

گزارش نصب و راه اندازی یک ماشین مجازی مبتنی بر KVM

تهیه و تنظیم: مبین خیری

شماره دانشجویی: 994421017

استاد راهنما: دکتر لیلا شریفی

چکیده:

در گزارش پیش رو قصد داریم روند نصب و راه اندازی یک ماشین مجازی مبتنی بر KVM را قدم به قدم تشریح کرده و در نهایت برای ارزیابی عملکرد این ماشین مجازی، یک فولدر با نام دلخواه خود به همراه یک فایل با شماره دانشجویی مرتبط با آن نام را در داخل ماشین مجازی ساخته و توسط دستور ls فایل و فولدر مربوطه را لیست کنیم. به منظور آشنایی بیشتر با روند حل این مسئله، ابتدا با مفهوم مجازی سازی و واژگان فنی مرتبط با آن آشنا شده و سپس روی حل این چالش تمرکز خواهیم کرد.

معرفی مجازی ساز KVM

مفهوم مجازی سازی مدتی است که وجود داشته و فناوری های کاملاً مدبرانه و مقرون به صرفه ای را به اثبات رسانده است. تیم های عملیاتی و کاربران دسکتاپ به طور یکسان می توانند چندین ماشین مجازی را داشته باشند و مجموعه گسترده ای از سیستم عامل ها را بدون نیاز به نصب هر کدام بر روی یک سرور فیزیکی جداگانه اجرا کنند. ماشین های مجازی با استفاده از هایپروایزر ایجاد می شوند. یکی از این Hypervisor های رایج KVM نام دارد که رایگان و منبع باز است.

پیش از آن که به پاسخ به سوال KVM چیست و معرفی مجازی ساز KVM بپردازیم بیایید کمی بیشتر در مورد هایپروایزرها و مجازی سازهای مختلف اطلاعات کسب کنیم.

هایپروایزر (Hypervisor)

یک نرم افزار لایه میانی که بین یک سرور اختصاصی فیزیکی و یک سیستم عامل اجرا می شود و به چندین سیستم عامل و برنامه اجازه می دهد مجموعه ای از سخت افزار فیزیکی اولیه را به اشتراک بگذارند. Hypervisor را می توان به عنوان یک سیستم عامل "ممتا" در محیط مجازی در نظر گرفت که می تواند دسترسی به تمام دستگاه های فیزیکی و ماشین های مجازی روی سرور را هماهنگ کند، بنابراین به آن مانیتور ماشین مجازی (VMM) نیز می گویند. هنگامی که سرور، Hypervisor را راه اندازی و اجرا می کند، مقدار مناسبی از حافظه، CPU، شبکه و منابع دیسک را به هر ماشین مجازی اختصاص می دهد و سیستم عامل کلاینت تمام ماشین های مجازی را بارگذاری می کند.

در واقع، هایپروایزر برای سیستم عامل همان چیزی است که سیستم عامل برای processها انجام می دهد. آن ها یک پلت فرم سخت افزار مجازی مستقل برای اجرا فراهم می کنند که به نوبه خود دسترسی مجازی کامل به ماشین زیرین را فراهم می کند. اما همه هایپروایزرها یکسان نیستند، که چیز خوبی است؛ زیرا لینوکس به انعطاف پذیری و انتخاب پذیری معروف است.

انواع هایپروایزرهای رایج:

نوع اول : (Bare-metal)-به این معنی است که VMM مستقیماً روی لایه سخت افزار عمل می کند تا منابع سخت افزاری زیرین را استفاده و مدیریت کند. سیستم عامل مهمان (Guest OS) از طریق VMM به منابع سخت افزاری واقعی دسترسی دارد VMM. به عنوان اپراتور مستقیم سخت افزار اصلی، درایور سخت افزار را دارد.

نوع دوم (Hosted) به این معنی است که سیستم عامل میزبان (Host OS) دیگری تحت VMM وجود دارد. از آنجایی که سیستم عامل مهمان باید از طریق سیستم عامل میزبان به سخت افزار دسترسی داشته باشد، سربار عملکرد اضافی را به همراه دارد، اما می تواند از درایورهای دستگاه و خدمات اساسی ارائه شده توسط سیستم عامل میزبان برای مدیریت حافظه، زمان بندی فرآیند و مدیریت منابع استفاده کامل کند.

مجازی سازی

مجازی سازی فرآیندی است که در آن سخت افزار فیزیکی زیربنایی پنهان می شود تا سیستم عامل های متعدد بتوانند به طور شفاف از آن استفاده کرده و به اشتراک بگذارند. نام رایج دیگر این معماری مجازی سازی پلتفرم است. در معماری لایه ای معمولی، لایه ای که مجازی سازی پلتفرم را فراهم می کند، Hypervisor نامیده می شود. سیستم عامل های مهمان ماشین های مجازی (VMS) نامیده می شوند زیرا سخت افزار به طور خاص برای این VMS مجازی سازی شده است.

در اصل بوسیله این تکنولوژی می توانید به طور همزمان چندین سیستم عامل را روی یک سرور (سخت افزاری) راه اندازی کنید. به بیانی دیگر، با نصب مجازی ساز بر روی یک سرور فیزیکی، می توانید چندین سرور مجازی ایجاد کنید. در واقع استفاده از مجازی ساز، روش منطقی برای تقسیم منابع سیستم است.

دلایل زیادی از جمله صرفه جویی در زمان، مدیریت بهتر، افزایش امنیت اطلاعات، افزایش کیفیت سرویس نرم افزارها و غیره مسبب استفاده کاربران از مجازی سازها است. از جمله این مجازی سازها می توانیم به اسامی معتبر و مشهوری همچون موارد زیر اشاره کنیم.

- VMWARE
- KVM
- CITRIX
- MICROSOFT
- RED HAT

- CISCO
- XEN
- OpenVZ

امروزه اغلب زیر ساخت های مجازی سازی بر پایه لینوکس هستند. البته مجازی سازی ویندوز با نام "Hyper V" نیز وجود دارد، ولی همانند سایر رقبای خود کارایی چندانی ندارد. از جمله پرکاربردترین زیرساخت های مجازی سازی که شرکت ها از آن ها استفاده می کنند، VMWARE و KVM هستند. در ادامه هر یک را بررسی خواهیم کرد.

VMWARE

معروف ترین و محبوبترین زیرساخت برای مجازی سازی است. بیشتر شرکت های هاستینگ از این زیرساخت استفاده می کنند. بسیاری از سازمان ها و شرکت ها هم از همین زیرساخت استفاده می کنند. زیر ساخت VMWARE در کشور ما رایگان نیست به همین علت از محصولات آن در ایران همیشه به صورت کرک شده استفاده می کنیم. محصول مجازی ساز این شرکت "ESXi" دارد که به صورت یک سیستم عامل ارائه شده است.

KVM

مجازی ساز "KVM" ماشین مجازی مبتنی بر هسته) بر پایه لینوکس و کاملاً رایگان است. اگر لینوکس ۲۰۰۶/۲ یا جدیدتر دارید، KVM را به صورت پیش فرض و خودکار دارید. این تکنولوژی اولین بار در سال ۲۰۰۶ اعلام شد و یک سال بعد در نسخه اصلی هسته لینوکس ادغام شد. لازم به ذکر است که redhat، یکی از توسعه دهنده های اصلی مجازی ساز KVM است.

همچنین KVM از مجازی سازی تودرتو پشتیبانی می کند که به کاربران امکان می دهد یک VM را در داخل ماشین مجازی دیگر اجرا کنند. بهتر است این را مدنظر داشته باشید که KVM hypervisor یک راه حل کامل مجازی سازی لینوکس بر روی سخت افزارهای x86 می باشد.

هسته استاندارد لینوکس تعبیه شده با ماژول KVM می تواند GuestOS بارگذاری شده از طریق ابزارهای KVM را پشتیبانی کند. بنابراین، تحت چنین پلتفرم سیستم عاملی، لایه مجازی سازی VMM مستقیماً روی لایه سخت افزار فیزیکی رایانه قرار دارد، اما هیچ لایه سیستم عامل مستقل برای هاست وجود ندارد. در چنین محیطی، سیستم عامل هاست یک VMM است.

در واقع هر GuestOS ایجاد شده توسط KVM یک فرآیند واحد (process) در سیستم عامل هاست) یا (VMM است).

همچنین مجازی سازی KVM نیاز به پشتیبانی سخت افزاری دارد) مانند فناوری Intel VT یا AMD V Technology که مجازی سازی کامل مبتنی بر سخت افزار است.

ویژگی های KVM

برخی از ویژگی های کلیدی آن شامل:

- پشتیبانی از طیف گسترده ای از پلتفرم های سخت افزاری لینوکس (سخت افزار x86 با پسوند مجازی سازی Intel-VT یا AMD-V)
- امنیت VM و ایزوله سازی پیشرفته تر را با استفاده از SELinux
- مجازی سازی امن (sVirt)
- پشتیبانی از Live Migration (مهاجرت یک VM در حال اجرا بین میزبان های فیزیکی)
- قابلیت افزایش بلادرنگ منابع سرور (ram,cpu,hard)

مزایای KVM

- پشتیبانی از KSM یا Kernel Same-page Merging برای حذف صفحات حافظه تکراری و کاهش حافظه مصرفی
- امنیت عالی KVM : از ترکیبی از لینوکس پیشرفته امنیتی (SELinux) و مجازی سازی امن (sVirt) برای امنیت و انزوا پیشرفته VM استفاده می کند.
- پشتیبانی از سیستم عامل ویندوز و لینوکس
- استفاده گسترده از KVM در میان نرم افزارهای اتوماسیون open source
- عدم وجود overseeing
- KVM می تواند از هرگونه حافظه رایج پشتیبانی شده توسط لینوکس ، از جمله برخی از Local Disk ها و ذخیره سازهای متصل به شبکه (NAS) استفاده کند.
- پشتیبانی از FILE SYSTEM به منظور رویت image های ماشین مجازی ها توسط هاست های مختلف

معایب KVM:

- پیچیدگی فرآیند پیکربندی به نسبت سایر سیستم های مجازی ساز
- تخصیص منابع باعث محدودیت تعداد VM ها با توجه به سخت افزار مورد استفاده خواهد بود.

نصب و راه اندازی KVM

نکته ای که در مورد این فناوری باید به آن دقت کنید، این است که برای استفاده از آن در لینوکس نیازی به نرم افزارهای جانبی ندارید. برای نصب و راه اندازی KVM تنها به نسخه ای از لینوکس نیاز دارید که بعد

از 2007 منتشر شده باشد و روی سخت افزار X86 باید نصب شود که از قابلیت های مجازی سازی پشتیبانی کند. همچنین شما می توانید از KVM برای اجرای ویندوز و لینوکس در ماشین های مجازی استفاده کنید.

پیش نیازها

- یک CPU با قابلیت پشتیبانی از فناوری مجازی سازی
- دانلود و نصب KVM
- پیکربندی

تفاوت KVM و QEMU

صحبت در مورد KVM، صحبت در مورد فناوری مجازی سازی یا در مورد ماژول های هسته (kvm.ko، kvm-intel.ko یا kvm-amd.ko) است. گاهی اوقات از KVM به عنوان ماشین مجازی یاد می شود، اما این درست نیست، زیرا KVM سخت افزار مجازی را ارائه نمی دهد. اینجاست که QEMU به بازی می آید.

QEMU یک VMM روی یک میزبان است که CPU را از طریق تبدیل باینری پویا شبیه سازی می کند و یک سری مدل های سخت افزاری را ارائه می کند تا سیستم عامل Guest فکر کند که مستقیماً با سخت افزار سروکار دارد. در واقع، آنها با سخت افزار شبیه سازی شده توسط QEMU سروکار دارند و QEMU این دستورالعمل ها را برای عملیات به سخت افزار واقعی ترجمه می کند. از طریق این حالت، سیستم عامل Guest می تواند با هارد دیسک، کارت شبکه، CPU، CD-ROM، دستگاه صوتی و دستگاه USB موجود در هاست تعامل داشته باشد. با این حال، از آنجایی که تمام دستورالعمل ها باید توسط QEMU ترجمه شوند، عملکرد ضعیف خواهد بود.

برای کسب اطلاعات بیشتر می توانید به این [سایت](#) مراجعه کنید.

مزایای مجازی سازی

ادغام سرورها: کاملاً قابل درک است که مجازی سازی باعث کاهش مصرف و ذخیره سازی انرژی می شود. مجازی سازی همچنین باعث کاهش مصرف در کل مراکز داده و نیز کاهش تعداد سرورها و دیگر تجهیزات فیزیکی از جمله رک (Rack) ها می شود. این امر موجب کاهش میزان مساحت مورد نیاز مراکز داده، کاهش مصرف انرژی و... می گردد. که در نهایت منجر به کاهش هزینه ها و بهبود در مصرف انرژی می شود. اما آیا مجازی سازی باعث بهره وری بیشتر در سخت افزار نیز می شود؟ با استفاده از مجازی سازی می توان دقیقاً به همان اندازه که ماشین مجازی نیاز به پردازنده (CPU)، حافظه (Ram) و منابع ذخیره سازی (Storage resources) دارد به آن منابع اختصاص داد در نتیجه باعث افزایش بهره وری شد.

جداسازی سرویس ها: فرض کنید که مجازی سازی وجود نداشت در این شرایط راه حل شما برای جداسازی سرویس ها چه بود؟ تنها راه حل جدا کردن فیزیکی سرورهاست با وجود اینکه با این روش میتوان از جداسازی سرویس ها اطمینان حاصل کرد اما آیا باعث رشد بی رویه ی سرورها و کاهش بهره وری و

افزایش هزینه ها نمی شود؟ مجازی سازی منجر به ایزوله سازی نرم افزارها و همچنین رفع مشکل عدم مطابقت برنامه ها می شود. این امر همچنین مدیریت سرویس ها را نیز آسانتر می کند.

راه اندازی سریعتر سرورها: راه اندازی سرور فیزیکی زمان بر است ،اما با مجازی سازی ،میتوان با استفاده از یک image از قبل تهیه شده یا از یک snapshot یک سرور مجازی راه اندازی کنید.

بازیابی از حادثه: (Disaster Recovery) هنگامی که شما یک مرکز داده مجازی داشته باشید بازیابی از حوادث بسیار ساده خواهد بود. مجازی سازی این امکان را به شما می دهد که snapshot های بروز از سرورهای خود داشته باشید. این Snapshot ها میتوانند به سادگی بازیابی و پیکربندی شوند تا سرورها به وضعیت قبل حادثه خود بازگردند. همچنین مجازی سازی قابلیت هایی مانند مهاجرت ماشین مجازی (Vm Migration) بصورت آنلاین و آفلاین را در اختیار شما قرار می دهد بنابراین شما همیشه می توانید ماشین های مجازی خود را به نقطه دیگری در مرکز داده منتقل کنید. این انعطاف پذیری کمک می کند تا یک طرح بازیابی با قابلیت اجرا و میزان موفقیت بالاتری داشته باشیم.

تنظیم بار پویا: (Dynamic load balancing) مجازی سازی این قابلیت را در اختیار ماشین های مجازی قرار می دهند تا بر اساس سیاست هایی که اعمال می کنید با تغییر بار سرورها ،سرویس ها از سروری که منابع آن بیش از اندازه در حال استفاده است به سروری که منابع آزاد دارد منتقل شوند (Live migration) ، اکثر راهکارهای مجازی سازی دارای چنین سیاست هایی برای کاربران هستند. این شیوه تنظیم بار پویا باعث بالا رفتن بهره وری در منابع سرورها می شود.

محیط های آزمایشی و توسعه ی سریعتر: اگر بخواهید محیطی را بشکل موقت تست کنید پیکربندی آن بر روی سرور فیزیکی دشوار است همچنین بدلیل موقت بودن آن ارزش چندانی هم ندارد. اما با استفاده از مجازی سازی راه اندازی محیط آزمایشی یا توسعه بسیار آسان است . با استفاده از یک سیستم عامل میهمان امکان پیکربندی سریع را با استفاده از جداسازی برنامه در یک محیط شناخته و کنترل شده به شما می دهد.

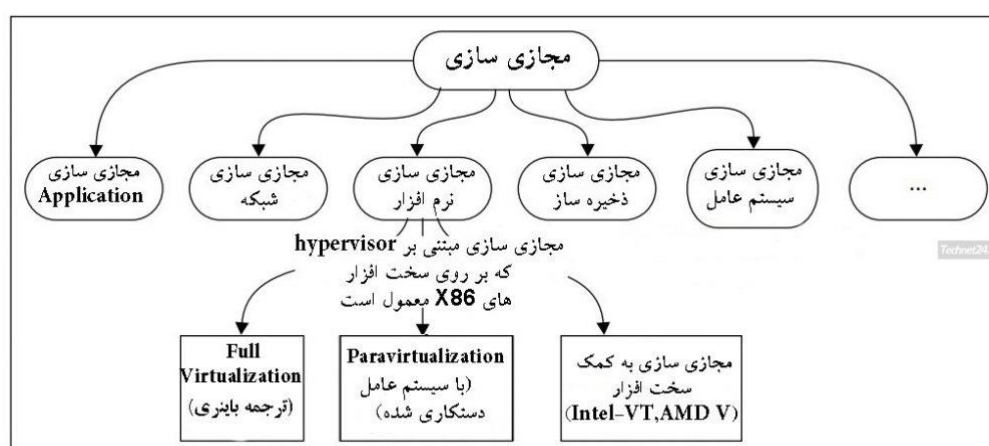


این کار همچنین باعث حذف بسیاری از فاکتورهای ناشناخته مانند کتابخانه (Libraries) های ناشناخته که از نصب نرم افزارهای بیشمار ناشی می شود را حذف می کند. مخصوصا اگر این محیط یک محیط آزمایشی یا توسعه باشد در هنگام اجرای آزمایش میتوان انتظار از کار افتادن سرور را داشت در صورتیکه اینکار در یک سرور فیزیکی انجام شود بازگردانی سرور احتیاج به ساعت ها نصب و پیکربندی دوباره سرور است درحالیکه با استفاده از سرور مجازی تنها احتیاج به یک فایل ایمیج مجازی است تا دوباره سرور را راه اندازی کنید.

بهبود امنیت و پایداری سیستم: راهکارهای مجازی سازی یک لایه انتزاعی (Abstraction layer) میان سخت افزار فیزیکی و ماشین های مجازی اضافه می کنند. بسیار شایع است که داده های روی دیسک فیزیکی آسیب ببینند و کل سرور را تحت تاثیر قرار دهند اما اگر این اطلاعات بر روی هارد دیسک ماشین مجازی ذخیره شوند با آسیب دیدن اطلاعات، دیسک فیزیکی از آسیب مصون می ماند و هارد دیسک های مجازی هم نیازی به نگرانی ندارند.

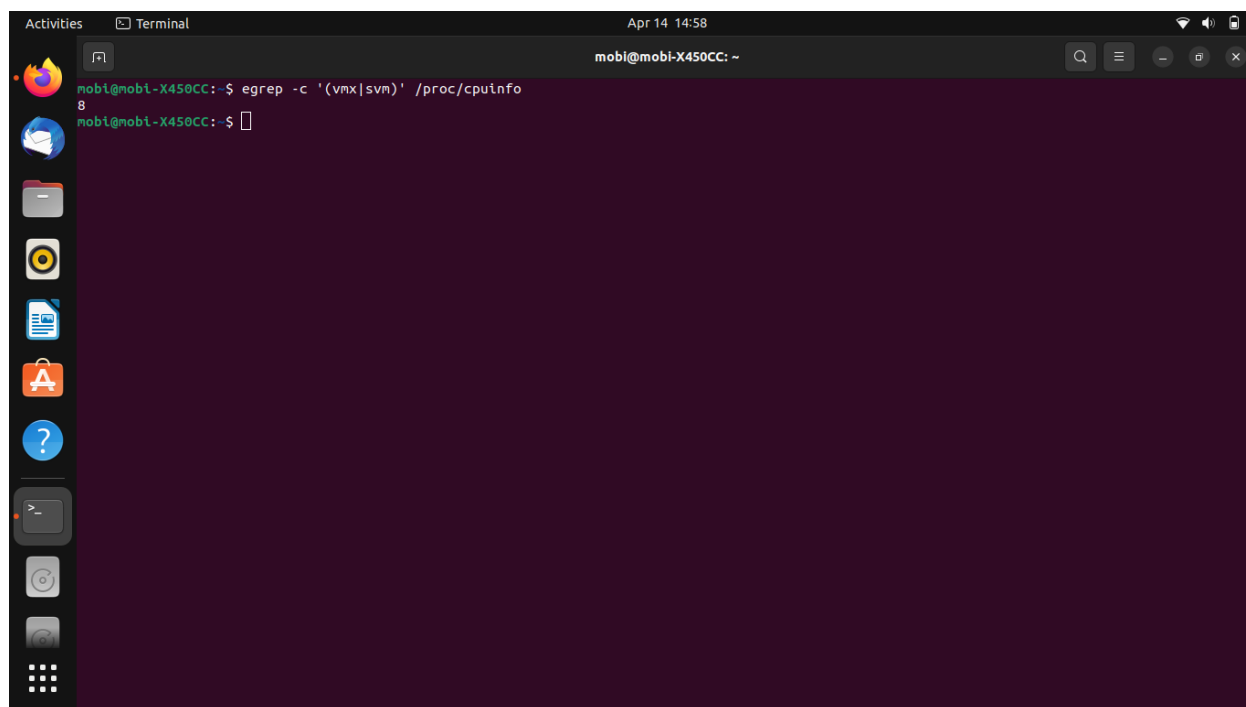
در موارد مشابه نیز مجازی سازی میتواند مانع از کار افتادن سیستم بخاطر آسیب دیدن حافظه توسط نرم افزارهایی مانند درایورها شود. مدیر سیستم (Administrator) امکان این را دارد که ماشین مجازی را در محیطی مجزا و مستقل پیکربندی کند. این پیکربندی سندباکس (Sandbox) ماشین های مجازی امنیت بیشتری به زیرساخت سیستم می دهد زیرا مدیر سیستم از این انعطاف پذیری برخوردار است تا بهترین تنظیمات ممکن را انتخاب کند. اگر مدیر تشخیص دهد که یک ماشین مجازی نیازی به دسترسی به اینترنت یا شبکه های دیگر ندارد می تواند ماشین مجازی را به آسانی به نحوی تنظیم کند تا از دسترسی به اینترنت منع شود. این امر به کاهش ریسک آلوده شدن یک سیستم که در ادامه باعث آلوده شدن سیستم های دیگر یا ماشین های مجازی نیز می شود کمک می کند.

جلوگیری از وابستگی به سخت افزار: مجازی سازی یک لایه ی انتزاعی مابین سخت افزار و سیستم عامل ایجاد میکند. اینکار باعث میشود تا از مشکلات ناشی از وابستگی به یک سازنده بخصوص (Vendor lock-in) جلوگیری شود، مخصوصا هنگامی که ماشین های مجازی به سخت افزاری که بر روی آن اجرا می شوند وابستگی خاصی نداشته باشند، به همین دلیل مدیران مراکز داده انعطاف پذیری بیشتری در هنگام انتخاب تجهیزات سرورها خواهند داشت. بطور خلاصه مزیت مجازی سازی مستقل بودن آن از سخت افزار و جداسازی ماشین های مجازی از سخت افزار سیستم فیزیکی است. این خاصیت به در دسترس بودن سیستم (Availability) و تداوم کار کمک میکند. یکی از نکات قوت راهکار مجازی سازی ایجاد لایه انتزاعی مابین سخت افزار و نرم افزار سیستم است.

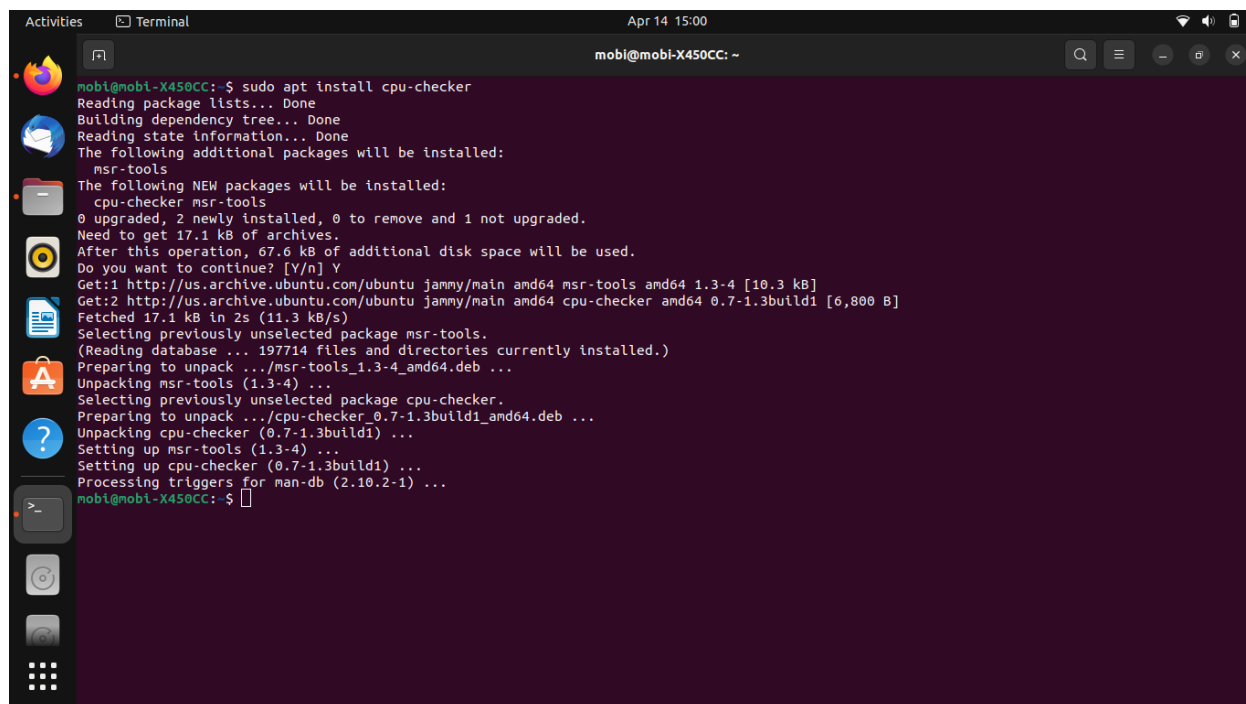


همانطور که در بخش های قبل ذکر شد، با وجود اینکه مجازی سازی به روش های متفاوتی قابل اجراست روش های مد نظر ما مجازی سازی سیستم عامل و مجازی سازی نرم افزار است.

حال نوبت به نصب و راه اندازی یک ماشین مجازی مبتنی بر KVM می‌رسد. در این قسمت به جهت جلوگیری از طولانی شدن گزارش، طبق دستورالعمل‌های آورده شده در این [وبسایت](#) عمل کرده و تنها تصاویر مربوط به اجرا و پیاده سازی را ثبت خواهیم کرد.



```
mobl@mobl-X450CC: ~  
mobl@mobl-X450CC:~$ egrep -c '(vmx|svm)' /proc/cpuinfo  
8  
mobl@mobl-X450CC:~$
```



```
mobl@mobl-X450CC:~$ sudo apt install cpu-checker  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  msr-tools  
The following NEW packages will be installed:  
  cpu-checker msr-tools  
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.  
Need to get 17.1 kB of archives.  
After this operation, 67.6 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 msr-tools amd64 1.3-4 [10.3 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 cpu-checker amd64 0.7-1.3build1 [6,800 B]  
Fetched 17.1 kB in 2s (11.3 kB/s)  
Selecting previously unselected package msr-tools.  
(Reading database ... 197714 files and directories currently installed.)  
Preparing to unpack ../msr-tools_1.3-4_amd64.deb ...  
Unpacking msr-tools (1.3-4) ...  
Selecting previously unselected package cpu-checker.  
Preparing to unpack ../cpu-checker_0.7-1.3build1_amd64.deb ...  
Unpacking cpu-checker (0.7-1.3build1) ...  
Setting up msr-tools (1.3-4) ...  
Setting up cpu-checker (0.7-1.3build1) ...  
Processing triggers for man-db (2.10.2-1) ...  
mobl@mobl-X450CC:~$
```



```
Activities Terminal Apr 14 15:01
mobi@mobi-X450CC: ~
mobi@mobi-X450CC:~$ sudo apt update
[sudo] password for mobi:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [41.6 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [18.5 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [471 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [996 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [101 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [900 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [608 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [269 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 DEP-11 Metadata [940 B]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Metadata [8,000 B]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11 Metadata [12.5 kB]
Fetched 3,763 kB in 19s (196 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
mobi@mobi-X450CC:~$
```

```
Activities Terminal Apr 14 15:04
mobi@mobi-X450CC: ~
Setting up libpmem1:amd64 (1.11.1-3build1) ...
Setting up libonig5:amd64 (6.9.7.1-2build1) ...
Setting up libgfxdr0:amd64 (10.1-1) ...
Setting up librdmacm1:amd64 (39.0-1) ...
Setting up librados2 (17.2.5-0ubuntu0.22.04.2) ...
Setting up qemu-system-common (1:6.2+dfsg-2ubuntu6.7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/qemu-kvm.service → /lib/systemd/system/qemu-kvm.service.
Setting up libjq1:amd64 (1.6-2.1ubuntu3) ...
Setting up libvirt-clients (8.0.0-1ubuntu7.4) ...
Setting up qemu-system-x86 (1:6.2+dfsg-2ubuntu6.7) ...
Setting up libnss-mymachines:amd64 (249.11-0ubuntu3.9) ...
First installation detected...
Checking NSS setup...
Setting up swtpm (0.6.3-0ubuntu3) ...
Setting up libpmemobj1:amd64 (1.11.1-3build1) ...
Setting up thin-provisioning-tools (0.9.0-2ubuntu1) ...
Setting up libdecor-0-plugin-1-cairo:amd64 (0.1.0-3build1) ...
Setting up libvirt-daemon-driver-qemu (8.0.0-1ubuntu7.4) ...
Setting up librbdl (17.2.5-0ubuntu0.22.04.2) ...
Setting up libvirt-daemon-system-systemd (8.0.0-1ubuntu7.4) ...
Setting up jq (1.6-2.1ubuntu3) ...
Setting up qemu-utils (1:6.2+dfsg-2ubuntu6.7) ...
Setting up libiscsi7:amd64 (1.19.0-3build2) ...
Setting up libvirt-daemon (8.0.0-1ubuntu7.4) ...
Setting up libgfrpc0:amd64 (10.1-1) ...
Setting up mdevctl (0.81-1) ...
Setting up qemu-system-gui (1:6.2+dfsg-2ubuntu6.7) ...
Setting up swtpm-tools (0.6.3-0ubuntu3) ...
Adding group 'swtpm' (GID 138) ...
Done.
Warning: The home dir /var/lib/swtpm you specified can't be accessed: No such file or directory
Adding system user 'swtpm' (UID 129) ...
Adding new user 'swtpm' (UID 129) with group 'swtpm' ...
Not creating home directory '/var/lib/swtpm'.
Setting up libgfat10:amd64 (10.1-1) ...
Setting up libvirt-daemon-system (8.0.0-1ubuntu7.4) ...
Progress: [ 95%] [#####.....]
```

```
Activities Terminal Apr 14 15:05
mobi@mobi-X450CC: ~
Setting up libvirt-daemon-system (8.0.0-1ubuntu7.4) ...
Adding user libvirt-qemu to group libvirt-qemu
Not enabling default network as no free network was found
Created symlink /etc/systemd/system/multi-user.target.wants/libvirtd.service → /lib/systemd/system/libvirtd.service.
Created symlink /etc/systemd/system/sockets.target.wants/virtlockd.socket → /lib/systemd/system/virtlockd.socket.
Created symlink /etc/systemd/system/sockets.target.wants/virtlogd.socket → /lib/systemd/system/virtlogd.socket.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd.socket → /lib/systemd/system/libvirtd.socket.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd-ro.socket → /lib/systemd/system/libvirtd-ro.socket.
Created symlink /etc/systemd/system/multi-user.target.wants/libvirt-guests.service → /lib/systemd/system/libvirt-guests.service.
virtlockd.service is a disabled or a static unit, not starting it.
virtlogd.service is a disabled or a static unit, not starting it.
Created symlink /etc/systemd/system/sockets.target.wants/libvirtd-admin.socket → /lib/systemd/system/libvirtd-admin.socket.
Created symlink /etc/systemd/system/sockets.target.wants/virtlockd-admin.socket → /lib/systemd/system/virtlockd-admin.socket.
Created symlink /etc/systemd/system/sockets.target.wants/virtlogd-admin.socket → /lib/systemd/system/virtlogd-admin.socket.
Setting up libvirt-daemon dnsmasq configuration.
Setting up qemu-block-extra (1:6.2+dfsg-2ubuntu6.7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/run-qemu.mount → /lib/systemd/system/run-qemu.mount.
Setting up liblvm2cmd2.03:amd64 (2.03.11-2.1ubuntu4) ...
Setting up dmeventd (2:1.02.175-2.1ubuntu4) ...
Created symlink /etc/systemd/system/sockets.target.wants/dm-event.socket → /lib/systemd/system/dm-event.socket.
dm-event.service is a disabled or a static unit, not starting it.
Setting up lvm2 (2.03.11-2.1ubuntu4) ...
update-initramfs: deferring update (trigger activated)
Created symlink /etc/systemd/system/sysinit.target.wants/blk-availability.service → /lib/systemd/system/blk-availability.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-monitor.service → /lib/systemd/system/lvm2-monitor.service.
Created symlink /etc/systemd/system/sysinit.target.wants/lvm2-lvmpolld.socket → /lib/systemd/system/lvm2-lvmpolld.socket.
Processing triggers for dbus (1.12.20-2ubuntu4.1) ...
Processing triggers for initramfs-tools (0.140ubuntu13.1) ...
update-initramfs: Generating /boot/initrd.img-5.19.0-38-generic
I: The initramfs will attempt to resume from /dev/sda8
I: (UUID=9d55cece-7425-4f96-8bdd-e1a546b7b2bf)
I: Set the RESUME variable to override this.
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
mobi@mobi-X450CC: $
```

```
Activities Terminal Apr 14 15:07
mobi@mobi-X450CC: ~
mobi@mobi-X450CC: $ sudo adduser 'mobi' libvirt
adduser: The user 'mobi' does not exist.
mobi@mobi-X450CC: $ sudo adduser "mobi" libvirt
The user 'mobi' is already a member of 'libvirt'.
mobi@mobi-X450CC: $
```

```
Activities Terminal Apr 14 15:08 mobi@mobi-X450CC: ~
mobi@mobi-X450CC:~$ sudo adduser "mobi" kvm
Adding user 'mobi' to group 'kvm' ...
Adding user mobi to group kvm
Done.
mobi@mobi-X450CC:~$
```

```
Activities Terminal Apr 14 15:08 mobi@mobi-X450CC: ~
mobi@mobi-X450CC:~$ virsh list --all
Id Name State
-----
mobi@mobi-X450CC:~$
```

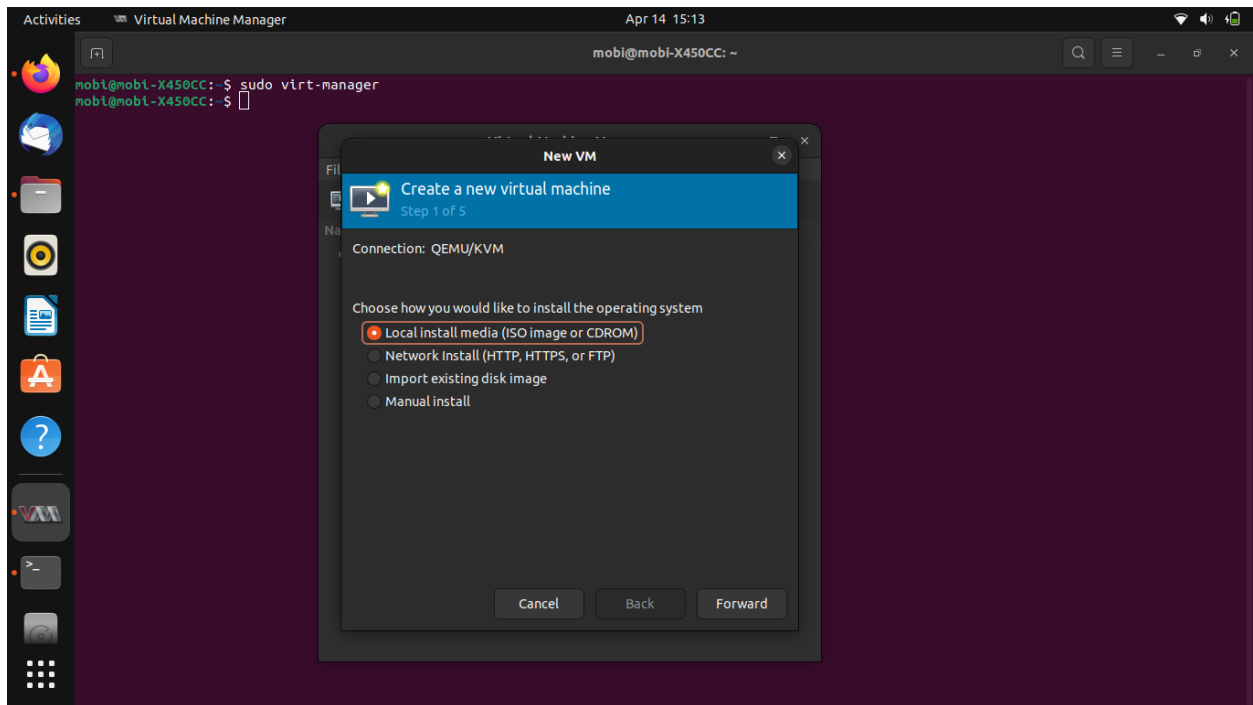
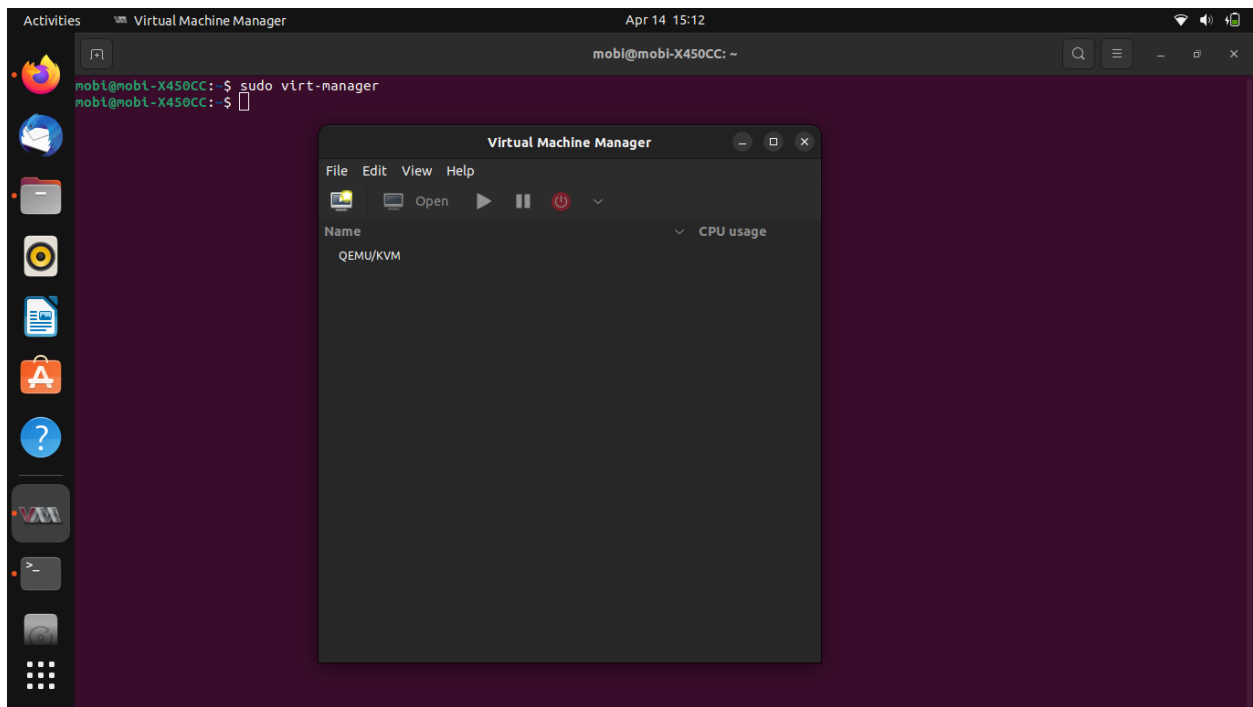
```
Activities Terminal Apr 14 15:09
mobi@mobi-X450CC: ~
mobi@mobi-X450CC:~$ virsh list --all
Id Name State
-----
mobi@mobi-X450CC:~$ sudo systemctl status libvirtd
● libvirtd.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirtd.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-04-14 15:04:39 +0330; 4min 19s ago
   TriggeredBy: ● libvirtd-ro.socket
                 ● libvirtd-admin.socket
                 ● libvirtd.socket
   Docs: man:libvirtd(8)
          https://libvirt.org
   Main PID: 7822 (libvirtd)
     Tasks: 19 (limit: 32768)
    Memory: 8.9M
       CPU: 476ms
    CGroup: /system.slice/libvirtd.service
            └─7822 /usr/sbin/libvirtd

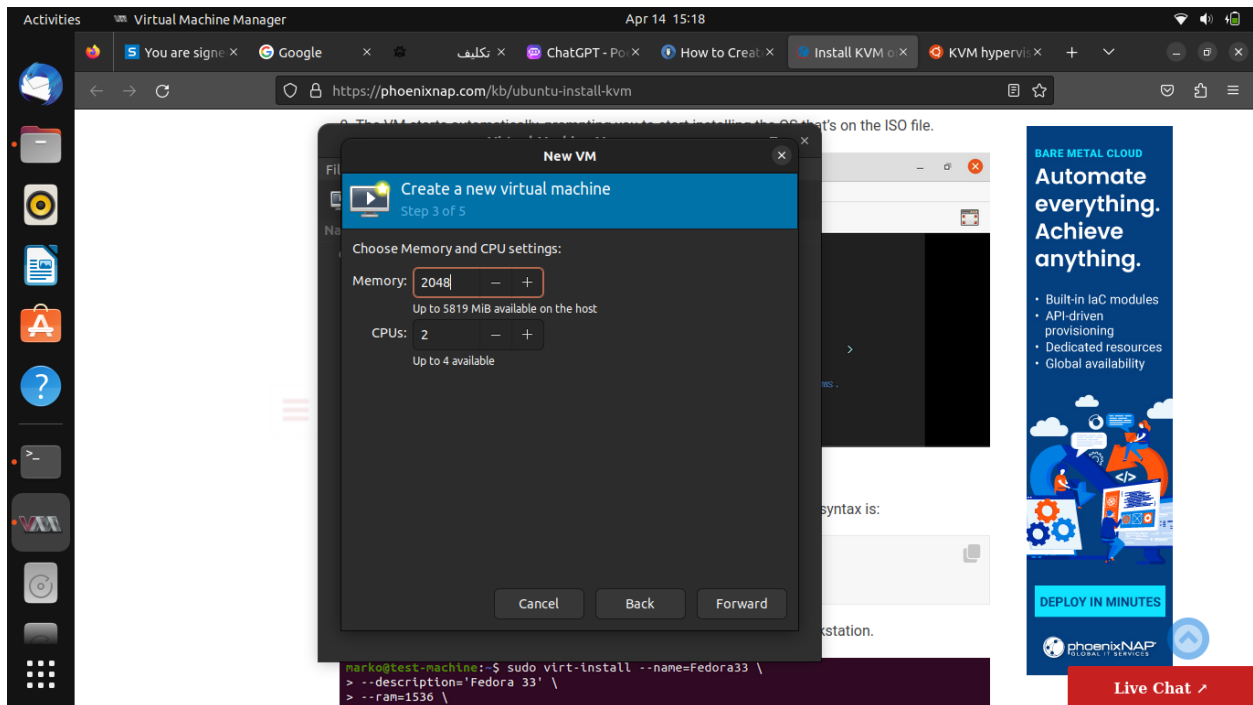
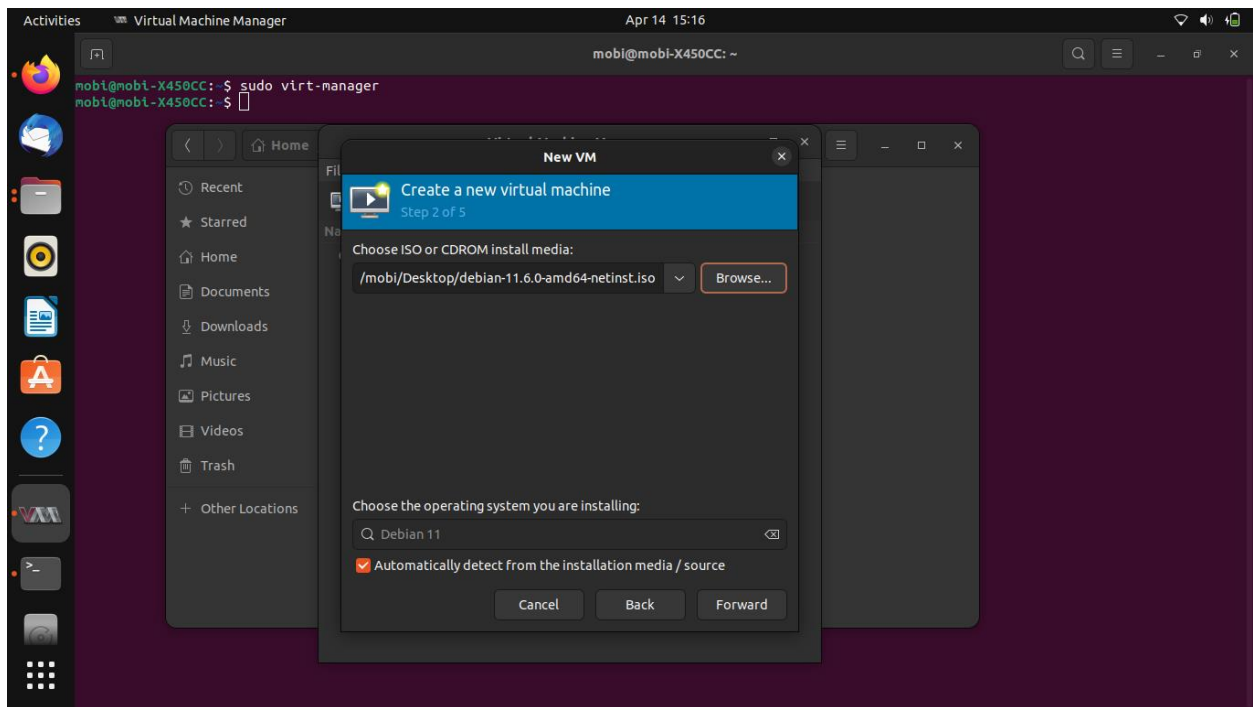
Apr 14 15:04:39 mobi-X450CC systemd[1]: Starting Virtualization daemon...
Apr 14 15:04:39 mobi-X450CC systemd[1]: Started Virtualization daemon.
mobi@mobi-X450CC:~$
```

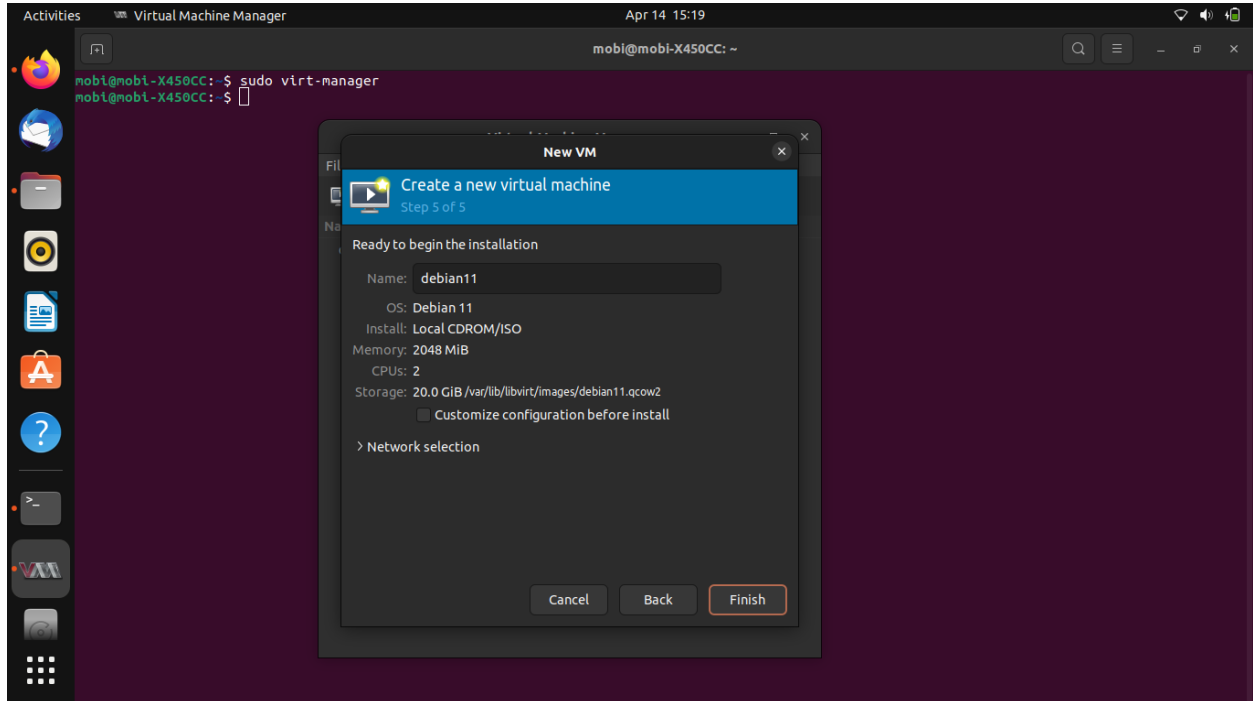
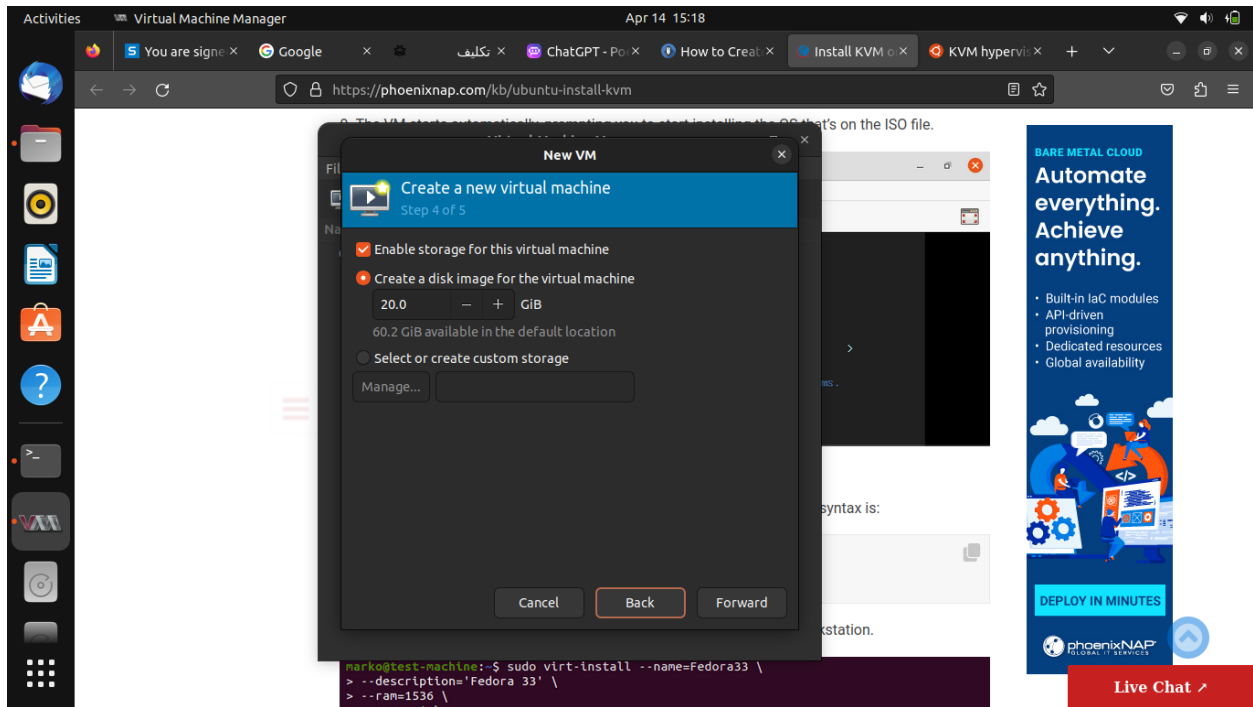
```
Activities Terminal Apr 14 15:09
mobi@mobi-X450CC: ~
mobi@mobi-X450CC:~$ sudo systemctl enable --now libvirtd
mobi@mobi-X450CC:~$
```

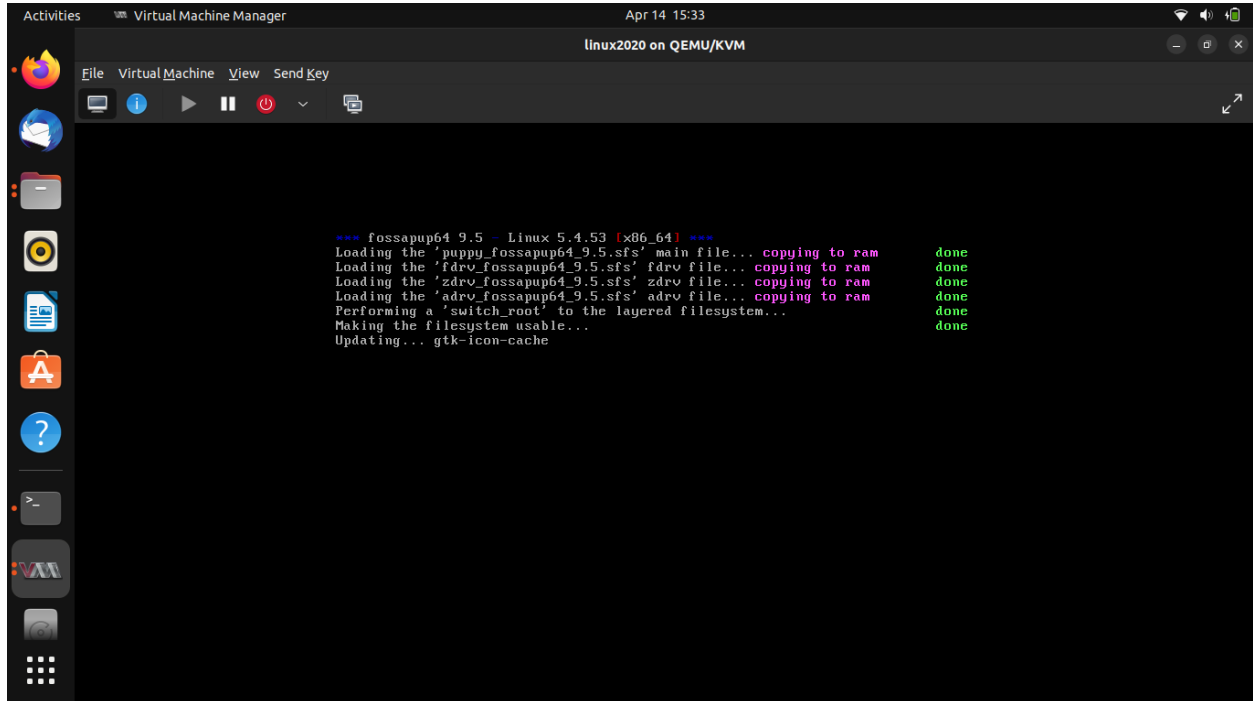
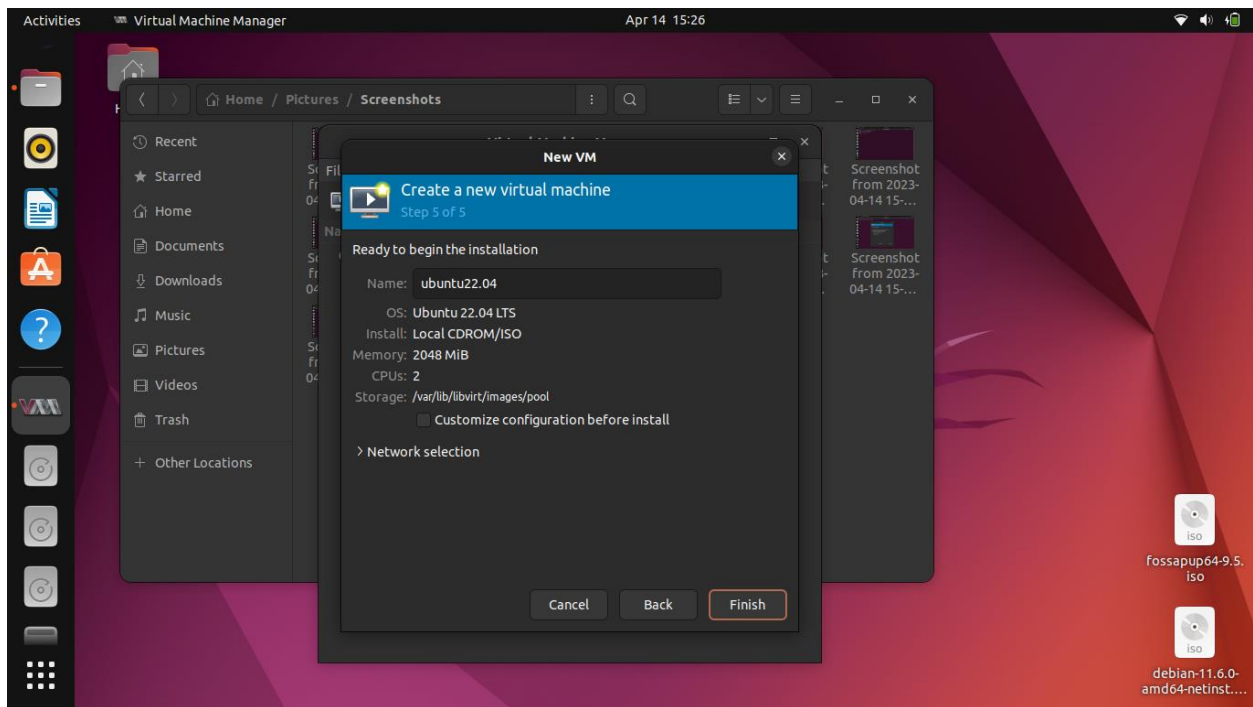
```
Activities Terminal Apr 14 15:11
mobi@mobi-X450CC: ~
Selecting previously unselected package libusbredirhost1:amd64.
Preparing to unpack .../07-libusbredirhost1.0.11.0-2build1_amd64.deb ...
Unpacking libusbredirhost1:amd64 (0.11.0-2build1) ...
Selecting previously unselected package libspice-client-glib-2.0-8:amd64.
Preparing to unpack .../08-libspice-client-glib-2.0-8.0.39-3ubuntu1_amd64.deb ...
Unpacking libspice-client-glib-2.0-8:amd64 (0.39-3ubuntu1) ...
Selecting previously unselected package gir1.2-spiceclientglib-2.0:amd64.
Preparing to unpack .../09-gir1.2-spiceclientglib-2.0.0.39-3ubuntu1_amd64.deb ...
Unpacking gir1.2-spiceclientglib-2.0:amd64 (0.39-3ubuntu1) ...
Selecting previously unselected package libva2:amd64.
Preparing to unpack .../10-libva2.2.14.0-1_amd64.deb ...
Unpacking libva2:amd64 (2.14.0-1) ...
Selecting previously unselected package libva-x11-2:amd64.
Preparing to unpack .../11-libva-x11-2.2.14.0-1_amd64.deb ...
Unpacking libva-x11-2:amd64 (2.14.0-1) ...
Selecting previously unselected package libspice-client-gtk-3.0-5:amd64.
Preparing to unpack .../12-libspice-client-gtk-3.0-5.0.39-3ubuntu1_amd64.deb ...
Unpacking libspice-client-gtk-3.0-5:amd64 (0.39-3ubuntu1) ...
Selecting previously unselected package gir1.2-spiceclientgtk-3.0:amd64.
Preparing to unpack .../13-gir1.2-spiceclientgtk-3.0.0.39-3ubuntu1_amd64.deb ...
Unpacking gir1.2-spiceclientgtk-3.0:amd64 (0.39-3ubuntu1) ...
Selecting previously unselected package libigdmm12:amd64.
Preparing to unpack .../14-libigdmm12.22.1.2+ds1-1_amd64.deb ...
Unpacking libigdmm12:amd64 (22.1.2+ds1-1) ...
Selecting previously unselected package intel-media-va-driver:amd64.
Preparing to unpack .../15-intel-media-va-driver.22.3.1+dfsg1-1ubuntu1_amd64.deb ...
Unpacking intel-media-va-driver:amd64 (22.3.1+dfsg1-1ubuntu1) ...
Selecting previously unselected package libburn4:amd64.
Preparing to unpack .../16-libburn4.1.5.4-1_amd64.deb ...
Unpacking libburn4:amd64 (1.5.4-1) ...
Selecting previously unselected package libgovirt-common.
Preparing to unpack .../17-libgovirt-common.0.3.8-1_all.deb ...
Unpacking libgovirt-common (0.3.8-1) ...
Selecting previously unselected package libgovirt2:amd64.
Preparing to unpack .../18-libgovirt2.0.3.8-1_amd64.deb ...
Unpacking libgovirt2:amd64 (0.3.8-1) ...
Progress: [ 26%] [#####.....]
```

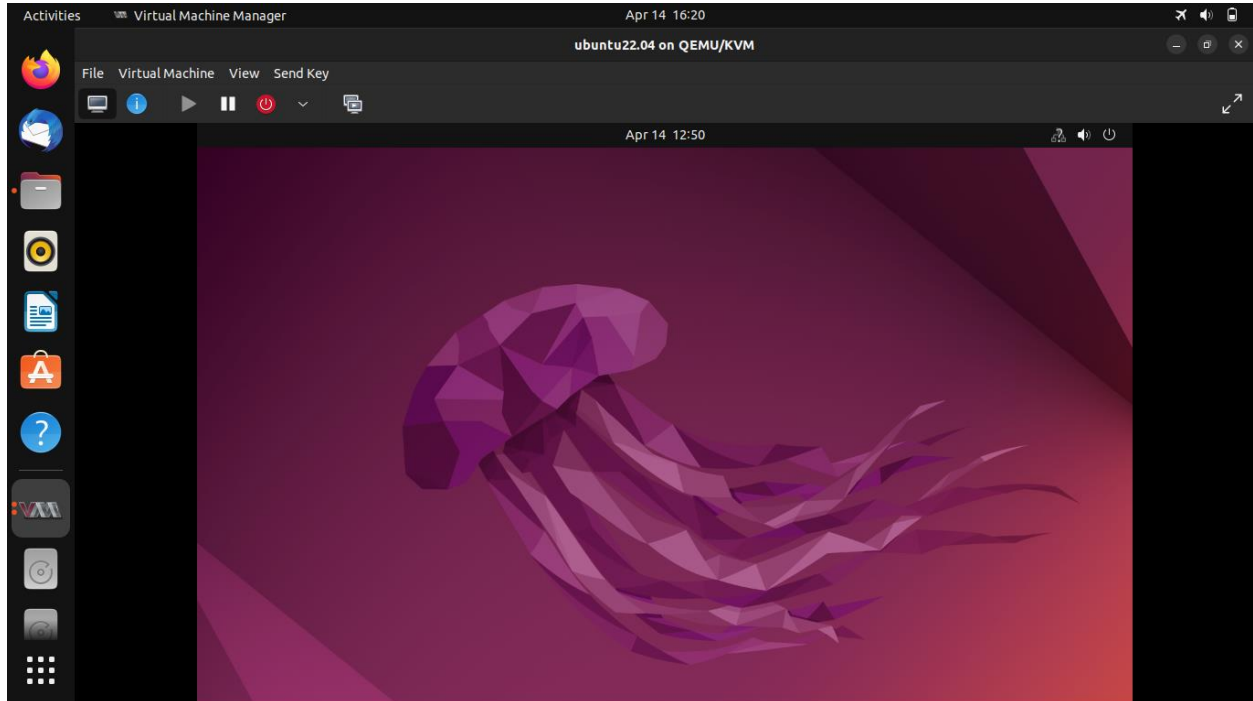
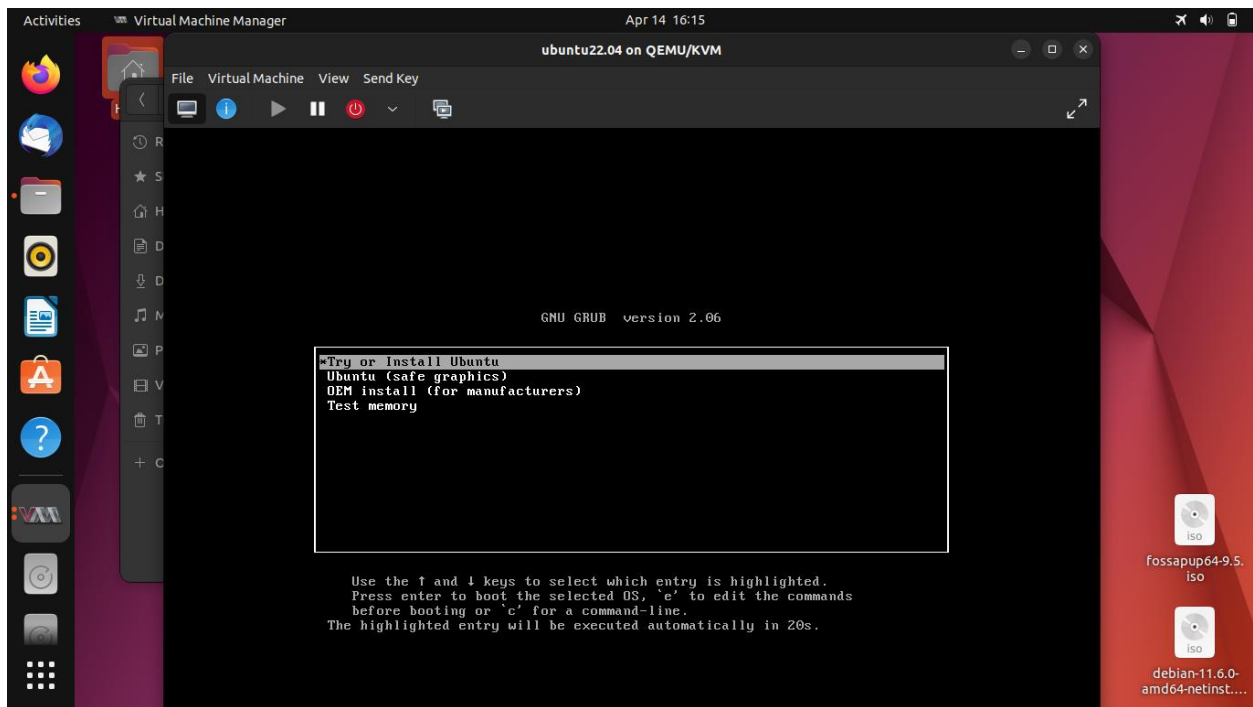
```
Activities Terminal Apr 14 15:12
mobi@mobi-X450CC: ~
Setting up intel-media-va-driver:amd64 (22.3.1+dfsg1-1ubuntu1) ...
Setting up osInfo-db (0.20220214-1ubuntu2.1) ...
Setting up libgovirt2:amd64 (0.3.8-1) ...
Setting up libvirt-glib-1.0-0:amd64 (4.0.0-2) ...
Setting up mesa-va-drivers:amd64 (22.2.5-0ubuntu0.1-22.04.1) ...
Setting up libburn4:amd64 (1.5.4-1) ...
Setting up libva-x11-2:amd64 (2.14.0-1) ...
Setting up gir1.2-ayatanaappindicator3-0.1 (0.5.90-7ubuntu2) ...
Setting up i965-va-driver:amd64 (2.4.1+dfsg1-1) ...
Setting up libgtk-vnc-2.0-0:amd64 (1.3.0-1ubuntu1) ...
Setting up libphodav-2.0-0:amd64 (2.5-1) ...
Setting up libisofs6:amd64 (1.5.4-1) ...
Setting up gir1.2-libvirt-glib-1.0:amd64 (4.0.0-2) ...
Setting up libosinfo-1.0-0:amd64 (1.8.0-1) ...
Setting up va-driver-all:amd64 (2.14.0-1) ...
Setting up gir1.2-gtk-vnc-2.0:amd64 (1.3.0-1ubuntu1) ...
Setting up libisoburn1:amd64 (1.5.4-2) ...
Setting up xorriso (1.5.4-2) ...
Setting up gir1.2-libosinfo-1.0:amd64 (1.8.0-1) ...
Setting up libspice-client-glib-2.0-8:amd64 (0.39-3ubuntu1) ...
Setting up virtinst (1:4.0.0-1) ...
Setting up gir1.2-spiceclientglib-2.0:amd64 (0.39-3ubuntu1) ...
Setting up virt-manager (1:4.0.0-1) ...
Setting up libspice-client-gtk-3.0-5:amd64 (0.39-3ubuntu1) ...
Setting up gir1.2-spiceclientgtk-3.0:amd64 (0.39-3ubuntu1) ...
Setting up virt-viewer (7.0-2build2) ...
Processing triggers for mailcap (3.70+nmulubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for libgl1.0-0:amd64 (2.72.4-0ubuntu1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for install-info (6.8-4build1) ...
mobi@mobi-X450CC: $
```

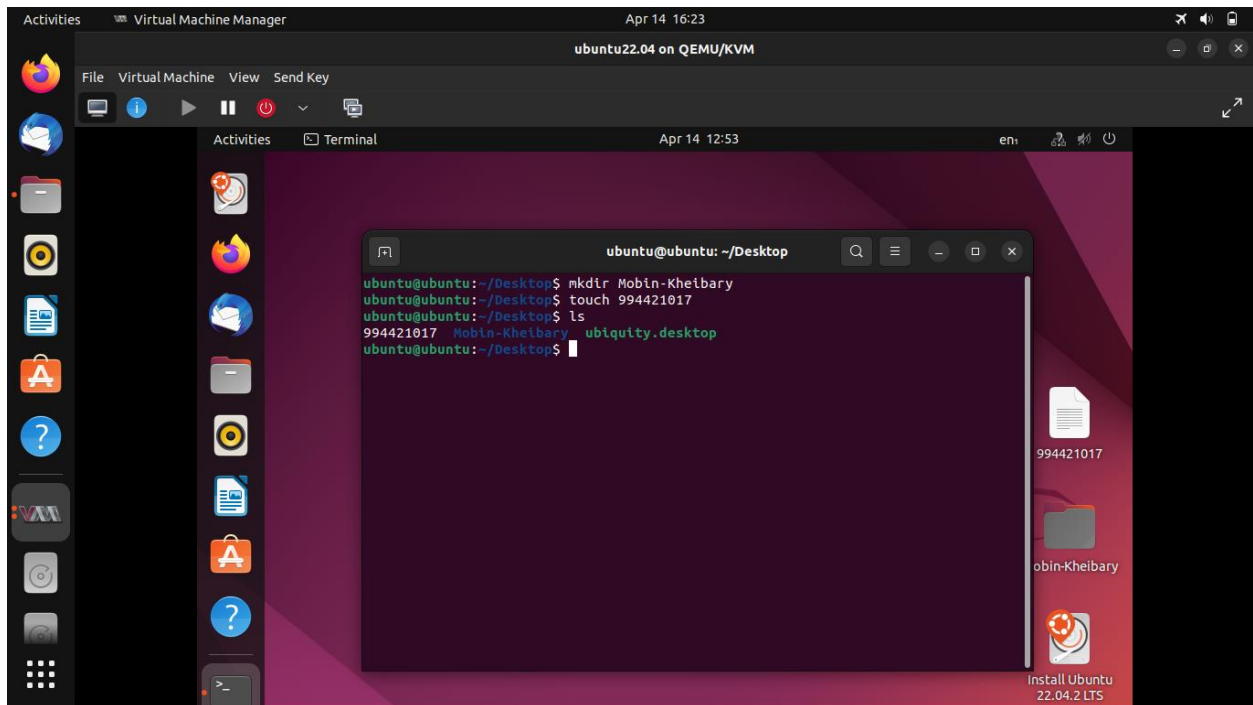
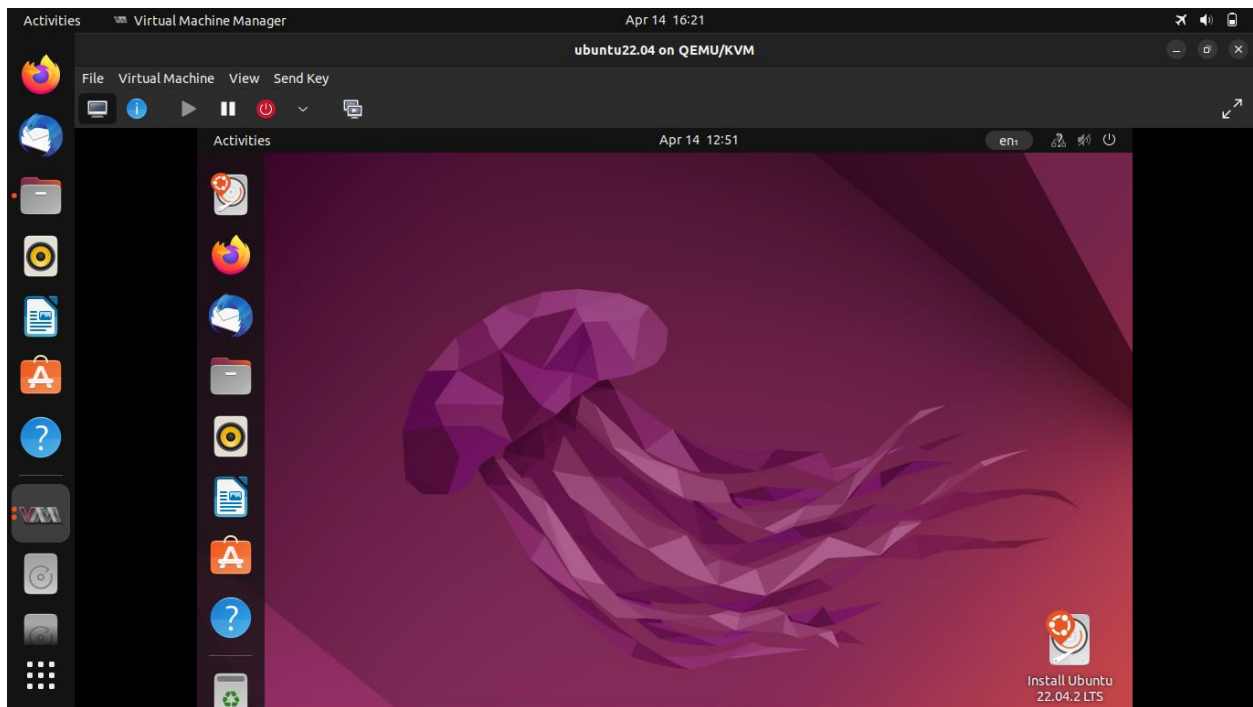












منابع استفاده شده جهت تهیهی این گزارش:

- i. <https://linuxconfig.org/how-to-create-and-manage-kvm-virtual-machines-from-cli>
- ii. <https://www.tecmint.com/create-virtual-machines-in-kvm-using-virt-manager/>
- iii. <https://phoenixnap.com/kb/ubuntu-install-kvm>
- iv. <https://www.linuxtechi.com/create-manage-kvm-virtual-machine-cli/>
- v. <https://www.geeksforgeeks.org/how-to-create-virtual-machines-in-linux-using-kvm-kernel-based-virtual-machine/>
- vi. <https://brewyourtech.com/how-to-create-virtual-machines-kvm-ultimate-guide/>
- vii. <https://serverhealers.com/blog/create-virtual-machines-in-linux-using-kvm>
- viii. <https://www.mobinhost.com/mag/kvm-virtualization/>
- ix. <https://technet24.ir/%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D8%A8%D8%B3%D8%AA%D8%B1-%D9%85%D8%AC%D8%A7%D8%B2%DB%8C-%D8%B3%D8%A7%D8%B2%DB%8C-%D9%85%D8%A8%D8%AA%D9%86%DB%8C-%D8%A8%D8%B1-%D9%84%DB%8C%D9%86%D9%88-8741>
- x. <https://technet24.ir/%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D8%A8%D8%B3%D8%AA%D8%B1-%D9%85%D8%AC%D8%A7%D8%B2%DB%8C-%D8%B3%D8%A7%D8%B2%DB%8C-%D9%85%D8%A8%D8%AA%D9%86%DB%8C-%D8%A8%D8%B1-%D9%84%DB%8C%D9%86%D9%88-8741>
- xi. <https://ubuntu.com/blog/kvm-hypervisor>

پایان.