Apache HLS Origin Server

Jul 2017

Introduction

First of all, this is a rather crude implementation of a Webdav server for HLS, while this configuration works, do not use this in a production environment, its full of holes as far as security goes.

However, if you need an Origin server for HLS pretty fast, this will get you going in almost no time, you can follow this guide step-by-step, or you can use the installation script and have everything done for you in a matter of seconds.

This guide assumes that you are going to use the path /var/www/html/webdav/ to store your HLS output(s), if you wish to use another location then you are going to have to read this page very carefully to make the necessary changes.

Install Apache

First we need to install Apache (yes I know the new kid on the block is nginx, but I haven't tried it yet). To install Apache (handily called httpd on CentOS/Red Hat):

```
sudo yum install httpd -y
```

Test Webdav is present

We need to ensure that Webdav is present, we can check this with the following command:

```
sudo httpd -M | grep fs
```

You should get an output like:

```
dav_fs_module (shared)
```

You can ignore the domain name error if you get one.

Create Webdav Directory

Next we need to create the location where we will publish our HLS outputs to, this will be /var/www/html/webdav and I suggest you don't change this if you are not sure how this all works.

```
mkdir /var/www/html/webdav
```

Set webdav Ownership and Rights

Next, change the ownership (to the Apache user) and the permissions for the webdav directory with the following commands:

```
sudo chown -R apache:apache /var/www/html/
sudo chmod -R 755 /var/www/html/
sudo chmod -R g+s /var/www/html
```

Password Authentication

It is important to secure your webdav directory with a password. You can do this by creating an .htpasswd file.

To create it, run the following command:

```
sudo htpasswd -c /etc/httpd/.htpasswd dev
```

You will be prompted to enter a password, then confirm it. This will create a password file for the user dev.

Now, you need to assign group ownership of the file to the Apache user, and lock down the permissions for everyone else. To do this, run the following commands:

```
sudo chown root:apache /etc/httpd/.htpasswd
sudo chmod 640 /etc/httpd/.htpasswd
```

Configure an Apache vhost for WebDAV

Next, you need to create a virtual host file for the webdav directory. Start by creating a new site configuration file called webdav.conf:

```
sudo nano /etc/httpd/conf.d/webdav.conf
```

Add the following content: <sxh [text], gutter: false; highlight: 1-20;> DavLockDB /var/www/html/DavLock <VirtualHost *:80>

```
     ServerAdmin webmaster@localhost
     DocumentRoot /var/www/html/webdav/
  ErrorLog /var/log/httpd/error.log
  CustomLog /var/log/httpd/access.log combined
     Alias /webdav /var/www/html/webdav
     <Directory /var/www/html/webdav>
         DAV On
         AuthType Basic
         AuthName "webdav"
         AuthUserFile /etc/httpd/.htpasswd
         Require valid-user
     </Directory>
```

</VirtualHost> </sxh> Now, restart Apache to activate the new configuration:

```
sudo apachectl restart
```

Test WebDav

Finally, WebDAV is ready for testing. Here, we will use a browser and a client to check WebDAV.

Test with a web browser
To test whether the authentication is working correctly or not, open your web browser and navigate to the URL

```
http://your.server.ip/webdav/
```

You should be prompted with a dialogue box to enter the username and password:

Username: dev Paswword: What you entered earlier (hope you remembered).

You need to test you can create files, I suggest using a application called CyberDuck, some information here:CyberDuck.

Automatic Script

Automatic you say? Yep that's right, just copy this to your server, chmod 777 on it, and run it. Your server must have access to the internet or it won't be able to download the files.

Before you run this file, please check this page to ensure you don't have any Carriage returns in your script:Remove Special Chars from Windows files

You can download the script from

HERE

.

<sxh [text], gutter: false; highlight: 1-41;> sudo yum install nano -y sudo yum install httpd -y sudo mkdir /var/www/html/webdav sudo chown -R apache:apache /var/www/html/ sudo chmod -R 755 /var/www/html/ sudo chmod -R g+s /var/www/html sudo htpasswd -c /etc/httpd/.htpasswd dev sudo chown root:apache /etc/httpd/.htpasswd sudo chmod 640 /etc/httpd/.htpasswd sudo touch /etc/httpd/conf.d/webdav.conf chkconfig httpd on

echo “DavLockDB /var/www/html/DavLock <VirtualHost *:80>

```
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/webdav/
ErrorLog /var/log/httpd/error.log
CustomLog /var/log/httpd/access.log combined
    Alias /webdav /var/www/html/webdav
    <Directory /var/www/html/webdav>
        DAV On
        AuthType Basic
        AuthName "webdav"
        AuthUserFile /etc/httpd/.htpasswd
        Require valid-user
    </Directory>
```

</VirtualHost>” > /etc/httpd/conf.d/webdav.conf

echo “# This file controls the state of SELinux on the system. # SELINUX= can take one of these three values: # enforcing - SELinux security policy is enforced. # permissive - SELinux prints warnings instead of enforcing. # disabled - No SELinux policy is loaded. SELINUX=disabled # SELINUXTYPE= can take one of three two values: # targeted - Targeted processes are protected, # minimum - Modification of targeted policy. Only selected processes are protected. # mls - Multi Level Security protection. SELINUXTYPE=targeted” > /etc/selinux/config

echo “Please Reboot...” </sxh>
Once rebooted you can use a Browser and CyberDuck to test access.