İSTANBUL BİLGİ UNIVERSITY
Department of Computer Science

http://cs.bilgi.edu.tr

CMPE 283 Worksheet 05

Project 05

Deadline 0900 Monday, October 27

Classwork

In this class we will finally get a complete multi-domain multi-user working virtual web server.

Students who wish to set up a virtual server on a different provider from Digital Ocean or who want to use different operating system choices are absolutely free to do so. You should note that the instructions in the worksheet may not work exactly in other Linux distributions, though they will almost certainly work with slight modifications.

Any student who does not want to set up a droplet for money reasons, should contact me immediately so we can look for ways to solve the problem.

You should also note that your lecturer has not used Microsoft software at all for the past 14 years, so I will not have the knowledge to help you solve problems with Microsoft software, though you are, of course, free to use it if you have the necessary licences and are prepared to make the effort to support it yourself.

Setting up Linux as an alternative boot on your own computer will make it easier for you to experiment and learn offline and make certificated access to your droplet or other server easier. I strongly recommend you do this.

You should be able to do many of the tasks required from this course from a tablet or even a smart phone. However, the easiest and most productive way to do the practical work in this course will be from a personal computer with Linux installed. Solving the problems you will encounter with *any* method of accessing your server is *part of the course.*

Setting up Linux on your own computer

See the intruction on previous workheet

Finally getting a full multidomain Apache setup

I suggest you have Linux up and running on your computer while you are working on the server. It will be handy, and also good practice.

If you do not yet have Linux on your computer, you will need to use a program like Putty to access your droplet.

Debian Apache is version 2.2.x. The latest version of Apache is 2.4.x. This is typical of Debian, a very conservative distribution. So be careful, when reading Apache documentation that you read the documents for 2.2 and not for the latest version. There are some annoying differences.

We are going to use a script called apachevirtualhostconfig. This is all my own work. So don't trust it!

Here are the contents of the apachevirtualhostconfig script:

apachevirtualhostconfig

```
#!/bin/bash
#
# apachevirtualhostconfig  sets up an apache virtual host using a template
#
# (c) Chris Stephenson 2014
# Licensed under GNU Public License v3 or later
#
# No warranty explicit or implied that this script is fit for purpose or even that
#
# must run with root privileges
# parameters - domain.name  sftp.user.name IP number
# example use: apachevirtualhostconfig myfirstdomain.com myfirstdomainsftpuser 111.
#
#
# Note: silly parameter values may make this script fail
# Avoiding that would require a sanity clause.
# And everyone knows there is no sanity clause.
#
#
# set up a user to own the directory for the domain, without command shell access
adduser --home /var/www/$1 --shell /bin/false --gecos ’ ’ $2
# add the user to the sftp group, so it will be locked in an sftp chroot jail
# note that we must have set up the sftp group and the sftp config first for this t
usermod  -G sftp $2
```

```
# remove unnecessary files from the home directory
rm /var/www/$1/.bash_logout /var/www/$1/.bashrc /var/www/$1/.profile
# run the template file through sed to create the actual config file for this virtu
cat /etc/apache2/sites-available/template | sed -e "s/_domain_/$1/g" | sed -e "s/_i
# set up the web root directory
if [ -d "/var/www/$1/webroot" ]; then
    echo "webroot ok"
else
    echo "creating webroot"
    mkdir "/var/www/$1/webroot"
fi
# make an empty robots.txt - cuts down the error messages in the log - can be fille
touch /var/www/$1/webroot/robots.txt
# make a default index.html so we can test our new domain and see requests are goin
echo "<html><head></head><body><h1>Test index page for $1 </h1></body></html>" >  /
# set up the correct file access permissions and ownership for the root and the doc
# virtual domain
#
# set group ownership to apache user (www-data) so apache can read the files
#
chown -R $2:www-data /var/www/$1/webroot
#
# Only group and user can read the document root
#
chmod -R 750 /var/www/$1/webroot
#
# Root must own the directory below webroot (sftp jail requirement)
#
chown root:$2 /var/www/$1
chown root:www-data /var/www/$1/*
chmod 755 /var/www/$1
chmod 755 /var/www/$1/webroot
chmod g+s /var/www/$1/webroot
chmod 750 /var/www/$1/webroot/*
#
# This section enables logging of sftp access by our sftp users
# This helps determine how and when our sftp users have broken things.
# Anyone who claims they really know what is happening in this set up doesn't fully
# understand the situation. I just tried and tried until it worked.
#
mkdir /var/www/$1/dev
chmod 755 /var/www/$1/dev
chown root:root /var/www/$1/dev
echo "\$AddUnixListenSocket /var/www/$1/dev/log" >> /etc/rsyslog.d/60-sftp.conf
#
```

```
# Set the sticky bit to try to ensure that newly created files have the correct per
#
find /var/www/$1/webroot -type d -exec chmod gu+sx '{}' \;

#
# Start up our virtual host
#
a2ensite www.$1
#
# restart Apache
#
service apache2 restart
```

We also need some other files a template file, called template, and files called default and zzz-deny-generic, three of which will go in /etc/apache2/sites-available, and another file, namevirtualhost22 that will go in /etc/apache2/conf.d. All of these files have been sent to you separately, but for completeness, here they are:

template

```
#
#   Virtual Host template  updated 2014-09-10
#
#   This is a template for setting up new virtual hosts on Apache under Debian
#   with some default security and other settings.
#   _domain_ and _ip_ are replaced by the actual domain name and ip number when the
#   this template through sed
#
#   Once the domain is set up you may need to hand edit the resulting config file
#   to meet the specific requirements of a particular virtual host. Don't forget
#   to comment your changes.
#
# (c) Chris Stephenson 2014
#
# Licensed under GNU Public License v3 or later
#
# No warranty explicit or implied that this script is fit for purpose or even that
#
#



#
#   Match this virtual host to _domain_ and www._domain_ and all other subdomains o
```

```
#
<VirtualHost _ip_:80>
    ServerName www._domain_
    ServerAlias _domain_
    ServerAlias *._domain_
    #
    # set up a standard webmaster. You may want to make this a common address for t
    # otherwise you should be sure to set up the appropriate mail account or alias.
    #
    ServerAdmin webmaster@_domain_
    #
    # Document root set up in the standard way. apachevirtualhostconfig will have c
    #
    DocumentRoot /var/www/_domain_/webroot
    <Directory /var/www/_domain_/webroot>
    # .htaccess and .htpasswd banned
        AllowOverride none
 # prevent directory listings
     Options -Indexes
    # allow access to any documents not explicitly banned
        Order Allow,Deny
    # everyone can look
        Allow from all
    </Directory>
    #
    # If you plan to support any foreign characters, utf-8 is best, but not globall
    # So we enable utf-8 in each domain separately via this template
    #
    AddDefaultCharset utf-8
    AddCharset utf-8 .html
    #
    # As a default we serve index.html if no filename is specified in a directory
    # It may be necessary to add index.htm or index.php or other names to this line
    # if we need these served as default. Order is significant.
    #
    <IfModule mod_dir.c>
        DirectoryIndex index.html
    </IfModule>
    #
    # Prevent php scripts going off and reading files from outside this directory.
    #
    php_admin_value open_basedir "/var/www/_domain_/"
    #
    # Set up a log file in /var/log/apache2 for this virtual host
    #
    CustomLog /var/log/apache2/_domain__access.log combined
    ErrorLog /var/log/apache2/_domain__error.log
</VirtualHost>
```

default

```
# default
#
# This is the braces to zzz-deny's belt
# Any non matching virtual host requests will default here and be denied
#
# (c) Chris Stephenson 2014
#
# Licensed under GNU Public License v3 or later
#
# No warranty explicit or implied that this script is fit for purpose or even that
#
# Replace _IP_ with the ip no for the server. Repeat if more than one ip no served
#
<VirtualHost _IP_:80>
ServerName deny
ServerAdmin webmaster@localhost

DocumentRoot /var/www/

<Directory />
AllowOverride None
Order allow,deny
deny from all
</Directory>


ErrorLog ${APACHE_LOG_DIR}/error.log

# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

zzz-deny-generic

```
# zzz-deny-generic
#
# (c) Chris Stephenson 2014
#
# Licensed under GNU Public License v3 or later
#
# No warranty explicit or implied that this script is fit for purpose or even that
#
# This is the belt to default's braces
# It tries to explicitly match any stray virtual host requests and deny them
# Not foolproof and needs to be done again for each ip on the server
# I couldn't find a better way.
# Insert the correct IP in the VirtualHost line. If there is more than one IP on th
# whole VirtualHost block with that IP at the beginning.
#
#

<VirtualHost _IP_:80>
   # match just about anything
ServerName *
ServerAlias *.*
ServerAlias *.*.*
ServerAlias *.*.*.*
ServerAlias *.*.*.*.*
ServerAdmin webmaster@localhost
DocumentRoot /var/www/
# ban any access
<Directory />
AllowOverride None
Order allow,deny
deny from all
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Finally we need a little file to ensure compatibility with Apache 2.2, the version we are working with. This file can be removed when we upgrade to Apache 2.4.

namevirtualhost22

```
# namevirtualhost22
#
# (c) Chris Stephenson 2014
#
# Licensed under GNU Public License v3 or later
#
# No warranty explicit or implied that this script is fit for purpose or even that
#
#This file can be removed when we upgrade to Apache 2.4
# While using 2.2 add a line for each IP number and port in use for virtual hosts
NameVirtualHost _IP_:80
```

Using the files

Save the files on your own computer, in some sensible directory in your home, call it virthostfiles (you will need to create the directory).

You can upload them to your home directory on the server by typing this on your own computer:

```
rsync -avz -e"ssh -p1729"  virthostfiles yourname@yourdomain:
```

Then look in you home directory /home/yourname on the server and you will see the files there.

Now you need to copy them to their final destination:

Four of them go to the apache config directories

```
mv /home/yourname/virthostfiles/template /etc/apache2/sites-available
mv /home/yourname/virthostfiles/default /etc/apache2/sites-available
mv /home/yourname/virthostfiles/zzz-deny-generic /etc/apache2/sites-available
mv /home/yourname/virthostfiles/namevirtualhost22 /etc/apache2/conf.d
```

But the correct place for a user written script is in /usr/local/bin

```
mv /home/yourname/virthostfiles/apachevirtualhostconfig  /usr/local/bin
```

We need to make sure our script is executable, so

```
chmod a+x /usr/local/bin/apachevirtualhostconfig
```

We also need to edit default, zzz-deny-generic and namevirtualhost22 to refer to our current IP.

If you do not edit these files by hand to contain the correct IP number, then several things will not work.

So open the files in an editor and replace `_IP_` with the actual IP number of the server

Finally, open /etc/apache2/ports.conf and remove the NameVirtualHost *:80 line from that file.

Now let's look at the apache2 config directories:

```
ls -l /etc/apache2/sites-enabled
```

You see just one (the default) site is currently enabled.

Now let us enable our deny site, which is just there to try to catch requests we do not want to serve.

```
a2ensite zzz-deny-generic
ls -l /etc/apache2/sites-enabled
```

Now we can see what a2ensite is doing. Actually it is also doing a lot of checks behind the scenes. That is why we use it.

Now to make some real websites

Using our real IP number and the script, to which we give three parameters like this:

```
apachevirtualhostconfig yourdomain yourdomainsftp <yourip>
```

We could have used any user name there, but it makes sense to have a system and stick to it.As the script runs, you will enter a password. Right clicking with the mouse, copy paste it and take the opportunity to save it somewhere else on your own computer.

Did that work?

If so, you are lucky! If not, give me feedback. Copy paste any error messages and we can debug. Feeling even luckier? Try another one.

```
apachevirtualhostconfig yourdomain01 yourdomain01sftp <yourip>
```

If that looks like it worked, now is the time to try it out.

Firstly are our sites now serving correctly?

From your browser try browsing to:

```
yourdomain
```

(this should work)

```
www.yourdomain
djhgj.yourdomain
```

(These two will not work unless you define these subdomains in your /etc/hosts file)

```
yourdomain01
```

(this should work)

```
www.yourdomain01
```

(This will not work unless you define this “”– subdomain in your /etc/hosts file) and so on.

You should see the appropriate page for each virtual host.

You could even try setting up another host.

Now let us try some things that should not work

Try browsing to

```
http://<yourip>
```

You should get an error message

Try browsing to a name that is set up in our local /etc/hosts, but not on the server

```
yourdomain04
```

You should get an error message

Assignment

1. Finish up what you started in class. Be prepared to show what you have done in the next lab class.
2. In particular, next week you *must* be able to show your assistant that your server is now serving pages in more than one domain.
3. Write up an account of what you did and what you learned. What was wrong or missing in these instructions? Could you now explain what you have done so far to a friend?
4. Post your account to online.bilgi.edu.tr along with a link to your droplet.