Deploying Node.js

With Forever and Nginx

“a no-downtime tale”

Node.js

- Runs as a standalone server, listen its own port
- Just need to proxy the request to Node.js and relay the result back
- Installing is easy (let's try in MacOS/Ubuntu)
  - sudo apt-get update
    sudo apt-get install python-software-properties python g++ make
    sudo add-apt-repository ppa:chris-lea/node.js
    sudo apt-get update
    sudo apt-get install nodejs

Sample server with Compound.js

- sudo npm install compound -g
- compound init todo-list-app
- cd todo-list-app && npm install

How to start the project:

- node .

Making Node.js run "forever"

- Using Forever.js
- Forever.js is a "daemonized version" of your Node app.
- Will check if the app is killed and re-spawn it

Forever.js setup and commands

- It's an npm package.
  - sudo npm install forever -g
- Main commands
  - start - **Start SCRIPT as a daemon**
  - stop - **Stop the daemon SCRIPT**
  - stopall - **Stop all running forever scripts**
  - restart - **Restart the daemon SCRIPT**
  - restartall - **Restart all running forever scripts**
  - list - **List all running forever script**

Let's try it!

- Identify the script to run
  - Inspecting package.json is often a good idea
  - Example with compound.js
    - , **"main": "server.js"**
  - Command: "node ." loads package.json, finds "main" script and executes it.
- Run on Forever:
  - forever start /full-path-to-script/server.js

Ok, a bit more of Forever.js

(featuring up and inspecting)

- How to keep the log
  - -l LOGFILE - Logs the forever output to LOGFILE
  - -o OUTFILE - Logs stdout from child script to OUTFILE
  - -e ERRFILE - Logs stderr from child script to ERRFILE
- Some other useful commands
  - -p PATH - Base path for all forever related files (pid files, etc.)
    - To make your line shorter
  - -c COMMAND - COMMAND to execute (defaults to node)
    - For when you have "more" than just "node" to run (server init params for example)
- List running processes
  - forever list (easy, right?)

Now, the cherry, "no downtime"

- **forever restart /full-path-to-script/script**

- If you're not sure about the script path, just do "**forever list**" and take it up from there

Now we're all set, let's get serious

- Nginx
  - Awesome
  - Event-driven
  - Asynchronous
  - 10k-friendly
- Not quite hard to setup (ubuntu)
  - **sudo apt-get install nginx**

Configure Nginx

- Sites management
  - Config files are best kept per-site
  - 1 file per site in
    - **/etc/nginx/sites-available/** (standard)
  - To activate a site, make a symbolic link to
    - **/etc/nginx/sites-enabled/**
  - Restart gracefully
    - **sudo service nginx reload**
  - Done.

Let's configure our proxy

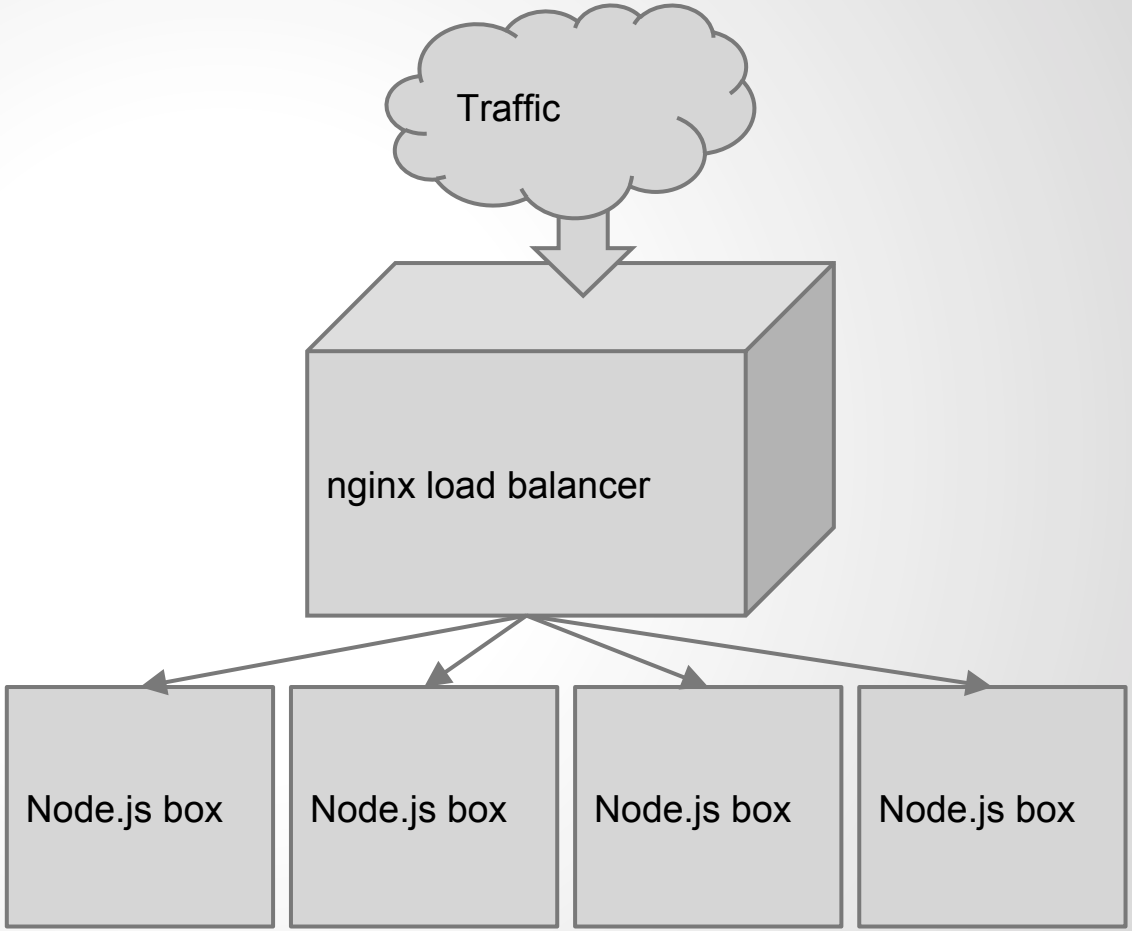
- cd /etc/nginx/sites-available/
- sudo view compoundapp.com
  - (name it based on your FQDN as a best practice)
- Create a proxy block
  - upstream node_boxes {

        server 127.0.0.1:3000;

    }
- Proxy Will be used as the relay

Architecture

- Traffic goes through the load balancer
- nginx redirects to the "upstream" node_box
- Want to scale? Just add more boxes and update nginx config.



Configs: Load balancer

```
server {
#  server_name yourdomain.com yourdomain; # set the domain name to bind to
       access_log /var/log/nginx/compountapp.com_access.log;     error_log /var/log/nginx/
   compountapp.com_error.log;
       # reroute all and pass the correct headers
       location / {
         proxy_set_header X-Real-IP $remote_addr;
         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         proxy_set_header Host $http_host;
         proxy_set_header X-NginX-Proxy true;

             # then proxy the request
         proxy_pass http://node_boxes/;
         proxy_redirect off;
       }
}
```

Load balancing?

- upstream node_boxes {
  server 50.20.14.8:3000;
  server 50.20.14.9:3001;
  server 50.20.14.10:3002;
  server 50.20.14.11:3000;
  }
- **sudo service nginx reload**

What's next?

- WebSockets
- Load balancing
- What about the Databases/services?
- Centralize your logs
- Performance analysis
- Cloud

Resources

- http://stackoverflow.com/a/5015178/519649
  - Config node.js as a proxy
- https://github.com/nodejitsu/forever
- https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager
  - Install Node.js via package manager

About us

Author: **Nicolas Embleton**

Find me at: **nicolas.embleton@gmail.com**

Presentation made for “**Javascript Ho Chi Minh City**” meetup group

You can find us at:

- http://meetup.com/JavaScript-Ho-Chi-Minh-City/
- https://www.facebook.com/JavaScriptHCMC
- https://plus.google.com/communities/116105314977285194967