

VirtualBox + phpvirtualbox

Введение

Иногда бывает необходимо, что-то протестировать или собрать какой-либо rpm пакет, но при этом совсем не хочется устанавливать на боевые сервера все необходимые devel пакеты. Я часто для этих целей использую виртуальные машины. Так же часто бывает необходимость проверить работу программы или связки на определенном дистрибутиве или версии ПО и в этой ситуации виртуальные машины значительно облегчают жизнь системным администраторам, и не только им.

На данный момент бесплатных решений в сегменте настольных виртуальных машин не так уж и много. Vmware к сожалению забросила свой продукт Vmware server, поэтому, думаю многие из вас используют VirtualBox развиваемый в данный момент Oracle. Итак, давай те рассмотрим пример установки и настройки VirtualBox на CentOS-6.3.

Установка VirtualBox

У нас в распоряжении имеется следующий компьютер - Intel Core 2 Duo 6320 @ 1.86GHz, 4 Gb Ram. Назвать его сервером просто язык не поворачивается, но тем не менее, со своими задачами, а это запуск 3-4 виртуальных машин он отлично справляется. На данном компьютере установлен CentOS-6.3 минимальная конфигурация с последними обновлениями.

```
# cat /etc/redhat-release
CentOS release 6.3 (Final)

# uname -a
Linux vb.example.net 2.6.32-279.2.1.el6.x86_64 #1 SMP Fri Jul 20 01:55:29
UTC 2012 x86_64 x86_64 x86_64 GNU/Linux
```

Устанавливаем две вспомогательных программы, облегчающих нам жизнь

```
# yum install wget mc
```

Подключаем репозитории [EPEL](#) и [RPMForge](#)

```
# wget http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt
# rpm --import RPM-GPG-KEY.dag.txt
# wget
http://pkgs.repoforge.org/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x
86_64.rpm

# rpm -K rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm: (sha1) dsa sha1 md5 gpg OK

# rpm -ivh rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
```

```

Preparing... #####
[100%]
  1:rpmforge-release #####
[100%]

# wget http://ftp.tlk-l.net/pub/mirrors/epel/RPM-GPG-KEY-EPEL-6
# rpm --import RPM-GPG-KEY-EPEL-6
# wget
http://ftp.tlk-l.net/pub/mirrors/fedora-epel/6/x86_64/epel-release-6-7.noarch.rpm

# rpm -K epel-release-6-7.noarch.rpm
epel-release-6-7.noarch.rpm: rsa sha1 (md5) pgp md5 OK

# rpm -ivh epel-release-6-7.noarch.rpm
Preparing... #####
[100%]
  1:epel-release #####
[100%]

```

А также ставим пакеты, которые нам потребуются для сборки DKMS модулей ядра, а так же запускаем сервис dkms. Сервис dkms гарантирует нам, что при обновлении ядра на хосте, автоматически будут собраны новые версии всех модулей, чтобы не было необходимости каждый раз после обновления ядра запускать команду vboxdrv setup.

```

# yum install dkms kernel-headers kernel-devel make gcc-c++
# service dkms_autoinstaller start
dkms: running auto installation service for kernel 2.6.32-279.2.1.el6.x86_64
Done.

```

Подключаем репозиторий самого VirtualBox

```

# cd /etc/yum.repos.d/
# wget http://download.virtualbox.org/virtualbox/rpm/rhel/virtualbox.repo

```

Теперь мы должны видеть VirtualBox в списке доступного для установки

```

# yum search virtualbox

===== N/S Matched: virtualbox
=====
VirtualBox-3.2.x86_64 : Oracle VM VirtualBox
VirtualBox-4.0.x86_64 : Oracle VM VirtualBox
VirtualBox-4.1.x86_64 : Oracle VM VirtualBox

```

Устанавливаем стабильную ветку 4.1

```

# yum install VirtualBox-4.1

```

В процессе установки вы должны будете увидеть следующее сообщение Creating group

'vboxusers'. VM users must be member of that group!

```
No precompiled module for this kernel found -- trying to build one. Messages emitted during module compilation will be logged to /var/log/vbox-install.log.
```

```
Stopping VirtualBox kernel modules [ OK ]
Uninstalling old VirtualBox DKMS kernel modules [ OK ]
Trying to register the VirtualBox kernel modules using DKMS [FAILED]
(Failed, trying without DKMS)
Recompiling VirtualBox kernel modules [ OK ]
Starting VirtualBox kernel modules [ OK ]
Verifying : VirtualBox-4.1-4.1.18_78361_rhel6-1.x86_64
1/1
```

```
Installed:
VirtualBox-4.1.x86_64 0:4.1.18_78361_rhel6-1
```

```
Complete!
```

Проверяем статус

```
# service vboxdrv status
VirtualBox kernel modules (vboxdrv, vboxnetflt, vboxnetadp, vboxpci) are loaded.
```

```
# lsmod | grep vbox
vboxpci          14160  0
vboxnetadp      18320  0
vboxnetflt     16681  0
vboxdrv        1869894  3 vboxpci,vboxnetadp,vboxnetflt
```

Для того, чтобы у нас работал веб-интерфейс управления virtualbox нам необходимо запустить службу vboxweb-service. Для этого создаем пользователя, от которого будет работать vbox-web-service и наш виртуальный хост.

```
# useradd -g vboxusers -d /vmi/ vbox
# passwd vbox
New password: *****
Retype new password: *****

# chown -R vbox:vboxusers /vmi/
```

После этого создаем два конфигурационных файла - **/etc/default/virtualbox** и **/etc/vbox/vbox.cfg**, необходимых для запуска vbox-web-service. В первом файле мы задаем имя пользователя, от имени которого будет работать сервис vbox-web-service, без этого файла vbox-web-service не будет запускаться.

```
# touch /etc/default/virtualbox
# echo "VBOXWEB_USER=vbox" > /etc/default/virtualbox
```

Второй файл так сказать общего назначения и содержит общие настройки для virtualbox. Мы укажем в нем путь к log файлу и путь установки virtualbox.

```
# touch /etc/default/virtualbox
# touch /etc/vbox/vbox.cfg
# echo "VBOXWEB_LOGFILE=/var/log/vboxweb.log" > /etc/vbox/vbox.cfg
# touch /var/log/vboxweb.log
# chown vbox:vboxusers /var/log/vboxweb.log
# chmod 640 /var/log/vboxweb.log
```

Запускаем сам сервис.

```
# service vboxweb-service start
Starting VirtualBox web service [ OK ]
```

Проверяем log файл

```
# cat /var/log/vboxweb.log
VirtualBox web service 4.1.18 r78361 linux.amd64 (Jun 20 2012 13:11:25)
release log
00:00:00.000 main      Log opened 2012-08-02T10:39:49.385736000Z
00:00:00.000 main      OS Product: Linux
00:00:00.000 main      OS Release: 2.6.32-279.2.1.el6.x86_64
00:00:00.000 main      OS Version: #1 SMP Fri Jul 20 01:55:29 UTC 2012
00:00:00.000 main      OS Service Pack: #1 SMP Fri Jul 20 01:55:29 UTC 2012
00:00:00.000 main      Executable: /usr/lib/virtualbox/vboxwebsrv
00:00:00.000 main      Process ID: 22484
00:00:00.000 main      Package type: LINUX_64BITS_RHEL_6
```

Последний шаг, который нам остался - установить extension pack

```
# wget
http://download.virtualbox.org/virtualbox/4.1.18/Oracle_VM_VirtualBox_Extension_Pack-4.1.18-78361.vbox-extpack
# VBoxManage extpack install
Oracle_VM_VirtualBox_Extension_Pack-4.1.18-78361.vbox-extpack
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Successfully installed "Oracle VM VirtualBox Extension Pack".
```

На этом установку VirtualBox можно считать завершенной. Теперь переходим непосредственно к настройке.

Установка phpvirtualbox

Я думаю из названия самого приложения понятно, что оно написано на php, а соответственно нам надо установить и настроить apache и php. В принципе вам никто не мешает использовать и nginx + php-fpm, но это уже совсем другая история. mpm-itk в данном примере мы будем

использовать в целях безопасности, так как он позволит запускать наш виртуальных хост от определенного нами пользователя и группы, задаваемых в параметре AssignUserID.

```
# yum install httpd httpd-itk php php-soap
```

Для того, чтобы включить `mpm-itk`, необходимо добавить следующие строки в файл **`/etc/sysconfig/httpd`**.

```
HTTPD=/usr/sbin/httpd.itk
umask 007
```

Для поддержки `php` добавляем следующие строчки в **`/etc/httpd/conf.d/php.conf`**.

```
<IfModule itk.c>
  LoadModule php5_module modules/libphp5.so
</IfModule>
```

Производим минимальную настройку `apache`, для этого редактируем файл **`/etc/httpd/conf/httpd.conf`**

```
# /etc/httpd/conf/httpd.conf

ServerTokens Prod
KeepAlive On

<IfModule itk.c>
  StartServers      8
  MinSpareServers   5
  MaxSpareServers   20
  ServerLimit       256
  MaxClients        256
  MaxRequestsPerChild 4000
</IfModule>

ServerAdmin root@example.net
ServerName vb.example.net:80

NameVirtualHost *:80
```

Все остальные параметры оставляем без изменений. Вы можете найти в сети интернет большое количество статей по настройке `apache`, так что здесь мы не будем вдаваться в подробности.

Создаем конфигурационный файл **`/etc/httpd/conf.d/phpvb.example.net.conf`** для нашего виртуального хоста

```
# /etc/httpd/conf.d/phpvb.example.net.conf
```

```
<VirtualHost *:80>
  ServerName phpvb.example.net
  DocumentRoot /var/www/vhosts/phpvb.example.net

  AssignUserID vbox vboxusers

  php_admin_value open_basedir /var/www/vhosts/phpvb.example.net/:/tmp/
  php_admin_value session.save_path /tmp
  php_admin_value upload_tmp_dir /tmp

  <Directory /var/www/vhosts/phpvb.example.net>
    AuthType Basic
    AuthName "VirtualBox administration area. Authentication required!"
    AuthUserFile /var/www/vhosts/phpvb.example.net/.htpasswd
    Require valid-user

    Options -Indexes
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

Скачиваем последнюю версию [phpvirtualbox](#), на момент написания статьи это была версия 4.1.7, и распаковываем содержимое архива в [/var/www/vhosts/phpvb.example.net](#). Так же не забываем изменить владельца файлов и выставить соответствующие права.

```
# mkdir -p /var/www/vhosts/phpvb.example.net/
# chown -R vbox:vboxusers /var/www/vhosts/phpvb.example.net
# chmod -R 770 /var/www/vhosts/phpvb.example.net/
```

Создаем файл с паролем, который будет использовать апачем при аутентификации

```
# htpasswd -c /var/www/vhosts/phpvb.example.net/.htpasswd admin
New password: *****
Re-type new password: *****
Adding password for user admin
```

Запускаем apache и добавляем его запуск при старте системы

```
# chkconfig --level 35 httpd on

# service httpd start
Starting httpd: [ OK ]
```

Все, что нам осталось сделать, перед тем, как мы начнем использовать [phpvirtualbox](#) - это настроить его основной конфигурационный файл.

```
# cd /var/www/vhosts/phpvb.example.net/
# cp config.php-example config.php
```

```
# chown -R root:vboxusers config.php
```

Вносим минимально необходимые изменения в файл config.php

```
<?php
class phpVBoxConfig {

// Имя пользователя и пароль, от имени которого работает vboxweb-service
var $username = 'vbox';
var $password = '1234567';

// Адрес SOAP интерфейса vboxweb-syrvice
var $location = 'http://127.0.0.1:18083/';

// Язык интерфейса по умолчанию
var $language = 'en';

// Отключаем аутентификацию и используем аутентификацию apache
var $noAuth = true;

// ip адрес, используемый для консольных подключений
var $consoleHost = '192.168.207.170';

// Отключить окно предпросмотра
var $noPreview = false;

// Интервал обновления изображения в окне предпросмотра
var $previewUpdateInterval = 30;

// Ширина окна предпросмотра
var $previewWidth = 180;

// Соотношение сторон в окне предпросмотра
var $previewAspectRatio = 1.6;

// Включить пользовательские иконки виртуальных машин
var $enableCustomIcons = true;

// Запрашивать подтверждение при удалении дисков из Virtual Media Manager.
var $deleteOnRemove = true;

// Позволяем пользователю выбирать только следующие типы файлов
var $browserRestrictFiles =
array('.iso', '.vdi', '.vmdk', '.img', '.bin', '.vhd', '.hdd', '.ovf', '.ova', '.xml',
'.vbox', '.cdr', '.dmg', '.ima', '.dsk', '.vfd');

// Ограничиваем пользователя при выборе места для виртуальных машин.
// С учетом использования open_basedir особого смысла не имеет.
var $browserRestrictFolders = array('/vmi');
```

```
// Принудительное использование локального обозревателя файлов, вместо обозрения,
предоставляемого vboxwebsrv
var $browserLocal = true;

// Отключение просмотра файлов/папок
var $browserDisable = true;

// Отключение опознавания дисков windows
var $noWindowsDriveList = true;

// Интервал обновления использования памяти
var $hostMemInfoRefreshInterval = 5;

// Отображать % свободной памяти, а не используемой
var $hostMemInfoShowFreePct = true;

/* При старте VM через phpvirtualbox сначала проверяется наличие свободной памяти
* для запуска соответствующей VM (RAM + video + 50 Mb). Если памяти недостаточно,
то
* появится диалоговое окно с запросом на подтверждения запуска VM.
*/
var $vmMemoryStartLimitWarn = true;
var $vmMemoryOffset = 100;

// Отображать версию гостевых дополнений на вкладке Details соответствующей VM
var $enableGuestAdditionsVersionDisplay = true;

// Скрыть вкладку Snapshots
var $disableTabVMSnapshots = false;

// Скрыть вкладку Console
var $disableTabVMConsole = false;

// Разрешение экрана на вкладке Console, которое будет доступно для выбора в
выпадающем комбобоксе.
var $consoleResolutions =
array('640x480', '800x600', '1024x768', '1280x720', '1440x900');

// Раскладка клавиатуры, доступная на вкладке Console. На данный момент
поддерживается только EN и DE
var $consoleKeyboardLayout = 'EN';

// Максимальное количество сетевых карт, доступных для одной виртуальной машины
var $nicMax = 4;

/* Включаем отображение дополнительных опций в интерфейсе phpvirtualbox:
- HPET(High Precision Event Timer)
- Host time sync
- Acceleration
- VRDE Net Address
- Additional NAT Settings
```


- Virtual Media Manager additional buttons

Данные опции не доступны в VirtualBox GUI

*/

```
var $enableAdvancedConfig = true;
```

```
// По какому признаку сортируется список виртуальных машин
```

```
var $vmListSort = 'name';
```

```
// Библиотека аутентификации
```

```
var $authLib = 'Builtin';
```

```
// VM ownership
```

```
#var $enforceVMOwnership = true;
```

```
// Per-user VM quota
```

```
#var $vmQuotaPerUser = 2;
```

```
// Позволяет настраивать опции сети VDE (Virtual Distributed Ethernet).
```

```
var $enableVDE = false;
```

```
// Refresh VM cache when VM Settings window is loaded. Default is true. Set to false to disable.
```

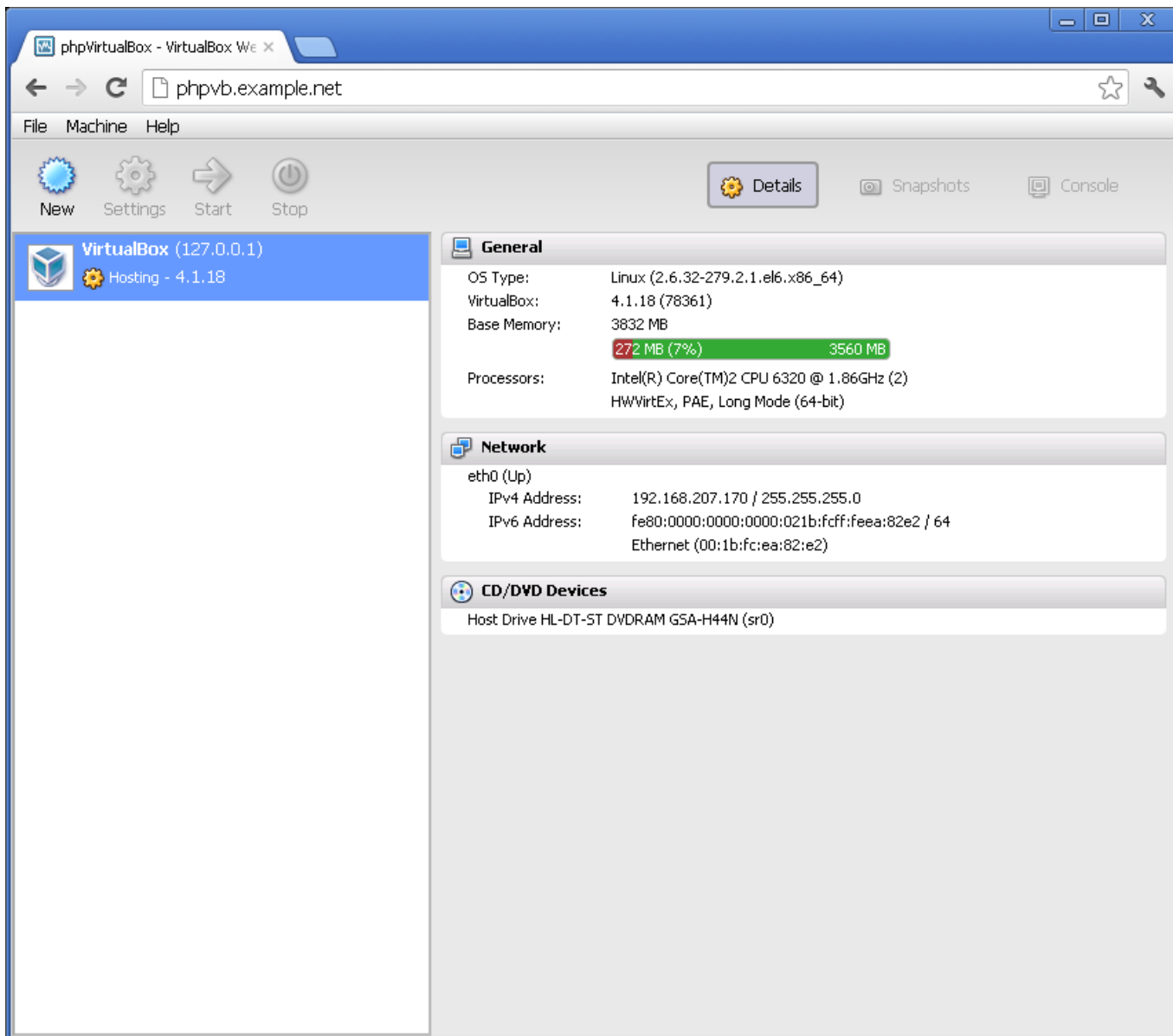
```
var $vmConfigRefresh = true;
```

```
// Path
```

```
#var $cachePath = '/tmp';
```

```
}
```

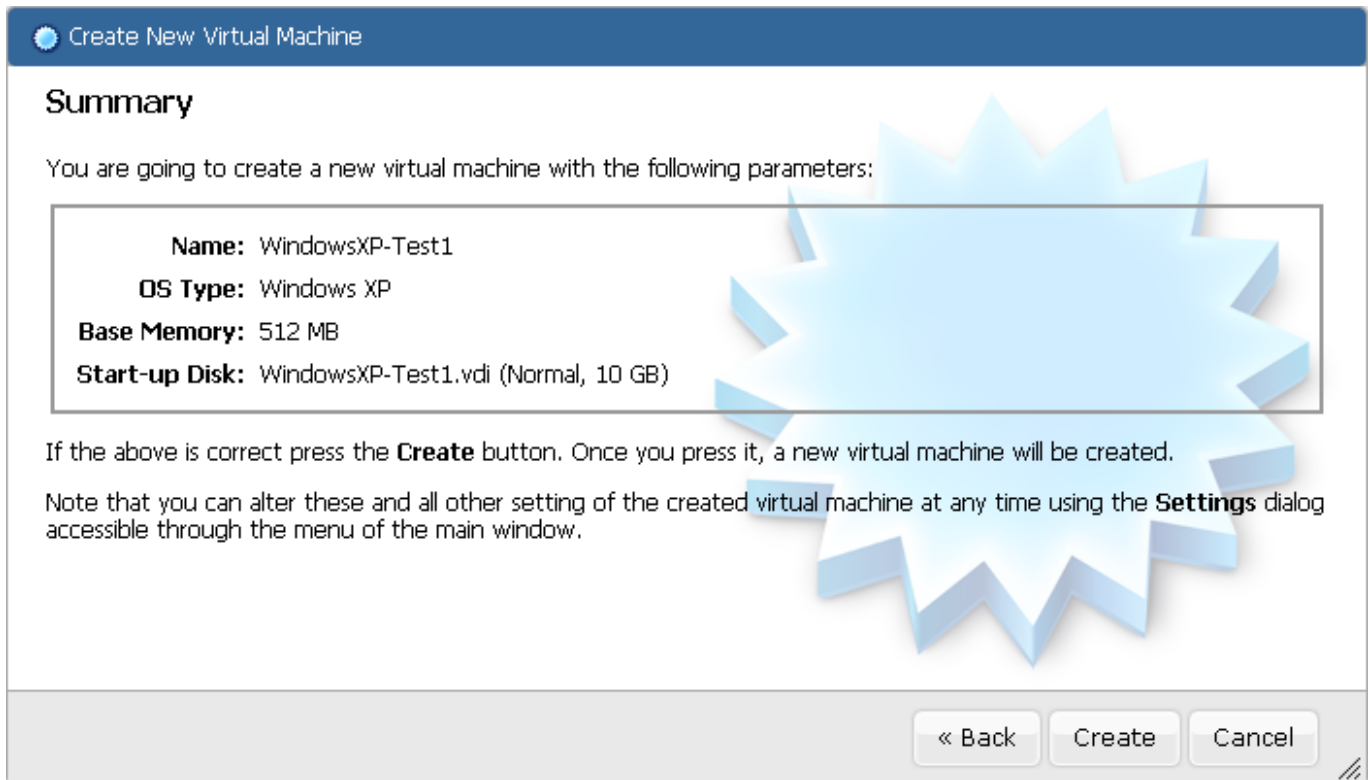
Теперь мы можем в любимом интернет-проводнике открыть страничку и увидеть примерно следующее



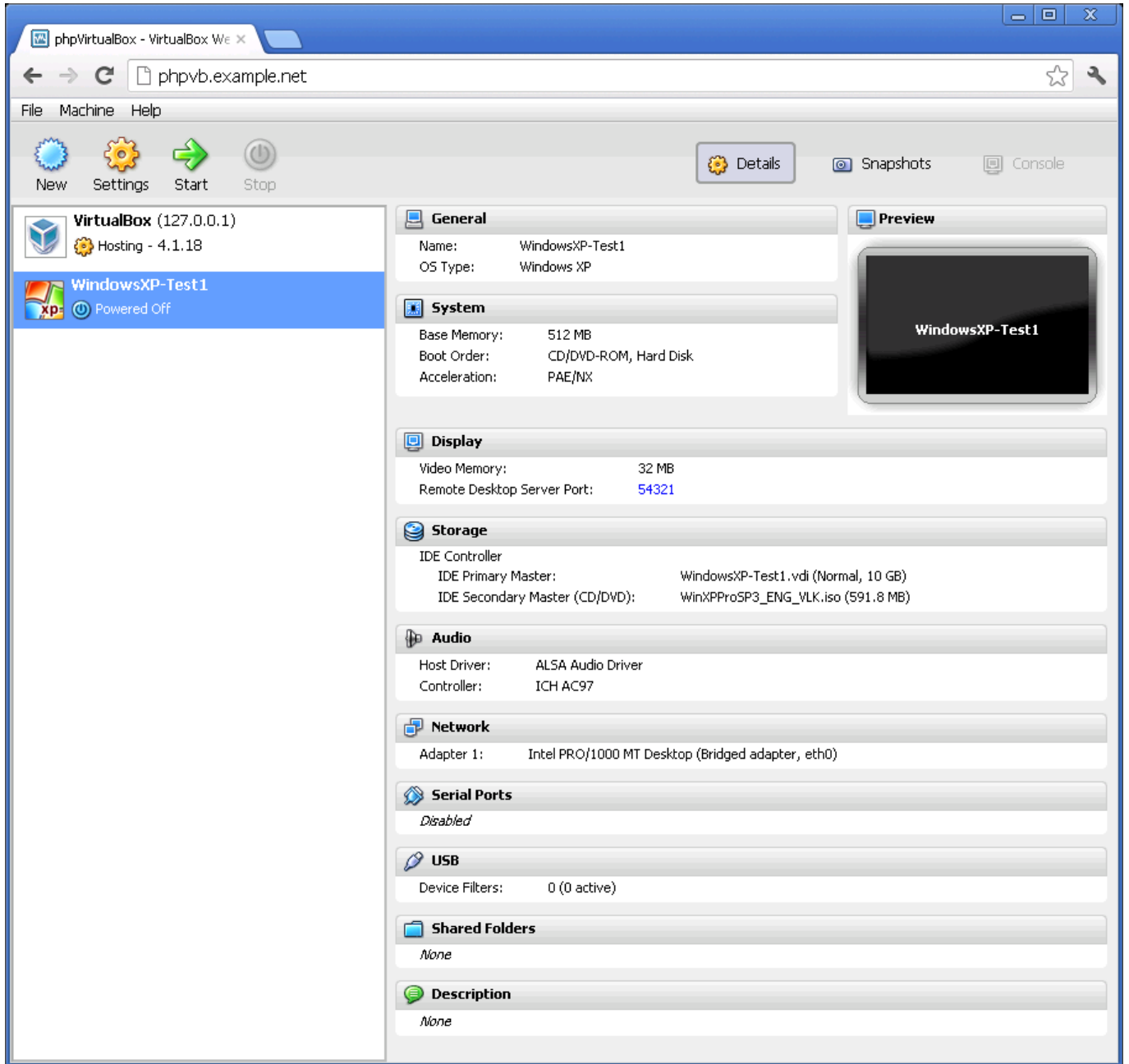
Так как у меня под образы виртуальных машин выделен отдельный раздел - /dev/sda5, то в настройках (**File → Preferences → General → Default Machine Folder**) указываем к нему путь.

```
# mount | grep sda
/dev/sda2 on / type ext4 (rw)
/dev/sda1 on /boot type ext3 (rw)
/dev/sda5 on /vmi type xfs (rw)
```

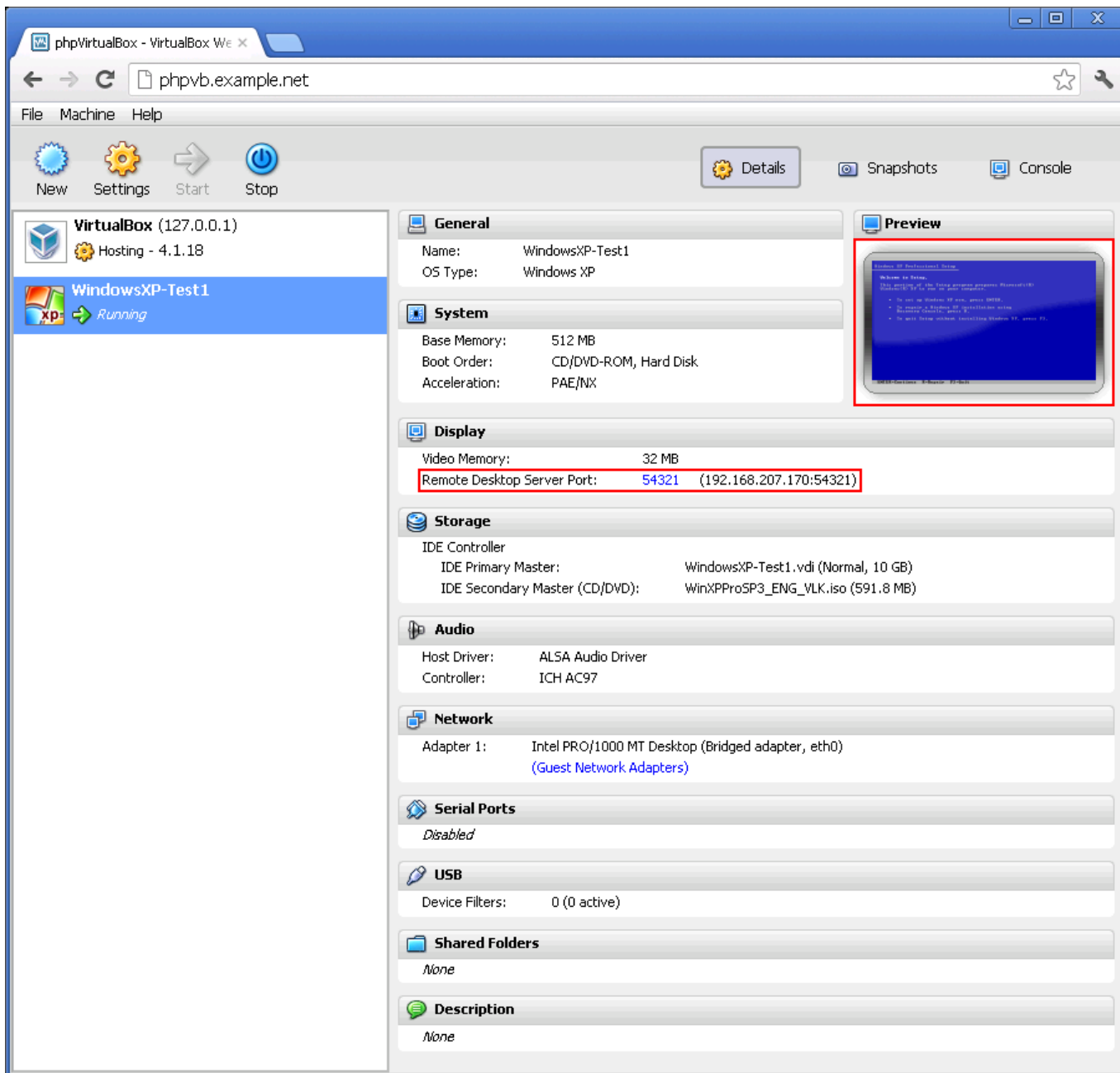
Теперь мы можем создать новую виртуальную машину. Сам процесс очень просто, поэтому подробно описывать его не буду. Приведу лишь экран с финальными настройками виртуальной машины



В итоге, после небольших настроек виртуальной машины у нас должны получиться примерно следующие параметры



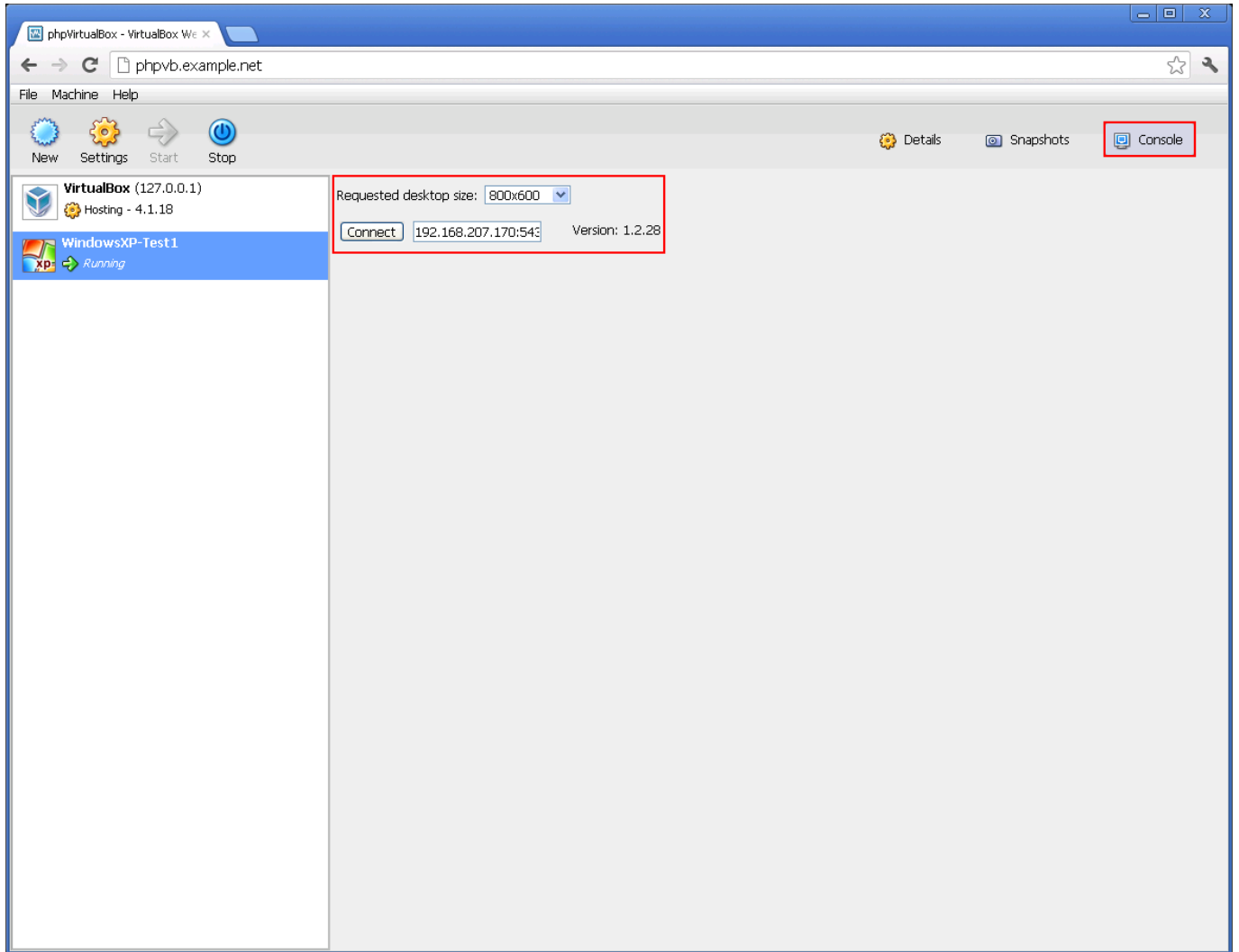
Собственно у нас все готово к запуску виртуальной машины, что мы и делаем.

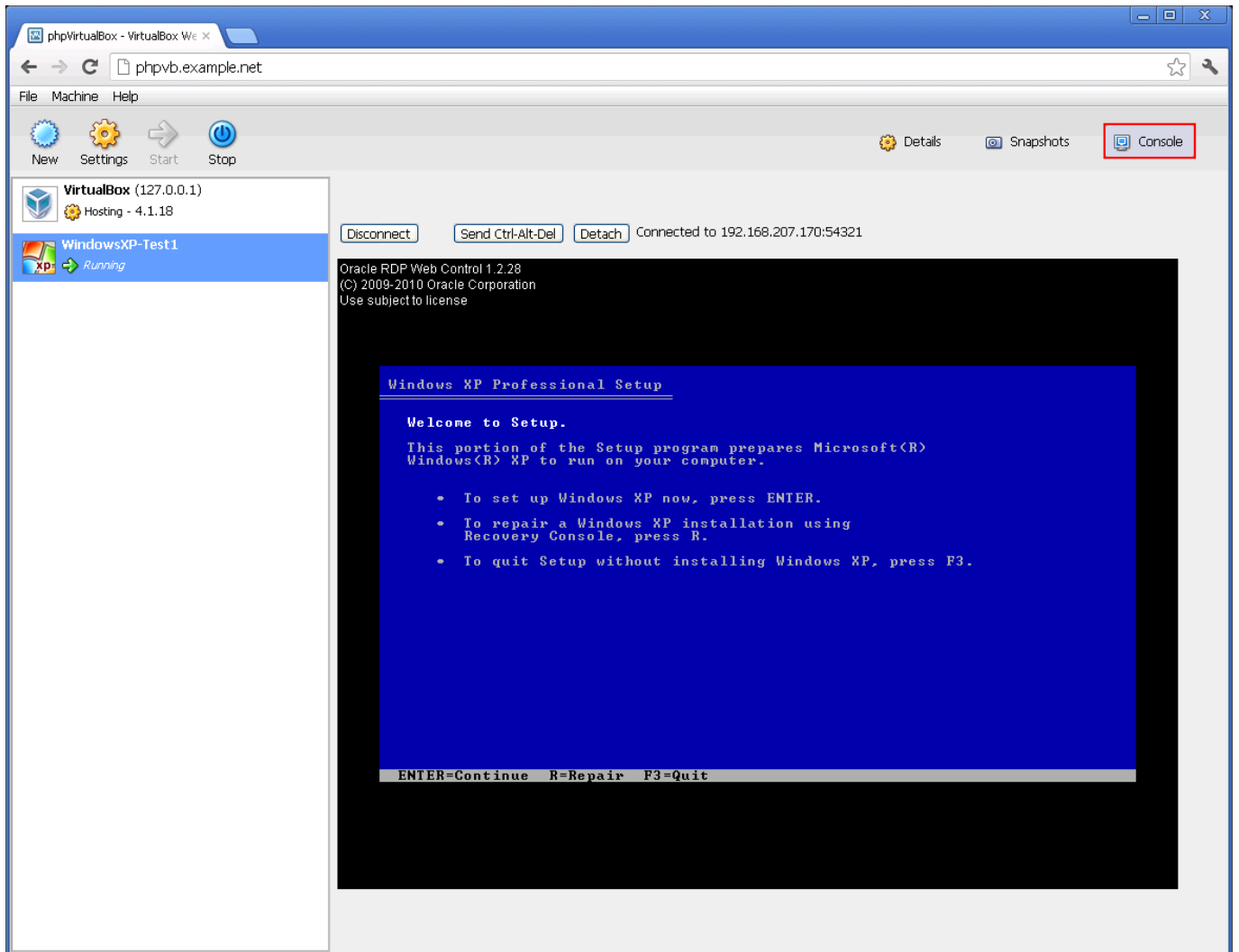


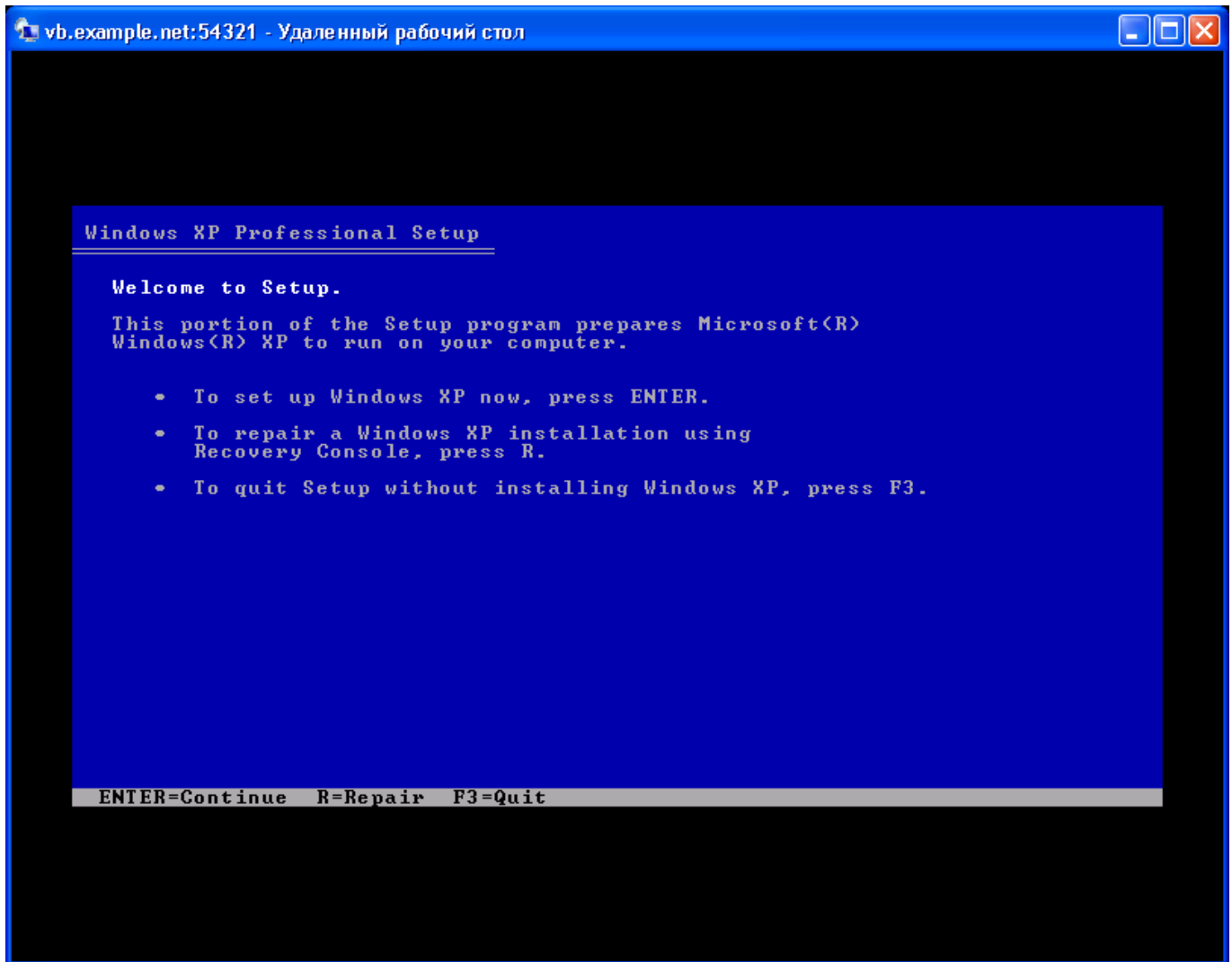
Теперь для продолжения установки гостевой ОС вы можете воспользоваться 3мя способами:

- использовать VRDE (VirtualBox Remote Desktop Extension), что дает возможность подключиться к гостевой ос с помощью rdp клиента
- использовать встроенный rdp клиент в phpvirtualbox, который доступен на вкладке **Console**
- использовать X11 Forwarding

Следует обратить внимание, что первых два пункта будут доступны только после установки [Extension Pack](#). Так же стоит учитывать, что Extension Pack доступен под лицензией [PUEL](#), которая разрешает использовать их бесплатно только в некоммерческих целях!

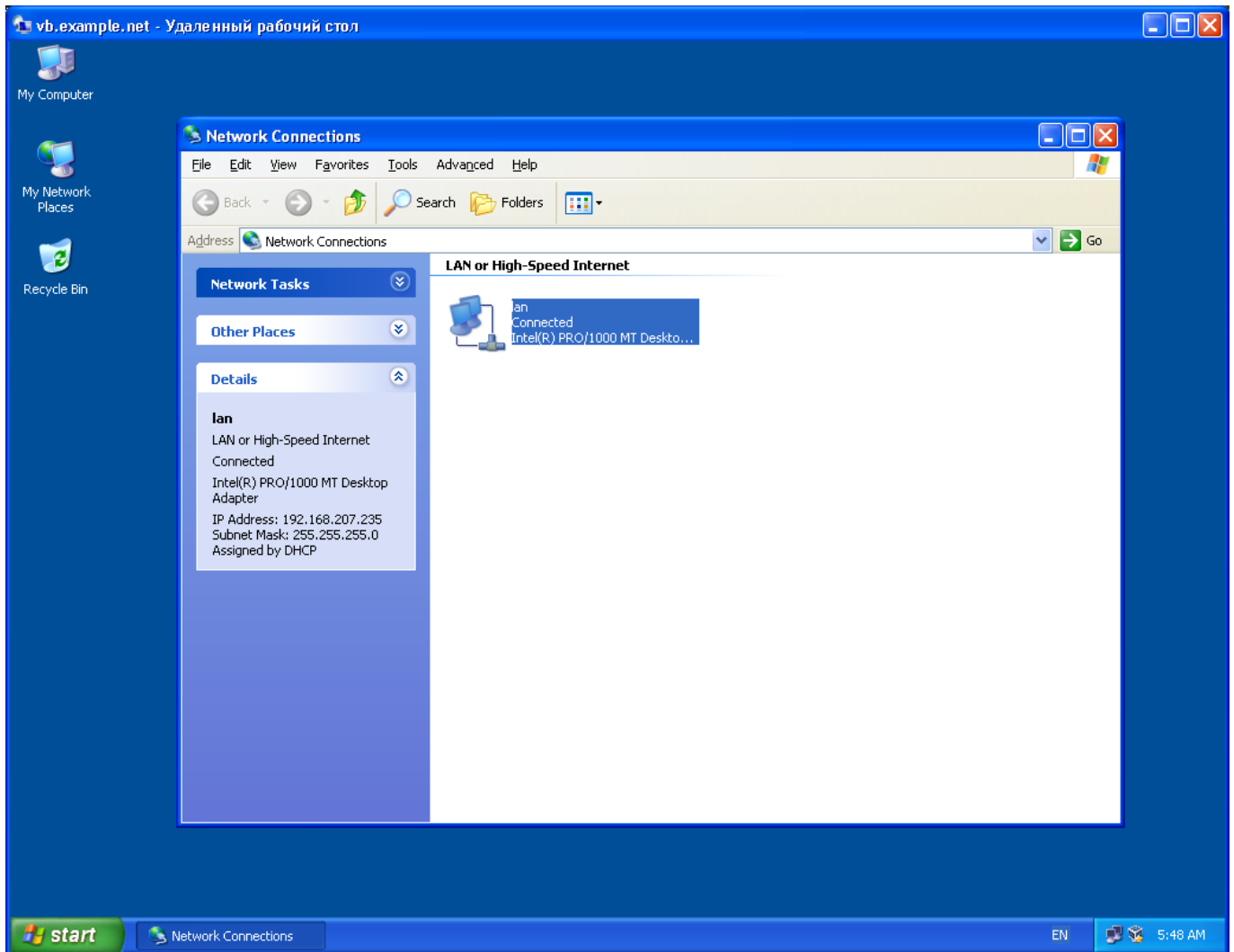






Не забываем, что по умолчанию Windows XP не установит драйвера для сетевой карты Intel 1000 MT, так что вам самим придется скачать их с официального сайта intel и установить внутри гостевой ОС.

Производим установку ОС, после чего не забываем установить Guest additions. Для этого достаточно смонтировать iso образ **[/usr/share/virtualbox/VBoxGuestAdditions.iso](#)** в виртуальную машину и в зависимости от типа установленной ОС произвести установку дополнений.



VirtualBox X11 Forwarding

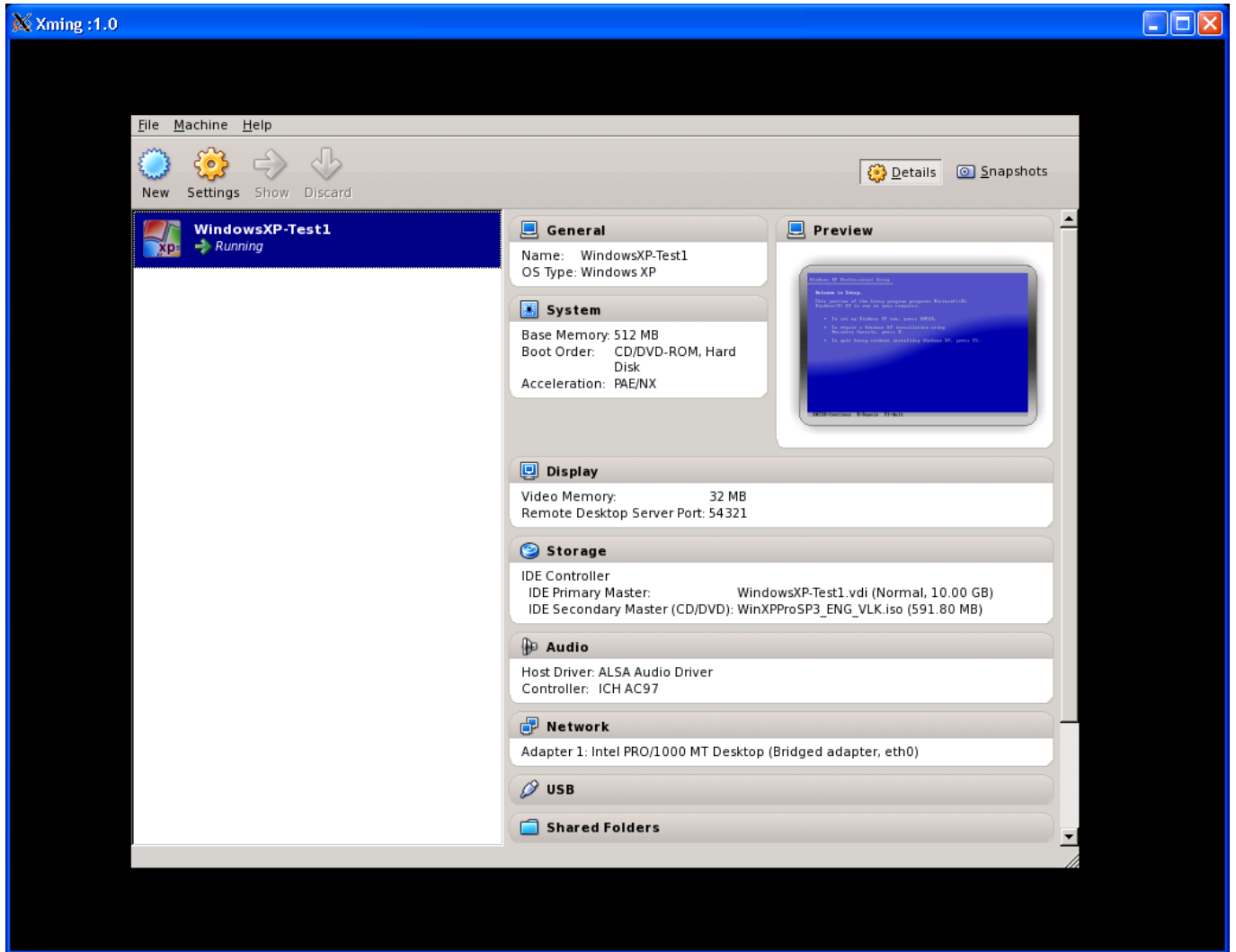
Иногда возникает необходимость получить так сказать непосредственный доступ к консоли виртуальной машины, особенно в случае сбоев и внештатных ситуаций или rdp/vrde сервисы работают нестабильно, бывает и такое. В таком случае самым удобным способом будет использование [X11 Forwarding](#). Для CentOS-6.x устанавливаем следующие пакеты

```
# yum install xorg-x11-xauth dejavu-sans-fonts
```

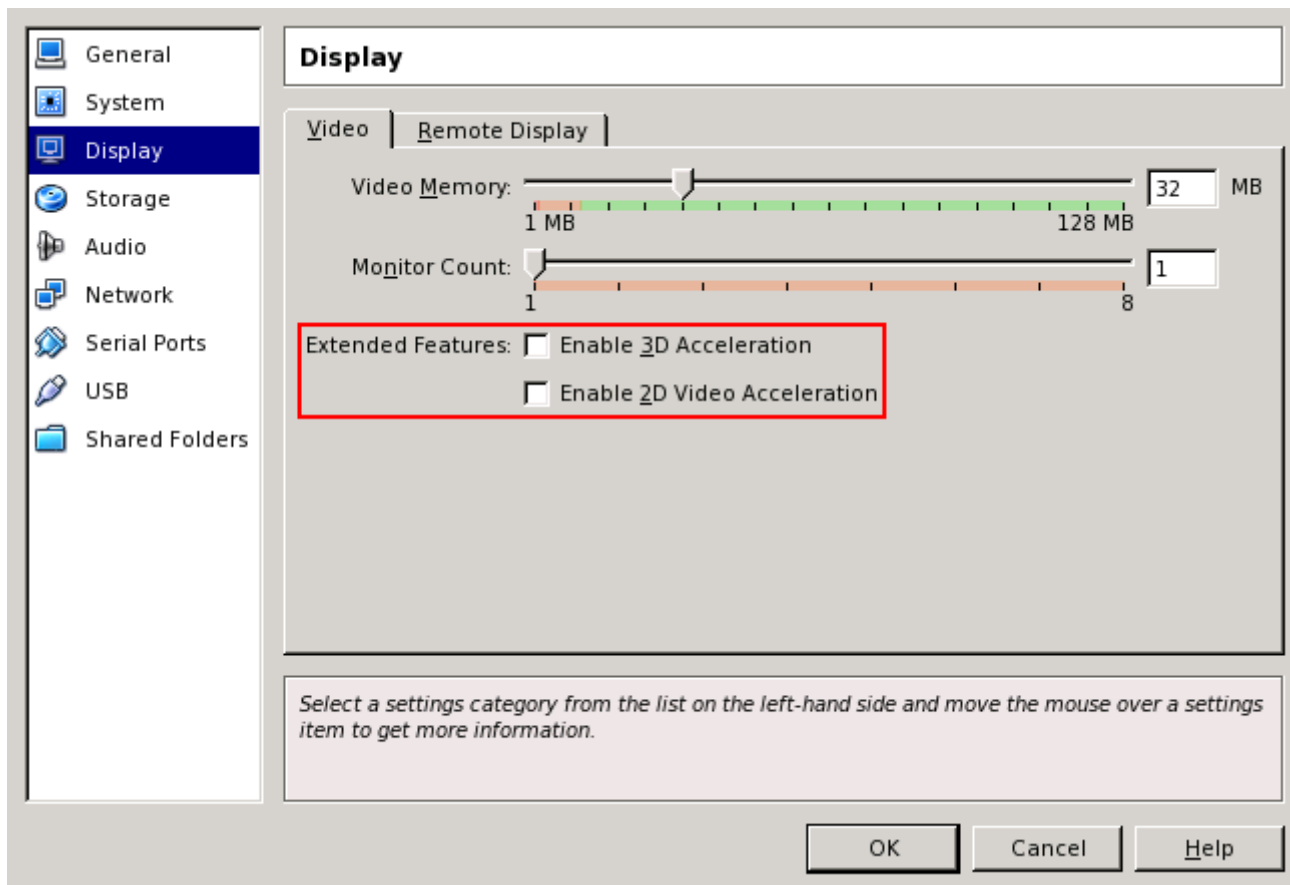
Запускаем у себя X сервер и в putty настраиваем перенаправление X11. Собственно теперь можно запустить и сам VirtualBox. Заходим под пользователем vbox и запускаем VirtualBox

```
$ VirtualBox
```

В итоге мы должны увидеть следующее окно



Например, здесь вы можете включить поддержку ускорения 2D/3D графики, недоступной в вебинтерфейсе.



Хотя в принципе все это можно сделать и из командной строки, как говорится

```
# VBoxManage --help
```

From:
<http://sys-adm.org.ua/> - wiki.sys-adm.org.ua

Permanent link:
<http://sys-adm.org.ua/virtualization/virtualbox>

Last update: **2016/02/11 23:48**

