Deploy Node.js app to Amazon EC2

Node.js application can be hosted by various web servers, Apache, IIS, Nginx, etc. In the post, I introduce how to deploy Node.js application to the Amazon Cloud Server, EC2.

There are mainly 5 steps:

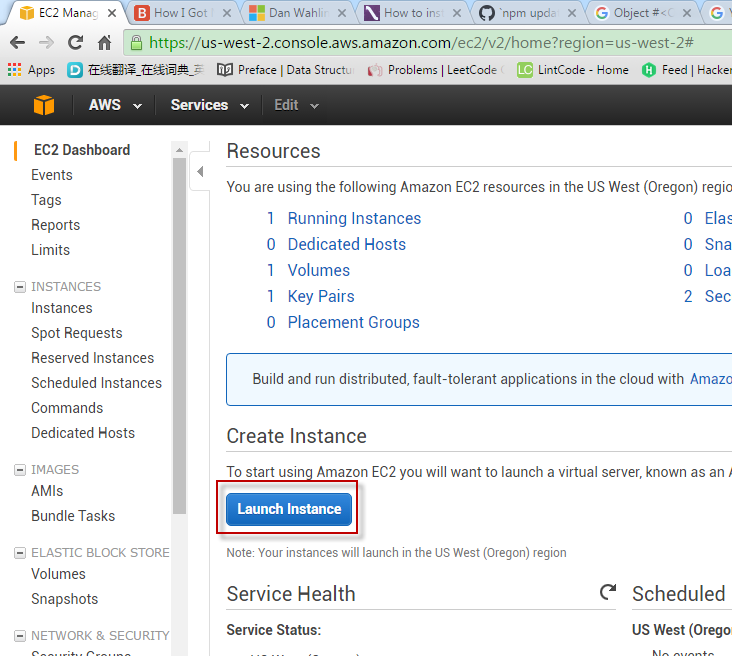
* Create EC2 Instance
* Use Putty to Connect EC2 Remote Server
* Setup Node.js environment in EC2 Instance
* Create simple node app and start Node server
* Deploy Local Node.js application to EC2 Instance

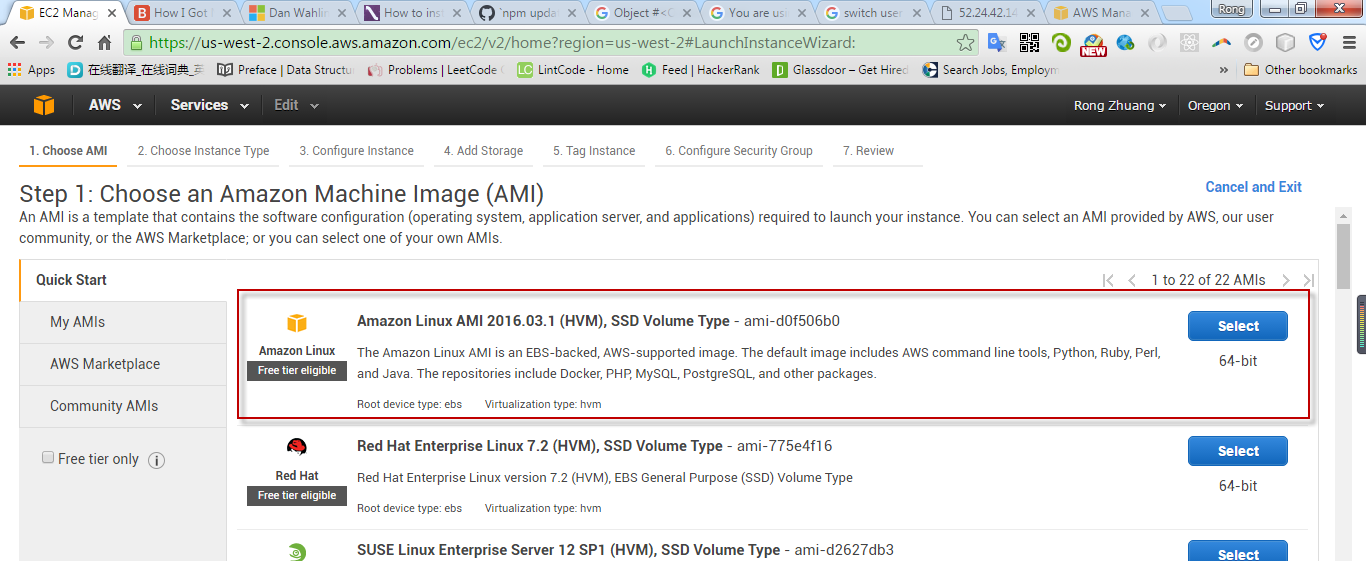
1. Create EC2 Instance
   1. Login to Amazon EC2

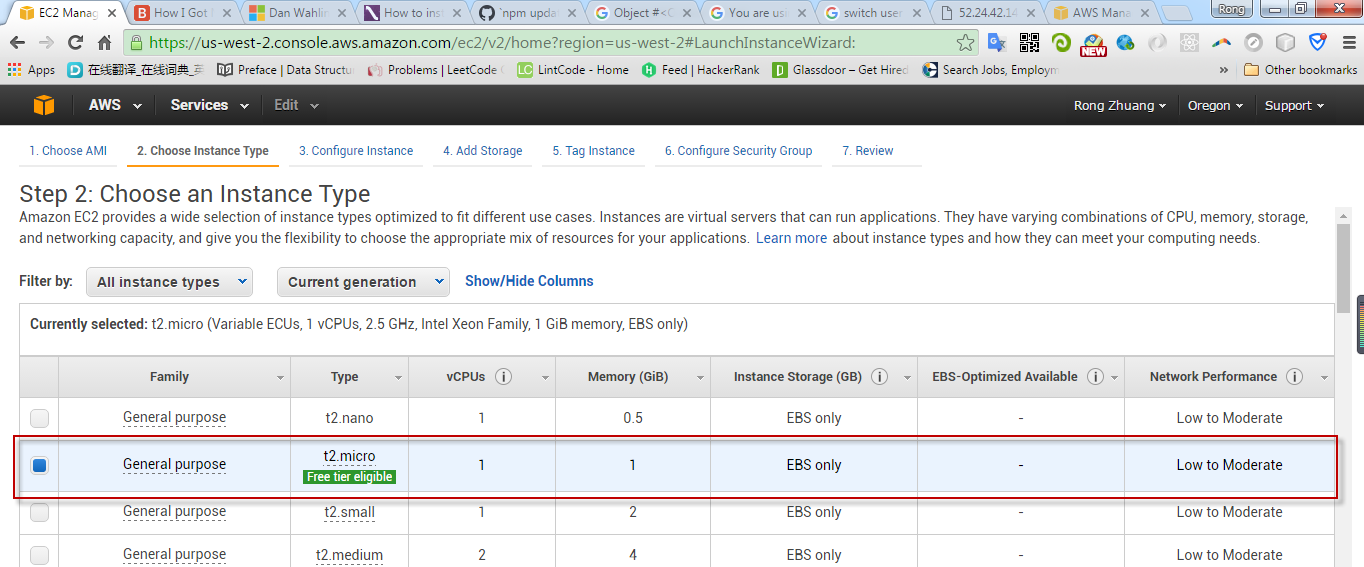
<http://aws.amazon.com/ec2/>

Sign up ‘Amazons free micro instance of EC2’ if you have no AWS Account yet.

* 1. Create Instance

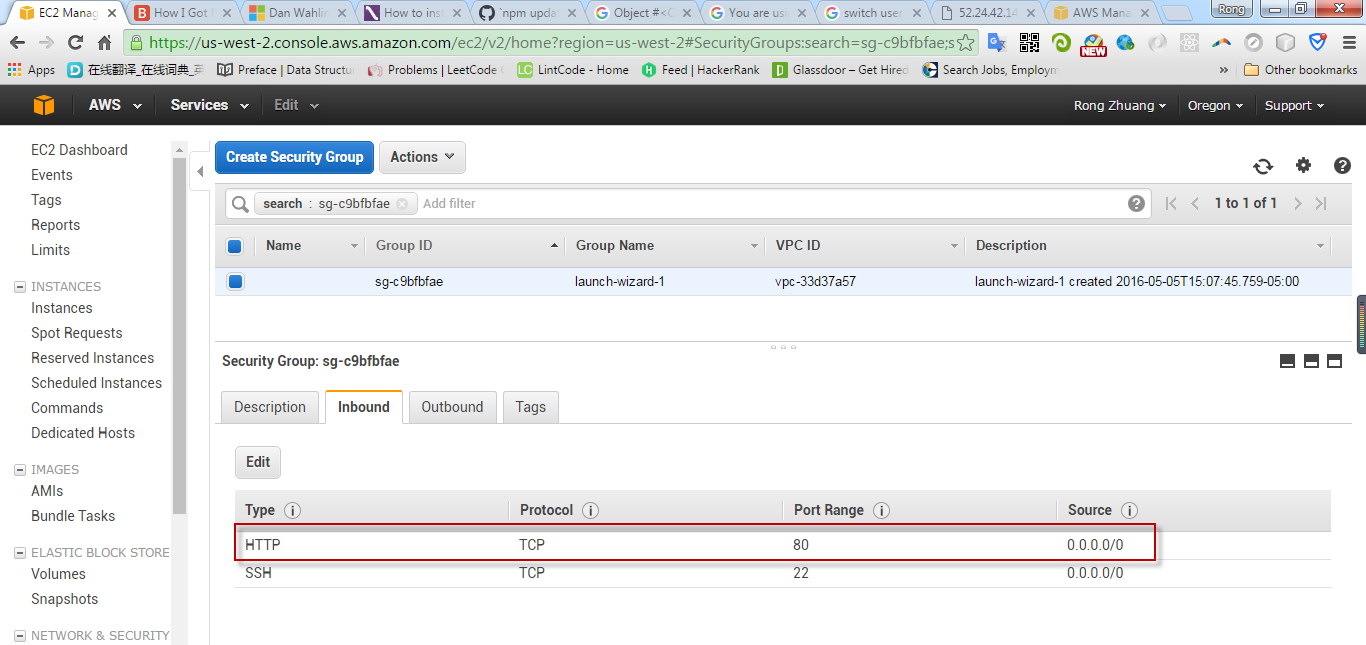




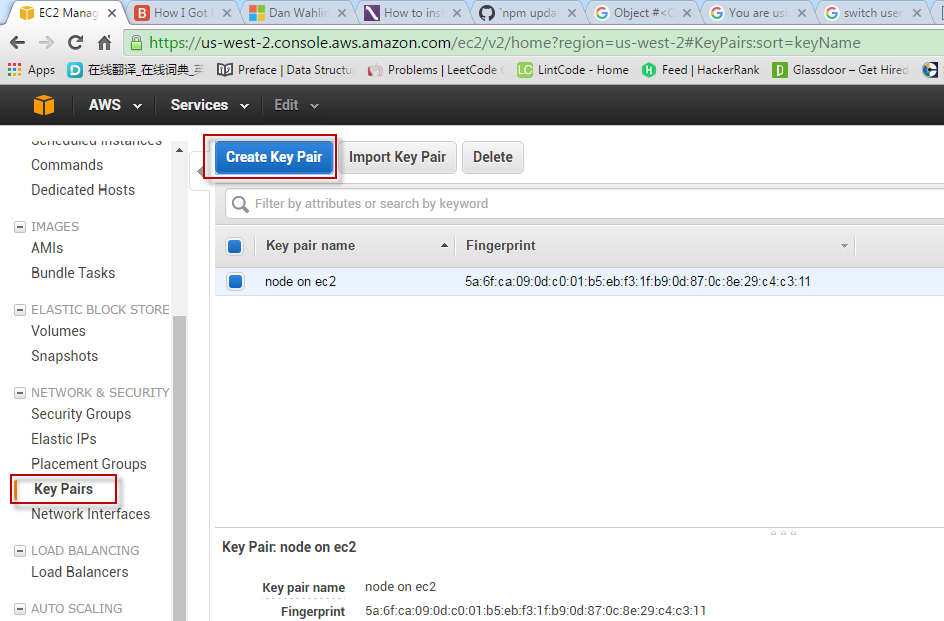


Finally, review and launch.

After the instance is generated, create security group.

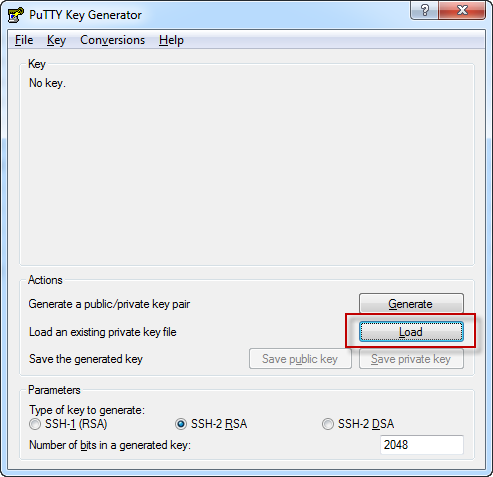


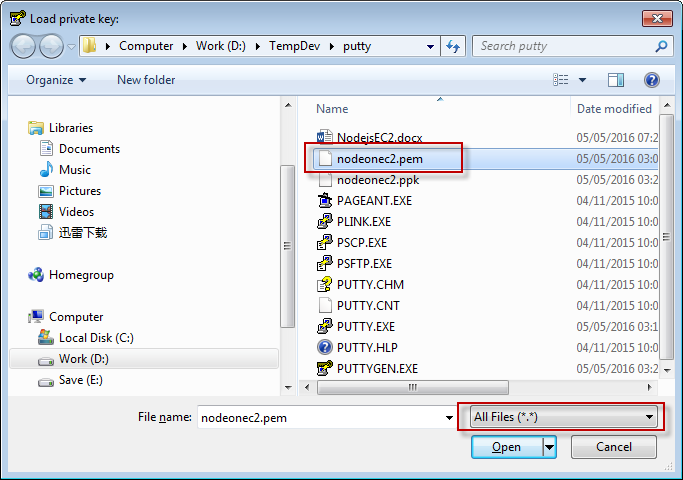
* 1. Create Key Pair

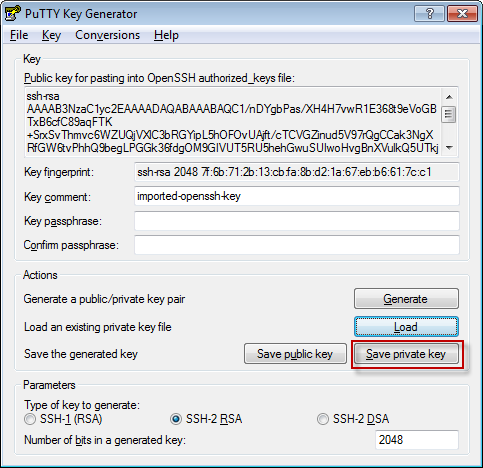


Down load the private key to local machine, eg. nodeonec2.pem.

1. Use Putty to Connect EC2 Remote Server
   1. User PUTTYGEN.EXE to convert key



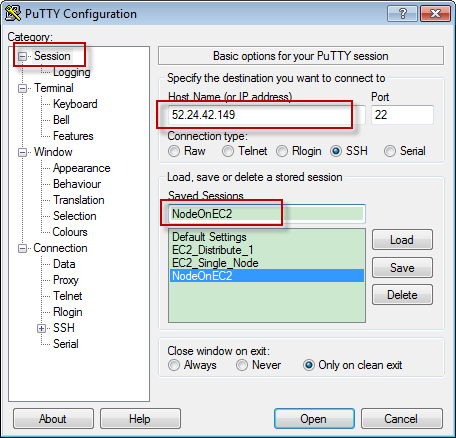




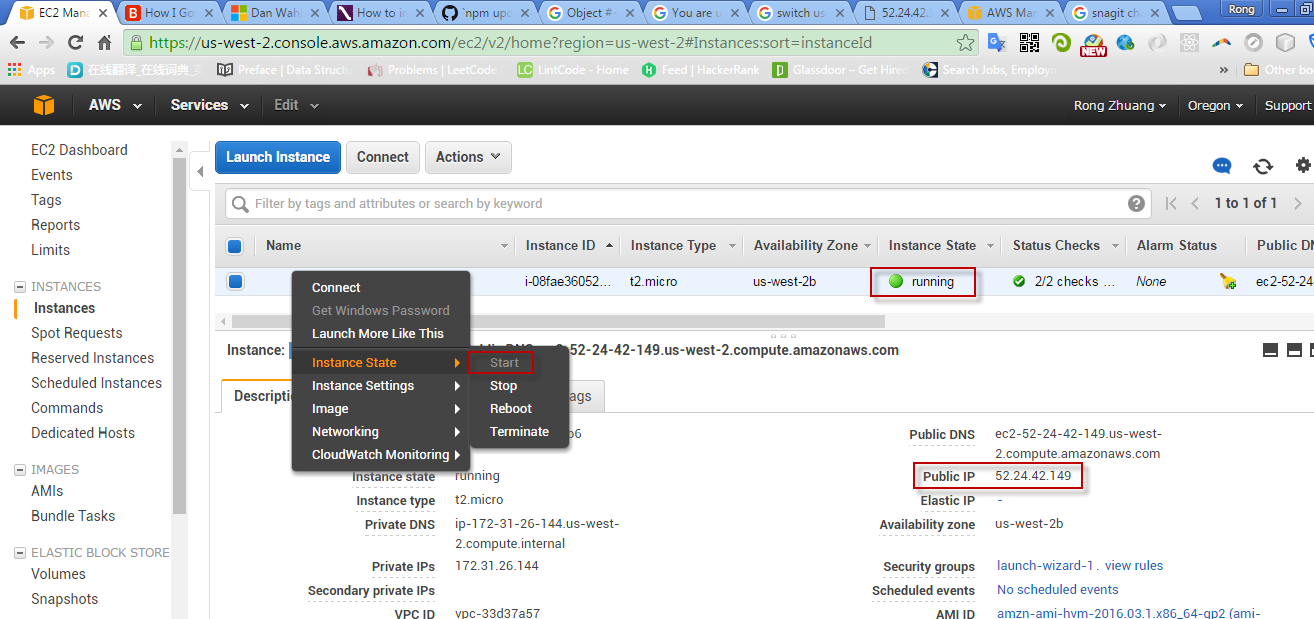
After save, nodeonec2.ppk is generated.

* 1. Configure PUTTY.EXE

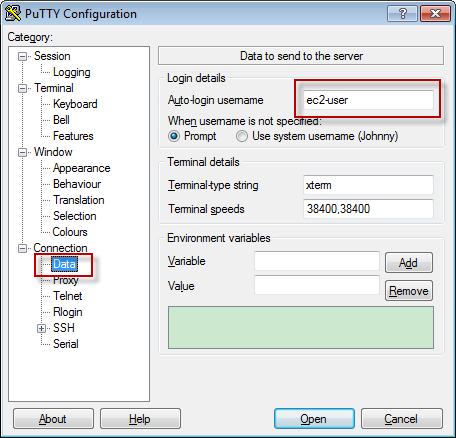
In session, provide IP address(The public ip of your EC2 instance) and session name.



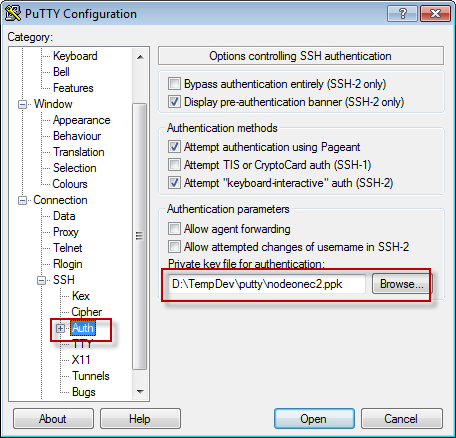
The IP is the Public IP of your EC2 instance. It is only available when the instance is running.



Connection->Data, add user, always ‘ec2-user’.



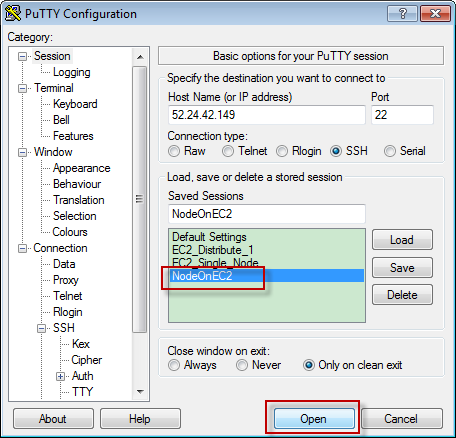
Connection ->SSH ->Auth, load the private key ppk file.

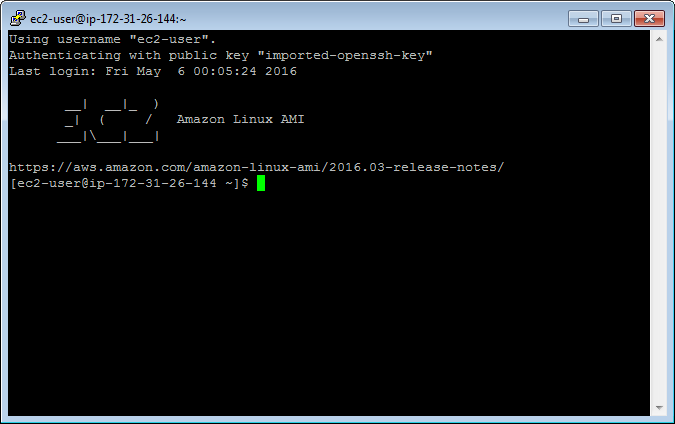


Back to session, save the configuration.

* 1. Connect to EC2 remote server

Choose the newly created session, double click it or click the ‘Open’ button.





Note, the IP address here is internal IP. When using putty to connect EC2 remote server, make sure launch the instance first. You have to change the IP in putty every time if you reboot the instance. The Public IP address of the EC2 instance changes to different value once it restarts.

1. Setup Node.js environment in EC2 Instance
   1. Update your EC2 Amazon Linux

sudo yum update

* 1. Install GCC

sudo yum install gcc-c++ make

sudo yum install openssl-devel

* 1. Install Node.js

sudo yum install git

git clone git://github.com/nodejs/node

cd node

./configure

make //it may take long time to compile

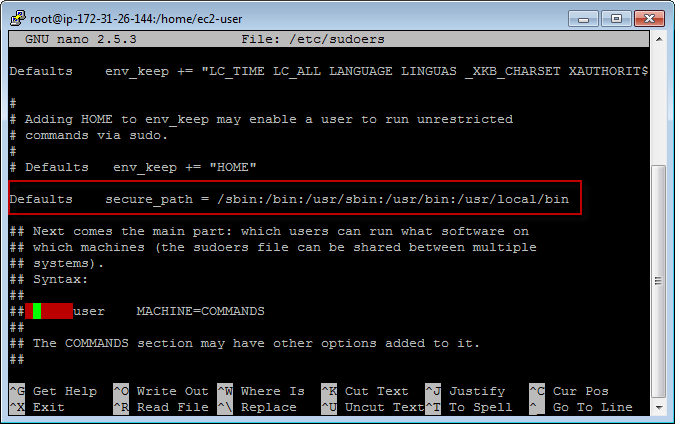
sudo make install

* 1. Add node folder to secure\_path

sudo su

nano /etc/sudoers

append :/usr/local/bin to the end of secure\_path



* 1. Install npm

git clone <https://github.com/npm/npm>

cd npm

sudo make install

1. Create simple node app and start Node server
   1. Create folder ‘site’

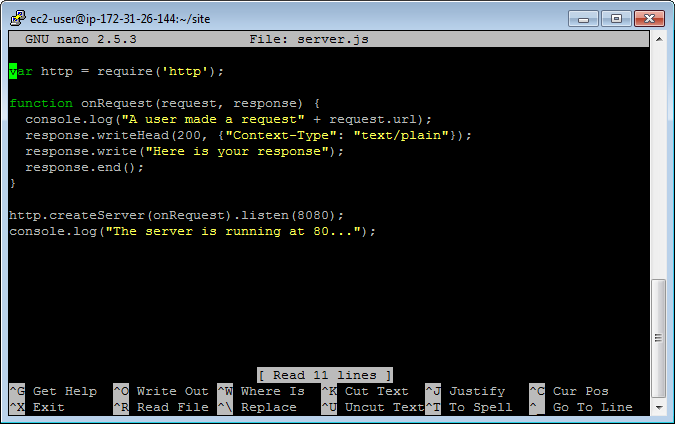
mkdir site

* 1. Create file ‘server.js’

nano server.js

append the following content to the file, save and exit.

|  |
| --- |
| var http = require('http');  function onRequest(request, response) {  console.log("A user made a request" + request.url);  response.writeHead(200, {"Context-Type": "text/plain"});  response.write("Here is your response");  response.end();  }  http.createServer(onRequest).listen(8080);  console.log("The server is running at 80..."); |

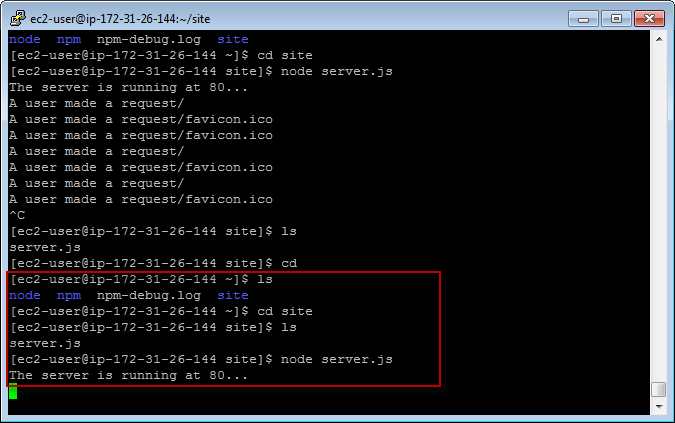


* 1. Redirect port

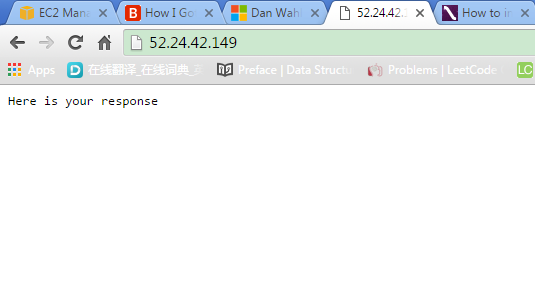
You cannot make node server listen to port 80. Run the following command to redirect requests from port 80 of EC2 server to port 8080 of our Node server.

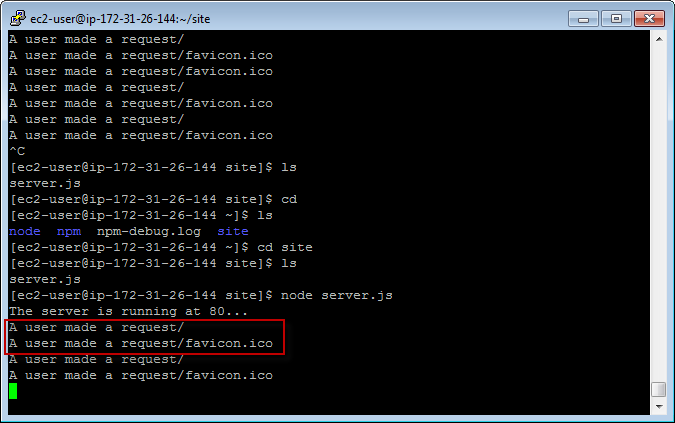
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to 8080

* 1. Start our Node server



* 1. Open web browser, access the site with public IP.

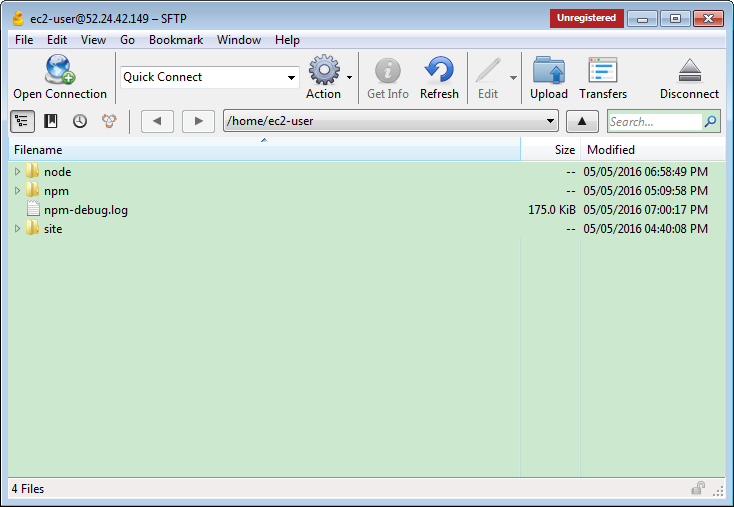


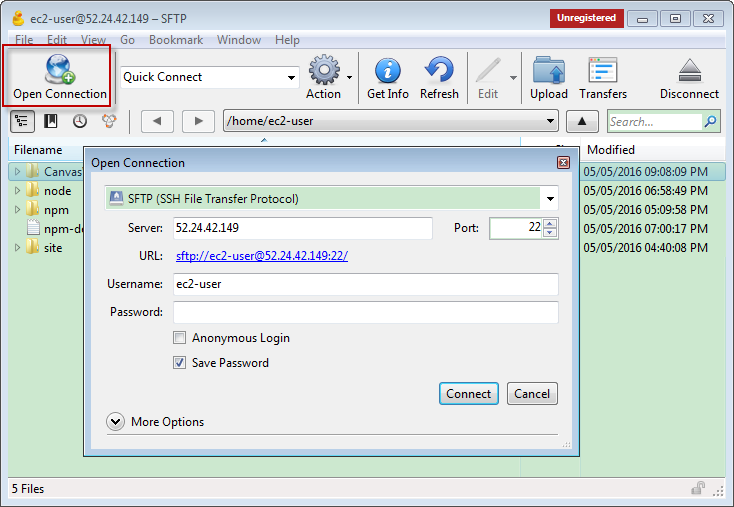


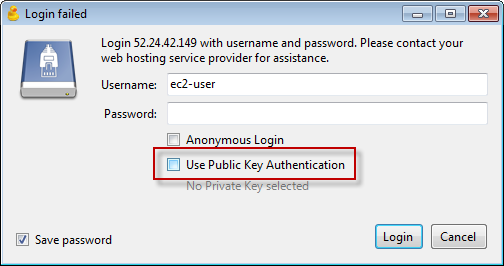
1. Deploy Local Node.js application to EC2 Instance
   1. Install CyberDuck

<https://cyberduck.io/?l=en>

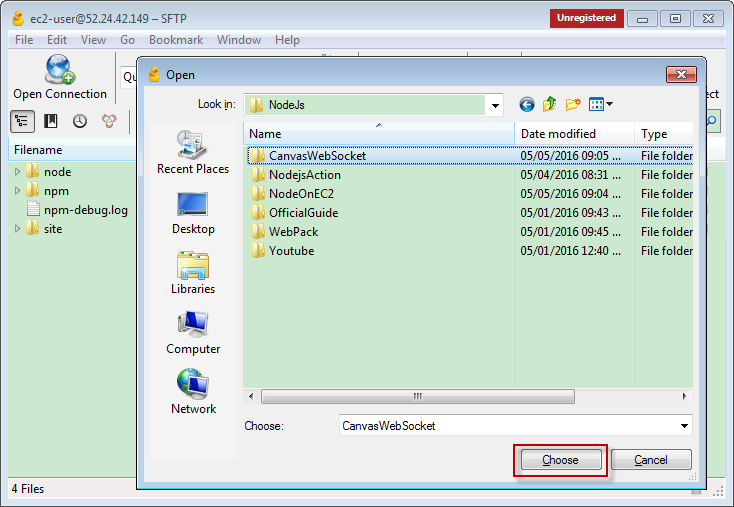
* 1. Launch CyberDuck and Upload

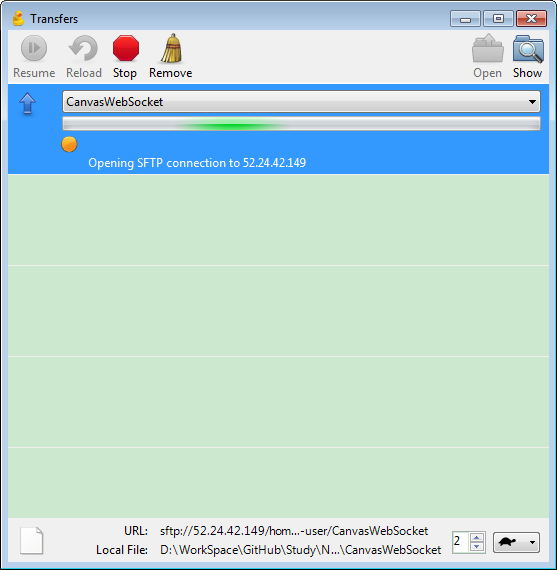


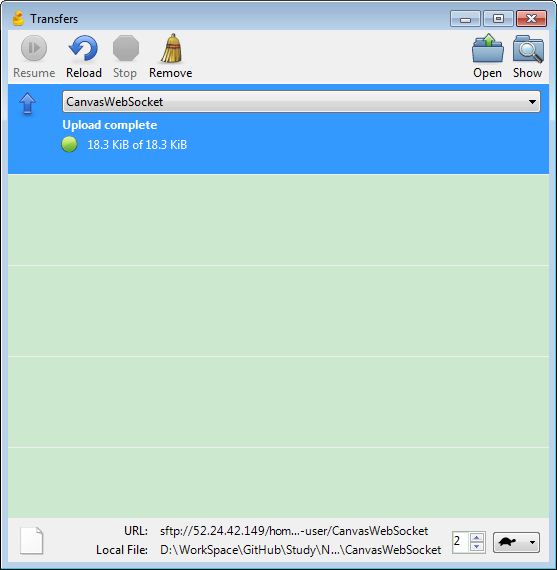




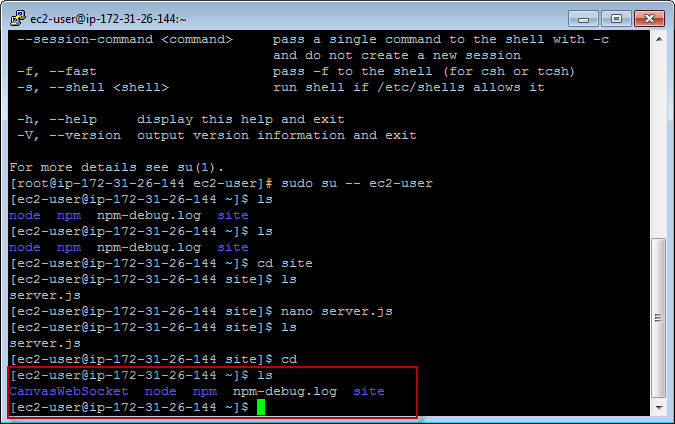
Select the folder, make sure delete all files in ‘node\_modules’ folder.







Refresh the folder in putty, the new folder exits.



* 1. Go into the folder

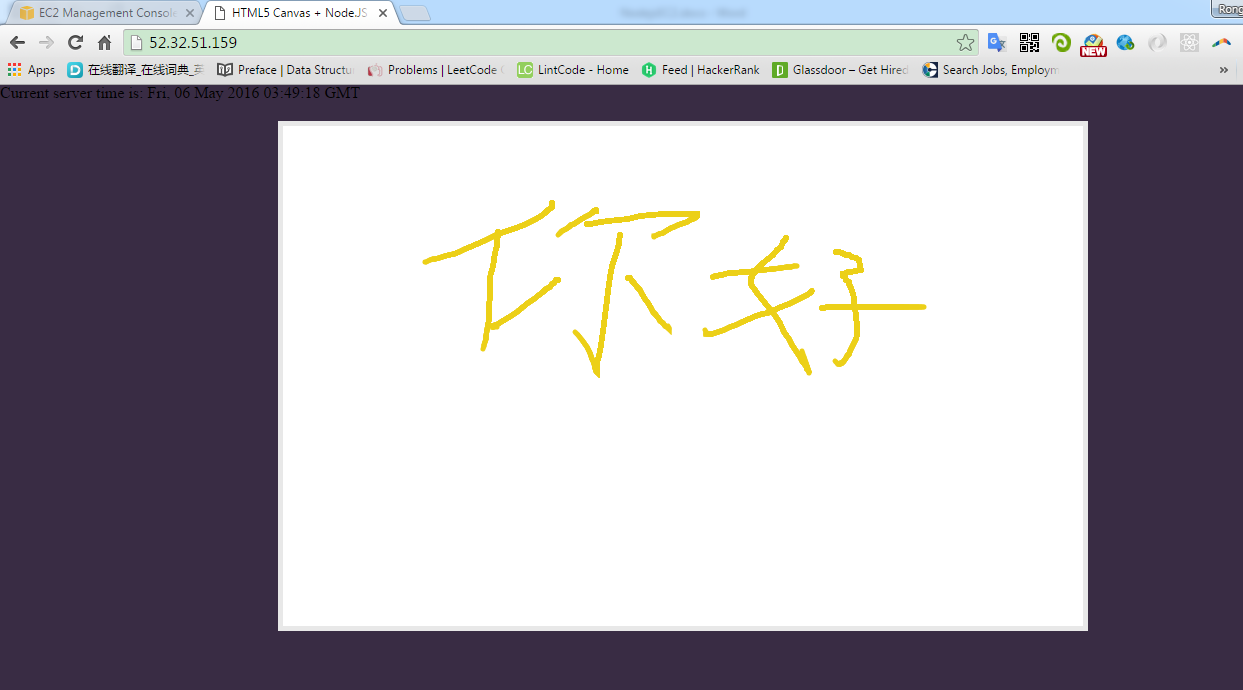
npm install

npm start

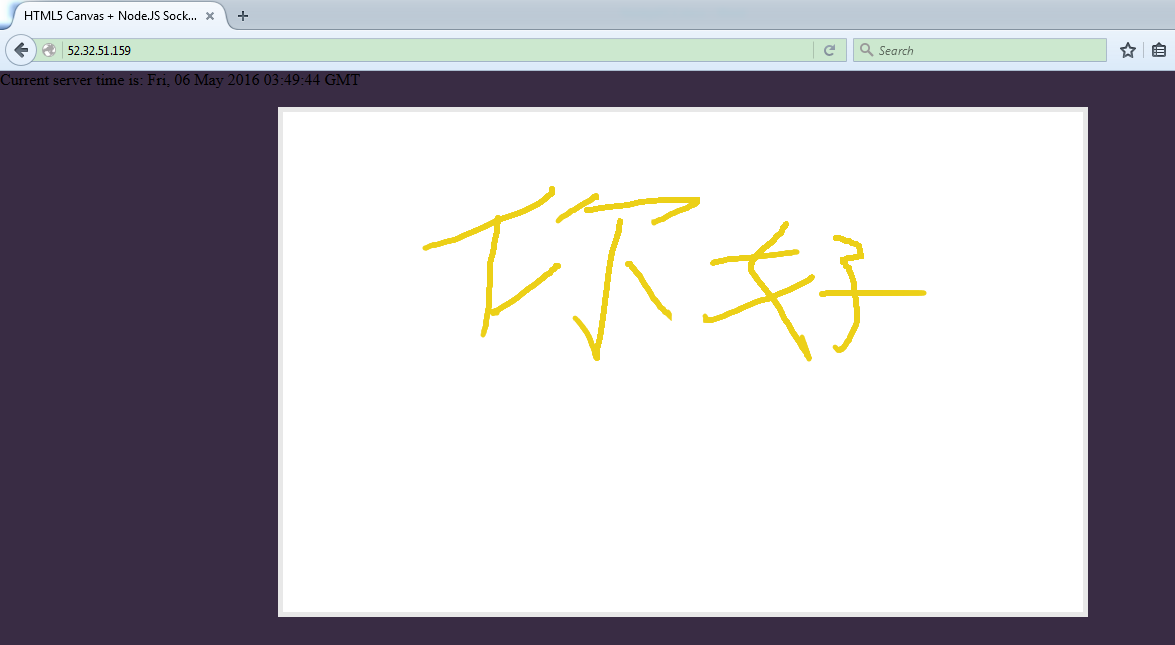
* 1. Open Chrome, Firefox, on different device.

This node project is a drawing application. It uses Socket.IO to broadcast the changes from one client to other clients.

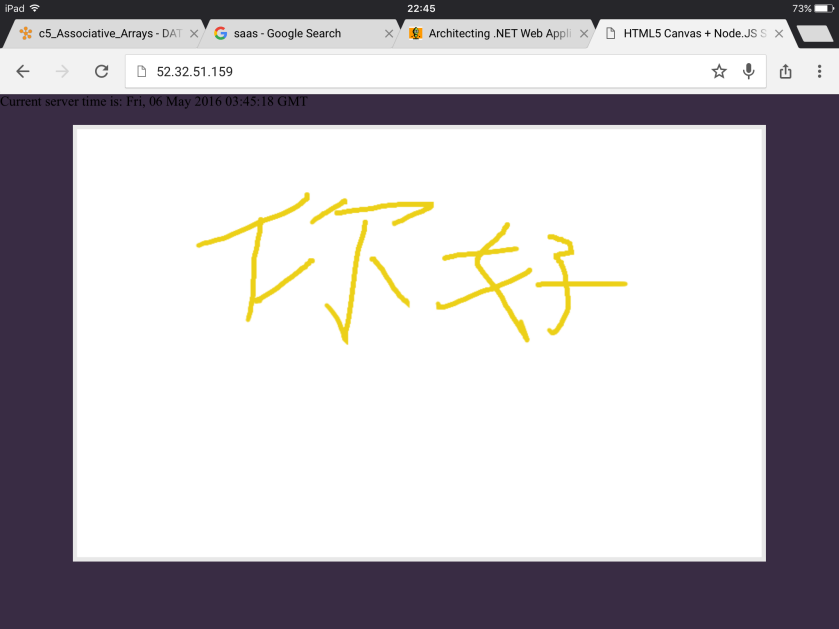
The first user opens it in chrome, waits others to join and draws something later.



The second user opens it in Firefox. The same painting as the first user draws.



The third user open it on iPad.



The forth user opens it on iPhone.



1. Useful command in linux

|  |  |
| --- | --- |
| **Command** | **Description** |
| ls | Show files/directories under the current folder |
| sudo su | Switch to root user |
| sudo su -- ec2-user | Switch to ec2-user |
| nano filename | Open/Create file with nano |
| mkdir foldername | Create folder |
| sudo make uninstall | Uninstall, go to the folder and run it. |

1. Reference

How I Got Node.js Running On A Linux Micro Instance Using Amazon EC2

<http://www.bennadel.com/blog/2321-how-i-got-node-js-running-on-a-linux-micro-instance-using-amazon-ec2.htm>

How to install & setup Node.js on Amazon EC2 – complete guide

<http://iconof.com/blog/how-to-install-setup-node-js-on-amazon-aws-ec2-complete-guide/>

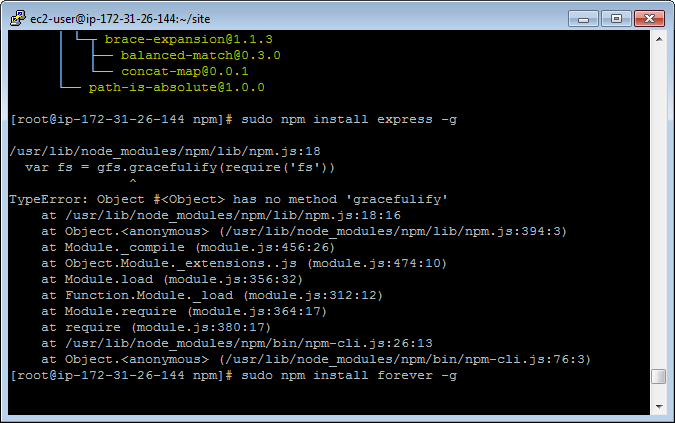
Setup Node.Js Enviroment on Amazon EC2 linux

<http://adndevblog.typepad.com/cloud_and_mobile/2014/12/setup-nodejs-enviroment-on-amazon-ec2-linux-1.html>

<https://docs.npmjs.com/misc/removing-npm>

1. Issues
   1. Remove npm

Sometime, npm itself doesn’t work properly.



Then we have to uninstall and install it again.

sudo npm uninstall npm -g

If it doesn’t work, go the ‘npm’ folder, run:

sudo make uninstall