





Run Your Own Video-Sharing Website with YouPHPTube on Debian 10 Buster

 Last Updated: August 11, 2019  Xiao Guo An (Admin)  0
Comment  Debian

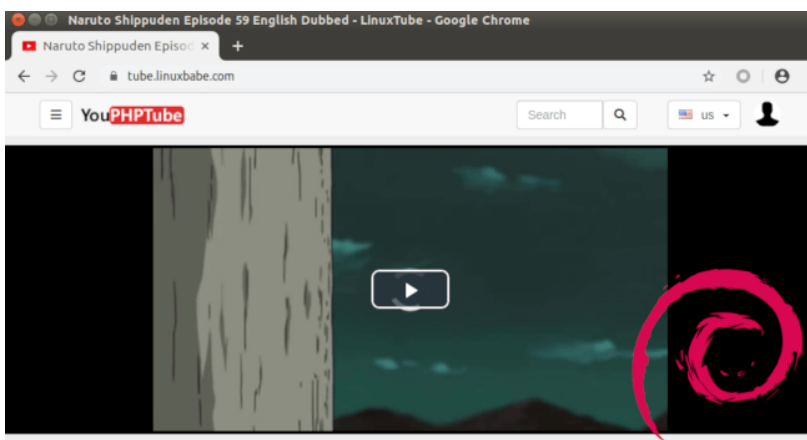
This tutorial is going to show you how to run your own video-sharing website with YouPHPTube on Debian 10 Buster.

YouPHPTube is an open-source, self-hosted alternative to video-sharing websites like YouTube, Vimeo, etc. With YouPHPTube, you can upload video to your own server and share it with the Internet.



YouPHPTube Features

- Unlimited and simultaneous live streams and you can have unlimited users connected to each live stream.
- Import and encode videos from other sites directly from the Internet
- Make private videos
- Totally integrated function with Youtube, letting you sync your videos with Youtube
- Transcode multiple formats of multimedia files to MP4, WebM and MP3 file
- Support Amazon S3 (Simple Storage Service)
- Responsive site layout. YouPHPTube looks incredible on any device. (You can choose YouTube or Netflix style layout.)
- User channels and user playlists
- Subscribe to channels you like
- You can monitor the performance of your videos with up-to-date metrics and reports in YouPHPTube Statistics.
- Extend site functionalities with plugins.
- SEO Optimized
- iOS and Android apps available



Prerequisites of Installing YouPHPTube on Debian 10 Buster



First, you need a Debian server with at least 2 CPU cores and 4GB RAM as a starting point. Obviously, if your video site gets more users, you need to upgrade your hardware as well. If you are looking for a VPS (virtual private server), then you can click [this special link](#) to create an account on Vultr to get \$50 free credit (for new users only). Vultr also provides dedicated servers if you want to go that route. Once you have an account at Vultr, install Debian 10 Buster on your server and follow the instructions below.

YouPHPTube requires PHP and MySQL/MariaDB. To follow this tutorial, you should have already set up a LAMP stack or LEMP stack. If you prefer to use Apache web server, then install LAMP stack.

- [How to Install LAMP \(Apache, MariaDB, PHP\) stack on Debian 10 Buster](#)

If you prefer to use Nginx web server, then install LEMP stack.

- [How to Install LEMP \(Nginx, MariaDB, PHP\) stack on Debian 10 Buster](#)

Note: YouPHPTube support with Nginx is only experimental. You may have issues if you use Nginx.

You also need a domain name. I registered my domain name at [NameCheap](#) because the price is low and they give whois privacy protection free for life.

YouPHPTube is comprised of two parts.

- The main part is the front end streamer server, which allow users to watch existing videos.
- The other part is the encoder, which allow users to upload their videos.

The public encoder uses the domain name `encoder.youphptube.com`. If you don't want users to leave





your site when uploading videos, you need to set up your private encoder. I will show you how in this article. Without further ado, let's get started.

Step 1: Download YouPHPTube on Debian 10 Buster Server

Log into your Debian 10 Buster server via SSH. Then go to the web root directory.

```
cd /var/www/
```

Clone the YouPHPTube streamer repository from Github.

```
sudo apt install git
sudo git clone https://github.com/YouPHPTube/
YouPHPTube.git
```

Then go into the directory.

```
cd YouPHPTube/
```



Clone the YouPHPTube encoder repository from Github.



```
sudo git clone https://github.com/YouPHPTube/
YouPHPTube-Encoder.git
```

It will be saved as `YouPHPTube-Encoder`. We rename it to `upload`, so users can have a better idea what this URL is for when uploading videos.

```
sudo mv YouPHPTube-Encoder upload
```

Next, we need to make `www-data` (the web server user) as the owner of the web root.

```
sudo chown www-data:www-data /var/www/YouPHPT
ube/ -R
```

Step 2: Install Dependencies

YouPHPTube uses `FFmpeg` to encode videos. We can easily install `FFmpeg` from the default Debian repository.

```
sudo apt install ffmpeg
```

To read and write meta information in multimedia files, we need to install the `libimage-exiftool-perl` package.

```
sudo apt install libimage-exiftool-perl
```

To fetch videos from other sites, we need to install `YouTube-DL`. Though it's included in the Debian repository, but it's outdated. We will install `YouTube-DL` from the Python Package Index, which always contains the latest version of `YouTube-DL`.

```
sudo apt install python3-pip
sudo -H pip3 install youtube-dl
```

It's very important that you use the latest version, or you might not be able to download videos from other sites. We can create a Cron job to automatically check and install the latest version.

```
sudo crontab -e
```

Add the following line at the end of the Crontab file to try upgrading YouTube-DL daily.

```
@daily sudo -H pip3 install --upgrade youtube
-dl
```

Step 3: Create Database and User in MariaDB

Log into MariaDB database server with the following command. Since MariaDB is now using `unix_socket` plugin to authentication user login, there's no need to enter MariaDB

root password. We just need to prefix the `mysql` command with `sudo`.

```
sudo mysql
```

Then create a database for YouPHPTube. This tutorial names the database `YouPHPTube`. You can use whatever name you like.

```
create database YouPHPTube;
```

Create the database user. Again, you can use your preferred name for this user. Replace `your-password` with your preferred password.

```
create user youphptube@localhost identified by 'your-password';
```

Grant this user all privileges on the `youthptube` database.

```
grant all privileges on YouPHPTube.* to youphptube@localhost;
```

We also need to create a separate database for the encoder.

```
create database YouPHPTubeEncoder;
```

Create a user for this database.

```
create user encoder@localhost identified by '  
your-password';
```

Grant privileges.

```
grant all privileges on YouPHPTubeEncoder.* t  
o encoder@localhost;
```

Flush the privileges table and exit.

```
flush privileges;  
  
exit;
```

Step 4: Setting Up Web Server

We can use Apache or Nginx web server.

Apache

If you prefer Apache, create a virtual host file for YouPHPTube with a command line text editor like Nano.

```
sudo nano /etc/apache2/sites-available/youphp  
tube.conf
```

Put the following text into the file. Replace the red-colored text with your actual data. I use a subdomain in this tutorial. You can use your main domain name if you prefer. Don't forget to set DNS A record for the domain name in your DNS record

manager. Also note that the document root directory is case-sensitive and you should not add a forward slash at the end.

```
<VirtualHost *:80>
    ServerName tube.yourdomain.com
    DocumentRoot /var/www/YouPHPTube

    <Directory /var/www/YouPHPTube>
        DirectoryIndex index.php
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/youphptube.error.log
    CustomLog ${APACHE_LOG_DIR}/youphptube.access.log combined

</VirtualHost>
```

Save and close the file. Then enable this virtual host.

```
sudo a2ensite youphptube.conf
```

We need to enable the rewrite module.

```
sudo a2enmod rewrite
```

Restart Apache for the changes to take effect.

```
sudo systemctl restart apache2
```

Now visit tube.yourdomain.com and you will be redirected to the setup wizard page



(tube.yourdomain.com/install/index.php). If you see the default Apache page instead of the setup wizard, then you need to disable the default virtual host.

```
sudo a2dissite 000-default.conf
```

And restart Apache.

Before entering any information in the setup wizard, we need to enable HTTPS.

Nginx

If you prefer Nginx, create a youphptube.conf file in /etc/nginx/conf.d/ directory.

```
sudo nano /etc/nginx/conf.d/youphptube.conf
```

Put the following text into the file. Replace the red-colored text with your actual data. I use a subdomain in this tutorial. You can use your main domain name if you prefer. Don't forget to set DNS A record for the domain name in your DNS record manager. Also note that the document root directory is case-sensitive.

```
server {  
    listen      80;  
    server_name tube.yourdomain.com;  
  
    root /var/www/YouPHPTube;  
    index index.php index.html index.htm;  
    charset utf-8;  
    client_max_body_size 2G;  
  
    location / {  
        try_files $uri $uri/ /index.php?$quer
```



```
y_string;
    }

    access_log /var/log/nginx/youthtube.access.log;
    error_log /var/log/nginx/youthtube.error.log;

    location ~ /\.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/run/php/php7.3-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location = /upload/index.php {
        rewrite ^(.*)$ /upload/view/index.php last;
    }

    location = /upload/isAdmin {
        rewrite ^(.*)$ /upload/view/isAdmin.php last;
    }

    location = /upload/removeStreamer {
        rewrite ^(.*)$ /upload/view/removeStream

er.php last;
    }

    location = /upload/priority {
        rewrite ^(.*)$ /upload/view/priority.php
```





```
last;
}

location = /upload/status {
    rewrite ^(.*)$ /upload/view/status.php 1
ast;
}

location = /upload/serverStatus {
    rewrite ^(.*)$ /upload/view/status.php?se
rverStatus=1 last;
}

location = /upload/upload {
    rewrite ^(.*)$ /upload/view/upload.php la
st;
}

location = /upload/listFiles.json {
    rewrite ^(.*)$ /upload/view/listFiles.jso
n.php last;
}

location = /upload/deleteQueue {
    rewrite ^(.*)$ /upload/view/deleteQueue.p
hp last;
}

location = /upload/saveConfig {
    rewrite ^(.*)$ /upload/view/saveConfig.ph
p last;
}
;
```



```
location = /upload/youtubeDl.json {
    rewrite ^(.*)$ /upload/view/youtubeDl.jso
n.php last;
```





```
}

location = /upload/send.json {
    rewrite ^(.*)$ /upload/view/send.json.php
last;
}

location = /upload/streamers.json {
    rewrite ^(.*)$ /upload/view/streamers.js
n.php last;
}

location = /upload/queue.json {
    rewrite ^(.*)$ /upload/view/queue.json.ph
p last;
}

location = /upload/queue {
    rewrite ^(.*)$ /upload/view/queue.php las
t;
}

location = /upload/login {
    rewrite ^(.*)$ /upload/objects/login.jso
n.php last;
}

location = /upload/logoff {
    rewrite ^(.*)$ /upload/objects/logoff.jso
n.php last;
}
```

```
location /upload/ {
    rewrite "^/getImage/([A-Za-z0-9=/]+)/([A-
Za-z0-9]{3})$" /upload/objects/getImage.php?b
ase64Url=$1&format=$2 last;
```



```
rewrite "^/getImageMP4/([A-Za-z0-9=]+)/
([A-Za-z0-9]{3})/([0-9.]+" /upload/objects/
getImageMP4.php?base64Url=$1&format=$2&time=
$3 last;
}

location /upload/getSpiritsFromVideo {
    rewrite ^/getSpiritsFromVideo/([A-Za-z0-9
=]+)/([0-9]+)/([0-9]+)$ /upload/objects/getS
piritsFromVideo.php?base64Url=$1&tileWidth=$2
&totalClips=$3 last;
}

location /upload/getLinkInfo {
    rewrite ^/getLinkInfo/([A-Za-z0-9=]+)/$ /
upload/objects/getLinkInfo.json.php?base64Url
=$1 last;
}
}
```

Save and close the file. Then test Nginx configuration.

```
sudo nginx -t
```

If the test is successful, reload Nginx for the changes to take effect.

```
sudo systemctl reload nginx
```

Now visit `tube.yourdomain.com` and you will be redirected

to the setup wizard page

(`tube.yourdomain.com/install/index.php`). Before entering any information in the setup wizard, we need to enable HTTPS.

Step 5: Enabling HTTPS

To encrypt the HTTP traffic, we can enable HTTPS by installing a free TLS certificate issued from Let's Encrypt. Run the following command to install Let's Encrypt client (certbot) on Debian 10 Buster server.

```
sudo apt install certbot
```

If you use Apache, you also need to install the Certbot Apache plugin.

```
sudo apt install python3-certbot-apache
```

And run this command to obtain and install TLS certificate.

```
sudo certbot --apache --agree-tos --redirect  
--hsts --staple-ocsp --email you@example.com  
--d tube.yourdomain.com
```

If you use Nginx, install the Certbot Nginx plugin.

```
sudo apt install python3-certbot-nginx
```

Next, run the following command to obtain and install TLS certificate.

```
sudo certbot --nginx --agree-tos --redirect -  
-hsts --staple-ocsp --email you@example.com -  
d tube.yourdomain.com
```

Where

- `--nginx`: Use the nginx plugin.
- `--apache`: Use the Apache plugin.
- `--agree-tos`: Agree to terms of service.
- `--redirect`: Force HTTPS by 301 redirect.
- `--hsts`: Add the Strict-Transport-Security header to every HTTP response. Forcing browser to always use TLS for the domain. Defends against SSL/TLS Stripping.
- `--staple-ocsp`: Enables OCSP Stapling. A valid OCSP response is stapled to the certificate that the server offers during TLS.

The certificate should now be obtained and automatically installed.

```
IMPORTANT NOTES:  
- Congratulations! Your certificate and chain have been saved at:  
  /etc/letsencrypt/live/tube.linuxbabe.com/fullchain.pem  
  Your key file has been saved at:  
  /etc/letsencrypt/live/tube.linuxbabe.com/privkey.pem  
  Your cert will expire on 2019-11-09. To obtain a new or tweaked  
  version of this certificate in the future, simply run certbot again  
  with the "certonly" option. To non-interactively renew *all* of  
  your certificates, run "certbot renew"  
- If you like Certbot, please consider supporting our work by:  
  
  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate  
  Donating to EFF: https://eff.org/donate-le
```

Step 6: Finish Installation with the Setup Wizard





Now go to `tube.yourdomain.com` and the setup wizard will appear. On the left side, you can see if your server meet the requirements.

The screenshot shows the YouPHPTube setup wizard interface. On the left, there are four status boxes: three green boxes indicating successful checks for Apache 2.4.38, PHP 7.3.4-2, and a writable videos directory; and two red boxes indicating failed checks for `post_max_size` (8M) and `upload_max_filesize` (2M). On the right, there are input fields for 'Your Site URL' (https://tube.linuxbabe.com/), 'System Path to Application' (/var/www/YouPHPTube/), 'Title of your Web Site' (YouPHPTube), 'Contact E-mail' (placeholder: Enter e-mail contact of your Web Site), 'Database Host' (localhost), and 'Database Port' (3306).

If you use Nginx, you can ignore the following message.

Your server is nginx/1.14.2, you must install Apache.

As you can see, the PHP `post_max_size` should be at least 100M and `upload_max_filesize` should be at least 100M. We need to edit the `php.ini` file to change the two values. If you use Apache web server, then

```
sudo nano /etc/php/7.3/apache2/php.ini
```

If you use Nginx with PHP-FPM, then you need to edit the FPM version of `php.ini` file.

```
sudo nano /etc/php/7.3/fpm/php.ini
```

Find the following two lines.

```
post_max_size = 8M
upload_max_filesize = 2M
```



Change the values according to the video file size you are going to upload. For example, if you allow uploading 1G video





file, then

```
post_max_size = 1024M
upload_max_filesize = 1024M
```

Save and close the file. Restart Apache for the changes to take effect.

```
sudo systemctl restart apache2
```

If you use Nginx with PHP-FPM, then restart PHP-FPM.

```
sudo systemctl restart php7.3-fpm
```

Now refresh the YouPHPTube install wizard page. Your server should pass all requirements. Now on the right side, you can enter a title your video site. Enter the YouPHPTube database name, user and password your created earlier. (Note: You should use the main database, not the encoder database. Also be aware of case-sensitive.) And set a system admin password.



Your Site URL**System Path to Application****Title of your Web Site****Contact E-mail****Database Host****Database Port****Database User****Database Password****Database Name** **Do you want to create database and tables?****Select the main Language****System Admin password****Confirm System Admin password**

Click the **Install now** button and YouPHPTube should be installed successfully.

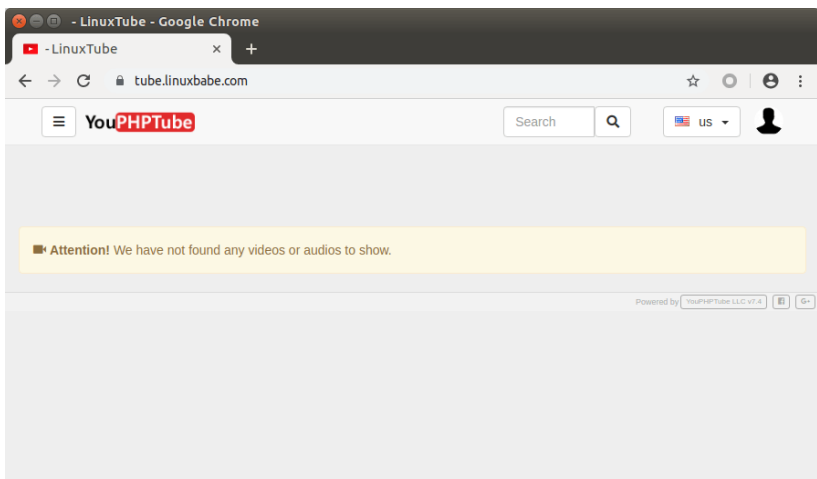
✔ Your system is installed, remove the `/var/www/YouPHPTube/install` directory to continue

[Go to the main page](#)

Now we need to remove the
`/var/www/YouPHPTube/install/` directory.

```
sudo rm /var/www/YouPHPTube/install/ -r
```

Click the Go to the main page button and you will see the YouPHPTube main page.



You can click the drop-down menu on the left and login as the admin user and with the password you set just a few moments ago.

A sign-in form titled 'Please sign in'. It contains three input fields: 'User' with the value 'admin', 'Password' with masked characters, and a 'Remember me' toggle switch that is turned on. Below the password field is a link that says 'I forgot my password'. At the bottom is a green button with a right-pointing arrow and the text 'Sign in'.

Step 7: Set Up the Encoder

Go to <https://tube.yourdomain.com/upload/>. You will be redirected to the Encoder setup wizard. If you are not automatically redirected, then you can manually enter the setup wizard URL.

```
https://tube.yourdomain.com/upload/install/index.php/
```

The left hand side will show you if your server meet the requirements of YouPHPTube Encoder.

✓ Apache/2.4.38 (Debian) is Present

✓ PHP 7.3.4-2 is Present

✓ Exiftool [/usr/bin/exiftool] is Present

✓ FFMPEG /usr/bin/ffmpeg is Present

✓ Your videos directory is writable

✗ Your max_execution_time is 30, it must be at least 7200
▸ Details

✓ Your post_max_size is 1024M

✓ Your upload_max_filesize is 1024M

✗ Your memory_limit is 128M, it must be at least 512M
▸ Details

If you use Nginx, you can ignore the following message.

Your server is nginx/1.14.2, you must install Apache.

As you can see the PHP max_execution_time should be at least 7200 and PHP memory_limit should be at least 512M. We need to edit the php.ini file to change the two values. If you use Apache web server, then

```
sudo nano /etc/php/7.3/apache2/php.ini
```

If you use Nginx with PHP-FPM, then you need to edit the FPM version of php.ini file.

```
sudo nano /etc/php/7.3/fpm/php.ini
```

Find the following two lines.

```
max_execution_time = 30
memory_limit = 128M
```

Change the values .

```
max_execution_time = 7200
memory_limit = 512M
```

Save and close the file. Restart Apache for the changes to take effect.

```
sudo systemctl restart apache2
```

If you use Nginx with PHP-FPM, then restart PHP-FPM.

```
sudo systemctl restart php7.3-fpm
```

Now refresh the YouPHPTube Encoder install wizard page. Your server should pass all requirements. Now on the right side, you need to enter the YouPHPTubeEncoder database name, user and password you created earlier. Be aware of case-sensitive and enter the streamer site's admin password.

Your Site URL

System Path to Application

Database Host

Database User

Database Password

Database Name

Allowed YouPHPTube Streamers Sites (One per line. Leave blank for public) ?

Default Priority ?

Do you want to create database and tables?

YouPHPTube Streamer Site URL ?

YouPHPTube Streamer Site admin User

YouPHPTube Streamer Site admin Password

Click the Install Now button and YouPHPTube Encoder will be installed.

✔ Your system is installed, remove the `/var/www/YouPHPTube/upload/install` directory to continue

[Go to the main page](#)

Remove the install directory.

```
sudo rm /var/www/YouPHPTube/upload/install/ -  
r
```





And go to the main page.

Please sign in

YouPHPTube Streamer Site

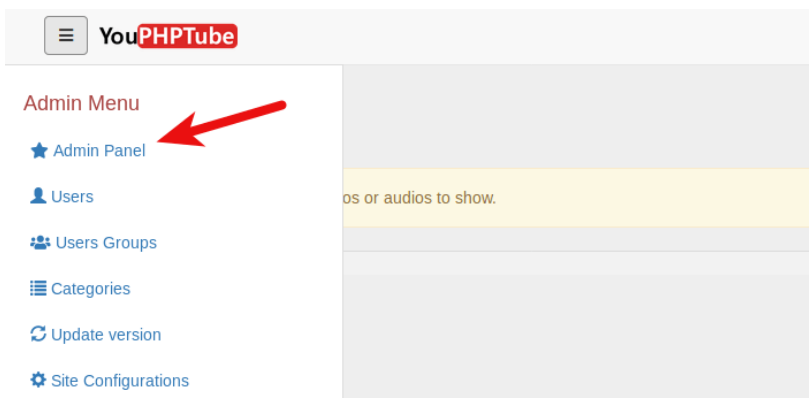
User

Password

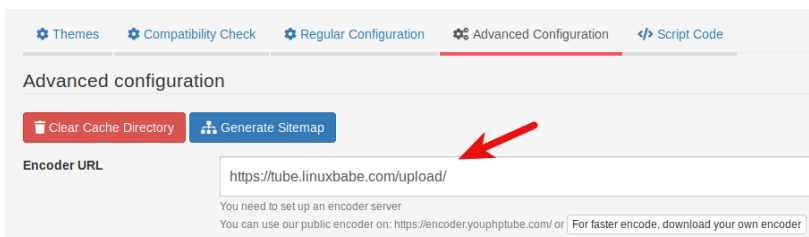
[Sign in](#)

Step 8 : Change the Encoder URL

By default, YouPHPTube uses the public encoder for uploading videos. To use your own encoder, sign in as admin in YouPHPTube (not YouPHPTube encoder) and go to the admin panel from the left side admin panel.



Go to settings -> site settings -> advanced configuration. Change the Encoder URL to `https://tube.yourdomain.com/upload/`. Save the settings, clear the cache directory and generate a new sitemap.



Now log out and log back in. If you click the `encode video` and `audio` button on the upper-right corner, you will be taken to your own encoder to upload videos. It should be noted that the encoding speed depends on your server's CPU power.

Step 9: Configure SMTP

To send out emails (such as account registration, password reset, etc), you need to configure a SMTP server. If you would like to use your own mail server to send emails to clients, please check out the following article to set up your own mail server.

- [How to quickly set up an email server on Debian 9 with Modoboa](#)

If you would like to use a SMTP relay service, I recommend Mailjet. You can follow the tutorial below to set up SMTP relay on your YouPHPTube server.

- [How to Set Up Postfix SMTP relay on Debian with Mailjet](#)

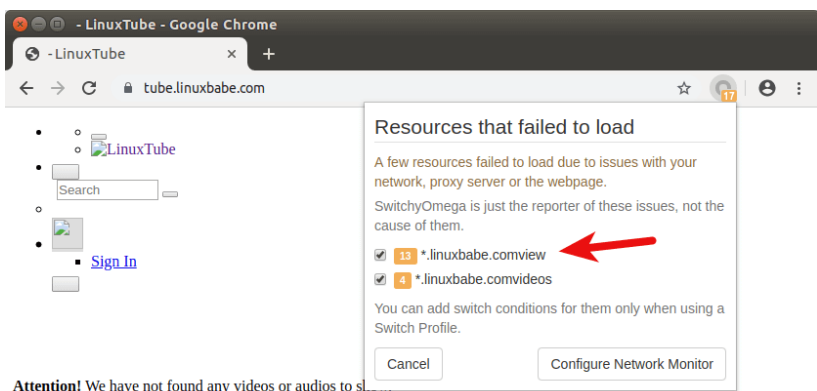
Mobile Apps

You can download the free android app or iOS app by doing a search for YouPHPTube on Google Play Store or YPT Mobile in Apple Store. White-label mobile app require a paid license.

Troubleshooting

If your YouPHPTube main page failed to load some CSS and JavaScript resources,





It's likely that you have added extra forward slash in the Apache or Nginx virtual host file for the web root directory. Instead of using

```
DocumentRoot /var/www/YouPHPTube/
```

You should use

```
DocumentRoot /var/www/YouPHPTube
```

Wrapping Up

I hope this tutorial helped you install YouPHPTube on Debian 10 Buster server. As always, if you found this post useful, then [subscribe to our free newsletter](#) to get more tips and tricks.

Take care ☐

Rate this tutorial

■ ■ ■ [Total: 1 Average: 5]

Business Software
 Debian
 Debian Server
 Linux
 Self Hosted
 YouPHPTube

You may also like:





**How To
Instal...**

**Set Up
Open...**

**How to
Instal...**

**How to
Instal...**

**How To
Instal...**

**2 Ways
to...**

**Set Up
Auth...**

**Getting
Starte...**

