
TRAVELSHOP

РАЗВЕРТЫВАНИЕ
СИСТЕМЫ

В данном документе описаны основные процедуры по развертыванию информационного комплекса для работы TRAVELSHOP.

ОГЛАВЛЕНИЕ

| | | |
|-----|---|----|
| 1 | Платформа..... | 4 |
| 1.1 | ОС Linux..... | 4 |
| 1.2 | Подготовка и запуск MySQL | 5 |
| 1.3 | Подготовка MySQL для 1С-Битрикс. | 6 |
| 1.4 | Apache..... | 8 |
| 1.5 | PHP | 9 |
| 2 | 1С-Битрикс | 11 |
| 3 | TravelShop | 11 |
| 3.1 | Установка | 11 |
| 3.2 | Обновление | 12 |
| 3.3 | Настройка..... | 12 |
| 4 | Обслуживание системы..... | 12 |
| 4.1 | Резервирование базы данных..... | 12 |
| 4.2 | Резервирование каталога веб-сайта | 13 |

1 ПЛАТФОРМА

На сервере должна быть установлена ОС Linux. Необходимо установить следующее ПО:

1. MySQL Community Edition 5.0 или выше.
2. PHP 5.1.6 или выше.
3. Apache 2.x.

В большинстве современных серверных дистрибутивах установка указанного ПО осуществляется автоматически на этапе конфигурирования системы. В противном случае необходимо провести установку вручную из пакетов.

1.1 ОС Linux

При установке выбрать: язык – English, местоположение – Russia, Moscow.

В меню *Installation Setting/Partition Setup/Expert Partitioner* создаем следующие разделы:

| Устройство | Номер раздела | Тип раздела | Размер | Точка монтирования |
|-------------------|---------------|-----------------|-----------|--------------------|
| sda sdb sdc | 1 | 0xFD Linux RAID | 256 Mb | - |
| sda sdb sdc | 2 | 0xFD Linux RAID | 16 Gb | - |
| sda sdb sdc | 3 | 0xFD Linux RAID | остаток | - |
| sdd | 1 | Ext3 | весь диск | /back |

В меню *RAID/Create RAID/RAID1* монтируем следующие разделы:

| Раздел | Тип раздела | Точка монтирования |
|-------------|-------------|--------------------|
| sda1 и sdb1 | Ext3 | /boot |
| sda2 и sdb2 | Ext3 | / |

| | | |
|-------------|------|-------|
| sda3 и sdb3 | Ext3 | /home |
|-------------|------|-------|

На этапе конфигурирования системы выбираем следующие группы пакетов:

- Server Base System
- Web and LAMP Server

Таким образом, в `/boot` установлен загрузчик, а `/home` будет использован под хранилище данных.

Для обеспечения сохранности данных в аварийных ситуациях (отключение питания) необходимо выключить кэши в дисках a, b, c и включить в диске d. Для этого необходимо:

- Скрипт `boot.wcache` надо положить в `/etc/init.d`.
- Сделать его исполняемым: `chmod +x /etc/init.d/boot.wcache`.
- После этого запустить `yast` и там, в *System/Runlevel editor*'е включить этот «сервис» (он там должен уже появиться к тому моменту).

1.2 Подготовка и запуск MySQL

При помощи `yast` (Software/Software management) нужно установить пакеты, необходимые для работы MySQL:

1. `mysql`
2. `mysql-Max`
3. `mysql-client`
4. `mysql-shared`
5. `mysql-devel`

Подготовить конфигурацию `mysql`-сервера:

1. создать каталог `/home/mysql`:

```
# mkdir /home/mysql
# chown mysql:mysql /home/mysql
# chmod 700 /home/mysql
```

2. выполнить команды:

```
# mv /var/lib/mysql/* /home/mysql
# rm -rf /var/lib/mysql
# ln -s /home/mysql /var/lib/mysql
```

3. В файле `/etc/my.cnf` установить параметры `mysql` и `InnoDB` (значения указаны для сервера с оперативной памятью 2ГБ, все указанные параметры должны быть раскомментированы!):

```
[mysqld]
log-bin = mysql-bin
table_cache = 512
query_cache_size = 64M
query_cache_type = 1
tmp_table_size = 32M
sync_binlog = 1
innodb_data_home_dir = /var/lib/mysql/
innodb_data_file_path = ibdata1:100M:autoextend
```

```
innodb_log_group_home_dir = /var/lib/mysql/  
innodb_log_arch_dir = /var/lib/mysql/  
innodb_buffer_pool_size = 780M  
innodb_additional_mem_pool_size = 20M  
innodb_log_file_size = 100M  
innodb_log_buffer_size = 16M  
innodb_flush_log_at_trx_commit = 1  
innodb_lock_wait_timeout = 50  
innodb_file_io_threads = 8  
innodb_file_per_table = 1
```

4. запустить скрипт `mysql_install_db` – он создаст системные таблицы. Кроме прочего, этот скрипт напечатает две команды (они начинаются с `/usr/bin/mysqladmin`), которые необходимо выполнить ПОСЛЕ него для установки нового пароля пользователя `root` в `mysql`-сервере. Записать эти команды на бумажку.

5. войти в `yast` (`System / System services (Runlevel)`) и включить там сервис `mysql`. При этом должен запускаться `mysql`-сервер. Просмотреть протокол `/home/mysql/mysqld.log` на предмет ошибок. В случае успешного запуска InnoDB в каталоге `/home/mysql` должны быть созданы такие файлы:

```
-rw-rw---- 1 mysql mysql 104857600 May 28 18:09 ib_logfile0  
-rw-rw---- 1 mysql mysql 104857600 May 28 18:09 ib_logfile1  
-rw-rw---- 1 mysql mysql 10485760 May 28 18:09 ibdata1  
-rw-rw---- 1 mysql mysql 98 May 28 18:17 mysql-bin.000001  
-rw-rw---- 1 mysql mysql 19 May 28 18:17 mysql-bin.index
```

(размеры файлов `mysql-bin*` могут отличаться от указанных)

6. выполнить команды `mysqladmin` из предыдущего пункта с указанием нового пароля пользователя `root` в `mysql`-сервере.
7. выполнить команду `mysql -p`, затем в ответ на запрос ввести новый пароль. В случае успеха должно быть выдано приглашение:
`mysql>`

Для ручной остановки `mysql`-сервера нужно использовать команду:
`/etc/init.d/mysql stop`

Для ручного запуска `mysql`-сервера нужно использовать команду:
`/etc/init.d/mysql start`

Для ручного перезапуска `mysql`-сервера нужно использовать команду:
`/etc/init.d/mysql restart`

Протокол `mysql`-сервера пишется в файл:
`/home/mysql/mysqld.log`

1.3 Подготовка MySQL для 1С-Битрикс.

После установки сервера нужно внести пользователей, которые имеют право работать с сервером баз данных. Если пользователь будет только один, то его можно задать при установке «1С-Битрикс».

В противном случае введите следующий запрос:

```
GRANT ALL PRIVILEGES ON *.* TO admin@localhost IDENTIFIED BY 'password' WITH GRANT OPTION;
```

Введенный вами запрос создаст пользователя *admin*, который будет иметь право выполнять любые операции со всеми базами данных. Данный пользователь будет иметь право подключаться к серверу с компьютера *localhost*, используя пароль *password*.

Маска **.** определяет, к каким базам данных и таблицам имеет право подключаться тот или иной пользователь. Первая звездочка определяет базу, а вторая – таблицу. Если вам нужно, чтобы пользователь *admin* имел право подключаться с любого хоста, используйте знак процента вместо имени хоста. В этом случае запрос будет выглядеть так:

```
GRANT ALL PRIVILEGES ON *.* TO admin@% IDENTIFIED BY 'password' WITH GRANT OPTION;
```

Вместо всех полномочий вы можете определить, какие действия может выполнять с базой тот или иной пользователь. Если вы являетесь хостинг-провайдером и предоставляете доступ пользователю к его базе данных, то вы можете использовать следующий запрос:

```
GRANT CREATE, DROP, SELECT, INSERT, UPDATE, DELETE, INDEX ON user.* TO user@% IDENTIFIED BY 'user_password';
```

Перед выполнением данного запроса необходимо создать базу данных *user*. Данный запрос позволяет пользователю *user* выполнять все операции с его базой данных.

Полный список полномочий представлен в таблице:

| ПОЛНОМОЧИЯ | ОПИСАНИЕ |
|--------------------------------|---|
| SELECT, INSERT, UPDATE, DELETE | Одноименные операции с данными: пользователь имеет право просматривать, добавлять, модифицировать, удалять данные в таблицах базы данных. |
| INDEX | Пользователь имеет право производить операции с индексами таблиц. |
| REFERENCES | Пользователь имеет право работать со ссылками в базах данных и таблицах. |
| CREATE, DROP | Создание и удаление таблиц и баз данных. |
| GRANT, ALTER | Операции с полномочиями и изменения структуры таблиц. |
| RELOAD, SHUTDOWN, PROCESS | Пользователь имеет право перезагружать, останавливать сервер и просматривать все |

| | |
|--|-------------------------|
| | процессы (подключения). |
|--|-------------------------|

Если запрос GRANT у вас не работает, то вы можете внести пользователя непосредственно в таблицу *user* базы данных *mysql*. Структура таблицы *user* выглядит следующим образом:

```
Host User Password Select_priv Insert_priv Update_priv Delete_priv Create_priv Drop_priv
Reload_priv Shutdown_priv Process_priv File_priv
```

Поля *Host, User, Password* – это, соответственно, узел, из которого пользователь может получить доступ, имя пользователя, пароль пользователя. Все остальные поля задают полномочия для пользователя. Если выполнение какой-нибудь операции разрешено пользователю, соответствующее поле должно быть равным «Y». В противном случае установите значение «N».

Например, нам нужно создать пользователя *admin*, который должен иметь все полномочия. Это можно сделать с помощью такого запроса SQL:

```
INSERT INTO user (Host, User, Password, Select_priv, Insert_priv, Update_priv,
Delete_priv, Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv)
VALUES ('localhost', 'admin', password ('4td56ls12'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y',
'Y', 'Y', 'Y');
```

При вводе запроса обратите внимание на регистр названий полей. **Сервер MySQL различает прописные и строчные буквы!** С помощью вышеприведенного запроса был создан пользователь *admin*, который имеет право регистрироваться на сервере только из узла *localhost*. Если вам нужно разрешить регистрацию из любой узла сети, используйте знак процента, однако это не совсем корректно с точки зрения безопасности. Пользователь *admin* обладает всеми возможными привилегиями. Пароль пользователя – *4td56ls12*.

Для создания обыкновенного пользователя используйте следующий запрос:

```
INSERT INTO user (Host, User, Password, Select_priv, Insert_priv, Update_priv,
Delete_priv) VALUES ('%', 'user', password ('qwerty'), 'Y', 'Y', 'Y', 'Y');
```

1.4 Apache

Для установки сервера Apache необходимо установить пакеты *apache* и *apache-docs*. В первом из них находится сам сервер, а во втором – документация. Желательно устанавливать самую новую версию. В последних версиях вам нужно установить еще и пакет *apache-common*, содержащий необходимые файлы для запуска сервера. Проще всего установку Apache можно произвести, введя следующую команду:

```
rpm -ih apache*
```

После установки сервер конфигурируется для запуска в режиме *standalone*, то есть он будет постоянно находиться в памяти. Не рекомендуется изменять этот режим. Для запуска и останова сервера Apache вы можете воспользоваться командами:


```
/etc/rc.d/init.d/httpd start
/etc/rc.d/init.d/httpd stop
```

или командами

```
service httpd start
service httpd stop
```

Для автоматического запуска Apache необходимо создать в каталоге `/etc/rc.d/rc3.d/` символическую ссылку на файл `/etc/rc.d/init.d/apache2`:

```
ln -s /etc/rc.d/init.d/apache2 /etc/rc.d/init.d/rc3.d/S15apache2
```

После успешной установки сервера отредактируйте файл `/etc/httpd/conf/httpd.conf`. В нем исправьте директиву – *ServerName*. Укажите реальное доменное имя вашего сайта, например, *your-server.ru*.

Корневым каталогом вашего Web-сервера является каталог `/home/www/public_html`. Когда пользователь введет <http://your-server.ru/> в строке браузера, ваш Web-сервер будет искать файл `index.html` именно в этом каталоге. Если данный каталог вас не устраивает, измените директиву *DocumentRoot* в файле конфигурации `httpd.conf`, после чего перезапустите Apache командой:

```
service httpd restart
```

Каталог `/home/www/cgi-bin` содержит CGI-приложения (из этого каталога возможен запуск программ). Журналы сервера находятся в каталоге `/var/log/httpd`.

Если планируется использование нескольких сайтов, то в случае *Apache 2.x* необходимо описать каждое доменное имя в файлах с расширением `.conf` в каталоге `/etc/apache2/vhosts.d` с помощью директивы *VirtualHost*.

Необходимо разрешить загрузку модуля `mod_rewrite`. В случае *Apache 2.x* это можно сделать, добавив в файле `/etc/sysconfig/apache2` в параметре `APACHE_MODULES` слово *rewrite*.

Внимание!

При настройке виртуальных хостов убедитесь, что существует каталог, заданный вами для хранения логов, например, `/home/www/log`. В противном случае Apache не запустится.

1.5 PHP

Остановите Apache, выполнив команду:

```
/etc/rc.d/init.d/httpd stop
```

или

```
service httpd stop
```

Проверьте наличие библиотеки *gd* – она необходима для работы с графикой в PHP:

```
rpm -qa | grep gd
```

Если библиотека *gd* не установлена, установите ее командой, предварительно указав ту версию библиотеки, которая у вас имеется:

```
rpm -ihv gd-1.8.4-4.i386.rpm
```

Установите пакет *php*, если вы его еще не установили:

```
rpm -Uh php*
```

После этого установите модуль Apache, обеспечивающий поддержку PHP:

```
rpm -Uh mod_php*
```

Данный модуль должен входить в состав дистрибутива и обычно находится на первом инсталляционном диске. Затем в файле *httpd.conf* найдите и раскомментируйте или добавьте следующую строку, если это не произошло автоматически:

```
AddType application/x-httpd-php5 .php
```

После этого файлы с расширением *.php* будут правильно обрабатываться сервером.

Все параметры интерпретатора PHP задаются в файле настроек */etc/php5/apache2/php.ini* (*/etc/php.ini*).

Так как PHP установлен как модуль Apache, настройки интерпретатора можно задавать в файле *httpd.conf* или файлах *.htaccess*. В этом случае перед именем каждого параметра нужно указывать префикс *php_* и указывать значение через пробел, без использования знака равенства.

Рекомендуется увеличить значение параметра *memory_limit* до 128 МБ, включить отображение ошибок и увеличить максимальное время выполнения скрипта до 120 секунд. Это можно сделать в *php.ini*:

```
memory_limit=128M
```

```
display_errors=On
```

```
max_execution_time=120
```

ВНИМАНИЕ! Для загрузки обновлений TAIS TravelShop через панель Битрикса необходимо увеличить ограничение на закачиваемые файлы с помощью следующих параметров:

```
post_max_size = 8M
```

```
upload_max_filesize=8M
```

ВНИМАНИЕ! Для работы модуля бронирования TAIS TravelShop необходимо в PHP подключить следующие модули:

- curl
- mbstring
- sockets
- ctype
- mysql
- soap (если используются платежные системы с интерфейсом SOAP).

2 1С-БИТРИКС

На сервер 1 устанавливается «1С-Битрикс: Управление сайтом» редакции «Бизнес/MySQL» или «Малый бизнес/MySQL». Инструкция по инсталляции продукта доступна на сайте <http://www.1c-bitrix.ru>.

Для получения дистрибутива необходимо приобрести продукт одной из указанной редакции и получить лицензионный ключ. Дистрибутив можно скачать на сайте <http://www.1c-bitrix.ru>, введя для доступа лицензионный ключ.

Откажитесь от установки тестового сайта (отключите «Установить демонстрационные данные» в разделе «Варианты установки»).

Если вы уже завели пользователя для работы с базой данных на этапе 1.2, то укажите его имя и пароль при установке «1С-Битрикс». В противном случае задайте имя и пароль для создания нового пользователя, указав имя и пароль суперпользователя MySQL (*root*).

Задайте имя базы данных, в которой будут храниться все таблицы «1С-Битрикс».

В ходе установки «1С-Битрикс» необходимо выбрать тип таблиц базы данных MySQL – **InnoDB**. Этот тип таблиц используется в системе TravelShop.

3 TRAVELSHOP

Установка TravelShop возможна только после установки Битрикса.

3.1 Установка

- В панели управления сайта (<http://адрес сайта/bitrix/admin/>) зайдите в *Настройки / Настройки продукта / Модули* и установите, если они еще не установлены, модули «Валюты», «Интернет-магазин», «Информационные блоки», «Компрессия», «Торговый каталог», «Управление структурой».
- Зайдите в *Контент / Структура сайта / Файлы и папки* и загрузите архивный файл TravelShop [travelshop.tar.gz](http://www.1c-bitrix.ru) в папку */bitrix/backup/* (создайте папку, если ее нет).
- Зайдите в *Настройки / Инструменты / Резервное копирование* и выполните восстановление системы из архива, для чего нажмите правую кнопку мыши на архивном файле [travelshop.tar.gz](http://www.1c-bitrix.ru) и выберите в открывшемся контекстном меню «**Распаковать**». Следуйте дальнейшим инструкциям системы.

- Снова зайдите в *Настройки / Настройки продукта / Модули* и установите, если он еще не установлен, модуль «TravelShop».
- Зайдите в *Контент / Структура сайта / Файлы и папки* и откройте для редактирования файл `/bitrix/php_interface/init.php` (нажмите правую кнопку мыши на файле и выберите в контекстном меню **«Редактировать как PHP»**). Добавьте строку


```
require_once (
$_SERVER["DOCUMENT_ROOT"]."/bitrix/modules/ibe/classes/ibe/init.php"
);
```

в конец файла (до символов «?»»), если ее там еще нет.

- Зайдите в *Настройки / Настройки продукта / Настройки модулей / TravelShop* и нажмите кнопку «Обновить базу». На экран в новом окне будут выведены надписи «Executing PHP script #2... ОК.», «Applying SQL script #2... ОК.» и т.д. Дождитесь появления надписи «Database is updated (...)».
- Обновите страницу в новом окне. Должна появиться надпись «Database is up to date».
- Снова выберите *Настройки / Настройки продукта / Настройки модулей / TAIS TravelShop* или обновите страницу (это необходимо сделать, т.к. обновление базы может повлиять на вид страницы настроек). После этого заполните поля «URL HTTP-интерфейса SIG», «IP-адрес SCI-интерфейса (XML) SIG», «Порт SCI-интерфейса (XML) SIG» соответствующими вашему SIG значениями.
- С этого момента с сайтом `http://адрес_сайта/` можно работать.

3.2 Обновление

- В панели управления сайта (http://адрес_сайта/bitrix/admin/) зайдите в *Контент / Структура сайта / Файлы и папки* и загрузите в папку `/bitrix/backup/` архивный файл `TravelShop travelshop.tar.gz` (создайте папку, если ее нет).
- Зайдите в *Настройки / Инструменты / Резервное копирование*. Нажмите правую кнопку мыши на архивном файле `travelshop.tar.gz` и выберите в открывшемся контекстном меню **«Распаковать»**. Следуйте дальнейшим инструкциям системы.
- Зайдите в *Настройки / Настройки продукта / Настройки модулей / TravelShop* и нажмите кнопку «Обновить базу». Дождитесь появления надписи «Database is updated (...)».
- Обновите страницу в новом окне. Должна появиться надпись «Database is up to date».
- С этого момента сайт можно считать обновленным.

3.3 Настройка

Необходимо добавить в Cron вызов каждые 5 минут следующей команды:

```
curl http://адрес_сайта/bitrix/components/travelshop/ibe.backoffice/async_run.php
```

4 ОБСЛУЖИВАНИЕ СИСТЕМЫ

4.1 Резервирование базы данных

Организуйте периодическое резервирование базы данных во время минимальной нагрузки на сервер, например, в ночные часы.

В ручном режиме это можно сделать средствами «1С-Битрикса» в разделе *Настройка / Резервное копирование*.

Более надежным является автоматическое резервирование по расписанию с использованием программы `mysqldump` с опциями `-- flush-logs`, `--lock-tables` и `--opt` (первая опция закрывает и открывает файлы журналов обновлений, вторая – блокирует все резервируемые таблицы, третья – активизирует все опции, ускоряющие считывание и резервирование данных). Используйте постоянную и легко понятную схему присвоения имен файлам архива, например, имя базы данных и дата резервирования. После создания файлы архивов лучше сжать, ведь они занимают много места. Время от времени рекомендуется удалять ненужные файлы архивов, так же как и файлы журналов, чтобы не заполнять жесткий диск ненужной информацией.

Резервируйте впоследствии архивные файлы MySQL с помощью средств резервирования файловой системы.

Размещайте файлы архивов и журналов на отдельном диске. Это снизит вероятность переполнения этими файлами диска, содержащего каталог данных, и потери архивной информации и журналов.

Обязательно активизируйте регистрацию обновлений. Журналы обновлений помогут вернуть базу данных в состояние, в котором она находилась перед сбоем. Для этого необходимо после восстановления заархивированных файлов базы данных запустить запросы журнала обновлений.

Для активизации процесса регистрации в MySQL используйте опцию `--log-bin`. В конфигурационном файле `my.cnf` можно задать имя файла журнала параметром `log_bin`.

4.2 Резервирование каталога веб-сайта

Организируйте периодическое резервирование каталога сайта (каталога Web-сервера).

В ручном режиме это можно сделать средствами «1С-Битрикса» в разделе *Настройка / Резервное копирование* и совместить с резервированием базы данных, получив единый архив сайта. Этот архив создается в папке `/bitrix/backup/` относительно корня веб-пространства и его можно скопировать в произвольное место средствами файловой системы.

Более надежным является автоматическое резервирование по расписанию средствами файловой системы. Размещайте файлы архивов сайта на отдельном диске. Это снизит вероятность переполнения этими файлами диска и их потери.