



AWS Deployment

Part 1: Get prepared

Core Setup Requirements

1. An AWS account
2. A GitHub or bitbucket account
3. An internet connection
4. Git should be installed locally

Get going

1. Create a local, functional project
2. Keep that version, version controlled via git - good practice anyway.

Make a GitHub/bitbucket repository

1. Push your project to that GitHub/bitbucket repository

Part 2: Set up AWS

1. Enter AWS, and click launch new instance.
2. Select Ubuntu 16.04 LTS
3. Select t2.micro
4. Set security settings:
 - ssh 0.0.0.0, (Anywhere or myIP)
 - http 0.0.0.0 (Anywhere)
 - https 0.0.0.0 (Anywhere, or don't set it)
5. Download a .pem key from AWS
6. Move the .pem file to an appropriate folder on your system
7. Change user permission on .pem `chmod 400 {{mypem}}.pem`

For PC, that command might require kitty or putty or bash terminal

We are now ready to enter the cloud server!

Part 3: Enter cloud server

Navigate to the folder where your .pem file is!

(you can use the 'connect' button on Amazon AWS to get the next line of code)

For PC, the command below might require kitty or putty or bash terminal

1. `ssh -i {{mypem}}.pem ubuntu@{{yourAWS.ip}}` again you can get this exact line of code from [Connect](#) on AWS. (The command is found below **example**).

2. In the ubuntu terminal: These establish some basic dependencies for deployment and the Linux server.

```
sudo apt-get update
sudo apt-get install -y build-essential openssl libssl-dev pkg-config
```

3. In the ubuntu terminal, one at a time because they require confirmation: (these install basic node and npm)

```
sudo apt-get install -y nodejs nodejs-legacy
```

Note: In case this does not work, try **sudo apt install nodejs-legacy** instead.

```
sudo apt-get install npm
sudo npm cache clean -f
(The cache clean -f, forcibly cleans the cache. This will give an interesting comment:))
```

4. In the ubuntu terminal: These install the node package manager **n** and updated node.

```
sudo npm install -g n
sudo n stable (or whichever node version you want e.g. 5.9.0)
```

5. `node -v` should give you the stable version of node, or the version that you just installed.

6. Install NGINX and git:

```
sudo apt-get install nginx
sudo apt-get install git
```

7. Make your file folder:

```
sudo mkdir /var/www
```

8. Enter the folder:

```
cd /var/www
```

9. Clone your project:

```
sudo git clone {{your project file path on github/bitbucket}}
```

At this point, you should be able to change directories into your project and run your server. It will most likely fail, because of not having mongod up and running, but running the project should be as simple as `node server.js` or a similar command like `npm start`.



BACK TO TRACKS

AWS Deployment • 100%
(/m/14/3710)

AWS Deployment (/m/

Part 4: Set up NGINX

1. Go to nginx's sites-available directory :

```
cd /etc/nginx/sites-available
```

2. Enter vim:

```
sudo vim {{project_name}}
```

vim is a terminal-based text editor for more info see: vim-adventures.com/ (<http://vim-adventures.com/>) or other vim learning tools. The key commands for us are i which allows us to type, esc which turns off insert and then after esc :wq which says write and quit.

3. Paste and modify the following code into vim after hitting i:

```
server {  
    listen 80;  
    location / {  
        proxy_pass http://{{PRIVATE-IP}}:8000;  
        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;  
    }  
}
```

This code says: have the reverse proxy server (nginx) listen at port 80. When going to root /, listen for http requests as though you were actually http:// your private ip and the port your server is listening e.g @8000 or @6789 etc.

Learn more from nginx: http://nginx.org/en/docs/http/nginx_http_proxy_module.html
(http://nginx.org/en/docs/http/nginx_http_proxy_module.html)

4. Remove the defaults from /etc/nginx/sites-available

```
sudo rm default
```

5. Create a symbolic link from sites-enabled to sites available:

```
sudo ln -s /etc/nginx/sites-available/{{project_name}} /etc/nginx/sites-enabled/{{project_name}}
```

- 6 Remove the defaults from /etc/nginx/sites-enabled/

```
cd /etc/nginx/sites-enabled/  
sudo rm default
```

Part 5: Project Dependencies and PM2

1. Install pm2 globally (<https://www.npmjs.com/package/pm2.5> (<https://www.npmjs.com/package/pm2.5>)) (<https://www.npmjs.com/package/pm2> (<https://www.npmjs.com/package/pm2>)). This is a production process manager that allows us to run node processes in the background.

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

2. Try some stuff with pm2!

```
cd /var/www/{{project_name}}
pm2 start server.js
pm2 stop 0
pm2 restart 0
sudo service nginx reload && sudo service nginx restart
```

Probably not quite working yet but close

3. You might have some components that you still need to install: (get your dependencies from npm (assuming your git project has a package.json))

```
sudo npm install
```

- IF USING BOWER (assuming you have a bower.json)

```
sudo npm install bower -g
sudo bower install --allow-root
```

Part 6: MongoDB

1. The last thing, setting up mongodb!
2. Set up a key

```
#version 16.04
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv OC49F3730359A14518585931BC711F9BA15703C6
```

3. Setup mongodb in a source list

```
#version 12.04
echo "deb [ arch=amd64 ] http://repo.mongodb.org/apt/ubuntu precise/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.
#version 14.04
echo "deb [ arch=amd64 ] http://repo.mongodb.org/apt/ubuntu trusty/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.2.lis
#version 16.04
echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-
```



That is all one line!

4. Reupdate to integrate mongo

```
sudo apt-get update
```

5. install mongo

```
sudo apt-get install -y mongodb-org
```

6. Start mongo (probably already started)

```
sudo service mongod start
```

7. Restart your pm2 project and make sure the nginx config's are working:

```
pm2 stop 0  
pm2 restart 0  
sudo service nginx reload && sudo service nginx restart
```

8. At this point, the nginx commands should have shown 2 OKs and you should be off and running. Go to the AWS public IP and see your site live!

[Privacy Policy](#)

To report a mistake, highlight the selection you believe is in error.

DONE