

Пошаговая инструкция по установке и настройке G-server.

Перед началом установки убедитесь, что у вас есть стабильное подключение к сети Internet и все необходимые файлы для установки G-server:

- 1)Файлы лицензий рабочих мест в формате user.lic
- 2)Файл ключа программы XCR

ВАЖНО!!!

В целях безопасности файл ключа XCR ограничен по времени действия одним днем. То есть, если ключ получен 01.01.2013 то установка и первое подключение к G-server должны производиться ТОЛЬКО в этот день до 24:00. В противном случае вам придется обратиться в компанию ООО «Геоспэйс» для получения актуального ключа.

ВАЖНО!!!

- 3)Файл дистрибутива G-server в формате geos.zip
- 4)Дистрибутив ОС Ubuntu Server 12.04 LTS

Устанавливаем операционную систему – Ubuntu server 12.04 LTS x64. Во время установки, в окне выбора компонентов, отметить чекбокс – Postgres server.

Устанавливаем пакеты PHP.

команда: sudo apt-get install php5 php5-gd php5-cli php5-mcrypt php5-imagick phppgadmin php5-curl

Устанавливаем пакет Ioncube.

скачиваем установочный пакет:

команда: cd /usr/local

команда: sudo wget http://downloads2.ioncube.com/loader_downloads/ioncube_loaders_lin_x86-64.tar.gz

распаковываем пакет:

команда: sudo tar xzf ioncube_loaders_lin_x86-64.tar.gz

убедимся, что расширения загружены в PHP:

команда: echo "zend extension=/usr/local/ioncube/ioncube_loader_lin_5.3.so" | sudo tee /etc/php5/conf.d/ioncube.ini

перезапустим Apache:

команда: sudo /etc/init.d/apache2 restart

Чтобы проверить что ionCube loader установлен, проверите вывод из `phpinfo()`; он должен содержать строку ionCube Loader в Дополнительных Модуляях.

Далее все действия должны выполняться с правами суперпользователя ROOT

команда: sudo su

Разрешим серверу Apache использовать htaccess

отредактируем файл default-ssl.

команда: nano /etc/apache2/sites-available/default-ssl

В блоке DocumentRoot /var/www найти строки AllowOverride none и поменять значение "None" на "All"

```
<----->DocumentRoot /var/www
<-----><Directory />
<-----><----->Options FollowSymLinks
<-----><----->AllowOverride All
<-----></Directory>
<-----><Directory /var/www/>
<-----><----->Options Indexes FollowSymLinks MultiViews
<-----><----->AllowOverride All
<-----><----->Order allow,deny
<-----><----->allow from all
<-----></Directory>
```

Настроим размер выделенной памяти для модуля PHP.

Для этого отредактируем файл php.ini.

команда: nano /etc/php5/apache2/php.ini.

Находим строку «memory_limit» и присвоим ей значение – 512 (по умолчанию – 128)

```
; Maximum amount of memory a script may consume (128MB)
; http://php.net/memory-limit
memory_limit = 512M
```

Настроим сервер Postgres

зайдем под пользователем postgres.

команда: su postgres

команда: psql

зададим пользователю postgres пароль.

команда: ALTER USER postgres WITH PASSWORD 'ваш пароль';

в подтверждение, что вы все сделали верно, в консоли выведется сообщение:
ALTER ROLE

```
postgres=# ALTER USER postgres WITH PASSWORD '██████████';
ALTER ROLE
postgres=# █
```

команда: \q

команда: exit

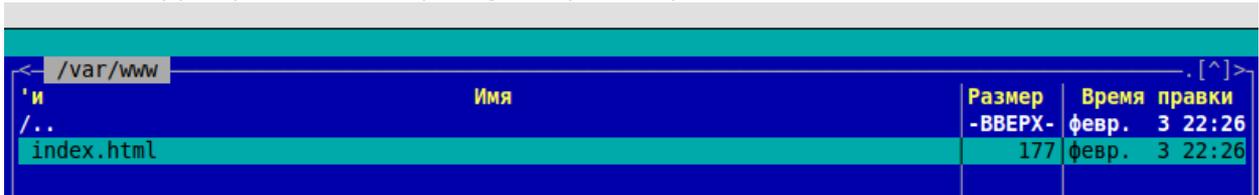
для упрощения дальнейших действий установим файловый менеджер «MC»

команда: apt-get install mc

Установка G-сервера

команда: mc

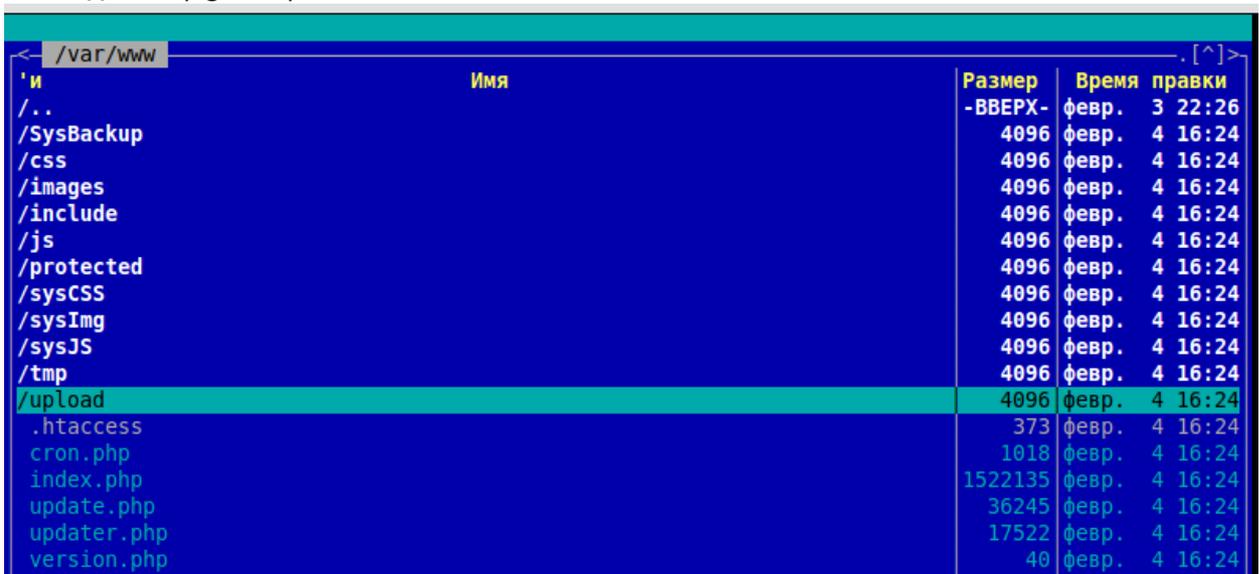
заходим в папку var/www и удаляем файл index.html
копируем установочный файл geos.zip в папку var/www



| Имя | Размер | Время правки |
|------------|---------|---------------|
| .. | -ВВЕРХ- | февр. 3 22:26 |
| index.html | 177 | февр. 3 22:26 |

развернем архив в эту же папку

команда: unzip geos.zip



| Имя | Размер | Время правки |
|-------------|---------|---------------|
| .. | -ВВЕРХ- | февр. 3 22:26 |
| /SysBackup | 4096 | февр. 4 16:24 |
| /css | 4096 | февр. 4 16:24 |
| /images | 4096 | февр. 4 16:24 |
| /include | 4096 | февр. 4 16:24 |
| /js | 4096 | февр. 4 16:24 |
| /protected | 4096 | февр. 4 16:24 |
| /sysCSS | 4096 | февр. 4 16:24 |
| /sysImg | 4096 | февр. 4 16:24 |
| /sysJS | 4096 | февр. 4 16:24 |
| /tmp | 4096 | февр. 4 16:24 |
| /upload | 4096 | февр. 4 16:24 |
| .htaccess | 373 | февр. 4 16:24 |
| cron.php | 1018 | февр. 4 16:24 |
| index.php | 1522135 | февр. 4 16:24 |
| update.php | 36245 | февр. 4 16:24 |
| updater.php | 17522 | февр. 4 16:24 |
| version.php | 40 | февр. 4 16:24 |

зададим права доступа для папки upload

команда: chmod -R 777 upload

зададим права доступа для папки SysBackup

команда: chmod -R 777 SysBackup

зададим права доступа для папки tmp

команда: chmod -R 777 tmp

заходим в папку var/www/protected

зададим права доступа для папки tmp

команда: chmod -R 777 tmp

выходим из MC

Создаем заготовку базы postgres

копируем файл admin.sql в папку /var/www/upload

заходим под пользователем postgres

команда: su postgres

команда: psql

создаем заготовку базы данных

команда: create database geos;

если вы все сделали верно то в консоли появится сообщение:
CREATE DATABASE

```
postgres=# create database geos;  
CREATE DATABASE  
postgres=# █
```

команда: \q

установим структуру базы данных на нашу заготовку

команда: psql geos < schema.sql

```
ALTER TABLE  
REVOKE  
REVOKE  
GRANT  
GRANT
```

создадим пользователя admin, сервера G-server, в базе данных Postgres

команда: psql geos < admin.sql

команда: exit

укажем модулю PHP логин и пароль для доступа к базе данных Postgres

команда: nano /var/www/protected/config/individual.php

в строке «username» вводим – postgres

в строке «password» вводим – пароль, заданный нами ранее, для пользователя postgres

```
Файл Правка Вкладки Справка
/var/www/protected/config/individual.php [----] 53 L:[ 1+ 5 6/ 23] *(167 / 482b) 0039
<?php
return array(
<-----><----->'db'=>array(
<-----><-----><----->'connectionString' => 'pgsql:host=localhost;port=5432;dbname=geos',
<-----><-----><----->'username' => 'postgres',
<-----><-----><----->'password' => '███████████',
<-----><-----><----->),
<-----><-----><----->
<-----><----->'mail' => array(
<-----><-----><----->'transportType'=>'smtp', /// case sensitive!
<-----><-----><----->'transportOptions'=>array(
<-----><-----><-----><----->'host'=>'smtp.gmail.com',
<-----><-----><-----><----->'username'=>'██████████████████████',
<-----><-----><-----><----->'password'=>'███████████',
<-----><-----><-----><----->'port'=>'465',
<-----><-----><-----><----->'encryption'=>'ssl',
<-----><-----><-----><----->),
<-----><-----><----->),
<-----><----->'user'=>array(
<-----><-----><----->'geosTimeout'=> 60*30
<-----><-----><----->),
);
```

Копируем файлы лицензии

файлы лицензий пользователей с расширением .lic копируются в папку -
/var/www/upload/sysuser
файл ключа программы XCR копируется в папку - /var/www/upload/

Настроим сервер Apache на использование соединений по протоколу SSL

запускаем модули SSL

- команда:** a2enmod ssl
- команда:** a2ensite default-ssl
- команда:** a2enmod rewrite

перезапустим Apache

- команда:** service apache2 restart

установим права доступа для папки runtime и файлов XCR, .htaccess

- команда:** chmod 766 .htaccess
(файл находится в папке /var/www)
- команда:** chmod 777 xcr
(файл находится в папке /var/www/upload)
- команда:** chmod 777 runtime
(путь до папки /var/www/protected/)

Создание и установка самоподписанного сертификата

Создание запроса на подпись сертификата (CSR)

- команда:** openssl genrsa -des3 -out server.key 2048

вывод должен быть примерно таким:

```
Generating RSA private key, 2048 bit long modulus
.....++++++
.....++++++
```

e is 65537 (0x10001)
Enter pass phrase for server.key:

Теперь вы можете ввести вашу кодовую фразу.
Повторите ввод для проверки. В случае корректного ввода ключ сервера будет создан и записан в файл server.key.

Теперь создадим небезопасный ключ, без кодовой фразы и перетасуем имена ключей:

команда: openssl rsa -in server.key -out server.key.insecure

команда: mv server.key server.key.secure

команда: mv server.key.insecure server.key

Небезопасный ключ теперь называется server.key и вы можете использовать его для создания CSR без кодовой фразы.

Для создания CSR выполните следующую команду в терминале:

команда: openssl req -new -key server.key -out server.csr

У вас будет запрошена кодовая фраза (при использовании ключа с паролем - прим. пер.). Если введена корректная фраза, у вас запросят название компании, имя сайта, email и пр. Как только вы введете все эти подробности, будет создан запрос CSR и сохранен в файл server.csr.

Для создания самоподписанного сертификата, запустите следующую команду в терминале:

команда: openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

Эта команда попросит вас ввести кодовую фразу. Как только вы введете корректную фразу, ваш сертификат будет создан и сохранен в файл server.crt.

Установка сертификата

Вы можете установить файлы ключа server.key и сертификата server.crt (или файл сертификата, выданный вашим CA), запуском следующих команд в терминале:

команда: sudo cp server.crt /etc/ssl/certs

команда: sudo cp server.key /etc/ssl/private

Для того что бы наш сертификат и файл ключа заработали укажем к ним путь в файле default-ssl

команда: nano /etc/apache2/sites-available/default-ssl

Найдем строки SSLCertificateFile и SSLCertificateKeyFile и поменяем путь и название файлов на нужные нам.

```
<-----># A self-signed (snakeoil) certificate can be created by installing
<-----># the ssl-cert package. See
<-----># /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
<-----># If both key and certificate are stored in the same file, only the
<-----># SSLCertificateFile directive is needed.
<----->SSLCertificateFile /etc/ssl/certs/server.crt
<----->SSLCertificateKeyFile /etc/ssl/private/server.key
```

перезапустим Apache

команда: service apache2 restart

Если вы все сделали правильно то при попытке подключения к G-server, браузер предупредит вас о необходимости принятия сертификата (процедура принятия сертификата описана в руководстве

пользователя G-WEB). После внесения сертификата в число доверенных вы сможете начать работать в среде G.