**AWS Problems and Solutions**

1. Can’t connect to ec2 instance through ssh
   1. Make sure you are using the original key-pair that was created when the instance was created.
   2. Connect using the following command “ssh -i “wwud.pem” [ec2-user@3.141.24.52](mailto:ec2-user@3.141.24.52)
   3. You will most likely get the following error:
      1. @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@ WARNING: UNPROTECTED PRIVATE KEY FILE! @

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Permissions for 'wwub.pem' are too open.

It is required that your private key files are NOT accessible by others.

This private key will be ignored.

Load key "wwub.pem": bad permissions

ec2-user@3.141.24.52: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).

* 1. Right click on the wwub.pem file and go to properties
  2. In the attributes and security section, make sure the file in unlocked and apply that change
  3. Click on the security tab and click advanced
  4. If inheritance is disabled, enable it then disable it again
     1. Select “Convert inherited permissions into explicit permissions on this object.”
  5. Remove all users expect the Administrators/your user login (ie: WIT/senderp & WITPF0TYJ35/Administrators)
  6. You should be able to connect now. If not, open a support ticket with AWS.

1. Need GCC compiler/other tools to get started with compiling and running software on VM
   1. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/compile-software.html>
   2. <https://ostechnix.com/install-development-tools-linux/> (scroll down)
2. Connection timeout when trying to connect through SSH
   1. Public IP address may have changed, try again with new public IP address

Installing nginx:

Make sure all outbound traffic is enabled

Sudo yum install nginx

sudo systemctl start nginx

sudo systemctl enable nginx

Locate your nginx.conf file. Likely at /etc/nginx/nginx.conf

Find the http block.

Somewhere in the http block, write include /etc/nginx/conf.d/\*.conf; This tells nginx to pull in any files in the conf.d directory that end in .conf. (I know: it's weird that a directory can have a . in it.)

Create the conf.d directory if it doesn't already exist (per the path in step 3). Be sure to give it the right permissions/ownership. Likely root or www-data.

Move or copy your separate config files (just like you have in /etc/nginx/sites-available) into the directory conf.d.

Reload or restart nginx.

Eat an ice cream cone.

Any .conf files that you put into the conf.d directory from here on out will become active as long as you reload/restart nginx after.

Note: You can use the conf.d and sites-enabled + sites-available method concurrently if you wish. I like to test on my dev box using conf.d. Feels faster than symlinking and unsymlinking

Add wwub.conf into conf.d/

Paste the following and put the EC2 public IPs OR LINKED domain name where it says “your\_domain.com”

server {  
listen 80;  
server\_name your\_domain.com www.your\_domain.com;  
location / {  
proxy\_pass [http://127.0.0.1:8080](http://127.0.0.1:8080/);  
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;  
proxy\_redirect off;  
}  
}

sudo systemctl reload nginx

Go to public IP (WITHOUT PORT SPECIFIED) to check if you get a 502 Bag Gateway message

https://keithweaverca.medium.com/setting-up-mern-stack-on-aws-ec2-6dc599be4737