# StreamView

## StreamView - Web server and web app Installation Walkthrough:

Stream View is a Day One Monetizable automated script and needs very little managing once it is setup. Follow this step-by-step installation guide to run the entire script on your live server. We recommend Digital Ocean server for hosting space, and we'll be using that for the example. We've been building products for 5 years and these guys have a good standing support for a running a scalable mobile app setup. You can check it out here : DigitalOcean.

## Prerequisites for Installation:

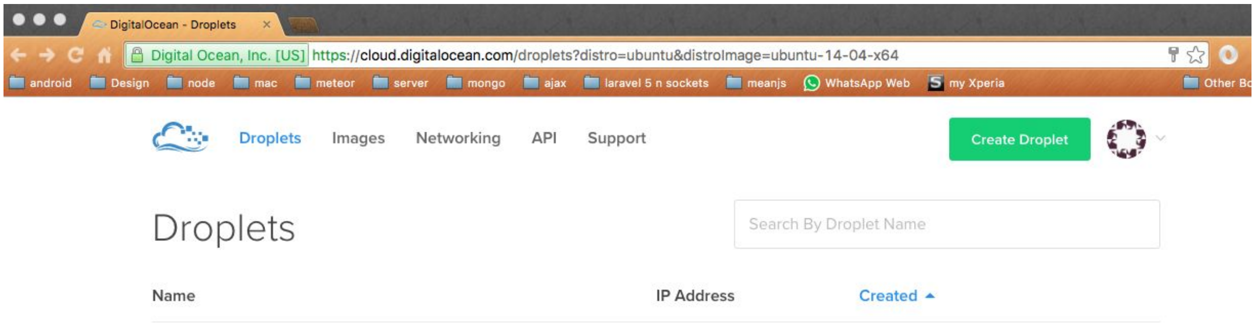
- Digital Ocean account
- Terminal/Command prompt
- Lamp install
- Node install
- NPM install
- Tmux
- Redis server
- FFmpeg
- Nginx
- Cron Job
- Project Setup

**Note**: Only for ubuntu 16.04 steps.

# Server Installation Procedure

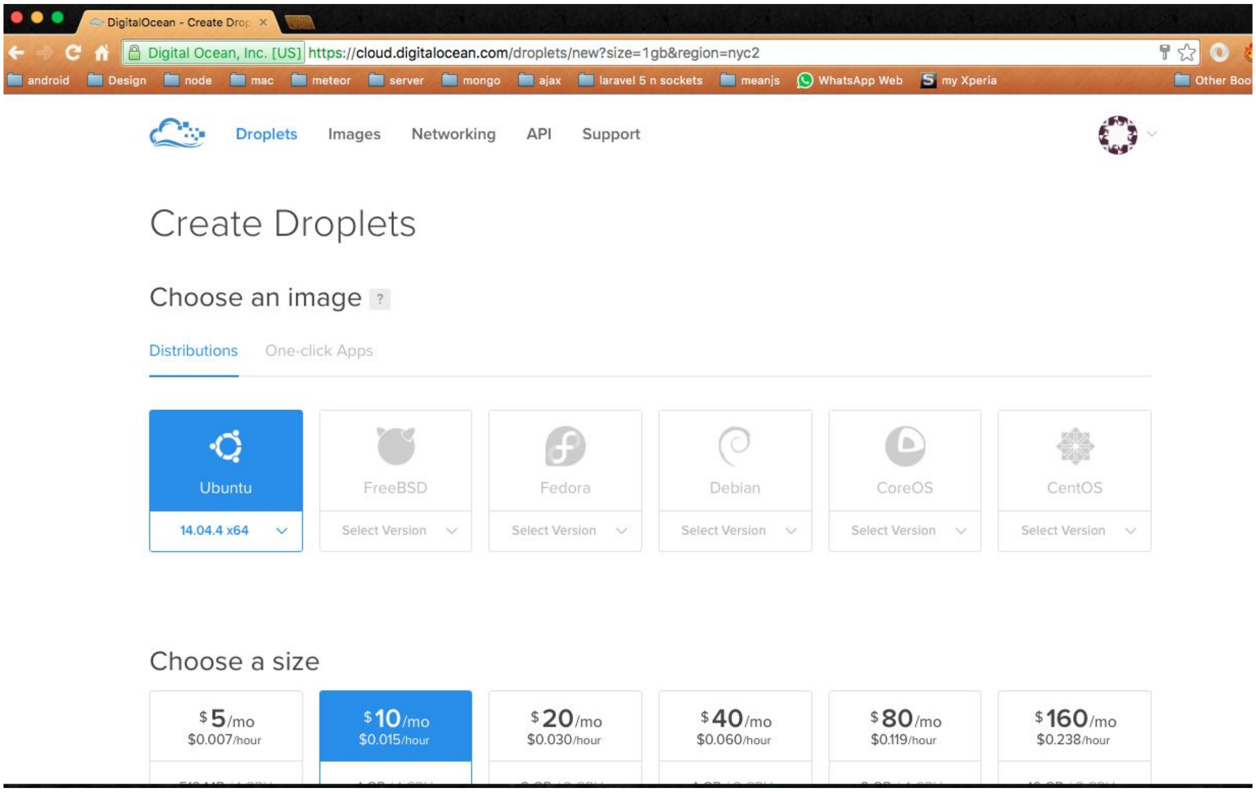
## STEP 1:

After logging into your account, It will take you to the Droplets(server) page, Click on Create Droplet.



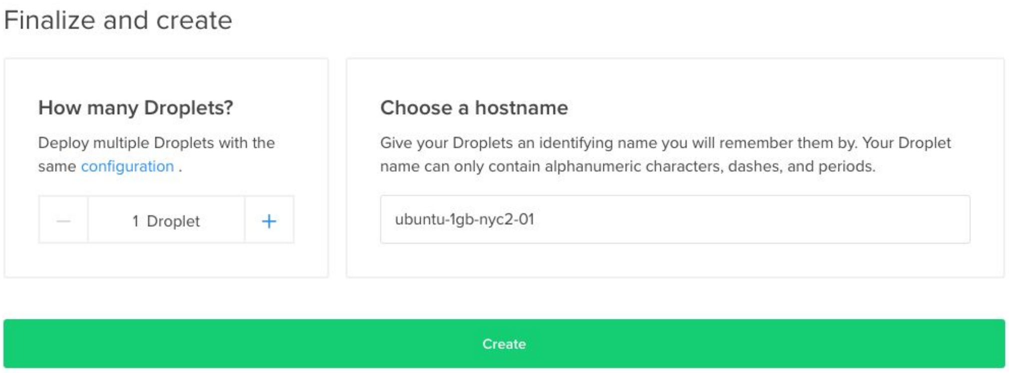
## STEP 2:

Once you click on create droplet, select the **ubuntu** server and we will be needing a minimum of **$10** bandwidth for running a proper streamview service.Choose **ubuntu 16.04**



## STEP 3:

You can leave the other options unchecked and scroll down to end. Specify a hostname if you want and Click **create** to finish creating a droplet.



Finalize and create

How many Droplets?

Deploy multiple Droplets with the same configuration .

Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

ubuntu-1gb-nyc2-01

## STEP 4:

Once a droplet is created, you will get a mail to your account with a **root password** and confirmation for creating a droplet. You can revert back to the digital ocean account and find the **droplet** details listed with **IP address**.

## STEP 5:

After having these details, You can open your Terminal/Command prompt(**Terminal 1**) and Type in the following command :

Open the server without SSH-KEY using this command.

```
ssh root@DROPLET_IP_ADDRESS
```

Open the server with SSH-KEY using this command.

```
ssh -i root-ssh-private-key root@DROPLET_IP_ADDRESS
```

It will prompt you for a password, enter the **root password** that was sent to your mail and press Enter. It will show details of your server and log you in.

## STEP 6:

Now that you have logged into your server from terminal, we'll have to create an alternate user account that you can use to access. From the logged in terminal, type in the following command:

```
adduser DESIRED_USERNAME
```

## STEP 7:

This would have created a User account. Now let's add admin privileges to this Account by typing the following command :

```
gpasswd -a USERNAME sudo
```

Please use the **Username** you selected in **STEP 6** for this.

## STEP 8:

For creating a secure server, we have to create a security key for your local machine. Let's do that by **Opening a new terminal window(Terminal 2)** and typing in the following command:

```
ssh-keygen
```

Once you type this command, It will prompt you for a question, Hit **Enter** without typing anything. It will ask again for a passphrase, Hit **Enter** without typing anything. Now your key is created.

## STEP 9:

Now we'll have to copy the key. To display the key, type in this command:

```
cat id_rsa.pub
```

The key will be a long sequence. Once the key is displayed, select it and copy it using your Mouse or cursor.

## STEP 10:

Let’s add this public key to our user **(switch to TERMINAL 1)** by typing in the following commands in sequence :

```
su - USERNAME
mkdir .ssh
chmod 700 .ssh
nano .ssh/authorized_keys
```

This will open up a blank file in your Server terminal. Paste the copied **ssh key** in this and press **“ctrl/cmd+x”,** it will prompt a question. Type **Y** and press enter again.

```
chmod 600 .ssh/authorized_keys
exit
```

## STEP 11: (Optional)

To not allow default root login and secure the server ssh daemon, type in the following command from logged in **Terminal 1** from the root folder:

```
nano /etc/ssh/sshd_config
```

This will open up the file. Now search for the line **“PermitRootLogin Yes”** and change it to **“PermitRootLogin no”** . Once this is done, let’s restart the ssh service by typing in this command:

```
service ssh restart
```

## STEP 12:

Now let’s close up all our terminals, Open a fresh new Terminal and try connecting to the server.

```
ssh -i ssh-key USERNAME@DROPLET_IP_ADDRESS

(Example)
```

```
ssh -i id_rsa USERNAME@DROPLET_IP_ADDRESS
```

We are don’t have ssh key only have password Means Follow this.

```
ssh  USERNAME@DROPLET_IP_ADDRESS
```

Type in your password and you’ll be logged in as the new User.

## STEP 13:

Now we’ll have to install apache for our servers to run. We can do that by typing these commands in the terminal after logging in as the new user.

```
sudo apt-get update
sudo apt-get install apache2
```

Type in your password when it asks for it. Let each command run and take it’s time.

## STEP 14:

To verify if it’s working or not, you can just open your browser and type in the droplet IP address as the url and check if Apache Ubuntu page is shown. Now, let’s install our database **MySQL**. Type in the following command from the terminal:

```
sudo apt-get install mysql-server
sudo mysql_secure_installation
```

The command line will prompt you for questions if any like MySQL passwords, etc., You can either change them or just hit Enter to continue with the installation.

## STEP 15:

For our script to work, we need php script to run. So let’s install PHP in our server with most needed Mcrypt plugin. Type in the following command:

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update

sudo apt install php7.1

sudo apt install php7.1-cli php7.1-xml php7.1-mysql php7.1-mbstring
php7.1-gettext php7.1-curl php7.1-common php7.1-opcache php7.1-readline
php7.1-mcrypt php7.1-zip
```

Change the Upload limits php.ini file

```
sudo nano /etc/php/7.1/apache2/php.ini

max_execution_time=-1

max_input_time=-1

memory_limit -1

post_max_size =3000M

upload_max_filesize = 2048M
```

After this changes completed ctrl+x and enter the save php.ini file

## STEP 16:

Now that php is installed, we want our server to look for Index.php first than index.html in order. So let's change that by opening up this server file. Type in the following command:

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

This will open up a file that has information like this.

```
<IfModule mod_dir.c>
   DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

**Change it to, cut the index.php and paste the before index.html**

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

## STEP 17:

Let’s restart the service and check to see if the process has gone correctly. Type in the command :

```
sudo service apache2 restart
```

And that’s it for installing PHP,MySQL and apache.

## STEP 18:

Let’s setup phpMyAdmin now for accessing our Databases. Type in the command in the terminal :

```
sudo apt-get install phpmyadmin php-mbstring php-gettext

sudo phpenmod mcrypt

sudo phpenmod mbstring
```

This will prompt a few questions to make the installation correctly. **When the first prompt appears, Hit “SPACE”, then “TAB” and “Enter”** . After this, display the yes or no you click the yes. you will be asked for a database administrator’s password. Next, you will be able to choose and confirm a password for **phpMyAdmin** application itself.

## STEP 19:

After enabling the mcrypt extension, restart apache service.

```
sudo service apache2 restart
```

## STEP 20:

Open the apache conf file . **Eg: sudo nano /etc/apache2/apache2.conf**

Change to this following lines.

```
<Directory />
        Options FollowSymLinks
        AllowOverride None -> Change None to All
        Require all denied
</Directory>

<Directory /usr/share>
        AllowOverride None  -> Change None to All
        Require all granted
</Directory>

<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None  -> Change None to All
        Require all granted
</Directory>
```

**Update the end of the file in this command. After save the apache2.conf file restart the apache2.**

```
Include /etc/phpmyadmin/apache.conf

sudo service apache2 restart
```

Now you can access **phpMyAdmin** by typing in your web browser URL "**http://IP_ADDRESS/phpmyadmin**". Once the page opens, you will be asked for a username and password. Use **"root"** for username and the password you set during phpMyAdmin installation for password and login. You will be able to create/delete databases in your server now.

As of now, all the basic requisites are installed in our server including, php, Mcrypt, MySQL, phpMyAdmin and we have set up a user account for secure access from our local machine.

## STEP 21:

Download the Streamview package in zip file. Now we have to put this code in our server that we have set up till now. Since we added a secure user and secure access to our server, we don't have to use a separate FTP client, we can use a **secure copy** method. Open a new terminal, do not log in to the server, just type this command :

```
scp -r -i id_rsa(ssh-key path) PATH_FOR_STREAMVIEW_PACKAGE
USERNAME@DROPLET_IP:/home/USERNAME
```

**id_rsa** - whatever you create a ssh key that private key path name.

**Note** : “Path for streamview package” means the path your downloaded package is, eg: C:/Users/PC name/Downloads/streamview

After this, You will have the streamview backend and frontend zip file in your server. Unzip the both file for using the this command.

```
sudo  apt-get  install unzip

unzip streamview-website.zip
```

## STEP 22:

### You will get 2 different folder from inside Streamview Folder

1. streamview-frontend
2. streamview-backend

streamview-frontend - It contains only angular js code - user panel.

streamview-backend - It contains API and admin panel.

Admin panel - streamview-backend/

Once this command is typed, it will ask you for the password, type in the password and press enter to securely copy all the code into our server. First, Login to the server using **ssh** **username@droplet_ip_address** command and sublink all the other folders to the root folder by typing in this command.

```
sudo ln -sf /home/USERNAME/Streamview/streamview-backend  /var/www/html
```

We'll have to create a symbolic for streamview-frontend as well.

```
sudo ln -sf /home/USERNAME/Streamview/streamview-frontend  /var/www/html
```

## STEP 23:

### # TMUX Installation:

We'll need config tmux session. So that we can multiple session in server. Let's start configuring tmux session.

```
wget https://aravinth.net/tmux.conf

mv tmux.conf ~/.tmux.conf
```

Open Nano by following link

```
sudo nano ~/.bashrc
```

Paste this in bottom of the File

```
tm() { tmux new -s "$1" ;}
ta() { tmux attach -t "$1"; }
tl() { tmux list-sessions; }
```

Run this command

```
source ~/.bashrc
tm session_name - Create New Session
```

Create multiple session use this key **Alt+n**

Switch the prefix and suffix use the **shift+ arrow** button (left and right)

List All Session

```
tl
```

**Open already existing session**

```
ta session_name
```

**control + b d** // To exit from tmux session without closing window.

## STEP 24:

### # Composer Installation:  ( Optional )

Update the this command in your server for **Username@droplet_name:~$**

```
sudo apt-get install composer

cd streamview-backend

composer update
```

## STEP 25:

### # Redis Server Installation

We’ll have to install redis server now

```
cd streamview-backend
sudo add-apt-repository ppa:chris-lea/redis-server
sudo apt-get update
sudo apt-get install redis-server
```

Now start Redis server using following command

```
sudo service redis-server start
```

If everything is okay, you'll see this message on Terminal

```
Starting redis-server: redis-server
```

## STEP 26:

### # FFmpeg Installation

We’ll have to install FFmpeg

```
sudo add-apt-repository ppa:mc3man/trusty-media
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install ffmpeg
```

### # NPM and NODEJS Installation

Next we need to install nodejs and npm on this server.

```

curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```

```
sudo apt-get install nodejs

sudo apt-get install build-essential

sudo apt-get install nodejs-legacy
```

Once the above process completed, start the npm

```
npm start
```

Switch the new window using **shift+arrow** button (left and right)

If you getting any errors in npm start. Install the npm with in your project.

**# Npm Install: (Optional)**

```
npm install
```

## STEP 27:

We'll have to install nginx server now.

**# NGINX INSTALL START**

```
cd ~
mkdir nginx
cd nginx
```

**# For compiler and git**

```
sudo apt-get install git gcc make
```

**# For the Http rewrite module which requires the PCRE library**

```
sudo apt-get install libpcre3-dev
```

# For SSL modules

```
sudo apt-get install libssl-dev

git clone https://github.com/arut/nginx-rtmp-module

wget http://nginx.org/download/nginx-1.4.3.tar.gz

tar zxpvf nginx-1.4.3.tar.gz

cd nginx-1.4.3

./configure --add-module=/home/USERNAME/nginx/nginx-rtmp-module/
--with-http_ssl_module --prefix=/usr/local/nginx-streaming/

sudo make

sudo make install

cd /usr/local/nginx-streaming/conf

sudo mv nginx.conf nginx.conf.bkp

sudo wget https://aravinth.net/nginx.conf

sudo apt-get update

sudo apt-get install nginx
```

```
Open Nginx Config file -> sudo nano nginx.conf
```

**// You need to change the path in nginx conf file. You will find //**

```
application vod2 {
 play /var/www/html/streamview-backend/public/uploads/videos/original;
  }

application hls {
  live on;
  hls on;
hls_path /var/www/html/streamview-backend/public/uploads/videos/original;
}

location / {
            # Serve HLS fragments
 types {
     application/vnd.apple.mpegurl m3u8;
     video/mp2t ts;
 }
 root /var/www/html/streamview-backend/public/uploads/videos/original;
 add_header Cache-Control no-cache;
}
```

// Start the Nginx

```
sudo /usr/local/nginx-streaming/sbin/nginx
```

// Stop the Nginx

```
sudo /usr/local/nginx-streaming/sbin/nginx -s stop
```

# NGINX INSTALL END

Reference - https://aravinth.net/ngnix-server-configuration-with-rtmp-module/

## STEP 28:

We'll have to create a database using phpMyAdmin. So open up a web browser, Type in the **URL** field : [http://IP_ADDRESS/phpmyadmin](http://IP_ADDRESS/phpmyadmin) and Log in using the username as root and the password you gave during installation.

After logging in, Click on the "**Databases**" tab, enter a desired database name(**Remember this database name**) in the **"create database"** field and click on create database to create an empty database.

Once you've created an empty database, Click on the **"Import"** tab and click on "**choose file**" button. Now, locate the **"Eg: streamview.sql"** database file inside the DB folder from the downloaded package and import it into the empty database. You can see a list of tables now on your left pane after import.

### Import Command:

```
mysql -u root -p database_name <
/home/USERNAME/Streamview/streamview-backend/DB/streamview.sql
```

## STEP 29:

Now from the server logged in terminal, type this following command to go into the **streamview-backend** directory :

### # .env File Configuration

```
cd streamview-backend
```

Open up the .env file using this following command :

```
sudo nano .env
```

After opening up the file, find the following piece of code and give the correct information.

```
DB_HOST=127.0.0.1 (Provide Database host here, default host is 127.0.0.1)
DB_PORT=3306 (Provide Database port here, default port is 3306)
DB_DATABASE= (Provide the Database name that was created in STEP 28)
DB_USERNAME= (Database username provided during configuration)
```

```
DB_PASSWORD= (Database password provided during configuration)

QUEUE_DRIVER=sync To QUEUE_DRIVER=redis

APP_URL =(Update the Admin URL)
```

And for mailer integration, (we are using simple SMTP gmail server Update you gmail username and password)

```
MAIL_USERNAME= (Provide your gmail username)
MAIL_PASSWORD= (Provide your gmail password)
```

After saving the file, revert back to the terminal command line.
Go to /home/USERNAME/streamview-backend/

```
sudo composer dump-autoload
php artisan view:clear
php artisan config:clear
php artisan cache:clear
php artisan config:cache
```

## STEP 30:

The database has been created with base tables and entries. For allowing access to specific folders for databases, type in the following commands in sequence:

We need to give permission - navigate to /streamview-backend/ - you can find storage,public/uploads, bootstrap/cache. Give 777 permission.

```
sudo chmod -R 777 storage
sudo chmod -R 777 public/uploads
sudo chmod -R 777 bootstrap/cache
sudo chmod -R 777 .env
```

## # Folders Creation - (Optional)

If your getting any error in  image uploading from your site do this commands.

Create a empty folders from inside public/uploads folder.

```
cd streamview-backend/public/uploads

mkdir images videos smil subtitles

cd videos

mkdir original
```

## STEP 31:

### # Cron Job Update

Update the Cron job Details for using publish video and payment expiry.

```
cd /home/USERNAME/streamview-backend
sudo crontab -e
```

Enter the 2 for nano editor.Update the end of the line save the file

```
* * * * * curl --request GET 'Admin Site url (Ex: http://domain name.com
)/publish/video'

0 0 * * * curl --request GET 'Admin Site url (Ex: http://domain name.com
)/notification/payment'

* * * * * curl --request GET 'Admin Site url (Ex: http://domain name.com
)/payment/expiry'

* * * * * curl --request GET 'Admin Site url (Ex: http://domain name.com
)/automatic/renewal'
```

### Project Setup

Copy custom ffmpeg into  vendor corresponding folder.

```
Ex: cp app/ffmpeg-custom/olaferlandsen/ffmpeg-php-class/src/FFmpeg.php
vendor/vidhyar2612/ffmpeg-php-class/src/FFmpeg.php
```

Open log file in a new window

Check the storage->log -> inside laravel.log file. After use this command

```
tail -f storage/logs/laravel.log
```

Configure socket url in site **admin  panel settings.**

```
Ex: http://ip_address or domain:3003/
```

Once everything is configured. We need to start queue listen. Open tmux new session. Go to streamview-backend code which will be

### # Start Queue

```
php artisan queue:listen --timeout=0
```

**Don't close the tmux session queue listen has to run in background continuously.**

**control + b d** // To exit from tmux session without closing it.

## STEP 32:

We'll have to configure angularjs. **Navigate to app.js** - It will be in

### # Update Frontend Url

```
cd streamview-frontend/

sudo nano app/app.js
```

Please change the following details.

```
var route_url = "https://domain.com/#"; // user panel url - angular code
( Ex: http://streamhash.com/ )

var apiUrl = "https://api.domain.com/"; // admin panel url
( Ex: http://admin.streamhash.com )

var angularUrl = "https://domain.com/#/"; // user panel url
( Ex: http://streamhash.com )
```

Create symlinks for subtitles and smil folder to /streamview-frontend/

```
sudo ln -sf /home/USER/streamview-backend/public/uploads/smil
/home/USER/streamview-frontend/assets

sudo ln -sf /home/USER/streamview-backend/public/uploads/subtitles
/home/USER/streamview-frontend/assets
```

## STEP 33:

We’ll have to create a virtual host for user panel and admin panel.

Open the virtual host in path

```
cd  /etc/apache2/sites-available

sudo cp 000-default.conf frontend.conf
```

Open the **000-default.conf** and update the server name and document root  - backend configuration

```
sudo nano 000-default.conf

ServerName backend-domain.com -> comments: remove # this

ServerAdmin webmaster@localhost -> comments: Update the you admin email (optional)

DocumentRoot /var/www/html/streamview-backend/public -> comments: update the root directory
```

Save the 000-default.conf file.

Open the **frontend.conf** and update the server name and document root  - frontend configuration

```
sudo nano frontend.conf

ServerName frontend-domain.com -> comments: remove # this

ServerAdmin webmaster@localhost ->  comments: Update the you admin email (optional)

DocumentRoot /var/www/html/streamview-frontend/ -> comments: update the root directory
```

Save the frontend.conf file.

Here is the reference link to create a virtual host.

https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-14-04-lts

After this process complete do the following commands.

```
sudo a2ensite frontend
sudo a2enmod rewrite
sudo service apache2 restart
```

## STEP 34:

This completes the server setup and installation for Streamview. Now please open up a browser and type in the URL : **http://admin.domain.com** and login with the default username and password.

### Admin Demo Login:

URL: http://admin.domain.com/admin/login

Username : admin@streamview.com

Password : 123456

### User Demo Login:

URL: http://user.domain.com/#/

Username: user@streamview.com

Password: 123456

### Moderator Demo Login

URL: http://admin.domain.com/moderator/login

Username: moderator@streamview.com

Password: 123456

## STEP 35:

Once the installation is completed

Configure the  RTMP, HLS url from admin -> settings -> video settings.

Configure the frontend url from admin -> settings -> site settings -> App URL.

## STEP 36:

Installation is complete. Browse around and check all features.
Start making money and invite us for your launch party :)

**IMPORTANT :** If you are having trouble making the installation, you can always opt for our premium installation package for just $79. This way, you can relax and we will get back to you with admin access to your Streamhash venture,You can reach us at contact@streamhash.com for further queries or talk to us on live chat at StreamHash.