**To Setup apache :**

**Link :** https://www.youtube.com/watch?v=wNr7YqjjzOY

Instructions:  
Go to [http://aws.amazon.com/](https://www.youtube.com/redirect?q=http%3A%2F%2Faws.amazon.com%2F&v=wNr7YqjjzOY&redir_token=l-jCqDe-srsJJJuUWb6iwloCmrF8MTUxMDE2OTY0MkAxNTEwMDgzMjQy&event=video_description) and create a new account if you do not have an account already.  
A credit card or debit is required.  
  
Sign into your AWS account under console management.  
  
Select EC2 and create a new instance.  
  
Choose one of the Ubuntu AMIs, and leave everything default. Under Security groups, create a new security group with SSH and HTTP.  
Name your key and download.  
  
Under the terminal, maneuver to the key and chmod 400 to make sure that the permissions are correct.  
Under windows, you may have to run as administrator the command prompt or cygwin to chmod properly.  
Or you can use the command that I did, which was:  
chown :Users testTutorial.pem  
Then I did:  
chmod 400 testTutorial.pem  
Allows only me to read  
Then:  
ssh -i testTutorial.pem ubuntu@...(public IP)  
  
Once you are on the Ubuntu AMI, you can start to mess around with the packages to install.  
I recommend the following:   
Run the command:  
sudo apt-get install apache2 libapache2-mod-php5 php5 mysql-server php5-mysql  
You can also get phpadmin if you would like.  
Remember to restart the apache service:  
service apache2 restart  
  
Then, run a secure installation of mysql:  
mysql\_secure\_installation  
  
Go put in your browser the public IP. It works! But what if you want to change the index.html  
  
Well, go cd ../../var/www  
In this directory www, you can edit the index.html  
sudo vim index.html  
You can change the index.html to anything you want using the editor of your choice.  
  
The site changes.

1. **Add red colored line in apache2.conf**

# Sets the default security model of the Apache2 HTTPD server. It does

# not allow access to the root filesystem outside of /usr/share and /var/www.

# The former is used by web applications packaged in Debian,

# the latter may be used for local directories served by the web server. If

# your system is serving content from a sub-directory in /srv you must allow

# access here, or in any related virtual host.

LoadModule alias\_module /usr/lib/apache2/modules/mod\_alias.so

<Directory />

Options FollowSymLinks

AllowOverride None

Require all denied

</Directory>

<Directory /var/www/html/>

AllowOverride none

Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch

AddHandler cgi-script .cgi .py

</Directory>

LoadModule cgi\_module /usr/lib/apache2/modules/mod\_cgi.so

2. **Run sample code ( Paste inside html folder as test.py)**

#!/usr/bin/python

import cgi,cgitb

cgitb.enable()

print "Content-type: text/html"

print

print "Hello"