Имеем фаршированный сервер HP DL380G6 с установленным на нем гипервизором VMware ESXi 4.1 Будем ставить 64 битный сервер 1С версии 8.2.15-289 + PostgreSQL 9.0.4 от Etersoft на x86_64 CentOS 6.1 (сервер 1С 8.2.14.540 не заработал без костылей. Постоянно отваливался ragent и в ТЖ писал всяческую чушь про Address already in use…) Качаем http://mirror.yandex.ru/centos/6.1/isos/x86_64/CentOS-6.1-x86_64-minimal.iso Создаем виртуальную машину в гипервизоре и ставим.

Затем:{codecitation class="light:true;"}yum update{/codecitation}ставим все жизненно необходимое:{codecitation class="light:true;"}yum install crontabs tmpwatch sudo wget file{/codecitation}Подправим файл /etc/hosts (Клиент 1С и сам сервер исключительно общается только по именам) оставим там только строки:
{codecitation class="light:true;"}127.0.0.1 localhost
192.168.1.1 srv1cv82{/codecitation}отключаем файервол, дабы не мешался пока:{codecitation class="light:true;"}chkconfig iptables off
service iptables stop
chkconfig ip6tables off
service ip6tables stop{/codecitation}отключаем SELinux (1С предыдущих версий не работал с ним, да и нам на сервере в локальной сети он особо не нужен) правим файл /etc/sysconfig/selinux {codecitation class="light:true;"}SELINUX=disabled{/codecitation}Теперь добавляем необходимые репозитории и ставим все, что потребуется серверу 1С: {codecitation class="light:true;"}rpm -ivh
http://dl.atrpms.net/el6.1-x86_64/atrpms/stable/atrpms-repo-6-4.el6.x86_64.rpm
rpm -ivh http://download.fedora.redhat.com/pub/epel/6/x86_64/epel-release-6-5.noarch.rpm
yum install xorg-x11-xfs chkfontpath ttf2pt1 unixODBC
rpm -ivh
http://elders.princeton.edu/data/puias/unsupported/5/x86_64/msttcorefonts-2.0-1.noarch.rpm
yum install ImageMagick{/codecitation}По зависимостям на сервер свалится куча всякого. Возможно, возникнет ошибка вида: /usr/bin/update-gdk-pixbuf-loaders: line 54: /etc/gtk-2.0/x86_64-unknown-linux-gnu/gdk-pixbuf.loaders: No such file or directory
, если так, то делаем небольшой извврат:{codecitation class="light:true;"} cd /etc/gtk-2.0/
ln -s x86_64-redhat-linux-gnu x86_64-unknown-linux-gnu
yum reinstall gtk2
rm x86_64-unknown-linux-gnu{/codecitation}Теперь все готово для установки постгреса и сервера 1С. перезагружаемся и скачиваем все rpm кроме postgre-etersoft9.0-seltaaddon-9.0.4-eter8rhel.x86_64.rpm отсюда: http://updates.etersoft.ru/pub/Etersoft/Postgres@Etersoft/stable/x86_64/CentOS/6/
, далее напускаем на скаченное:{codecitation class="light:true;"}yum localinstall --nogpgcheck *.rpm{/codecitation}инициализируем базу::{codecitation class="light:true;"}su postgres
initdb -D /var/lib/pgsql/data --locale=ru_RU.UTF-8
exit{/codecitation}Добавляем постгрес в автозагрузку и стартуем его:{codecitation class="light:true;"}chkconfig postgresql on
service postgresql start{/codecitation}Задаем пароль для пользователя постгреса:{codecitation class="light:true;"}su postgres

psql postgres
postgres=# alter user postgres with password 'YOUR_PASSWORD';
ALTER ROLE
postgres=# q
exit{/codecitation}с базой будем соединяться локально, поэтому подправим файл /var/lib/pgsql/data/pg_hba.conf
раскомментируем строку {codecitation class="light:true;"}host all all 127.0.0.1/32 trust{/codecitation}и заменим
trust
на
md5
{codecitation class="light:true;"}host all all 127.0.0.1/32 md5{/codecitation}Далее правим /var/lib/pgsql/data/postgresql.conf
{codecitation class="light:true;"}listen_addresses = '127.0.0.1' # будем конектиться только локально
effective_cache_size = 16384MB # выставим в половину от установленной на сервер памяти (в моем случае под виртуалку выделил 32 гига)
fsync = on # оставил как есть, т.к. у меня SCSI контроллер с включенным write-back. fsync = off дает прирост в 1 попугай в тесте Гилёва, поэтому посчитал надежность важнее{/codecitation}остальные настройки требуют детального изучения и ставяться исходя из вашего железа и базы 1С. Все, постгрес настроен, переходим к установке сервера 1С: качаем технологическую платформу 8.2 http://dl01.v8.1c.ru/get/Info/Platform/8_2_15_289/rpm64.tar.gz (нужно авторизоваться на сайте
http://users.v8.1c.ru
) распаковываем архив и ставим:{codecitation class="light:true;"}yum localinstall --nogpgcheck *.rpm{/codecitation}затем запускаем скрипт
config_server
{codecitation class="light:true;"}cd /opt/1C/v8.2/x86_64/utils/
./config_server{/codecitation} если выдаст ls: cannot access /lib/libc-*.so: No such file or directory то поставим 32 битные либы glibc
{codecitation class="light:true;"}yum install glibc.i686{/codecitation} также может ругнуться на отсутствие libglib (потому что ищет его в /usr/lib64, а он находится в /lib64) делаем символьную ссылку {codecitation class="light:true;"}ln -s /lib64/libglib-2.0.so.0.2200.5 /usr/lib64/libglib-2.0.so{/codecitation} запускаем скрипт еще раз и если он призадумался и ничего не выдал, то все OK! результатом должен быть файл
type.xml
созданный в
/opt/1C/v8.2/x86_64/conf/grcmncfg
Теперь нужно запустить агент сервера ragent в режиме демона.{codecitation class="light:true;"} /etc/rc.d/init.d/srv1cv82 stop
/opt/1C/v8.2/x86_64/ragent -daemon
/etc/rc.d/init.d/srv1cv82 restart{/codecitation} Т.к. клиент 1С-ки соеденяется с сервером только по имени, то ставим самбу (можно как вариант прописать соответствие IP к имени на клиентских машинах в файл hosts) {codecitation class="light:true;"}yum install samba{/codecitation} Правим /etc/samba/smb.conf под свою сеть Ставим на автостарт

демон nmbd (если нужно только NETBIOS имя) {codecitation class="light:true;"}chkconfig nmb on{/codecitation}

На этом сервер готов (1С-ка покупалась с софтварными лицензиями, тем самым избежал увлекательного приключения с HASP ключами), переходим к клиентским машинам.
Ставим клиент с компонентом «Администрирование сервера 1С:Предприятия»
Запускаем «Администрирование серверов 1С Предприятия» Кликаем правой кнопкой по «Central 1C:Enterprise 8.2 servers» и создаем «Центральный сервер 1С:Предприятия 8.2″
Запускаем «1С Предприятие» и создаем базу:
В диалоге «Добавление информационной базы/группы» выбираем «Тип СУБД:» PostgreSQL
во вкладке «Сервер баз данных» прописываем 127.0.0.1
пользователь: postgres и соответственно ваш пароль, который задавали после инициализации базы постгреса.

На последок можно задействовать простенькие правила файервола: {codecitation class="brush:bash;"}#!/bin/bash
Очищаем предыдущие записи
iptables -F
Установка политик по умолчанию
iptables -P INPUT DROP
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
Разрешаем локальный интерфейс
iptables -A INPUT -i lo -j ACCEPT
Закрываемся от кривого icmp
iptables -I INPUT -p icmp -f -j DROP
REL, ESTB allow
iptables -A INPUT -p tcp -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp -m state --state RELATED,ESTABLISHED -j ACCEPT
Защита сервера SSH от брутфорса
iptables -A INPUT -p tcp --syn --dport 22 -m recent --name root --set
iptables -A INPUT -p tcp --syn --dport 22 -m recent --name root --update --seconds 30 --hitcount 3 -j DROP
Разрешаем рабочие порты
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p udp --dport 137 -j ACCEPT
iptables -A INPUT -p tcp --dport 1540 -j ACCEPT
iptables -A INPUT -p tcp --dport 1541 -j ACCEPT
iptables -A INPUT -m multiport -p tcp --dports 1560:1591 -j ACCEPT
Разрешение главных типов протокола ICMP
iptables -A INPUT -p icmp --icmp-type 3 -j ACCEPT
iptables -A INPUT -p icmp --icmp-type 8 -j ACCEPT iptables -A INPUT -p icmp --icmp-type 11 -j ACCEPT

iptables -A INPUT -p icmp --icmp-type 12 -j ACCEPT {/codecitation}ну и кинуть в /etc/cron.daily скрипт бэкапа/оптимизации базы. Я бэкаплю базу на виндовую шару (нужно поставить cifs-utils {codecitation class="light:true;"}yum install cifs-utils{/codecitation} для монтирования шары):{codecitation class="brush:;"}#!/bin/sh

```
OLDDAYS=60
DATE=`date +%F`
BACKUPDIR=/mnt/tmp.grnt/base_main/daily
BACKUPNAME=${BACKUPDIR}/main.psql.${DATE}
echo `date +%a_%d_%T` `hostname` Begin vacuum MAIN BASE >> /var/log/cron
vacuumdb -d main -z -f -q -U postgres
echo `date +%a_%d_%T` `hostname` Vacuum MAIN BASE is DONE >> /var/log/cron
mount -t cifs -o user=your_user_name,pass=your_password "//192.168.1.50/backups41C"
/mnt/tmp.grnt
if [ -d /mnt/tmp.grnt/base_main/daily ] ; then
   echo `date +%a_%d_%T` `hostname` Begin dumping MAIN BASE into ${BACKUPNAME} >>
/var/log/cron
   pg_dump -U postgres -Fc -Z9 -c -f ${BACKUPNAME} main
   echo `date +%a_%d_%T` `hostname` End dumping MAIN BASE into ${BACKUPNAME} >>
/var/log/cron
   # Deleting old backups
   find ${BACKUPDIR} -mtime +${OLDDAYS} -delete
   umount /mnt/tmp.grnt
  else
   echo `date +%a_%d_%T` `hostname` Backup base MAIN FAILED!!! >> /var/log/cron
fi
## RESTORE COMMAND ##
##
## pg_restore -U postgres -c -d DATABASE_NAME -Fc DATABASE_BACKUP_FILE
##
#####################{/codecitation}
```

Оригинал данной публикации на www.achlab.ru