15301" is used.

Application Setup

Start by installing what is necessary for the application to run. For Node.js (and Ruby, Python...), there are two choices: either use the system's runtime, or a user-specific installation (e.g. using [nvm](#), [rbenv](#), [RVM](#), [virtualenv](#), etc).

Using the system node

Arch:

```
# pacman -S nodejs
```

Fedora:

```
# yum install nodejs
```

Using a user-specific install

This has to be installed in the application's home directory, i.e. `/home/srv-node-sample`, which is most easily done by logging in as that user:

```
# su srv-node-sample
```

```
$ cd
$ curl https://raw.githubusercontent.com/creationix/nvm/master/install.sh | sh
$ source ~/.nvm/nvm.sh
$ nvm install 0.10
$ nvm alias default 0.10
```

Then take note of where the node binary is installed:

```
$ which node
/home/srv-node-sample/.nvm/v0.10.22/bin/node
```

Deploy your application

While logged in to `srv-node-sample`, deploy your code. This is an example only, your process will vary.

```
$ git clone git@server.company.tld:user/repo.git .
$ npm install
$ grunt deploy
```

For this tutorial, the following sample application is used:

```
var http = require('http');
http.createServer(function(req, res) {
    res.end('<h1>Hello, world.</h1>');
}).listen(15301);
```

Then return to root:

```
$ exit
```

Nginx Setup

Place this in your server block:

```
location / {
    proxy_pass http://localhost:15301/;
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}
```

Then set up its daemon:

```
# systemctl enable nginx
# systemctl restart nginx
```

Systemd Setup

Create a service file for the application, in `/etc/systemd/system/node-sample.service`.

There's a few variables that need to be filled in:

- [node binary] This is the output of "which node" as the srv-node-sample user. Either /usr/bin/node or the ~/.nvm/. . . path noted above.

- [main file] This is the main file of your application. Here, "index.js".

- Don't forget to replace srv-node-sample!

```
[Service]
ExecStart=[node binary] /home/srv-node-sample/[main file]
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=node-sample
User=srv-node-sample
Group=srv-node-sample
Environment=NODE_ENV=production
[Install]
WantedBy=multi-user.target
```

Now start the service:

```
# systemctl enable node-sample
# systemctl start node-sample
```

Usage

Status

systemctl status node-sample

```
node-sample.service
Loaded: loaded (/etc/systemd/system/node-sample.service; enabled)
Active: active (running) since Fri 2013-11-22 01:12:15 UTC; 35s ago
Main PID: 7213 (node)
CGroup: name=systemd:/system/node-sample.service
        └─7213 /home/srv-node-sample/.nvm/v0.10.22/bin/node /home/srv-nod...
```

```
Nov 22 01:12:15 d02 systemd[1]: Started node-sample.service.
```

~~~~

## Logs

—

~~~~

journalctl -u node-sample

```
-- Logs begin at Thu 2013-11-21 19:05:17 UTC, end at Fri 2013-11-22 01:12:15 UTC. --
Nov 22 01:12:15 d02 systemd[1]: Starting node-sample.service...
Nov 22 01:12:15 d02 systemd[1]: Started node-sample.service.
Nov 22 01:12:30 d02 node-sample[7213]: Sample message from application
~~~~
```

Restart, stop, etc

—

Force a restart:

```
# systemctl restart node-sample
```

Stop the application:

```
# systemctl stop node-sample
```

The application will be automatically restarted if it dies or is killed:

```
# systemctl status node-sample
node-sample.service
  Loaded: loaded (/etc/systemd/system/node-sample.service; enabled)
  Active: active (running) since Fri 2013-11-22 01:12:15 UTC; 35s ago
  Main PID: 7213 (node)
  CGroup: name=systemd:/system/node-sample.service
          â"â"€7213 /home/srv-node-sample/.nvm/v0.10.22/bin/node
/home/srv-nod...
Nov 22 01:12:15 d02 systemd[1]: Started node-sample.service.
# kill 7213
# systemctl status node-sample
node-sample.service
  Loaded: loaded (/etc/systemd/system/node-sample.service; enabled)
  Active: active (running) since Fri 2013-11-22 01:54:37 UTC; 6s ago
  Main PID: 7236 (node)
  CGroup: name=systemd:/system/node-sample.service
          â"â"€7236 /home/srv-node-sample/.nvm/v0.10.22/bin/node
/home/srv-nod...
Nov 22 01:54:37 d02 systemd[1]: node-sample.service holdoff time over, sch...t.
Nov 22 01:54:37 d02 systemd[1]: Stopping node-sample.service...
```

```
Nov 22 01:54:37 d02 systemd[1]: Starting node-sample.service...
Nov 22 01:54:37 d02 systemd[1]: Started node-sample.service.
```

The PID has changed, showing the application has indeed been killed and restarted.

Websockets

—

If the application uses websockets, the following lines have to be added to the Nginx configuration:

```
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection "upgrade";
proxy_http_version 1.1;
```

and Nginx has to be reloaded:

```
# systemctl reload nginx
```