

# Converting a VMware Linux Guest to HyperV

There are some situations where we fail to move Linux based Guest VMs to Hyper V from VMware. If you haven't figured it out already, the basic problem here is that the Linux VMs (that came from the VMware environment) don't have the Hyper-V drivers configured because they weren't needed at installation (again, on VMware), but the Installation ISOs *do* have the drivers at the ready when booting the "rescue system".

The below article will provide you the correct way to carry out the tasks;

## Step 1: Install MVMC2

First, the installation. Go to the [MVMC2](#) download site, get the software, and install.

## Step2: Convert the Disk

Open Powershell *as Administrator* (right-click, run as Administrator), and load the module:

```
Import-Module "C:\Program Files\Microsoft Virtual Machine Converter\MvmcCmdlet.psdl"
```

We need to create a temporary folder for our converted disk – in my case on a separate drive from where my source VMs reside, to speed things up a tad:

```
md c:MvmcResults
```

```
cd c:MvmcResults
```

And then we can convert our source disk:

```
ConvertTo-MvmcVirtualHardDisk -SourceLiteralPath "E:\Virtual Machines\Opensuse12.30\Opensuse12.3.vmdk"
```

Next, move the newly converted disk to wherever you keep your virtual hard disks (I assume you have a designated location).

Then, create a Generation 1 Virtual Machine in Hyper-V (try to use the same name, memory settings, and so on as before), but choose to "Use an existing

virtual hard disk” and set it to the newly converted hard disk. But before you start it, attach the correct installation ISO (I use the tiny “network install” ISO).

### **Step 3: Boot and Mount the Alternate Root**

Start the VM (booting from ISO), and choose “Rescue System” as the boot choice. Tip: while the splash screen is up, hit the escape key and notice the Hyper-V drivers it chose; in my case it was only hv\_netvsc and hv\_storvsc, but you may have others.

Once at the “Rescue” prompt, enter “root” as the login.

Now, mount the proper disk partition for your root filesystem; this may take some guessing if you don’t remember which is which. If you don’t know, you might have to mount a few of them and look and see what’s in them. And of course, if you have a separate /boot, you’ll have to mount that too. But for the examples that follow, we’ll assume the full root file system is all on /dev/sda2.

So mount the root filesystem under the alternate mount point, like this example:

```
# mount the root
mount /dev/sda2 /mnt
# you may have to mount /boot too, depending on your setup
# mount /dev/sda1 /mnt/boot
# you must re-mount the live dev and proc
mount --rbind /dev /mnt/dev
mount --rbind /proc /mnt/proc
# set the chroot environment
chroot /mnt
...and then we’re ready to actually do some fixing.
```

### **Step 4: Fix the Modules**

Use *vi* to edit */etc/sysconfig/kernel* to include the Hyper-V modules

```
INITRD_MODULES="mptspi ata_piix ata_generic vmxnet3 vmw_pvscsi vmxnet"
```

...or perhaps like this (this example does not have VMtools drivers):

```
INITRD_MODULES="mptspi ata_piix ata_generic"
```

...and with your change, you're making it look more like this (again, we're only adding the hv\_ modules to the end):

```
INITRD_MODULES="mptspi ata_piix ata_generic vmxnet3 vmw_pvscsi vmxnet hv_vmbus  
hv_netvsc hv_storvsc"
```

And finally, recreate the initrd with something similar to the following command (this example is taken from one of my older ones). The kernel and initrd specified in the command must match your current kernel the machine boots with.

```
mkinitrd -k /boot/vmlinuz-3.7.10-1.11-desktop.gz -i /boot/initrd-3.7.10-1.11-  
desktop
```

...and you're done! Type "exit" to end the chroot environment, and "init 0" to shut down. Go to the settings of the VM, and detach the ISO, and boot it up.

## **Hyper-V On Ubuntu 16.04**

The first step is to edit the "modules" file located in /etc/initramfs-tools using the below command:

```
nano /etc/initramfs-tools/modules
```

Once in nano, navigate to the last line in the file and enter the following lines:

```
hv_vmbus  
hv_storvsc  
hv_blkvsc  
hv_netvsc
```

Save the file by hitting the ctrl + x and then issuing "y" for yes and then enter. Next, run the following commands to re-initialize the "modules" file, install the virtual tools and reboot the machine:

```
apt-get install --install-recommends linux-tools-virtual-lts-xenial linux-cloud-  
tools-virtual-lts-xenial  
update-initramfs -u  
reboot
```

### **Hyper-V On Ubuntu 18.04**

```
nano /etc/initramfs-tools/modules
```

```
hv_vmbus
```

```
hv_storvsc
```

```
hv_blkvsc
```

```
hv_netvsc
```

```
apt-get install linux-virtual linux-cloud-tools-virtual linux-tools-virtual  
update-initramfs -u  
reboot
```