

**Securing Zoom in a Virtual Machine**  
Daniel Gray

Written for...



April 3, 2020  
Version 1.0



## Contents

<b>1</b>	<b>Download the necessary files</b>	<b>2</b>
<b>2</b>	<b>Verifying the files</b>	<b>2</b>
2.1	Windows 10 . . . . .	2
2.2	Mac OS . . . . .	4
2.3	Linux . . . . .	5
<b>3</b>	<b>Setting up the VM</b>	<b>6</b>
<b>4</b>	<b>Installing Zoom</b>	<b>16</b>
<b>5</b>	<b>Snapshots</b>	<b>17</b>



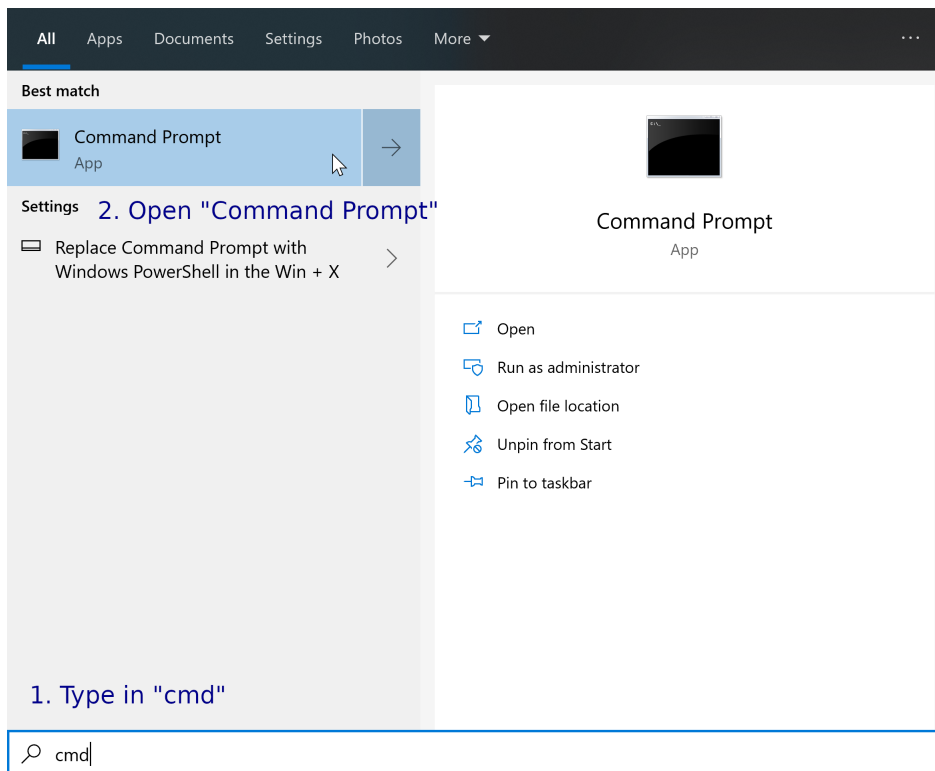
## 1 Download the necessary files

1. Download VirtualBox for your platform from the [download site](#). Get the “SHA256 checksums” too. It is assumed you saved these files in `Downloads/VirtualBox`.
2. If you plan on using a webcam then also download the “VirtualBox Extension Pack” to this directory.
3. Download Debian. The files you want are `debian-10.3.0-amd64-xfce-CD-1.iso` (latest at time of writing), `SHA256SUMS.sign` and `SHA256SUMS`.  
You can do so by [HTTP](#) (slower download) or [BitTorrent](#) (preferred as it’s faster). A popular BitTorrent client is [Transmission](#). It is assumed you saved these files to `Downloads/Debian`.

## 2 Verifying the files

### 2.1 Windows 10

1. Download [gpg4win](#) to `Downloads/gpg4win`.
2. Next verify the [checksum](#) matches that on the [package integrity](#) page. You may need to scroll down to the part that says “SHA256 checksums”.
3. Click on “Start”, type in “cmd” and then open “Command Prompt”.



4. Assuming you saved the downloaded files to the above directory, type in:

#### Input

```
cd %USERPROFILE%\Downloads\gpg4win
certutil -hashfile gpg4win-3.1.11.exe SHA256
cd %USERPROFILE%\Downloads\VirtualBox
certutil -hashfile VirtualBox-6.1.4-136177-Win.exe SHA256
certutil -hashfile Oracle_VM_VirtualBox_Extension_Pack-6.1.4.vbox-extpack SHA256
```



## Output

```
SHA256 hash of gpg4win-3.1.11.exe:
156de9f3f50bb5a42b207af67ae4ebcb2d10a7aaf732149e9c468eaf74ce7ffc
CertUtil: -hashfile command completed successfully.

SHA256 hash of VirtualBox-6.1.4-136177-Win.exe:
66218bcd4f118e7b5d50e1804e5ec7e0b26d20976415b56992ffb0143b8fa7c
CertUtil: -hashfile command completed successfully.

SHA256 hash of Oracle_VM_VirtualBox_Extension_Pack-6.1.4.vbox-extpack:
3b73798d776ff223ea8025b1a45001762f8d4e5bcd1ea61449773c1249935800
CertUtil: -hashfile command completed successfully.
```

Check that the corresponding output matches the SHA256 checksum on the [package integrity](#) page. Also check the output matches `Downloads\VirtualBox\SHA256SUMS`. You can drag `SHA256SUMS` into Notepad to check.

5. Open the gpg4win installer and follow the steps presented.
6. In the command window we want to [verify the authenticity](#). We download the Debian public signing key.

## Input

```
gpg --keyserver hks://keyring.debian.org \
--recv-key DF9B9C49EAA9298432589D76DA87E80D6294BE9B
```

## Output

```
gpg: key DA87E80D6294BE9B: public key "Debian CD signing key <debian-cd@lists.debian.org>"
imported
gpg: Total number processed: 1
gpg: imported: 1
```

7. Then we check the authenticity of the SHA256SUMS for Debian:

## Input

```
cd %USERPROFILE%\Downloads\Debian
gpg --verify SHA256SUMS.sign
```

8. The result should be "Good signature":

## Output

```
gpg: key DA87E80D6294BE9B: public key "Debian CD signing key <debian-cd@lists.debian.org>"
imported
gpg: Total number processed: 1
gpg: imported: 1

gpg: assuming signed data in 'SHA256SUMS'
gpg: Signature made Sun 09 Feb 2020 02:01:05 UTC
gpg: using RSA key DF9B9C49EAA9298432589D76DA87E80D6294BE9B
gpg: Good signature from "Debian CD signing key <debian-cd@lists.debian.org>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
Primary key fingerprint: DF9B 9C49 EAA9 2984 3258 9D76 DA87 E80D 6294 BE9B
```

9. Finally check the checksum for the Debian `debian-10.3.0-amd64-xfce-CD-1.iso` matches the one in the `Downloads\Debian\SHA256SUMS` file.



## Input

```
certutil -hashfile debian-10.3.0-amd64-xfce-CD-1.iso SHA256
```

## Output

```
SHA256 hash of debian-10.3.0-amd64-xfce-CD-1.iso:  
47671cf75c68b8f0a2169857a7e2fe371553de3c956b87688604cc920bceb52e  
CertUtil: -hashfile command completed successfully.
```

## 2.2 Mac OS

1. Download [GPGTools](#), verify the [checksum](#) matches that on the GPGTools page. You'll need to mouse over "SHA256" to see it.
2. Open Terminal.app and run these commands

## Input

```
cd ~/Downloads/GPGTools  
shasum -a 256 GPG_Suite-2019.2.dmg  
cd ~/Downloads/VirtualBox  
shasum -a 256 VirtualBox-6.1.4-136177-OSX.dmg  
shasum -a 256 Oracle_VM_VirtualBox_Extension_Pack-6.1.4.vbox-extpack
```

The output should look like this:

## Output

```
98e26e3dc2fad3563ef1add6bc2cdaefa986462aeae256a20e894e16118a179  GPG_Suite-2019.2.dmg  
  
2bc5d7282d9af9ce12dffddb528dcf6c9eb7ea92e644885d805e1e56fd55bacf  
VirtualBox-6.1.4-136177-OSX.dmg  
  
3b73798d776ff223ea8025b1a45001762f8d4e5bcd1ea61449773c1249935800  
Oracle_VM_VirtualBox_Extension_Pack-6.1.4.vbox-extpack
```

3. If you already have an installed installation of `gpg` you can verify the signature and file in the Terminal with:

## Input

```
gpg --keyserver htps://keys.openpgp.org --recv-key 85E38F69046B44C1EC9FB07B76D78F0500D026C4  
cd ~/Downloads/GPGTools  
gpg --verify GPG_Suite-2019.2.dmg.sig
```

## Output

```
gpg: key 76D78F0500D026C4: 3 signatures not checked due to missing keys  
gpg: key 76D78F0500D026C4: public key "GPGTools Team <team@gpgtools.org>" imported  
gpg: no ultimately trusted keys found  
gpg: Total number processed: 1  
gpg:             imported: 1  
  
gpg: assuming signed data in 'GPG_Suite-2019.2.dmg'  
gpg: Signature made Fri 15 Nov 2019 15:23:17 UTC  
gpg:             using RSA key 8C31E5A17DD5D932B448FE1DE8A664480D9E43F5  
gpg: Good signature from "GPGTools Team <team@gpgtools.org>" [unknown]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg:             There is no indication that the signature belongs to the owner.  
Primary key fingerprint: 85E3 8F69 046B 44C1 EC9F  B07B 76D7 8F05 00D0 26C4  
Subkey fingerprint: 8C31 E5A1 7DD5 D932 B448  FE1D E8A6 6448 0D9E 43F5
```



## 4. Verify the authenticity of Debian:

## Input

```
gpg --keyserver hks://keyring.debian.org \  
--recv-key DF9B9C49EAA9298432589D76DA87E80D6294BE9B  
cd ~/Downloads/Debian  
gpg --verify SHA256SUMS.sign
```

## Output

```
gpg: key DA87E80D6294BE9B: public key "Debian CD signing key <debian-cd@lists.debian.org>"  
imported  
gpg: Total number processed: 1  
gpg: imported: 1  
  
gpg: assuming signed data in 'SHA256SUMS'  
gpg: Signature made Sun 09 Feb 2020 02:01:05 UTC  
gpg: using RSA key DF9B9C49EAA9298432589D76DA87E80D6294BE9B  
gpg: Good signature from "Debian CD signing key <debian-cd@lists.debian.org>" [unknown]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg: There is no indication that the signature belongs to the owner.  
Primary key fingerprint: DF9B 9C49 EAA9 2984 3258 9D76 DA87 E80D 6294 BE9B
```

## 5. Finally check the SHA256 checksum against the Debian ISO.

## Input

```
shasum -c --ignore-missing SHA256SUMS
```

## Output

```
debian-10.3.0-amd64-xfce-CD-1.iso: OK
```

## 2.3 Linux

1. Linux distributions already include **gpg** so it does not need to be installed, open a terminal and type:

## Input

```
gpg --keyserver hks://keyring.debian.org \  
--recv-key DF9B9C49EAA9298432589D76DA87E80D6294BE9B  
cd ~/Downloads/Debian  
gpg --verify SHA256SUMS.sign
```

## Output

```
gpg: key DA87E80D6294BE9B: public key "Debian CD signing key <debian-cd@lists.debian.org>"  
imported  
gpg: Total number processed: 1  
gpg: imported: 1  
  
gpg: assuming signed data in 'SHA256SUMS'  
gpg: Signature made Sun 09 Feb 2020 02:01:05 UTC  
gpg: using RSA key DF9B9C49EAA9298432589D76DA87E80D6294BE9B  
gpg: Good signature from "Debian CD signing key <debian-cd@lists.debian.org>" [unknown]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg: There is no indication that the signature belongs to the owner.  
Primary key fingerprint: DF9B 9C49 EAA9 2984 3258 9D76 DA87 E80D 6294 BE9B
```



2. Verify the file against the ISO image:

#### Input

```
sha256sum -c --ignore-missing SHA256SUMS
```

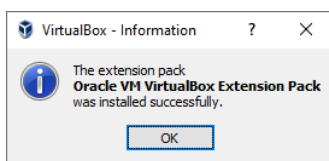
#### Output

```
debian-10.3.0-amd64-xfce-CD-1.iso: OK
```

3. For VirtualBox on Linux follow the instructions for your distribution on the [Linux Downloads](#) page.

## 3 Setting up the VM

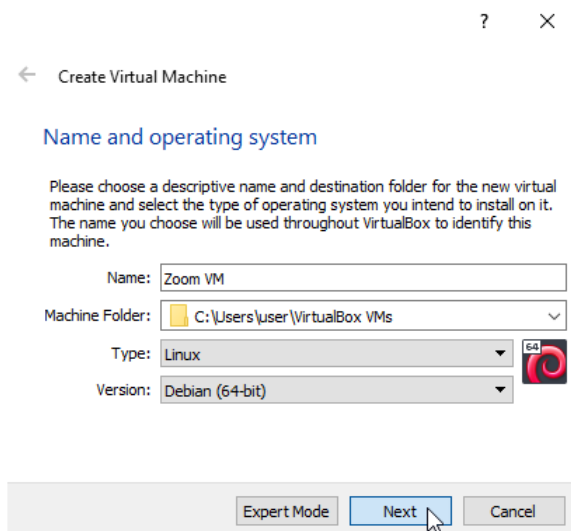
1. Once VirtualBox is installed open `Oracle_VM_VirtualBox_Extension_Pack-6.1.4.vbox-extpack`.



2. Create a new VM



3. Name VM and select location





- Set a memory size. Assuming you have 8GB of RAM total, you could allocate 3-4 GB to Debian.

?

×

← Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024 MB**.

4 MB 16384 MB

4096 MB

Next Cancel

- Hard disk, “Create a virtual hard disk now”.

?

×

← Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **8.00 GB**.

☐ Do not add a virtual hard disk

☒ Create a virtual hard disk now

☐ Use an existing virtual hard disk file

Empty

Create Cancel





6. Leave hard disk file type as “VDI (VirtualBox Disk Image)”.

? ×

← Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

☒ VDI (VirtualBox Disk Image)

☐ VHD (Virtual Hard Disk)

☐ VMDK (Virtual Machine Disk)

Expert Mode Next Cancel

7. Leave as storage as “Dynamically allocated”.

? ×

← Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

☒ Dynamically allocated

☐ Fixed size

Next Cancel

8. Set a size for the VM, 15GB should be ample.

? ×

← Create Virtual Hard Disk

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

C:\Users\user\VirtualBox VMs\Zoom VM\Zoom VM.vdi

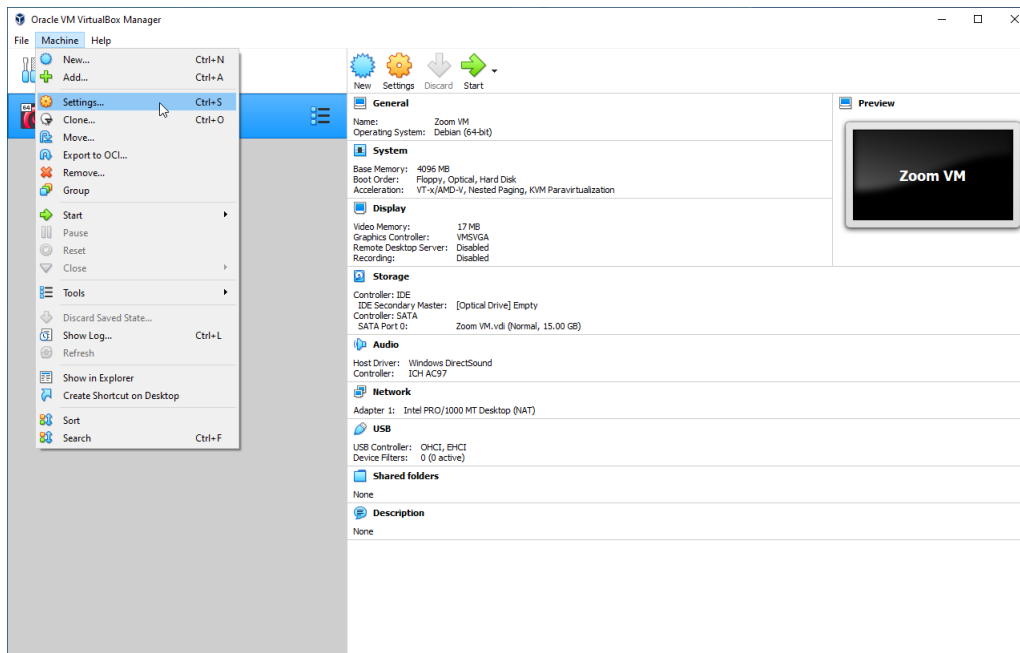
Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4.00 MB 2.00 TB 15.00 GB

Create Cancel

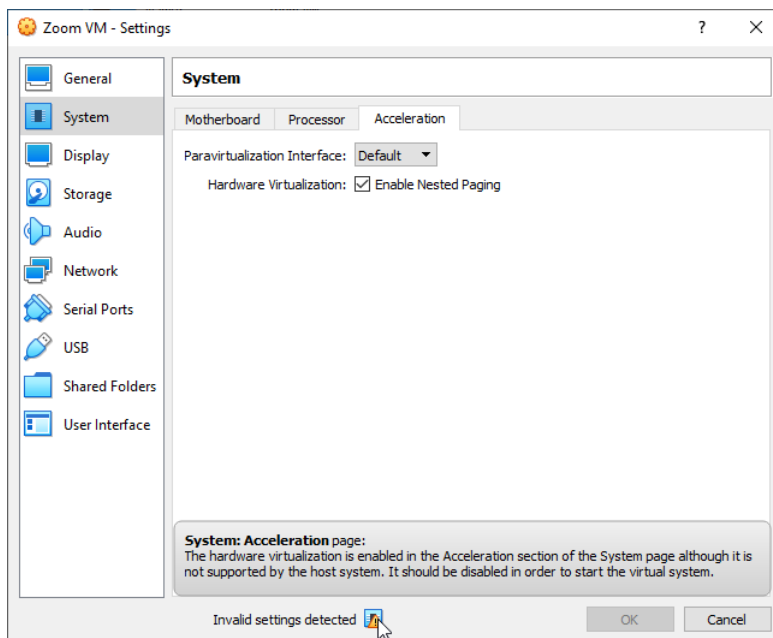


## 9. Enter the VM's Settings



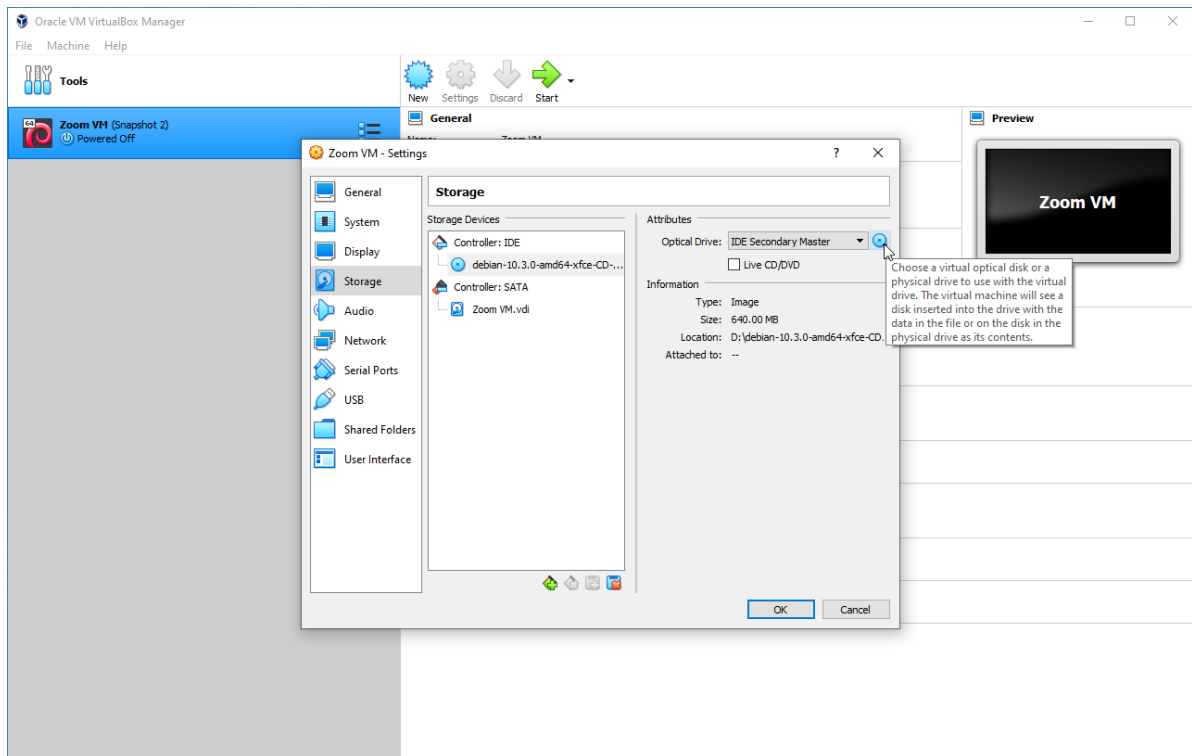
10. Optional: For some AMD based processors it may be required that users enable [Secure Virtual Machine \(SVM\)](#) in the [Unified Extensible Firmware Interface \(UEFI\)](#) menu during boot.

The UEFI menu is different for each PC, but usually you press DELETE or F12 or sometimes even ESC to enter it. There might even be an item called “Enter Setup” or something like that. It is able to manipulate certain hardware options on the computer at a lower level than the operating system. Mac users won’t have to worry about this. You’ll know if you have to do this if you see this error:

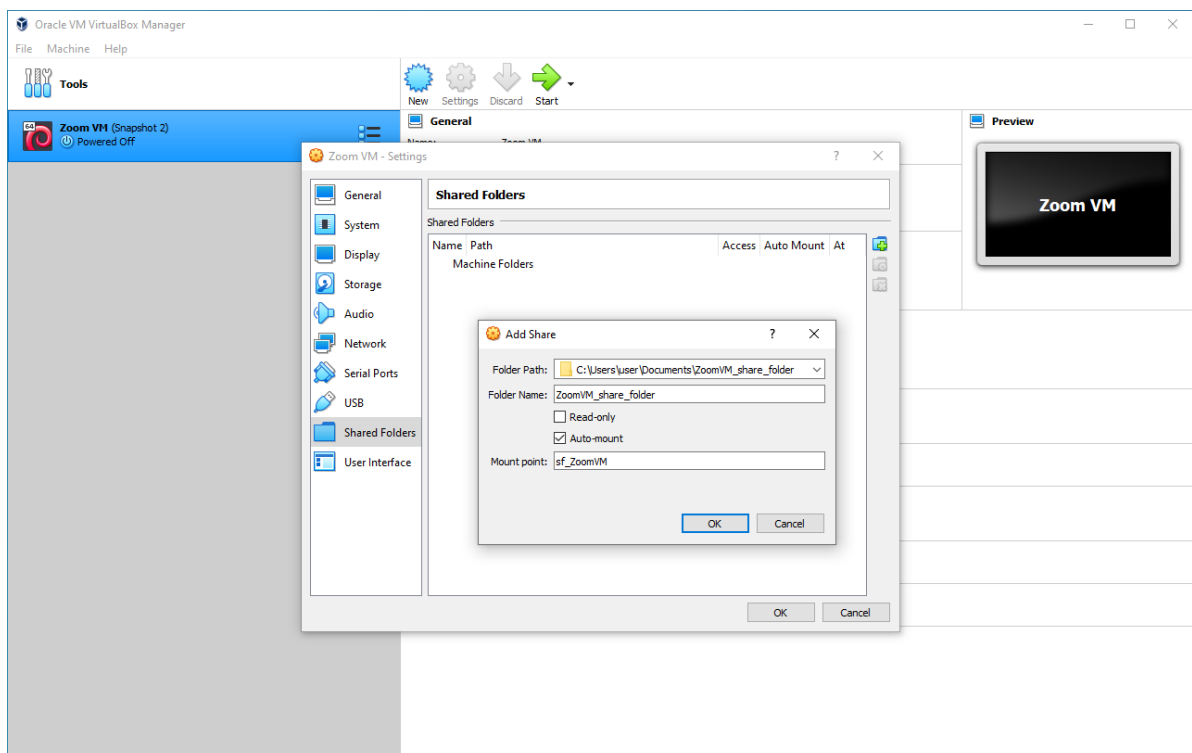




11. Mount a Debian CD to install

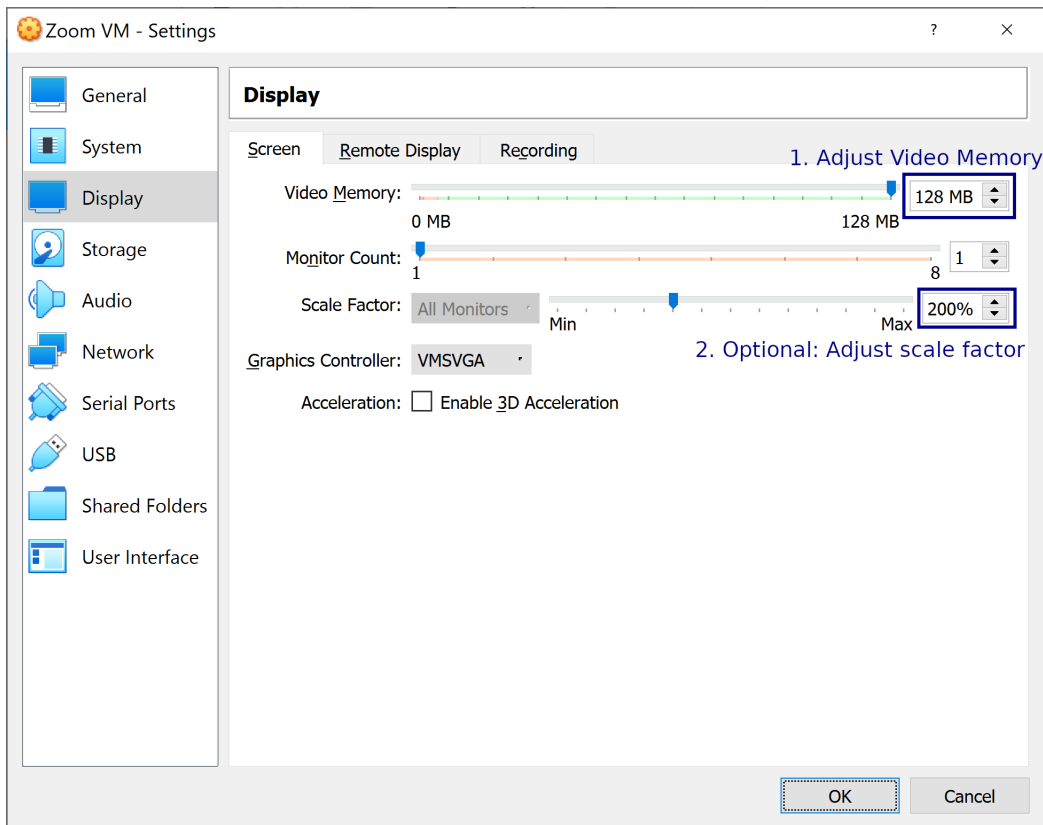


12. Add a shared folder, this is used for transferring files into the VM





13. Adjust video memory.



14. Optional: If you have a [4K UHD](#) screen you may want to set a scale factor under “Settings” → “Display”. This includes Macs with a [Retina display](#) for example.
15. Click on the green “Start” arrow to start the VM. Run the through the Debian installer. If you need help with that take a look at [Chapter 6. Using the Debian Installer](#).

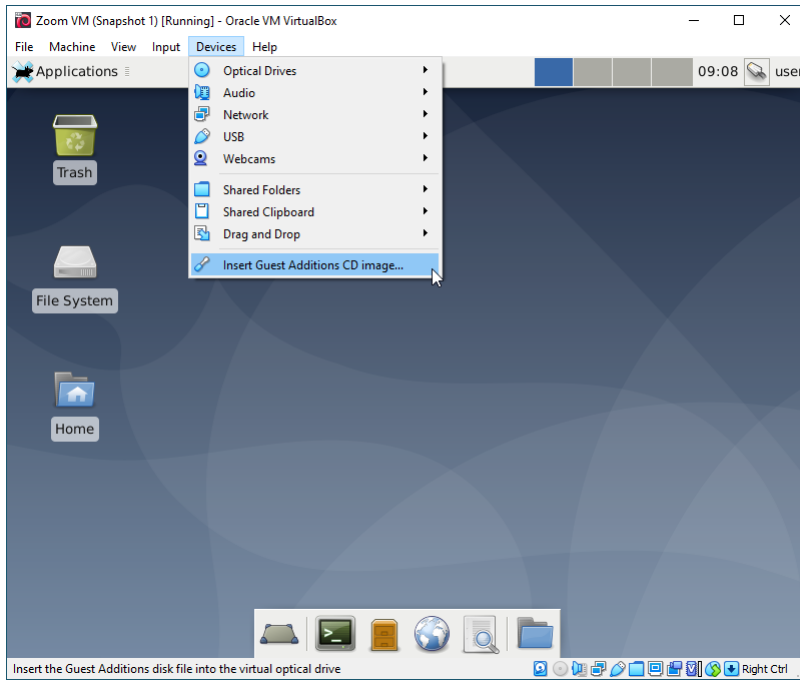


16. Optional: If you’re on a Linux host make sure to [add yourself to the vboxusers group](#) so you can use shared folders in your guest.

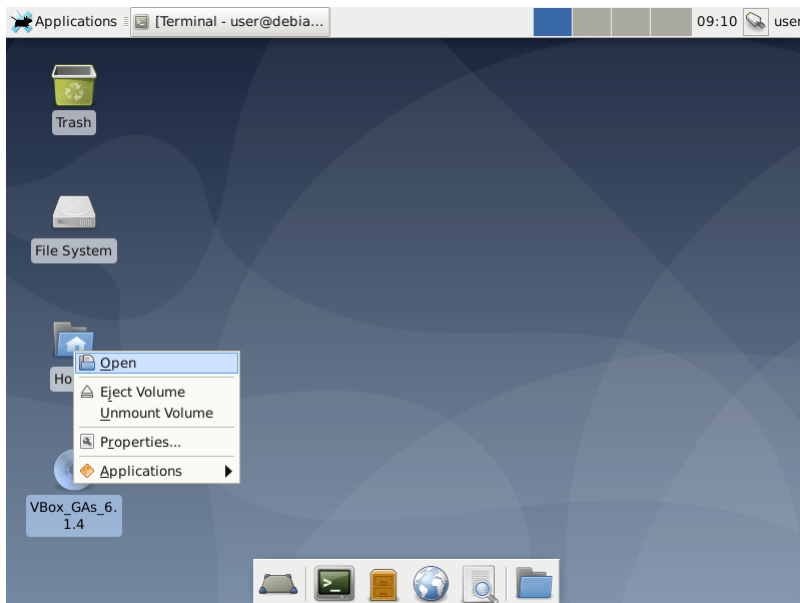
```
Input
sudo gpasswd -a $USER vboxusers
```



17. Once the virtual machine has started install the guest tools.

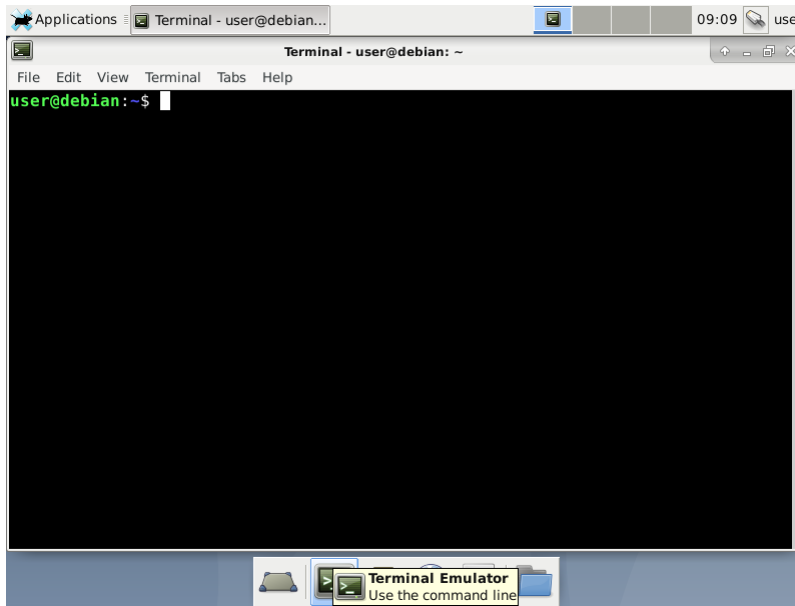


18. Open the CD that appeared on the desktop. The contents will be mounted to `/media/cdrom`





19. Open a terminal within the VM



20. Now install the guest tools. We need to install some development tools such as a compiler and kernel headers.

#### Input

```
sudo apt install build-essential linux-headers-$(uname -r)
cd /media/cdrom
sudo sh ./VBoxLinuxAdditions.run
```

21. In the VM you'll want to add yourself to the `video` and `vboxsf` group. Omit the video group if you don't plan on using your webcam in the virtual machine

#### Input

```
sudo gpasswd -a $USER video
sudo gpasswd -a $USER vboxsf
```

22. In Windows you may need to enable your camera and microphone. This is in “Start” → “Settings”.



## Camera

### Allow access to the camera on this device

If you allow access, people using this device will be able to choose if their apps have camera access by using the settings on this page. Denying access blocks Microsoft Store apps and most desktop apps from accessing the camera. It does not block Windows Hello.

Camera access for this device is on

Change

1. Enable camera

### Allow apps to access your camera

If you allow access, you can choose which apps can access your camera by using the settings on this page. Denying access blocks apps from accessing your camera. It does not block Windows Hello.





☒ On 2. Allow apps to use camera

Some desktop apps may still be able to access your camera when settings on this page are off. [Find out why](#)

### Choose which Microsoft Store apps can access your camera

Turning off an app prevents it from directly accessing your camera. It does not prevent the app from accessing the camera indirectly through

## Camera

	Cortana	<input type="checkbox"/> Off
	Desktop App Web Viewer	<input type="checkbox"/> Off
	Microsoft Edge Sites still need permission	<input type="checkbox"/> Off
	Microsoft Store	<input type="checkbox"/> Off

### Allow desktop apps to access your camera

Some apps and Windows features need to access your camera to work as intended. Turning off this setting here might limit what desktop apps and Windows can do.

☒ On 3. Allow desktop apps

Some desktop apps might not appear in the following list or are not affected by this setting. [Find out why](#)

23. You will also need to enable access to your microphone if you wish to use that.

## Microphone

### Allow access to the microphone on this device

If you allow access, people using this device will be able to choose if their apps have microphone access by using the settings on this page. Denying access blocks Windows features, Microsoft Store apps, and most desktop apps from accessing the microphone.

Microphone access for this device is on

Change

1. Enable microphone

### Allow apps to access your microphone





If you allow access, you can choose which apps can access your microphone by using the settings on this page. Denying access blocks apps from accessing your microphone.

☒ On 2. Allow apps to use microphone

Some desktop apps may still be able to access your microphone when settings on this page are off. [Find out why](#)

If an app is using your microphone, you'll see this icon: 

## Microphone

	Desktop App Web Viewer	<input type="checkbox"/> Off
	Microsoft Edge Sites still need permission	<input type="checkbox"/> Off
	Microsoft Store	<input type="checkbox"/> Off
	Take a Test	<input type="checkbox"/> Off

### Allow desktop apps to access your microphone

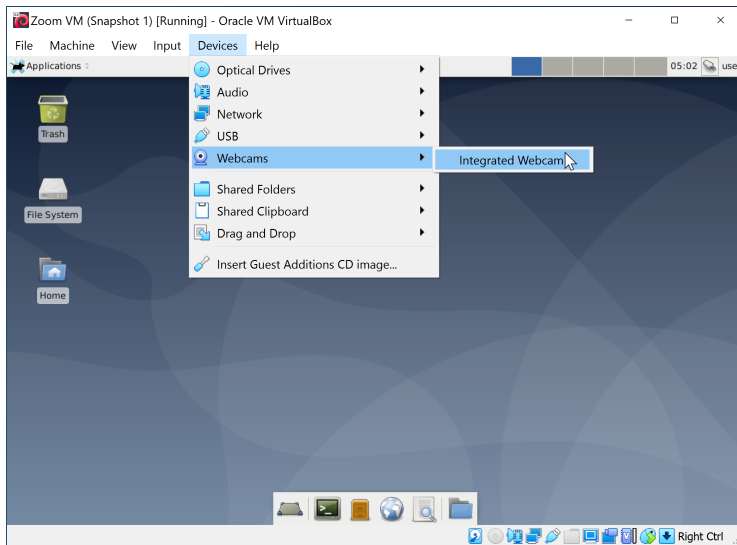
Some apps and Windows features need to access your microphone to work as intended. Turning off this setting here might limit what desktop apps and Windows can do.

☒ On 3. Allow desktop apps

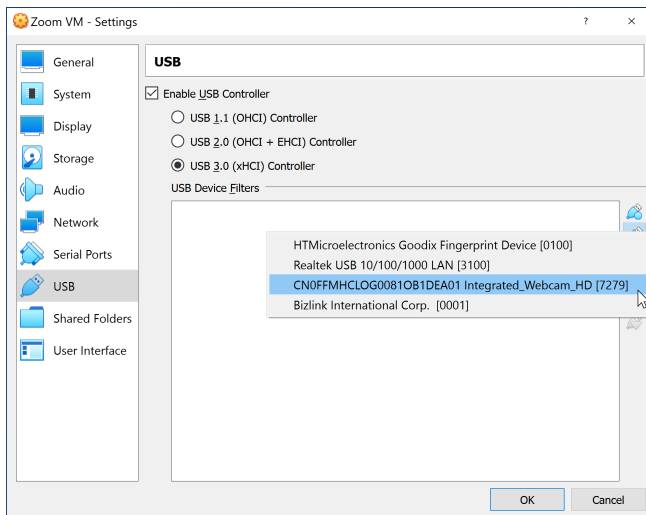
Some desktop apps might not appear in the following list or are not affected by this setting. [Find out why](#)



24. Optional: Add the camera to the VM. See the “Devices” → “Webcams” menu in the VM. Unfortunately some USB 2.0 cameras will not work. There is a related bug [No image with USB 2.0 webcams](#). Most will likely work however.



25. Optional: If the previous step didn't work, you can try to use USB passthrough instead. Start the VM guest. Once started go to “Settings” → “USB” and attach the camera..



26. Inside the VM you can test if the camera is working by trying to record something with a program called “gucvview”

#### Input

```
sudo apt install guvcview
```

You should then be able to open it from the menu, or by typing “gucvview” in the terminal.

27. Finally reboot the VM, you can use this command in Terminal or click reboot from the menu.

#### Input

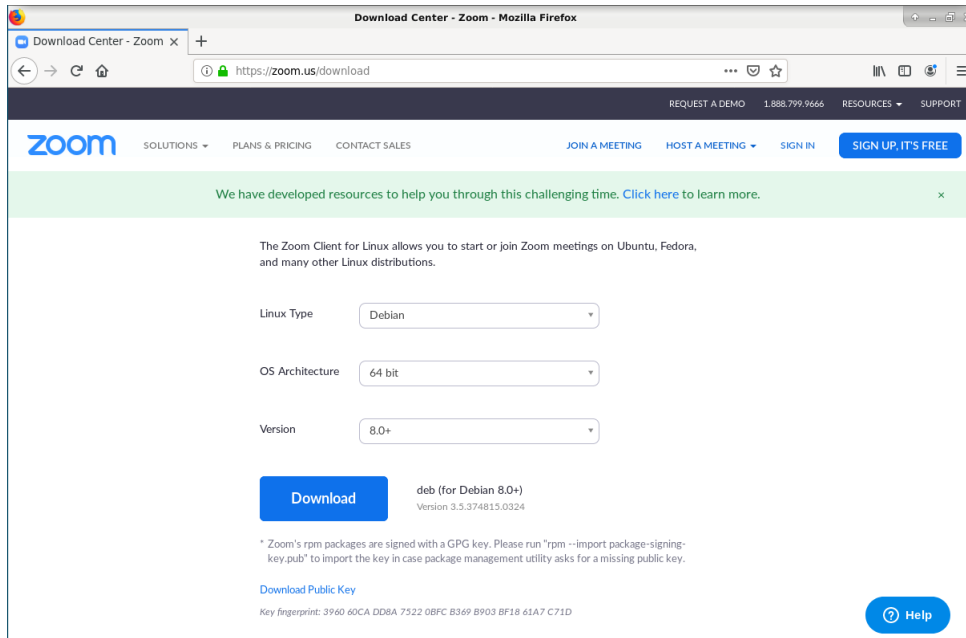
```
reboot
```





## 4 Installing Zoom

1. When signing up to Zoom, use a throwaway email address if possible.
2. Download Zoom. You can either use the desktop client:

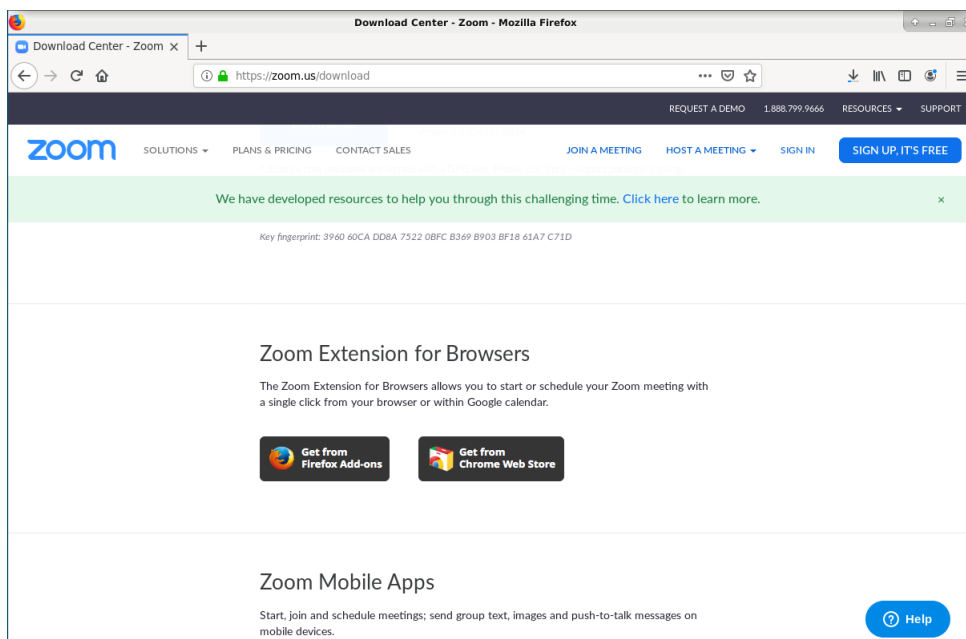


To install Zoom and add missing dependencies:

### Input

```
cd ~/Downloads
sudo apt install ./zoom*.deb
```

Or if you scroll down a bit further, the browser plugin:



If you do use the browser plugin you may find [zoom-redirector](#) useful. It will make sure that Zoom links you click on in your browser open in your browser.



## 5 Snapshots

3. It also might be a good idea to create a “Snapshot” after you’ve got everything working. Any persistent files will not be kept when you restore the Snapshot.

