Lab 3:

Virtualization   
with VMware:  
host-based   
& bare metal

Datacenter Virtualization

2024-2025

© Daan Pareit

CONTENT

[1 Introduction 3](#_Toc180067688)

[1.1 Learning goals 3](#_Toc180067689)

[1.1.1 Knowledge 3](#_Toc180067690)

[1.1.2 Skills 3](#_Toc180067691)

[1.2 Prerequisites 3](#_Toc180067692)

[2 Exploring Hosted Virtualization 4](#_Toc180067693)

[2.1 Preface 4](#_Toc180067694)

[2.2 Exploring a Virtual Machine 4](#_Toc180067695)

[3 Bare metal server virtualization 10](#_Toc180067696)

[3.1 Preface 10](#_Toc180067697)

[3.2 Installation 10](#_Toc180067698)

[3.2.1 Preparing your OS for nested virtualization 10](#_Toc180067699)

[3.2.2 Creating the VM to emulate the hypervisor 11](#_Toc180067700)

[3.3 Explore the DCUI 12](#_Toc180067701)

[3.4 Explore the Web GUI 14](#_Toc180067702)

[3.4.1 Networking 14](#_Toc180067703)

[3.4.2 Storage 18](#_Toc180067704)

[3.4.3 Creating a VM on ESXi 20](#_Toc180067705)

[3.4.4 Using your VM 21](#_Toc180067706)

[3.5 Explore connection from within VMware desktop application 26](#_Toc180067707)

# 

# Introduction

Nowadays, a lot of services are running virtually. Either using container-based virtualization (see later) or within Virtual Machines.

In this lab, we will explore the world of Virtual Machines, first the “end user” side using software like VirtualBox, Workstation, Parallels or Fusion (= “hosted virtualization” or “type 2 virtualization”).   
But eventually, we will learn to work with the, more performant, “*Bare Metal”* type of virtualization (= “type 1 virtualization”) by installing the hypervisor “VMware ESXi”

A logo of a company

Description automatically generated

## Learning goals

### Knowledge

* Know how virtualization works
* Know how VMware host-based virtualization works
* Know what ESXi is
* Know how VMware bare metal virtualization works

### Skills

* Be able to detect the parts that make up a VM
* Be able to configure Virtual Machines
* Set up and manage ESXi

## Prerequisites

This lab will be performed on VMware Workstation Pro or VMware Fusion Pro, which supports running ESXi as virtual machine. See earlier lab on how to install this software.

You’ll need the following:

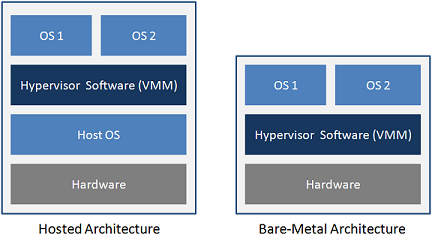
* An installation image for VMware ESXi (VMware-VMvisor-Installer-<release number>.x86\_64.iso), can be found on <https://filesender.belnet.be/?s=download&token=6cd41890-c249-424d-aeb4-22ab67045fa8> .
* To test a low-resource-requiring VM running on ESXi, we will boot into a simple Operating System. We’ll use a PuppyLinux distro “VoidPup64”, for which the ISO file can be found on SourceForge: <https://sourceforge.net/projects/pb-gh-releases/files/latest/download>

# Exploring Hosted Virtualization

## Preface

There are two types of so-called “Operating System Virtualization”:

* Hosted Virtualization
* Bare Metal Virtualization

  
*Source:* [*http://www.ni.com/white-paper/9629/en/*](http://www.ni.com/white-paper/9629/en/)

The difference is in architecture and foremost: performance!

* Hosted: a regular Operating System is in control of the hardware
* Bare Metal: a custom optimized-for-virtualization Operating System is in control of the hardware

## Exploring a Virtual Machine

Before playing with bare metal virtualization, let’s make sure you thoroughly understand what VMs are and what they’re made of.

1. Let’s create a VM and see how this works, open Workstation and create a new VM, select the **Typical method** and **Install the OS later**, select **Debian 12.x** (not 64-bit) and a random name.  
   Please also select **Store virtual disk as a single file**.  
   Keep everything else default but **remember where the VM is located**.

A screenshot of a computer

Description automatically generated  
*By default VMs are stored in “My Documents”*

1. Now let’s look at the settings of the VM before booting it. Open the settings and take notes:
   1. **Q: Select “Processor”: is the “VT-x/EPT or AMD-V/RVI” for processor virtualization option selected? Why (not)?**

**It is not selected. This looks like additional virtualization assists. They are not selected by default because most users do not need them.**

* 1. **Q: Go to “Options” > Advanced, does this VM use UEFI or BIOS firmware?**

**BIOS is the default for VMs as far as I remember.**

**A screenshot of a computer

Description automatically generated  
Q: Any idea why this setting is greyed out?**

**Because we selected the 12.x version of Debian, and not x64, most probably because of that.**

* 1. Try to remember or take note how many CPU’s / cores and memory the VM has

A screenshot of a computer

Description automatically generated

1. Now close VMware Workstation/Fusion completely and navigate to the **folder** where the VM is stored.   
   There should be 4 files visible: a VMDK, a VMX, a VMSD and a VMXF file

**A screenshot of a computer

Description automatically generated**

* 1. Now **delete all files except the VMDK and VMX file**

**A screenshot of a video game

Description automatically generated**

* 1. **Q: What is the purpose of the VMDK and VMX file? Try opening the VMX file with a text editor and explain**

**VMX contains metadata regarding the VM itself (like the settings and everything)**

**VMDK is like the hard disk information of the VM**

* 1. (Again) open the VMX file in any editor and change the Memory Size to 512MB

**A screen shot of a computer

Description automatically generated**

1. Open Workstation/Fusion again and
   1. Verify (without booting the VM) in the VM settings the change in memory size.

**A screenshot of a computer

Description automatically generated**

* 1. Inspect which files are **automatically** added to the folder, without even running the VM.

**A screenshot of a computer

Description automatically generated**

1. Close your Workstation/Fusion again and inspect your VM folder: what has changed now?

**A screenshot of a computer

Description automatically generated**

**.lck folder/files are gone.**

1. Again, open Workstation/Fusion and now boot your VM (it’s OK not to have any ISO image in the CD tray). Same question as before: what has changed in your VM folder since your VM is running?

**Q: What is the purpose of the “.lck” files?**

**It is a file that makes sure that the VM can not be used/opened in other instances.**

**Q: What is the purpose of the “.mem” file? What size is it? Explain. Hint: explore the difference between powering off and pausing your VM.**

**A black screen with white text

Description automatically generated**

**After pausing it remains the same. This file represents the RAM of the VM.**

1. Try booting the VM into the firmware (also called BIOS). Hint: there is a special Workstation menu to easily boot into the BIOS

**A screenshot of a computer

Description automatically generated**

* 1. Verify the change in memory size (you’ve manually set in the vmx file) is correct.

**A screen shot of a computer

Description automatically generated**

**It is pretty much correct**

* 1. Change the Boot Order so that **CD-ROM Drive** is above the Hard Drive in the boot order and reboot the VM

**A grey background with blue text

Description automatically generated**

1. Now, let’s ‘insert’ a live boot CD by selecting the ‘*VoidPup64’* ISO file to be configured for the CD drive of our VM.
   1. Boot (or reboot) the VM  
      Note: Use **Ctrl+Alt+Insert** (which always translates to Ctrl+Alt+Del into the VM) to reboot
   2. It should boot from the CD and start the distro, without installing anything on the (virtual) disk. Press Enter and let it boot

It can boot if you do not even touch anything.

1. Once booted, you’ll be presented with a wizard in which you can choose a keyboard layout and hit ‘OK’. You can explore the live Puppy Linux a bit.

A screenshot of a computer

Description automatically generated

We’ve investigated the different files a VM is made of, and we’ve quickly booted a sample live distro on hosted virtualization. This VM can now be powered off. Let’s run the same distro on bare metal server virtualization and explore the differences.

# Bare metal server virtualization

## Preface

This is where we move to a new world, the world of combining an Operating System and a virtualization stack. This is called: **A Hypervisor**.

A diagram of a computer chip

Description automatically generated  
*Source: vSphere Install, Configure & Manage*

The Hypervisor from VMware is called **ESXi**. It used to be called ESX, when it was a hypervisor combined with an OS based on Red Hat.

But ESXi (‘i' from ‘integrated’) is actually one monolithic Operating System that is custom made from scratch by VMware, it incorporates *“Busybox”* which supports commands similar to Unix, but ESXi is not Unix.

ESXi incorporates open-source software components, but also uses a unique proprietary OS kernel – the **VMkernel** – and which implements unique features that are not available in any other operating system (e.g. the **VMFS** file system).

## Installation

Compared to Windows and Linux, only a certain amount of hardware is supported with ESXi. E.g. it will probably not install natively on your laptop (without “hacks”).

There is an online list of supported hardware: the ESXi [Hardware Compatibility List](https://www.vmware.com/resources/compatibility/search.php) (HCL).

We will not install the VMware ESXi hypervisor on real hardware, but we will install this inside a VM. Luckily this is supported by VMware Workstation/Fusion.

### Preparing your OS for nested virtualization

A hypervisor’s purpose is to run virtual machines. But if you install the hypervisor itself within a virtual machine in your VMware Workstation/Fusion, the VMs running on the hypervisor will be VMs running inside your virtual machine. This is what we call **nested virtualization** and the VMs running inside the hypervisor in your VM, are sometimes called **vVMs** (virtual VMs).

If you already have an active virtualization layer on your host OS, running vVMs on your hypervisor VM in VMware Workstation/Fusion would add two more layers of virtualization. That would be too complex to handle and your VM will fail to run.

This is most notably the case if you run a Windows host OS with Hyper-V enabled (required for e.g. WSL2). In that case:

* You need to (temporarily) disable Hyper-V: see ‘hyperv-vs-vmware.txt’ on Leho

bcdedit /set hypervisorlaunchtype off

shutdown /r /t 0

bcdedit /set hypervisorlaunchtype auto

shutdown /r /t 0

* If you get the warning that Intel VT-x/EPT is not supported on your platform, it might be due to Credential Guard. It can be disabled using this Microsoft PowerShell script: <https://www.microsoft.com/en-us/download/details.aspx?id=53337>

### Creating the VM to emulate the hypervisor

Let’s create the VM which will contain your hypervisor:

1. Open VMware Workstation/Fusion and create a new VM: make sure to select *VMware ESXi 8* as the Guest OS. Store the virtual disk as a single file and if possible, customize hardware to allocate 4 CPUs instead of the default 2 CPU cores.

142GB… Why so much gosh…

A screenshot of a computer program

Description automatically generated

* 1. Before booting the VM, open the settings and look at the Processor Settings  
     **Q: Compare the CPU options with the ones from our Linux/Windows VMs (see above)**

A screenshot of a computer program

Description automatically generated

+ UEFI selected

* 1. Make sure to select the VMvisor ISO as the CD/DVD device. Don’t forget the checkbox “Connect at power on”.

A screenshot of a computer

Description automatically generated

* 1. Boot the VM

A screenshot of a computer error

Description automatically generated

Nah I am disabling it, I am not really in the mood to run some scripts from Microsoft.

1. Let the installer load and install ESXi, use defaults (except for correct keyboard layout, e.g. ‘Belgian’ if appropriate) and a simple password like Rootroot123-

Rootroot123-

* 1. Notice that the number of options is limited when installing this Hypervisor
  2. Once installed it will reboot, use this time to look at the **size of the VMDKQ: In other words what is the Disk Footprint of ESXi?**



ALMOST 1GB.

A screenshot of a computer

Description automatically generated  
*Once installed, we are greeted with this Direct Console User Interface*

*A screenshot of a computer

Description automatically generated*

## Explore the DCUI

The Direct Console User Interface (DCUI) lets us do some basic configurations:

1. Log in and look at the options, walk through all of them and take notes of what they are for.
   1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   2. …

Basically you can set up some things like default machines, and also other things.

A screenshot of a computer

Description automatically generated

1. Eventually: select “*Reset System Configuration*”, perform this action and reboot

## Explore the Web GUI

Once ESXi has rebooted, its IP address should show: the main configuration is to be done via the web interface. Open a browser and type in the IP address.

A screenshot of a computer

Description automatically generated

Use root and <no password> to log in.

As the root password was set to nothing by your system reset, change the root password by clicking “Manage” > “Security & Users” > “Users”. It goes without saying that a hypervisor without root password is not a good idea.

### Networking

Networking on ESXi uses some new concepts:

1. In ‘Networking’, click on *Physical NICs*  
   “Physical” interfaces or NICs always have a name starting with ***vmnic***.
   1. So instead of having *ens33* and *ens36* or *eth0* and *eth1*, ESXi uses *vmnic0* and *vmnic1*
   2. In our case, the “physical” NIC is the one from Workstation, so the driver is “nvmxnet3”

A screenshot of a phone

Description automatically generated

* 1. **Q: What is the reported Link speed? Why?**

**10000 Mbps.**

**Like 1 GBps.**

1. Click on *Virtual switches*  
   Virtual switches: are identical in operation to any managed physical layer2 switch
   1. By default, every ESXi installation must have one, needed to access the hypervisor.   
      The default name is *vSwitch0*
   2. They have about 2560 interfaces (can change dynamically)
   3. Some ports are used for *Uplinks*, which are connections to the “physical” world
   4. On the virtual side: ports are divided into two kinds:  
      **VMkernel Port Groups** and **Virtual Machine Port Groups**

A screenshot of a computer

Description automatically generated

A diagram of a virtual switch

Description automatically generated  
*Source: vSphere Install, Configure & Manage 6.5, 5-7*

🡪 Virtual Machine Port Groups can only be used by VMs, to allow them access to any given network  
🡪 VMkernel ports cannot be used by VMs and are for the **Hypervisor Kernel** (called ***VMkernel***), these port groups are needed to allow the hypervisor to have access. E.g. for Management (the IP that we are using now) or for storage etc …

1. Click on *Port groups***Q: What is the name of the default VM Port Group in our system? And the VMkernel Port Group?**

A screenshot of a computer

Description automatically generated

VMkernel Port Group – Management Network??? (since it is also said we use it)

VM Port Group – VM Network???

### Storage

Now click on ‘Storage’ in the left-hand pane, there are two interesting concepts here:

1. Click on *Adapters*  
   Physical storage adapters always have a name starting with ***vmhba***.
   1. For example, the “physical” SCSI adapter gets the name *vmhba0*
   2. Whereas the “physical” IDE adapter (for our CD/DVD Drive) gets *vmhba64*

A screenshot of a computer

Description automatically generated

1. Click on *Devices* to confirm this, notice the device path for the disk itself

A screenshot of a computer

Description automatically generated

1. Most important, click on *Datastores*
   1. A partition that can contain usable data is called a ***datastore****.* 
      * By default, the space on our primary hard drive that was not taken by ESXi itself, has been partitioned and formatted to be a datastore.
      * If there would be no datastore, open the VM settings and add a **new hard** drive (size 40GB is enough)  
        Then, on the website, go to Datastores and click **New datastore**, select the new hard drive, name it *datastore1* and just click next and finish

Thankfully I have one.

A screenshot of a computer

Description automatically generated

* 1. **Q: Which custom File System is the datastore formatted as? What is the block size (comparable to cluster size in NTFS) and why is this important?**

**VMFS6**

**A screenshot of a computer

Description automatically generated**

**It is way bigger than NTFS (default 4K).**

**It is probably needed as we install machines, and those machines will not probably have local storage, hence it is probably easier to manage bigger chunks of blocks.**

**A screenshot of a computer

Description automatically generated**

**I was right!.**

1. Let’s browse the default datastore (*datastore1*)
   1. Upload the VoidPup64 ISO to the root of this datastore

A screenshot of a computer

Description automatically generated

### Creating a VM on ESXi

Let’s create a vVM. For the sake of ease, we’ll use the wizard.

1. On the ESXi website: Click on *Virtual Machines* and then *Create / Register VM*.
2. We’ll create a VM and call it “**<yourfirstname>-<yourlastname>-Puppy-VM**”, select the “*Linux”* Guest OS family and “*Other Linux (64-bit)*” Guest OS version.  
   Note that no other Server Virtualization vendor support this amount of guest types.

A screenshot of a computer

Description automatically generated

1. At the “*Customize settings*” page
   1. Use a memory size of 512MB.
   2. **Q: What are the default settings for “*Hard drive 1*”?**

A screenshot of a computer

Description automatically generated

Thick provisioned, lazily zeroed

8 GB

IOPs Unlimited

🡪 Change the provisioning accordingly in order not to pre-allocate disk space (for the sake of your laptops disk space).

SO THIN PROVISIONING.

* 1. **Make sure** that the “*CD/DVD Drive 1”* has the VoidPup64 ISO file connected

1. Finish the VM and boot it.
2. This is a live boot distro, which didn’t require any installation on a (virtual) disk. Nevertheless, it’s a functional one, running on your ESXi.

A screenshot of a computer

Description automatically generated

**Q: Create a screenshot of your personal Puppy-VM within the ESXi web GUI**

**OF COURSE I CAN NOT RUN OR CREATE IT PROPERLY BECAUSE THE NESTED VIRTUALIZATION**

**I ran the script to disable (both credential guard and another setting for the sake of this lab) some random stuff, and check my BIOS has got AMD-V enabled (thank you AMD), and then pray for ESXi to actually boot with the setting enabled. And thank god it did.**

**A screenshot of a computer

Description automatically generated**

### Using your VM

1. From within the web interface, you can click the distro screenshot and interact with your live distro.
2. Let’s make it more tangible by ssh’ing to that VM from your laptop. First, you’ll need to start the SSH server which isn’t running by default. As the distro is very small, it doesn’t use systemd (see Linux Server Security course) to do so. Here is how to start the ssh server:
   1. Use ‘which sshd’ to find the location of the sshd binary
   2. Now, run that binary. It will exit with an error: which one?

Sshd: no hostkeys available -- existing.

* 1. To fix that, run “ssh-keygen -A”
  2. Now, run the sshd binary again. It will exit again with a different error: which one?

Missing privilege separation directory: /var/chroot/ssh

A screen shot of a computer

Description automatically generated

* 1. Fix that by creating what it is asking for
  2. Now, running the sshd binary should work

1. Create a user <yourfirstname-yourlastname> using ‘adduser’

User – serafim-ciobanu

Password - void

1. Use ‘visudo’ to add your user to the sudoers file.
2. Check the IP address of your vVM and try to login to your vVM over SSH with your new user. You’ll see the SSH server is listening, but you’re denied access. This is because this small live distro isn’t using PAM (see Linux Server Security course) either.
   1. Stop the sshd process with “pkill”
   2. Change the sshd\_config file to not use PAM

Cd /etc/ssh

* 1. Restart the sshd binary.
  2. Now ssh’ing with your new user to the vVM should succeed.

A computer screen with white text

Description automatically generated

1. Within the SSH session, now execute some commands to verify you’re indeed now within the fully operational vVM on ESXi (with limited features as it’s a small live distro):

**Q: Take a screenshot of your ssh session with successful execution of:   
whoami; hostname ; cat /etc/os-release; sudo ping google.be -c 4**

**A screenshot of a computer screen

Description automatically generated**

1. We had allocated a virtual disk to the vVM but haven’t used it so far. Have a look in the datastore at the size of the virtual disk file.
2. Now let’s use that disk by partitioning it and formatting it with a file system. In your VM, type *sudo fdisk /dev/sda* to manipulate the disk and create a single partition taking all disk space (see Forensics course). Format the partition with ext4 by typing *mkfs.ext4 /dev/sda1*

Better use cfdisk, then do lsblk, and then do mkfs.ext4. And check with lsblk -f

1. Now inspect again the size of the virtual disk file of your VM in the datastore.

A screenshot of a computer

Description automatically generated

1. Now for a surprise: download the VMDK file. There is a big difference in VMDK files between the ESXi hypervisor and hosted virtualization. Try opening the downloaded VMDK in notepad. Notice that this actually does not contain any guest data!  
   **Q: What is the actual filename of the VMDK-file containing the binary data?**

A screenshot of a computer program

Description automatically generated

Serafim-Ciobanu-Puppy-VM-flat.vmdk is supposed to contain the binary data.

1. Download the VMX file and compare its contents with a VMX file on your laptop  
   **Q: Can you make out some details on the File System layout that ESXi uses?** **Take notes**

I am not sure how much different it is, but what I know is for sure that ESXi creates folders for the VMs (pretty much like we do), and then it might have information regarding specific server things setting up.

## Explore connection from within VMware desktop application

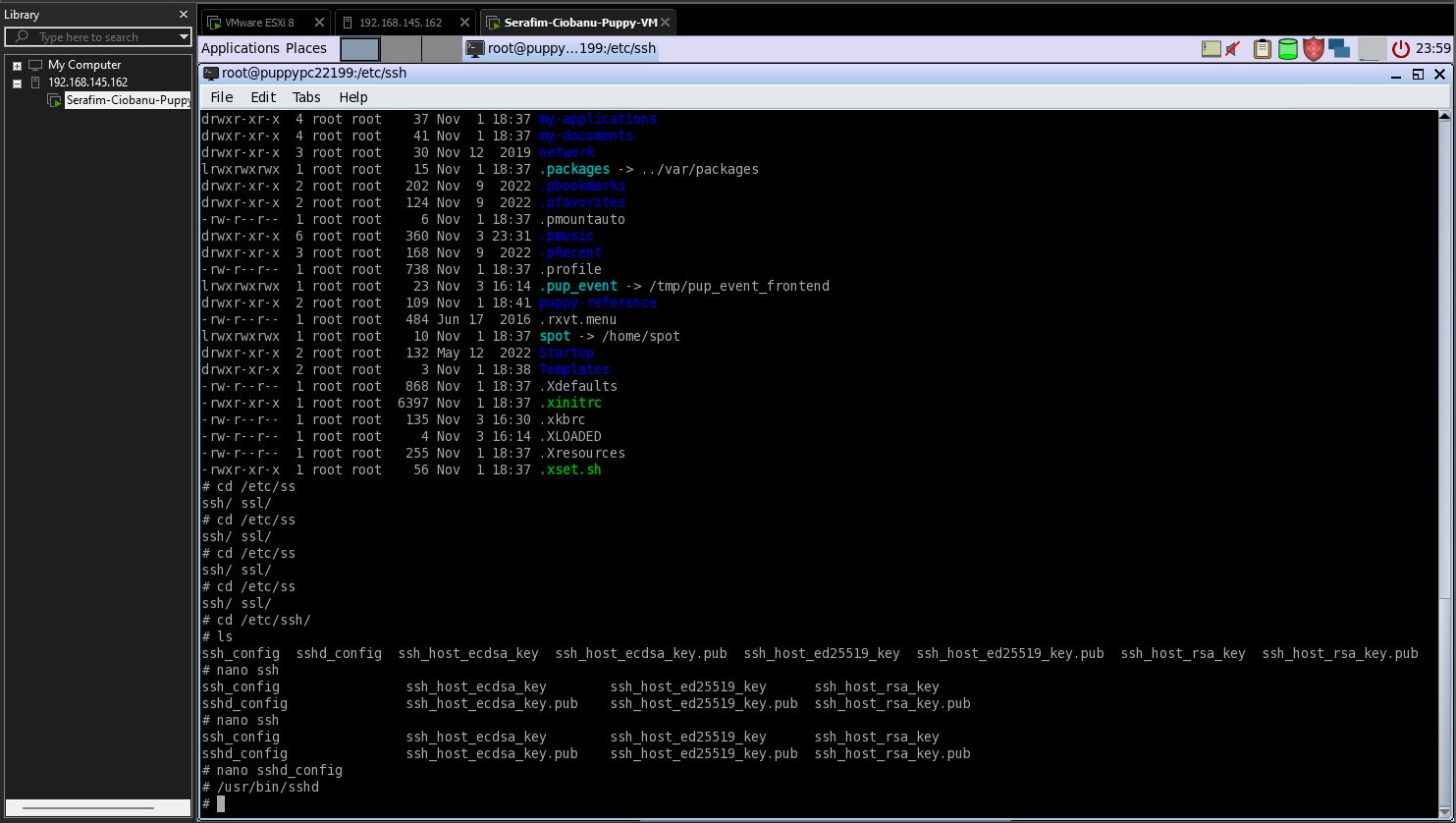
Finally, you could also use your desktop VMware Workstation/Fusion application instead of the ESXi web interface to connect to the ESXi server and use/manage your VMs from within this application, similar as the VMs on your laptop.

1. In VMware, choose to “Connect to server”. Provide the IP address and credentials of the ESXi server. Now, you can access the VMs on that server from within your desktop application.

**Q: Take a screenshot of your Puppy-VM from within your VMware workstation/fusion**

**A screenshot of a computer

Description automatically generated**

****