Установка 1С:Предприятие 8.3 на Linux сервер

В этой статье мы постарались собрать наиболее полную информацию по установке сервера приложений 1С:Предприятие в среде Linux с использованием СУБД PostgreSQL и публикацией информационной базы и веб-сервисов на веб-сервере Apache. Рассматривается наиболее полная установка сервера в домен Microsoft Windows. Пользователи работают на рабочих станциях в среде Windows, авторизация в системе 1С:Предприятие производится через Active Directory.

План работ

1. Установить операционную систему
2. Установить СУБД PostgreSQL
3. Установить сервер 1С:Предприятие
4. Зарегистрировать сервер в домене
5. Опубликовать информационную базу и веб-сервисы
6. Настроить резервное копирование

Перед началом работ трезво оцените свои силы. Для подготовленного специалиста все перечисленные ниже операции займут не более половины рабочего дня, при чем большая часть этого времени уйдет на ожидание завершения операций копирования, установки, загрузки и т.п.
Если установка производится поверх уже работающей системы, сделайте резервную копию, например, сохраните образ дисков. Если что-то пойдет не так, Вы сможете быстро восстановить прежнюю работоспособность системы

Установка операционной системы

Поддерживаются наиболее популярные deb (Ububtu, Debian и пр.) и rmp (CentOS, Fedora и пр.) дистрибутивы. Каких-либо явных рекомендаций по выбору того или иного дистрибутива Linux не существует. Рекомендуем выбрать наиболее знакомый Вам дистрибутив, в нашем случае это Ubuntu актуальной на момент написания статьи версии 16.04 LTS, все дальнейшее описание будет основываться именно на этом выборе. Если Вы выбрали другой deb-based дистрибутив, то все описанное подойдет и для него, rmp-based дистрибутивы используют другие команды для управления пакетами, учитывайте это при адаптации статьи для своих целей. Скачайте дистрибутив Ubuntu Server. Применительно к Ubuntu желательно использовать LTS дистрибутивы, они выпускаются раз в два года и поддерживаются в течение 5 лет. Дистрибутив распространяется в виде ISO образа загрузочного диска, можно записать образ на DVD диск, но гораздо удобней подготовить загрузочную флешку.

В серверном дистрибутиве Ubuntu отсутствует графическая оболочка. Установка и последующая настройка будет производится в командной консоли. Не надо устанавливать графическую оболочку, хоть это и просто, но совершенно бессмысленно, она ничем не поможет в настройке. Для удобства администрирования потом можно установить, например, Webmin

Сам процесс установки достаточно прост. Инсталлятор будет задавать Вам различные вопросы, Вам нужно на них отвечать, тем самым выбирая как устанавливать операционную систему. Вначале Вам предложат выбрать язык и страну, часовой пояс, выбирайте в соответствии со своими предпочтениями.

Инсталлятор предложит Вам разметить диски под операционную систему. Это самый важный вопрос в установке. Внимательно ознакомьтесь с нижеследующими рекомендациями.

Если у Вас один физический диск (или RAID-массив), разделите его на 3 логических диска. В начале диска выделите 60-120 Гб собственно под операционную систему (диск 1), оставшуюся часть поделите примерно поровну под домашние папки (диск 2) и файлы СУБД (диск 3). Монтируйте следующим образом:

```
/ - диск 1
/home - диск 2
/var - диск 3
```

Если же дисковая система Вашего сервера более разнообразна, то самый медленный ее элемент рекомендуем монтировать как `/home`, а самый быстрый как `/var`.

Далее инсталлятор предложит Вам выбрать наборы пакетов для установки. Выберите для установки следующие наборы:

Как можно заметить, инсталлятор услужливо предлагает вариант установки PostgreSQL database. Не соблазняйтесь, нам нужен другой PostgreSQL, с патчем для 1С.

Инсталлятор закончит свою работу, перезапустит систему и пред Вами предстанет приглашение в консоль примерно такого вида:

```
Ubuntu 16.04.1 LTS MyServer tty1

MyServer login:
```

Заходите, и чувствуйте себя как дома.

Первым делом нужно настроить сетевые подключения. Мы имеем дело с сервером и подразумеваем, что сетевых адаптеров в нем может быть больше одного. Давайте посмотрим, что у нас есть, вводим в консоль команду

```
$ sudo lshw -C network
```

Здесь использована команда `sudo` - специальная конструкция deb-based дистрибутивов Linux для выполнения команд с правами root. Когда Вы делаете это первый раз система попросит Вас ввести пароль и на какое-то время запомнит его.

В ответ Вы получите нечто похожее на

```
  *-network
       description: Ethernet interface
       product: L2 100 Mbit Ethernet Adapter
```

```
       vendor: Attansic Technology Corp.
       physical id: 3
       bus info: pci@0000:03:00.0
       logical name: enp0s3
       version: a0
       serial: 00:00:00:00:00:00
       size: 100MB/s
       capacity: 100MB/s
```

Нас интересует вот этот фрагмент: `logical name: enp0s3`. Если сетевых адаптеров больше одного, то и таких блоков в выдаче будет по количеству сетевых адаптеров. Запоминаем логические имена. Сетевые подключения настраиваются в конфигурационном файле, откроем его для редактирования

```
$ sudo nano /etc/network/interfaces
```

Допустим, у нас есть два сетевых адаптера. `enp0s3`, его мы настроим на статический IP адрес и будем использовать для подключения к домену. Еще есть `enp0s8`, он получит адрес от DHCP сервера в локальной сети, просто так, для примера. Файл настроек выглядит так:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
iface enp0s3 inet static
address 192.168.0.8            # здесь указываем статический адрес
netmask 255.255.255.0          # маска подсети
gateway 192.168.0.1            # шлюз по умолчанию
dns-nameservers 192.168.0.101  # DNS сервера, можно указать несколько через пробел
auto enp0s3

# The second network interface
iface enp0s8 inet dhcp
auto enp0s8
```

Сохраните файл нажав Ctrl-O и закройте редактор Ctrl-X. После редактирования перезапустим сеть:

```
$ sudo /etc/init.d/networking restart
```

и проверим что у нас получилось

```
$ ifconfig
```

В выдаче этой команды внимательно смотрим на значения `inet addr` - в нашем примере там должен быть статический адрес, заданный в конфигурационном файле для первого адаптера и динамический адрес выданный DHCP сервером для второго.

После того, как настроены сетевые подключения можно покинуть серверную. Дальнейшую

настройку удобнее производить с рабочей станции, подключившись по протоколу SSH.

Для дистанционного управления сервером с рабочей станции Windows мы будем использовать PuTTy. Скачайте, установите и подключайтесь. Адрес сервера в нашем примере указывается так `user@192.168.0.8`, где `user` - имя пользователя, порт по умолчанию 22.

Далее нам потребуется копировать файлы на наш сервер. Что бы делать это максимально комфортно, настроим пакет Samba, позволяющий подружить Linux и Windows по локальной сети.

Задача минимум сейчас - расшарить папку на сервере, куда мы сможем копировать нужные файлы.

```
$ sudo nano /etc/samba/smb.conf
```

В конфигурационном файле находим и раскомментируем блок

```
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mask = 0700
   directory mask = 0700
```

Добавим нашего пользователя в Samba

```
$ sudo smbpasswd -a user
```

Указываем ему пароль и включаем

```
$ sudo smbpasswd -e user
```

И перезапускаем Samba

```
$ sudo service smbd restart
```

Попробуем зайти из проводника Windows. В нашем примере это будет выглядеть так `\\192.168.0.8\user`, вводим имя, пароль и попадаем в домашнюю папку.

Обратите внимание, нужно вводить сетевой адрес полностью, указывая не только адрес сервера, но и имя каталога пользователя

На всякий случай обновите русскую локаль

```
$ sudo update-locale LANG=ru_RU.UTF-8
```

Установка СУБД PostgreSQL

1С:Предприятие работает со специально пропатченной версией PostgreSQL. Версия из репозитория Ubuntu не подойдет.

Существует несколько источников, где можно взять дистрибутив PostgreSQL совместимый с 1С:Предприятие. Можно, например, как рекомендует сама фирма «1С» скачать его с портала ИТС и установить из deb или rpm пакетов, такой подход вполне привычен для пользователей Windows, скачивать и устанавливать. В Linux другой подход. Например, для поклонников Gentoo Linux привычно было бы скачать исходники PostgreSQL, пропатчить их и скомпилировать. А философия Ubuntu базируется на репозиториях, откуда пользователи получают нужные программы, устанавливая их менеджерами пакетов. К сожалению фирма «1С» не озаботилась созданием такого репозитория для распространения специальной версии PosgreSQL, зато это сделала компания Postgres Professional.

К сожалению, компания PostgresPro ограничила доступ к своим репозиториям, на официальном сайте указано, что они доступны в рамках сервисного контракта. Скачайте и установите DEB пакеты с портала 1С:ИТС.

Сначала всё-таки стоит добавить официальный репозиторий PostgreSQL в Ubuntu. Это будет полезно для разрешения зависимостей пакетов.

Качаем файл `postgresql_10.5_24.1C_amd64_deb.tar.bz2` где `5_24` - актуальный на момент написания статьи номер сборки, к моменту, когда Вы это прочтете он может стать другим, берите самую актуальную сборку.

Распакуем, перейдем в папку с пакетами, установим и зафиксируем версии, что бы Ubuntu при автоматическом обновлении не заменила эти пакеты на стандартные из репозитория:

```
$ tar xvjf postgresql_10.5_24.1C_amd64_deb.tar.bz2
$ cd postgresql-10.5-24.1C_amd64_deb
$ sudo dpkg -i libpq5_10.5-24.1C_amd64.deb
$ sudo apt install postgresql-common
$ sudo dpkg -i postgresql-client-10_10.5-24.1C_amd64.deb
$ sudo dpkg -i postgresql-10_10.5-24.1C_amd64.deb
$ sudo apt-mark hold libpq5
$ sudo apt-mark hold postgresql-client-10
$ sudo apt-mark hold postgresql-10
```

Не забываем заменять `5-24` на номер скачанной сборки.

Настроим права на подключение к СУБД из консоли.

```
$ sudo nano /etc/postgresql/10/main/pg_hba.conf
```

В открывшемся файле найдем строку

```
local all postgres peer
```

и приведем ее к виду

```
local all postgres trust
```

Только что мы настроили PostgreSQL доверять всем локальным подключениям. Это не совсем безопасно, мы исправим это чуть позже, после того как установим пароль пользователю postgres.

Если не хотите, что бы сервер СУБД был виден кому-либо, кроме сервера 1С:Предприятие, который мы чуть позже установим, поправьте настройки безопасности

```
$ sudo nano /etc/postgresql/10/main/postgresql.conf
```

В открывшемся файле находим строку

```
listen_addresses = '*'
```

и приводим ее к виду

```
listen_addresses = 'localhost'
```

Такая настройка ограничит видимость PostgreSQL только этим сервером. Если сервер 1С:Предприятие будет установлен на другой машине, вместо localhost укажите ее адрес.

После такой установки PostgreSQL его служба сама не стартует и не прописывается в автозагрузку. Сделаем это

```
$ sudo service postgresql start
$ sudo update-rc.d postgresql enable
```

И еще немного настроек безопасности. Установим пароль `password` на пользователя `postgres`, именем которого будут производится все операции с базами данных (вместо `password` укажите свой пароль).

```
$ psql -U postgres -d template1 -c "ALTER USER postgres PASSWORD 'password'"
```

Поправим методы аутентификации пользователей

```
$ sudo nano /etc/postgresql/10/main/pg_hba.conf
```

В открывшемся файле устанавливаем все методы в `md5`, например, так

```
local all postgres md5
```

Перезапускаем службу PostgreSQL

```
$ sudo service postgresql restart
```

Готово!

Установка сервера 1С:Предприятие

При выборе архитектуры дистрибутива i386 или x86-64 следует учитывать не только архитектуру сервера и установленной на него операционной системы, но и имеющуюся у Вас лицензию на сервер 1С:Предприятия. Здесь описывается процесс установки 64-разрядного сервера в x86-64 архитектуру. Установка 32-разрядного сервера в i386 архитектуру аналогична с точностью до наименования некоторых файлов и каталогов. Установка 32-

разрядного сервера в архитекруту x86-64 потребует решения проблем зависимостей пакетов, и этот подвиг достоин отдельной статьи

Предварительно установим пакеты, необходимые для работы сервера 1С:Предприятие

```
$ sudo apt install imagemagick unixodbc libgsf-bin ttf-mscorefonts-installer
```

В процессе установки пакета `ttf-mscorefonts-installer` потребуется принять условия пользовательского соглашения EULA.

Любым законным способом получаем дистрибутив платформы 1С:Предприятия для deb-based Linux, например, скачиваем его с портала ИТС. Сервер 1С:Предприятие состоит из 3-х пакетов:

- `1c-enterprise83-common` - общие компоненты
- `1c-enterprise83-server` - собственно сам сервер, пакет зависим от общих компонент
- `1c-enterprise83-ws` - веб-расширения сервера, пакет зависим от самого сервера

Если в 1С:Предприятии кроме русского и английского Вам потребуются другие языки, берите пакеты в наименовании которых присутствует NLS

Скопируйте только эти 3 файла по сети в домашний каталог пользователя, в нашем примере `\\192.168.0.8\user` Устанавливать пакеты нужно именно в такой последовательности. Случайно или нет, они по алфавитному порядку выстраиваются именно так, как нужно, а значит их можно установить одной командой

```
$ sudo dpkg -i 1c*.deb
```

Сервер установлен. Запускаем

```
$ sudo service srv1cv83 start
```

Подразумевается, что активация сервера будет проводится программной лицензией. Если по каким-либо причинам у вас появится желание воткнуть в сервер ключ HASP, скачайте и установите пакет haspd от Etersoft
Если конфигурация 1С:Предприятие имеет дополнительную защиту, установите СЛК

К серверу уже можно подключаться через консоль администрирования серверов, активировать лицензии и создавать информационные базы.

Регистрация сервера в домене

Процедура имеет смысл только если используется домен Microsoft Windows и планируется авторизация пользователей 1С:Предприятие по учетной записи Active Directory

Как обычно, начнем с установки необходимых пакетов

```
$ sudo apt install krb5-user
```

Если при установке системы не устанавливали Samba, то самое время сделать это

```
$ sudo apt install samba
```

И еще может понадобиться, а может нет, но лучше установить

```
$ sudo apt install libpam-krb5
```

Настроим DNS

DNS потребуется нам для того, что бы сервер мог обращаться к ресурсам локальной сети по логическим именам.

Сначала разберемся с именем самого сервера

```
$ sudo nano /etc/hostname
```

В этом файле должно быть указано имя компьютера, данное ему при установке операционной системы. Если по каким-то причинам там не оно - исправьте.

Теперь научим его находить самого себя по своему короткому и полному доменному имени.

```
$ sudo nano /etc/hosts
```

Здесь нужно сопоставить имена с IP адресом любого сетевого соединения, например, так

```
127.0.0.1   localhost
127.0.0.1   myserver.domain.ru    myserver
```

Теперь собственно настроим DNS. Допустим, контроллер домена у нас находится на адресе `192.168.0.1`. Если IP адрес задается статически, делаем так

```
$ sudo nano /etc/resolvconf/resolv.conf.d/head
```

и прописываем имена и адреса

```
domain domain.ru
search domain.ru
nameserver 192.168.0.1
```

Если есть еще один сервер службы имен в домене, добавьте в файл еще одну строчку `nameserver` с его адресом.

При получении IP адреса от DHCP по идее все должно настроиться автоматически, но есть нюансы, записи домена могут не определиться. Исправляем

```
$ sudo nano /etc/dhcp/dhclient.conf
```

Добавляем запись домена

```
supersede domain-name "domain.ru";
```

Чтобы добавить еще один сервер службы имен домена находим в этом файле строку `prepend`

domain-name-servers, раскомментируем ее и указываем нужный IP адрес.

Перезапустим службу сети

```
$ sudo /etc/init.d/networking restart
```

Проверим, что у нас получилось. Попробуйте пингануть контроллер домена по короткому и полому доменному имени имени, допустим его зовут server

```
$ ping server
$ ping server.domain.ru
```

Синхронизируем часы

Чуть далее мы будем использовать протокол Keberos, а для его корректной работы важно, что бы системные часы контроллера домена и нашего сервера были синхронизированы.

Забегая еще далее скажем, что для авторизации пользователей 1С:Предприятие через учетную запись Active Directory нужно что бы и часы рабочих станций были синхронизированы с контроллером домена.

Установим часового демона, который будет периодически синхронизировать время с контроллером домена по протоколу NTP

```
$ sudo apt install ntpd
```

настроем его

```
$ sudo nano /etc/ntp.conf
```

укажем источник точного времени в домене - контроллер домена

```
server server.domain.ru
```

и перезапустим

```
$ sudo /etc/init.d/ntp restart
```

Настроим авторизацию через Kerberos

Открываем конфигурационный файл

```
$ sudo nano /etc/krb5.conf
```

И приводим его к такому виду, разумеется заменяя имена и IP адреса нашего примера под свои условия

```
[libdefaults]
```

```
    default_realm = DOMAIN.RU  # заглавными буквами
    kdc_timesync = 1
    ccache_type = 4
    forwardable = true
    proxiable = true
    v4_instance_resolve = false
    v4_name_convert = {
        host = {
            rcmd = host
            ftp = ftp
        }
        plain = {
            something = something-else
        }
    }
    fcc-mit-ticketflags = true

[realms]
    DOMAIN.RU = {
        kdc = server                    # сервер службы имен
               #kdc = dns-server                # второй сервер имен, если есть
        admin_server = server           # контроллер домена
        default_domain = DOMAIN.RU      # имя домена указывайте именно так -
заглавными буквами
    }

[domain_realm]
    .domain.ru = DOMAIN.RU
    domain.ru = DOMAIN.RU
[login]
    krb4_convert = false
    krb4_get_tickets = false
```

В Linux, в отличии от Windows, регистр букв имеет значение. Например, в одной папке совершенно спокойно могут находится два файла file.txt и File.txt, и никто не помешает создать еще пару FILE.txt и file.TXT. Обратите внимание на написание имени домена в конфигурационном файле - это важно!

Пробуем войти в домен как его пользователь `ivanov`

```
$ kinit ivanov@DOMAIN.RU
```

Если все настроено правильно, система запросит пароль пользователя домена и Вы не увидите никаких сообщений об ошибках. Посмотреть полученный билет можно командой

```
$ klist
```

Он выглядит примерно так

```
Valid starting       Expires              Service principal
15.02.2017 23:08:18  16.02.2017 09:08:18  krbtgt/DOMAIN.RU@DOMAIN.RU
```

```
        renew until 16.02.2017 23:08:07
```

И он нам больше не нужен, удалим его

```
$ kdestroy
```

Настроим Samba и включим сервер в домен

Мы уже немного настраивали Samba для того, что бы нам было удобнее копировать дистрибутивы по сети. Эти настройки будут полезны в будущем при обновлении платформы 1С:Предприятие, оставьте их. В этом разделе описаны другие настройки.

Открываем файл конфигурации Samba

```
$ sudo nano /etc/samba/smb.conf
```

И приводим его примерно к такому виду

```
[global]
   workgroup = DOMAIN         # заглавными буквами
   realm = DOMAIN.RU          # заглавными буквами

   security = ADS             # авторизация через Active Directory
   encrypt passwords = true   # с шифрованием паролей

   dns proxy = no
   socket options = TCP_NODELAY

   domain master = no         # этот блок настроек указывает, что мы не главные в
домене
   local master = no
   preferred master = no
   os level = 0
   domain logons = no

   load printers = no         # принтерами управлять не будем
   show add printer wizard = no
   printcap name = /dev/null
   disable spoolss = yes

# добавьте сюда блоки и настройки, которые мы делали в разделе установки операционной
системы
```

Проверим правильность настроек

```
$ testparm
```

Если ошибок нет, и правильно установлена роль

```
Server role: ROLE_DOMAIN_MEMBER
```

то мы готовы включиться в домен, разумеется, если знаем пароль администратора домена `domainadministrator`

```
$ sudo net ads join -U domainadministrator -D DOMAIN
```

Если Вы видите нечто похожее на

```
Using short domain name -- DOMAIN
Joined 'MyServer' to dns domain 'domain.ru'
```

то поздравляем, у Вас все получилось, Вы в домене!

Авторизация пользователей 1С:Предприятие на Linux сервере через Active Directory

В клиент-серверном варианте работы 1С:Предприятие аутентификация пользователей производится на сервере. Если сервер работает под управлением Windows, то он без проблем получает нужные права для того, что бы проверить учетную запись Active Directory.

В Linux дела обстоят иначе. Сервер 1С:Предприятие у нас работает под учетной записью `usr1cv8`, которая имеет какие-либо права только в пределах нашего linux-сервера. В домене у этого linux-пользователя нет никаких прав, соответственно сервер 1С:Предприятие не может получить ничего от Active Directory.

Что бы исправить эту проблему серверу 1С:Предприятие нужно выдать специальный «билет» и сопоставить его с пользователем Active Directory по протоколу Keberos.

Создайте такого пользователя в Active Directory, назовите его, например, `linux1cv8`. Права у него могут быть совершенно любые, главное, что бы в настройках его учетной записи была отключена опция `Use DES encryption types with this account`.

Выполняемые на сервере 1С:Предприятие операции в домене будут иметь привилегии именно этого пользователя. Дайте этому пользователю права на чтение и запись сетевых ресурсов, с которыми Вы планируете работать через сервер 1С:Предприятие. Это могут быть, например, каталоги для обмена данными с другими информационными системами.

Теперь нам потребуется командная строка Windows рабочей станции, включенной в домен и утилита `ktpass`, найти ее можно в пакете «Windows Support Tools» сервера или «Средства удаленного администрирования» полнофункциональных версий настольных Windows. Эта утилита генерирует специальный ключ, который не-windows системы могут использовать для авторизации в домене по протоколу Keberos.

Сделаем это

```
ktpass -princ linux1cv8/myserver.domain.ru@DOMAIN.RU -mapuser usr1cv8 -pass
<пароль AD пользователя> -out usr1cv8.keytab
```

Вот тут следует уделить особое внимание именам пользователей. Этой командой нам надо, что бы в файл usr1cv8.keytab был записан ключ, с помощью которого linux-пользователь usr1cv8 через протокол Keberos будет представлен в домене как пользователь AD

linux1cv8@domain.ru вошедший в домен с хоста myserver.domain.ru

Полученный файл переместите по сети в домашнюю папку на linux-сервер \\myserver\user. А на сервере его нужно переместить в каталог, где установлено 1С:Предприятие, поменять ему владельца и назначить безопасные права

```
$ sudo mv /home/user/usr1cv8.keytab /opt/1C/v8.3/x86_64
$ sudo chown usr1cv8:grp1cv8 /opt/1C/v8.3/x86_64/usr1cv8.keytab
$ sudo chmod 600 /opt/1C/v8.3/x86_64/usr1cv8.keytab
```

Проверим, все ли сделано правильно

```
$ sudo -u usr1cv8  klist -e -k -t /opt/1C/v8.3/x86_64/usr1cv8.keytab
```

Если Вы видите нечто похожее на

```
Keytab name: FILE:usr1cv8.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
---
   4 01.01.1970 03:00:00 usr1cv8/myserver.domain.ru@DOMAIN.RU (arcfour-hmac)
```

то у Вас все получилось, поздравляем!

Файл .keytab нужно сделать только для одного пользователя AD, в нашем примере это linux1cv8@domain.ru. Не надо делать такие файлы для всех учетных записей AD пользователей 1С:Предприятие
В 1С:Предприятии для авторизации через Active Directory учетные записи нужно указывать маленькими буквами и с полным наименованием домена, вот так: \\domain.ru\ivanov

Публикация на веб-сервере Apache

Использовать один и тот же сервер в качестве сервера приложений (а также СУБД) и web-сервера безопасно только в закрытой сети предприятия Интранет. Для публикации в Интернет правильно будет использовать отдельный web-сервер и разместить его в DMZ
1С:Предприятие релизов до 8.3.9 поддерживало публикацию информационной базы на веб-сервере Apache версии не выше 2.2. Для использования Apache 2.4, включенного в репозитории Ubuntu 16.04 LTS обновитесь до актуального релиза технологической платформы 1С:Предприятие.

Как всегда, начинаем с установки нужных нам пакетов. Если Вы разворачиваете отдельный сервер для web-публикации, при установке на него операционной системы достаточно указать, что этот сервер будет работать как LAMP (Linux Apache MySQL Php).

Если сервер уже установлен без Apache

```
$ sudo apt install apache2
```

Уточним, что мы установили из репозитория

```
$ apache2 -v
```

Запоминаем версию

```
Server version: Apache/2.4.18 (Ubuntu) Server built: 2016-07-14T12:32:26
```

2.4 - этот номер версии нам нужно будет указать при публикации, соответственно в параметре публикации указываем `-apache24`. Все остальные параметры указаны для установки Apache 2.4 по-умолчанию.

```
$ sudo /opt/1C/v8.3/x86_64/webinst -apache24 -wsdir base1c -dir
/var/www/html/base1c -connstr "Srvr=myserver;Ref=base1c;" -confPath
/etc/apache2/apache2.conf
$ sudo service apache2 restart
```

Проверяем, заходим через браузер `http://myserver/base1c` и видим веб-клиента 1С:Предприятие.

Опубликуем веб-сервисы. Утилита webinst этого не сделала, а значит придется сделать это самим. Для этого нужно отредактировать файл публикации информационной базы, созданный этой утилитой

```
$ sudo nano /var/www/html/base1c/default.vrd
```

В этот XML файл после блока, описывающего подключение к информационной базе нужно добавить блоки с описанием публикуемых веб-сервисов информационной базы, например, для работы мобильного приложения 1С:Заказы с поддерживаемыми конфигурациями нужно опубликовать веб-сервис `CustomerOrdersExchange`. Подробнее см. информацию в документации к прикладному решению на платформе 1С:Предприятие.

После изменений в конфигурационных файлах не забывайте перезапускать Apache.

Мы настроили публикацию информационной базы по протоколу HTTP. Это вполне годится для Интранет, но ни в коем случае не публикуйте так базу в открытом доступе в сети Интернет. По протоколу HTTP все данные, в т.ч. пароли пользователей передаются в незашифрованном виде. Обязательно настройте безопасный протокол HTTPS

Настройка HTTPS

Для совсем правильного и безопасного доступа по протоколу HTTPS нужно получить цифровой сертификат, подписанный удостоверяющим центром (Certificate Authority). Такие сертификаты гарантируют не только шифрование трафика между клиентом и сервером, но и обезопасят ваших пользователей от кражи паролей, особенно, если они будут подключаться из публичных wi-fi сетей.

Такой сертификат можно приобрести практически у любого хостинг-провайдера, многие предлагают первый выпуск базового сертификата сроком действия 3 или 12 месяцев бесплатно. Стоимость продления начинается примерно от $50, в зависимости от опций.

Еще вариант - получить сертификат центра сертификации Let’s Encrypt. Его выпуск и продление совершенно бесплатны. Сертификат выпускается сроком на 3 месяца, зато процесс его выпуска, продления и установки в веб-сервер Apache полностью автоматизированы.

Здесь же мы опишем выпуск т.н. самоподписанного сертификата.

Никто, кроме Вас не сможет проверить правильность самоподписанного сертификата и при подключении к информационной базе у пользователя не будет никакой гарантии, что он подключается именно к Вашему серверу. Трафик будет шифроваться, так что это относительно безопасно

Создадим SSL сертификат

```
$ openssl req -new -x509 -days 365 -keyout myserver.key -out myserver.pem
```

Укажите `PEM pass phrase` - пароль приватного ключа. На остальные вопросы можете отвечать как угодно, кроме `Common Name (eg, YOUR name) []:` - тут следует указать доменное имя сайта. В результате в домашней папке появится два файла `myserver.key` - приватный ключ и `myserver.pem` - публичный сертификат. Что бы Apache при загрузке не спрашивал непонятно у кого пароль приватного ключа, снимем его

```
$ cp myserver.key{,.orig}
$ openssl rsa -in myserver.key.orig -out myserver.key
$ rm myserver.key.orig
```

Переместим их в безопасное место и настроим права доступа только для `root`

```
$ sudo mv myserver.pem /etc/ssl/certs/
$ sudo mv myserver.key /etc/ssl/private/
$ sudo chmod 0600 /etc/ssl/private/server.key
```

Переходим к настройкам Apache, включаем поддержку SSL

```
$ sudo a2enmod ssl
```

И настраиваем доступ по HTTPS. Открываем файл конфигурации сайта по-умолчанию

```
$ sudo nano /etc/apache2/sites-enabled/000-default
```

и приводим его к такому виду

```
<VirtualHost *:80>
    # Перенаправим все поступающие по HTTP запросы на HTTPS хост
        Redirect / https://mysite.ru/  # mysite.ru - это доменое имя или IP адрес вашего сайта
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost *:443>
        ServerAdmin webmaster@mysite.ru

        ServerName mysite.ru
        DocumentRoot /var/www/html      # Это путь по-умолчанию

        ErrorLog /var/log/apache2/mysite.ru-error.log
```

```
        CustomLog /var/log/apache2/mysite.ru-access.log combined

        # SSL
        SSLEngine on
        SSLProtocol all -SSLv2
        SSLCertificateFile    /etc/ssl/certs/myserver.pem
        SSLCertificateKeyFile /etc/ssl/private/myserver.key
</VirtualHost>
</IfModule>
```

Перезапускаем Apache

```
sudo service apache2 restart
```

Готово!

Мы полностью запретили использование незащищенного протокола HTTP. Все запросы, поступающие на него будут перенаправляться на HTTPS
Это еще не все, Вы остаетесь уязвимы, например, для атаки посредника, подробнее см. в отдельной статье.

Резервное копирование

В клиент-серверном варианте работы 1С:Предприятие резервное копирование производится средствами СУБД. Т.к. СУБД PostgreSQL у нас теперь работает на Linux-сервере резервное копирование будет выполняться скриптами на этом сервере

Резервное копирование в PosegreSQL выполняется утилитой `pg_dump`. Подробная справка по синтаксису выводится командой

```
$ pg_dump --help
```

Например, что бы сделать резервную копию базы `base1c` нужно выполнить такую команду

```
$ pg_dump -f base1c.sql.tar -F t -d base1c -h localhost -U postgres
```

Обратите внимание, запуск утилит резервного копирования и восстановления производится не от суперпользователя, без конструкции sudo. Файл бэкапа будет сформирован в текущем каталоге и его владельцем будет пользователь

Для восстановления базы из архива, созданного командой `pg_dump` используется команда `pg_restore`. Подробная справка по синтаксису

```
$ pg_restore --help
```

Пример восстановления базы `base1c` из бэкапа `base1c.sql.tar`

```
$ pg_restore -d base1c -F t -U postgres base1c.sql.tar
```

Не пытайтесь восстановить бэкап в базу, подключенную к серверу 1С:Предприятие. Восстанавливать нужно в новую базу, после восстановления в консоли управления сервером

1С:Предприятие переподключить для восстанавливаемой информационной базы 1С базу данных PostgreSQL

Бэкап сохранился в домашней папке пользователя, это лучше, чем ничего, но хранить резервные копии баз данных на самом сервере баз данных слишком самоуверенно. Когда-то давно резервные копии делали на ленты стримера и увозили их куда подальше в надежное место. Сейчас для резервного копирования используются сетевые или облачные NAS хранилища, расположенные в таком надежном месте. В нашем примере копирование будет выполняться на сетевое NAS хранилище, включенное в домен под сетевым именем nas.

Обеспечим к нему доступ нашего пользователя. В Active Directory создадим пользователя backupuser, дадим ему права на запись в папке, где будем складывать резервные копии.

Прочим пользователям полностью запретите доступ к этой папке или оставьте права только на чтение. Так резервные копии будут целее.

Примонтируем сетевую папку, сначала создадим точку монтирования

```
$ sudo mkdir /mnt/nas
```

затем настроим автоматическое монтирование при старте системы, для чего сначала создадим файл-мандат на подключение к сетевому хранилищу

```
$ sudo nano /root/.smbcredintals
```

Прописываем в файл параметры подключения

```
username=backupuser
password=backupuserpassword
domain=domain.ru
```

и скроем его от посторонних глаз

```
$ sudo chmode 0600 /root/.smbcredintals
```

Пропишем подключение сетевой папки при старте системы

```
$ sudo nano /etc/fstab
```

в конец файла добавляем

```
//nas/backups    /mnt/nas        cifs
credentials=/root/.smbcredentials,dir_mode=0777,file_mode=0777 0 0
```

и даем команду примонтировать все, что мы прописали

```
$ sudo mount -a
```

В результате в точке монтировали /mnt/nas должна появиться папка сетевого хранилища \\nas\backups, куда можно складывать резервные копии.

Для выполнения автоматического резервного копирования нужно создать скрипт, который будет запускаться по расписанию демоном `cron`. К счастью, писать этот скрипи самому не нужно, в сообществе PostgreSQL уже написано много таких скриптов, рекомендую использовать канонический скрипт.

Альтернативный вариант - установить Webmin - веб интерфейс для администрирования Linux-сервера, и уже с его помощью настроить резервное копирование в службе PostgreSQL

Используйте скрипт `pg_backup_rotated.sh`, он сохраняет резервные копии за несколько дней, автоматически удаляя старые копии.

Создайте в домашней папке пользователя файл `pg_backup_rotated.sh` и разместите в нем скрипт по указанной выше ссылке.

Не делайте это в Windows редакторах. Из-за различия в служебных символах, обозначающих конец строки в Windows и Linux интерпретатор скриптов bash будет выдавать ошибки.

Установите для этого файла признак исполняемости

```
$ chmod +x pg_backup_rotated.sh
```

В том же каталоге создайте файл настроек `pg_backup.config` и заполните его как по указанной выше ссылке. Все настройки можно оставить по умолчанию, кроме

```
BACKUP_DIR=/mnt/nas/
```

Проверьте работоспособность, выполните команду

```
$ ./pg_backup_rotated.sh
```

Если не увидели никаких ошибок и процесс копирования пошел, то все сделано правильно, можно поставить задачу в расписание. Редактируем задания для демона `cron`

```
$ crontab -e
```

добавляем строчку с заданием

```
0 19 * * 1-5 /home/user/pg_backup_rotated.sh
```

Резервное копирование будет выполняться в 19:00 с понедельника по пятницу.

Поздравляем! Только что Вы сэкономили много тысяч рублей на лицензиях Microsoft. Если остались вопросы или нужна помощь - обращайтесь, будем рады помочь.

- Twitter
- Facebook
- Google+
- LinkedIn
- Pinterest
- Tumblr
- Reddit
- Taringa
- StumbleUpon
- Telegram
- Hacker News
- Xing
- Vk
- Email

From:
https://wiki.lineris.ru/ - **ЛИНЕРИС**

Permanent link:
https://wiki.lineris.ru/linux_server?rev=1564425827

Last update: **2019/07/29 18:43**