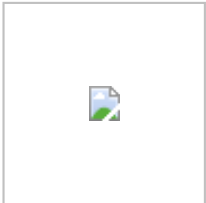


JavaScript (<http://www.sitepoint.com/javascript/>)

Configuring Nginx and SSL with Node.js



(<http://www.sitepoint.com/author/spanda/>)

Sandeep Panda (<http://www.sitepoint.com/author/spanda/>)

Published September 16, 2014

 **Tweet** (<https://twitter.com/Share?Text=Configuring+Nginx+And+SSL+With+Node.Js&Via=Sitepointdotcom>)

Subscribe (<https://confirmsubscription.com/H/Y/1FD5B523FA48AA2B>)

Nginx is a high performance HTTP server as well as a reverse proxy. Unlike traditional servers, Nginx follows an event driven asynchronous architecture. As a result the memory footprint is low and performance is high. If you are running a Node.js based web app you should seriously consider using Nginx as a reverse proxy. Nginx can be very efficient in serving static assets. For all other requests it will talk to your Node.js backend and send the response to the client. In this tutorial we will discuss how to configure Nginx to work with Node.js. We will also see how to setup SSL in the Nginx server.

Installing Nginx

Assuming you already have Node.js installed on your machine let's see how to install Nginx.

Installation on Mac

If you are on Mac you can use [Homebrew \(<http://brew.sh/>\)](http://brew.sh/) to install Nginx easily. The steps are as following:

- Homebrew needs the directory `/usr/local` to be `chown` 'd to your username. So, run the following command in terminal first:

```
1 | sudo chown -R 'username here' /usr/local
```

- Now the following two commands will install Nginx on your system.

```
1 | brew link pcre
2 | brew install nginx
```

- Once the installation is complete you can type the following command to start Nginx:

```
1 | sudo nginx
```

- The Nginx config file can be found here: `/usr/local/etc/nginx/nginx.conf` .

Installation on Ubuntu

If you are running Ubuntu you can use the following command to install Nginx:

```
1 | sudo apt-get update
2 | sudo apt-get install nginx
```

Once Nginx is installed it will start automatically.

Installation on Windows

For Windows, head over to the Nginx [downloads \(http://nginx.org/en/download.html\)](http://nginx.org/en/download.html) page and get the zip. The next step is unzipping the archive and moving to the directory in the command prompt as following:

```
1 | unzip nginx-1.3.13.zip
2 | cd nginx-1.3.13
3 | start nginx
```

As you can see the command `start nginx` will start Nginx.

Now that the installation is done, let's see how you can configure a simple server.

Setting up a Node.js Server

First, let's create a simple Node.js server. If you want you can download a simple Express based app from [here \(https://github.com/jsprodotcom/source/blob/master/nodejs-nginx-ssl-demo-app.zip\)](https://github.com/jsprodotcom/source/blob/master/nodejs-nginx-ssl-demo-app.zip). Once you have the code, extract it and move to the `demoApp` directory in your terminal. Next, enter the following commands to start the server on port 3000.

```
1  npm install
2  node bin/www
3  1
4
5  <h2 id="configuring-nginx">Configuring Nginx</h2>
6  <p>Now let's open up Nginx config file. As I am on Mac, I can just use nano to start editing
7
8  1
9  nano /usr/local/etc/nginx/nginx.conf
```

If you want, you can go directly to the directory and open up the config file with your favorite text editor. As you scroll down you will find a `server` block. It looks something like this:

```
1  server {
2      listen      8080;
3      server_name localhost;
4
5      ....
6      more config goes here
7  }
```

Next, we will configure the server block to suit our needs. We want to serve static files from our website with the help of Nginx and pass all other requests to our Node.js backend. So, replace the above `server` block with a new block as shown below:

```
1  server {
2      listen      8080;
3      server_name localhost;
```

```
4
5     location / {
6         proxy_pass http://localhost:3000; (http://localhost:3000;)
7         proxy_http_version 1.1;
8         proxy_set_header Upgrade $http_upgrade;
9         proxy_set_header Connection 'upgrade';
10        proxy_set_header Host $host;
11        proxy_cache_bypass $http_upgrade;
12    }
13
14    location /public {
15        root /usr/local/var/www;
16    }
17
18 }
```

As you can see our Nginx server listens on `http://localhost:8080` . The `location /` block tells Nginx what to do when any request comes. Inside the `location` block we use `proxy_pass` to point to the Node.js backend which is `http://localhost:3000` in our case.

Now we need another block `location /public` to tell Nginx how to serve static assets. Inside this `location` block we set the root to `/usr/local/var/www` . You can choose a different directory if you like. As a result whenever there is a request like `http://localhost:8080/public/somepath/file.html` Nginx will serve the file from `/usr/local/var/www/public/somepath/file.html` .

At this point you should save the file and type the following to restart Nginx:

Mac:

```
1 | sudo nginx -s stop && sudo nginx
```

Ubuntu:

```
1 | sudo service nginx restart
```

Or

```
1 | sudo /etc/init.d/nginx restart
```

Windows:

```
1 | nginx -s reload
```

Next, let's serve our stylesheet from Nginx instead of Node.js. Our Node.js template uses a stylesheet with the URL `/public/stylesheet/style.css`. Create a file named `style.css` inside the directory `/usr/local/var/www/public/stylesheet` and Nginx will serve it correctly. For example, you can put the following CSS in `style.css`:

```
1 | body {
2 |     padding: 50px;
3 |     font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;
4 | }
5 |
6 | a {
7 |     color: #00B7FF;
8 | }
```

Once this is done you can go to `http://localhost:8080` to see our app in action. Although you are accessing the Nginx server you will get the actual response from the Node.js backend while only the stylesheet will be served by Nginx.

Setting Up SSL

On a production website many times you will need to configure SSL to protect sensitive information. Usually you will go to a CA (Certificate Authority) and get a certificate issued. You can also create a self signed certificate. The only problem is that browsers will show a warning that the "Certificate is not Trusted" when someone visits your website. But for testing on your local machine that's perfectly fine. Here is a nice [tutorial](https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-nginx-for-ubuntu-12-04) (<https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-nginx-for-ubuntu-12-04>) that shows how to self sign your SSL certificate.

Once you have the certificate and a private key you can setup SSL in Nginx. You need to modify our previous server block to the following:

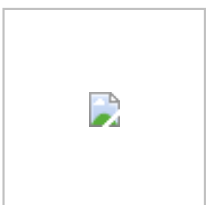
```
1 | server {
2 |     listen      8080;
3 |     listen      443 ssl;
```

```
4     server_name  localhost;
5
6     ssl_certificate  /etc/nginx/ssl/server.crt
7     ssl_certificate_key /etc/nginx/ssl/server.key
8
9     location / {
10         proxy_pass http://localhost:3000; (http://localhost:3000;)
11         proxy_http_version 1.1;
12         proxy_set_header Upgrade $http_upgrade;
13         proxy_set_header Connection 'upgrade';
14         proxy_set_header Host $host;
15         proxy_cache_bypass $http_upgrade;
16     }
17
18     location /public {
19         root /usr/local/var/www;
20     }
21
22 }
```

That's it! Now if you access `https://localhost:8080` your SSL set up will work. Here we assume that the certificate file and private key are located at `/etc/nginx/ssl/server.crt` and `/etc/nginx/ssl/server.key` respectively. You can change these locations if you want.

Conclusion

In this article, we saw how to use Nginx as a reverse proxy for our Node.js app and configure SSL. Letting Nginx work in the front and handle the static resources you can definitely take some load off your Node backend. Give it a try and do let us know your questions/suggestions through comments.



[\(http://www.sitepoint.com/author/spanda/\)](http://www.sitepoint.com/author/spanda/)

Sandeep Panda (<http://www.sitepoint.com/author/spanda/>)

Sandeep is the Co-Founder of devmag.io (<https://devmag.io>). He loves startups and web technologies.

[!\[\]\(35e4f762fc1cfea5610d92e2d225d5b4_img.jpg\) \(https://twitter.com/Sandeepg33k\)](https://twitter.com/Sandeepg33k)


[!\[\]\(d84e7ea36f695d92cb39ec32c307ac93_img.jpg\) \(http://facebook.com/sandeep.panda92\)](http://facebook.com/sandeep.panda92)

[!\[\]\(feabb98897b440bc8695a03336a6e2df_img.jpg\) \(http://www.linkedin.com/pub/sandeep-panda/45/768/b23\)](http://www.linkedin.com/pub/sandeep-panda/45/768/b23)

[!\[\]\(9dfdaff1d86ba3c1f8353b4d1b61b8c5_img.jpg\) \(https://plus.google.com/+SandeepPanda\)](https://plus.google.com/+SandeepPanda)

You might also like:

[Build a Node.js-powered Chatroom Web App: Connecting with WebSockets](http://www.sitepoint.com/build-node-js-powered-chatroom-web-app-connecting-websockets/) (<http://www.sitepoint.com/build-node-js-powered-chatroom-web-app-connecting-websockets/>) 

[Book: Jump Start JavaScript](https://learnable.com/books/jump-start-javascript?utm_source=sitepoint&utm_medium=related-items&utm_content=js-javascript) (https://learnable.com/books/jump-start-javascript?utm_source=sitepoint&utm_medium=related-items&utm_content=js-javascript) 

[Configuring Nginx to Speed Up Your Ghost.io Blog](http://www.sitepoint.com/configuring-nginx-speed-ghost-blog/) (<http://www.sitepoint.com/configuring-nginx-speed-ghost-blog/>) 



Free *JavaScript: Novice to Ninja* Sample

Get a free 32-page chapter of *JavaScript: Novice to Ninja* and receive updates on exclusive offers from SitePoint.

email address

Claim Now

8 Comments

SitePoint

 **dongguangming** ▾

 Recommended 1  Share

Sort by Best ▾



Join the discussion...



ambled • 7 months ago

nice and concise, however I don't think you meant to include the port number 8080 in the reference to `https://localhost`

[startssl.com](#) also issues free SSL certificates

2 ^ | ▾ • Reply • Share ›



Sharry • a month ago

Can you use nginx to serve static content from a node site please?

^ | ▾ • Reply • Share ›



s.shivasurya • 4 months ago

@sandeep doing this affects performance of app ? can u explain how to do vhost in Nginx?

^ | ▾ • Reply • Share ›



Nisim Joseph • 7 months ago

you have an error on the config of "location" to static items.

you need to use location /public

```
{
alias [folder location];
}
not "root".
```

inside a "server" -> "location" it only alias.

in the server level it "root".

^ | ▾ • Reply • Share ›



ruffiem • 7 months ago

`sudo nginx -s reload` works on mac as well :) nice topic, nice post!

^ | ▾ • Reply • Share ›



Guest → **ruffiem** • 4 days ago

This looks like it allows the server to be accessed over HTTP or HTTPS. How would I

▼ This looks like it allows the server to be accessed over HTTP or HTTPS. How would I redirect the user to only use HTTP?

^ | v • Reply • Share ›



Guest → Guest • 4 days ago

Ooops. I attached this to the wrong person...

^ | v • Reply • Share ›



cwołoszynski • 4 days ago

This looks like it proxies for both 8080 and 443 and routes them to the back end. I want to use nginx to redirect the 8080 to 443 and then only proxy the 443 to the node.js. Can you suggest how to do that?

^ | v • Reply • Share ›

About

[About us \(/about-us/\)](/about-us/)

[Advertise \(/advertising/\)](/advertising/)

[Press Room \(/press/\)](/press/)

[Legals \(/legals/\)](/legals/)

[Feedback \(mailto:feedback@sitepoint.com\)](mailto:feedback@sitepoint.com)

[Write for Us \(/write-for-us/\)](/write-for-us/)

Our Sites

[Learnable \(https://learnable.com\)](https://learnable.com)

[Reference \(http://reference.sitepoint.com\)](http://reference.sitepoint.com)

[Web Foundations \(/web-foundations/\)](/web-foundations/)

Connect



[\(/feed\)](/feed/) [\(/newsletter\)](/newsletter/) [\(/https://www.facebook.com/sitepoint\)](https://www.facebook.com/sitepoint)



[\(/http://twitter.com/sitepointdotcom\)](http://twitter.com/sitepointdotcom)

[\(/https://plus.google.com/+sitepoint\)](https://plus.google.com/+sitepoint)