

CONSEJO TÉCNICO

El software Command Recording sobre el sistema operativo Linux: un tutorial

Introducción

A partir de la versión 6.1, ahora es posible instalar el software Command Recording (CRS) de March Networks en máquinas con el sistema operativo Linux. Esta nueva y altamente escalable versión de Linux de nuestro software de gestión de video (VMS) soporta hasta 3,000 cámaras en un solo servidor. Esta escalabilidad es ideal para grandes sedes corporativas y otras empresas en crecimiento. Con esta solución, los clientes pueden reducir los costos de infraestructura utilizando menos servidores. También pueden migrar al almacenamiento centralizado al transmitir el video de las cámaras a un solo servidor del centro de datos in situ.

El sistema operativo Linux generalmente se considera una mejor alternativa que el sistema operativo Windows en términos de ciberseguridad, mantenimiento y longevidad. Por otro lado, conocer y manejar bien los procedimientos y metodología de Linux no es tan común en la industria de la seguridad y el primer encuentro puede ser frustrante para el usuario sin experiencia. Este documento proporciona algunas directrices sobre cómo configurar un sistema básico para probar nuestra aplicación CRS. La configuración completa de un servidor Linux es una tarea que requiere habilidades avanzadas y solamente debe gestionarse por un usuario con experiencia.

Este documento es una guía para realizar la instalación estándar de CRS para Linux, comenzando con la instalación del sistema operativo y la configuración básica. Utilizaremos el enfoque de virtualización, ejecutando Linux en un ambiente virtualizado, siendo este el primer paso que un usuario debe tomar para probar la aplicación sin la necesidad de una máquina dedicada.

Prerequisitos

El proyecto VirtualBox, adquirido por Oracle, es un virtualizador completo de uso general para el hardware x86 y es la herramienta que utilizaremos para lograr nuestro objetivo. El instalador de VirtualBox puede descargarse gratis del sitio web oficial: <https://www.virtualbox.org/>

Esta herramienta es equivalente a VMWare y puede crear una máquina virtual (VM) en una PC anfitriona. Lo más común es que la PC anfitriona utilice el sistema operativo Windows (esto se conoce como el sistema operativo anfitrión). En este caso será necesario seleccionar el instalador Windows.

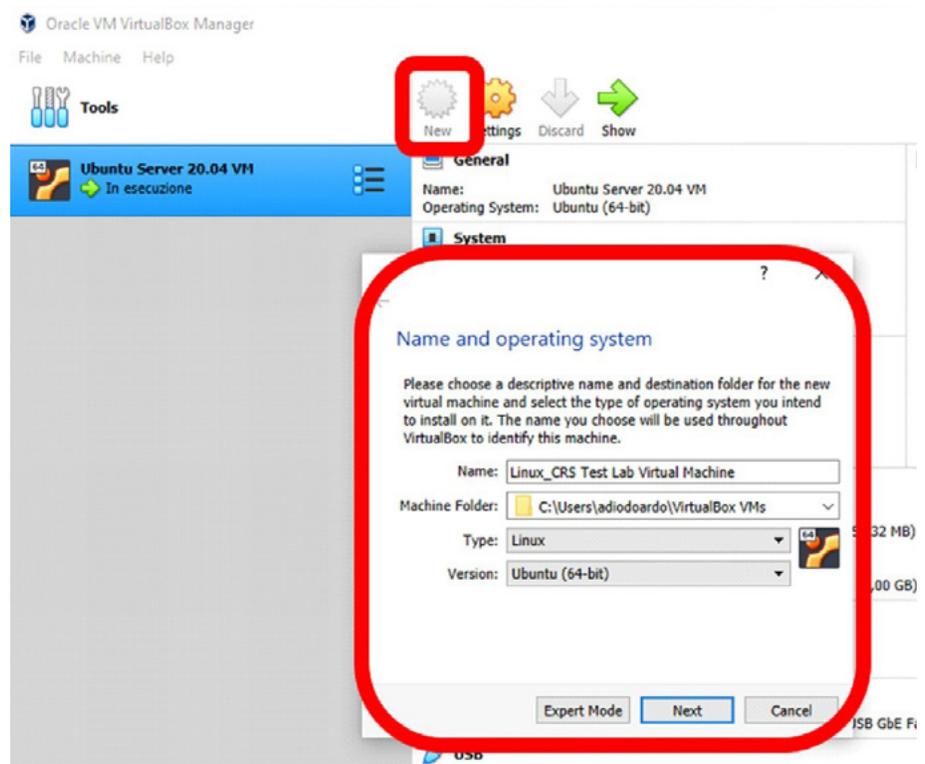
Una vez instalado en una PC Windows, es posible lanzar la interfaz del manejador Virtual Box donde se puede crear y gestionar el VMS.

En este punto, para crear una máquina virtual Ubuntu (en este caso Ubuntu es el sistema operativo invitado) es necesario descargar la imagen ISO del sistema operativo en cuestión: Ubuntu Server 20.04 versión LTE es una opción razonable para probar el CRS en Linux (por favor tome en cuenta que el CRS en Linux solamente soporta el sistema operativo de 64-bits). Se puede descargar gratis del sitio web oficial: <https://ubuntu.com>

Una vez completada en la interfaz VBox, se puede seleccionar “New” para crear una nueva VM. En el primer panel emergente se puede especificar el nombre — una carpeta de destino para almacenar todos los archivos y propiedades básicas de la VM (“Type” y “Version” reflejarán el sistema operativo invitado que deseamos instalar).



Figura 1: En la interfaz de gestión de Virtual Box, una vez que se haya seleccionado “New”, una herramienta asistente guiará al usuario para crear la máquina virtual.



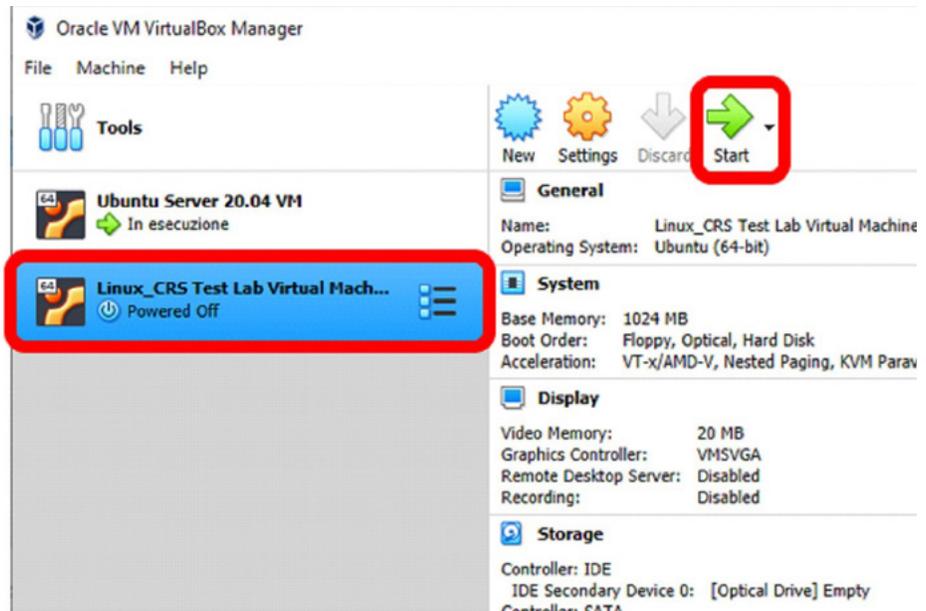
Hacer clic en “Next” nos permitirá decidir cuánta memoria deseamos asignar a la VM: ésta debería ser menos del 50 % del total de memoria RAM disponible en la PC anfitriona. Como se trata de un entorno de prueba, podríamos asignar solo 2 GB (si es necesario este valor se puede modificar más tarde). Se deberían hacer consideraciones similares para el número de núcleos del procesador que se asignarán a la VM: deben asignarse un mínimo de 2 núcleos a la nueva VM.

El siguiente paso se refiere al disco duro virtual que se usará para almacenar el sistema operativo invitado: sugerimos que utilice uno de 20 GB como mínimo, y deje la opción VDI por defecto y “Dynamically allocated” por defecto para reducir el tamaño inicial de la VM en la máquina anfitriona. Una vez que se asigne un nombre al disco duro, el sistema creará la estructura virtual según las opciones seleccionadas.

En este punto, después de algunos segundos, la nueva VM aparecerá en el panel izquierdo de la interfaz de gestión. Una vez seleccionada, se podrá iniciar la VM que acabamos de crear.



Figura 2: Hacer clic en "Start" ejecutará la VM que se creó vacía. Ahora es el momento de cargar la imagen ISO del sistema operativo.



El primer panel pedirá la imagen ISO que se usará para instalar el sistema operativo invitado. Escogeremos la imagen ISO de Ubuntu que descargamos y comenzaremos con la instalación del sistema operativo. A partir de ahora, la instalación será exactamente igual que la de un servidor físico. Necesitaremos seleccionar todos los ajustes estándar para el nuevo sistema operativo y asegurarnos de escoger la interfaz del teclado correcta (el servidor Ubuntu es solamente un sistema operativo para terminal y si se tiene el teclado incorrecto puede ser problemático). Como parte del procedimiento hay que crear un usuario para acceder al sistema: tome nota de la información de acceso del usuario para uso futuro. También es importante seleccionar un paquete docker durante el procedimiento ya que se necesitará para lanzar el CRS (en caso de no hacerlo, es posible instalar manualmente el paquete más tarde).

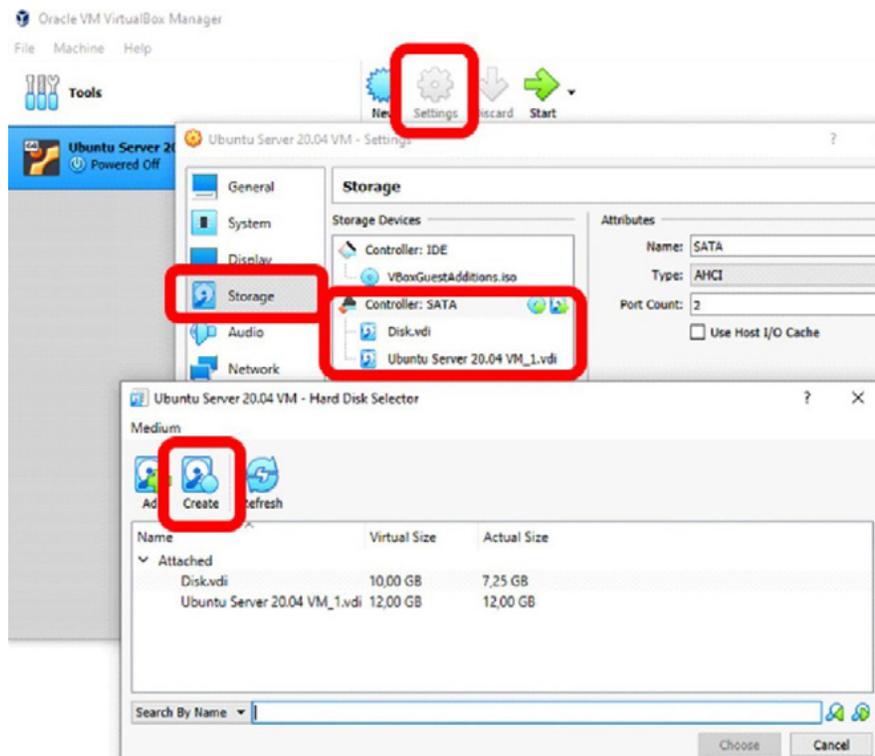
Al final del procedimiento, es posible que haya que esperar que el sistema actualice todos los paquetes o bien puede ignorar este paso y hacerlo más tarde. Ahora ya tenemos una máquina virtual ejecutando el sistema operativo Ubuntu Server.



Configuración de la máquina virtual

Antes de inicializar la máquina virtual Ubuntu que acabamos de crear, agregaremos un disco duro adicional para utilizarlo como almacenamiento dedicado. En la interfaz de gestión, abra los ajustes para seleccionar el almacenamiento y crear un nuevo disco duro bajo el controlador SATA.

Figura 3: Para agregar un disco duro a la máquina virtual se requiere abrir el panel de ajustes y pasar a la sección de "Storage".



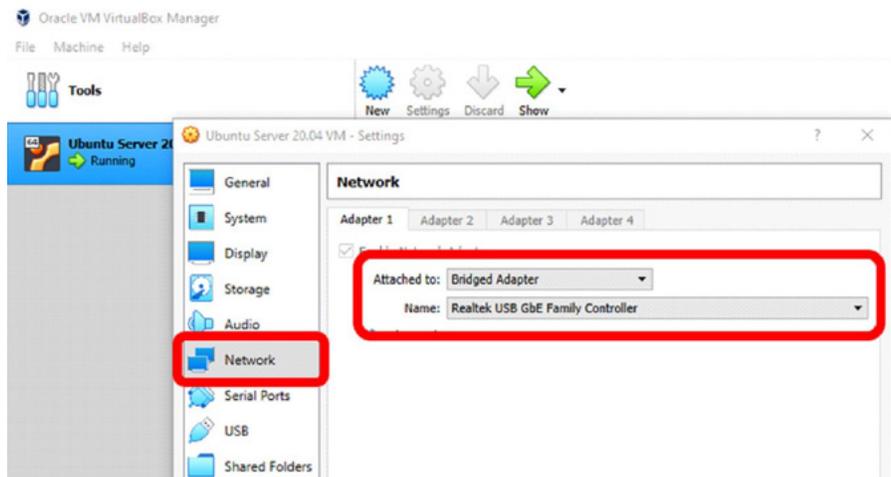
Dejaremos el formato VDI por defecto para el nuevo disco duro y seleccionaremos "Fixed Size", ya que ofrece un rendimiento superior. Una vez inicializado por el CRS, usaremos el tamaño completo (escoja un valor compatible con el almacenamiento disponible en la PC anfitriona).

El último paso antes de dirigirse a la consola de terminal Linux es asegurarnos de que podremos asignar una dirección IP válida a la máquina virtual, ya que probablemente necesitemos conectar las cámaras y la PC cliente una vez que hayamos terminado la configuración.

Asumiendo que está disponible una red de laboratorio con un servidor DHCP, navegaremos a la pestaña de "Network" y seleccionaremos "Bridged Adapter".



Figura 4: "Bridged Adapter" garantizará que el mismo servicio DHCP utilizado por la PC anfitriona también asigne una dirección IP válida a la VM.



Por supuesto, es necesario seleccionar el controlador conectado a la red local que alberga el servicio DHCP.

Este es el último paso preliminar. Ahora es el momento de iniciar Ubuntu pulsando el botón de Inicio.

Configuración del sistema operativo

Una vez que haya iniciado la sesión, el primer paso (suponiendo que se omitió durante la instalación del sistema operativo) es actualizar todos los paquetes con el siguiente comando:

```
sudo apt-get upgrade
```

Una vez que termine, se recomienda crear una carpeta CRS para guardar los archivos de instalación del CRS y una carpeta de datos CRS donde el instalador pondrá todos los datos del programa (configuración, registros, etc.). Crearemos dichas carpetas en la sección de inicio con el comando:

```
sudo mkdir CRS
```

```
sudo mkdir CRS_data
```

Una vez hecho esto, lo verificaremos con:

```
ls -la
```

para confirmar que las carpetas se crearon.

```
diodo@diodovm:~$ ls -la
total 44
drwxr-xr-x 6 diodo diodo 4096 Jan 26 13:36 .
drwxr-xr-x 3 root root 4096 Jan 25 11:40 ..
-rw----- 1 diodo diodo 1847 Jan 26 18:09 .bash_history
-rw-r--r-- 1 diodo diodo 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 diodo diodo 3771 Feb 25 2020 .bashrc
drwx----- 2 diodo diodo 4096 Jan 25 12:50 .cache
drwxr-xr-x 2 diodo diodo 4096 Jan 26 16:16 CRS
drwxrwxr-x 3 diodo diodo 4096 Jan 26 14:32 CRS_data
-rw-r--r-- 1 diodo diodo 807 Feb 25 2020 .profile
drwxr-xr-x 3 diodo diodo 4096 Jan 25 17:06 snap
-rw-r--r-- 1 diodo diodo 0 Jan 25 17:06 .sudo_as_admin_successful
-rw----- 1 diodo diodo 847 Jan 26 13:36 .viminfo
diodo@diodovm:~$
```

Figura 5: "Los directorios se muestran en azul. Utilizar la opción "-la" proporciona detalles adicionales al comando básico "ls".



También crearemos una carpeta donde montaremos el disco duro de almacenamiento. Crearemos una carpeta dedicada bajo “/media” con el siguiente comando:

```
sudo mkdir /media/CRS_Storage
```

Ahora es el momento de montar el disco duro de almacenamiento en la carpeta recién creada.

Es necesario identificar el nombre del dispositivo asignado por Linux al disco duro de almacenamiento. Utilizaremos el siguiente comando:

```
sudo fdisk -l |less
```

Al pulsar el botón se desplazará a la siguiente página (sin “|less” podría perder la salida inicial del comando - pulse q al final de la lista). Necesita identificar un disco mapeado en “/dev/sdb” que necesita ser inicializado. Este disco debe ser del mismo tamaño que especificó anteriormente. A continuación, ejecutaremos el comando:

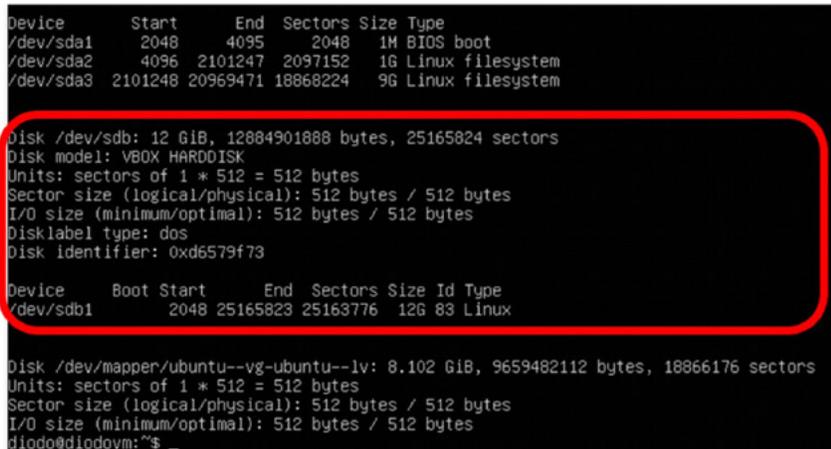
```
sudo fdisk /dev/sdb
```

Esto abrirá el procedimiento para crear una nueva partición en el disco. Seleccione “n” para la nueva partición, “p” para la partición primaria y deje los demás ajustes por defecto. Pulse “w” para guardar todos los ajustes en el disco. Suponiendo que el procedimiento tuvo éxito, podemos intentarlo de nuevo con:

```
sudo fdisk -l |less
```

Esto verificará que una nueva sección fue creada.

Figura 6: La partición del disco fue creada y ahora aparece como la salida del comando fdisk.



```
Device      Start      End  Sectors  Size Type
/dev/sda1   2048      4095    2048    1M BIOS boot
/dev/sda2   4096    2101247 2097152   1G Linux filesystem
/dev/sda3  2101248 20969471 18868224   9G Linux filesystem

Disk /dev/sdb: 12 GiB, 12884901888 bytes, 25165824 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd6579f73

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdb1                   2048 25165823 25163776   12G 83 Linux

Disk /dev/mapper/ubuntu--vg-ubuntu--lv: 8.102 GiB, 9659482112 bytes, 18866176 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
dlodo@dlodovm:~$ _
```

Ahora es posible formatear la partición que acabamos de crear. Utilizaremos el sistema de archivos “xfs”:

```
sudo mkfs.xfs /dev/sdb1
```

Ahora debemos asegurarnos de que esta partición se monta automáticamente en el momento de arranque. Tendremos que editar el archivo de configuración “fstab” en la carpeta “etc”. Antes de aplicar cualquier cambio a este archivo, crearemos una copia de respaldo:

```
sudo cp /etc/fstab /etc/fstab.backup
```

Luego, usando vi (o cualquier otro editor de texto, por ejemplo, nano) añadiremos dos líneas al final del archivo “fstab”:

```
sudo vi /etc/fstab
```



Con la ayuda de las flechas, hay que posicionar el cursor en la última línea. En este punto, al pulsar la letra "o" se añadirá una nueva línea después de la posición del cursor y el modo cambiará a INSERT. Ahora se podrá escribir el texto requerido (la primera línea es sólo un comentario, en la segunda línea use TAB y no ESPACIO para separar cada campo):

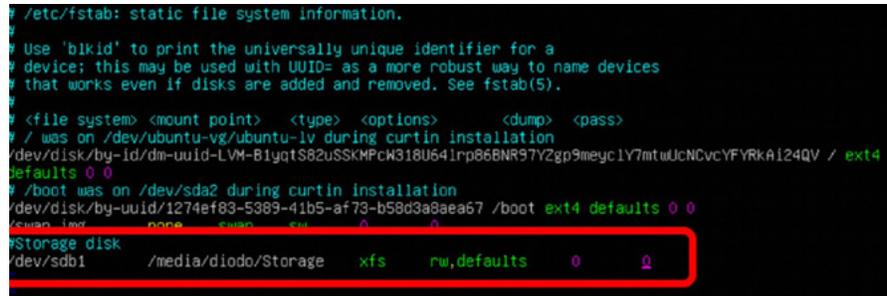
```
#CRS Storage disk
/dev/sdb1/media/CRS_Storage xfs,rw,defaults 0 0
```

Para guardar y cerrar el archivo, pulse ESC para salir del modo INSERT luego teclee:

```
:wq!
```

para salir del editor.

Figura 7: Tendremos que añadir dos líneas (la primera es sólo un comentario) para que la nueva partición se monte en el arranque.



```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# file system  mount point  type  options  dump  pass
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-uuid/1274ef83-5389-41b5-af73-b58d3a8aea67 /boot ext4 defaults 0 0
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/1274ef83-5389-41b5-af73-b58d3a8aea67 /boot ext4 defaults 0 0
# swap_lvm none swap su 0 0
#Storage disk
/dev/sdb1 /media/diodo/Storage xfs rw,defaults 0 0
```

Ahora es posible montar por fin el disco con el comando :

```
sudo mount -a
```

Para verificar el procedimiento, utilice el comando:

```
sudo cat /etc/mtab | less
```

A continuación, busque una lista de líneas /dev/sdb1.

Estamos casi al final del procedimiento, pero antes de instalar el CRS, se recomienda reiniciar la máquina para actualizar todos los ajustes con el comando :

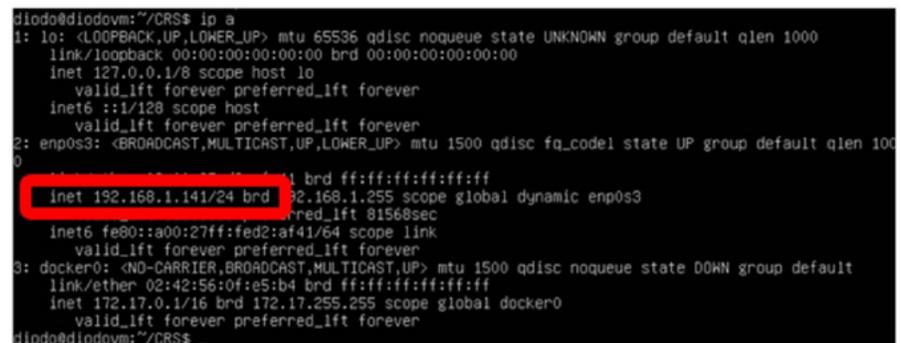
```
sudo shutdown -P now
```

Una vez que la máquina reinicie, tenemos que copiar los archivos de instalación a la PC Ubuntu. Hay varias formas de transferir archivos entre una PC con Windows y una con Linux. Utilizaremos una herramienta de terceros llamada WinSCP, que se descarga gratis de Internet, para pasar rápidamente los archivos entre el sistema anfitrión y el invitado.

Para que esto funcione, hay que descubrir la dirección IP de la máquina VM. Utilizaremos el comando:

```
ip a
```

Figura 8: Es necesario descubrir la dirección IP para conectarse al sistema y transferir los archivos.

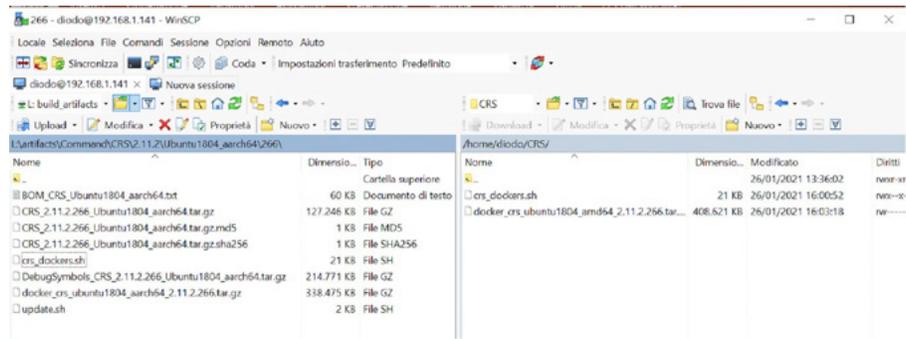


```
diodo@diodovm:~/CRS$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:42:56:0f:e5:b4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.141/24 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 81568sec preferred_lft 81568sec
    inet6 fe80::a00:27ff:fed2:af41/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:56:0f:e5:b4 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
diodo@diodovm:~/CRS$
```



Figura 9: Los archivos de instalación de CRS se componen de un script .sh y una imagen docker del CRS que necesitan copiarse localmente a la carpeta “/home/CRS” en la máquina Linux.

Una vez que se establezca la nueva conexión (utilizando el protocolo SCP y la información del usuario de Linux) transferiremos los archivos requeridos.

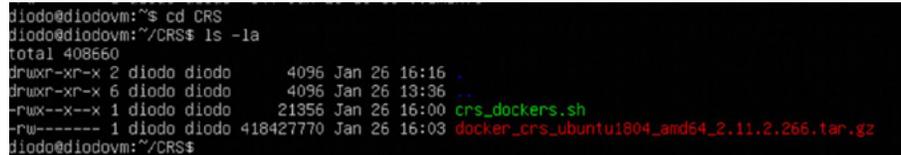


En la consola Linux, hay que hacer que el script sea ejecutable con el comando:

```
sudo chmod a+x crs_dockers.sh
```

Utilizando de nuevo el comando “ls-la”, es posible verificar que el script aparecerá en verde y las capacidades incluirán la x (ejecutar).

Figura 10: Ahora es posible ejecutar el script una vez que se ha hecho ejecutable con el comando “chmod”.



Ahora estamos listos para ejecutar el script para crear la primera instancia docker del CRS. Le recomendamos que pruebe una sola instancia la primera vez para asegurarse de que el sistema está bien configurado (probablemente será necesario aumentar la memoria y los núcleos del procesador si se están ejecutando varias instancias en paralelo).

Ejecución de nuestra primera instancia CRS

Dependiendo de la versión del script, la sintaxis podría ser ligeramente diferente. Se recomienda consultar la ayuda en pantalla embebida para ver la lista de las opciones disponibles. Como referencia, en el caso de la versión 2.11.2, la línea de comandos para crear una única instancia de docker (en la zona horaria italiana utilizando las carpetas de almacenamiento y datos previamente creadas) sería la siguiente:

```
sudo ./crs_dockers.sh -C -r 1 -i docker_crs_ubuntu1804_
amd64_2.11.2.266.tar.gz -f /home/ diodo/CRS_data -t
Europe/Rome -s /media/CRS_Storage
```

Con los comandos docker, es posible verificar si se ha creado correctamente la instancia crs1 y si está funcionando correctamente.



Figura 11: `sudo docker ps` mostrará las instancias activas que se están ejecutando en la máquina. Ponga especial atención a la columna STATUS.

```
diodo@diodovm:~/CRS$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS
e4b9ae927971  crs:current  "/bin/sh -c ./run.sh"   3 hours ago    Up 53 minutes
```

Figura 12: `sudo docker logs` listará todos los registros generados por cada instancia. Aquí es donde hay que comenzar si el sistema falla.

```
diodo@diodovm:~/CRS$ sudo docker logs
"docker logs" requires exactly 1 argument.
See 'docker logs --help'.

Usage: docker logs [OPTIONS] CONTAINER

Fetch the logs of a container
diodo@diodovm:~/CRS$ sudo docker logs crs1
Starting CRS Docker container...
- server_data loading...
- reg_data loading...
- enabling core dumps generation (requires "/var/crash/core_%.e_%.p" set from host)
Starting CRS...
./plugins/libUbuntu1804_amd64_DVR.so: undefined symbol: _ZN14HSCCMInterface8SendTextEiPKci
./plugins/libUbuntu1804_amd64_DVR.so: undefined symbol: _ZN14HSCCMInterface8SendTextEiPKci
Starting CRS Docker container...
- server_data loading...
- reg_data loading...
- enabling core dumps generation (requires "/var/crash/core_%.e_%.p" set from host)
Starting CRS...
./plugins/libUbuntu1804_amd64_DVR.so: undefined symbol: _ZN14HSCCMInterface8SendTextEiPKci
./plugins/libUbuntu1804_amd64_DVR.so: undefined symbol: _ZN14HSCCMInterface8SendTextEiPKci
diodo@diodovm:~/CRS$ _
```

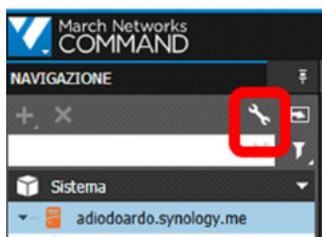
En este punto, es posible lanzar el Command Client en la máquina anfitriona (o cualquier otra estación de trabajo en la misma red local) y abrir la dirección IP en Linux especificando el puerto 44301 para el protocolo HTTPS.

Figura 13: Por defecto, el usuario admin no tiene contraseña. El puerto https es 44301 y se puede cambiar durante la creación de docker.



Al seleccionar el nombre del servidor, es posible lanzar el configurador del sistema y, como primer paso, registrar la grabadora en el CES para obtener la licencia. Por su diseño, esta es la única manera de licenciar CRS en Linux.

Figura 14: Una forma rápida de lanzar la interfaz de configuración de CRS es haciendo clic en la herramienta desde la interfaz cliente.



A partir de ahora, el sistema se comportará exactamente como la versión más tradicional de Windows y es posible consultar los manuales y el material de formación en línea para obtener las instrucciones sobre cómo configurar la grabadora



Para más detalles, por favor póngase en contacto con el equipo de ingenieros de ventas de March Networks. Ellos trabajarán con usted para identificar la mejor opción para las necesidades específicas de su negocio.

Atención y soporte al cliente

Norteamérica, Sudamérica, y Australia

Directo: 1.613.591.1441

Número gratuito (EE.UU. y Canadá):

1.800.472.0116

Correo electrónico:

techsupport@marchnetworks.com

Europa, Medio Oriente y África (EMEA)

Directo: +39 0362 17935 ext. 3

Correo electrónico:

supporteurope@marchnetworks.com

Conclusión

El CRS es un software muy potente y flexible que se puede instalar en muchas plataformas diferentes de hardware para adaptarse al tamaño, el costo y el rendimiento deseados. Tradicionalmente, esto sólo ha sido posible utilizando sistemas operativos basados en Windows. Con la versión 6.1, ahora también soportamos el sistema operativo Linux y la tecnología de contenedores Docker. Esto les permitirá a los usuarios escalar la solución para gestionar hasta varios miles de cámaras con la misma máquina física o virtual (simplemente creando múltiples instancias de la misma imagen CRS). También puede instalar el CRS en su Raspberry para tener un sistema embebido con la tecnología de March Networks en un factor de forma pequeño. Por favor, póngase en contacto con su representante de ventas de March Networks para más detalles sobre cómo iniciar CRS en Linux.

Generalidades de la compañía

March Networks® ayuda a las organizaciones a transformar el video en inteligencia de negocios a través de la integración de la videovigilancia, las analíticas, los datos de los sistemas de información del negocio y los dispositivos IoT. Las empresas de todo el mundo utilizan nuestras soluciones de software para mejorar la eficiencia y el cumplimiento normativo, reducir las pérdidas y los riesgos, mejorar el servicio al cliente y ser más competitivas. Con una presencia consolidada en la seguridad por video y redes, March Networks también es reconocida como líder en gestión de video escalable de clase empresarial y servicios alojados. Estamos orgullosos de trabajar con muchas de las instituciones financieras, comercios, operadores de cannabis y autoridades de transporte más grandes del mundo. Ofrecemos nuestro software y sistemas a través de una extensa red de distribuidores y socios en más de 70 países. Fundada en el 2000, March Networks tiene sus oficinas centrales en Ottawa, Ontario, Canadá. Para más información, por favor visite

www.marchnetworks.com.



Norteamérica	1 800 563 5564
Latinoamérica.....	+5255 5259 9511
Europa	+39 0362 17935
Asia	+65 6818 0963
Australia y Nueva Zelanda.....	+61 1300 089 419
Oriente Medio y África	+971 4 399 5525

© 2021. March Networks. Todos los derechos reservados. La información contenida en este documento está sujeta a modificaciones sin previo aviso. MARCH NETWORKS, March Networks Command, March Networks Searchlight, March Networks RideSafe y el logotipo de MARCH NETWORKS son marcas comerciales de March Networks Corporation. Todas las demás marcas comerciales son propiedad de sus respectivos dueños. 060-3462-SP-A
marchnetworks.com

