Deploying Node.js Applications with nginx and PM2

But first: SSH Keys

(Do this once per server.)

Why?

- You will `git clone` your code to your AWS EC2 server
- To get updates: `git pull origin master`
- If needed: `git push origin master`

Generate ssh keys on AWS

SSH into your AWS server and run this command:

```
ssh-keygen -t rsa -b 4096 -C "your_github_email@example.com"
```

Then, output the public key to the screen and copy it:

```
cat ~/.ssh/id_rsa.pub
```

Paste the public key as a new key in github.com

Now, you can clone your project

```
git clone <the clone URL of your project>
```

And pull/push changes

```
git pull origin master
git push origin master
```

pm2: a Node process manager

pm2 is

- a program that manages Node.js programs
- can run them when the server (re)boots

Install the pm2 module *globally*

Do installation once per server

```
npm install -g pm2
```

Starting an app

Do this once per app.

```
cd your-project-name
pm2 start bin/www
```

Replace `bin/www` with the name of your main .js file.

Viewing the list of running apps

```
pm2 list
```

Stopping an app

```
pm2 stop app-name
```

The app-name is the one shown in `pm2 list`

Deleting an app

```
pm2 delete app-name
```

Reloading an app

```
pm2 restart app-name
```

Starting the app with a custom name

```
pm2 start bin/www --name "awesomeness"
```

Viewing the Process Manager

```
pm2 monit
```

Setting up a startup script

Do this once per app.

```
pm2 startup systemd
```

`systemd` manages processes on your Linux box

Save the startup script

Do this once per app.

```
pm2 save
```

This should start pm2 when the system is rebooted

nginx

What is nginx?

- HTTP Server
- Reverse Proxy
- Russian

Installing nginx

Do this once per server.

```
sudo apt-get update
sudo apt-get install nginx
```

Configuring nginx

Do this once per app.

```
sudo nano /etc/nginx/sites-available/default
```

Setting up a Reverse Proxy from nginx to your Node app

```
location / {
  proxy_pass http://localhost:3000;
  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;
}
```

Checking for syntax errors

```
sudo nginx -t
```

Restarting nginx to reload the config

Do this only once, after setting up your app.

EC2

```
sudo systemctl restart nginx
```

BONUS: Installing Postgresql

Install Postgresql

Do this once per server.

```
sudo apt-get install postgresql -y
```

Change to the Postgres user

Do this any time you need to create a database or a user.

```
sudo su - postgres
```

In the terminal, as postgres: Create a user

Do this once per server

```
createuser ubuntu
```

In the terminal, as postgres: Create a database

Do this once per app.

```
createdb dbname
```

In the terminal, as postgres: Open a psql shell

Do this any time you need to set a postgres password or change permissions

```
psql
```

In psql: Set a user's password

```
alter user <username> with encrypted password '<password>';
```

Note: leave off the <> symbols

In psql: Give a user permissions for a database

```
grant all privileges on database <dbname> to <username>;
```

Note: leave off the <> symbols