Setting up an OpenVMS Virtual Machine on VMware ESXi

Hints & kinks to avoid the most common pitfalls when configuring an
OpenVMS VM within an ESXi environment

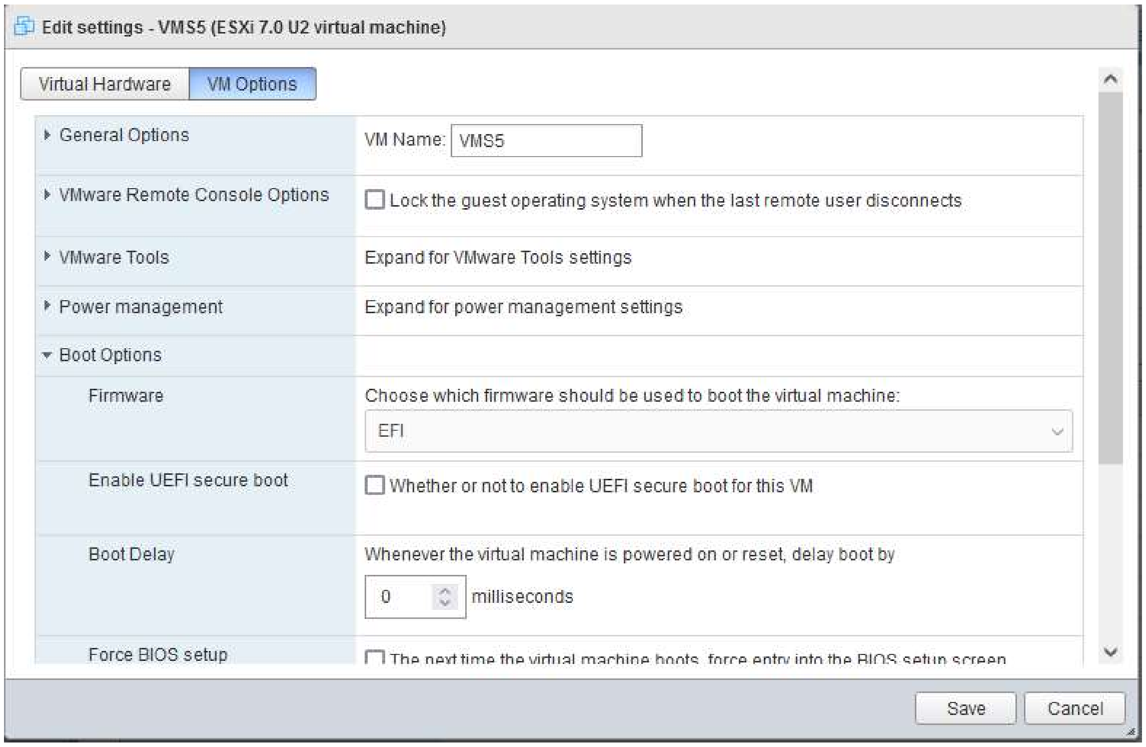
Thilo Lauer, Teracloud GmbH   March 21st, 2023

VMware ESXi is a powerful, highly configurable (and therefore potentially complicated), type-1 hypervisor. Although preparing a virtual machine (VM) in general, may be a straightforward exercise, some special requirements of the OpenVMS boot manager, such as the use of the EFI shell and the required need for serial line setup, justify a closer look into certain aspects of performing a proper configuration and a successful setup of a working OpenVMS VM environment.

This document has been created as a companion guide to the demonstration session held in Eindhoven at the OpenVMS Forum on March 7th, 2023. It is not intended to serve as a self-contained, complete description on how to setup an OpenVMS virtual machine. The reader is likely to want to consult the VSI OpenVMS installation Guide, the VSI Boot Manager User Guide and, of course, the VMware documentation for ESXi.

Topic 1 – Boot configuration / Firmware

An OpenVMS system relies on functions being provided by the EFI layer of the underlying hardware. Make sure to select "EFI" as the active Firmware in the "Boot options" entry in the "VM Options tab of the "Edit settings" window in the ESXi Web interface. Failure to do so will result in a black-and-white console window running BIOS with the classic progress bar at the bottom, trying to perform a network boot since BIOS didn't find any bootable device.


Edit settings - VMS5 (ESXi 7.0 U2 virtual machine)

Virtual Hardware | VM Options

General Options	VM Name: VMS5
VMware Remote Console Options	☐ Lock the guest operating system when the last remote user disconnects
VMware Tools	Expand for VMware Tools settings
Power management	Expand for power management settings
Boot Options	
Firmware	Choose which firmware should be used to boot the virtual machine: EFI
Enable UEFI secure boot	☐ Whether or not to enable UEFI secure boot for this VM
Boot Delay	Whenever the virtual machine is powered on or reset, delay boot by 0 milliseconds
Force BIOS setup	☐ The next time the virtual machine boots, force entry into the BIOS setup screen

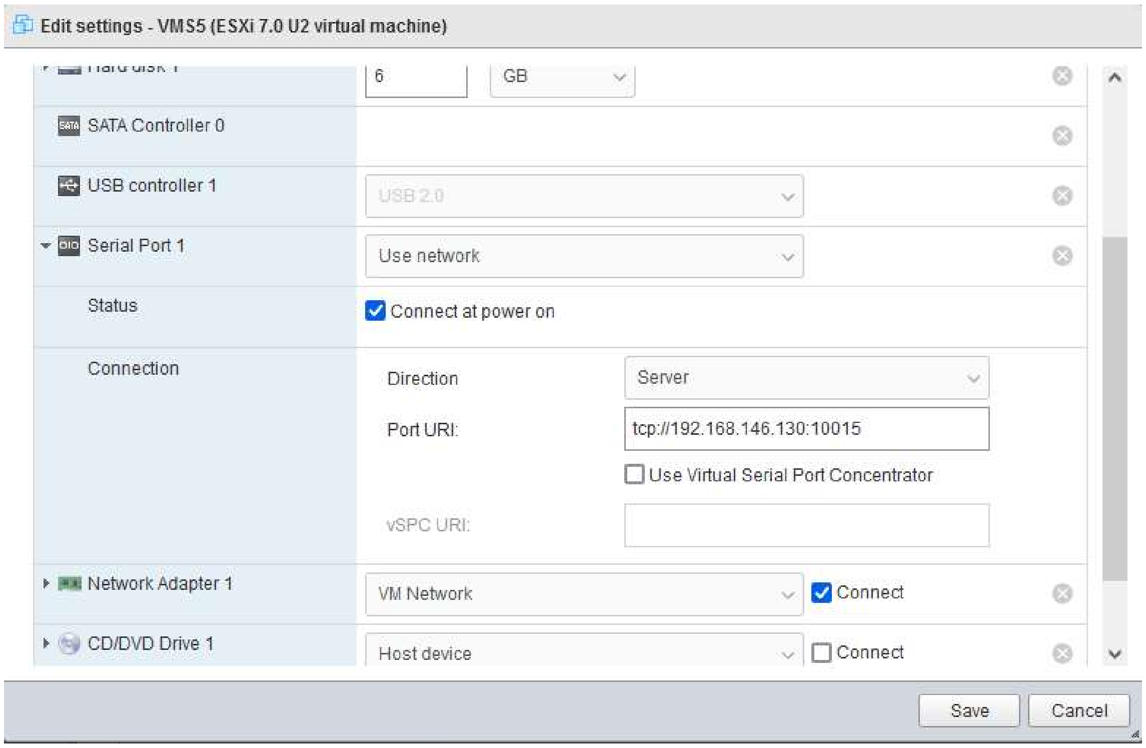
Topic 2 – Configuration of the serial line connection

To manage the boot process, to get access to the EFI shell, and to install OpenVMS, a serial line connection to the VM is required. This "serial line" can be provided by several alternatives:

- a true serial port,
- a named pipe, which only works within the same hypervisor, i.e., from an existing VM running a terminal emulator to the new OpenVMS VM environment,
- a TCP-based connection to any telnet client that has network access to the hypervisor.

By far the most common option one might want to use is the TCP-based setup. Several steps are needed to establish a working connection.

First, the VM needs to be configured to use a TCP-based serial line. The following picture taken from the VM's edit window shows an example configuration.

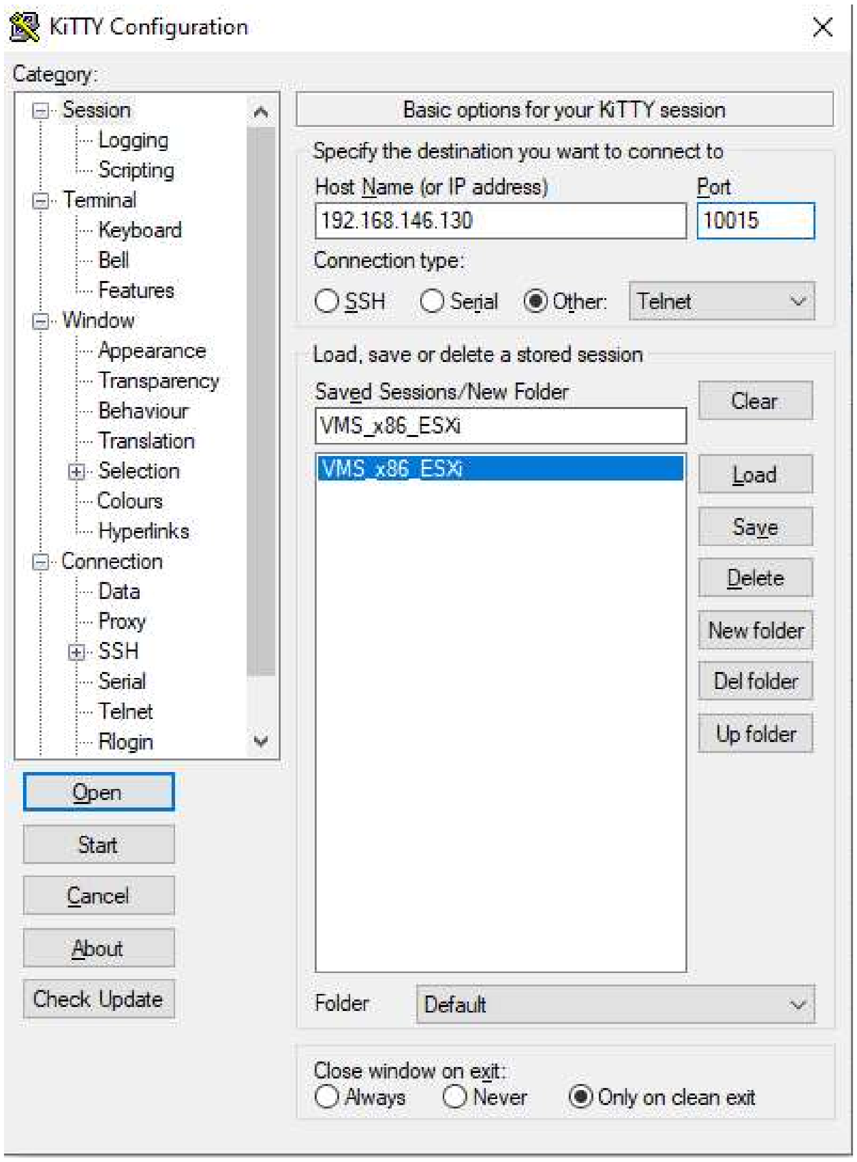


Edit settings - VMS5 (ESXi 7.0 U2 virtual machine)

Hard disk 1	6 GB	
SATA Controller 0		
USB controller 1	USB 2.0	
Serial Port 1	Use network	
Status	[✓] Connect at power on	
Connection	Direction: Server	
	Port URI: tcp://192.168.146.130:10015	
	[ ] Use Virtual Serial Port Concentrator	
	vSPC URI:	
Network Adapter 1	VM Network	[✓] Connect
CD/DVD Drive 1	Host device	[ ] Connect

Watch out for the correct syntax of the Port URI: it needs to have a unique port number appended to the IP address of the ESXi host. In this case it is port 10015. This value must be unique with regard to other existing or planned OpenVMS VM instances to distinguish the particular VM one wants to connect to later on (and not conflict with any services which may be running on the host machine).

The correct setting for the counterpart on the other end of the serial line connection, your terminal emulator, is shown in the following picture with using Kitty, a follow-on product of the widespread putty tool. While the host name to use is the IP address of the ESXi hypervisor, the port name specified in the "Port" field is the one that was assigned in the above step.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.146.130

Port: 10015

Connection type: SSH [✓ Other: Telnet] Serial

Load, save or delete a stored session

Saved Sessions/New Folder: VMS_x86_ESXi

VMS_x86_ESXi

Folder: Default

Close window on exit: Always Never [✓ Only on clean exit]

With these settings, the terminal emulator should – in theory - be able to establish a connection to the VM. However, the integrated ESXi firewall might still block access via a remote serial port.

Topic 3 – Configuration of the ESXi firewall

Even if all of the above settings have been applied correctly, access to the serial port of the VM might still be blocked. In this case the configuration of the ESXi firewall needs to be checked.

These steps are needed to control – and modify, if needed – the ESXi firewall setup:

- check the existing firewall rules regarding remote serial ports

  ```
  esxcli network firewall ruleset rule list -r remoteSerialPort
  Ruleset           Direction  Protocol  Port Type  Port Begin  Port End
  ----------------  ---------  --------  ---------  ----------  --------
  remoteSerialPort  Outbound   TCP       Dst                 0     65535
  remoteSerialPort  Inbound    TCP       Dst                23        23
  remoteSerialPort  Inbound    TCP       Dst              1024     65535
  ```

  The line of interest is the last one, controlling Inbound serial port connections. This example shows the default setting of allowing port numbers in the range of 1024 – 65535.

- However, this rule, while already being defined, might not be enabled. To check for that, use the following command (same as above, but without the keyword `rule`):

  ```
  esxcli network firewall ruleset list -r remoteSerialPort
  Name              Enabled
  ----------------  -------
  remoteSerialPort    false
  ```

  In this example, the remoteSerialPort rule is indeed not activated yet, and access therefore will be blocked. To activate this rule, use the command:

  ```
  esxcli network firewall ruleset set -r remoteSerialPort -e true
  ```

- Use this command to check if the firewall is active at all:

  ```
  esxcli network firewall get
  ```

  The following command will enable the firewall:

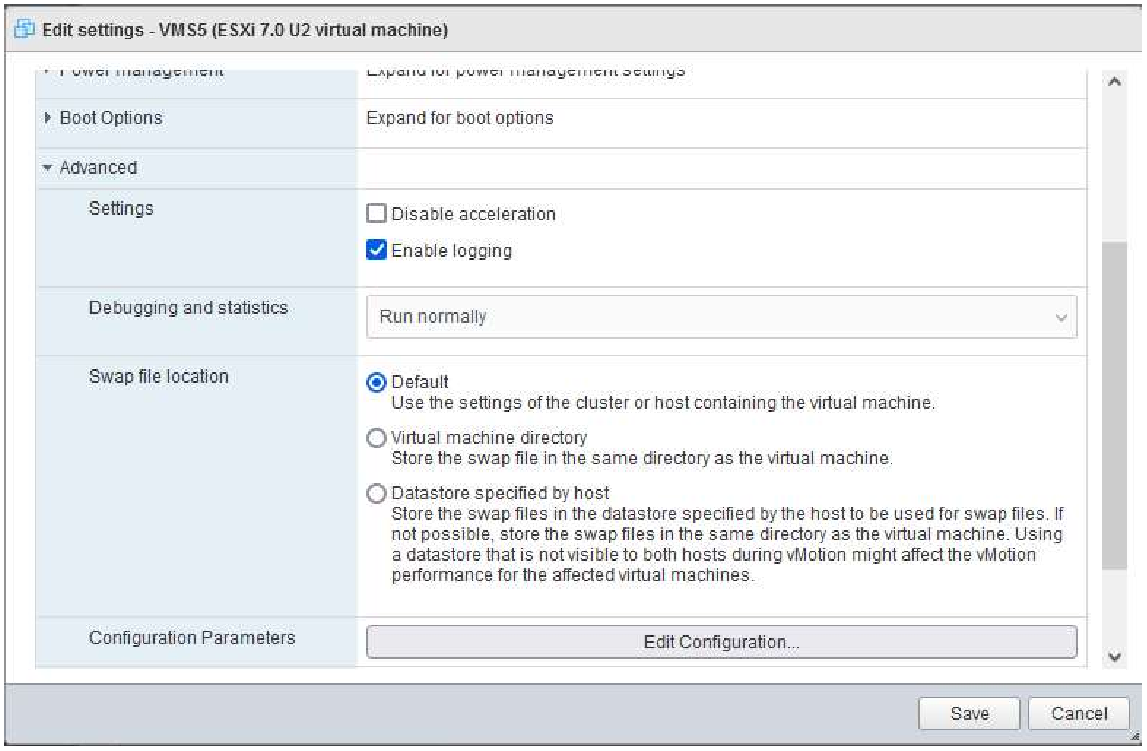
  ```
  esxcli network firewall set -e true
  ```

  while this one will deactivate (disable) the firewall completely. Use this command as a temporary troubleshooting method only!

  ```
  esxcli network firewall set -e true
  ```

Topic 4 – Setting ESXi VM variables

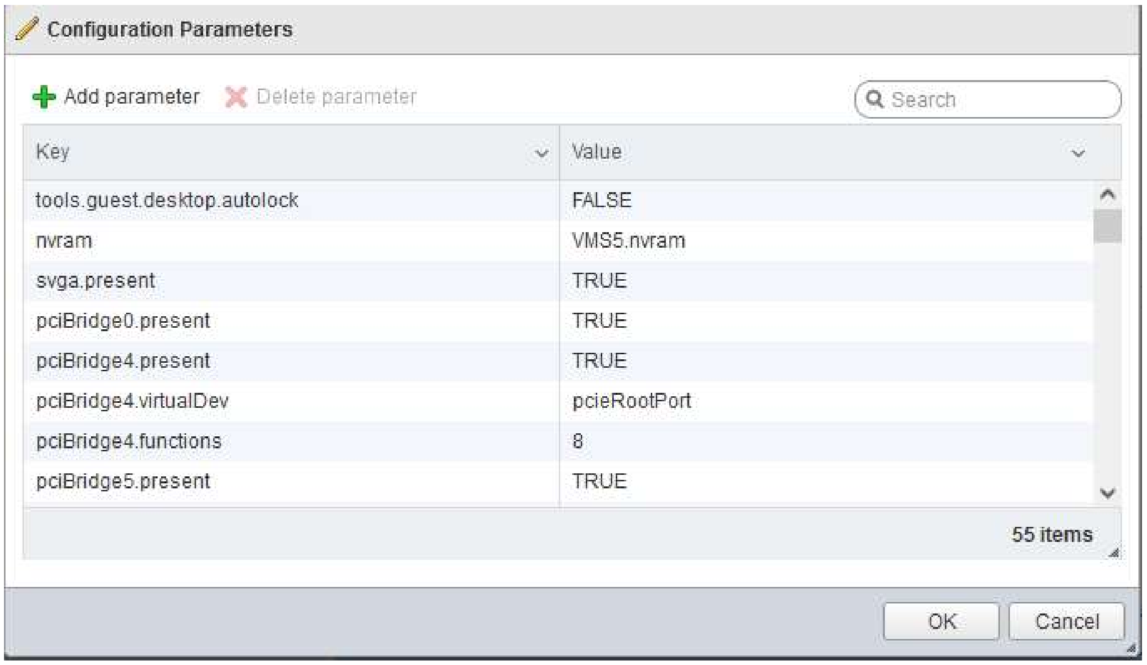
Certain variables available within a VM's configuration file may need to be defined or adjusted to better serve the specific needs of a virtualized OpenVMS environment. Modifying or crea-ting variables is performed within a VM's "Edit" window at the Tab "VM Options", entry "Advanced":



Edit settings - VMS5 (ESXi 7.0 U2 virtual machine)

Power management	Expand for power management settings
Boot Options	Expand for boot options
Advanced	
Settings	☐ Disable acceleration ☑ Enable logging
Debugging and statistics	Run normally
Swap file location	● Default Use the settings of the cluster or host containing the virtual machine. ○ Virtual machine directory Store the swap file in the same directory as the virtual machine. ○ Datastore specified by host Store the swap files in the datastore specified by the host to be used for swap files. If not possible, store the swap files in the same directory as the virtual machine. Using a datastore that is not visible to both hosts during vMotion might affect the vMotion performance for the affected virtual machines.
Configuration Parameters	Edit Configuration...

Click on the "Edit Configuration" button and the following window will appear, allowing you to either change the values of existing variables, or add new variables to the VM configuration file.



Configuration Parameters

Key	Value
tools.guest.desktop.autolock	FALSE
nvram	VMS5.nvram
svga.present	TRUE
pciBridge0.present	TRUE
pciBridge4.present	TRUE
pciBridge4.virtualDev	pcieRootPort
pciBridge4.functions	8
pciBridge5.present	TRUE

55 items

You can search for an existing variable using the search box in the upper right corner, or click on the “Add parameter” icon to actually add a new one.

The following variables might be of interest:

- `efi.serialconsole.enabled = "TRUE"`

  This variable, when being set to TRUE, will enable remote access to UEFI over a serial port. The default VM configuration does not have this variable defined, with the effect that any terminal I/O before the SYSBOOT stage of a VMS boot (including the initial boot from the ISO to start an installation) will not be sent/received to/from a connected terminal emulator, but will only be visible/accepted within the build-in Console window of ESXi.

  Once this variable is defined as shown above, access to the VM console from the very beginning, including the EFI shell and the platform boot manager will be possible.

- `efi.shell.activeByDefault = "TRUE"`

  VMware by default does not allow the **automatic** execution of the EFI shell during a VM boot. This is true for VMware Workstation as well as ESXi. What will happen instead is the execution of the available boot options in the platform boot manager, in their order of appearance in the boot options list, until an entry that results in a successful boot is found.

  To allow for the automatic execution of the EFI shell, define the variable as shown above.

- `efi.quickBoot.enabled = "FALSE"`

  By default, this variable is set to “TRUE”, which results in only performing a reduced device scan during the startup of a VM. This might result in an incorrect device discovery for OpenVMS virtual machines, especially if you are adding more controllers to your VM after the initial VM configuration. Setting this variable to “FALSE” will inhibit the quick boot and therefore will always perform a full device scan during boot.