



Utkarsh Verma

[Follow](#)

Tech @hackinoutco

Feb 2 · 3 min read

# Configure Nginx as a web server and reverse proxy for Nodejs application on AWS Ubuntu 16.04 server

## Introduction

Nginx was created in response to [C10k challenge](#) for handling at least 10,000 simultaneous client connections on a single server. Nginx uses asynchronous, event-driven architecture to handle these massive amount of connections. This architecture makes handling high and fluctuating loads much more predictable in terms of RAM usage, CPU usage, and latency.

Nginx can be used as web server, reverse proxy, load balancer and HTTP cache.

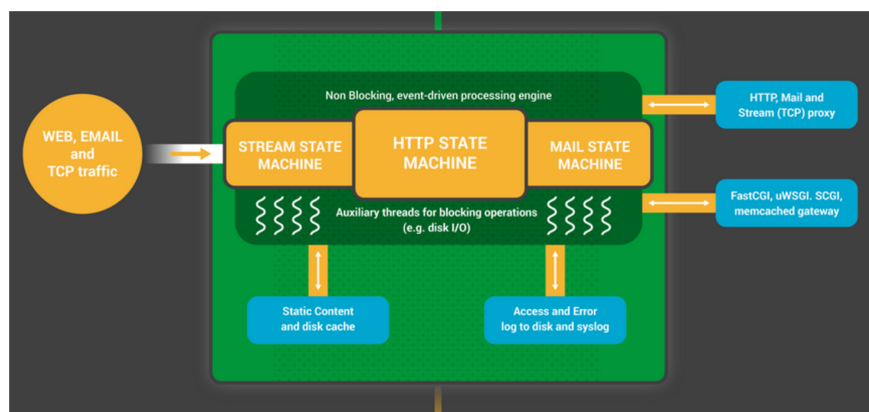


Image Credits

## Installing Nginx on AWS ec2 instance with Ubuntu 16.04

```
$ sudo apt-get update && sudo apt-get upgrade -y  
$ sudo apt-get install nginx -y
```

## Setup Nginx

Check status of Nginx and start it using the following commands:-

```
$ sudo systemctl status nginx    # To check the status of  
nginx  
$ sudo systemctl start nginx     # To start nginx
```

Make sure that Nginx will run on system startup by using command below:-

```
$ sudo systemctl enable nginx
```

Nginx should be up and running, now we need to setup Nodejs on server.

## Setup Nodejs

Check if Nodejs and npm are already installed on server by using

```
$ node --version # For checking nodejs version  
$ npm --version  # For checking npm version
```

*If you get the Nodejs and npm versions as output then it's already installed, otherwise follow the instructions below to install Nodejs on ec2.*

Install Nodejs using the following commands:-

```
$ sudo apt-get update  
$ sudo apt-get install nodejs
```

Now install npm using the following command:-

```
$ sudo apt-get install npm
```

Verify Nodejs and npm installations using below commands

```
$ node --version  
$ npm --version
```

*It should return the Nodejs and npm versions installed .*

### Setting up Nginx as a reverse proxy for Nodejs application

Get the private ip address of your instance by using the command below, do make sure you are logged into your instance using ssh:-

```
$ wget -q -O - 'http://169.254.169.254/latest/meta-data/local-ipv4'
```

Remove and add default file to **sites-available** by using following commands:-

```
$ sudo rm /etc/nginx/sites-available/default  
$ sudo vi /etc/nginx/sites-available/default
```

Now start the server block and start adding code. It should look something like this:-

```
server {  
    listen 80;  
  
    server_name your_domain.com;  
  
    location / {  
        proxy_pass http://private_ip_address:8080;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
}
```

Replace **your\_domain.com** with your ec2 domain name and **private\_ip\_address** with the private ip address associated ( which you obtained previously)

Save and quit file using **wq** vim command.

Test the configuration of Nginx using the command below:-

```
$ sudo nginx -t
```

Then reload Nginx if **OK** is displayed.

```
$ sudo /etc/init.d/nginx reload
```

## Create your first Nodejs application using Expressjs

Create your project directory and install Expressjs

```
$ mkdir myapp  
$ cd myapp  
$ npm install express
```

Create a simple express application

```
$ vi app.js
```

and make it listen at port **8080** and **private\_ip\_address**. It should look something like this:-

```
var express = require('express');  
var app = express();  
  
app.get('/', function(req, res){  
  res.send("Hello World!");  
});
```

```
app.listen(8080, 'private_ip_address');
```

Replace **private\_ip\_address** with your private ip address.

Run your app by using the command below:-

```
$ node app.js
```

Now open your browser and browse your domain. It should display **Hello World!**.

## Install PM2 Advanced, production process manager for Nodejs

```
$ sudo npm install pm2 -g
```

Run your application using **pm2** to make sure your application runs automatically when the server restarts.

```
$ pm2 start app.js
```

Voila!!!

Nginx is now setup as a reverse proxy for your application.

Good luck with building your application.

Do let me know if you face any issues during the configuration :)

I am available at [Utkarsh Verma](#)

. . .

*Stay tuned for future write-ups that will cover more detailed info of my dev setup.*



