Installing mShots on a Production Server

Adnan Haque
4:34 pm on October 22, 2021

Recently I installed mShots on a VPS. I had to take some steps apart from those mentioned in the docs and needed to do some changes in my system as well. So, I am documenting the process of starting a complete production setup.

Cloud Provider and Distro

I created a **2GB memory 1 vCPU** droplet on DigitalOcean. Initially, I tried to use Ubuntu, but it doesn't seem to have VFB(Virtual Frame Buffer) in its Kernel. So I installed **Debian 11**.

Preparing the Environment

As the first step, you have to install the required packages. For simplicity, I am running all commands as root.

```
apt install nodejs npm apache2 php7.4 memcached curl php7.4-
{curl,gd,memcache}
```

mShots need to be placed on `/opt/mshots` to work. So we will clone it and place it there.

```
cd /opt && git clone https://github.com/Automattic/mShots &&
mv mShots mshots
```

We need to add a symlink so that mShots can find node.

```
mkdir -p /usr/local/node/bin && ln -s /usr/bin/node
/usr/local/node/bin/node
```

We need to configure the webserver to serve the mshots API. We will use apache2. As the first step, we need to change the web directory to `/opt/src/public_html` . So we need to edit the config file: `/etc/apache2/sites-available/000-default` and change **DocumentRoot** to `/opt/src/public_html` .

To make sure, the API can support slashes in the URL of the web pages we are taking screenshots of, we also need to add the following line to the same file.

```
AllowEncodedSlashes On
```

```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        AllowEncodedSlashes On
        ServerAdmin webmaster@localhost
        DocumentRoot /opt/mshots/public_html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

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

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Next, we need to allow .htaccess override on our web directory. So these lines need to be added on `/etc/apache2/apache2.conf` .

```
<Directory /opt/mshots/public_html>
        Options Indexes FollowSymLinks
        AllowOverride all
        Require all granted
</Directory>
```

Of course we need to enable mod-rewrite:

```
a2enmod rewrite
```

We need to install chrome and related dependencies.

```
wget -q -O - https://dl-
ssl.google.com/linux/linux_signing_key.pub | apt-key add

sh -c 'echo "deb [arch=amd64]
http://dl.google.com/linux/chrome/deb/ stable main" >>
/etc/apt/sources.list.d/google.list'

apt-get update && apt-get install -y google-chrome-unstable
fonts-ipafont-gothic fonts-wqy-zenhei fonts-thai-tlwg fonts-
kacst fonts-freefont-ttf --no-install-recommends
```

Running mShots

mShots should not run as root as, chrome needs to be sandboxed for security. So,
we will add a new user and hand over everything to that.

```
adduser mshots
chown -R mshots:mshots mshots
```

To start mShots, we can switch to that user...

```
su mshots
```

And finally, run it

```
./mshots_ctl.sh start
```

Using

If you have done everything correctly, you can now access mShots on:

```
<server>/mshots/v1/example.com
```

The first request to a URL is always redirected to a default image as it queues for snapping. Once the screenshot is ready you can access the screenshot on the same URL on subsequent requests.

---

1 reply

Mark George 4:51 am on October 25, 2021

Nice work @adnan007 ! 🙂