**How to host your MEAN app on AWS for free**

[**http://www.techprimelab.com/blogs/how-to-host-your-mean-app-on-aws-for-free**](http://www.techprimelab.com/blogs/how-to-host-your-mean-app-on-aws-for-free)

As the title suggests, we are going see how to host a **MEAN** stack app on the **AWS** EC2 instance server for **free**.

Before we get started with the tutorial, I will briefly introduce the technology stacks that we are going to use.

**What is MEAN?**

**M** MongoDB - NoSQL database program.  
**E**  Express - Node.js web application framework.  
**A**  Angular - front-end web application platform.  
**N**  Node.js - JavaScript run-time environment for server-side.

Amazon Web Services(AWS) provides on-demand cloud computing services. Services are free to join, pay only for what you use.

Amazon has the largest market share in the cloud business, leading Microsoft, Google, and IBM by a huge margin.

Once you sign up for AWS account you can start using their free tier services. On sign up, they will ask for your debit/credit card information. AWS debits small amount from your account which is credited back to your account after verification.

**Prerequisites**

1. AWS account
2. Putty - SSH client on your local computer
3. FileZilla - FTP client on your local computer
4. MEAN App (GitHub<https://github.com/linnovate/mean> or you can use your own)

Let’s start with the tutorial.

**1. Creating EC2 instance on AWS**

1. Sign in to your console
2. Services → EC2
3. Click Launch Instance
4. Select suitable instance (I am going with **Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** for this tutorial)
5. You can keep all the defaults and jump to Configure security groups
6. You will need to configure your inbound traffic rules. Click Edit & Add Rule.

Note: *The below configuration will suffice for this tutorial purpose.*

*Please note that this configuration is not intended for production purpose.*

*You can read more about how to configure the security groups for production release.*

| **Type** | **Protocol** | **Port Range** | **Source** |
| --- | --- | --- | --- |
| Custom TCP Rule | TCP | 8080 | 0.0.0.0/0 |
| SSH | TCP | 22 | 0.0.0.0/0 |

1. Click review → click Launch
2. A pop up asking you to download key pair will appear.
3. Choose ‘Create a new key pair’ → give it a proper name
4. Download key pair (Keep it in a safe location on your local drive.)
5. Click Launch.

Your EC2 instance is now ready.

**2. SSH into your EC2 instance using Putty**

Using Putty we are going to access the terminal of the server and run few commands.

We are going to use the ‘key pair’ file that we earlier downloaded to access the server via Putty. The key pair file is in ‘.pem’ format, but we need it in ‘.ppk’ format. Putty comes with **PuTTYgen** where we can make the conversion.

1. Open PuTTYgen → Click ‘Load’
2. Select ‘.pem’ key pair file that you downloaded from AWS.

Note: when browsing for your pem file be sure to select All Files in the dropdown list that is located to the right of the File name field.

1. Click ‘Save private key’.
2. You will receive a warning message asking if you want to save this key without a passphrase. Select Yes.
3. Provide a name for your ppk file and click save.
4. Now, open your AWS account, Go to Services → EC2
5. Click on running instances → Click on the instance you created in the earlier step.
6. At bottom of your browser you can see many tabs, on the first tab with title ‘*description’* → Find & Copy IPv4 Public IP
7. Now open PuTTY.
8. Paste the ‘IPv4 Public IP’ into the ‘Host Name (or IP address)’ input box.
9. Enter ‘22’ into ‘Port’ input box.
10. Now on the left side panel expand SSH → Click ‘Auth’
11. Click browse in the ‘Private key file for authentication’ input box.
12. Select the ‘.ppk’ key pair that you created using PuTTYgen.
13. Click open
14. Again Click open on PuTTY configuration Dialog box. You will see the terminal window, where it will ask you to enter the username ‘login as:’
15. Sign in to your instance with username (generally ‘ubuntu’ for Ubuntu instance & ‘ec2-user’ for Linux instance) ‘login as: ubuntu’ hit enter.

Congrats! You have logged into your AWS server terminal.

**3. Installing Node.js on AWS server**

Now that you have access to your server’s terminal, we need to set up our environment to run our MEAN App. First, we need to install Node.js which will also install NPM

We have to run few commands for this installation as below.

1. sudo apt-get update
2. sudo apt-get install build-essential libssl-dev
3. curl <https://raw.githubusercontent.com/creationix/nvm/master/install.sh> | sh
4. source ~/.profile
5. nvm ls-remote
6. nvm install 8.6.0
7. To check installation was correct run  node -v
8. It also installs npm , check it with npm -v

Now we have Node and NPM installed on our server.

**4. Installing MongoDB**

Again we have to run commands to install MongoDB.

1. sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
2. echo "deb [ arch=amd64,arm64 ] <https://repo.mongodb.org/apt/ubuntu> xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.6.list
3. sudo apt-get update
4. sudo apt-get install -y mongodb-org
5. To start MongoDB as a service → sudo service mongod start
6. To check status → sudo service mongod status
7. To Stop Service →  sudo service mongod stop

Great! MongoDB is up and running.

**5. Clone your MEAN App repo onto the server.**

1. I am using this official MEAN stack repo:  <https://github.com/linnovate/mean>, run following command to clone this repo on your server.
2. git clone<https://github.com/linnovate/mean.git>
3. cd mean
4. npm install

Your MEAN App is now on the server and ready for deployment.

**6. Transfer prod build dist on server.**

It will not be possible to build prod dist on the server with 1GB ram, So we will build locally and transfer our dist to the server via FTP (FileZilla) .

1. So make a build locally by running npm run build:prod
2. Open FileZilla.
3. Open Site Manager. File > Site Manager
4. Click New site
5. Enter your server’s  IPv4 Public IP in ‘host’ field
6. Select ‘SFTP - SSH File Transfer Protocol’
7. Logon Type: ‘Key file’
8. User: ‘ubuntu’ for Ubuntu server & ‘ec2-user’ for Linux server
9. Select the key file that you earlier downloaded. (.ppk)
10. Click connect
11. After you are connected to your server navigate to the local folder where your ‘dist’ is located.
12. Transfer this ‘dist’ folder to the server by clicking ‘Upload’ from the context menu.
13. Make sure MongoDB service is running.
14. After the transfer is complete Open putty, make sure you are in ‘mean’ directory then run npm run server:prod
15. In your browser paste the IPv4 Public IP followed by ***:8080***

**Voila!! You have hosted your app & Launched it to the World.**

Note: Be sure to stop your instance on AWS after you are done with tutorial and tested your app.

You get a limited number of hours per month with the free account, so don’t keep the server running all the time. If you exhaust your free usage, billing will start.