

راهنمای نصب KVM در ابونتو

یک نرم افزار مجازی این سورس و نسبتا جدید است که در عین حال کار مجازی سازی سخت افزار را با قدرت زیاد انجام میدهد . KVM بدین صورت کار میکند که با قرارگیری در کنار هسته لینوکس به آن قابلیت مجازی سازی میدهد و از تکنولوژی Full Virtualization برای مجازی سازی ماشین های مجازی سازی استفاده میکند KVM . به عنوان یک ماژول هسته لینوکس زمان بارگذاری سیستم عامل اجرا میشود. در این روش هر ماشین مجازی به عنوان یک فرایند منظم در لینوکس محسوب میشود. در این ابزار به راحتی میتوان از لینوکس و ویندوز به عنوان ماشین های مجازی بهره برد.

اگر روی سیستم میخواهید نصب کنید باید در قسمت بایوس قابلیت virtualization را Enable کنید.

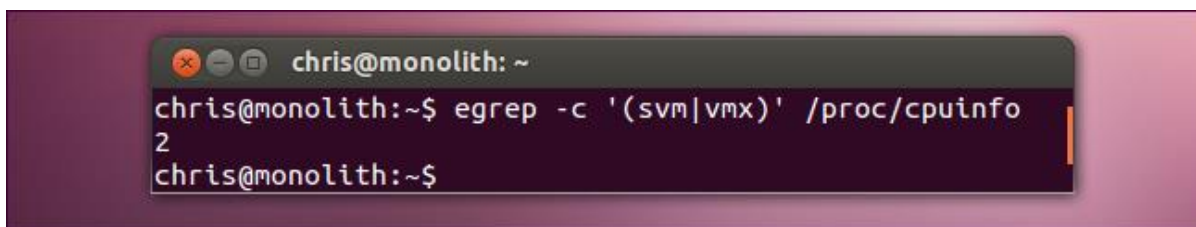
اگر می خواهید در Vmware نصب کنید ابتدا باید گزینه های مربوط به virtualization آن را فعال کنید.

تست اینکه CPU از قابلیت مجازی سازی پشتیبانی میکند یا نه!

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

اگر در خروجی ۰ نمایان شد یعنی از قابلیت مجازی سازی پشتیبانی نمیکند.

اگر در خروجی ۱ یا بیشتر نمایان شد یعنی از مجازی سازی پشتیبانی میکند. علاوه بر این بیانگر ۶۴ بیتی بودن CPU نیز می باشد.

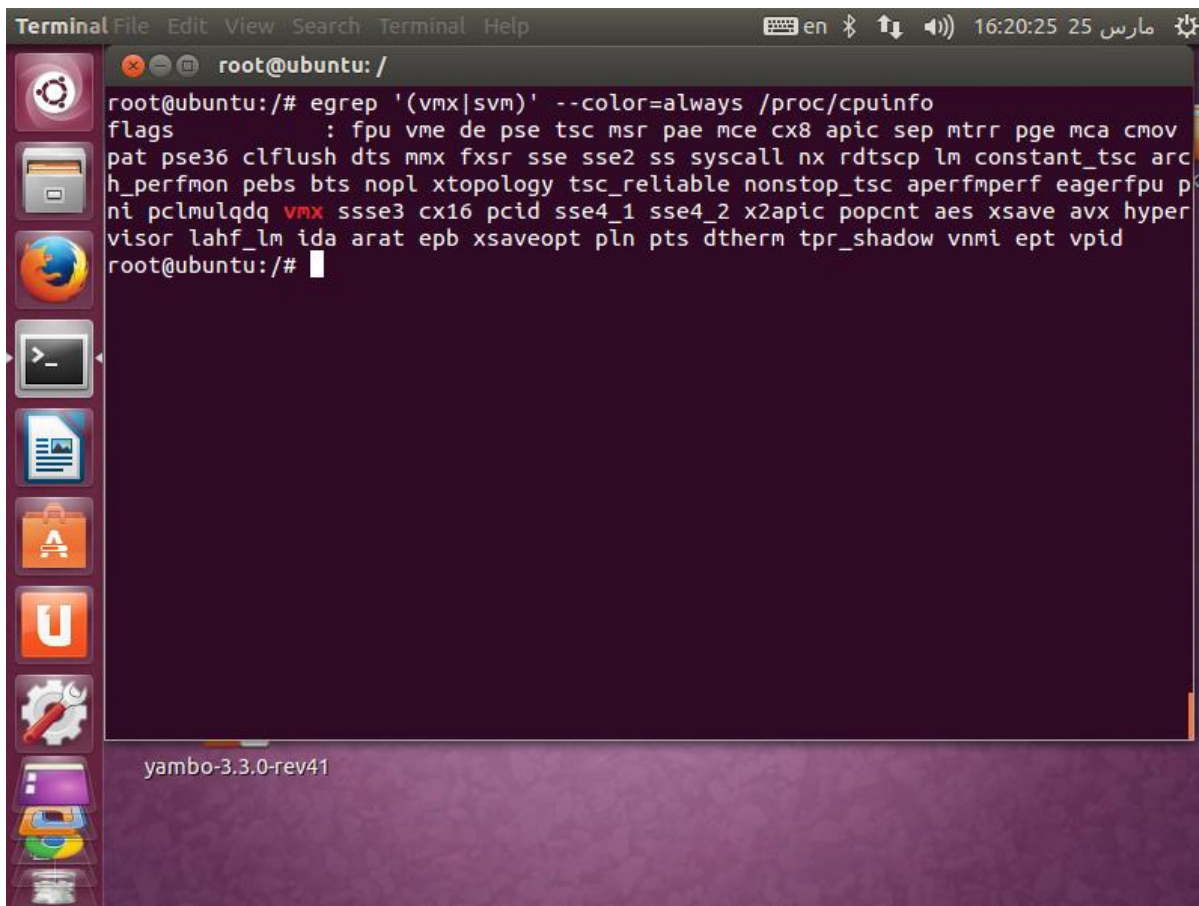


```
chris@monolith: ~  
chris@monolith:~$ egrep -c '(svm|vmx)' /proc/cpuinfo  
2  
chris@monolith:~$
```

یا از دستور زیر استفاده کنید:

```
egrep '(vmx|svm)' --color=always /proc/cpuinfo
```

اگر چیزی نشان نداد یعنی CPU از مجازی سازی پشتیبانی نمی کند و همینجا باید کار را متوقف کنید و اگر خروجی مطابق تصویر زیر نشان داده شد میتوانید ادامه دهید. VMS برای CPU های INTEL می باشد و SVM برای CPU های AMD.



```
Terminal File Edit View Search Terminal Help en 16:20:25 25 مارس
root@ubuntu: /
root@ubuntu:/# egrep '(vmx|svm)' --color=always /proc/cpuinfo
flags      : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts mmx fxsr sse sse2 ss syscall nx rdtscp lm constant_tsc arc
h_perfmon pebs bts nopl xtopology tsc_reliable nonstop_tsc aperfmperf eagerfpu p
ni pclmulqdq vmx ssse3 cx16 pcid sse4_1 sse4_2 x2apic popcnt aes xsave avx hyper
visor lahf_lm ida arat epb xsaveopt pln pts dtherm tpr_shadow vnmi ept vpid
root@ubuntu:/#
```

این دستور تمام نرم افزارهای مورد نیاز و وابستگی های آن را نصب میکند.

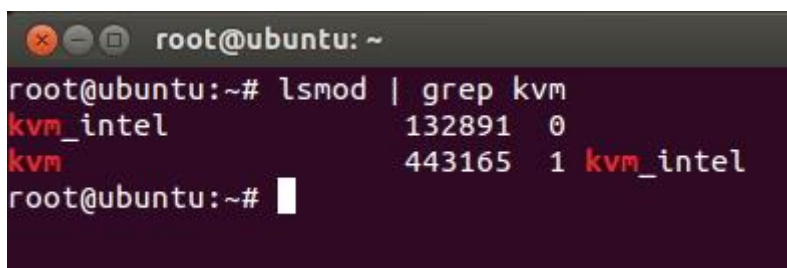
```
apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils kvm virt-
viewer virt-manager virt-top uuid
```

بعد از کامل شدن نصب بهتره تمام نرم افزار ها و سیستم ری استارت بشن، زیرا ممکن است با خطاهای غیرقابل پیش بینی مواجه شوید، مثل به آرامی اجرا شدن ماشین های مجازی.

یکی از ویژگی های KVM این است که دو تا ماژول را هنگام نصب در کنار هسته ایجاد میکند که بعد از ری استارت شدن سیستم به همراه هسته KVM هم اجرا می شود. این ماژول ها را می توان در ترمینال با دستورهای زیر بررسی کرد.

```
lsmod | grep kvm
```

که خروجی آن معمولا به صورت زیر هست:



```
root@ubuntu: ~
root@ubuntu:~# lsmod | grep kvm
kvm_intel          132891  0
kvm                443165  1 kvm_intel
root@ubuntu:~#
```

برای اجرای KVM نیاز هست که سرویس Libvirt-bin در پس زمینه لینوکس اجرا باشد، برای بررسی آن میتوان دستور زیر رو وارد کرد.

```
service libvirt-bin status
```

اگر این سرویس اجرا نبود با دستور زیر آن را اجرا کنید:

```
service libvirt-bin start
```

Libvirt یک API، سرویس (daemon) و ابزار مدیریتی اپن سورس برا مدیریت پلتفرم مجازی سازی است. همچنین میتواند برای مدیریت KVM، Xen، VMware ESX، QEMU و دیگر تکنولوژی های مجازی سازی استفاده شود این API ها به طور گسترده ای در مرتب سازی لایه Hypervisor برای توسعه یک راه حل با پایه Cloud استفاده می شود.

در زمان فرایند نصب یک گروه با نام libvirt ایجاد می شود که نیاز هست تا یوزرها را به آن اضافه کرد تا بتوان به آن دسترسی داشت. در مثال زیر می توان نام یکی از یوزرهایی که عضو گروه libvirt شده را مشاهده کنید، برای این کار لازم است با یوزر مورد نظر به سیستم لاگین کنید و در ترمینال دستور id را وارد کنید.

```
ahmad@ubuntu: ~  
ahmad@ubuntu:~$ id  
uid=1001(ahmad) gid=1001(ahmad) groups=1001(ahmad),126(libvirt)  
ahmad@ubuntu:~$
```

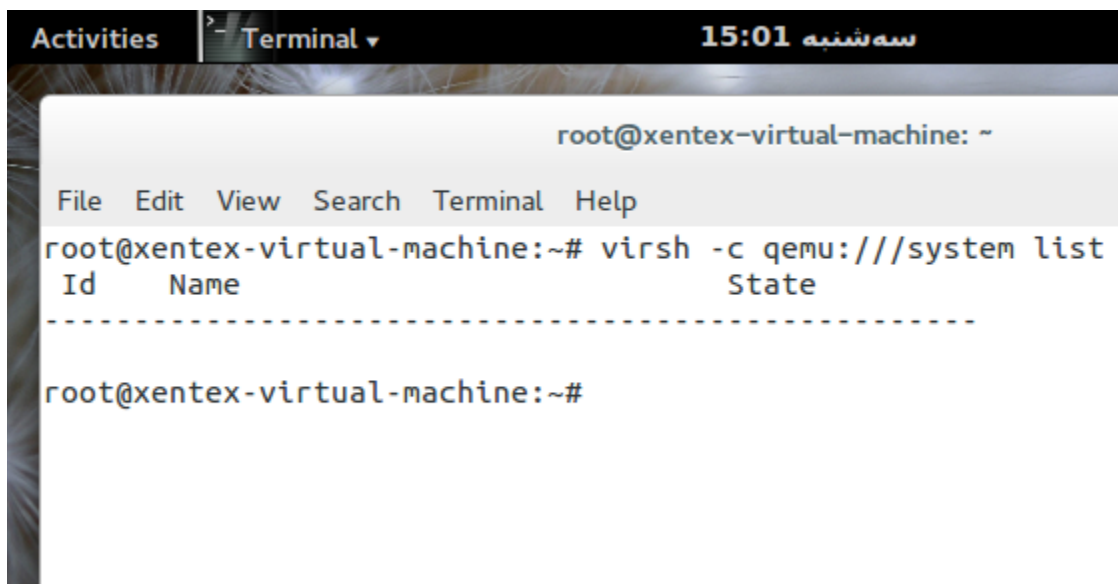
برای انجام اینکار می توان به ترتیب زیر عمل کرد:

در مثال زیر یوزری که قبلا به نام ahmad ایجاد کرده بودیم را عضو گروه libvirt می کنیم.

```
adduser ahmad libvirt
```

بعد از اجرای این دستور، از لینوکس log off کنید و دوباره با یوزر احمد log in کنید. اجرای این دستور بعد از لاگین کردن باید لیست خالی از ماشین های مجازی را مشاهده کنید. این نشان می دهد که همه چیز به درستی کار می کند.

```
Virsh -c qemu:///system list
```



```
root@xentex-virtual-machine: ~  
File Edit View Search Terminal Help  
root@xentex-virtual-machine:~# virsh -c qemu:///system list  
Id      Name  
-----  
root@xentex-virtual-machine:~#
```

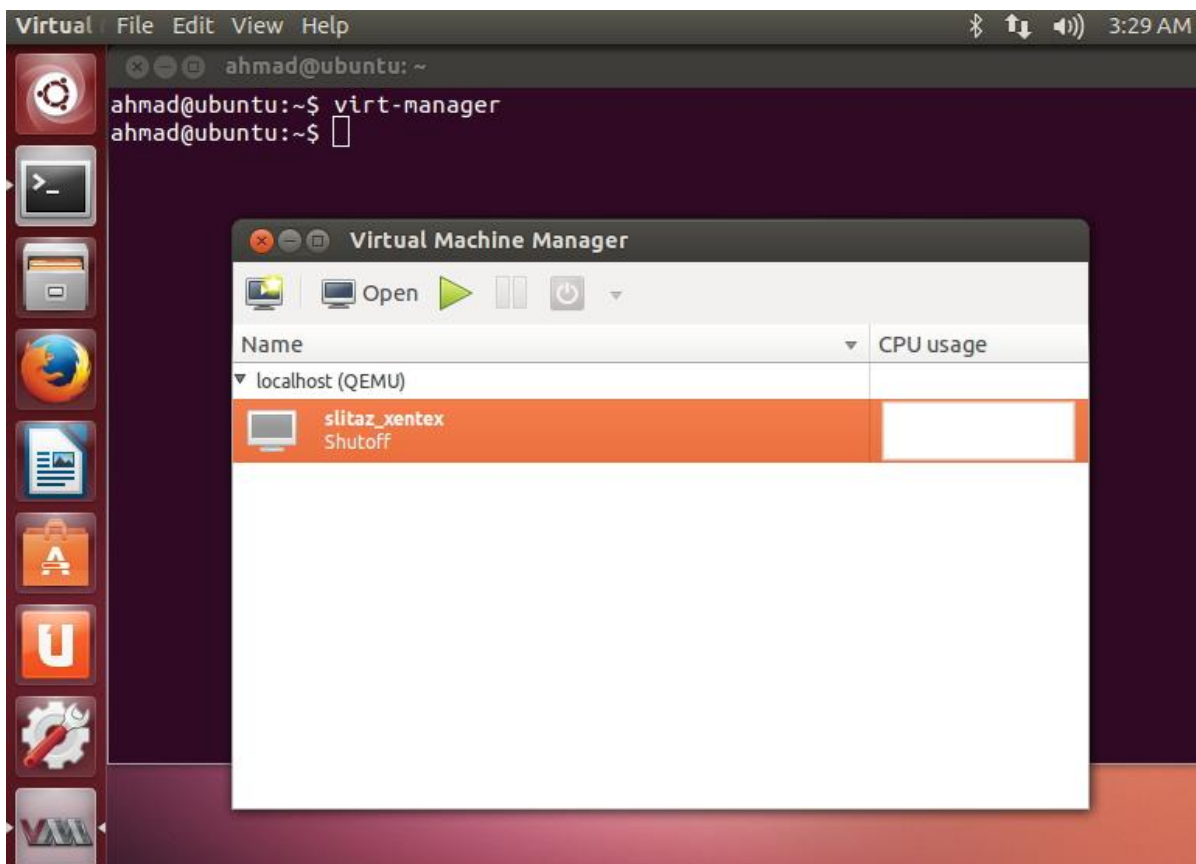
سیستم های مجازی رو به راحتی می توان کانفیگ کرد، از دو ابزار که یکی از طریق خط فرمان با نام virt-install و دیگری محیط گرافیکی با نام virt-manager را ارائه می دهد.

برای استفاده از ابزار گرافیکی میتوان از دو روش استفاده کرد.

در قست Dash عبارت virtual machine manager را تایپ کرد و بر روی آیکن کلید کرده تا پنجره آن باز شود.



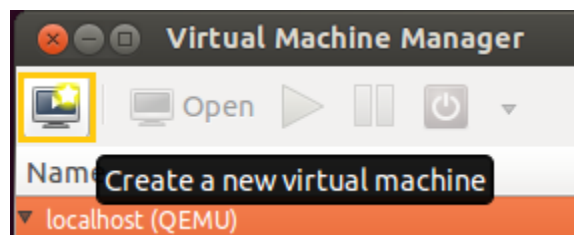
روش دوم در ترمینال عبارت vit-manager را تایپ کرده تا مطابق شکل زیر پنجره آن ظاهر شود.



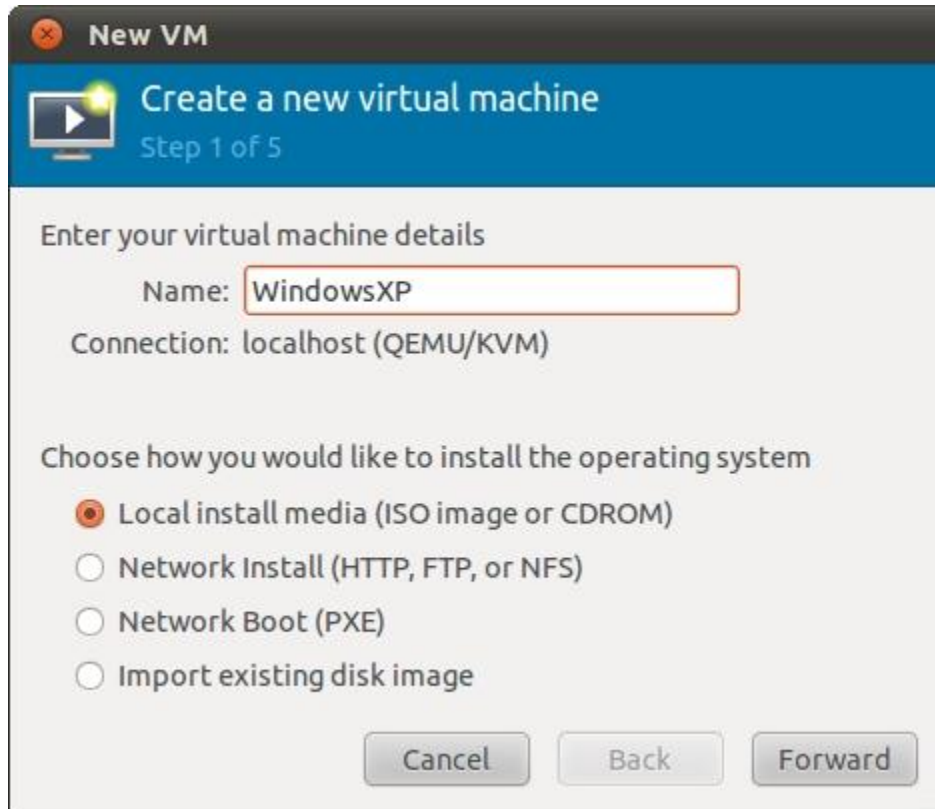
در پنجره باز شده لیستی از ماشین های مجازی موجود در این سیستم را نمایش می دهد. در تصویر بالا یک ماشین مجازی که قبلا ایجاد کرده ایم را مشاهده می کنید.

قبل از اینکه بر روی دکمه connect برای ایجاد اتصال کلیک کنید، ابتدا اطمینان حاصل کنید که برای QEMU/KVM هایپروایزر تنظیم شده باشد. همچنین توجه داشته باشید که Virt-manager نیز برای مدیریت مجازی سازی از طریق ریموت شدن به سیستم های دیگر هم استفاده کرد.

برای ایجاد یک ماشین مجازی جدید روی دکمه create a new virtual machine کلیک کنید که در بالا و سمت چپ پنجره virtual machine manager وجود دارد.

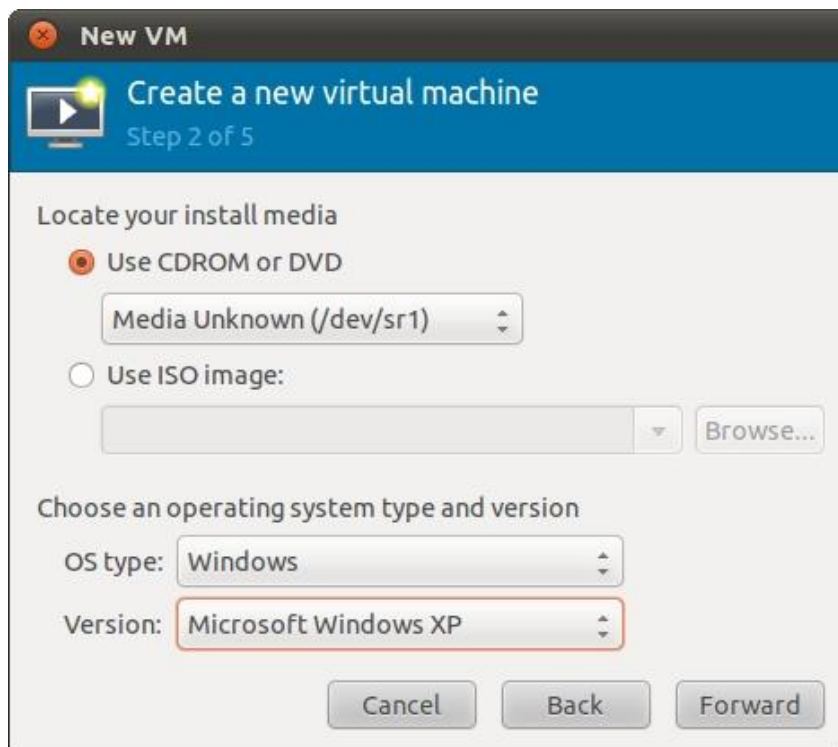


در پنجره باز شده نامی که می خواهید به ماشین مجازی اختصاص دهید را مشخص نمایید و روشی که می خواهید از فایل نصب استفاده کنید را مشخص نمایید.

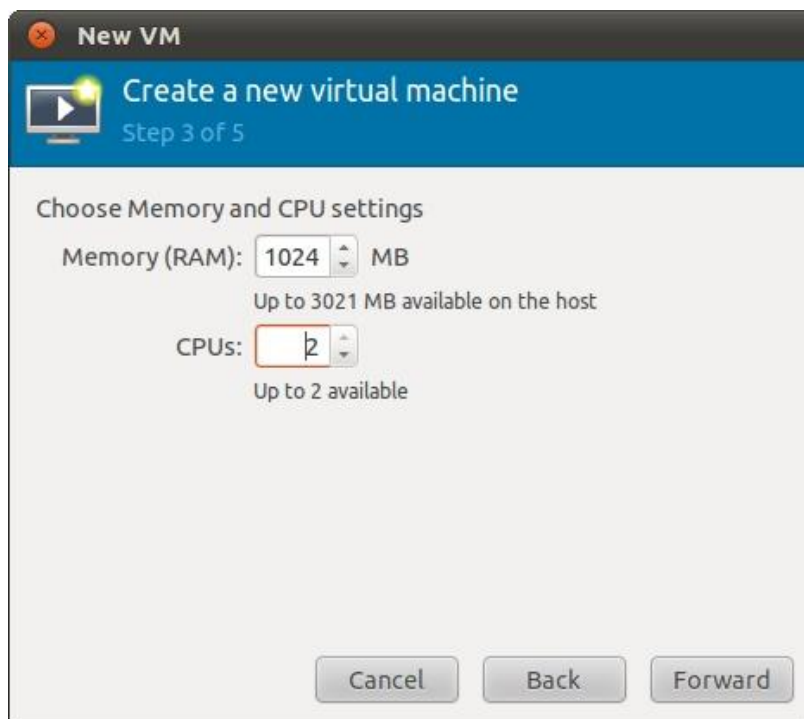


توجه داشته باشید که در زیر فیلد Name: نوع ارتباط نشان داده می شود. (در این تصویر QEMU/KVM می باشد). اگر ارتباط روی سیستم شما QEMU را نمایش می دهد، سیستمتان از قابلیت مجازی سازی INTEL-VT یا AMD-V پشتیبانی نمی کند، یا این ویژگی در قسمت بایوس سیستم غیر فعال می باشد. با شروع این مورد ماشین های مجازی درون شبیه ساز QEMU اجرا می شوند و از مجازی سازی که مبتنی بر سرعت بیشتر سخت افزار که توسط KVM ارائه می شود استفاده نمی کند.

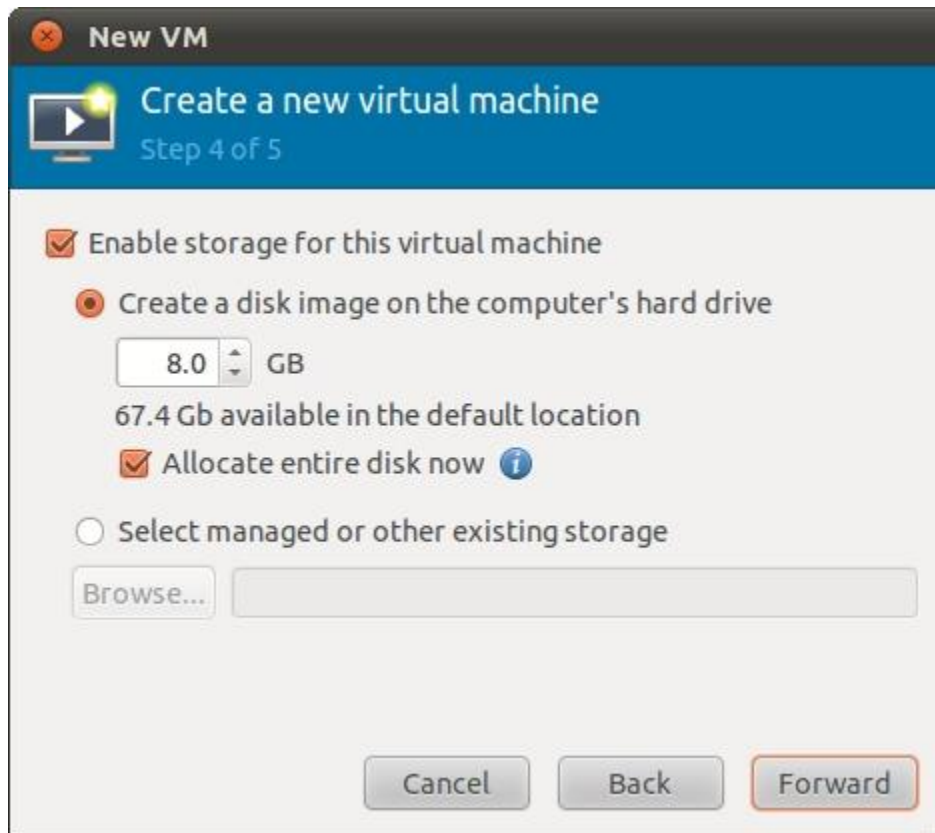
در این قسمت محلی که از آنجا می توان فایل سیستم عامل را به ماشین جهت نصب شناساند نمایش داده میشود. همچنین نوع و ورژن سیستم عامل هم می توان انتخاب کرد.



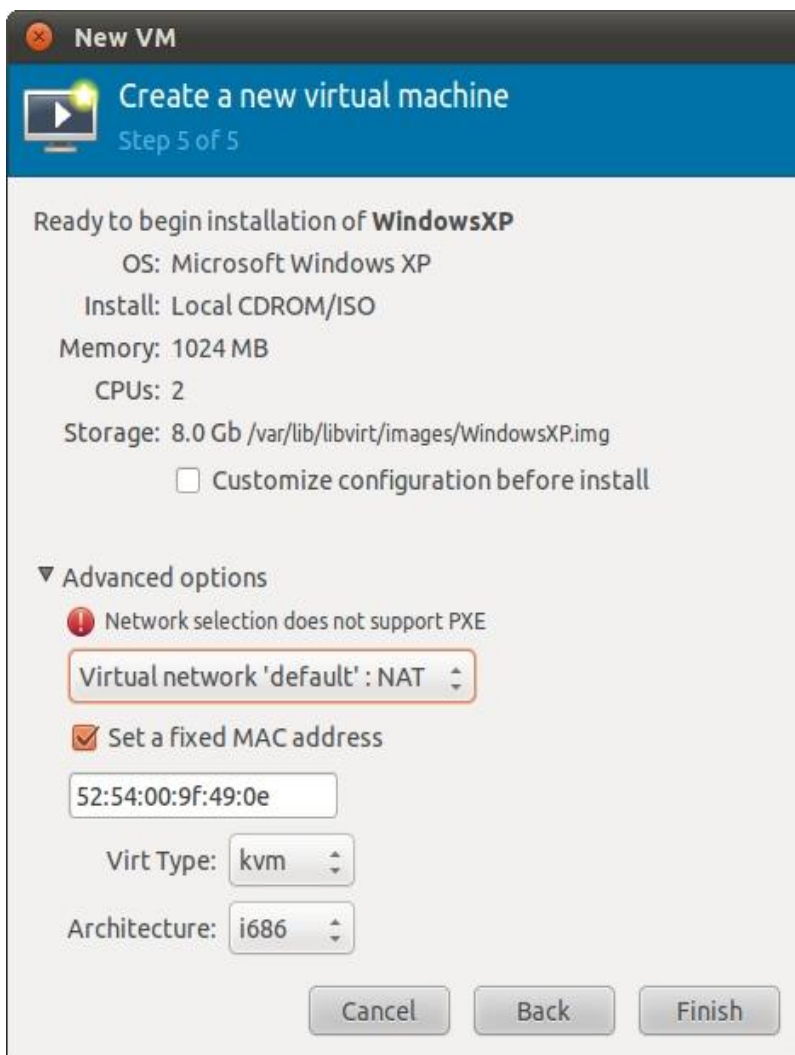
در اینجا مقدار RAM و CPU می خواهید به ماشین مجازی اختصاص دهید را مشخص می کنید.



در اینجا هاردی که می‌خواهیم به ماشین مجازی بدهیم رو مشخص می‌کنیم. می‌توانیم این مقدار را به طور جدید معرفی کنیم یا از فضایی که قبلا ایجاد شده استفاده کرد.



در تصویر زیر که آخرین گام در ایجاد یک ماشین مجازی می‌باشد خلاصه از پیکربندی های انجام شده یا چکیده ای از اطلاعات نمایش می‌دهد. همچنین تنظیمات پیشرفته تری برای تنظیمات شبکه ای حالت **bridge** در دسترس می‌باشد (یعنی ماشین مجازی به کارت شبکه هاست دسترسی مستقیم دارد) جای اینکه از شبکه مجازی در حالت پایه ای NAT استفاده کند. واگر به یک پیکربندی خاصی در **MAC** آدرس نیاز داشته باشد. اگر گزینه **Bridge Networking** در گزینه های منوی زیر نشان داده نشد نیاز هست یک سری تنظیمات در اوبونتو ایجاد شود تا این گزینه فعال شود.



اگر دکمه **finish** بالا را بزنید فرایند ایجاد ماشین مجازی شروع می‌شود. مدیریت مجازی سازی قبل از شروع به کار کردن دیسک را ایجاد و تنظیمات ماشین مجازی را ایجاد می‌کند. در پنجره اصلی **virt-manager** ماشین مجازی جدید نمایان می‌شود. همچنین وضعیت آن را هم می‌توان مشاهده کرد که در تصویر زیر ماشین مجازی استارت شده.

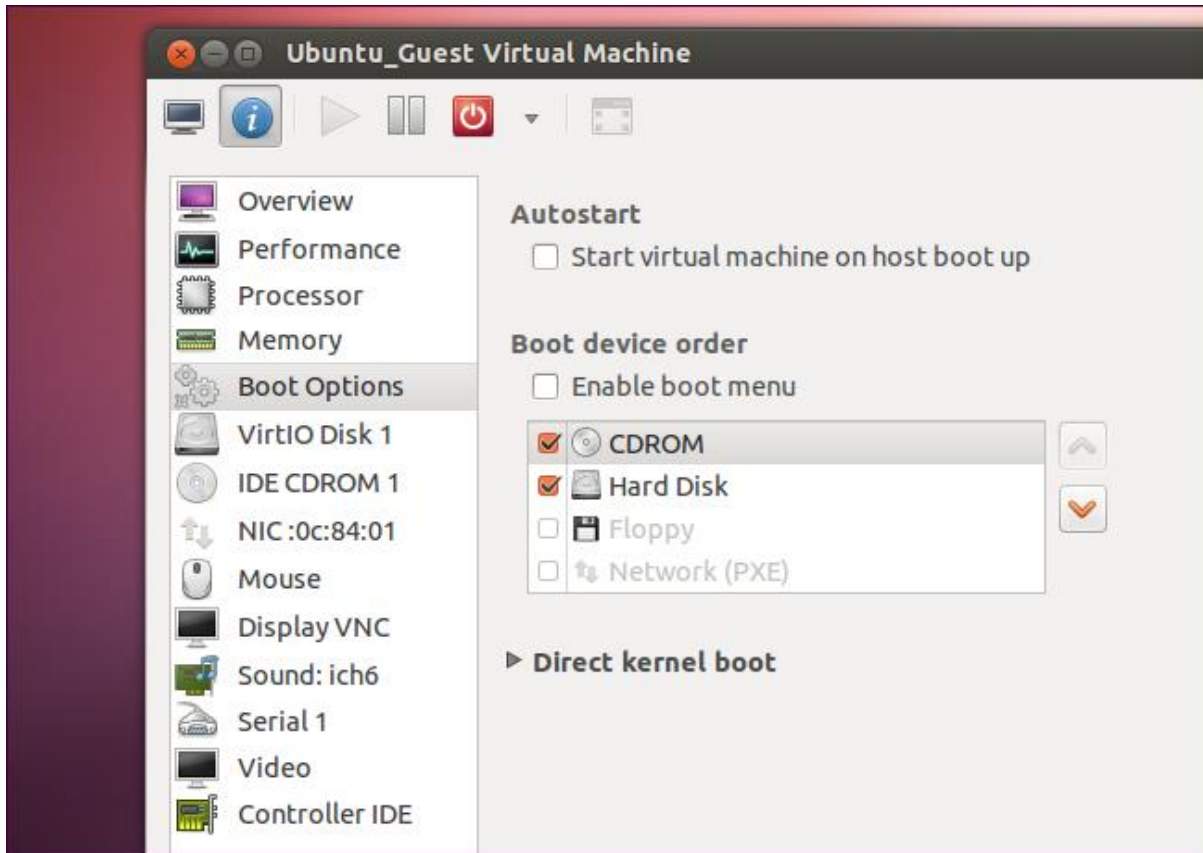


به صورت پیشفرض یک کنسول برای ماشین مجازی بعد از اجرا برای مخاطب به نمایش در می آید و اگر اینطور نشد برای اینکه به پنجره جدید باز شود بعد از انتخاب کردن ماشین مجازی دکمه OPEN که در منوی بالا وجود دارد کلیک کنید. می توانید پنجره ای مانند تصویر زیر مشاهده کنید.



از این نقطه به بعد به راحتی می توان مراحل نصب سیستم عامل را دنبال کرد. در طی فرایند نصب این مورد رو در نظر داشته باشید بعد از کلیک کردن در پنجره کنسولی که برای ماشین مجازی باز شده، کنترل موس و کیبورد در آن محیط متمرکز می شود. برای خروج از محیط ماشین مجازی و بازگشت به سیستم عامل اصلی نیاز است تا کلید های ترکیبی CTRL+ALT را بزنید.

برای مدیریت ماشین مجازی ایجاد شده می توانید بر روی آن ماشین مجازی راست کلیک کنید و به یک سری از امکانات دسترسی داشته باشید. همچنین برای دیدن اطلاعات درباره ماشین مجازی و پیکربندی سخت افزار مجازی آن با کلیک بر روی آیکن-i shaped در پنجره ماشین مجازی به این امکان می توان دست یافت.



اگر به یاد داشته باشید در زمان نصب در جایی اشاره شد که برای استفاده از حالت Bridge نیاز هست تا آن راه اندازی شود، در ادامه این کار را انجام خواهیم داد.

تنظیم شبکه با مدیریت اتصال Bridge

چند راه مختلف برای دسترسی ماشین های مجازی به شبکه های خارجی وجود دارد. به طور پیش فرض پیکربندی شبکه مجازی به عنوان `usermode networking` شناخته شده است و ترافیک از حالت NAT با خارج ارتباط دارد. همچنین شما می توانید پیکربندی `Bridge Networking` را فعال کنید تا هاست از بیرون دسترسی مستقیم به سیستم عامل ماشین های مجازی را داشته باشد.

Usermode networking

در پیکربندی پیشفرض سیستم عامل ماشین های مجازی به سریس های شبکه دسترسی دارند، اما در شبکه دیگر ماشین های مجازی دیده نخواهد شد. ماشین مجازی در اینچنین حالتی فقط یک سری توانایی دارد، به طور مثال از طریق مرورگر به وب دسترسی دارد. اما قادر نمی باشد دسترسی یک هاست سرور وب را داشته باشد.

بطور پیشفرض سیستم عامل مجازی `ip` خود را از رنج `ip` سیستم عامل هاست می گیرد و قادر می باشد از درون سیستم عامل ماشین مجازی به درون هاست ارتباط `SSH` برقرار کند و با استفاده از `SCP` نیز فایل کپی کند.

اگر این تنظیمات برای اهداف شما مناسب است به تنظیمات دیگر نیازی ندارید.

Bridge networking

شبکه های حالت `Bridge` به واسط مجازی اجازه می دهند تا از طریق واسط های فیزیکی به دستگاه های به شبکه های بیروی اتصال پیدا کنند. این را طوری ایجاد می کند که در نتیجه در شبکه مانند یک هاست معمولی به نظر می رسد.

نکته: شبکه های `Bridge` کار نمی کند وقتی دستگاه کارت شبکه (مانند `eth1` و `ath0`) بخواهند برای دستگاه کارت وایرلس (`ipw3945`) استفاده شود. مانند بیشتر درایور های کارت وایرلس که از حالت `Bridge` پشتیبانی نمی کنند.

ایجاد یک network bridge بر روی هاست

برای ایجاد باید سرویس آن را نصب کنید که برای این کار می توانید از دستور زیر استفاده کنید.

نکته: این سرویس قبلا به همراه سرویس های دیگر نصب کرده ایم.

```
sudo apt-get install bridge-utils
```

الان می‌رویم تا شبکه را پیکربندی کنیم برای اینکه این کار به درستی انجام گردد نیاز هست اول شبکه را متوقف کنیم:

```
sudo invoke-rc.d networking stop
```

اگر شما از طریق ریموت به آن اتصال دارید بنابراین نمی‌توانید شبکه را متوقف کنید. می‌توانید به جای متوقف ساختن، آن را بعد از پیکربندی مجدداً راه‌اندازی کنید.

برای تنظیم یک واسط `bridge` در مسیر `etc/network/interfaces` رفته و با استفاده از یک ویرایشگر دستورات زیر را جایگزین تنظیمات موجود کنید:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet static
    address 192.168.0.10
    network 192.168.0.0
    netmask 255.255.255.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

اگر می‌خواهید از DHCP استفاده کنید:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual

auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_stp off
    bridge_fd 0
    bridge_maxwait 0
```

اگر ماشین مجازی در هاست شما وقتی که از حالت **brigde** در شبکه استفاده می‌کند، برای چندثانیه بعد از استارت یا متوقف شدن یک ماشین مجازی KVM فریز (**freezes**) شده است، به این دلیل است یک **bridge** لینوکسی از بین تمام واسط‌هایی که به آنها اتصال دارد کمترین عدد واسط خارجی را به عنوان آدرس سخت‌افزاری می‌گیرد.

برای اینکه بدین شکل کار کند خط زیر را به تنظیمات **Bridge** اضافه کنید.

```
post-up ip link set br0 address f4:6d:04:08:f1:5f
```

و **f4:6d:04:08:f1:5f** جایگزین آدرس سخت‌افزاری از یک کارت شبکه فیزیکی می‌شود که همیشه قسمتی از **bridge** خواهد بود.

این دستورات یک رابط مجازی **br0** ایجاد می‌کند. تنظیمات را ذخیره کرده و از آن خارج شوید.

الان با دستور زیر شبکه را از نو با پیکربندی جدید راه‌اندازی می‌کنیم تا تنظیمات جدید اعمال گردد:

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

در تصویر زیر می‌توانید تنظیمات اعمال شده بر روی کارت شبکه را مشاهده کنید.

```
xentex@xentex-virtual-machine: ~  
File Edit View Search Terminal Help  
xentex@xentex-virtual-machine:~$ ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3f:77:38  
          inet addr:192.168.58.137  Bcast:192.168.58.255  Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe3f:7738/64 Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:53 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:112 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:16150 (16.1 KB)  TX bytes:14288 (14.2 KB)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128 Scope:Host  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:48 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:4709 (4.7 KB)  TX bytes:4709 (4.7 KB)  
  
virbr0    Link encap:Ethernet  HWaddr ca:d2:8b:0e:2a:59  
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
xentex@xentex-virtual-machine:~$ █
```