

Stream View

StreamView 1.2 web server and web app Installation Walkthrough :

Stream Now 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](#).

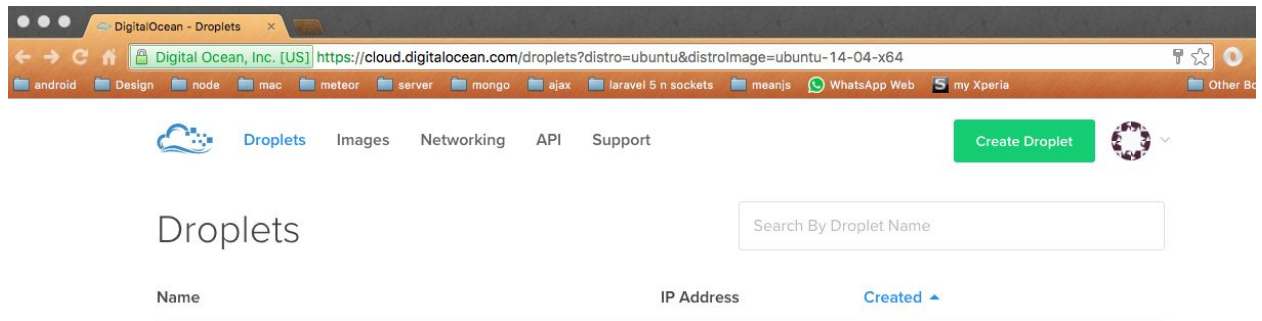
Pre-requisites for installation:

- [Digital Ocean account](#)
- Terminal/Command prompt
- Ngnix
- FFmpeg
- Redis server

Server Installation Procedure

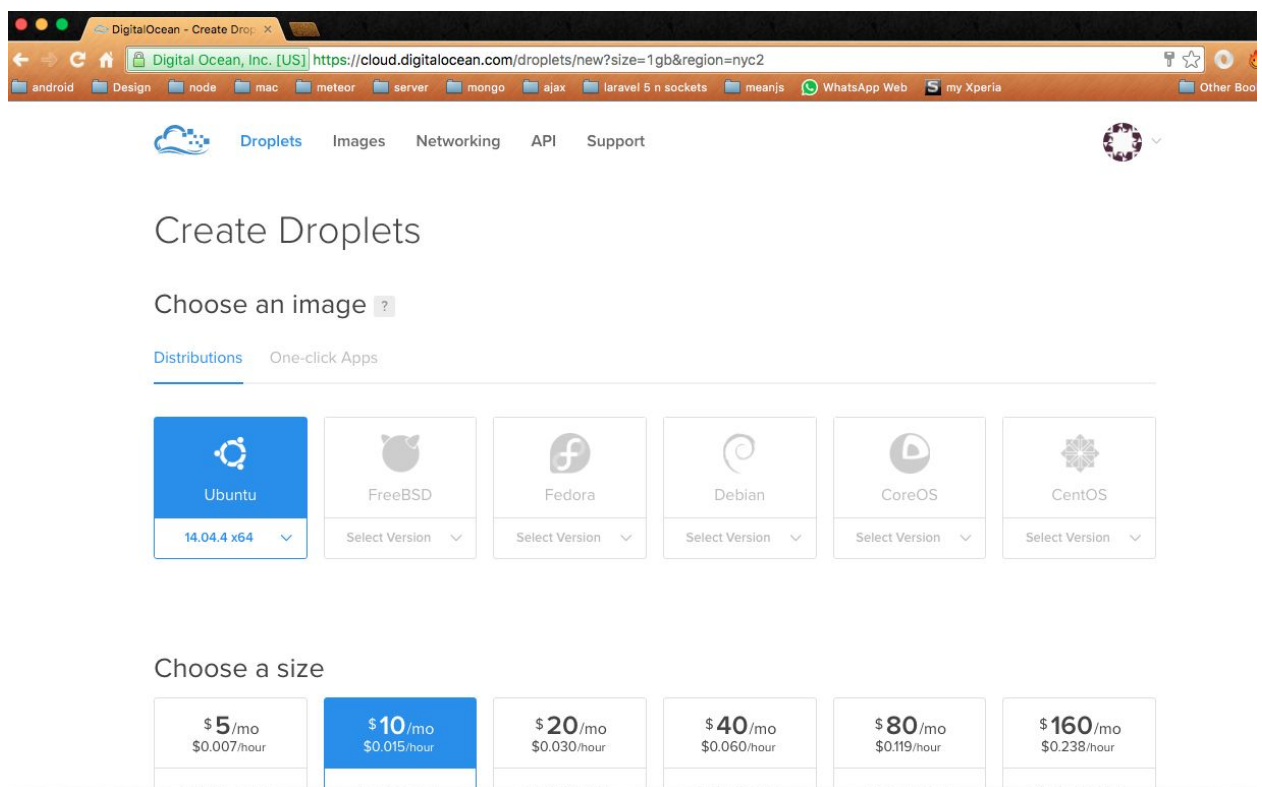
STEP 1:

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.



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](#) .

—

1 Droplet

+

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

Create

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 :

```
ssh 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 ~/.ssh/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 :

1. **su - USERNAME**
2. **mkdir .ssh**
3. **chmod 700 .ssh**
4. **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.

5. **chmod 600 .ssh/authorized_keys**
6. **exit**

STEP 11:

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 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.

1. **sudo apt-get update**
2. **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:

1. **sudo apt-get install mysql-server php5-mysql**
2. **sudo mysql_install_db**
3. **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 apt-get install php5 libapache2-mod-php5 php5-mcrypt

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,

```
<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
```

This will prompt a few questions to make the installation correctly. **When the first prompt appears, Hit "SPACE", then "TAB" and "Enter"** . After this, 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:

We have to enable to **mcrypt** extension that we installed previously in an explicit way. Type in the command :

```
sudo php5enmod mcrypt
```

After enabling the mcrypt extension, restart apache service.

```
sudo service apache2 restart
```

STEP 20:

Now you can access **phpMyAdmin** by typing in your web browser URL "http://DROPLET_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 Stream Now package and extract it. 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 PATH_FOR_STREAMVIEW_PACKAGE  
USERNAME@DROPLET_IP:/home/USERNAME
```

Note : "Path for streamview package" means the path your downloaded package is, eg: C:/Users/PCname/Downloads/streamview

STEP 22:

You will get 2 different 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 -s /home/USERNAME/streamview-backend /var/www/html
```

Please let each command in this step to take its time to download node JS and execute the plugins. Once each command finishes, type in the next one.

STEP 24:

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

1. wget <http://aravinth.net/tmux.conf>
2. mv tmux.conf ~/.tmux.conf

3. Open Nano by following link
4. `sudo nano ~/.bashrc`
5. Paste this in bottom of the File
6. `tm() { tmux new -s "$1" ;}`
7. `ta() { tmux attach -t "$1"; }`
8. `tl() { tmux list-sessions; }`
9. Run this command
10. `source ~/.bashrc`
11. Open A new Session
12. `tm session_name`
13. List All Session
14. `tl`
15. Open already existing session
16. `ta session_name`

Reference link - <http://www.hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/>

STEP 25:

We'll have to install redis server now

`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

Now open .env and do the changes - `QUEUE_DRIVER=sync` To `QUEUE_DRIVER=redis`

STEP 26:

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`

Here is the reference link -

<https://www.faqforge.com/linux/how-to-install-ffmpeg-on-ubuntu-14-04/>

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 xzpf 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
```

```
mv nginx.conf nginx.conf.bkp
```

```
wget http://aravinth.net/nginx.conf
```

```
// You need to change the path in nginx conf file.
```

```
// You will find
```

```
application vod2 {
```

```
play /var/www/html/streamview-backend/uploads/videos/original;
```

```
}
```

```
application hls {
```

```
    live on;
```

```
    hls on;
```

```
hls_path /var/www/html/streamview-backend/uploads/videos/original;
```

```
}
```

```
location / {
```

```
    # Serve HLS fragments
```

```
    types {
```

```
        application/vnd.apple.mpegurl m3u8;
```

```
        video/mp2t ts;
```

```
    }
```

```
root /var/www/html/streamview-backend/uploads/videos/original;  
add_header Cache-Control no-cache;  
}
```

// Please check all the path are correct.

```
sudo cp /home/USERNAME/nginx/nginx-rtmp-module/stat.xsl /var/www/  
sudo apt-get update  
sudo apt-get install nginx  
sudo /usr/local/nginx-streaming/sbin/nginx
```

NGINX INSTALL END

Here is the reference link - <http://aravinth.net/nginx-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://DROPLET_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 “**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.

STEP 29:

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

```
cd streamview-backend
```

Open up the .env file using this following command :

```
nano .env
```

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

DB_HOST= (Provide your Database URL here)
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)
SOCKET_SERVER= http://DROPLET_IP_ADDRESS:3000 (The URL should have domain name and port id of node server)

And for mailer integration, (we are using simple SMTP gmail server)

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

QUEUE_DRIVER=redis
// for configure redis server.

After saving the file, revert back to the terminal command line.

Go to /home/USERNAME/streamview-backend/

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 0777 storage
sudo chmod -R 0777 public/uploads
sudo chmod -R 0777 bootstrap/cache

STEP 31:

Once everything is configured. We need to start queue listen. Open tmux session.

```
tm streamview // Create new session.  
tmux new-window // Create a new window  
Go to streamview-backend code which will be  
cd /home/USERNAME/streamview-backend  
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 **streamview-frontend/app/app.js**

```
nano app.js
```

Please change the following details.

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

```
var apiUri = "https://api.domain.com/"; // admin panel url Ex:  
http://admin.streamhash.com.
```

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

STEP 33:

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

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

STEP 34:

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

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>

STEP 35:

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.

Username : admin@streamview.com

Password : 123456

STEP 36:

Once the installation is completed - you need to configure RTMP, HLS and Frontend URL in admin panel - Settings - other settings.

STEP 37:

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 livechat at StreamHash.