Node.js

Node.js is a server side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.

Official Site: <https://nodejs.org/en/>

INSTALLATION:

curl -sL https://deb.nodesource.com/setup\_5.x | sudo -E bash -

sudo apt-get install -y nodejs

Source Link:

1. <https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions>
2. <https://www.digitalocean.com/community/tutorials/how-to-install-node-js-on-an-ubuntu-14-04-server>

NPM:

npm makes it easy for JavaScript developers to share and reuse code, and it makes it easy to update the code that you're sharing.

npm install npm -g

Source: https://docs.npmjs.com/getting-started/what-is-npm

Express:

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Source: <http://expressjs.com/>

[Features](https://www.npmjs.com/package/express#features):

* Robust routing
* Focus on high performance
* Super-high test coverage
* HTTP helpers (redirection, caching, etc)
* View system supporting 14+ template engines
* Content negotiation
* Executable for generating applications quickly

Source: <https://www.npmjs.com/package/express>

npm install -g express

npm -g install will install the express and supervisor modules from npm software repository and make it available to the whole system. The -g switch in this command means "global" - the express and supervisor commands will be available accross the whole system.

Error: Error: Cannot find module 'express'

Solution: use npm link to resolve modules

npm link express

**Installing NODE-MYSQL:**

node-mysql can be installed via npm# npm install node-mysql

HTTPS and SSL configuration:

To create an HTTPS server, you need two things: an SSL certificate, and Node's built-in https module.

We need to start out with a word about SSL certificates. Speaking generally, there are two kinds of certificates: those signed by a 'Certificate Authority', or CA, and 'self-signed certificates'. A Certificate Authority is a trusted source for an SSL certificate, and using a certificate from a CA allows your users to be trust the identity of your website. In most cases, you would want to use a CA-signed certificate in a production environment - for testing purposes, however, a self-signed certicate will do just fine.

Source: <https://docs.nodejitsu.com/articles/HTTP/servers/how-to-create-a-HTTPS-server>

openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365

Source: <http://blog.mgechev.com/2014/02/19/create-https-tls-ssl-application-with-express-nodejs/>

Error: SSL\_CTX\_use\_PrivateKey\_file("/etc/nginx/ssl/server.key") failed (SSL: error:0906D06C:PEM

openssl rsa -in key.pem -out newkey.pem && mv newkey.pem key.pem

Source: http://blog.mgechev.com/2014/02/19/create-https-tls-ssl-application-with-express-nodejs/

PM2:

P(rocess) M(anager) 2

Production process manager for Node.JS applications with a built-in load balancer.

PM2 is a production process manager for Node.js applications with a built-in load balancer. It allows you to keep applications alive forever, to reload them without downtime and to facilitate common system admin tasks.

Source: <https://www.npmjs.com/package/pm2>

Ref: <http://pm2.keymetrics.io/docs/usage/pm2-doc-single-page/>

1. It is going to keep your site up by restarting the application if it crashes. These crashes should NOT happen, but it is good know that PM2 has your back. (Some people may be aware of Forever.js, another tool that is used to keep node based sites running - I think you will find that PM2 has a lot to offer.)
2. It is going to help you by restarting your node application as a service every time you restart the server. Some of use know of other ways to do this, but pm2 makes it easier, and it has some added flexibility.

Install PM2 by typing thr following at the command line:

sudo npm install pm2 -g

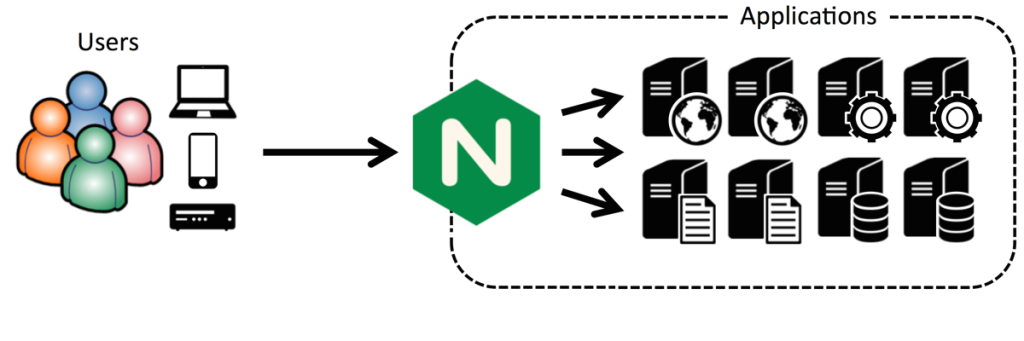
Nginx:

Nginx (pronounced "engine x") is a web server. It can act as a reverse proxy server for HTTP, HTTPS, SMTP, POP3, and IMAP protocols, as well as a load balancer and an HTTP cache.

apt-get install Nginx

Created by Igor Sysoev in 2002, Nginx runs on Unix, Linux, BSD variants, OS X, Solaris, AIX, HP-UX, and Windows.Released under the terms of a BSD-like license, Nginx is free and open source software.

1. Implement a reverse proxy server
2. Cache static files
3. Balance loads across multiple servers
4. Proxy WebSocket connections
5. Implement SSL/TLS and HTTP/2



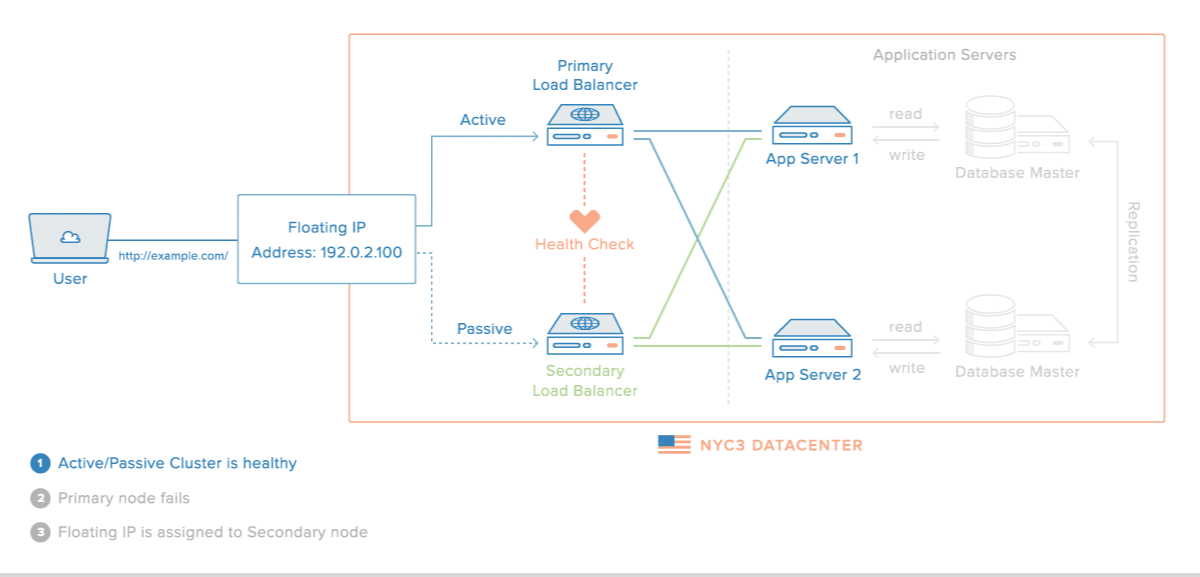
Source: <https://www.nginx.com/blog/5-performance-tips-for-node-js-applications/>

Nginx HTTP Proxying, Load Balancing:

Nginx proxying can be made more robust by adding in a redundant set of load balancers, creating a high availability infrastructure.

A high availability (HA) setup is an infrastructure without a single point of failure, and your load balancers are a part of this configuration. By having more than one load balancer, you prevent potential downtime if your load balancer is unavailable or if you need to take them down for maintenance.

Here is a diagram of a basic high availability setup:



Source Link: <https://www.digitalocean.com/community/tutorials/understanding-nginx-http-proxying-load-balancing-buffering-and-caching>

Note: Nginx Plus Demo Page Link: http://demo.nginx.com/status.html?\_ga=1.231644952.609420999.1462249673#