

25 декабря 2020г.

Оглавление

- [Начало](#)
- [Имя машины \(компьютера, сервера\)](#)
- [PostgreSQL](#)
 - [pgAdmin](#)
 - [Для возможности подключений к PostgreSQL с других компьютеров](#)
 - [Для предотвращения ошибок нехватки разделяемой памяти при бэкапе](#)
 - [Для возможности делать бэкапы SQL на локальной машине без пароля](#)
- [Эмулятор HASP](#)
 - [Дампы ключей](#)
 - [Пошагово для CentOS 7](#)
 - [Обновление модулей ядра после обновления ядра](#)
 - [USB Redirect Server](#)
 - [Настройки USB Redirect Server](#)
 - [Без HASPD](#)
 - [Несколько эмуляторов для нескольких клиентов на одной машине](#)
- [HTTPD \(Apache\)](#)
 - [Конфигурация в отдельном файле](#)
- [KDE](#)
- [1C](#)
 - [1C версий 8.3.18.*](#)
 - [1C версий 8.3.20.*](#)
- [x2Go](#)
 - [x2godesktopsharing](#)
 - [Аутентификация по ключу](#)
 - [Клиенты x2Go](#)
- [Полезные команды](#)
 - [Выгрузка в DT](#)
 - [Бэкап SQL](#)
 - [Восстановление из бэкапа SQL](#)
 - [Завершение сеансов на сервере 1C для заданной базы](#)

Начало

Рассмотрен запуск CentOS 7 с сервером и клиентами 1C и возможностью терминального доступа с помощью x2Go. Были попытки запустить на CentOS 8, но на данный момент, и клиенты 1C, и x2Go работали на нём крайне плохо, вплоть до того, что в x2Go не было некоторых пакетов и приходилось брать их из Fedora 33. Выделенный сервер 1C на CentOS 8 работал замечательно. Т.к., на момент установки, аппаратных ключей 1C не было, использован эмулятор HASP.

Устанавливается CentOS в минимальной конфигурации. Обновляется до актуальной версии. SELinux - отключить:

```
yum -y update
sed -i s/^SELINUX=.*$/SELINUX=disabled/ /etc/selinux/config
```

Крайне желательно отключить поддержку IPv6 либо добавив в параметры ядра

```
ipv6.disable=1
```

либо добавить в /etc/sysctl.conf

```
net.ipv6.conf.all.disable_ipv6=1
```

Для корректной работы 1C должна быть установлена русская локаль ru_RU.utf8

Правила IPTables для доступа других компьютеров к службам 1C и PostgreSQL средствами администрирования и клиентами:

```
# Server 1C
-A INPUT -p tcp -m state --state NEW --dport 1540 -j ACCEPT
-A INPUT -p tcp -m state --state NEW --dport 1541 -j ACCEPT
-A INPUT -p tcp -m state --state NEW --dport 1560:1591 -j ACCEPT
# PostgreSQL
-A INPUT -p tcp -m state --state NEW --dport 5432:5433 -j ACCEPT
```

Имя машины (компьютера, сервера)

Служба 1C крайне чувствительна к имени машины. Изменение имени может привести к неработоспособности службы.

В /etc/hosts **ОБЯЗАТЕЛЬНО** должно быть прописано соответствие имени компьютера его IP адресу. Без этого может не запуститься служба 1C.

Переименование машины:

```
hostnamectl set-hostname lin1c.test
```

и изменить имя машины в файлах, если кластер 1C уже существует:

```
/home/usr1cv8/.1cv8/1C/1cv8/1cv8wsrv.lst
/home/usr1cv8/.1cv8/1C/1cv8/reg_1541/1CV8Clst.lst
```

PostgreSQL

Установка из репозитория:

```
yum install http://repo.postgrespro.ru/pg1c-archive/pg1c-12.5/keys/postgrespro-1c-12.centos.yum-12-0.3.noarch.rpm
yum install -y postgrespro-1c-12-server postgrespro-1c-12-contrib
```

По умолчанию, устанавливается в "/opt/pgpro"
Файлы конфигураций находятся в "/var/lib/pgpro/1c-12/data"
Служба называется "postgrespro-1c-12"

От root-а выполнить:

```
/opt/pgpro/1c-12/bin/pg-setup initdb --locale=ru_RU.UTF-8
systemctl enable --now postgrespro-1c-12
```

Установить пароль для пользователя postgres (с этим именем и этим паролем будут подключаться базы 1C) (Пароль установлен то-же, что и для sa в MS SQL):

```
sudo -u postgres /opt/pgpro/1c-12/bin/psql
ALTER USER postgres WITH ENCRYPTED PASSWORD 'yourpassword';
\q
```

и проверка:

```
sudo -u postgres /opt/pgpro/1c-12/bin/psql -l
```

pgAdmin

```
yum -y install https://ftp.postgresql.org/pub/pgadmin/pgadmin4/yum/pgadmin4-redhat-repo-1-1.noarch
```

```
.rpm
yum -y install pgadmin4-desktop
```

<https://www.pgadmin.org/download/pgadmin-4-rpm/>

Отключить запрос мастер-пароля:

```
echo "MASTER_PASSWORD_REQUIRED=False" >> /usr/pgadmin4/web/config_local.py
```

Для возможности подключений к PostgreSQL с других компьютеров

Проверить, слушает-ли сервер запросы, кроме localhost-a. Строка в файле "/var/lib/pgpro/1c-12/data/postgresql.conf" должна быть такой (или с указанием IP адресов):

```
listen_addresses = '*'
```

В файле "/var/lib/pgpro/1c-12/data/pg_hba.conf" должна присутствовать строка:

```
host all all 0.0.0.0/0 md5
```

Для предотвращения ошибок нехватки разделяемой памяти при бэкапе

В файле "/var/lib/pgpro/1c-12/data/postgresql.conf" увеличить параметр "max_locks_per_transaction" в два раза. По умолчанию, он установлен в 64 и закомментирован.

Для возможности делать бэкапы SQL на локальной машине без пароля

В файле "/var/lib/pgpro/1c-12/data/pg_hba.conf" поменять

```
local ... peer
```

на

```
local ... trust
```

Эмулятор HASP

Не установится в контейнер LXC или в подобные, потому, что используются самосборные модули ядра. Устанавливать только в виртуальную (или реальную) машину.

[mini_manual.pdf](#): Инструкция по самостоятельной сборке (PDF)
[install_hasp_centos7.sh](#): Скрипт от castorsky для автоматической сборки

Дампы ключей

[1c_server_x64.json](#): Ключ для сервера 1С
[x86_server.json](#): Добавочный серверный ключ, если с сервером предусмотрена работа толстых клиентов x32
[50user.json](#): Серверный ключ пользовательских лицензий на 50 пользователей
[100user.json](#): Серверный ключ пользовательских лицензий на 100 пользователей
[300user.json](#): Серверный ключ пользовательских лицензий на 300 пользователей

В 1С "О программе" эти ключи идентифицируются, как "Локальный HASP4 EN8SA 1" и "Сетевой HASP4 ORGL8 50|100|300".

[linux-hasp.txt](#): Полное описание (TXT)

Пошагово для CentOS 7

Фактически повторяет команды из скрипта, но без комментариев и проверок.

Запоминание для текущей сессии версии ядра в переменную (понадобится в вызываемых командах) и установка необходимых пакетов:

```
export KVER=`uname -r`
yum install -y epel-release
yum install -y gcc gcc-c++ make kernel-devel jansson-devel libusb elfutils-libelf-devel git usbutils
ls tar chkconfig
```

В какой-либо пустой папке:

```
curl -Os http://download.etersoft.ru/pub/Etersoft/HASP/last/CentOS/7/haspd-7.90-eter2centos.x86_64.rpm
curl -sL https://sourceforge.net/projects/usb-vhci/files/linux%20kernel%20module/vhci-hcd-1.15.tar.gz/download > vhci-hcd-1.15.tar.gz
curl -sL https://sourceforge.net/projects/usb-vhci/files/native%20libraries/libusb_vhci-0.8.tar.gz/download > libusb_vhci-0.8.tar.gz
git clone https://github.com/sam88651/UsbHasp.git
tar -xpf libusb_vhci-0.8.tar.gz
tar -xpf vhci-hcd-1.15.tar.gz
```

```
yum -y install haspd-7.90-eter2centos.x86_64.rpm
systemctl enable haspd
```

```
cd vhci-hcd-1.15
mkdir -p linux/${KVER}/drivers/usb/core
cp /usr/src/kernels/${KVER}/include/linux/usb/hcd.h linux/${KVER}/drivers/usb/core
sed -i 's/\#define DEBUG/\#\#define DEBUG/' usb-vhci-hcd.c
sed -i 's/\#define DEBUG/\#\#define DEBUG/' usb-vhci-iocifc.c
make -s KVERSION=${KVER}
make -s install
echo "usb_vhci_hcd" >> /etc/modules-load.d/usb_vhci.conf
modprobe usb_vhci_hcd
echo "usb_vhci_iocifc" >> /etc/modules-load.d/usb_vhci.conf
modprobe usb_vhci_iocifc
cd ..
```

```
cd libusb_vhci-0.8
./configure
make -s
make -s install
echo "/usr/local/lib" >> /etc/ld.so.conf.d/libusb_vhci.conf
ldconfig
cd ..
```

```
cd UsbHasp
sed -i 's/\(CC=gcc\)\/\1 -std=gnu11/' nbproject/Makefile-Release.mk
make -s
cp dist/Release/GNU-Linux/usbhasp /usr/local/sbin
mkdir /etc/usbhaspkey/
```

```
cat <<EOF >> /etc/systemd/system/usbhaspemul.service
[Unit]
Description=Emulation HASP key for 1C
Requires=haspd.service
After=haspd.service
```

```
[Service]
Type=simple
ExecStart=/usr/bin/sh -c 'find /etc/usbhaspkey -name "*.json" | xargs /usr/local/sbin/usbhasp'
Restart=always
```

```
[Install]
WantedBy=multi-user.target
EOF
```

```
systemctl daemon-reload
```

В папку "/etc/usbhaspkey/" положить файлы дампов ключей и запустить службу эмулятора:

```
systemctl enable --now usbhaspemul
```

Обновление модулей ядра после обновления ядра

При обновлении ядра перестанут работать vhci-hcd и libusb_vhci. Перекомпилировать их можно следующим скриптом (в папке со скриптом должны быть папки с распакованными vhci-hcd-1.15 и libusb_vhci-0.8, можно использовать папки от предыдущей компиляции):

```
# cat after-kernel-update

#!/bin/sh

# Обновление модулей ядра после обновления ядра
# Должны присутствовать папки:
# vhci-hcd-1.15
# libusb_vhci-0.8

KVER=`uname -r`

cd vhci-hcd-1.15
mkdir -p linux/${KVER}/drivers/usb/core
cp /usr/src/kernels/${KVER}/include/linux/usb/hcd.h linux/${KVER}/drivers/usb/core
make -s KVERSION=${KVER}
make install
modprobe usb_vhci_hcd
modprobe usb_vhci_iocifc
cd ..

cd libusb_vhci-0.8
./configure --enable-shared
make -s
make install
ldconfig

systemctl restart usbhaspemul
systemctl restart haspd

echo "*****"
lsusb | grep HASP
echo "*****"
/usr/sbin/usbkeytest --list
echo "*****"
systemctl status usbhaspemul
echo "*****"
systemctl status haspd
echo "*****"
systemctl status srvcv83
```

Проверка наличия подключенных ключей:

```
# lsusb | grep HASP
Bus 009 Device 004: ID 0529:0001 Aladdin Knowledge Systems HASP copy protection dongle
Bus 009 Device 003: ID 0529:0001 Aladdin Knowledge Systems HASP copy protection dongle
Bus 009 Device 002: ID 0529:0001 Aladdin Knowledge Systems HASP copy protection dongle
```

Проверка работы этих ключей (если не запущен haspd и usbhaspemul, выхлоп пустой):

```
# /usr/sbin/usbkeytest --list
ID: 0529:0001, Bus: 009, Device: 004, Major: 189, Minor: 1027
```

```
ID: 0529:0001, Bus: 009, Device: 003, Major: 189, Minor: 1026
```

```
ID: 0529:0001, Bus: 009, Device: 002, Major: 189, Minor: 1025
```

USB Redirect Server

Установка сервера "USB Redirect" для возможной передачи ключей на другую машину. В этом случае, локально, ключи использовать нельзя:

```
curl -sL http://www.incentivespro.com/usb-redirector-linux-x86_64.tar.gz > usbredirector.tar.gz
tar -xpf usbredirector.tar.gz
cd usb-redirector-linux-x86_64
sh ./installer.sh install-server
usbsrv -autoshareon
usbsrv -createcallback 192.168.0.0:32032
```

Настройки USB Redirect Server

Показать все устройства:

```
usbsrv -l
```

Не "расшаривать" устройство:

```
usbsrv -t 1
```

Задать имя устройства:

```
usbsrv -nickname "Reserved" 8
```

Без HASPD

Пакет "haspd-7.90-eter2centos.x86_64.rpm" включает в себя драйверы для HASP и службу HASPD для раздачи ключей по сети.

Если раздавать ключи по сети нет необходимости, то данный пакет можно не устанавливать или отключить.

Если машина с CentOS 7 используется для раздачи ключей посредством "USB Redirect Server", то драйверы HASP на ней не требуются, потому, что они устанавливаются на машинах с клиентами "USB Redirect".

Если-же, машина с CentOS 7 используется, как сервер 1C, то вместо HASPD можно установить драйверы HASP от Sentinel:

Со страницы загрузки (<https://thales-sentinel.ru/helpdesk/download-space/>) скачать "Sentinel_LDK_Run-time_linux.zip", найти в архиве "Sentinel_LDK_Linux_Run-time_Installer_script/aksusbd-8.21.1/pkg/aksusbd-8.21-1.x86_64.rpm" и установить его.

Чтобы служба эмулятора запускалась без наличия HASPD, необходимо закомментировать или удалить строки в файле "/etc/systemd/system/usbhaspemul.service":

```
Requires=haspd.service
After=haspd.service
```

Несколько эмуляторов для нескольких клиентов на одной машине

Для раздачи ключей посредством "USB Redirect" для нескольких клиентов, можно запустить несколько служб "usbhaspemul" с уникальными именами, например, вместо файла "/etc/systemd/system/usbhaspemul.service", создать несколько файлов "hasp-1.service...hasp-N.service" включить и запустить их (можно пакетным файлом):

```
systemctl enable --now hasp-1
.
.
.
systemctl enable --now hasp-N
```

В этих файлах необходимо закомментировать строки:

```
Requires=haspd.service  
After=haspd.service
```

После каждого изменения в вышеуказанных файлах, необходимо выполнять:

```
systemctl daemon-reload
```

Службу HASPD можно отключить или удалить.

Файлы

Архивы и пакеты, используемые в командах (скачаны 25.12.2020г.):

[haspd-7.90-eter2centos.x86_64.rpm](#)
[libusb_vhci-0.8.tar.gz](#)
[UsbHasp.zip](#)
[usbredirector.tar.gz](#)
[vhci-hcd-1.15.tar.gz](#)
[aksusbd-8.21-1.x86_64.rpm](#)

HTTPD (Apache)

```
yum install httpd  
systemctl enable httpd --now
```

Регистрация базы test_buh на http сервере:

```
webinst -apache24 -wsdir test_buh -dir /var/www/html/test_buh -connstr "Srvr=linlc;Ref=test_buh;"  
-confPath /etc/httpd/conf/httpd.conf  
systemctl reload httpd
```

Удаление публикации (таже строка с первым параметром -delete):

```
webinst -delete -apache24 -wsdir test_buh -dir /var/www/html/test_buh -connstr "Srvr=linlc;Ref=tes  
t_buh;" -confPath /etc/httpd/conf/httpd.conf  
systemctl reload httpd
```

Конфигурация в отдельном файле

Лучший результат достигается при использовании отдельного файла для публикации баз на WEB сервере. Например, в файле "/etc/httpd/conf.d/1c.conf":

```
webinst -apache24 -wsdir test_buh -dir /var/www/html/test_buh -connstr "Srvr=linlc;Ref=test_buh;"  
-confPath /etc/httpd/conf.d/1c.conf
```

В этом случае, конфигурация находится в файле "/etc/httpd/conf.d/1c.conf" и имеет такой вид:

```
LoadModule _1cws_module "/opt/1C/v8.3/x86_64/wsap24.so"  
  
# 1c publication  
Alias "/test_buh" "/var/www/html/test_buh/"  
<Directory "/var/www/html/test_buh/">  
    AllowOverride All  
    Options None  
    Require all granted  
    SetHandler 1c-application
```

```
ManagedApplicationDescriptor "/var/www/html/test_buh/default.vrd"
</Directory>
```

KDE

```
yum -y groupinstall "KDE Plasma Workspaces"
yum -y install kde-l10n-Russian
```

Чтобы в KDE отображались флаги стран в переключателе раскладок, можно установить пакет [kde-runtime-flags-4.13.3-3.fc20.noarch.rpm](https://rpmfind.net/linux/rpmdb/view.php?name=kde-runtime-flags-4.13.3-3.fc20.noarch.rpm):

```
rpm -ivh --nodeps kde-runtime-flags-4.13.3-3.fc20.noarch.rpm
```

1C

Брать можно отсюда - <https://rebel666.com/platf82.html>
Необходимые пакеты для установки сервера и клиентов 1C:

1C_Enterprise83-server - Сервер 1C
1C_Enterprise83-client - Толстый и тонкий клиенты
1C_Enterprise83-common - Общие файлы для толстого клиента и сервера 1C
1C_Enterprise83-ws - Необязательный пакет для доступа к серверу 1C WEB клиентов. Работает совместно с HTTPD

Установить шрифты Windows ([msttcore-fonts-2.0-3.noarch.rpm](https://rpmfind.net/linux/rpmdb/view.php?name=msttcore-fonts-2.0-3.noarch.rpm)). Об этом 1C будет постоянно настаивать.
Для возможности удалённого управления кластером 1C нужно запускать при старте машины (например, из /etc/rc.d/rc.local):

```
/opt/1C/v8.3/x86_64/ras --daemon cluster
```

Служба 1C - srv1cv83
Файл конфигурации сервера 1C: /etc/sysconfig/srv1cv83

1C версий 8.3.18.*

С этих версий платформа стала устанавливаться по аналогии с Windows. Стал возможным запуск нескольких серверов 1C различных версий.

Для этого был изменён путь установки компонентов 1C и, соответственно, путь к бинарникам.
Теперь бинарники 1C находятся не в папке "/opt/1C/v8.3/x86_64", а в "/opt/1cv8/x86_64/8.3.18.1208". Если на сервере не предполагается использование нескольких версий 1C одновременно, для запуска бинарников, как ранее, рекомендуется создать символическую ссылку:

```
ln -s /opt/1cv8/x86_64/8.3.18.1208 /opt/1C/v8.3/x86_64
```

И изменять её при каждом изменении версии платформы.

Также, изменился способ конфигурирования службы. Теперь инсталлятор не создаёт службу сам, а предоставляет это администратору.

Чтобы сделать запуск службы по аналогии с предыдущими версиями, после установки необходимо:

- Скопировать файл "/opt/1cv8/x86_64/8.3.18.1208/srv1cv83" в папку "/etc/init.d/"
- Скопировать файл "/opt/1cv8/x86_64/8.3.18.1208/srv1cv83.conf" в папку "/etc/sysconfig/" и переименовать его (в папке "/etc/sysconfig/") в "srv1cv83"
- Выполнить регистрацию службы:

```
systemctl daemon-reload
```

- Проверить наличие службы и возможность её автозапуска:


```
chkconfig
```

- Запустить службу:

```
systemctl start srv1cv83
```

Сервер 1C версии 8.3.18.1208 не заработал с PostgreSQL для 1C версии 12.1. PostgreSQL был обновлён до версии 12.5 из репозитория PostgreSQL 1C.

1C версий 8.3.20.*

Версии с платформонезависимым установщиком (как в Windows). По умолчанию установщик запускается в графическом режиме. Если запускать его в удалённой консоли, его окна пробрасываются в иксы клиенту.

Установщик запускать из-под root-а.

Установка в текстовой консоли в "тихом" режиме:

```
./setup-full-8.3.20.1549-x86_64.run --mode unattended --enable-components client_full,client_thin,ru,server,ws,server_admin,additional_admin_functions
```

При установке не регистрируется служба и не копируется файл конфигурации службы. Делать это вручную, как в предыдущих версиях.

Для удаления в текстовой консоли, в папке с бинарниками, запустить:

```
./uninstaller-full --mode unattended
```

x2Go

ВНИМАНИЕ! Устанавливать x2Go надо **ПОСЛЕ** установки KDE и пакета "kde-l10n-Russian", иначе в KDE, возможно, будет нарушение локализации - часть останется на английском языке и локаль в KDE будет POSIX, хотя вне KDE, у пользователя будет ru_RU.utf8.

Пакеты, необходимые для установки сервера x2go:

```
x2goserver-xsession
x2goserver
cups-x2go
x2goagent
x2goserver-common
x2goserver-printing
x2godesktopsharing
x2goserver-desktopsharing
```

```
yum -y install x2goserver-xsession x2goserver cups-x2go x2goagent x2goserver-common x2goserver-printing x2goserver-desktopsharing x2godesktopsharing
```

ВНИМАНИЕ! Для разрешения печати, необходимо в файле "/usr/lib/systemd/system/cups.service" поменять "PrivateTmp=true" на "PrivateTmp=false" и

```
systemctl daemon-reload && systemctl restart cups
```

Чтобы пользователю выводилось меньше окон требующих дополнительный ввод пароля root-а, можно создать файл, например, "/usr/share/polkit-1/rules.d/10-x2go-policy.rules" со следующим содержимым:

```
polkit.addRule(function(action, subject) {
  if (action.id == "org.freedesktop.login1.set-user-linger") {
    return polkit.Result.YES;
  }
});
```

```

    }
});

polkit.addRule(function(action, subject) {
    if (action.id == "org.freedesktop.color-manager.create-device" ||
        action.id == "org.freedesktop.color-manager.create-profile" ||
        action.id == "org.freedesktop.color-manager.delete-device" ||
        action.id == "org.freedesktop.color-manager.delete-profile" ||
        action.id == "org.freedesktop.color-manager.modify-device") {
        return polkit.Result.YES;
    }
});

```

Если будут выскакивать ещё какие-либо запросы, можно отследить, какие правила нужно переназначить, предварительно запустить слежение за журналом:

```
journalctl -u polkit -f
```

Если запуск программ через x2Go производится без использования рабочего стола, можно использовать переключение раскладки клавиатуры средствами X сервера:

```
setxkbmap -layout 'us,ru' -option 'grp:alt_shift_toggle,grp_led:scroll'
```

x2godesktopsharing

ВНИМАНИЕ Для доступа к сессии пользователя, и администратор и пользователь должны входить в группу "x2godesktopsharing", а у пользователя должен быть запущен "x2godesktopsharing" с включенным разрешением подключений.

Можно запустить "x2godesktopsharing" в автозапуске пользователя с ключом "--activate-desktop-sharing"

```
x2godesktopsharing --activate-desktop-sharing
```

Администраторов, которым разрешено подключение к пользователю, необходимо указать в файле профиля пользователя "/home/user1c/.x2godesktopsharing/settings", параметр "whitelist", например:

```
whitelist=admin,gendir
```

Выявилась проблема, что при отключении пользователя от сеанса и повторного входа в него, "x2godesktopsharing" запускался в режиме запрета подключений.

Были взяты исходники "x2godesktopsharing" и в файле "sharetray.cpp", в районе 233-строки, изменено (добавлен вызов "slotStartSharing()"):

```
loadSystemSettings();
loadUserSettings();
setTrayIcon();
trayIcon->show();
```

на:

```
slotStartSharing();
loadSystemSettings();
loadUserSettings();
setTrayIcon();
trayIcon->show();
```

Для включения "тихого" режима, когда не показывается значок и не выводятся какие-либо сообщения, закомментировать вызов "trayIcon->show()".

Откомпилированный вариант - [x2godesktopsharing](#).

Так-же это изменение влияет на повторное подключение после обрыва сеанса. Но для этого бинарник необходимо поместить в "/usr/bin/", иначе, при повторном подключении, вызывается неизменённый вариант.

Чтобы "x2godesktopsharing" запускался для всех пользователей при начале сеанса, можно создать файл "/etc/X11/xinit/xinitrc.d/x2godesktopsharing" со следующим содержимым:

```
#!/bin/sh
x2godesktopsharing &>/dev/null &
```

В этом случае не требуется запускать "x2godesktopsharing" у каждого пользователя в автозапуске.

Аутентификация по ключу

Допустим, что на клиентском компьютере с Windows, пользователь "client", подключается к серверу x2Go с именем пользователя "user1c", то манипуляция с ключами выглядит так:

В клиенте Windows в качестве "RSA/DSA ключ для ssh соединения" указывается приватный ключ пользователя "C:/Users/client/.x2go/etc/ssh_host_rsa_key".

Публичный ключ пользователя "C:/Users/client/.x2go/etc/ssh_host_rsa_key.pub" добавляется на сервер в файл пользователя "/home/user1c/.ssh/authorized_keys".

Клиенты x2Go

Все версии клиентов для Windows: <https://code.x2go.org/releases/binary-win32/>

"Ночные" сборки для Windows: <https://code.x2go.org/releases/binary-win32/x2goclient/heuler/mingw32-4.8/qt-4.8/>

Полезные команды

Выгрузка в DT

Выгрузка в DT в текущую папку (можно запускать из под root):

```
/opt/1C/v8.3/x86_64/1cv8 CONFIG /S "lin1c\BASE" /N 1CUser /P 1CPassword /DumpIB "BASE-$(date +%y%m%d-%H%M%S).dt" /Out "BASE-$(date +%y%m%d-%H%M%S).log"
```

Для выгрузки в пакетном режиме, лучше использовать Xvfb (не выводит окна 1C клиента на экран):

```
yum install xorg-x11-server-Xvfb
xvfb-run /opt/1C/v8.3/x86_64/1cv8 CONFIG ...
```

Бэкап SQL

Бэкап SQL в bzip2 (самый долгий с наилучшим сжатием):

```
pg_dump -U postgres -w BASE | bzip2 -9 > "BASE-$(date +%y%m%d-%H%M%S).bz2"
```

Бэкап SQL (самый быстрый, без сжатия):

```
pg_dump -U postgres -w BASE > "BASE-$(date +%y%m%d-%H%M%S).sql"
```

Бэкап SQL в gzip (оптимальное соотношение скорости и сжатия):

```
pg_dump -U postgres -w BASE | gzip > "BASE-$(date +%y%m%d-%H%M%S).gz"
```

Восстановление из бэкапа SQL

Бэкап восстанавливается, исключительно, в пустую базу. Иначе, при восстановлении, появляются ошибки и, впоследствии, база не работает!

```
dropdb -U postgres BASE_NAME
createdb -U postgres BASE_NAME
gunzip -c BASE_NAME.gz | psql -U postgres BASE_NAME
```

Завершение сеансов на сервере 1C для заданной базы

Необходим запущенный "ras"

[1c-sessions-close](#)

```
#!/bin/bash

# Завершает сеансы заданной базы на сервере 1C
# Имя базы задаётся параметром командной строки, например:
# 1c-sessions-close ERP_DEMO

if [[ "$1" == "" ]]
then
    echo "Нет параметров в командной строке"
    echo "1c-sessions-close <ИМЯ_БАЗЫ>"
    exit
fi

base_name="$1"

cluster=$(rac cluster list | grep cluster | awk '{print $3}')
echo -e "Cluster:\t$cluster"
infobase=$(rac infobase --cluster=$cluster summary list | grep -w $base_name -B 1 | grep infobase | awk '{print $3}')
echo -e "InfoBase:\t$infobase"
sessions=$(rac session list --cluster=$cluster | grep $infobase -B 2 | grep "session " | awk '{print $3}')

if [[ "$sessions" == "" ]]
then
    echo "В базе $base_name нет активных сеансов."
    exit
fi

echo "$sessions" | while read session; do
    echo -e "Session:\t$session"
    rac session terminate --cluster=$cluster --session=$session
done
```

Файлы

install_hasp_centos7.sh	4,46 КБ	25.12.2020	Oleg
after-kernel-update.sh	478 байта	25.12.2020	Oleg
mini_manual.pdf	122 КБ	25.12.2020	Oleg
50user.json	4,82 КБ	25.12.2020	Oleg
1c_server_x64.json	20,6 КБ	25.12.2020	Oleg
100user.json	4,85 КБ	25.12.2020	Oleg
300user.json	4,86 КБ	25.12.2020	Oleg
x86_server.json	2,58 КБ	25.12.2020	Oleg
linux-hasp.txt	3,93 КБ	25.12.2020	Oleg
libusb_vhci-0.8.tar.gz	357 КБ	25.12.2020	Oleg
UsbHasp.zip	34,3 КБ	25.12.2020	Oleg
haspd-7.90-eter2centos.x86_64.rpm	2,18 МБ	25.12.2020	Oleg
usbredirector.tar.gz	895 КБ	25.12.2020	Oleg
vhci-hcd-1.15.tar.gz	56,3 КБ	25.12.2020	Oleg
kde-runtime-flags-4.13.3-3.fc20.noarch.rpm	139 КБ	25.12.2020	Oleg

x2godesktopsharing	606 КБ	25.12.2020	Oleg
msttcore-fonts-2.0-3.noarch.rpm	3,32 МБ	25.12.2020	Oleg
1c-sessions-close	1014 байта	26.12.2020	Oleg
aksusbd-8.21-1.x86_64.rpm	8,52 МБ	07.06.2021	Oleg