

# CODING FOR ENTREPRENEURS

MERN SERVER  
SETUP

# TABLE OF CONTENTS

<b>Overview</b>	<b>3</b>
Requirements for this guide	3
Purchase your domain	3
Buy a server on Digital Ocean	3
Connect to server using OS X Terminal	3
Basic Server Setup - Ubuntu 18.04	3
Connect to server using Transmit FTP	3
Web App Configuration	3
Test server configuration	3
<b>Purchase a domain</b>	<b>4</b>
Buy a domain on Dotster	4
<b>Purchase a server</b>	<b>5</b>
Buy a server on Digital Ocean	5
<b>Connect to the server</b>	<b>6</b>
Connect to server using OS X Terminal	6
<b>Initial server setup</b>	<b>7</b>
Basic Server Setup - Ubuntu 18.04	7
<b>Setup FTP</b>	<b>8</b>
Connect to server using Transmit FTP	8
<b>Web App Configuration</b>	<b>9</b>
Configure server for web app overview	9

Create a www folder in srv	9
Setup web folder for backend	9
Setup web folder for frontend	9
Install NVM, node version manager	9
Install node using nvm	10
Install required backend packages using npm	10
Run the backend server	10
Run MongoDB database	10
Check backend server accessible and working	11
Install required frontend packages using npm	11
Check frontend server accessible and working	11
Setup pm2	12
Start front end server using pm2	12
Install and setupl nginx	12
Setup main domain	12
Setup reverse proxies in nginx (continue nginx setup)	13
Setup subdomain	13
Setup SSL	14
<b>Testing</b>	<b>15</b>
Test server configuration Add a server	15

# OVERVIEW

## Requirements for this guide

- A Mac computer
- ~ \$5 for a server
- ~ 10 for a domain name
- Transmit FTP

## Purchase your domain

You will need a domain name so that other people will be able to access it via a web address. Any domain name service should work, but this guide uses Dotster.

## Buy a server on Digital Ocean

Once you have your domain name you will need a server to point the domain (web address) to. Digital Ocean provides low cost servers that work great for commercial sites or practicing web application development.

## Connect to server using OS X Terminal

The server you purchase from Digital Ocean will be an Ubuntu server. This type of server does not have a graphical user interface like you normally see in Windows 10 or Mac OS, so you will need to access it from the command line on your Mac. The application on your Mac used to access the server remotely via command line is called the Terminal.

## Basic Server Setup - Ubuntu 18.04

The servers from Digital Ocean are bare bones/ servers that have not been configured at all. You will need to set up basic configuration for these servers.

## Connect to server using Transmit FTP

FTP is a communication method that is used often used to copy files to a remote server. We will configure an FTP application called Transmit to connect to your server, and later copy the web application files to the server.

## Web App Configuration

This is where install required tools and applications, and configure settings so that your app may be accessed via the internet.

## Test server configuration

Once everything is up and running we need to test the configuration. Your app should be accessible using the domain name you purchased from Dotster.

# PURCHASE A DOMAIN

## Buy a domain on Dotster

- Visit [www.dotster.com](http://www.dotster.com)
- Purchase a domain name
  - Make sure to purchase privacy protection

# PURCHASE A SERVER

## Buy a server on Digital Ocean

- Select Droplet
- Select MarketPlace
- Select MongoDB, Ubuntu 18.04
- Select \$5 or \$10 droplet, under starter
- SSH keys for authentication
- Select New SSH, or an existing SSH key saved in DigitalOcean
  - New SSH Key **\*\* in progress\*\***
  - Existing SSH Key
    - If you name it something generic like digitalocean\_ssh\_key, it will make more sense to reuse that key for other servers so you are not using multiple keys.
- For more on ssh:
  - <https://cloudsupport.digitalocean.com/s/>
  - <https://www.digitalocean.com/docs/droplets/how-to/add-ssh-keys/>
  - Enable backups since they are only \$1 a month

# CONNECT TO THE SERVER

## Connect to server using OS X Terminal

- Copy the IP of the server from digitalocean
- Open the terminal
- New Remote Connection
- User: Root
- SSH: Automatic

# INITIAL SERVER SETUP

## Basic Server Setup - Ubuntu 18.04

<https://www.digitalocean.com/community/tutorials/initial-server-setup-with-ubuntu-18-04>

- Create a new user
  - `adduser username`
  - password
  - Basic info
- Grant the new user administrative privileges
  - `usermod -aG sudo username`
- Enable external access to new user
  - `rsync --archive --chown=username:username ~/.ssh /home/username`
  - `rsync --archive --chown=sammy:sammy ~/.ssh /home/sammy`
- Try to login remotely using the new user
  - `ssh username@your_server_ip_address`
- Want to temporarily disable ufw to make easier to set if server is working in setup
  - `sudo ufw disable`
  - Turn back on with `sudo ufw enable`



# SETUP FTP

## Connect to server using Transmit FTP

- For Mac use Transmit
- Add a server
- Protocol: SFTP
- Address: use the ip of the server and port 22
- Provide username
- Set up new SSH Key **"in progress"**
- Select existing SSH Key: id\_rsa, select the key icon and choose id\_rsa, if it worked, key should turn blue
- Connect
- Change directory to the root directory by selecting ip

# WEB APP CONFIGURATION

## Configure server for web app overview

Xxxxxx

### Create a www folder in srv

- Change directory to srv
- Sudo mkdir www
- Make your user the owner of www
  - sudo chown username:root www
  - //The above command provide permissions to create folders in the the FTP application
  - Example: Sudo chown user:group www

### Setup web folder for backend

- In ftp program navigate to www
- Create a folder backend
- Copy backend server from local machine to remote server
  - Do not copy node\_modules folder it is too large

### Setup web folder for frontend

- In ftp program navigate to www
- Create a folder frontend
- Copy frontend node server from local machine to remote server
  - Do not copy node\_modules folder it is too large
- Create react build on local machine.. make sure API points to the new servers IP address and port
  - npm run build
- Copy react build form local machine to remote server

### Install NVM, node version manager

- Run
  - curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
- Close Terminal windows

- Open new remote connection to the server
- <https://github.com/nvm-sh/nvm>

## Install node using nvm

- Nvm install `--lts` //installs latest stable version of node
  - Nvm use node `?????`
- Nvm install node //installs latest version of node
  - Nvm use `--lts` `??????`

## Install required backend packages using npm

Navigate to the backend folder in the terminal

`Npm install`

## Run the backend server

- Node `index_es5.js`
- Running may fail due to bcrypt conflict with node version `????` This may no longer be needed
  - Make sure to install the latest bcrypt with
    - `npm install bcrypt@latest --save`
  - Make sure in install node-pre-gyp
    - `npm install node-pre-gyp -g`
  - And possibly node-gyp if still errors
    - `npm install -g node-gyp`
  - May need to install bcrypt again after node-pre-gyp

## Run MongoDB database

- Run `mongod`
  - Mongo may already be running
- May receive an error:
  - exception in initAndListen: NonExistentPath: Data directory /data/db not found., terminating
  - Missing directory
    - `sudo mkdir -p /data/db`
  - Database should be in `/var/lib/mongodb`
  - Permissions should be `drwxr-xr-x 4 mongodb mongodb`

- Check permissions with `ls -ld mongodb`
- Try restarting the server, logging in as cfoutlier and starting the mongod database service
  - Keep Mongod running
    - `sudo service mongod start`
    - `sudo mongod --fork --logpath /var/log/mongodb.log`. ← — works

## Check backend server accessible and working

- Start the backend server again
  - `Node index_es5.js`
  - Should receive server running on port 5002
- Load the backend server in the browser
  - `http://ipaddress:5002/user`
  - Should receive blank, {"error": "please authenticate."}, but no connection errors
- Post a user using postman
  - Post: `http://167.99.50.58:5002/user`
  - `userFirstName`
  - `userLastName`
  - `userPassword`
  - `UserEMail`
- Load the backend server in the browser again
  - `http://ipaddress:5002/user`
  - Should receive see the user just posted to the database

## Install required frontend packages using npm

- Navigate to the frontend folder in the terminal
- `Npm install`

## Check frontend server accessible and working

- Run the front end server
  - `Node server.js`
- Load the front end server in the browser
  - `ipaddress:5000`

- Try to login using the the user you posted with Postman

## Setup pm2

- `npm install pm2 -g`
- <https://www.npmjs.com/package/pm2>

## Start front end server using pm2

- Navigate to frontend folder
- `Pm2 start server.js`
- See if it is still running with `pm2 list`
- Load the frontend server in the browser

## Install and setupl nginx

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

Type to enter  
text

- Check Nginx by loading the base IP address without any ports

## Setup main domain

- Go to your domain registrar and change the name servers to
  - `ns1.digitalocean.com`
  - `ns2.digitalocean.com`
  - `ns3.digitalocean.com`
- Add domain to DigitalOcean
  - Login to digital ocean
  - Go to networking
  - Add domain

- Create "a" records for
  - .yourdomain.com
  - www.yourdomain.com
  - api.yourdomain.com
- Check to see if domain loads Nginx

## Setup reverse proxies in nginx (continue nginx setup)

- (yourdomain.com, www.yourdomain.com) and (api.yourdomain.com)
- Login to terminal and navigate to
- /etc/nginx/sites-available/
- Sudo Nano default to edit
  - Copy over reverse proxy code domains
  - Comment out reference to port 80
  - Root path:
  - Server name:
  - Location:
  - See saved text file nginx sites-available for reference
- Save the file
- sudo systemctl disable nginx
- sudo systemctl enable nginx
- sudo systemctl restart nginx
- Check to see if the domain loads to the app now instead of nginx page
- <https://www.digitalocean.com/community/tutorials/nginx-essentials-installation-and-configuration-troubleshooting>
- <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>
- <https://www.keycdn.com/support/nginx-reverse-proxy>
- <https://www.youtube.com/watch?v=Db1Jr9TKPuc>
- <https://www.hostinger.com/tutorials/how-to-set-up-nginx-reverse-proxy/>

## Setup subdomain

- Same steps as the reverse proxy except in a new server block
- Dont need root path

## Setup SSL

- Should be able to run Certbot to configure Nginx using link below
- <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>
- Use Certbot to setup ssl for yourdomain.com and www.yourdomain.com
- User Certbot to setup ssl separately for api.yourdomain.com
- Certbot
- Additional resources
  - Nginx with SSL
    - <https://www.digitalocean.com/community/questions/how-to-run-node-js-server-with-nginx>
  - SSL
    - Digital ocean SSL
    - <https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-ubuntu-18-04>
    - <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>
    - <https://www.youtube.com/watch?v=jo6xdzwwAqY>

# TESTING

## Test server configuration Add a server

- Xxxx