

## Аппаратные и программные требования для работы ПО «Видеостена»

Минимальные требования для работы

- ОС: Ubuntu 16.04 LTS;
- Процессор: 2\*2GHz;
- Оперативная память: 2 Гб;
- Размер диска: 30 Гб.

Это могут быть как физические сервера, так и виртуальные машины\контейнеры.

## Центральный сервер хранения и сбора изображений с филиалов.

Размер системного диска следует выбирать в зависимости от предполагаемого объема хранимых видеозаписей, либо смонтировать дополнительный раздел внутри директории программы в: /var/www/archieve/video.

Для работы программы на стороне центрального сервера хранения понадобятся установленные:

- Mysql/MariaDB
- PHP версии не ниже 7.4
- Apache/Nginx
- ffmpeg
- Node.js

Для работы в веб-интерфейсе на стороне клиента необходим браузер Google Chrome (последняя стабильная версия)

### Процесс установки ПО (На примере ОС Ubuntu 22.04)

- 1) Подключиться к серверу, на который будет устанавливаться ПО консольно, либо по SSH
- 2) Скопировать из архива, например, с помощью программы WinSCP файлы php скриптов директорию сервера:  
/var/www/
- 3) Скопировать из архива, например, с помощью программы WinSCP файлы bash скриптов в директорию сервера /root/scripts

- 4) Скопировать из архива, например, с помощью программы WinSCP дампы базы данных, например в домашний каталог пользователя, где admin –логин вашего пользователя:

```
/home/admin/videowall.sql
```

- 5) Обновить все установленные пакеты:

```
sudo apt update
sudo apt upgrade
```

- 6) Установка nginx:

```
sudo apt install nginx -y
```

- 7) Настройка nginx:

создать и заполнить конфиг для сайта, где your.domain заменить на ваш домен, по которому будет доступен сайт:

```
sudo mcedit /etc/nginx/sites-available/wall.your.domain.conf
```

```
server {
    listen 0.0.0.0:80;

    root /var/www/;
    index index.php index.html index.htm;

    location / {
        autoindex on;
        autoindex_exact_size on;
        autoindex_localtime on;
        charset utf-8;

        add_header Access-Control-Allow-Origin *;

        # pass the PHP scripts to FastCGI server listening on 1$
        #
        location ~ \.php$ {
            try_files $uri =404;
            fastcgi_split_path_info ^(.+\.php)(.*)$;
            fastcgi_pass unix:/run/php/php8.1-fpm.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            include fastcgi_params;
        }
    }
}
```

```
}  
}
```

```
sudo rm /etc/nginx/sites-enabled/default  
sudo ln -s /etc/nginx/sites-available/wall.your.domain.conf /etc/nginx/sites-  
enabled/wall.your.domain.conf
```

```
sudo systemctl restart nginx
```

#### 8) Установка MariaDB:

```
sudo apt install mariadb-server  
sudo mysql_secure_installation
```

Enter current password for root – нажать enter

Switch to unix\_socket authentication - n

Set root password? – n

Далее на все вопросы - y

#### 9) Создание базы данных и пользователя с правами на неё

```
sudo mysql -u root
```

```
CREATE DATABASE videowall;
```

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON videowall.* TO 'user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

#### 10) Импорт дампа базы:

```
sudo mysql -u root videowall < /home/admin/videowall.sql
```

#### 11) Установка PHP:

```
sudo apt install php-common php8.1-cli php8.1-common php8.1-fpm php8.1-json  
php8.1-mysql php8.1-opcache php8.1-readline
```

#### 12) При желании можно добавить в DNS А-запись для домена wall.your.domain, ведущую на сервер (или в файл HOSTS).

- 13) Файлы из директории со скриптами запускаем планировщиком /etc/cron.d/wall к примеру так для забора каждую минуту с 1 филиала, каждые 5 минут с 2, и каждые 10 минут с 3 филиала.:

```
SHELL=/bin/bash
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=""
```

```
*/1 * * * * root /home/root/scripts/filial1.sh > /dev/null 2>&1;
```

```
/home/root/scripts/thumbnailizer filial1
```

```
*/5 * * * * root /home/root/scripts/filial2.sh > /dev/null 2>&1;
```

```
/home/root/scripts/thumbnailizer filial2
```

```
*/10 * * * * root /home/root/scripts/filial3.sh > /dev/null 2>&1;
```

```
/home/root/scripts/thumbnailizer filial3
```

```
0 0 * * * root /home/root/scripts/spisok-video.sh > /dev/null 2>&1
```

```
55 9 * * * root killall wget
```

```
58 9 * * * root killall ffmpeg
```

- 14) Директории с уменьшенными изображениями лучше хранить в оперативной памяти, что бы снизить нагрузку на диск. В файле /etc/fstab добавляем строки:

```
tmpfs /var/www/thumb tmpfs
```

```
rw,size=94M,noexec,nodev,nosuid,uid=33,gid=33,mode=1777 0 0
```

```
tmpfs /var/www/thumbs tmpfs
```

```
rw,size=64M,noexec,nodev,nosuid,uid=33,gid=33,mode=1777 0 0
```

При необходимости, размер можно изменять.

- 15) Удаляем сгенерированные видео старше указанного количества дней. 90 в данном примере, с помощью задания в планировщике /etc/cron.d/deletes

```
SHELL=/bin/bash
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=""
```

```
15 0 * * * root /usr/bin/find /var/www/archieve/video -type f -mtime +90 -delete > /dev/null 2>&1
```

```
30 0 * * * root /usr/bin/find /var/www/archieve/ -iname "*.jpg" -type f -mtime +40 -delete > /dev/null 2>&1
```

```
* * * * * root /usr/bin/find /var/www -type f -name "*.jpg" -size 0 -delete > /dev/null 2>&1
```

Две последние строки защищают от ошибок.

16) Устанавливаем nodejs.

```
sudo apt install nodejs
```

17) Настраиваем запуск службы в файле /etc/cron.d/node-js

```
SHELL=/bin/bash
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=""
```

```
* * * * * admin /usr/bin/flock -w 0 /tmp/videowall.lock /home/root/socket/start.sh
```

18) Файл /root/scripts/filial1.sh должен быть заполнен таким образом:

```
#!/bin/sh
```

```
mkdir /var/www/archieve/
```

```
mkdir /var/www/archieve/ filial1/
```

```
mkdir /var/www/archieve/ filial1/camera1/
```

```
mkdir /var/www/archieve/ filial1/camera2/
```

```
cp /var/www/filial1-camera1.jpg /var/www/archieve/filial1/camera1/camera1-`date +%d-%m-%Y-%H%M%S`.jpg
```

```
wget http://192.168.50.31:81/cam_videowall/filial1-camera1.jpg -O "/var/www/ filial1-camera1.jpg" --tries=1
```

```
cp /var/www/filial1-camera2.jpg /var/www/archieve/filial1/camera2/camera2-`date +%d-%m-%Y-%H%M%S`.jpg
```

```
wget http://192.168.50.31:81/cam_videowall/filial1-camera2.jpg -O "/var/www/ filial1-camera2.jpg" --tries=1
```

Где:

- 192.168.50.31 = ip адрес маршрутизатор удалённого филиала.
- 81 = порт по которому настроен port forward на маршрутизаторе в web порт удалённого сервера.

19) Файл /root/scripts/spisok-video.sh должен быть заполнен так:

```
#filial1
```

```
sh /root/scripts/videocompile.sh filial1 camera1
```

```
sh /root/scripts/videocompile.sh filial1 camera2
```

# Сервер для сбора изображений с камер на филиале.

Минимальные требования для работы

- ОС: Ubuntu 16.04 LTS;
- Процессор: 2\*2GHz;
- Оперативная память: 2 Гб;
- Размер диска: 30 Гб.

Это могут быть как физические сервера, так и виртуальные машины\контейнеры.

Для сбора изображений с камер и их передачи нам понадобятся установленные:

- Apache/Nginx
- ffmpeg
- curl
- wget

## Процесс установки ПО (На примере ОС Ubuntu 22.04)

- 1) Подключиться к серверу, на который будет устанавливаться ПО консольно, либо по SSH
- 2) Создадим папки  

```
mkdir -p /var/www/html/cam_videowall  
mkdir -p /root/scripts
```
- 3) Сменим владельца  

```
chown -R www-data:www-data /var/www/html/cam_videowall
```
- 4) Скопировать из архива, например, с помощью программы WinSCP файлы bash скриптов, получающие и сжимающие картинки с камер в директорию сервера:  

```
/root/scripts
```
- 5) Когда камер много, запись картинок на диск отъедает много ресурсов диска, по этому можно использовать файловую систему в оперативной памяти, с помощью добавления строки в `/etc/fstab`  

```
tmpfs /var/www/html/cam_videowall tmpfs  
rw,size=64M,noexec,nodev,nosuid,uid=33,gid=33,mode=1700 0 0
```

Где `uid=33,gid=33` - ID пользователя и группы `www-data`
- 6) Делаем все скрипты исполняемыми.

```
chmod -R u+x /root/scripts
```

- 7) Настроим планировщик забирать изображение с камер каждую минуту в файле  
/etc/cron.d/videowall

```
SHELL=/bin/bash  
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin  
MAILTO=""
```

```
* * * * * root /root/scripts/getwget.sh filial1 camera1  
http://admin:admin@192.168.1.220/cgi-bin/jpg/image.cgi
```

```
* * * * * root /root/scripts/getwget.sh filial1 camera2  
http://admin:admin@192.168.1.221/cgi-bin/jpg/image.cgi
```

Где:

- `http://admin:admin@192.168.1.221/cgi-bin/jpg/image.cgi` = ссылка на статичное изображение с камеры, зависит от модели камеры, в этом примере приведена камера производителя beward ссерии BD.
- `admin:admin` = логин:пароль от камеры.
- `getwget.sh` = метод забора картинки (http через wget)

Не каждая камера может отдавать статичную картинку через http протокол, ниже пример строки для планировщика, с камеры №3 по RTSP протоколу.

```
* * * * * root /root/scripts/getrtsp.sh filial1 camera3  
rtsp://admin:admin@192.168.1.222/live/sub
```

В данном примере приведена камера производителя trassir, в случае с другими моделями ссылка будет другой, ищите в документации производителя камеры.

Если есть выбор, то HTTP протокол более предпочтителен, т.к. работает поверх транспортного протокола TCP, обеспечивающего целостность данных.

Если вы не уверены в работоспособности ссылок, проверьте их с помощью браузера (HTTP) или VLC player (RTSP). Если вы не видите изображений с камер через них, то дальнейшие действия бессмысленны. Обратитесь к документации либо в техническую поддержку производителя IP-камеры.

- 8) Устанавливаем необходимые пакеты  
`sudo apt-get install wget curl nginx ffmpeg`

- 9) Настроим веб сервер:

Создаём конфиг для веб сервера /etc/nginx/sites-available/wall.conf и пишем в него:

```
server {  
    listen    0.0.0.0:80;
```

```

        root /var/www/html/;
        index index.php index.html index.htm;

    location / {
        autoindex on;
        autoindex_exact_size on;
        autoindex_localtime on;
        charset utf-8;

        add_header Access-Control-Allow-Origin *;

        # pass the PHP scripts to FastCGI server listening on 1$
        #
        location ~ /\.php$ {
            try_files $uri =404;
            fastcgi_split_path_info ^(.+\.php)(.*)$;
            fastcgi_pass unix:/run/php/php7.4-fpm.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
            include fastcgi_params;
        }

    }
}

```

10) Делаем символическую ссылку на наш конфиг и выключаем дофолтный  
**sudo ln -s /etc/nginx/sites-available/wall.conf /etc/nginx/sites-enabled/wall.conf**

**sudo rm /etc/nginx/sites-enabled/default**  
**sudo systemctl restart nginx**

# Сервер пользовательской и администраторской веб панелей

Минимальные требования для работы

- ОС: Ubuntu 16.04 LTS;
- Процессор: 2\*2GHz;
- Оперативная память: 2 Гб;
- Размер диска: 30 Гб.

Это могут быть как физические сервера, так и виртуальные машины\контейнеры.

Для работы веб интерфейса нам понадобятся установленные:

- Apache/Nginx
- Mysql/MariaDB
- PHP версии не ниже 7.4

## Процесс установки ПО (На примере ОС Ubuntu 22.04)

1. Подключиться к серверу, на который будет устанавливаться ПО консольно, либо по SSH.
2. Скопировать файлы из архива, например, с помощью программы WinSCP в директорию сервера:  
`/var/www`
3. Скопировать из архива, например, с помощью программы WinSCP дампы базы данных, например в домашний каталог пользователя, где admin –логин вашего пользователя:  
`/home/admin/db-inventa.sql`
4. Обновить все установленные пакеты:

```
sudo apt update  
sudo apt upgrade
```

5. Установка Apache:

```
sudo apt install apache2 apache2-utils -y  
systemctl status apache2  
sudo a2enmod rewrite
```

6. Настройка Apache:

7. Создать и заполнить конфиг для пользовательского сайта, где `your.domain` заменить на ваш домен, по которому будет доступен сайт:

```
sudo mcedit /etc/apache2/sites-available/videowall.your.domain.conf
```

```
<VirtualHost *:80>
DocumentRoot "/var/www/videowall/web/"
ServerName videowall.your.domain
<Directory "/var/www/videowall/web/">
allow from all
#Options None
Options Indexes FollowSymLinks SymLinksIfOwnerMatch
Require all granted
</Directory>
</VirtualHost>
```

```
sudo a2ensite videowall.your.domain.conf
sudo systemctl restart apache2
```

8. Создать и заполнить конфиг для администраторской панели, где `your.domain` заменить на ваш домен, по которому будет доступен сайт:

```
sudo mcedit /etc/apache2/sites-available/adminka.your.domain.conf
```

```
<VirtualHost *:80>
DocumentRoot "/var/www/inv/web/"
ServerName adminka.your.domain
<Directory "/var/www/inv/web/">
allow from all
#Options None
Options Indexes FollowSymLinks SymLinksIfOwnerMatch
Require all granted
</Directory>
</VirtualHost>
```

```
sudo a2ensite adminka.your.domain.conf
sudo systemctl restart apache2
```

9. Изменим упоминания `ip` адреса `1.1.1.1` в конфигах сайта на `ip` адрес или доменное имя Вашего центрального сервера хранения изображений:

```
find /var/www/inv -type f -exec sed -i 's/1.1.1.1/192.168.1.100/g' {} \;
```

Где `192.168.1.100` = адрес Вашего центрального сервера хранения изображений.

10. Создадим символическую ссылку  

```
ln -s /var/www/inv/bundles/app/web /var/www/inv/web/bundles/app
```
11. Установка MariaDB:  

```
sudo apt install mariadb-server
sudo mysql_secure_installation
```

Enter current password for root – нажать `enter`

Switch to unix\_socket authentication - n  
Set root password? – n  
Далее на все вопросы – y

12. Создание базы данных и пользователя, где user – логин, а password - пароль которые будут впоследствии прописаны в настройках сайта, чтобы он имел доступ к базе данных:

```
sudo mysql -u root
```

```
CREATE DATABASE inventa;
```

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
```

```
GRANT ALL PRIVILEGES ON inventa.* TO 'user'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

13. Импорт дампа базы:

```
sudo mysql -u root inventa < /home/admin/db-inventa.sql
```

14. Установка PHP:

```
sudo apt install php php-mbstring php-zip php-gd php-json php-curl libapache2-mod-php
```

15. Добавить в DNS A-запись для домена videowall.your.domain, ведущую на сервер (или в файл HOSTS).

16. Создать файл .htaccess для настройки проверки прав доступа:

```
sudo mcedit /var/www/videowall/.htaccess
```

```
Options +FollowSymLinks
```

```
RewriteEngine On
```

```
RewriteBase /
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule . /index.php [L]
```

```
RewriteRule ^.htaccess$ - [F]
```

```
AuthType Basic
```

```
AuthName "Authorization"
```

Options Indexes FollowSymLinks SymLinksIfOwnerMatch

AuthUserFile /var/www/videowall\_users/.htpasswd

Require valid-user

Order allow,deny

Allow from 127.0.0.1

satisfy any

Options -Indexes

17. Создать файл с учетной записью admin, через который будут авторизоваться пользователи на сайте:

18. `sudo htpasswd -c /var/www/videowall_users/.htpasswd admin`

Последующие пользователи добавляются без ключа -c

19. Шаги 16-18 повторить и для администраторского сайта `adminka.your.domain`.